

# Amelio-rater: Detection of Driving Abnormal Behavior and Road Anomalies for Automated Ratings

Mariam ElAshram, Noha AlMasry, and Passant El-Dorry  
Supervised by Dr. Ayman Ezzat and Eng. Huda ElTouny

January 16, 2017

## 1 Introduction

### 1.1 Purpose of this document

The purpose of this Software Requirements Specification document is to outline the requirements for Amelio-rater: detection of abnormal driving behaviors and road anomalies for automated ratings. Amelio-rater will be built on Node.js, Express.js, MongoDB, Firebase, Android Studio , and Heroku. It will consist of two mobile applications accessible with any Android Phone and a web application independent and accessible with any standard compliant browser. After meeting with Vodafone, they have specified some requirements. Those requirements included the detection of driver ID with external Government which will be verified if found, and if speeding will be considered as an ADB which is already available in the predefined data set. They have also asked about the data transmission, that what if there is no internet connection, and this could be done by storing the data in a local database and synchronized for further future analysis.

### 1.2 Scope of this document

This document targets the end users like businesses that would integrate their systems with the Amelio-rater system. It will also be beneficial and helpful for designers and developers that may work on the Amelio-rater system in the future.

### 1.3 Overview

Real time Monitoring and observing of abnormal driving behavior is the bedrock to enhance driving evaluation systems. The rating systems of the drivers' should mainly aim to provide consumers with trustworthy feedback; which also assists the business owner gain high accuracy overview of the performance quality of the hired professional drivers. This document proposes "Amelio-rator" which is a system that performs detection and identification of driving abnormal behavior using sensors fusion found in smart phones. Amelio-rater be focusing further improvement in the quality of the current rating systems found in most of transportation companies. Amelio-rater will consist of a mobile application that will handle the real time data that will be collected from sensors (accelerometer, gyroscope, and GPS), this data will be analyzed to create automatically generated ratings that would be used by the business owner to keep track of the drivers' performances and location. The business owner will be provided an interface through which he could monitor each driver's and trip's generated data.

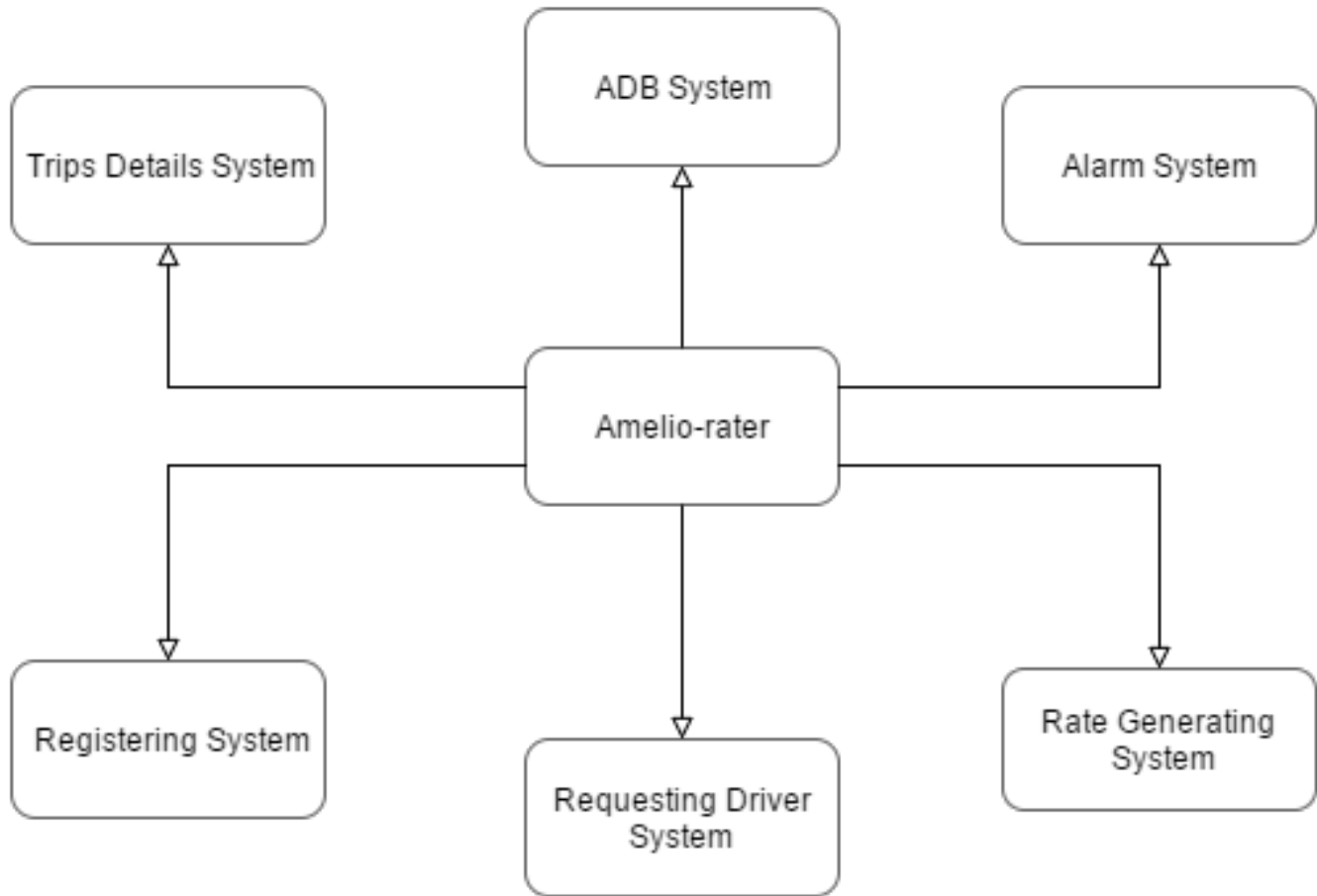


Figure 1 : Block Diagram.

1. Subsystem 1: ADB System - This subsystem detects and classifies the ADB.
2. Subsystem 2: Alarm System - This subsystem handles the notifications sent to the driver.
3. Subsystem 3: Rate Generating System - This subsystem handles calculation of the automatically generated rates.
4. Subsystem 4: Requesting Driver System - This subsystem handles sending and receiving of requests between consumers and drivers.
5. Subsystem 5: Registering System - This subsystem handles the addition of new users of different types to the system.
6. Subsystem 6: Trips Details System - This subsystem handles collecting information about each trip individually.

