

Bachelor's thesis  
Degree program  
Information technology  
2010

Maamar zeddek

# DESIGNING SHIFT CALENDAR FOR MOBILE PHONE



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT  
TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme | Information Technology

Completion of the thesis | Total number of pages 35

Instructor Tiina Ferm

Maamar Zeddek

## DESIGNING SHIFT CALENDAR FOR MOBILE PHONE

At Nokia production site employee's work is divided into three shifts, each of 8 hours, and each employee works just one of those shifts. There is a need for a shift calendar for, since scheduling the staff for specific shifts sets an expectation for both the company and employee. Scheduling also allows the employee to plan his/her free time and budget.

This thesis shows how to design a Java application by using NetBeans IDE targeting cell phones, creating a shift calendar application leads to use J2ME. Java Platform, Micro Edition, is a Java platform designed for embedded systems which offers two methods, the file connection API JSR75, and the chosen option the RMS which is an a Record aggregation formed into columns and rows. The development environments (IDE) used is the NetBeans version 6.9 used by many developer as it provides a built-in support for developing in Java, C, C++, XML, and HTML. Two different platforms are available for Nokia devices ;series 40 which occupies the biggest number of phones on the market and the series 60 or the "symbians" used as platform for smart devices, S60 is currently amongst the most-used Smartphone platforms in the world. It was created by Nokia, who made the platform an open API. After determining the shift calendar application most important requirements and creating the architecture, this includes: making the main classes of this project, designing the flow of the application by using the visual mobile designer and append the RecordStore procedure to the project main and finally compiling and running the project.

This study explains how to design a MIDlet capable to read in RecordStore note available in phone's memory by using the development kit NetBeans IDE 6.9. The user can define his/her own shift by name, ok command and the definitions will be persistent in the phone and can be modified or deleted. The application then browses the current week shift schedule for the all shifts. The results are further discussed and developed at the production

KEYWORDS: Java2ME, Persistent Storage RMS, Shift calendar, NetBeans

# CONTENT

## LIST OF ABBREVIATIONS (OR) SYMBOLS

<b>1 INTRODUCTION</b>	<b>6</b>
<b>2 TOOL AND METHODS</b>	<b>8</b>
2.1 NetBeans IDE	8
2.2 Development Platform	9
2.2.1 Java ME	9
2.3 File connection API JSR 75	12
2.4 RecordStore and Packages	13
<b>3 NOKIA DEVICE PLATFORMS</b>	<b>17</b>
3.1 Symbian overview	18
3.2 Series 40	19
3.3 Symbian features	22
<b>4 PROCESS OF DEVELOPMENT</b>	<b>25</b>
4.1 Installing the application on mobile phone	32
<b>5 CONCLUSION</b>	<b>34</b>
<b>SOURCE MATERIAL</b>	<b>35</b>

## **PICTURES**

Picture 1. NetBeans Start Page	9
Picture 2. Use Case Diagram	26
Picture 3. The flow after the basic classes are created	28
Picture 4. The full Application flow	30

## **FIGURES**

Figure 1. The J2ME Stack	10
Figure 2. The MIDlet Lifecycle	12
Figure 3. Nokia devices used S40 as platform	19
Figure 4. Nokia devices using Symbian platform	22
Figure 5. Keypad for a device using Symbian	23
Figure 6. The Class Shift.java	27
Figure 7. Manage Workdays Class	27
Figure 8. The Class Diagram	31

## **TABLES**

Table 1. MIDlet communication with AMS	11
Table 2. Linking the display items procedure	29

## LIST OF ABBREVIATIONS (OR) SYMBOLS

AMS	Application Management Software
API	Application Programming Interface
CDC	Connected Device Configuration
CLDC	Connected Limited Device Configuration
IDE	Integrated Development Environment
JAR	Java Archive
J2ME	Java 2 Platform Micro Edition Designed for the small memory devices.
JSR	Java Specification Request
JVM	Java Virtual Machine
MIDlet	An application used in J2ME to use the MIDP of a CLDC.
MIDP	Mobile Information Device Profile
SDK	Software Development Kit
VMD	Visual Mobile Designer

# 1 INTRODUCTION

This work aimed to incorporate the technology aspect with the working life of the Salo Nokia plant production employees in order to make the access to shift calendar easier.

In this pacing world we become critically dependent on this cell phone devices, but until now the majority of the employees at Salo Nokia plant still using the traditional printed paper sheet as shift calendar, there're different shift calendar in use for three, four or five shifts, depend on model in use, while this small gadgets can contain all the content of these traditional printed papers.

This work will reveal the tasks and steps passed to achieve the target by discovering all the tools needed and methods to pass through.

The tools necessary to develop mobile applications are also different, thus, different constraints must be addressed while developing mobile applications such as the development platform and tools.

Java Micro Edition has been chosen as the development platform. Java2 Micro Edition is a java platform designed by sun Microsystems for mobile devices and embedded systems and is supported by all the mobile phone manufacturers

In order to develop this MIDlet, it is necessary to use the Sun Java Wireless Toolkit NetBeans IDE, The MIDP (Mobile Information Device Profile) profile (API) defines elements such as: High level user interface elements, Application lifecycle management, Local data storage and Connectivity, All applications for the MID Profile must be derived from a special class, MIDlet. The MIDlet class manages the life cycle of the application. It is located in the package javax.microedition.midlet. MIDlets can be compared to J2SE applets, except that their state is more independent from the display state. The MIDlet exist in four states: loaded, active, paused, and destroyed. The RMS has been chosen as method to develop this MIDlet's. The RMS (Record Management System) defines the

following conceptual operations on a record store: add a record, delete a record, change a record, retrieve a record and enumerate all records.

