Interactive Qualifying Project:

iOS Application Design for Laboratory Use

Alexander G. Orphanos ME 2012
David A. Polito BME 2012
John D. Neary BB 2012
John A. Erikson CE 2012

Advisors:
Prof. Mike Buckholt
Prof. Jill Rulfs

E10, A10, B10, C11

Introduction

The focus of this IQP will be the development of an iDevice application used to streamline and enhance the laboratory learning process. This topic was chosen through a combination of desire to rework and reorganize the presentation of laboratory materials, especially within biology and chemistry laboratory work. As is, procedures, data collection, and tutorial videos remain fragmented and often inconveniently located in separate locations within MyWPI. In order to complete their laboratory work on time, students in these labs must spend time learning where these fragmented elements are, and how to juggle them effectively, rather than directly proceeding with the laboratory work, as well as disrupting the learning experience intermittently throughout the lab period. The proposed iDevice application attempts to take these disparate elements of laboratory material, and combines them into one streamlined and efficient experience for the student.

In the last few decades education has undergone several significant revolutions. Prior to the Internet, students were somewhat physically tied to the education process: whether actually attending class, obtaining a copy of a recorded lecture, or attending a correspondence course and testing in a registered test-taking center. However, with the explosion of the Internet into the mainstream in the last fifteen years, the learning process has become somewhat delocalized; the learning materials can now be accessed from any computer with Internet access. As can be expected, this has resulted in learning at a distance, and degrees from reputable colleges can be obtained solely through online work. By 1997, the United States government launched the Advanced Distributed Learning Initiative. This program, headed by the Department of Defense, was tasked with the following goals:

Comment [m1]: Describe what this is in more detail so the reader knows how it relates to your work.

- Identify and recommend standards for training software and associated services purchased by Federal agencies and contractors.
- Facilitate and accelerate the development of key technical training standards in industry and in standards-development organizations.
- Establish guidelines on the use of standards and provide a mechanism to assist DoD and other Federal agencies in the large-scale development, implementation, and assessment of interoperable and reusable learning systems.

Since the time of its creation, the ADLI has “fostered the development, dissemination, and maintenance of guidelines, tools, methodologies, and policies for the cost-effective use of advanced distributed learning resource sharing across DoD, other Federal agencies, and the private sector. It has also supported research and documentation of the capabilities, limitations, costs, benefits, and effectiveness of advanced distributed learning” [1]. Additionally, the Initiative has resulted in the development of SCORM, the Sharable Courseware Object Reference Model, a common coding suite for all online-based education, down to website protocol, format and proper coding language applications for education sites to implement [2]. This format has since been widely adopted by both government agencies conducting delocalized training of its employees, as well as private corporations and educational institutions.

By 2005, 55% of universities and colleges surveyed offered online undergraduate courses, and 26% offered online graduate courses that adhered to the SCORM protocol [3]

.In recent years, online sites like YouTube and GoogleDocs have allowed instructors to post their lectures and problem solutions online, and many universities have an equivalent video sharing program on campus; Massachusetts Institute of Technology has gone so far as release the majority of MIT class content to the Internet under the OpenCourseWare header (http://ocw.mit.edu/about/). While these developments have tremendously increased the potential of distance learning, the same tools can and have been tweaked to provide valuable advantages to the on-campus university experience.

Comment [m2]: You should see if you can have real references for all of these statements. How many schools and how do they use it. What about course management solutions like Blackboard (Mywpi)?

Comment [YUN3]: So I tried to find some real data for the percentage of universities and colleges that use Blackboard, or similar Web 2.0 portals but had no real luck. Will continue to search.

As referenced above, many of the tools and resources needed by students are kept in separate locations electronically. As a result, a successful student must also learn to utilize these disparate tools in an intelligible manner in order to succeed academically, as well as learning the academic material itself. Keeping track of updated course information is often difficult when the possible changes can be found in your e-mail inbox, course notifications given by university electronic class management sites or in a class GoogleDocs account. It becomes even harder to manage when some classes use different combinations of the aforementioned notification systems. While class management sites like Blackboard collate class information, video lectures, and links in a single location, there is little communication between the disparate elements. This disorganization is seen most explicitly in the WPI laboratory experience as students must use GoogleDocs to store class data for analysis, view laboratory technique videos online via Blackboard or other media streaming websites, and stay updated on lab dates and actual lab procedure through e-mail or posted notifications on MyWPI itself. However, recent progress in the availability of online devices allows for a number of possible solutions; why not keep all the relevant class data in a single location, and why not ensure all the tools required for a laboratory project are located within the same, easy to navigate system?

By streamlining the laboratory materials and resources, instructors and students would be able to jump directly into the lab work. Additionally, with the inclusion and increased ease of use of video tutorial material attached to the written procedure, students would be less prone to wasting resources through unclear procedure. While it would be possible to streamline the laboratory process in a new program on a laptop or personal computer, there is little incentive to do so: a laptop or personal computer has the computing power to juggle all possible material, while also allowing students to access the Internet and possibly disturb the focus of the learning. Additionally, not all students possess laptops or personal computers that can easily be transported across campus. However, with the recent explosion of cell phones into the young adult market, only 0.2% of college age students do not own a personal cell phone [4]. Many of these cellphones have the hardware and software potential of smaller computers in their own right, and can access the Internet, like the iPhone, in its fourth iteration on the market. In spring 2009, a quantitative survey conducted by the EDUCAUSE Center for Applied Research of nearly 32,000 college freshmen at 115 educational institutions found that 51% of all undergraduates already possessed a web-enabled smartphone, and another 12% of respondents were planning on purchasing a web-enabled phone in the next year [5]. With this information in mind, and the fact that smart phones now constitute more than a third of all phones owned by young adults from the ages of 15-24 [6], make mobile computing devices the clear choice for the next generation laboratory experience. Interestingly, only 27% of college students owned smartphones as February 2009; it appears that college age students have been the fastest to adopt new mobile device [4]. Even further, 90% of students that reported possessing a smartphone actually used their phone's Internet functionality on a regular basis [4]. They possess the hardware and software to facilitate such a streamlined experience, they allow the student complete freedom to review laboratory information through applications, access to school websites, streaming video sites like YouTube, and more importantly, are already owned by a large proportion of the student body.

In the last few years, smartphones like the iPhone and the Android have entered the mobile device market, and enjoyed explosive growth. In Q2 2008, total smartphone penetration in the

US was at 10%, and it's been projected by the market research group Nielsen that penetration rates will pass 50% in 2011 [7], an astounding amount of growth in only three years.
It's an easy prediction to make that smartphone usage will continue to grow explosively especially as mobile programming languages continue to become easier to use, or go completely open source, like Google's Android operating system or Nokia's Symbian operating system. The potential uses for such devices have only just begun to be explored, as evidenced by the success of Apple's App Store for its iPhone and iPad; the store "opened" on July 10th, 2008 with roughly 500 applications, and as of June 7th, 2010 there were well over 220,000 applications available for purchase. In the first four days of the App Store's existence, over ten million applications were downloaded [8], and over $1 billion dollars have been paid out to iDevice application developers, as reported by Steve Jobs at the annual Apple developer's conference last June. [9].
It should be noted, however, that the presence of educational applications and functionality on smartphones has been traditionally limited, hindered by the relatively slow development of internet capability among mobile network carriers [10]. As of February 2010, educational applications presented only 2% of Apple App Store profits, while game apps accounted for nearly 60% of total developer profits [11]. The lack of educational services in mobile smartphones may be due to the fact that most mobile services view educational applications to be a non-profitable market, but the recent success of a Stanford team of programmers suggests that just like the early days of the Internet, it only takes a creative approach and a good idea to monetize a previously non-profitable service. In particular, apps like BirdSight have been able to monetize services to niche markets, i.e. bird watchers, allowing them to catalogue and share their bird sightings with other enthusiasts who have also purchased the application [13].

A little over a year ago, a team of programmers from Stanford wrote a smartphone application called iStanford, a collection of Stanford resources like campus map, course guide, and staff directory. Since that time, the application has seen an enormous explosion in usage, as well as the addition of several new features, like an interactive map, and real-time scores of Stanford sports games. The program was so successful that the team was approached by Duke University to apply the same treatment; DukeMobile was released March 2009. The seamless packaging of university information allows the student to stay updated with all relevant campus information; all in the same program [12].

The programming kit used by the Stanford team was the iPhone Software Development Kit, a powerful suite of versatile programming languages. In practice, the suite allows you to program the same commands in any number of ways, allowing the programmer to use the strengths of each programming language, and avoid many of the weaknesses. Our IQP team used the same SDK to create an application similar to the iStanford application. Where iStanford was a general campus information suite, our application will be used at the classroom and laboratory level, allowing students to quickly and efficiently use the laboratory or class data. With biology labs in mind, the application has been designed to allow students to quickly scan through multiple sections of the procedure simultaneously, minimizing error, as well as providing links to online tutorial videos.

Comment [CCC4]: How are you using data here? Perhaps resources or information captures the concept better?

Methodology

Programming for the iPhone SDK was very easy to learn and use . A person with little to no programming can pick up App Building for the iOS quickly. There are two ways to learn how to use the Xcode platform and Interface Builder: iOS Programming Guides and online blogs/video tutorials from other App Development. The importance of efficient coding and Interface Layout cannot be stressed enough due to the iOS's need for efficient and correct coding. Without it, the application will crash and fail to work.

The typical development cycle for applications usually has around (for the time used in development) 25% testing, 25% debugging, 45% coding and 5% designing. In the iPhone development process the cycle goes 8% testing, 17% debugging, 17% coding and about 58% designing. The reason for this is because of the SDK package. The integrated use of the Interface Builder and the Xcode platform allows an organized union of coding and design that can save much of the agony of straight coding.

Comment [CCC5]: Is there a reference for this?

Comment [AO6]: The reference for this is from (11-20)

The first step taken in developing the application was to decide what the application needed to do. This is an extremely crucial point in developing, considering Apple informs developers of the apparent 58% of the development process will be design. The key is functionality and simplicity. For example, iPhoto on the iPhone is “Just for sharing digital photos”. The five-word description of the application literally explains its function. The group’s statement was to create an application that “Organizes laboratory procedures for iPhone/iPad use”.