Interactions between the subsystems:

1. The Requesting Driver Subsystem triggers the ADB subsystem as detection and classification will only start when a trip is being conducted.
2. The Alarming subsystem is triggered by the ADB subsystem as when a number of ADBs are detected, the driver is notified.

#### 1.4 Business Context

As the abnormal driving behavior detection is a huge research nowadays, it is extending to be adopted in related businesses such as private taxi companies. Detecting the driving abnormal behaviors could benefit the drawbacks of the current commercial systems, allowing consumers to gain more trust in the systems and businesses gain more control over their systems. Companies

such as Careem, Uber and other private taxis may use the Amelio-rater and integrate it with their system which currently runs based on the ratings manually entered by the customers. Amelio-rater will generate the automatic rating alongside the ratings entered by the consumer, so there will be no interference between both rating systems. It could work independently without the integration of another business' system such as Careem for example, and could also be integrated with the business' system with some modifications done.

## 2 General Description

### 2.1 Product Functions

1. Business Owner can monitor the drivers' driving behavior.
2. Business Owner can view trips' duration.
3. System rates individual drivers based on their average driving behavior during all their trips.
4. System will detect if there is a road bump.
5. Business Owner can view individual trips' ratings.
6. Business Owner can view individual drivers' ratings.
7. System will have different users.
8. System rates each individual trip.

### 2.2 Similar System Information

In the past years, the rating systems mainly relied on manual ratings recorded by individuals regardless of whether the driver deserves the rate whether it is good or bad. D3[2] did develop a system to aid in reducing accidents by detecting a driving abnormal behavior taking place by the driver regardless of other works[6][7] based on pre-deployed infrastructure. This project found that already existing works[4][3] only detects an abnormal behavior and due to the sensors' sensitivity and different car types the threshold is affected therefore identification of the abnormal behavior could not be provided. Our system strives to accentuate the accuracy rates reached by D3 system after proposing the classification techniques alongside the detection. The main sensor that can not be let go of under any circumstances in such systems is the accelerometer. As mentioned by Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore in[5] the importance of the accelerometer in recognizing an ongoing activity is crucial in processes such as data collection for example. They used an application with an interface that allowed them to get users' permissions to get some of their details and sensors' data along with the frequency of the data collection occurrence. Such proven and accurate experiments assists us in our system as the accelerometer is one of the sensors on which most of our readings and data depend. MyDrive[1] is a system that works only on the speed of the moving car; however, our proposed system will be working on other driving behaviors alongside the acceleration of the car. They use an analytic web based UI portal to analyze the data, an smart phone sensors to collect the required data.

### 2.3 User Characteristics

1. Business Monitor:
  - Must have basic knowledge in using computers and web browsers.

2. Driver:

- Must have basic knowledge in using Android mobile devices.

3. Consumer:

- Must have basic knowledge in using Android mobile devices.

## 2.4 User Problem Statement

Detection and improvement of classification accuracy of driving abnormal behaviors and road conditions to automatically generate ratings in real-time.

## 2.5 User Objectives

Using Amelio-rater, the consumer can view rating of the driver not only based on how other consumers have rated him/her, but also rating based on his driving behavior. Moreover, business monitors will be given the privilege to monitor their drivers throughout any trip in the real time; this includes showing him/her when a driver is doing an abnormal behavior.

## 2.6 General Constraints

1. GPS accuracy: updates after every 25 meters.
2. Mobile application applicable for android mobile devices only.

# 3 Functional Requirements

## 3.1 User Class 1 - The Admin

### 3.1.1 ID: FR1

-Title: Adding a Driver.

-DESC: The admin adds a driver to the system.

-Input: Name , Email , Password , Telephone , Gender , National ID , User type (Driver).

-Action: Checks if all fields are filled and if so the data is entered in a new record in the database accordingly.

-Output: Confirmation message or error message if something went wrong upon validating the fields.

-Pre-condition: None.

-Post-condition: Database is updated with a new driver's account.

-DEP: None.

### 3.1.2 ID: FR2

-Title: Editing a Driver's Account.

-DESC: The Admin edits a driver's information.

-Input: Any of the following: Name , Email , Password , Telephone , Gender , National ID.

-Action: The information changed is taken and sent to the corresponding attribute in the driver's record in the database to be updated.

-Output: Confirmation message or error message if something went wrong upon validating the fields.

-Pre-condition : Desired driver is already registered in the database.

-Post-condition : Desired driver's record in the database is updated with the new information in the database.

-DEP: FR1.

### **3.1.3 ID: FR3**

- Title: Deleting a Driver's Account.
- DESC: The Admin deletes a driver's account from the system.
- Input: Driver's name.
- Action: Checks for the selected driver's record in the database to be deleted.
- Output: Confirmation message or error message if something went wrong upon removing the driver from the system.
- Pre-condition : Desired driver is already registered in the database.
- Post-condition : Desired driver's record is removed from the database.
- DEP: FR1.

### **3.1.4 ID: FR4**

- Title: Listing all Drivers.
- DESC: The Admin lists all the drivers found in the system.
- Input: User Type.
- Action: Retrieves information about the drivers registered in the system from the database.
- Output: All drivers registered in the system and their information are previewed.
- Pre-condition : At least one driver is registered in the database.
- Post-condition : None.
- DEP: FR1.

### **3.1.5 ID: FR5**

- Title: Searching for a Driver.
- DESC: The Admin searches for a desired driver.
- Input: Name.
- Action: Retrieves information about the driver, whose name has been entered by the admin, from the database.
- Output: The desired driver's information is previewed.
- Pre-condition : Desired driver is already registered in the database.
- Post-condition : None.
- DEP: FR1.

### **3.1.6 ID: FR6**

- Title: Adding a Business Monitor.
- DESC: The admin adds a business monitor record to the database.
- Input: Name , Email , Password , Telephone , Gender , User type (Business Monitor).
- Action: Checks if all fields are filled and if so the data is entered in a new record in the database accordingly.
- Output: Confirmation message or error message if something went wrong upon validating the fields.
- Pre-condition: None.
- Post-condition: Database is updated with a new business monitor's account.
- DEP: None.

### **3.1.7 ID: FR7**

- Title: Editing a Business Monitor's Account.
- DESC: The Admin edits a business monitor's account to the database.
- Input: Any of the following: Name , Email , Password , Telephone , Gender.
- Action: The information changed is taken and sent to the corresponding attribute in the business monitor's record in the database to be updated.
- Output: Confirmation message or error message if something went wrong upon validating the fields.
- Pre-condition : Desired business owner is already registered in the database.

