

Final Year Project

WXET 3182

WAP MESSAGING APPLICATION

BY

YEW HOOI PHAIK

WET 98066



Under the supervision of

Mr. Ling Teck Chaw

Session 2000/2001

*Faculty Of Computer Science and Information Technology
University Malaya
2000/2001*

ABSTRACT

Statistics have shown that more and more people are switching to WAP (Wireless Application Protocol) handphones and most people would have heard of WAP before. Undeniably WAP's future is bright and WAP applications prospects are even better. This alone has encouraged many programmers into developing WAP applications. Study had shown that this is true in part of the world with more and more hooking up to WAP handphones.

Internet Messaging programs have existed for a long time. Currently there's no WAP Messaging Application (WAP-MA) but it's just a matter of time before somebody invented it. This project is involved in developing a WAP Messaging Application. It would be a simplified messaging program compared to other more complicated internet messaging programs because WAP's limitations also constrain WAP-MA to have many functions.

The system is developed in Windows NT platform integrating Apache Web Server, Oracle 8i and a SMTP Mail Server. Java Server Pages (JSP) and Wireless Markup Language (WML) are the two main programming languages used as the development technology. The development software used in testing the system is Nokia WAP Toolkit which is an emulator by Nokia to enable WAP programs to be tested out on computer.

ACKNOWLEDGEMENT

First of all, I would like to greet all my readers a warm welcome. There are a few person I would personally like to take this opportunity to thank them.

I would like to express my sincerest gratitude and utmost appreciation to my supervisor Mr. Ling Teck Chaw for his invaluable advices and guidance to me throughout the whole thesis development.

My moderator Dr Mazliza Othman a driving force and a helping hand in voicing out her ideas in improving my system through various ways.

My family, friends and course-mates for their support and assistance when I needed them most which contributes to the success and completion of this project.

Last but not least is a special thanks to my two project partners Miss Goh Bee Yen and Miss Goh Chiau Foon, who have been great friends to me. They have been very co-operative during the whole project development and supportive in facing the ups-and -downs together with me.

TABLE OF CONTENTS

CHAPTER 1 : INTRODUCTION.....	i
1.1 : PROJECT INTRODUCTION	2
1.2 : PROJECT OBJECTIVES	3
1.3 : PROJECT SCOPE	5
1.3.1 : The User.....	6
1.3.2 : The Administrator.....	6
1.4 : PROJECT SCHEDULE.....	6
1.5 : REPORT ORGANISATION.....	8
 CHAPTER 2 : LITERATURE REVIEW	 10
2.1 : INFORMATION SOURCE & FINDINGS	10
2.1.1 : INTERNET	10
2.1.2 : ARTICLES FROM MEDIA.....	11
2.1.3 : NEWSGROUP.....	12
2.1.4 : GROUP DISCUSSION	13
2.1.5 : OTHERS.....	14
2.2 : COMPARISON WITH EXISTING SYSTEM	14
2.2.1 : CURRENT WAP APPLICATIONS	15
2.2.2 : PROBLEMS	16
2.2.3 : SOLUTION	16
2.3 : TOOLS CONSIDERED	17
2.3.1 : SOFTWARE ARCHITECTURE	17
2.3.2 : DEVELOPMENT PLATFORM.....	18
2.3.3 : DEVELOPMENT SERVER.....	21
2.3.4 : DEVELOPMENT SOFTWARE	25
2.3.5 : DEVELOPMENT TECHNOLOGY.....	29
 CHAPTER 3 : SYSTEM REQUIREMENT.....	 34
3.1 : PROJECT DEFINITION	34
3.1.1 : FUNCTIONAL REQUIREMENT	34
3.1.2 : NON-FUNCTIONAL REQUIREMENT	39
3.2 : APPROACH.....	41
3.3 : TOOLS CHOSEN.....	42
 CHAPTER 4 : SYSTEM DESIGN.....	 44

4.1	: PROJECT ARCHITECTURE	44
4.2	: APPLICATION DESIGN.....	45
4.3	: NETWORK DESIGN	50
4.4	: DATABASE DESIGN	51
4.4.1	: ENTITY-RELATIONSHIP MODEL	54
4.5	: GRAPHICAL USER INTERFACE DESIGN.....	56
4.6	: USER AND HUMAN FACTORS	61
4.7	: CONCLUSION.....	62
CHAPTER 5 : SYSTEM IMPLEMENTATION.....		63
5.1	: DEVELOPMENT TOOLS IMPLEMENTATION	63
5.2	: MODULE IMPLEMENTATION.....	64
5.2.1	: Admin Module Implementation.....	65
5.2.2	: User Module Implementation	77
CHAPTER 6 : SYSTEM TESTING		87
6.1	: UNIT TESTING	87
6.1.1	: Administrator Module Testing.....	87
6.1.2	: User Module Testing	89
6.2	: MODULE TESTING.....	90
6.2.1	: Administrator Module Testing.....	90
6.2.2	: User Module Testing	99
6.3	: INTEGRATION TESTING.....	102
6.3.1	: AddFriend Module & ApproveRequest Module	102
6.4	: SYSTEM TESTING	104
6.4.1	: Administrator Module Testing.....	104
6.4.2	: User Module Testing	107
CHAPTER 7 : CONCLUSION.....		108
7.1	: EVALUATION	108
7.2	: STRENGTH.....	108
7.2.1	: Platform Independent.....	108
7.2.2	: Wide Accessibility	109
7.2.3	: Interoperability.....	109
7.2.4	: Coding Reusability And Compatibility	109
7.2.5	: Confidentiality	110
7.2.6	: Expansion And Scalability.....	110

7.3	: LIMITATION	110
7.3.1	: Speed	110
7.3.2	: Insufficient WAP Systems	111
7.3.3	: Browser Cache Limitation	111
7.4	: FUTURE ENHANCEMENTS	112
7.4.1	: Security	112
7.4.2	: Database Enhancement	112
7.4.3	: Auto Detect Location	113
7.4.4	: WAP Gateway	113
7.5	: KNOWLEDGE GAINED	113
7.5.1	: Network Systems	113
7.5.2	: Setup Servers	114
7.5.3	: Java Technology	114
7.5.4	: Oracle And SQL	115
7.6	: CONCLUSION	115

REFERENCES.....	117
------------------------	------------

LIST OF FIGURES

Figure 1.1 : Project Schedule	7
Figure 4.1: WAP Overview	44
Figure 4.2: WAP-MA Context Level Diagram.....	46
Figure 4.3 : WAP-MA User Flow.....	47
Figure 4.4 : WAP-MA Super Admin Flow.....	48
Figure 4.5 : WAP-MA Normal Admin Flow.....	49
Figure 4.6 : WAP-MA Network Design	50
Figure 4.7 : Super Admin Privilege Entity Relationship Diagram	55
Figure 4.8 : Normal Admin Privilege Entity Relationship Diagram	55
Figure 4.9 : User Privilege Entity Relationship Diagram	55
Figure 5.1 : Form Links for Super Administrator Module	65
Figure 5.2 : Form Links for Normal Administrator Module	66
Figure 5.3 : Form Links for User Module.....	78
Figure 6.1 : Login box for Login Function	88
Figure 6.2 : Delete table for Delete User Function	88
Figure 6.3 : Screen for User Query Admin Function.....	89
Figure 6.4 : Error Screen because of maximum characters	90
Figure 6.5 : Login Page.....	91
Figure 6.6 : Super Administrator Main Menu Page.....	94
Figure 6.7 : Normal Administrator Main Menu Page.....	95
Figure 6.8 : Delete User Page	96
Figure 6.9 : Delete User Mail	97
Figure 6.10 : Delete User Fail Message.....	97
Figure 6.11 : Accessing SQLPlus to verify the deletion.....	98
Figure 6.12 : User 'bruce' trying to log into WAP-MA	99
Figure 6.13 : Login Fail	99
Figure 6.14 : Friends Menu.....	100
Figure 6.15 : Write Message	100
Figure 6.16 : Choose Recipients	101
Figure 6.17 : Message Status	101
Figure 6.18 : Adding Friend.....	102
Figure 6.19 : Adding Friend Status.....	103
Figure 6.20 : Approve Request	103
Figure 6.21 : Approve Request Status	103
Figure 6.22 : Adding the Requestor in return	104
Figure 6.23 : MainMenu of Administrator Module	105

Figure 6.24 : Add User Module	106
Figure 6.25 : Main Menu	107
Figure 6.26 : Friends Menu.....	107

LIST OF TABLES

Table 4.1 : Field in the 'UserDetails' table.....	51
Table 4.2 : Field in the 'UserApproval' table	52
Table 4.3 : Field in the 'UserFriends' table	53
Table 4.4 : Field in the 'UserIncoming' table.....	53
Table 4.5 : Field in the 'AdminDetails' table	54
Table 5.1 : Development Tools table.....	64

LIST OF SCREENS

Screen 4.1 : WAP-MA First Page.....	57
Screen 4.2 : Login Menu.....	57

LIST OF FILES

File 5.1 : 'ConnectDB.jsp' file.....	67
File 5.2 : 'CloseDB.jsp' file.....	67
File 5.3 : Part of the file 'Login.jsp'	68
File 5.4 : Part of the file 'Admin_LoginVerify.jsp'	69
File 5.5 : Part of the file 'User_Add.jsp'	72
File 5.6 : Send Mail Function	73
File 5.7 : Part of the file 'User_Delete.jsp'	75
File 5.8 : 'User_Link1.jsp' file	81
File 5.9 : Part of the file 'User_Reply.jsp'	83

CHAPTER 1 : INTRODUCTION

Communication started off in as early as the dawn of man. The proverb “No Man Is An Island” describes very clearly that every human on earth needs to communicate with one another. The early success of communication started off with the invention of telephone by a Scottish, Alexander Graham Bell in 1876.

Then came the mobile phone where one doesn't have to stay static when communicating through the radio-telephone especially catered for people who is always on the go in the 1940s. Not so long ago back in 1996, NTT Docomo came out with WAP technology, an abbreviation for Wireless Application Protocol to be used for mobile internet. Since then, mobile internet is in it's meteoric rise as the most potential technology with more than 250 million WAP subscribers in the world by 2003 according to Nokia. Although this is a very conservative figure but Nokia and many telephone vendors are aggressively rolling out their WAP phones.

Mobile internet is widely accepted globally because it is highly contagious infecting from country to country. Moreover, it have conquered over 90% of the global handset market, carriers with more than 100 million subscribers therefore leading infrastructure providers, software developers and other organizations have teamed up to form WAP Forum in order to provide solutions to the wireless industry. WAP Forum's primary goal is to bring together companies from all segments of the wireless industry value chain to ensure product interoperability and growth of wireless market. The WAP Forum is indeed the industry association comprising over 200 members that has developed the de-facto world standard for wireless information and telephony services on digital mobile phones and other wireless terminals.

WAP is an open, global specification that empowers mobile users with wireless devices to easily access and interact with information and services instantly. The main reason for implementing WAP is to enable easy fast delivery of relevant information and services to mobile users.

The next possible stop of WAP reaches Malaysia by early 2000. WAP was well received by Malaysians and still is. Currently there are 3 out of 6 mobile service providers namely Maxis, Celcom and Adams have started to provide WAP services to it's subscribers. As for handphones vendors namely Nokia, Siemens, Ericsson and Motorola Malaysia, they have welcomed WAP with their variety of WAP handphones models to be offered to WAP subscribers to choose from.

1.1 : PROJECT INTRODUCTION

Introduction of WAP has provided internet surfers another alternative to surf internet. They not only have to resolve only to their computers when it comes to accessing the internet but they can also turn to handphones as another means of internet surfing.

The recent introduction of WAP in Malaysia have received warm welcome from businessman who needs information while they are on the go to youngsters who can afford them for fashion sake because it is considered the 'in-thing'.

Not so long ago, Short Message Services better known as SMS is introduced to every mobile subscribers where they can message and receive message from their handphones from friends also registered to the same service providers as they are for example SMS from Celcom subscriber to Celcom subscriber.

WAP Messaging application (WAP-MA) is the next possible application to be developed because WAP is all about information and communication. WAP-MA is not just a means of communication but it is an effective and efficient way of communication. The main reason of WAP-MA is to entitled WAP subscribers with the same benefits of using the same communication application that they can get from internet such as I-Seek-You (ICQ) application, Yahoo Messenger and MSN Messenger.

The WAP-MA is a cost effective communication to WAP subscribers because they just need to login into the WAP-MA website to register and then they can use the services for free. WAP-MA also takes into account security against privacy intrusion when being developed, hence making it a more reliable application.

1.2 : PROJECT OBJECTIVES

The fundamental objective is to enable communication between 2 WAP handphone users through a application being hosted in a web server. This is not just any means of communication but it also meets the objectives stated :

- 1) WAP-MA is a messaging application that provides a fast way to communicate. It is categorized as fast communication because user is expected only to write short messages not more than 100 characters in each message to a friend. This application is especially beneficial when the user have to send messages to multiple friends in a short time span.

- 2) It is also an effective way to communicate because this application allows authentication from the administrator to reject unwanted users. Other than that, a user will be able to add another registered user into the 'friend's list' but he needs authorization from the other party before he can start to message and receive message from that particular user. Therefore, this will ensure that those inside the user's 'friends list' are really friends of the user. Therefore, unwanted messages will not be received and the user would have to waste time going through these messages before he goes to the messages that he would really want to receive from.
- 3) WAP-MA is accessible through a handphone at all time too. Therefore, it is best for people who is always on the go and they won't be bounded by location when they want to message friends.

Mobile access fees to WAP is as follows :

Peak rate (9am - 7pm)	RM 0.30/minute
Off peak rate (7pm - 12 midnight)	RM 0.15/minute
Super off-peak (12 midnight - 9am)	RM 0.025/minute

Therefore, the connection fees are more expensive than the normal dial-up internet connection which costs only RM0.025 per minute all day. Therefore unless one access WAP at super off-peak hours, otherwise the costs will still be high. WAP-MA will help cost cutting because one message can be delivered to multiple recipients and it saves time too because the same message would not have to be re-keyed in.

- 4) The application also has some security features. This is to avoid unauthorized intrusion or imitation to be the user, that is why password

would not be exposed unnecessarily. Therefore, password would not be leaked inside the e-mail unless requested by the user in situation where he had forgotten the password

1.3 : PROJECT SCOPE

The scope of the project means the coverage of the project as in what is the limitations of the application and what are the boundaries set upon the application.

- 1) The limitation of the applications is user can only accessed to the applications through WAP handphone. As for the administrator, he will access through internet browser such as Internet Explorer or Netscape Navigator.
- 2) Since this application did not set a quota for maximum users, it is unlimited to users as long as they are registered with the system.
- 3) For security reasons, an administrator has the right to approve a new user's registration or reject it because of unforeseen problems.
- 4) To maintain the application to run smoothly, all users must be responsible and not use the application in the wrong ways. Therefore, the administrator has the rights to delete users who is did not oblige to the application's rules.



Below are stated the scopes; boundaries of what can be done and what not to be done in 2 viewpoints namely the administrator (both super and normal administrator) and the user.

1.3.1 : The User

- To register with the system and wait for approval
- To configure his/her own personal details
- To maintain friend's list by adding and deleting users
- To be added and approved request when added in as friend by another user
- To message and receive message from friends
- To query and contact administrators

1.3.2 : The Administrator

- To approve or reject new registration
- To send approval or rejection notice via e-mail to the newly successfully registered user or rejected new user.
- To add a new administrator or new user
- To delete a new administrator or existing user
- To maintain data integrity for each registered user.
- To notify by messaging the user whom has been deleted by his own friends.
- To contact users and answer users' queries.

1.4 : PROJECT SCHEDULE

The project has started off around June 2000 and will end around January 2001. Basically, there are 5 main stages in WAP-MA development life cycle. The Gantt Chart in Figure 1.1 shows the project schedule.

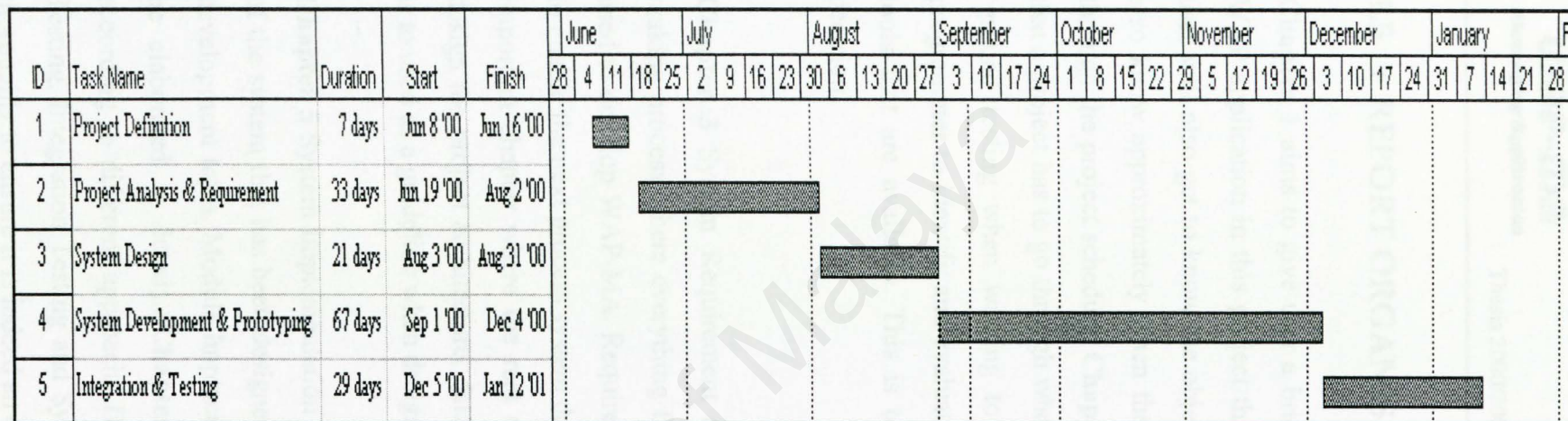


Figure 1.1 : Project Schedule

1.5 : REPORT ORGANISATION

Chapter 1 aims to give user a brief and overview picture of the WAP and the WAP application in this project that will be developed. Other than introducing, user will also get to know the objectives and the scope of the project. User will also know approximately when the project started and is going to be complete through the project schedule. **Chapter 2** Literature review is the first few process that a project has to go through when wanting to develop a system. It is similar to 'ground testing when wanting to build a house'. It is indeed vital for the programmer to identify and evaluate what information on hand and development tools that are available. This is to avoid mistakes because of making wrong decision.

Chapter 3 System Requirement aims to show system analysis and decision making process where everything from the development tools used till approach used to develop WAP-MA. Requirements of the system have also been identified to enable the next process system design to begin. **Chapter 4** System design is an important chapter where the real design of the system is taking place. It will design the project architecture, database design, graphical user interface and this is to used as a guideline when designing and developing the real system.

Chapter 5 System Implementation will define the whole implementation process of the system that has been designed earlier on in Chapter 4. It defines briefly on development tools. Module Implementation picks on a few important modules to be elaborated in details. **Chapter 6** System Testing tested out the system according to different approach. The approaches such as Unit Testing, Module Testing, Integration Testing and System Testing are all aimed to test the system thoroughly to ensure it is indeed an efficient, reliable and effective system.

Chapter 7 Conclusion focuses on the findings and evaluations of the system in meeting its objectives as outlined in Chapter 1. The strengths, weaknesses as in limitations and future enhancements are the main points that the chapter is trying to outline. This chapter is therefore important for future programmer to refer on when deciding to enhance the system.

2.1 : INFORMATION SOURCE & FINDINGS

Many approaches to find information on WAP had been utilized. Since WAP is still very new to many, therefore the available information on WAP is not much. My information findings were as general as possible. I divided into five general internet articles from media, newspaper, group discussion and others.

2.1.1 : INTERNET

Internet has played a big role in speeding this information finding process. Because there is a variety of information from history of WAP until the most up-to-date WAP development.

From the internet, websites of mobile service providers around the globe, WAPforum, book vendors for WML and WMLScript (the books are not published yet but review are available in the website) are available. Other than that, WAP developing programs which are provided by mobile service providers such as Phone Simulator, WML and WMLScript are available for downloading free. Programs such as WAP Server emulator could be downloaded or a

CHAPTER 2 : LITERATURE REVIEW

In this literature review, it will be divided into three parts namely the approaches, findings and analysis. Firstly, literature review will identify all the approaches that were used to find information for the project, and then provide sources and summary of the findings, and finally to analyse the strengths and weaknesses of all the information that were found.

2.1 : INFORMATION SOURCE & FINDINGS

Many approaches to find information on WAP had been utilized. Since WAP is still very new to many, therefore the available information on WAP is not much. My information findings sources are generally divided into five namely internet, articles from media, newsgroup, group discussions and others.

2.1.1 : INTERNET

Internet has played a major role in succeeding this information finding process because there is a vast variety of information from history of WAP until the most up-to-date WAP development.

From the internet, websites of mobile service providers around the globe, WAPForum, book vendors for WML and WMLScript (the books are not published yet but review are available in the website) are available. Other than that, WAP developing programs which are provided by mobile service providers such as Phone Simulator, WML and WMLScript help are available for downloading free. Programs such as WAP Server however might be freeware or a

shareware depending on the vendors. Basically, internet has played an important role in assisting a lot in the early development of this project.

Findings

From WebClimbing.com, one can access all sort of information with a WAP phone: traffic, updated news, entertainment events, shopping, weather, stocks, send and check e-mails too and it is pretty much everything that can be accessed with a normal computer and Internet connection. Sites like Amazon, Yahoo and Sportal are already using WAP to enhance user's choices.

Based on the findings from the internet from the website gsmworld.com, there are at least four WAP toolkits available for software developers to use to assist in the speedy development of WAP-based services. These are supplied by Dynamical System Research (DSR), Ericsson, Nokia, Phone.com and Motorola. As far as concerned, these WAP Development Toolkits can be downloaded for free but the supporting tools such as WAP server would have to be paid.

2.1.2 : ARTICLES FROM MEDIA

Media such as newspapers and magazines have been a great help in finding information about WAP. Newspapers that were usually frequent include in-Tech from Star and Computetimes from News Straits Times. Both of these newspapers have at least 1 WAP article in each issue and how WAP is progressing whether it is in Malaysia or overseas.

Magazines such as PC World, PC.com, netv@lue which is a pullout from The Edge, Asia Computer Weekly, also discusses WAP development and it's usage globally.

Findings

Quoted from ComputerWorld June 2000 edition, WAP is undoubtedly one of the top technologies of the year. WAP 1.1 has achieved considerable support amount wireless handset vendors, wireless infrastructure vendors and wireless application software providers, most notably through widespread implementation of WAP's XML-based Wireless Markup Language (WML). WAP phones will dominate the market because according by many researches by 2002, more people will access the Internet by phone than by computer.

