Software Requirement Specification Document for "Navigation Application with Safety Features"

Andrew Usama, Habiba Elwany, Monica Medhat, Moustafa Waly,
Eng Youssef Mobarak, DR. Mohamed ElGazzar

April 22, 2019

1 Introduction

1.1 Purpose of this document

The main purpose of this Software Requirements Specification document is to illustrate our system, Navigation system with safety features, requirements. That is mainly the detection and classification of road safety, like the risk probability, to tackle some of the main causes that lead to high accident rates which are junctions, the curvature of the roadway and high traffic flow. Our application shall be a mobile application, that will be built over the "Android studio" IDE platform. we shall provide a description of each single stage input, output, processing and algorithms used in these stages. Along with illustration for each stage requirements and development process. And stating system constraints, that we are going to face during development and how should we interact with them.

1.2 Scope of this document

This document targets users aiming to have safer trips by using our application to help them detect the safety features they need to accomplish that. Allowing the user to take the safest route with fewer risk factors such as Junctions, The small curve radius, security and proper lighting. Which aims to add some, much needed, safety features to our day-to-day navigation.

1.3 Overview

t our system is a user-friendly navigation system, that implements the safety feature, allowing the user to take the safest route with fewer risk factors that could lead to accidents. This application focuses on keeping the user updated with regards to anything from blocked or closed roads, bridges, to even accidents in these places. The application will gather this information through feedbacks

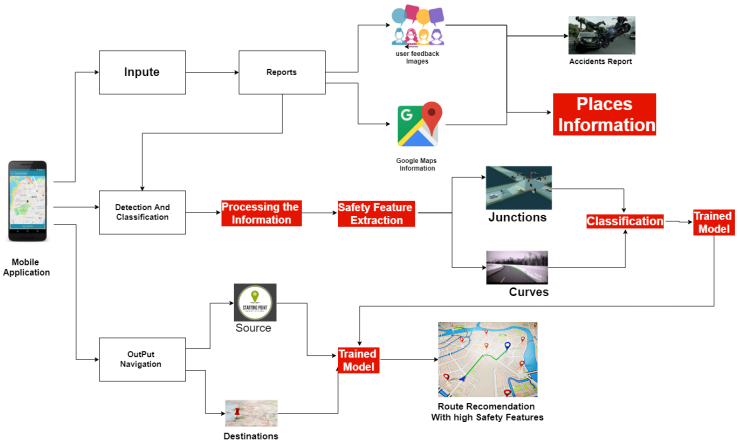


Figure 1: Project Overview Diagram

collected from multiple users, google maps information and CNN. The Safety parameter which are junctions, curves and accidents rate, are going to be determined based on several aspects that will be analysed by an artificial neural network known as CNN and R-CNN to determine the safest route for the user. Then, it will use these safety factors to provide a safer road for all users alike.

1.4 Business Context

As the process of Navigation systems always faces several challenges aiming to provide the user with the shortest path. While completely neglecting the element of Safety as it is an important factor for guaranteeing the safety of the public drivers and pedestrians alike through navigation. According to Egypt Independent, a news website, in 2017, the number of car accidents that occurred recorded 11,098 with 3,747 deaths, 13,998 injuries, and 17,201 damaged vehicles [1]. So, our proposed system aims to increase the safety of the people through day-to-day navigation by gathering several safety features that will help us provide the user a safer path.

2 General Description

2.1 Product Functions

The general functionality of our system is to provide a safer path for all users to be able to perform their navigation successfully, safely and securely. we shall also give our users a certain percentage that represents how safe some areas are over other areas. all while using Google Maps API as a background to guarantee that all users who know how to use Google Maps can also use our application.

