Integrating BlackBerry Wireless Devices into Computer Programming and Literacy Courses

Qusay H. Mahmoud and Allan Dyer
Department of Computing and Information Science
University of Guelph, Guelph, ON, N1G 2W1 Canada
{qmahmoud,adyer}@uoguelph.ca

ABSTRACT

In this paper we describe our experience in integrating the RIM's BlackBerry handheld wireless device into programming and literacy courses at the University of Guelph and the University of Guelph-Humber. The courses are lab-intensive where students experiment with the devices, and develop and deploy applications for them. We believe that teaching computer programming in the context of simple wireless mobile applications provides a motivating framework for students and inspires them to work hard due to the practical experience they get that allows them to program their own cellular phones. In addition, many of our students who spend their co-operative work terms or internships at RIM value this experience very much as it offers them an advantage over students from other institutions. More importantly, students learn about the programming models for developing applications for wireless devices and appreciate the unique opportunities such devices offer, but also become aware of the development challenges they present.

Categories and Subject Descriptors

D.3.3 [**Programming Languages**]: Language Constructs and Features – *abstract data types, control structures.*

General Terms

Design, Experimentation, Management, Human Factors, Languages.

Keywords

Programming for Fun, BlackBerry, PDA, Java ME, Computer Literacy.

1. INTRODUCTION

Wireless technology is becoming an integral part of our everyday lives. Most students have a cellular phone or a personal digital assistant (PDA) used for voice communications, checking emails, or organizing personal information. This is the technology of the 21st century, which has incredible opportunities and various applications. One such application is mobile learning or m-learning [1,8, 10], which is the marriage of mobile devices (such as RIM's BlackBerry) and e-learning. A key benefit of m-learning is its potential for increasing productivity by making learning available anytime, anywhere.

The BlackBerry device is a general purpose communicating device that presents a great deal of opportunities for educators in computer science as well as other fields to be integrated in the curriculum. Students in introductory programming courses usually develop and test their applications or programming assignments on a platform similar to the one on which it will be tested by the instructor or teaching assistant. But this is not the case for wireless mobile applications which are developed on one platform (such as Microsoft Windows) and deployed and run on a totally different platform (such as a cellular phone or PDA). Thus, there exists an opportunity to introduce students to different programming models, but more importantly we believe such experience will be as practical as possible since it will inspire students to be excited about learning as it will enable them to eventually program their own Java-enabled wireless devices.

Over the past 15 – 20 years the basic applications for computer literacy courses have been word processing, spreadsheets, presentation software, and database management on desktop computers. Recently, these courses may include the Internet and Web development to be somewhat current. However, with the wide spread use of iPods, cell phones with multimedia applications, digital cameras and pocket personal computers, there needs to be innovation in the area of computer literacy if it has any relevance in the university programs simply because there is a huge gap between academic environments and how student use their computers

The University of Guelph and the University of Guelph-Humber, which is an initiative between the University of Guelph and Humber Institute of Technology and Advanced Learning, are introducing BlackBerry devices in their computer literacy and programming courses.

The rest of this paper is organized as follows. Section 2 provides a brief overview of the Java Platform, Micro Edition (Java ME) [9]. The BlackBerry wireless device and its network architecture are presented in Section 3. Section 4 discusses the computer literacy course in which we have integrated the Blackberry devices. The programming courses are discussed in Section 5. The programming models are discussed in Section 6. Section 7 presents the laboratory environment, assignments, and a sample project. Conclusions and future work are discussed in Section 8.

ACMSE'07, MARCH 23–24, 2007, WINSTON-SALEM, NC, USA.

2. JAVA PLATFORM, MICRO EDITION

The Java Platform, Micro Edition (Java ME) [9] provides a complete set of solutions for creating state-of-the-art networked applications for consumer and embedded devices, the small but increasingly smart gadgets whose market is growing explosively. Java ME enables device manufacturers, service providers, and application developers to deploy compelling applications and services to their customers in two broad categories: (1) personal, mobile, connected, information devices, such as cell phones, two-way pagers, and organizers; and (2) shared, fixed, connected, information devices, such as set-top boxes, Internet TVs, and in-car entertainment navigation systems.

To support this disparate collection of product groups, Java ME includes a variety of API packages called configurations, profiles, and optional packages, organized as shown in Figure 1.

