



Theses and Dissertations

2006-01-12

A User-Centric Tabular Multi-Column Sorting Interface For Intact Transposition Of Columnar Data

David B. L. Miles
Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Databases and Information Systems Commons](#)

BYU ScholarsArchive Citation

Miles, David B. L., "A User-Centric Tabular Multi-Column Sorting Interface For Intact Transposition Of Columnar Data" (2006). *Theses and Dissertations*. 344.
<https://scholarsarchive.byu.edu/etd/344>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

A USER-CENTRIC TABULAR MULTI-COLUMN SORTING INTERFACE FOR
INTACT TRANSPOSITION OF COLUMNAR DATA

by

David B. L. Miles

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

School of Technology

Brigham Young University

April 2006

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

David B. L. Miles

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

_____	_____
Date	Dr. Ronald F. Gonzales, Chair
_____	_____
Date	Dr. Joseph J. Ekstrom
_____	_____
Date	Dr. Larry E. Wood

BRIGHAM YOUNG UNIVERSITY

FINAL READING APPROVAL

I have read the thesis of David B. L. Miles in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Ronald F. Gonzales
Chair, Graduate Committee

Approved for the department

Thomas L. Erekson
Director, School of Technology

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of Engineering
and Technology

ABSTRACT

A USER-CENTRIC TABULAR MULTI-COLUMN SORTING INTERFACE FOR INTACT TRANSPOSITION OF COLUMNAR DATA

David B. L. Miles

School of Technology

Master of Science

Many usability features designed in software applications are not procedurally intuitive for software users. A good example of software usability involves tabular sorting in a spreadsheet. Single-column sorting, activated with a mouse click to a column header or toolbar button, often produces rearranged listings that reduce cognitive organization beyond the sorted column. Multi-column sorting, generated through menu-driven processes, provides derived organization, however, locating feature options through menu-based systems can be confusing. A means to overcome this confusion is *prioritized selection of database arrays issued to columnar displays for the purpose of intact transposition of data*. This is a unique process designed as a user-centric tabular multi-column sorting interface. Designed within this experimental software application is a “trickle down” logic perceived as a navigation rule. The design offers logic

associated with decision choices as used to pursue a software solution. The solution in this instance is a compiled resultant of separate and distinct columnar data sorting.

The design was initially implemented in a software application housing thousands of examination scores. Observations of the design concept's effectiveness in practice led to further investigation through this master's thesis. To validate the research design, research participants were introduced to an example of traditional database sort/selection with practice examples. These users were also provided sorting exercises to reinforce the discussed concepts—both experimental and traditional. Finally, a survey questionnaire allowed them to provide feedback about the different task methods for sorting as well as the experience of using these dissimilar methods.

The hypothesis was not validated through the research survey. Consideration of observations of the design in a production environment for four years, however, provided impetus to suggest further research of the design concept.

ACKNOWLEDGEMENT

I would like to express sincere and deep appreciation to Dr. Ronald F. Gonzales for his enthusiasm, support, encouragement, guidance, and open door. He realized the research value of user interface designs that are developed to simplify user interactions. This initial assistance as well as continued suggestions, comments, and reviews have significantly contributed to this work. I am also grateful to Dr. Joseph J. Ekstrom, who has been a mentor for the past nine years as the former CTO (Chief Technology Officer) and site manager of SwitchSoft Systems and VPNX.com (Virtual Private Network Exchange), Chief Scientist at Internap Network Services, and now as one of my professors at BYU. His continued support and guidance have facilitated my work not only with this thesis but with my education in general and life pursuits. I appreciate Dr. Larry Wood for providing insights into the realm of user interface and providing guidance on the methodology for testing. I thank Dr. Dennis Eggett from the Statistics Department for his insights and guidance on matters of statistical significance.

I thank fellow students at Brigham Young University who have directly and indirectly supported my projects and education. I express appreciation to those providing opportunities that have fostered my ideas and understanding leading to this research project.

TABLE OF CONTENTS

A USER-CENTRIC TABULAR MULTI-COLUMN SORTING INTERFACE FOR INTACT TRANSPOSITION OF COLUMNAR DATA	i
GRADUATE COMMITTEE APPROVAL	ii
FINAL READING APPROVAL	iii
ABSTRACT	iv
ACKNOWLEDGEMENT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	ix
CHAPTER 1 INTRODUCTION.....	1
STATEMENT OF THE PROBLEM	9
PURPOSE OF THE STUDY	9
RESEARCH QUESTIONS	10
HYPOTHESES.....	10
LIMITATIONS OF THE STUDY	11
CHAPTER 2 REVIEW OF LITERATURE	13
CHAPTER 3 METHOD	33
RESEARCH DESIGN.....	33
■ <i>The Comparison</i>	34
■ <i>Assumptions</i>	41
■ <i>Population and Sampling</i>	44
■ <i>The Tool for Comparison (A Brief Overview)</i>	45
■ <i>Administrative Procedures</i>	47
■ <i>Criterion Measures</i>	47
■ <i>Statistical Treatment</i>	48
IMPLEMENTATION	49
MAINTAINING CONFIDENTIALITY	97
SUMMARY	97
CHAPTER 4 RESULTS AND FINDINGS	99
COMBINED RESULTS.....	100
NON-RELIABLE SURVEY FEEDBACK	100
RELIABLE SURVEY FEEDBACK.....	101
RELIABILITY OF EXERCISES USED IN SURVEY	107
ANSWERS TO QUESTIONS DISCUSSED IN CHAPTER 1	108
EVALUATIONS OF HYPOTHESES DISCUSSED IN CHAPTER 1	110
SUMMARY	111
CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS	113
CONCLUSIONS	113
RECOMMENDATIONS.....	114
REFERENCES	119

APPENDICES	123
APPENDIX A: DEFINITIONS AND TERMS	125
APPENDIX B: THE TABLE DATA	133
APPENDIX C: THE LIST OF ITEMS TO SORT	139
APPENDIX D: THE WEB PROGRAM CODE	151
<i>Appendix D.1 - default.asp</i>	<i>153</i>
<i>Appendix D.2 - subAndFuncAdmin.asp</i>	<i>157</i>
<i>Appendix D.3 - subroutinesAndFunctions.asp.....</i>	<i>179</i>
<i>Appendix D.4 - thesisAdmin.asp</i>	<i>219</i>
<i>Appendix D.5 - thesisDatabaseConnection.asp</i>	<i>221</i>
APPENDIX E: THE EXERCISE RESULTS	223
APPENDIX F: EXERCISE RESULTS - NO OUTLIERS / OUTLIERS REMOVED.....	233
APPENDIX G: EXERCISE RESULTS RELIABILITY	243
APPENDIX H: THE SURVEY	251
APPENDIX I: THE SURVEY/QUESTIONNAIRE RESULTS	257
APPENDIX J: THE QUESTIONNAIRE RESULTS - SUMMARY	263
APPENDIX K: GRAPHS OF QUESTIONNAIRE SUMMARY RESULTS	265
APPENDIX L: QUESTIONNAIRE RESPONSE DATA RELIABILITY	277

LIST OF FIGURES

Figure 2.1	Columns vs Rows	15
Figure 3.1	Patterns: Single-Column Sorting	34
Figure 3.2	Multi-Column, All Columns, Intact Column-Data Transposition	35
Figure 3.3	Standard/Single Column Sorting Methodology – Microsoft Excel /1	38
Figure 3.4	Standard/Single Column Sorting Methodology – Microsoft Excel /2	39
Figure 3.5	Prioritized Multi-Column Sorting Example 1	40
Figure 3.6	Prioritized Multi-Column Sorting Example 2	41
Figure 3.7	Multi-Column Sorting For Some Columns, No Column Data Transposition	43
Figure 3.8	Study Flow Diagram	46
Figure 3.9	Introduction as Found on Website	53
Figure 3.10	Wording From Website Survey	54
Figure 3.11	Website Snapshot: Introduction	55
Figure 3.12	Wording for Standard Sort View Methodology Explanation	56
Figure 3.13	Website Snapshot: Standard Sort View Methodology Explanation	57
Figure 3.14	Wording for Multi-Column Sort Methodology Explanation	58
Figure 3.15	Website Snapshot: Introduction, Prioritized Left-to-Right Column Sort Explanation	59
Figure 3.16	Wording for the Left-to-Right Prioritized Multi-Column Sort Methodology	60
Figure 3.17	Website Snapshot: More Full View, Prioritized Left-to-Right Multi-Column Sort	61
Figure 3.18	Website Snapshot: Introduction, Visual Comparison of the Two Methodologies, View I	62
Figure 3.19	Website Snapshot: Introduction, Visual Comparison of the Two Methodologies, View II	63
Figure 3.20	Website Snapshot: Introduction, Visual Comparison of the Two Methodologies, View III	64
Figure 3.21	Website Snapshot: Sample Exercise 1, View I	67
Figure 3.22	Website Snapshot: Sample Exercise 1, View II	68
Figure 3.23	Website Snapshot: Sample Exercise 1, View III	69
Figure 3.24	Website Snapshot: Sample Exercise 1, View IV	70
Figure 3.25	Website Snapshot: Sample Exercise 2, View I	71
Figure 3.26	Website Snapshot: Sample Exercise 2, View II	72
Figure 3.27	Website Snapshot: Sample Exercise 2, View III	73
Figure 3.28	Website Snapshot: Sample Exercise 3, View I	74
Figure 3.29	Website Snapshot: Sample Exercise 3, View II	75
Figure 3.30	Website Snapshot: Sample Exercise 4	76

Figure 3.31	Website Snapshot: Sample Exercise 5, View I	77
Figure 3.32	Website Snapshot: Sample Exercise 5, View II	78
Figure 3.33	Website Snapshot: Sample Exercise 6, View I	79
Figure 3.34	Website Snapshot: Sample Exercise 6, View II	80
Figure 3.35	Website Snapshot: Questionnaire, View I	89
Figure 3.36	Website Snapshot: Questionnaire, View II	90
Figure 3.37	Website Snapshot: Questionnaire, View III	91
Figure 3.38	Website Snapshot: Questionnaire, View IV	92
Figure 3.39	Website Snapshot: Questionnaire, View V	93
Figure 3.40	Website Snapshot: Questionnaire, View VI	94
Figure 3.41	Website Snapshot: Questionnaire, View VII	95
Figure 3.42	Website Snapshot: Exercise Results	96
Figure 3.43	Website Snapshot: Multiple Choice Questionnaire Results	96
Figure 3.44	Website Snapshot: Participants List	97
Figure 4.1	Survey Questions With Reliable Results And Answered With Disagree, Neutral, or Agree Responses (or Strongly Disagree/Agree)	104
Figure 4.2	Survey Questions With Reliable Results And Answered With Single\ Column Sort, Multiple-Column Sort or Both	105
Figure 4.3	Survey Questions With Reliable Results And Answered With True, False, and Not Sure	106

CHAPTER 1

INTRODUCTION

This thesis topic was motivated by the need to create an application that (1) required software users to input various records and extract tabular reports; (2) was practical for intermittent, untrained users; and (3) minimized the difficulty in adapting users' habits and interactions with existing systems, thereby reducing the tendency to revert back to manual systems and corresponding loss of employee productivity. The resulting application implemented the feature "Prioritized selection of database arrays issued to columnar displays for the purpose of intact transposition of data." This feature is a process designed as a user-centric tabular multi-column sorting software application interface. This thesis proposes this process, by design, as a more intuitive approach to multi-column sorting as compared with other sorting methods reviewed and reported within this research. This feature was incorporated into a software application housing thousands of examination scores. New records were appended on a daily or weekly basis. Reports formatted with this feature were drawn on a periodic basis to selectively match job requirements to individuals or provide status updates on individuals, groups or the organization as a whole. As referenced in this thesis, the term "user" refers to a person whose job or preference involves the task of navigating within software applications for the purpose of successfully completing the task.

Harold Thimbleby (2001), a usability scholar, noted the negative impact of non-usable or non-user-friendly systems. He stated that failed systems typically result from a lack of user-centered design, and poor usability designs are often correlated with program failures in the marketplace. The production and development costs saved on inadequate software designs, as associated with less skill and budget, would likely be consumed in redesigns, extra support staff to lead users through the program, and loss of needed but exasperated customers. In brief, inadequate consideration of software usability often results in negative consequences.

Thimbleby (2001, p. 10) asserted that “users deserve as much attention in pre-production design of programs as the computers get in their programs.” Implied within his comment are those numerous design considerations that iteratively occur for computing hardware and software compatibility, but which do not occur for software usability. A cliché that has permeated the computer industry helps put his comments into perspective. It has been stated that the easier a software application is made for a user, the more difficult and time consuming it becomes for programmers to produce; conversely, the easier a software application is to program, the more difficult and time-consuming it becomes for users to operate. The essence of the cliché is similar to Thimbleby’s assertion in that they both express the need to dedicate more time and consideration to assure usability.

Usability experts, such as Thimbleby, researched the essentials required to enable humans to interface with computers successfully. Investigations highlighted learnability, efficiency, memorability, error handling, and user satisfaction (Drier, 2004). Jakob Nielsen (2005) furnished usability heuristics that were stronger than mere suggestions;

these procedures have become industry accepted principles of software usability. Failure to follow these methods or best practices have resulted in instances of negative usability interpretations. That is, users are failing to correctly discern intended meanings. The ten heuristics of software usability that Nielsen advocated were:

1. “Visibility of system status: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time” (Nielsen, 2005, p. 1).
2. “Match between system and the real world: The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order” (Nielsen, 2005, p. 1).
3. “User control and freedom: Users often choose system functions by mistake and will need a clearly marked ‘emergency exit’ to leave the unwanted state without having to go through an extended dialogue. Support undo and redo” (Nielsen, 2005, p. 1).
4. “Consistency and standards: Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions” (Nielsen, 2005, p. 1).
5. “Error prevention: Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action” (Nielsen, 2005, p. 1).

6. “Recognition rather than recall: Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate” (Nielsen, 2005, p. 1).
7. “Flexibility and efficiency of use: Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions” (Nielsen, 2005, p. 1).
8. “Aesthetic and minimalist design: Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility” (Nielsen, 2005, p. 1).
9. “Help users recognize, diagnose, and recover from errors: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution” (Nielsen, 2005, p. 1).
10. “Help and documentation: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large” (Nielsen, 2005, p. 1).

According to Nielsen’s research, the incorporation of these software usability essentials contributes to the success of human computer interfacing.

For software applications whose scope is long-term and intended for multiple customers and users, these design iterations that have incorporated the usability heuristics have achieved successful results. Examples of successful tabular user interfaces are found in numerous electronic mail applications. Email interfaces display basic information for each email record such as a name, subject and date. Records are sorted by the contents of one column or another (single-column sorting). This sorting practice facilitates the rapid reorganization of record entries in either ascending or descending letters, numbers, or both letters and numbers. The data arrays maintain their integrity regardless of their displayed order in a recompiled list. A user-friendly mechanism for selecting a column data sort incorporates basic logic of human choice as determined by prioritization. When the user clicks the header, sorting occurs on that column. The key to success for these tabular interface applications can be found in Nielsen's heuristics. A key heuristic associated with the concept of logic followed the sixth heuristic which discussed recognition rather than recall. The principle of recognition is understanding immediately both the meaning and what to do upon seeing the prompt. By virtue of what is displayed and how it appears, intuition processes a cognitive mapping that a user might normally and naturally select as a column header to trigger the desired sorting result.

Other tabular applications meeting similar usability success are spreadsheets such as Microsoft Corporation's Excel. Spreadsheets have a relatively easy method for users to perform sorting based upon the contents of one column. Spreadsheets allow for generalized content input with each column representing a category or field of information and rows representing a distinct record from each of the categories. As an example, each column could represent a telephone book data field such as last name, first

name, telephone number, or address. Rows represent data associated with entries for a single person. Users select a column for sorting by placing the cursor into one of its fields, then clicking on a button on the tool/menu bar. New users recognize the association of fields and records and quickly learn the process to sort data entries.

The process for performing single column sorting in some spreadsheet applications is easy for users based on Nielsen's heuristics. Only two steps are required in a single column sort: (1) place the cursor in any field within the column to be sorted, and (2) click an icon on the tool bar for sorting. As noted by using a combination of Nielsen's sixth and eighth heuristics, reducing this overall process to only two clicks has improved the usability.

Even though the single column sort has been useful, the procedure has not been effective at organizing data when involving multiple rows where some columns contain and then display identical data. For example, numerous people share the same last name, such as Smith. In single-column sorting based upon the last name column, all rows with Smith as the last name would have the other columns appear unsystematically. Four records with the last name of Smith might have the first names appear in the order of John, David, Zac, and Jed; the telephone numbers might appear as 555-3456, 555-2323, 555-6756, and 555-1234. Similarly, addresses might not sort in any logical order. When numerous records mirror identical data, visual scanning through the disorganized arrays can become awkward. If the contents of the previous telephone book example were found in a much larger area edition, a list of those with Smith as the last name might not only include John, David, Zac, and Jed, but also Ted, Jeremiah, Adam, Michael, Joseph, Emma, Mary, Isaiah, Lori, Tana, Herman, Leonard, Ronald, Zena, Larry, Michael Jr.,

Steve, Laura, Fred, Wilma, Barney, Andy, Andrew, Robert, and Renee. The solution to this disorganization is to sort by more than one column—the usability solution is a combination of Nielsen’s first, second, sixth and eighth heuristics.

Sorting on more than one column provides a more organized, thus more user-friendly, display of information. As an example, consider again the concept of the telephone book presented earlier with last name, first name, address, and telephone number. Visual scanning required users to depend on secondary and tertiary column fields (first name and address) to scan through rows of personal information when a person’s last and first name are shared by other patrons in the phone listing. Quick visual scanning through multiple rows of personal information in a phone book is possible because listings are displayed in a sorted, multi-column manner. Viewing information that is organized logically through all of its columns is much easier than viewing information that is not organized logically.

Sorting by more than one column adds a dimension of complexity handled in a non-intuitive, non-heuristic manner by spreadsheet designers. A sequence of dependent steps requires recall rather than recognition. A user has to remember which menu and series of submenu items are needed to open the dialog box for sorting multiple columns. A software dialog box offers the user a selection in which up to three of the available columns could be sorted; the placement of these selections is prioritized/sequential, indicating that the topmost selection would be sorted first while the succeeding columns are sorted in descending order. The process of accessing this feature can be so daunting that a software user frequently prefers only a single column sort. The need to complete additional column sorting is often entered into as a series of single column sorts.

Accessing multi-column sorting within current applications requires additional conceptual processes as compared to single-column sorting. This added thought appears to be the result of designers failing to consider some of Nielsen's usability heuristics. First, the user has to recall rather than recognize; the user has to pause and consider the steps to locate this function. Second, the user has to remember system terms such as *data* on a menu prior to choosing the desired *sort* function. When users are unable to remember how to access the desired function, they have to turn to a help system, if one exists. And if they have to use "help", the experience raises issues of user recall and terminology; users have to remember proper terminology to search for necessary help instructions.

In summary, sorting in a tabular environment involves simple concepts that are made complex by virtue of recalled events. Current applications require users to recall many steps. If users are not able to recall the sequence, they need to recall the proper terms to request assistance from the help menu—if a help menu is available; or, they need to ask another person. Software designers often fail to consider usability issues during the design and production phase. Consequently, designers and programmers regularly engage users in a software application lacking sufficient clues to establish clear navigation practices. This experience results in user frustration, leading to reduced or improper usage, and a search for software applications that more satisfactorily and intuitively reinforce cognition or the mental practice of successful navigating software routines.

STATEMENT OF THE PROBLEM

Software programs succeed to the extent users can grasp a program's logic and effectively use functional procedures. Lack of usability confuses users, and restricts their level of skill functionality. The inability to successfully navigate within a program contributes to a users sense of frustration. Designers need to provide sufficient interaction within programs to establish strong linkages of cognition and successful navigation. The research associated with this thesis attempted to demonstrate that, when designers and developers follow guidelines based on usability design principles as contained in the experimental model, designers and developers will find that these usability design principles contribute to the user's ability to effectively navigate within a software application. The experimental model is titled *Prioritized selection of database arrays issued to columnar displays for the purpose of intact transposition of data*. It is referred to as PSDA in this paper. PSDA simplifies column sorting for users of varying levels of computer expertise, reduces line-by-line search time for specific records, increases accuracy for choosing among desired records, and provides users a positive experience. This design feature has produced usability benefits by applying user-centric design techniques in constructing its sorting interface.

PURPOSE OF THE STUDY

The purpose of this research was to design a software programming process as an effective interface for the purpose of facilitating software usability of multi-column database arrays for the purpose of completing an intact transposition of data.

RESEARCH QUESTIONS

- Question 1: Are users more easily able to facilitate a multi-column data array sort with the experimental model as compared to a contemporary popular method used to facilitate a sort?
- Question 2: Does use of the experimental model demonstrate a reduced line-by-line search time for specific records of information?
- Question 3: Does use of the experimental model demonstrate a higher level of accuracy for selecting desired records from a listing when this model is compared to a contemporary popular method for facilitating a sort?
- Question 4: Does use of the experimental model provide users with a more pleasant and desirable experience when this model is compared to a contemporary popular method for facilitating a sort?

HYPOTHESES

- Hypothesis 1: Users more easily facilitate a sort with the experimental model when this model is compared to a contemporary popular method for facilitating a sort.
- Hypothesis 2: Use of the experimental model shows a reduced line-by-line search time for specific records of information when this model is compared to a contemporary popular method for facilitating a sort.
- Hypothesis 3: Use of the experimental model shows a higher level of accuracy for selecting desired records from a listing that includes undesirable records when this model is compared to a contemporary popular method for facilitating a sort.

Hypothesis 4: Use of the experimental model provides the user with a more pleasant and desirable experience when this model is compared to a contemporary popular method for facilitating a sort.

LIMITATIONS OF THE STUDY

Limitation 1: This study is intended to be a preliminary investigation into the usefulness of PSDA.

Limitation 2: The study only addresses the questions and hypotheses as stated in the research questions and hypotheses sections.

Limitation 3: The study uses participants from a broad set of backgrounds, experiences, understandings, perspectives, and ages.

Limitation 4: The study does not use a randomization of participants representative of a type of group because finding a substantive size group is not possible.

Limitation 5: The group of participants used within this study is a sub-element of a greater number of individuals given the opportunity to participate. In short, those who took part were those willing to volunteer.

Limitation 6: The research task provided real life exercises that were representative of a tabular display listing with sorting capabilities. When it displayed the sort direction for the date field, the arrow that indicated latest to earliest showed the big end as representing the earlier dates and the tip representing later dates. This did not follow a convention as noted with Microsoft Corporation's Outlook Express program, which would have the representations reversed. This discovery was made after the data collection was finished. No changes were subsequently realized. The

error was not deemed by advisors to be sufficiently critical to redo the study.

CHAPTER 2

REVIEW OF LITERATURE

The following review of literature is concerned with the concept of *prioritized selection of database arrays issued to columnar displays for the purpose of intact transposition of data* (PSDA) as a usability solution for tabular sorting. This topic area involves a perception of difficulty that humans display in their practice of using computer programs. The ability of a person to successfully navigate and complete a computing task implies the need for software to be designed to facilitate humans' perception of their computing needs.

Astrachan (2003) authored *Bubble Sort: An Archaeological Algorithmic Analysis*, in which he stated that algorithms for sorting with computers have existed since the mid-fifties. This was approximately ten to fifteen years after the introduction of the first electronic computers. This relative early appearance of electronic sorting is significant because it demonstrated the desirable nature of organizing and presenting data base records.

Peter and Phyllis Mackie (1984) authored *Spreadsheet Dilemma*, a book about electronic spreadsheets when spreadsheets were new. They defined spreadsheet design and provided a brief history of spreadsheets. Finally, they described its functionality and the features available in the earliest spreadsheets. The Mackies defined a spreadsheet as "A computerized electronic worksheet" (1984, p. 1). They stated that a worksheet is also

known as “a columnar pad, accountant’s pad, ledger, and ledger sheet” (Mackie, 1984, p. 2). In its original form, a spreadsheet was an electronic a columnar pad, accountant’s pad, ledger, and ledger sheet.

The first spreadsheet, VisiCalc, developed by Robert Frankston and Dan Bricklin, was released in 1978 (Mackie, 1984). “VisiCalc combined calculator and personal computer capabilities and emerged to replace the more traditional pencil, calculator, and columnar pad” (Mackie, 1984, p. 7). Within a few years, several dozen other spreadsheets became available (Mackie, 1984). Considering their popularity, electronic spreadsheets appear to have filled a void and/or created a demand within the business profession in a relatively short period of time.

The Mackies’ overview of spreadsheet core concepts included tabular formatting (Mackie, 1984). The essence of tabular formatting is based on a design of grid-like intersections of rows and columns. Rows represent records of information and columns indicate a logical categorization of record data known as fields (refer to figure 2.1). A telephone directory, for example, provides a display of customer records in columns of last name, first name, address and telephone number. Records as rows and individual record categories in specific columns is fundamental to a logical display of database contents.

Functions for sorting rows did not exist in the earliest spreadsheet programs (Mackie, 1984). The lack of automated sorting in early applications would have been normal because spreadsheets were initially used for data entry accounting. Later, feature sets expanded to include sorting.

The availability of electronic tools for displaying data in a columnar fashion filled a need for quick retrieval of data records. However, as with many inventions, it also created a new set of challenges. Spreadsheet programs were comparatively simple in their initial designs and use. However, as stated by the Mackies, these applications did not have sorting capabilities (1984). Over time, spreadsheet programs advanced to include capabilities for sorting contents of columns and rows. A challenge posed by these improvements was how operators could access data records and retrieve them in varying formats.

Column 1	Column 2	Column 3	Column 4	Column 5

Row 1				
Row 2				
Row 3				
Row 4				
Row 5				

Figure 2.1 Columns Vs Rows

An initial step to improving human computer interactions is to determine what and where usability issues exist. According to Treu (1994), those stated elements related to usability include “the physical surface and facilities between human user and computer which provide the medium through which they can connect and interact.” (1994, p. 9).

Treu also stated that HCI includes “the physical (visual, audio, tactile) means, methods, and patterns that support human-computer interaction” (1994, p. 9). The interaction itself is “the combination of physical, logical, conceptual, and language-based actions between a human user and a computer, toward achieving some purpose” (Treu, 1994, p. 9). Usability issues surface in a number of ways. When considering how to access or use a feature for sorting one or more columns, an operator might think, “How was I supposed to get to that sorting feature?” or “How was I supposed to use that sorting button? When I click it, nothing happens.” In brief, there are numerous ways human operators can become confused while using computers.

Martin and Eastman (1996) classify computer users as either *intuitive* or *hands-on* (Martin, 1996). “Intuitive people build large abstract models or worlds in their minds which to them are as real as reality. Hands-on people are practical and prefer empirical knowledge. They don’t truly understand anything until they have had their hands on it” (Martin, 1996, p. 38). Intuitive people conceptualize within their minds while hands-on people find understanding through visualizing with tangible items outside of their minds. As a result, intuitive users often prefer command line interfaces where, among other things, they can control the minutest details of a computer system. Conversely, hands-on users prefer graphical user interfaces (GUIs) where process details are hidden and understanding is simply the cause and effect relationship of clicking a button or menu item and having an action take place. These differences associated with people’s perspectives affect how individual users might generally prefer to interact with the computer programs. Since hands-on personalities compose a high percentage of the

population, there has been great motivation to improve graphical user interfaces to aid these users (Martin, 1996).

Given the consideration for designing by preferences, the question emerges asking to which audience PSDA might be more suited. PSDA is not command based; it is based on the graphical user interface model. As a result, it is evident that although intended for all audiences, PSDA, is more suited for individuals with the hands-on personality.

In his book, *The Design of Everyday Things*, Norman (2002) states that “the human mind was tailored to make sense of the world; when the mind was given the slightest clue it immediately uses this information to give explanation, rationalization, and understanding” (2002, p. 2). He believes humans are capable of deriving meaning correctly from hints or clues in program designs. These hints come in many forms such as colors, shapes, embedded pictures, or object location. Clues allow users to accomplish tasks quickly because they are prompted to choose from among available options. Hints reduce the chance for users to experience frustration from being unable to deduce what to do.

Norman uses everyday objects, such as televisions and radios, to show the difference between well-designed objects, that are easy to interpret and understand, and poorly designed objects, that are difficult to use and understand (2002). By reviewing Norman’s examples in detail, designers can draw analogies and determine which level of usability their program designs are providing to users.

Norman (2002) said one of the most important characteristics of good design is visibility; he wrote, “It is a lack of visibility that makes so many computer-controlled

devices so difficult to operate” (Norman, 2002, p. 8). He highlights the importance of natural signals to provide immediate interpretation of use or functionality. Norman cites the design use of doors as examples of good or poor usability. If a door has a handle, there is a natural tendency to pull; if the door has a plate, there is a natural tendency to push. If a door with a handle requires pushing to open, it has a poor design. A poor design would be a push plate on a door requiring a pull to open. These concepts of good and poor design can also be applied to software applications. One way is to ask usability questions such as, “Is such and such symbol being used appropriately?” Another question could be, “Is what I think the users are going to think when they come across this symbol in fact what they will usually think?” In essence, good designs rely on properly used visible cues.

Norman (2002) has also proposed the use of visual cues to help describe the benefit of “visualization.” Norman’s (2002) models consist of *affordances*, *constraints*, and *mappings*. Technically speaking, *mappings* are indicators of what to do by determining what appears possible. *Constraints* are indicators of the limits of what can be done. *Affordances* are indicators for showing what is allowed to be done. To show the relationship among the three parts of his model, Norman cites an example using scissors. A pair of scissors has holes considered *affordances*; holes show what is used to perform an action. The size of the holes would be considered the *constraints* because the hole size indicates the limit of the size of the fingers designed to fit the holes. In a way, *mapping* ties constraints and affordances together by describing the courses of action possible. This modeling of usability has helped to describe visualization and to pin-point where and why good visual cues exist within designs.

An additional point Norman (2002) makes about models and other usability issues is that they must take into consideration the user's point of view. Consider the previous scissor example. If a user had never seen a pair of scissors, he might not know how to use these when he came across them. However, having not used a pair of scissors for a relatively long period of time, the operator would generally draw from previous experience and recognize how to use these should they come into her possession. If the scissors were small cuticle scissors, hair scissors, or large (slightly modified to just handles and no holes) hedge cutters, the concepts would be the same and the function would be recognizable. The design of scissors provides easily recognizable mappings due to their affordances and constraints. Other easily recognizable mappings include toggle switches (these flip back and forth with force from a finger), push buttons (like on and off switches), spin dial (like a round door handle or "old" style radio tuning), and sliding controls (similar to temperature controls). Again, these and the scissors generally depend on an operator having had past experience as a basis for transferring to other situations or uses.

Norman's concept of using a person's point of view correlates directly to uses associated with computer applications. Take into consideration a hypothetical situation with Jane sitting at a computer. She notices an application available providing the following features: 1) a small square field, and 2) a button labeled *search*. She can see by the clues that this computer application allows her to input data within the text field, click the button labeled *search*, and depending upon the availability of the requested information, she should receive a listing of some type. If Jane uses another application with similar clues (small square field, button labeled *find* instead of *search*), she might

surmise that she will be able to perform a comparable search function. Her point of view is now affected by her experience with the search engine. According to Norman, her future experience with search engines whether they read *search*, *find*, or *locate* for that matter will be affected by the one she just completed.

Jane's experience is an account with an ideal result; she understands and is able to process what to do. There are situations where negative results occur. Consider the scissor example with a variation. If sharp, the scissor blades have the potential to cut more than the intended object. If the ends are pointed, they can pierce an eye or puncture the skin or other object. According to Norman (2002), designs should foresee and protect against the occurrence of "errors." A foreseen fix for sharp edges and pointed tips on children's scissors is dull blades with rounded tips. In a software scenario, an operator might press an unintended key or button which might result in exiting an unsaved program or turning off the computer in the middle of a process. Solutions to minimize damage can be programmed into software applications. As an example, some vendors provide applications that periodically save backups of open files. If there is a power loss or the programs shut down for any reason, the backup files remain. When these applications are run again, an option is given to users to open and review files containing the last saved instance. The designers of these programs foresaw potential mishaps with unsaved data and took into account ways that protected users from the potential harm; accidentally or purposely, they followed guidance provided by Norman to, in essence, protect users from themselves.

Norman (2002, p. 2) says "a well-designed object is easy to interpret and understand while a poorly designed object is difficult to use and causes frustration."

Within this distinction between good design and poor design, no reference is made to users' emotions when usability is high. However, the emotion of frustration is used to describe the user's feelings when the tool is difficult to use. Simple analysis might lead one to understand that a measure of success for a tool is not necessarily that the user is happy or excited about the tool, but rather, that the operator is not frustrated.

Norman (2002) presents compelling cases for usability in general and provides specific design recommendations. He also provides a rationale for usability as well as ways to consider usability when designing programs. Though he provides specific examples in different areas of expertise, his recommendations are general and available for incorporation in all fields. Norman (2002) highlights humans' ability to use clues to derive meaning from tools and use these clues to accomplish tasks. He discusses technical aspects of how meaning is derived and how designs can and should be made. While users may take for granted the fact that a tool works properly, Norman notes that the negative emotion of frustration occurs when things do not work properly; working tools result in a null emotional state while non-functioning tools result in a negative emotional state.

Another author, Robert W. Bailey, states that software needs to be designed so that it presents the user interface (UI) in naturally understandable ways (2001). When one considers the phrase "naturally understandable," one must realize that this implies that what the operator sees on his screen should correlate to something he has seen in the physical world or has experienced. As an example, software applications perform functions that inherently take time. Some of these applications display clocks with moving hands suggesting to the operator to have patience and wait for a process to

complete. Designers of these and other graphics that indicate progress recognize the correlation between these visual aids and users' backgrounds. An assumption is made that a great percentage of users would have experienced a situation at some point in their lives where they had to wait, as implied by a clock. In summary, Bailey infers that prompts, cues, and feedback should be designed to accommodate users.

Placement of objects in an interface are as much a part of feedback as objects and symbols. Bailey (2001, p. 280) says, "Probably the designer's most important consideration when arranging displays and controls is to make decisions that are consistent with what the user expects. With controls, for example, some direction-of-motion expectancies seem almost natural—pushing a throttle forward to increase speed or turning a wheel..." An additional example of Bailey's comment is the placement of speaker selection slide bars in an audio system. If a vertical slide bar is used to indicate output balance between two speakers, it would be difficult to decide which speaker is the one on top when physically each is on either side of the computer. If the same slide bar is placed horizontally such that movements to the left provide increasing output to the right speaker, the action and result would be considered counter-intuitive. The ideal would be to have the left speaker increase in output as a result of moving the slide bar to the left. Again, the placement and method of use of objects in an interface are as much a part of feedback as the objects and symbols.

Up until now, the discussion has mentioned some objects and symbols found within graphical user interfaces; these objects help support mechanisms for HCI within GUIs. Two distinct methods for interacting with users within a GUI are direct manipulation devices and menu driven interfaces. Menu driven interfaces provide an

orderly logical flow to features and utilities while direct manipulation devices complete tasks without involving intermediate steps.

Menu-driven interfaces implement a hierarchical menu system to gain access to specific features or utilities. The application interface generally contains a series of topics across the top such as “File,” “Edit,” and “Help.” Clicking on a topic will activate a menu containing terms for available functions. If a utility is clicked, it becomes activated; if a subtopic is clicked, another menu will appear containing more terms for utilities or more subtopics. The number of sub-layers can vary. The menu system requires a user to memorize (or have some other method for remembering) the path to get to particular utilities or features. Menu-based systems are one method for humans to interact with computers in a graphical user interface.

Direct Manipulation Devices (DMDs) are another method used to interact with users within a GUI. There are a variety of DMDs to include objects known as icons and buttons (Bailey, 2001). Just as the name implies, these devices are directly manipulated by operators (Bailey, 2001). Bailey described dialog boxes with fields and buttons as DMDs (2001); users could directly add, edit or delete the contents of the fields as well as press buttons to activate desired functions. In short, DMDs allow users to interact directly with applications to accomplish desired functions without intermediate steps.

Although there are benefits to reducing the number of steps to complete software usability processes, accuracy is not one of them. According to Bailey, direct access to functionality does not enhance accuracy above that which menu based systems provide (2001). As a result, it does not appear that structural differences between DMDs and menu based interfaces affect accuracy.

There are, however, performance differences between DMDs and menu based interfaces. DMDs lead to faster task performance for new and occasional users (Bailey, 2001). Completing procedures in fewer steps equates to accomplishing tasks quicker. Having fewer steps also equates to having less to remember. New and occasional users would have a reduced chance to get confused; they could master completing tasks more quickly. It appears that structural difference between DMDs and menu based interfaces allow users to accomplish tasks quicker with DMDs.

Fewer steps in processes are good with only one being best. Bailey (2001) asserts that an optimized condition of human performance involves a one-step operation to complete a task. Examples of one-step (for the user) processes include such tasks as opening an application and saving a file to disk. In each case, an operator can click a single button to execute a series of background processes that accomplish a task. Confusion and complications can arise from multi-step processes. As an example, some sorting tools require the user to place the cursor in the column that will be sorted and then click a button on the tool bar. All too often, users fail to position the cursor appropriately. They stop, they wonder, and when they realize their mistake, they place the cursor appropriately and then re-click the button. Bailey says, “the fastest performance is only possible if conscious thought can be avoided” (2001, p. 107). Stopping to wonder what is going wrong reduces efficiency. The sorting illustrated in the example might be accomplished more efficiently if operators need to only click on a column header to sort on a particular column; the trigger for activating the sorting process is found simultaneously in the column desired for sorting. As a note, single column

sorting based upon triggers found by clicking column headers is used in commercial applications.

Minimizing steps is important for effective software designs. According to Bailey (2001), tasks perceived as difficult take a longer time to complete than those perceived as being easy; whereas tasks perceived as being easy take less time (Bailey, 2001). Designers may want to consider the psychological advantages of minimizing difficulty in that time for performing tasks can be reduced both because it is made shorter and because it is perceived to be simpler.

An irony associated with performance involves task completion, functionality, and ease of use over the long-term impact, or lack thereof, as applied to each users' emotional states. One might surmise that user satisfaction might occur with processes that perform well, complete tasks, or are easy to use. According to Bailey, however, research has not shown a correlation between task completion or ease of use with satisfaction (2001). The lack of this type of correlation does not seem to be conclusive because other authors such as Mayhew present opposite findings.

By following Bailey's conclusions, however, one might suppose that user satisfaction is much like a fleeting fad. A user's initial response to new tools might be intense satisfaction because from the users' perspective these tools reduce complexity, decrease time to complete one or more processes, increase efficiency, or provide empowerment. As with fads, however, time passes and the user's elation with the tools shrinks; processes become routine with expectations that the improved results will continue. New problems, concerns, or shortcomings distract the user's attention.

Although a greater discussion on factors that affect user satisfaction is beyond the scope of this paper, re-visiting an issue discussed during the literature review on Norman seems appropriate. The idea behind an association between user satisfaction and application usability is to avoid keeping customers away from one's product. Providing functionality that does not produce amazement is satisfactory as long as it does not cause the user undue frustration. As an example, a standard shovel might not cause a customer overwhelming joy, but if the handle snaps or the blade cracks under normal conditions, the customer would be upset and dissatisfied with the tool. Bailey's (2001) conclusions are similar to Norman's observation; it appears that having no response, either negative or positive, is desirable. As long as the user is not upset, the design of the product is adequate.

In summary, Bailey makes a number of important points. He articulates a simplistic definition of human and computer interactions. He expresses similar conclusions to those of Norman regarding the need for designs to be properly understood. He believed direct manipulation devices do not reduce errors but help decrease the time required for new users to become productive. He emphasizes the optimized state of a one-step process. He maintains that prompts, cues, and feedback are important usability tools when they are easy to understand. Finally, his interpretation of his personal findings indicates that it is difficult to create or measure an association between user satisfaction and functionality or usability.

Mayhew (1992) provided insights into visibility issues and discusses direct manipulation interfaces (DMIs) considerably. Her research helps to establish methods

for improving performance, which can then be associated with reduced eyes-on line-by-line search time in properly sorted tabular displays.

Direct manipulation was discussed previously by Bailey (2001); a discussion presented in Mayhew's book provides a baseline for DMIs in this research. According to Mayhew (1992, p. 289), "The term direct manipulation was first coined by Shneiderman ... to describe new interfaces having the following characteristics: (1) Continuous representation of objects; (2) Physical actions or labeled button presses in place of command language; and (3) Rapid incremental reversible operations with immediately visible results." For the most part, Mayhew emphasizes visibility as a core issue with DMIs.

Mayhew continues by describing technical tools that make these interfaces functional and the interaction possible. She says, "Sometimes called 'point-and-select' interfaces, direct manipulation interfaces often include a pointing device such as a mouse, trackball, or touch screen and often make heavy use of graphics in displaying objects and actions" (Mayhew, 1992, p. 289). In brief, the tools used to control DMIs are relatively simple, at least from the user's perspective.

Humans can be far removed from understanding how a process is achieved; as a result, a person's natural attributes are harnessed for message communication. Mayhew (1992) believes direct manipulation dialogs exploit visual-spatial cues and are less prone to error. She also states that these provide context and instant, visual feedback (Mayhew, 1992).

Considering a combination of issues discussed by Mayhew and other authors, DMIs are very helpful for users. Some of the advantages of direct manipulation dialogs

are that they are easily learned and remembered (Mayhew, 1992). Direct manipulation interfaces provide an environment in which a user does not need to recall terms from memory as is necessary when using command line interfaces. Rather, users can use a mouse pointer or can tab over to text and *touch* an object such as a word or figure; and, if an action is associated with the word or figure, the action triggers. An important reason for GUI success and value is that users can operate a mouse to point, click, and produce a desired result. The user does not need to understand all the operations performed by the application. In brief, use of DMIs result in reduced memorization, association of process results with labels (graphics or simple text), and reduced need for understanding the underlying events or tasks completing processes.

DMIs reduce impact on operators. As a result, utilizing DMIs in applications has increased the usefulness of special case users. Special case uses include people with the following traits or skills (Mayhew, 1992):

- Negative attitudes or low motivation
- Low typing skills
- Moderate system experience
- Moderate to high task experience
- Moderate application experience
- High frequency of use of other systems
- Low computer literacy
- Low frequency of use
- Moderate training

- Discretionary use
- Often away from the technology or prefer minimal interaction.

Special case uses include organizations with the following issues (Mayhew, 1992):

- High turnover rate
- Low task importance
- Low task structure (Mayhew, 1992)

In brief, DMIs are especially useful in certain parts of the workforce or population that, generically speaking, need specialized training to work more intimately with a computer system or application.

One type of DMI feature which can be self-explanatory and efficient is the icon. This is a feature Bailey (2001) mentioned, as discussed previously. Icons are both self-explanatory and efficient. Icons are symbols which represent applications or actions; and when clicked they trigger routines to open applications or perform pre-designated functions. When designs and uses for icons are considered, Mayhew (1992, p. 293) says, “Designers need to understand that employing icons ... does not necessarily in and of itself make a usable interface. Care must be taken to build in both semantic and articulatory directness in order to fully achieve the potential benefits of direct manipulation interfaces. This is an extremely important point.” She states that recognizable icons are difficult to design; and depending on the use, icons can take more screen *real estate* than words (Mayhew, 1992). In brief, icons are symbols which when clicked, trigger processes; their use is evident when properly symbolized.

Coinciding with icons and user expectation is the perceived ease or difficulty of use and how these perceptions affect performance. Mayhew (2001, p. 310) says, “An

experiment suggested that when *cognitive* complexity [mental exertion] of the task is considerably reduced, the overall task time is also reduced.” This description agrees with Bailey’s comments on usability heuristics. When tasks are thought to be difficult, people perform them that way.

As people make things more difficult for themselves, they might increase the frequency of their mistakes. Regardless of the mindset when mistakes are made, DMIs are meant to have direct, flexible, and easily reversible actions (1992). In short, DMIs are built with the operator in mind; operators make mistakes and the intention is to help them easily recover.

In summary, there are many positive aspects of DMIs as Mayhew (1992) has claimed. Mayhew discusses direct manipulation interfaces in detail. She has commented that DMIs have revolutionized computer applications because users did not need to be highly proficient to obtain desired results. She also discusses circumstances where DMIs are of exceptional value because they essentially guarantee some form of production from employees who might not otherwise contribute. Mayhew covers the pros and cons of using icons as DMIs. Finally, Mayhew highlights an experiment which suggests users do tasks quicker when tasks are perceived as easy and take longer to do a similar task when these are perceived as difficult.

Galitz (1997) describes the meaningful ordering of elements (such as icons) as a way to help items symbolize their use. Interfaces need to appear understandable. The author states the need to “provide an ordering of elements that... is logical and sequential... is rhythmic, guiding a person’s eye through the display... encourages natural movement sequences... and minimizes cursor and eye movement distances” (Galitz,

1997, p. 120). He summarizes the concept with the statement, “The arrangement of screen items should appear logical and sensible to the user” (Galitz, 1997, p. 120). Part of understanding is to have things set up so that they make sense.

A benefit from Galitz’s comments is the significant role location plays in providing context for object meaning, purpose, and use. Location can be used as a clue or hint towards functionality. Referencing the door example used by Norman (2002), if a handle were placed on the hinge side of the door, its utility would be questioned. In addition, if a handle were long, horizontal, and holding a towel draped over it, its purpose would also be questioned. In software applications involving user input (a click of a button) to move from page to page, setting a button named “next page >” in the middle of the text would not make sense. It would be better to place such a button at the end of the required reading. Likewise, naming the button with a title other than something like “next page” would confuse users. From these examples, the importance of two major points Galitz makes is evident: (1) interface objects need to make sense, and (2) interface objects need to be placed so that they make sense.

This review of literature covered usability, tabular formatting, and sorting. Two major points were made. The first was the importance of sorting capabilities for software applications with tabular information listings. The second was that operators need computer interfaces designed such that they are easily understandable, easy to figure out, and simple to use.

CHAPTER 3

METHOD

RESEARCH DESIGN

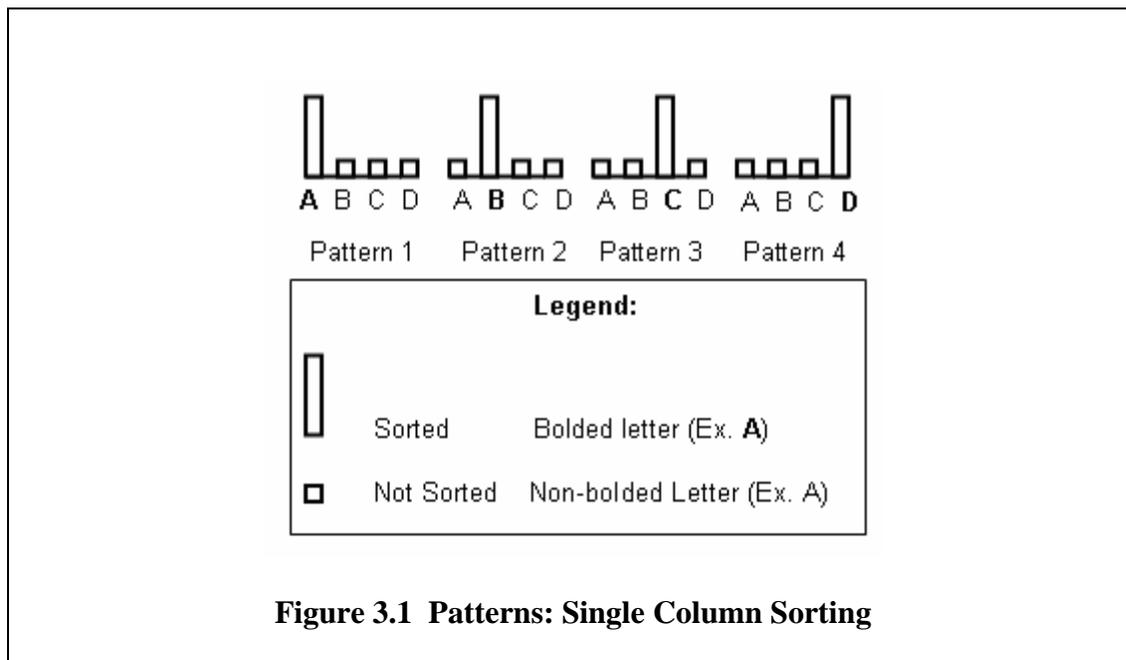
This study primarily followed a qualitative research model. The qualitative research design primarily emphasizes the interpretation of each user's perspective. This model provided a framework from which an analysis could be made of users' comparisons of methods between *prioritized selection of database arrays issued to columnar displays for the purpose of intact transposition of data* (PSDA) and a contemporary, frequently used single-sort method. The objective for contrasting the two sorting methods was to support or contradict the following hypothesis when using PSDA, (1) users with varied levels of experience are more easily able to facilitate a sort; (2) users are able to reduce line-by-line search time for specific records of information; (3) users have increased accuracy for selecting or choosing from desired records; and (4) users believe they have a more pleasant and desirable experience with sorting tasks.

The use of a standard sorting method in this research provided a frame of reference or baseline for assessing the effects of a PSDA method. This baseline has set a standard from which participants might determine if specific design elements of PSDA increased efficiency and facilitated successful outcomes from software navigation.

■ The Comparison

The sorting methodology used for comparison will be referred to as the standard-sort. It is a methodology for sorting that involves organizing a multi-column data listing by the contents of one of the columns. The reason for the ‘standard-sort’ reference is that it was and is commonly used in numerous applications implementing record sorting.

The essence of the standard-sort is that rows of data are listed such that the contents of one of the columns appears alphabetically, numerically, alphanumerically, or chronologically. In cases where rows share the same data within the column that is being sorted, those rows remain unsorted and appear unsystematically (refer to Figure 3.1). As



an example, consider a telephone directory sorted by only the last name column. If a series of records have the same last name of *Smith*, then the first names might appear

unsorted as David, then Amber, John, Bill, Ted, Zac, Paul, Jane, and Joe. Without a pattern (such as alphabetization), users would need to read down the records through all names to reach a desired one. A similar disorganized state would be true of the telephone and address columns. The standard-sort organizes a multi-column display by the contents of one and only one column.

To yield a more effectively organized list of records, PSDA was created. PSDA provides a sorting hierarchy in which all columns are sorted in the event that identical data between rows are shared across multiple columns (refer to Figure 3.2). Returning to

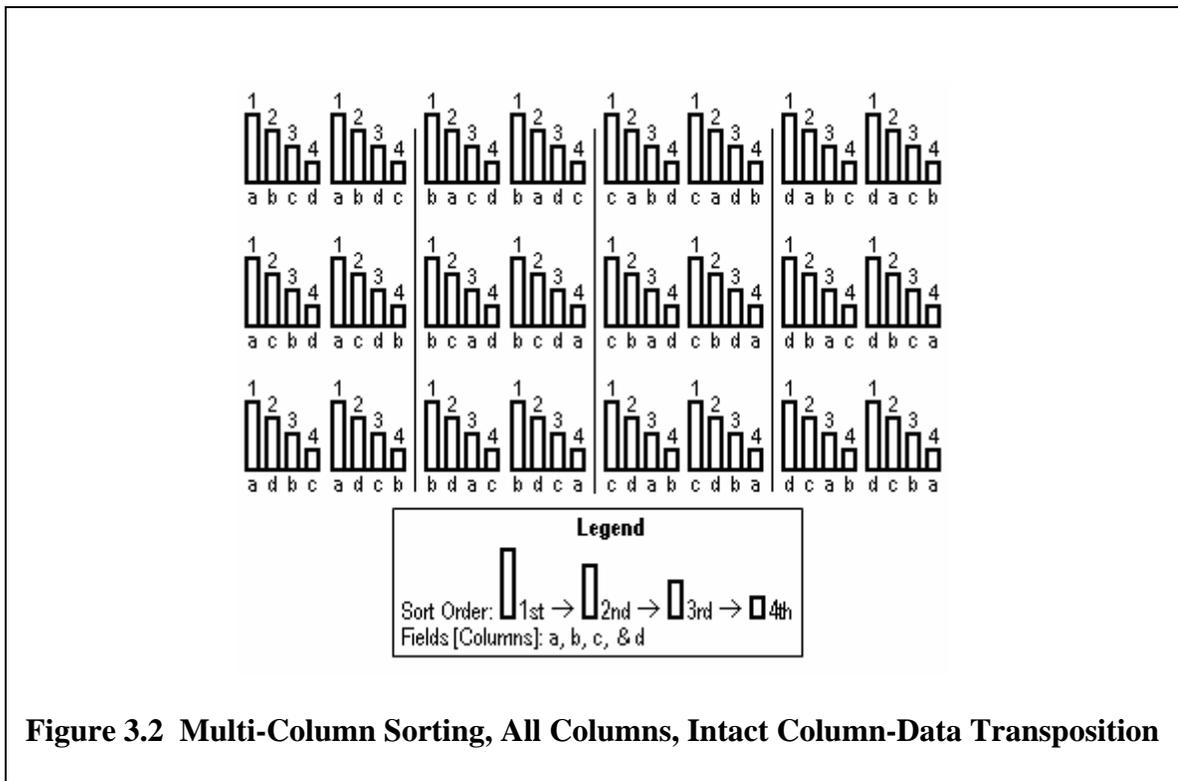


Figure 3.2 Multi-Column Sorting, All Columns, Intact Column-Data Transposition

the telephone directory example, the last name column (sorted first) is Smith and the first name column (next in sorting priority for this example) contents appear in order as

Amber, then Bill, David, Jane, Joe, John, Paul, Ted, and Zac. PSDA offers a more completely organized method for sorting than the standard-sort.

PSDA dictates that as part of the prioritization, fields appear from left to right, each in a column, according to their sort order. Figure 3.2 shows the ordering integrity maintained by PSDA while fields (noted by the letters) are transposed from column to column. Operators choose the order of prioritization to assure the arrangement meets their criteria. PSDA allows operators to visually transfer sorting priority to a different field of data as this is displayed in a different column. For usability purposes, left-to-right sorting prioritization optimizes users' eyes to flow effortlessly from column to column to pinpoint specific records. Each subsequent column is the next level of prioritization and thus further defines and refines the sort.

In contrast, the standard-sort methodology maintains categories to columns. That is, for standard-sort each field, category or definition of data for a column perpetually remains within the same column. A result is that a user must cognitively retain which column has been selected for sorting. For visual scanning and searching purposes, the sorted column is the initial area of interest or reference. The standard-sort preserves column and category integrity.

PSDA dictates that as part of the prioritization, fields appear from left to right, each in a column, according to their sort order. Figure 3.2 shows the ordering integrity maintained by PSDA while fields (noted by the letters) are transposed from column to column. Operators consciously select the order of prioritization to assure the arrangement meets their criteria. PSDA allows operators to visually transfer sorting priority to a different field of data as this is displayed in a different column. For usability

purposes, left-to-right sorting prioritization allows users' eyes to flow effortlessly from column to column to pinpoint specific records; each subsequent column is the next level of prioritization and thus further defines and refines the sort.

In contrast, the standard-sort methodology maintains categories to columns. That is, for standard-sort each field, category or definition of data for a column perpetually remains within the same column. A result is that a user must consciously consider which column has been selected for sorting. For visual scanning and searching purposes, the sorted column is the initial area of interest or reference. The standard-sort preserves column and category integrity.

A commercial application making use of the standard-sort methodology is Microsoft Corporation's spreadsheet program, "Excel." The short telephone directory example with only one last name as found in Figure 3.3 was sorted by the name column from a disorganized state. To sort, a two-step procedure was used. The user first clicked the mouse cursor in a field within the "Name" column. The user then clicked on the toolbar button denoted by the letters "A," "Z," and an arrow pointing down. Although the procedure appears simple, multiple entries with the first name "John" associate to telephone numbers and addresses which appear unsystematically. A sort by last name, first name, then telephone number, is produced by either of two approaches. One avenue is to sort the telephone number then the first name. Note in Figure 3.4 that the first names are sorted and where the first names repeat, the telephone numbers appear organized numerically. Each of these sorts requires the two-step procedure. The other possibility is to use a function accessed via the menu system. Either method requires memorization for proper usage.

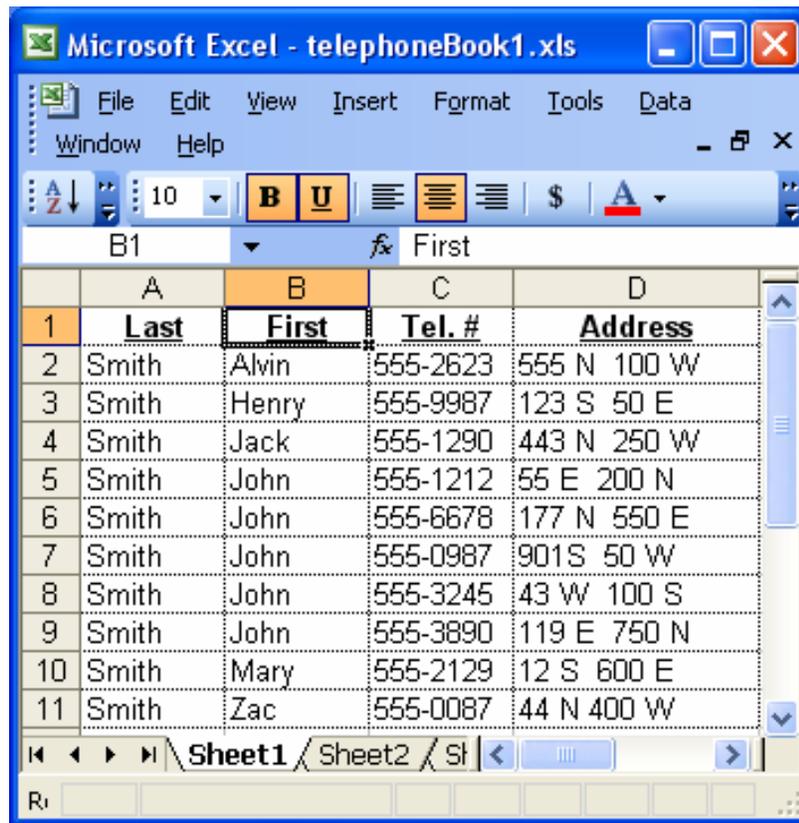


Figure 3.3 Standard/Single Column Sorting Methodology

Microsoft ® Excel / 1

In contrast, PSDA requires three obvious controls per column for its sorting as illustrated in Figure 3.5. These are push-buttons. The left and right arrows indicate the direction the column will move (transpose) when the button is pressed; the button with the up or down arrow in between the left and right arrows denotes the sort direction (ascending or descending). Figures 3.5 and 3.6 show the cues and the result from clicking the right arrow above the “Name” column or the left arrow above the “Subject” column. As illustrated, the difference between PSDA and the standard-sort method is

that the column “definition” and contents transpose left or right for PSDA while remaining the same for the standard-sort; in each case, sorting is effected. Although there are three buttons, each has symbols indicating current use; in each case, a one-step process is used to affect change.

The screenshot shows a Microsoft Excel window titled "Microsoft Excel - telephoneBook1.xls". The spreadsheet contains the following data:

	A	B	C	D
1	Last	First	Tel.#	Address
2	Smith	Alvin	555-2623	555 N 100 W
3	Smith	Henry	555-9987	123 S 50 E
4	Smith	Jack	555-1290	443 N 250 W
5	Smith	John	555-0987	901 S 50 W
6	Smith	John	555-1212	55 E 200 N
7	Smith	John	555-3245	43 W 100 S
8	Smith	John	555-3890	119 E 750 N
9	Smith	John	555-6678	177 N 550 E
10	Smith	Mary	555-2129	12 S 600 E
11	Smith	Zac	555-0087	44 N 400 W

Figure 3.4 Standard/Single Column Sorting Methodology

Microsoft © Excel / 2

 Name	 Subject	 Date Time
<input type="checkbox"/> Bob Beuford	US History Buffs sought	2003 Oct 17 2:04 pm
<input type="checkbox"/> Nathan Breker	Get great results on LSAT	2003 Oct 17 11:44 pm
<input type="checkbox"/> Nathan Breker	where am I going wrong?	2003 Oct 17 5:32 pm

Figure 3.5 Prioritized Multi-Column Sorting Example 1

PSDA provides the procedure to sort listings by each column. Figure 3.2 illustrates the transposition of columns where the highest sorting priority remains on the left-most column; subsequent columns to the right represent decreasing levels of sorting priority. Sometimes this is referred to as “trickle-down logic.” Through this methodology, a logical flow of data is sorted from the left column through the right columns. As identical field contents repeat from row to row within the same column, sorting becomes the responsibility of the next column located to the right which has the next level of order prioritization. In the example referenced by Figure 3.5, the name “Nathan Breker appears twice after “Bob Beuford.” For the two “Nathan Breker” records to be sorted using the PSDA design, the contents of the column to the immediate right need to be sorted. The benefit of PSDA is that sorting is set up to occur on all columns.

 Subject	 Name	 Date Time
<input type="checkbox"/> Get great results on LSAT	Nathan Breker	2003 Oct 17 11:44 pm
<input type="checkbox"/> US History Buffs sought	Bob Beuford	2003 Oct 17 2:04 pm
<input type="checkbox"/> where am I going wrong?	Nathan Breker	2003 Oct 17 5:32 pm

Figure 3.6 Prioritized Multi-Column Sorting Example 2

When the identity, definition, and content of the first two columns are transposed, as found in Figure 3.6, the subject category (in the first column) is sorted first followed by the name category and then the date category if necessary. Since the contents for the various rows of the left-most column displayed under the “Subject” heading are not identical, no additional sorting is required. However and most importantly, the capability for additionally sorting is inherently available should the need arise.

In summary, PSDA is a “seeing” process and therefore utilizes visibility as a core issue. PSDA is a strong logic alignment process and consequently leads the user in a DMD process of selection. And finally, PSDA represents a visible clue design process of software navigation.

■ Assumptions

This study adhered to the following five assumptions regarding tabular displays:

Assumption 1: Tabular displays of data do not provide completely unique data for each field (column) from record to record. Example 1: A telephone directory displays user accounts (the telephone number) with names and addresses as assigned to or

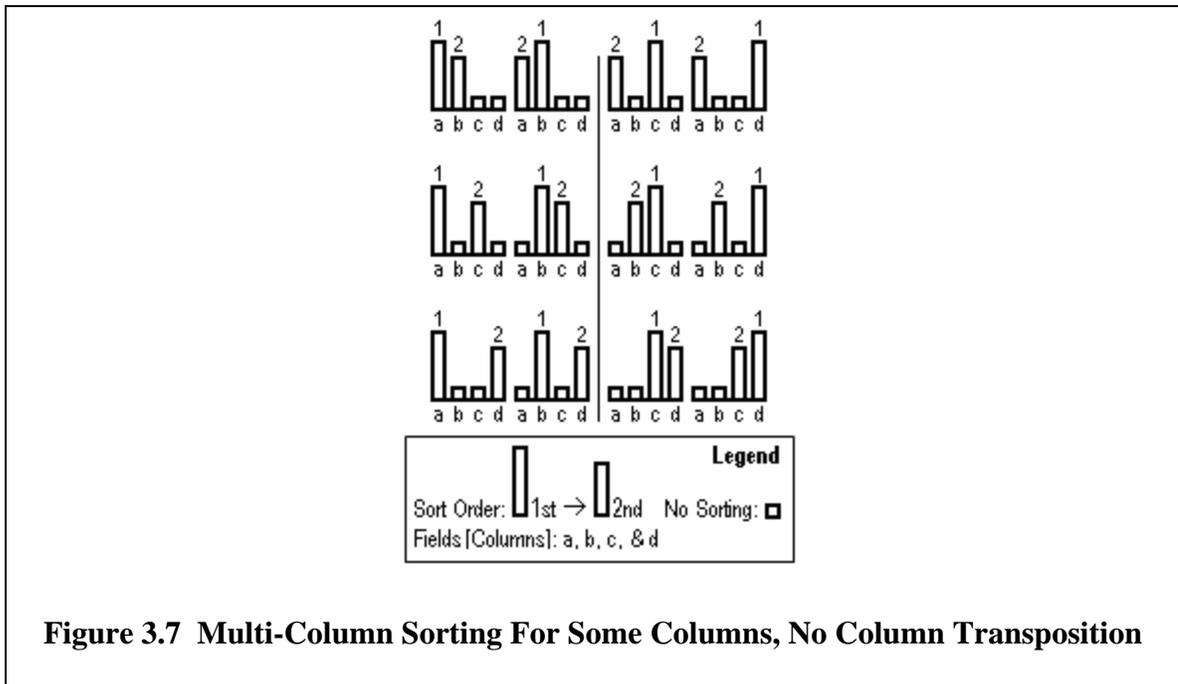
provided by the customer; multiple customers may share last names, first names and/or addresses (multiple lines in one home are one consideration). Example 2: A generic email account displays the sender's email address, subject line, and email arrival date/time. The same sender might send more than one email to a single recipient; the same sender might also send multiple emails containing the same subject line. In essence, data within the same column across numerous records may be identical.

Assumption 2: Sorting, as a software organizing tool, provides users a format for isolating data into groups. Grouping data reduces the need for users to scan all records in a display to locate a specific record or group of records. A user needs only to logically apply selection criteria to the first group, then to a subgroup of that group, and so on until the desired record is located. The key then is that sorting is an organization tool that has the capacity to isolate information.

Assumption 3: In a tabular environment where sorting occurs on only one column and identical data appear within the sorted column, the order of appearance for data in subsequent columns remains unsorted. Note Figure 3.1.

Assumption 4: In a tabular environment where sorting can occur on multiple columns, sorting will only occur on those columns selected for sorting. For columns included in sort ordering, records with identical data in the same column(s) will appear ordered; for columns not included in sort ordering, records with identical data in the same column(s) will not appear ordered. In essence, where additional sorting could be useful in columns not selected for sorting, the contents will appear unsorted and unsystematic. Figure 3.7 is representative of a case where sorting occurs on two of four columns. The identity of each column is fixed to one field. The column represented by the numerical

‘1’ represents the column which received the highest level of prioritization. In cases where data is repeated in the priority one column, subsequent ordering occurs within the column labeled with the number two. In cases, where records shared identical data within both of the two top priority fields or columns, no additional sorting occurred. If there are fifty records that share the same data in the two top priority fields, then these fifty records will display unsystematically; attempts to locate data might result in some confusion. Again, in a tabular environment where sorting occurs on multiple columns, sorting will only occur on those columns which have been selected for sorting.



Menu accessed sorting has been described previously. In short, a user must remember and select a sequence of menu topics to reach a dialog box where the

functionality is enabled. Sorting and order priority is then specified. The main disadvantage from using this method is recalling how to access the function utilizing it.

Sorting with a trail is when the last column chosen has sorting priority and previous selections retain sorting priority in the reverse order they were selected. No column transposition occurs. In a display with four columns where each had been selected, the last one selected would have sorting priority, the one selected just prior to it would have the next level of sorting priority, the one chosen prior to that one would have the third level of priority, and the first one selected would have the last level of sorting priority. Confusion arises from recalling the order the columns were selected.

The purpose of PSDA is to remove complexity from the user experience. There is no menu accessing requirement. There is no need to recall what order the columns were selected.

■ Population and Sampling

PSDA addresses sorting needs for a cross-section of the computer using populace. It is intended for users of any skill level so that application creators need only to create one listing format. As a result, target applications incorporating PSDA as a sort feature are those with diversified users possessing varying degrees of computer proficiencies.

For this research, a group representative of the intended population was considered. That is, users of varying computer proficiency levels. Consequently, the field of participation was open to anyone at any computer skill level. There was one requirement for participation, however. Each individual needed to have access to a computer with a connection to the internet. Participation amounted to interacting with a

web-based system developed and administered for this research. A description of this testing will be discussed next.

