

*CAR DRIVERS MONITORING
BY MEANS OF MOBILE
PHONE APPLICATION*

Author: Boris Yordanov

Supervisor: Andreas Kamlaris | Critical Observer: Hans Scholten

Abstract

Fatigue in drivers is one of the leading causes of traffic accidents, yet a lot of people are not able to recognise when it sets in or when to take a break, leading to an overestimation of their current driving capabilities. Exhaustion tracking is readily available in newer vehicles, but is inaccessible for the bigger part of the population. This research project aims to identify the causes of drowsiness in drivers, obtain the possible indicators of fatigue and detect it using solely the sensors of a mobile phone, making it more accessible and hopefully prevent a number of accidents. A literature review is conducted in order to get a clearer picture of the work that was previously done on the topic of driving fatigue, gathering information about the state of tiredness, as well as already available products that serve a similar purpose. Consequently, the most appropriate combination of sensors was identified and mock-ups of the user interface and application features were produced. Additionally, a model for fatigue prediction was designed, using Logistic Regression as a basis. Thereafter, the application was developed for the Android mobile platform using various internal and external libraries. The application was released on Google Play Store for user testing in order to assess its feasibility and collection of feedback data, later used for the improvement of the prediction model and algorithm. The positive aspects were the user interface design, configurability and the approach for detection and alerts. The main issue testers experienced was false-positives, stemming from factors such as using the application in a non-monotonous setting or using the phone while the tracking activity was running.

Table of Contents

ABSTRACT	1
1. INTRODUCTION	4
2. STATE OF THE ART	5
2.1 CAUSES OF FATIGUE AND ITS EFFECTS ON DRIVING	5
2.2 PREDICTING FATIGUE	6
2.3 USING SMARTPHONES TO DETECT FATIGUE	7
2.4 RELATED WORK	9
2.4.1 DRIVELERT	9
2.4.2 DRIVEAWAKE	10
2.5 CONCLUSION	11
3. IDEATION	13
3.1 GENERAL CONSIDERATIONS AND REQUIREMENTS	13
3.2 THE USER INTERFACE	14
3.2.1 THE “NOW TRACKING” SCREEN	14
3.2.2 POP-UP ALERTS	14
3.2.3 STATISTICS PANEL	14
3.3 THE DETECTION ALGORITHM	16
3.3.1 THE PREDICTION MODEL	16
3.4 ETHICAL IMPLICATIONS	16
3.4.1 PERFORMANCE TRACKING	16
3.4.2 PRIVACY AND SECURITY	16
3.4.3 SERVER STORAGE AND INFORMED CONSENT	17
4. REALISATION	18
4.1 PRODUCT DEVELOPMENT	18
4.1.1 THE LOGISTIC REGRESSION MODEL	18
4.1.2 THE USER INTERFACE	19
4.1.3 THE BACKGROUND SERVICES	22
4.1.4 FEEDBACK AND DATA COLLECTION	23
4.2 USER TESTING AND EVALUATION	24
4.2.1 INITIAL RELEASE	24
4.2.2 UPDATES	24
4.2.3 PROMOTION	25
4.2.4 COLLECTED FEEDBACK	27
4.3 IMPROVING THE MODEL	28

5. DISCUSSION	29
5.1 FUTURE WORK	30
APPENDIX A: INITIAL MOCK-UPS	31
APPENDIX B: DATA COLLECTION	32
1. SURVICATE SURVEY	32
2. FEEDBACK FORM	32
3. COLLECTED DATA	33
REFERENCES	35

1. Introduction

Driving in a state of fatigue is one of the most common reasons for car crashes worldwide. According to European statistics, 10% to 20% of all traffic accidents are caused by diminished attention levels caused by fatigue [1]. Moreover, drowsy driving is particularly dangerous, due to the fact that the driver cannot take any evasive actions because of falling asleep. According to the National Highway Traffic Safety Administration in the US, the average fatality rate in accidents related to fatigued driving is 3.6% [2]. Drowsiness in drivers is caused by a wide range of factors such as sleep deprivation, driving duration, the surrounding environment or the circadian effect [3]. In most cases, drivers are not fully aware of their fatigue level and would underestimate its hazardous nature. Therefore, it is crucial to identify the symptoms of tiredness and prevent the occurrence of unfortunate events.

Such a detection mechanism requires a fault-measurement device with reasonable accuracy, yet it should be accessible to people and must not obstruct their driving capability. Possibilities are integration within the car's chassis, external sensors or smartphones. Contemporary mobile phones are equipped with multiple sensors such as GPS, accelerometer, gyroscope, camera(s) and are powerful enough to process large quantities of data from them. They can be used for identifying driving actions typical for fatigued drivers - hard braking or accelerating, lane drift or eye closure [4]. What's more, 64.6% of people aged 18-64 in the US own a smartphone [5] and this rate is constantly rising. All this makes smartphones a great platform to build a fatigue prevention mechanism that alerts drivers and prevents them from falling asleep.

This project aims to explore the feasibility of using smartphones as a tool for detection of fatigue, which would result in fewer accidents on the road. It provides a critical overview of the current research done on falling asleep behind the wheel, as well as on techniques for detection and prevention. Therefore, the main research question can be formulated as:

“How can smartphones be used for accurate, yet non-intrusive detection and prevention of driver fatigue?”

To help define the answer to that question, it is necessary to look into what causes fatigue in drivers, the methodologies that are used to predict fatigue as well as explore in detail what smartphone sensors could potentially be used to detect fatigue with fair accuracy.

2. State of the Art

This chapter provides a critical overview of the current research done on the causal factors of falling asleep behind the wheel, as well as on methods of detection and prevention. In the beginning some details on what the effects of fatigue are on driving performance were mentioned, followed by research on fatigue prediction and using smartphones to detect and prevent drowsy driving. Similar applications and their feasibility will also be discussed.

2.1 Causes of Fatigue and its Effects on Driving

Regardless of the causes, drowsy driving leads to unfortunate events and even fatal accidents. D. Mollicone et al. [6] have shown that fatigue causes a decrease in the vigilant attention and worsens the reaction time of the driver. Researchers have also distinctively categorised fatigue into sleep-related (SR) and task-related (TR) based on the causal factors [7]. Examples for SR fatigue are extended duration of wakefulness and the effect of circadian rhythm, which produces a dip in alertness during the early hours of the afternoon. This is in direct correlation with the increased amount of crashes between 2 and 6 AM as well as 2-4 PM [8], where the cause of the accident is drowsy driving. TR fatigue, on the other hand, is caused by the driving task itself or the environment and commonly occurs when there is a presence of monotony and prolonged driving.

Sleep-related fatigue, being highly dependent on the individual, is less susceptible to detection by means of technology, which makes task-related fatigue the focus of interest for such intervention systems. According to [9] and [10], TR fatigue can be subcategorized into active and passive forms. Active fatigue is caused by mental overload (highly demanding driving conditions such as traffic, corners, visibility), where passive is caused by mental underload (driving on highway/straight road with no traffic). Gastaldi, Rossi and Gecchele [11] primarily focus on the analysis of the passive TR (task-related) fatigue and studies the effects of highway driving on the driver's state. A peak of optimal driving performance was found between 20-30 minutes of driving. Similarly, the time of day theory was evident as experiments conducted in the early afternoon revealed that the variables were higher than those in the morning. However, it was found that driving in the afternoon within a monotonous environment would decrease performance, the exact opposite would happen in the morning since the circadian effect is absent. What is more, E. A. Schmidt et al. [12] have concluded that drivers tend to misjudge their level of tiredness during prolonged and monotonous driving, which further increases the possibility of crashing. This distinction between SR and TR fatigue is

necessary when designing the model for detection, since some variables may have higher weights, while others are considered secondary.