The design for the interface of the application was a UITabBar. This interface allows the user to have a bar at the bottom of the screen that has multiple tab bar buttons. Every tab bar button loaded a view that was created in another file and then uploaded to its respective tab bar. The first tab contained the WPI logo and the course and lab number on a static background. The second tab contained the procedures of the lab, which were placed in scrollable text view and a static background. The third contained all three safety procedures in scrollable text view with the WPI logo and a static background.

In preparation of the project, a multitude of coding books were read to increase the groups knowledge of programming in the C++, Cocoa touch, and Objective C languages which we would be spending a lot of time with during the development of the application **[11-20]**. The next step was to figure out how to set up with Apple's Development Program so that testing and device use could be done when it was time. The group needed to become aware of the uses and tools of the SDK(Software Development Kit), Xcode, and Interface Builder.

The SDK contains all the libraries of information that Apple gives to developers. Every platform has an SDK package, for instance, there is also an Android SDK. For Apple and iOS platforms, it contains all the coding languages(C++, Objective C, Cocoa Touch, Java, etc.) and Interface Builder tools so that you can access it all with ease. Xcode uses the IDE or "Integrated Development Environment" which was designed to considerably reduce the amount of time you would spend coding on most of other platforms. It organizes all the files that are in the project in a list on the left side of the program and each file can be accessed and opened in another screen for editing and whatnot. The Interface Builder application allows the programmer to design the user interfaces in a graphical nature. In other words, it allows the programmer to visually place and organize all the interface options on a screen instead of having to code in the exact locations of the various interface options. This takes a large portion of the coding process out of the loop and allows the programmer to concentrate on the design of the user's application. The Interface Builder works in unison with Xcode, allowing any actions or objects written in the code to be linked and hierarchically connected with the user options (i.e. - a Button that is connected to an action called in the code).

Comment [CCC7]: Is this all written in your own words, and should you be referencing some source here? It sounds a bit like promotional material or instructions....

Comment [AO8]: This is all in my own words. Theres no reference needed in my opinion because it's a bunch of coding mumbo jumbo that just lists a few examples of what can be done with xcode and interface builder.

Most of the preparation was going through each book and doing sample applications so we could become aware of all the options and procedures that needed to be done for the lab application to become what it is today. The format was finally decided to be a UITabbar application. This took going through various online resources that varied from video tutorials of step-by-step processes to online articles with the sample code and written explanations of their process **[21-23]**. These resources were the most valuable because of their relevance to the project and use of simplified coding and to allow the app to run effectively

Design

The lab app initially was intended to be a tab bar application with an opening WPI log-in screen, a tab for each section of procedures, another tab for safety videos and then a tab at the end to insert data into either a Google document or blank page to type into. Due to the lack of programming knowledge, the group wasn't able to create an application with that much variety. The iPhone SDK is programmed to shutdown and restart if the program can't handle the processing so good and efficient coding is required for such a multi-layered app. Instead the group went with simplicity: a tab bar application that organizes the lab material in a new mobile platform.

The group decided on the BB2901 Lab 1. This was due to its simple lab procedure, which included using pipettes and waiting for bacteria to form. This lab was also good to test with because the design of the application could be simple and the lab could be tested relatively quickly and wouldn't waste lab materials either. The design for the BB2901 Lab 1 included one tab for the opening screen, one tab for both procedures, and a third tab for Safety Instructions in lab procedures. This design choice was made so that a series of tab applications could be made, and each one adjusted to its procedure and material respectively. Once again, the lack of coding knowledge and speed at which the group could learn at was not suitable for the time given for the project. The final design was meant to provide a usable test app that can provide the basics of mobile-platformed lab material to the lab procedure and give us a chance to test it with lab students and gauge the interest on this new laboratory format for procedure and preparation.

Comment [CCC9]: This either deserves expalantion…what is BB 2901? Or to be omitted

The opening screen tab included a WPI logo banner with the lab name and section. This tab was meant to be a simple title screen, informing the user of the lab and section. The procedure tab was where the procedures were placed. The purpose of this tab was to organize the procedures separately and allow the user to scroll through the text and access all the procedures for the lab on one tab. For this lab, it happened to be two vertically placed scroll text views that could be scrolled and set individually to keep ones place in the procedure. The safety screen tab included all three documents on safety procedures while in the laboratory. The purpose of this was to allow all users to have the safety procedures available to them at all times instead of printed out and hopefully in their notebooks. It was designed with the WPI logo at the top, and the three documents organized in a scrollable view for each document.

Comment [CCC10]: For use?

The final design was the most effective design that we had decided on because of its simplicity towards the organization of the lab resources and procedures so that users only have to run one application on their mobile platform to access over 30 pages of paper material. In the Apple Development Design process, this app's action is clearly defined by the statement: a tab bar application that organizes the lab material in a new mobile platform.

For testing purposes we had uploaded the lab application on the iPad, iPhone 4, two iPhone 3G's and two iPod Touches. This way we could have our test subjects test the app on a variety of iOS

Testing

The first test was done with the previous version of the app. The differences included no button on the title page, and an error that occurred when the user double tapped the text in any text view. The error was that the iOS keyboard was called upon with no way to exit the keyboard due to the design. The first test was conducted by six volunteers who had never taken the lab course, but were familiar with the lab process before. They were to test the applications functionality and format and then indicated whether they would like to use this format for lab sessions. The appeal on the format was a mixture of yes and no's while two issues were evident: The iOS keyboard issue was to be solved and the title page needed direction from the user to the tabs on the bottom.

Comment [CCC11]: Either here otr elsewhere you need to indicate how these people were identified and recruited. Were they people who had taken the lab for example?

Comment [CCC12]: This is a result, not a method

The keyboard issue was solved by making sure the text views were un-editable; they were previously selected to be editable. This allowed the keyboard to be accessed but with no option programmed in to allow the board to be exited. The button was added to attract the user's attention to the bottom of the screen. The button itself has no action but is still press-able. Now that the app was user-ready it was time to start the second round of testing which would involve the lab.

Comment [CCC13]: This last sentence is a bit informal

The second test was done in the laboratory with seven volunteers who were all involved in the biology major. We allowed them to conduct the test and use the devices in the same environment as the lab would provide. They used water in the test lab and didn't have to wait for the incubation process so that the testing could be done in under an hour. The results from the surveys were very informative and in favor of the new mobile platform for laboratory procedures.

Comment [CCC14]: What does this mean? Were they BBT majors? Had they been/were they enrolled in the lab you were testing? How were they recruited and selected? Maybe this needs to be a separate section in the methods.

Comment [CCC15]: This needs clarification. The uninformed reader has no idea what you are saying. Make it clear that there was a mock experiment going on to simulate the lab environment.

Comment [CCC16]: RESULTS

At the end of the testing process, we decided that there is in fact a place in our laboratories for mobile applications that organize the procedures. This decision could lead to paper-less lab sessions and a market for college level laboratory mobile platforms. Hopefully a group of CS Majors can take the progress we ended with and create a series of lab applications that will become standard in our laboratory classes.

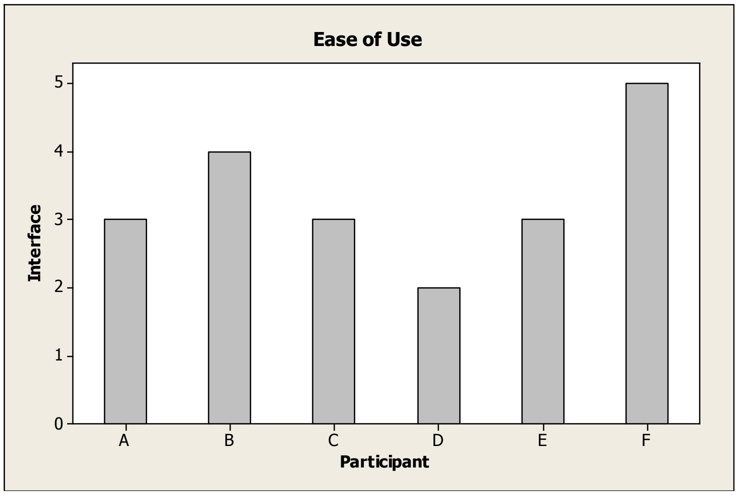
Comment [CCC17]: DISCUSSION. Methods just tell what you did!

Results

A preliminary study was conducted in order to assess different aspects of the application. Six WPI students of various non-biology majors were iven either an iPhone or an iPad and asked to navigate through the application. After reviewing the application for a period of twenty to thirty minutes, they were asked to complete a short survey (Appendix 1). Figures 1-5 give a visual representation of the responses to the first five questions of the survey. Each number on the x-axis corresponds to one of the six survey participants. The numbers on the y-axis correspond to how they ranked each category, with one representing “not at all” and five representing “very”. The categories are ease of use, ease of navigation within the application, aesthetics, relevance to labs, and preference of the application to written lab documents. Each question was ranked on a scale of 1 to five. A response of 1 indicates below average, 5 indicates above average, and 3 indicates average. Responses were presented on a five point Likert scale where 1 indicates “not at all” and 5 “very.

Comment [CCC18]: Each figure needs this sort of legend so it can stand alone an be interpreted by the reader.

Figure 1: Ease Of Use. Results for each individual tester are shown. Ease of use is a representation of how easily the participants could make effective use of the application.



Comment [CCC19]: After each figure, you need text to guide the reader. You also need to say what the intermediate choices on the scale were. IS three a neutral response?

Figure 2: Ease Of Navigation. Results for each individual tester are shown . Ease of navigation is a representation of how easily the participants could make navigate through the separate portions of the application.

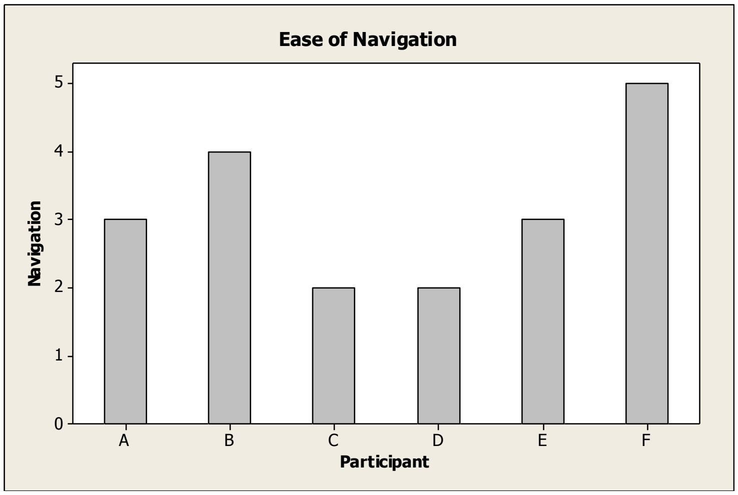


Figure 3: Aesthetically Pleasing. Results for each individual tester are shown. Aesthetically pleasing is a representation of how attractive the participants found the application.

