

**WIRELESS MAINTENANCE REPORTING SYSTEM  
(WMRS)**

by

Wan Non Zarina Bt Wan Abd Ghani

Dissertation Report in partial fulfillment of  
the requirement for the  
Bachelor of Technology (Hons)  
Business Information Systems

JANUARY 2007

Universiti Teknologi PETRONAS  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan

6  
TK  
5105.586

.WZ44

2007

1) Wireless communication system

**CERTIFICATION OF APPROVAL**

**WIRELESS MAINTENANCE REPORTING SYSTEM**

By

Wan Non Zarina Bt Wan Abd Ghani

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

Approved by,



(Mr. Low Tang Jang)

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK  
JANUARY 2007

## TABLE OF CONTENTS

<b>CERTIFICATION</b>	.	.	.	.	.	.	.	.	.	<b>i</b>
<b>ABSTRACT</b>	.	.	.	.	.	.	.	.	.	<b>ii</b>
<b>ACKNOWLEDGEMENT</b>	.	.	.	.	.	.	.	.	.	<b>iii</b>
<b>CHAPTER 1:</b>	<b>INTRODUCTION</b>									
	1.1	Background of Study	.	.	.	.	.	.	.	1
	1.2	Problem Statement	.	.	.	.	.	.	.	2
	1.3	Objectives and Scope of Study	.	.	.	.	.	.	.	3
<b>CHAPTER 2:</b>	<b>LITERATURE REVIEW AND THEORY</b>									
	2.0	Wireless Technology	.	.	.	.	.	.	.	4
	2.1	Mobile Application Types	.	.	.	.	.	.	.	4
	2.2	Wireless Application Protocol (WAP)	.	.	.	.	.	.	.	5
	2.2.1	WAP Architecture	.	.	.	.	.	.	.	5
	2.2.2	Wireless Markup Language (WML)	.	.	.	.	.	.	.	6
	2.3	Java 2 Micro Edition (J2ME)	.	.	.	.	.	.	.	6
	2.3.1	Introduction to J2ME	.	.	.	.	.	.	.	6
	2.3.2	Java and J2ME	.	.	.	.	.	.	.	6
	2.4	WAP and J2ME	.	.	.	.	.	.	.	8
	2.4.1	Mobile Information Device Profile (MIDP)	.	.	.	.	.	.	.	9
	2.4.2	MIDP Development and Deployment	.	.	.	.	.	.	.	10
	2.4.3	Sun Java Wireless Toolkit	.	.	.	.	.	.	.	11
	2.5	MySQL Database	.	.	.	.	.	.	.	12
	2.5.1	Java Database Connectivity (JDBC)	.	.	.	.	.	.	.	15
	2.6	Database Architecture Prototype	.	.	.	.	.	.	.	15

<b>CHAPTER 3:</b>	<b>METHODOLOGY</b>	
	3.0 Methodology and Project Work . . . . .	18
	3.1 System Development Life Cycle (SDLC). . . . .	18
	3.1.1 Planning Phase . . . . .	18
	3.1.2 Analysis Phase . . . . .	19
	3.1.3 Design and Coding . . . . .	19
	3.1.4 Implement the Emulator Testing . . . . .	19
	3.2 Tools and Equipments . . . . .	20
	3.2.1 Software and Hardware Requirements . . . . .	20
<b>CHAPTER 4:</b>	<b>RESULTS AND DISCUSSION</b>	
	4.0 Results and Discussion . . . . .	21
	4.0.1 Wireless Maintenance Reporting System . . . . .	21
	4.0.2 Discussion . . . . .	23
<b>CHAPTER 5:</b>	<b>CONCLUSION AND RECOMMENDATION</b>	
	5.0 Conclusion and Recommendation. . . . .	30
<b>REFERENCES</b>	. . . . .	31
<b>APPENDICES</b>	. . . . .	33

## **LIST OF FIGURES**

- Figure 1: WAP Network Structure
- Figure 2: Java end-to-end Review
- Figure 3: Hello.java
- Figure 4: MIDP application Development
- Figure 5: Wireless Toolkit
- Figure 6: Class Diagram
- Figure 7: Use Case Diagram
- Figure 8: Draft Architecture
- Figure 9: SDLC
- Figure 10: WMRS Main Screen
- Figure 11: User Details
- Figure 12: Defect Form
- Figure 13: WMRS Workflow
- Figure 14: Welcome screen
- Figure 15: Welcome Screen
- Figure 16: Main Menu
- Figure 17: Staff Information
- Figure 18: Query Result screen
- Figure 19: Main Menu
- Figure 20: Defect Form
- Figure 21: Query Result screen
- Figure 22: Updated Database

## **LIST OF TABLES**

- Table 1: user
- Table 2: contractor
- Table 3: report

## **LIST OF ABBREVIATIONS**

<b>WMRS</b>	Wireless Maintenance Reporting System
<b>C&amp;S</b>	Civil and Structure
<b>UTP</b>	Universiti Teknologi Petronas
<b>J2ME</b>	Java Platform, Micro Edition
<b>WAP</b>	Wireless Application Protocol
<b>HTTP</b>	Hypertext Transfer Protocol
<b>CLDC</b>	Connected Limited Device Configuration
<b>CDC</b>	Connected Device Configuration

## 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 acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



---

**WAN NON ZARINA BT WAN ABD GHANI**

## **ABSTRACT**

This project proposed how the mobile application can be implemented through the Wireless Application Protocol (WAP) for Wireless Maintenance Reporting System (WMRS) in UTP. WMRS enables user to send report to maintenance department by using their mobile phone. The report from the user can be transferred through the telephone service provider and then links it to the Internet and finally reaching the maintenance department server. This project was developed using J2ME technology as the main development platform. The outcome will focus on the implementation of WMRS in the UTP Maintenance Department.

This project applied the Software Development Life Cycle (SDLC) which consists of planning phase, analysis phase, design, coding phase and implementing the testing through emulator.



## **ACKNOWLEDGEMENT**

First of all, I would like to express our heartiest gratitude, first and foremost to Allah, The Al-Mighty for giving me the strength and ability to carry out all the research and task required. I never would have completed this project without the help of many people. I convey my highest appreciation to my project supervisor, Mr Low Tang Jung for his time in consulting and facilitating me throughout my final year project. I also would like to thank officer and staff on Maintenance Department specifically for their help in giving me information about maintenance report. Without their information, I will know nothing about how the process of reporting problems being made. Most of all, I would like to thank my family because without their support, I never could completed this project.

Thank You.

# CHAPTER 1

## INTRODUCTION

### 1.0 INTRODUCTION

#### 1.1 Background of Study

The academic complex at UTP consists of 16 buildings out of 23 planned general buildings, and 2 out of 4 lectures and administration buildings. These are adequate for the current requirements of the university. Each general building belongs to one faculty, and all the buildings have similar design. The first three floors consist of tutorial rooms or laboratories. The topmost floor holds the lecturer offices for the respective faculties. There are 2 lecture or administrative buildings (Pocket C and Pocket D) consist of a collection of lecture theaters.

All maintenance work or processes are handled by Maintenance Department. Any reports about the damages in the buildings will be reported to Maintenance Department. In Maintenance Department, they have 4 specialized sections namely Civil and Structure(C&S), mechanical, electrical and landscape. For this project, only the C&S will be covered. The C&S section will handle the damages reported by users, such as leak ceiling, damage door and panel, leak pipe, broken mirror and others. The users usually report to Maintenance Department by telephone or email. Then the Maintenance Department will send the technicians to the location to fix the damages accordingly. The maintenance department benchmark is to fix the problems within 24 hours. Currently most of the flows are handled manually.

Based on this situation, the idea of converting the manual reporting system to wireless reporting system has resulted to the **Wireless Maintenance Reporting System (WMRS)**.The main advantage is to reduce the use of paper base report and to give users more flexibility in reporting any buildings problems to

Maintenance Department. WMRS enable user to send report on building damages through mobile phone.

The report will send directly to the server and it will alert the Maintenance Department. The user also will be notified about the status of their message whether it already successfully received by the Maintenance Department through delivery message. Once the Maintenance Department receives the message sent by the user, the Maintenance Department will send the technician to fix the problem. After the problems have been fixed, technician will send reports of completion through their mobile phone to the database and to acknowledge the person who had reported the damages.

## **1.2 Problem Statement**

Currently, the Maintenance Department does not have database to store all the reports. Its all paper base reports and they put them in files. This way of storing reports is not efficient and they may hinder the search of former reports quickly and effectively.

The department does not have the standard way of reporting method by the user. User can report the damage by email, phone or control room. The report made by the user is very important as they will use it as prove that the damage is really occurred at the building.

The **Wireless Maintenance Reporting System (WMRS)** can reduce the use of paper as it will update the records of details through wireless mode. With **WMRS**, the user can report to Maintenance Department anywhere and anytime. The technician can also send the report wirelessly and this application may help them to save time.

### **1.3 Objectives and Scope of study**

The objectives and scope of project to be achieved are as follows:

- **Main Objectives**
  - To create a wireless maintenance reporting system for UTP to increase efficiency in maintenance handling and reporting.
  - To provide a wireless solution towards a paperless maintenance process flow.
  
