



UNIVERSITY CLASS PRE-REGISTRATION BY
WIRELESS APPLICATION PROTOCOL (WAP)

by

Mr. Nopparit Suptippayaratana

A Final Report of the Three - Credit Course
CE 6998 Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer and Engineering Management
Assumption University

April 2001

St. Gabriel's library, An

UNIVERSITY CLASS PRE-REGISTRATION BY WIRELESS APPLICATION PROTOCOL (WAP)

by
Mr. Nopparit Suptippayaratana

A Final Report of the Three-Credit Course
CE 6998 Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer and Engineering Management
Assumption University

April 2001


Project Title	University Class Pre-registration by Wireless Application Protocol (WAP)
Name	Mr. Nopparit Suptippayarattana
Project Advisor	Dr. Chamnong Jungthirapanich
Academic Year	April 2001


The Graduate School of Assumption University has approved this final report of the three-credit course, CE 6998 PROJECT, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer and Engineering Management.

Approval Committee:


(Dr. Chamnong Jungthirapanich)
Dean and Advisor


(Prof. Dr. Srisakdi Charmonman)
Chairman


(Asst. Prof. Dr. Boonmark Sirinaovakul)
Member


(Assoc. Prof. Somchai Thayarnyong)
MUA Representative

April 2001

ABSTRACT

The Wireless Application Protocol (WAP) is the de-facto world standard for the presentation and delivery of wireless information and telephony services on mobile phones and other wireless terminals. Users can connect to the WAP sites and find the information they need or doing some data processing process from their wireless device that makes them more convenient than using desktop PC. Therefore, this project is to develop the system to provide the new channel to do the class pre-registration process of university by wireless terminal devices.

The current existing class pre-registration System is based on the web and extranet application on workstation PC. Most of students that would like to do the registration process must connect their computers to the university's network, while some students must come to the university and use the university's computer. It requires expensive PC and complex procedure of computer configuration, and has to face the general problems of portability, which are inconvenience and having a high cost.

The new WAP System can be developed to both integrate and replace the existing system since this project scope only for wireless application and doesn't include the integration method to the existing system. All data will be kept in the server that uses Microsoft Access 2000 as database management system, and are accessed through by WAP server, Microsoft Internet Information Server 5.0 on Microsoft Windows 2000 Service Pack1. The user interfaces, moreover, are designed and implemented for WAP browser, mobile phones or wireless devices. It will reduce cost of user's computer and easier to do the configuration. It is also more portable than the PC system.

ACKNOWLEDGEMENTS

Several people have made contributions to this project. The writer would like to acknowledge their efforts and thank them for their contributions.

First, the writer would like to thank Dr. Chamnong Jungthirapanich, Dean and the writer's project advisor, for his valuable suggestions and advice given in to preparation of this project.

Next the writer would like to extend his sincere thanks to Mrs. Sansanee Niramis, his lovely friend, for her kindly assistance about the project format and information provided to the writer while doing his project.



TABLE OF CONTENTS

<u>Chapter</u>	<u>Page</u>
ABSTRACT	i
ACKNOWLEDGEMENTS	ii
LIST OF FIGURES	v
LIST OF TABLES	vi
I. INTRODUCTION	1
1.1 Objectives	2
1.2 Scope	3
II. LITERATURE REVIEW	4
2.1 Wireless Application Protocol (WAP)	4
2.2 Web Server and WAP Gateway	6
2.3 Wireless Markup Language (WML) and WML Script	8
2.4 Programming Editor Tools and Phone Simulator	11
2.5 Relational Database	13
2.6 Active Server Page (ASP)	18
III. SYSTEM DESIGN	20
3.1 System Specification	20
3.2 System Infrastructure	23
3.3 Interface Design	24
3.4 Database Design	25
IV. SYSTEM DEVELOPMENT	27
4.1 Code Development	27
V. SYSTEM IMPLEMENTATION	42

<u>Chapter</u>	<u>Page</u>
5.1 Microsoft Internet Information Service Configuration	43
5.2 Mobile Configuration	45
5.3 Database Setup	48
VI. SYSTEM EVALUATION	50
6.1 Evaluation Method	50
6.2 Verification	51
6.3 Validation	53
VII. CONCLUSIONS AND RECOMMENDATIONS	55
7.1 Conclusions	55
7.2 Recommendations	56
7.3 Further Works	58
APPENDIX A THE SUPPORTED CHARACTER SET	60
APPENDIX B NOKIA WAP TOOL KIT 2.1 INSTALLATION	65
APPENDIX C NOKIA PHONE SIMULATOR INSTALLATION	67
BIBLIOGRAPHY	70

LIST OF FIGURES

Figure	Page
2.1 WAP Gateway and Web Server	7
2.2 Markup Language Migration	9
2.3 Example of Relational Database Table	13
2.4 Example of Key Navigation between Tables	16
3.1 Network Infrastructure	23
3.2 Interface Transition	24
3.3 Database Relationship	25
5.1 HTTP Headers Configuration	43
5.2 File Types Configuration	44
5.3 Service Ontion Setting	45
5.4 Connection Setting	45
5.5 Connection Profiles Setting	46
5.6 Profile Setting	46
5.7 Example Setting	47
6.1 NOKIA 7110	51
6.2 SIEMENS S35	52
6.3 Ericsson R380	53
A.1 Nokia Phone Simulator Installation Step 1	69
A.2 Nokia Phone Simulator Installation Step 2	69

LIST OF TABLES

<u>Table</u>	<u>Page</u>
2.1 Relational Database Vocabulary	14
3.1 Schedule Table Field's Definition	25
3.2 Prereg Table Field's Definition	26
3.3 User Table Field's Definition	26
5.1 User Table Data	48
5.2 Schedule Table Data	48
5.3 PreReg Table Data	49



I. INTRODUCTION

Nowadays Technology comes to have much effect in human life. There are new technologies happening every day everywhere in the world. Wherever you are, the technology is always there. Many people try to adapt themselves to become familiar with the coming technologies day by day. They study and practice to have more skills then apply their skill to be an advantage in their work or life. Also the information system is another important for every people. Who can reach their required information faster who can take more benefit than others. So the technologies to manage and gather the information required are much more important than before. These include communication technology, information technology and information system management.

Every business has its own procedure and process to do data communication among employees, management and customers. There are many types of media and technologies for communication, for example, Email, Web, File Transfer, Internet application including WAP (Wireless Application Protocol). WAP is a new technology that allows the mobile phone users can use the Internet application across the network. Although WAP just launch in Thailand but there are so many people know that it is a breakthrough technology that can change everyone life.

Since the technology is changing rapidly, so people can gather any information and doing business activities easier. You can find everything or doing business application via the Internet from your desktop PC or notebook. But now with WAP technology, mobile phones can be more portable than computer PC. Since the number of mobile users is enormous and the price of mobile phone is cheaper than PC, so WAP technology can be the new channel of commerce to reach to the customer easier.

For this project, the writer will establish an application model by using WAP technology. Mobile users must connect to the WAP Gateway that is a carrier of connection between mobile network and Internet network to connect to WAP server. In this project, the writer will setup the WAP and Web server that will store the registration database and the application source code for his application, thus the writer have to connect his mobile phone to the WAP gateway such as AIS before the writer can access to internet. the writer will design WAP application interface for users to communicate with my application.

In conclusion for this project the writer separates it into three parts. First, the writer will set up WAP and Web server to be the place for storing his database and application. Second, the writer will design and develop the application to connect between mobile phone and server. Finally, the writer will develop the code for retrieving and storing the data into the database by ASP (Active Server Page) technology.

1.1 Objectives

The objectives of this project are list below.

- (1) To establish the application model with WAP and ASP Technology to handle the University Class Pre-Registration by mobile phones and desktop computers.
- (2) To establish the database with relational database management system to do the University Class Pre-Registration by mobile phones and desktop computers.
- (3) To provide interactive application for mobile users and desktop computer to do the pre-registration.

1.2 Scope

This project focuses on the real working of WAP application for registration on mobile phone. A few user account and available course data will be setup to do the testing. These data will be chosen from the real data of CEM students and CEM courses.



II. LITERATURE REVIEW

2.1 Wireless Application Protocol (WAP)

The Wireless Application Protocol (WAP) is the de-facto world standard for the presentation and delivery of wireless information and telephony services on mobile phones and other wireless terminals. It combines pertinent technical considerations, such as limited bandwidth, display sizes, and memory, with existing know-how, including HTTP, XML, and scripting (Wireless Internet Today June 2000).

Wireless Application Protocol (WAP) is a result of continuous work to define an industry wide standard for developing applications over wireless communication networks. The WAP Forum is an industry group dedicated to the goal of enabling sophisticated telephony and information services on handheld wireless devices, originally founded by Ericsson, Motorola, Nokia, and Unwired Planet. WML was formed to create the global wireless protocol specification that works across differing wireless network technology types, for adoption by appropriate industry standards bodies.

The WAP Forum's membership roster now includes computer industry heavyweights such as Microsoft, Oracle, IBM, and Intel along with several hundred other companies. According to the WAP Forum, the goals of WAP are to be:

- (1) To bring Internet content and advanced data services to wireless phones and other wireless terminals.
- (2) To create a global wireless protocol specification that works across all wireless network technologies.
- (3) To enable the creation of content and applications that scale across a wide range of wireless bearer networks and device types.

- (4) To embrace and extend existing standards and technology wherever possible and appropriate (Wireless Internet Today June 2000).

As part of the Forum's goals, WAP will also be accessible to (but not limited to) the following:

- (1) GSM-900, GSM-1800, GSM-1900
- (2) CDMA IS-95
- (3) TDMA IS-136
- (4) 3G systems - IMT-2000, UMTS, W-CDMA, Wideband IS-95

WAP defines a communications protocol as well as an application environment. In essence, it is a standardized technology for cross-platform, distributed computing. It sound similar to the World Wide Web. WAP is very similar to the combination of HTML and HTTP except that it adds in one very important feature: optimization for low-bandwidth, low-memory, and low-display capability environments. These types of environments include PDAs, wireless phones, pagers, and virtually any other communications device (Wireless Internet Today June 2000).

From a certain viewpoint, the WAP approach to content distribution and the Web approach are virtually identical in concept. Both concentrate on distributing content to remote devices using inexpensive, standardized client software. Both rely on back-end servers to handle user authentication, database queries, and intensive processing. Both use markup languages derived from SGML for delivering content to the client. In fact, as WAP continues to grow in support and popularity, it is highly likely that WAP application developers will make use of their existing Web infrastructure (in the form of application servers) for data storage and retrieval.

WAP (and its parent technology, XML) will serve to highlight the Web's status as the premier n-tier application in existence today. WAP allows a further extension of this

concept as existing "server" layers can be reused and extended to reach out to the vast array of wireless devices in business and personal use today. Note that XML, as opposed to HTML, contains no screen formatting instructions; instead, it concentrates on returning structured data that the client can use as it sees fits.

2.2 Web Server and WAP Gateway

A web server is a software program dedicated to retrieving web pages upon request. A Web Server also denotes the physical location of a computer that stores web documents so that we can download them on request. Web server software performs an interesting job; it simply runs through an endless loop on the host computer "listening" for requests from web browsers. When a browser request comes in, it springs to action. The server locates the requested document and copies it first to the network connection then back to the requesting web browser, then it goes back to listening again. A web server services thousands of clients in a day.

The wonderful thing about WAP content is that to be able to serve up wireless content to everyone in the world, all needs to do is add MIME types to web server configuration. Specific file types are used when creating wireless content, and simply have to identify those to the server.