-Post-condition : Desired business owner's record in the database is updated with the new information in the database.  
-DEP: FR6.

### **3.1.8 ID: FR8**

-Title: Deleting a Business Monitor's Account.  
-DESC: The Admin deletes a business monitor's account from the database  
-Input: Business Monitor's name.  
-Action: Checks for the selected business monitor's record in the database to be deleted.  
-Output: Confirmation message or error message if something went wrong upon removing the business monitor from the system.  
-Pre-condition : Desired business monitor is already registered in the database.  
-Post-condition : Desired business monitor's record is removed from the database.  
-DEP: FR6.

### **3.1.9 ID: FR9**

-Title: Listing all Business Monitors.  
-DESC: The Admin lists all the business monitors found in the system.  
-Input: User Type.  
-Action: Retrieves information about the business monitors registered in the system from the database.  
-Output: All business monitors registered in the system and their information are previewed.  
-Pre-condition : At least one business monitor is registered in the database.  
-Post-condition : None.  
-DEP: FR6.

### **3.1.10 ID: FR10**

-Title: Searching for a Business Monitor.  
-DESC: The Admin searches for a desired business monitor.  
-Input: Name.  
-Action: Retrieves information about the business monitor, whose name has been entered by the admin, from the database.  
-Output: The desired business monitor's information is previewed.  
-Pre-condition : Desired business monitor is already registered in the database.  
-Post-condition : None.  
-DEP: FR6.

### **3.1.11 ID: FR11**

-Title: Deleting a Consumer's Account.  
-DESC: The Admin deletes a consumer's account from the database  
-Input: Consumer's name.  
-Action: Checks for the selected consumer's record in the database to be deleted.  
-Output: Confirmation message or error message if something went wrong upon removing the customer from the system.  
-Pre-condition : Desired consumer is already registered in the database.  
-Post-condition : Desired consumer's record is removed from the database.  
-DEP: FR18.

### **3.1.12 ID: FR12**

-Title: Listing all Consumers.  
-DESC: The Admin lists all the consumers found in the system.  
-Input: User Type.  
-Action: Retrieves information about the consumers registered in the system from the database.  
-Output: All consumers registered in the system and their information are previewed.  
-Pre-condition : At least one consumer is registered in the database.

-Post-condition : None.  
-DEP: FR18.

### **3.1.13 ID: FR13**

-Title: Searching for a Consumer.  
-DESC: The Admin searches for a desired consumer.  
-Input: Name.  
-Action: Retrieves information about the consumer, whose name has been entered by the admin, from the database.  
-Output: The desired consumer's information is previewed.  
-Pre-condition : Desired consumer is already registered in the database.  
-Post-condition : None.  
-DEP: FR18.

## **3.2 User Class 2 - Business Owner**

### **3.2.1 ID: FR14**

-Title: Log in.  
-DESC: The business owner logs into his account.  
-Input: Email and password.  
-Action: Checks if all fields are filled and compares data entered to that in the database records.  
-Output: The homepage is previewed(Monitoring Page) and Login successful message or error message upon validating the fields.  
-Pre-condition : Business Owner is already registered in the database.  
-Post-condition : Redirected to the monitoring page.  
-DEP: FR6.

### **3.2.2 ID: FR15**

-Title: Searching for a Drivers.  
-DESC: The Business Owner searches for a desired driver.  
-Input: Name of Driver.  
-Action: Retrieve the driver(s) with the name corresponding to the name entered by the business monitor from the database.  
-Output: The desired driver's information is previewed.  
-Pre-condition : Desired driver is already registered in the database.  
-Post-condition : None.  
-DEP: FR1, FR14.

### **3.2.3 ID: FR16**

-Title: Real Time Monitoring of Abnormal Driver Behaviors and Locations.  
-DESC: The Business Monitor monitors the cars in current trip viewing their locations and Abnormal Driving Behaviors on the maps.  
-Input: None.  
-Action: The cars' locations are retrieved from the database and updated on the map and if an abnormal driving behavior occurs, the car's color changes accordingly.  
-Output: Cars on the map.  
-Pre-condition : At least one driver registered in the database.  
-Post-condition : None.  
-DEP: FR1, FR14.

### **3.2.4 ID: FR17**

-Title: Viewing Detailed Ratings of Driver.  
-DESC: The Business Monitor views the details of the driver's ratings, including details about the driver's trips.  
-Input: None.

- Action: Retrieve the ratings data from the database that correspond to the selected driver. Data retrieved will be for both personal ratings and system auto-generated ratings.
- Output: Detailed ratings of the driver's trips are previewed.
- Pre-condition : At least one driver registered in the database.
- Post-condition : None.
- DEP: FR1, FR15, FR14.

### **3.3 User Class 3 - Consumer**

#### **3.3.1 ID: FR18**

- Title: Registering.
- DESC: The consumer registers with his/her information to create an account.
- Input: Name , Email , Password , Telephone , Gender.
- Action: Checks if all fields are filled and if so the data is entered in a new record in the database accordingly.
- Output: Confirmation Message and asks consumer to log in or error message upon validating the fields.
- Pre-condition: None.
- Post-condition: Database is updated with a new consumer's account.
- DEP: None.

#### **3.3.2 ID: FR19**

- Title: Log in.
- DESC: The consumer logs into his account.
- Input: Email and password.
- Action: Checks if all fields are filled and if so compares data entered to that in the database records.
- Output: The homepage is previewed and Login successful message or error message upon validating the fields.
- Pre-condition : Consumer is already registered in the database.
- Post-condition : Redirected to the homepage.
- DEP: FR18.

#### **3.3.3 ID: FR20**

- Title: Request Driver.
- DESC: The consumer requests a driver sending his/her current location which is sent to drivers nearby.
- Input: current location.
- Action: The consumer chooses to request a driver and allows his/her current location to be sent.
- Output: Message that request has been sent or error message that it hasn't.
- Pre-condition : At least one driver registered in the system.
- Post-condition : Redirected to the homepage.
- DEP: FR1, FR18, F19.

#### **3.3.4 ID: FR21**

- Title: View cars' locations.
- DESC: The consumer views the drivers' current locations on a map.
- Input: None.
- Action: The locations of each car is retrieved from the database to be previewed accordingly.
- Output: The cars previewed on the map according to their location.
- Pre-condition : At least one driver registered in the system.
- Post-condition : None.
- DEP: FR1, FR18, F19.