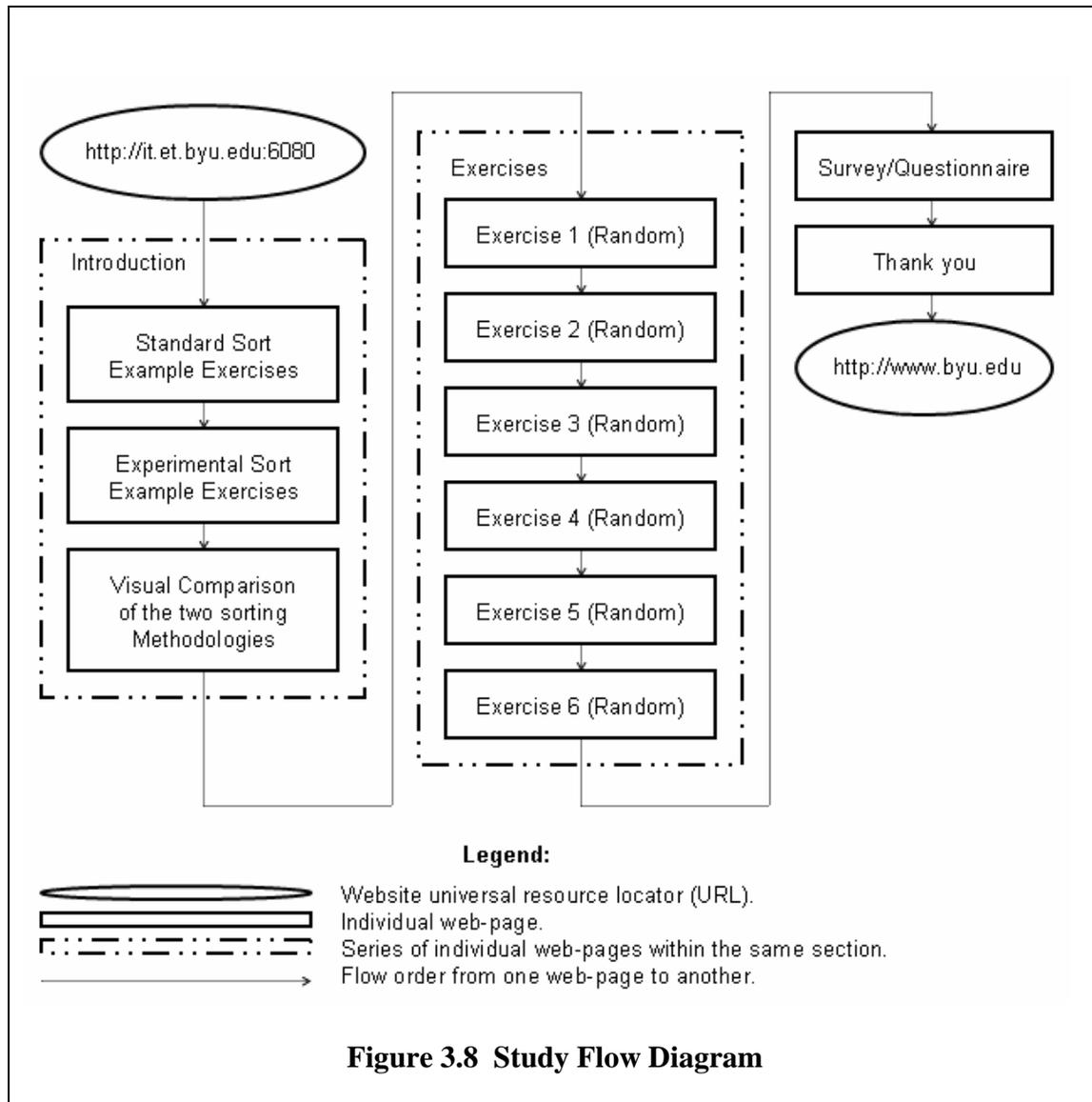
Two phases of testing were conducted. The first phase was a pilot study to evaluate the suitability of the task users were asked to perform. Data are not reported for the first phase. The second phase was the actual experiment where responses were recorded and analyzed.

Respondents were individuals with whom researchers had academic, professional or personal contact. There were eighty-one participants who had varying backgrounds and interests as well as different proficiencies of computer understanding and knowledge. Occupations consisted of librarians, computer engineers, surveyors, farmers, homemakers, lawyers, language interpreters, and business owners. Some were full or part-time college students. Their ages ranged from mid-teens to more than fifty years. The time of day each participant completed the task varied unsystematically. The only thing these individuals had in common was self-selection. Of approximately two-hundred people provided the opportunity to participate, eighty-one completed the task.

■ **The Tool for Comparison (A Brief Overview)**

The study was web-based and was made available via the internet on a university hosted server. The study included a brief introduction, a short tutorial, the research exercises, and a survey questionnaire. Figure 3.8 is a flow diagram showing the sequence of events for the survey. The introduction discussed the purpose of the study and the design heuristics within. A short tutorial discussed how the standard-sort and PSDA worked. Usability became a design component of both the introduction and tutorial. The exercises gave participants an opportunity to briefly experience each of the sorting methods. A

survey recorded user responses to a number of questions designed to solicit information about each participant as well as the tasks. Each part of the study was sequentially designed to introduce the next sections which then culminated in the questionnaire. Each component of the web-based research task and survey tool is discussed.



■ Administrative Procedures

A primary concern with collecting results was to assure consistent and reliable measures and reduce the likelihood for introducing bias into the research. As a result, participants received instruction and information about the research project through the web-based survey. Outside of the information provided within the web-survey, the only information supplied to potential participants was the address of the web-survey, the approximate time to complete the survey, and the purpose of the survey as research support for a Master's thesis. Any discussions about the survey or the research prior to a participant completing the survey was discouraged. The survey and its instructions will be discussed further.

■ Criterion Measures

1a) The survey program measured how much time it took a person to complete different parts of the survey. Results from participants as a whole would have to be significantly quicker for PSDA than the standard-sort.

1b) A survey question requested feedback about a participant's general background with computers. The purpose of this was to help determine whether the usefulness of PSDA was affected by a participant's prior understanding of computer systems.

1c) Participants' feedback would help establish whether they felt sorting was easier. Depending upon the scenario, if the user felt that sorting with PSDA was quicker, perhaps there would be a reason. Requesting reasons for preference might influence consideration for future testing. If a person felt they were faster but their actions did not reflect their performance, retrieving their personal feedback might not be important.

2) The survey program measured how much time it took a person to complete different searches. If search times for PSDA were shorter, it might be an indicator that PSDA reduces line-by-line searching.

3a) The survey program counted the number of errors made for each question and calculated a percentage. If the responses for PSDA showed a statistically significant higher accuracy than standard-sort, then PSDA would be more effective.

3b) Participants' personal opinions as to whether they felt they were more accurate with PSDA might be as significant as the resulting measures of accuracy discussed in 3a. Getting their feedback might reflect their confidence level when using PSDA.

■ Statistical Treatment

Participant response to each exercise question was timed and considered on a basis of accuracy using percentages. In addition, one question might have multiple corresponding responses. To get a correct answer required all true responses to have been chosen. Reliability of the answers to the exercise questions would then be tested with the “*t* Method with Same-Subject Groups.” Reliability differences were discussed in association with each question content.

Participants' response to a survey question resulted in categorical classifications. A calculation of the responses for the entire group showed frequencies for each of the response categories. When compared to the whole, these frequencies resulted in percentages. For example, one question resulted in the following responses: 2% strongly agreed, 23% agreed, 25% expressed neutrality, 45% disagreed, and 5% strongly disagreed. It would not appear that there was not a lot of agreement with the related

question or comment. Using classifications allowed for statistical treatment of participant responses.

An analysis of the reliability for the responses to each question was conducted by the Brigham Young University statistics department. As a process to strengthen reliability, responses from the strongly agreed category would be combined with the responses from the agreed category and the responses from the strongly disagreed category should be combined with the responses from the disagreed category. As a result of these changes, only three categories needed to be tested: agreed, neutral, and disagreed. Reliability of the responses to the exercise questions was then tested with the “Chi-square formula”. In summary, the feedback provided from the BYU statistics department to reduce the number of categories provided an initial way to increase the likelihood that responses to each question would be reliable; using the Chi-square formula provided a statistical means to measure the likelihood of reliability.

IMPLEMENTATION

PSDA was designed and built into the reporting interface of a software application housing thousands of records with examination scores. This creation occurred during the latter part of the year two-thousand and one. Each record contained multiple score fields. Several score related fields correlated to a subject area. Each score related field within a specific subject area corresponded to an actual score or a type of interpretation.

The reporting interface consisted of a three step procedure for selecting columns to display, filtering records, then sorting. The basis for reporting was data-entry into a separate electronic form containing approximately forty categories (columns) of data.

The first step for building the report was the selection of the categories to include in the report if these categories were necessary for viewing, filtering, or sorting. The second step was for filtering. Only categories selected in the first step could be used as a basis for filtering. As an example, if the date category was selected, the operator would be able to select all records meeting other criteria as well as the time period between and including a start and end date. The final step (step three) displayed a tabular listing of all categories (columns) selected in step one as well as all records meeting the filtered criteria chosen in step two. The initial presentation of the tabular listing automatically positioned, and therefore prioritized, the categories (columns) in the sequence of appearance noted on the data-entry electronic form. As an example, if the category *last name* was numbered one and the category *first name* was numbered two, then the last name column would be the first column on the left and the first name column would follow in the adjacent column to the right. Operators were then able to re-sort the display using PSDA by clicking on the appropriate arrows provided at the top of each column.

There were practical reasons for creating a tri-step procedure for the reporting tool within this examination score software application. These reasons included the handling of the large quantity of data categories, the accessibility of data, and the anticipated skill level of computer users. As was discussed previously, approximately forty categories of data were available for each record produced. This number was substantial and posed a cumbersome situation if one were to have to work with all of them simultaneously in a report. The environment was web-based and therefore operators would not have direct access to the database without planning and coordinating an extract of the contents. Even with direct access to data, operators had varying backgrounds and skill levels with

computers. Some operators had the ability to turn on a computer and just get to applications they had learned to use while other operators had a greater depth and knowledge about computers. When building the system, a designer could not assume that computer proficient operators would be readily available to support users. One way to work around these concerns was to create and format individual queries for every possible type of query. Given the number of columns and the different combinations for sorting records, creating individual queries was not realistic. To consider a subset of queries also did not seem sensible because it would be impossible to forecast which queries would be desired. The practical approach was to allow users to select desired categories, provide them a way to filter records, and finally provide them a visually logical way to sort the resulting list.

Since this initial implementation of PSDA, the feature has been regularly used to retrieve specific and group information for reporting purposes. The reasons for some of the reports have been to match tasks to those having skills or proficiencies in those areas. Reports have also been used to establish potential capabilities based upon the group size of those meeting certain criteria.

Observations of those using the software application and specifically the feature incorporating PSDA provided insights regarding its usefulness. Two highlights from these observations are as follows. First, the sorting feature, PSDA, required little training to show operators how to properly use. Second, some of the multiple users had extended periods of time (several months) in which they did not use the program. Rarely was there a request for retraining on how to do sorting. By viewing the interface, operators were able to recall how to use the PSDA function for sorting.

For the purposes of this research, a web-based survey tool was created with the PSDA feature. This software-built, database-driven application was hosted on a web-server located with the Information Technology Program in the College of Engineering and Technology at Brigham Young University and was accessible to participants who completed the survey over the internet.

Several levels of coordination were necessary to properly host the application. Permission to host the web-application on a department server was obtained through an IT faculty member. An administrator of the department's servers provided access to a web server; the administrator also assured that necessary applications were also available. As an example, the survey application was created using Microsoft (MS) Corporation's Active Server Pages scripting language; this required the MS Internet Information Server (IIS) as well as the MS Access database to store content and results. Additionally, working from off campus required the ability to use programs such as the Remote Desktop Connection to manipulate or transfer files on the web server. The server administrator assured that these applications were available. Coordination with faculty and system's administrators resulted in successful hosting of the application.

The study was composed of several divisions as found in Figure 3.8. It included a brief introduction, a short tutorial, exercises, and then a survey. The introduction provided a familiarization to the sorting methods used within the program. The exercises allowed participants to experience the difference between the two sorting methods. The survey questions gleaned a few facts about the participants as well as their perceptions and views relating to each of the sorting methods. A more comprehensive discussion of these divisions follows.

The introductory page was divided into several sections. The first section explained expected time commitments for the different portions of the questionnaire. The second section explained the general purpose for the study. The third section described the methodology for the Standard Sort and provided a quick tutorial and experiment. Likewise, the fourth section described the methodology for the Prioritized Left-To-Right Multi-Column Sort and provided a quick tutorial and experiment for that method.

The first section appeared as found in Figure 3.9. From this portion, participants were reminded or made aware of the anticipated time commitment.

This web-based research has the following small sections:			
Section	Topic	Approximate	Time
		Commitment	(minutes)
1)	Introduction with an explanation of the two types of sorting. (This section)	5	
2)	Quick exercises on both versions of the sorting.	12	
3)	A short questionnaire	4	
4)	Quick thank you for your participation.	1	
TOTAL		22	

Figure 3.9 Introduction as Found on Website

INTRODUCTION

First of all, thank you for participating in this research project.

The general purpose of this research is to compare a computer user's response, feelings, and concerns to two different column-sorting methods.

Examples of sorting occur when alphabetizing email lists. In an attempt to find a specific received email, a user might recall who sent it and then click on the "Name" column to sort alphabetically by the sender's "Name". On the other hand, a user might recall what day it was sent and then click on the "Date" column to sort alphabetically by the "Date" the email was received. Sorting on the Subject or other columns (depending on the email program) might allow sorted listings based upon the respective column.

The specific purpose of the research you are assisting is to determine the effectiveness of sorting methods that would help pin-point one email among many that has either been sent by a specific user account, and/or within a specific time frame, and/or with a similar subject line, and/or other characteristics that might be influenced by the type of sorting.

Consider the display below, which is representative of incoming emails. It represents a common method or standard for displaying email. That is, sorting is constrained to the listing of a single column. By selecting sorting on another column, alphabetization is done on the items of that selected column. Influence on alphabetization is limited to the selected column. Please click on the button above the column headers to see the changes that occur.

Try the practice examples then click continue at the bottom of the page.

Figure 3.10 Wording From Website Survey

The second section appeared as found in Figure 3.10. Participants were introduced to the sorting concepts involved in the study.

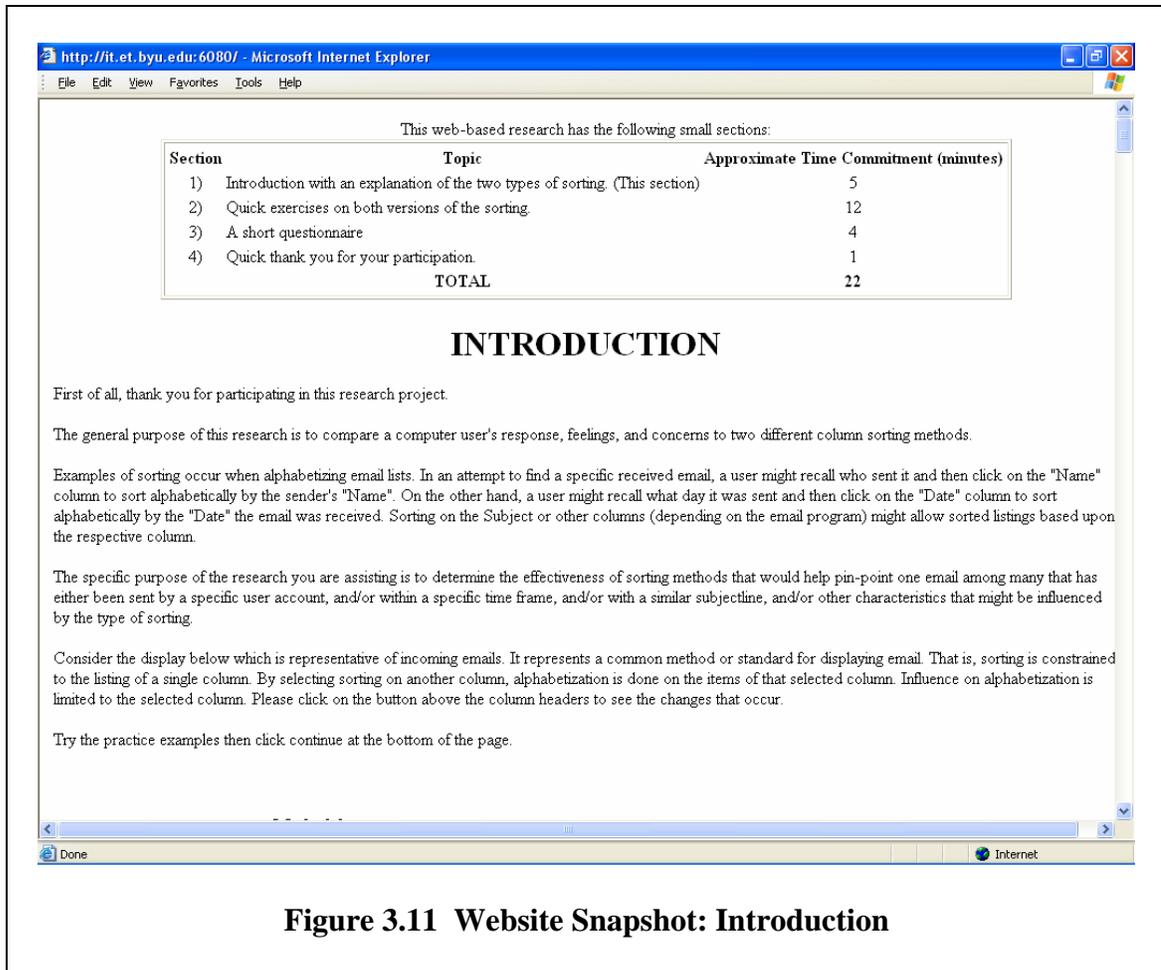


Figure 3.11 Website Snapshot: Introduction

The two first sections appeared on the web interface as found in Figure 3.11.

Methodology Standard Sort View

Note that as a column header button is selected, the column in question becomes alphabetized. Random alphabetization on other columns may occur by chance. [Clicking on the same button a second time allows reverse alphabetical listings to occur. Z to A rather than A to Z.]

	v	.	.
	Name	Subject	DateTime
<input type="checkbox"/>	Betty Benter	What kind of food do you like?	2003 Mar 22 1:02 am
<input type="checkbox"/>	Betty Benter	I know a place in Heber	2003 Mar 27 10:44 am
<input type="checkbox"/>	Betty Benter	Cool that you help with	2003 Mar 26 10:58 pm
<input type="checkbox"/>	Betty Benter	Thanks for help in dance class	2003 Mar 23 6:54 pm
<input type="checkbox"/>	Betty Benter	Prep for graduation	2003 Apr 07 11:33 am
<input type="checkbox"/>	Betty Benter	Saw you cheering at the game	2003 Mar 25 10:56 pm
<input type="checkbox"/>	Betty Benter	You seem a little sick	2003 Apr 14 5:22 pm
<input type="checkbox"/>	Betty Benter	We need to talk when you ...	2003 Apr 16 1:49 am

FOR PRACTICE

Given --
Date: '2003 Jan 01'
Subject: 'opening in: Marketing'
What **Names** correspond with this?
If you found 'Cisco Jobs', you are correct.

Given --
Name: 'YourMagazine.net'
Date: '2003 Sep 25'
What **Subjects** correspond with this?
If you found 'Your subscription ends in...!', you are correct.

Given --
Subject: 'opening in:'
Name: 'Novell Opportunity'
What **Dates** correspond with this?
If you found '2003 Jul 01', '2003 Jul 01', '2003 Jul 02', '2003 Jul 02', '2003 Jul 04', '2003 Jul 04', '2003 Jul 05', '2003 Jul 05', '2003 Jul 12', '2003 Jul 27', '2003 Aug 07', '2003 Sep 25', **and** '2003 Sep 30', you are correct.

More records exist which are not displayed here.

This portion repeated multiple times for user benefit.

Figure 3.12 Wording for Standard Sort View Methodology Explanation

The third section contained information as found in Figure 3.12. This section provided participants an awareness of sorting with which they might be familiar; or for those using it for the first time, it showed them how single column sorting generally functions. Participants were provided an opportunity to review their understanding of the single-column sorting methodology through a search based upon given criteria. These simple exercises were provided to the right of the listing and included the corresponding correct answers. The exercises were formatted to encourage participants to experiment with the single-column sorting methodology. To conserve space, a major portion of the listings was intentionally omitted from this document.

Section three appeared on the web-page as found in Figure 3.13.

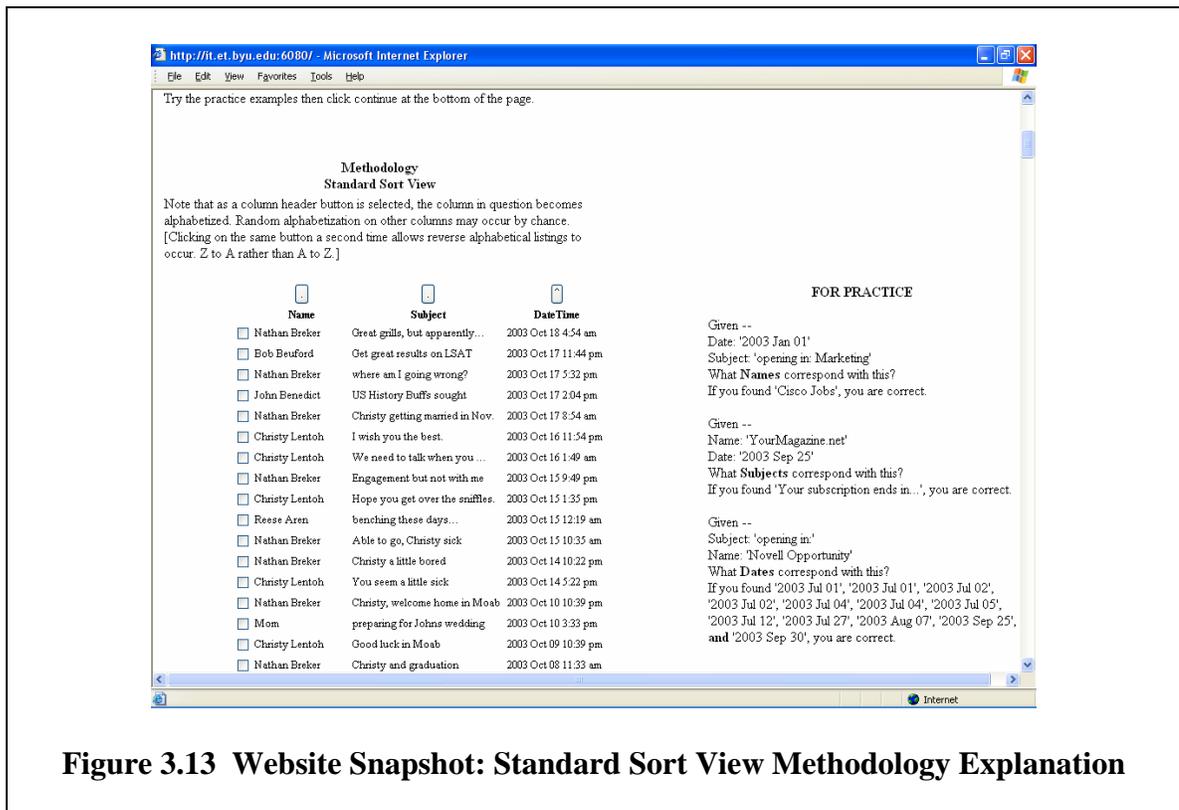


Figure 3.13 Website Snapshot: Standard Sort View Methodology Explanation

The introductory comments in section four are found in Figure 3.14. This section introduced the participants to the concept of PSDA. Users read through the steps to learn how the functionality worked. It was assumed that they would have participated by clicking on the corresponding buttons and then they would have noted the resulting changes in the sort ordering within the columns.

INTRODUCTION

Consider the display below, which uses a non-standard method for displaying a listing of emails. The records are displayed by prioritized selection (note the listing on the right half of the page) in which sorting occurs on each column with the left-most column taking top-precedence, the second column taking secondary precedence, and so on with the last column having the least amount of precedence. Please click on the column headers and note the change that occurs. (Entire columns of data are transposed, that is, the contents of two side-by-side columns are exchanged.)

Try the practice examples then click continue at the bottom of the page.

Figure 3.14 Wording for Multi-Column Sort Methodology Explanation

Figure 3.15 displayed a portion of the web-page containing the contents of Figure 3.14.

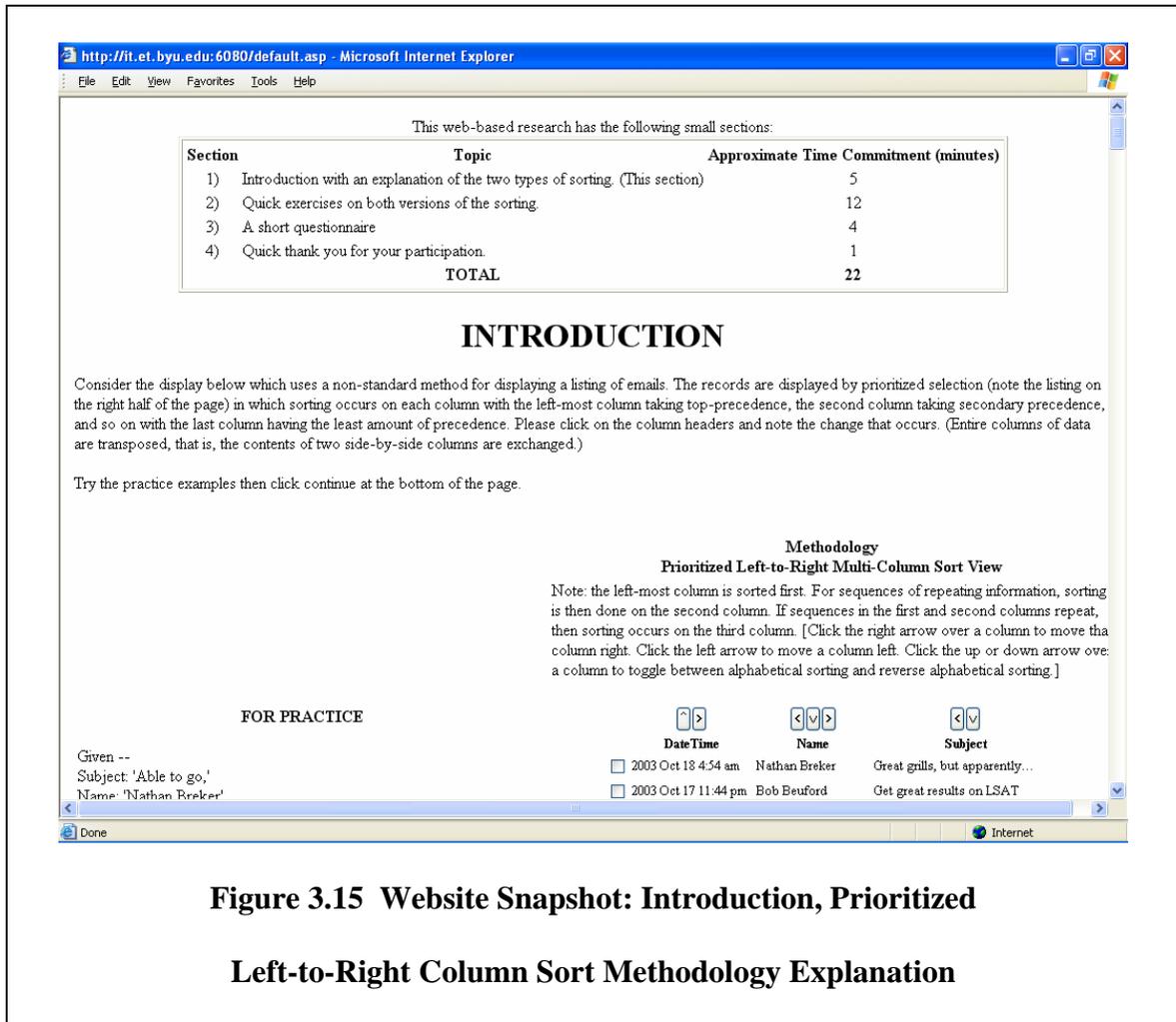


Figure 3.15 Website Snapshot: Introduction, Prioritized Left-to-Right Column Sort Methodology Explanation

Figure 3.16 showed contents for the fourth section of the introduction. Further explanation of the experimental method was provided.

Methodology
Prioritized Left-to-Right Multi-Column Sort View

Note: the left-most column is sorted first. For sequences of repeating information, sorting is then done on the second column. If sequences in the first and second columns repeat, then sorting occurs on the third column. [Click the right arrow over a column to move that column right. Click the left arrow to move a column left. Click the up or down arrow over a column to toggle between alphabetical sorting and reverse alphabetical sorting.]

FOR PRACTICE

Given --
 Subject: 'Able to go,'
 Name: 'Nathan Breker'
 What **Dates** correspond with this?
 If you found '2003 Apr 15' **and** '2003 Oct 15', you are correct.

Given --
 Name: 'Jobs R Us'
 Date: '2003 Jan 02'
 What **Subjects** correspond with this?
 If you found 'jobs: clerical typist' and 'jobs: newspaper editor', you are correct.

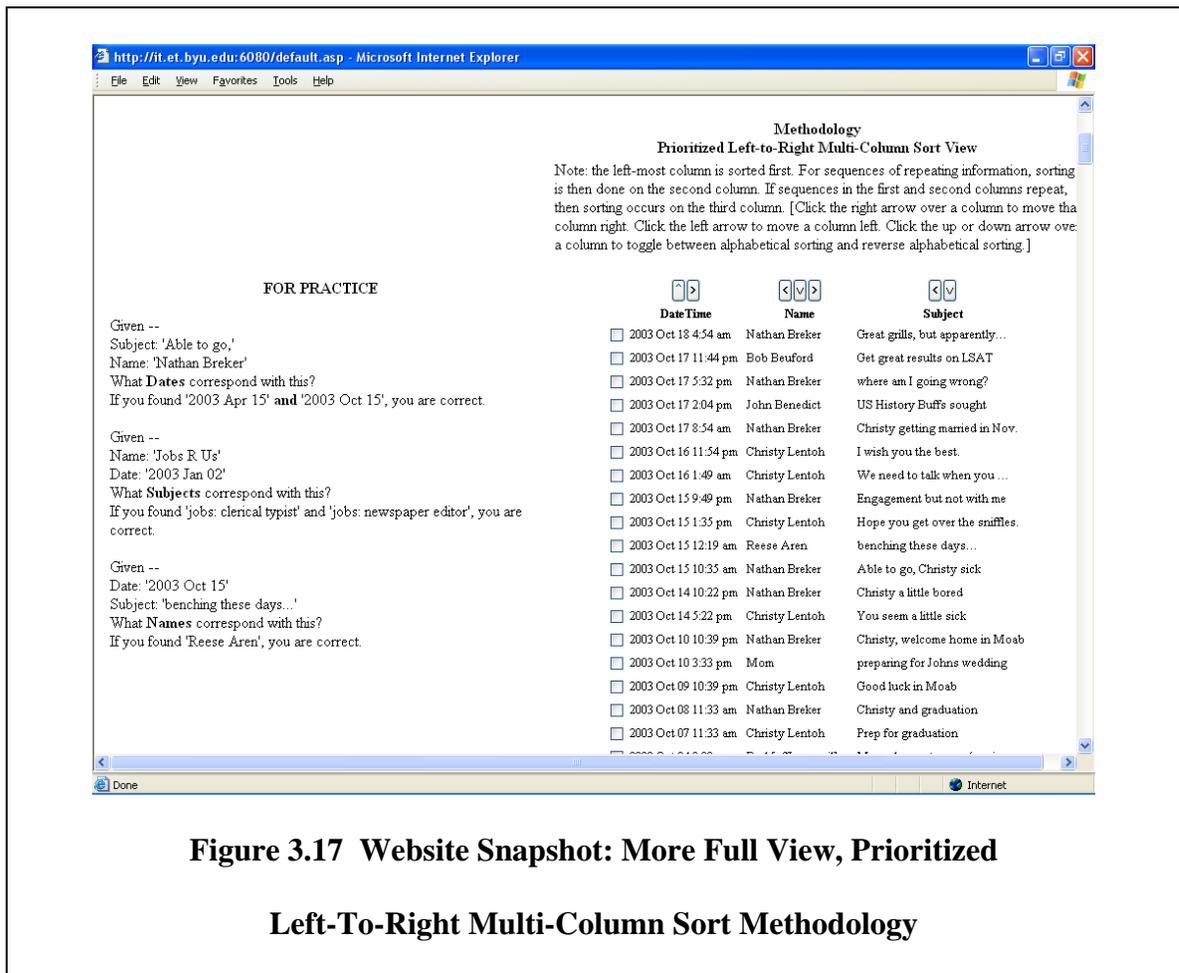
Given --
 Date: '2003 Oct 15'
 Subject: 'benching these days...'
 What **Names** correspond with this?
 If you found 'Reese Aren', you are correct.

	^ >		< v >		< v		
	Date	Time	Name	Subject			
<input type="checkbox"/>	2003 18	Oct 4:54 am	Nathan Breker	Great grills, but apparently...			
<input type="checkbox"/>	2003 17	Oct 11:44 pm	Bob Beuford	Get great results on LSAT			
<input type="checkbox"/>	2003 17	Oct 5:32 pm	Nathan Breker	where am I going wrong?			
<input type="checkbox"/>	2003 17	Oct 2:04 pm	John Benedict	US History Bufs sought			
<input type="checkbox"/>	2003 17	Oct 8:54 am	Nathan Breker	Christy getting married in Nov.			

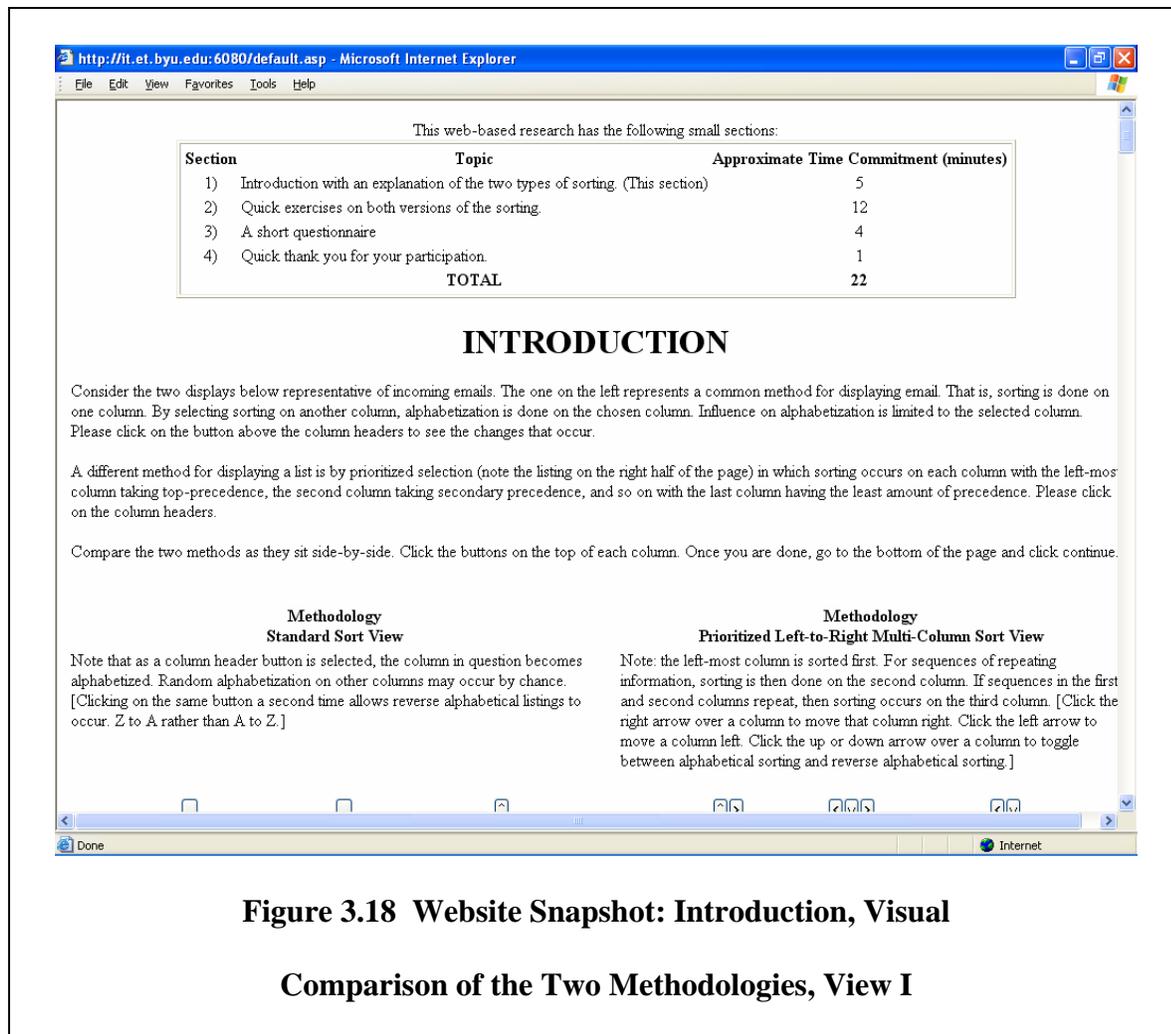
Figure 3.16 Wording for the Left-to-Right Prioritized

Multi-Column Sort Methodology

Figure 3.17 provided the image web view display depicting how the participants visualized the explanation of the experimental sorting method. As might have been noticeable, it included practice exercises with answers to allow users to test their understanding of the concept. After they searched for the answers, immediate reinforcement provided a response to either a correct or incorrect response.



The introduction then followed with a presentation of the sorting methodologies, side-by-side, to give participants a direct visual comparison of the two methods. Note Figures 3.18, 3.19, and 3.20.



**Figure 3.18 Website Snapshot: Introduction, Visual
Comparison of the Two Methodologies, View I**

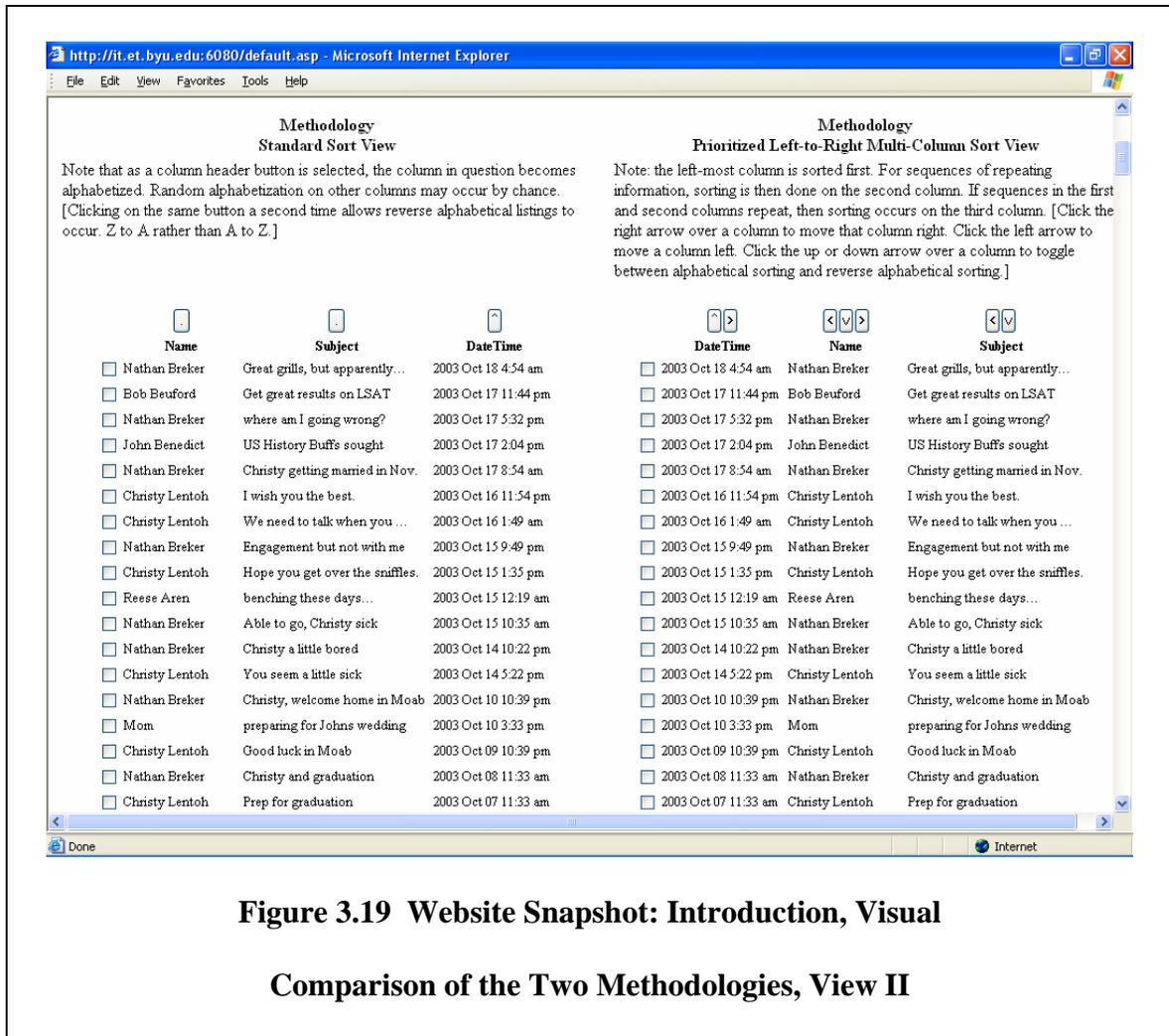


Figure 3.19 Website Snapshot: Introduction, Visual

Comparison of the Two Methodologies, View II

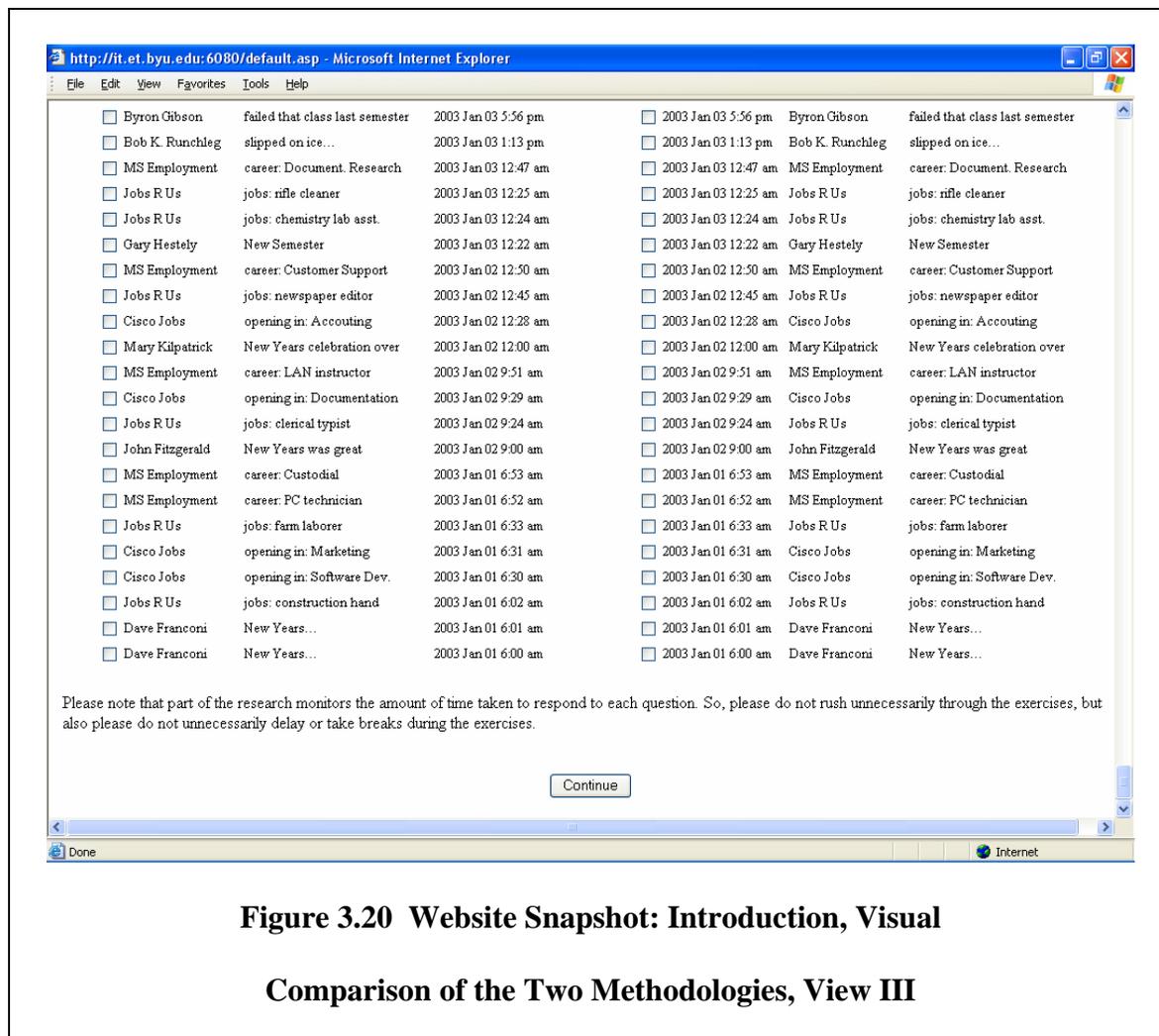


Figure 3.20 Website Snapshot: Introduction, Visual

Comparison of the Two Methodologies, View III

The introductory section was followed by six exercises. These exercises were displayed individually and, just as the practice exercises in the introduction, these exercises provided two parameters of data which matched one or more records. The plan was for participants to locate the records corresponding to the criteria and in the process simplify their search by sorting the rows. As participants determined which records matched, they were to mark the checkbox on the same row. Upon completing each

exercise, participants were to click a button at the bottom of the page to move onto the next exercise or to the survey once the exercises were completed.

The questions for each exercise were parametrically generated by the survey software application. Three exercises used a tabular display organized with the standard sort method and three exercises used a tabular display organized with PSDA. Two exercises, one per group (standard sort or PSDA), were to have the operator locate one or more records based upon a provided subject and date. Another question from each group was to have the operator locate one or more records based upon a provided name and date. The last question from each group was to have the operator locate one or more dates based upon a provided name and subject.

The application then determined the order in which the type of questions would appear. For one participant, the exercise question-types might appear as follows: (1) using PSDA, find records containing the following subject and date...; (2) using PSDA, find records containing the following name and subject...; (3) using standard sort, find records containing the following name and date...; (4) using PSDA, find records containing the following name and date...; (5) using standard sort, find records containing the following name and subject...; (6) using standard sort, find records containing the following subject and date.... A different participant might have question-type exercises ordered as illustrated in Figures 3.21 through 3.34. These are ordered as follows: (1) using standard sort, find records containing the following name and subject...; (2) using PSDA, find records containing the following subject and date...; (3) using PSDA, find records containing the following name and date...; (4) using standard sort, find records containing the following name and date...; (5) using standard sort, find

records containing the following subject and date...; (6) using PSDA, find records containing the following name and subject... Other participants might find themselves with other question-type sequence combinations.

The purpose for non-systematic selection of parametric values as well as the arbitrary sequence of the question-type appearance was to ensure that bias was not being inserted into the results and thus a way to increase statistical reliability. In a static environment, difficult questions might have been inadvertently fixed to one methodology. Although individuals might have anomalies with their questions and results, the group considered as a whole would generally have acceptable results.

One item of consistency was employed, however, for usability purposes. Whenever a question about the standard sort would appear, the 'email' listing would appear in the left column with the question appearing in the right column. Whenever a question about PSDA would appear, the 'email' listing would appear in the right column and the question would appear in the left column. This difference allowed the participants to be aware by site of which type of methodology was expected of them to use to help respond to questions. Additionally, to assist the participant, the question would appear in its respective column multiple times so that the user would not need to continually scroll to the top of the page to remember it. In short, the question was always in sight. It was a feature requested by someone who participated during earlier trials.

The exercises provided participants a live environment to experience both sorting methods. The plan was for participants to have a better understanding of each method to better answer the survey which posed questions about each method and the participants' experience.

In Figure 3.21 which illustrates the first exercise, the participant could have initially sorted on the Name column since one of the parameters provided was a Name.

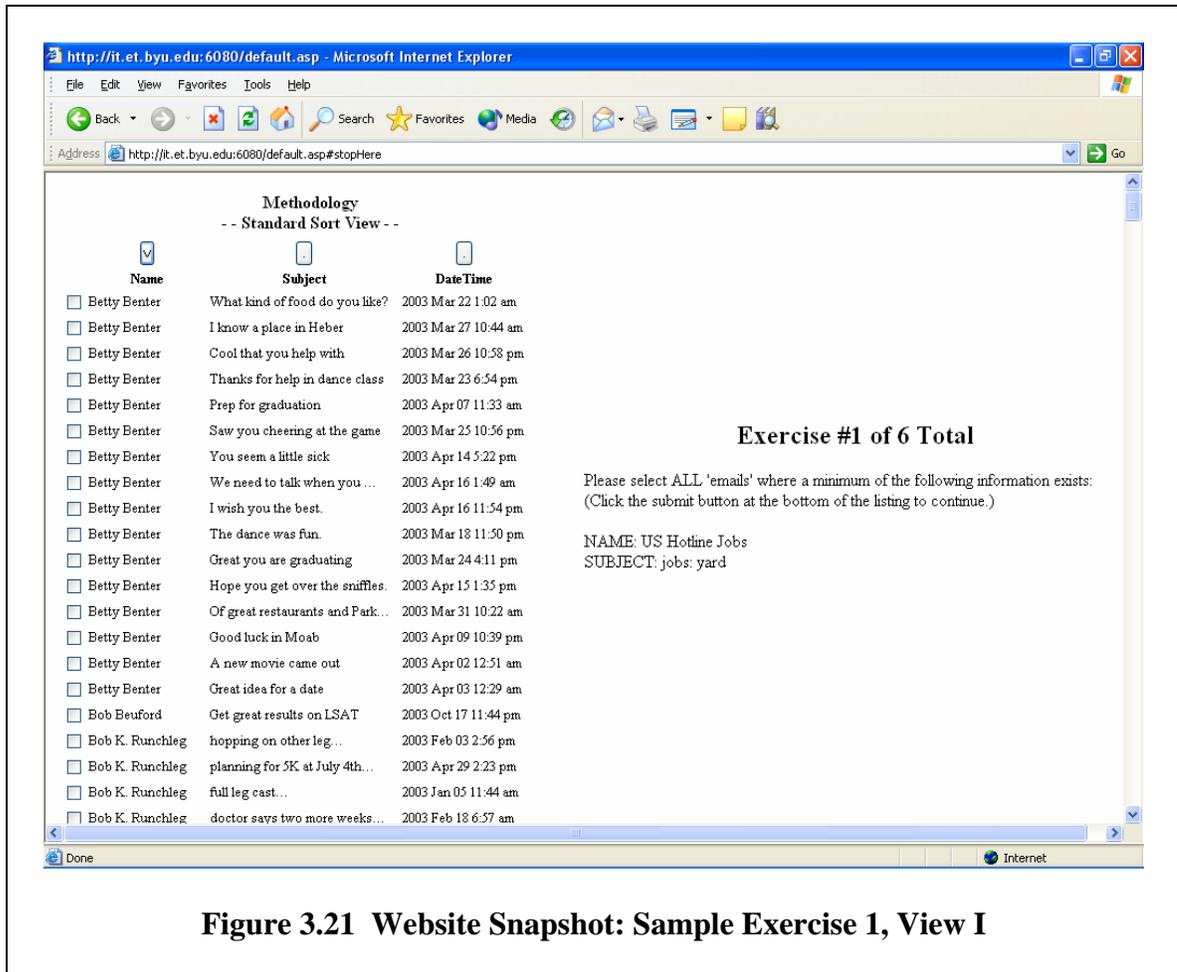


Figure 3.21 Website Snapshot: Sample Exercise 1, View I

In Figure 3.22, the illustration displays numerous records with US Hotline Jobs as the Name in the first column. Participant may have found it difficult to read through the numerous corresponding subjects which appeared unsorted. As a result, the participant may have returned to the column headers and selected a sort on the Subject column and returned a listing as displayed in Figure 3.23. There are fewer records with the Subject

‘jobs: yard’ than there are records with the Name ‘US Hotline Jobs.’ Upon finding the appropriate record, the participant would then mark the checkbox as seen in Figure 3.23 then click the submit button at the bottom of the web-page as found in Figure 3.24.

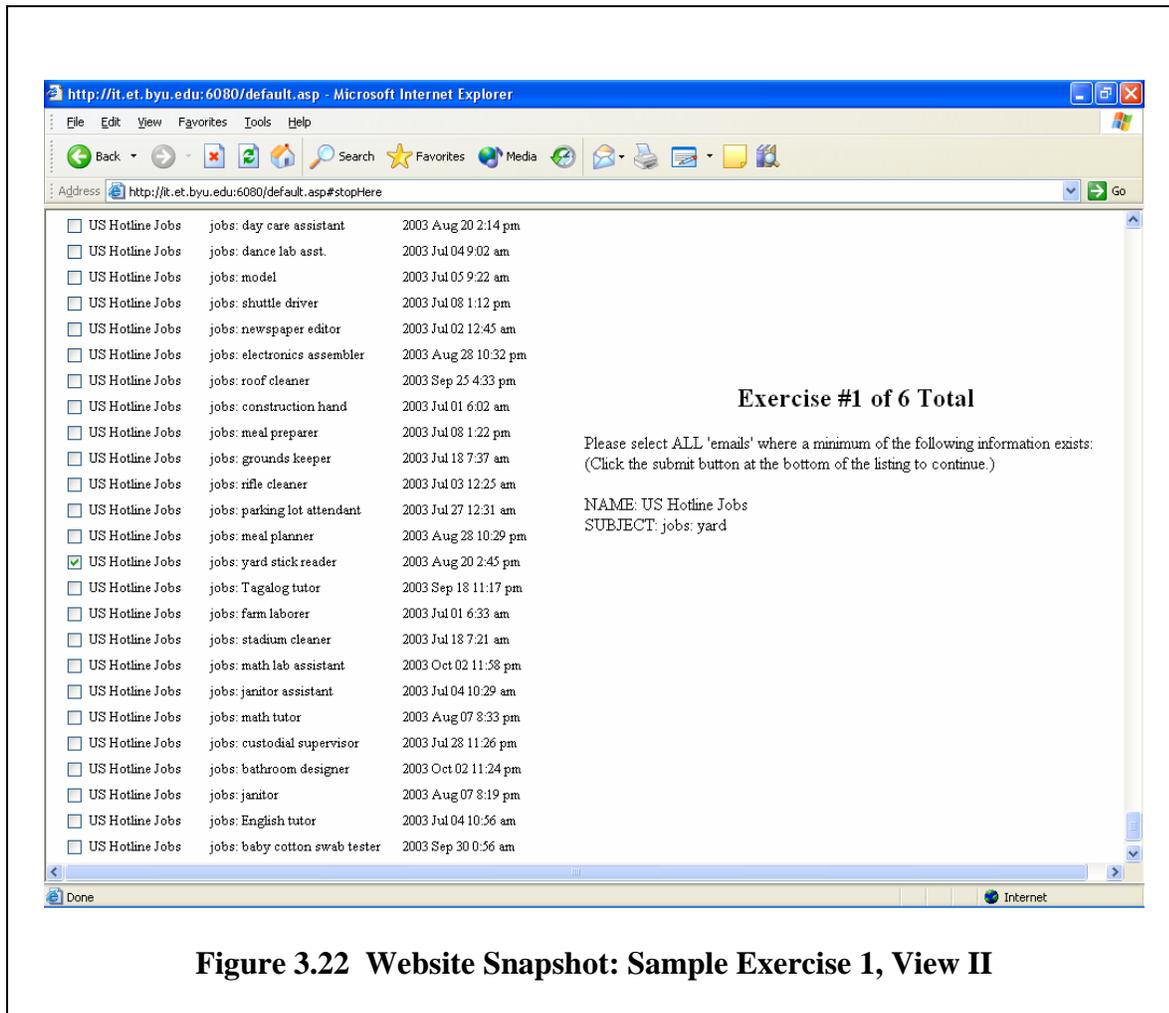


Figure 3.22 Website Snapshot: Sample Exercise 1, View II

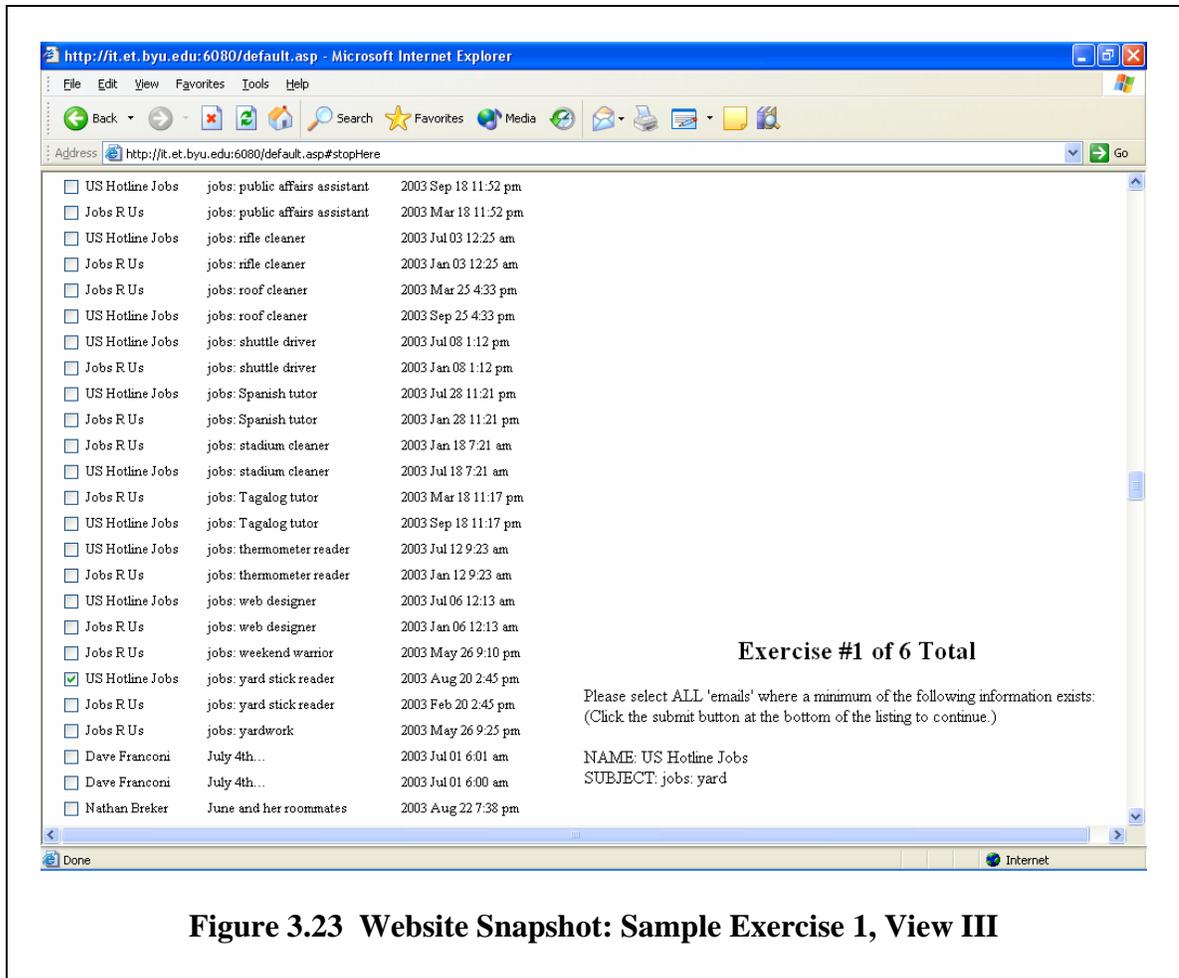


Figure 3.23 Website Snapshot: Sample Exercise 1, View III

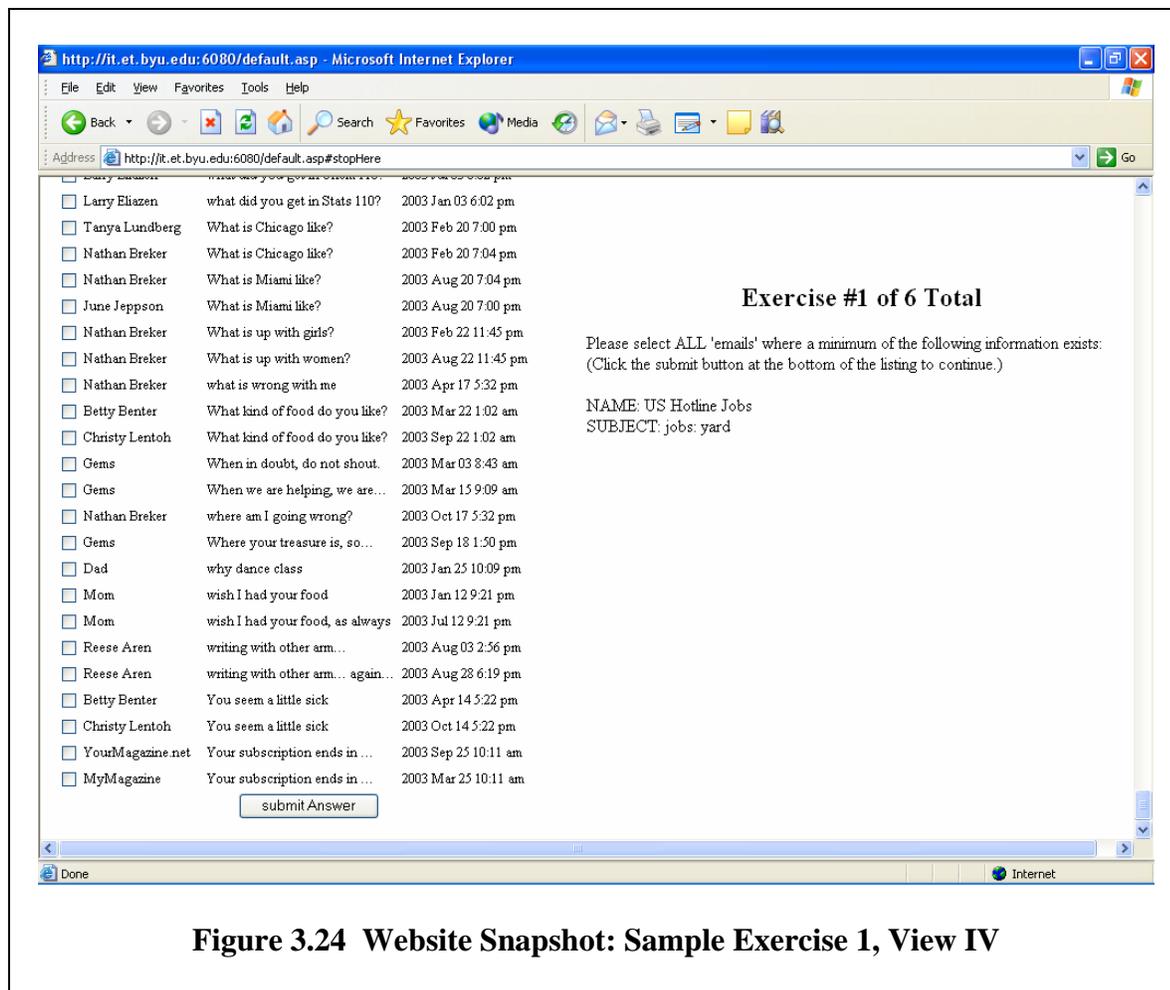


Figure 3.24 Website Snapshot: Sample Exercise 1, View IV

Figure 3.25 illustrates the second exercise. Within this exercise, the participant must find applicable records for the given Subject and Date. To facilitate the search, the participant could sort on the Subject then Date or Date then Subject. In this situation, the columns were sorted by Subject then Date. Figure 3.26 illustrates the correctly selected record. Figure 3.27 again illustrates the placement of the submit button which the user must press to move onto the next question. In the event that no records are selected, the application will not advance to the next exercise in the sequence. The application will retransmit the same data it transmitted previously for the exercise that has not yet been

completed. As a note, the survey participant is not given a mechanism to return to previous exercises and change the responses. One reason for not allowing participants to complete the exercises sequentially is that the exercises are timed and it would otherwise be difficult to determine the time it took to complete each exercise.

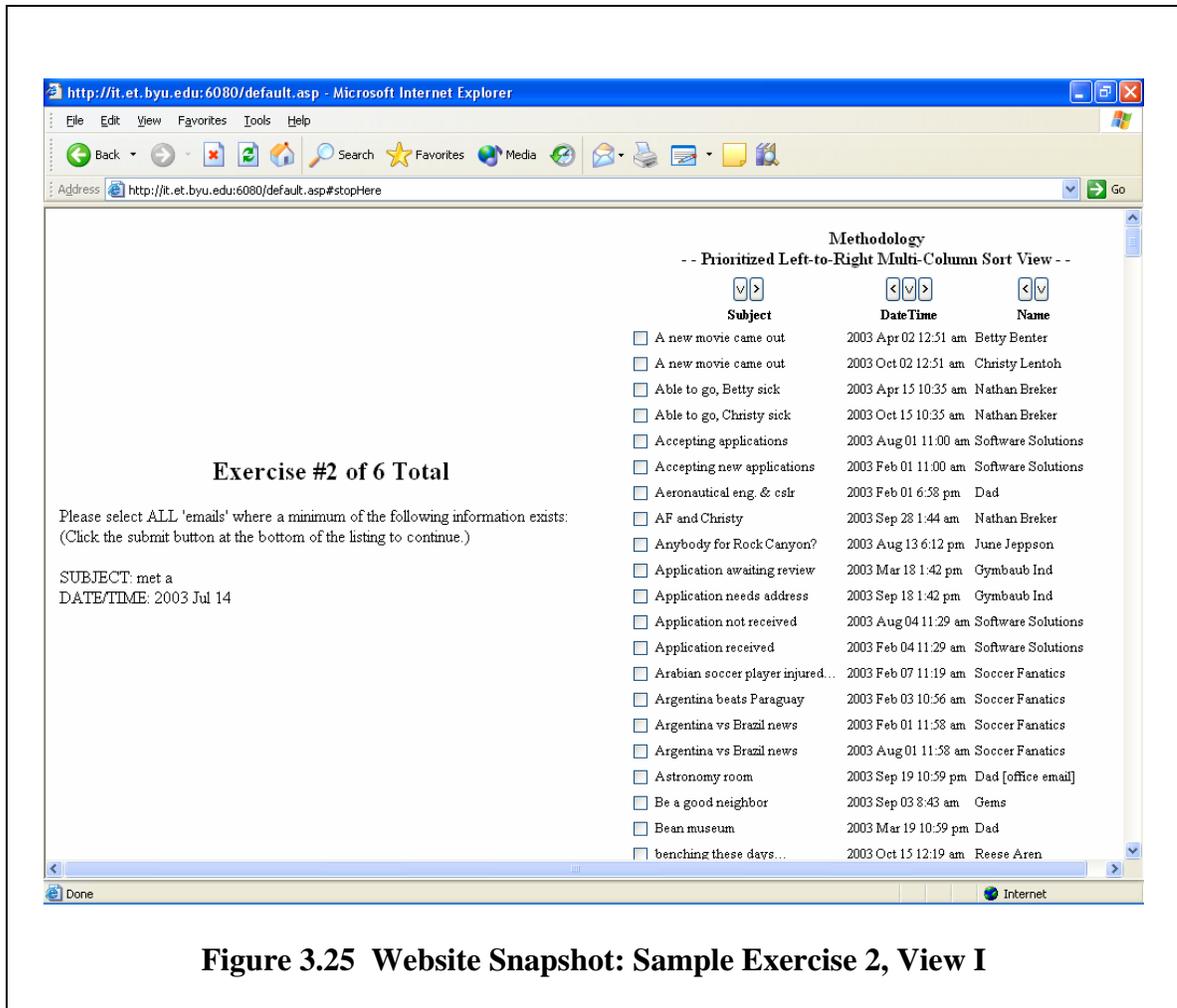


Figure 3.25 Website Snapshot: Sample Exercise 2, View I

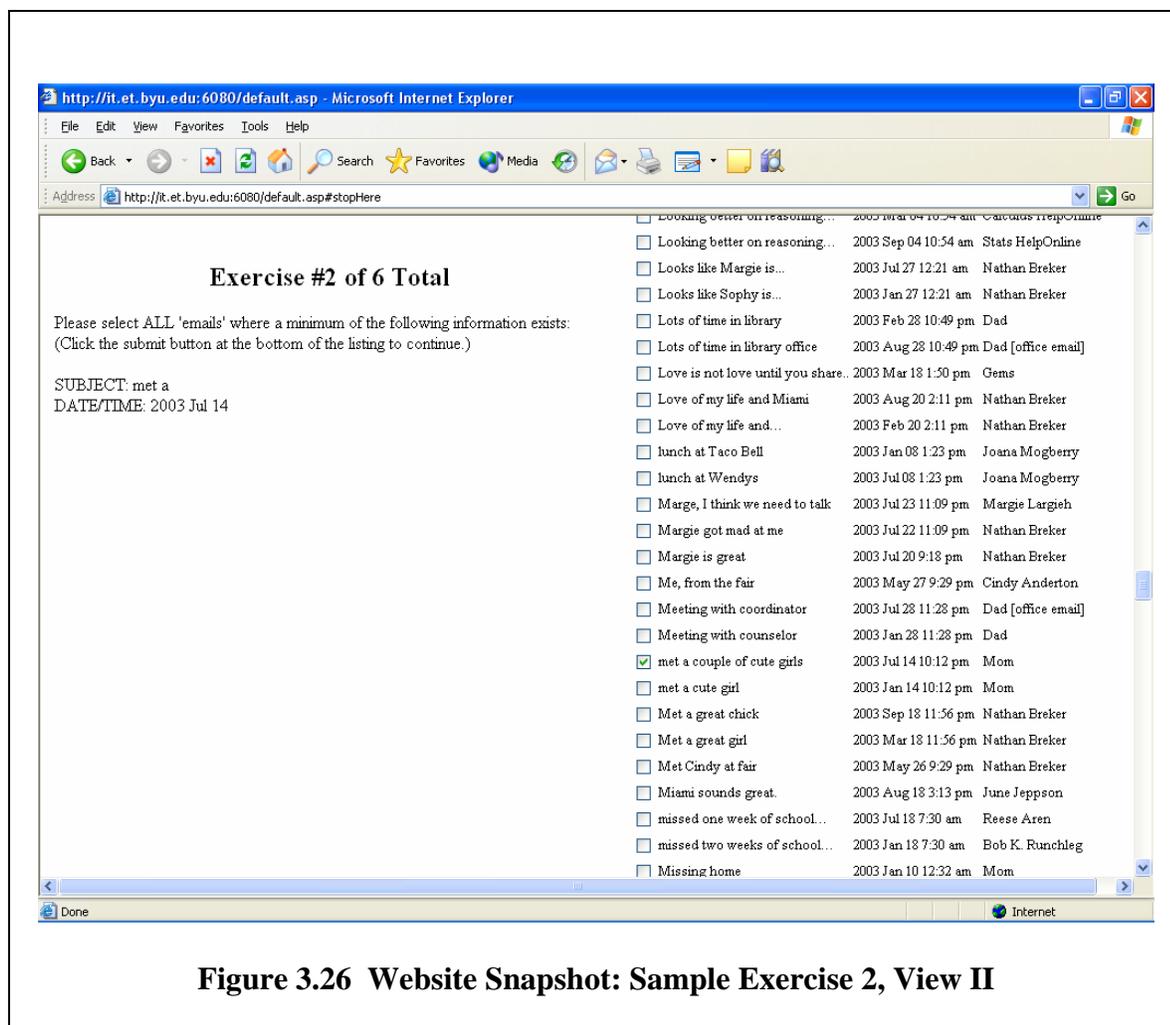


Figure 3.26 Website Snapshot: Sample Exercise 2, View II

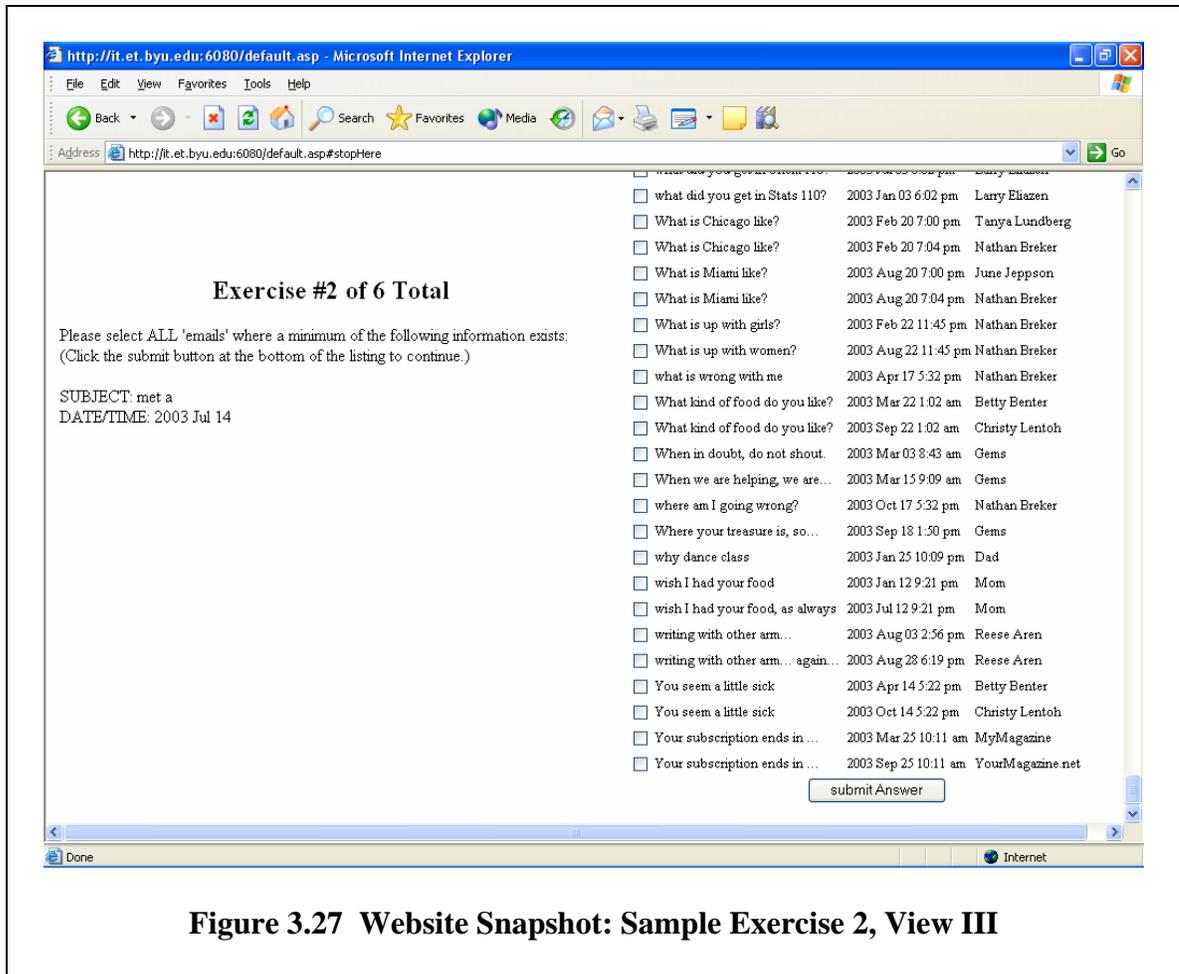


Figure 3.27 Website Snapshot: Sample Exercise 2, View III

Figure 3.28 illustrates the third exercise. This exercise happened to result in a search for records based upon Name and Date and given the PSDA interface. After a preferred sort order had been established, the participant could then find the applicable records as found in Figure 3.29. Important to note with this exercise is that values found for any two parameters might result in more than one record match.