Quoted from netv@lue July 10, 2000 edition, according to Celcom "WAP fees has to be extremely cheap, very fast, strong local content and it has to be personalized to suit each customer's needs in order to gain a people trust to switch to WAP and Celcom to be specific." According to netv@lue, competition in the mobile internet is so intense that even before most Malaysians can rush out to buy their WAP phones and enjoy the promise of the wireless internet, their telcos are already promising General Packet Radio Services (GPRS). This proves how fast a rate mobile internet is going through.

2.1.3 : NEWSGROUP

WAP Newsgroup such as from the website <http://news.wapunderground.com:81/> is very useful especially when wanting to receive information shared out by other WAP developers regarding their findings and problems faced. The real notion of WAP Newsgroup is to basically understand what WAP developers around the

globe is doing and the information has be very useful in developing WAP-MA. Other than that, Newsletter from <http://www.wap.com> had been helpful too in keeping track the on-goings in the WAP world.

Findings

Based on the newsgroup, information about which WAP gateways are useful and which WAP Toolkit had been useful and most widely used. Currently, Nokia WAP Toolkit 2.0 is the most widely used by developers around the globe because of it's compatibility with a lot of development platforms and it's easy to navigate because of its' user-friendly interface.

2.1.4 : GROUP DISCUSSION

Group discussion with 2 other friends who are both developing WAP applications also had been helpful indeed when it on-the-spot help is needed rather than waiting a few days for the newsgroup to respond. Sharing information of WAP and useful websites found are among a few situation where a group discussion is needed important.

Findings

Many useful websites and useful source codes examples form various websites had been found as a result from the discussion. Apart from that, conclusion of developments tools that will be used are one of few success gained from group discussion.

2.1.5 : OTHERS

Although WAP is relatively new in Malaysia, but there are already 3 service providers Maxis, Celcom and Adams launching their WAP services in the middle of the year. Maxis especially take an extra step in informing the public with more information on WAP in an exhibition area called i-Center in Kuala Lumpur City Center (KLCC) building.

In there, one can find information on WAP and what services are currently available to Maxis WAP's subscribers, fees charged and also which models of handphones that can be used to access WAP.

Findings

Maxis which is pioneer in WAP technology had indeed made an extra effort in equipping it's center with WAP kits for customer convenience and other mobile providers are eager in promoting their WAP services too. Therefore, WAP's future in Malaysia is indeed bright although it started off late comparatively to many countries.

2.2 : COMPARISON WITH EXISTING SYSTEM

There is no existing system on WAP Messaging Application. Therefore no comparison can be made but there are more than countable messaging applications on the internet. Among the very famous Internet Messaging program are ICQ, Yahoo Messenger and MSN Messenger. So far, ICQ is the most successful and it's success is proved when it is sold to America Online (AOL) for millions when in fact it is being developed by a group of youngsters. Many

gigantic internet companies such as Yahoo and Microsoft have their own messaging applications: Yahoo Messenger and MSN Messenger respectively because they know that messaging applications are among the successful applications in the internet. Therefore, it shows how well a messaging program is received by the internet users. Hence, chances for WAP Messaging program to succeed is very high.

2.2.1 : CURRENT WAP APPLICATIONS

Maxis has started it's WAP service in June, Celcom has started in late July and Adams in August. All these indicated that WAP is still a very new concept in our country. The websites provided to WAP subscribers have to be coded in Wireless Markup Language (WML) compared to the existing websites that are coded in Hyper-text Markup Language (HTML) using Active Server Pages (ASP) technology. Therefore, it is troubling for existing websites to be changed and the hosts see no significant reasons of changing until they see a brighter future to WAP in Malaysia.

In Malaysia, current most famous WAP application is bank transactions applications where OCBC has started launching their bank application through WAP as an added value to their customers. They are not free because one pays for the services and has to have an account with the bank before one can use it's services.

Maxis in the other hand is offering My Maxis to check username@maxis.net.my account mails, M-News for the latest news, M-Sports to get sport updates, M-Money for stock information, M-Fun for movies and entertainment, M-Buy to get information on various items sold in town, M-Travel, M-Finder and M-Links.

Where else, Celcom's GSM WAP contents are divided into 19 main groups; Top Stories Today, News, Business, Sports, Technology, Weather, Solat, checking e-mails and the list goes on.

2.2.2 : PROBLEMS

The problems with the current WAP applications are there are not as much variety of applications that is available compared to the internet. Other than that, Celcoms users' feedbacks indicate there are 1000 WAP sites provided by Celcom and mostly are irrelevant and it's waste of time for them to scroll down. To be short, WAP users are looking for useful links and relevant applications to be run on their WAP handphones.

2.2.3 : SOLUTION

WAP-MA is not the sole solution but one of the many solutions to resolve current WAP problem that has no interesting application for WAP subscribers. Therefore, WAP-MA is a free of charge and free to be used by anyone as long as they oblige to the rules of the application. It is to promote WAP's usage among Malaysians by providing interesting applications while fulfilling the personal main objective to communicate.

2.3 : TOOLS CONSIDERED

2.3.1 : SOFTWARE ARCHITECTURE

Currently there are many software architecture namely mainframe architecture, file sharing architecture, client/server architecture and web client/server architecture.

2.3.1.1 : MAINFRAME ARCHITECTURE

Mainframe architecture is about central hosting with many clients interacting with it. User can interact with the host using PCs and UNIX workstations. A setback with this architecture is it does not support graphical user interface or able to access multiple databases.

2.3.1.2 : FILE SHARING ARCHITECTURE

File sharing architecture is centralized on a server to download files to several PCs or upload files from several PCs to the server. The setback is the shared usage and the volume of data to be transferred is low to ensure that it is successful.

2.3.1.3 : CLIENT/SERVER ARCHITECTURE

Client/server computing is an extension of the modular programming idea. Client/server architecture recognizes that a large piece of applications can be divided and be executed in a few different computers. Therefore one computer

can be the “server” where database and other related servers are stored. While the other computer or computers will be called “client” where the application program is stored and executed there. This program in the client computer will call (requests service) from the server computer (provide service).

The client/server architecture works when the client will directly connect to the server computer and the results will then be transferred back to the client computer. It has proven to be successful in network computing. Therefore, most business applications nowadays are being written using this architecture.

Web client/server architecture is a subset of client/server architecture. Web client/server architecture still works as the old client/server architecture does but with a little bit of modification. The client will use web browser or other browsers such as handphone if it's WAP application to connect to the web server stored inside the server computer.

Therefore, WAP-MA will choose to integrate this web client/server software architecture because of it's suitability to the project's requirements.

2.3.2 : DEVELOPMENT PLATFORM

2.3.2.1 : MICROSOFT WINDOWS NT SERVER 4.0

Windows NT Server 4.0 is a network operating system because it possesses multi purpose servers. It's functionality is specific to build and deploy business application faster and easier than before. When joined with other Windows NT-related products such as BackOffice family and Windows NT Option Pack. When

these products are integrated into Windows, it makes a complete and powerful environment in building applications of all kind. Windows NT 4.0 Option pack makes it a complete platform for building and hosting web-based application.

Other than being a complete environment in building applications, Windows NT 4.0 is chosen as the development platform because it possesses the following benefits.

- 1) Easy to use – the user interface is friendly, has many icons which is easy to navigate between applications, and on top of that Windows NT looks similar to Windows 95 which is use widely and the world's most popular operating system.
- 2) Security – Windows NT has security unlike it's precedent Windows 95 or Windows 98. One must be able to provide the correct password before allowed to enter to the main window. Therefore Windows NT has been approved as being C2-compliant by the National Computer Security Center (NCSC). C2 security level includes object protection on a user and group bases, password protected, auditing of security-related events.
- 3) NTFS – Windows NT uses New Technology File System which is a new file system technology that was specifically used with Windows NT. The first advantage of using NTFS is if the system fails, then the operating system can user the transactions logs to undo or redo any incomplete transactions. Secondly, NTFS can recover from disk errors caused by bad sectors by cluster remapping.
- 4) Reliability – Windows NT is more reliable compared to Windows 95 or Windows 98 because it uses separate memory space for different 16 bit applications. If one application fails, it would not affect other applications.

2.3.2.2 : UNIX

Unix is also another network operating system like Windows NT. Unix has many unique features. Like other operating system, the Unix system is a control program for computers. It also has a family of utility programs and a set of tools that allows users to connect and use these utilities to build system and applications.

Unix also possesses the following unique features :

- 1) Many Users – Unix can support one to hundred users with each concurrently running a different set of programs. Therefore the cost is low because many people can use the computer at the same time rather than one person to one computer.
- 2) Multitasking – User can run more than one job at a time. User can run several jobs in the background while giving all your attention to the job being displayed on the terminal.
- 3) Kernel – The kernel is responsible for controlling the computer's resources and scheduling user jobs so that each one gets it's fair share including the CPU, disk storage, printers and tape drivers.
- 4) Security – Unix system allows users to protect their data from being accessed by other users. User can share selected data with an effective protection scheme.

2.3.2.3 : LINUX

Linux is a free Unix-type operating system originally created by Linus Torvalds with the assistance of developers around the world. Linux is very similar to Unix.

Linux also has a variety of supporting tools and the number is increasing because Linux has become more famous day after day because it is free.

Linux is also possesses the following features :

- 1) Utilities – Linux also can serves as a server because it has a variety of supporting tools such as Apache web server and can support Oracle and SQL database.
- 2) Stability and Reliability – Linux is a very stable operating system which means that all programs operating would not fail.
- 3) Security – Linux also supports multi-user and each user can only access to the system through password. Other than that, each user's resources is also segregated accordingly by the administrator.

2.3.3 : DEVELOPMENT SERVER

2.3.3.1 : APACHE WEB SERVER

Apache Web Server is the most famous and popular web server since 1995 until now mainly because of it's free license fee. The reason why Apache still remains to be free is due to the believes of it's inventor. He believes "Apache is an organic entity; those who benefit from it by using it often contribute back to it by providing feature enhancements, bug fixes, and support for others in public newsgroups."

Formerly, Apache can run well in Unix Operating System but the Apache team has already comes up with it's new version which can also be run in Windows OS.

The main reasons Apache was chosen as the Web Server for WAP-MA is because of it's free license fee and OS independent. Other administrators might be using Unix or Linux operating system and with Apache Web Server, they are still able to run this WAP-MA.

Apache also includes the following functions :

- 1) Installation made easy - Proper Windows-style install procedure with pre-compiled binaries
- 2) Ability - Ability to run ISAPI server extensions (apart from filters)
- 3) Multithreading - Use of multithreading on Windows
- 4) Modules - Ability to load modules as "dll" files specified in the configuration files at runtime. So there will be no need to recompiled Apache to add modules.

2.3.3.2 : MS INTERNET INFORMATION SERVER (IIS)

IIS is an Internet file and application server included in the Windows NT option pack. IIS is very user friendly because it is easy to configure and can be used alone as a Web Server. IIS also guarantees the same security, networking, and administration and user functionality because it inherits all Windows NT features. IIS also can help administer secure Websites, and to develop and deploy server-intensive Web applications. Other than that, IIS can support a variety of applications such as Virtual Server, Connection to ODBC database, Common

Gateway Interface (CGI), Active Server Pages (ASP) and Secure Socket Layer (SSL).

2.3.3.3 : ORACLE 8.0

Oracle server is a multi-user relational database management system (DBMS) that runs on numerous operating systems. Oracle8, the world's most powerful object-relational database is the heart of the open, standards-based Network Computing Architecture. Network Computing Architecture allows IT organizations to spend less time struggling with interoperability issues and more time focusing on deploying solutions.

Standards-based network architectures make it possible to introduce objects into mainstream enterprise environments. Oracle8's development environment allows users to ease into object-relational functionality while providing the industrial strength properties required by network-based applications.

Network based architectures involve multiple hardware and software platforms and Oracle8 delivers on all the major platforms, including UNIX and NT. Enterprises that explore the competitive advantages of network computing will soon discover that Oracle8's data management, security, reliability, and ease of use, is uniquely designed to meet the new demands of the network era. For mainframe system, parallel servers environments, or desktops, Oracle8 is the database of choice.

On top of all the above positive features of Oracle 8i, it is known to have the following features :

- 1) Security – It has many levels of security and has seamless integration with Windows NT, therefore provides security, a web application environment.
- 2) Compatibility – It is also compatible with many operating system and it has it's own product line such as JDeveloper, Portal-to-go and many more.

2.3.3.4 : MS SQL SERVER

Microsoft SQL Server is a full-featured Sequential Query Language server that is designed for enterprise databases use. SQL Server is a relational database management system for Windows platform. It is designed to meet requirement of a distributed client-server environment. The SQL Server driver enables application to access data in Microsoft SQL Server databases through the Open Database Connectivity (ODBC) interface. Structured Query Language (SQL) is used to access data in a SQL server database. All the client workstations communicate with SQL Server across a star network with TCP/IP protocol

2.3.3.5 : NOKIA WAP SERVER

Nokia WAP Server is basically also a WAP Gateway is a software manufactured by Nokia. This product can run on numerous platforms such as Windows NT, Unix, Sun Solaris and HP-UX. It's function mainly is to connect WAP-enabled terminals to content and applications hosted by Web Server on any other servers on the internet and private intranet.

It was chosen because it was the only WAP Server which allows trial for 30 days before having to pay in order to download from the internet. It's other features include :

- 1) Compliance – Nokia WAP Server is full WAP 1.1 compliance which means all WAP applications which is mostly written in WAP 1.1 standard can run with this server.
- 2) Hosting – Nokia WAP Server API, the open side for open server side development, it can also host applications directly without any external application server and still able to access third party back-end system.
- 3) Easy installation and administration - It is easy to install and administer is made easy through a clear graphical interface and practical command line interface.
- 4) Security – The optional Security Pack is a high-level end-to-end security solution. It encrypts all traffic between the WAP terminal and Nokia WAP Server.
- 5) User Management – It includes a user database where authorized known users and their terminals can be specified.

2.3.4 : DEVELOPMENT SOFTWARE

2.3.4.1 : ORACLE JDEVELOPER

JDeveloper is one of the product in the product line of Oracle Business Components as a complete Java development environment for developing and deploying applications ranging from Java and HTML clients to server-based business components for the internet computing platform.

Oracle JDeveloper includes Oracle Business Components for Java which is a standards-based, server-side framework for creating scalable, high-performance

internet applications. The framework provides design-time facilities and runtime services to drastically simplify the task of building, debugging, customizing, and reusing business components. Business Components for Java implements design patterns necessary for delivering real-world, performant J2EE applications. Also, applications developed with Business Components for Java can be accessed from a wide variety of clients including Web browsers, professional desktop clients, hand-held and wireless devices.

Since WAP-MA will be developed in JSP running on Oracle Database, therefore it is the most suitable development environment. Its compatibility with Oracle Database is one of the main reason why it was chosen to develop and deploy Java Server Pages in this project.

Other functionalities of JDeveloper :

- 1) Platform independent – All web applications developed by JDeveloper are platform independent
- 2) Compatibility – Since JDeveloper is an Oracle product therefore it will be compatible and works well with Oracle Database. It will also run efficiently in Windows environment because it's Operating System independent.
- 3) Simplicity - simplifies the development, delivery, and customization of Java business applications for the enterprise.
- 4) Increase Productivity - Productively author and test business logic in components which automatically integrate with relational databases,
- 5) Flexibility - Flexibly reuse business logic through multiple SQL-based views of data, supporting different application tasks,
- 6) Efficiency - Efficiently access and update the views from Servlets, JavaServer Pages (JSPs), XML clients and thin-Java Swing clients

- 7) Customizable - Easily customize application functionality in layers without requiring modification of the delivered application.
- 8) Database Tools – JDeveloper provides a complete set of tools for integrating database with dynamic Web applications. It provides database programming and database designing tools.

2.3.4.2 : NOKIA WAP TOOLKIT 2.0

Nokia WAP Toolkit is a free software by Nokia for WAP developers to develop their WAP applications easily using the Wireless Markup Language (WML). It offers an environment for creating, testing and demonstrating WAP applications.

Nokia WAP Toolkit include :

- 1) WML and WMLScript Encoders
- 2) Phone simulator on which to display content
- 3) WAP applications debugging support
- 4) User documentation

Nokia WAP Toolkit was chosen because it possesses the following features :

- 1) Compatibility – It can run on Windows environment. IIS web server that had been chosen also can support WML applications.
- 2) Supporting Tools – Other than WAP Toolkit, Nokia also produces Nokia WAP Server that acts as a WAP gateway. Therefore, applications developed using WAP Toolkit would not be clashing when it is tested using the Nokia WAP Server.
- 3) Documentation – Nokia WAP Toolkit has a variety of documents whether it is for the users or the administrators.

2.3.4.3 : UP.SDK DEVELOPMENT KIT

UP.SDK Development Kit is a product by Phone.Com. It is also a WAP Toolkit like WAP Toolkit and it can also run in Windows development platform.

It was not chosen as the WAP Toolkit for WAP-MA project because of the following reasons :

- 1) Time Inefficiency – UP.SDK did not provide any editor space. Therefore, WML applications have to be written using a notepad program and then only run it in UP.SDK program. This is a waste of time and requires double effort in developing the same WML applications compared to using Nokia WAP Toolkit.
- 2) No Supporting Tools – Phone.com did not have a line of WAP products such as WAP Server and WAP Browser. Therefore WAP-MA still needs to use Nokia WAP Server and an incompatibility problem might arise because the application is written and tested using UP.SDK but then it will be running using Nokia WAP Server.
- 3) Lack Of Documentation – UP.SDK provides less documentation compared to Nokia. The developer will have to download these documentation separately because it did not come together with the software.

2.3.5 : DEVELOPMENT TECHNOLOGY

2.3.5.1 : JAVA SERVER PAGES (JSP)

Java Server Pages (more famously known as JSP) is an invention by Sun Microsystems. Although it is invented by Sun Microsystems, any vendors can implement JSP in their own system. JSP allows Web developers and designers to rapidly develop and easily maintain, information-rich, and able to support dynamic Web Pages that leverage existing business system.

As part of the Java family, JSP technology enables rapid development of Web-based applications that are platform independent. JavaServer Pages technology separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content. JavaServer Pages technology is an extension of the Java Servlet Technology. Servlets are platform-independent, 100% pure Java server-side modules that fit seamlessly into a Web server framework and can be used to extend the capabilities of a Web server with minimal overhead, maintenance, and support. Unlike other scripting languages, servlets involve no platform-specific consideration or modifications; they are Java application components that are downloaded, on demand, to the part of the system that needs them. Together, JSP technology and servlets provide an attractive alternative to other types of dynamic Web scripting/programming that offers platform independence, enhanced performance, separation of logic from display, ease of administration, extensibility into the enterprise and most importantly, ease of use.

JSP has many similarities with Microsoft Active Server Pages (ASP). However, it is more easily extensible than ASP and is not proprietary to any vendor or any

particular web server. It is also platform independent. Sun has made the JSP specification freely available to the development community, with the goal that every Web server and application server will support the JSP interface. JSP pages share the "Write Once, Run Anywhere" characteristics of Java technology.

JSP technology is being chosen of the below features :

- 1) Software Architecture – WAP-MA will be developed in web client/server architecture and JSP is suitable for multi-tier Internet applications.
- 2) Compatibility – WAP-MA will be developed in Windows NT and Apache Web Server. JSP can work together with them to provide a comprehensive set of key software technologies which enables secure exchange of information over public networks.
- 3) Interaction – JSP can interact with existing web-based technology and other Java technologies such as JavaScript and Java Applets.
- 4) Suitability – It is suitable for publishing and collecting data on the web.

2.3.5.2 : JAVA SCRIPT

Java Script is a technology by Sun Microsystems. Java Script is an object-based scripting language designed to add programmatic capabilities and cross platform of events, objects and methods to web pages. Java Script can be integrated into JSP files and it is one of the reasons why it is applied in WAP-MA. On top of that, Java Script is also chosen based the following features :

- 5) Interpretation – Java Script is interpreted and not compiled.
- 6) Object Oriented – It is an object-based language and the codes use built extensible objects but no classes or inheritance.

- 7) Compatibility – Java Script can be integrated with and embedded in HTML.
- 8) Loose typing – It has the flexibility to not declare it's variable data types.
- 9) Dynamic binding – It's object references are only checked at run time.

2.3.5.3 : ACTIVE SERVER PAGES (ASP)

Active Server Pages (ASP) a technology by Microsoft is a great tool for creating dynamic web pages. It works by providing users the functionality of a programming language that will generate HyperText MarkUp Language (HTML) for the web page dynamically. ASP combined HTML, Scripting and components to create powerful internet applications that run on Windows NT Server. It can also create dynamic and interactive web pages that include Active-X component and Java Applet as well. ASP is decoded and able to run on IIS which is a web server. Although ASP might be a dynamic web language, but the fact that it is not platform independent made it lose out to JSP.

2.3.5.4 : COMMON GATEWAY INTERFACE (CGI)

The Common Gateway Interface (CGI) is an interaction program between the web server and other programs. Forms processing is a good example. When a user input information in a form normally written in HTML, the information will then be sent to the web server using CGI applications. CGI can save it to either a text file or store it into the database. Usually, CGI is written in Perl Language.

2.3.5.5 : ACTIVE-X

Active-X is a set of technologies that is being used to write software components that inter-operate, regardless of the language used to create them. It can be written in C++, Visual Basic and Delphi. Active-X files has a *.OCX files.

2.3.5.6 : JAVA APPLET

Java Applets is a technology by Sun Microsystem. It is written using JAVA programming language and is platform independent. Therefore it can be supported on many web browsers. It requires an interpreter to interpret it's codes and web browsers usually have this function. The main reason for using Java Applets are to be able to use reusable parts for the applications.

2.3.5.7 : WIRELESS MARKUP LANGUAGE

Wireless MarkUp Language (WML) is an Extensible MarkUp Language (XML) and so it inherits the XML document character set. It is developed for specifying content and user interface for narrowband devices such as handphones and pagers. Currently, WML is the only language to work with small, wireless devices.

WML has 4 characteristics :

- 1) Small display screens – WML works with small display screen that only fits a few lines of text.
- 2) Limited capacity – A wireless device has limited input capacity such as a numeric keypad and a few additional function-specific keys but not a mouse or other pointing device.

- 3) Computational resources – Wireless devices have limited low power CPU, a small memory and power constraints.
- 4) Network limitation – Network for wireless devices offers low bandwidth around 300 bit/second – 10 kilobit/second and high latency of 5-10 second round-trip. Therefore it constraints audio and video streaming which requires high network speed.

CHAPTER 3 : SYSTEM REQUIREMENT

3.1 : PROJECT DEFINITION

WAP-MA is defined as a messaging application where it can be accessed by logging into a website. It is an internet application which means it is not only restricted to internal users in the faculty. The developer of the project will initially become the administrator of the project. The users of WAP-MA could be anybody and it is not restricted as long as the user possesses a handphone to log onto the system.

The functionalities of the WAP-MA will be explained in further details under functional requirements and non-functional requirements.

3.1.1 : FUNCTIONAL REQUIREMENT

The functional requirements of the WAP-MA are divided into 2 modules because there are 2 types of users. The first module would be the user module and the second module will be the administrator module.

