WEB-BASED ACADEMIC ADVISING SYSTEM

By

Xundong Ding

A Thesis submitted to the Faculty of

The College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

Florida Atlantic University

Boca Raton, Florida

May 2002

UMI Number: 1407886

UMI®

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

WEB-BASED ACADEMIC ADVISING SYSTEM

by
Xundong Ding

This thesis was prepared under the direction of the candidate's thesis advisors, Dr. Sam Hsu and Dr. Oge Marques, Department of Computer Science and Engineering, and has been approved by the members of her supervisory committee. It was submitted to the faculty of The College of Engineering and was accepted in partial fulfillment of the requirements for the degree of Master of Science.

SUPERVISORY COMMITTEE:

Thesis Advisor, Dr. Sam Hsu

Thesis Co-Advisor, Dr. Oge Marques

Dr. Mohammad Ilyas

Chairman, Department of Computer Science and Engineering

Dean, College of Engineering

Vice Provost

4/19/12
Date

To my parents, my daughter, and my husband.

ACKNOWLEDGMENTS

I sincerely thank Dr. Sam Hsu who has not only been an excellent guide throughout my thesis work but also been my mentor in professional life. I also extend my gratefulness to Dr. Marques for his expert suggestions and professional advice from the beginning to the end and Dr. Ilyas's initiation of this project.

I also sincerely thank Dr. Levow for his practical and detailed feedback and suggestions for the project.

The continued support of my family has been a key factor for my education in this country. Words fail to convey the contributions of my family in the completion of this arduous task.

Last but not the least, I thank the staff of the Department of Computer Science and Engineering for their help and support during my study at FAU.

ABSTRACT

Author:	Xundong Ding
Title:	Web-based Academic Advising System
Institution:	Florida Atlantic University
Thesis Advisors:	Dr. Sam Hsu and Dr. Oge Marques
Degree:	Master of Science
Year:	2002

Academic advising is an important and time-consuming task and different tools and techniques can be used to make it an effective and efficient process. This thesis describes the design and development of a Web-based advising system that supplements the conventional advising process. The goals of the system include: to minimize repetitive tasks performed by advisors, to encourage students to adopt a proactive attitude towards advising, to make advising-related information available to remote students in a single place, in electronic format, and to minimize inconsistencies in the advising process. The system supports three different types of users (students, advisors, and secretaries).

This thesis proposes a new Web-based advising system model. It also presents its architecture and an implementation of a prototype. Web-based advising system introduces a new approach towards advising over the Internet. Lessons learned from various experiments of the prototype are discussed in this thesis.

Table of Contents

FIGURES

Chapter 1

INTRODUCTION

Web advising for students is gaining popularity recently among universities across the country. Although only few universities have this kind of system at present, academicians are contemplating to introduce this for various benefits. The Department of Computer Science and Engineering at Florida Atlantic University is working on a project that integrates conventional academic advising and Web-based advising to form a hybrid advising model that can be easily applied toward advising process. Web advising reduces the load for advisors and faculties to some extent by giving an idea to students prior to approaching them. While the advisor assists them in this process, the final responsibility rests on the shoulders of the student. This means that students must actively become involved in their own education. The primary focus of the Web-based advising is to provide educational planning and advising resources in support of students' academic objectives.

1.1 Motivation

Academic advising is an important and time-consuming task and different tools and techniques can be used to make it an effective and efficient process. Most of the process, however, relies on personal interactions between students and advisors, which leads to problems such as inconsistencies among different advisors and poor utilization of resources, since very often a good portion of the advisors' time is spent answering recurrent questions and solving trivial class scheduling problems. In response to these problems, the Department of Computer Science and Engineering (CSE) at Florida Atlantic University (FAU) has been working since 1999 on a project that supplements the conventional advising process with a Web-based advising system.

The CSE department at FAU currently has approximately 700 undergraduate Computer Science (CS) and 300 undergraduate Computer Engineering (CE) students. Advising undergraduate CS and CE students is an important and time-consuming task in the department. Academic advising is one of the faculty's major duties. To accomplish it effectively and efficiently several different schemes have been tried in the past few years, ranging from totally distributed to totally centralized. Starting in the 1999-2000 academic year, the CSE department made another move by assigning all undergraduate advising to three faculty members. Two advisors share the responsibility for the CS students while the third is primarily in charge of the CE students.

Other than adding more faculties to participate in the undergraduate advising, the department also welcomed innovative ideas that could help improving the efficiency of the advising process. One of these ideas was the creation and maintenance of a Web-based advising system for undergraduate CS and CE students, described in the remainder of this thesis.

Before start to design a new advising model, some survey about the related project is very important. Besides that, there are some successful systems existing and helpful for the proposed system. One of them is SASS (Student Academic Support System) and the other is FACTS (Florida Academic Counseling and Tracking for students).

1.2 Survey

This section surveys the existing advising practice and current Web-based advising projects. They are the starting point of our proposed system.

1.2.1 Existing advising practice

Many universities offer students **one-on-one access to advisors.** Different approaches have been developed to open up different communication channels between advisors and students. Some provide advising via email and assign each student to a specific advisor; others develop an advising pool with one or more advisors "on call" at any time to respond to all students' email questions. In some places, advisors have established online chat rooms to provide opportunities for live or asynchronous discussions of students' questions. Weber State University is an example [1].

North Dakota State University adopts a different approach [2]. In order to meet students' needs for any time, anywhere convenience in planning their academic careers, just as many other universities, simply posts all requirements and relevant information on the Web. Although students still do not know exactly what courses they need to take next semester, **at a minimum, all general education and major requirements have been defined clearly and concisely on the Web.**

The web page of Indiana University-Purdue University Columbus provides descriptive or interactive materials [3]. Their **self-help pointers** help students think through the process of choosing a major or other issues.

Link to automated transfer/articulation information is another approach for advising. Such information can help students who are interested in transferring among institutions in the same state. This approach includes information on course equivalencies and guides that enable students to identify community college courses applicable to academic programs they plan to pursue. Some institutions provide transfer/articulation materials specifically relevant to their own programs. In addition, there is a number of useful statewide transfer guides available. ASSIST (Articulation System Stimulating Interinstitutional Student Transfer) [4] is California's official statewide repository of transfer information, offering easy access to a single database. ASSIST can help students determine if they will receive credit for courses they have already taken, and how those courses will apply to specific academic goals.

Washington State University makes **advising guides accessible online to students, and/or advisors** [5]. Online curriculum guides and handbooks can help students, advisors, and faculty stay up to date on current requirements and advising guidelines. Also included are tips for effective advising and information on referrals to other sources of assistances.

1.2.2 Current Web-based advising systems

Universities have recently started to use the Web to help in advising. Although some of them only use Web to link to many helpful sites and some only post the contact information of the advisors, it is a trend that Web plays an important role in academic advising. The following sections show the tasks some universities fulfill in their advising systems on the Web. From these examples we can see that Web-based advising is very popular these days while they still have some limitations.

a) Indiana University -- Insite [6]

The **Indiana Student Information Transaction Environment (INSITE)**, is a Web-based service for Indiana University students, faculty, and staff. It provides secure access to student information. For students, access to information is protected by Student ID number (SIDN) and PIN.

Following information is available through **Insite**:

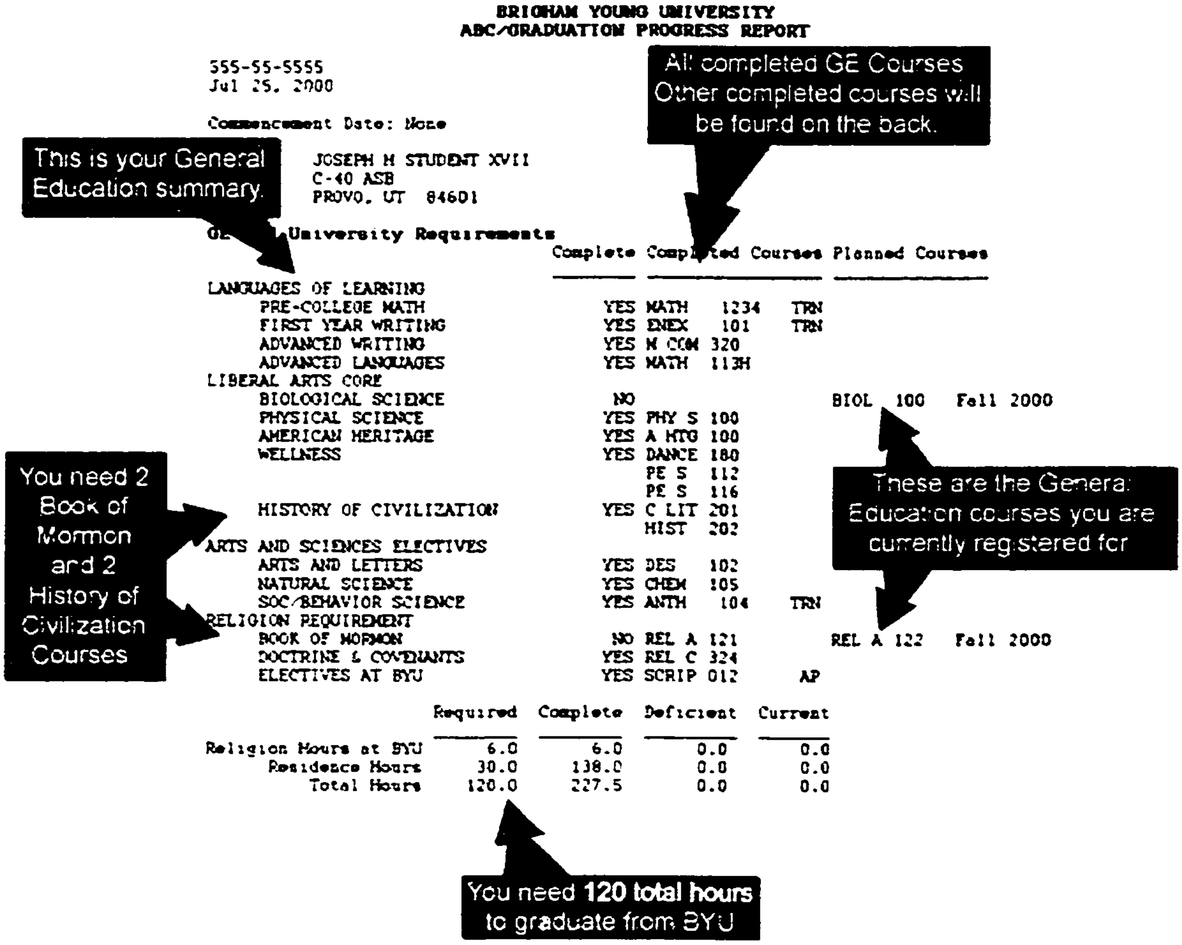
- Produce an advising report for a student's current major.
- Produce an advising report for a different major
- Produce an advising report for a special purpose program.
- Produce an advising report for a program under development.
- Produce an advising report for a special purpose program under development.
- Generate a student transcript (course history).
- See how in-progress courses apply to a student's advising report.

b) Brigham Young University -- Student Planning System [7]

Academic Advisement at Brigham Young University (BYU) combines the benefits of College Advisement Centers and integrates other academic services including the use of the AIM system, Advisement by Computer (ABC) report, and faculty advising to provide direction and support to each student's academic progress. These resources are specifically geared toward addressing specific academic needs and contribute to the quality of each student's overall university experience.

Route Y is the BYU's campus intranet. It is designed to be a secure and personalized information source for current BYU patrons. Patrons include current BYU students, faculty and staff.

To help monitor students' general education and major progress, BYU has created the Advisement by Computer (ABC) report. Students can view and print their ABC report by using the Route Y AIM system. Sample report is in Figure 1.1.



```
                    BRIGHAM YOUNG UNIVERSITY
                 ABC/GRADUATION PROGRESS REPORT

555-55-5555
Jul 25, 2000

Commencement Date: None

                JOSEPH H STUDENT XVII
                C-40 ASB
                PROVO, UT  84601

GE   University Requirements
                              Complete Completed Courses Planned Courses
                              -------- ----------------- ---------------
LANGUAGES OF LEARNING
     PRE-COLLEGE MATH              YES MATH   1234  TRN
     FIRST YEAR WRITING            YES ENEX   101   TRN
     ADVANCED WRITING              YES M COM 320
     ADVANCED LANGUAGES            YES MATH  113H
LIBERAL ARTS CORE
     BIOLOGICAL SCIENCE             NO                   BIOL  100   Fall 2000
     PHYSICAL SCIENCE              YES PHY S 100
     AMERICAN HERITAGE             YES A HTG 100
     WELLNESS                      YES DANCE 180
                                       PE S  112
                                       PE S  116
     HISTORY OF CIVILIZATION       YES C LIT 201
                                       HIST  202
ARTS AND SCIENCES ELECTIVES
     ARTS AND LETTERS              YES DES   102
     NATURAL SCIENCE               YES CHEM  105
     SOC/BEHAVIOR SCIENCE          YES ANTH   104   TRN
RELIGION REQUIREMENT
     BOOK OF MORMON                 NO REL A 121         REL A 122   Fall 2000
     DOCTRINE & COVENANTS          YES REL C 324
     ELECTIVES AT BYU              YES SCRIP 012      AP

                      Required  Complete  Deficient  Current
                      --------  --------  ---------  -------
Religion Hours at BYU      6.0       6.0        0.0      0.0
      Residence Hours     30.0     138.0        0.0      0.0
          Total Hours    120.0     227.5        0.0      0.0
```

All completed GE Courses Other completed courses will be found on the back.
This is your General Education summary.
You need 2 Book of Mormon and 2 History of Civilization Courses
These are the General Education courses you are currently registered for
You need **120 total hours** to graduate from BYU

Figure 1.1 BYU sample report [8]

c) California State University, Monterey Bay (CSUMB) – PLANNER WEB [9]

Planner Web for student advising is the first step in the Advising and Registration process for continuing and newly admitted CSUMB students. All undergraduates and second BA/BS candidates must submit their proposed schedule in order to register. For more information about the registration process, refer to the Advising and Registration Guidelines.

d) Duke University – Pre-major Advising [10]

The Web site doubles as a question site and an appointment site. Students can access advising FAQs and they can make appointments with their advisors.

e) West Washington University -- Academic Advising Center [11]

At West Washington University, academic advising is one of the most important services students will receive. The process of academic advising is a shared responsibility where a student works with faculty and staff academic advisers. Freshmen are assigned a faculty adviser at orientation who serves until a major is declared.

f) North Carolina State University – Advising Central [12]

Advising Central at NC State is designed to provide web and Internet based advising to NC State undergraduate students.

The goals of Advising Central are to:

- Make academic policies clear and meaningful for the students.
- Help students navigate through NC State's human resources to locate and secure advice from the most knowledgeable person in a particular field or college, and help students identify sources that will aid them in clarifying their academic direction and strengthening their academic skills.

The above survey concluded that there are currently many universities in the United States with Web-based advising systems at work. Most of the pages entitled *Web-based advising* are typically a bulletin board with advising-related announcements; a repository of official documents in PDF or HTML format; a collection of useful links that help students get official advising-related information off the Web; or a combination of those. They do not include any scripts or ASP (Active Server Page) [13] programs to process specific student information and produce customized advice for students. Indiana University is an exception, but the Web system is only for advisors to get their advisee's information, it is not designed for students.

From a commercial perspective, to our best knowledge the only product that seems to address the issue of academic advising in a way similar to our work is DegreeWorks by Software Research Northwest, Inc [14]. It is worth mentioning, however, that part of the functionality provided by DegreeWorks and INSITE (obtaining transcripts and audit reports for advising purposes) is already provided by FACTS (Florida Academic Counseling and Tracking System) [15], a statewide Web-based system for academic advising.

1.3 Goals and Objectives

The following section defines the advising objectives and goals.

1.3.1 Advising purpose

a) To help students select courses necessary to complete his/her program. The student should have a clear idea of what the advisor recommends for the next semester (in writing).

b) To let the advisor be the person that the student identifies with as a contact for assistance at the college. Advisor is the person they should seek out for information, and referrals. This can make a difference in retention.

c) To provide career information to the student.

d) To refer the student to other areas of the college as needed.

1.3.2 Project overall goals and design objectives

Based on the above discussion, we are proposing a Web-based undergraduate advising system with the following five overall goals and two design objectives.

Five overall goals:

a) **To minimize repetitive tasks currently performed by advisors**: The proposed system aims at minimizing the amount of time and energy spent by advisors on repetitive tasks such as answering frequently asked questions in person, by phone, or email.

b) **To encourage students to adopt a proactive attitude towards advising-related issues**: By making all the information available in one place and providing the student with tools that help answering their most frequent advising questions, it is expected that the students' attitude towards academic advising will move from passive ("let the advisor tell me which courses I should take next") to proactive ("let me check the system to see which courses I will be able to take next").

c) **To extend the availability of official advising-related information to remote students**: One of the most immediate and visible benefits of our project is to make official relevant advising information available on the Web. In addition to being accessed from anywhere, efforts have been made to guarantee that the information is up-to-date and consistent.

d) **To provide academic guidance in a consistent way**: By having all the reference information stored electronically in one place and using the same (set of) program(s) to advise students on which courses to take next, inconsistencies that used to occur with personal advising are minimized.

e) **To make advising-related information available in a single place, in electronic format**: It is expected that the system be a portal for any undergraduate CSE student in need of advising-related information.

Two design objectives:

a) **To maintain an on-line repository of the most frequently asked questions (FAQs).** Providing a comprehensive, well structured, up-to-date repository of FAQs allows students to find the answers to their questions in a fast and easy way. Moreover, it saves advisors' time, reduces congestion in the advisors' offices during peak times, encourages advisors to document the answers and post them in electronic format, allows for easier supervision of the quality of the advising process, and reduces inconsistencies and ambiguities in the process.

b) **To develop a set of on-line forms that helps students interactively to get a list of courses to take next.** This is the "intelligent" portion of the system: based on the user's input (courses already taken), the degree requirements, the database of existing and available courses and their prerequisites, the system provides the student with customized advice on which classes to register for in the next term(s).

1.4 Organization of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 describes the design of the Web-based academic advising system, highlighting the design of use interface and database. It further describes the algorithm of the advising process. Chapter 3 is on the implementation of the Web-based advising system, concentrating on three different subsystems and implementation issues behind its development. Chapter 4 analize the results from the survey collected from students, advisors and administrators. Finally Chapter 5 concludes the thesis with a summary of general features of the proposed system and also discusses the possible future extension for this system, this chapter also describes the advantages and disadvantages of the system.

Chapter 2

SYSTEM DESIGN

This chapter is concerned mainly with the design of the Web-based advising system. The first few sections are on the design overview, followed by discussions of the three constituent subsystems.

2.1 Overview

The user requirements drive the design of the Web-based advising system [16]. The system supports three different types of users: students (who will use the system to get academic advising), advisors (who will post, update, and manage information used by the system and made available to the student users), and secretaries (who will access, update, and manage information used by the system and made available to the student users). Each different type of user should have a different user interface. Some of the user interfaces are static files, others are dynamically generated. Backend data and business logic are other important parts of the system besides user interface.

2.1.1 Role based User Interface

The system's framework basically consists of a main page from which all the options are accessible, and a set of control programs. The main page has links to each major's degree requirements and career guide, other advising-related information of interest, the FAQ page, and the forms to input course information and get customized advice on next courses to take.

The design of the system is centered on three types of users, namely students, advisors (faculty), and administrators (secretaries).

- **Student users**: who will use the system to get academic advising, and browse the FAQs, email their advisor, and get to many useful links.
- **Faculty users (advisors)**: who will post, update, and manage information used by the FAQ subsystem
- **Administrative users (secretaries)**: who will access, update, and manage information used by the "next courses to take" subsystem.

Each different type of user has a different GUI (Graphical User Interface), a set of privileges/rights, and a set of possible actions. Access to privileged information (maintained by faculty and/or administrative users) is password-protected. This relationship is illustrated in Figure 2.1.

