

Software Design Document for 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

April 26, 2017

1 Introduction

1.1 Purpose

The purpose of the software design document is to provide a full description of the Amelio-rater and how it is expected to be built based on the use cases. It also defines the functional requirements and their impact on the system architecture and design. It also shows the functionality of each subsystem and component and how components interact with each other.

1.2 Scope

Amelio-rater is designed to become a smart rating system to drivers. Systems that use manual rating systems may deal with unfairness in the recorded rates. Also the system offers a real-time monitoring to the business owner, along side with a notification system to alert the driver when abnormal behaviors occur repeatedly. The main objective detection and improvement of classification accuracy of driving abnormal behaviors and road conditions to automatically generate ratings in real-time. Amelio-rater creates a smart classification system that can differentiate between different abnormal behaviors as sudden changing of lanes, weaving, and sudden single weave.

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.

1.4 Definitions and Acronyms

1. MongoDB - A noSQL open-source cross-platform document-oriented database program.
2. Google Firebase - A mobile and web application platform.
3. Android Studio - An integrated development environment (IDE) for Android platform development.
4. Google Maps API - and API that allows using the google maps functionalities elsewhere.

5. ADB - Abnormal Driving behavior.
6. FR - Functional Requirement.
7. KNN - K Nearest Neighbour.
8. SVM - Support Vector Machine.
9. DTW - Dynamic Time Warping.
10. ANN - Artificial Neural Network.
11. RTM - Real Time Monitoring, monitoring the users(drivers) in the real time on a map.
12. BM - Business Monitor

2 System Overview

Amelio-rater is a pattern recognition solution built to complement current driving rating systems. Amelio-rater is designed for private transportation companies and can be used during trips being conducted by drivers to detect and classify the drivers' abnormal driving behaviors. It aims to introduce a sensor-fusion solution into the modern rating systems for drivers; eliminating malicious rating data that is found in the current rating systems. This leads to better trustworthy rating systems, and motivates drivers to improve their driving behavior. Amelio-rater allows the business to be able to have a person(business monitor) monitoring the drivers' in a real time, informing him/her when a driver does an abnormal behavior; specifying the behavior's type and location. Also, the driver is notified when (s)he exceeds a number of abnormal driving behaviors.

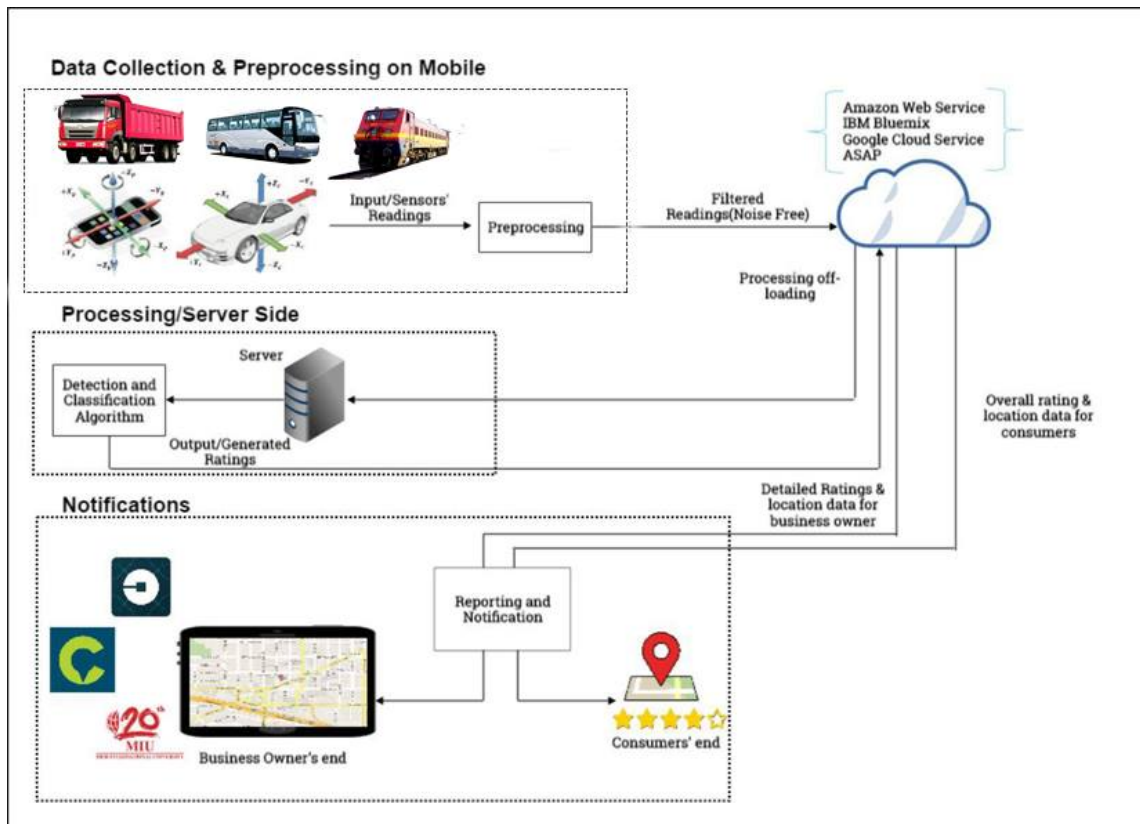


Figure 1: System Overview.

The proposed solution provides fast detection and classification of an abnormal driving behavior that occurred by a driver according to his location. The data obtained is sent over a network so that it is stored in the database for further retrieval of data for the real time monitoring and rate generation.

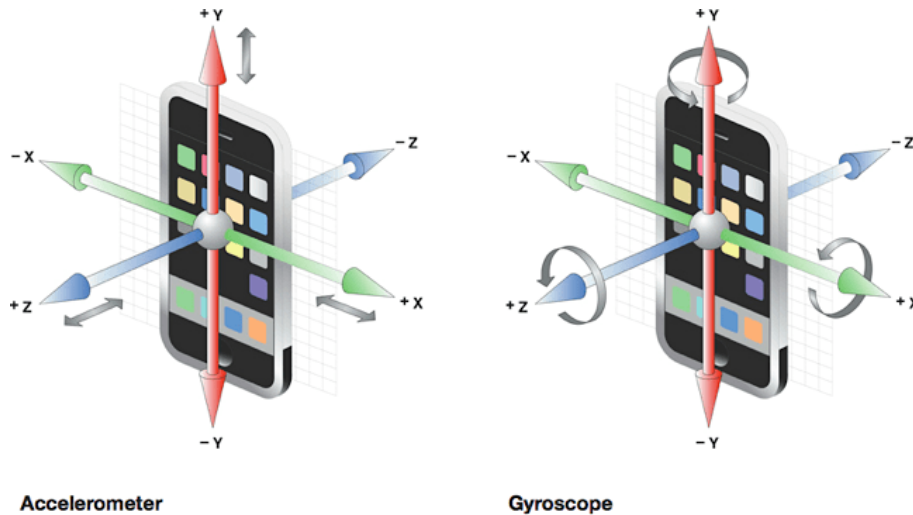


Figure 2: Sensors' Different Readings.

Amelio-rater provides the readings that are illustrated in the image above, those are the main readings on which the computations take place. The gyroscope readings indicate the rotation in the different directions(x,y,x) whereas the accelerometer indicates the change in position in the different directions(x,y,z).

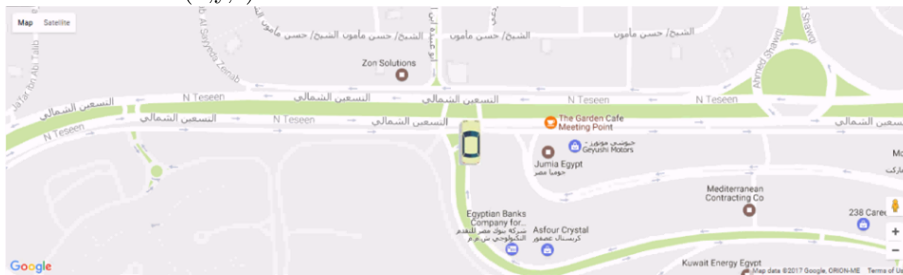


Figure 3: Data Visualisation on the Map.

Illustrated above is how the Amelio-rater visualises the data. This is the map that appears to the business monitor, where the data represented is in the real time. The car that appears show the car that is currently in a trip. The car changes colour according to the behaviour that is being done. For example, if the car is moving in a normal behaviour, the car is shown in a silver colour. On the other hand, if the car is weaving for instance, it is shown in a red colour, and so on.

3 System Architecture

3.1 Architectural Design

Amelio-rater is designed in an MVC architecture.