- Our application shall provide a safer path

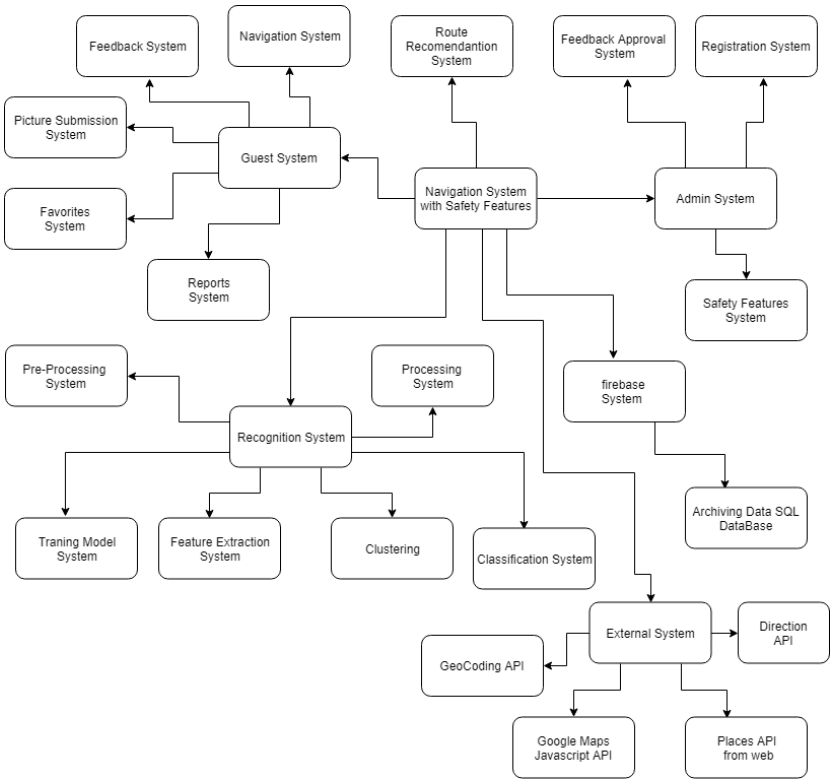


Figure 2: Context Diagram

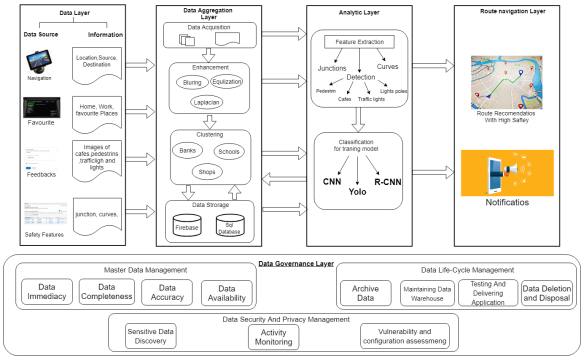


Figure 3: Block Diagram

- Our application shall provide the user with everything that they get while using Google Maps including but, not limited to, Setting "Home", "Work", & "Favorite" locations.

- our application shall have a web-based back-end to control and work with users data to further improve the service we provide via taking the user's feedback that range from pictures to reports in order to improve our CNN
- Our application shall allow users to submit several reports including but, not limited to, "New Bank","New Police Station","New School","New Shopping district", and "Incident reports".
- our application should allow users to submit pictures of roads for our back-end to process whether a street is busy and hence, safe or not.
- Our application shall receive User-Feedbacks in order to better assess whether an area is safe from the users point of view.
- Our application shall be able to identify how many junctions are on the user's path.
- Our application shall be able to send notifications to the user to warn them about dangerous junctions
- Our application shall be able to identify how many narrow turns are on the user's path.
- Our application shall be able to send notifications to the user to warn them about dangerously narrow turns.

2.1.1 Data layer

- Navigation: the user enters the source and destination of the trip.
- Favorite: the user enters the work,home, and favorite places
- Feedbacks: the user uploads the feedbacks of accidents reports, banks, schools, and shops.
- Safety Features: the user add a new safety feature, and the admin approves it.

2.1.2 Data Aggregation Layer

- Data Acquisition: we get the data from the data layer which was submitted by the user
- Enhancements: using of enhancements algorithms to ready the images for detection and remove the noise.
- Clustering: cluster the banks, schools, shops, and accidents report to classify the safety of the routes.
- Data storage: saving the data in Firebase and archiving in SQL database.

2.1.3 Analytic Layer

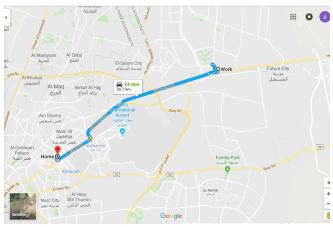
- Feature Extraction: after enhancement, the images, detection, and extraction of the safety features are applied which are junction, curves and accidents rate.
- Classification and training: classification the data using CNN and R-CNN.

2.1.4 Route Navigation Layer

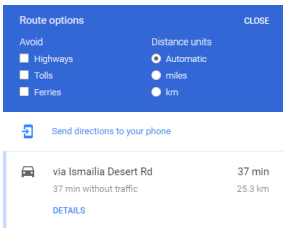
- Route Recommendation: the final output for the user which is the route with the highest safety.
- Notifications: notify the user if the Route safety changes.

2.2 Similar System Information

1. Google Maps: Google Maps is a web mapping service developed by Google. It offers satellite imagery, street maps, 360 panoramic views of streets, real-time traffic conditions, and route planning for traveling by foot, car, bicycle, or public transportation.



(a) Google maps 1

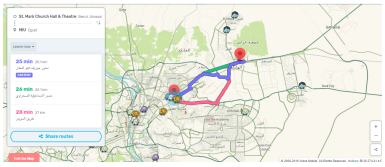


(b) Google maps 2

   - Pros:
     - Can help avoid some unwanted areas like highway
     - Provide the shortest route
     - Allows a user to add favorite places
   - Cons:
     - it doesn't provide any information with regards to newly closed roads
     - Doesn't provide all route at once for the user to choose from
     - Doesn't provide the safest path

2. Waze: is a GPS navigation software app. It works on smartphones and tablet computers that have GPS support. It provides turn-by-turn navigation information and user-submitted travel times and route details, while downloading location-dependent information over a mobile telephone network.

(a) Waze 1



(b) Waze 2

- Pros:
  - The application is organized
  - The application provides all the routes
  - The application provides the user important information like traffic data
- Cons:
  - The Application can be very slow
  - The Application is very childish
  - Hard to navigate using the application
  - Doesn't provide the safest path

What makes our application different from the other applications?

- Our application provides features like clustering areas that are deemed safe such as police station, banks and schools
- Our application provides curve detection
- our application provides junction detection

2.3 User Characteristics

This could be concisely clarified through the P.A.C.T., also known as person people, activities, context and technology, Analysis which will help clarify " People undertake activities, in contexts using technologies." so first we start with:

- People: Our target users are essentially those people who can use Google Maps but instead of finding a shorter path would rather find a safer path for the trip and/or activity theyre undertaking.
- Activities: Our application will be able to undertake the same activities Google Maps can with the exception of shorter paths which will be replaced with safer ones.

