

Longhorn Walk

App Overview:

Longhorn Walk aims to help newly admitted UT students gets acquainted with campus by:

- Giving students points to their account by checking in at the location via Longhorn Walk.
- The students will then get a status level depending on the total amount of points in their account.
- Students can change their account details that are stored in a firebase firestore DB such as:
 - Username
 - Profile Picture
- Directions to the landmarks can be given from the student's current location via MapKit.
- Previously visited locations are stored in Core Data and refreshed daily so students can revisit locations on another day to score more points.

Core Audience:

- Newly admitted UT Austin students.
- Students who want to walk around campus.

Technologies Used:

- Core Data
- Core Location
- Core Audio
- MapKit
- User Defaults
- User Profile Path Using Camera and Photo Library
- Gesture Recognition
- Firebase Auth
- Firebase Firestore
- Firebase Storage
- TableView with Custom Cells

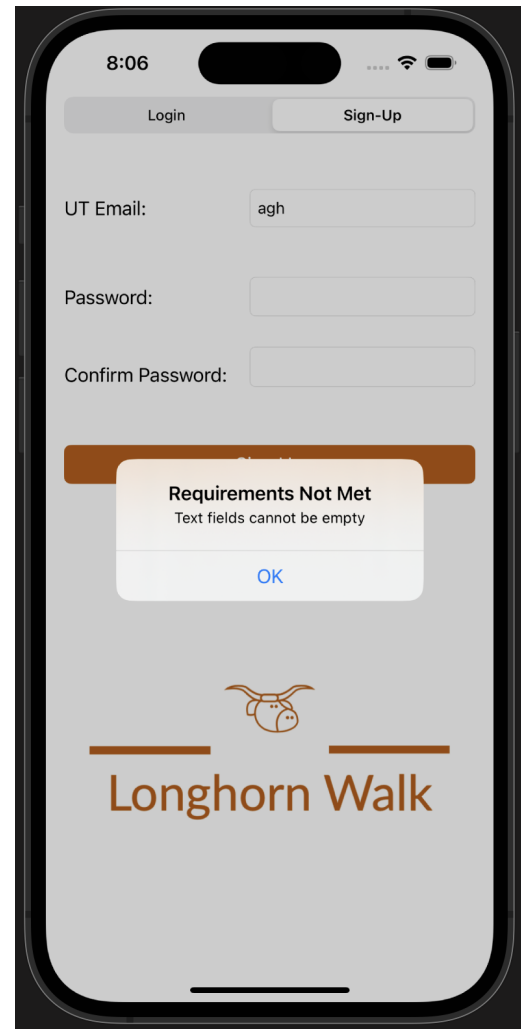
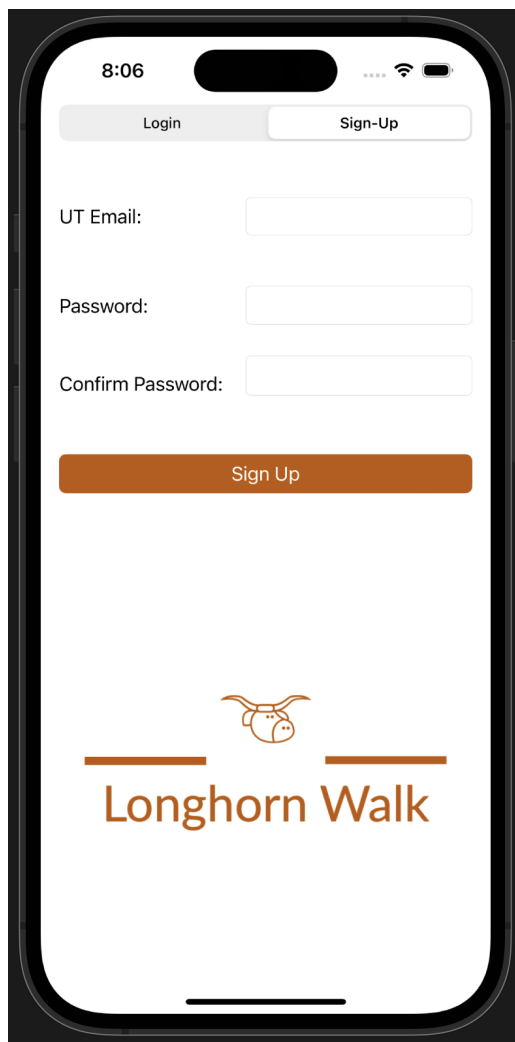
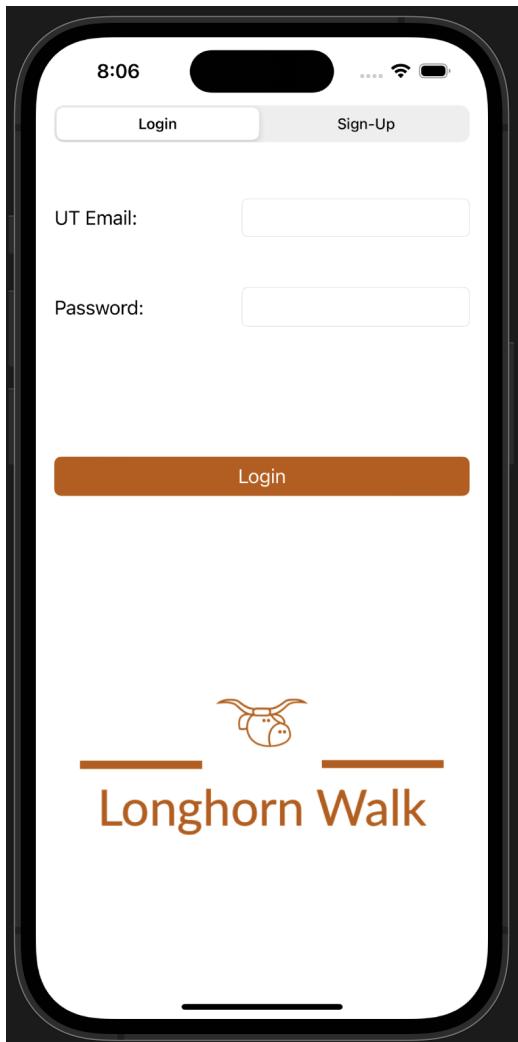
Launch Screen