The application will display on the phone screen shift and working day for chosen date and also six days within same week.

Customer installs application from PC with USB-cable and copies wanted shift model (RecordStore note) to mobile phone.

## 2 TOOL AND METHODS

The J2ME Wireless Toolkit used to achieve the development of this application is the NetBeans IDE version 6.9 right now it's the latest version which is owned by the sun Microsystems.

### 2.1 NetBeans IDE

NetBeans provides a complete development environment to write and test MIDP applications. The download includes tools, documentation, an emulation environment, examples, and a module to integrate with Forte for Java. This Toolkit for CLDC has three components is the most important

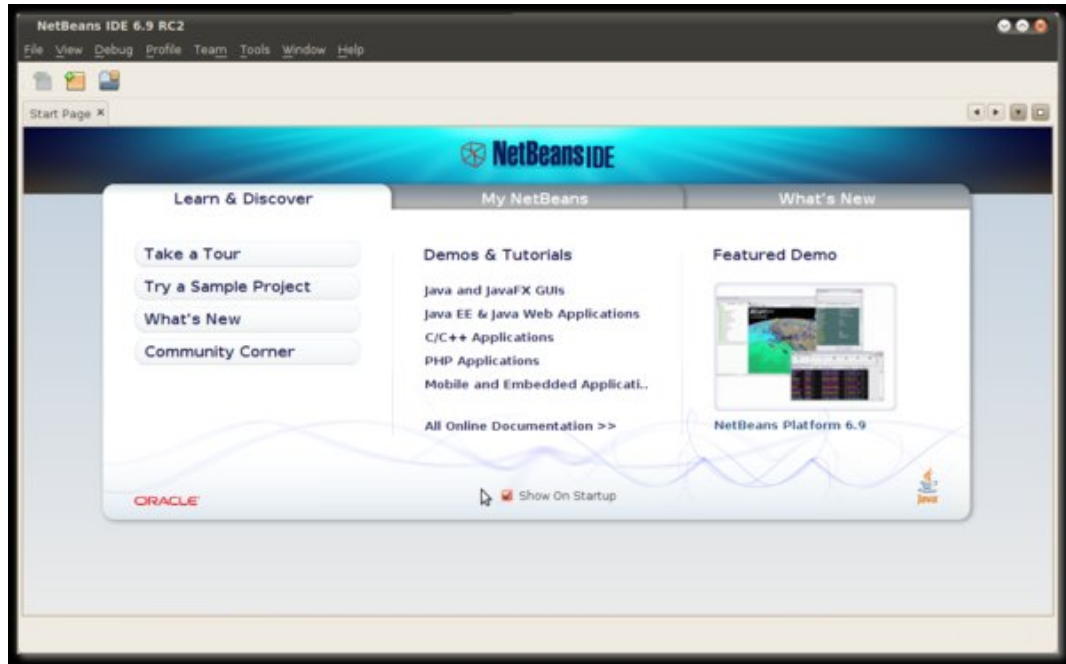
- The user interface automates many of the tasks involved in creating MIDP application.
- The emulator is a simulated mobile phone.
- A collection of utilities provides other useful functionality, including a text messaging console and cryptographic utilities.

It is free, open source Integrated Development Environment for software developers. You can get all the tools you need to create professional desktop, enterprise, web, and mobile applications with the Java platform, as well as C/C++, PHP, JavaScript, Groovy, and Ruby.

NetBeans IDE is available for free downloaded at <http://www.netbeans.org> [1]

The NetBeans Platform is a broad Swing-based framework on which we can base large desktop applications. The IDE itself is based on the NetBeans Platform. The Platform contains APIs that simplify the handling of windows, actions, files, and many other things typical in applications, each distinct feature in a NetBeans Platform application can be provided by a distinct NetBeans module, which is comparable to a plug-in. NetBeans module is a group of Java classes that provides an application with a specific feature; you can also create new modules for NetBeans IDE itself. For example, you can write modules that make your favorite cutting-edge technologies available to users of NetBeans IDE. Alternatively, you might create a module to provide an additional editor feature. [1]





Picture 1. NetBeans IDE start page

## 2.2. Development platform

Java Micro Edition has been chosen as the development platform. Java Me is a java platform designed for mobile devices and embedded systems and is supported by the main mobile telephone manufacturers

### 2.2.1 Java ME

Java Micro Edition targeted at limited size devices and memory and standardizes the use of the java technology in the wireless devices such as cell phones.

Due to the diversity of the devices the architecture of the java2 micro edition must be modular and scalable, that is why j2me provides a range of virtual machines, here we're going to take a full and quick look at J2ME.

