Bilkent University

Department of Computer Engineering

Senior Design Project

SAVE: safe drive

Project Specifications Report

Elif Demir, Utku Kalkanlı, Büşra Ünver, Celal Bayraktar, Münevver Uslukılıç

Supervisor: Dr. Hamdi Dibeklioğlu

Jury Members: Dr. Can Alkan and Dr. Çiğdem Gündüz Demir

October 12, 2020

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

1.0 INTRODUCTION

Everyday people use transportation to reach from somewhere to another place. After the Covid-19 pandemic, people try to avoid using public transportation. As a result, private vehicle numbers in traffic increased significantly. More individual vehicles mean more errors in traffic. Most road accidents occur due to human error, and one of the most common types of error is driver fatigue. Traffic Accident Commission(TAC) of Australia states that 20% of fatal road accidents involve driver fatigue. Furthermore, 30% of the severe crashes again involve driver fatigue [1]. Additionally, the driver's medical condition is also important in traffic. A study done by the Centre for Automotive Safety Research, University of Adelaide shows that 13% of the crashes investigated are caused by a medical condition [2].

Car companies supply some features in their cars to decrease such accidents caused by human error. However, the cost of those cars is not affordable in general and cars produced previously does not benefit from the latest developments in technology. Considering all these problems, we tried to come up with a solution to decrease human error factors in the traffic and to make the latest driver assistant features available for everyone. As a result, SAVE came up.

In the development process of the idea, we conduct market research on similar products on both vehicle and mobile application markets. After research, we find out there are similar products in the mobile application market like DRIvision [3] and car manufacturers already implement similar driver assistance applications on the vehicles. However, SAVE differs from these applications. Unlike DRIvision, SAVE mainly focuses on the driver along with the road related warnings. Although there are vehicles that provide similar features, we aim to supply these features with a much lower cost to drivers.

As mentioned before, the main focus of the SAVE will be on driver behavior. Additionally, we also provide other driver assistance like the crash emergency message and more. The features of the application will be explained in detail in the following sections.

2.0 PROJECT DESCRIPTION

SAVE is a mobile application project for the Android platform, targeted for drivers to maintain a safe driving experience. In SAVE, the expected features are listed below:

In the first step of the project, the application will have the expected features as follows:

2.1. Fatigue Detection

Fatigue during a long trip can cause accidents and can be easily ignored. SAVE will detect the fatigue of the driver and warn according to their level of fatigue. We want the driver to stay aware and vigorous, the application will constantly watch the driver using the front camera of a smartphone attached to the front window of the car. In a situation where the driver has a level of fatigue that is risky, if the driver is connected to Spotify via Bluetooth, SAVE will play the music that is energetic and uplifting in order to prevent the driver from sleeping or low level of focusing on the road. Also, SAVE will collect data that is specific for a driver, the application will analyze and create a profile for the driver including favorite stopovers, gas stations, hours of driving without being tired, previous speeds done in a road trip, comparing these data to a generalized driver behaviour data. When the driver creates a road trip on Google Maps, our app will propose stopover locations that are suitable for the driver's characteristic, (e.g if a healthy driving period is 2 hours, it will propose a stopover location within 2 hours of reaching time). Proposed stopover suggestions can be controlled by the driver via voice control.

2.2. Health Condition Recognition

Heart rate is an important indicator of one physiological state and drivers can face high levels of stress in traffic that can cause an increased heart rate. In a situation where the health of the driver is endangered, such as abnormal pulse rate, stress level, respiratory rate, etc., the application will warn the user that the situation is unhealthy. In a situation where the drivers' health condition rates are extremely abnormal, SAVE will show appropriate health centers, hospitals via Google Maps and create a route. By voice output, it will ask the driver whether they can drive to these locations if not it may make an emergency call. We plan to use methodologies and techniques described in X.Li, J.Chen, G.Zhao and M.Pietikäinen, in order to measure remotely the heart rate of the driver from the camera by looking at their face [4]. Additionally, we plan to use the work of D.McDuff, S.Gontarek and R.Picard, in order to measure stress levels remotely [5].

