Measuring Driving Behaviour Using Phone Sensors and API's

Preetpal Singh

UP941594

School of Computing
Final-year Project
Engineering

Abstract

Over the years, an increase in number of car transports in the UK has raised concerns about road safety especially when over 77% of households owns a car. Since driving a car requires a lot of skills and experience, the lack of driving measuring tools becomes a concern for majority of the people. Considering the scenario, the prototype aims to utilise both machine learning and other algorithms to build a platform for car drivers to measure four main driving behaviours including dangerous maneuverers, rapid acceleration, harsh braking, and over-speeding. The outcome of this prototype would be a mobile website in which users can interact with the system with necessary features extracted from conducted questionnaire and user scenarios. The system will be developed using necessary machine learning models found from the literature and tested using relevant validation methods to ensure reliability and performance. The ultimate results will raise opportunities for future developments to make refinements to the system and produce an overall end-software which can be released in the coming future.

Acknowledgement

The author would like to thank both supervisor Dr. Zhaojie Ju and moderator Dr. Honghai Liu to become part of this project and their contribution towards providing wonderful guidance throughout challenging times to help produce valuable work in the final year at university. Also, a big thanks to family and all friends for all their help and support throughout.

Table of Contents

Chapter 1 – Introduction

1.1 Project Context

In recent years attention has been focused mainly on the increasing number of transports in the UK, in which cars are the most popular way of travelling to work (Statista, 2022). However, driving a car itself requires a good level of experience and skills that inspires drivers to travel safely for longer journeys. A study indicates that improvements in self-awareness by training in traffic situations decreases the risk of collision (Amado et al., 2014). Additionally, the driver should have a perception of their driving skills to ensure the safety of road users (Sundström, 2008). Nevertheless, Individuals have rated their driving ability better than the average drivers during a self-assessment study of driving skills (Pugnetti & Elmer, 2020), although some drivers consider themselves slower, safer and less accident liable according to (Milleville-Pennel & Marquez, 2020).

With regards to technology, smartphone devices at present are capable of sensing various motions with the use of embedded sensors like accelerometer, gyroscope, or GPS (Cervantes-Villanueva et al., 2016). The increasing capacity of sensing allow smartphones to monitor and profile driving behaviours, resulting in increased attention in smartphone-based telematics systems (Castignani et al., 2015). For the safety of other road users, increased awareness about vehicle's behaviour is beneficial for car drivers (Fazeen et al., 2012), allowing individuals to enhance their driving performance using their own smartphones.

This project aims to develop a mobile website, a platform for car drivers to monitor their driving behaviour. A website that focuses on four main skills of car drivers such as car acceleration, harsh braking, driving maneuverer and over-speeding. The system would benefit drivers, allowing them to reflect upon their previous driving journeys. Ultimately, it encourages individuals to compare their driving skills with peers, increasing driving awareness by improving their skills and protecting them from other road risks.

1.2 Project Aims and Objectives

Aim:

The aim of this project is to train machine learning models on the data collected from smartphone sensors to observe a variety of patterns from different driving actions. Additionally, predictions will be made on the outcome of patterns and the driving behaviour data will be presented on a mobile website.

Objectives:

1. Gather data from smartphone sensors
   - Request user permission to access smartphone sensors for the website to perform further functionality.
   - Collect data from applicable sensors such as accelerometer, gyroscope, and global positioning system (GPS).
   - Use appropriate data filtering techniques to choose the right proportion of data for further calculations.

2. Train and apply machine learning algorithms
    - Conduct a literature review on how various machine algorithms have been used to solve similar problems.
    - Justify the choice of appropriate algorithms as well as understanding the concept behind each algorithm.
    - Apply selected algorithm on filtered data from different driving actions to classify various driving behaviours.

3. Develop a mobile website for end-users
    - Setup HTTPS server locally for IOS devices to request user permission for sensors.
    - Store collected data in relational database for the purpose of better data structuring such as tables in SQL.
    - Represent data on a user interface for end-users to easily analyse their driving behaviour.

1.3 Project Constraints

- Due to limited time and other modules running simultaneously, it will be a challenge for the author to learn, apply and program the right machine learning algorithms for the purpose of solving the highlighted problem.

- The data collected from phone sensors could vary on different smartphone devices therefore it may affect the core functionality of the system. Essentially, testing should be performed on different devices to resolve this issue.

- Since the website is developed for mobile users, majority of the debugging and testing is likely to take place on mobile devices. It will be a challenge as the mobile browsers lack a debugging tool for developers.

Chapter 2 – Literature Review

2.1 Smartphone sensors and API's usage

Modern mobile devices have built-in accelerometer sensor that measures the acceleration forces applied to the device, where a gyroscope sensor has the capability to measure the rate of rotation. Both sensors type collects readings around 3 axes (x, y and z). Embedded smartphone sensors have been used in many applications from human activity tracking (Ronao & Cho, 2016) to transportation modes detection (Xia et al., 2014). In the context of safe driving, these sensors have aided multiple researchers to study driving behaviour such as drunk driving detection or an analysis of various driver behaviours (Papadimitriou et al., 2019). In research from (Dai et al., 2010) has analysed drunk driving related behaviour based on lateral and longitudinal movement of the vehicle with the use of accelerometer and orientation sensors. The study has highlighted these mechanical sensors to be more efficient and accurate, as well as consume little energy while in use and affordable as they are already installed into mobile devices. (Li et al., 2021) has made a comparison of different sensory data (i.e accelerometer, gyroscope, magnetometer), in result the study has labelled accelerometer and gyroscope as the most suitable sensors for detecting vehicle maneuverers. Although, the sensory data can capture complete vehicle motions, it can be useful for detecting other driving events such as harsh braking or aggressive acceleration. To detect such events, the right portion of the data is utilised for further processing and classification. For instance, during harsh braking event, the Z-axes from accelerometer sensor will be affected if the phone is attached to the vehicle's windshield.

Application Programming Interface (API) is considered as valuable tool as mobile sensors. Geolocation API provides set of objects allowing websites to request location information of a device in GPS coordinates (Pejic et al., 2010). The fact that it is widely available in rural areas, the location coordinates can be accessed right from the browser. There are several implementations of other APIs in studies, Google Maps Directions API offers route information for a given destination, are easily accessible in the use of transportation from cycling to driving (Schrader-Patton et al., 2010). The use of Google Roads API has proven to be effective in the case of retrieving additional data about different roads a vehicle is travelling on such as road speed limit, nearest roads segments. Ultimately, the provided data from these APIs can be processed for further calculations such as finding the current location of a device, measuring over speeding of a vehicle on a certain road. These tools are highly effective for the purpose of this research, though previous driving detection related research studies have not shown indications of these API. However, these APIs will allow the author to measure rapid car acceleration and over speeding of a vehicle as well as implementing further techniques to achieve desired results. Since, both over-speeding and rapid acceleration events have not shown a change in both accelerometer and gyroscope sensor data, these events can be detected using the APIs.

2.2 Previous driving behaviour studies

Multiple studies have been conducted to measure the driving behaviour with the use of smartphone sensors. Although some research findings have performed classification of events like aggressive or drunk driving detection, where others have focused mainly on manoeuvres, lane changing or overtaking. The study carried out by (Dai et al., 2010b) collected data from accelerometer and gyroscope sensors. The research detected dangerous or abnormal driving manoeuvres related to drunk driving situations while the system was programmed to detect the device orientation in a moving vehicle. However, (Ramah et al., 2021) trained deep learning models to classify driving manoeuvres on the training dataset collected from smartphone sensors. There are several other implementations of detecting driving behaviour. (Castignani et al., 2015) proposed a fuzzy logic mechanism to categorise driving actions such as over-speeding, accelerating and braking. A scoring system was then developed that considered different weather conditions and rush hours to profile a driver's behaviour.

The approach taken by (Ramah et al., 2021) estimated the phone's orientation and position with respect to vehicle's frame. The method turns out to be suitable when the vehicle is on a motorway and the phone is attached to the vehicle's dashboard. However, this approach could be improved if the phone was mounted to vehicle's windscreen instead of dashboard to get complete range of vehicle motions. As this would also prevent the device from collecting dashboard vibrations in the sensor data which further requires the filtering of noisy data. There are several studies that have experimented this approach when the phone was fixed to vehicle's windscreen as this is one of the effective ways to capture driving events (Carvalho et al., 2017).

2.2.1 Data Collection and Feature Extraction

In research from (Johnson & Trivedi, 2011), vehicle movement classification is performed using sensor fusion output of gyroscope, accelerometer and magnetometer, shown that gyroscope signals are used to indicate maneuverers as well as used other 2 sensors to capture movements of the vehicle. Since, the sensory data comes in 3 axes, the study extracted y-axis from gyroscope, y-axis from accelerometer and x-axis from magnetometer for vehicle manoeuvres classification. However, the samples taken from sensors in this research were only at a rate of 25Hz which leads to lower number of measurements taken and potentially least accurate results of the model. Additionally, for the purpose of detecting manoeuvres, the study has used y-axes gyroscope signals meaning the device was attached to vehicle's windshield. In another research, (Li et al., 2021) developed an application to collect sensor data for the purpose of collecting training samples, allowing them to customise frequency rate beforehand as well as saving data automatically to a CSV file.

To overcome the problem of collecting inadequate data for driving behaviour profiling, (Carvalho et al., 2017) has used a dataset available on Github repo. The available dataset is sufficient as it includes multiple samples of different driving events such as aggressive left or right turn, aggressive braking as well as non-aggressive events. The experiment conducted 4 car trips of approximately 13 minutes each performed by drivers with 15 years of experience. In addition, the dataset is also useful as the mobile device was attached to vehicle windscreen and the sampling rate of data was between 50 and 200 Hz which is quite crucial for the purpose of our study. However, the events performed by each driver are captured into same CSV file, therefore the data needs to be separated into different driving events before training into machine learning.

Data transformation is considered as an integral part in (Ramah et al., 2021) research as it is useful for using the right piece of the data. The fact that training dataset is a series of data, it may involve sample frequencies in variable length, where some samples may result in no driving maneuverer performed when the car is parked or moving on a straight road. To overcome this problem, (Ramah et al., 2021b) has proposed a sliding window technique that aims to extract potential events from a given time series data. The use of The Latter segmentation approach divides time series into segments of a given size, the method has a greater impact on detecting potential events in given window size with different starting and ending times. However, the segmentation approach would only decrease the dataset into smaller chunks which still requires feature extraction. Therefore, the dataset from each window size needs to be merged into one for further data processing in machine learning. Although, feature extraction would only be required for certain ML models that cannot process time-series data. To achieve this target, the set of data will be divided by number of datasets in each window size to find the mean value. Ultimately, data transformation improves the data quality by collecting relevant data as well as increases the accuracy of resulting models to make precise predictions.

2.3 Machine learning models

Given that the data gathered from different driving events needs to be processed using data transformation techniques such as time-series segmentation or feature extraction. However, to classify various driving events, there is a need of machine learning models to assign extracted data to a set of categories using classifiers. Though, these classifiers would only be useful for maneuverer detection and harsh braking of a vehicle, for other 2 types of behaviours the author would use the APIs to design algorithms that measure over-speeding and rapid acceleration events.

2.3.1 Support Vector Machine (SVM)

SVM is one of the widely used supervised based machine learning model, used in various classification problems such as image processing, speech recognition, pattern classification and protein classification (bioinformatics). The fact that the statistical learning model has the capacity to work on large datasets, it has also achieved a superior classification performance against other machine learning models according to (Tango & Botta, 2013). The method has its kernel function to map data to a higher dimensional space, followed by separating the training instances to identify the maximum margin hyperplane. The given datasets fall into each side of the hyperplane forming its own class, resulting in separated data into different classes. The study conducted by (Tango & Botta, 2013) has figured out that SVM model is able to classify both positive and negative instances, indicating the driving events can classified more accurately. Additionally, on average the model completed its tasks with a time frame of 47.06 seconds which was relatively low in comparison to other ML models. However, SVM does not support multiclass classification which could result in data classified into 2 classes. In driving turning event detection, this approach would not suitable as it would only classify maneuverers as aggressive or non-aggressive which is quite straightforward. Nevertheless, due to its popularity and the accuracy of ending results, for the purpose of our driving behaviour study, the method can be useful for identifying harsh braking events. Since, the likelihood of braking events is comparatively low, and the past studies have also considered this behaviour as least important in driving behaviour situations. Therefore, the author has considered SVM model for the classification of “normal” vs “harsh” braking events.

2.3.2 K-Means Clustering

K-Means clustering is another example of ML model used for clustering datasets into related groups when number of clusters are known. The method has been recognized to be effective and easy to implement in unsupervised machine learning style. The fact that the model has been originated from signal processing, it has been used for clustering customer segmentation in various business studies (Kansal et al., 2018). However, the model has been considered as least commonly used approach in previous driving detecting studies. There is no clear evidence indicating that K-means clustering can provide results with higher accuracy as well as low computational cost for the detection of different driving events.

2.3.3 Convolutional Neural Networks (CNN)

Inspired by the concept of neurons in human brain, CNN allows computer programs to solve problems by recognising patterns and the method is used in the field of machine learning. This model has a contribution in various deep learning applications like natural language processing, image classification or even driving behaviour detection. According to (Shahverdy et al., 2020), CNN consists of basic components such as convolutional layer, pooling layer, activation functions and fully connected layer. The study has applied this model for driving behaviour detection and has presented a confusion matrix to highlight the performance of the classification model, in which all instances were classified correctly such as drowsy, aggressive, inattentive, and drunk driving. Since CNN could classify events into multiple groups, the model can be useful for detecting different driving maneuverers. However, in another research from (Ramah et al., 2021c) has highlighted that the model has high precision rate of 72% when the number of samples provided are between 5 and 9. Despite the fact, CNN has overcome the problem of multiclass classification from SVM model, but the precision rate is low therefore it may have a negative impact on detecting similar maneuverers. In many use cases, the model is trained on images instead of sensory data, since it has high performance rate in classifying images, the model may not perform well in classifying sensory data type. Class imbalance is another major issue while training CNN where the data belongs in one class is lower, therefore in some driving events the model may struggle to classify events.

2.3.4 Long-Short Term Memory (LSTM)

LSTM model is type of recurrent neural network that can work with dynamics of sequences, especially in time series classification where the input data may vary in length. In research from (Wollmer et al., 2011) has introduced LSTM to detect driver’s distraction based on captured head tracking data in real traffic environment, achieving a high precision rate of 96.6%. Since the model is known as highly suitable for classification problems, the memory cells can store information for longer time periods, allowing the model to classify new events more accurately based on previous classified events. For driving events, LSTM model has overcome the problem of training sensory data as well as a high precision rate which is an important aspect of a machine learning model. At the core of this network is a sequence input layer that can take array of sensor input data, followed by LSTM layer, ultimately 3 other layers that helps predict the class labels once the model has been trained. Additionally, (Ramah et al., 2021) experimented different ML models to recognise driving behaviour in which LSTM outperformed Fast-Forward and Convolutional neural network, resulting in precision rate of 90%. Based on the literature review of all ML models, LSTM has turned out to be the most suitable and effective approach for detecting driving events, it has also overcome the problems of other models from classifying multiple events with SVM and training sequences of data in CNN.

2.4 Conclusion

To conclude, several studies have conducted driving behaviour detection using various approaches such as using multiple mobile sensors, collecting data from certain axes of either gyroscope or accelerometer, applying data transformation techniques or training various types of machine learning models. Due to the complexity of the process of measuring driving behaviour, these studies have performed limited driving events to get higher accuracy. Therefore, the proposed research would build an artefact that is able to detect multiple driving events. In addition, to measure rapid acceleration and over-speeding events, the use of APIs would be beneficial as it allows the users to get more accurate results during these events in comparison to measuring these events with sensory data. Since sensory readings does not change during these 2 events, Geolocation API is effective to measure the speed of the vehicle which can be extracted by the distance travelled between 2 coordinates points over time. The speed of the vehicle would help determine over-speeding and acceleration events. On the other hand, the use of sensory data is crucial to measure harsh braking and aggressive maneuverers. Since the dataset available in GitHub repository is sufficient, the data will be extracted for data transformation. Ultimately, LSTM model is then trained for the purpose of classifying these driving events and finally the model will be tested to figure out if it is accurate enough for building the end artefact.

Chapter 3 – Methodology

3.1 Project Planning Methodology

The project is managed using Microsoft Excel that clearly combines related practices within each task to deliver a project with detailed planning and management. The fact that online planning tools have pre-defined templates which does not bring much freedom and flexibility to best manage project workload. The use of Excel allows the author to plan each task with the help of spreadsheets in which multiple timescales, resources and word schedule can be designed from scratch. Since not all projects' goals are achieved according to the planned schedule, the ease of refinements to the planning can be done with a single click.

For the planning of this project, the tasks related to each chapter were assigned including a time frame with the order provided in the mark scheme. For each task, weekly planner was utilised to have more detailed working schedule which involves both weekdays and weekends which are suitable for the author to spend on the project. Additionally, the author can keep track of the work produced within a planned goal to figure out if there is a requirement of refinements to the plan. However, due to the flexibility of Excel, the planning does become time consuming as every task is planned from scratch including weekly planning of all months covered in the project. Afterall, the overall benefit of using this method dominates all the drawbacks which consequently results in effective and appropriate project planning that helps the project meet its objectives successfully before the deadline.

3.2 Software Development Lifecycle

3.2.1 Overview

A software methodology is critical for the purpose of creating an engineered solution from initial project idea by ensuring that the software being developed meets the purpose and gathered requirements with reliability and high level of quality. These principles can be achieved with set of methods and processes that are relevant to the problem defined above. A high-level documented methodology is followed by everyone, involving set of related activities to achieve the complexity behind a given problem and effectively produce a software system. Though, a single methodology may not be suitable for all project types due to extremely varied software scenarios and flexibility. However, the existence of various methods resolves the problem of complex software processes where a set of techniques can be combined from other methodologies to produce a methodology that best suits a complex problem.

3.2.2 Scrum (Agile)

Scrum is a process framework that implements Agile principles in which various processes and techniques can be employed to manage work on complex software. The project management and work techniques become clear in the framework to improve the software itself and the working environment. There are certain benefits that makes this approach suitable for the development of driving behaviour website, a product backlog is managed by the owner that contains a list of requirements desired for the website. To achieve the goals of this project, the backlog can be filled with user stories to help optimise predictability as the customers perspective is crucial to build an effective product that meets their needs. As the author can work closely with customers at early stage to gather user scenarios, involving the flow of events the user and system expects, the approach is useful to get valuable feedbacks from the customers at an early stage which is beneficial for the development of this prototype.

There are certain aspects that relates to the development of our system. The fact that each iteration length is known as a sprint, it usually lasts from two weeks to 1 month. Before the start of each sprint, a meeting is attended by everyone to set an objective that can be met through the implementation of product backlog. In this case, the project supervisor calls this meeting to track the backlog of work to be done as well as measuring the progress against the backlog to ensure the productivity and quality standards are met within the given time.

Additionally, a sprint backlog meeting is held at the end of each sprint to assess the project against sprint goal. During this period, product backlog can be adjusted to meet new opportunities which is crucial for this project when there is a demand of a new feature. However, the drawbacks of this framework are severe on this project, as there is not a regular client who can feedback on each iteration after a sprint and the customers involvement would only be in the early stage. Therefore, this prevents the customers to provide regular feedbacks on the developed system and the system become undeliverable to customers on regular basis for feedbacks. Ultimately, there are some aspects of Scrum that are appropriate for the development of this project which can be effective if merged with other methodology to reach a desired solution.