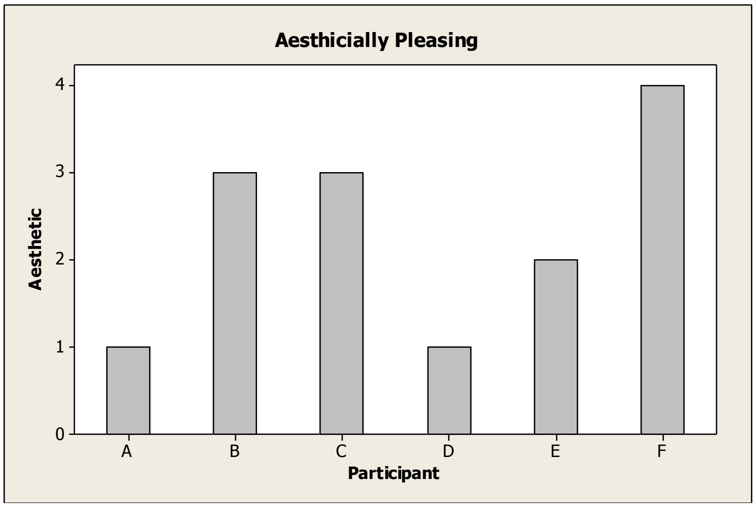


Figure 4: Relevance to Labs. Results for each individual tester are shown. Relevance to labs is a representation of how relevant the participants found the application to the performance of laboratory procedures.

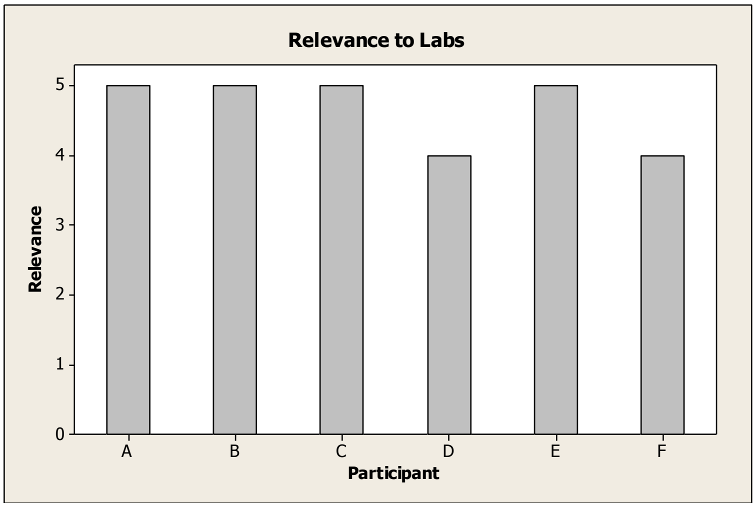
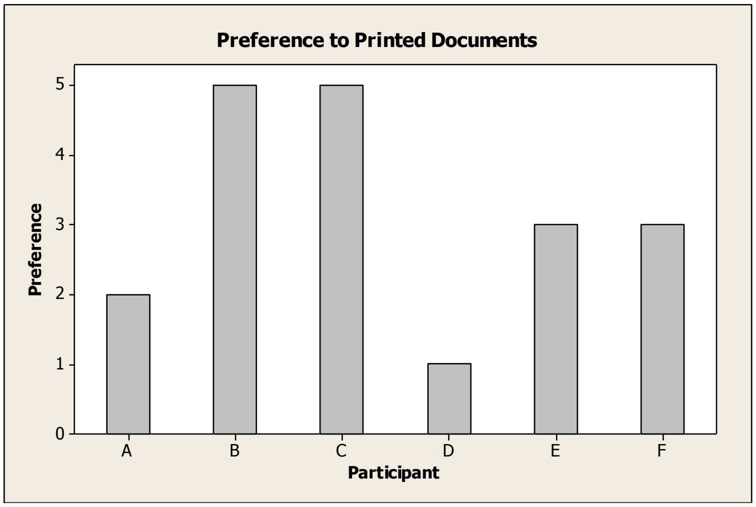


Figure 5: Preference to Printed Documents. Results for each individual tester are shown. Preference to printed documents indicates whether or not the participants prefer the lab application or printed documents.



After reviewing the survey results, the app was updated. Since the biggest issue was the keyboard appearing and causing the app to reset, this was addressed first. The code was changed so that the user was no longer able to edit the text, although they are still able to copy text for later use. Some participants were also confused by the first tab and what it was for. In order to make it clear that it was just a title page, a button was added that tells you to use the tabs below to navigate through the app.

Once the app was updated, a survey of seven biology majors was scheduled. In order to address certain information that didn't pertain to the non-biology participants, the survey was updated. The original five questions as mentioned before were kept, but three more were added. The survey asks if the application helped in completing the lab, if the device physically interfered with lab proceedings, and whether or not it would be possible to operate the device while wearing latex gloves and goggles. The following charts (Figures 6-13) give a visual representation of the first eight questions.

Comment [CCC20]: See Appendix....??)

Comment [CCC21]: Same comments as above

Figure 6: Ease Of Use. Results for each individual tester are shown. Ease of use is a representation of how easily the participants could make effective use of the application.

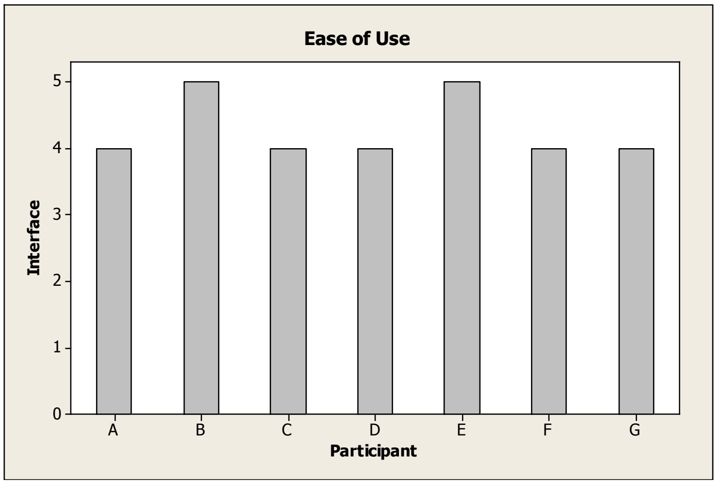


Figure 7: Ease Of Navigation. Results for each individual tester are shown . Ease of navigation is a representation of how easily the participants could make navigate through the separate portions of the application.

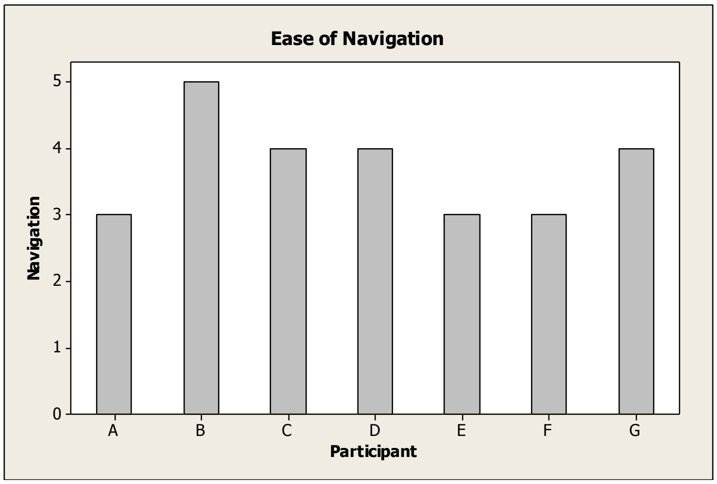


Figure 8: Aesthetically Pleasing. Results for each individual tester are shown. Aesthetically pleasing is a representation of how attractive the participants found the application.

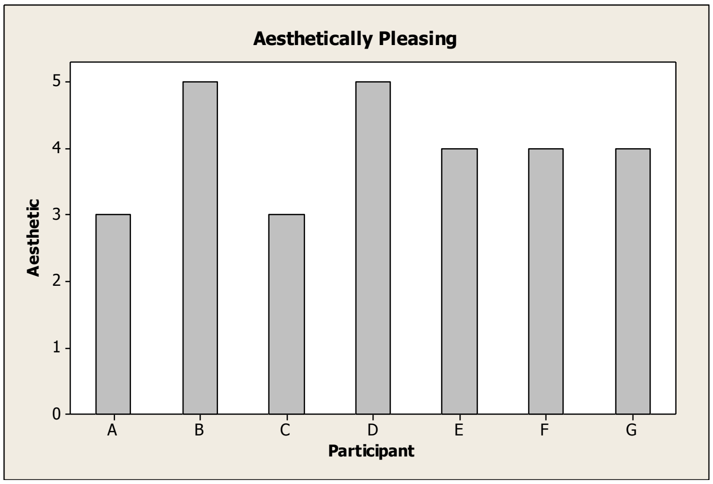


Figure 9: Relevance to Labs. Results for each individual tester are shown. Relevance to labs is a representation of how relevant the participants found the application to the performance of laboratory procedures.