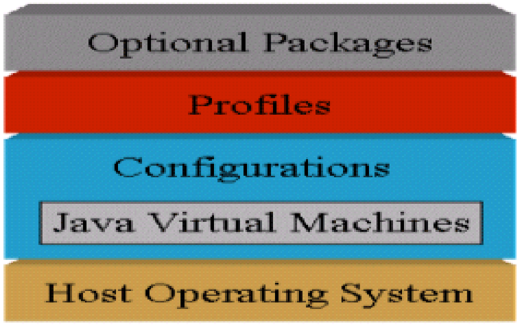


Figure 1. High-level architecture of Java ME.

At the bottom of the Java ME software stack is the configuration, either the Connected Device Configuration (CDC) or a subset of it called the Connected Limited Device Configuration (CLDC) that provide basic application management. At the heart of each is a device-specific implementation of the JVM, which runs on top of the device's operating system. CDC and CLDC support high-end and low-end devices, respectively. The distinction between high-end and low-end is fuzzy, however, and continue to get fuzzier as today's smaller and less expensive devices feature more battery power, more storage, and better user interfaces. In our programming courses, we focus on the CLDC as it targets devices such as cellular phones and the BlackBerry wireless device.

A profile tailors the software stack, collecting together a broad set of capabilities commonly needed by applications used on a collection of similar devices. For example, the Mobile Information Devices Profile (MIDP) supports interactive applications by providing a rich set of APIs for user interfaces and other common operations. Applications developed based on MIDP are called MIDlets.

Optional packages are collections of APIs that developers may choose to include in the software stack to support specific kinds of functionality, such as messaging or multimedia. In our courses we didn't need any of the optional packages.

3. THE BLACKBERRY DEVICE

The BlackBerry device, which is developed by Research In Motion (RIM) [7], is a handheld wireless device whose major selling feature to date has been instant, secure, mobile access to email. BlackBerry devices are offered by Nextel, Telus, T-Mobile, Rogers, and many other wireless carriers worldwide. Once a device is on a carrier's network, it is linked to RIM's Network Operating Center (NOC), which has direct connections to all RIM's carrier partners and to BlackBerry Enterprise Servers (BES) deployed all over the world. The BES software is a middleware that links handheld devices to corporate email services such as Microsoft Exchange and Lotus Notes, as depicted in Figure 2.

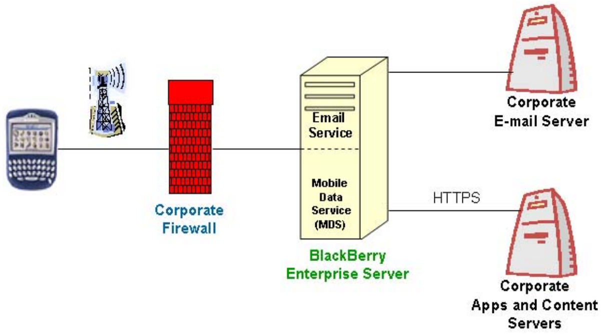

Email Service
Mobile Data Service (MDS)
Corporate E-mail Server
HTTPS
Corporate Firewall
BlackBerry Enterprise Server
Corporate Apps and Content Servers

Figure 2: BlackBerry architecture with gateways.

Installed behind the corporate firewall, a BES can be configured with additional services. The Mobile Data Service (MDS), for instance, gives devices access to servers in the corporate intranet that wouldn't be accessible from public networks otherwise. For our purpose, we would like students to experiment with network applications (such as a real-time stock quote application) deployed on the BlackBerry. Since the devices we have are not Wi-Fi enabled and thus cannot be used freely on our Wi-Fi, we have made arrangements with a wireless carrier for data service subscription.

While some BlackBerry devices are based on C++, many new ones support Java ME primarily because Java technology makes developing applications so much easier. Its platform-independence eliminates many porting woes and its automatic garbage collector lets developers concentrate on application logic rather than memory management [2].

RIM's support for Java ME includes development of its own Java Virtual Machine (JVM), which supports the Connected Limited Device Configuration (CLDC), and the Mobile Information Device Profile (MIDP). BlackBerry devices also come with additional BlackBerry specific APIs, however, that enable developers to create applications that have the Blackberry-native look and feel, and are more sophisticated than standard MIDlets developed using MIDP. The RIM's specific APIs are, however, more complicated than the CLDC/MIDP APIs, and therefore we have focused on the CLDC/MIDP standard APIs in our programming courses.