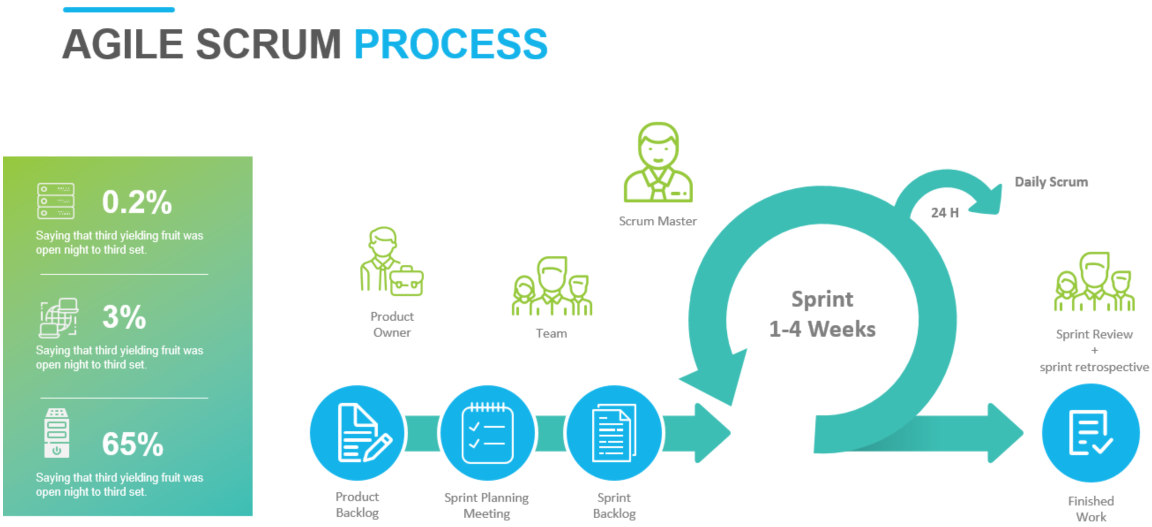


Figure 1 [https://www.google.com/url?sa=i&url=https%3A%2F%2Fpowerslides.com%2Fpowerpoint-business%2Fproject-management-templates%2Fagile-scrum-process%2F&psig=AOvVaw3d2yCbFoTtECQ7ZUApAb2e&ust=1643369597377000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCOCr_Pzq0fUCFQAAAA]

3.2.3 The Waterfall Development Model

The waterfall model belongs to traditional ways of developing a software where system is fully specifiable meaning the requirements are clear and fixed as well as built with extensive planning without any uncertainties. The reason this approach is suitable for this project is because of its simplicity and suitability to the type of problem. Waterfall methodology encourages individuals to become specialised in the assigned role, which is quite appropriate to given project as there is an author and the supervisor to work on the whole project. In contrast, modern methodology like Agile encourages role interchangeability in self-organising teams where a team of workers manage their associated responsibilities based on the given role and can adapt to other responsibilities within the team.

The development model in Waterfall follows a traditional life cycle that consists of series of stages that are followed in sequential order from requirements, design, implementation, verification to maintenance. At each stage, the management decision confirms whether the system has completed one stage and ready to move to another.

This prevents the approach from becoming flexible for our scenario, since there is limited time to complete this prototype, the process of working on each stage at a time can be time consuming. Additionally, the artefact would only become visible to end-users once the system is integrated and ready for maintenance. The process would not accept a change in the system during the development process, which could have a major impact on the end-results as the system development is already complex to implement as it involves developing, training and testing machine learning models. In result, if a change is introduced at later stage when the implementation of that change is nearly impossible, this would potentially lead the author to redesign the whole system which could be costly and time consuming.

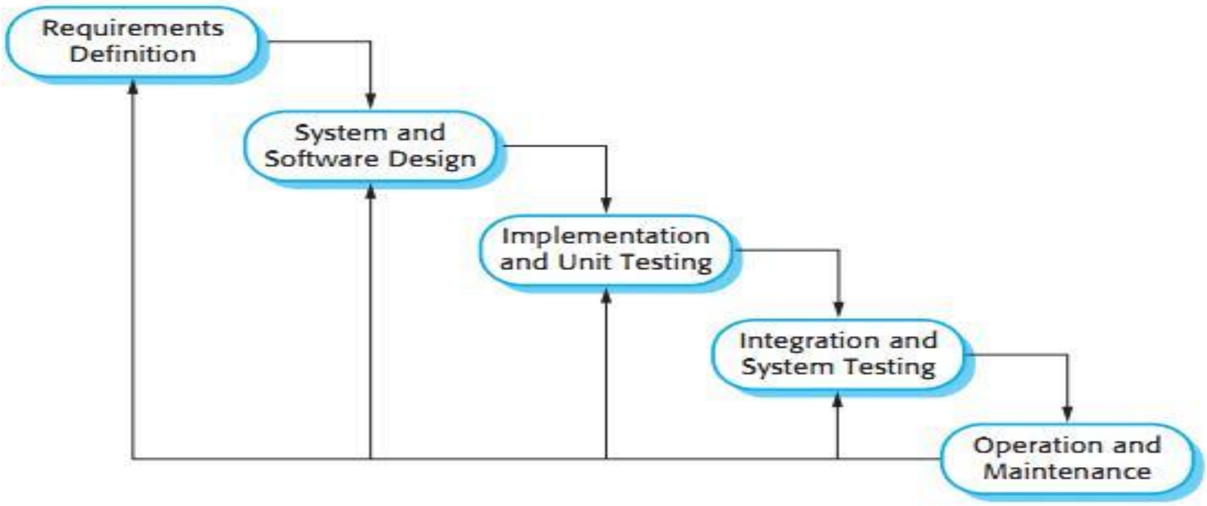


Figure 2 https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FWaterfall-Process-Model-Sommerville-2011_fig8_319403342&psig=AOvVaw2TZNl0SheiOpECBmgnnGbd&ust=1643386107849000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCNDWs7yo0vUCFQAAAAAdAA

3.2.4 Chosen methodology - Agile-Waterfall Hybrid Model (Scrum-Waterfall)

After the justification of both Scrum and Waterfall, it has become clear that a single methodology isn't suitable to solve this type of problem. Relying on a single methodology could affect the development process as the practices and methods involved may not the help this project reaches its overall objectives. All principles of Scrum are difficult to maintain as it requires the involvement of customers throughout the development process which is not possible since the project does not have any client. Though, there is some involvement of users which is at early stage of the process to gather requirements through questionnaire and user scenarios.

Afterall, the consideration of user scenario for product backlog is the only aspect of Scrum. Another useful aspect is the of sprint which is appropriate to this project. In our scenario, the sprint is held every 2-3 weeks where the project supervisor and the author hold a meeting for progress measurement against the project backlog. The effectiveness of sprint helps the project meets its short-term goals and define probable items for the next sprint.

To conclude, the combination of both Scrum and Waterfall can be appropriate to meet project's aim that cannot be achieved by using either of them on their own. Since the main issue with Waterfall was time consumption due to its lengthy stages, the life cycle can be divided between Scrum and Waterfall to speed up the whole process. The planning stages including requirement gathering, design can be done using waterfall model, considering the product backlog can act like requirement stage in Waterfall. Additionally, the design phase is followed by the requirements elicitation process which involves the completion of questionnaire and user scenarios. However, the implementation and testing stages are completed using short sprints of Scrums by organising and releasing smaller pieces of software to speed up the delivery of the artefact. In result, the project will benefit from Agile-Waterfall methodology by adhering to agile practices during development stage as well as following traditional Waterfall method to complete the first half of software life cycle.

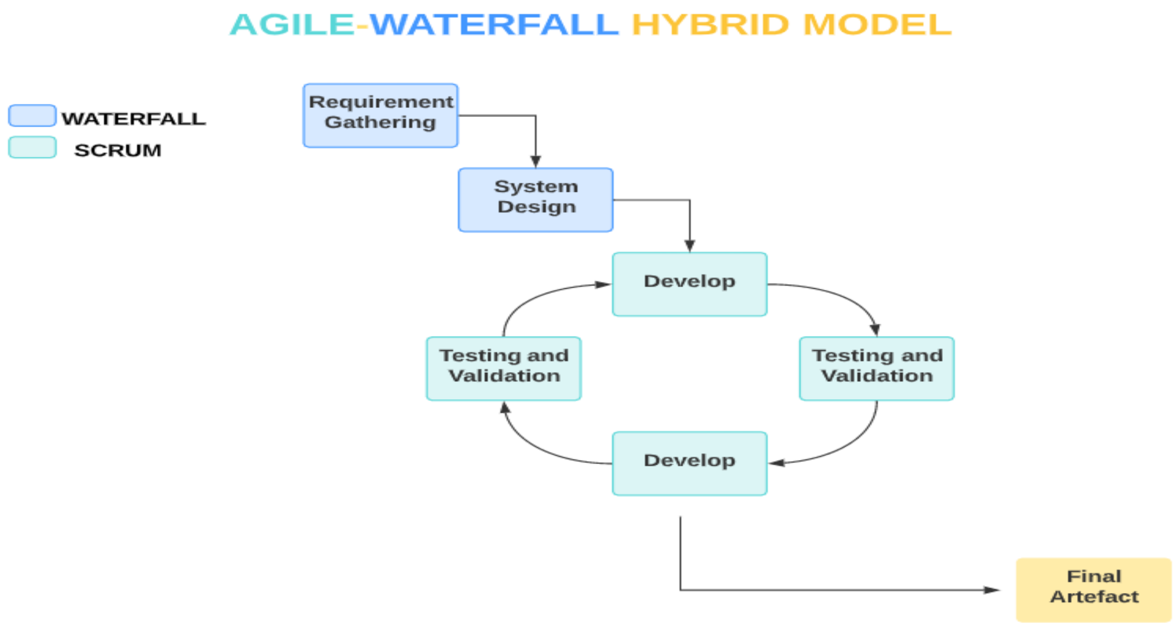


Figure 3 - Agile-Waterfall Hybrid Model highlighting the waterfall methodology for requirement and design section as well as development and testing phase using Scrum (Agile)

Chapter 4 - Requirements

4.1 Process of gathering requirements

The requirement specification plays a crucial role in the development of this artefact, the description of the services for the end-users are gathered. The constraints under which the system operates and developed are elicited through the requirement process. The development of driving behaviour website would target users who are car drivers to define the needs of the end-artefact.

Since the goal of this project is focus on four main skills of car drivers such as rapid acceleration, harsh braking, dangerous maneuverers, and over-speeding. These gathered requirements from the literature are crucial when building the artefact and should be achieved. However, to complete given requirements, additional considerations are required from the users to effectively produce an artefact that meets all user needs. For instance, if a car driver concentration is affected in longer journeys, then the website should not monitor behaviour after certain period. Therefore, some user experience needs to be considered when achieving main requirements.

A suitable requirement process would involve getting responses on a questionnaire designed for measuring the behaviour of certain users as well as to find out the type of services a user expects from this system. The use of questionnaire would allow the author to gather requirements from several users with quick and easy questions and it's an effective approach due to COVID pandemic. The aim of this approach is to gather quantitative data which can be analysed precisely as well as removing biases by considering important variables such as features. Since, the four main requirements are gathered from the literature previously, the use of questionnaire can help obtain additional information from users to effectively produce a successful artefact.

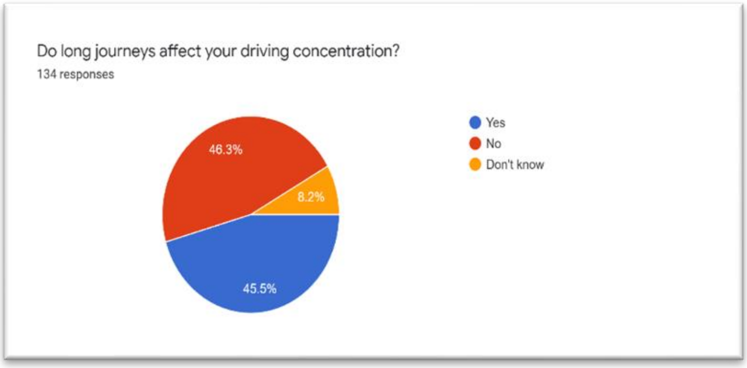
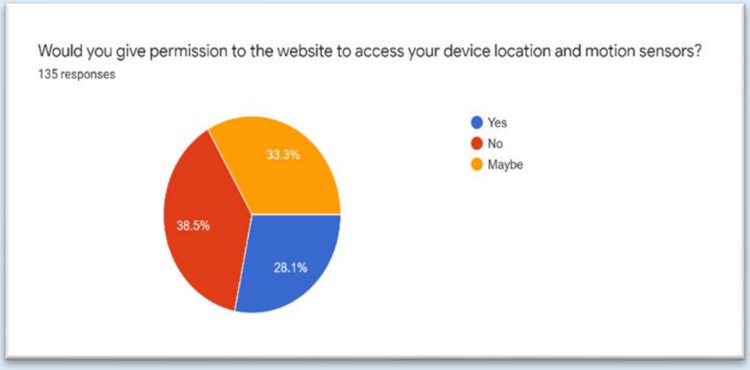
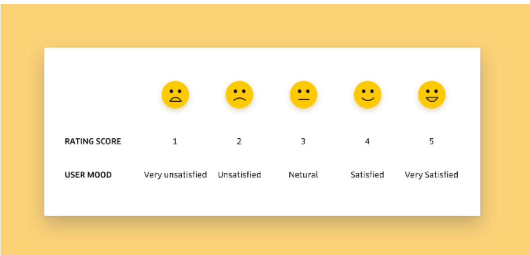
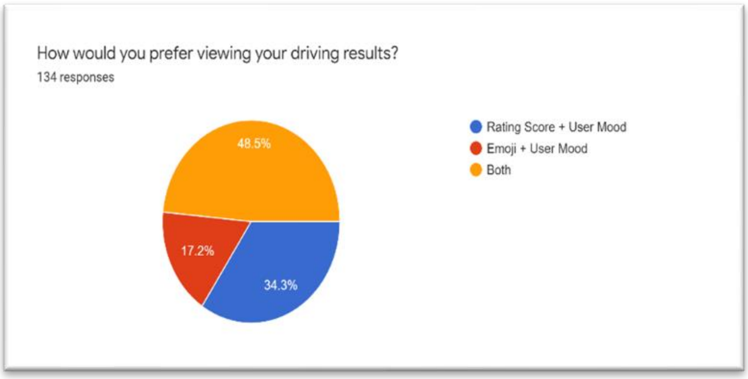
Another suitable approach would be scenario-based elicitation to capture what the users and system expects in certain situations. An appropriate place to conduct user scenarios would be a petrol station to target mostly car drivers. As the author is already working in a petrol station during the development of this project, this is a good opportunity to gather users' point of view on the flow of events that occurs when the system is running.

4.2 Analysis of the gathered user requirements

The designed questionnaire was released for the students at University of Portsmouth. The reason being is that the questionnaire would reach a good number of users to effectively analyse the results. Since, the target users were car drivers, the designed questionnaire was mostly applicable to students from all faculties which was a plus point. In result, a total of 135 responses have been collected from all users that have decided to participate in the form. This has led the author to analyse user responses as well as listing the user requirements from the process. Ultimately, the functional and non-functional requirements are designed from user requirements to ensure the usability and effectiveness of this artefact.

4.3 Requirements

4.3.1 User Requirements

List of User Requirements	Questionnaire Results
The system shall monitor four main behaviours of car drivers such as dangerous driving maneuverers, harsh braking, rapid acceleration, and over-speeding.	The followings are main user requirements which the research would implement in the prototype.
The Driving Behaviour website shall provide a feature for end-users to monitor their driving behaviour after a certain period. For instance, the users shall allocate certain time for the system to monitor their behaviour, when the timeout has been reached, the system shall stop monitoring the behaviour.	
The system shall ask users to access both their motion sensors and location permission before start measuring user's behaviour. The following requirement shall be achieved based on the user scenario below. (User Scenario 2)	
The driving results shall be displayed based on the results such as "emoji", "rating score" and "user mood". 	

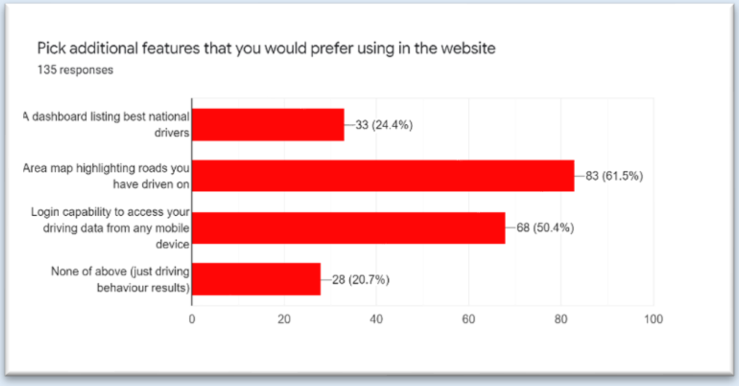
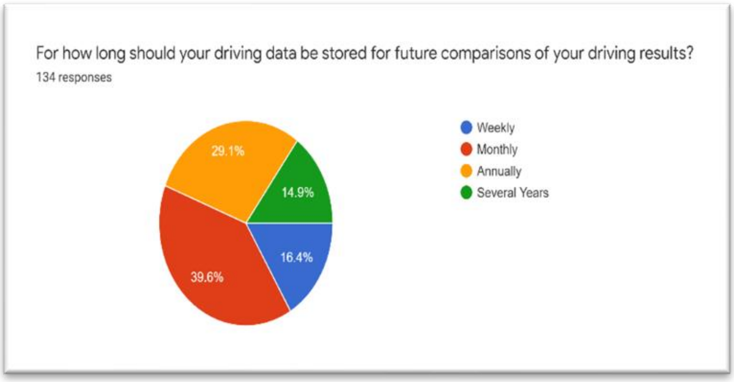
The system shall have login capability for users to access their driving results from all devices. Additional features such as a dashboard and area map shall also be considered.	
A monthly result shall be available for all users. Within given time, the system shall provide weekly driving results.	

