ASSUMPTION UNIVERSITY (ABAC), BANGKOK

An Application on mobile phone

by

Ms. Mallika Panumasrusme

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in Information Technology
Assumption University

March 2003

An Application on mobile phone

by

Ms. Mallika Panumasrusme

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology Assumption University

Mar, 2003

The Faculty of Science and Technology

Master Project Approval

Project Title An Application on mobile phone

By Ms. Mallika Panumasrusme
Project Advisor Dr. Jirapun Daengdej

Academic Year $^{2}/_{2}002$

The Department of Information Technology , Faculty of Science and Technology of Assumption University has approved this final report of the **three** credits course. **IT6900 Master Project,** submitted in partial fulfillment of the requirements for the degree of Master of Science in Information Technology .

Approval Committee:

pun Daengdej)
Advisor

(Asst.Prof.Dr. Thotsapon Sortrakul)
Committee Member

(Dr. Kittipong Srisansanee)
Committee Member

(Dr. Thiraphong Charoenkhunwiwa)
Committee Member

Faculty Approval:

(Asst.Prof.Dr. Thotsapon Sortrakul)
Program Director

Supavadee N.
(Ass .P of. Supavadee Nontakao)
Dean

February / 2002

ACKNOWLEDGEMENTS

The author would like to express her sincere thanks and most gratitude to her advisor Dr. Jirapun Daengdej who has been instrumental in helping her choose the appropriate topic for her project and for his invaluable advice, precious guidance and encouragement towards the completion of this project. Together, much gratitude is also expressed to all committee members Asst. Prof. Dr. Thotsapon Sortrakul, Dr. Kittipong Srisansanee and Dr. Thiraphong Charoenkhunwiwat for their suggestions and valuable comments.

The author extends her sincere thanks to all faculty and staff in Information Technology Program for supporting her study and organizing the program curriculum. Special thanks are due to the staff of academic secretariat office who provided her with the necessary information and documents for the implementation of her project.

The author also wish to acknowledge the A.U. friends for their friendship during her study at A.U. and she is very grateful to Mr. Surakiat Lymsittikool for his help in writing up of this project and assisting her in completion of her project.

Finally, the greatest gratitude and all thanks are never to be forgotten to go to her respected parents for their love and support throughout her whole life.

ABSTRACT

The mobile phone devices play an essential factor in people life increasingly. There is a high competition in the mobile phone market sharing. It has caused businesses to experiment with innovative ways of marketing to consumers. Mobile phones being sold in the market not only have the same communication basic but some of them are designed to contain a lot of facilities and capabilities such as organizer function, calculator, voice adjusting, games, and entertainment function. Nevertheless, those special capabilities are not much and limited for adjusting to meet the user's needs. In an attempt to improve the new facilities, a lot of options have been proposed and developed using of J2ME software or Java 2 Platform, Micro Edition, one of interesting technology to upgrade or adjust the application on the mobile phone.

The performance of this project depends on the methodology to develop the J2ME application which runs on devices supporting MIDP called MIDlets. The J2ME emulator is a tool that enables one to run MIDlets on a desktop PC and simulate how the MIDlet will run on a physical mobile device. The project application consist of 3 parts that are Appointment, First aid and project information.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

Although it is not absolutely be in the era of Mobile Data, it can be said that the fierce competition of market sharing has led to launching new facilities, a lot of options, and very cheap cost for the consumers. One of the interesting technologies that is used for developing application on small portable devices is called J2ME or Java 2 Platform, Micro Edition.

Mobile phones being sold in the market not only have the same basic communication but some of them are designed to contain a lot of facilities and capabilities such as organizer function, calculator, voice adjusting, games, and entertainment function. Nevertheless, those special capabilities are not much and limited for adjusting to meet the user's needs in the ways that the users can not replace the game. Sometime the manufacturers launch new software but the users have to take the mobile phone to the service center, which is inconvenient to the users.

I would like the readers to take a step backward and look over the current electronic world. The readers would recognize that beside mobile phone there are so many electrical gadgets in the same status mentioned above such as common organizer (Talking dictionary), pager, small computer used to calculate and record the sale volume, and POS in the department store. The new model of air-conditioner, washer, and new hi-technology software-based facilitator that can be accounted to be in the in-house electrical containing recorder software have the same limitation, that is difficult to upgrade or adjust the software inside, as mobile phone. Purchasing those instruments containing the functions that meet the user's needs does not directly solve the problem because if we compare it to a PC we can recognize that PC is easy to develop a software to meet the user's needs instead of purchasing a new one.

Considering the expanding of the demand of communication tool and electrical gadgets, especially in the case of mobile phone that is expected to have a dramatic growth in volume to be 350 million and 1,000 million in the year 2001 and 2003 respectively is too much higher than PC. It implies that the market target is so enormous than never before. Furthermore, the manufacturers are planning to design its production to be able to link with the Internet network in order that the users can obtain the benefit from the network's capabilities such as information transferring, retrieving and updating the information of either the organization or private. Focusing on mobile communicator and mobile PC or organizer, we can see that the users can upgrade or adjust the capability of those instruments by directly downloading software from suppliers' web-site.

In order to push up the demand of using those instruments in the future, it is important to pay attention to designing the built-in software of the current existing electrical communicator. Current designing standards are various depending on each manufacturer. It is crucial to create a new software standard to be the important foundation for applying. It is pleasure that Sun Microsystems, the Java discoverer, try to establish Java to be the standard language by launching a new language called J2ME to be applied in small instrument containing limitation of operation. It is different from using Java that PC contains so much resources (memory and efficiency of CPU).

The project application consists of 3 parts. There are Appointment, First aid and project information. First aid is the initial care of the injured or sick. It is the care administered by a concerned person as soon as possible after an accident or illness. It is this prompt care and attention prior to the arrival of the ambulance, that sometimes means the difference between life and death, or between a full or partial recovery. That is the reason why I choose this topic to be included in my project application.

1.2 Objectives of the project

- Increase knowledge of software tool for mobile phone such as J2ME.
- Develop an application by using a software development for mobile phone device called J2ME.
- Develop an application on mobile phone device that we can carry to anywhere and can get the information any time we want.

1.3 Scope and Limitation

- Study of Java 2 Platform, Micro Edition (J2ME) software, configuration, coding, and developing project application.
- Generate an application on mobile phone which consists of Appoinment, First Aid, and project information.
- Generate Appointment application on mobile phone to record the appointment information including name, place, appointment date, appointment time and other information and provide alert when the appointment time is due.
- Generate First Aid Information with step by step instructions guiding through commonly occuring emergency medical situations on mobile phone. Topics include asphyxiation, bites and stings, bleeding, burns, choking, concussions and contusions, convulsions, cuts and abrasions,

dislocation, electric shock, painting, fractures, frostbite, heatstroke, hypothermia, nosebleeds, sprains, poisoning, and strains.

- Generate About project application to inform the user of the project description.
- Use the J2ME emulator as the tool to test the application on a desktop PC and simulate how the application will run on a physical mobile phone device.

1.4 Work Processes

The work processes of the project are as follows:-

1. To study the existing application, to understand the operations and the information technology to be used in this study.
2. To analyze the existing application.
3. To study the management and the requirements of application.
4. To design the record store format used in the application.
5. To design the proposed system, input/output user interface, and data management.
6. To develop the proposed system by using J2ME development tool.
7. To test and implement the proposed system after application development to enable users to test the proposed application.
8. To maintain and evaluate after the testing and implementation.
9. To organize the documentation and user manual.

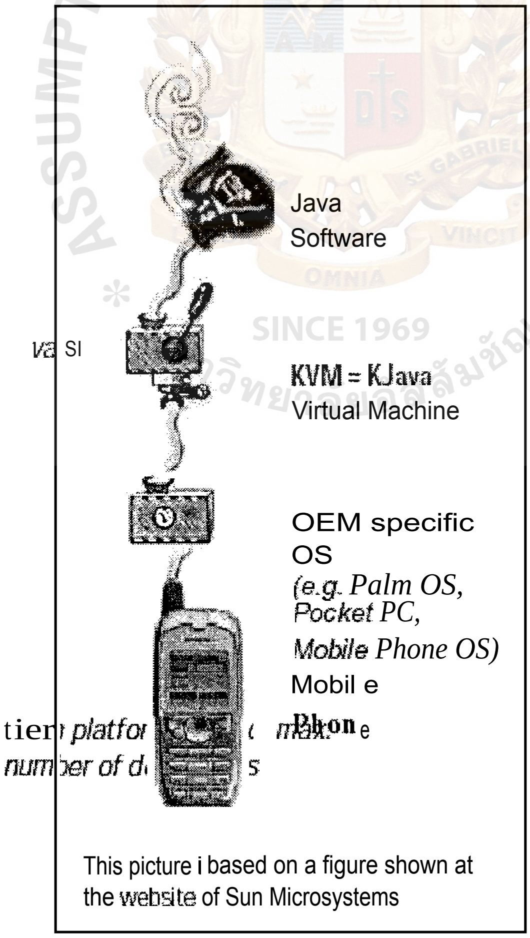
CHAPTER 2

IMPLEMENTATION PLATFORM

Why should we provide a mobile phone with a Java Platform?

The idea of Java is quite simple: "Java technology based software is designed to work just about everywhere – no matter what kind of computer, phone, TV, or operating system. It runs on any kind of compatible device that supports the Java platform". This is a component of the platform called *Java virtual machine* - a kind of translator that turns general Java platform instructions into commands executable on that device.