- Context: Context: our application can be used in multiple contexts all of which include the elements of a user wanting to take a trip somewhere using an application that considers safety as the primary element affecting route choice.
- Technology: The technology we using is widely available which includes an application that develops an application for Android OS which is the Android Studio IDE and the Google Maps API which is free to download.

2.4 User Problem Statement

Navigation systems always face focus on giving the user the shortest path, while completely neglecting the element of safety. In our project, we know that safety matters and that it varies from one person to another. So, The main problem is the suggestion of the best routes that takes into consideration the safety of the users by factoring in the junction count, crowded areas, and areas that have increased security due to the presence of banks, schools, or police stations.

2.5 User Objectives

What is expected of our system is listed below:

- The application should be fast
- The application shall be easy to use for those with experience with its predecessor (Google Maps)
- Users shall be able to put 2 or more locations and receive, in return, the safest path between the given locations

2.6 General Constraints

Some of the general constraints are:

1. The user's data need to be safe and securely stored
2. The user's trip information needs to be confidential
3. The application is going to require a phone that can at least run our application's predecessor
4. The application will also require as much space if not slightly more than its predecessor
5. There is a time constraint on this project which is that the project is due on 25th of June 2019

3 Functional Requirements

3.1 Set Start Point

3.1.1 Description

The user shall be able to enter the trip's start point

3.1.2 Inputs

Longitude and latitude

3.1.3 Outputs

void

3.1.4 Criticality

10/10

3.1.5 schedule

This is scheduled to be delivered on the 27th of November

3.1.6 Dependencies with other requirements

N/A

3.2 Set Destination

3.2.1 Description

The user shall be able to enter the trip's Destination point

3.2.2 Inputs

Longitude and latitude

3.2.3 Outputs

void

3.2.4 Criticality

10/10

3.2.5 schedule

This is scheduled to be delivered on the 27th of November

3.2.6 Dependencies with other requirements

Set Start Point

3.3 Set Favorites

3.3.1 Description

The user shall be able to enter their favorite location and as many as they want in case they want to use those locations as a trips destination in the future

3.3.2 Inputs

Name, Longitude, latitude

3.3.3 Outputs

void

3.3.4 Criticality

8/10

3.3.5 schedule

This is scheduled to be delivered on the 27th of November

3.3.6 Dependencies with other requirements

N/A

3.4 Give Feedback

3.4.1 Description

The user shall be able to enter their Feedback with regard to a location and whether or not the location in question is considered safe or not

3.4.2 Inputs

Location, Feedback, safety margin (1-10)

3.4.3 Outputs

void

3.4.4 Criticality

9/10

3.4.5 schedule

This is scheduled to be delivered on the 27th of November

3.4.6 Dependencies with other requirements

N/A

3.5 Submit a Picture

3.5.1 Description

The user shall be to submit pictures of a street as per our request

3.5.2 Inputs

location, picture

3.5.3 Outputs

void

3.5.4 Criticality

9/10

3.5.5 schedule

This is scheduled to be delivered on the 27th of November

3.5.6 Dependencies with other requirements

N/A

3.6 Submit Incident reports

3.6.1 Description

The user shall be able to submit a report, with a screenshot if possible, of any flaws or bugs they see in the application to be fixed later on

3.6.2 Inputs

Incident Picture, Incident Report

3.6.3 Outputs

void

3.6.4 Criticality

6/10

3.6.5 schedule

This is scheduled to be delivered on the 27th of November

3.6.6 Dependencies with other requirements

N/A

3.7 Submit Accident Reports

3.7.1 Description

The user shall be to submit a detailed accident report (with a picture if possible) of any accidents that is causing road blockage so that other people can be rerouted to safer and less congested roads and the accident will remain on the screen for 3 hours

3.7.2 Inputs

Accident Location, Accident Picture, Accident Report

3.7.3 Outputs

void

3.7.4 Criticality

5/10

3.7.5 schedule

This is scheduled to be delivered on the 27th of November

3.7.6 Dependencies with other requirements

N/A

3.8 Submit an "Unsafe Road" Report

3.8.1 Description

The user shall be to submit a report (with a picture if possible) of any roads that they deem to be unsafe so it can be taken into account

3.8.2 Inputs

Road Location, Road Picture, Road Report

3.8.3 Outputs

void

3.8.4 Criticality

7/10

3.8.5 schedule

This is scheduled to be delivered on the 27th of November

3.8.6 Dependencies with other requirements

Submit a Picture

3.9 Submit "New Bank" report

3.9.1 Description

The user shall be to submit a report (with a picture if possible) of any newly opened banks which helps us update the application's clustering and updating the areas classified as "safe"

3.9.2 Inputs

Bank Location, Bank Picture, Bank Report

3.9.3 Outputs

void

3.9.4 Criticality

4/10

3.9.5 schedule

This is scheduled to be delivered on the 27th of November

3.9.6 Dependencies with other requirements

N/A

3.10 Submit "New Police Station" report

3.10.1 Description

The user shall be to submit a report (with a picture if possible) of any new police stations which helps us update the application's clustering and updating the areas classified as "safe"