The Launch Screen is shown upon launching the application with a quick animation and a sound file. After the sound file and animation are done, the user is then sent to the Login/Registration ViewController.



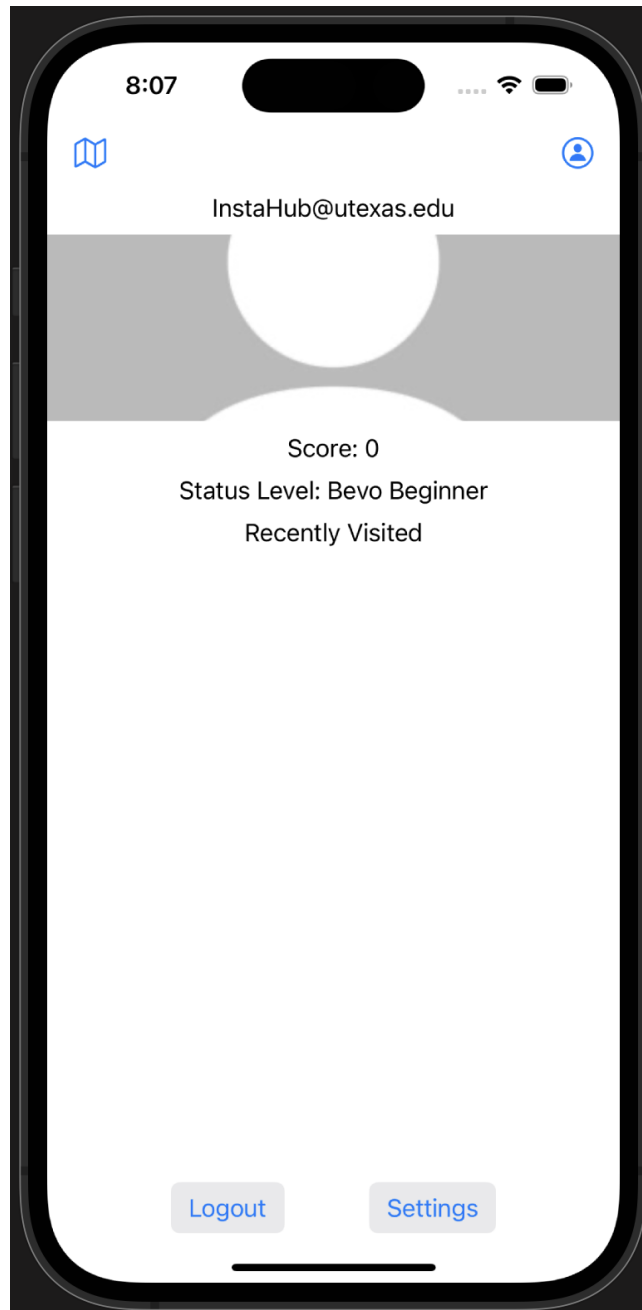
Login/Registration Screen ViewController

The login/registration viewcontroller uses a segmented controller to allow the user to either login or sign-up, with the login/registration view controller populating the screen with the corresponding text fields. Login and registration processes are handled through Google Firebase's Auth. Upon errors, alerts will populate letting the user know what the problem is with their login/registration.