4. COMPUTER LITERACY COURSES

Over the past 15-20 years, the basic applications for computer literacy courses have been word processing, spreadsheets, presentation software, and database management on desktop computers. Recently, these courses may include the Internet and Web development to be somewhat current. There needs to be innovation in the area of computer literacy if it has any relevance in the university programs. Most student activities focus on the social side of computing – instant messaging, e-mail, music, video, Internet and games rather than the basic applications. More and more students are buying wireless laptop computers and most academic computing labs are cabled desktop computers. There is a huge gap between academic environments and how student use their computers. Hence, there is a need to integrate small mobile computing devices into computer literacy courses while maintaining some of mainstream traditional applications

4.1 Computing for Non-CS Students

The University of Guelph has successfully introduced BlackBerry devices in its computer literacy course "Introduction to Computing". The purpose is to provide students with a small mobile device that will address many of their current social uses. Most students in our literacy programs have not had exposure to these devices. It permits them to experiment with the applications and decide for themselves if this is a tool in their future. It moves away from the standard computing device and on to a smaller platform that may indicate a future trend.

Introduction to Computing is a first year introduction to computing for non-computing students. Its goal is to familiarize the non-computing students with those software tools that they will encounter in their personal and professional lives. Mobile technologies are now so ubiquitous that a well-informed citizen of the 21st century must have a working knowledge of the technology and its impact on society. The BlackBerry devices are used in a structured lab setting to introduce the students to a professional tool; students get to experiment with the devices and its applications such as email and web access, and learn how to use them effectively as described in this lab (http://cis1200.cis.uoguelph.ca/blackberry.htm) to manage electronic information. The labs are designed to challenge the students' appreciation for the importance of mobile communications to the business environment. This changes the face of computing as it attempts to move them beyond their comfort zone as most of application students don't think that far ahead with their interaction with computers.

5. PROGRAMMING COURSES

We have integrated BlackBerry devices into programming courses offered in the first and third semesters of study for students in our Applied Computing program at the University of Guelph-Humber. All of our courses are designed to be lab-intensive in which students learn by doing. The theory students learn in the classroom is reinforced through lab work and projects of a substantial size.

5.1 Computer Programming I

This is the first programming course that aims to provide a step-by-step approach to programming [3]. No previous experience with computers in general, or programming in particular is required. The course is 13 weeks long with 3-hour lectures per week plus a total of ten, 3-hour, labs for the semester. This course is revolutionary as it is meant to provide an alternate introduction to CS1 by focusing on using computers to build applications for fun [4]; students learn how to build static homepages, interactive homepages, simple JavaScript applications, and simple stand-alone Java applications. In other words, the course focuses on using computers for 'programming for fun' rather than 'frustration'. This is accomplished by integrating HTML, JavaScript, and Java in a one-semester course – each topic prepares the students to the next harder topic. This is coupled with the use of Pair Programming in order to promote knowledge sharing between students [5, 6].

We have integrated the BlackBerry devices in this course by dedicating two weeks of lectures and 6-hrs of labs, towards the end of the semester, for wireless application development. Students are introduced to the Java ME platform, and the BlackBerry device and its services. Students learn about the life-cycle of developing applications for such devices. In the first lab students learn how to use the device itself and how to download and deploy existing Java ME applications (or MIDlets) onto the device using a USB cable. In the second lab, students learn how to develop and test their own MIDlets using Sun's Java Wireless Toolkit and its emulation environment, convert the MIDlets into a file format suitable for the BlackBerry device, and then deploy them onto the device. In part of the final assignment, students are asked to develop a mortgage calculator for the BlackBerry device.

5.2 User-Centered Programming

The User-Centered Programming course is a technically intensive introduction to Graphical User interface (GUI) programming using Java AWT/Swing. Highly complex user-centered concepts, such as interactive programming, threads, graphics, and event models are explored. The objective of this course along with another course on data-centered programming, which are taken concurrently in the third semester of study, is to produce software developers capable of designing and developing fully functional interactive applications that communicate with back-end databases.

