Project #4:

Implementation and Partial Demo

GO

Standard Addressing System

1. Project Description

GO (Standard Addressing System), is a mobile application which will generate unique addresses for the user. This address is similar to the standard address used in developed countries around the world.

Sign In/ Sign Up: For users to claim the address they must sign in. First time users should sign up and register their account. Once logged in, the user will be guided to the navigation page.

Claim Address: On the navigation page, users can zoom into their home and use the marker to locate their home. Once the user drops the marker to their desired location an address will be displayed. The address will be a standard address with their house number, city name, province name, country name and zip code. Users will be able to claim the address by pressing the claim button on the navigation page.

User's Profile: After claiming the address user will be able to view their address and account information in the user's profile page. The portal button in the navigation page will guide users to their account.

Navigation: All the signed in users will be able to use the navigation system. They must enter the standard address to the field displayed as start destination and end destination. Upon entering this information users will be able to view the route and navigate by pressing the start button.

Upon successfully testing this app, the app should solve the major global problem of standard addressing. Governments spend millions of dollars to create a street address. Our system will be a very cheap and reliable alternative. This will also help people to navigate to any location thus opening huge opportunities for businesses.

2. Implementation Overview

- The project had various stages to be implemented. As we adopted the agile methodology in order to complete the project, we broke the project GO into various parts. This helped us to divide the project among each team member and work on our particular topic. By this we would be able to achieve the goal of our project in time. We also created a repository in the github, this will help us to track down each team members accomplishment. Also, our work up to that point would be safe somewhere.

- After we decided what we were going to do for the project, we broke the project into two halves. AS we only had four members in our group, we also divided our group into two.

Two members in the group will start working on User Interface and Database. Whereas, other two members in the group will start work on generating the standard address.

- The tools we used for the overall project are:
  - ❖ Android studio
  - ❖ Firebase
  - ❖ Google API
  - ❖ Java programming language
  - ❖ Qgis
  - ❖ Python

3. Project implementation update

Currently the project is well towards achieving its goal. Major functionalities have all been implemented. Currently we are working together on importing all the generated standard addresses to the firebase. Both the teams have to coordinate to make sure the proper standard address has been generated in the correct region. Once the importing is done then the navigation page can use this address to display the address and provide a navigation route.

4. Functionalities Implemented

GO application will have the following functionalities:

Application Functionalities: The functionalities here include all the cases we have implemented in the front end and back end. Front end functionalities handle all the user interactions while backend functionality handles the logics behind user interaction and database interaction.

1. SignIn/SignUp
   Firebase Authentication is used for secure sign in. User's name and email is used for signing in and signing up. The sign-up data is stored in firebase and verified while signing in