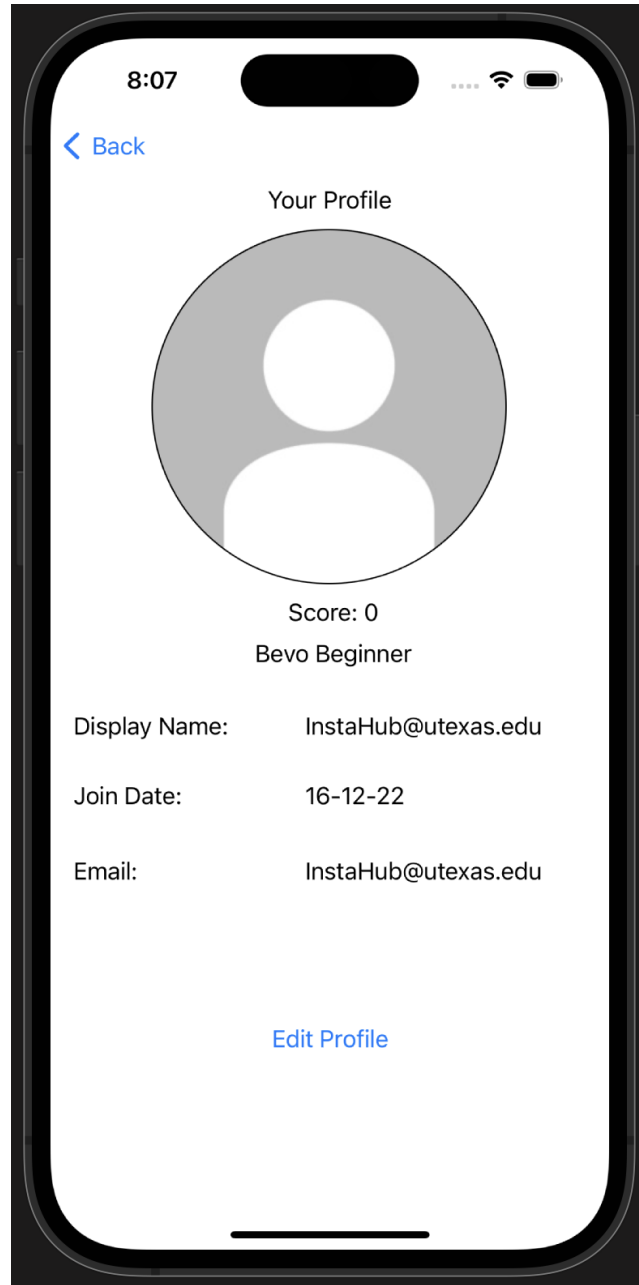
Home Screen ViewController

The screen is then populated with the student's username, score, status level, and a tableview that holds the locations visited by the student on the