Figure 2-1 Java technology



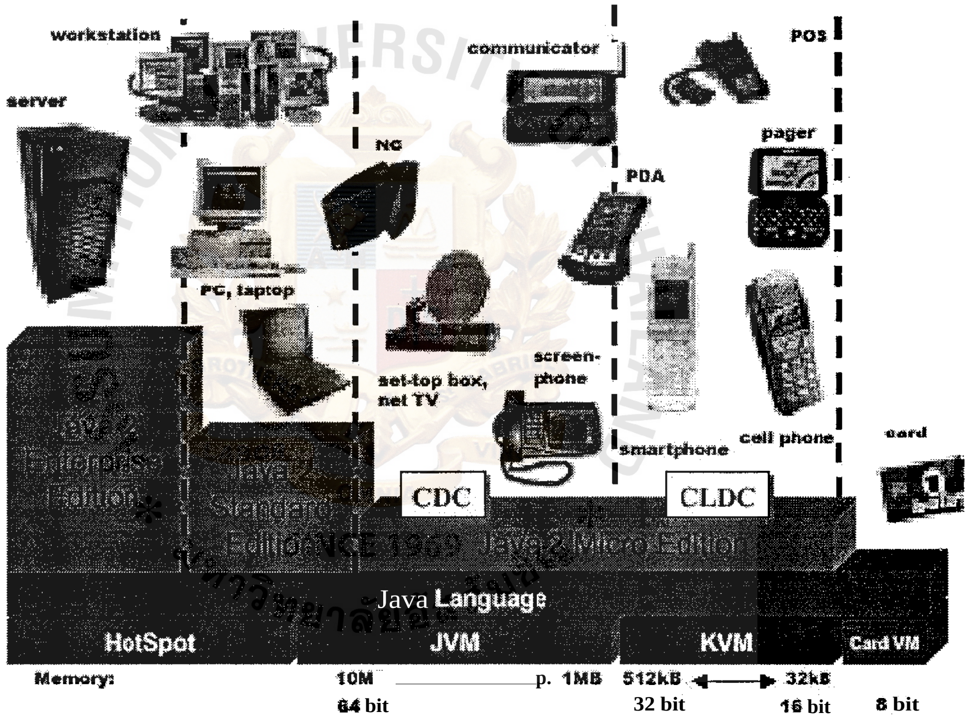
This picture i based on a figure shown at the website of Sun Microsystems

2.1 Concepts of **J2ME**

Sun Microsystems' Java technology actually consists of 3 editions, aimed at different market/technology sectors:

- Java 2 Enterprise Edition J2EE
- Java 2 Standard Edition J2SE
- Java 2 Micro Edition J2ME

Figure 2-2 J2ME technology



Source; J2ME Technology for Creating Mobile Devices; White Paper; Sun Microsys terns; May 2000

Java 2 Platform, Micro Edition (J2ME) aims to adapt Java for the wide spectrum of consumer products ranging from small devices such as palm pilots, mobile phones, to home appliances such as TV, refrigerator, etc. Although J2ME is still using the same Java language as the other editions, it has to reduce the size of the Java runtime environment in order to run in devices that have various memory and power constraints. It achieves this by removing unnecessary classes from the Java 2 Platform, Standard Edition and augmenting it with new classes that are suitable for small devices with limited storage and processing power. This is done by the adaptation of three components:

- The Java virtual machine
- A specific set of libraries and APIs
- The tools for deployment and device configuration

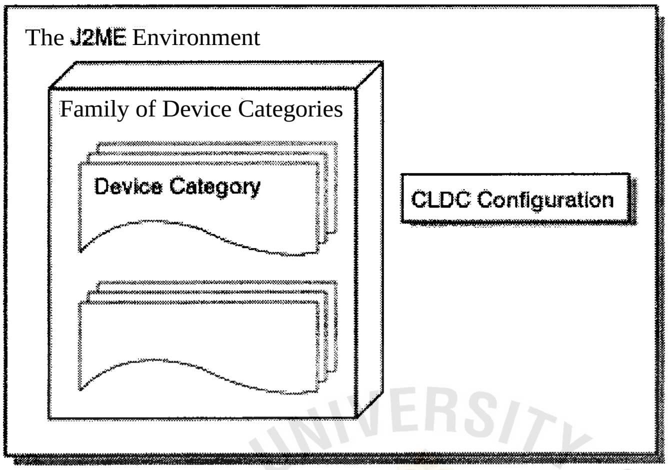
The J2ME platform targets two categories of products:

- Shared, fixed, connected information devices. In the figure above this category is represented by the grouping labeled CDC (Connected Device Configuration).
- Personal, mobile, connected information devices, labeled CLDC (Connected, Limited Device Configuration).

2.2 The J2ME architecture and configuration

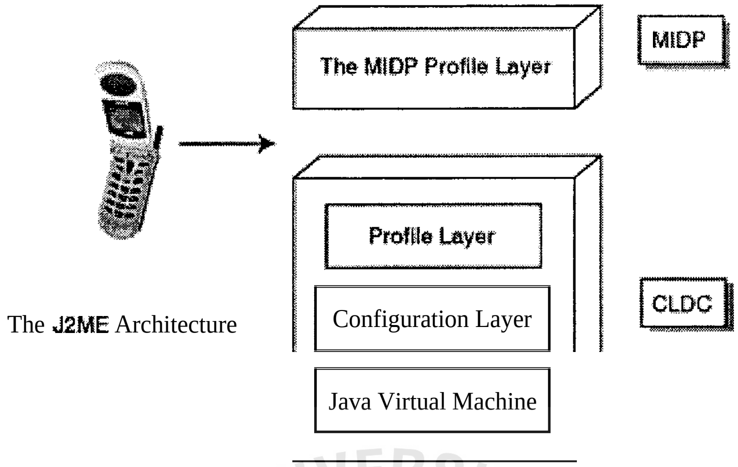
The J2ME architecture is based on *families* and *categories* of devices. A category defines a particular kind of device; cellular telephones, simple pagers, and organizers are separate categories. A family of devices is made up of a group of categories that have similar requirements for memory and processing power. Together, cellular phones, simple pagers, and simple personal organizers make up a single family of small-footprint devices.

Figure 2-3 Families and categories of devices (defines the relationship between the families and categories of devices in the context of J2ME.)



In order to support the kind of flexibility and customizable deployment demanded by the family of resource-constrained devices, the J2ME architecture is designed to be modular and scalable. This modularity and scalability is defined by J2ME technology in a complete application runtime model, with four layers of software built upon the host operating system of the device.

Figure 2-4 The J2ME architecture.



- **Java Virtual Machine layer:** This layer is an implementation of a Java Virtual Machine that is customized for a particular device's host operating system and supports a particular J2ME configuration.
- **Configuration layer:** The configuration layer defines the minimum set of Java Virtual Machine features and Java class libraries available on a particular category of devices. In a way, a configuration defines the commonality of the Java platform features and libraries that developers can assume to be available on all devices belonging to a particular category. This layer is less visible to users, but is very important to profile implementers.
- **Profile layer:** The profile layer defines the minimum set of application programming interfaces (APIs) available on a particular family of devices. Profiles are implemented upon a particular configuration. Applications are written for a particular profile and are thus portable to any device that

supports that profile. A device can support multiple profiles. This is the layer that is most visible to users and application providers.

- **MIDP** layer: The Mobile Information Device Profile (MIDP) is a set of Java APIs that addresses issues such as user interface, persistence storage, and networking.

The Java Virtual Machine layer, configuration layer, and profile layer together constitute the Connected Limited Device Configuration (CLDC). The MID Profile and CLDC provide a standard runtime environment that allows new applications and services to be dynamically deployed on end-user devices.

Programming J2ME with MIDP APIs: The building blocks - The combination of CLDC and MIDP provides a complete environment for creating applications on cell phones and simple two-way pagers.

The core of a MID Profile is a MIDlet application. The application extends the MIDlet class to allow the application management software to control the MIDlet, retrieve properties from the application descriptor, and notify and request state changes.

All MIDlets extend the MIDlet class -- the interface between the runtime environment (the application manager) and the MIDlet application code. The MIDlet class provides APIs for invoking, pausing, restarting, and terminating the MIDlet application.

The application management software can manage the activities of multiple MIDlets within a runtime environment. In addition, the MIDlet can initiate some state changes by itself, and notify the application management software of those changes.

The whole set of MIDP API classes can be broken down into two categories:

- **MIDP** APIs for the user interface: These APIs are designed so that interaction with the user is based around a succession of screens, each of which presents a reasonable amount of data to the user. Commands are presented to the user on a per-screen basis. The APIs allow the application to determine what screen to display next, what computation to perform, and what request to make of a network service.
- **MIDP** APIs for handling the database: These APIs organize and manipulate the devices database, which comprises information that remains persistent across multiple invocations of the MIDlet.

The underlying CLDC API is used to handle strings, objects, and integers. A subset of the Java 2 API is also provided to handle I/O and network communications.

Figure 2-5 The building blocks of J2ME.

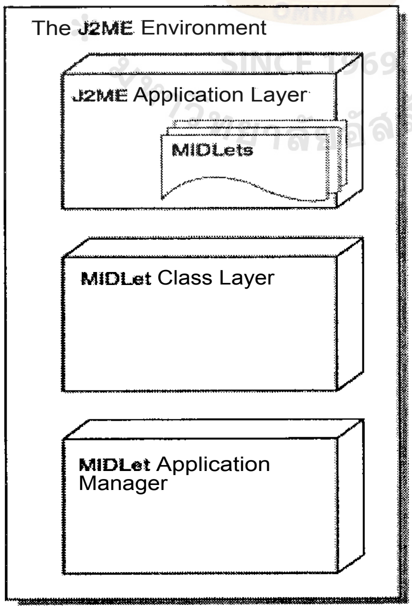
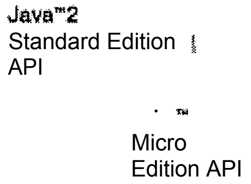


Figure 2-6 The relationship between the J2ME and J2SE APIs

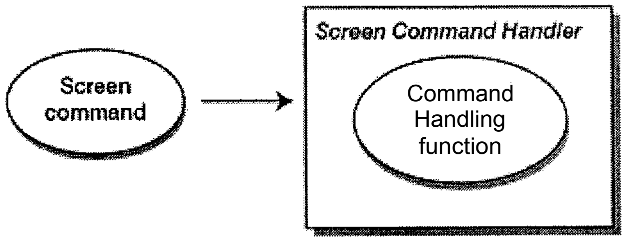


Event handling in J2ME, in contrast to event handling on the desktop version of the Java platform, is based around a succession of screens. Each screen carries a certain small amount of data.

Commands are presented to the user on a per-screen basis. The Command object encapsulates the name and information related to the semantics of an action. It is primarily used for presenting a choice of actions to the user. The resulting command behavior is defined in a CommandListener associated with the screen.

Each Command contains three pieces of information: a *label,* a *type,* and a *priority.* The label is used for the visual representation of the command; the type and priority are used by the system to determine how the Command is mapped onto a concrete user interface.

Figure 2-7 The event handling mechanism in J2ME.



2.3 Features enabled through MIDP

The MIDP specifications were defined in 1999-2000 by the above mentioned MIDPEG. Members of this group are among others: AOL, Ericsson, Fujitsu, Hitachi, J-Phone, Matsushita, Mitsubishi, Motorola, NEC, Nokia, NTT DoCoMo, Palm, Samsung, Sharp, Siemens, Sony, Sun Microsystems and Symbian.

The main goal of the MIDPEG is to establish an open, third party application development environment for MIDs (Mobile Information Devices).

MIDs span a potentially wide set of capabilities. Rather than try to address all such capabilities, the MIDPEG agreed to limit the set of APIs specified, addressing only those APIs that were considered absolute requirements to achieve broad portability. These APIs are:

2.3.1 Applications

Applications that run on devices supporting MIDP are called MIDlets. The Application API defines the interaction between MIDlets and the MID (e.g. their installation) and how MIDlets are controlled.

A MIDlet consists of

- a descriptor (containing application attributes)
- Java classes
- resources (bitmaps etc.)