We can split the J2ME into three portions, as shown in Figure 1, configuration, a profile, and optional packages. The configuration it holds the java virtual

machine and some class libraries. A profile builds on top of these base class libraries by giving a useful lot of APIs; and optional packages, an optional set of APIs that is in use or not in use when creating the application. Optional packages are usually not packaged by the constructor of the device, and must be packaged and distributed with the application. The profile and configuration are prepared and embedded in the device by the constructor of the device. [2]

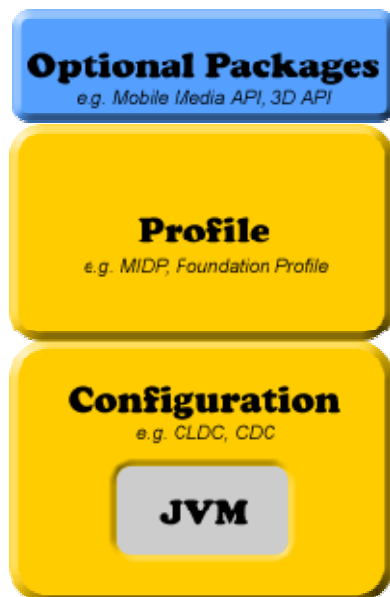


Figure1. The J2ME stack [2]

The most popular profile and configuration that Sun provides are the Mobile Information Device Profile (MIDP) and Connected Limited Device Configuration (CLDC), respectively. As the name suggests, CLDC meant for device that has limited configuration; for device that has a range from 128KB to 512KB of available memory for J2ME. Consequently, the JVM that it provides is very limited and supports only a small number of traditional Java classes. Limited JVM is called the KVM. (Referenced)[2]

The MIDP complements the CLDC configuration can reduce the memory space needed and the power consumption for these small devices. Well it gives the main API that is used to develop applications for these devices. For instance, it offers the `javax.microedition.lcdui` package that let us to create the graphical user interface component that can be shown on a device running the MIDP on peak of CLDC configuration. (Referenced)[2]

The MIDlet Lifecycle:

All mobile devices, including the emulators also, react with the MIDlet by utilizing the software that they possess, that is named as Application Management Software. [2]

The Application Management System can perform: starting, pausing, resuming (the active state of the MIDlet), and destroying the MIDlet. Also the Application Management System can set up or delete the MIDlet, to make this easy to understand we can say that the MIDlet can be: Paused, Resumed (active) or Destroyed and that can be done by the MIDlet class method.

Method	Description
<code>notifyPaused()</code>	The MIDlet is ready to be paused
<code>notifyDestroyed()</code>	The MIDlet is ready to be destroyed
<code>resumeRequest()</code>	The MIDlet is ready to run.

Table 1. MIDlet communication with AMS

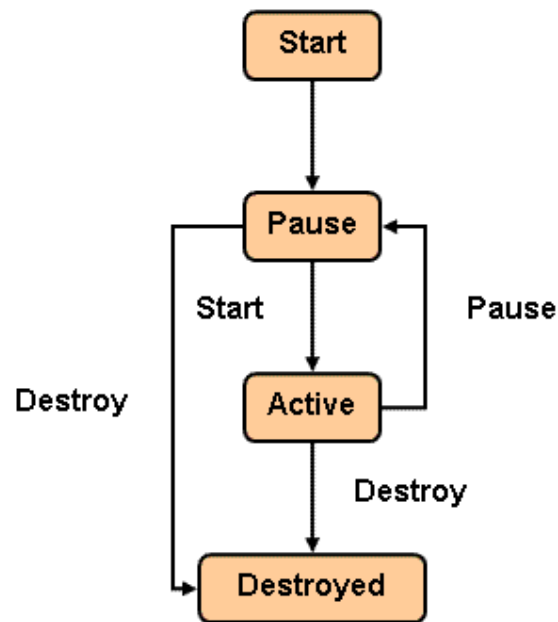


Figure2: The MIDlet lifecycle [3]

### 2.3. FILE CONNECTION API JSR-75

JSR 75 can furnish helpful APIs that J2ME developers are in need for, to optimize the use of the features found on PDA in the J2ME environment, in the form of two optional packages which can heighten software stacks based on CLDC: [4]

1. The File Connection APIs offer for J2ME device an access to file systems inside on mobile devices, mainly access to removable storage like a memory sticks or memory cards.
2. The PIM (personnel information management) APIs provide the mobile device access to personal information management information to mobile phone, like agenda, address books, calendars, reminder and so on.(Referenced)[4]

The File connection (FC) and the personnel information manager (PIM) packages are totally independent and different from each other. [4]

As a consequence the devices able to support the CLDC1.0 can automatically support the File Connection JSR 75, also we can deploy the File Connection on the top of any connected device configuration or connected limited device configuration. [4]

#### 2.4. RecordStore and packages

RecordStore called also persistent storage is a sort of storing data, that can gives ability to retain the stored (Records), whenever, the device is powered or not. In some cases of MIDlets applications the demand of objects existence even after the end of the application that created those objects. Without the help of Record Management System (RMS), the objects are destroyed after the application quits. But thanks to the persistent storage that offer the possibility to save the objects, and gives them much longer lifetime than the application that create them, so these objects and their states are available to be retrieved whenever needed.(Referenced) [5]

“The Record Management System services provided in the Java Standard Edition platform, such as Object in serial form APIs, does not fit for small devices with limited memory. This is due to the storage demand change importantly from one to another resource limited device. [5]

Introduction to the RMS:

The MIDP provides a way for MIDlets function to persistently store data and retrieve it later. This way-function is a simple record-oriented database called the Record Management System (RMS). A MIDP database (or a record store) consists of a collection of records that remain persistent after the MIDlet exits. When you invoke the MIDlet again, it can retrieve data from the persistent record store. [5]

### Introducing the Record Store:

Record stores known also as binary files are platform-dependent because they are created in platform-dependent locations. MIDlets within a single application (a MIDlet suite) can create multiple record stores (database files) with different names. The RMS APIs provide the following functionality:

Allow MIDlets to manipulate (add and remove) records within a record store.

