M-Learning Using Mobile Learning Engine

by

Noorfaizah Binti Abas

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

DECEMBER 2006

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

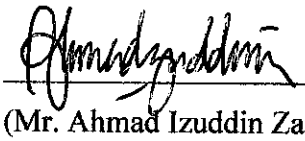
M-Learning Using Mobile Learning Engine

by

Noorfaizah Binti Abas

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

Approved by,



(Mr. Ahmad Izuddin Zainal Abidin)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

December 2006

CERTIFICATION OF ORIGINALITY

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

NOORFAIZAH BINTI ABAS

ABSTRACT

The objective of M-Learning is to integrate the technology combined with the education in order to enhance the effectiveness of student's traditional learning process. In order to explore the use of mobile and handheld IT devices as a learning tool, many factors need to be considered such as its constraint and limitations. Therefore, intensive researches need to be done. The main purpose of M-Learning is to create flexible learning environment for students where the implementation of just-in-time learning is applied. Besides, it creates new approach of learning style. The most challenging part in implementing M-Learning is in delivering the content as how users will view the materials in mobile devices instead of the usual large screen desktops. Apart from that, the technology of M-Learning is still new in Malaysia, therefore, a lot of risk and challenges involved in this project. Meanwhile, the target user is students. In this project the methodology used follows four processes which are planning, analysis, design and implementation. Efficiency and flexibly together with ease of use become the essential elements in constructing the final system.

ACKNOWLEDGEMENT

First of all, my gratitude goes to Allah S.W.T for giving me the strength, skills, knowledge and good health in completing my Final Year Project, titled M-Learning.

My warmest gratitude to my supervisor, Mr. Ahmad Izuddin Zainal Abidin for his unwavering support and guidance while completing this project. His generosity, understanding and help have been an inspiration to me. My sincerely appreciation also goes to the FYP Committee for all their patient and understanding in guiding and assisting me in this project.

My special appreciation goes to all Information Communication Technology/Business Information System lecturers for their valuable expertise, guide and support in completing this project. Special thanks to Mr. Anang Hudaya, who gave so generously of his time and effort in guiding and assisting me throughout completing the project.

I also like to thank all the people who have been involved directly or indirectly in completing this project. The contributions you have made throughout the development of this project really valuable to me.

Last but not least, my deepest gratitude and special thanks goes to my family and friends for their support and share of knowledge and feeling that they have shown me.

Again, thank you for your encouragement and continuous support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND OF STUDY

This Final Year Research Project is focused on education as the cornerstone of economic development in any nation. Traditionally, formal education has been offered in classroom where the teacher and student are able to interact with each other face-to-face. However, there is problem when not everybody has the opportunity to go to school. For example, perhaps they are living to far or they are handicap person. These limitations lead the idea of M-Learning as the whole and as another form of learning style.

1.2 PROBLEM STATEMENT

1.2.1 Problem Identification

Current situation;

Students used the computer devices in order to access the E –learning and another website provide by the university. However, due to server and network problem that always happened, they will be the barrier for them from accessing it. The idea or need of the M – Learning arises from the facts that students can easily access to it if there another approach can be used.

The significant of this situation is that students can access the information wherever they are, any time and according to their convenience using mobile devices that are

portable with it. The implement of M-Learning also means that it eliminates the long wait to use computer lab.

1.2.2 Significance of Project

This system is an addition or another solution of the current learning methods where the user will be able to access information of learning materials using mobile device. Through this system, it provides the concept of flexibility where users can have information anywhere, anytime. For the beginning, the system only provide user with one subject and enable them to download lecture materials from that application as far as view the contents.

1.3 OBJECTIVES AND SCOPE OF STUDY

The main objectives are:

1. The ability of students to access easily the learning by using mobile devices.
2. To create the flexible learning environment for students where use the concept of just-in-time learning. (wherever you want to use M-Learning, it is always available)
3. The enhancement of the learning process by using mobile devices

1.3.1 The relevancy of The Project

With the increasing use of mobile phones and handheld devices among teenagers or student, the opportunity of implement M-Learning in academic institution is quite high. However, this project will involve with lots of research on the technology used, the limitations and advantages of M-Learning. The concept of just-in time learning is apply in this project where the students are able to access the information wherever they want using the devices like mobile phone. Although there are lots of constraints

need to be considered such as cost, it's become an interesting issue In the future as people adapt with the changing of technology.

1.3.2 Feasibility of the Project within the scope and Time Frame

Initial feasibility of the project can be evaluated through economic, technical and schedule factors.

- **Economic Feasibility**

This application supports feasibility in terms of economic by increasing effectiveness and productivity of existing education system. By implement this application, learning process will be more efficient and lead to enhancing of learning process.
From hardware perspective, some of relevant software needed. Perhaps the use of real mobile device is needed in completing this project.

- **Technical Feasibility**

This project can be completed with technical guidance from IT lecturer, the author also find the solution from the web based on the paper prepared by student from other country that done research regarding the Mobile Learning application.

- **Schedule Feasibility**

The author has schedule to the time given according to the suggested milestone that can be viewed from the Final Year Project Guidelines. Please refer to Appendix A for the proposed project timeline.

CHAPTER 2
LITERATURE REVIEW AND THEORY

2.1 Introduction to M-Learning

There are few mobile tools currently used to provide learning opportunities to users. It is including mobile phones (the latest one provide with 3G application), PDAs, game consoles and Tablet Pcs and laptops that have wireless capability.

The idea of M-learning arises from the use of E-learning. The concept is the same except the way to get the information using difference devices. Electronic or e-learning incorporates all forms of online instruction using personal computers [4]. However, here we narrow down our focus by allowing the students to access one topic only. The relevant of doing that to emphasize the project and ensure the project run well. Perhaps in future upgrade, full function of E-learning will totally adopt in the M-learning.

Generally, the combination of e-learning and mobile computing is called mobile learning (m-learning) and promises the access to applications that support learning anywhere, anytime [5]. However, most of the proposed examples in the past uses handheld computers or laptops and are mainly supporting adults in the workplace—people who know what they want. Subsequently, m-learning has become an attractive target application area for corporate mobile devices. However, meanwhile hardware is considered as a solved problem; innovative, affordable and usable software remains still the greatest challenge.

From the research have been made, Mobile learning can give big impact in learning environment. Today a learner can be a Net Generation college-age student; adult

learning seeking new skills or career advancement; or a faculty member, staff member, or administrator interested in using technology to improve students' success [4]. It is prove that many organization and individual will benefited of it. Thus, the flexibility and potential of m-Learning can not be underestimated.

For this project, lots of researches carry in and author found another approaches to implement it using the **J2ME – Mobile Learning Engine**. After did further research, the author did some comparison toward the tools and application that will be used in implementing this project.

2.2 Differences between E-Learning and M-Learning

E-Learning

The term E-Learning is analogous to online education. Online education is categorized by the separation of teachers and learners which distinguishes it from face-to-face, the influence of an educational organizational which distinguishes it from self-study and private tutoring, the use of a computer network to present or distribute some educational content, and the provision of two-way communication via a computer network so that students may benefit from communication with each other, teachers, and staff. [7]

Time and place are no longer restrictions as the learning experience can be implementing everywhere one has a computer and access to the internet. The teacher's role become more flexible as they now can tutor only from offices or homes. Student-teacher, student-student interact the E-Learning through e-mail, discussion forum, chat rooms and others.

The model in Figure 2.0 illustrates how E-Learning works. From the screen is where the students receive course content, have access to the web and also other materials.

The student also is able to communicate with the teacher or lecturer or peers from here via email or others.

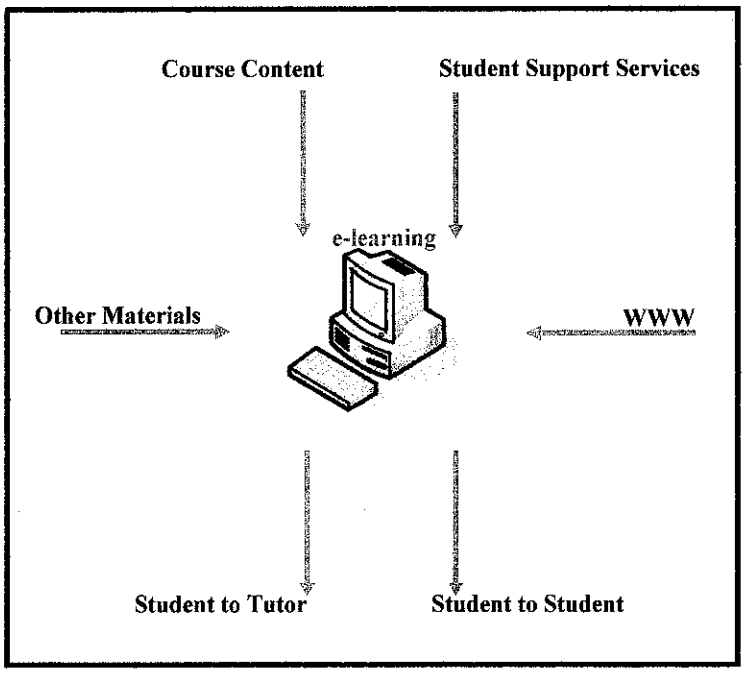


Figure 2.0 Wired Virtual Learning Environment of Today (Miziana Abdul Rahman, 2005, p. 7.)

M-Learning

M-learning is mobile learning: using mobile technologies (such as mobile phones and hand-held computers) to enhance the learning process. With a background of more than 4 years of trialling, research and development, M-learning has helped thousands of learners from all walks of life to develop their skills, confidence and motivation to learn.

Inspirational m-learning projects have taken place across the country in a variety of learning contexts, demonstrating the flexibility of this learning medium. The feedback received from the learners and tutors involved has shown that M-learning is a success. Inspirational M-learning projects have taken place across the country in a variety of learning contexts, demonstrating the flexibility of this learning medium. The feedback

received from the learners and tutors involved has shown that M-learning is a success. [18]

2.3 Challenges in M-Learning

There are many things that are different when comparing mobile device with a desktop. PC. Apart from that, the issue of new technology that still needs intensive development with challenges where variety of platforms now being used and non-compatibly between these platforms needs to be taken into account. For examples; existing PC applications (especially heavy text applications like email) are not easily integrated into mobile technology.

Hand-held computers are becoming more common. The quality and capability are increasing as costs continue to drop. However, M-learning faces two major technological issues, which the first one is managing learning through intermittent connection. In order to function as an effective learning device, the application must be able to link with Learning Management System on the same platform. The users need to access all materials and exchange information that is independent of system preferences. Let say the Web connection is broken; all information will be lost until the connectivity is restored.

The second issue is device - independent delivery. M- Learning solution must work for range of devices. In addition, a broad range of devices precludes specific solutions. For example is eXtensible Markup Language (XML) that can specify the content and determine on how it will appear for different type of devices. It will require some revisions to the content development process, and associated tools. [9]

When we try to transfer services provided by E-learning platform into services in an M-Learning platform, we can identify some of them should change to fulfill the limitations of the small devices whereby some are imposable to be delivered in certain

context. Therefore, access to the Web through personal electronic devices, with its small screen size, has been take not consideration for lots of researchers.

2.4 M-Learning Design issues

From the research have been made, there is also barrier need to be thinking of in this project in designing mobile learning application. The following figure examines the main dimensions of mobile learning systems in relation to (currently available) technologies. [12]

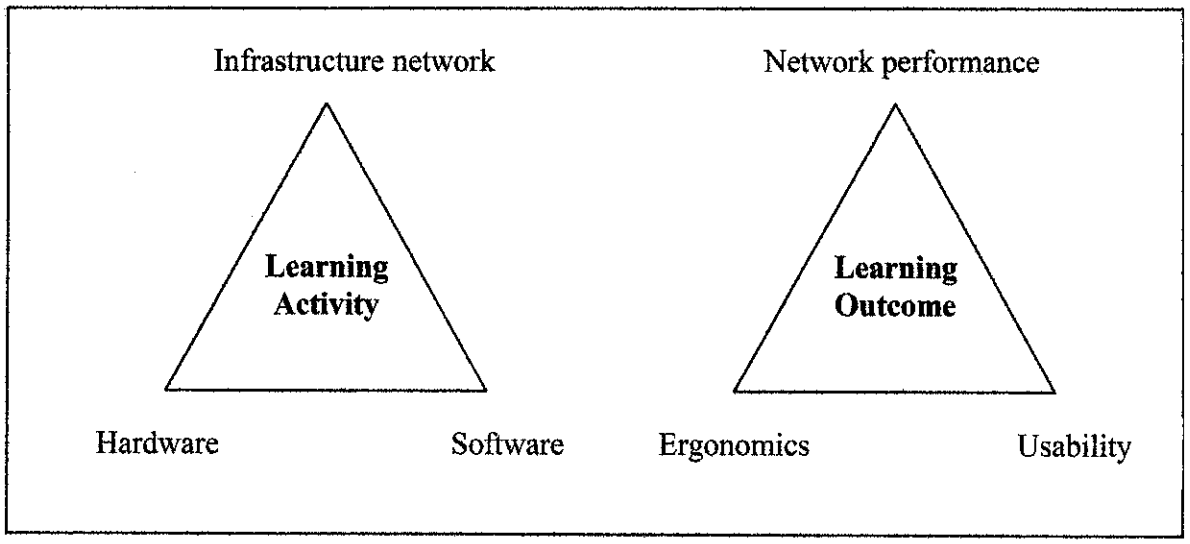


Figure 2.1 Dimensions for the (a) design and (b) evaluation of mobile learning systems (Miziana Abdul Rahman, 2005, p. 13.)

According to Figure 2.1, the choices of infrastructure networks, hardware and software are interdependent with the design of the learning activities. The Infrastructure network component includes decisions about the selection of communication networks and, in the case of location-aware applications, the selection of positioning systems. Example of communication networks such as satellites, wireless local networks, moble telephony and others. Selection of the Infrastructure network based on number of user,

security requirements, the costs to the provider, the need for range, connectivity data access, the time and place used and also possible interference with other devices.

Besides, the issue of designing M-Learning including the factor listed below.

- The limited processing power and resources
- The variety of screen sizes and the general low resolution of the display
- The variety of different input possibilities
- The variety of different operating systems

Especially the first two restrictions lead to the conclusion that not every mobile phone is suitable for a mobile learning application. Therefore more powerful devices are needed, like the next generation of mobile phones called Smartphones. Smartphones are a combination of a Personal Digital Assistant (PDA) and a mobile phone. They could be described as a pocket computer that can also be used as a mobile phone.

Most of the new mobile phones available to date are in fact Smartphones or even wireless Handhelds. The definition of a wireless Handheld is similar to the definition of a Smartphone. In general, wireless handhelds are more powerful than Smartphones and look more like a PDA.

CHAPTER 3
METHODOLOGY/PROJECT WORK

3.1 PROCEDURE IDENTIFICATION

The methodology that will be used in this project is the waterfall development-based where the operations moves forward from phase to phase. Deliverables for each phase presented to project supervisor for approval.

