

UTP Mobile Summon System

by

Mohd Nawawi B Md Abdul Jabar

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Business Information System)

JANUARY 2007

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

t
TK

5105.586

.11617

2007

1) wireless communication system
2) mobile computing

CERTIFICATION OF APPROVAL


UTP Mobile Summon System

by

Mohd Nawawi B Md Abdul Jabar

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(BUSINESS INFORMATION SYSTEM)

Approved by,



MOHD HILMI HASAN

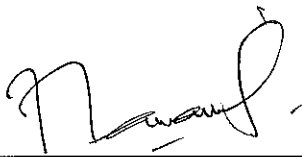
UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

January 2007

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgement, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



MOHD NAWAWI MD ABDUL JABAR

ABSTRACT

As I go through some observation, I found that most of the system now days is using wireless mobile application that most likely to improve the efficiency of handling information via network. Wireless data continues to grow at a rapid rate and, at long last, technologies such as GPRS and SMS are finally taking hold in most of the country. The security department of Universiti Teknologi PETRONAS is responsible in enforcing the rules and regulations such as issuing a summons towards a student who commits any misbehavior offence or actions which are against the rules and regulations stated. The current summon system in Universiti Teknologi PETRONAS posses several problems such as tasks duplication, difficulty in determining the specific compound and difficulty in obtaining the records of the particular person. The focus of this project paper is the study of a java based system which consists of wireless application as well as a portal web in order to solve identified problems as mentioned. One of the main objectives is to provide with an alternative system to the current system. Thus, I would like to suggest to the Security Department of Universiti Teknologi PETRONAS to implement the mobile application system. As I find out, there are several ways of doing this project. J2ME or Java2 Platform Micro Edition would be one of the suitable approaches for this project. J2ME is a version of Java used for developing applications that run on a wireless device platform for instance PDA or cell phone. During the system development process, the System Development Life Cycle (SLDC) methodology is deployed in modeling the business environment of the summons system. The report consists of five chapters named Chapter 1 to 5 accordingly. For the abstract part is under Chapter 1. While in the Chapter 2 describes the background of study, problem statement, objective and literature review. In Chapter 3, describe on the methodology part which explained about the procedures used and the tools/ software used in developing this project. During the system development process, the System Development Life Cycle (SLDC) methodology is deployed in modeling the business environment of the summons system. The result and discussion and conclusion are describes in the Chapter 4 and Chapter 5 respectively. Hopefully this project would be the milestone towards developing the real summons application by the security department of UTP.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Allah SWT and both of my parents, who always there by my side, providing full support and guidance the whole time throughout the project timeline. Very special thanks to my supervisor; Mr Mohd Hilmi Hasan who had shown his support, patience and tolerability during these last two semesters. Not forget to mention Mrs. Aliza Sarlan as the head of FYP Committee and other ICT / BIS lecturers.

Furthermore, I would like to thank all the staffs in the Security Department of University Technology PETRONAS for providing me with the enough information and data that I needed in order to pursue with my project. If it wasn't for them, my project would've not gone far.

Last but not least, I would like to thank all of my fellow friends which either directly or indirectly involved in helping me completing this project. Thank you.

TABLE OF CONTENTS

CERTIFICATION.....	i
ABSTRACT.....	iii
ACKNOWLEDGEMENT.....	iv
CHAPTER 1: INTRODUCTION	
1.1 Background of Study.....	1
1.1.1 History of Java Platform.....	1
1.1.2 Java 2 Micro Edition J2ME.....	3
1.1.3 J2ME Virtual Machine.....	4
1.1.4 Configurations.....	6
1.1.5 Profiles.....	10
1.1.6 Java Server Pagers.....	12
1.2 Problem Statement.....	13
1.3 Objective and Scope of Study.....	13
1.4 Justification.....	14
CHAPTER 2: LITERATURE REVIEW.....	16
2.1.1 Wireless Technology.....	16
2.1.2 Online Testing Engine.....	17
2.1.3 Online Ordering System.....	18
2.1.4 Mobile Web Services.....	19
2.1.5 Target Game Practice.....	20

CHAPTER 3:	METHODOLOGY / PROJECT WORK	
	3.1 Procedure Identification.....	21
	3.2 Tool and Software Required.....	24
CHAPTER 4:	RESULTS.....	21
	4.0.1 System architecture.....	24
CHAPTER 5:	IMPLEMENTATION & SYSTEM TESTING	30
	5.1 System environment.....	29
	5.2 Implementation.....	29
	5.3 System testing.....	39
	5.3.1 MIDlet application testing.....	39
	5.3.2 Web application testing.....	44
CHAPTER 6:	RESULT & DISCUSSION.....	47
	6.1 Introduction.....	47
	6.2 Result achievement.....	47
	6.3 System drawback.....	48
	6.4 Future recommendation.....	48
REFERENCES.....		49
APPENDICES.....		50

LIST OF FIGURES

Figure 1-1: Java Family.....	3
Figure 1-2: Connected Limited Device Configuration (CLDC).....	8
Figure 3-1: SDLC process.....	20
Figure 4-1: System Flow.....	24
Figure 4-2: Use Case diagram for System Administrator on Web Portal.....	26
Figure 4-3: Use Case diagram for User (Security guard) on Web Portal.....	27
Figure 4-4: Use Case diagram for User (Security guard) on MIDlet application...28	
Figure 5-1: Code for User Log in.....	30
Figure 5-2: Code for checking user identity.....	30
Figure 5-3: Code for displaying successful log in process.....	31
Figure 5-4: Code for displaying error warning log in process.....	31
Figure 5-5: Code for summon determination.....	32
Figure 5-6: Code shown parameter separations.....	32
Figure 5-7: Code for search based on categories.....	33
Figure 5-8: Code for checking vehicle records	33
Figure 5-9: Code for updating student and vehicle records	34
Figure 5-10: Code displaying successful update process	34
Figure 5-11: Code for updating student information at server side.....	34
Figure 5-12: Code for User Registration.....	35
Figure 5-13: Code for User Registration.....	36
Figure 5-14: Code for User Profile.....	36
Figure 5-15: Code for displaying user information.....	37

Figure 5-16: Code for updating function.....	37
Figure 5-16: Code Student sub module.....	38
Figure 5-17: Code for updating student information.....	38
Figure 5-18: Shows connection and verifying user ID and password testing.....	41
Figure 5-19: Shows summon registration testing.....	42
Figure 5-20: Shows search testing.....	43
Figure 5-21: Shows User Registration testing.....	44
Figure 5-22: Shows user verifying testing.....	45
Figure 5-23: Shows search testing.....	46

LIST OF TABLES

Table 5-1: Shows connection and verifying user ID and password testing.....	40
Table 5-2: Shows summon registration testing.....	42
Table 5-3: Shows search testing.....	43
Table 5-4: Shows User Registration testing.....	44
Table 5-5: Shows user verifying testing.....	45
Table 5-6: Shows search testing.....	46

CHAPTER 1

1.0 INTRODUCTION

1.1 BACKGROUND OF STUDY

As I go through some observation, I found that most of the system now days is using wireless mobile application that most likely to improve the efficiency of handling information via network. Wireless data continues to grow at a rapid rate and, at long last, technologies such as GPRS and SMS are finally taking hold in most of the country. Thus, I would like to suggest to the Security Department of Universiti Teknologi PETRONAS to implement the mobile application system. As I find out, there are several ways of doing this project. J2ME or Java2 Platform Micro Edition would be one of the suitable approaches for this project. J2ME is a version of Java used for developing applications that run on a wireless device platform for instance PDA or cell phone. During the system development process, System Development Life Cycle (SDLC) methodology is deployed in modeling the business environment of the summons system.

1.1.1 History of Java Platform

Java came to the limelight with the release of HotJava browser, which demonstrated the suitability of Java for the Internet. It established that dynamic and interactive Web pages could be created with Java. The craze for applets also caught up with its release. Subsequently, Netscape 2 for Windows became the first commercial browser to support Java 1.0. Java was at its maximum hype at this time, because it seemed to offer capabilities on the Web that were not otherwise possible. The promise of cross-platform applications also added to Java's reputation.

Since then, Java has come a long way. Its acceptance is so widespread that people can find APIs for carrying out most of the things that can be done in other programming languages. From applets to RMI and from Servlets to speech APIs, users find everything under the Java umbrella. Perhaps this accumulation of tools providing all kinds of functionalities contributed to Sun Microsystem's decision to organize the Java family better. Another reason for this could be the boom of nonconventional computing devices. These devices had actually started emerging before the birth of Java. In fact, the Java language itself was initially aimed at providing programming solutions for such devices. Computing was no longer restricted to servers or PCs. And the world of small devices spread far beyond organizers possessing 16K memory and toy processors. Newer devices not only had increased memory and processor power, but also were proliferating in shapes, sizes, display types and areas, input and output methods, and networking.

Sun Microsystems came up with the idea of dividing the Java universe into three separate platforms for different purposes. The domain of the three platforms was to be determined by the sizes for which they were meant. Thus the platform for conventional computing was called the *Java 2 Standard Edition (J2SE)* and that for enterprise applications and distributed computing became the *Java 2 Enterprise Edition (J2EE)*.

The third platform was meant for the nonconventional consumer devices. These devices are characterized by some typical features, such as mobility, limited memory and processing power, incapability to access power from the mains (being battery powered), small display areas, and limitations and variety with respect to input and output methods. Of course, not all these features must be present on one device. One thing common to most of these devices is that they are all connected to some network, even if not always. The Java platform meant for these devices was called the *Java 2 Micro Edition (J2ME)*.

1.1.2 Java2 Micro Edition (J2ME)

Java 2 Micro Edition platform is not restricted to small devices. It has two *design centers*, depending on memory constraints. That is why this platform has two configurations: *CLDC (Connected Limited Device Configuration)* and *CDC (Connected Device Configuration)*. We need to keep this fact in mind while talking about J2ME. Whenever the programmer are programming for a mobile phone, they are as much within the fold of the J2ME as they are if the programmer are programming for a set-top box with almost as much computing power as a desktop system. But it is true that J2ME is more associated with small devices that have modest amounts of power and memory. Yet, when people program for devices such as set-top boxes, the difference between J2ME and J2SE narrows, and people who have been writing code for the standard edition feel quite at ease with J2ME. It is while writing code for devices such as mobile phones that the peculiarities of J2ME become relevant.

The micro edition platform was conceived and designed in the form of layers (Figure 1-1). Each layer adds some functionality. The primary requirement is, of course, a virtual machine for running Java applications. These, together with the virtual machine, form part of the configuration. On top of this configuration resides a profile. The exact profile that can be added is dictated by the device and also by the configuration that is being used. But it is not restricted to one profile. It can be added another profile on top of the first one.

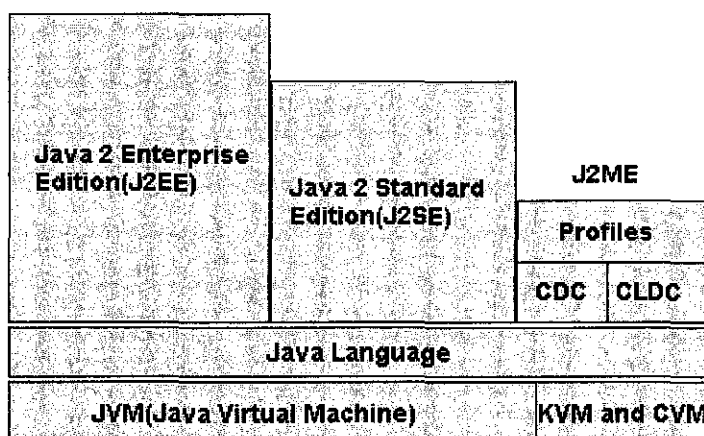


Figure 1-1: Java family

For the time being, they do not have many profiles to choose from: the profile released till recently was Mobile Information Device Profile. The newer Foundation Profile does not really match the concept of a profile because it has no libraries for providing user interface. Even after that day comes that people do have more choices, J2ME is not limited to Configurations and profiles, and it can also be added optional packages over and above configurations and profiles. Thus they may consider a *platform* as parts joined together to form a whole that suits the devices for which their applications are meant. Thus they find a great deal of flexibility in regard to implementing the programming model. Device manufacturers as well as independent vendors can provide their own versions of virtual machines if they follow the Java Virtual Machine Specification. If so, their virtual machines would be no less capable than those provided by Sun. Devices can be sold in Java-enabled condition, or they can be Java enabled later on by the user. Similarly, applications may be loaded onto the devices before the devices are sold, or the user may download them from the Web.

1.1.3 J2ME Virtual Machines

The Java virtual machine used in the standard edition and the enterprise edition is meant for use only on desktop systems and servers. Its size renders it unfit for the small devices, which need a virtual machine with a very small footprint. These are devices such as the mobile phones, two-way pagers, PDAs or handheld devices, screen phones, smart phones, and so on. But J2ME also targets other devices that can accommodate a larger footprint but that still cannot use the conventional Java virtual machine in its original form. This category of devices includes set-top boxes, car navigation systems, handheld PCs, and so on.