2.2 Predicting Fatigue

The most common method that is used to detect fatigue is tracking the driver's current state. The measurements encompassed by this method include eye closure tracking, gaze detection, blinking rate or yawning. Methodologies like the PERCLOS (Percent Eye Closure) are camera-based and, in good lighting conditions, can detect symptoms on drowsiness accurately [13]. However, using a camera may produce many false alarms as there are outside factors, such as sunlight, headlights, camera position, that would alter its recording performance, as can be concluded from the tests in [14]. Using the driver's state, although extensively researched and used, is not of primary interest since it can be intrusive and is highly related to sleep-related fatigue, which was previously identified as very subjective.

Instead of tracking the driver directly, some methods focus on detecting hazardous driving actions stemming from fatigued driving. Such systems track lane-keeping and the absolute position of the vehicle. Speeding and hard-braking may provide additional indication of the driver's performance. Mollicone et al. [6] analyse the relation and occurrence between hard-braking events and fatigue, particularly in truck drivers. In total 7320h of driving were recorded with a total of 186 hard-braking events, out of which 58% were associated with a (median) fatigue score of 5.9 or greater. As expected, the frequency of hard-braking events increased as the predicted level of fatigue climbed. Another indirect performance measure is the driving duration. In lorry drivers, it is stated that the odds of a crash start increasing after 5 hours and are twice as high in that second half of the drive. After 10-11 hours, the risk is said to increase 7 times [15]. When driving for prolonged distances it has been identified that the optimum driving time on a highway is 80 min [16]. All this evidence suggests that there is a strong relationship between driving performance and fatigue and variables such as hard-braking and duration can be used for confidently detecting drowsiness.

A combination of both the state and performance is also used for the detection of drowsy driving. In 2004, the Centre for Research and Technology Hellas completed the project AWAKE [17], funded by the European Union. The aim was to deliver a system that would be used to monitor a driver's vigilance in a non-intrusive way using eyelid movement, lane tracking, steering grip and position, and braking. Lee and Chung [18] also follow a similar approach – tracking performance variables such as speed while having a vital signs sensor on the driver's finger as well as facial tracking with the camera. The combined measurement

of state and performance of measurements has a serious disadvantage - it requires a very complex system of sensors and algorithms that either need to be implemented in the car itself or need external devices, which makes it not easily accessible for otherwise potential users.

2.3 Using Smartphones to Detect Fatigue

The camera in a smartphone can track and evaluate the driver's face to classify the state of sleepiness. Such a method was developed in [18], where the smartphone's front-facing camera was used together with a wireless vital signs sensor attached to the hand. Tests were made where the average true awake state and true drowsy state predictions were 96% and 97% accurate respectively. On the other hand, some drawbacks of having additional sensors were identified in [14] and therefore a system that solely uses the mobile phone's camera was constructed. The smartphone camera is used to extract a variety of facial properties such as blinks, gaze, orientation, smile. WakeApp had 99% accuracy in detection during the day and when the driver's head was moving, but that drops significantly to 30% during night time. Wearing eyeglasses decreased the accuracy insubstantially to 95%, while driving with tinted sunglasses had a success detection rate of only 20%. What can be concluded from this implementation is that facial tracking is an excellent method of fatigue detection, only if the conditions are ideal. Therefore, for providing a more versatile detection, other sensors are needed.

By using an accelerometer one can measure the rotation of the device with respect to a reference point. It does so by taking into account gravity [19]. When mounted directly on the steering wheel, this can be translated into SWM (Steering Wheel Movement). Lawoyin et. al [20] implement this low-cost method that solely uses an accelerometer to detect drowsiness. The system was tested with 4 drivers and resulted in 72.0825% accuracy using only the accelerometer, which is not ideal but good for only using a single sensor. What may be considered problematic with measuring steering wheel movement is the mounting of the sensor, as being directly on the steering wheel may be way too intrusive for the driver. False triggers may also occur more often, which leads to the conclusion that additional measurements are necessary.

A more accurate and sensible approach is to use a combination of accelerometer, gyroscope, and magnetometer, which can be found in most smartphones nowadays. In addition to the accelerometer, the gyroscope would add a correction with respect to the rotation of the phone around itself, while the magnetometer does so with respect to the magnetic north. Similar sensor-fusion system is used in [21] and [4] in order to track driving behaviour. There was a

difference of 0.43g between aggressive and nonaggressive turning on average, except for U-turns where the average difference is 0.35g. The Bayes algorithm in [4] had 93.3% accuracy, more than 20% improvement over using only the accelerometer. Although sharp-turning is not necessarily a symptom of fatigue, acceleration and hard-braking can be measured using the very same mechanism, only with a change in direction. The accuracy can even further be increased by measuring variables from supplementary sensors.

Global Positioning System, although vastly different from the aforementioned sensors, can be used to measure the vehicle's speed, but also provides additional information about the lateral position and direction. Coupled with a map, it could also be used to identify the type of road the vehicle is located on (e.g. motorway, speed road, inner-city road or mountain road). Mohammad, Ali and Ismail [22] propose a system built to detect abnormal driving using GPS. The speed and direction are the variables that were tracked in order to detect irregularities in driving. A total of 8142 units of data were collected over 8 field tests on real roads with different conditions (rain, traffic, day and night). GPS was also the sensor of choice in [23] for speed estimation and detection of turns in combination with cameras. It is evident that GPS is an excellent choice for providing secondary data in order to make the detection mechanism even more accurate.

2.4 Related Work

In this section applications serving a similar purpose will be analysed. This is done for exploration as well as to identify potential issues and positive aspects that will be considered when developing the end product.

2.4.1 Drivelert

Drivelert is an application for Android developed by *Akash Gupta*, available on the Google Play Store, that helps combat the state of fatigue when driving by using an eye-closure detection algorithm. Its detection rate can be adjusted by the user. The app sets off an alarm whenever it detects closure of the eyes for longer than the specified by the driver rate. After testing, it was found that the detection mechanism works accurately when pointed directly at the face and at slight angles, but puts out many false alarms in case of slight deviation in the environment, such as light, rapid movement, and others.