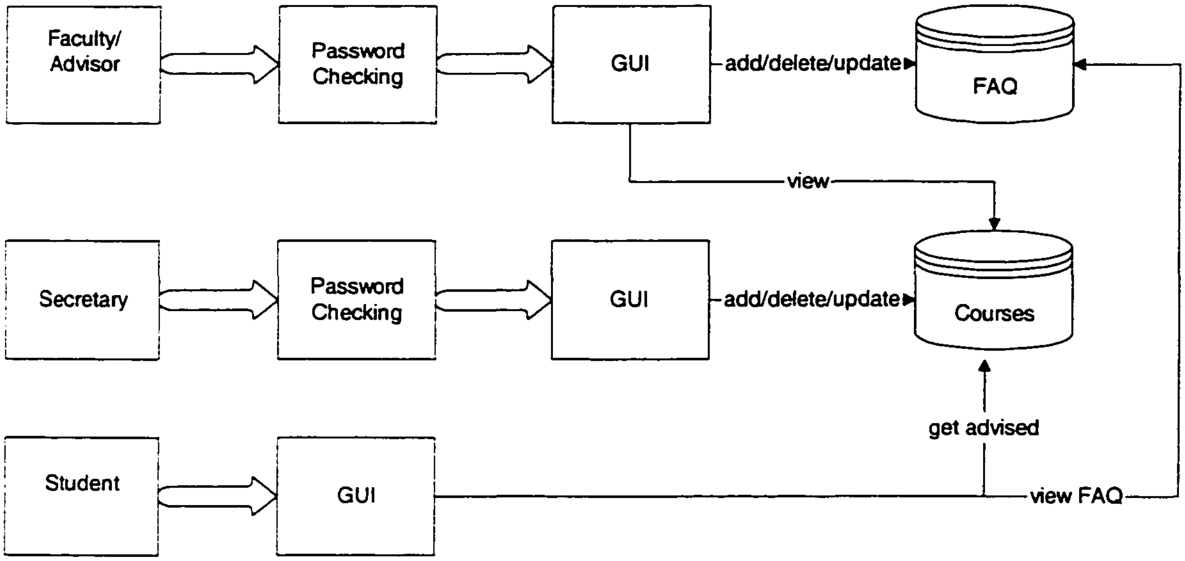


Figure 2.1 The Web-based system overview

2.1.2 System Architecture

The logical structure of the Web-based advising system is a three-tier system. It consists of user interface, a Web server and back-end data. Tier one is user interfaces which are supported by the Web browser. Users directly interact with the GUIs whose functionality is to collect users' input data for future processing. Tier two is the business logic part which resides on the Web server. The Web server will interpret the information it receives from the browser, execute the commands and send the results back to the browser. Tier three is backend data which is stored in the tables in the database.

The Web-based advising system is composed of three subsystems: the FAQ subsystem, the Course subsystem and the Advising subsystem. Each subsystem has its own set of functions and they are three-tier systems too. All the three subsystems can be accessed from the system main page, and from every other page of the system. Each subsystem consists of its own GUI, control programs and its own tables. The system structure is shown in Figure 2.2.

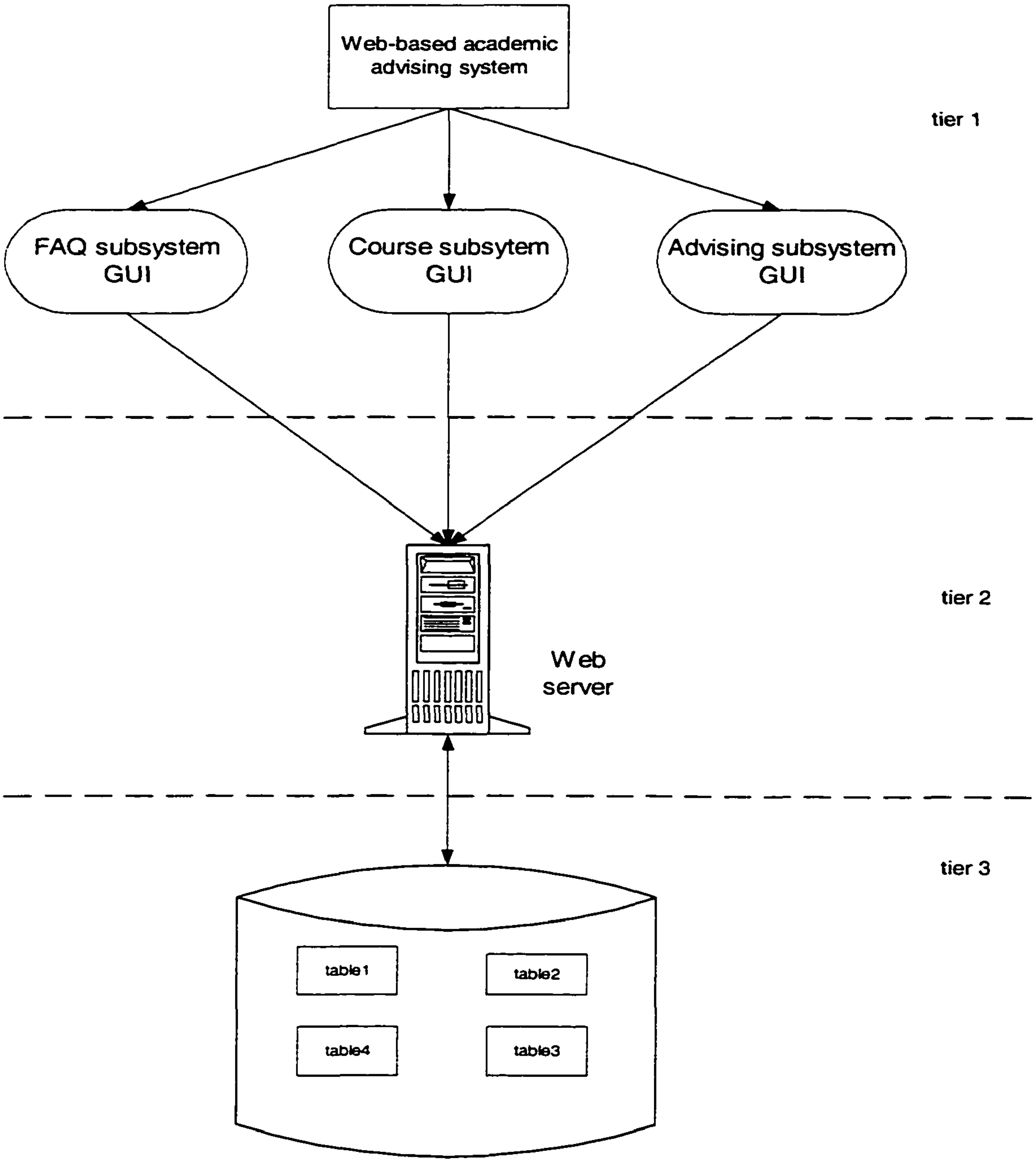


Figure 2.2 Web-base academic advising system architecture

2.1.3 Design requirements

The following functional requirements are the basic guidelines for designing the Web-based academic advising system.

a) The system should support at least three different types of users: students, advisors and secretaries.

b) Each different type of user has a different view of the system. A *view* in this context includes a pleasant and friendly GUI, a set of privileges/rights, and a set of possible actions.

c) Classified or sensitive information (accessed by faculty and/or administrative users) should be password-protected.

d) Any user should be able to navigate through the site in an easy and intuitive manner.

e) Advising-related information should be up-to-date and organized in a meaningful way.

f) All incoming data input by student users should be checked against the department's degree requirements.

g) The system should support multiple simultaneous users.

h) Student users should be able to :

- Input information on the courses they have already taken using a friendly GUI that resembles the worksheets currently used by CSE advisors.
- Get advice on which courses to take next based on the information input to the system.
- Have easy access to relevant related information (regardless of its location), such as course descriptions, advising FAQs, and career opportunities, among many others.

i) Faculty users (advisors) will be able to:

- Login and Logout the system.
- Input and update general advising-related information.
- Input and update information on courses that will be offered next semesters and related data.
- Input and update information on course prerequisites.
- Input and update information on degree requirements.

j) Administrative users (secretaries) will be able to :

- Input and update information on courses that will be offered next semester and on related data.
- Input and update information on course prerequisites.
- Input and update information on degree requirements.

The following discussions will focus on the three subsystems.

2.2 The FAQ subsystem

The FAQ (Frequent Asked Quesiton) subsystem is a system that maintains a list of FAQs of undergraduate student advising. Students can view the FAQs which are stored in the database while secretaries and advisors can edit them. Since questions may change when university requirements change, an editing tool to help the FAQ maintainers to update relevant FAQs is thus needed. This subsystem is designed for those who are not familiar with HTML to finish all tasks easily and conveniently.

Students can access to the system's main page, and then a top link will guide them to the FAQ page which includes all the FAQs for both Computer Science and Computer Engineering students. If they have any questions, they can click a button to send an email to their advisors directly. The email gateway will provide them quick and satisfactory service for contacting their advisors without leaving the Web browser.

Advisors can browse the FAQs the same way as students do. And they can also login to a user interface which is designed specifically for them. It is very straightforward for them to add, delete and update target FAQs.

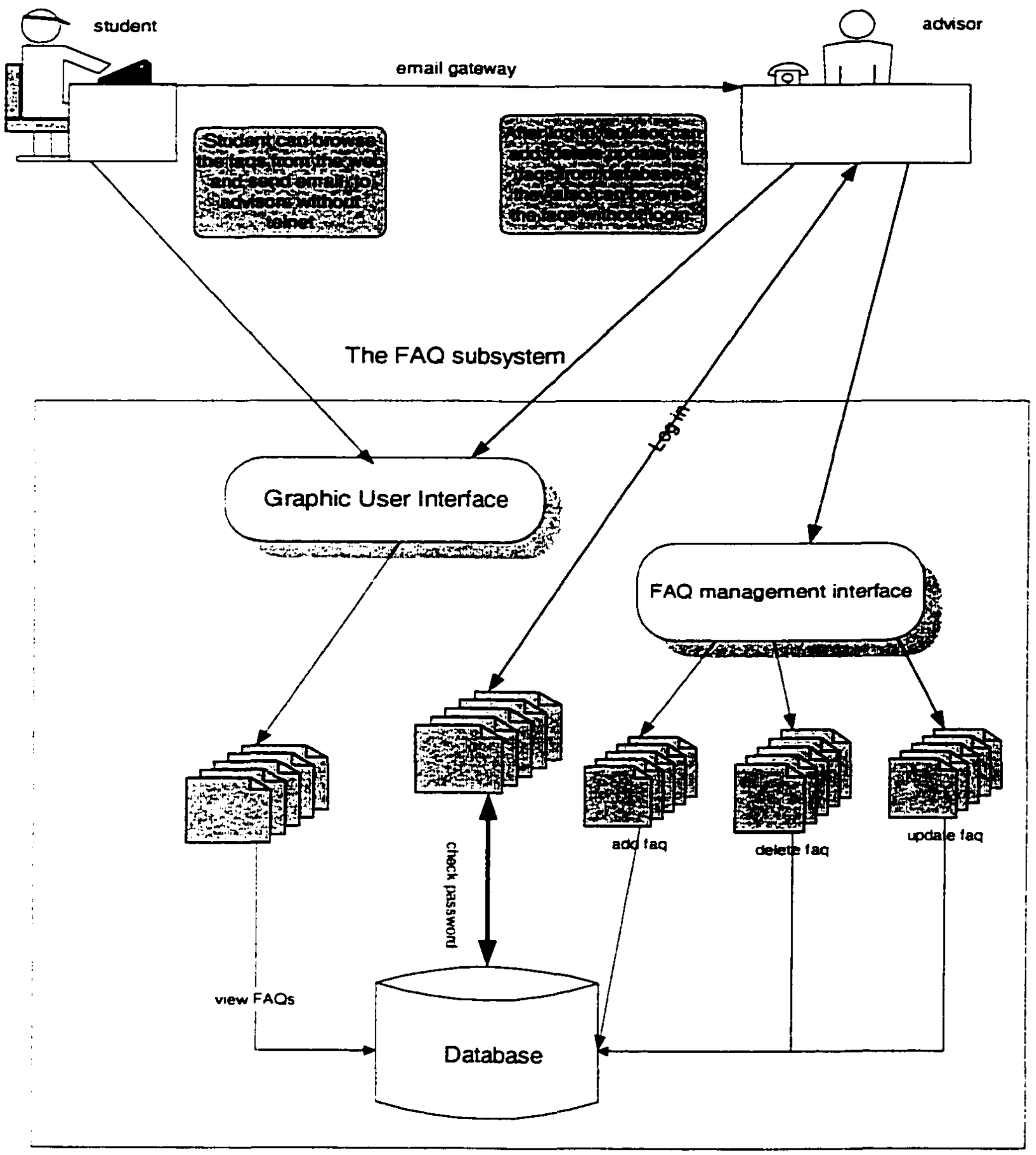

student
advisor
email gateway
Student can browse the faqs from the web and send email to advisors without telnet
After log in, advisor can add/delete/update the faqs from database; they also can browse the faqs without login
The FAQ subsystem
Log in
Graphic User Interface
FAQ management interface
check password
add faq
delete faq
update faq
view FAQs
Database

Figure 2.3 The FAQ subsystem

2.3 The Course Subsystem

The Course subsystem is a system similar to the FAQ subsystem. However, it is used by advisors or administrators who will maintain the course information. Students do not have privileges to go directly into this subsystem. This system is dedicated to support the Advising subsystem. Therefore the students use the Course subsystem indirectly by getting advised. The main functionality of this system is to maintain the information of all the courses that the undergraduate Computer Science and Computer Engineering students need to take, including core courses, electives, prerequisites, relevant university requirements etc. Administrators can login to the system and update the course information such as credits, prerequisites. Please see Figure 2.4 for an overall view of the course system.

The Course subsystem has a front-end GUI to manage the system. The GUI is designed to help the administrator maintain the underlying database with ease.

Figure 2.4 shows the basic design of the Course Subsystem. Administrators need to login through a login interface. Their username and password will be checked first before they can get access to a Course Management user interface. Operations to add, delete, update the course information can start from there.

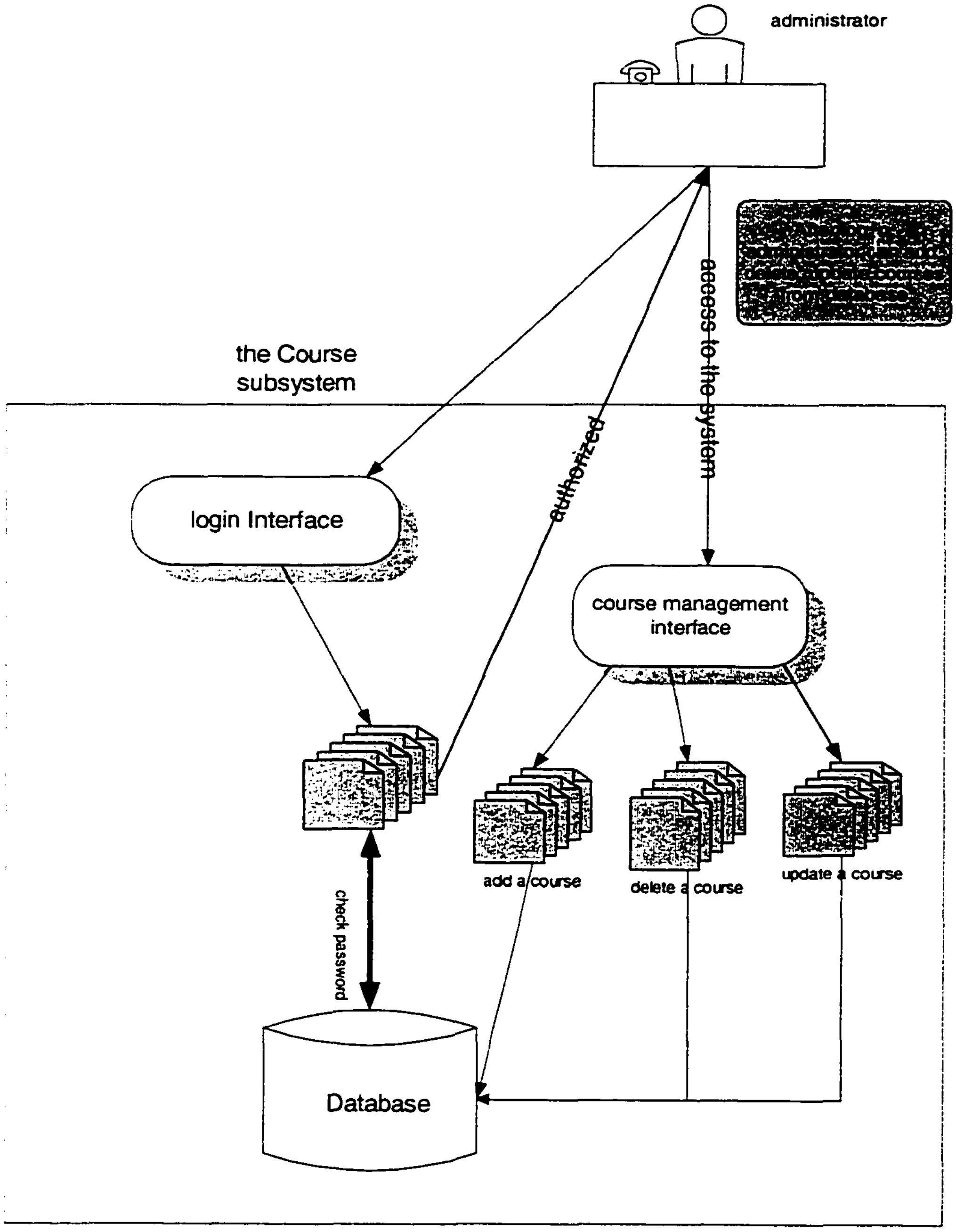


Figure 2.4 The course subsystem

2.4 The Advising Subsystem

The Advising subsystem is the most important subsystem among the three subsystems since it is the place where students get advised. Figure 2.5 shows the design of the subsystem. Different GUI will be generated dynamically according to different degree requirements. User input of the courses a student has taken will be captured by the GUI and sent to a control program, which will process the input using a dedicated algorithm and access the data in the database to produce a final result of advising. The advising process succeeds with students getting their expected advising results.

The Advising subsystem put its emphasis on the algorithm to sift all the information, rules, and to get a reasonable and correct result. The Advising subsystem does not produce the final result simply by doing the subtraction, it can "think" intelligently under different scenarios by using the rules and requirements stored in the database.

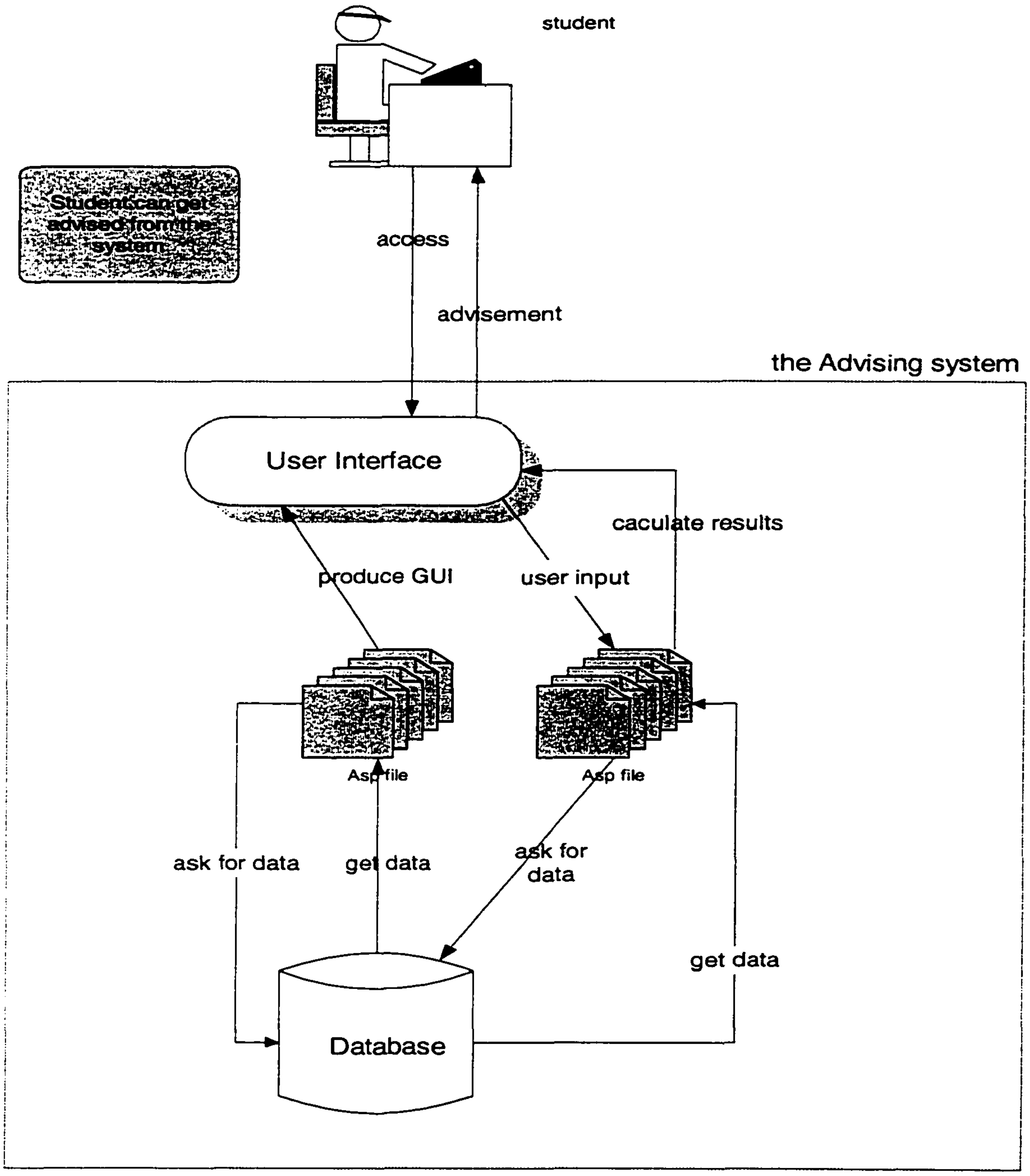


Figure 2.5 The Advising Subsystem

2.4.1 Advising procedure

Background :

There are three different types of degree program under the two undergraduate Computer Science and Computer Engineering tracks [17]. Four-year student program, second-bachelor student program and transfer student program. Since, in essence the second-bachelor program and transfer student program are part of the four-year student program with their own special requirements, and Computer Science and Computer Engineering major have the same course structure and share many courses, only the Computer Science four-year program will be introduced in detail here.