- **Scope of Study**
  - To create a wireless maintenance reporting system for Civil and Structure Section of the UTP Maintenance Department.

## **CHAPTER 2**

### **LITERATURE REVIEW AND THEORY**

#### **2.0 Wireless Technology**

Wireless communication operates without a bounded media such as wires or fiber optics cable. The wireless technology had given many benefits in many situations, for example, a monitoring station on a ship at sea. This need connectivity with the outside world, however if we deploy traditional wired technologies it would provide an expensive solution to this problem and also may damaged the environment. With wireless, it gives greater flexibility and mobility for users where they can do transactions anywhere and anytime. This project involves wireless technology whereby mobile phone is use as a device to send information to the server.

#### **2.1 Mobile Application Types**

Mobile applications can be divided into two categories, which is voice and messaging services and general applications. For this project, the author will focus only on general applications. General applications include personal information software, intriguing games, and many more varieties of mobile software. These applications come in two flavors;

- Browser based, typically using the Wireless Application Protocol (WAP) or NTT DoCoMo's i-mode
- Downloaded, created by developers who may base an application on devices' own proprietary interfaces, on Java Platform, Micro Edition (J2ME) or on the Binary Runtime Environment for Wireless (BREW).

## 2.2 Wireless Application Protocol (WAP)

Wireless application Protocol or WAP is an open standard that enables easy delivery of information and services to mobile users [1]. It is a set of communication protocol designed to enable different kind of wireless device to communicate and access the internet. It is intended primarily for internet enabled digital phones, pagers and other handheld devices. The website specifically designed for wireless handheld that have small screens and low bandwidth constraints.

### 2.2.1 WAP Architecture

Any WAP enable system consists of WAP Gateway, HTTP web server and the WAP device. WAP gateway is a 'middleman' between mobile phone and HTTP web server. It is a proxy server which acts between the user agent and the web server. The gateway can be located either in a telecom network or in computer network (ISP). The HTTP web server will receive the request from WAP gateway and process the request before sending the output to the WAP gateway, which in turn it sends the information to the WAP devices (mobile phone) using its wireless network. Figure 1 illustrates an example structure of WAP network.

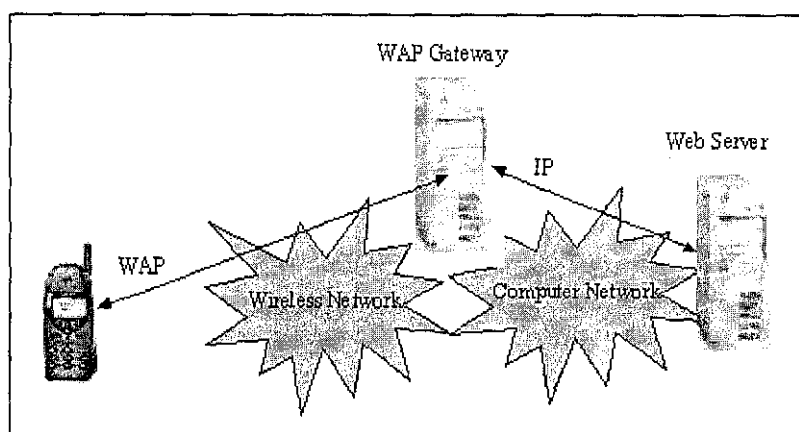


Figure 1: WAP Network Structure

The WAP device (mobile phone) is part of wireless network. WAP device sends the request to WAP gateway, which in turns translates WAP requests to WWW requests, so WAP client is able to submit requests to the web server. After received the response from HTTP web server, WAP gateway translates web response into WAP response or a format understood by the WAP client and send it to WAP device.

## **2.3 Java 2 Micro Edition (J2ME)**

### **2.3.1 Introduction to J2ME**

J2ME stands for Java 2 Micro Edition; it is a platform which provides a means of creating applications, and access to data through a wireless network. J2ME is a version of the Java platform that is optimized for resource limited devices. As the project specification requires the application to be developed for mobile devices, J2ME was chosen for reasons detailed later in this report.

### **2.3.2 Java and J2ME**

Java 2 Micro Edition (J2ME) platform will be use as it consists of a set of technologies and specifications developed for small devices like PDAs, mobile phones, pagers and other consumer electric and embedded devices [2]. The other Java programming environments are Java 2 Enterprise Edition (J2EE platform), Java 2 Standard Edition (J2SE) and Java Card (for SIM cards). Below is the figure of Java end-to-end review [3];

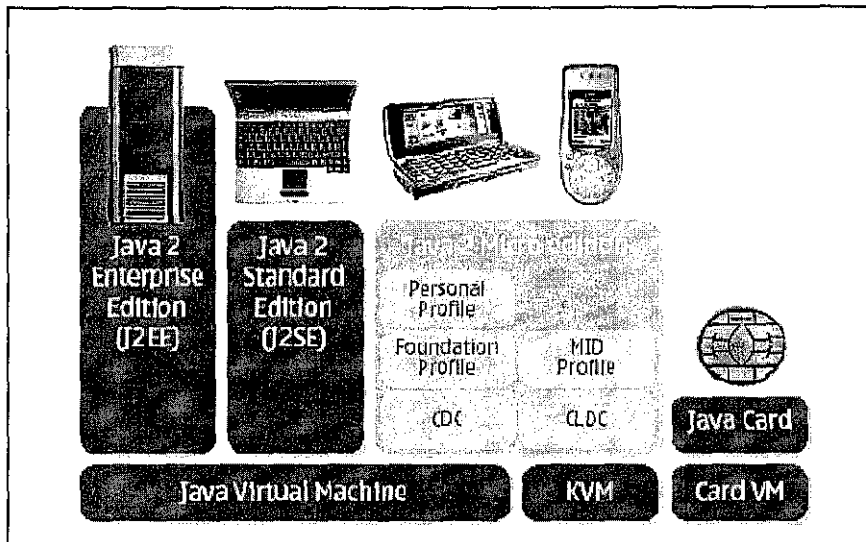


Figure 2: Java end-to-end Review

The reason why of using the Java platform for wireless device development is that we are able to produce portable code that can run on multiple platforms. Java is one of the most widely used programming languages in the world. It is particularly appropriate for computers implementing internet-based and intranet-based applications and any other software for devices that communicate over the networks, including cell phones, pagers and PDA [4].

J2ME is divided into configurations, profiles and optional APIs (Application Programming Interface) which provide specific information about APIs and different families of devices [5]. The overall purpose of configurations and profiles is to have virtual machine and class libraries optimized for each group.

There are 2 types of configuration which are, Connected Limited Device Configuration (CLDC) and Connected Device Configuration (CDC). Configuration defines the minimum Java technology that an application developer can expect on a broad range of implementing devices. CLDC is the configuration that interests us as it encompasses mobile phone, PDA and it aimed a smaller device than CDC. CLDC devices usually wireless and it needs 160k-512k of memory available for Java. Moreover, typically it has limited power or battery operated.



The J2ME's profile is to define additional sets of APIs and features for particular device category. Mobile Information Device Profile (MIDP) is a CLDC profile that provide user interface, persistence storage, networking and application model APIs for wireless devices such as low-end and two-way pagers [6]. An application on a MIDP device is a MIDlet which is more like Java Applet. A MIDlet is a class that extends the class `javax.microedition.midlet`.

The CLDC virtual machine is far smaller than that required by the CDC and supports fewer features. The virtual machine for the CLDC is called Kilo Virtual Machine (KVM).

## **2.4 WAP and J2ME**

WAP is a markup language similar to HTML in the internet world. WAP is a technology that supports a minimal web browser on a mobile device. The main difference between J2ME and WAP is that WAP is not an application, but it is a protocol.

WML, WAP's markup language works well for certain kind of application, such as text-centric application like text weather reports, stock quotes, etc. But there are a number of weaknesses with WAP which Java technology can overcome. WAP-based application require a constantly available network connection, since the applications itself reside on a server and require frequent use of wireless connection.

J2ME applications have a clear benefit in terms of user interface. WAP is fairly constrained in terms of the user interface; J2ME presents a much greater flexibility in user interface. The API's in J2ME allow easy to use and interesting interfaces to be developed.

### 2.4.1 Mobile Information Device Profile (MIDP)

When used in conjunction with the Connected Limited Device Configuration, MIDP provides a standard run time environment for the most popular Java enabled portable information devices such as mobile phones. The specification for MIDP was designed by leading device manufacturers, wireless carriers and software vendors.

The goal of the MIDP profile specification was to provide a complete set of APIs and a properly configured Virtual Machine so that applications written using this specification would have the greatest compatibility possible with as many devices in the mobile market as is possible. Essentially, if an application was built and runs to MIDP specification it should run on any phone certified for J2ME and MIDP 2.0 compliant.

The MIDP programming model is a mix of Java programming model and web programming model. MIDlets are developed using Java and compiled with the same way we compiled any Java application. Below are the sample of MIDlet as shown in Figure 3.