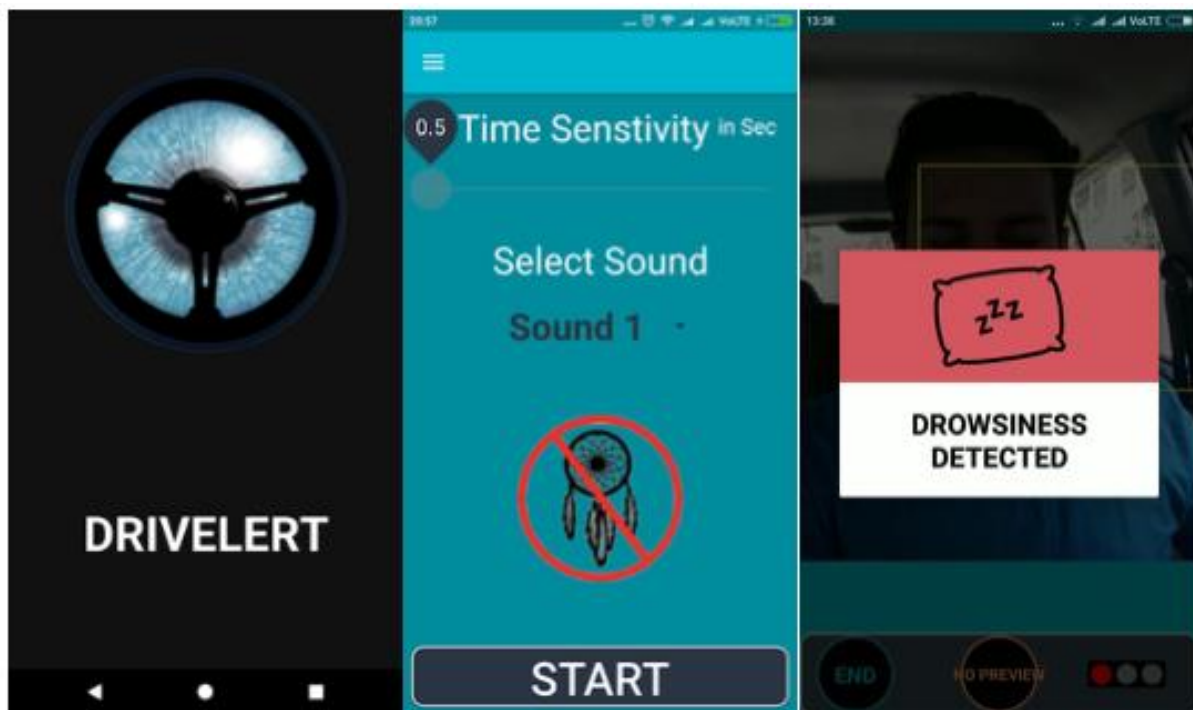


Figure 1: DRIVERLERT Application

2.4.2 DriveAwake

DriveAwake is an application for Android developed by 1Moby Co, Ltd. It uses the same principle as Drivelert, namely the detection of eye closure. It has several additional features, such as Night Mode and most importantly - shows a map with the nearest cafe after an alarm has been triggered. The interface of the application, however, is very outdated and the map function is only designed to work with a particular cafe, which means it is limited to certain parts of the world.

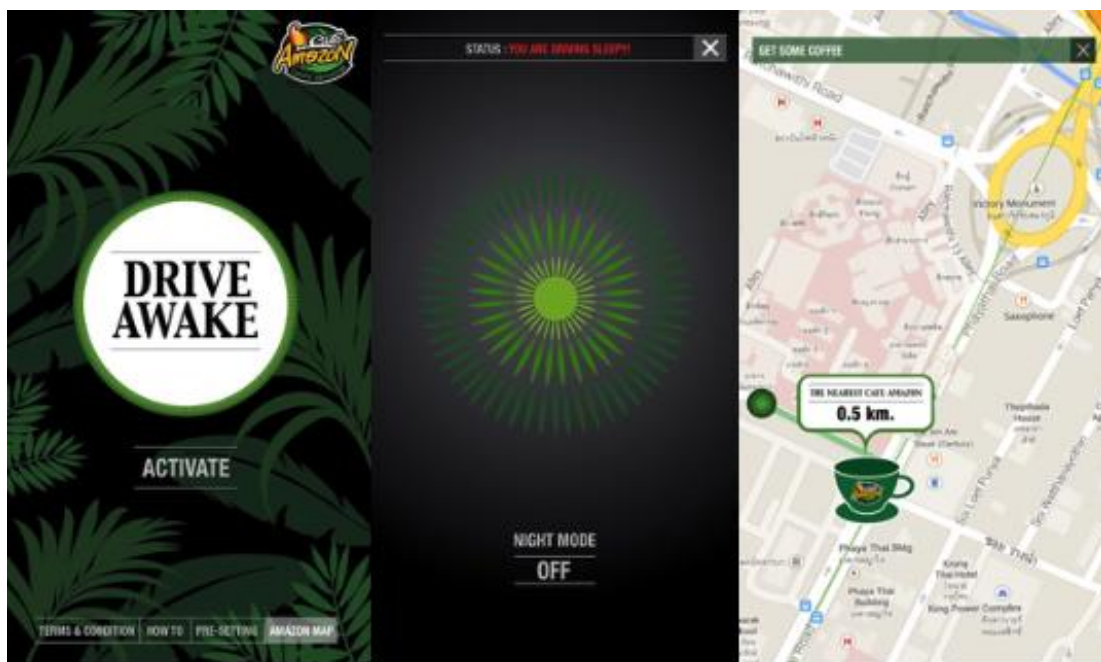


Figure 2: DriveAwake Application

2.5 Conclusion

This review explored the current work done on the topic of detection and prevention of drowsiness in drivers, Fatigue, being a complex state to measure, is divided into categories depending on the inducing factors. It was identified that technological measures are best suited for prevention task-related fatigue since sleep-related fatigue is dependent on the individual.

Thereafter, different kinds of prediction methodologies are proposed according to the symptoms. In this second part, where fatigue prediction was discussed, the techniques were classified into performance and state measures, as well as a combination of both. This is crucial when fine-tuning the model of the prevention system, since different variables may have higher or lower weights in the detection algorithm. The variables that are best suited for the project according to the evidence are time of day, driving duration, hard-braking and possibly speeding.

The third part reflects on the versatility and feasibility of using a smartphone as a detection platform. Because of housing a number of sensors and being popular amongst people, contemporary smartphones are a promising platform to use. The sensors that were discussed together with the symptoms they can be used to detect are provided in *Table 1*.

After analysing the evidence, the most appropriate combination to use for this project was identified as accelerometer, gyroscope, magnetometer, and GPS. This combination has been shown to provide good accuracy, low intrusiveness (the phone does not necessarily need to be mounted in a steady position), which is the primary goal of the application. Camera, although providing an accurate measurement, resulted in false triggers after slight deviations in the environment. Further research is required in order to make the camera a feasible sensor to use and solve the problems of privacy and false alerts resulting from a change in outside factors.

The existing products mostly use the phone's camera or external implementations of the accelerometer, gyro, magnetometer, and GPS instead. What this project aims to do differently is to provide a product that solely relies on the mobile phone for measurements, without using the camera. What is more, the existing applications focus completely on functionality, leaving the user interface and design behind. For this project, the goal is to provide a fully-fledged product that is designed with the user in mind and provides decent functionality.

Symptom	Sensor				
	Accelerometer	Gyro	Magnetometer	GPS	Camera
Eye Closure					x
Head Nod					x
Sharp Steering	x	x*	x*		
Hard-Braking	x	x	x		
Speeding				x	

*can be measured solely by accelerometer or in combination with gyro and magnetometer

Table 1. Sensors and the Symptoms they are used to measure

3. Ideation

The ideation chapter aims to identify the design principles and guidelines prior to developing the mobile application. It provides an overview of the early ideas and considerations that will be taken into account during the implementation phase.

3.1 General Considerations and Requirements

To define the general criteria for the realization phase of the application, a list of considerations and requirements was constructed. It depicts the main aspects and functionalities of the application design, tracking, and data handling.

The application interface must:

- IR1.** *Be designed in a driver-oriented manner*
- IR2.** *Be non-intrusive*
- IR3.** *Store and send information in a safe manner*
- IR4.** *Provide details back to the user after use*
- IR5.** *Provide options for user configuration of the detection mechanism*
- IR6.** *Have versatile ways for collecting user feedback*
- IR7.** *Remind the driver to use the app*
- IR8.** *Ask for consent whenever data is sent to an external server*

The background services must:

- SR1.** *Track deceleration (hard-braking) using the accelerometer, gyroscope, and magnetometer*
- SR2.** *Track current and average speed using the GPS*
- SR3.** *Track elapsed time*
- SR4.** *Track time of day*
- SR5.** *Calculate the tiredness level based on the previous variables*
- SR6.** *Alert the user in case of tiredness level higher than a pre-defined cut-off value*

3.2 The User Interface

In this part, implications about the possible user interface of the mobile application are considered. Mock-up designs were also produced using Adobe Photoshop, which will later guide the implementation process. The full list of mock-ups can be found in Appendix A: Initial Mock-ups.

3.2.1 The “Now Tracking” screen

This will be the screen the user will see while the application is running. The crucial aspects here are to provide sufficient information, without distracting the driver. A short, but concise message that reminds the user how to use the app should be present. Additionally, elapsed time since start may be displayed, together with the number of detections.

