IOWA STATE UNIVERSITY
Digital Repository

Graduate Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

2009

Information management system using 2D barcodes and cell phone technology

Ruchita Sirkanungo
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

Part of the Computer Sciences Commons

Recommended Citation

Sirkanungo, Ruchita, "Information management system using 2D barcodes and cell phone technology" (2009). *Graduate Theses and Dissertations*. 10764.
https://lib.dr.iastate.edu/etd/10764

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Information management system using 2D barcodes and cell phone technology

by

Ruchita Sirkanungo

A thesis submitted to the graduate faculty

in partial fulfillment of requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Johnny Wong, Major Professor
Wensheng Zhang
Simanta Mitra

Iowa State University

Ames, Iowa

2009

Copyright © Ruchita Sirkanungo, 2009. All rights reserved

للاستشارات المنارة

TABLE OF CONTENTS

LIST OF FIGURES .... **III**

LIST OF TABLES .... **IV**

ACKNOWLEDGEMENTS .... **V**

ABSTRACT .... **VI**

CHAPTER 1. INTRODUCTION **1**

1.1 Overview 1
1.2 Motivation 2
1.3 Potential applications 6
1.4 Objectives 8
1.5 Contributions 8
1.6 Road Map 9

CHAPTER 2. RELATED WORK AND EMERGING TECHNOLOGIES **10**

2.1 Technologies 10
2.2 Related Work 15
2.2.1 Barcode technology 15
2.2.2 Barcode and cell phone technology 16
2.2.3 Healthcare technology 18
2.2.4 Cell phone and healthcare technology 20

CHAPTER 3. SYSTEM DESIGN **22**

3.1 System overview 22
3.2 Usage of Barcode and XML technologies in system design 24
3.2.1 Barcode generation and decoding 25
3.2.2 Barcode contents 27
3.2.3 Use of XML in server response generation 35
3.3 System components description 37
3.3.1 Mobile based sub-system 38
3.3.2 External server based sub-system 47

CHAPTER 4. IMPLMENTATION AND EVALUATION **49**

4.1 Tools used and technical specifications 49
4.2 Implementation 50
4.3 System evaluation 61

CHAPTER 5. CONCLUSION AND FUTURE WORK **68**

5.1 Conclusion 68
5.2 Future work 69

BIBLIOGRAPHY **71**

APPENDIX **75**

XMLSchema 75
FoodProductXMLSchema 75

LIST OF FIGURES

Figure 1: Encoding format 33

Figure 2: Sample XML format file 34

Figure 3: XML records of user and allergies 35

Figure 4: XML records of conflicting ingredients 36

Figure 5: XML record of product information...37

Figure 6: System Architecture 38

Figure 7: Decoding of information and assigning content type 51

Figure 8: Class diagram representing the subclasses of ComplexItemParser.........................53

Figure 9: Classification, parsing and display of information 55

Figure 10: Class diagram representing the interface Content 56

Figure 11: Class diagram showing relationship among the classes for conflict checking...58

Figure 12: Class diagram showing relationships of XMLParser 59

Figure 13: Entity Relationship diagram for home database.....................................60

Figure 14: 1D barcode images captured from camera 62

Figure 15: Difference among the 2D barcode images 63

LIST OF TABLES

Table 1: Effect on decoding with the change in image size 63

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude towards all those who have supported me in various ways throughout my research work and writing of the thesis. First, I would like to thank Dr. Johnny S. Wong for his constant support, outstanding guidance, encouragement and patience throughout this research. His valuable suggestions, clear vision and immense knowledge have highly inspired me in conducting this research.

I would also like to thank my committee members Dr. Simanta Mitra for his insightful comments and discussions during our research meeting and Dr. Wensheng Zhang for his appreciation of the research work and consent to be on my program of study committee. I would also like to thank Dr. Hen-I Yang for all his valuable inputs, constructive criticism and moral support in accomplishing my work.

I extremely enjoyed working with all the members of Smart Home research group: Ryan Babbitt, Alamo Reyes Jose, Fuchao Zhou, Fred Philip Stanley and Tanmoy Sarkar. The discussions with them in the meetings and their creative insights have helped me to remain enthusiastic and positive about my research.

I would like to offer my regards to Linda Dutton for her encouragement and help all throughout my studies as a graduate student in the department of Computer Science. I would also like to thank Cindy Marquardt, Maria-Nera Davis for their administrative support and being friendly to me. Finally, I would like to offer my special thanks to my parents and maternal grandparents for their love and moral support in accomplishment of my career goals.

ABSTRACT

One of the challenging problems of pervasive computing is to link a physical object with digital information because many of the pervasive computing applications require manual inputs or complex image processing to obtain information related to a real object. The use of 2D barcodes eliminates such excess processing to acquire the needed information. The 2D barcodes have high capacity to store data, are less prone to human input error and act as a tool to acquire information on site without network access. The currently available solutions use 1D barcodes to represent dynamic information residing in a database and use 2D barcodes to represent only static information that also encode only URLs. In all such applications, the source of information gets restricted to either a database or the static data encoded inside a 2D barcode. None of such solutions takes advantage of 2D barcode capabilities to collect information from different sources and attach it to the real world entity. Moreover, a 2D barcode can also represent and categorize complex text information.

Our approach integrates the capabilities of 1D barcode into 2D barcode to represent and classify the complex digital information collected from different sources. We design and implement an information management system on a handheld device that has image processing and barcode decoding capabilities to address the above-mentioned problem. Our prototype provides a generic framework to decode either 1D or 2D barcode, parse the complex information (both dynamic and static) inside the 2D barcode, differentiate the complex information based on content types and classify the image based on the barcode format. It also assists users in decision-making and information analysis. An example system application can be deployed in grocery stores as a part of the enterprise information management system.

CHAPTER 1. INTRODUCTION

In this section, we describe the overview of technological advancement in mobile applications and barcode technology, motivation behind our work, applications of the system, objectives and contribution.

1.1 Overview

Due to recent technological development and integration of mobile and barcode technologies, several new mobile applications are developed every now and then. Nowadays various barcodes are used in mobile applications other than the traditional commercial systems. Digital barcodes provide a great means to store information in a portable way. Barcodes represent machine-readable information that can be easily stored, transferred and processed. Barcodes have the capability to improve the productivity and reliability of nearly all applications as they are printed and processed by machines. They are processed faster than human data entry and have a higher degree of accuracy. Barcodes have many applications and are mainly used for product identification, inventory marking, shipping container marking, and much more. Barcodes not only provide a simple and inexpensive method to present diverse commercial data, but also improve mobile user experience by reducing their manual inputs [1].

The barcode technology has been further developed with the creation of 2D barcodes to increase the data capacity of 1D barcodes. With the integration of cameras, mobile phones act as scanners, barcode readers and portable data storages and maintaining network connectivity. When used together with such camera phones, 2D-barcodes work as a tag to connect the digital and physical world [3]. Today most of the mobile applications encode a URL or a website address inside a 2D barcode to visit a web page containing a video clip or a document that can be accessed from a mobile web browser. A 2D barcode can also represent a business card, an advertisement coupon, product information and it can be used in visual cryptography. Since it can encode a variety of text information, several mobile centric systems can be developed to manage information belonging to different domains including manufacturing industry, government organizations and public sector, for accurate asset tracking, healthcare, system logistics etc.

1.2 Motivation

Over the years, many applications are developed in home automation area assisting especially elderly and people with special needs in their daily life. One of the primary concerns for elderly is how technology can assist them in buying food items and medicines as well as notify them about what to take. Moreover, the information must be easily accessible anytime from anywhere. Generally, cell phone is the best platform to create applications that can handle such kind of information. Since cell phone is portable and easy to handle, almost everyone carries it and thus can be easily used by elderly people. This opens a new doorway of applications, which focuses on providing healthcare related information on cell phone [4], [5].

Since barcodes can carry a lot of information either by referring to the source of information or by encoding all the information in it, they are used in several applications like point of sale, inventory control, shipping, packaging and data collection. Medical and dental practices rely on complex patient forms and barcodes can help in entering the detailed information in the computer, making it an easy task of gathering large amount of information. It also reduces data collection costs and better services are the results. Barcodes are primarily used for entering data into a system, efficiently and reliably, thereby improving the productivity [6]. Nowadays 2D barcodes can embed a lot of data which can be used to provide useful information. 1D barcode does not provide such static information; rather it refers to an external database to collect the information. However, 2D barcode contains information, which cannot be updated but the information referred by 1D barcode can be easily modified. Therefore, if we use barcodes for carrying information we should come up with a way to get the benefits of both the 1D and 2D barcodes.

RFID is another technology that can be used for information management. However using RFID is not cost effective especially for identifying the different products in the market and in the applications that can assist in buying products, sharing personal information, directing and guiding people on their way to some place located geographically distant or near, performing surveys, filling online forms etc. In these cases, a barcode can be easily scanned because of the use of camera and the barcode scanning applications on the cell phones [1], [7], [8], [9], whereas RFID requires a RFID reader to read the information. Most of the 2D barcodes encode URL, which can further lead to a web page or website. Moreover, a 2D barcode can encode much more information than a URL and can contain large amount of text information. Therefore, there should be some applications that can encode different kinds of information inside a 2D barcode and use it for different purposes.

Currently, there is no application, which can take the benefits of 1D and 2D barcodes to manage information not only related to medicine or food but about anything and any domain and categorizes it accordingly. In addition, such system can provide a way to assist the cell phone user including elderly in collecting and viewing the information, provides functionalities according to the type of information, which would be of great public use.

There is definitely a need to create an application that can manage not only health and food product related information, but can also assist in managing information related to our daily activities. Now the problem is how we can create such a system that is easily available to everyone and can provide such kind of information in the best possible way.

In order to address the aforementioned concerns and the related problem, we propose to design and implement an information management system using 1D and 2D barcodes and cell phone technologies. We build an information management system consisting of a cell phone based subsystem as well as some external subsystems (which may be required depending on the functions the user has to perform). The cell-based component can access information encoded inside the barcodes and can refer to the external sources for more information. This component can decode 1D and 2D barcodes and parse the information gathered from all the sources collectively to present to the user according to the type of information and user choice.

The information management system is developed on Google Android platform suited for Google cell phones (Gphones). It is built in Java as the Android platform supports Java-programming language. It also consists of external servers and databases to which the Android applications run.

للاستشارات المنارة

We will describe below the proposed technique in detail and the way it solves the problem:

1. The system provides a generic way to access any kind of text information encoded inside 1D and 2D barcodes. It also encodes 1D data inside a 2D barcode therefore taking the benefits of both kinds of barcodes.

2. It differentiates the information encoded inside the barcode depending on its content type. Since the information encoded in 2D can be any text not necessarily a URL, this technique can categorize the information so that the user would know the type of information. For example, a 2D barcode can contain a calendar event; by decoding it, the user can see the content type. He can decide whether he wants to see the complete information related to it or not.

3. It provides different functionalities to the cell phone user depending on the content type of information. For instance if the barcode contains contact information the application will provide the option to save the data in contact book but if it contains a URL, it will ask the user to visit the web page.

4. It can collect information not only from barcodes but also from external sources and can process all the information to generate meaningful result that can further help the user in information analysis and decision making. For example, it can help elderly at the time of grocery shopping in deciding whether to buy a particular food item or not, after scanning the barcode on the product based on this technology.

5. It is a portable, easy to use system and may not even require access to Internet if used in a certain way. Since it is user friendly, anyone can use it to access and comprehend the needed information.

6. The information, visible to the user remains private and confidential as long as the user puts restrictions on the usage of his cell phone. Unless the user does not give his cell phone to others and let them scan the barcodes, there is no way to access the information inside the barcode as well as from the external sources, which need user authentication to access the data and only the authorized user can know about it.

1.3 Potential applications

There are many uses of such system, out of which some of them are implemented as functionalities provided by the system and some can be added to the existing system:

1. One of the most important applications of the system is in grocery shopping. It can assist people in making decision about what to buy and what not. When a user goes to a grocery shop, he can scan the 2D barcode on the grocery item and store it in the cell phone. We implemented the above functionality by allowing the user to store the images of the barcodes in the sdcard. He can scan more than one barcode, store it and then use this application to decode and parse all the images stored. The user will then get a list of all the barcodes that he scanned. He can click on one of the barcodes to see the information available from the barcode. For example, it will list the ingredients of the food item and then he can check whether he is allergic to such item or not. He can also check if any of those is in conflict with other food items that can lead to

some ailments or disease. Moreover, the user can also access the current price at the grocery store other than the price mentioned in the 2D barcode. He can also calculate the total price of all the items he puts in the cart before buying so that he would know the total cost before going to the billing counter. He can remove some of the items in case if the total price exceeds his budget.

2. Another possible usage of the application can be in buying medicines. Currently the system does not implement this but it can be added to the system without a lot of change in the code.

3. Another application is in saving the contacts to the address book of cell phone and knowing the image formats of all the images that are decoded.

4. Other applications include calling the phone number if the information encoded inside the barcode shows telephone number, asking the user to generate a calendar event to notify if the encoded information is of calendar event type.

5. Another benefit is that one can think of some more applications and can add them as a part of the overall system. Especially those applications can be easily added which may not require Internet access and can embed 1D data inside 2D barcode just as we did in grocery shopping application. Moreover, one can also encode more complex text information inside the barcode and use the system for decoding it. Since we also propose the way to encode such information in this project, one can easily use this mechanism to encode more and more complex information in a certain format.

1.4 Objectives

The system is designed with the following objectives:

1. To show what kind of information a 2D barcode can encode and how that information can be put to use in the application context.

2. To illustrate how one can make several applications centered on 2D and 1D barcodes and can assist people in making decisions, information analyses and data comparisons.

3. To demonstrate how a 2D barcode can be used to encode information of any product

4. To show the importance of using 2D barcodes for encoding not only food item relevant information but also any product related information in a broad sense and the possible applications based on the usage.

1.5 Contributions

The main contributions of the system are:

1. Existing solutions do not integrate the capabilities of 1D and 2D barcodes, nor take advantage of all the capabilities like high data density, on site data collection, no manual input entry of a 2D barcode
2. Our solution takes advantage of 2D barcodes to handle not only simple but also complex text information by defining its format that can be used for encoding, differentiating the information based on its content type
3. Our technique presents a method to handle information from various sources, process the information to present meaningful result for decision-making on cell phone platform which is portable and compact

1.6 Road Map

The rest of this thesis is organized as follows. Chapter 2 details some of the related work and the technologies used. In Chapter 3, design of our system is explained and illustrated. Chapter 4 describes the implementation details, tools used and system evaluation. Chapter 5 concludes the thesis and discusses future work.

CHAPTER 2. RELATED WORK AND EMERGING TECHNOLOGIES

This chapter discusses the technologies used in designing the system. We also describe the literature survey for the topics of barcode technology, mobile and pervasive computing applications. We start with the work done in barcode technology, then in the mobile applications using barcode technology followed by healthcare applications and smart home technology.