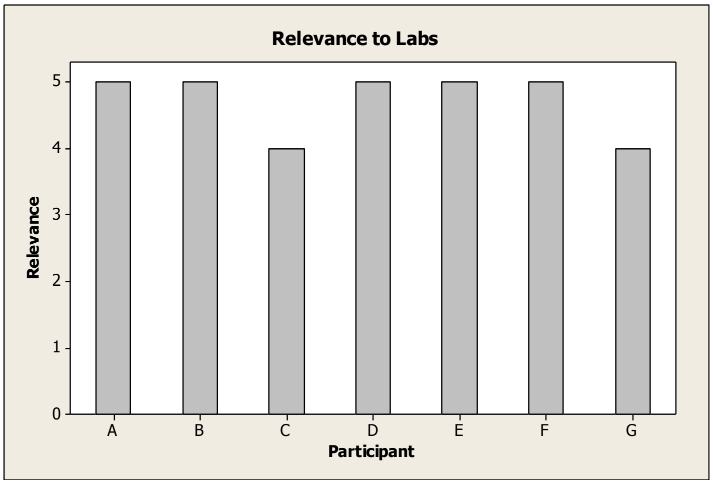


Figure 10: Preference to Printed Documents. Results for each individual tester are shown. Preference to printed documents indicates whether or not the participants prefer the lab application or printed documents.

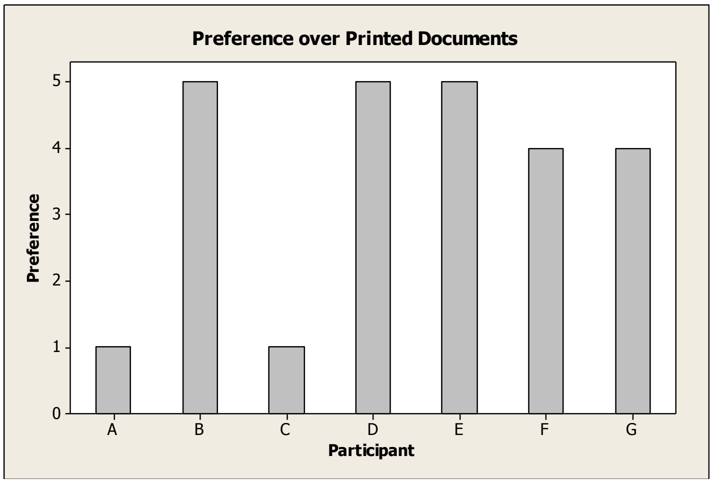


Figure 11: Helpfulness in Lab Completion. Results for each individual tester are shown. Helpfulness in Lab Completion indicates how helpful the participants found the application in completing the procedure.

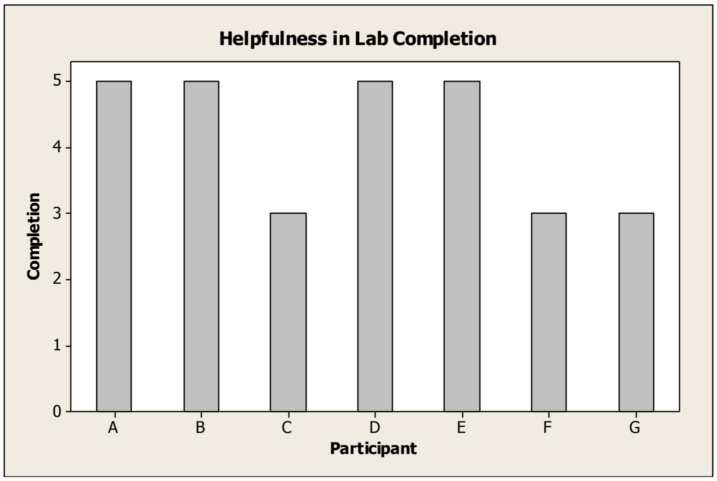


Figure 12: Possibility of Use without Interference. Results for each individual tester are shown. Possibility of use without interference indicates how much

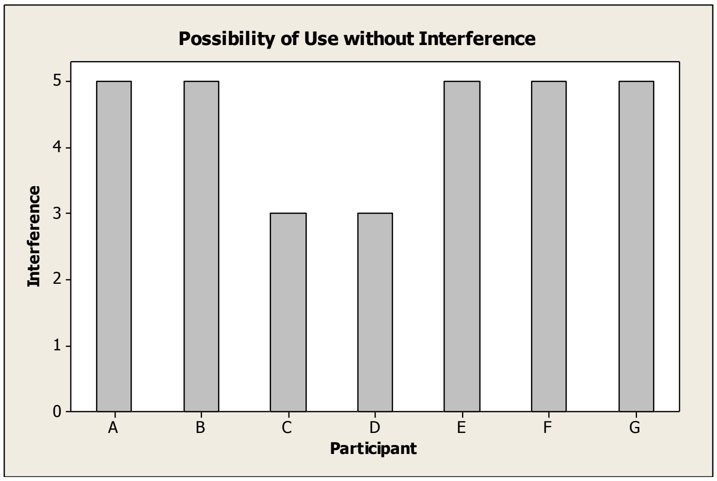
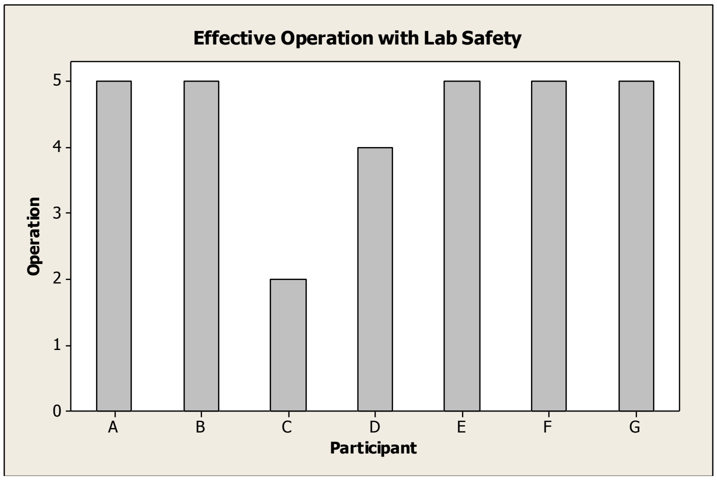


Figure 13: Effective Operation with Lab Safety. Results for each individual tester are shown. Effective operation with lab safety indicates how easily the application can be used while wearing latex gloves and safety goggles.



The results from the bio majors were much more favorable than the results from the non-bio majors. Although the two tests used slightly different versions of the application, they are similar enough to be able to compare the different answers from the test participants. The main changes involved the interface of the application, and the difference in results is clear from the two separate tests. The statistics in the following table (Table 1) were calculated using Minitab software. The table shows the average score for each category as well as the standard deviation.

Comment [CCC22]: But this is also a switch in the app, no? Am I confused here, or a re you in fact comparing apples and oranges??

Table 1: Results from Application Evaluation

Comment [CCC23]: Again with the significant figures. It was a Likert scale!!

	Bio Majors		Non-Bio Majors	
	Mean	Std Dev	Mean	Std Dev
Navigate	3.7	0.7	3.1	1.1
Interface	4.2	0.4	3.3	1.0
Aesthetic	4.0	.8	2.3	1.2
Relevance	4.5	.5	4.6	0.5
Preference	3.5	1.6	3.1	1.6
Completion	4.4	0.9	N/A	N/A
Interference	4.4	0.9	N/A	N/A
Operation	4.4	1.1	N/A	N/A

Discussion

There are a few conclusions that can be drawn from the responses of these lab tests. There are mixed reviews about the navigation of the application, but in general it seems that people did not find it very easy to use. Similarly, the interface was confusing to some. Aesthetically speaking, the app did not appeal to the survey participants. Relevance, however, received very high ratings, which indicates that the participants think the app would be helpful in a lab setting. Preference had the widest range of reviews, which is not surprising. Even if the app is easy to use, some students may still prefer using paper copies.

The last three questions of the survey were open response. The first asked what part of the app was difficult to use, the second asked what the participant liked about the app, and the third was features that should be included in the app that currently are not.

One of the most prevalent complaints received on the application was the fact that the lab procedure could be edited, and when the keyboard was brought up the application had to be reset. Most subjects also commented on the spacing and size of the text when it was displayed on the iPhone. Due to the small nature of the device, many of them suggested dividing up the procedure into smaller sections in order to allow room for larger text font. There were also complaints that navigating through the three different on the bottom of the app was not intuitive.

Comment [CCC24]: These comments are results of your survey and need to be reported in the RESULTS section. You can discuss those results here.

Generally speaking, most of the survey participants liked that the iPhone app eliminated the need for paper copies of the lab. It was also noted that including the lab safety procedures was helpful as well, as it is not a document that students normally bring to class. In addition, the survey participants were pleased that you could return to any point in the lab at any time as well as scroll through the text.

Each participant had multiple suggestions for ways to improve the app. A common theme was the inclusion of media in the application, i.e. being able to open and view videos. One participant suggested a reference sheet that includes information pertaining to various laboratory procedures. It was also suggested that a calculator be included in the app as a separate tab. It was also noted that more visuals, specifically pictures of the lab equipment and of a person performing the procedures, would be helpful

The data for the non-biology majors is not as consistent as the data for the biology majors. All of the standard deviations, excluding relevance, are over 1. This could be attributed to a few different factors. The participants of the first survey were from multiple different majors, so their opinions on how lab proceedings should work are most likely variable. They were also not using the device in a lab, which makes determining the usefulness of the app difficult. Preference has the highest standard deviation in both groups. This is to be expected, because even if a subject liked the app, they might still want to be able to use printed documents.

Reading the text was still an issue with the second test group. The two most prevalent complaints were that very little of the lab procedure could be seen at one time and that the boxes for the text were too small. This is an issue that will most likely continue to present itself due to the small nature of the device and its screen. Another concern was an inability to input data. The lab questions are displayed in the text, but there is currently no way to enter any answers. Hyperlinks to online resources cannot be directly clicked on in the app, they must be opened separately. This presented a problem for participants because they had to switch back and forth between the app and a web browser.

When asked what the participants liked about the app, almost all of them responded that it was simple and easy to use. The entirety of the app is fairly intuitive and even somebody with no experience with the device could operate it. Several participants also noted that it was more convenient to have than paper due its compact size and condensed procedure.

The most common suggestion for improving the app was to include a way to enter data, whether it is Google Docs or some other medium. It was also recommended that a way to take notes and highlight certain areas be added. One participant noted that being able to click on words such as "spectrophotometer" should bring up a popup screen that defines what it is as well as display its location in the lab.

When asked to compare the app to printed documents, there were mixed reviews. A few participants preferred the app, although it was mentioned that there is always the possibility of accidentally damaging the device. A few people were frustrated with the fact that you cannot take notes with the app, and as such they preferred printed documents.

