City Art

Application Design Documentation

By

Adam Metelski

TABLE OF CONTENTS

|  |  |
| --- | --- |
| 1. Description | **3** |
| 1. Prototype overview | **3** |
| 1. UI Walk-Through | **4** |
| 1. Class Diagram | **9** |
| 1. MVC Architecture | **10** |

**Project Description**

For my class project I am going to implement an ios app called City Art. This app will be aimed at connecting street artist with the people interested in this form of art and helping them locate the art.

The app will work in two ways. One, if the art discovered is new to the app the user can take a picture of the art, mark it on the integrated map, and maybe write a description of the location/art. Two, using the app users will be able to look at a map and see street art near them. They will also be able to click on the objects in the map and see a photo and the description of the art.

My project will incorporate maps, gps location, and firebase to store all of the gps points, photos, and descriptions.

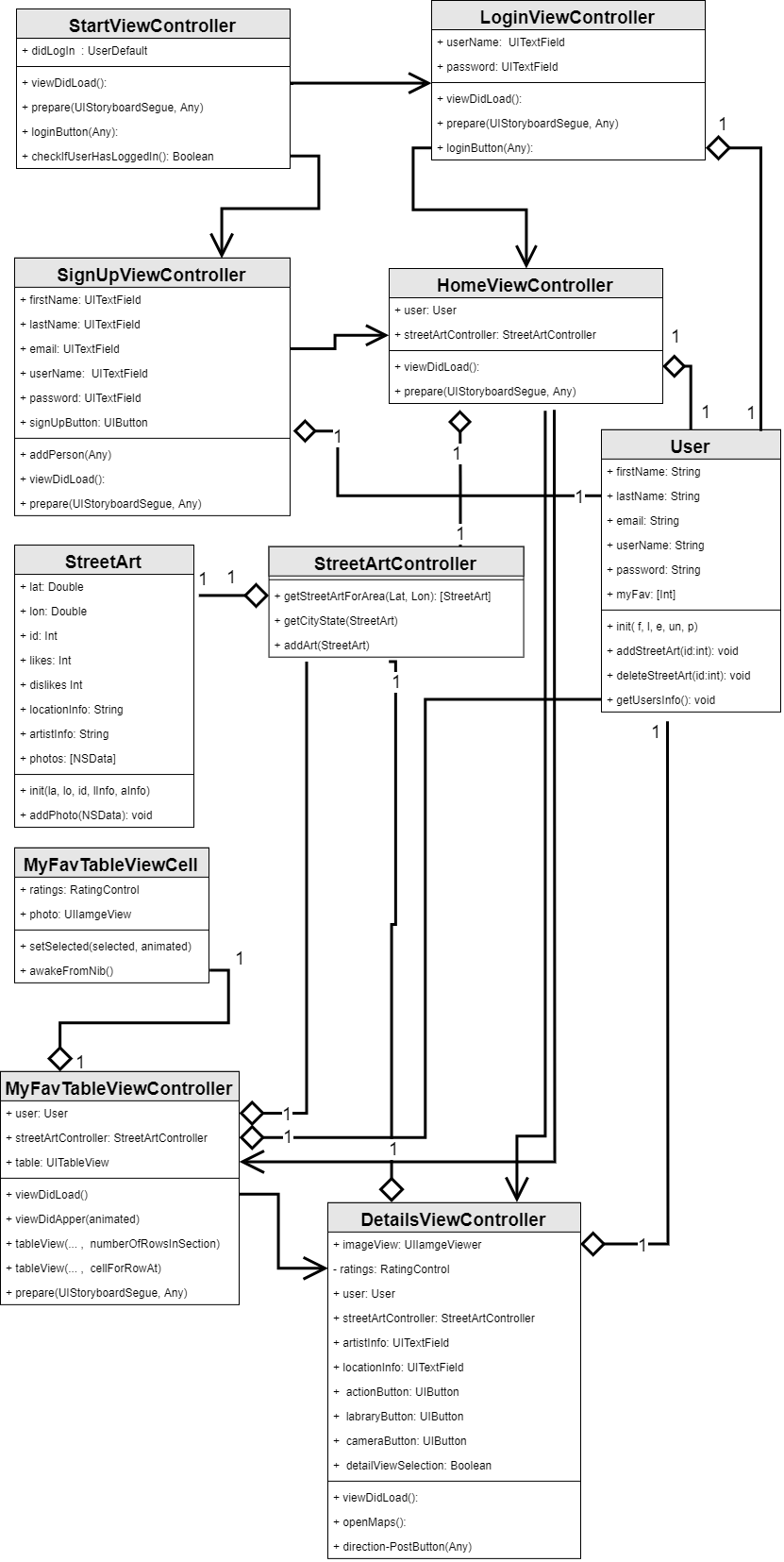
**Prototype Overview**

I created this initial prototype to start investigating the technologies that I plan to use in the application. My first task was to learn how to integrate with Google maps SDK. This involved acquiring an API key, installing Podfiles that would download and install the SDK into the project, and finally creating a simple view to show that the connect was established between my app and Google maps.