In view of all these, two virtual machines were developed for the two design centers. The devices in the first category are supposed to use the smaller footprint *K Virtual Machine (KVM)*, while those in the second category use the larger footprint *C Virtual Machine (CVM)*.

The K Virtual Machine (KVM)

As we are probably aware by now, the KVM is a new, highly optimized Java virtual machine for resource-constrained devices. Appropriately, the K virtual machine is very small in size (about 40-80K), and every care has been taken to make it suitable for devices such as pagers, cell phones, and PDAs. Like any other Java virtual machine, it enables user to download and execute applications. Like applets, these applications provide a method for presenting dynamic content.

The KVM can run on any system that has a 16-bit/32-bit processor and 160-512K of total memory. As of now, the KVM has no support for certain features such as determinism, long and float data types, and so on. The design of KVM is based on some important considerations, including the small size to conserve as much space on the device memory as possible (both in terms of storage and runtime memory) and its capabilities to run on low-power processors, to enable partitioning of the VM, and to fulfill the Java assurance of portability. This last issue could have been a big problem, considering the diversity of small devices. There is actually no full portability, but porting is not very difficult because the KVM was implemented in C language.

Just as applets are written using the APIs available in the standard edition, these applications (spotlets, MIDlets, and more to come) are written using Java APIs available in CLDC, CDC, MIDP, Foundation Profile, and so on. And just as applets are run using the conventional JVM, these applications are run using the KVM.

We should note here that, when we talk of KVM, we mean the reference implementation of KVM provided by Sun. However, J2ME can use any VM that is written according to the Sun's Java Virtual Machine Specifications (JVMS) and can handle as much work as the K virtual machine.

The K Virtual Machine can theoretically run several profiles, but it cannot run quite perfectly all the profiles and APIs that are not specifically designed for it, just as it cannot be used to run the CDC. It is basically meant for CLDC and, at present, for only one profile: MIDP. This means that applications written for the conventional JVM or even for CVM most probably cannot run on the KVM without some changes. But the converse is not true, applications written for KVM can easily run on the CVM or the normal JVM.

A more recent version of KVM (version 1.02) has some improved capabilities, such as a better garbage collector, debugging support with KDWP, a better preverifier, and so on. KDWP is a Debug Proxy (Debug Agent) tool. Debug proxy is tool that is interposed between a Java IDE (Integrated Development Environment) such as Forte and the KVM for debugging purposes. This version includes an implementation for Linux, too. Certain other bugs have been removed and the performance of the interpreter has been improved.

1.1.4 Configurations

As mentioned, configurations form the base of the J2ME platform. They provide the basic minimum functionality for a particular category of devices. This functionality is to be common to all devices in that category, irrespective of other factors. The very reason for dividing the platform into configurations and profiles is to serve this purpose. The devices targeted by J2ME, even those in the same category, are so diverse in terms of their capabilities that it would have been difficult, if not impossible, to implement the J2SE kind of model for them.

The whole platform had to be conceived in the form of layers, operating one on top of another. The base is formed by one configuration, over which you add one or more profiles. A *configuration* includes the virtual machine and the required basic minimum libraries. Because the devices can be broadly divided into two categories, two configurations were needed. These are the *Connected Limited Device Configuration*

(*CLDC*) and the *Connected Device Configuration (CDC)*. Their domains may overlap in some cases, because there is not any definite border between the two.

With some devices, it would be difficult to decide whether CLDC or CDC is the more suitable configuration. For example, some screen phones may have more resources which can be best explored by CDC-based applications where as some may not have enough resources to run the heavier CDC-based applications.

Configurations define the horizontal limits of the platform for a category or family of devices. They specify what parts of the Java language, what features of the Java virtual machine, and what core libraries are supported. As mentioned previously, users may sometimes have the option of using either of the two configurations, but they cannot use both simultaneously. They must choose one before start running Java applications on the device. In fact, the manufacturer may have decided this for the user.

CLDC

The Connected Limited Device Configuration, or CLDC, has been designed for severely resource-constrained devices such as today's cell phones, PDAs, and so on. Therefore, it has everything optimized to a great extent. The virtual machine (KVM or any other similar implementation adhering to the specification) is small, and some of the Java language features are not supported. But the real difference is that the libraries available are very few in number. What this means is that we do not need to learn the basics again, what we learned about the standard edition suffices for the most part, but the libraries must be used with great care (Figure 1-2).

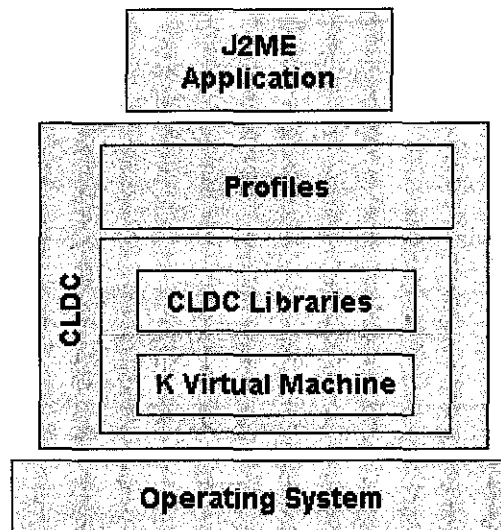


Figure 1-2: CLDC

Some of the main features of CLDC are as follows:

- Data types long and float are not supported. All the methods of J2SE inherited classes that use these data types have been removed.
- Several runtime exceptions are present, but the number of runtime errors has been reduced significantly for the classes included in CLDC. In fact, only the following three errors are available:

- `java.lang.Error`
- `java.lang.OutOfMemoryError`
- `java.lang.VirtualMachineError`

Other errors are handled in an implementation-specific manner.

- To make garbage collection simple, support for finalization is not provided. There is no *finalize* method in the `java.lang.Object` class.
- Java Native Interface (JNI) is not supported in CLDC. The purpose to eliminate-dependence such that the applications can be ported to any platform containing the virtual machine. .

- In the standard edition, we can mark objects for possible garbage collection. This cannot be done with CLDC. In other words, there is no support for weak references.
- Verification of classes to check whether the code is well-formed is done off-device, that is, on the desktop system on which the applications are developed. This is done by a tool called *preverifier*.
- A different security model is used in CLDC, which is somewhat similar to the one used in browsers for downloaded applets. The reason is that the model used in the standard edition is too heavy for small devices, and the security needs of the connected devices are similar to those of the browsers.

There are only four packages available in CLDC. Most of the J2SE packages such as `java.lang.awt`, `java.lang.beans`, and so on have been dropped. CLDC contains only the following packages:

- `java.io`: Stripped-down version of the J2SE `java.io` package. It contains the classes required for data input and output using streams.
- `java.lang`: Stripped-down version of the J2SE `java.lang` package. It contains the classes that are basic to the Java language such as the wrapper classes for data types.
- `java.util`: Stripped-down version of the J2SE `java.util` package. It contains classes such as `Calendar`, `Date`, `Vector`, and `Random`.
- `javax.microedition.io`: A newly introduced CLDC-specific class that defines the Generic Connection Framework. It contains the classes for handling all types of connections by using the same framework.

1.1.5 Profiles

Configurations provide very little functionality; they just prepare the ground on which to add whatever we need. It is the profiles that give functionality, such as the graphical user interface. For example, if we have just CLDC installed on a device such as a PDA, we cannot create any user interface objects. No GUI is possible in the applications. For that, we need a profile. This is why Sun added a package called *KJava* with CLDC to enable testing and development, even if KJava is to be abandoned later. As a matter of fact, Sun Microsystems has provided tools for Palm development with its new wireless toolkit version 1.0.2 and the kit is based on CLDC and MIDP; there is no KJava now.

A profile may add other kinds of functionality, such as better networking support, database management, distributed computing, and so on. Like configurations, profiles may also be device-category specific. Some profiles may be useful for small devices, while others may be suitable for less-constrained devices. For example, MIDP and PDA Profile are used for mobile phones and PDA-like devices, respectively, on top of CLDC. On the other hand, Personal Profile is used for devices such as set-top boxes, on top of CDC.

Mobile Information Device Profile (MIDP)

CLDC can be used for writing applications for small devices. But it gives very limited functionality. There is no way to provide a graphical user interface, unless we use KJava, which may become obsolete soon. For this and other reasons, it is necessary that we use some kind of profile if we want to build an effective application. The only profile so far available for small devices is the *Mobile Information Device Profile (MIDP)*. The most common mobile information devices are cell phones, so this profile is considered the profile for cell phones. Now that the Palm port has become available, it can be used for Palm devices also.

MIDP sticks to the CLDC approach of minimizing resource usage but provides ways to add reasonably a good user interface, given the constraints. It introduces a new application model in which every application is called a MIDlet and behaves somewhat like an applet. It can have three states: active, paused, and destroyed. The application manager software manages the lifecycle of the application. There is also a method to make data persistent. The classes it contains in addition to those provided by the CLDC are as follows:

- `javax.microedition.midlet`: It is this package that defines the application model used in MIDP. It has a single class called MIDlet, with methods for enabling the application-managing software to create, pause, and destroy the application and perform some other tasks.
- `javax.microedition.lcdui`: This is the package responsible for providing the user interface. It includes classes such as those for creating user-interface elements (buttons, text fields, choice groups, gauges, images, and so on) and for handling events (listeners). This is basically a game-oriented user interface package, but can be used for other UI purposes.
- `javax.microedition.rms`: This package provides the capability to make data persistent. For this purpose, the main class that's included is the `RecordStore` class. In addition, there are interfaces for comparison, enumeration, filtering, and listening.

1.1.6 Java Server Pages (JSP)

A server-side technology, Java Server Pages is an extension to the Java servlet technology that was developed by Sun.

JSPs have dynamic scripting capability that works in tandem with HTML code, separating the page logic from the static elements; the actual design and display of the page; to help make the HTML more functional (i.e. dynamic database queries).

A JSP is translated into Java servlet before being run and it processes HTTP requests and generates responses like any servlet. However, JSP technology provides a more convenient way to code a servlet. Translation occurs the first time the application is run. A JSP translator is triggered by the `.jsp` file name extension in a URL. JSPs are fully interoperable with servlets. It can include output from a servlet or forward the output to a servlet, and a servlet can include output from a JSP or forward output to a JSP. JSPs are not restricted to any specific platform or server. It was originally created as an alternative to Microsoft's ASPs (Active Server Pages).

1.2 PROBLEM STATEMENT

UTP currently has the summons system that is black and white or in other word manually handled and managed by the respective security department. This current system is not efficient as I discovered lately that students are having problem checking their status on summons. Besides, there are also some job redundancy and difficulty in determine the summons compound. The system is not really compatible as the result of most students do not realize on updated summons. Mostly, the security guards have to manually keep the entire particular summons into the database. Meaning they have to use some sort of summons form in order to fill up information of particular student. Then they have to go back to their office to key in all those data in the database. In order words, the task is done twice and this would be waste of time.

1.3 OBJECTIVE AND SCOPE OF STUDY

1. To provide an alternative way to implement the job.
2. To enable security guards to check students and vehicles records.
3. To eliminate the repetition of doing the same task.

In doing this project, several scopes have been determine. The scopes are as follow:

- i. The system will have security guard as user who capable of retrieving records from the database and as well as system administrator.
- ii. The identity authentication is based on staff ID and IC number.
- iii. MIDlet application will be develop using J2ME platform. It is going to be Web-based system whereby the Web Portal will be develop by using Java Server Pages (JSP) and Servlet in which specifically for administration used.
- iv. The system is based on UTP Security Department and all types of summon, procedure and information needed will be gathered from the security department.
- v. For the security purposes, the Bouncy Castle Cryptography package will be implemented using Rijndael algorithm in which is going to be explained later on.

- vi. MIDlet application provides the synchronization mechanism for keeping those records in the database even if it is offline. This will be explaining in detail as well.

1.4 JUSTIFICATION

This project has its own justification for ensuring the definite objective in implementing the system. The justification is needed to provide benefits and advantages of the project so that it can be understood by the end user of the system. This application is developed leading to the usage of Java technology integrated with rules and regulation of UTP. The project also helps the security department in keeping all the information into the database effectively. Thus, time can be saved as the system is automated.

Besides, another additional feature of this system is that students can settle their summon compound on the spot which means at the specific location where the summons take place and all necessary information can be transmitted directly to the database. As we all know, it is different from the current system which uses a tangible form by the security guard that the form itself may seem to expose to several risks such as lost or tear up or misplace.

In order to solve the problem identified in the current system, research and analysis has been done which lead to a list of several techniques or methods or useful technology that can be used to provide solutions. Thus, J2ME platform is chosen to support the wireless application platform because it is free to download and could save cost as well. It is also considered as new technology which owns great future in network and communication especially in wireless application. PDA is selected as mobile devices as it provides user with touch screen and stylus that would make them easy to enter data which going to be processed. GPRS technology has been chosen because it leads to great potential to be used in wireless application. It is much less costly as users do not have to use it all time, instead using only when it is needed.