3.1.1.1 : USER MODULE

- **Modify Own Details** – User would be able to change personal details just in case there is a change in e-mail address. This is to ensure user e-mail address that is critical because it is the only way for the system to communicate with the user is up-to-date from time to time.

- **Read New Message** – When the user logs onto the system, user will know whether he has any new incoming messages or not. In the incoming function, only new messages are stored there.

User will read the messages and there will be 2 options whether to reply to the message or quit and do nothing about the message. All the messages that had been read before will automatically not be displayed anymore in the incoming messages section.

- **Message A Friend** – User will be able to write a message to a friend. User has the option to choose whether the message is to be delivered for one friend or for multiple friends.

The system will then send the message and put it into the recipient/(s) tables to be retrieved when the recipients log onto the system.

- **Query Administrator** – User will be able to message and pose questions to WAP-MA administrators. Reason is users might need clarification on certain matters and this is easiest to ask for assistance.

- **Add A Friend** – There are many existing users in the system but a user will not be able to search through the whole database to decide on which user to add inside the friends list. He would have to know the user in person and asked for the UserID of the friend he would like to add inside the friends list.

Since WAP-MA is not a chat application where one user would be able to make friend with another user through this program. Therefore, it does not allow any user to search through the database to find out who can be

added into the friends list. It is either the user knows another user in person before this or he would not be able to know the person after using this application.

To add a friend in, the user would have to know another person's UserID. After adding the friend in, the user would have to wait for the other party's consent before he can start to message the or receive message from user. A message would be sent out to the user being added to ask for approval and he would be able to receive this message as soon as he logs into the applications.

- **Delete A Friend** – Each user will have a list of friends where he can message and receive message from. The user will also have an option to delete a friend from the friends list. Then, this friend who was being deleted will be informed about this deletion so as to tell him not to expect any message from this particular user in the future.
- **View history** – Each user will have a list of histories about incoming and outgoing messages. The user will have to choose a friend from his list of friends to determine the incoming and outgoing messages that he will see. The messages will be arranged and sorted out according to time and date.
- **Approve Awaiting Request** – This list shows which existing users in the system wishes to add him into their friends' list. The user will then have to decide whether to accept or reject them. Only one user is accepted or rejected at one time. A message will be sent out to the requestor whether it the user who they had been requesting for accepts or rejects.

If the user accepts the request, then he would have a choice whether as to add them back in return.

- **Query Admin** – When face with problems regarding the system, user will then be able to pose questions to the administrator of WAP-MA.

3.1.1.2 : ADMINISTRATOR MODULE

There are 2 types of administrator. This is to distinguish one type of administrator from another. The first type who has more authority and power is referred as 'super administrator' and the less power administrator is referred as 'normal administrator'.

Super Administrator's Module

- **Modify Own Details** _ Super administrator will be able to modify his own details in the database
- **Query Administrator** – Query all existing normal administrators in WAP-MA.
- **Add Administrator** – The only way to add another administrator is through super administrator. It would only be done when service of an normal administrator is needed.
- **Delete Administrator** – Super administrator will have power to delete an existing administrator from the system because of any reason and will send an e-mail to inform the normal administrator about the deletion.

Normal Administrator's Module

- **Modify Own Details** – Normal administrator will also be able to modify his own details in the database
- **Query User** – A normal administrator can query any existing user in particular or all users. This is to keep track of the amount of users.
- **Add User** – User can register with WAP-MA through 2 ways. The first one is to register WAP-MA through handphone and the second way is through administrator. Once the administrator adds the user in, the user will be authorized straight away.
- **Delete User** – Other than being able to add user, administrator can also delete any existing user. The reason can vary from the user not abiding rules or the server might not be able to cope with the tremendous number of users.
- **Contact User** – Administrator can also contact or inform users. Therefore, contacting users is an important function too.
- **Answer Questions** – Contacting always goes both ways. An administrator can contact users and vice versa. Therefore, administrator is obliged to answer users' questions also at the same time.
- **Authorize New User** – New users registered to the system will not be approved immediately and they could not login to the system. Administrator will have to authorize these new registrations and an e-mail will be sent out to the each user informing them that their registrations

have been successful. Then only, they can start to login into the WAP-MA system.

- **Reject New User** – Administrator has a choice whether to authorize or reject new registrations. Rejection can be due to server congestion or already reach the maximum number of users. Anyhow, an e-mail will be sent out to inform user about the rejection but user can still register again with the system hoping they might be approved someday.

3.1.2 : NON-FUNCTIONAL REQUIREMENT

Security

The system must have sufficient security measures to ensure the program runs smoothly without fear of being corrupted. Therefore, only administrators are authorized to control existing users and each user would not have access to other user's data.

Some security measures include :

- 1) After the administrator have authorized the new user, an e-mail will be sent out together with their unique login UserID that has been chosen by them in the first place. Password will not be sent together in the mail. This is to ensure even though other people may have access to the e-mail but they would not know the password and thus will not be able to gain access.

The system has a policy not to reveal the password unless it is really necessary.

- 2) Each user will need a login userID and password to access the system.
- 3) Only the administrator will be able to authorize new user.
- 4) Even administrator is not allowed to change users' details because even administrators might misuse their power for personal gains. They who want to imitate the user might modify the target user's e-mail address and put in their own. After that, they could click on the forgotten password option and the password would be sent to them immediately.
- 5) Password in the database will be encrypted.

Reliability

The system should be reliable in performing its functions as expected by both user and administrators. The system would be able to identify the correct login to give access to the correct users. For example, user will only expect to see his information whether it be his personal details or the friends list without expecting to see other user's data. The system would have to ensure the users that their datas inside the system is secure and strictly confidential to other users.

Availability and Manageability

The system should be available to users only through handphones and to administrators through internet browsers. It must available all the time other than when the server down. This is to ensure the user's messages will be delivered successfully.

The system is easy to manage for the administrators because their jobs are made easy by easy navigation through the system such as clicking and not filling in the blanks manually.

Usability

The system must be easy to use and user friendly because most users are not very computer literate. The main reason they joined WAP-MA is because they need the messaging application that saves their time and therefore it the system must meet the users objectives by providing friendly and easy to understand interface. If the system is difficult to use, the user will have read carefully what to do next and this will be a waste of time.

User Interface

The user interfaces must be user-friendly and easy to navigate by having clear and consistent menus so that the user would know what to expect next rather than being surprised by what they are actually getting. The interface must be easy to navigate using the scroll bar on the handphone because the user would not be using the conventional 'mouse click'.

3.2 : APPROACH

In WAP-MA, prototyping approach is used. Prototyping is an information-gathering technique to modify system plans with minimum expense and disruption because changes are made from the time to time to improve the system rather than when the system is fully developed. Prototyping stress the importance of the potential users and it seeks user reactions, suggestions, innovations, and revision plans in order to make improvements to the prototype even in the development stage of the project.

Prototyping is used because of the following advantages :

- 1) Early changes – Successful system requires early and frequent user feedback, which is used to help modify the system and making it more responsive to the user's actual needs and expectations from the system.
- 2) Scrapping – Undesirable module could be scrapped off which is not what users and analysts want from the system. This can saves time on developing the unwanted module right from the start rather than waiting till the end of the project.
- 3) User's System – User has always been the reason why a system is developed. With user involvement in prototyping approach, users can voice out their needs, expectations and gives feedbacks on the system from time to time and modifications can be made to meet these expectations. This could indeed save time because the end product would not have to do a lot of modifications since during the development, it has thus been doing that.

3.3 : TOOLS CHOSEN

Based on the literature review in Chapter 2, WAP-MA will be implemented using web client/server N-tier architecture because of its networking environment. N-tier architecture will be discussed in details in Chapter 4.1 Project Architecture.

The system will be developed in Windows NT platform with Apache Server as the Web Server, Oracle 8i as the Database Server and faculty's Sun Solaris SMTP Mail Server as the mail server. All the servers will be setup exactly like shown in the N-tier architecture. Web Server would be in the middle tier while the Database Server and the Mail Server would be in the third tier. A WAP Server is put in between the client-tier and the middle-tier is needed as a middle-man to enable the WAP handphone to communicate with the Web Server.

CHAPTER 4 - SYSTEM DESIGN

The clients of this project will be internet browser and the handphone. JSP technology will be needed for the internet browsers and it will also integrate WML language develop handphone applications and interface. JSP and WML will be the two main programming languages with the help of JavaScripts and HTML.

The overall view for the system architecture is as follows:



The client for WAP-MA consists of 3 types of clients. The first type is a internet browser (Internet Explorer or Netscape Navigator). The second type is the phone simulator by Nokia WAP Toolkit. The third type would be the handphone itself.

The client for WAP-MA consists of 3 types of clients. The first type is a internet browser (Internet Explorer or Netscape Navigator). The second type is the phone simulator by Nokia WAP Toolkit. The third type would be the handphone itself.

CHAPTER 4 : SYSTEM DESIGN

4.1 : PROJECT ARCHITECTURE

WAP-MA will be implemented using the N-tier web client/server architecture. The overall view for the system architecture is as follows :

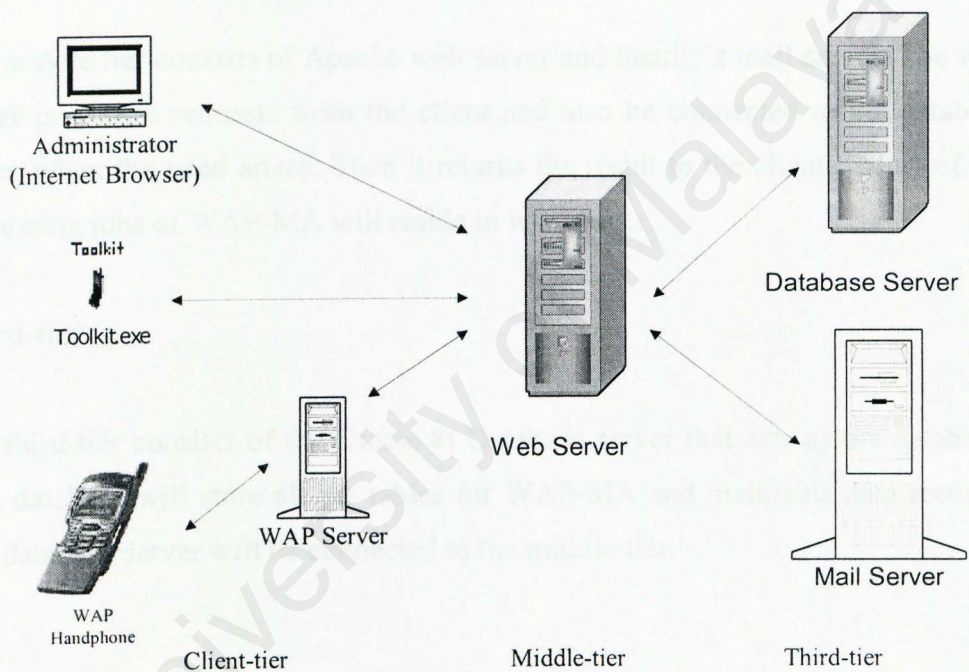


Figure 4.1: WAP Overview

Client-tier

Client-tier for WAP-MA consists of 3 types of clients. The first type is a internet browser (Internet Explorer or Netscape Navigator). The second type is the phone simulator by Nokia WAP Toolkit. The third type would be the handphone itself

but handphone needs a WAP Server which acts like a WAP gateway in order for the phone to be connected to the Web Server. Therefore the WAP Server will be considered another tier.

Handphone as a client would be in a 4-tier architecture while the web browser and the WAP Toolkit program would both be a 3-tier architecture.

Middle-tier

The middle tier consists of Apache web server and faculty's mail server. The web server processes requests from the client and also be connected to the database server when the need arises. Then it returns the result to the clients. Most of the processing jobs of WAP-MA will reside in web server.

Third-tier

The third-tier consists of the Oracle 8i Database server that acts as the database. This database will store all the tables for WAP-MA and maintains data records. The database server will be connected to the middle-tier.

4.2 : APPLICATION DESIGN

WAP-MA has 2 main modules had been described in the functional requirement. It has the user module and the administrator module. However, there are 2 levels of administrator namely the super administrator and normal administrator. In each module there are sub modules.

For an overview look of the WAP-MA, a context level diagram as below shows 2 types of entities are basically involved which are the user and administrator. The relationships between the entities are the applications are shown very clearly with the inputs to and outputs from the system towards the two entities.

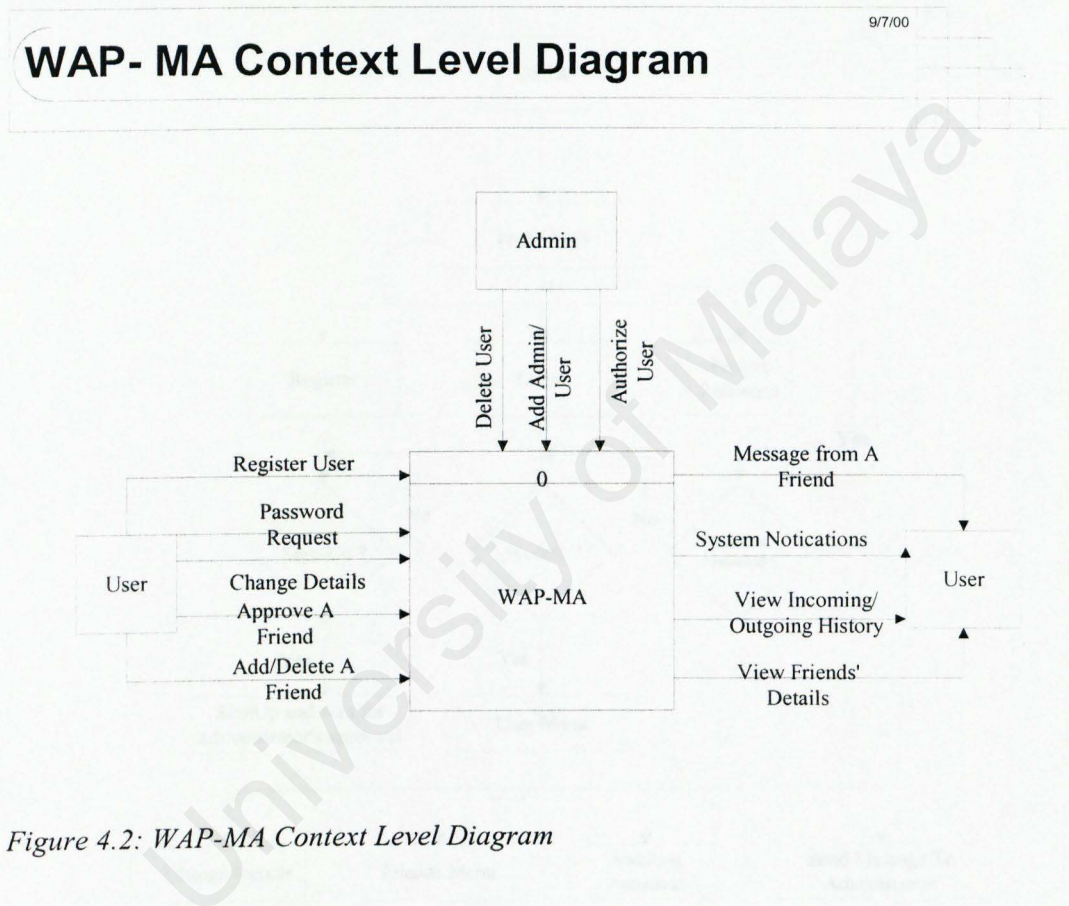


Figure 4.2: WAP-MA Context Level Diagram

User Module

The flow chart for user when user logs into the application is in Figure 4.3.

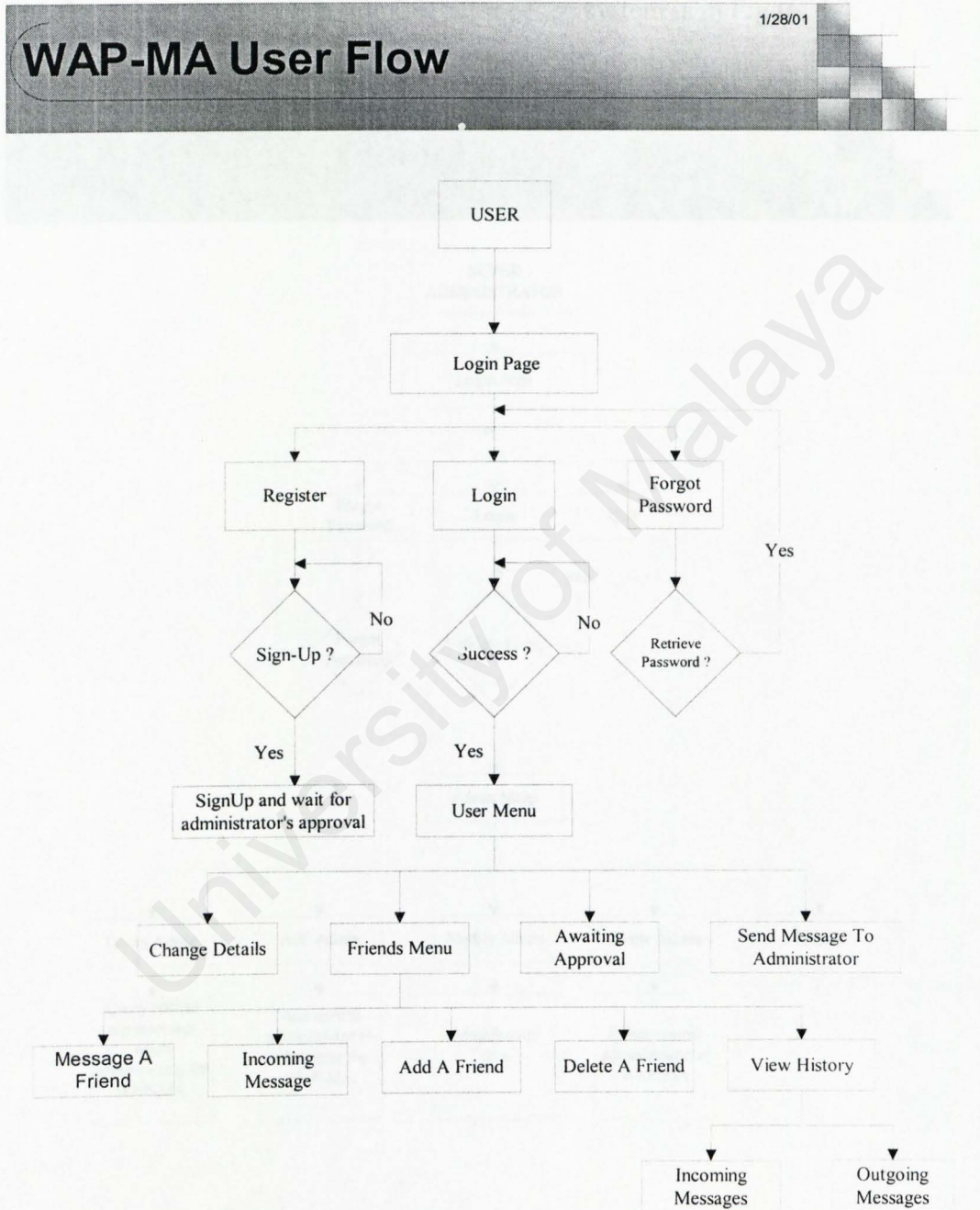


Figure 4.3 : WAP-MA User Flow

Administrator Module

Super Administrator Sub-Module

The flow chart for the super administrator of the system is in Figure 4.4.

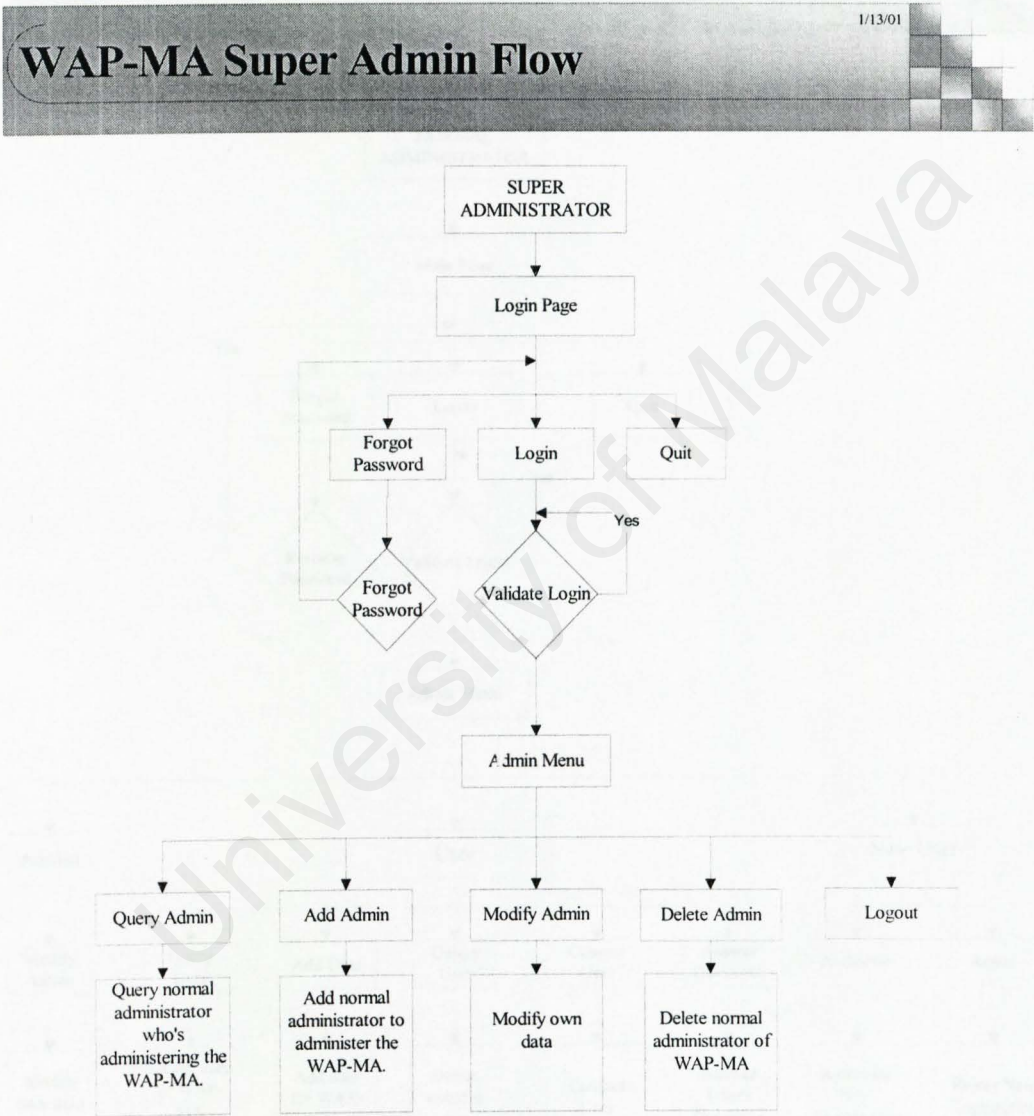


Figure 4.4 : WAP-MA Super Admin Flow

Normal Administrator Sub-Module

The flowchart for the normal administrator of the system is in Figure 4.5.