### **3.3.5 ID: FR22**

- Title: View Driver's rating.
- DESC: The consumer views the rating of the driver that has accepted his request.
- Input: None.
- Action: The rating of the driver that has accepted the request is retrieved from his/her record in the database.
- Output: The value of the driver's rating is previewed.
- Pre-condition : The driver is registered in the system.
- Post-condition : None.
- DEP: FR1, FR18, F19, FR20.

### **3.3.6 ID: FR23**

- Title: Rate the Driver.
- DESC: The consumer views the rating of the driver that has accepted his request.
- Input: Rate value.
- Action: The rating entered is used to calculate the new overall rating for the driver. Note that this is the personal rating.
- Output: A message confirming that the rating has been done or error message upon validating fields.
- Pre-condition : The driver is registered in the system.
- Post-condition : Driver's personal rating is updated in the driver's record in the database.
- DEP: FR1, FR18, F19, FR20.

## **3.4 User Class 4 - Driver**

### **3.4.1 ID: FR24**

- Title: Log in.
- DESC: The driver logs into his account.
- Input: Email and password.
- Action: Checks if all fields are filled and if so compares data entered to that in the database records.
- Output: The homepage is previewed and Login successful message or error message upon validating fields.
- Pre-condition : Driver is already registered in the database.
- Post-condition : Redirected to the homepage.
- DEP: FR1.

### **3.4.2 ID: FR25**

- Title: Accept Request.
- DESC: The driver accepts a request sent by a consumer.
- Input: None.
- Action: Sends confirmation of request acceptance to the consumer, and retrieves the consumer's location details.
- Output: Consumer's location details.
- Pre-condition : At least one consumer has sent a request from a location nearby.
- Post-condition : None.
- DEP: FR1, FR20, FR24.

### **3.4.3 ID: FR26**

- Title: View Trips' Ratings.
- DESC: The driver views the ratings of trips he/she has performed.
- Input: None.
- Action: Retrieves the details of the trips that were performed by the logged in driver.
- Output: Detailed information about the trips is previewed.
- Pre-condition : Driver has to have performed at least one trip.

-Post-condition : None.  
-DEP: FR1, FR24.

#### **3.4.4 ID: FR27**

-Title: View his/her Rating.  
-DESC: The driver views his/her overall ratings.  
-Input: None.  
-Action: Retrieves the driver's ratings. Data retrieved will be for both personal ratings and system auto-generated ratings.  
-Output: Detailed information about his/her overall rating is previewed.  
-Pre-condition : None.  
-Post-condition : None.  
-DEP: FR1, FR24.

#### **3.4.5 ID: FR28**

-Title: Detection of ADB.  
-DESC: The ADB model will detect if an ADB has been made by the driver during the trip.  
-Input: accelerometer, gyroscope, and location readings.  
-Action: Compares the sensors' readings to the normal readings in a predefined data set, and detects whether the readings are normal or abnormal.  
-Output: None.  
-Pre-condition : A trip is currently being performed by the driver.  
-Post-condition : None.  
-DEP: FR1, FR24, FR25.

#### **3.4.6 ID: FR29**

-Title: Classification of ADB.  
-DESC: The ADB model will classify a detected ADB into its type.  
-Input: accelerometer, gyroscope, and location readings of the detected ADB.  
-Action: Compares the sensors' ADB readings to the ADB readings in a predefined data set, and classifies the readings to its type (weaving, speeding, etc.).  
-Output: None.  
-Pre-condition : A trip is currently being performed by the driver.  
-Post-condition : The database record with the readings is updated with the type of ADB.  
-DEP: FR1, FR24, FR25, FR28.

#### **3.4.7 ID: FR30**

-Title: Generating Automatic Rating of Driver Based on Driving.  
-DESC: The ADB model will automatically generate the new rating of the driver based on his/her driving behavior in all trips performed including the one he/she just finished.  
-Input: Previous rating if there is and the statistics from the trip that has just ended such as number of ADB that were made by the driver.  
-Action: Calculates the new rating of the driver.  
-Output: None.  
-Pre-condition : A trip has been just ended by the driver.  
-Post-condition : The driver's profile is updated by the new auto-generated rating by the ADB model.  
-DEP: FR1, FR24, FR25, FR28.

#### **3.4.8 ID: FR31**

-Title: Notifying the driver when exceeding a number of ADBs.  
-DESC: The alarm model will send a notification to the driver to notify him/her that he/she has exceeded a number of ADBs.  
-Input: Count of ADBs that were made in the current trip.  
-Action: Checks if the ADBs count has exceeded the predefined maximum limit and sends a

- notification to the driver accordingly.
- Output: A beeping sound on the driver's mobile device.
- Pre-condition : A trip is currently being performed by the driver.
- Post-condition : None.
- DEP: FR1, FR24, FR25, FR28.