2.3. Emotion Detection

In order for them to act with good sense, SAVE wants the driver to stay emotionally as balanced as possible. In cases where the driver is furious, happy, sad, or neutral, SAVE will detect the emotion of the driver by constantly tracking the face of the driver from the front camera. When the driver is furious, it may cause them to make a mistake, so SAVE will open a song that will reduce the temper if the driver is connected to Spotify via Bluetooth. Additionally, we will use the methodologies and pre-trained models described in [6], in order to detect emotions of the driver.

2.4. Collision Emergency Message

The application will measure G-Force constantly, and in a situation where the G-Force changes dramatically, it will predict that a possible collision has occurred, it will ask for a voice response, if no response is gathered it will send a message which contains GPS location to emergency links previously chosen by the driver. We plan to use the Car G-Force Meter API for measurement of g force.

2.5 Additional Features

If we have time after completing our primary goals, we plan to implement following additional functionalities as well.

2.5.1 Traffic Light Assistant

Drivers can miss traffic light changes if they are not focused on the lights constantly. To reduce the workload of the driver, SAVE will track from the rear camera for any changes on traffic lights. In a situation where a change in traffic light occurs, it will give a sound notification to the driver. We are planning to utilize achievements done in Saini, Sanjay [7].

2.5.2 Pedestrian Warning

The application will give a warning to the driver if any pedestrian occurs on the sidewalk. If any pedestrian starts crossing the road, the level of warning will increase.

2.5.3 Distance Tracking Assistant

The application will track the distance between the car in the front and give a warning if the driver is too close than the usual following distance according to the speed of the car.

2.5.4 Distractibility Detection

The attention of a driver is a must on a road trip. By watching the driver from the front camera, the application will check whether the driver is focused on the road or distracted by external features. If the driver is distracted it will give a warning for the driver to focus on the road.

3.0 CONSTRAINTS

3.1 Implementation Constraints

- In order to maintain coordination among project members, GitHub will be used.
- Object-Oriented Programming patterns will be used.
- The client-side of the program will be an Android app and for the UI implementation React Native and Semantic will be used.
- For face recognition, fatigue detection, and remote PPG, mainly Python will be used. In addition to this, for some parts of the implementation of remote PPG C++ will be used.
- Some parts that require higher-level knowledge of machine learning will be implemented with the help of existing models and open-source and third-party libraries such as OpenCV and dlib.
- Some features of our app will be stored and will be ready to work on the client's mobile phones. However, other features that require image processing will require an

Internet connection, since they will be run on cloud servers to decrease CPU and GPU usage.

- For the cloud service, we will be using Amazon Cloud Service EC2.
- The backend of the application will be in Java Spring.
- For the dependency management Maven will be used.
- Android Studio will be used.
- For the road and driver recording, the mobile phone's front and rear cameras will be used.

3.2 Economic Constraints

- For technological equipment such as smartphones and cars, no purchasing will be made.
- In order to place the smartphones inside the car, a phone-holder will be required which costs around 30TL [8].
- We will use free and open-source libraries.
- 25$ will be paid for the Google Play store [9].
- For cloud services, free and limited clouds will be used from Amazon Cloud Service.
- The GitHub and GitHub domain will be free.

3.3 Health and Safety Constraints

- Since SAVE will be used in a moving car, it can cause distraction. In order to minimize this and to make sure that the screen of our app will not distract the driver, we will be using voice command. All the inputs can be given via speaking and the application will read the responses and warnings to the driver.
- SAVE will not be using any sensitive data related to the end-user such as credit card numbers.

- The personal data such as emergency call number and setting preferences will be stored in our encrypted database [10].

3.4 Resource Constraints

- We have two cars to test the application.
- We will use android smartphones to download the application.
- We will use a phone holder to attach the phone to the car.