2.1 Technologies

1. Android: Android is a software platform that includes an operating system, middleware and crucial applications for mobile devices. It is based on Linux operating system and developed by Google and open handset alliance. All Android applications are written using Java programming language. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack. The Android SDK provides the tools and APIs necessary to develop applications on the Android platform using Java [10].
2. ZXing: ZXing (pronounced "zebra crossing") is an open-source, multi-format 1D/2D barcode image-processing library implemented in Java. It is used for providing APIs to decode barcodes, which can be scanned by built-in camera on mobile phones [11]. This library is divided into several components; most are actively maintained and supported:
    i. **core**: The core image decoding library, and test code
    ii. **javase**: J2SE-specific client code
    iii. **android**: Android client, called Barcode Scanner

    iv. **androidtest**: Android test applications
    v. **android-integration**: Supports integration with our Barcode Scanner app via Intent
    vi. **zxingorg**: The source behind zxing.org/w
    vii. **zxing.appspot.com**: The source behind our web-based barcode generator

3. Barcodes: Barcodes are a way of representing information, which is machine-readable. They allow collecting data quickly and accurately. Barcodes can be read by optical scanners called barcode readers, or scanned from an image by special software. They are of two types: 1D and 2D. The 1D barcodes hold information about a product item such as a product code or any other ID. Commonly 1D barcodes are used in shops and supermarkets where there is a barcode on each product. The barcode is read at the checkout to retrieve the price from a central database and update customer and stock information. 2D barcodes allow more information to be stored in small amount of space. They are generally used in storing name, address, and other information on business reply cards and website or web page address to connect to a particular site.

    i. Barcodes are much quicker, more efficient entry of information and are inexpensive and easy to implement. As barcodes are printed and processed by machines, they are processed much faster than standard human data entry and with a much higher degree of accuracy. Barcodes have the potential of dramatically improving productivity and reliability of nearly all applications. For an example, on an average it takes 6 seconds for an operator to enter 12 characters of data, where as scanning a 12 character barcode takes only 300 milliseconds. The error rate for typing is higher than scanning barcodes. A data entry error will translate into additional costs for a business that ranges from the cost of re-keying the data to shipping the wrong product to the wrong customer [6].

ii. There is another technology termed as RFID technology other than the barcodes, which is used for carrying information. Radio frequency identification (RFID) uses radio waves as means of identification. The most common way to use RFID is to store a serial number that identifies a person or an object, and perhaps a few other pieces of information, on a microchip that is attached to an antenna. The chip and the antenna together are called a RFID transponder or a RFID tag. The antenna enables the chip to transmit the identification information to a reader that converts the radio waves reflected back from the RFID tag into digital information. This information can then be passed on to computers that can make use of it.

iii. The primary difference between the two technologies is that barcode scans a printed label with optical laser or imaging technology, while RFID reader interrogates a tag using radio frequency signals. Barcode requires a line of sight where as RFID does not. Barcodes can be read only in a smaller range but RFID tags can be read at a much greater distance. RFID tags have a read and write capability but barcodes can only be read. Barcodes may be easily damaged but RFID are more durable. Cost and privacy are two big concerns with RFID. For instance, a RFID implanted in running shoes could be used as a tracking device.

iv. Barcodes are used to efficiently and reliably enter data into a system with little or no human interaction, effectively eliminating the human-error element from data entry. For instance, if the user needs to visit a website on his cell phone he can easily do it by scanning and decoding a barcode containing a long URL rather than typing which might lead to some errors. Barcodes, as machine-readable representation of information in a visual format, can be easily stored, transferred, processed, and validated.

v. Barcodes have countless applications and are widely used for product identification, inventory marking, shipping container marking, and much more. 2D barcodes are almost like 1D barcodes - They identify an object uniquely. The big difference is that 2D codes can be used to virtually identify anything. Barcodes are extremely common and can be found in all types of industries, government organizations and public services. Barcodes provide a simple and inexpensive method of encoding text information which can be easily read using electronic readers. Barcode technology and processing provide a fast and accurate tool to enter data without keyboard data entry.

vi. The traditional barcodes are called one-dimensional barcodes. This is because they are scanned, or "read", in only one direction - horizontally. The vertical height of the barcode makes for easy scanning, but in itself does not add any additional information. The next generation of barcodes is generally referred to as 2D, two-dimensional, barcodes. These new style barcodes get their names from the ability to be read both horizontally and vertically, therefore increasing the density of information that can be encoded in the same amount of space.

vii. 1D barcodes hold a limited amount of information (generally less than 40 characters although there are exceptions) while 2D barcodes can hold considerably more (in some case up to 2000 characters). A 1D barcode only encodes data along the width of the barcode, while 2D barcodes encode data along both the width and height. 2D barcodes are therefore much denser than 1D barcodes and encode much more data in the same space. 2D barcodes are also much better at detecting and correcting errors in damaged barcodes.

viii. However, 2D barcodes require more expensive barcode scanners to read the barcode, and in many cases, such scanners do not operate through the computer's keyboard input - and so may not interface readily to your specific application. Many 1D-barcode scanners (known as keyboard wedge scanners) connect in series with the keyboard - so, from an application's point of view, scanning a barcode is just like typing on the keyboard [12], [13], [14].

ix. A one-dimensional (1D) barcode is used to reference an external database. Essentially, the 1D barcode contains no meaningful data, e.g. a packet of cornflakes is scanned at the checkout in a store, the reference number contained in the 1D code is checked against an external database, and only at this time do we get a description/price for the article. A 2D code typically contains the data. There is no need to reference an external database, so a 2D barcode has much higher data capacity, and therefore it is not mandatory to reference an external database. A 2D code can serve as a portable data file. While 2D barcodes are much more powerful than 1D barcodes and are able to encode more data in the same space, some applications may not require the added functionality that 2D barcodes provide. 1D barcode is often a cheaper alternative to 2D barcode and is easier to print as it is less dense than 2D barcode. 1D barcode scanner is typically less expensive than 2D barcode scanner [12], [13], [14] .

x. Two dimensional codes offer higher data capacity, a better fit in tiny spaces (matrix symbols) and depending on the scanning device, the ability to be read without regard to orientation. Even with these advantages, 2D symbology developments may not replace most linear bar code applications. The scanning performance tradeoffs should be weighed when choosing 2D over 1D. In general if the application requires encoding less than 50 characters or space is limited to an inch, linear codes should be considered unless a supplier compliance directive to use a specific 2D symbology is either strongly encouraged or required.

2.2 Related Work

2.2.1 Barcode technology

Significant work has been done on the topic of barcode technology and mobile applications. With the advancement of technology, a new technique came to enhance the capability of 1D barcodes especially in data capacity [15], termed as 2D barcodes. 2D barcode can carry a lot more information and can act as a small record of a database itself without requiring any data access. It has the capacity to encode alphanumeric data [1].

Just like 2D barcodes, another technology that gained popularity is smart hand held device with high computing capabilities because of the size and portability. These mobile phones act as pervasive computing tools and they are always available to the users, provide connectivity and also one can access the Internet [16]. Nowadays mobile phones come with variety of sensors like accelerometer, GPS, camera etc. Camera has now become a very powerful sensor to interact with physical objects including two dimensional barcode [16].

2.2.2 Barcode and cell phone technology

Earlier a lot of work regarding the recognition of 1D barcodes using camera phones has already been made. Steffen et al [17] had presented an algorithm for recognising 1D barcodes using camera phones which was highly robust regarding image distortions. Barcodes are ubiquitous and are powerful way to link information and services with physical world. Although, the demand of RFID technology has increased in recent years but using this technology for product identification and labelling especially using RFID tags on retail products is very unlikely in the upcoming years. Robert Adelmann [8] describes the work with the barcode technology and built in camera that enable a simple and fast interaction of user with every day products while shopping.

As mentioned before, cell phones with camera can capture image and process it, therefore many 2D barcodes have also been developed for mobile phone users [9]. H. Hahn and J.K. Joung came up with an algorithm to decode 2D-bacode PDF-417 [18] in 2002. In 2004, E. Ohbuchi et al [19] presented an algorithm for decoding QR code. 2D barcodes can store data in both vertical and horizontal directions and have the capability for error detection and fault correction. There are many different barcode symbologies in use today and many of them are introduced specifically to achieve higher data density such as PDF417, MaxiCode, Data Matrix, Code 49 and Quick Response (QR) code [1].

Robert Adelmann et al [20] created a free, easily usable, and robust barcode recognition system for mobile phones, together with an open resolving framework that facilitates rapid prototyping and deployment. Madhavapeddy et al. [21] used SpotCodes for enhancing human computer interaction by using a camera-phone as a pointing and selection device.

Seven 2D-barcodes namely QR Code ,VeriCode, Data Matrix, mCode, Visual Code, ShotCode and ColorCode are used nowadays for camera phone applications. The first four barcode in addition to their higher data capacities can enhance the reading robustness of the codes by error detection and correction capability. Hence, these 2D-barcodes can operate as robust and portable data files. With these barcodes, users can access the information they need at anytime, anywhere, regardless of network connectivity. As for the last three 2D-Barcodes, they focus more on robust and reliable reading, taking into account the reading limitations of built-in cameras in mobile phones. They differ greatly from the other barcodes in terms of data capacity [3]. H. Kato and K.T. Tan identified three key factors to improve the robustness of 2D-barcode reading, the range of the reading distance and the stability of the readers [3]. Teoh Chin Yew et al. [22] conducted a study which performs spatial analysis on several 2D barcodes using various size data sets. In addition the work also includes the analysis of several compression applications which can be used in minimizing barcode’s size.

The combination of barcodes with mobile phones opens a door way for new applications. Tsung-Yu Liu et al. [23] constructed a 2D barcode and handheld augmented reality supported learning system called HELLO (Handheld English Language Learning Organization), to improve students’ English level. Mobile phones with Internet capability are very popular among students in Japan, so Naomi Fujimura and Masahiro Doi [24] implemented a classroom support system to collect students’ degree of comprehension of course content using mobile phones. They devised this system to offer better ease of use for both students and teachers. Jun Rekimoto et al. [25] describes examples of augmented reality applications based on CyberCode, and discusses some key characteristics of tagging technologies that must be taken into account when designing augmented reality environments.

With the swift increase in the number of mobile device users, more wireless information services and mobile commerce applications are needed. Jerry Zeyu et al. [1] discuss 2D barcode concepts, types and classifications, major technology players, and applications in mobile commerce. They also report a research project to develop a 2D barcode processing solution to support mobile applications. S. Lisa, G. Piersantelli describes the features, applications, functionalities and benefits of the 2D barcodes, pictorial machine-readable representations of information and data [26]. Jonathan M. McCune et al. [27] present and analyse Seeing-Is- Believing, a system that utilizes 2D barcodes and camera phones to implement a visual channel for authentication and demonstrative identification of devices.

2.2.3 Healthcare technology

A lot of research work in ubiquitous computing area has been done on improving health care technology. Katie A. Siek et al. present a formative study that examines what, when, and how participants in a chronic kidney disease (stage 5) population input food items into an electronic intake monitoring application. Participants scanned food item barcodes or voice recorded food items they consumed during a three week period. The results indicated that a learning curve was associated with barcode scanning [28]. Furthermore, visually-impaired people living independently face problems in determining the contents of packaged foods, both at home and when shopping in modern, ‘self-service’ stores. So a project has been done by R.I. Damper et al. [29] which was aimed at assessing the feasibility of using product barcodes as an aid to the identification of package contents. Technical feasibility has been addressed by production of a prototype barcode scanner featuring synthetic speech output.

Martín López-Nores et al. [5] introduce an intelligent medicine cabinet as a new element of a residential network, acting as a secure place to store sensitive health information, and there from access a range of interactive health care applications. Their paper describes the functionalities related to monitoring the intake of prescription and over-the-counter drugs, harnessing recent advances in smart medicine packaging and home networking.

Smart Homes, with modern technology designed especially for the elderly and persons with special needs, have been a research subject in the last few years. One of the areas in which the elderly and persons with special needs would need assistance from the Smart Home is in their medicine intake management. José M. Reyes Álamo et al. [30] present the Medicine Information Support System (MISS) which integrates the patient’s information to assist with the prescriptions management. The system checks for conflicting medicines, health conditions and food items. The data generated is used to feed other subsystems in the Smart Home such as the calendar reminder and medicine inventory.

2.2.4 Cell phone and healthcare technology

Mobile and wireless devices have penetrated the health care sector due to their increased functionality, low cost, high reliability and easy-to-use nature. However, in such health care applications the privacy and security of the transmitted information must be preserved. DasunWeerasinghe et al. [31] propose a protocol for XML based security between a patient and the health services that builds upon a mobile communication environment. Murugaraj Shanmugam et al. [32] present an initial, yet a novel, approach to transfer the medical records of an individual in a secure way by using the existing infrastructure to mobile operators.

Another research project, known as MIMOSA, initiated by J.M. Quero et al. [4], aims to make Ambient Intelligence applications. It talks about building health care applications on mobile phone centric smart sensor network. Trinetra, a system proposed by Patrick E. Lanigan et al. [33], aims for cost-effective, assistive technologies to provide an independent grocery-shopping experience for the blind by leveraging barcodes and networking diverse embedded COTS devices.

All the above mentioned works include the concept of ubiquitous computing along with mobile technology or barcode technology or their combination. Some of the work also includes health care and mobile technology. However none of them discussed about the impact of combining all the three different technologies namely barcode, mobile and health care (technology to assist in daily living). In our project, we created a system that belongs not only to a mobile-based barcode technology application but also a pervasive computing application that can assist people in managing information. Since information system can be related to health care, food (grocery shopping), geography, website so it provides a much more generic way to interact with the user to access information belonging to any domain in a secure way. Trinetra [33] uses 1D barcode to collect information from public databases through the Internet but does not use 2D barcodes as well as the store database to access information. The information available from the Internet may not be available for all grocery products and updated frequently with the latest changes in the price of the product. Although our system is not meant for blind people, it provides a more efficient way of collecting information even if it is recently updated. It is also built on the MISS [30] as one of its subcomponent performs conflict and allergies checking, not only on the basis of the name of the food items but on the basis of the ingredients of the items.

CHAPTER 3. SYSTEM DESIGN

This chapter describes the overall design of the information management system developed in this project. Firstly a brief overview about the system components is given. It then explains in detail the usage of barcodes and XML technologies in building the system followed by the detail description of all the components used in development of the system.

3.1 System overview

The overall system consists of a cell phone based component and some external components through which the mobile-based system interacts to perform certain operations and to gather data. These external components can be on the same or on different machine and their number can vary. The interaction between the cell phone and external component is just like a client-server interaction in which the mobile-based system acts like a client while the external system like a server. The client sends a request to the server to get data, processes the data and display the result to the user. However, the connection between the mobile-based system and the external server is temporary and is built only when some event is triggered. The mobile-based system requires human intervention to trigger those events. In other words, the cell phone based component interacts with a user and triggers an event depending on the user chosen operation.

The mobile-based system is designed to decode a variety of barcodes, provides information to the cell phone user about the barcode, its type and content. Furthermore, based on the content of the encoded information, the application can distinguish between the different functionalities that can be provided to the user. It can perform different operations depending on the content type.

Our focus is on creating a generic system in which one can add the operations to the existing system depending on the content type defined in the interfaces and classes and adding new interfaces and classes for a new functionality.

In order to define the usage of such system, we have divided the system into sub-components based on the operations it performs. These subcomponents are inter-related and can have overlapping functionalities. By dividing the system into components, we can emphasize on the main functionalities that are provided by each of them and what can be the possible applications of the components.

One of the sub-components of the cell phone based system describes the importance of this system by showing its usage in real world through a small application in which a 2D barcode encodes information of a food product. Any mobile user can use this sub-component installed on his handset at the time of grocery shopping to decode the barcode and perform intended operations. Since the component has a practical application for assisting user with grocery shopping, we term it as a grocery-shopping assistant. This component of the system shows how a 2D barcode can be used to encode information related to different products. It can assist any mobile user in shopping and gathering interested information to the user and displaying result of the computation.

Moreover, the system has the capability to decode information (text only) of different content types like URL, contact information, calendar, telephone number, email and any other text information and depending on the content type, it will provide different functionalities. This module is termed as generic content processor.

Another important component of the system is that if a barcode contains contact information, then the system will detect the barcode's content type, display the content type to the user and provides him an option to save the contact information in the contact book of the cell phone. This sub-component is named as contact saving application.

The server-based system consists of a request processor subsystem that handles the request sent by the mobile client and generates response. The request processor processes the request sent and interacts with the database connector for further processing. The other component, termed as database connector, connects to the external database to query the database for the appropriate data asked by the request processor and sends the results back.

3.2 Usage of Barcode and XML technologies in system design

In this project, we try to emphasize on how digital barcodes can provide an effective means for mobile E-commerce application. Our system can decode both 1D and 2D barcodes. We use Qrcode (2D barcode) for encoding food product information as well as 1D barcode data displayed on the food product to provide useful information to the user of the system. We embed the data associated with 1D barcode along with the other information in a 2D barcode. So a 2D barcode not only represent static information embedded inside it but dynamic information (the one that can be obtained from external sources) as well. It is dynamic because we can use the 1D barcode data for accessing the external database. So we are not only providing the user with contents encoded inside the 2D barcode but also the contents that can be accessed from external sources through the use of data encoded inside 1D barcode. For accessing the data from outside, we may or may not connect to the Internet and depends on the location from where we are trying to access the information. If we are trying to access the food item related data inside the grocery store then we may not require Internet otherwise we need to.