Handling of applications is performed by the Java Application Manager (JAM). The JAM manages the installation, launch, version management and removal of MIDlets.

2.3.2 User interface, or UI

Defines keyboard input, display management, vibra and sound usage.

Although it maintains a constrained profile, the MIDP API provides a complete set of UI elements. The following are some of the most important ones:

- An Alert acts as a screen to provide information to the user about an exceptional condition or error.
- A Choice implements a selection from a predefined number of choices.
- A ChoiceGroup provides a group of related choices.
- A Form acts as a container for the other UI elements.
- A List provides a list of choices.
- A StringItem acts as a display-only string.
- A TextBox is a screen that allows the user to enter and edit text.
- A TextField allows the user to enter and edit text. Multiple TextFields can be placed in a Form.
- A DateField is an editable component for presenting date and time information. A DateField can be placed in a Form.
- A Ticker acts as a scrollable display of text.

2.3.3 Persistent storage

This API provides a mechanism for MIDlets to persistently store data and retrieve them later. This persistent storage mechanism, called the Record Management System (RMS), is modelled after a simple record-oriented database. It is used by database-oriented applications like phonebook, recipes, etc.

The MIDP provides a set of classes and interfaces to organize and manipulate a device's database: RecordStore, RecordComparator, and RecordFilter. A RecordStore consists of a collection of records, which remain persistent across multiple invocations of the MIDlet. Comparing records in a RecordStore or extracting sets of records from a RecordStore is functionality provided by RecordComparator and RecordFilter interfaces.

2.3.4 Networking

Ensures the support of commonly used protocols like HTTP and WAP, required for data transfer and interaction. This is very important for transaction-based applications like banking, navigation, etc., and also to make downloads possible.

2.3.5 Timers

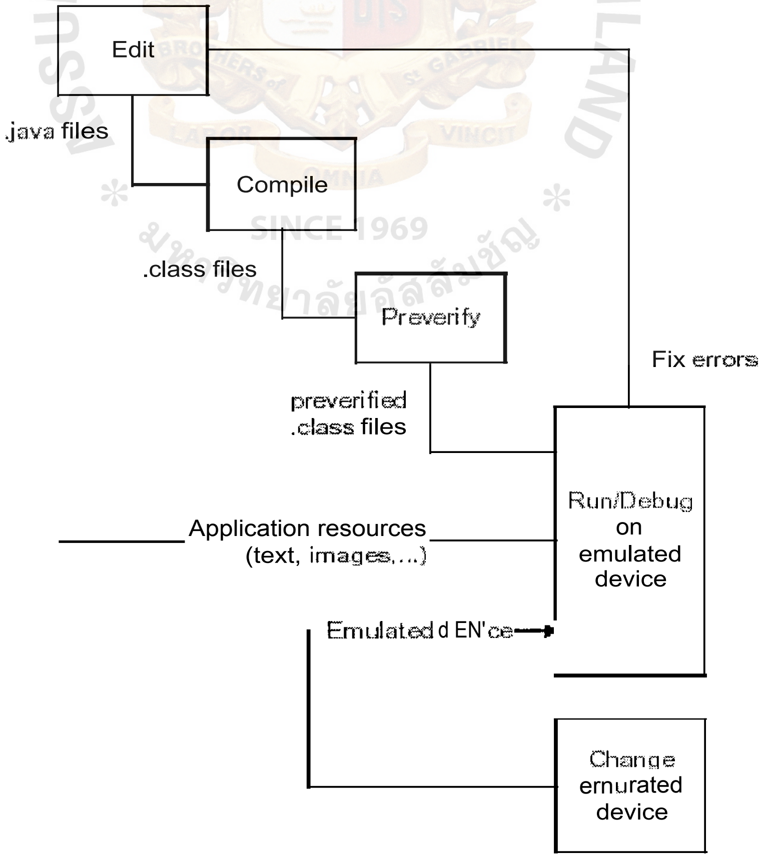
The MIDP adds functions that allow an application to set timers and be notified of their expiration. This is extremely important for PIM (Personal Information Management) applications.

CHAPTER 3

DEVELOPING APPLICATION WITH J2ME

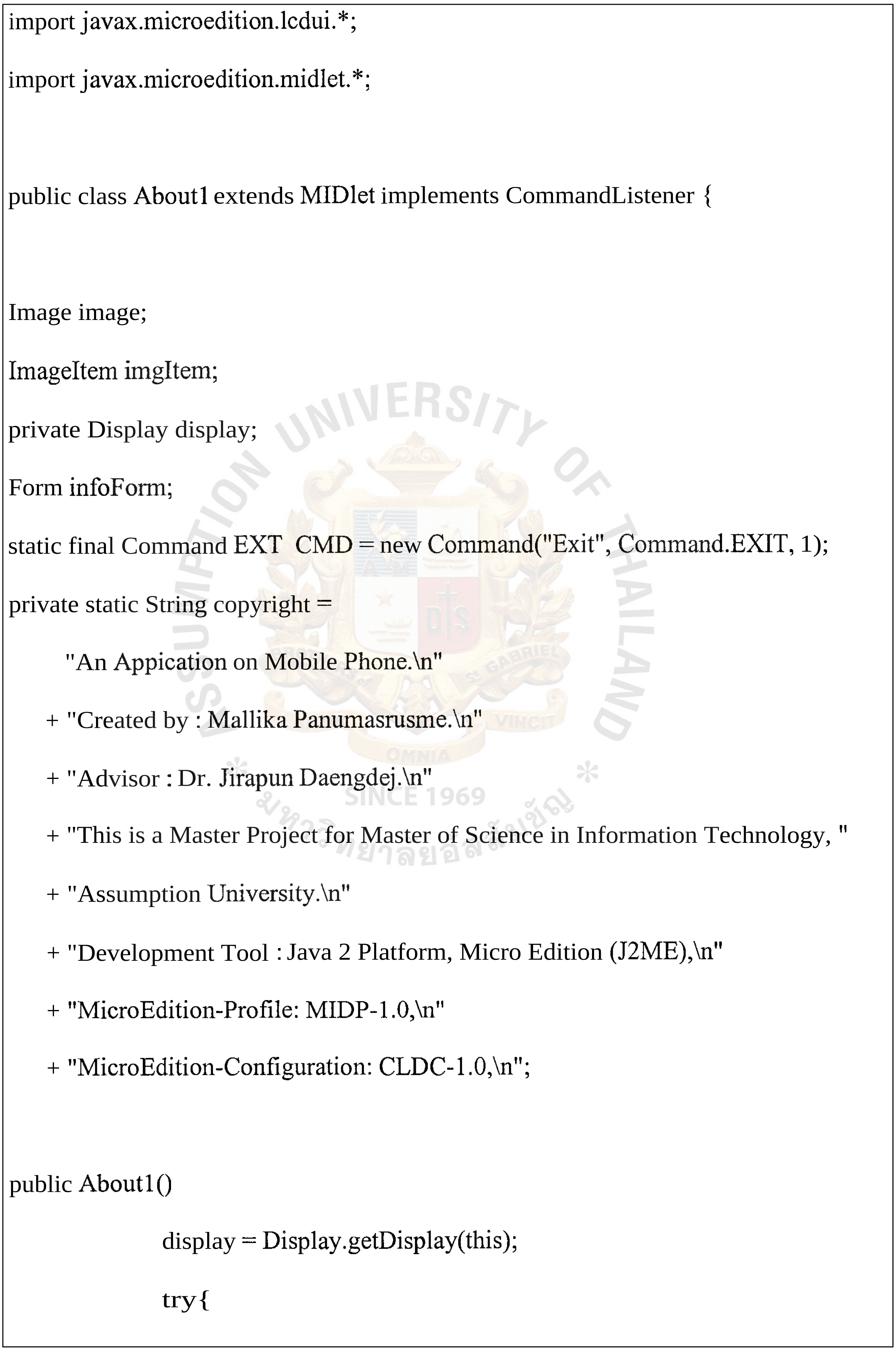
We can develop the application by running at the command line by ourselves, or by relying on development environments that automate a large part by using J2ME Wireless Toolkit. I decide to develop the application in the command line myself. The reason is that although development with the J2ME Wireless Toolkit provide even more convenience such as it allows you to compile, package, and execute or debug MIDP applications, it requires a lot of memory and the process is slow. The phases are illustrated in the following diagrams.

Figure 3-1 Developing and Testing application



3.1 Coding

Write program (.java files) by using EditPlus Text Editor. For example :-



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

import javax.microedition.midlet.*;



public class About1 extends MIDlet implements CommandListener {



Image image;

ImageItem imgItem;

private Display display;

Form infoForm;

static final Command EXT  CMD = new Command("Exit", Command.EXIT, 1);

private static String copyright =

      "An Appication on Mobile Phone.\n"

    + "Created by : Mallika Panumasrusme.\n"

    + "Advisor : Dr. Jirapun Daengdej.\n"

    + "This is a Master Project for Master of Science in Information Technology, "

    + "Assumption University.\n"

    + "Development Tool : Java 2 Platform, Micro Edition (J2ME),\n"

    + "MicroEdition-Profile: MIDP-1.0,\n"

    + "MicroEdition-Configuration: CLDC-1.0,\n";



public About1()

                display = Display.getDisplay(this);

                try{
                        image = Image.createImage("/nu9.png");
                } catch (java.io.IOException x) {
                // just don't append the image.
                }
                imgItem = new ImageItem( null, image, 3 , null );
   }

public void startApp() {
                infoForm = new Form("About Project");

                infoForm.append(new StringItem(null, copyright));
                infoForm.append(imgItem);
                infoForm.addCommand(EXT_CMD);
                infoForm.setCommandListener(this);
                display.setCurrent(infoForm);
   }

public void destroyApp(boolean unconditional) {
}

public void pauseApp() {
}

public void commandAction(Command c, Displayable s) {
```

```
            if (c == EXT_CMD)
                                    destroyApp(false);
                                    notifyDestroyed();
                  }
         }
}
```

3.2 Compilation and **Preverification**

Create the class files by using the Java 2 SDK, Standard Edition (J2SETM SDK) compiler. Open the command prompt in the directory that store java files and run command:-

```
C:\project\ap>  javac -bootclasspath c:\j2me\midp1.0.3fcs classes Ap1.java
```

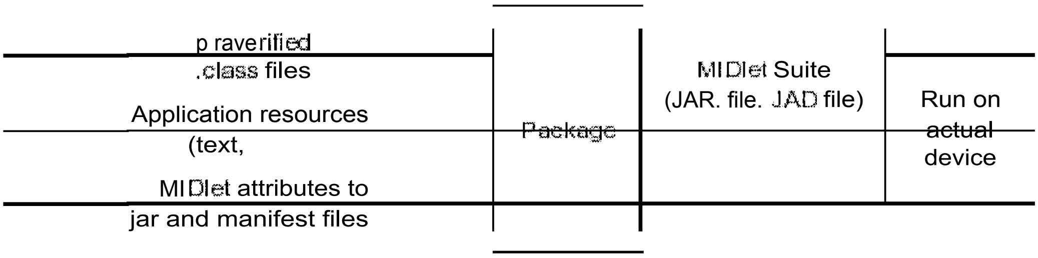
After compiling the sources, we will get the class files. Then we must preverify all class files. Preverifying is nesccessary for file loading into the portable device as command:-