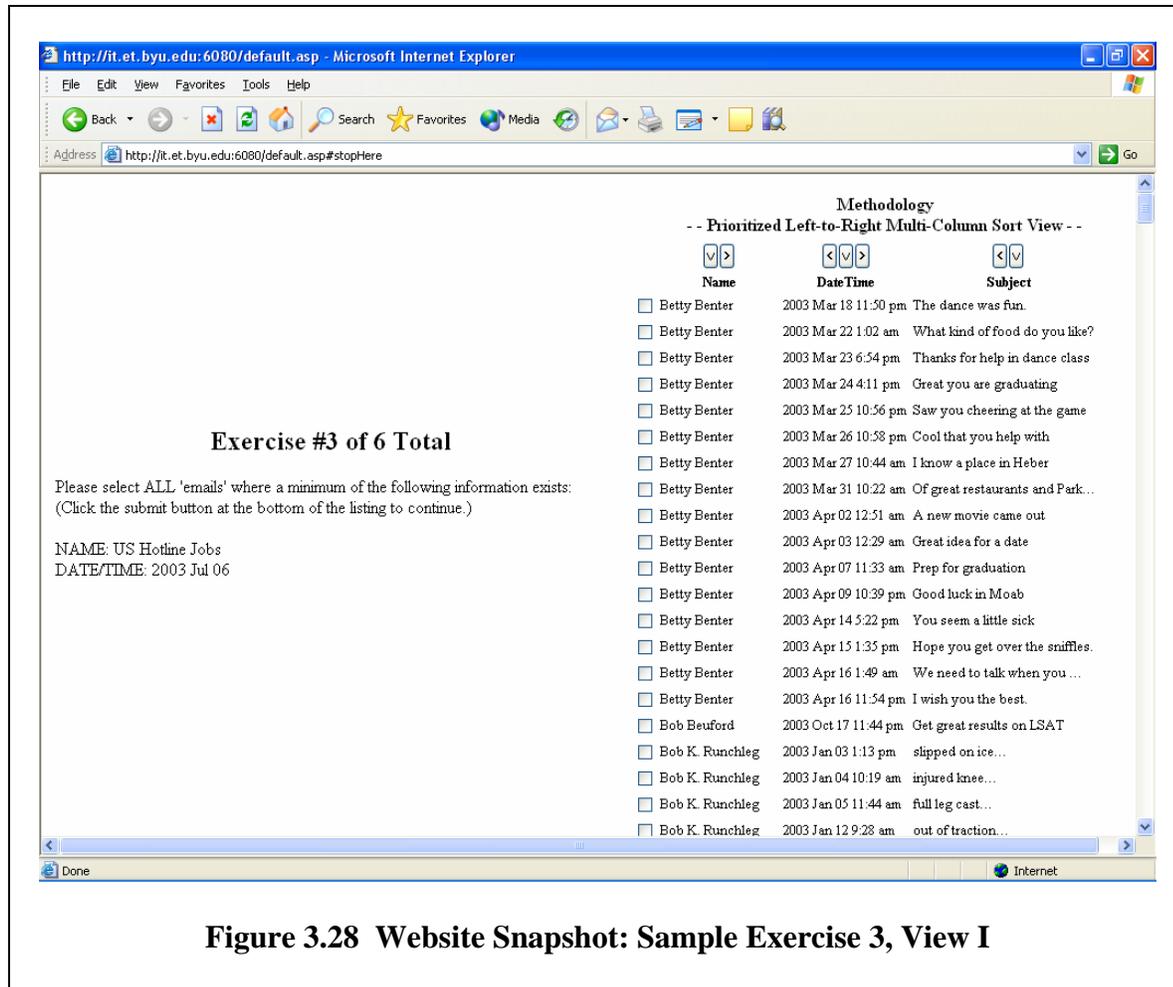


Figure 3.28 Website Snapshot: Sample Exercise 3, View I

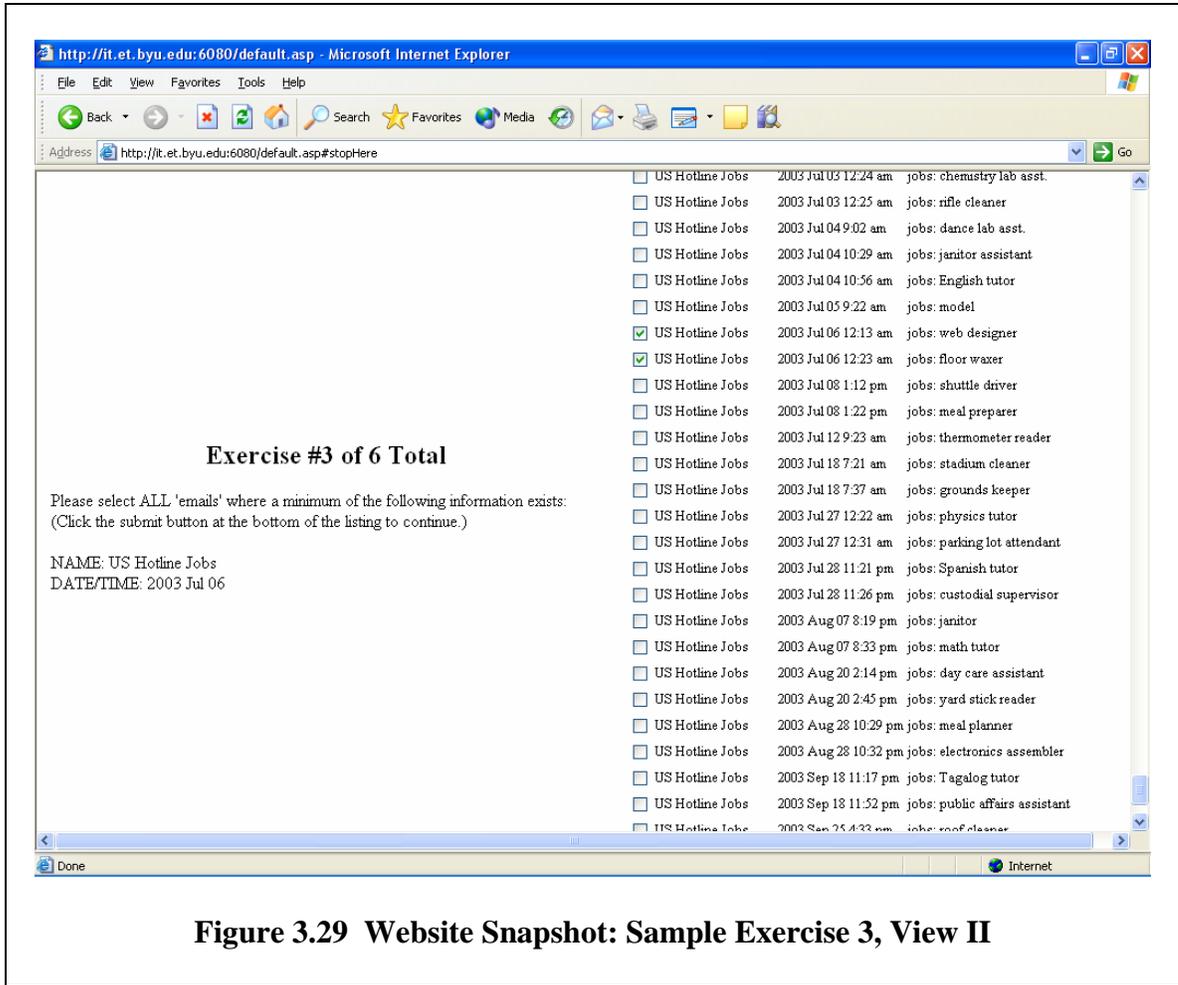


Figure 3.29 Website Snapshot: Sample Exercise 3, View II

Figure 3.30 illustrated the fourth exercise. In this case, once the Name column had been sorted, the corresponding record was easily located.

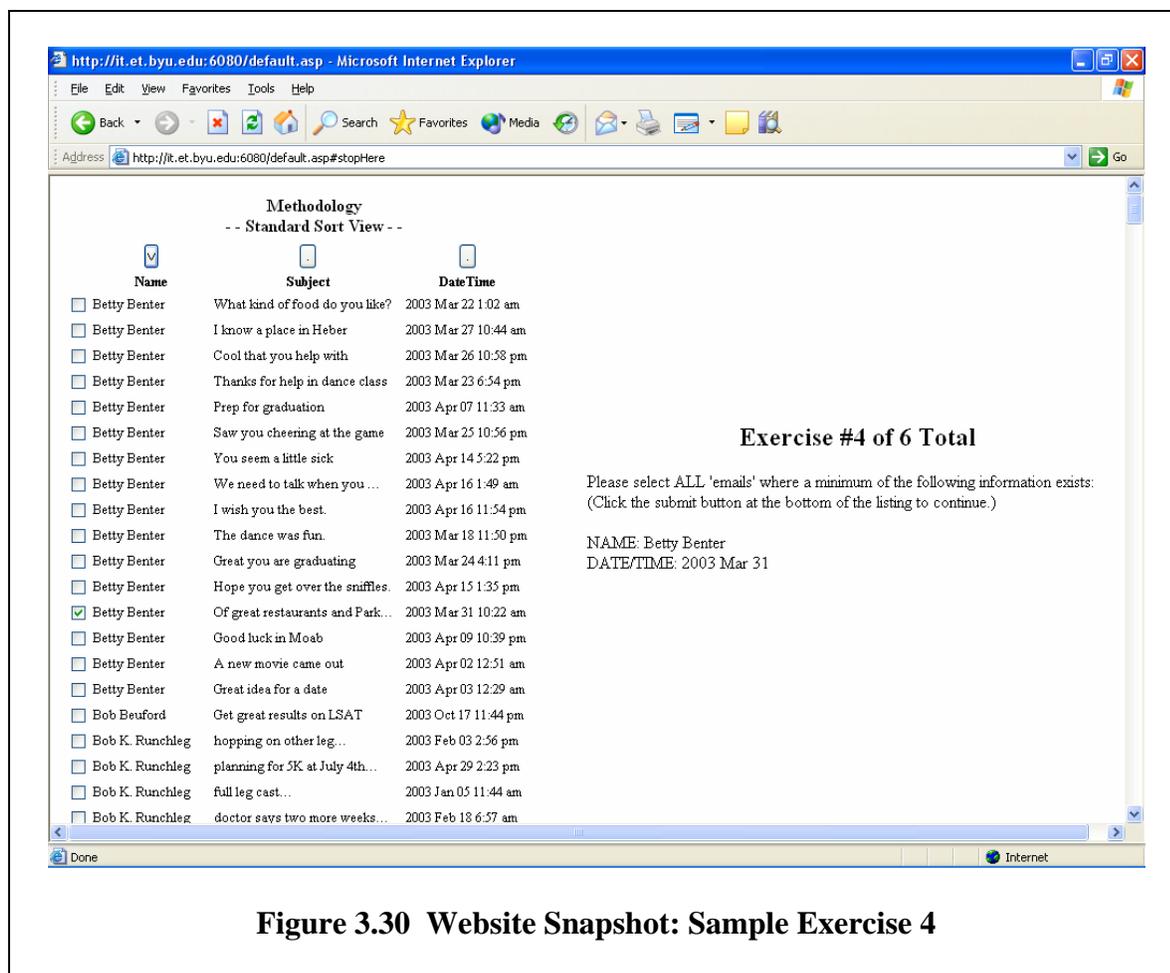


Figure 3.30 Website Snapshot: Sample Exercise 4

Figure 3.31 illustrates exercise five. This exercise has the user identify records based upon data provided for the Subject and Date columns. The participant could choose sorting by the date. Normally, records organized on the date column begin the presentation with the latest date first and sequentially by date show the earliest records last. In the illustration in Figure 3.31, the Date column was clicked once to first select the Date column and then a second time to reverse the order of appearance of records from earliest date to latest date. Figure 3.32 illustrates the corresponding record for the given criteria.

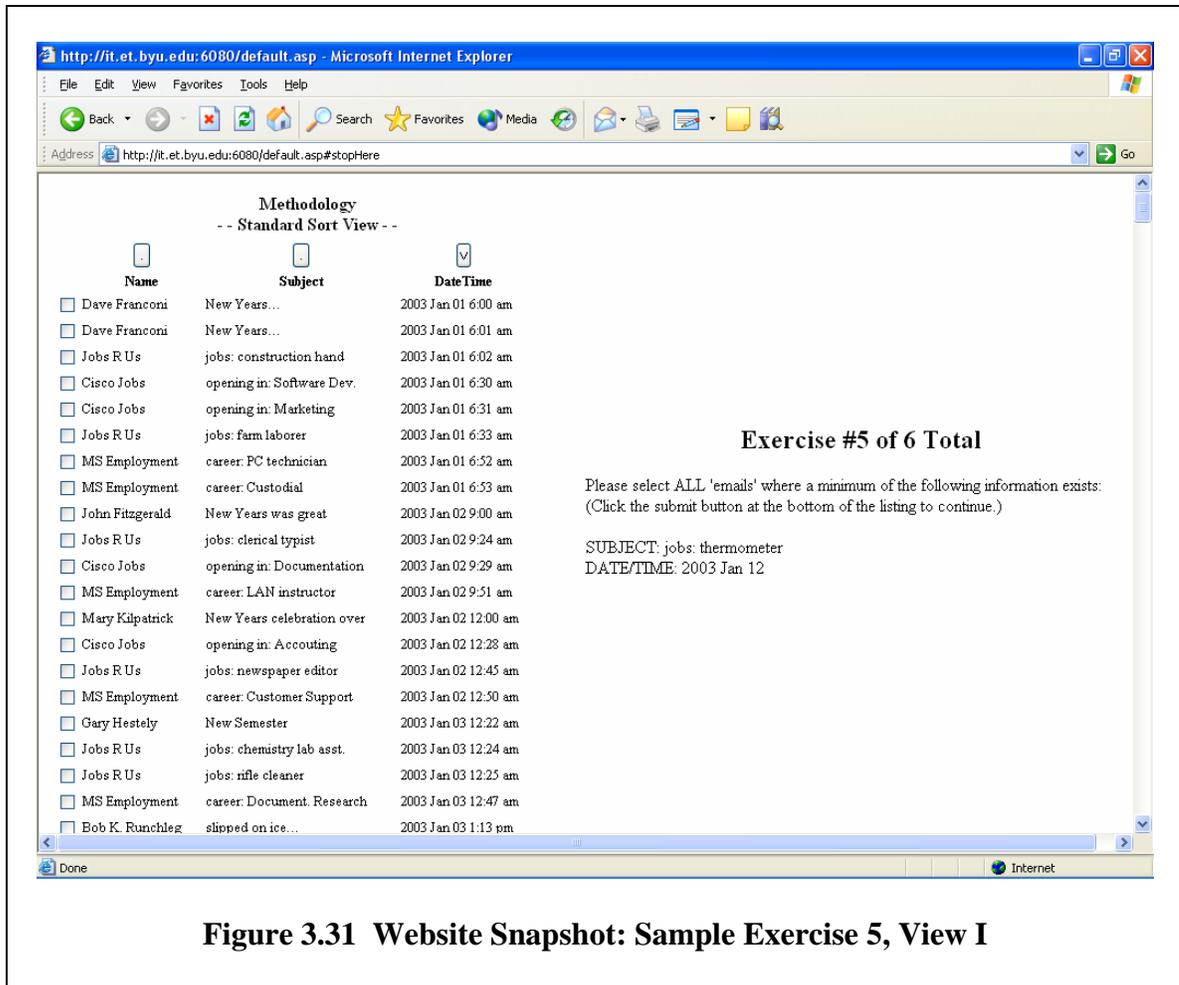


Figure 3.31 Website Snapshot: Sample Exercise 5, View I

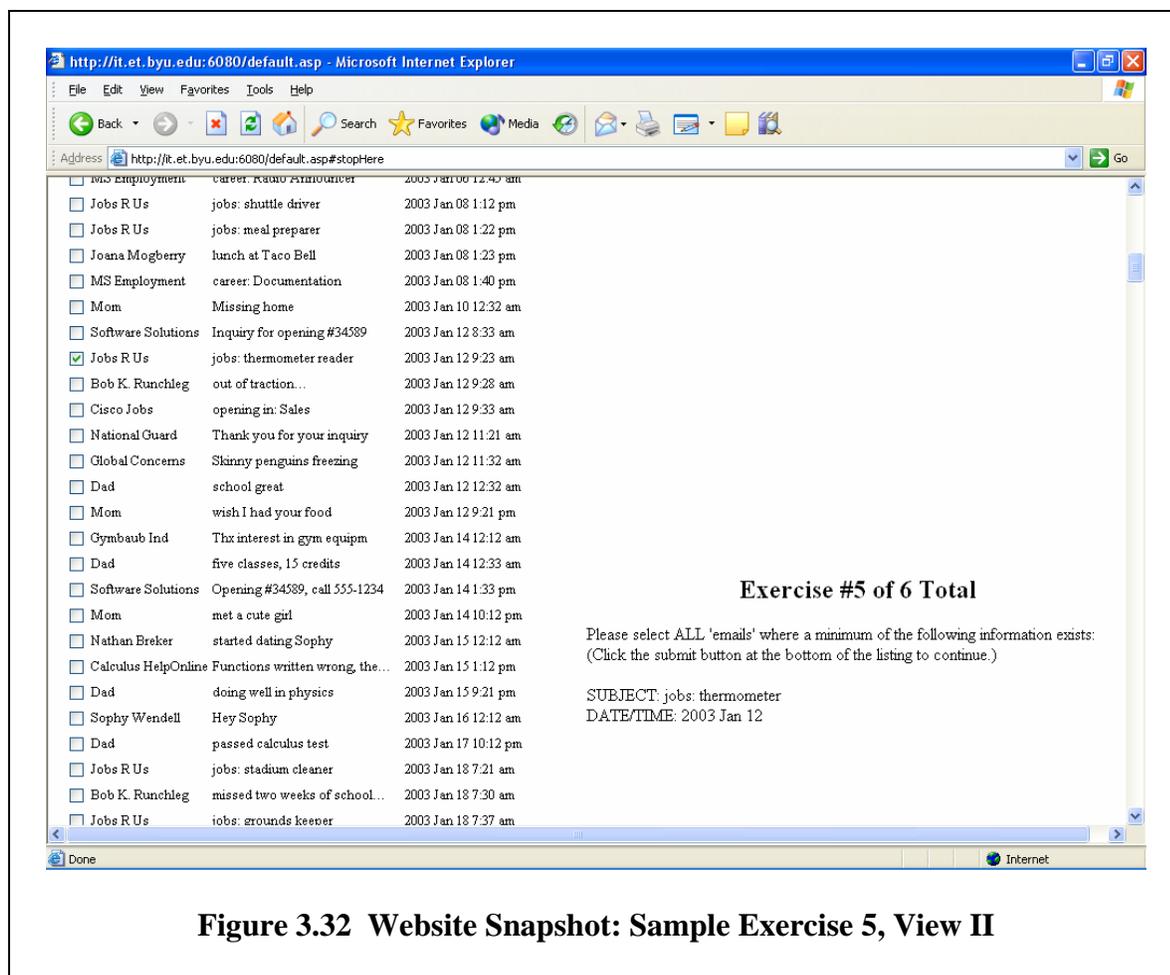


Figure 3.32 Website Snapshot: Sample Exercise 5, View II

Figure 3.33 illustrates the sixth and final exercise for this participant. The provided criteria are based upon the Name and Subject fields and the records are located using PSDA. In this case, a participant could base sorting upon the Name column first then the Subject column. Figure 3.34 illustrates the record corresponding to this criteria.

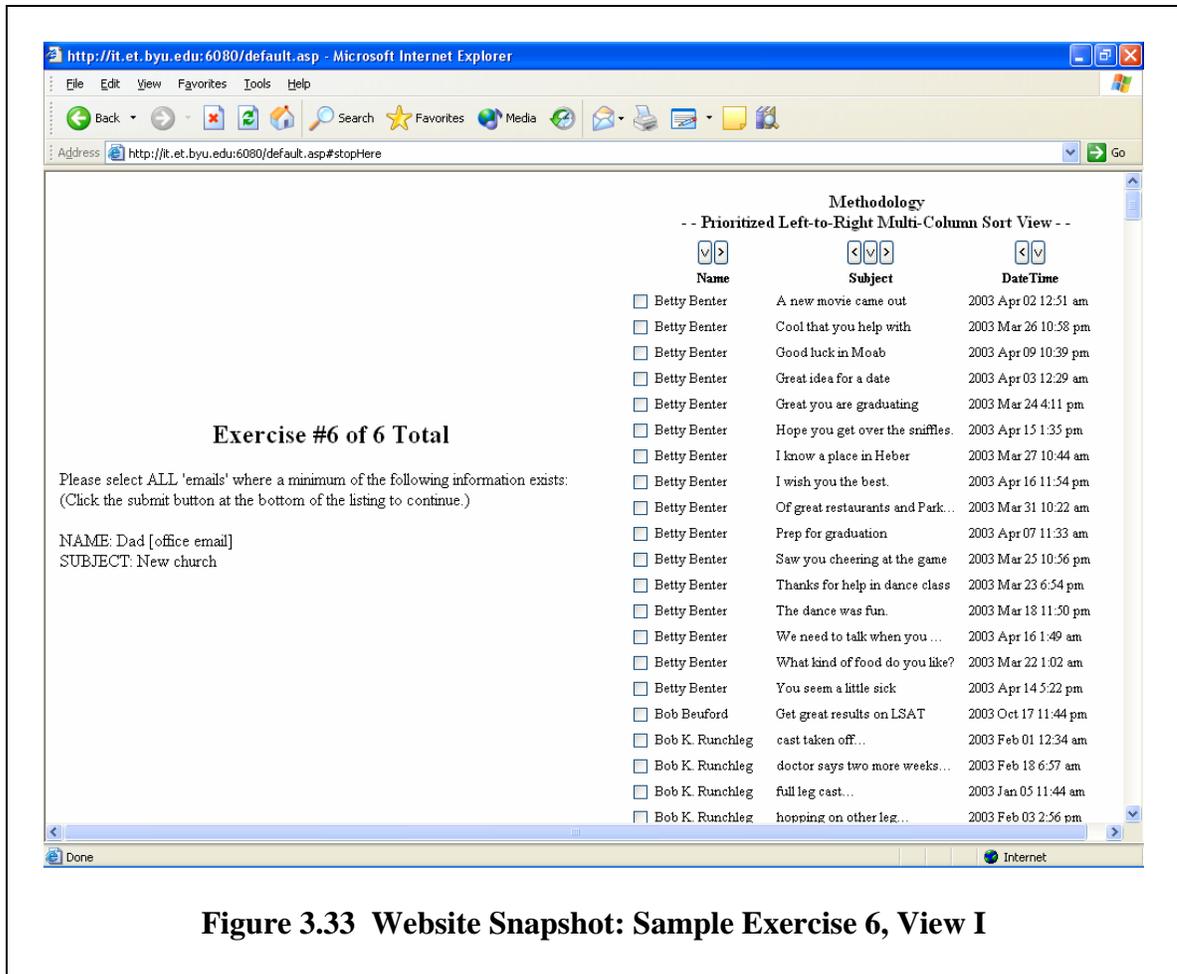


Figure 3.33 Website Snapshot: Sample Exercise 6, View I

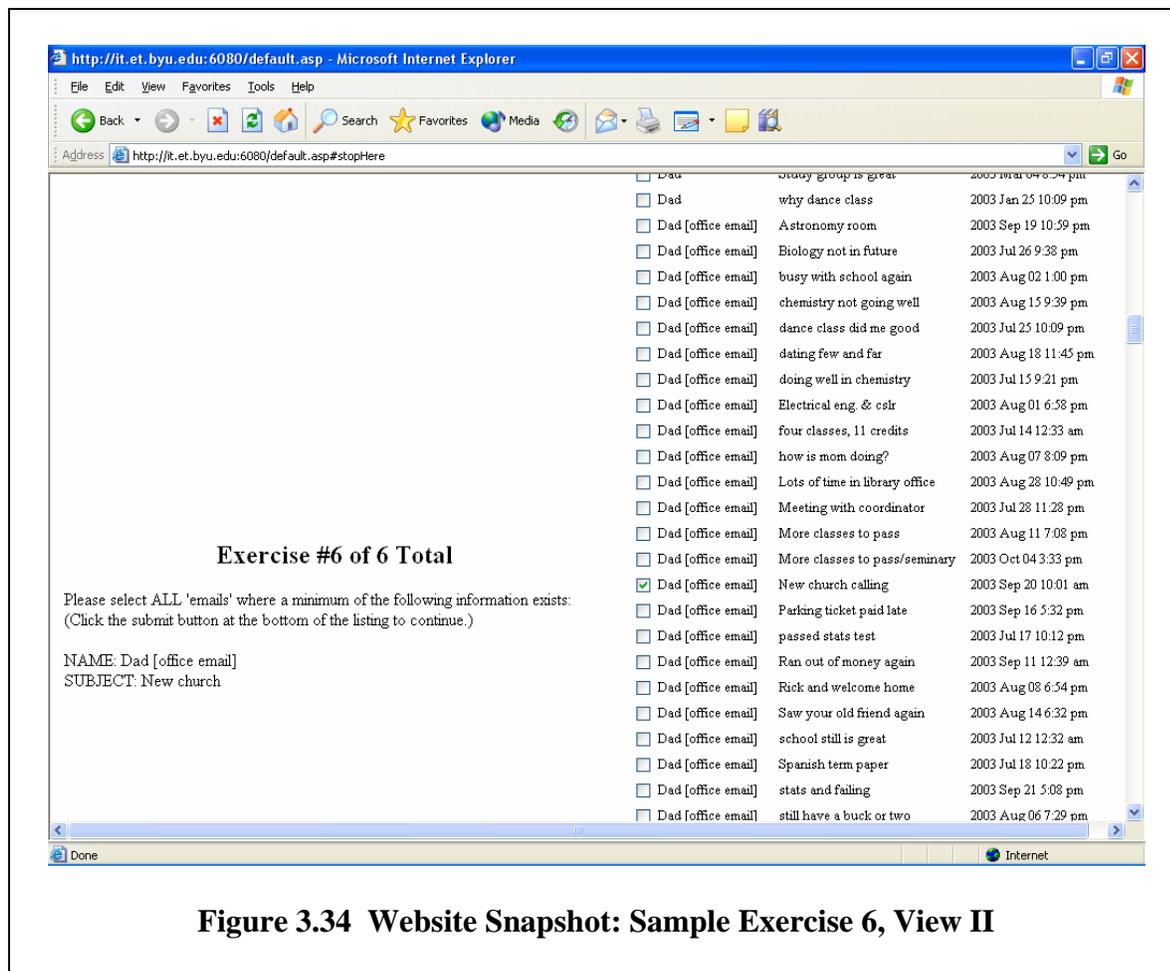


Figure 3.34 Website Snapshot: Sample Exercise 6, View II

When the participants completed the exercises, they were asked to respond to a questionnaire. There were twenty-one questions which appeared in an unsystematic order. Each participant eventually viewed all of the same survey questions, however, not in the same sequence. The purpose for presenting these in an arbitrary sequence was also to remove the possibility of bias.

The questions are provided below. The acronym, QTID, stands for Questionnaire Transferable Identification. It was a term used during programming. Because the survey

questions were not presented to each participant in the same order, the acronym has been useful as a reference for correlating responses to originating questions.

QTID.3

Question: I got confused by moving the columns back and forth when using the Prioritized Left-to-Right Multi-Column sort.

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

QTID.4

Question: I liked sorting on multiple-columns by transposing or shifting an intact column left or right (as found with the Prioritized Left-to-Right Multi-Column Sort).

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

QTID.5

Question: I got confused by having to sort with only one column with the Standard Sort (single column method).

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

QTID.7

Question: Regardless of the sorting method used, I prefer being able to sort on more than one column.

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

QTID.8

Question: Once I had selected the appropriate column selection using the Standard (single column) sort, it was fairly simple to find the exercise answer.

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

QTID.9

Question: Once I had my columns properly arranged with the Prioritized Left-to-Right Multi-Column Sort, it was fairly simple to find the answer to the question.

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

QTID.13

Question: Given the choice, which method of sorting would you like other programs you use to have?

Standard Sort (single column)

Prioritized Left-to-Right Multi-Column Sort

Both

Neither

Undecided

QTID.16

Question: I have used the Prioritized Left-to-Right Multi-Column sort in other applications?

True

False

Not sure

QTID.17

Question: I have used the Standard (single-column) sort in other applications?

True

False

Not sure

QTID.18

Question: How often do you use computers? (select most correct)

Daily

Every Other Day

Once Per Week

Several Times per Month

Once per Month

QTID.19

Question: What level of computer expertise do you self-evaluate yourself as having?

I know how to turn on a computer, log in, and check my email.

I can do the above task as well as use basic features of word processing tools such as Word and other applications.

I can do the above tasks as well as use advanced features of word processing tools such as Word and other applications.

I feel comfortable with computers and can figure out how to get around most applications and my operating system needs such as install programs and some troubleshooting.

I can do the above as well as manipulate computer software with programming and more.

QTID.22

Question: I liked sorting on only one column (as found with the Standard (single-column) sort.

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

QTID.26

Question: My gender:

Female

Male

QTID.27

Question: My age (yrs):

12-25

26-33

34-41

42-49

50+

QTID.28

Question: I believe that overall I was more accurate in my responses when using the Standard (single-column) sort.

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

QTID.29

Question: I believe that overall I was more accurate in my responses when using the Prioritized Left-to-Right Multi-Column sort.

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

QTID.30

Question: I am enrolled in the following course at BYU:

IT 101

IT 355

IT 391

More than one

None

QTID.31

Question: The Standard (single-column) sort methodology is more user friendly.

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

QTID.32

Question: The Prioritized Left-to-Right Multi-Column sort methodology is more user friendly.

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

QTID.33

Question: I believe that overall I was quicker at answering the exercise questions when using the Standard (single-column) sort.

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

QTID.34

Question: I believe that overall I was quicker at answering the exercise questions when using the Prioritized Left-to-Right Multi-Column sort.

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

Figures 3.35 through 3.41 illustrate the sequence of survey questions which may have appeared for some users. Again, all the same questions were provided to each participant.

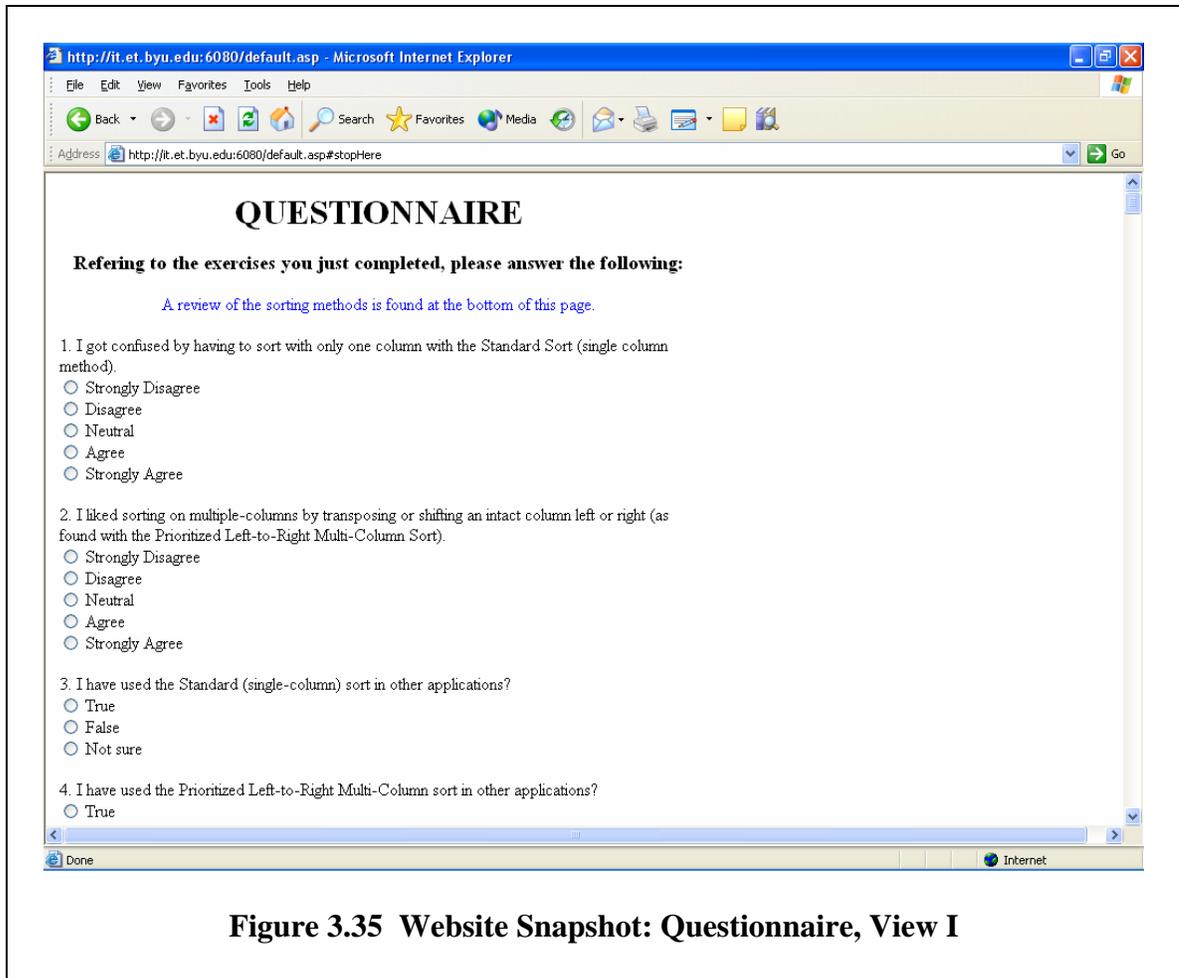


Figure 3.35 Website Snapshot: Questionnaire, View I

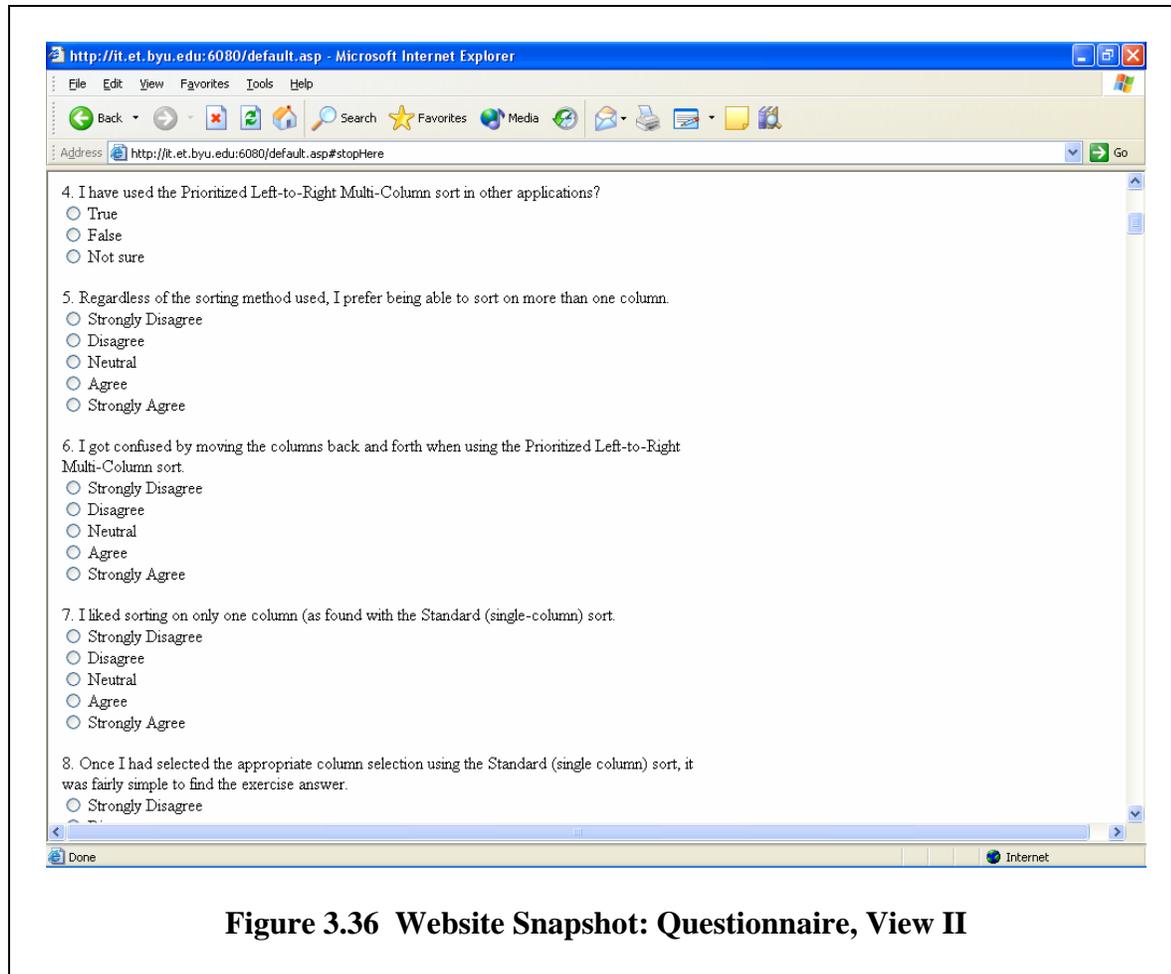


Figure 3.36 Website Snapshot: Questionnaire, View II

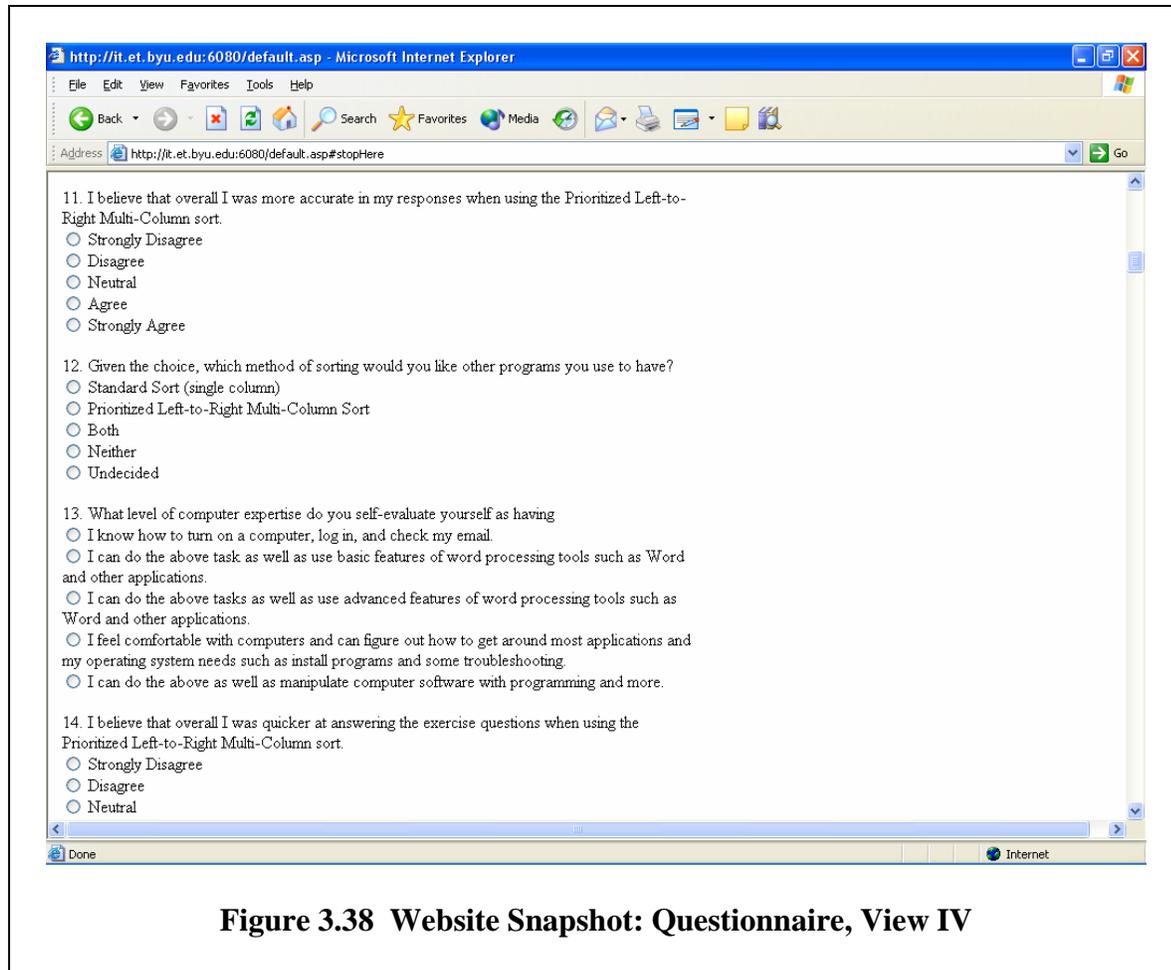


Figure 3.38 Website Snapshot: Questionnaire, View IV

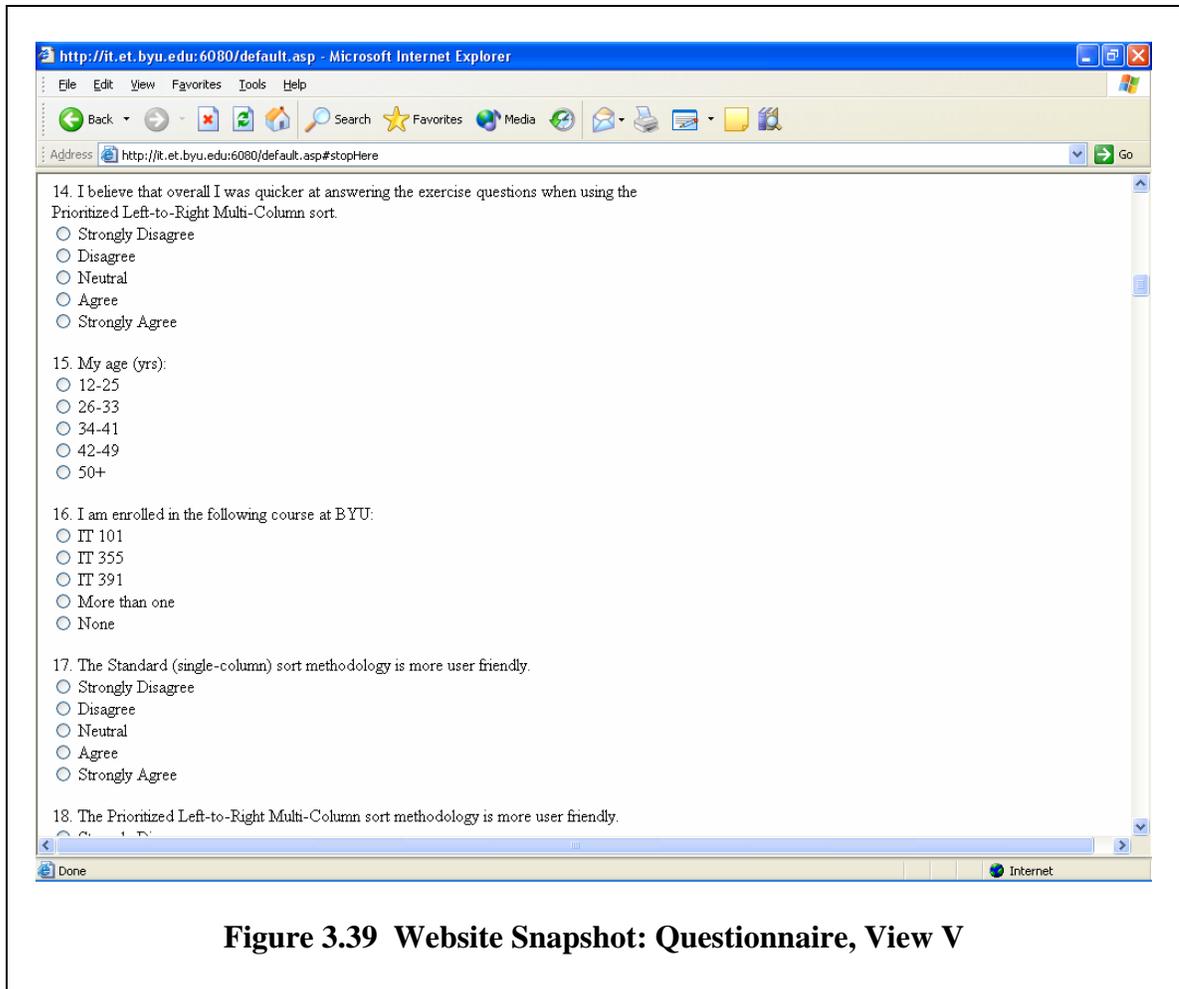


Figure 3.39 Website Snapshot: Questionnaire, View V

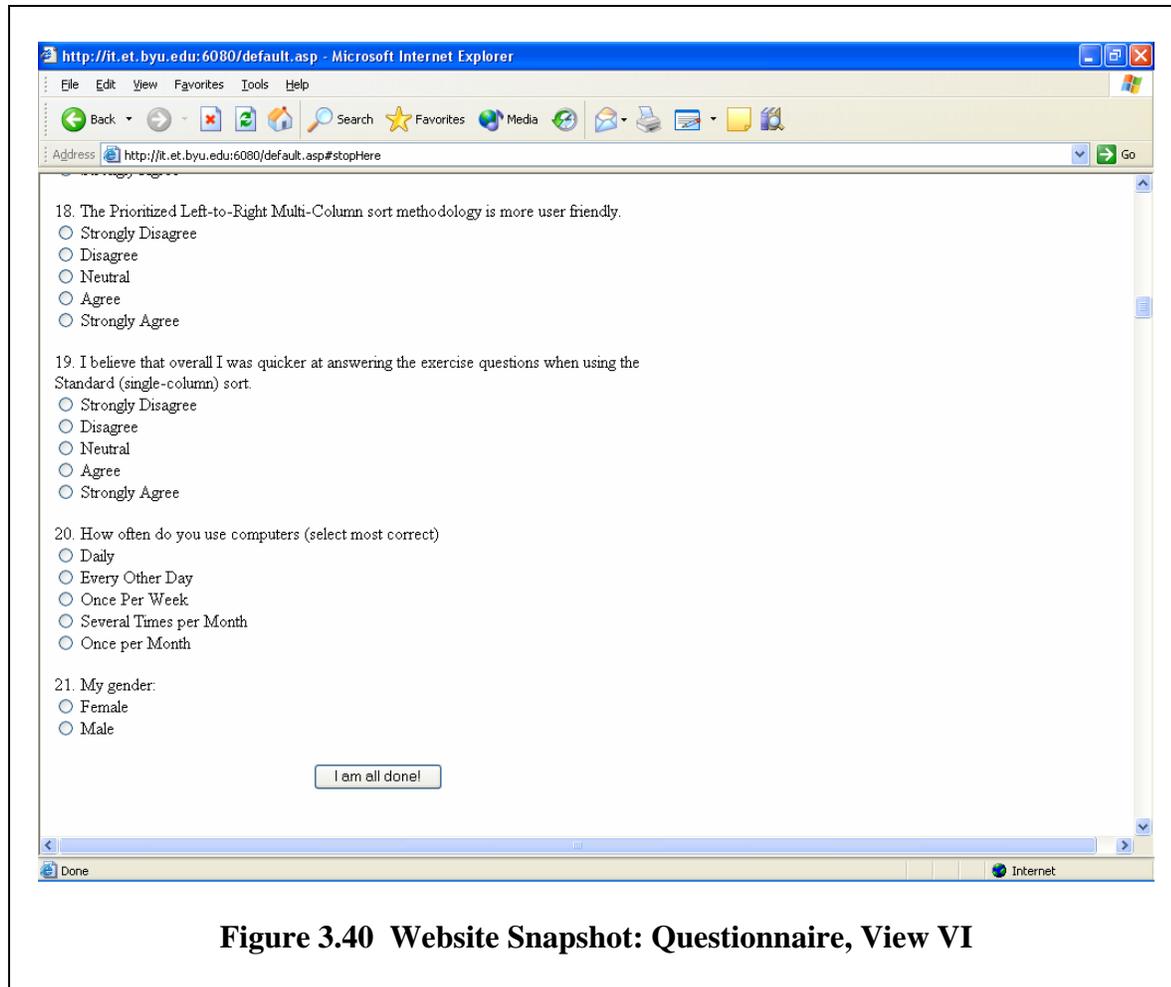


Figure 3.40 Website Snapshot: Questionnaire, View VI

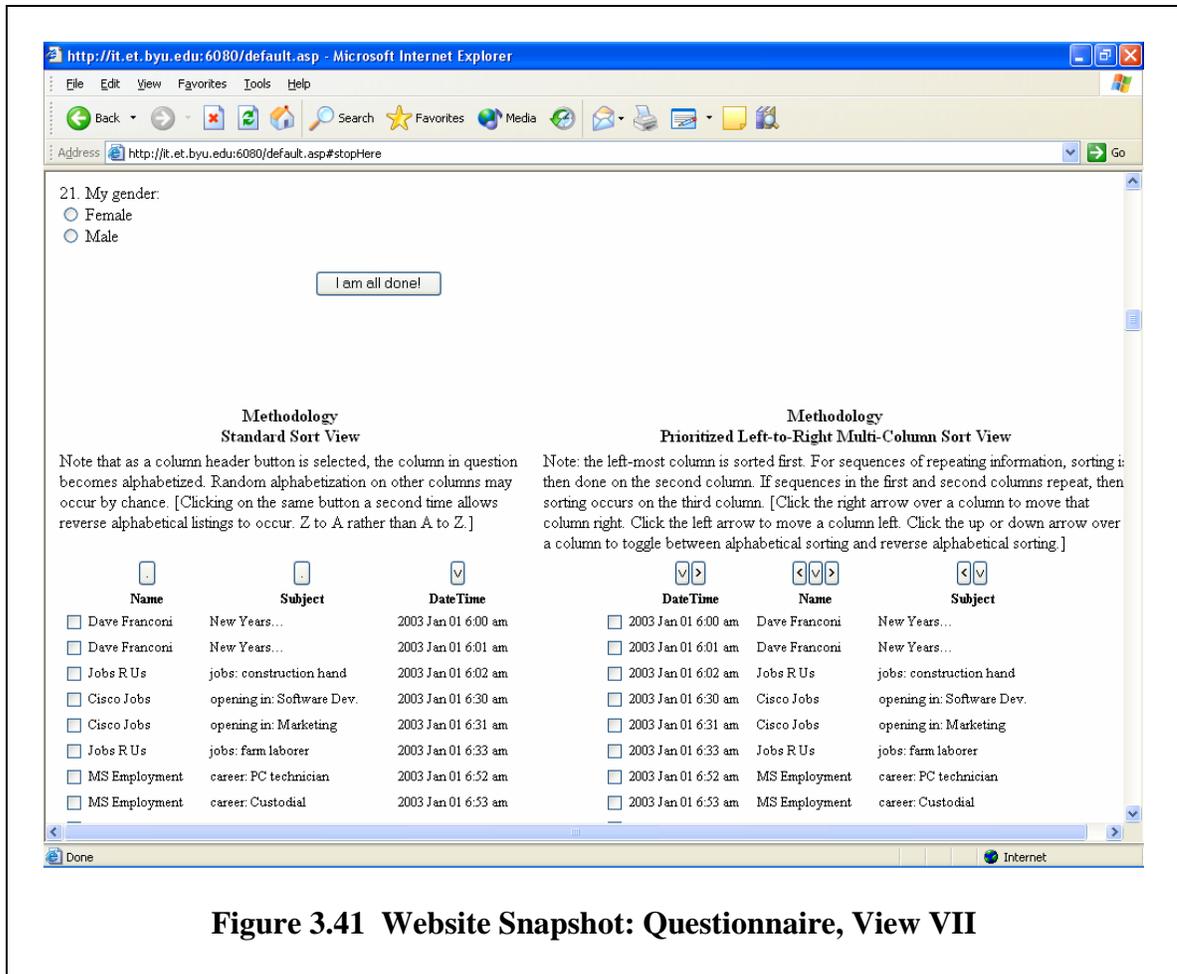


Figure 3.41 Website Snapshot: Questionnaire, View VII

Additional functionality was built into the website to enable quick access to the results. This portion was called the Administration Page and was only accessible by the research team. Figures 3.42, 3.43, and 3.44 represent a display of the Exercise and Questionnaire results, and the Participants' list respectively. The information generated in these sections was ported into Microsoft® Excel for further analysis.

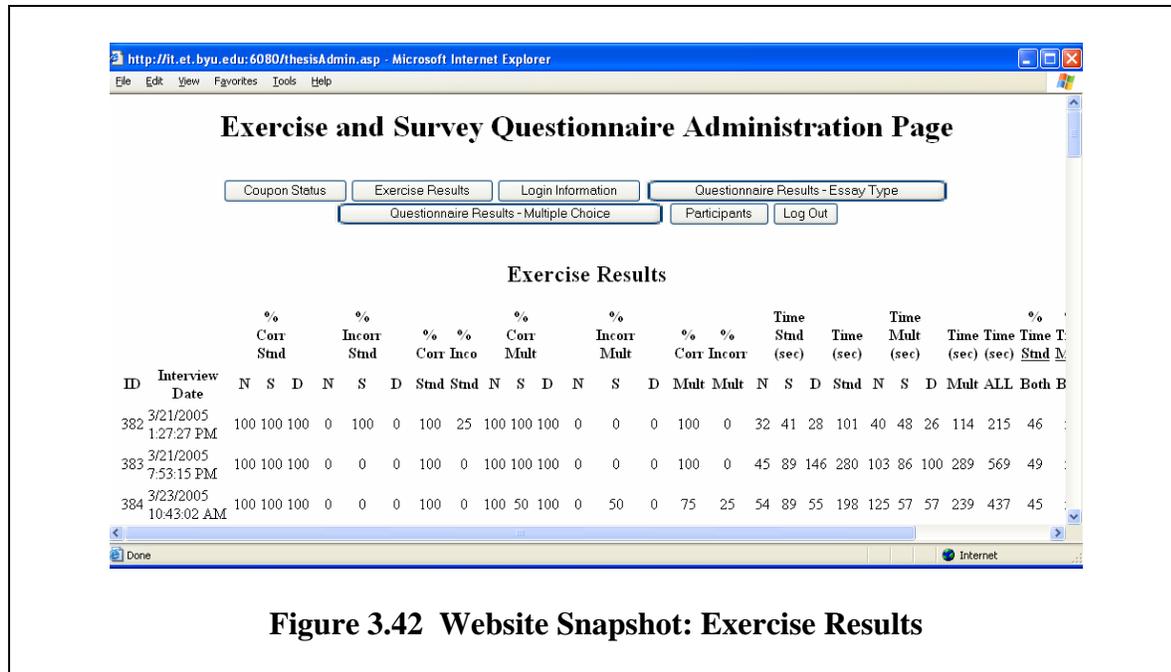


Figure 3.42 Website Snapshot: Exercise Results

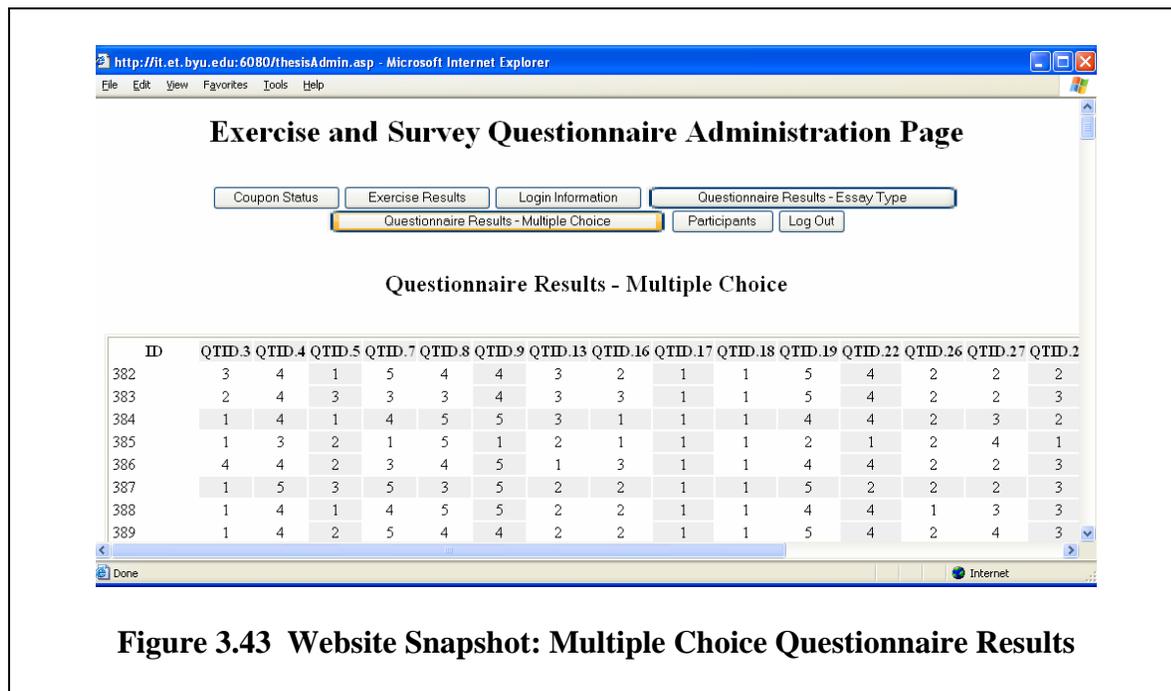


Figure 3.43 Website Snapshot: Multiple Choice Questionnaire Results

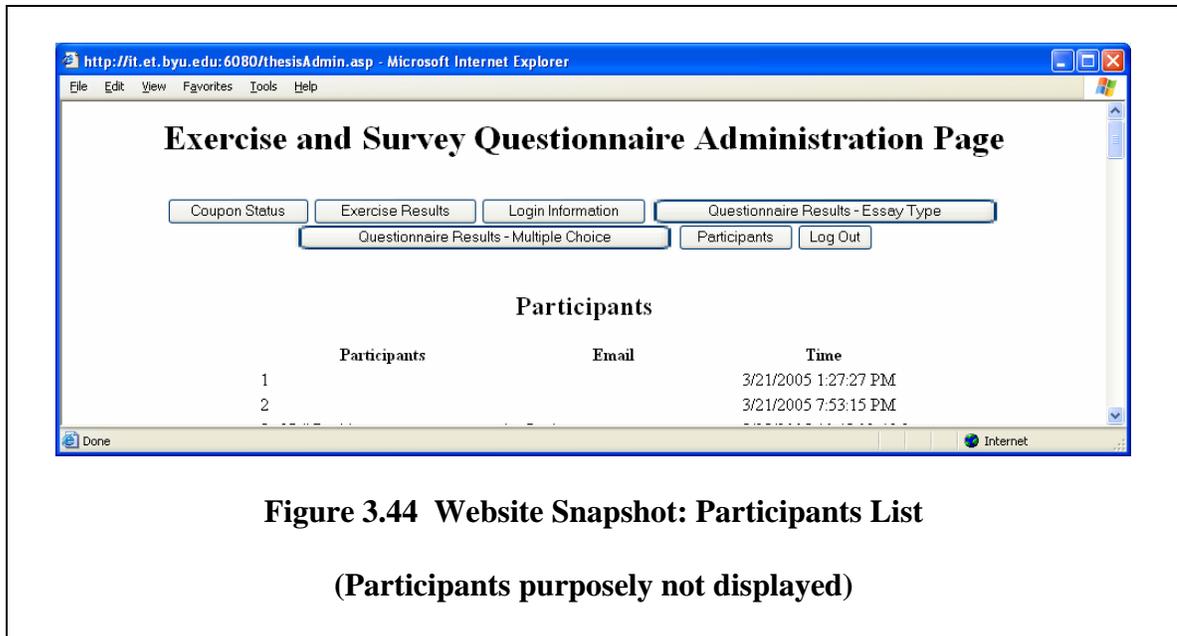


Figure 3.44 Website Snapshot: Participants List

(Participants purposely not displayed)

MAINTAINING CONFIDENTIALITY

To assure that participants felt they could express themselves honestly and openly, a guarantee of confidentiality was promised. At the end of the survey, however, an option was provided for them to give their name and contact information. If the participants left their contact information and if there were questions about the results, participants might have been contacted. These measures should have provided sufficient assurance to the users that their answers were confidential.

SUMMARY

Usability of the feature was tested using web-based exercises which were compared concurrently using the standard-sort method. A survey followed the exercises which elicited feedback in the form of measured assessment from participants to consider the experimental method's usefulness.

CHAPTER 4

RESULTS AND FINDINGS

Data collection for this thesis followed the methodology discussed within the previous chapter. Approximately two-hundred contacts within academia and professional or personal environments were invited to take part in the web-based survey. Eighty-one or approximately 40.5% accepted. Participants registered backgrounds, ages, and computer competency skills. Individuals participating might have had careers, part-time employment, no employment or have been retired. Some were full or part-time university students; some were not students. At the time of the survey, less than 5% of participants were enrolled in one or more of the following BYU IT courses: 101, 355, and 391 (QTID.30).

Of those surveyed, 30.9% were between the ages of 12 and 25, 23.5% were between the ages of 26-33, 22.2% were between the ages of 34-41, 14.8% were between the ages of 42-49, and 8.6% were over the age of 50 (QTID.27). 35.8% of respondents were female and 64.2% were male (QTID.26).

Participants had varying backgrounds and expertise with computers. Of those surveyed, 87.7% worked with computers daily, 8.6% used a computer every other day, 2.4% used a computer once per week, and 1.2% indicated that they used a computer once per month (QTID.18). When asked what level of computer expertise they possessed, 6.2% knew how to turn on a computer, log in, and check their email; 6.2% could do these

as well as use basic features of word processing programs such as Word; 18.5% could do these as well as manipulate advanced features of word processing programs such as Word; 50.6% could do these as well as navigate most applications and manage operating system needs such as installing programs and troubleshooting; 18.5% could do these as well as manipulate computer software through programming skills and more (QTID.19).

Not all potential participants took part in the study. Several said their computers were unable to make the network connection to the survey web-page. One or two claimed they were kicked off the system. No statistics were kept for these issues.

COMBINED RESULTS

Several survey questions elicited preference based responses. Participants were asked to state their degree of preference by selecting among the following options: 1) Strongly Disagree, 2) Disagree, 3) Neutral, 4) Agree, and 5) Strongly Agree.

By recommendation and for statistical purposes, the *Strongly Disagree* and *Disagree* levels of preference were combined as one, Disagree. Also combined were *Agree and Strongly Agree* as Agree. These changes were advised because the level of reliability of the conclusions of the study increased as possible selections decreased; furthermore, relatively few participants had selected the extremes.

NON-RELIABLE SURVEY FEEDBACK

Datum counts for each survey question were tested with the Chi-square formula which determined whether category frequencies were reliably different from each other. The reliability calculations for these survey questions are found in the appendix.

- QTID.22: Of those surveyed, 39.5% respondents preferred sorting on only one column as found with the Standard (single-column) Sort, 28.4% were neutral, and 32.1% did not like this sorting method.
- QTID.28: 39.5% were neutral as to whether their responses were more accurate when using the Standard Sort, 24.7% responded they were more accurate, and 35.8% responded that they were not more accurate.
- QTID.31: 44.2% responded that the Standard Sort was more user friendly, 25.9% were neutral, and 29.6% responded that it was not more user friendly.
- QTID.32: 39.5% were neutral as to whether the Prioritized Left-to-Right Multi-Column Sort was more user friendly, 32.1% responded that it was not more user friendly, and 28.2% indicated that it was more user friendly.
- QTID.33: 35.8% responded that they were quicker at answering the exercise questions when using the Standard Sort, 32.1% were neutral, while 32.1% responded that they were not quicker.