The advantages of using this methodology are:

- Identify system requirements before start the development of each implementation phase begins.
- Minimize requirement changes as the project proceeds.

3.1.1 Phases of Project Development

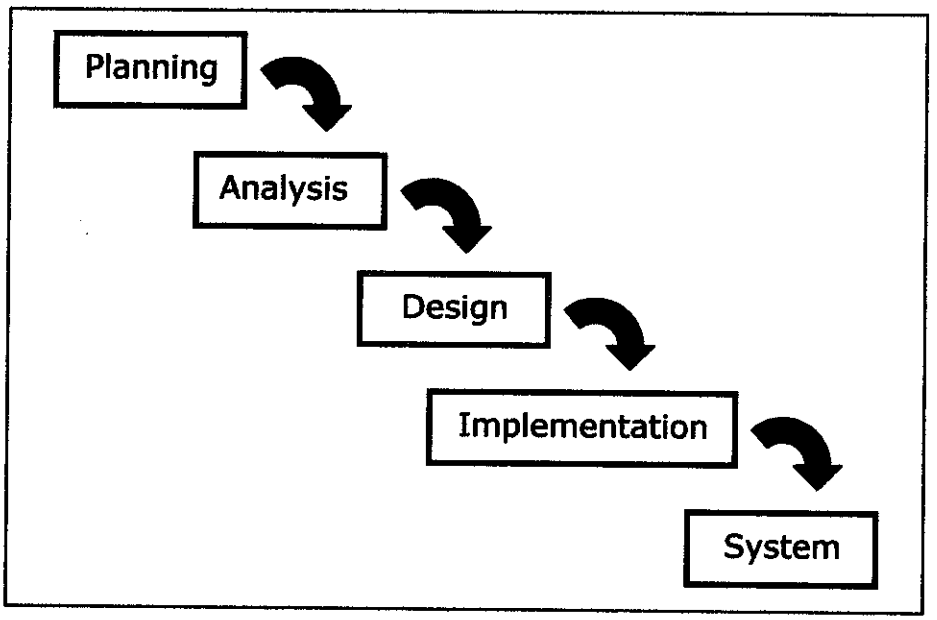


Figure 3.0 Project Development

3.1.1.1 Planning Phase

In the planning phase, the author chooses among several factors to set the application goals. These goals include anticipating and deciding on the target audience, purpose and objectives for the information. Besides, the author has planned for all stages involved in the completion of this project and possibility error that might be occurred.

3.1.1.2 Analysis Phase

In order to improve the application's quality, the author did gathering and comparing about the application and its functionality. It is done to help the author in decision making to gone through the process of planning, design and implementing. Here, the author comes out with system conceptual model as the base of this project. Please refer to appendix A-1 to system conceptual flow chart.

3.1.1.3 Design Phase

For the design phase, the author concentrates on the limitation and the constraint in developing this project to implement it via mobile devices. Here, the application's purposes, audience and objective is very important in implement this project. Besides, the author needs to do some research in designing the interface that is user-friendly and emphasizes in terms of its clarity and user classification. The interface will continue design in the stage of this project.

3.1.1.4 Implementation Phase

In this phase, the author builds this system by using the available tools. The author uses Java 2 Micro Edition (J2ME) integrate with the NetBeans as the development tool and Java as the language in building it. Basically, this is the main phase that currently being focus by the author in completing the project.

3.2 TOOLS/EQUIPMENT REQUIRED

These are the tools that are to be used throughout the project:

Table 3.0 Tools and Utilities

NO	ELEMENTS	TOOLS
1	Documentation	▪ Microsoft Word ▪ Microsoft Power Point ▪ Microsoft Visio
2	Hardware	▪ Hard disk space 1.5 GB (at least) ▪ 166 MHz processor or higher ▪ 256 MB RAM or higher
3	Software	▪ J2ME (Java 2 Micro Edition) Wireless Toolkit ▪ NetBeans IDE 4.1 ▪ JCreator

Following are the details for the tools and equipment required:

3.2.1 Hardware Requirements

Wireless devices have to meet certain criteria to be able to support J2ME. To run the KVM efficiently with the CLDC libraries, devices must have at least:

- 160KB to 512KB of total memory budget available for the Java platform.
- A 16-bit or 32-bit processor with 25MHz speed.
- Low power consumption, often operating with battery power.
- Connectivity to some kind of network, often with wireless, intermittent connection and with limited (often 9600bps or less) bandwidth.
- 128 kilobytes of non-volatile memory available for Java virtual machine and CLDC libraries.
- 32 kilobytes of volatile memory available for the Java runtime and object memory.

The MIDP impose the following requirements on hardware:

- Display
    - Screen-size: 96x54
    - Display depth: 1 bit
    - Pixel shape (aspect ratio): approximately 1:1

- Input
    - One or more of the following user-input mechanisms: one-handed keyboard, two-handed keyboard, or touch screen.

- Memory
    - 128 kilobytes of non-volatile memory for MIDP components.
    - 8 kilobytes of non-volatile memory for application-created persistent data.
    - 32 kilobytes of volatile memory for Java runtime (for example, the java heap).

- Networking
    - Two-way wireless, possibly intermittent, with limited bandwidth.

3.2.2 Software Requirements

J2ME (Java 2 Micro Edition)

The software needed in order to develop the application is J2ME (Java 2 Micro Edition) platform which is implemented by using Java language. Any Java platform can be used in writing the codes of the application, for example JBuilder.

JDK1.3 is required for J2MEWTK to function properly. Forte can be installed optionally since J2MEWTK can be run standalone or can be run as an integrated component with Forte for Java. The J2ME Wireless Toolkit installation package is also required for the application to run.

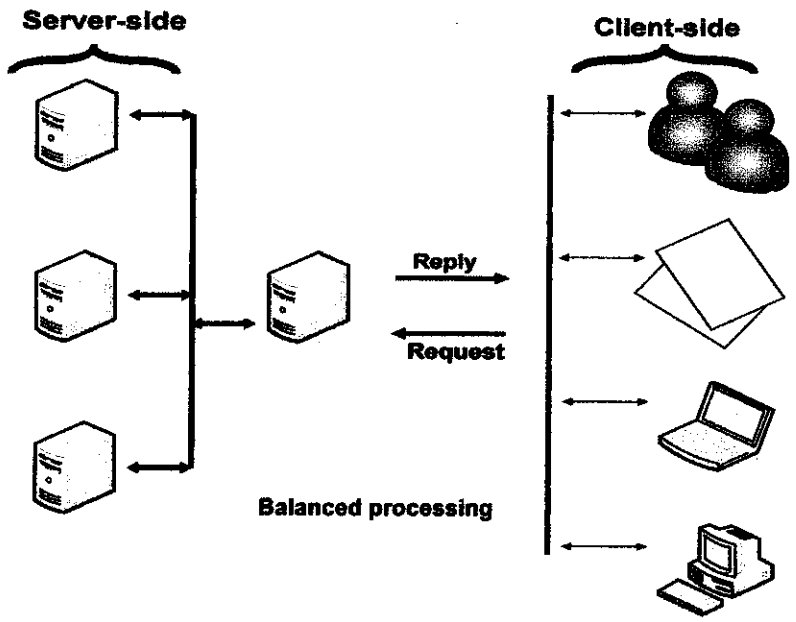


Figure 3.1 A J2ME application is a balance between local and server-side processing (James Keogh, 2004, p. 79.)

J2ME allows developers to use Java and the J2ME wireless toolkit to create applications and programs for wireless and mobile devices. J2ME consists of two elements -- **configurations** and **profiles.**

J2ME configurations

There are two configurations for J2ME.

1. Connected Limited Device Configuration (CLDC)

- Designed for 16-bit or 32- bit small computing devices with limited amounts of memory.
- Usually have between 160KB and 512KB of available memory and are battery powered.
- Use an inconsistent, small-bandwidth network wireless connection.
- Not have a user interface.

- Use the KJava Virtual Machine (KVM) implementation- strip down version of the JVM.
- Devices: pagers, personal digital assistants, cell phones dedicated terminals and handheld consumer devices between 128 KB and 512 KB of memory.

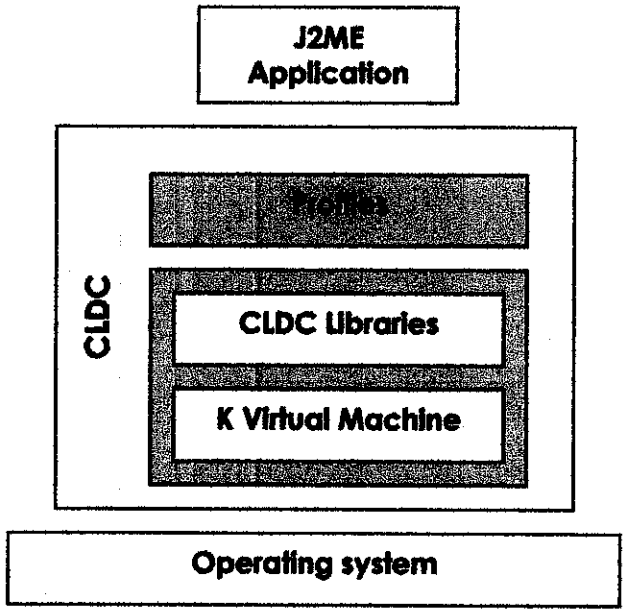


Figure 3.2 CLDC (Dreamtech Software Team, 2002, p. 58)

2. Connected Device Configuration (CDC)

- Devices use a 32-bit architecture.
- Have at least 2 MB of memory available.
- implement a complete functional JVM
- Devices: smart phones, digital set-top boxes, home appliances, navigation system and point-of-sale terminals.

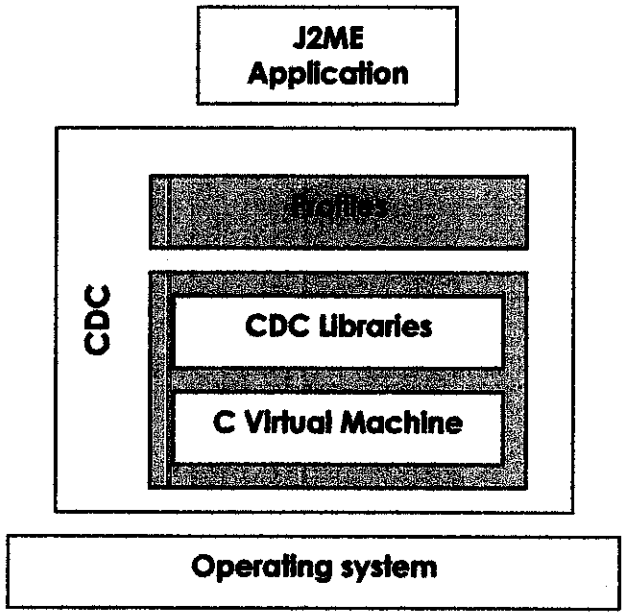


Figure 3.3 CDC (Dreamtech Software Team, 2002, p. 60)

J2ME Profiles

A profile consists of Java classes that enable implementation of features for either a particular small computing device or for a class of small computing device. Small computing technology continues to evolve, and with that, there is an ongoing process of defining J2ME profiles. Seven profiles have been defined. There are:

1. Foundation profile –
    - Used with CDC configuration.
    - Contains core Java classes.

2. Game profile
    - Also used with CDC configuration.
    - Contains necessary classes for developing game applications for any small computing device.

3. Mobile Information Device Profile (MIDP)
    - Used with CLDC configuration.
    - Contains classes that provide local storage a user interface and networking capabilities
    - Used with wireless Java applications.

4. PDA Profile (PDAP)
    - Used with CLDC configuration.
    - Contains classes that utilized sophisticated resources found on personal assistant.

5. Personal Profile
    - Used with CLDC configuration and Foundation profile.
    - Contains classes to implement a complex user interface.

6. Personal Basis Profile
    - Similar to Personal Profile.
    - Provide classes to implement a simple user interface.

7. RMI Profile
    - Used with CLDC configuration and Foundation profile.
    - Provide Remote Method Invocation.

J2ME Architecture

The modular design of the J2ME architecture enables an application to be scaled based on constraints of a small computing device. Instead, J2ME architecture consists of layers located above the native operating system, collectively referred to as the Connected Limited Device Configuration (CLDC). The CLDC, which is installed on top of the operating system, forms the run-time environment for small computing device.

The J2ME architecture comprises three software layers (Figure 3.3). The first layer is the configuration layer that includes the Java Virtual Machine (JVM), which directly interacts with the native operating system. The configuration layer also handles interactions between the profile and the JVM. The second layer is the profile layer, which consists of the minimum set of application programming interfaces (APIs) for the small computing device. The third layer is the Mobile Information Device Profile (MIDP). The MIDP layer contains Java APIs for user network connections, persistence storage, and the user interface. It also has access to CLDC libraries and MIDP libraries.

A small computing device has two components supplied by the original equipment manufacturer (OEM). These are classes and applications. OEM classes are used by the MIDP to access device-specific features such as sanding and receiving messages and accessing device-specific persistent data. OEM applications are programs

provided by the OEM, such as an address book. OEM applications can be accessed by the MIDP.

A word of caution: accessing OEM classes and OEM applications from the MIDP restricts the portability of a J2ME application since not all small computing device manufacturers use the same OEM classes or OEM applications.

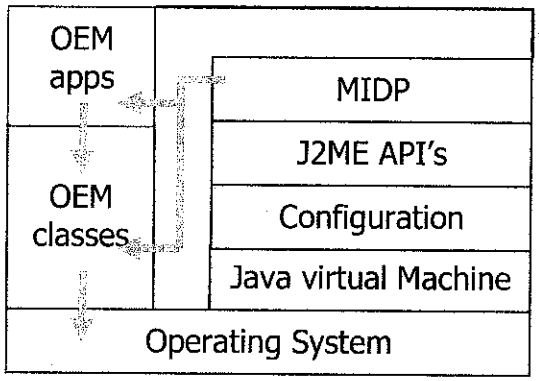


Figure 3.4 Layers of the J2ME Architecture (Dreamtech Software Team, 2002, p.106.)

3.3 REQUIREMENT ANALYSIS

3.3.1 Nonfunctional Requirements

Operational requirements

- The system will operate based on J2ME platform.
- The system will operate trough emulator and if possible, implemented in the hand phone.

Performance requirements

- Launching the application
- Launching time elapsed to opening application must be less than 2 second.
- The application should be available as long as the hand phone (or any devices used) is on and ready.

Security Performance

- No special security actions are required as it is not part of this code.

Cultural and political requirements

- No special cultural and political requirements are anticipated

3.3.2 Functional Requirements

Flexibility

- Platform independency
- The site will not use java.
- The site will not use PDF documents.

Functionality

- Users will be able to access to the log in page.
- Users will be able to search the courses, grade, assignment and schedule.
- User will be able to browse all the information in the M-Learning.
- The system will be able guide the users throughout the application by providing user friendly navigation to easily access to the overall the application.