WAP-MA Normal Admin Flow 1/18/01

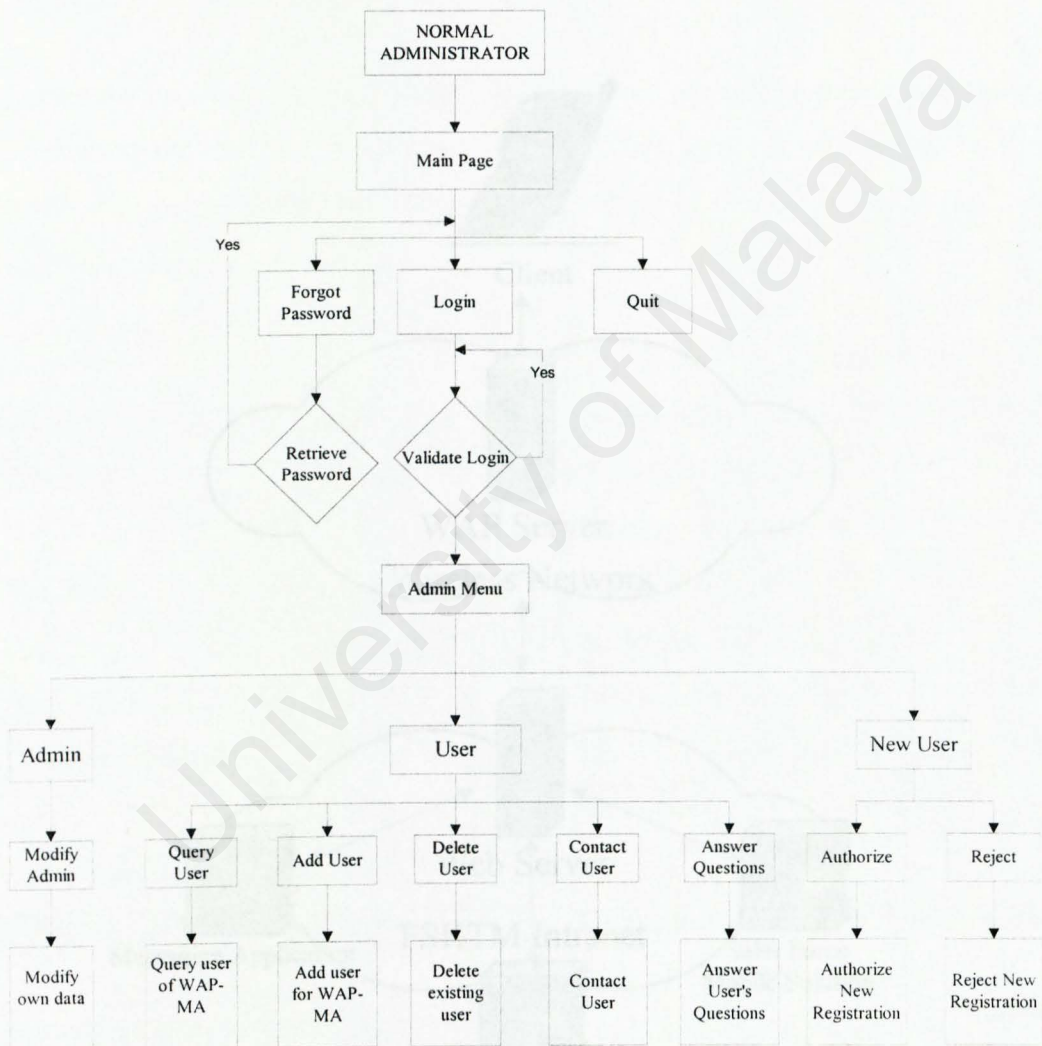


Figure 4.5 : WAP-MA Normal Admin Flow

4.3 : NETWORK DESIGN

The three WAP applications are Messaging Applications, Sales Force Mobile Solutions from Goh Bee Yen, Cinema Ticket Reservation from Goh Chiau Foon will be stored inside the same IIS Web Server inside FSKTM intranet. Once, the user logs onto the WAP website, they will have an option to choose which application that they want to access. The network design diagram is in Figure 4.6.

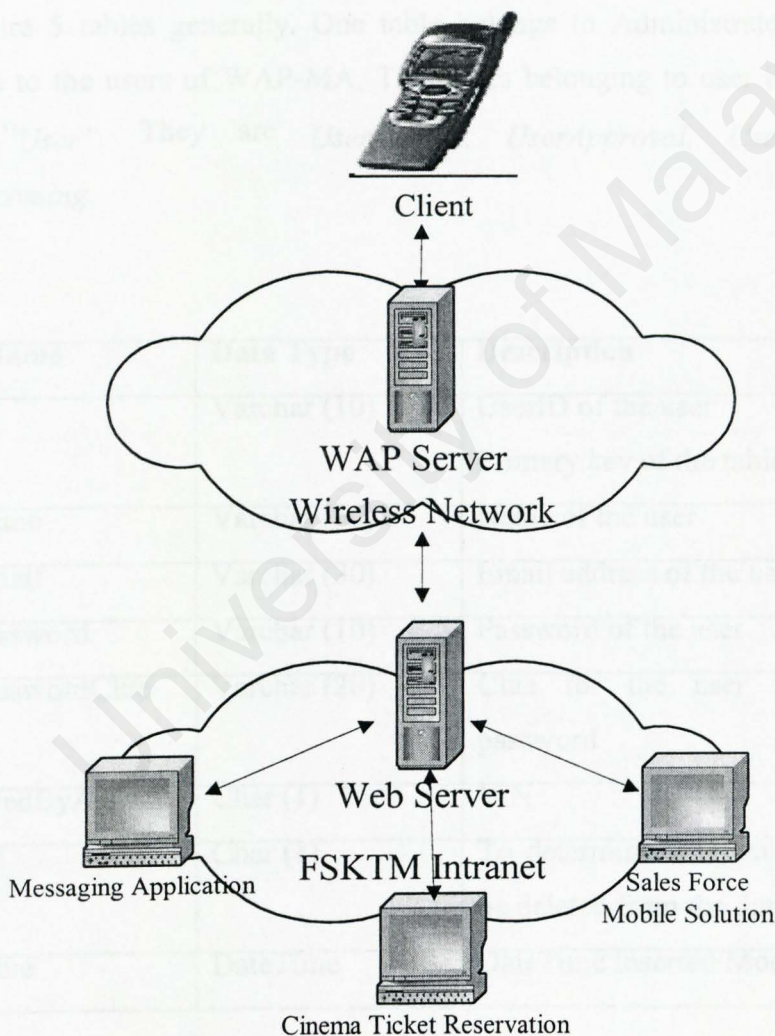


Figure 4.6 : WAP-MA Network Design

4.4 : DATABASE DESIGN

Oracle 8i will be used as the database server for the system. The database is connected to Apache Web Server using Open Database Connectivity (ODBC). With ODBC, the Oracle Server is mapped to the web application stored inside the Apache Web Server and this enable access to database on remote computer through network.

There are 5 tables generally. One table belongs to Administrator and the rest belongs to the users of WAP-MA. The tables belonging to user begins with the word "User". They are *UserDetails*, *UserApproval*, *UserFriends* and *UserIncoming*.

Field Name	Data Type	Description
UserID	Varchar (10)	UserID of the user Primary key of the table.
UserName	Varchar (30)	Name of the user
UserEmail	Varchar (30)	Email address of the user
UserPassword	Varchar (10)	Password of the user
UserPasswordClue	Varchar (20)	Clue for the user to remember password
ApprovedByAdm	Char (1)	Y/N
Deleted	Char (1)	To determine whether this user can be deleted from the database.
DateTime	DateTime	DateTime Inserted/Modified

Table 4.1 : Field in the 'UserDetails' table

Table 4.1 : Field in the 'UserFriends' table

Field Name	Data Type	Description
UserID	Varchar (10)	The UserID of the person asking permission from another existing user to become his friend. Primary key of the table is combination of UserID & NewFriendID. Foreign key linkage to UserID in the <i>UserDetails</i> table
NewFriendID	Varchar (10)	UserID of the person being requested. Foreign key linkage to UserID in the <i>UserDetails</i> table
DateTime	DateTime	DateTime Inserted/Modified

Table 4.2 : Field in the 'UserApproval' table

Field Name	Data Type	Description
UserID	Varchar (10)	UserID of a certain user. Primary key is UserID & FriendID Foreign key linkage to UserID in the <i>UserDetails</i> table
FriendID	Varchar (10)	UserID of a friend to the UserID above. Foreign key linkage to UserID in the <i>UserDetails</i> table

Table 4.3 : Field in the 'UserFriends' table

Field Name	Data Type	Description
UserID	Varchar (10)	Primary key of the table is UserID, FriendID, Msg & DateTime. The combination of UserID and FriendID will form a foreign key. Foreign key linkage are (FriendID, UserID) linking to (UserID, FriendID) in <i>UserFriends</i> table
FriendID	Varchar (10)	FriendID is the UserID of the sender.
Msg	Varchar (100)	Message from the friend
NewMsg	Char (1)	Y/N
DateTime	DateTime	DateTime Inserted/Modified

Table 4.4 : Field in the 'UserIncoming' table

Finally, there is a table specially created for the administrators to store their personal details like the table previously created for the users of the applications.

Field Name	Data Type	Description
AdminID	Varchar (10)	AdminID of the Administrator Primary key is AdminID
AdminName	Varchar (30)	Name of the Administrator
AdminPassword	Varchar (10)	Password of the Administrator
AdminPasswordClue	Varchar (20)	Clue for the admin to remember the Password.
AdminEmail	Varchar (30)	Email address of the user

AdminLevel	Char (1)	The level of the Administrator
DateTime	DateTime	DateTime Inserted/Modified

Table 4.5 : Field in the 'AdminDetails' table

All in all there are a total of 5 tables in the WAP-MA. They also had the field DateTime because it is a good way of keeping track of the activities and it might be of use when the in situation such as determining when or which is the first incoming message and then the next outgoing message when the option in the Friends Menu 'View History' is chosen.

4.4.1 : ENTITY-RELATIONSHIP MODEL

Entity-relationship model is used to describe the structure of the database and the relationship among entities in the system. The objects which are using the system are called 'entities' while the association among the entities are called 'relationship'.

Therefore, the E-R model assists in defining the connection between relationships between the entities within the system. WAP-MA has 2 types of entities; administrator and user. Users are also divided into 2 types which is new and existing user.

There are 3 entity-relationships diagram to represent the relationships between administrator and user inside WAP-MA.

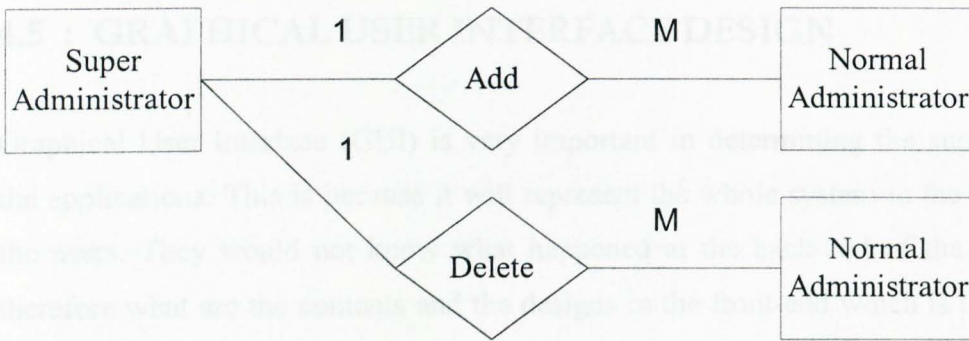


Figure 4.7 : Super Admin Privilege Entity Relationship Diagram

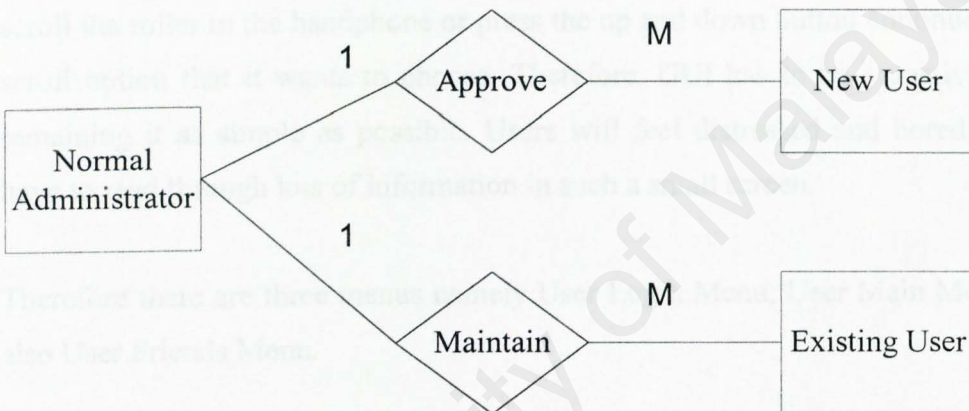


Figure 4.8 : Normal Admin Privilege Entity Relationship Diagram

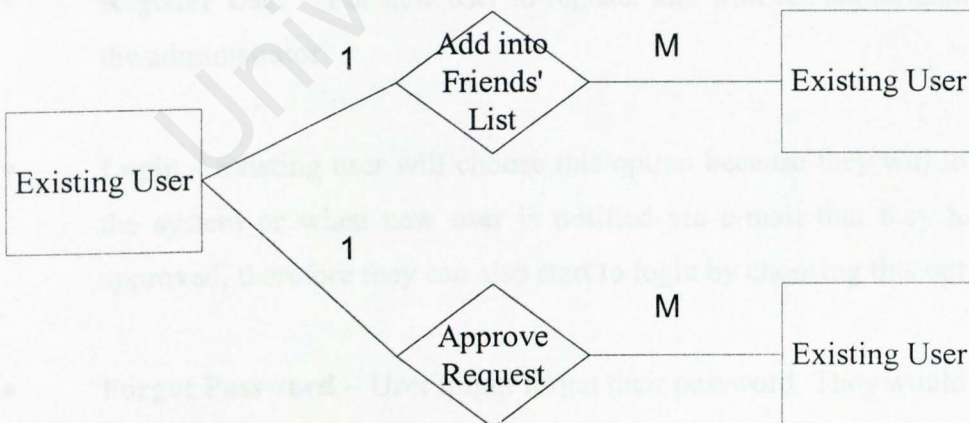


Figure 4.9 : User Privilege Entity Relationship Diagram

4.5 : GRAPHICAL USER INTERFACE DESIGN

Graphical User Interface (GUI) is very important in determining the success of the applications. This is because it will represent the whole system in the eyes of the users. They would not know what happened at the back-end of the system therefore what are the contents and the designs in the front-end which is the GUI are very important.

Navigation is very limited because user of the WAP-MA applications can only scroll the roller in the handphone or press the up and down button continuously to scroll option that it wants to choose. Therefore, GUI has to be attractive while remaining it as simple as possible. Users will feel distressed and bored if they have to read through lots of information in such a small screen.

Therefore there are three menus namely User Login Menu, User Main Menu and also User Friends Menu.

First Menu : User Login Menu

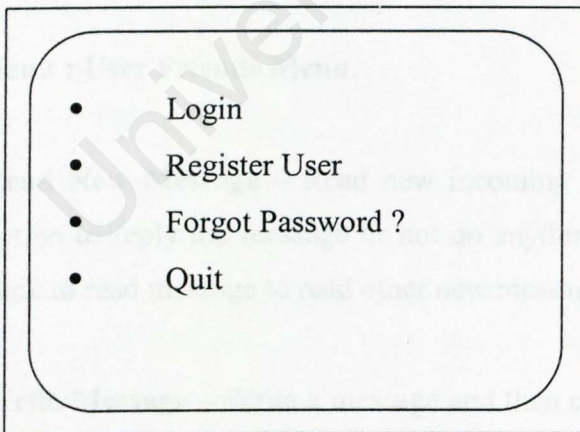
- **Register User** – For new user to register and wait for authorization from the administrator
- **Login** - Existing user will choose this option because they will login into the system or when new user is notified via e-mail that they had been approved, therefore they can also start to login by choosing this option.
- **Forgot Password** – User might forget their password. They would have to fill in their UserID and the password will be sent to the registered e-mail account.

- **Quit** – To quit the program

For a better understanding of how the handphone screen looks like, there are 2 screen designs below.



Screen 4.1 : WAP-MA First Page



Screen 4.2 : Login Menu

The second and third menu's contains the followings :

Second Menu : User Main Menu

- **Modify Own Details** – The user will change or update their details such as a change in their e-mail address.
- **Friends Menu** – This will lead the user to the third menu : User Friends Menu
- **Approve Request** – User will have to see whether any user has added him in and then whether chooses to authorize or otherwise.
- **Query Admin** – User can ask or query administrator of WAP-MA regarding anything.
- **Quit** – Quit from this menu and go back to the main menu.

Third Menu : User Friends Menu.

- **Read New Message** – Read new incoming messages and then have an option to reply the message or not do anything about it. The user can go back to read message to read other new messages too.
- **Write Message** – Write a message and then choose the recipients that the message is intended.

- **Add A Friend** – User will have to key in another UserID in order to add that person into the friends list. There is no option to select another user from the database.
- **Delete A Friend** – Flexibility to add is given, therefore user is able to delete a friend from the friends list.
- **View History** – User will have to choose whether to view all incoming or outgoing messages. Then the next step is to choose a friend to view all the messages.

Administrator who is also another user of the applications also need to be comfortable with the system before he can use it. Hence, GUI is also very important to the administrator.

However, both types of administrators only have one menu. Super Administrator's menu is called Super Admin Menu and Normal Administrator's menu is called Normal Admin Menu.

Super Admin Menu

- **Modify Own Details** – Modify or update own details
- **Query Administrator** – Query all the normal administrators of the system.
- **Add Administrator** – Super administrator will be able to add another normal administrator.

- **Delete Administrator** – Deleting administrator is one of the priorities owned by the super administrator because his service might not be required anymore.
- **Quit** – To quit the program

Normal Admin Menu

- **Modify Own Details** – Modify or update own details
- **Query User** – Query a particular user or all existing users of the system.
- **Add User** – Normal administrator will be able to add user of WAP-MA.
- **Delete User** – Deleting user is one of the priority owned by the normal administrator because the user might not be an active user anymore or he did not abide to the rules of the system. An email will be sent out to notify the user of his deletion.
- **Contact User** – Administrator can contact users when the need arises such as wanting to inform users about certain issues regarding to the system.
- **Answer Questions** – Users can also contact administrators and query them. Therefore, administrator will have to answer users' questions and send a message back to them.

- **Authorize or Reject New User** – The administrator will check the lists of new users waiting for authorization. He will either approve or reject the user and e-mail will be sent out to the users whether to notify whether they are being approved or rejected.
- **Quit** – To quit the program

4.6 : USER AND HUMAN FACTORS

Users and human factors are indeed very important. It has many similarities with other web applications also. A few factors are taken into account when developing WAP-MA. They are :

- 1) Error messages would be prompted out if errors occurred. Then, user would have to click 'back' to the previous place where the mistake is being made. They would have to correct their mistakes.
- 2) Outgoing messages should not contain more than 100 characters because the maximum characters allowed for each message is 100 characters. Thus, when keying in message, user will only be allowed to input 100 characters.
- 3) User would be notified from time to time whether they are being added or deleted by other users through messages via WAP-MA. When a user is banned or in other words deleted by the administrator, an e-mail will be sent out to notify the user that he would not be able to login into the system again so as not keep the user in suspense what has happened to his account.

4.7 : CONCLUSION

System design has always been an important section in developing a system .It outlines how the system should be designed and what should be added into the design. Therefore, the system can later on be developed using systems design as a guideline. The next two chapters namely system integration and system testing will integrate and test out the whole system to ensure the system is indeed reliable, durable and efficient.

Tools	Details
Windows NT	Windows NT 4 with Service Pack 3. Most of the servers are required to run on Service Pack 3.
Apache Web Server	Used to host the system's web pages. It is to be accessible by administrators through internet browser and users through WAP handphones. A folder called 'www' is created to signify that all files belong to this system goes under this directory. 2 short sessions are created under 'www' because there are 2 types of users. A dummy existing internet browser was used to type in this browser address: 202.163.109.163/8080/admin/manager while the WAP browser address is 202.163.109.163/8080/wap/manager. When WAP handphones access to the system.
ORACLE	Oracle is the database for the WAP-MA system. All the 3 tables of the system will be stored in the database under the username of 'wap'. Oracle requires the concept that every user will have different tables. Therefore, when logging into SQL*Plus (an application of Oracle), username must be wap before the tables can be accessed.
SMTP MAIL	The faculty Mail Server will be configured into the system for

CHAPTER 5 : SYSTEM IMPLEMENTATION

5.1 : DEVELOPMENT TOOLS IMPLEMENTATION

WAP-MA is developed using client-server software architecture, running under Windows NT integrating Apache Web Server, Oracle 8i Database and also a Mail Server. For further details about these development tools and how it was setup to develop the system, please refer to table 5.1.

Tools	Details
Windows NT	Windows NT 4 with Service Pack 5 is installed. Most of the servers are required to run on Service Pack 5.
Apache Web Server	Used to host the system's web page to be accessible by administrators through internet browsers and users through WAP handphones. A folder called 'wam' is created to signify that all files belonging to this system goes under this directory. 2 more additional folders are created under 'wam' because there are 2 types of users. Administrator using internet browser will have to type in this link http://202.185.109.163:8080/wam/admin/login.jsp while user types in http://202.185.109.163:8080/wam/client/login.jsp on their WAP handphone to access to the system.
Oracle 8i	Oracle is the database for the WAP-MA system. All the 5 tables of the system will be stored in the database under the username of 'yhp'. Oracle practises the concept that every user will have different tables. Therefore, when logging into SQLPlus (an application of Oracle), username must be yhp before these tables can be accessed.
SMTP Mail	The faculty Mail Server will be integrated into the system for

Server	<p>sending mails purposes. The Mail Server is a built-in application from Sun Solaris 7.0. The mail server is links to all the computers inside the faculty. Due to the fact that it can be accessed in any computer in the faculty, administrators can access the system anywhere in the faculty.</p> <p>The Mail Server's DNS is fsktm.um.edu.my and its IP is 202.185.108.1.</p>
--------	---

Table 5.1 : Development Tools table

5.2 : MODULE IMPLEMENTATION

WAP-MA has three types of users. They are Super Admin, Normal Admin and Users. Basically, administrators will administer the system to ensure it is running smoothly while the users will use the system for messaging.

In this system, there are many similarities between Super Admin and Normal Admin web pages such as the design and the functions. Super Admin will be able to administer Normal Admin while Normal Admin will administer Users.

Firstly, section 5.2.1 will discuss about administrator's modules and how it will be implemented and the after that section 5.2.2 discusses about user's module and how it will be implemented.