We have integrated the use of BlackBerry devices in the user-centered course as follows: we spend three weeks on introducing the programming model and the development lifecycle of applications for the BlackBerry and other Java-based devices. Students learn about the nature of the applications for small devices (e.g. no overlapping windows), the Java ME model for event handling, and the HTTP-based network programming model of Java ME using the HTTP connection APIs. In the labs, students learn how to develop and deploy applications onto the BlackBerry. In the final assignment, students are asked to develop a quiz application for the BlackBerry, as well as a networked application that enables students to retrieve their course marks from a remote database using the HTTP connection APIs.

5.3 Practicing Pair Programming

In order to enable students to share knowledge and learn about tips and tricks that other students might be familiar with, Pair Programming [5, 6] was used for both, labs and assignments in the introductory programming course, and for labs-only in the user-centered programming course. In the pairing process, we didn't want to allow students to pair themselves up fearing that friends, and possibly strong programmers, pair together. We wanted to have a fair pairing process, which was done by pairing students randomly and rotating pairs for all labs and assignments.

6. PROGRAMMING MODELS

The BlackBerry device supports two application models: (1) Browser-based model that allows developers to focus on developing back-end content in a standard markup language, such as the Wireless Markup Language (WML). Using existing browsers' client capabilities free the developer from worrying about the client interface – but it does limit client functionality to what the browser provides, and there is no support for offline browsing; and (2) Custom Java applications enable developers to develop customized user interfaces and navigation, and support content beyond text and images. Developers can also build applications that users can download and install on wireless devices, so they can continue to use offline while out of wireless coverage. The two programming models are introduced to students in the programming courses, but the focus is on the Java ME platform.

6.1 Application Deployment

Students in programming courses usually build, test, and evaluate an application on a platform similar to the one on which it will be deployed and run. Development of wireless applications is more challenging because they typically are developed on one platform (such as Microsoft Windows) and deployed on a totally different platform (such as a BlackBerry device). One consequence is that while emulators and simulators enable developers to do some of their testing on the development platform, ultimately they must test and evaluate the application in the very different environment of a live wireless network. In order to demonstrate the challenges of this to students, we follow a development life-cycle that comprises the following three steps:

1. Write the application. We ask students to use Sun Java Wireless Toolkit (WTK) or RIM's Java Development Environment to develop their applications.
2. Test the application in an emulation or simulation environment. Once the application compiles nicely, we ask students to test it in a WTK emulator or BlackBerry simulator. Deploying and running applications in such environments is a very good way to test the logic and flow of an application. An application performance may be stunning in an emulation environment, which has all the processing power and memory of the desktop machine at its command, but will it perform well on the handheld device with its limited memory and processing power, and low bandwidth? Students appreciate being exposed to such technology.
3. Download the application to a physical device and test it. Once the students are happy with the application's performance on the emulator or simulator, we ask them to download the application to a real device and test it there. Students experiment with downloading applications using a USB cable. Students are able to test networked applications in a live wireless environment since the devices we use have wireless data subscription with a wireless carrier. This is especially useful for testing networked applications such as the Stock Portfolio application that interacts with Yahoo's finance server to retrieve stock quotes, to ensure that its performance is acceptable.

6.2 Software Tools and Devices

In addition to the Java Platform, Standard Edition (Java SE) which is the standard Java development environment available from Sun Microsystems, the laboratory environment we used to introduce students to wireless application development included the following tools which are available free of charge from Sun Microsystems and RIM:

- Sun Java Wireless Toolkit 2.5 for CLDC (http://java.sun.com/products/sjwtoolkit).
- The BlackBerry Simulator (http://www.blackberry.com/developers/downloads).
- The BlackBerry Java Development Environment (JDE) available at the same URL above as the simulator.

Sun's Java Wireless Toolkit provides an excellent environment for getting started with Java ME development, and RIM's BlackBerry Simulator provides an excellent tool for simulating the functionality of the BlackBerry device as shown in Figure 3. However, we believe that allowing students to experiment with the physical devices provides a totally different kind of experience. We provided the students with active Blackberry devices to experiment with during the labs.