3.2.2 Pop-Up Alerts

Whenever the mechanism detects a fatigue level higher than a predefined value, a pop-up alert is triggered. Its purpose is to catch the attention of the driver in the background and notify them of apparent tiredness. The main features include an alert sound, together with a short suggestion to take a break. Additionally, the pop-alert may have a button that gives direction to the nearest coffee place.

3.2.3 Statistics Panel

The function of the statistics panel will be to provide an overview to the driver about their previous driving sessions, in the hopes to improve their driving performance and for comparison.



Figure 4. Mock-up of "Now Tracking"

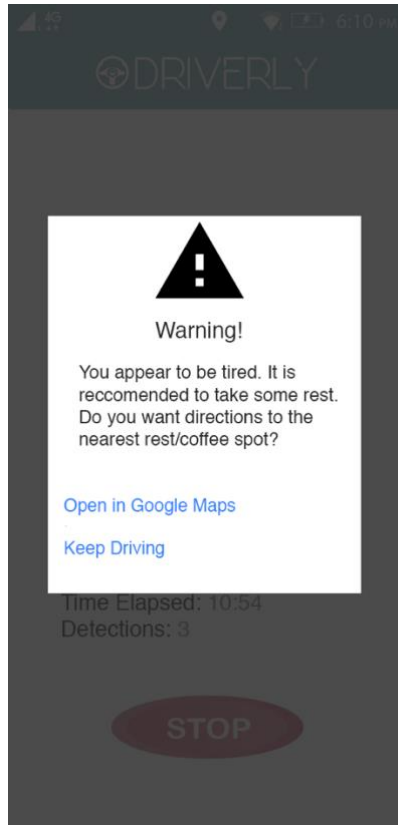


Figure 5. Mock-up of an alert

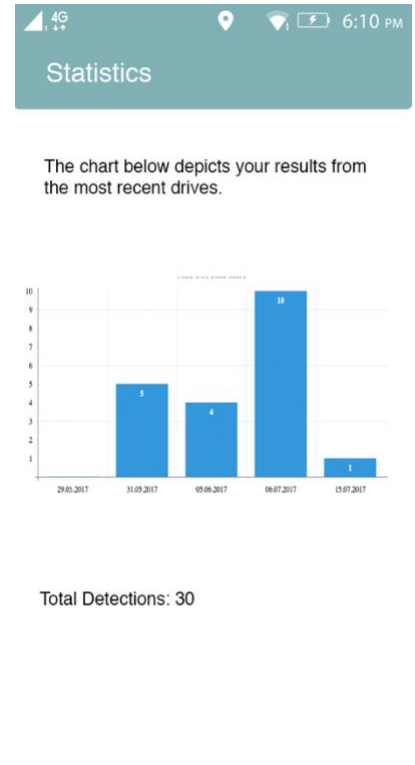


Figure 3. Mock-up of "Statistics"

3.3 The Detection Algorithm

To provide reasonable accuracy, different methodologies of detection should be explored and assessed. A model that would indicate the probability that one is tired is a substantial part of the detection method and will thereafter determine the cut-off score value at which the driver is alerted.

3.3.1 The Prediction Model

The first step of the project execution consisted of designing a model for accurate prediction of fatigue. During the ideation phase, four main independent variables were selected for tracking: hard-braking events, time of day, driving time and sudden speed changes. Additionally, fatigue is defined as a state that either happens or does not (a driver is either tired or is not) and its probability should be determined by the taken measurements. Therefore, a model that fits these specific needs has to be selected.

3.4 Ethical Implications

To ensure that users are satisfied and use the application frequently, ethical considerations need to be made aside from the technicalities. They serve as a separate guideline that will be used when developing both the user interface and tracking mechanism, identifying potential failure scenarios to be avoided before the realisation phase.

3.4.1 Performance Tracking

A possible ethical issue is the fact the software tracks all those variables mentioned before. The user might feel like they are constantly being watched for their driving performance and as a result not feel comfortable using it. Because GPS calculates average speed by the difference between two locations and time, it would be assumed that the application is also tracking location which is a potential problem as well. Drivers might be worried that the data may be unintentionally used for tracking where they go, how fast they drive, for how long they stop and others.

3.4.2 Privacy and Security

As well as monitoring the behaviour of the driver, the application stores that information to build statistical graphs that would afterwards provide an overview of the detections. This brings up a well-known ethical concern of privacy and security. The information needs to be properly stored and not exposed to others. The graphs themselves need to be designed in such a way

that the user does not feel intimidated in case of a high number of detections, but rather reflect on what to improve in his driving behaviour.

3.4.3 Server Storage and Informed Consent

Last but not least, sending the data to an external server might be an ethical concern for the users as well. Even though the purpose of the project and the data that is being recorded are explicitly stated, the users do not see the process because it happens in the background and might be concerned whether the application truly sends what is meant to be recorded. Finally, some people might be worried about the data being used for purposes different than research. The app must be designed in such a way that it asks for the user's consent at the end of every session whether they agree to send their data to an external party, which serves as a solution to this ethical issue.

4. Realisation

This chapter provides concise information on the project execution, including the fatigue model that is used for calculating the probability of the user being tired, the mobile application development phase and feedback collection and processing.

4.1 Product Development

The application will be developed for Android OS using mainly Android Studio, an Integrated Development Environment for Google's operating system. Android provides several crucial advantages over iOS development, such as quick approval and upload times to the Google Play Store as well as a wide range of available libraries and customisation options.

4.1.1 The Logistic Regression Model

Before developing the application itself, It was determined that a Binary Logistic Regression model would be best suited for the model requirements mentioned in the ideation phase. Logistic Regression is a classification method that determines the probability of an outcome that is dichotomous (binary). Predictor variables that preferably have little to no correlation between each other are used to determine this output, with their respective weights. A linear relationship is assumed between those predictor variables:

$$\ell = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3. \dots$$

Equation 1. Relationship between Predictor Variables

The odds are then recovered by exponentiating the log-odds:

$$\frac{p}{1-p} = b^{\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3 \dots}$$

Equation 2. Recovering the odds

Which leads to the final equation that produces the probability that Y=1:

$$p = \frac{1}{1 + b^{-(\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3 \dots)}}$$

Equation 3. The final equation for the probability of Y=1

Initially, a logistic regression code was developed in Java and it evaluated the score on the mobile phone itself. This, however, was not optimal. It contributed to a degrade in performance since logistic regression requires high iteration rates to produce an accurate result. Since the training data is not dynamic, this implementation was also wasting resources which would mean a higher battery discharge rate for the user. Therefore, it was decided to train the model offline.

4.1.2 The User Interface

The user interface was designed on the base of the requirements and mock-ups presented in the ideation phase. It includes 6 different screens in total plus a splash screen during launch, providing various functionalities. They were all designed on the foundation of Android Material Design, which provides a simple, yet multifunctional theme.

The main screen is there to navigate the user to the different options available to the user, as well as starting the tracking process. When the application is launched for the first time, three pop-up dialogs appear: first, to ask the user for the necessary permissions (GPS and storage access), second to inform the driver that this application is a part of a research project and third to ask for their typical time for driving, which will be used to set reminder notifications at the selected period.

The “Now Tracking” screen, initiated by the green start button on the main screen, is the activity the driver sees when the application is working. Features include a short message to let the user know the app is running, elapsed time chronometer, detections counter information about the current state of the GPS tracker. An additional feature is a dynamic line graph, representing the user’s current score. Its purpose is to let the driver know, at a glimpse, whether they are currently close to being fatigued. In case the driver had detection during the session, a pop-up dialog is displayed when they try to exit this screen, asking for their consent for sending the data to an external server.