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class HelloMIDlet extends MIDlet {
    // The display for this MIDlet
    private Display display;
    // TextBox to display text
    TextBox box;

    public HelloMIDlet() {
    }
    public void startApp() {
        display = Display.getDisplay(this);
        box = new TextBox("First Example", "This is my 1st example", 20, 0);
        display.setCurrent(box);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
}
```

Figure 3: Hello.java

In a sense, Java technology is very good at many of the things in which WAP is lacking. WAP and J2ME MIDP are not competing technologies; in fact it is complementary technologies [7].

#### 2.4.2 MIDP Development and Deployment

There are a few steps to take a MIDP Java file from creation to deployment. The first step is to write the MIDP application (Figure 4). For this project, author use text editor to create Java program. JAD file also needs to be created but this is done using IDE such as Wireless Toolkit. Next is to compile .java file using the javac command which will produce a .class file. The .class file needs to preverify to make it usable on J2ME device. Then, package the verified .class file into a JAR (.jar) file. Lastly, deploy the JAR file and JAD file to the device.

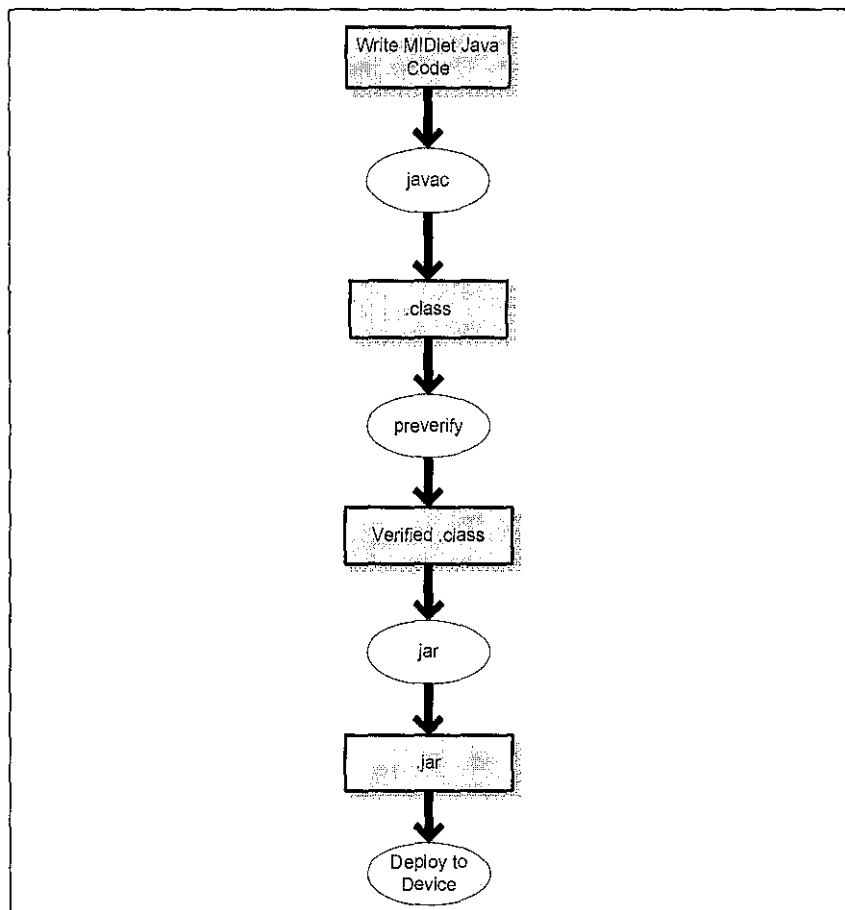


Figure 4: MIDP application Development

### 2.4.3 Sun Java Wireless Toolkit

The Sun Java Wireless Toolkit is a state-of-the-art toolbox for developing wireless applications that are based on J2ME's Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP), and designed to run on cell phones, mainstream personal digital assistants, and other small mobile devices. The toolkit includes the emulation environments, performance optimization and tuning features, documentation, and examples that developers need to bring efficient and successful wireless applications to market quickly [8]. As shown on Figure 5, the wireless toolkit that author used during development.

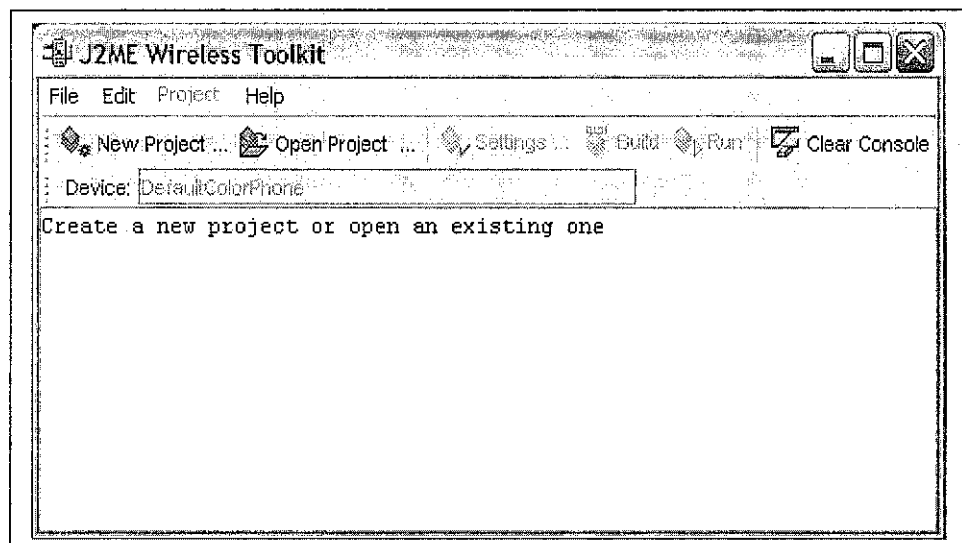


Figure 5: Wireless Toolkit

## 2.5 MySQL Database

MySQL is characterized as a free, fast, reliable open source relational database. It does lack some sophistication and facilities, but it has an active development team and, as it goes from release to release, more capabilities are added.

Currently, Maintenance Department did not have database to record all the reports made by the user. They only manually record all the report and keep it in file. For this research, author had decided to create database to record all reports made by the user.

Author had created database name *WMRS* to record the problem details sent by user through wireless. Below is the class diagram for database *WMRS*.

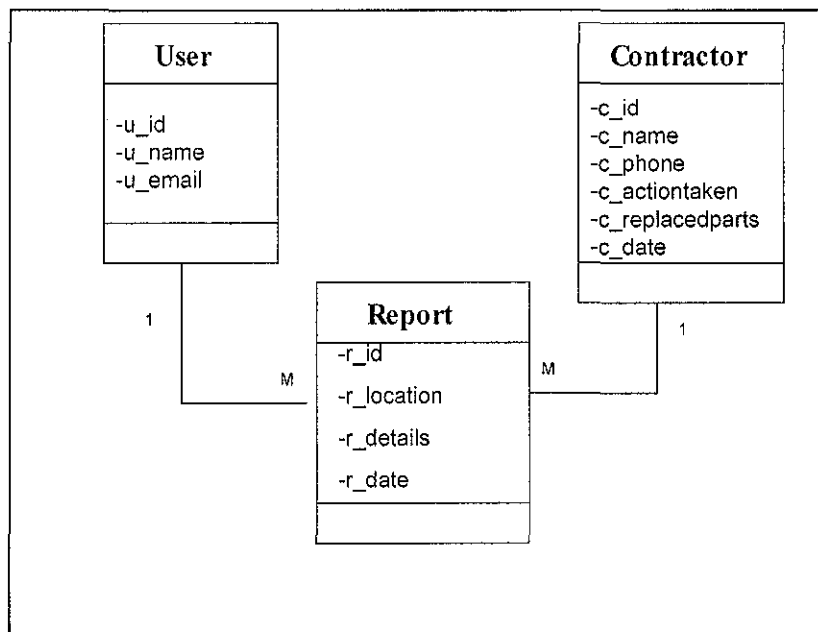


Figure 6: Class Diagram

There are 3 tables which author had created, namely *user*, *contractor* and *report*.

**Database wmrs - table user**

Field	Type	Attributes	Null	Default	Extra
u_id	int(10)		Yes	0	
u_name	varchar(30)		Yes		
u_email	varchar(20)		Yes		

**KeyName Unique Field**  
PRIMARY Yes u\_id

Table 1: user

**Database wmrs - table report**

Field	Type	Attributes	Null	Default	Extra
r_id	int(10)		Yes		auto_increment
r_location	varchar(8)		Yes		
r_details	varchar(20)		Yes		
r_date	date		Yes		

**KeyName Unique Field**  
PRIMARY Yes r\_id

Table 2: report

**Database wmrs - table contractor**

Field	Type	Attributes	Null	Default	Extra
c_id	int(10)		Yes	0	
c_name	varchar(8)		Yes		
c_phonenumber	varchar(13)		Yes		
c_actiontaken	varchar(50)		Yes		
c_replacedparts	varchar(20)		Yes		

**KeyName Unique Field**  
PRIMARY Yes c\_id

Table 3: contractor

A use case is a sequence of actions that provide a measurable value to an actor. Another way to look at it is a use case describes a way in which a real-world actor interacts with the system. Below is the use case diagram for WMRS;

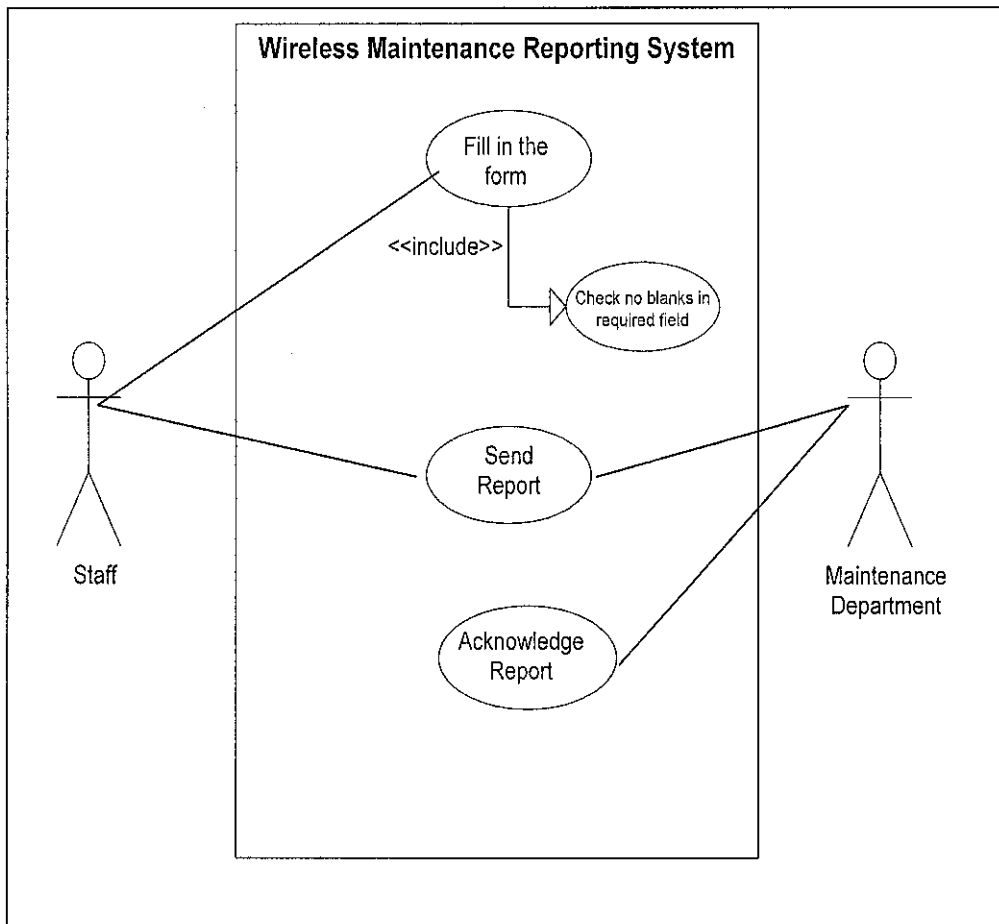


Figure 7: Use Case Diagram

### 2.5.1 Java Database Connectivity (JDBC)

Java Database Connectivity (JDBC) provides Java developers with standard API (Application Programming Interface) which consists of a set of Java classes, interfaces and exceptions and specification to which both JDBC driver vendors and developers adhere when developing applications. [9]

### 2.6 Database Architecture Prototype

The aim of this project is to enable user to send report to maintenance department through J2ME enable mobile phone. As shown below is draft architecture for this project,

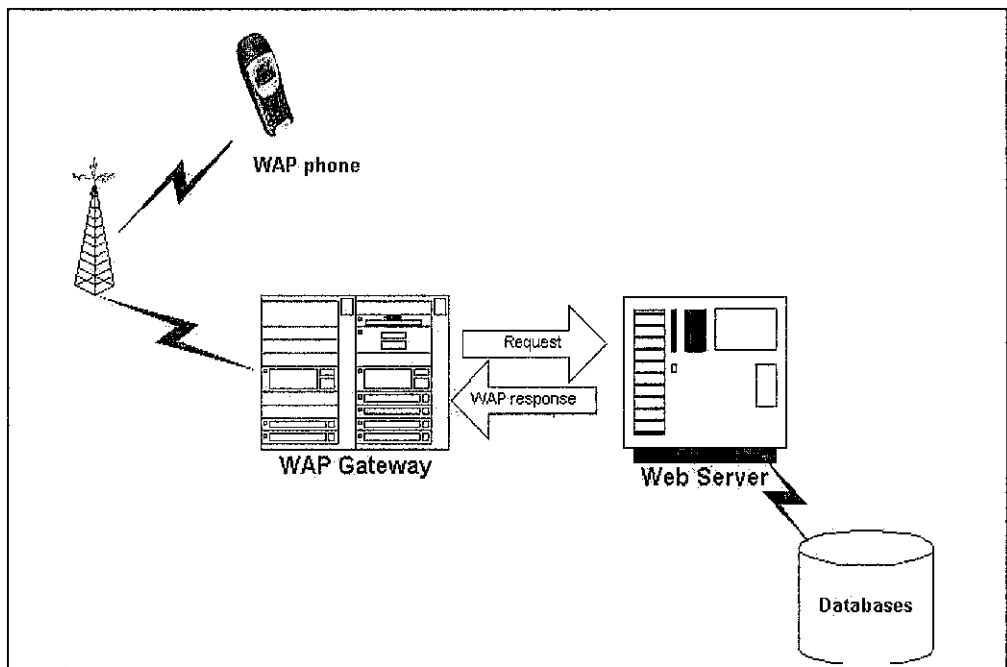


Figure 8: Draft Architecture

