ABSTRACT

App Inventor for Android is a new programming environment that allows novice programmers to build applications for Android mobile devices. In this paper, we describe a healthcare IT course in which students with little or no programming background built healthcare related mobile applications. The course was designed based on the principles of studio-based learning, with students completing and sharing projects of their own choosing. Mobile applications were used to illustrate the role of IT applications in healthcare and teach design and user interface principles. In this paper, we discuss our experiences with this augmented course, describing the rationale, the process of introducing the new material, the student-designed applications, and the results from the course evaluation.

Categories and Subject Descriptors
K.3.2 [COMPUTERS AND EDUCATION]: Computer and Information Science – Information Systems Education.

General Terms
Experimentation, Human Factors, Languages,

Keywords
Healthcare informatics, education, studio-based learning, mobile applications

INTRODUCTION

Healthcare IT (HIT) is a field attracting a great deal of interest within the computer science and information technology community. There is currently a push to expand HIT education in the United States, in part due to the American Recovery and Reinvestment Act of 2009, which mandates that healthcare providers adopt certified electronic medical record systems by 2015 or face reduced Medicare reimbursement. As a result, hospitals need to hire many additional HIT specialists. A report by the Computing Community Consortium on Information Technology Challenges for Healthcare, mentions the need to train “fledgling” computer scientists in the biomedical domain as a critical need [6]. For this reason, educational institutions are ramping up their offerings in this area.

At St John’s University, we have established a HIT undergraduate major. This degree is housed within the computer science department but has been developed in conjunction with our healthcare administration department. Students take coursework in healthcare administration, computer science and information technology, and in a set of specialized HIT courses. Students who graduate from this program may end up specifying, purchasing, designing, and even implementing healthcare-related software systems. Thus, it is important that students learn about the system design process as it relates to the healthcare setting.

Like many other IT fields, there is currently a lot of interest in mobile applications in the HIT world. Mobile devices, such as smartphones and tablets, are uniquely suited to the healthcare setting because of their portability and ubiquity. Healthcare professionals are on their feet, moving around constantly, in the course of their work. Computer applications that require physicians and nurses to enter data into a stationary computer have long been a source of complaint, since it is often difficult to locate computers right at the site of patient care.

One issue for HIT education is that programming mobile devices is relatively difficult. Developing for the Android or the iOS platforms requires a strong knowledge of object-oriented concepts and multithreading. HIT majors in our program, however, only take two computer programming courses and thus do not have the background to tackle mobile device development in the traditional fashion. However, the introduction of a new Android development environment aimed at novice programmers, App Inventor [15], has changed all of this. Even students who have never programmed before can develop full-fledged mobile applications using this environment. It has been used in a number of computer science courses to teach computational thinking. In our program, however, we are using it to teach the process of designing applications for the healthcare setting.

We decided to focus on mobile applications for the healthcare setting in one of the core HIT courses in our sequence. This course is the second semester in the sequence, and focuses on data and information in the healthcare setting, the role of electronic applications in healthcare, and the systems lifecycle. We used mobile applications as a means of discussing these topics, and added material on building applications in App Inventor, case studies of mobile applications in healthcare, and material on design of user interfaces for healthcare. The course was designed based on the principles of studio-based learning [8], with students completing and sharing projects of their own choosing. In this paper, we discuss our experiences with this augmented course, describing the rationale, the process of introducing the new material, the student-designed applications, and the results from the course evaluation.
2. BACKGROUND

2.1 Mobile applications in healthcare

Applications for mobile devices may be designed for patients or for providers. Patient oriented applications often are intended to reinforce healthy behaviors, educate about a health related topic, or allow a patient to record treatment and symptom data. According to Klasnja and Pratt [10], key drivers of adoption by patients of mobile technology include the fact that these devices are pervasive – everyone seems to own one, and they are carried around everywhere – and they have the capability of interacting with the user’s environment through technologies such as GPS, cameras, and the ability to communicate with external software services. A recent survey by the Pew Internet and American Life Project found that nearly 3 in 10 Americans have accessed medical information on their phones. In addition, about 10% use health-related applications on their phones [3].

Similar reasons are driving provider adoption. An amazing 84% of physicians use smartphones [7]. In addition, needed infrastructure such as networking is in place at most hospitals. Hospitals are finding it cheaper and faster to develop mobile applications in-house [5]. And finally, the portability of these devices makes a lot of sense in the hospital setting, where it is often difficult to locate computer terminals right at the point of care. Mobile applications allow providers to enter and access patient information, to write orders, and to capture charges for billing purposes in real time [1].