The Computer Science four-year student program requires students meet both university requirements and department requirements for graduation. University requirements are the courses required to take by undergraduate students by university policy. These courses include: Humanities courses, Social Sciences courses, English, Science electives and Speech. Department requirements include core courses, Computer Science electives, Math courses, Math electives, Physics, and Foreign Language knowledge.

There are several rules and requirements need to be implemented during advising according to the department policy. They are:

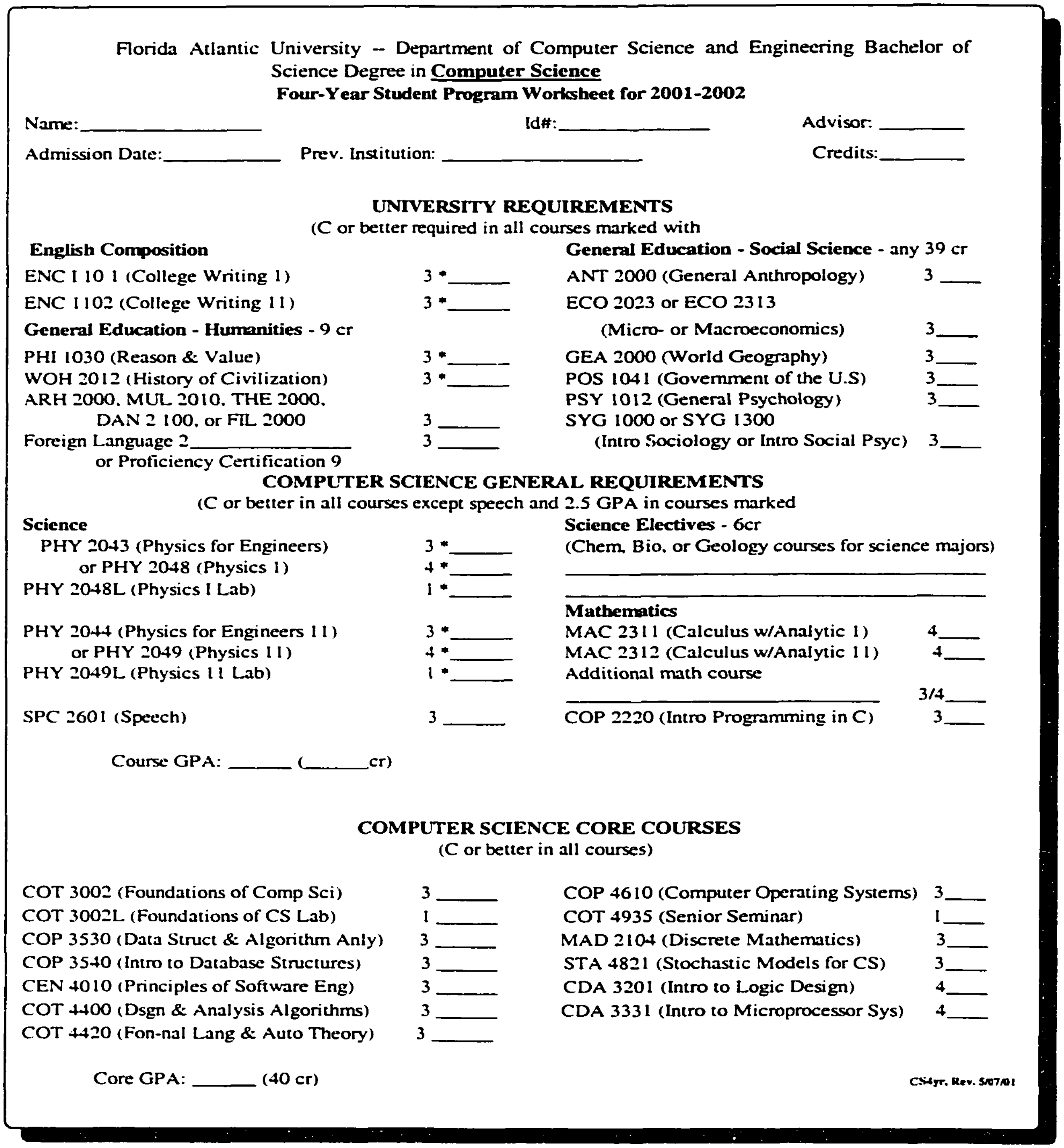
a) Prerequisites --- Many Computer Science core courses have prerequisites. If a course has prerequisites, then these prerequisites need to be taken before taking the course.

b) One out of a three-course set --- students only need to take one course out of CDA 4105 (Structured Computer Architecture), CDA 4150 (Computer Design I), and CDA 4170 (CAD-Based Computer Design).

c) Number limitation ---- Students need to take three Humanity courses and three Social Science courses, and two Science electives.

d) Other rules --- These rules are not currently checked by the proposed system, but the system will remind students. MAC 2311 (Calculus w/Analytic I) can be taken if the student has passed a pre-test or he has taken MAC 1147 or MAC 1140 and MAC 1114 (these courses are offered in community college).

Above are the basic rules and requirements guides through every advising process. The department of Computer Science and Engineering offers students a program worksheet to help them do a self-check which is the basic idea where the Advising system originates. Figure 2.6 is a sample worksheet of Computer Science four-year student program.



Florida Atlantic University -- Department of Computer Science and Engineering Bachelor of Science Degree in <u>Computer Science</u>

Four-Year Student Program Worksheet for 2001-2002

Name:________________ Id#:______________ Advisor: ________

Admission Date:___________ Prev. Institution: ___________________ Credits:________

UNIVERSITY REQUIREMENTS

(C or better required in all courses marked with

English Composition	
ENC I 10 1 (College Writing 1)	3 *______
ENC 1102 (College Writing 11)	3 *______
General Education - Humanities - 9 cr	
PHI 1030 (Reason & Value)	3 *______
WOH 2012 (History of Civilization)	3 *______
ARH 2000, MUL 2010, THE 2000, DAN 2 100, or FIL 2000	3 ______
Foreign Language 2______________ or Proficiency Certification 9	3 ______

General Education - Social Science - any 39 cr	
ANT 2000 (General Anthropology)	3 ____
ECO 2023 or ECO 2313 (Micro- or Macroeconomics)	3____
GEA 2000 (World Geography)	3____
POS 1041 (Government of the U.S)	3____
PSY 1012 (General Psychology)	3____
SYG 1000 or SYG 1300 (Intro Sociology or Intro Social Psyc)	3____

COMPUTER SCIENCE GENERAL REQUIREMENTS

(C or better in all courses except speech and 2.5 GPA in courses marked

Science	
PHY 2043 (Physics for Engineers)	3 *______
or PHY 2048 (Physics 1)	4 *______
PHY 2048L (Physics I Lab)	1 *______
PHY 2044 (Physics for Engineers 11)	3 *______
or PHY 2049 (Physics 11)	4 *______
PHY 2049L (Physics 11 Lab)	1 *______
SPC 2601 (Speech)	3 ______

Science Electives - 6cr	
(Chem, Bio, or Geology courses for science majors)	
________________________________	
________________________________	
Mathematics	
MAC 2311 (Calculus w/Analytic 1)	4____
MAC 2312 (Calculus w/Analytic 11)	4____
Additional math course ___________________________	3/4____
COP 2220 (Intro Programming in C)	3____

Course GPA: ______ (_______cr)

COMPUTER SCIENCE CORE COURSES

(C or better in all courses)

COT 3002 (Foundations of Comp Sci)	3 ______
COT 3002L (Foundations of CS Lab)	1 ______
COP 3530 (Data Struct & Algorithm Anly)	3 ______
COP 3540 (Intro to Database Structures)	3 ______
CEN 4010 (Principles of Software Eng)	3 ______
COT 4400 (Dsgn & Analysis Algorithms)	3 ______
COT 4420 (Fon-nal Lang & Auto Theory)	3 ______

COP 4610 (Computer Operating Systems)	3____
COT 4935 (Senior Seminar)	1____
MAD 2104 (Discrete Mathematics)	3____
STA 4821 (Stochastic Models for CS)	3____
CDA 3201 (Intro to Logic Design)	4____
CDA 3331 (Intro to Microprocessor Sys)	4____

Core GPA: ______ (40 cr)

CS4yr, Rev. 5/07/01

Figure 2.6 Sample worksheet for CS four-year program

Advising procedure:

a) Produce GUI as to user's choice

b) Get user input

c) Process Core courses

d) Undergraduate status

e) Process Humanity courses

f) Process Social Science courses

g) Process Computer Science/Engineering electives

h) Process Science electives

i) Process Math electives

Now the following discussions are based on the details of these steps.

2.4.2 Detail Processing

a) **Produce user interface** – As a student under different program, different user interface will be created.

b) **Get user input** – User input in forms are captured by control programs

c) **Process core courses**

- Get courses that have no prerequisites left except "senior status" courses. These courses are the courses chosen from all the core courses existing in the database that will be taken next semester and all their prerequisites have been taken.

- Eliminate duplicates between the courses that will be taken next semester and the core courses that have been taken.

- Process "one out of a three-course set" scenario --- Get required number from the standard table from the database, find how many need to be taken, how many have been taken and calculate the final results.

b) Process undergraduate status

Undergraduate status as freshman, sophomore, junior and senior is an important standard by which to decide if a student can take a senior level course such as COT 4935 (senior seminar) for both Computer Science students and Computer Engineering students and EGN 4914 (Engineering Design I) and EGN 4915 (Engineering Design II) for Computer Engineering students.

c) Process Humanities and Social Science courses

Humanities and Social Science courses do not have prerequisites. They only have the requirements of the number of the courses to take. If we hard code all the requirement numbers in the system, it will be difficult for the administrator to trace all the numbers and make changes. The solution is to build a table in the database, put all the requirements for different courses in the table. The required number will be extracted from the table during advising. Students will get advised based on the number of courses they input and the number of courses required from the table. We also provide a list of Humanities and Social Science courses which is included in the advising results for the student to choose for the next semester.

d) Process Physics courses

Since Physics courses are all prerequisites of EEL 3111 (Network Analysis I) for Computer Engineering students and they take MAC 2311 (Calculus I) and MAC 2312 (Calculus II) as prerequisites. They are under categories of core courses.

e) Process Chemistry courses

Chemistry courses are different from physics courses. They do not have any prerequisites and they are not prerequisites for any particular course. But they are still treated as core courses.

f) Process Math core courses

Math core courses like MAC 2311 (Calculus I), MAC 2312 (Calculus II) and MAC 2313 (Calculus III) are most basic and important prerequisites for Computer Science and Computer Engineering major. They will be treated as part of the core courses through advising process. And students will get a reminder that they should have taken MAC 1147 or MAC 1140 and MAC 1114 or pass pre test before they take MAC 2311.

g) Process Computer Science and Computer Engineering electives

The system will get user input by asking user to choose the number of computer electives they have taken. Then the input number will be compared to the required number from database. Student will get advised on how many more computer electives they need to take.

h) Process Science electives (Computer Science only)

The system will get user input by asking use to choose the number of science electives they have taken. Then the input number will be compared to the required number from database. Students will get advised how many more science electives they need to take.

i) Process Math electives (Computer Science only)

The system will get user input by asking use to choose the number of math electives they have taken. Then the input number will be compared to the required number from database. Students will get advised how many more math electives they need to take.

2.5 Summary

The advising procedure described above is a general advising procedure for all majors, and there are some variations for Computer Science and Computer Engineering majors, as well as for four-year, second bachelor and transfer students. But the main logic used is the same.

Now a Web-based academic system has been designed according to the functional requirements. It is the frame work for the application and it remains to be seen whether it is feasible for implementation.

Chapter 3

IMPLEMENTATION

This chapter introduces the implementation of the proposed system. Sample tables, different interfaces and advising results of each subsystem are provided for a clear view of how the subsystems are integrated together to make the advising procedure successful. The structures of the three subsystems are described in detail.

3.1 Introduction

There are three subsystems in the Web-based advising system. All of the three subsystems have their own functions and intended users. Although they have different tasks while providing services to their users, the basic operations are similar. They are retrieving data from the database, processing the data, and then send back the final results to the user who is using the web browser. Another important Web technology is used to allow the user to send request and dynamically generate Web contents using server-side scripting. This technology is ASP which is supported by IIS (Internet Information Server) server. From the whole system point of view, the main structure of the system consists of a database, ASP files and a Microsoft IIS server [18]. They will be discussed in detail later.

3.2 Database creation

Microsoft Access is the database management system used in the implementation of the proposed system. It is ideal for small to medium size database and has enough capacity and ability to dedicate to this project.

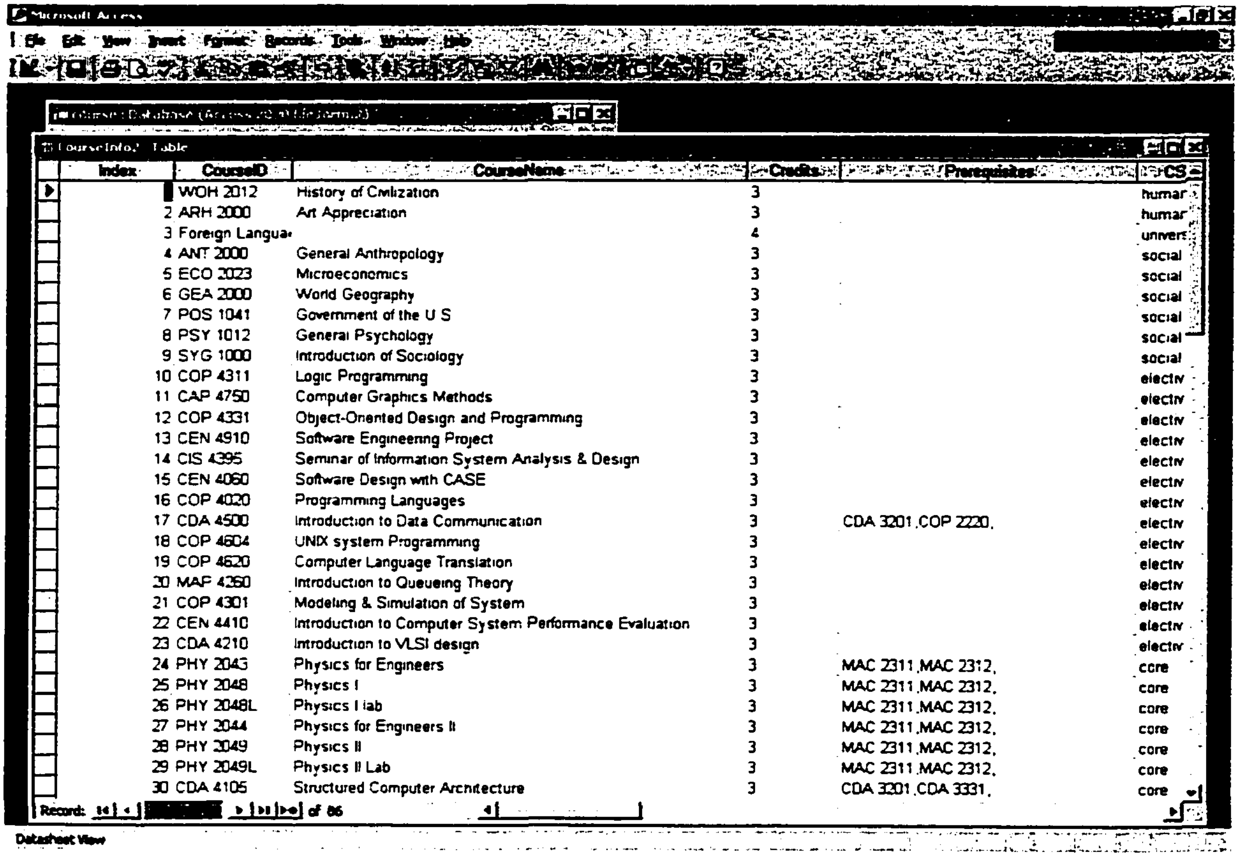
3.2.1 Database implementation

A relational database [19] is created for the web-based advising system. The name of the database is *course*. It consists of seven tables, namely: *CourseInfo, Prerequisite, standard, FAQ, advisor_login,* and *administrator_login*. The *Index* field in *courseInfo* is also the field in *prerequisite* table. The *Index* field makes these tables relate to each other.

3.2.2 Tables

Table 1: *CourseInfo*: It is the most important table in the database. It has nineteen fields, currently eighty records. This table stores the information of a course, its brief name, full name, credits, prerequisites and some other information needed for advising process. Both Course subsystem and Advising subsystem will access this table.

Figure 3.1 shows a sample of *CourseInfo* table.



Index	CourseID	CourseName	Credits	Prerequisites	CS
	WOH 2012	History of Civilization	3		humar
2	ARH 2000	Art Appreciation	3		humar
3	Foreign Langua		4		univers
4	ANT 2000	General Anthropology	3		social
5	ECO 2023	Microeconomics	3		social
6	GEA 2000	World Geography	3		social
7	POS 1041	Government of the U S	3		social
8	PSY 1012	General Psychology	3		social
9	SYG 1000	Introduction of Sociology	3		social
10	COP 4311	Logic Programming	3		electiv
11	CAP 4750	Computer Graphics Methods	3		electiv
12	COP 4331	Object-Oriented Design and Programming	3		electiv
13	CEN 4910	Software Engineering Project	3		electiv
14	CIS 4395	Seminar of Information System Analysis & Design	3		electiv
15	CEN 4060	Software Design with CASE	3		electiv
16	COP 4020	Programming Languages	3		electiv
17	CDA 4500	Introduction to Data Communication	3	CDA 3201,COP 2220,	electiv
18	COP 4604	UNIX system Programming	3		electiv
19	COP 4620	Computer Language Translation	3		electiv
20	MAP 4260	Introduction to Queueing Theory	3		electiv
21	COP 4301	Modeling & Simulation of System	3		electiv
22	CEN 4410	Introduction to Computer System Performance Evaluation	3		electiv
23	CDA 4210	Introduction to VLSI design	3		electiv
24	PHY 2043	Physics for Engineers	3	MAC 2311,MAC 2312,	core
25	PHY 2048	Physics I	3	MAC 2311,MAC 2312,	core
26	PHY 2048L	Physics I lab	3	MAC 2311,MAC 2312,	core
27	PHY 2044	Physics for Engineers II	3	MAC 2311,MAC 2312,	core
28	PHY 2049	Physics II	3	MAC 2311,MAC 2312,	core
29	PHY 2049L	Physics II Lab	3	MAC 2311,MAC 2312,	core
30	CDA 4105	Structured Computer Architecture	3	CDA 3201,CDA 3331,	core

Figure 3.1 A sample CourseInfo table in database

Table 2: Prerequisite: This table has two fields. It stores the information of the relationship of a course and its prerequisites. There is no primary key in this table, as one course can have many prerequisites and one course can be a prerequisite course to more than one course. This is another important table for the Advising subsystem.

Table 3: standard: This table has two fields. They are the course type and its required number. For example, Computer Science student needs to take three humanity courses, this information will be stored in the table for future advising purposes.

Table 4: FAQ: This is an important table for FAQ subsystem. The advising subsystem and course subsystem will not use it. The table stores the information of the questions and answers, their input time and the name of the advisor who edit the question.

Table 5: advisor_login: This table stores the user name and password of the advisors. It will be accessed when the advisors need to login the FAQ subsystem.

Table 6: administrator_login: This table stores the user name and password of the administrators. It will be accessed when the administrators need to login to the Course subsystem.

