

WAP ALUMNI REGISTRATION SYSTEM

ARIFF SAFWAN BIN RADZI

A thesis submitted in fulfillment of the
requirements for the award of the degree of
Bachelor of Electrical Engineering (Electronics)

Faculty of Electrical and Electronics Engineering
Universiti Malaysia Pahang

NOVEMBER 2007

“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : _____

Author : ARIFF SAFWAN BIN RADZI

Date : 20 NOVEMBER 2007

To my mother, father, grandma and brother with love

ACKNOWLEDGMENTS

Thank you Allah,

I wish to extend my sincere appreciation to Miss Mahfuzah bt Mustafa, my former supervisor, for her continuous supervision, guidance, advice and constructive ideas throughout the course of this study and making this project successful. Without her support, this would have been a very difficult assignment.

Special thanks also to Miss Haszuraidah bt Ishak, my supervisor for her assistant and critical analysis of this document. There are other lecturers, students, and colleagues who also helped in various ways with my bachelor, it would be impossible to list them all. Lastly, thank you to my parents and family for encouragement and your non-technical assistance with my bachelor degree.

ABSTRACT

Nowadays, mobile computing is said to be a new style of information access emerging at the intersection of two current trends; producing portable device in computer industry and wireless communication in telecommunication industry. This process will give a lot of benefits to human being in terms of its mobility and availability. Wireless Application Protocol (WAP) is a set of standards designed to extend Internet services to mobile phones, pagers, and personal digital assistants (PDAs). In view of these, the present study was conducted to use WAP in the alumni registration process. The main objectives of this study are to enable graduate students register alumni and share information using WAP technology via a cell phone or PDA. The system was develop using Microsoft Visual Web Developer, Openwave Simulator, Internet Information Services (IIS) as a web server and Microsoft Access as database management. The system also use Wireless Markup Language (WML) integrated with Active Server Pages (ASP) as a scripting language. This project involves four main phases. The first phase was doing a literature review; which all software and hardware specifications needed are met. The second phase is development and programming the codes for client and administrator side. The third phase is simulation and testing into real cell phone and the last phase is writing off the report.

ABSTRAK

Di era ini, pengkomputeran mudah alih dikatakan sebagai cara terbaru yang muncul di pasaran kini. Terdapat dua kaedah bagi memenuhi pengkomputeran mudah alih iaitu; penghasilan peralatan mudah alih dalam industri komputer and komunikasi tanpa wayar. Proses ini akan memberi pelbagai kemudahan kepada manusia dalam konteks mudah alih dan ketersediaannya. WAP adalah satu set piawaian yang direka untuk menyalurkan kemudahan Internet tanpa wayar kepada telefon bimbit, alat kelui, dan PDA. Berdasarkan pandangan ini, kajian telah dibuat untuk menggunakan teknologi WAP bagi tujuan pendaftaran alumni. Objektif utama kajian ini adalah untuk memastikan graduan dapat mendaftarkan diri mereka sebagai alumni dan berkongsi maklumat menggunakan teknologi WAP menerusi telefon selular atau PDA. Sistem ini telah dibangunkan menggunakan perisian *Microsoft Visual Web Developer*, *Openwave Simulator*, *Internet Information Services (IIS)* sebagai pelayan dan *Microsoft Access* sebagai pengurus pangkalan data. Sistem ini juga menggunakan teknologi *Active Server Pages (ASP)* yang digabungkan bersama *Wireless Markup Language (WML)* sebagai bahasa pengaturcaraan. Keseluruhannya, projek ini merangkumi 4 fasa utama; iaitu fasa pencarian maklumat, dimana semua keperluan perisian dan perkakasan dipenuhi. Fasa kedua ialah pembangunan dan pengaturcaraan perisian dan kod untuk pelayan dan juga pelanggan. Fasa ketiga ialah proses simulasi dan pengujian terhadap telefon bimbit sebenar, dan fasa terakhir ialah penulisan laporan.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	TITLE PAGE	i
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGMENTS	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	x
	LIST OF FIGURES	xi
	LIST OF ABBREVIATIONS	xiii
	LIST OF APPENDICES	xiv
1	INTRODUCTION	1
	1.1 Project Background	1
	1.2 Project Objectives	1
	1.3 Project Scopes	2
	1.4 Problem Statement	3
2	LITERATURE REVIEW	4
	2.1 Introduction	4
	2.2 Alumni	4
	2.3 Web Programming Technologies and Languages	5
	2.3.1 Hypertext Markup Language (HTML)	5
	2.3.2 Dynamic Hypertext Markup Language (DHTML)	5
	2.3.3 Wireless Application Protocol (WAP)	6

	2.3.4	Wireless Markup Language (WML)	9
2.4		Web Server	11
	2.4.1	Architecture	11
	2.4.2	Internet Information Services (IIS)	11
	2.4.3	External web server	12
2.5		Dynamic page	12
	2.5.1	Active Server Pages (ASP)	12
	2.5.2	Database	13
2.6		Device	14
	2.6.1	Phone Simulator	14
	2.6.2	Mobile phone	14
2.7		Comparison	15
3		METHODOLOGY	17
3.1		Introduction	17
3.2		Requirements Definition	19
	3.2.1	Development Platform	19
	3.2.2	Database Requirement	19
	3.2.3	Web Server	20
	3.2.4	Hardware and Software Specification	20
3.3		Administrator-side Development	22
	3.3.1	Microsoft Access 2003	22
	3.3.2	ASP Coding	23
3.4		Client-side Development	25
	3.4.1	Architecture	25
	3.4.2	MIME type configuration	26
	3.4.3	ASP Coding	27
3.5		Integration & Simulation Testing	30
	3.5.1	Localhost Operation	30
	3.5.2	Openwave Simulator	31
	3.5.3	External Web Server	32
	3.5.4	FTP Configuration	32
3.6		Data analysis	33

4	RESULT	34
4.1	Administrator-side	34
4.1.1	Default & Sign-in page	35
4.1.2	Menu selection page	36
4.1.3	Alumni member list page	37
4.1.4	News management page	38
4.2	Client-side	39
4.2.1	Home & selection page	40
4.2.2	Registration page	41
4.2.3	News page	42
4.2.4	Member search page	43
4.2.5	Profile update page	44
4.3	Constraints	45
5	DISCUSSION AND CONCLUSION	46
5.1	Discussion	46
5.2	Conclusion	47
5.3	Future Recommendations	48
5.3.1	Costing and Commercialization	48
	BIBLIOGRAPHY	49
	APPENDICES	51
	APPENDIX A: Project Schedule	51
	APPENDIX B: Codes	53 - 71

LIST OF TABLES

TABLE NO.	TITLE	PAGE
2.1	A comparison between available WAP-enabled systems	16

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	Flowchart of software system	2
2.1	WAP protocol architecture	7
2.2	Architecture of a dynamic, Interactive & data-driven website	11
2.3	ASP and ADO Object Model Structure	12
2.4	Displaying WAP content using a gateway and a simulator	15
3.1	Project Flow Model	18
3.2	DSN-less connection tag	23
3.3	SQL statement tag	24
3.4	Architecture for client-side development	25
3.5	MIME type configuration menu	26
3.6	WML document prologue tag	27
3.7	Cache control tag	28
3.8	DSN-less connection tag	28
3.9	SQL statement tag	29
3.10	Close tag	29
3.11	Openwave simulator V7 main screen	31
3.12	Site properties window in cuteFTP	32
4.1	Administrator sign-in page	35
4.2	Menu selection page	36
4.3	Registered alumni member view page	37
4.4	News management page	38
4.5	Default page for client-side	40
4.6	Selection page for client-side	40

4.7	Registration page for client-side	41
4.8	News page for client-side	42
4.9	Member search page for client-side	43
4.10	Member search result page	43
4.11	Profile login update page	44
4.12	Profile update page	44

LIST OF ABBREVIATIONS

ADO	- ActiveX Data Object
ASP	- Active Server Pages
COM	- Component Object Model
CSS	- Cascading Style Sheets
DHCP	- Dynamic Host Configuration Protocol
DHTML	- Dynamic Hypertext Markup Language
DSN	- Data Source Name
DTD	- Document Type Definition
EDGE	- Enhanced Data Rate for GSM Evolution
FTP	- File Transfer Protocol
GB	- Gigabyte (1 trillion byte)
GPRS	- General Packet Radio System
HTML	- Hypertext Markup Language
HTTP	- Hypertext Transfer Protocol
IIS	- Internet Information Services
IP	- Internet Protocol
ISP	- Internet Service Provider
ISBN	- International Standard Book Number
ISP	- Internet Service Provider
MIME	- Multi-Purpose Internet Mail Extension
OLE DB	- Object Linking and Embedding Database
PDA	- Personal Digital Assistant
SQL	- Structured Query Language
TCP/IP	- Transmission Control Protocol/Internet Protocol
UMTS	- Universal Mobile Telecommunications System
WAP	- Wireless Application Protocol
WML	- Wireless Markup Language
XHTML	- Extensible HTML
XML	- Extensible Markup Language

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Project Schedule Gantt chart	52
B	Project Codes for Administrator-side development	54
C	Project Codes for Client-side development	62

CHAPTER 1

INTRODUCTION

1.1 Project Background

After graduation, graduates will detach and live their own. But to keep their relationships, they need alumni as a communication platform. So this system will help them to register their alumni using the fastest and easiest way via Wireless Application Protocol (WAP) technology. It also can help administrator to manage their alumni page easily, so that the graduates can get the latest news and information regarding their alumni.

1.2 Project Objectives

The objectives of this project are:

- i. To enable graduate students register alumni using WAP technology.
- ii. To enable graduate students share an activity and information regarding their alumni.

This project also comes with administrator site management system. The advantages of administrator site are:

- i. Easy for administrator to key-in data regarding alumni activities and information;
- ii. Data entered by alumni member are easily managed and intended for data analysis process;
- iii. Let the information spread between alumni member and university.

1.3 Project Scopes

The scope of this project is to ensure alumni become a communication platform between graduate students. Figure 1.1 below shows the flowchart of overall software system design for this project.

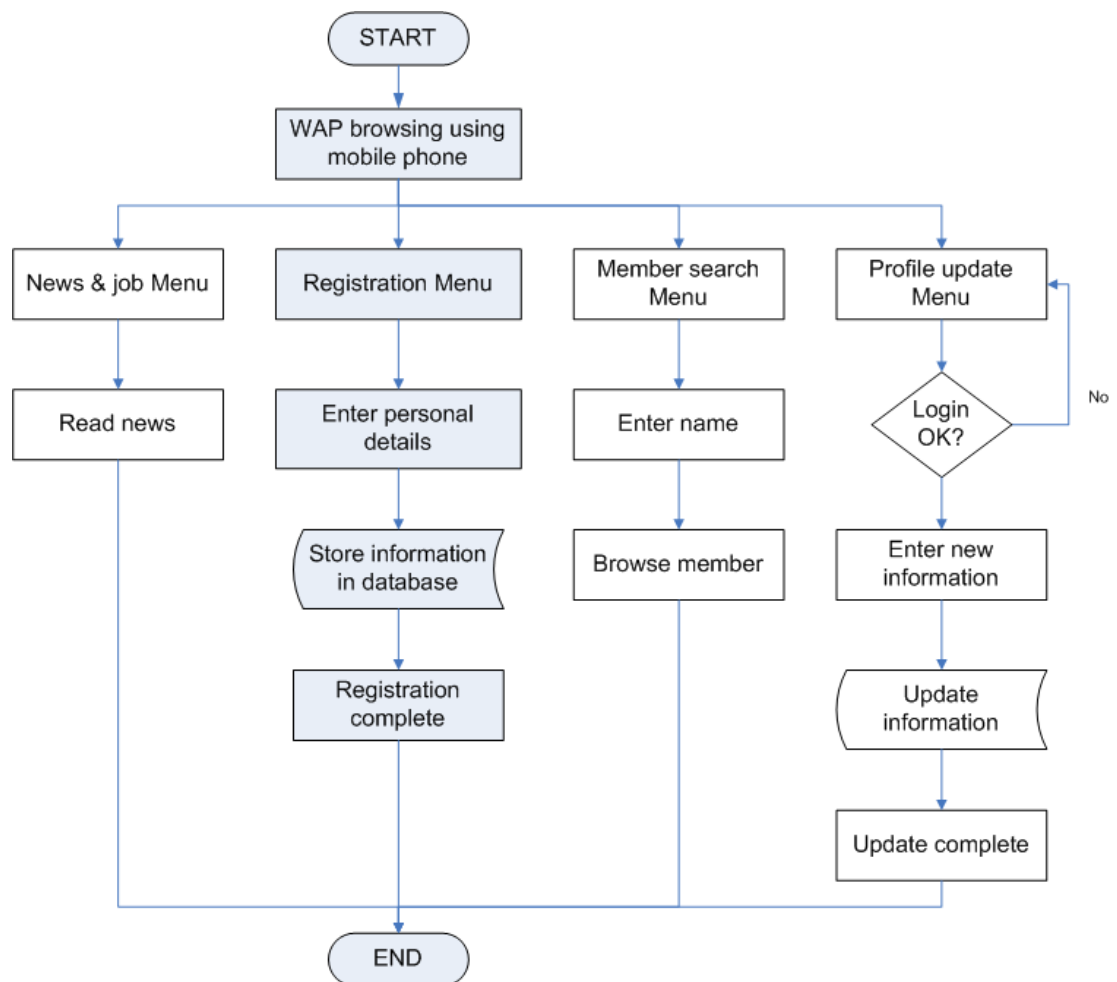


Figure 1.1: Flowchart of software system design for WAP Alumni Registration System

The process started when user browse into WAP Alumni Registration page using any WAP-enabled device such as hand phone or a Personal Digital Assistant (PDA). When the connection is established between the microbrowser and web server, the main menu will appear. User can make selection between these four menus; news and job, registration menu, member search menu and the last one was update profile menu. After that they point the microbrowser to registration menu and

start to fill their personal details like name, address, password, contact number, matrix number, occupation and year of graduation. Password is used to update the profile in profile update menu. If the registration is success, the data (personal details) will be stored in database. If it fails, user must enter the data back accordingly to ensure the registration is succeeded. To search for existing member, user can choose member search menu and they must key-in at least 2 characters, the result like name, phone number and address will come out if the user existed in database. The other menu will let user to read recent news and job vacancy regarding their alumni.

