**Mob tech assignment - 1915084**

Delight Travel is a travel destination app in which you can find a list of your next travel destinations and you can add your favorite destination.

**Technologies used:**

* Swift
* Storyboards.
* UIKit Framework.
* MVC Design Pattern.
* Firebase Database.

**Functionalities achieved based on mod book:**

Below listed functionalities are achieved:

* 7 Screens with user interaction.
* Firebase Realtime Database used to store Users and Destinations.
* Firebase Authentication user for Signing up new users.
* Firebase APIs used to get and store data in the Firebase Database.
* UserDefaults local storage used to persistency.
* Firebase Storage to save pictures.
* Device Camera and Gallery used to get the destination images.

**Credentials Firebase:**

[delighttravel29@gmail.com](mailto:delighttravel29@gmail.com)

Nissan786#

**Overall Storyboard:**

UIs and flow in the storyboard file.

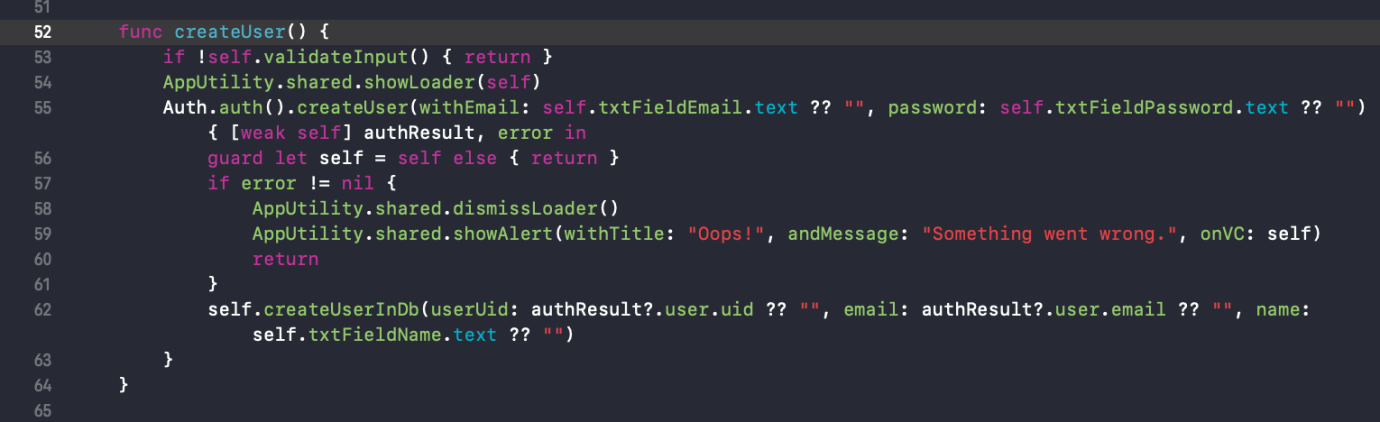
****

**User Onboarding Flow:**

For user onboarding I made the signup and login screens with functionality.

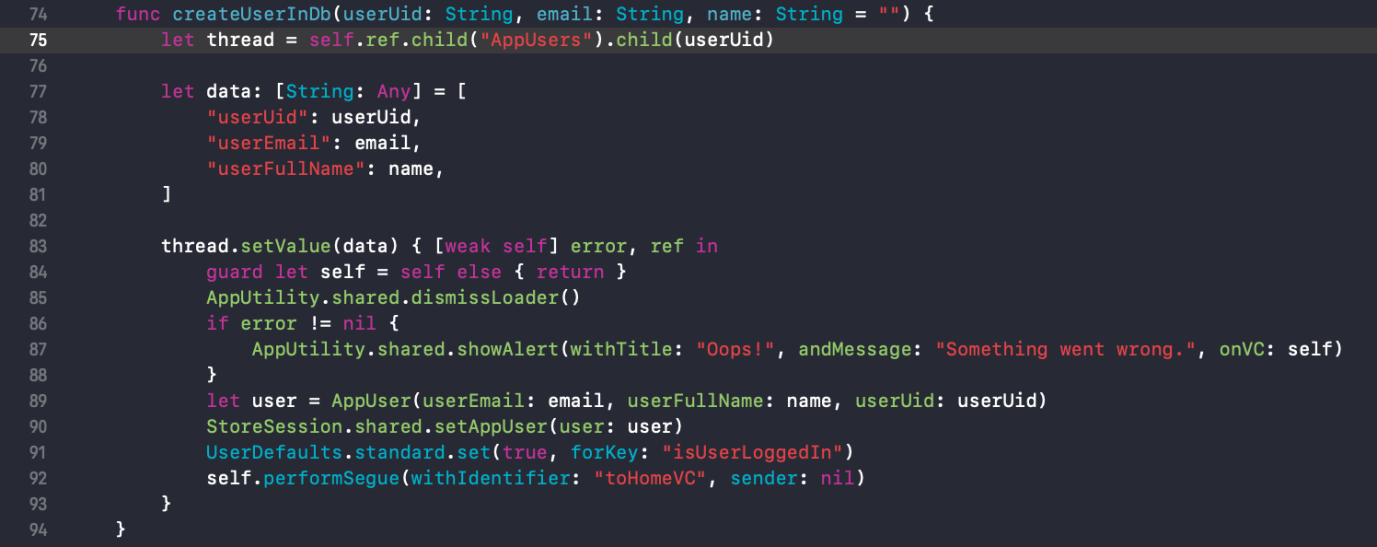
Sign up flow is basically build up on Firebase Authentication and Realtime database, when the user gets signed in it gets authenticated with Firebase and once it gets authenticated it gets saved in the Database in AppUsers Table.