3.3 The GUI

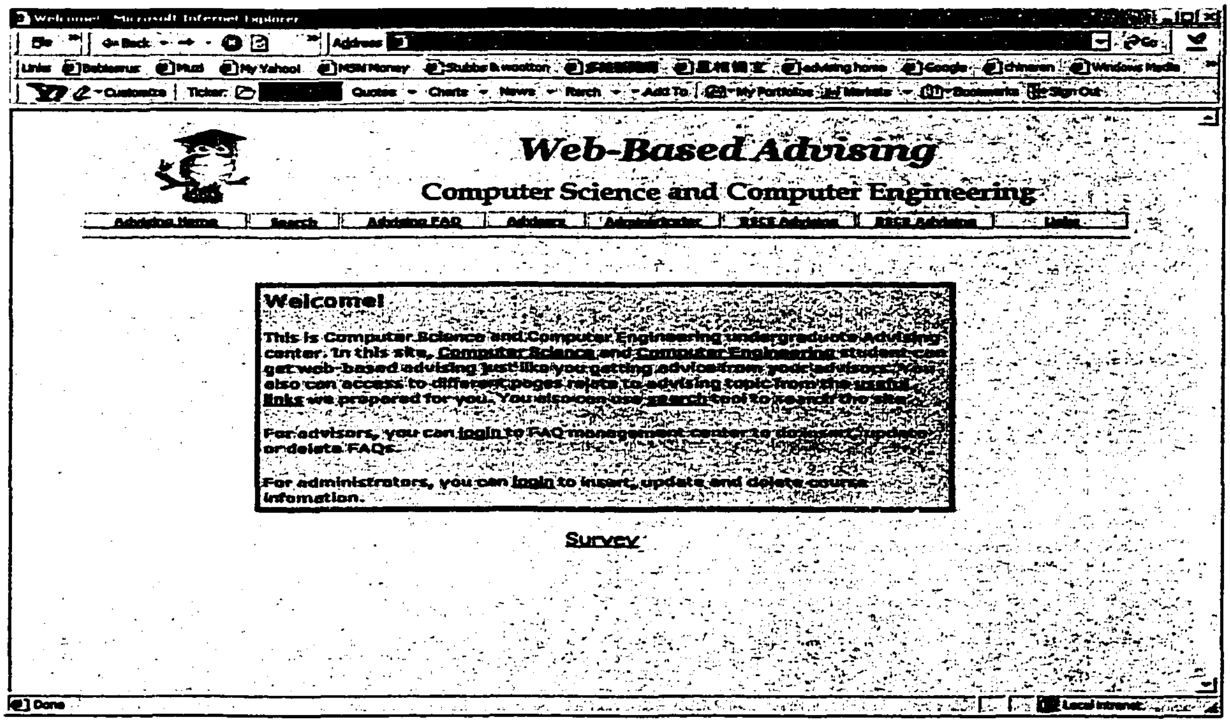
The GUI is designed using ASP only. The design of the User Interface provides a friendly environment to let the user navigate through pages easily and smoothly. To provide the user a consistent navigation, Computer Science pages and Computer Engineering pages have their own but similar background. Therefore ASP files are used instead of HTML files to include a header file and a footer file. A GUI page contains mainly Check boxes, Radio Buttons and Input Buttons

a) Check boxes are named differently to suit the programming requirements. Course Number is provided as the value of the check box while designing the GUI. The value of the check boxes can be obtained by referring to the name of the check box.

b) Radio Buttons: Value of the radio buttons is obtained similar way as check boxes. Radio buttons allow user to select only one option out of many.

Input Buttons: These buttons help to navigate between pages.

Here is a sample of the main page of the whole site. See Figure 3.2.



Welcome!

This is Computer Science and Computer Engineering undergraduate Advising center. In this site, Computer Science and Computer Engineering student can get web-based advising just like you getting advice from your advisor. You also can access to different pages relate to advising topic from the useful links we prepared for you. You also can use search tool to search the site.

For advisors, you can login to FAQ management center to do insert, update or delete FAQs.

For administrators, you can login to insert, update and delete course infomation.

Figure 3.2 Web-based advising main page

3.4 ASP programs

ASP programs are the joint between the database and the user's browser. They get data from the database, processing it, and send back to the users. Let us discuss the ASP technology briefly here.

ASP technology:

Active Server Pages is a technology originally created by Microsoft as an answer to the sometimes complex problems posed by CGI application development. It allows the user to use any scripting language, from VBScript to Python, to create real-world Web applications. VBScript is used in this project.

The ASP technology is encapsulated in a single, small DLL called ASP.DLL. This DLL is an ISAPI filter that resides in the same memory space as Internet Information Server. Whenever a user requests a file whose file extension is .ASP, the ASP ISAPI filter handles the interpretation. ASP then loads any required scripting language interpreter DLLs into memory, executes any server-side code found in the Active Server Page, and passes the resulting HTML to the Web server, which then sends it to the requesting browser. Figure 3.3 illustrates this process.

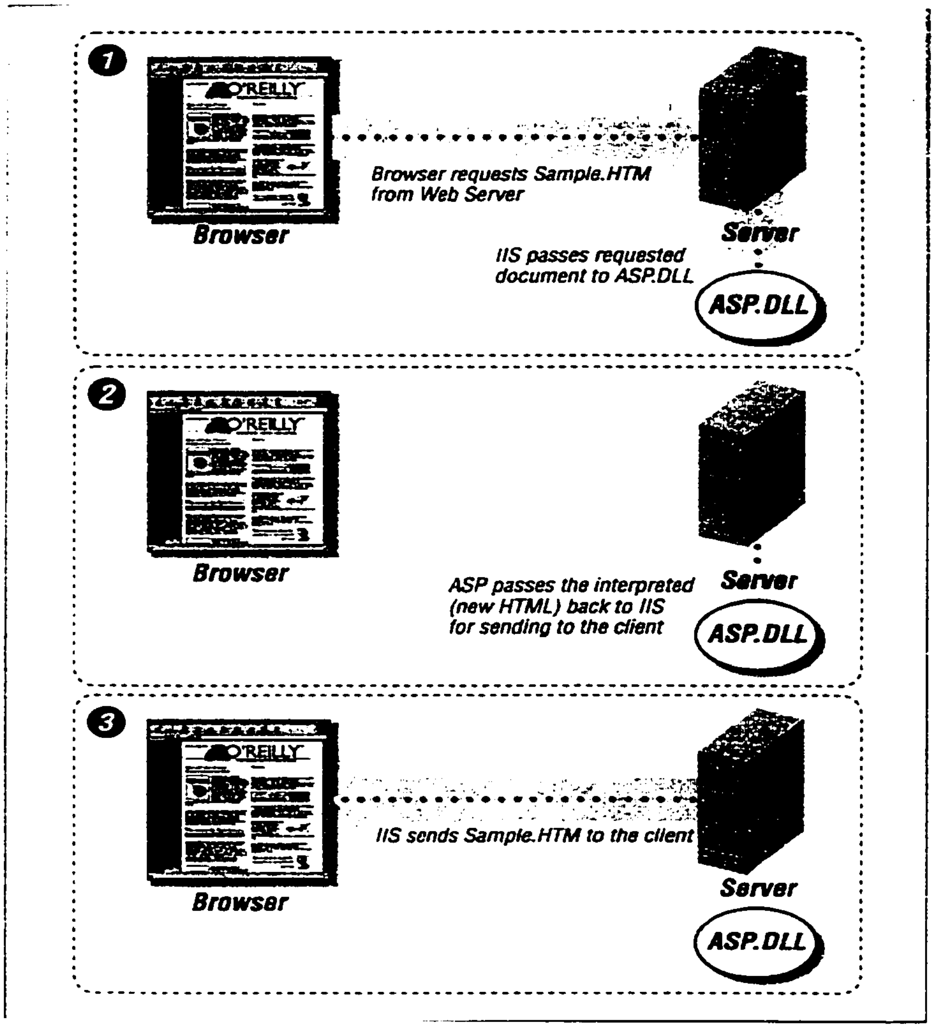


Figure 3.3 Dynamical generation of Web contents [21]

ASP files:

There are 32 ASP files in the whole system. VBScript is used to embed in the HTML except the *email gateway* which is implemented in ASP script.

3.5 Email gateway

The email gateway in the project is an implementation of the Simple Mail Transfer Protocol (SMTP) [20] which provides the interface and the means to send email from Active Server Pages (ASP) with only a few lines of code. The proposed system only concentrates on a single object: NewMail. The NewMail object provides the basic functionality to send email, such as the To, From, CC, BCC, Subject, and Body properties and the Send method. (Figure 3.5 email.asp).

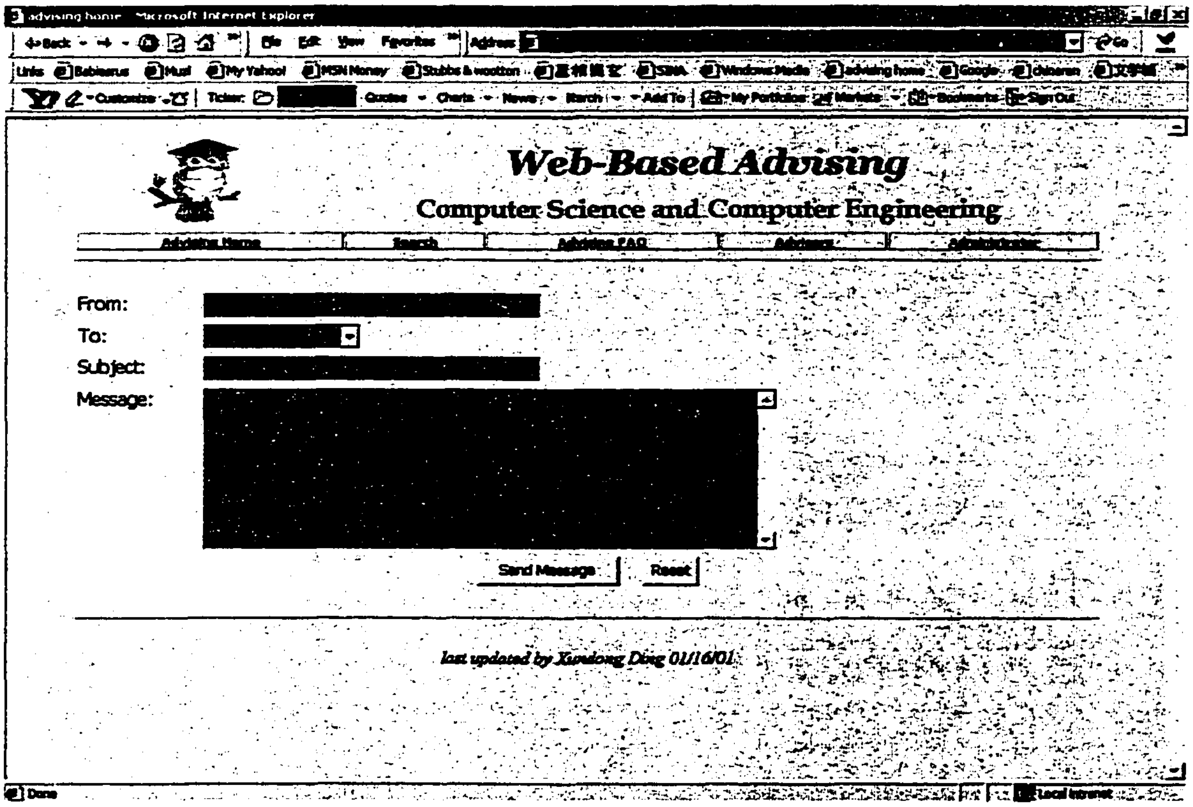
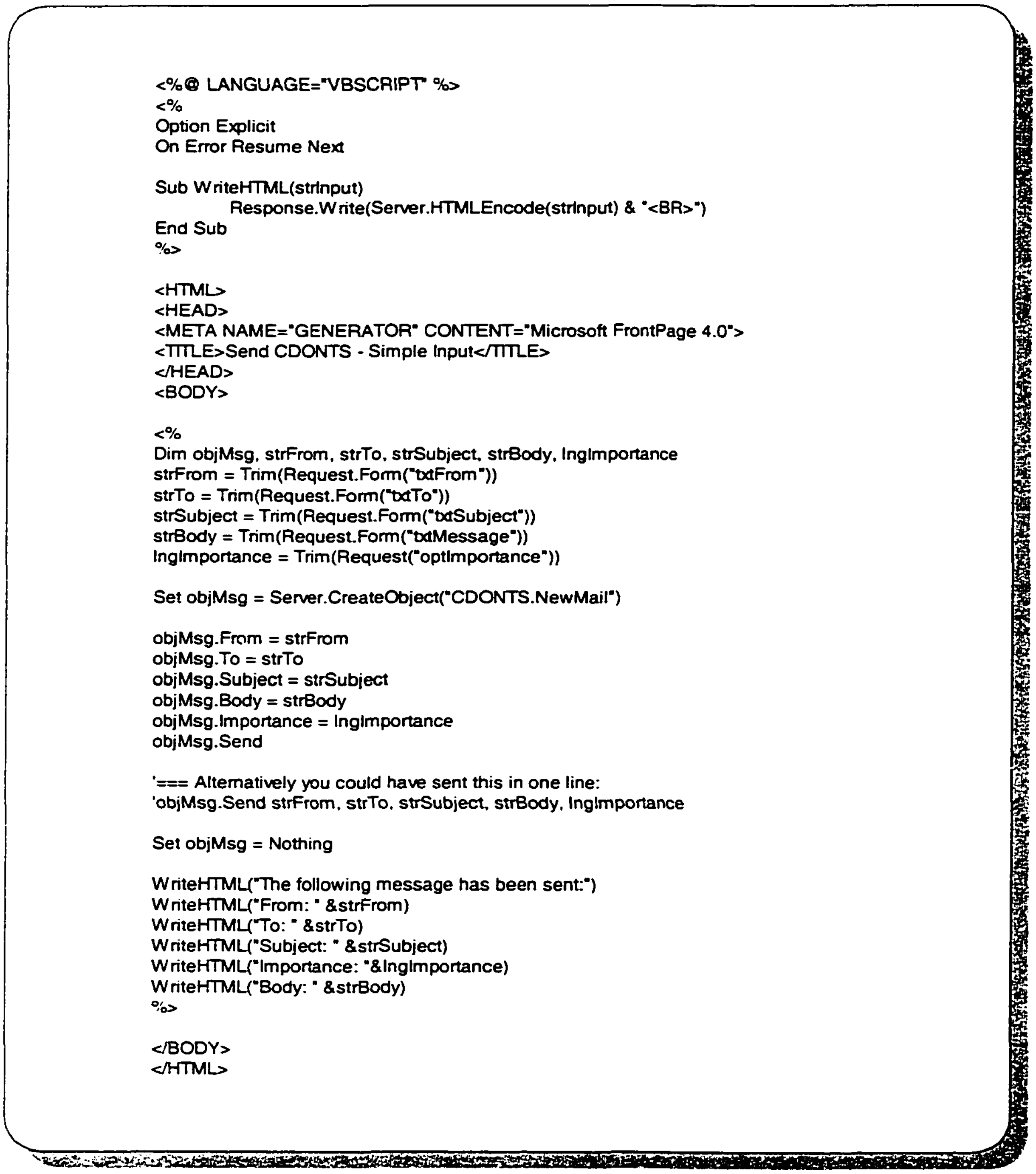


Figure 3.4 Email gateway User Interface



```
<%@ LANGUAGE="VBSCRIPT" %>
<%
Option Explicit
On Error Resume Next

Sub WriteHTML(strInput)
        Response.Write(Server.HTMLEncode(strInput) & "<BR>")
End Sub
%>

<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft FrontPage 4.0">
<TITLE>Send CDONTS - Simple Input</TITLE>
</HEAD>
<BODY>

<%
Dim objMsg, strFrom, strTo, strSubject, strBody, lngImportance
strFrom = Trim(Request.Form("txtFrom"))
strTo = Trim(Request.Form("txtTo"))
strSubject = Trim(Request.Form("txtSubject"))
strBody = Trim(Request.Form("txtMessage"))
lngImportance = Trim(Request("optImportance"))

Set objMsg = Server.CreateObject("CDONTS.NewMail")

objMsg.From = strFrom
objMsg.To = strTo
objMsg.Subject = strSubject
objMsg.Body = strBody
objMsg.Importance = lngImportance
objMsg.Send

'=== Altematively you could have sent this in one line:
'objMsg.Send strFrom, strTo, strSubject, strBody, lngImportance

Set objMsg = Nothing

WriteHTML("The following message has been sent:")
WriteHTML("From: " &strFrom)
WriteHTML("To: " &strTo)
WriteHTML("Subject: " &strSubject)
WriteHTML("Importance: "&lngImportance)
WriteHTML("Body: " &strBody)
%>

</BODY>
</HTML>
```

Figure 3.5 email.asp

3.6 Subsystems

There are three subsystems included in the system. Each will be discussed in details in this section.

3.6.1 The FAQ subsystem

The main purpose of the FAQ subsystem is to provide students access to view the FAQs and help advisors maintain the FAQs. A simple click could let students view all the Computer Science and Computer Engineering FAQs. If the students have their own questions other than the existing ones, they can go through the email gateway, click the button to open the email form and type the questions, and then their questions will be sent to their advisors immediately.

The advisors will use the FAQ system via a different track. They need to login to the system first. After typing their user name and password, the advisors will enter a user interface designed for them. FAQs are displayed in a table as hyperlinked questions and with radio buttons on the left and some other operation buttons at the bottom of the page. If the advisor wants to delete a question, he can click the radio button beside the question, and then click the **delete** button. After confirming his deletion in a confirm page, the question can be deleted and the table will refresh immediately without the deleted question. If the advisor wants to add a new FAQ, click the **add** button. A form to enter a new FAQ will pop up, enter the question and answer, click the **add** button, the question will be added to the **FAQ** table in the database.

Advisor can choose to add some HTML tags for a special format to display later. Updating is evoked by clicking on the hyperlinked question, after which both the question and the answer will display in the form similar to the form that it was added, which makes it easier to modify.

a) File structure

The FAQ subsystem composes of 12 ASP files designed to manipulate different functions of the subsystem. Here is the overall file structure of the subsystem and their basic functions. (Figure 3.6)