```
C:\project\ap>  preverify -classpath c:\j2me\midp1.0.3fcs\classes;. -d . Ap1
```

3.3 Packaging

MIDP applications, or MIDlets, are packaged into a *MIDlet suite,* a grouping of MIDlets that can share resources at runtime. The following diagram illustrates how a MIDlet suite is organized.

Figure 3-2 Pakaging an Application



More formally, a MIDlet suite includes:

- **A Java Application Descriptor (JAD) file.** This file contains a predefined set of attributes (denoted by names that begin with “MIDlet-”) that allow application management software to identify, retrieve, and install the MIDlets. All attributes appearing in the JAD file are made available to the MIDlets. You can define your own application-specific attributes and add them to the JAD file.
- **A Java Archive (JAR) file.** This contains:
  - Java classes for each MIDlet in the suite.
    Java classes shared between MIDlets.
  - Resource files used by the MIDlets (for example, image files).
  - A manifest file describing the JAR contents and specifying attributes used by application management software to identify and install the MIDlet suite.

Copy all preverified class files into subdirectory class and create a jar file as command :-

```
C:\project\ap   jar cv I Ap  .jar -C c:\project\ap\class .
```

Create a j ad file by using EditPlus Text Editor as follows:-

```
MIDlet-Name: Project Application

MIDlet-Version: 1.0

MIDlet-Vendor: Core J2ME Technology

MIDlet-Description: Project Application

MIDlet-Jar-URL: Project.jar

MIDlet-Jar-Size: 90368

MIDlet-1: Appointment, /Cal3.png, Ap1

MIDlet-2: First Aid, /Plus.png, Aid

MIDlet-3: About Project, /App.png, About1
```

3.4 Running and Debugging

Run the j ad file as command :-