Authenticating User:



In this function I am authenticating the user using Auth.auth().createUser APIs method provided by Firebase.

Creating User:



In this function I am saving an authenticated user in Firebase Realtime Database and the table name is AppUsers.

Regarding the implementation of the Login Feature below, in login I am taking two steps, first is getting Authenticated user against the Email and Password and once I get the user I am calling that specific user’s data from Firebase Database.

Sign In with Email and Password:



In the LoginVC I am logging in with user’s email and password, this is the Firebase Authentication API method which takes 2 parameters, Email and Password and returns the User.

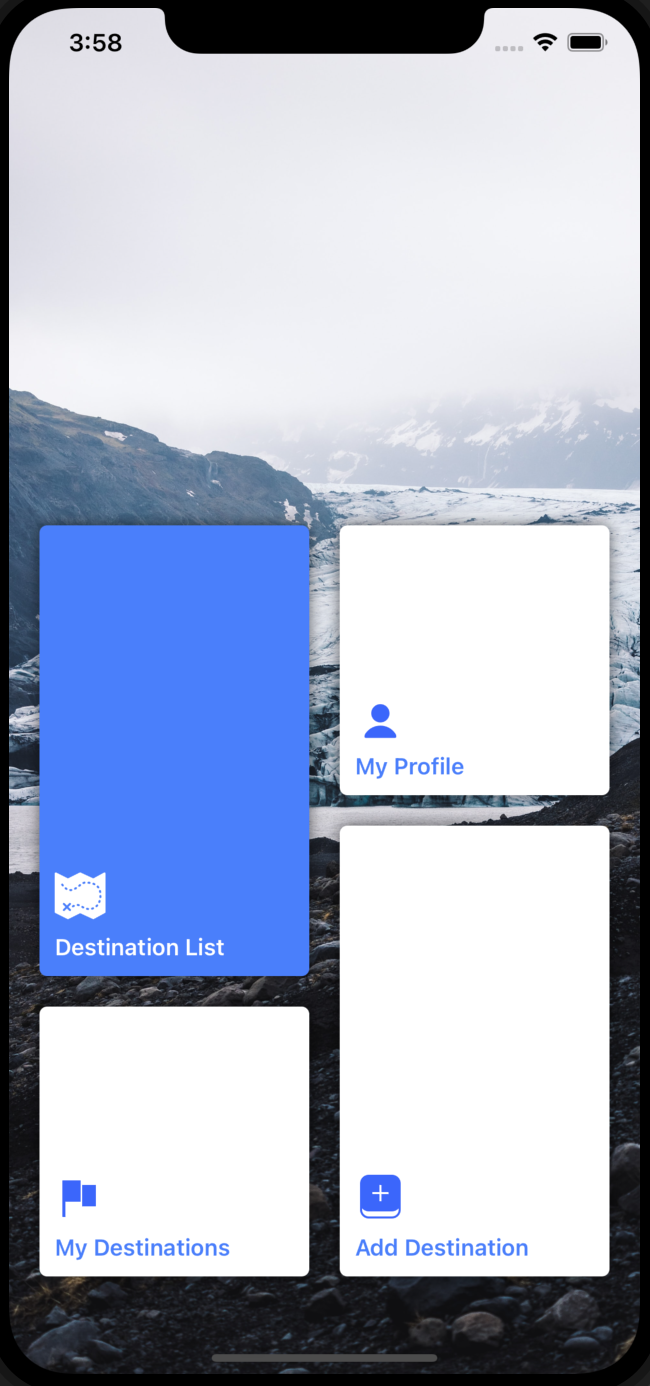
Getting User From Database:



After successfully logging in, I get the User UID and using the user UID I am Querying the Firebase Database to get the user against that UID. Once I get the user I am saving the user in the StoreSession Singleton Class for the future use in the app.