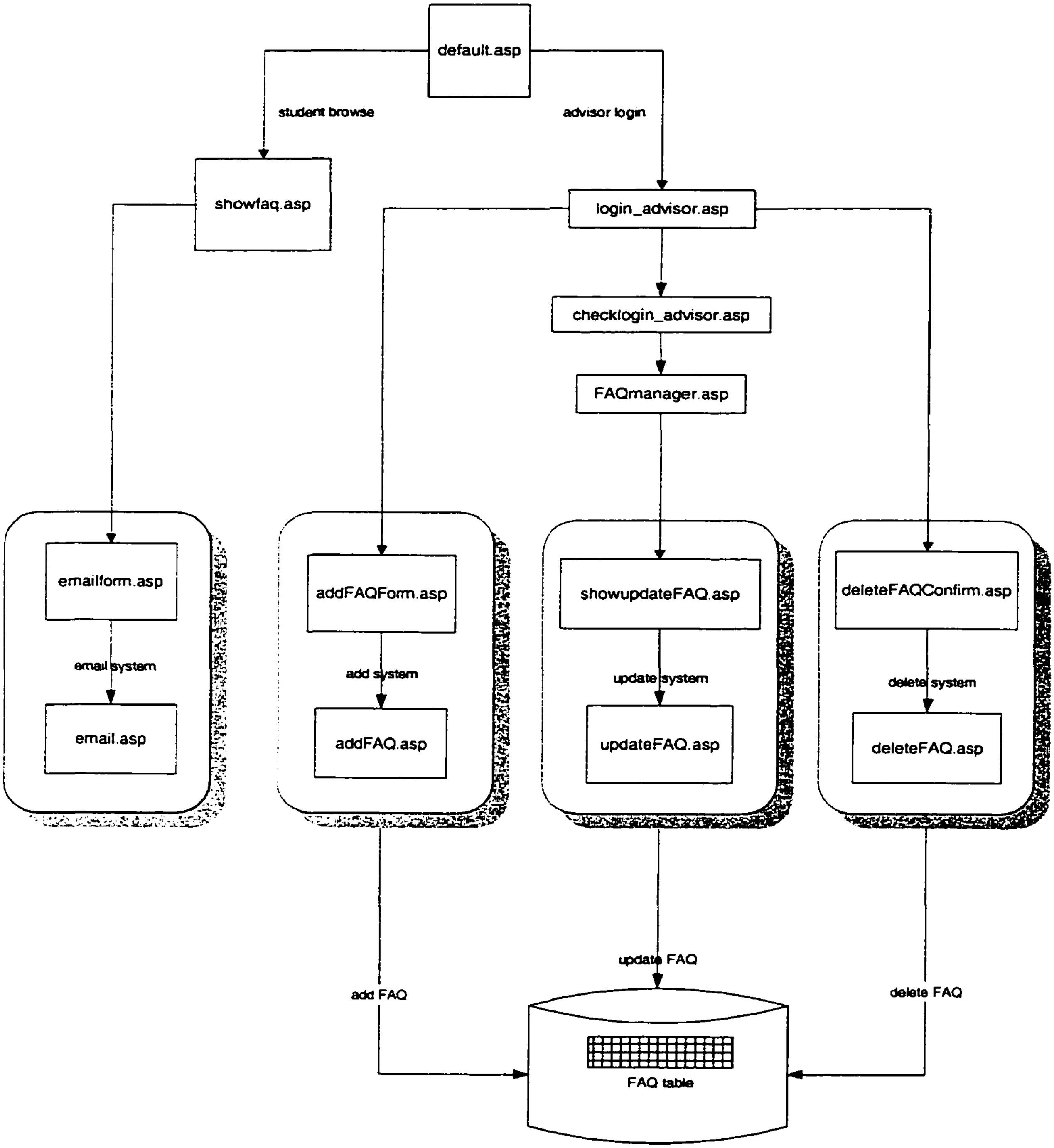


Figure 3.6 FAQ subsystem file structure

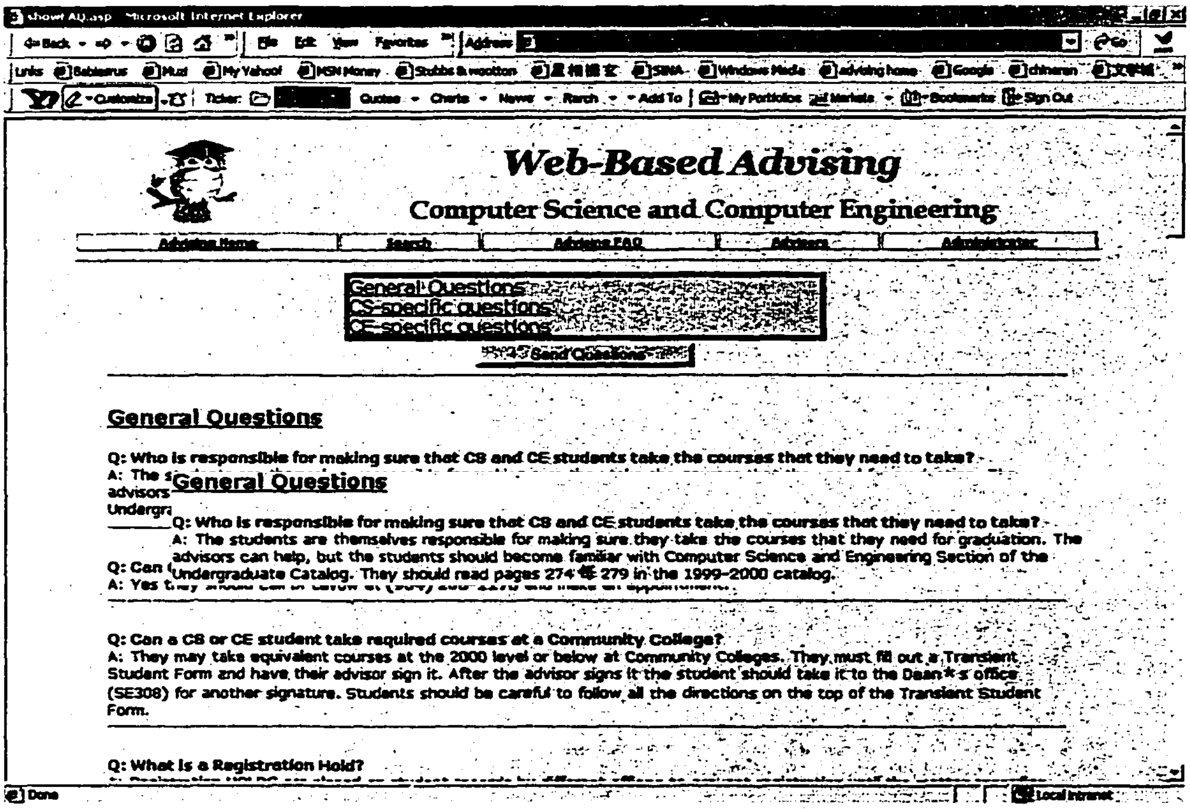
These are the ASP files in the FAQ subsystem:

- Showfaq.asp – student can browse FAQs
- Emailform.asp – user interface to send email to advisor
- Email.asp – email gateway to send email to the advisor
- Login.asp – user interface for advisor to login
- Checklogin.asp – to check username and password and redirect page to FAQmanager.asp
- FAQmanager.asp – user interface to add , update and delete FAQs
- addFAQForm.asp – user interface of a form to add a new FAQ to the database
- addFAQ.asp – to add FAQ to the faq table in the database
- showupdate.asp – user interface for user to change the information of a FAQ
- update.asp – to update a FAQ in database
- deleteConfirm.asp – user interface to let user confirm their deletion
- delete.asp – to delete FAQs from database

b) User Interface for FAQ system

There are several user interfaces included in the FAQ system. Interface for user to send email to advisor, for advisors to login to the system, and for advisors to add, update and delete FAQ. Figure 3.7 shows a page that students can view without the login process, and they can send the advisor an email from this page. Advisors can login under the page shown in Figure 3.8. Another major user interface is created by FAQmanager.asp. As shown in Figure 3.9, the advisor can check the radio buttons to select the questions they want to delete. They also can press the **Add** button to get to the interface for adding a FAQ (Figure 3.10) or hyperlinked to get to the update interface, shown in Figure 3.11.

A: They may take equivalent courses at the 2000 level or below at Community Colleges. They mu
Student Form and have their advisor sign it. After the advisor signs it the student should take it t
(SE308) for another signature. Students should be careful to follow all the directions on the top o
Form.



General Questions

Q: Who is responsible for making sure that CS and CE students take the courses that they need to take?

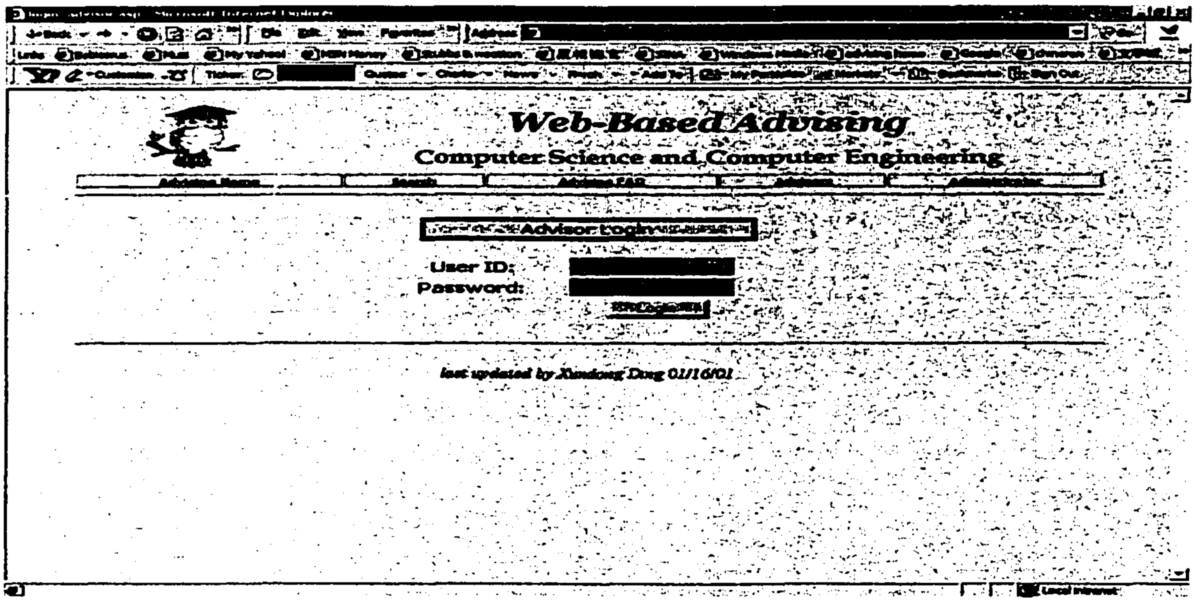
General Questions

Q: Who is responsible for making sure that CS and CE students take the courses that they need to take?
A: The students are themselves responsible for making sure they take the courses that they need for graduation. The advisors can help, but the students should become familiar with Computer Science and Engineering Section of the Undergraduate Catalog. They should read pages 274 – 279 in the 1999-2000 catalog.

Q: Can a CS or CE student take required courses at a Community College?
A: They may take equivalent courses at the 2000 level or below at Community Colleges. They must fill out a Transient Student Form and have their advisor sign it. After the advisor signs it the student should take it to the Dean*s office (SE308) for another signature. Students should be careful to follow all the directions on the top of the Transient Student Form.

Q: What is a Registration Hold?

Figure 3.7 Students' view of FAQ page



Advisor Login

User ID:

Password:

last updated by Xiaodong Ding 01/16/01

Figure 3.8 Advisor login interface

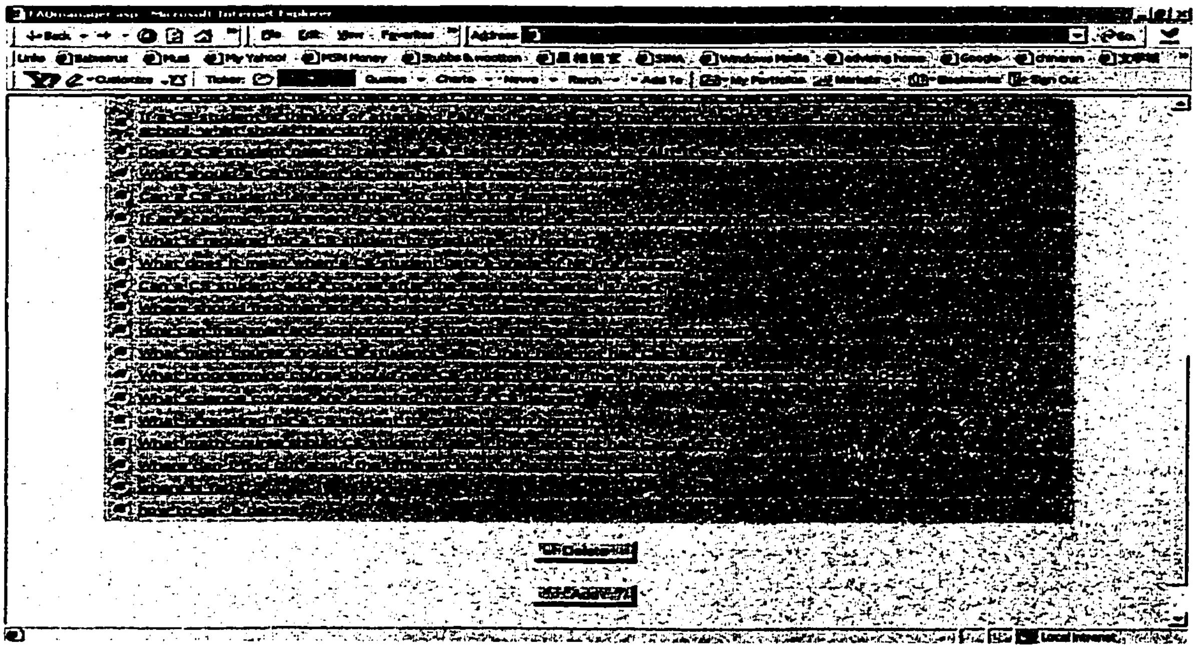


Figure 3.9 FAQ management interface

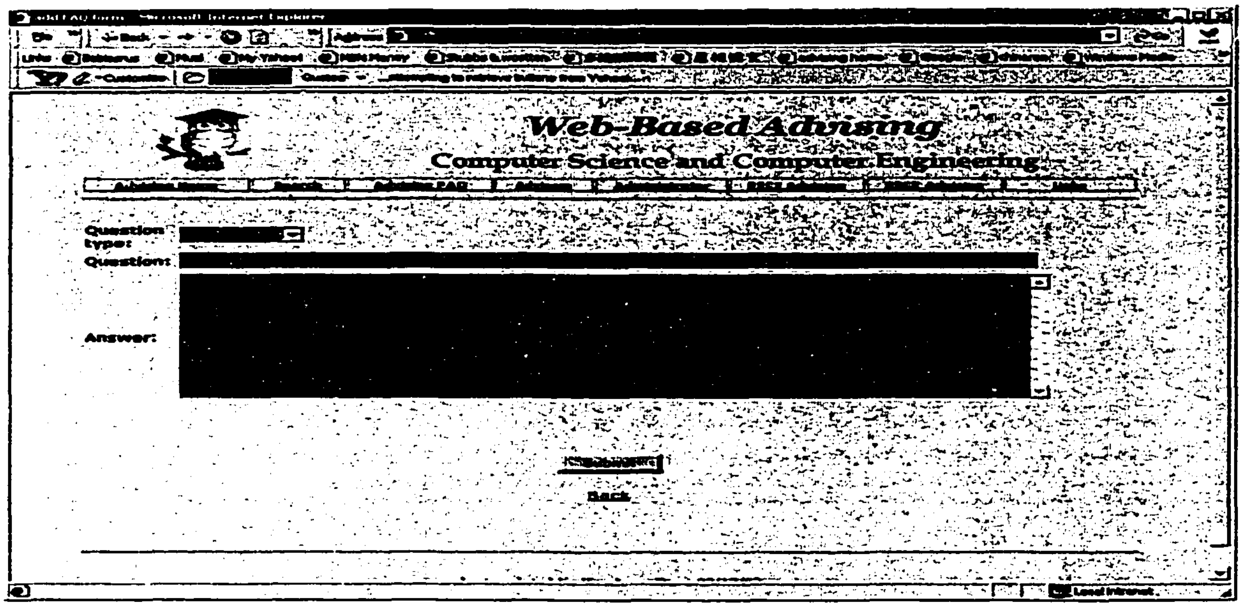


Figure 3.10 GUI in FAQ system to add a FAQ

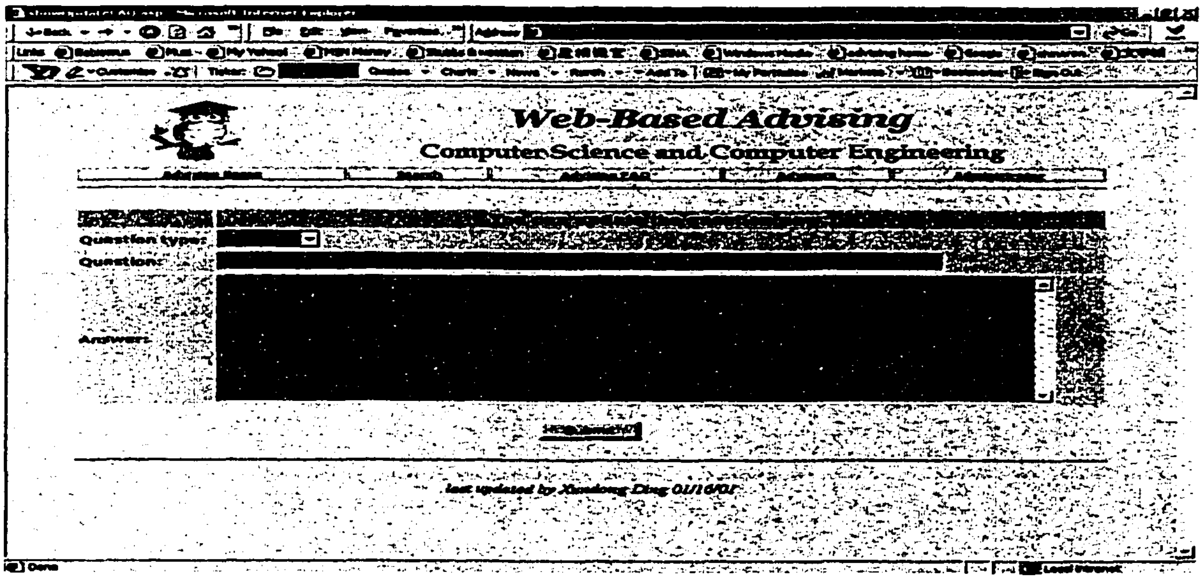

Web-Based Advising
Computer Science and Computer Engineering
Search
Question type:
Question:
Answer:
last updated by Xiaodong Ding 01/16/01
Done
Local intranet

Figure 3.11 GUI in FAQ system to update a FAQ

3.6.2 The Course subsystem

The main function of the Course subsystem is to provide administrators to maintain the course information and the students will use the system indirectly while being advised.

The administrators need to login to the system first. After entering their user names and passwords correctly, the advisors will be presented a user interface designed for the management of the courses. The courses will display in a table with hyperlinked courses and radio buttons on the left. If the advisor wants to delete a course, he can click the radio buttons beside the course, then click the **delete** button. After confirming the deletion in a confirm page, the selected courses can be deleted and the table after the delete operation will refresh immediately. If the advisor want to add a new course, click the **add** button. A form to add a new course will appear, entering the information of a course, click the **add** button, the course will be added to the **CourseInfo** table in the database. Updating process is invoked by click the hyperlinked course IDs or course Names. After clicking, all the course information will show up in the form similar to the form that it was added, which makes it easier to modify.

a) File structure

The FAQ subsystem composes of nine ASP files designed to manipulate different functions of the subsystem. The overall file structure of the subsystem and their basic functions are depicted in the following diagram. (Figure 3.12)

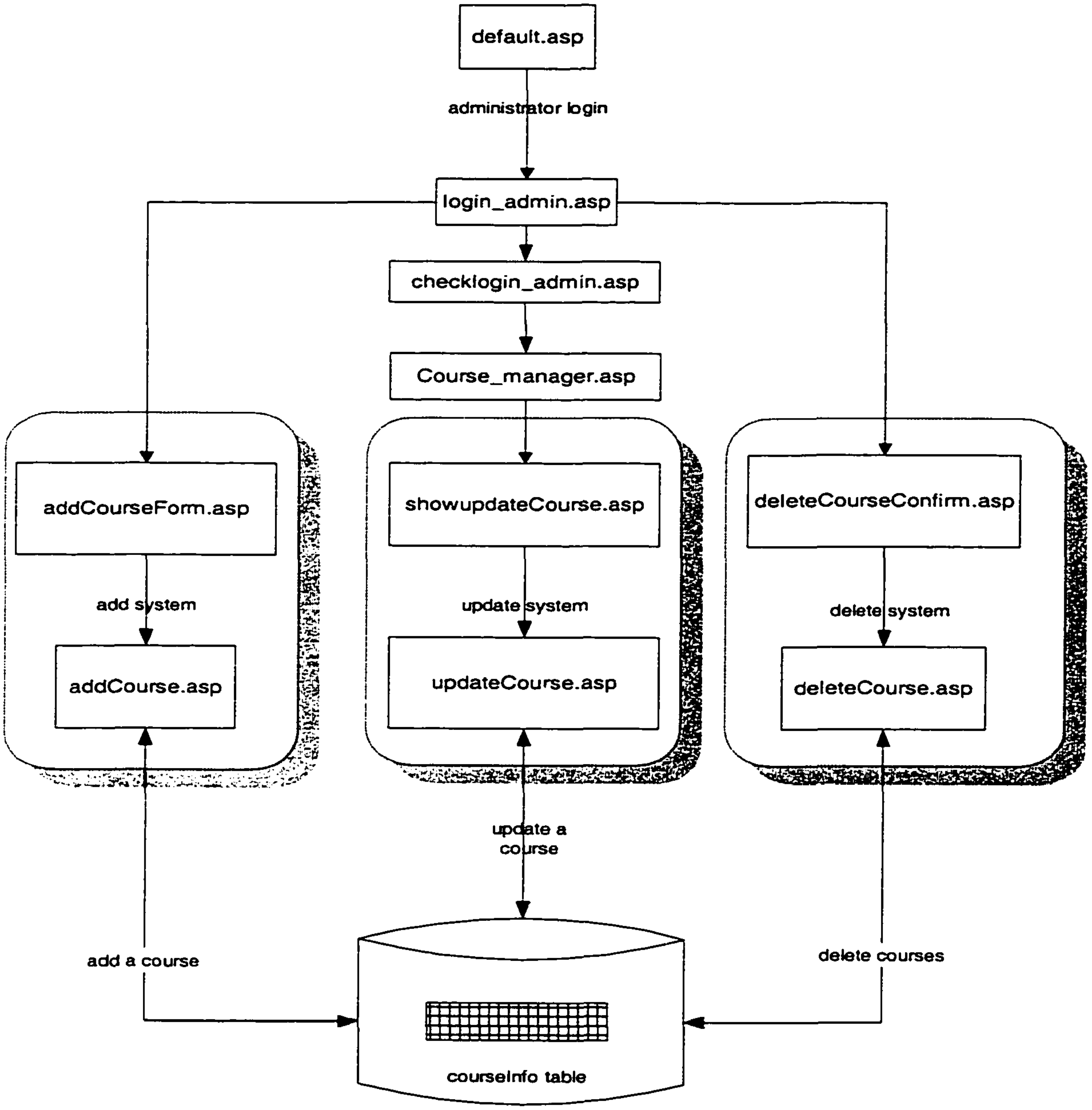


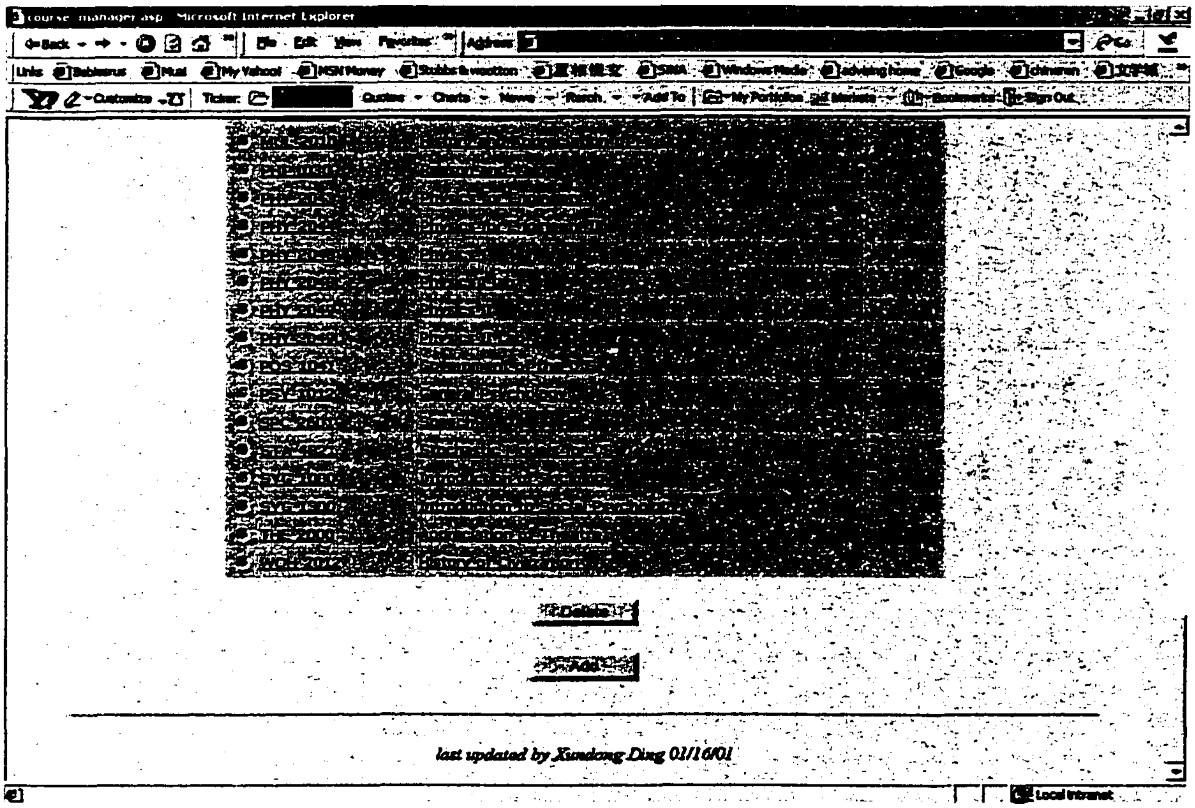
Figure 3.12 Page structure of course subsystem