3.2.1 Barcode generation and decoding

In order to test whether the system properly decodes the barcode or not, we used some online free barcode generators to generate barcode images. There are many free generators available in the market some of which are online and others can be downloaded. These generators allow you to encode data of your choice in the barcode. Some of them also generate image of any type in 1D or 2D. One of the online generators that we used is [34] and the other one is [35] for 1D barcode.

We also used another generator provided by ZXing from Google to generate 2D barcode images and of QR Code type. This is the one, which is used heavily in the development and testing of the system. The advantage of this QR Code generator is that you can encode information belonging to certain category of text. It provides the user an option to choose the content type or the kind of information needed to encode. These options include calendar event, contact information, URL, email address, geographical location, phone number and text that can be anything [36].

Depending upon the option or the content type chosen, the generator asks the user to enter certain information that can be encoded. It then selects the specific standard format for that particular content type, formats the information entered by the user according to the standard, encodes it inside the barcode and creates a barcode. Once the barcode is created, image can be easily downloaded. For example if the user selects contact information, the generator will automatically provides text boxes for the user to enter data for name, address, phone number and then use all this data to create a specific format. This specific format is visible when we decode the barcode image. Since the options available to the user for the content type is fixed, the generator knows how to format the contents according to the standard for a particular content type.

Moreover, if the user selects the content type as text, he can encode any information and there is no specific format used by the generator to encode the data. One can see the same data after decoding without any format. Therefore, in this case, the user has to come up with his own way of formatting the contents and then providing the formatted information to the generator to build a barcode image. Since the generator provides a single textbox to enter data for text content type, one can enter data of any complexity as well as in any format. In our project, we used the "text" content type to encode the complex information type. We have also defined some categories to represent such complex information and came up with our own standard of formatting this information. This formatted information can be fed to the generator [36] to generate the 2D barcode image.

There are many barcode decoding softwares available in the market. To build this project we used the APIs provided by Google known as ZXing [11]. As mentioned before, ZXing is a multi-format 1D/2D barcode reader library implemented in Java. Out of all the packages of the APIs we are using com.google.zxing, com.google.zxing.client, com.google.zxing.common, com.google.zxing.oned, com.google.zxing.datamatrix, com.google.zxing.qrcode packages in implementing the system.

3.2.2 Barcode contents

2D barcodes encode text, generally, but that text can represent many things. 2D barcodes can encode many types of actionable text. Text representing contact information, when recognized, could trigger a prompt to add the contact to an address book. However, this only works when readers understand that text encodes contact information. For this, we need standards too. There are some standards already in use to encode URL, email address, telephone number, contact information, sms, mms etc. [37].

As mentioned earlier, we define a new category of information and decide its format. The new category that we defined is for embedding information of a food item in a barcode. Typically, one can see information like nutrition facts, percentage and type of vitamins and minerals, calories, ingredients, price, expiration date, manufacturing location, 1D barcode label, website and contact information on any food product item that is available in a grocery store. All this information needs to be included inside a 2D barcode for a food item. We use all the above-mentioned information for creating a 2D barcode. To represent this information in a meaningful way, which the system can understand, we define a specific format. With this specific format, we create the barcode image and then use our system to parse and decode the image.

We use XML format for encoding the product related information. The XML file contains the following elements:

1. FP: It stands for food product and is the root element of the file. Since the barcode contains information of only a single food product, the XML file contains information of only one product and not more than one. Therefore,

the root element is termed as FoodProduct to represent one single item. It also removes the need to create a child element of the root to represent one food item and repeat it for several items. It has NF, C, E, P, L, O, U, and T as the child elements. We will describe each of them in detail.

2. NF: This element is the direct child of the root element and stands for nutrition facts. On any food product item, we always find a table that provides the nutrition facts of the item. This table contains information about the amount per serving, calories, nutrients, vitamins and minerals and some additional information at the bottom of the table. To represent such information we used the following elements like s, Ca, N, Vi, M, e respectively as the child elements of NF.
3. s: It stands for starting text and is the child element of NF. It specifies the amount per serving and some additional information about the product. It is usually found on the top of the table of Nutrition Facts on any food item.
4. Ca: It represents calories. It is also the child element of NF. It specifies the number of calories in the food item and is generally represented as a number.
5. N: It is the child element of NF and represents nutrients. It provides the information about the name of the nutrient and the percentage or the amount (in milligrams or grams) of carbohydrates, proteins and fats in the item. Since this element can represent any of the nutrients from carbohydrates, proteins and fats, it repeats itself by specifying the data associated with it as either carbohydrates or proteins or fats and a child element to represent the amount of the corresponding nutrient. It also has an id attribute which increments itself each time the element repeats.

6. v: This is the child element of N and stands for value. It consists of the amount information for any of the nutrients. For example, if the amount of carbohydrates in a food item is 27grams then this element contains data as 27g.
7. Vi: It stands for vitamins and is the child element of NF. It provides the information about the name of the vitamin and the percentage of that vitamin in the item. Since this element can represent any vitamin, it repeats itself by specifying the data associated with it as the name of the vitamin and a child element to represent the amount of the vitamin. It also has an id attribute which increments itself each time the element repeats.
8. a: This is the child element of Vi and stands for amount. It consists of the percentage of a particular vitamin (represented by the parent element) in the food product. For example, if the percentage of Vitamin B in a food item is 4% then this element contains data as 4%.
9. M: It stands for minerals and is the child element of NF. It provides the information about the name of the mineral and the percentage of mineral in the item. Since this element can represent any mineral, it repeats itself by specifying the data associated with it as the name of the mineral and a child element to represent the quantity of the mineral. It also has an id attribute which increments itself each time the element repeats.
10. q: This is the child element of M and stands for quantity. It consists of the percentage of a particular mineral (represented by the parent element) in the food product. For example, if the percentage or quantity of Iodine in a food item is 2% then this element contains data as 2%.

11. e: It stands for ending text and is the child element of NF. It provides some additional information about the product. One can put some important notes or remarks in this field.
12. C: It stands for contents. It is the child element of FP. It indicates the ingredients of the food item. It has one child element, which repeats itself and represents the name of an ingredient.
13. I: It stands for ingredient and is the child element of C. Since a food item contains more than one ingredient, therefore this element repeats itself. It contains the name of the ingredient and has an id attribute. The attribute increments when the element repeats itself to represent more than one ingredient.
14. E: It stands for expiration date and is the child element of FP. One can represent date in any format for this element.
15. P: It is the child element of FP and stands for manufacturing price.
16. L: It is the child element of FP and stands for manufacturing location.
17. O: It stands for 1D barcode information. We keep the barcode number represented as a 1D-barcode label on the food item as one of the elements of the food product.
18. U: It stands for URL and is the child element of the food product. It specifies the web address of the manufacturing company of the food product. This information is optional.
19. T: This stands for phone number and is the child element of the food product. It provides the contact number of the customer support offered by the manufacturing company of the food item.

We used XML file to encode the barcode information because it provides a generic way to represent information and any one can use and create it easily by referring to the XML schema. It also provides all the advantages of being an XML file. It is easy to understand as compared to a text format. A text format can vary from one developer to the other and one needs to learn the format to use it. It is also hard to understand the exact conventions that one has followed to represent the information unless it is standardized. XML has become a standard format, it is easy to learn and it is well formatted as it is written in a simple way.

It becomes difficult to represent very complex information in text format, as we have to come up with lots of new convention that can again decrease the readability of the information. We also have to write our own parsing mechanism to parse the data but in case of XML, we do not have to write our own code for parsing. We can use the already developed and available parsers (the SAX parser) for parsing XML file.

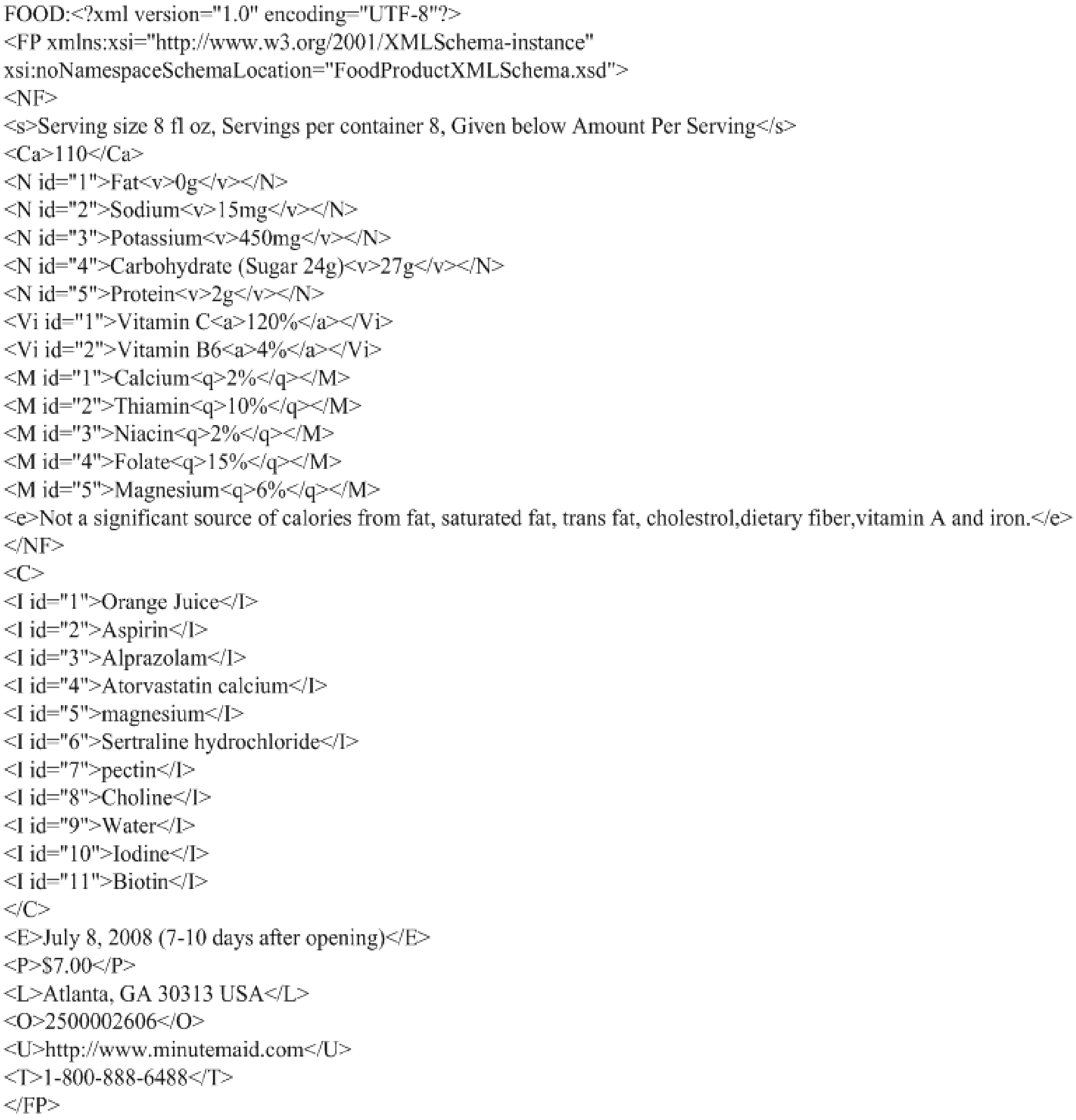
All the elements that we describe in the XML file have very short names. We tried to give simple names to all the elements so that they do not occupy a lot of space as barcodes have some limitations on the amount of data that can be encoded inside them. The number of characters that can be encoded inside a barcode is fixed and depends on the format of barcode. The barcode generator will not generate a barcode image if the number of characters exceeds the maximum limit. Therefore, if we use too many characters for naming the elements then we will not be able to represent the actual data completely.

Moreover, when we use a generator we need to make sure that the XML file contains no spaces between any elements. We tried avoiding unnecessary use of spaces between the elements to prepare the file as this also occupies a lot of space.

We can use that space to encode data because there is a limitation on the number of characters that can be encoded inside a barcode.

On the contrary, if we use a text format instead of XML file to represent the information then we can represent a lot more data than an XML file, considering the maximum limit of the data that we can keep in the barcode. However, in text format we need to make sure not to provide spaces between different sets of data. In general, if there is a lot of data to represent we can use a text format instead of XML.

Along with this XML file, an initial tag is placed before the XML file to specify that the information inside the XML is about a food product. This tag can help the program to understand that the XML file belongs to a particular category. In this case, it represents food item. However, one can use the XML file to encode any information. For example, one can encode information related to a medicine in the form of an XML file and can use a tag named as “Medicine” in the first line of the XML before the content of the file start. For instance, the diagram below shows the information that we can encode inside the barcode or the format of the food product information. It contains a tag named as “FOOD” with a colon just before the XML coding starts to represent a food item.

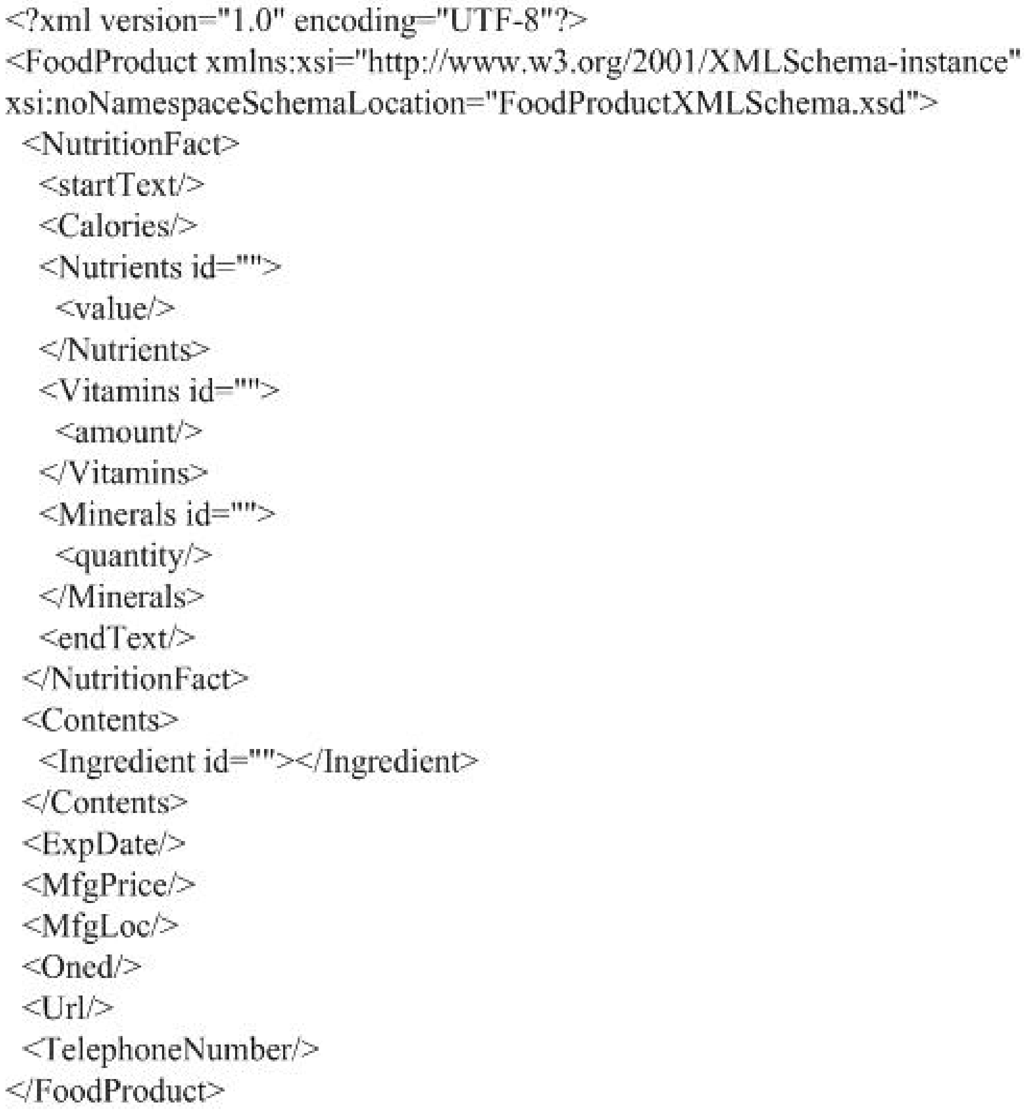


```
FOOD:<?xml version="1.0" encoding="UTF-8"?>
<FP xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FoodProductXMLSchema.xsd">
<NF>
<s>Serving size 8 fl oz, Servings per container 8, Given below Amount Per Serving</s>
<Ca>110</Ca>
<N id="1">Fat<v>0g</v></N>
<N id="2">Sodium<v>15mg</v></N>
<N id="3">Potassium<v>450mg</v></N>
<N id="4">Carbohydrate (Sugar 24g)<v>27g</v></N>
<N id="5">Protein<v>2g</v></N>
<Vi id="1">Vitamin C<a>120%</a></Vi>
<Vi id="2">Vitamin B6<a>4%</a></Vi>
<M id="1">Calcium<q>2%</q></M>
<M id="2">Thiamin<q>10%</q></M>
<M id="3">Niacin<q>2%</q></M>
<M id="4">Folate<q>15%</q></M>
<M id="5">Magnesium<q>6%</q></M>
<e>Not a significant source of calories from fat, saturated fat, trans fat, cholestrol,dietary fiber,vitamin A and iron.</e>
</NF>
<C>
<I id="1">Orange Juice</I>
<I id="2">Aspirin</I>
<I id="3">Alprazolam</I>
<I id="4">Atorvastatin calcium</I>
<I id="5">magnesium</I>
<I id="6">Sertraline hydrochloride</I>
<I id="7">pectin</I>
<I id="8">Choline</I>
<I id="9">Water</I>
<I id="10">Iodine</I>
<I id="11">Biotin</I>
</C>
<E>July 8, 2008 (7-10 days after opening)</E>
<P>$7.00</P>
<L>Atlanta, GA 30313 USA</L>
<O>2500002606</O>
<U>http://www.minutemaid.com</U>
<T>1-800-888-6488</T>
</FP>
```