1.4 Problem Statement

The majority of graduate students are not interested in joining Alumni. Instead, some of them really don't know about alumni. This phenomenon also occurs at Universiti Malaysia Pahang.

Student must go to Student Affairs office to get an Alumni form, fill-up the form and submit back to complete the registration process. But some of them think the process to become an alumni member is inconvenient. So, hopefully this problem can be reduce after developing this system.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter reviews about Web programming technologies and languages, web server, dynamic page and the device used in developing this system.

2.2 Alumni

Previous research suggests students who are active in life of college campus, primarily through voluntary association in student activities and organizations are likely to be more philanthropic toward their alma mater than those who less involved, or not involved at all. According to Breazeale, institutional commitment created from student social involvement continues after graduation. Breazeale also stated that the student should [1]:

- i. Be made aware of Alumni Association, by making registration easier.
- ii. Develop leaders among the graduate student.
- iii. Involve alumni members in the affairs of the university.

One key assumption to ensure future alumni support lies in making entertainment students graduate with a positive overall attitude about their educational experience. Graduate who leaves with negative feelings about their alma mater will most likely never respond to request for financial support.

For this study, Alumni refer to persons with records on the internal database of Universiti Malaysia Pahang. Mostly, alumni registration is done by completing the form. Now in the new era of computer and telecommunication industry, the registration can be made by cellular phone anywhere.

2.3 Web Programming Technologies and Languages

2.3.1 Hypertext Markup Language (HTML)

Hypertext markup language (HTML) with its hypertext transfer protocol (HTTP) represents the third generation of Internet navigation technology. HTML can format Web pages for transferring and presenting text, graphics, sound, and full motion images on the Web. Users can view, save, or print Web pages containing text, graphics, sound, full-motion images, and hyperlinks from their web browsers. However, web pages formatted with HTML can only deliver information in a static presentation without interactivity. An interactive capability, for example would allow the user to do calculations, make purchases, pay bills, or manage bank accounts on the web.

2.3.2 Dynamic Hypertext Markup Language (DHTML)

In response to the demand for this type of dynamic, interactive and data-driven Web applications for e-business transactions, researchers have developed the fourth generation of Internet navigation technology, Dynamic hypertext markup language. DHTML is a combination of HTML and scripting languages such as JavaScript and Visual Basic Script. The basic purpose of DHTML is to embed scripts into HTML files to make Web page dynamic and interactive. With DHTML, developers can create Web applications that involve their users and enable them to carry out a wide variety of activities. For example, alumni registration system, class

attendance system, ticket and hotel reservation, bank account management and online purchasing via Web.

2.3.3 Wireless Application Protocol (WAP)

Wireless Application Protocol (WAP) is a specification for a set of communication protocols to standardize the way that wireless devices, such as cellular telephones and radio transceivers, can be used for Internet access, including e-mail, the World Wide Web, newsgroups, and Internet Relay Chat (IRC) [2]. WAP is the fifth generation of Internet navigation technology that enables people to enjoy wireless Web browsing and e-business transaction by using properly cell phone or a PDA.

Within the United States and Europe, the Wireless Application Protocol is the most widely used protocol for Wireless application. In contrast, in Japan, the most widely protocol is I-MODE whose use is driven by NTT DoCoMo. I-MODE is a technology similar to WAP, but it uses data capabilities only found on that network. Meaning that, this technology just can be used in NTT DoCoMo, Japan networks only.

To create a WAP-based site, developers normally use languages such as Wireless Markup Language (WML), a subset of Extensible Markup Language (XML). But for I-MODE site, developers use cHTML (compact HTML) or a compressed form of HTML.

WAP structure is very similar to the well-established International Standards Organization (ISO) network model with a transport protocol similar to the generally fixed-line HTTP. However, in this case, it is focused on broadcast requirements, which use less bandwidth. The WAP protocol architecture is shown in Figure 1.2 alongside a typical Internet Protocol stack. It consists of layers, which describe and specify the application/browser (WAE), sessions (WSP), transactions (WTP),

security (WTLS), transports (WDP), and bearers (SMS, USSD, CSD, IS-136, CDMA, GPRS, etc.)[5].

Figure 2.1: WAP protocol architecture [5]

Following are brief descriptions of the main features of the Wireless Application Protocol layers:

i. Wireless Application Environment (WAE)

The *WAE* is the top layer of the WAP stack and is of most interest to content developers because it contains, among other things, device specifications and the content development programming languages, WML and WMLScript. It is an application environment that is based on a combination of mobile telephony technologies and the World Wide Web.

The purpose of the WAE is to establish an environment to build applications and services. The WAE includes a microbrowser environment that defines how the wireless device interprets and presents WML and WMLScript. It also contains components that specify the following:

- WML for creating WAP applications.
- Wireless Telephony Application Interface (WTAI), which provides telephony services for WML decks running on phone-based devices

- Content formats that define a set of data formats, including images, phone book records, and calendar information.

WAE depends upon a WAP-compatible proxy server to translate between WAP and HTTP transactions and WAP and Internet Protocols.

ii. Wireless Session Protocol (WSP)

The *WSP* is a sandwich layer that links the WAE to two session services: one connection oriented service that operates above the Wireless Transaction Protocol and one connectionless service operating above the Wireless Datagram Protocol. It is basically a binary-formatted tokenized version of HTTP, designed to provide low bandwidth browser handling on long latency networks. Unlike HTTP, WSP has been designed by the WAP Forum to provide fast connection suspension and reconnection. It has also been designed to provide content push capabilities that allow unsolicited transmission of data to user agents, which in turn allows WAP device users to be alerted, for example, to incoming e-mails, telephone calls, and faxes.

iii. Wireless Transaction Protocol (WTP)

The *WTP* runs on top of a datagram service such as User Datagram Protocol (UDP) and is part of the standard suite of TCP/IP protocols used to provide a simplified protocol suitable for low bandwidth wireless stations. It offers three classes of transaction service: unreliable one-way request, reliable one-way request, and reliable two-way request response. WTP supports protocol data unit concatenation and delayed acknowledgment to help reduce the number of messages sent, and to attempt to optimize the user experience by providing the information that is needed when it is needed.

iv. Wireless Transport Layer Security (WTLS)

WTLS incorporates security features that are based upon the established Transport Layer Security (TLS) protocol standard. It includes data integrity checks, privacy, service denial, and authentication services. Developers can access WTLS by using HTTPS instead of HTTP in the URL.

v. Wireless Datagram Protocol (WDP)

The *WDP* allows WAP to be bearer-independent by adapting the transport layer of the underlying bearer. The WDP presents a consistent data format to the higher layers of the WAP protocol stack, thereby offering the advantage of bearer independence to application developers.

vi. Bearers

Below the WDP sit all of the *bearer* networks. These include Short Message Service (SMS), a facility for sending short messages; Unstructured Supplementary Service Data (USSD); Code Division Multiple Access (CDMA), for the reuse of scarce radio resources in adjacent areas; and Cellular Digital Packet Data (CDPD). Through the Internet Protocol stack, the WAP client communicates with the WAP gateway, which sits between the wireless carrier's network on one side and the public Internet or corporate intranet on the other. Gateways can be located within carrier or corporate firewalls or both. In addition to taking care of various housekeeping tasks such as keeping track of the WAP client's bookmarks and managing its cache, the WAP server handles the interface between the two sets of network protocols, wireless (WAP) and wired (TCP/IP).

2.3.4 Wireless Markup Language (WML)

WML is markup language like HTML. WML is a Document Type Definition (DTD) of XML. Pages or services created using HTML do not very well on small handheld devices (cell phone), since they were specifically develop for use on desktop computer with larger color screen. Also, low bandwidth wireless bearers wouldn't be suitable for delivering the large files that HTML pages consists of. Therefore, WML is develop to overcome these restrictions.

WML provides a navigation model for devices with small display screens and limited input facilities (no mouse and a limited keyboard). In order to save valuable bandwidth in wireless network, WML can be encoded into a compact binary format for transmission between the phone and the network and vice-versa. Encoding WML is one of the tasks performed by the WAP gateway, which is the entity that connects the wireless domain with the Internet [3].

2.4 Web Server

2.4.1 Architecture

A web server, also known as HTTP servers, provides the infrastructure for the World Wide Web. A Web server functions as a service center for information sharing, data exchange and management within Internet or Intranet. Figure 2.1 below shows the Architecture of a dynamic and data-driven website.

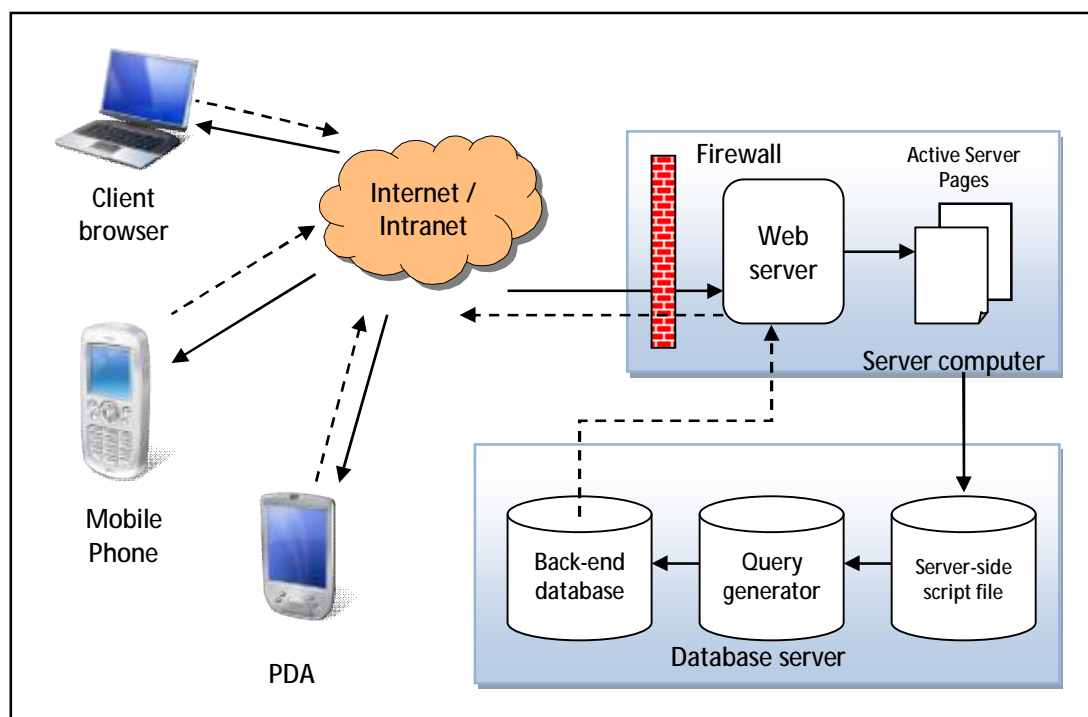


Figure 2.2: Architecture of a dynamic, interactive, data-driven website [3]

2.4.2 Internet Information Services (IIS)

Microsoft® Internet Information Services 5.1 (IIS) is a Web server that works with Windows XP Professional platform to deliver Web-based applications. IIS is a free, secure platform for building database-driven web page based on Active Server Pages (ASP) technologies [4].

2.4.3 External web server

External web server is a web hosting service that runs by other people with a complete mainframe system. Their server also will have a permanent unique Internet Protocol (IP) number. So that, user can access stored web page by typing full HTML address or IP address directly. Usually this service required user to subscribe and pay fees for web server maintenance. But this type of web hosting service is very reliable, fast, bigger space and supports a lot server language compared to free web hosting.

2.5 Dynamic page

2.5.1 Active Server Pages (ASP)

Microsoft Active Server Pages (ASP) is a server-side scripting environment that can be use to create interactive Web pages and build powerful Web applications. ASP is an open, compile-free, language-independent application technology that combines HTML, server-side scripting and robust database publishing for creating powerful, dynamic Web application [6]. Figure 2.3 below show ASP and ADO Object Model Structure.