After the user logs in or signs up successfully I move him to the HomeVC screen.

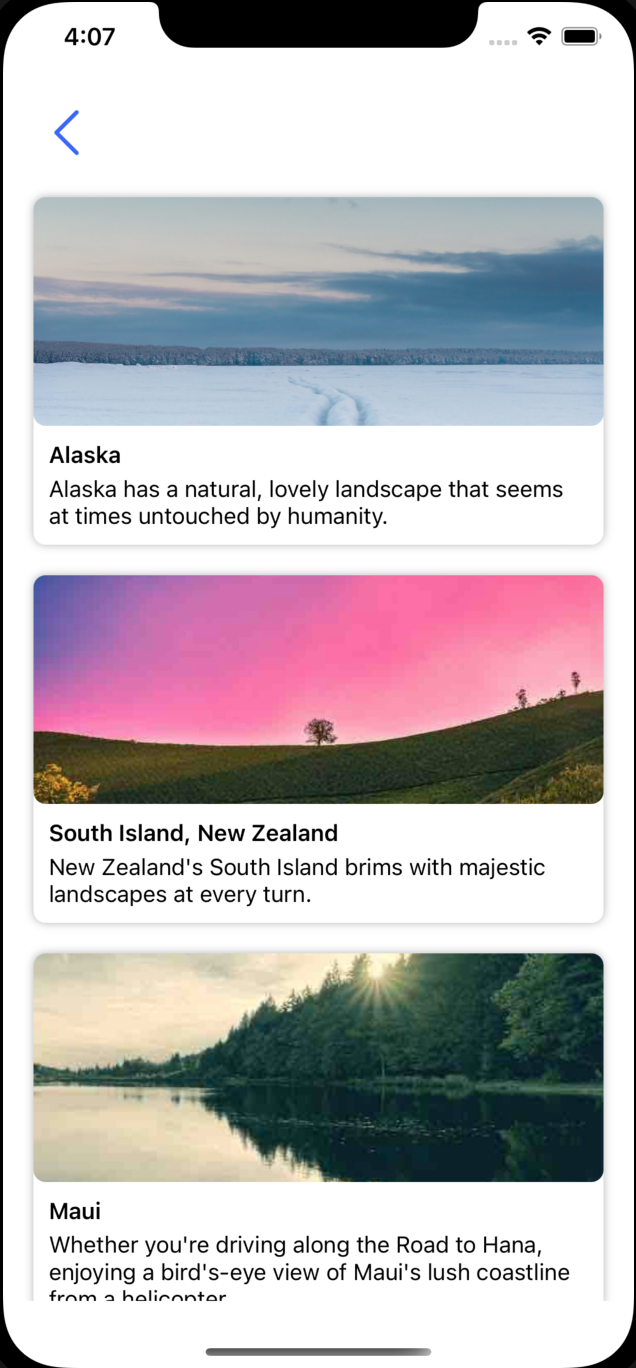
Home Flow:



Home screen based on 4 options, which are Destination List, My Profile, My Destination and Add Destination.

Destination List:

This is the list of the destinations which are added by the users.