Table 1 List of user requirements elicited from the conducted questionnaire

4.3.2 Functional Requirements

No.	System Service	Description of Functional Requirement Analysis	Validation Process	Level
1.1	Start/Stop measuring behaviour button	When clicked, the system shall ensure all permissions are granted and the sensor's data is collected while its measuring. The behaviours are based on certain sensor axis such as y-axis from gyroscope sensor to detect dangerous maneuverer and y-axis from accelerometer to detect braking events. The button shall change its content to 'STOP' to indicate that its currently measuring their behaviour. When button clicked again, the system shall stop measuring all behaviours and return to its original state.	If any of the permission is denied, the user shall not have access to measure any behaviour. The system shall alert users about the permissions that are denied and instruct users on how they can be permitted again.	High
1.2	Viewing driving results	The users shall access their driving results for both recent and previous journeys. The results data shall be available once a journey has been completed.	If previous journeys exist in the database, the system shall ensure all journeys are displayed to end-users. Otherwise, the users shall be alerted if no journeys have been made.	High

1.3	Access Sensor Permission (*User Scenario 2*)	The system shall ask user to access sensor permission before the behaviour is monitored.	When all permissions are permitted, the system will alert users when it is ready to monitor and start monitoring behaviour right after.	High
1.4	Instructions to follow before measuring. (*User Scenario 3*)	The system shall instruct user, every time a user starts to monitor, to follow instructions while the website is in use. E.g., Do not use mobile while its measuring. Do not close website page. Avoid phone calls.	The user shall agree to instructions every time they try to monitor to ensure the user has read and followed them.	High
1.5	Sign up/ Sign in (Login capability) (*User Scenario 1*)	Based on the inputs from sign up details, the system shall ensure the details match when users signs into the website. Each user using the system shall be uniquely identified by the system.	Multiple accounts with same usernames shall not be created. For instance, if username already exists, the user must be alerted straightaway.	Medium
1.6	Area map highlighting roads	The system shall display a map highlighting roads a driver has driven on. (After each journey the map shall update to display new roads)	The map shall only update after each journey. (Especially when the driver chooses to stop measuring their behaviour)	Low
1.7	A dashboard listing best national drivers' and their scores	Each user using the system shall view a list of best drivers in the country. The scores are based on each driver's behaviour in previous month journey.	After each car journey, each driver gets a score from a combination of driving events. The driver with a best score would obtain a top position in the dashboard.	Low

Table 2 - The functional requirements containing the requirement analysis plus the validation process and priority level

4.3.3 Non-Functional Requirements

No.	Title	Description of Non-Functional Requirement Analysis	Validation Process	Level
1.1	Mounting mobile device on windscreen shield.	To avoid dashboard vibrations, the mobile device of users must be mounted on windscreen shield instead of dashboard mount.	Giving instructions for users to follow before starting to measure driving behaviour. The user must agree and adhere to the instructions.	High
1.2	Monthly driving results (Storage requirement)	The driving results must be stored in the database for at least previous 30 days period for each user.	Each driving journey is stored in the database with a date of the journey. If a journey has exceeded 30 days' time, it shall be deleted.	High

1.3	Website compatibility	The website shall work on almost all devices that have both motion sensors and location access.	The system shall ensure whether the mobile device has relevant sensors to monitor driving behaviour.	High
1.4	System Availability (Reliability)	The system shall be available to end-users 99.9% of the time to ensure better performance and reliability.	Regular monitoring of the system to ensure better uptime and availability.	Medium
1. 5	User Interface Design	Regardless of complexity of the system, the website shall be easy to use, considering end-user with minimal technical skills.	Avoid unnecessary elements/features on the screen for users to feel more comfortable when using the system.	Medium

Table 3 The non-functional requirements each with priority and validation process.

4.3.4 User Scenario 1 – Login capability

TASK TO BE COMPLETED:
The driver has decided to use the website to measure driving behaviour. The system requires the user to create an account or log in before giving access to any other features.

FLOW OF EVENTS:
A user expects to create an account by entering their full name, date of birth, email, and password.

Once account has been created, a user logs into the system using their credentials. In other case, if an account already exists, the user will attempt to login straightaway.

When successfully logged in, the user will be redirected to website's dashboard page.

WHAT CAN GO WRONG AND HOW IT IS HANDLED:
A user with similar email address may already exist, therefore the user needs to enter a different email. The password standards are not met meaning the entered input may not fulfil the criteria of password letter or digits.

In a similar scenario, a user may not already have an existing email address, so they should either create a new email account or sign up with a unique username.

The user may leave some required fields blank which will prevent the user to create an account. To handle this situation, blank fields are marked as "*required" and highlighted to show an indication of missing fields.

OTHER ACTIVITIES HAPPENING AT THE SAME TIME:
While user is entering email address, the system may inform the user if the email address is already in use. This will prevent user from entering details repeatedly.

SYSTEM STATE AFTER COMPLETION:
User has logged into the system. The details are stored in the system's database. The user is given access to rest of the features.

4.3.5 User Scenario 2 – Access sensors and location

TASK TO BE COMPLETED:
The driver has logged into the system. The system requires the user to give access to location and motion sensors to start monitoring their behaviour.

FLOW OF EVENTS:
A user expects an alert prompt to provide a list of accesses the system requires.

User can either hide the prompt to change the settings themselves.
Or the system can redirect the user to device's settings page which only requires the user to turn on the access.

The user returns to the website.

WHAT CAN GO WRONG AND HOW IT IS HANDLED:
User may fail to locate the right settings which the website requires. In this case, a user returns to the website to use "redirect to settings" option.

In another scenario, a user may decide to turn on location settings without giving access to motion sensors. In that case, the user can see list of accesses in which location will marked as "yes" and motion sensors with "no". The user will expect another prompt message when the website has been opened again to give multiple opportunity to provide access, bearing in mind, the user may forget to follow these events.

Since modern mobile devices have different types of location access such as while using the app, always allow access to location. The website expects the user to give "always access" to prevent users from repeatedly giving access as the user will return to the website more frequently in other occasions.

OTHER ACTIVITIES HAPPENING AT THE SAME TIME:
The user may decide not to monitor driving behaviour, instead, they may prefer checking out their previous driving results. In return, a user can access the website to view their results, however, they can't gain access to monitor their driving behaviour for this moment.

SYSTEM STATE AFTER TASK COMPLETION:
The system get access to both location and motion sensors. The user is redirected to the main dashboard of the site to further start monitoring their driving behaviour or view results of previous journeys.

4.3.6 User Scenario 3 – Providing Instructions

TASK TO BE COMPLETED:
A user is travelling to work in the morning and has decided to use the website to measure their driving behaviour.

FLOW OF EVENTS:
User visits the website, logs in and the access to location and sensors is already provided. The system reminds the user of some instructions to follow while it is monitoring the behaviour.
Instructions:

- Ensure the mobile device is attached to vehicle's windscreen.
- Do not close the website while it is monitoring.
- Avoid using your phone.
- Stop monitoring when reached your destination.

Once the user has followed instructions and is satisfied, he starts measuring his driving behaviour.
The user has reached his destination therefore stops the system from monitoring.

WHAT CAN GO WRONG AND HOW IT IS HANDLED:
A user may place the device somewhere else in the car, leading the website to record data from wrong axes of the sensors. Also, if the website page has been closed by the user, the system is not able to run even in background. In both scenarios, the system cannot measure any data from that journey. Therefore, a user is shown an error stating the instructions that are not followed and is asked to monitor their journey again.

Phone calls on the device can cause the device to be moved from windshield mount, leading the device measure inaccurate data. For prevention, the user needs to stop monitoring and restart again. However, they will be notified for the driving period that has not been considered in measuring their driving results.

OTHER ACTIVITIES HAPPENING AT THE SAME TIME:
The system can start using the collected data from the sensors to categorise each driving event as normal or dangerous using sliding window technique.

SYSTEM STATE AFTER TASK COMPLETION:
User is returned to website dashboard. The results from this journey will undergo ML model which can take a while depending on the duration of the journey. The user is asked to wait and check their results later.

4.4 Summary

The requirement elicitation process has been successful in both ways, the use of questionnaire has helped achieve some user experience and features for the website itself. Based on the results, the author was able to distinguish both functional and non-functional requirements. Followed by user-scenarios that have been crucial to get face-to-face interaction with the users to gather real-life scenarios of how the system is going to be used by the end-users. Though, the list of scenarios has not been gathered from considerable users as it could have resulted in the process becoming time consuming which is a constraint on this project. However, the focus was to gather real perspective of users to effectively produce the end-artefact which eventually meets user needs. Ultimately, both approaches have helped obtain users perspectives on main features of the website that have higher priority and are used by majority of the users.

Chapter 5 – Design

The design document is important to illustrate key system components represented using various design methods as well as user interactions with the system itself to provide an overview of how the system would behave in certain situations. The decisions made to choose key design concepts are derived from requirement section where user scenarios are gathered. In addition, a system architecture is produced with use case and sequence diagrams to provide a visual understanding of how system components interact with each other and how users' expectations are met in certain scenarios.

5.1 Architecture Model

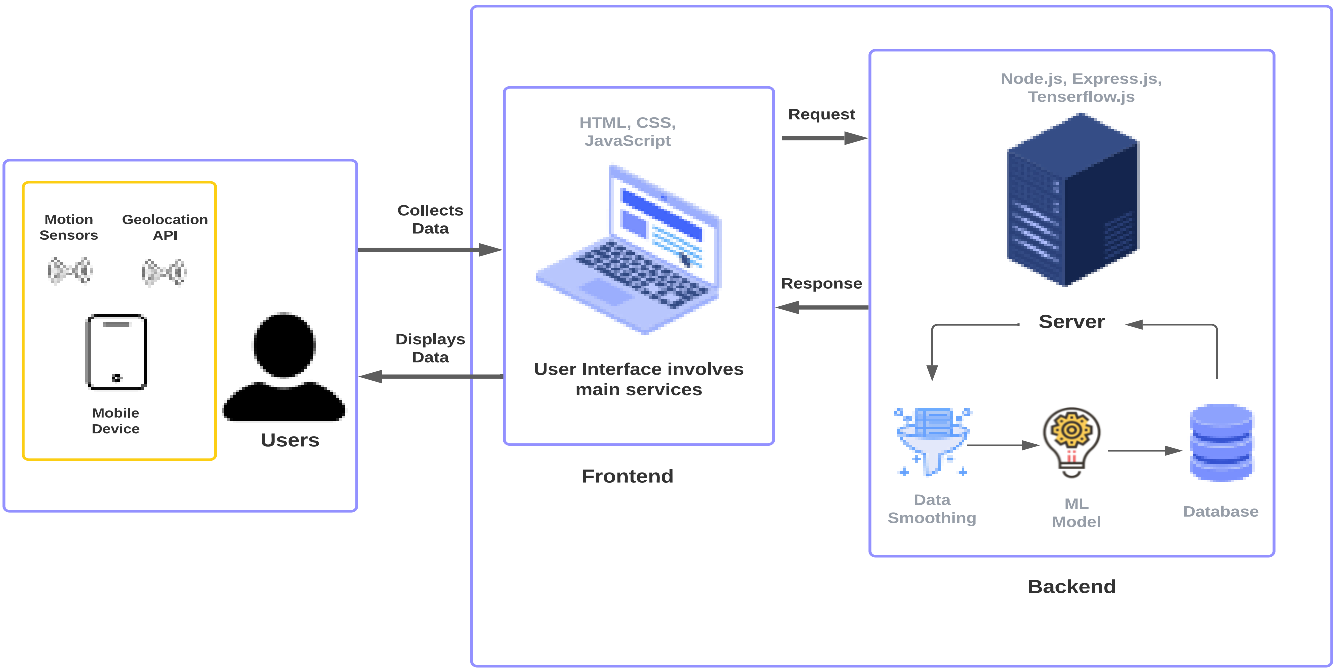


Figure 4 - High-level architecture model highlighting key system components. The data collected from the users passes through data smoothing and trained ML model that detects potential driving events which then get stored into the database.

For the nature of the problem, an architecture model represents a better understanding of system components. Key components are highlighted which the system consists of such as mobile device with sensors, user interface, data smoothing module, machine learning and the database for storing potential events. The communication between components is represented with arrows to indicate the flow of data from one component to another. For the system to run efficiently, these components are event based due to the nature of the programming language chosen. When the user chooses to start monitoring their driving behaviour, a motion sensing event is fired to collect data from user's device.

At runtime, only frontend and users' components are in use, involving the process of gathering users' mobile sensors data. The responsibility of frontend system also includes posting sensory data to the server when the user stops monitoring their behaviour. Dealing with low level computation on the client-side which involves gathering and posting data prevents the load on user's browsers, which could result the website becoming slower and inefficient. In addition, the non-functional requirements prioritised system availability to be 99.9% therefore, high level computation such as data smoothing on time series sensory data can be programmed on the server-side. The fact that the server can be scaled during peak time, this provides the benefit of reducing overloading on the browser side which can have impact on the system, especially when the drivers choose to monitor their behaviour for longer period.

Considering these counter points, the backend is designed in similar way to deal with high level computations. Node.js, a JavaScript runtime environment, provides efficient performance and easier development process with the help of node package manager (NPM) packages such as Tenserflow.js library developed by Google, used in developing machine learning models and Data Smoothing module to remove the noise from time series data. Furthermore, database management system is also part of this library for storing large datasets. However, the discussion and choices of these modules will be discussed below in the implementation section.

5.2 Use Case Diagram

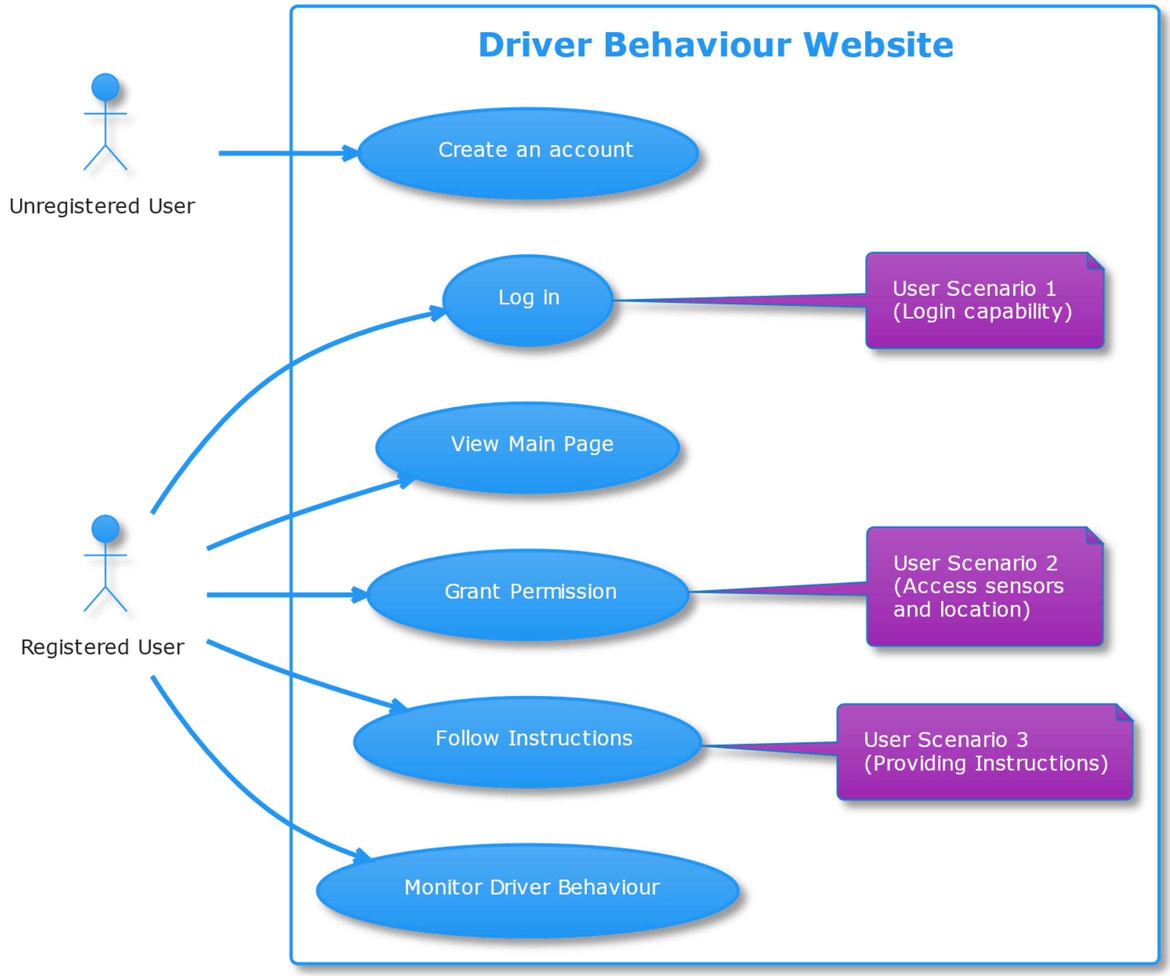


Figure 5 - A use case diagram representing high level features and the flow of events a user expects from the system. The description of important tasks is referenced with the user scenarios discussed above in the requirement section.

Another example of design methods is a use case diagram. The fact that the requirement gathering involved user scenarios to figure out the flow of events a user expects. The consideration of use case diagrams provides a high-level visual overview of how the user interacts with the system and its environments. Above representation highlights how an unregistered user has only access to one feature before it performs any other activities on the website. A user expectation from the system is limited due to the fact that they must create an account before monitoring their driving behaviour. On the other hand, a use case diagram is used to represent features a registered user has access to such as viewing main page and monitoring driving behaviour. Since, a use case is developed with both the diagram and description table, important features such as logging in, grant permission and follow instructions are derived from user scenarios. The description for each task is provided in the user scenarios that considered the flow of events and error handling during unexpected situations.

5.3 Sequence diagrams

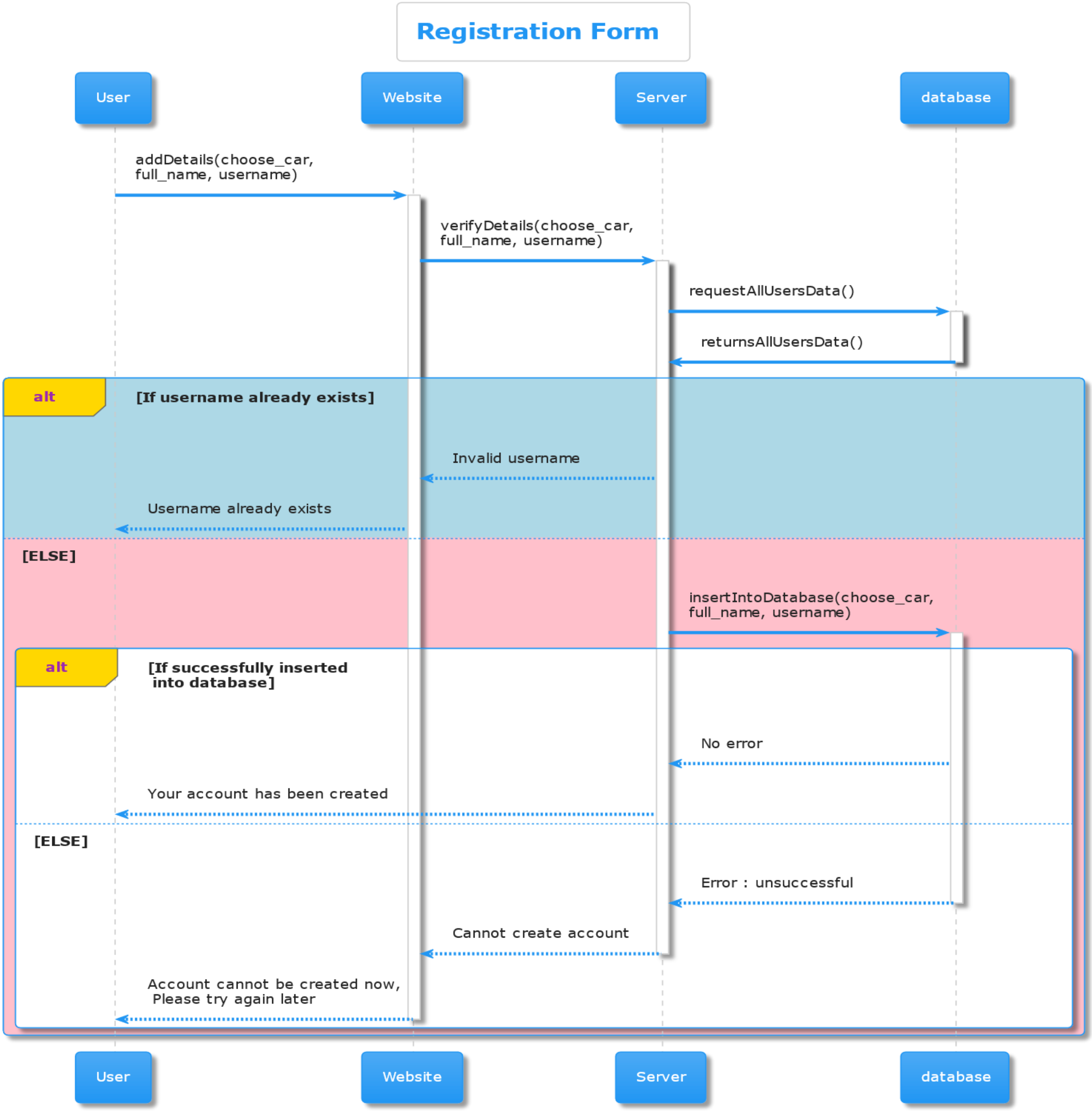


Figure 6 - A sequence diagram representing the registration form, a sequence of events that occurs when user is trying to sign up. The error cases are also highlighted when certain error occur.