The alert dialogs, that get triggered when the fatigue score is higher than the predetermined value, consist of two parts. First, the user is presented with a standard Android dialog with a concise message, letting them know that they may be tired and suggesting to take a break. The message is randomly selected from an array of 5 strings for versatility. This dialog and has two buttons, letting the user ignore the warning or give them directions to the nearest coffee place. Together with the appearance of this dialog, a short 3-second ringtone is played to catch the driver's attention. If the user does not respond to this dialog, it is automatically cancelled in 50 seconds. The consequent alert is considered of higher importance and therefore a song is played instead of the ringtone (user-configurable in the Settings panel). In

the case when the person responds to the alert, they are also asked to determine the accuracy of this detection. This will later be used to improve the accuracy of the model, together with the data recorded. Because the user will be driving at the time of appearance, it is made clear and as least intrusive as possible to select their desired answer by using three different smiley faces that are easily distinguishable and correspond to the following options: Not tired (Red), A bit tired (Yellow), Indeed tired (Green).

The Settings panel, accessible from the menu button on the main screen provides the user with various configurations that alter the experience. The available options for customization are:

- SO1.** Detection Sensitivity – allows for incrementation of the cut-off value at which an alert is triggered
- SO2.** Second Alert – lets the person choose what happens to the consequent alert when they do not respond to the first one
- SO3.** Night Mode – a toggle switch which inverts the colours of the theme, allowing for a more night-friendly experience with a dark background and white text, crucial for night drivers and is becoming a standard feature for recent application development
- SO4.** Reminder Notifications – allows the user to enable/disable the notifications that remind them to use the application, as well as choose the specific time at which they will be broadcasted.

The Statistics panel, accessible from the main screen, provides statistical information about the 5 most recent drives of the user in the form of a combined Bar and Line chart. For this function to become available, the person needs to have used the application at least once, driving for longer than a minute. The bars in this figure represent the driving time and are bound to the left axis. The overlaying line chart stands for the number of detections at a specific period. Both are bound to the same X-axis variables, labelled according to the driving session, which consists of five instances with a time stamp on top. The library that was used to generate this graph is MPAndroidChart, version 3.1.0.

Other parts of the interface include a Feedback screen, where the user may directly share their problems and ideas for improvement, as well as a static “Help” activity, which serves the purpose of explaining how to use the application properly, in case anyone experiences difficulty setting it up.



Figure 6. Navigation Diagram of the Application

4.1.3 The Background Services

To measure hard-braking events (deceleration), a combination of accelerometer, gyroscope, and magnetometer is used. The Android SDK conveniently provides a virtual sensor, namely `TYPE_LINEAR_ACCELERATION`. It uses the accelerometer with an added correction from the gyroscope and magnetometer and also subtracts gravity to output the linear acceleration in three axes. Furthermore, the output is also passed through a High Pass Filter in order to filter out insignificant changes. The service class takes the output from this virtual sensor, converts it from m/s to G and after that calculates the total acceleration from the three axes according to Equation 1. A cut-off value of 0.3G is then used to classify a brake as a hard-brake. To avoid false positives, a single addition to the calculated score is considered only when two hard-braking events are detected within a period of 10 minutes.

$$a_{total} = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

Equation 4. Total Acceleration

In order to track the current speed of the vehicle, a separate class was constructed. It uses Android `LocationManager`, which outputs speed calculated by taking the differences between two coordinates every second. This measurement is highly accurate compared to using the accelerometer for speed estimation. The output is converted to km/h and the mean speed for the last 5 minutes is taken using the Apache Commons Math library. Whenever current speed exceeds the average by 40 km/h, a sudden speed change is detected. Again, to avoid false positives, two sudden speed changes are needed within 10 minutes to add an instance to the score calculation. Because Android classifies GPS as a sensible measurement, the user needs to allow the use of this sensor if it is disabled. Therefore, the above functionalities will be ignored if the driver does not allow GPS use when prompted at the start.

As previously mentioned, a measurement of elapsed time and time of day is necessary. A time class was therefore coded implementing the `Chronometer` object available in the Android SDK. It is initiated at the start of the session and visually displays the elapsed time to the user. If the session is longer than 80 minutes, an addition to the score is initiated, according to the research done in Chapter 2. Additionally, the time of day is checked every second. If the current time falls between 20:00 and 08:00, the score gets incremented accordingly. This variable has the highest weight of all, as concluded from the research.

Score calculation is implemented in a singleton class, which means within a single launch of the application, only a single instance of this class exists. This allows the speed, acceleration

and time implementation to only interact with the same instance, avoiding unexpected adjustments. Several methods are implemented, one for configuring the score, second for obtaining the current value and a listener method, which is used to detect when the value has changed. If the score has indeed been altered, and it is higher than the cut-off value, the alert dialogs are triggered in the main class.

4.1.4 Feedback and Data Collection

Obtaining user feedback is crucial for the future development of the application, as it provides real-life information about its functionalities. Therefore, it was decided to offer various options for drivers to give their opinion, contributing to the project. The first method is a simple Feedback screen, located in the options menu on the main screen. It once again explains the purpose and main goal of the project, letting people know that their feedback is valuable. Below, an editable text field is available as well as a Send button, which redirects the user to their email client. The second method uses Survicate SDK. Survicate is an online feedback collection platform, allowing for the implementation of surveys on desktop platforms, mobile phone applications, and websites. A survey of 4 questions was constructed, which appears only once after five launches of the application. An overview of the feedback collection methods can be found in Appendix B: Data Collection, 1. Survicate Survey, 2. Feedback Form.

Data collection would later serve to improve the fatigue model and the detection algorithm. Various information is collected, namely the elapsed time at the time of detection, the number of hard-brakes, the current score, day or night and in case GPS use was permitted by the user – the number of sudden speed changes, mean speed, minimal speed, maximum speed, and the standard deviation. As mentioned previously, the application asks the user for their consent for the recorded data to be sent to a third party. If the user agrees, the session statistics are sent to an external FTP server. If they disagree, the data is stored locally on the phone only.

4.2 User Testing and Evaluation

Following the development stage is the testing phase. The plan is to release and promote the application, having drivers evaluate its functionality and provide data for improving the model.

4.2.1 Initial Release

The application was released for the first time in the Google Play Store on 15th May 2019 by the supervisor of the project – Andreas Kamilaris. Before promoting the application, preliminary testing was done and several crucial disadvantages were found:

- P1.** The application interface was not adaptive – some elements appeared hidden or overlapping on lower-resolution phones
- P2.** On versions lesser than Android 9, the alert dialog was not appearing, even though the detection counter showed an incident
- P3.** The graph in the Statistics panel was showing meaningless data – sessions with a driving time less than a minute (launch and stop) and was considered vague
- P4.** The application did not react whenever the user stops for a break and keeps running nevertheless
- P5.** The application did not consider reminding the users to use it on their drives – risking people downloading and forgetting about it

These disadvantages were considered substantial and it was decided not to promote the product until an update that addresses them is issued.

4.2.2 Updates

Two weeks after the initial release, the application was updated. All of the issues present in the initial release were resolved. After that, a second update was released, improving the functionalities even further. The following changes were present in the second update:

- F1.** A live line graph that displays the user's current fatigue score together with the limit line at which an alert is triggered.
- F2.** If substantial data were recorded during the session, the user is presented with a summary of this last drive, including all the available measurements.
- F3.** Automatic restart of the current session whenever the user stops for a break. It works by waiting 3 minutes of mean speed is less than 5km/h, which indicates the vehicle is not moving
- F4.** The threshold for hard-braking was increased to 0.5G

4.2.3 Promotion

After the first update, the promotion phase was started. The goal is to obtain enough drivers to use the application for a significant amount of data to be recorded. The product was promoted together with the supervisor of the project – Andreas Kamilaris. It was posted on several discussion platforms such as Reddit, using identification tags that would relate to the target group – drivers. It was also promoted within groups of friends and relatives. In the end, this resulted in downloads in the range of 70-100, providing a good base for evaluation of the mechanism and improvement of the model. The install statistics can be found in Figure 7. and Figure 8. Some immediate comments and suggestions were collected including:

- S1.** “The Feedback section could directly upload the form to the server instead of redirecting to an email client”
- S2.** “It seems a bit too easy to trigger a detection”
- S3.** “Also many states are entirely hands free. Instead of just a pop up display can the app send a text message to your phone so that your car would read it out loud to you?”
- S4.** “Very interesting app :) I am curious how the accuracy affects: traffic jams or bad road surface / repairs.”
- S5.** “if you could add accelerometer to a Bluetooth headset to give your app additional inputs of human factors like not turning to milk at mirrors often enough or head droop and other factors that precede falling asleep or indicate drowsiness, it would probably be useful. Is heart rate also a factor? All smart watches can monitor that.”
- S6.** “If I hard brake, it's because something bad almost just happened and I'm full of adrenaline”

Installs by user ?