Figure 1: Encoding format

It is clear from Figure 1 about the way we create the XML file. An XML schema is defined for the XML file and is called as FoodProductXMLSchema.xsd (refer to XMLSchema mentioned in the APPENDIX). By defining the schema, anyone can use the schema to create its own XML file with the proper standard.

In the above figure, the file does not contain any space and one can copy the content from the figure and feed it to the generator to generate the barcode image for the food item. In order to show with clarity how the XML file looks like, one can refer the sample file shown below in Figure 2. It shows the figure of an XML file with all the elements that are used to create the file.



```xml
<?xml version="1.0" encoding="UTF-8"?>
<FoodProduct xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FoodProductXMLSchema.xsd">
  <NutritionFact>
    <startText/>
    <Calories/>
    <Nutrients id="">
      <value/>
    </Nutrients>
    <Vitamins id="">
      <amount/>
    </Vitamins>
    <Minerals id="">
      <quantity/>
    </Minerals>
    <endText/>
  </NutritionFact>
  <Contents>
    <Ingredient id=""></Ingredient>
  </Contents>
  <ExpDate/>
  <MfgPrice/>
  <MfgLoc/>
  <Oned/>
  <Url/>
  <TelephoneNumber/>
</FoodProduct>
```

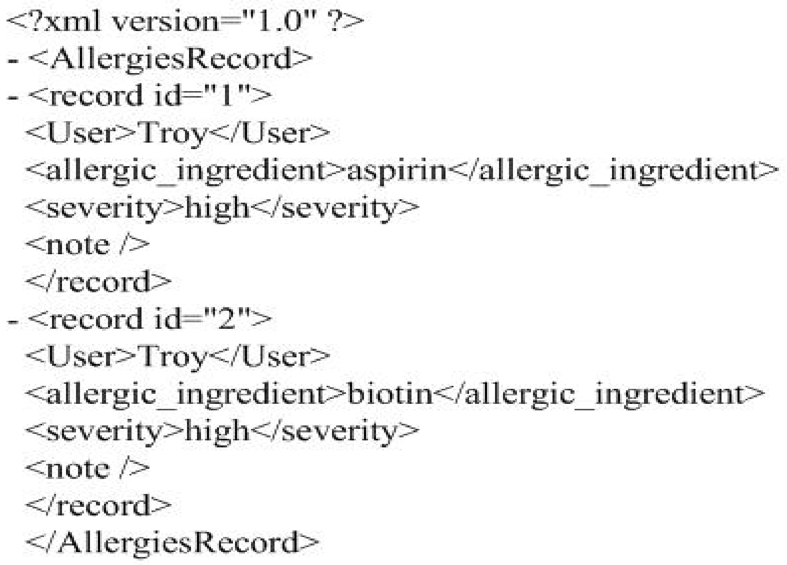
Figure 2: Sample XML format file

The XML file in Figure 2 shows the same format as the one in Figure 1. The one in Figure 1shows the file with a tag “FOOD”, before the XML file starts. This tag is used to distinguish different types of information stored in the XML file. We use the file shown in Figure 1 to generate the barcode image, which the system can use for decoding the information inside the image.

3.2.3 Use of XML in server response generation

In this project, we have used XML file not only for encoding information inside a 2D barcode but also for dynamically generating response files from one of the subsystems. The system, depending on the operation, can request one of its subcomponent to process the data and generate a result file that is in XML format. Several result files are generated and are categorized depending on the operation performed and the data it contains. One of the files is termed as conflictRecords file, the other one as allergiesRecords file and the third one as productAtStoreRecords file. All these files must be generated for proper functioning of the system. We will explain how the files are generated and its detail in Chapter 4. We will describe below the format of each of these files that are generated on demand for specific purposes.

1. allergiesRecords

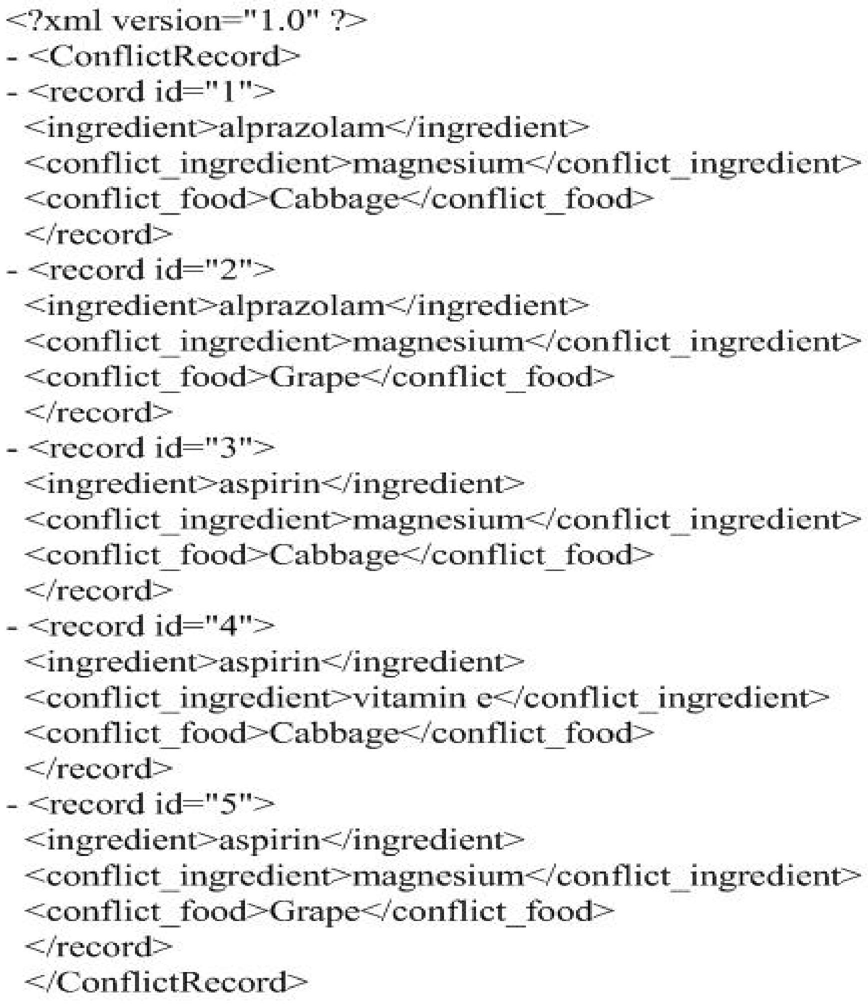


```
<?xml version="1.0" ?>
- <AllergiesRecord>
- <record id="1">
  <User>Troy</User>
  <allergic_ingredient>aspirin</allergic_ingredient>
  <severity>high</severity>
  <note />
  </record>
- <record id="2">
  <User>Troy</User>
  <allergic_ingredient>biotin</allergic_ingredient>
  <severity>high</severity>
  <note />
  </record>
  </AllergiesRecord>
```

Figure 3: XML records of user and allergies

The above diagram displays information about the user and his allergies. The element allergic_ingredient represents the ingredient in the food item which the user is allergic to.

2. conflictRecords

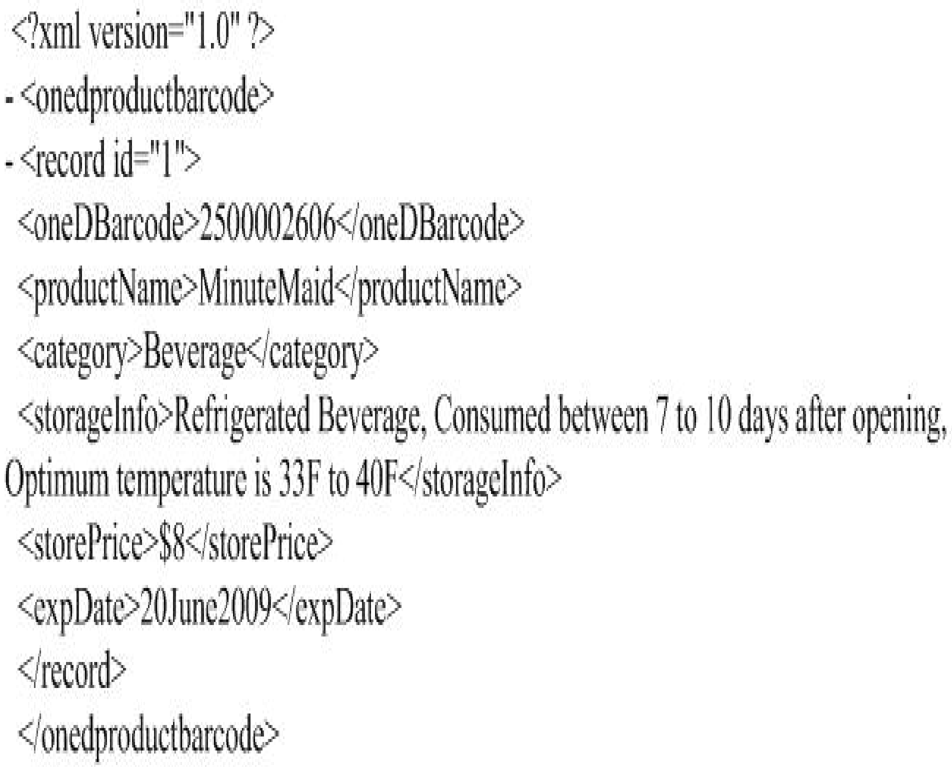


```
<?xml version="1.0" ?>
- <ConflictRecord>
- <record id="1">
  <ingredient>alprazolam</ingredient>
  <conflict_ingredient>magnesium</conflict_ingredient>
  <conflict_food>Cabbage</conflict_food>
  </record>
- <record id="2">
  <ingredient>alprazolam</ingredient>
  <conflict_ingredient>magnesium</conflict_ingredient>
  <conflict_food>Grape</conflict_food>
  </record>
- <record id="3">
  <ingredient>aspirin</ingredient>
  <conflict_ingredient>magnesium</conflict_ingredient>
  <conflict_food>Cabbage</conflict_food>
  </record>
- <record id="4">
  <ingredient>aspirin</ingredient>
  <conflict_ingredient>vitamin e</conflict_ingredient>
  <conflict_food>Cabbage</conflict_food>
  </record>
- <record id="5">
  <ingredient>aspirin</ingredient>
  <conflict_ingredient>magnesium</conflict_ingredient>
  <conflict_food>Grape</conflict_food>
  </record>
  </ConflictRecord>
```

Figure 4: XML records of conflicting ingredients

The above diagram displays the conflict between the ingredients. The ingredient element represents the ingredient in the food item for which we want to know the conflicting ingredient, which is denoted by the element conflict_ingredient in the file. The element conflict_food represents the food in which the conflicting ingredient is present.

3. productAtStoreRecords



```
<?xml version="1.0" ?>
- <onedproductbarcode>
- <record id="1">
 <oneDBarcode>2500002606</oneDBarcode>
 <productName>MinuteMaid</productName>
 <category>Beverage</category>
 <storageInfo>Refrigerated Beverage, Consumed between 7 to 10 days after opening,
Optimum temperature is 33F to 40F</storageInfo>
 <storePrice>$8</storePrice>
 <expDate>20June2009</expDate>
 </record>
 </onedproductbarcode>
```

Figure 5: XML record of product information

The above figure displays the information of a particular food item stored in the local store database. The element oneDbarcode represents the oneD barcode number associated with the food item and is used by the store to refer to the information stored in the store database.

3.3 System components description

We will describe each of the components of the mobile-based system and the server based system in detail in the following subsections.

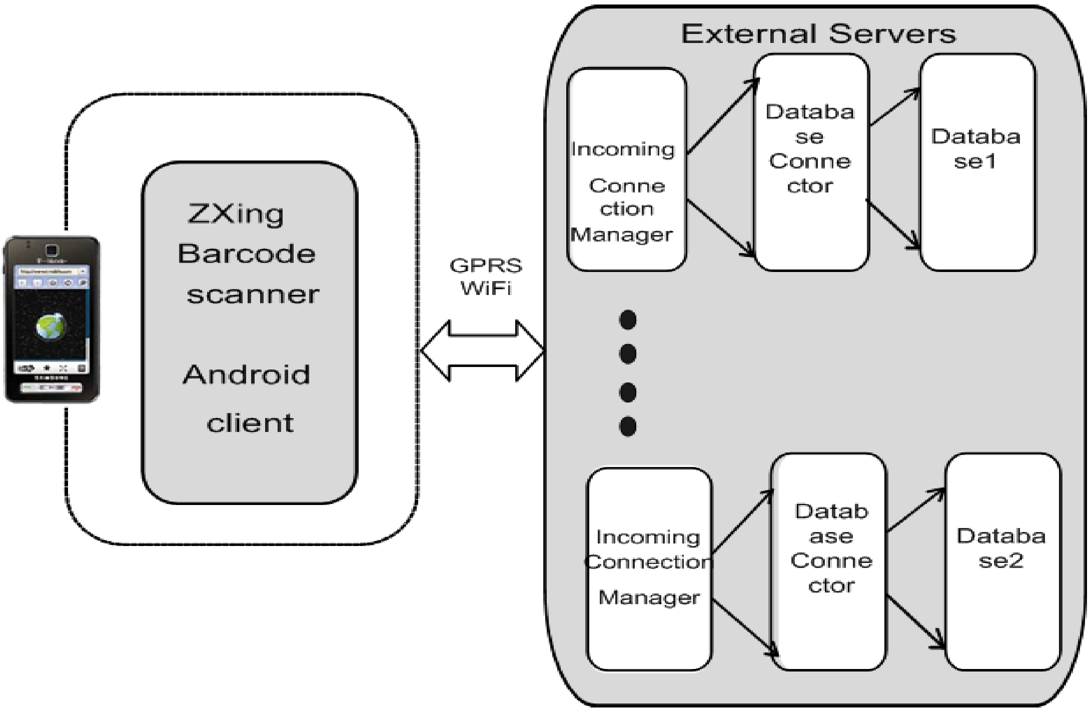

ZXing
Barcode
scanner
Android
client
GPRS
WiFi
External Servers
Incoming
Conne
ction
Manager
Databa
se
Conne
ctor
Databa
se1
Incoming
Connection
Manager
Datab
ase
Conne
ctor
Databa
se2

Figure 6: System Architecture

3.3.1 Mobile based sub-system

Grocery shopping assistant:

This component is designed to show how a 2D barcode can be used to encode information of a grocery product and how it can assist in grocery shopping. We describe below the working and design of the component:

If a 2D barcode is present on a food product, it can be scanned and saved in the sdcard of the cell phone. The image can then be decoded and can display the useful product related information to the user. It can also inform user if he should buy the product or not in case he is allergic to the ingredients of the food product or it might contain some ingredients, which can cause certain ailments when taken in combination with the other food item or medicine present at his home.