The user will send the report information through WAP device and press a send button. Before it sends the request message to remote WAP gateway, the user agent will convert the message into compact binary format to decrease data size and hence increase the rate of transmission. When WAP gateway receives the



message, it will decode the binary message back into text-based format and forward it to the web server (in this case maintenance department reporting server). Once the web server receives the request, it will update the database and the maintenance officer can check the report send by the user.

A wireless Health Outcomes Monitoring System (WHOMS), similar concepts that author found from *BMC Medical Informatics and Decision making* article [10]. The main idea of this study is to develop a new system for transmitting patients' self-reported outcomes using mobile phone or the internet, and to test whether patients can and will use the system via a mobile phone.

They have developed a prototype of Wireless Health Outcomes Monitoring System, which allows structured questionnaires to be sent to the patient by their medical management team. The prototype was designed and developed in order to satisfy two main objectives which are to allow patients to receive self-report structured questionnaires via either WAP or Web and to allow the physician to examine data reported on questionnaires through a graphical and chromatic interface.

The prototype has been developed through use of open source software. The data structured was implemented through MySQL database, which based on Tomcat Apache Web Server. An emulator was employed to develop and run preliminary tests on the WAP component of the system.

From this sample of application, author found that the concept of the system is similar with this study where the uses of mobile phone as a device to send report to the database.

When using a wireless device for Internet browsing, such as a cell phone, one of the most frequent actions a user performs is to submit data in a form, such as a login box. There are several ways to implement this functionality with Common Gateway Interface (CGI) scripts and servlets as the most popular.

The Connected Limited Device Configuration (CLDC) provides a generic Connection framework that you can use to develop network-based applications. In addition, the Mobile Information Device Profile (MIDP) provides the `HttpConnection` interface, which is part of the `javax.microedition.io` package that defines the necessary methods and constants for an HTTP connection.

## CHAPTER 3

### METHODOLOGY

#### 3.0 METHODOLOGY AND PROJECT WORK

#### 3.1 System Development Life Cycle (SDLC)

The method being used in the development is System Development Life Cycle (SDLC). The method used refers to the waterfall model. It is a software process model that involves systematic progression. It suggests a systematic, sequential approach to software development that begins at the system level and progress through planning, analysis, design and coding and implementing.

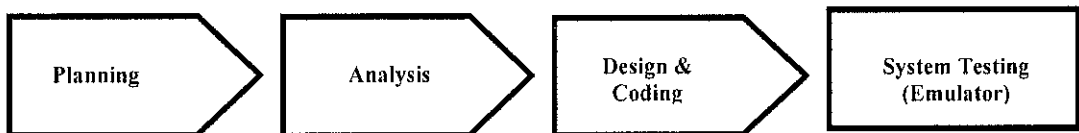


Figure 9: SDLC

##### 3.1.1 Planning Phase

In the first phase, problems are identified from the actual situation. Author had interviewed the Maintenance Department Officer in order to have an overview on how the reporting process was being conducted. In addition, the current way of collecting reports is using 'Defect Form', a special form that they need to fill and endorsed by maintenance engineer before the problems being fixed. The hardware and software specifications were also one of the major focuses in this phase.

### **3.1.2 Analysis Phase**

This phase is to identify what are the tools and equipments best suit to be applied on the project. The author had to ensure that the compatible hardware and software used will fulfill the need of having a working prototype and demonstrate the prototype according to the objectives specified.

### **3.1.3 Design and Coding**

Design phase is where the author has to come out with;

- The server design
- The user interface design

The server design is developed using Jakarta Tomcat version 3.3.2. Apache Tomcat powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations.

For database, author use MySQL Server 5.0. As for user interface design, author use Sun Wireless Toolkit to design and test the application using emulator. It will develop under J2ME platform where it involves the design on application interface to the user for browsing and data entry purposes.

### **3.1.4 System Testing Through Emulator**

Emulator is used in testing the prototype that will be developed. The application will be tested to check whether it is working according to specifications.

