Homework 2 - Individual Project Proposal REVISED

1. Project Overview:

- Due to the pandemic, classes have now gone virtual. Instructors are forced to rely on multiple websites like Canvas, or Moodle to host their assignments and due dates. As a result, students must jump from website to website just to find their homework - the repercussions of which sometimes lead to late assignments, or missing assignments altogether. My project aims to eliminate this problem.

- This project will be a mobile application that utilizes image recognition technology to allow users to take a photo of their assignment's due date, and store that information in a centralized location. Instead of manually writing/typing out every assignment's due date, a student can simply take a photo!

- Although building a cross-platform app is possible, it is extremely time consuming. Due to time constraints, I will build the application for Apple (iOS) mobile devices only (ex: iPhone, iPad). This is because I have much more experience in iOS development than Android.

2. Technologies To Use:

- Programming Languages:
  - Swift
  - C++ / Objective C

- IDE:
  - XCode

- API:
  - Google Cloud Vision API

- Database:
  - Google Firebase

- Libraries:
  - Firebase SDK
  - Apple's UIKit
  - GoogleMobileVision/TextDetector

- Devices:
  - iPhone 7+
  - Macbook Pro (2017)

- Design:
    - Sketch

3. Essential Parts

Part	Reason Essential
iPhone	The project will be an iOS application where the functionality relies on the rear facing camera. Without an iOS device, I can't test the app.
Database	Firebase is a database platform developed by google for mobile and web applications. Although Firebase itself is not essential, having a database is, and I plan to use Firebase. Without a database, the app will not be able to store the information it needs to function, such as images.
UIKit	UIKit is a library provided by Apple to help manage the app's interactions with the system and provides classes to manage the app's data and resources. Without this, a UI can't be built (at least in a timely way, and time is essential).
Google Cloud Vision API	The Google Cloud Vision API allows developers to easily integrate vision detection features within applications, including image labeling. Without this, the app will not be able to recognize text within an image.
XCode	Xcode is Apple's integrated development environment for macOS, used to develop software for macOS, iOS, iPadOS, watchOS, and tvOS. Although other IDE's exist, Xcode is essential because it streamlines the development process. Personally, I can't think of any other IDE that comes close to XCode in terms of features, and ease of use, for iOS development.
Sketch	Sketch is a vector graphics editor for macOS. I will be using this software to design the user interface as well as storyboard the user pathways throughout the app. This is essential because faulty UI/UX renders an application unusable.

4. Outside Resources:

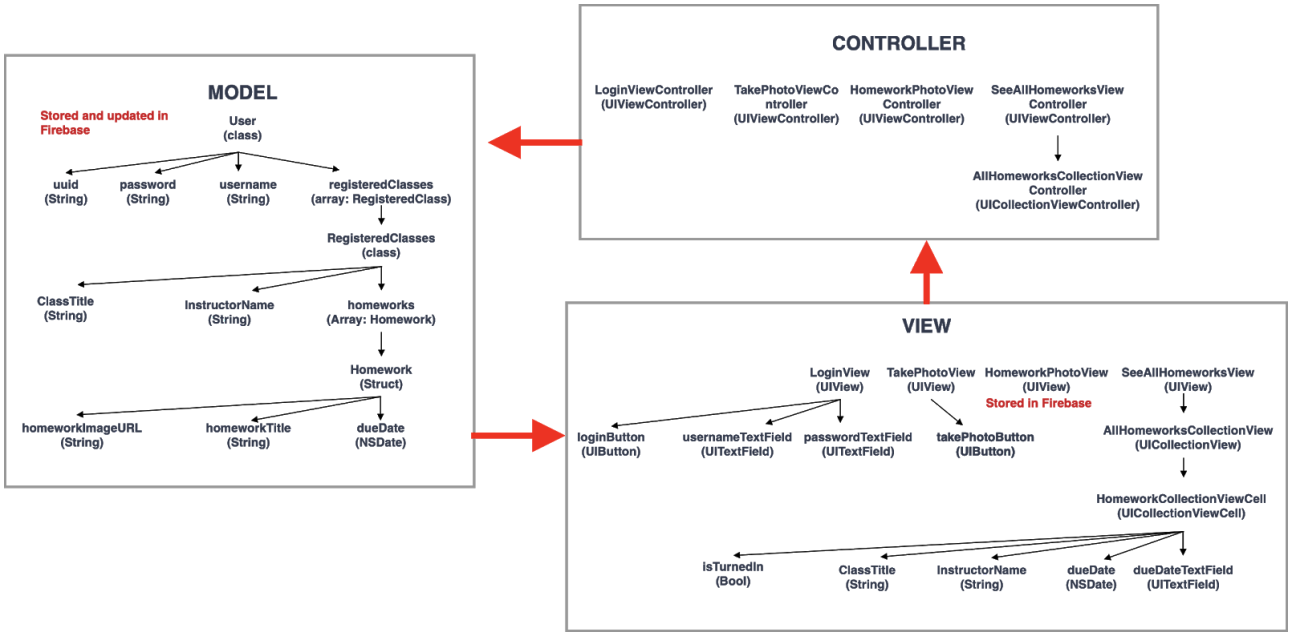
- Google Firebase
- Google Cloud Vision API