current day. This screen uses images for their bar items, which segue to their corresponding viewcontrollers.



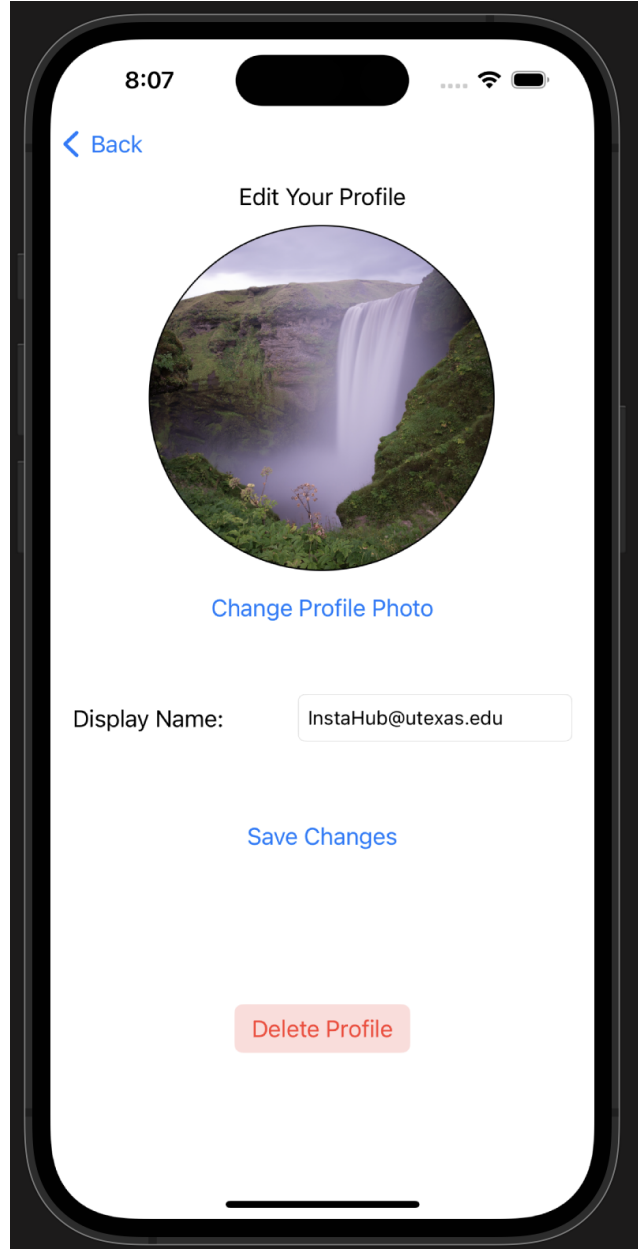
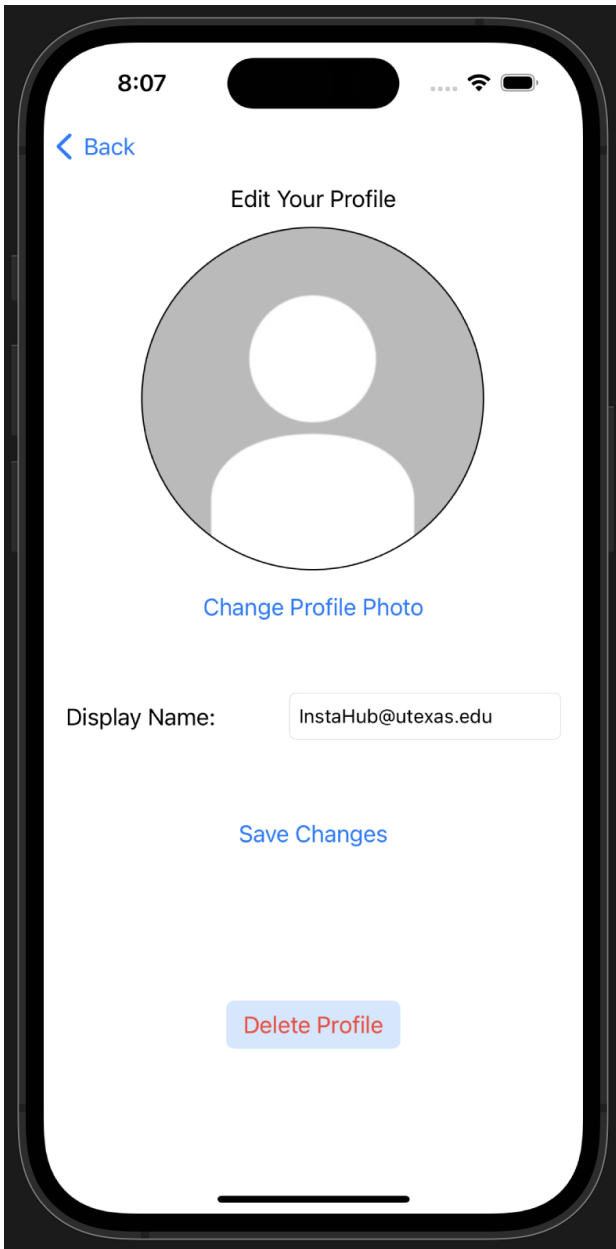
User Profile ViewController