Allow MIDlets in the same application to share records (access one another's record store directly). [5]

### Names of the Record Store:

The names for Record store are very sensible, and the long must be in maximum not exceeding than thirty two characters. In the same application it is not possible for the MIDlet to create two different record stores holding the same name, on the other hand, is allowed for the MIDlet to create a record store with the same name in different application. Creating a new record store is stored in a "NOJAM" directory. For this project, when I'm using the NetBeans the new record created are stored under the directory C:\J2MEWTK. Well this project name is Shift application and the record store is myshift, the record store is created C:\J2MEWTK\NOJAM category, with the name myshift.db. (Referenced) [5]

### The Packages of RMS

The package of RMS exists in four different interfaces as follow: five exceptions classes and one class.

For the interface:

RecordListener: Receiving the records that have been moved out, shifted, or renewed within record store.

RecordComparator:in order to compare between two records.

RecordEnumeration: to define the tow-way records enumeration

RecordFilter: to specify filter to fetch records.

The Classes of RecorStroe:

RecordStore: stand for a record store (RMS).

The RMS Exceptions are as listed below:

RecordStoreNotFoundException:if we get this exception that means the record Store cannot be found.

RecordStoreException: to inform that is about a general exception

InvalidRecordIDException: to inform that the Record ID is invalid.

RecordStoreFullException: to inform that the record store file system is full. [5]

RecordStoreNotOpenException: this exception means that for this operation the Record store is closed.

How to open a Record Store:

Opening the record store:

By using the method "openRecordStore ()". We can open the RecordStore. [5]

The complete application must create from RecordStore and read from RecordStore then delete record from RecordStore.



### 3 NOKIA DEVICE PLATFORMS

Nokia offers two device platforms supporting Java™ ME: Series 40 and Symbian (formerly known as "S60"). Different versions of the platforms are identified as editions and feature packs. Editions represent significant updates in functionality, while feature packs are incremental improvements within an edition. Each combination of platform, edition, and feature pack has its own SDK.

For developers using the java as programming language:

The user interface style and the touch both are determined by the established platform in the series 40 edition 6. This can signify Java applications designed for a "QVGA" screen surface to be run on a Type and Touch device without need to make any major changes, in very few cases some changes are required; but they are very limited if I don't say it's negligible.

This Series 40 Edition 6, offer new perspectives to intensify the user experiences for Java applications with touch. To get the optimum benefit using the feature of touch, we have 2 application programming interfaces which can give the ability to detect the motion happened during the screen touching, also computing the additive vitality and the stagnant.[6]

By using these application programming interfaces we can get the optimum touch of the application.

More information on these APIs is available, at Nokia forum / the Java Developer's Library/Touch interaction in Series 40. [6]

### 3.1 The Symbian

This platform developed by Nokia is used mainly for the smart phones. This platform is executed and found in various scopes of different devices, can give the application developers logical and easy features.

The Symbian is capable of being changed (flexible), this characteristic offer the opportunity for classic cell phones to use the standard keypad and screen QVGA, until the most sophisticated cell phones known as “smart devices” it provides touch with high definition for the screen, full keyboard, tactual comment and sensual element device fitted in innovative form. Developing application such as music or games equally at home delivering advanced enterprise apps, games, or music, the Symbian platform offers for java ME developers a unique possibility in the small devices word. [6]

Developers can use to develop applications the recommended approach, using the (Nokia Qt SDK) called Qt, the symbian C++ by using native Symbian and Nokia S60 platform on OS application programming interfaces.

Collection of an open C and C++ application programming interfaces, in the language of Java the use of MIDP 2.1 with large scope of java specific requests, the web runtime by using the technology standards, and the Flash Lite by adobe.

The java application developers have a large support for video, audio and the image. The adobe Flash Lite possibly used for animated content is an asset too,

The symbian support web browsing technology, for desktops so graphic designers use this characteristic to create new topic or subjects for symbian devices which can entirely change the sound and look of a device. [6]

When the applications are created, the use of “OMA DRM” can offer for developers’ intellectual property protection.

The s60 platform or symbian offer for developers to create valuable applications with remunerative financial income. This symbian platform is sophisticated with all needed technology and supporting tools, there is plenty of examples documentations, which can offer for developers, chance to win time and money in the market battle.

### 3.2 Series 40 platform

This S 40 platform is used widely among the devices on the market actually is the biggest due to the cost and size. A very big scope of devices merged with the platform of Series 40 is accessible for almost all the standards of wireless.

In order to seek more information about Series 40 devices, you can see the Forum Nokia Device Specifications.

Figure3 bellow shows some Nokia devices using S 40 as platform.