The coverage area is also larger compared to other technology such as Bluetooth. Java Server Pages (JSP) will be used to develop the Web based for administrator as this well known technology allows HTML to be combined with Java on the same page. The Java provides the processing, and the HTML provides the page layout that will be rendered in the Web browser.

CHAPTER 2

2.1 LITERATURE REVIEW

From the author's research, and some other reference as well, it shows that there are a number of previous works that has been done that relates to the wireless technology.

2.1.1 Wireless Technology

When viewed as a method of data transport, wireless technology appears very similar to wired technology. We have a piece of hardware, a method of transmission, and connections on both ends that transform data from human-intelligible to transportable and back. For both wired and wireless technology, the range of transmission is an issue. We can't move our laptop 15 feet from the wall jack when depending on a 10-foot cable. Similarly, we can't go out for a jog and expect our in-home cordless phone to keep a connection five miles away from its receiver. But if we get either a 20-foot cable or a wireless connector of sufficient power, we can move our laptop 15 feet away from the wall jack; and if we get a cellular phone, we can go jogging five miles away from our house and still take calls (as long as our service provider has a reasonable antenna set up).

The methods of connection and ranges of service available vary in wireless technology just as they do in wired technology. Home telephones with a wireless handset have a more limited range than cellular phones; infrared transmissions have a more limited range than radio-wave (including microwave) transmissions. Different types of wireless solutions can communicate ten feet, ten miles, or with a satellite in orbit.

Everyone is talking about the Wireless Internet, and that means we will all soon be able to surf the 'net with our palm pilots, cell phones, and other mobile devices. Currently, there is a lot of interest in creating applications for this market, and currently there are a few ways to do it. The Wireless Application Protocol (WAP) is one way, and another is to use Sun Microsystems's Java 2 Micro Edition (J2ME) Mobile Information Device Profile (MIDP).

Most Internet technologies are designed for desktop and large computers running on reliable networks with relatively high bandwidth. Hand-held wireless devices, on the other hand, have a more constrained computing environment. They tend to have less memory, less powerful CPUs, different input devices, and smaller displays. Further, wireless networks have less bandwidth and more latency (delay) compared to wired computer networks.

The Wireless Application Protocol (or WAP), which is a specification developed by the WAP Forum, takes advantage of the several data-handling approaches already in use. Developing wireless applications using WAP technologies is similar to developing web pages with a markup language because it is browser based. Another approach to developing wireless applications is to use the Mobile Information Device Profile (MIDP).

With either WAP or MIDP, the Java programming language plays an important role. In WAP, Java Servlets and Java Server Pages (JSPs) can be used to generate Wireless Markup Language (WML) pages dynamically, and in MIDP, applications (also called MIDlets) are written in the Java programming language.

2.1.2 Online Testing Engine

One of the project or application developed was the Online Testing Engine. In which we are clearly know that Internet has made it possible to do many things remotely and is now a fact of life. People can buy things, play games, chat with anybody sitting anywhere in the world, and do innumerable other things. One of the areas where the Net can play a very positive role is education. Examinations are an important part of education. There are applications to manage these exams.

This Testing Engine application is called **TestManager**. It allows the user to take exams remotely. Many such applications are available. The specialty of this application is that it works on a Palm device; In other words, people don't need a PC to take exams. Since they can do it on a PDA, theoretically there is no need to sit in one place for the whole duration of the exam. However, whether the examination authorities who may

potentially use such an application allow them to move around during the period of the exam is another matter.

The TestManager uses CLDC and Kjava – it is basically a spotlet. In addition, it makes use of ASP and a third-party XML parser to read XML output from ASP scripts. The project can be divided into two parts: the server side and the client side. On the server side, it has ASP scripts and an MS Access database. On the client side, it has Java (J2ME) classes, including those of the parser.

2.1.3 Online Ordering System

Another application developed was an Online Ordering System. Even as e-commerce becomes popular, m-commerce is taking its place: handling business on the move is the new mantra. Anybody with anything to sell or any business proposal wants to make it available via mobile devices. Because J2ME is meant for these devices, people believe it very relevant to present an example of a J2ME application that enables people to do business on, say, a mobile phone. They have already built a CLDC application as mentioned above, so now turn to MIDP. This is certainly appropriate, considering mobile phones remain the most common small devices and are growing at the fastest rate.

This application is called **OrderManager**. It enables the user to place an order by using a cell phone. A customer may place an order merely after viewing a list of available items, which is what he can do on a cell phone. But still the application is useful if it makes the user aware of the products a supplier offers and, as a result, the user decides to order while on the move. This application presents the opportunity to place such an order. If the ordering option is not available in such a scenario, the customer's impulse may subside or he may opt for another supplier.

OrderManager uses the Mobile Information Device Profile (MIDP) of the J2ME platform, along with JSP and a third-party XML parser. The application also can be

divided into two parts: server side and client side. Because the J2ME classfile is on the mobile phone, the client side is what interests us here.

2.1.4 Mobile Web Services

Next application that has been developed was Mobile Web Services. Many things that once required a trip to the market or elsewhere can now do on the Internet. People can purchase books, groceries, and even vegetables on the Net. An ocean of information is also their reach if they can access the Net. People can contact among them without making an international call and avail themselves of many other services.

The next phase was to integrate already available services and add new ones to reduce the user's involvement in installing different software on a machine to a minimum. In the resulting network of services, various components would interact automatically to deliver what exactly they need. This goal was the idea behind Microsoft's .NET, as well as Sun Microsystems' SunOne initiatives. These initiatives aim to provide services not only on desktop system that connect to the Internet, but on mobile devices as well.

The last few years also saw an explosion of mobile and/or wireless devices. The same services that they find on larger systems, or at least any that the device can deliver- are now becoming available on small, mobile, wireless devices.

Again as mentioned the application was called *Mobile Web Services*. It offers the user three services- namely, Weather, Movies and News. As part of the Weather service, the user gets information about the weather in certain cities. The Movies service enables the user to discover what films are running at which movie hall and also to book a ticket for any film. The News service makes available the headlines and detailed news to the user on a mobile phone. The application was created using MIDP so that user can run it from a mobile phone or a two-way pager. Because MIDP is now available for Palm too so it can port the same application to a Palm device with very little effort.

2.1.5 Target Practice Game

Game development has always been one of the most popular programming exercises. Whether a programming environment facilitates developing good games is one important criterion for judging the effectiveness of the environment. In fact, some programmers criticize MIDP because it is not possible to create higher-end games using it.

Anyway, this project was considered a simple game in which by using CLDC. But of course, this can be done only by using KJava package, since that is where the user interface is there. The fact KJava is a temporary arrangement does not really matter much, because in practice, CLDC will always be backed up with user interface functionality.

The game developed was a simple one. The purpose is to illustrate how some more classes of CLDC can be used for developing applications. One of these classes is the `Thread` class of the `java.lang` package. Classes of KJava package are used for UI part. The KJava package is likely to become a regular feature of some profile in a modified version.

The project is a game of “shooting”. A target moves, and player has to shoot at it. He gets points for every successful hit. Not very exciting for a seasoned player used to play games on his PC, but a newcomer to J2ME platform can hopefully get some excitement from developing an application such as this with just CLDC.

CHAPTER 3

3.0 METHODOLOGY / PROJECT WORK

3.1 PROCEDURE IDENTIFICATION

During the system development process, System Development Life Cycle (SDLC) methodology is deployed in modeling the business environment of the system. To manage this, a number of system development life cycle (SDLC) models have been studied: waterfall, fountain, spiral, build and fix, rapid prototyping, incremental, and synchronize and stabilize. Figure 3-1 shows SDLC process.

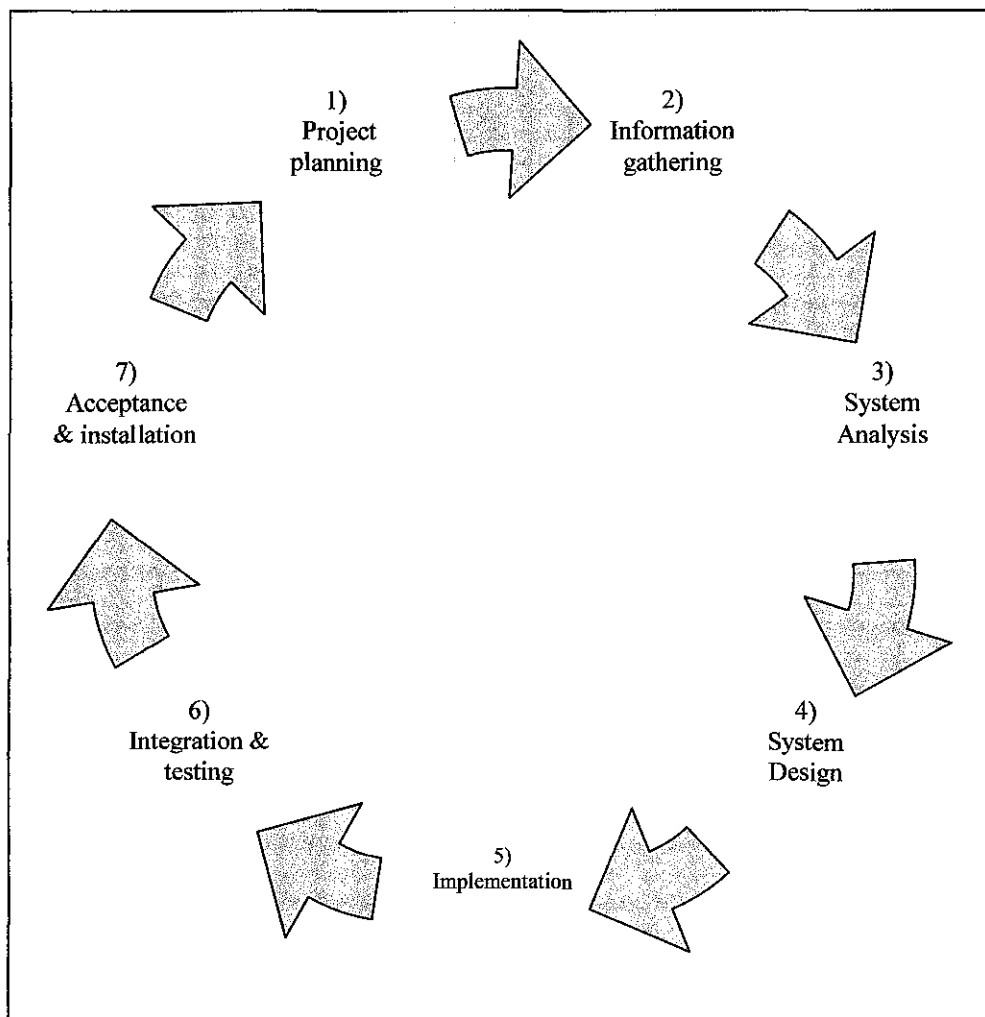


Figure 3-1: SLDC process

Finally as several research and study has been done, I have decided to use the waterfall model: a sequence of stages in which the output of each stage becomes the input for the next. These stages can be characterized and divided up in different ways, including the following:

- **Project planning, feasibility study:** Establishes a high-level view of the intended project and determines its goals.
- **Information gathering:** Any necessary and useful information will be collected through out this stage.
- **Systems analysis, requirements definition:** Refines project goals into defined functions and operation of the intended application. Analyzes end-user information needs.
- **Systems design:** Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudocode and other documentation.
- **Implementation:** The real code is written here.
- **Integration and testing:** Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.
- **Acceptance, installation, deployment:** The final stage of initial development, where the software is put into production and runs actual business.

3.1.1 Project planning, feasibility study

In the first phase, developer required identifying any problems surrounding them and in this case it will be focused on UTP itself. Any selection and determination of topic would be provided in the project proposal submitted to respective supervisor. Problems that have been identified are task duplication, difficulty in accessing the database and problem in retrieving desired information online. Thus, this could be an opportunity to improve efficiency of the current summon system in UTP by implementing wireless application. The identified objectives would be the process of transmitting data and information online, help to improve the effectiveness in retrieves and updating data and also to enhance efficiency of database management system.

3.1.2 Information gathering

In this phase, will involved in determine any necessary information required to develop the entire system. One of the methods used to gain information is by interviewing some of the employees working in the respective department. The information gained such as how the current system works and also some other data are very helpful in developing this project. Besides, the knowledge and understanding about J2ME, JSP, GPRS, PDA and others might be very useful in order to smooth things up. It is also to assist developer in making justification for choosing the suitable techniques.

3.1.3 System analysis, requirements definition

In this stage, process of analyzing system will be taken place here. The system requirements will be identified along with tools and techniques to be applied in developing this system. Unified Modeling Language (UML) approach which is object-oriented modeling would be suitable way to determine the system requirement. By using this method, use case diagram will be created and actors who involved in an interaction and names the type of interaction will be identified. Actors identified are security guard and system administrator.

