

FINAL PROJECT REQUIREMENTS

MPCS51030 - Summer 2017

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 vision. It would be better to submit a polished app with less features, than 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 one page summary 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 warnings.

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.

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 short description of its functionality
- Any custom methods you write should include a name and description. Methods that are provided by the iOS 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:
 - All file paths and URLs the app reads or writes to
 - Any target-action methods (e.g. a button pressed)
 - Any URLs for data downloaded from the Internet

Splash Screen

- 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. This is not the same as the Launch Screen that is provided in Storyboard.

Application Operation

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

Instructions

- Applications should provides instructions for use (e.g. splash screen, info button, etc.) somewhere in the application.

Connectivity

- If an application requires an Internet connection, the application should should operate under all normal usage conditions.
- If necessary, it should degrade functionality if appropriate. For example:
 - Present an alert notifying user that a connection cannot be made
 - Cache data needed for uploads
 - Handle airplane mode
- Present an UIActivityIndicatorView 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

Data

- It is not a requirement to have the application interface with outside data sources
- It is acceptable to fake the server calls and download a static file

Adaptive Layout

- Your application can use whatever layout is appropriate for the design you specify
- 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 (e.g. @2x,@3x for retina display and iPhone6+, respectively)

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

- Applications should utilize an UIAlertView somewhere within the application
- If it is not part of the normal user interface of your application, Have the alert view prompt users “Rate this App” in the app store on the 5th launch.

Final Presentations

Each student is required to present their application to the class during 5-10 minutes. The presentation is not part of the final project grade. You should prepare a limited number of slides (~5) that, at the very least, show off screenshots of the application. You may also elect to show your application by connecting a device to the projector (or showing the Xcode simulator). The presentations should include:

- Present a very high-level overview of the implementation details
- Showcase a unique, or challenging, feature of your application
- Present the marketing materials of your application
- Perform a demo or show screenshots of the application

Due Date

Everything is due August 24th at 11:59 AM (CST). Graduating students should have everything submitted by August 16th at 11:59PM (CST).

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*