5. Architecture Proposal.

- Software Design Pattern:
    - I will be utilizing the Model-View-Controller (MVC) development design pattern in this project. In MVC, the Controller manipulates the Model (data) which in turn updates the Views. The user sees the Views and uses the Controller.

- My Model will be composed of a User that will be composed of elements like username, userPassword, userUUID, EnrolledClass, and so on. It will also include custom data for UI elements like text and buttons.
- My Controller will be composed of UIViewControllers that will update other Views.
- My Views will be composed of UIViews, UIButtons, UITextfields, and so on.

- Hierarchy



6. UI/UX Proposal

- If the user has not onboarded, the user will go through onboarding process, otherwise, the user will be brought to the home screen upon opening the app.

- Note: each ViewController by default creates a back button if the current ViewController is not the superview of the ViewController. This helps ensure the user can not get stuck in the app. There must be circular UI/UX flow.

- Onboarding Process:
  - User is brought to the login screen:
    - Login screen contains 2 textfields: usernameTextfield, passwordTextfield, and 2 buttons: signUpButton, signInButton
  - User taps signUp button:
    - UIChanges to show 2 textfields: emailTextfield, passwordTextfield, and 2 buttons: goBackButton, signUpButton
  - User fills out text fields (correctly) and taps signUpButton:
    - Account is created and user is segued to the home screen.

- Home Screen:

    - Has buttons: seeAllHomeworksButton, takePhotoButton
    - User taps seeAllHomeworksButton
        - User is segued to SeeAllHomeworksView via SeeAllHomeworksViewController
    - User taps takePhotoButton
        - User is segued to the camera (TakePhotoView) via TakePhotoViewController
    - User takes photo of their homework due date
        - Google Vision API call is made to extract the due date
        - User is segued to HomeworkView
            - Extracted due date is shown to user.
            - User may retake photo, cancel, or continue
            - If user taps retake, the user is sent back to the camera. If user cancels, then user is sent to the home screen. If user continues, the due date is saved, and the user is segued to seeAllHomeworks View where the new due date appears.

7. Project Timeline & Staying Engaged In the Course:

While working on this project I will stay engaged in the course by coming to class and diligently working on lecture activities. Since the programming language for my project is not C++, I will stay up-to-date with C++ programming exercises by using class notes, doing my own research, and communicating with other students in the class.

- Homework 2, Part 2&3 - Feb 28th
    - I submit a complete mockup and storyboard for the UI and UX.
    - Will take between 2-4 hours to create in Sketch
- Homework 3 - March 12th
    - Since functionality is key, I want to be able to integrate the API and show the concept works.
    - I should be able to take a picture of text with due date using my iphone and have it extract the due date.
- Homework 4 - April 6th
    - I complete the development of all screen UI excluding the auth UI (functionality is key and I don't want to use up too much time to create a login/signup)
- Homework 5, checkpoint - April 29th
    - At this point, I should be able to take a photo, extract the due date, and update the UI with a new homework due date. Basically, I want to have the core functionality complete. However, If I can't get all core functionality complete, I will complete it for the final due date because I'd rather present a working app than a pretty app.
- Homework 6, final due date - May 2nd
    - If core functionality already complete by Homework 5 then I will use this time to Implement non-core functionalities like auth and flush out the UI to the design.