```
C:\project\ap   midp -classpath Project   ar -Xdescriptor Project.jad
```

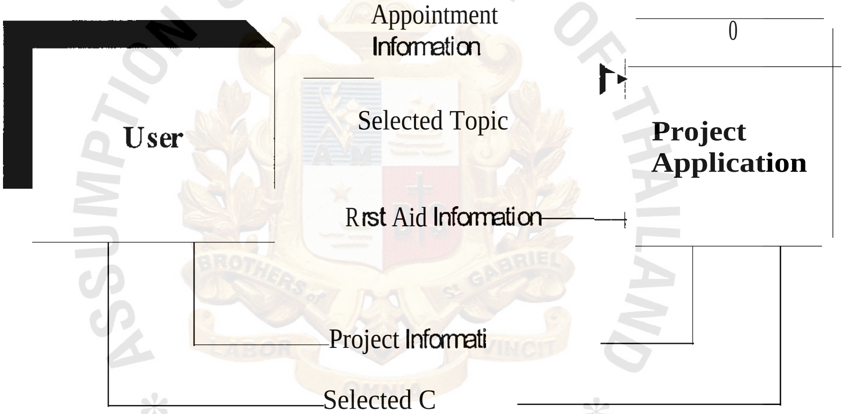
CHAPTER 4

ANALYSIS AND DESIGN

The Project Application is consist of 3 parts :- Appointment application, First Aid application and project information.

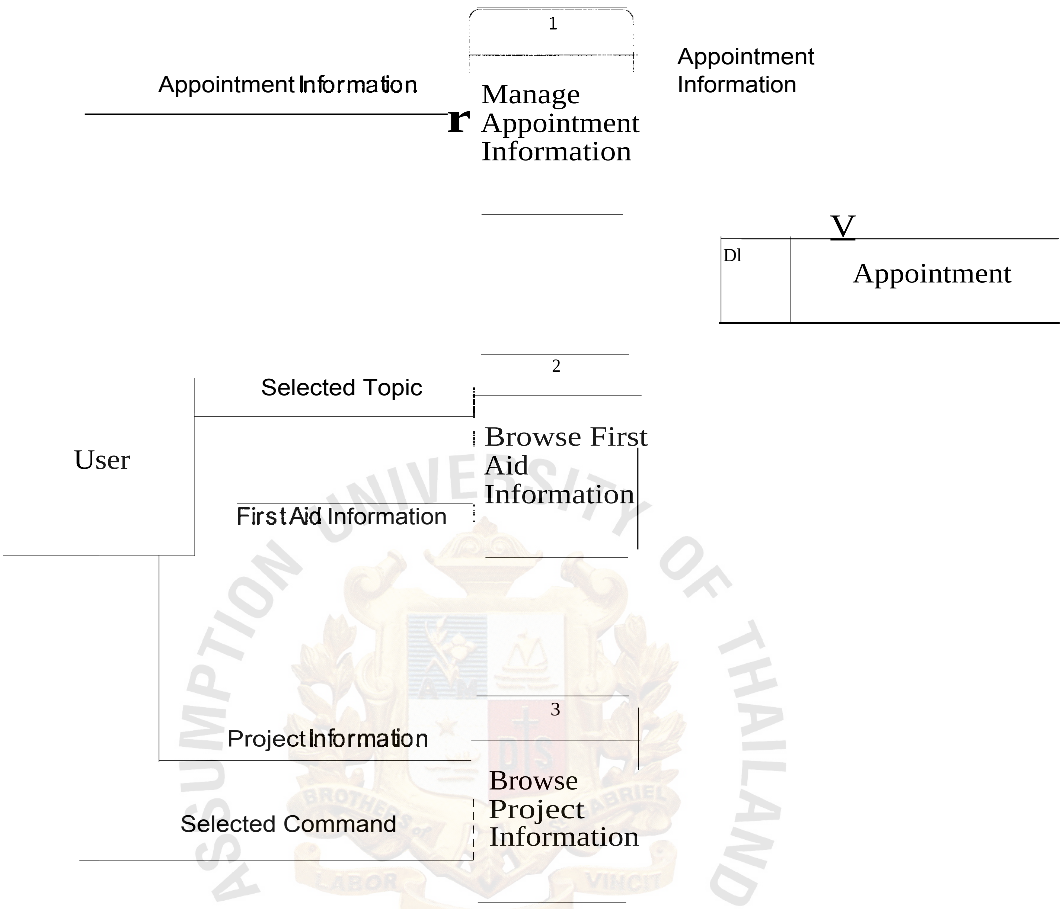
4.1 Context Diagram

Figure 4-1 Context Diagram



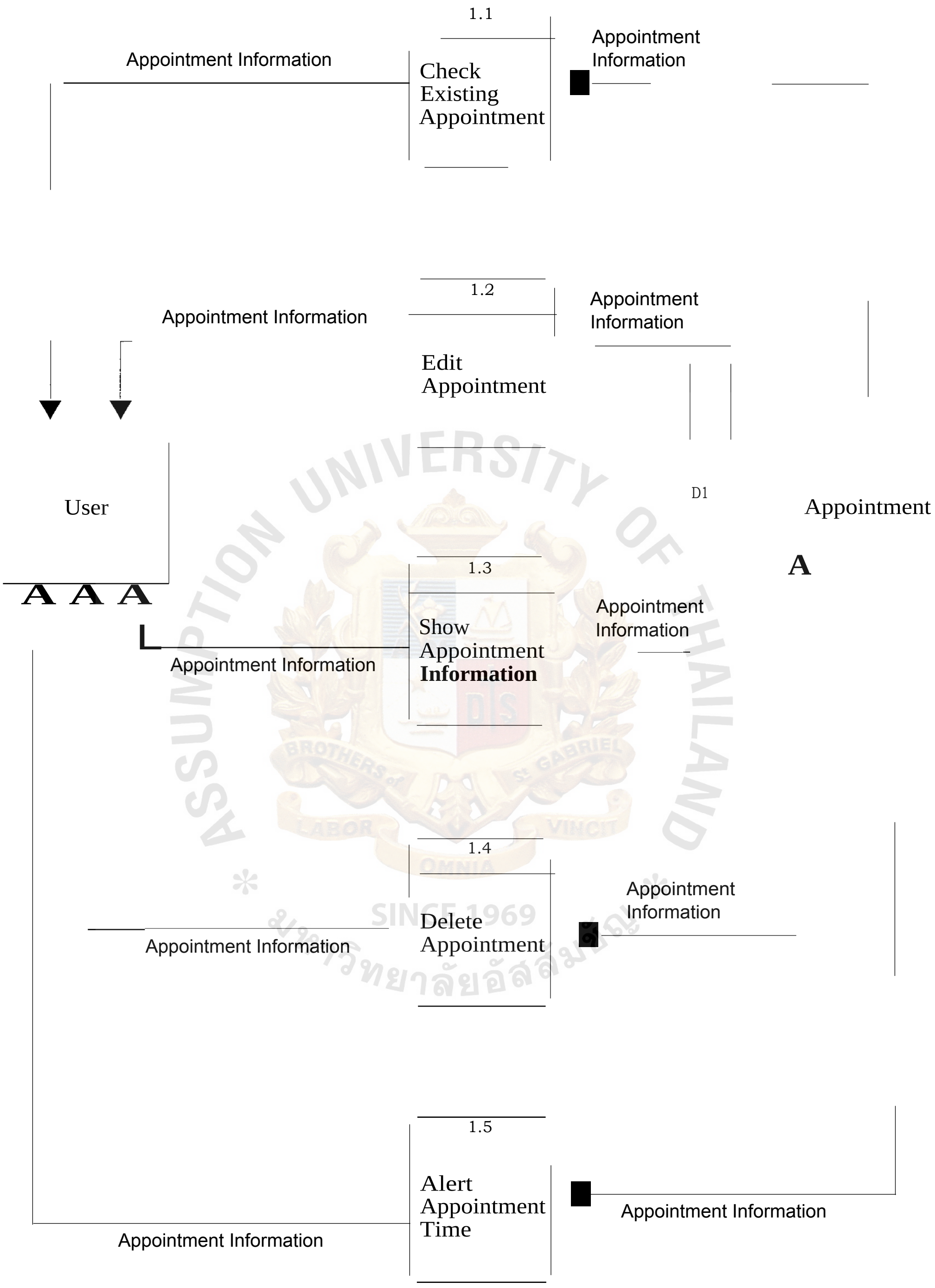
4.2 Data Flow Diagram level 0

Figure 4-2 Data Flow Diagram level 0



4.3 Data Flow Diagram level 1

Figure 4-3 Data Flow Diagram level 1



4.4 Record Format of Appointment application

The record that stores the appointment information including of data field :

Table 4-1 Appointment record format

Field name	Type	Size	Format
Date	Date	10	“nnnnynnmnn”
Time	Time	8	“hh:mm AM" or “hh:mm PM"
Name	String	35	
Surname	String	35	
Place	String	35	
Building	String	35	
Floor	Numeric	3	
Room	String	10	
Malady	String	35	
Telephone no.	String	**10**	
Alert Time	Time	8	“hh:mm AM" or “hh:mm PM"
Comment	String	35	

4.5 Application memory

With the limitation of memory of mobile phone device so the important part we must concentrate in design phase is the phone memory that is available for J2ME applications installation. Such as

- Nokia 7250 provide 3 pre-set, max download size 64KB per application.
- Motorola V601 provide 917KB dedicated memory for applications. Max application size is approximately 150KB. There is 128KB dedicated memory for 'temp files': high scores, application data, etc.

And another higher memory phone devices: Nokia 3650 with 4MB internal dynamic memory for phone numbers, messages, photos and External memory on 3V MMC memory card.

For advanced technology, The P 800 is the most advanced, latest offer cell phone, from SonyEricsson which provide 16 + 16 MB flash memory, 12 MB available for images, contacts etc. and expandable memory: Memory Stick DUO. The P800 comes with a 16MB Memory Stick DUO which more than doubles the storage capacity and allows you to transfer files from your phone to your PC and vice versa. Use it to: backup media, Transfer between PC and P800 or between P800 and another P800 and distribute software.

4.6 Project application user interface screen

Figure 4-4 Main menu screen

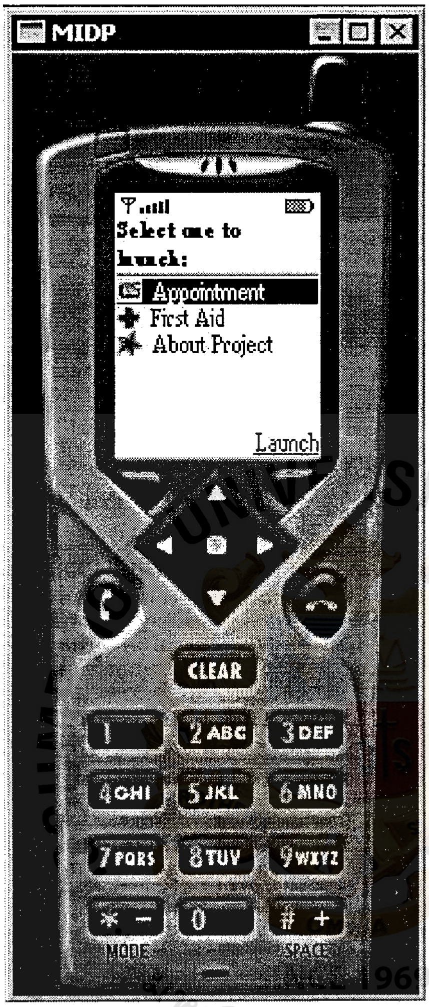


Figure 4-5 Appointment application first page

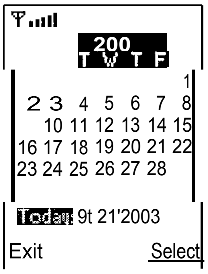
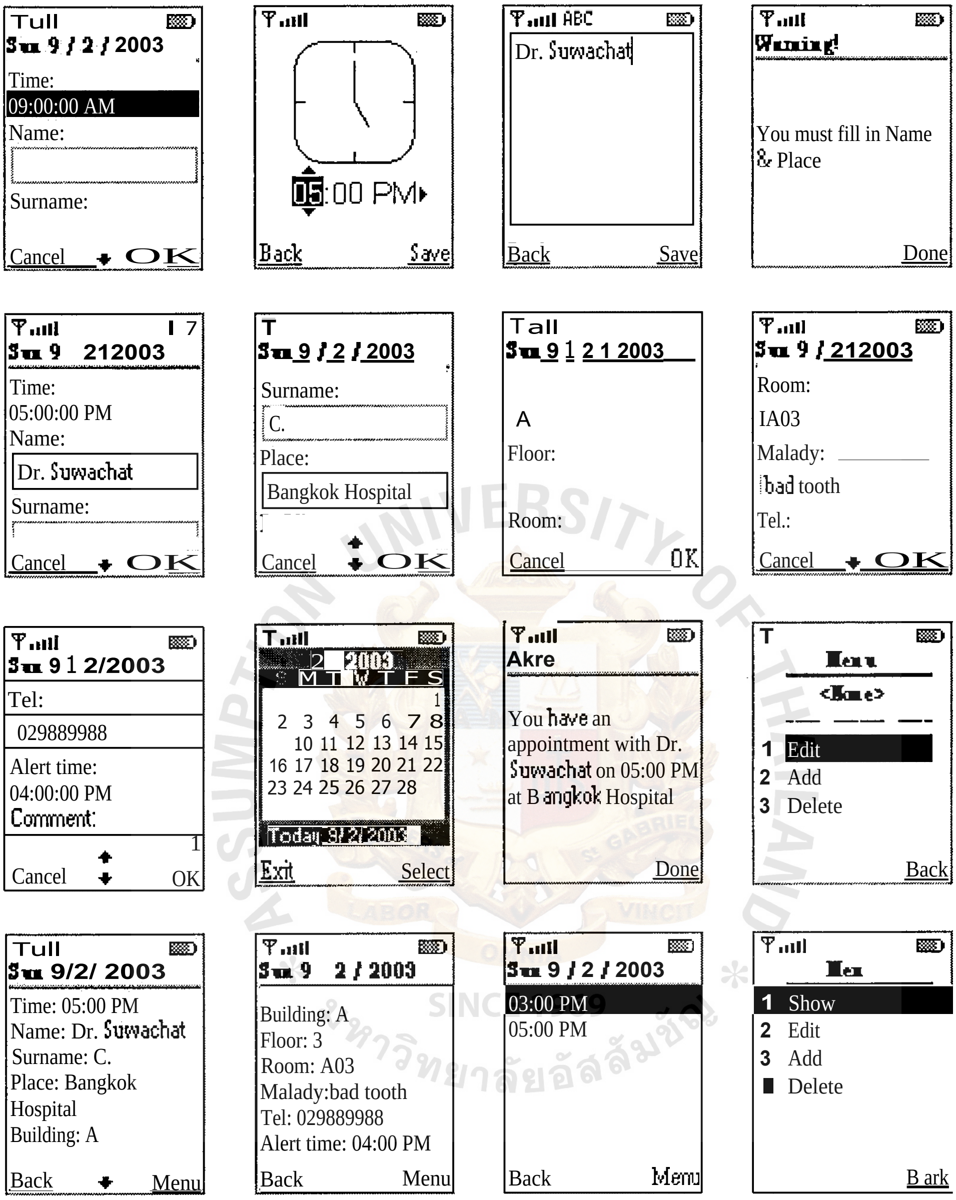


Figure 4-6 Appointment application page


Tull
Su 9 / 2 / 2003
Time:
09:00:00 AM
Name:
Surname:
Cancel OK

05:00 PM
Back Save

ABC
Dr. Suwachat
Back Save

Warning!
You must fill in Name & Place
Done

Su 9 212003
Time:
05:00:00 PM
Name:
Dr. Suwachat
Surname:
Cancel OK

Su 9 / 2 / 2003
Surname:
C.
Place:
Bangkok Hospital
Cancel OK

Tall
Su 9 1 2 1 2003
A
Floor:
Room:
Cancel OK

Su 9 / 212003
Room:
IA03
Malady:
bad tooth
Tel.:
Cancel OK

Su 9 1 2/2003
Tel:
029889988
Alert time:
04:00:00 PM
Comment:
Cancel OK

2 2003
S M T W T F S
1
2 3 4 5 6 7 8
10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28
Today 9/2/2003
Exit Select

Akre
You have an appointment with Dr. Suwachat on 05:00 PM at Bangkok Hospital
Done

Menu
<Name>
1 Edit
2 Add
3 Delete
Back

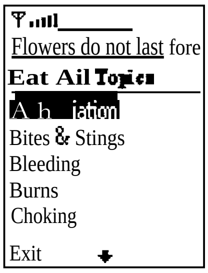
Tull
Su 9/2/ 2003
Time: 05:00 PM
Name: Dr. Suwachat
Surname: C.
Place: Bangkok Hospital
Building: A
Back Menu

Su 9 2 / 2003
Building: A
Floor: 3
Room: A03
Malady:bad tooth
Tel: 029889988
Alert time: 04:00 PM
Back Menu

Su 9 / 2 / 2003
03:00 PM
05:00 PM
Back Menu

Men
1 Show
2 Edit
3 Add
Delete
Bark

Figure 4-7 First Aid application first screen



Flowers do not last fore

Eat Ail Topics

A h iation

Bites & Stings

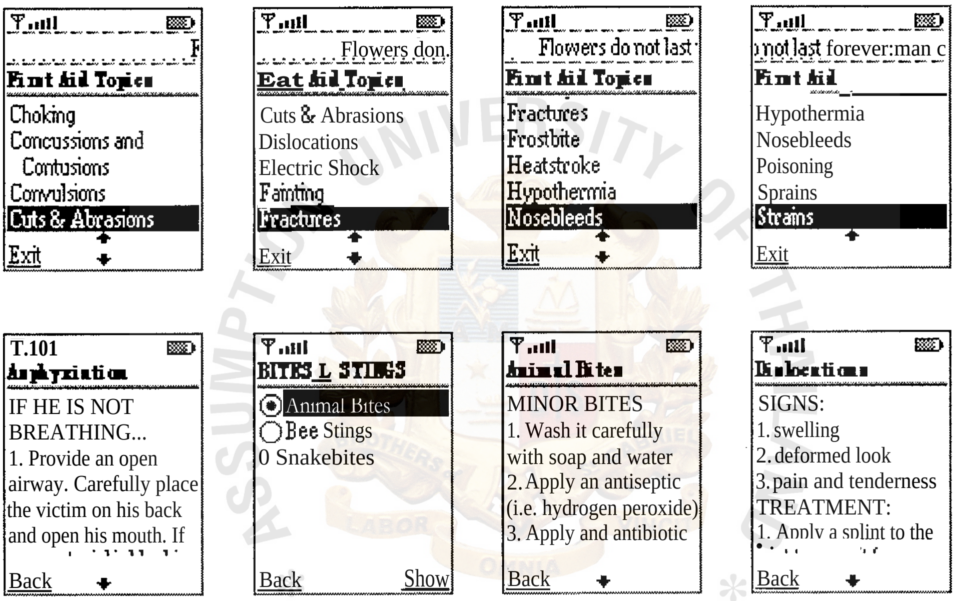
Bleeding

Burns

Choking

Exit

Figure 4-8 First Aid application screen



F

First Aid Topics

Choking

Concussions and Contusions

Convulsions

Cuts & Abrasions

Exit

Flowers don.

Eat Aid Topics

Cuts & Abrasions

Dislocations

Electric Shock

Fainting

Fractures

Exit

Flowers do not last

First Aid Topics

Fractures

Frostbite

Heatstroke

Hypothermia

Nosebleeds

Exit

) not last forever:man c

First Aid

Hypothermia

Nosebleeds

Poisoning

Sprains

Strains

Exit

T.101

Asphyxiation

IF HE IS NOT BREATHING...

1. Provide an open airway. Carefully place the victim on his back and open his mouth. If

Back

BITES L STINGS

● Animal Bites

○ Bee Stings

0 Snakebites

Back Show

Animal Bites

MINOR BITES

1. Wash it carefully with soap and water

2. Apply an antiseptic (i.e. hydrogen peroxide)

3. Apply and antibiotic

Back

Dislocations

SIGNS:

1. swelling

2. deformed look

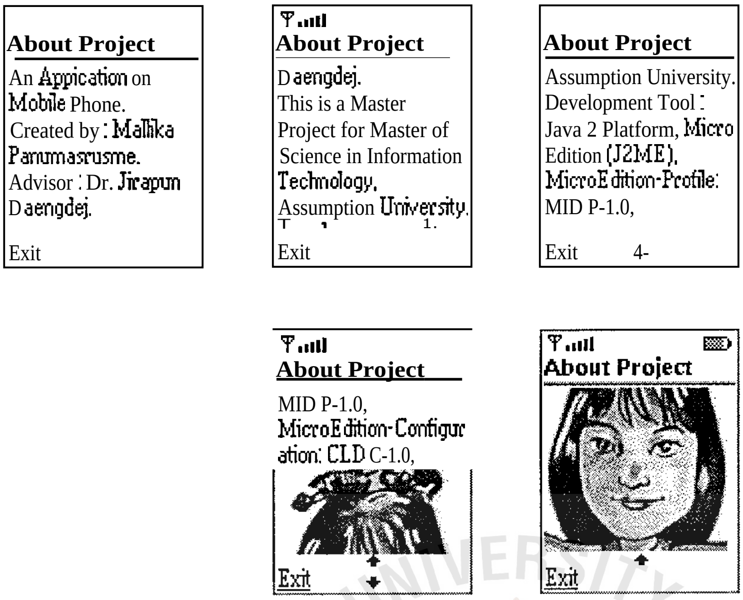
3. pain and tenderness

TREATMENT:

1. Apply a splint to the

Back

Figure 4-9 About Project ap plication screen



About Project

An Appication on Mobile Phone.
Created by : Mallika Panumasrusme.
Advisor : Dr. Jirapun Daengdej.

Exit

About Project

Daengdej.
This is a Master Project for Master of Science in Information Technology, Assumption University.

Exit

About Project

Assumption University.
Development Tool : Java 2 Platform, Micro Edition (J2ME), MicroEdition-Profile: MID P-1.0,

Exit 4-

About Project

MID P-1.0,
MicroEdition-Configuration: CLD C-1.0,

Exit

About Project

Exit

CHAPTER 5

SYSTEM ARCHITECTURE AND IMPLEMENTIONS

5.1 System Requirements

1. Operating system : Window 98 SE, Windows XP, Windows NT 4.0 or Window 2000.
2. Memory : RAM 64 MB
3. Disk space : 30 MB
4. Development Tools :

   - Java Development Kit (JDK) contain the compiler for compiling Java source code and utility to create Java Archive (JAR) file. We can download Java^{TM} 2 SDK, Standard Edition Version 1.3.1 from http://java.sun.com/products/jdk/1.3.1/.

     Connected Limited Device Configuration (CLDC) provide subset of Java 2 Standard Edition (J2SE) which combine the several classes from

     - Java.io
     - Java.lang
     - Java.util

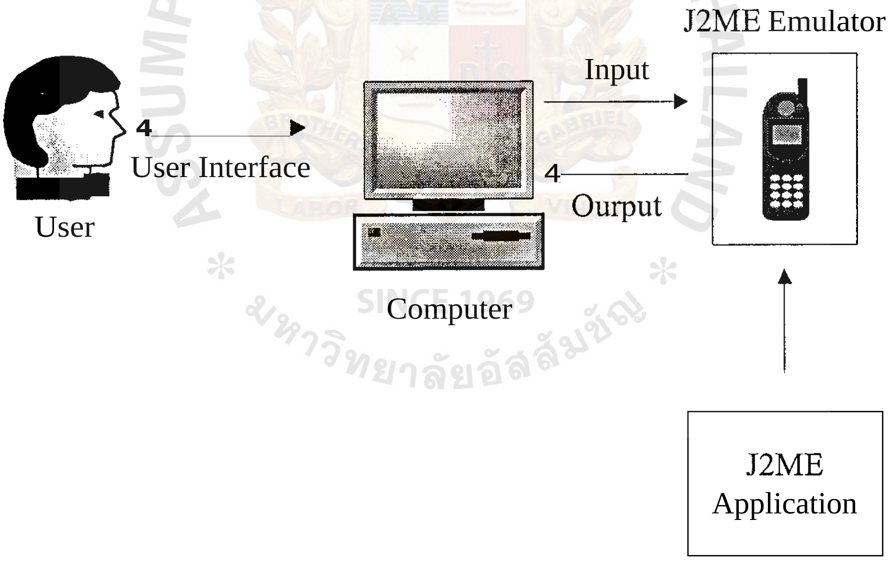
     Otherwise javax.microedition.io is included in CLDC which provide classes and several interface to help to access to the data storage and network system. We can download CLDC versioin 1.0.3 from http://java.sun.com/products/cldc.

     Mobile Information Device Profile which supports any device containing limited screen (for example, 96 pixels width and 54 pixels length), memory and processing power. We can download MIDP version 1.0.3 from http://java.sun.com/products/midp.