A WAP gateway serves as a translator from one network to another. Wireless content delivery uses two networks: a wireless carrier network and the Internet. When the WAP sites wants their users to be able to see their WAP content from mobile phones that use Sprint PCS wireless data service. The site needs to be sure that web server, which is open to the Web via TCP/IP, can serve up content wirelessly via WAP. The gateway is the agent that allows WAP and TCP/IP to talk to each other.

The client is wireless device such as mobile phone that needs the content from the WAP site. The client must connect itself to the WAP gateway to initiate the connection

and prepare the session to access though information on the WAP sites. The following picture shows the process of content providing (Thaiwapportal Online 2001).

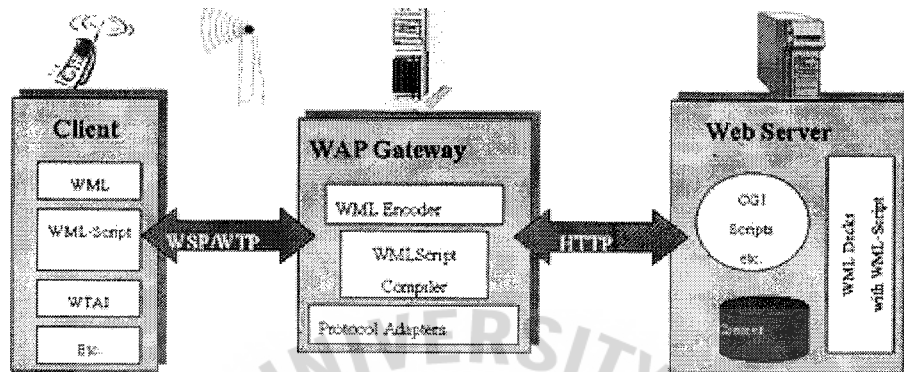


Figure 2.1. WAP Gateway and Web Server.

A WAP request is routed through a WAP gateway which acts as an intermediary between the "bearer" used by the client (GSM, CDMA, TDMA, etc.) and the computing network that the WAP gateway resides on (TCP/IP in most cases). The gateway then processes the request, retrieves contents or calls CGI scripts, Java servlets, or some other dynamic mechanism, then formats data for return to the client. This data is formatted as WML (Wireless Markup Language), a markup language based directly on XML. Once the WML has been prepared (known as a deck), the gateway then sends the completed request back (in binary form due to bandwidth restrictions) to the client for display and/or processing. The client retrieves the first card off of the deck and displays it on the monitor. (Thaiwapportal Online 2001)

To sum up, the client makes a request. This request is received by a WAP gateway that then processes the request and formulates a reply using WML. When ready, the WML is sent back to the client for display. As mentioned earlier, this is very

similar in concept to the standard stateless HTTP transaction involving client Web browsers.

2.3 Wireless Markup Language (WML) and WML Script

WML (Wireless Markup Language) is a markup language based on XML, and is intended for use in specifying content and user interface for narrowband devices, including cellular phones and pagers. WML is designed with the constraints of small narrowband devices in mind. These constraints include:

- (1) Small display and limited user input facilities;
- (2) Narrowband network connection;
- (3) Limited memory and computational resources.

WML includes four major functional areas:

- (1) Text presentation and layout - WML includes text and image support, including a variety of formatting and layout commands;
- (2) Deck/card organizational metaphor - all information in WML is organised into a collection of cards and decks;
- (3) Inter-card navigation and linking - WML includes support for explicitly managing the navigation between cards and decks;
- (4) String parameterization and state management - all WML decks can be parameterised, using a state model.

While HTML predefines a "canned" set of tags guaranteed to be understood and displayed in a uniform fashion by a Web browser, XML allows the document creator to define any set of tags he or she wishes to. This set of tags is then grouped into a set of grammar "rules" known as the Document Type Definition, or DTD. If a phone or other communications device is said to be WAP-capable, this means that it has a piece of

software loaded onto it (known as a micro browser) that fully understands how to handle all entities in the WML 1.2 DTD (Wireless Developer Network 2000).

Markup language migration

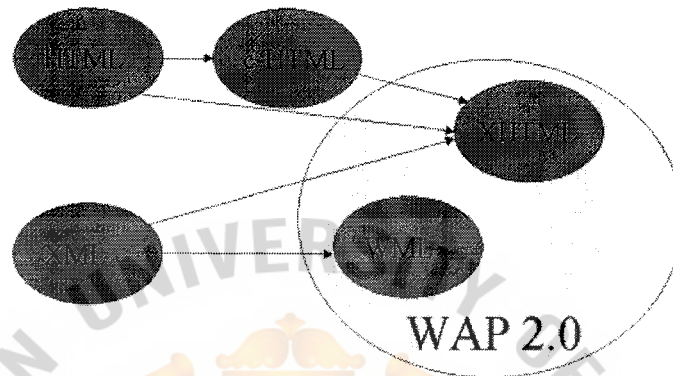


Figure 2.2. Markup Language Migration.

WML was designed for low-bandwidth, small-display devices. As part of the design, the concept of a deck of cards was utilized. A single WML document (i.e. the elements contained within the <wml> document element) is known as a deck. A single interaction between a user agent and a user is known as a card. The beauty of this design is that multiple screens can be downloaded to the client in a single retrieval. Using WMLScript, user selections or entries can be handled and routed to already loaded cards, thereby eliminating excessive transactions with remote servers. Of course, with limited client capabilities comes another trade off. Depending on your client's memory capabilities, it may be necessary to split multiple cards up into multiple decks to prevent a single deck from becoming too large (Wireless Developer Network 2000).

The deck of cards metaphor is designed specifically to take advantage of small display areas on handheld devices. Instead of continually requesting and retrieving cards

(the WAP equivalent of HTML pages), each client request results in the retrieval of a deck of one or more cards. The client device can employ logic via embedded WMLScript (the WAP equivalent of client-side JavaScript) for intelligently processing these cards and the resultant user inputs (Wireless Developer Network 2000).

WMLScript is the WAP corollary to the JavaScript scripting language that was popularized by Netscape Communications. Standardization efforts by Netscape helped produce the ECMAScript standard, a standard that WMLScript was based on. While JavaScript has since been coopted by server tool vendors (including Netscape and Microsoft), WMLScript is a client-only scripting platform used in combination with WML to provide client side procedural logic. Like WML, WMLScript is compiled via a WAP gateway into binary form to provide intelligence to mobile clients.

WMLScript syntax is based on the ECMAScript programming language. Unlike ECMAScript, however, the WMLScript specification also defines a bytecode and interpreter reference architecture for optimal utilization of current narrowband communications channels and handheld device memory requirements. The following help summarize some basic syntactical features of the language:

- (1) The smallest unit of execution in WMLScript is a statement and each statement must end with a semicolon (;).
- (2) WMLScript is case-sensitive.
- (3) Comments can either be single-line (beginning with //) or multi-line (bracketed by /* and */). This syntax is identical to both C++ and Java.
- (4) A literal character string is defined as any sequence of zero or more characters enclosed within double (") or single (') quotes.
- (5) Boolean literal values correspond to true and false.
- (6) New variables are declared using the var keyword (i.e. var x;)

2.4 Programming Editor Tools and Phone Simulator

The WAP programming tool offers developers a PC environment for creating, testing and demonstrating WAP applications. It includes tools for creating WML and WMLScript content, adding WBMP graphics, as well as debugging and simulating WAP applications on WAP enabled handsets. This release provides application developers with their first chance to take advantage of new push functionality. A reference implementation of the WAP Forum's June 2000 Specifications is also included (Nokia Corporation 2001).

WAP application developers can create complete applications without even a handset or access to carrier infrastructure. Testing and demonstrating WAP applications is straightforward, and developers can navigate and request URLs on any WAP gateway or on any web server on the Internet. In addition, applications can be stored and queried directly from the PC file system (Nokia Corporation 2001).

There are so many tools from several software vendors that are available today. Nokia also provides the free toolkit for WAP developer to develop the WAP application. The Nokia WAP Toolkit is used in this project development and it is a good tool that offers a development environment for people who want to provide Internet services and content for mobile terminals. Everyone can download the latest version of the Nokia WAP Toolkit for free at <http://www.forum.nokia.com>.

Nokia WAP Toolkit 2.1 Key feature.

- (1) WML and WMLScript editors for creating and editing static WML and WMLScript content
- (2) Expanded functionality with WAP Forum June 2000 specifications, including support for cache operations and multi-part push services

- (3) Device simulators, including Nokia phone simulators for displaying WAP content and a server simulator for use when a WAP gateway is unavailable
- (4) Debugging and testing tools showing detailed error information and location
- (5) Wireless Bitmap graphics editor for creating WBMPs or converting from GIF or JPEG graphic formats
- (6) Several sample applications in source code format to use as a development starting point

Nokia WAP Toolkit 2.1 Components

- (1) WML, WMLScript, WBMP and multi-part editors
- (2) Phone simulators
- (3) WAP Forum June 2000 specifications-based WML Browser in Reference Implementation.
- (4) Script interpreter and run-time libraries
- (5) Debugging tools
- (6) Sample applications and Java Servlets
- (7) Push simulator

System Requirements for Nokia WAP Toolkit 2.1

Installation of Java Runtime Environment (JRE) 1.3

Pentium class 233MHz or faster (300MHz recommended)

128MB RAM (256MB recommended)

Windows NT 4.0 with SP4 or Windows 98/2000

64K color with 800x600 resolution

100MB of hard disk space (Nokia Corporation 2001).

2.5 Relational Database

The relational database model represents data in the form of two-dimension tables. Each table in the represents some real-world person, place, thing, or event about which information is collected. A relational table is a flat file composed of a set of named columns and an arbitrary number of unnamed rows. The columns of the tables contain information about the table. The rows of the table represent occurrences of the "thing" represented by the table. A data value is stored in the intersection of a row and column. Each named column has a domain, which is the set of values that may appear in that column. The following figure shows the relational tables for a simple bibliographic database that stores information about book title, authors, and publishers (Academic Computing and Instructional Technology Services (ACITS) 2000).

A Relational Database					
AUTHOR					
au_id	au_lname	au_fname	address	city	state
172-32-1176	White	Johnson	10932 Bigge Rd.	Menlo Park	CA
213-46-8915	Green	Marjorie	309 63rd St..411	Oakland	CA
238-95-7766	Carson	Cheryl	589 Darwin Lrt.	Berkeley	CA
267-41-2394	O'Leary	Michael	22 Cleveland Av. .14	San Jose	CA
274-80-9391	Straight	Dean	5420 College Av.	Oakland	CA
341-22-1782	Smith	Meander	10 Mississippi Dr.	Lawrence	KS
409-56-7008	Bennet	Abraham	6223 Bateman St.	Berkeley	CA
427-17-2319	Dull	Ann	3410 Blonde St.	Palo Alto	CA
472-27-2349	Gringlesby	Burt	PO Box 792	Covelo	CA
486-29-1786	Locksley	Charlene	18 Broadway Av.	San Francisco	CA

TIT L E				
title_id	title	type	price	pub_id
BU1032	The Busy Executive's Database Guide	business	19.99	1389
BL1111	Cooking with Computers	business	11.95	1389
BU2075	You Can Combat Computer Stress!	business	2.99	736
BU7832	Straight Talk About Computers	business	19.99	1389
MC2222	Silicon Valley Gastronomic Treats	mod_cook	19.99	877
MC3021	The Gourmet Microwave	mod_cook	2.99	877
MC3026	The Psychology of Computer Cooking	UNDECIDED		877
PC1035	But Is It User Friendly ?	popular_comp	22.95	1389
PC8888	Secrets of Silicon Valley	popular_comp	20	1389
PC9999	Net Etiquette	popular_comp		1389
PS2091	Is Anger the Enemy?	psychology	10.95	736

PUBLISHER		
pub...Ad	pub.--name	city
736	New Moon Books	Boston
877	Binnet & Hardley	Washington
1389	Algodata Infosy sterns	Berkeley
1622	Five Lakes Publishing	Chicago
1736	Ramona Publishers	Dallas
9901	GGG&G	hilinchen
9952	Scootney Books	New York
9999	Lucerne Publishing	Paris

AUTHOR TITLE	
au_id	
172-32-1176	P53333
213-46-8915	BU1032
213-46-8915	8U2075
238-95-7766	PC1035
267-41-2394	BU1111
267-41-2394	TC7777
274-80-9391	BU7832
409-56-7008	BU1032
427-17-2319	PC8888
472-27-2349	TC7777

Figure 2.3. Example of Relational Database Table.

