

Mobile Application Development Project

CIS 357 – Fall 2021

Final Due Date: End of the day December 10, 2021 (Individual viewing/rating of peer projects due on Blackboard discussion board by end of day December 14).

Objective

The objective of this project is to give students an opportunity to demonstrate their ability to acquire in-depth technical proficiency on a native mobile platform in an area that is not directly addressed in lecture / homework, and the ability to creatively apply the knowledge gained in a fully functional demonstration app and communicate what they've learned via a written coding tutorial / presentation.

Overview

Students will **optionally self-organize into groups of 3 or less¹** for the semester project. Each group of students must identify which platform their project will work on (Android or iOS), and which focus area they wish to study. The project involves acquiring in-depth proficiency in the focus area, by reading platform documentation and relevant books, online tutorials, etc. Students will identify a demo app idea that clearly generates the functionality of their focus area in a creative and realistic use case. Finally, students will author a coding tutorial on their focus area that helps other developers get up to speed on that aspect of the platform. In addition to publishing the tutorial online, each pair will give a short overview presentation to the class on what they have learned and their demo app.

Group Formation Instructions

Students are free to self-organize into project groups of three or less students. If you would like to work in a group but do not know who else might be willing to work with you, feel free to start a conversation thread in the #project channel on Slack created for this purpose. To recruit other students it might be helpful to mention your platform preference and topic interest if you have any.

Once you've formed a group, have one person in your group create a private channel that consists of the last names of each team member of the group separated by hyphens. For example, if Jane Doe, Juan Perez and Zhang San choose to work together, their group name would be group-doe-perez-san. Add each member of the group to the channel along with the instructor. Then post which platform focus area your group has chosen to work on.

¹ Students who prefer to work solo on the project may do so. Project requirements remain the same whether you complete a solo project or a group project.

If you choose to work solo, indicate your intent to work solo in a direct message to the instructor on Slack. Include in the direct message which focus area you have chosen to work on. Note that you can form a group with anybody in the course - you do not need to limit your groups to students that have been meeting on the same day you do.

Identifying the Platform Focus Area

Each group of students or solo student will pick **one** of the focus areas below for their project, and only one group or solo student will be allowed to study each given focus area. Focus areas will be allocated on a first-come-first-serve basis. Once you've become part of a group or decided to work solo you can post your focus area selection as described in the previous section. I will cross out topics as they are selected below. Select topics from the lists below:

Android Topics:

[ML Kit](#) / [TensorFlow](#) (irons)
[Data Storage](#) (on device options) (trapp)
[Wear OS](#) (kaip-rice)
[Google Fit](#) (olosimo-saunders-fang)
[ARCore](#) (davis-rockow)
[Game Development](#) (hoeve-vanbronkhorst)
[OpenGL ES](#) (herrera-santos)
[Maps SDK for Android](#) (omwenga)
[Location and Context APIs](#) (champoux)
[Motion sensors](#) (lobbestael)
[Camera](#) / [CameraX](#) (colosimo)
[Google Play Billing Library](#) / [Google Play In-App Billing](#)
[Google Mobile Ads SDK](#)
[Google Flutter](#) (arnott-cain-garret)
[Notifications](#) / [Firebase Cloud Messaging](#) (ruiz)
Android Launcher

iOS Topics:

[Core ML](#)
[Core Data](#)
[Watch Kit](#) (albaitus-stone)
[Health Kit](#)
[ARKit](#) (van-dam-york)
[GameKit](#)
[SceneKit](#) / [SpriteKit](#) / [Metal](#)
[MapKit](#) (clinthorne-sweet-shamburger)
[Core Location](#) (dykema)
[Core Motion](#)
[Camera and Media Capture](#)

Doc. Rev.1.2 28-September-2021

[StoreKit](#)
[Google Ads SDK](#)
[SwiftUI](#)
[PushKit](#) / [Apple Push Notification Service](#)

Coming up with a demonstration app idea

As you start to become technically proficient in your chosen focus area, start to think of a *creative and realistic* app scenario that you will implement to demonstrate the in-depth area of the platform you are studying. It is acceptable to use other platform features as well (e.g. if you are demoing Core Location, you might want to use MapKit as well) but keep the focus of your demo app on the area you have chosen. Avoid creating a “Swiss army knife” demonstration app!