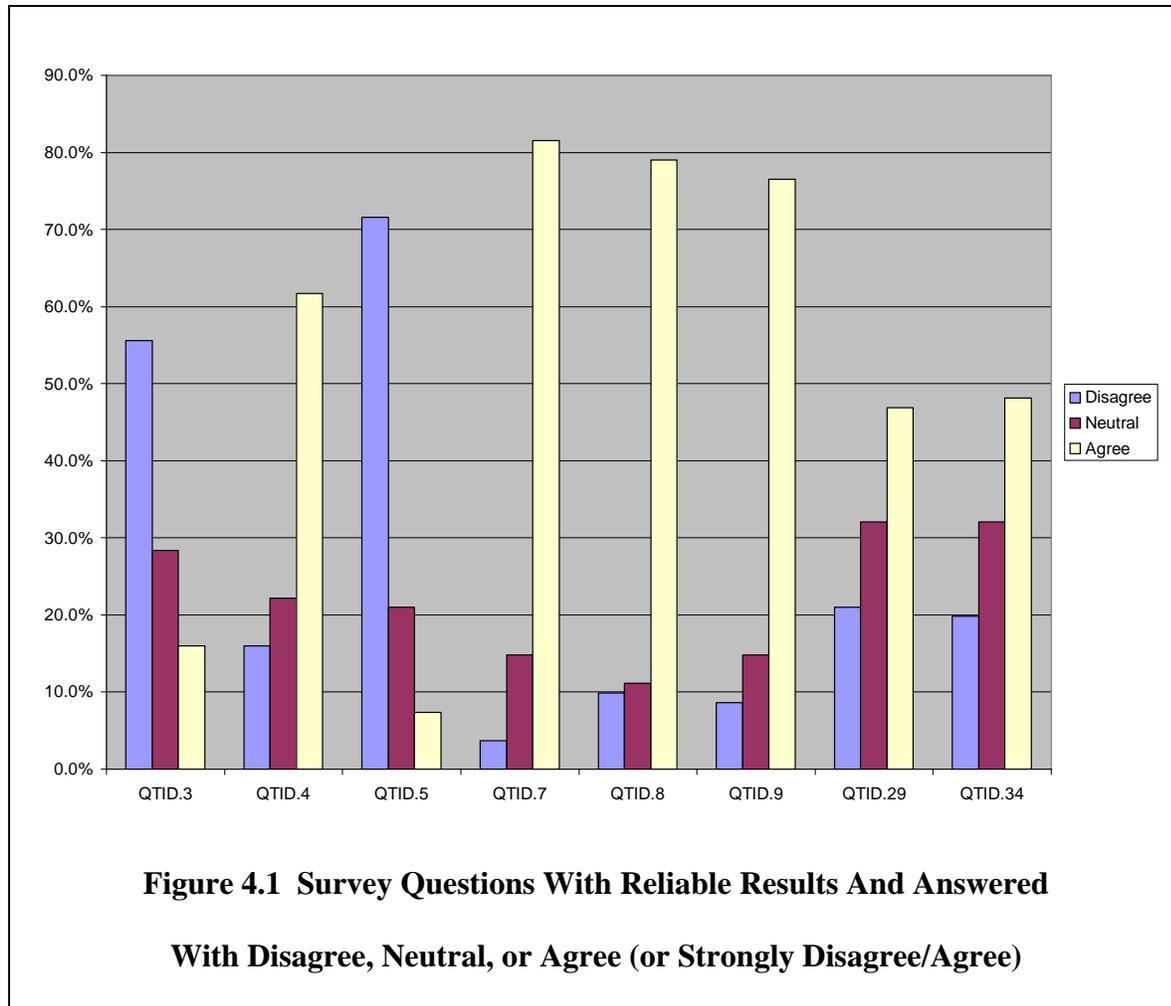
RELIABLE SURVEY FEEDBACK

Most of the survey questions returned feedback that according to the Chi-square formula was determined to be reliable. These are listed below.

- QTID.3: 55.6% did not get confused by moving the columns back and forth when using the Prioritized Left-to-Right Multi-Column Sort; 28.4% were neutral, and 16% were confused.
- QTID.4: 61.7% liked sorting on multiple-columns by transposing or shifting an intact column left or right (as found with the Prioritized Left-to-Right Multi-Column Sort); 22.2% were neutral, while 16% did not like it.

- QTID.5: 71.6% responded that to some degree they did not get confused by having to sort with only one column as found with the Standard Sort (single column method); 21% were neutral, while 7.4% were left confused.
- QTID.7 81.1% preferred being able to sort on more than one column; 14.8% were neutral, while 3.7% did not prefer multi-column sorting.
- QTID.8 79% said that once they had selected the appropriate column selection using the Standard Sort, it was fairly simple to find the exercise answer; 11.1% were neutral, while 9.9% did not find it simple.
- QTID.9 76.5% responded that once they had the columns properly arranged with the Multi-Column Sort, it was fairly simple to find the answer to the question; 14.8% were neutral, while 8.6% did not find it fairly simple.
- QTID.13 42% would prefer their programs to employ the Multi-Column Sort; 18.5% prefer the Standard Sort, 33.3% preferred having both, 1.2% did not prefer either, and 4.9% were undecided.
- QTID.16 60.5% had not used the Multi-Column Sort in other applications; 21% had used it, and 18.5% were not sure.
- QTID.17 84% had used the Standard Sort in other applications, 8.6% had not, and 7.4% were not sure.
- QTID.29 46.9% believed they were more accurate when using the Multi-Column Sort; 32.1% were neutral, 17.3% disagreed, and 3.7% strongly disagreed.
- QTID.34 48.1% believed they answered exercise questions quicker when using the Multi-Column Sort; 32.1% were neutral, and 19.8% believed they were slower.

Figures 4.1, 4.2, and 4.3 displayed comparisons between participants' responses for those questions calculated as having had reliable feedback. Figure 4.1 represented questions or comments eliciting responses of disagreement, neutrality or agreement. The first three adjoining columns grouped together above the QTID.3 label indicated a response in which few participants, 16%, were confused by moving the columns back and forth when using the Prioritized Left-to-Right Multi-Column Sort. To the right of it above label QTID.4, many people preferred sorting on multiple-columns by transposing or shifting an intact column left or right (as found with the Prioritized Left-to-Right Multi-Column Sort). The next series of columns above label QTID.5 showed that few participants were confused by having to sort with only one column with the Standard Sort (single column method). QTID.7 showed an overwhelming agreement that regardless of the sorting method used, many survey participants preferred being able to sort on more than one column. Once participants had selected the appropriate column selection using the Standard (single column) Sort, a large majority believed it was fairly simple to find the exercise answer; noted in portion of graphic above QTID.8 graphic. Similarly for the responses displayed above label QTID.9, once participants had columns properly arranged with the Prioritized Left-to-Right Multi-Column Sort, a vast majority felt it was fairly simple to find the answer to the question. Responses for QTID.29 and QTID.34 did not produce great disparities in responses. However, the chart seemed to indicate that overall participants believed that they were more accurate in their responses when using the Prioritized Left-to-Right Multi-Column Sort. Likewise, participants seemed to believe that overall they were quicker at answering the exercise questions when using the Prioritized Left-to-Right Multi-Column Sort.

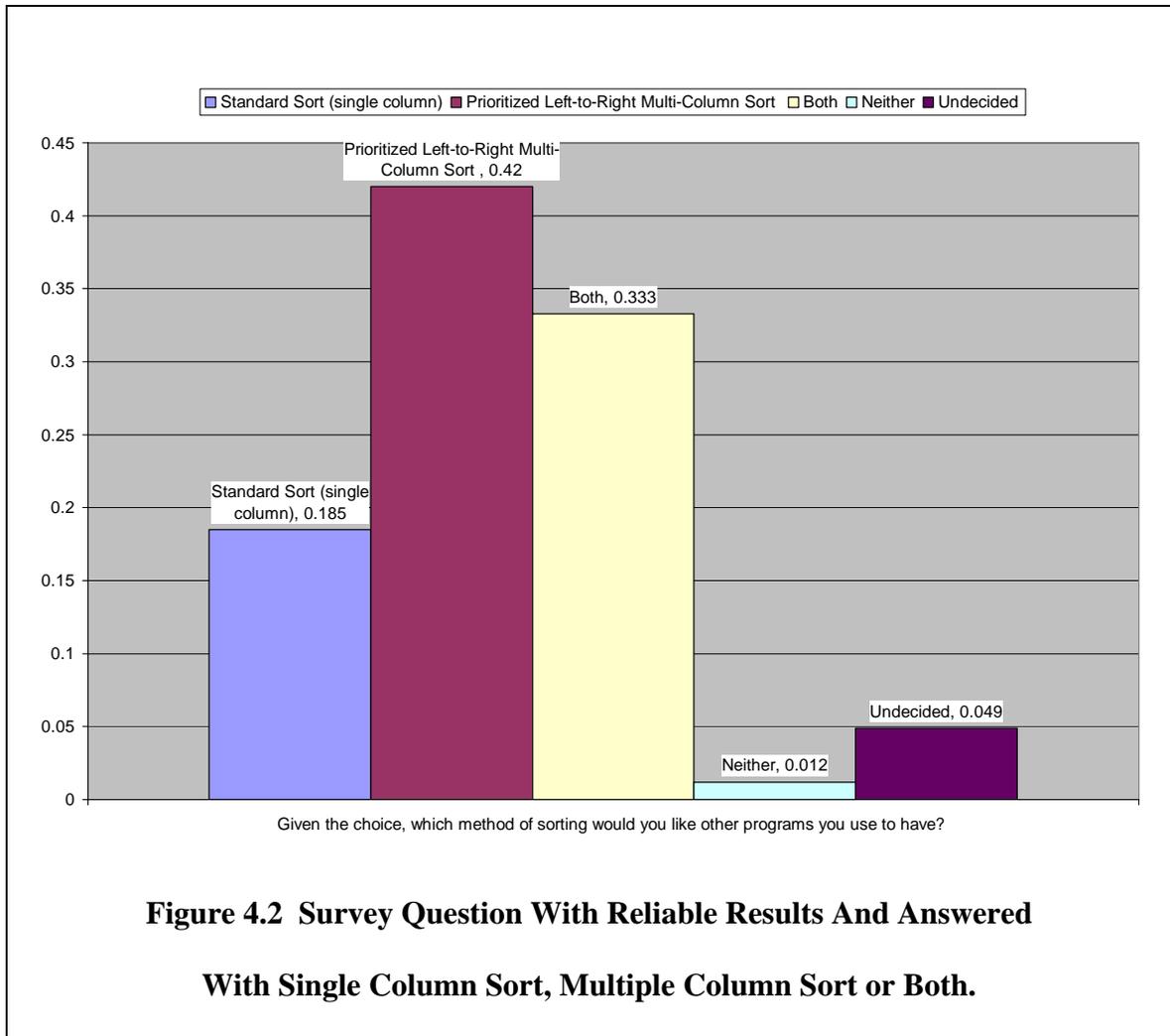


The exact questions or comments posed to participants were as follows:

QTID.3: I got confused by moving the columns back and forth when using the Prioritized Left-to-Right Multi-Column Sort.

QTID.4: I liked sorting on multiple-columns by transposing or shifting an intact column left or right (as found with the Prioritized Left-to-Right Multi-Column Sort).

QTID.5: I got confused by having to sort with only one column with the Standard Sort (single column method).



QTID.7: Regardless of the sorting method used, I prefer being able to sort on more than one column.

QTID.8: Once I had selected the appropriate column selection using the Standard (single column) Sort, it was fairly simple to find the exercise answer.

QTID.9: Once I had my columns properly arranged with the Prioritized Left-to-Right Multi-Column Sort, it was fairly simple to find the answer to the question.

QTID.29: I believe that overall I was more accurate in my responses when using the Prioritized Left-to-Right Multi-Column Sort.

QTID.34: I believe that overall I was quicker at answering the exercise questions when using the Prioritized Left-to-Right Multi-Column Sort.

Figure 4.2 represented the question, “Given the choice, which method of sorting would you like other programs you use to have?” As was noticeable, a majority expressed an interest in having multi-column sorting available in programs they had used.

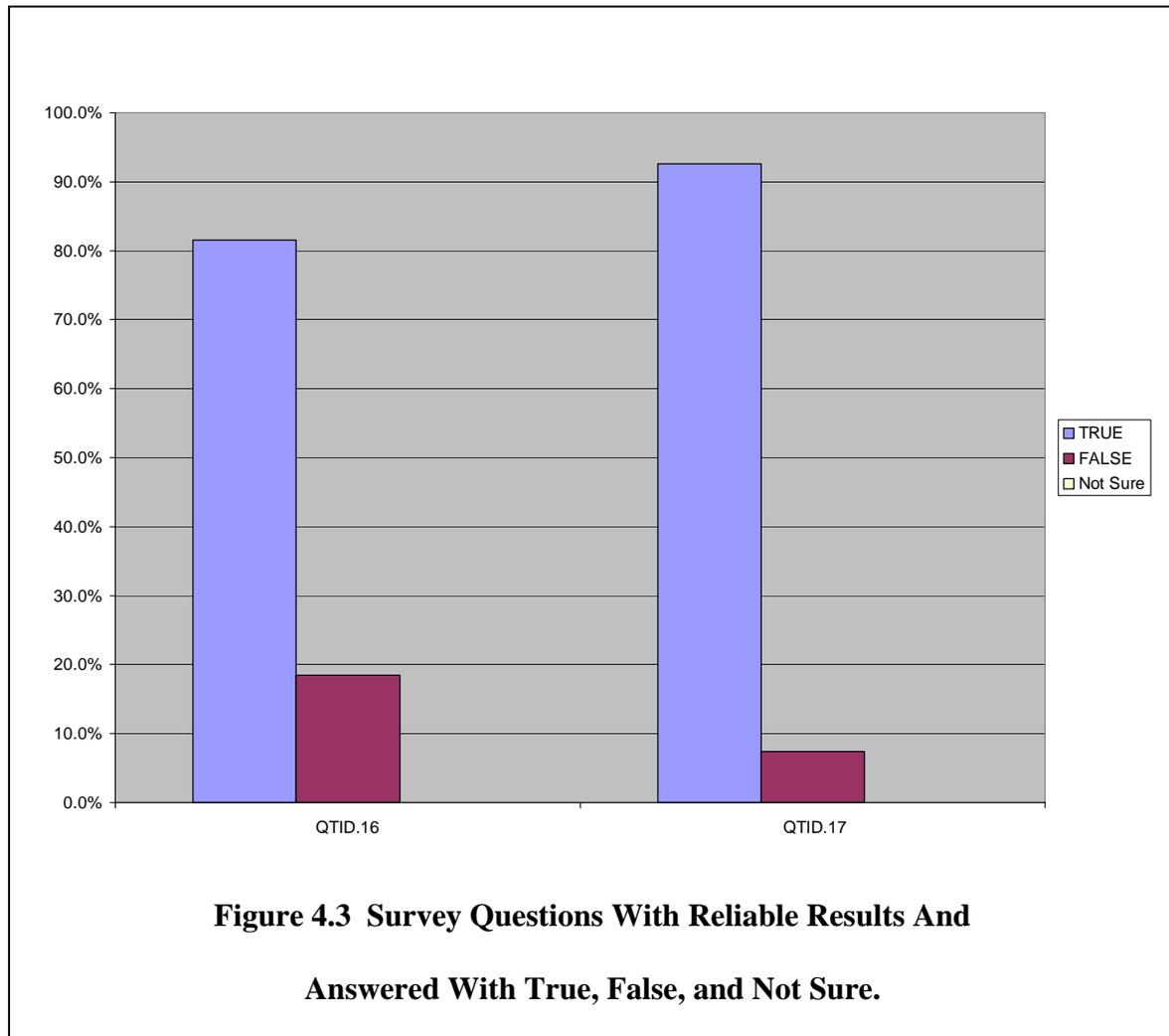


Figure 4.3 represented questions eliciting true or false responses. In situations where participants might have otherwise felt uncomfortable stating true or false, an option was provided for them to have stated that they were not sure. An interesting point was that a majority of participants stated they had used Prioritized Left-to-Right Multi-Column Sort in other applications; note QTID.16. Also interesting was the number of positive responses to the question posed about use of the Standard (single-column) Sort; this methodology is prevalent. The exact questions posed were as follows:

QTID.16: I have used the Prioritized Left-to-Right Multi-Column Sort in other applications?

QTID.17: I have used the Standard (single-column) Sort in other applications?

RELIABILITY OF EXERCISES USED IN SURVEY

Results from the exercises were statistically tested for reliability with the “*t* Method with Same-Subject Groups.” The following three areas were statistically tested: (1) the comparison of the percentage of correct responses between the sorting methods, (2) the comparison of the percentage of incorrect responses between the sorting methods, and (3) the comparison of the total time taken to read and answer exercise questions between each type of sorting method.

For the comparison of the percentage of correct responses between the sorting methods, the resulting *t* value was 0.952272127. The respective tabled *t* value for this statistical treatment was 1.99. To be considered reliable, which it was not, the calculated value needed to be equal to or greater than the tabled *t* value. For further information, please refer to the appendix.

For the comparison of the percentage of incorrect responses between the sorting methods, the resulting t value was 0.196321457. The respective tabled t value for this statistical treatment was 1.99. To be considered reliable, which it was not, the calculated value needed to be equal to or greater than the tabled t value. For further information, please refer to the appendix.

For the comparison of the total time taken to read and answer exercise questions between each type of sorting method, the resulting t value was 0.620754114. The respective tabled t value for this statistical treatment was 1.99. To be considered reliable, which it was not, the calculated value needed to be equal to or greater than the tabled t value. For further information, please refer to the appendix.

In summary, the test results from these three sets of groups indicated that, overall, the data were not reliable. In each case, the calculated t was not equal to or greater than the tabled t value. The appendix contains additional information for each in tables titled as follows: % Correct Answers: Standard Vs. Multi-Column (No Outliers), % Incorrect Answers: Standard Vs. Multi-Column (No Outliers), and Time To Do Exercises: Standard Vs. Multi-Column (No Outliers).

ANSWERS TO QUESTIONS DISCUSSED IN CHAPTER 1

Question 1: Are users more easily able to facilitate a multi-column data array sort with the experimental model as compared to a contemporary popular method for facilitating a sort?

Answer 1: The exercise results do not show conclusive evidence that users more easily facilitated a multi-column data array sort with the experimental

model when this model was compared to a contemporary popular method for facilitating a sort.

Question 2: Does use of the experimental model demonstrate a reduced line-by-line search time for specific records of information?

Answer 2: The exercise results do not show conclusive evidence that use of the experimental model yielded a reduced line-by-line search time for specific records of information when this model was compared to a contemporary popular sorting method. However, user response to survey questions indicated a trend toward a more rapid search for records when using the experimental model.

Question 3: Does use of the experimental model demonstrate a higher level of accuracy for selecting desired records from a listing when this model is compared to a contemporary popular method for facilitating a sort?

Answer 3: The exercise results do not produce conclusive evidence that use of the experimental model showed a higher level of accuracy for selecting desired records from a listing when this model was compared to a contemporary popular sorting method. However, user response to survey questions indicated a trend towards greater accuracy at finding records when using the experimental model.

Question 4: Does use of the experimental model provide the user with a more pleasant and desirable experience when this model is compared to a contemporary popular method for facilitating a sort?

Answer 4: It was not possible to measure whether a user had a more pleasant or desirable experience using PSDA. However, user feedback from the survey suggests that users do like to be able to sort on more than one column. Refer to survey questions QTID.7 and QTID.13.

EVALUATIONS OF HYPOTHESES DISCUSSED IN CHAPTER 1

Hypothesis 1: The user more easily facilitates a sort with the experimental model when this model is compared to a contemporary popular method for facilitating a sort.

Evaluation 1: The hypothesis which stated that the user would more easily facilitate a multi-column data array sort with the experimental model when this model was compared to a contemporary popular method for facilitating a sort was not substantiated.

Hypothesis 2: Use of the experimental model shows a reduced line-by-line search time for specific records of information when this model is compared to a contemporary popular method for facilitating a sort.

Evaluation 2: The hypothesis stating that use of the experimental model would show a reduced line-by-line search time for specific records of information when this model was compared to a contemporary popular method for facilitating a sort was not substantiated.

Hypothesis 3: Use of the experimental model shows a higher level of accuracy for selecting desired records from a listing including non-desired records when this model is compared to a contemporary popular method for facilitating a sort.

Evaluation 3: The hypothesis stating that use of the experimental model would show a higher level of accuracy for selecting desired records from a listing including non-desired records when this model was compared to a contemporary popular method for facilitating a sort was not substantiated.

Hypothesis 4: Use of the experimental model provides the user with a more pleasant and desirable experience when this model is compared to a contemporary popular method for facilitating a sort.

Evaluation 4: The hypothesis stating that use of the experimental model would provide the user with a more pleasant and desirable experience when this model was compared to a contemporary popular method for facilitating a sort had some basis for merit; however, it was not conclusive.

SUMMARY

Results of the sorting tasks and survey assessment indicated that despite some concerns with ease of use, the experimental method might have provided users a more pleasant and desirable tabular sorting experience than the standard-sort. Whether or not the experimental method helped facilitate a multi-column data array sort, reduced line-by-line search time, increased accuracy for finding desired records was not substantiated.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

CONCLUSIONS

Statistical treatment of the web survey did not support conclusive measured evidence to reinforce the feature detailing *prioritized selection of database arrays issued to columnar displays for the purpose of intact transposition of data* (PSDA). Reviewing the survey results in reference to the initial assertions, there was no evidence showing that users of varied levels of computer expertise and background were able to facilitate database sorts more easily, reduce line-by-line search time for specific records, increase accuracy for selecting or choosing from desired records, or have a more pleasant and desirable experience. Also not determined through the web survey was whether the PSDA design was able to improve sorting efficiency when compared to the standard-sort method.

Although the web-survey results did not provide conclusive measured evidence supporting PSDA, four years of observing those using it as a feature within a software application indicates that it provides utility. When comparing PSDA as used within the four-year-old examination software application to the example provided within the web survey, it appears that there is a significant difference. Listings in the old program are rarely displayed in less than four to six columns while the web-survey only accommodated three column listings. Also, data within the same columns in the examination software program repeat considerably more than they do in the web-survey.

From this observation, it is plausible to conclude that this study was not properly designed and therefore was unable to show the significance of PSDA. For proper treatment, more columns were needed as well as a additional non-unique data among the records.

RECOMMENDATIONS

After reviewing the purpose of the study and literature, considering the procedures taken to collect data, and analyzing the results, there are several important recommendations for further research in this topic area. One recommendation is to create a two-part study where a group of participants takes the survey twice with a several week interval between surveys. Another is to use a variety of comparison tools; namely, one that has additional columns and another that is menu based. And finally, study more carefully the concept of the “null response” which was discussed in reference to user satisfaction. These ideas should assist in the further development of PSDA.

A two-part study tests a person’s ability to recall how functionality works. I recommend that two-part studies be conducted wherein the same participants return to use PSDA in comparison to other methods to determine how well they remember the different routines. In the first meeting, the participants should receive familiarization training to learn about the methods involved. In the second meeting, the participants should receive no familiarization to see how well they can recall how to perform the necessary actions to get the desired information. An interesting variation would be to ensure that among the participants are those with characteristics mentioned in Mayhew’s description of likely candidates for direct manipulation interfaces (1992) such as those who rarely use computers, are not motivated, or cannot type well. If Mayhew’s (1992)

descriptions and conclusions were correct, PSDA should work well with these types of users. The two-part study would test the level of memorization required for use of PSDA and if this methodology could be reused primarily with recall.

A variety of sorting tools such as menu accessed sorting and sort by trailing, as discussed previously, need to be compared against PSDA. Results from the standard-sort tool should not be used as the sole source for basing a conclusion on PSDA. In brief, I do not believe that the email paradigm used for the exercises in the survey process was strong enough to make the case for the multi-column array sort I had proposed. A comparison between PSDA and another sorting tool, each with many more columns could better show the organizing power found within the PSDA sorting capability. That is, follow more closely the telephone example and have a column for last name, another for first name, then address, and telephone number. To show the power of PSDA functionality, the survey should include numerous instances of repetition such as same last names and first names. As an example, if there were fifty cases where “Smith” and “John” appeared as the last name then first name respectively, participants would become aware of how much easier it is to find “Smith,” “John,” “1010 Anywhere Street, ” and “801-555-1234,” with a multi-column sort as opposed to the single-column sort. Another sorting feature to compare against is that which is found through a menu-based system; navigating through a menu-based system usually results in accessing a dialog box providing multi-column sorting with three layers of sorting priority. A difficulty with menu-based systems is remembering the proper navigation path to a particular feature. I believe a challenge or a comparison between PSDA and either of these tools would show PSDA to be a more user-friendly methodology.

As a note, I believe it was helpful to conduct exercises comparing methodologies prior to providing a questionnaire. The exercises were important as part of the survey because they appeared to provide context for participants to discuss their understanding of the different sorting methodologies. I would recommend that exercises be used again prior to having participants fill out a questionnaire.

Finally, the topic of “null response” or no response should be explored further. This subject was discussed several times in chapters two and three. The premise for this concept is that a designer should aspire to develop a tool that does not frustrate an operator; however, the operator does not necessarily have to be happy. The middle ground for neither frustrated nor happy is the void associated with the term “null.” The null response results from a condition, state of mind or emotion that may be experienced by users when working with a software application, or any tool for that matter. Within this state of being, the user feels neither happiness nor frustration towards a tool when the tool is used as a means of accomplishing a task. As an example, as I have written this research paper I have used a word processor created by Microsoft Corporation known as *MS Word*. Analyzing my experience, I realize I have rarely focused on the use of the word processor but rather on writing and editing this thesis. The application has met my needs and expectations; I have taken it for granted. Although I have felt both frustration and enthusiasm, these were not directed towards the application but rather my objective of completing the paper. As I perceive it, the power which results from the null response condition is that a tool does not need to be built to continually be in the spotlight and make the user happy. The null response condition allows the tool to be transparent as it

consistently meets users' needs. The importance of further exploration of the “null response” is that it may be a breakthrough in the way designs should be considered.

In summary, this research conducted to explore PSDA has unlocked the door for further consideration and development of this methodology for sorting and brought out a new concept (null response). Additional studies need to be made to consider more fully the effectiveness and usefulness of each concept. The recommendations I have discussed provide a brief overview of starting points for further research.

REFERENCES

- Astrachan, O. (2003). *Bubble Sort: An Archaeological Algorithmic Analysis*. Retrieved May 28, 2005, from <http://portal.acm.org/citation.cfm?id=611918> and on May 30, 2005 from <http://www.cs.duke.edu/~ola/papers/bubble.pdf>.
- Bailey, R. W. (1996). *Human Performance Engineering: Designing High Quality, Professional User Interfaces for Computer Products, Applications, and Systems* (3rd ed.). Upper Saddle River, New Jersey: Prentice-Hall PTR.
- Bitpipe. IT Information: White Papers, Product Literature, Webcasts and Case Studies. Retrieved May 12, 2005 from <http://www.bitpipe.com/tlist/Customer-Satisfaction.html>.
- ComputerUser.com: Technology Trends and Tools (Definition engine.). Retrieved May 12 & 28, 2005, from <http://www.computeruser.com/resources/dictionary/>.
- Drier, M. (2004, September). eLearning-Reviews. Retrieved July 7, 2005 from <http://www.elearning-reviews.org/topics/human-computer-interaction/usability/1993-nielsen-usability-engineering/>.
- Franklin, B. & Nitecki, D. (1999, April 6). *ARL New Measures, User Satisfaction White Paper*. Retrieved May 12, 2005 from <http://www.arl.org/libqual/geninfo/usersatisfaction.pdf>.
- Freeseach (Definition engine.) Retrieved May 12, 2005, from <http://www.freeseach.co.uk/dictionary/spreadsheets>.

- Galitz, W. O. (1997). *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. New York: John Wiley & Sons, Inc.
- Gove, P. B. et al.(Ed.). (1993). *Webster's Third New International Dictionary of the English Language Unabridged*. Springfield, MA: Merriam-Webster Inc., Publishers.
- Hyperdictionary.com (Definition engine.). Retrieved May 12, 2005, from <http://www.hyperdictionary.com/computing/usability>.
- InvestorDirectory.com (Definition engine.). Retrieved May 12, 2005, from <http://www.investordictionary.com/definition/Spreadsheet.aspx>
- Mackie, P., & Mackie, P. H. (1984). *Spreadsheet Dilemma*. Beaverton, OR: Dilithium Press.
- Martin, A. & Eastman, D. (1996). *The User Interface Design Book*. New York: John Wiley & Sons, Inc.
- Mayhew, D. J. (1992). *Principles and Guidelines in Software User Interface Design*. Englewood Cliffs, New Jersey: Prentice Hall.
- Microsoft® Office Excel 2003 (2003). Microsoft Corporation.
- Nielsen, J. (2005). *Ten Usability Heuristics*. Retrieved Jun 20, 2005, from http://www.useit.com/papers/heuristic/heuristic_list.html.
- Norman, D. A. (2002). *The Design of Everyday Things*. New York: Basic Book.
- Thimbleby, H. (2001). The Computer Science of Everyday Things. *IEEE*, 0-7695-0969-X/01, 1-12.
- Treu, S. (1994). *User Interface Evaluation: A Structured Approach*. New York: Plenum Press.

Welie, M. van & Traetteberg, H. (2000). *Interaction Patterns in User Interfaces*.

Retrieved May 12, 2005, from <http://www.idi.ntnu.no/~hal/publications/design-patterns/PLoP2k-Welie.pdf>.

APPENDICES

APPENDIX A: DEFINITIONS AND TERMS

Affordance: “Involves the perceived and actual properties of an object that suggest how the object is to be used. Example: ‘Now, how does this object work? Oh, I get it ...’” (Welie and Trattenberg, 2000, p. 2).

Cognition: [Recall cognition.] The extent to which a person responds to a previously learned term or process.

Conceptual models: “A good conceptual model is one in which the user's understanding of how something works corresponds to the way it actually works. This way the user can confidently predict the effects of his actions. Example: ‘To perform the task, I provide the necessary information and gave this command ... and it seems to work as I expected it to...’” (Welie and Trattenberg, 2000, p. 2).

Constraint: “A constraint reduces the number of ways to perform a task and the amount of knowledge necessary to perform a task, making it easier to figure out. Example: ‘Oh no, what do I have to enter here? Ok, I just have these choices...’” (Welie and Trattenberg, 2000, p. 2).

Database: “A large collection of data organized for rapid search and retrieval. A program that manages data, and can be used to store, retrieve, and sort information. Database examples are Lotus Approach, Microsoft Access, Filemaker, and dBASE” (ComputerUser.com, 2005, search “Database”).

Design Principle: “A fundamental truth or belief about a design; a rule explaining the expected behavior or appearance of a design object; a rule of action or conduct of a design object; a design goal” (Treu, 1994, p. 9).

Direct Manipulation Interface: “Generally, this is an interface in which users *perform actions directly on visible objects*. This is in contrast to interfaces in which users *indirectly specify actions, parameters, and objects* through language (for example, command language or menu interfaces)” (Mayhew, 1992, p. 289).

Feedback: “An indicator to the user that a task is being done and that the task is being done correctly. A feedback example: ‘Great it worked!’” (Welie and Trattenberg, 2000, p. 2).

Flexibility: “Users may change their mind and each user may do things differently. Example: ‘Now that I think about it, that parameter should have been ...’ A flexibility example: ‘Cancel it, I want to change the order.’” (Welie and Trattenberg, 2000, p. 2).

Human-Computer Interface: “The physical surface and facilities, between human user and computer, providing the medium through which they can connect and interact; the physical (visual, audio, tactile) means, methods, and patterns that support human-computer interaction” (Treu, 1994, p. 9).

Human-Computer Interaction: “The combination of physical, logical, conceptual, and language-based actions between a human user and a computer, toward achieving some purpose” (Treu, 1994, p. 9).

Icon: “In a graphical user interface, a small picture on the screen which represents something. Files and programs have icons, and open when the user clicks twice on the icon. There is an icon to show which program is currently running. Moving and copying files is done by dragging their

icons to the desired location. A file is deleted by moving its icon to the picture of a trash can” (ComputerUser.com, 2005, search “Icon”).

Menu Item: “Menu items are the controls contained in menus. Types of menus are menu bars, related pull-downs, and pop-ups. Types of choices include action, dialog (...) and cascading (>>)” (Bailey, 2001, p. 352).

Natural mapping: “A natural mapping creates a clear relationship between what the user wants to do and the mechanism for doing it. Example: ‘To perform my task, I need to select this option, enter that information, and then press this button...’” (Welie and Trattenberg, 2000, p. 2).

Pushbuttons: “A control containing text or graphics (or both) that represents an action or dialog choice. The control is activated when ‘pressed’ by users (i.e., a single mouse click or finger touch)” (Bailey, 2001, p. 354).

Safety: “The user needs to be protected against unintended actions or mistakes. Example: ‘Oops! I made a mistake and here is how I correct it. Now I understand and I’ll try again.’” (Welie and Trattenberg, 2000, p. 2).

Sort: “To organize information in the desired order. Some computer programs, for example, database and word processing programs, have sort functions which can organize items alphabetically, numerically, or by date, in ascending or descending order” (ComputerUser.com, 2005, search “Sort”).

Spreadsheet: “A computerized electronic worksheet” (Mackie, 1984, p. 1); “a computer program, used especially in business, which allows you to do financial calculations and plans” (Freeseearch, 2005, search “Spreadsheet”). “A

spreadsheet is a rectangular table (or grid) of information, often financial information. (It is, therefore, a kind of matrix.) The word came from ‘spread’ in its sense of a newspaper or magazine item (text and/or graphics) that covers two facing pages, extending across the centerfold and treating the two pages as one large one. The compound word ‘spreadsheet’ came to mean the format used to present bookkeeping ledgers—with columns for categories of expenditures across the top, invoices listed down the left margin, and the amount of each payment in the cell where its row and column intersect, for example—which were traditionally a ‘spread’ across facing pages of a bound ledger (= book for keeping accounting records) or on oversized sheets of paper ruled into rows and columns in that format and approximately twice as wide as ordinary paper” (InvestorDirectory.com, 2005, search “Spreadsheet”). Another definition of a spreadsheet is, “A table which displays numbers in rows and columns, used for accounting, budgeting, financial analysis, scientific applications, and other work with figures. Originally, paper spreadsheets were used; the computerized versions have the advantage of being able to perform automatic calculations on changing data. Each location in a spreadsheet is called a cell, and each cell has a number, such as A3, B4, etc. The value of a cell may be calculated from a formula involving other cells; for example, C5 may be the sum of A5 and B5. If the data entered in A5 changes, then C5 would be automatically recalculated by the program. This feature makes it possible to use a spreadsheet to project the results of

a change in one or more factors. References between cells may be designated as absolute or relative; an absolute reference refers to a specific cell, and a relative reference describes a cell in its relation to the current cell (as two rows above, two columns to the left, etc.). The data within a cell may be a label, a number, or a formula, and may be copied to other cells. There are two- and three-dimensional spreadsheets. The first commercially available spreadsheet program was VisiCalc for Apple II; now there are spreadsheet programs from Microsoft, Borland, Lotus, and many other companies. Microsoft Excel and Lotus 1-2-3 are popular examples” (ComputerUser.com, 2005, search “Spreadsheet”).

Table: “A presentation of information organized in rows and columns. Spreadsheet and database programs provide the best tools for working with tables. A database can show and work with intricate relationships between and among fields and databases. In programming it is the same as an array” (ComputerUser.com, 2005, search “Table”).

Tabular: “Of, relating to, or arranged in a table. Set up in rows and columns. Used in setting up a table. Derived from or computed by means of a table” (Gove, 1993, p. 2325).

Transposition: “An act, process, or instance of transposing or being transposed. Transfer or removal from one place or time to another. A change or interchange in order or place especially of letters or words. Rearrangement” (Gove, 1993, p. 2431). For the purposes of this paper, transposition will refer to

the exchange of one column of data with another column of data without changing the integrity of the displayed record information.

Usability: “The effectiveness, efficiency, and satisfaction with which users can achieve tasks in a particular environment of a product. High usability means a system is: easy to learn and remember; efficient, visually pleasing and fun to use; and quick to recover from errors” (Hyperdictionary, 2005, search “Usability”).

User Dissatisfaction: Dissatisfaction is “the quality or state of being dissatisfied, unsatisfied, or discontented: uneasiness, disturbance, or distress resulting from a lack of gratification; displeasure” (Gove, 1993, p. 656.). Defining *dissatisfy*: “to fail to satisfy, to fail to provide with something desired, expected or hoped for or to the extent desired, expected or hoped for: frustrate wishes or expectations of” (Gove, 1993, p. 656.) As a note, “satisfaction is often a short-term measure” (Franklin & Nitecki, 1999, p. 1). For the purposes of this paper, user dissatisfaction is then taken to mean a user’s *dissatisfied* feelings about a product or service.

User Satisfaction: “Complete fulfillment of a need or want: attainment of a desired end, contentment, pleasure” (Gove, 1993, p. 2017). Another term for user satisfaction is called customer satisfaction, which is described as “a qualitative measure of performance as defined by customers, which meet their basic requirements and standards” (Bitpipe). “Satisfaction is often a short-term measure” (Franklin & Nitecki, 1999, p. 1).

Visibility: “Capability of being readily noticed” (Gove, 1993, p. 551). “Gives the user the ability to figure out how to use something just by looking at it. Example: ‘Hmm, I think this feature might do it...’” (Welie and Trattenberg, 2000, p. 2).

Visual Cue: By removing the adjective *visual*, *cue* is defined as “(1) a signal to begin an action: a stimulus; (2) a hint, intimation, or suggestion as to what course of action to take or when to take it; (3) an item or feature acting as an indication of the nature of the object or situation perceived; and (4) a prompt” (Gove, 1993, p. 551). For the purposes of this paper, a *visual cue* then is considered to be (1) a *visible* signal to begin an action: a *visible* stimulus; (2) a *visible* hint, *visible* intimation, or *visible* suggestion as to what course of action to take or when to take it; (3) a *visible* item or feature acting as an indication of the nature of the object or situation perceived; and (4) a *visible* prompt.

Worksheet: “A working paper used by an accountant to assemble figures for financial statements of a business: *specifically*: a sheet with a sufficient number of columns to provide for entering the trial balance, adjusting entries, profit and loss, and balance sheet items” (Gove, 1993, p. 2635). A worksheet is also known as “a columnar pad, accountant’s pad, ledger, and ledger sheet” (Mackie, 1984, p. 2). “A table which displays numbers in rows and columns, used for accounting, budgeting, financial analysis, scientific applications, and other work with figures. Same as spreadsheet” (ComputerUser.com, 2005, search “Worksheet”).

APPENDIX B: THE TABLE DATA

Table: ArrayItems

Columns

Name	Type	Size
ID	Long Integer	4
Name	Text	255
Subject	Text	255
DateTime	Text	50
ScrambleSort	Double	8

Table: Coupon

Columns

Name	Type	Size
ID	Long Integer	4
Active	Yes/No	1
Item	Text	50
Origin	Text	50
NumberOriginalCount	Long Integer	4
NumberRemaining	Long Integer	4
ValidLocations	Text	255

Table: CouponIntervieweeInfo

Columns

Name	Type	Size
ID	Long Integer	4
MailingAddress	Memo	-
EmailAddress	Text	255
CouponID	Long Integer	4
ItemSent	Yes/No	1
EntryDateTime	Text	50
ItemSentDateTime	Text	50

Table: Login

Columns

Name	Type	Size
ID	Long Integer	4
loginID	Text	50
loginPassword	Text	50
loginTime	Text	50

Table: LoginHistory

Columns

Name	Type	Size
ID	Long Integer	4
loginID	Text	50
loginTime	Text	50

Table: ExerciseResponse

Columns

Name	Type	Size
ID	Long Integer	4
IntervieweeID	Long Integer	4
IntervieweeCode	Text	50
ResponseType	Text	50
CorrectAnswer	Text	50
IntervieweeResponse	Text	50
TimeStart	Text	50
TimeEnd	Text	50
SecondsElapsed	Text	50
CorrectAnswerNumberOf	Long Integer	4
AnswerIntervieweeVsCorrectSame	Long Integer	4
AnswerIntervieweeVsCorrectDifferent	Long Integer	4
AnswerCorrectVsIntervieweeSame	Long Integer	4
AnswerCorrectVsIntervieweeDifferent	Long Integer	4

Table: Interviewee

Columns

PercentIncorrectBothWays	Long Integer	4
TotalTimeStandardWayFindName	Long Integer	4
TotalTimeStandardWayFindSubject	Long Integer	4
TotalTimeStandardWayFindDate	Long Integer	4
TotalTimeMultipleWayFindName	Long Integer	4
TotalTimeMultipleWayFindSubject	Long Integer	4
TotalTimeMultipleWayFindDate	Long Integer	4
TotalTimeStandardWay	Long Integer	4
TotalTimeMultipleWay	Long Integer	4
TotalTimeBothWays	Long Integer	4
PercentTimeStandardOverBoth	Long Integer	4
PercentTimeMultipleOverBoth	Long Integer	4
RatioTimeMultipleOverStandard	Long Integer	4
ExerciseDisplayOrder	Memo	-
QuestionnaireDisplayOrder	Memo	-
QuestionnaireTimeStart	Text	50
QuestionnaireTimeEnd	Text	50
QuestionnaireTotalTime	Text	50

Table: QuestionnaireQuestion

Columns

Name	Type	Size
ID	Long Integer	4
Active	Yes/No	1
QuestionTransferableID	Long Integer	4
QuestionPurpose	Text	50
QuestionType	Text	50
TimeOfAppearance	Text	50
Question	Text	255
PossibleResponse1	Text	255
PossibleResponse2	Text	255
PossibleResponse3	Text	255
PossibleResponse4	Text	255
PossibleResponse5	Text	255

Table: Interviewee

Columns

Name	Type	Size
ID	Long Integer	4
UniqueWebCode	Text	50
Name	Text	50
EmailAddress	Text	255
InterviewDate	Date/Time	8
TotalPossibleStandardWayFindName	Long Integer	4
TotalPossibleStandardWayFindSubject	Long Integer	4
TotalPossibleStandardWayFindDate	Long Integer	4
TotalCorrectStandardWayFindName	Long Integer	4
TotalCorrectStandardWayFindSubject	Long Integer	4
TotalCorrectStandardWayFindDate	Long Integer	4
TotalIncorrectStandardWayFindName	Long Integer	4
TotalIncorrectStandardWayFindSubject	Long Integer	4
TotalIncorrectStandardWayFindDate	Long Integer	4
TotalPossibleStandardWay	Long Integer	4
TotalCorrectStandardWay	Long Integer	4
TotalIncorrectStandardWay	Long Integer	4
PercentCorrectStandardWayFindName	Long Integer	4
PercentCorrectStandardWayFindSubject	Long Integer	4
PercentCorrectStandardWayFindDate	Long Integer	4
PercentIncorrectStandardWayFindName	Long Integer	4
PercentIncorrectStandardWayFindSubject	Long Integer	4
PercentIncorrectStandardWayFindDate	Long Integer	4
PercentCorrectStandardWay	Long Integer	4
PercentIncorrectStandardWay	Long Integer	4
TotalPossibleMultipleWayFindName	Long Integer	4
TotalPossibleMultipleWayFindSubject	Long Integer	4
TotalPossibleMultipleWayFindDate	Long Integer	4
TotalCorrectMultipleWayFindName	Long Integer	4
TotalCorrectMultipleWayFindSubject	Long Integer	4
TotalCorrectMultipleWayFindDate	Long Integer	4
TotalIncorrectMultipleWayFindName	Long Integer	4
TotalIncorrectMultipleWayFindSubject	Long Integer	4
TotalIncorrectMultipleWayFindDate	Long Integer	4
TotalPossibleMultipleWay	Long Integer	4
TotalCorrectMultipleWay	Long Integer	4
TotalIncorrectMultipleWay	Long Integer	4
PercentCorrectMultipleWayFindName	Long Integer	4
PercentCorrectMultipleWayFindSubject	Long Integer	4
PercentCorrectMultipleWayFindDate	Long Integer	4
PercentIncorrectMultipleWayFindName	Long Integer	4
PercentIncorrectMultipleWayFindSubject	Long Integer	4
PercentIncorrectMultipleWayFindDate	Long Integer	4
PercentCorrectMultipleWay	Long Integer	4
PercentIncorrectMultipleWay	Long Integer	4
TotalPossibleBothWays	Long Integer	4
TotalCorrectBothWays	Long Integer	4
TotalIncorrectBothWays	Long Integer	4
PercentCorrectBothWays	Long Integer	4

Table: QuestionnaireResponseMemo

Columns

Name	Type	Size
ID	Long Integer	4
IntervieweeID	Long Integer	4
IntervieweeCode	Text	50
QuestionTransferableID	Long Integer	4
Response	Memo	-

Table: QuestionnaireResponseMultipleChoice

Columns

Name	Type	Size
ID	Long Integer	4
IntervieweeID	Long Integer	4
IntervieweeCode	Text	50
QuestionTransferableID	Long Integer	4
Response	Text	10

APPENDIX C: THE LIST OF ITEMS TO SORT

ArrayItems				
ID	Name	Subject	DateTime	ScrambleSort
295	Jobs R Us	jobs: roof cleaner	200303251633	2
186	Mom	met a couple of cute girls	200307142212	3
179	Reese Aren	missed one week of school...	200307180730	12
396	Software Solutions	Resume incomplete	200302071009	14
449	Cisco Jobs	opening in: Sales	200301120933	16
467	Sarah Jeeves	new semester and stuff	200301050903	16
310	Dad	Calculus and failing	200303211708	17
212	Sarah Jeeves	new term and stuff	200307050903	19
81	Nathan Breker	June not interested	200309032043	21
207	Mel Giston	getting ready for Bio class	200307051932	23
394	Soccer Fanatics	Paraguayan soccer player...	200302071254	25
343	Dad	Lots of time in library	200302282249	26
57	Global Concerns	Dozens of penguins dead	200309201101	27
323	Gems	Love is not love until you share..	200303181350	27
445	Mom	wish I had your food	200301122121	29
56	Inquiries Ldating	My name is Zena, #50099	200309211208	31
462	Mel Giston	getting ready for Physics class	200301051932	37
116	Reese Aren	doctor says one more week...	200308180657	38
354	Nathan Breker	Tanya and her roommates	200302221938	39
96	June Jeppson	I realize you are busy..	200308251310	47
257	Dad	passed all my finals	200304271623	47
110	Nathan Breker	June is moving to Miami	200308191513	47
214	Novell Opportunity	opening in: Russian Translat.	200307041036	47
339	Gems	When in doubt, do not shout.	200303030843	50
154	Inquiries Ldating	My name is Bobette, #34448	200308011134	52
305	Software Solutions	Interview scheduled for 29 Mar	200303221254	52
476	Dale Yortber	sister wrecked during break...	200301031832	53
368	Mom	guidance counselor smart	200302182345	54
143	Global Concerns	Penguin population zapped	200308061029	55
95	Nathan Breker	June has been really busy	200308251313	56
94	Nathan Breker	Saw June with another guy	200308272105	61
436	Dad	passed calculus test	200301172212	61
377	Mom	career choice a bummer	200302142139	62
286	Nathan Breker	Park City and Beauty Betty	200303310022	63
75	Global Concerns	Weather cooling for penguins	200309121339	66
159	US Hotline Jobs	jobs: Spanish tutor	200307282321	67
44	YourMagazine.net	Your subscription ends in ...	200309251011	73
228	US Hotline Jobs	jobs: chemistry lab asst.	200307031224	74
164	US Hotline Jobs	jobs: physics tutor	200307271222	77
206	US Hotline Jobs	jobs: web designer	200307061213	81
193	National Instruments	Unable to locate your phone...	200307121121	82
334	Dad	Study group is great	200303042054	84
309	Betty Benter	What kind of food do you like?	200303220102	85
252	Nathan Breker	Met Cindy at fair	200305262129	89
391	Jobs R Us	jobs: janitor	200302072019	90
60	Stats HelpOnline	Question did not make sense.	200309191259	93

ArrayItems				
ID	Name	Subject	DateTime	ScrambleSort
14	Nathan Breker	Christy, welcome home in Moab	200310102239	93
446	Dad	school great	200301121232	97
399	Mom	burnt pancakes	200302041929	98
366	Tanya Lundberg	Living in Chicago	200302191411	100
18	Christy Lentoh	Prep for graduation	200310071133	105
454	MS Employment	career: Documentation	200301081340	109
196	US Hotline Jobs	jobs: thermometer reader	200307120923	110
256	Mom	see you next weekend	200304281623	111
431	Sophy Wendell	See you at school	200301182118	112
500	Dave Franconi	New Years...	200301010600	114
250	Cindy Anderton	Me, from the fair	200305272129	116
342	Nathan Breker	Tanya not returning calls	200303022221	119
423	Dad	why dance class	200301252209	122
62	US Hotline Jobs	jobs: public affairs assistant	200309182352	123
459	Jobs R Us	jobs: floor waxer	200301061223	124
491	Jobs R Us	jobs: clerical typist	200301020924	125
269	Nathan Breker	Betty, welcome home in Moab	200304102239	126
315	Calculus HelpOnline	Question did not make sense.	200303191259	127
267	Nathan Breker	Betty a little bored	200304142222	128
495	Jobs R Us	jobs: farm laborer	200301010633	130
311	Inquiries Ldating	My name is Roset, #54699	200303211208	131
455	Joana Mogberry	lunch at Taco Bell	200301081323	131
306	Soccer Fanatics	South Africa team meets ...	200303221112	131
69	Gymbaub Ind	Application needs address	200309181342	132
92	Reese Aren	writing with other arm... again...	200308281819	133
173	Mom	going to classes in new build...	200307222209	134
458	MS Employment	career: Radio Announcer	200301061245	137
432	MS Employment	career: Network Engineer	200301180744	139
427	Nathan Breker	Sophy got mad at me	200301222309	140
333	Dad	Ran out of money	200303111239	147
137	Dad [office email]	how is mom doing?	200308072009	156
294	MS Employment	career: Chauffer	200303251654	163
408	Soccer Fanatics	Argentina vs Brazil news	200302011158	166
126	June Jeppson	Anybody for Rock Canyon?	200308131812	167
78	Dad [office email]	Ran out of money again	200309111239	168
344	MS Employment	career: Network Technician	200302282238	169
251	MS Employment	career: Software Development	200305262143	172
80	Stats HelpOnline	Looking better on reasoning...	200309041054	174
203	Tandem Newsflash	career: Radio Announcer	200307061245	174
223	Larry Eliazen	what did you get in Chem 110?	200307031802	175
289	Betty Benter	I know a place in Heber	200303271044	175
401	Bob K. Runchleg	hopping on other leg...	200302031456	177
291	Betty Benter	Cool that you help with	200303262258	177
107	Tandem Newsflash	career: LAN supervisor	200308201439	178
49	Nathan Breker	Christy is such a great gal	200309221854	184
384	Nathan Breker	Tanya is great	200302110722	186

ArrayItems				
ID	Name	Subject	DateTime	ScrambleSort
161	Mom	chemistry class kind of hard	200307272328	188
298	Inquiries Ldating	My name is Liz, #45009	200303251432	192
170	Mom	chemistry class cool	200307242138	193
389	Cisco Jobs	opening in: French Translat.	200302072035	194
410	Software Solutions	Accepting new applications	200302011100	194
181	Dad [office email]	passed stats test	200307172212	196
274	Dad	Religion class/seminary	200304041533	198
411	MS Employment	career: Eng/Russ Translat.	200301282341	198
64	Tandem Newsflash	career: Eng/Span Translat.	200309182342	202
302	Betty Benter	Thanks for help in dance class	200303231854	208
27	Reese Aren	started to bench...	200310020808	210
88	Dad [office email]	Lots of time in library office	200308282249	212
494	MS Employment	career: PC technician	200301010652	214
204	US Hotline Jobs	jobs: floor waxer	200307061223	215
255	Bob K. Runchleg	planning for 5K at July 4th...	200304291423	217
442	Software Solutions	Opening #34589, call 555-1234	200301141333	217
166	Dad [office email]	Biology not in future	200307262138	219
236	US Hotline Jobs	jobs: clerical typist	200307020924	220
413	Jobs R Us	jobs: custodial supervisor	200301282326	220
241	Novell Opportunity	opening in: Marketing	200307010631	222
468	Jobs R Us	jobs: English tutor	200301041056	229
360	Tanya Lundberg	What is Chicago like?	200302201900	231
108	US Hotline Jobs	jobs: day care assistant	200308201414	239
127	June Jeppson	June, did yoy pass the quiz?	200308112222	244
2	Bob Beuford	Get great results on LSAT	200310172344	245
149	Stats HelpOnline	Formula needs adjusting...	200308021200	245
17	Nathan Breker	Christy and graduation	200310081133	246
220	US Hotline Jobs	jobs: dance lab asst.	200307040902	247
463	Bob K. Runchleg	full leg cast...	200301051144	252
353	Tanya Lundberg	Tanya, please answer.	200302222340	253
112	Dad [office email]	dating few and far	200308182345	256
73	Reese Aren	lifting almost normally...	200309150909	256
104	Nathan Breker	What is Miami like?	200308201904	263
101	Reese Aren	TV and arm on huge pillow...	200308221232	264
224	Mel Jarboe	passed that class last semstr	200307031756	265
405	Dad	Aeronautical eng. & cslr	200302011858	265
103	June Jeppson	Hope you get better.	200308212015	266
211	US Hotline Jobs	jobs: model	200307050922	266
79	Dad [office email]	Study group is is not happening	200309042054	267
153	Soccer Fanatics	Argentina vs Brazil news	200308011158	272
273	Betty Benter	Prep for graduation	200304071133	272
225	Reese Aren	slipped on ice...	200307031313	274
308	Nathan Breker	Did I tell you about Betty?	200303220112	275
296	Cisco Jobs	opening in: Publications	200303251632	278
416	Mom	physics class kind of hard	200301272328	280
371	Bob K. Runchleg	doctor says two more weeks...	200302180657	280

ArrayItems				
ID	Name	Subject	DateTime	ScrambleSort
9	Christy Lentoh	Hope you get over the sniffles.	200310151335	281
176	Margie Largieh	See you at class	200307182118	284
230	Tandem Newsflash	career: Customer Support	200307021250	287
293	Betty Benter	Saw you cheering at the game	200303252256	296
483	Jobs R Us	jobs: chemistry lab asst.	200301031224	297
376	Nathan Breker	We went to the dance	200302150945	297
133	Dad [office email]	Rick and welcome home	200308081854	298
392	Dad	how is mom?	200302072009	299
437	Sophy Wendell	Hey Sophy	200301161212	302
415	Nathan Breker	Sad news from Sophy	200301280934	303
122	Mom	career choice looking better	200308142139	306
202	US Hotline Jobs	jobs: shuttle driver	200307081312	306
421	Dad	Biology in future	200301262138	309
464	MS Employment	career: Employee Relations	200301050949	310
443	Dad	five classes, 15 credits	200301141233	312
388	Dad	Rick and mission	200302081854	312
452	Software Solutions	Inquiry for opening #34589	200301120833	313
42	Nathan Breker	Of Christy, graduation, etc.	200309251611	315
86	June Jeppson	We need to talk, please call.	200309022343	319
266	Nathan Breker	Able to go, Betty sick	200304151035	322
314	Dad	Bean museum	200303192259	324
72	National Instruments	Please call at earliest conve...	200309160732	325
407	Bob K. Runchleg	cast taken off...	200302011234	327
380	Tanya Lundberg	There is a dance...	200302140945	329
362	MS Employment	career: LAN supervisor	200302201439	333
231	US Hotline Jobs	jobs: newspaper editor	200307021245	335
237	John Fitzgerald	Birthday was great	200307020900	337
258	Nathan Breker	Great grills, but none for me	200304180454	337
447	Global Concerns	Skinny penguins freezing	200301121132	341
115	Nathan Breker	June loves airplanes	200308180822	344
412	Dad	Meeting with counselor	200301282328	345
450	Bob K. Runchleg	out of traction...	200301120928	347
498	Jobs R Us	jobs: construction hand	200301010602	351
87	Nathan Breker	June not returning calls	200309022221	355
66	Soccer Fanatics	Leeds team to be undefeted	200309181356	355
216	Reese Aren	injured elbow...	200307041019	356
395	Soccer Fanatics	Arabian soccer player injured...	200302071119	359
124	Nathan Breker	June and I are	200308141812	363
90	US Hotline Jobs	jobs: electronics assembler	200308282232	369
402	Soccer Fanatics	Argentina beats Paraguay	200302031056	379
434	Bob K. Runchleg	missed two weeks of school...	200301180730	380
268	Betty Benter	You seem a little sick	200304141722	381
40	US Hotline Jobs	jobs: roof cleaner	200309251633	382
393	Mom	dancing with the ward	200302071854	383
277	Jobs R Us	jobs: math lab assistant	200304022358	384
105	June Jeppson	What is Miami like?	200308201900	387

ArrayItems				
ID	Name	Subject	DateTime	ScrambleSort
77	Inquiries Ldating	My name is Lana, #02668	200309111249	387
397	Dad	still have a few bucks left	200302061929	389
400	Software Solutions	Application received	200302041129	391
265	Bob K. Runchleg	running these days...	200304151219	393
418	Jobs R Us	jobs: parking lot attendant	200301271231	395
313	Dad	Church calling	200303201001	396
297	Nathan Breker	Of Betty, graduation, etc.	200303251611	397
188	Dad [office email]	four classes, 11 credits	200307141233	402
262	Betty Benter	We need to talk when you ...	200304160149	403
242	Novell Opportunity	opening in: Software Dev.	200307010630	408
243	US Hotline Jobs	jobs: construction hand	200307010602	410
195	Reese Aren	out of traction...	200307120928	411
386	Mom	dancing class fun but hard	200302091832	412
210	Novell Opportunity	opening in: Network Testing	200307050927	414
317	Jobs R Us	jobs: public affairs assistant	200303182352	415
364	Nathan Breker	Love of my life and...	200302201411	415
58	Dad [office email]	New church calling	200309201001	416
45	Christy Lentoh	Great you are graduating	200309241611	417
201	US Hotline Jobs	jobs: meal preparer	200307081322	420
372	Tanya Lundberg	Thinking about a car.	200302171722	420
83	Gymbaub Ind	Placement position opening	200309030940	422
52	Gems	Stay on course	200309221102	424
4	John Benedict	US History Buffs sought	200310171404	424
332	Inquiries Ldating	My name is Lori, #02148	200303111249	425
490	Cisco Jobs	opening in: Documentation	200301020929	426
279	MS Employment	career: Newscaster	200304022346	431
278	Nathan Breker	Movies with awesome girl	200304022351	440
261	Betty Benter	I wish you the best.	200304162354	442
263	Nathan Breker	Engagement ring but not mine	200304152149	443
280	Jobs R Us	jobs: bathroom designer	200304022324	443
441	Mom	met a cute girl	200301142212	450
292	Nathan Breker	Betty leads cheers	200303252356	459
15	Mom	preparing for Johns wedding	200310101533	461
456	Jobs R Us	jobs: meal preparer	200301081322	461
409	Inquiries Ldating	My name is Geneen, #32148	200302011134	461
222	Fred Crunchberry	got sick again	200307031822	464
318	Betty Benter	The dance was fun.	200303182350	464
480	Bob K. Runchleg	slipped on ice...	200301031313	465
178	US Hotline Jobs	jobs: grounds keeper	200307180737	466
19	Dad [office email]	More classes to pass/seminary	200310041533	466
322	Software Solutions	Interview request	200303181352	469
141	Software Solutions	Resume needs attachment	200308071009	470
50	Software Solutions	Interview scheduled for 29 Sep	200309221254	470
374	Dad	Physics not going well	200302152139	472
109	Nathan Breker	Love of my life and Miami	200308201411	473
249	Cindy Anderton	So, what is Idaho like?	200305302129	474

ArrayItems				
ID	Name	Subject	DateTime	ScrambleSort
84	Gems	Be a good neighbor	200309030843	474
227	US Hotline Jobs	jobs: rifle cleaner	200307031225	479
331	Mom	girls come and go	200303121239	488
327	National Instruments	Have not heard from you...	200303160732	489
325	Soccer Fanatics	Lisbon soccer team wins...	200303181317	489
444	Gymbaub Ind	Thx interest in gym equipm	200301141212	490
284	Jobs R Us	jobs: baby cotton swab tester	200303310056	491
358	Tanya Lundberg	Hope you feel better.	200302212015	492
348	Tanya Lundberg	Hope you feel better.	200302272111	493
175	Dad [office email]	Spanish term paper	200307182222	497
304	Nathan Breker	Betty is such a great gal	200303221854	498
146	Reese Aren	writing with other arm...	200308031456	499
71	Dad [office email]	Parking ticket paid late	200309161732	507
54	Christy Lentoh	What kind of food do you like?	200309220102	509
184	Stats HelpOnline	Functions written wrong, the...	200307151312	509
26	Christy Lentoh	A new movie came out	200310021251	510
51	Soccer Fanatics	Nigeria to host games for...	200309221112	515
163	US Hotline Jobs	jobs: parking lot attendant	200307271231	515
300	Betty Benter	Great you are graduating	200303241611	515
117	June Jeppson	Thinking about a motorcycle.	200308171722	519
91	US Hotline Jobs	jobs: meal planner	200308282229	520
485	MS Employment	career: Customer Support	200301021250	520
357	Nathan Breker	Tanya unable to go to...	200302212019	521
76	Mom	girls come and go and go...	200309121239	523
106	US Hotline Jobs	jobs: yard stick reader	200308201445	526
320	Jobs R Us	jobs: Tagalog tutor	200303182317	528
478	Larry Eliazen	what did you get in Stats 110?	200301031802	531
351	Tanya Lundberg	I realize you are busy..	200302251310	533
345	Jobs R Us	jobs: electronics assembler	200302282232	534
114	June Jeppson	Miami sounds great.	200308181513	534
475	Jobs R Us	jobs: dance lab asst.	200301040902	536
429	Nathan Breker	Sophy is great	200301202118	539
321	Soccer Fanatics	Denmark team traveling to...	200303181356	539
53	Nathan Breker	Did I tell you about Christy?	200309220112	545
337	Tanya Lundberg	Please call.	200303032040	547
65	US Hotline Jobs	jobs: Tagalog tutor	200309182317	547
183	Dad [office email]	doing well in chemistry	200307152121	550
481	MS Employment	career: Document. Research	200301031247	550
48	PowerMan	Bicep pumping tips	200309231354	551
238	Tandem Newsflash	career: Custodial	200307010653	552
197	Software Solutions	Inquiry for opening #66709	200307120833	552
433	Jobs R Us	jobs: grounds keeper	200301180737	552
387	Nathan Breker	started dating Tanya	200302090632	557
148	Dad [office email]	busy with school again	200308021300	559
244	Dave Franconi	July 4th...	200307010601	560
356	Bob K. Runchleg	TV and leg on coffee table...	200302221232	562

ArrayItems				
ID	Name	Subject	DateTime	ScrambleSort
138	Mom	stake dance	200308071854	563
240	US Hotline Jobs	jobs: farm laborer	200307010633	566
111	June Jeppson	Living in Miami	200308191411	576
20	Nathan Breker	Saturday date with Christy	200310041229	578
435	Jobs R Us	jobs: stadium cleaner	200301180721	584
38	Christy Lentoh	Saw you cheering at the game	200309252256	585
385	Tanya Lundberg	How are classes going?	200302100632	589
128	Dad [office email]	More classes to pass	200308111908	593
190	Mom	wish I had your food, as always	200307122121	594
264	Betty Benter	Hope you get over the sniffles.	200304151335	597
102	Nathan Breker	June unable to go to...	200308212019	598
330	Global Concerns	Weather warming for penguins	200303121339	599
8	Nathan Breker	Engagement but not with me	200310152149	601
13	Christy Lentoh	You seem a little sick	200310141722	601
270	Mom	preparing for finals	200304101533	603
283	Betty Benter	Of great restaurants and Park...	200303311022	606
233	Mary Kilpatrick	Birthday celebration come and	200307021200	607
383	Dad	Religion class	200302111908	607
378	Dad	Saw your old friend	200302141832	607
440	Nathan Breker	started dating Sophy	200301151212	609
152	Reese Aren	cast taken off...	200308011234	611
235	Novell Opportunity	opening in: Documentation	200307020929	615
346	Jobs R Us	jobs: meal planner	200302282229	617
381	Tanya Lundberg	How about hiking the Y?	200302131812	617
16	Christy Lentoh	Good luck in Moab	200310092239	619
484	Gary Hestely	New Semester	200301031222	621
424	Sophy Wendell	I hope you are doing better	200301251221	622
185	Nathan Breker	started dating Margie	200307151212	623
232	Novell Opportunity	opening in: Accouting	200307021228	623
7	Christy Lentoh	We need to talk when you ...	200310160149	626
162	Novell Opportunity	opening in: Secretarial	200307271237	626
271	Betty Benter	Good luck in Moab	200304092239	628
425	Mom	religion class cool	200301242138	629
82	June Jeppson	Please call.	200309032040	629
469	Cisco Jobs	opening in: Russian Translat.	200301041036	630
174	Nathan Breker	Margie is great	200307202118	630
403	Dad	busy with school	200302021300	634
67	Software Solutions	Interview request accepted	200309181352	634
119	Dad [office email]	chemistry not going well	200308152139	638
439	Calculus HelpOnline	Functions written wrong, the...	200301151312	639
182	Margie Largieh	Hey Margie	200307161212	640
47	Christy Lentoh	Thanks for help in dance class	200309231854	642
140	Soccer Fanatics	Fans storm city hall	200308071119	643
63	Christy Lentoh	The dance was fun.	200309182350	651
35	Nathan Breker	Christy is the greatest	200309262358	652
398	Global Concerns	Penguin population decreasing	200302061029	656

ArrayItems				
ID	Name	Subject	DateTime	ScrambleSort
10	Reese Aren	benching these days...	200310151219	656
489	MS Employment	career: LAN instructor	200301020951	662
426	Sophy Wendell	Soph, I think we need to talk	200301232309	663
219	Susan Wandter	new semester and rulers	200307040908	664
120	Reese Aren	started dancing...	200308152012	667
465	Cisco Jobs	opening in: Network Testing	200301050927	669
39	Tandem Newsflash	career: Chauffer	200309251654	671
180	US Hotline Jobs	jobs: stadium cleaner	200307180721	677
328	Bob K. Runchleg	walking almost normally...	200303150909	682
365	Nathan Breker	Tanya is moving to Chicago	200302191513	682
352	Nathan Breker	What is up with girls?	200302222345	683
59	Dad [office email]	Astronomy room	200309192259	684
390	Jobs R Us	jobs: math tutor	200302072033	685
155	Software Solutions	Accepting applications	200308011100	690
361	Jobs R Us	jobs: yard stick reader	200302201445	692
473	Cisco Jobs	opening in: Web Design	200301040926	692
89	Tandem Newsflash	career: Network Technician	200308282238	695
36	Christy Lentoh	Cool that you help with	200309262258	705
3	Nathan Breker	where am I going wrong?	200310171732	706
99	Nathan Breker	June and her roommates	200308221938	706
497	Cisco Jobs	opening in: Software Dev.	200301010630	711
419	Jobs R Us	jobs: physics tutor	200301271222	713
22	US Hotline Jobs	jobs: math lab assistant	200310022358	714
226	Tandem Newsflash	career: Document. Research	200307031247	717
254	Jobs R Us	jobs: weekend warrior	200305262110	717
312	Global Concerns	Hundreds of penguins dead	200303201101	719
451	Jobs R Us	jobs: thermometer reader	200301120923	720
477	Fred Crunchberry	getting better after last sem...	200301031822	722
215	US Hotline Jobs	jobs: janitor assistant	200307041029	724
145	Software Solutions	Application not received	200308041129	725
135	US Hotline Jobs	jobs: math tutor	200308072033	725
239	Tandem Newsflash	career: PC technician	200307010652	728
472	MS Employment	career: Eng/French Translat.	200301040948	734
471	Bob K. Runchleg	injured knee...	200301041019	736
319	MS Employment	career: Eng/Span Translat.	200303182342	737
147	Soccer Fanatics	Finland vs Great Britain	200308031056	737
157	Dad [office email]	Meeting with coordinator	200307282328	741
382	Tanya Lundberg	Tanya, did yoy pass the quiz?	200302112222	741
350	Nathan Breker	Tanya has been really busy	200302251313	749
218	Novell Opportunity	opening in: Web Design	200307040926	751
61	Nathan Breker	Met a great chick	200309182356	752
479	Byron Gibson	failed that class last semester	200301031756	753
221	Dale Yortber	sister better after wreck	200307031832	754
85	Reese Aren	using arm...	200309022345	755
34	Christy Lentoh	I know a place in Heber	200309271044	756
169	Margie Largieh	I hope you get better soon	200307251221	758