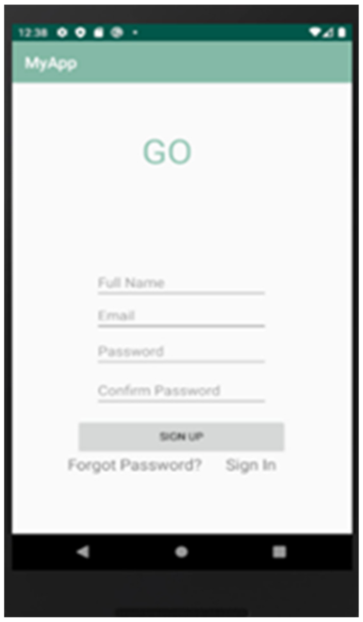


Fig: Sign In

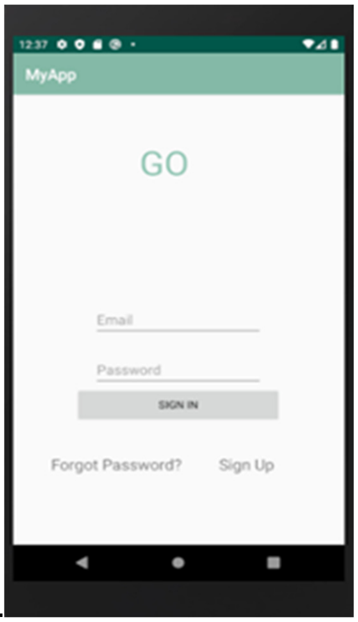
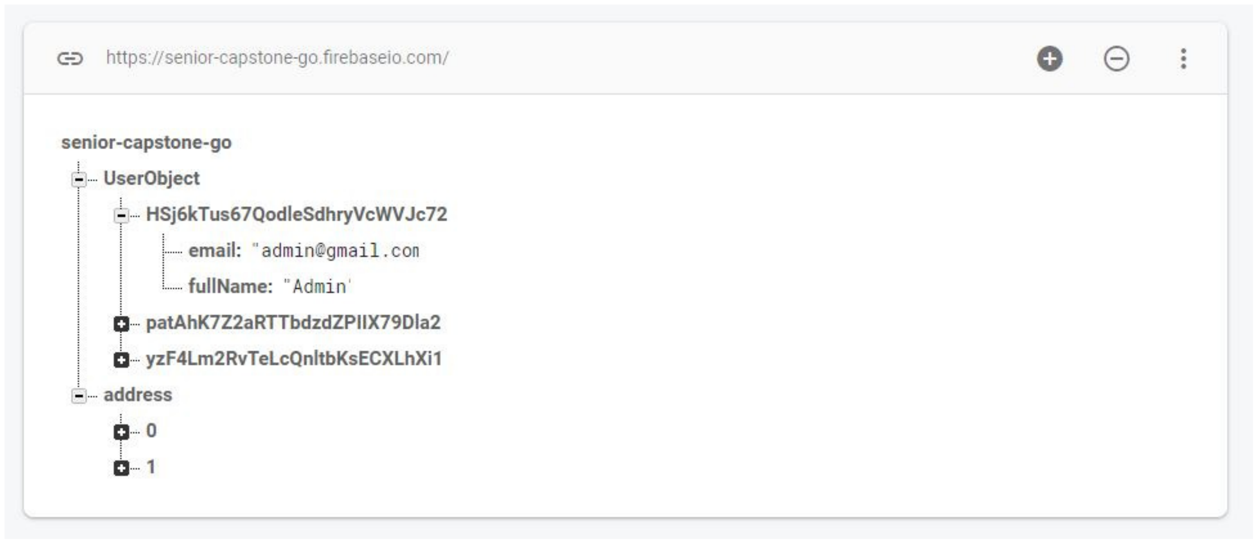


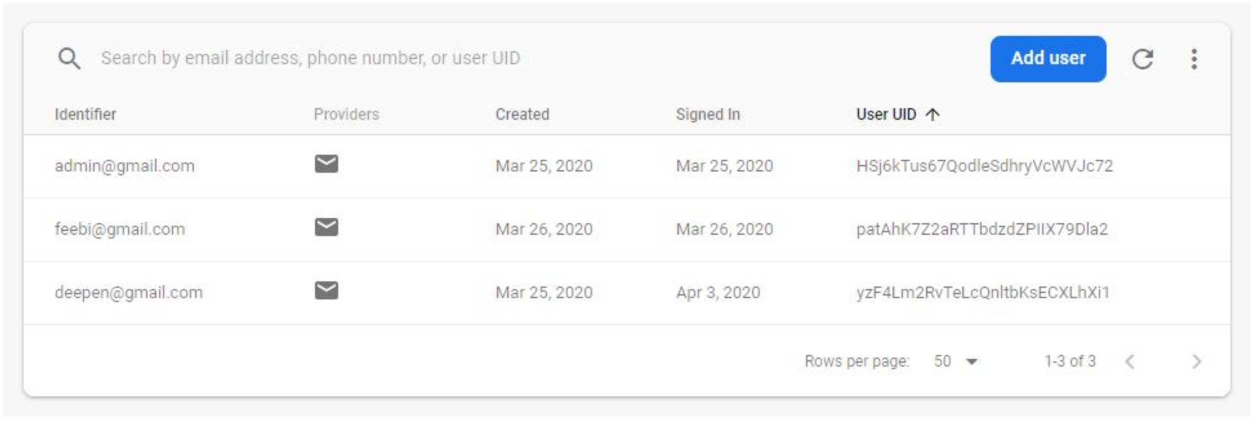
Fig: Sing Up



```
https://senior-capstone-go.firebaseio.com/

senior-capstone-go
  UserObject
    HSj6kTus67QodleSdhryVcWVJc72
      email: "admin@gmail.com
      fullName: "Admin'
    patAhK7Z2aRTTbdzdZPIIX79Dla2
    yzF4Lm2RvTeLcQnltbKsECXLhXi1
  address
    0
    1
```