Due to the complexity behind registration form, above sequence diagram represents sequence of interactions between system components during registration use case. The flow of messages from one component to another is drawn with straight arrows and the return messages in dashed arrows. The fact that functions between components are called in a sequential order, the choice of sequence diagrams gives a high-level overview of the system to the developers and other stakeholders. In this case, the registration form requires complex decision made by developer to ensure the users does not exist with accounts with same user id. The use of sequence diagram gives the ease to understand the concept behind programming a registration use case. In addition, the invalid cases are also encountered with better error handling such as invalid return messages if the username does not meet certain type characters or digits.

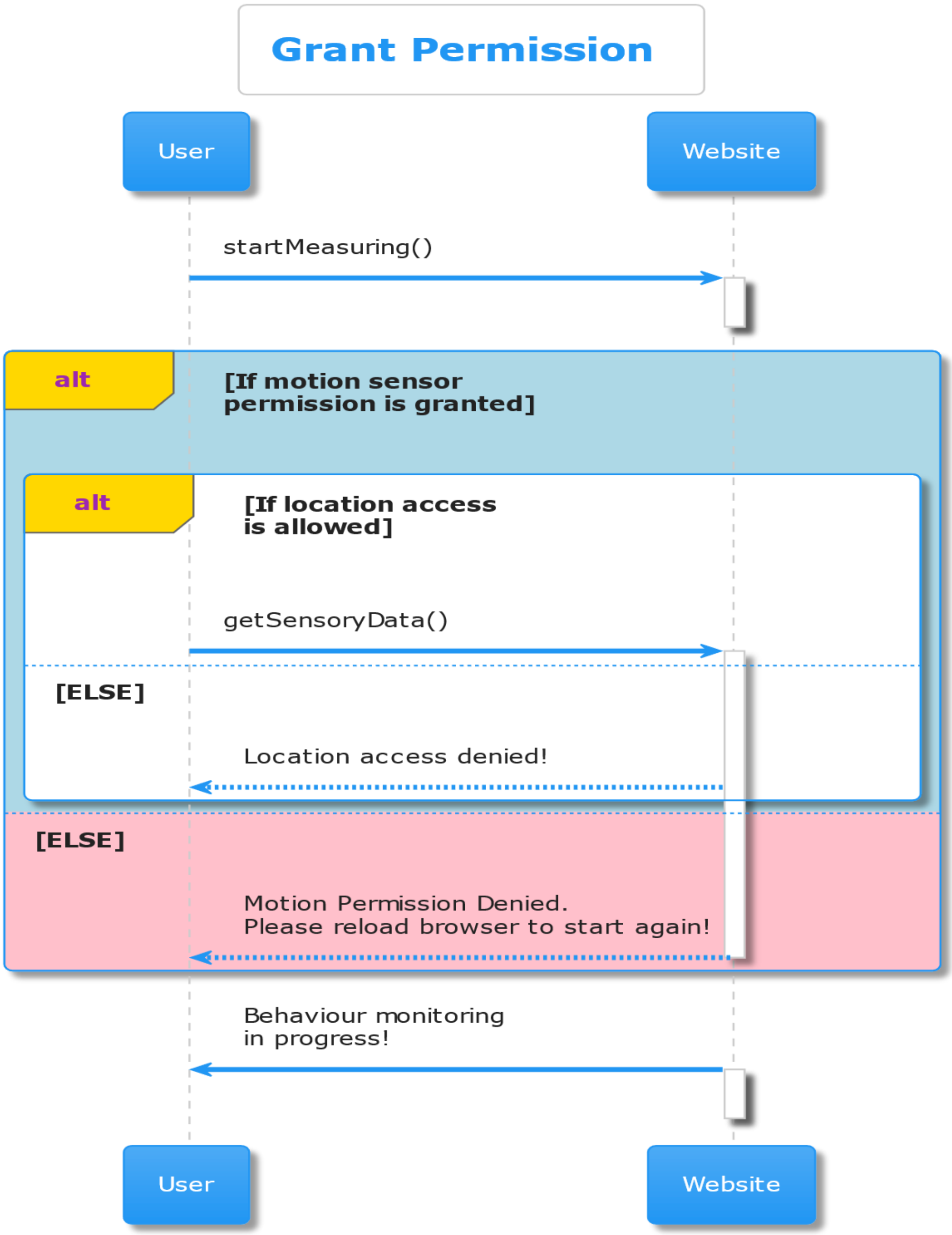


Figure 7 - Granting permission before monitoring driving behaviour, both motion and GPS access is required. If any access is denied, the user will need to follow the process before going forward.

Another important use case for the system is granting permission to monitor behaviour. The sequence of events is not complex to understand. However, it involves two main components of the system, both the user and website to exchange messages. The first message requests motion sensor access which is crucial to get access to sensory data followed by requesting location access. If either is denied, the user will not be able to monitor their behaviour since both are required to fulfil the requirements of measuring 4 main behaviours. From developers' perspective, the sequence diagram highlights the importance of meeting and

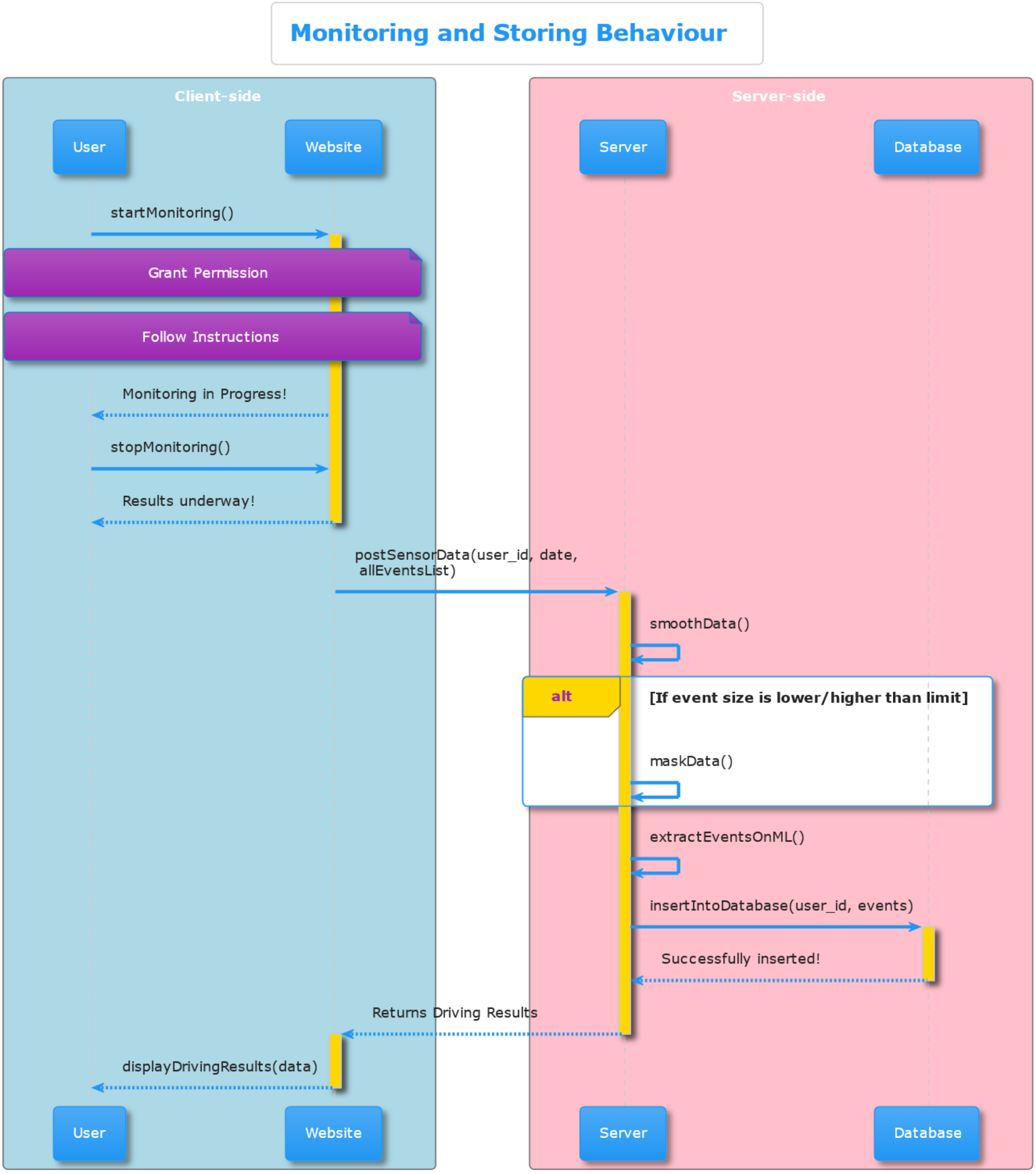


Figure 8 - A sequence of events occurs when the user decides to monitor behaviour. Starting from granting permission to storing behaviours in the database.

understanding highly prioritized requirement to achieve the overall goal of this project.

The architecture model illustrates a high-level overview of how the communication occur between system components from the users to system environment. The above sequence diagram provides a lower-level overview of system components including the sequence of functions from client-side when a user decides to monitor behaviour. Followed by server-side functionality to extract potential events from the journey, which ultimately get stored in the database. The step-by-step sequence of events also describes a detailed functionality which is helpful during the implementation phase to program functions. It sets the tone for developers to expect certain return values from functions. For instance, "maskData()" function must return masked data, meaning, the length of all data should not exceed a defined value. Ultimately, after some time the user can view their driving results once it has been detected by the trained machine learning model.

In summary, all design methods have turned out to be effective to get a visual overview of system requirements. The use of architecture model has indicated key system components categorised to provide a high-level overview of the user interactions and the communication between components themselves. However, an architecture model was not enough to provide interactions of different types of users within the system. Use case diagrams were useful to represent the user expectations once they have completed a given task and provide an overview of user scenarios gathered in the requirements. However, the choice of sequence diagrams was crucial for lower-level overview of system components, the sequence of functions called at each stage are useful to make early implementation decision. In results, these key design decisions play a key role in developing an artefact effectively with better functionality and design.

Chapter 6 - Implementation

6.1 Choosing implementation environments (JavaScript & Node.js)

A very initial stage of developing a software involves the decision of choosing the right software tools required to build the end artefact that meets the elicited requirements and is responsive on all web platforms. The fact that this prototype is a web application, JavaScript programming language is suitable and brings the dynamic interactivity to the webpage, allowing developers to easily meet their requirements by running scrips on both client and server sides. The choice of building a web application rather than a mobile app is made since a mobile app cannot be directly accessed from the internet. Rather, it must be downloaded from the provided platforms of different mobile devices such as Google play store on Android or app store on IOS devices. Further, a website can be developed responsively that can automatically scale based on the screen size of user's device. Ultimately, the scalability of the prototype gives more freedom to the author and users to access from multiple platforms. In fact, for given scenario, the users can monitor their driving behaviour from the mobile devices and view their journey results from laptops or desktop computers.

There are several other benefits of using JavaScript for the development of a web application. In client-side, the event handlers on HTML elements lets you execute code when an event is fired. The window object in the browser is useful for accessing mobile sensors data whether for Android or IOS. The device motion event gets fired on regular intervals that detects the amount of physical force of acceleration and the rate of rotation a device is receiving. The combination of several events is what makes up this artefact useful and successful in many ways. Further, these benefits of using events help users to monitor their driving behaviour and perform other activities on the website. However, it would not be possible to develop such scalable and high-performance website by using other programming languages. The fact that every programming language has its own purpose therefore JavaScript is perfectly suitable and has been mainly used for the web development for years.

In fact, JavaScript on its own and in the client-side does not provide much benefit to the author as the data collected from the sensors would not be useful if it has not been used for any other purpose. To come across such problem, Node.js an open-source, cross-platform, JavaScript runtime environment for server-side can be handy to move data across to the server for further pre-processing. The node environment can be set up quite easily by downloading Node.js in the system and installing relevant dependencies. Node.js built-in module runs Hyper Text Transfer Protocol (HTTP) server that listens to server ports and sends response across to the client. HTTP on its own has solved the problem of sending and receiving data from the client. However, when running the application locally on the mobile device it does not secures the connection between the user and server. Due to this problem, the IOS devices prevents the application from requesting sensors permission which is absolute necessity for the development of this artefact. In result, a separate "https" node module has been installed along with Express (in figure below) to protect the integrity and confidentiality of the data, by running a secure and private connection for the user. The decision of using HTTPS would not only secure a strong connection but it also allows this prototype to be available across all mobile devices that requires the system to be secured when giving access to their built-in sensors.

svr.js Settings

```
import express from 'express';
import https from 'https';

const app = express();
const port = 8080;

const httpsOptions = {
  key: fs.readFileSync('key/key.pem'),
  cert: fs.readFileSync('key/cert.pem')
};

const server = https.createServer(httpsOptions, app)
app.use(express.static('files'));
app.use(bodyParser.json({limit: '50mb'}));
server.listen(port, () => {
  console.log('server running at ' + port)
})
```

Figure 9 Node.js Express set up with HTTPS, listening on PORT 8080

To install relevant dependencies, Node Package Manager (NPM) is a great built-in command line tool that install modules into a folder which later can be imported into the scripts. In the above example, we have installed express and https using the NPM and imported these modules into the server file. The “httpsOptions” variable reads the private key and certificate from given folders which is required to create a secure connection to run the server. Ultimately, we have assigned Express to use “files” folder which contains all the client-side files (*HTML, CSS, JS*) that creates the user interface.

6.2 Data collection and feature extraction

As mentioned in the literature, data collection and feature extraction are the most crucial stages before training a machine learning model. The right data collection is what indicates the success of the end results, feature extraction goes hand in hand, the right data needs to be extracted for purpose of training the model. In the given scenario, the decision of using pre collected was made because the available data is high-quality and was collected using same characteristics to what this study is aiming for. To train the model, the dataset from the GitHub repository is used which has a collection of smartphone sensors data of different driving events. The experiment of 4 car tips was performed with the mobile device attached to vehicle’s windshield. This makes the dataset appropriate for training on the LSTM model as it has followed same instructions to what we have found from the literature (keeping mobile attached to vehicle’s windshield instead of dashboard).

Table 4: Driving events performed with associated number of samples

Driving Event	Number of samples
Aggressive braking	12
Aggressive left turn	11

Aggressive right turn	11
Non-aggressive events	12
Total	46

With the given dataset, feature extraction is one of the most time-consuming stages where the right data needs to be extracted. Since each driver performed several events for approximately 13 minutes journey, the data was collected from multiple sensors such as accelerometer, linear acceleration, magnetometer, and gyroscope. Each sensor had its own x, y, z axis as well as timestamp to indicate the time when certain events were performed during the experiment. The collected data had a lot of complexity since each dataset file contained multiple events of sensory data with thousands of lines.

To solve the problem of extracting events, a small algorithm was developed using JavaScript to extract the right events from each file with the right axis. The fact that we are only interested in using a single axis from a given sensor therefore the algorithm would avoid collecting the other 2 axis as mentioned in the literature. In return, the size of the dataset reduces, and the right feature are extracted which are only required to train the machine learning model.

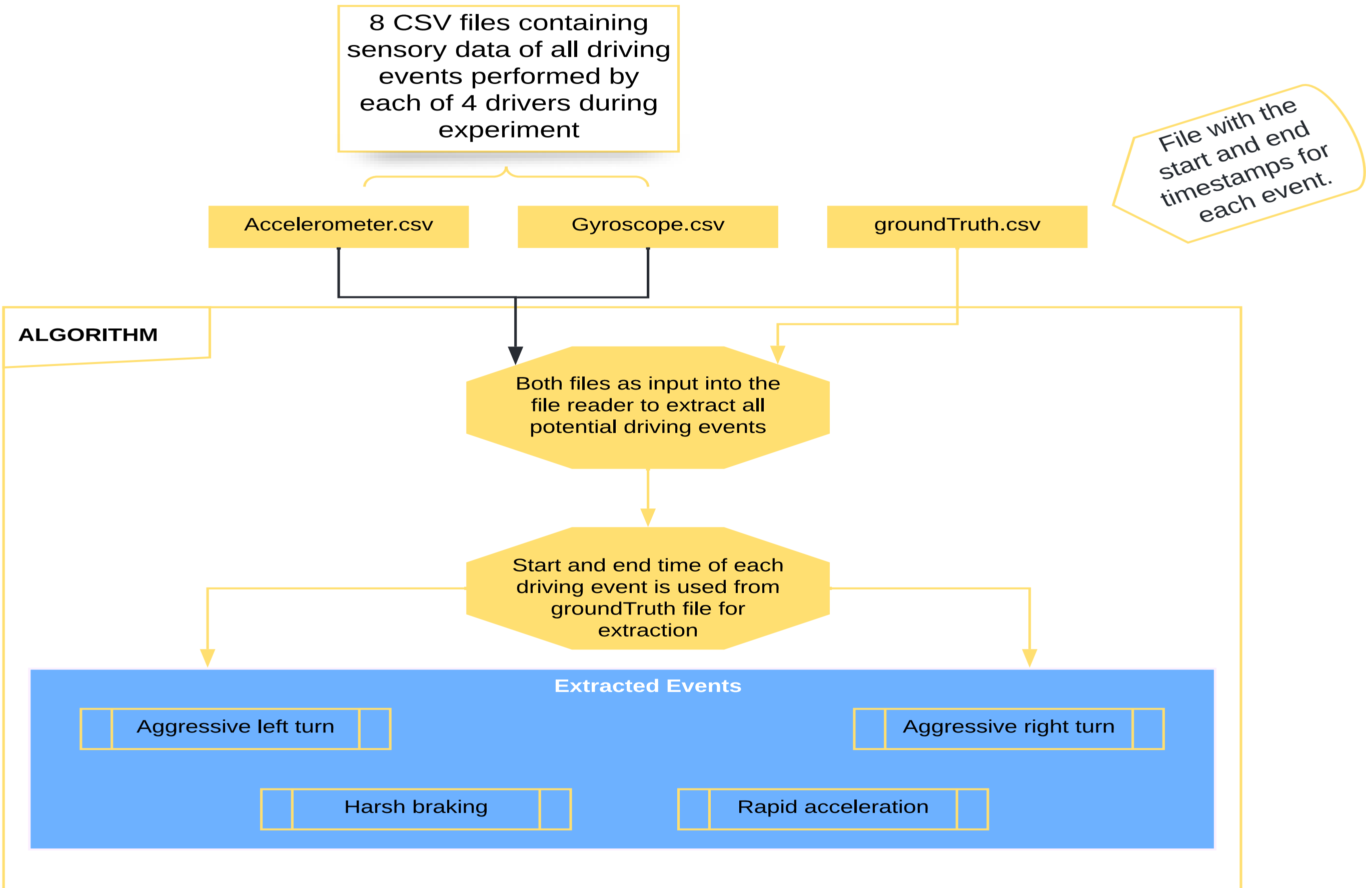


Figure 10 - Extracting all events from CSV files.