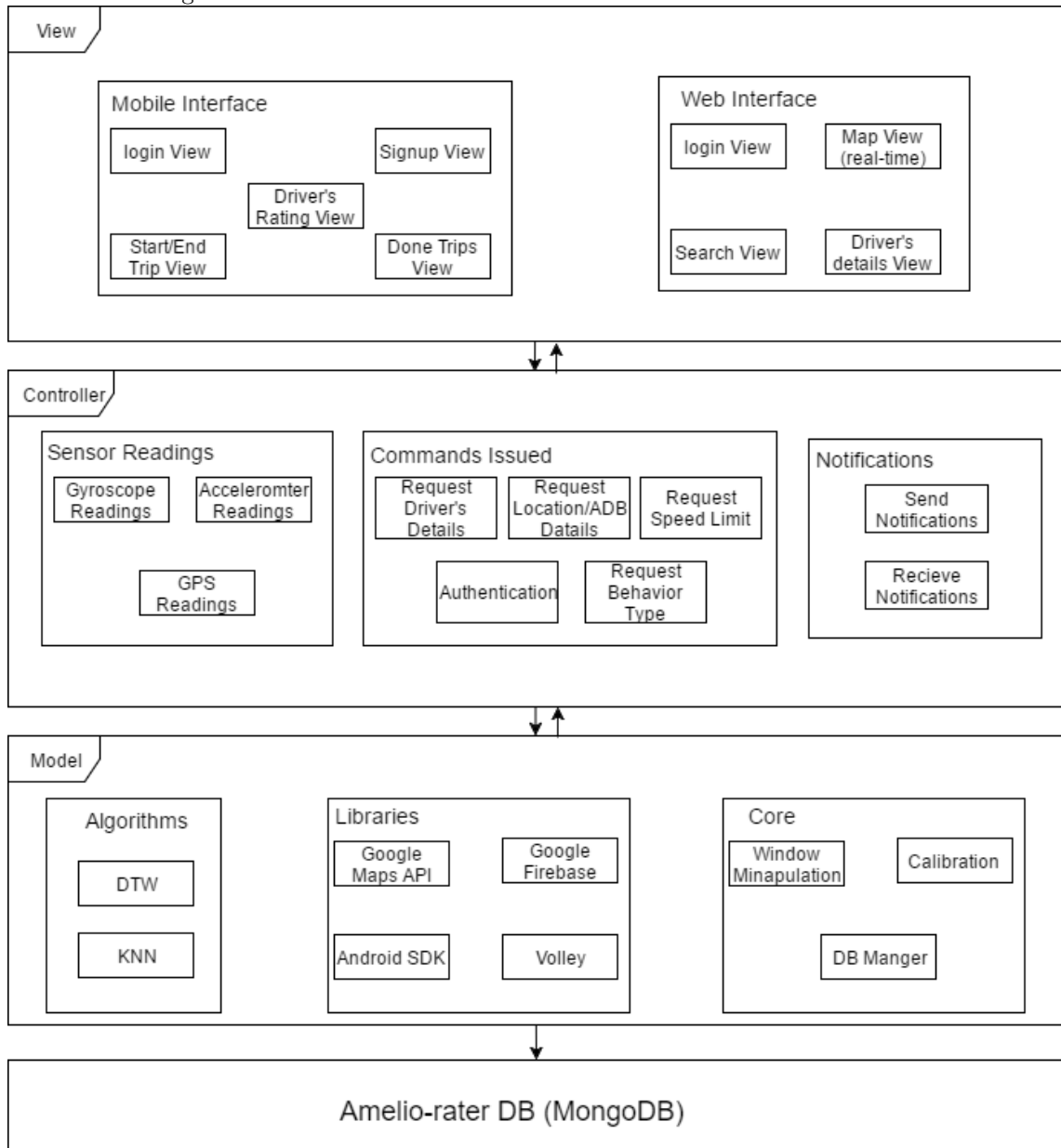


Figure 4: MVC Architecture Diagram

3.1.1 View

This represents the user interface which is divided into two interfaces.

- **Business Interface:** consists of a menu to help the user navigate through the functionalities provided by the web interface. Each page represents a certain functionality, such as searching, logging in, viewing the real time monitoring on maps.
- **Drivers' interface:** consists of a screen that has buttons to start and end a trip .

Through the interfaces, users can do different actions and and all the information they need that is shown after being processed.

3.1.2 Controller

This binds the View with the Model. Takes input from the View and sends it to the database and retrieves data from the Model and sends it to the view to be previewed there. The handling of all the data coming from the input hardware devices including gyroscope readings coming from the gyroscope sensor, accelerometer reading coming from the accelerometer sensor and location readings. In the Amelio-rater, it is responsible for distribution of the raw data between the View and the Model to be processed and analysed.

3.1.3 Model

- Core:
 - Model Manipulator
 - * Calibration: the readings pass through an equation to remove noise found in the readings such as gravity to improve interpretation and computation on data.
 - * Window Manipulation: the data is segmented to the size of the window that we set, the data in that window is the data being used for the manipulation at that time. When moving on to new data for the computation, about 25 percent of the end of the previous data is kept at the start of the window and the rest of the window is filled with new data to start computation. This is done to not lose any behaviour that is done in case of the behaviour starting at the end of the new window and part of it is in the beginning of the new window.
 - Gesture Recording: data from the sensors will be recorded for further use in computations.
 - Gesture Classifier
 - * DTW: short for Dynamic Time Warping. It is used to compare the timeseries of the sensors' readings within the window (mentioned above, which is the testing dataset) against the training datasets[11]. It gets the Eclidean distances between them, and chooses the best fit behaviour according to the training dataset and the threshold set based on analysis obtained from averages of samples of each training dataset. We chose DTW after researching and finding a comparative study [1] showed that testing samples were correctly classified with accuracy 99.7 percent; which is fairly high but the gestures type was human gestures whereas we will be using driving behaviours and road conditions.
 - * KNN: short for K Nearest Neighbour. It is a classification algorithm where a set of examples is collected and each one has a known class. This set is used as a training dataset for the algorithm so when a new data is fed to it a comparison takes place between the training dataset and the new data. K-nearest neighbor is a classification algorithm where a set of examples is collected and each one has a known class. NN(Nearest Neighbour) is found based on the value of **K**. This is an effective and simple approach in the case of large training datasets.
 - Libraries:
 - * Google Maps API: this is the API that is being used to provide the business monitor with a map display on the web interface and use its functionalities.
 - * Android SDK: the Amelio-rater has an android based mobile application which is the drivers' interface. Since it is android platform, then it requires the Android SDK to be able to perform the functionalities on the android devices and use the sensors.
 - * Volley: a library that assists in creating a network in the android application. In the Ameliorater, it is used for interacting with the database.
 - * Firebase: a mobile and web application platform with tools and infrastructure designed to help developers build high quality applications. In the Amelio-rater, it will be used for sending notifications to the driver when exceeding a number of abnormal behaviors.

3.2 Decomposition Description

3.2.1 Class Diagram

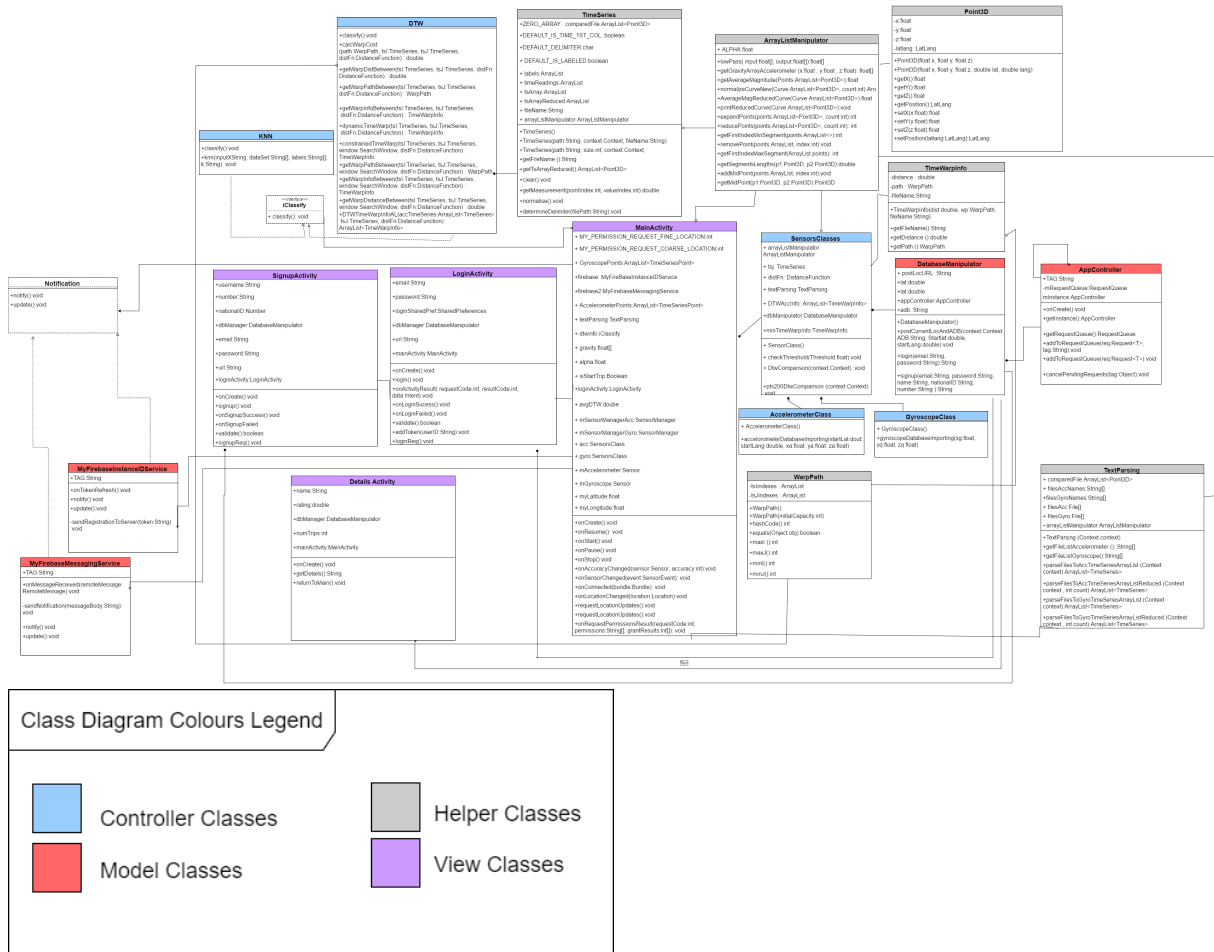


Figure 5: This figure is the class diagram of the driver’s android mobile application and legend for colours illustrated.