2.5.1 Table

Table is a set of rows and columns. This is very important, because a set does not have any predefined sort order for its elements. Each row is a set of columns with only one value for each. All rows from the same table have the same set of columns, although some columns may have NULL values, i.e. the values for that rows was not initialized. Note that a NULL value for a string column is different from an empty string (Academic Computing and Instructional Technology Services (ACITS) 2000).

There are alternate names used to describe relational tables. Some manuals use the terms tables, fields, and records to describe relational tables, columns, and rows, respectively. The formal literature tends to use the mathematical terms, relations, attributes, and tuples. The following table summarizes these naming conventions (Academic Computing and Instructional Technology Services (ACITS) 2000).

Table 2.1. Relational Database Vocabulary.

In This Document	Formal Terms	Many Database Manuals
Relational Table	Relation	Table
Column	Attribute	Field
Row	Tuple	Record

2.5.2 Relationship

A relationship is an association between two or more tables. Relationships are expressed in the data values of the primary and foreign keys (Academic Computing and Instructional Technology Services (ACITS) 2000).

A **primary key** is a column or columns in a table whose values uniquely identify each row in a table. A **foreign key** is a column or columns whose values are the same as

the primary key of another table. You can think of a foreign key as a copy of primary key from another relational table. The relationship is made between two relational tables by matching the values of the foreign key in one table with the values of the primary key in another (Academic Computing and Instructional Technology Services (ACITS) 2000).

Keys are fundamental to the concept of relational databases because they enable tables in the database to be related with each other. Navigation around a relational database depends on the ability of the primary key to unambiguously identify specific rows of a table. Navigating between tables requires that the foreign key is able to correctly and consistently reference the values of the primary keys of a related table. For example, the figure below shows how the keys in the relational tables are used to navigate from AUTHOR to TITLE to PUBLISHER. AUTHOR TITLE is an all key table used to link AUTHOR and TITLE. This relational table is required because AUTHOR and TITLE have a many-to-many relationship (Academic Computing and Instructional Technology Services (ACITS) 2000).

Navigating Between Tables Using Keys

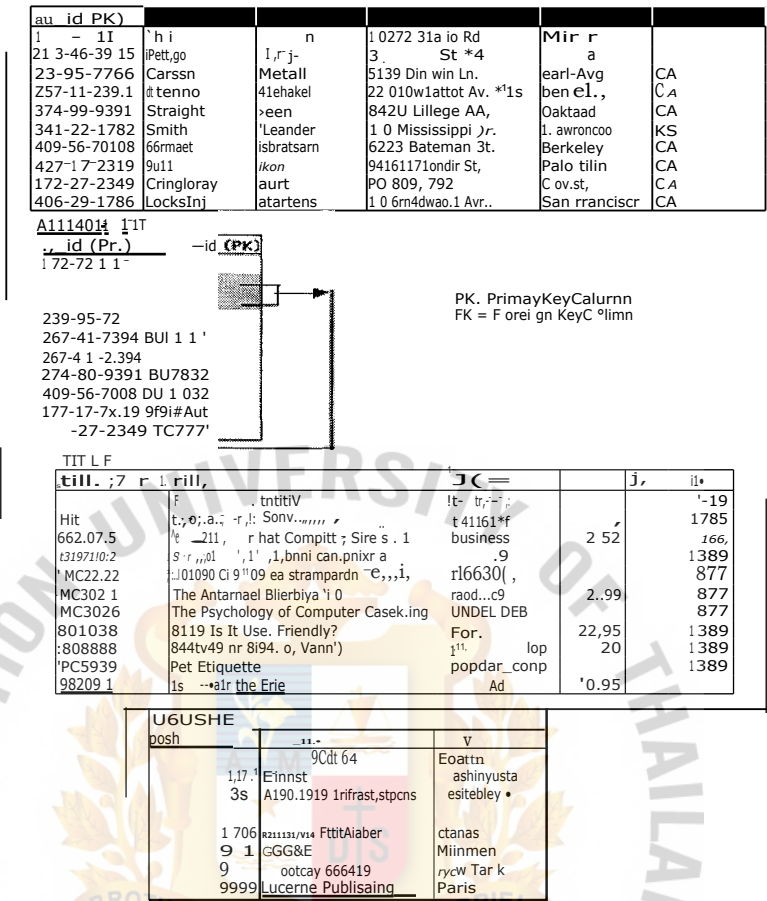


Figure 2.4. Example of Key Navigation between Tables.

2.5.3 Data Integrity

Data integrity means, in part, that you can correctly and consistently navigate and manipulate the tables in the database. There are two basic rules to ensure data integrity; entity integrity and referential integrity.

The entity integrity rule states that the value of the primary key, can never be a null value (a null value is one that has no value and is not the same as a blank). Because a primary key is used to identify a unique row in a relational table, its value must

always be specified and should never be unknown. The integrity rule requires that insert, update, and delete operations maintain the uniqueness and existence of all primary keys.

The referential integrity rule states that if a relational table has a foreign key, then every value of the foreign key must either be null or match the values in the relational table in which that foreign key is a primary key (Academic Computing and Instructional Technology Services (ACITS) 2000).

2.5.4 Normalization

Normalization is a design technique that is widely used as a guide in designing relational databases. Normalization is essentially a two step process that puts data into tabular form by removing repeating groups and then removes duplicated from the relational tables.

Normalization theory is based on the concepts of normal forms. A relational table is said to be a particular normal form if it satisfied a certain set of constraints. There are currently five normal forms that have been defined. In this section, we will cover the first three normal forms that were defined by E. F. Codd (Academic Computing and Instructional Technology Services (ACITS) 2000).

The goal of normalization is to create a set of relational tables that are free of redundant data and that can be consistently and correctly modified. This means that all tables in a relational database should be in the third normal form (3NF). A relational table is in 3NF if and only if all non-key columns are (a) mutually independent and (b) fully dependent upon the primary key. Mutual independence means that no non-key column is dependent upon any combination of the other columns. The first two normal forms are intermediate steps to achieve the goal of having all tables in 3NF. In order to better understand the 2NF and higher forms, it is necessary to understand the concepts

of functional dependencies and lossless decomposition (Academic Computing and Instructional Technology Services (ACITS) 2000).

2.6 Active Server Page (ASP)

Microsoft Active Server Pages (ASP) is a server-side scripting environment that you can use to create and run dynamic, interactive Web server applications. With ASP, you can combine HTML pages, script commands, and ActiveX components to create interactive Web pages or powerful Web-based applications. ASP applications are easy to develop and modify.

An ASP script begins to run when a browser requests an .asp file from your Web server. Your Web server then calls ASP, which reads through the requested file from top to bottom, executes any script commands, and sends a Web page to the browser.

Because your scripts run on the server rather than on the client, your Web server does all the work involved in generating the Web pages that you send to browsers. You need not worry whether a browser can process your scripts: your Web server does all the script processing, transmitting standard HTML to the browser. Server-side scripts cannot be readily copied because only the result of the script is returned to the browser. Your users cannot view the script commands that created the page they are viewing. Active Server Pages have a couple of other things going for it. First of all, if you're working on an NT platform and using IIS, ASP and ASP's most useful components are free. ASP is also reasonably language-neutral. You can write your ASP code in the scripting language of your choice: the default is VBScript (although that can be changed via IIS), but if you prefer, you can use JavaScript or PerlScript by simply declaring it at the beginning of your code. That being said, we'll use VBScript in this tutorial.

One of the main reasons folks are so interested in ASP (or any server-side scripting language for that matter) is database manipulation and file access. ASP is

particularly strong in these arenas, with ActiveX controls such as ADO providing some nice tools for manipulating outside data sources. We'll play a bit with the File Access component later in this article.

ASP is also extensible. Components are modules that add specific areas of functionality to ASP, and, although IIS comes with several useful prefabricated components, more can be obtained from third-party vendors or written yourself with anything from Visual Basic to C++ to Java, if you have the know-how. As of late ASP components have become a hot business on the web. If there's a something you want ASP to do, and you aren't able to make it happen with the standard components, it's a fair bet that you can find one on the web that will do the trick or write one yourself.



III. SYSTEM DESIGN

3.1 System Specification

To provide the registration's service for the wireless terminal users, The system must be available to the user all the time that causes our system to have the permanent network media to connect to the public network that allow the users from everywhere to access through the server from cellular network all the time. The user who would like to do the registration process must connect their mobile or wireless terminal that supports WAP to WAP gateway. This project don't include the part of WAP gateway deployment but we leave the process of setup gateway to the WAP provider such as Nokia, AIS or TAC the big telecom company in Thailand. All we need to do is to setup the system that must handle all process to communicate with the WAP gateway.

The project is simulated for doing the class pre-registration process of the CEM students so the system won't gather all data from the production system. The data will be sampling from the actual system and have a smaller size. Anyway the system must work in the real environment and can be used in the real work if the real data are stored in the system.

Another important thing that scopes out from this project is that the project doesn't consider any process of doing the security's control. The system seems to be the guideline for others to extend the development to be use in real situations that they can apply and appropriate security control methods to ensure that system and data will be protected from any treats.