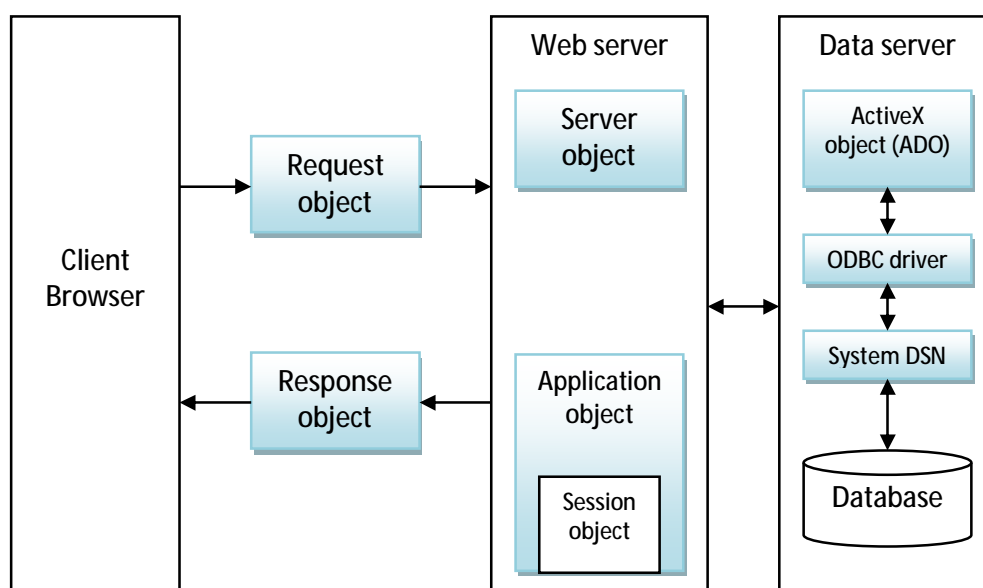


Figure 2.3: ASP and ADO Object Model Structure [6]

When the server receives a request for an ASP file, it processes server-side scripts contained in the file to build the Web page than, it sent back to the browser. In addition to server-side scripts, ASP files can contain HTML (including related client-side scripts) as well as calls to COM components that perform a variety of tasks, such as connecting to a database or processing business logic.

ASP creates this connection with the help of five primary ASP objects. It connects, or relates these objects to ActiveX Data Objects (ADO), as shown in figure 2.3. ADO is a high-level data access programming interface to an underlying data access technology (such as OLE DB), implemented by COM. While Object Linking and Embedding Database (OLE DB) is a data-access interfaces providing consistent access to Structured Query Language (SQL) and non-SQL data sources across the enterprise and the Internet [4]. The five ASP Objects are Request object, Response object, Server Object, Session object and Application object.

2.5.2 Database

Database has major impact on the growing use of computers; it plays a critical role in an almost every area where computers are used, including business, engineering, medicine, law, education and library science.

Microsoft Access, SQL Server and Oracle are all Web-compatible databases and are similar in functionality. In this project, Microsoft Access was chosen to be used as a database management. The significant are the data is not quiet big compared to e-commerce application (the migration of the data between WAP device and server it is not critical). Microsoft Access also provides additional tools for analyzing the data itself; like error checking and improved sorting control.

2.6 Device

2.6.1 Phone Simulator

Phone simulator is a program that implements a WAP microbrowser, but is designed to run on a non-WAP device, such as a Windows PC. Simulators are often used by developers to speed up the development process, as well as to reduce the costs of using WAP during testing each time a change is made. It is an application to test how wireless applications work. The simulator includes the same browser, messaging, and other application code that is embedded in real mobile phones. The phone simulator is installed with built-in WAP gateway and use Internet connection to browse WAP page in web server.

2.6.2 Mobile phone

Mobile phone is a telephone which is connected to the telephone system by radio, rather than by a wire, and can therefore be used anywhere where its signals can be received. The phone that used in this project must support WAP technologies and must subscribe to the Internet service using either General Packet Radio System (GPRS), Enhanced Data Rate for GSM Evolution (EDGE) or Universal Mobile Telecommunications System (UMTS). The mobile browser or a microbrowser that embedded in cell phone will connect to WAP gateway first, before WAP gateway transfer the request using HTTP to web server. The respond will transferred back to WAP gateway and lastly to microbrowser. Figure 2.4 shows how to display WAP content using a gateway and a simulator.

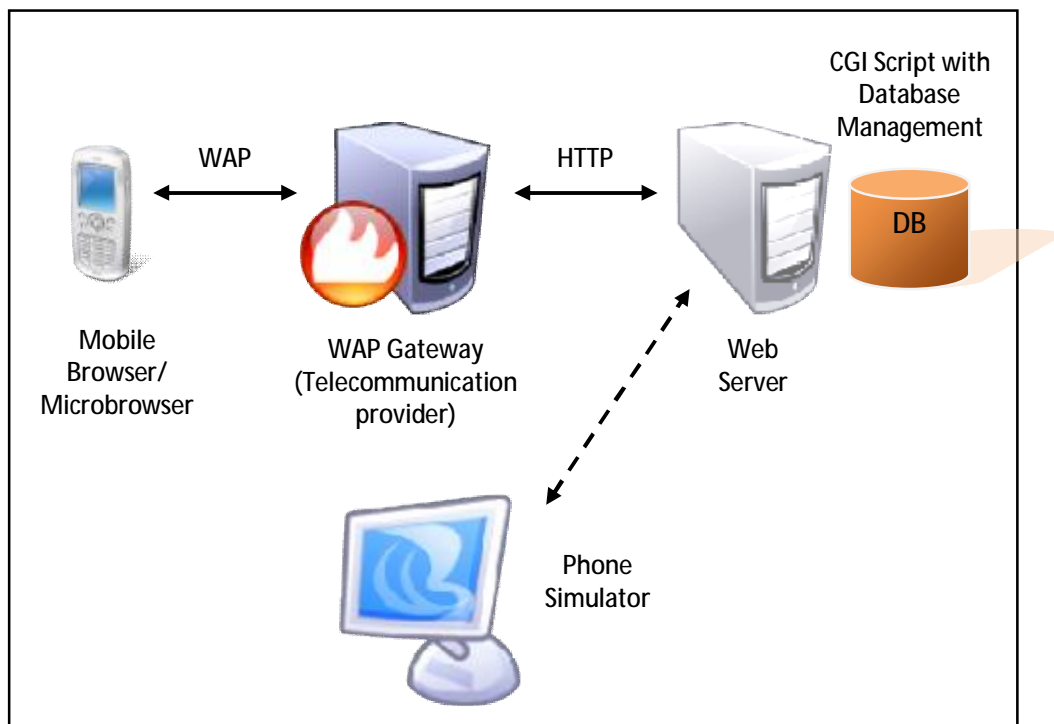


Figure 2.4: Displaying WAP content using a gateway and a simulator

2.7 Comparison

Technically, WAP technology has been used since 5-7 years ago. A lot of mobile application or a system has been developed based on WAP technology. But the difference between each system is the scripting language itself used to develop a dynamic WAP content. Examples of scripting language used nowadays are ColdFusion, ASP, PHP and Perl. These scripting techniques were used to create a powerful dynamic Web application. In this section, a comparison was made to compare between an available system and current research that implemented WAP technology. The element that counted in this evaluation is scripting language, server and data management software.

The first system is Pharmaceutical Information Processing. The system was used to test a practical approach that ported a relational database of Danish pharmaceutical catalogue using open source software [7]. The system was developed using PHP 3.0 as scripting language and Apache 1.3 as a server. For database

management, this system used MySQL 3.22. The advantages of using MySQL database over MS Access are My SQL handles multiple user connections better and is far better suited to web applications. So, it is suited to the system since the system will have a lot of user and a connection between WAP device and a web server.

The second system is Malaysian Airline System (MAS) flight schedule. This system will let user to know the exact time and date when the flight will depart. It was develop using ASP 3.0, Apache 1.3 and SQL Express database.

The third system is a Robot Interface Using WAP Phone. The system was develop by CSIRO Manufacturing Science & Technology for online control and monitoring of a robot. This system is quiet different with other type discussed in above because this system involved with robot interface. There are other communication layer used called Remote Tool Control (RTC) to let WAP communicate with robot [8]. Table 2.1 below show the complete comparison between available systems that implemented WAP technology.

Table 2.1: A comparison between available WAP-enabled systems

WAP System Element	Pharmaceutical Information Processing	MAS Flight Schedule	Robot Interface Using WAP Phone	UMP Alumni Registration
Scripting Language	PHP 3.0	ASP 3.0	PHP 3.0	ASP 3.0
Server Management	Apache v1.3 on Linux Server	Apache v1.3	QCAT webserver	IIS v5.1
Data Management Software	MySQL v3.22	SQL Express	n/a	Microsoft Access 2003

CHAPTER 3

METHODOLOGY

3.1 Introduction

The WAP Alumni Registration System developed is a web and mobile based online application that is to help graduate student register their alumni via mobile phone. It consists of mobile page of alumni`s activities and mobile online forms for the users to key in their personal details and submit the form. The data are processed by the system, and then submitting the request to the server.

The system uses a database to store all personal information like name, address, graduation year, email, contact number and occupation from the user. This sensitive information is not shown to normal user but only can be accessed by administrator only using a password.

The development model that is used in developing this system is the flow model. This model consists of five different stages that are Literature Review, Requirements Definition, Server and Client-Side Development, Integration and Simulation Testing, and finally Data Analysis. This could be seen through the model in Figure 3.1.

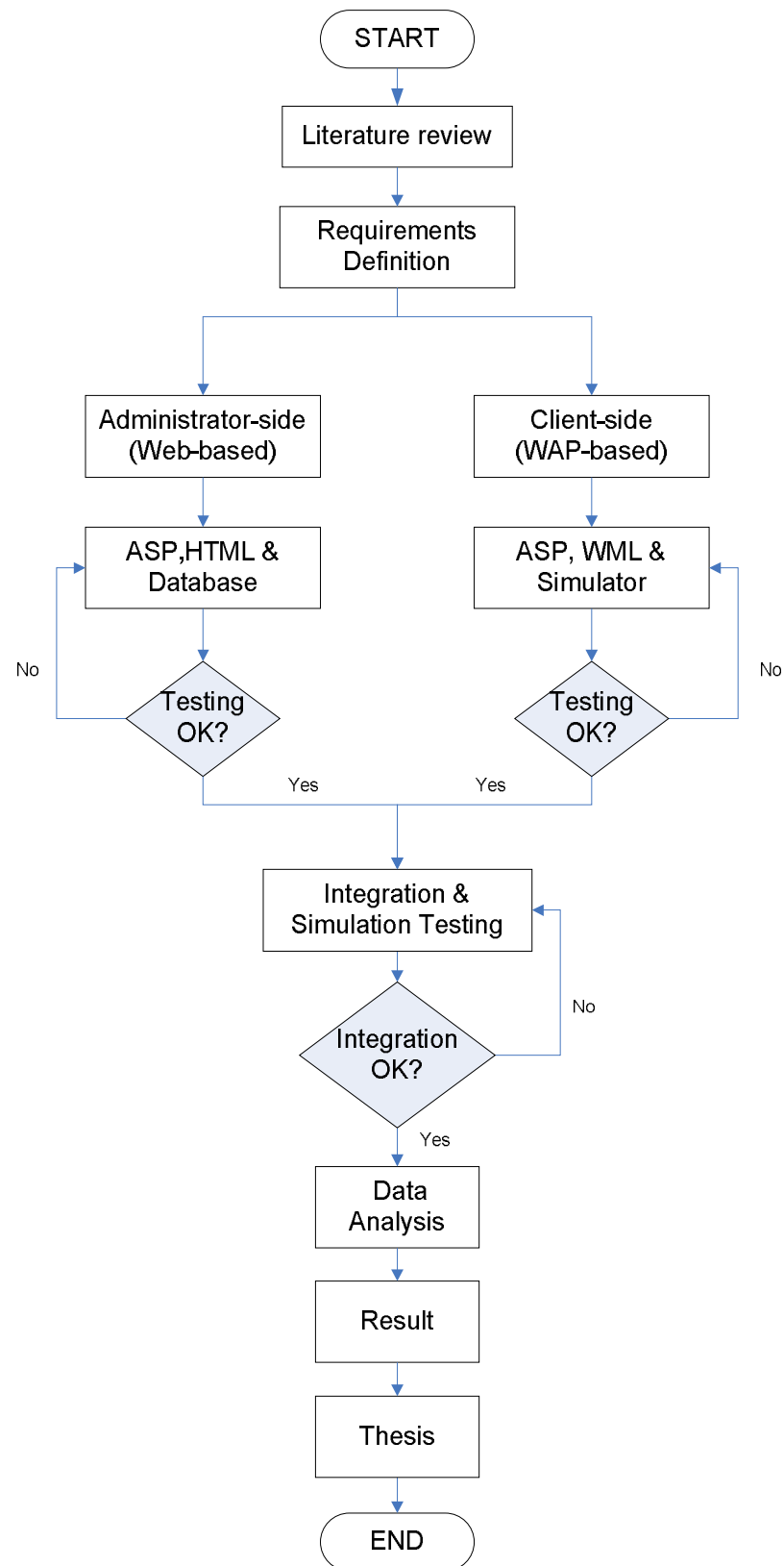


Figure 3.1: Project flow model

The system is a fully online system that cannot be obtained in offline mode. The system could not be downloaded and used when the user's WAP device is not connected to internet. However, there is an exception. During the testing phase, the system can be accessed in local web server (using IIS and it is offline from the internet). The reason for this is because the system process the information submitted by user in the web server and transfer back the request to user in separated process. So that, the sources code cannot be viewed from client browser and hence, cannot be stolen. To establish the connection between cell phone and internet, the user must first subscribe to GPRS, EDGE or UMTS (3G) service and using WAP compatible phones.