This class diagram implements the MVC Design Pattern which is used to separate the application’s view from it’s model by interacting via the controller.

- Model represents an object or JAVA POJO carrying data. It can also have logic to update controller if its data changes.
- View represents the visualization of the data that model contains.
- Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.

Snaps of the driver’s class diagram:

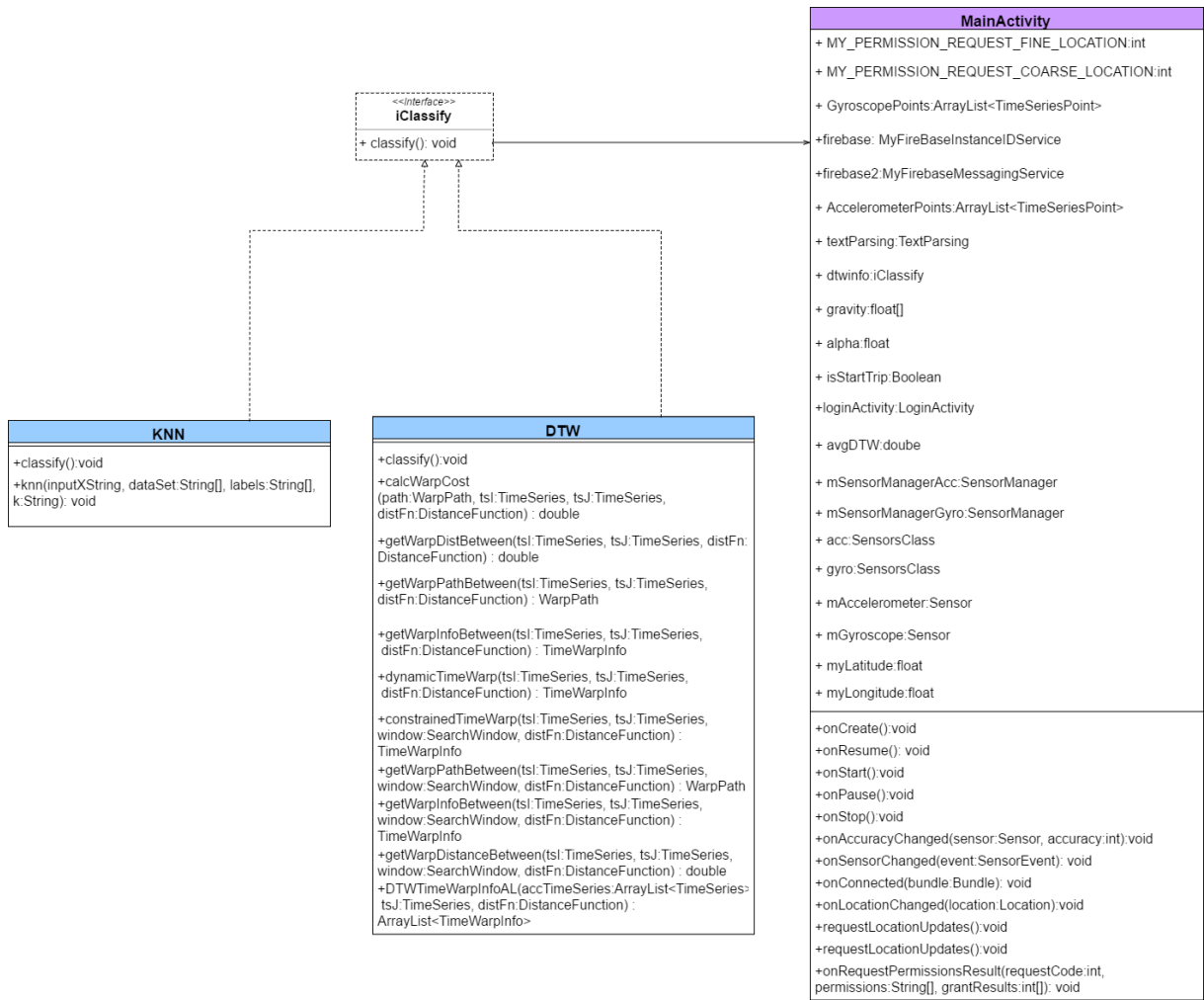


Figure 6: Strategy Design Pattern for the classifiers. In Strategy pattern, a class behavior or its algorithm can be changed at run time. It is used in our case as we may be using different algorithms for the classification of abnormal driving behaviours process.

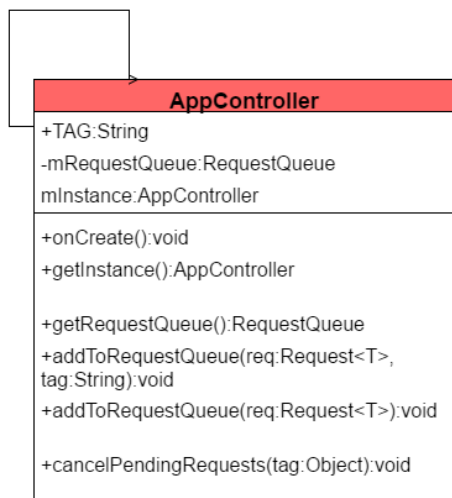


Figure 7: Singleton Design Pattern. In Singleton Design Pattern, the class extends itself. In our case, the AppController class is a singleton class that extends itself and it is responsible for the database connection and the volley requests.

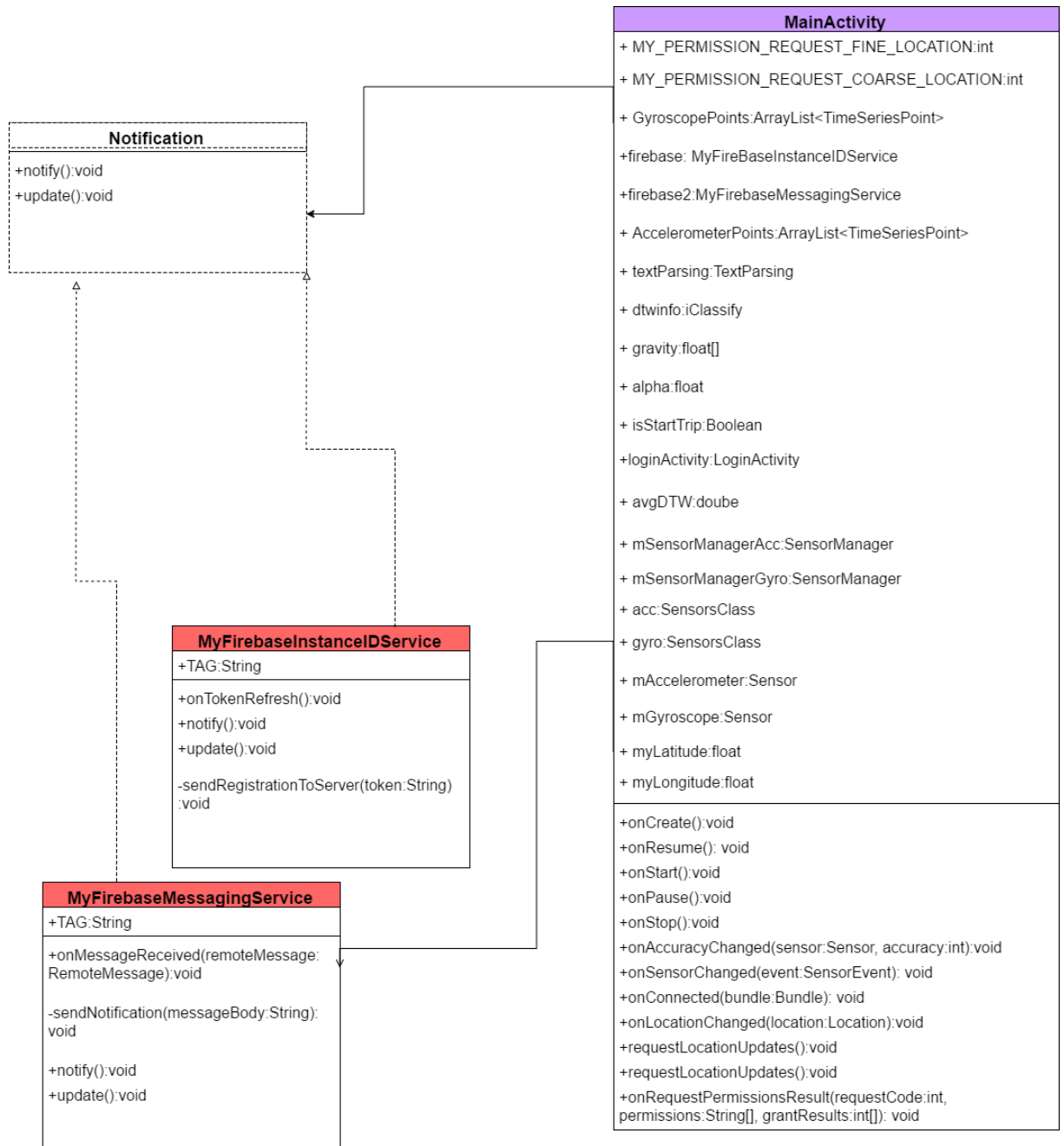


Figure 8: Observer Design Pattern. The Observer Design Pattern is used in the Amelio-rater for the notifications feature. Firebase services is being used in this case for handling the notifications for the Drivers' mobile application.