5.2.1 : Admin Module Implementation

The Super Admin module will have 4 main functions in it. They are 'Modify Own Details' and the other three functions are to administer Normal Admin which are 'Query Admin', 'Add Admin' and 'Delete Admin'. How these modules are related to each other is best illustrated in a form link diagram as shown in Figure 5.1.

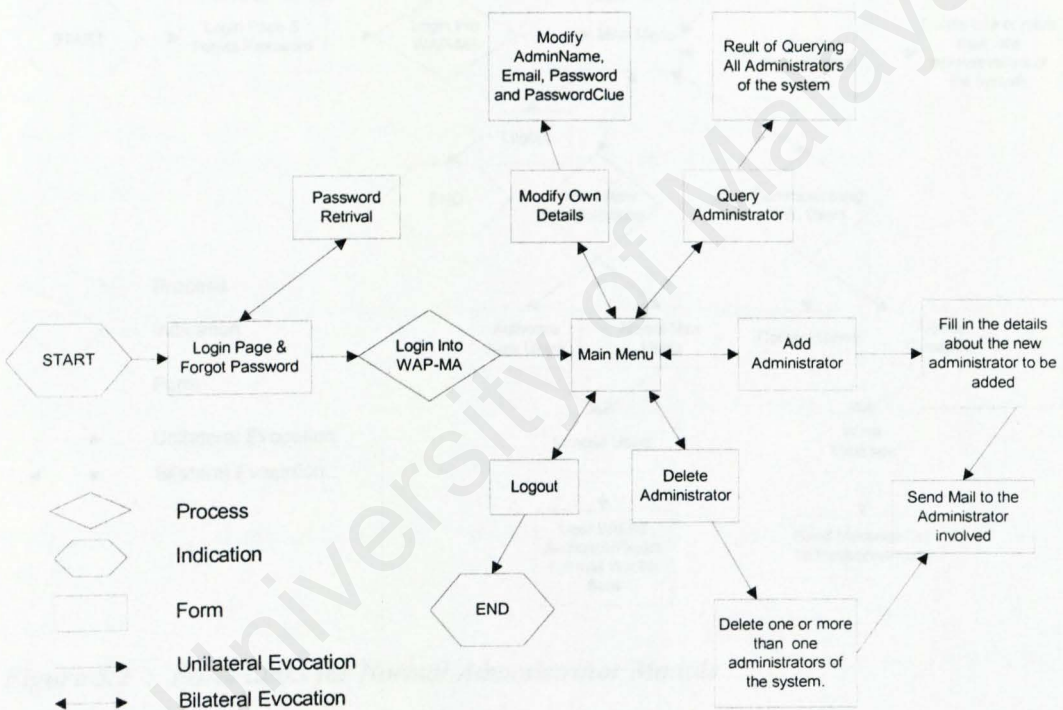


Figure 5.1 : Form Links for Super Administrator Module

Figure 5.2 below shows form links for Normal Administrator module.

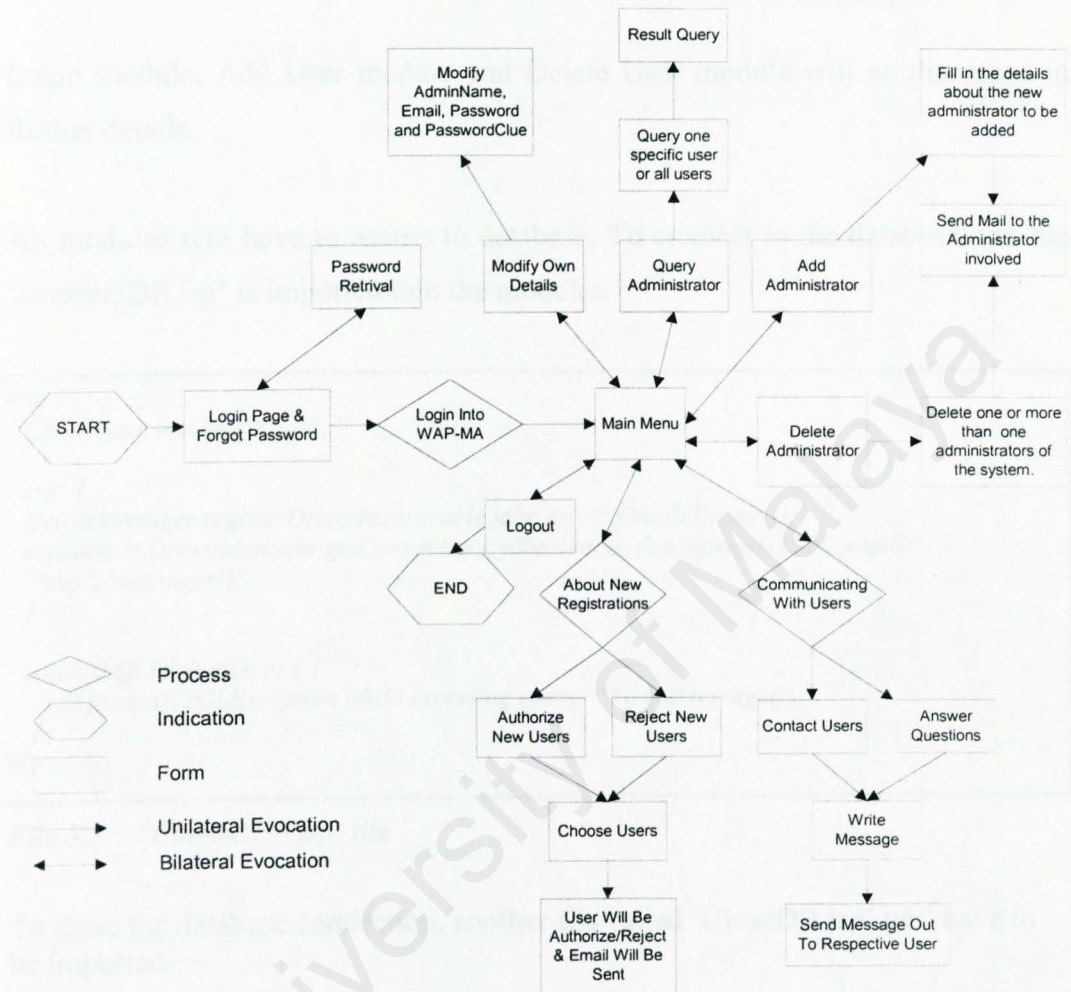


Figure 5.2 : Form Links for Normal Administrator Module

If observed closely, there are many similar functions in the 2 types of administrators' modules. Both administrators have a login module to verify their login. After successfully logged into the system, both administrators can modify their details. Besides that, they can administer by querying, adding and deleting normal administrators if he is super administrator and users if he is a normal administrator. However, normal administrator has additional functions such as

dealing with new registrations, whether to authorize or reject their applications. They can also contact users and vice versa.

Login module, Add User module and Delete User module will be discussed in further details.

All modules will have to access to database. To connect to the database, the file 'ConnectDB.jsp' is imported into the modules.

```
<%
    Connection myConn = null;

    try {
        DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
        myConn = DriverManager.getConnection("jdbc:oracle:thin:@wap1:1521:wapdb",
            "yhp", "manager");
    }

    catch (SQLException e) {
        out.println("SQLException while executing query: "+e.getMessage());
    }
%>
```

File 5.1 : 'ConnectDB.jsp' file

To close the database connection, another file called 'CloseDB.jsp' will have to be imported.

```
<%
    // Close database connection , query statement and result set

    stmt.close();
    myConn.close();
%>
```

File 5.2 : 'CloseDB.jsp' file

• Login Module

First of all, administrator will login into WAP-Ma system accessing the file 'login.jsp'. Administrator will login by providing LoginID and Password. LoginID will be stored in AdminID and the password provided will be stored in AdminPassword.

```
<form method="POST" action ="Admin_LoginVerify.jsp">
<table BGCOLOR="#a0b8c8" cellpadding=4 cellspacing=2 BORDER="2" cols="48%,"*
width="400">

<tr>
<td> AdminID : </td>
<td><input name="AdminID" size=15 maxlength=10> </td>
</tr>

<tr>
<td>Password : </td>
<td> <input name="AdminPassword" type="password" size="15" maxlength=10> </td>
</tr>

</table>
<br>
<input type="submit" value="Submit">
<input type="reset" value="Reset"><br>
```

File 5.3 : Part of the file 'Login.jsp'

This file will accept the values input by the administrator and post it into another file called Admin_LoginVerify.jsp. The next file will check for validation whether it is a correct login or otherwise. The few SQL statements and actions to be taken if it is a validate login and otherwise is shown in File 5.4. To connect to the database, the code is as below

```
<%@ include file="ConnectDB.jsp"%>
stmt = myConn.createStatement ();
```

Therefore to execute any database query, the code will be

```
stmt.executeQuery
```

```

if ( AdminID.equals("") || AdminPassword.equals("")) {
    EmptyFieldFound = 1;
}

else {
    try {
        ExistAdmin = stmt.executeUpdate("Select * from AdminDetails where AdminID = " +
        AdminID + " AND AdminPassword= " + AdminPassword + " ");
        rs = stmt.executeQuery ("Select * from AdminDetails where AdminID = " + AdminID + "
        AND AdminPassword= " + AdminPassword + " ");
        if (rs != null) {
            while (rs.next()) {
                AdminID = rs.getString ("AdminID");
                AdminLevel = rs.getInt ("AdminLevel");
                session.putValue("SessionID", AdminID);
                session.putValue("SessionLevel", Integer.toString(AdminLevel));
            } //end while
        } //end if
    } //end try

    catch (SQLException e) {
        out.println("SQLException while executing query: "+e.getMessage());
    }
} //end else

if (EmptyFieldFound == 1) { %>
    <h3>Empty Field Found. Please
    <a href="Admin_Login.jsp">click here</a>
    to try login again.</h3> <% }

else if (EmptyFieldFound == 0 && ExistAdmin == 0) { %>
    <h3>AdminID or Password can be wrong.Please
    <a href="Admin_Login.jsp">click here</a>
    to try login again.</h3> <% }

else if (EmptyFieldFound == 0 && ExistAdmin == 1) {
    if ( AdminLevel==1 ) {
        response.sendRedirect("Admin_MainMenu1.jsp");
    }
    else if ( AdminLevel==2 ) {
        response.sendRedirect("Admin_MainMenu2.jsp");
    }
}

else { %>
    <h3>Login attempt failed. Please
    <a href="Admin_Login.jsp">click here</a>
    to try login again.</h3>
    <% } %>

```

File 5.4 : Part of the file 'Admin_LoginVerify.jsp'

The values are taken into the 'Admin_LoginVerify.jsp' file. The file will check for empty fields. If they were found, the system will prompt the error "Empty fields found. Please [click here](#) to login again." The click here link will bring the administrator back to the login page.

If no empty field were found, the system will then call the database to verify the AdminID and AdminPassword with the statement below.

```
ExistAdmin = stmt.executeUpdate("Select * from AdminDetails where AdminID = '" + AdminID  
+ "' AND AdminPassword= '" + AdminPassword + "'");
```

```
rs = stmt.executeQuery ("Select * from AdminDetails where AdminID = '" + AdminID + "  
AND AdminPassword= '" + AdminPassword + "'");
```

If it is incorrect, then another error message read "AdminID or Password can be wrong. Please [click here](#) to try login again." will be prompt. The administrator can go back to the login page to try again.

If no empty field were found and the login is correct, then the system will capture the AdminID and AdminLevel to be stored inside sessions so that other files inside the system can access the same information. AdminID is stored in the variable SessionID and AdminLevel is stored in SessionLevel. The codes are shown below.

```
AdminID = rs.getString ("AdminID");  
AdminLevel = rs.getInt ("AdminLevel");  
session.putValue("SessionID", AdminID);  
session.putValue("SessionLevel", Integer.toString(AdminLevel));
```

Based on the value stored in AdminLevel, the system will determine whether the login administrator is a super admin or normal admin. This is to enable different Main Menu to be loaded.

```
if ( AdminLevel==1 ) {
    response.sendRedirect("Admin_MainMenu1.jsp");
}

else if ( AdminLevel==2 ) {
    response.sendRedirect("Admin_MainMenu2.jsp");
}
```

• Add User Module

Inside the Main Menu, a normal administrator will be able to add user by accessing the page 'User_Add.jsp'. Super administrator has the same function to add the administrator but by accessing 'Admin_Add.jsp'. Admin will have to fill all the fields required in the form such as UserID, UserName, UserEmail, UserPassword and UserPasswordClue as shown in File 5.5. These inputs will be stored and send to another file 'User_AddVerify.jsp' for processing.

```
<table cellpadding=4 cellspacing=2 border=0 width=520 center>
<FORM NAME=emailform METHOD=POST ACTION=User_AddVerify.jsp
onSubmit="return emailCheck(this.UserEmail.value);">

<th bgcolor="#CCCCFF" colspan=2>
<font size=5>USER REGISTRATION</font>
<br>
<font size=1><sup>*</sup> Required Fields</font>
</th>

<tr bgcolor="#c8d8f8">
<td valign=top>
<b>Name<sup>*</sup></b><br>
<input type="text" name="UserName" value="" size=20 maxlength=30></td>
<td valign=top>
<b>Login ID<sup>*</sup></b><br>
<input type="text" name="UserID" value="" size=10 maxlength=10></td>
</tr>

<tr bgcolor="#c8d8f8">
<td colspan=2>
<b>E-Mail<sup>*</sup></b><br>
<input type="text" name="UserEmail" value="" size=35 maxlength=50><br>
</td>
</tr>
```

```
<tr bgcolor="#c8d8f8">
<td valign=top>
<b>Password<sup>*</sup></b>
<input type="password" name="Password1" size=10 value="" maxlength=10><br>
<b>Confirm Password<sup>*</sup></b>
<input type="password" name="Password2" size=10 value="" maxlength=10><br>
</td>

<td valign=top>
<b>Password Clue<sup>*</sup></b><br>
<input type="text" name="PasswordClue" size=15 value="" maxlength=20>
</td><br>
</tr>

<tr bgcolor="#c8d8f8">
<td align=center colspan=2>
<input type="submit" value="Add User">
<input type="reset" value="Reset">
</td>
</tr>

</table>
```

File 5.5 : Part of the file 'User_Add.jsp'

The system will validate whether all the fields are filled in with this code

```
if (UserID.equals("") || UserName.equals("") || UserEmail.equals("") || Password1.equals("") ||
    Password2.equals("") || PasswordClue.equals("")) {
    EmptyFieldFound = 1;
}
```

Next, the system checks whether the 2 password inputs 'Password1' and 'Password2' the same with the code below. If they are the same then the next thing the system will check is whether there is a user with the same UserID in the table 'UserDetails' in the database.

```
if (Password1.equals>Password2)) {
    SamePassword = 1;
    CheckExistID = stmt.executeUpdate ("Select * from UserDetails where UserID = " + UserID
    + " ");
```

If there is no duplicate of UserID, then the system will start to send mail to the user to inform him that he is now a user of WAP-MA. If mail is sent successfully,

then the system will insert all the values into the table 'UserDetails'. A mail must be sent to the user before the details are inserted into the database. This is to avoid situation where users do not realize they were added. Send mail functions will be the same for all modules but with different mail subject and message. The code for send mail is shown in File 5.6. A class sun.net.smtpClient is imported before the function works.

```
<%@ page import= "sun.net.smtp.SmtpClient" %>
try {
sun.net.smtp.SmtpClient sm = new sun.net.smtp.SmtpClient("fsktm.um.edu.my");
    sm.from("hpyew@tm.net.my");
    sm.to(UserEmail);
//    sm.subject("Welcome To WAP-MA");
    PrintStream msg = sm.startMessage();
    msg.println("To: ");           // Note dont use + for Performance
    msg.println("Dear " + UserName );
    msg.println("Subject: You Are Now A WAP-MA User");
    msg.println("=====");

    msg.println("We the Administrators of WAP-MA are pleased ");
    msg.println("to welcome you to WAP-MA.");
    msg.println("You can login to WAP-MA system by logging into");
    msg.println("http://202.185.109.163:8080/wam/client/login.jsp");
    msg.println("on your WAP handphone.");
    msg.println();
    msg.println("Your login information is as below : ");
    msg.println("Login ID : " + UserID );
    msg.println("Password : " + Password1 );
    msg.println();
    msg.print("This mail brought to you by Admin of WAP-MA");
    sm.closeServer();
    out.println("Your Mail Has Been Sent to ");
    out.println(UserName);
    out.println();
    MailSuccessful = 1;
} //end 2nd try

catch (Exception e) {
    out.println("The mail couldn't be sent, probably because the mailhost wasnt set correctly.");
    out.println("The error message I am getting is: ");
    out.println(e.getMessage());
}
```

File 5.6 : Send Mail Function

If sending mail is successful, then all the values will be inserted into the table 'UserDetails'. On top of that, the user's list of friends must also include 'admin' to enable the administrators of WAP-MA to contact them.

```
if (MailSuccessful == 1) {  
    InsertUser = stmt.executeUpdate ("Insert into UserDetails (UserID, UserName,  
    UserEmail, UserPasswordClue, ApprovedByAdm, Deleted, DateTime )  
    values ('" + UserID + "', '" + UserName + "', '" + UserEmail + "', '" + Password1 + "',  
    '" + PasswordClue + "', 'y','y',sysdate) ");  
    InsertFriend1 = stmt.executeUpdate ("Insert into UserFriends (UserID, FriendID) values  
    ('admin', '" + UserID + "') ");  
    InsertFriend2 = stmt.executeUpdate ("Insert into UserFriends (UserID, FriendID) values ('" +  
    UserID + "', 'admin' ) ");  
} //end if
```

The Add Admin Module is exactly like Add User Module except that the table accessed is 'AdminDetails'.

- **Delete User Module**

Inside the Main Menu, a normal administrator will be able to delete user by accessing the page 'User_Delete.jsp'. Super administrator has the same function to delete the administrator by accessing 'Admin_Delete.jsp'. Admin will have a list of users to choose whom to delete. More than one user can be deleted at one time.

The page 'User_Delete.jsp' will load a list of friends that can be deleted. The administrator will choose which user or users to delete by checking the checkbox in the same row as the UserID. The codes are shown in File 5.7.

```
<TABLE border=1 cellpadding=4 cellspacing=2 width="520">
<FORM NAME=DeleteForm METHOD=POST ACTION=User_DeleteVerify.jsp>

<th bgcolor="#CCCCFF" colspan=4> Authorized User Of WAP Messaging Application </th>
<tr>
  <td><b> User ID </td>
  <td><b> User Name </td>
  <td><b> User Email </td>
  <td><b> Delete ? </td>
</tr>

<%
  Statement stmt = myConn.createStatement ();
  ResultSet rsUser1 = stmt.executeQuery ("select * from UserDetails where ApprovedByAdm =
                                     'y' and Deleted='y' ");

  if (rsUser1 != null) {
    while (rsUser1.next ()) {
      UserID = rsUser1.getString ("UserID");
      UserName = rsUser1.getString ("UserName");
      UserEmail = rsUser1.getString ("UserEmail");
    }
  }
  %>

<tr>
  <td> <%=UserID %> </td>
  <td> <%=UserName %> </td>
  <td> <%=UserEmail %> </td>
  <td><Input type=checkbox name=chkDelete value=<%= UserID %> > </td>
</tr>

<%
  } //for while
  } //for if
%>

<input type=button name=CheckAll value="Check All"
onClick="this.value=check(this.form.chkDelete)">

  <INPUT TYPE=submit name=submit value="Delete">
  <INPUT TYPE=reset name=reset value="Reset">

</FORM>
</TABLE>
```

File 5.7 : Part of the file 'User_Delete.jsp'

After checking the checkboxes, then the values will be sent to 'User_DeleteVerify.jsp' for deleting process. Once the values are taken in and

mail is sent successfully to the particular user, then deleting process will take place and the code is as below. The mail function codes are similar to the codes in the File 5.6.

```
if (MailSuccessful == 1) {  
    rowsAffected1 = stmt.executeUpdate("Delete from UserIncoming where UserID = '" +  
    chkDelete[i] + "'");  
    rowsAffected2 = stmt.executeUpdate("Delete from UserIncoming where FriendID = '" +  
    chkDelete[i] + "'");  
    rowsAffected3 = stmt.executeUpdate("Delete from UserFriends where UserID = '" +  
    chkDelete[i] + "'");  
    rowsAffected4 = stmt.executeUpdate("Delete from UserFriends where FriendID = '" +  
    chkDelete[i] + "'");  
    rowsAffected5 = stmt.executeUpdate("Delete from UserApproval where UserID = '" +  
    chkDelete[i] + "'");  
    rowsAffected6 = stmt.executeUpdate("Delete from UserApproval where NewFriendID = '" +  
    chkDelete[i] + "'");  
    rowsAffected7 = stmt.executeUpdate("Delete from UserDetails where UserID = '" +  
    chkDelete[i] + "'");  
} //end if
```

To delete a user off the database, any data related to the user in different table will have to be deleted also. First of all, any messages send by or sent out to the user have to be deleted in table 'UserIncoming', any friends of the user has to be deleted in table 'UserFriends', any requests to add this user will also be deleted in table 'UserApproval' and lastly details of the user in table 'UserDetails' will be deleted also.

5.2.2 : User Module Implementation

The codes for all User modules are written in JSP files. JSP files can integrate WML codes in it and it is dynamic by being able to retrieve information from the database. Therefore, codes are best saved in JSP files.

In the login page, user can choose to login, register or retrieve password. An unregistered user can register by filling in all the fields required in the 'Register' module. Password forgotten can also be retrieved through 'Forgot Password' module.

User can login into the system through 'Login' module. Once the user is logged into the system, he has 8 main functions to use. The 8 main functions are also referred as modules in system implementation. They are 'Modify Own Details', 'Approve Add Request', 'Query Admin', 'Read New Message', 'Write Message', 'Add A Friend', 'Delete A Friend' and 'View Message History'. Many modules are related to each other in User Module and will be illustrated in a form link diagram as shown in Figure 5.3.

