



Personal Capstone

The intent of a personal project is to build something using technologies you have been taught, as well as to learn new ones on your own. You will learn basic AGILE principles and work with a project manager.

To graduate you must get your application reviewed by a mentor and meet competent or higher on all. You can view the grading rubric [here](#).

Include a link to your Github repository when you submit this assignment.

Requirements

→ Networking

- ◆ Must use either a custom backend
 - Firebase or CloudKit
 - Create / POST **
 - Read / GET
 - Update **
 - Delete **
 - ** Must manipulate the database
- ◆ Or Two (2) Separate API Endpoints
 - Read / GET
 - Create / POST (Optional)
 - Update (Optional)
 - Delete (Optional)

→ Framework

- ◆ Your capstone must use a new framework not taught in class
 - Ex: MapKit, ARKit, Etc.
 - A list of all Apple Frameworks can be found [here](#)

- Must explain why you chose this framework
 - What challenges will you face using this Framework?
 - What will you learn using this Framework?

→ Views

- ◆ Your capstone project must have multiple ViewControllers
 - Minimum of two View Controllers
 - Ex: Login View, ListView, Etc.

→ A fully planned project

- ◆ Defined MVP Features
- ◆ Defined API Endpoints
- ◆ Minimum 20 tasks in Trello (or Jira)
- ◆ **You need to pass off your MVP with your mentor**

Planning and Design

Project Planning

- Planning is extremely important. It's said, with tongue in cheek, "Weeks of programming can save you hours of planning". It can be tempting to jump right into code immediately at the start of a project, but you'll quickly start making mistakes and losing time. Taking a good amount of time to plan a project out, before you even write a line of code, will make the coding process significantly smoother and quicker.
- An important concept in planning is the "MVP". This stands for Minimum Viable Product. Simply, what is the smallest, simplest, least-amount-of-features version of your product that you could make that would still be considered completed and passable? For example, I am making a simple chat app. What are some "MVP Features"? The ability to type in a message, the ability to send a message, and the ability to receive a message. If I want to add any other features, I need to think carefully and gauge if they are part of my MVP. A cool animation whenever I send a message? A notification sound when I receive a message? Those aren't absolutely necessary for my app to function, so I need to put them aside until I finish all my MVP features. When planning, make a list of all the features your app **needs** to function and work on them until they are done.

- Another important facet of planning is setting tasks for yourself. You can waste a lot of time trying to figure out what exactly to do next. Switching constantly between thinking about what to do and actually doing can really slow you down. It's a good idea to take some time to split your project into tasks. The more the better! A good goal is to see if you can split your entire project into hour chunks. If you think you have made as many tasks as you can think of, break the tasks you have into smaller tasks. This will allow you to focus on actually doing the tasks, which will speed up your development time. [Trello](#) is a great site for task management.
- It's worth noting: plans don't need to be set in stone! You'll find the need for some extra endpoint, or you'll realize a given feature really isn't MVP status; that's completely fine. It's important to be agile. It's important to plan, but it's also important not to waste time trying to come up with every single eventuality.
- To aid in planning, you might want to use the [App Canvas](#) we've created for you.