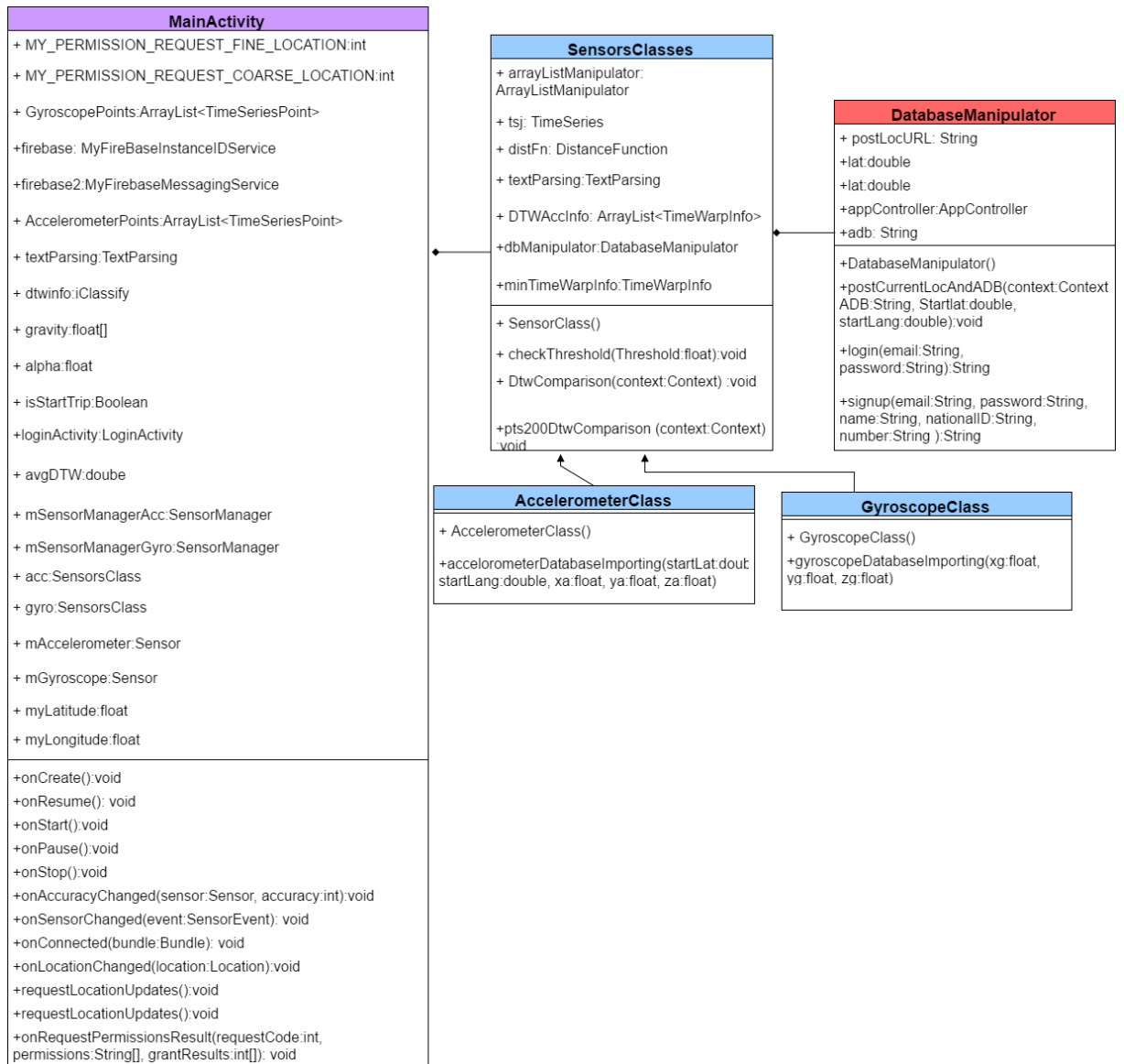


Figure 9: This shows two different sensors(Accelerometer and Gyroscope) and how they are connected to the main activity and database manipulator class.

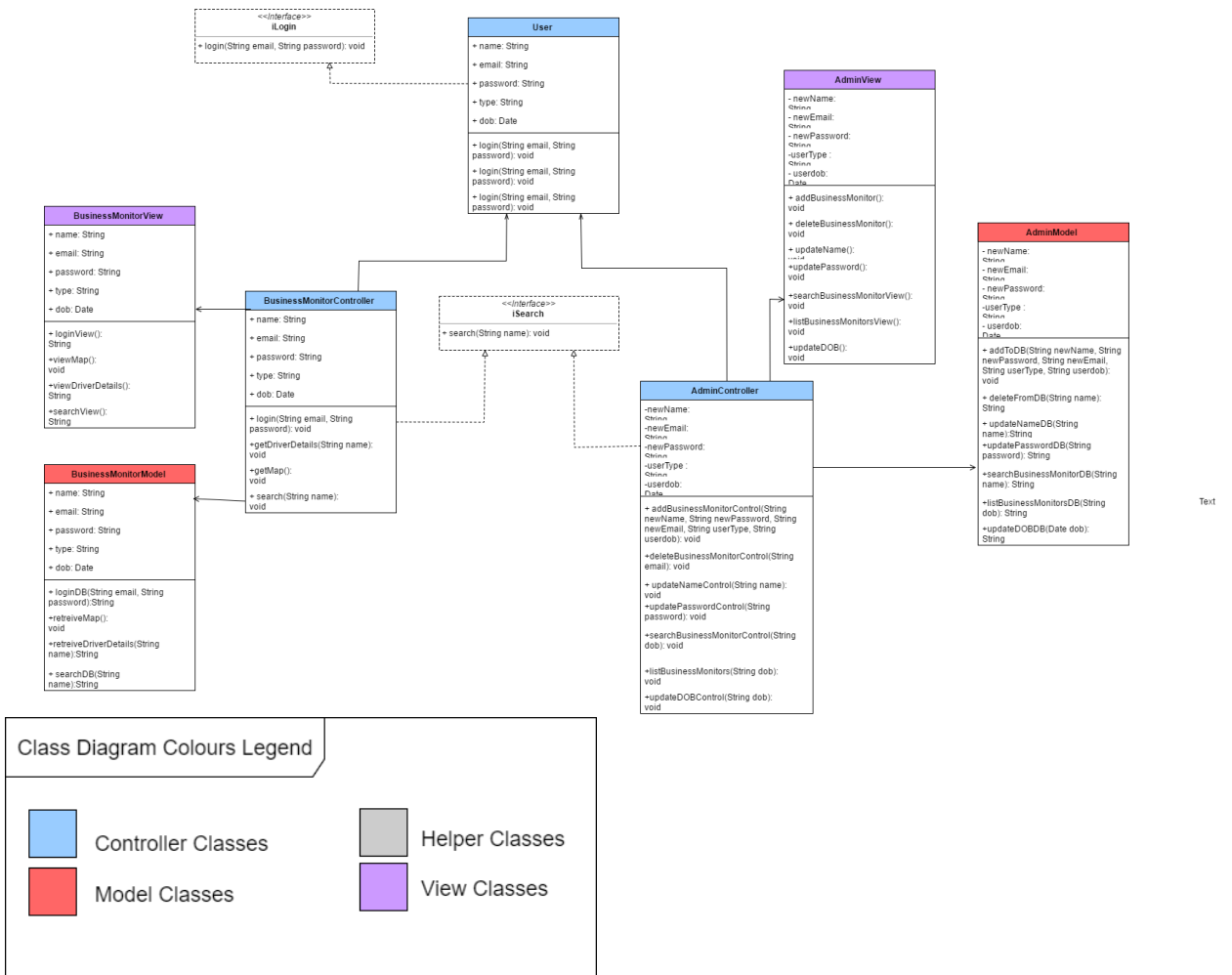


Figure 10: This figure is the class diagram of the web portal and legend for colours illustrated.

3.2.2 Sequence Diagrams

This is a sequence diagram showing the sequence of the application for the classification done on the accelerometer's sensor readings.