3.2 Requirements Definition

3.2.1 Development Platform

The system is developed using Microsoft Visual Web Developer Express Edition. The mark-up language used is Wireless Markup Language (WML), Hypertext Markup Language (HTML) and the web forms are done using Active Server Pages (ASP). The scripting for the system is done using Microsoft Visual Web Developer.

3.2.2 Database Requirement

The system requires less extensive of database. This is because all the information regarding alumni member like name, address, contact number, email, occupation just stored on web server in one way. Meant that, the database will not be able to update by user, just administrator can manipulate the data.

The database is designed using Microsoft Access 2003 as it could be easily integrated into Active Server Pages (ASP) script using a Database Connection Network (DSN)-less connection.

3.2.3 Web Server

Both of the internal and external web servers for this project must support Active Server Pages (ASP) to ensure that the server and client-side of scripting works. For internal web server, IIS 5.1 was used since it guaranteed supports the ASP scripting. These two technologies are designed by Microsoft to bring the power of web computing for their Windows`'s user.

For external web server in this project, NoFeeHost web hosting was chosen based on these features:

- i. Free and lifetime membership
- ii. Support Active Server Pages (ASP) 3.0
- iii. 100 MB space storage
- iv. Support Microsoft Access database
- v. Using Windows 2003 IIS 6
- vi. Support File Transfer Protocol (FTP)

3.2.4 Hardware and Software Specification

Generally, the hardware required in this project are a computer and a cell phone. The computer requirements used in this project are:

- i. Intel Pentium M 1.4 GHz
- ii. 1GB Random Access Memory (RAM)
- iii. At least 1 GB free hard disk space
- iv. Microsoft Windows XP Professional Edition with IIS 5.1

The cell phone requirements are:

- i. WAP-enabled (using either GPRS, EDGE, 3G services)
- ii. A display with minimum resolution of three lines of twelve fixed-width characters
- iii. Support text entry
- iv. A minimum of two programmable user interface mechanism

The other software used for this project are:

- i. Openwave V7 Phone Simulator
- ii. Internet Explorer Browser
- iii. CuteFTP 8.0 Professional

Openwave V7 Phone Simulator is a windows-based application that used to test how wireless application work. The simulator includes the same browser, messaging, and other Openwave application code that is embedded in real mobile phones. The version of this Openwave Simulator can display content delivered in a number of formats, including XHTML Mobile Profile 1.0 (XHTML-MP) with Cascading Style Sheets (CSS), and Wireless Markup Language (WML).

Internet Explorer browser is free, and comes with any Microsoft operating system that used as a platform to surfing the internet. In this project, Internet Explorer is used to display the administrator side of the system. The reasons of using this browser compared to another browser are lot of people was using this browser as their default browser and it consumed less memory of Random Access Memory (RAM). Thus, the environment of Internet Explorer can be taken as reference in developing this system.

CuteFTP 8.0 Professional is a powerful tool that used to maintain the web sites in external web server. Using this software, administrator can easily and securely upload and download the files within the external web server.

3.3 Administrator-side Development

3.3.1 Microsoft Access 2003

This software is used to create database of graduate student. A database is a collection of information that's related to a particular subject or purpose, such as indexing the books in library by its title or ISBN. Microsoft Access 2003 is database management system software.

Using Microsoft Access, user can manage all the information from a single database file. Within the file, the data can be divided into separate storage containers called tables. User also can view and update the data using online forms; find and retrieve the desired data by using queries; and analyzed or print data in a specific layout by using reports.

There are four steps in designing the database for this project:

i. Determine all field

Gather all the data that application will use and study how the output will use the data. In this project, the related fields are Name, Address, Year Graduated, Phone Number, Matrix ID and Occupation.

ii. Group related field

All this fields are group into *users* table. These fields are entered accordingly based on the desired system design and all the data type is determine correctly to ensure there is no error when integrating with ASP codes.

iii. Create table

The other table created is *reg_user* and *news*. *Reg_user* table is to stored table for username and password's field for admin to access the administrator site.

iv. Normalized the data

Normalize the data means to eliminate redundancy. Redundancy will leads to errors. The solution to avoid this problem is normalizing the data.

3.3.2 Active Server Pages (ASP) coding

i. DSN-less connection

There are many methods to link-up the document with the database management system (DBMS) such as using (Open Database Connectivity) ODBC DSN which is come along with IIS 5.1 and DSN-less connection using SQL OPEN statement. For this project, the method used is DSN-less connection. The reasons using this method are easier to link-up with Access database and the syntax will take only 3 lines. This method also well-suited with the web server since web server required a DSN-less connection to link-up the database.

```
1 CONNECT="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &  
2 Server.MapPath("db\USERMAN.MDB")  
3 Set DBCONN=Server.CreateObject("ADODB.Connection")  
4 DBCONN.OPEN CONNECT
```

Figure 3.2: DSN-less connection tag

The first and second line is to specify the type of database and the location of database file used. At line 2, the server map path is used to specify the location of the database include its subfolder. In this case USERMAN.MDB is the Access database name. The last line is to open a connection to selected database.

ii. SQL statement

Structured Query Language (SQL) Statement is used to make or call a data to do some operation like *update*, *insert* or *delete* a record in the database. SQL statement tag in figure 3.3 tells the server to do a delete operation.

```
1 sql = "DELETE newsno FROM news WHERE newsno= " & padam &" "  
2 conn.Execute(sql)  
3 conn.close
```

Figure 3.3: SQL statement tag

The first line of statement tag tells the server to delete a *newsno* record from *news* table based on *newsno* condition. *Newsno* is the reference number for each record. *Newsno* condition will be passed from another files using *postfield* method and administrator will choose what number of record they want to delete. If the number is available with the existing record, then it can be deleted.

The second line tells the server to execute/run the SQL statement. While the last line tells the server to closed-up the connection with the database.

3.4 Client-side development

3.4.1 Architecture

This section will give a detail overview on how the client-side development systems work.

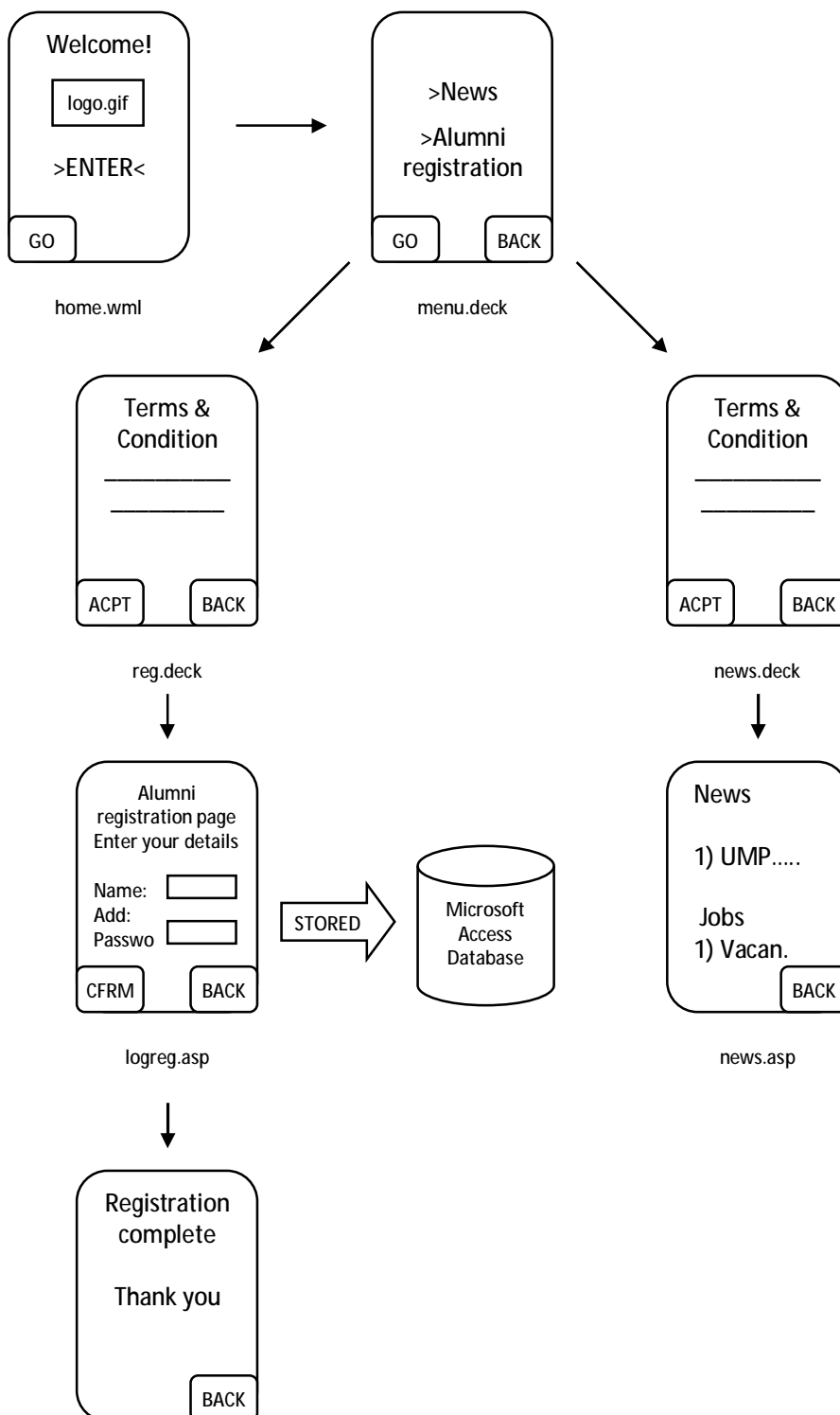


Figure 3.4: Architecture for client-side development

3.4.2 MIME type configuration

MIME is a short form for Multi-purpose Internet Mail Extensions. It is a way of configuring browsers to view files that are in multiple formats. An extension of the Internet mail protocol that enables sending 8-bit based e-mail messages, which are used to support extended character sets, voice mail, facsimile images, and so on [4].

These MIME type is needed by the browser to determine how the file should be processed. The file types used for WAP devices have a new set of MIME types and the web server must have these types added. The figure for the MIME configuration is shown below. The MIME type for *.wml* is *text/vnd.wap.wml*. So this type must be added into MIME Map to ensure that the WAP device can read this file type.

The MIME mapping button can be selected after web sites properties window showed up. Web sites properties menu can be access by right clicking at desired web page folder in Internet Information services window.

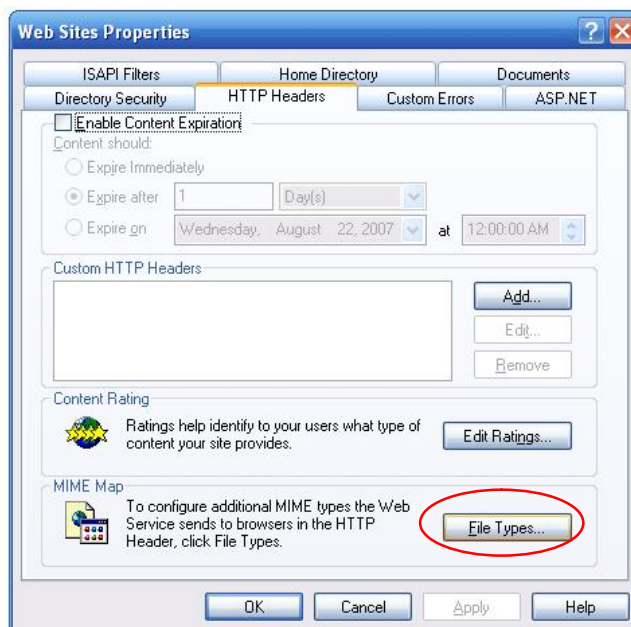


Figure 3.5: MIME type configuration menu

3.4.3 Active Server Pages (ASP) Coding

i. WML Document Prologue

Every WML deck that written contains a document prologue at the top line before the codes. Compilers on the device, WAP gateways, and remote servers all use document prologues to interpret the code. Below is the document prologue in WML:

```
1 <%Response.ContentType = "text/vnd.wap.wml"%>
2 <?xml version="1.0">
3 <!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
4 "http://www.wapforum.org/DTD/wml_1.1.xml">
```

Figure 3.6: WML document prologue tag

The first line of prologue tells the web server to return the contents with a MIME type for WML instead of ASP file. This is because the WAP device or microbrowser will reject the ASP type file, as it will only accept only WML files.

The second line of the prologue designates the XML version of the WAP server and WML compiler. WAP servers and WML compilers use XML to interpret your code. These servers and compilers then transform this information back into WML, so that a WAP device can display the information.

The third line of the prologue defines the version of WML used. This line of code states the WML version used is version 1.1.

The fourth line specifies the location of the WML document type definition (DTD). In this prologue, it is refer to WAP Forum's site. Any additional extensions or information for the WAP server or compiler are available from this site.

ii. Cache control tag

```
<meta http-equiv="Cache_Control" content="max-age=0" />
```

Figure 3.7: Cache control tag

This tag makes sure that the card is not read from the cache after the user has finished with it, but instead that the deck is re-read from the server. Using this tag ensures that there are no problems with old content if the user has to come back and enter his or her ID again. It also ensures that anyone picking up a lost phone will not automatically have access to the password by looking through the cache.

iii. DSN-less connection

The method used to link up with the database for client side development is also same with the administrator side, using DSN-less connection. Figure below shows the DSN-less connection for client side development.

```
1 CONNECT="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &  
2 Server.MapPath("db\USERMAN.MDB")  
3 Set DBCONN=Server.CreateObject("ADODB.Connection")  
4 DBCONN.OPEN CONNECT
```

Figure 3.8: DSN-less connection tag

iv. SQL Statement

```

1 sql = "UPDATE Users SET Address1=' " &
2 Request.QueryString("Uadd1") & "' " & "WHERE MatrixID=' " &
3 Request.QueryString("MatID") & "' "
4 conn.Execute(sql)

```

Figure 3.9: SQL statement tag

The function of SQL statement is same as describe in administrator-side coding. The first line in tag above tell the computer to update a *Users* table and set *Address1* into the requested data from *Uadd1* that have been pass from a previous postfield value.