A food product related 2D barcode encodes the following information: nutrition facts, nutrients, vitamins, minerals, calories, ingredients, manufacturing related information like price, expiration date, manufacturing location, 1D barcode text, website and contact information. All this information is structured by using an XML file and then providing this XML file as an input to the 2D barcode generator to encode the file in a barcode.

We fix the format of the information encoded so that after decoding the barcode, the information can be processed based on the content. For example if we keep calories and its value as well as the mfg price and its value, we need to distinguish these two different fields and for that we need to define some tags that can be parsed later, or we can create an XML file to keep all the data and the corresponding values.

Functionalities provided by the application to the mobile user and the basic operations that the user has to perform to use these functionalities, are described below:

1. Application can decode the data inside the 2D barcode and display it to the user so that the user can get access to all of the above information on his handset. He does not have to access even the Internet for this purpose. Since the barcode contains food product information, it displays the entire ingredient list to the user after decoding.
    a. Assumption: It is assumed that every item in the grocery store has a 2D barcode along with a 1D barcode. The 2D barcode is placed on the food product at the same time as the 1D barcode is placed.
    b. Operation performed by user to access the functionality: Since all the grocery item related information resides on a 2D barcode, a user only has to scan the barcode, stores it in the sdcard and starts the application

by clicking on the scanning button. User needs to be present inside the store to scan the barcode however after he saves the image, he can use the application at any time and from anywhere to decode the information.

2. It also provides various options to display the decoded information other than the ingredient list like nutrition facts, nutrients, vitamins, mineral, calories, manufacturing related information to the user.
    a. Operation performed by user to access the functionality: User needs to click on the menu to see the options and then click on various options to get the desired content.
3. User can perform conflict checking for ingredients. Since the entire ingredient list is visible to the user, this list is used to find conflicts with the ingredients present in the other food items at his home. If any conflict is found then user can see those conflicts and will get cautious about not eating the conflicting food items together. By conflict, we mean that one ingredient can be harmful to take with another ingredient and might cause some ailments to the body. Therefore, it is necessary to check if any of the ingredients present in the food item can cause some disease when taken with some other ingredients of the other food items and one should avoid eating such combinations. The list of such combination of conflicting food items can help the user to decide whether he should buy it or not and if he buys then which of the food items (present in his home) he should not eat with this item.
    a. Operation performed by user for conflict checking: User only has to click on the menu option for conflict checking and he will be able to see the results on the screen. Moreover, he can perform conflict

    checking while buying the item as well as after buying the item and keeping it in his house. Since the barcode image resides inside the sdcard of the phone, one can use the application at anytime from anywhere whether inside the grocery store or at his home to decode the barcode to check the conflicts. This way it is easier for him to remember about the food conflicts.

4. User can also perform checking of allergies with any of the ingredients inside the food item not only for him but also for the family members. The information containing the set of allergic ingredients from the food item can be visible to the user on the screen.
    a. Operation performed by user: Just like in conflict checking, user can click on the menu for accessing the allergy related information and can access the information from the database at his home or in the store. However, it is better to check for allergies in the food item at the time of buying inside the store rather than after buying.
5. Application also provides the user to get the product related information fixed by the grocery store like the latest and the discounted price. Since the price of a product can vary from one store to other, therefore it is possible that the price mentioned inside the 2D barcode at the time of manufacturing can be different from what the store has decided as the selling price. So in order to know the actual and the current selling price as well as other information related to a product that can vary from one store to other, the user can use this information.
    a. Operation performed by user to access the functionality: The only prerequisite is that the user should be present inside the grocery store

then only he/she can access information using this application. He has to click on one of the menu options to get the desired result.

The information encoded inside a 2D barcode corresponding to a food item can be in XML or text format. When the format is XML, the decoded information needs to be parsed with an XML parser to display the contents. So, when the mobile user saves the barcode image in the sdcard, the application decodes the barcode from the image location and then parses it without connecting to an external server. Since the parsing is done by the sub-component on the cell phone (client-side), there is no need to access the Internet or any other network. SAX parser is used for parsing the XML file. Once the parsing is done, all the data corresponding to the different tags in the XML file are stored in the corresponding data fields of the Java object and some data is displayed to the user. In the case of food items, ingredients are shown on the cell phone screen while for other contents the data for display can be different.

When the user selects an option from the menu an event is triggered which in turn display some result based on the event after some background processing. The options provided to the user are categorized into three groups:

1. General information about the product: This includes getting the basic information like nutrition facts, price, location, nutrient, vitamins, minerals, calories of the food item. In other words, all of the information present in the XML file and stored in the Java object can be retrieved and displayed to the user through this option. For instance if the user selects the option for knowing the nutrition facts of the item, an event is triggered in which a method is called to get the data from the Java object and display it to the user.

2. User specific information: When a particular user selects any of the sub options from this group, the information displayed after the triggering of event is related and relevant only to him and may change from on user to other. Unlike the general information group, where the information displayed to the user after the event processing is only about the product, this group displays content that is dependent on the food item as well as the user, therefore may not be same for every user and varies. The operations provided include conflict checking, checking allergies of a particular user and all his family members.
    a. The system requirements are
        i. User will have a home database and a server setup.
        ii. The home database will contain information about the conflicting food items, food items that can cause allergies for the user and his family members.
        iii. User has already provided the address information of the home server to the application, which will ask the user for the server information when it starts for the first time to connect to the server. It is not necessary to connect to the Internet to access the database if the user performs these operations at home and has a network setup to which the mobile client can connect.
    b. Conflict checking: In this operation, the entire ingredient list of the food item scanned is checked for conflicts against the food items stored at the user's home.
        i. When a user selects the option for this operation, then the application contacts the home server where information about the conflicting food

items and ingredients is stored. It sends a request containing the complete list of ingredients to the server.

    ii. Server first retrieves the conflict list from the home database in which each item in the list contains the combination of conflicting ingredients. Then it compares one of the ingredients from the combination with each ingredient in the ingredient list sent by the client. If a match is found, the server stores the combination in an XML file. Thus, this XML file contains a list of such combinations.
    iii. Server then sends the generated XML response file to the cell phone client, which parses the XML file using SAX parser and display the result to the user.

  c. Allergies checking: In this operation, the entire ingredient list of the food item scanned is checked to find out if the user and his family members are allergic to any of the ingredients present in the food item.
    i. First, the application sends a request containing the complete list of ingredients to the server.
    ii. On receiving the request, server connects to the home database and queries the database for getting the set of ingredients to which the user is allergic and which are present in the ingredient list sent by client.
    iii. Once the server gets the result of the query from the database, it generates an XML response file containing the results of the query.
    iv. The client then receives the XML file, parses it and displays the result to the user.

3. Product related information at the store: This option provides the user who is present inside the store to get the product related information, which may vary from one store

to the other. The product related information becomes accessible by including the 1D barcode data inside the XML or text used for encoding information in a 2D barcode.

a. Significance of 1D barcode along with the 2D barcode: By keeping the 1D barcode text information inside the 2D barcode, one can use the 1D data to query the product related information residing inside the grocery store's database. Since the database at the grocery store saves the food product information based on the 1D barcode number, which is unique for every product, one can access the data related to a product by using the 1D barcode number.

b. 1D barcode and the store database: In practice, 1D barcodes are scanned and used to determine the current selling price offered by the grocery store on every item in almost all of the grocery stores. It can also be used to retrieve other information related to an item, which is liable to change by the store and can vary from one store to the other. This information can be only known and accessible to the store staff from the database owned by the grocery store to which a normal user does not have permission nor a way to access the data residing inside the database.

c. Permission from the store: In the current situation, one can access the information only if grocery store agrees to expose some useful product related information like current price, expiration date to the user from the database but can still restrict the access to the repository. This seems to be quite feasible to be done by the store and in turn provides a way to interact with each customer without the intervention of people working in the store.

d. Mechanism to access the data at the store repository: We propose a way to get access to the store’s database in a way that it still maintains the privacy of the information owned by the grocery store.

    i. In this case, the store can create its own private network whose range is within the store and no information is visible outside the store such that the user can connect to the network once they come inside but get disconnected automatically when they are outside.

    ii. This private network would have a server that exposes certain data and functionality in the form of a web service to any mobile client.

    iii. Any mobile client can use the web service to get the desired information. In this way, the database remains private and the server can expose some of the data of the database through its web service component.

    iv. Once the user connects to the network, the mobile client can automatically ask for the server address information and can connect to the server. Another alternative would be to scan a barcode containing the server address and then the mobile client can connect to the server. This would require the user intervention as the user has to scan the barcode that is placed somewhere inside the grocery store and save it on the cell phone, which can then be recognized by the application to connect to the server.

    v. The user can then easily access the information by clicking on the operation listed on his handset to view the product information at the store.

vi. Internally, when the user clicks to get the product information, the client sends a request to the server containing the 1D barcode number of the product. The server processes the request and sends back the response for that product which can be used to display the result to the user.

Contact saving:

The other sub-component of the system is decoding the contact information and saving it to the contact book of the cell phone. Once the barcode is decoded, its contents are stored in a Java object as its data members. If the user clicks on the option provided by the system to save the contact information, this information in the Java object is stored in the contact book. Similarly, if the barcode contains a URL, it will ask the user whether he wants to connect to its website.

3.3.2 External server based sub-system

Request processor:

It listens for request and if needed performs data processing. The mobile-based client sends a request by setting up an HTTP connection with this component. Once it receives the data, it connects to the database and generates query according to the data sent by the client. It queries the database, receives the desired data from database, generates an XML response file by filling the data received from the database in the file and sends the response back to the client.

Database connector:

It is used to create connection between the server and the database. The database can be present at the local machine or it can be accessed remotely.

Database:

It contains the data that is required by the application to generate results. For instance, the database at grocery store will contain the product related information like price, expiration date. On the other hand, the repository at home will contain the user specific data related to allergies and conflicts for food.

CHAPTER 4. IMPLMENTATION AND EVALUATION

This chapter describes the implementation details of the system. It also describes the tools used for the software development and the system specification. The system implementation describes the interfaces, classes, the static and dynamic relationships among the classes and interfaces with the use of diagrams. The evaluation part includes an experiment and its result, testing of system functionalities, measurement of usefulness of system through software quality metrics.

4.1 Tools used and technical specifications

The tools used for the development of the project includes eclipse Ganymede Version 3.4.2, Apache Tomcat v6.0 and Tomcat plugin for eclipse, Android SDK 1.5 (android-sdk-windows-1.5_r1), eclipse ADT 0.9 plugin for Android, ZXing API 1.3, MySQL 5.1, MySQL GUI tools 5.0, barcode generator software and modeling tool named as Visual Paradigm along with SDE version 5.0 for Eclipse. We used MySQL for creating the database, Java servlets to represent external server component and Android platform to develop the mobile-based system using Android and Java programming. We also used barcode technology, XML and web services.

The system requires Android software stack to run. It also requires Wi-Fi capability. Therefore, the cell phone must have the Android platform with a built-in camera and Wi-Fi in order to use this system. The total storage size of the application is 456 kb in which the application size is 452 kb and data size is 4kb.

4.2 Implementation

For implementing the system we grouped the components into several packages, each package contains Java classes some of which are related to each other.

Cell phone based component:

The cell phone based component must be installed on the cell phone and external servers need to be up and running for the proper functioning of the system. The starting point of the cell-based system is an activity class knows as ZxingTestActivity. This is the main class from where the Android application starts.

The ZxingTestActivity can start two other activities: One is UserSettings and the other is ZxingActivity. If the system is run for the first time on the cell phone, it will ask the user to enter the address information of the server set up at home. This information can then be used to connect to the server. Therefore, for the first time, ZxingTestActivity will call the UserSettings. Once the system has stored the server information, every time it starts the application with the ZxingTestActivity class it will check if the information is already stored or not. If not, it will start the UserSetings otherwise ZxingActivity.

One can also modify the server address information through UserSettingModify classes. ZxingActivity provides user with the options to modify the settings or to decode the barcodes in the sdcard. The decoding process starts with the BenchmarkActivity class. Every barcode image stored in sdcard is decoded and the barcode related information like the encoded text and barcode format are stored in an object of type BenchmarkItem.

The BenchmarkActivity can have more than one BenchmarkItem type objects depending on the number of images in the sdcard. These objects are then used by

BenchmarkSubActivity to display the content type and the image name of each of the barcodes decoded in the form of a list on the user screen. When a user clicks on any of the rows of this list, depending upon the content type of the barcode image the system decides what to display on the next screen.

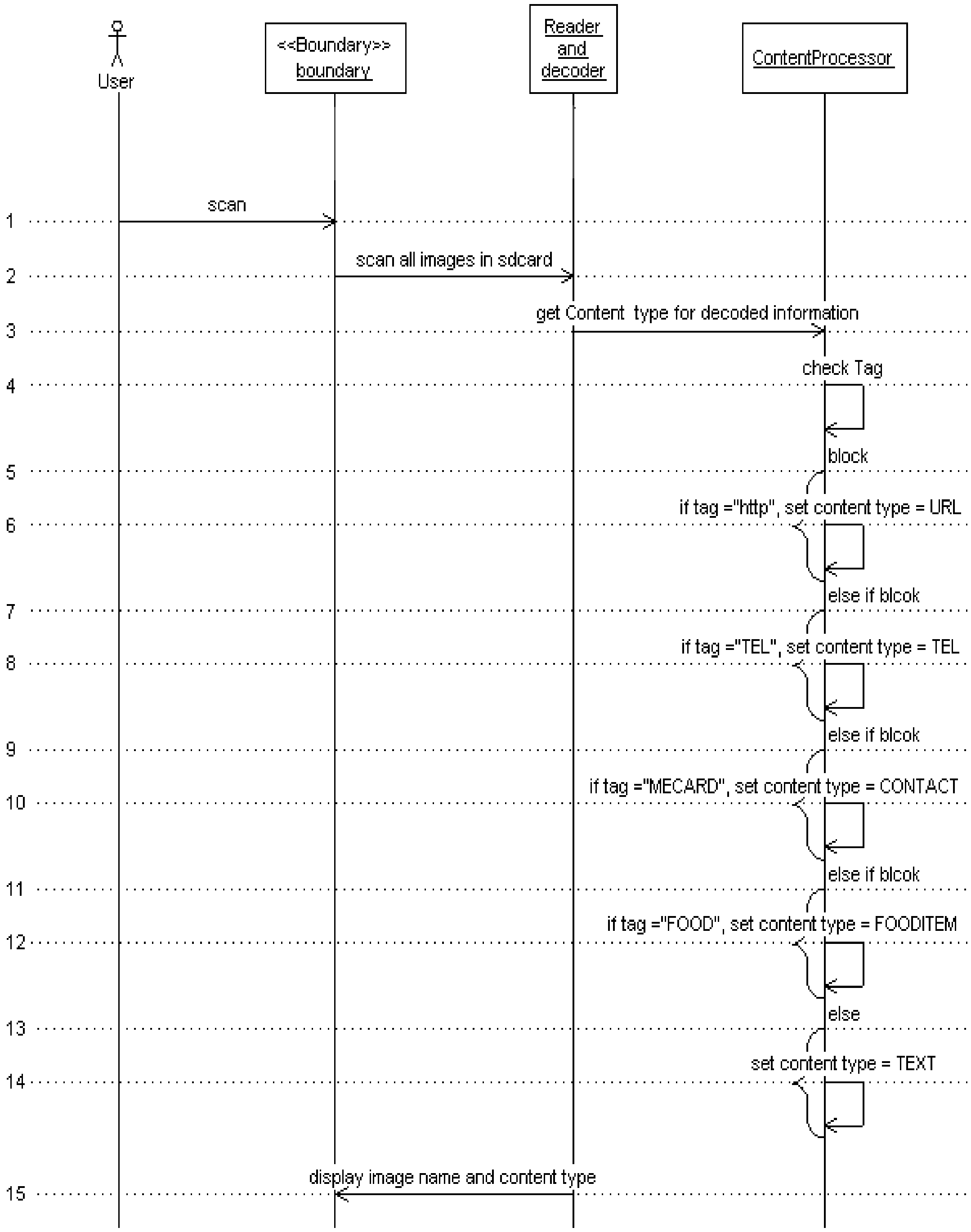

User
<<Boundary>>
boundary
Reader
and
decoder
ContentProcessor
1
scan
2
scan all images in sdcard
3
get Content type for decoded information
4
check Tag
5
block
6
if tag ="http", set content type = URL
7
else if blcok
8
if tag ="TEL", set content type = TEL
9
else if blcok
10
if tag ="MECARD", set content type = CONTACT
11
else if blcok
12
if tag ="FOOD", set content type = FOODITEM
13
else
14
set content type = TEXT
15
display image name and content type