## 4 Interface Requirements

### 4.1 User Interfaces

#### 4.1.1 GUI

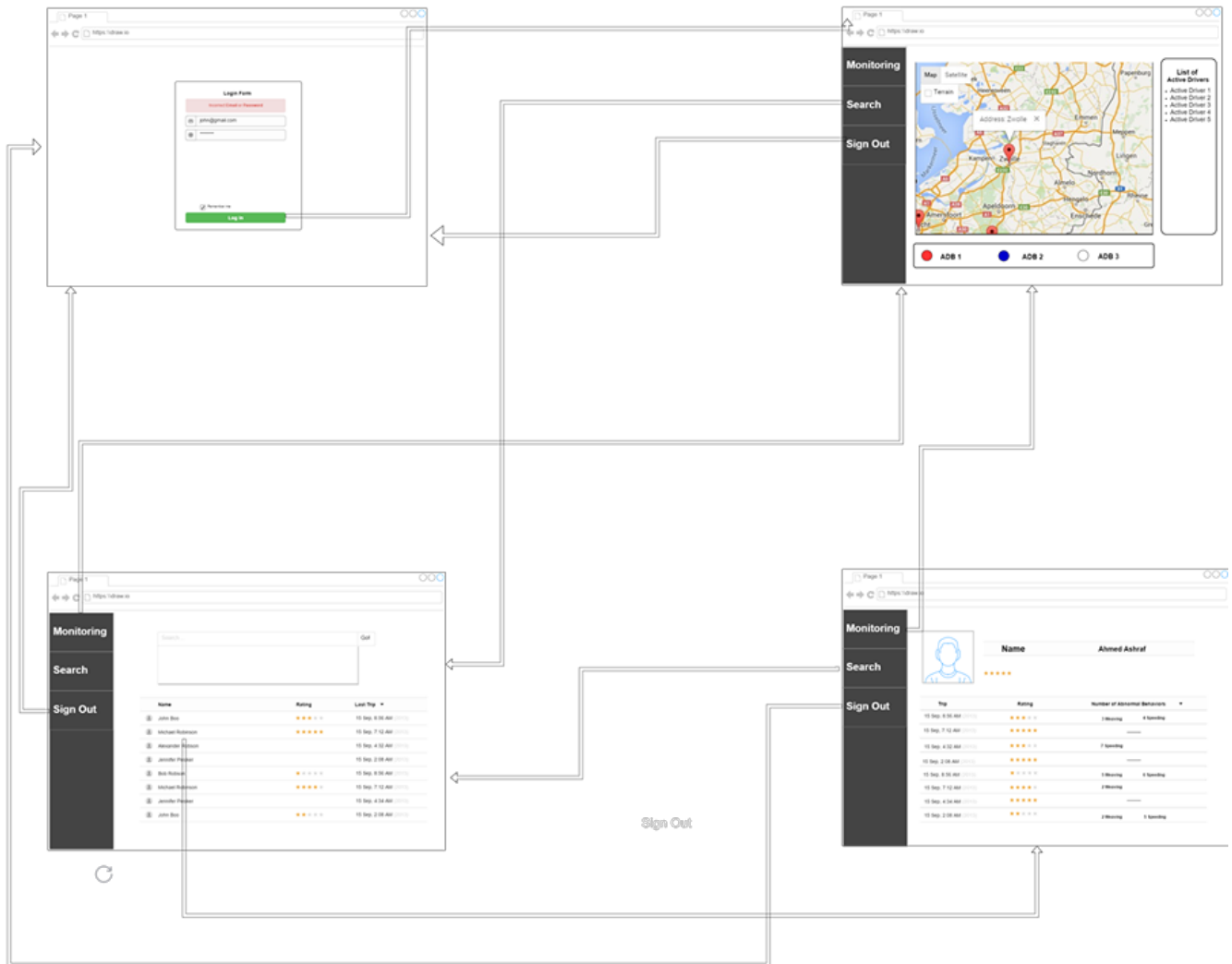


Figure 2 : Website Wireframe for the business monitor portal.



Figure 3 : Consumer’s Mobile Application Wireframe.

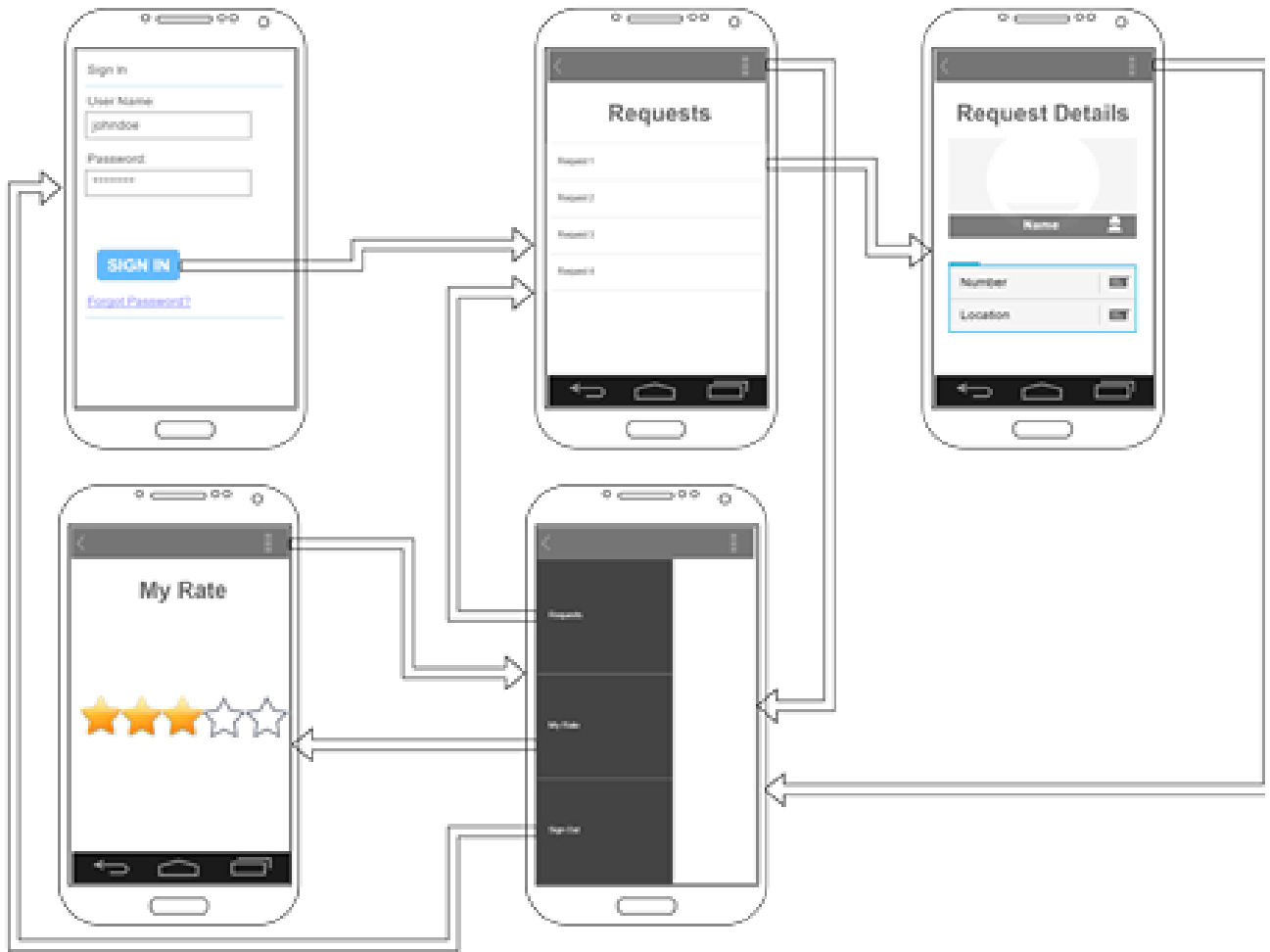


Figure 4 : Driver's Mobile Application Wireframe.

#### 4.1.2 API

1. Google Maps API
2. Mongoose
3. Passport

## 5 Performance Requirements

Data would be sent to server every 0.5 seconds over GSM network.