2.2 Mobile technology in IT education

Currently, there seems to be little or no published research in teaching the design and implementation of mobile applications in healthcare IT courses. Standards such as the IMIA draft recommendations do not explicitly mention mobile technology, although the recommendations do contain categories that would likely encompass mobile technology [13].

There are published papers on teaching mobile development in the computer science education literature. Examples include [16], [9], [18] and [2]. Most of these are upper division courses that focus strictly on learning advanced programming technologies such as the Android SDK or Objective C for iOS. Massey, et al. [14] however, describe an approach that is closer to our approach. They created a problem-based environment with a IT graduate course in which students with diverse backgrounds, including non-programmers, collaborated to explore the both the use and the design and development of mobile technologies in specific environments.

The closest effort to ours was a studio-based computer science course that used App Inventor [4]. This was an introductory computer science course that taught fundamental programming concepts such as iteration and procedures. The course was organized using studio-based principles; students learned through projects of their own design and shared their projects with each other. The course differed from ours, however, in a significant way; since their course was a programming course, it focused on correct use of the programming concepts rather than user interface design or the role of the application being designed in an organization. Their course also was completely devoted to App Inventor. In our course, however, we use App Inventor as a means to another end: to learn about the role of mobile apps in healthcare and to explore how design choices affect the user experience.

2.3 App Inventor

App Inventor [15] is a new technology that was developed to allow nonprogrammers to build Android applications. App Inventor was initially hosted at Google but now is hosted by MIT as part of its Center for Mobile Learning. It draws on the lessons of earlier programming environments for beginners, especially Scratch [17] and Lego Mindstorms [11], sharing their visual approach to programming. What makes App Inventor an especially good platform for students to explore the world of design with computers is that it is relatively easy to create an application that looks and behaves like other mobile phone applications. Unlike earlier programming environments for beginners, applications created with App Inventor can interact with the outside world. This means that creators of these applications can assess their creation with respect to the world at large and ask questions about the impacts of their design choices and the effects of their application on real people.

App Inventor is similar to Scratch in that it uses a blocks metaphor for specifying the logic of an application. Users are freed from learning the intricacies of a programming language syntax, and instead choose from a menu of behavior blocks to build the program. App Inventor differs from Scratch, however, in that there is a first phase in application development in which the design and layout of the user interface is specified. This forces the developer to think about interface design very early in the process of building an application. The App Inventor environment includes a realistic phone emulator which shows changes to the user interface in real time; in addition, students were able to test on a set of inexpensive Android phones. Figure 1 shows the blocks editor and the emulator running a very simple program in App Inventor. This program simply switches colors based on a random number.

![App Inventor Blocks Editor](image)

3. COURSE DESCRIPTION

The course that is described in this paper, HCI1002, “Healthcare Information Flow and Data Management”, is the second semester course in the HIT sequence. Most of the students who take it are HIT majors; however, computer science majors also take it either as part of an HIT minor or simply as an elective. During the
semester when the augmented course was offered, ten students finished the course. Most of the students were HIT majors; three had never taken a programming course, two were taking the first programming course concurrently, and two had taken over a year of programming. Thus, there was wide variation among the students in programming knowledge.

The course was run as a hybrid class, which meant there was a significant online component, as well as one class meeting a week. Since the mobile application development component of the class was intended to be hands-on, we focused on that during the class meeting each week, and the chapter readings on healthcare applications and systems lifecycle formed the online component. The course was designed to use studio-based learning principles. One of the key components of the studio-based approach is that students should share their thought processes as they design their projects, as well as sharing the finished projects. To that end, students were asked to create and maintain a course portfolio, using Google Sites. One of the roles of the portfolio pages was to contain student reflections on their projects.

3.1 App Inventor labs
During the first half of the course, in-class sessions were devoted to guided labs designed to teach various aspects of mobile application development. For students who had never programmed before, it was their first encounter with constructs such as conditional and iterative logic. Even for students who had taken a programming course before, there were many concepts to be learned since none had ever seen event-driven programming before or worked so closely with a graphical user interface. We made use of the educational materials maintained on the Center For Mobile Learning’s educator website, as well as the tutorials in [20], but modified a number of them to make them more specific to healthcare applications.

![Figure 2: Lab Page in Student Portfolio](image)

Students were asked to create a page for each lab on their class portfolios. Many of the labs included discussion questions in which students were asked to relate the concept in the lab to healthcare applications. Students placed their answers to these questions, screenshots of their applications, and screenshots of the blocks that they used on their lab page in their portfolio. Figure 2 shows an example of such a page.