Fig: Firebase Database Showing User Object


Search by email address, phone number, or user UID

Identifier	Providers	Created	Signed In	User UID ↑
admin@gmail.com	✉	Mar 25, 2020	Mar 25, 2020	HSj6kTus67QodleSdhryVcWVJc72
feebi@gmail.com	✉	Mar 26, 2020	Mar 26, 2020	patAhK7Z2aRTTbdzdZPIIX79Dla2
deepen@gmail.com	✉	Mar 25, 2020	Apr 3, 2020	yzF4Lm2RvTeLcQnltbKsECXLhXi1

Fig: Firebase Authorization Implementation

2. Claim Address
   Upon claim standard address data from firebase is received and then stored in the user object along with their name and email.

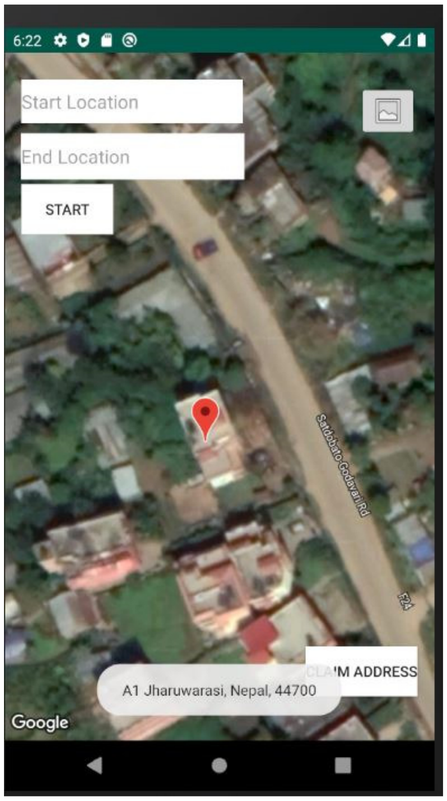
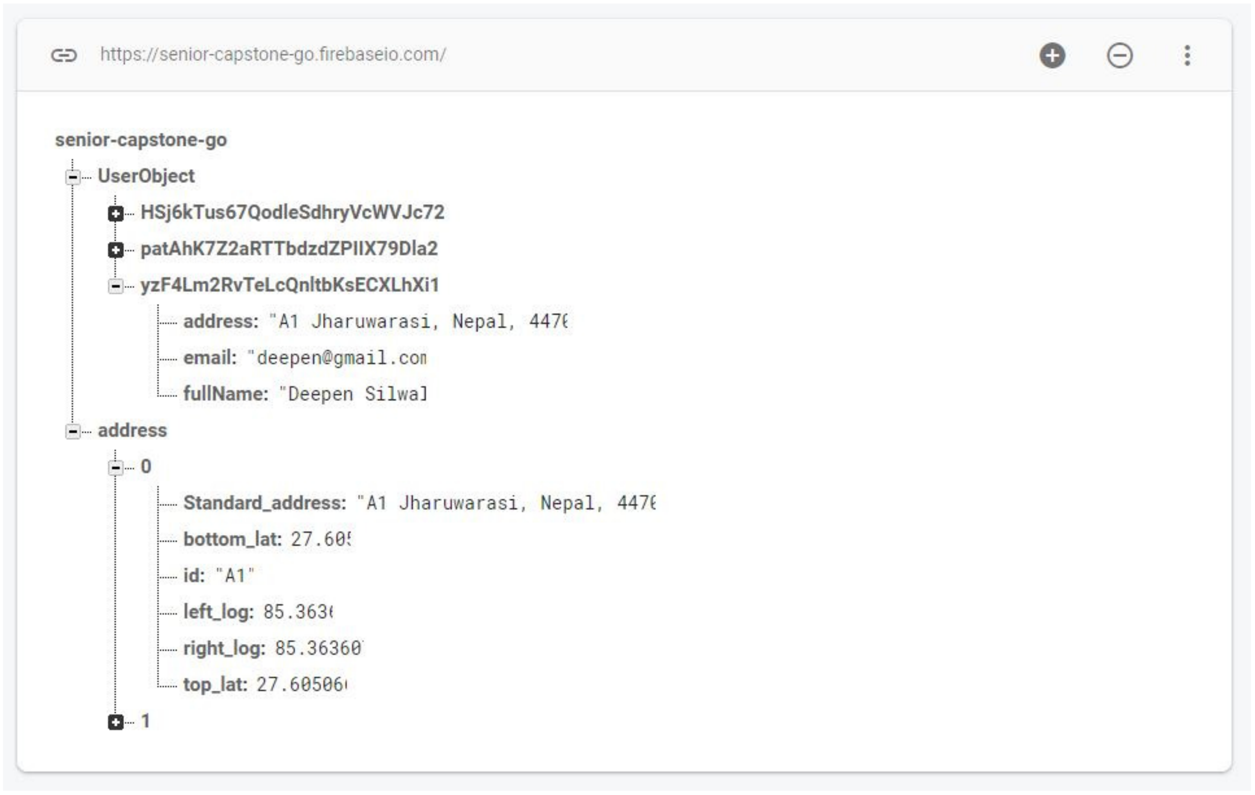