3.2 TOOLS / SOFTWARES

Tools/equipment required:

Software/Technologies:

- Operating System Windows 98,2000/NT/XP
- Java2 Micro Edition Wireless toolkit 2.0
- Java Server Pages (JSP)
- Microsoft Project
- Macromedia Dreamweaver MX
- Microsoft SQL Server 2000 – For database
- IBM DB2 Universal Database Version 8
- JCreator 2.5 Pro
- MIDlet
- Apache Tomcat 4.1.29

Hardware tools:

- PDA – Personal Digital Assistant that support Java, MIDP, CDLC & GPRS application.

CHAPTER 4

4.0 RESULT AND DISCUSSION

After going through the first, second, and third stages which are project planning, information gathering and system analysis respectively, as a result I managed to come out with the system architecture for this particular project.

4.0.1 System architecture

System Flow

The flow of the system will be provided before moving into detail on system design. This system consists of 3 main components which are MIDlet application on mobile devices, Web based application and Servlet.

Each and every component is important and interacting with each other. Firstly, the normal system flow starts with user request from the MIDlet or Web based application to server in which connected to database through Internet connection. Process and operation involved will be managed by *servlet*. The system flow shows in Figure 4-1.

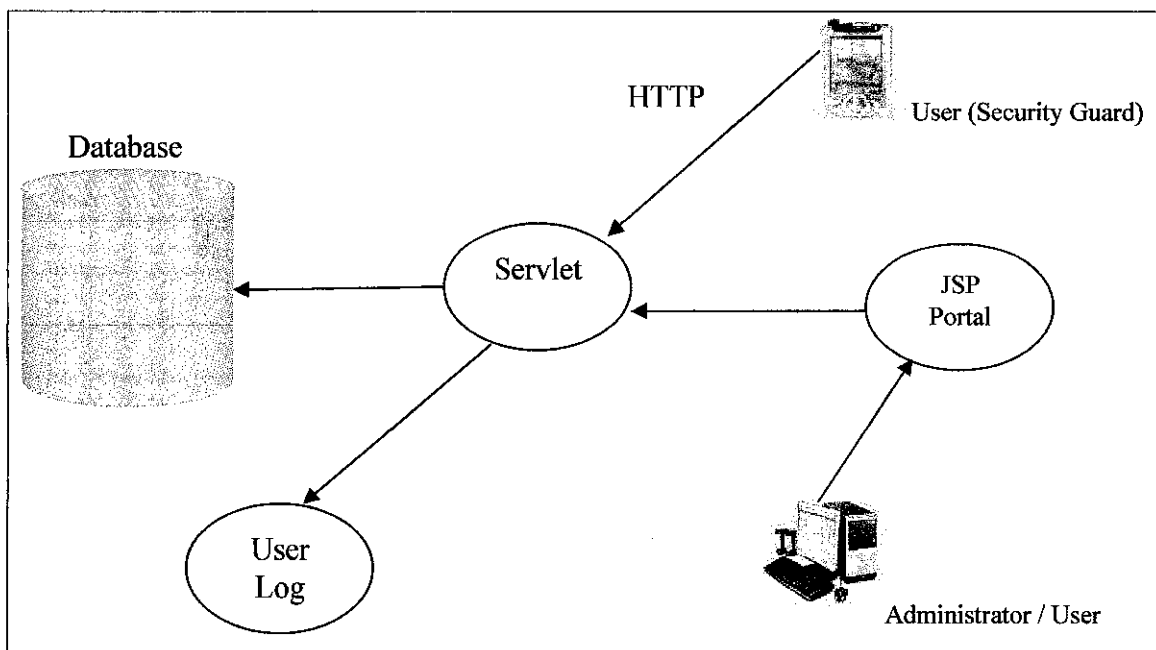


Figure 4-1: System Flow

For web portal user which is system administrator, action taken that required any operation connecting to the database will be served by *servlet*. The same applied for any request from MIDlet application. However, for information display and information editing will be based on level of user. Only those who own authentication or privilege is allow editing and updating records.

Conceptual Modeling / Design

Unified Modeling Language (UML) approach selected in modeling process because Java is an object-oriented base. The objective of system modeling is to provide developer with clear view of the system being developed, how it functions and the interactions between system and its end-user and also among components inside the system. Figure 4-2 shows use case diagram with modules that is available in the system.

The system administrator will need to log in before capable of proceeding to other actions. The system administrator also responsible in registering new user before they are capable using functions provided in MIDlet application or Web portal. System administrator will own highest level of authentication and will have functions such as search, view, retrieve and display information as well as insert, update, edit and delete information. For normal user, they have limitation in accessing the information from database compare to administrator. This showed in Figure 4-3

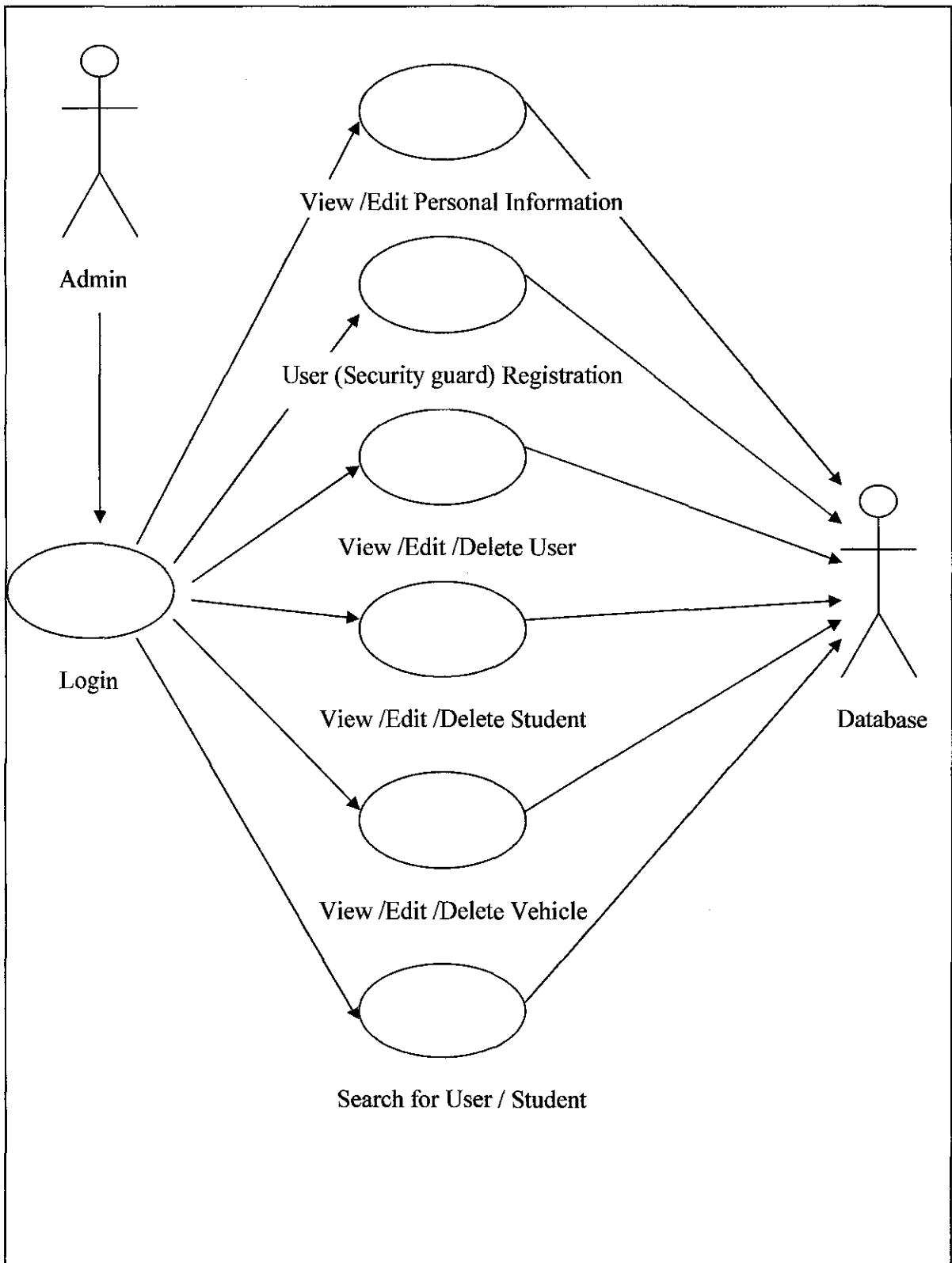


Figure 4-2: Use Case diagram for System Administrator on Web Portal

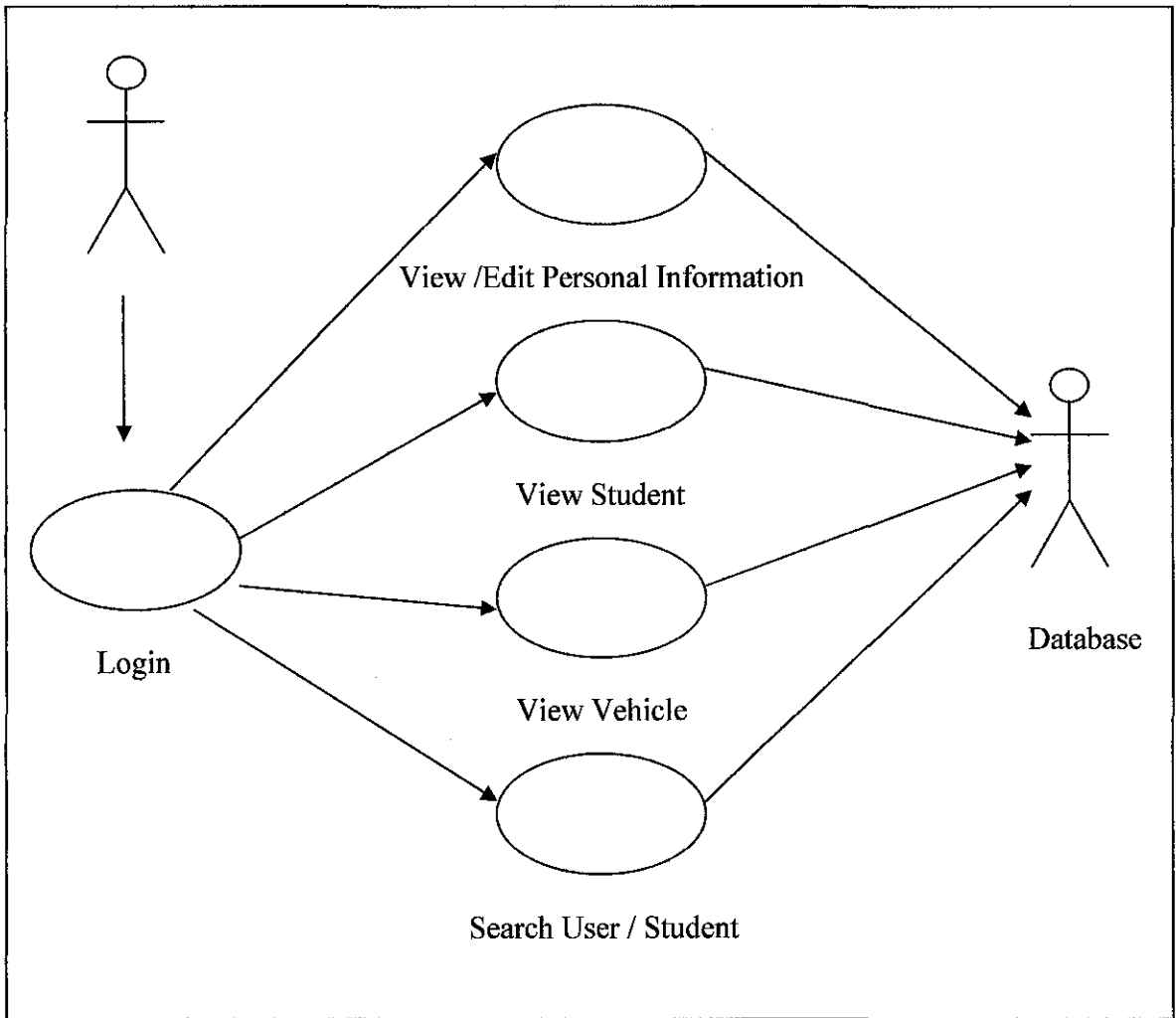


Figure 4-3: Use Case diagram for User (Security guard) on Web Portal

Once registration for user has been made, a user may eligible to use the MIDlet application by using mobile devices. Figure 4-4 shows actions can be taken by respective user on MIDlet application.