3.10.2 Inputs

Police Station Location, Police Station Picture, Police Station Report

3.10.3 Outputs

3.10.4 Criticality

4/10

3.10.5 schedule

This is scheduled to be delivered on the 27th of November

3.10.6 Dependencies with other requirements

N/A

3.11 Submit "New Shopping Districts" report

3.11.1 Description

The user shall be to submit a report (with a picture if possible) of any newly Malls or shopping districts which helps us update the application's clustering and updating the areas classified as "safe"

3.11.2 Inputs

Mall Location, Mall Picture, Mall Report

3.11.3 Outputs

void

3.11.4 Criticality

4/10

3.11.5 schedule

This is scheduled to be delivered on the 27th of November

3.11.6 Dependencies with other requirements

N/A

3.12 Set "Home" location

3.12.1 Description

The user shall be to set his home location so it can be readily used by the users whenever they like

3.12.2 Inputs

Location

3.12.3 Outputs

void

3.12.4 Criticality

3/10

3.12.5 schedule

This is scheduled to be delivered on the 27th of November

3.12.6 Dependencies with other requirements

N/A

3.13 Set "Work" location

3.13.1 Description

The user shall be to set his work location so it can be readily used by the users whenever they like

3.13.2 Inputs

Location

3.13.3 Outputs

void

3.13.4 Criticality

2/10

3.13.5 schedule

This is scheduled to be delivered on the 27th of November

3.13.6 Dependencies with other requirements

N/A

3.14 Detect dangerous curve radius

3.14.1 Description

The application shall be able to detect roads with a dangerous curve radius using the Douglas-Peucker algorithm

3.14.2 Inputs

Curve Picture

3.14.3 Outputs

Curve Radius

3.14.4 Criticality

10/10

3.14.5 schedule

This is scheduled to be delivered on the 27th of November

3.14.6 Dependencies with other requirements

N/A

3.15 process images from reports

3.15.1 Description

The application shall be able to process any images and apply several functions that will help in refining the pictures that are given with reports

3.15.2 Inputs

Images

3.15.3 Outputs

Processed images

3.15.4 Criticality

9/10

3.15.5 schedule

This is scheduled to be delivered on the 27th of November

3.15.6 Dependencies with other requirements

Submit a Picture, Submit Accident Reports, Submit an "Unsafe Road" Report, Submit "New Bank" report, Submit "New Police Station" report, Submit "New Shopping Districts" report

3.16 Detect dangerous junctions

3.16.1 Description

The application shall be able to detect dangerous junctions and feed it into the CNN to give more accurate weights to all paths given to the users

3.16.2 Inputs

Route, StepDisplay, Map, MarkerArray

3.16.3 Outputs

Route with marked junctions

3.16.4 Criticality

10/10

3.16.5 schedule

This is scheduled to be delivered on the 27th of November

3.16.6 Dependencies with other requirements

N/A

3.17 Set Weights for all the roads given to the user

3.17.1 Description

The application shall be able to give any road to the user marked with a weight which will indicate how safe that road is

3.17.2 Inputs

User Feedback, Accident Reports, Unsafe Roads, Bank locations, Police Station locations,Shopping District locations, dangerous curves, Processed images from reports, Dangerous junctions

3.17.3 Outputs

Safest Route

3.17.4 Criticality

10/10

3.17.5 schedule

This is scheduled to be delivered on the 27th of November

3.17.6 Dependencies with other requirements

Give Feedback, Submit a Picture, Submit Accident Reports, Submit an "Unsafe Road" Report, Submit "New Bank" report, Submit "New Police Station" report, Submit "New Shopping Districts" report, Detect dangerous curve radius, Detect dangerous junctions

3.18 Encrypt

3.18.1 Description

The application shall encrypt all users data which includes, but not limited to, trip data, passwords, Emails, survey feedback, reports ,and favorite location, in order to ensure the safety of our users' data.

3.18.2 Inputs

Data

3.18.3 Outputs

Encrypted Data

3.18.4 Criticality

10/10

3.18.5 schedule

This is scheduled to be delivered on the 27th of November

3.18.6 Dependencies with other requirements

N/A

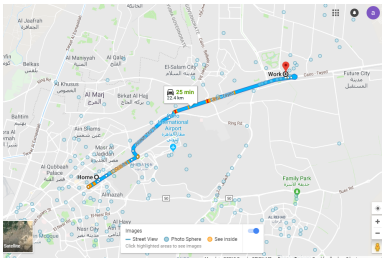
4 Interface Requirements

This section describes how the software interfaces with other software products or users for input or output.

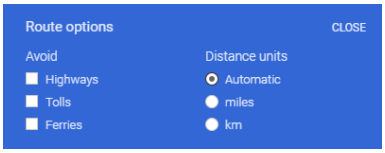
4.1 User Interfaces

Our system user interface is very usable and clear. You can login whether you are an admin or a guest; the system navigates you to dierent screens accordingly. System admins have some duties such as the manipulation of user feedback, Incident reports, accident reports and other things. While guest are able to send feedback, file Incident, accident reports, and "unsafe roads" report. Not to mention the basic functionalities such as setting a trip's start point and end point, setting their favorite places ,if they want to, and their home and work locations.