## **3.2 Tools and Equipments**

For this project, the following tools will be used in project development.

### **3.2.1 Software and hardware Requirements**

#### **Development Tools:**

- J2ME Wireless Toolkit Beta 2
- J2SE (J2SDK 1.4.2\_04)
- MySQL-Connector Java 3.0.1 Beta
- MySQL Server 5.0
- Tomcat 4.1 server

## CHAPTER 4

### RESULTS AND DISCUSSION

#### 4.0 RESULTS AND DISCUSSION

The outcome of the project is a prototype of wireless technology and application that is, **Wireless Maintenance Reporting System (WMRS)** that can provide Maintenance Department a new way of receiving report from the user. The interface of the reporting page can be tested with the J2ME SDK emulator.

##### 4.0.1 Wireless Maintenance Reporting System

Below are the screenshot of the application;

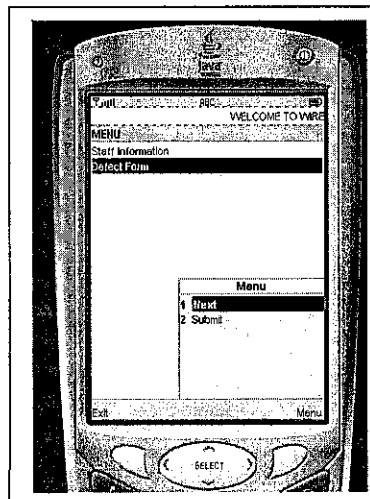


Figure 10: WMRS Main Screen

As shown on Figure 10, it is the first screen of the application. There are list of menu for user, namely *Staff information* or *Defect Form*. They have to fill both of them before their report can be sent to Maintenance Department.

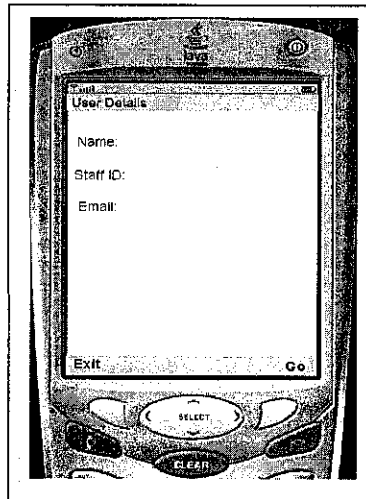


Figure 11: User Details

When users select *Staff Information*, this is the screen for users to fill in (Figure 11). The details are *Name*, *Staff ID* and *Email Address*. Next, they will proceed to Defect Form screen (Figure 12). Defect form is the form where users need to fill in and send to Maintenance Department for them to alert a technician to fix the problems reported.

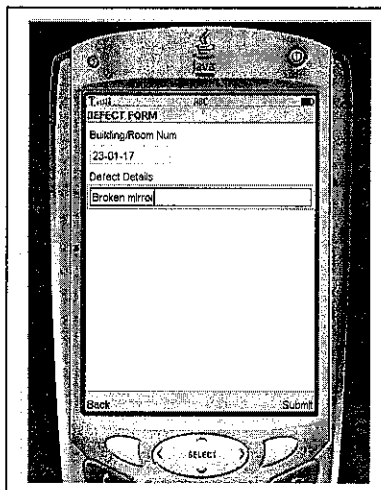


Figure 12: Defect Form

The fields that user need to fill the *Building / Room Number* and *Defect Details* fields. Lastly, when user select *Submit* button, the report will be sent to database and notify the Maintenance Department that a new report has been sent.

#### 4.0.2 Discussion

J2ME provides many benefits for wireless application development such as platform independence, simple programming language and graphical user interface. Wireless application developed using Java can be run on different devices from different vendors. It greatly improves the program's portability.

There are also some limitations in wireless devices such as less memory, smaller screen display, limited input device for example phone keypad, limited battery life and low bandwidth network connectivity.

During the development of WMRS, author has faced some problems. The flow of the system was successfully run but has problem in configuring database connection with emulator. To connect with database, author used JDBC. Although MySQL server and JDBC driver were successfully installed, author still faced problems with database connection. This problem made WMRS not meeting expectations as author can not show how details sent by user to server can automatically updated the database in Maintenance Department. Moreover it takes time to learn about J2ME and its API.

System testing through emulator showed how the system flows from the welcome screen to the menu for user to select. Next, user has to fill in their personal and defect information. After they fill in the details, the record will be submitted to the server and their report will be updated in database. The problem that author faced here was again no connection with database. It is actually to prove that when the user submits the form, the database will automatically update the record and alert the Maintenance Department about the report received. Basically the flows of the system are shown as Figure 13.



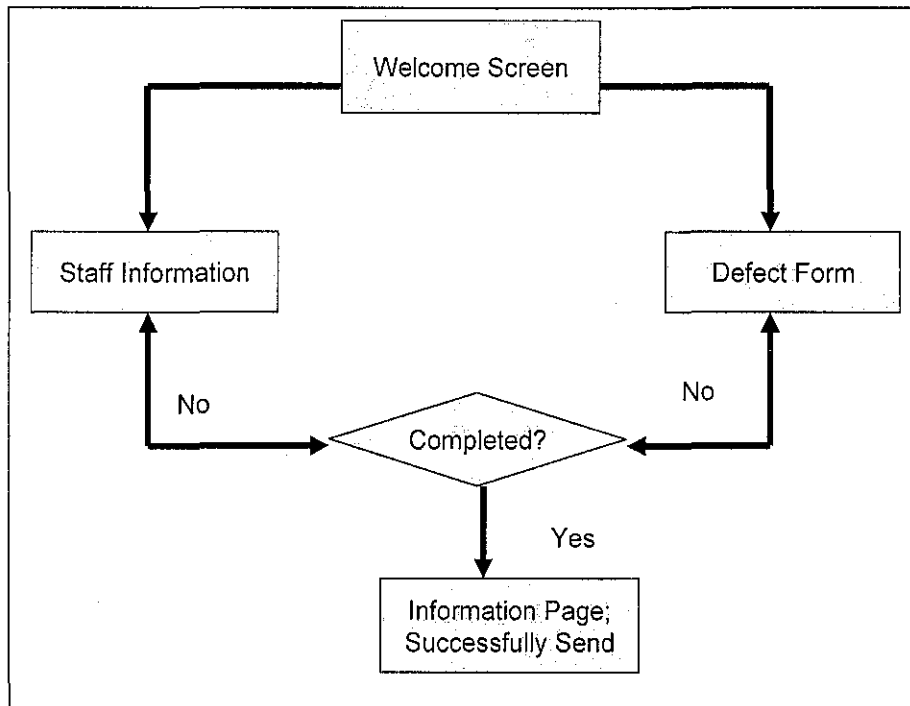


Figure 13 : WMRS Workflow

Author had tried another alternative which has similar concept with WMRS which used J2ME as the platform. In order to meet the results specifications, author has implemented the testing using WML and WAP as a testing platform. This method had similar concept with J2ME where author had created defect form and when the user submit the form, it will automatically updated database in Maintenance Department.

The results met the expectations where the system flow correctly from the beginning where user input their information and details of the problems and send the report to the server at Maintenance Department. Database also automatically updated when user *Submit* their report. The results of implementation using WAP platform are as follow:

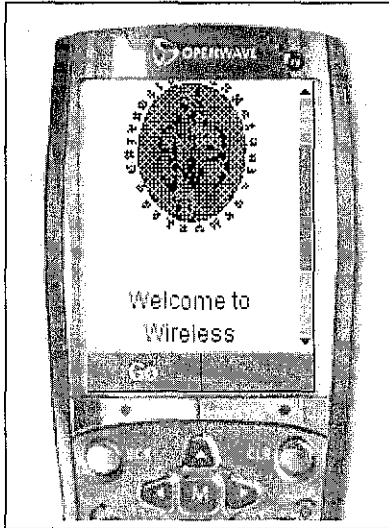


Figure 14: Welcome screen

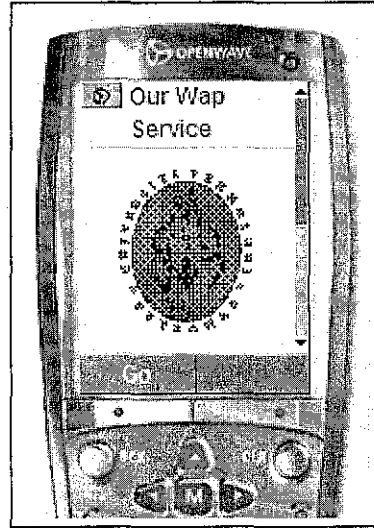


Figure 15: Welcome Screen

As shown above (*see figure 14 and figure 15*) are the welcome screen of **WMRS**. When user click **Go**, the next screen are the menu for user to choose (*see figure 16*)

