

FINAL PROJECT REQUIREMENTS

MPCS51033 - AUTUMN 2019

Application

All applications will be graded on implementing and complying with the following specifications, as well as the implementation of their specific functionality. The completeness of the submitted backend and application is the most important objective, not necessarily a full realization of the developers vision.

While the focus of this project will be on the backend functionality of your application, your companion mobile applications is expected to look and behave in a manner that is consistent with an application that would be "App Store" ready.

It would be better to submit a polished app with less features, than fully featured app that has usability issues.

Marketing Materials

Each student should prepare marketing materials for their application to submit along with the application code. These should include the same information that is required for an App Store submission:

- ☐ Define a category
- ☐ Description (2000 character max)
- ☐ Description (2000 character max)
- ☐ Keywords (100 characters)
- ☐ Screenshots (up to 5 per device)
- ☐ Define age group

Executive Summary

Provide a (no more than) one page summary of the application and its features.

- ❑ Discuss the backend solution that you selected for your application and the key considerations that guided your decision.
- ❑ Discuss the problem that your app solves for the user and how it fits into the App Store landscape.
- ❑ Include a high-level overview of the technical details about the app and the backend. For example, this app collects location data based, aggregates it on a server and broadcasts it to a group of predefined users.

Backend Application

The backend application should interact with the mobile application in a significant way to realize the functionality of the application. The exact functionality is up to you, but as a guideline, the application should not be able to fully function without the backend. The backend should be deployed and fully functional with the mobile application.

- ❑ Provide as much code as possible for your backend solution.
- ❑ List all URLs where the service is hosted in a file named `BACKEND.md`.
- ❑ Provide a list of the API calls for your backend code in the `BACKEND.md` file (if you are using your own service). You do not have to do this if you are exclusively using CloudKit or Firebase.
- ❑ You must use appropriate authentication for your backend services. If you write your own, you can define what is appropriate.

Mobile Application

- ❑ All applications should compile with no warnings.
- ❑ Applications should show a splash screen with (at least) the developer's name and the name of the application. This should appear when your application launches.
- ❑ Applications should behave appropriately (for your application) under all multi-tasking states:
 - Launching
 - Transition to background

- Being awoken
- Being terminated
- Applications should provide instructions for use (e.g. splash screen, info button, etc.) somewhere in the application.
- The application should operate under all normal connectivity conditions and if necessary, it should degrade functionality if appropriate. For example:
 - Present an alert notifying user that a connection cannot be made
 - Handle airplane mode
- Applications should have an icon.

Data

Data should be structured and handled in a manner that is best suited for your application. There are no specific requirements for how to structure your data, but consider best practices discussed in class.

Please consider the following for your application (not all will apply):

- Data should be cached for increased mobile application performance
- Data should be structured to optimize for the backend platform you choose
- Local data should be mirrored to cloud data

Code Comments

Both the backend and mobile applications should be verbosely commented so as to be understood by a third party. The following rules should be applied throughout the code:

- Each custom class should include a short description of its functionality
- Any custom methods you write should be self descriptive or include a comment description. Methods that are provided by the SDK (eg. `viewWillAppear`, `viewDidLoad`, `didReceiveMemoryWarning`;) do not need to be commented.

Logging

- Any data that is important to the normal flow of your application should be logged out to the console. For example:
 - File paths and URLs the app reads or writes to
 - Request/Responses from your backend service
 - Responses from third party services
 - URLs for data downloaded from the Internet

Performance

- All applications will be tested on a device, not on the simulator. If possible, please test applications on devices to detect any performance issues.
- Utilize techniques for handling large operations in background threads. For example:
 - Requesting data from Internet
 - Image transformations

Third-Party Frameworks

Third-party frameworks are allowed in the final projects, although they should not contribute the majority of the functionality of the application. You should be prepared to explain how the framework works in great detail if requested.

- Document any frameworks you use in comments and in a README.md file located in your project bundle.
- All applications should compile with no warnings. This includes warnings from third-party frameworks you are using.

Due Date

Final projects are due December 1st at 11:59 PM (CST). Commit all files to your final project Github repository. Include the marketing materials in the git repo in a folder named "MarketingMaterials" and use the following naming convention.

- *github-username_marketing.pdf*
- *github-username_executive.pdf*