ArrayItems				
ID	Name	Subject	DateTime	ScrambleSort
460	Duane Lester	movies to see... Star Wars	200301061219	760
253	Jobs R Us	jobs: yardwork	200305262125	761
177	Tandem Newsflash	career: Network Engineer	200307180744	764
496	Cisco Jobs	opening in: Marketing	200301010631	764
493	MS Employment	career: Custodial	200301010653	766
347	Bob K. Runchleg	hopping on other leg... again...	200302281819	766
438	Dad	doing well in physics	200301152121	767
375	Bob K. Runchleg	started dancing...	200302152012	769
329	Gems	When we are helping, we are...	200303150909	770
248	Cindy Anderton	There is a dance on Friday	200306012129	771
217	Tandem Newsflash	career: Eng/French Translat.	200307040948	771
165	Nathan Breker	Looks like Margie is...	200307271221	772
373	Bob K. Runchleg	slipped on wet grass... ouch...	200302161604	773
245	Dave Franconi	July 4th...	200307010600	774
172	Nathan Breker	Margie got mad at me	200307222309	775
428	Mom	going to classes	200301222209	779
28	Christy Lentoh	Of great restaurants and Park...	200309301022	781
5	Nathan Breker	Christy getting married in Nov.	200310170854	782
130	June Jeppson	How is class going?	200308100632	783
167	Margie Largieh	How is the cold	200307260934	784
139	Soccer Fanatics	Bolivian Indian chief to allow...	200308071254	784
470	Jobs R Us	jobs: janitor assistant	200301041029	785
158	US Hotline Jobs	jobs: custodial supervisor	200307282326	787
448	National Instruments	Thank you for your inquiry	200301121121	790
25	US Hotline Jobs	jobs: bathroom designer	200310022324	790
194	Novell Opportunity	opening in: Sales	200307120933	790
457	Jobs R Us	jobs: shuttle driver	200301081312	794
316	Nathan Breker	Met a great girl	200303182356	794
136	US Hotline Jobs	jobs: janitor	200308072019	795
499	Dave Franconi	New Years...	200301010601	796
41	Novell Opportunity	opening in: Publications	200309251632	802
24	Tandem Newsflash	career: Newscaster	200310022346	804
466	Jobs R Us	jobs: model	200301050922	805
260	Nathan Breker	Betty getting married in Nov.	200304170854	810
21	Christy Lentoh	Great idea for a date	200310031229	813
129	Nathan Breker	June is great	200308110722	815
70	Soccer Fanatics	Japan in tournament	200309181317	819
275	Nathan Breker	Saturday date with Betty	200304041229	821
282	Bob K. Runchleg	started to jog...	200304020808	831
474	Susan Wandter	new semester and books	200301040908	832
301	Calculus HelpOnline	Correction on previous reply.	200303241011	832
335	Calculus HelpOnline	Looking better on reasoning...	200303041054	834
132	Nathan Breker	started dating June	200308090632	834
144	Mom	expert at waffles	200308041929	838
74	Gems	It starts with you	200309150909	840
23	Nathan Breker	Movies with sweet girl	200310022351	841

ArrayItems				
ID	Name	Subject	DateTime	ScrambleSort
213	US Hotline Jobs	jobs: English tutor	200307041056	842
307	Gems	Keep the faith	200303221102	843
285	Cisco Jobs	opening in: Spanish Translat.	200303310034	845
367	Dad	Dates few and far	200302182345	846
209	Tandem Newsflash	career: Employee Relations	200307050949	846
336	Nathan Breker	Tanya not interested	200303032043	853
142	Dad [office email]	still have a buck or two	200308061929	855
359	Nathan Breker	What is Chicago like?	200302201904	856
198	Mom	Missing home, again	200307101232	856
414	Jobs R Us	jobs: Spanish tutor	200301282321	859
349	Nathan Breker	Saw Tanya with another guy	200302272105	860
326	Dad	Parking ticket	200303161732	860
461	Jobs R Us	jobs: web designer	200301061213	861
488	Mary Kilpatrick	New Years celebration over	200301021200	862
187	Software Solutions	Opening #66709, call 555-1234	200307141333	863
32	Nathan Breker	AF and Christy	200309280144	871
229	Gary Hestely	New Semester, new job	200307031222	871
68	Gems	Where your treasure is, so...	200309181350	872
200	Joana Mogberry	lunch at Wendys	200307081323	872
113	Mom	program coordinator neat dude	200308182345	872
324	Gymbaub Ind	Application awaiting review	200303181342	876
1	Nathan Breker	Great grills, but apparently...	200310180454	877
29	US Hotline Jobs	jobs: baby cotton swab tester	200309300056	878
43	Inquiries Ldating	My name is Leslie, #45049	200309251432	883
31	Nathan Breker	Park City and Christy	200309300022	885
189	Gymbaub Ind	Thx interest in water equipm	200307141212	886
134	Novell Opportunity	opening in: French Translat.	200308072035	886
150	Dad [office email]	Electrical eng. & cslr	200308011858	890
100	June Jeppson	June, we really need to talk.	200308221930	892
234	Tandem Newsflash	career: LAN instructor	200307020951	905
37	Nathan Breker	Christy leads cheers	200309252356	909
12	Nathan Breker	Christy a little bored	200310142222	912
281	Betty Benter	A new movie came out	200304021251	912
118	Reese Aren	slipped on wet grass... ouch...	200308161604	914
6	Christy Lentoh	I wish you the best.	200310162354	919
93	June Jeppson	Hope you feel better.	200308272111	921
121	Nathan Breker	We went to the canyon	200308150945	921
420	Nathan Breker	Looks like Sophy is...	200301271221	925
55	Dad [office email]	stats and failing	200309211708	926
131	Mom	stats class fun but hard	200308091832	927
247	Cindy Anderton	I am starting out myself...	200306072359	927
430	Dad	English term paper	200301182222	937
482	Jobs R Us	jobs: rifle cleaner	200301031225	938
453	Mom	Missing home	200301101232	938
404	Calculus HelpOnline	Formula needs adjusting...	200302021200	939
151	Mom	saw Wilma Duherst	200308011300	941

ArrayItems				
ID	Name	Subject	DateTime	ScrambleSort
486	Jobs R Us	jobs: newspaper editor	200301021245	944
363	Jobs R Us	jobs: day care assistant	200302201414	944
191	Dad [office email]	school still is great	200307121232	946
246	Nathan Breker	Cindy is a freshman..	200306092359	946
288	Mom	planning for next term	200303271533	947
123	Dad [office email]	Saw your old friend again	200308141832	948
208	Reese Aren	full arm cast...	200307051144	951
97	Nathan Breker	What is up with women?	200308222345	952
341	Tanya Lundberg	We need to talk, please call.	200303022343	954
417	Cisco Jobs	opening in: Secretarial	200301271237	955
125	June Jeppson	There is a festival...	200308140945	955
276	Betty Benter	Great idea for a date	200304031229	957
487	Cisco Jobs	opening in: Accouting	200301021228	957
205	Duane Lester	movies to see... Rocky 10	200307061219	957
422	Sophy Wendell	How are the sniffles	200301260934	959
379	Nathan Breker	Tanya and I are	200302141812	960
370	Nathan Breker	Tanya loves cars	200302180822	963
287	Nathan Breker	Heber and Betty	200303280144	963
156	Tandem Newsflash	career: Eng/Russ Translat.	200307282341	968
160	Nathan Breker	Sad news from Margie	200307280934	969
259	Nathan Breker	what is wrong with me	200304171732	971
168	Dad [office email]	dance class did me good	200307252209	972
171	Margie Largieh	Marge, I think we need to talk	200307232309	973
338	Gymbaub Ind	Sales position opening	200303030940	973
369	Tanya Lundberg	Chicago sounds great.	200302181513	975
11	Nathan Breker	Able to go, Christy sick	200310151035	976
340	Bob K. Runchleg	standing on leg...	200303022345	977
355	Tanya Lundberg	Tanya, we really need to talk.	200302221930	978
303	PowerMan	Food tips for the body builder	200303231354	978
290	Nathan Breker	Betty is the greatest	200303262358	978
299	MyMagazine	Your subscription ends in ...	200303251011	982
30	Novell Opportunity	opening in: Spanish Translat.	200309300034	984
46	Stats HelpOnline	Correction on previous reply.	200309241011	985
192	Global Concerns	Skinny penguins baking	200307121132	986
272	Nathan Breker	Betty and graduation	200304081133	986
33	Mom	planning for next semester	200309271533	987
492	John Fitzgerald	New Years was great	200301020900	987
406	Mom	saw Donetta Adamson	200302011300	991
98	June Jeppson	June, please answer.	200308222340	993
199	Tandem Newsflash	career: Documentation	200307081340	993

APPENDIX D: THE WEB PROGRAM CODE

Appendix D.1 - default.asp

Appendix D.2 - subAndFuncAdmin.asp

Appendix D.3 - subroutinesAndFunctions.asp

Appendix D.4 - thesisAdmin.asp

Appendix D.5 - thesisDatabaseConnection.asp

Appendix D.1 - default.asp

```
<%@ LANGUAGE=VBScript%>
<!-- #include file = "subroutinesAndFunctions.asp" -->
<html>
<body>
<form method="post" action="default.asp#stopHere">
<%
'Response.Write(Request.Form())
'Response.Write("<br>" & Request.Form("theReply"))

session("setsToDisplayPerSortType") = 1

'things to do:
' compare given answers to real answers, get a percentage correct
thingamajig
'show order thing amajig

''' "current exercise number" section;
''' this section must occur sometime before the introduction
ExerciseQuestionnaire or coupon section

'Response.Write("where am I " & Request.Form())

'session("introductionExercisesQuestionnaireOrCoupon")="Coupon"

if ((Request.Form("submitButtonCouponRequestComplete")<>"") AND
(session("introductionExercisesQuestionnaireOrCoupon")="Coupon")) then
    saveIntervieweeResponsesCoupon()
    session("introductionExercisesQuestionnaireOrCoupon")="AllDone"
end if

if ((Request.Form("submitButtonQuestionnaireComplete")<>"") AND
(session("introductionExercisesQuestionnaireOrCoupon")="Questionnaire")
) then
    ' check responses for the multiple choice ones to make sure that at
    least one radio button has been set
    ' hold on, this should be a text response as well
    session("radioButtonQuestionsMissed") =
verifyRadioButtonQuestionsAnswered()
    if(session("radioButtonQuestionsMissed") = "") then
        session("QuestionnaireTimeEnd") = Now()
        setTimeStartAndEndQuestionnaire()
        saveIntervieweeResponsesExerciseAndQuestionnaire()
        session("introductionExercisesQuestionnaireOrCoupon")="Coupon"
    end if
'Response.Write("in here ")
else
'Response.Write("not in ")

end if

if ((session("introductionExercisesQuestionnaireOrCoupon")="Exercises")
OR
```

```

(session("introductionExercisesQuestionnaireOrCoupon")="introduction"))
then
  if(Request.Form("currentExerciseNumber") <> "") then
    currentExerciseNumber = Request.form("currentExerciseNumber")
    If((InStr(Request.Form(),"sort")=0) AND
(InStr(Request.Form(),"swap")=0)) Then 'don't reset the time if the
user is changing the column sort
      session("timeEnd" & session("exerciseDisplayOrder" &
currentExerciseNumber)) = Now()
      setTimeStartAndEnd(currentExerciseNumber)
      currentExerciseNumber =
updateReplyForExerciseNumber(currentExerciseNumber)
    End If
  else
    currentExerciseNumber = 1
  end if
end if

''' "introduction Exercise Questionnaire or coupon" section
'''This section must occur sometime after the "current exercise number"
section
if(session("introductionExercisesQuestionnaireOrCoupon")="") then
  session("introductionExercisesQuestionnaireOrCoupon")="introduction"
elseif(Request.Form("introductionSubmitButton") = "Continue") then
  session("introductionExercisesQuestionnaireOrCoupon")="Exercises"
elseif((Request.Form("exerciseReplySubmitButton") <> "") AND
((session("setsToDisplayPerSortType")*3*2) < currentExerciseNumber+0))
then
  session("introductionExercisesQuestionnaireOrCoupon")="Questionnaire"
end if

if(session("introductionExercisesQuestionnaireOrCoupon")="AllDone")
then
  Response.Redirect("http://www.byu.edu")
  'response.write("<a href=http://www.byu.edu>click here</a>")
elseif(session("introductionExercisesQuestionnaireOrCoupon")="Coupon")
then
  displayThankYouCoupon
elseif(session("introductionExercisesQuestionnaireOrCoupon")="Questionn
aire") then
  displayQuestionnaire()
  session("QuestionnaireTimeStart") = Now()
elseif(session("introductionExercisesQuestionnaireOrCoupon")="Exercises
") then
  'Response.Write("<br>Exercise Number: " & currentExerciseNumber)
  'Response.Write("<br>" & Request.form())
  displayExercise(currentExerciseNumber)
  If((InStr(Request.Form(),"sort")=0) AND
(InStr(Request.Form(),"swap")=0)) Then 'don't reset the time if the
user is changing the column sort
    session("timeStart" & session("exerciseDisplayOrder" &
currentExerciseNumber)) = Now()
  End If

```

```

'Response.Write("<br>Previous Time Start" & session("timeStart" &
session("exerciseDisplayOrder" & currentExerciseNumber-1)))
'Response.Write("<br>Previous Time End: " & session("timeEnd" &
session("exerciseDisplayOrder" & currentExerciseNumber-1)))
'Response.Write("<br>Current Time Start" & session("timeStart" &
session("exerciseDisplayOrder" & currentExerciseNumber)))

else
    numberOfAllowableQuestionnaires = 100
    tempNumber = numberOfCompletedQuestionnaires()
' response.write("Number of Completed Questionnaires: " & tempNumber)
' if(numberOfAllowableQuestionnaires <= tempNumber) then
'     response.write("Thank you for your consideration, however, the
maximum number of participants for this survey has been reached.<br><a
href=http://www.byu.edu>click here</a>")
' else
'     initializeFields()
'     displayIntroduction()
' end if

    initializeFields()
    displayIntroduction()

end if

%>
</form>
</body>
</html>

```


Appendix D.2 - subAndFuncAdmin.asp

```
<%

sub subCouponStatus()

'Response.Write(Request.Form())

Dim Conn 'Our connection object
Dim RSAuthors
Dim SQL

%><!--      #include file = "thesisDatabaseConnection.asp" --><%

    if(Request.Form("submitButtonAdminCouponStatus") <> "") then
        for tempCount = 1 to Request.Form("numberOfCoupons")
            if(Request.form("adminCouponActive" & Request.Form("couponNumber"
& tempCount)) <> "") then theActive = "TRUE" else theActive = "FALSE"
end if
            s =      " Active                = " & theActive
& "      "
            s = s & ", Item                = '" &
convertToDatabaseSingleQuotes(Request.form("adminCouponItem"
& Request.Form("couponNumber" & tempCount))) & "' "
            s = s & ", Origin                = '" &
convertToDatabaseSingleQuotes(Request.form("adminCouponOrigin"
& Request.Form("couponNumber" & tempCount))) & "' "
            s = s & ", NumberRemaining      = " &
Request.form("adminCouponNumberRemaining"
& Request.Form("couponNumber" & tempCount)) & " "
            s = s & ", NumberOriginalCount = " &
Request.form("adminCouponNumberOriginalCount"
& Request.Form("couponNumber" & tempCount)) & " "
            s = s & ", ValidLocations      = '" &
convertToDatabaseSingleQuotes(Request.form("adminCouponValidLocations"
& Request.Form("couponNumber" & tempCount))) & "' "
            SQL = "UPDATE Coupon SET " & s & " WHERE ID = " &
Request.Form("couponNumber" & tempCount)
            Conn.Execute(SQL)
        next
    end if

    if(Request.Form("submitButtonAdminIntervieweeCouponStatus") <> "")
then
        for tempCount = 1 to Request.Form("numberOfCouponIntervieweeInfo")
            theNewItemSentStatus =
Request.form("adminIntervieweeCouponItemSent"                &
Request.Form("couponIntervieweeInfoNumber" & tempCount))
            theOldItemSentStatus =
Request.form("adminIntervieweeCouponItemSentCurrentDatabaseEntry" &
Request.Form("couponIntervieweeInfoNumber" & tempCount))

            includeItemSentDateTime = FALSE
            if(theNewItemSentStatus <> "") then
```

```

        theItemSent = "TRUE"
        if(theNewItemSentStatus <> theOldItemSentStatus) then
            includeItemSentDateTime = TRUE
            theItemSentDateTime = now()
        end if
    else
        theItemSent = "FALSE"
        theItemSentDateTime = ""
        includeItemSentDateTime = TRUE
    end if
    s = " ItemSent = " & theItemSent & " "
    if(includeItemSentDateTime = TRUE) then
        s = s & ", ItemSentDateTime = '" & theItemSentDateTime & "'
    end if
    SQL = "UPDATE CouponIntervieweeInfo SET " & s & " WHERE ID = " &
Request.Form("couponIntervieweeInfoNumber" & tempCount)
    Conn.Execute(SQL)
next
end if

if(Request.Form("submitButtonAdminCouponAddItem") <> "") then
    sField = " Active "
    sField = sField & " , Item "
    sField = sField & " , Origin "
    sField = sField & " , NumberRemaining "
    sField = sField & " , NumberOriginalCount "
    sField = sField & " , ValidLocations "

    sValue = " " & Request.form("CouponAddItem_Active")
& " "
    sValue = sValue & ", '" &
convertToDatabaseSingleQuotes(Request.form("CouponAddItem_item"))
& "' "
    sValue = sValue & ", '" &
convertToDatabaseSingleQuotes(Request.form("CouponAddItem_Origin"))
& "' "
    sValue = sValue & ", " &
convertToDatabaseSingleQuotes(Request.form("CouponAddItem_NumberRemaini
ng")) & " "
    sValue = sValue & ", " &
convertToDatabaseSingleQuotes(Request.form("CouponAddItem_NumberOrigina
lCount")) & " "
    sValue = sValue & ", '" &
convertToDatabaseSingleQuotes(Request.form("CouponAddItem_ValidLocation
s")) & "' "

    SQL = "INSERT INTO Coupon (" & sField & ") VALUES (" & sValue & ")
"
    Conn.Execute(SQL)
end if

SQL = "Select * from Coupon ORDER BY ID "
RS.Open SQL,Conn
Response.Write("<table border=1 align=center><tr><td>")
Response.Write("<table align=center>")
Response.Write("<tr>")

```

```

Response.Write(" <th width=20>ID</th>")
Response.Write(" <th width=20>Active</th>")
Response.Write(" <th width=170>Item</th>")
Response.Write(" <th width=150>Origin</th>")
Response.Write(" <th width=220>Remaining / Original Count</th>")
Response.Write(" <th width=150>Valid Locations</th>")
Response.Write("</tr>")
count = 0
Do While Not RS.EOF
    count = count + 1
    theID = RS("ID")
    theActive = RS("Active")
    Response.Write("<tr>")
    Response.Write("<td><center>" & theID & "</center></td>")
    if(theActive = "True") then checkedActive = "checked" else
checkedActive = "" end if
    Response.Write("<td><center><input type='checkbox'
name='adminCouponActive" & theID & "' value='True' " &
checkedActive & "></center></td>")
    Response.Write("<td><input type='text' name='adminCouponItem" &
theID & "' value='" & RS("Item") & "' size='25'" > </td>")
    Response.Write("<td><input type='text' name='adminCouponOrigin" &
theID & "' value='" & RS("Origin") & "' size='25'" > </td>")
    Response.Write("<td><center>")
    Response.Write(" <input type='text'
name='adminCouponNumberRemaining" & theID & "' value='" &
RS("NumberRemaining") & "' size='3' maxlength='3'" > ")
    Response.Write(" <input type='text'
name='adminCouponNumberOriginalCount" & theID & "' value='" &
RS("NumberOriginalCount") & "' size='3' maxlength='3'" > ")
    Response.Write("</center></td>")
    Response.Write("<td><input type='text'
name='adminCouponValidLocations" & theID & "' value='" &
RS("ValidLocations") & "' size='25'" > </td>")
    Response.Write("</tr>")
    Response.Write("<input type='hidden' name='couponNumber" & count &
"' value='" & theID & "'>")
    RS.MoveNext
Loop
RS.Close
Response.Write("<input type='hidden' name='numberOfCoupons' value='"
& count & "'>")
Response.Write("<tr>")
if(Request.Form("submitButtonAdminCouponStatus") <> "") then
bgcolorUpdateCouponStatus = "bgcolor = '#FFDDFF'" end if
Response.Write(" <td colspan='6' align=center " &
bgcolorUpdateCouponStatus & "><input type='submit'
name='submitButtonAdminCouponStatus' value='update'" ></th>")
Response.Write("</tr>")
Response.Write("<tr><td>&nbsp;&lt;/td></tr>")
Response.Write("<tr>")
Response.Write(" <td><input type='submit'
name='submitButtonAdminCouponAddItem' value='add'" ></td>")
Response.Write(" <td><center><input type='checkbox'
name='CouponAddItem_Active' value='True' checked></center></td>")
Response.Write(" <td><input type='text' name='CouponAddItem_Name'
value='...item...'></td>")

```

```

Response.Write(" <td><input type='text' name='CouponAddItem_Origin'
value='...origin...'></td>")
Response.Write(" <td><center>")
Response.Write(" <input type='text'
name='CouponAddItem_NumberRemaining' value='1' size='3' maxlength='3'>
/ ")
Response.Write(" <input type='text'
name='CouponAddItem_NumberOriginalCount' value='1' size='3'
maxlength='3'>")
Response.Write(" </center></td>")
Response.Write(" <td><input type='text'
name='CouponAddItem_ValidLocations' value='...valid
location(s)...'></td>")
Response.Write("</tr>")
Response.Write("</table>")
Response.Write("</td></tr></table>")

Response.Write("<br><br>")

SQL = "Select CII.ID AS theID, CII.MailingAddress AS MailingAddress,
CII.EmailAddress AS EmailAddress, C.Item AS Item, "
SQL = SQL & " C.Origin AS Origin, C.ValidLocations AS ValidLocations,
CII.ItemSent AS ItemSent, "
SQL = SQL & " CII.EntryDateTime AS EntryDateTime,
CII.ItemSentDateTime AS ItemSentDateTime "
SQL = SQL & " from CouponIntervieweeInfo CII, Coupon C WHERE
CII.CouponID = C.ID ORDER BY CII.ID "
RS.Open SQL,Conn
Response.Write("<table border=1 align=center><tr><td>")
Response.Write("<table align=center>")
Response.Write("<tr>")
Response.Write(" <th width=20 valign='top'>ID</th>")
Response.Write(" <th width=170 valign='top'><font
color='white'>_____</font>Mailing<font
color='white'>_</font>Address<font color='white'>_____</font></th>")
Response.Write(" <th width=170 valign='top'>Email Address</th>")
Response.Write(" <th width=170 valign='top'>Initial Entry</th>")
Response.Write(" <th width=150 valign='top'>Coupon</th>")
Response.Write(" <th width=150 valign='top'>Origin</th>")
Response.Write(" <th width=150 valign='top'>Valid Locations</th>")
Response.Write(" <th width=100 valign='top'>Shipped Y/N</th>")
Response.Write(" <th width=100 valign='top'>Shipped<font
color='white'>_</font>Time</th>")
Response.Write("</tr>")
count = 0
Do While Not RS.EOF
count = count + 1
theID = RS("theID")
theItemSent = RS("ItemSent")
if(theItemSent = "False") then tmpBgColor = "bgcolor = '#FFAAFF'"
else tmpBgColor = "" end if
Response.Write("<tr " & tmpBgColor & ">")
Response.Write("<td align=center valign='top'><center>" & theID &
"</center></td>")
Response.Write("<td align=center valign='top'>" &
RS("MailingAddress") & "</td>")

```

```

        Response.Write("<td align=center valign='top'>" &
RS("EmailAddress") & "</td>")
        Response.Write("<td align=center valign='top'>" &
RS("EntryDateTime") & "</td>")
        Response.Write("<td align=center valign='top'>" & RS("Item")
& "</td>")
        Response.Write("<td align=center valign='top'>" & RS("Origin")
& "</td>")
        Response.Write("<td align=center valign='top'>" &
RS("ValidLocations") & "</td>")
        if(theItemSent = "True") then
            checkedActive = "checked"
            fromDatabase = "True"
        else
            checkedActive = ""
            fromDatabase = ""
        end if
        Response.Write("<td align=center><center><input type='checkbox'
name='adminIntervieweeCouponItemSent' & theID & '' value='True' " &
checkedActive & "></center>")
        Response.Write(" <input type='hidden'
name='adminIntervieweeCouponItemSentCurrentDatabaseEntry" &
theID & '' value='' & fromDatabase & "'></center></td>")
        Response.Write("<td align=center>" & RS("ItemSentDateTime") &
"</td>")
        Response.Write("</tr>")
        Response.Write("<input type='hidden'
name='couponIntervieweeInfoNumber' & count & '' value='' & theID &
''>")
        RS.MoveNext
    Loop
    RS.Close
    Response.Write("<input type='hidden'
name='numberOfCouponIntervieweeInfo' value='' & count & ''>")
    Response.Write("<tr>")
    if(Request.Form("submitButtonAdminIntervieweeCouponStatus") <> "")
then bgcolorUpdateIntervieweeCouponStatus = "bgcolor = '#FFDDFF'" end
if
    Response.Write(" <td colspan='9' align=center " &
bgcolorUpdateIntervieweeCouponStatus & "><input type='submit'
name='submitButtonAdminIntervieweeCouponStatus' value='update'></th>")
    Response.Write("</tr>")
    Response.Write("</table>")
    Response.Write("</td></tr></table>")

end sub

sub subExerciseResults()
    Dim Conn 'Our connection object
    Dim RSAuthors
    Dim SQL

%><!-- #include file = "thesisDatabaseConnection.asp" --><%

    Response.Write("<table align=center>")
    Response.Write("<tr>")
    Response.Write(" <th valign='bottom'></th>")

```

```

    Response.Write(" <th valign='bottom'></th>")
    Response.Write(" <th valign='bottom'
colspan='3'>%<br>Corr<br>Stnd</th>")
    Response.Write(" <th valign='bottom'
colspan='3'>%<br>Incorr<br>Stnd</th>")
    Response.Write(" <th valign='bottom'>%<br>Corr</th>")
    Response.Write(" <th valign='bottom'>%<br>Inco</th>")
    Response.Write(" <th valign='bottom'
colspan='3'>%<br>Corr<br>Mult</th>")
    Response.Write(" <th valign='bottom'
colspan='3'>%<br>Incorr<br>Mult</th>")
    Response.Write(" <th valign='bottom'>%<br>Corr</th>")
    Response.Write(" <th valign='bottom'>%<br>Incorr</th>")
    Response.Write(" <th valign='bottom'
colspan='3'>Time<br>Stnd<br>(sec)</th>")
    Response.Write(" <th valign='bottom'>Time<br>(sec)</th>")
    Response.Write(" <th valign='bottom'
colspan='3'>Time<br>Mult<br>(sec)</th>")
    Response.Write(" <th valign='bottom'>Time<br>(sec)</th>")
    Response.Write(" <th valign='bottom'>Time<br>(sec)</th>")
    Response.Write(" <th valign='bottom'>%<br>Time<br><u>Stnd</u></th>")
    Response.Write(" <th valign='bottom'>%<br>Time<br><u>Mult</u></th>")
    Response.Write(" <th
valign='bottom'>Ratio<br>Time<br><u>Mult</u></th>")
    Response.Write(" <th valign='bottom'>Time<br><u>Qstnr</u></th>")
    Response.Write(" <th
valign='bottom'>Time<br>Exrcs+<br><u>Qstnr</u></th>")
    Response.Write("</tr>")

    Response.Write("<tr>")
    Response.Write(" <th>ID</th>")
    Response.Write(" <th>Interview Date</th>")
    Response.Write(" <th>N</th><th>S</th><th>D</th>")
    Response.Write(" <th>N</th><th>S</th><th>D</th>")
    Response.Write(" <th>Stnd</th>")
    Response.Write(" <th>Stnd</th>")
    Response.Write(" <th>N</th><th>S</th><th>D</th>")
    Response.Write(" <th>N</th><th>S</th><th>D</th>")
    Response.Write(" <th>Mult</th>")
    Response.Write(" <th>Mult</th>")
    Response.Write(" <th>N</th><th>S</th><th>D</th>")
    Response.Write(" <th>Stnd</th>")
    Response.Write(" <th>N</th><th>S</th><th>D</th>")
    Response.Write(" <th>Mult</th>")
    Response.Write(" <th>ALL</th>")
    Response.Write(" <th>Both</th>")
    Response.Write(" <th>Both</th>")
    Response.Write(" <th>Stnd</th>")
    Response.Write(" <th></th>")
    Response.Write(" <th></th>")
    Response.Write("</tr>")

    session("PercentCorrectStandardWayFindName") = 0
    session("PercentCorrectStandardWayFindSubject") = 0
    session("PercentCorrectStandardWayFindDate") = 0
    session("PercentIncorrectStandardWayFindName") = 0
    session("PercentIncorrectStandardWayFindSubject") = 0

```

```

session("PercentIncorrectStandardWayFindDate") = 0
session("PercentCorrectStandardWay") = 0
session("PercentIncorrectStandardWay") = 0
session("PercentCorrectMultipleWayFindName") = 0
session("PercentCorrectMultipleWayFindSubject") = 0
session("PercentCorrectMultipleWayFindDate") = 0
session("PercentIncorrectMultipleWayFindName") = 0
session("PercentIncorrectMultipleWayFindSubject") = 0
session("PercentIncorrectMultipleWayFindDate") = 0
session("PercentCorrectMultipleWay") = 0
session("PercentIncorrectMultipleWay") = 0
session("TotalTimeStandardWayFindName") = 0
session("TotalTimeStandardWayFindSubject") = 0
session("TotalTimeStandardWayFindDate") = 0
session("TotalTimeStandardWay") = 0
session("TotalTimeMultipleWayFindName") = 0
session("TotalTimeMultipleWayFindSubject") = 0
session("TotalTimeMultipleWayFindDate") = 0
session("TotalTimeMultipleWay") = 0
session("TotalTimeBothWays") = 0
session("PercentTimeStandardOverBoth") = 0
session("PercentTimeMultipleOverBoth") = 0
session("RatioTimeMultipleOverStandard") = 0
session("QuestionnaireTotalTime") = 0
session("TotalTimeExercisePlusQuestionnaire") = 0

```

```

SQL = "Select * from Interviewee ORDER BY ID "
RS.Open SQL,Conn
count = 0
Do While Not RS.EOF
    count = count + 1
    PercentCorrectStandardWayFindName =
RS("PercentCorrectStandardWayFindName")
    PercentCorrectStandardWayFindSubject =
RS("PercentCorrectStandardWayFindSubject")
    PercentCorrectStandardWayFindDate =
RS("PercentCorrectStandardWayFindDate")
    PercentIncorrectStandardWayFindName =
RS("PercentIncorrectStandardWayFindName")
    PercentIncorrectStandardWayFindSubject =
RS("PercentIncorrectStandardWayFindSubject")
    PercentIncorrectStandardWayFindDate =
RS("PercentIncorrectStandardWayFindDate")
    PercentCorrectStandardWay =
RS("PercentCorrectStandardWay")
    PercentIncorrectStandardWay =
RS("PercentIncorrectStandardWay")
    PercentCorrectMultipleWayFindName =
RS("PercentCorrectMultipleWayFindName")
    PercentCorrectMultipleWayFindSubject =
RS("PercentCorrectMultipleWayFindSubject")
    PercentCorrectMultipleWayFindDate =
RS("PercentCorrectMultipleWayFindDate")
    PercentIncorrectMultipleWayFindName =
RS("PercentIncorrectMultipleWayFindName")

```

```

PercentIncorrectMultipleWayFindSubject =
RS("PercentIncorrectMultipleWayFindSubject")
PercentIncorrectMultipleWayFindDate =
RS("PercentIncorrectMultipleWayFindDate")
PercentCorrectMultipleWay =
RS("PercentCorrectMultipleWay")
PercentIncorrectMultipleWay =
RS("PercentIncorrectMultipleWay")
TotalTimeStandardWayFindName =
RS("TotalTimeStandardWayFindName")
TotalTimeStandardWayFindSubject =
RS("TotalTimeStandardWayFindSubject")
TotalTimeStandardWayFindDate =
RS("TotalTimeStandardWayFindDate")
TotalTimeStandardWay = RS("TotalTimeStandardWay")
TotalTimeMultipleWayFindName =
RS("TotalTimeMultipleWayFindName")
TotalTimeMultipleWayFindSubject =
RS("TotalTimeMultipleWayFindSubject")
TotalTimeMultipleWayFindDate =
RS("TotalTimeMultipleWayFindDate")
TotalTimeMultipleWay = RS("TotalTimeMultipleWay")
TotalTimeBothWays = RS("TotalTimeBothWays")
PercentTimeStandardOverBoth =
RS("PercentTimeStandardOverBoth")
PercentTimeMultipleOverBoth =
RS("PercentTimeMultipleOverBoth")
RatioTimeMultipleOverStandard =
RS("RatioTimeMultipleOverStandard")
QuestionnaireTotalTime =
RS("QuestionnaireTotalTime")

```

```

Response.Write("<tr>")
Response.Write(" <td>" & RS("ID") & "</td>")
Response.Write(" <td>" & RS("InterviewDate") & "</td>")
Response.Write(" <td align='center'>" &
PercentCorrectStandardWayFindName & "</td><td align='center'>" &
PercentCorrectStandardWayFindSubject & "</td><td align='center'>" &
PercentCorrectStandardWayFindDate & "</td>")
Response.Write(" <td align='center'>" &
PercentIncorrectStandardWayFindName & "</td><td align='center'>" &
PercentIncorrectStandardWayFindSubject & "</td><td align='center'>" &
PercentIncorrectStandardWayFindDate & "</td>")
Response.Write(" <td align='center'>" & PercentCorrectStandardWay
& "</td>")
Response.Write(" <td align='center'>" &
PercentIncorrectStandardWay & "</td>")
Response.Write(" <td align='center'>" &
PercentCorrectMultipleWayFindName & "</td><td align='center'>" &
PercentCorrectMultipleWayFindSubject & "</td><td align='center'>" &
PercentCorrectMultipleWayFindDate & "</td>")
Response.Write(" <td align='center'>" &
PercentIncorrectMultipleWayFindName & "</td><td align='center'>" &
PercentIncorrectMultipleWayFindSubject & "</td><td align='center'>" &
PercentIncorrectMultipleWayFindDate & "</td>")

```

```

    Response.Write(" <td align='center'>" & PercentCorrectMultipleWay
& "</td>")
    Response.Write(" <td align='center'>" &
PercentIncorrectMultipleWay & "</td>")
    Response.Write(" <td align='center'>" &
TotalTimeStandardWayFindName & "</td><td align='center'>" &
TotalTimeStandardWayFindSubject & "</td><td align='center'>" &
TotalTimeStandardWayFindDate & "</td>")
    Response.Write(" <td align='center'>" & TotalTimeStandardWay &
"</td>")
    Response.Write(" <td align='center'>" &
TotalTimeMultipleWayFindName & "</td><td align='center'>" &
TotalTimeMultipleWayFindSubject & "</td><td align='center'>" &
TotalTimeMultipleWayFindDate & "</td>")
    Response.Write(" <td align='center'>" & TotalTimeMultipleWay &
"</td>")
    Response.Write(" <td align='center'>" & TotalTimeBothWays &
"</td>")
    Response.Write(" <td align='center'>" &
PercentTimeStandardOverBoth & "</td>")
    Response.Write(" <td align='center'>" &
PercentTimeMultipleOverBoth & "</td>")
    Response.Write(" <td align='center'>" &
RatioTimeMultipleOverStandard/100 & "</td>")
    Response.Write(" <td align='center'>" & QuestionnaireTotalTime &
"</td>")
    Response.Write(" <td align='center'>" & TotalTimeBothWays +
QuestionnaireTotalTime & "</td>")
    Response.Write("</tr>")

```

```

    session("PercentCorrectStandardWayFindName") =
session("PercentCorrectStandardWayFindName") +
PercentCorrectStandardWayFindName
    session("PercentCorrectStandardWayFindSubject") =
session("PercentCorrectStandardWayFindSubject") +
PercentCorrectStandardWayFindSubject
    session("PercentCorrectStandardWayFindDate") =
session("PercentCorrectStandardWayFindDate") +
PercentCorrectStandardWayFindDate
    session("PercentIncorrectStandardWayFindName") =
session("PercentIncorrectStandardWayFindName") +
PercentIncorrectStandardWayFindName
    session("PercentIncorrectStandardWayFindSubject") =
session("PercentIncorrectStandardWayFindSubject") +
PercentIncorrectStandardWayFindSubject
    session("PercentIncorrectStandardWayFindDate") =
session("PercentIncorrectStandardWayFindDate") +
PercentIncorrectStandardWayFindDate
    session("PercentCorrectStandardWay") =
session("PercentCorrectStandardWay") +
PercentCorrectStandardWay
    session("PercentIncorrectStandardWay") =
session("PercentIncorrectStandardWay") +
PercentIncorrectStandardWay
    session("PercentCorrectMultipleWayFindName") =
session("PercentCorrectMultipleWayFindName") +
PercentCorrectMultipleWayFindName

```

```

    session("PercentCorrectMultipleWayFindSubject") =
session("PercentCorrectMultipleWayFindSubject") +
PercentCorrectMultipleWayFindSubject
    session("PercentCorrectMultipleWayFindDate") =
session("PercentCorrectMultipleWayFindDate") +
PercentCorrectMultipleWayFindDate
    session("PercentIncorrectMultipleWayFindName") =
session("PercentIncorrectMultipleWayFindName") +
PercentIncorrectMultipleWayFindName
    session("PercentIncorrectMultipleWayFindSubject") =
session("PercentIncorrectMultipleWayFindSubject") +
PercentIncorrectMultipleWayFindSubject
    session("PercentIncorrectMultipleWayFindDate") =
session("PercentIncorrectMultipleWayFindDate") +
PercentIncorrectMultipleWayFindDate
    session("PercentCorrectMultipleWay") =
session("PercentCorrectMultipleWay") +
PercentCorrectMultipleWay
    session("PercentIncorrectMultipleWay") =
session("PercentIncorrectMultipleWay") +
PercentIncorrectMultipleWay
    session("TotalTimeStandardWayFindName") =
session("TotalTimeStandardWayFindName") +
TotalTimeStandardWayFindName
    session("TotalTimeStandardWayFindSubject") =
session("TotalTimeStandardWayFindSubject") +
TotalTimeStandardWayFindSubject
    session("TotalTimeStandardWayFindDate") =
session("TotalTimeStandardWayFindDate") +
TotalTimeStandardWayFindDate
    session("TotalTimeStandardWay") =
session("TotalTimeStandardWay") +
TotalTimeStandardWay
    session("TotalTimeMultipleWayFindName") =
session("TotalTimeMultipleWayFindName") +
TotalTimeMultipleWayFindName
    session("TotalTimeMultipleWayFindSubject") =
session("TotalTimeMultipleWayFindSubject") +
TotalTimeMultipleWayFindSubject
    session("TotalTimeMultipleWayFindDate") =
session("TotalTimeMultipleWayFindDate") +
TotalTimeMultipleWayFindDate
    session("TotalTimeMultipleWay") =
session("TotalTimeMultipleWay") +
TotalTimeMultipleWay
    session("TotalTimeBothWays") =
session("TotalTimeBothWays") + TotalTimeBothWays
    session("PercentTimeStandardOverBoth") =
session("PercentTimeStandardOverBoth") +
PercentTimeStandardOverBoth
    session("PercentTimeMultipleOverBoth") =
session("PercentTimeMultipleOverBoth") +
PercentTimeMultipleOverBoth
    session("RatioTimeMultipleOverStandard") =
session("RatioTimeMultipleOverStandard") +
RatioTimeMultipleOverStandard

```

```

        session("QuestionnaireTotalTime")           =
session("QuestionnaireTotalTime")                 +
QuestionnaireTotalTime
        session("TotalTimeExercisePlusQuestionnaire") =
session("TotalTimeExercisePlusQuestionnaire")    + TotalTimeBothWays +
QuestionnaireTotalTime

        RS.MoveNext
Loop
        RS.Close
        Response.Write("<tr><td>&nbsp;</td></tr>")
        Response.Write("<tr>")
        Response.Write("  <td></td>")
        Response.Write("  <th>AVERAGES</th>")
        Response.Write("  <td align='center'>" &
session("PercentCorrectStandardWayFindName")\count & "</td><td
align='center'>" &
session("PercentCorrectStandardWayFindSubject")\count & "</td><td
align='center'>" & session("PercentCorrectStandardWayFindDate")\count &
"</td>")
        Response.Write("  <td align='center'>" &
session("PercentIncorrectStandardWayFindName")\count & "</td><td
align='center'>" &
session("PercentIncorrectStandardWayFindSubject")\count & "</td><td
align='center'>" & session("PercentIncorrectStandardWayFindDate")\count
& "</td>")
        Response.Write("  <td align='center'>" &
session("PercentCorrectStandardWay")\count & "</td>")
        Response.Write("  <td align='center'>" &
session("PercentIncorrectStandardWay")\count & "</td>")
        Response.Write("  <td align='center'>" &
session("PercentCorrectMultipleWayFindName")\count & "</td><td
align='center'>" &
session("PercentCorrectMultipleWayFindSubject")\count & "</td><td
align='center'>" & session("PercentCorrectMultipleWayFindDate")\count &
"</td>")
        Response.Write("  <td align='center'>" &
session("PercentIncorrectMultipleWayFindName")\count & "</td><td
align='center'>" &
session("PercentIncorrectMultipleWayFindSubject")\count & "</td><td
align='center'>" & session("PercentIncorrectMultipleWayFindDate")\count
& "</td>")
        Response.Write("  <td align='center'>" &
session("PercentCorrectMultipleWay")\count & "</td>")
        Response.Write("  <td align='center'>" &
session("PercentIncorrectMultipleWay")\count & "</td>")
        Response.Write("  <td align='center'>" &
session("TotalTimeStandardWayFindName")\count & "</td><td
align='center'>" & session("TotalTimeStandardWayFindSubject")\count &
"</td><td align='center'>" &
session("TotalTimeStandardWayFindDate")\count & "</td>")
        Response.Write("  <td align='center'>" &
session("TotalTimeStandardWay")\count & "</td>")
        Response.Write("  <td align='center'>" &
session("TotalTimeMultipleWayFindName")\count & "</td><td
align='center'>" & session("TotalTimeMultipleWayFindSubject")\count &

```

```

"</td><td align='center'>" &
session("TotalTimeMultipleWayFindDate")\count & "</td>")
  Response.Write(" <td align='center'>" &
session("TotalTimeMultipleWay")\count & "</td>")
  Response.Write(" <td align='center'>" &
session("TotalTimeBothWays")\count & "</td>")
  Response.Write(" <td align='center'>" &
session("PercentTimeStandardOverBoth")\count & "</td>")
  Response.Write(" <td align='center'>" &
session("PercentTimeMultipleOverBoth")\count & "</td>")
  Response.Write(" <td align='center'>" &
(session("RatioTimeMultipleOverStandard")\count)/100 & "</td>")
  Response.Write(" <td align='center'>" &
session("QuestionnaireTotalTime")\count & "</td>")
  Response.Write(" <td align='center'>" &
session("TotalTimeExercisePlusQuestionnaire")\count & "</td>")
  Response.Write("</tr>")

```

```
Response.Write("</table>")
```

```
end sub
```

```
function functionLogin(tempRequest)
```

```
  if(tempRequest = "checkLogonCredentials") then
```

```
%><!--      #include file = "thesisDatabaseConnection.asp" --><%
```

```

    wSQL =          " WHERE loginID          = '" & Request.Form("loginID")
& "' AND "
    wSQL = wSQL & "          loginPassword = '" & Request.Form("pword")
& "' "
    SQL = "Select * from Login " & wSQL

```

```
RS.Open SQL,Conn
```

```
If Not RS.EOF then
```

```
  RS.Close
```

```
  s = " loginTime = '" & now() & "' "
```

```
  SQL = "UPDATE Login SET " & s & wSQL
```

```
  Conn.Execute(SQL)
```

```
  sField =          " loginID "
```

```
  sField = sField & ", loginTime "
```

```
  sValue =          " '" & Request.Form("loginID") & "' "
```

```
  sValue = sValue & ", '" & now()          & "' "
```

```
  SQL = "INSERT INTO LoginHistory (" & sField & ") VALUES (" &
sValue & ") "
```

```
  Conn.Execute(SQL)
```

```
  session("LoginID") = Request.Form("loginID")
```

```
  functionLogin = session("loginInformation")
```

```
else
```

```
  session("LoginID") = ""
```

```
  functionLogin = ""
```

```
end if
```

```
else
```

```
  Response.Write("<table align=center>")
```

```

        Response.Write(" <tr>")
        Response.Write(" <td align='right'>Login ID:</td>")
        Response.Write(" <td><input type='text' name='loginID'
value=''></td>")
        Response.Write(" </tr>")
        Response.Write(" <tr>")
        Response.Write(" <td align='right'>Password:</td>")
        Response.Write(" <td><input type='password' name='pword'
value=''></td>")
        Response.Write(" </tr>")
        Response.Write(" <tr>")
        Response.Write(" <td colspan=2 align='center'><input
type='submit' name='submitButtonLogin' value='Login'></td>")
        Response.Write(" </td>")
        Response.Write(" </tr>")
        Response.Write("</table>")
        functionLogin = ""
    end if

end function

sub subLoginInformation()

%><!-- #include file = "thesisDatabaseConnection.asp" --><%

    if(Request.Form("submitButtonChangePword") <> "") then
        errorMessageExist = FALSE
        nonErrorMessageExist = FALSE
        if(len(Request.Form("pword_New")) < 4) then
            errorMessageExist = TRUE
            errorMessage = "New Password must be at least 4 characters long"
        elseif(Request.Form("pword_New") <>
Request.Form("pword_NewVerify")) then
            errorMessageExist = TRUE
            errorMessage = "New Password and New Password Verify Do Not
Match"
        else
            wSQL = " WHERE loginID = '" & session("loginID")
& "' AND "
            wSQL = wSQL & " loginPassword = '" &
Request.Form("pword_OLD") & "' "
            SQL = "Select * from Login " & wSQL
            RS.Open SQL,Conn
            If RS.EOF then
                errorMessageExist = TRUE
                errorMessage = "Either Old Password is not correct or New
Passwords Do Not Match"
            end if
            RS.Close
        end if
        if(errorMessageExist = FALSE) then
            wSQL = " WHERE loginID = '" & session("loginID")
& "' AND "
            wSQL = wSQL & " loginPassword = '" &
Request.Form("pword_OLD") & "' "
            s = " LoginPassword = '" & Request.Form("pword_New") & "' "
            SQL = "UPDATE Login SET " & s & wSQL

```

```

Conn.Execute(SQL)
nonErrorMessageExist = TRUE
nonErrorMessage = "Password Reset"
end if

'check to see if old passwords match
'if so

end if

Response.Write("<table align=center>")
Response.Write(" <tr>")
Response.Write(" <td align='right'>Old Password:</td>")
Response.Write(" <td><input type='password' name='pword_Old'
value=''></td>")
Response.Write(" </tr>")
Response.Write(" <tr>")
Response.Write(" <td align='right'>New Password:</td>")
Response.Write(" <td><input type='password' name='pword_New'
value=''></td>")
Response.Write(" </tr>")
Response.Write(" <tr>")
Response.Write(" <td align='right'>New Password:<br><font size='-
1'>(again-verification)</font></td>")
Response.Write(" <td><input type='password' name='pword_NewVerify'
value=''></td>")
Response.Write(" </tr>")
Response.Write(" <tr>")
Response.Write(" <td colspan=2 align='center'><input type='submit'
name='submitButtonChangePword' value='Submit Change'></td>")
Response.Write(" </td>")
Response.Write(" </tr>")
if(errorMessageExist = TRUE) then
Response.Write(" <tr bgcolor = '#FFAAFF'>")
Response.Write(" <td colspan=2 align='center'>" & errorMessage &
"</td>")
Response.Write(" </td>")
Response.Write(" </tr>")
end if
if(nonErrorMessageExist = TRUE) then
Response.Write(" <tr bgcolor = '#FFFFAA'>")
Response.Write(" <td colspan=2 align='center'>" &
nonErrorMessage & "</td>")
Response.Write(" </td>")
Response.Write(" </tr>")
Response.Write(" <tr bgcolor = '#FFAAFF'>")
Response.Write(" <td colspan=2 align='center'>" &
nonErrorMessage & "</td>")
Response.Write(" </td>")
Response.Write(" </tr>")
Response.Write(" <tr bgcolor = '#AAFFFF'>")
Response.Write(" <td colspan=2 align='center'>" &
nonErrorMessage & "</td>")
Response.Write(" </td>")
Response.Write(" </tr>")
Response.Write(" <tr bgcolor = '#FFAAAA'>")

```

```

        Response.Write("    <td colspan=2 align='center'>" &
nonErrorMessage & "</td>")
        Response.Write("    </td>")
        Response.Write(" </tr>")
        Response.Write(" <tr bgcolor = '#AAFFAA'>")
        Response.Write("    <td colspan=2 align='center'>" &
nonErrorMessage & "</td>")
        Response.Write("    </td>")
        Response.Write(" </tr>")
        Response.Write(" <tr bgcolor = '#AAAAFF'>")
        Response.Write("    <td colspan=2 align='center'>" &
nonErrorMessage & "</td>")
        Response.Write("    </td>")
        Response.Write(" </tr>")
    end if
    Response.Write("</table>")

    Response.Write("<br><br>")

    SQL = " SELECT * FROM LoginHistory WHERE loginID = '" &
session("loginID") & "' ORDER BY ID DESC"
    RS.Open SQL,Conn
    Response.Write("<table align='center'>")
    Response.Write("<tr><th> Login History </th></tr>")
    Do While Not RS.EOF
        Response.Write("<tr><td>" & RS("loginTime") & "</td></tr>")
        RS.MoveNext
    Loop
    RS.Close
    Response.Write("</table>")

end sub

sub subParticipants()

    Dim Conn 'Our connection object
    Dim RS
    Dim SQL

%><!--    #include file = "thesisDatabaseConnection.asp" --><%

    SQL = " SELECT * FROM Interviewee ORDER BY ID ASC"
    RS.Open SQL,Conn
    Response.Write("<table align='center'>")
    Response.Write("<tr><th></th><th> Participants </th><th> Email
</th><th> Time </th></tr>")
    count = 0
    Do While Not RS.EOF
        count = count + 1
        Response.Write("<tr>")
        Response.Write("<td>" & count & "</td>")
        Response.Write("<td>" & RS("Name") & "</td>")
        Response.Write("<td>" & RS("EmailAddress") & "</td>")
        Response.Write("<td>" & RS("InterviewDate") & "</td>")
        Response.Write("</tr>")
        RS.MoveNext

```

```

Loop
RS.Close
Response.Write("</table>")

end sub

sub subQuestionnaireResultsEssayType()

    Dim Conn 'Our connection object
    Dim RSAuthors
    Dim SQL

    %><!--      #include file = "thesisDatabaseConnection.asp" --><%

    sSQL =          " SELECT QQ.QuestionTransferableID AS QQID, QQ.Question
AS theQuestion "
    sSQL = sSQL & ", QRM.IntervieweeID AS IntervieweeID, QQ.Active AS
ActiveQuestion "
    sSQL = sSQL & ", QRM.QuestionTransferableID AS QRMID, QRM.Response
AS theResponse "
    fSQL =          " FROM QuestionnaireQuestion QQ,
QuestionnaireResponseMemo QRM "
    wSQL =          " WHERE QQ.QuestionTransferableID =
QRM.QuestionTransferableID "
    wSQL = wSQL & " AND QQ.Active = TRUE "
    oSQL =          " ORDER BY QRM.IntervieweeID, QQ.QuestionTransferableID
"

    SQL = sSQL & fSQL & wSQL & oSQL

'Response.Write("<br>" & SQL)
RS.Open SQL, Conn
count = 1
theIntervieweeID_OLD = -1
Do While Not RS.EOF
    theIntervieweeID = RS("IntervieweeID")
    session("theInterviewee" & count) = theIntervieweeID
    if((theIntervieweeID_OLD+0 <> theIntervieweeID+0) AND (1 <
count+0))then
        Response.Write("</table></td></tr></table>")
    end if
    if(theIntervieweeID_OLD+0 <> theIntervieweeID+0) then
        Response.Write("<br><table border=1 width=800
align=center><tr><td><table>")
        Response.write("<tr><td align=center>Interviewee ID: " &
session("theInterviewee" & count) & "</td></tr>")
        count = count + 1
    else
        Response.Write("<tr><td>&nbsp;&lt;/td></tr>")
    end if
    Response.Write(" <tr><td>")
    Response.Write("    Question: " & RS("theQuestion") & "<br>")
    Response.Write("    Response: " & RS("theResponse"))
    Response.Write(" </td></tr>")
    theIntervieweeID_OLD = theIntervieweeID+0

```

```

        RS.MoveNext
    Loop
    RS.Close
    if(1 < count+0)then
        Response.Write("</table></td></tr></table>")
    end if

    Response.Write("<br><br>Number of Respondents: " & count-1)

end sub

sub subQuestionnaireResultsMultipleChoice()

    Dim Conn 'Our connection object
    Dim RSAuthors
    Dim SQL

%><!--      #include file = "thesisDatabaseConnection.asp" --><%

    SQL = " Select * from QuestionnaireQuestion WHERE Active = TRUE AND
QuestionType LIKE 'radio%' ORDER BY QuestionTransferableID "
    RS.Open SQL, Conn
    Response.Write("<br><table border=1 width=800
align='center'><tr><td><table><tr>")
    Response.Write("<th>ID</th>")
    columnCount = 0
    Do While Not RS.EOF
        columnCount = columnCount + 1
        tempQTID = RS("QuestionTransferableID")
        session("column" & columnCount) = tempQTID
        if(count MOD 3 = 0) then bgcolor = "bgcolor = '#EEEEEE'" else
bgcolor = "" end if
        Response.Write("<th " & bgcolor & ">QTID." & tempQTID & "</th>")
        RS.MoveNext
    Loop
    RS.Close

    for RowCount = 1 to 5
        for count = 1 to columnCount
'NumberOfColumnXAnsweringY
            session("NumberOfColumn" & session("column" & count) &
"Answering" & RowCount) = 0
        next
    next

    sSQL =          " SELECT QQ.QuestionTransferableID AS QQID, QQ.Question
AS theQuestion "
    sSQL = sSQL & ", QRMC.IntervieweeID AS IntervieweeID, QQ.Active AS
ActiveQuestion "
    sSQL = sSQL & ", QRMC.QuestionTransferableID AS QRMCID,
QRMC.Response AS theResponse "
    fSQL =          " FROM QuestionnaireQuestion QQ,
QuestionnaireResponseMultipleChoice QRMC "
    wSQL =          " WHERE QQ.QuestionTransferableID =
QRMC.QuestionTransferableID "
    wSQL = wSQL & " AND QQ.Active = TRUE "

```

```

oSQL = " ORDER BY QRMC.IntervieweeID,
QQ.QuestionTransferableID "

SQL = sSQL & fSQL & wSQL & oSQL

'Response.Write("<br>" & SQL)
RS.Open SQL, Conn
theIntervieweeID_OLD = -1
RS.MoveFirst
countRespondents = 0
Do While Not RS.EOF
theIntervieweeID = RS("IntervieweeID")
if(theIntervieweeID_OLD <> theIntervieweeID+0) then
countRespondents = countRespondents + 1
session("theInterviewID" & countRespondents) = theIntervieweeID
end if
if((theIntervieweeID_OLD <> theIntervieweeID+0) AND (1 <
countRespondents+0))then
Response.Write("<tr>")
Response.Write(" <td>" & session("theInterviewID" &
countRespondents-1) & "</td>")
for count = 1 to columnCount
if((count MOD 3 = 0) OR ((countRespondents-1) MOD 3 = 0)) then
bgcolor = "bgcolor = '#EEEEEE' "
else
bgcolor = ""
end if

if(trim(session("columnCode" & session("column" & count))) <>
"") then
session("NumberOfColumn" & session("column" & count) &
"Answering" & trim(session("columnCode" & session("column" & count))))
= session("NumberOfColumn" & session("column" & count) & "Answering" &
trim(session("columnCode" & session("column" & count)))) + 1
end if

Response.Write("<td align=center " & bgcolor & ">" &
session("columnCode" & session("column" & count)) & "</td>")
next
Response.Write("</tr>")
end if
tempQTID = RS("QRMCID")
session("columnCode" & tempQTID) = RS("theResponse")
theIntervieweeID_OLD = theIntervieweeID+0
RS.MoveNext
Loop
Response.Write("<tr>")
Response.Write(" <td>" & theIntervieweeID & "</td>")
for count = 1 to columnCount
if((count MOD 3 = 0) OR (countRespondents MOD 3 = 0)) then
bgcolor = "bgcolor = '#EEEEEE' "
else
bgcolor = ""
end if

if(trim(session("columnCode" & session("column" & count))) <> "")
then

```



```

RS.Open SQL, Conn
Response.Write("<br><table border=1 width=800
align='center'><tr><td><table><tr>")
Response.Write("<tr><td><h2><center>Questions
[Legend]</center</h2></td></tr>")
Response.Write("<tr><td>&nbsp;</td></tr>")
columnCount = 0
Do While Not RS.EOF
tempQTID = RS("QuestionTransferableID")
tempPurpose = RS("QuestionPurpose")
tempType = RS("QuestionType")
tempQuestion = RS("Question")
tempResponse1 = RS("PossibleResponse1")
tempResponse2 = RS("PossibleResponse2")
tempResponse3 = RS("PossibleResponse3")
tempResponse4 = RS("PossibleResponse4")
tempResponse5 = RS("PossibleResponse5")
Response.Write("<tr>")
Response.Write(" <td valign = top><ul>")
Response.Write(" <li>Transferable ID: QTID." & tempQTID)
Response.Write(" <li>Purpose: " & tempPurpose)
Response.Write(" <li>Type: " & tempType)
' Response.Write(" </td>")
' Response.Write(" <td valign=top>")
Response.Write(" <li>Question: " & tempQuestion)
Response.Write(" <li>(Matrix Value: 1) Possible Response 1: " &
tempResponse1)
Response.Write(" <li>(Matrix Value: 2) Possible Response 2: " &
tempResponse2)
if(tempType = "radio5") then
Response.Write(" <li>(Matrix Value: 3) Possible Response 3: " &
tempResponse3)
Response.Write(" <li>(Matrix Value: 4) Possible Response 4: " &
tempResponse4)
Response.Write(" <li>(Matrix Value: 5) Possible Response 5: " &
tempResponse5)
end if
Response.Write(" </ul></td>")
Response.Write("</tr><tr><td>&nbsp;</td></tr>")
RS.MoveNext
Loop
RS.Close

Response.Write("<table>")

end sub
%>

```


Appendix D.3 - subroutinesAndFunctions.asp

```
<%  
  
function convertToDatabaseSingleQuotes(tempResponse)  
  if(InStr(tempResponse, "'") <> 0) then ' single quotes cannot go  
  directly into the database without having two single quotes  
    temp = tempResponse  
    tempNew = ""  
    tempCount = 0  
    while (InStr(temp, "'") <> 0) and tempCount < 100 ' tempCount is  
  for runaways  
      tempNew = tempNew & mid(temp,1,InStr(temp, "'"))  
      temp = mid(temp,InStr(temp, "'")+1,len(temp) - InStr(temp, "'"))  
      tempNew = tempNew & "'"  
      tempCount = tempCount + 1  
    wend  
    tempResponse = tempNew & temp  
  end if  
  convertToDatabaseSingleQuotes = tempResponse  
end function  
  
function dateProperDisplay(dateTimeInput)  
  theYear = mid(dateTimeInput,1,4)  
  theMonth = mid(dateTimeInput,5,2)  
  theMonth = getMonthName(theMonth)  
  theDay = mid(dateTimeInput,7,2)  
  
  if(8 < len(dateTimeInput)) then  
    theHour = mid(dateTimeInput,9,2)  
    theMinute = mid(dateTimeInput,11,2)  
    theTime = getAmPm(theHour, theMinute)  
  end if  
  
  dateOutput = "" & theYear  
  dateOutput = dateOutput & " " & theMonth  
  dateOutput = dateOutput & " " & theDay  
  if(8 < len(dateTimeInput)) then  
    dateOutput = dateOutput & " " & theTime  
  end if  
  dateProperDisplay = dateOutput  
end function  
  
sub displayExercise(currentExerciseNumber)  
  
%>  
  
<tr>  
  
<input type='hidden' name='currentExerciseNumber'  
value='<%=currentExerciseNumber%>'>  
<table border=0 valign=top width=1000>  
<tr>  
<th valign=top align=center>  
  <%if(session("responseSortMethodology" &  
session("exerciseDisplayOrder" & currentExerciseNumber)) = 1) then %>
```



```

'Response.Write("<br> currentExercise#: datetime = " &
session("theDateTime" & currentExerciseNumber))
'Response.Write("<br> exerciseDisplayOrder#: name = " &
session("theName" & session("exerciseDisplayOrder" &
currentExerciseNumber)))
'Response.Write("<br> exerciseDisplayOrder#: subject = " &
session("theSubject" & session("exerciseDisplayOrder" &
currentExerciseNumber)))
'Response.Write("<br> exerciseDisplayOrder#: name = " &
session("theDateTime" & session("exerciseDisplayOrder" &
currentExerciseNumber)))
'Response.Write("<br><br><br>")
    Response.Write("Please select ALL 'emails' where a minimum of the
following information exists:")
    Response.Write("<br>(Click the submit button at the bottom of the
listing to continue.)<br>")
    if(session("responseType" & session("exerciseDisplayOrder" &
currentExerciseNumber))<> "Name") then
        Response.Write("<br>NAME: " & session("theName" &
session("exerciseDisplayOrder" & currentExerciseNumber)))
    end if
'Dad vs Dad [Home]... we need to consider multiples of name

    if(session("responseType" & session("exerciseDisplayOrder" &
currentExerciseNumber))<> "Subject") then
        Response.Write("<br>SUBJECT: " & session("theSubject" &
session("exerciseDisplayOrder" & currentExerciseNumber)))
    end if
    if(session("responseType" & session("exerciseDisplayOrder" &
currentExerciseNumber))<> "DateTime") then
        Response.Write("<br>DATE/TIME: " &
dateProperDisplay(session("theDateTime" &
session("exerciseDisplayOrder" & currentExerciseNumber))))
    end if

'MORE explanation is needed, e.g. jobs: option the following fit
jobs: option 'car washer'
    Response.Write("<br><br><br><br><br><br><br><br>")
    Response.Write("<br><br><br><br><br><br><br><br>")
next
end sub

sub displayIntroduction()

    if (instr(Request.form("firstPageExampleListing"),"multiple") <> 0)
then
        session("firstPageExampleListing") = "multiple"
    elseif (instr(Request.form("firstPageExampleListing"),"standard") <>
0) then
        session("firstPageExampleListing") = "standard"
    elseif (instr(Request.form("firstPageExampleListing"),"both") <> 0)
then
        session("firstPageExampleListing") = "both"
    elseif (session("firstPageExampleListing") = "") then
        session("firstPageExampleListing") = "standard"
    end if

```

```

pract1 =          "<table><tr><td> "
pract1 = pract1 & "<center><b>FOR PRACTICE</b></center> "
pract1 = pract1 & "<br>Given -- "
pract1 = pract1 & "<br>Date: '2003 Jan 01'"
pract1 = pract1 & "<br>Subject: 'opening in: Marketing'"
pract1 = pract1 & "<br>What <b>Names</b> correspond with this?"
pract1 = pract1 & "<br>If you found 'Cisco Jobs', you are correct."
pract1 = pract1 & "<br>"
pract1 = pract1 & "<br>Given -- "
pract1 = pract1 & "<br>Name: 'YourMagazine.net' "
pract1 = pract1 & "<br>Date: '2003 Sep 25' "
pract1 = pract1 & "<br>What <b>Subjects</b> correspond with this? "
pract1 = pract1 & "<br>If you found 'Your subscription ends in...',"
you are correct. "
pract1 = pract1 & "<br> "
pract1 = pract1 & "<br>Given -- "
pract1 = pract1 & "<br>Subject: 'opening in:' "
pract1 = pract1 & "<br>Name: 'Novell Opportunity' "
pract1 = pract1 & "<br>What <b>Dates</b> correspond with this? "
pract1 = pract1 & "<br>If you found '2003 Jul 01', '2003 Jul 01',"
'2003 Jul 02',<br> "
pract1 = pract1 & "'2003 Jul 02', '2003 Jul 04', '2003 Jul 04', '2003
Jul 05',<br> "
pract1 = pract1 & "'2003 Jul 12', '2003 Jul 27', '2003 Aug 07', '2003
Sep 25',<br> "
pract1 = pract1 & " <b>and</b> '2003 Sep 30', you are correct. "
pract1 = pract1 & "</td></tr></table> "

pract2 =          "<table><tr><td>"
pract2 = pract2 & "<center><b>FOR PRACTICE</b></center>"
pract2 = pract2 & "<br>Given -- "
pract2 = pract2 & "<br>Subject: 'Able to go,' "
pract2 = pract2 & "<br>Name: 'Nathan Breker' "
pract2 = pract2 & "<br>What <b>Dates</b> correspond with this? "
pract2 = pract2 & "<br>If you found '2003 Apr 15' <b>and</b> '2003
Oct 15', you are correct. "
pract2 = pract2 & "<br> "
pract2 = pract2 & "<br>Given -- "
pract2 = pract2 & "<br>Name: 'Jobs R Us' "
pract2 = pract2 & "<br>Date: '2003 Jan 02' "
pract2 = pract2 & "<br>What <b>Subjects</b> correspond with this? "
pract2 = pract2 & "<br>If you found 'jobs: clerical typist' and
'jobs: newspaper editor', you are correct. "
pract2 = pract2 & "<br> "
pract2 = pract2 & "<br>Given -- "
pract2 = pract2 & "<br>Date: '2003 Oct 15' "
pract2 = pract2 & "<br>Subject: 'benching these days...' "
pract2 = pract2 & "<br>What <b>Names</b> correspond with this? "
pract2 = pract2 & "<br>If you found 'Reese Aren', you are correct. "
pract2 = pract2 & "</td></tr></table> "

```

%>

```

<table border=0 valign=top width=1000>
<tr>

```

```
<td align=center valign=top colspan=3>
```

This web-based research has the following small sections:

```
<table border=1><tr><td>
<table>
  <tr>
    <th valign=bottom>Section</th>
    <th valign=bottom>Topic</th>
    <th>Approximate Time Commitment (minutes)</th>
  </tr>
  <tr>
    <td align=center valign=top>1)</td>
    <td>Introduction with an explanation of the two types of sorting.
(This section)
    </td>
    <td align=center>5</td>
  </tr>
  <tr>
    <td align=center>2)</td>
    <td>Quick exercises on both versions of the sorting.</td>
    <td align=center>12</td>
  </tr>
  <tr>
    <td align=center>3)</td>
    <td>A short questionnaire</td>
    <td align=center>4</td>
  </tr>
  <tr>
    <td align=center>4)</td>
    <td>Quick thank you for your participation.</td>
    <td align=center>1</td>
  </tr>
  <tr>
    <td></td>
    <th>TOTAL</th>
    <th align=center>22</th>
  </tr>
</table>
</td></tr></table>
```

```
<br>
</td>
</tr>
```

```
<tr>
<td valign=top colspan=3>
```

```
<center>
<h1>INTRODUCTION</h1>
</center>
```

```
<% if (session("firstPageExampleListing") = "standard") then %>
<p>
First of all, thank you for participating in this research project.
</p>
```

<p>The general purpose of this research is to compare a computer user's response, feelings, and concerns to two different column sorting methods.
</p>

<p>Examples of sorting occur when alphabetizing email lists. In an attempt to find a specific received email, a user might recall who sent it and then click on the "Name" column to sort alphabetically by the sender's "Name". On the other hand, a user might recall what day it was sent and then click on the "Date" column to sort alphabetically by the "Date" the email was received. Sorting on the Subject or other columns (depending on the email program) might allow sorted listings based upon the respective column.
</p>

<p>The specific purpose of the research you are assisting is to determine the effectiveness of sorting methods that would help pin-point one email among many that has either been sent by a specific user account, and/or within a specific time frame, and/or with a similar subjectline, and/or other characteristics that might be influenced by the type of sorting.
</p>
<% end if %>

<% if (session("firstPageExampleListing") = "both") then %>
<p>Consider the two displays below representative of incoming emails. The one on the left represents a common method for displaying email. That is, sorting is done on one column. By selecting sorting on another column, alphabetization is done on the chosen column. Influence on alphabetization is limited to the selected column. Please click on the button above the column headers to see the changes that occur.
</p>

<p>A different method for displaying a list is by prioritized selection (note the listing on the right half of the page) in which sorting occurs on each column with the left-most column taking top-precedence, the second column taking secondary precedence, and so on with the last column having the least amount of precedence. Please click on the column headers.
</p>
<% end if %>