77 +3,750.00% vs previous period



Uninstalls by user ?

22 +633.33% vs previous period

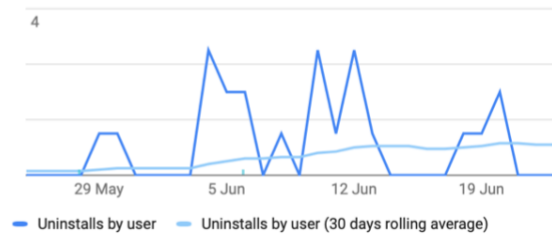


Figure 7. Installs and Uninstalls of Driverly

Installs on active devices, by Android Version

[EXPORT REPORT](#) ▼

Mon, 27 May 2019 – Tue, 25 Jun 2019

Installs on active devices ■ All Android versions ■ Android 9 ■ Android 8.0 ■ Android 7.0 ■ Android 8.1

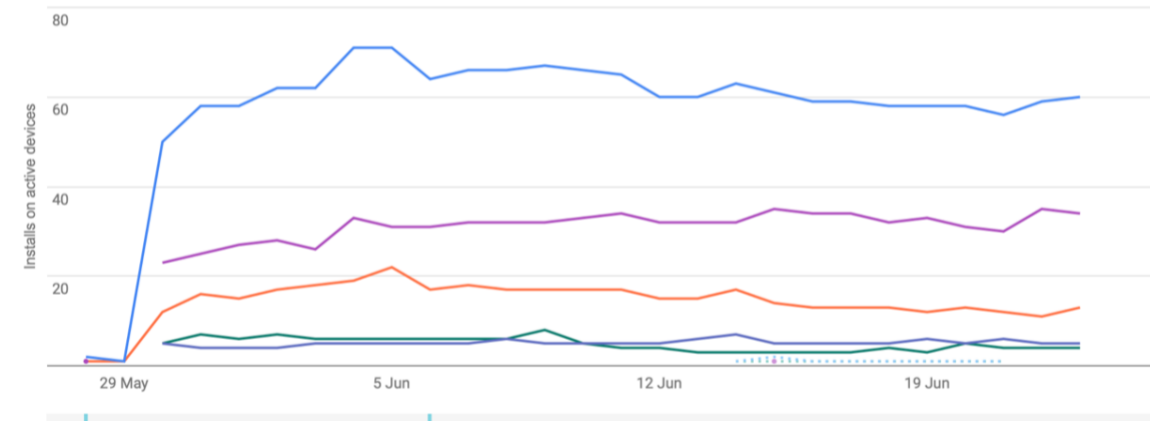


Figure 8. Versions of Android within the testers

4.2.4 Collected Feedback

20 days after the start of the promotion of the application, the data collected was analysed. In total, there were 10 driving sessions uploaded to the FTP server, together with 3 feedback forms. Each session had a number of detections in the range of 1 to 17, resulting in 76 data entries. Out of them, 59 were entries that will be used for training the model. 17 were discarded because of sensor malfunction, which is apparent by the improbably large number of hard-brake detections for the given period.

Preliminary analysis was conducted. Based on the data, the following observations can be made:

- O1.** The average driving time at which detection was triggered was 25 minutes, with the highest being 2 hours and 12 minutes
- O2.** 30 detections were triggered during night-time, while 29 were during the day
- O3.** The mean speed at which detections were triggered was 50.5 km/h
- O4.** The average amount of hard-braking events at the time of detections was 32
- O5.** 2 people confirmed being “Indeed Tired”, while 4 answered with “A bit tired”. 16 reported “Not tired”, however, 37 of the testers did NOT answer this dialog at all.

Stemming from those observations, several conclusions can be made. With the mean speed being 50.5 km/h and the mean hard-braking events 32, one can assume that users mostly drove on non-monotonous roads that are not highways or in cities. This may be the reason for a significant amount of false-positives. Moreover, the majority of the users did not answer the dialog concerning the accuracy, which is normal due to the person driving. A smarter way to collect this aspect of the feedback is necessary. The average driving time at which detections occurred is on the lower side, which is a potential inducer of more false-positives. A dynamic limit could be introduced so that the detections only get triggered after a particular combination of the other variables. The full table with the data can be found in Appendix B: Data Collection, 3. Collected data.

	Mean	Minimum	Maximum	ST. Deviation
Speed	50.5 km/h	34 km/h	67 km/h	10 km/h
Hard-Brakes	32	1	127	32
Driving Time	0:25:08	0:05:14	2:12:30	0:24:16

Table 2. Descriptive Statistics

4.3 Improving the Model

After collecting the data, the fatigue prediction model is to be improved. The way this is intended to be done is by training a logistic regression model using the values of the collected variables as predictors and the answer of the users as the outcome variable. First, the data was cleaned up, removing the extra features such as the standard deviation, minimum and maximum speed, and the score. Labels were also discarded. What is then left is an array of Elapsed Time, Day/Night, Sudden Speed Changes, Hard Brakes and Outcome. After that, the dataset was moved to an Excel spreadsheet, where extra calculations such as the Logit, Probability, and Log-Likelihood were made. In the beginning, a value of 0,001 for all coefficients (weights) was inputted, which is arbitrary. These values are to be later calculated using the Solver add-in method, which implies maximising the Log-Likelihood sum of all entries based on changing those coefficients. Table 3. represents the values found for the weights after running the steps mentioned above.

What can be concluded from the improvement of the model is that some variables are of lesser importance than what was initially found in the research stage. The time of day had a substantially smaller weight on producing a positive trigger, as well as the hard-braking events. Sudden speed changes, however, had a bigger impact, while the elapsed time remains relatively unchanged. These findings can be used to estimate more accurate weights within the application mechanism itself, reducing the number of false positives.

β_0 (constant)	β_1 (Elapsed Time)	β_2 (Time of Day)	β_3 (Sudden Speed Changes)	β_4 (Hard-Brakes)
0,490341252	0,127958	0,00663781	0,17440006	0,0310538

Table 3. Logistic Regression Calculated Weights

5. Discussion

The goal of this chapter is to provide an overview of the project execution, the advantages, and disadvantages of the used methods, as well as to what extent the main goal of the project was reached. Additionally, recommendations for future development are provided.

This project aimed to provide a non-intrusive product that detects and prevents fatigue in drivers. The interface of the application was built with ease-of-use in mind and can be considered successful. Apart from some issues with scalability in the initial release, there were no complaints about the design and features like the dark mode were appreciated. Features like the automatic restart when the user stops for a break were mistaken for a crash as well, which led to adding explanatory labels whenever it is activated. Plenty of configuration options were available and it was noticed that drivers made use of them.