These are the ASP files in the Course subsystem:

- Login_admin.asp – to show the login interface for administrators.
- Checklogin_admin.asp – to check user name and password, grant access permission or deny access permission.
- Course_manager.asp – to show Course subsystem management GUI with all the courses.
- addCourseForm.asp – to show a form to add a new course.
- addCourse.asp – to add a new course to the database.
- showupdateCourse.asp – to show the form for updating a course information.
- updateCourse.asp – to update a course information in the database.
- deleteCourseConfirm.asp – to let user confirm the deletion of selected courses.
- deleteCourse.asp – to delete a course from the database.

b) User Interface for course subsystem

There are several user Interfaces included in the course subsystem. Interface for administrators to login and for them to add, update and delete a course. One major user interface is created by Course_manager.asp, as shown in Figure 3.13. In this page, Administrator can check the radio buttons to select the course they want to delete. They also can press the Add button to get to the interface for adding a course. Administrators also can click on the course name, which has been hyperlinked to guide them to an interface for updating that particular course.



last updated by Xundong Ding 01/16/01

Figure 3.13 Course management Interface

3.6.3 The Advising subsystem

The advising subsystem is the most important subsystem which takes the major role in the advising process and is the central part of the Web-based advising system. Students under different program will see a different GUI. The GUI provides a form to prompt the student for courses taken. The collected information will be fed into the ASP file for processing, and then final results will be sent back to the student. The kernel part here is the algorithm of how to decide what courses the student needs to take next semester based on the information received from the student and the existing rules stored in the database. A typical graph sorting algorithm is used to help getting the idea and make our own case. This will be discussed in the section later. Let us take a look at the file structure in this subsystem first.

a) File structure

There are 14 ASP files in the Advising subsystem, as shown in Figure 3.14. They are:

- Csmain.asp: computer science advising main page.
- Cemain.asp: computer engineering advising main page.
- CS4Year_form.asp:GUI for Computer Science Four Year degree.
- CsSb_form.asp : GUI for Computer Science Second Bachelor's.
- CsTranfer_form.asp:GUI for Computer Science Transfer student.
- Ce4Year_form.asp:GUI for Computer Engineering Four Year degree.

- CeSb_form.asp:GUI for Computer Engineering Second Bachelors.
- CeTranfer_form.asp: GUI for Computer Engineering Transfer student.
- CS4Year.asp: to process the user selection of Computer Science four-year degree program.
- CsSb.asp: to process user selection of Computer Science second bachelors program.
- CsTransfer.asp: to process the user selection of Computer Science Transfer student program.
- Ce4Year.asp: to process the user selection of Computer Engineering four-year degree program.
- CeSb.asp: to process user selection of Computer Engineering Second bachelors program.
- CeTransfer.asp: to process user selection of Computer Engineering transfer student program.

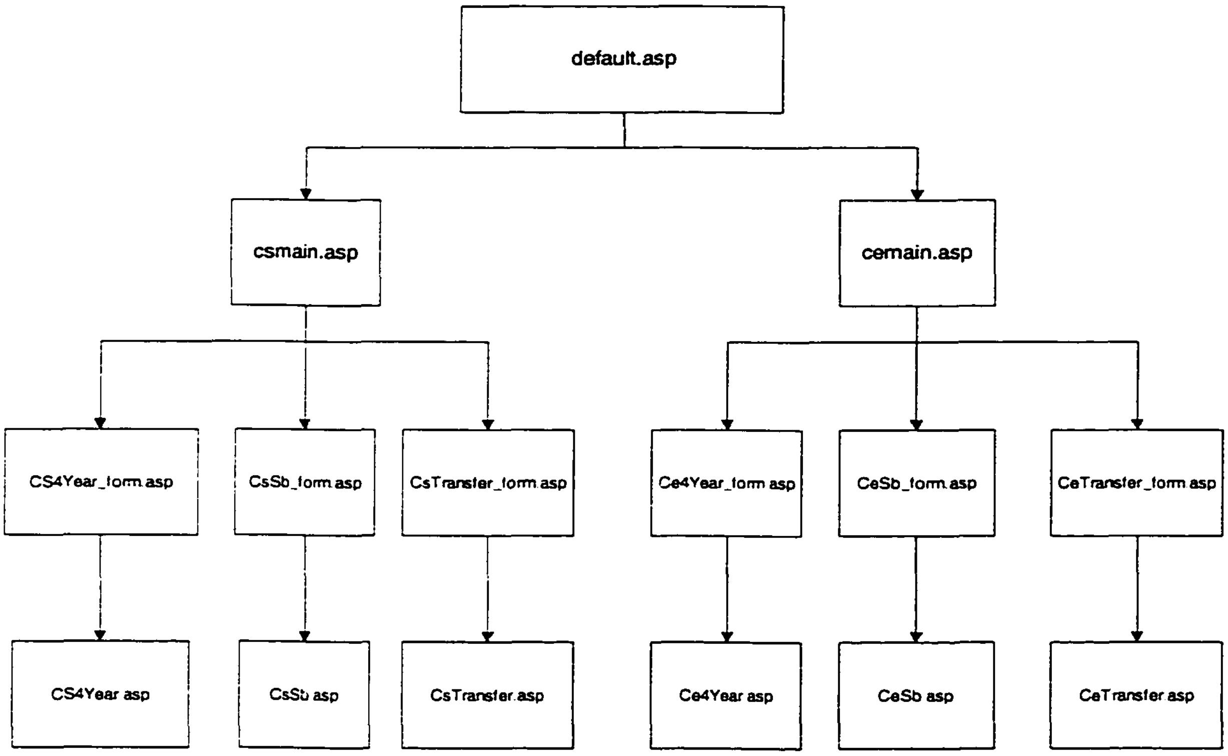
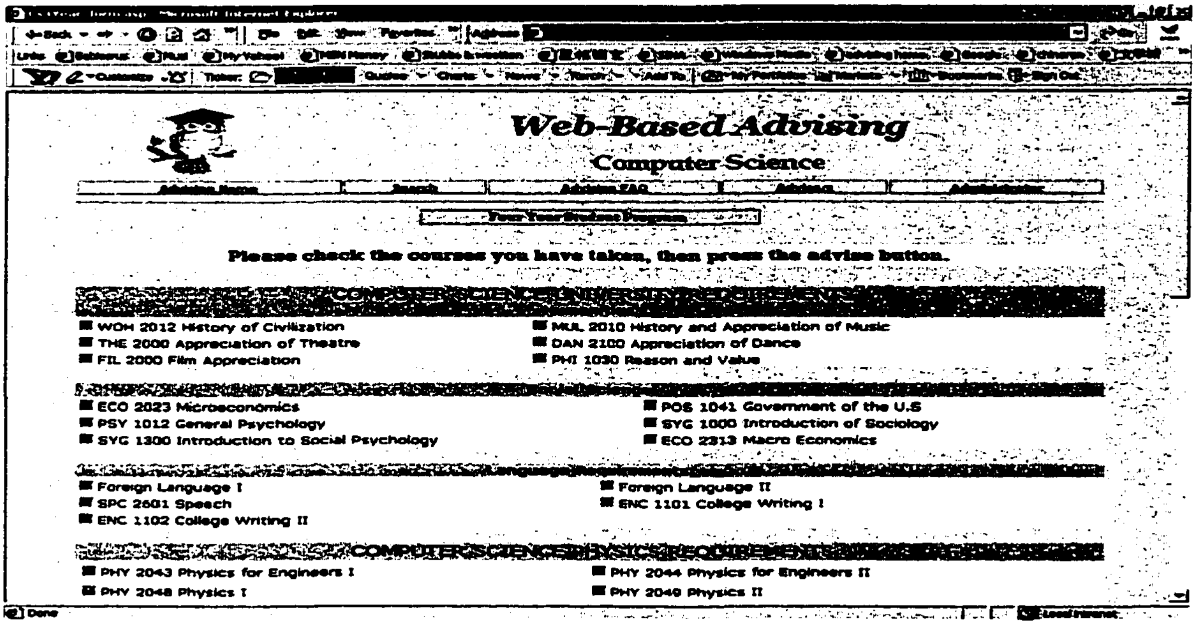


Figure 3.14 Advising subsystem file structure

b) User interfaces

The user interfaces are created via asp files according to six different programs. Two examples are shown in the following screen shot (Figure 3.15, Figure 3.16). They are for Computer Science and Computer Engineering four-year student program. Students can check the check boxes to choose the courses they have taken. Press the *advise* button at the end of the page and get advised.



Please check the courses you have taken, then press the advise button.

COMPUTER SCIENCE UNIVERSITY REQUIREMENTS

■ WOH 2012 History of Civilization
■ THE 2000 Appreciation of Theatre
■ FIL 2000 Film Appreciation
■ MUL 2010 History and Appreciation of Music
■ DAN 2100 Appreciation of Dance
■ PHI 1030 Reason and Value

■ ECO 2023 Microeconomics
■ PSY 1012 General Psychology
■ SYG 1300 Introduction to Social Psychology
■ POS 1041 Government of the U.S
■ SYG 1000 Introduction of Sociology
■ ECO 2313 Macro Economics

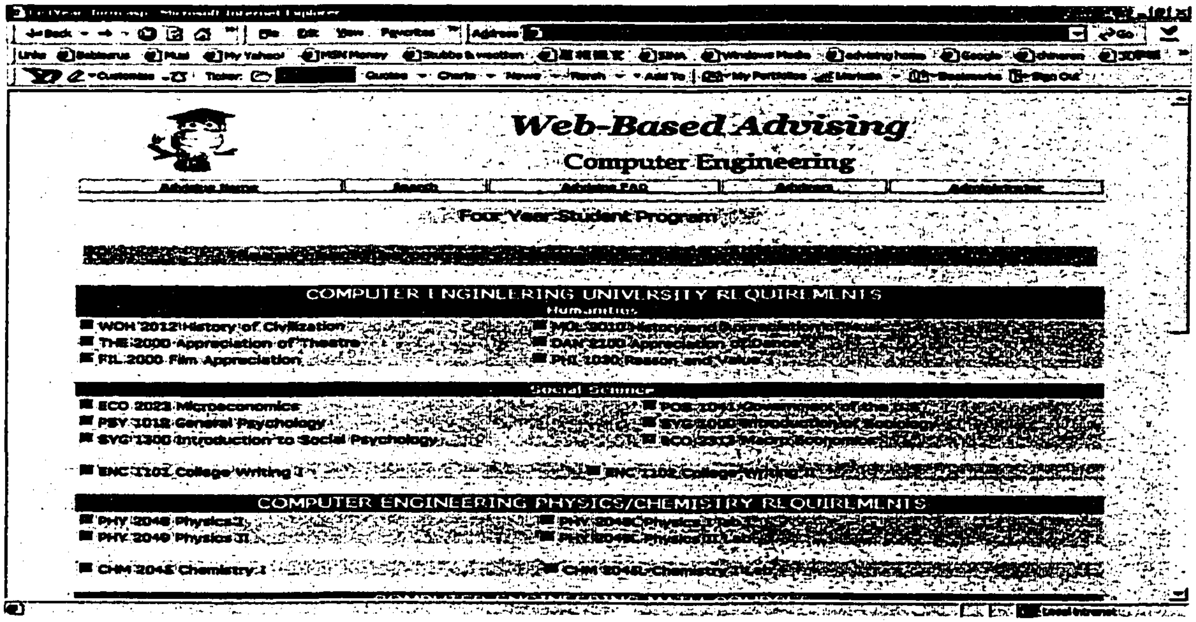
Language Requirements

■ Foreign Language I
■ SPC 2601 Speech
■ ENC 1102 College Writing II
■ Foreign Language II
■ ENC 1101 College Writing I

COMPUTER SCIENCE PHYSICS REQUIREMENTS

■ PHY 2043 Physics for Engineers I
■ PHY 2048 Physics I
■ PHY 2044 Physics for Engineers II
■ PHY 2049 Physics II

Figure 3.15 GUI for Computer Science four-year student



COMPUTER ENGINEERING UNIVERSITY REQUIREMENTS
Humanities

■ WOH 2012 History of Civilization
■ THE 2000 Appreciation of Theatre
■ FIL 2000 Film Appreciation
■ MUL 2010 History and Appreciation of Music
■ DAN 2100 Appreciation of Dance
■ PHI 1030 Reason and Value

Social Science

■ ECO 2023 Microeconomics
■ PSY 1012 General Psychology
■ SYG 1300 Introduction to Social Psychology
■ POS 1041 Government of the U.S
■ SYG 1000 Introduction of Sociology
■ ECO 2313 Macro Economics

■ ENC 1101 College Writing I
■ ENC 1102 College Writing II

COMPUTER ENGINEERING PHYSICS/CHEMISTRY REQUIREMENTS

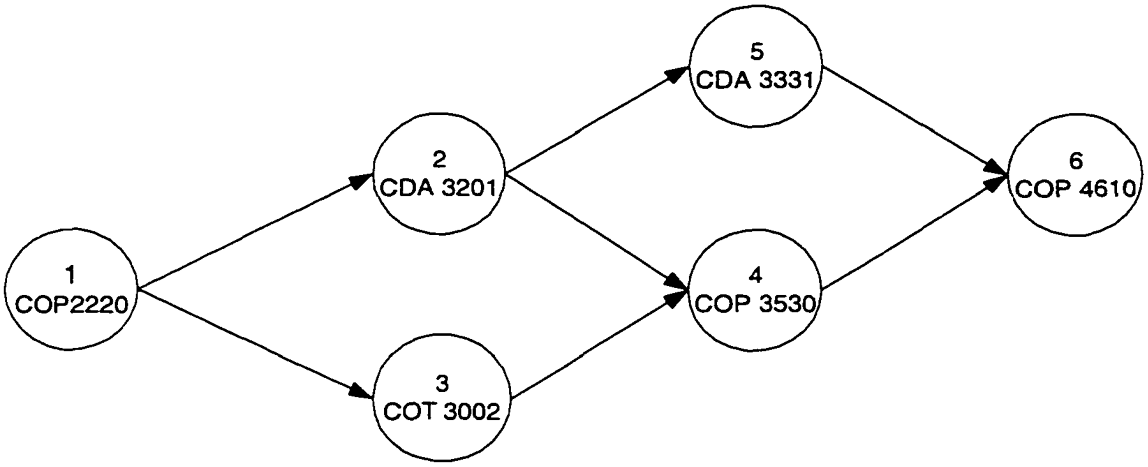
■ PHY 2048 Physics I
■ PHY 2049 Physics II
■ PHY 2048C Physics I Lab
■ PHY 2049C Physics II Lab

■ CHM 2045 Chemistry I
■ CHM 2045L Chemistry I Lab

Figure 3.16 GUI for Computer Engineering four-year student

c) Topological sorting:

Topological sorting [21, 22] is an efficient sorting method for graphs. It is used in the Web-based advising system indirectly as part of the sorting algorithm because of its complex advising scenarios. A topological sort of a graph G = (V,E) is a linear ordering of all its vertices such that if G contains an edge (u, v), then u appears before v in the ordering.



→ represents a perquisite relationship

Figure 3.17 Relationship between core courses

The above diagram is a simple example for the relationship between courses. Prerequisites point to their upper level courses. From this diagram, two tables can be built to be used for the topological sorting and the entire management procedure.

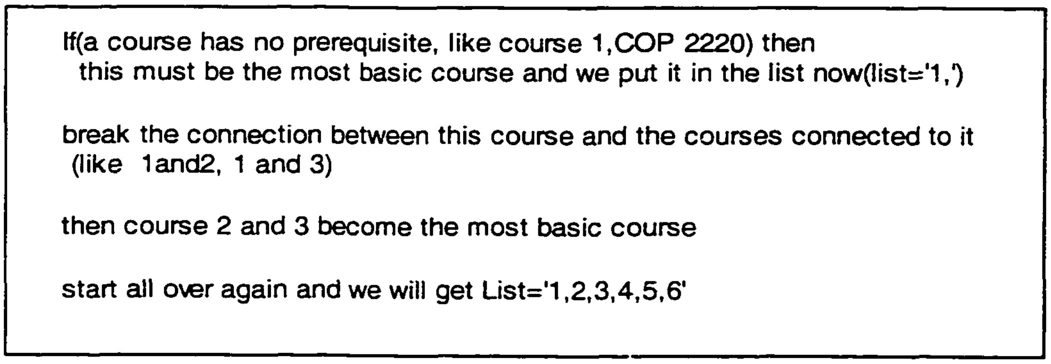
Table 3.1 Course information table

CourseInfo	
Index	CourseID
1	COP 2220
2	CDA 3201
3	COT 3002
4	COP 3530
5	CDA 3331
6	COP 4610

Table 3.2 Course Relationship table

Prerequisites	
BaseCourse	Prerequisites
2	1
3	1
4	2
4	3
5	2
6	5
6	4

Topological Sorting algorithm implementation on this case:



```
If(a course has no prerequisite, like course 1,COP 2220) then
  this must be the most basic course and we put it in the list now(list='1,')

break the connection between this course and the courses connected to it
 (like  1and2, 1 and 3)

then course 2 and 3 become the most basic course

start all over again and we will get List='1,2,3,4,5,6'
```

Figure 3.18 Pseudocode for topological sorting sample

Topological sorting database implementation ASP code and running results are stated in Appendix B.