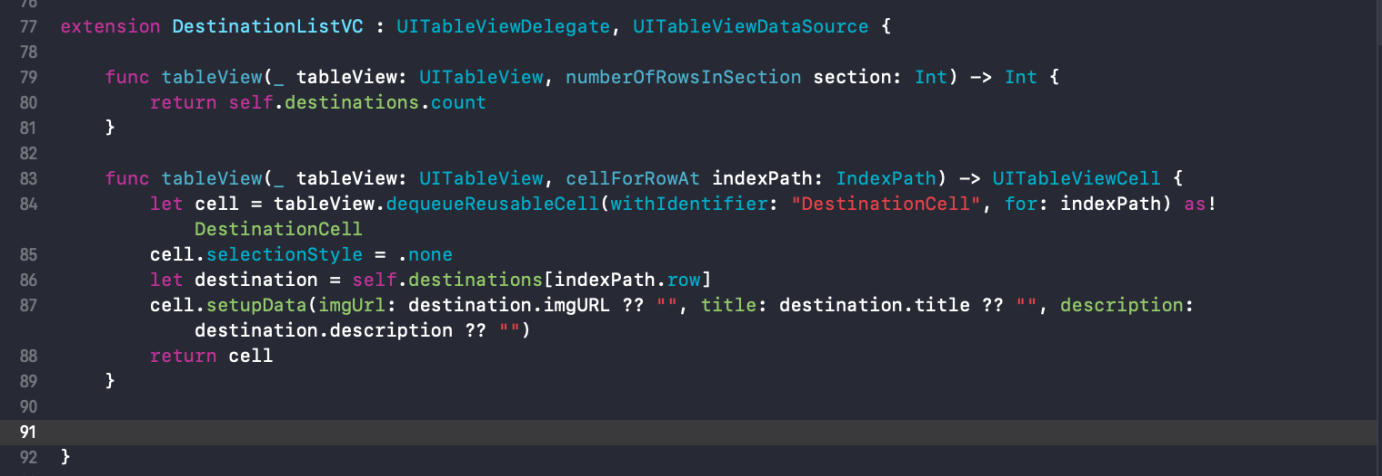
This list is created by using UITableView inside the DestinationListVC. And the cells are auto growing according to the content size.

Populating Destinations:



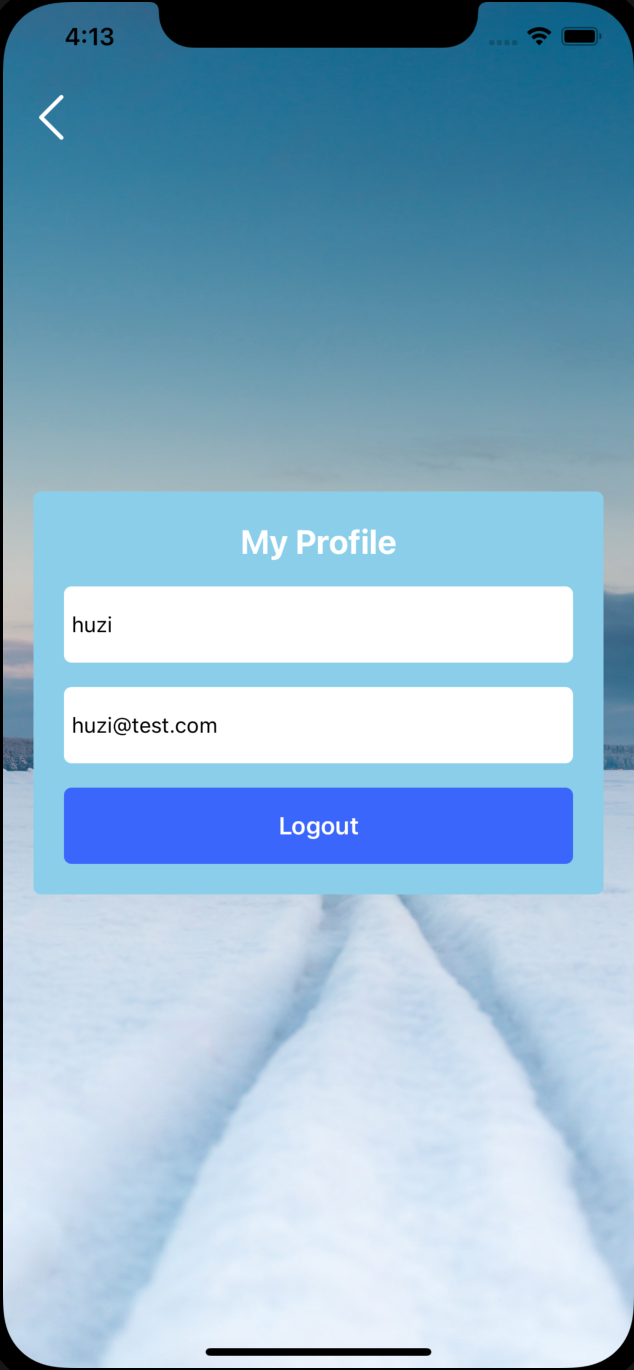
In the DestinationListVC I am again calling the Firebase Database APIs functions to get the list of the destinations. Once I have the list I am loading and showing it in the TableView.