- J2ME Wireless Toolkit version 1.0.3 or higher that can be freely downloaded at http://java.sun.com/products/j2mewtookit/.
- EditPlus Text Editor version 2.10c.
- In addition, there are many tools for J2ME development that are described in Appendix C.

5.2 System Architecture

Figure 5-1 System architecture



J2ME Application run on devices supporting MIDP called MIDlets. Unlike traditional Java applets and applications, which can be executed using a Web browser or Java interpreter during the development phase, MIDlets must be executed using a special emulator that emulates a physical mobile device. A J2ME emulator is an important tool that ships with both the standard J2ME Wireless Toolkit and with other similar toolkits that are made available by device vendors such as Motorola. Although it is certainly possible to test a MIDlet directly on a mobile device, an emulator streamlines the process and enables you to work entirely on a desktop computer throughout the development process. It is still important to test a MIDlet on a physical device as it nears completion, but the emulator plays a critical role in testing a MIDlet prior to that point.

The J2ME emulator is the tool that enables you to run MIDlets on a desktop PC and simulate how the MIDlet will run on a physical device. Before getting into the specifics of how to run MIDlets within the emulator, I'd like to go over some of the benefits of a J2ME emulator, along with a few of its limitations. Following are three primary benefits of using a J2ME emulator:

1. You can delay testing on a physical device until the final stage of MIDlet development.

2. You can test a MIDlet on a variety of different target devices, including custom devices.

3. You can track specific aspects of a MIDlet's execution such as class loading, method calls, and garbage collection.

The first benefit is that it reflects the fact that the emulator serves as a great stand-in for a physical device during the early and middle stages of MIDlet development; you'll still want to run a MIDlet through its paces on a physical device late in the development process.

The second benefit alludes to the fact that the emulator operates with respect to a device profile. The emulator is very flexible in that it enables you to test a MIDlet on multiple device profiles, which effectively tests the MIDlet on multiple devices.

You can even define custom devices (through custom profiles) and test MIDlets on them.

The last benefit listed has to do with the emulator's capability to provide diagnostic information about a MIDlet as it is executing.

Before you begin to think that the J2ME emulator is the most amazing development tool ever envisioned, I need to temper your excitement by highlighting a few of its limitations. None of these limitations are killers, but it is important to understand where the emulator falls short of a physical device in terms of testing MIDlets. Then you can focus on these aspects of a MIDlet when you do test it on a physical device. Following are the aspects of a MIDlet that cannot be controlled or tested by the emulator:

- Execution speed
- Available memory
- Application manager

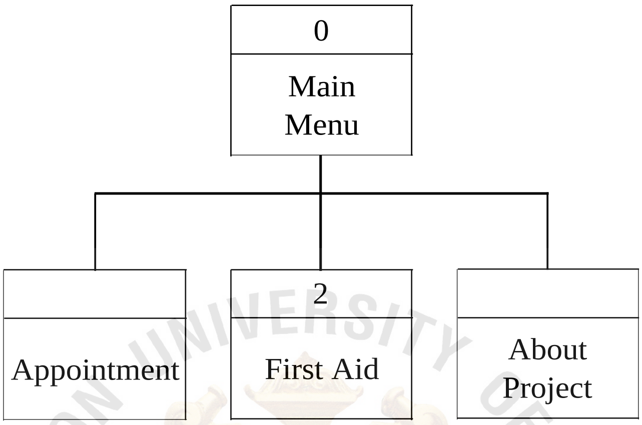
Physical mobile devices vary widely in their hardware, which means that processor speed is most definitely a variable that cannot be nailed down by a MIDlet developer unless you target a very specific device and ignore others. Even though you can't necessarily make assumptions about the speed of the processor in a device, it would be nice to be able to test a MIDlet at a variety of different speeds to see how it responds. Unfortunately, the J2ME emulator doesn't currently factor device speed into the emulation process. So you'll have to resort to testing a MIDlet on physical devices when it comes to assessing the speed of execution across different processors. The processor speed will likely vary considerably across different mobile devices, and the memory available to a MIDlet will quite likely vary as well. Because available memory can dramatically impact the execution of a MIDlet, it is important to test a

MIDlet within the memory constraints of each target device. Unfortunately, the J2ME emulator doesn't currently enable you to vary the memory available to a MIDlet, so it doesn't help much in terms of assessing the impact of available memory on a MIDlet. Chalk up memory as another aspect of your MIDlet that you'll need to test directly on a physical device, along with the speed of execution.

The last limitation of the J2ME emulator is less critical than the first two, and has to do with the application manager that is used on mobile devices to manage MIDlets. The application manager on a device is responsible for allowing you to install, remove, and execute individual MIDlets. Currently no support is available for application management in the emulator, which means that you can't completely test the management of MIDlets on a device using the emulator. However, this is a task that is easy enough to test on a physical device when your MIDlet is ready for deployment. Beyond these easily identifiable weaknesses in the J2ME emulator, you also must consider the fact that any emulator is merely approximating a physical device. This means that an emulator will inevitably yield slightly different results from a real device, which is why you can't rely 100% on emulators for all J2ME software testing. So, although emulators serve a great role in enabling you to accomplish a great deal of developmental testing on a PC, you must always plan on spending ample time testing on a physical device.

5.3 Application Prototype

Figure 5-2 Application prototype



5.4 Application detail

Project application main menu consists of 3 topics :- Appointment, Fist Aid and About Project.

- **Appointment**

Objective : Provide a calendar form of today month for recording the appointment on the specific date.

Algorithm

1. User can see the calendar in the first page and a square circular today's day, and a text that shows today's date.
2. Press the **Arrow** key to the top line to change month and year of calendar form.