In line 2, *where* option is used to verify the statement whether it is match with the existing record or not. If it is not, then the new record will not save. The last line is to tell the server to run/execute the SQL statement.

v. Tidying Up

```

1 rsUserAdd.CLOSE           ` close the Recordset object
2 SET rsUserAdd = NOTHING   ` release Recordset object mem.

```

Figure 3.10: Close tag

After finished with the Recordset object, the statement above is include to tidy up and close it down, thereby releasing any memory used. The statement above can be put at any point where the Recordset object have finished.

3.5 Integration & Simulation Testing

3.5.1 Localhost Operation

This part will explain about integration and simulation testing using localhost server, which mean the server and simulator will run in the same network (intranet) or use the same computer.

To run the server (IIS 5.1) in Windows XP, the computer must be configured properly. The first step, IIS must be installed using Add or Remove Program menu in Control Panel. After that, click at Add/Remove windows component and tick at Internet Information Services (IIS). Then click ok. After that the web server will run. To ensure whether the web server is run, the browser was point to *http://localhost*. A confirmation page will appear if the web server is successfully run.

After setting a Web server, Windows will create a folder named *Inetpub* under C:\ directory. The entire involved file like *.asp*, *.wml*, *.html* including the database file (*.mdb*) must be created in one single folder and must be put under *Inetpub* directory. This will ensure web server to process the ASP files.

3.5.2 Openwave Simulator

Openwave Simulator V7 is a powerful simulator that can be used to test the WAP page directly, either using an internal or external web server. To access the WAP page within the intranet connection or the host, just enter the Internet Protocol (IP) address or URL address at *http://localhost* and followed by the directory of the page. The example is shown below.



Figure 3.11: Openwave simulator V7 main screen

3.5.3 External Web Server

Basically, the external web server is a slightly same with the internal web server. The difference is the computer/server run by other people. For this project, NoFeeHost web hosting was used as an external web server. After register with this web hosting service, user will be given one account with password to upload their page into the server. To access the page that uploaded into this web hosting, user/administrator must type the correct address that given by this company. The address is *http://users3.nofeehost.com/umpnet/home.wml*.

3.5.4 FTP Configuration

File Transfer Protocol (FTP) is the protocol used for copying files to and from remote computer systems on a network using Transmission Control Protocol/Internet Protocol (TCP/IP), such as the Internet. This protocol also allows users to use FTP commands to work with files, such as listing files and directories on the remote system [4]. For this project, software used as a FTP client is cuteFTP 8.0 Professional. After registration with NoFeeHost web server, the information such as username, password and host address must be key-in into cuteFTP software to establish the connection between computer and virtual directory at NoFeeHost web hosting. Figure 3.5 below shows the site properties window in cuteFTP.



Figure 3.12: Site properties window in cuteFTP

3.6 Data analysis

This part will discuss about the limitation and comparison of the system. Theoretically, the size of database will increase constantly due to increasing in registration request. When the system is upload to external web server, this limitation must be concern. At NoFeeHost web hosting, the space given is only 100MB. So, if the data expand till the maximum size provided, there will be an error in database system. The number of registered person is unlimited. But the space storage will be a problem. So, the number of maximum registered person cannot be determine since the entered personal information between persons is not same. To overcome this problem, a large space web hosting, usually paid web hosting can be used as an external web server. The other option is administrator must replace this packed database with a new, blank database into the web server. So that, another person can register back their alumni.

CHAPTER 4

RESULT

4.1 Administrator side

A data driven interactive WAP alumni registration system was established. The application can be tested at <http://users3.nofeehost.com/umpnet/login.htm> by using Internet browser such as Opera and Internet Explorer. The login information to enter the administrator site is as below:

Username: admin

Password: password

Response time for accessing a login page was usually less than five second on a stable network connection. Viewing the alumni list was about 5 seconds also. The maximum amount of data received during this session is about 300 KB.

4.1.1 Default & Sign-in page

This administrator-side development phase was simulate using Internet Explorer Browser. At the default page, administrator must key-in their username and password to enter the restricted administrator page. If the password or user name is incorrect the system will inform to enter it again. If the username and password is correct, administrator will be redirect to menu selection page. The figure for default page was shown below.

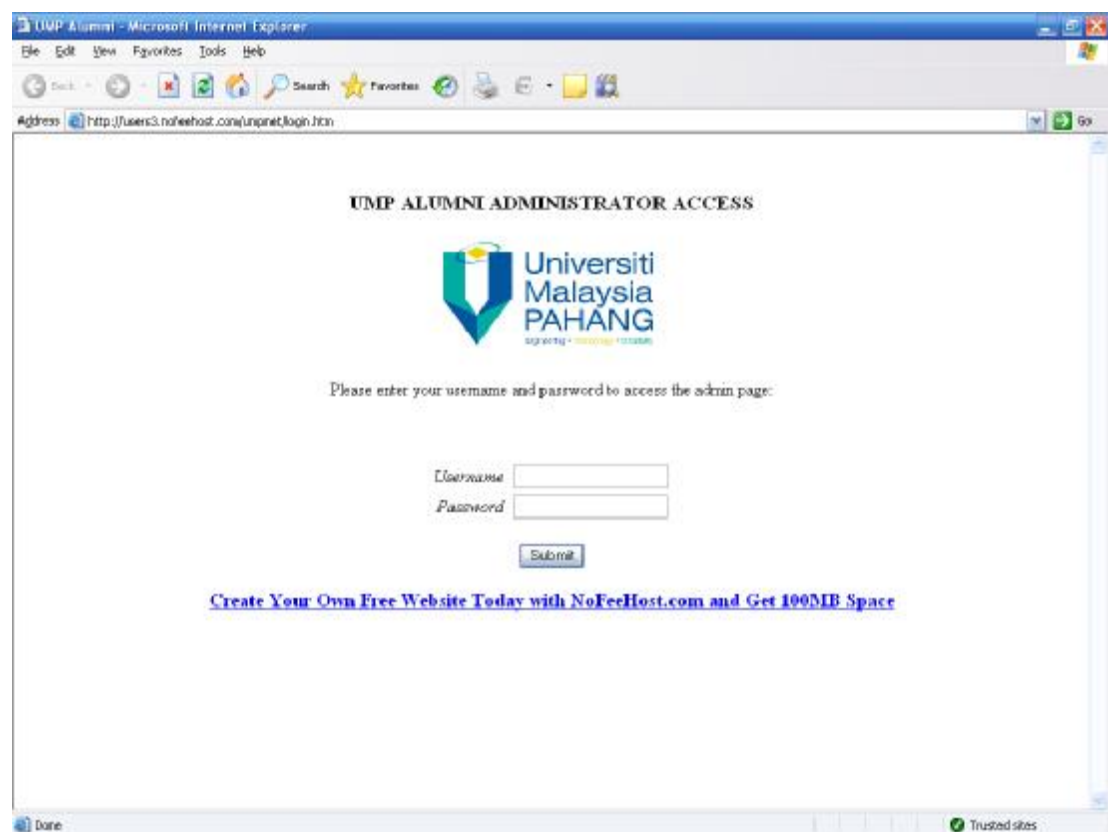


Figure 4.1: Administrator sign-in page

4.1.2 Menu selection page

In this page, administrator can select their desired menu display. They can select either news entering menu or alumni member list. News entering menu will open a new page that let the administrator to key-in the latest news and job vacancies within the university, while the member alumni list view menu will show a complete table of registered alumni member. Figure 4.2 below shows the selection page.

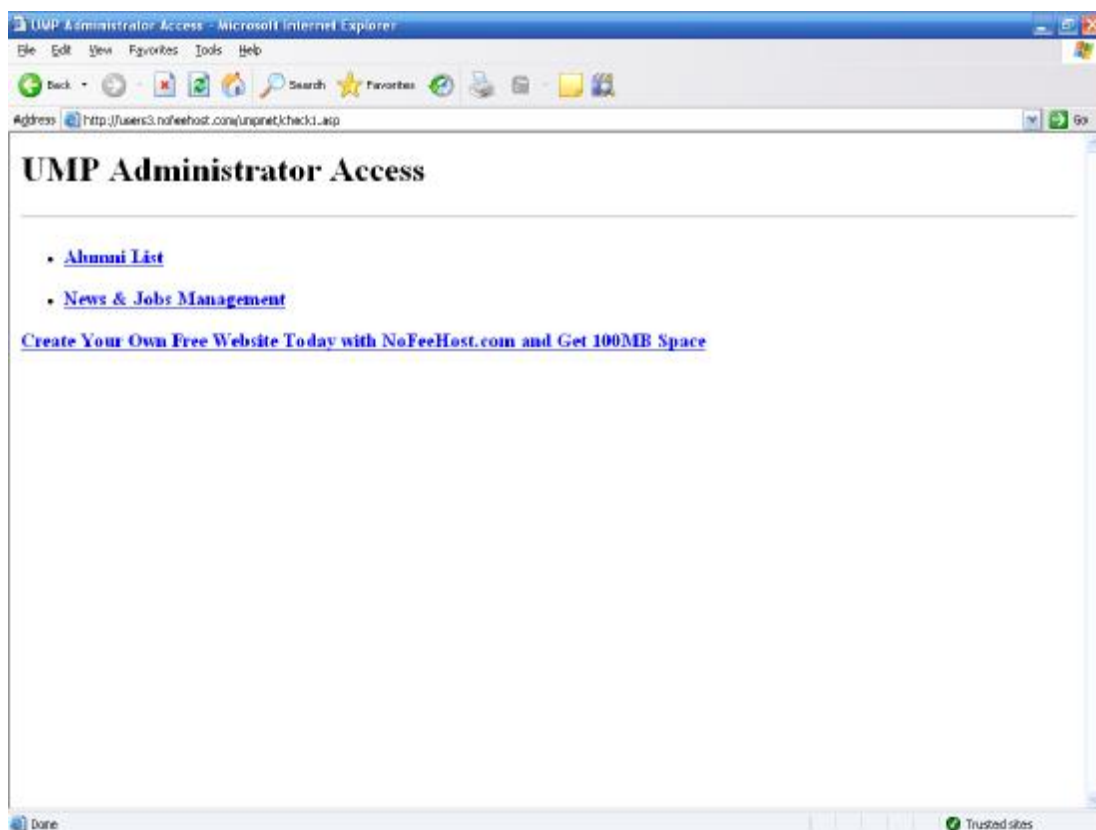


Figure 4.2: Menu selection page

4.1.3 Alumni view list page

This page will enable administrator to view an entire page that shows the registered alumni member. Administrator also can delete, print, copy this information and export it into Microsoft Excel. This information can be sorted according to their category and be manipulated in Excel. Thus, it will help a lot in doing data analysis phase. The page is shown below.



The screenshot shows a web browser window titled 'UMP Alumni Member List - Microsoft Internet Explorer'. The address bar shows 'http://users3.noiphost.com/umpnet/alex2.asp'. The main content area displays the 'Universiti Malaysia Pahang Alumni List' with the following table:

NO	NAME	MATRIX ID	PASSWORD	ADDRESS 1	ADDRESS 2	GRAD YEAR	PHONE	OCCUPATION
1	MOHAMAD ZULFIRDAUS BIN MOHAMAD DIN	EG04010	LAT	BUKIT MERTAJAM	PENANG	2008	60194024771	ENGINEER
4	MOHAMMAD HADI BIN ISMAIL	EG04007	41869	EG FT ABD RAHMAN,MK4,SPG KIRL83000	BATU PAHAT, JOHOR	2008	60127642897	TRAINEE
5	MOHD ARIFFUDDIN BIN MOKHTAR	EG04008	159873	6075 LRG 7, TMN DAMAI	PONTIAN, JOHOR	2008	60172040597	ENGINEER
6	ANIS ZAFIRAH	EG04017	ANIS	103, TMN ORKID, FASA 4, 08000	SUNGAI PETANIL, KEDAH	2008	60194240626	ENGINEER
15	KHALIDUL ANWAR B ISHAK	EG04027	mduj	1373 LAHAR YOOI	13300 TASEK GELUGOR, P. PINANG	2008	60194581373	ENGINEER
16	Siti fatimah	EG04012	bedah	Blok b-2-5, Jem bukat galing, 25990 kuantan	Kuantan	2008	60196076996	Petani
17	Nor Robairi	EG04025	safwan	Eg kandang, kota setia, 36000 tekuk intan, perak.	Tekuk Intan	2008	60199100964	Suamimah
18	MARLEENA AKMAL	EG04006	akmal	30, JLN LEP 4/15, TMN LESTARI PUTRA	BANDAR PUTRA PERMAI, SELANGOR	2008	60137253545	CEO

Below the table, there is a form to delete a record by Matrix ID (Case Sensitive):

Delete a record by Matrix ID (Case Sensitive):

Figure 4.3: Registered alumni member view page

4.1.4 News management page

This page enables administrator to place the latest news and activities regarding the university. Besides, job vacancies also can be put in this page to inform the user who surf this WAP page. Administrator can manage the database easily without download it first. The unwanted news or job can be remove by typing its number. Figure 4.4 below shows news management page.