The testing portion of the project was very informative. The first round of testing allowed the group to tackle any user errors and fix the application if there~~ir~~ were any flaws in the design. The results helped us find the iOS keyboard glitch and clarified the user’s confusion on the introduction tab. The second round of testing gave us a good input of the students who would be using this format in their future labs. The results proved fruitful, as the surveys showed that if they were given the choice between paper and mobile procedures, they would like to see the format in future labs.

Conclusion

The creation of this application took on various stages of design, coding, and resource searching to become a reality. Coding as a language is vast and ever growing, as more and more programmers are able to create and design almost anything they want. The process to learn how to effectively use the language to create what you want is painful at best. It is best described as "hitting your head against a wall until finally the answer is clear, either by someone else or a break through in your coding know-how." Unfortunately, the entire process was like this because of the group's lack of coding knowledge. A programmer with decent coding knowledge would be able to tackle the SDK with ease. Most of the programmer's adjustment would be with the layout of Xcode and the various attributes in the Inspector inside the Interface Builder [Appendices A-G].

Comment [CCC25]: Why are there quotes here? Are you just clarifying your lack of expertise here? Would a coder have these issues?

The design portion of the lab application involved a number of steps. The hardest step in our design was narrowing down a definition of what we wanted our application to do. The simplicity of this process was realized when the group decided to throw out the original proposal and start from scratch. The group wanted to do an application that did too much for what our project was trying to do. We wanted a solution to the mobile lab procedure without actually having the knowledge to do so. So instead the group came to the decision to test the usefulness and appeal of an iOS platform in a simple lab procedure. The design statement states our mission statement: "a tab bar application that organizes the lab material in a new mobile platform".

The coding portion of the lab application was done in various stages. The first stage was basic and included simple theory and knowledge of the various coding languages. This included online resources, print resources and meetings with some of the CS professors at WPI who are knowledgeable in mobile platform languages and procedures. The most helpful were the online tutorials and videos that went step-by-step, explaining the reasoning for their process. These resources were able to identify the problems the group was having in the coding, as opposed to most of the print, which would skim over the basics and not go into much detail about programming variety and what you can actually do with a sample program.

The resource searching for the lab application was a give-and-take of good and bad. The application wanted hasn’t been done by anyone yet, so most of the issues that arised were new and unfound on the Internet. For most Internet sources, the discussion was of sample applications and did not delve into any variations of the application at all. Instead it was more of a discussion of the process and coding understanding. Most discussion boards were over a year old and were done on older versions of the iOS SDK. The SDK sample applications provided by the Developer program at Apple were also out of date, leading the group to use intuition to bridge the gaps between the upgraded and older screenshots and procedures. All in all, the resources were key to the advancement of the application to its current status.

The application was created to start the introduction of mobile lab procedures in WPI lab sessions. The goal after this project was for another group to continue the process and create a series of lab applications that are on a mobile platform. This would entail a group with superior coding knowledge so that they can really make it user friendly and a more advanced application. The future goal is for WPI to actually implement and create an application that students can access when they sign up for the class. They will be able to have their lab information and procedures on their mobile phones and they can access it at anytime in order to prepare for the lab session.

Given the results of both surveys, the application certainly has potential for use in labs. There was a good amount of positive feedback from both groups, especially considering that it is just a prototype. There are still many things that can be added and improved upon, which is apparent from the open response questions. Based on the data that was gathered, it is fair to say that smart phone usage in the laboratory is a real possibility.

Refrences:

[1] " ADL Initiative." ADL Initiative. Web.
18 Jan. 2011.
<http://www.adlnet.gov/About/Pages/adlinitiative.aspx>.

[2] "Advanced Distributed Learning Network." The Technology Source Archives. Web. 18 Jan. 2011. <http://technologysource.org/article/advanced_distributed_learning_network/>.

[3]
<http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6W4X-4T2DM0Y-1&_user=74021&_coverDate=12%2F31%2F2008&_rdoc=1&_fmt=high&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=1412303172&_rerunOrigin=scholar.google&_acct=C000005878&_version=1&_urlVersion=0&_userid=74021&md5=d717b2aa9ef02ea9baf51a3d4a7eb39d#secx2)>.

[4] Ransford, Marc. "Smart Phones Displace Computers for More College Students." Ball State University - Home. Web. 18 Jan. 2011. <http://www.bsu.edu/news/article/0,1370,7273-850-64351,00.html>.

[5] "The ECAR Study of Undergraduate Students and Information Technology, 2009 | EDUCAUSE." What Is EDUCAUSE? | EDUCAUSE. Web. 18 Jan. 2011. <http://www.educause.edu/Resources/TheECARStudyofUndergraduateStu/187215>.

[6] "Young and Mobile: A Global View of Cellphones and Youth | Nielsen Wire." Web. 18 Jan. 2011. <http://blog.nielsen.com/nielsenwire/online_mobile/a-global-view-of-cellphones-and-youth/>.

[7] "Nielsen: US Smartphone Penetration to Be over 50% in 2011." GPS BUSINESS NEWS. 30 Mar. 2010. Web. 18 Jan. 2011. <http://www.gpsbusinessnews.com/Nielsen-US-Smartphone-Penetration-to-Be-over-50-in-2011_a2154.html>.

[8] "IPhone App Store Downloads Top 10 Million in First Weekend." Apple. Web. 18 Jan. 2011. <http://www.apple.com/pr/library/2008/07/14appstore.html>.

[9] "AppleInsider | Apple Says App Store Has Made Developers over $1 Billion." AppleInsider | Apple Insider News and Analysis. Web. 18 Jan. 2011. <http://www.appleinsider.com/articles/10/06/07/apple_says_app_store_has_made_developers_over_1_billion.html>.

[10] Goodwin-Jones, Robert "EMERGING TECHNOLOGIES MOBILE-COMPUTING TRENDS: LIGHTER, FASTER, SMARTER" Language Learning & Technology 12.3 (October 2008): 3-9. <http://llt.msu.edu/vol12num3/emerging.pdf> Web. 15 May 2009.

[11] "When It Comes To IPhone Games, What Sells Is Action, Adventure, and Arcade." TechCrunch. Web. 12 Mar. 2011. <http://techcrunch.com/2010/02/25/iphone-games-what-sells-distimo/>.

[12] "IStanford 2.0 Released." Stanford News. Web. 18 Jan. 2011. <http://news.stanford.edu/news/2009/march18/iphone-031809.html>.

[13] "BirdSight North America by Natural Guides, LLC". Web. 24 Jan. 2011. <http://itunes.apple.com/app/birdsight-north-america/id321285943?mt=8>

[14]. Apple Inc. AppleScript Language Guide. Cupertino, CA: Apple, 2008. Print.

[15]. Apple Inc. Cocoa Fundamentals Guide. Cupertino, CA: Apple, 2010. Print.

[16]. Apple Inc. "Manuals." Apple - Support. Web. 03 Feb. 2011. <http://support.apple.com/manuals/>.

[17]. Barnard, David. IPhone User Interface Design Projects. [New York]: Apress, 2009. Print.

[18]. Bondo, Joachim. IPhone Advanced Projects. [New York]: Apress, 2009. Print.

[19]. Dalrymple, Mark, and Scott Knaster. Learn Objective-C on the Mac. New York: Apress, 2009. Print.

[20]. Hillegass, Aaron. Cocoa Programming for Mac OS X. Upper Saddle River, NJ: Addison-Wesley, 2008. Print.

[21]. Mark, Dave, and Jeff LaMarche. Beginning IPhone Development: Exploring the IPhone SDK. Berkeley, CA: Apress, 2009. Print.

[22]. Mark, Dave, and Jeff LaMarche. More IPhone 3 Development: Tackling IPhone SDK 3. [New York, N.Y.]: Apress, 2009. Print.

[23]. Piper, Ian. Learn Xcode Tools for MAC OS X and IPhone Development. New York: Apress, 2009. Print.

[24]. Usman Ismail. IPhone SDK: TabBars:Programatically Creating Tab Bars [Internet]. Version 15. Knol. 2008 Oct 29. Available from: http://knol.google.com/k/usman-ismail/iphone-sdk-tabbars/34oprzanmpe7q/7.

[25]. UIPageControl - IPhone - Tutorial Base. Perf. Stefano Lo Duca. UIPageControl - IPhone - Tutorial Base. Vimeo 1 Year Ago 1 Year Ago: Sun, Dec 6, 2009 12:37pm EST (Eastern Standard Time), 6 Dec. 2009. Web. 13 Sept. 2010. <http://vimeo.com/8015346>.

[26]. IPhone SDK Articles. "UIWebView Tutorial." IPhone SDK Articles. 9 Aug. 2008. Web. 20 Dec. 2010. <http://www.iphonesdkarticles.com/2008/08/uiwebview-tutorial.html>.
Glyphish.com, Joseph Wain. IPhone App Icons. Digital image. Glyphish.com. Web. 20 Dec. 2010.