After running the developed algorithm on each driver's file, we ended up with a several files containing all driving events with associated labels. At this stage, the author decided to choose 50% of samples for training the model and the rest 50% to efficiently test the model once it has finished training. Since there were enough samples for training, a high number of testing samples allows the model to show its accuracy on unseen data. However, the percentages can be adjusted later when we have finished training the model. If the model accuracy falls par below therefore, more data can be fed into the model to increase its accuracy, so the model had seen all possible events which in results leads to classifying testing data more precisely.

6.3 Training and Testing Long Short-Term Memory model using Tenserflow.js

6.3.1 Importing relevant modules

When developing the prototype using Node.js, the accessibility to a range of NPM modules gives a lot of freedom to train any machine learning model which we have found appropriate in the literature. The literature highlighted LSTM to be suitable model when training time series data. In this case, we have access to the model using Tenserflow.js, a free and open-source library for machine learning and artificial intelligence. Since the system architecture model showed that high computation such as training/testing ML model needs to be performed on the server-side to increase system's availability and performance, Tenseflow.js has become a perfect choice to train the LSTM model on server side. The figure below imports Tenserflow.js using a variable along training and testing datasets imported from separate files which later is used for training of the network.

```
const tf = require('@tensorflow/tfjs-node');
let training = require('./dataset.js');
let testing = require('./testingData.js');
const smooth = require('array-smooth');

let trainingData = training.dataset;
let testingData = testing.testingData;
```

Figure 11 - Importing Tenserflow.js, training and testing datases alongside array-smooth to smoothen the dataset.

6.3.2 Training dataset smoothing to remove noisy data

During the training process, we import the relevant modules as well as importing collected training and testing datasets from the other 2 files. Consequently, array-smooth is another type of module to smoothen the datasets as shown in figure below. The fact that, the sensory data may include noise which needs to be smoothened, the array-smooth module is utilised for this purpose to remove the noise while allowing the patterns of different driving events to stand out clearly. This technique is useful for the model to identify pattens/events that helps increase its ability to accurately identify different driving events.

```
function smoothDataSet(DATASETS) {
  const ALLDATASETS = []
  for(const SAMPLE of DATASETS) {
    // WINDOW SIZE OF 4 WHICH SMOOTHS THE DATA
    ALLDATASETS.push([smooth(SAMPLE[0], 4),SAMPLE[1]]);
  }
  return ALLDATASETS
}
```

Figure 12 - The following function smoothens a list of training samples. Smoothing constant value is kept at 4.

6.3.3 Masking dataset

Another important consideration during data processing is dealing with event samples frequencies in variable length, leading to samples' length to decrease into smaller chunks. This issue was discussed in the literature. However, the solution is derived using TensorFlow documentation which requires the datasets to be masked by adding unknown values to samples with variable lengths. The approach tells the sequence-processing layers to skip those values in the sample while training. To solve the problem, we iterate through all samples in the training data and choose a sample with biggest length. Followed by adding unknown value to each sample that has a smaller length than the biggest sample. In results, we end up with all samples with same length which is a requirement to train the model.

So far, we have smoothened each sample by removing the noise and masked to have all samples with the same length. This is how the training dataset looks at this stage in the image below.

```
const dataset = [
                    [[0.4, -0.02, 0.13, 0.15 ... 232 more items], 1]
                    [[0.2, -0.02, 0.13 ... 233 more items], 2]
                    ... 24 more samples ]
```

Figure 13 - Current state of training samples at this stage

The dataset variable has a list of all samples in which each sample is an array itself which contains the first index an array of values of a single event and the second index is a label for a particular driving event. In fact, when working with Tenserflow.js, the dataset variable would be turned into tensors. The tensors can represent multidimensional arrays by generalising vectors and matrices into higher dimension. This makes tensors stand out from arrays and the new representation of the dataset is given below.

```
Tensor {
  kept: false,
  isDisposedInternal: false,
  shape: [ 24, 236, 1 ],
  dtype: 'float32',
  size: 5664,
  strides: [ 236, 1 ],
  dataId: {},
  id: 0,
  rankType: '3'
}
```

Figure 14 - A tensor representing 24 training samples with a length of 236 after smoothening and masking.

During the training process, the formed tensors can be fed in the LSTM model that can identify potential events. Additionally, the same process is followed when testing the model which requires the coming testing data to be turned into tensors. However, the testing data does not need any labels as the classification is performed by the network itself.

6.3.4 One Hot Encoding of training labels

In a similar way, the label vector of the training dataset forms one dimensional tensor. However, label tensor is not fed straight into the model as it does not improve classification accuracy when given in decimal numbers. To solve such problem, TensorFlow has the process of One Hot Encoding that converts the number of unique values in the original label vector into new categorical binary vector that represents each driving event. Following this principle, the model can predict results in probabilities which are useful to identify the level of aggression of single driving event. Ultimately, the aggression rate of each event would indicate the driving score to inform users about the likelihood of a particular driving event.

Table 5: The conversion of original training labels into one-hot-encoding.

Driving Event	Original label	One Hot Encoding label
Aggressive left turn	0	[1, 0, 0, 0]
Aggressive right turn	1	[0, 1, 0, 0]
Non-aggressive events	2	[0, 0, 1, 0]
Aggressive braking	3	[0, 0, 0, 1]

6.3.5 Setting up, defining the topology of LSTM model and training

Defining the topology of the network is one of the crucial stages. During the training course, the network was trained using various topologies in which we considered using multiple layers. During first training phase, the model was trained using a hidden dense layer that acted as an input layer with sigmoid activation function, LSTM layer as well as output layer with SoftMax function. Afterall, the topology resulted in model accuracy of 68% when experimented on 50% of testing data. Even though, the first training phase was completed with various number of epochs (10k, 15k and 20k epochs). However, the accuracy of the model remained below par which was a concern because the network would not classify events precisely when unseen data is presented. In result, this would have major impact on user experience when they monitor their driving behaviour. In some scenario, if a user driving journey only involves non-aggressive events, the model may misclassify those events as harsh braking or so.

To improve its accuracy, some major changes took place. The first major problem was the length of single sensor axis which was more than 15 digits. The training dataset involved 24 samples with each sample length of approx. 236. As indicated in the tensor above, the size of training data included 5664 values with each value length above 15 digits. The higher the value the more the variations in the values in each sample, therefore the model is enforced to deal with much bigger data when assigning the weight vector to the input. In result, the size was capped at 2 decimal place which helped the network to correctly identify each sample quickly and precisely.

```
const model = tf.sequential();

const lstm = tf.layers.lstm({
  units: 48,
  activation: 'sigmoid',
  batchInputShape: [24,236,1]
})
const output = tf.layers.dense({
  units: 4,
  activation: 'softmax'
})
```

Figure 15 - LSTM network topology with 2 layers. Input layer: number of input neurons being 48 with an activation function as well as input shape of training samples. Alongside output layer with softmax activation function that produces 4 outputs for a single classification.

The second adjustment was the change in the topology of the model. In this instance, the topology involved the LSTM layer that inputs the values of a sample, performs sigmoid activation functions that produces a real value of the output when it has got the weighted sum of all the inputs and weights. Ultimately, the output layer has SoftMax activation function to produce the probabilities in the output labels. The combination of layers has produced an accuracy of 82% which has significantly improved after making these major changes. Even though, the model trained on 10k epochs which are quite common when working with large datasets. Afterall, the training loss in initial epochs was above 1.40, however, as the network learnt to classify different driving events, the loss has reduced considerably below 0.1. This indication shows how well the model has learnt during its training and its ability to classify unseen data. Ultimately, to prevent the training of the network repeatedly, the model was saved into a model folder which can be used for prediction.

```
-----------------------------
Epoch 10000 / 10000
eta=0.0 ==================================================>
1405ms 66905us/step - loss=0.104 val_loss=0.0442
Epoch: 10000
Training loss: 0.10377491265535355
-----------------------------
```

Figure 16 - The ultimate values of loss and validation functions that are significantly decreased once the model has trained on 10,000 epochs.

6.3.6 Testing the model

The testing phase involved similar approaches to the training phase where we import testing data from the file, smoothen the dataset as well as masking any sample with variables length. It is vital for the network to see data in a similar way as it prevents overfitting when the model fails to classify certain events. As discussed above, we have kept 50% testing dataset to test its accuracy on unseen data. In result, the model was able to classify 18 out of 22 samples given its accuracy of 82%.

Ultimately, the end results highlight the success of LSTM model as it has identified majority of the testing samples. The other 4 misclassified samples were part of each driving events therefore more likely the data available may be inadequate which has led the model to misclassify these events. However, the overall precision rate could have been improved by increasing the number of samples which may have had a significant impact on the results. Unfortunately, lack of samples is one of the drawbacks of using pre collected data. Event though, the dataset involved good qualities of collected events, but it still lacked the number of samples which needs consideration when training the model again in the future.

6.4 Measuring other driving events (Over-speeding and Acceleration)

6.4.1 Detecting vehicle over-speeding

Over-speeding is one of the 4 four main requirements that the artefact would monitor. The process of measuring this event has involved the use of Global Positioning System that can detect the GPS coordinates of a user's device. By accessing the location services of any device allow the author to locate users position on different intervals to measure their speed between 2 GPS coordinates.

However, the decision of using GPS would only be useful for locating users. As a result, there was additional needs of obtaining the distance travelled between 2 coordinates and finding a road information such as road name/speed to compare the road speed limit with users' speed. This need has led the implementation of Haversine formula that determines the distance between 2 coordinate points given their latitude and longitude on a sphere. The formula has proved to be accurate during an experiment performed by (Rendy,2018) in which the results have only differentiated by a margin of 0.2 metre when compared with manual distance computation. In result, this makes Haversine formula a perfect choice to implement in this artefact since it can be installed using NPM packages and the GPS coordinates are ease to access. The distance travelled over time would also help obtain vehicles speed.

Followed by using OpenStreetMap API to find road speed which could be used for comparison. To make API calls, node-fetch module is another useful module that is part of node package manager to retrieve data from an API.

```
import fs from 'fs';
import fetch from 'node-fetch';
import haversine from 'haversine';
```

Figure 17 - Importing node-fetch for making API calls to fetch road information alongside haversine module to measure the distance between 2 GPS coordinates points. The fs module is used to read saved road information from a JSON file.

As found in the literature, the rate of collecting sensory data should be higher therefore the GPS coordinates were collected at a rate of 60hz. As a result, the data consisted of 60 GPS samples per second. The initial stage involves the method of finding speed intervals when the sensory data has reached the server.

```
1  // FIRST PARAM = ALL COLLECTED GPS COORDINATES DURING JOURNEY
2  // TIME BETWEEN SPEED CHECKING = FOR EX. CHECK SPEED EVERY 4 SECONDS
3  // THIRD PARAM = MIN SPEED TO TARGET = IF DRIVER EXCEEDS 30mph THEN WE CHECK ROAD SPEED LIMIT
4  function findSpeedIntervals(coords, secs, minSpeed) {
5    const exceededIntervals = [];
6    if(coords.length > 120) {
7      for (let i = 0; i < coords.length; i=i+secs*60) {
8          if(coords[i+1]) {
9            let [start, end] = [{latitude: coords[i].lat, longitude: coords[i].lon}, {latitude: coords[i+1].lat, longitude: coords[i+1].lon}];
10           let calculatedSpeed =  haversine(start, end, {unit: 'mile'}) * (3600/secs);
11           if(calculatedSpeed > minSpeed) {
12             const obj = [{lat: coords[i].lat, lon: coords[i].lon, time: coords[i].time, speed: calculatedSpeed},
13                          {lat: coords[i+1].lat, lon: coords[i+1].lon, time: coords[i+1].time, speed: calculatedSpeed}]
14             exceededIntervals.push(obj)
15           }
16         }
17     }
18   }
19   return exceededIntervals
20 }
```

Figure 18 - Detecting potential over-speeding intervals from all GPS coordinates

The function above ensures that the user has minimum journey time of 2 seconds in the if statement on line 6, before it finds any speed intervals. It is crucial to perform error handling earlier, otherwise it may break the whole system if succeeding variables receives undefined or null data. Furthermore, we iterate through all the data to find the distance travelled between 2 coordinates with a difference of 4 seconds which is enough time to accurately measure the speed. The haversine formula returns the distance in miles by taking the starting and ending coordinates. Ultimately, the function returns a list of all GPS coordinates that have surpassed a threshold of 30mph since it is the most common speed limit on the roads in the UK. However, the threshold can be adjusted according to the situation of a driver.

Once we have access to all speed intervals during a journey, the coordinates extracted are now useful for locating user's road location to find road speed limit. The data returned from previous function includes a list of coordinates and the speed of the vehicle.

Since we have extracted speed intervals with 4 seconds' margin. The end results would contain hundreds or even thousands of speed intervals in longer journeys. Although, each interval requires an API call which fetches road information. This would result in the process becoming time consuming when comparing vehicle speed to road speed. Overall, this method would extremely slow down the server for other users during peak times.

To overcome the issue, another algorithm was implemented that would initially sort extracted coordinates in the journey. Followed by obtaining the lowest and highest GPS coordinates to calculate the overall radius in which the user has travelled in. The use of radius prevents several API calls, instead a single call can be invoked to retrieve all roads between a given radius. Ultimately, within a given radius the API return a list of all the road names and speed limits.

```
// THIS FUNCTION IS USED TO SAVE ALL NEARBY ROADS DATA INTO JSON
async function getallRoads([radius, lat, lon]) {
  const response = await fetch(`https://www.overpass-api.de/api/interpreter?data=[out:json];way(around:
  ${radius},${lat},${lon})[maxspeed];out;`);
  const data = await response.json();
  fs.writeFile("allRoads.json", JSON.stringify(data), function(err, result) {
      if(err) console.log('error', err);
  });
  return data
}
```

Figure 19 - The function returns & saves all road information within a given radius when latitude and longitude is provided

Ultimately, the extracted speed intervals are compared with each road speed limit in which the user has travelled on. Each iteration on all intervals ensures that the user hasn't travelled on 2 different roads in which case, those roads may not have same road limits. In this scenario, we avoid measuring over-speeding as there is an uncertainty about which road speed is relevant to our scenario. Correspondingly, the same process is followed, if the user has travelled on 2 different roads and the information about such road is not available in the API. In all other scenario, we end up with all relevant data required to measure over-speeding such as road name, speed limit, user's speed during the interval and the time this incident took place. These decision of avoiding certain speed intervals is crucial in this stage to ensure the user satisfaction and they have the best experience when using this website.

6.4.2 Detecting vehicle acceleration events

The approach behind acceleration events does not vary much from detecting over-speeding events. In fact, there are only 2 main stages that have led the measurement of vehicle acceleration. Since, a driving journey contains a list of GPS coordinates, majority of the collected data is irrelevant as the acceleration events would only occur when a vehicle is at rest and accelerates from 0mph.

The initial stage involves detecting intervals when the vehicle is at rest to ensure we do end up measuring acceleration when the vehicle is travelling on a road. The strategy involves targeting constant values to figure out intervals when a vehicle is at rest and removing varied coordinates that indicates that the vehicle is moving. However, due to the accuracy of the modern GPS signals, the latitude and longitude values extracted within the same period differentiated in the example below.

```
// GPS COORDINATES OF A CAR JOURNEY
const listOfCoordinates = [
  { lat: 50.911010873678165, lon: -1.393002284446745, time: 1648286306 }
  { lat: 50.911010887272425, lon: -1.3930022833776006, time: 1648286306 }
  { lat: 50.911010873324976, lon: -1.393002284474523, time: 1648286306 }
]
```

Figure 20 - A small example of a driving journey's GPS coordinates when the vehicle is at rest.

To overcome the problem, a threshold function was implemented to reduce the size of each value to be used for further comparison. The function takes a number as a parameter and returns a new coordinate with reduced length. Using this approach, the author can extract out all rest intervals from a car journey and measure the acceleration by using these intervals. Additionally, these values are utilised in the next stage to measure vehicle's acceleration.

```
// FOCUS ON MULTIPLE INTERVALS BECAUSE CAR COULD STOP AND ACCELRRATE MANY TIMES DURING A JOURNEY
export function showDistance(allCoords, extractedCoords, secs, limit) {
  const allAcceleration = []
    extractedCoords.forEach((coor) => {
      const res = allCoords.find(item => {
        return item.time === coor.time + secs
      })
      if(res !== undefined) {
        let [start, end] = [{latitude: coor.lat, longitude: coor.lon}, {latitude: res.lat, longitude: res.lon}];
        const acc = haversine(start, end, {unit: 'mile'}) * (3600/secs);
        if(acc > limit) {
          allAcceleration.push({speed: (acc).toFixed(2) + ' mph', time: res.time});
        }
      }
    });
    return allAcceleration
}
```

Figure 21 - The algorithm finds the acceleration of a vehicle by measuring the distance travelled over a period. If the acceleration has exceeded a limit hence it considers it as a rapid acceleration event.

So far, we have a list of all rest intervals that are useful for finding the acceleration of the vehicle. Using extracted intervals, the above algorithm iterates through each interval to measure the speed of a car by comparing it with another interval that occurred within 4 seconds. To achieve, Haversine formula was applied again to calculate the speed in miles and if the speed exceeds a limit therefore, that interval is considered as aggressive acceleration.

Chapter 7 - Testing

7.1 Testing strategy and environment used

The nature of chosen methodology has led the implementation and testing phase to be performed simultaneously by developing and testing each system component within short sprints. The strategy benefits the author to perform development testing at initial stage to discover and fix bugs while each component is under development. Due to the complexity behind programming of driving behaviours using various modules and algorithms, the chosen testing strategy brings ease and freedom for the author to carry each component development testing with a range of techniques. The fact that each driving behaviour is classified using both machine learning model and other APIs, development testing carries strategies such as unit, component and system testing to validate various system units incrementally to simplify debugging and reduce complexity.

In the given scenario, the author used Jest, a powerful JavaScript testing framework to bring the qualities of development testing to validate each system component within a sprint. Since the artefact is built on Nodejs, Jest can be installed and set up using node package manager.

```
npm install --save-dev jest
```

7.2 Testing within sprints

7.2.1 Sprint 1

The initial sprint was dedicated for the development of user interface (UI) as well as choosing the right environments as discussed in the section 6.1. During the development of UI, majority of the debugging was performed on the mobile browser since the artefact is reserved for mobile users. The author faced several challenges during debugging due to the lack of browser-based mobile web developer debugging tool which led to the solution of using alert

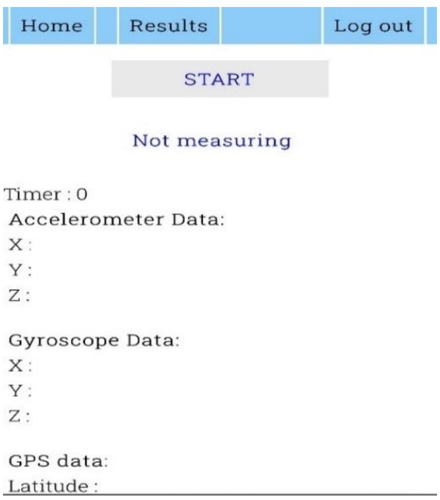


Figure 22 - Measuring driving behaviour page (non-measuring state)

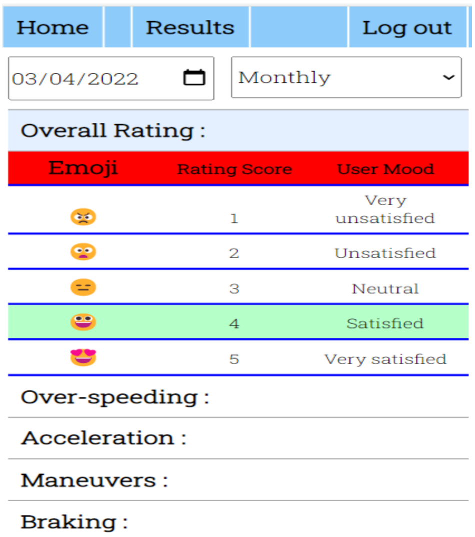


Figure 23 - Viewing driving journey results page

messages to view error messages from the console. Considering the problem, the length of this sprint was extended from a 1 week to 2 weeks as well as delaying the whole development and testing process. Ultimately, the author was unable to perform client-side functionality testing due to lack of resources and time. However, to ensure the performance and reliability of the artefact, the author performed testing manually by running and debugging the application several times to figure out the desired outcome of the user interface is achieved.