CHAPTER 4
RESULT AND DISCUSSION

4.1 FINDINGS

4.1.1 M-Learning in the Education Field

M- Learning is a natural extension of E-Learning and has the potential to make learning even more widely available and accessible than existing E-Learning environment. M- Learning can contribute to improve the quality of education. It offers opportunities to optimize interaction between lecturers and students.

Mobile learning, the next generation of the computer-aided and multimedia based Learning is based on mobile phones. It can be said, that the great majority of the population has a mobile phone and carries it with them most of the time. Because of this fact the independency of time and space for learning is fulfilled to nearly a hundred percent.

As a consequence the main advantage of mobile learning is learning wherever and whenever you want to learn. You can use idle periods for learning. For example: The times while you are traveling, while you are waiting for the bus or while you are waiting at the restaurant or at the train-station. Nearly every unused and wasted time can now be used for efficient and effective learning. Therefore mobile learning will be an important instrument for lifelong learning, because it will help us to use our time more efficiently.

4.1.2 Advantages of Mobile Learning

- ***Seamless access to learning resources:***
  With mobile learning, you can learn and study anywhere - from the classroom to your desktop, or laptop to your pocket. A true mobile learning system allows users to take a course on any device.

- ***Freedom, power, and choice:***
  M-learning students can choose where, when, and how they'll study. The new range of options includes online synchronized, online self-paced, downloaded courseware, and computer-based training. M-learning offers new levels of freedom with the ability to exercise control over learning patterns.

- ***Organized productivity:***
  With only a cell phone, handheld device, PDA, or hybrid unit, users can access administrative functions, download courses, and review their learning history through a learning management system. M-learning offers an efficient way for learners to access key information and maximize their time.

- ***Flexible, portable convenience:***
  The ability to customize learning schedules is a key advantage to m-learning. Learners are not restricted to a specific physical environment, a particular delivery channel, or a fixed set of times for undertaking training and education. Using the latest technology, students can update their knowledge base on a just-in-time basis to prepare for meetings or presentations.

4.1.3 J2ME Wireless Toolkit Installation Process

Getting started with developing applications (henceforth called "MIDlets") for the J2ME platform is easy. Although device manufacturers install and prepackage their devices with this JVM (and associated APIs), we still need to install the J2ME Wireless Toolkit 2.2 on our development machine. Before that, however, we must also have the Java Development Kit (JDK), version 1.4.2 or greater, installed.

We need this Toolkit because it contains tools that are important in generating MIDlets. This Toolkit provides the development environment for the MIDP 2.0 and CLDC 1.1 (and for MIDP 1.0 and CLDC 1.0, since these parameters are backwards compatible), and it provides the optional packages required for the optional libraries, like 3D and Mobile Media applications. Lastly, it provides the ability to sign your MIDlets so that they can be authenticated before installation on a remote mobile device.

Once you download the installation package for the Toolkit, install it in the directory of your choice. The default, on Windows, is *C:\WTK23*, and this will be the installation directory for the examples in this series as well.

To install all the toolkit and packages needed, we have to only double click the .exe files and follow the instruction the installer process gives.

4.2 DISCUSSION

4.2.1 Results

Throughout the research done by author, the author find out that the use of J2ME to develop Mobile Learning Engine (MLE) can be executed. However, lots of research need to be done in order to understand the application concept and on how to integrate it with mobile devices.

The following figure will give the reader roughly idea about how the project will be integrate within its component.

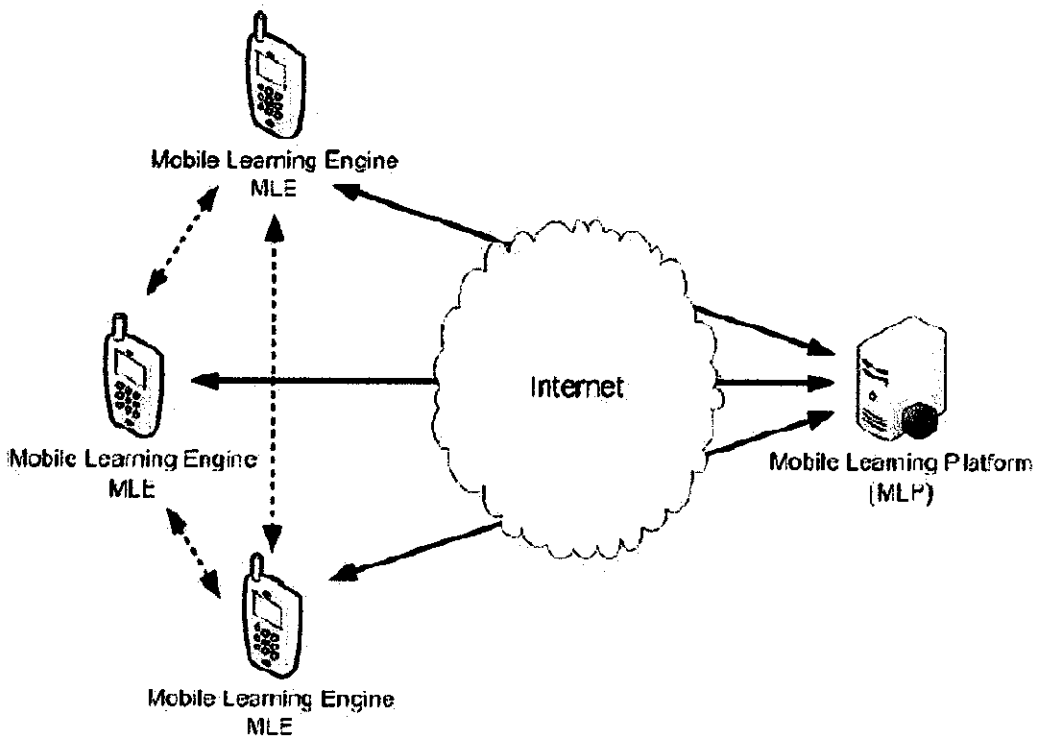


Figure 4.0 Shows the relationship between the MLE and the learning platform

Source: http://drei.fh-joanneum.at/mle/start.php?sprache=en&id=1

The distribution of new learning objects is done by a central internet server: the learning platform. Here all available learning objects are hosted. If the user wants to use a new learning object he can download it right away from the server with the MLE. If the learning object has once been downloaded, it can be used without a new connection to the server.

As the author work on the implementation phase, the focus is in building the application. By using the emulator as stated in previous section, conceptual view of how the system will look on the real mobile device, which means that it, is able to emulate the real application that is to be used. So here, the author comes out with sample output that going to construct. From the sample, perhaps some basic ideas of implement the **M-Learning** can be implemented.

Based on the research above, prototype of M-Learning was developed using the specified tools. In this M-Learning, there are several modules which is My Courses, My Grade, My Schedule and My Assignment. Each of the modules was developed based on the research done earlier. Each of the module are analyzed to meet the user requirements as well as to accommodate the technology available nowadays.

The main concern here in develops the modules of M-Learning is the screen limitation of the emulators that is being used to test the application. As the time constraint, emulator being used is to have the clear view on how the system will look like using J2ME Wireless Toolkit together with the NetBeans IDE. Although using the emulator, the system has not fully emulated successfully to show the application. So that the author have to codes the module separately in order to show the real application.

In general, the system supposed to work like normal system where users still need to login first before can enter the system. If the system not authenticated first the user not allowed to access to the page. Please refer to Appendix A-1 for the conceptual model of M-learning.

4.2.2 System Construction

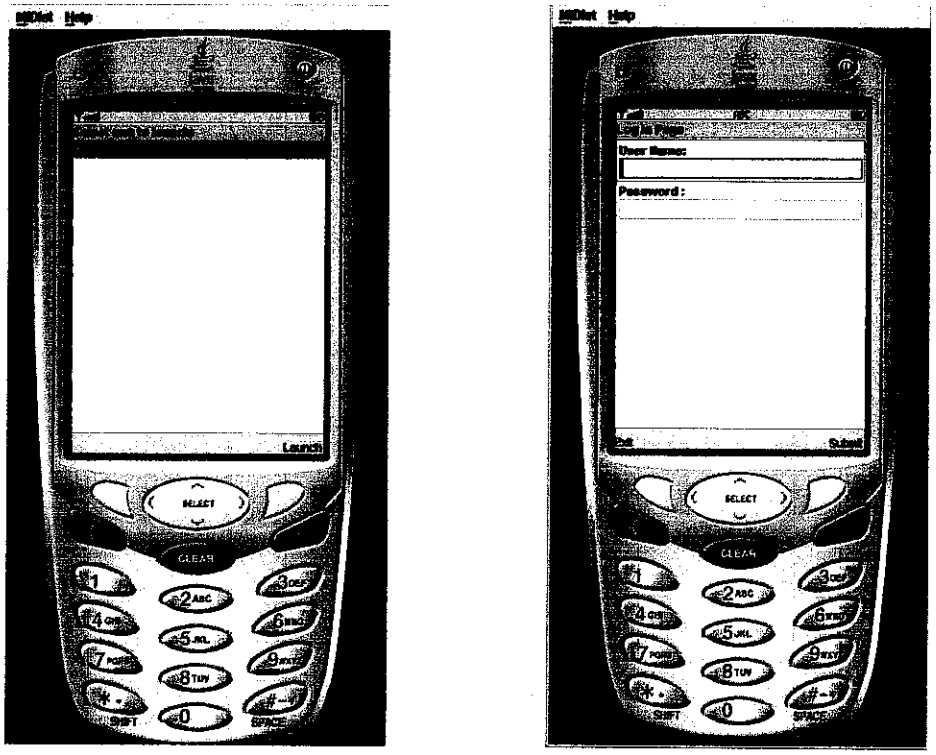


Figure 4.1 Login page of M-Learning

Figure 4.1 above shows Login page of M-Learning where the user need to enter their username and password.

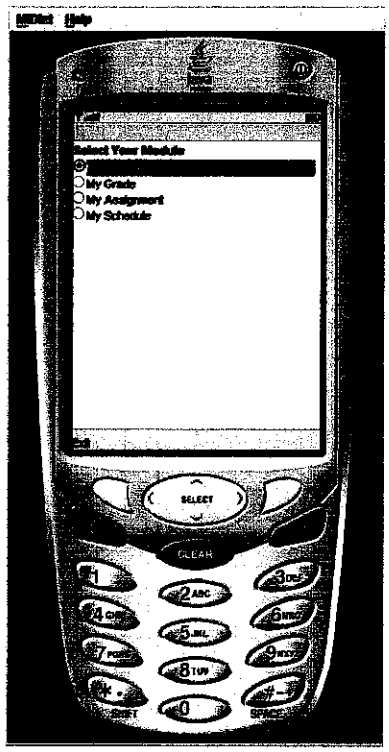


Figure 4.2 M-Learning module

Figure 4.2 above shows M-Learning module that consist of My Courses, My Grade, My Schedule and My Assignment. If the user chooses My Courses, a list of subject will displayed as shown in Figure 4.3

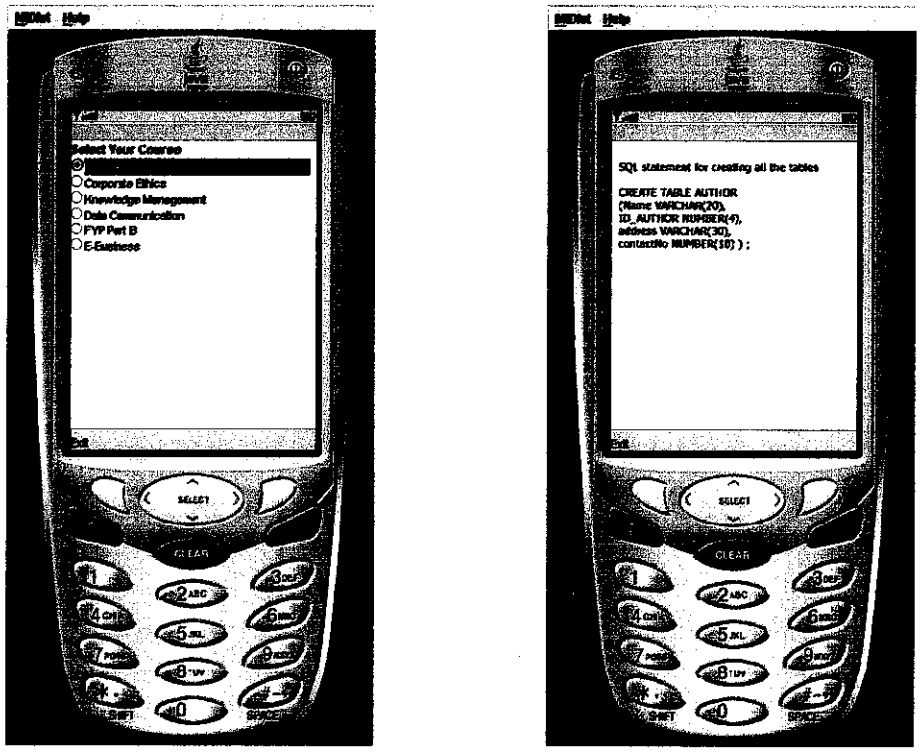


Figure 4.3 My Courses Module

Figure 4.3 above shows My Courses Module that will list down all the courses taken by a student within that semester.

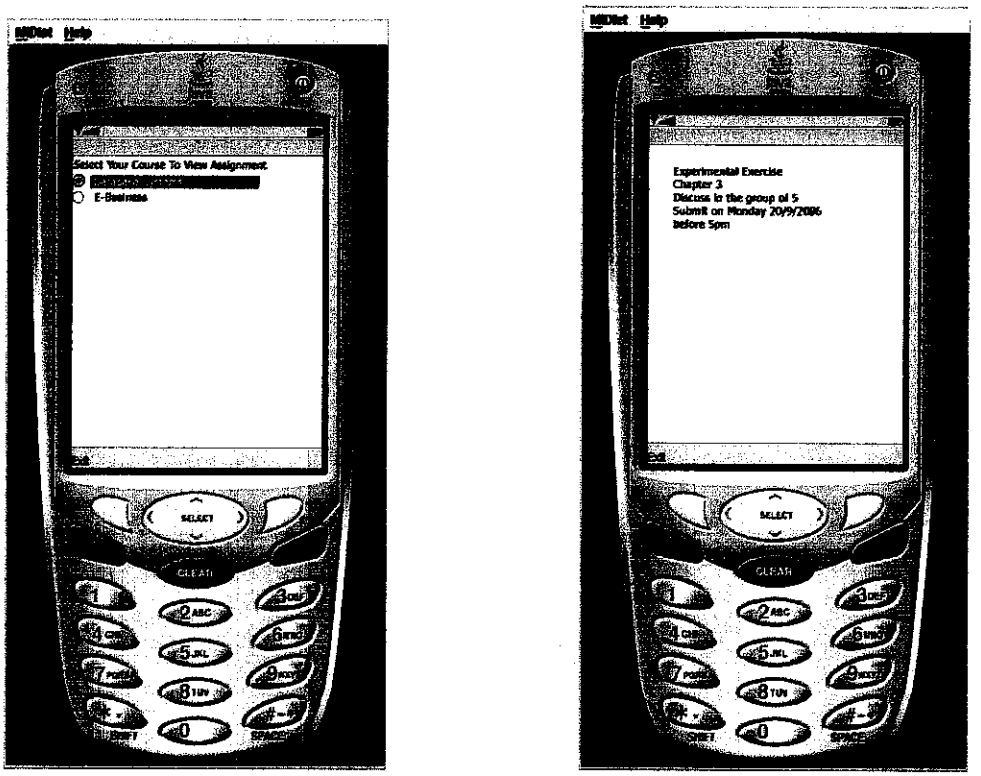


Figure 4.4 My Assignment Module

The My Assignment module enables students to open text file from lecturers to view the assignments that has been given. The above figure is the example of the page.