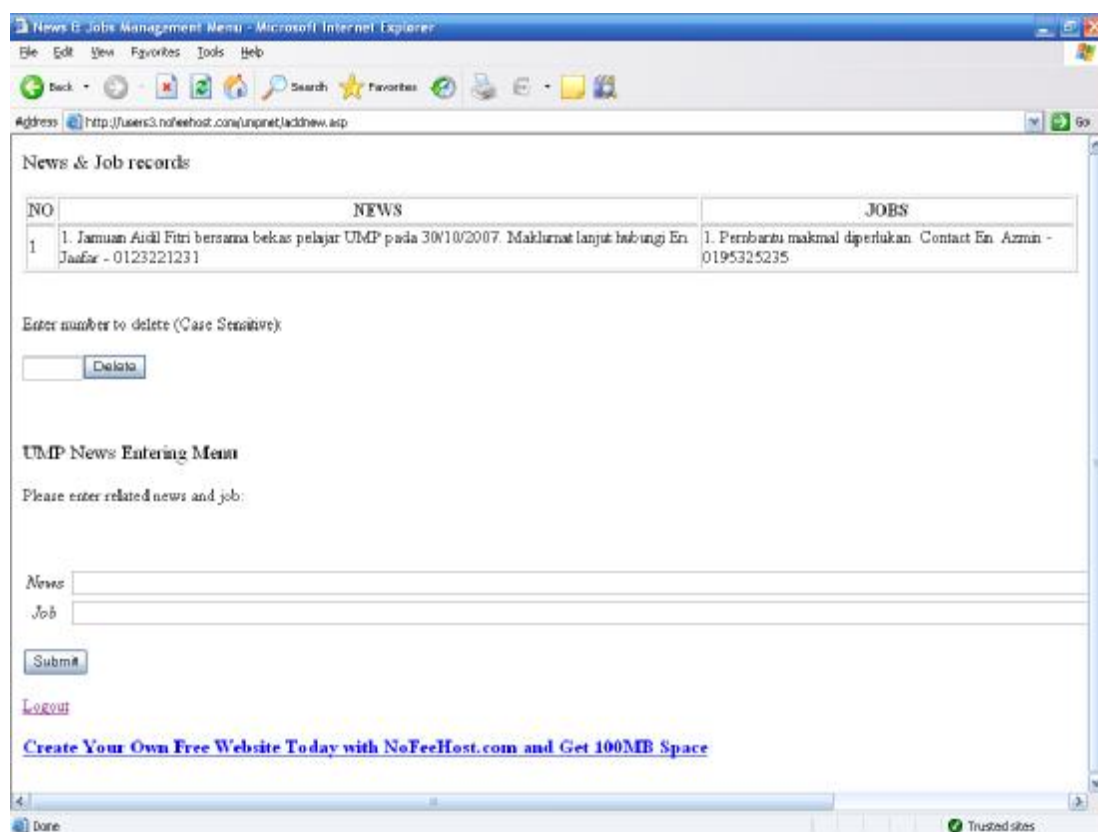


Figure 4.4: News management page

4.2 Client side

A data driven interactive WAP alumni registration system was established. The application can be tested at <http://users3.nofeehost.com/umpnet/home.wml> by using a public gateway or by using phone simulator. Response time for accessing the main/home page was about 10 seconds. Submitting the alumni information takes about 5 seconds for processing. The application was tested on Nokia 6300 WAP phones. Several device-specific limitations were revealed. The display resolution is 240 x 320 pixels allowing 8 lines of text to be displayed. The maximum amount of data used was about 50 KB for a complete registration process. The result afterwards will show both simulation and real device implementation.

4.2.1 Home & selection page

The default page shows the welcome greeting and enter menu. After select the enter menu, user will redirect to second page. This page enables user to choose either news, alumni registration, member search or profile update page. After make a selection, user will redirect to terms andc page that explain to them about the terms before they become a member. The figure is shown below.

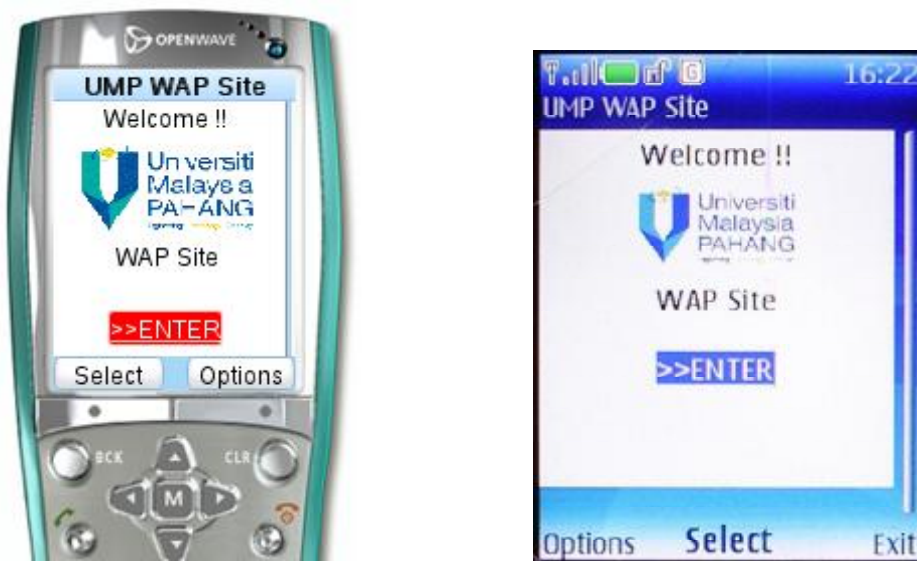


Figure 4.5: Default page for client-side



Figure 4.6: Selection page for client-side

4.2.2 Registration page

In this page, user can enter their personal information to register as alumni member. The information includes name, password, address (hidden), address (visible), matrix id, graduation year, contact number and present occupation. The password will be used to let user update their profile and for future function. After entering all these details, user must press confirm button to submit the request. After that, a confirmation message will appear if the request was successfully recorded into database. The figure is shown below.

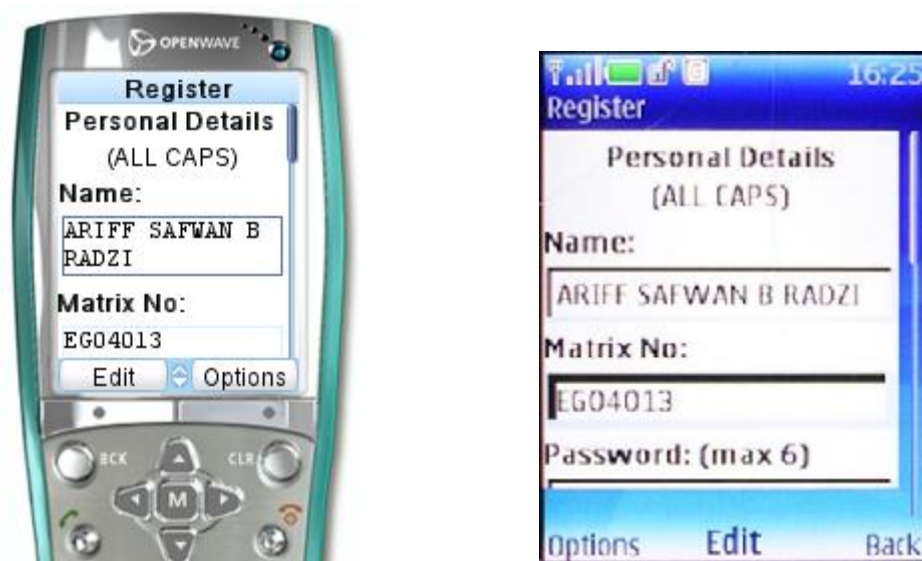


Figure 4.7: Registration page for client-side

4.2.3 News page

Within this page, user can get the latest news, information and activities regarding the university. All this information must be entered into database first by administrator before the user can access it. The figure of this page is shown below.

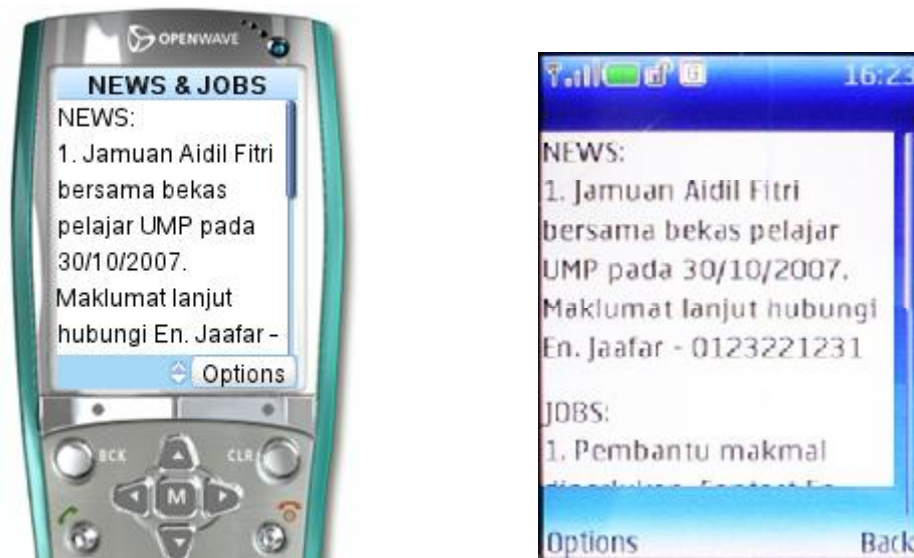


Figure 4.8: News page for client-side

4.2.4 Member search page

In this page user will have an access to search their registered member within the system. What they need is the name of their member that not exceed of 8 characters. After pressing the search button, the result screen that show their member name, address and phone number will appear. Note that this system will show the address that only visible to others. Figure below shows the search page.



Figure 4.9: Member search page for client-side



Figure 4.10: Member search result page

4.2.5 Profile update page

This page will let user to update their profile. They must key-in the matrix id with their password to enter the update menu. After correct login, user can choose which personal details they want to update like address, phone number and occupation. To update a record, user must re-enter the matrix id as a confirmation step. The figure is shown below.

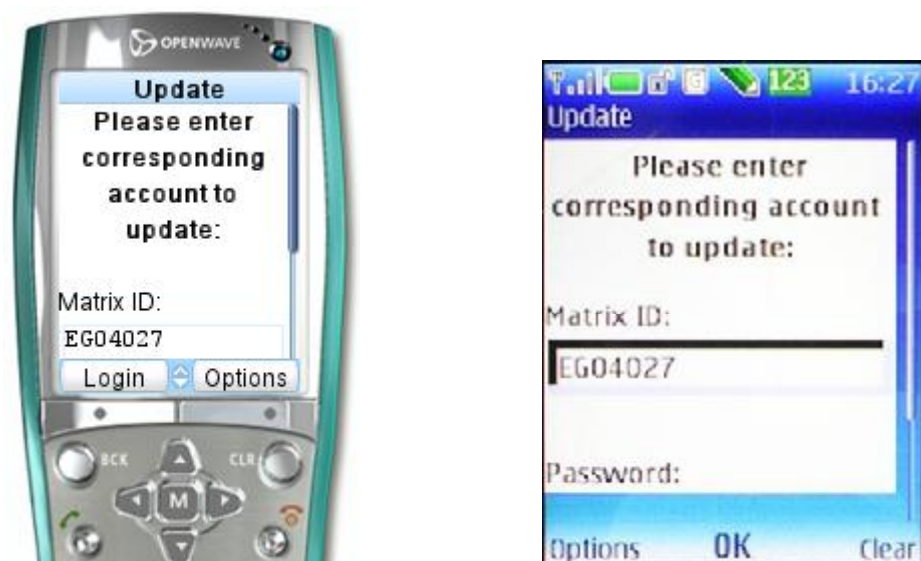


Figure 4.11: Profile login update page

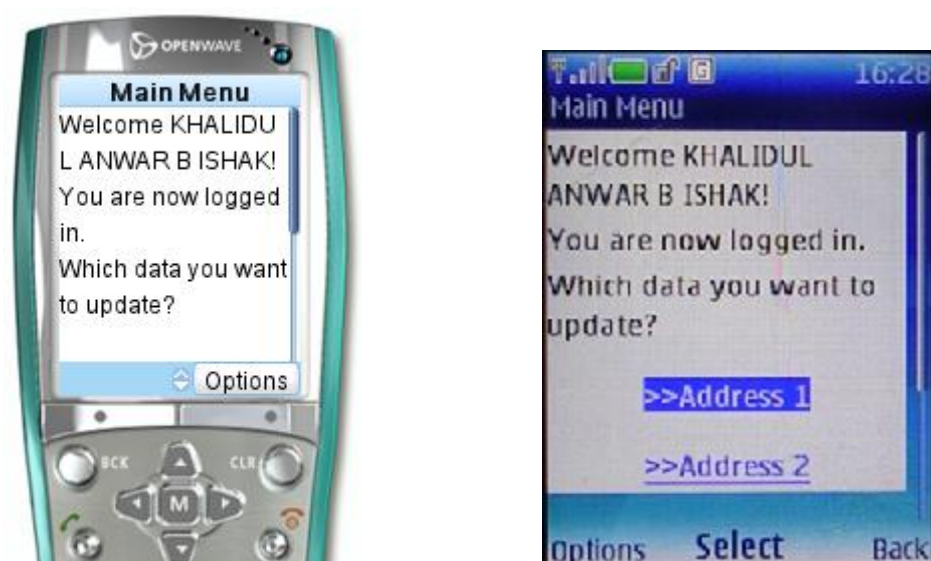


Figure 4.12: Profile update page

4.3 Constraints

In designing this system, there are few constraints that must be overcome such as:

- i. The display of each cell phone is different in terms of resolution and colour display. So this system will display differently on different cell phone. The solution was used a standard display for modern phone which is 240x320 pixels. The cell phone used in this project was Nokia 6300 that support the standard resolution.
- ii. The connection of GPRS is a little bit slow and not stable. This will affect the time to process the script and the GPRS cost usage. To overcome this problem, Celcom Unlimited Daily Broadband was used to test the system and upload the codes into web server. This package was offered by Celcom at RM 8 per day.
- iii. The available free external web hosting that supports ASP technology is harder to find and its storage is limited. The solution taken was used NoFeeHost web server as describe in section above.