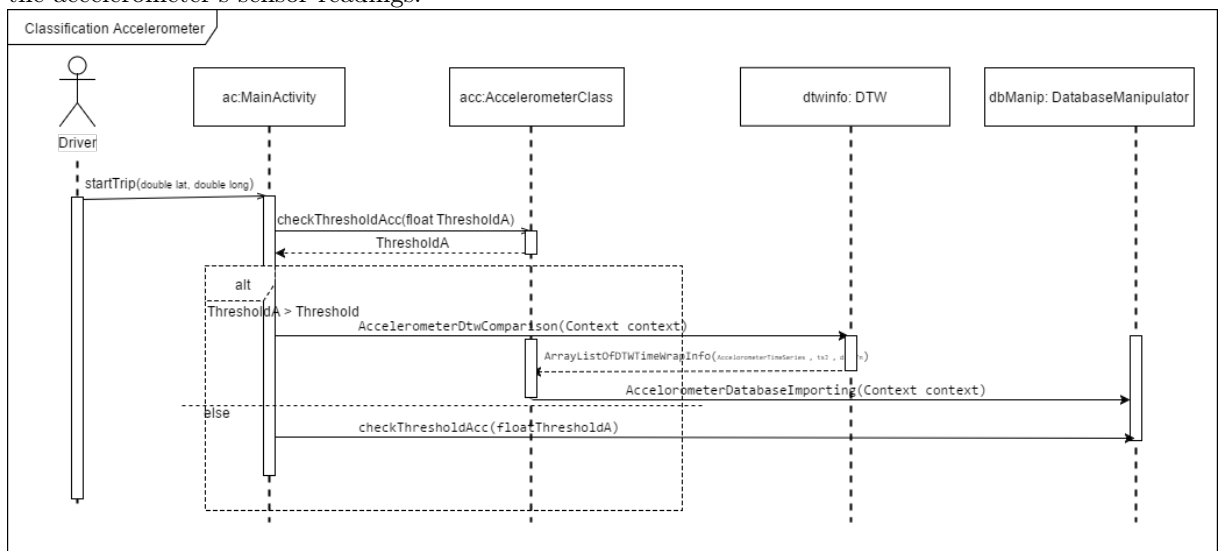


Figure 11: Accelerometer Classification Sequence Diagram.

This is a sequence diagram showing the sequence of the application for the classification done

on the gyroscope's sensor readings.

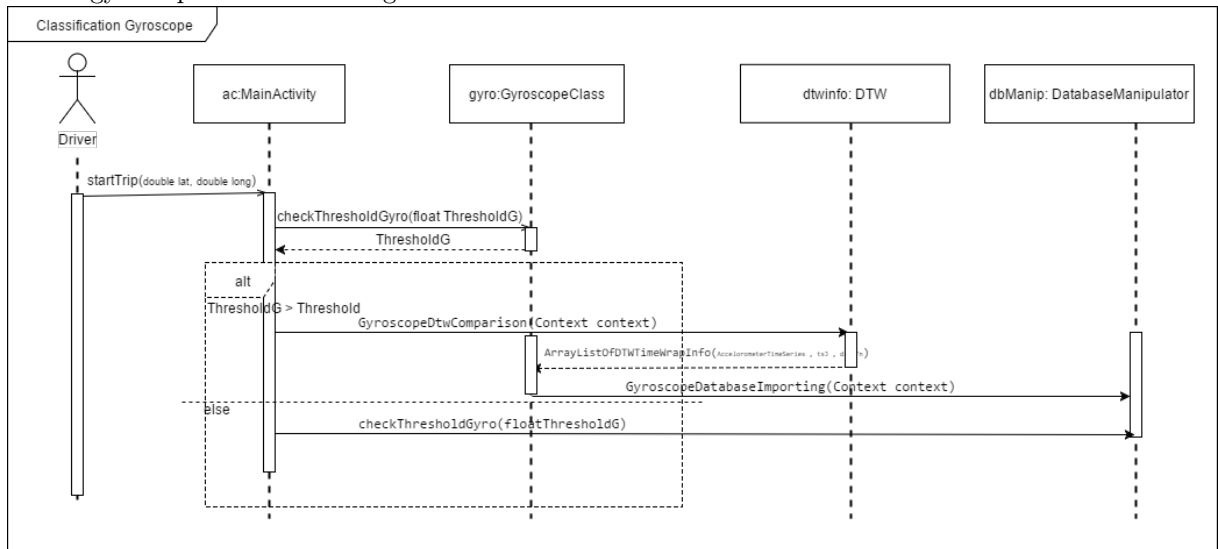


Figure 12: Accelerometer Classification Sequence Diagram.

This is a sequence diagram showing the sequence of the rating.

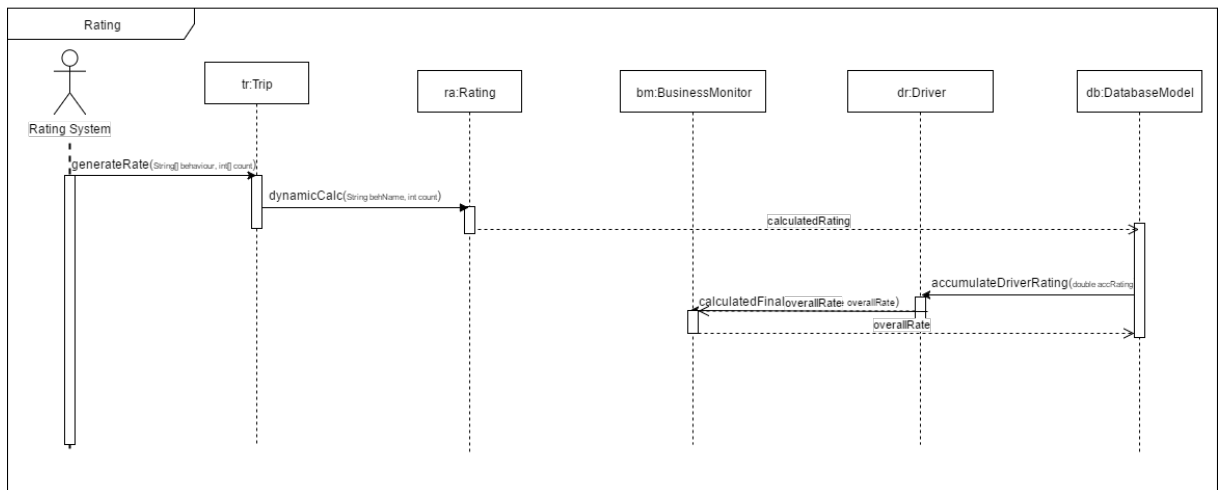


Figure 13: Rating Sequence Diagram.

3.3 Design Rationale

Since the Amelio-rater is based on MVC, it's architecture diagram is in the MVC form. MVC is the separation of Model, View and Controller classes. Main advantage of MVC architecture is differentiating the layers of a project in Model,View and Controller for the Re-usability of code and better maintenance.

4 Data Design

4.1 Data Description

This is a relational preview of the database.

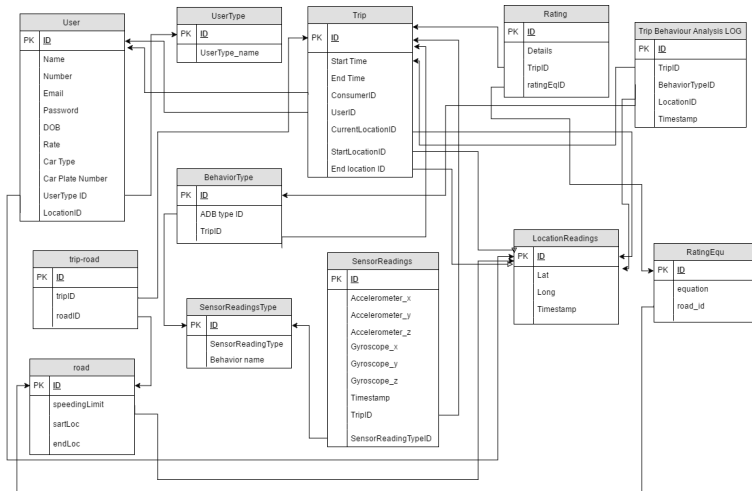


Figure 14: Relational Database.

As the Amelio-rater has to have fast posting of data and data retrieval (speed sensitive), we use MongoDB non-relational database, and the mapping of the relational database to non-relational is illustrated below.

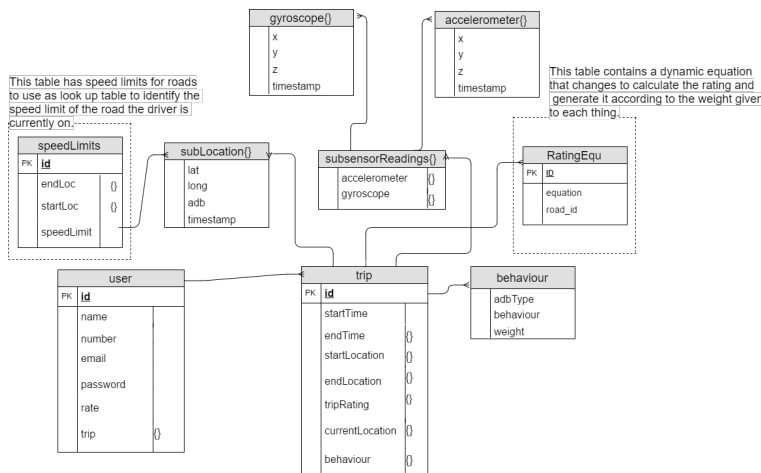


Figure 15: Non-relational Database.

- **User:** This collection has the details of the users of the Amelio-rater. It stores their details like name, number, email, password, overall rate, and a reference to their trips.
- **Trip:** This collection has the details of the trips performed by the users(drivers) of the Amelio-rater. It stores details like start and end time of the trip, start and end location of the trip, trip rating, current location, and a reference of the behaviours done by the driver in each trip.
- **Behaviour:** This is a collection with static data which consists of the ADB type(an abbreviation of the behaviour), the behaviour’s name, and the weight of the behaviour.
- **Sublocation:** This is a collection that has details of the location which are the lat, long, behaviour that is done at that location, and the timestamp. Locations from this table is used by the trip collection to get the start, end, and current location. This data may be used to perform analysis on roads.
- **Speed Limits:** This collection has static data stored to use to look up at when wanting to know if the driver is exceeding a speed limit. Data stored in this collection is start and end location of roads with their corresponding speed limits. Current location and speed of the driver is compared against these data to notify the driver if (s)he has exceeded the speed limit.