Appendix A: FirstViewController.h

```
//
//  FirstViewController.h
//  IQPgamma
//
//  Created by Alexander G. Orphanos on 11/4/10.
//  Copyright Worcester Polytechnical Institute 2010. All rights reserved.
//
#import <UIKit/UIKit.h>
@interface FirstViewController : UIViewController {
}

@end
```

Appendix B: IQPgammaAppDelegate

```
//  IQPgammaAppDelegate.h
//  IQPgamma
//
//  Created by Alexander G. Orphanos on 11/4/10.
//  Copyright Worcester Polytechnical Institute 2010. All rights reserved.
//
#import <UIKit/UIKit.h>
@interface IQPgammaAppDelegate : NSObject <UIApplicationDelegate, UITabBarControllerDelegate> {
    UIWindow *window;
    UITabBarController *tabBarController;
}
@property (nonatomic, retain) IBOutlet UIWindow *window;
@property (nonatomic, retain) IBOutlet UITabBarController *tabBarController;
@end
```

Appendix C: FirstViewController.m

```
FirstViewController.m
//  IQPgamma
//  Created by Alexander G. Orphanos on 11/4/10.
//  Copyright Worcester Polytechnical Institute 2010. All rights reserved.
#import "FirstViewController.h"
@implementation FirstViewController
/*
// The designated initializer. Override to perform setup that is required before the view is loaded.
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
    if ((self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil])) {
        // Custom initialization}
    return self;}
*//*
// Implement loadView to create a view hierarchy programmatically, without using a nib.
- (void)loadView {}
*//*
// Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];}
*//*
// Override to allow orientations other than the default portrait orientation.
- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
    // Return YES for supported orientations
    return (interfaceOrientation == UIInterfaceOrientationPortrait);}
*/
- (void)didReceiveMemoryWarning {
        // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];
        // Release any cached data, images, etc that aren't in use.}
- (void)viewDidUnload {
        // Release any retained subviews of the main view.
        // e.g. self.myOutlet = nil;}
- (void)dealloc {
    [super dealloc];}
@end
```

Appendix D: IQPgammaAppDelegate.m

```
//  IQPgammaAppDelegate.m
//  IQPgamma
//
//  Created by Alexander G. Orphanos on 11/4/10.
//  Copyright Worcester Polytechnical Institute 2010. All rights reserved.
//
#import "IQPgammaAppDelegate.h"
@implementation IQPgammaAppDelegate
@synthesize window;
@synthesize tabBarController;
#pragma mark -
#pragma mark Application lifecycle
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    // Add the tab bar controller's view to the window and display.
    [window addSubview:tabBarController.view];
    [window makeKeyAndVisible];
    return YES;
}
- (void)applicationWillResignActive:(UIApplication *)application {
    /*
     Sent when the application is about to move from active to inactive state. This can occur for certain types of
temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and
it begins the transition to the background state.
     Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should
use this method to pause the game.
     */
}
- (void)applicationDidEnterBackground:(UIApplication *)application {
    /*
     Use this method to release shared resources, save user data, invalidate timers, and store enough application state
information to restore your application to its current state in case it is terminated later.
     If your application supports background execution, called instead of applicationWillTerminate: when the user
quits.
     */
}
- (void)applicationWillEnterForeground:(UIApplication *)application {
    /*
     Called as part of  transition from the background to the inactive state: here you can undo many of the changes
made on entering the background.
     */
}
- (void)applicationDidBecomeActive:(UIApplication *)application {
    /*
     Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was
previously in the background, optionally refresh the user interface.
     */}
```

```
- (void)applicationWillTerminate:(UIApplication *)application {
    /*
     Called when the application is about to terminate.
     See also applicationDidEnterBackground:.
     */}
#pragma mark -
#pragma mark UITabBarControllerDelegate methods
/*
// Optional UITabBarControllerDelegate method.
- (void)tabBarController:(UITabBarController *)tabBarController didSelectViewController:(UIViewController
*)viewController {
}
*//*
// Optional UITabBarControllerDelegate method.
- (void)tabBarController:(UITabBarController *)tabBarController
didEndCustomizingViewControllers:(NSArray *)viewControllers changed:(BOOL)changed {
}
*/
#pragma mark -
#pragma mark Memory management
- (void)applicationDidReceiveMemoryWarning:(UIApplication *)application {
    /*
     Free up as much memory as possible by purging cached data objects that can be recreated (or reloaded from
disk) later.
     */
}
- (void)dealloc {
    [tabBarController release];
    [window release];
    [super dealloc];
}
@end
```

Appendix E: tabbartestAppDelegate.h

```
//
//  tabbartestAppDelegate.h
//  tabbartest
//
//  Created by MacBook Pro on 1/30/10.
//  Copyright Apple Inc 2010. All rights reserved.
//
#import <UIKit/UIKit.h>
@interface tabbartestAppDelegate : NSObject <UIApplicationDelegate, UITabBarControllerDelegate> {
    UIWindow *window;
    UITabBarController *tabBarController;
}
@property (nonatomic, retain) IBOutlet UIWindow *window;
@property (nonatomic, retain) IBOutlet UITabBarController *tabBarController;
@end
```

Appendix F: tabbartestAppDelegate.m

```
//
//  tabbartestAppDelegate.m
//  tabbartest
//
//  Created by MacBook Pro on 1/30/10.
//  Copyright Apple Inc 2010. All rights reserved.
//
#import "tabbartestAppDelegate.h"
@implementation tabbartestAppDelegate
@synthesize window;
@synthesize tabBarController;
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Add the tab bar controller's current view as a subview of the window
    [window addSubview:tabBarController.view];
    [window makeKeyAndVisible];
	return YES;
}
- (void)tabBarController:(UITabBarController *)tabBarController didSelectViewController:(UIViewController
*)viewController {
}
/*
// Optional UITabBarControllerDelegate method
- (void)tabBarController:(UITabBarController *)tabBarController didEndCustomizingViewControllers:(NSArray
*)viewControllers changed:(BOOL)changed {
}
*/
- (void)dealloc {
    [tabBarController release];
    [super dealloc];
}
@end
```

Appendix G: IQPgamma_Prefix.pch

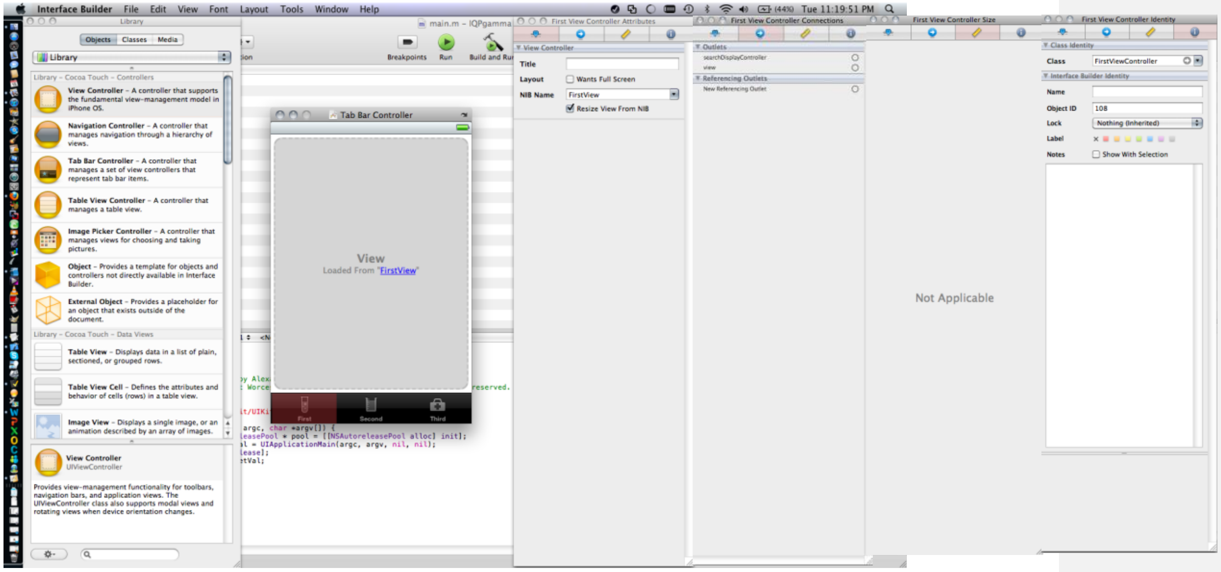
```
//
// Prefix header for all source files of the 'IQPgamma' target in the 'IQPgamma' project
//
#ifdef __OBJC__
    #import <Foundation/Foundation.h>
    #import <UIKit/UIKit.h>
#endif
```

main.m

```
//
//  main.m
//  IQPgamma
//
//  Created by Alexander G. Orphanos on 11/4/10.
//  Copyright Worcester Polytechnical Institute 2010. All rights reserved.
//
#import <UIKit/UIKit.h>
int main(int argc, char *argv[]) {
    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}
```

Appendix H: MainWindow.xib

H-1)Tab Bar Controller: View(loaded from "FirstView") and attributes from Inspector



Interface Builder File Edit View Font Layout Tools Window Help
(44%) Tue 11:19:51 PM
Library
Objects Classes Media
Library
main.m – IQPgamma
Breakpoints Run Build and Run
Tab Bar Controller
View
Loaded From "FirstView"
First Second Third
First View Controller Attributes
First View Controller Connections
First View Controller Size
First View Controller Identity
Not Applicable

Library – Cocoa Touch – Controllers

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

Tab Bar Controller – A controller that manages a set of view controllers that represent tab bar items.

Table View Controller – A controller that manages a table view.

Image Picker Controller – A controller that manages views for choosing and taking pictures.

Object – Provides a template for objects and controllers not directly available in Interface Builder.

External Object – Provides a placeholder for an object that exists outside of the document.

Library – Cocoa Touch – Data Views

Table View – Displays data in a list of plain, sectioned, or grouped rows.

Table View Cell – Defines the attributes and behavior of cells (rows) in a table view.

Image View – Displays a single image, or an animation described by an array of images.

View Controller
UIViewController

Provides view-management functionality for toolbars, navigation bars, and application views. The UIViewController class also supports modal views and rotating views when device orientation changes.

```
y Alex
 Worce
                                    reserved.
t/UIKi
 argc, char *argv[]) {
leasePool * pool = [[NSAutoreleasePool alloc] init];
l = UIApplicationMain(argc, argv, nil, nil);
lease];
tVal;
```