CHAPTER 5

DISCUSSION & CONCLUSION

5.1 Discussion

Based on the research that has been done, there are a few discussions that can be sum up. Below is the list of related discussion:

- i. All the setting of the related software must be configured properly to ensure that there is no codes error/compile error at the output phase.
- ii. The speed to access the WAP page is depend on both type of connection and network bandwidth of the service (telecommunication provider).
- iii. One main reason the external web server was used instead of IIS is the user outside the intranet network or using a real cell phone cannot access the system because the network in UMP use Dynamic Host Configuration Protocol (DHCP) technology built with server firewall. The entire computers connected with UMP network will split up main IP address that given by Internet Service Provider (ISP). So, the other computer outside UMP can't access the internal server.
- iv. All WAP application must be tested first on real cell phone to ensure that the page can be viewed clearly and meet the requirements needed. There are a few constraints that must be considered when testing using real cell phone like [9]:
 - Caching – a technology that enables cell phone to stored visited WAP site information on phone memory to speed up the connection when revisited to that WAP site.

- Cookie support – a technology that enables cell phone to remember the submitted information to web server like login information and password.
- GET and POST method – a method that microbrowser pass or submit the information to the web server. All phone simulator support both of this method, but some of cell phone just support only one of this method.
- Look and feel – for different type of cell phone model, there will be a different in terms of view and corresponding keys. Hence, testing on real device is the only way to ascertain the look and feel of this application.
- Maximum size of WAP binary – every cell phone have its maximum size of WAP binary as the information transferred from the web server to the WAP device are compiled into bytecode format know as WAP binary. The problem will occurred if the size of WAP deck is larger that supported WAP binary size. Because of this, the WAP binary that is sent to device must not exceed its memory capacity

5.2 Conclusion

This project opens perspectives for a further integration of WAP phone functions in alumni information processing. The university also can manipulate the WAP technologies to design a new system that give benefit to university`s members. WAP system can be implemented as mobile student portal. Students and staffs can access the system to read, sent memo, check their pointer and use all the components like embedded in HTML portal with their fingertips. An independent WAP gateway may be placed within university firewalls, which may be an advantage with respect to security. However, if WAP phones are to become an effective tools for students, special attention must be paid to the limitations of the devices. The viewable area and text entry must be designed properly to suite with the cell phone.

5.3 Future Recommendations

There are a few recommendations to improve this project

- i. The WAP page can be optimize and add some interactive menu using Cascading Style Sheet (CSS) technology.
- ii. The university can placed a WAP gateway within it firewall to let other people access this system.
- iii. The other feature like a forum can be added in the WAP portal to let alumni member exchange the information.
- iv. Alumni member also can upload photo into WAP portal to facilitate others in finding process.

5.3.1 Costing and Commercialization

Generally, the cost to develop this project is about RM 50.00. Since it is not involved any hardware, the cost only covered the GPRS service usage to upload and test the WAP application. The system also used a free web hosting service, so there is no renting fee for the service. However the space is limited to 100MB for small or medium database storage. Paid web hosting service should be use for larger database storage.

This project has a potential to be commercialized especially in university and educational sector whereby involved former students society. The commercial value for this project is depends on developer himself, based on the programming level and customer requirements. As a conclusion, this project can be modified and commercialized by the customer to suite on any system that needs WAP device as a communication platform.

BIBLIOGRAPHY

- [1] Breazeale, Sandra Elizabeth Uardy, (1999) : *The Alumni-L Listserv*
http://wwwlib.umi.com/dissertations/preview_page/9937989
- [2] PDA Cortex homepage
<http://www.pdacortex.com/glossary.htm>
- [3] Dale Bulbrook (2001) : *WAP: A beginner`s guide*. Osborne / McGraw Hill
- [4] Microsoft Internet Information Services 5.1 (IIS) Documentation
<http://localhost/iisHelp/>
- [5] Paul Ashworth, (2001) : *Getting Started with WAP and WML*. Sybex Inc.
- [6] H.M. Deitel, T. R. Neito (2002). *Wireless Internet & Mobile business: How to program*. N.J, Prentice-Hall
- [7] Jens Dorup, Wireless access to a pharmaceutical database: A demonstrator for data driven Wireless Application Protocol applications in medical information processing. *J Med Internet Res* 2001; 3(1): e4
<http://www.jmir.org/2001/1/e4>
- [8] Pablo d' Angelo, Peter Corke (2001). *A Robot Interface Using WAP Phone*. CSIRO Manufacturing Science & Technology
<http://www.cat.csiro.au/cmst/automation>

- [9] Soo Mei Foo (2001) : *Dynamic WAP Application Development*. Manning Publications Co
GSM (2000a), *An overview of wireless application protocol*
<http://www.gsmworld.com/technology/wap.html>
- [10] W3schools homepage
<http://www.w3schools.com/wap/default.asp>
- [11] 3G Newsroom homepage
<http://www.3gnewsroom.com/html/glossary/w.shtml>
- [12] Margaret van Steenderen (2002) : *Business applications of WAP*. The electronic library
Available at <http://www.emeraldinsight.com/0264-0473.htm>
- [13] Chris C, Anne J, Peter E, Terry S (2002) : *Development of effective WAP and m-commerce service*. Aslib Proceedings
Available at <http://www.emeraldinsight.com/0001-253X.htm>
- [14] Whitehead P. (2000) : *Active Server Pages 3.0*. IDG Books Worldwide Inc
- [15] Kris Jamsa (2002). *WML & WMLScript: A Beginner`s Guide*. Osborne/McGraw-Hill
- [16] Iskandar A. R., Zaiton Ismail (2004): *WMLScript untuk Laman WAP Telefon bimbit*. Venton Publishing
- [17] Marc Ramsay (2001) : *Midley Irritating: a WAP usability study*. Aslib Proceedings
- [18] Chris Tull (2002) : *WAP 2.0 Development*. Que Publishing

APPENDIX A

PROJECT SCHEDULE

PROJECT SCHEDULE

ID	Activities	Jan 2007			Feb 2007				Mar 2007				Apr 2007				May 2007				Jun 2007				Jul 2007				Aug 2007				Sep 2007				Oct 2007				Nov 2007					
		1/7	1/14	1/21	1/28	2/4	2/11	2/18	2/25	3/4	3/11	3/18	3/25	4/1	4/8	4/15	4/22	4/29	5/6	5/13	5/20	5/27	6/3	6/10	6/17	6/24	7/1	7/8	7/15	7/22	7/29	8/5	8/12	8/19	8/26	9/2	9/9	9/16	9/23	9/30	10/7	10/14	10/21	10/28	11/4	11/11
1	Title	█																																												
2	Literature review	█																																												
3	Proposal																																													
4	Correct proposal & resubmit																																													
5	Preparation for seminar I																																													
6	Seminar I																																													
7	Report																																													
8	Develop Administrator-side - create database using MS Access, asp and html - testing - troubleshoot																																													
9	Develop Client-side - script using wml and asp - testing - troubleshoot																																													
10	Integrate server with client																																													
11	Testing/simulating																																													
12	Troubleshoot																																													
13	Result																																													
14	Analysis																																													
15	Preparation for seminar II																																													
16	Seminar II																																													
17	Thesis	█																																												
18	Submit thesis																																													

APPENDIX B

PROJECT CODES FOR
ADMINISTRATOR-SIDE DEVELOPMENT

```
<html>
<head>
  <title>UMP Alumni</title>
</head>
<body style="text-align: center">

<Form name = "request" action ="check1.asp" method ="post">
<p style="text-align: center">
  <span style="font-size: 14pt"><strong></strong></span>&nbsp;</p>
  <p style="text-align: center">
    <span style="font-size: 14pt"><strong>UMP ALUMNI ADMINISTRATOR
ACCESS</strong></span></p>
  <p style="text-align: center">
    &nbsp;</p>
  <p style="text-align: center">
    Please enter your username and password to access the admin page:
  </p>
  <p>
    <strong></strong>&nbsp;</p>
<table border ="0">
  <tr><td align = "right"><i>Username</i></td>
    <td><input type ="text" size = 20 name = "username"></td>
  </tr>
  <tr><td align = "right"><i>Password</i></td>
    <td><input type ="password" size = 20 name = "password"></td>
  </tr>
</table>
<p style="text-align: center"><input type ="submit" value ="Submit" />
  </p>

</Form>
</body>
</html>
```

```
<html>
<head>
  <title>UMP Administrator Access</title>
</head>
<body>

<%

dim myConnection
dim rsTitleList
dim connectString
dim sqlString
dim username
dim password

connectString="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.
  MapPath("db\USERMAN.MDB")
Set myConnection=Server.CreateObject("ADODB.Connection")
myConnection.OPEN connectString

set rsTitleList = Server.CreateObject("ADODB.Recordset")

username = Request.Form("username")
password = Request.Form("password")

sqlString = "SELECT * from reg_user where username = '" & username & "'"
set rsTitleList=myConnection.Execute(sqlString)

if (rsTitleList.BOF) and (rsTitleList.EOF) then
response.Write("Sorry, invalid username or password. Please try again")
%>

<br><br><font size=3>Return to <a href="login.htm">Login Page</a></font>

<% elseif rsTitleList("password")=password then %>

<h1>UMP Administrator Access</h1>
<HR><UL>
<LI><H3><a href="exec2.asp">Alumni List</A></H3></LI>
<LI><H3><a href="addnew.asp">News & Jobs Management</H3></LI>
</UL>

<% else %>
<p><font size=3>Sorry, invalid username or password. Please try again</
font></p>
<font size=3>Return to <a href="login.htm">Login Page</a></font>

<% end if %>
</body>
</html>
```

```
<html>
<head>
  <title>UMP Alumni Member List</title>
</head>
<body>

<h2>Universiti Malaysia Pahang Alumni List</h2>
<P>

<%
dim connectString
dim connectionToDatabase
dim recordCollection

connectString="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.
  MapPath( "db\USERMAN.MDB" )
Set connectionToDatabase=Server.CreateObject("ADODB.Connection")
connectionToDatabase.OPEN connectString

set recordCollection = Server.CreateObject("ADODB.Recordset")

Set recordCollection=connectionToDatabase.Execute("SELECT * FROM users")

Response.Write("<TABLE BORDER=1><TR><TH>NO</TH><TH>NAME</TH><TH>MATRIX ID
  </TH><TH>PASSWORD</TH><TH>ADDRESS 1</TH><TH>ADDRESS 2</TH><TH>GRAD
  YEAR</TH><TH>PHONE</TH><TH>OCCUPATION</TH></TR>")

Do While NOT recordCollection.EOF
  Response.Write("<TR>")
  For x=0 to recordCollection.Fields.Count-1
    Response.Write("<TD>")
    Response.Write(recordCollection(x))
    Response.Write("</TD>")
  Next

  Response.Write("</TR>")
  recordCollection.MoveNext

Loop
Response.Write("</TABLE>")

connectionToDatabase.Close
Set connectionToDatabase=Nothing
%>

<br>
<br>
Delete a record by Matrix ID (Case Sensitive):
<form method="POST" action="delete.asp">
  <p><input type="text" name="delete" size="5"><input type="submit" value=
    "Delete" name="button2"><br>
<br />
<a href="login.htm">Logout</a>
</p>
</form>

</body>
</html>
```

```
<%  
  
name = server.HtmlEncode(request.form("delete"))  
  
Set Conn = Server.CreateObject("ADODB.Connection")  
  
Conn.Provider = "Microsoft.Jet.OLEDB.4.0"  
Conn.ConnectionString = "Data Source=" & Server.MapPath ("db\userman.mdb")  
Conn.Open  
  
Set Rs = Server.CreateObject("ADODB.Recordset")  
Rs.Open "SELECT * from users", Conn, 1,3  
  
do while not rs.EOF  
if Rs("MatrixID") = name then Rs.Delete ' if it's = name then delete  
rs.MoveNext ' Movenext  
loop  
  
set Rs = nothing  
set Conn = nothing  
  
Response.Redirect("exec2.asp")  
%>
```

```
<html>
<head>
  <title>News & Jobs Management Menu</title>
</head>
<body>
<span style="font-size: 14pt">News & Job records</span></p>

<%
dim connectString
dim connectionToDatabase
dim recordCollection
connectString="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.
  MapPath("db\USERMAN.MDB")
Set connectionToDatabase=Server.CreateObject("ADODB.Connection")
connectionToDatabase.OPEN connectString

set recordCollection = Server.CreateObject("ADODB.Recordset")
Set recordCollection=connectionToDatabase.Execute("SELECT * FROM news")
Response.Write("<TABLE BORDER=1><TR><TH>NO</TH><TH>NEWS</TH><TH>JOBS</TH>
  </TR>")

Do While NOT recordCollection.EOF
  Response.Write("<TR>")
  For x=0 to recordCollection.Fields.Count-1
    Response.Write("<TD>")
    Response.Write(recordCollection(x))
    Response.Write("</TD>")
  Next
  Response.Write("</TR>")
  recordCollection.MoveNext

Loop
Response.Write("</TABLE>")
connectionToDatabase.Close
Set connectionToDatabase=Nothing
%>

<br>
<br>
Enter number to delete (Case Sensitive):
<form method="POST" action="deletenews2.asp">
  <p><input type="text" name="delete" size="5"><input type="submit" value=
    "Delete" name="button2"><br>
<br />
</p>
</form>
<br>
<br>
<Form name = "request" action ="addnewstry6.asp" method ="POST">
<p>
  <span style="font-size: 14pt">UMP News Entering Menu</span></p>
  <p>Please enter related news and job:</p>
  <p><strong></strong>&nbsp;</p>
  <table border ="0">
    <tr><td align = "center"><i>News</i></td>
      <td><input type ="text" size = 200 name = "varNews"></td>
    </tr>
    <tr><td align = "center"><i>Job</i></td>
      <td><input type ="text" size = 200 name = "varJob"></td>
    </tr>
  </table>
  <p><input type ="submit" value ="Submit" />
  </p>
</Form>
<a href="login.htm">Logout</a>
</body>
</html>
```



```
<html>
<head>
  <title>UMP Alumni registration page</title>
</head>
<body>

<h2>Record updated..</h2>
<P>

<%
berita = Request.Form("varNews")
kerja = Request.Form("varJob")

Dim conn, rs, sql
Set conn = Server.CreateObject("ADODB.Connection")
Set rs = Server.CreateObject("ADODB.Recordset")
'---Opens the connection to the database---
conn.open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" & Server.
MapPath("db/userman.mdb")

sql = "INSERT INTO news (berital,berita2) VALUES ('" & berita & "','" &
& kerja & "'"")
conn.Execute(sql)
conn.close
set conn = Nothing
Response.Redirect("addnew.asp")
%>

</body>
</html>
```

```
<html>
<head>
  <title>UMP Alumni registration page</title>
</head>
<body>

<h2>Record updated..</h2>
<P>

<%
  padam = Request.Form("delete")

  Dim conn, rs, sql
  Set conn = Server.CreateObject("ADODB.Connection")
  Set rs = Server.CreateObject("ADODB.Recordset")
  '---Opens the connection to the database---
  conn.open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" & Server.
  MapPath("db/userman.mdb")

  sql = "DELETE newsno FROM news WHERE newsno= " & padam & " "
  conn.Execute(sql)
  conn.close
  set conn = Nothing
  Response.Redirect("addnew.asp")
%>

</body>
</html>
```