## 6 Other non-functional attributes

### 6.1 Performance and Speed

-DESC: The Amelio-rater must be interactive and the delays involved must be reduced. So in every action-response of the Amelio-rater, there will be no immediate delays. Data sent such as the driver's location needs to be sent with very high speed as it is updated every 0.5 seconds. Detection and Classification of an ADB must have no delays in order for the business monitor to be seeing the correct type of behavior(whether normal or abnormal) of the driver when he/she is using the real time monitoring service provided by the Amelio-rater.

-FR Dependent on this: FR16,FR21,FR28,FR29.

## 6.2 Security

-DESC: Information transmission should be securely transmitted to server without any changes in information; data transmitted should also be secured in a way that it could not be accessed by any interference. Location of the driver and the consumer should be secured and not given local access. Also personal information about the users such as mobile numbers and passwords for instance must be protected.

-FR Dependent on this: FR1, FR6, FR14, FR18, FR19, FR20, FR24, FR25.

## 6.3 Reliability

-DESC: As the Amelio-rater provides the right tools, it must be made sure that the system is reliable in its operations. This would be mainly focusing on the detection and classification of the ADB, as the detected sensors' readings should be accurate and error free. When the detected behavior is being classified, it is very important for the classifier to identify the behavior type correctly without errors in its precision. Also the user should be able to trust that the Amelio-rater's fail rate is almost 0 percent.

-FR Dependent on this: FR28, FR29.

## 6.4 Scalability

-DESC: As the Amelio-rater is a system that could be integrated with other systems as mentioned above in the document, it should be easy to maintain to minimize the amount of changes that would be done to the code in that case. Other systems may have a rating system that would have to be kept separate from the rating system of the Amelio-rater, hence keeping the code scalable in such an area is important.

-FR Dependent on this: FR23, FR30.

## 7 Preliminary Object-Oriented Domain Analysis

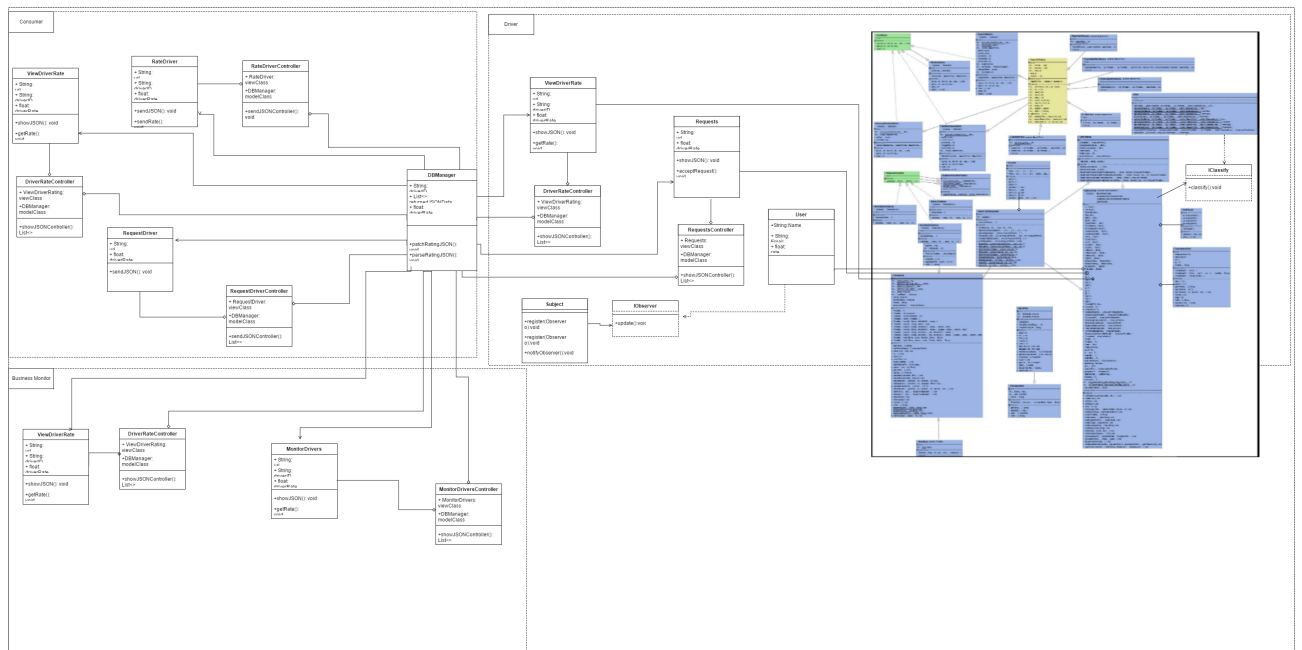


Figure 5 : Primitive Class Diagram.

### 7.1 IClassify

-Purpose: An interface used for the Strategy Design Pattern, as different kinds of Classifiers may be used by the Amelio-rater.

## 7.2 IObserver

-Purpose: An interface used for the Observer Design Pattern, to send notifications to the users(in this case the drivers) subscribed to this service.

## 8 Operational Scenarios



Figure 6 : Use Case Diagram for System.



## 8.1 Scenarios

Here will be illustrated some of the scenarios that are shown in the system's use case diagram.

### 8.1.1 Scenario 1: Requesting a Driver.

The users included in this scenario are the driver and the consumer. This scenario requires a consumer to send a request, which is received by all the drivers that are located nearby the consumer's location. A driver then accepts the request. Once a driver has accepted a request the trip starts and the request is no longer pending.

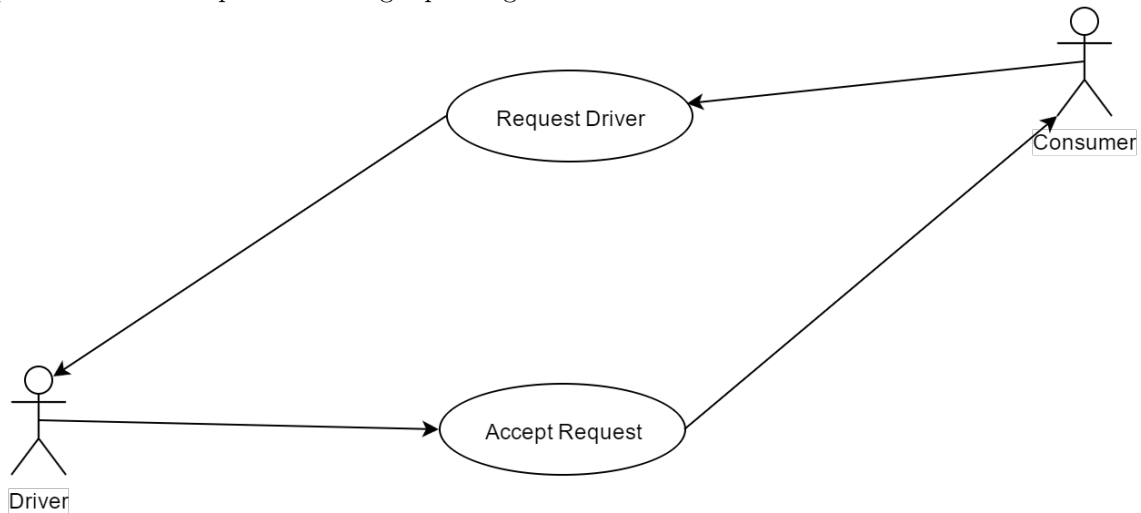


Figure 7 : Requesting a Driver Use Case.