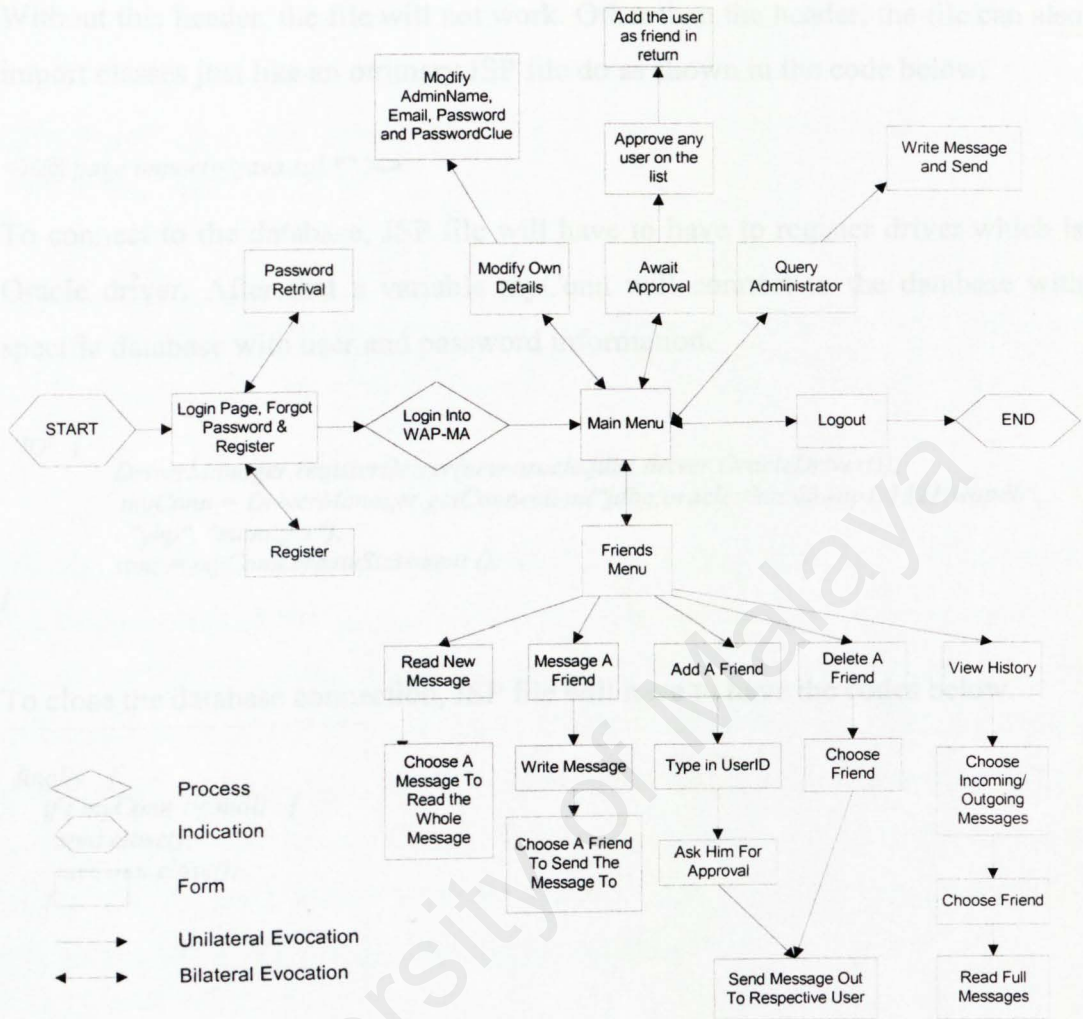


Figure 5.3 : Form Links for User Module

In any JSP file integrating WML codes, the header of the file is very important and different from just plain JSP file integrating HTML codes. The header should be :

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<%@ page language="java" contentType="text/vnd.wap.wml" %>
  
```

Without this header, the file will not work. Other than the header, the file can also import classes just like an ordinary JSP file do as shown in the code below.

```
<%@ page import="java.sql.*" %>
```

To connect to the database, JSP file will have to have to register driver which is Oracle driver. After that a variable myConn will connect to the database with specific database with user and password information.

```
try {  
    DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());  
    myConn = DriverManager.getConnection("jdbc:oracle:thin:@wap1:1521:wapdb",  
        "yhp", "manager");  
    stmt = myConn.createStatement ();  
}
```

To close the database connection, JSP file will have to have the codes below.

```
finally {  
    if ( myConn != null) {  
        stmt.close();  
        myConn.close();  
    }  
}
```

• **Modify Own Details Module**

‘Modify Own Details’ module basically deals with modifying values of user details inside the table ‘UserDetails’. User inputs are stored in variables. The variables and how the values are retrieved from the previous page is shown in the codes below.

```
UserID = request.getParameter("UserID");  
UserName = request.getParameter ("UserName");  
UserEmail = request.getParameter("UserEmail");  
Password1 = request.getParameter("Password1");  
Password2 = request.getParameter("Password2");  
PasswordClue = request.getParameter ("PasswordClue");
```

First of all, the two passwords are checked whether they are the same because they must be the same before changes are made. The code `if(Password1.equals>Password2))` is true then the SQL statement inside the if block will be executed.

```
if (Password1.equals>Password2)) {
    UpdateUser = stmt.executeUpdate ("Update UserDetails set UserName=" + UserName
    + "", userEmail=" + userEmail + "", UserPassword= " + Password1 + "",
    UserPasswordClue = " + PasswordClue + "" where UserID=" + UserID + "" ");
} //end if
```

Finally, a message will be displayed informing the user whether his modification has been successful or otherwise. There are three cards in the file to indicate there are three types of messages that can appear in the screen and each message has different course of actions to be taken. If the modification fails, then the system will lead the user back to 'User_Modify.jsp' file. If modification is successful, the system will include the file 'User_Link1.jsp' where users will have other links to go to after the modification module. The file 'User_Link1.jsp' can be included using this code `<%@ include file="User_Link1.jsp" %>`.

```
<wml>

<%
    if (UpdateUser == 0 && !(Password1.equals>Password2)) { %>

<card id="Redirect1" title="Wrong Input" ontimer="User_Modify.jsp" >
<timer value="50" />
<p align="center">
    <br />
    Password Incompetency !
</p>
<do type="Accept" label="Retry">
    <go href="User_Modify.jsp"/>
</do>

</card>
<% } //end if
```

```

else if (UpdateUser == 0) { %>

<card id="Redirect2" title="Wrong Input" ontimer="User_Modify.jsp" >
<timer value="50" />
<p align="center">
    Update Failed !
</p>
<do type="Accept" label="Retry">
    <go href="User_Modify.jsp"/>
</do>

</card>
<% } //end else if

else { %>

<card id="Redirect3" title="Update Successful" >
<p align="center">
    Update Successful !
</p>

<%@ include file = "User_Link1.jsp" %>
</card>

<% } // end else
    
```

The file User_Link1.jsp is shown in File 5.8 and is always included in most modules because after successfully finishing one module, the user can then choose which module he wants to go next.

```

<do type="Accept" name="MainMenu" label="Main Menu">
<go href="User_MainMenu.jsp#Menu1">
    <postfield name="UserID" value="$UserID" />
</go>
</do>

<do type="Accept" name="FriendsMenu" label="Friends Menu">
<go href="User_FriendsMenu.jsp#Menu2">
    <postfield name="UserID" value="$UserID" />
</go>
</do>

<do type="Accept" name="Logout" label="Logout">
    <go href="User_Login.jsp"/>
</do>
    
```

File 5.8 : 'User_Link1.jsp' file

- **Read New Message & Write Message Module**

The 'Read New Message' module is basically related to 'Write Message' module because after reading message, user has an option either to reply the message which the codes are similar to the codes in 'Write Message' module or do nothing about the message after reading it. First of all, the users will have a screen showing all the new messages. The file 'User_InMsg.jsp' will extract all the new messages from the database using the code below. A resultset variable called rs is used to store all the results from the SQL statement.

```
TotalNewMsg = stmt.executeUpdate ("Select * from UserIncoming where UserID = '" + UserID
                                   + "' AND NewMsg='y' ");
rs = stmt.executeQuery ("Select * from UserIncoming where UserID = '" + UserID + "' AND
                        NewMsg='y' ");

if (rs != null) {
    while (rs.next()) {
        FriendID = rs.getString ("FriendID");
        Msg = rs.getString ("Msg");
        TempDateTime = rs.getString("DateTime");
        DateTimeLength = TempDateTime.length();
        DateTime = TempDateTime.substring(0,DateTimeLength-2);
    }
}
```

Later on, the message will be displayed according to DateTime first followed the sender ID. User can click on the link to read the full message. The code for this function is written below.

```
<anchor><%= DateTime + " " + FriendID%><br/><go href="User_InMsg2.jsp">
    <postfield name="UserID" value="<%= UserID %>" />
    <postfield name="FriendID" value="<%= FriendID %>" />
    <postfield name="Msg" value="<%= Msg %>" />
    <postfield name="DateTime" value="<%= DateTime %>" />
</go>
</anchor>
```

When the message link is clicked to read the full message, the message will be updated from being a new message (NewMsg='y') to not new message (NewMsg='n'). NewMsg is a field in table 'UserIncoming'. The SQL statement below will update the message.

```
ReadMsg = stmt.executeUpdate("Update UserIncoming set NewMsg='n' where UserID='" +
UserID + "' and FriendID='" + FriendID + "' AND Msg='" + Msg + "' AND DateTime = to_date
('" + DateTime + "', 'yyyy-mm-dd hh24:mi:ss') ");
```

After reading the message, user has an option to reply to the message. The coding to reply is shown below and this is extracted from the file 'User_Reply.jsp'.

```
<wml>
  <card id="Card1" title="Write Message" >

    <p>Message : </p>
    <input name="Msg" type="text" maxlength="100"/>

    <do type="Accept" name="SendMsg" label="Send Message">
      <go href="User_OutMsg4.jsp ">
        <postfield name="UserID" value="$UserID" />
        <postfield name="FriendID" value="$FriendID" />
        <postfield name="Msg" value="$Msg" />
      </go>
    </do>

    <do type="Accept" name="MainMenu" label="Main Menu">
      <go href="User_LoginVerify.jsp#Menu1 ">
        <postfield name="UserID" value="$UserID" />
      </go>
    </do>

    <do type="Accept" name="FriendsMenu" label="Friends Menu">
      <go href="User_FriendsMenu.jsp#Menu2 ">
        <postfield name="UserID" value="$UserID" />
      </go>
    </do>

  </card>
```

File 5.9 : Part of the file 'User_Reply.jsp'

In this File 5.9, after writing the message and when wanting to send, <do name="SendMsg" is called and the values will be posted to the file 'User_OutMsg4.jsp'. This particular file is linked to 'Write Message' module. Therefore, there's a proven linkage between this two modules. Other than that, user can also go back to MainMenu or FriendsMenu without writing the message.

This is because, the user might have come into here accidentally and so the system must provide a way out for the user.

- **Approval & Add A Friend Module**

Other than 'Read New Message' module related to 'Write Message' module, 'Add A Friend' module is also related to 'Approval' module. One user has to add and ask for approval from another user to become his friend. The other user is only his friend once he approves. The second user who has just given his approval can ask to add the first user to become his friend in return. Therefore, 'Approval' module will be discussed in details to show how to approve the request and add that particular user in return works.

First the user will click into the 'Approval' module. In the screen, the user can see a list of users wanting to add him and waiting for his approval. The code below shows how the ResultSet `rsNewFriendID` stores UserID of the friends wanting to add him. All the friends will be showed on the screen because they are store inside the array called `ArrayNewFriendID[]`.

```
rsNewFriendID = stmt.executeQuery ("Select UserID from UserApproval where NewFriendID  
= " + UserID + " ");
```

```
if (rsNewFriendID != null) {  
    while (rsNewFriendID.next () ) {
```

```
        NewFriendID = rsNewFriendID.getString ("UserID");  
        ArrayNewFriendID[i] = NewFriendID;  
        i = i + 1;
```

```
    } //end while  
} //end if
```

Only one user can be approves at one time. If the user approves a particular user, then the four SQL statements below will be executed.

```
RowsAffected = stmt.executeUpdate("Delete from UserApproval where userID = '" +
    NewFriendID + "' and NewFriendID = '" + UserID + "' and Authorize='n' ");
RowsInserted = stmt.executeUpdate("Insert into UserFriends (UserID, FriendID) values ('" +
    NewFriendID + "','" + UserID + "') ");
RowsFriendExist = stmt.executeUpdate("Select * from UserFriends where UserID= '" + UserID
    + "' and FriendID = '" + NewFriendID + "' ");
RowsFriendApproval = stmt.executeUpdate("Select * from UserApproval where UserID= '" +
    UserID + "' and NewFriendID = '" + NewFriendID + "' and Authorize = 'n' ");
```

The first statement is to delete this approval from the 'UserApproval' table because his request is already approved. The second statement is to insert a statement into 'UserFriends' to indicate that the logged in user is already a friend of the user asking for approval. The third and fourth SQL statements check whether this user whose request has just been approved is in fact a friend of the logged in user.

```
<% if (RowsAffected == 1 && RowsInserted == 1 ) { %>
    <p align="center"> <%= NewFriendID %> is authorized.<br/><br/>

    <% if (RowsFriendExist == 0 && RowsFriendApproval == 0 ) { %>
        Click

        <anchor> here <br/><go href="User_AddFriend3.jsp">
            <postfield name="UserID" value="<%= UserID %>" />
            <postfield name="NewFriendID" value="<%= NewFriendID %>" />
        </go>
        </anchor>

        if you want to add <%= NewFriendID %> in return
        <% } //end if %>
    </p>
    <% } //end if
```

If he is already a friend, the code "<%= NewFriendID %> is authorized" will be executed and this message will appear on the screen. If he is not a friend, then the variables RowsFriendExist and RowsFriendApproval are both 0. This will execute the codes in between the anchor block. The file 'User_AddFriend3.jsp' will be called. This file belongs to module 'Add A Friend'.

In 'User_AddFriend3.jsp' file, the SQL statement will be executed and this will insert a new row in the table 'User_Approval' stating that the logged in user now wishes to add this user to become his friend.

```
RowsAffected = stmt.executeUpdate("Insert into UserApproval (UserID, NewFriendID,
Authorize) values ('" + UserID + "','" + NewFriendID + "','" + n' )");
```

A message will be returned on the screen informing the user about the result of executing the above SQL statement.

```
<% if (RowsAffected == 1) { %>
  <p align="center"> User added and waiting for authorization from <%= NewFriendID
%></p>
<% } //end if
else { %>
  <p align="center"> Fail to add existing user ! </p>
<% } %>
```

If the variable RowsAffected equals to 1, then the user is added successfully and is now waiting for his approval and if it's not successful then message "Fail to add existing user" will appear.

CHAPTER 6 : SYSTEM TESTING

System testing is executed after system implementation. Whatever that is implemented in the system should be tested thoroughly to ensure it's a reliable, efficient yet effective system. In short, a system that does what it is supposed to do is consider a good system.

Therefore, all pages must go through unit testing which is the simplest test of all followed by the module testing. Then, all modules are integrated and integration testing is performed. Finally, system testing is done on the entire system to ensure it works as a whole.

6.1 : UNIT TESTING

Unit testing will be done on both Administrator module and User Module.

6.1.1 : Administrator Module Testing

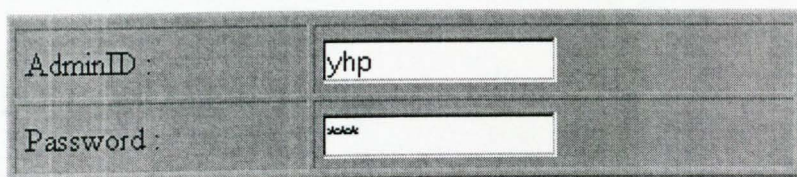
In Administrator module, unit testing is usually done on a function or on the whole file.

- **Login Function**

In the login file 'Admin_Login.jsp', there are two input fields waiting for admin to input data. The two input fields name are 'AdminID' and 'AdminPassword'.

```
<input name="AdminID" size=15 maxlength=10>  
<input name="AdminPassword" type="password" size="15" maxlength=10>
```

This test will tests out whether the data input by the admin in Figure 6.1 will actually be what the system is retrieving. Te system will declare two strings to store the input values. The variables are named 'AdminID' and 'AdminPassword'. If the system retrieves correctly then the string AdminID must be storing 'yhp' and AdminPassword must be storing '123'.



A screenshot of a login form with two input fields. The first field is labeled 'AdminID :' and contains the text 'yhp'. The second field is labeled 'Password :' and contains the text '123'.

Figure 6.1 : Login box for Login Function

• Delete Function

This file lets the administrator to choose more than one user to delete from the system as shown in Figure 6.2.

Authorized User Of WAP Messaging Application			
User ID	User Name	User Email	Delete ?
emily	Yong Mei Ling	emilyyong@maxis.net.my	<input type="checkbox"/>
yhp	Yew Hooi Phaik	hpyew@tm.net.my	<input type="checkbox"/>
guan	Lee Kok Guan	guan@hotmail.com	<input checked="" type="checkbox"/>
james	Tan Kee Meng	jamestan@yahoo.com	<input checked="" type="checkbox"/>

Figure 6.2 : Delete table for Delete User Function

In this case, two users were chosen and therefore the string array variable called `chkDelete` in 'User_DeleteVerify.jsp' which is the subsequent file called by 'User_Delete.jsp' file should store the 2 UserID values. It means the `chkDelete` should be storing ['guan','james']. If `chkDelete[]` stores correctly, then the unit testing is successful.

6.1.2 : User Module Testing

• User Query Admin Function

In this file, the user will write a message and post it to the admin. To run unit test on it, the message written by the user must exactly be the message stored in the variable in the User Query Admin Function. The screen is shown in Figure 6.3. The string 'What is the maximum number of users in this system?' should be stored in variable `Msg`.

```
<input name="Msg" type="text" maxlength="100" />
```

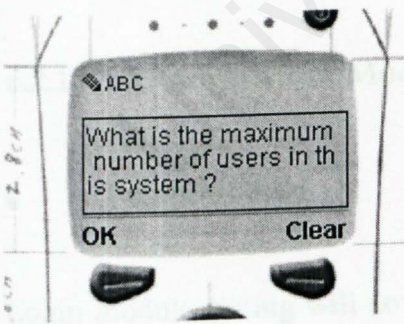


Figure 6.3 : Screen for User Query Admin Function

Other than checking whether the text is the same as being input, length must also be validated so that the when trying to insert into the database, no error will occur.

The string is less than 100 characters, therefore the database will accept. In this test, the text box will not accept. Therefore, the message is only half written as shown in Figure 6.4.

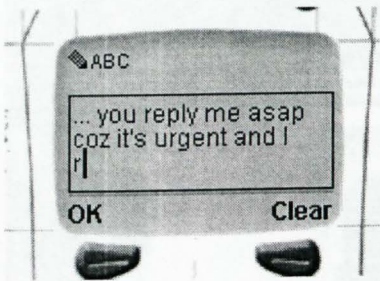


Figure 6.4 : Error Screen because of maximum characters

6.2 : MODULE TESTING

Module testing covers a wider scope of testing than unit testing. It will take a module and test out from the beginning of the module until the end of it and the result or output will be compared with the expected result.

6.2.1 : Administrator Module Testing

- **Login Module**

Login module testing will cover tests from the beginning when the user starts to input his LoginID and Password to verification of the LoginID and Password entered. If the login is a valid one, then the values will be stored into two session objects and finally the login module will direct the users to other modules of the system. Otherwise, an error message will appear stating what error has occurred.

Figure 6.5 shows the login page for the administrator where administrator can fill in the text boxes and click on the 'submit' button.

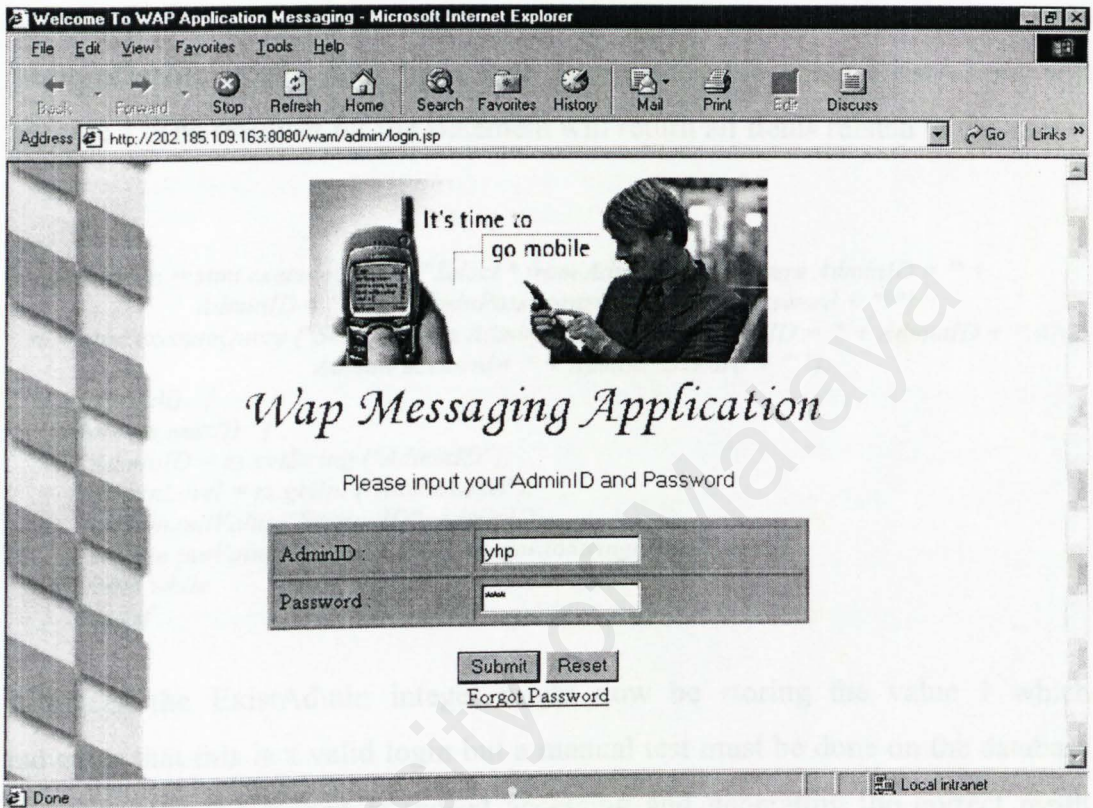


Figure 6.5 : Login Page

The system will store the two values inside two variables called 'AdminID' and 'AdminPassword'. The first test is to check whether the two variables are storing the correct values as input by the admin. The AdminID and AdminPassword variables should be storing the values 'yhp' and '123' as shown by the code below. If it's correct, then the second test is performed.

```
try {
    AdminID = request.getParameter ("AdminID");
    AdminPassword = request.getParameter ("AdminPassword");
}
```

The second test is to extract the data from the database to be compared with the values being input to validate whether it is the correct admin or otherwise. First of all, the SQL statement as shown below should be "Select * from AdminDetails where AdminID = 'yhp' And AdminPassword= '123' ". If the table 'AdminDetails' really does have such admin then the value ExistAdmin will become 1. The resultset for the statement will return all fields related to the user.

```
try {  
    ExistAdmin = stmt.executeUpdate("Select * from AdminDetails where AdminID = '" +  
        AdminID + "' And AdminPassword= '" + AdminPassword + "'");  
    rs = stmt.executeQuery ("Select * from AdminDetails where AdminID = '" + AdminID + "' AND  
        AdminPassword= '" + AdminPassword + "'");  
    if (rs != null) {  
        while (rs.next()) {  
            AdminID = rs.getString ("AdminID");  
            AdminLevel = rs.getInt ("AdminLevel");  
            session.putValue("SessionID", AdminID);  
            session.putValue("SessionLevel", Integer.toString(AdminLevel));  
        } //end while  
    } //end if  
}
```

Although the ExistAdmin integer might now be storing the value 1 which indicates that this is a valid login but a manual test must be done on the database to verify that the system is indeed accessing and generating the correct result when it accesses the database

After that, the values AdminID and AdminLevel are retrieved from the resultset and later on store in SessionID and SessionLevel consecutively.

The codes show what errors message to prompt out if the situation is such. If empty field is found, the error "Empty Field Found. Please [click here](#) to try login again.