7.2.2 Sprint 2

Second sprint involved the development and testing of the machine learning model using one of the strategies of development testing. As discussed in section (6.3), the training and testing of LSTM model integrated of multiple units to create the overall component. The use of component testing becomes useful to figure out how precisely the model classifies the right driving events. To test its accuracy, the testing data is fed into the trained model in which the data processing is achieved through similar system units to which it was initially trained.

```
import * as ml from '../testingData/mlTestingData.js';

describe('LSMT model testing', () => {
  const inputData = ml.testingData;
  const targetData = ml.mlModelResults;
  inputData.forEach((drivingEvent, i) => {
      test(`EXPECTED ${targetData[i].eventType}(${targetData[i].number}) TO EQUAL ${drivingEvent[1]}`, () => {
        try {
          expect(drivingEvent[1]).toEqual(targetData[i].number);
      } catch (e) {
        console.log('UNCALSSIFIED EVENT: ', targetData[i].eventType);
      }
    })
  });
})
```

Figure 22 - Each testing sample's target result is compared with the outcome of LSTM network to check its accuracy on unseen data. If the network fails to classify then log that event out in the console.

In figure above, for each sample in the testing data, the expected results are compared with the LSTM output to visualise the results of the model. Since the model has not achieved 100% accuracy, in which case it might not classify all the events. Therefore, the use of try and catch statement ensures the test suite does not break while it's comparing the results. In return,

```
:\Users\Prince Singh\Desktop\FYP_CODE\Final_Artefact>npm run test ./tests/ml.test.js

> final_artefact@1.0.0 test
> jest "./tests/ml.test.js"

PASS tests/ml.test.js
 LSMT model testing
   √ EXPECTED LEFT CURVE(0) TO EQUAL 0 (4ms)
   √ EXPECTED LEFT CURVE(0) TO EQUAL 0
   √ EXPECTED LEFT CURVE(0) TO EQUAL 0
   √ EXPECTED LEFT CURVE(0) TO EQUAL 0
   √ EXPECTED LEFT CURVE(0) TO EQUAL 0 (1ms)
   √ EXPECTED RIGHT CURVE(1) TO EQUAL 0 (44ms)
   √ EXPECTED RIGHT CURVE(1) TO EQUAL 1 (1ms)
   √ EXPECTED RIGHT CURVE(1) TO EQUAL 1 (1ms)
   √ EXPECTED RIGHT CURVE(1) TO EQUAL 1
   √ EXPECTED RIGHT CURVE(1) TO EQUAL 1 (1ms)
   √ EXPECTED LEFT CURVE(0) TO EQUAL 1 (3ms)
   √ EXPECTED NON AGGRESSIVE(2) TO EQUAL 1 (3ms)
   √ EXPECTED NON AGGRESSIVE(2) TO EQUAL 2 (1ms)
   √ EXPECTED NON AGGRESSIVE(2) TO EQUAL 2
   √ EXPECTED NON AGGRESSIVE(2) TO EQUAL 2
   √ EXPECTED NON AGGRESSIVE(2) TO EQUAL 2 (1ms)
   √ EXPECTED BRAKING(3) TO EQUAL 2 (2ms)
   √ EXPECTED NON AGGRESSIVE(2) TO EQUAL 2
   √ EXPECTED BRAKING(3) TO EQUAL 3
   √ EXPECTED BRAKING(3) TO EQUAL 3 (1ms)
```

Figure 23 - All testing samples are passed

```
console.log
  UNCALSSIFIED EVENT:  RIGHT CURVE

    at Object.<anonymous> (tests/ml.test.js:18:17)

console.log
  UNCALSSIFIED EVENT:  LEFT CURVE

    at Object.<anonymous> (tests/ml.test.js:18:17)

console.log
  UNCALSSIFIED EVENT:  NON AGGRESSIVE

    at Object.<anonymous> (tests/ml.test.js:18:17)

console.log
  UNCALSSIFIED EVENT:  BRAKING

    at Object.<anonymous> (tests/ml.test.js:18:17)

Test Suites: 1 passed, 1 total
```

Figure 24 - Unclassified driving events are logged out in the console and 4 classified out of 22 testing samples

all samples are passed through the test and unclassified events are logged out in the console.

7.2.3 Sprint 3: (Over-speeding and Acceleration testing)

Over-speeding:

So far, we have tested 2 driving events such as dangerous manoeuvres and harsh braking using the machine learning model. The testing of other driving events is performed using unit testing in which individual program of the component is tested. As the functionality of both over-speeding and acceleration events is similar, these components were developed and

```
describe('Over-speeding functions testing', () => {
  const extractedCoords = overSpeedingTestData.realSpeedData();
  const findMaxRadius = overSpeedingFunctions.findMaxRadius(extractedCoords);

  test('Finding max radius function testing', () => {
    expect(findMaxRadius).toHaveLength(3);
    expect(overSpeedingFunctions.findMaxRadius([])).toEqual(expect.stringContaining('EMPTY GPS COORDS'));
  })

  test('Find max radius AND Fetch all roads within that radius', async () => {
    const getallRoads = await (overSpeedingFunctions.getallRoads(findMaxRadius));
    const visitedRoads = ['Northumberland Road', 'Derby Road', 'Durnford Road', "St Alban's Road", 'Northam Road'];
    // Object.keys(getallRoads).forEach((prop)=> console.log(prop));
    const roadsInRadius = getallRoads.elements.map((road) => { return road.tags});
    visitedRoads.forEach((road, i) => {
      try {
        expect(roadsInRadius).toEqual(
          expect.arrayContaining([
            expect.objectContaining({
              'name': road
            })
        ]));
      } catch (e) {
        console.log('ERROR: ROAD NOT FOUND ==> ', road);
      }
    });
  })
```

Figure 25 - Testing functionality of over-speeding component. The describe functions acts as a class and tests 2 main functions such as findMaxRadius () and getAllRoads ()

tested simultaneously.

To experiment with over-speeding events, the author gathered real data by visiting nearby roads to check the ability of Overpass API to return visited roads by using the collected GPS coordinates. As mentioned towards the end of section 6.4.1, certain scenarios are considered if the road data has gone missing from the API. Consequently, the speeding events are voided by considering the deficiency of detecting these events in scenarios where the speed limits of some roads are missing.

The describe function above runs 2 different tests to achieve the criteria, firstly by ensuring there is existence of some radius before the API fetches any road information. Followed by fetching all roads data within the given radius in the second test. Ultimately, the existence of each visited road in the fetched data confirms the availability of the road in the API. In return, both tests were passed in the function as well as listing 2 roads that were not found in the

Overpass API. The result shows the success of these tests, indicating the elimination of missing roads when detecting over-speeding events.

```
PASS  tests/over-speeding.test.js
  Over-speeding functions testing
    √ Finding max radius function testing (5ms)
    √ Find max radius AND Fetch all roads within that radius (1825ms)

  console.log
    ERROR: ROAD NOT FOUND ==>  Durnford Road

      at forEach (tests/over-speeding.test.js:31:17)
          at Array.forEach (<anonymous>)

  console.log
    ERROR: ROAD NOT FOUND ==>  St Alban's Road

      at forEach (tests/over-speeding.test.js:31:17)
          at Array.forEach (<anonymous>)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        4.01s, estimated 14s
Ran all test suites matching /.\\tests\\over-speeding.test.js/i.
```

Figure 26 - Both tests have passed, and not found- roads are logged out indicating these roads shall not considered to detect over-speeding events.

Acceleration:

As mentioned in the development section, acceleration intervals occur while the vehicle accelerates from rest (not moving) intervals. To increase the precision rate of detecting these events, the GPS coordinates are collected at high sample rate resulting in several hundred coordinates within minutes. Testing the functionality programmed behind these scenarios is crucial to make sure the system performs similarly to the expectations. Considering the situation, multiple tests were created to test each system unit by feeding in various types of data.

```
test('Extracted GPS coords to be bigger than 0 ', () => {
  const extractedCoords = acc.extractGPSlocations(accTestData.gpsData(),6);
  expect(extractedCoords.length).toBeGreaterThan(1);
});
```

After inputting several hundred GPS coordinates containing acceleration events, the test extracts all acceleration events from a given journey. The expected results shall be greater than 1 since there was presence of at least one acceleration event performed.

```
test('Same extracted coords to be 0 ', () => {
  const extractedCoords = acc.extractGPSlocations(accTestData.sameGpsData(),6);
  expect(extractedCoords.length).toBe(0);
});
```

Similarly, another test was created to expect 0 acceleration events since the data contained same GPS coordinates therefore no acceleration events occurred during the journey. The test makes sure the system does not produce unpredictable events in scenarios when the car has not moved from one place to another.

```
test('Reduce coords value', () => {
  const latitude = -1.325688745;
  expect(acc.reduceNumber(latitude,6)).toBe('-1.325688')
});
```

Reducing the length of GPS coordinates was vital to detect the movement of the vehicle. The test confirms the accuracy of the given function that reduces the value, to expect different value when a given length is passed as an argument. The length value of latitude shall be reduced to 6 after the dot.

```
// TESTED ON COORDS EXTRACTED FROM GOOGLE MAP AND RETURNED EXPECTED RESULTS
// COORDS CAN FOUND IN accTestData.js file in testingData
test('Find acceleration of present extracted coords', () => {
  const extractedCoords = acc.extractGPSlocations(accTestData.realgpsData(),6);
  const allCoords = accTestData.realgpsData();
  const seconds = 4;
  const accLimit = 55;
  expect(acc.showDistance(allCoords, extractedCoords, seconds, accLimit)).toEqual(
    [{speed: "59.71 mph", time: 1648553167}]
  )
```

Ultimately, above test detects potential acceleration events from extracted GPS coordinates by considering other factor which can be changed according to the situation such as minimum acceleration speed (55mph) to detect an acceleration event. The example expects any acceleration events that exceeds the given speed and considers it as a rapid acceleration event in a journey.

```
PASS  tests/acceleration.test.js
  Acceleration functions testing
    √ Extracted GPS coords to be bigger than 0  (6ms)
    √ Same extracted coords to be 0  (2ms)
    √ Empty extracted coords to be 0  (1ms)
    √ Reduce coords value (1ms)
    √ Find acceleration of present extracted coords (2ms)

Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        2.696s, estimated 3s
Ran all test suites matching /.\\tests\\acceleration.test.js/i.
```

All above test cases have passed the test suite highlighting the success of detecting acceleration events. The expected results are achieved by the system to accurately detect these events by considering other scenarios and factors which allows the system to adapt in the future. For instance, the acceleration speed may vary from a vehicle to vehicle where some vehicles are light and faster than other. In result, the artefact can accurately detect rapid acceleration events and be adapted according to the situation and become more reliable.

7.3 Discussion of testing and coverage

Testing coverage is a great tool to highlight how much of our coding functionality is covered during the testing phase by distinguishing the coding coverage of each component. As discussed earlier, time is major constrain on this project which has led certain deadline boundaries to be extended during initial sprints when the user interface testing was required.

Resulting in limited time to develop other components as well as perform testing, considering the complexity of all the functionality behind machine learning and other algorithms.

The fact that unit testing provides coverage for all the features and tests all possible states of functions, it was a suitable approach for UI testing. However, due to its effectiveness, it is one of the most time-consuming approaches which is one of possible reason it was not utilised within all areas of the application. However, the author has managed to test majority of the code, taking into consideration all techniques of development testing in areas where it was necessary such as unit testing for acceleration and component testing for machine learning model.

```
> final_artefact@1.0.0 test
> jest "--coverage"

 PASS  tests/ml.test.js
 PASS  tests/acceleration.test.js
 PASS  tests/over-speeding.test.js (14.645s)
---------------------------|---------|----------|---------|---------|-------------------
File                       | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
---------------------------|---------|----------|---------|---------|-------------------
All files                  |   96.84 |    69.04 |     100 |   98.88 |
 serverComponents          |   96.55 |    69.04 |     100 |   98.78 |
  acceleration.js          |     100 |       80 |     100 |     100 | 34-40
  over-speeding.js         |   95.58 |    65.62 |     100 |   98.41 | 56
 testingData               |     100 |      100 |     100 |     100 |
  accTestData.js           |     100 |      100 |     100 |     100 |
  mlTestingData.js         |     100 |      100 |     100 |     100 |
  over-speedingTestData.js |     100 |      100 |     100 |     100 |
---------------------------|---------|----------|---------|---------|-------------------

Test Suites: 3 passed, 3 total
Tests:       30 passed, 30 total
Snapshots:   0 total
Time:        15.934s, estimated 18s
Ran all test suites.
```

Figure 27 - The ultimate results of all the test suites with each test cases within the testing coverage

The figure above highlights the coverage of all the tests suites indicating each file name that have been tested followed with its features represented in percentages. Column 4 shows that all functionality within each component has been covered fully in the testing as well as percentage of lines tested and the line numbers that are uncovered in each test suites in the following columns. However, % Branch column has some decreased percentages which is a result of uncovered lines during the testing. These are in fact 7 uncovered lines resulted from null cases when the test suite has not considered in the coverage. Overall, 30 out of 30 test cases are passed covering all tests designed in 3 test suites.

In conclusion, the testing phase has been successfully performed by the author to figure out the accuracy of each system component by using various testing approaches. The results of all tested components indicate the positive impact the artefact would bring for the end-users by precisely detecting driving events and displaying results for self-awareness of driving skills. Nevertheless, lack of UI testing may not have a much of negative impact, considering the fact that majority of the functionality in the client-side has been debugged multiple times.

Chapter 8 - Evaluation against requirement

8.1 Achieved Objectives

Objective 1: Gather data from smartphone sensors (Achieved in Implementation section 6.2)

The overall aim of this project involved the training of machine learning models on data collected from smartphone sensors to identify potential driving behaviour events. To achieve the aim, several objectives were set to get a desired outcome. From which, gathering data from smartphone sensors was initial objective before the data is processed using any other methods. Within this objective, multiple goals were established and achieved during the development stage including requesting user permission to access smartphone sensors, using applicable sensors such as accelerometer and gyroscope as well as GPS coordinates.

Realistically, this objective has been achieved in a different manner to what it was planned for in the PID. The decision of using online available dataset was made after going through several literatures that shown successful results on this driving behaviour dataset. The fact that the dataset was gathered using efficient and professional approach involving assistants in the back seat that asked the drivers to execute a specific driving event. Furthermore, the drivers' performed given events had a good previous driving experience which has led the experiment to be accomplished in a safe manner. Additionally, the experiment covered four main behaviours listed on the user requirements such as dangerous driving maneuverers, harsh braking, rapid acceleration, and over-speeding.

Given the fact, smartphone used for this experiment was attached to vehicle's windshield which is a positive point since the literature shown similar approach to avoid dashboard vibrations which potentially adds noise to the dataset (satisfies non-functional requirement 1.1). Afterall, this artefact has achieved positive results when detecting driving events with a good success rate. In result, this objective was successfully achieved by using appropriate smartphone sensors for the detection of driving events and training the data on machine learning model.

Objective 2: Train and apply machine learning algorithms (Achieved in Implementation section 6.3)

Since the author had no previous experience of working with machine learning, it was a challenge to learn the concept and train appropriate ML models. According to the situation, the second objective included conducting a literature review on various ML models to choose a suitable one to train on driving event samples of variable length. For this scenario, the literature has shown the right direction by revealing the dominance of Long-Short-Term model for the training of each driving event sample with a larger input size. Since the LSTM model gained popularity by working with dynamics of sequences, it was a perfect option for classifying driving behaviours. Nevertheless, models such as K-means clustering, and support vector machine would not have been suitable due to their dominance in other problems where the data input scope is smaller and lesser classifications to be made respectively. Considering the situation, these models were idle for a given problem hence disregarded for the purpose of detecting driving events.

Though, LSTM has achieved a success rate of 82% by generalising testing samples, still it has lacked capabilities of detecting driving events in unsupervised approach. For instance, the model was trained on the dataset containing multiple driving events with timestamps when an event has occurred. Realistically, the timestamps would not appear in a real driving journey, therefore, the model may fail to classify those events. However, the problem has been flagged for future consideration where the feature extraction of potential driving event is required before the model classifies any events. Overall, the objective was fulfilled during the development and testing of LSTM model using Tenserflow.js and successful results achieved when tested on testing samples.

Since the requirements included the measurement of four driving events from which the model could only classify 2 (manoeuvres and braking), the other 2 events were implemented and tested successfully with the use of haversine formula and road APIs. The contribution of these driving behaviours brings opportunities for drivers to become more aware of their over-speeding and acceleration events that are also essential for road safety and driving skills awareness. Though, these behaviours were not part of this project's objectives, still the literature has shown their significance in the driving world. Considering their importance, the author has successfully implemented these behaviours by involving all related scenario to ensure that the users are completely satisfied with the ultimate feature.

Ultimately, the existence of several functional requirements indicates the importance of testing the system to provide the artefact that meets users' needs. This is achieved by performing testing on majority of the components on the server-side. Additionally, functional requirements alongside non-functional are fully met and satisfied only when the functionality behind components such as acceleration.js, over-speeding.js and ml.js produces desired results. For confirmation, the testing coverage towards the end of testing phase highlighted the accomplishment of these components to successfully detect potential driving events by considering all scenarios. Nevertheless, the testing coverage results weren't completely satisfied due to lack of testing on some branches (if statements) during the testing phase. Besides, majority of functionality were tested indicating that the objective was fulfilled successfully.

Objective 3: Develop a mobile website for end-users (Achieved during the implementation phase)

The ultimate objective was to develop a user interface for end-users to be able to measure their driving behaviour and view results of their journeys. Within given objective, the development stage involved testing and debugging of the system on HTTPS server locally to design an artefact that is available on all mobile platforms such as android and IOS. Since non-functional requirement (1.3) stated the compatibility of the website to be on all mobile platforms, HTTPS became a necessity as the IOS devices would deny the pop up to access sensor permission if the website is served on unencrypted HTTP.

Other requirement such as SQL database became a necessity to have a complete artefact that stores users' driving events into a relational database. Since the database provides freedom to access data points related to one another, it was a perfect option to store user's details and driving behaviours into separate tables to have an easy understanding, better structure, and bigger storage (satisfies non-functional requirement 1.2). Further, these benefits allow the ease of viewing and accessing of driving events into results section by retrieving data based on event date or type. Ultimately, the objective is achieved by representing results for end-users in statistics to easily analyse their driving behaviour and have interactions with other features such as viewing weekly and monthly driving results.