4.1.1 GUI



(a) Route Planning



(b) Route Options

4.1.2 API

initmap()

1. Description: Acts like the main; it is used to initialize the Google Map and encloses all the functions related to Google Maps
2. Arguments: Null
3. Return Values: Null
4. Example of invocation: function initMap()

Calculate and Display Route

1. Description: Display the route between the initial start and end selections.
2. Arguments:directionsDisplay, directionsService, markersArray, stepDisplay, map
3. Return Values: Markers for each junction
4. Example of invocation: function calculateAndDisplayRoute(directionsDisplay, directionsService, markerArray, stepDisplay, map)

showSteps()

1. Description: For each step, place a marker, and add the text to the marker's infowindow.Also attach the marker to an array so we can keep track of it and remove it when calculating new routes.
2. Arguments: directionResult, markerArray, stepDisplay, map
3. Return Values: Gets the path and the direction
4. Example of invocation: function showSteps(directionResult, markerArray, stepDisplay, map)

4.2 Communications Interfaces

Some services at Google provide external access to data and functionality through APIs that utilize the Google Data (GData) Protocol. The GData Protocol is a REST-inspired technology for reading, writing, and modifying information on the web. It supports two primary modes of access, AtomPub and JSON.

5 Performance Requirements

The system is expected and required to consume as little memory as possible and and it is also required to be quick enough to perform reliably, in comparison to its predecessor with minor changes due to the change of the application's objective. The application will use somewhere between 200 MB to 750 MB of space and anywhere between 1 GB to 3 GB of RAM.

6 Design Constraints

Here is a list of all design constraints

6.1 connection limitation

Our connection limitation presents itself in the user's need for a good internet connection in order to use our application

6.2 Client limitation

The main design constraint is the that the application cant be accessed through the web

7 Other non-functional attributes

Specifies any other particular non-functional attributes required by the system.

7.1 Security

Our application shall provide security by encrypting all user data that includes trip data passwords, Emails, survey feedback, reports ,and favorite location, in order to protect our users' data.

7.2 Compatibility

Since our application is designed on "Android Studio", it is compatible with any mobile application which allows it to work on any smartphone as long as that smartphone runs an android OS and that meets the space, memory, and minimum SDK requirements.

7.3 Reliability

Our application is reliable because it has high error tolerance given that there are delays when errors come up and no unexpected behaviour and multiple alerts placed throughout the code to warn the user should any error occur.

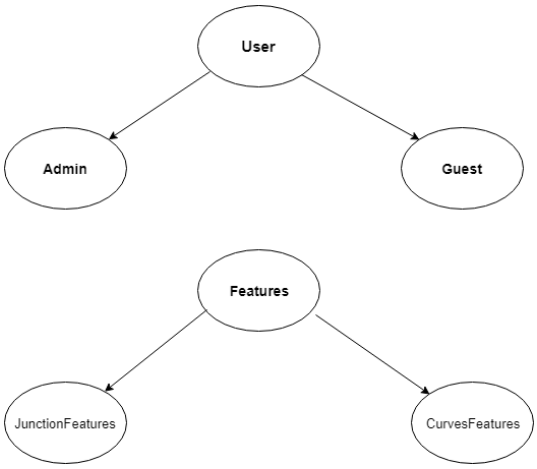
8 Preliminary Object-Oriented Domain Analysis

This section presents a list of the fundamental objects that must be modeled within the system to satisfy its requirements. The purpose is to provide an alternative, "structural" view on the requirements stated above and how they might be satisfied in the system.

8.1 Inheritance Relationships

8.2 Class descriptions

8.2.1 Class name: User
Type: Concrete
superclasses: N/A
Subclasses: Admin, guest.
Purpose: class to encapsulate different user-types with common attributes



(a) Inheritance relationships

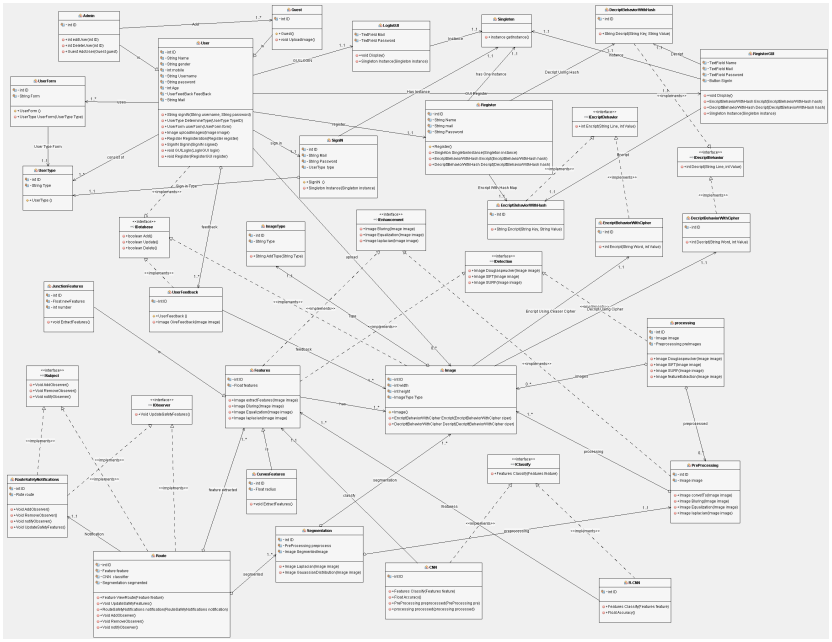


Figure 4: Class Diagram

Collaborations: this class aggregate UserFeedback, inherited by class Admin,Guest, associated by class UserType, register, sinIN, UserForm, Image.
Attributes: ID, Name, gender, mobile, Username, password, age, feedback, mail
Operations: signIN(username,password),determineType(typeID),userForm(form), uploadImage(image), Registration(register),SignIN(signIN).