```
if (EmptyFieldFound == 1) { %>
  <h3>Empty Field Found. Please
  <a href="Admin_Login.jsp">click here</a>
  to try login again.</h3>
<% }
```

If no empty field found and this admin does not exist, the error “AdminID or Password can be wrong. Please click here to try login again.

```
else if (EmptyFieldFound == 0 && ExistAdmin == 0) { %>
  <h3>AdminID or Password can be wrong.Please
  <a href="Admin_Login.jsp">click here</a>
  to try login again.</h3>
<% }
```

Therefore, the test will test whether the correct error message is prompt out at a specific error purposely generated by the admin. If empty fields are purposely left and ‘submit’ button is clicked, then the message “Empty Field Found.....” should be displayed. If the fields are filled and the admin is an invalid admin, therefore the message “AdminID or Password can be wrong.....” will be prompted.

The last test on module is to test whether the correct level of administrator is accessing the correct menu. Super Administrator should be accessing a different main menu where he can administer normal administrators as shown in Figure 6.6 while normal administrator’s main menu should enable him to administer users as shown Figure 6.7.

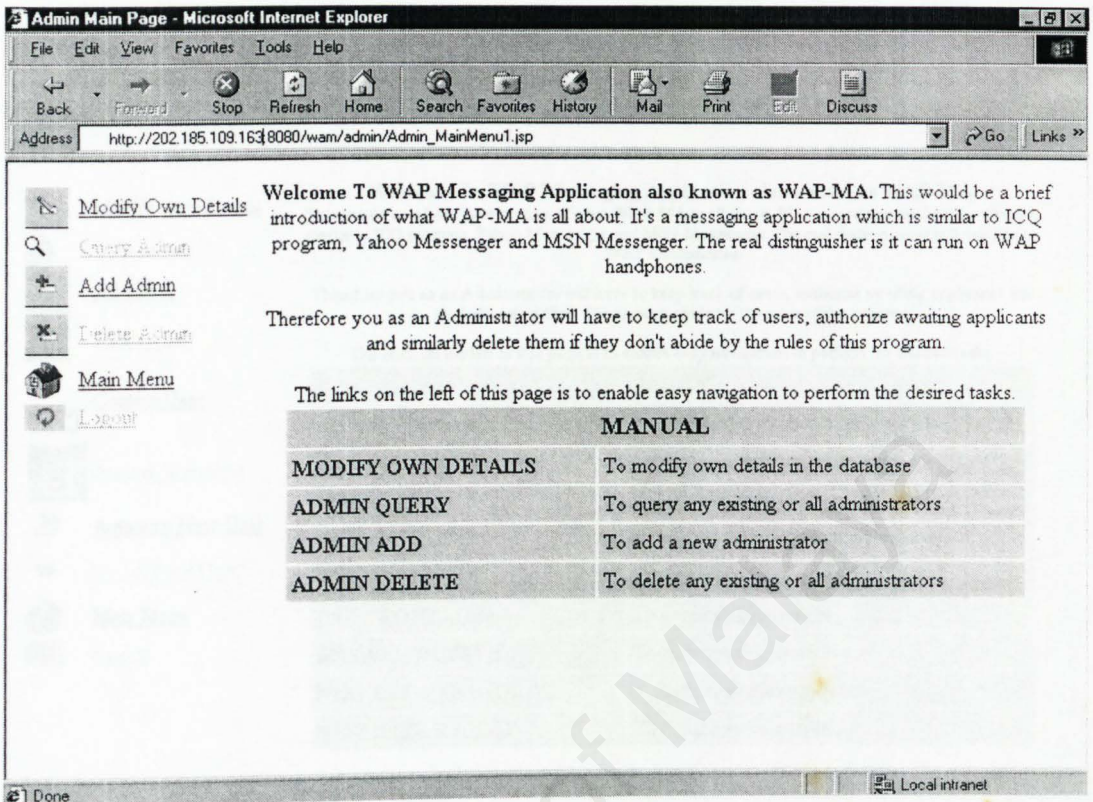


Figure 6.6 : Super Administrator Main Menu Page

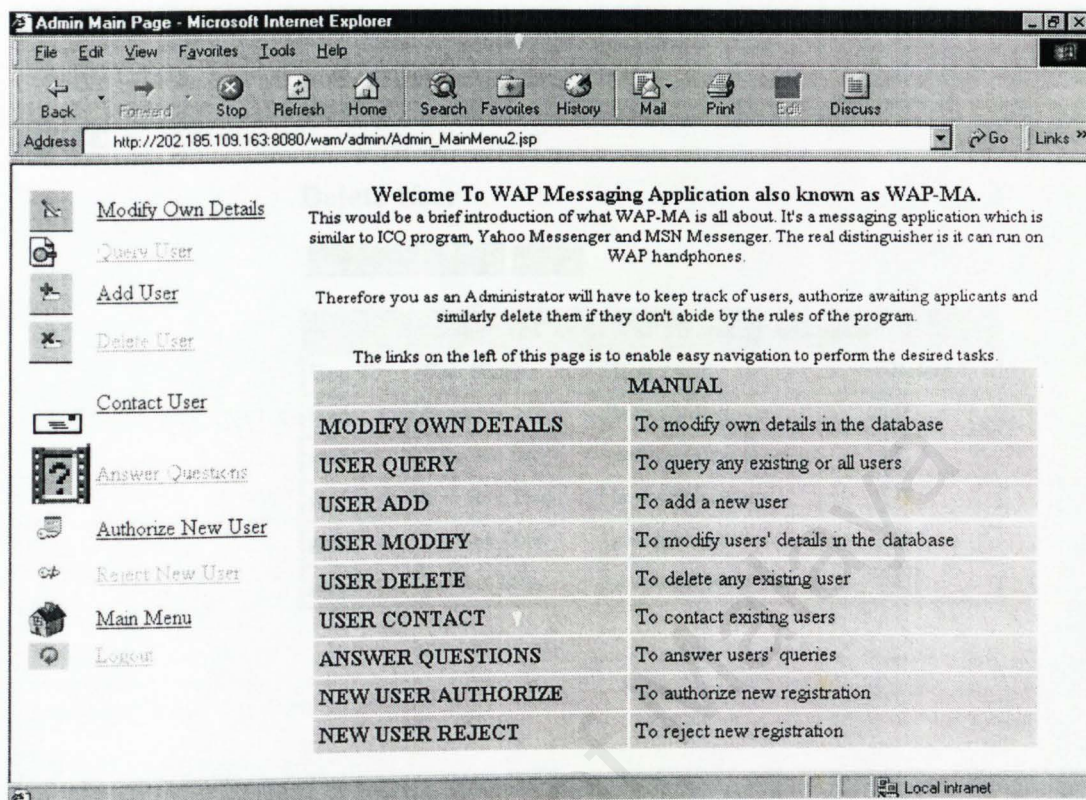


Figure 6.7 : Normal Administrator Main Menu Page

• Delete User Module

Delete User module testing comprises of testing administrator's choice of users to be deleted is the same as the users being deleted until the users are deleted and mail is sent to them successfully. First of all, administrator will check on the names to be deleted as shown in Figure 6.8. In this case, the users with the UserID 'bruce' and 'james'.

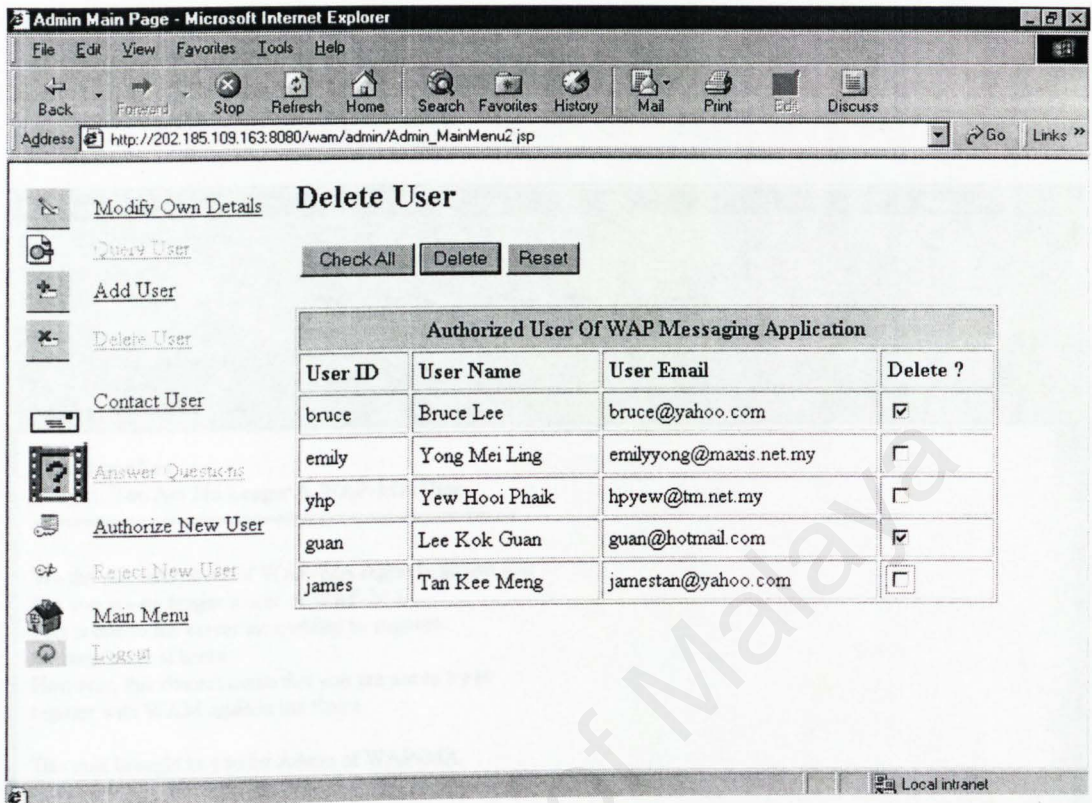


Figure 6.8 : Delete User Page

When the users are deleted, mails will be sent to them. The next test will test whether the users receive their e-mails, check the table 'AdminDetails' manually to verify that the user's information is deleted and finally to test whether they can still login to the system using their 'LoginID' and 'Password'.

The user being deleted will receive an email informing him about it as shown in Figure 6.9. Only if the mail is sent successfully, then only the database is accessed to delete the user from the table 'UserDetails'. The code to delete user is shown below.

```
if (MailSuccessful == 1) {
    rowsAffected = stmt.executeUpdate("Delete from UserDetails where UserID = '" +
                                     chkDelete[i] + "'");
}
```

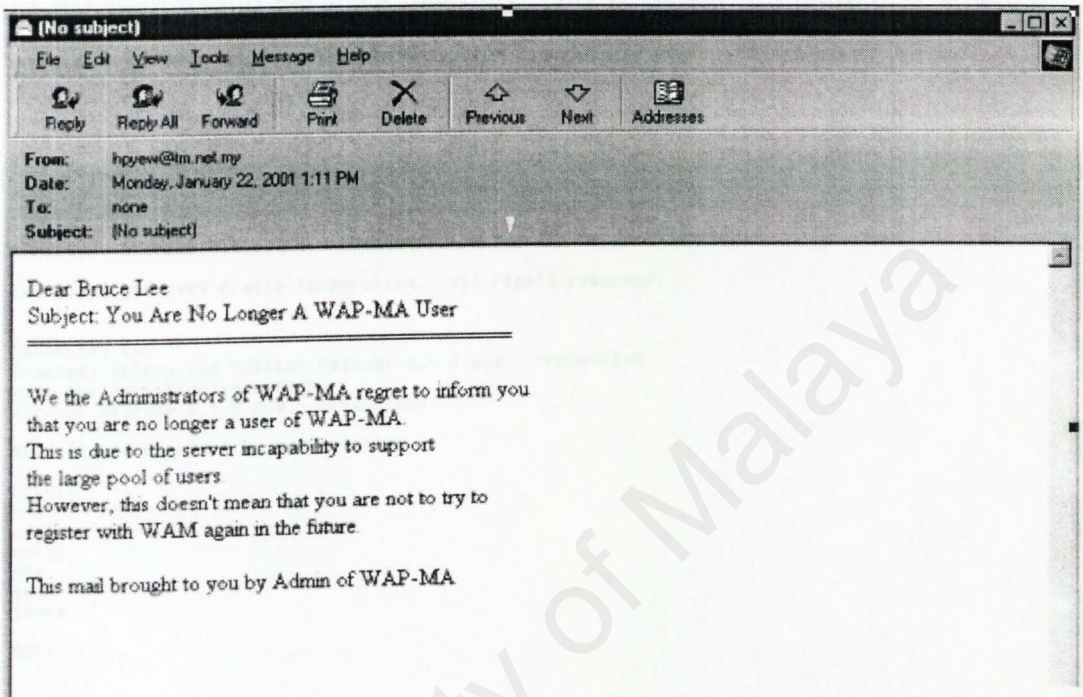


Figure 6.9 : Delete User Mail

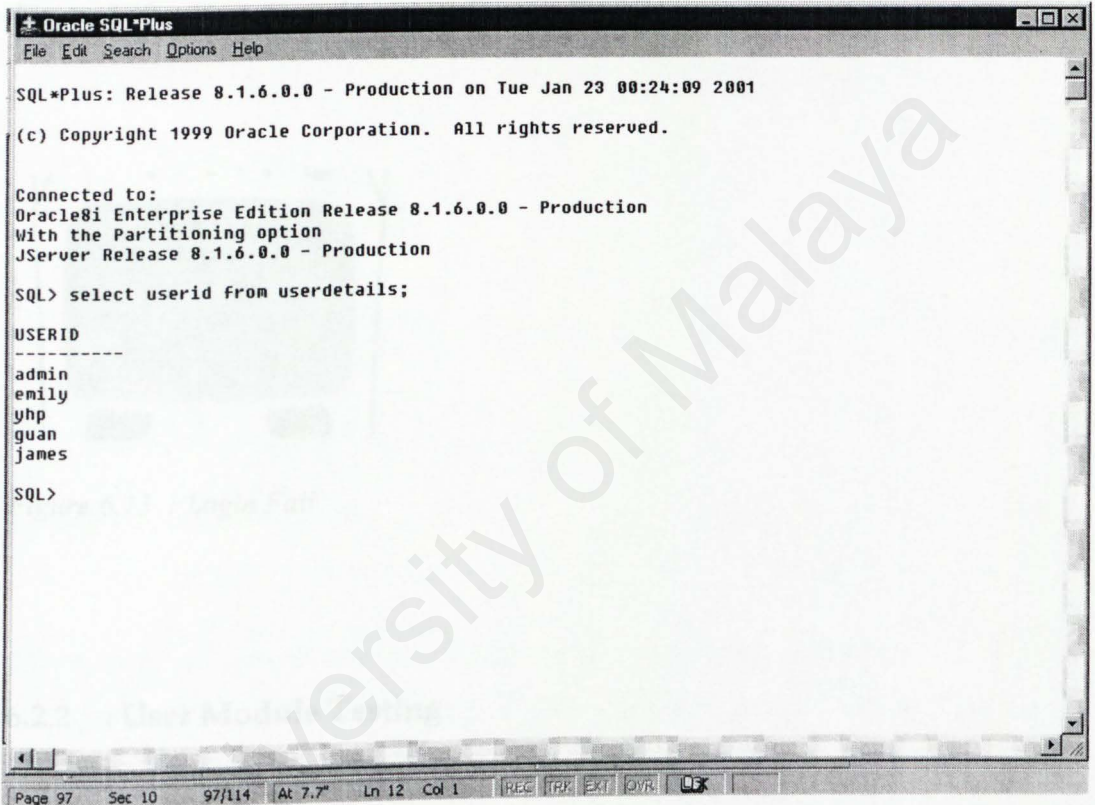
Otherwise if user deletion fails, a error message will be displayed in the deletion page as shown in Figure 6.10.

User Deletion Summary

The mail couldn't be sent, probably because the mailhost wasnt set correctly.
The error message I am getting is: mailhost

Figure 6.10 : Delete User Fail Message

Accessing the database manually is by logging into the SQLPlus application. To verify whether the user has been deleted or otherwise, the statement is “Select UserID From UserDetails”. If the user with the UserID ‘bruce’ has been deleted, the database would not have that data. This is shown in Figure 6.11 where user ‘bruce’ has been deleted and therefore his details are not present anymore.



```

Oracle SQL*Plus
File Edit Search Options Help

SQL*Plus: Release 8.1.6.0.0 - Production on Tue Jan 23 00:24:09 2001
(c) Copyright 1999 Oracle Corporation. All rights reserved.

Connected to:
Oracle8i Enterprise Edition Release 8.1.6.0.0 - Production
With the Partitioning option
JServer Release 8.1.6.0.0 - Production

SQL> select userid from userDetails;