3.5 Technological Constraints

- The application must be downloaded to an android mobile phone with a well-working front and back camera, speaker, and microphone.
- In order to use some features such as music play, the device must be connected to the Bluetooth of the car.
- In order to use the advanced features the mobile phone must be connected to the Internet.
- Since SAVE requires energy, the user should connect the phone to the car to obtain a better performance.
- SAVE will work on the Android 9.0 and newer.

3.6 Language Constraints

- SAVE will be in English.
- In the future, we are planning to add Turkish.

3.7 Technical Sustainability Constraints

- SAVE will continue to evolve and change according to the user feedback through its lifetime.
- We will be using updated technologies among the open-source ones.

- The application will be compatible with Android devices that can provide technological necessities as stated in technological constraints.

3.8 Environmental Sustainability Constraints

- The application will not require a huge amount of data in terms of the user side. However, we need to use datasets to train our algorithm, which requires cloud storage, and therefore; energy.
- SAVE will be using power; therefore, it needs to be connected to the car's electric supply to provide a better experience. This will also require energy.

3.9 Time Constraints

- The application will be implemented during two academic semesters Fall 2020 and Spring 2021.
- For the first semester, the fundamental features of the application, fatigue recognition, and emotion recognition will be implemented with their UI design and dependencies. Other features will be implemented in the second half of the year.

4.0 PROFESSIONAL AND ETHICAL ISSUES

While identifying the following professional and ethical issues, we follow the ACM Code of Ethics and Professional Conduct and A Roadmap for Ethics-Aware Software Engineering principles:

- According to the Code of Ethics, a computing professional should contribute to society and human well-being [11]. SAVE is designed to serve vehicle drivers' uses of the application, other drivers' in the traffic, and the pedestrians' that may be affected by a traffic accident well-being.
- Code of Ethics suggests that a computing professional should respect the work that is used in the development process [11]. During the development process of SAVE, all articles, libraries, code segments that will be used will be referenced. If required, all permissions will be taken before using in the project.
- Computing professionals should respect the privacy of the users of their products as the Code of Ethics stated [11]. In SAVE, none of the features proposed require any storage of personal data. All personal information given in the sign-up process will be

stored in the application's database within the scope of GDPR and will not be shared with third-party applications or individuals.

- Sustainability [12] is one of the most important ethical issues today when taking into consideration the scarce resources of the earth. Both in the development process and during the usage of application energy consumption will be minimized.
- Accountability [12] issues occur, due to proposed features of the SAVE will be enabled in cases of dangerous situations in traffic. Because all controls of the vehicle are in the hands of the driver and SAVE is just a supportive warning application that has no access to vehicle controls, the driver of the vehicle is accountable for any accident situation.

5.0 REQUIREMENTS

5.1 Functional Requirements

- The application can detect the fatigue of the driver by using the data of driver’s eye-blink rate, mouth expressions such as yawning and head gestures.
- In order to keep users away from the effects of being fatigued while driving, the application can warn the user about being fatigued or drowsy.
- It can show a warning sign and list the nearest gas stations using Google Maps with sound effects after it detects fatigue.
- The application can show an analysis of a user's favorite resting areas and hours of driving without being tired.
- The application can show a warning if there is an unexpected health situation of the user while monitoring the heart rate with remote PPG.
- It can show a warning sign and list the nearest health centers and hospitals.
- The application can detect the emotional situation by facial expressions if the driver looks angry, impatient or sad in traffic.
- The application can play music using Spotify if the mobile phone is connected to the car with Bluetooth. The fatigue detection feature plays energetic lists and the emotional discomfort detection feature play calming lists.
- The application can measure the G Force.
- It can make an emergency call if the driver does not reply to the voice message in a situation that detected an unhealthy situation or dramatic change in G Force constant.

- The application can track the traffic lights and warn the user with sound when the light change occurs.
- The application can warn the user with sound if any pedestrian occurs on the sidewalk.
- The application can calculate the proper following distance and warn the user with sound if it is exceeded.
- This application can detect distractibility if the user is not focused on the road and warn the user with sound.
- The user can create a profile for name, surname, emergency call number and select/deselect the features to be used in the application.