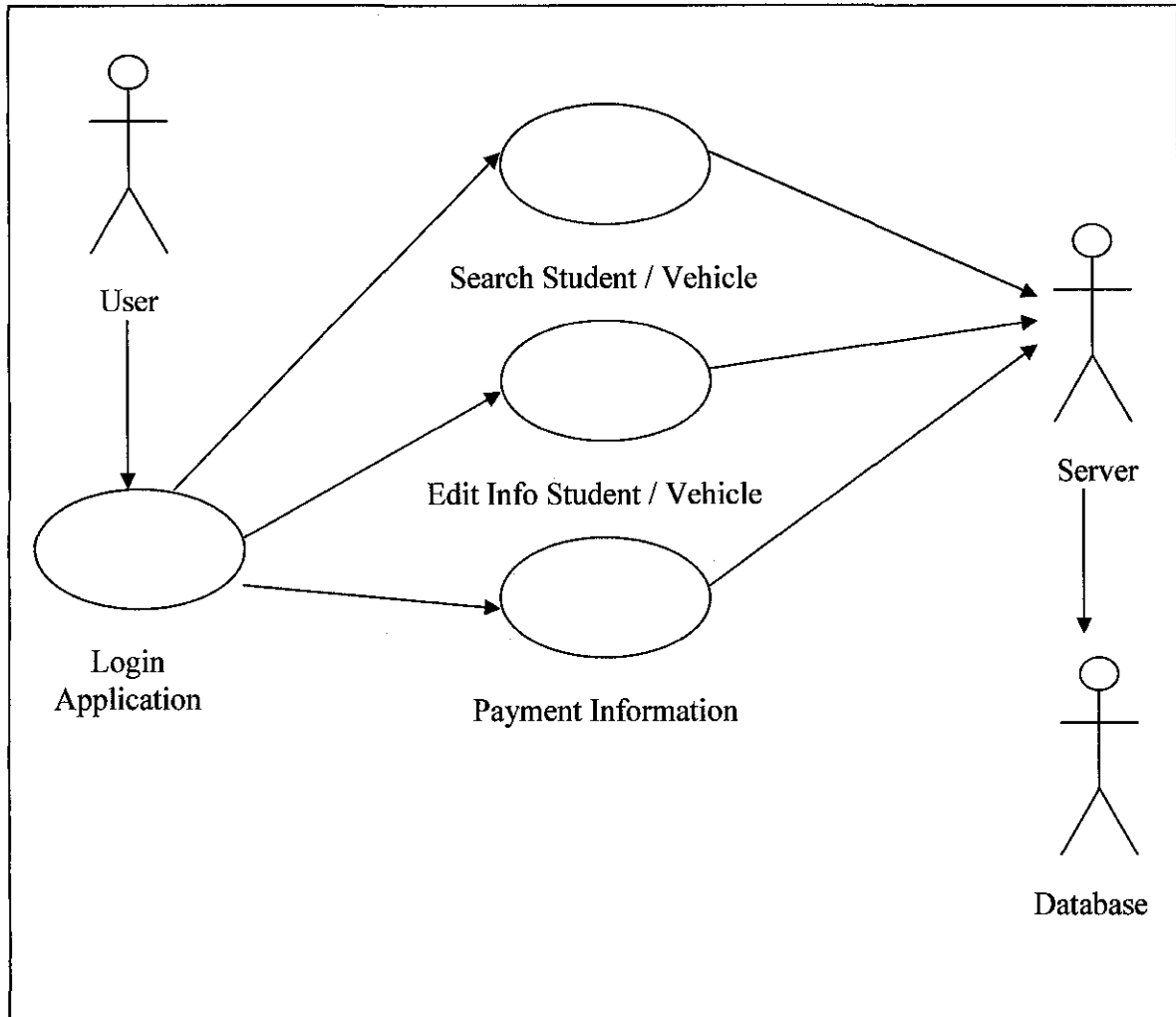


Figure 4-4: Use Case diagram for User (Security guard) on MIDlet application

CHAPTER 5

5.0 IMPLEMENTATION & SYSTEM TESTING

In this chapter will be discussing about system development involved installation required for server and client side, system implementation in which showing several steps in developing this system as well as code implementation for main component, data input and output and analysis of the system.

5.1 System Environment

This system is developed using MS SQL server 2000 for database part. Meanwhile, Apache Tomcat 4.1 is installed at the server side and J2ME Wireless Toolkit 2.1 application is installed at user side in order for the system to run.

5.2 Implementation

In the implementation process, system prototype consists of three main components. The components are MIDlet application which will be used by normal user, web based application, as well as web server. User will log into the system based on user level either normal user or as system administrator and type of application used whether web or mobile application. System server will be processing each and every action triggered by user and keep them inside specific log file.

5.2.1 MIDlet application implementation

MIDlet application will be running on user PDA. In this application, user can log in, submit summon information, check student and vehicle information and update student and vehicle information. Below will be showing several sub module involved.

5.2.1.1 User log in sub module

User log in process will determine whether users are registered or not and will them be capable of using the system.

```
HttpConnection c = null;
InputStream iStrm = null;
String app = "mobile", user, ic, encData;
user = username.getString();
ic = password.getString();
encData = encrypt(user+" "+ic);
// Data is passed at the end of url for GET
String url = "http://172.16.8.54:8000/Saman/formProcessor?" +
"app="+app+"&action=LoginUser&encData=" +encData;
System.out.println(">>: "+url);
try
{
startTime = (new Date ().getTime());
c = (HttpConnection)Connector.open(url);
```

Figure 5.1 Code for User Log in

```
if(!request.getParameter("app").equals("web"))
{
decData = decrypt(request.getParameter("encData"));
StringTokenizer st = new StringTokenizer(decData);
while (st.hasMoreTokens()){
str[i] = st.nextToken();
System.out.println(str[i]);
i++;
}
}else{
str[0] = request.getParameter("idUser");
str[1] = request.getParameter("ICno");
}
stmt = con.prepareStatement("SELECT * FROM user WHERE
idUser = ? AND ICno = ?");
stmt.setString(1, str[0]);
stmt.setString(2, str[1]);
rs = stmt.executeQuery();
```

Figure 5.2 Code for checking user identity

```

iStrm = c.openInputStream();
//DataInputStream dOS = c.openDataInputStream();
// 2) Get header information – none 3) Get body (data)
int length = (int) c.getLength();
if (length > 1)
{
byte servletData[] = new byte[length];
iStrm.read(servletData);
endTime = (new Date()).getTime();
interval = endTime - startTime;
paparMenuLogin(decrypt(new String(servletData))+ " in "+interval+"
milisec");
}

```

Figure 5.3 Code for displaying successful log in process

```

{
alert = new Alert("Warning", "Bad Login. Please try again!",
warning,AlertType.WARNING);
formLogin.removeCommand(offCmd);
formLogin.addCommand(exitCmd);
display.setCurrent(alert, formLogin);
}

```

Figure 5.4 Code for displaying error warning log in process

5.2.1.2 Summon determination sub module

This sub module will transmit summons information based on respective categories to the server. It will also act as sub module for synchronization process during offline mode.

```
if(pol.equals("170A"))
{
formReg = formvecSummon;
if(on)
{
vecSummon = vecSummon.getString();
loc = place.getString();
fault = offence.getString();
regNo = regno.getString();
}else{
noPol170 = omPol170A.getString();
loc = omplace.getString();
fault = omoffence.getString();
regNo = omregno.getString();
}
String url170A = "http://localhost:8000/Saman/formProcessor?" + "app="
+app+"&action=addPol170A&initData=" +encrypt(noPol170+
"+regNo)+"&offence=" +encrypt(fault)+"&lokasi="+encrypt(loc);
url = url170A;
```

Figure 5.5 Code for summon determination

```
decData = decrypt(request.getParameter("initData"));
StringTokenizer st = new StringTokenizer(decData);
while (st.hasMoreTokens()){
str[i] = st.nextToken();
System.out.println(str[i]);
i++;
}
//str1 = st.nextToken(); str2 = st.nextToken();
str[2] = decrypt(request.getParameter("place"));
str[3] = decrypt(request.getParameter("offence"));
tarikh = String.valueOf(masa.toLocaleString());
stmt = con.prepareStatement("INSERT INTO vecSummon VALUES(?, ?, ?, ?,
?, ?, ?)");
```

Figure 5.6 Code shown parameter separations

5.2.1.3 Search records sub module

In this sub module, it is important as to retrieve and display student and vehicle records.

```
if(src.equals("student"))
{
urlDrv = "http://localhost:8000/Saman/formProcessor?" + "app="+app+
"&action=viewPemandu&encData=" +encrypt(drv);
url = urlDrv;
}
else
{
urlVec = "http://localhost:8000/Saman/formProcessor?" +
"app="+app+"&action=viewKenderaan&encData=" +encrypt(vec);
url = urlVec;
}
```

Figure 5.7 Code for search based on categories

```
stmt = con.prepareStatement("SELECT k.regno, k.vectype,
k.color, m.noSiri, m.exdate FROM vehicle k, roadtax m
WHERE k.regno = ? AND k.regno != m.regno");
stmt.setString(1, srcVec);
rs = stmt.executeQuery();
colCount = rs.getMetaData().getColumnCount();
vec1 = new Vector();
vec1 = oneColumnQuery(rs, colCount, vec1, s);
if (vec1.size() > 0)
{
System.out.println("viewing detail vehicle&taxation info");
System.out.println(vec1.size());
session.setAttribute("vec1",vec1);
if(!request.getParameter("app").equals("web"))
{
response.setContentType("text/plain");
out.print(encrypt((String)vec1.elementAt(0)+
'+(String)vec1.elementAt(1)+
'+(String)vec1.elementAt(2)+
'+(String)vec1.elementAt(3)+
'+(String)vec1.elementAt(4)));
```

Figure 5.8 Code for checking vehicle records

5.2.1.4 Update records sub module

This sub module required for the process of updating on whatever changes made in student and vehicle records.

```
String urlDrv = "http://localhost:8000/Saman/formProcessor?" +
"app="+app+"&action=UpdateDriver&initData="
+encrypt(p_ICno.getString()+" "+p_noTel.getString()+" "
+p_gender.getString()+" "+p_umur.getString()+" "
+p_nationality.getString()+" "
+p_noLicense.getString()+ "&data1="+encrypt(p_nama.getString()+
"&data2="+encrypt(p_alamat.getString()+ "&data3="+encrypt(p_addressKerj
a.getString()+
"&data4="+encrypt(p_J.getString()+ "&data5="+encrypt(p_tamat.getStrin
g());
String urlVec = "http://localhost:8000/Saman/formProcessor?" +
"app="+app+
"&action=UpdateVehicle&initData=" +encrypt(k_regno.getString()+
"+k_noSiri.getString()+
"&data1="+encrypt(k_type.getString()+ "&data2="+encrypt(k_color.getStri
ng()+ "&data3="+encrypt(k expiry.getString());
```

Figure 5.9 Code for updating student and vehicle records

```
alert = new Alert("Success",
"Updating Successful!", duke,
AlertType.INFO);
display.setCurrent(alert, formSrch);
```

Figure 5.10 Code displaying successful update process

```
stmt = con.prepareStatement("UPDATE student SET name = ?, noTel =
?, address = ?, programme = ?, gender = ?, age = ?, nationality
= ? WHERE ICno = ?");
stmt.setString(1, str[7]);
stmt.setString(2, str[1]);
stmt.setString(3, str[8]);
stmt.setString(4, str[9]);
stmt.setString(5, str[2]);
stmt.setString(6, str[3]);
stmt.setString(7, str[4]);
stmt.setString(8, str[0]);
stmt.executeUpdate();
```

Figure 5.11 Code for updating student information at server side

5.2.2 Web portal implementation

Web portal application can be implemented by system administrator and security officer as normal user using Internet connection. Users are required to register in order for them to use the system as authorized by system administrator. This web portal provides several administration functions to the system administrator also allowing other user to update personal information and to view summons records as well as student and vehicle records.