3. Press the **Select** button to the specific date to deal with appointment information.

- If there is not a record store in that date, the system prompt the input form. The system will set current time to appointment time and alert time at the initial screen. If user does not input the alert time the system will set the alert time to an hour before appointment time in the record saving. User must input the information at least name and place before pressing the **OK** button for saving record. If the user pressed **OK** button when he did not complete filling name and place, the system will prompt alert screen. Press **Cancel** button to return to the calendar form without update.
- If there is an appointment record in that specific day then the system will prompt the browsing form (can not change any information) and the form has a **Menu** button that consists of **Edit, Add** and **Delete** command. Select **Edit** command to update the current appointment information. Select **Add** command to add new appointment information. Select **Delete** command to delete current appointment record. Press **Back** button to return to the previous page.
- If there are more than one record in the specific day then the system will show the list of sort appointment time and **Menu** button. Press **Menu** button to show the list of commands **Show, Edit, Add** and **Delete.** Select **Show** command to browse appointment information. And other commands do the same process as previously mentioned.

4. The system uses the timer to check alert time of that day appointment in every second. The current time is the same time as the alert time in the appointment record. So the system prompts an appointment message to the user. Press Done to return to the previous page.
5. Press Exit button to return to main menu.

- First Aid

Objective : Provide the first aid information browsing.

Algorithm

1. User can see the symptoms of a disease topics list in the first page.
2. There is a text ticker running in the first page.
3. Select the topic to see the detail information.
4. Press Exit button to return to main menu.

- About Project

Objective : Provide user the project information.

Algorithm

1. Show the information of the project development in the first page.
2. Press Exit button to return to main menu.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

The present research intended to develop an Application on mobile phone with J2ME as development tool by testing application in the Emulator.

The Java 2 Platform, Micro Edition (J2ME) offers great tools for developers, porting the Java platform's network-centric and platform-agnostic worldview down to memory- and processor-limited devices.

Personalized and intelligent information appliances are necessities in our life today. These appliances, which include cell phones, two-way pagers, smart cards, personal organizers, and palmtops, tend to be special-purpose, limited-resource, network-connected devices, and not the general-purpose desktops we have known until now. To specifically address this vast consumer space, the Java 2 Platform, Micro Edition (J2ME) provides a plethora of innovative Java technologies.

J2ME is a significant wireless move from the portable, network-centric Java Virtual Machine. The flexibility in development and deployment of J2ME applications will efficiently cater to the increasing requirements of the wireless world.

There are the advantages and disadvantages of this study that can be concluded as follows:

Advantages:

- Appointment Application on mobile phone helps you to manage time more efficiently.
- First Aid Application on mobile phone will make people help other person or themselves promptly.

- With J2ME application development, we can create or update the information on mobile phone whatever we want that for the effective use, life convenience and more business opportunities.
- The J2ME application provides the upgrade or adjust application on mobile phone to meet the user needs instead of purchasing a new one.
- We can easily and quickly access information via mobile phone anywhere and anytime.

Disadvantages:

- There is no standard operation on the different mobile phone devices. Some application that are created for the specific device may not be compatible with another mobile phone device.
- It consumes time to study and compose J2ME program.
- Some limitations are small screen display, hard to input data, tiny kilobyte memory and not high processor ability (in most 16-bit processors) of mobile phone.

6.2 Recommendations and Future Works

The result of this project is to propose the method to develop the application on mobile phone with J2ME.

With the limitation of time to develop the system, the process has been developed for browsing the application at the default emulator. For further work including :

- Develop application by using J2ME Wireless Toolkit (new version 1.0.4) which provides the user convenience for development and features such as Memory Monitoring, Network monitoring, Device speed emulation.

- Develop application for the specific mobile phone device and install for using on real mobile phone.

REFERENCES

"Introducing the SL45i/6688I." White Paper-Siemens AG, 2001.

Ghosh, Soma."Think small with J2ME." <http://www.midletzone.com>, November, 2001.

"J2ME[tm] Devices." <http://wireless.java.sun.com/device/J2ME[tm] Devices.htm>, November 27, 2002.

<http://expage.com/page/lacieking>, Pacific time, February 22, 2002.

S., Mubashir and P., Azam. "SONY Ericson P800." <http://welectronics.com/ericson p800.Shtml.htm>, OVERSEAS Electronics.

"Nokia 3650 SIM Free." <http://www.mphone.co.uk/Nokia/3650.html>, mPhone L td.

"Buy Nokia 7250 SIM Free mPhone online." <http://www.mphone.co.uk/Nokia/7250 .html>, mPhone Ltd.

"Motorola V60i SIM Free." <http://www.mphone.co.uk/Motorola/v60i.html>, mPhon e Ltd.

APPENDIX A: USER GUIDE

Environment Setting:

1. Install Java Development Kit.
2. Install CLDC into directory c:\j2me\j2me_cldc.
3. Install MIDP into directory c:\j2me\midp1.0.3fcs.
4. Install EditPlus text editor program.
5. In addition, install J2ME Wireless Toolkit into directory c:\j2me for seeing the example application.
6. Add the following lines into your autoexec.bat.