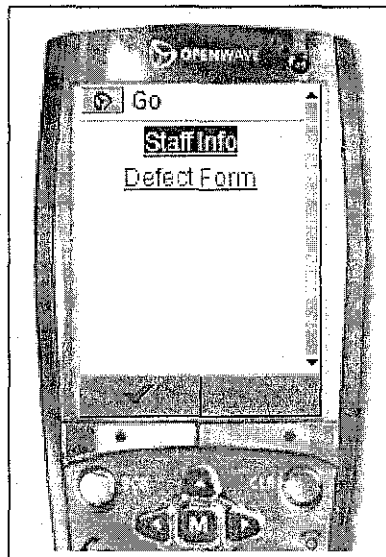


Figure 16: Main Menu

Both of the links are the details that user need to fill in before they can submit the report. When user click **Staff Info** link, the form that required staff information will appear as shown on *figure 17*.

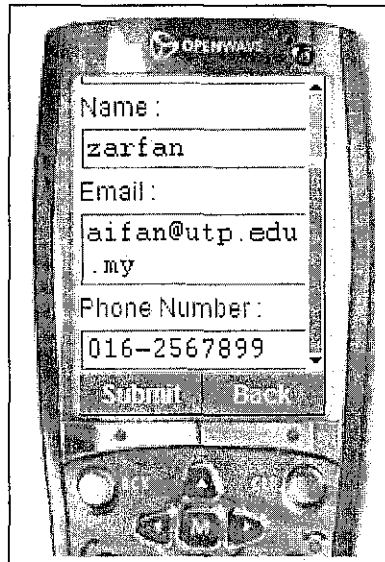


Figure 17: Staff Information

The details that user needs to provide are ID, Name, Email address and Phone Number. When they click **Submit**, a screen that informs their report has been successfully sent will appear. (See figure 18)

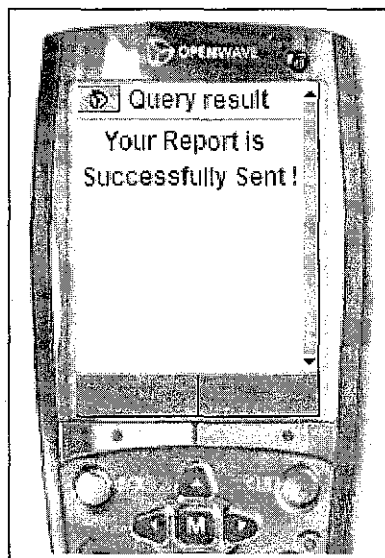


Figure 18 : Query Result screen

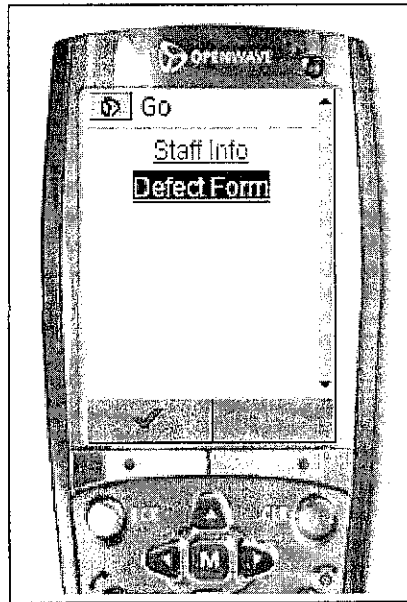


Figure 19: Main Menu

After completing the Staff Information, user has to fill in the *Defect Form*. The information required are *Location* and *Details*. (See figure 20). After user submits the details, the screen that informs user their report has been successfully sent will appear. (See figure 21)

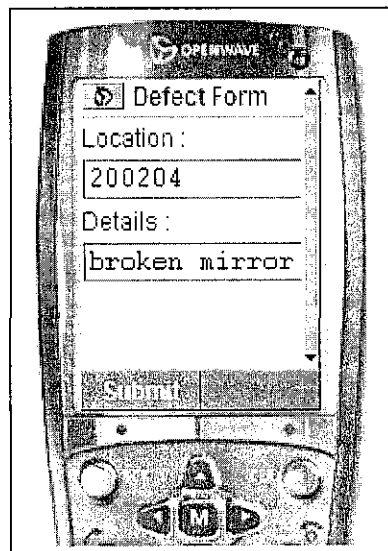


Figure 20: Defect Form

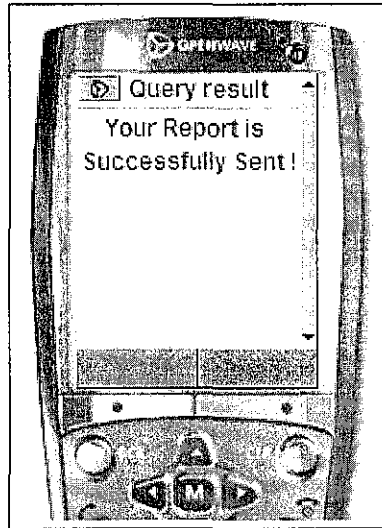


Figure 21: Query Result screen

For this testing, the record that has been sent by the user was successfully updated in database. This result was up to expectations of this project where the report sent by user will automatically insert in database and alert the Maintenance Department about the new report they received. The sample of updated database shown below; *(See figure 22)*

repid	name	address	tel	
10023	zarfan	aifan@utp.edu.my	016-2567899	Edit Delete

id	loc	detail	
7	200204	broken mirror	Edit Delete

Figure 22: Updated Database

This prototype can be modified for future enhancement. Author recommends that the form should be verified first before the details being sent to Maintenance Department. For example, if the form was not completed, the user should be given a notice to notify that their application was not completed. All details are essential for Maintenance Department record.

If this prototype can fulfill the needs of Civil and Structure section, other sections also can adopt the same concept with **WMRS**. This method can save their time and also give user more flexibility in sending report

## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATION**

#### **5.0 CONCLUSION AND RECOMMENDATION**

The aim of Wireless Maintenance Reporting System (WMRS) is to make the reporting system more efficiency and flexibility for users. The use of wireless technology is to improve the maintenance department's way of receiving report from the end users.

The key benefits of this system are to increase productivity, ability to work anywhere and reduced downtime and costs with external office moves.

For future development, author hopes that there will be more wireless solution being applied in UTP itself. Not only for Civil and Structure section, but other section in the Maintenance Department too.

## REFERENCES

- [1] Yu Feng, Dr Jun Zhu, 2001, *Wireless Java Programming with J2ME, Micro Edition*.
- [2] Mobility Overview. *Why Use the Java Platform for Wireless Development?*  
<<http://developers.sun.com/techttopics/mobility/overview.html>>  
Retrieved on 9<sup>th</sup> September 2006.
- [3] The Wireless Java Application Environment, *Java Technology*  
<[http://www.3gamericas.org/pdfs/java\\_mar2005.pdf#search=%22JAVA%20API%20for%203G%20wireless%20application%22](http://www.3gamericas.org/pdfs/java_mar2005.pdf#search=%22JAVA%20API%20for%203G%20wireless%20application%22)>  
Retrieved on 11<sup>th</sup> September 2006.
- [4] Nieto Stein-Buhler, 2002, *Wireless Internet & mobile business, How To Program*.
- [5] Jonathan Knudsen, 2003, *Wireless Java: Developing with J2ME, Second Edition*.
- [6] Yu Feng, Dr Jun Zhu, 2001, *Wireless Java Programming with J2ME, Micro Edition*.
- [7] Article, *J2ME MIDP and WAP Complementary Technologies*  
<<http://developers.sun.com/techttopics/mobility/midp/articles/midpwap/>>  
Retrieved on 15<sup>th</sup> October 2006
- [8] Sun java wireless toolkit for cldc  
<<http://java.sun.com/products/sjwtoolkit/>>  
Retrieved on 20<sup>th</sup> March 2007.
- [9] Faisal Khan, 2005, *A Brief Introduction to JDBC*  
<<http://stardeveloper.com/articles/display.html?article=2003082701&page=1>>  
Retrieved on 20<sup>th</sup> March 2007.



- [10] Article, *A wireless health outcomes monitoring system (WHOMS): Development and field testing with cancer patients using mobile phones.*

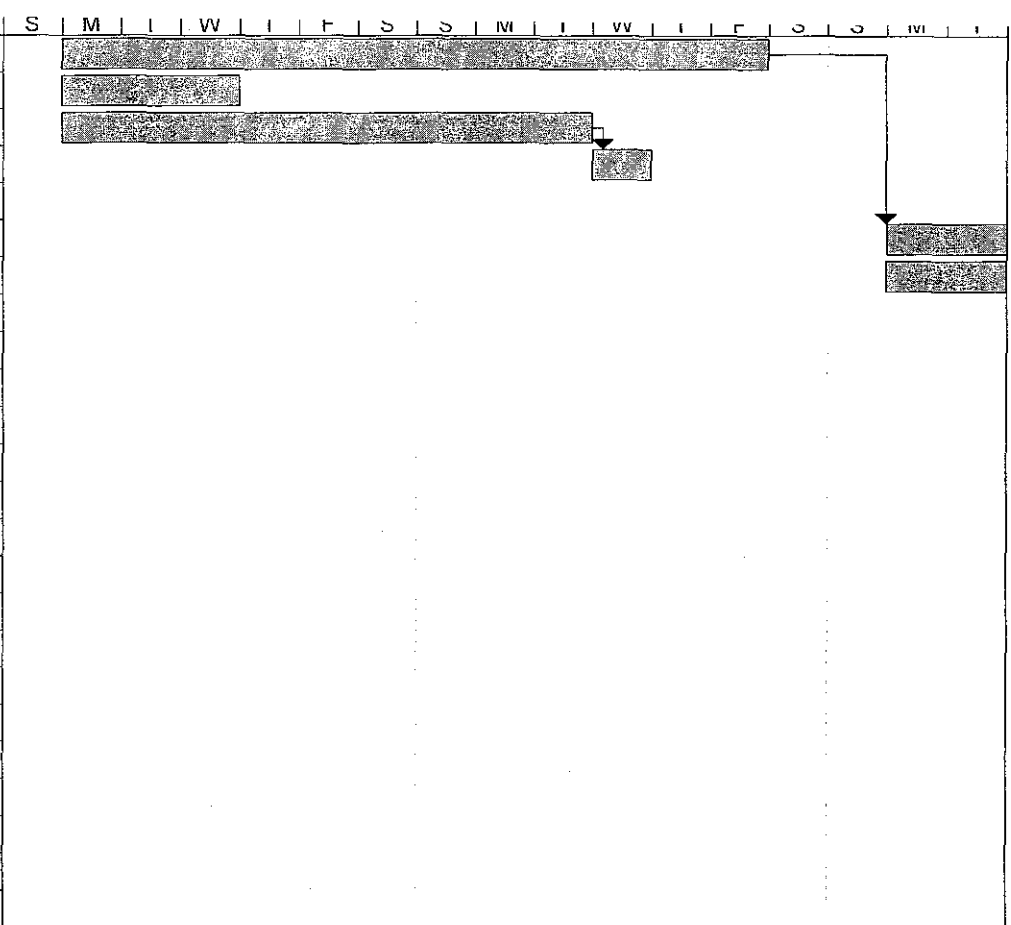
<http://www.biomedcentral.com/1472-6947/4/7>

Retrieved on 16<sup>th</sup> October 2006.