5.2.2.1 User registration sub module

This sub module is specifically for system administrator. Each and every user will have to register into the system to get their user ID and password before they can use the MIDlet application.

```
stmt = con.prepareStatement("INSERT INTO user  
VALUES(?,?,?,?,?,?,?,?)");  
stmt.setString(1, request.getParameter("Staffid"));  
stmt.setString(2, request.getParameter("ICno"));  
stmt.setString(3, request.getParameter("name"));  
stmt.setString(4, request.getParameter("postcode"));  
stmt.setString(5, request.getParameter("address"));  
stmt.setString(6, request.getParameter("hired date"));  
stmt.setString(7, request.getParameter("noTel"));  
stmt.setString(8, request.getParameter("email"));  
stmt.setString(9, request.getParameter("dept"));  
stmt.executeUpdate();
```

Figure 5.12 Code for User Registration

5.2.2.2 User profile sub module

By using this sub module, users will be allowed to view and edit their own personal information. Each and every sub module inside web portal will require below codes implementation.

```
<%@ page import="java.util.*, staff.UserProfile" %>
<%
UserProfile member =
(UserProfile)session.getAttribute("UserProfile");
if (member == null)
{
%>
<jsp:forward page="SignIn.jsp"/>
<%}
%>
```

Figure 5.13 Code for User Registration

```
public void setICno(String ICno)
{
this.ICno = ICno;
}
public String getICno()
{
return ICno;
}
public void setName(String nama)
{
this.name = name;
}
public String getName()
{
return name;
}
public void setPostcode(String postcode)
{
this.postcode = postcode;
}
```

Figure 5.14 Code for UserProfile

```

<tr>
<td width="25%" valign="top" class="forTexts"><div align="left">Staff ID</div></td>
<td colspan="2" valign="top" class="forTexts"> <div align="left"><%=
member.getIdStaff() %></div></td>
</tr>
<tr>
<td width="25%" valign="top" class="forTexts"><div align="left">No
KP </div></td>
<td colspan="2" valign="top" class="forTexts"> <div align="left"><%=
member.getICno() %></div></td>
</tr>

```

Figure 5.15 Code for displaying user information

```

stmt = con.prepareStatement("UPDATE staff SET ICno = ?, name = ?,
Postcode = ?, address = ?, hireddate= ?, noTel = ?, email
= ?, dept = ? WHERE idStaff = ?");
stmt.setString(1, request.getParameter("ICno"));
stmt.setString(2, request.getParameter("name"));
stmt.setString(4, request.getParameter("Postcode"));
stmt.setString(5, request.getParameter("address"));
stmt.setString(6, request.getParameter("hireddate"));
stmt.setString(7, request.getParameter("noTel"));
stmt.setString(8, request.getParameter("email"));
stmt.setString(9, request.getParameter("dept"));
stmt.setString(10, request.getParameter("idStaff"));
int success = stmt.executeUpdate();

```

Figure 5.16 Code for updating function

5.2.2.3 Student sub module

In this sub module, functions in which involving student such as view, update and delete student records. Thus below code is required in order to implement those functions.

```

stmt = con.prepareStatement("SELECT p.Mno, p.name, p.noTel, p.address,
p.Programme, p.gender, p.age, p.nationality,
m.noLesen, m.produced, m.validdate
FROM student p, maklLesen m WHERE p.Mno= ? AND
p.Mno*=m.Mno");
stmt.setString(1, srcDrv);
rs = stmt.executeQuery();
colCount = rs.getMetaData().getColumnCount();
driver = new Vector();
driver = oneColumnQuery(rs, colCount, driver, s);stmt.setString(8,
request.getParameter("email"));
stmt.setString(9,request.getParameter("jabatan"));
stmt.setString(10, request.getParameter("idStaff"));
int success = stmt.executeUpdate();

```

Figure 5.16 Code Student sub module

```

stmt1 = con.prepareStatement("SELECT Mno FROM InfolLesen WHERE
Mno= ?");
stmt1.setString(1, str[0]);
rs1 = stmt1.executeQuery();
colCount1 = rs1.getMetaData().getColumnCount();
v = new Vector();
v = oneColumnQuery(rs1, colCount1, v, s);
System.out.println (v.size());
if(v.size(>0)
{
stmt2 = con.prepareStatement("UPDATE InfolLesen SET
noLesen = ?, produced= ?, validdate = ?
WHERE Mno = ?");
stmt2.setString(1, str[5]);
stmt2.setString(2, str[10]);
stmt2.setString(3, str[6]);
stmt2.setString(4, str[0]);
stmt2.executeUpdate();
}else{
stmt2 = con.prepareStatement("INSERT INTO maklLesen VALUES(?, ?, ?,
?)");
stmt2.setString(1, str[5]);
stmt2.setString(2, str[0]);
stmt2.setString(3, str[10]);
stmt2.setString(4, str[6]);
stmt2.executeUpdate();
}
//CallableStatement cs = con.prepareCall("{call
updateUser(?,?,?,?,?,?,?,?)");
stmt = con.prepareStatement("UPDATE student SET name = ?, noTel =
?, address = ?, programme = ?, gender = ?, age = ?, nationality
= ? WHERE Mno = ?");

```

Figure 5.17 Code for updating student information

5.3 System Testing

System testing has been done by checking the operational logical programming. When programming codes have been compiled and free from any bugs or error then it has to be in the running process. *Start Tomcat* icon will be clicked in order to run Apache Tomcat 1.4.29. Web application testing process will be done by accessing Web browser for instance like Internet Explorer web browser. Then need to setting up the IP address of the system application which is *http://localhost:8000/Saman/index.jsp*. For MIDlet application testing process, Java Wireless Toolkit emulator will be used before installing the application inside the real PDA. The white box testing process will be involved in testing to each and every sub module.

For each of programming codes functions which are done, testing process will be implemented. This is important in order to identify any bugs or error and do the correction part and also to check the User Interface part of the system. Sub module testing is done by entering input value to the parameter on the user interface. Next the output value will be observed in order to check whether is there any bugs or errors of the programming code.

5.3.1 MIDlet application testing

Several important testing have been conducted for the MIDlet application such as connection testing and verifying user, entering summon testing, search testing and many more. Below shows several tests results conducted.

Test No	Testing	Expected result	Real result
1	User log in when there is no connection	Error message will be displayed and user will be given options to re-login or choos the offline mode	Error message will be displayed and user will be given options to re-login or choose the offline mode
2	User does not enter user ID & password	Error message will be displayed	Error message will be displayed
3	User enter correct user ID and password	Menu screen will be displayed	Menu screen will be displayed

Table 5.1 Shows connection and verifying user ID and password testing

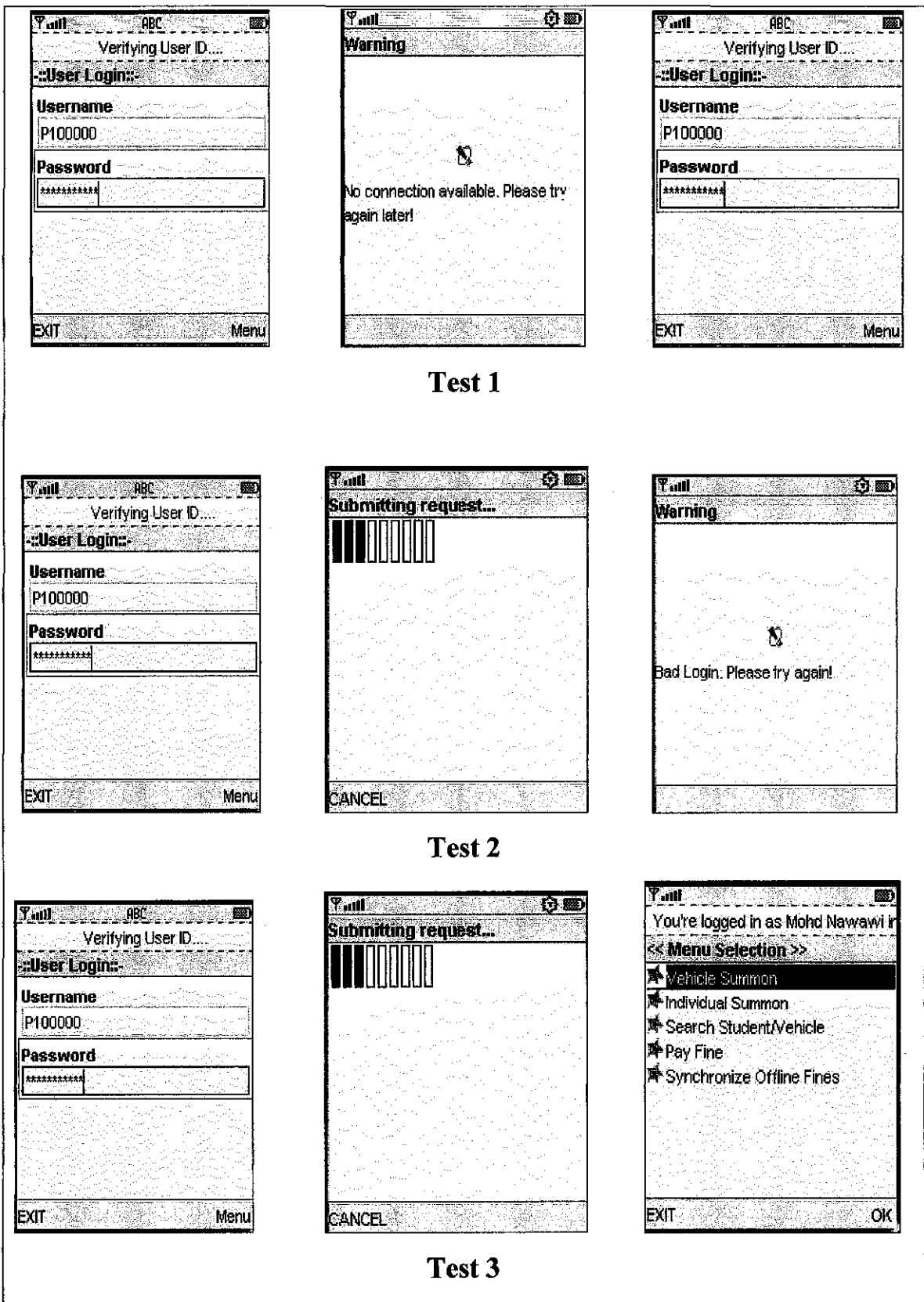


Figure 5.18 Shows connection and verifying user ID and password testing

Test No	Testing	Expected result	Real result
1	User enters the same serial number together with other summon records	Error message will be displayed	Error message will be displayed
2	User enters the correct serial number of summon	Successful message will be displayed	Successful message will be displayed

Table 5.2 Shows summon registration testing

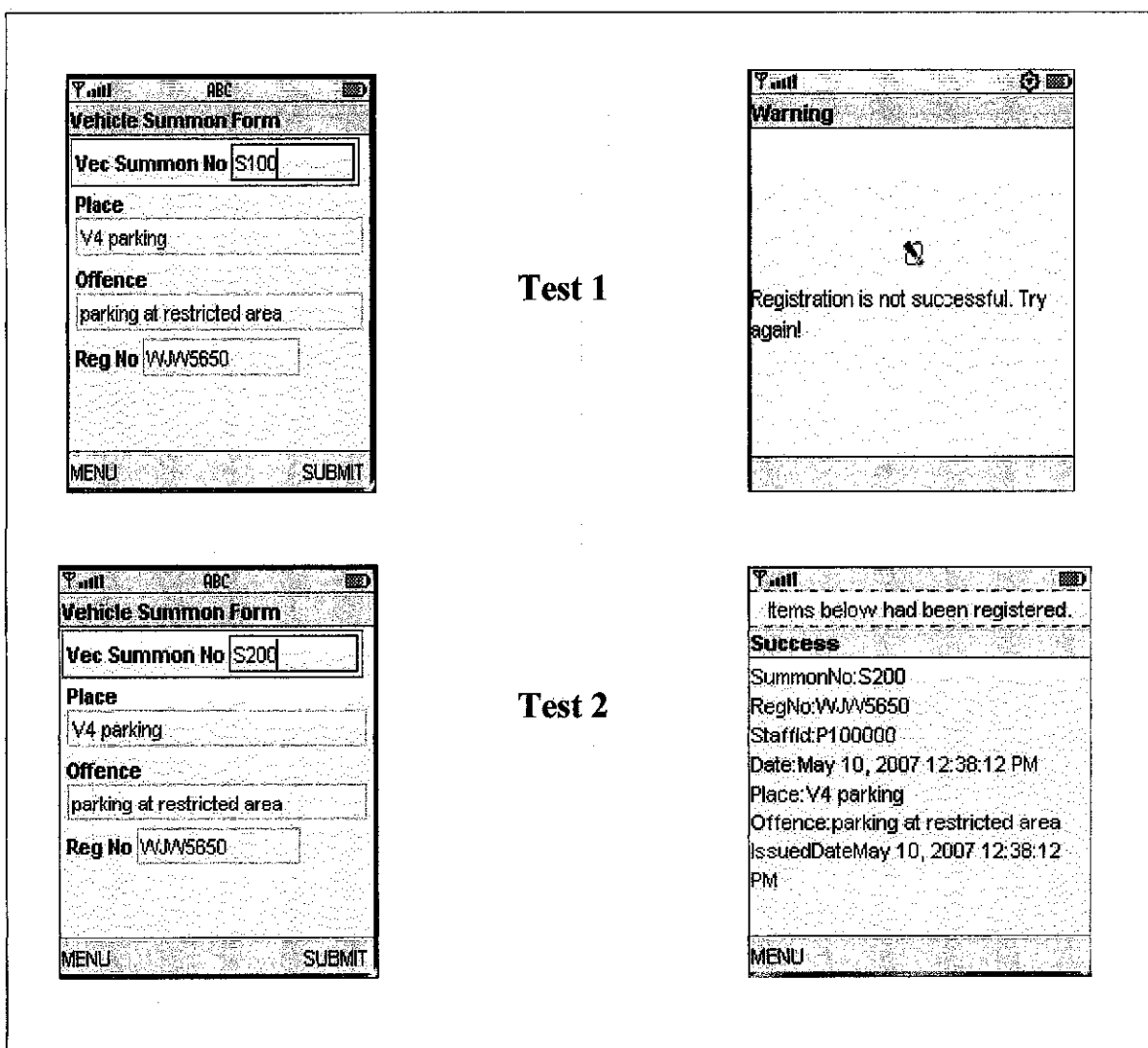


Figure 5.19 Shows summon registration testing

Test No	Testing	Expected result	Real result
1	User enters data which is not in the database records	Error message will be displayed	Error message will be displayed
2	User enters data which is in the database records	Search result will be displayed	Search result will be displayed