8.2.2 Class name: Admin
Type: Concrete
superclasses: User
Subclasses: none.
Purpose: to represent the user admin
Collaborations: this class extends class user, associated by class guest.
Attributes: ID
Operations: AddUser(guest), DeleteUser(ID),EditUser(ID).

8.2.3 Class name: Guest
Type: Concrete
superclasses: User
Subclasses: none.
Purpose: to represent the user Guest
Collaborations: this class extends class user, associate class admin.
Attributes: ID
Operations: Guest(), UploadImage().

8.2.4 Class name: Register
Type: Concrete
superclasses: None
Subclasses: None.
Purpose: To Allow the users to register an account
Collaborations: this class associate class user,associated by class EncriptBehaviorWithHash,DecriptBehaviorWithHash,Singleton.
Attributes: ID, Name, Mail, Password
Operations: Register(), SingletonInstance(instance), Encript(hash),Decript(hash)

8.2.5 Class name: SignIN
Type: Concrete
superclasses: None
Subclasses:None
Purpose: To allow the user to sign in
Collaborations: This class associate class user,aggregate class UserType,associated by class Singleton.
Attributes: ID, Mail, Password,Type.
Operations: SignIN(),SingletonInstance(instance).

8.2.6 Class name: UserForm
Type: concrete
superclasses: None
Subclasses:. None
Purpose: To handle the forms for the user according to the sign In
Collaborations: this class associated by class user,associate class UserType.
Attributes: ID, Form
Operations: UserForm(),UserForm(type).

8.2.7 Class name: UserType
Type: concrete
superclasses: None
Subclasses:. None
Purpose: To allow scalability for adding new user Type,
Collaborations: This class associate the class user,UserForm,signIN.
Attributes: ID, Type
Operations: UserType()

8.2.8 Class name: UserFeedback
Type: concrete
superclasses: None
Subclasses: None
Purpose: TO handle the user feedbacks
Collaborations: This class aggregate class user, associated by class image.
Attributes: ID
Operations: userfeedback(), givefeedback(image)

8.2.9 Class name:image
Type: concrete
superclasses: None
Subclasses: None
Purpose: to contain the image
Collaborations: This class is aggregated by class segmentation, preprossessing,processing associate class feature, user, userfeedback, aggregate class ImageType, associated by class EncriptBehaviorWithcipher,DecriptBehaviorWithcipher.
Attributes: ID, width, height,type
Operations: image(), Encript(cipher),Decript(cipher)

8.2.10 Class name: segmentation
Type: concrete
superclasses: None
Subclasses: None
Purpose: To remark the part of the images that interest to us
Collaborations: This class aggregate class image, preprocessing,aggregated by class Route.
Attributes: ID, Prepossess, segmentedImage
Operations: laplacian(image), gauassianDistribution(image)

8.2.11 Class name:preprocessing
Type: concrete
superclasses: None
Subclasses: None
Purpose: To do some processing on image that was taken by user feedback to prepare before segmentation

Collaborations: this class aggregates class image, aggregated by segmentation, implement IEnhancement, aggregated by class processing.
Attributes: ID, image
Operations: convertTo(image),bluring(image),equalization(image),laplacian(image)

8.2.11 Class name: features
Type: concrete
superclasses: None
Subclasses: None
Purpose: To contain the basic features of the image
Collaborations: This class is inherited by class junctionFeature, curveFeature, associated by class image,CNN,R-CNN, implements IEnhancement,IDetection, aggregated by class Route.
Attributes: ID, feature
Operations: extractFeatures(image),blurring(image),equalization(image),laplacian(image)

8.2.12 Class name:junctionFeatures
Type: concrete
superclasses: features
Subclasses: None
Purpose: to extract the junction features from the image
Collaborations: this class inherits class features
Attributes: ID, newFeature, number
Operations: Extractfeature()

8.2.13 Class name: curvesFeatures
Type: concrete
superclasses: Features
Subclasses: None
Purpose: To extract the curves features from the image
Collaborations: this class inherits class features
Attributes: ID, radius
Operations: ExtractFeatures()

8.2.14 Class name: CNN
Type: concrete
superclasses: None
Subclasses: None
Purpose: To allow classification using CNN classifier
Collaborations: This class associated by class features, implements Class IClassify.
Attributes: ID
Operations: classify(features),accuracy(), preprocessed(pre), processed(processed)

8.2.15 Class name: IEnhancement
Type: interface

superclasses: None
Subclasses: None
Purpose: To allow enhancement of images with different algorithms
Collaborations: This class is implemented by class features, preprocessing
Attributes: None
Operations: Blurring(image),equalization(image), Laplacian(image)

8.2.16 Class name: IDetection
Type: interface
superclasses: None
Subclasses: None
Purpose: To allow feature detection of images with different algorithms
Collaborations: This class is implemented by class features, processing
Attributes: None
Operations: Douglaspeucker(image),SIFT(image),SURF(image)