<% if (session("firstPageExampleListing") = "standard") then %>
<p>Consider the display below which is representative of incoming emails. It represents a common method or standard for displaying email. That is, sorting is constrained to the listing of a single column. By selecting sorting on another column, alphabetization is done on the items of that selected column. Influence on alphabetization is

limited to the selected column. Please click on the button above the column headers to see the changes that occur.

```
</p>
```

```
<% end if %>
```

```
<% if (session("firstPageExampleListing") = "multiple") then %>
```

```
<p>Consider the display below which uses a non-standard method for displaying a listing of emails. The records are displayed by prioritized selection (note the listing on the right half of the page) in which sorting occurs on each
```

```
column with the left-most column taking top-precedence, the second column taking secondary precedence, and so on
```

```
with the last column having the least amount of precedence. Please click on the column headers and note the change that occurs.
```

```
(Entire columns of data are transposed, that is, the contents of two side-by-side columns are exchanged.)
```

```
</p>
```

```
<% end if %>
```

```
<p>
```

```
<% if ((session("firstPageExampleListing") = "standard") OR (session("firstPageExampleListing") = "multiple")) then %>
```

```
<!--Try the practice examples (only available when viewing one type of sort at a time only). -->
```

```
Try the practice examples then click continue at the bottom of the page.
```

```
<% end if %>
```

```
</p>
```

```
<% if (session("firstPageExampleListing") = "both") then %>
```

```
<!--p>
```

```
Once you have found the desired records, please read the information at the bottom of this page.
```

```
</p-->
```

```
<p>
```

```
Compare the two methods as they sit side-by-side. Click the buttons on the top of each column. Once you are done, go to the bottom of the page and click continue.
```

```
</p>
```

```
<% end if %>
```

```
<br>
```

```
<% if(Request.Form("firstPageExampleListing") = "") then %>
```

```
<br><a name="stopHere"></a>
```

```
<% end if %>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<th valign=top align=center width=500>
```

```
<% if ((session("firstPageExampleListing") = "standard") OR (session("firstPageExampleListing") = "both")) then %>
```

```
Methodology<br>
```

```
Standard Sort View
```

```
<% else %>
```



```

<td valign=top align=center>
  <% if ((session("firstPageExampleListing") = "multiple") OR
(session("firstPageExampleListing") = "both")) then %>
    <% displayListingNewFormat("no submit features") %>
  <% elseif (session("firstPageExampleListing") = "standard") then
    revolutionsTemp = 14
    for count = 1 to revolutionsTemp
      Response.write(pract1)
      if (count+0 < revolutionsTemp+0) then

Response.Write("<br><br><br><br><br><br><br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br><br><br><br><br><br><br>")
      end if
    next
  end if %>
</td>
</tr>
<tr>
<td valign=top colspan=3>
<br>
<% if (1=2) then %>
<p>
At this point, you should be familiar with both sorting methods. If
not, please return to the listings and click on the
column header buttons to note the functionality. A short exercise of
<%=session("setsToDisplayPerSortType")*3*2%> questions follows in which
you will be
asked to identify a series of records in an environment similar to
those found in an email listing.
</p>
<% end if %>

  <% if (session("firstPageExampleListing") = "both") then %>
<p>
Please note that part of the research monitors the amount of time taken
to respond to each question. So, please do not rush unnecessarily
through the
exercises, but also please do not unnecessarily delay or take breaks
during the exercises.
</p>
<% end if %>
<br>

<center>
  <% if (session("firstPageExampleListing") = "multiple") then %>
<!--input type='submit' name='firstPageExampleListing' value='click to
view standard column sort only'-->
  <% end if %>

  <% if (session("firstPageExampleListing") = "multiple") then %>
<!--input type='submit' name='firstPageExampleListing' value='click to
view both column sort method examples simultaneously'-->
  <% end if %>

  <% if (session("firstPageExampleListing") = "standard") then %>

```

```

<!--input type='submit' name='firstPageExampleListing' value='click to
view prioritized multiple column sort only'-->


```

```
function displayListingNewFormat(showSubmitFeatures)
```

```
' Response.Write(Request.Form())
```

```
Dim Conn 'Our connection object
Dim RSAuthors
Dim SQL
```

```

    if session("Column1Header") = "" then session("Column1Header") =
"DateTime" end if
    if session("Column1AscDesc") = "" then session("Column1AscDesc") =
"Desc" end if
    if session("Column2Header") = "" then session("Column2Header") =
"Name" end if
    if session("Column2AscDesc") = "" then session("Column2AscDesc") =
"Asc" end if
    if session("Column3Header") = "" then session("Column3Header") =
"Subject" end if
    if session("Column3AscDesc") = "" then session("Column3AscDesc") =
"Asc" end if

```

```

if(request.form("swap12") <> "") then
    temp = session("Column1Header")
    session("Column1Header") = session("Column2Header")
    session("Column2Header") = temp
    temp = session("Column1AscDesc")
    session("Column1AscDesc") = session("Column2AscDesc")
    session("Column2AscDesc") = temp
end if

```

```

if(request.form("swap23") <> "") then
    temp = session("Column2Header")
    session("Column2Header") = session("Column3Header")
    session("Column3Header") = temp
    temp = session("Column2AscDesc")
    session("Column2AscDesc") = session("Column3AscDesc")
    session("Column3AscDesc") = temp
end if

for count = 1 to 3
    if(request.form("sortdown" & count) <> "") then
        session("Column" & count & "AscDesc") = "Desc"
    end if
    if(request.form("sortup" & count) <> "") then
        session("Column" & count & "AscDesc") = "Asc"
    end if
next

%><!--      #include file = "thesisDatabaseConnection.asp" --><%

    oSQL = session("Column1Header") & " " & session("Column1AscDesc") &
    ", "
    oSQL = oSQL & session("Column2Header") & " " &
    session("Column2AscDesc") & ", "
    oSQL = oSQL & session("Column3Header") & " " &
    session("Column3AscDesc")

    SQL = "Select * from ArrayItems ORDER BY " & oSQL

    RS.Open SQL,Conn

    Response.Write("<table valign=top>")

    Response.Write "<tr>"
'   if(showSubmitFeatures = "") then
        Response.Write "<th></th>"
'   end if
    Response.Write "<td align=center>"
    if(session("Column1AscDesc") = "Asc") then
        Response.Write "<input type='submit' name='sortdown1' value='v'>"
    else
        Response.Write "<input type='submit' name='sortup1' value='^'>"
    end if
    Response.Write "<input type='submit' name='swap12' value='>'>"
    Response.Write "</td>"
    Response.Write "<td align=center>"
    Response.Write "<input type='submit' name='swap12' value='<'>"
    if(session("Column2AscDesc") = "Asc") then
        Response.Write "<input type='submit' name='sortdown2' value='v'>"
    else
        Response.Write "<input type='submit' name='sortup2' value='^'>"
    end if
    Response.Write "<input type='submit' name='swap23' value='>'>"
    Response.Write "</td>"
    Response.Write "<td align=center>"
    Response.Write "<input type='submit' name='swap23' value='<'>"
    if(session("Column3AscDesc") = "Asc") then

```

```

        Response.Write "<input type='submit' name='sortdown3' value='v'>"
    else
        Response.Write "<input type='submit' name='sortup3' value='^'>"
    end if
    Response.Write "</td>"
    Response.Write "</tr>"

    Response.Write "<tr>"
'   if(showSubmitFeatures = "") then
        Response.Write "<th></th>"
'   end if
    Response.Write "<th><font size=-1>" & session("Column1Header") &
"</font></th>"
    Response.Write "<th><font size=-1>" & session("Column2Header") &
"</font></th>"
    Response.Write "<th><font size=-1>" & session("Column3Header") &
"</font></th>"
    Response.Write "</tr>"

Do While Not RS.EOF
    Response.Write "<tr>"
    theID = RS("ID")
    theReply = "theReply=" & theID & "a"
    if(showSubmitFeatures = "") then
        theReplyName = "theReply"
    else
        theReplyName = ""
    end if
    if(InStr(Request.Form(), theReply) AND NOT
Request.Form("exerciseReplySubmitButton") <> "") then
        Response.Write "<td><input type=checkbox name='" & theReplyName &
"' value = '" & theID & "a' checked></font></td>"
    else
        Response.Write "<td><input type=checkbox name='" & theReplyName &
"' value = '" & theID & "a'></font></td>"
    end if

    session("tempCol1") = RS(session("Column1Header"))
    session("tempCol2") = RS(session("Column2Header"))
    session("tempCol3") = RS(session("Column3Header"))

    for count = 1 to 3
        if(session("Column" & count & "Header") = "DateTime") then
            session("tempCol" & count) =
dateProperDisplay(session("tempCol" & count))
        end if
    next

    Response.Write "<td><font size=-1>" & session("tempCol1") &
"</font></td>"
    Response.Write "<td><font size=-1>" & session("tempCol2") &
"</font></td>"
    Response.Write "<td><font size=-1>" & session("tempCol3") &
"</font></td>"
    Response.Write "</tr>"
    RS.MoveNext
Loop

```

```

Response.Write("</table>")

if(showSubmitFeatures = "") then
    Response.Write("<center><input type='submit'
name='exerciseReplySubmitButton' value='submit Answer'></center>")
end if
RS.Close
Conn.Close

displayListing = ""
end function

function displayListingOldFormat(showSubmitFeatures)
' Response.Write(Request.Form())

Dim Conn 'Our connection object
Dim RSAuthors
Dim SQL

Dim NSD(3)

NSD(1) = "Name"
NSD(2) = "Subject"
NSD(3) = "DateTime"

for count = 1 to 3
    if(request.form("sort" & count) <> "") then
        session("chosenColumn") = NSD(count)
        if(request.form("sort" & count) = "v") then
            session("chosenColumnDirection") = "Desc"
        else
            session("chosenColumnDirection") = "Asc"
        end if
    else
        session("Column" & count & "AscDescOld") = "."
    end if
next

if(session("chosenColumn") = "") then
    session("chosenColumn") = NSD(3)
    session("chosenColumnDirection") = "Desc"
end if

%><!--      #include file = "thesisDatabaseConnection.asp" --><%

    oSQL = session("chosenColumn") & " " &
session("chosenColumnDirection")

    SQL = "Select * from ArrayItems ORDER BY " & oSQL & ", scrambleSort
Asc"

    RS.Open SQL,Conn

    Response.Write("<table valign=top>")

    Response.Write "<tr>"

```

```

' if(showSubmitFeatures = "") then
    Response.Write "<th></th>"
' end if
for count = 1 to 3
    Response.Write "<td align=center>"
    if(session("chosenColumn") = NSD(count)) then
        if(session("chosenColumnDirection") = "Asc") then
            Response.Write "<input type='submit' name='sort" & count & "'
value='v'>"
        else
            Response.Write "<input type='submit' name='sort" & count & "'
value='^'>"
        end if
    else
        Response.Write "<input type='submit' name='sort" & count & "'
value='.'>"
    end if
    Response.Write "</td>"
next
Response.Write "</tr>"

Response.Write "<tr>"
' if(showSubmitFeatures = "") then
    Response.Write "<th></th>"
' end if
Response.Write "<th><font size=-1>Name</font></th>"
Response.Write "<th><font size=-1>Subject</font></th>"
Response.Write "<th><font size=-1>DateTime</font></th>"
Response.Write "</tr>"

Do While Not RS.EOF
    Response.Write "<tr>"
    theID = RS("ID")
    theReply = "theReply=" & theID & "a"
    if(showSubmitFeatures = "") then
        theReplyName = "theReply"
    else
        theReplyName = ""
    end if
    if(InStr(Request.Form(), theReply) AND NOT
Request.Form("exerciseReplySubmitButton") <> "") then
        Response.Write "<td><input type=checkbox name='" & theReplyName &
'' value = '" & theID & "a' checked></font></td>"
    else
        Response.Write "<td><input type=checkbox name='" & theReplyName &
'' value = '" & theID & "a'></font></td>"
    end if
    Response.Write "<td><font size=-1>" & RS("Name") & "</font></td>"
    Response.Write "<td><font size=-1>" & RS("Subject") &
"</font></td>"
    Response.Write "<td><font size=-1>" &
dateProperDisplay(RS("DateTime")) & "</font></td>"
    Response.Write "</tr>"
    RS.MoveNext
Loop
Response.Write("</table>")

```

```

    if(showSubmitFeatures = "") then
        Response.Write("<center><input type='submit'
name='exerciseReplySubmitButton' value='submit Answer'></center>")
    end if

    RS.Close
    Conn.Close

    displayListing = ""
end function

sub displayQuestionnaire()
    initializeQuestionnaire()

%>
<table border=0 width=600>
<tr><td>
<center><h1>QUESTIONNAIRE</h1>
<p><h3>Referring to the exercises you just completed, please answer the
following:</h3>
<font color="blue">A review of the sorting methods is found at the
bottom of this page.</font>
</center>
<%

    if ((Request.Form("submitButtonQuestionnaireComplete")<>"") AND
(session("introductionExercisesQuestionnaireOrCoupon")="Questionnaire")
) then
        response.write("<font color='FF0000'><b><center>One or more
multiple-choice questions were not answered. <br>Please review
highlighted question(s) below.</center></b></font>")
    end if

'session("exerciseDisplayOrder" & count)

    for count = 1 to session("NumberOfQuestionnaireQuestions")
        session("questionTransferableIDrelationship" &
session("questionnaireDisplayOrder" & count)) =
session("QuestionTransferableID" & session("questionnaireDisplayOrder"
& count))

        if(InStr(session("radioButtonQuestionsMissed"),"#" &
session("questionnaireDisplayOrder" & count) & "#")<>0) then
            Response.Write("<p><b><font color='FF0000'>" & count & ". " &
session("Question" & session("questionnaireDisplayOrder" & count)) & "
</font></b>")
        else
            Response.Write("<p>" & count & ". " & session("Question" &
session("questionnaireDisplayOrder" & count)))
        end if

        if(InStr(session("QuestionType" &
session("questionnaireDisplayOrder" & count)), "radio")<>0) Then

            if(CStr(Request.Form("response" &
session("questionnaireDisplayOrder" & count)) = "1")) then checked =
"checked" else checked = "" end if

```

```

Response.Write("<br><input type='radio' name='response' &
session("questionnaireDisplayOrder" & count) & "' value='1' " & checked
& "> " & session("PossibleResponse1" &
session("questionnaireDisplayOrder" & count)))

```

```

if(CStr(Request.Form("response" &
session("questionnaireDisplayOrder" & count)) = "2")) then checked =
"checked" else checked = "" end if
Response.Write("<br><input type='radio' name='response' &
session("questionnaireDisplayOrder" & count) & "' value='2' " & checked
& "> " & session("PossibleResponse2" &
session("questionnaireDisplayOrder" & count)))

```

```

if(session("QuestionType" & session("questionnaireDisplayOrder" &
count)) = "radio5") then

```

```

if(CStr(Request.Form("response" &
session("questionnaireDisplayOrder" & count)) = "3")) then checked =
"checked" else checked = "" end if
Response.Write("<br><input type='radio' name='response' &
session("questionnaireDisplayOrder" & count) & "' value='3' " & checked
& "> " & session("PossibleResponse3" &
session("questionnaireDisplayOrder" & count)))

```

```

if(session("PossibleResponse4" &
session("questionnaireDisplayOrder" & count)) <>"") then

```

```

if(CStr(Request.Form("response" &
session("questionnaireDisplayOrder" & count)) = "4")) then checked =
"checked" else checked = "" end if
Response.Write("<br><input type='radio' name='response' &
session("questionnaireDisplayOrder" & count) & "' value='4' " & checked
& "> " & session("PossibleResponse4" &
session("questionnaireDisplayOrder" & count)))

```

```

if(session("PossibleResponse5" &
session("questionnaireDisplayOrder" & count)) <>"") then
if(CStr(Request.Form("response" &
session("questionnaireDisplayOrder" & count)) = "5")) then checked =
"checked" else checked = "" end if
Response.Write("<br><input type='radio' name='response' &
session("questionnaireDisplayOrder" & count) & "' value='5' " & checked
& "> " & session("PossibleResponse5" &
session("questionnaireDisplayOrder" & count)))

```

```

end if
end if
end if
else
Response.Write("<br><textarea name='response' &
session("questionnaireDisplayOrder" & count) & "' rows='8'
cols='80'></textarea>")

```

```

end if
next
%>
</td></tr>
<tr><td>
<center><br>

```



```

        <th width=20></th>
        <th width=150><font size=-1>Item</font></th>
        <th width=120><font size=-1>Sponsor</font></th>
        <th width=150><font size=-1>Valid Locations</font></th>
    </tr>
<%
    SQL = "Select * from Coupon Where Active = True AND NumberRemaining >
0 ORDER BY ID "

    RS.Open SQL,Conn

    count = 1
    Do While Not RS.EOF
        if(count = 1) then checked = "checked" else checked = "" end if
        Response.Write "<tr>"
        Response.Write "<td><input type=radio name='couponCouponID' value =
'" & RS("ID") & "' " & checked & "></td>"
        Response.Write "<td><font size=-1>" & RS("Item") & "</font></td>"
        Response.Write "<td><font size=-1>" & RS("Origin") & "</font></td>"
        Response.Write "<td><font size=-1>" & RS("ValidLocations") &
"</font></td>"
        Response.Write "</tr>"
        RS.MoveNext
        count = count + 1
    Loop
    RS.Close
    Conn.Close
%>
</table>
</td></tr>
<tr><td>
<br><center><b>Mailing Info</b><br>
<font size='-1'>(Name, Address, City, State, Zip code) <br> (As would
be found on an envelope.)<br></font>
<textarea cols=40 rows=4 name=couponMailingAddress></textarea>
<br><b>Email Address:</b> <input type='text' name='couponEmailAddress'
value=''><br>
<font size='-1'>(Email Address: For quick contact regarding gift
certificate)</font>
</center>
</td></tr>
<%
end if
%>

<tr><td>
<br><br><br><br>
If there are any questions regarding your responses to this
questionnaire, would you agree to further contact?
If so, please enter your contact information here:
<center><table>
<tr><td><b>Name:</b></td><td><input type='text' name='intervieweeName'
value=''></td></tr>
<tr><td><b>Email Address:</b></td><td><input type='text'
name='intervieweeEmailAddress' value=''></td></tr>
</table></center>
</td></tr>

```

```

<tr><td><br><br><br><br><center>
<% if displayTheCoupon = "Yes" then %>

<input type='submit' name='submitButtonCouponRequestComplete'
value='Get Gift' width='35'>
<% else %>
<input type='submit' name='submitButtonCouponRequestComplete'
value='Submit' width='35'>
<% end if %>
</center></td></tr>
</table>

```

```

<%
IwantToShowThis = "No"
If (IwantToShowThis = "Yes") Then
%>
<br>Name: <input type='text' name='' value=''>
<br>email address:<input type='text' name='' value=''>

```

To get extra, please forward this onto someone you know and have them put you in as a referrer.

Make sure to include that the research will be conducted through a specific date so that people won't be trying to get a coupon after a certain date.

```

<input type='submit' value='email to friend' id='submit'1
name='submit'1>

```

Note to self:

Questionnaire: to demonstrate controlled environment, did someone assist you with starting this project or did someone just email you this link and you are trying this for the fun of it?

Your responses will not be associated to you unless you would like them to be.

```

<%
End if
%>

```

```

<%
end sub

```

```

function getAmPm(hourIn, minuteIn)
  if(hourIn+0 <= 12) then timeOut = (hourIn + 0) & ":" & minuteIn & "
am" end if
  if(12 < hourIn+0) then timeOut = (hourIn - 12) & ":" & minuteIn & "
pm" end if
  getAmPm = timeOut
end function

```

```

function getMonthName(monthIn)
  if(monthIn+0 = 1) then monthOut = "Jan" end if

```

```

if(monthIn+0 = 2) then monthOut = "Feb" end if
if(monthIn+0 = 3) then monthOut = "Mar" end if
if(monthIn+0 = 4) then monthOut = "Apr" end if
if(monthIn+0 = 5) then monthOut = "May" end if
if(monthIn+0 = 6) then monthOut = "Jun" end if
if(monthIn+0 = 7) then monthOut = "Jul" end if
if(monthIn+0 = 8) then monthOut = "Aug" end if
if(monthIn+0 = 9) then monthOut = "Sep" end if
if(monthIn+0 = 10) then monthOut = "Oct" end if
if(monthIn+0 = 11) then monthOut = "Nov" end if
if(monthIn+0 = 12) then monthOut = "Dec" end if
getMonthName = monthOut
end function

function getPartialSubject(incomingSubject)
theLength = len(incomingSubject)
if(theLength+0 < 10) then
theValue = incomingSubject
else
thePhrase = ""
theSpace = 0
for getPSCount = 1 to theLength
theChar = mid(incomingSubject,getPSCount,1)
thePhrase = thePhrase & theChar
if(theChar = " ") then theSpace = theSpace + 1 end if
if(2 <= theSpace+0) then getPSCount = theLength end if
next
theValue = thePhrase
end if
getPartialSubject = theValue
end function

sub initializeFields()

Dim Conn 'Our connection object
Dim RS
Dim SQL

%><!-- #include file = "thesisDatabaseConnection.asp" --><%

Randomize()
theNumber = ((2*Rnd) \ 1)
for count = 1 to session("setsToDisplayPerSortType")*3*2
if(theNumber=1) then theNumber = 2 else theNumber = 1 end if
session("responseSortMethodology" & count) = theNumber
if((count MOD (2*3)) = 0) then
if(theNumber=1) then theNumber = 2 else theNumber = 1 end if
end if
next

session("IntervieweeCode") = (1000000000*rnd)\1 & "-" & now()
listOfNums = ""

'session("initializedValues") = ""

for count = 1 to session("setsToDisplayPerSortType")*3*2

```

```

newNum = ((500*Rnd) \ 1)
if(InStr(listOfNums,"a" & newNum & "z") = 0) then
  listOfNums = listOfNums & "a" & newNum & "z"
  session("theID" & count) = newNum
  if((count mod 3) = 0) then
    session("responseType" & count) = "Name"
  elseif((count mod 3) = 1) then
    session("responseType" & count) = "DateTime"
  else
    session("responseType" & count) = "Subject"
  end if
  SQL = "Select * from ArrayItems WHERE ID = " & session("theID" &
count)
  RS.Open SQL,Conn
  Do While Not RS.EOF
    session("theName" & count) = RS("Name")
    session("theSubject" & count) =
getPartialSubject(RS("Subject"))
    session("theDateTime" & count) = mid(RS("DateTime"),1,8)
    RS.MoveNext
  Loop
  RS.Close

  SQL = "Select ID from ArrayItems WHERE "

  if(session("responseType" & count) <> "Name") then
    SQL = SQL & " Name = '" & session("theName" & count) & "' "
  end if

  if(session("responseType" & count) <> "Subject") then
    If(InStr(SQL, "Name = ") <> 0)then SQL = SQL & " AND " end if
    SQL = SQL & " Subject like '" & session("theSubject" & count) &
"%'" "
  end if

  if(session("responseType" & count) <> "DateTime") then
    SQL = SQL & " AND DateTime like '" & session("theDateTime" &
count) & "%'" "
  end if

  SQL = SQL & " ORDER BY ID "

  RS.Open SQL,Conn
  session("correctAnswer" & count) = "."
  newCount = 0
  Do While Not RS.EOF
    theID = RS("ID")
    newCount = newCount + 1
    session("correctAnswer" & count) = session("correctAnswer" &
count) & theID & "."
    RS.MoveNext
  Loop
  session("CorrectAnswerNumberOf" & count) = newCount
'session("initializedValues") = session("initializedValues") & "<br>" &
newCount
'possible Correct answers

```

```

        RS.Close
    else
        count = count - 1
    end if
next

'Response.Write("<br>" & session("initializedValues"))

Conn.Close

for count = 1 to session("setsToDisplayPerSortType")*3*2
    session("exerciseDisplayOrder" & count) = 0
next

theNumberString = "."
for count = 1 to session("setsToDisplayPerSortType")*3*2
    theNumber = ((session("setsToDisplayPerSortType")*3*2*Rnd) \ 1)
    if(theNumber+0 = 0) then theNumber = 1 end if
    while(InStr(theNumberString, "." & theNumber & ".") <> 0)
        theNumber = theNumber + 1
        if(session("setsToDisplayPerSortType")*3*2 + 0 < theNumber+0 )
then
            theNumber = 1
        end if
    wend
    theNumberString = theNumberString & theNumber & "."
    session("exerciseDisplayOrder" & count) = theNumber
'Response.Write("<br>session(exerciseDisplayOrder" & count & ")" &
session("exerciseDisplayOrder" & count))
'Response.Write(" : session(correctAnswer" &
session("exerciseDisplayOrder" & count) & ")" & session("correctAnswer"
& session("exerciseDisplayOrder" & count)))
next
    session("ExerciseDisplayOrder") = theNumberString
end sub

sub initializeQuestionnaire()
' Dim Conn 'Our connection object
' Dim RS
' Dim SQL

%><!--      #include file = "thesisDatabaseConnection.asp" --><%

Randomize()

SQL = "Select * from QuestionnaireQuestion WHERE Active = Yes ORDER
BY QuestionPurpose, QuestionType DESC, TimeOfAppearance "
RS.Open SQL,Conn
count = 0
earlyTimeOfAppearance = 0
Do While Not RS.EOF
    count = count + 1
    session("QuestionTransferableID" & count) =
RS("QuestionTransferableID")
    session("QuestionPurpose" & count) = RS("QuestionPurpose")
    session("QuestionType" & count) = RS("QuestionType")
    session("Question" & count) = RS("Question")

```

```

session("TimeOfAppearance" & count) = RS("TimeOfAppearance")
session("PossibleResponse1" & count) = RS("PossibleResponse1")
session("PossibleResponse2" & count) = RS("PossibleResponse2")
session("PossibleResponse3" & count) = RS("PossibleResponse3")
session("PossibleResponse4" & count) = RS("PossibleResponse4")
session("PossibleResponse5" & count) = RS("PossibleResponse5")
if(session("TimeOfAppearance" & count) = "early") then
    earlyTimeOfAppearance = earlyTimeOfAppearance + 1
end if
RS.MoveNext
Loop
session("NumberOfQuestionnaireQuestions") = count
RS.Close
Conn.Close

theNumberString = "."
for count = 1 to earlyTimeOfAppearance
    theNumber = (earlyTimeOfAppearance*Rnd \ 1)
    if(theNumber+0 = 0) then theNumber = 1 end if
    while(InStr(theNumberString, "." & theNumber & ".") <> 0)
        theNumber = theNumber + 1
        if(earlyTimeOfAppearance + 0 < theNumber+0 ) then
            theNumber = 1
        end if
    wend
    theNumberString = theNumberString & theNumber & "."
    session("questionnaireDisplayOrder" & count) = theNumber
next
for count = (earlyTimeOfAppearance+1) to
(session("NumberOfQuestionnaireQuestions")+0)
    theNumberString = theNumberString & count & "."
    session("questionnaireDisplayOrder" & count) = count
next
session("QuestionnaireDisplayOrder") = theNumberString
'make sure to add the strings with the order number into the database
for the employee record
end sub

function numberOfCompletedQuestionnaires()
    Dim Conn 'Our connection object
    Dim RS
    Dim SQL

%><!--      #include file = "thesisDatabaseConnection.asp" --><%

    SQL = "Select * from Interviewee "
    RS.Open SQL,Conn
    count = 0
    Do While Not RS.EOF
        count = count + 1
        RS.MoveNext
    Loop
    RS.Close

    numberOfCompletedQuestionnaires = count

end function

```

```

sub saveIntervieweeResponsesCoupon()
    Dim Conn 'Our connection object
    Dim RSAuthors
    Dim SQL

%><!--      #include file = "thesisDatabaseConnection.asp" --><%

    s =      " Name          = '" &
convertToDatabaseSingleQuotes(Request.form("intervieweeName"))
& "' , "
    s = s & " EmailAddress = '" &
convertToDatabaseSingleQuotes(Request.form("intervieweeEmailAddress"))
& "' "

    SQL = "UPDATE Interviewee SET " & s & " WHERE ID = " &
session("RecordID")
    Conn.Execute(SQL)

    if((Request.form("couponMailingAddress")<>"") OR
(Request.form("couponEmailAddress")<>"") OR
(Request.form("couponCouponID")<>"")) then
        sField = " MailingAddress "
        sField = sField & " , EmailAddress "
        sField = sField & " , CouponID "
        sField = sField & " , EntryDateTime "

        sValue =      " '" &
convertToDatabaseSingleQuotes(Request.form("couponMailingAddress")) &
"' "
        sValue = sValue & " , '" &
convertToDatabaseSingleQuotes(Request.form("couponEmailAddress")) &
"' "
        sValue = sValue & " , " & Request.form("couponCouponID")
& " "
        sValue = sValue & " , '" & Now()
& "' "

        SQL = "INSERT INTO CouponIntervieweeInfo (" & sField & ") VALUES ("
& sValue & " ) "
        Conn.Execute(SQL)

        SQL = "SELECT NumberRemaining FROM Coupon WHERE ID = " &
Request.form("couponCouponID") & " "
        RS.Open SQL,Conn
        If NOT RS.EOF then
            numberRemaining = RS("NumberRemaining")
            RS.Close
            s =      " NumberRemaining = " & numberRemaining - 1 & " "
            SQL = "UPDATE Coupon SET " & s & " WHERE ID = " &
Request.form("couponCouponID") & " "
            Conn.Execute(SQL)
        Else
            RS.Close
        End if
    End If
end sub

```

```

end sub

sub saveIntervieweeResponsesExerciseAndQuestionnaire()
    'load the person
    'verify the person .. get the person's ID
    'load users responses

    Dim Conn 'Our connection object
    Dim RSAuthors
    Dim SQL

%><!--      #include file = "thesisDatabaseConnection.asp" --><%

    Randomize()

    RecordID = 0
    count = 0
    while (RecordID = 0) AND (count < 40)
        count = count + 1 '(prevent runaways)
        SQL = "INSERT INTO Interviewee (UniqueWebCode) VALUES ('" &
session("IntervieweeCode") & "') "
        Conn.Execute(SQL)
        SQL = "SELECT ID FROM Interviewee WHERE UniqueWebCode = '" &
session("IntervieweeCode") & "'"
        RS.Open SQL,Conn
        If NOT RS.EOF then
            RecordID = RS("ID")
            RS.Close
        Else
            RS.Close
        End if
    wend

    session("RecordID") = RecordID

    session("TotalPossibleStandardWayFindName")      = 0
    session("TotalPossibleStandardWayFindSubject")    = 0
    session("TotalPossibleStandardWayFindDate")      = 0

    session("TotalCorrectStandardWayFindName")      = 0
    session("TotalCorrectStandardWayFindSubject")    = 0
    session("TotalCorrectStandardWayFindDate")      = 0

    session("TotalIncorrectStandardWayFindName")     = 0
    session("TotalIncorrectStandardWayFindSubject")  = 0
    session("TotalIncorrectStandardWayFindDate")     = 0

    session("TotalPossibleStandardWay")              = 0
    session("TotalCorrectStandardWay")                = 0
    session("TotalIncorrectStandardWay")              = 0

    session("PercentCorrectStandardWayFindName")    = 0
    session("PercentCorrectStandardWayFindSubject") = 0
    session("PercentCorrectStandardWayFindDate")    = 0

    session("PercentIncorrectStandardWayFindName")  = 0
    session("PercentIncorrectStandardWayFindSubject") = 0

```

```

session("PercentIncorrectStandardWayFindDate") = 0

session("PercentCorrectStandardWay") = 0
session("PercentIncorrectStandardWay") = 0

session("TotalPossibleMultipleWayFindName") = 0
session("TotalPossibleMultipleWayFindSubject") = 0
session("TotalPossibleMultipleWayFindDate") = 0

session("TotalCorrectMultipleWayFindName") = 0
session("TotalCorrectMultipleWayFindSubject") = 0
session("TotalCorrectMultipleWayFindDate") = 0

session("TotalIncorrectMultipleWayFindName") = 0
session("TotalIncorrectMultipleWayFindSubject") = 0
session("TotalIncorrectMultipleWayFindDate") = 0

session("TotalPossibleMultipleWay") = 0
session("TotalCorrectMultipleWay") = 0
session("TotalIncorrectMultipleWay") = 0

session("PercentCorrectMultipleWayFindName") = 0
session("PercentCorrectMultipleWayFindSubject") = 0
session("PercentCorrectMultipleWayFindDate") = 0

session("PercentIncorrectMultipleWayFindName") = 0
session("PercentIncorrectMultipleWayFindSubject") = 0
session("PercentIncorrectMultipleWayFindDate") = 0

session("PercentCorrectMultipleWay") = 0
session("PercentIncorrectMultipleWay") = 0

session("TotalPossibleBothWays") = 0
session("TotalCorrectBothWays") = 0
session("TotalIncorrectBothWays") = 0

session("PercentCorrectBothWays") = 0
session("PercentIncorrectBothWays") = 0

session("TotalTimeStandardWayFindName") = 0
session("TotalTimeStandardWayFindSubject") = 0
session("TotalTimeStandardWayFindDate") = 0

session("TotalTimeMultipleWayFindName") = 0
session("TotalTimeMultipleWayFindSubject") = 0
session("TotalTimeMultipleWayFindDate") = 0

session("TotalTimeStandardWay") = 0
session("TotalTimeMultipleWay") = 0
session("TotalTimeBothWays") = 0

session("PercentTimeStandardOverBoth") = 0
session("PercentTimeMultipleOverBoth") = 0
session("RatioTimeMultipleOverStandard") = 0

for count = 1 to session("setsToDisplayPerSortType")*3*2
  sField = " IntervieweeID "

```

```

sField = sField & " , IntervieweeCode "
sField = sField & " , ResponseType "
sField = sField & " , CorrectAnswer "
sField = sField & " , IntervieweeResponse "
sField = sField & " , TimeStart "
sField = sField & " , TimeEnd "
sField = sField & " , SecondsElapsed "
sField = sField & " , CorrectAnswerNumberOf "
sField = sField & " , AnswerIntervieweeVsCorrectSame "
sField = sField & " , AnswerIntervieweeVsCorrectDifferent "
sField = sField & " , AnswerCorrectVsIntervieweeSame "
sField = sField & " , AnswerCorrectVsIntervieweeDifferent "

sValue = RecordID
sValue = sValue & " , '" & session("IntervieweeCode")
& "' "
sValue = sValue & " , '" & session("responseType" & count)
& "' "
sValue = sValue & " , '" & session("correctAnswer" & count)
& "' "
sValue = sValue & " , '" & session("intervieweeResponse" & count)
& "' "
sValue = sValue & " , '" & session("timeStart" & count)
& "' "
sValue = sValue & " , '" & session("timeEnd" & count)
& "' "
sValue = sValue & " , '" & session("secondsElapsed" & count)
& "' "
sValue = sValue & " , '" & session("CorrectAnswerNumberOf" &
count) & "' "
sValue = sValue & " , '" &
session("AnswerIntervieweeVsCorrectSame" & count) & "' "
sValue = sValue & " , '" &
session("AnswerIntervieweeVsCorrectDifferent" & count) & "' "
sValue = sValue & " , '" &
session("AnswerCorrectVsIntervieweeSame" & count) & "' "
sValue = sValue & " , '" &
session("AnswerCorrectVsIntervieweeDifferent" & count) & "' "
SQL = "INSERT INTO ExerciseResponse (" & sField & ") VALUES (" &
sValue & ") "
Conn.Execute(SQL)
'Response.Write("<br>" & SQL)

If(count MOD 6 = 1) Then
    session("TotalPossibleStandardWayFindName") =
session("TotalPossibleStandardWayFindName") +
session("CorrectAnswerNumberOf" & count)
    session("TotalCorrectStandardWayFindName") =
session("TotalCorrectStandardWayFindName") +
session("AnswerIntervieweeVsCorrectSame" & count)
    If(session("AnswerIntervieweeVsCorrectDifferent" & count)+0 <
session("AnswerCorrectVsIntervieweeDifferent" & count)+0) Then
        temp = session("AnswerCorrectVsIntervieweeDifferent" & count)
    Else
        temp = session("AnswerIntervieweeVsCorrectDifferent" & count)
    End If

```

```

        session("TotalIncorrectStandardWayFindName") =
session("TotalIncorrectStandardWayFindName") + temp
        session("TotalTimeStandardWayFindName") =
session("TotalTimeStandardWayFindName") + session("secondsElapsed"
& count)
    ElseIf(count MOD 6 = 2) Then
        session("TotalPossibleStandardWayFindSubject") =
session("TotalPossibleStandardWayFindSubject") +
session("CorrectAnswerNumberOf" & count)
        session("TotalCorrectStandardWayFindSubject") =
session("TotalCorrectStandardWayFindSubject") +
session("AnswerIntervieweeVsCorrectSame" & count)
        If(session("AnswerIntervieweeVsCorrectDifferent" & count)+0 <
session("AnswerCorrectVsIntervieweeDifferent" & count)+0) Then
            temp = session("AnswerCorrectVsIntervieweeDifferent" & count)
        Else
            temp = session("AnswerIntervieweeVsCorrectDifferent" & count)
        End If
        session("TotalIncorrectStandardWayFindSubject") =
session("TotalIncorrectStandardWayFindSubject") + temp
        session("TotalTimeStandardWayFindSubject") =
session("TotalTimeStandardWayFindSubject") +
session("secondsElapsed" & count)
    ElseIf(count MOD 6 = 3) Then
        session("TotalPossibleStandardWayFindDate") =
session("TotalPossibleStandardWayFindDate") +
session("CorrectAnswerNumberOf" & count)
        session("TotalCorrectStandardWayFindDate") =
session("TotalCorrectStandardWayFindDate") +
session("AnswerIntervieweeVsCorrectSame" & count)
        If(session("AnswerIntervieweeVsCorrectDifferent" & count)+0 <
session("AnswerCorrectVsIntervieweeDifferent" & count)+0) Then
            temp = session("AnswerCorrectVsIntervieweeDifferent" & count)
        Else
            temp = session("AnswerIntervieweeVsCorrectDifferent" & count)
        End If
        session("TotalIncorrectStandardWayFindDate") =
session("TotalIncorrectStandardWayFindDate") + temp
        session("TotalTimeStandardWayFindDate") =
session("TotalTimeStandardWayFindDate") + session("secondsElapsed"
& count)
    ElseIf(count MOD 6 = 4) Then
        session("TotalPossibleMultipleWayFindName") =
session("TotalPossibleMultipleWayFindName") +
session("CorrectAnswerNumberOf" & count)
        session("TotalCorrectMultipleWayFindName") =
session("TotalCorrectMultipleWayFindName") +
session("AnswerIntervieweeVsCorrectSame" & count)
        If(session("AnswerIntervieweeVsCorrectDifferent" & count)+0 <
session("AnswerCorrectVsIntervieweeDifferent" & count)+0) Then
            temp = session("AnswerCorrectVsIntervieweeDifferent" & count)
        Else
            temp = session("AnswerIntervieweeVsCorrectDifferent" & count)
        End If
        session("TotalIncorrectMultipleWayFindName") =
session("TotalIncorrectMultipleWayFindName") + temp

```

```

        session("TotalTimeMultipleWayFindName") =
session("TotalTimeMultipleWayFindName") + session("secondsElapsed"
& count)
    ElseIf(count MOD 6 = 5) Then
        session("TotalPossibleMultipleWayFindSubject") =
session("TotalPossibleMultipleWayFindSubject") +
session("CorrectAnswerNumberOf" & count)
        session("TotalCorrectMultipleWayFindSubject") =
session("TotalCorrectMultipleWayFindSubject") +
session("AnswerIntervieweeVsCorrectSame" & count)
        If(session("AnswerIntervieweeVsCorrectDifferent" & count)+0 <
session("AnswerCorrectVsIntervieweeDifferent" & count)+0) Then
            temp = session("AnswerCorrectVsIntervieweeDifferent" & count)
        Else
            temp = session("AnswerIntervieweeVsCorrectDifferent" & count)
        End If
        session("TotalIncorrectMultipleWayFindSubject") =
session("TotalIncorrectMultipleWayFindSubject") + temp
        session("TotalTimeMultipleWayFindSubject") =
session("TotalTimeMultipleWayFindSubject") +
session("secondsElapsed" & count)
        ElseIf(count MOD 6 = 0) Then
            session("TotalPossibleMultipleWayFindDate") =
session("TotalPossibleMultipleWayFindDate") +
session("CorrectAnswerNumberOf" & count)
            session("TotalCorrectMultipleWayFindDate") =
session("TotalCorrectMultipleWayFindDate") +
session("AnswerIntervieweeVsCorrectSame" & count)
            If(session("AnswerIntervieweeVsCorrectDifferent" & count)+0 <
session("AnswerCorrectVsIntervieweeDifferent" & count)+0) Then
                temp = session("AnswerCorrectVsIntervieweeDifferent" & count)
            Else
                temp = session("AnswerIntervieweeVsCorrectDifferent" & count)
            End If
            session("TotalIncorrectMultipleWayFindDate") =
session("TotalIncorrectMultipleWayFindDate") + temp
            session("TotalTimeMultipleWayFindDate") =
session("TotalTimeMultipleWayFindDate") + session("secondsElapsed"
& count)
'Response.Write("<br>" & session("TotalTimeMultipleWayFindDate"))
        End If

'Response.Write("<br>" & session("secondsElapsed" & count))

    If(((count MOD 6) <= 3) and (0 < (count MOD 6))) Then
        session("TotalPossibleStandardWay") =
session("TotalPossibleStandardWay") + session("CorrectAnswerNumberOf"
& count)
        session("TotalCorrectStandardWay") =
session("TotalCorrectStandardWay") +
session("AnswerIntervieweeVsCorrectSame" & count)
        If(session("AnswerIntervieweeVsCorrectDifferent" & count)+0 <
session("AnswerCorrectVsIntervieweeDifferent" & count)+0) Then
            temp = session("AnswerCorrectVsIntervieweeDifferent" & count)
        Else
            temp = session("AnswerIntervieweeVsCorrectDifferent" & count)
        End If

```

```

        session("TotalIncorrectStandardWay") =
session("TotalIncorrectStandardWay") + temp
        session("TotalTimeStandardWay")
=
session("TotalTimeStandardWay")      + session("secondsElapsed" &
count)
    Else
        session("TotalPossibleMultipleWay") =
session("TotalPossibleMultipleWay") + session("CorrectAnswerNumberOf"
& count)
        session("TotalCorrectMultipleWay") =
session("TotalCorrectMultipleWay") +
session("AnswerIntervieweeVsCorrectSame" & count)
        If(session("AnswerIntervieweeVsCorrectDifferent" & count)+0 <
session("AnswerCorrectVsIntervieweeDifferent" & count)+0) Then
            temp = session("AnswerCorrectVsIntervieweeDifferent" & count)
        Else
            temp = session("AnswerIntervieweeVsCorrectDifferent" & count)
        End If
        session("TotalIncorrectMultipleWay") =
session("TotalIncorrectMultipleWay") + temp
        session("TotalTimeMultipleWay")
=
session("TotalTimeMultipleWay")      + session("secondsElapsed" &
count)
    End If

        session("TotalPossibleBothWays") =
session("TotalPossibleBothWays") + session("CorrectAnswerNumberOf" &
count)
        session("TotalCorrectBothWays") = session("TotalCorrectBothWays")
+ session("AnswerIntervieweeVsCorrectSame" & count)
        If(session("AnswerIntervieweeVsCorrectDifferent" & count)+0 <
session("AnswerCorrectVsIntervieweeDifferent" & count)+0) Then
            temp = session("AnswerCorrectVsIntervieweeDifferent" & count)
        Else
            temp = session("AnswerIntervieweeVsCorrectDifferent" & count)
        End If
        session("TotalIncorrectBothWays") =
session("TotalIncorrectBothWays") + temp
        session("TotalTimeBothWays")      = session("TotalTimeBothWays")
+ session("secondsElapsed" & count)
    next

        session("PercentCorrectStandardWayFindName")      =
(session("TotalCorrectStandardWayFindName")*100) \
session("TotalPossibleStandardWayFindName")
        session("PercentCorrectStandardWayFindSubject") =
(session("TotalCorrectStandardWayFindSubject")*100) \
session("TotalPossibleStandardWayFindSubject")
        session("PercentCorrectStandardWayFindDate")      =
(session("TotalCorrectStandardWayFindDate")*100) \
session("TotalPossibleStandardWayFindDate")

        session("PercentIncorrectStandardWayFindName")    =
(session("TotalIncorrectStandardWayFindName")*100) \
session("TotalPossibleStandardWayFindName")

```

```

    session("PercentIncorrectStandardWayFindSubject") =
(session("TotalIncorrectStandardWayFindSubject")*100) \
session("TotalPossibleStandardWayFindSubject")
    session("PercentIncorrectStandardWayFindDate") =
(session("TotalIncorrectStandardWayFindDate")*100) \
session("TotalPossibleStandardWayFindDate")

    session("PercentCorrectStandardWay") =
(session("TotalCorrectStandardWay")*100) \
session("TotalPossibleStandardWay")
    session("PercentIncorrectStandardWay") =
(session("TotalIncorrectStandardWay")*100) \
session("TotalPossibleStandardWay")

    session("PercentCorrectMultipleWayFindName") =
(session("TotalCorrectMultipleWayFindName")*100) \
session("TotalPossibleMultipleWayFindName")
    session("PercentCorrectMultipleWayFindSubject") =
(session("TotalCorrectMultipleWayFindSubject")*100) \
session("TotalPossibleMultipleWayFindSubject")
    session("PercentCorrectMultipleWayFindDate") =
(session("TotalCorrectMultipleWayFindDate")*100) \
session("TotalPossibleMultipleWayFindDate")

    session("PercentIncorrectMultipleWayFindName") =
(session("TotalIncorrectMultipleWayFindName")*100) \
session("TotalPossibleMultipleWayFindName")
    session("PercentIncorrectMultipleWayFindSubject") =
(session("TotalIncorrectMultipleWayFindSubject")*100) \
session("TotalPossibleMultipleWayFindSubject")
    session("PercentIncorrectMultipleWayFindDate") =
(session("TotalIncorrectMultipleWayFindDate")*100) \
session("TotalPossibleMultipleWayFindDate")

    session("PercentCorrectMultipleWay") =
(session("TotalCorrectMultipleWay")*100) \
session("TotalPossibleMultipleWay")
    session("PercentIncorrectMultipleWay") =
(session("TotalIncorrectMultipleWay")*100) \
session("TotalPossibleMultipleWay")

    session("PercentCorrectBothWays") =
(session("TotalCorrectBothWays")*100) \
session("TotalPossibleBothWays")
    session("PercentIncorrectBothWays") =
(session("TotalIncorrectBothWays")*100) \
session("TotalPossibleBothWays")

    session("PercentTimeStandardOverBoth") =
(session("TotalTimeStandardWay")*100) \ session("TotalTimeBothWays")
    session("PercentTimeMultipleOverBoth") =
(session("TotalTimeMultipleWay")*100) \ session("TotalTimeBothWays")
    session("RatioTimeMultipleOverStandard") =
(session("TotalTimeMultipleWay")*100) \ session("TotalTimeStandardWay")

s = " InterviewDate = '" & now() & "', "

```

```

s = s & " TotalPossibleStandardWayFindName      = " &
session("TotalPossibleStandardWayFindName")    & ", "
s = s & " TotalPossibleStandardWayFindSubject = " &
session("TotalPossibleStandardWayFindSubject") & ", "
s = s & " TotalPossibleStandardWayFindDate     = " &
session("TotalPossibleStandardWayFindDate")    & ", "

s = s & " TotalCorrectStandardWayFindName      = " &
session("TotalCorrectStandardWayFindName")    & ", "
s = s & " TotalCorrectStandardWayFindSubject = " &
session("TotalCorrectStandardWayFindSubject") & ", "
s = s & " TotalCorrectStandardWayFindDate     = " &
session("TotalCorrectStandardWayFindDate")    & ", "

s = s & " TotalIncorrectStandardWayFindName     = " &
session("TotalIncorrectStandardWayFindName")  & ", "
s = s & " TotalIncorrectStandardWayFindSubject = " &
session("TotalIncorrectStandardWayFindSubject") & ", "
s = s & " TotalIncorrectStandardWayFindDate    = " &
session("TotalIncorrectStandardWayFindDate")  & ", "

s = s & " TotalPossibleStandardWay             = " &
session("TotalPossibleStandardWay")           & ", "
s = s & " TotalCorrectStandardWay             = " &
session("TotalCorrectStandardWay")           & ", "
s = s & " TotalIncorrectStandardWay           = " &
session("TotalIncorrectStandardWay")         & ", "

s = s & " PercentCorrectStandardWayFindName    = " &
session("PercentCorrectStandardWayFindName")  & ", " 'correct /
possible
s = s & " PercentCorrectStandardWayFindSubject = " &
session("PercentCorrectStandardWayFindSubject") & ", "
s = s & " PercentCorrectStandardWayFindDate    = " &
session("PercentCorrectStandardWayFindDate")  & ", "

s = s & " PercentIncorrectStandardWayFindName   = " &
session("PercentIncorrectStandardWayFindName") & ", " 'incorrect /
possible
s = s & " PercentIncorrectStandardWayFindSubject = " &
session("PercentIncorrectStandardWayFindSubject") & ", "
s = s & " PercentIncorrectStandardWayFindDate   = " &
session("PercentIncorrectStandardWayFindDate") & ", "

s = s & " PercentCorrectStandardWay            = " &
session("PercentCorrectStandardWay")          & ", "
s = s & " PercentIncorrectStandardWay          = " &
session("PercentIncorrectStandardWay")        & ", "

s = s & " TotalPossibleMultipleWayFindName     = " &
session("TotalPossibleMultipleWayFindName")  & ", "
s = s & " TotalPossibleMultipleWayFindSubject = " &
session("TotalPossibleMultipleWayFindSubject") & ", "
s = s & " TotalPossibleMultipleWayFindDate    = " &
session("TotalPossibleMultipleWayFindDate")  & ", "

```

```

s = s & " TotalCorrectMultipleWayFindName      = " &
session("TotalCorrectMultipleWayFindName")    & ", "
s = s & " TotalCorrectMultipleWayFindSubject = " &
session("TotalCorrectMultipleWayFindSubject") & ", "
s = s & " TotalCorrectMultipleWayFindDate     = " &
session("TotalCorrectMultipleWayFindDate")    & ", "

s = s & " TotalIncorrectMultipleWayFindName    = " &
session("TotalIncorrectMultipleWayFindName")  & ", "
s = s & " TotalIncorrectMultipleWayFindSubject = " &
session("TotalIncorrectMultipleWayFindSubject") & ", "
s = s & " TotalIncorrectMultipleWayFindDate    = " &
session("TotalIncorrectMultipleWayFindDate")  & ", "

s = s & " TotalPossibleMultipleWay            = " &
session("TotalPossibleMultipleWay")           & ", "
s = s & " TotalCorrectMultipleWay            = " &
session("TotalCorrectMultipleWay")            & ", "
s = s & " TotalIncorrectMultipleWay          = " &
session("TotalIncorrectMultipleWay")          & ", "

s = s & " PercentCorrectMultipleWayFindName    = " &
session("PercentCorrectMultipleWayFindName")  & ", "
'correct/possible
s = s & " PercentCorrectMultipleWayFindSubject = " &
session("PercentCorrectMultipleWayFindSubject") & ", "
s = s & " PercentCorrectMultipleWayFindDate    = " &
session("PercentCorrectMultipleWayFindDate")  & ", "

s = s & " PercentIncorrectMultipleWayFindName   = " &
session("PercentIncorrectMultipleWayFindName") & ", "
'incorrect/possible
s = s & " PercentIncorrectMultipleWayFindSubject = " &
session("PercentIncorrectMultipleWayFindSubject") & ", "
s = s & " PercentIncorrectMultipleWayFindDate    = " &
session("PercentIncorrectMultipleWayFindDate") & ", "

s = s & " PercentCorrectMultipleWay            = " &
session("PercentCorrectMultipleWay")           & ", "
s = s & " PercentIncorrectMultipleWay          = " &
session("PercentIncorrectMultipleWay")          & ", "

s = s & " TotalPossibleBothWays                = " &
session("TotalPossibleBothWays")               & ", "
s = s & " TotalCorrectBothWays                = " &
session("TotalCorrectBothWays")               & ", "
s = s & " TotalIncorrectBothWays              = " &
session("TotalIncorrectBothWays")             & ", "

s = s & " PercentCorrectBothWays              = " &
session("PercentCorrectBothWays")             & ", "
s = s & " PercentIncorrectBothWays            = " &
session("PercentIncorrectBothWays")           & ", "

s = s & " TotalTimeStandardWayFindName        = " &
session("TotalTimeStandardWayFindName")       & ", "

```

```

s = s & " TotalTimeStandardWayFindSubject = " &
session("TotalTimeStandardWayFindSubject") & ", "
s = s & " TotalTimeStandardWayFindDate = " &
session("TotalTimeStandardWayFindDate") & ", "

s = s & " TotalTimeMultipleWayFindName = " &
session("TotalTimeMultipleWayFindName") & ", "
s = s & " TotalTimeMultipleWayFindSubject = " &
session("TotalTimeMultipleWayFindSubject") & ", "
s = s & " TotalTimeMultipleWayFindDate = " &
session("TotalTimeMultipleWayFindDate") & ", "

s = s & " TotalTimeStandardWay = " & session("TotalTimeStandardWay")
& ", "
s = s & " TotalTimeMultipleWay = " & session("TotalTimeMultipleWay")
& ", "
s = s & " TotalTimeBothWays = " & session("TotalTimeBothWays")
& ", "

s = s & " PercentTimeStandardOverBoth = " &
session("PercentTimeStandardOverBoth") & ", "
s = s & " PercentTimeMultipleOverBoth = " &
session("PercentTimeMultipleOverBoth") & ", "
s = s & " RatioTimeMultipleOverStandard = " &
session("RatioTimeMultipleOverStandard") & ", "

s = s & " ExerciseDisplayOrder = '" &
session("ExerciseDisplayOrder") & "', "
s = s & " QuestionnaireDisplayOrder = '" &
session("QuestionnaireDisplayOrder") & "', "

s = s & " QuestionnaireTimeStart = '" &
session("QuestionnaireTimeStart") & "', "
s = s & " QuestionnaireTimeEnd = '" &
session("QuestionnaireTimeEnd") & "', "
s = s & " QuestionnaireTotalTime = '" &
session("QuestionnaireTotalTime") & "' "

SQL = "UPDATE Interviewee SET " & s & " WHERE ID = " & RecordID
Conn.Execute(SQL)

session("questionTransferableIDrelationship" & count) =
session("QuestionTransferableID" & count)

for count = 1 to session("NumberOfQuestionnaireQuestions")
sField = " IntervieweeID "
sField = sField & " , IntervieweeCode "
sField = sField & " , QuestionTransferableID "
sField = sField & " , Response "

sValue = RecordID
sValue = sValue & " , '" & session("IntervieweeCode")
& "' "
sValue = sValue & " , '" &
session("questionTransferableIDrelationship" & count) & "' "
sValue = sValue & " , '" &
convertToDatabaseSingleQuotes(Request.form("response" & count)) & "' "

```

```

        if(session("QuestionType" & count) = "comment") then
            SQL = "INSERT INTO QuestionnaireResponseMemo (" & sField & ")
VALUES (" & sValue & ") "
        else
            SQL = "INSERT INTO QuestionnaireResponseMultipleChoice (" &
sField & ") VALUES (" & sValue & ") "
        end if

        Conn.Execute(SQL)
    next
end sub

sub setTimeStartAndEnd(theExerciseNumber)
    ts = session("timeStart" & (session("exerciseDisplayOrder" &
theExerciseNumber)))
    te = session("timeEnd" & (session("exerciseDisplayOrder" &
theExerciseNumber)))
    ds = Day(ts) 'if they are doing this at midnight on the last day of
the month
    de = Day(te) ' we need to make sure that it doesn't calculate wrong
    if(de<ds)then de = ds+1 end if
    tst = ds*24*3600 + Hour(ts)*3600 + Minute(ts)*60 + Second(ts)
    tet = de*24*3600 + Hour(te)*3600 + Minute(te)*60 + Second(te)
    session("SecondsElapsed" & session("exerciseDisplayOrder" &
theExerciseNumber)) = tet - tst
end sub

sub setTimeStartAndEndQuestionnaire()
    ts = session("QuestionnaireTimeStart")
    te = session("QuestionnaireTimeEnd")
    ds = Day(ts) 'if they are doing this at midnight on the last day of
the month
    de = Day(te) ' we need to make sure that it doesn't calculate wrong
    if(de<ds)then de = ds+1 end if
    tst = ds*24*3600 + Hour(ts)*3600 + Minute(ts)*60 + Second(ts)
    tet = de*24*3600 + Hour(te)*3600 + Minute(te)*60 + Second(te)
    session("QuestionnaireTotalTime") = tet - tst
end sub

function timeProperDisplay(dateTimeInput)
    theHour = mid(dateTimeInput,9,2)
    theMinute = mid(dateTimeInput,11,2)
    theTime = getAmPm(theHour, theMinute)

    timeOutput = theTime
    timeProperDisplay = timeOutput
end function

function updateReplyForExerciseNumber(currentExerciseNumber)
    if(Request.Form("theReply") <> "") then
'Response.Write("<br>")
        session("AnswerIntervieweeVsCorrectSame" &
session("exerciseDisplayOrder" & currentExerciseNumber)) = 0
        session("AnswerIntervieweeVsCorrectDifferent" &
session("exerciseDisplayOrder" & currentExerciseNumber)) = 0
    end if
end function

```

```

        session("AnswerCorrectVsIntervieweeSame" &
session("exerciseDisplayOrder" & currentExerciseNumber)) = 0
        session("AnswerCorrectVsIntervieweeDifferent" &
session("exerciseDisplayOrder" & currentExerciseNumber)) = 0
        theReply = Request.Form("theReply")
        lenTheReply = len(theReply)
        thePhrase = "."
        theWord = "."
        for count = 1 to lenTheReply
            theChar = mid(theReply,count,1)
            if(theChar="a") then
                theWord = theWord & "." 'compare each entry to see if it was
supposed to be there
                thePhrase = thePhrase & "."
                if(InStr(session("correctAnswer" &
session("exerciseDisplayOrder" & currentExerciseNumber)),theWord) <> 0)
then
                    session("AnswerIntervieweeVsCorrectSame" &
session("exerciseDisplayOrder" & currentExerciseNumber)) =
session("AnswerIntervieweeVsCorrectSame" &
session("exerciseDisplayOrder" & currentExerciseNumber)) + 1
                else
                    session("AnswerIntervieweeVsCorrectDifferent" &
session("exerciseDisplayOrder" & currentExerciseNumber)) =
session("AnswerIntervieweeVsCorrectDifferent" &
session("exerciseDisplayOrder" & currentExerciseNumber)) + 1
                end if
            elseif((theChar=",") OR (theChar=" ")) then
                theWord = "."
            else
                theWord = theWord & theChar
                thePhrase = thePhrase & theChar
            end if
            session("intervieweeResponse" & session("exerciseDisplayOrder" &
currentExerciseNumber)) = thePhrase
        next
'Response.Write("<br>intervieweeResponse" &
session("exerciseDisplayOrder" & currentExerciseNumber) & " : " &
theWord)
'Response.Write("<br>AnswerIntervieweeVsCorrectSame" &
session("exerciseDisplayOrder" & currentExerciseNumber) & " : " &
session("AnswerIntervieweeVsCorrectSame" &
session("exerciseDisplayOrder" & currentExerciseNumber)))
'Response.Write("<br>AnswerIntervieweeVsCorrectDifferent" &
session("exerciseDisplayOrder" & currentExerciseNumber) & " : " &
session("AnswerIntervieweeVsCorrectDifferent" &
session("exerciseDisplayOrder" & currentExerciseNumber)))
'Response.Write("<br>" & session("intervieweeResponse" &
session("exerciseDisplayOrder" & currentExerciseNumber)))
        lenTheCorrect = len(session("correctAnswer" &
session("exerciseDisplayOrder" & currentExerciseNumber)))
        theWord = ""
        for count = 1 to lenTheCorrect
            theChar = mid(session("correctAnswer" &
session("exerciseDisplayOrder" & currentExerciseNumber)),count,1)
            if(theChar=".") then
                if(2 <= count) then

```

```

        theWord = "." & theWord & "."
'Response.Write("<br>InStr(session('intervieweeResponse' & " &
session("exerciseDisplayOrder" & currentExerciseNumber) & "), theWord)
<> 0) ")
'Response.Write("<br>InStr(" & session("intervieweeResponse" &
session("exerciseDisplayOrder" & currentExerciseNumber)) & ", " &
theWord & ") <> 0) ")
        if(InStr(session("intervieweeResponse" &
session("exerciseDisplayOrder" & currentExerciseNumber)), theWord) <>
0) then
            session("AnswerCorrectVsIntervieweeSame" &
session("exerciseDisplayOrder" & currentExerciseNumber)) =
session("AnswerCorrectVsIntervieweeSame" &
session("exerciseDisplayOrder" & currentExerciseNumber)) + 1
'Response.Write(" : Same")
        else
            session("AnswerCorrectVsIntervieweeDifferent" &
session("exerciseDisplayOrder" & currentExerciseNumber)) =
session("AnswerCorrectVsIntervieweeDifferent" &
session("exerciseDisplayOrder" & currentExerciseNumber)) + 1
'Response.Write(" : Different")
        end if
    end if
    theWord = ""
    else
        theWord = theWord & theChar
    end if
next
'Response.Write("<br>theword" & " : " & theWord)
'Response.Write("<br>intervieweeResponse" &
session("exerciseDisplayOrder" & currentExerciseNumber) & " : " &
session("intervieweeResponse" & session("exerciseDisplayOrder" &
currentExerciseNumber)))
'Response.Write("<br>correctAnswer" & session("exerciseDisplayOrder" &
currentExerciseNumber) & " : " & session("correctAnswer" &
session("exerciseDisplayOrder" & currentExerciseNumber)))
'Response.Write("<br>AnswerCorrectVsIntervieweeSame" &
session("exerciseDisplayOrder" & currentExerciseNumber) & " : " &
session("AnswerCorrectVsIntervieweeSame" &
session("exerciseDisplayOrder" & currentExerciseNumber)))
'Response.Write("<br>AnswerCorrectVsIntervieweeDifferent" &
session("exerciseDisplayOrder" & currentExerciseNumber) & " : " &
session("AnswerCorrectVsIntervieweeDifferent" &
session("exerciseDisplayOrder" & currentExerciseNumber)))

        currentExerciseNumber = currentExerciseNumber + 1 'this
"currentExerciseNumber" needs to stay the same
    end if
    updateReplyForExerciseNumber = currentExerciseNumber
end function

function verifyRadioButtonQuestionsAnswered()
    ' returns missing questions: ex. #2#, #8#, #11#, #12# (returns empty if
none missing)
    'functional flow
    'query QTIDs for valid radio button answers

```

```

    'compare valid QTIDs with those answered as 'responseX' where
X=QTID
    'missing QTIDs are added to the tempMissing variable
tempMissing = ""
for count = 1 to session("NumberOfQuestionnaireQuestions")
    tempNumber = session("questionnaireDisplayOrder" & count)
    if (Request.Form("Response" & tempNumber) = "") then
        tempMissing = tempMissing & "#" & tempNumber & "#,'"
    end if
next

'response.write("<br>You have not filled out the following multiple-
choice questions: " & tempMissing)
    verifyRadioButtonQuestionsAnswered = tempMissing

end function

%>

```


Appendix D.4 - thesisAdmin.asp

```
<%@ LANGUAGE=VBScript%>
<!-- #include file = "subAndFuncAdmin.asp" -->
<!-- #include file = "subroutinesAndFunctions.asp" -->
<%
couponStatus = "Coupon Status"
exerciseResults = "Exercise Results"
questionnaireResultsEssayType = "Questionnaire Results - Essay Type"
questionnaireResultsMultipleChoice = "Questionnaire Results - Multiple
Choice"
session("loginInformation") = "Login Information"
participants = "Participants"

if(Request.Form("submitButtonAdmin") <> "") then
    session("AdminPage") = Request.Form("submitButtonAdmin")
end if

if(Request.Form("submitButtonLogin") <> "") then
    session("AdminPage") = functionLogin("checkLogonCredentials")
end if

%>
<html>
<body>
<form method="post" action="thesisAdmin.asp">

<center><h1>Exercise and Survey Questionnaire Administration
Page</h1></center>

<% if(Request.form("submitButtonLogout") <> "") then %>
    <% session("AdminPage") = "" %>
    <% session("LoginID") = "" %>
<% elseif(session("AdminPage") <> "") then %>
<br>
<center>
<input type='submit' name='submitButtonAdmin'
value='<%=couponStatus%>'>
<input type='submit' name='submitButtonAdmin'
value='<%=exerciseResults%>'>
<input type='submit' name='submitButtonAdmin'
value='<%=session("loginInformation")%>'>
<input type='submit' name='submitButtonAdmin'
value='<%=questionnaireResultsEssayType%>'>
<input type='submit' name='submitButtonAdmin'
value='<%=questionnaireResultsMultipleChoice%>'>
<input type='submit' name='submitButtonAdmin'
value='<%=participants%>'>
<input type='submit' name='submitButtonLogout' value='Log Out'>
</center>
<br><br>
<% end if %>

<% if(session("AdminPage") = couponStatus) then %>
<center><b><h2><%=couponStatus%></h2></b></center>
```

```

<% subCouponStatus()%>
<% elseif(session("AdminPage") = exerciseResults) then %>
<center><b><h2><%=exerciseResults%></h2></b></center>
<% subExerciseResults()%>
<% elseif(session("AdminPage") = session("loginInformation")) then %>
<center><b><h2><%=session("loginInformation")%></h2></b></center>
<% subLoginInformation()%>
<% elseif(session("AdminPage") = questionnaireResultsEssayType) then %>
<center><b><h2><%=questionnaireResultsEssayType%></h2></b></center>
<% subQuestionnaireResultsEssayType()%>
<% elseif(session("AdminPage") = questionnaireResultsMultipleChoice)
then %>
<center><b><h2><%=questionnaireResultsMultipleChoice%></h2></b></center
>
<% subQuestionnaireResultsMultipleChoice()%>
<% elseif(session("AdminPage") = participants) then %>
<center><b><h2><%=participants%></h2></b></center>
<% subParticipants()%>
<% else %>
<center><b><h2><%=loginInformation%></h2></b></center>
<% session("AdminPage") = functionLogin("") %>
<% end if %>

</form>
</body>
</html>

```

Appendix D.5 - thesisDatabaseConnection.asp

```
<%  
Set Conn = Server.CreateObject("ADODB.Connection")  
Set RS = Server.CreateObject("ADODB.Recordset")  
Conn.Open "DSN=arrayItems;UID=sa;pwd=;"  
%>
```


APPENDIX E: THE EXERCISE RESULTS

Exercise Results: Participant Interview Dates

<u>Count</u>	<u>ID</u>	<u>Interview Date</u>	<u>Count</u>	<u>ID</u>	<u>Interview Date</u>
1	382	3/21/2005 13:27	42	423	4/8/2005 15:06
2	383	3/21/2005 19:53	43	424	4/9/2005 11:09
3	384	3/23/2005 10:43	44	425	4/9/2005 11:43
4	385	3/23/2005 12:19	45	426	4/9/2005 13:02
5	386	3/23/2005 12:50	46	427	4/11/2005 12:48
6	387	3/24/2005 8:13	47	428	4/11/2005 16:28
7	388	3/24/2005 8:28	48	429	4/11/2005 21:24
8	389	3/24/2005 8:46	49	430	4/12/2005 9:57
9	390	3/24/2005 9:01	50	431	4/12/2005 15:54
10	391	3/24/2005 10:30	51	432	4/13/2005 7:17
11	392	3/25/2005 14:38	52	433	4/13/2005 8:47
12	393	3/25/2005 16:18	53	434	4/13/2005 9:27
13	394	3/25/2005 16:27	54	435	4/13/2005 12:15
14	395	3/29/2005 13:21	55	436	4/13/2005 13:36
15	396	3/29/2005 13:48	56	437	4/13/2005 13:54
16	397	3/29/2005 15:59	57	438	4/13/2005 14:22
17	398	3/29/2005 20:50	58	439	4/13/2005 15:00
18	399	3/30/2005 9:35	59	440	4/13/2005 16:30
19	400	3/30/2005 9:49	60	441	4/13/2005 16:35
20	401	3/30/2005 12:34	61	442	4/13/2005 17:03
21	402	3/30/2005 12:45	62	443	4/13/2005 17:25
22	403	3/30/2005 14:52	63	444	4/13/2005 17:29
23	404	3/30/2005 15:52	64	445	4/13/2005 19:22
24	405	3/30/2005 21:38	65	446	4/13/2005 19:41
25	406	3/31/2005 14:29	66	447	4/14/2005 13:03
26	407	3/31/2005 17:11	67	448	4/14/2005 19:49
27	408	3/31/2005 19:26	68	449	4/15/2005 11:01
28	409	3/31/2005 22:07	69	450	4/15/2005 17:36
29	410	4/1/2005 12:14	70	451	4/16/2005 13:10
30	411	4/4/2005 0:48	71	452	4/17/2005 12:59
31	412	4/4/2005 21:32	72	453	4/18/2005 21:45
32	413	4/5/2005 19:21	73	454	4/19/2005 9:34
33	414	4/6/2005 20:15	74	455	4/19/2005 21:18
34	415	4/6/2005 20:59	75	456	4/20/2005 13:37
35	416	4/6/2005 21:17	76	457	4/20/2005 14:55
36	417	4/6/2005 21:29	77	458	4/21/2005 12:50
37	418	4/6/2005 23:50	78	459	4/28/2005 8:08
38	419	4/7/2005 9:50	79	460	5/10/2005 13:53
39	420	4/7/2005 14:49	80	461	5/15/2005 20:27
40	421	4/8/2005 12:07	81	462	5/16/2005 16:17
41	422	4/8/2005 13:56			

LEGEND

Count represents the chronological number for the participant.