Topological sorting helps us get the order of all the courses and make it clear the relationship between the course and its prerequisites. It also shows us the method to state the relationship among the courses. We use it primarily in the advising of core courses. Topological sorting alone is not sufficient in this system, because getting the remaining courses that a student need to take is not a task of subtraction from the topological order. We need more algorithms for different rules during the advising session to make the final result.

d) Advising Process

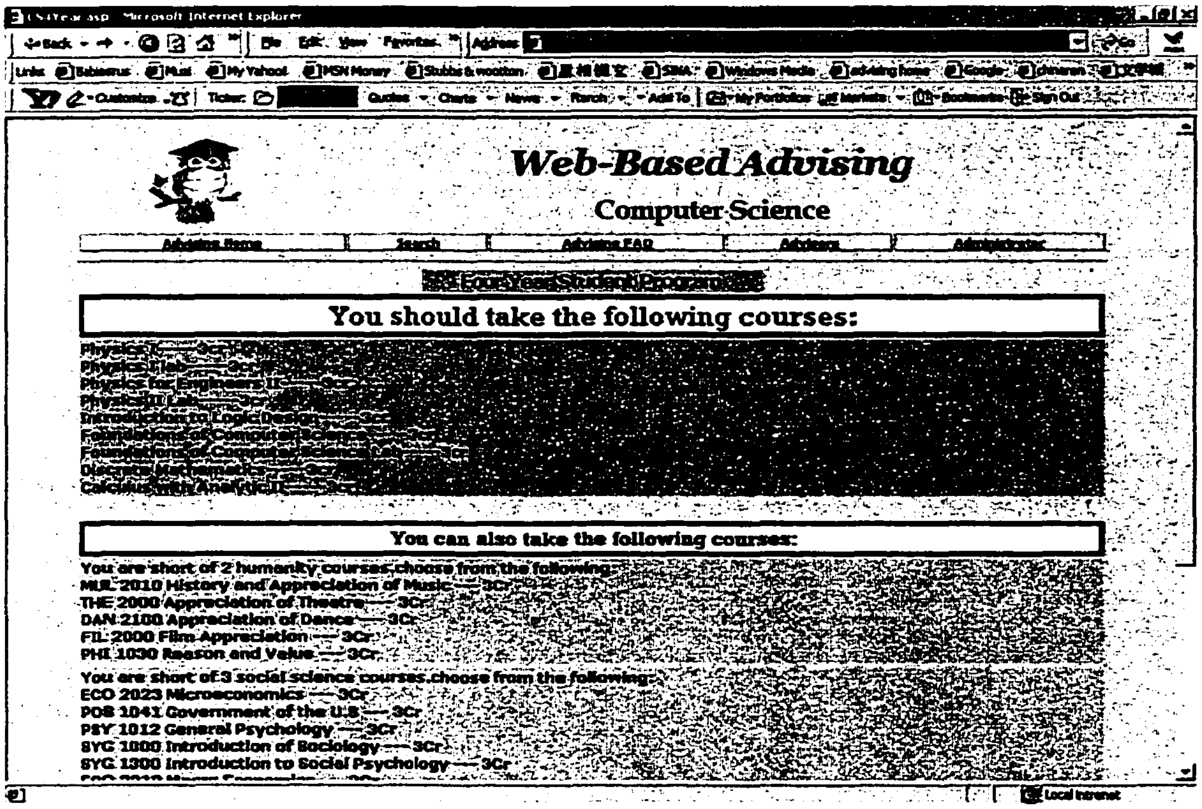
The advising process starts with collecting user input using forms. After getting the data we will feed them into our *choosing machine* which is built on the requirements and rules to get a check. These so called *choosing machines* are the codes to process the user input using different algorithms. The entire advising session starts with the core course advising. Only core courses need to be concerned of prerequisites. The psudocode for core course advising is shown in Figure 3.19. Core course advising is the most complex part of advising. There are quite a few rules involved in this subsection. We need to take the consideration of the *one out of a three-course set case*, the *two out of a three-course set case*, and the prerequisites limitation.

```
                ASP code for core course advising
coreCourse = resquest.form("coreCourse") ;
select * from exclusion where course_taken in 'coreCourse'
If not empty
     repelList1 = repelList & Index
end of

select * from CourseInfo where Index = 'core' and not In
      ('senior status courses') ;

for each Index
   select * from prerequisite where BaseCourse = Index and PrereqCourse
     not in 'coreCourse'

If the collection empty ;
   Select * from exclusion where course_taken  = Index and status <>'locked'
 If not empty
           Repelled_course = rstemp("repelled_course")
            reqelledList2 = repelledList2 & Repelled_course
          List = List & Index ;
 End if
End if
compareList = repelList1 & repelList2 & coreCourse
Take_List = elimination(List, compareList)
```

Figure 3.19 Pseudocode to process core courses

e) Results

After pressing the *advise* button, students can get advised immediately. The following screen shot shows the advice for a four-year Computer Science student. The results for six different scenarios are similar, but results are based on the data in the database and based the situation of each student. The following screen shots captures the advising results for Computer Science four-year student and Computer Engineering four-year student program.



Microsoft Internet Explorer
Back
File Edit View Favorites
Address
Go
Links
Customize
Ticker:
Quotes Charts News Research Add To
My Portfolios Markets Bookmarks Sign Out
Local intranet

Web-Based Advising

Computer Science

CS Four-Year Student Program

You should take the following courses:

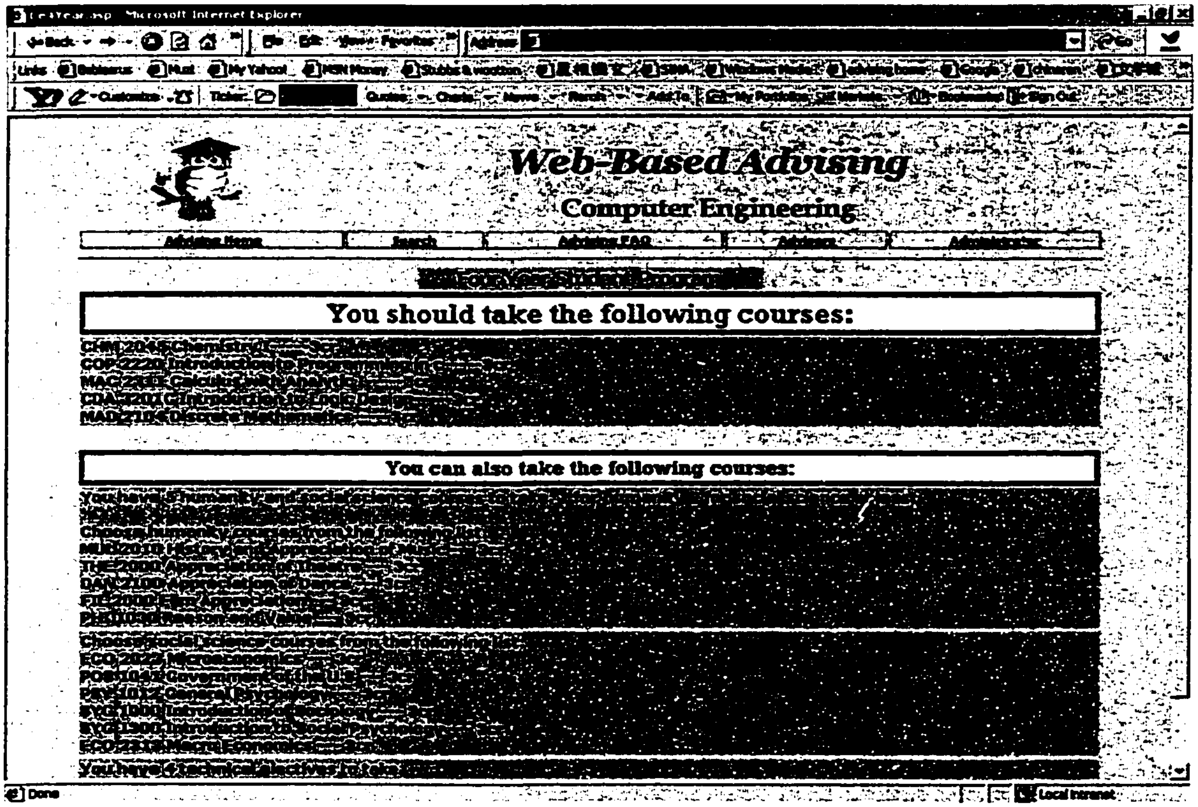
Physics I — 3Cr
Physics I Lab — 3Cr
Physics for Engineers II — 3Cr
Physics II Lab — 3Cr
Introduction to Logic Design — 3Cr
Foundations of Computer Science — 3Cr
Foundations of Computer Science Lab — 3Cr
Discrete Mathematics — 3Cr
Calculus with Analytic II — 3Cr

You can also take the following courses:

You are short of 2 humanity courses,choose from the following:
MUL 2010 History and Appreciation of Music — 3Cr
THE 2000 Appreciation of Theatre — 3Cr
DAN 2100 Appreciation of Dance — 3Cr
FIL 2000 Film Appreciation — 3Cr
PHI 1030 Reason and Value — 3Cr

You are short of 3 social science courses,choose from the following:
ECO 2023 Microeconomics — 3Cr
POS 1041 Government of the U.S — 3Cr
PSY 1012 General Psychology — 3Cr
SYG 1000 Introduction of Sociology — 3Cr
SYG 1300 Introduction to Social Psychology — 3Cr

Figure 3.20 Advising results for Computer Science four-Year student



Web-Based Advising

Computer Engineering

Advising Home | Search | Advising FAQ | Advisors | Administrator

Four Year Student Program

You should take the following courses:

CHM 2045 Chemistry I [illegible]
COP 2220 Introduction to Programming in C [illegible]
MAC 2311 Calculus with Analytic [illegible]
CDA 3201C Introduction to Logic Design [illegible]
MAD 2104 Discrete Mathematics [illegible]

You can also take the following courses:

You have 3 humanity and social science courses to take [illegible]
[illegible]
Choose humanity courses from the following list:
MUH 2011 History and Appreciation of Music [illegible]
THE 2000 Appreciation of Theatre [illegible]
DAN 2100 Appreciation of Dance [illegible]
FIL 2000 Film Appreciation [illegible]
PHI 1010 Reason and Value [illegible]

Choose social science courses from the following list:
ECO 2023 Microeconomics [illegible]
POS 1041 Government of the U.S. [illegible]
PSY 1012 General Psychology [illegible]
SYG 1000 Introduction of Sociology [illegible]
SYG 1300 Introduction to Social Psychology [illegible]
ECO 2013 Macro Economics [illegible]

You have 4 technical electives to take [illegible]

Figure 3.21 Advising results for Computer Engineering four-year student

3.7 Summary

Now the Web-based academic system is in shape after implementation. It helps students get advised remotely. What to discuss next is the benefits and usage of the proposed system and the way to make the system more stable and useful which will be discussed in the following chapter.

Chapter 4

EXPERIMENTS

The Web-based advising project started with the objective to ease the burden of the advisors and give students a flexible choice of getting advised. The initial phase of the development of the Web-based system involved the architectural design and the implementation of the proposed system. A prototype of the system is implemented. It is accessible on-line (URL: http://cidet.cse.fau.edu/ugadvising). This chapter discusses the experiments of the system conducted among the advisors and Undergraduate students and administrators.

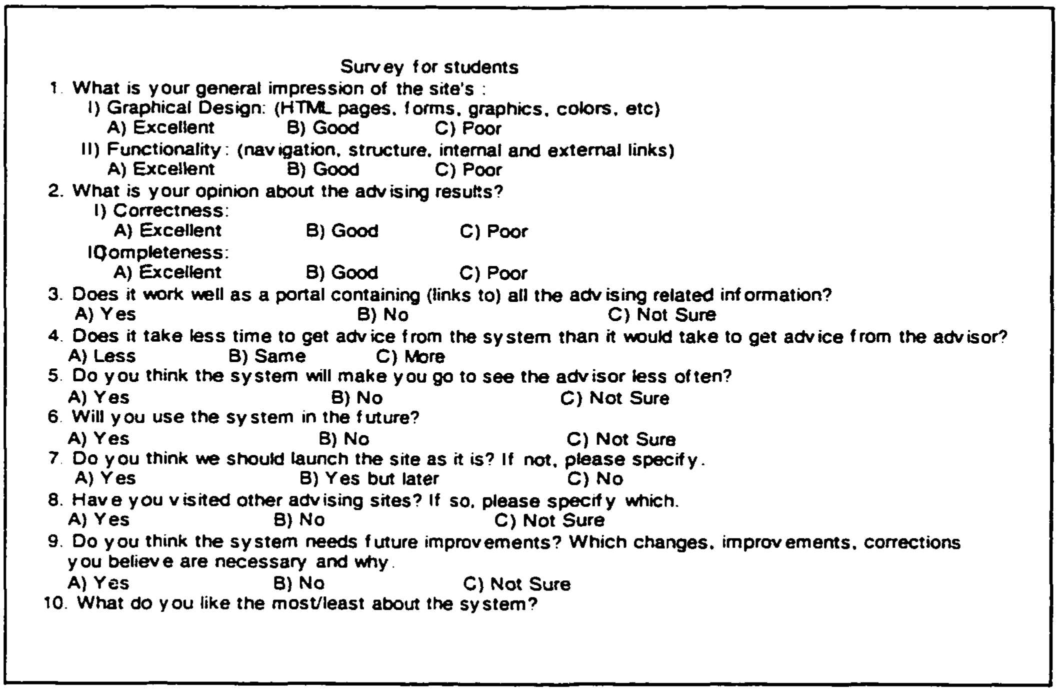
4.1 Introduction

The assessments of the prototype are based on questionnaires designed for both advisors and students. Questions include the graphic design part, the functionality part and future improvement part. The survey for advisors/administrators and the survey for students have some questions in common and some are specific to each group of users.

About twenty-four students, five faculty advisors and one administrator who have used the prototype were asked to answer a set of questions ranging from the graphical design, functionality to usability of the prototype. Their feedbacks are analyzed and tabulated for the assessment purpose of the proposed advising system.

4.2 Survey feedbacks from students

The following questions were used on the survey questionnaire for students.



Survey for students

1. What is your general impression of the site's :
I) Graphical Design: (HTML pages, forms, graphics, colors, etc)
A) Excellent B) Good C) Poor
II) Functionality: (navigation, structure, internal and external links)
A) Excellent B) Good C) Poor
2. What is your opinion about the advising results?
I) Correctness:
A) Excellent B) Good C) Poor
IQompleteness:
A) Excellent B) Good C) Poor
3. Does it work well as a portal containing (links to) all the advising related information?
A) Yes B) No C) Not Sure
4. Does it take less time to get advice from the system than it would take to get advice from the advisor?
A) Less B) Same C) More
5. Do you think the system will make you go to see the advisor less often?
A) Yes B) No C) Not Sure
6. Will you use the system in the future?
A) Yes B) No C) Not Sure
7. Do you think we should launch the site as it is? If not, please specify.
A) Yes B) Yes but later C) No
8. Have you visited other advising sites? If so, please specify which.
A) Yes B) No C) Not Sure
9. Do you think the system needs future improvements? Which changes, improvements, corrections you believe are necessary and why.
A) Yes B) No C) Not Sure
10. What do you like the most/least about the system?

Figure 4.1 Questionnaire for students

The survey results from students are summarized below. In general, the feedbacks are very encouraging. It amounts to a positive recognition of the proposed system.

The answers for each question are summed and tabulated in table 4.1. Note: the table does not include the comment portion that some questions have. It is discussed separately in the later part of this section.

a) The graphical layout, color, and HTML pages have provided eye-pleasing effects. (All the students think the graphical design is great.)

b) The system's functionality is good as to provide students a consistent and effective experience while using the system. (23 out of 24 students give functionality of the system a positive score.)

c) The correctness of the system is good. (23 out of 24 students think the correctness of the system is good.)

d) The system provides complete coverage on advising-related information. (23 out of 24 students think the completeness of the system is good.)

e) The system functions as a portal for undergraduate Computer Science and Computer Engineering students. (20 out of 24 students agree that the site works like a portal.)

f) The system will make students see their advisors less often, therefore less the traffic to advisors' office. (19 out of 24 students respond that they will see their advisors less often in the future.)

g) The system attracts the student users. (20 out of 24 students say that they will use the system in the future.)

h) The system brings advantages into students' academic life in FAU. (21 out of 24 students find it take less time to get advice than from their advisors.)

Table 4.1 Students' survey results

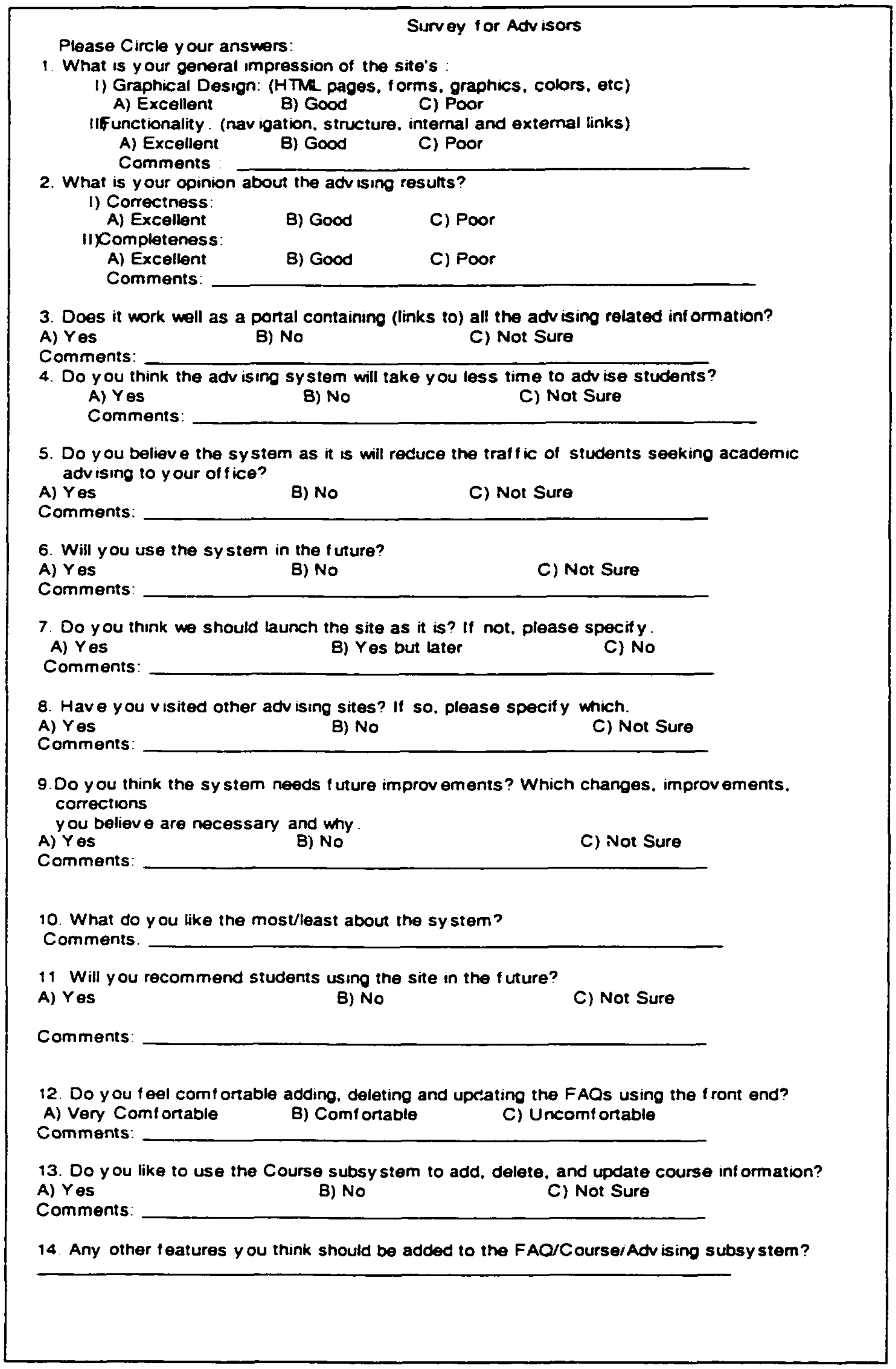
Question	**Choice #1**	**Choice #2**	**Choice #3**
	Excellent	**Good**	**Poor**
1(I)	7	17	
1(II)	5	18	1
2(I)	6	17	1
2(II)	5	18	1
	Yes	**No**	**Not Sure**
3	20	3	1
5	19	1	4
6	20	2	2
8	1	23	
9	11	6	7
	Less	**Same**	**More**
4	21	2	1
	Yes	**Yes but**	**No**
7	13	7	4
10	Plain		

From the above statistical results, we can conclude that the Web-based advising system is a useful and helpful tool for undergraduate Computer Science and Computer Engineering students. It not only provides students advising service and all other advising related information but also helps students getting advised instead of having a long waiting outside advisors' office during the office hours. It is successful so far from the students' point of view.

There are some good suggestions from students. Some students suggest have advisors' office hour included in the system. Although the system make students see their advisors less often, they still need to see their advisors for issues that individually related in the system. Some students are very excited that they can get advised 24 hours a day without waiting in a long line outside the advisor's office. Also some students are very observant to point out some mistakes in the advising process which we have overlooked.

4.3 Survey feedbacks from advisors/administrators

The following questions were used on the survey for advisors. The results are shown in Table 4.2.