The student's profile details will be shown and have the option to edit their details if they press the button to send them to the edit user profile viewcontroller.



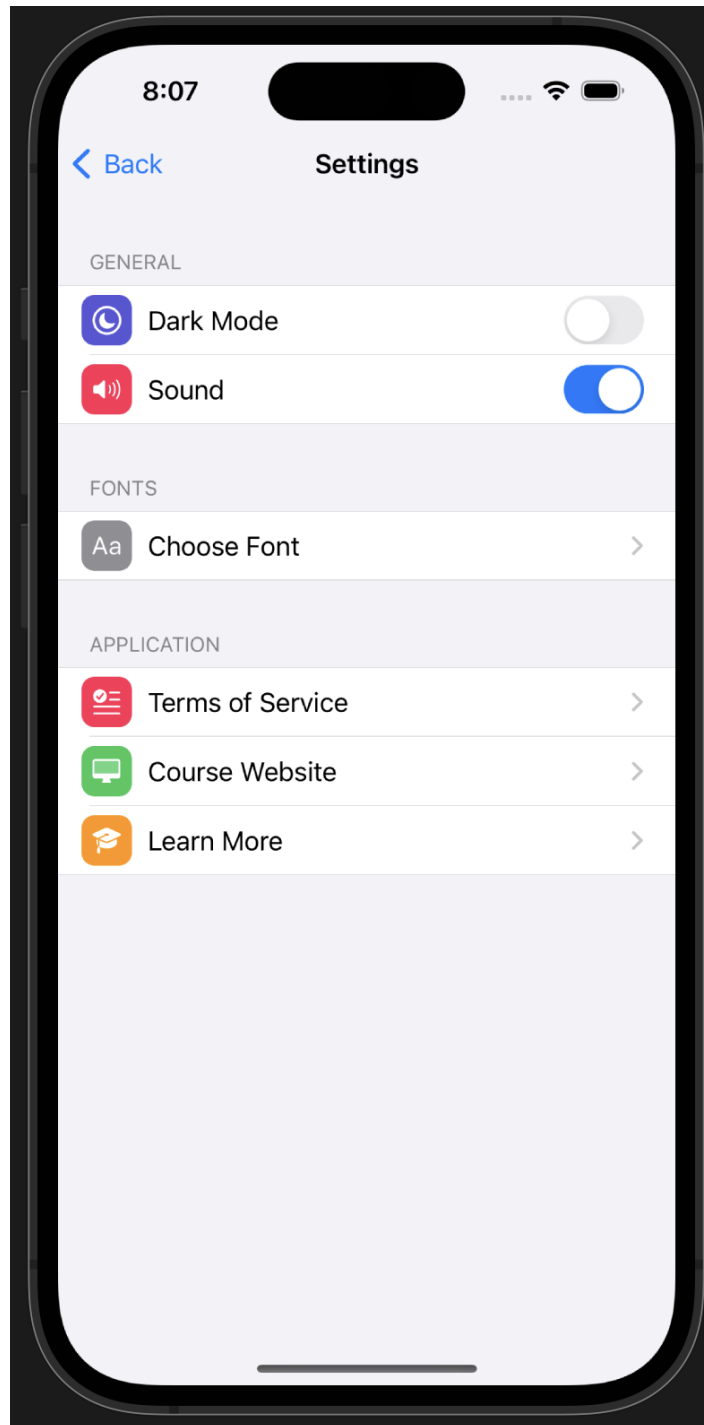
Edit User Profile ViewController

The student after getting to this screen will be able to change their username and profile picture. The picture can be swapped with a picture from the student's photo library or their camera. All changes made here will need to be saved first before the changes are registered. Upon saving the picture will be stored in Firebase Storage, with the file path to the picture being stored in Firebase Firestore with the newly updated username.



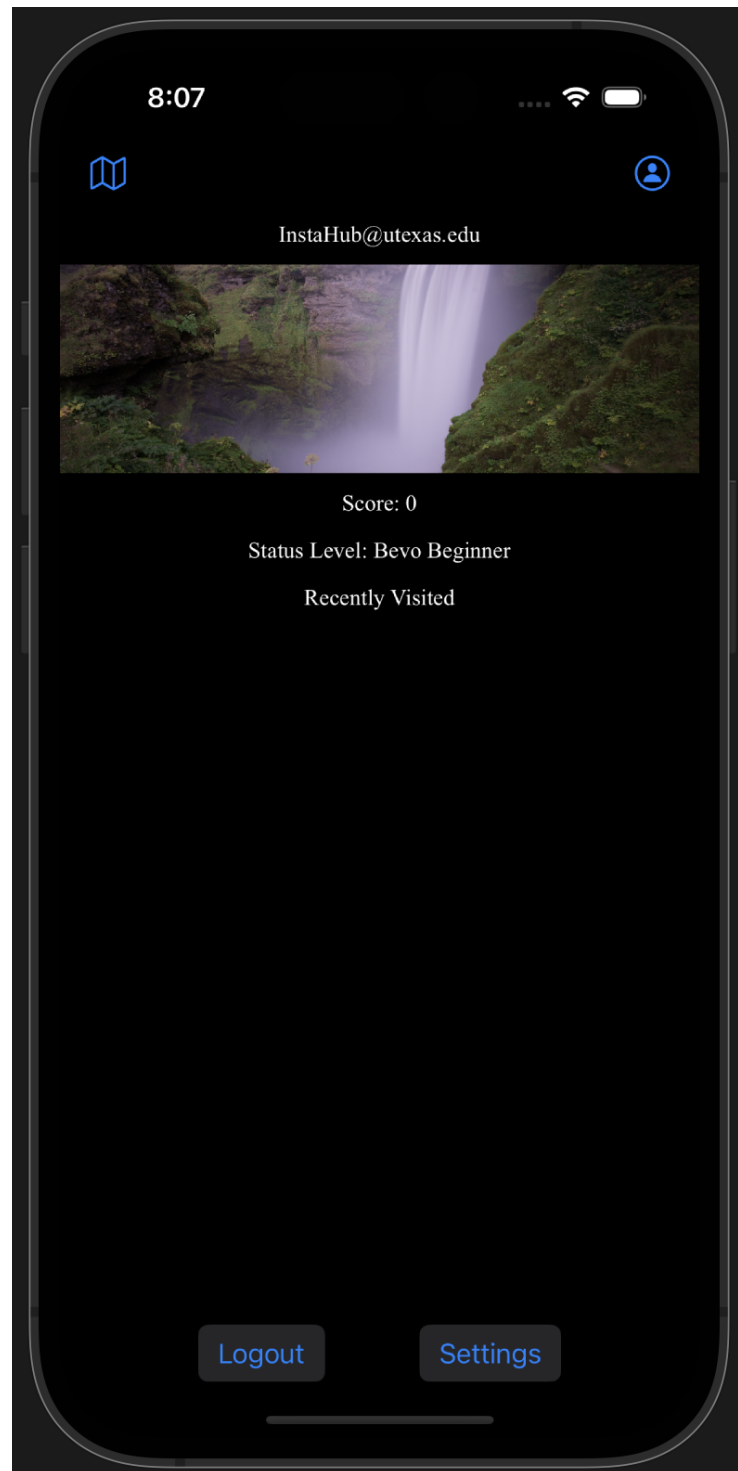
Settings View Controller

The settings screen is an iPhone system settings clone with switches for dark/light mode and audio on/off. Action sheet for user to select font. Course website button opens a safari to our professors course website. The settings are then saved to user defaults so that the changes will always be present.