Writing your tutorial

As you develop proficiency in your focus area, author a coding tutorial that will help peer developers quickly come up to speed. You should assume the target audience is proficient in general on the platform you are working on, but has no experience in the focus area you have studied. Your tutorial should consist of the following sections:

1. **Overview.** Description of the platform focus area you studied as well as the demo app your reader will be implementing as they follow your tutorial. Use screenshots and/or video clips liberally to help your reader understand the proposed functionality of the demo app.
2. **Getting started.** Describes the software development environment, the reader will need on his/her machine to complete your tutorial, including things like IDE version, dependencies, and jump off links on where to download, install tutorials, etc.
3. **Step-by-step coding instructions.** This could be subdivided into multiple sections as you see fit, but basically this will be the “meat” of your tutorial. It will consist of code snippets, text discussing the code, figures / videos of perhaps concepts that are hard to describe in text / code.
4. **Further Discussion/Conclusions.** Summary of your tutorial, including any other alternative approaches developers might pursue to implement the same functionality (e.g. third party components that can be used in lieu of the native features you have studied) related platform features the reader might want to refer to for further study, as well as a link to a github repo of your complete source code.

Each project group will post their complete tutorial publicly on [GitHub Pages](#) when complete. Alternatively, you can use the wiki on your GitHub repo as long as you make it publicly viewable.

Example Final Presentations / Demos

Here are a few final presentations / demos produced by students in recent semesters to give you an idea of the scope of the project.

- Example Project #1: [Motion Sensors \(Android\)](#)
- Example Project #2: [Mobile Gaming with Unity \(Android\)](#)
- Example Project #3: [HealthKit \(iOS\)](#)

Final Project Submission Requirements

Please see [project presentation and submission](#) guidelines for more details on the final presentation requirements and actual submission. Here is a summary:

- A link to the GitHub repo of your project must be submitted via Blackboard by the project due date. Your project MUST be self-contained and build without further manual intervention by the instructor. 10 points will be automatically deducted from any project that does not build “out of the box”. Groups are free to use whatever workflow paradigm they wish in managing their code as the project progresses, but do make sure your final commits are all reflected on the *main* branch, so the *Insights* -> *Contributors* graphs accurately represent individual effort. It is important that every member of the team makes meaningful code commits to the repo. If you are pair programming, make sure each team member has a chance to “drive” and commit code under his/her identity. *It is not acceptable for one person in the group to do all the work on the project!*
- A link to your final tutorial published on GitHub Pages.
- A link to your final presentation Google Slides.

Project Deliverable Timeline

10/20: Identify Groups and chosen focus areas via Slack as instructed above. Focus areas are assigned on a *first-come-first-served* basis, so don’t delay in figuring out what you want to work on!

10/27: Submit a short paragraph description of your proposed demo app idea on your group’s private slack channel, or in a direct message if working solo.

12/10: Final project submission (code + tutorial + presentation) is due end-of-day.

Exam Week: View and rate the other project submissions posted on the Blackboard discussion board. Must be completed by End-of-day 12/14.

Doc. Rev.1.2 28-September-2021

Grading Rubric

Your work will be graded on the overall quality/creativity of your work and whether or not it meets the documented requirements. Grading will follow the rubric given in Figure 1 below.

Requirements	Points	Pts Earned
Demo App		
- Creativity / Breadth of Coverage in Focus Area	20	
- Authenticity	15	
- Implementation	20	
Tutorial		
- Overall organization of tutorial	5	
- Sufficient level of detail for intended audience	10	
- Use and quality of code snippets and visuals (figures, video)	10	
- Formatting / general readability and correctness of tutorial	10	
Final Presentation		
- Final Presentation to the Class / Viewing & Rating peer projects	10	
Total Points	100	

Figure 1. Project Grading Rubric