Final Project

Goal:

The goal of the final project is to demonstrate the skills learned in class via the lecture, homework assignments, and other study to build a functional iOS app.

Deliverables:

A project proposal and any supporting material committed to your final project repository in a folder named Proposal.

An implementation of the proposed app in your final project repository. Make intermittent commits as you feel is appropriate. When you have finished your work make sure that you have pushed your commits to origin and they appear there as expected when a new copy of the repository is cloned.

Requirements:

Proposal

You will create a project proposal that will describe the app that you intend to implement for your final project. The proposal needs to include a description of your data model and the screens of your app. The app that you propose is what you will be expected to implement and part of your final project grade will be based on how close you get to implementing the app that you propose.

Your app should be at least as complicated as the LumberCart app that was developed in the homework assignments. It needs to be complex enough to represent 2 weeks of significant work and 40% of the grade for the course. Your proposal will be reviewed and approved once your plan is clear. If it does not adequately describe your plan (or your plan is not sufficient) you will be asked to revise it.

The description of your data model should include your entities, their properties, and their relationships. You can use whatever technology makes sense to store the data for your app, although in most cases Core Data is the appropriate choice. If you propose a technique that is not appropriate (i.e., storing large amounts of user generated content in UserDefaults) you will be asked to revise your proposal.

You should include rough sketches of the screens and the controls / UI elements that will appear on them. These sketches don't need to be high fidelity, taking photos of drawings on a white board or paper is sufficient; the main goal of the drawings is to communicate your idea.

Your screens should utilize controls / UI elements that are provided by the UI framework you are using (UIKit or SwiftUI) and can include custom controls if you wish.

In general the function of your app can be whatever you find interesting. It is best to propose an app that you are interested in so that you are motivated by that idea when implementing it. There are many technologies available to explore in the SDK and you do not have to limit yourself to what we have discussed in class (and there will be time in lecture to cover additional topics, so what is proposed may influence what is discussed). Some examples of things you could use:

1. Interesting gesture recognition.
2. Using maps in your app.
3. Incorporating the camera, photo library, or other media.
4. Using contact information from the address book.
5. Local or remote notifications (push notifications).
6. Sending email or text messages.
7. Any of the many APIs that are available on the internet
8. Creating a custom control or UI element

However, please do **not** propose either of the following:

1. Do not propose a game. Games interest many of us but they are complex to implement and generally the hardest parts of creating a game do not directly involve the skills we are learning in this course. If you feel that you have an idea for a game that might be an exception to this, you should ask for permission before spending time on the proposal. In general game ideas will be rejected however (it just usually does not work out well).
2. Do not propose a clone of the homework assignment or some other example. Your app can use ideas and elements from these things (i.e., you can have a list and a detail screen, you can have a form to add new content, etc.), but if it is too similar you will be asked to update your proposal with changes to make it more unique.

Implementation

Once your proposal is approved you should start implementing your app. You will be expected to accomplish the following in your implementation:

1. Implement the design that you proposed. The app you implement should be what you have proposed in your approved proposal. If during implementation you feel that you need to make significant changes to your design, you will need to update your proposal and get approval for the changes to avoid losing points.
2. Have appropriate separation of concerns in your app code. You should organize your app cleanly to separate different aspects of the code. This means your app should have distinct components like services, models, view models and/or view controllers, and views as was seen in the homework and lecture code.
3. Utilize dependency injection and unit tests. Your app should utilize dependency injection to facilitate unit testing and have good unit test coverage.
4. Provide good accessibility and UI tests. Your app should provide a reasonable accessibility user interface and UI tests that exercise all of the functionality in your app.
5. Implement code of good quality that is stable. Your app should demonstrate your ability to write Swift code. You should make an effort to make your code readable with good variable names, white space, and coding style. Your project should not have warnings or errors and your app should not crash.