HomeScreen ViewController After Changes

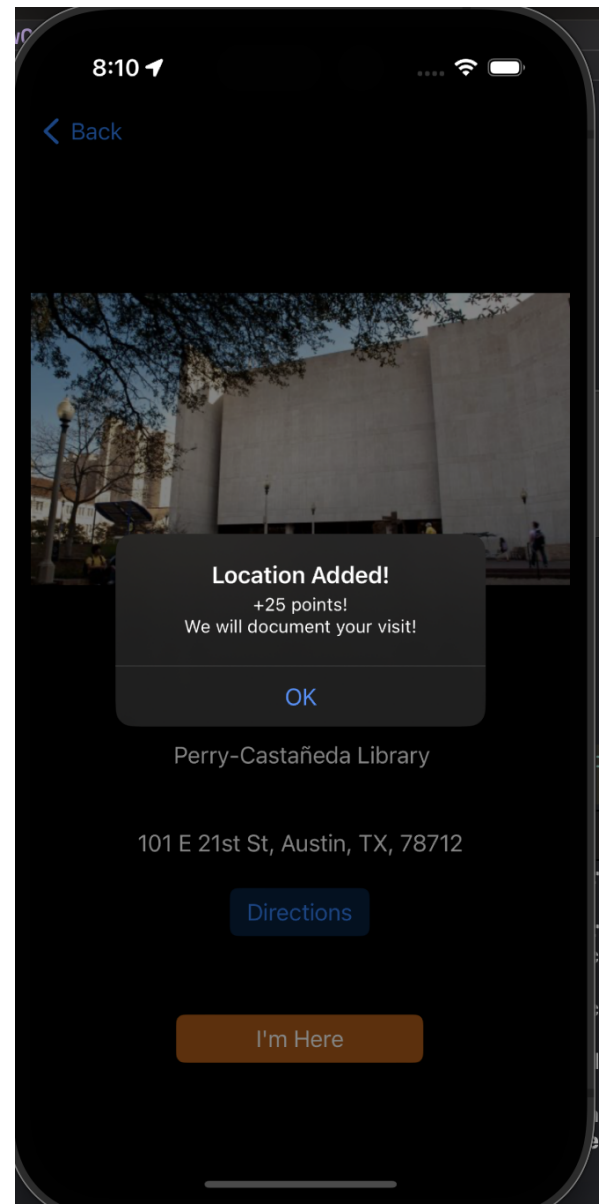
Dark mode was applied with a font change. The new profile picture is now showing as well.

