

## Capstone\_Stage 1

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

**GitHub Username:** [eslammedhat765](#)

## Car Security

### Description

Car security app is used for secure your own car from thieves and track the location of the car and put marker of its current location on Google maps, the app will get the current location of the mobile and put marker, then locate the car location and put another marker, then draw an route between the car location and the mobile to calculate the distance, also the app give a list view of emergency numbers, also you can make a profile and add photo from gallery.

### Intended User

We target everyone have a car and need to secure it with mobile app.

## Features

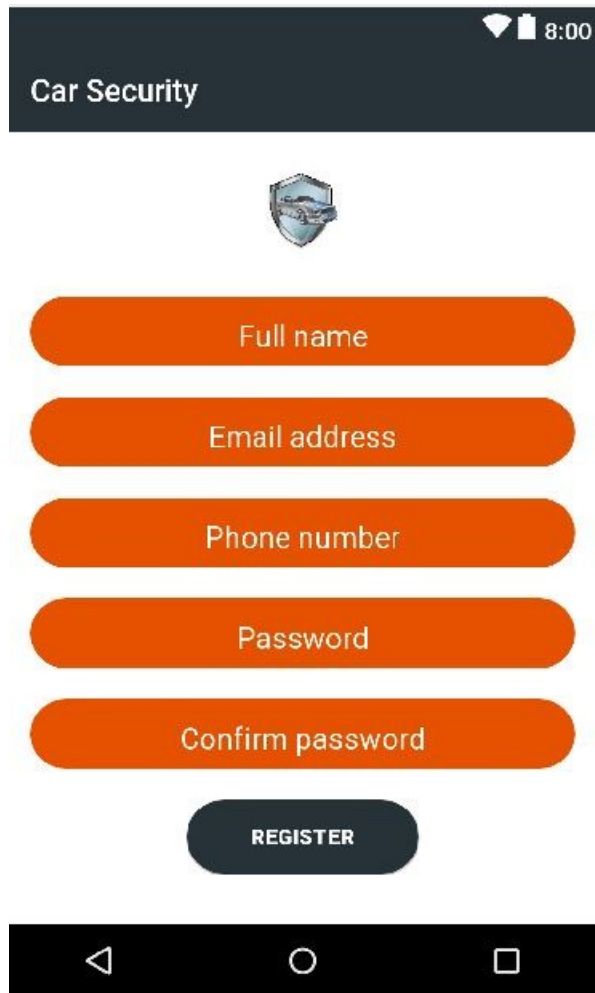
- App is written solely in the java programming language.
- Android Studio version 3.1.3
- Gradle version 4.4
- Tracking Location with Google maps.
- Emergency numbers for help
- User profile and upload pictures
- Google Maps API
- Firebase Authentication
- Firebase Database
- Firebase Storage
- Adding UI Widget
- Java language will be used for development in all this designs
- All versions of gradle and android studio are stable.
- Adding AsyncTask

## User Interface Mocks

### Screen 1

- in this screen (Login Activity) with Firebase authentication, user can log in with his own account to can access to the app and start tracking the car.
- in this screen also there is a validation to double check the username and password and make sure that data exist in the firebase database.

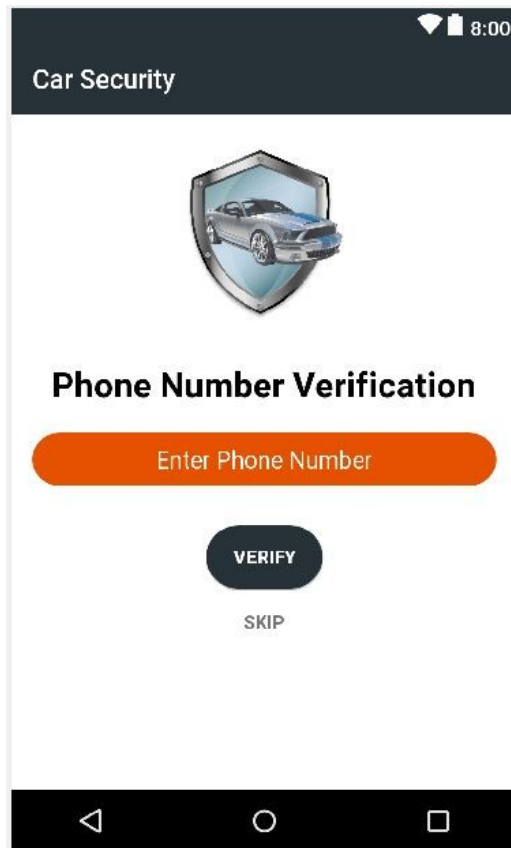
## Screen 2



-In this screen (Registration activity), if the user didn't have an account, he can register a new one and create a database using firebase database to store user profile data.