Figure 3. Nokia devices used the S 40 as platform. [6]

The display on S 40:

The following displays are supported by the platform S 40:

- 128 x 128
- 128 x 160
- 208 x 208 for only S40 2nd Edition and the 3rd Edition
- 240 x 320 for S 40 3rd Edition
- 320 x 480 for S 40 6th Edition

The color depth is 16-bit in minimum, and the latest devices can be able to support, color even up to 24-bit in depths. [6]

The S40 platform Input:

These input keys are for series 40 without touch screen:

- 2 soft keys (SK1 and SK2)
- 5 way direction for keypad arrows ,selection
- Call handling keys for “ Send” and “End”
- The digits keys from 0 to 9 and the (\*) ,(#)
- The power key ,older Series 40 devices use the “end” key to replace the “power” key

(Referenced)[6]

The S 40 supports too some other keys from 3<sup>rd</sup> or 3G and these keys are as

Listed bellow:

- key for camera
- press to talk key
- keys of volume

Referenced [6]

The latest version of this platform is able to react with the touch screen technologies. [6]

These input keys are included in devices occupied by the S40:

- keys of call handling with the keys Send , End
- keypad with digits from 0 to 9 and the symbols \* , # (referenced) [7]

The virtual keys are replacing the physical soft keys, the Touch-enabled s

Devices can support these keys also:

- Camera key
- The key guard switch
- The Power Key
- The key of volume (referenced ) [7]

To get more details about these and more can be found on Nokia library.

The memory of devices using S40 as a platform:

Devices occupied with Platform of S40 manage their memory by means of using java static cumulated unchanging memory. Thus, for most S40 devices, the size is 2MB maximum for MIDlets that are active, and MIDlet which is running on background is 2.5MB.

Devices with the S40 platform can support a java files with memory size exceeding than two MB. [7]

### 3.3 Symbian features:

The S60 platform known as symbian nowadays is a very advanced and can provide as software platform plenty of useful features, and enhancing the capability of applications. Symbian devices give the opportunity to focus on the consumer device cleavage, like entertainment, and games, music and so on.

Figure4 shows some of Nokia devices model using the Symbian as a platform



Figure 4. Nokia devices using the Symbian as platform [6]

The display of symbian platform devices:

The platform of symbian does support these sizes of pixels per square inch

Within display:

- 176 x 208
- 240 x 320
- 352 x 416
- 360 x 640 (referenced)[6]

Despite the presence of few differences between the devices, most of situation faced we can trust on the size of pixels per square inch for the display to be:

If the version of the platform is older than S60 3rd edition the resolution is: 176 x 208 / 352 x 416.

If the platform version is more new than that, the resolution is: 240 x 320.

The color depth ranges supported is 12bit colors in minimum, of the oldest devices until the actual 24bit colors.

We can find some exception cases that device can offer resolution of 208 x 208 pixel. (Referenced) [6]

The user interface of the symbian can be in the interval of two modes, the portrait and landscape. That means in the display, the height and width can be shifted. The mode of landscape, in some component of the user interfaces like the soft keys possibly shift the places. The MIDlets developed for the purpose specified resolution may change to different resolutions which the device can support. [6]

The Input of the symbian devices platform:

The figure bellow reveals all the main keys required in a device using a Symbian as a platform without screen touch.



Figure 5. Keypad for a device using a symbian [6]

For a device not using a screen touch technology and enabled with the symbian platform the required keys are as listed bellow:

- Two soft keys
- Five different position navigation keypad

In the latest edition of the symbian we can find on a device nine different direction of navigation on the keypad.

- “call” and “end” keys
- Applications key
- keypad with digits from 0 to 9 and symbols \* and #
- the “clear” key [6]

In some few devices the “edit” key is missing, so it is replaced by the “Press” key in the majority. (Referenced) [6]



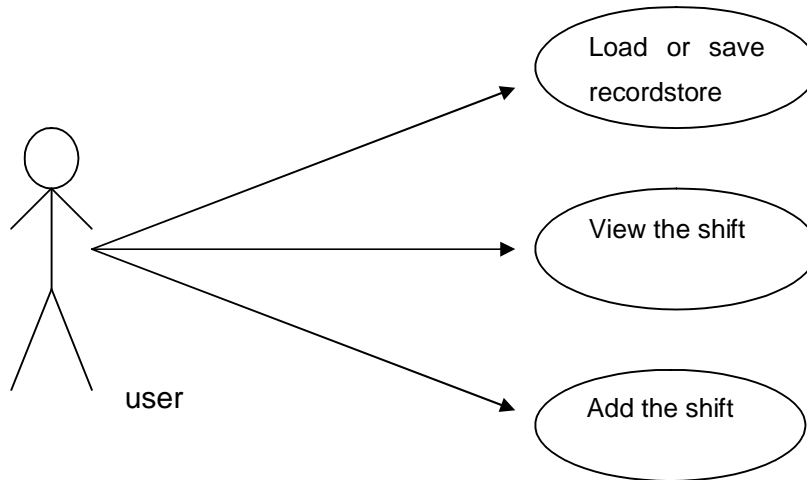
## 4 PROCESS OF DEVELOPMENT

Designing a shift calendar application for a mobile device require taking in account the limited memory and the battery energy saving, of the device that my application is targeting , the best way is using the RMS record management system offer the chance to store data and retrieve it whenever needed .the RecordStore is a sort of small database ,the J2ME offer an other alternative also to perform this application ,which is the data file reading “file connection JSR75”API ,this option is better to use when reading data from an external memory such as memory stick.

The development of this application can be divided into the following parts:

- 1 purpose
- 2 SDK (NetBeans)
- 3 define the new classes
- 4 using the VMD to create the flow
- 5 making the connection of parts in the flow
- 6 combine RMS capability with the classes

Before starting with the NetBeans lets define the user case diagram and the class diagram (this section we'll talk about it same time with the part of defining new classes), the user case diagram will be as follow (see picture2).