Fig: Address Displayed for Location Marked

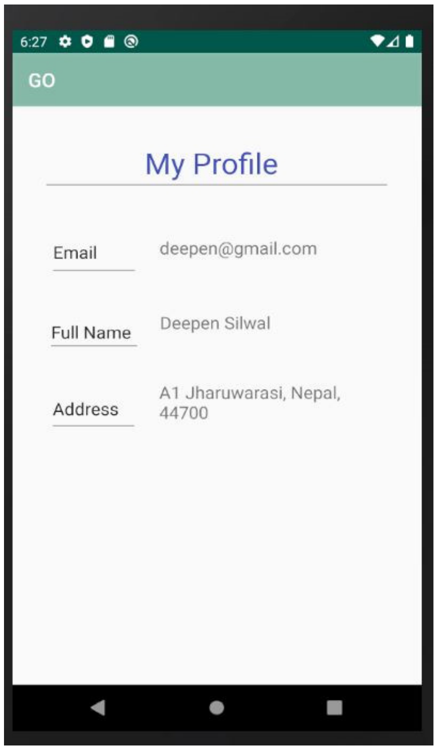
When the user clicks on the claim address for the displayed address. The address is retrieved from the address object and stored in the user object for the signed in user.


```
https://senior-capstone-go.firebaseio.com/

senior-capstone-go
  UserObject
    HSj6kTus67QodleSdhryVcWVJc72
    patAhK7Z2aRTTbdzdZPIlX79Dla2
    yzF4Lm2RvTeLcQnltbKsECXLhXi1
      address: "A1 Jharuwarasi, Nepal, 447(
      email: "deepen@gmail.con
      fullName: "Deepen Silwal
  address
    0
      Standard_address: "A1 Jharuwarasi, Nepal, 447(
      bottom_lat: 27.60!
      id: "A1"
      left_log: 85.363(
      right_log: 85.36360
      top_lat: 27.60506(
    1
```

Fig: Address displayed Is stored in the User Object

3. View Profile
   Signed In users will be able to view their profile with information such as their name, email and address. Address is only displayed if they have claimed the address. To view the profile, users need to click on the profile icon located at the top right of the navigation page. The information about the user is retrieved from the firebase database user object for signed in user.



My Profile

Email	deepen@gmail.com
Full Name	Deepen Silwal
Address	A1 Jharuwarasi, Nepal, 44700

Fig: User profile

4. Navigation
   The navigation uses the standard address generated by the app. You need to know the address from our system so that you can get the route. For now the app will only display the route. Turn by turn navigation cannot be demonstrated as the region for this system is in Nepal