Hardware Specification

(1) Web Server

Intel Pentium III 866 MHz

SDRAM Hitachi 256 MByte/ 133 MHz

Main board Asus CUSL-2

Hard disk Quantum 20 GB

Ethernet card SMC 10/100

Monitor 15"

Standard Keyboard/Mouse

CDROM 50X

(2) Development Workstation

Intel Pentium III 733 MHz

SDRAM Hitachi 256 Mbyte/ 133 MHz

Main board Asus CUSL-2

Hard disk Quantum 20 GB

Ethernet card SMC 10/100

Monitor 15"

Standard Keyboard/Mouse

CDROM 50X

(3) Network Hub

Accton EH3016a-sw

(4) Network Router

Cisco 1601R

(5) Network Cable for Local Area Network

AMP Cat5 cable with AMP CatS Outlet

- (6) For Wide Area Network

Lease Line 256 Kbps to ISP

- (7) Nokia 7110 mobile phone

Software Specification

- (1) Server Software

Window 2000 Advance Server

Internet Information Server 5.0

- (2) Development Workstation Software

Window 2000 Professional

Microsoft Office 2000 Professional

Nokia WAP Toolkit 2.0

Internet Explorer 5.5

- (3) Nokia 7110 Firmware v5.0



3.2 System Infrastructure

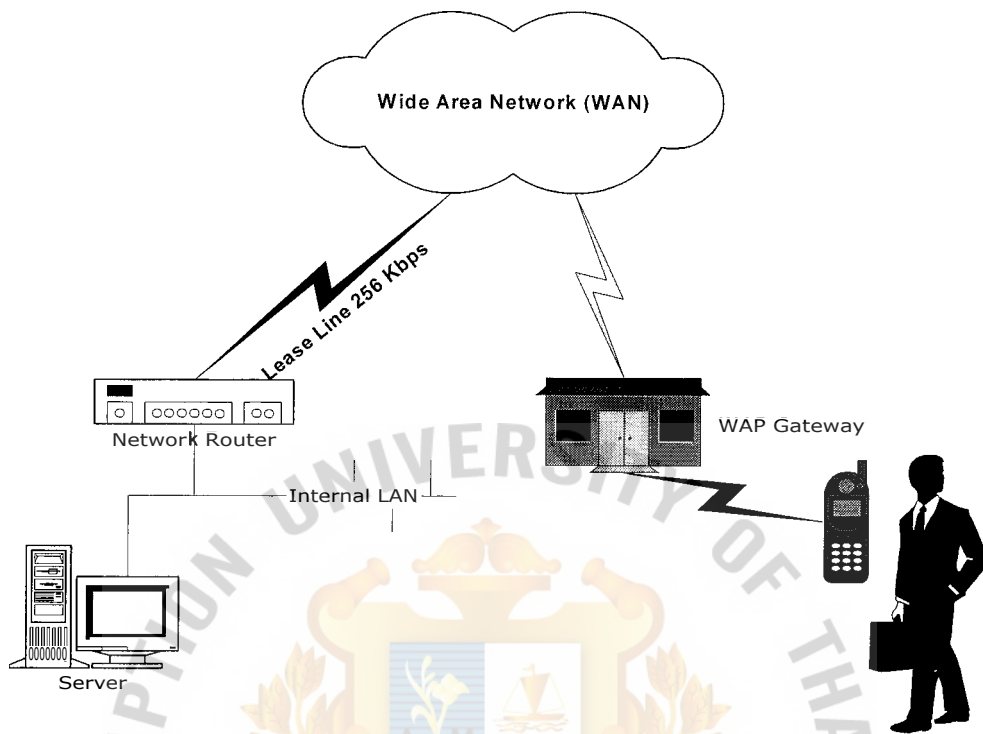


Figure 3.1. Network Infrastructure.

The infrastructure seem very simple compose one server that connects to the Internet by permanent lease line connection to the Internet Service Provider though the public network. WAP users can connect their mobile to the WAP providers such as AIS, Nokia or MWEB to use the WAP provider's system as gateway to our system.

The primary function of the server is providing the WEB application and protocol to the user. They are installed with Windows 2000 Advance Server and Internet Information Server 5.0 to use as WEB server and have an additional configuration for support WAP. This is very easy to implement if we already have the WEB server in our network. All we do need to do is adding the MIME message type in the server. This process will be shown in the appendix part of this document.

3.3 Interface Design

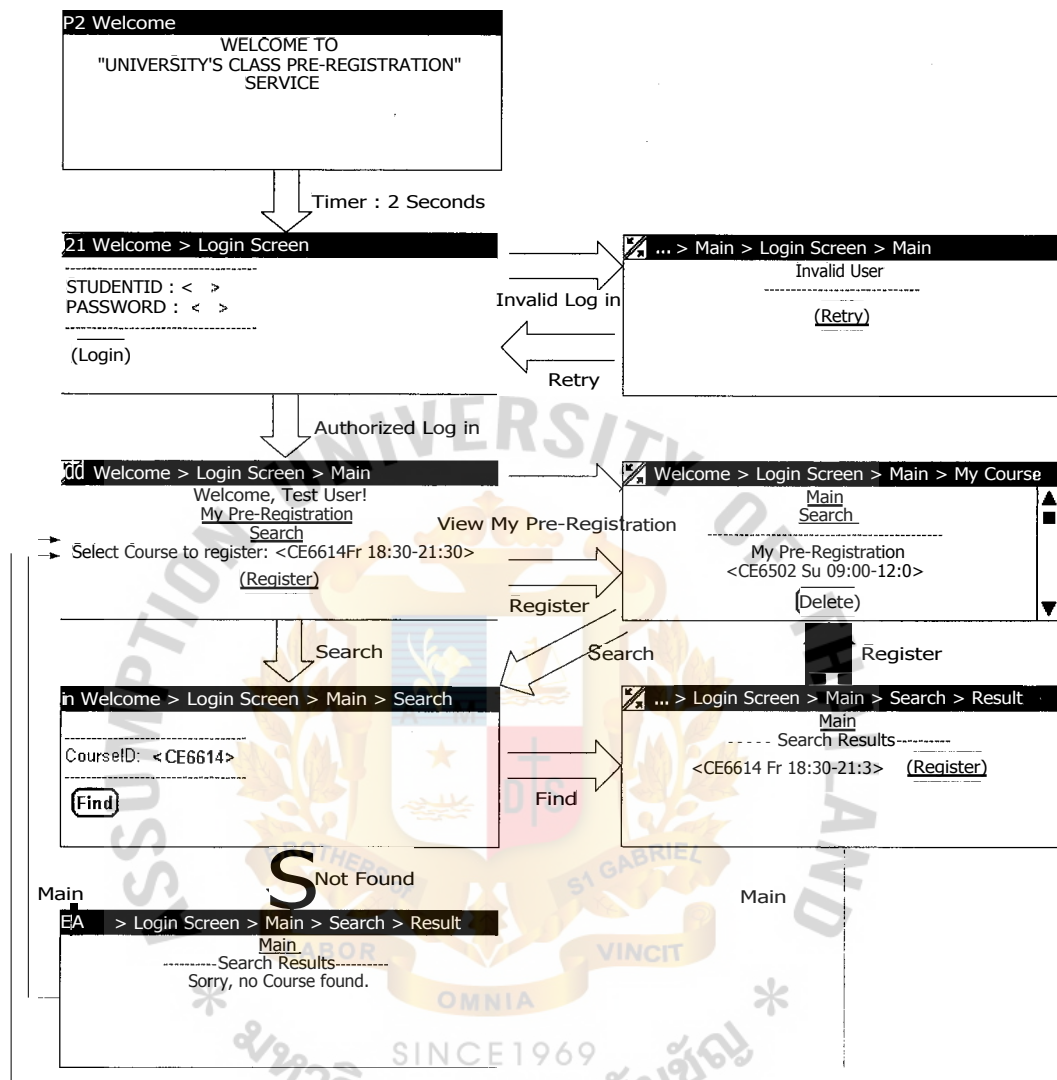


Figure 3.2. Interface Transition.

3.4 Database Design

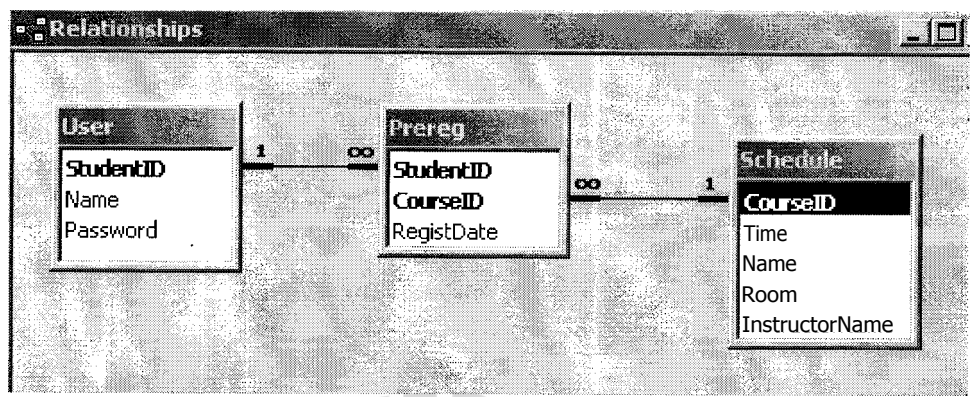


Figure 3.3. Database Relationship.

The server maintains a database named REGISTER.MDB, containing the following tables: USER, PREREG and SCHEDULE. The relationship for these tables is as above.

Table 3.1. Schedule Table Field's Definition.

Schedule : Table			
	Field Name	Data Type	Description
	CourseID	Text	Field Size = 7; Required;
	Time	Text	Field Size = 14;
	Name	Text	Field Size = 40;
	Room	Text	Field Size = 4;
	InstructorName	Text	Field Size = 40;

The SCHEDULE table contains CourseID, time schedule of the course and its name, as well as room and instructor name. CourseID is the unique identifier. The CourseID of the same course which has different time schedules, room or instructor are different for example the CourseID of CE6614 subject that have the different instructor and room may be CE6614A and CE6614B.

Table 3.2. Prereg Table Field's Definition.

Field Name	Data Type	Description
StudentID	Text	Field Size = 8; Required;
CourseID	Text	Field Size = 7; Required;
RegistDate	Text	Field Size = 10;

The PREREG table is used to capture the registration records made by the university's student. It use two fields StudentID and CourseID to identify themselves. The Registdate is the third field that is the production of the automatically process of calculating the present date when the registration process is done.

Table 3.3. User Table Field's Definition.

Field Name	Data Type	Description
StudentID	Text	Field Size = 8; Required;
Flame	Text	Field Size = 30;
Password	Text	Field Size = 8; Required;

The USER table contains student login information, password and the name of the university's student. The student login is the university's student identification number that is limit to maximum 8 characters same as the password. The name field is the actual name of each student that is combined from first and last name.

IV. SYSTEM DEVELOPMENT

4.1 Code Development

As stated in the literature review, WAP application is written by WML that will separate the program into several decks and each deck composed of many cards. So the program is quite easy to understand if the writer just describes the program deck by deck. One deck means one file in physical storage that is the way the writer will describe.

The program is separated into four ASP files and one database file:

- (1) Index.asp
- (2) Login.asp
- (3) Course.asp
- (4) Search.asp
- (5) Register.mdb

The writer will describe them more deeply detail in the next paragraph that will make them more clearly and easier understandable.

Filename: Index.asp

Description:

The first page when you start to browse to the application. It shows the welcome screen and delays the welcome screen for 2 second before automatically change the screen to the login screen. So it means that this file or we call it deck, compose of 2 cards. First is Welcome card that must be shown and delay for 2 second and then link to the second card names LoginScr automatically. The LoginScr card have two input text boxs for getting the StudentID and Password from the user, keep it in the temporary

variable and then pass the value to the next file (Login.asp) when the user select the login button.

```
<% Response.ContentType = "text/vnd.wap.wml" %>
```

```
<?xml version="1.0"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1 //EN"
```

```
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
```

```
<!-- Welcome Screen -->
```

```
<card id="Welcome" title="Welcome" ontimer="#LoginScr">
```

```
  <timer value="20"/>
```

```
  <p align="center">
```

```
    WELCOME TO <b>"UNIVERSITY'S CLASS PRE-REGISTRATION"
```

```
<br/>SERVICE
```

```
  </p>
```

```
</card>
```

```
<!-- Login Screen -->
```

```
<card id="LoginScr" title="Login Screen">
```

```
  <p align="left">
```

```
    -----
```

```
  <br/>
```

```
  STUDENTID : <input name="StudentID" type="text" maxlength="8" />
```

```
  <br/>
```

```
  PASSWORD : <input name="Password" type="password" maxlength="8" />
```

<do type="accept" label="Login">

<go href="Login.asp" method="post">

<postfield name="StudentID" value="\$StudentID" />

<postfield name="Password" value="\$Password" />

</go>

</do>

<p>

</card>

</wml>

Filename: Login.asp

Description:

Once the student has keyed in his/her StudentID and password and pass the value of both field into this deck, the application will proceed to authenticate him/her.

If the student fails the authentication, the screen shows the error message to inform the student.

