FINAL PROJECT REQUIREMENTS

MPCS51030 - Winter 2022

Application

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

It would be better to submit a polished app with less features, then a fully featured app with no polish. Applications receiving the highest grade will be those that are "ready for the App Store".

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 summary (one page max) of the application and its features.
- ☐ Discuss the problem that it solves for the user and how it fits into the App Store landscape.
- ☐ If the app is for private company use, define its utility within the organization.
- ☐ Include a high-level overview of the technical details about the app. *For example, this app collects location data based, aggregates it on a server and broadcasts it to a group of predefined users.*

Code

- ☐ All applications should compile with no code warnings. Layout warnings will be acceptable as long as it does not impact the user interface.
- ☐ It is your responsibility to ensure that you application can be downloaded from GitHub and run. Make sure that all of your assets and other dependencies are included in your project.

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 the 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 code warnings. This includes warnings from you implementation of the third-party frameworks. If there are known warnings from the third-party code base (eg. from a GitHub issue, bug tracker, etc.) the warning will be acceptable as long as they don't impact the applications functionality. Please provide a link to the know issue in the README.md file.

Code Comments

All 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 succinct description of its functionality.
- ☐ Any custom methods you create should include a succinct description. You do not need to document all parameters. *Methods that are provided by the iOS SDK (eg. viewWillAppear, viewDidLoad, didReceiveMemoryWarning:) do not need to be commented.*

Logging

- ☐ Any critical data that is important to the flow of your application should be logged out to the console. This allows us to to see what is happening in your application since we are unfamiliar with the data you are using. For example:
  - All file paths and URLs the app uses to read or write
  - Any target-action methods (e.g. a button pressed)
  - Any URLS for data downloaded from the Internet
  - A dump of data retrieved from a URL

Splash Screen

- ☐ Applications should show a splash screen including the developer's name and the name of the application. To be "App Store Ready" this screen should be styled to match your application.
- ☐ The splash screen should appear when your application launches. This is not the same as the LaunchScreen.storyboard. It is an additional screen that shows.
- ☐ The implementation details of it are up to you, but it should dismiss after a short predefined time (~1s) or when the user taps on it.

Application Multi-tasking Operation

- ☐ Applications should behave appropriately (for your application) under all multi-tasking states:
  - Launching
  - Transition to background
  - Being awaken
  - Being terminated

Application Use Instructions

- ☐ Applications should provides instructions on how to use the application.
- ☐ The implementation details are to you, but consider the following:
  - When would the user be most likely to need the instructions?
  - Will a user action show the instructions or will you present them?
  - How frequently will a user need to reference them?

- ☐ Applications should provides instructions on how to use the application.
  - Consider an “onboarding” experience if your application requires significant instructions.

Connectivity

- ☐ If an application requires an Internet connection, the application should should operate under all normal usage conditions (ie. internet connection, no connection, and poor connection).
- ☐ The application should maintain a great user experience if the device is unable to make a necessary connection. For example:
  - Show an spinner
  - Present an alert notifying user that a connection cannot be made
  - Cache to prevent data loss
  - Degrade functionality (eg. show placeholder images)
- ☐ As a good citizen of the platform, activate the UIActivityIndicator during Internet usage.

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 manipulation
- ☐ If the API, for example URLSession, already works on a background thread, there is nothing additional you need to do.

Data

- ☐ It is not a requirement to have the application interface with outside data sources
- ☐ If your application requires a networked data source, then it should retrieve the data. It is acceptable to retrieve mock data (ie. you don't need to have a server generate the data).

Adaptive Layout

- ☐ Your application can use whatever layout is appropriate for the design *you specify*. For example, if you state your app is Universal, then it should look good on all devices. If you state it is an iPhone only app, then only consider the iPhone in its design.
- ☐ Make sure that your app is restricted to the devices that you specify
- ☐ If appropriate, ensure that you use adaptive layout techniques to make sure the content looks good across all devices

Icons

- ☐ Applications should have all the appropriate icons for each capable device and display.

Settings

- ☐ Every application should include a Settings.bundle to handle user preferences.
- ☐ The bundle should have, at minimum, a row with the developers name. More rows can be added as needed for the app functionality.
- ☐ The application should have the following behavior to handle settings:
  - Initialize the user preferences at launch (ie. register defaults)
  - Create a NSUserPreference named "Initial Launch" that stores an NSDate on first launch

Alerts

- ☐ Create an alert view that prompt users to "Rate this App in the App Store" on the 3rd launch. Do not use the official App Store rating control, this should be an alert view that you create.

App Preview

Each student is required to prepare an App Preview of their application that will be shared with the class during the final exam week. App Previews are a maximum of 30 seconds and should show off the "best" of your application. It does not need to show off everything in the application. No one want to see your settings screen. You can include text copy and voice over narration. Follow all of the official guidelines for an App Preview (https://developer.apple.com/app-store/app-previews/)).

In the past, the Department has shared some of the best App Previews on the University's main social media accounts. This has provided incredible visibility for some aspiring developers.

Submission

Commit all files to your final project Github repository. Include the marketing materials in the the git repo in a folder named "MarketingMaterials" and use the following naming convention.

- *github-username*_marketing.pdf
- *github-username*_executive.pdf