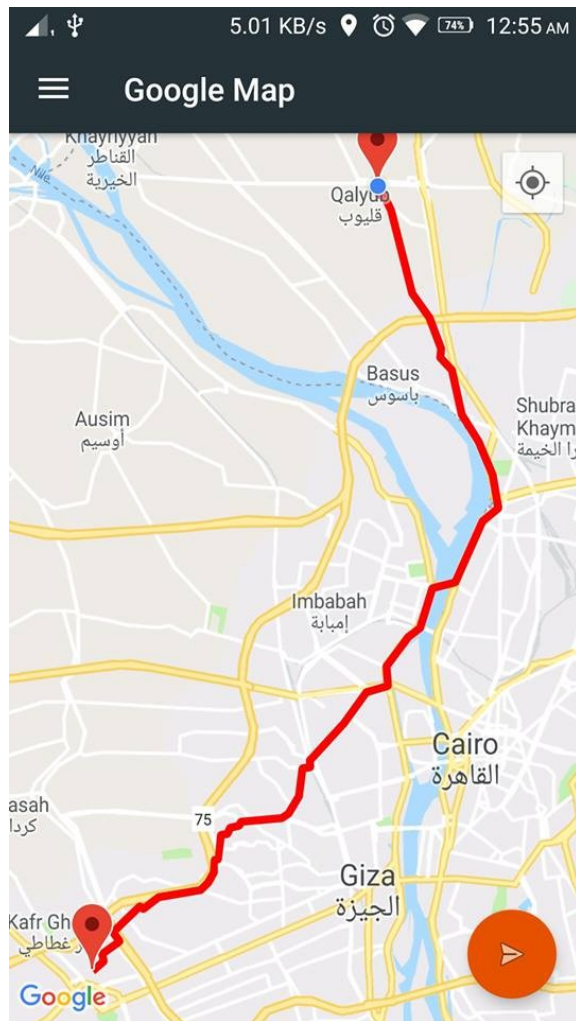
-Also there is a validation to double check the data is filled correct and the password is Strong.

-Screen 3



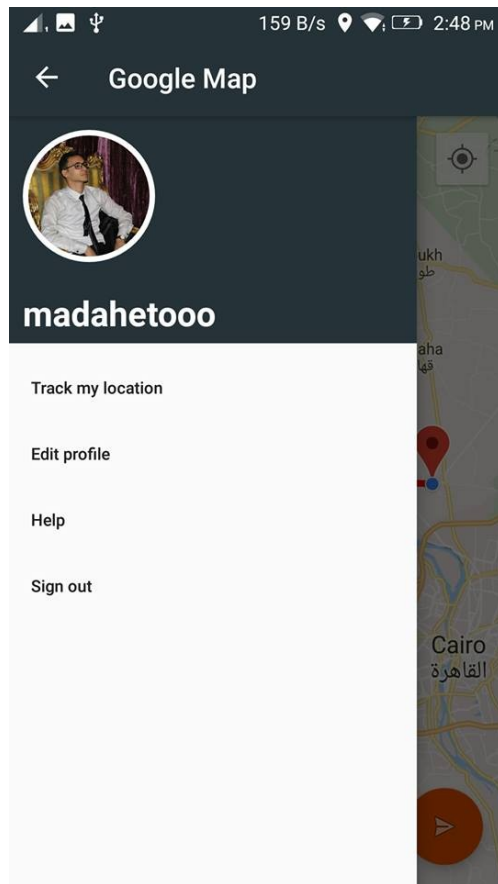
-in this screen(Verification activity ) user can verify his own phone number by writing it and click on verify then wait an SMS with Code, and once the app detect that the message sent , will navigate it directly to Maps activity.

-Screen 4



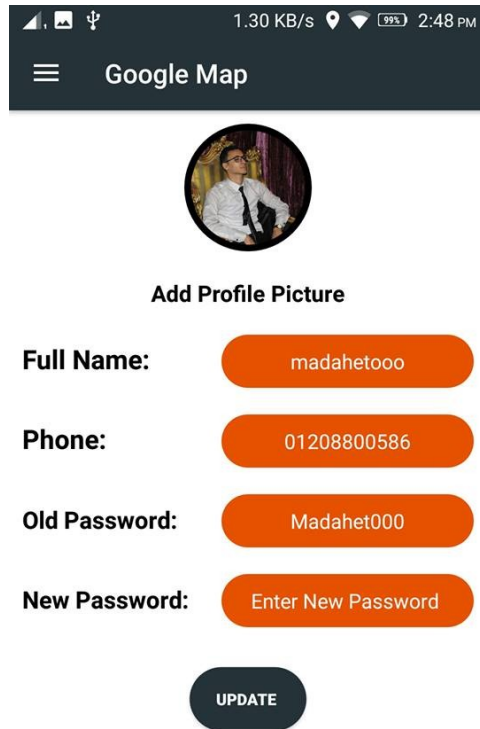
-in this screen user can track his device location and put marker , then track car location and put another marker , then draw an route from device to car to calculate the distance .

-Screen 5



-in this screen (Navigation drawer ) it make it easy to transfer between activities and give the app good feeling about UX ,also it contain username and default photo and will update the data once the user update it from Edit profile activity.