Once the user is authenticated, the ASP script will proceed to generate the list of course available and let the user select the course that they required to register and click on the register button to start the registration process in the other file (course.asp) by passing the value of selected CourseID.

Another parts of this card are, the writer used the <anchcr> element to create a hyperlink "My Pre-Registration" to link to the other deck (course.asp) to view the content of the current pre-registration course. As for the list of course available in the schedule table, the writer used the <select> and <option> elements to display every record in the schedule table.

The application also has one most powerful function to search for the course required by use another hyperlink "Search" that leads the user to the search function in the other file (search.asp).

In this application the writer has squeezed quite a fair bit of information into a single line. On the emulator, long lines are wrapped. However, you must note that different WAP browsers have different treatment for long lines and may cropped off all excess characters that cannot fit on the same line. In a nutshell, where possible limit the line width to about 12 characters and also consider the target platform that your application would most likely be run on.

```
<% Response.ContentType = "text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml 1.1.xml">
```

```
<wml>
```

```
<%
```

```
Dim conn, rs
```

```
Set conn = Server.CreateObject("ADODB.Connection")
```

```
Set rs = Server.CreateObject("ADODB.Recordset")
```



```

conn.open "DRIVER= {Microsoft Access Driver (*.mdb)} ;DBQ=" &
Server.MapPath("Register.mdb") & ";

%>

<!-- Login Card -->

<card id="Login" title="Main">

    <p align="center">

        sqlQuery = "SELECT * FROM User WHERE StudentID=" &
Request.Form("StudentID") & " AND Password=" & Request.Form("Password") & ""

        set rs = conn.Execute(SQLquery)

        if rs.EOF then
            Response.Write "Invalid User <br/>"
            Response. Write "-----<br/>"
            Response.Write "<do type='accept' label='Retry'>"
            Response. Write " <go href=index.asp#LoginScr'/>"
            Response.Write "</do>"
        else
            Session("StudentID") = Request.Form("StudentID")
            Session("Password") = Request.Form("Password")
            Response.Write "<strong>Welcome, " & rs.Fields("Name") & "! </strong><br/>"
            sqlQuery = "SELECT * FROM Schedule"
            Set rs = conn.Execute(SQLquery)

        %>

        <anchor>

```

My Pre-Registration

```
<go href="course.asp" method="post"> <postfield name="StudentID"
value="$(StudentID)" /></go>
```

```
</anchor>
```

```
<br/>
```

```
<anchor>
```

Search

```
<go href="Search.asp" />
```

```
</anchor>
```

```
<br/>Select Course to register:
```

```
<select name="CourseID" value="<% =rs("CourseID") %>">
```

```
while not rs.EOF
```

```
• %> <option value="<% =rs("CourseID") %>">
```

```
<% =rs("CourseID") %> <% =rs("Time") %>-<% =rs("InstructorName") %>-
```

```
<% =rs("Name") %> </option>
```

```
<% rs.MoveNext
```

```
Wend
```

```
%>
```

```
</select>
```

```
<do type="accept" label="Register">
```

```
<go href="course.asp" method="post">
```

```
<postfield name="StudentID" value="$(Student1D)" />
```

```
<postfield name="CourseID" value="$(CourseID)" />
```

```
</go>
```

```
</do>
```

```
<%
```

```
end if
```

```
%>
```

```
</p>
```

```
</card>
```

```
</wml>
```

Filename: Course.asp

Description:

The user can browse into this page by doing one of these three ways. The first way is that the user clicks on the hyperlink "My Pre-Registration" on the main page then it will leads the user to this deck while the application will start to retrieve the registration records of the current user from the Prereg table in database. If there is no record found then the application will show the message "None!" to notify the user. If there is existing record, then it will display every record to the user within the tag <select> and <option> element.

The second way is, once the user select course available and click on the registration button in the Main card. The registration process will be started. The application will check on the database and insert the registration record into the Prereg

table if there is no existing record registered for the same user. The application will also calculate the present date and put it in the RegistDate field in the table while it insert the new record.

If the course is registered correctly, the content of the registered course are displayed in tag <select> and <option> element. The reason that the application displays the record list in these tags because of they allow the user to select the course that the user require to delete and click on the delete button to delete it easily. After deletion process has done the application will refresh itself effect this deck to be refreshed and display the updated information. So this is the third way to browse to this deck.

Another parts of this deck, the writer again uses anchor tag to link this card to another card "Main" and "Search".

```

<% Response.ContentType = "text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml 1.1.xml">
<wml>
<%
    Dim conn, rs

    Set conn = Server.CreateObject("ADODB.Connection")

    Set rs = Server.CreateObject("ADODB.Recordset")

    conn. open "DRIVER= {Microsoft Access Driver (*.mdb)} ;DBQ=" &
Server.MapPath("register.mdb") & ";"
%>

```

```
<template>
```

```
<do type="accept" label="Delete">
```

```
<go href="Course.asp" method="post">
```

```
<postfield name="StudentID" value="$StudentID" />
```

```
<postfield name="delete" value="$RCourse" />
```

```
</go>
```

```
</do>
```

```
</template>
```

```
<card id="Course" title="My Course">
```

```
<p align="center">
```

```
<%
```

```
ItemToDelete = Request.Form("delete")
```

```
if ItemToDelete<>"" then
```

```
'---Delete an item into the table---
```

```
sqlQuery = "DELETE FROM Prereg WHERE StudentID=" &  
Request.Form("StudentID") & " AND CourseID=" & Request.Form("delete") & ""
```

```
On Error Resume Next
```

```
set rs = conn.Execute(sqlQuery)
```

```
rs.Close
```

```
elseif Request.Form("CourseID ")<>" " then
```

```
'---insert an item into the table---
```

```
sqlQuery = "INSERT INTO Prereg (StudentID, CourseID, RegistDate) Values el &  
Request.Form("StudentID") & ",!" & Request.Form("CourseID") & ",!" & date & ""
```


On Error Resume Next '---prevent duplicate items from crashing my program---

set rs = conn.Execute(sqlQuery)

rs.Close

end if

%>

<anchor>

Main

<go href="login.asp" method="post">

<postfield name="StudentID" value="\$(StudentID)" />

<postfield name="Password" value="\$(Password)" />

</go>

</anchor>

<anchor>

Search

<go href="Search.asp" />

</anchor>

-----My Pre-Registration-----

<%

'---Displays the content---

'---Response.Write "
"

sqlQuery = "SELECT * FROM Prereg INNER JOIN Schedule ON

Prereg.CourseID=Schedule.CourseID WHERE StudentID="

Request.Form("StudentID") & ""

set rs = conn.Execute(sqlQuery)

if not rs.EOF then

Response.Write "<select name='RCourse'>"

While not rs.EOF

Response.Write "<option value='" & rs("CourseID") & "'>" & rs("CourseID") & "

" & rs("Time") & "-" & rs("Name") & "</option>"

rs.MoveNext

Wend

Response.Write "</select>"

rs.Close

else

%>

None!

```

<do type="accept" label="Main">

    <go href="login.asp" method="post">

        <postfield name="StudentID" value="$(StudentID)" />

        <postfield name="Password" value="$(Password)" />

    </go>

</do>

<%

    end if

%>

</p>

</card>

</wml>

```

Filename: Search.asp

Description:

There are two cards in this deck. The first one is "Search" card. When you click on hyperlink "Search", the application leads you to this card. The user can input their word to search for the CourseID that he/she requires then click on the "Find" button. The program will search into the table schedule and find any CourseID that match with the word phase then display the result in the second card name "Result". If there is no word match found the program will show the message "Sorry, no Course found."

The result records are display in tag <select> and <option> that allow the user select them and click on the register button to do the registration process in this page.

There is also one tag <anchor> name "Main" that let the user can link back to the main page and start another command as well.

```

-----

<% Response.ContentType = "text/vnd.wap.wml" %>

<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1 //EN"
"http://www.wapforum.org/DTD/wml 1.1.xml">

<wml>

<%

    Dim conn, rs

    Set conn = Server.CreateObject("ADODB.Connection")

    Set rs = Server.CreateObject("ADODB.Recordset")

    conn.open "DRIVER= {Microsoft Access Driver (*.mdb)} ;DBQ=" &
Server.MapPath("Register.mdb") & ";"

    SearchStr = Request.QueryString("CourseID")

    if SearchStr="" then
%>
<card id="Search" title="Search">

    <p>

        <be>

        -----

        <br/>

        CourseID: <input name="CourseID" type="text" maxlength="6" />

        <br/>

        -----

        <br/>

        <do type="accept" label="Find">

```

```

        <go href="Search.asp" method="get">

            <postfield name="CourseID" value="$CourseID" />

        </go>

    </do>

</p>

</card>

<%

else'---perform a SQL search---

    sqlQuery = "SELECT * FROM Schedule WHERE CourseID LIKE '%" &
SearchStr & "%'"

    Set rs = conn.Execute(SQLquery)

%>

<card id="SearchResult" title="Resule">

    <p align="center">

        <anchor>

            Main *

            <go href="login.asp" method="post">

                <postfield name="StudentID" value="$Student1D" />

                <postfield name="Password" value="$Password" />

            </go>

        </anchor>

        <br/>

        - - - - .Search Results- - - - .

        <br/>

    <%

```



```
if not rs.EOF then

    Response.Write "<select name='CID'>"

    While not rs.EOF

        Response.Write "<option value=' & rs("CourseID") & "'>" &
rs("CourseID") & " " & rs("Time") & " " & rs("InstructorName") & " " & "</option>"

        rs.MoveNext

    Wend

    Response.Write "</select>"

    Response.Write "<do type='accept' label='Register'>"

    Response. Write " <go href='Course. asp' method='post'>"

    Response. Write "      <postfield name='StudentID' value=S(StudentID)' />"

    Response. Write "      <postfield name='CourseID' value='$CID' />"

    Response.Write " </go>"

    Response.Write "</do>"

else

    Response.Write "Sorry, no Course found."

end if

%>

</p>

</card>

<%

end if

%>

</wml>
```

V. SYSTEM IMPLEMENTATION

The system implementation will describe about how to configure the system both client and server to provide the WAP service. Also the writer will mention about how to setup the database that will keep the information and provide the detail of how to integrate everything to work together correctly. So this implementation part will be separate into three subjects.

The first one is how to configure the web server that uses Microsoft Internet Information Service to provide the WAP service to the user. The second is how to configure the WAP mobile phone to connect to the gateway and work with our system. The last topic will mention about how to setup the data in database to keep our application information and provide the user.

When everything is configured properly they will work together to provide the service and information to the user. One thing that everyone should know is that there are so many web server software and mobile phone model. Each of this software and mobile phone need different methods of configuration. So please ask your vendor to provide you about how to configure your software to work with WAP application and how to configure your mobile. The details of how to configure the mobile phone often appears in the web sites of your mobile vendor so you can go there and see how to configure it.

Some mobile phones have upgraded version of firmware so they may require to be upgraded before they can use WAP application. You can go to your local customer service center to ask them to upgrade your mobile phone firmware that will result in better utilize your hardware resource and displaying method.

5.1 Microsoft Internet Information Service Configuration

- (1) Open the Internet Service Manager program
- (2) Right click at the website that we would like to set it up and then click on property in pop up menu.
- (3) Click on HTTP Headers TAB as the picture show below where the mouse pointer exists.