Since the system is developed with a lot of complexity behind each component, however, having a lesser complex UI design is always a plus point by considering all types of users with minimal technical skills. Considering the scenario, the use of questionnaire and user scenario successfully gathered users' perspective about how the system should look like for at least some features. Alongside this, users were also asked to choose additional preferable features for future considerations. Although. additional features were listed in both user and functional requirements, due to time limitations on this project, these requirements were prioritised as low therefore these would go as 'not met' requirements. On the other hand, other main functional requirements prioritised as high, and medium were achieved and tested successfully in the implementation stage. The participants in requirement elicitation process have played a significant role in making the process effective by indicating the design options for displaying driving results which eventually was designed according to the feedbacks and providing valuable feedbacks on how the system should behave in certain situations in the user scenarios.

8.2 Evaluation Against Requirements

8.2.1 Evaluation of functional Requirements

No.	System Service	Description of Functional Requirement Analysis	Validation Process	Requirement Achieved	Requirement Evaluation
1.1	Start/Stop measuring behaviour button	When clicked, the system shall ensure all permissions are granted and the sensor's data is collected while its measuring. The behaviours are based on certain sensor axis such as y-axis from gyroscope sensor to detect dangerous maneuverer and y-axis from accelerometer to detect braking events. The button shall change its content to 'STOP' to indicate that its currently measuring their behaviour. When button clicked again, the system shall stop measuring all behaviours and return to its original state.	If any of the permission is denied, the user shall not have access to measure any behaviour. The system shall alert users about the permissions that are denied and instruct users on how they can be permitted again.	Requirement met successfully	The button was successfully designed and programmed according to the description of given requirement analysis. Users can view and measure using both start and stop states of the button. Validation criteria is also met by not letting the users to measure until all sensors' permissions are granted. Additionally, users will be redirected to their device settings to turn their location access on.
1.2	Viewing driving results	The users shall access their driving results for both recent and previous journeys. The results data shall be available once a journey has been completed.	If previous journeys exist in the database, the system shall ensure all journeys are displayed to end-users. Otherwise, the users shall be alerted if no journeys have been made.	Requirement met successfully	"View driving results" page has been successfully developed according to the feedbacks gathered from the questionnaire (by considering emojis and user satisfaction). Both recent and previous results are accessible from the page including weekly and monthly journeys results.
1.3	Access Sensor Permission (*User Scenario 2*)	The system shall ask user to access sensor permission before the behaviour is monitored.	When all permissions are permitted, the system will alert users when it is ready to monitor and start monitoring behaviour right after.	Requirement met successfully	All use cases have been considered from the user scenario by prompting users to provide access to sensors beforehand. The validation process is successfully met to ensure the requirement is satisfied.

1.4	Instructions to follow before measuring. (*User Scenario 3*)	The system shall instruct user, every time a user starts to monitor, to follow instructions while the website is in use. E.g., Do not use mobile while its measuring. Do not close website page. Avoid phone calls.	The user shall agree to instructions every time they try to monitor to ensure the user has read and followed them.	Requirement met successfully	The instructions alert is set up successfully before the behaviour is measured. By ensuring, the users are reminded before all journeys to follow important and useful instructions to avoid incorrectly classified behaviours and for road safety.
1.5	Sign up/ Sign in (Login capability) (*User Scenario 1*)	Based on the inputs from sign up details, the system shall ensure the details match when users signs into the website. Each user using the system shall be uniquely identified by the system.	Multiple accounts with same usernames shall not be created. For instance, if username already exists, the user must be alerted straightaway.	Requirement partly met	A login field has been set up successfully based on requirement analysis. However, it has lacked some functionality mentioned in user scenario such as creating account with email, password, and date of birth. But the future considerations should involve the following features. Afterall, the validation process was also successfully met.
1.6	Area map highlighting roads	The system shall display a map highlighting roads a driver has driven on. (After each journey the map shall update to display new roads)	The map shall only update after each journey. (Especially when the driver chooses to stop measuring their behaviour)	Requirement not met	This requirement is an additional feature considered for future work. Low prioritise requirement not met due to time restriction. (Explained under achievement of objective 3)
1.7	A dashboard listing best national drivers' and their scores	Each user using the system shall view a list of best drivers in the country. The scores are based on each driver's behaviour in previous month journey.	After each car journey, each driver gets a score from a combination of driving events. The driver with a best score would obtain a top position in the dashboard.	Requirement not met	This requirement is an additional feature considered for future work. Low prioritise requirement not met due to time restriction. (Explained under achievement of objective 3)

Table 4 - Evaluation against each functional requirement

8.2.2 Evaluation non-functional Requirements

No.	Title	Description of Non-Functional Requirement Analysis	Validation Process	Requirement achieved	Requirement Evaluation
1.1	Mounting mobile device on windscreen shield.	To avoid dashboard vibrations, the mobile device of users must be mounted on windscreen shield instead of dashboard mount.	Giving instructions for users to follow before starting to measure driving behaviour. The user must agree and adhere to the instructions.	Requirement met successfully	This requirement was achieved successfully when dataset used for the training of the LSTM network was gathered while the mobile was attached to vehicle's windshield. The users are also advised to follow same approach in the instructions.
1.2	Monthly driving results (Storage requirement)	The driving results must be stored in the database for at least previous 30 days period for each user.	Each driving journey is stored in the database with a date of the journey. If a journey has exceeded 30 days' time, it shall be deleted.	Requirement met successfully	All driving results are stored in the database. Since each user has unique ID, these results are retrieved and displayed to end-users on results page. All journeys in the past week and month are easily accessible by the users.
1.3	Website compatibility	The website shall work on almost all devices that have both motion sensors and location access.	The system shall ensure whether the mobile device has relevant sensors to monitor driving behaviour.	Requirement met successfully	The UI is developed with considerations when the certain devices may deny permission access prompt if served on HTTP. Therefore, both the development and potential release version of the website is served on HTTPS.
1.4	System Availability (Reliability)	The system shall be available to end-users 99.9% of the time to ensure better performance and reliability.	Regular monitoring of the system to ensure better uptime and availability.	Requirement met successfully	Majority of the system components have been tested with a positive results outcome. The error handling during testing ensures the system downtime is unlikely to occur.
1. 5	User Interface Design	Regardless of complexity of the system, the website shall be easy to use, considering end-user with minimal technical skills.	Avoid unnecessary elements/features on the screen for users to feel more comfortable when using the system.	Requirement met successfully	The user interface is designed with minimal HTML elements to avoid complexity and considering users with minimal technical skills.

Table 5 - Evaluation against each non-functional requirement

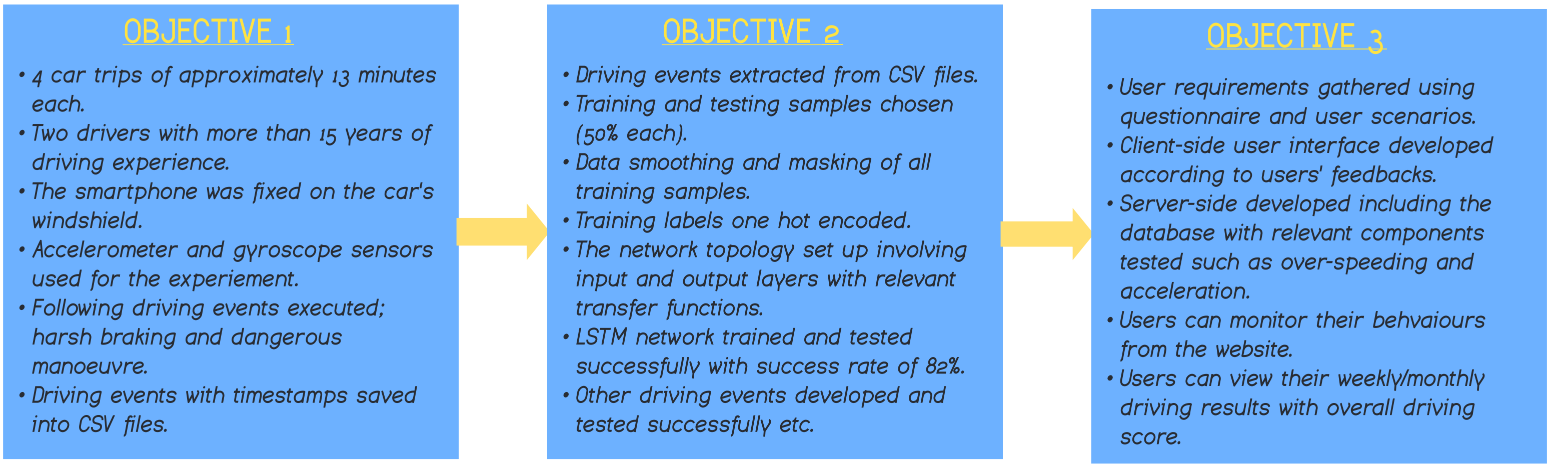


Figure 28 - A flowchart of how all objectives were met including the path taken to achieve each objective.

Afterall, all non-functional and important functional requirements were met in the development of the prototype according to users' perspective and feedbacks. In a result, an artefact has been developed with necessary features where the users can easily access and interact the system. From a broader view, the system allows users to monitor their driving behaviours by following important instructions as well as view their driving result based on dates which eventually helps drivers raise awareness about their driving skills by having control over rapid acceleration, dangerous maneuverer, harsh braking, and over-speeding events.

Chapter 9 - Conclusion

9.1 Summary

In summary, the project aimed to produce a platform for car drivers to measure four main driving behaviours using smartphone sensors. Initially, the project conducted a literature review on relevant smartphone sensors to capture behaviours as well as finding suitable data transformation techniques to extract and process the right data. Followed by choosing Long-Short-Term Memory machine learning model to train on driving samples taken from online dataset. In fact, only two behaviours such as dangerous maneuverer and harsh braking were detected using the model, the other two including over-speeding and rapid acceleration required the use of APIs and algorithms. While a mobile website was developed as a platform for users to measure their behaviours and view driving results.

9.2 Conclusions

In conclusion, the project has successfully achieved its aims and objectives meeting every goal set under all objectives. The proposed model has been developed based on the outcome of the literature review and the features using the feedbacks from both the questionnaire and users' scenarios respectively. The testing results have proven the success of LSTM model which has successfully identified 82% of all testing samples making the model more appropriate and effective for given dataset and problem. However, the model's success rate can be improved by training on high number of samples which eventually results in more accuracy as the model would have seen all possibilities of driving behaviours. Furthermore, based on the feedbacks from the requirement elicitation process, the user interface was successfully developed aiding users to interact with certain functionality on the system. Ultimately, all aspects of the objectives were achieved throughout the project that have contributed towards the solution of the problem of car drivers to increase driving awareness, eventually improving road safety.

9.3 Recommendations

There are number of recommendations for the prototype to be more successful. The first being the model's unsupervised approach of detecting driving behaviours as discussed in the evaluation section. This approach would make the model more powerful in terms of detecting various driving behaviours. However, this means the model would require considerable training samples leading to better generalisation of input samples. Additionally, large dataset would also require several training techniques which may involve training the machine learning model using various topologies (number of layers and neurons), cross-validation and training the network using different algorithms. These factors would play a significant role in increasing the accuracy of the model. Ultimately, further research in these characteristics may result in effective artefact that can be released as a real-world website/application.

Following previous improvement, additional features elicited from the questionnaire can also be added to the system. For instance, the users preferred "dashboard listing best national drivers" and "area map" features for the future. However, due to time restriction these features were held for future work as it requires a lot of time. Ultimately, these additions would fulfil users' satisfaction and encourages them to be loyal on the system which is a positive point since the overall system is developed for end-users.

Chapter 10 – Bibliography (APAV7)

Statista. (2022, March 25). *Passenger transportation in Great Britain by mode 2014-2020*. https://www.statista.com/statistics/300705/overall-use-of-transport-for-all-trips-in-the-united-kingdom/

Amado, S., Arıkan, E., Kaça, G., Koyuncu, M., & Turkan, B. N. (2014). How accurately do drivers evaluate their own driving behavior? An on-road observational study. *Accident Analysis & Prevention*, *63*, 65–73. https://doi.org/10.1016/j.aap.2013.10.022

Sundström, A. (2008). Construct Validation and Psychometric Evaluation of the Self-Efficacy Scale for Driver Competence. *European Journal of Psychological Assessment*, *24*(3), 198–206. https://doi.org/10.1027/1015-5759.24.3.198

Pugnetti, C., & Elmer, S. (2020). Self-Assessment of Driving Style and the Willingness to Share Personal Information. *Journal of Risk and Financial Management*, *13*(3), 53. https://doi.org/10.3390/jrfm13030053

Milleville-Pennel, I., & Marquez, S. (2020). Comparison between elderly and young drivers' performances on a driving simulator and self-assessment of their driving attitudes and mastery. *Accident Analysis & Prevention*, *135*, 105317. https://doi.org/10.1016/j.aap.2019.105317

Cervantes-Villanueva, J., Carrillo-Zapata, D., Terroso-Saenz, F., Valdes-Vela, M., & Skarmeta, A. (2016). Vehicle Maneuver Detection with Accelerometer-Based Classification. *Sensors*, *16*(10), 1618. https://doi.org/10.3390/s16101618

Fazeen, M., Gozick, B., Dantu, R., Bhukhiya, M., & González, M. C. (2012). Safe Driving Using Mobile Phones. *IEEE Transactions on Intelligent Transportation Systems*, *13*(3), 1462–1468. https://doi.org/10.1109/tits.2012.2187640

Castignani, G., Derrmann, T., Frank, R., & Engel, T. (2015). Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring. *IEEE Intelligent Transportation Systems Magazine*, *7*(1), 91–102. https://doi.org/10.1109/mits.2014.2328673

Ronao, C. A., & Cho, S. B. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*, *59*, 235–244. https://doi.org/10.1016/j.eswa.2016.04.032

Xia, H., Qiao, Y., Jian, J., & Chang, Y. (2014). Using Smart Phone Sensors to Detect Transportation Modes. *Sensors*, *14*(11), 20843–20865. https://doi.org/10.3390/s141120843

Papadimitriou, E., Argyropoulou, A., Tselentis, D. I., & Yannis, G. (2019). Analysis of driver behaviour through smartphone data: The case of mobile phone use while driving. *Safety Science*, *119*, 91–97. https://doi.org/10.1016/j.ssci.2019.05.059

Dai, J., Teng, J., Bai, X., Shen, Z., & Xuan, D. (2010). Mobile phone based drunk driving detection. *Proceedings of the 4th International ICST Conference on Pervasive Computing Technologies for Healthcare*. Published. https://doi.org/10.4108/icst.pervasivehealth2010.8901

Li, P., Abdel-Aty, M., & Islam, Z. (2021). Driving Maneuvers Detection using Semi-Supervised Long Short-Term Memory and Smartphone Sensors. *Transportation Research Record: Journal of the Transportation Research Board*, *2675*(9), 1386–1397. https://doi.org/10.1177/03611981211007483

Pejic, B., Pejic, A., & Covic, Z. (2010). Uses of W3C's Geolocation API. *2010 11th International Symposium on Computational Intelligence and Informatics (CINTI)*. Published. https://doi.org/10.1109/cinti.2010.5672223

Schrader-Patton, C., Ager, A., & Bunzel, K. (2010). GeoBrowser deployment in the USDA forest service. *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application - COM.Geo '10*. Published. https://doi.org/10.1145/1823854.1823887

Ramah, S. E., Bouhoute, A., Boubouh, K., & Berrada, I. (2021b). One Step Further Towards Real-Time Driving Maneuver Recognition Using Phone Sensors. *IEEE Transactions on Intelligent Transportation Systems*, *22*(10), 6599–6611. https://doi.org/10.1109/tits.2021.3065900

Castignani, G., Derrmann, T., Frank, R., & Engel, T. (2015). Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring. *IEEE Intelligent Transportation Systems Magazine*, *7*(1), 91–102. https://doi.org/10.1109/mits.2014.2328673

Carvalho, E., Ferreira, B. V., Ferreira, J., de Souza, C., Carvalho, H. V., Suhara, Y., Pentland, A. S., & Pessin, G. (2017). Exploiting the use of recurrent neural networks for driver behavior profiling. *2017 International Joint Conference on Neural Networks (IJCNN)*. Published. https://doi.org/10.1109/ijcnn.2017.7966230

Johnson, D. A., & Trivedi, M. M. (2011). Driving style recognition using a smartphone as a sensor platform. *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. Published. https://doi.org/10.1109/itsc.2011.6083078

Tango, F., & Botta, M. (2013). Real-Time Detection System of Driver Distraction Using Machine Learning. *IEEE Transactions on Intelligent Transportation Systems*, *14*(2), 894–905. https://doi.org/10.1109/tits.2013.2247760

Kansal, T., Bahuguna, S., Singh, V., & Choudhury, T. (2018). Customer Segmentation using K-means Clustering. *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*. Published. https://doi.org/10.1109/ctems.2018.8769171

Shahverdy, M., Fathy, M., Berangi, R., & Sabokrou, M. (2020). Driver behavior detection and classification using deep convolutional neural networks. *Expert Systems with Applications*, *149*, 113240. https://doi.org/10.1016/j.eswa.2020.113240

Wollmer, M., Blaschke, C., Schindl, T., Schuller, B., Farber, B., Mayer, S., & Trefflich, B. (2011). Online Driver Distraction Detection Using Long Short-Term Memory. *IEEE Transactions on Intelligent Transportation Systems*, *12*(2), 574–582. https://doi.org/10.1109/tits.2011.2119483

Winarno, E., Hadikurniawati, W., & Rosso, R. N. (2017). Location based service for presence system using haversine method. *2017 International Conference on Innovative and Creative Information Technology (ICITech)*. https://doi.org/10.1109/innocit.2017.8319153

J. (2017, February 14). *GitHub - jair-jr/driverBehaviorDataset*. GitHub. https://github.com/jair-jr/driverBehaviorDataset