- **Rating Equ:** This collection stores a dynamic equation that is used to calculate the rating of the driver based on the behaviours done. The equation uses the behaviours that are classified by the system.
- **Sensor Readings:** This collection has a reference to the two collections: accelerometer and gyroscope. The readings are those of the accelerometer and gyroscope sensors which are built-in the drivers' smartphones.
 - **accelerometer:** This collection stores the x, y, z readings of the accelerometer sensors and the timestamp at which those readings were taken.
 - **gyroscope:** This collection stores the x, y, z readings of the gyroscope sensors and the timestamp at which those readings were taken.

Retrieval from the non-relational database is in form of JSON strings or objects. Here is an example of how they would look like.

```

{
  "_id" : ObjectId("58dac9e3c0fba914e42dd6ae"),
  "name" : "passant",
  "number" : "01000437282",
  "email" : "passant@example.com",
  "password" : "$2a$10$7zeaC14VZUVepVA158hbY0jmViDDm7xEtjb7r2G1kixIORCop5AMy",
  "trips" : [
    ObjectId("58dacaec198be00f06c0507")
  ],
  "__v" : 1
}

```

Figure 16: JSON Object Format Example.

5 Component Design

5.1 DTW : Dynamic Time Wrap

Used to find the similarities between two time series as figure shown below; showing two separate time series and the vertical lines indicating the similar points in each time series.

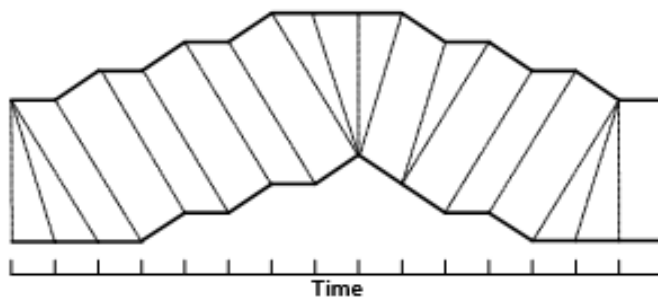


Figure 17: A warping between two time series[11].

The DTW measure the distance between the two time series using the Euclidean distance calculation which is the sum of squared distances of each nth point in one time series to the nth point in the other time series[11]

In order for the DTW to find similarities between two time series, the lengths of the time series may vary[6]. As mentioned above that it uses the Euclidean distance as a distance measure; however, the results of that are not intuitive. This occurs when the time series are slightly or very much out of phase; DTW however overcomes that by ignoring both local and global shifts in the time dimension[12]. Difference is illustrated in the diagrams below .

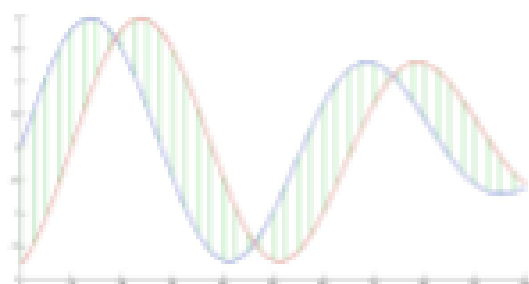


Figure 18: Euclidean Distance[6].

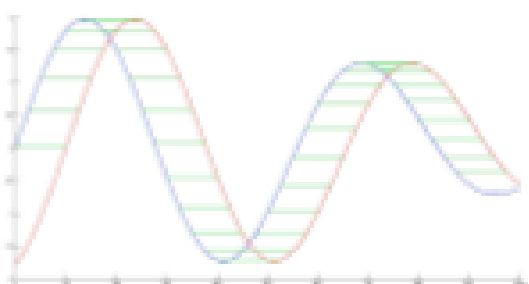


Figure 19: Dynamic Time Warping[6].

We chose the DTW algorithm for the Amelio-rater after researching for accuracies levels of different algorithms for similar gestures. A comparative study [1] showed that testing samples were correctly classified with accuracy 99.7 percent; which is fairly high but the gestures type was human gestures whereas we will be using driving behaviours and road conditions.

5.1.1 Primitive Results

When we tried classifying the direction alongside the behaviour, the results were not accurate and accuracy was better when both directions for each behaviour were combined together into one template for that behaviour to compare against.

2 templates for each behaviour		
ADB	Accuracy	
SLL	8/9	
SLR	5/9	
SWLR	0/9	read as SLL and SLR one after the other
SWRL	0/9	
W	0/9	read as SLL , SLR and SWLR

Figure 20: This is a confusion matrix for the results that were collected when using the directions of behaviours as well as the behaviour; which initiated the idea of joining both aspects into one template for the behaviour.

ADB	Accuracy	
SL	15/18	
SW	9/18	
	Window of 200pts: 0/9	read as SW and SLR
W	Window of 300pts : 9/9	

Figure 21: This is the confusion matrix for when the directions were merged as one behaviour.

For the weaving, we found that the window manipulation taking place is not fit for it. So we tested with a window of 200 points and a window of 300pts; and as shown in the figure above the window with the 300 points the weaving is classified correctly. The issue here was discovered that the window size needs to be bigger as it segments the data read and the weaving is read as Single Weave. Another option would be taking 80 percent only of the new window with the 20 percent of the previous window using that to classify the data by checking part of the data from the previous window.

5.2 KNN : K Nearest Neighbour

K-nearest neighbor is a classification algorithm where a set of examples is collected and each one has a known class. This set is used as a training dataset for the algorithm so when a new data is fed to it a comparison takes place between the training dataset and the new data. Thus the nearest k training datasets to the testing data and its labels are checked. Among the k training datasets that have been selected, the label that is found repeated the most is the label set to the testing data; thus classifying it as that label. KNN has different calculation methods; they are euclidean, manhattan and cosine similarity distances[7].

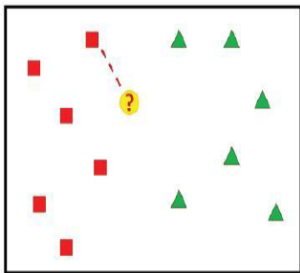


Figure 22: The 1-NN decision rule: the point ? is assigned to the class on the left[7].

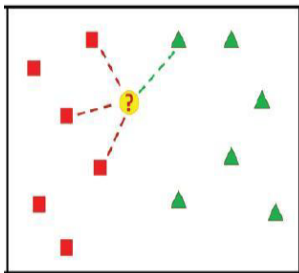


Figure 23: The KNN decision rule, with K= 4: the point ? is assigned to the class on the left as well[7].

5.2.1 Equations

- Euclidean Distance

This gives efficiency and productivity[4]. It calculate the square root of the square of two different points of the given data, which is the distance between two points in Euclidean space[13]. Formula is shown below.

$$Dist_{xy} = \sqrt{\sum_{k=1}^m (x_{ik} - x_{jk})^2}$$

Figure 24: Euclidean Equation[10].

- Manhattan Distance

This is a simple absolute magnitude of the difference between two points from the given data. Formula is shown below.

$$Dist_{xy} = |X_{ik} - X_{jk}|$$

Figure 25: Manhattan Equation[10].

- Chebychev Distance(Maximum Value Distance)

Absolute magnitude of the difference between two points from the given dataset. Formula is shown below.

$$Dist_{xy} = \max_k |x_{ik} - x_{jk}|$$

Figure 26: Chebychev Equation[10].

NN(Nearest Neighbour) is found based on the value of **K**. This is an effective and simple approach in the case of large training datasets. The drawback of this approach is that it is based on weights[2].

The kNN algorithm is as shown below[8]:

1. Determine k nearest neighbors and D set of training example.
2. for each test example xi do Calculate d (xi, yi) based on distance measure.
3. Select the k closest training examples yi to test example xi.
4. Use majority voting to classify the test examples
5. End for.

5.3 ANN : Artificial Neural Network

ANN is a mathematical/computational model and simple clustering of primitive artificial neurons that is inspired by the way the biological nervous system such as brain information system; hence the term neural network[9]. It is made up of "interconnected artificial neurons" that programmatically mimic biological neurons behaviours; those neurons work simultaneously to solve a specific problem introduced to them.

The input layer in the structure of the ANN(shown in the diagram below, which is the same for all ANNs) receives data either from input files or directly from electronic sensors in real-time applications. The output later sends information directly to the outside world or other devices that have mechanical control systems. Between both layers is many hidden layers containing many neurons in various interconnected structures.

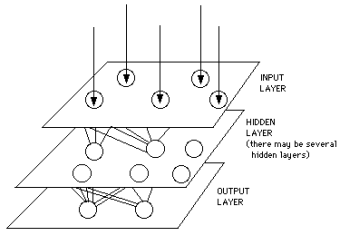


Figure 24: A simple Neural Network Diagram[9].

There is a concept called competition, where some networks want a neuron to inhibit other neurons in same layer and this is mainly used in the output layer. This is known as literal inhibition. There is another connection type called feedback; here the output layer routes back to a previous layer. The diagram below demonstrates both concepts and connections.