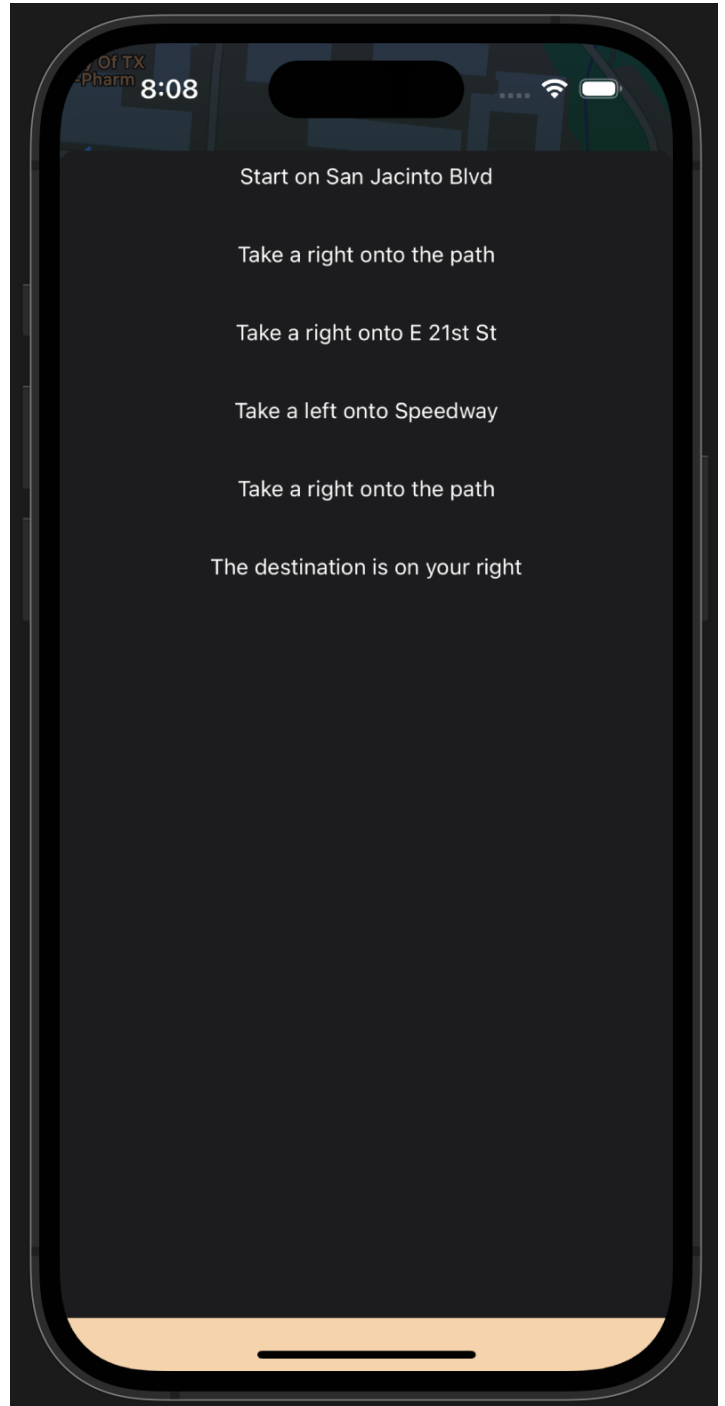
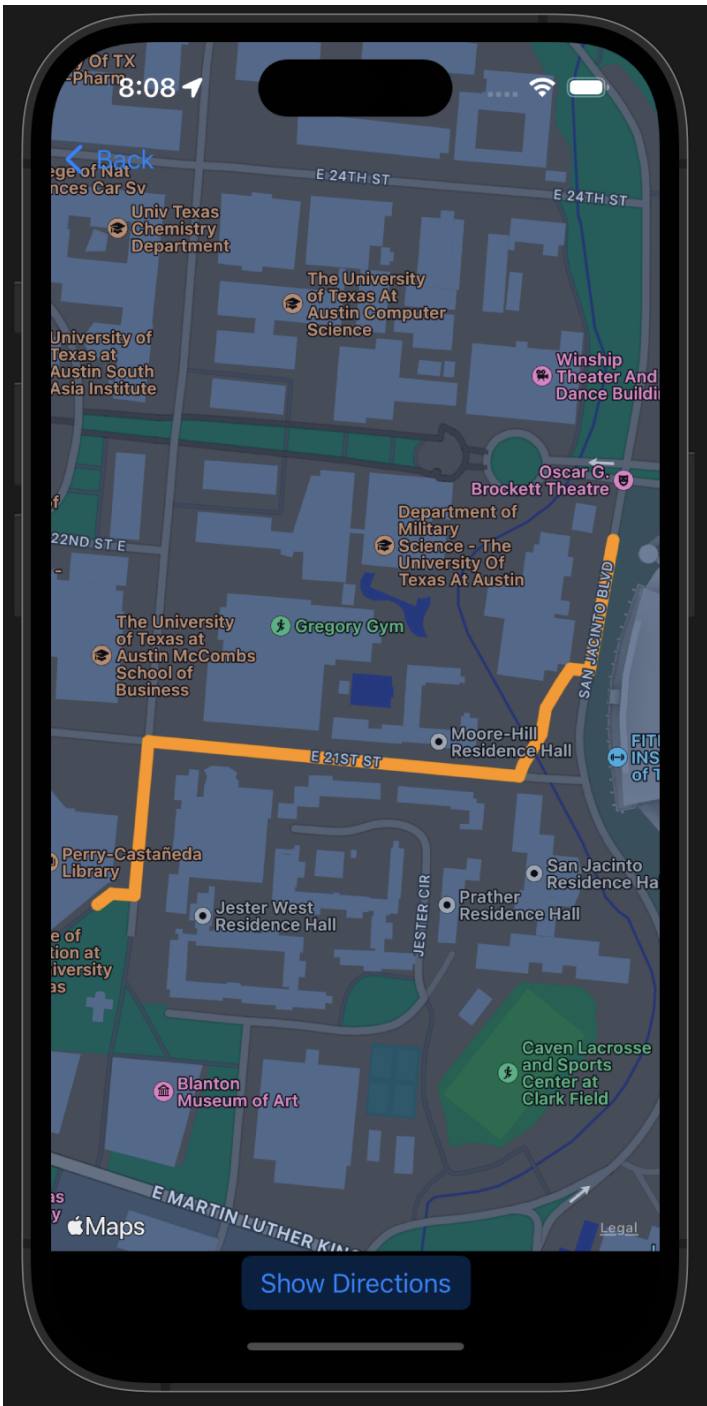


Location View Controller

Upon launch the user is asked for permission to allow location services for the student's location to be checked to grant the student points. The app currently has 10 locations to visit in a day and can be indexed by swiping left or right on the screen by using gesture recognition. When a student is at the location that is present on their screen, they can press "I'm Here". This will verify the location of the student by using Core Location to verify the student is at the location on the screen. If the checkLocation() function returns correct, points are awarded to the student and the location is saved in Core Data so that locations can't be awarded more than once in a day.