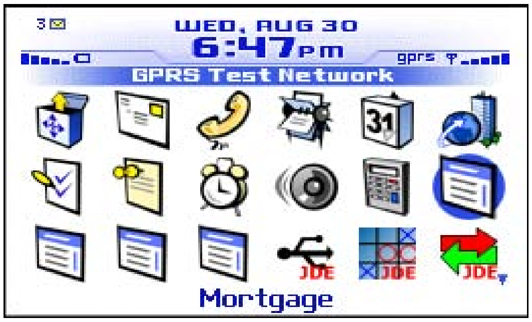


Figure 3. The BlackBerry simulator.

7. LABS, ASSIGNMENTS, AND PROJECTS

The labs are an important component of our courses. In each of the programming courses discussed above, the labs are worth 10 - 15% of the final mark. In each course, we have a total of ten 3-hour labs where students do experiments and exercises related to the topics covered in the classroom during that week.

In the first lab, students learn how to use the BlackBerry Simulator, and deploy existing MIDlets onto the simulator and experiment with the deployed applications. Students learn how to run existing MIDlets on the physical BlackBerry device, and this enables them to use some of RIM's JDE tools to accomplish the job. To run a standard MIDlet on a Java-enabled BlackBerry device, students first need to convert the ".jad" and ".jar" files to the ".cod" format using the rapc command-line tool that comes with RIM's JDE. As an example, the following command converts a MIDlet called LoginMIDlet into a BlackBerry-compatible format (LoginMIDlet.cod):

```
rapc import="c:\JDE4.0.1\lib\net_rim_api.jar"
    codename=LoginMIDlet –midlet jad=LoginMIDlet.jar
    LoginMIDlet.jar
```

Students then can load the resulting LoginMIDlet.cod file into the BlackBerry device from the desktop computer over a USB cable To do this, students use the javaloader command of the JDE installation. As an example, the following command can be used to load the LoginMIDlet.cod into the BlackBerry device:

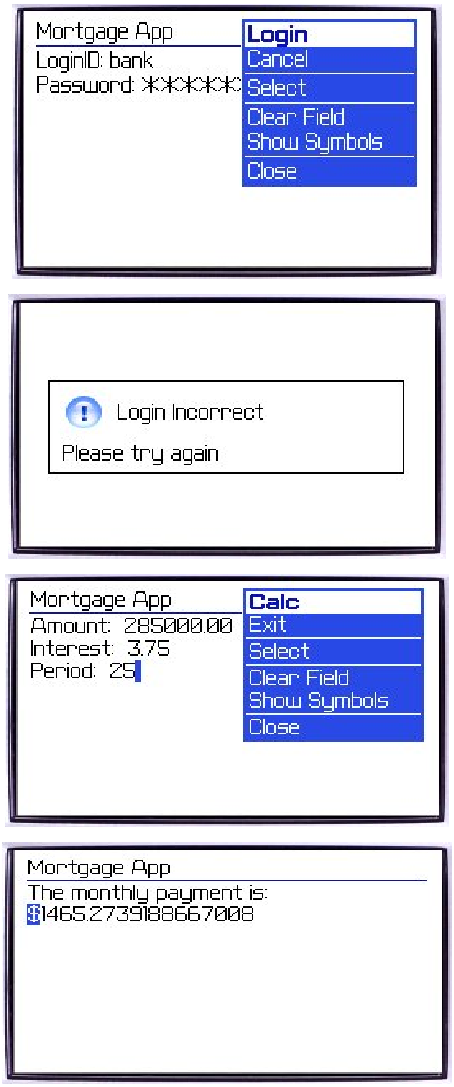
```
javaloader –usb load LoginMIDlet.cod
```

Once the application is loaded into the BlackBerry, students can run it as if it were a native application.

Students learn how to program networked applications capable of downloading information from remote servers. For example, the students experiment with a stock portfolio application that allows the user to enter a stock symbol and the application connects to

Yahoo's finance server and retrieves stock quotes. The application allows the user to add and delete stocks. In this exercise in the User-Centered Programming course, students learn about the Java ME database management system known as Record Management System (RMS) for mobile devices, and appreciate the use of threads in programming networked applications.

The assignments are an integral part of learning in our courses and they are worth 15 - 20% of the final mark. For the final assignment in the introductory programming course, students were asked to develop two versions of a mortgage calculator application: one for the desktop and another for the BlackBerry device. Students learn that the logic of the application doesn't change no matter on which device it will run. It is the user interface and interaction model that changes and thus students learn about the different methods for reading input from the user and handling events. The screen shots of a sample mortgage calculator developed by students are shown in Figure 4.