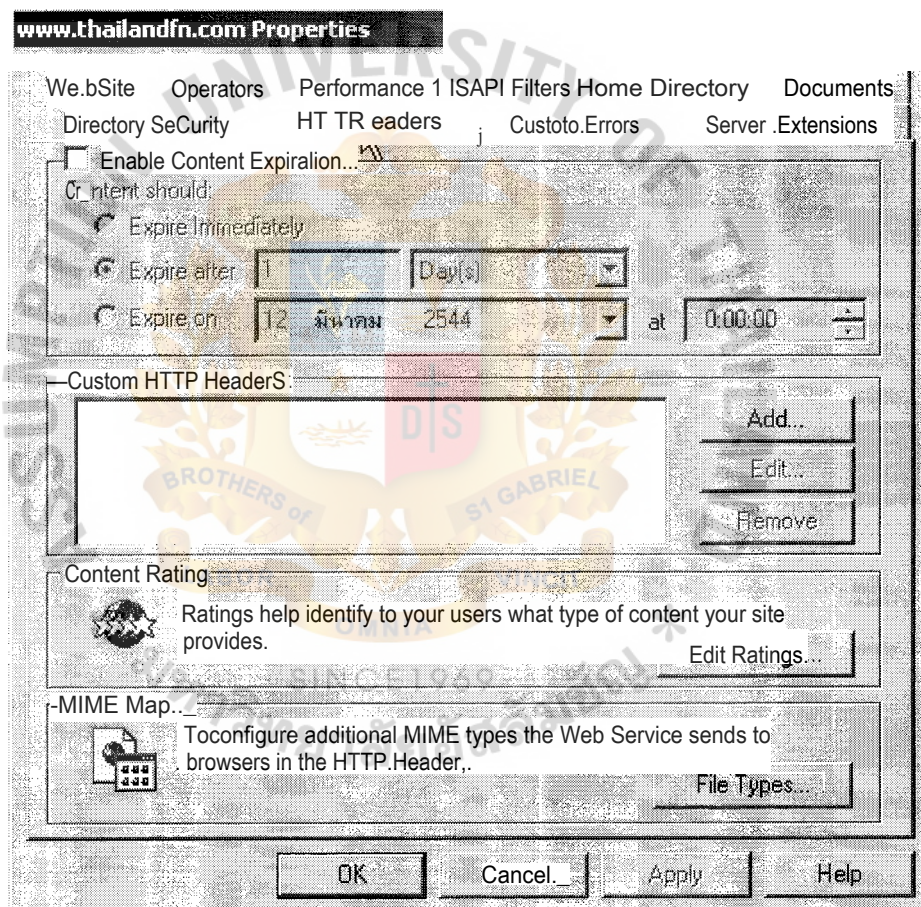


Figure 5.1. HTTP Headers Configuration.

- (4) In the MIMI Map section, find the "File Types..." button to open the "File Types..." dialogue.

- (5) The "File Types..." dialogue will be opened as the picture shown below.

This is where we add types of MIME files.

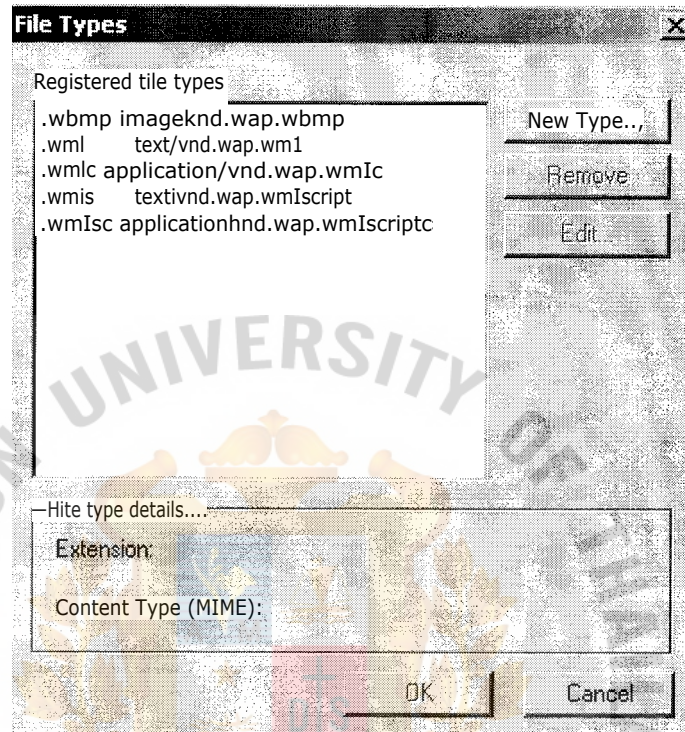


Figure 5.2. File Types Configuration.

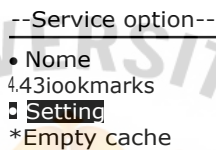
- (6) Click on the "New Type..." button to add every MIME Type to the File type as the picture shown above. All MIME Type are list here.

.wbmp image/vnd.wap.wbmp
.wml text/vnd.wap.wml
.wmlc application/vnd.wap.wmlc
.wmls text/vnd.wap.wmlscript
.wmlsc application/vnd.wap.wmlscriptc

5.2 Mobile Configuration

The configuration setting for Nokia 7110

- (1) Move Navi Roller to Menu service and then click at Navi Roller for choosing Menu
- (2) Then move Navi Roller tab to Setting, after this choosing from setting menu (please see the picture as follows)



--Service option--

- Nome
- 4.43iookmarks
- *Setting
- *Empty cache

Figure 5.3. Service Option Setting.

- (3) The setting menu composed of:

3.1 Connection

3.2 Security Certificate

We must choose at Connection (as the follows picture)



--Settings--

- *Connection
- 4. Security Certificate

Figure 5.4. Connection Setting.

- (4) The Connection setting composed of Set 1, Set 2 ... to Set 5 , Profile has its

own function as the alternative for connecting to required ISP. We must choose at Set 1.

-7connection,setting--

- Profile_
- Profile3
- *Profil05

Figure 5.5. Connection Profiles Setting.

(5) There are 3 Menu in Profile 1:

1. Activate 2. Edit 3. Rename

We must choose at Edit.

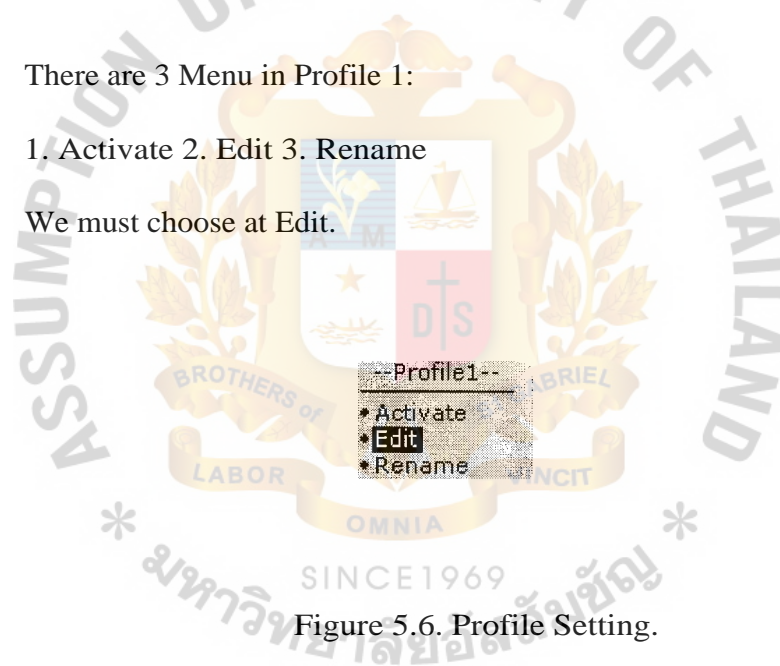


Figure 5.6. Profile Setting.

(6) Until now, We gonna reach the sky. There are a lot of subsequent menu then it must be done one- by one process.

(6.1) Homepage is wap.mweb.co.th

(6.2) Connection type is Continuous

(6.3) Connection security is Off

(6.4) Bearer is Data

(6.5) For Dial up number, if you use AIS, the number will be 900937,

900933, 900934. TAC number will be +6616120012

(6.6) IP Address of AIS is 202.183.232.107 but TAC is 203.155.200.133

(6.7) Authentication is normal

(6.8) Data call type is ISDN

(6.9) Data call speed is Auto bounding

(6.10) User name : ais

(6.11) Password : ais

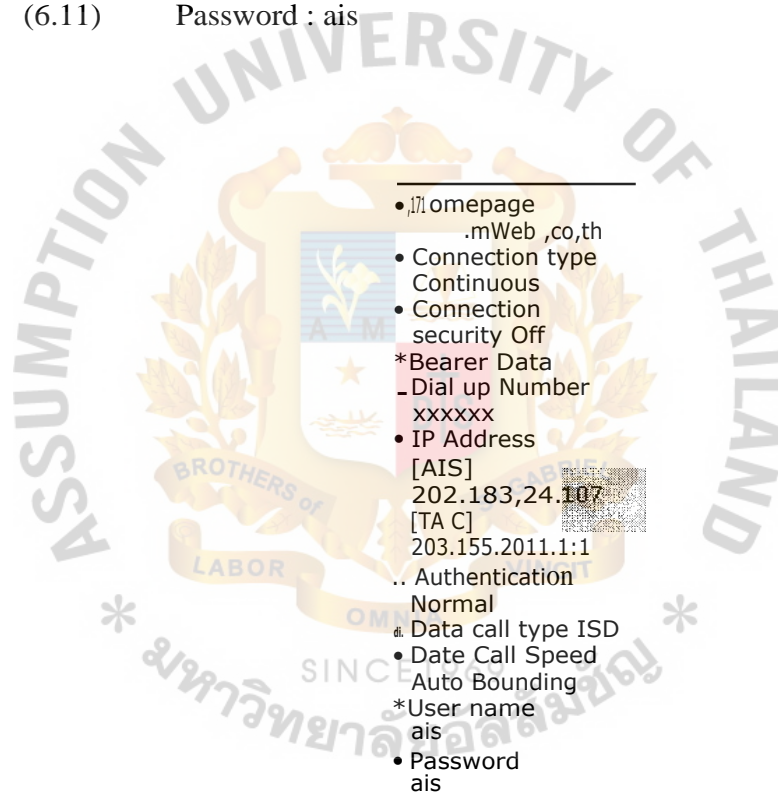


Figure 5.7. Example Setting.

Click Back button to Profile menu

If you remember, there is an alternative named " Activate " in profile menu.

Be chosen on Activate.

(9) The last step, click back button to Service menu, then go to Home. The mobile will be connected automatically.

- 5.3 Database Setup

(1) User table setup

Table 5.1. User Table Data.

B User : Table		
StutputID		Password
64219341	Thawee Sudthikul	
64219342	Napapan Vorached	
64219343	Verachai Chantanumas	
64219344	Chatchai Jareonwong	
Record	7	7

(2) Schedule table setup

Table 5.2. Schedule Table Data.