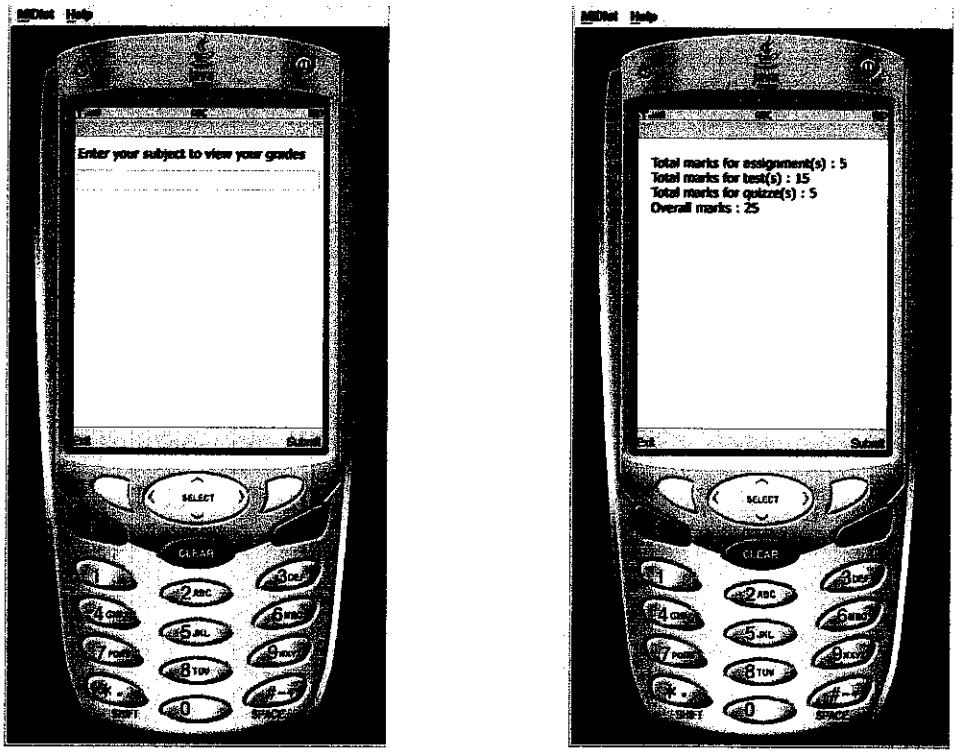


Figure 4.5 My Grade Module

Figure 4.5 above shows My Grade Module that will list down all the marks taken by a student within that semester.

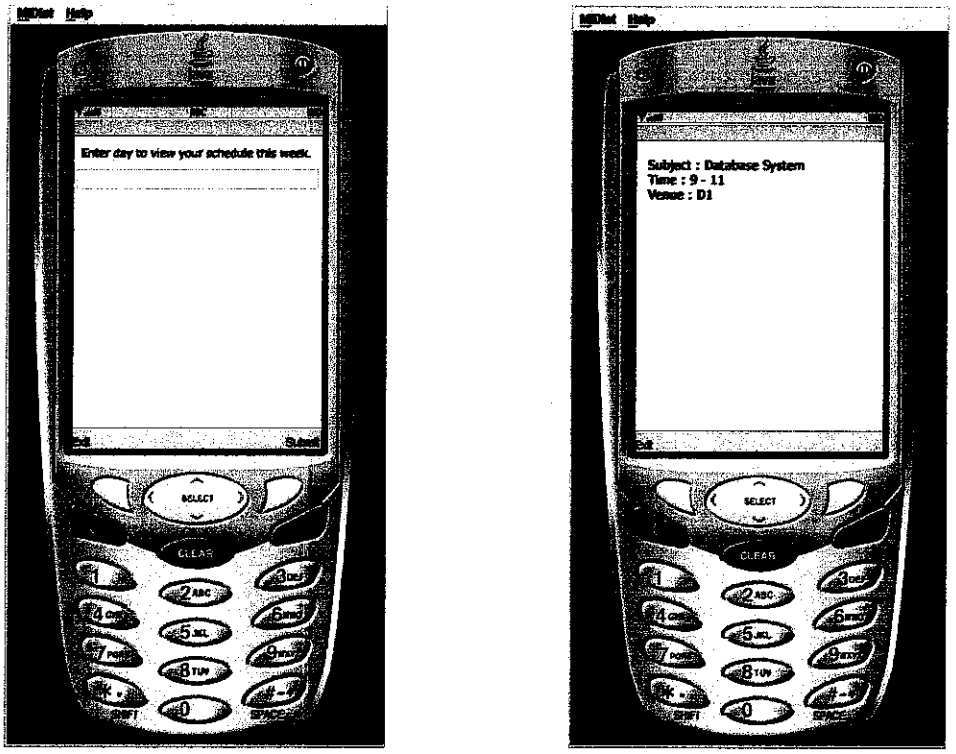


Figure 4.6 My Schedule Module

This My Schedule module allows users to search for a schedule according to day within that week.

CHAPTER 5
CONCLUSION

5.1 Relevancy to the objective

With the increasing use of mobile phones and handheld devices among teenagers or student, the opportunity of implement M-Learning in academic institution is quite high. However, this project will involve with extensive research on the technology used, the limitations and advantages of M-Learning. The concept of just-in time learning is apply in this project where the students are able to access the information wherever they want using the devices like mobile phone.

Besides, this system provides user with new look of learning style. The Mobile Learning Engine enables you to learn **wherever** you want to learn, **whenever** you want to learn and **whatever** you want to learn. Here, in this system, user can choose either to download or only view the course. This in turn introduces the concept of flexibility, however, its demand new pedagogies, and new approaches to deliver the course. A user friendly system is achieved by integrating all the research area in the development process.

Although there are constraints that need to be considered such as cost, it's become an interesting issue in the future as people adapt with the changing of technology.

5.2 Suggested future work for expansion and Continuation

For future recommendation, as the emulator is not run well, the author suggests that more study in using J2ME and NetBeans IDE need to be done. By having lots of exercises, one person can experience better in using it. The author believes if there are more allocated time, she will be able to come out with running program. Perhaps, implementing in real devices can be done accurately.

As we know, the emulator is not functioning well. So that after overcoming the problem, perhaps several modules will be included into the system. As the previous system focuses only to one subject only, in future more subjects will be implementing within one system.

Besides, the system can add on with functionality such as quiz for quick revision and for sharpen the understanding of student. This modules able the student to access it whether there are not inside the campus. This module also has been suggested by Mr. Nordin, one of the evaluator during the presentation.

In future, it is very useful of mobile learning as the system allow the communication between student and student or between lecturers and student so that interactive learning environment can be implemented. As the communication is one main demands of constructivism. Synchronous and asynchronous communication instruments would be for example: Chat, email, video and audio conference, blackboard and others. These instruments should help build up relationships between the students and to the instructors. Students with a similar knowledge can discuss problems or simply ask a human teacher, no matter where the people are located. Here, the concept of just-in-time learning is achieving its goals.

Finally, the author suggests that user's personalization to be included into the mobile system. This personalization enable the user to change the way the system will be displayed in terms of color, font and others based on their preferences.

REFERENCES

[1] www.formatex .org/micte2005/147.pdf

[2] http:/fieengrng.pitt.edu/fie2005/papers/1558.pdf

[3] www.mleran.org.za/cb/papers/Silander.pdf

[4] http://www3.telus.net/~kdeanna/mlearning/

[5] Tatar, D., Roschelle, J., Vahey, P. & Penuel, W. R. (2003), Handhelds Go to School: Lessons Learned. Computer, 36, 9, 30-37.

[6] http://hotlavasoftware.com/newsdesk_info.php?newsdesk_id=100

[7] Geddes, S.J. 2004, "Mobile learning in the 21st century: Benefits for learners" in the knowledge Tree, *An ejournal of flexible learning n VET*, Australian flexible learning framework: Australian national training Authority. Edton 6, October.

[8] http://msdn2.microsoft.com/en-us/library/excw86bz.aspx

[9] Quinn, C., (2000), Mobile, Wireless, n-Your _Pocket Learning. Retrieved September 25th, 2002 from World Wide Web: (URL: http:/www.linezine.com/2.1/features/cqmmwiyp.htm)

[10] http://drei.fh-joanneum.at/mle/start.php?sprache=en&id=1

[11] www.elearnmag.org/subpage. cfm?section=research&article=6-1 - 26k

[12] Vavoula G., karagiannidis C., Designing Mobile Learning Experience, institute of Educational technology (2000)

[13] http://www.javaworld.com/javaworld/jw-02-2003/jw-0221-wireless.html

[14] http://www.devx.com/Intel/Article/17507

[15] www.javasun.com.my

[16] The complete reference J2ME by James Keogh

[17] Cracking the Code Wireless Programming with J2ME

[18] http://www.m-learning.org

[19] http://www.developer.com/ws/j2me/article.php/1378921

[20] http://www.wirelessdevnet.com/channels/java/features/j2me.html

APPENDIX A

Project Timeline and Milestone

Project timeline and Milestone (FYP Part 1)															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.	Selection of Project Topic														
	• Research on interested topic														
	• Submission of project proposal			●											
	• Topic Approval and Supervisor assigned				●										
2.	Requirements analysis and definition														
	• Project background identification														
	• Literature review														
	• Project relevancy analysis (survey)														
3.	Submission of Preliminary Report									●					
4.	System design														
	• System architecture														
5.	Submission of Interim Report										●				
6.	Interface development														
	• Interface design														

7.	Oral Presentation														•

Process • Milestone

Project timeline and Milestone (FYP Part 2)													
		1	2	3	4	5	6	7	8	9	10	11	12
1.	System Development												
	• Programming and coding												
2.	Submission of progress report			●									
3.	Implementation and testing												
	• Set up system												
4.	Verification												
	• Problem Identification												
5.	Maintenance												
	• Correcting error												
6.	Submission of Final Draft							●					
7.	Final Report								●				
8.	Exhibition										●		
9.	Oral Presentation										●		

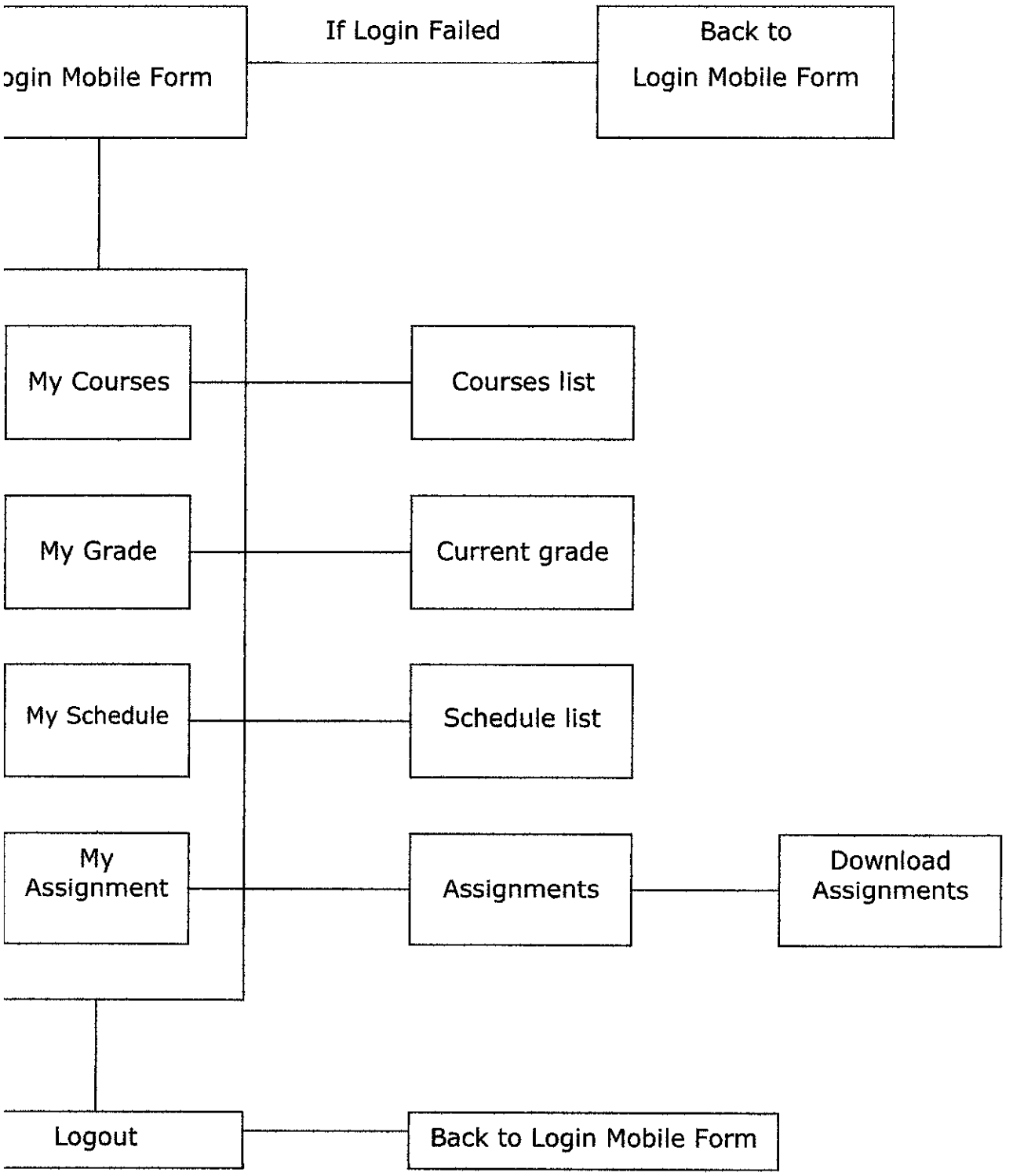
10.	Submission of Dissertation												•