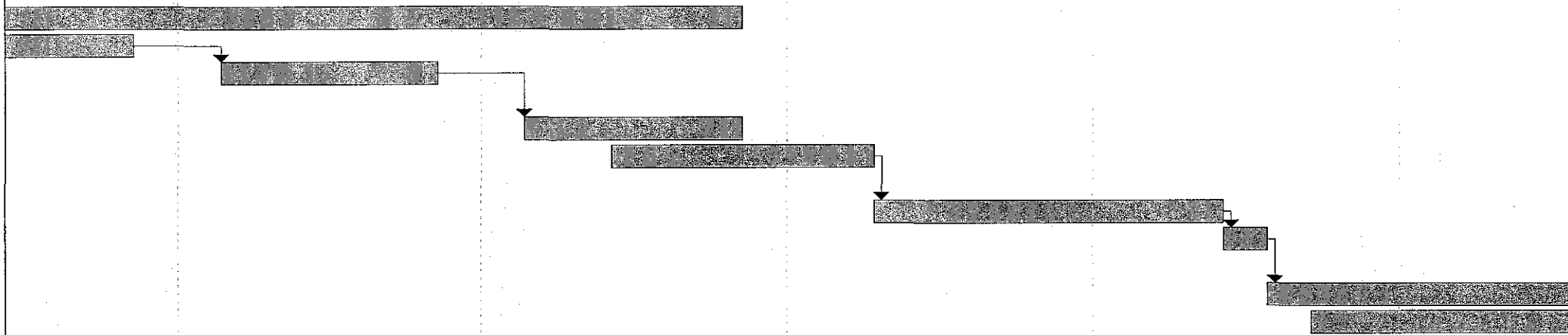
# **APPENDICES**

## **APPENDIX 1: GANTT CHART**










1		1.0 Project work continue	10 days	Mon 1/29/07	Fri 2/9/07
2		Define project scope	3 days	Mon 1/29/07	Wed 1/31/07
3		Prepare progress report	7 days	Mon 1/29/07	Tue 2/6/07
4		Submit Progress Report 1	1 day?	Wed 2/7/07	Wed 2/7/07
5					
6		<b>2.0 Requirements Analysis Phase</b>	15 days	Mon 2/12/07	Fri 3/2/07
7		Identify Process	5 days	Mon 2/12/07	Fri 2/16/07
8		Identify Interface	5 days	Mon 2/19/07	Fri 2/23/07
9					
10		<b>3.0 Decision analysis Phase</b>	5 days	Mon 2/26/07	Fri 3/2/07
11		Tools and Equipment to use	4 days	Wed 2/28/07	Mon 3/5/07
12					
13		<b>4.0 Design Phase</b>	6 days	Tue 3/6/07	Tue 3/13/07
14		Submit Preliminary Report II	1 day?	Wed 3/14/07	Wed 3/14/07
15					
16		<b>5.0 Construction Phase</b>	15 days	Thu 3/15/07	Wed 4/4/07
17		Build system	8 days	Fri 3/16/07	Tue 3/27/07
18		Seminar	1 day	Mon 4/2/07	Mon 4/2/07
19		Bug Errors	2 days	Tue 4/3/07	Wed 4/4/07
20		Submit Final Report	1 day	Fri 4/6/07	Fri 4/6/07
21					
22		<b>6.0 Implementation Phase</b>	2 days	Mon 4/23/07	Tue 4/24/07
23		Product Deliverable	1 day	Mon 4/23/07	Mon 4/23/07
24		Final Presentation	1 day?	Mon 4/23/07	Mon 4/23/07



Project: gantChart Date: Mon 5/7/07	Task		Milestone		External Tasks	
	Split		Summary		External Milestone	
	Progress		Project Summary		Deadline	



Project: gantChart  
Date: Mon 5/7/07

Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	



## **APPENDIX II: SOURCE CODE**

## SOURCE CODE

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.CommandListener;

public class GuiTests extends MIDlet implements CommandListener {
    // display manager
    Display display = null;

    private ImageItem imageItem;
    private Form form_welcome;

    // a menu with items
    List menu = null; // main menu

    // textbox
    TextBox input;

    //TextField input;

    // ticker
    Ticker ticker = new Ticker("WELCOME TO WIRELESS MAINTENANCE REPORTING SYSTEM");

    // form
    Form form = new Form("DEFECT FORM");

    // text field
    //TextField textfield = new TextField(null, null, 5, 0);
    //TextField textfield2 = new TextField(null, null,50, 0);

    // command
    static final Command backCommand = new Command("Back", Command.BACK, 0);
    static final Command mainMenuCommand = new Command("Main", Command.SCREEN, 1);
    static final Command exitCommand = new Command("Exit", Command.STOP, 2);
    static final Command nextCommand = new Command("Next", Command.SCREEN, 1);
    static final Command submitCommand = new Command("Submit", Command.SCREEN, 1);
    String currentMenu = null;

    // constructor.
    public GuiTests() {
    }

    /**
     * Start the MIDlet by creating a list of items and associating the
     * exit command with it.
     */
    public void startApp() throws MIDletStateChangeException {

        //Image and ImageItem
        try {
            Image image = Image.createImage("/info.gif");
            imageItem = new ImageItem("ImageItem is below:", image, ImageItem.LAYOUT_NEWLINE_BEFORE,
"Alternate text for the image file");
            form.append(image);
            form.append(imageItem);
        }

        catch (java.io.IOException e) {}

        display = Display.getDisplay(this);
    }
}
```



```

menu = new List("MENU", Choice.IMPLICIT);

menu.append("Staff Information", null);
menu.append("Defect Form", null);
menu.addCommand(exitCommand);
menu.addCommand(nextCommand);
menu.addCommand(submitCommand);
menu.setCommandListener(this);
menu.setTicker(ticker);

mainMenu();
}

public void pauseApp() {
    display = null;

    menu = null;
    ticker = null;
    form = null;
    input = null;

    //textfield = null;
}

public void destroyApp(boolean unconditional) {
    notifyDestroyed();
}

// main menu
void mainMenu() {
    display.setCurrent(menu);
    currentMenu = "Main";
}

/**
 * Test the TextBox component.
 */
public void testTextBox() {

    input = new TextBox("Name", "", 10, TextField.ANY);

    input.setTicker(new Ticker("Staff Information"));
    input.addCommand(backCommand);
    input.addCommand(nextCommand);
    input.setCommandListener(this);
    input.setString("");
    display.setCurrent(input);
    currentMenu = "input";
}

/**
 * Test the Form component.
 */
public void testForm() {

    //form.append(gauge);
    form.append("Building/Room Num");
    TextField textfield = new TextField(null, null, 8, 0);
    form.append(textfield);
    form.append("Defect Details");
    TextField textfield2 = new TextField(null, null, 50, 0);
    form.append(textfield2);
    form.addCommand(backCommand);
}