### 8.1.2 Scenario 2: User Manipulation.

The user included in this scenario is the admin. The admin is in control of manipulating the users in the system (business monitors, consumers, and drivers) and this includes:

1. Adding a user by type (business monitor or driver).
2. Editing information of users of different types (business monitor or driver).
3. Removing users of different types (business monitor, consumer, or driver).
4. Listing all users of specific type per selection (business monitor, consumer, or driver).
5. Searching for a specific user in a specific type of users (business monitor, consumer, or driver).

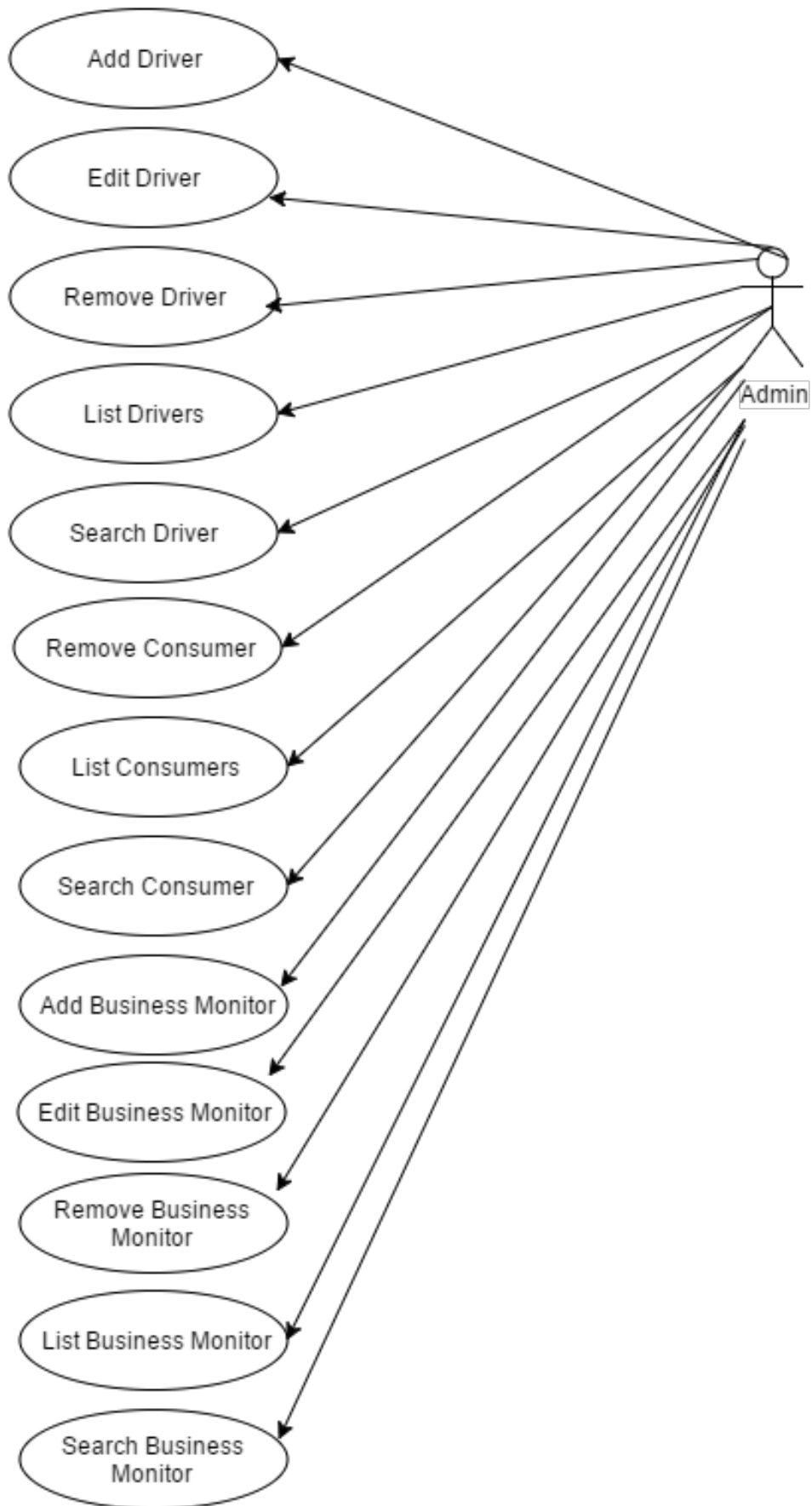


Figure 8 : User Manipulation Use Case.

### 8.1.3 Scenario 3: Detecting and Classifying ADBs to Generate the Automated Ratings.

These actions are done by the ADB Model system which is a subsystem in the Amelio-rater System. When a driver is undergoing a trip, the model detects if an abnormal driving behavior was made by the driver by comparing the sensors' readings to the normal readings stored in a predefined data set. Those readings that have been detected, are then compared against another predefined data set that consists of readings of different types of ADBs to classify the detected reading as a type of ADB. Once the trip is over, the count of ADBs is calculated and a rating is generated which is accumulated with previous rating value to generate and store the new rating value based on the driving behavior of the driver.