Schedule : Table				
CourselD	Time	Name	Ruum	InstructorName
CE6401	We 18:30-21:30	Management of Computer Center	A52	
EFALL.i	Sa 17:30-20:30	Management of Comuter Security	A71	
+ Li,E13404	Tu 18:30-21:30	System Analysis and Design	A51	
+ CE6501	Sa 09:00-12:00	Principles of Engineering Management I	A72	
+ CE6502	Su 09:00-12:00	Principles of Engineering Management II	A102	
+ CE6506	Mo 18:30-21:30	Computer and Engineering Economy	A72	
+ CE6614	Fr 18:30-21:30	Packaging Management	A51	
+ CE6998	Th 18:30-21:30	Project		
CS6102	Mo 18:30-21:30	Management Information Systems	A51	
Record: 14 of 10				

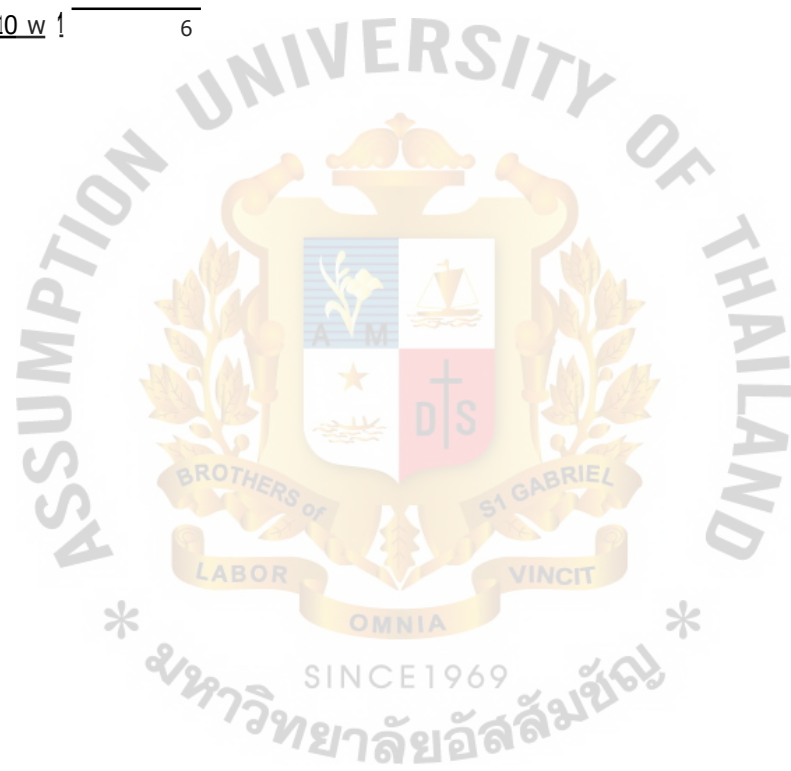
(3) PreReg table setup

Table 5.3. PreReg Table Data.

N Prereg : Table 251

StudentID	CourseID	RegistDate
G4219340	CE6401	11/5/2001
G4219340	CE6506	1/5/2001
G4219340	C 56102	11/5/2001
G4219341	C E6401	11/5/2001
G4219342	OE6506	11/5/2001

Record: 10 w 1 6



VI. SYSTEM EVALUATION

There are so many mobile manufacturers that result in so many brands and models of mobile available today and each model has its own WAP browser that has different method to display and process the application's instruction. The limitation of each model is not also the same. Some models have much memory and can display the screen up to seven or eight lines but some have a little memory and the few display lines.

6.1 Evaluation Method

To test the application in this project, the writer used several ways to ensure that the application will be applicable in different mobile models. The writer tries to use both mobile emulator and the real mobile to test the application many times (refer to Figure 3.2 to see the using manual). These includes:

- (1) NOKIA 7110
- (2) SIEMENS S35
- (3) ERICSSON R 380 emulator

All mobiles and emulators that are listed above can work very well with the application but there is some different displaying method depends on type and model of the mobile or emulator. The response time is also our one considering factor. It depends on the WAP gateway's provider and public network bandwidth and has a great impact with the user's satisfaction.

The writer concentrates on two points of evaluated method. First is verification that will result in user's satisfaction. And the second point is validation that will try to find the error and the limitation of the system.

6.2 Verification

In verification process, the writer tries to test on the important factor that will result in user's satisfaction. These factors include accuracy, user friendliness and response time.



Figure 6.1. NOKIA 7110.

Many of the most important features of the Nokia 7110 is the large display with seven lines character display with the microbrowser, NaviTM Roller and predictive text input were developed to make it faster and easier to access WAP services available over the Internet by using a mobile phone. Anyway the WAP browsers have the worst treatment for long line text and it cropped off all excess characters that cannot fit on the same line. This is the weakness point of this model.

The response time is very good since the WAP gateway that the writer connects to is AIS. AIS is the WAP gateway service provider that use the ISDN network technology that will enhance the speed of the telephone line network.



Figure 6.2. SIEMENS S35.

For SIEMENS S35, The method of displaying character is better than in NOKIA 7110. The number of text displaying line is seven equal to the NOKIA 7110 but it can wrap the excess line character to the next line and have a good page alignment method. Anyway it has less memory than 7110 that make it can not display the large deck and the developer must separate the code into several smaller decks.

Anyway the response time is the same as in Nokia 7110 since the WAP gateway provider is the same.

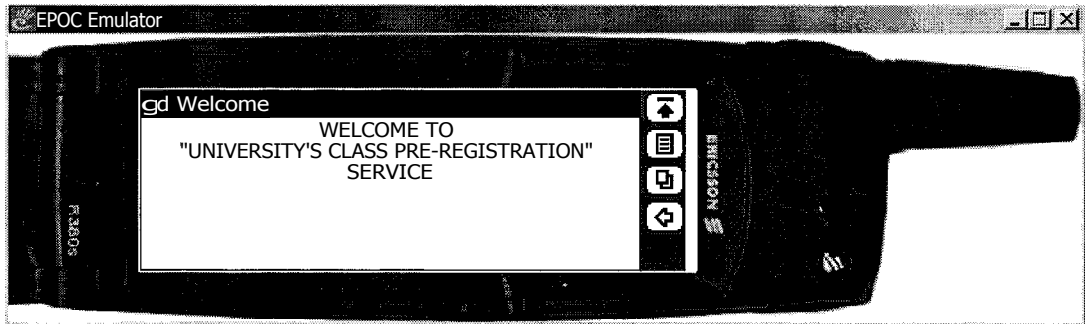


Figure 6.3. Ericsson R380.

ERICSSON R 380 emulator has the largest display that help the user to view the content very easily and the touch screen function that allow user very easily to typing the character or selecting the menu items.

A big memory helps it in coping with the large application program deck that is it advantage. The programmer can reduce the number of deck that will result in reduce the number of times of re-connection in the session to get the new card.

The response time is again the same as in NOKIA 7110 and SIEMENS S35 while the provider is AIS.

6.3 Validation

In validation process, the writer tries to find the error in program and the limitation of the program that depend on each mobile brand. Each brand has different memory size so it means there is a limitation of card size for each mobile brand. The below is the list of possible card size for each brand.

- (1) Nokia 7110 1397 Bytes
- (2) Ericsson R320 3000 Bytes
- (3) Ericsson R380 3800 Bytes

So this means that the application has a limitation of displaying the number of record to the user. If there are too many records of course available then it couldn't be display in one card. But luckily that in real situation the number of total course available should not over the limitation of the card size.

The important thing is that the application don't have the limit of registered coursed so that the user can do the registration process every time they need. But they should know that the first four registered courses only could be applied.

Another limitation for this program is that the text that will display to the user can be only in form of English language. If someone tried to put Thai language or another language into the database it will not show to the user correctly and make the user unreadable.



VII. CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

In this project, the writer has shown a typical application that is commonly found on E-commerce web site can be adapted to run on a WAP device. Apparently, the screen size of WAP devices poses a lot of constraints for web developers who are used to designing applications for the web. Web development skills like HTML and ASP is still essential when creating WAP applications. What we foresee is that for the next couple of months, many companies will be writing two different versions of their Internet applications: one for the web and one for WAP.

Another tricky issue that must be identified is the portability of WAP applications. Care must be taken to test out the application on the major WAP browsers used in the market. Different WAP browsers have different design guidelines and developers must be aware of these differences if they want to write portable WAP applications. It is just like creating web pages that work and display well on both Microsoft Internet Explorer and Netscape Navigator.

Another problem that the writer encountered was the limitations of the development environment. The writer particularly likes to use the Nokia WAP toolkit and Nokia WAP simulator to test my WAP application before trying it on the other platforms. As WAP gains in popularity in the next couple of months, expect to see more user-friendly development tools that offer more debugging assistance.

The WAP technology is still moving its first steps, nevertheless it promises to be one of the most revolutionary IT developments in the last few years. In this project the writer has explained how to use ASP to build services for WAP telephones. The writer has built a plausible application and warned about possible problems and pitfalls that

will encounter if you have to build a WAP service in the near future. Even though the multimedia possibilities of the Internet are far from being a reality on the handheld, one advantage of WAP is the built-in authentication mechanism of mobile telephones, which opens new horizons for E-commerce.

7.2 Recommendations

You will see that the writer's application work is very well suitable in most environment. The response time is very short that make the user more likely and satisfaction. The data are accurate and enough for the user to do their application although the application doesn't have any attractive picture or image animation like in the web page. There is no graphic menu, hint or online help like the most application on the desktop. Every card or screen display only plain text, input form and function required for the application.

This is the most important point of developing the WAP application that will applicable in the real environment that the network bandwidth of mobile phone network in Thailand is very low only 9,600 bps compare with another modem connection from desktop computer to Internet Service Provider at least 33,600 bps in present.

WAP developer should know that they must minimize the program to the smallest size because of two reasons. One is the cost of transmission the data across the network and two is the maximum size of program that different mobile brand can accept. The big card program may process and display on one mobile model but it mightn't be processed and display in another model depend on the capability of that model.

The slow connection speed may cause the trouble to user if our program card is too big. They may wait too long before receiving the image logo of the company that make them boring to use the application.

So what is the good characteristic for WAP application? The objective of WAP application is to serve the need of user that would like to do the application by wireless terminal. So portable application is the most meaning for the user that will let the users do their transaction or process from everywhere they needs.

Not only portability factor is important but also the application must response to the users fast enough and must provide enough information before the users can decide to do the application transaction. This will be return back to the problem of the size of information and the displaying method for each mobile model. The information that will be provided to the user should be much enough to them to make their decision refer to the basic rule of need to know. Another improper information should be cut off

Another thing that the developer should know is that don't put many image or picture into the program because it will consume the time to transmit the information and wouldn't help anything to the user. Also the developer should know that one deck means one connection. If the users would like to browse to the card that isn't currently in the deck, they must connect to the server again to get the required card from another deck. This means that they must wait until all cards in that deck already transfer to their mobile. Try to think about the program that has several decks and each deck has many cards that link to the card in another deck. This means you must reconnect to the server every time you do the action and wait for the responding. So try to keep the relate card into the same deck would help the user to reduce the time to wait for the response.

Another issue is that if you try to keep everything in one deck. Your deck would be very large and couldn't fit into the memory of mobile phone. So the good program should separate the card into the proper deck with the acceptable size and don't have many deck that will cause the trouble of response time to the users.

7.2 Further Works

As the writer has shown you in this project that the application that available on web technologies today can be adjusted to applied with wireless device by WAP technologies. This means that not only our application can work in real situation but also another application can be easily adjusted and developed for use with wireless device.

University Class Pre-Registration is only one basic application in the most university. Another application can be developed such as the payment system that will link the process of registration to the payment system.

The bulletin board also can be applied to the WAP technology. So the examination result and other news can be easily notified to the university student. This application can be developed by two different ways. First one is the bulletin that every student must connect their mobile phone to the gateway for use WAP service and browse to the location of WAP pages or by second way the application is developed by Short Message Service (SMS) technology to send the news and information to the users by short message.

Using email by WAP is the common application that can be developed today. It is a very powerful method for the users or student that can read their emails everywhere every time by connect their wireless device to the system and get their emails from mail server by WAP application.