Table 5.3 Shows search testing

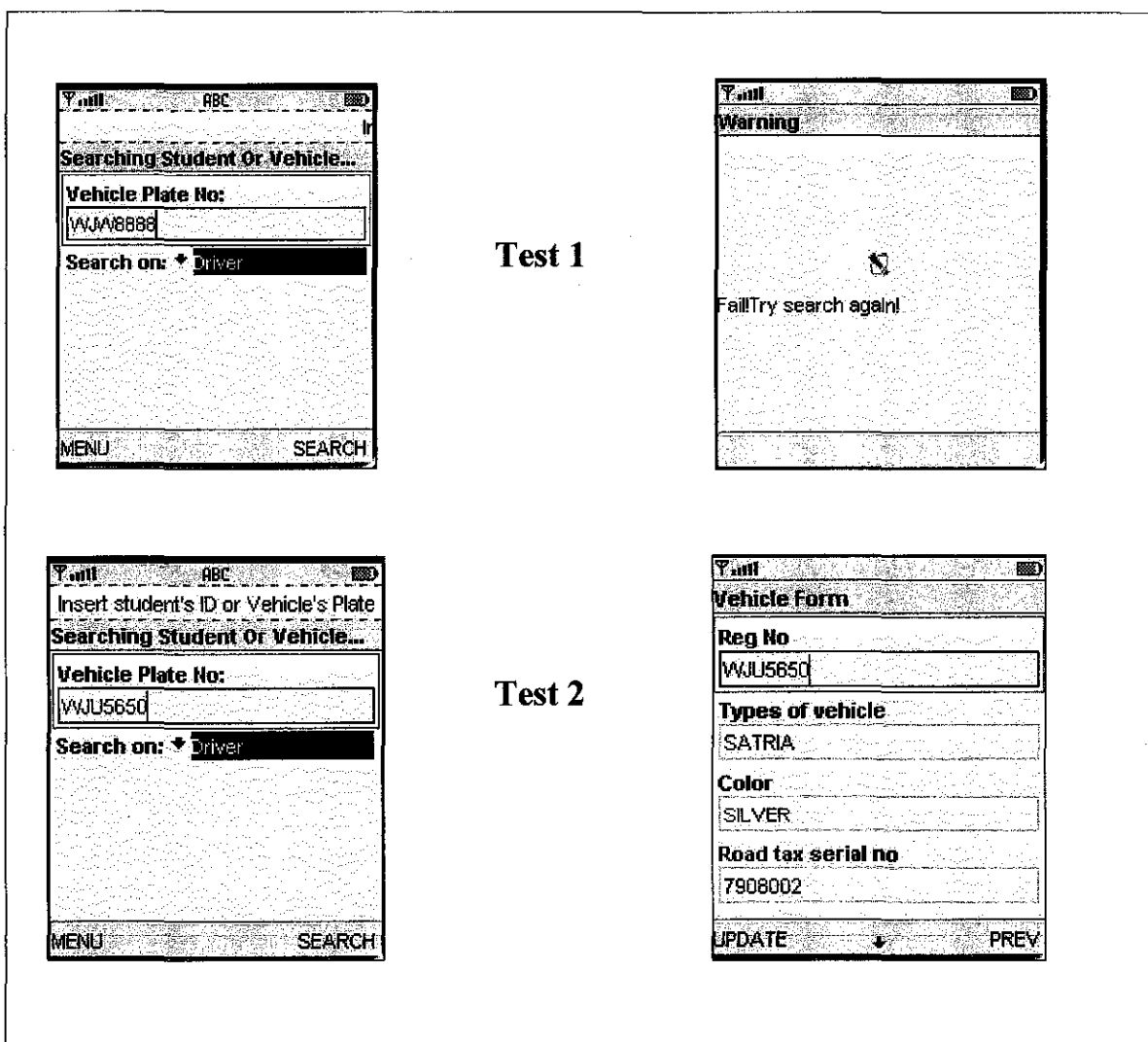


Figure 5.20 Shows search testing

5.3.2 Web application testing

Several important tests have been conducted for the Web application such as user registration testing, user verifying testing and search testing.

Test No	Testing	Expected result	Real result
1	User enters information which invalid	Error message will be displayed	Error message will be displayed
2	User enters information which is valid	My account page will be displayed	My account page will be displayed

Table 5.4 Shows User Registration testing

Figure 5.21 Shows User Registration testing

Test No	Testing	Expected result	Real result
1	User enters incorrect user ID and password	Error page will be displayed	Error page will be displayed
2	User enters the correct user ID and password	Main page will be displayed	Main page will be displayed

Table 5.5 Shows user verifying testing

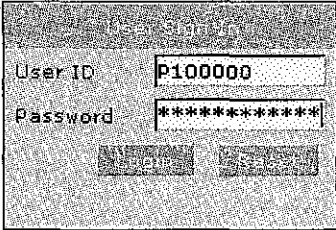
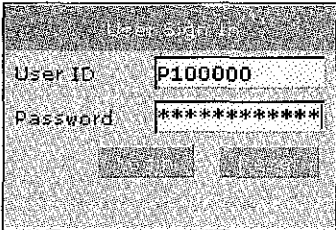
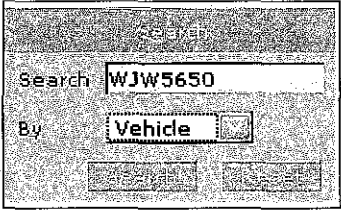
 <p>User ID: P100000 Password: *****</p>	<p>Test 1</p>	<p>Error page!</p> <p>#Incorrect user ID & password #Records could not be retrieve. No records in database.</p>
 <p>User ID: P100000 Password: *****</p>	<p>Test 2</p>	<p>Welcome! Mohd Nawaz</p> <p>You have successfully logged in within 0 mill second.</p> <p>My Account</p> <p>This system will allow system administrator to manage all summons information, vehicle information. Normal users can only view and display these records. For those you can send email to azkran99@hotmail.com</p>

Figure 5.22 Shows user verifying testing

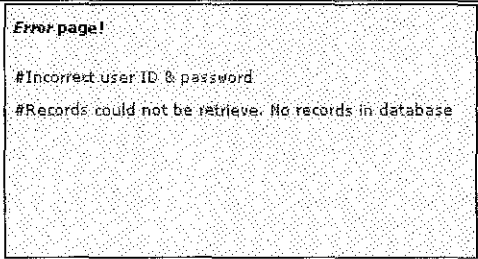
Test No	Testing	Expected result	Real result
1	User search for record which is not in the database	Error page will be displayed	Error page will be displayed
2	User search for record which is in the database	Search result page will be displayed	Search result page will be displayed

Table 5.6 Shows search testing

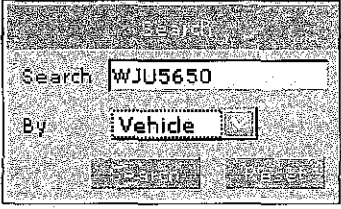


Search: WJW5650
By: Vehicle

Test 1

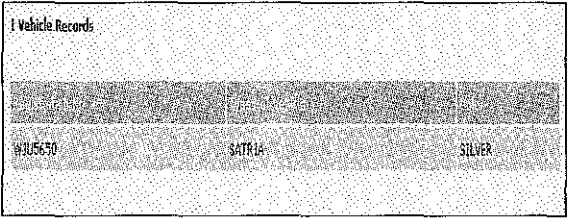


Error page!
#Incorrect user ID & password
#Records could not be retrieve. No records in database



Search: WJU5650
By: Vehicle

Test 2



1 Vehicle Records

WJU5650	SATRIA	SILVER
---------	--------	--------

Figure 5.23 Shows search testing

CHAPTER 6

6.0 RESULT & DISCUSSION

6.1 Introduction

Based on the final prototype of UTP Mobile Summon System, can be concluded I have been successfully developed the system prototype using J2ME technology for the MIDlet application and Servlet as well as by using Java Server Pages for the Web portal application. This system is separated between two groups of users which on the MIDlet application is for security officer while on the Web portal application consists of system administrator and also security officer who registered as normal user in order for them to be able to use with the system.

6.2 Result achievement

The web portal application has been developed first in order to support the implementation of the system. That will lead to the development of database for allowing all data to be saved and retrieve by the system. During the development phase, there are several development issues raised related to web portal and databases such as connection to the database at the web portal and also some other operational functions dealing with view, edit, update and delete records from databases. Then, MIDlet application development will take place once the web portal application has been successfully developed. It is one of the strategy in order to speed up development phase since development of MIDlet application is more complicated compared to web portal application in several issues raised such as limited memory size of PDA, limited display screen, processing speed and others.

After several tests conducted through this system, I found that this system managed to achieve its objectives which have been determined at the early stage such as provide with an alternative system for UTP security department to implement their daily job, enable user for transmitting summons information online as well as retrieving student and vehicles records online. Besides, kriptografi Bouncy Castle package is used for the security aspect of the system in order to secure the wireless transmitting of information.

6.3 System drawback

After several tests conducted to the system, there are several of drawbacks identified from this system. The main drawback would be the search and retrieving information is quite slow and some of the enhancements need to be done at the coding side. As this system is still fragile and there are a lot of room of improvement in order to get a smooth and solid running system for instance on the error displaying message perhaps can be enhanced in order to have better interaction with system user.

6.4 Future recommendation

As mentioned before, system developed is just to show that this could be an alternative to the current system running at UTP security department and I admitted that it is still fragile and there is a lot of room of improvement in order to make this system run smoothly and efficiently in the future. Several examples of improvement could be at the searching part, the error displaying message and system user interface so that each and every user will managed to get better respond and good interaction between user and the system.

REFERENCES

Vikas Gupta, Avnish Dass, Yashraj Chauhan (2002), "Wireless Programming with J2ME™", Hungry Minds, Inc.

H. M. Deitel, P. J. Deitel, T.R. Neito (2002), "Internet & World Wide Web How To Program", Deitel & Associates, Inc.

Kenneth E. Kendall, Julie E. Kendall (2002), "Systems Analysis and Design", Prentice-Hall International, Inc.

Stephanie (2002). "Java 2 Platform, Micro Edition"
<http://java.sun.com/software>

HowStuffWorks, Inc. "How PDAs Work" in <http://www.howstuffworks.com>

Java Essentials: What Is Wireless Java? By Steve Anglin 11/06/2002
<http://www.onjava.com/pub/a/onjava/2002/11/06/wireless.html>

Software Engineering 6th Edition, Ian Sommerville
Addison Wesley Publishers Limited 1989, 2001

PURE Java Server pages (JSP), A Code-Intensive Premium Reference (2000) By James Goodwill, Published by Techmedia

Cracking the Code, Wireless Programming with J2ME by Dreamtech Software Team

Other sources:

<http://www.cites.uiuc.edu/101/wireless101.html>

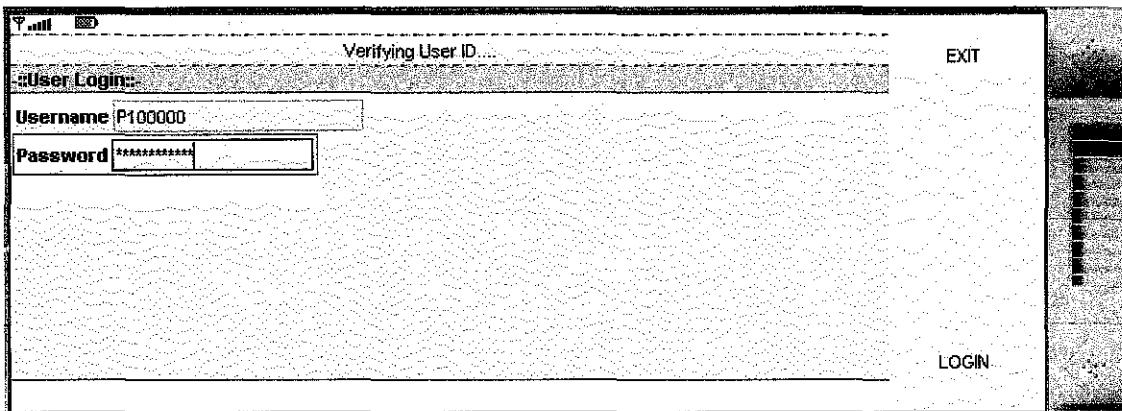
http://www.devbuzz.com/content/zinc_wireless_what_is_pg1.asp

<http://www.computerworld.com/developmenttopics/development/story/0,10801,71151,00.html>

User Manual

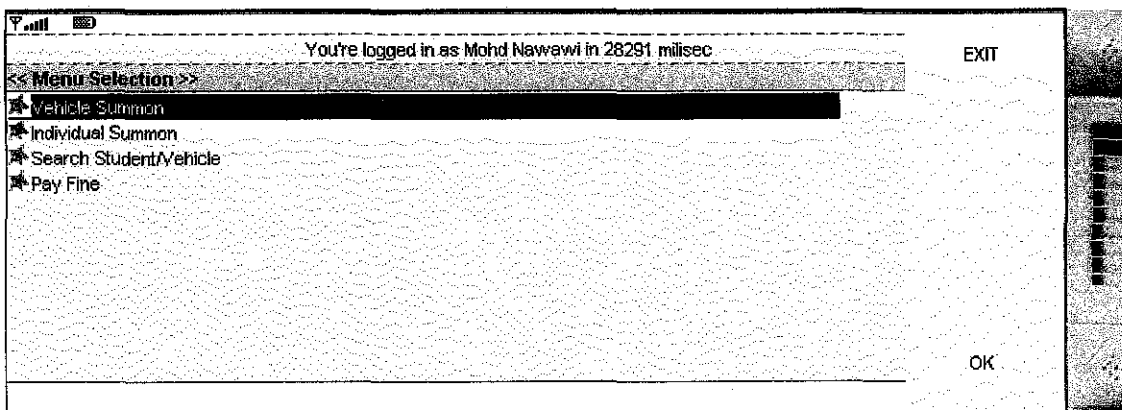
Display Screen – MIDlet Application

MIDlet application will be running on the user PDA. Inside the application, user will log in and will be able to search and edit students and vehicles records, submit vehicle and individual summon. User will insert username and password given and then press *LOGIN* button or *EXIT* button to exit the system as shown below:



Screenshot for User Login

Once the Internet connection is available, then Menu Selection screen will appear indicates that the log in process is successful. Here user will be given a selection to be taken such as Vehicle Summon, Individual Summon, Search, also Pay Fine.