8.2.17 Class name: processing
Type: concrete
superclasses: None
Subclasses: None
Purpose: To do some processing on image before classify
Collaborations: This class is implements class IDetection, aggregates class Image, preprocessing.
Attributes: ID, Image, preimages
Operations: Douglaspeucker(image),SIFT(image),SURF(image), featureExtraction(image)

8.2.18 Class name: Route
Type: concrete
superclasses: None
Subclasses: None
Purpose: To represent the nal result received from the classication.
Collaborations: This class is aggregates class Feature, Segmentation, Associated by class RouteSafetyNotification, Implements class IObserver, ISubject
Attributes: ID, feature, classifier, segmented
Operations: ViewRoute(feature), UpdateSafetyFeatures(), notification(notification),AddObserver(),RemoveOb

8.2.19 Class name:RouteSafetyNotification
Type:concrete
superclasses:None
Subclasses: None
Purpose: To define one-to many dependency between notification and the route.
Collaborations: This class associate class Route, Implements class ISubject, IObserver.
Attributes:ID, Route.
Operations: AddObserver(),RemoveObserver(),notifyObserver(),Void Update-

SafetyFeatures().

8.2.20 Class name:ISubject
Type: interface
superclasses: None
Subclasses: None
Purpose: to allow adding of many subjects easy
Collaborations: This class is implemented by RouteSafetyNotification, Route.
Attributes: None
Operations: AddObserver(), RemoveObserver(), notifyObserver().

8.2.21 Class name:IObserver
Type: interface
superclasses: None
Subclasses: None
Purpose: To notify Observer when subject state changes
Collaborations: This class is implemented by RouteSafetyNotification, Route.
Attributes: None
Operations: UpdateSafetyFeatures()

8.2.22 Class name: ImageType
Type: Concrete
superclasses: None
Subclasses: None
Purpose: To allow scalability for adding new user Type
Collaborations: This class is aggregated by class Image.
Attributes: ID, Type.
Operations: AddType(Type)

8.2.23 Class name:Singleton
Type: Concrete
superclasses: None
Subclasses: None
Purpose: A design pattern used to restrict instantiating of class to one object class with single instance
Collaborations: This class associate class Register,SignIN.
Attributes: None
Operations: getInstance().

8.2.24 Class name: R-CNN
Type: Concrete
superclasses: None
Subclasses: None
Purpose: To allow classification using CNN classifier
Collaborations: This Class implements class IClassify, associated by class Feature.

Attributes: ID
Operations: Classify(Features feature),Accuracy().

8.2.25 Class name: EncriptBehaviorWithHash
Type: concrete
superclasses: None
Subclasses: None
Purpose: To encrypt saved data with hash map encryption
Collaborations: this class associate class Register, implements class IEncriptBehavior.
Attributes:ID
Operations: Encript(Key,Value).

8.2.26 Class name: DecriptBehaviorWithHash
Type: concrete
superclasses: None
Subclasses: None
Purpose: To Decript data with hash map decryption
Collaborations:this class implements class IDecriptBehavior, associate class Register.
Attributes:ID
Operations:Decript(Key,Value)

8.2.27 Class name: EncriptBehaviorWithCipher
Type: concrete
superclasses: None
Subclasses: None
Purpose: To encrypt image data with Caesar cipher encryption
Collaborations: this class implements class IEncriptBehavior, associate class Image.
Attributes:ID
Operations: Encript(Word,Value)

8.2.28 Class name: DecriptBehaviorWithCipher
Type: concrete
superclasses: None
Subclasses: None
Purpose: To decrepit image data with Caesar cipher decryption
Collaborations: this class implements class IDecriptBehavior, associate class Image.
Attributes: ID
Operations: Decript(Word,Value)

8.2.29 Class name: IClassify
Type: interface
superclasses: None

Subclasses: None
Purpose: to add multiple classifiers
Collaborations: this class is implemented by class CNN, R-CNN.
Attributes: None
Operations: Classify(feature)

8.2.30 Class name: IEncriptBehavior
Type: interface
superclasses: None
Subclasses: None
Purpose: is to implement multiple encryption algorithms
Collaborations: this class is implemented by EncriptBehaviorWithHash, EncriptBehaviorWithCipher.
Attributes: None
Operations: Encript(Line,Value)

8.2.31 Class name: IDecriptBehavior
Type: interface
superclasses: None
Subclasses: None
Purpose: is to implement multiple decryption algorithms
Collaborations: this class is implemented by class DecriptBehaviorWithHash, DecriptBehaviorWithCipher.
Attributes: None
Operations: Decript(Line, Value)

8.2.32 Class name: RegisterGUI
Type: GUI
superclasses: None
Subclasses: None
Purpose: is the view that Allow the users to register an account
Collaborations: his class associate class user,associated by class EncriptBehaviorWithHash,DecriptBehaviorWithHash,Singleton
Attributes: Name,mail,password,register.
Operations: Display(),SingletonInstance(instance), Encript(hash),Decript(hash)