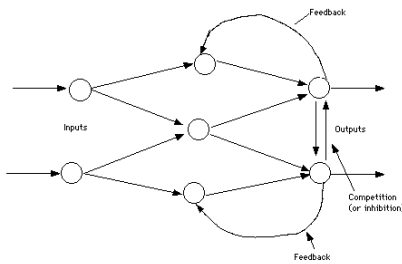


Figure 25: A simple Neural Network with Feedback and Competition[9].

Training the ANN algorithm could be supervised or unsupervised (Adaptive Training).

5.4 SVM : Support Vector Machine

SVM is a classifier that is a powerful machine method developed from statistical learning and has made achievement in a specified field; it is applied on different data that have two or multi classes[5].It has many attractive features and promising empirical performance; as for instance, it does not suffer limitations of data dimensionality and limited samples[3][14]. Performance of SVM is very sensitive to how the cost parameter and kernel parameters are set. For some datasets that are in a 2D view, it is necessary to move it to a 3D view. Mapping the data into a higher dimension is known as kernelling.

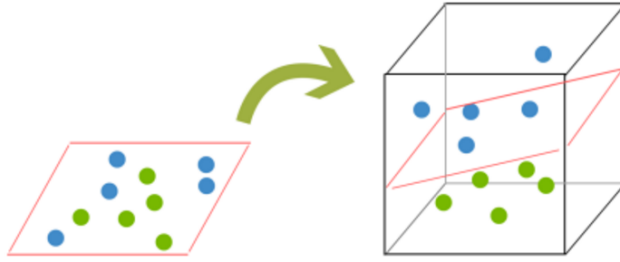


Figure 26: Kernelling demonstration of separating balls of different colours when suspended in the air[5].

SVM is well known for its accuracy and more efficient given that it uses a subset of training points since it works well on smaller, cleaner datasets. On the other hand, it is less effective on noisier datasets with overlapping classes.

5.5 Cross Validation - Training Dataset Templates Selection Method

We have collected dataset for multiple abnormal driving behaviours from the different sensors - accelerometer , gyroscope and gps - in the last 4 months. The behaviours include: sudden lane change, Single weave, weaving (W), speeding (S), and sudden braking (SB). Each behaviour's training dataset template contains sensors' readings related to that behaviour only. Three training datasets for each behaviour has been selected from among eleven different training datasets made for each behaviour. The selection is done based on cross validation. The cross validation process occurs on each behaviours' datasets separately. Each template from the eleven training dataset templates of a behaviour is compared against the rest ten templates and the distance between it and each other template is derived and then the average of these distances is calculated. After average distance is calculated for all 11 templates, the three templates with the minimum average distances are selected to be the training dataset templates to compare the testing datasets against. Same is done with all behaviours' dataset templates to obtain a library of training dataset templates for all behaviours to compare the testing datasets against along the way[6].

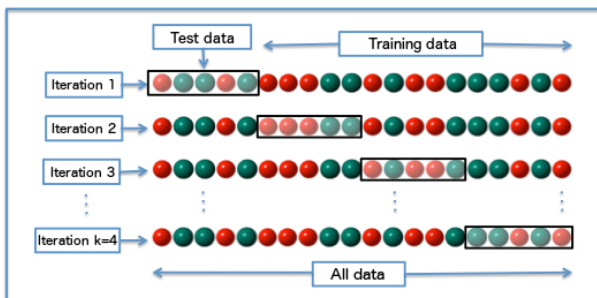


Figure 27: Cross Validation Illustration (K-fold where $k = 4$).

6 Human Interface Design

6.1 Overview of User Interface

Since there is a web interface for the Amelio-rater for the business monitor end user, and an android based mobile application for the driver end user; therefore, we have two different user interfaces.

- Business Monitor Web Interface: The business monitor will be given his/her account to be

able to log in. Once logged in, (s)he is redirected to the homepage which contains the real time monitoring map. All web pages will be provided a navigation menu, too be able navigate through the website as desired. The menu has three buttons for navigation: one for the monitoring page, one for the search page, and one to sign out of the currently logged in account. The homepage will contain a map interface powered by Google, and on this map will be previewed the cars currently performing trips. The cars' positions will be updated on the map according to it's current location in the trip. Each car will be previewed with a colour; this colour depends on the behaviour the driver is currently doing. The web page will have a legend for those colours below the map interface, in order for the business monitor to be able to identify which colour stands for which behaviour. Next we have the search page, in this web page the business monitor will be able to search for a driver by name. (S)he will enter the name of the driver (s)he desires, and the drivers with that name will appear in the web page for the business monitor to choose from. Once a driver is chosen, the business monitor is redirected to another web page which consists of the driver's accumulated rating and details of all the trips performed by that driver including: trip date, trip rating, and number of abnormal driving behaviours performed in that trip.

- Driver Android Application Interface: The driver will be able to log in to his/her account and will be shown only one activity screen which will allow him/her to start and end a trip.

6.2 Screen Images

Here are some screenshots from the web interface of the business monitors' web portal.

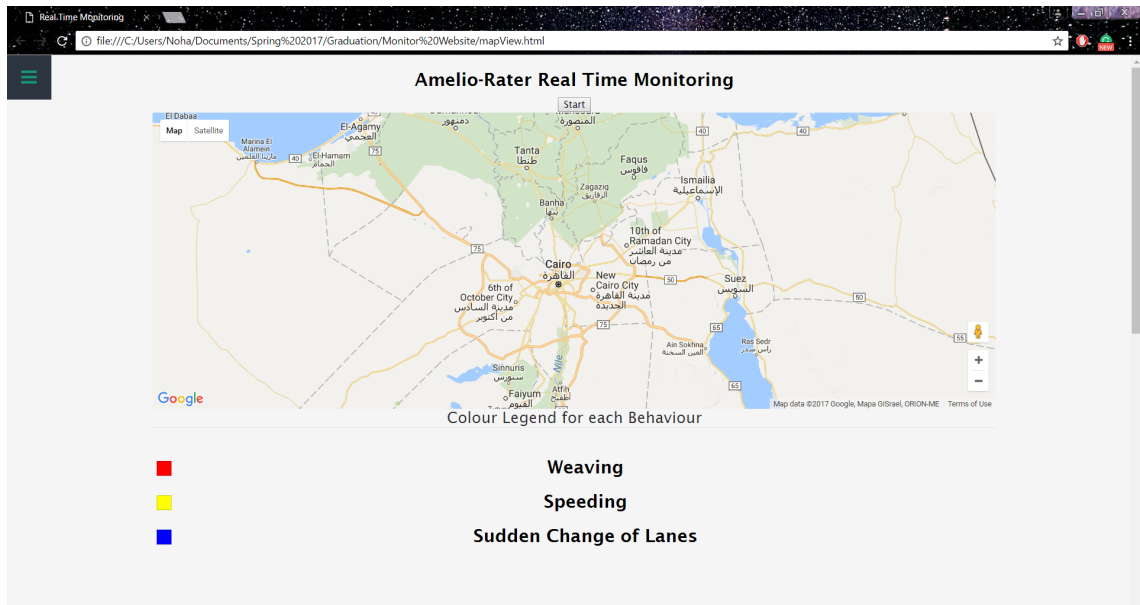


Figure 28: Real Time Monitoring Web Page.

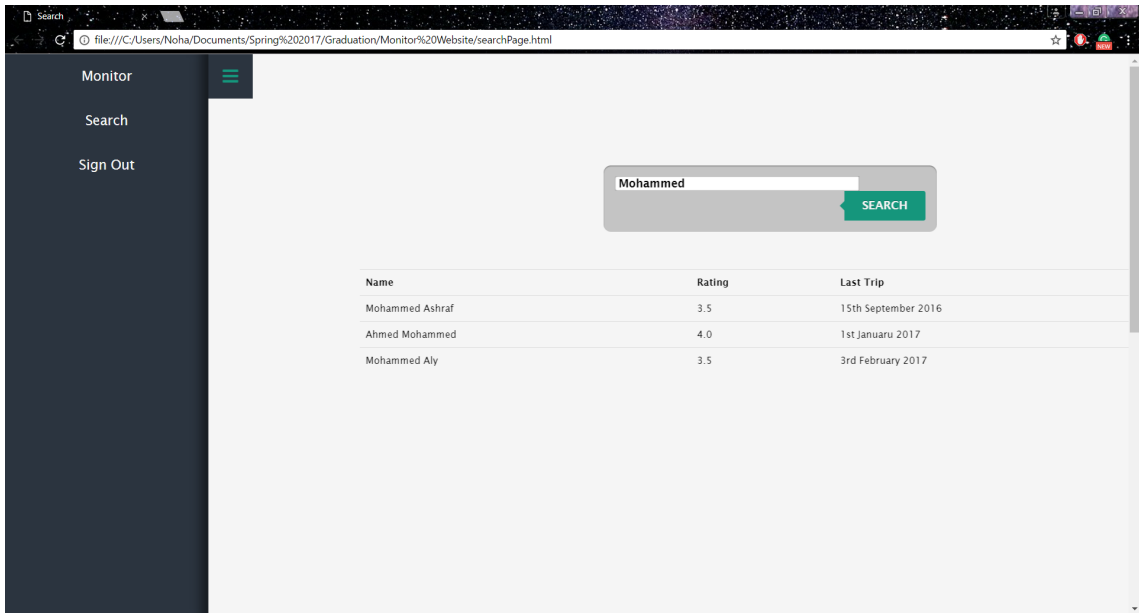


Figure 29: Search Web Page.