Figure 7: Decoding of information and assigning content type

When the user clicks on one of the listings, the text encoded in that image is passed to ContentProcessor, which processes the content and depending upon the content type stores it in an appropriate object of interface type “Content”. Every barcode image can store text data but text can be anything. It can be a URL, telephone number, contact information, a message or all of them combined in one image, so to distinguish between the different kinds of text information a standard format is used which is already defined for some of the content types like URL, telephone number, contact information [37]. However, for encoding information related to a food product we define our own standard. By defining a standard, we can encode any complex text information. With the use of a standard format, the ContentProcessor can read the text and can categorize the information to a certain content type. For example, if a URL is encoded in a barcode, the text will always contain “http” as that is the standard to define URL. ContentProcessor will process the text and check if it contains “http”. If it does, its content type is URL.

Once the ContentProcessor distinguishes the text information, it can decide to either store the text or parse it depending on the content type and the complexity of information. It might have to parse the entire content and then store it in a Java object. For instance, if the contact information is encoded inside a barcode, ContentProcessor will recognize its content type as “Contact”. After recognizing its content type as contact, it knows that it contains various fields like name, telephone number with different tags to differentiate each of the information. Since the format of encoding information of any content type is fixed and standardized, therefore the format of encoding contact information is also standardized and fixed. The format for food related information is also fixed, although it was not pre defined. A format to encode the food related information is defined.

Since the format is fixed, ContentProcessor knows how to parse the data depending on the content type. It uses a parser, which parses the whole content and stores each of the information of different fields in different data types. This kind of information is complex and needs a parser to parse the information based on the format.

ContentProcessor uses a parser class for parsing the complex information. This parser class is termed as ComplexItemParser. ComplexItemParser is an abstract class that defines abstract and non-abstract functions to help in parsing the data. This class is sub-classed to parse the information belonging to different complex content category. If the category is of content type contact, the parser will parse the data based on different tags present in the information. Similarly, for food product, the parser will parse the data based on different tags. The tags used in representing food product information will be different as compared to the tags used by contact informaing methodology will be different. Therefore, we define two sub-classes called as FoodProductParser and ContactParser to parse the different category of information. One can easily add a new Parser class by making it a sub-class of the ComplexItemParser class.

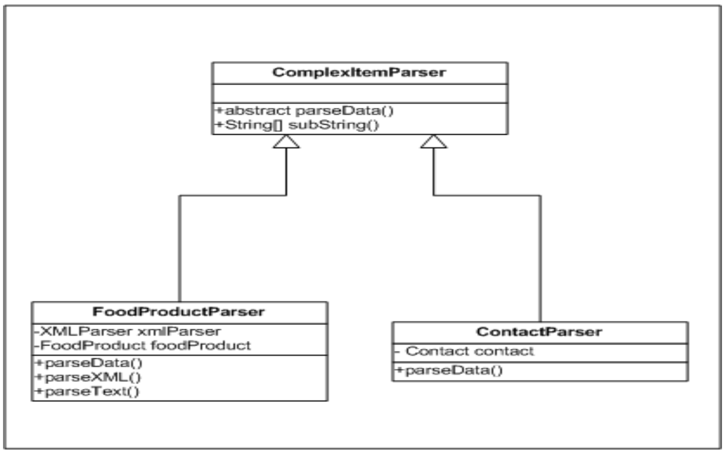

ComplexItemParser
+abstract parseData()
+String[] subString()
FoodProductParser
-XMLParser xmlParser
-FoodProduct foodProduct
+parseData()
+parseXML()
+parseText()
ContactParser
- Contact contact
+parseData()

Figure 8: Class diagram representing the subclasses of ComplexItemParser

Since the information encoded is of a particular format, it might follow some conventions in text format or it can define an XML format to encode the information. In the latter case, one encodes the whole XML file inside the 2D barcode while in the former case the information can be a user defined text format. Contact Parser parses the data specified in text format while the Food product Parser parses the data specified in XML format but it can also parse the data in text format if the text format is defined appropriately. We have defined the XML format for encoding the food product and defined the parsing method to parse the content of the file by creating the class FoodProductParser and a parsing method to parse. FoodProductParser also exposes a parsing function for parsing the data in text format, this function can be used to define one's own method of parsing.

FoodProductParser uses the SAX parser for parsing the XML file. The SAX parser parses the XML document based on events like the start and end of document, start and end of element, reading the characters between the start and end tag of the element. An XMLEventHandler_FoodProduct class is created to inform the SAX parser about how to parse the document. It also stores the elements of the file in the corresponding data fields of the Java object. Therefore, its function is to extract the data from the file and put it inside the Java object. Here the Java object belongs to the FoodProduct class, which stores all the information related to the food product like nutrition facts, ingredients, price in the data members of its object.

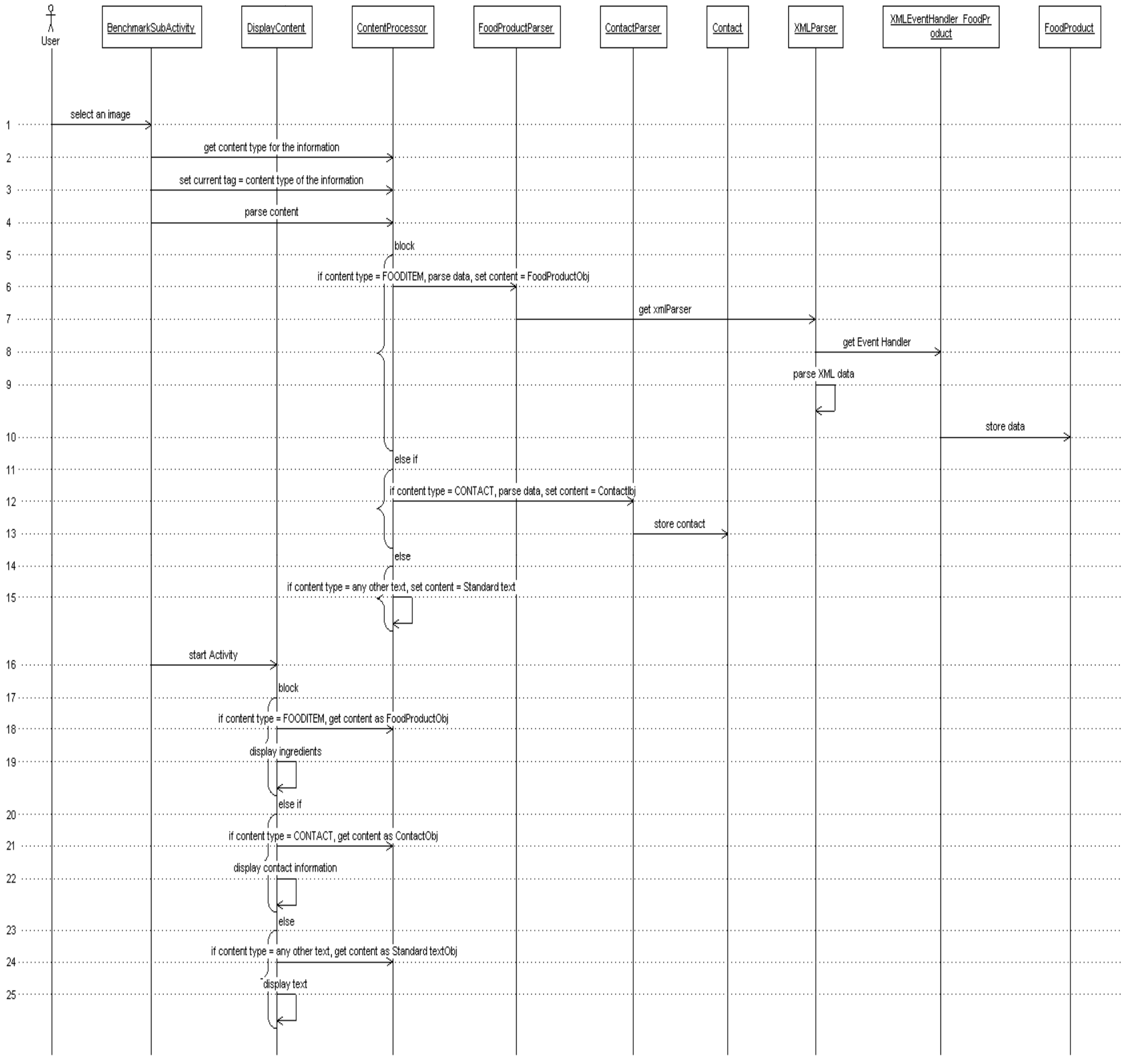

User
BenchmarkSubActivity
DisplayContent
ContentProcessor
FoodProductParser
ContactParser
Contact
XMLParser
XMLEventHandler_FoodProduct
FoodProduct
1
select an image
2
get content type for the information
3
set current tag = content type of the information
4
parse content
5
block
6
if content type = FOODITEM, parse data, set content = FoodProductObj
7
get xmlParser
8
get Event Handler
9
parse XML data
10
store data
11
else if
12
if content type = CONTACT, parse data, set content = Contactlbj
13
store contact
14
else
15
if content type = any other text, set content = Standard text
16
start Activity
17
block
18
if content type = FOODITEM, get content as FoodProductObj
19
display ingredients
20
else if
21
if content type = CONTACT, get content as ContactObj
22
display contact information
23
else
24
if content type = any other text, get content as Standard textObj
25
display text

Figure 9: Classification, parsing and display of information

Once the information is parsed, we store the complex text information in Java object because the information is complex and cannot be represented through primitive types. For instance to represent the contact information we need a corresponding Java object that can store the information about each of the fields (name, phone number). Therefore, we define certain classes that can represent such complex information and provides an interface, which is implemented by all such classes. The interface is a marker interface termed as Content. The classes implementing this interface are FoodProduct, Contact, and StandardText where the FoodProduct class stores information for food product, Contact stores contact information and StandardText stores information representing anything other than FoodProduct or Contact. It can be any text belonging to either simple or complex category and all the information whose format has not been defined yet can be stored in the object of this class.

One can add new classes to the existing set of classes to represent a different category, for example, one can store the product related information of an electronic gadget in a separate class by implementing the same interface Content.

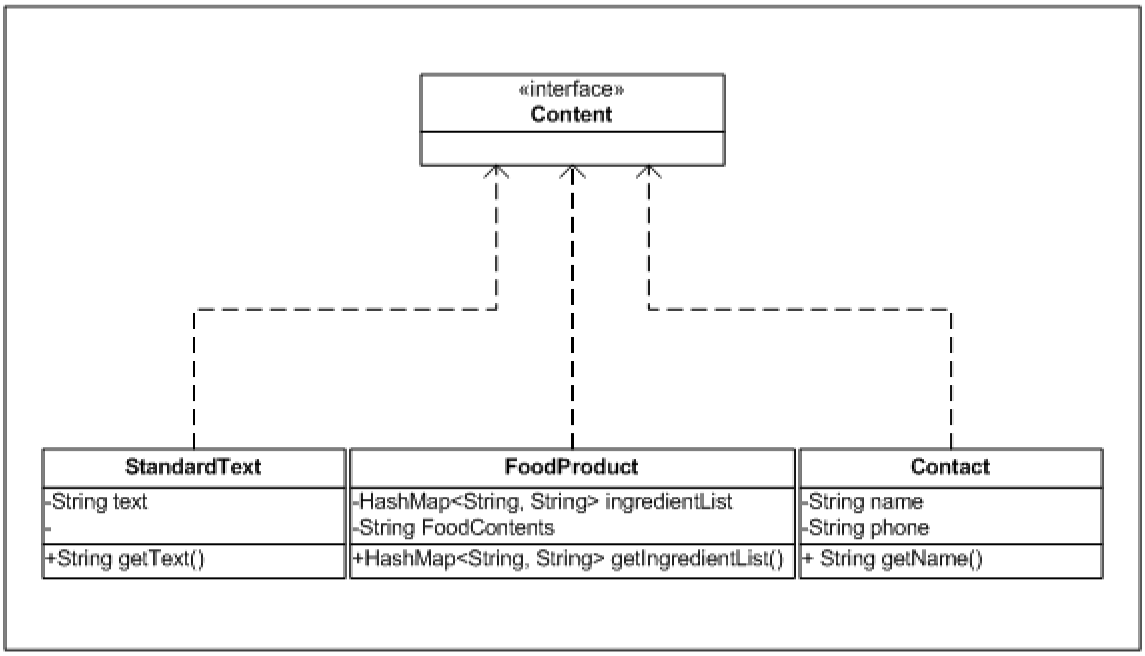

«interface»
Content
StandardText
-String text
+String getText()
FoodProduct
-HashMap<String, String> ingredientList
-String FoodContents
+HashMap<String, String> getIngredientList()
Contact
-String name
-String phone
+ String getName()

Figure 10: Class diagram representing the interface Content

The DisplayContent class uses the information stored in the Java object to show the contents encoded inside the barcode on the user screen. In case of food product, content processor parses the content and stores the information in the class FoodProduct. The DisplayContent class will then display the ingredients of the food item on the screen. One can display any information related to that product on the user screen. Accordingly, a menu is also created to provide user with various options.

In case of standard text, an option to get the information related to the barcode like barcode type and format is provided. For contact information, one additional option to save the contact is provided. For food product, there are three more options namely general information, user specific information and product information at store in addition to the option of providing the information about the barcode.

When the user clicks on the user specific information, he gets the option to choose one of the several operations. When he chooses one of the options, the corresponding method in UserSpecificInfo class is called. All the operations defined in this class require a connection with a server and they get the response data from the server once the method is executed. The response data is in XML format and needs to be parsed. The method calls the XMLParser class to handle the data. The response data is parsed with the SAX parser in the same way as the food product information is parsed. For each operation provided as the sub-option to the user there is a corresponding class, which is used by the SAX parser to parse the information.

For instance, for checking conflicts there is a class called XMLEventHandler_Conflict class that the SAX parser uses for parsing and storing the elements of XML in another Java object belonging to the class FoodConflictRecord. The response file generated is termed as conflictRecords file and a copy of the file is shown in Figure 4. If the XML file contains more than one record where each record is mapped to one Java object of class FoodConflictRecord then a hashmap is used to store all the objects representing different records. This hashmap object is passed to the class that displays the information to the user.