Mortgage App
LoginID: bank
Password: ******

Login Incorrect
Please try again

Mortgage App
Amount: 285000.00
Interest: 3.75
Period: 25

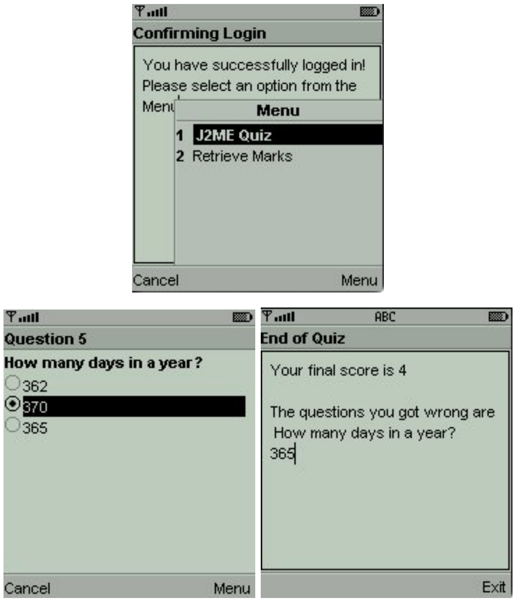
Mortgage App
The monthly payment is:
$1465.2739188667008

Figure 4. Mortgage calculator for the BlackBerry.

It is worth noting that CLDC 1.0 doesn't support floating point numbers; students were asked to use CLDC 1.1. In addition, the CLDC 1.1 math library doesn't support functions such as Math.pow() needed for mortgage calculation. Such utility function was provided to students.

In the User-Centered Programming course, students were asked to develop two applications for the BlackBerry: (1) Apply the Model-View-Controller design pattern to design and develop a multiple-choice quiz that will be marked on the fly as shown in Figure 5, and (2) Develop an application that enables students to retrieve their marks from a remote Web server by interacting with a Perl script. Students have developed these applications successfully, and have enjoyed the experience immensely.

As mentioned earlier, the BlackBerry comes with proprietary APIs that enable developers to build applications that take advantage of specific BlackBerry capabilities, however, in our courses we have used the standard CLDC and MIDP APIs, and the advantage of this is that the applications students develop can run on any Java-enabled device and not just BlackBerry devices. As an example, Figure 5 depicts the multiple choice questions quiz running in the Sun's Java Wireless Toolkit emulator. In other words, students would be able to load the applications they develop on their own Java-enabled devices.


Confirming Login
You have successfully logged in!
Please select an option from the
Menu

Question 5
How many days in a year?
○ 362
● 370
○ 365

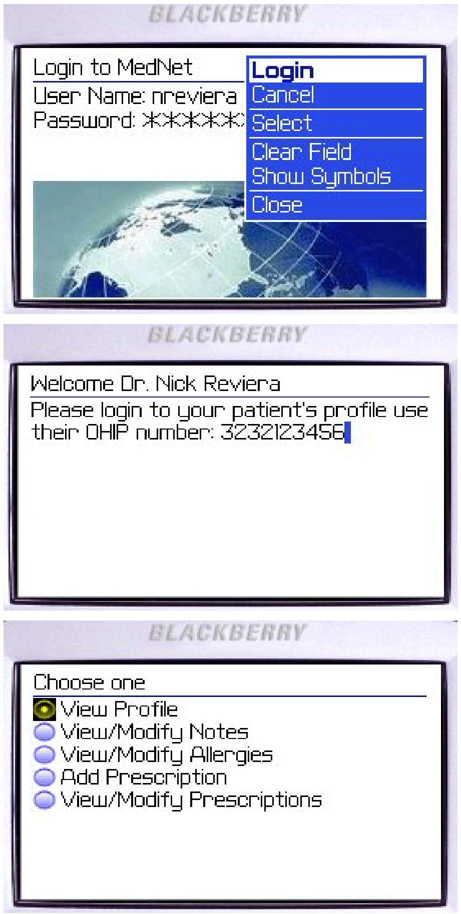
End of Quiz
Your final score is 4

The questions you got wrong are
How many days in a year?
365

Figure 5. Quiz with multiple choice questions.