 Process

• Milestone

APPENDIX B

Appendix B **Conceptual Flow model of M-learning**



APPENDIX C

Appendix C Overview of the three platforms

	.Net Compact Framework	J2ME Connected Device Configuration	J2ME Connected Limited Device Configuration
Device requirement	**Powerful, expensive**	**Powerful, expensive**	**Cheap, pervasive**
Cost	High	High	Medium
Market focus	**Enterprise**	**Enterprise**	**Consumer and enterprise**
Language support	C#, VB.Net	Java	Java
Platforms	**Pocket PC, Windows CE**	**Major mobile platforms except Palm OS**	**All mobile platforms**
Byte code compatibility	Standard .Net CLR	Standard Java 2	Not compatible with J2SE or CDC
API compatibility	**Subset of .Net**	**Subset of J2SE plus standard optional packages**	**Partial compatibility with CDC with additional standard optional packages**
Native APIs	P/Invoke; consistent across supported devices	JNI; device- and OS-specific	N/A
Development tools	**VS.Net 2003**	**Command line, vendor SDKs, CodeWarrior, and WebSphere**	**Command line, vendor SDKs, all major Java IDEs**
Specification process	Single company	Community	Community
Service gateway	**N/A**	**Run gateways as OSGi servlets; run gateway clients via vendor-specific SDKs**	**Run gateway clients via vendor-specific SDKs**
Security model	Simplified .Net model	Full Java security manager	Limited Java 2 model supplemented by OTA specification
Client installation	**ActiveSync, Internet Explorer download**	**Sync, download**	**Formal OTA specification**
Life cycle management	N/A	OSGi for gateway apps, J2EE Client Provisioning Specification for generic clients	Included in OTA spec, works with J2EE Client Provisioning Specification

APPENDIX D

SAMPLE OF SOURCE CODES

SAMPLE SOURCE CODE FOR M-LEARNING

Source code for Log in Mobile Form

```
/*
 * MLearningMIDlet.java
 *
 * Created on October 10, 2006, 5:00 PM
 */

package MLearning;

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

/**
 *
 * @author  Pekjah
 * @version
 */
public class MLearningMIDlet extends MIDlet
                    implements CommandListener

{
 private Display display;
 private Form form;
 private List list;
 private Command submit;
 private Command exit;
 private TextField textfield1;
 private TextField textfield2;
  public MLearningMIDlet()

 {
 display = Display.getDisplay(this);
 submit = new Command("Submit", Command.SCREEN, 1);
   exit = new Command("Exit", Command.EXIT, 1);
 textfield1 = new TextField("User Name:", "", 30, TextField.ANY);
 textfield2 = new TextField("Password :", "", 30, TextField.ANY | TextField.PASSWORD);
 form = new Form("Log In Page");
 form.addCommand(exit);
 form.addCommand(submit);
 form.append(textfield1);
 form.append(textfield2);
 form.setCommandListener(this);
 }
 public void startApp()
 {
 display.setCurrent(form);
 display.setCurrent(list);
 }
 public void pauseApp()
 {
 }
 public void destroyApp(boolean unconditional)
```

```
{
}
public void commandAction(Command command, Displayable displayable)
{
 if (command == submit)
 {
  list = new List("Select one", List.EXCLUSIVE);
  list.append("Male", null);
  list.append("Female", null);
  exit = new Command("Exit", Command.EXIT, 1);
  submit = new Command("Submit", Command.SCREEN,2);

  list.addCommand(exit);
  list.addCommand(submit);
  list.setCommandListener(this);
 }
 else if (command == exit)

 {
   destroyApp(false);
   notifyDestroyed();

 }
 }

 }
```

Source code for My Courses Module

```
/*
 * CourseMidlet.java
 *
 * Created on October 11, 2006, 10:30 AM
 */

package Courses;

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

/**
 *
 * @author  Pekjah
 * @version
 */
public class CourseMidlet extends MIDlet
        implements ItemStateListener, CommandListener
{
 private Display display;
 private Form form;
 private Command exit;
 private Item selection;
 private ChoiceGroup radioButtons;
 private int defaultIndex;
 private int radioButtonsIndex;
 public CourseMidlet ()
 {
  display = Display.getDisplay(this);
  radioButtons = new ChoiceGroup(
                  "Select Your Course",
                   Choice.EXCLUSIVE);
  radioButtons.append("Database System", null);
  radioButtons.append("Corporate Ethics", null);
  radioButtons.append("Knowledge Management", null);
  radioButtons.append("Data Communication", null);
  radioButtons.append("FYP Part B", null);
  radioButtons.append("E-Business", null);

  radioButtons.setSelectedIndex(defaultIndex, true);
  exit = new Command("Exit", Command.EXIT, 1);
  form = new Form("");
  radioButtonsIndex = form.append(radioButtons);
  form.addCommand(exit);
  form.setCommandListener(this);
  form.setItemStateListener(this);
 }
 public void startApp()
 {
   display.setCurrent(form);
 }
 public void pauseApp()
 {
```

```
}
public void destroyApp(boolean unconditional)
{
}
public void commandAction(Command command,
                    Displayable displayable)
{
 if (command == exit)
 {
   destroyApp(true);
   notifyDestroyed();
 }
}
public void itemStateChanged(Item item)
{
 if (item == radioButtons)
 {
   StringItem msg = new StringItem("Your color is ",
        radioButtons.getString(radioButtons.getSelectedIndex()));
   form.append(msg);
 }
}
}
```

Source code for M-Learning Module

```
/*
 * ModuleMidlet.java
 *
 * Created on October 11, 2006, 9:52 AM
 */

package Module;

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

/**
 *
 * @author  Pekjah
 * @version
 */
public class ModuleMidlet extends MIDlet
        implements ItemStateListener, CommandListener
{
 private Display display;
 private Form form;
 private Command exit;
 private Item selection;
 private ChoiceGroup radioButtons;
 private int defaultIndex;
 private int radioButtonsIndex;
 public ModuleMidlet ()
 {
  display = Display.getDisplay(this);
  radioButtons = new ChoiceGroup(
                  "Select Your Module",
                   Choice.EXCLUSIVE);
  radioButtons.append("My Courses", null);
  radioButtons.append("My Grade", null);
  radioButtons.append("My Assignment", null);
  radioButtons.append("My Schedule", null);
  radioButtons.setSelectedIndex(defaultIndex, true);
  exit = new Command("Exit", Command.EXIT, 1);
  form = new Form("");
  radioButtonsIndex = form.append(radioButtons);
  form.addCommand(exit);
  form.setCommandListener(this);
  form.setItemStateListener(this);
 }
 public void startApp()
 {
  display.setCurrent(form);
 }
 public void pauseApp()
 {
 }
 public void destroyApp(boolean unconditional)
 {
```

```
  }
  public void commandAction(Command command,
                    Displayable displayable)
  {
   if (command == exit)
   {
     destroyApp(true);
     notifyDestroyed();
   }
  }
  public void itemStateChanged(Item item)
  {
   if (item == radioButtons)
   {
     StringItem msg = new StringItem("Your selection module is ",
          radioButtons.getString(radioButtons.getSelectedIndex()));
     form.append(msg);
   }
  }
}
```

APPENDIX E

J2ME Wireless Toolkit Installation Process

Getting started with developing applications (henceforth called "MIDlets") for the J2ME platform is easy. Although device manufacturers install and prepackage their devices with this JVM (and associated APIs), we still need to install the J2ME Wireless Toolkit 2.2 on our development machine. Before that, however, we must also have the Java Development Kit (JDK), version 1.4.2 or greater, installed.

We need this Toolkit because it contains tools that are important in generating MIDlets. This Toolkit provides the development environment for the MIDP 2.0 and CLDC 1.1 (and for MIDP 1.0 and CLDC 1.0, since these parameters are backwards compatible), and it provides the optional packages required for the optional libraries, like 3D and Mobile Media applications. Lastly, it provides the ability to sign your MIDlets so that they can be authenticated before installation on a remote mobile device.

Once you download the installation package for the Toolkit, install it in the directory of your choice. The default, on Windows, is *C:\WTK23*, and this will be the installation directory for the examples in this series as well.

To install all the toolkit and packages needed, we have to only double click the .exe files and follow the instruction the installer process gives.

Application Development and Running Procedure

After installation process has been done successfully, now we are ready to develop and run the application. There are three components needed in order to develop and run the application:

1. The .java file
2. J2ME Wireless Toolkit
3. WTK23 folder

All those components are provided inside the CD. In the CD, the .java file is named with **AddressMIDlet.java**. The step by step process of developing and running the application is shown by the following:

1. start > All Programs > Sun Java Wireless Toolkit 2.3 Beta > KToolbar

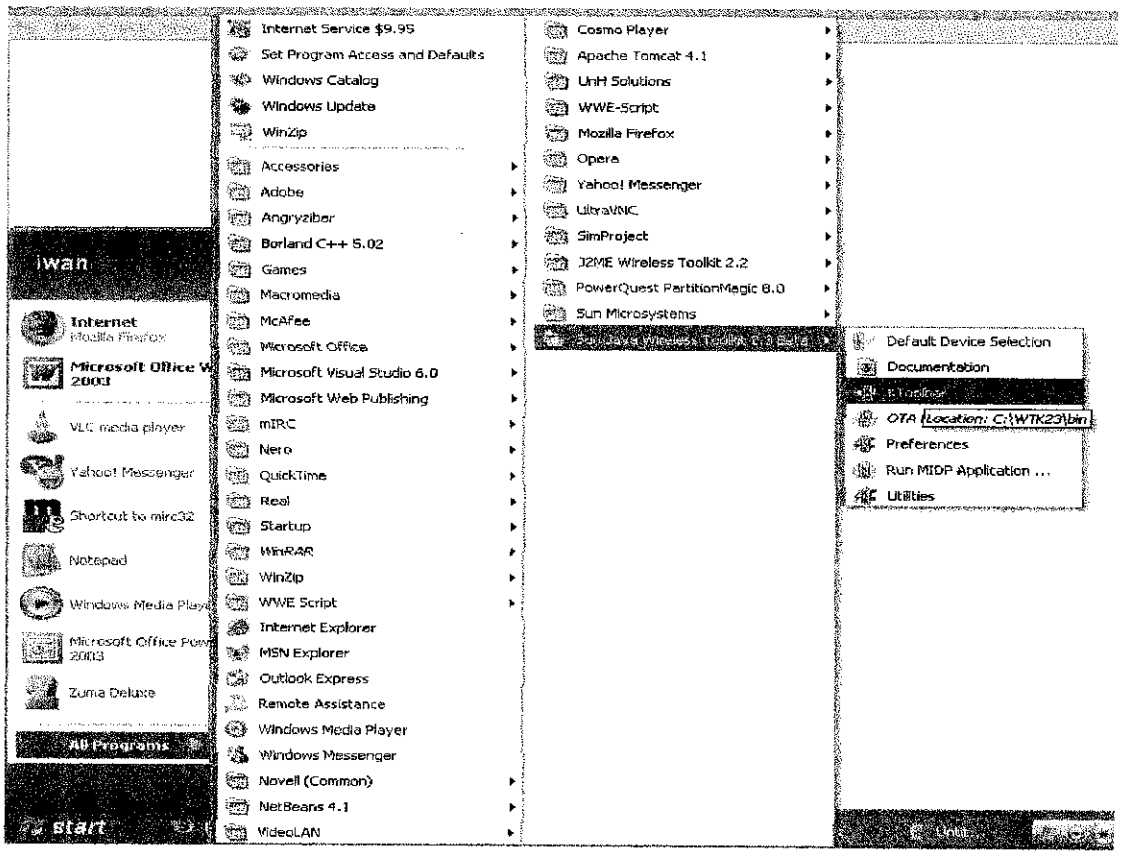


Figure 4: Location of J2ME Wireless

2. The following figure will appear as we start the J2ME application platform:

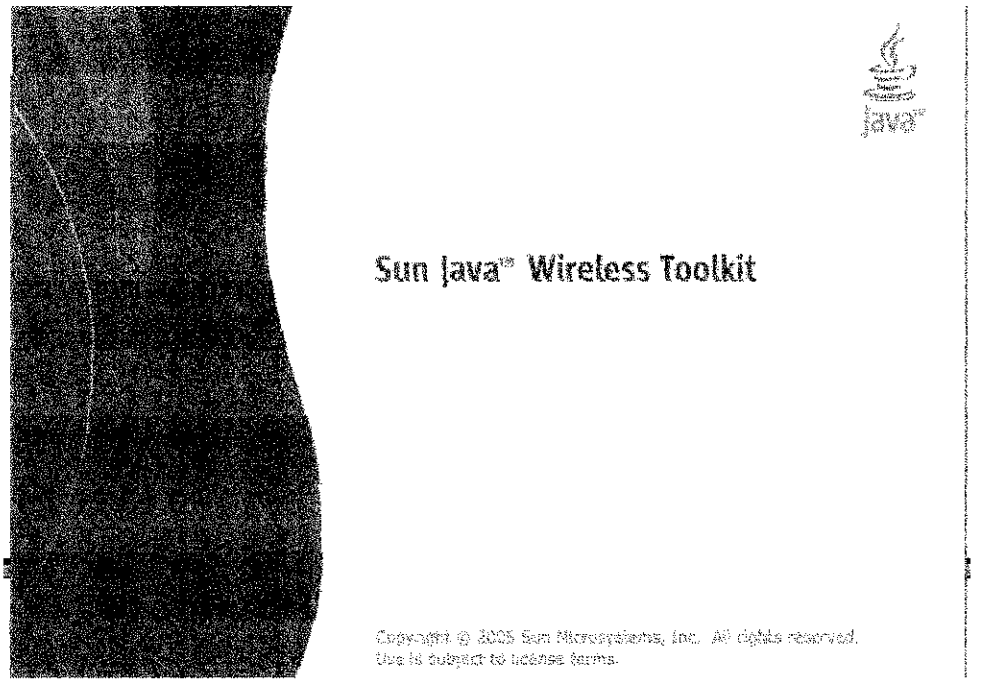


Figure 5: Sun Java Wireless Toolkit Startup

3. After starting up process is done, the platform of Sun java Wireless Toolkit appears. Here we may create new project or open existing project. **Important:** The project name and the MIDlet class name MUST be the same as the .java file that we have. In this case, the project name and MIDlet class name MUST be AddressBookMIDlet.