Picture 2. User case diagram

- The Purpose: this project is intended to design a J2ME application which can be able to deployed on mobile device, read, write and delete from the RecordStore. it must be able tostore the shifts and shift time.
- SDK (software development kit): Netbeans IDE 6.9.1and the mobility pack developed and offered free of use by the sun Microsystems which is the owner; it helps J2ME developers and other programming languages for writing, testing and debugging.

Installing the IDE:

The Sun Java Wireless Toolkit NetBeans IDE has been used as a tool to design this application.

Download and install the Netbeans IDE version 6.9.1 from <http://netbeans.org>

- Then creating the main classes we do it as follow

In NetBeans after creating a new project a MIDP application will be included in the source package, we open this, and we rename the class as shiftjava.

By clicking in the dialog box of "HELLO MIDlet, the NetBeans at same time is creating a helloMIDlet.java class, in the source package. Then we add some changes in the class: shift.java, the class is as shown in figure6.

shift java
Shift symbol: string = null
Shift time : int = 0
Write to disk(): void
Read from disk(record ID:int) : int

Figure 6. The class shift.java

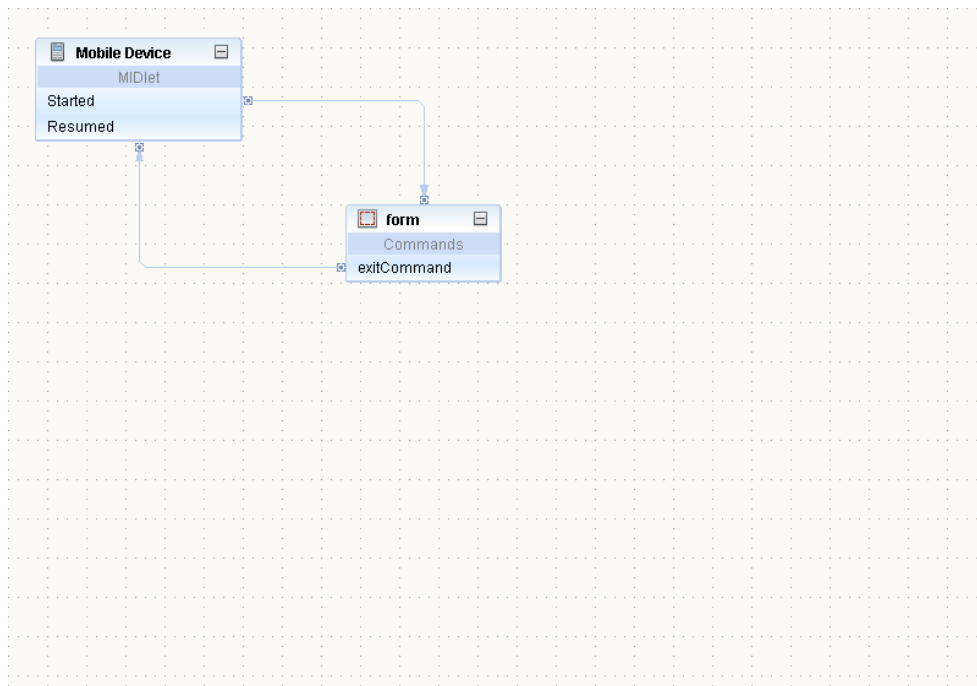
Then we need other class that can manage the working days, let's call it as manageworkdays.java. It's done by extending the source package of the project, click on option key, new, and then a MIDlet. The class is (see figure7).

Manage workdays
Manageworkdays[ ]: shift java
Cout : int
Added : int
Write to disk ():void
Read from disk ():void
Add workdays(): void

Figure 7. Manage Workdays Class

- After the two needed classes are created we go to create the flow by using the visual mobile designer.

The VMD visual mobile designer is a graphical interface within NetBeans Mobility Pack that gives you the ability to design mobile applications, so we will do it like this: one click the HelloMIDlet.java tab, At the top of the code, one click on the flow, the displayable palette is opened, and we choose the right items according to the application, see picture 3.



Picture 3. Flow after the basic classes are created

We change the names as we desire to fit in our application components that including the form item and the list item too, this can be done by right clicking on the list and selecting properties option. This list will be used to display the main menu. The procedure to do so, see table 2, as follow.

In menu list we add the exit command. It is done by: click on option key, add, and we choose the exit

Two List items to the menu are needed, we can add it by drag and drop we change the name as shift and View shift.

Another tow List items should be added and renamed as viewshift. and addshift

Back command must be added to it named as backCommandView. By click on the option key and choose the command.

One click on Screen view, then option key click and we choose two text-fields. "shiftgroup" and "shifttime". will be typed

By the drop and drag we add a new Formfrom the displayable:

We rename it as addshift

We give the title as addshift.

Then we need for a cancel command and ok command to the form simply by clicking on the option key, add, and we choose the command desired.

From the displayable we drag a splash screen item and drop it into the flow application.