USERID
-----
admin
emily
yhp
guan
james

SQL>
    
```

Figure 6.11 : Accessing SQLPlus to verify the deletion.

Lastly, the user login page of WAP-MA is shown in Figure 6.12 and the error message that is displayed because ‘bruce’ is not a valid user anymore will be shown in Figure 6.13.

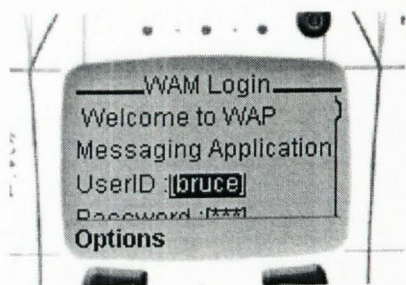


Figure 6.12 : User 'bruce' trying to log into WAP-MA

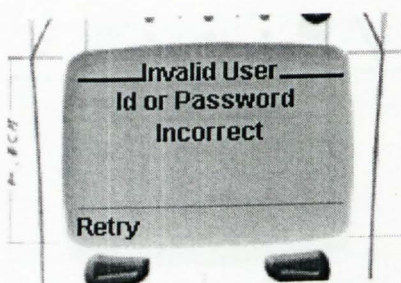


Figure 6.13 : Login Fail

6.2.2 : User Module Testing

• Write Message Module

The write message module will at first allows the user to write a message, choose intended recipients which are his friends and send the message out. The test will cover that the message sent is the original message written and sent out to correct recipients. At first the user will go into FriendsMenu as shown in Figure 6.14 and write a message as shown in Figure 6.15.

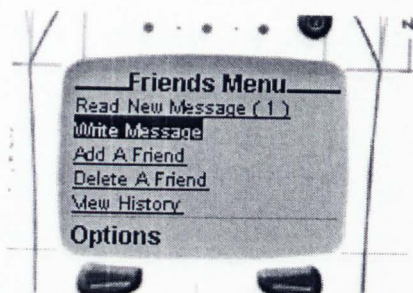


Figure 6.14 : Friends Menu

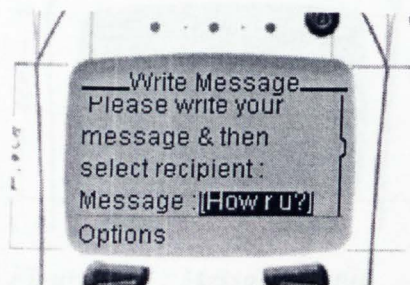


Figure 6.15 : Write Message

The system will check whether the message value stored is 'How r u ?' to make sure the sent message is the same as the original message written. This can be checked later on in the database manually after the message is sent. The next step is to choose recipients of the message which is shown in Figure 6.16. Therefore, the message will be sent to 'emily' because the name was marked in the screen to indicate that emily is the intended recipient. If message is sent successfully, the message "Message Sent Successfully" will be displayed in the next screen as in Figure 6.17.

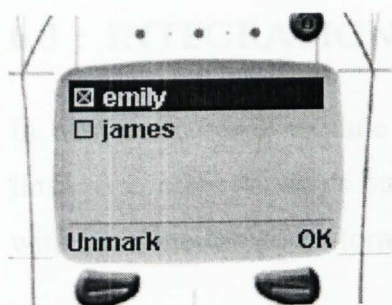


Figure 6.16 : Choose Recipients

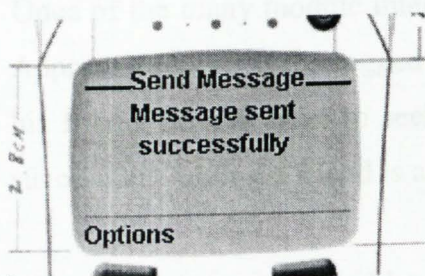


Figure 6.17 : Message Status

The test performed is executing SQL statements in SQLPlus to verify whether the message has been sent and also to make sure the original message is sent to intended recipient.

6.3 : INTEGRATION TESTING

In a system, some module might work as an individual module without any linkage to other modules but there are some modules where they must integrate with other module to perform its function.

6.3.1 : AddFriend Module & ApproveRequest Module

Once of the many module integration in the system is the module AddFriend and ApproveRequest. If the logged in user would want to add another user to become his friend, he will have to seek for the other party's approval before addition is successful. Adding a friend is a prior step to messaging each other.

When the user needs to add a friend, he will go into the Add Friend module as shown in Figure 6.18. In this case, the user to be added is called 'guan'. When the user clicks on the link 'Search User', the system will access the database to determine whether the particular user exist. If the user exists, then the screen will be like Figure 6.19 meaning the system is seeking approval from 'guan'.

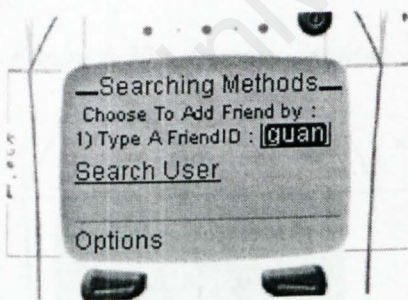


Figure 6.18 : Adding Friend



Figure 6.19 : Adding Friend Status

The next test is to test whether when the user 'quan' logs into the system, the system will seek approval from him or not. User 'quan' will have to access the ApproveRequest module as shown in Figure 6.20. He will be able to see that user 'yhp' is seeking his approval to add him as friend. If he approves the request, then Figure 6.21 will appear.

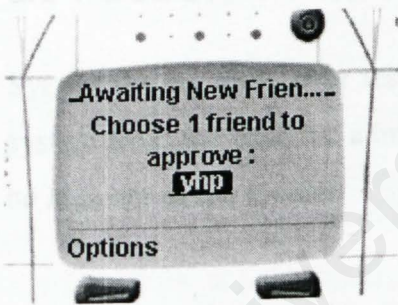


Figure 6.20 : Approve Request



Figure 6.21 : Approve Request Status

If Figure 6.21, user 'guan' is given a choice whether he wants to add user 'yhp' in return because the system has determined that the user 'yhp' is not a friend of 'guan'. Therefore, user 'guan' can add 'yhp' and seeks for her approval. If he clicks on the link, then system will seek approval from 'yhp' as shown in Figure 6.22.

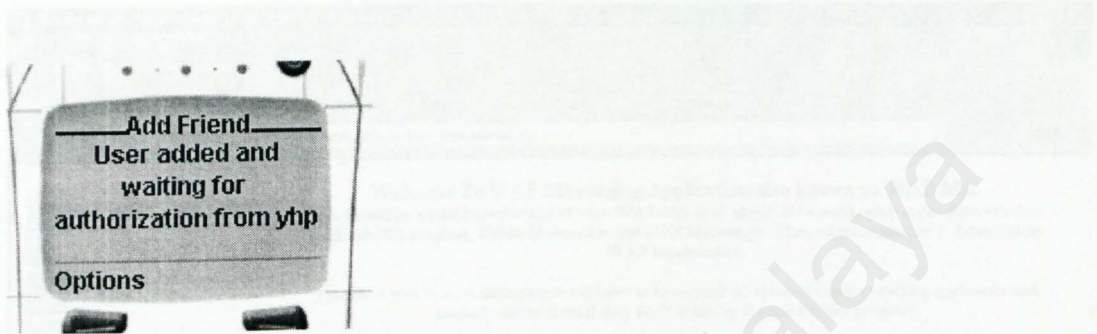


Figure 6.22 : Adding the Requestor in return

6.4 : SYSTEM TESTING

System testing will cover testing for the whole WAP-MA system. It will test the system from the time the admin or an user is log in, the modules he accesses until he logs out of the system.

6.4.1 : Administrator Module Testing

In a system, it is very important that the user can understand the system by just one look on the interface or the layout of the system. Therefore, WAP-MA is specially designed to cater for understanding needs. It is a rather simple system with just one main page called MainMenu as shown in Figure 6.23. MainMenu has just one link lists on the left of the page and is rather easy for the admin to know what is there in the system with just one look.

When the admin logs into the system, he will expect to see all the links workable leading him to somewhere. There should be no link that is not working. Basically, it means all the links of the left of the MainMenu as shown in Figure 6.23 should work.

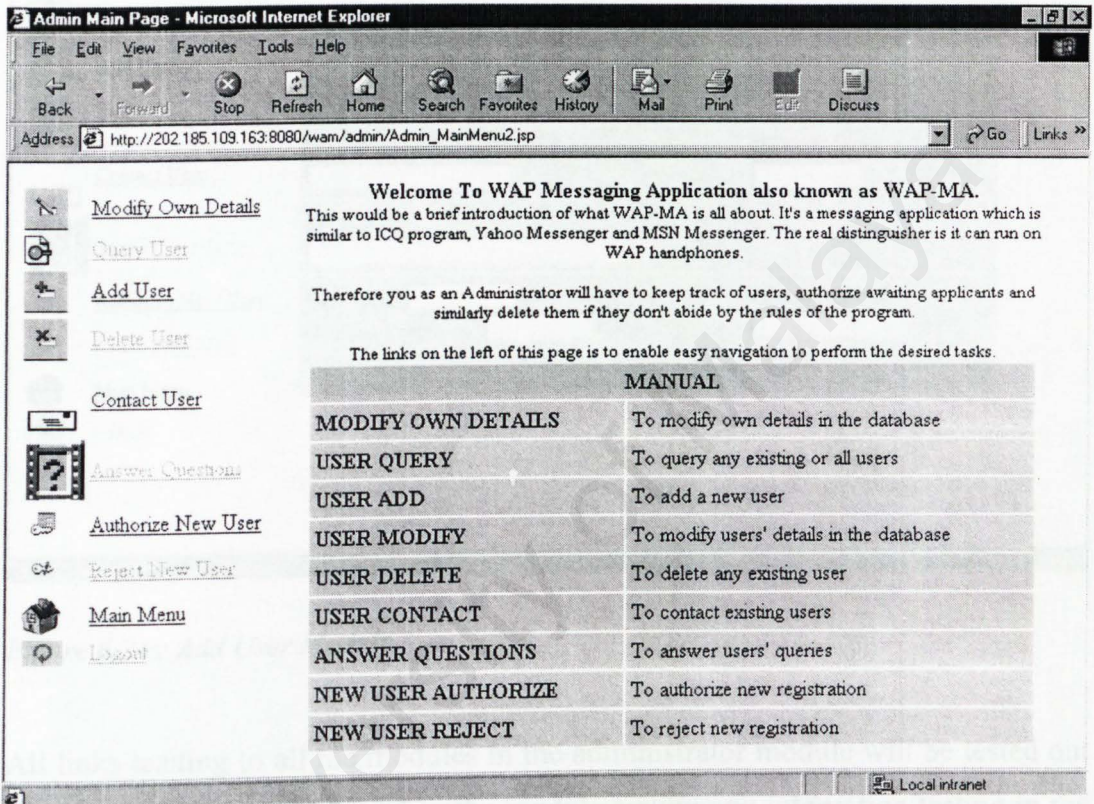
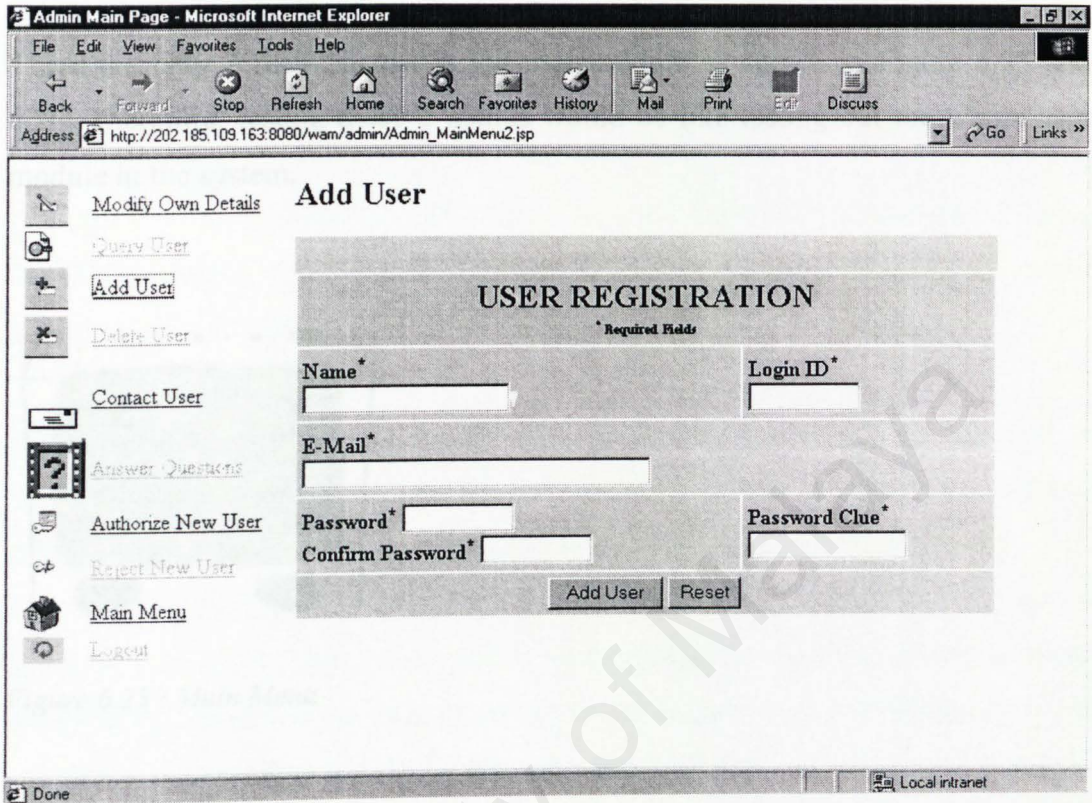


Figure 6.23 : MainMenu of Administrator Module

Another important point is the name of the links should not be misleading. For example, if the link is to enable the admin to add a new user, therefore the name of the link should be 'Add User' and not just 'Add'. The link should bring the admin to the correct page as described by the name of the link. If the link is to add a user, therefore the admin will expect to see a page loads to enable him to add a new user and not something else. The admin will expect to see something like the screen in Figure 6.24.



Admin Main Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss

Address http://202.185.109.163:8080/wam/admin/Admin_MainMenu2.jsp Go Links »

Add User

USER REGISTRATION

*Required Fields

Name* Login ID*

E-Mail*

Password* Password Clue*

Confirm Password*

Add User Reset

Done Local intranet

Figure 6.24 : Add User Module

All links leading to all the modules in the administrator module will be tested out and this test has been covered in the module testing and integration testing in the previous sections.

6.4.2 : User Module Testing

There are only 2 main menus in the user module as shown in Figure 6.25 and 6.26. Therefore, testing these 2 menus would be like testing out the whole user module in the system.

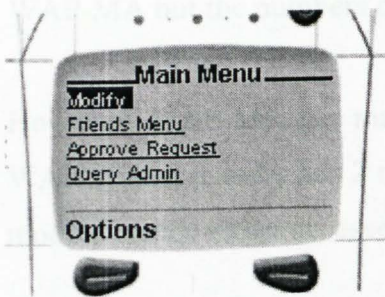


Figure 6.25 : Main Menu

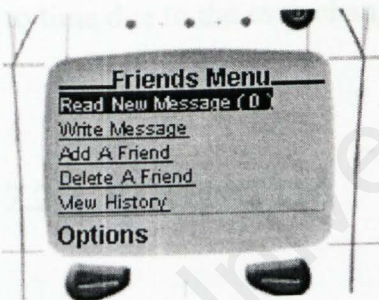


Figure 6.26 : Friends Menu

The main test is to make sure all the links in the 2 menus above works and leads the users into the correct module. Functions inside the module and how it works are well covered in module testing. How 2 or more modules integrate with each other has also been covered in integration testing.

CHAPTER 7 : CONCLUSION

7.1 : EVALUATION

Evaluation of WAP-MA is a vital process in order to become competitive when it is ready to penetrate into the market of WAP. There is no similar WAP system to WAP-MA but the numbers of WAP systems available are growing rapidly.

However, WAP-MA has many similarities with many existing WAP systems. A WAP system usually has 2 modules; for administrator and user. An administrator module can be either accessed through internet browser or WAP handphone.

WAP-MA being the sole WAP system with messaging capability does possess competitive advantage but it is bound to be taken over someday if solutions are not found to overcome its limitation. The system should be upgraded from time to time due to the ever-changing technology of WAP.

7.2 : STRENGTH

7.2.1 : Platform Independent

One of the many reasons JSP was chosen is to give flexibility to the system to be able to run in any operating system or platform and be accessed through any internet browser in any platform. Accessibility of the system is a main concern therefore it must be platform independent to realize wide accessibility as outlined in 7.2.2.

7.2.2 : Wide Accessibility

The main objective of WAP-MA is interaction. Therefore, it has to be deployed and be accessible through the internet so that administrators and users can access the system easily. The requirement for administrators to access WAP-MA is an internet browser installed inside their computers. WAP-MA works well with any browsers because of Java Server Pages language is platform and browser independent. Users will need a WAP enabled handphone to access the system.

7.2.3 : Interoperability

WAP-MA is flexible when it comes to determining where the web server, database server and mail server should be installed. Web server, database server and mail server can trio be installed in three different computers and it will still work. The system is built upon multi-tier architecture theory where it should still be operable although three of it's servers are installed separately. This is to overcome unseen limitation where one computer is only capable to support one server.

7.2.4 : Coding Reusability And Compatibility

Java Server Pages integrating Java technology recognizes codes reusability. Subsequently, JSP can import existing Java classes and use it and this has been practiced in WAP-MA. Besides that, JSP also recognizes include function where one file's codes can be re-included into another file. This eliminates repetitive codes and increase performance. WML language also works well when integrated into JSP files and no conflict arises. Therefore, JSP is compatible with WML to develop a stable system.

7.2.5 : Confidentiality

Confidentiality of WAP-MA is a major concern to ensure authorized personnel logging into the system and avoid intrusion. The password is only revealed in the Forgot Password link in the login page when all fields asked are filled in correctly.

A session object is created when the administrator logs into the system. This ensures the correct level of administrator is administering the correct tasks. For example, super administrator who is level 1 is administering all normal administrators of WAP-MA and normal administrator is level 2 and will only be administering users of WAP-MA.

7.2.6 : Expansion And Scalability

WAP-MA is a new system and realizes there might be a need for expansion in the future. Therefore WAP-MA needs to be highly scalable and it has been proven that the database and webserver that are currently installed together in the same computer can be separated and installed inside different computers.

7.3 : LIMITATION

7.3.1 : Speed

The speed of data access on WAP handphone depend on the services provided by their respective mobile service providers. Currently what is being offered is 9.6

kilo bits per second which is rather slow. The speed is a real setback because of WAP small bandwidth and does not enable fast transaction. This setback has caused many people dislike in WAP and why WAP is not a big hit compared to internet when it first made appearance in the 1990s. Many mobile users still prefer to surf the internet to get information rather than changing into WAP handphones because of it's speed and the fees. Currently WAP fees differ with each mobile provider but the users are still charged based on per-minute access.

7.3.2 : Insufficient WAP Systems

The package offered for WAP users are not attractive enough to woe in more users to sign up with WAP services due to the fact it is still a new technology and many people has not invested on it by providing WAP systems. In time to come, WAP systems might increase and future will be brighter for WAP.

7.3.3 : Browser Cache Limitation

Internet browser has this capability to store visited sites in their cookies. Therefore, administrator will be easily confused when they see same site again when they clicked on the 'back' button on the toolbar of the internet browser. However this can be overcome by setting up the internet browser to disable the cache capability.

7.4 : FUTURE ENHANCEMENTS

7.4.1 : Security

Security of the system has always been a big issue in most systems. Programmers want to ensure their users that their systems are secure and hack-proof. Truth is no systems are. The more secure you think you are, people will try even harder to hack into your system.

Currently, WAP-MA's security level is still very low and security is a major concern. Therefore, future enhancement and improvement on the current security system is important such as encrypting password and multi layer password restriction by securing the login page with another level of password. This is to ensure only authorize personnel are able to load the page.

Another way to improve security is to configure the Apache web server and the NT application server to the most secure limits possible. As Apache and Microsoft continuously improve their products, WAP-MA will also be improved based on the capabilities of their most current security technology.

7.4.2 : Database Enhancement

Database can be enhanced from time to time with Oracle's improved release from time to time. Each new release will correct their mistakes in the previous release whilst improving on the previous release. Therefore, WAP-MA will also be improved based on the database enhancement.

7.4.3 : Auto Detect Location

WAP-MA currently does not have auto detect location capability. WML and WML Script release 1.1 currently does not have such capability to enable auto detect location. WML and WML Script is controlled by WAP Forum which is a group setup by world's mobile manufacturers such as Nokia, Motorola and many more. However, the future has yet to see on WAP and it is believed further improvement will be done in time to come.

7.4.4 : WAP Gateway

WAP gateway is a gateway to enable WAP handphone to call into and access the system. Currently no WAP gateway is installed and set. Therefore, to enable real WAP handphone to access into the system, a WAP gateway has to be installed.

7.5 : KNOWLEDGE GAINED

WAP-MA is a system where theories learned in classroom can be applied to practical work. The term "practise makes perfect" fits inside the picture here because what is learned if not put into practise will go into waste. The more practise one has, the more practical one will be.

7.5.1 : Network Systems

Network system and how it works is very important when developing and deploying WAP-MA. Different software architecture of the networks have been

studied and on top of that, WAP architecture has to be studied also to enable installing and setting up of the handphone simulator correctly on the computer.

Other than that, network systems and intranet of the faculty is well known because information is needed when setting up servers at different locations. Usually, a firewall is setup in a network system to prevent intrusion. Therefore the security of the systems must be known beforehand to prevent possible network chaos.

7.5.2 : Setup Servers

Since WAP-MA is developed in a Windows NT environment integrating web server, database server and mail server, therefore knowing how procedures in setting up all servers concerned are important. The WAP project comprising three different WAP systems have to be inside the same domain in order to utilise resources (servers) easily. Knowledge of how to setup a new domain and workstations are learned.

7.5.3 : Java Technology

JSP is a language invented based on Java technology. It's main purpose is to utilise existing Java technology. Therefore, when JSP is chosen as the language, Java technology is automatically learned because it was deployed in the system. Technology such as importing classes, servlets, packages and object oriented programming. Java being the forefront web developing language and understanding the concepts will be an advantage in the future when developing any web software.

7.5.4 : Oracle And SQL

Oracle being the most popular database server is the most widely used in the world is bound to be market value for Oracle knowledge. Hence, knowledge gained by setting up and maintaining the server is extremely useful and helpful because it can be applied in real working life. Oracle database which can be manipulated through SQL language in one of it's application SQLPlus added an extra point to the user. Now, the user heeds others by knowing how to execute SQL language on top of being able to maintain Oracle.

7.6 : CONCLUSION

"WAP is a new protocol that will turn mobile telephones into small Internet browsers" as claimed by many. Some claimed it to be the ground laying to open the path for GPRS (General Packet Radio Services).

Therefore, this system is built to further support these beliefs. Although the above beliefs have not been proven right but the system is completed successfully and it's objectives met. The main objective is to build an interactive messaging system specially catered to people on the go niche market. WAP-MA has enabled users to send and receive messages from their friends and having a friend list is to prevent unnecessary disturbance. It has proven and projected at the same time that whatever is possible on the computer through the internet is definitely possible on WAP handphone through wireless protocol.

The completion of the system is an eye-opener to other existing protocols namely WTP besides HTTP. Enrichment of technology knowledge when deploying and

applying such technologies in developing WAP-MA is something beneficial and will be useful in the future because these technologies are used worldwide.

Lastly, WAP-MA is a workable system that can be up and running with minimal changes when deployed in real WAP environment. The future for WAP will be even brighter once it has overcome the setbacks that it's currently facing. By then, WAP will be a hit and WAP-MA will follow suit.

3. Ng Wei He, "Mobile Devices Ride The Wireless Wave", *Wired*, June 2000 Edition.
4. David Butler, John Karkhanavich, Jan T. Libby, James David Sussman, Chris Ullman, Joe Clark, "Configuring Active Server Pages 3.0", Wiley Press Limited, 1999.
5. "Developer's Guide Nokia WAP 2.0", Nokia Corporation, 2000.
6. "Programmer's Guide Nokia WAP Server", Nokia Corporation, 2000.
7. "WML 2.0 Version 1.1", Nokia Corporation, 2000.
8. Kenneth Davis, Jr., "Deliberate Practicing Fundamentals, Design and Implementation", Prentice Hall International Inc., 1998.
9. Multiple authors, "Professional 3.0", Wiley Press Ltd, 2000.

REFERENCES

1. Kenneth E. Kendall & Julie E. Kendall, "Systems Analysis And Design, 4th Edition", Prentice-Hall, Inc., 1998.
2. Karamjit Singh, "Celcom's Turn in WAP Limelight", The Edge, July 10, 2000 Edition.
3. Ng Wei En, "Mobile Devices Ride The Wireless Wave", ComputerWorld, June 2000 Edition.
4. David Buser, Jogn Kaufmann, Juan T. Libre, Brian Francis, David Sussman, Chris Ullman, Jon Duckett, "Beginning Active Server Pages 3.0", Wrox Press Limited, 1999.
5. "Developer's Guide Nokia WAP Toolkit 2.0", Nokia Corporation, 2000.
6. "Programmer's Guide Nokia WAP Server", Nokia Corporation, 2000.
7. "WML Reference Version 1.1", Nokia Corporation, 2000.
8. Kroenke David M., "Database Processing Fundamentals, Design and Implementation", Prentice Hall International Inc., 1998.
9. Multiple authors, "Professional JSP", Wrox Press Ltd, 2000.

10. Retrieved on 22 Jun 2000 from the World Wide Web :
<http://www.wapforum.org>
11. Retrieved on 24 Jun 2000 from the World Wide Web :
<http://www.gelon.net>
12. Retrieved on 11 July 2000 from the World Wide Web :
<http://www.forum.nokia.com>
13. "Download WAP Trial Server" retrieved on 11 July 2000 from the World Wide Web : http://www.nokia.com/corporate/wap/trial_download.html
14. "Nokia WAP Toolkit" retrieved on 11 July 2000 from the World Wide Web : <http://www.nokia.com/corporate/wap/sdk.html>
15. "Join WAP Newsletter" retrieved on 20 July 2000 from the World Wide Web : <http://www.wap.com>
16. "Maxis WAP" retrieved on 5 August 2000 from the World Wide Web : <http://www.maxis.com.my/wap/index.html>
17. "Maxis WAP Pricing" retrieved on 5 August 2000 from the World Wide Web : <http://www.maxis.com.my/wap/price/index.html>
18. "Celcom GSM WAP" retrieved on 10 August 2000 from the World Wide Web : <http://www.celcom.com.my/mobiles/gsm/celcom-wap.html>

19. "GSM - WAP-Clients and Gateways" retrieved on 13 August 2000 from the World Wide Web :
http://www.gsmworld.com/technology/wap_09.html
20. "GSM - WAP-Developers Toolkit" retrieved on 13 August 2000 from the World Wide Web : http://www.gsmworld.com/technology/wap_07.html
21. "Newsgroup" joined newsgroup on 3 July 2000 from the World Wide Web : <http://www.wapunderground.com:81/>
22. "Bean Mailer" retrieved on 5 October 2000 from the World Wide Web :
<http://www.jspinsider.com/beans/beans/email/BeanMailer>
23. "Free Javascript" retrieved on 9 October 2000 from the World Wide Web :
<http://javascript.internet.com/buttons/check-all.html>
24. "Oracle Technology Network" retrieved on 12 October 2000 from the World Wide Web : <http://techr.et.oracle.com/>
25. "Apache 1.3 Documentation" retrieved on 16 October 2000 from the World Wide Web : <http://www.apache.org>
26. "Oracle" retrieved on 28 October 2000 from the World Wide Web :
<http://www.oracle.com/>
27. "Advanced form processing using JSP" retrieved on 2 November 2000 from the World Wide Web :
http://classic.tep.ru/~serge/publications/java/jw/jw-03-2000/f_jw-0331-ssj-forms_p.html

28. "JSPInsider -Jsp Help- JSP Session Object" retrieved on 10 November 2000 from the World Wide Web : <http://www.jspinsider.com/>
29. "Oracle SQL Data Manipulation" retrieved on 15 November 2000 from the World Wide Web :
<http://www.uwp.edu/academic/mis/baldwin/orasql.htm>
30. "Oracle : SQL Plus Tutorial" retrieved on 23 November 2000 from the World Wide Web :
<http://www.uwp.edu/academic/mis/baldwin/sqlplus.htm>
31. "SQL Tutorial" retrieved on 7 December 2000 from the World Wide Web : <http://w3.one.net/~jhoffman/sqltut.htm>
32. "Building Your Own JSP Components" retrieved on 15 December 2000 from the World Wide Web :
http://developer.iplanet.com/viewsource/fields_jspcomp/fields_jspcomp.html
33. "jGuru Java Mail FAQ" retrieved on 10 January 2001 from the World Wide Web <http://www.jguru.com/jguru/faq/printablefaq.jsp>
34. "Maxis WAP-WAP just got better" retrieved on 28 November from the World Wide Web : <http://www.maxis.com.my/consumer/index2.html>