Mobile Application Development Project

CIS 657 – Spring 2016

Final Due Date: June 21, 2016 @ 6pm

Objective

The objective of this project is to give students first hand experience creating, implementing and publishing a non-trivial fully functional mobile application on the iOS platform.

Pairing Option

Students are allowed to work in pairs on this project. While the overall requirements are the same whether you work solo or as a pair, the actual scope of the final deliverable should be more significant if you are working together on the project.

Coming up with an app idea

One of the more challenging (and hopefully fun!) aspects of this project is that you are to come up with your own application idea. While there are general requirements your application must meet (see details below) you are free to innovate and design a mobile experience that you are personally interested in and passionate about. Here are a few suggestions to help facilitate the creative process:

- Browse iTunes AppStore[1] and search for apps related to areas you are interested in. You can use your iOS device or borrow a friend's or family member to try apps out. In lieu of an actual device, you'll also find plenty of video reviews that show apps in action.
- Participate in app brainstorm sessions in the classroom (more details will be provided.)
- Consider looking for mobile app use cases at your place of employment. Do make sure if you take this route you are still able to submit your source code to your instructor for assessment purposes!
- Visit blogs, etc. where new mobile apps are often reviewed and discussed.
- Follow people on twitter who tweet about mobile and have interests similar to your own.
- Ask acquaintances what they think a "killer app" idea might be, though you don't necessarily need a killer app to survive this course!

You can review some of the prior projects students have completed for this course here (scroll down to the heading "Course Projects":

http://masl.cis.gvsu.edu/our-apps/

[1] You will need to install Apple's iTunes software on your computer in order to browse the iTunes AppStore. http://www.apple.com/itunes/. Accessed on 5/10/16.

Note that the requirements of these past projects vary somewhat from the requirements below, but this should give you a feeling for the scope of the app you will develop.

Minimal app requirements

Your app must meet the requirements described below at a minimum for a passing grade. *If you want a stellar grade (e.g. an A) you need to go well beyond the minimal requirements.*

Application Requirements:

There are a number of requirements that your app must meet.

Innovation Requirement:
Your app must be original and creative. It is ok to come up with a better mousetrap, but don't just replicate an app that already exists.

Base Requirements:
Base requirements are essential requirements that the typical non-trivial app will implement. Your application must implement *all* of the following base requirements:

- The application must have a fully functional and well-designed foreground user interface.
- The application must persist data between user sessions via local storage on the device, or via web services.
- The application must be stable and not crash frequently.

App Integration Requirement:
iOS apps use URL Schemes to support inter-application communication/interaction. In order to meet the application integration requirement on iOS your app must support *one* of the following features:

- Use an app extension of an existing app.
- Have your app implement an app extension of its own.
- Use the URL scheme of an existing app to integrate it with your app.
- Have your app implement a custom URL scheme of its own.

Network Integration Requirement:
Mobile apps inevitably use the device's networking capabilities to integrate with the world beyond the device itself. Your application must support a minimum of one of the following *network integration requirements*.

- Directly integrate with a data store in the network via web services. This can be an existing data store from another service, or something you have implemented/deployed yourself.

- Frame existing web content in a seamless way within your app. (note: the content should be rendering for small screens!).
- Integrate with a popular social media platform using the Social Framework.

Advanced Integration Requirement:
A number of interesting more advanced features are enumerated below. Your application must support a minimum of *one* of the following *advanced integration requirements*:

- Directly integrate an interactive map.
- Support a status bar notification via a background service that launches your app when acted upon by the user.
- Support a non-trivial[2] location aware feature.
- Non-trivial integration of 3D graphics via OpenGL ES.
- Support a non-trivial multi-device use case, e.g. multiple devices with your app installed collaborate in real-time.
- Integrate the onboard camera.
- Integrate with any commonly available sensors, such as the accelerometer, magnetometer, proximity sensor or ambient light sensor.
- Integrate with the Bluetooth radio.
- Any other platform features you want to integrate to meet this requirements should be approved by the instructor in advance.

Third Party Integration Requirement:
There are numerous third party components, libraries, etc. available as open source projects to iOS developers. As discussed in class, CocoaPods (https://cocoapods.org/) is the de facto standard dependency management system for iOS. For this project, you are to identify and integrate two different pods into your project that enhance your app in a meaningful way.

Testing Requirement:
As taught in lecture, XCode integrates a variety of unit and UI testing features. Your final project should make good use of the unit test capabilities Apple provides. Try to write your code in a way that gives you maximum test coverage. You may also utilize the UI testing capabilities, though this is optional.

Extra Credit Challenge - Deployment Requirement

In order to meet this challenge you will have to publish your app in the Apple iTunes AppStore for *external beta testing* at minimum. Do note that the AppStore is curated. This means Apple engineers will manually vet your app submission for quality purposes prior to allowing it to appear in the AppStore. Assuming the process goes without any issues (probably a stretch when trying for the first time), you should allow approximately a week for approval. In addition, you must be a registered iOS developer ($99 annual fee) in order to submit the app. Students who deploy apps will receive a 10% extra

[2] Use common sense when interpreting the non-trivial qualifier. If you spend a sum total of 30 minutes to code up the feature that meets the given requirement, it's quite likely that would be a trivial feature.

credit adjustment to their final project grade. For example, if you earned a score of 76 on your project, you would receive a 76 + 7.6 ~ 84.

Project Deliverable Timeline

May 17: One-page app description, submitted to the instructor via Blackboard. The instructor will review and approve the description, or contact you for further discussion if the app description does not appear to have the potential to meet the app requirements. Note that you do not need a detailed description of how you will meet every one of the above requirements, as this will take some time. The goal here is to make sure the direction you are taking at the outset is likely to lead you to a successful conclusion.

May 24: First cut of your app's wireframes in order to participate in an in class design critique exercise.

June 21: Lightening Talk - Deliver a compelling 5 minute presentation of your app to the class during the final exam period. More details on the lightening talk will be provided by the instructor as that time draws closer.

June 21: A final written report, copy of lightening talk presentation[3], and final code all submitted to the instructor via Blackboard. Your final report should consist of the following:

- A brief description of your application. This could be a refined version of the app description you submitted on May 17 that incorporates any changes you made along the way.
- Your final wireframe should include any improvements made based on feedback received during the in class peer reviews.
- A short and precise description of how your app meets each of the above requirements. If you were unable to meet a particular requirement, simply indicate that the requirement was not met.
- A link to your app's entry in the AppStore if you implemented the extra credit challenge.

[3] If you prepared viewgraphs for your lightening talk, a PDF copy of the viewgraphs is what is meant here. If you demo'd your app, you could also submit a video (or link to a video) which is a screen capture of your demo.

Grading Rubric

Grading will follow the rubric given in Figure 1 below.

Requirements	Points	Pts Earned
Innovation Requirement:		
- Originality / Creativity	20	
Functional Requirements:		
- Base Requirements	20	
- Application Integration Requirement	15	
- Network Integration Requirement	15	
- Third party Integration Requirement	10	
- Testing Requirement	10	
Final Presentation / Report		
- Final Report	5	
- Final Presentation to the Class	5	
Total Points	**100**	

Figure 1. Project Grading Rubric