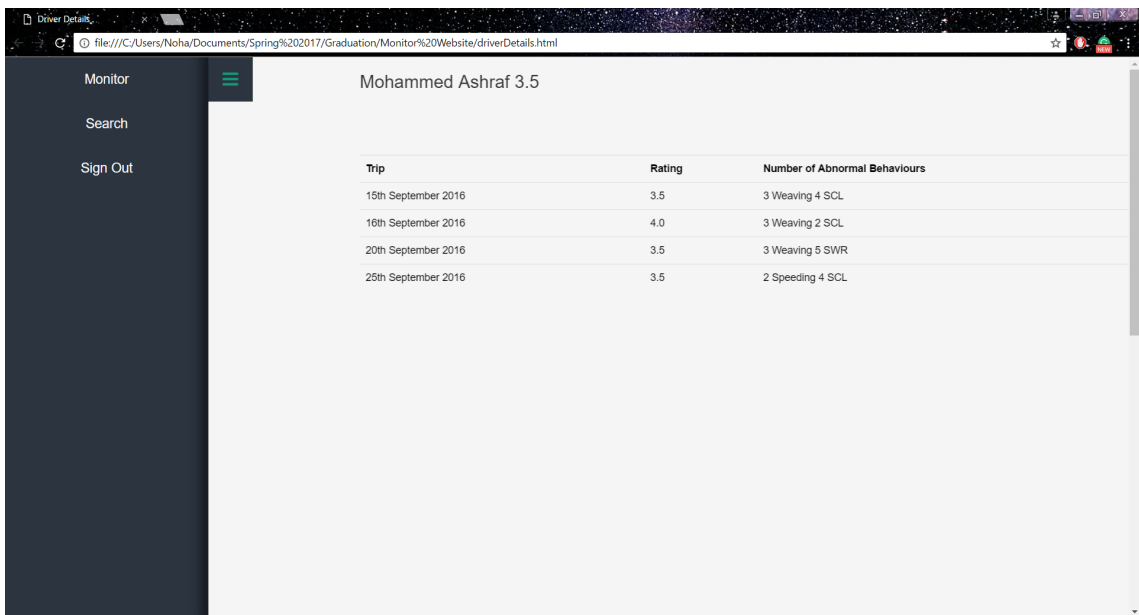


Figure 30: Driver Details Web Page.

The main activity of the driver showing the accelerometer and the gyroscope results and the action buttons.

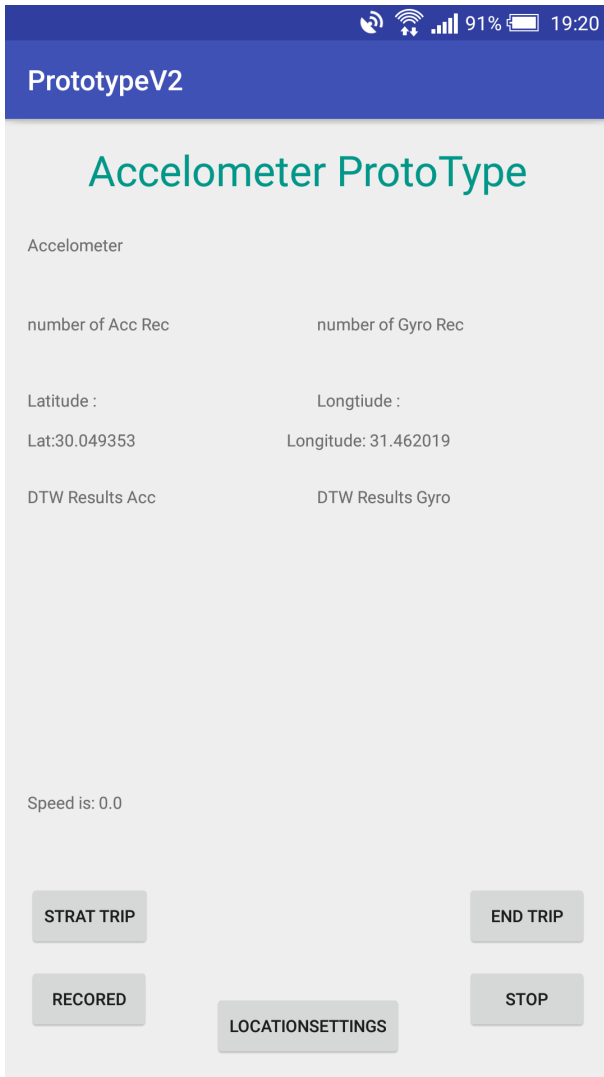


Figure 31: Driver's start and end trip activity.

6.3 Screen Objects and Actions

Business Monitor Web Portal:

- Navigation Buttons

- Real Time Monitoring

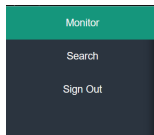


Figure 32: Navigate to Real Time Monitoring Web Page.

- Search

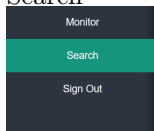


Figure 33: Navigate to Real Time Monitoring Web Page.

– Sign Out

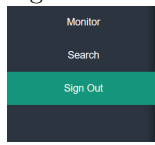


Figure 34: Navigate to Real Time Monitoring Web Page.

- Search Button, to search for a specific driver by name.

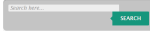


Figure 35: Navigate to Real Time Monitoring Web Page.

- Selecting a hyperlinked driver name to redirect the web page with his/her details.

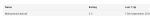


Figure 36: Navigate to Real Time Monitoring Web Page.

Driver's Android Application:

- Start Trip Button

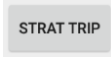


Figure 37: The button that allows the driver to start a trip where (s)he will be previewed in the real time monitoring from that point.

- End Trip Button

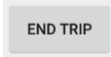


Figure 38: The button that allows the driver to end a trip where (s)he is no longer being monitored and the triggers automatic rating calculation for the trip.

7 Requirements Matrix

FR/SDD Module	Driver Signup	Driver Login	View Details	Start Trip	End Trip	Admin Register BM	Admin Edit BM	Admin List BM	Admin Search BM	Admin Delete BM	BM Login	BM View RTM	BM Search Driver	BM View Driver Details	Detection /Classification of ADB	Generate Auto Rating	Notify Driver
Driver Signup	X	X											X	X			
Driver Login		X	X	X													
View Details			X														
Start Trip				X	X							X			X		X
End Trip					X									X		X	
Admin Register BM						X	X	X	X	X	X						
Admin Edit BM							X										
Admin List BM								X									
Admin Search BM									X								
Admin Delete BM										X							
BM Login											X	X	X				
BM View RTM												X					
BM Search Driver													X				
BM View Driver Details														X			
Detection /Classification of ADB												X			X	X	X
Generate Automatic Rating																X	
Notify Driver																	X

Figure 39: Requirements Matrix, mapping between functional requirements and the sdd modules.

References

- [1] Aya Hamdy Ali, Ayman Atia, and Mostafa Sami. “A comparative study of user dependent and independent accelerometer-based gesture recognition algorithms”. In: *International Conference on Distributed, Ambient, and Pervasive Interactions*. Springer. 2014, pp. 119–129.
- [2] T Bailey and AK Jain. “A note on distance-weighted k -nearest neighbor rules”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 4 (1978), pp. 311–313.
- [3] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. “A training algorithm for optimal margin classifiers”. In: *Proceedings of the fifth annual workshop on Computational learning theory*. ACM. 1992, pp. 144–152.
- [4] Belur V Dasarathy. “Nearest neighbor ($\{NN\}$) norms: $\{NN\}$ pattern classification techniques”. In: (1991).
- [5] K Srivastava Durgesh and B Lekha. “Data classification using support vector machine”. In: *Journal of Theoretical and Applied Information Technology* 12.1 (2010), pp. 1–7.
- [6] Nicholas Gillian, Benjamin Knapp, and Sile O’Modhrain. “Recognition Of Multivariate Temporal Musical Gestures Using N-Dimensional Dynamic Time Warping.” In: *NIME*. 2011, pp. 337–342.
- [7] Sadegh Bafandeh Imandoust and Mohammad Bolandraftar. “Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background”. In: *International Journal of Engineering Research and Applications* 3.5 (2013), pp. 605–610.
- [8] Asha Gowda Karegowda, MA Jayaram, and AS Manjunath. “Cascading k-means clustering and k-nearest neighbor classifier for categorization of diabetic patients”. In: *International Journal of Engineering and Advanced Technology* 1.3 (2012), pp. 147–151.
- [9] Sonali B Maind, Priyanka Wankar, et al. “Research paper on basic of Artificial Neural Network”. In: *International Journal on Recent and Innovation Trends in Computing and Communication* 2.1 (2014), pp. 96–100.
- [10] Punam Mulak and Nitin Talhar. “Analysis of Distance Measures Using K-Nearest Neighbor Algorithm on KDD Dataset”. In: () .
- [11] S Salvador and P Chan. “FastDTW: Toward accurate dynamic time warping in linear time and space. 3rd Wkshp. on Mining Temporal and Sequential Data, ACM KDD’04”. In: *Seattle, Washington (August 22–25, 2004)* (2004).
- [12] Stan Salvador and Philip Chan. “Toward accurate dynamic time warping in linear time and space”. In: *Intelligent Data Analysis* 11.5 (2007), pp. 561–580.
- [13] Archana Singh, Avantika Yadav, and Ajay Rana. “K-means with Three different Distance Metrics”. In: *International Journal of Computer Applications* 67.10 (2013).
- [14] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.