```
SET PATH=c:\jdk1.3.1_03\bin;c:\j2me\midp1.0.3fcs\bin

SET CLASSPATH=c:\j2me\midp1.0.3fcs\classes;.

SET MIDP  HOME=c:\j2me\midp1.0.3fcs
```

APPENDIX B

J2ME DEVICES

Manufacturer	Model	Wireless Technology	Frequency (MHz)	Software	Screen	Available
Casio	A3012CA	CDMA2000 1X	800	MIDP 1.0, CLDC 1.0	132x176/14 bits	Yes
Casio	C452CA	CDMA	800	MIDP 1.0, CLDC 1.0	120x133/8 bits	Yes
Fujitsu	F503i	PDC	800	CLDC 1.0	120x130/8 bits	Yes
Fujitsu	F503iS	PDC	800	CLDC 1.0	120x130/10 bits	Yes
Fujitsu	F504i	PDC	800	CLDC 1.0	16 bits	Yes
Hitachi	C3001H	CDMA	800	MIDP 1.0, CLDC 1.0	120x162/12 bits	Yes
Hitachi	C451H	CDMA	800	MIDP 1.0, CLDC 1.0	120x143/8 bits	Yes
Kyocera	C3002K	CDMA	800	MIDP 1.0, CLDC 1.0	120x160/16 bits	Yes
LG Electronics	C-nain 2000	CDMA2000 1X	800	MIDP 1.0, CLDC 1.0	120x133/8 bits	Yes
LG Electronics	C-nain 2100	CDMA2000 1X	800	MIDP 1.0, CLDC 1.0	8 bits	Yes
LG Electronics	CX-300L	CDMA	1900	CLDC 1.0	120x160/8 bits	Yes
LG Electronics	Cyber-ez-X1	CDMA	1900	CLDC 1.0	128x128/2 bits	Yes
LG Electronics	I-Book	CDMA	1900	CLDC 1.0	128x128/2 bits	Yes
LG InfoComm	LX5350	AMPS, CDMA2000 1X	800, 1900	MIDP 1.0, CLDC 1.0	120x198/16 bits	Yes
LG InfoComm	VX1	AMPS, CDMA2000 1X	800, 1900	MIDP 1.0, CLDC 1.0	128x104	Yes

Manufacturer	Model	Wireless Technology	Frequency (MHz)	Software	Screen	Available
Mitsubishi	D2101V	W-CDMA		CLDC 1.0	132x162/18 bits	Yes
Mitsubishi	D5031	PDC	800	CLDC 1.0	132x142/10 bits	Yes
Mitsubishi	D503iS	PDC	800	CLDC 1.0	132x142/10 bits	Yes
Mitsubishi	D504i	PDC	800	CLDC 1.0	18 bits	Yes
Mitsubishi	J-D05	PDC	1500	MIDP 1.0, CLDC 1.0	12 bits	Yes
Motorola	A388	GSM/GPRS	900, 1800, 1900	MIDP 1.0, CLDC 1.0	240x320/2 bits	Yes
Motorola	A820	GSM/GPRS, W-CDMA	900, 1800, 1900		176x220/12 bits	Not yet
Motorola	Accompli 008/6288	GSM/GPRS	900, 1800	MIDP 1.0, CLDC 1.0	240x320/2 bits	Yes
Motorola	Accompli 009	GSM/GPRS	900, 1800, 1900	MIDP 1.0, CLDC 1.0	240x160/8 bits	Yes
Motorola	i50sx	iDEN	800	MIDP 1.0, CLDC 1.0	111x100/2 bits	Yes
Motorola	i55sr	iDEN	800	MIDP 1.0, CLDC 1.0	111x100/2 bits	Yes
Motorola	i80s	iDEN	800	MIDP 1.0, CLDC 1.0	119x64/1 bit	Yes
Motorola	i85s	iDEN	800	MIDP 1.0, CLDC 1.0	111x100/2 bits	Yes
Motorola	i90c	iDEN	800	MIDP 1.0, CLDC 1.0	111x110/2 bits	Yes
Motorola	i95cl	iDEN	800	MIDP 1.0, CLDC 1.0	120x160/8 bits	Yes

Manufacturer	Model	Wireless Technology	Frequency (MHz)	Software	Screen	Available
Motorola	T280i	GSM/GPRS	**900, 1800, 1900**	MIDP **1.0,** CLDC **1.0**	**128x100/2 bits**	**Yes**
Motorola	T720	GSM/GPRS	**900, 1800, 1900**	MIDP **1.0,** CLDC **1.0**	**120x160/12 bits**	**Yes**
Motorola	T720	**AMPS,** CDMA2000 1X	**800, 1900**	MIDP **1.0,** CLDC **1.0**	**120x160/12 bits**	**Yes**
Motorola	V60i	GSM/GPRS	**900, 1800, 1900**	MIDP **1.0,** CLDC **1.0**	**96x64**	**Yes**
Motorola	V60i	**AMPS,** CDMA	**800, 1900**	MIDP **1.0,** CLDC **1.0**	**96x64**	**Yes**
Motorola	V60i	**AMPS,** TDMA	**800, 1900**	MIDP **1.0,** CLDC **1.0**	**96x64**	**Yes**
Motorola	V66i	GSM/GPRS	**900, 1800, 1900**	MIDP **1.0,** CLDC **1.0**	**96x64**	**Not yet**
NEC	**N2002**	W-CDMA		CLDC **1.0**	**16 bits**	**Yes**
NEC	N503i	**PDC**	**800**	CLDC **1.0**	**120x130/10 bits**	**Yes**
NEC	N503iS	**PDC**	**800**	CLDC **1.0**	**120x130/10 bits**	**Yes**
NEC	N504i	**PDC**	**800**	CLDC **1.0**	**16 bits**	**Yes**
Nokia	3410	**GSM**	**900, 1800**	MIDP **1.0,** CLDC **1.0**	**96x65/1 bit**	**Yes**
Nokia	3510i	GSM/GPRS	**900, 1800**	MIDP **1.0,** CLDC **1.0**	**96x65/12 bits**	**Not yet**
Nokia	3530	GSM/GPRS	**900, 1800**	MIDP **1.0,** CLDC **1.0**	**96x65/12 bits**	**Yes**
Nokia	3570	CDMA2000 **1X**	**1900**	MIDP **1.0**	**96x65/2 bits**	**Not yet**
Nokia	3585	**AMPS,** CDMA2000 1X	**800, 1900**	MIDP **1.0,** CLDC **1.0**	**96x65/2 bits**	**Yes**

Manufacturer	Model	Wireless Technology	Frequency (MHz)	Software	Screen	Available
Nokia	3585i	**AMPS,** CDMA2000 **1X**	**800, 1900**	MIDP **1.0,** CLDC **1.0**	**96x65/2 bits**	**Not yet**
Nokia	3590	GSM/GPRS	**850, 1900**	MIDP **1.0,** CLDC **1.0**	**96x65/1 bit**	**Yes**
Nokia	3650	**GSM**	**900, 1800, 1900**	WMA **1.0,** MMAPI **1.0,** MIDP **1.0,** CLDC **1.0**	**176x208/12 bits**	**Not yet**
Nokia	5100	GSM/GPRS	**900, 1800, 1900**	MIDP **1.0,** CLDC **1.0**	**128x128/12 bits**	**Not yet**
Nokia	6100	GSM/GPRS	**900, 1800, 1900**	MIDP **1.0,** CLDC **1.0**	**128x128/12 bits**	**Not yet**
Nokia	6200	GSM/GPRS	**850, 1800, 1900**	MIDP **1.0,** CLDC **1.0**	**128x128/12 bits**	**Not yet**
Nokia	6310i	GSM/GPRS	**900, 1800, 1900**	MIDP **1.0,** CLDC **1.0**	**95x65/1 bit**	**Yes**
Nokia	6610	GSM/GPRS	**900, 1800, 1900**	MIDP **1.0,** CLDC **1.0**	**128x128/12 bits**	**Yes**
Nokia	6650	GSM/GPRS, **W-** CDMA	**900, 1800**	MIDP **1.0,** CLDC **1.0**	**128x160/12 bits**	**Not yet**
Nokia	6800	GSM/GPRS	**900, 1800**	MIDP **1.0,** CLDC **1.0**	**128x128/12 bits**	**Not yet**
Nokia	6800	GSM/GPRS	**850, 1900**	MIDP **1.0,** CLDC **1.0**	**128x128/12 bits**	**Not yet**
Nokia	7210	GSM/GPRS	**900, 1800, 1900**	MIDP **1.0,** CLDC **1.0**	**128x128/12 bits**	**Yes**

Manufacturer	Model	Wireless Technology	Frequency (MHz)	Software	Screen	Available
Nokia	7250	GSM/GPRS	900, 1800, 1900	MIDP 1.0, CLDC 1.0	128x128/12 bits	Not yet
Nokia	7650	GSM/GPRS	900, 1800	MIDP 1.0, CLDC 1.0	176x208/12 bits	Yes
Nokia	8910i	GSM/GPRS	900, 1800	MIDP 1.0, CLDC 1.0	96x65/12 bits	Not yet
Nokia	9210 Communicator	GSM	900, 1800	MIDP 1.0, CLDC 1.0, JavaPhone 1.0, PersonalJava 1.1.1	640x200/12 bits	Yes
Nokia	9210i Communicator	GSM	900, 1800	MIDP 1.0, CLDC 1.0, JavaPhcne 1.0, PersonalJava 1.1.1	640x200/12 bits	Yes
Nokia	9290 Communicator	GSM	1900	MIDP 1.0, CLDC 1.0, JavaPhone 1.0, PersonalJava 1.1.1	640x200/12 bits	Yes
Panasonic	C3003P	CDMA	800	MIDP 1.0, CLDC 1.0	132x176/16 bits	Yes
Panasonic	P2101V	W-CDMA		CLDC 1.0	176x220/18 bits	Yes
Panasonic	P503i	PDC	800	CLDC 1.0	120x130/8 bits	Yes

Manufacturer	Model	Wireless Technology	Frequency (MHz)	Software	Screen	Available
Panasonic	P503iS	PDC	800	CLDC 1.0	120x130/8 bits	Yes
Panasonic	P504i	PDC	800	CLDC 1.0	16 bits	Yes
Research In Motion	Blackberry 5810	GSM/GPRS	1900	MIDP 1.0, CLDC 1.0	160x160/1 bit	Yes
Research In Motion	Blackberry 5820	GSM/GPRS	900, 1800	MIDP 1.0, CLDC 1.0	160x160/1 bit	Yes
Samsung	SCH-X130	CDMA2000 1X	800	MIDP 1.0, CLDC 1.0	128x128/2 bits	Yes
Samsung	SCH-X230	CDMA2000 1X	800	MIDP 1.0, CLDC 1.0	120x160/8 bits	Yes
Samsung	SCH-X250	CDMA2000 1X	800	MIDP 1.0, CLDC 1.0	120x160/8 bits	Yes
Samsung	SCH-X350	CDMA2000 1X	800	MIDP 1.0, CLDC 1.0	128x128/2 bits	Yes
Samsung	SGH-S100	GSM/GPRS	900, 1800, 1900	MIDP 1.0, CLDC 1.0	128x160/16 bits	Yes
Samsung	SPH-A500	AMPS, CDMA	800, 1900	MIDP 1.0, CLDC 1.0	128x128/12 bits	Yes
Samsung	SPH-N400	AMPS, CDMA	800, 1900	MIDP 1.0, CLDC 1.0	128x96/16 bits	Yes
Samsung	SPH-X4209	CDMA2000 1X	800	MIDP 1.0, CLDC 1.0	128x160	Yes
Sanyo	A3011SA	CDMA2000 1X	800	MIDP 1.0, CLDC 1.0	132x176/16 bits	Not yet
Sanyo	SCP-4900	AMPS, CDMA2000 1X	800, 1900	MIDP 1.0, CLDC 1.0	120x96/12 bits	Yes

Manufacturer	Model	Wireless Technology	Frequency (MHz)	Software	Screen	Available
Sanyo	SCP-5300	AMPS, CDMA2000 1X	800, 1900	MIDP 1.0, CLDC 1.0	128x132/16 bits	Yes
Sharp	J-SH07	PDC	1500	MIDP 1.0, CLDC 1.0	120x160/16 bits	Yes
Sharp	J-SH08	PDC	1500	MIDP 1.0, CLDC 1.0	122x162	Yes
Sharp	J-SH51	PDC	1500	MIDP 1.0, CLDC 1.0	122x162	Yes
Siemens	C55	GSM/GPRS	900, 1800			Not yet
Siemens	M46	GSM/GPRS, TDMA	800, 900, 1900			Not yet
Siemens	M50	GSM/GPRS	900, 1800	MIDP 1.0, CLDC 1.0	101x64/1 bit	Yes
Siemens	SL42	GSM/GPRS	900, 1800	MIDP 1.0, CLDC 1.0	101x80/1 bit	Yes
Siemens	SL45i/6688i	GSM	900, 1800	MIDP 1.0, CLDC 1.0	101x80/1 bit	Yes
Sony Ericsson	A3014S	CDMA2000 1X	800	MIDP 1.0, CLDC 1.0	120x120/16 bits	Yes
Sony Ericsson	P800	GSM/GPRS	900, 1800, 1900	MIDP 1.0, CLDC 1.0, PersonalJava 1.1.1	208x320/12 bits	Not yet
Sony Ericsson	SO503i	PDC	800	CLDC 1.0	128x128/16 bits	Yes
Sony Ericsson	SO503iS	PDC	800	CLDC 1.0	128x128/16 bits	Yes
Sony Ericsson	SO504i	PDC	800	CLDC 1.0	128x128/16 bits	Yes

Manufacturer	Model	Wireless Technology	Frequency (MHz)	Software	Screen	Available
Toshiba	A3013T	CDMA2000 1X	800	MIDP 1.0, CLDC 1.0	144x176/16 bits	Not yet
Toshiba	C5001T	CDMA	800	MIDP 1.0, CLDC 1.0	144x176/12 bits	Yes
Toshiba	J-T06	PDC	1500	MIDP 1.0, CLDC 1.0	16 bits	Yes

APPENDIX C: DEVELOPMENT TOOLS

Tool	Producer	Compatibility
J2ME Wireless Toolkit	Sun Microsystems	CLDC 1.0, MIDP 1.0
J2ME MIDP	Sun Microsystems	CLDC 1.0, MIDP 1.0
Yospace Motorola Accompli 008 Emulator	Yospace	CLDC 1.0, MIDP 1.0
Nokia J2ME Tools	Nokia	CLDC 1.0, MIDP 1.0
Motorola iDEN Tools	Motorola	CLDC 1.0, MIDP 1.0
Siemens SL45i / 6688i SDK	Siemens	CLDC 1.0, MIDP 1.0
WHITEboard Wireless Java SDK	Zucotto Wireless	CLDC 1.0, MIDP 1.0
Jbed Micro Edition	esmertec	CLDC 1.0, MIDP 1.0
MicroEmulator	Bartek Teodorczyk	CLDC 1.0, MIDP 1.0
J2ME CLDC	Sun Microsystems	CLDC 1.0
iJADE Lite	Zentek Technology	CLDC 1.0 iAppli specs
LG TeleCom ez-Java emulator	LG TeleCom online support (in Korean)	CLDC
iEmulator	Taisuke Fukuno	CLDC 1.0 iAppli specs
jBuilder MobileSet	Borland	CLDC 1.0, MIDP 1.0
Forte	Sun Microsystems	
Reqwireless	Reqwireless	
IBM Visual Age	IBM	MIDP, CLDC
J2ME Wireless Toolkit	Sun Microsystems	CLDC 1.0, MIDP 1.0