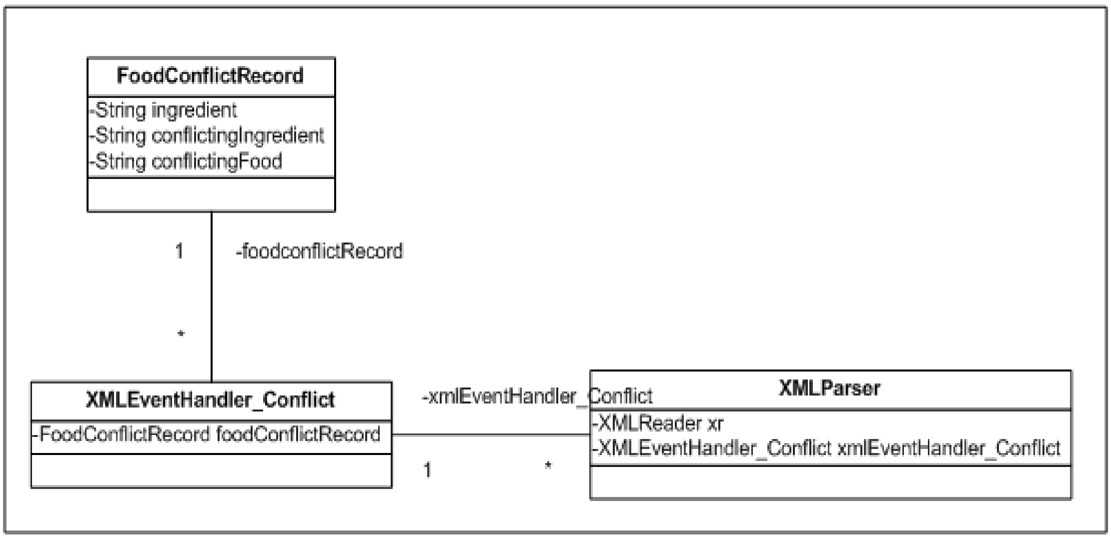

FoodConflictRecord
-String ingredient
-String conflictingIngredient
-String conflictingFood
1
-foodconflictRecord
*
XMLEventHandler_Conflict
-FoodConflictRecord foodConflictRecord
-xmlEventHandler_Conflict
1
*
XMLParser
-XMLReader xr
-XMLEventHandler_Conflict xmlEventHandler_Conflict

Figure 11: Class diagram showing relationship among the classes for conflict checking

Similarly, for checking allergies, the SAX parser uses XMLEventHandler_Allergies class and FoodAllergiesRecord class is used to store each XML record as a Java object. The response file generated is termed as allergiesRecords file and a copy of the file is shown in Figure 3. For finding the product-information at store, XMLEventHandler_ProductAtStore class is used and each XML record is stored in ProductAtStoreRecord object. The response file generated is named as productAtStoreRecords file and a copy of the file is shown in Figure 5.

The XMLParser class parses all kinds of XML file depending on the number of elements present in the file and the structure of the XML defined. In the diagram given below, Figure 12, we can see there are many event handlers defined for different XML files to extract the data from the files.

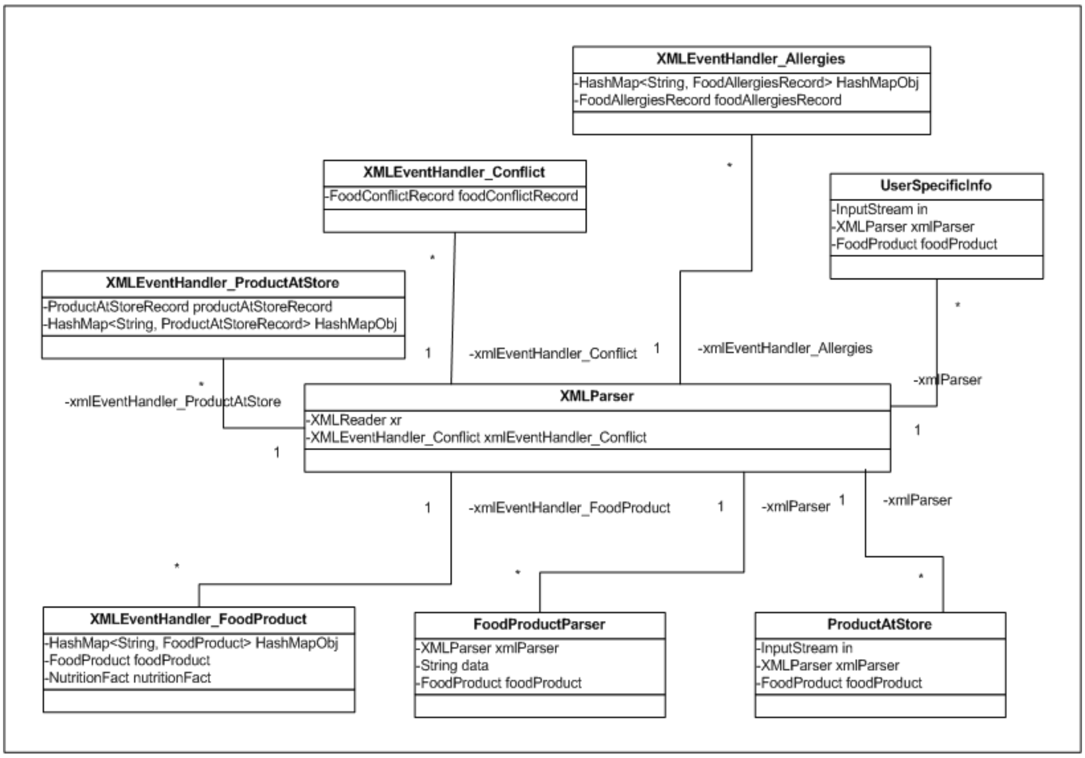

XMLEventHandler_Allergies
-HashMap<String, FoodAllergiesRecord> HashMapObj
-FoodAllergiesRecord foodAllergiesRecord
XMLEventHandler_Conflict
-FoodConflictRecord foodConflictRecord
UserSpecificInfo
-InputStream in
-XMLParser xmlParser
-FoodProduct foodProduct
XMLEventHandler_ProductAtStore
-ProductAtStoreRecord productAtStoreRecord
-HashMap<String, ProductAtStoreRecord> HashMapObj
*
*
*
*
1
-xmlEventHandler_Conflict
1
-xmlEventHandler_Allergies
-xmlParser
-xmlEventHandler_ProductAtStore
XMLParser
-XMLReader xr
-XMLEventHandler_Conflict xmlEventHandler_Conflict
1
1
1
-xmlEventHandler_FoodProduct
1
-xmlParser
1
-xmlParser
*
*
*
XMLEventHandler_FoodProduct
-HashMap<String, FoodProduct> HashMapObj
-FoodProduct foodProduct
-NutritionFact nutritionFact
FoodProductParser
-XMLParser xmlParser
-String data
-FoodProduct foodProduct
ProductAtStore
-InputStream in
-XMLParser xmlParser
-FoodProduct foodProduct

Figure 12: Class diagram showing relationships of XMLParser

When the data from the XML file is parsed, it is stored in the corresponding Java object and all such objects are stored in hashmap. The UserSpecificInfo class retrieves the hashmap object after the method execution and returns the hashmap object to the calling intent class. The intent class is the one, which is called when the user clicks on one of the operations. It prepares the GUI to display the data once the operation triggered by the user is performed. Therefore, this intent class further calls the methods defined in the UserSpecificInfo and waits for the result from it. Once it gets the result, it displays them on the screen.

Server based component:

The request processor component of the system is a servlet that handles the request. Since data is sent in the form of http Post request, we define the post method of server properly to generate the response.

The Database connector component is a Java class, which connects to the database through a JDBC connection and returns the connection information to request processor.

The databases are used to keep different sets of data maintained by different groups of people and can be present in different locations. A database can represent a home database or database at store. In case of home database, the database consists of family member's information, their allergies, information about various food items and their ingredients, information about conflicting ingredients. To represent the home database, we have created the following tables in the database called as conflict, food, ingredient, food_ingredient, allergies and family see Figure 13.

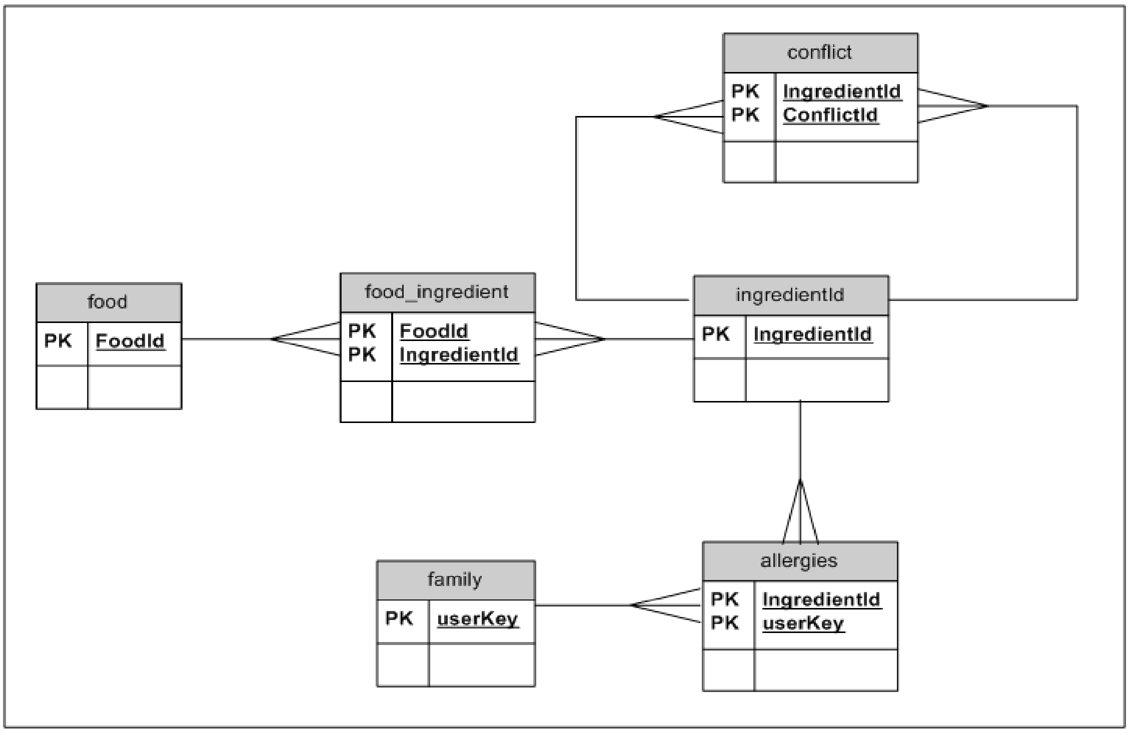

conflict
PK IngredientId
PK ConflictId
food
PK FoodId
food_ingredient
PK FoodId
PK IngredientId
ingredientId
PK IngredientId
family
PK userKey
allergies
PK IngredientId
PK userKey

Figure 13: Entity Relationship diagram for home database

In grocery store database, the information stored is about the various food products. We have defined a relation known as onedproductbarcode in the store database to represent the data related to the product, which varies from one store to the other. The table contains the following information: oneDId, oneDBarcode (1D barcode data), productName (name of the product), category (category of food item) example beverage, snacks, storageInfo (temperature of the storage where the item is kept), storePrice (price at the store) and expDate (expiration date).

4.3 System evaluation

The system is developed by using Android SDK 1.5 and tested on the emulator provided with the Android SDK. The testing of the system is performed iteratively starting from a small module to the whole component during the development phase and after the implementation is completed.

Experiments are conducted in order to evaluate how the system can handle different sorts of images. Some software quality metrics are used to evaluate the system and test cases are created to test the functionalities. A complete system testing is performed to check whether all the components are working properly or not.

We analyzed the response of the system with respect to different barcode images given as input. These barcode images vary in their types, size, source of generation and the complexity of information. Based on the analysis of the result and the inferences made from the result we showed the following:

a. The best possible source of generating barcode images that can be used by the system.
b. The appropriate size range of the barcode images for efficient use of the system.

Response Analysis of the experiment conducted:

The ZXing API provides the code for scanning and decoding barcodes. Since we are using Android SDK, we cannot scan the images rather we keep the images in the sdcard and then use the API to decode the images. Several options were tried to see how the API decodes the barcode images and what kind of images. We performed experiments with a variety of 1D and 2D images of barcodes from various sources and fed to the system to see if they can be decoded or not. We wanted to know how the scanner should work so that it can decode any kind of image obtained from any source. We also needed to know how the mobile-based content differentiates between the different content types and the response of the system in each case. We provided different sorts of images and checked whether the mobile based system works properly and generates the correct response or not.

Inferences made from the experiment:

1. Most of the 1D images generated from different sources get decoded. Even if the image is little unclear and faded then also it is decoded. For example, the images shown below are taken from a camera. Even these images are decoded by the system.

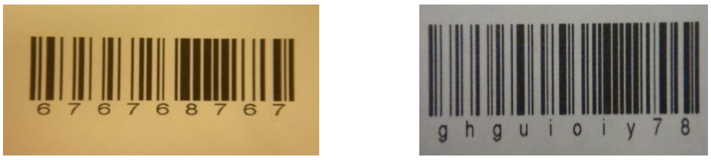


Figure 14: 1D barcode images captured from camera

2. Some of the 2D images captured from camera give error while some cannot decode. Only the image named as three_35_2.jpg and three_34.jpg are decoded. If we compare these images with the one, which has largest size, we see the following pattern: three_3.jpg > three_35_2.jpg > three_34.jpg. This

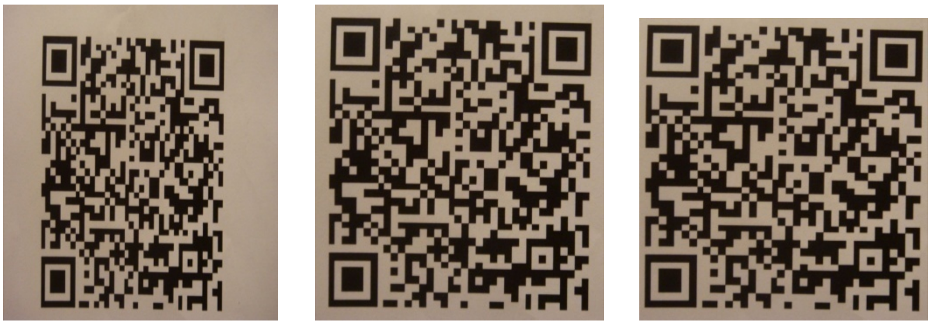
means that the size of three_34.jpg is the smallest. So if we keep on decreasing the size there is a possibility that the image can be decoded.

3. The images whether 2D or 1D obtained from barcode generator have small size. Most of them are less than 10KB. While the images captured from camera have larger size. The 2D image that has the size of 197KBs causes memory error. As we increase the image size, the probability of images getting decoded decreases. The table below shows the result

Decoding	Decode and categorize the image	Cannot decode the image	Error in decoding the image
Image Size (KBs)	10-100	100-150	150-200

Table 1: Effect on decoding with the change in image size

4. The other observation is that when we decrease the width we are trying to crop the white border on both the sides of the barcode. The more we crop the white border the more the possibility that the image can be decoded.



three_3.jpg **three_35_2.jpg** **three_34.jpg**

Figure 15: Difference among the 2D barcode images

The width of the image three_35_2.jpg (1176*1131) is optimal in the sense that if we increase the width by allowing a little more white space around the barcode, then it will not decode. However, we can still decrease the width to remove the white border completely. In the above diagram Figure 15, three_3.jpg (1680*1260) has a lot of white space around the 2D image, which is cropped in three_35_2.jpg (1176*1131) and further cropped in three_34.jpg (1111*1110).

5. Moreover, as the complexity of information increases the time it takes to decode the barcode also increases. For instance, the time to decode the barcode containing food product information is more than the time it takes to decode an image encoding contact information. The decoding time also increases with the number of images present in the sdcard. The more the images, the more time system will take for decoding and presenting it to the user. The number of images that can be stored in the sdcard is also limited.