View Controller

Title

Layout ☐ Wants Full Screen

NIB Name FirstView

☑ Resize View From NIB

Outlets

searchDisplayController ○

view ○

Referencing Outlets

New Referencing Outlet ○

Class Identity

Class FirstViewController

Interface Builder Identity

Name

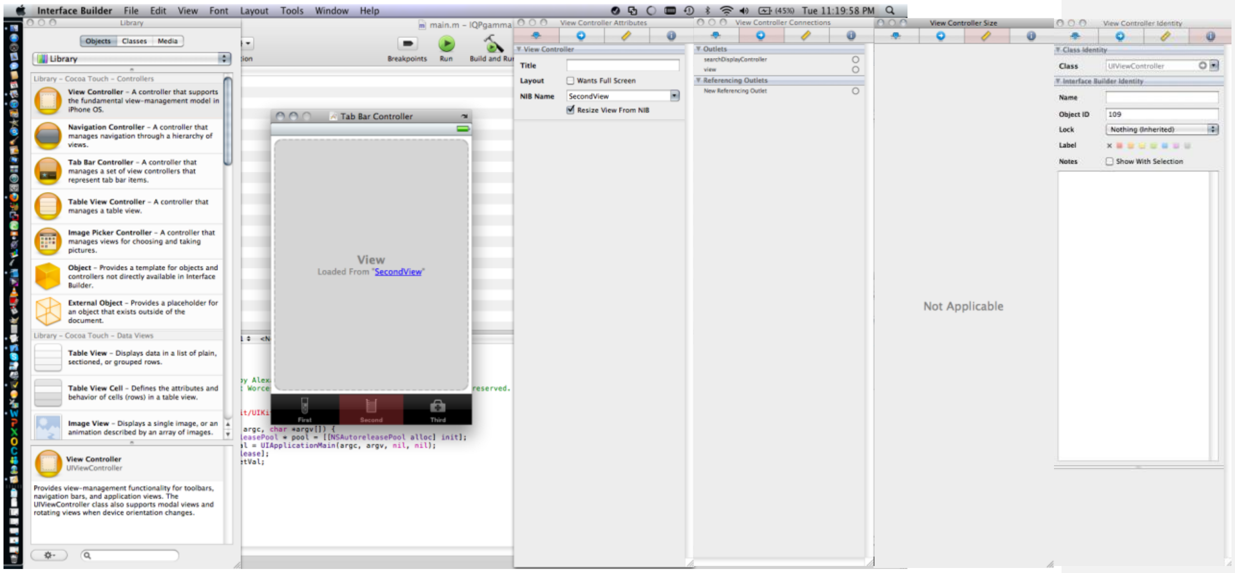
Object ID 108

Lock Nothing (Inherited)

Label ×

Notes ☐ Show With Selection

H-2)TabBar Controller: View(loaded from "SecondView") and attributes from Inspector



Library – Cocoa Touch – Controllers

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

Tab Bar Controller – A controller that manages a set of view controllers that represent tab bar items.

Table View Controller – A controller that manages a table view.

Image Picker Controller – A controller that manages views for choosing and taking pictures.

Object – Provides a template for objects and controllers not directly available in Interface Builder.

External Object – Provides a placeholder for an object that exists outside of the document.

Library – Cocoa Touch – Data Views

Table View – Displays data in a list of plain, sectioned, or grouped rows.

Table View Cell – Defines the attributes and behavior of cells (rows) in a table view.

Image View – Displays a single image, or an animation described by an array of images.

View Controller
UIViewController

Provides view-management functionality for toolbars, navigation bars, and application views. The UIViewController class also supports modal views and rotating views when device orientation changes.

```
y Alex
 Worce
                                    reserved.
t/UIKi
 argc, char *argv[]) {
leasePool * pool = [[NSAutoreleasePool alloc] init];
l = UIApplicationMain(argc, argv, nil, nil);
lease];
tVal;
```

View Controller

Title

Layout ☐ Wants Full Screen

NIB Name SecondView

☑ Resize View From NIB

Outlets

searchDisplayController ○

view ○

Referencing Outlets

New Referencing Outlet ○

Class Identity

Class UIViewController

Interface Builder Identity

Name

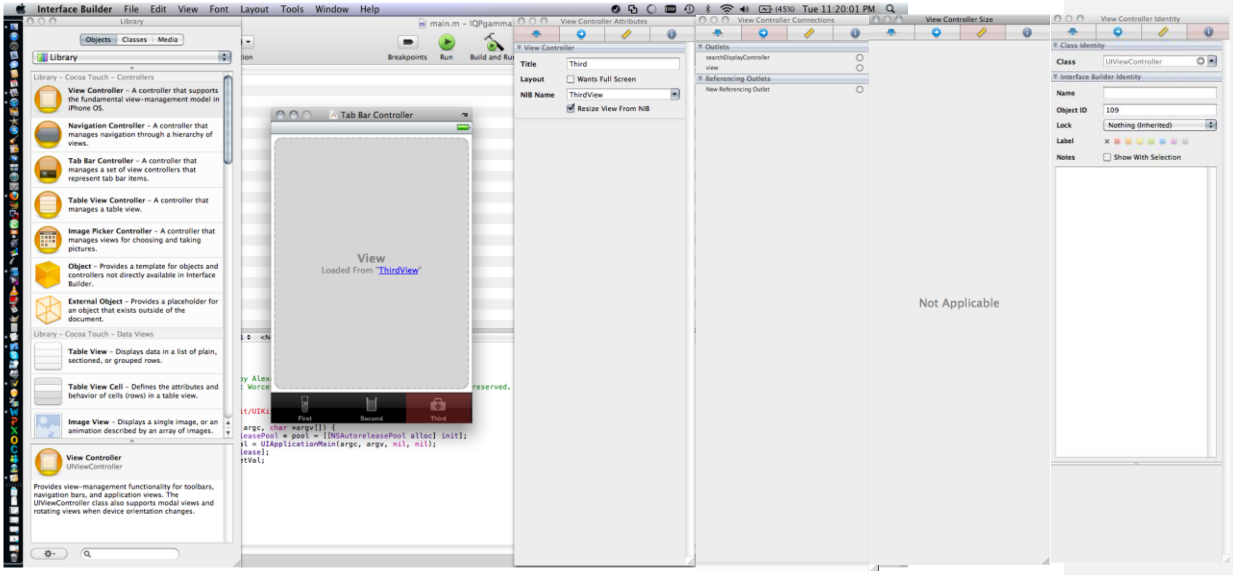
Object ID 109

Lock Nothing (Inherited)

Label ×

Notes ☐ Show With Selection

H-3) Tab Bar Controller: View(loaded from “ThirdView”) and attributes from Inspector



View Controller Attributes

Title: Third

Layout: ☐ Wants Full Screen

NIB Name: ThirdView

☑ Resize View From NIB

View Controller Connections

Outlets: searchDisplayController, view

Referencing Outlets: New Referencing Outlet

View Controller Size: Not Applicable

View Controller Identity

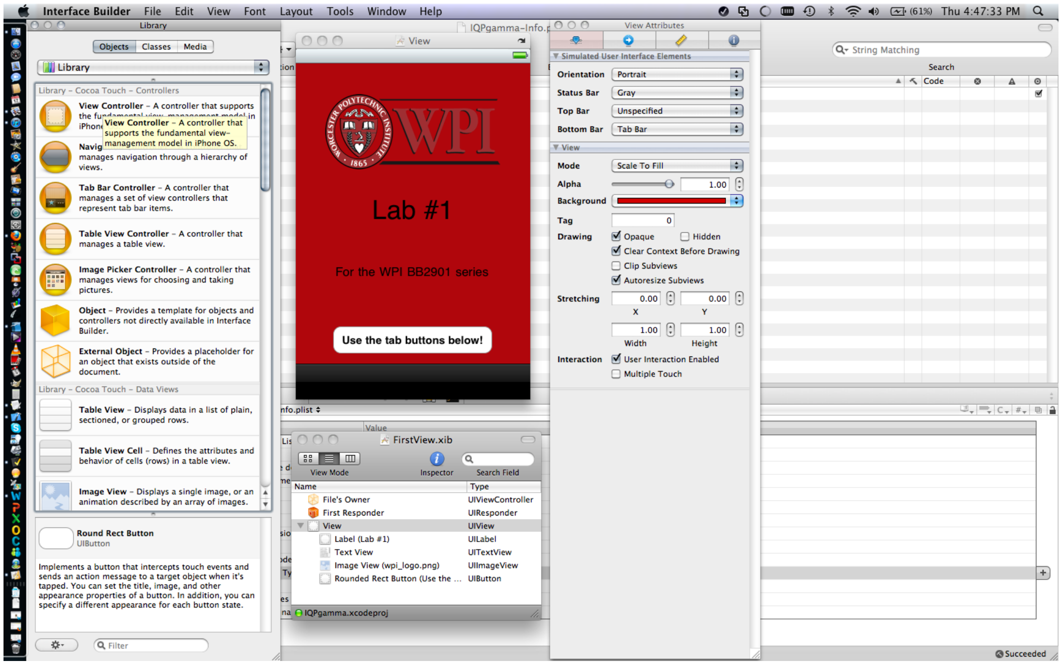
Class: UIViewController

Object ID: 109

Lock: Nothing (Inherited)

Notes: ☐ Show With Selection

H-4) FirstView.xib(view and UI properties)



View Attributes

Simulated User Interface Elements

Orientation: Portrait

Status Bar: Gray

Top Bar: Unspecified

Bottom Bar: Tab Bar

View

Mode: Scale To Fill

Alpha: 1.00

Tag: 0

Drawing: ☑ Opaque ☐ Hidden

☑ Clear Context Before Drawing

☐ Clip Subviews

☑ Autoresize Subviews

Stretching: X 0.00, Y 0.00, Width 1.00, Height 1.00

Interaction: ☑ User Interaction Enabled

☐ Multiple Touch

Lab #1

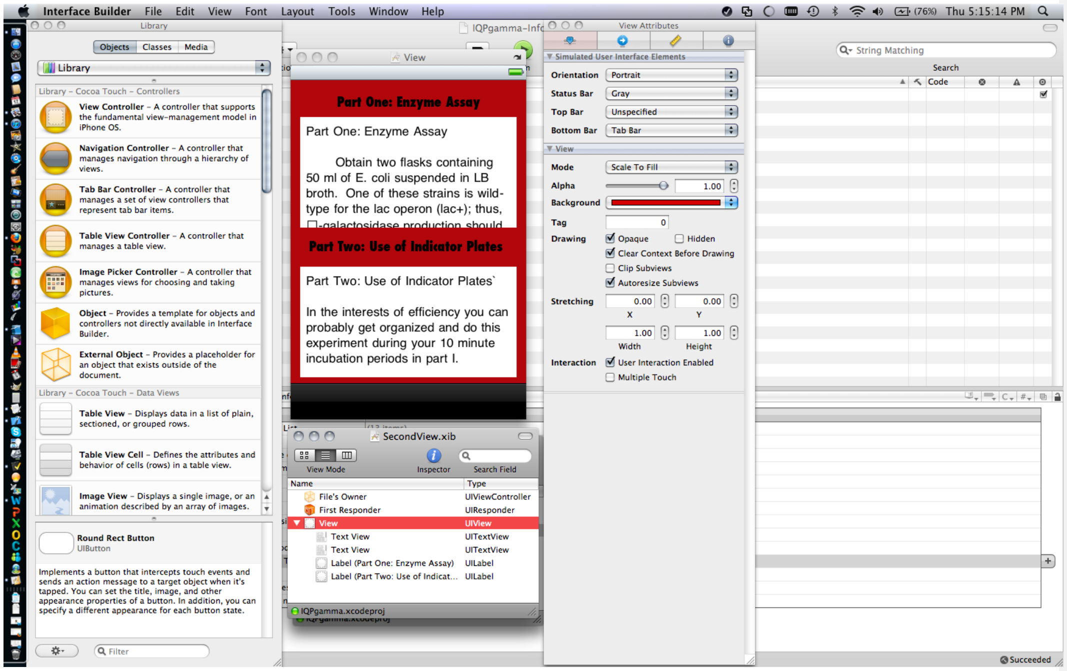
For the WPI BB2901 series

Use the tab buttons below!