TableView Delegates:



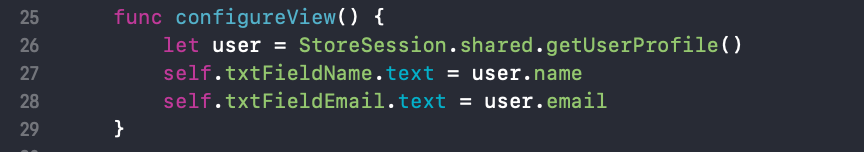
I am using tableview delegate function to populate the listing. These functions are automatically called.

My Profile:



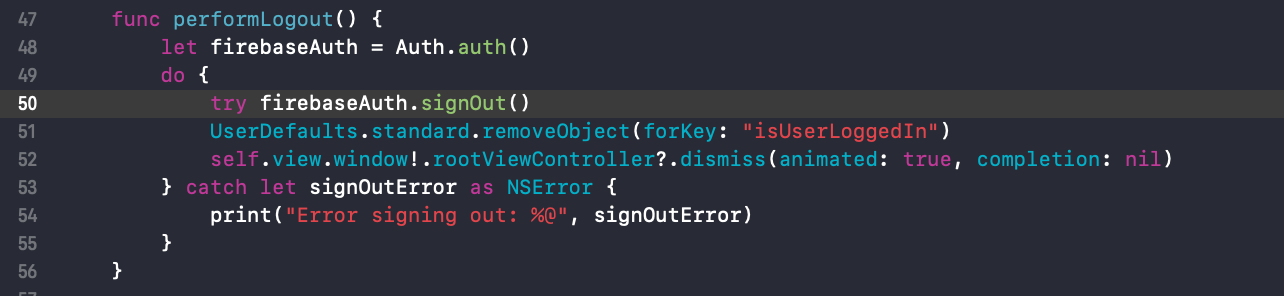
In my profile screen I am showing the profile details of the user which I have saved in the StoreSession class after the login and sign up.

Populating User Data:



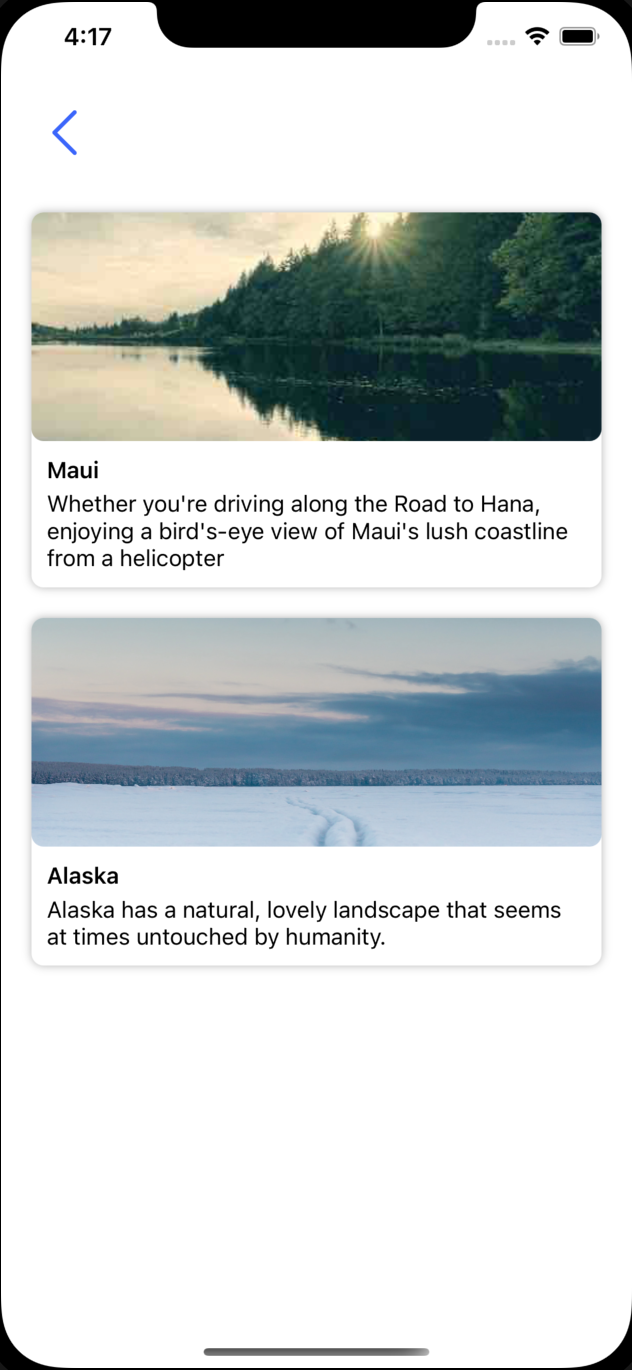
Populating user data from StoreSession class.