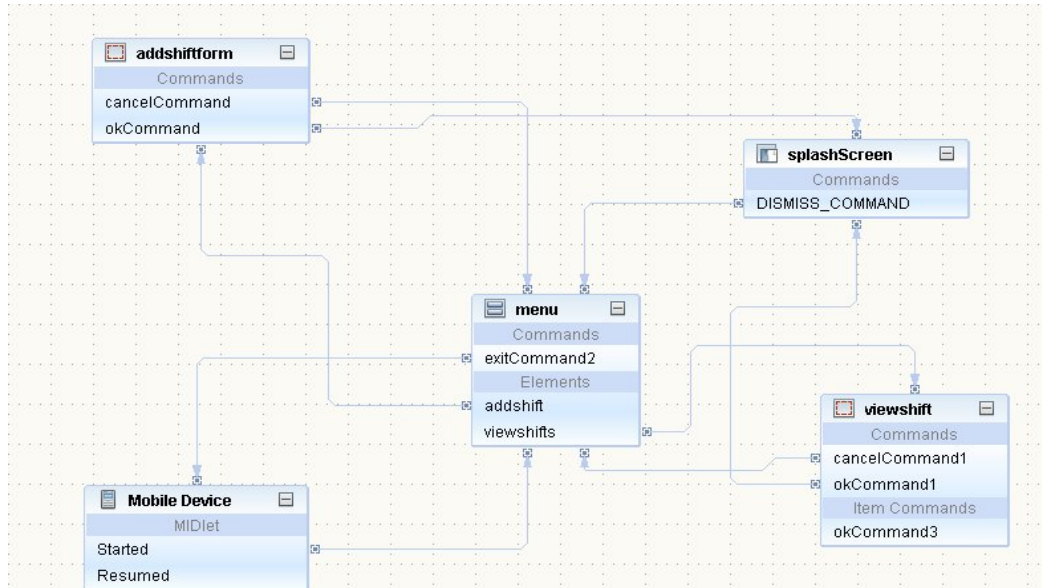
Always by using the mouse option key we can add the desired world to be displayed on this splash screen, here I put "shift added"

Table2. Linking the display items procedure

We will connect the five different frame boxes present in our flow diagram as we

Wish that our application will work, the flow diagram of the whole application will

be as shown in picture 4.



Picture 4. The full application flow

Now at this stage we are able to combine the RecorStore capabilities with the different classes: the shift java, and manage workdays.java

We open the project “shift application”, we got the “helloMIDlet.java” and

Other classes “shiftjava.java” and “manageworkdays.java” By one click on

Source window the source code will be opened, in generate the method we

Can add this code:

```
myworkdays.addworkdays(shiftSymbol.getString(),shiftTime.
getString()); shiftTime.setString(""); shiftSymbol.setString("");
myworkdays.writeToDisk();
```