Labs started with topics such as setting up the portfolio page and writing a simple app, and progressed through topics such as using the drawing canvas, adding sound, simple games and animations, and basic programming concepts such as variables, loops and conditionals. Later labs were more complex and more focused on healthcare and design issues. Here are some of the later labs:

- Lab4: OrderDroid is an application described in [19] that takes user-input orders and sends email. Once students completed this app, they were asked to come up with a healthcare application that uses the same idea of reading a collection of user input and sending a message based on that input. They then designed and implemented it.
- Lab5: Students completed a quiz app described in [20], and then were asked to identify a health education need and a target audience (children, the elderly, etc) and design a similar quiz application, addressing the health education topic they had chosen. This was done in order to get them thinking about designing for particular audiences before getting to the final project.
- Lab6: Students built an application that lets a user enter patient vitals and then uses Google Charts, a web-based service, to build a chart showing temperature over time.

Once students had completed these labs, they had acquired a toolbox of techniques for building applications and were ready for the next phase of the course: researching, designing, and building a more complex healthcare application.

3.2 Project organization
In preparation for the final project, students were asked to read several papers describing types of mobile applications in healthcare, as well as a paper that described in detail the process of designing and evaluating a mobile user interface for elderly patients [12]. We also spent one class period looking at healthcare related mobile applications and discussing their purpose, target users, and design. Then, the students, grouped into pairs for the purposes of the project, were given the project assignment. There were several phases to the project.

First, the students were asked to choose a general healthcare need for their project. Students chose areas related to their personal experiences for the most part, such as asthma control, diabetes education, and nutrition. One student worked for a care facility for developmentally disabled adults, and chose to focus on daily record logging in that facility. After choosing the healthcare need, they were asked to write a short paper surveying other software applications, including desktop, web-based, and mobile systems, addressing the same need. This familiarized them with the range of applications in the chosen area.

The next step was to submit a design on their portfolio site. The design had to be specified as a series of screen mockups, showing clearly how interaction with the app would proceed. In addition, they were asked to answer the following questions as part of the design submission:
• What need does the project address and why is this an important need?
• Who are the target users?
• How does your design meet the special needs of your target users?

Once the designs were posted, the next step was for the students to critique each others’ designs. Students were required to comment on every design, keeping in mind aspects such as target users and visual design. Critiques were posted as comments on the design sites so that students could respond to each other. Most of the comments were essentially just agreement with a design. But a number of student comments were more insightful, and led to real changes in the applications. Here are some sample comments:

“I like this app a lot it looks very organized and well set up. I suggest maybe adding a chart or page that shows a schedule that displays the times the user has to take his medicine. You can make it so that the user enters the times and this can be connected to the alarm that you are trying to incorporate into the app. Then you can make it so that at the end of the application it displays if the user took all his medications.”

“Your app looks very clear and right to the point. I suggest maybe adding pictures or more color to the app so that it won’t look too plain and more user friendly. You can also include an age input box because from what I have read I know that asthma attacks are more frequent at a younger age, and I think that maybe should be included for record.”

“Its a very good idea. You should add a pop of color or some other type of graphic so it can engage the user a bit more. Also it could be cool if that data could somehow be transmitted into a cloud of some sort with the user’s medical history”

Students were expected to respond to the critiques and incorporate them as changes in their projects. The pair that designed the daily log app for the care facility was advised by other students that their design was too busy for a small smartphone. That group realized that the busy interface had to do with the care facility’s requirements for the app, and decided to design for a tablet interface instead, even managing to get a tablet as a loaner from the care facility.

The final step was to implement the project. The students had 3 weeks in which to accomplish this. For the most part, the students were careful to base their projects on App Inventor constructs that they already knew. The one problem that several groups encountered was the lack of support within App Inventor for multiple screens. One student, however, found a workaround on a discussion board, and showed the other project groups how to do it. In addition, another group had included Twitter posts in their original design, but was unable to make that functionality work.

During the final week, the students gave presentations and demos of their projects to the rest of the class. They also created final project pages on their portfolios. The instructions for creating the final page stated: “you should put another page on your portfolio with screenshots of the final app, a description of how it works, a description of changes you made and why, and reasons why you incorporated or did not incorporate suggestions made by the other students. Your final project page should “sell your system” a little bit – it should be attractive and make readers want to try your app.”. Students were also directed to put screenshots of their blocks on the page, as well as a downloadable version of the app so that others, including the instructor, could try it out.