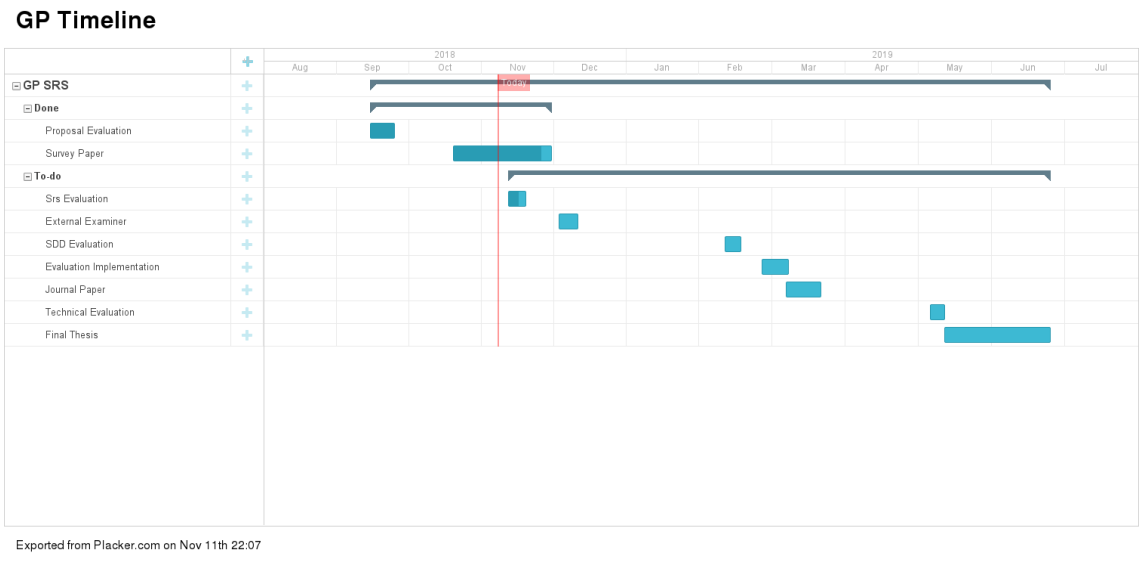
8.2.33 Class name: LogInGUI
Type: GUI
superclasses: None
Subclasses: None
Purpose: is the view that Allow the users to log in his account
Collaborations: his class associate class user,aggregate class UserType,associated by class Singleton.
Attributes: Mail,password
Operations: Display(),SingletonInstance(instance)

9 Operational Scenarios

Once the application is opened the user will be required to sign up by submitting some basic information such as name, age, an email for a username and a password. And after there he will see the map and top, enter he will find an empty bar awaiting a destination to be received and once a destination is received the application will give the user the safest routes from the set location to the destination point. The user can also enter several destinations instead of just one and the application will work the same. The user will be allowed to enter his favorite destinations, set home and work locations, and also filing several reports ranging from accident reports to bug reports to aid us in making the system choose safer roads. If an admin,on the other hand, is the one to log in then they will see different screens that essentially allows them to work with the user data that is voluntarily submitted by the user which includes the, previously mentioned, reports that are filed by each user which the admins will use in order to enhance the applications quality and general functionality and to add, if possible, more features in the future based on user requests.

10 Preliminary Schedule Adjusted

This section provides an initial version of the project plan, including the major tasks to be accomplished, and their initial start/stop dates.



We are currently in the SRS phase which is due on the 27th of november.

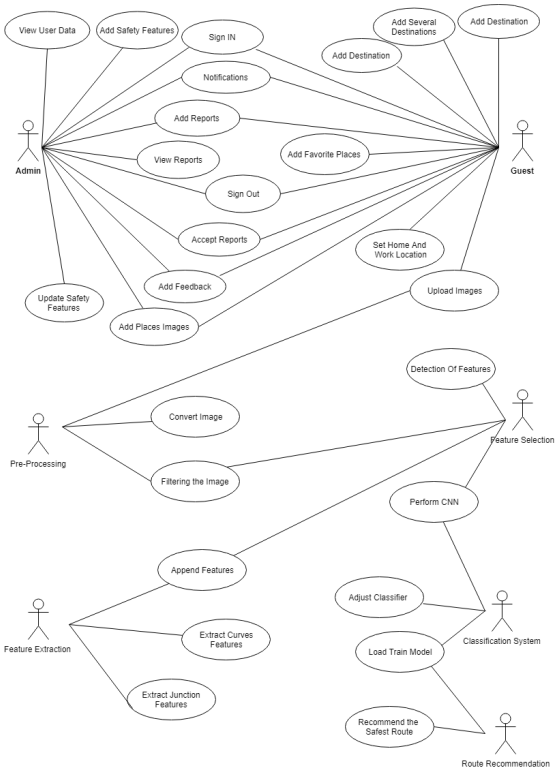


Figure 5: Use Case Diagram

11 Preliminary Budget Adjusted

Any smartphone with Android version 5.0 and up. On the user end,the minimum cost of any sufficient internet package required to operate the program which varies from 50 EGP to 250 EGP. on the back end, we also require a Non-sql database such as Firebase for real-time updates which costs around $1,650. The price of an appropriate mid-range sever is between 4000$ & 4,495$

12 Definitions, Acronyms, Abbreviations

Word	Meaning
IDE	integrated development environment
CNN	Convoluted Neural Network
API	Application Programmable Interface
SRS	Software Requirement Specification
MB	Megabyte
Gb	Gegabyte
RAM	Random-Access Memory
SDK	Software Development kit
OS	Operating System

13 References

[1] https://www.egyptindependent.com/car-accidents-decline-24-6-2017-capmas/