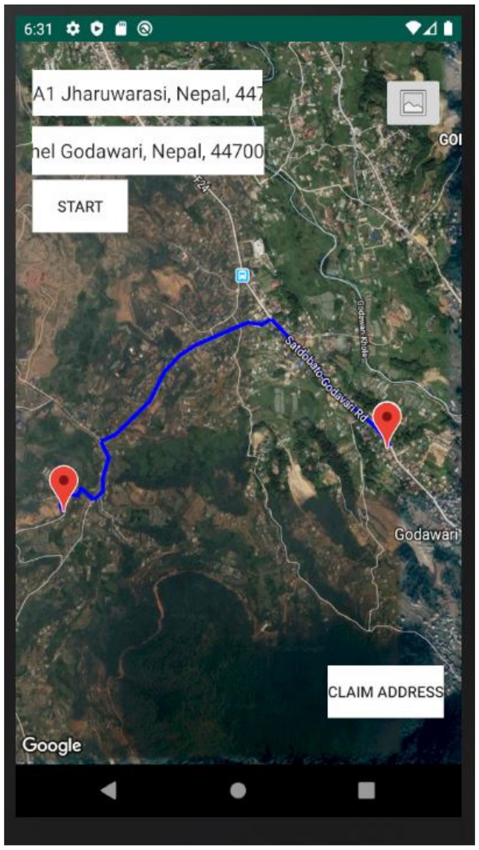


Fig: Navigation Route Displayed Using Standard Address

Standard Address Functionalities: These are the functionalities used to generate the standard address.

1. **Boundaries selected using QGis**

   The main reason for choosing QGis is that it provides the vast possibilities of manipulating imported maps and geographic information. QGis provides the vector layer and raster layer functionalities to divide the map into grids. Vector layers are layers containing three different feature types, points, lines or polygons. A raster layer is quite different and is made up from pixels encoded with color information and geographic information.

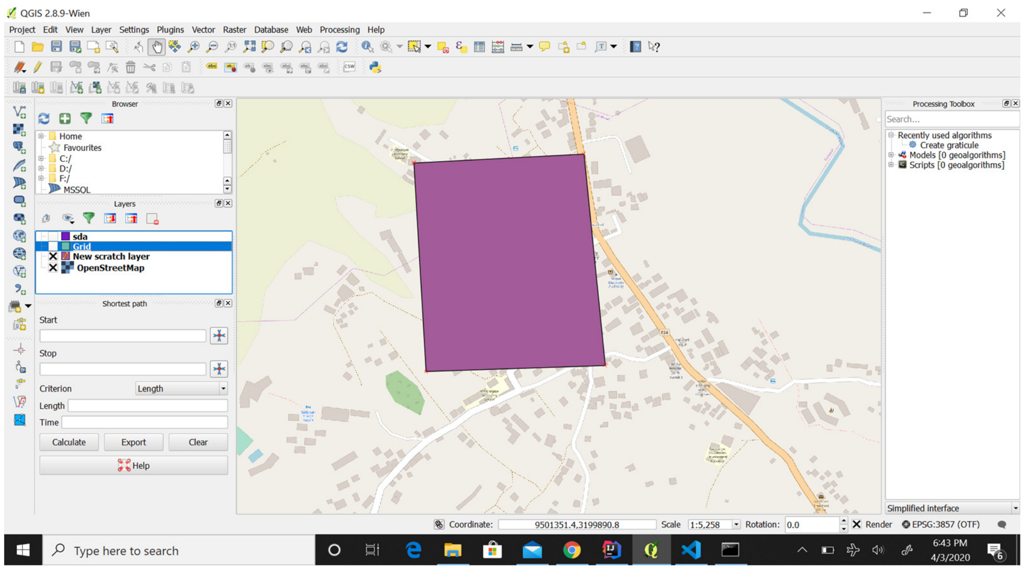


Fig: Boundaries selection using QGis

To divide the map into grids we used the vector layer functionality and polygon feature in it. A polygon is an area outlined by the series of points. First of all, we chose the small area from the map and created the polygon in it. It gave the scratch layers of points in the map.

2. Matrix of Latitude and Longitude is obtained

The scratch layers from the map is then converted into 3*3 m grids using the graticule functionality provided by QGis. From this we obtained the latitude and longitude points in matrix form. All the data is then saved into the csv file on the local computer.

3. TopLat, BottomLat, Leftlong, RightLong generated

The four points in the matrix from, top, bottom, left and right are exactly at 3 m apart. One random point in between these matrix forms is chosen for generating addresses at that point in that area.