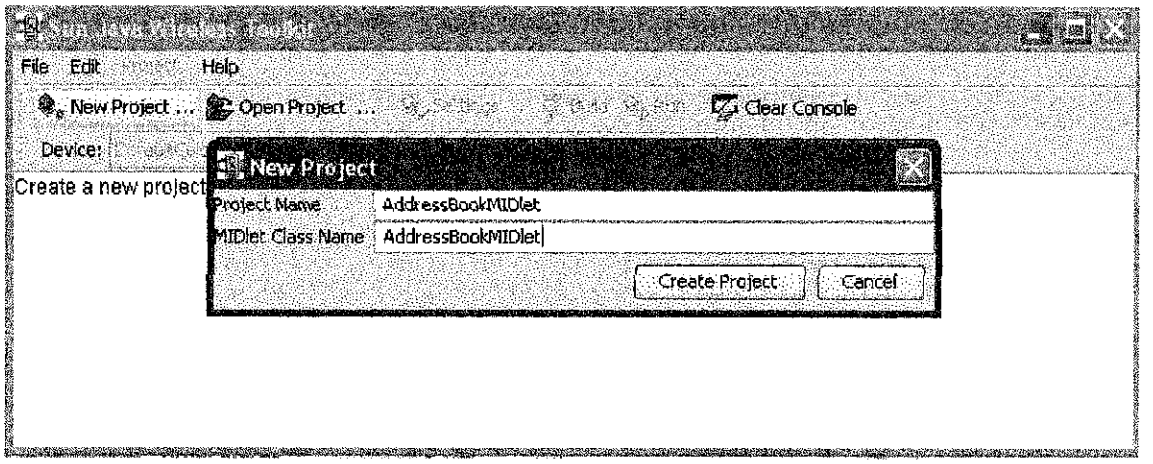


Figure 6: Creating New Project

4. Write the .java file to be implemented. In this case, we already provide the .java file included in the CD. The file's name is **AddressBookMIDlet.java**.

5. Save the AddressBookMIDlet.java file inside the folder of the project which we have just created. The project folder can be found by following the path below:

C:\WTK23\apps\AddressBookMIDlet\src

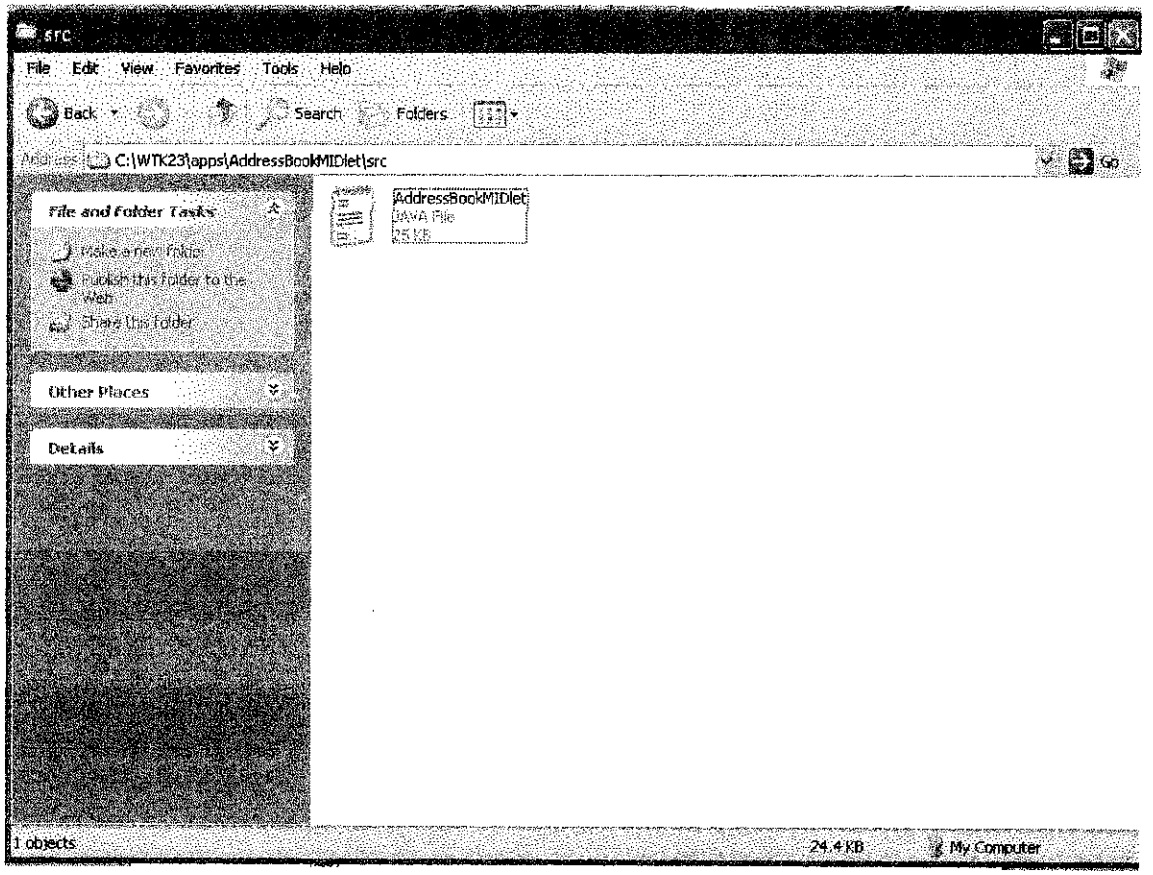


Figure 7: Saving the .java file into the project folder

6. Open the project from the Sun Java Wireless Toolkit platform. For your information, Sun Java Wireless platform can be open by doing step number 1 until 3.

   For now, we are not creating a new project, but opening an existing project. Step 3 above shows how to create a project. So, in order to open an existing project, follow the following figure:

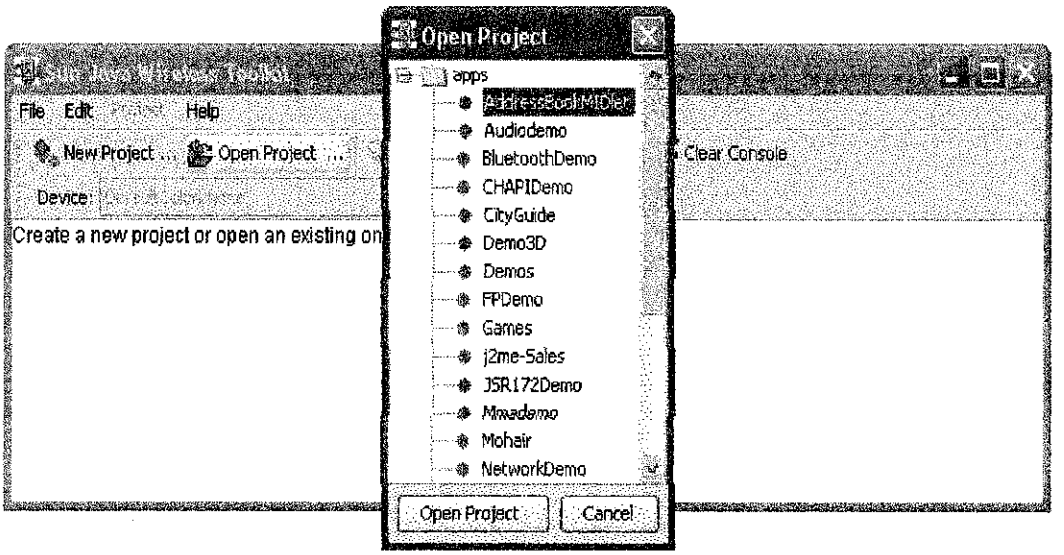


Figure 8: Opening an existing project

7. The following should appear as the project is open successfully.

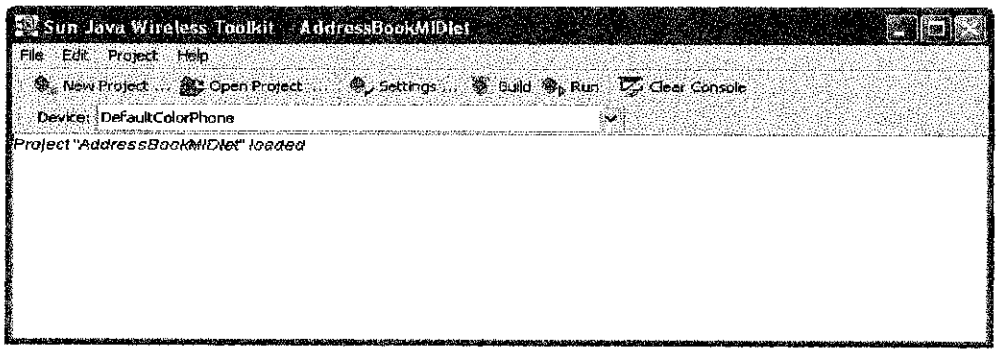


Figure 9: Project loaded

8. Build the project by clicking *Build* button in the menu bar.

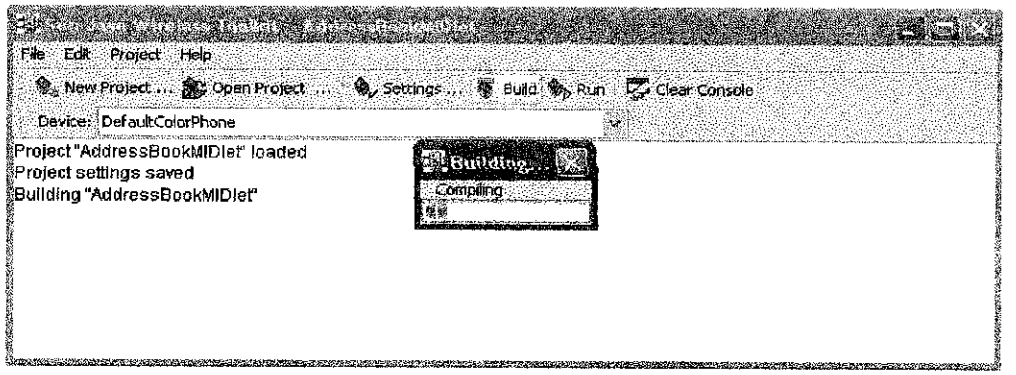


Figure 10: Building the project

9. After the building process has been done smoothly without error, an acknowledgement of completed process should appear as shown below.

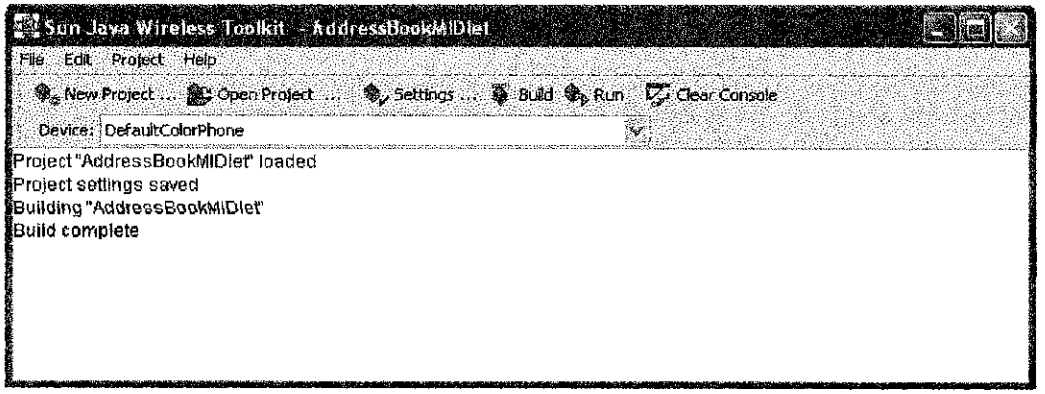


Figure 11: Building project completed

10. Run the project by clicking the *Run* button n the menu bar.

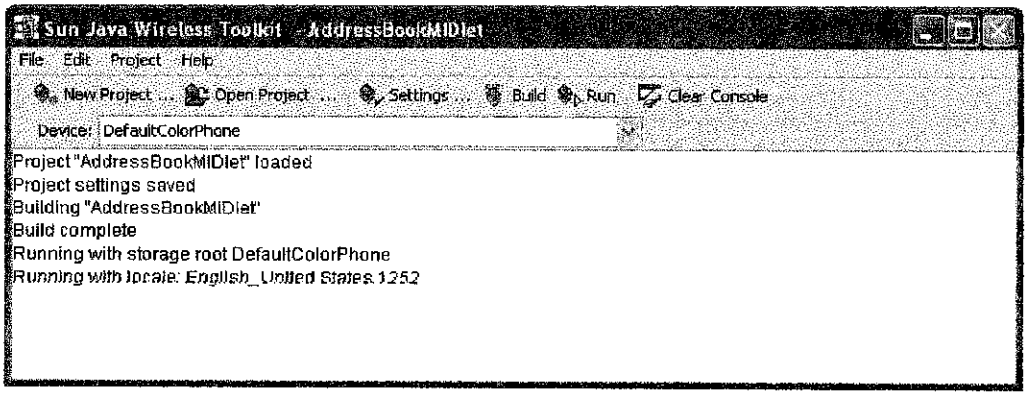


Figure 12: Running the project

11. After all of the process have been done and followed, the emulator of J2ME Wireless Toolkit should appear and the users may interact with it. The interaction between the emulator and the users has been elaborated in the SNAPSHOTS, the detail of navigation system of application.

Application Snapshots

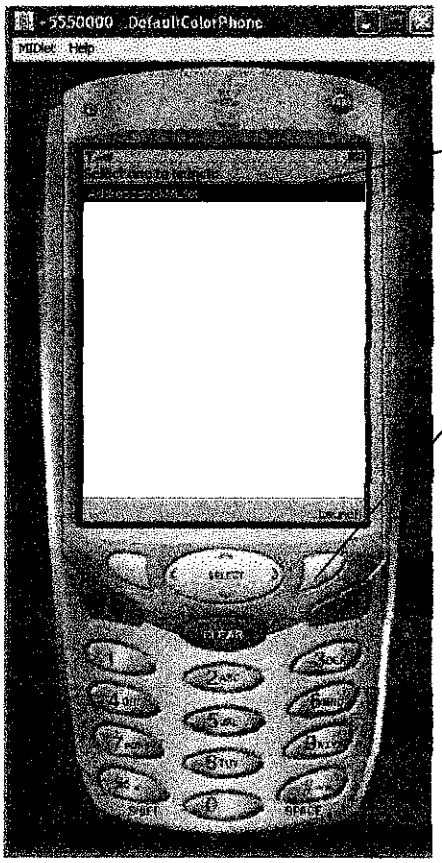


Figure 13: Launch the Application

1. **INITIAL PAGE**

The application launched by selecting the `AddressBookMIDlet`

Click on the `Launch` soft key

The application starts by showing the initial page

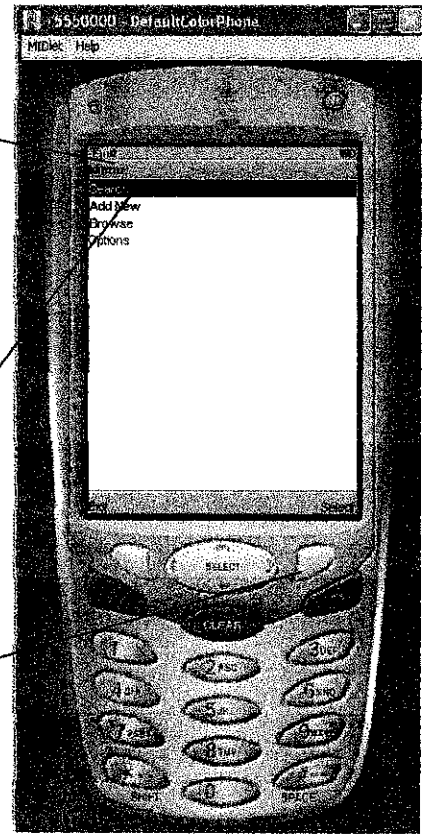


Figure 14: Initial Page

2. **SEARCH**

In The initial page, select the `Search` menu

Click on the `Select` soft key

The `Search` page will appear shortly

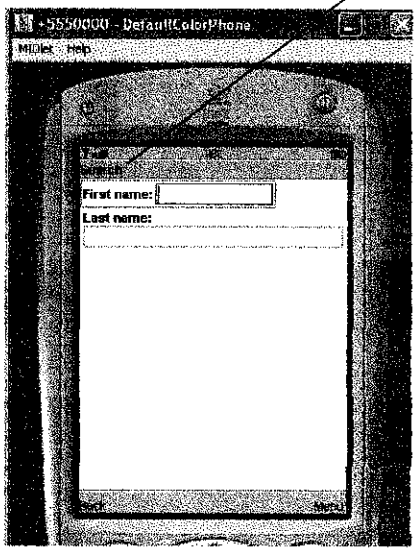


Figure 15: Search

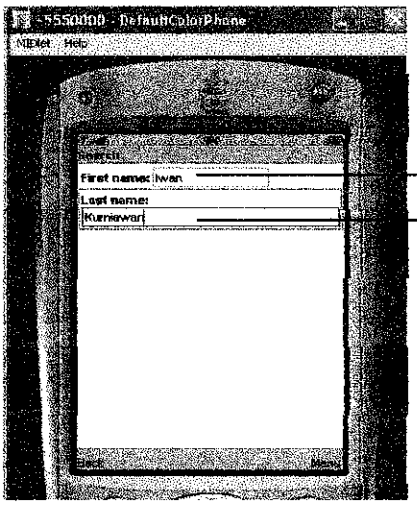


Figure 16: Search Page

3. SEARCH: Local Content

- Key in the `First name` and `Last name`
- Click on the `Menu` soft key
- Choose `Local`
- Click on the `SELECT` button
- The `Search Result` page will appear shortly