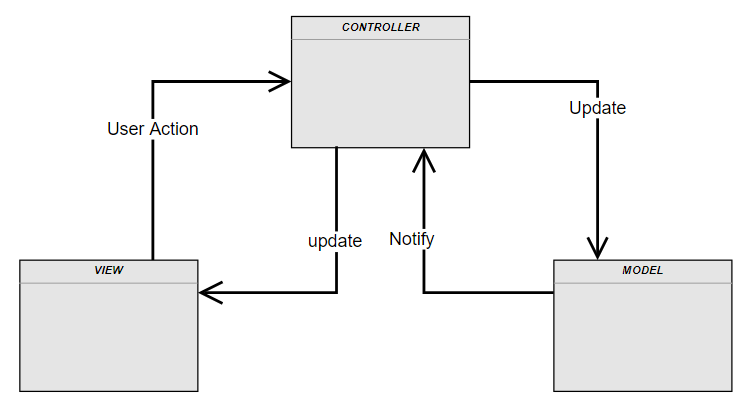
The second task was to setup Firebase. Similar to Google maps, I had to add a line to the Podfile that will all the dependencies to be add to my project. I then imported the Firebase class into the project. I will need to do further research into how to implement the connection between my app and Firebase.

**UI Walk-Through**

|  |  |
| --- | --- |
|  | This will be the main start up page when the app is started for the first time.  The user will then have the choice to sign-up or to login if they have already signed up |
|  | User will fill out this form in order to sign up for the app. |
|  | If they press log in they will need to provide there log in information. Once they have done this the app will store there information so the user will not have to get this information in every time. |
|  | Once the user has logged in this will become the welcome screen.  From here the user can select  - My Favorites  - This will store places that they have rated.  - Find Art Near Me  - This will go to the next page and show art  near the person location  - Post Art Near Me  -This will take the user to a map page where they can drop a pin and save the art to the site. |
|  | Here the users will be able to see art near them. The user will be able to select the pins and a detail view can be displayed on another page. |
|  | Once the user has selected to see the details of art near them they will be able to view photos of the art, location information, and finally artist information.  The user will also be able to rate the art.  If the user doesn’t know how to get to the destination they can press the direction button which will open ios maps. |
|  | In the post art view the user will be able to drop a pin and mark that location as a place where art exists.  Once the mark location button is pressed the app will take the user to the details page |
|  | Here the user can select to use the camera or selection photos from their library.  They will also be able to rate the art as well as add a location description, and artist info if known.  The POST button will save the data. |
|  | This is the My Favorites view. The user will be able to come back and view all the art they have rated over 3 stars. |
|  | Here they will be able view the details and get directions back to their favorite street art. |
|  |  |

**CLASS DIAGRAM**

**MVC ARCHITECTURE**



|  |  |
| --- | --- |
| **MODEL** | |
| Responsibility | * Firebase data storage and manipulation for the user objects * Firebase data storage and manipulation of the StreetArt Objects |
| **TRANSITIONS** | |
| Notify | * Will notify the controller if a user is valid or invalid * Notify if users login Information is correct * Notify when StreetArt has been updated |

|  |  |
| --- | --- |
| **VIEW** | |
| Responsibility | * Displaying all UI components on the Initial Log In / Sign-up page. * Displaying all UI components on Sign-up page * Displaying all UI components on Log-In page * Displaying all UI components on User Options * Displaying all UI components on Find Art Near Me page * Displaying all UI components on Post Art Near Me page * Displaying all UI components on Art / Post art / MyFavorite Details Page * Displaying all UI components on My Favorites page |
| **TRANSITIONS** | |
| User Action | * On Sign up button press have CONTROLLER and MODEL validate and add user * On Log In button press have CONTROLLER and MODEL validate users login information * On pin selection have CONTROLLER get details for the StreetArt selected * On Mark Location button press pass pin location to CONTROLLER |

|  |  |
| --- | --- |
| **CONTROLLER** | |
| Responsibility | * Determining if the user has login before or not * Pass user information to the MODEL for validation and registration * Retrieves StreetArt for the users give region * Interact with Google maps API and SDK. * Interact with Firebase database * Interact with ios device to open ios maps * Get list of My Favorite to populate tableView * Delete cell from My Favorite tableView * Grant access to application for accessing the Camera and Library * Get users GPS location * Upon receiving pin drop locations to see if location has been added before |
| **TRANSITIONS** | |
| Update View | * Notifying the user if they entered incorrect information on the sign-up page. * Notifying the user if login information is incorrect on login page * Opens ios maps when user need directions to the StreetArt location. * Update My Favorite Table View * Give StreetArt Details * Updating the users position * Updating google map view with StreetArt pins |
| Update Model | * Sending all the new user’s information to the MODEL to be added to the Firebase DB. * Send StreetArt object to MODEL for CRUD operations |