### 3.3 The projects
There were five projects submitted in the class. Here is a list:

- Diabetes app. This app consists of two parts. The first part is educational, consisting of a step by step guide on using a glucometer. The second part allows the user to record blood sugar readings and medications and chart them over time. Figures 3 and 4 are illustrations of the app from the students’ final presentations.
- Keep Breathing. The Keep Breathing application is designed to help asthma patients record their asthma attacks and their peak flow readings. Users of the application can record the date and time of a peak flow reading. Later, they can view this data on a chart. The app also allows patients to record the details of their asthma attack, details such as date, time, location, trigger, symptoms and treatment.
- Care Chart. One of the members of the pair working on this project works in a care facility for developmentally disabled adults. He observed that the workers record myriad pieces of information throughout the day, such as medication times, food intake, seizures, and other observations, on paper. He thought that a mobile application would ease the burden. The application was designed for a tablet rather than a phone, and designed to be as easy to use as possible: many data points are entered simply by touching a circle. The students were able to get feedback from the caregivers at the care facility in order to improve the application, and there is interest in finishing it for use by the caregivers.
- Healthy Choices. This is a game aimed at children, to help teach healthy food choices. The game presents pairs of food choices and asks the player to choose the healthier one. Figure 5 shows the project page for this app, and Figure 6 shows the AppInventor blocks used to create the app.
- Another app aimed at children is designed to encourage them to exercise. This app, which is started as the user begins to exercise, shows how close the user is to meeting recommended daily exercise targets, using an hourglass metaphor. This app also has a mechanism for submitting the data to Twitter, although this is not fully functional.

The following figures show illustrations from some of the student project pages. These give the reader an idea of the student work.

![Application Part 1- How to use a glucometer.](image)

Figure 3: Diabetes app
In order to measure student perspectives on what they had learned in the course, a 3 part survey was developed and administered at the end of the semester. The first part of the survey consisted of 7 questions on student background before taking the course. We found that 7 students were majoring in HIT, 3 in CS, and 1 in security. Only 2 of the students had worked at a healthcare related job or internship, and 6 had used a healthcare related smartphone app. All 10 owned smartphones, either Android or iOS.

The second part of the survey asked 3 questions about their experience with App Inventor. Students were asked to rate the difficulty of working with App Inventor, using a scale of Easy, Somewhat Easy, Medium, Somewhat Difficult, and Difficult. The most common response was Medium, chosen by 7 students; Somewhat Difficult was chosen by 3 students. All 10 students responded with Yes to the question “Do you think that learning to develop mobile applications will be useful to you in the future”? Students were also asked to list the aspects of App Inventor that they found most difficult. By far, implementing multi-screen applications was chosen as the most difficult aspect, with 8 students mentioning this. App Inventor does not have explicit support for multi-screen applications, so this result was not surprising.

In the third part of the survey, 6 class topics were listed. Students were asked to list the ones they had learned the best, the ones that they felt were most important to their future career, and the ones that they felt were least important to their future career. The questions were also more open ended and allowed them to insert comments. The topics were

1. Understanding how data is used in the healthcare setting
2. Understanding how electronic medical records are implemented
3. Understanding how the Internet has impacted healthcare information systems
4. Understanding how mobile applications are designed and built
5. User interface design and its impact on user acceptance
6. Group communication and working in a team

Table 1 lists the numbers of students choosing each response:

<table>
<thead>
<tr>
<th>topic</th>
<th>Most important</th>
<th>Least important</th>
<th>Learned it the best</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

The responses do not add up to the number of students in the class because students could choose more than one option; in addition, 2 students chose “None” in response to the question about least important topic. Also, 2 students did not answer the question about the topic that was learned the best. While the numbers are too small to be significant, it is noticeable that more students chose the mobile development, user interface design, and group communication topics as most important, and the fewest students chose those as least important.

5. DISCUSSION AND SUMMARY
The survey results showed that the App Inventor programming environment was well suited to the students’ backgrounds, with most choosing Medium or Somewhat Difficult as the level of difficulty. In addition, all of the students agreed that learning about mobile applications in the healthcare world would be important to their futures, with some students adding comments such as “yes, very”, and “extremely useful”. Thus, we believe that...
adding the focus on mobile applications and App Inventor was engaging and educational for the students, and we plan to offer the course in the same fashion next year. We are also exploring partnerships with area healthcare agencies so that the students can work on applications for real clients, adding another level of critique.

In sum, App Inventor seems to be a promising technology for quickly bootstrapping students with little programming experience into mobile application design, while powerful enough at the same time to engage more experienced students. Applications built in App Inventor are full-fledged mobile applications that harness the capability of mobile devices to interact with the external world through messaging, GPS, cameras, and other features of the platform. This allows students in fields such as HIT to explore design issues for mobile devices without having to spend years learning advanced programming techniques.

6. ACKNOWLEDGMENTS
This project has been supported by a grant from the St John’s University Teaching and Technology Fellows program.

7. REFERENCES