## -Screen 6



1.30 KB/s 99% 2:48 PM

Google Map

Add Profile Picture

Full Name: madahetooo

Phone: 01208800586

Old Password: Madahet000

New Password: Enter New Password

UPDATE

-in this screen ( Edit profile ) user can upload his own photo from gallery and view his registration data and can change it ,then click on update to upload the data on firebase storage and change the data on firebase database .

## -Screen 7

354 B/s  99% 2:48 PM

Police : 122

Ambulance : 123

Firefighter : 180

Winch 1 : 01225066250

Winch 2 : 01002971775

Winch 3 : 01001967743

Winch 4 : 01097950005

Winch 5 : 01220787081

Winch 6 : 01274644112

Winch 7 : 01288811148

Winch 8 : 01148023313


Winch 9 : 01224552222

-in this screen (Help activity)

User can get emergency numbers like: police, ambulance, firefighter, etc, for emergency situations.

## -Screen 8

Car Security



Username

Email

latitude



Now this widget activity will show the username of the user and his email address and Gson data from API with latitude and longitude using ASYNCTASK.

## Key Considerations

### How will your app handle data persistence?

- My app can handle data beginning: firebase Authentication for authenticates users with email and password and phone .etc...
- Also will handle the registration data with firebase database.
- store images in edit profile activity with firebase storage.
- I added all strings to strings.xml file and allow the RTL support.

### How the app will support accessibility?

- I added android:focusable="true" into the layout to avoid missing data and make the app more accessibility .

### Describe any edge or corner cases in the UX.

- I handle it with Alert dialogs to ask the user if he intent to close the app to avoid closing it with fault.
- if the user logged in with email and password and close the app , if he back again he will open the app directly without asking for username and password again , and it is with firebase on state listener to check the current situation if the user signed out or not .

### Describe any libraries you'll be using and share you're reasoning for including them.

- 'com.android.support.constraint:constraint-layout:1.0.2' >>for using constraint layouts in my design to be adaptable for more than one screen .
- 'com.android.support:design:26.1.0' >> to can use material design in the app.
- 'com.google.android.gms:play-services-maps:11.8.0' and 'com.google.android.gms:play-services-location:11.8.0' >> to be able using google maps service
- 'com.squareup.picasso:picasso:2.5.2' >> for media .

- `'de.hdodenhof:circleimageview:2.1.0'` >> for customize the photo in a circle image with borders.
- `'com.google.firebase:firebase-auth:11.8.0'` >> to authenticate the user with email and password , phone , etc....
- `'com.google.firebase:firebase-storage:11.8.0'` >> to be able to storage data(Files) like photos and more.
- `'com.google.firebase:firebase-database:11.8.0'` >> to be able to store data (TEXT) in the database like registration data .
- `'com.google.firebase:firebase-core:11.8.0'` >> to allow analytics for the app .
- `'com.macroyau: thingspeakandroid:0.2.2'` >> to be able to connect to THINGSPEAK servers and get an API link for get the longitude and latitude to draw the route.
- `'com.squareup.retrofit2: retrofit:2.4.0'` >> to be able to use web services like POST and GET .
- `'com.squareup.retrofit2: converter-gson: 2.3.0'` >>to convert the response data from API to gson format .
- `'com.google.code.gson: gson:2.8.2'` >> to be able to interact with gson data in the app .
- `'com.akexorcist: googledirectionlibrary:1.1.1'` >> to draw routing in the google maps .

**Describe how you will implement Google Play Services or other external services.**

1-I used "Google Maps location service" to be able to use Google maps and track location for helping the user to find his car.

2- I Used "Firebase service" to be able to get features of the firebase like: firebase authentication, firebase database, firebase storage and more ....

## Next Steps: Required Tasks

### Task 1: Project Setup

First of all I start with adding libraries beginning from Google maps api , then firebase to be able to use their features .

- Configure libraries (Google Maps, Firebase, Retrofit, ThingSpeak, etc...)

### Task 2: Implement UI for Each Activity and Fragment

First of all start with design Wire Frame for the app , then start implement it with XML.

- Build UI for SplashScreen Activity.
- Build UI for MainActivity (Log In Activity)
- Build UI for Registration Activity.

- Build UI for Verification Activity.
- Build UI for Google Maps Activity.
- Build UI for Edit Profile Activity.
- Build UI for Help Activity.
- Build UI for Widget Activity.

### **Task 3:**

The third task is to get our device location using GPS service and put marker on the Google maps.

- Create red marker for current location device.

### **Task 4:**

The fourth task is to get the car location from Thing Speak API and put another marker on the map to show the car location on Google map.

### **Task 5:**

The fifth task is to create a route between current device location and car location to calculate the distance between them.

### **Task 6:**

Add edit profile activity to let the user able to change and update his new data and upload profile photo.

### **Task 7:**

Adding help activity with emergency situation numbers to let the user use it in emergency situations.

### **Task 8:**

Adding Sign out button to let the user signing out from the app.

### **Task 9:**

Adding UI Widget activity for users.