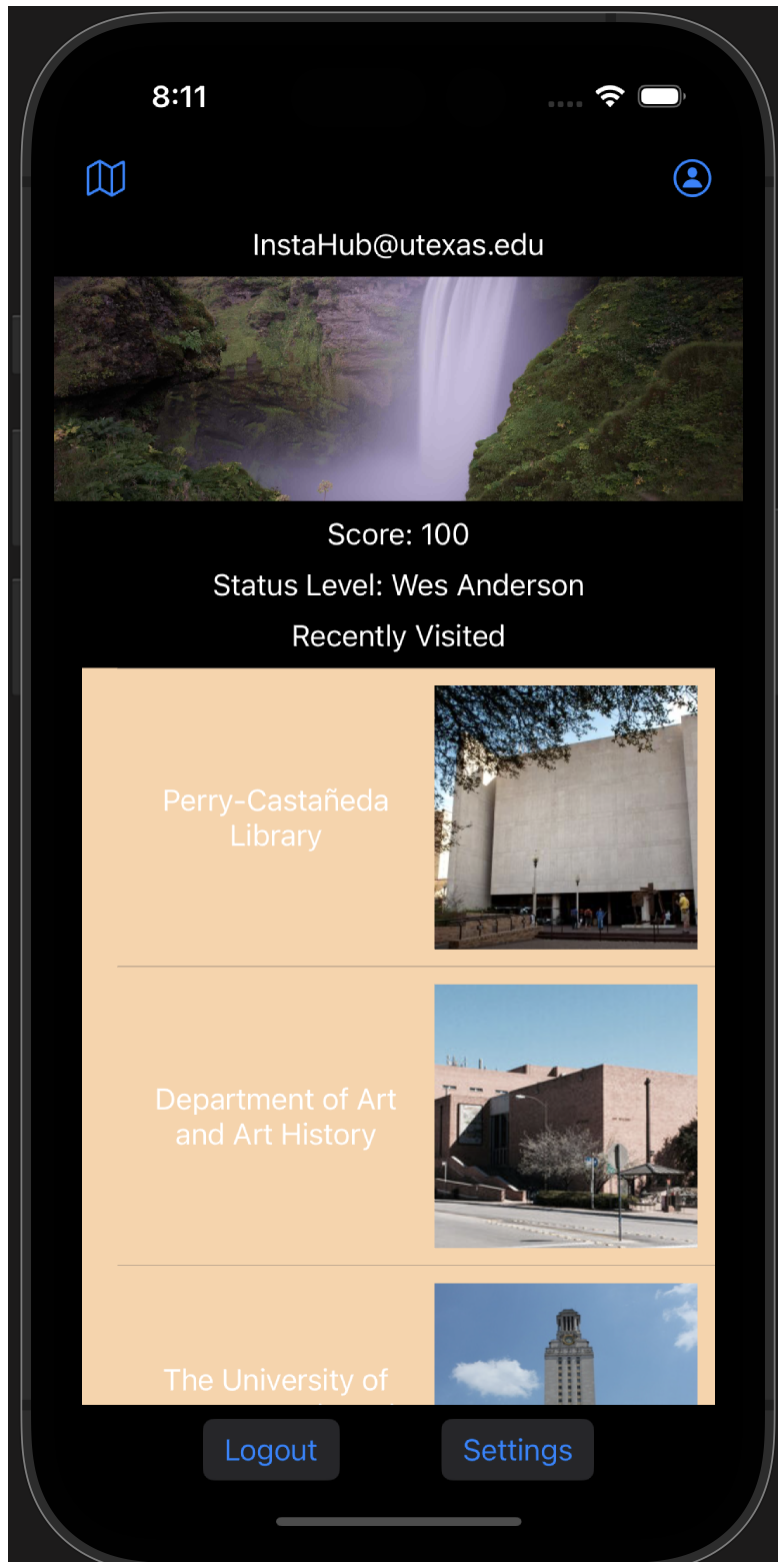
The user can also get directions to the current location on the screen, by pressing the "directions" button, which will use the student's current location and give directions to the current location.





Home Screen ViewController After Locations Added

The home screen is now populated with the locations visited in a custom tableview cell that are retrieved from Core Data. The score as you can see is also changed, which will give the student a different status level.





Firestore and Storage

Attached are photos of the account's information that are stored in our Firestore DB and profile picture stored in Firebase Storage.

+ Add field

displayName: "InstaHub@utexas.edu"



email: "InstaHub@utexas.edu" (string)  


joinDate: "16-12-22"


lastUpdate: "16-12-22"

profilePicturePath: "profilePictures/InstaHub@utexas.edu.jpg"

score: 100

<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	 InstaHub@utexas.edu.jpg	273.48 KB	image/jpg	Dec 16, 2022
<input type="checkbox"/>	 defaultProfilePicture.jpeg	5.82 KB	image/jpeg	Dec 3, 2022

 InstaHub@utexas.e... ×



Name
[InstaHub@utexas.edu.jpg](#)

Size
280,040 bytes

Type
image/jpg

Created
Dec 16, 2022, 8:08:08 PM

Updated
Dec 16, 2022, 8:08:08 PM

Future Considerations

Future features to implement in this project, should further work be pursued:

- Ability for users to add locations to the current 10 locations by taking pictures of locations on campus.
 - These locations would first be verified to be on campus by using reverse geocoding.
 - These user-added locations could then be visited by other users and rated based on the user's thoughts on the location posted.