4. Standard Address Generated

The random point obtained from the matrix points is then passed into the google API. Google API gives the normal address (general city name) in that point which is then saved into the corresponding point in that csv file. We have given each grid a unique identity. The unique identity combined with the google generated address now becomes the standard address for that point.

5. Functionalities currently being implemented
Currently we are working towards importing generated addresses to the firebase. We are testing the standard address to make sure these addresses are correct. The expected output format for our address is House Number, City Name, Country Name, and Zip Code. We are testing to see if our generated address matches this format.

Also, we are testing whether the matrix of latitude and longitude forms 3m square or not. Each 3m square box will have an address. Both of our divided groups are testing these criteria. Upon completion of testing the json file of address will be updated in firebase. These data will be then used in the navigation page to display the address and to display the navigation route.

6. Functionalities Remaining
There are some functionalities that need to be implemented. These functionalities will be completed by final demo.

Claim Address Error: We will need to implement a function for users to only claim one address. If the user has already claimed the address and if they again try to claim the new address they need to be notified and warned.

User Permission for Location: We need to ask users for their permission to use their location. Currently we have provided the user's location manually. The permission will include whether the user would like to give us the permission to use their location only while using the app or all the time or they don't want to allow us to use their location.

Display Address: Since we are working together in testing the standard address, we only have generated the standard address for some sample locations. Once the testing is done and the address is imported on the firebase database, we will be able to display the address and provide a navigation route of the entire region we have selected.

Display Boundary: When the user points the marker to some location, we would like to display the square box for the address. As we will have a standard address for 3m square grids, we would like to display the grid along with the address.

7. Implementation Issue
There are some limitations in our app that we will not be able to implement. Some of these are functional limitations while some are testing limitations.

Turn by turn navigation: We won't implement turn by turn navigation currently as we will not be able test this feature. The reason being is the region for which we are developing standard address is in Nepal. So, for now we will only display the route between two addresses.

Standard address for only selected region: The standard address is only displayed for selected region. The app can display addresses and generate routes only within this region. The project is to showcase the proof of concept, so implementing in the larger regions will have very demanding challenges of maintaining larger databases and powerful servers.

Claimed address is correct or not: As we are not using machine learning for this project, currently we have no way of testing whether the user has claimed the right address or not. This functionality is beyond this project and should be implemented for business development of the app.

8. Each member contribution

For better handling and distribution of work we have divided our group further into two groups. First group will be working to implement all the applications front end and back end functionalities while the second group will be working to implement standard addressing functionalities.

First Group:

[redacted]**:** Feebi Sharma is working towards the front end of the app. She is responsible for the overall front-end design of the app. The use cases she has implemented are: SignIn, SignUp, User Profile, Forgot Password, and Navigation. She is currently working on implementing all the generated standard addresses in the navigation page.

[redacted]: Deepen Silwal is mainly working towards the backend of the app. He is responsible for managing database and backend design. The use case he implemented are: Navigation, Database Design, Object Dataset, Address Object Dataset, Firebase Authentication and Firebase Realtime Database.

He will be working with Feebi Sharma to handle all the backed issues. He is currently working with a second group to import standard addresses into firebase.

Overall, the first group needs to implement all the necessary UI interactions that take place at the front end as well as the back end.

Second Group:

[redacted]**:** working on the Qgis application to divide the region into a 3 square meter grid. A layer of the desired location is created from the physical map which is divided into the grids using the grid creation tools. Latitude and longitude attributes are added to the grid. Then, the dataset is exported as CSV with all the information on latitude and longitude.

[redacted]**:** working with the dataset obtained from Bijay. Random latitude and longitude of every grid is obtained and stored in the new row of the CSV file. Using python, every latitude and longitude is passed into the loop which obtains the address provided by google map. Google reverse geocoding API is used to obtain this information. The obtained address is stored in the CSV file. A unique code for each grid is created and combined with the address obtained from the Google API to get the standard address.

Second groups overall task is to select a region and generate the standard address for that region.

9: Links to the Demo:

Google Maps Address Display: The first link shows and explains how the google maps displays the address in the region we are trying to implement this app.

- **https://www.youtube.com/watch?v=CpVh0Q0TJ-8&feature=youtu.be**

GO APP Demo: The link below is the actual demo of our app with all the functionalities.

- **https://www.youtube.com/watch?v=JxHy9ICm0_M**