Chapter 11 – Appendices

Appendix A – Project source code

This GitHub repository has the artefact developed in this project

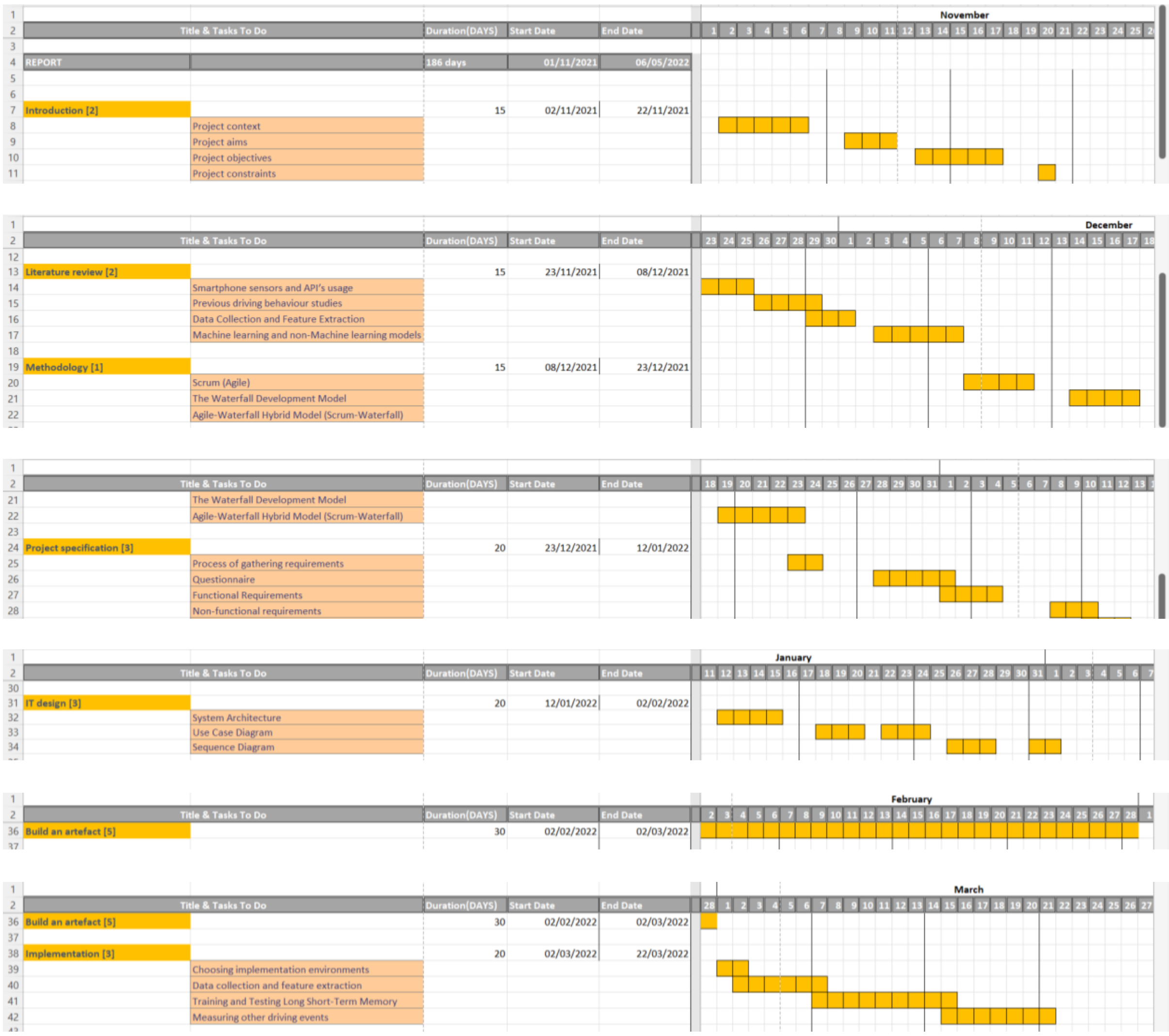
https://github.com/UP941594/FYP

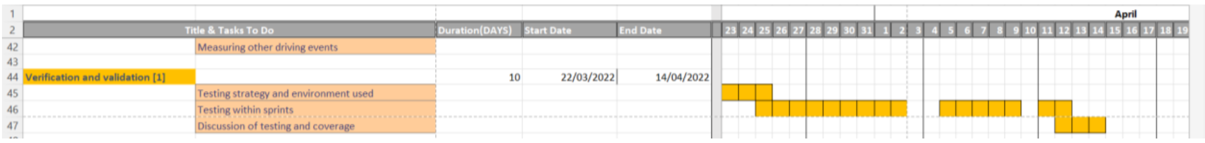
Appendix B – Online Driving Behaviour Dataset

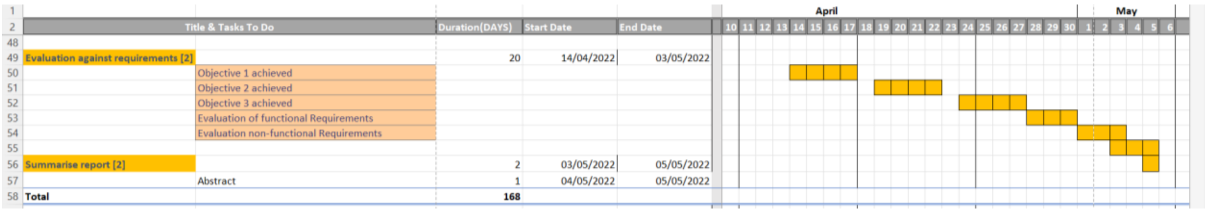
A GitHub repository containing all the experiments conducted using smartphone sensors for driving events.

https://github.com/jair-jr/driverBehaviorDataset

Appendix C – Project Planning on Microsoft Excel







Appendix D – Questionnaire Results for Requirement Analysis

Measuring Driving Behaviour Using Mobile Phone Questionnaire

This questionnaire is for a website that measures your driving behaviour while your phone is attached to car's dashboard or windshield. Every journey counts. After each car journey, the website will display statistics of following driving behaviours: rapid acceleration, harsh braking, dangerous maneuverers and over-speeding.

This questionnaire takes less than 5 minutes to complete.
Participants can withdraw at any time.

Questionnaire about your driving experience

What is your age?

- (selected) 16-25
- 26-35
- 36-45
- Over 45

On average, how many miles you drive a day?

- (x) 1-10
- ( ) 11-20
- ( ) 21-30
- ( ) Over 30

On a scale of 1 to 5, how do you rate your driving behaviour/performance?

- ( ) 1 - Very Good
- (x) 2 - Good
- ( ) 3 - Average
- ( ) 4 - Poor
- ( ) 5 - Very Poor

Do long journeys affect your driving concentration?

- (x) Yes
- ( ) No
- ( ) Don't know

Questionnaire about the website

Would you give permission to the website to access your device location and motion sensors?

- ○ Yes
- ○ No
- ◉ Maybe

Where do you prefer placing your phone when driving?

◉ Dashboard Mount

○ Windshield Mount

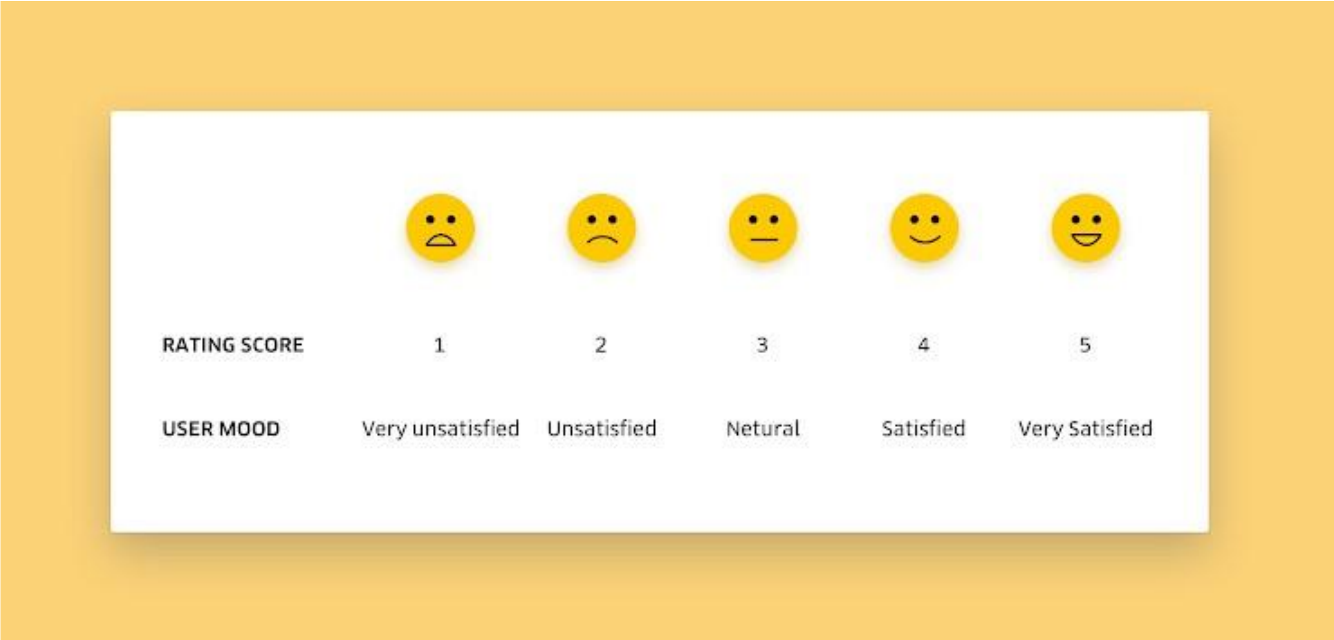
Would you prefer automatic driving monitoring in background or while the website's page is open?

- Automatic driving monitoring
- Website's page is opened

Which time of the day should the website monitor driving behaviour?

- Day
- Night
- Day+Night

How would you prefer viewing your driving results?



- [ ] Rating Score + User Mood
- [x] Emoji + User Mood
- [ ] Both

For how long should your driving data be stored for future comparisons of your driving results?

- [x] Weekly
- [ ] Monthly
- [ ] Annually
- [ ] Several Years

Pick additional features that you would prefer using in the website

- [ ] A dashboard listing best national drivers
- [ ] Area map highlighting roads you have driven on
- [ ] Login capability to access your driving data from any mobile device
- [x] None of above (just driving behaviour results)

Contact details:
For any further questions, please contact on UP941594@myport.ac.uk

This form was created inside University of Portsmouth Myport.

Google Forms

Appendix E – Ethics Review Certificate

Appendix F – Project Initiation Document (PID)

School of Computing
Project Initiation Document

{Preetpal Singh}

{Measuring Driving Performance Using Phone
Sensors and API's}

{PJE40}

1. Basic details

Student name:	Preetpal Singh
Draft project title:	Measuring Driving Performance Using Phone Sensors and API's
Course:	Software Engineering
Project supervisor:	Zhaojie Ju
Client organisation:	
Client contact name:	

2. Degree suitability

Please describe how your project satisfies the criteria for your current course. For example, if you are a Software Engineering student, please explain why your project is suitable for a Software Engineering degree.

In each section please write your text below ours in regular (non-italic) font.

- There is always a better way of solving a problem using the software. I believe, the problem outlined in this project can be solved by the knowledge of technologies that I have gained over the past few years in this course and by learning new techniques of Machine Learning. The idea behind this project includes the use of programming and problem-solving skills which are the main principles of software engineering.

3. Outline of the project environment and problem to be solved

<table>
<tr><td>For engineering projects without a client:</td><td>For projects with a client:</td><td>For theoretical or study projects:</td></tr>
<tr><td>What is the problem that you will investigate?

- The problem has arisen from lack of tools availability for car drivers to measure their driving performance. The investigation would mainly focus on helping car drivers to enhance their driving skills.

Why is it worth working on?

- Driving a car has become an important aspect in our daily lives, it does not only help us travel faster but also provides the opportunity to travel anywhere we like. I believe, to take comfort of these preferences, good driving skills are essential. Therefore, it would be beneficial for the drivers to be aware of how well they are doing with their driving. Ultimately, an individual can make note of their driving behavior and may consider improving it by not repeating the same mistakes again. Overall, this idea will help reduce number of road accidents as well as providing better safety on the road.</td><td>Who is the client?
What do they do?
What is their problem? Why does it need to be solved?</td><td>Who is the intended readership/audience?
What is the contextual significance of this topic?
What are the research questions you are seeking to answer?</td></tr>
</table>

4. Project aim and objectives

What is the overall aim of the project?(Description of the overall project)

- Build a mobile website to help car drivers to measure their driving performance.

What are the objectives that will lead to you meeting that aim?
Objectives are goals that will lead me to meet the aim

- User will be asked to grant permission before accessing their mobile sensor's data.
- Get raw sensor data from accelerometer, gyroscope and Geolocation API.
- Apply machine learning algorithms (MLA) and Data filtering methods to raw data collected from sensors.
- Use patterns produced from MLA to create good and bad driving statistics for the user.
- The statistics will be presented to users on a mobile website user interface.

5. Project deliverables

For an engineering project, what information system artefacts will be developed? What documents will be produced? This always includes your project report, but could also include supporting documentation for your client such as requirement and design specifications, test strategies, user guides, that are useful outside of the project report.

For a study project, are there anticipated outcomes besides the report, for example datasets or recommendations to external bodies?

- The main deliverable is the report itself and the artefact. Both would consist of the main problem solved, key requirements (based on literature), testing strategies used, maintenance (evolving the artefact over time) and designing a final artefact.
- The artefact is a mobile website in a repository that has all necessary folders such as JavaScript files, database, server and other key files included.

6. Project constraints

What constraints are there on your solution to the problem? For example, you could not test a medical system on real patients.

1. During the testing phase, I will not be able to drive the car to test the system myself. This may put other road users under danger. Instead, I would need someone to drive the car for me while I can measure/record the resulting data from phone sensors.

2. Different types of smartphones will be required during testing to measure sensor data. The fact that all smartphone sensors are not manufactured by the same companies. Therefore, the data emitted by the sensors may vary from phone to phone, as well as at different frequency.

3. Before any data is collected from the sensors, the mobile phone acquires user permission which must be granted otherwise no data will be collected from user's phone.

4. The data released from sensors needs to be calculated to measure the driving behavior. The calculation will be done using machine learning algorithms. It will be a challenge for me to learn and apply these algorithms.

7. Project approach

How will you go about doing your project? What background research do you need to do?
For an engineering project, how will you establish your requirements?
For a study project - can you refine your larger research area into research questions that you can meaningfully answer? What skills do you require and how are you going to acquire those that you do not already have? What methodologies are you going to use?

Background-Research:

- Find out the rate at which the data emitted by different phone sensors.
- How machine learning algorithms can be used to measure the data collected from sensors.

Additional approach:

- Find and apply appropriate software model to design the artefact.
- Work out different testing strategies to validate the data.
- How driving behavior changes on different road types such as rural, urban or motorway.

Building/Testing the artefact:

- While developing locally, the server needs to be running over HTTPS otherwise the IOS device would not authorize to obtain user permission to access phone sensors.

- During testing phase, the phone-holder will be attached to car's windscreen to achieve full motions of the car. Dictation

Requirements:

The basic requirements are gathered from the literature. The focus behind this project is to fulfill basic needs of car drivers such as measuring the followings:

1. **rapid car acceleration** (using Geolocation API to get distance travelled between 2 points in time)

- Supervised Machine Learning approach such as linear regression, Support Vector Machine etc.

2. **dangerous driving maneuvers** (Gyroscope sensor)

- Unsupervised machine learning approach because the data is unlabeled and unclassified. This approach will self-discover any natural patterns using its algorithms. The common methods like k-means, Artificial Neural Networks (feedforward NN), Stacked Neural Network will be used.

3. **over speeding** (Geolocation API and ROAD SPEED API)

- Using OpenStreeMap API which is free open-source project. The API will be used to get road speed data which is then compared with vehicle speed to find out if driver is over speeding.

4. **unexpected/harsh braking** (Gyroscope sensor) - Unsupervised machine learning process.

5. Literature review plan

What are the starting points for your research? (e.g. specific books or papers in journals, existing reports or documents, online resources, existing systems)

Link for the literature review plan:

- https://docs.google.com/spreadsheets/d/1K6SePsBZkb_zkThWACOZaDl_y34qB4HQ/edit?u sp=sharing&ouid=107072164636980488658&rtpof=true&sd=true

Tools & Tips:

- Important tools such as EBSCO Discovery Service and Google Scholar.
- Good source material usually has high citation count therefore they can be relied upon.

How to get raw sensor and geolocation API data:

1. Open-source driving assistance system has raw sensor data of Public GPS Traces

- https://www.openstreetmap.org/traces
- https://github.com/dragonpilot-community/dragonpilot

2. Smartphone recorded driving sensor data

- https://databank.illinois.edu/datasets/IDB-4650469#

3. Driver Behavior data collected while performing different driving events (gyroscope and accelerometer data included). Events such as aggressive breaking, left or right turns as well as non-aggressive events

- https://github.com/jair-jr/driverBehaviorDataset

6. Facilities and resources

What computing/IT facilities will you use/require?
What other facilities/resources will you use/require?
Are there constraints on their availability? If funds are required to acquire them, have these been allocated? Will they be available in time?
For example, you might need a specialist lab or equipment at the university, which might be in use in teaching and by other project students. Your own computer and free software, or software you already have, do not normally need to be mentioned.

- No specific computing/facilities is required for this project. However, I already have a car, in which I will be testing the system as well as a friend of mine who is available anytime to drive the car for me, while I can test the system.

7. Log of risks

What risks will you encounter when doing your project? What backup plans do you have if identified things go wrong?
What is your plan for reviewing risks? Remember that risk probabilities, and hence priorities, will change over the course of the project, so this section should be maintained. Use a table like below.

Description	Impact	Likelihood	Mitigation	First indicator
COVID-19 outbreak means I cannot get into a lab for usability testing	Severe	Likely	Get in while I can, prioritize lab tasks in time. Make an alternate test plan that does not need the lab.	University informs that lab closure is likely
Bad weather such as heavy rain or snow could affect driving while testing the system.	Minor	Likely	Check weather forecast in advance before deciding the days for testing and when it should be done.	Weather forecast app such as Weather app on IOS or BBC weather.
Low mobile battery during testing phase.	Severe	Unlikely	Keeping phone battery topped up before going out and test the system.	Set reminders to keep battery charged up so if it is low then can plug it in beforehand.

8. Project plan

What do you need to do to create the artefact / do the primary research and write the report? Walk through your proposed approach and break it down into tasks. When are you planning to perform these tasks? When do you need access to other people or resources? Usually a Gantt chart is a good way of presenting the plan. Note that plans can change over the course of the project, so this plan should be maintained.

- The project plan consists of 2 sections, one has got PID plan which I have used throughout to plan my PID and second has got a plan for the report.

Link for the Project Plan:

- https://docs.google.com/spreadsheets/d/1ai5-HyrTdQRMQ8Hm9BZMlqEHwNCqfGG1/edit?usp=sharing&ouid=10707216463698048865 8 &rtpof=true&sd=true

9. Legal, ethical, professional, social issues (mandatory)

What are the legal/ethical/professional/social issues that may impose constraints on the project? How will you ensure that they will be addressed, or what steps will you take to avoid/mitigate their effects?

Whatever project work you are doing, you must consider its security implications, for the data you generate or use, or for the software artefact itself. Please describe how you are taking these into account. There is also a question about security on the ethics review form (see below)

*All students must complete the ethics review form at https://sums.soc.port.ac.uk/ethics at this time. Has your supervisor (and the FEC representative, if required) seen and approved your ethics form? **Remember** – this is obligatory and must be completed **now**. The school's FEC representatives are Dr Matt Dennis and Dr Philip Scott.*

The **legal issues** which need to be avoided for this project are the followings:

- Wear a seatbelt when driving or sitting in the passenger seat.
- Do not use hand-held phone when driving.
- Obeying traffic rules to keep everybody else on road safe.

It is important to obey these rules mentioned above, these road offences could lead to a fixed penalty notice or imprisonment. Planning is essential to overcome these issues, making sure the seatbelt is on before turning the car ignition on, using phone holders instead of hand-holding the phone as well as obeying traffic rules.

There are no ethical issues which could arise when considering working on this project. However, some **social and professional issues** which we may consider are listed below:

- To avoid car pollution, fuel efficient cars will be used during the testing phase to reduce emissions as well as to save fuel. I will also be keeping the engine off when the car is not in use or parked.
- It is important to ensure that the project does not affect the reputation of university therefore, adhering to certain rules and regulations when working on this project as well as when testing the artefact.

Certificate of Ethics Review - Preetpal Singh (UP941594) - TETHIC-2021-101469.pdf

- https://portdotacdotukmy.sharepoint.com/:b:/g/personal/up941594_port_ac_uk/ETZQbqp7CtNoL4l7EXWv5IB2anmpWd6odjj7XM_W7cjfw?e=FzpeUy