The main issue with the application was false positives, although a mechanism to prevent false triggers was implemented in the form of requiring two hard-brakes or sudden speed changes within 10 minutes. This issue was present with the deceleration tracking in particular, which may be considered a disadvantage of the sensor itself. It may get triggered by simply lifting the phone since forces are applied on the accelerometer. To avoid this, the sensor needs to be steadily mounted at all times, which is not possible with a mobile phone. False positives were also present in situations where the driver has to step on the breaks more often (e.g. on a downhill drive).

After collecting data from testers, it was concluded that most users drove in a non-monotonous setting, which may have contributed to a higher amount of false-positives. Therefore, it is interesting to look into developing methods for the classification of a scene as monotonous or not and adjusting the detection mechanism accordingly. This will ensure that the user gets less untrue triggers as well as provide more accurate data for training the model.

Using the existing data, a logistic regression model was trained in Excel. A substantial difference was found between the weights of the variables that were initially used (based on the research done in Chapter 2) and the outcome after calculation. Sudden speed changes and Elapsed time had a higher effect on an indeed tired result, while hard-braking events and time of day were not major contributors. Variable weights in the application were changed accordingly.

5.1 Future Work

To improve the accuracy of the method, additional inputs may be implemented. The microphone can be used to detect patterns in the cabin, such as yawning, having conversations with passengers or the loudness level of the radio. Localisation may be implemented to increase the number of potential users along the way. This includes translation of the user interface, including region-specific functionalities and parameters.

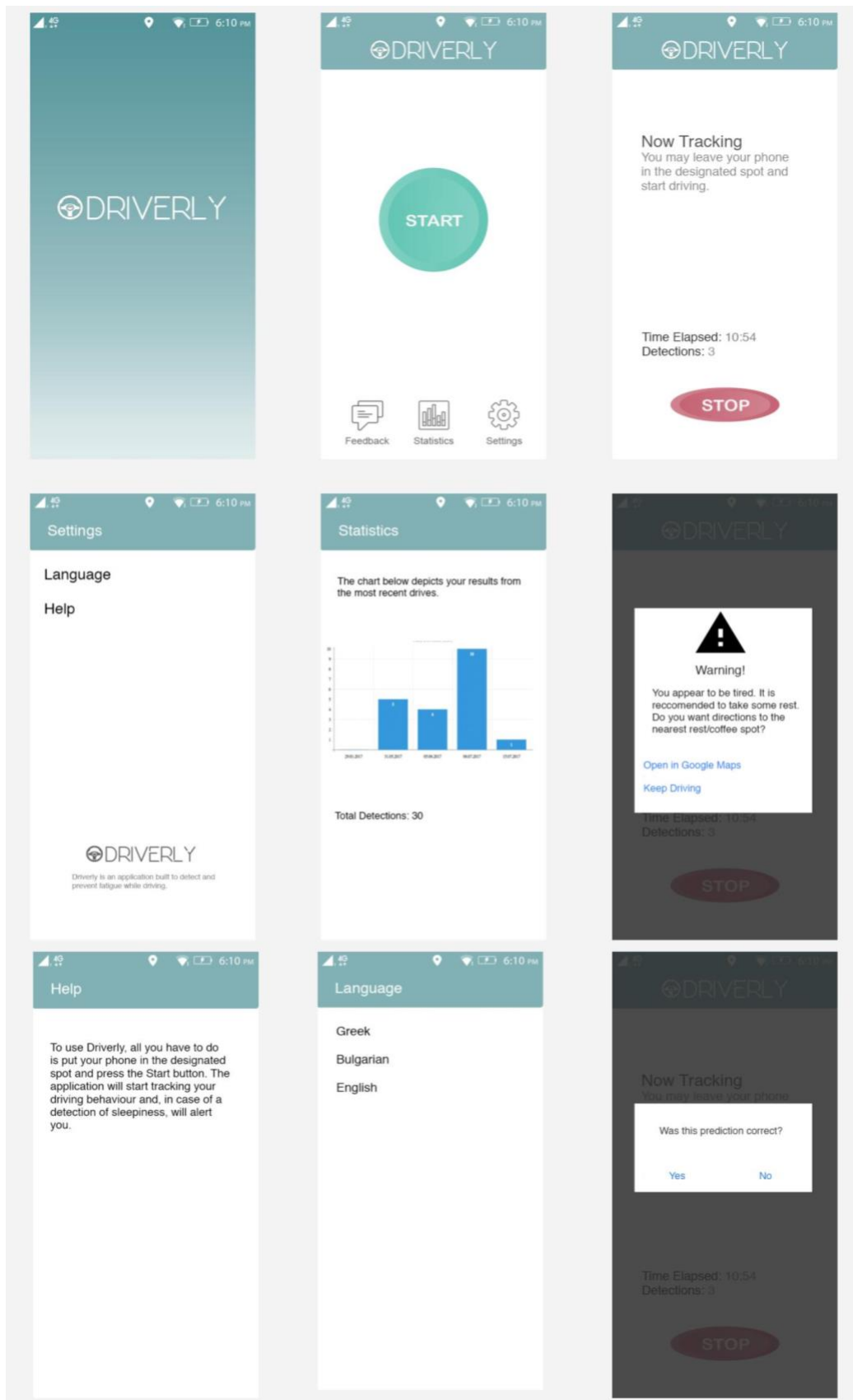
Additionally, the speed limit can be used as a possible indicator of driving style and therefore performance. There are readily available APIs like the Google Roads which would allow for speed limit tracking. The application functionality can then be even further increased by providing not only tiredness tracking but also unsafe driving style alerts.

Camera, although considered intrusive within this project, may be implemented in a way that the users themselves choose to have this option enabled, provided that the phone is mounted in an appropriate position. It was concluded from the research that this method provides good accuracy when environmental conditions were appropriate, so further filtering of false positives will be necessary.

In case that the vehicle is equipped with Android Auto, the loudness of the music or radio could be increased as a consequence of getting an alert. As of now, Android Auto provides limited automation and control over the vehicle's settings. However, at Google I/O 2019, a revamped version of the software was presented which will be even better integrated with newer vehicles. In the case that this gets developed completely, it may be possible to also work with changing the climate control options, rolling down a window and implementing an even better way to handle alerts.

A "serious game" implementation could be made using the statistics panel to provide an active comparison with other drivers, providing a measurement that could trigger changes in driving behaviour.

Appendix A: Initial Mock-ups



Appendix B: Data Collection

1. Survicate Survey

Question 1: Smiley Scale	<i>How would you rate your experience with this application?</i>	<ul style="list-style-type: none">○ Extremely unsatisfied○ Unsatisfied○ Neutral○ Happy○ Extremely happy
Question 2: Text Answer	<i>Did you notice any bugs or issues?</i>	(Answer)
Question 3: Text Answer	<i>What problems did you encounter?</i>	(Answer)
Question 4: Text Answer	<i>Do you have any ideas for improvement?</i>	(Answer)

2. Feedback Form

Driverly is an application built as a part of a research project on how to use smartphones in order to prevent driving fatigue. Any input as to how well the application serves its function, the easibility of use or intrusiveness will be of great benefit and will be used to improve the quality of the product.