5.2 Non-Functional Requirements

5.2.1 Usability [13]

- This Mobile application should be easy to use and easy to understand how to use its functions to be user friendly. This application should have small and accessible buttons to navigate.
- As the users are drivers, they can interact with this application using buttons that are accessible within 3 seconds or with voice control so that they don't have to view the application screen while driving.
- To not distract the driver application should have warm colors and a night mode that users can choose to use.
- The application should use both front and rear cameras.
- The application should be compatible with all android phones above Android 6.0.0.
- For best performance application should be used with a phone holder stick to the car window.

5.2.2 Reliability [14]

- The application's main functions such as drowsiness detection should be used without internet requirement.

- The application can be used during night driving in a road with street lights

- The application can be used with glasses, the reflection on the glass should be noticed and ignored by the application.

5.2.3 Performance

- The application should react to the user within seconds to inform them of any behaviors that are tracked by the application, it should not be greater than 3 seconds.

- The application should assist the driver all the time during driving.

5.2.4 Security

- The application should protect the user's personal data from any cyber attacks using encryption.

6.0 REFERENCES

[1] “Fatigue statistics,” *Transport Accident Commision*. [Online]. Available: https://www.tac.vic.gov.au/road-safety/statistics/summaries/fatigue-statistics.

[2]T. Lindsay, “Medical conditions as a contributing factor in crash causation.” ResearchGate, May, 2018.

[3] *DRIvision*. [Online]. Available: https://drivision.wordpress.com/. [Accessed: 12-Oct-2020].

[4] X.Li, J.Chen, G.Zhao and M.Pietikäinen. “Remote Heart Rate Measurement From Face Videos Under Realistic Situations”. In *CVPR*, 2014.

[5] D.McDuff, S.Gontarek and R.Picard. “Remote Measurement of Cognitive Stress via Heart Rate Variability”. https://affect.media.mit.edu/pdfs/14.McDuff_et_al-EMBC.pdf [Accessed: 12-Oct-2020].

[6] R.Puri, M.Tiwari, A.Gupta, N.Pathak, M.Sikri, S.Goel. “Emotion Detection using Image Processing in Python”. *5th International Conference on “Computing for Sustainable Global Development”.* 2018.

[7] Saini, Sanjay & Nikhil, S & Konda, Krishna reddy & Bharadwaj, Harish & Ganeshan, N. (2017). An efficient vision-based traffic light detection and state recognition for autonomous vehicles. 606-611. 10.1109/IVS.2017.7995785.

[8] “Araç İçi Telefonu Tutucu,” Hepsiburada. [Online]. Available: https://www.hepsiburada.com/anka-long-neck-uzayabilen-arac-ici-telefonu-tutucu-p-HBV00000QVZH0. [Accessed: 12-Oct-2020]

[9] “Google Play Console,” Google. [Online]. Available: https://play.google.com/apps/publish/signup/. [Accessed: 12-Oct-2020].

[10] “EncryptionUtils,” EncryptionUtils (Spring Security 3.0.8.RELEASE API), 08-Oct-2012. [Online]. Available: https://docs.spring.io/spring-security/site/docs/3.0.x/apidocs/org/springframework/security/util/EncryptionUtils.html. [Accessed: 12-Oct-2020].

[11] ACM Code of Ethics and Professional Conduct https://www.acm.org/code-of-ethics , [Accessed: 12-Oct-2020].

[12] F. B. Aydemir and F. Dalpiaz, “A roadmap for ethics-aware software engineering,” Proceedings of the International Workshop on Software Fairness - FairWare 18 , 2018.

[13] Nielsen, Jakob (4 January 2012). "Usability 101: Introduction to Usability". Nielsen Norman Group. Archived from the original on 1 September 2016. Retrieved 7 August 2016. Available: https://www.nngroup.com/articles/usability-101-introduction-to-usability/

[14] O'Connor, Patrick D. T. (2002), Practical Reliability Engineering (Fourth Ed.), John Wiley & Sons, New York. ISBN 978-0-4708-4462-5.