ID represents the database primary key number for the participant.

Interview Date represents the date and time the participant completed the exercises and survey.

Exercise Results: Percent Correct on Single Column Sorting									
<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>COR</u>	<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>COR</u>
1	100	100	100	100	44	100	100	100	100
2	100	100	100	100	45	100	100	100	100
3	100	100	100	100	46	100	100	100	100
4	100	100	100	100	47	100	100	100	100
5	100	100	100	100	48	100	100	100	100
6	100	100	100	100	49	100	100	100	100
7	100	100	100	100	50	100	100	100	100
8	100	100	100	100	51	100	100	100	100
9	100	100	100	100	52	100	100	100	100
10	100	100	100	100	53	100	100	100	100
11	100	100	100	100	54	100	100	100	100
12	100	100	100	100	55	100	100	100	100
13	100	100	100	100	56	100	100	100	100
14	100	100	100	100	57	100	100	100	100
15	100	100	100	100	58	100	100	100	100
16	100	100	100	100	59	100	100	100	100
17	100	100	100	100	60	100	100	100	100
18	100	100	100	100	61	100	100	100	100
19	100	100	100	100	62	0	0	0	0
20	100	100	100	100	63	50	100	100	80
21	100	100	100	100	64	100	100	100	100
22	100	100	100	100	65	100	100	100	100
23	100	100	100	100	66	100	100	0	66
24	100	100	100	100	67	100	100	100	100
25	50	100	100	75	68	100	100	100	100
26	50	100	100	75	69	100	100	100	100
27	100	100	100	100	70	100	100	100	100
28	100	0	100	66	71	100	100	100	100
29	100	100	100	100	72	100	100	100	100
30	100	100	100	100	73	100	100	100	100
31	100	100	100	100	74	100	100	0	75
32	100	100	100	100	75	100	100	100	100
33	100	100	100	100	76	100	100	100	100
34	100	100	100	100	77	100	100	100	100
35	100	100	100	100	78	100	100	100	100
36	100	100	100	100	79	100	100	100	100
37	100	100	100	100	80	100	100	100	100
38	100	100	100	100	81	100	100	100	100
39	100	100	100	100					
40	100	100	100	100	TOTAL	N	S	D	COR
41	100	100	100	100	AVG	96	97	96	96
42	100	100	100	100					
43	100	100	100	100					

Legend

Count represents the chronological number for the participant.

N represents the questions giving a Subject and Date. The results would provide the **Name**.

S represents the questions giving a Name and Date. The results would provide the **Subject**.

D represents the questions giving a Name and Subject. The results would provide the **Date**.

COR is the average number of correctly chosen records. If the total number of correctly chosen records was 4 and the total possible number of records was 5, the COR would be 80%.

TOTAL takes into account all records.

AVG is the average.

Exercise Results: Percent Incorrect on Single Column Sorting									
<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>INC</u>	<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>INC</u>
1	0	100	0	25	44	100	0	0	33
2	0	0	0	0	45	0	0	0	0
3	0	0	0	0	46	0	0	0	0
4	0	0	0	0	47	0	0	0	0
5	0	0	0	0	48	0	0	0	0
6	0	0	0	0	49	0	0	0	0
7	0	0	0	0	50	0	0	0	0
8	0	0	0	0	51	0	0	0	0
9	0	0	0	0	52	300	5100	300	1500
10	0	0	0	0	53	0	0	0	0
11	0	0	0	0	54	0	0	0	0
12	0	0	0	0	55	0	0	0	0
13	0	0	0	0	56	0	0	0	0
14	0	0	100	25	57	0	0	0	0
15	0	0	100	33	58	0	0	0	0
16	0	0	0	0	59	0	0	0	0
17	0	0	0	0	60	200	800	100	420
18	0	0	0	0	61	0	0	200	66
19	0	0	0	0	62	1650	1000	8600	3225
20	0	0	0	0	63	50	0	0	20
21	0	2400	100	833	64	0	0	100	25
22	0	0	0	0	65	0	0	0	0
23	0	0	0	0	66	0	0	100	33
24	0	0	0	0	67	0	0	0	0
25	50	0	0	25	68	0	0	0	0
26	50	0	0	25	69	0	1500	0	107
27	0	0	0	0	70	0	0	800	200
28	0	100	0	33	71	0	0	0	0
29	100	0	300	125	72	0	0	100	33
30	0	0	0	0	73	0	0	0	0
31	0	0	0	0	74	0	0	100	25
32	0	0	0	0	75	0	0	0	0
33	0	0	0	0	76	0	0	0	0
34	0	0	0	0	77	500	1750	500	1125
35	0	0	0	0	78	0	0	0	0
36	0	0	0	0	79	0	0	0	0
37	0	0	0	0	80	0	0	0	0
38	0	0	0	0	81	0	0	0	0
39	0	0	0	0					
40	0	0	0	0	TOTAL	N	S	D	INC
41	0	0	0	0	AVG	37	157	141	97
42	0	0	0	0					
43	0	0	0	0					

Legend

Count represents the chronological number for the participant.

N represents the questions giving a Subject and Date. The results would provide the **Name**.

S represents the questions giving a Name and Date. The results would provide the **Subject**.

D represents the questions giving a Name and Subject. The results would provide the **Date**.

INC is the average number of incorrectly chosen records. If the total number of incorrectly chosen records was 1 and the total possible number of records was 5, the INC would be 20%.

TOTAL takes into account all records.

AVG is the average.

Exercise Results: Percent Correct on Multiple Column Sorting (Experimental Method)									
<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>COR</u>	<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>COR</u>
1	100	100	100	100	44	100	100	100	100
2	100	100	100	100	45	100	100	100	100
3	100	50	100	75	46	100	100	100	100
4	100	100	100	100	47	100	100	100	100
5	100	100	100	100	48	100	100	100	100
6	100	100	100	100	49	100	100	100	100
7	100	100	100	100	50	100	100	100	100
8	100	100	100	100	51	100	50	100	93
9	100	100	100	100	52	100	100	100	100
10	100	100	100	100	53	100	100	100	100
11	100	100	100	100	54	100	100	100	100
12	100	100	100	100	55	100	100	100	100
13	100	100	100	100	56	91	100	100	92
14	100	100	100	100	57	100	100	100	100
15	100	100	100	100	58	100	100	100	100
16	100	100	100	100	59	100	100	100	100
17	100	100	100	100	60	100	100	100	100
18	100	100	50	75	61	100	100	100	100
19	100	100	50	80	62	41	0	100	42
20	100	100	100	100	63	0	100	0	33
21	100	100	100	100	64	100	100	100	100
22	100	100	100	100	65	100	100	100	100
23	100	100	100	100	66	100	100	100	100
24	100	100	100	100	67	100	100	100	100
25	100	100	100	100	68	100	100	100	100
26	100	100	100	100	69	100	100	100	100
27	100	100	100	100	70	100	100	100	100
28	100	100	100	100	71	100	100	100	100
29	100	100	100	100	72	100	100	100	100
30	50	100	100	80	73	100	100	100	100
31	50	100	100	75	74	100	100	100	100
32	50	100	100	75	75	100	100	100	100
33	100	100	100	100	76	100	100	100	100
34	100	100	100	100	77	100	100	100	100
35	100	100	100	100	78	100	100	100	100
36	50	100	100	66	79	100	100	100	100
37	100	100	100	100	80	100	100	100	100
38	100	100	100	100	81	100	100	100	100
39	91	100	100	93					
40	100	100	100	100	TOTAL	N	S	D	COR
41	100	100	100	100	AVG	95	97	97	96
42	100	100	100	100					
43	100	100	100	100					

Legend

Count represents the chronological number for the participant.

N represents the questions giving a Subject and Date. The results would provide the **Name**.

S represents the questions giving a Name and Date. The results would provide the **Subject**.

D represents the questions giving a Name and Subject. The results would provide the **Date**.

COR represents the average number of correctly chosen records. If the total number of correctly chosen records was 4 and the total possible number of records was 5, the COR would be 80%.

TOTAL takes into account all records.

AVG is the average.

Exercise Results: Percent Incorrect on Multiple Column Sorting (Experimental Method)									
<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>INC</u>	<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>INC</u>
1	0	0	0	0	44	0	0	0	0
2	0	0	0	0	45	0	400	0	160
3	0	50	0	25	46	0	0	0	0
4	0	0	0	0	47	0	0	0	0
5	0	0	0	0	48	0	0	0	0
6	0	0	0	0	49	0	0	0	0
7	0	0	0	0	50	0	0	0	0
8	0	0	0	0	51	0	50	0	6
9	0	0	0	0	52	3900	7900	100	3950
10	0	0	0	0	53	0	0	0	0
11	0	0	0	0	54	0	0	0	0
12	0	0	0	0	55	0	0	0	0
13	0	0	0	0	56	8	0	0	7
14	0	0	0	0	57	0	0	0	0
15	0	0	0	0	58	0	0	0	0
16	0	0	0	0	59	0	0	0	0
17	0	0	0	0	60	2500	2700	800	2000
18	0	0	50	25	61	0	0	0	0
19	0	0	50	20	62	975	14800	5000	2250
20	0	0	0	0	63	200	800	100	366
21	0	19900	1100	2009	64	100	0	0	33
22	0	0	0	0	65	0	0	0	0
23	0	0	100	25	66	0	0	100	25
24	0	0	0	0	67	0	0	0	0
25	0	0	0	0	68	0	0	0	0
26	0	0	0	0	69	0	300	0	75
27	0	0	0	0	70	0	0	0	0
28	0	0	0	0	71	0	0	0	0
29	0	0	0	0	72	0	0	0	0
30	50	0	0	20	73	0	0	0	0
31	50	0	0	25	74	0	0	0	0
32	50	0	0	25	75	0	0	0	0
33	0	0	0	0	76	0	0	0	0
34	0	0	0	0	77	1600	800	500	966
35	0	0	0	0	78	0	0	200	50
36	50	0	0	33	79	0	0	0	0
37	0	0	0	0	80	0	0	0	0
38	0	0	0	0	81	0	0	0	0
39	8	0	0	6					
40	0	0	0	0	TOTAL	N	S	D	INC
41	0	0	0	0	AVG	117	588	100	149
42	0	0	0	0					
43	0	0	0	0					

Legend

Count represents the chronological number for the participant.

N represents the questions giving a Subject and Date. The results would provide the **Name**.

S represents the questions giving a Name and Date. The results would provide the **Subject**.

D represents the questions giving a Name and Subject. The results would provide the **Date**.

INC is the average number of incorrectly chosen records. If the total number of incorrectly chosen records was 1 and the total possible number of records was 5, the INC would be 20%.

TOTAL takes into account all records.

AVG is the average.

Exercise Results: Time (seconds) Taken on Single Column Sorting									
<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>TOT</u>	<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>TOT</u>
1	32	41	28	101	44	121	146	109	376
2	45	89	146	280	45	70	53	74	197
3	54	89	55	198	46	84	89	65	238
4	49	70	71	190	47	123	107	105	335
5	61	60	53	174	48	115	35	74	224
6	29	53	38	120	49	204	68	110	382
7	162	38	36	236	50	52	85	78	215
8	48	100	40	188	51	44	62	184	290
9	75	34	44	153	52	76	79	316	471
10	79	56	48	183	53	93	68	77	238
11	124	77	68	269	54	162	20	32	214
12	79	48	82	209	55	280	88	90	458
13	107	47	63	217	56	194	49	33	276
14	75	68	97	240	57	98	35	85	218
15	212	79	65	356	58	69	68	70	207
16	40	56	59	155	59	60	65	59	184
17	57	60	55	172	60	70	75	41	186
18	76	32	48	156	61	33	79	83	195
19	21	41	55	117	62	63	23	109	195
20	46	43	28	117	63	63	68	29	160
21	187	113	63	363	64	89	69	119	277
22	109	58	54	221	65	86	80	35	201
23	100	100	96	296	66	34	75	58	167
24	36	51	37	124	67	266	88	108	462
25	53	34	34	121	68	39	42	50	131
26	112	84	57	253	69	33	44	27	104
27	40	54	61	155	70	84	79	442	605
28	64	134	730	928	71	92	49	60	201
29	180	120	98	398	72	57	72	64	193
30	100	33	38	171	73	30	24	33	87
31	93	138	195	426	74	26	118	58	202
32	31	64	35	130	75	46	62	33	141
33	40	35	71	146	76	38	56	56	150
34	65	74	41	180	77	53	126	89	268
35	104	166	45	315	78	112	180	816	1108
36	36	51	77	164	79	153	315	28	496
37	61	117	136	314	80	55	57	70	182
38	89	138	67	294	81	60	42	59	161
39	16	24	115	155					
40	26	50	60	136	TOTAL	N	S	D	TOT
41	142	38	40	220	AVG	82	73	95	252
42	50	92	367	509					
43	74	48	48	170					

Legend

Count represents the chronological number for the participant.

N represents the questions giving a Subject and Date. The results would provide the **Name**.

S represents the questions giving a Name and Date. The results would provide the **Subject**.

D represents the questions giving a Name and Subject. The results would provide the **Date**.

TOTAL takes into account all records.

AVG is the average.

Exercise Results: Time (seconds) Taken on Multiple Column Sorting (Experimental Method)									
<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>TOT</u>	<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>TOT</u>
1	40	48	26	114	44	68	69	82	219
2	103	86	100	289	45	29	77	33	139
3	125	57	57	239	46	42	57	86	185
4	93	60	49	202	47	78	108	191	377
5	47	34	118	199	48	100	88	86	274
6	46	34	29	109	49	90	51	69	210
7	71	46	52	169	50	80	115	30	225
8	44	43	66	153	51	35	63	26	124
9	28	57	24	109	52	167	256	80	503
10	69	75	56	200	53	54	180	59	293
11	171	85	50	306	54	51	50	41	142
12	238	42	132	412	55	438	42	56	536
13	40	105	101	246	56	194	46	324	564
14	71	134	61	266	57	40	50	86	176
15	50	77	106	233	58	66	58	94	218
16	108	67	97	272	59	74	298	45	417
17	97	89	70	256	60	300	82	49	431
18	66	82	48	196	61	81	29	36	146
19	40	120	85	245	62	152	161	68	381
20	37	52	54	143	63	210	84	144	438
21	93	543	78	714	64	110	64	35	209
22	63	54	50	167	65	53	76	142	271
23	70	67	94	231	66	31	56	70	157
24	70	78	40	188	67	48	47	41	136
25	45	35	66	146	68	43	44	58	145
26	107	157	127	391	69	30	27	33	90
27	24	64	73	161	70	49	95	172	316
28	60	80	97	237	71	68	45	44	157
29	95	70	86	251	72	51	52	77	180
30	282	86	48	416	73	24	82	35	141
31	384	132	352	868	74	43	31	122	196
32	55	28	31	114	75	55	142	99	296
33	73	36	45	154	76	42	48	38	128
34	120	452	112	684	77	70	76	81	227
35	43	50	33	126	78	114	53	424	591
36	93	41	36	170	79	49	40	55	144
37	64	56	87	207	80	49	67	141	257
38	40	52	39	131	81	46	75	43	164
39	24	41	54	119					
40	59	34	49	142	TOTAL	N	S	D	TOT
41	58	77	47	182	AVG	85	83	80	249
42	44	48	29	121					
43	66	40	50	156					

Legend

Count represents the chronological number for the participant.

N represents the questions giving a Subject and Date. The results would provide the **Name**.

S represents the questions giving a Name and Date. The results would provide the **Subject**.

D represents the questions giving a Name and Subject. The results would provide the **Date**.

TOTAL takes into account all records.

AVG is the average.

Exercise Results: Time (seconds) Total and Comparisons										
<u>Count</u>	<u>Total</u>	<u>S/T %</u>	<u>M/T %</u>	<u>M/S %</u>	<u>Count</u>	<u>Total</u>	<u>S/T %</u>	<u>M/T %</u>	<u>M/S %</u>	
1	215	46	53	1.12	44	595	63	36	0.58	
2	569	49	50	1.03	45	336	58	41	0.7	
3	437	45	54	1.2	46	423	56	43	0.77	
4	392	48	51	1.06	47	712	47	52	1.12	
5	373	46	53	1.14	48	498	44	55	1.22	
6	229	52	47	0.9	49	592	64	35	0.54	
7	405	58	41	0.71	50	440	48	51	1.04	
8	341	55	44	0.81	51	414	70	29	0.42	
9	262	58	41	0.71	52	974	48	51	1.06	
10	383	47	52	1.09	53	531	44	55	1.23	
11	575	46	53	1.13	54	356	60	39	0.66	
12	621	33	66	1.97	55	994	46	53	1.17	
13	463	46	53	1.13	56	840	32	67	2.04	
14	506	47	52	1.1	57	394	55	44	0.8	
15	589	60	39	0.65	58	425	48	51	1.05	
16	427	36	63	1.75	59	601	30	69	2.26	
17	428	40	59	1.48	60	617	30	69	2.31	
18	352	44	55	1.25	61	341	57	42	0.74	
19	362	32	67	2.09	62	576	33	66	1.95	
20	260	45	55	1.22	63	598	26	73	2.73	
21	1077	33	66	1.96	64	486	56	43	0.75	
22	388	56	43	0.75	65	472	42	57	1.34	
23	527	56	43	0.78	66	324	51	48	0.94	
24	312	39	60	1.51	67	598	77	22	0.29	
25	267	45	54	1.2	68	276	47	52	1.1	
26	644	39	60	1.54	69	194	53	46	0.86	
27	316	49	50	1.03	70	921	65	34	0.52	
28	1165	79	20	0.25	71	358	56	43	0.78	
29	649	61	38	0.63	72	373	51	48	0.93	
30	587	29	70	2.43	73	228	38	61	1.62	
31	1294	32	67	2.03	74	398	50	49	0.97	
32	244	53	46	0.87	75	437	32	67	2.09	
33	300	48	51	1.05	76	278	53	46	0.85	
34	864	20	79	3.8	77	495	54	45	0.84	
35	441	71	28	0.4	78	1699	65	34	0.53	
36	334	49	50	1.03	79	640	77	22	0.29	
37	521	60	39	0.65	80	439	41	58	1.41	
38	425	69	30	0.44	81	325	49	50	1.01	
39	274	56	43	0.76						
40	278	48	51	1.04	TOTAL	Total	S/T %	M/T %	M/S %	
41	402	54	45	0.82	AVG	501	49	49	1.12	
42	630	80	19	0.23						
43	326	52	47	0.91						

Legend

Count represents the chronological number for the participant.
Total (lowercase) and T: The total number of seconds to complete the exercises.
S represents the **standard** sort method. S/T is (total standard time) / (total stand + mult time)
M represents the **multi-column** sort method. S/T is (total mult. time) / (total stand + mult. time)
TOTAL (uppercase) takes into account all records.
AVG is the average.

Exercise Results: Time (seconds) on Questionnaire and Total for Questionnaire and Exercises

<u>Count</u>	<u>Qstnr</u>	<u>Exer/Qstnr</u>	<u>Count</u>	<u>Qstnr</u>	<u>Exer/Qstnr</u>
1	162	377	44	152	747
2	179	748	45	202	538
3	245	682	46	133	556
4	193	585	47	433	1145
5	160	533	48	216	714
6	178	407	49	121	713
7	109	514	50	204	644
8	96	437	51	212	626
9	98	360	52	175	1149
10	239	622	53	173	704
11	204	779	54	83	439
12	164	785	55	181	1175
13	315	778	56	182	1022
14	214	720	57	139	533
15	262	851	58	334	759
16	431	858	59	128	729
17	218	646	60	152	769
18	194	546	61	198	539
19	265	627	62	63	639
20	99	359	63	173	771
21	352	1429	64	209	695
22	305	693	65	150	622
23	57	584	66	190	514
24	190	502	67	251	849
25	18	285	68	131	407
26	222	866	69	119	313
27	179	495	70	191	1112
28	396	1561	71	278	636
29	380	1029	72	152	525
30	378	965	73	104	332
31	336	1630	74	272	670
32	178	422	75	148	585
33	114	414	76	146	424
34	245	1109	77	178	673
35	197	638	78	888	2587
36	525	859	79	153	793
37	429	950	80	179	618
38	164	589	81	169	494
39	177	451			
40	194	472	<u>TOTAL</u>	<u>Qstnr</u>	<u>E/Q</u>
41	201	603	AVG	211	713
42	156	786			
43	157	483			

Legend

Count represents the chronological number for the participant.

Qstnr represents the time it took participants to complete the questionnaire.

E/Q represents the time it took participants to complete both the exercises and the questionnaire.

TOTAL takes into account all records.

AVG is the average.

APPENDIX F: EXERCISE RESULTS - NO OUTLIERS / OUTLIERS REMOVED

NO OUTLIERS

Exercise Results: Percent Correct on Single Column Sorting

<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>COR</u>	<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>COR</u>
1	100	100	100	100	44	100	100	100	100
2	100	100	100	100	45	100	100	100	100
3	100	100	100	100	46	100	100	100	100
4	100	100	100	100	47	100	100	100	100
5	100	100	100	100	48	100	100	100	100
6	100	100	100	100	49	100	100	100	100
7	100	100	100	100	50	100	100	100	100
8	100	100	100	100	51	100	100	100	100
9	100	100	100	100	53	100	100	100	100
10	100	100	100	100	54	100	100	100	100
11	100	100	100	100	55	100	100	100	100
12	100	100	100	100	56	100	100	100	100
13	100	100	100	100	57	100	100	100	100
14	100	100	100	100	58	100	100	100	100
15	100	100	100	100	59	100	100	100	100
16	100	100	100	100	61	100	100	100	100
17	100	100	100	100	63	50	100	100	80
18	100	100	100	100	64	100	100	100	100
19	100	100	100	100	65	100	100	100	100
20	100	100	100	100	66	100	100	0	66
22	100	100	100	100	67	100	100	100	100
23	100	100	100	100	68	100	100	100	100
24	100	100	100	100	69	100	100	100	100
25	50	100	100	75	70	100	100	100	100
26	50	100	100	75	71	100	100	100	100
27	100	100	100	100	72	100	100	100	100
28	100	0	100	66	73	100	100	100	100
29	100	100	100	100	74	100	100	0	75
30	100	100	100	100	75	100	100	100	100
31	100	100	100	100	76	100	100	100	100
32	100	100	100	100	78	100	100	100	100
33	100	100	100	100	79	100	100	100	100
34	100	100	100	100	80	100	100	100	100
35	100	100	100	100	81	100	100	100	100
36	100	100	100	100					
37	100	100	100	100	<u>TOTAL</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>COR</u>
38	100	100	100	100	AVG	98.03	98.68	97.37	97.86
39	100	100	100	100					
40	100	100	100	100					
41	100	100	100	100					
42	100	100	100	100					
43	100	100	100	100					

Legend

Count represents the chronological number for the participant.

N represents the questions giving a Subject and Date. The results would provide the **Name**.

S represents the questions giving a Name and Date. The results would provide the **Subject**.

D represents the questions giving a Name and Subject. The results would provide the **Date**.

COR represents the average number of correctly chosen records. If the total number of correctly chosen records was 4 and the total possible number of records was 5, the COR would be 80%.

TOTAL takes into account all records.

AVG is the average.

NO OUTLIERS

Exercise Results: Percent Incorrect on Single Column Sorting

<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>INC</u>	<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>INC</u>
1	0	100	0	25	44	100	0	0	33
2	0	0	0	0	45	0	0	0	0
3	0	0	0	0	46	0	0	0	0
4	0	0	0	0	47	0	0	0	0
5	0	0	0	0	48	0	0	0	0
6	0	0	0	0	49	0	0	0	0
7	0	0	0	0	50	0	0	0	0
8	0	0	0	0	51	0	0	0	0
9	0	0	0	0	53	0	0	0	0
10	0	0	0	0	54	0	0	0	0
11	0	0	0	0	55	0	0	0	0
12	0	0	0	0	56	0	0	0	0
13	0	0	0	0	57	0	0	0	0
14	0	0	100	25	58	0	0	0	0
15	0	0	100	33	59	0	0	0	0
16	0	0	0	0	61	0	0	200	66
17	0	0	0	0	63	50	0	0	20
18	0	0	0	0	64	0	0	100	25
19	0	0	0	0	65	0	0	0	0
20	0	0	0	0	66	0	0	100	33
22	0	0	0	0	67	0	0	0	0
23	0	0	0	0	68	0	0	0	0
24	0	0	0	0	69	0	1500	0	107
25	50	0	0	25	70	0	0	800	200
26	50	0	0	25	71	0	0	0	0
27	0	0	0	0	72	0	0	100	33
28	0	100	0	33	73	0	0	0	0
29	100	0	300	125	74	0	0	100	25
30	0	0	0	0	75	0	0	0	0
31	0	0	0	0	76	0	0	0	0
32	0	0	0	0	78	0	0	0	0
33	0	0	0	0	79	0	0	0	0
34	0	0	0	0	80	0	0	0	0
35	0	0	0	0	81	0	0	0	0
36	0	0	0	0					
37	0	0	0	0	TOTAL	N	S	D	INC
38	0	0	0	0	AVG	4.61	22.37	25.00	10.96
39	0	0	0	0					
40	0	0	0	0					
41	0	0	0	0					
42	0	0	0	0					
43	0	0	0	0					

Legend

Count represents the chronological number for the participant.
N represents the questions giving a Subject and Date. The results would provide the **Name**.
S represents the questions giving a Name and Date. The results would provide the **Subject**.
D represents the questions giving a Name and Subject. The results would provide the **Date**.
INC is the average number of incorrectly chosen records. If the total number of incorrectly chosen records was 1 and the total possible number of records was 5, the INC would be 20%.
TOTAL takes into account all records.
AVG is the average.

NO OUTLIERS

Exercise Results: Percent Correct on Multiple Column Sorting (Experimental Method)

<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>COR</u>	<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>COR</u>
1	100	100	100	100	44	100	100	100	100
2	100	100	100	100	45	100	100	100	100
3	100	50	100	75	46	100	100	100	100
4	100	100	100	100	47	100	100	100	100
5	100	100	100	100	48	100	100	100	100
6	100	100	100	100	49	100	100	100	100
7	100	100	100	100	50	100	100	100	100
8	100	100	100	100	51	100	50	100	93
9	100	100	100	100	53	100	100	100	100
10	100	100	100	100	54	100	100	100	100
11	100	100	100	100	55	100	100	100	100
12	100	100	100	100	56	91	100	100	92
13	100	100	100	100	57	100	100	100	100
14	100	100	100	100	58	100	100	100	100
15	100	100	100	100	59	100	100	100	100
16	100	100	100	100	61	100	100	100	100
17	100	100	100	100	63	0	100	0	33
18	100	100	50	75	64	100	100	100	100
19	100	100	50	80	65	100	100	100	100
20	100	100	100	100	66	100	100	100	100
22	100	100	100	100	67	100	100	100	100
23	100	100	100	100	68	100	100	100	100
24	100	100	100	100	69	100	100	100	100
25	100	100	100	100	70	100	100	100	100
26	100	100	100	100	71	100	100	100	100
27	100	100	100	100	72	100	100	100	100
28	100	100	100	100	73	100	100	100	100
29	100	100	100	100	74	100	100	100	100
30	50	100	100	80	75	100	100	100	100
31	50	100	100	75	76	100	100	100	100
32	50	100	100	75	78	100	100	100	100
33	100	100	100	100	79	100	100	100	100
34	100	100	100	100	80	100	100	100	100
35	100	100	100	100	81	100	100	100	100
36	50	100	100	66					
37	100	100	100	100	<u>TOTAL</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>COR</u>
38	100	100	100	100	AVG	95.82	98.68	97.37	96.54
39	91	100	100	93					
40	100	100	100	100					
41	100	100	100	100					
42	100	100	100	100					
43	100	100	100	100					

Legend

Count represents the chronological number for the participant.

N represents the questions giving a Subject and Date. The results would provide the **Name**.

S represents the questions giving a Name and Date. The results would provide the **Subject**.

D represents the questions giving a Name and Subject. The results would provide the **Date**.

COR represents the average number of correctly chosen records. If the total number of correctly chosen records was 4 and the total possible number of records was 5, the COR would be 80%.

TOTAL takes into account all records.

AVG is the average.

NO OUTLIERS

Exercise Results: Percent Incorrect on Multiple Column Sorting (Experimental Method)

<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>INC</u>	<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>INC</u>
1	0	0	0	0	44	0	0	0	0
2	0	0	0	0	45	0	400	0	160
3	0	50	0	25	46	0	0	0	0
4	0	0	0	0	47	0	0	0	0
5	0	0	0	0	48	0	0	0	0
6	0	0	0	0	49	0	0	0	0
7	0	0	0	0	50	0	0	0	0
8	0	0	0	0	51	0	50	0	6
9	0	0	0	0	53	0	0	0	0
10	0	0	0	0	54	0	0	0	0
11	0	0	0	0	55	0	0	0	0
12	0	0	0	0	56	8	0	0	7
13	0	0	0	0	57	0	0	0	0
14	0	0	0	0	58	0	0	0	0
15	0	0	0	0	59	0	0	0	0
16	0	0	0	0	61	0	0	0	0
17	0	0	0	0	63	200	800	100	366
18	0	0	50	25	64	100	0	0	33
19	0	0	50	20	65	0	0	0	0
20	0	0	0	0	66	0	0	100	25
22	0	0	0	0	67	0	0	0	0
23	0	0	100	25	68	0	0	0	0
24	0	0	0	0	69	0	300	0	75
25	0	0	0	0	70	0	0	0	0
26	0	0	0	0	71	0	0	0	0
27	0	0	0	0	72	0	0	0	0
28	0	0	0	0	73	0	0	0	0
29	0	0	0	0	74	0	0	0	0
30	50	0	0	20	75	0	0	0	0
31	50	0	0	25	76	0	0	0	0
32	50	0	0	25	78	0	0	200	50
33	0	0	0	0	79	0	0	0	0
34	0	0	0	0	80	0	0	0	0
35	0	0	0	0	81	0	0	0	0
36	50	0	0	33					
37	0	0	0	0	<u>TOTAL</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>INC</u>
38	0	0	0	0	AVG	6.79	21.05	7.89	12.18
39	8	0	0	6					
40	0	0	0	0					
41	0	0	0	0					
42	0	0	0	0					
43	0	0	0	0					

Legend

Count represents the chronological number for the participant.

N represents the questions giving a Subject and Date. The results would provide the **Name**.

S represents the questions giving a Name and Date. The results would provide the **Subject**.

D represents the questions giving a Name and Subject. The results would provide the **Date**.

INC is the average number of incorrectly chosen records. If the total number of incorrectly chosen records was 1 and the total possible number of records was 5, the INC would be 20%.

TOTAL takes into account all records.

AVG is the average.

NO OUTLIERS

Exercise Results: Time (seconds) Taken on Single Column Sorting

<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>TOT</u>	<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>TOT</u>
1	32	41	28	101	44	121	146	109	376
2	45	89	146	280	45	70	53	74	197
3	54	89	55	198	46	84	89	65	238
4	49	70	71	190	47	123	107	105	335
5	61	60	53	174	48	115	35	74	224
6	29	53	38	120	49	204	68	110	382
7	162	38	36	236	50	52	85	78	215
8	48	100	40	188	51	44	62	184	290
9	75	34	44	153	53	93	68	77	238
10	79	56	48	183	54	162	20	32	214
11	124	77	68	269	55	280	88	90	458
12	79	48	82	209	56	194	49	33	276
13	107	47	63	217	57	98	35	85	218
14	75	68	97	240	58	69	68	70	207
15	212	79	65	356	59	60	65	59	184
16	40	56	59	155	61	33	79	83	195
17	57	60	55	172	63	63	68	29	160
18	76	32	48	156	64	89	69	119	277
19	21	41	55	117	65	86	80	35	201
20	46	43	28	117	66	34	75	58	167
22	109	58	54	221	67	266	88	108	462
23	100	100	96	296	68	39	42	50	131
24	36	51	37	124	69	33	44	27	104
25	53	34	34	121	70	84	79	442	605
26	112	84	57	253	71	92	49	60	201
27	40	54	61	155	72	57	72	64	193
28	64	134	730	928	73	30	24	33	87
29	180	120	98	398	74	26	118	58	202
30	100	33	38	171	75	46	62	33	141
31	93	138	195	426	76	38	56	56	150
32	31	64	35	130	78	112	180	816	1108
33	40	35	71	146	79	153	315	28	496
34	65	74	41	180	80	55	57	70	182
35	104	166	45	315	81	60	42	59	161
36	36	51	77	164					
37	61	117	136	314	TOTAL	N	S	D	TOT
38	89	138	67	294	AVG	82.33	72.64	94.13	249.11
39	16	24	115	155					
40	26	50	60	136					
41	142	38	40	220					
42	50	92	367	509					
43	74	48	48	170					

Legend

Count represents the chronological number for the participant.

N represents the questions giving a Subject and Date. The results would provide the **Name**.

S represents the questions giving a Name and Date. The results would provide the **Subject**.

D represents the questions giving a Name and Subject. The results would provide the **Date**.

TOTAL takes into account all records.

AVG is the average.

NO OUTLIERS

Exercise Results: Time (seconds) Taken on Multiple Column Sorting (Experimental Method)

<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>TOT</u>	<u>Count</u>	<u>N</u>	<u>S</u>	<u>D</u>	<u>TOT</u>
1	40	48	26	114	44	68	69	82	219
2	103	86	100	289	45	29	77	33	139
3	125	57	57	239	46	42	57	86	185
4	93	60	49	202	47	78	108	191	377
5	47	34	118	199	48	100	88	86	274
6	46	34	29	109	49	90	51	69	210
7	71	46	52	169	50	80	115	30	225
8	44	43	66	153	51	35	63	26	124
9	28	57	24	109	53	54	180	59	293
10	69	75	56	200	54	51	50	41	142
11	171	85	50	306	55	438	42	56	536
12	238	42	132	412	56	194	46	324	564
13	40	105	101	246	57	40	50	86	176
14	71	134	61	266	58	66	58	94	218
15	50	77	106	233	59	74	298	45	417
16	108	67	97	272	61	81	29	36	146
17	97	89	70	256	63	210	84	144	438
18	66	82	48	196	64	110	64	35	209
19	40	120	85	245	65	53	76	142	271
20	37	52	54	143	66	31	56	70	157
22	63	54	50	167	67	48	47	41	136
23	70	67	94	231	68	43	44	58	145
24	70	78	40	188	69	30	27	33	90
25	45	35	66	146	70	49	95	172	316
26	107	157	127	391	71	68	45	44	157
27	24	64	73	161	72	51	52	77	180
28	60	80	97	237	73	24	82	35	141
29	95	70	86	251	74	43	31	122	196
30	282	86	48	416	75	55	142	99	296
31	384	132	352	868	76	42	48	38	128
32	55	28	31	114	78	114	53	424	591
33	73	36	45	154	79	49	40	55	144
34	120	452	112	684	80	49	67	141	257
35	43	50	33	126	81	46	75	43	164
36	93	41	36	170					
37	64	56	87	207	TOTAL	N	S	D	TOT
38	40	52	39	131	AVG	81.03	74.74	80.83	236.59
39	24	41	54	119					
40	59	34	49	142					
41	58	77	47	182					
42	44	48	29	121					
43	66	40	50	156					

Legend

Count represents the chronological number for the participant.

N represents the questions giving a Subject and Date. The results would provide the **Name**.

S represents the questions giving a Name and Date. The results would provide the **Subject**.

D represents the questions giving a Name and Subject. The results would provide the **Date**.

TOTAL takes into account all records.

AVG is the average.

NO OUTLIERS

Exercise Results: Time (seconds) Total and Comparisons

<u>Count</u>	<u>Total</u>	<u>S/T %</u>	<u>M/T %</u>	<u>M/S %</u>	<u>Count</u>	<u>Total</u>	<u>S/T %</u>	<u>M/T %</u>	<u>M/S %</u>
1	215	46	53	1.12	44	595	63	36	0.58
2	569	49	50	1.03	45	336	58	41	0.7
3	437	45	54	1.2	46	423	56	43	0.77
4	392	48	51	1.06	47	712	47	52	1.12
5	373	46	53	1.14	48	498	44	55	1.22
6	229	52	47	0.9	49	592	64	35	0.54
7	405	58	41	0.71	50	440	48	51	1.04
8	341	55	44	0.81	51	414	70	29	0.42
9	262	58	41	0.71	53	531	44	55	1.23
10	383	47	52	1.09	54	356	60	39	0.66
11	575	46	53	1.13	55	994	46	53	1.17
12	621	33	66	1.97	56	840	32	67	2.04
13	463	46	53	1.13	57	394	55	44	0.8
14	506	47	52	1.1	58	425	48	51	1.05
15	589	60	39	0.65	59	601	30	69	2.26
16	427	36	63	1.75	61	341	57	42	0.74
17	428	40	59	1.48	63	598	26	73	2.73
18	352	44	55	1.25	64	486	56	43	0.75
19	362	32	67	2.09	65	472	42	57	1.34
20	260	45	55	1.22	66	324	51	48	0.94
22	388	56	43	0.75	67	598	77	22	0.29
23	527	56	43	0.78	68	276	47	52	1.1
24	312	39	60	1.51	69	194	53	46	0.86
25	267	45	54	1.2	70	921	65	34	0.52
26	644	39	60	1.54	71	358	56	43	0.78
27	316	49	50	1.03	72	373	51	48	0.93
28	1165	79	20	0.25	73	228	38	61	1.62
29	649	61	38	0.63	74	398	50	49	0.97
30	587	29	70	2.43	75	437	32	67	2.09
31	1294	32	67	2.03	76	278	53	46	0.85
32	244	53	46	0.87	78	1699	65	34	0.53
33	300	48	51	1.05	79	640	77	22	0.29
34	864	20	79	3.8	80	439	41	58	1.41
35	441	71	28	0.4	81	325	49	50	1.01
36	334	49	50	1.03					
37	521	60	39	0.65	TOTAL	Total	S/T %	M/T %	M/S %
38	425	69	30	0.44	AVG	485.70	50.38	48.63	1.09
39	274	56	43	0.76					
40	278	48	51	1.04					
41	402	54	45	0.82					
42	630	80	19	0.23					
43	326	52	47	0.91					

Legend

Count represents the chronological number for the participant.

Total (lowercase) and T: The total number of seconds to complete the exercises.

S represents the **standard** sort method. S/T is (total standard time) / (total stand + mult time)

M represents the **multi-column** sort method. S/T is (total mult. time) / (total stand + mult. time)

TOTAL (uppercase) takes into account all records.

AVG is the average.

NO OUTLIERS

Exercise Results: Time (seconds) on Questionnaire and Total for Questionnaire and Exercises

<u>Count</u>	<u>Qstnr</u>	<u>E/Q</u>	<u>Count</u>	<u>Qstnr</u>	<u>E/Q</u>
1	162	377	44	152	747
2	179	748	45	202	538
3	245	682	46	133	556
4	193	585	47	433	1145
5	160	533	48	216	714
6	178	407	49	121	713
7	109	514	50	204	644
8	96	437	51	212	626
9	98	360	53	173	704
10	239	622	54	83	439
11	204	779	55	181	1175
12	164	785	56	182	1022
13	315	778	57	139	533
14	214	720	58	334	759
15	262	851	59	128	729
16	431	858	61	198	539
17	218	646	63	173	771
18	194	546	64	209	695
19	265	627	65	150	622
20	99	359	66	190	514
22	305	693	67	251	849
23	57	584	68	131	407
24	190	502	69	119	313
25	18	285	70	191	1112
26	222	866	71	278	636
27	179	495	72	152	525
28	396	1561	73	104	332
29	380	1029	74	272	670
30	378	965	75	148	585
31	336	1630	76	146	424
32	178	422	78	888	2587
33	114	414	79	153	793
34	245	1109	80	179	618
35	197	638	81	169	494
36	525	859			
37	429	950	TOTAL	Qstnr	E/Q
38	164	589	AVG	213.38	699.08
39	177	451			
40	194	472			
41	201	603			
42	156	786			
43	157	483			

Legend

Count represents the chronological number for the participant.

Qstnr represents the time it took participants to complete the questionnaire.

E/Q represents the time it took participants to complete both the exercises and the questionnaire.

TOTAL takes into account all records.

AVG is the average.

APPENDIX G: EXERCISE RESULTS RELIABILITY

FORMULA FOR CALCULATING RELIABILITY

(Taken from the Text *Human Performance Engineering* by Bailey pgs 538 - 542)

t Method with Same-subject Groups

Steps:

1. List the data in the form of a table, with the scores for each subject listed side by side.
2. Obtain the difference (D) between each pair of scores.
3. Square each difference and add these squared values.
4. Count the number of pairs of scores. Call this the value N .
5. Multiply the value obtained in step 3 by the value obtained in step 4.
6. Add together the differences from step 2.
7. Square the answer from step 6.
8. Subtract the value obtained in step 7 from the value obtained in step 5.
9. Take the value from step 4 (N) and square it.
10. Take the value from step 4 (N) and subtract 1.
11. Multiply the value in step 9 by the value in step 10.
12. Divide the value obtained in step 8 by the number obtained in step 11.
13. Use a calculator to obtain the square root of the number obtained in step 12.
14. Obtain the average of group 1 by adding the scores in column 1 and dividing by the value obtained in step 4 (N).
15. Obtain the average of group 2 by adding the scores in column 2 and dividing by the value obtained in step 4 (N).
16. Subtract the value in step 14 from the value in step 15.
17. Divide the value obtained in step 16 by the value obtained in step 13 to obtain the t value.
18. Determine the tabled t by consulting Table 18-3. In the first column, find the number closest to the value calculated in step 10. The tabled t is the number shown to the right of it.
19. Compare the calculated t value with the tabled t value. If the calculated t value is the same as or larger than the tabled t value, the findings are reliable (i.e., statistically significant).

% Correct Answers: Standard Vs. Multi-Column (No Outliers)

Step 1	Step 2	Step 3	Step 4	
% Corr Std	% Corr Mult	Diff. Std-Mult	Total	is # pairs
			11014	76
100	100	0	0	Step 5
100	100	0	0	is Step 3*Step 4
100	75	25	625	837064
100	100	0	0	
100	100	0	0	Step 6
100	100	0	0	Add sub elements
100	100	0	0	of Step 2
100	100	0	0	
100	100	0	0	100
100	100	0	0	
100	100	0	0	Step 7
100	100	0	0	is (Step 6)^2
100	100	0	0	
100	100	0	0	10000
100	100	0	0	
100	100	0	0	Step 8
100	100	0	0	is Step 5-Step 7
100	100	0	0	
100	100	0	0	827064
100	100	0	0	
100	75	25	625	Step 9
100	80	20	400	is (Step 4)^2
100	100	0	0	
100	100	0	0	5776
100	100	0	0	
100	100	0	0	Step 10
100	100	0	0	is (Step 4) -1
75	100	-25	625	
75	100	-25	625	75
100	100	0	0	
66	100	-34	1156	Step 11
100	100	0	0	is (Step 9)*(Step 10)
100	100	0	0	
100	80	20	400	433200
100	75	25	625	
100	75	25	625	Step 12
100	100	0	0	is (Step 8) / (Step 11)
100	100	0	0	
100	100	0	0	1.909196676
100	100	0	0	
100	100	0	0	Step 13
100	66	34	1156	is SQRT(Step 12)
100	100	0	0	
100	100	0	0	1.381736833
100	100	0	0	
100	93	7	49	Step 14
100	100	0	0	is AVG Group 1
100	100	0	0	
100	100	0	0	97.85526316
100	100	0	0	
100	100	0	0	Step 15
100	100	0	0	is AVG Group 2
100	100	0	0	
100	100	0	0	96.53947368
100	100	0	0	
100	100	0	0	Step 16
100	100	0	0	is ABS((Step 15) - (Step 14))
100	100	0	0	
100	100	0	0	1.315789474

100	100	0	0	
100	93	7	49	Step 17 is t Value
100	100	0	0	is (Step 16) / (Step 13)
100	100	0	0	0.952272127
100	100	0	0	
100	92	8	64	Step 18
100	100	0	0	Determine tabled t
100	100	0	0	from Table 18-3
100	100	0	0	use (Step 10) for left #
100	100	0	0	1.99
80	33	47	2209	
100	100	0	0	Step 19
100	100	0	0	if (Step 17) >= (Step 18)
66	100	-34	1156	then reliable
100	100	0	0	else not reliable
100	100	0	0	
100	100	0	0	NOT RELIABLE
100	100	0	0	
100	100	0	0	
100	100	0	0	
100	100	0	0	
100	100	0	0	
75	100	-25	625	
100	100	0	0	
100	100	0	0	
100	100	0	0	
100	100	0	0	
100	100	0	0	
100	100	0	0	

% Incorrect Answers: Standard Vs. Multi-Column (No Outliers)

Step 1	Step 2		Step 3	Step 4	
% Inco Stnd	% Incorr Mult	Difference	Total	Square and add	is # pairs
			221565		76
25	0	25	625	Step 5	
0	0	0	0	is Step 3*Step 4	
0	25	-25	625		16838940
0	0	0	0		
0	0	0	0	Step 6	
0	0	0	0	Add sub elements	
0	0	0	0	of Step 2	
0	0	0	0		-93
0	0	0	0		
0	0	0	0	Step 7	
0	0	0	0	is (Step 6)^2	
0	0	0	0		8649
0	0	0	0		
25	0	25	625	Step 8	
33	0	33	1089	is Step 5-Step 7	
0	0	0	0		16830291
0	0	0	0		
0	25	-25	625	Step 9	
0	20	-20	400	is (Step 4)^2	
0	0	0	0		5776
0	0	0	0		
0	25	-25	625	Step 10	
0	0	0	0	is (Step 4) -1	
25	0	25	625		75
25	0	25	625		
0	0	0	0	Step 11	
33	0	33	1089	is (Step 9)*(Step 10)	
125	0	125	15625		433200
0	20	-20	400		
0	25	-25	625	Step 12	
0	25	-25	625	is (Step 8) / (Step 11)	
0	0	0	0		38.85108726
0	0	0	0		
0	0	0	0	Step 13	
0	33	-33	1089	is SQRT(Step 12)	
0	0	0	0		6.233064034
0	0	0	0		
0	6	-6	36	Step 14	
0	0	0	0	is AVG Group 1	
0	0	0	0		10.96052632
0	0	0	0		
0	0	0	0	Step 15	
33	0	33	1089	is AVG Group 2	
0	160	-160	25600		12.18421053
0	0	0	0		
0	0	0	0	Step 16	
0	0	0	0	is ABS((Step 15) - (Step 14))	
0	0	0	0		1.223684211
0	0	0	0		

0	6	-6	36	Step 17
0	0	0	0	is (Step 16) / (Step 13)
0	0	0	0	0.196321457
0	0	0	0	
0	7	-7	49	Step 18
0	0	0	0	Determine tabled t
0	0	0	0	from Table 18-3
0	0	0	0	use (Step 10) for left #
66	0	66	4356	1.99
20	366	-346	119716	
25	33	-8	64	Step 19
0	0	0	0	if (Step 17) >= (Step 18)
33	25	8	64	then reliable
0	0	0	0	else not reliable
0	0	0	0	
107	75	32	1024	NOT RELIABLE
200	0	200	40000	
0	0	0	0	
33	0	33	1089	
0	0	0	0	
25	0	25	625	
0	0	0	0	
0	0	0	0	
0	50	-50	2500	
0	0	0	0	
0	0	0	0	
0	0	0	0	

Time To Do Exercises: Standard Vs. Multi-Column (No Outliers)

Step 1	Step 2	Step 3	Step 4	
Time (sec)	Time (sec)	Square and add	is # pairs	
Std	Mult	Total		
		2328065		76
101	114	-13	169	Step 5
280	289	-9	81	is Step 3*Step 4
198	239	-41	1681	176932940
190	202	-12	144	
174	199	-25	625	Step 6
120	109	11	121	Add sub elements
236	169	67	4489	of Step 2
188	153	35	1225	951
153	109	44	1936	
183	200	-17	289	Step 7
269	306	-37	1369	is (Step 6)^2
209	412	-203	41209	904401
217	246	-29	841	
240	266	-26	676	Step 8
356	233	123	15129	is Step 5-Step 7
155	272	-117	13689	176028539
172	256	-84	7056	
156	196	-40	1600	Step 9
117	245	-128	16384	is (Step 4)^2
117	143	-26	676	5776
221	167	54	2916	
296	231	65	4225	Step 10
124	188	-64	4096	is (Step 4) -1
121	146	-25	625	75
253	391	-138	19044	
155	161	-6	36	Step 11
928	237	691	477481	is (Step 9)*(Step 10)
398	251	147	21609	433200
171	416	-245	60025	
426	868	-442	195364	Step 12
130	114	16	256	is (Step 8) / (Step 11)
146	154	-8	64	406.3447345
180	684	-504	254016	
315	126	189	35721	Step 13
164	170	-6	36	is SQRT(Step 12)
314	207	107	11449	20.15799431
294	131	163	26569	
155	119	36	1296	Step 14
136	142	-6	36	is AVG Group 1
220	182	38	1444	249.1052632
509	121	388	150544	
170	156	14	196	Step 15
376	219	157	24649	is AVG Group 2
197	139	58	3364	236.5921053
238	185	53	2809	
335	377	-42	1764	Step 16
224	274	-50	2500	is ABS((Step 15) - (Step 14))

382	210	172	29584	12.51315789
215	225	-10	100	
290	124	166	27556	Step 17
238	293	-55	3025	is (Step 16) / (Step 13)
214	142	72	5184	0.620754114
458	536	-78	6084	
276	564	-288	82944	Step 18
218	176	42	1764	Determine tabled t
207	218	-11	121	from Table 18-3
184	417	-233	54289	use (Step 10) for left #
195	146	49	2401	1.99
160	438	-278	77284	
277	209	68	4624	Step 19
201	271	-70	4900	if (Step 17) >= (Step 18)
167	157	10	100	then reliable
462	136	326	106276	else not reliable
131	145	-14	196	
104	90	14	196	NOT RELIABLE
605	316	289	83521	
201	157	44	1936	
193	180	13	169	
87	141	-54	2916	
202	196	6	36	
141	296	-155	24025	
150	128	22	484	
1108	591	517	267289	
496	144	352	123904	
182	257	-75	5625	
161	164	-3	9	

APPENDIX H: THE SURVEY

(Note: QTID = Questionnaire Transferrable Identification. This ID was an extra ID built-in originally. Its functionality or purpose could have been taken care of with a primary key. Since it had been used extensively within the program, it was kept.)

QTID.3

Question: I got confused by moving the columns back and forth when using the Prioritized Left-to-Right Multi-Column sort.

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

QTID.4

Question: I liked sorting on multiple-columns by transposing or shifting an intact column left or right (as found with the Prioritized Left-to-Right Multi-Column Sort).

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

QTID.5

Question: I got confused by having to sort with only one column with the Standard Sort (single column method).

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

QTID.7

Question: Regardless of the sorting method used, I prefer being able to sort on more than one column.

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

QTID.8

Question: Once I had selected the appropriate column selection using the Standard (single column) sort, it was fairly simple to find the exercise answer.

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

QTID.9

Question: Once I had my columns properly arranged with the Prioritized Left-to-Right Multi-Column Sort, it was fairly simple to find the answer to the question.

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

QTID.13

Question: Given the choice, which method of sorting would you like other programs you use to have?

Possible Response 1: Standard Sort (single column)

Possible Response 2: Prioritized Left-to-Right Multi-Column Sort

Possible Response 3: Both

Possible Response 4: Neither

Possible Response 5: Undecided

QTID.16

Question: I have used the Prioritized Left-to-Right Multi-Column sort in other applications?

Possible Response 1: True

Possible Response 2: False

Possible Response 3: Not sure

QTID.17

Question: I have used the Standard (single-column) sort in other applications?

Possible Response 1: True

Possible Response 2: False

Possible Response 3: Not sure

QTID.18

Question: How often do you use computers? (select most correct)

Possible Response 1: Daily

Possible Response 2: Every Other Day

Possible Response 3: Once Per Week

Possible Response 4: Several Times per Month

Possible Response 5: Once per Month

QTID.19

Question: What level of computer expertise do you self-evaluate yourself as having?

Possible Response 1: I know how to turn on a computer, log in, and check my email.

Possible Response 2: I can do the above task as well as use basic features of word processing tools such as Word and other applications.

Possible Response 3: I can do the above tasks as well as use advanced features of word processing tools such as Word and other applications.

Possible Response 4: I feel comfortable with computers and can figure out how to get around most applications and my operating system needs such as install programs and some troubleshooting.

Possible Response 5: I can do the above as well as manipulate computer software with programming and more.

QTID.22

Question: I liked sorting on only one column (as found with the Standard (single-column) sort.

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

QTID.26

Question: My gender:

Possible Response 1: Female

Possible Response 2: Male

QTID.27

Question: My age (yrs):

Possible Response 1: 12-25

Possible Response 2: 26-33

Possible Response 3: 34-41

Possible Response 4: 42-49

Possible Response 5: 50+

QTID.28

Question: I believe that overall I was more accurate in my responses when using the Standard (single-column) sort.

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

QTID.29

Question: I believe that overall I was more accurate in my responses when using the Prioritized Left-to-Right Multi-Column sort.

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

QTID.30

Question: I am enrolled in the following course at BYU:

Possible Response 1: IT 101

Possible Response 2: IT 355

Possible Response 3: IT 391

Possible Response 4: More than one

Possible Response 5: None

QTID.31

Question: The Standard (single-column) sort methodology is more user friendly.

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

QTID.32

Question: The Prioritized Left-to-Right Multi-Column sort methodology is more user friendly.

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

QTID.33

Question: I believe that overall I was quicker at answering the exercise questions when using the Standard (single-column) sort.

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

QTID.34

Question: I believe that overall I was quicker at answering the exercise questions when using the Prioritized Left-to-Right Multi-Column sort.

Possible Response 1: Strongly Disagree

Possible Response 2: Disagree

Possible Response 3: Neutral

Possible Response 4: Agree

Possible Response 5: Strongly Agree

APPENDIX I: THE SURVEY/QUESTIONNAIRE RESULTS

Count represents the chronological number for the participant.

Count #	1	2	3	4	5	6	7	8	9	10	11	12
ID	382	383	384	385	386	387	388	389	390	391	392	393
QTID.3	3	2	1	1	4	1	1	1	3	1	4	3
QTID.4	4	4	4	3	4	5	4	4	5	3	2	5
QTID.5	1	3	1	2	2	3	1	2	1	2	2	2
QTID.7	5	3	4	1	3	5	4	5	4	5	4	4
QTID.8	4	3	5	5	4	3	5	4	5	4	4	3
QTID.9	4	4	5	1	5	5	5	4	5	5	5	4
QTID.13	3	3	3	2	1	2	2	2	2	2	5	2
QTID.16	2	3	1	1	3	2	2	2	2	1	2	3
QTID.17	1	1	1	1	1	1	1	1	2	1	1	1
QTID.18	1	1	1	1	1	1	1	1	2	1	1	2
QTID.19	5	5	4	2	4	5	4	5	1	4	4	3
QTID.22	4	4	4	1	4	2	4	4	3	2	2	4
QTID.26	2	2	2	2	2	2	1	2	2	2	1	1
QTID.27	2	2	3	4	2	2	3	4	3	4	3	1
QTID.28	2	3	2	1	3	3	3	3	2	2	2	3
QTID.29	2	3	4	1	3	3	3	4	5	5	3	4
QTID.30	5	4	5	5	5	5	5	5	5	5	5	5
QTID.31	3	4	4	4	4	2	4	4	4	2	3	4
QTID.32	4	4	2	2	2	4	2	2	2	4	3	3
QTID.33	2	4	4	3	4	3	4	4	4	2	3	3
QTID.34	4	3	5	3	3	4	3	3	2	4	3	4

Count #	13	14	15	16	17	18	19	20	21	22	23	24
ID	394	395	396	397	398	399	400	401	402	403	404	405
QTID.3	2	3	2	1	1	4	2	2	4	2	2	1
QTID.4	3	1	3	5	4	4	4	2	3	4	5	4
QTID.5	3	3	2	1	2	2	3	2	2	2	4	1
QTID.7	3	5	4	5	4	4	4	5	4	4	4	4
QTID.8	1	4	4	5	4	4	2	4	4	4	3	4
QTID.9	3	1	4	5	5	4	4	4	4	4	4	4
QTID.13	5	3	3	3	2	3	2	3	1	2	2	2
QTID.16	2	2	2	3	2	2	2	1	2	2	2	2
QTID.17	1	1	1	1	1	1	2	1	2	2	1	1
QTID.18	1	2	1	1	1	1	1	1	1	3	3	1
QTID.19	5	4	4	2	4	4	4	4	3	3	1	5
QTID.22	3	2	4	4	3	4	3	2	4	3	1	3
QTID.26	2	1	2	1	2	2	2	2	2	1	1	2
QTID.27	1	4	5	5	5	4	1	3	5	3	5	5
QTID.28	3	3	4	3	3	3	2	3	5	3	2	3
QTID.29	3	4	2	3	4	2	4	3	3	3	5	4
QTID.30	5	5	5	5	5	5	5	5	5	5	5	5
QTID.31	3	3	4	5	3	4	3	3	4	2	2	3
QTID.32	3	3	2	2	3	2	3	3	2	3	4	3
QTID.33	3	3	4	3	3	4	3	4	4	2	2	2
QTID.34	3	4	2	3	3	2	4	2	3	3	5	4

Count represents the chronological number for the participant.

Count #	25	26	27	28	29	30	31	32	33	34	35	36
ID	406	407	408	409	410	411	412	413	414	415	416	417
QTID.3	1	2	3	3	3	3	4	4	2	2	3	2
QTID.4	5	4	3	3	3	3	2	4	3	3	4	5
QTID.5	2	2	2	2	2	2	2	1	2	2	3	2
QTID.7	5	4	5	4	4	2	4	5	4	3	4	4
QTID.8	5	4	4	4	4	4	4	3	4	4	4	2
QTID.9	5	4	4	4	4	4	3	5	3	4	4	5
QTID.13	2	3	2	1	1	3	1	3	3	3	3	3
QTID.16	2	2	2	3	2	1	2	3	1	1	3	2
QTID.17	1	1	1	3	3	1	1	1	1	1	1	2
QTID.18	1	1	1	2	2	1	1	1	1	1	1	1
QTID.19	5	4	3	1	3	4	4	5	4	4	4	5
QTID.22	4	4	2	4	4	1	4	3	4	3	2	1
QTID.26	1	2	1	2	1	2	1	2	1	1	2	1
QTID.27	1	4	4	5	4	3	4	1	1	1	1	4
QTID.28	2	3	3	4	2	4	4	2	4	3	4	2
QTID.29	4	4	3	4	2	3	2	4	3	2	4	5
QTID.30	2	5	5	5	5	5	5	3	5	5	5	5
QTID.31	2	3	3	4	4	4	4	4	3	3	3	2
QTID.32	5	3	3	3	2	3	2	2	3	3	3	3
QTID.33	2	3	2	4	3	3	4	2	3	3	4	2
QTID.34	4	3	4	4	2	3	2	4	3	3	4	4

Count #	37	38	39	40	41	42	43	44	45	46	47	48
ID	418	419	420	421	422	423	424	425	426	427	428	429
QTID.3	5	4	3	2	2	1	1	2	2	3	3	4
QTID.4	2	4	4	5	1	4	4	4	4	3	2	3
QTID.5	1	3	2	3	1	3	2	2	4	2	2	2
QTID.7	4	4	4	5	4	5	1	4	5	4	3	3
QTID.8	4	5	4	4	5	3	4	4	2	4	4	5
QTID.9	4	1	5	5	5	4	1	5	5	4	4	3
QTID.13	1	2	3	2	1	2	2	5	2	1	1	1
QTID.16	2	2	2	2	1	2	1	2	2	2	2	3
QTID.17	1	3	2	1	1	1	1	1	1	1	1	1
QTID.18	1	2	1	1	1	1	1	1	1	1	1	1
QTID.19	3	4	4	4	5	3	5	3	4	4	4	3
QTID.22	5	3	2	1	2	2	2	2	1	4	4	1
QTID.26	2	2	2	2	2	1	2	1	2	2	2	2
QTID.27	2	2	1	3	2	2	4	3	2	4	1	1
QTID.28	4	2	4	1	2	2	2	3	1	4	4	4
QTID.29	2	4	3	5	2	4	5	3	5	3	2	2
QTID.30	5	5	5	5	5	5	5	5	5	5	5	5
QTID.31	5	4	4	1	4	3	2	3	2	4	4	4
QTID.32	1	3	2	5	2	3	4	3	4	3	2	2
QTID.33	5	1	4	1	4	2	2	3	2	4	4	4
QTID.34	2	5	4	4	2	4	4	3	4	2	2	2

Count represents the chronological number for the participant.

Count #	49	50	51	52	53	54	55	56	57	58	59	60
ID	430	431	432	433	434	435	436	437	438	439	440	441
QTID.3	3	3	2	3	2	3	3	3	2	2	1	4
QTID.4	4	4	4	3	5	1	4	3	2	5	5	2
QTID.5	2	2	3	2	3	2	2	2	3	1	1	2
QTID.7	5	5	4	4	4	5	5	4	3	5	5	3
QTID.8	5	2	2	4	3	5	4	4	4	4	4	5
QTID.9	5	5	4	4	5	1	3	3	3	5	4	4
QTID.13	3	2	2	5	3	3	1	2	4	2	2	1
QTID.16	3	2	2	2	2	1	1	3	2	1	1	2
QTID.17	3	1	1	1	1	1	1	3	1	1	1	1
QTID.18	2	1	1	1	1	1	5	1	1	1	1	1
QTID.19	3	3	4	1	4	4	2	4	4	5	3	4
QTID.22	3	3	3	3	2	1	5	3	3	3	5	4
QTID.26	1	2	1	1	2	2	1	2	1	2	2	2
QTID.27	2	1	1	1	3	1	2	2	3	2	1	1
QTID.28	3	2	2	3	2	3	3	4	2	1	3	4
QTID.29	5	4	4	3	4	1	3	4	3	5	5	4
QTID.30	5	5	5	5	5	5	5	5	5	5	5	5
QTID.31	4	4	2	3	2	3	4	2	3	4	3	5
QTID.32	3	3	4	3	4	5	3	4	3	2	3	2
QTID.33	4	2	2	3	2	3	4	3	3	2	3	4
QTID.34	4	4	4	3	4	2	3	3	4	4	3	4

Count #	61	62	63	64	65	66	67	68	69	70	71	72
ID	442	443	444	445	446	447	448	449	450	451	452	453
QTID.3	2	3	3	3	2	1	4	2	2	3	1	4
QTID.4	4	3	2	4	4	5	2	4	4	2	4	3
QTID.5	2	3	3	4	4	1	2	2	2	1	3	1
QTID.7	4	3	3	4	4	5	3	5	3	4	4	4
QTID.8	4	3	4	4	4	4	4	4	4	5	4	5
QTID.9	5	3	4	3	5	5	3	4	3	3	4	4
QTID.13	2	3	3	1	2	2	1	3	3	1	3	3
QTID.16	2	2	3	2	2	1	2	3	3	3	2	2
QTID.17	1	1	1	1	1	1	1	1	3	1	1	1
QTID.18	1	1	1	1	1	1	1	1	1	1	1	1
QTID.19	4	1	4	4	3	4	4	4	5	4	4	4
QTID.22	4	1	4	4	3	3	4	4	3	5	3	4
QTID.26	1	1	2	2	2	2	1	2	2	2	1	2
QTID.27	1	2	1	1	2	3	1	1	1	2	3	3
QTID.28	3	3	4	4	2	2	5	3	3	4	3	2
QTID.29	3	2	3	2	5	4	2	4	3	2	3	3
QTID.30	5	2	5	5	5	5	5	5	5	5	5	5
QTID.31	4	1	4	5	2	1	5	2	3	4	4	2
QTID.32	2	5	3	2	4	5	2	4	3	2	2	2
QTID.33	3	1	4	4	2	1	4	4	3	4	3	3
QTID.34	4	3	3	2	4	4	2	2	3	2	3	3

Count represents the chronological number for the participant.