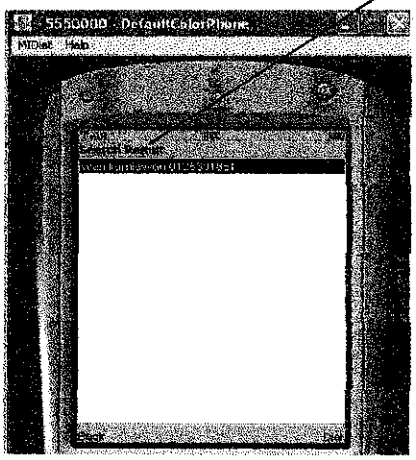


Figure 18: Search Result

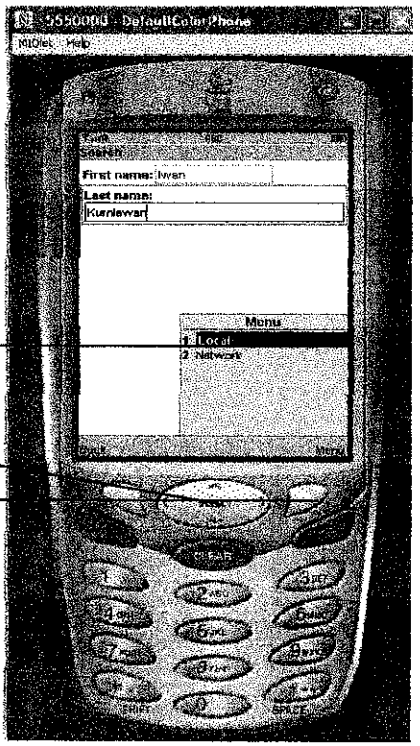


Figure 17: Search Local Content

4. SEARCH: Network

- After Key in the `First name` and `Last name`
- Click on the `Menu` soft key
- Choose `Network`
- Click on the `SELECT` button

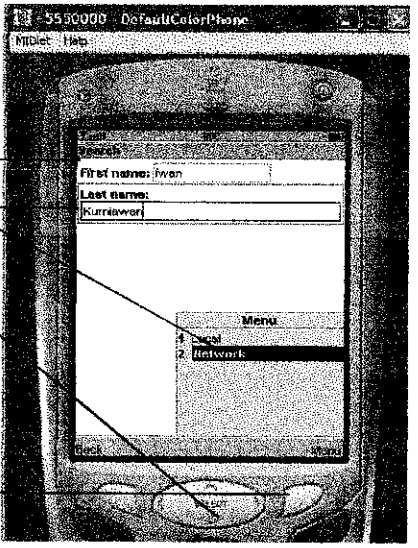


Figure 19: Search Network

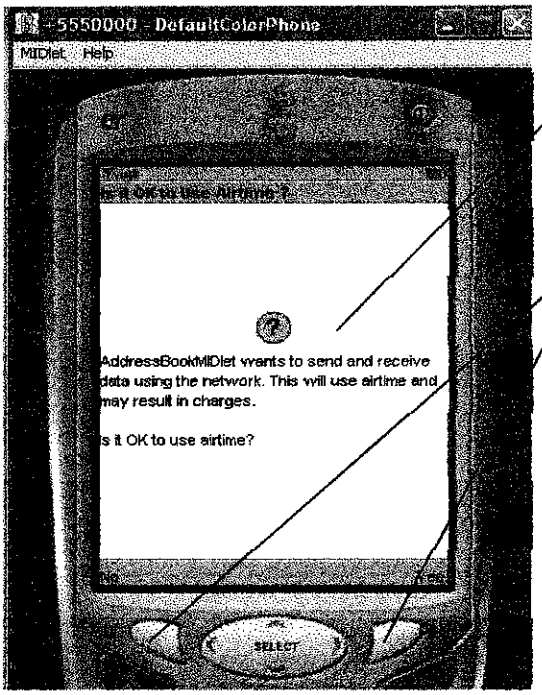


Figure 20: Airtime Confirmation

The confirmation dialog will appear shortly asking you whether to use `airtime` that may result in charges

Click on the `Yes` soft key to proceed

Click on the `No` soft key to cancel

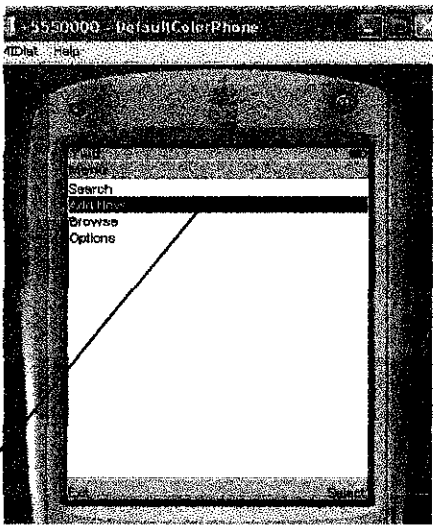


Figure 21: Add New Contact

5. ADD NEW CONTACT

In the initial page select the `Add New`

The `Add New` Contact page will appear shortly

Key in the `First name`, `Last name`, as well as the `Phone Number`

The New Contact will be added to the record

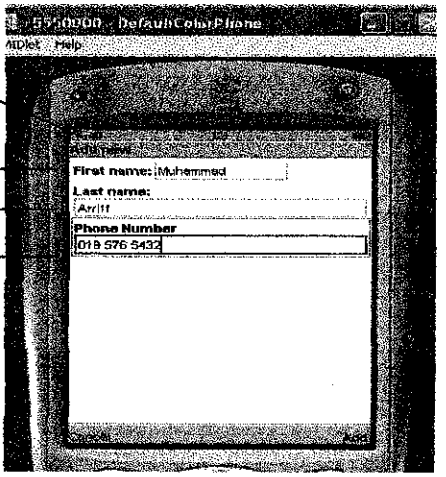


Figure 22: Add New Contact Page

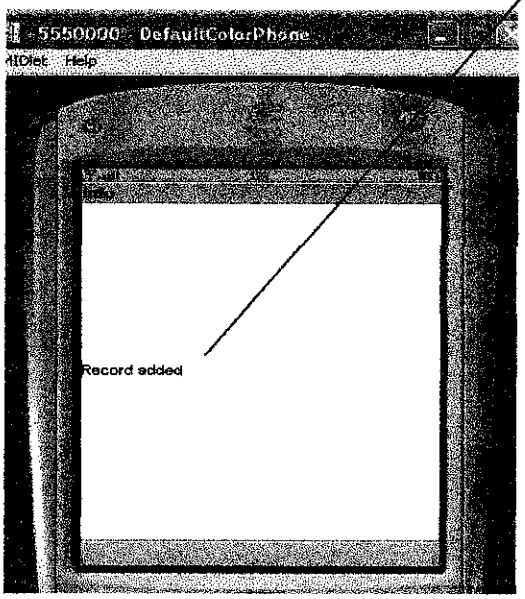


Figure 23: New Contact Added

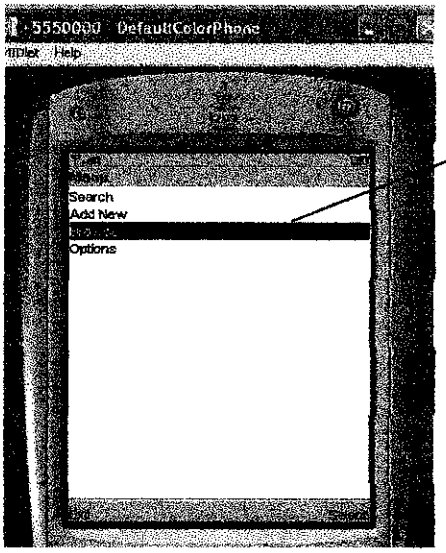


Figure 24: Browse Contact

6. BROWSE CONTACT

In the initial page select `Browse`

The `Browse` page will appear shortly showing the content of the record

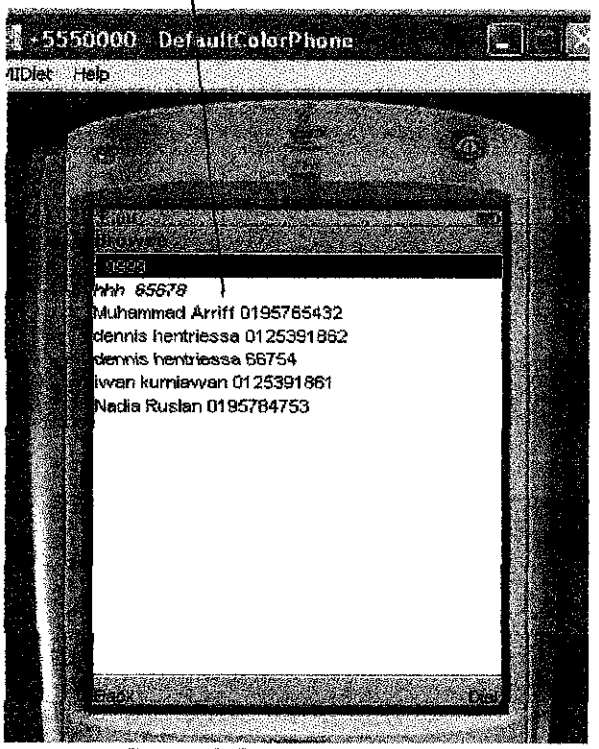


Figure 25: Browse Contact Page

7. OPTIONS

In the initial page select `Options`

The `Options` page will appear shortly giving you the options to sort the content of the record based on their `First name` or `Last name`

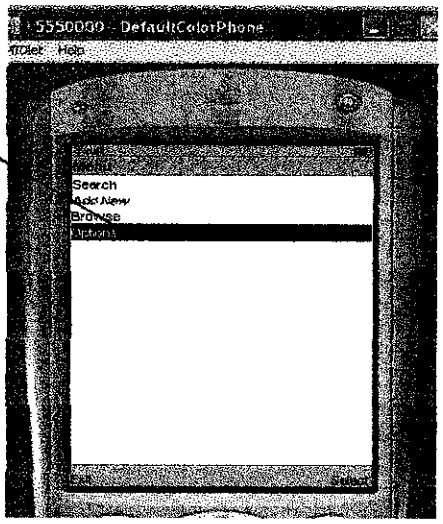


Figure 26: Options

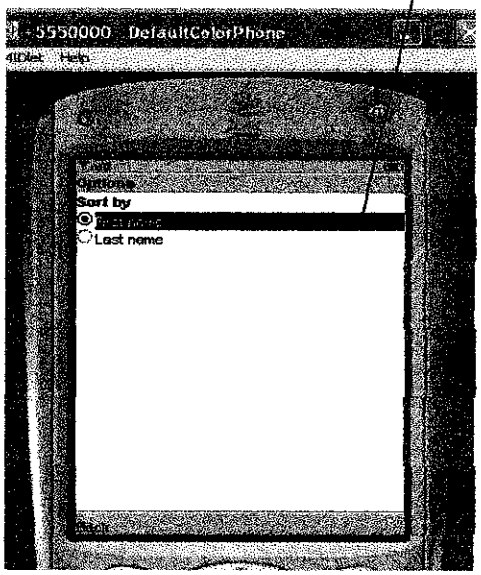


Figure 27: Options Page

Docs & Support

J2ME MIDP Development for NetBeans IDE 4.1 Quick Start Guide

This document takes you through the basics of using NetBeans IDE 4.1 to create a Java™ 2 Platform, Micro Edition (J2ME™ platform), Mobile Information Device Profile (MIDP) application. This document is designed to get you going as quickly as possible. More information about using the J2ME support in the NetBeans development environment is provided in the following documents:

- **Importing Existing J2ME MIDP Source Code into the NetBeans IDE**

The following short tutorial takes you through some of the basic steps of working with the new project system. We will show two v a J2ME MIDP project named "MyHello" that displays the text "Hello World" in a device emulator.

Requirements

You must have NetBeans IDE 4.1 and the NetBeans Mobility Pack 4.1 installed before you can start J2ME MIDP development. See **MIDP Development Download page** for instructions on downloading and installing the complete environment.

Creating a MIDP Application

The NetBeans IDE provides a wizard that enables you to quickly create a MIDP project. When creating the project, you can choose your application in the Visual Mobile Designer or in the Source Code Editor.