There are so many things that we can extend our project to applied with the current system as the writer suggests above. Every business today must start to think twice about the new methodology that can be the new channel to provide the information and service to their customer. Which one should be developed is depend on which is applicable and which can give them back the worth profit. Anyway the cost of

WAP development is also a little compare with another system because it doesn't require new hardware and software. WAP require only the adjustment of current web technology with some code developing process that shouldn't take the project time too much. It is the easy developing program that is the part of XML that is well known today.





The Supported Character Set.

- UTF-8
- UTF-16
- UCS4
- ISO-8859-1
- ISO-8859-2
- ISO-8859-3
- ISO-8859-4
- ISO-8859-5
- ISO-8859-6
- ISO-8859-7
- ISO-8859-8
- ISO-8859-9
- Latin6

Character Mappings

Alias	Mapped to
Iso-ir-100	ISO-8859-1
ISO 8859-1:1987	ISO-8859-1
ISO 8859-1	ISO-8859-1
Latin 1	ISO-8859-1
Latin-1	ISO-8859-1
L1	ISO-8859-1
IBM819	ISO-8859-1
CP819	ISO-8859-1

CslSOLatin1	ISO-8859-1
Iso-ir-101	ISO-8859-2
ISO 8859-2:1987	ISO-8859-2
ISO 8859-2	ISO-8859-2
Latin2	ISO-8859-2
Latin-2	ISO-8859-2
L2cslSOLatin2	ISO-8859-2
Iso-ir-109	ISO-8859-3
ISO 8859-3:1988	ISO-8859-3
ISO 8859-3	ISO-8859-3
Latin3	ISO-8859-3
Latin-3	ISO-8859-3
L3	ISO-8859-3
Cs1SOLatin3	ISO-8859-3
Iso-ir-110	ISO-8859-4
ISO 8859-4:1988	ISO-8859-4
ISO 8859-4	ISO-8859-4
Latin4	ISO-8859-4
Latin-4	ISO-8859-4
L4	ISO-8859-4
CslSOLatin4	ISO-8859-4
ISO-Latin-4	ISO-8859-4
Iso-ir-144	ISO-8859-5
ISO 8859-5:1988	ISO-8859-5

St. Gabriel's Library

ISO 8859-5	ISO-8859-5
Cyrillic	ISO-8859-5
Cs1SOLatinCyrillic	ISO-8859-5
Iso-ir-127	ISO-8859-6
ISO 8859-6:1987	ISO-8859-6
ISO 8859-6	ISO-8859-6
ECMA-114	ISO-8859-6
ASMO-708	ISO-8859-6
Arabic	ISO-8859-6
Cs1SOLatinArabic	ISO-8859-6
Iso-ir-126	ISO-8859-7
ISO 8859-7:1987	ISO-8859-7
ISO 8859-7	ISO-8859-7
ELOT 928	ISO-8859-7
ECMA-118	ISO-8859-7
Greek	ISO-8859-7
Greek8	ISO-8859-7
Cs1SOLatinGreek	ISO-8859-7
Iso-ir-138	ISO-8859-8
ISO 8859-8:1988	ISO-8859-8
ISO 8859-8	ISO-8859-8
Hebrew	ISO-8859-8
Cs1SOLatinHebrew	ISO-8859-8
Iso-ir-148	ISO-8859-9

ISO 8859-9:1989	ISO-8859-9
ISO 8859-9	ISO-8859-9
Latin5	ISO-8859-9
Latin-5	ISO-8859-9
L5	ISO-8859-9
Cs1SOLatin5	ISO-8859-9
Iso-ir-157	Latin6
16	Latin6
ISO 8859-10:1992	Latin6
ISO 8859-10	Latin6
Cs1SOLatin6	Latin6
UTF 8	UTF-8
UTF8	UTF-8
UTF 16	UTF-16
UTF16	UTF-16
UCS-2	UTF-16
UCS 2	UTF-16
UCS2	UTF-16
UCS-4	UCS4
UCS 4	UCS-4
Unicode	UTF-16
10646	UTF-16
ISO/IEC10646-1	UTF-16



APPENDIX B

NOKIA WAP TOOL KIT 2.1 INSTALLATION

Installation Procedure

The Toolkit is available as the downloadable, self-extracting executable file setup.exe. It contains the Basic Toolkit, Server Simulator, and Blueprint phone simulation.

In order to use the Toolkit, you must always install the file setup.exe. You can install and uninstall the optional phone simulations at any time.

To run setup.exe:

1. Choose Run from the Windows Start menu.
2. In the Run dialog box that appears, enter the path of the setup.exe file and click OK or using Windows Explorer, browse to the directory where the Nokia WAP Toolkit installation file setup.exe is located.
3. Double-click the file setup.exe.
4. The InstallShield installation program runs, displaying the Installation Screen Dialogs
5. In Welcome dialog Title Choose Next.
6. Read the license agreement. You must choose Yes.
7. Choose Destination Location or the installation directory. The default location is recommended. Then choose Next.
8. Select Components (By default, both components are initially selected).After selecting components, choose Next.
9. Select Program Folder for Toolkit icons. Then choose Next
10. Start copying files, Then choose Next.
11. Setup Complete Choose Finish.



APPENDIX C

NOKIA PHONE SIMULATOR INSTALLATION

This appendix describes the optional Nokia phone simulations that you can install for use in the Toolkit. You can download the following simulations from <http://www.forum.nokia.com/wapforum/main/toolkit>.

Simulation File

- (1) Nokia 6210 6210S imulator.exe
- (2) Nokia 7110 (Sep2000) 7110Sep2000Simulator.exe
- (3) Nokia 7110 (Jan2000) 7110Jan2000Simulator.exe

Installing Optional Nokia Phone Simulations

After downloading the desired Nokia phone simulation, install the simulation (here termed <simulator.exe>) as follows:

- (1) Run the simulation executable (.exe) file.
- (2) Choose Run from the Windows Start menu.
- (3) In the Run dialog box, enter the path of the <simulator.exe> file and click **OK** or Using Windows Explorer, browse to the directory where <simulator.exe> is located.
- (4) Double-click the file <simulator.exe>.
- (5) The InstallShield installation program runs, displaying the following screen (here the Nokia 6210 is shown):

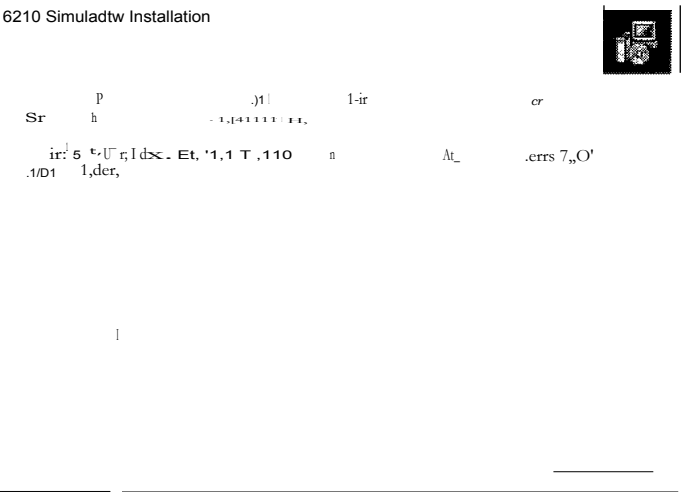


Figure A.1. Nokia Phone Simulator Installation Step 1.

- (6) As described in the screen above, you must install the simulation into the directory where Toolkit is installed. Choose Next. The following screen is displayed:

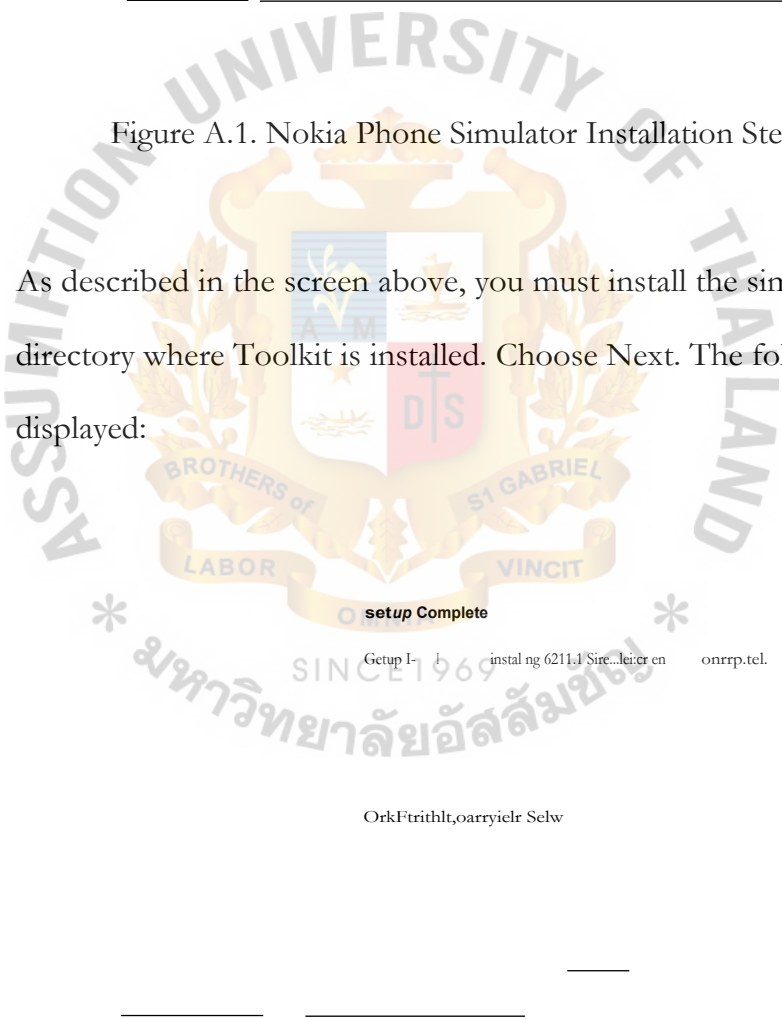


Figure A.2. Nokia Phone Simulator Installation Step 2.

- (7) Choose Finish

BIBLIOGRAPHY

1. Academic Computing and Instructional Technology Services (ACITS). "Data Modeling." United States: <http://www.utexas.edu/cc/database/datamodeling/rm/>, 2000.
2. Date, C. J. An Introduction to Database Systems, 5th Edition. United States: Addison-Wesley, 1990.
3. Fleming, Candace C. and Barbara von Halle. Handbook of Relational Database Design. United States: Addison-Wesley, 1989.
4. Kroenke, David. Database Processing, 2nd Edition. United States: Science Research Associates, 1983.
5. Martin, James. Information Engineering. United States: Prentice-Hall, 1989.
6. Nokia Corporation. "Nokia WAP Toolkit 2.1." Finland: <http://www.nokia.com>, 2001.
7. Reingruber, Michael C. and William W. Gregory. The Data Modeling Handbook: A Best-Practice Approach to Building Quality Data Models. United States: John Wiley & Sons, Inc., 1994.
8. Simsion, Graeme. Data Modeling Essentials: Analysis, Design, and Innovation. United States: International Thompson Computer Press, 1994.
9. Teory, Toby J. Database Modeling & Design: The Basic Principles, 2nd Edition. United States: Morgan Kaufmann Publishers, Inc., 1994.
10. Thaiwapportal Online. "Wireless Application Protocol." Thailand: <http://www.nokiathai.conil>, 2001.
11. Wireless Developer Network. "The Wireless Markup Language (WML)." United States: <http://www.wirelessdevnet.com/channels/wap/training/wml.html>, 2000.
12. Wireless Internet Today. "Wireless Application Protocol White Paper." United States: <http://www.wapforum.org>, June 2000.