Logout User:



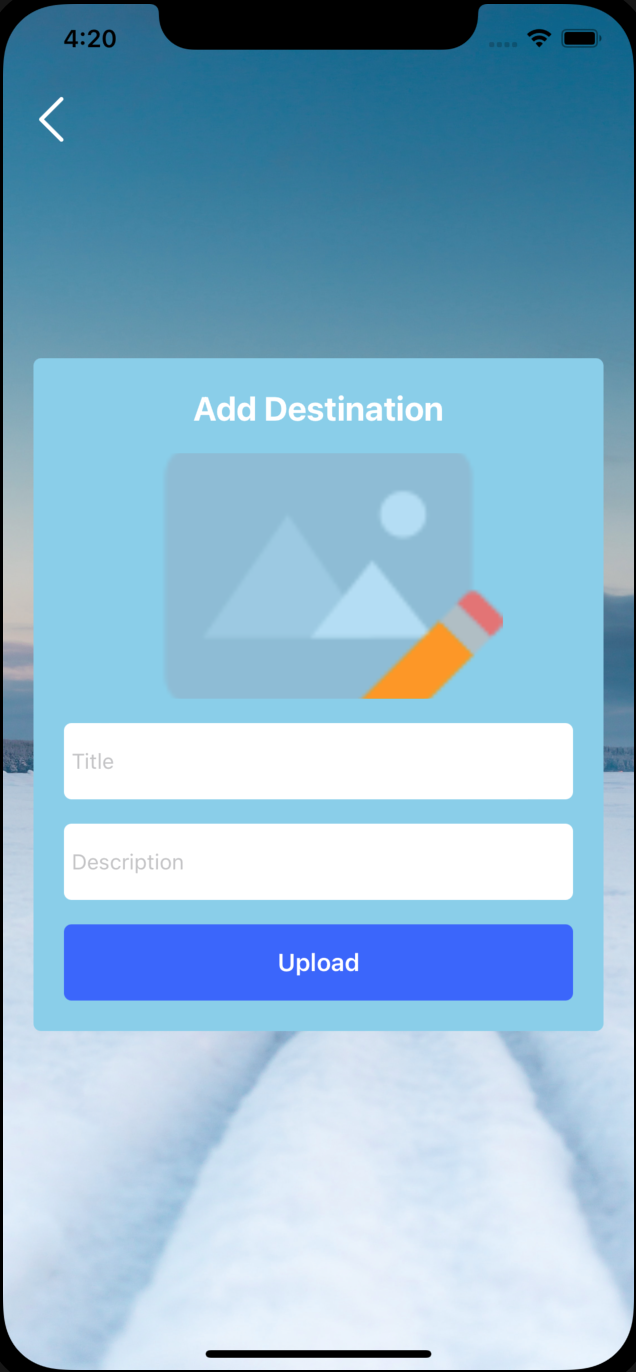
To logout, Firebase Authentication APIs provide the SignOut method and after successfully signing out, I am navigating the user back to the first screen which can be login or signup.

My Destination List:



This screen is very similar with the All Destination screen, the main difference is just that in this screen I am querying the firebase database to just get the Signed In in uploaded lists.

Add Destination:



In this screen I am uploading the picture and saving the destination related data, which in Firebase the table name is Destinations. Pictures are saved in the firebase Storage.

Taking picture from Camera or Gallery:



This is the delegate function of the image picker, when any picture captured through camera or selected through gallery, this method automatically executes and returns the selected Picture.

Uploading Picture: Text

Description automatically generated

After getting the picture from camera or gallery I am uploading that picture to Firebase Storage and getting the Downloadable url for future use.

Adding Destination:

Text

Description automatically generated

After uploading the picture and getting the url, I am saving the rest of the destination data, Title and Description if Destinations Table in Firebase Database.

I have always done functional documentations for my assignments as I did now, as I wasn’t sure if a business requirements one was needed, I learnt firebase much more than sqlite so I used it instead for this assignment.