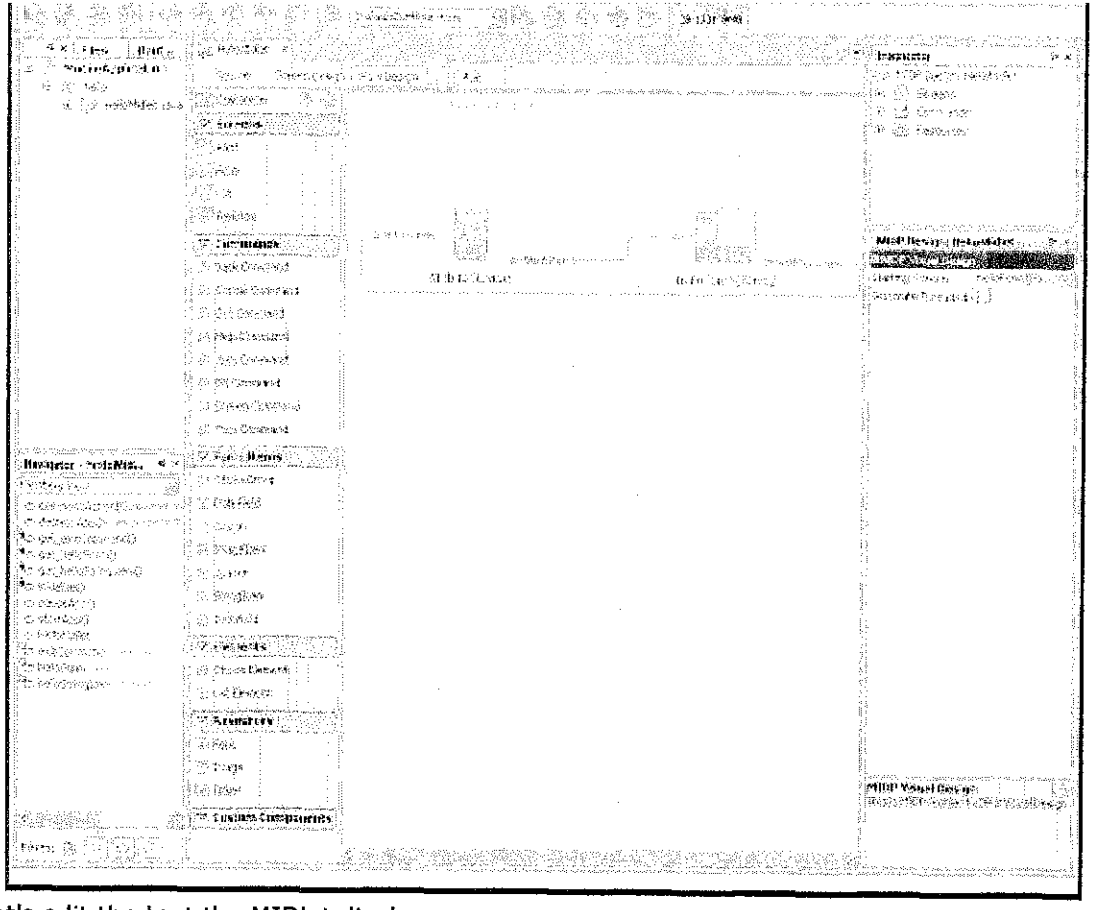
Using the Visual Mobile Designe gives you the ability to graphically plan out the flow of the application and design the screens the use. The designer automatically creates the code for the application. One limitation of this approach, however, is that if you edit tl code in the Source Editor, you cannot continue to edit the code in the Visual Designer. You also need to use callback methods to c preprocessor blocks for your project configurations.

Using the Source Code Editor, you manually create the code for your MIDlets. Creating code in the Source Code Editor gives you r when editing the code, and enables you to create preprocessor code blocks.

Creating a MIDP Application Using the Visual Mobile Designer

project

1. Choose File > New Project (Ctrl-Shift-N). Under Categories, select Mobile. Under Projects, select Mobile Application and click Next.
2. Enter `MyHello` in the Project Name field. Change the Project Location to any directory on your system. From now on, we will refer to this directory as `$PROJECTHOME`.
3. Check the Set as Main Project and Create Hello MIDlet checkboxes (both are checked by default). Click Next.
4. Leave the J2ME Wireless Toolkit as the selected Target Platform.
5. Click Finish. The IDE creates the `$PROJECTHOME./MyHello` project folder. The project folder contains all of your sources and project metadata, such as the project Ant script.
   The application itself is displayed in the Flow Design window of the Visual Mobile Designer.



Editing the Java Source Code

Now let's edit the text the MIDlet displays.

1. Click on Screen Design.
   This opens the Screen Designer window, and displays the `helloTextBox[Textbox]` screen, which is the only screen available in the application.
2. Click on the helloTextBox screen and type in some text. The default text is, "Hello, World!"

Compiling and Running a Project

1. Choose Run > Run Main Project (F6) from the Run menu. Double-click the Output window to maximize it so you can see all the output. Note that the `HelloMIDlet.java` file is built before it is executed. A device emulator opens to display the results of the executed MIDlet. The default device emulator is

launches the MIDlet and displays the text you entered in the source code.
3. Click on the button below Exit to close the MIDlet. Then click on the button in the upper right corner of the device to close the emulator window.

Creating a MIDP Application Using the Source Editor

Creating a new J2ME MIDP project

1. Choose File > New Project (Ctrl-Shift-N). Under Categories, select Mobile. Under Projects, select Mobile Applica Next.
2. Enter `MyHelloMIDlet` in the Project Name field. Change the Project Location to any directory on your system. F we will refer to this directory as `$PROJECTHOME`.
3. Check the Set as Main Project checkbox and remove the check from the Create Hello MIDlet checkbox. Click Ne
4. Leave the J2ME Wireless Toolkit as the selected Target Platform.
5. Click Finish. The IDE creates the `$PROJECTHOME./MyHelloMIDlet` project folder. The project folder contains all sources and project metadata, such as the project Ant script.
6. Right-click the `MyHelloMIDlet` node in the Explorer window and choose New > File/Folder.
7. Under Categories, choose MIDP. Under File Types, choose MIDlet. Click Next.
8. Enter `HelloMIDlet` as the MIDlet name. Click Finish. The `HelloMIDlet.java` is created.
9. Double click the HelloMIDlet.java to display the source code in the Editor.
10. Click in the Source Editor and change

```
public class HelloMidlet extends MIDlet
```

to

```
public class HelloMIDlet
             extends MIDlet implements javax.microedition.lcdui.CommandListener
             {
```

11. Add the following text before the `startApp()` method:

```
public HelloMIDlet() {
}
private void initialize() {
  javax.microedition.lcdui.Display.getDisplay(this).setCurrent(get_helloTextBox());
}
```

```
        if (displayable == helloTextBox) {
            if (command == exitCommand) {
                javax.microedition.lcdui.Display.getDisplay(this).setCurrent(null);
                destroyApp(true);
                notifyDestroyed();
            }
        }

    }

    private javax.microedition.lcdui.TextBox get_helloTextBox() {
        if (helloTextBox == null) {
            helloTextBox = new javax.microedition.lcdui.TextBox(null, "Test String",120, 0x0);
            helloTextBox.addCommand(get_exitCommand());
            helloTextBox.setCommandListener(this);
        }
        return helloTextBox;
    }

    private javax.microedition.lcdui.Command get_exitCommand() {
        if (exitCommand == null) {
            exitCommand = new javax.microedition.lcdui.Command("Exit", javax.microedition.lcdui.Cc
      1);
        }
        return exitCommand;
    }

    javax.microedition.lcdui.TextBox helloTextBox;
    javax.microedition.lcdui.Command exitCommand;
```

12. Add a line `initialize();` to the `startApp()` method, so it looks like the following:

```
    public void startApp() {
     initialize();
      }
```

Editing the Java Source Code

Now let's add some text for our MIDlet to display.

1. In the `startApp()` method, replace the "test string" code with the text of your choice. For example, "Hello Wor

Compiling

1. Choose Run > Run Main Project (F6) from the Run menu. Double-click the Output window to maximize it so yo

a Project

2. In the device emulator window, click on the button below the Launch command. The device emulator launches displays the text you entered in the source code.
3. Click on the button below Exit to close the MIDlet. Then click on the button in the upper right corner of the devi emulator window.

Changing the Emulator Platform

Changing the Default Emulator Device

You can create different project configurations to test your MIDlet on different emulator platforms.

1. Right-click the MyHello Project node and choose Properties. In the Properties dialog, choose the Platform node. You can change the device for the default configuration.
2. Click the Device dropdown menu and choose QwertyDevice. Click OK.
3. Run the application again, and the applicaton runs in the `QwertyDevice` emulator.

Adding a new Emulator Platform

1. Choose Tools > Java Platform Manager from the main toolbar.
2. In the Java Platform Editor dialog, click the Add Platform button.
3. In the Choose Platform Folder page of the Add Platform wizard, Use the Browse button to navigate to the parent directory of the emulator platform you want to install. For example, `c:\WTK22`. Click Next.
4. In the Configure the Platform Page, the IDE detects and configures the platform and displays information about the platform. Click finish to complete the configuration.
5. Choose File > "MyHello" Properties.
6. Choose the Platform node, then choose the name of the new emulator (for example, J2ME Wireless Toolkit 2.2) from the Emulator Platform dropdown menu. Click OK.

Advanced: Using Configurations

Adding a Configuration

You can use configurations to create more than one set of distribution JAR and Java Application Descriptor project. This enables you to create one set of source code and customize it for each mobile device you're

1. Right-click the MyHello Project node and choose Properties. In the Properties dialog, Choose Add Cc Project Configuration combo box. This opens the Add Configuration dialog.
2. Enter a name for your configuration in the Configuration Name field. The name should identify the t you will deploy the JAR/JAD created for this configuration. For this example, enter `BlackWhiteDisp` have just created a new configuration
3. Configurations can also be added by clicking the Manage Configurations button in the Properties dia Project Configuration Manager dialog which contains options for adding, removing, or duplicating co

Customizing a Configuration

You can add as many configurations as you would like to your project. You can then modify settings in the dialog for each configuration that you've added.

1. Right-click the MyHello Project node and choose Properties. In the Properties dialog, choose the Pla the J2ME MIDP Configuration dialog.

by the Default configuration for this panel. Uncheck the Use Default Project Configuration Values op panel. All components on this panel are now editable.
4. Change the Device option to DefaultGrayPhone.
5. Toggle the Project Configuration dialog at the top of the Properties dialog and observe that the Dev on which configuration is currently selected. This method of configuration customization works for e the general panel) in the dialog. Click OK to save your configuration changes and exit the Properties

Creating an Ability

When you create an ability, you associate it with one or more project configurations that support a specifi attributes. You can then associate the ability with one or more code blocks that support the attribute. If y remove configurations, you won't have to go through all your code and associate the configuration with ea you need only to associate the configuration with an ability. The configuration is then automatically associ that is associated with the ability.

1. Right-click on the MyHello project node and choose Properties.
2. Choose the Abilities page from the Category menu tree.
3. Choose a configuration from the Project Configuration dropdown menu. This is the configuration wit be associated.
4. If necessary, uncheck the Use Values from "DefaultConfiguration" checkbox.
5. Click the Add Ability button.
   This opens the Add Ability dialog.
6. Click the New Ability button.
7. In the New Ability dialog, enter a name for the ability.
   Click Ok to close the New Ability dialog. The ability is now associated with the selected project confi

You can associate the ability with other configurations by choosing a different configuration from the Proje and clicking the Add Ability button.

Adding Configuration-Specific Code (Preprocessor blocks)

Preprocessor blocks enable you to create, manage and track code that is specific to one or more project c The code is enabled (or disabled) only for the configurations or abilities you identify.

1. The editor window for `HelloMIDlet.java` has a combo-box in the upper right corner which reads "I default. Change this to your newly created configuration, BlackWhiteDisplay.
2. Right-click on the source code line where TextBox is instantiated. Right-click on the line and choose Create If/Else Block. In the first, gray block, append the word "Gray" to the "Hello World" string. In add "Color" to the "Hello World" string.
3. Toggle the configuration combo box in the upper right corner of the editor and notice that blocks ar based on which configuration is selected. The first block will be commented in whenever the selecte the value in the header and footer of the block. The second block will be commented in whenever tl does **not** match the value in the header and footer. This combo box also controls which properties f Properties are used when taking action on the project.

Running

Configurations can be built and run individually or simultaneously.

and then a choose a different configuration. This can also be done with the configuration combo bo
the project again.

3. Two emulators appear, one color and one gray. The gray emulator displays "Hello World Gray" and displays "Hello World Color"

Creating a Client-Server Application

Using the Wireless Connection Wizard

The J2ME Wireless Connection Wizard enables you to generate the connecting classes for a mobile client/server application without hand-coding or using libraries.

To use the J2ME Wireless Connection Wizard, you need to have a mobile project to contain the generated MIDP code, and an opened web project containing at least one J2SE service class to be exported.

To create the client/server connection:

1. Open the Wireless Connection Wizard by selecting a Mobile Project from the Projects view and choosing File > New File, then choosing MIDP > Wireless Connection Wizard.
2. Select a name, web project and package for the generated server code.
3. Identify the services to be exported from the server to the client application.
4. Select a name and package for the generated client class.
5. Determine the types of code to be generated:

- Stub methods for each exported service
- Code enabling multiple calls per session
- Tracing code
- A sample MIDlet you can examine and modify

The wizard creates:

- a J2ME client class
- a servlet and supporting classes
- a mapping file in xml format.
- if you choose, a MIDlet you can examine and modify.

The J2ME client , mapping file and the MIDlet are generated under the J2ME project. The servlet and supporting classes are generated under the web project. The servlet is also added to the web application (`web.xml`).

generate the the original application, or when you open the Settings file in the Editor window.

Localizing Your Application

Adding Localization Support

1. Create the java classes that initialize the localization support and message property files.
    1. Right-click the MyHello Project node and choose New > File/Folder.
    2. Under Categories, select MIDP. Under File Types, select Localization Support Class. Click Next.
    3. Accept the default values by clicking Finish. This creates the files `LocalizationSupport.java` and `messages.properties`.
2. Change the Internationalization String and Search formats.
    1. Choose Tools > Options and click the Classic View button.
    2. Select the Internationalization node in the Options menu tree.
    3. In the properties window:
        - Change the value of the II8N String Format property to `LocalizationSupport.getMessage("{key}")`
        - Change the value of the II8N Search Format property to `(getString|getBundle|getMessage)[:space:]*\([:space:]*{hardString}|//NOI18N`
3. Add localization keys and their values.
    1. Choose Tools > Internationalization > Internationalization Wizard. This opens the Internationalization wizard.
    2. Select the `HelloMIDlet` MIDlet as the source file to internationalize. Click Next.
    3. Select the `Messages` resource file as the file in which to store the values for the internationalized strings. Click Next.
    4. Modify the Value fields for each key identified for the `HelloMIDlet` MIDlet.
    5. After all the strings have been added to the `messages.properties` file, do the following to add values for each new locale:
        1. Right-click on the file node in the Projects view and choose Add Locale.
        2. Choose a locale you want to support from the Predefined Locales list box, or use the combo boxes at the top of the form to define a new locale.
        3. Expand the `message.properties` node in the Projects view, and double-click on the newly-added locale.
        4. Translate all properties into the appropriate language.
4. Add code to initialize localization.
    1. Enter the following string in your MIDlet wherever you want a key localized:
       `LocalizationSupport.getMessage("Key");`
       For example, `LocalizationSupport.getMessage("LAB_NextButton");`