APPENDIX C

PROJECT CODES FOR CLIENT-SIDE DEVELOPMENT

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
<card id="login" title="Registration Successful">

<p align="center">

Thank you for entering your details.<br><br>
<a href="home.wml#menu">Main Menu</a>

</p>

<%

DIM CONNECT,DBCONN,SQL
CONNECT="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.MapPath(
    "db\USERMAN.MDB")
Set DBCONN=Server.CreateObject("ADODB.Connection")
DBCONN.OPEN CONNECT

DIM rsUserAdd
    Set rsUserAdd = Server.CreateObject("ADODB.Recordset")

DIM FNAME,ADDRESS,TAHUN,PHONE,MATRIX,OCCUPATION
rsUserAdd.open "users", dbconn, 2, 3

FNAME      = REQUEST.QUERYSTRING("F_Name")
MATRIX     = REQUEST.QUERYSTRING("Matrix")
PASSWORD   = REQUEST.QUERYSTRING("Passw")
ADDRESSH   = REQUEST.QUERYSTRING("AlamatH")
ADDRESSV   = REQUEST.QUERYSTRING("AlamatV")
TAHUN      = REQUEST.QUERYSTRING("Grad")
PHONE      = REQUEST.QUERYSTRING("Telefon")
OCCUPATION = REQUEST.QUERYSTRING("Occ")

    rsUserAdd.AddNew
    rsUserAdd("FNAME")      = FNAME
    rsUserAdd("MATRIXID")   = MATRIX
    rsUserAdd("PASSWORD")   = PASSWORD
    rsUserAdd("ADDRESS1")   = ADDRESSH
    rsUserAdd("ADDRESS2")   = ADDRESSV
    rsUserAdd("YEAR")       = TAHUN
    rsUserAdd("PHONE")      = PHONE
    rsUserAdd("OCCUP")      = OCCUPATION

    rsuserAdd.Update
    rsUserAdd.CLOSE
    SET rsUserAdd = NOTHING
%>

</p>
</card>
</wml>
```

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.
    org/DTD/wml_1.1.xml">

<wml>

    <card id="News" title="NEWS & JOBS">
        <do type="prev" label="Back">
            <prev/>
        </do>

<%
Set Conn = Server.CreateObject("ADODB.Connection")

Conn.Provider = "Microsoft.Jet.OLEDB.4.0"
Conn.ConnectionString = "Data Source=" & Server.MapPath ("db\userman.mdb")
Conn.Open

Set Rs = Server.CreateObject("ADODB.Recordset")
Rs.Open "SELECT * from news", Conn, 1,3

do while not rs.EOF
Response.Write ("NEWS:")
Response.Write ("<BR>")
Response.Write (rs("berital"))
Response.Write ("<BR>")
Response.Write ("<BR>")
Response.Write ("JOBS:")
Response.Write ("<BR>")
Response.Write (rs("berita2"))
Response.Write ("<BR>")
Response.Write ("<BR>")

rs.MoveNext ' Movenext
loop

set Rs = nothing
set Conn = nothing
%>

<br><br>
<a href="home.wml#menu">Main Menu</a>
</card>
</wml>
```

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<!-- user welcome and login -->
<wml>
  <head>
    <meta http-equiv="Cache_Control" content="max-age=0"/>
  </head>
  <card id="Update" title="Update">
    <do type="accept" label="Login">
      <go href="updatelogin2.asp?User=$(User)&Pwr=$(Pwr)"/>
    </do>
    <p align="center">
      <b>Please enter corresponding account to update:</b>
    </p>
    <p>
      <br/>Matrix ID:
      <input name="User" type="text" format="AANNNNN" emptyok="false"/><br
  </p>
    <p>
      <br/>Password:
      <input name="Pwr" maxlength="6" type="password" emptyok="false"/>
    </p>
  </card>
</wml>
```



```
Matrix ID? (7)

<br/>
<do type="accept" label="Confirm">
  <go href="update_add1.asp" method="GET">
    <postfield name="Uadd1" value="$(varAdd)" />
    <postfield name="MatID" value="$(varMatrix)" />
  </go>
</do>
<br/>
</p>
</card>

<card id="uadd2" title="Update Address 2">
  <p align="center">
    New Address (Viewable) ? (max 50)
    <input name = "varAdd2"/>
    Matrix ID? (7)
    <input name = "varMatrix" type="text" format="AANNNNNN"/>
    <br/>
    <do type="accept" label="Confirm">
      <go href="update_add2.asp" method="GET">
        <postfield name="Add2" value="$(varAdd2)" />
        <postfield name="MatID" value="$(varMatrix)" />
      </go>
    </do>
    <br/>
  </p>
</card>

<card id="uphone" title="Update Phone Number">
  <p align="center">
    New Phone Number? (max 11)
    <input name = "varPhonen" type="text" format="NNNNNNNNNNNN"/>
    Matrix ID? (7)
    <input name = "varMatrix" type="text" format="AANNNNNN"/>
    <br/>
    <do type="accept" label="Confirm">
      <go href="update_phone.asp" method="GET">
        <postfield name="Phonen" value="$(varPhonen)" />
        <postfield name="MatID" value="$(varMatrix)" />
      </go>
    </do>
    <br/>
  </p>
</card>

<card id="uocc" title="Update Occupation">
  <p align="center">
    New Occupation? (max 15)
    <input name = "varOcc"/>
    Matrix ID? (7)
    <input name = "varMatrix" type="text" format="AANNNNNN"/>
    <br/>
    <do type="accept" label="Confirm">
      <go href="update_occ.asp" method="GET">
        <postfield name="Occu" value="$(varOcc)" />
        <postfield name="MatID" value="$(varMatrix)" />
      </go>
    </do>
    <br/>
  </p>
</card>
</wml>
<%
rs.Close
set rs = nothing
```



```
<% Response.ContentType = "text/vnd.wap.wml"
Response.Expires = -1
%>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.
org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="Update Address 1">
    <p>
      <%
        Dim conn, rs
        Set conn = Server.CreateObject("ADODB.Connection")
        Set rs = Server.CreateObject("ADODB.Recordset")
        '---Opens the connection to the database---
        conn.open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" & Server.
MapPath("db/userman.mdb") & ";"
        '---Updates the qty
        sql = "UPDATE Users SET Address1='" & Request.QueryString("Uadd1") &
"'" & "WHERE MatrixID='" & Request.QueryString("MatID") & "'"
        conn.Execute(sql)
        Response.Write "Record succesfully updated!"
      %>
    </p><br>
    <a href="home.wml#menu">Main Menu</a>
  </card>
</wml>
```

```
<% Response.ContentType = "text/vnd.wap.wml"
Response.Expires = -1
%>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.
org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="Update Address 2">
    <p>
      <%
        Dim conn, rs
        Set conn = Server.CreateObject("ADODB.Connection")
        Set rs = Server.CreateObject("ADODB.Recordset")
        '---Opens the connection to the database---
        conn.open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" & Server.
MapPath("db/userman.mdb") & ";"
        '---Updates the qty
        sql = "UPDATE Users SET Address2='" & Request.QueryString("Add2") &
'" & " WHERE MatrixID='" & Request.QueryString("MatID") & "'"
        conn.Execute(sql)
        Response.Write "Record succesfully updated!"
      %>
    </p><br>
    <a href="home.wml#menu">Main Menu</a>
  </card>
</wml>
```

```
<% Response.ContentType = "text/vnd.wap.wml"
Response.Expires = -1
%>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.
org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="Update Occupation">
    <p>
      <%
        Dim conn, rs
        Set conn = Server.CreateObject("ADODB.Connection")
        Set rs = Server.CreateObject("ADODB.Recordset")
        '---Opens the connection to the database---
        conn.open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" & Server.
MapPath("db/userman.mdb") & ";"
        '---Updates the qty
        sql = "UPDATE Users SET Occup='" & Request.QueryString("Occu") & "'"
        & " WHERE MatrixID='" & Request.QueryString("MatID") & "'"
        conn.Execute(sql)
        Response.Write "Record succesfully updated!"
      %>
    </p><br>
    <a href="home.wml#menu">Main Menu</a>
  </card>
</wml>
```

```
<% Response.ContentType = "text/vnd.wap.wml"
Response.Expires = -1
%>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.
org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="Update Phone No.">
    <p>
      <%
        Dim conn, rs
        Set conn = Server.CreateObject("ADODB.Connection")
        Set rs = Server.CreateObject("ADODB.Recordset")
        '---Opens the connection to the database---
        conn.open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" & Server.
MapPath("db/userman.mdb") & ";"
        '---Updates the qty
        sql = "UPDATE Users SET Phone=" & Request.QueryString("Phonen") & "
WHERE MatrixID='" & Request.QueryString("MatID") & "'"
        conn.Execute(sql)
        Response.Write "Record succesfully updated!"
      %>
    </p><br>
    <a href="home.wml#menu">Main Menu</a>
  </card>
</wml>
```

```
<%@ Language = "JScript"; %>
<% Response.ContentType = "text/vnd.wap.wml";
%><?xml version="1.0"?>
<!-- #include file="adojavas.inc" -->
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.
wapforum.org/DTD/wml_1.1.xml">

<wml>
<card id="search" title="Search Results">
<p>
<strong>Member Search Results:</strong><br><br>
<%
// dsn-less connection to userman.mdb

conn = Server.CreateObject("ADODB.Connection");
rs = Server.CreateObject("ADODB.Recordset");

conn.Open("DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" +
Server.MapPath("db\userman.mdb") + ";");

// create a string to hold the SQL query
sqlQuery = "SELECT * FROM users where FNAME like '%" + Request.
QueryString("SNAME") + "%' ORDER BY FNAME";

// execute the query on the database
rs.Open(sqlQuery, conn, adOpenForwardOnly, adLockReadOnly,
adCmdText);

while (!rs.EOF)
{
Response.Write("<strong>Name: </strong><br>");
Response.Write(rs("FNAME") + "<br/>");
Response.Write("<strong>Phone No: </strong><br>");
Response.Write(rs("PHONE") + "<br/>");
Response.Write("<strong>Address: </strong><br>");
Response.Write(rs("ADDRESS2") + "<br/>");
Response.Write("<strong>Graduate: </strong><br>");
Response.Write(rs("YEAR") + "<br/>");
Response.Write("-----");
Response.Write("<br/>");
rs.MoveNext();
}
conn.Close();
%>
<do type="accept" label="Search Again">
<prev/>
</do>

</p>
</card>
</wml>
```