Survey for Advisors

Please Circle your answers:

1. What is your general impression of the site's :

I) Graphical Design: (HTML pages, forms, graphics, colors, etc)
A) Excellent B) Good C) Poor

II)Functionality: (navigation, structure, internal and external links)
A) Excellent B) Good C) Poor
Comments : ____________________

2. What is your opinion about the advising results?

I) Correctness:
A) Excellent B) Good C) Poor

II)Completeness:
A) Excellent B) Good C) Poor
Comments: ____________________

3. Does it work well as a portal containing (links to) all the advising related information?
A) Yes B) No C) Not Sure
Comments: ____________________

4. Do you think the advising system will take you less time to advise students?
A) Yes B) No C) Not Sure
Comments: ____________________

5. Do you believe the system as it is will reduce the traffic of students seeking academic advising to your office?
A) Yes B) No C) Not Sure
Comments: ____________________

6. Will you use the system in the future?
A) Yes B) No C) Not Sure
Comments: ____________________

7. Do you think we should launch the site as it is? If not, please specify.
A) Yes B) Yes but later C) No
Comments: ____________________

8. Have you visited other advising sites? If so, please specify which.
A) Yes B) No C) Not Sure
Comments: ____________________

9. Do you think the system needs future improvements? Which changes, improvements, corrections
you believe are necessary and why.
A) Yes B) No C) Not Sure
Comments: ____________________

10. What do you like the most/least about the system?
Comments. ____________________

11 Will you recommend students using the site in the future?
A) Yes B) No C) Not Sure

Comments: ____________________

12. Do you feel comfortable adding, deleting and updating the FAQs using the front end?
A) Very Comfortable B) Comfortable C) Uncomfortable
Comments: ____________________

13. Do you like to use the Course subsystem to add, delete, and update course information?
A) Yes B) No C) Not Sure
Comments: ____________________

14 Any other features you think should be added to the FAQ/Course/Advising subsystem?
____

Figure 4.2 Questionnaire for advisors/administrators

General results are displayed in table 4.2 and summarized as follows:

a) Overall graphical design is favored by all advisors and administrators. (Advisors/administrators rate the graphic design good.)

b) Functionality of the system is good. (4 out of 5 advisors rate the functionality of the system good.)

c) The system provides correct advising results to students. (4 of 5 advisors are satisfied with the correctness of the system.)

d) The system covers a complete advising-related information and tool. (4 out of 5 advisors have no comments on the completeness of the system.)

e) The system functions like a portal containing links to all the advising related information. (All advisors think the site functions like a portal.)

f) The system will take less time to advise the students than advisors will do. (All advisors agree that the system can take less time to advise the students.)

g) The system might reduce the traffic to the advisors' offices. (3 out of 5 advisors agree that the system may reduce the traffic to their offices.)

Table 4.2 Advisors' survey results

Question	**Choice #1**	**Choice #2**	**Choice #3**
	Excellent	**Good**	**Poor**
1(I)		5	
1(II)		4	1
2(I)		4	1
2(II)		4	1
	Yes	**No**	**Not Sure**
3	5		
4	5		
5	3		2
6	5		
8	1	4	
9	3	2	
11	5		
13	4		1
	Yes	**Yes but**	**No**
7	2	3	
	Very	**Comfortable**	**Uncomfortable**
12	5		
10,14	Plain		

Overall, there are encouraging responses of the use of the Web-based Advising system. Feedbacks for the graphic design and time-saving effect for advisors are all positive. Some advisors show great excitement of the project which brings a brand new tool for them. They have some experience visiting other advising Web sites before, but this site is better comparatively. The functionality and completeness is very satisfactory. The Web-based advising system actually **does** the advising and integrates variety of information into the site.

In addition to all the praises and excitement expressed by the advisors, there are suggestions for other functions to be added to make the system more complete and usable.

4.4 Conclusion

From the experiment runs, it can be concluded that the Web-based advising system is a successful project, based on survey feedbacks from students, advisors and administrators. Students and advisors are satisfied with the system and will use it in the future. Comments regarding changes and upgrades have been recommended to make the system more complete and useful. Furthermore, we will take seriously some constructive comments with respect to the future extension of the system for the improvement of the system in the future.

Chapter 5

CONCLUSION

This thesis designs and develops a sophisticated web-based advising system that can be used to conduct on-line academic advising. A survey of existing advising systems including the advantages and disadvantages has helped us in formulating a new Web-based advising system model. A summary of the features of the Web-based advising system including general and unique features is presented in this chapter. It concludes with a list of future enhancements for further development of the Web-based advising system.

5.1 General Features in Web-based academic advising

a) Advise student which courses to take next semester.

b) Input and update general advising-related information.

c) Input and update information on course prerequisites.

d) Input and update information on degree requirements.

e) Input and update course information.

f) Enable student user to ask questions to advisors through an email gateway.

g) Have easy access to relevant related information, such as course description.

5.2 Experiments Results

Experiment is a very important part of the project. The system needs to know what the users really want and what the feedbacks are from them. After a well-designed survey for advisors and undergraduate students, useful information was gathered to make the system more usable and stable. Five professors and around twenty students got involved in this activity and it is really successful. Almost all students stated that they would use the system in the future which makes the system meet its original goal to let students get advised remotely.

5.3 Flexibility of the system

The target audience for this system is the CSE undergraduate student body. Most of the system's functionality will be tailored to the students' needs. Student users will access information that needs frequent updating as new courses are created, prerequisites change, among other reasons. Secretaries and advisors are allowed to perform those changes and updates. So the flexibility of the system becomes more important.

5.3.1 Prerequisite rules

The prerequisite relationship between courses can be expressed using *prerequisite* table, a relational table *CourseInfo* and the *standard* table. If a course changes its prerequisites, however what we need to do is only to change the records in the database, no need to change any code at all.

5.3.2 Rules with number limitation

The rules using number as a standard like Humanity courses, Social Science courses, some electives are expressed by storing the number in the *standard* table, each time these rules are changed, we only need to change the information in the database instead of changing the hard code.

5.3.3 Rule like "two out of three courses"

As the rule of "two out of three courses" can be stored as a record in the *standard* table, if the rule is changed to "three out of five" next time, it is not necessary to change the codes. Therefore minor changes in the tables will make advising procedure complete successfully.

5.4 Maintenance

Maintenance is a critical issue for this project. There are many rules included in this project, six tables in the database, and six different scenarios for advising. For each different course, there is a good amount of information that needs to be maintained, like the type of the course, its subtype, prerequisites and so on. Administrators will feel it difficult to make changes in the code. To improve this situation, the design of a front-end comes in to solve this problem and make maintenance job much easier and dynamic. For the advisors, they can use the front-end to add, delete and update FAQs. For the administrators, they can update course information, rule information using the password protected front-end.

5.5 Total coverage

Web-based advising system covers almost all of the advising tasks according to the survey which was completed before launching the system. The system covers three different degree requirements, four year student program, second bachelor student program and transfer student program. It also covers Computer Science major and Computer Engineering major. The system also has many links for student easily navigate to related sites with advising information. But it has not covered more sophisticated feature like GPA calculation, degree audit and the current system can not check if the courses are available against university offerings.

5.6 Merits and Demerits

a) Merits:

- Easy access to the system from anywhere using web browser.
- Reduces advising traffic to advisor(s), faulty and staff.

- Saves time for all the groups.
- Helps students to become aware of all graduation requirements prior to meeting advisors.
- Helps students to plan their programs, and makes students aware of the university polices and requirements.
- Helps students and advisors to make use of some useful features available in the system like email gateway.

b) Demerits:

- No system is a perfect system, so results should not be taken as final, they should be crosschecked with the advisor.
- Minimizes student-advisor interaction.
- Key decisions can not be taken using the system.
- System does not provide answers to all the questions.
- System does not provide support in special cases like drop outs, incomplete and failures. Need to contact advisors for decision making in special cases.
- System does not take consideration strength and weakness while advising courses.

5.7 Future Extensions

The Web-based advising system now has its starting prototype. We will expand it to a more completed and sophisticated module in the future after adding more functions. (Figure 5.1) for our future model.

a) **GPA calculation** – help calculating student's GPA, not just the general GPA, but for the Computer Science and Computer Engineering only. It can provide information of what grade students need to get for the remaining courses to meet the university requirements.

b) **Import SASS database, provide students access to their own records.** Perhaps the most effective way to provide online advising services is to integrate Web applications with the institution's student information system (SIS) and to allow students access to parts of their individual student record. Access to these interactive services is based on a secure login. These tools enable students to assume responsibility for many routine functions such as recording address changes. They also may permit students to check their account balances or to assess their own progress toward a degree through a degree audit system. In some institutions, advisors also have access to students' records, including the ability to maintain their own password-protected access to records, including the ability to maintain their own password-protected access to advising notes on students.

c) **Access to official schedule of classes**

After students get advised from the system, if the course list that they will take next semester can also tell students which courses are available next semester, it will be more convenient for students.

d) **Security Issues**

Security improvements will be implemented in future model to make the information stored in the database more stable.

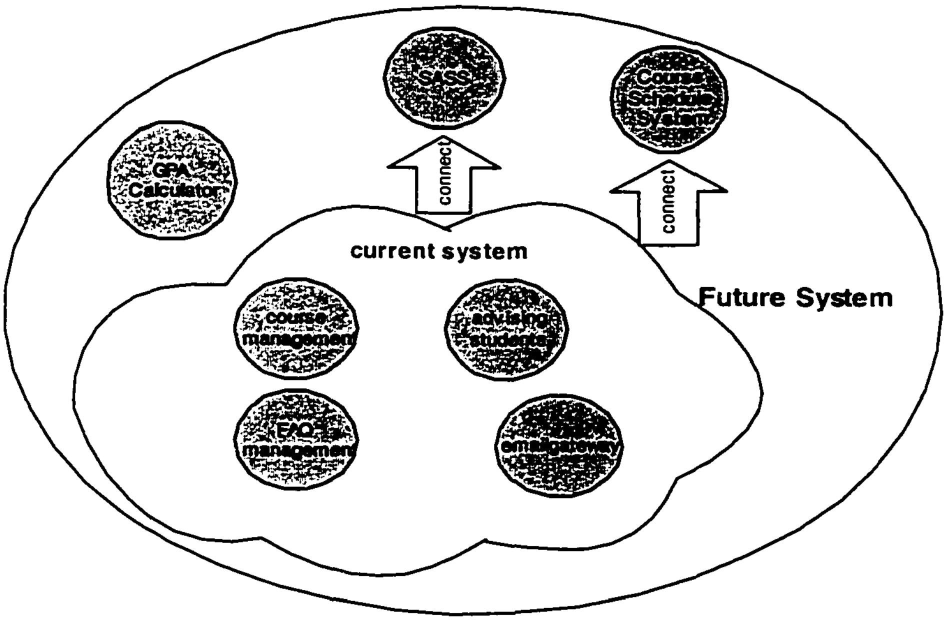


Figure 5.1 Web-based advising system future extension model

APPENDIX A

Tables

Table 1: CourseInfo: This table has 19 fields.

1) **Index**: This is a primary key for this table. This field has to be unique and not empty. Courses are recognized by this field. And Indexes are used for sorting and searching.

2)**CourseID**: A brief course title and numeric designation are stored in this field.(MAC 2311) This is one of the key fields. Input string from user selection is a comma separated course no's.

3)**CourseName**: This field stores a course's full title (Formal Language and Automata).

4) **Credits**: This field indicates credits for a course. This field is not used very much in the current program but a valuable field for future changes.

5)**Prerequisites**: This is an important field storing all prerequisites, separate by a comma each.

6)**CS4Year**: This field shows the course type as to four-year computer science students. One course acts as different course type as to one major.

7)**CS4YearNote**: This field shows some additional information for four-year computer science students.

8)**CSSecondBachelor**: This field shows what is the type of the course as to the Computer Science Second Bachelor degree. One course acts as different course type as to one major.

9)**CSSBNote**: This field shows some additional information for computer science second bachelor students.

10)**CSTransfer** This field shows the course type as to computer science transfer students. One course acts as different course type as to one major.

11)**CSTransferNote**: This field shows some additional information for computer science transfer students.

12)**CE4Year**: This field shows the course type as to four-year computer engineering degree. One course acts as different course type as to one major.

13)**CE4YearNote** This field shows some additional information for computer four-year engineering student.

14)**CESecondBachelor**: This field shows the course type as to computer engineering second bachelor degree. One course acts as different course type as to one major.

15)**CESBNote**: This field shows some additional information for computer engineering second bachelor students.

16)**CETransfer**: This field show the course type as to computer engineering transfer students. One course acts as different course type as to one major.

17)**CETransferNote**: This field shows some additional information for computer engineering transfer students.

18)**Name**: This field stores the name of the administrator who adds a new record to the database.

19)**TimeStamp**: This field shows the time administrator adds the record to the database.

The following figure shows a sample of **CourseInfo** table.

Table 2: Prerequisite: This table has 2 fields.

1)**BaseCourse**: This field is the course Index that has prerequisites.

2) **PrereqCourse**: This field is the Index of a prerequisite course for the base course.

Table 3: standard: This table has 2 fields.

1)**CourseType**: This field is the course type such as core, humanities, social science.

2) **required_no**: This field is the required number of each course type such as Humanities .

Table 4: FAQ: This table has 6 fields.

1)**index**: This field is the question index. It is the primary key of the table.

2) **category**: This field is the category of the question, computer science or computer engineering.

3)**question**: This field is the question.

4)**answer**: This field is the answer of the question, with html tags inserted.

5)**time**: This field is the time the question is added.

6)**advisor**: This field is the name of the advisor who adds the FAQ.

Table 5: advisor_login: This table has 2 fields.

1)**username**: This field is the username of the advisor.

2) **password**: This field is the password of the advisor .

Table 6: administrator_login: This table has 2 fields.

1)**username**: This field is the username of the administrator.

2) **password**: This field is the password of the administrator .

APPENDIX B

Topological sorting Database implementation:

```
count=5
'do until count=0,means there is nothing left for sorting
Do while Not count=0
List=""
'select the courses we need to sort, exclude the non-core courses and the courses that has been sorted and exist in the sortList
mySQL="select * from courseinfo2 where CS4Year in ('core') and Index not in(46," & sortList & ")"
rsCourse.open mySQL
'the following if means if not empty,we still have courses to sort
If Not rsCourse.BOF Then
Do While Not rsCourse.EOF
Index=rsCourse("Index")
    response.write "<br>" &Index
'select the courses that has prerequisites
mySQL1="select * from prerequisite where BaseCourse=" & Index & " and  PrereqCourse not in
```

```
("& sortList & ")"
set rsPrereq=conntemp.execute(mySQL1)
response.write "prerequisites ---"
If Not rsPrereq.BOF Then
  Do While Not rsPrereq.EOF
   Prerequisite=rsPrereq("PrereqCourse")
    response.write Prerequisite & ","
    rsPrereq.movenext
  Loop
  'if this course has no prerequisites, this is the course we want
    Else
    'List is the index numbers put in sortList each round
  List=List & Index &","
  response.write "<br>List is" & List
          end if
 rsCourse.movenext
Loop
else
' there are no records.
```

```
count=0
end if
   'sortList is the final sorting list
sortList=sortList & List
rsCourse.close
Loop
```

Running Result for topological sorting:

If a course has no prerequisites,that means this course is the most basic course among the remaining courses so,this course should be put in the sort list now.

```
36--COP 2220--prerequisites ---
List is36,
37--MAC 2311--prerequisites ---
List is36,37,
39--CDA 3201--prerequisites ---36,
40--COT 3002--prerequisites ---36,
41--MAD 2104--prerequisites ---36,37,53,
42--STA 4821--prerequisites ---37,53,
44--CDA 3331--prerequisites ---39,36,

45--COP 3530--prerequisites ---40,52,41,36,37,53,

47--COP 4610--prerequisites ---45,44,39,36,40,52,36,41,37,53,

48--COP 3540--prerequisites ---45,40,52,41,37,53,36,

49--CEN 4010--prerequisites ---45,40,52,36,41,37,53,

50--COT 4420--prerequisites ---41,45,37,53,40,52,36,40,52,41,37,36,53,

51--COT 4400--prerequisites ---45,40,52,36,41,37,53,

52--COT 3002L--prerequisites ---36,

53--MAC 2312--prerequisites ---37,

39--CDA 3201--prerequisites ---

List is39,

40--COT 3002--prerequisites ---

List is39,40,

41--MAD 2104--prerequisites ---53,

42--STA 4821--prerequisites ---53,

44--CDA 3331--prerequisites ---39,

45--COP 3530--prerequisites ---40,52,41,53,

47--COP 4610--prerequisites ---45,44,39,40,52,41,53,

48--COP 3540--prerequisites ---45,40,52,41,53,
49--CEN 4010--prerequisites ---45,40,52,41,53,
50--COT 4420--prerequisites ---41,45,53,40,52,40,52,41,53,
51--COT 4400--prerequisites ---45,40,52,41,53,
52--COT 3002L--prerequisites ---
List is39,40,52,
53--MAC 2312--prerequisites ---
List is39,40,52,53,
41--MAD 2104--prerequisites ---
List is41,
42--STA 4821--prerequisites ---
List is41,42,
44--CDA 3331--prerequisites ---
List is41,42,44,
45--COP 3530--prerequisites ---41,
47--COP 4610--prerequisites ---45,44,41,
48--COP 3540--prerequisites ---45,41,
49--CEN 4010--prerequisites ---45,41,
50--COT 4420--prerequisites ---41,45,41,
51--COT 4400--prerequisites ---45,41,
45--COP 3530--prerequisites ---

List is45,

47--COP 4610--prerequisites ---45,

48--COP 3540--prerequisites ---45,

49--CEN 4010--prerequisites ---45,

50--COT 4420--prerequisites ---45,

51--COT 4400--prerequisites ---45,

47--COP 4610--prerequisites ---

List is47,

48--COP 3540--prerequisites ---

List is47,48,

49--CEN 4010--prerequisites ---

List is47,48,49,

50--COT 4420--prerequisites ---

List is47,48,49,50,

51--COT 4400--prerequisites ---

List is47,48,49,50,51,

sortList is0,36,37,39,40,52,53,41,42,44,45,47,48,49,50,51,
```

BIBLIOGRAPHY

[1] "Academic Advisement Center", Weber State University

http://departments.weber.edu/aac/

[2] "Academic Advising", North Dakota University

http://www.ndsu.nodak.edu/instruct/mcclean/ag_www/advise.htm#useful

[3] "Academic Programs", Indiana University -- Purdue University Columbus

http://www.columbus.iupui.edu/AcademicPrograms.html

[4] ASSIST(Articulation System Stimulating Interinstitutional Student Transfer) advising system,

http://www.assist.org

[5] "On-line catalog", Washington State University,

http://www.wsu.edu/admissions/academics.html

[6] INSITE (Indiana Student Information Transaction Environment), Indiana University

http://insite.indiana.edu/

[7] Student Planning System -- Brigham Young University,

http://ar.byu.edu/dept_academ_advise/sect3.htm

[8] "Brigham Young University sample annual report", BYU,
http://www.nacada.ksu.edu/Clearinghouse/chouse/awrdwppa.pdf

[9] "PLANNER WEB", California State University, Monterey Bay
http://infoserver.csumb.edu/#planner

[10] "Pre-major academic advising", Duke University,
http://pmac-www.aas.duke.edu/

[11] "Academic Advising Center", Western Washington University,
http://www.acadweb.wwu.edu/advising/gur_humanities.html

[12] "Advising Central", North Carolina University,
http://www.ncsu.edu/advising_central/advisers/preprof.html

[13] "All About Scripting", Microsoft MSDN online library,
http://msdn.microsoft.com/workshop/author/script/allabout.asp

[14] "Degreeworks", Software Research Northwest http://www.srn.com/

[15] "Academic Advising", FACTS (Florida Academic Counseling and Tracking for Students), http://www.facts.org/

[16] Oge Marques, Xundong Ding, and Sam Hsu, "Design and Development of a Web-based Academic Advising System", *2001 ASEE/IEEE Frontiers in Education Conference (FIE '01)*, Session: Web Software, Oct. 10-13, Reno, Nevada.

[17] "Computer Science and Computer Engineering Degree Requirements", FAU, http://www.cse.fau.edu/degree.html

[18] Alex Homer, et al, *Professional Active Server Pages 3.0,* Wrox, September, 1999

[19] C. J Date, *An Introduction to Database Systems,* 7th edition, Addison Wesley Longman, Inc, 2000

[20] J. Klensin, Editor, "Simple Mail Transfer Protocol", RFC 2821, April, 2001.

[21] "Topological sorting ", Microsoft office Solutions, http://www.officevba.com/features/1998/11/vba199811rs_f/vba199811rs_f.asp

[22] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. *Introduction to Algorithms,* The Massachusetts Institute of Technology, 1990