(Enter Answer)

3. Collected Data

ET	ToD	SDNSPCH	spSTDEV	spMIN	spMAX	HB	Tired?	Score
0:00:00	Day	0	7	91	111	5	No Answer	6
0:05:14	Day	2	23	0	63	11	No Answer	9
0:05:26	Day	0	2	88	98	4	No Answer	8
0:05:26	Night	0	0	112	114	3	Not Tired	8
0:05:43	Day	2	7	38	63	30	No Answer	16
0:05:48	Night	0	0	111	114	6	Not Tired	8
0:06:04	Night	0	0	111	114	7	Not Tired	10
0:06:12	Night	0	11	0	41	2	Indeed Tired	8
0:06:25	Day	0	11	29	64	13	No Answer	8
0:06:28	Day	2	8	23	52	48	No Answer	38
0:06:56	Day	0	13	29	69	21	No Answer	20
0:07:00	Day	0	18	32	91	8	No Answer	6
0:07:09	Day	2	10	23	57	55	No Answer	36
0:07:47	Day	0	25	0	70	28	No Answer	16
0:07:53	Day	2	7	24	51	58	No Answer	14
0:07:58	Day	0	20	0	59	31	No Answer	14
0:07:58	Day	0	6	34	56	33	No Answer	8
0:07:58	Day	0	10	22	59	40	No Answer	8
0:07:58	Day	0	7	22	63	43	No Answer	8
0:07:58	Day	0	12	19	59	49	No Answer	8
0:07:58	Day	0	20	0	50	53	A bit tired	10
0:11:00	Day	0	10	83	110	5	A bit tired	6
0:12:12	Night	14	8	85	120	2	A bit tired	9
0:12:21	Day	3	16	0	45	66	No Answer	8
0:12:39	Night	14	1	111	116	4	Indeed Tired	8
0:13:31	Day	3	12	0	45	77	Not Tired	16
0:14:25	Day	3	17	0	45	81	No Answer	22
0:14:56	Day	12	17	0	56	1	No Answer	8
0:16:04	Day	3	7	0	23	91	No Answer	8
0:16:58	Day	3	12	0	35	97	Not Tired	20
0:17:42	Night	0	0	112	115	11	Not Tired	8
0:17:53	Night	0	0	112	115	12	Not Tired	10
0:22:16	Night	0	9	39	80	17	Not Tired	8
0:22:25	Night	0	13	31	80	18	Not Tired	12
0:22:49	Night	0	15	28	80	22	Not Tired	8
0:23:10	Night	0	14	15	74	24	Not Tired	10
0:23:34	Night	1	20	15	93	24	No Answer	8
0:25:05	Night	1	13	29	79	31	Not Tired	8

0:25:45	Night	1	18	12	70	34	Not Tired	16
0:26:04	Night	1	6	5	33	35	Not Tired	10
0:27:13	Day	0	18	22	81	9	No Answer	8
0:29:23	Day	18	20	0	47	2	A bit tired	13
0:29:42	Day	3	21	6	88	9	No Answer	8
0:32:37	Day	7	17	1	57	9	No Answer	8
0:33:38	Day	20	22	0	62	4	No Answer	8
0:34:58	Day	8	14	0	48	26	Not Tired	8
0:36:49	Night	29	3	100	110	4	No Answer	23
0:48:47	Night	14	1	111	114	8	No Answer	8
0:48:47	Night	14	1	110	114	9	Not Tired	10
0:55:27	Night	14	0	111	113	11	No Answer	8
0:55:41	Night	14	5	0	16	33	No Answer	8
0:55:17	Night	14	1	0	5	50	No Answer	46
0:55:27	Night	14	0	0	1	67	No Answer	38
0:55:23	Night	14	0	0	1	86	No Answer	38
0:55:45	Night	14	0	0	0	103	No Answer	42
0:55:51	Night	14	0	0	0	120	No Answer	38
0:55:32	Night	14	0	0	2	127	No Answer	38
1:36:42	Night	29	7	88	112	5	No Answer	9
2:12:11	Night	29	21	0	57	5	No Answer	8

References

- [1] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, and M. E. Lopez, "Real-Time System for Monitoring Driver Vigilance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 63–77, Mar. 2006.
- [2] R. R. Knippling and J. Wang, "Crashes & Fatalities Related To Driver Drowsiness/Fatigue," 1998.
- [3] S. K. L. Lal and A. Craig, "A critical review of the psychophysiology of driver fatigue," *Biological Psychology*, vol. 55, no. 3, pp. 173–194, Feb. 2001.
- [4] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 234–239.
- [5] A. Smith, "Smartphone ownership–2013 update" Washington DC, Jun. 2013.
- [6] D. Mollicone et al., "Predicting performance and safety based on driver fatigue," *Accident Analysis & Prevention*, Apr. 2018.
- [7] J. F. May and C. L. Baldwin, "Driver fatigue: The importance of identifying causal factors of fatigue when considering detection and countermeasure technologies," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 12, no. 3, pp. 218–224, May 2009.
- [8] A. I. Pack, A. M. Pack, E. Rodgman, A. Cucchiara, D. F. Dinges, and C. W. Schwab, "Characteristics of crashes attributed to the driver having fallen asleep," *Accident Analysis & Prevention*, vol. 27, no. 6, pp. 769–775, Dec. 1995.
- [9] P. A. Desmond and P. A. Hancock, "Active and passive fatigue states.," 2001.
- [10] P. T. Gimeno, G. P. Cerezuela, and M. C. Montanes, "On the concept and measurement of driver drowsiness, fatigue and inattention: implications for countermeasures," *International Journal of Vehicle Design*, vol. 42, no. 1/2, p. 67, 2006.
- [11] M. Gastaldi, R. Rossi, and G. Gecchele, "Effects of Driver Task-related Fatigue on Driving Performance," *Procedia - Social and Behavioral Sciences*, vol. 111, pp. 955–964, Feb. 2014.
- [12] E. A. Schmidt, M. Schrauf, M. Simon, M. Fritzsche, A. Buchner, and W. E. Kincses, "Drivers' misjudgement of vigilance state during prolonged monotonous daytime driving," *Accident Analysis & Prevention*, vol. 41, no. 5, pp. 1087–1093, Sep. 2009.
- [13] Wen-Bing Horng, Chih-Yuan Chen, Yi Chang, and Chun-Hai Fan, "Driver fatigue detection based on eye tracking and dynamic template matching," in *IEEE International Conference on Networking, Sensing and Control, 2004*, Taipei, Taiwan, 2004, pp. 7–12.
- [14] F. Maysoon, A. Hesham, and F. Lamiaa, "Using Mobile Platform to Detect and Alerts Driver Fatigue," *International Journal of Computer Applications*, vol. 123, no. 8, pp. 27–35, Aug. 2015.
- [15] F. Sagberg, P. Jackson, H.-P. Kruger, A. Muzet, and A. Williams, *Fatigue, sleepiness and reduced alertness as risk factors in driving*. Oslo: The Institute, 2004.
- [16] P.-H. Ting, J.-R. Hwang, J.-L. Doong, and M.-C. Jeng, "Driver fatigue and highway driving: A simulator study," *Physiology & Behavior*, vol. 94, no. 3, pp. 448–453, Jun. 2008.
- [17] E. Bekiaris, K. Wevers, and A. Amditis, "Advanced driver monitoring: the awake project," presented

at the *8th World Congress on ITS*, Sydney-Australia, 2001.

[18] B.-G. Lee and W.-Y. Chung, "A Smartphone-Based Driver Safety Monitoring System Using Data Fusion," *Sensors*, vol. 12, no. 12, pp. 17536–17552, Dec. 2012.

[19] Nidhi Kalra and Divya Bansal, "Analyzing driver behavior using smartphone sensors: a survey," *International Journal of Electronic and Electrical Engineering.*, vol. 7, no. 7, pp. 697–702, 2014.

[20] S. Lawoyin, X. Liu, D.-Y. Fei, and O. Bai, "Detection methods for a low-cost accelerometer-based approach for driver drowsiness detection," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, San Diego, CA, USA, 2014, pp. 1636–1641.

[21] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," presented at the *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 1609–1615.

[22] I. Mohamad, M. A. M. Ali, and M. Ismail, "Abnormal driving detection using real time Global Positioning System data," in *2011 IEEE International Conference on Space Science and Communication (IconSpace)*, 2011, pp. 1–6.

[23] C.-W. You et al., "CarSafe App: Alerting Drowsy and Distracted Drivers Using Dual Cameras on Smartphones," in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, New York, NY, USA, 2013, pp. 13–26.