The class diagram for the application looks like the following diagram

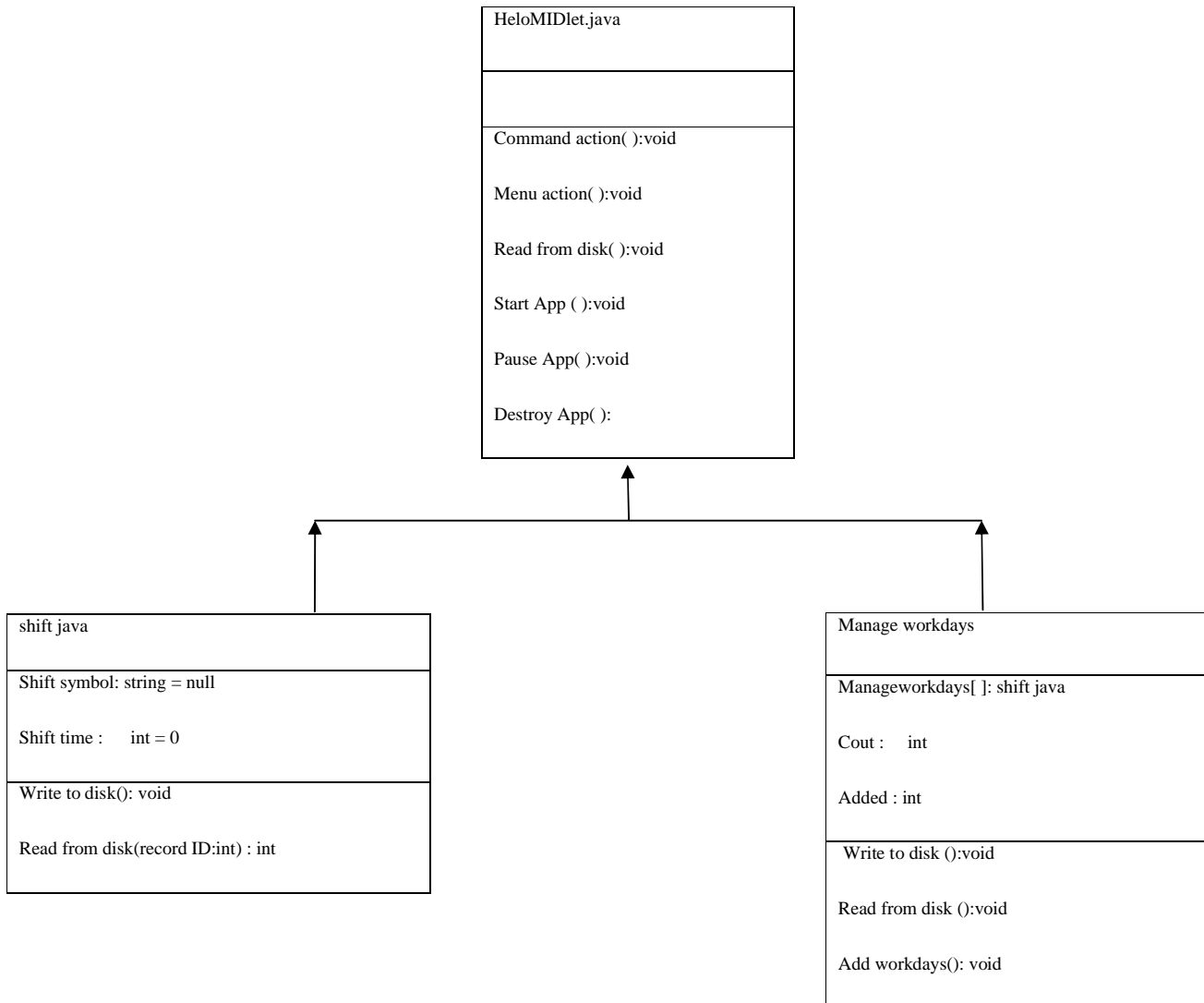


Figure 8: The class diagram

The final step is to compile and run the project, then the emulator will display on Screen menu the add shift and view shift, then the application is ready to be deployed to the mobile phone.

#### 4.1 . Installing the application on the Mobile phone

J2ME application can be deployed to the mobile phones by several ways:

By the use of USB cable is possible deploy the application from the computer to the mobile and is the common mode to install the mobile applications, and by using Bluetooth.

Unlike the symbian the devices using S40 as platform it's easy to install the mobile application. In the symbian case or devices with S60 platform deploying such application can be done manually, I mean that can be done by following the instruction given by the device.

By the use of PC suite (is software used to perform an interface between PC and mobile device produced by Nokia), can make the application installing on devices using symbian, much more easily.

We can also save the mobile application in the JAD file or the java archive file on a server, then by internet connection with the mobile device is possible to download the MIDlet and install it, this way is known also as "OTA".

There are four different ways to install the application into a mobile phone:

1. Installing by USB cable:

This mode of MIDlet installing is the most used, and it's based on the simple connection of both devices (mobile and PC) by means of USB cable, since almost all devices has a USB port.

2. Using the WAP (wireless application protocol):

We can send WAP Push message to a specified mobile device (this message include a URL link), by connecting the device to the link, thus, we can download the application, which was previously available on the link.



### 3. Installing MIDlets by Bluetooth and Infrared:

Developers are using this mode to carry over the mobile application from the PC to mobile devices, by activating the Bluetooth for both devices and then deploy the MIDlet.

### 4. Over the Air:

The abbreviation is “OTA” and it means that we can send the MIDlet to a server and by the presence of available network permit for the mobile device to connect to the internet and download the MIDlet from a server by means of wireless technology.

## 6 CONCLUSION

The purpose of this thesis was to create simple application for mobile phone to help the shift rotation employee for scheduling, managing, and planning their time using mobile phone.

The application designed for mobile phone is pretty much different from desktop application the interface with the user is so limited because of the small display and keys in the keypad. In order to build and test MIDlets, it is necessary to use the Sun Java Wireless Toolkit; The NetBeans IDE used here to develop such application, NetBeans is oriented on wide audience of developers from beginners up to experts. Each of them can find useful set of the development tools that are embedded in the IDE or can be integrated with. This IDE is the modular system and it can be configured according to user needs.

The development platform is Java2ME is a java platform designed for mobile devices and embedded systems, and the method used is Record Management Store RMS. The RMS defines the following conceptual operations on record store: add a record, delete a record, change a record, Look up a record, and enumerate all records.

This thesis has described all the steps throughout the work process to develop shift calendar application developing this software can be done by means of two options: The file connection API jsr-75 or RecordStore (RMS)

Because the application must read from the RecordStore note that is already saved into phone memory, the RecordStore was the easiest way.

Create an application with J2ME to provide facility for storing shifts, read and display, was challenging, due to the limitations of MIDP and the problems of developing for the mobile device environment but good organization, along with careful design and documentation, will help to create the shift calendar. The emulator screen (SDK 3.0) view shift and add shift, as options to select for the current week, so the MIDlet is ready to be deployed.

## SOURCE MATERIAL

[1] 2010, Oracle Corporation and/or its affiliates available: <http://www.netbeans.org> [December 2010].

[2] Vikram Goya, 2006 Pro Java ME MMAPi mobile media API for java [2010-01-26] also available at: <http://today.java.net/pub/a/today/2005/02/09/j2me1.html>

[3] Figure available on December 2010 at <http://j2mesalsa.com/elearning/lifecycle.html>

[4] QusayMahmoud, December 2004, Oracle Corporation, checked [December 2010]. Available at: <http://developers.sun.com/mobility/apis/articles/fileconnection/>

[5] QusayMahmoud, December, 2000.checked [December, 2010] available:

<http://developers.sun.com/mobility/midp/articles/persist/>

[6] The Nokia forum library checked [December, 2010] available at:

[http://library.forum.nokia.com/index.jsp?topic=/Java\\_Developers\\_Library/GUID-3B0C0034-9913-40AF-B627-27564A5448D6.html](http://library.forum.nokia.com/index.jsp?topic=/Java_Developers_Library/GUID-3B0C0034-9913-40AF-B627-27564A5448D6.html)