FirstView.xib

Name	Type
File's Owner	UIViewController
First Responder	UIResponder
View	UIView
Label (Lab #1)	UILabel
Text View	UITextView
Image View (wpi_logo.png)	UIImageView
Rounded Rect Button (Use the ...	UIButton

H-5) SecondView.xib(view and UI properties)



Library – Cocoa Touch – Controllers

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

Tab Bar Controller – A controller that manages a set of view controllers that represent tab bar items.

Table View Controller – A controller that manages a table view.

Image Picker Controller – A controller that manages views for choosing and taking pictures.

Object – Provides a template for objects and controllers not directly available in Interface Builder.

External Object – Provides a placeholder for an object that exists outside of the document.

Library – Cocoa Touch – Data Views

Table View – Displays data in a list of plain, sectioned, or grouped rows.

Table View Cell – Defines the attributes and behavior of cells (rows) in a table view.

Image View – Displays a single image, or an animation described by an array of images.

Round Rect Button
UIButton

Implements a button that intercepts touch events and sends an action message to a target object when it's tapped. You can set the title, image, and other appearance properties of a button. In addition, you can specify a different appearance for each button state.

Part One: Enzyme Assay

Part One: Enzyme Assay

Obtain two flasks containing 50 ml of E. coli suspended in LB broth. One of these strains is wild-type for the lac operon (lac+); thus, □-galactosidase production should

Part Two: Use of Indicator Plates

Part Two: Use of Indicator Plates`

In the interests of efficiency you can probably get organized and do this experiment during your 10 minute incubation periods in part I.

SecondView.xib

View Mode · Inspector · Search Field

Name	Type
File's Owner	UIViewController
First Responder	UIResponder
▼ View	UIView
Text View	UITextView
Text View	UITextView
Label (Part One: Enzyme Assay)	UILabel
Label (Part Two: Use of Indicat…	UILabel

IQPgamma.xcodeproj

View Attributes

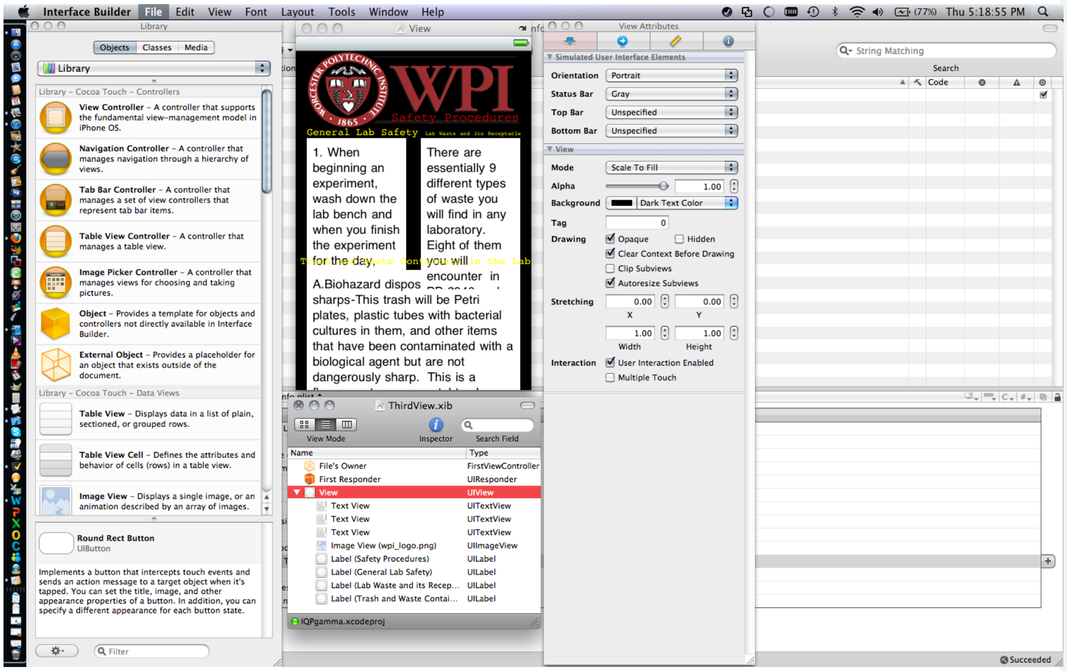
▼ Simulated User Interface Elements

Orientation	Portrait
Status Bar	Gray
Top Bar	Unspecified
Bottom Bar	Tab Bar

▼ View

Mode	Scale To Fill
Alpha	1.00
Background	🟥
Tag	0
Drawing	☑ Opaque ☐ Hidden ☑ Clear Context Before Drawing ☐ Clip Subviews ☑ Autoresize Subviews
Stretching	0.00 X 0.00 Y 1.00 Width 1.00 Height
Interaction	☑ User Interaction Enabled ☐ Multiple Touch

H-6) ThirdView.xib(view and UI properties)



Library – Cocoa Touch – Controllers

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

Tab Bar Controller – A controller that manages a set of view controllers that represent tab bar items.

Table View Controller – A controller that manages a table view.

Image Picker Controller – A controller that manages views for choosing and taking pictures.

Object – Provides a template for objects and controllers not directly available in Interface Builder.

External Object – Provides a placeholder for an object that exists outside of the document.

Library – Cocoa Touch – Data Views

Table View – Displays data in a list of plain, sectioned, or grouped rows.

Table View Cell – Defines the attributes and behavior of cells (rows) in a table view.

Image View – Displays a single image, or an animation described by an array of images.

Round Rect Button
UIButton

Implements a button that intercepts touch events and sends an action message to a target object when it's tapped. You can set the title, image, and other appearance properties of a button. In addition, you can specify a different appearance for each button state.

WORCESTER POLYTECHNIC INSTITUTE 1865 WPI

Safety Procedures

General Lab Safety

Lab Waste and its Receptacle

1. When beginning an experiment, wash down the lab bench and when you finish the experiment for the day,

There are essentially 9 different types of waste you will find in any laboratory. Eight of them you will encounter in

Trash and Waste Containers in the Lab

A.Biohazard dispos sharps-This trash will be Petri plates, plastic tubes with bacterial cultures in them, and other items that have been contaminated with a biological agent but are not dangerously sharp. This is a

ThirdView.xib

View Mode · Inspector · Search Field

Name	Type
File's Owner	FirstViewController
First Responder	UIResponder
▼ View	UIView
Text View	UITextView
Text View	UITextView
Text View	UITextView
Image View (wpi_logo.png)	UIImageView
Label (Safety Procedures)	UILabel
Label (General Lab Safety)	UILabel
Label (Lab Waste and its Recep…	UILabel
Label (Trash and Waste Contai…	UILabel

IQPgamma.xcodeproj

View Attributes

▼ Simulated User Interface Elements

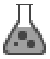
Orientation	Portrait
Status Bar	Gray
Top Bar	Unspecified
Bottom Bar	Unspecified

▼ View

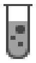
Mode	Scale To Fill
Alpha	1.00
Background	■ Dark Text Color
Tag	0
Drawing	☑ Opaque ☐ Hidden ☑ Clear Context Before Drawing ☐ Clip Subviews ☑ Autoresize Subviews
Stretching	0.00 X 0.00 Y 1.00 Width 1.00 Height
Interaction	☑ User Interaction Enabled ☐ Multiple Touch

H-7) Icons (Glyphish.com)

50-beaker.png



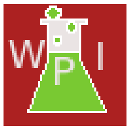
91-beaker-2.pmg



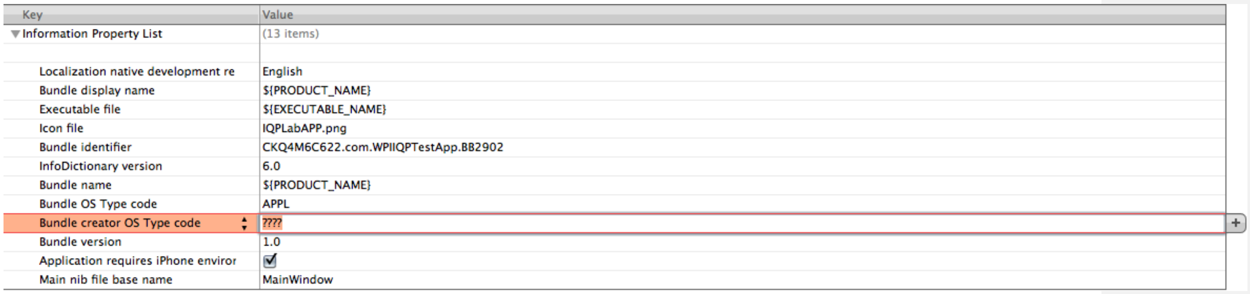
92-test-tube.png

79-medical-bag.png

H-8) IQPLabAPP.png



H-9) IQPgamma-Info.plist



Key	Value
▼ Information Property List	(13 items)
Localization native development re	English
Bundle display name	${PRODUCT_NAME}
Executable file	${EXECUTABLE_NAME}
Icon file	IQPLabAPP.png
Bundle identifier	CKQ4M6C622.com.WPIIQPTestApp.BB2902
InfoDictionary version	6.0
Bundle name	${PRODUCT_NAME}
Bundle OS Type code	APPL
Bundle creator OS Type code	????
Bundle version	1.0
Application requires iPhone enviror	☑
Main nib file base name	MainWindow

H-10) wpi_logo.png