MAD I - Final Project Deliverables

Note: Documenting takes time - and it's part of your final project grade - don't put this off until the night before.

Source Code including all external frameworks and libraries and resources (images, sounds, etc). If I can't run what you submit, your grade is a zero.

A copy of your presentation. It should include: A brief description of your project; Challenges you faced; Any third-party libraries you used and why; and anything else you think would be helpful to someone doing a similar project.

In a Word Doc: - Product Definition Statement

- List of features you implemented

- Wish list of features you could put in the next version.

- Self evaluation and documentation

  Grade yourself and justify it. Point out where you did A-level work. Refer to "What does an A project look like" below.

  Describe what each class does and how the pieces fit together. A UML diagram is optional.

  Document your "above and beyond"

  Declare ALL third-party frameworks you used, and any code snippets or tutorials you utilized. Give original author names and URLs.

3) Install your app on your iOS hardware and bring it to class. For Covid semesters, if you have your own device, you can provide a screen recording if you wish, it's not required. Your borrowed devices should be returned before you head home for break.

What does an A project look like? -

Properly scoped

- Meets product definition statement and feature list. **The app should do at least one thing really well.**

- Dynamic data (not a hard-coded array)

- Software Design Patterns
    - MVC
    - Delegation and Protocols
    - Notifications
    - Properties
    - Lazy Loading
- Software Engineering
    - Loosely Coupled/Reusable code (see design patterns above)
    - Factor out repeated code into methods or classes - remember DRY - "Don't Repeat Yourself"
    - #defines and enums to reduce/eliminate string and number literals (ie no "magic numbers" in your code)

- Polish in Interaction and UI Patterns
    - Appropriate use of NavigationView, TabView, etc.
    - Nice VC layout and good use of color.
    - Animation?

- Respects iPhone usage patterns and the mobile context
    - Frequent and short use (be sure to save state)
    - Carried everywhere (easy to use no matter environment)
    - Single user (remember their info, don't make them retype things to login, this isn't a web app!)
    - Social context (let them share things from your app)

Respects the end user:

- Don't make the user wait.
- Let the user know what's going on - spinners? animation?
- Save state when the user quits (NSUserDefaults or?). Restore that state when "the app starts back up.
- Fail gracefully". For example, if there is no Internet Connection, let the use know and give them some sort of offline functionality.

- Take advantage of iPhone hardware capabilities (if appropriate)
    - Location
    - Multi-touch
    - Accelerometer