```

```

form.addCommand(submitCommand);
form.setCommandListener(this);
display.setCurrent(form);
currentMenu = "form";
}

/**
 * Handle events.
 */
public void commandAction(Command c, Displayable d) {
    String label = c.getLabel();
    if (label.equals("Exit")) {
        destroyApp(true);
    } else if (label.equals("Back")) {
        if(currentMenu.equals("list") || currentMenu.equals("input") ||
            currentMenu.equals("date") || currentMenu.equals("form")) {
            // go back to menu
            mainMenu();
        }
    } else {
        List down = (List)display.getCurrent();
        switch(down.getSelectedIndex()) {
            case 0: testTextBox();break;
            case 1: testForm();break;
        }
    }
}
}
}

```

```
<?php
```

```

Header("Content-type: text/vnd.wap.wml");
Header("Cache-Control: no-cache, must-revalidate");
Header("Pragma: no-cache");
echo ("<?xml version='1.0'?>");
?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml" >

```

```
<wml>
```

```

<card id="HomeCard" ontimer="#MainCard" title="Our Wap Service">
<timer value="250"/>
<do type="accept" label="Go">
<go href="#MainCard"/>
</do>
<p align="center"><b>

```

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

```
<br/><br/>
```

```
Welcome to Wireless Maintenance Reporting System<br/>
```

```
</p>
```

```
<br/><br/>
```

```
</p>
```

```
</card>
```

```

<card id="MainCard" title="Go">
<do type="prev" label="Back"><prev/></do>
<p align="center">
<a href="#repid">Staff Info</a><br/>
<a href="#defect">Defect Form</a><br/>

```

```
</p>
```

</card>

```
<card id="repid" title="Staff Info">
<do type="prev" label="Back"><prev/></do>
<p align="left">
ID : <br/>
<input type="text" name="repid"/>
</p>
<p align="left">
Name : <br/>
<input type="text" name="name"/>
</p>
<p align="left">
Email : <br/>
<input type="text" name="address"/>
</p>
<p align="left">
Phone Number : <br/>
<input type="text" name="tel"/>
</p>
```

```
<do type="accept" label="Submit">
<go method="post" href="mysql.php">
<postfield name="repid" value="{repid}"/>
<postfield name="name" value="{name}"/>
<postfield name="address" value="{address}"/>
<postfield name="tel" value="{tel}"/>
<do type="prev" label="Back"><prev/></do>

</go>
</do>
```

```
<do type="prev" label="Back"><prev/></do>
<go method="post" href="#defect">
```

```
</card>
<card id="defect" title="Defect Form">
<do type="prev" label="Back"><prev/></do>
<p align="left">
Location : <br/>
<input type="text" name="loc"/>
</p>
<p align="left">
Details : <br/>
<input type="text" name="detail"/>
</p>
```

```
<do type="accept" label="Submit">
<go method="post" href="mysql2.php">
<postfield name="loc" value="{loc}"/>
<postfield name="detail" value="{detail}"/>
</go>
</do>
```

</card>

</wml>

```

<SCRIPT LANGUAGE="PHP">
header("Content-type: text/vnd.wap.wml");
$REMOTE_ADDR=$_SERVER["REMOTE_ADDR"];
$login=$_REQUEST["login"];
$password=$_REQUEST["password"];

echo "<?xml version=".chr(34)."1.0".chr(34)."?">\n";
echo "<!-- generated by WinMySQL Professional 1.0 (http://www.winmysql.com/) - Unregistered copy -->\n";
echo "<!DOCTYPE wml PUBLIC ".chr(34)."-/WAPFORUM//DTD WML 1.1//EN".chr(34)."
".chr(34)."http://www.wapforum.org/DTD/wml_1.1.xml".chr(34).">\n\n";
echo "<wml>\n";
echo "<card id=".chr(34)."MainAnswer".chr(34)." title=".chr(34)."Query result".chr(34).">\n";
echo "<do type=".chr(34)."prev".chr(34)." label=".chr(34)."Back".chr(34)."><prev/></do>\n";
echo "<p align=".chr(34)."center".chr(34).">\n";

if (isset($login))
$_CntResult=@mysql_connect("localhost",$login,$password);
else
$_CntResult=@mysql_connect("localhost","root","");

if ($_CntResult<1)
echo "connection to database refused - MySQL unavailable or access denied !\n";
else
{ $_WebQuery="insert into tbl_usr (";

if (isset($_REQUEST["repid"]))
{ $repid=$_REQUEST["repid"]; }
if (isset($repid))
$_WebQuery=$_WebQuery."repid"." ";

if (isset($_REQUEST["name"]))
{ $name=$_REQUEST["name"]; }
if (isset($name))
$_WebQuery=$_WebQuery."name"." ";

if (isset($_REQUEST["address"]))
{ $address=$_REQUEST["address"]; }
if (isset($address))
$_WebQuery=$_WebQuery."address"." ";

if (isset($_REQUEST["tel"]))
{ $tel=$_REQUEST["tel"]; }
if (isset($tel))
$_WebQuery=$_WebQuery."tel";

if (substr($_WebQuery,strlen($_WebQuery)-2,2)==" ")
$_WebQuery=substr($_WebQuery,0,strlen($_WebQuery)-2);

$_WebQuery=$_WebQuery.") values(";

if (isset($repid))
$_WebQuery=$_WebQuery.$repid"." ";

if (isset($name))
$_WebQuery=$_WebQuery.chr(34).$name.chr(34)." ";

if (isset($address))
$_WebQuery=$_WebQuery.chr(34).$address.chr(34)." ";

if (isset($tel))
$_WebQuery=$_WebQuery.chr(34).$tel.chr(34);

if (substr($_WebQuery,strlen($_WebQuery)-2,2)==" ")
$_WebQuery=substr($_WebQuery,0,strlen($_WebQuery)-2);

$_WebQuery=$_WebQuery.")";

$_QryResult=mysql_db_query("db_wap",$_WebQuery,$_CntResult);

if (mysql_errno()==0)

```

```
echo "<b>Your Report is Successfully Sent !</b>\n";
else
echo "<b>Record not inserted, reason given by server :<br/>\n<br/>\n".mysql_error()."</b>\n";

echo "</p>\n";
echo "</card>\n";
echo "</wml>\n";
}

</SCRIPT>
```

```
<?php
header("Content-type: text/vnd.wap.wml");
echo "<?xml version='1.0'?>";
echo "<!DOCTYPE wml PUBLIC '-//WAPFORUM//DTD WML 1.1//EN'"
. "'http://www.wapforum.org/DTD/wml_1.1.xml'">";
?>

<wml>
<card id="card2" title="add">
<p>

<?php

//connect to database server
$connection = mysql_connect("localhost", "root", "")
or die ("connection failed");

//Select the database
$db = mysql_select_db("wmrs", $connection)
or die ("failed to select database <br />\n");

$sql = mysql_query("INSERT INTO user (u_id, u_name, u_email) Values ('$_POST[repid]', '$_POST[name]',
'$_POST[address]')") or die("Input failed");

if (!$sql) {
echo "Error cannot Insert record.!!!<br/>"; mysql_error;exit;
}else {
echo "record inserted";
}

//insert OK inform user

print "<b>Customer Database:</b> <br/>";

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

```

<SCRIPT LANGUAGE="PHP">
header("Content-type: text/vnd.wap.wml");
$_REMOTE_ADDR=$_SERVER["REMOTE_ADDR"];
$_login=$_REQUEST["login"];
$_password=$_REQUEST["password"];

echo "<?xml version=".chr(34)."1.0".chr(34)."?>\n";
echo "<!-- generated by WinMySQL Professional 1.0 (http://www.winmysql.com/) - Unregistered copy -->\n";
echo "<!DOCTYPE wml PUBLIC ".chr(34)."-//WAPFORUM/DTD WML 1.1//EN".chr(34).".chr(34).\"http://www.wapforum.org/DTD/wml_1.1.xml\".chr(34).\">\n\n";

echo "<wml>\n";
echo "<card id=".chr(34)."MainAnswer".chr(34)." title=".chr(34)."Query result".chr(34).">\n";
echo "<do type=".chr(34)."prev".chr(34)." label=".chr(34)."Back".chr(34)."><prev/></do>\n";
echo "<p align=".chr(34)."center".chr(34).">\n";

if (isset($_login))
$_CntResult=@mysql_connect("localhost",$_login,$_password);
else
$_CntResult=@mysql_connect("localhost","root","");

if ($_CntResult<1)
echo "connection to database refused - MySQL unavailable or access denied !\n";
else
{ $_WebQuery="insert into tbl_report (";

if (isset($_REQUEST["loc"]))
{ $_loc=$_REQUEST["loc"]; }
if (isset($_loc))
$_WebQuery=$_WebQuery."loc"." , ";

if (isset($_REQUEST["detail"]))
{ $_detail=$_REQUEST["detail"]; }
if (isset($_detail))
$_WebQuery=$_WebQuery."detail"." , ";

if (substr($_WebQuery,strlen($_WebQuery)-2,2)==" , ")
$_WebQuery=substr($_WebQuery,0,strlen($_WebQuery)-2);

$_WebQuery=$_WebQuery.") values(";

if (isset($_loc))
$_WebQuery=$_WebQuery.$loc." , ";

if (isset($_detail))
$_WebQuery=$_WebQuery.chr(34).$_detail.chr(34)." , ";

if (substr($_WebQuery,strlen($_WebQuery)-2,2)==" , ")
$_WebQuery=substr($_WebQuery,0,strlen($_WebQuery)-2);

$_WebQuery=$_WebQuery.")";

$_QryResult=mysql_db_query("db_wap",$_WebQuery,$_CntResult);

if (mysql_errno()==0)
echo "<b>Your Report is Successfully Sent !</b>\n";
else
echo "<b>Record not inserted, reason given by server :<br/>\n<br/>\n".mysql_error()."</b>\n";

echo "</p>\n";
echo "</card>\n";
echo "</wml>\n";
}

</SCRIPT>

```