Conclusion:

We can see from the observations mentioned in the inferences made from the experiment that the best barcode images that can be decoded by the system are the ones that are generated from the online generators rather than captured from the camera. Moreover, if the sdcard contains only one such image every time the system is used for decoding then the size range of the barcode can vary from 5KBs to 100KBs for the proper functioning of the system. However if there is more than one image in the sdcard then one need to ensure that the total size of all the images stored in sdcard does not exceed the size limit of the card. It is always better to use the images of small size than the larger ones.

Some of the software quality metrics to evaluate the system are described below:

Availability

The system has a 24/7 availability as the mobile client is installed on the cell phone and is easily accessible all the time. One can use it from anywhere and at any time. All the external components are always up and running.

Security and Privacy

The information obtained from the barcode cannot be modified, neither the information on the barcode can be changed. The information obtained from home database is only for viewing purpose and it cannot be changed. Moreover, the user can access this information from anywhere but the most secure place to access the information is the user's home. Since the home server need not have to be connected to internet and can be a part of user's private network, the user or the owner of the network can easily connect to the private network without accessing the internet. The owner must be at home to connect to the network and access the data in a secure way.

The system also maintains privacy of information obtained from a store (grocery store) because the store has its own private network and the information can be accessed only through that network. One has to be inside the store to access the information. The information available to the user from the store cannot be modified on the cell phone. It cannot be accessed directly from store database so the database remains secure and confidential. The web services or the server generating an XML response file is the only way to read the information. There is no other way to read and modify information by connecting to the server.

Our assumption is that the database present at the store and at home or anywhere is protected by the private network and it is behind an external server to which the access is restricted. The server is also protected from unauthorized access.

Maintainability

The business logic is clearly separated from UI to allow different user to develop interfaces in the future. Android provides the flexibility to developer to define the layout or view of the UI in an XML file. Therefore, anyone can change the UI without making changes to any of the business logic code. One can add new functions and operations by creating a new activity class, defining its UI and business logic without modifying the existing code. One can define its own class to represent the information encoded inside the barcode and define a new content type in one of the existing classes.

Modules are as independent as possible so that changes in one module will not produce exceptions in another part of the application. The external component and the mobile client are separate modules and anyone can be replaced without affecting the other. Moreover, the services offered by server include generating an XML response file that can be accessed by any client not necessarily a mobile client. The web services can also be used by any client. If the servers go down for some reason, it will not affect the major portion of the client functionalities although some functions may not work properly. The database can also be replaced easily without changing a lot of code on server side.

It will not be difficult even to use the system as a desktop application with a desktop client rather than a cell phone based client. The only code that needs to be changed will be the UI code, the activity classes that are the fundamental parts of

Android application. The rest of the classes are plain Java classes and can be used without many modifications in the code. The decoding API provided by Google known as ZXing also support javase client component and its core package contains pure Java classes. So, one can easily create a javase system to represent a desktop client.

Portability

This software is for small computing devices so it is extremely portable.

Usability

The system is user friendly. The contents on the screen are easily readable as not much data is displayed on every screen. User does not have to type a lot to perform any operations. The functionalities are very simple to operate and comprehend. Moreover, the time of scanning and decoding the images may vary depending on the number of images present in the sdcard. The system informs the user about the scanning process by displaying a screen indicating that the scanning process is going on.

CHAPTER 5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

This work describes an information management system, which uses barcode technology and is implemented for the cell-based platform. It shows the potential applications of using digital barcodes to carry useful information, how the system can be helpful in providing information to the mobile users as it is implemented for the cellular platform and how it can help the users in making decisions. This system is designed as part of a ubiquitous computing application as well as mobile-based commercial system. Since this system can help people in making decisions and assist them in their daily activities involving shopping, eating and taking medicine, it does fall under the category of pervasive computing system. It also uses the advantage of mobility of cell phone, importance of a hand-held device, use of powerful sensor like camera to create a cell based application, which manages and processes information. Therefore, it can be viewed as a powerful information management system, which is handy and mobile. Lastly, it introduces the use of barcodes that is a very popular technology nowadays to use along with cell-based applications.

Recently many applications have been created that exploits the features of cell phones like camera and Wi-Fi along with the barcode technology to process information. However, the information management system developed in this work converges the impact of not only mobile-based application and the barcode technology but also the pervasive computing application into one. It shows the advantages of combining the different realms of technologies to handle information.

In this work, we develop a system that not only decodes information but also categorizes it. It gathers information from different sources including the 1D and 2D barcodes, from the user and user information repository and from the external catalogs or databases containing information of a specific domain. After gathering the information, it processes and presents it to the user in a meaningful way, which can help the user in decision-making. The best application of such system is in grocery shopping which assists the user to buy the items, what to buy and eat and what not, what combination of items can be eaten together and what can be the total price of all grocery items in the cart prior to purchasing. However, this is just one application; the same idea can be used when buying medicines also. Other than that, it also provides the application user to perform different operations based on the information encoded and its content type.

Moreover, the packages developed in this work include some Java classes and interfaces, which are reusable and can be used for a desktop application. It provides some APIs that are completely independent of the other code. One can also add new APIs to the existing system to use it in other software developments. For instance, one can come up with a new content type and formatting scheme as well as new sets of operations for that content type to extend the scope of work. The packages from both the mobile-based system and server-based system can lead to development of other newer applications in future.

5.2 Future work

Currently, the system holds the information associated with the barcode as long as the barcode is stored in the sdcard. Once the barcode is deleted, the information also wipes away. In future, there could be some mechanism devised to store the information efficiently either in the cell phone or in the external database.

Future work also includes extending the components for grocery shopping application to provide functions, which are dependent not only on content type but also on time. For instance, during the time of sickness special alerts can be sent to the user. The user database can have information about the different food items and medicine a user can have during sickness, what to eat, what combination can be harmful in illness, etc. There is a time dependency involved based on the time when the user is sick or in good health, the application can provide meaningful result to the user.

Moreover, in future the system can be enhanced to gather data from many independent external sources rather than only from user owned database or a company owned catalog. These external sources can include global repositories like Google health database for collecting medicine or food information, catalogs that contain data collected through web search results and other global databases containing information from different domains.

BIBLIOGRAPHY

[1] J. Gao, L. Prakash, and R. Jagatesan, “Understanding 2D-BarCode Technology and Applications in M-Commerce - Design and Implementation of A 2D Barcode Processing Solution,” *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International*, 2007, pp. 49-56.

[2] R. Adelmann, M. Langheinrich, and C. Flörkemeier, “Toolkit for Barcode Recognition and Resolving on Camera Phones – Jump Starting the Internet of Things - Google Search.”

[3] H. Kato and K. Tan, “First read rate analysis of 2D-barcodes for camera phone applications as a ubiquitous computing tool.,” *TENCON 2007 - 2007 IEEE Region 10 Conference*, 2007, pp. 1-4.

[4] J. Quero, C. Tarrida, J. Santana, V. Ermolov, I. Jantunen, H. Laine, and J. Eichholz, “Health Care Applications Based on Mobile Phone Centric Smart Sensor Network,” *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, 2007, pp. 6298-6301.

[5] M. Lopez-Nores, J. Pazos-Arias, J. Garcia-Duque, and Y. Blanco-Fernandez, “Monitoring medicine intake in the networked home: The iCabiNET solution,” *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, 2008, pp. 116-117.

[6] “Jolly Knowledge Center - Technical Information,” *http://www.jollytech.com/support/knowledge_center/technical_information.php#a16*.

[7] H. Kato and K. Tan, “Pervasive 2D Barcodes for Camera Phone Applications,” *Pervasive Computing, IEEE*, vol. 6, 2007, pp. 76-85.

[8] R. Adelmann, “Mobile Phone Based Interaction with Everyday Products - On the Go,” *Next Generation Mobile Applications, Services and Technologies, 2007. NGMAST '07. The 2007 International Conference on*, 2007, pp. 63-69.

[9] H. Kato and K. Tan, “2D barcodes for mobile phones,” *Mobile Technology, Applications and Systems, 2005 2nd International Conference on*, 2005, pp. 8 pp.-8.

[10] “What is Android? | Android Developers,” *http://developer.android.com/guide/basics/what-is-android.html*.

[11] “zxing - Google Code,” *http://code.google.com/p/zxing/*.

[12] “Barcode 101 » What Is A Barcode?,” *http://barcode.com.sg/blog/?p=4*.

[13] “Barcode - Wikipedia, the free encyclopedia,” *http://en.wikipedia.org/wiki/Barcode*.

[14] “Barcode Guide - AM Labels,” *http://www.amlabels.co.uk/Barcode_Guides#what-is-a-barcode*.

[15] Barnes, D.B., Bradshaw, J., Schott, T., and Wilson, “Two Dimensional Bar Coding,” 1999.

[16] M. Rohs, “Real-World Interaction with Camera Phones,” *Ubiquitous Computing Systems*, 2005, pp. 74-89.

[17] S. Wachenfeld, S. Terlunen, and Xiaoyi Jiang, “Robust recognition of 1-D barcodes using camera phones,” *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1-4.

[18] Hee Il Hahn and Joung Koo Joung, “Implementation of algorithm to decode two-dimensional barcode PDF-417,” *Signal Processing, 2002 6th International Conference on*, 2002, pp. 1791-1794 vol.2.

[19] E. Ohbuchi, H. Hanaizumi, and L.A. Hock, “Barcode Readers using the Camera Device in Mobile Phones,” *Proceedings of the 2004 International Conference on Cyberworlds*, IEEE Computer Society, 2004, pp. 260-265.

[20] “Adelmann: A Toolkit for Bar-Code-Recognition and-Resolving on Camera Phones–Jump Starting the Internet... - Google Scholar.”

[21] A. Madhavapeddy, R. Sharp, D. Scott, and E. Upton, “Using Camera-Phones to EnhanceInteraction,” *Proceedings of Ubiquitous Computing (Adjunct Proceedings: Demos), 2004.*, 2004.

[22] Teoh Chin Yew, S. Mazleena, and I. Subariah, “Spatial resource analysis of two dimensional barcodes,” *Innovations in Information Technology, 2008. IIT 2008. International Conference on*, 2008, pp. 421-425.

[23] Tsung-Yu Liu, Tan-Hsu Tan, and Yu-Ling Chu, “2D Barcode and Augmented Reality Supported English Learning System,” *Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on*, 2007, pp. 5-10.

[24] N. Fujimura and M. Doi, “Collecting students' degree of comprehension with mobile phones,” *Proceedings of the 34th annual ACM SIGUCCS conference on User services*, Edmonton, Alberta, Canada: ACM, 2006, pp. 123-127.

[25] J. Rekimoto and Y. Ayatsuka, “CyberCode: designing augmented reality environments with visual tags,” *Proceedings of DARE 2000 on Designing augmented reality environments*, Elsinore, Denmark: ACM, 2000, pp. 1-10.

[26] S. Lisa and G. Piersantelli, “Use of 2D Barcode to Access Multimedia Content and the Web from a Mobile Handset,” *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, 2008, pp. 1-3.

[27] J. McCune, A. Perrig, and M. Reiter, “Seeing-is-believing: using camera phones for human-verifiable authentication,” *Security and Privacy, 2005 IEEE Symposium on*, 2005, pp. 110-124.

[28] K.A. Siek, K.H. Connelly, Y. Rogers, P. Rohwer, D. Lambert, and J.L. Welch, “When Do We Eat? An Evaluation of Food Items Input into an Electronic Food Monitoring Application,” *Pervasive Health Conference and Workshops, 2006*, 2006, pp. 1-10.

[29] R. Damper, D. Garner, G. Jordan, A. Rahman, and C. Saunders, “A barcode-scanner aid for visually-impaired people,” *Engineering in Medicine and Biology Society, 1996. Bridging Disciplines for Biomedicine. Proceedings of the 18th Annual International Conference of the IEEE*, 1996, pp. 397-398 vol.1.

[30] J.M.R. Álamo, J. Wong, R. Babbitt, and C. Chang, “MISS: Medicine Information Support System in the Smart Home Environment,” *Proceedings of the 6th international conference on Smart Homes and Health Telematics*, Ames, IA, USA: Springer-Verlag, 2008, pp. 185-199.

[31] D. Weerasinghe, K. Elmufti, M. Rajarajan, and V. Rakocevic, “XML Security based Access Control for Healthcare Information in Mobile Environment,” *Pervasive Health Conference and Workshops, 2006*, 2006, pp. 1-6.

[32] M. Shanmugam, S. Thiruvengadam, A. Khurat, and I. Maglogiannis, “Enabling Secure Mobile Access for Electronic Health Care Applications,” *Pervasive Health Conference and Workshops, 2006*, 2006, pp. 1-8.

[33] P. Lanigan, A. Paulos, A. Williams, D. Rossi, and P. Narasimhan, “Trinetra: Assistive Technologies for Grocery Shopping for the Blind,” *Wearable Computers, 2006 10th IEEE International Symposium on*, 2006, pp. 147-148.

[34] “Free Online Barcode Generator: Create Barcodes for Free!,” *http://www.tec-it.com/online-demos/tbarcode/barcode-generator.aspx*.

[35] “Free Barcode Generator (with Multiple Formats) - by Barcoding Inc.,” *http://www.barcoding.com/upc/*.

[36] “QR Code Generator,” *http://zxing.appspot.com/generator/*.

[37] “BarcodeContents - zxing - A rough guide to standard encoding of information in barcodes - Google Code,” *http://code.google.com/p/zxing/wiki/BarcodeContents*.

APPENDIX

XMLSchema

FoodProductXMLSchema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="a">
    <xs:complexType mixed="true" />
  </xs:element>

  <xs:element name="C">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="I" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Ca">
    <xs:complexType mixed="true" />
  </xs:element>

  <xs:element name="e">
    <xs:complexType mixed="true" />
  </xs:element>

  <xs:element name="FP">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="NF" />
        <xs:element ref="C" />
        <xs:element ref="E" />
        <xs:element ref="P" />
        <xs:element ref="L" />
        <xs:element ref="O" />
        <xs:element ref="U" />
        <xs:element ref="T" />
      </xs:sequence>
      <xs:attribute name="id" type="xs:NMTOKEN" use="optional" />
    </xs:complexType>
  </xs:element>

  <xs:element name="I">
    <xs:complexType mixed="true">
      <xs:attribute name="id" type="xs:NMTOKEN" use="required" />
    </xs:complexType>
  </xs:element>

  <xs:element name="L">
  <xs:complexType mixed="true" />
</xs:element>

<xs:element name="M">
  <xs:complexType mixed="true">
    <xs:choice>
      <xs:element ref="q" />
    </xs:choice>
    <xs:attribute name="id" type="xs:NMTOKEN" use="required" />
  </xs:complexType>
</xs:element>

<xs:element name="N">
  <xs:complexType mixed="true">
    <xs:choice>
      <xs:element ref="v" />
    </xs:choice>
    <xs:attribute name="id" type="xs:NMTOKEN" use="required" />
  </xs:complexType>
</xs:element>

<xs:element name="NF">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="s" />
      <xs:element ref="Ca" />
      <xs:element ref="N" maxOccurs="unbounded" />
      <xs:element ref="Vi" maxOccurs="unbounded" />
      <xs:element ref="M" maxOccurs="unbounded" />
      <xs:element ref="e" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="O">
  <xs:complexType mixed="true" />
</xs:element>
<xs:element name="E">
  <xs:complexType mixed="true" />
</xs:element>

<xs:element name="P">
  <xs:complexType mixed="true" />
</xs:element>

<xs:element name="q">
  <xs:complexType mixed="true" />
</xs:element>

<xs:element name="s">
  <xs:complexType mixed="true" />
</xs:element>

<xs:element name="T">
  <xs:complexType mixed="true" />
  </xs:element>

  <xs:element name="U">
    <xs:complexType mixed="true" />
  </xs:element>

  <xs:element name="v">
    <xs:complexType mixed="true" />
  </xs:element>

  <xs:element name="Vi">
    <xs:complexType mixed="true">
      <xs:choice>
        <xs:element ref="a" />
      </xs:choice>
      <xs:attribute name="id" type="xs:NMTOKEN" use="required" />
    </xs:complexType>
  </xs:element>

</xs:schema>
```