Screenshot for Menu Selection

In order to insert information on particular summon, for instance Vehicle Summon, user need to fill in particulars such as registration number, place, offence and summon number as shown below. Then press *SUBMIT* button to save the information or press *MENU* button to go back to Menu Selection.

Vehicle Summon Form		MENU
Vec Summon No	VS1001	
Place	Village 5E	
Offence	parking at restricted area	
Reg No	WJW2222	
		SUBMIT

Screenshot for Vehicle Summon Form

Once user has fill in all the information required and the submission is successfully registered, screen as below will be display.

Items below had been registered.		MENU
Success		
SummonNo:	VS1001	
RegNo:	WJW2222	
StaffId:	P100000	
Date:	May 12, 2007 4:55:10 AM	
Place:	Village 5E	
Offence:	parking at restricted area	
IssuedDate:	May 12, 2007 4:55:10 AM	

Screenshot for successful submission

Search function will be divided into 2 which are driver (student) and vehicle. For instance, if user selecting by vehicle, then registration number of that particular vehicle needs to be inserted as below.

Y.mil	
Insert student's ID or Vehicle's Plate Number!	
Searching Student Or Vehicle...	
Vehicle Plate No:	WJL5650
Search on:	Driver
MENU	
SEARCH	

Screenshot for Search on vehicle

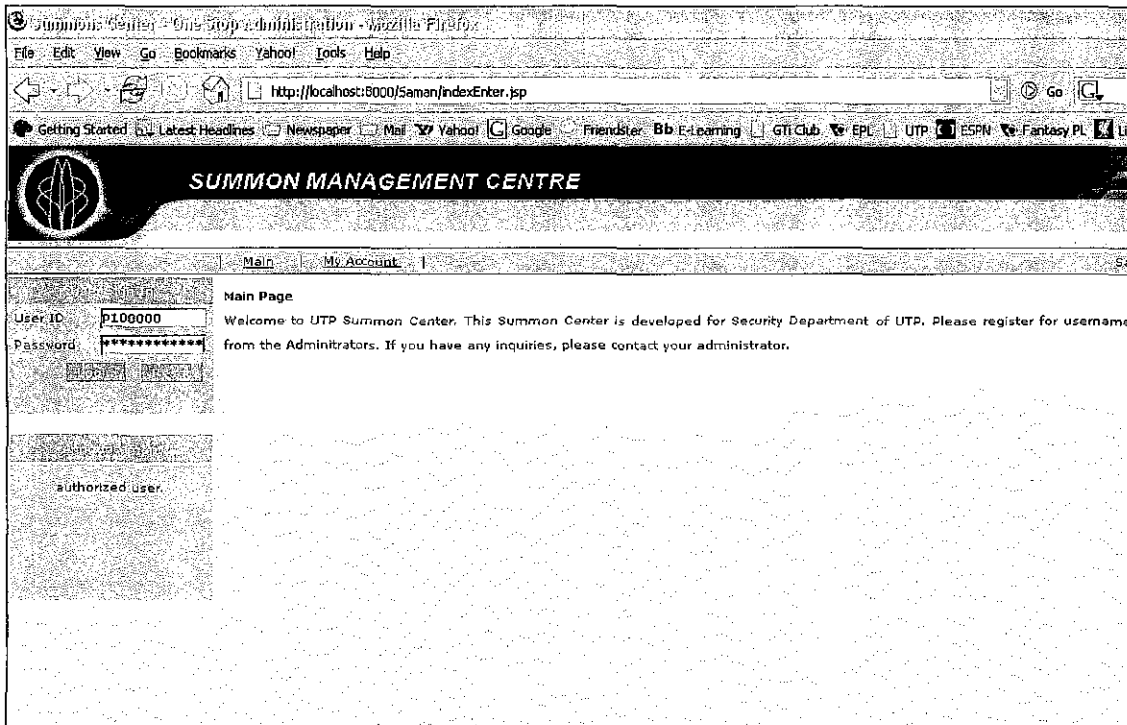
If the records exist in the database and the retrieving process successful, then screen as below will be display.

Y.mil	
Vehicle Form	
Reg No	WJL5650
Types of vehicle	SATRIA
Color	SILVER
Road tax serial no	7908002
Expiry date	2008-08-02 00:00:00.000
UPDATE	
PREV	

Screenshot for search result on vehicle

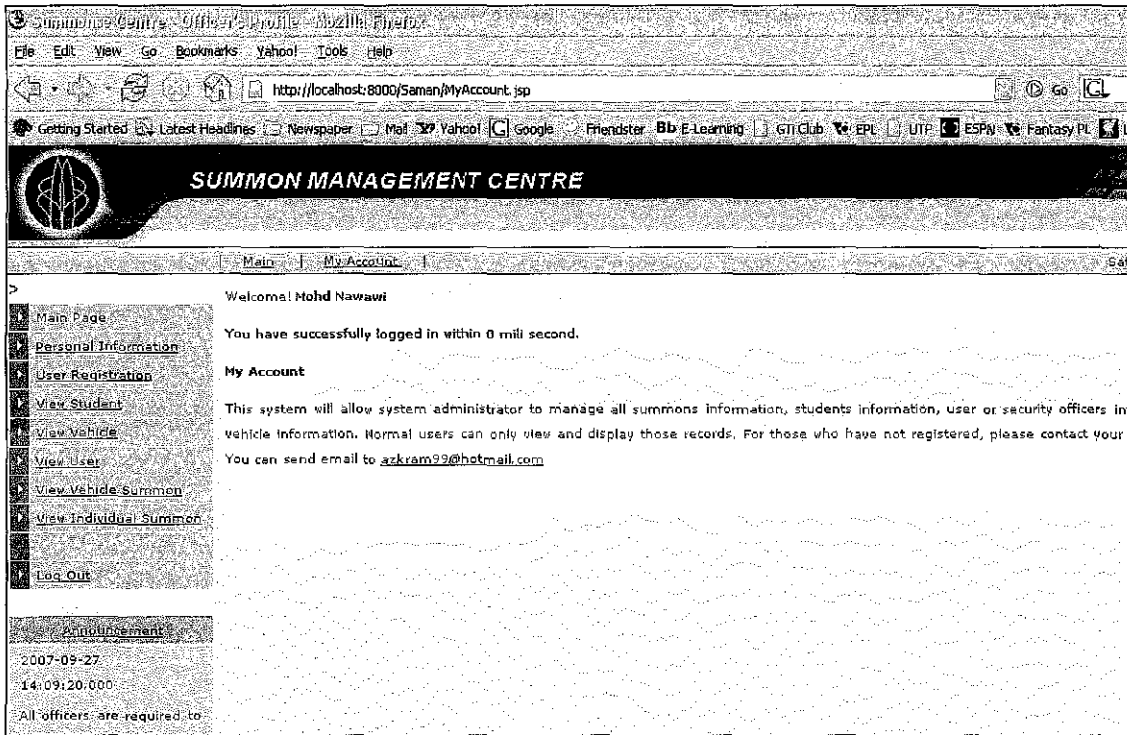
Display Screen – Web Application

On the Web Application, user will require to sign user ID and password. This is to differentiate between normal user and system administrator as shown below. Once user has fill in user ID and password, then press *Login* button.



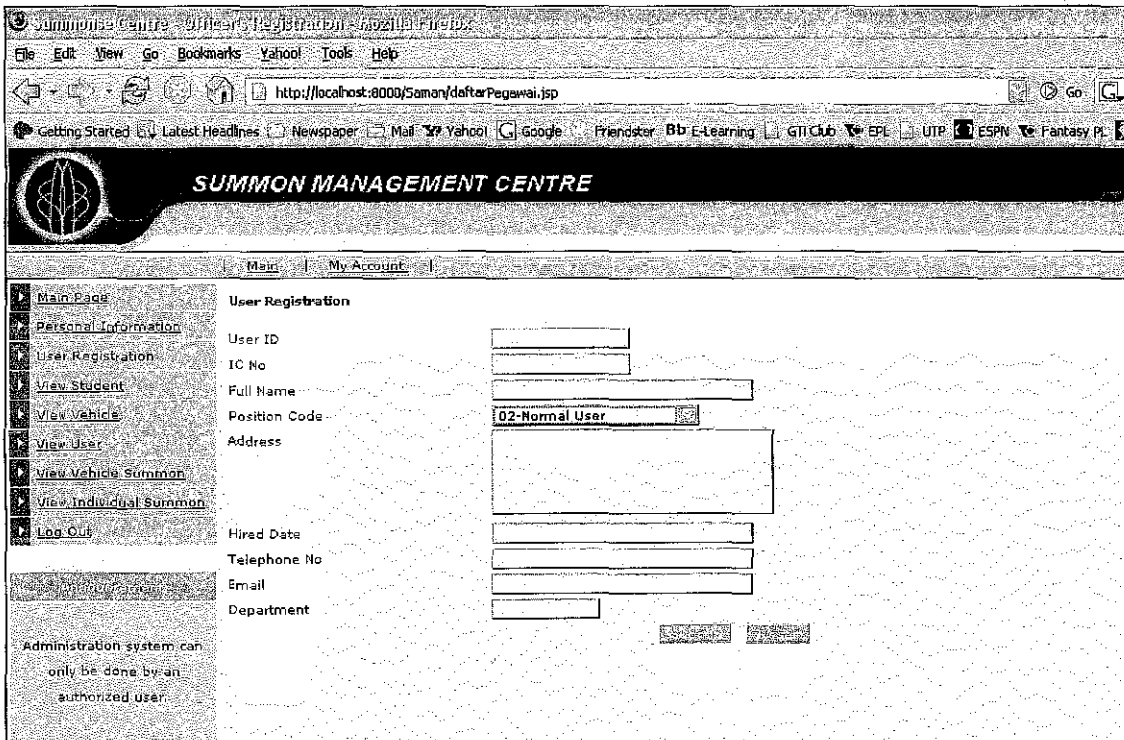
Screenshot for User Sign In

If the user ID and password entered is correct, then screen as below will appear indicates that user has successfully logged in the system and for example given as system administrator.



Screenshot for available functions for System Administrator

Only system administrator will be given authorization to register new user as shown below unlike normal user the User Registration function will be hidden.



Screenshot for User Registration

Administration Centre - Official Profile - Mozilla Firefox

File Edit View Go Bookmarks Yahoo! Tools Help

http://localhost:8000/Samanypol170A.jsp

Getting Started Latest Headlines Newspaper Mail Yahoo! Google Friendster Bb E-Learning GTI Club CPL UTP ESPN Fantasy Pl Live

SUMMON MANAGEMENT CENTRE

Main My Account Sat 1

- Main Page
- Personal Information
- User Registration
- View Student
- View Vehicle
- View User
- View Vehicle Summon
- View Individual Summon
- Log Out

3 Vehicle Summon Records

Vehicle ID	Registration No	Plate No	Registration Date	Location	Area	Time	Status
VS1001	WJW 2222	P100000	2007-05-12 04:55:10.000	village SE	parking at restricted area	2007-05-12 04:55:10.000	null
VS100	WJU5650	P100000	2007-04-24 07:35:45.000	Village S C	parking at restricted area	2007-04-24 07:35:45.000	null
S200	WJW5650	P100000	2007-05-10 12:38:12.000	V4 parking	parking at restricted area	2007-05-10 12:38:12.000	null

Administration system can only be done by an authorized user.

Screenshot for viewing vehicle summon records