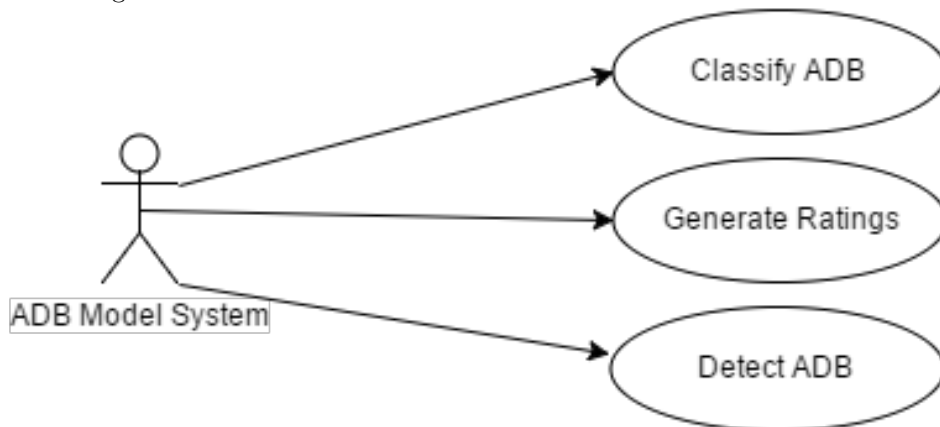


Figure 9 : Detecting and Classifying ADBs to Generate the Automated Ratings Use Case.

## 9 Preliminary Schedule Adjusted

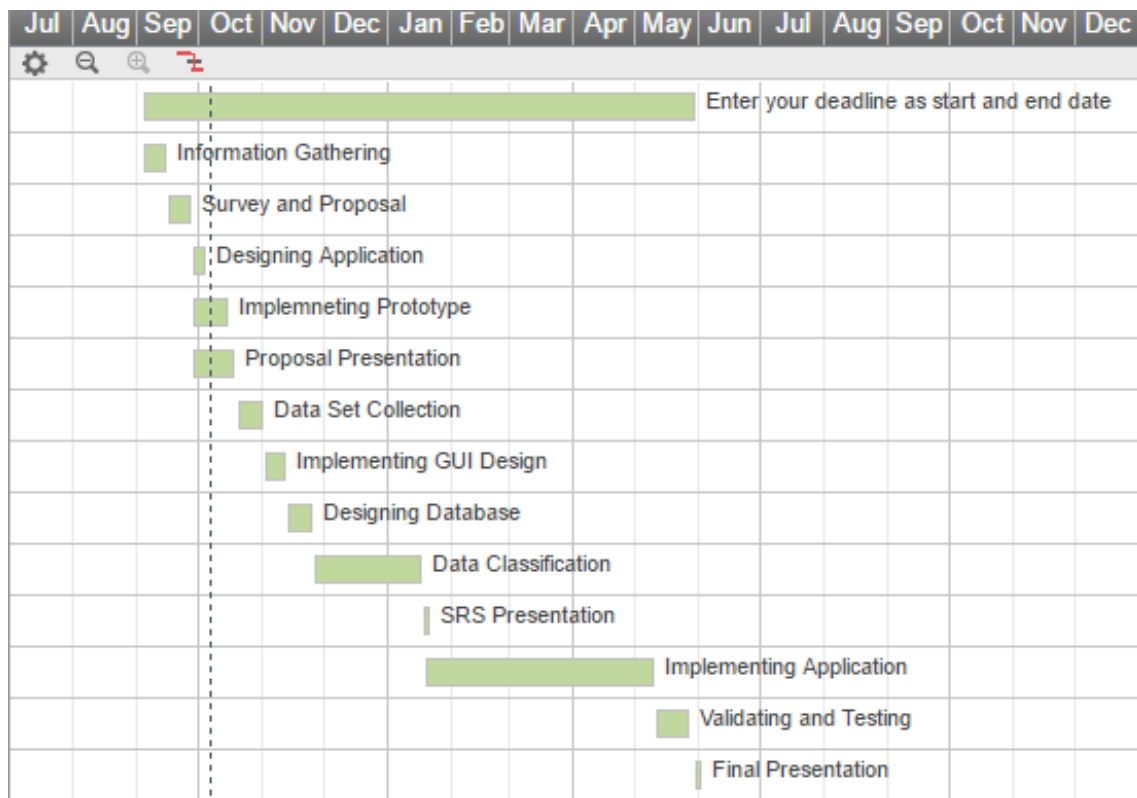


Figure 10 : Gantt Chart.

## 10 Preliminary Budget Adjusted

Item	Quantity	Cost
Huawei Ascend X (U9000)	1	5607 EGP
Accelerometer Sensors	1	1000 EGP

Figure 11 : Budget.

Amazon Web Services	\$0.0314 per GB
mLab Cloud Storage	\$180

Figure 12 : Budget for web server and Cloud Storage.

## 11 Appendices

### 11.1 Definitions, Acronyms, Abbreviations

1. Node.js - An open-source, cross-platform JavaScript runtime environment.
2. Express.js - Web framework for Node.js.
3. MongoDB - A noSQL open-source cross-platform document-oriented database program.
4. Firebase - A mobile and web application platform.
5. Android Studio - An integrated development environment (IDE) for Android platform development.
6. Heroku - A cloud Platform-as-a-Service.
7. ADB - Abnormal Driving behavior.
8. GSM - Global System for Mobile.
9. FR - Functional Requirement.

## References

- [1] Tanushree Banerjee, Arijit Chowdhury, and Tapas Chakravarty. “MyDrive: Drive Behavior Analytics Method And Platform”. In: *Proceedings of the 3rd International on Workshop on Physical Analytics*. ACM. 2016, pp. 7–12.
- [2] Zhongyang Chen et al. “D 3: Abnormal driving behaviors detection and identification using smartphone sensors”. In: *Sensing, Communication, and Networking (SECON), 2015 12th Annual IEEE International Conference on*. IEEE. 2015, pp. 524–532.
- [3] Jiangpeng Dai et al. “Mobile phone based drunk driving detection”. In: *2010 4th International Conference on Pervasive Computing Technologies for Healthcare*. IEEE. 2010, pp. 1–8.
- [4] Mohamed Fazeen et al. “Safe driving using mobile phones”. In: *IEEE Transactions on Intelligent Transportation Systems* 13.3 (2012), pp. 1462–1468.
- [5] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. “Activity recognition using cell phone accelerometers”. In: *ACM SigKDD Explorations Newsletter* 12.2 (2011), pp. 74–82.
- [6] Saif Al-Sultan, Ali H Al-Bayatti, and Hussein Zedan. “Context-aware driver behavior detection system in intelligent transportation systems”. In: *IEEE transactions on vehicular technology* 62.9 (2013), pp. 4264–4275.
- [7] Mervyn VM Yeo et al. “Can SVM be used for automatic EEG detection of drowsiness during car driving?” In: *Safety Science* 47.1 (2009), pp. 115–124.