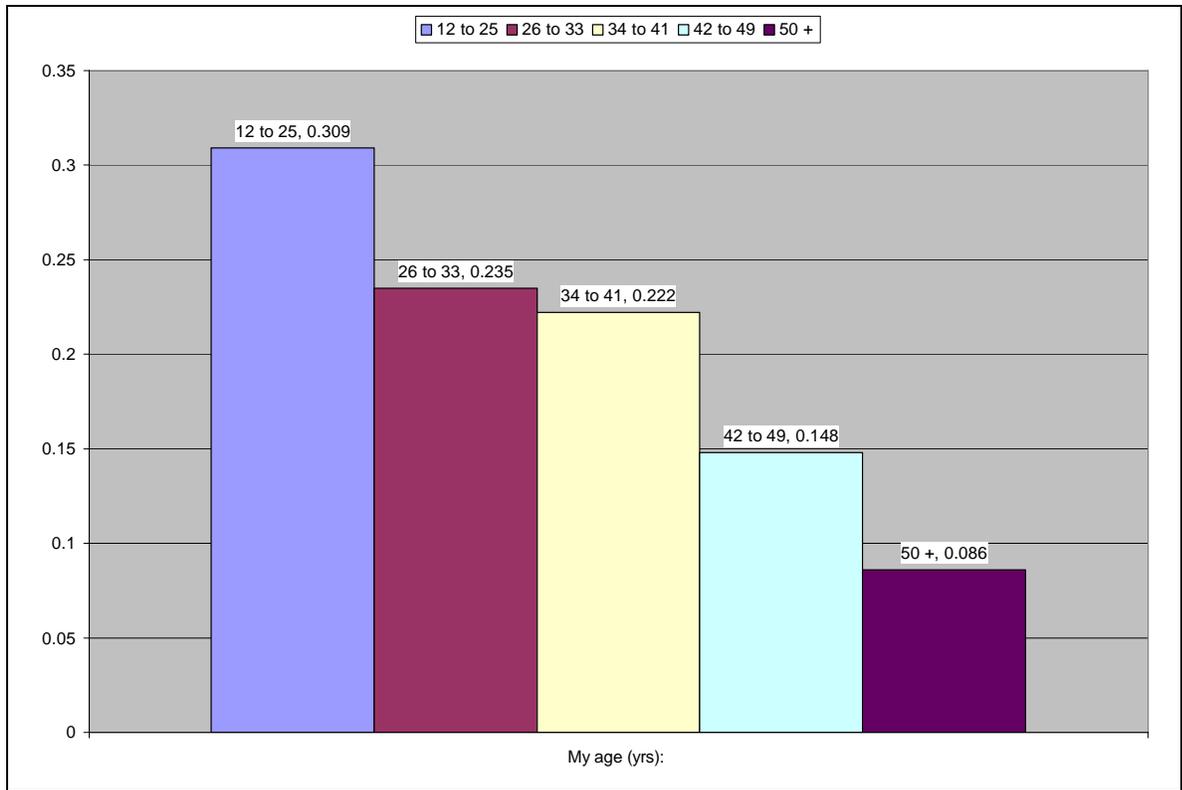
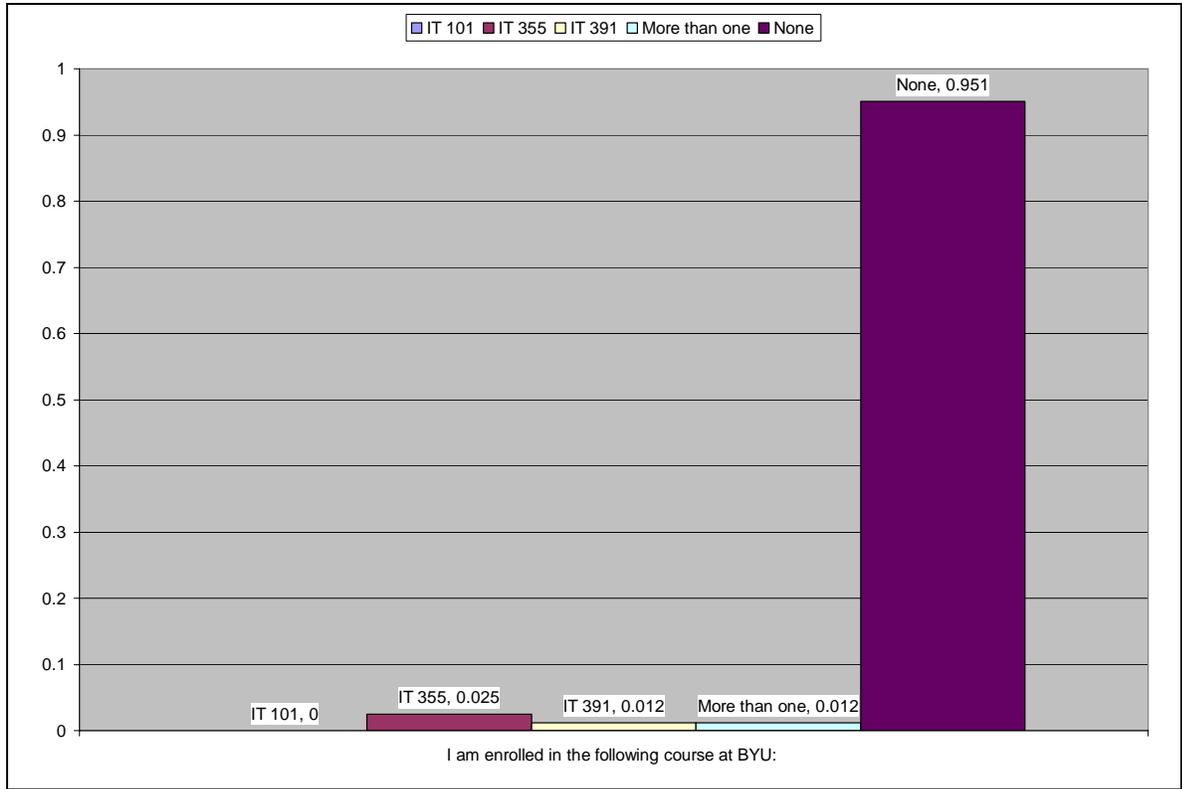
Count #	73	74	75	76	77	78	79	80	81
ID	454	455	456	457	458	459	460	461	462
QTID.3	2	3	1	2	2	2	2	4	2
QTID.4	4	3	4	4	4	4	4	5	4
QTID.5	4	2	1	3	3	2	2	2	4
QTID.7	5	4	4	4	4	5	5	5	5
QTID.8	4	4	3	4	4	4	1	1	4
QTID.9	5	4	4	4	4	4	1	1	5
QTID.13	3	2	2	2	2	3	2	2	2
QTID.16	1	2	2	1	2	2	2	1	3
QTID.17	1	2	1	1	1	1	1	1	1
QTID.18	1	1	1	1	1	1	1	1	1
QTID.19	5	3	4	3	2	4	2	4	5
QTID.22	2	2	4	3	4	3	3	1	2
QTID.26	2	2	2	1	1	2	1	1	2
QTID.27	1	3	2	2	1	2	3	3	3
QTID.28	1	2	3	2	3	4	4	3	3
QTID.29	5	4	4	4	3	4	4	1	3
QTID.30	5	5	5	5	5	5	5	5	5
QTID.31	3	2	2	2	2	2	4	2	2
QTID.32	3	4	4	4	3	4	3	4	4
QTID.33	2	2	2	2	4	3	4	3	2
QTID.34	4	4	4	4	4	4	3	4	4

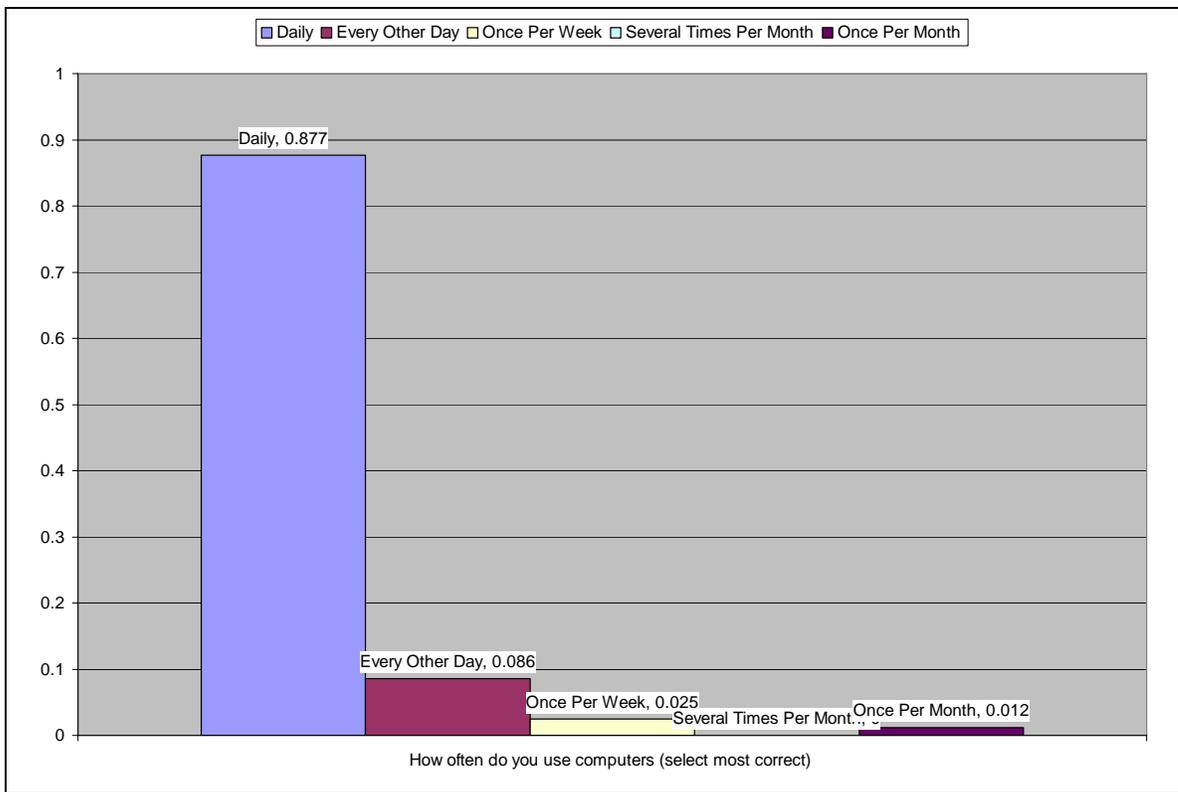
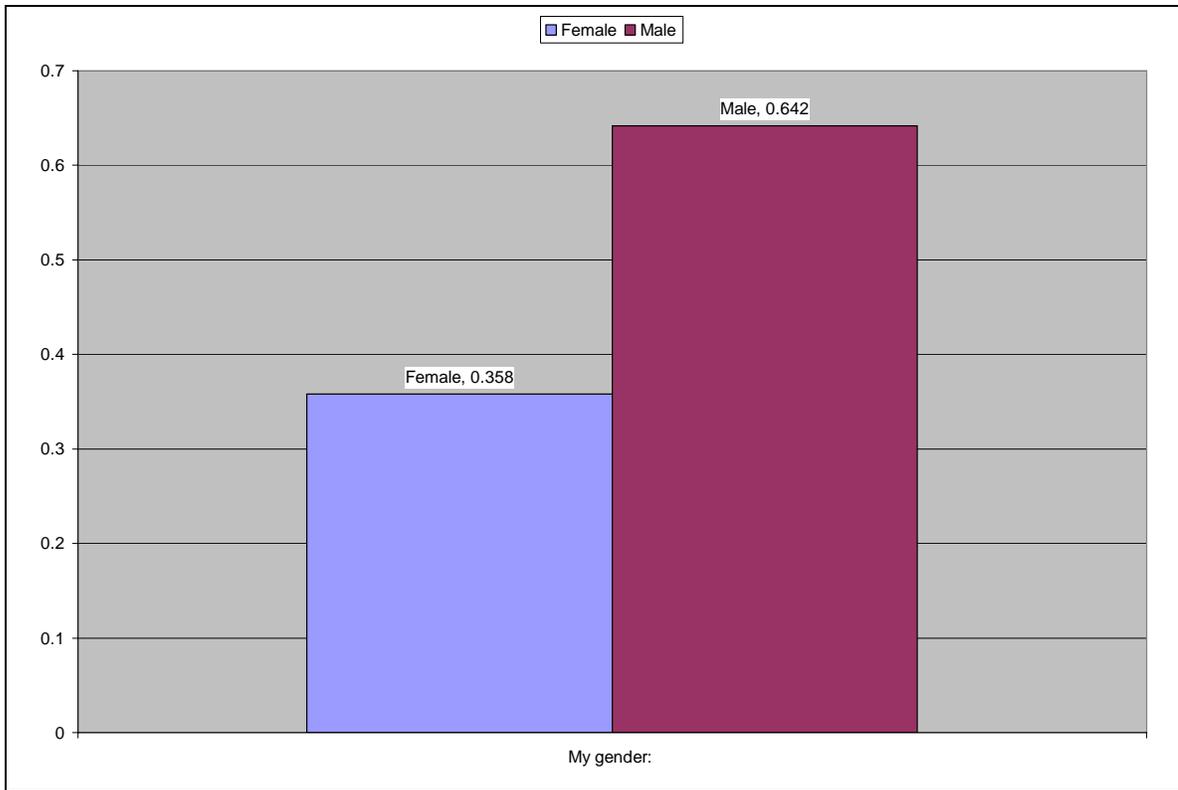
APPENDIX J: THE QUESTIONNAIRE RESULTS - SUMMARY

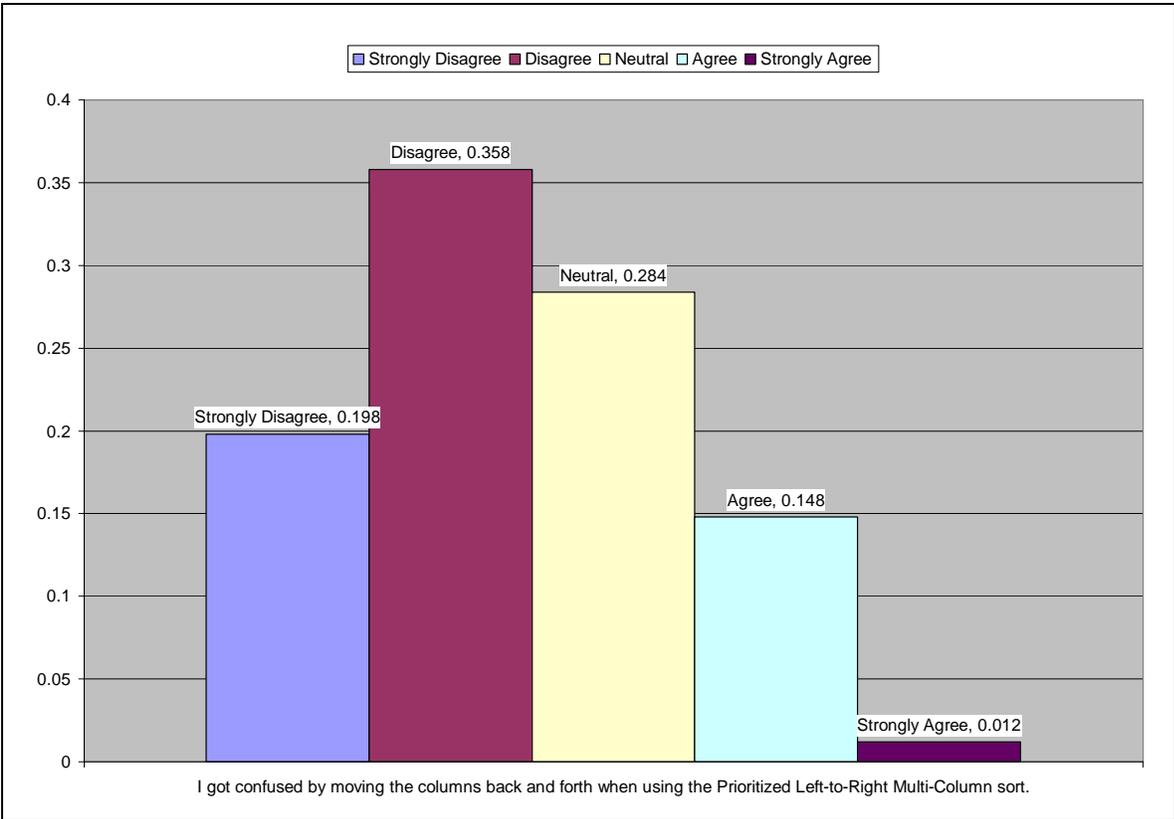
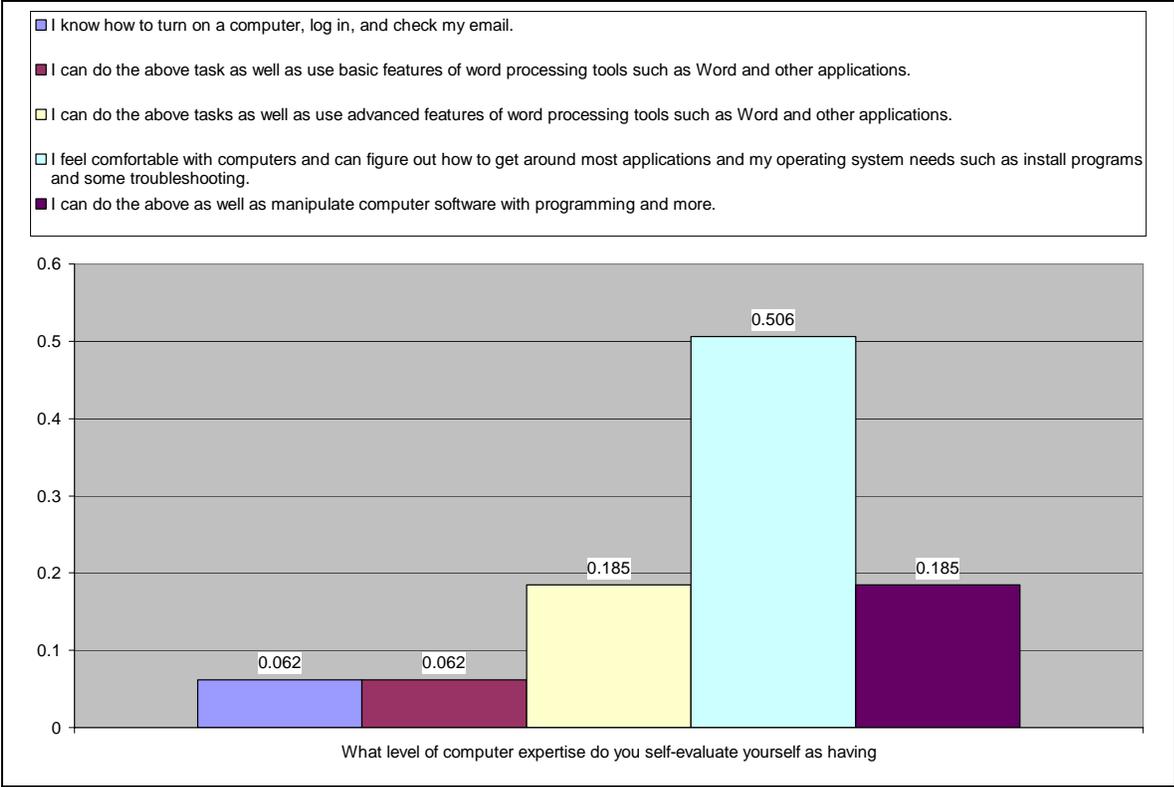
ID	Answer 1	Answer 2	Answer 3	Answer 4	Answer 5
QTID.3	16	29	23	12	1
QTID.4	3	10	18	37	13
QTID.5	15	43	17	6	0
QTID.7	2	1	12	40	26
QTID.8	3	5	9	50	14
QTID.9	7	0	12	36	26
QTID.13	15	34	27	1	4
QTID.16	17	49	15		
QTID.17	68	7	6		
QTID.18	71	7	2	0	1
QTID.19	5	5	15	41	15
QTID.22	10	16	23	28	4
QTID.26	29	52			
QTID.27	25	19	18	12	7
QTID.28	5	24	32	18	2
QTID.29	3	14	26	26	12
QTID.30	0	2	1	1	77
QTID.31	3	21	21	31	5
QTID.32	1	25	32	18	5
QTID.33	4	22	26	28	1
QTID.34	0	16	26	36	3

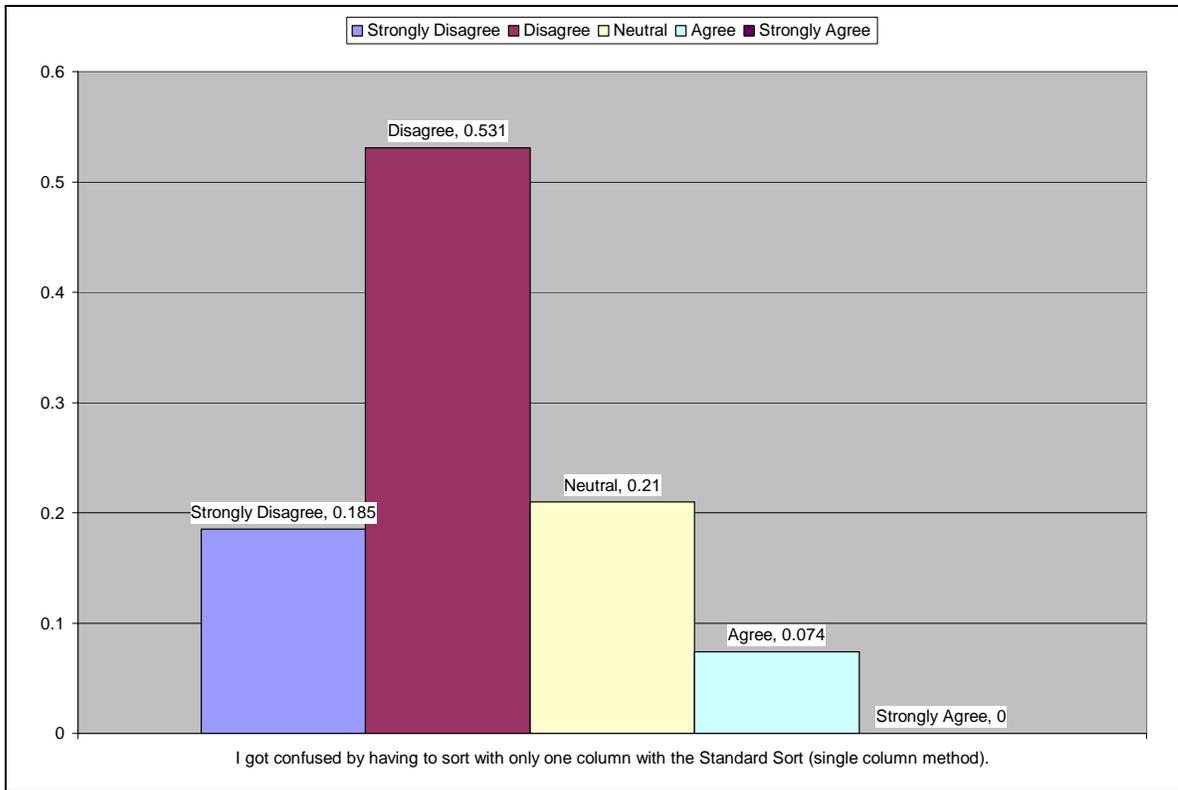
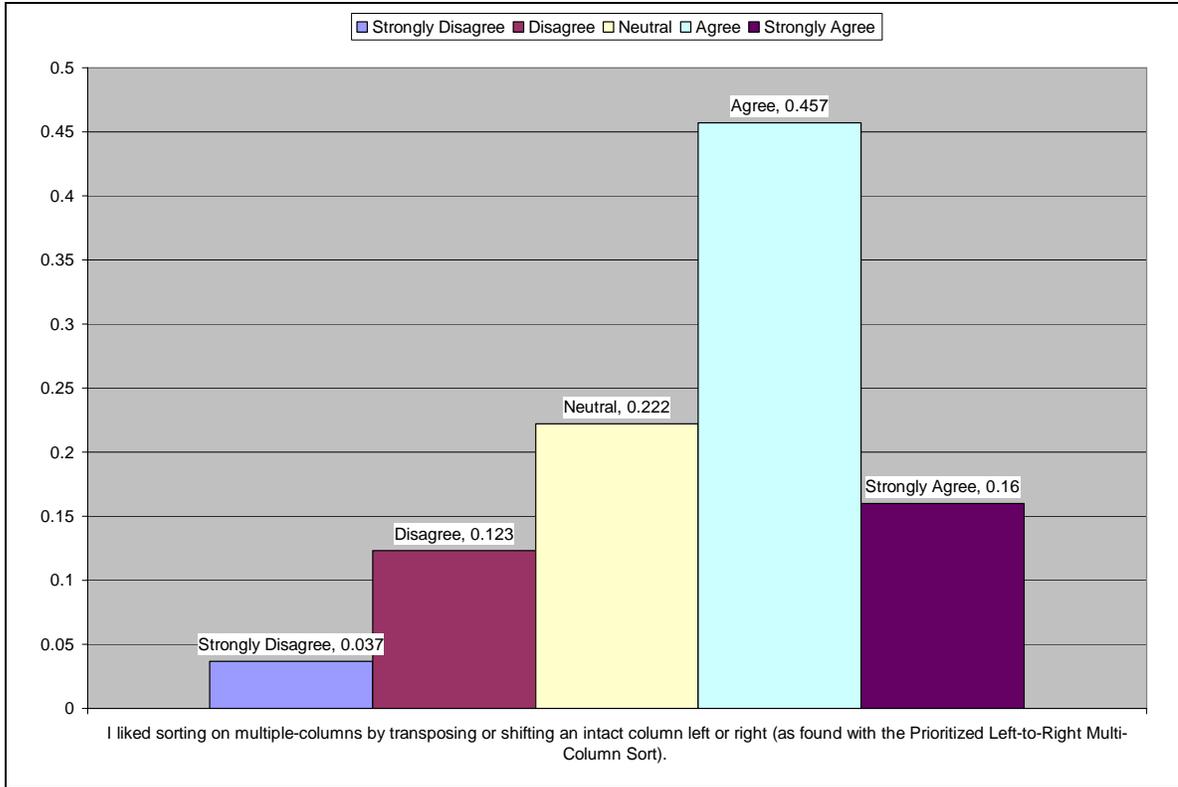
ID	Answer 1	Answer 2	Answer 3	Answer 4	Answer 5
QTID.3	19.80%	35.80%	28.40%	14.80%	1.20%
QTID.4	3.70%	12.30%	22.20%	45.70%	16.00%
QTID.5	18.50%	53.10%	21.00%	7.40%	0.00%
QTID.7	2.50%	1.20%	14.80%	49.40%	32.10%
QTID.8	3.70%	6.20%	11.10%	61.70%	17.30%
QTID.9	8.60%	0.00%	14.80%	44.40%	32.10%
QTID.13	18.50%	42.00%	33.30%	1.20%	4.90%
QTID.16	21.00%	60.50%	18.50%		
QTID.17	84.00%	8.60%	7.40%		
QTID.18	87.70%	8.60%	2.50%	0.00%	1.20%
QTID.19	6.20%	6.20%	18.50%	50.60%	18.50%
QTID.22	12.30%	19.80%	28.40%	34.60%	4.90%
QTID.26	35.80%	64.20%			
QTID.27	30.90%	23.50%	22.20%	14.80%	8.60%
QTID.28	6.20%	29.60%	39.50%	22.20%	2.50%
QTID.29	3.70%	17.30%	32.10%	32.10%	14.80%
QTID.30	0.00%	2.50%	1.20%	1.20%	95.10%
QTID.31	3.70%	25.90%	25.90%	38.30%	6.20%
QTID.32	1.20%	30.90%	39.50%	22.20%	6.20%
QTID.33	4.90%	27.20%	32.10%	34.60%	1.20%
QTID.34	0.00%	19.80%	32.10%	44.40%	3.70%

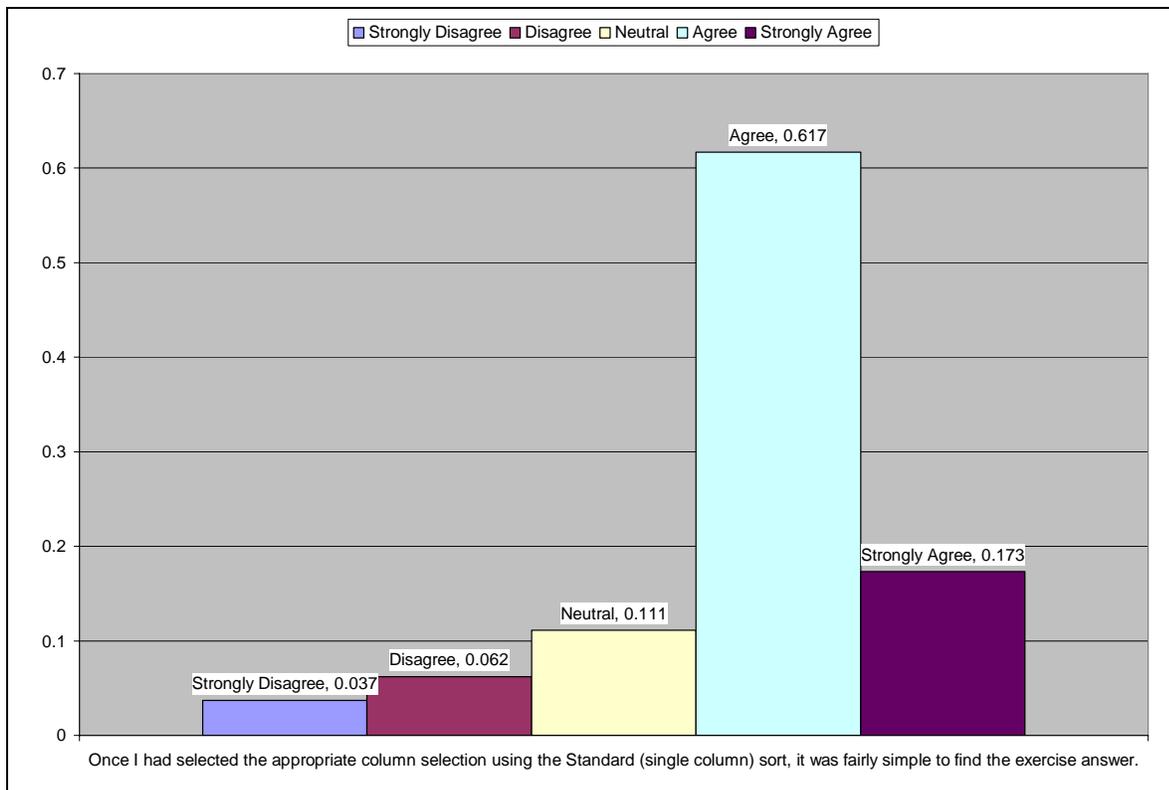
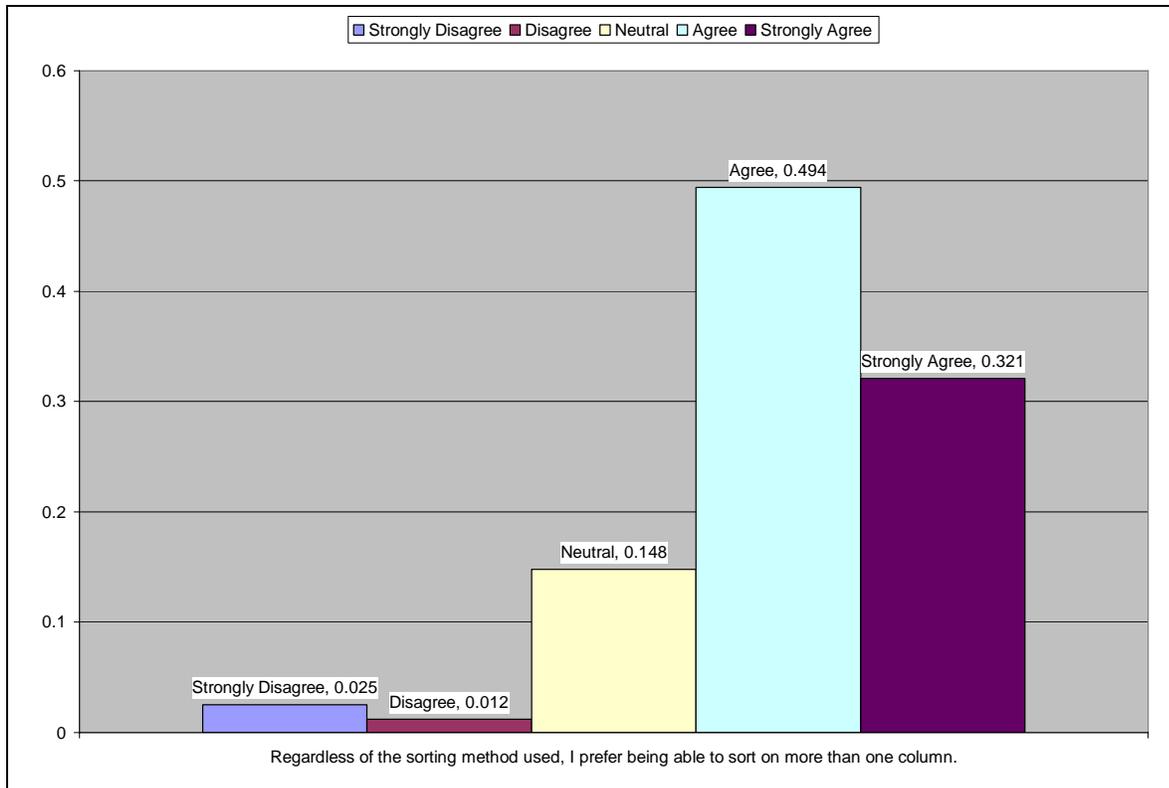
APPENDIX K: GRAPHS OF QUESTIONNAIRE SUMMARY RESULTS

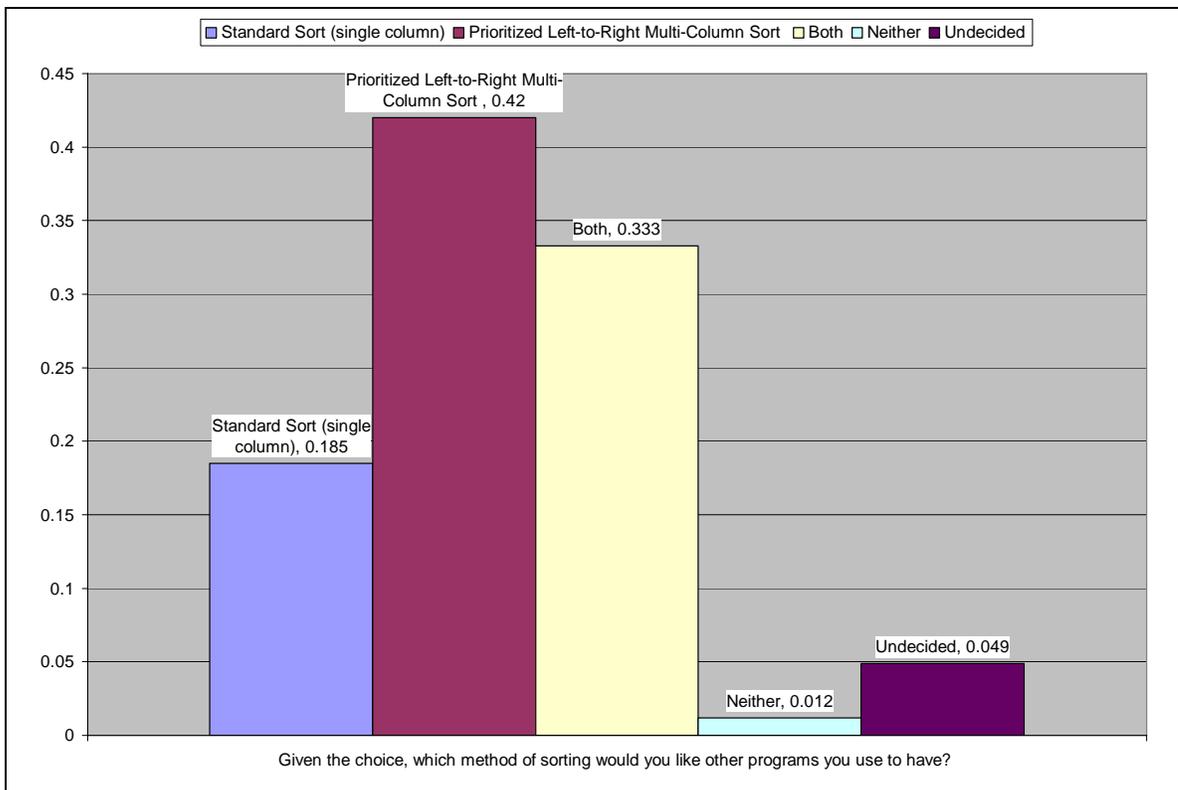
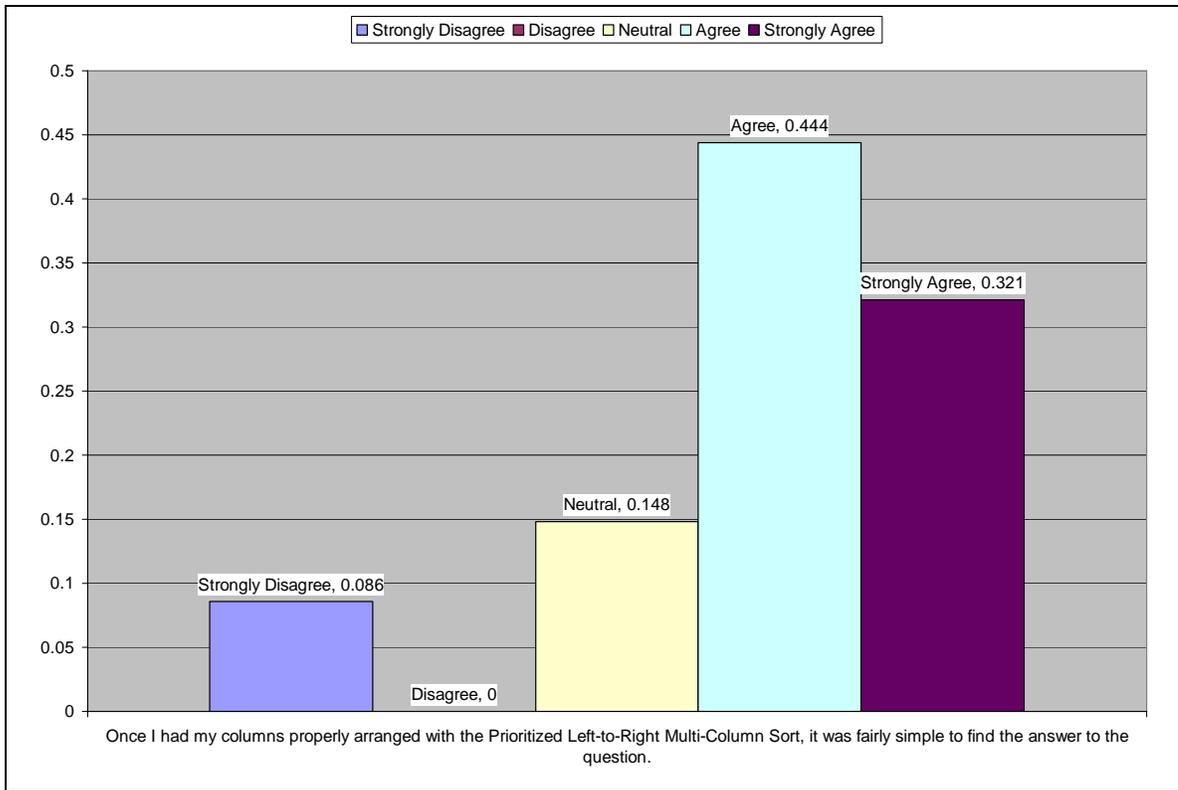


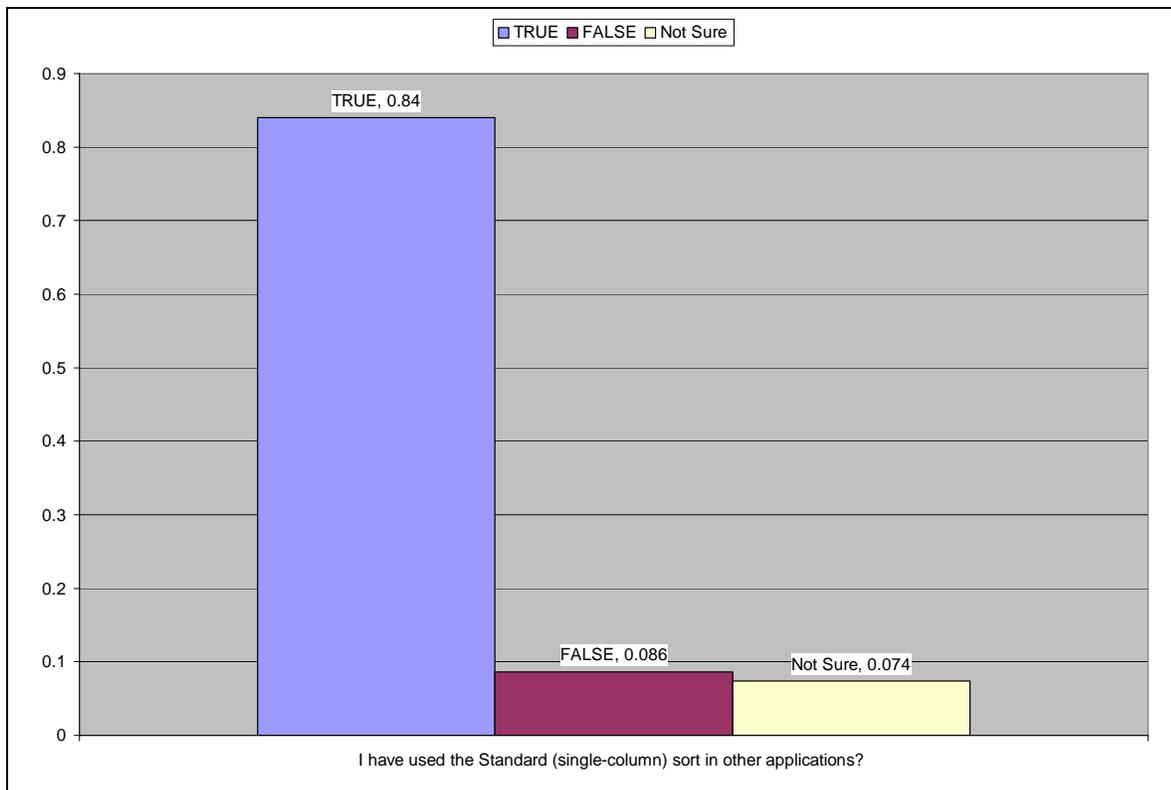
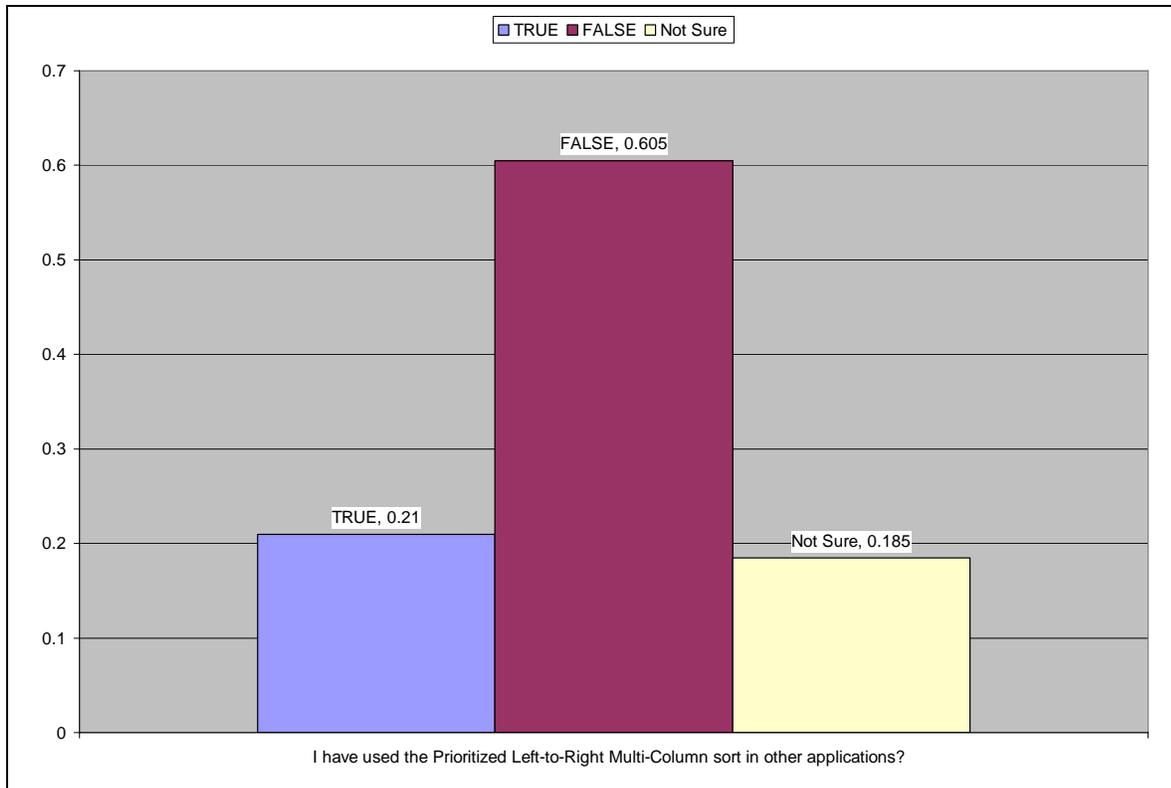


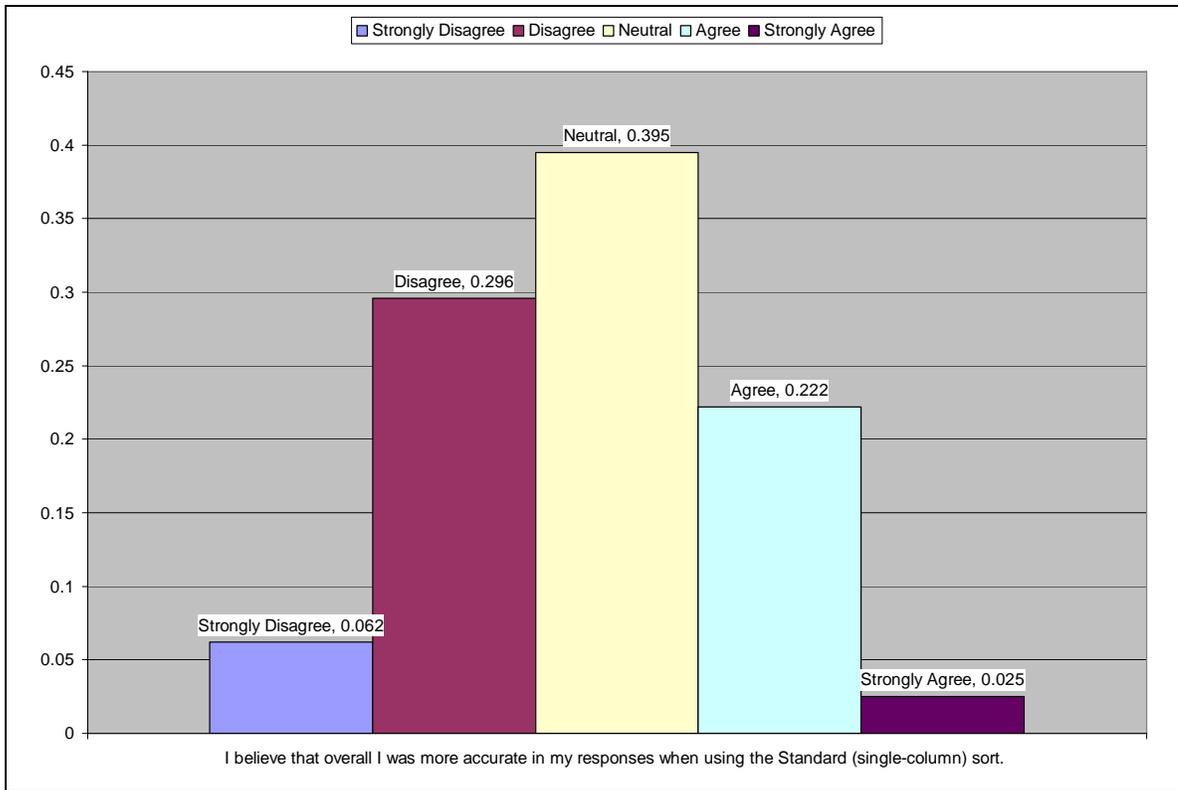
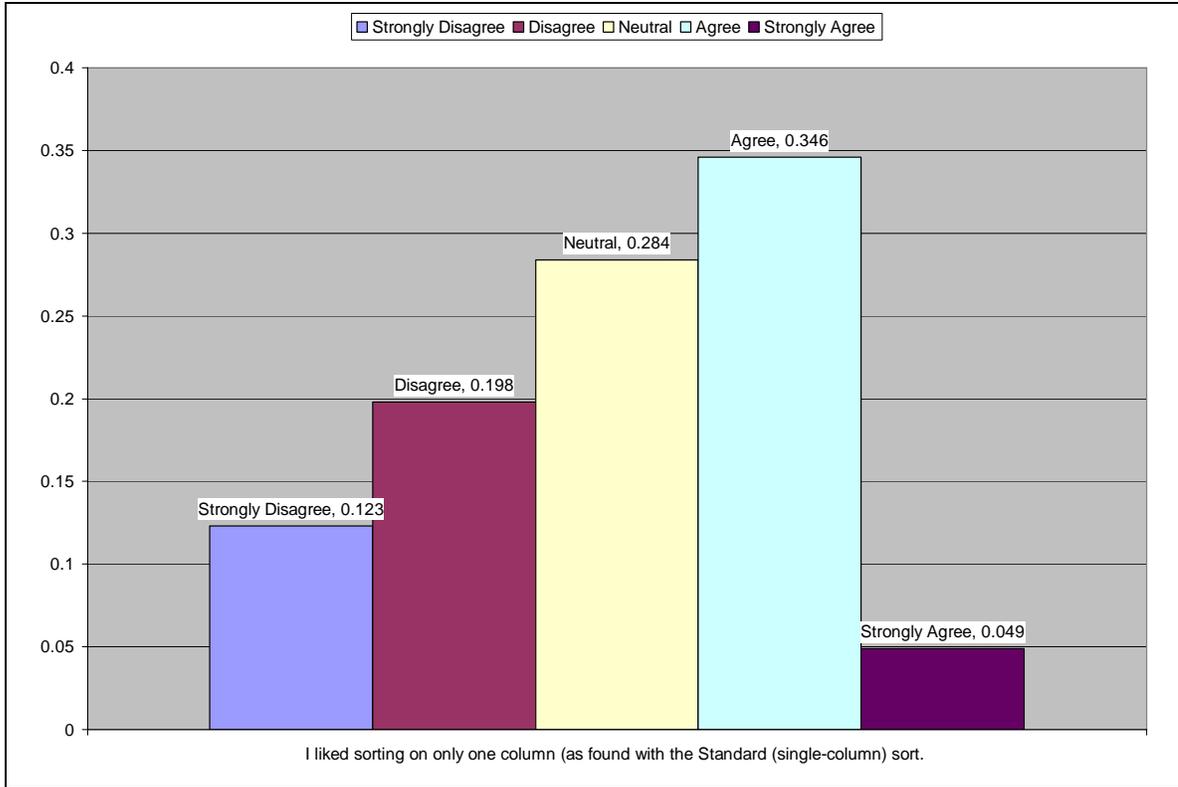


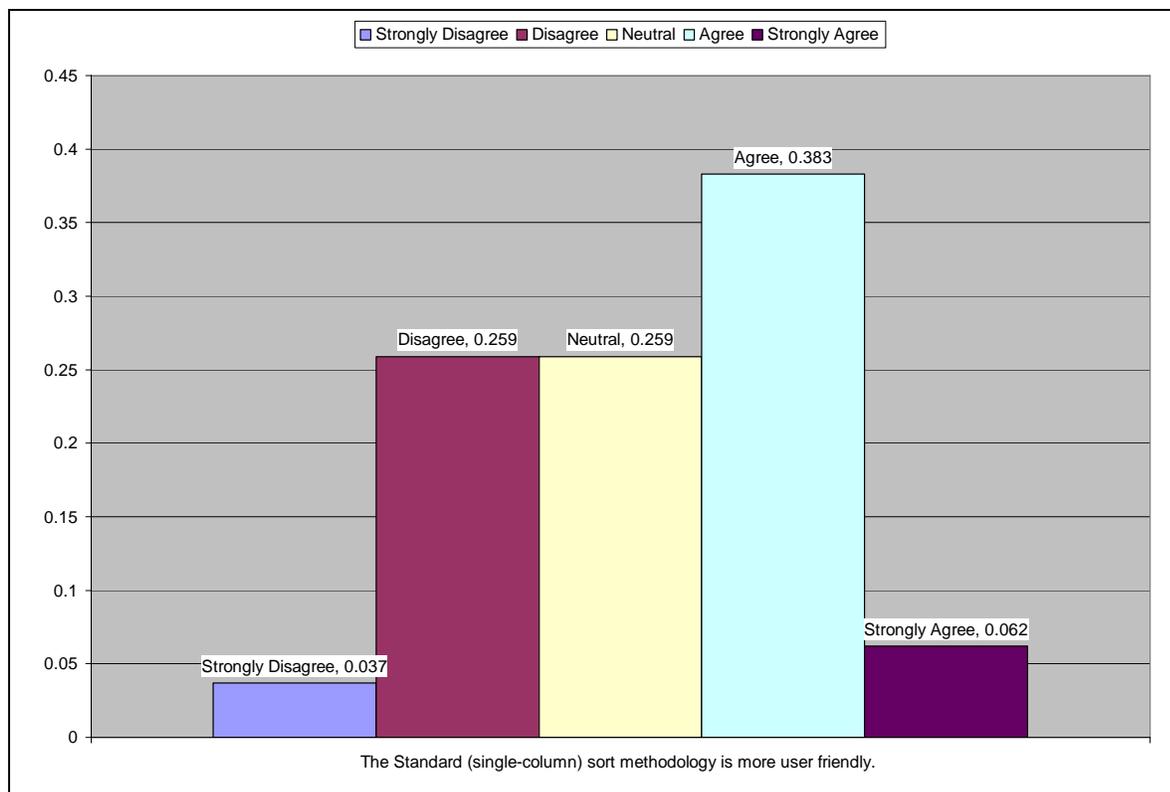
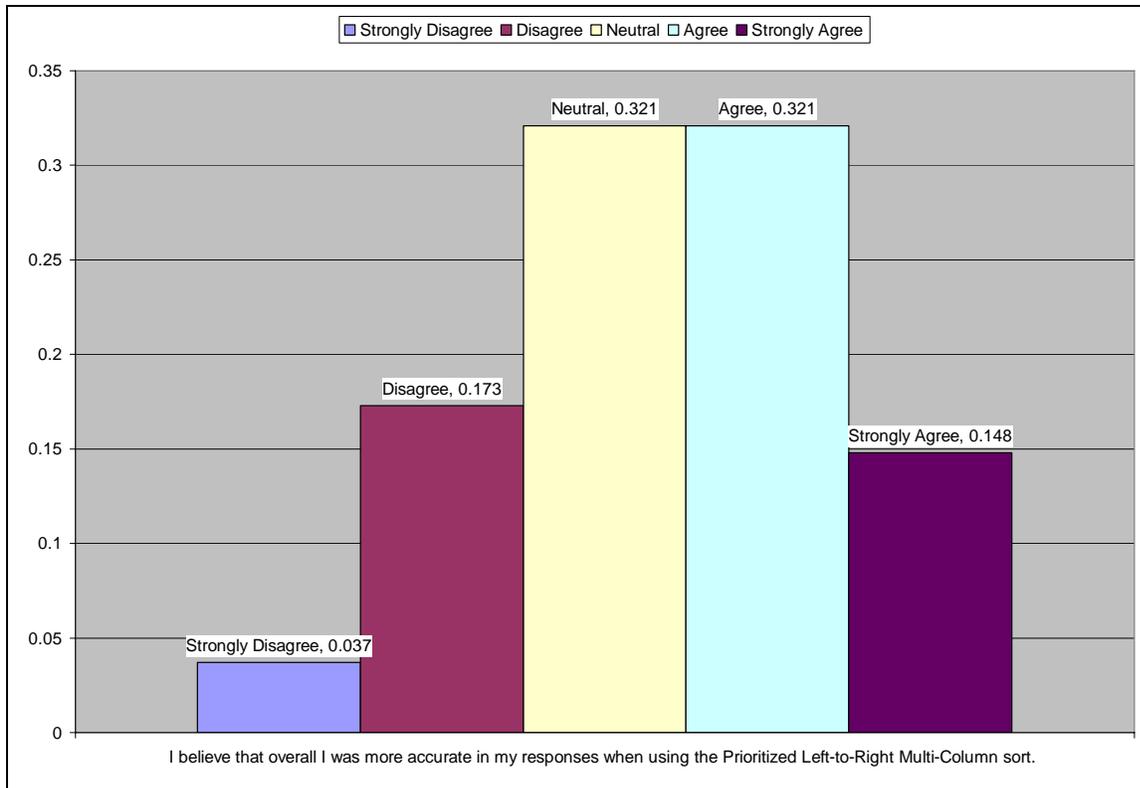


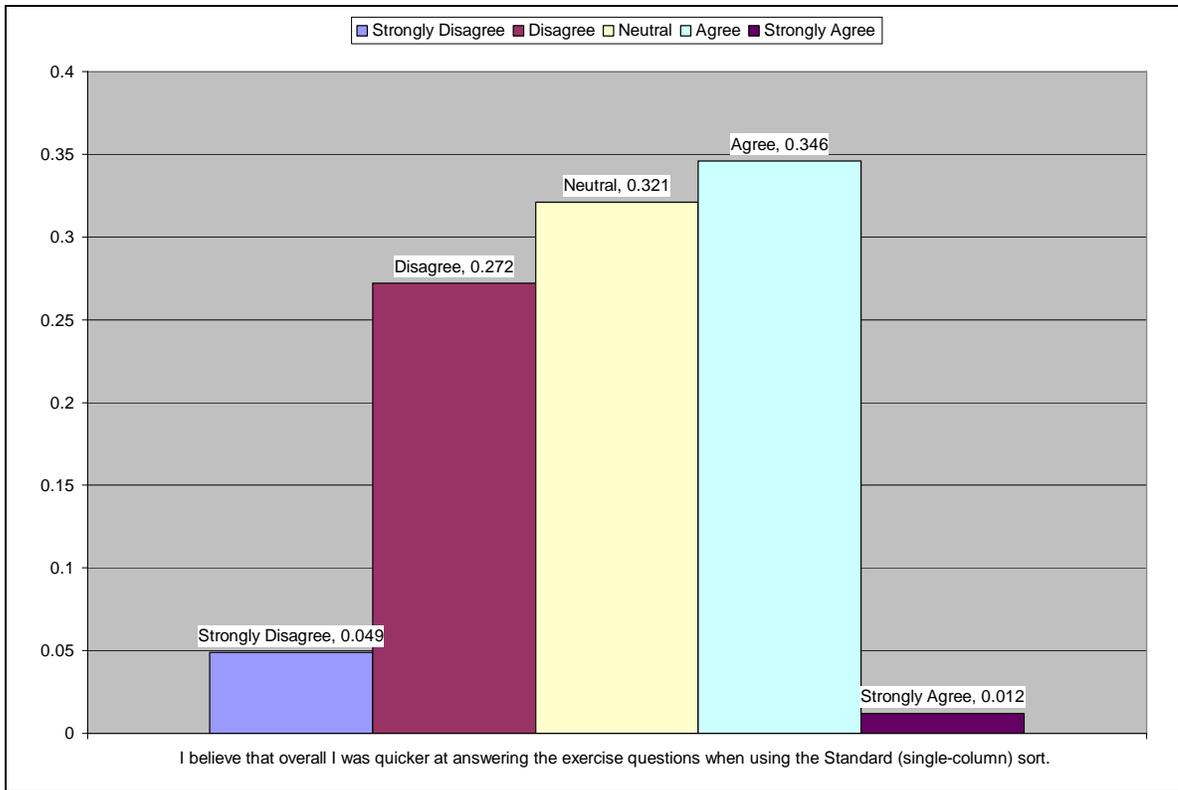
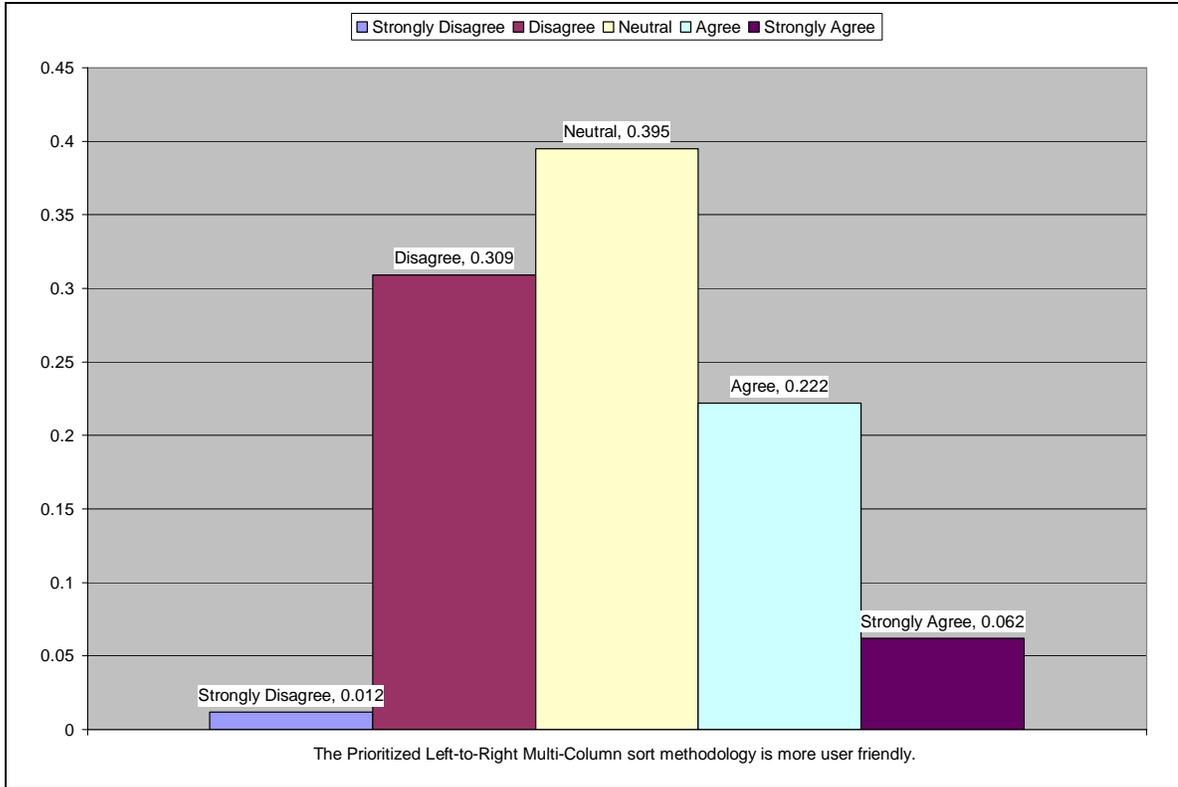


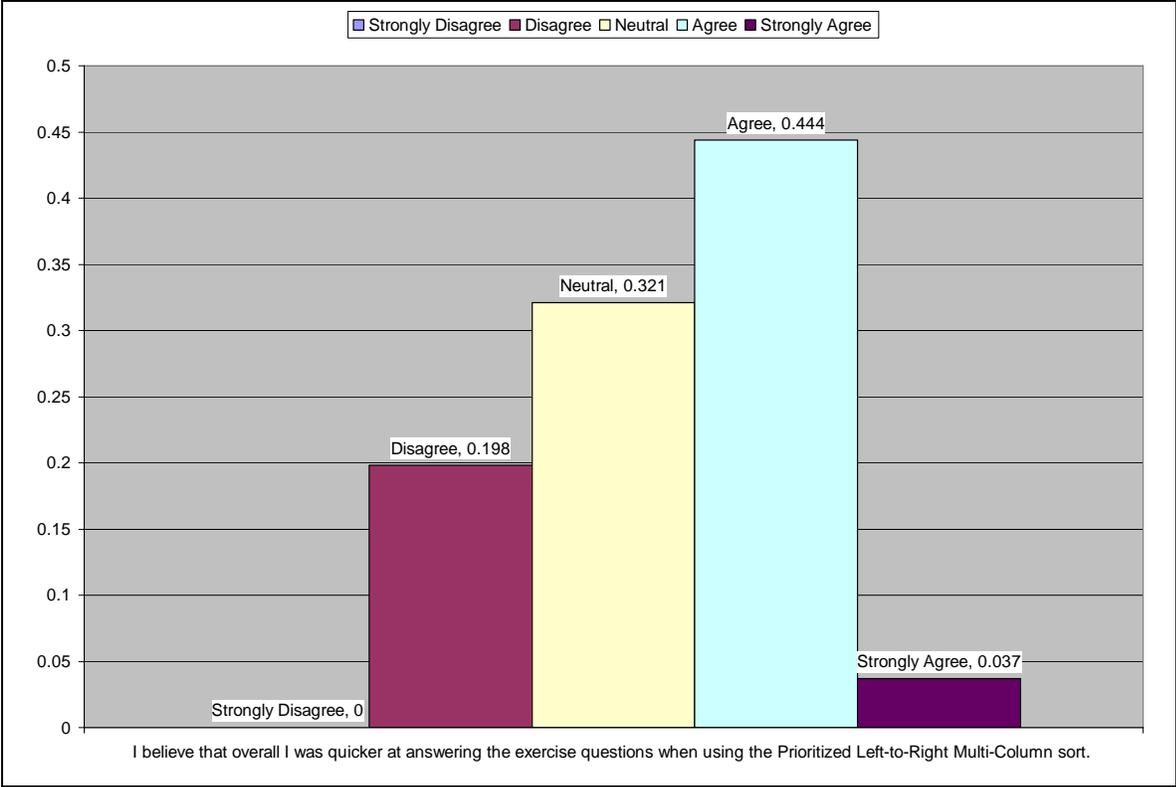












APPENDIX L: QUESTIONNAIRE RESPONSE DATA RELIABILITY

FORMULA FOR CALCULATING RELIABILITY

Pages 550-551 Human Performance Engineering

Step 1: List the frequencies (number of people preferring each alternative) in one row. These are the *observed* frequencies.

Step 2: Determine the *expected* frequencies by dividing the total of observed frequencies by the number of observations. This suggests that there is an expectation that about the same number of people prefer each of the alternatives.

Step 3: For each column, subtract the expected frequency from the observed frequency.

Step 4: Square the values from step 3.

Step 5: Divide the quantities in step 4 by the expected frequencies.

Step 6: Chi square is determined by adding the values in step 5.

Step 7: Count the number of columns and subtract 1.

Step 8: Determine the tabled chi square by consulting Table 18-5 (page 552). In the first column, find the value from step 7. The tabled chi square is the number shown to the right of it.

Step 9: Compare the calculated chi square with the tabled chi square. If the calculated chi square is the same as or larger than the tabled chi square, then the findings are reliable.

Data Reliability (CHI-SQUARE) (1st page of 3)

ID	QTID.3	QTID.4	QTID.5	QTID.7	QTID.8	QTID.9
Step 1. List frequencies.						
Answered 1&2	45	13	58	3	8	7
Answered 3	23	18	17	12	9	12
Answered 4&5	13	50	6	66	64	62
Step 2. Expected freq.	27	27	27	27	27	27
Step 3.						
Observed - Expected						
Answered 1&2	18	-14	31	-24	-19	-20
Answered 3	-4	-9	-10	-15	-18	-15
Answered 4&5	-14	23	-21	39	37	35
Step 4: (Step 3)squared						
Answered 1&2	324	196	961	576	361	400
Answered 3	16	81	100	225	324	225
Answered 4&5	196	529	441	1521	1369	1225
Step 5:						
(Step 4) / Step 2						
Answered 1&2	12.00	7.26	35.59	21.33	13.37	14.81
Answered 3	0.59	3.00	3.70	8.33	12.00	8.33
Answered 4&5	7.26	19.59	16.33	56.33	50.70	45.37
Step 6: CHI square						
(Stp 5)[1&2]+[3]+[4&5]	19.85	29.85	55.63	86.00	76.07	68.52
Step 7: Columns -1	2	2	2	2	2	2
Step 8: Tabled Chi Sqr	5.99	5.99	5.99	5.99	5.99	5.99
Table 18-5						
Step 9: Reliability	Reliable	Reliable	Reliable	Reliable	Reliable	Reliable
ID	QTID.3	QTID.4	QTID.5	QTID.7	QTID.8	QTID.9

QTID.3: I got confused by moving the columns back and forth when using the Prioritized Left-to-Right Multi-Column sort.

QTID.4: I liked sorting on multiple-columns by transposing or shifting an intact column left or right (as found with the Prioritized Left-to-Right Multi-Column Sort).

QTID.5: I got confused by having to sort with only one column with the Standard Sort (single column method).

QTID.7: Regardless of the sorting method used, I prefer being able to sort on more than one column.

QTID.8: Once I had selected the appropriate column selection using the Standard (single column) sort, it was fairly simple to find the exercise answer.

QTID.9: Once I had my columns properly arranged with the Prioritized Left-to-Right Multi-Column Sort, it was fairly simple to find the answer to the question.

Data Reliability (CHI-SQUARE) (2nd page of 3)

ID	QTID.16	QTID.17	QTID.22	QTID.28	QTID.29	QTID.31
Step 1. List frequencies.						
Answered 1&2	17	68	26	29	17	24
Answered 3	49	7	23	32	26	21
Answered 4&5	15	6	32	20	38	36
Step 2. Expected freq.	27	27	27	27	27	27
Step 3.						
Observed - Expected						
Answered 1&2	-10	41	-1	2	-10	-3
Answered 3	22	-20	-4	5	-1	-6
Answered 4&5	-12	-21	5	-7	11	9
Step 4: (Step 3)squared						
Answered 1&2	100	1681	1	4	100	9
Answered 3	484	400	16	25	1	36
Answered 4&5	144	441	25	49	121	81
Step 5:						
(Step 4) / Step 2						
Answered 1&2	3.70	62.26	0.04	0.15	3.70	0.33
Answered 3	17.93	14.81	0.59	0.93	0.04	1.33
Answered 4&5	5.33	16.33	0.93	1.81	4.48	3.00
Step 6: CHI square						
(Stp 5)[1&2]+[3]+[4&5]	26.96	93.41	1.56	2.89	8.22	4.67
Step 7: Columns -1	2	2	2	2	2	2
Step 8: Tabled Chi Sqr	5.99	5.99	5.99	5.99	5.99	5.99
Table 18-5						
Step 9: Reliability	Reliable	Reliable	Not Reliable	Not Reliable	Reliable	Not Reliable
ID	QTID.16	QTID.17	QTID.22	QTID.28	QTID.29	QTID.31

QTID.16: I have used the Prioritized Left-To-Right Multi-Column sort in other applications?

QTID.17: I have used the Standard (single-column) sort in other applications?

QTID.22: I liked sorting on only one column (as found with the Standard (single-column) sort.

QTID.28: I believe that overall I was more accurate in my responses when using the Standard (single-column) sort.

QTID.29: I believe that overall I was more accurate in my responses when using the Prioritized Left-to-Right Multi-Column sort.

QTID.31: The Standard (single-column) sort methodology is more user friendly.

Data Reliability (CHI-SQUARE) (3rd page of 3)

ID	QTID.32	QTID.33	QTID.34
Step 1. List frequencies.			
Answered 1&2	26	26	16
Answered 3	32	26	26
Answered 4&5	23	29	39
Step 2. Expected freq.	27	27	27
Step 3.			
Observed - Expected			
Answered 1&2	-1	-1	-11
Answered 3	5	-1	-1
Answered 4&5	-4	2	12
Step 4: (Step 3)squared			
Answered 1&2	1	1	121
Answered 3	25	1	1
Answered 4&5	16	4	144
Step 5:			
(Step 4) / Step 2			
Answered 1&2	0.04	0.04	4.48
Answered 3	0.93	0.04	0.04
Answered 4&5	0.59	0.15	5.33
Step 6: CHI square			
(Stp 5)[1&2]+[3]+[4&5]	1.56	0.22	9.85
Step 7: Columns -1	2	2	2
Step 8: Tabled Chi Sqr	5.99	5.99	5.99
Table 18-5			
Step 9: Reliability	Not Reliable	Not Reliable	Reliable
ID	QTID.32	QTID.33	QTID.34

QTID.32: The Prioritized Left-to-Right Multi-Column sort methodology is more user friendly.

QTID.33: I believe that overall I was quicker at answering the exercise questions when using the Standard (single-column) sort.

QTID.34: I believe that overall I was quicker at answering the exercise questions when using the Prioritized Left-to-Right Multi-Column sort.

Number of Respondents: 81

CHI-SQUARE on 5

Step 1. List frequencies.	
Answered 1	15
Answered 2	34
Answered 3	27
Answered 4	1
Answered 5	4
Step 2. Expected freq.	16.2
Step 3.	
Observed - Expected	
Answered 1	-1.2
Answered 2	17.8
Answered 3	10.8
Answered 4	-15.2
Answered 5	-12.2
Step 4: (Step 3)squared	
Answered 1	1.44
Answered 2	316.84
Answered 3	116.64
Answered 4	231.04
Answered 5	148.84

Step 5:	
(Step 4) / Step 2	
Answered 1	0.09
Answered 2	19.56
Answered 3	7.20
Answered 4	14.26
Answered 5	9.19
Step 6: CHI square	
(Stp 5) sum(Ans1:Ans5]	50.30
Step 7: Columns -1	4
Step 8: Tabled Chi Sqr	9.49
Table 18-5	
Step 9: Reliability	Reliable
ID	QTID.13

QTID.13: Given the choice, which method of sorting would you like other programs you use to have? 1) Standard Sort (single column) 2) Prioritized Left-to-Right Multi-Column Sort 3) Both 4) Neither 5) Undecided