7.1 A Second Year Group Project

Students in the User-Centered Programming course are required to work on a group project in which they are given a specification and asked to design and implement a software system. In Fall 2006, they were asked to develop a system for linking doctors and pharmacists to manage patients data and their prescriptions electronically. Since a database is required, this is a shared project with the Data-Centered Programming course. The system is meant to be stand-alone (not Web based) but students were encouraged to develop a Java ME client for the BlackBerry for extra marks. Two of the five groups (each group has six students) were up to the challenge and developed an impressive Java ME client for the BlackBerry. The client interacts with a Perl Script or a Java Servlet which will in turn perform the interaction with the database. A sample view of the doctor's interfaces are demonstrated in Figure 6.



Login to MedNet
User Name: nreviera
Password: *****

Welcome Dr. Nick Reviera
Please login to your patient's profile use their OHIP number: 3232123456

Choose one
● View Profile
○ View/Modify Notes
○ View/Modify Allergies
○ Add Prescription
○ View/Modify Prescriptions

Figure 6. The doctor view of the medical application.

8. CONCLUSIONS AND FUTURE WORK

As cellular phones and personal digital assistants become more pervasive than traditional personal computers (PCs), it is our responsibility as computer science educators to advocate for enhancing the curriculum by integrating mobile devices into educating our students in order to prepare them to work on state-of-the-art devices as their programming model is different than conventional desktop or Web applications. Such devices present challenges in developing software applications as well as unique opportunities. In this paper we have described our experience in integrating Blackberry devices into three computer programming courses. We believe that teaching computer programming in the context of simple wireless mobile applications provides a motivating framework for students and inspires them to work hard since most of them carry the latest cellular Java-enabled phones, and the experience they get is as practical as possible since.

The integration of BlackBerry devices into programming courses enable us to let students work on more interesting and exciting assignments and projects in such courses as well as in upper level courses. We believe this will benefit the students as they are exposed to a new and rapidly advancing area of computing. In addition, our Applied Computing program has two eight-month co-operative education work placements, and the integration of Blackberry devices into our computing curriculum offers our students an advantage over other students as they will be prepared for co-operative jobs at RIM or other companies that look for expertise in wireless mobile application development.

We encourage you to integrate such devices into the curriculum to introduce students to the programming model of such devices.

For future work, we plan to encourage fourth-year students to do their capstone projects on handheld devices in order to develop custom applications for such devices so that such applications can be used by first and second year students as well as students who carry such devices from other disciplines.

9. ACKNOWLEDGMENTS

This paper is based on a project made possible by a donation of blackberry devices made by Toyota Canada, and another donation of SIM cards, activation fee waiver, and discounted data service fees made by Rogers.

10. REFERENCES

[1] Alexander, B.: Going Nomadic: Mobile Learning in Higher Education, Educause Review, Vol. 39, No. 5, Sep/Oct 2004, pp. 28-35.

[2] BlackBerry Developer Program: http://www.blackberry.com.

[3] Forte, A.: Programming for Communication: Overcoming Motivational Barriers to Computation for All. Proceedings of the IEEE Symposia on Human-Centered Computing Languages and Environments, Auckland, New Zealand, 2003, pp. 285-286.

[4] Mahmoud, Q.H., Dobosiewicz, W., Swayne, D.: Redesigning Introductory Computer Programming with HTML, JavaScript, and Java. In Proceedings of the 35th Technical Symposium on Computer Science Education (SIGCSE 2004), Norfolk, Virginia, USA, March 2004, pp. 120 - 124.

[5] Nagappan, N., William, L., Ferzil, M., Wiebe, E., Yang, K., Miller, C., and Balik, S.: Improving the CS1 Experience with Pair Programming. Proceedings of the 34th SIGCSE Technical Sympoisum on Computer Science Education, Reno, Nevada, USA, March 2003, pp. 359-362.

[6] Pair Programming: http://www.pairprogramming.com.

[7] Research In Motion (RIM): http://www.rim.com.

[8] Robson, R.: Mobile Learning and Handheld Devices in the Classroom, Eduworks Corporation White Paper, Corvallis, Oregon, USA, 2006.

[9] Sun Microsystems, Java ME: http://java.sun.com/javame.

[10] Wagner, E.D.: Enabling Mobile Learning, Educause Review, Vol. 40, No. 3, May/June 2005, pp.40-53.