Bilkent University
Department of Computer Engineering

Senior Design Project

SAVE: safe drive

Project Analysis 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

November 21, 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 a destination. 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 do 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. Asa 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 at 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 Current System

In the mobile application market, there are lots of applications similar to Save but our application differs from others by gathering a lot of features into one application.

- DriveAlert is a mobile application for drowsiness detection however it is required to have a smartwatch to keep track of the heart rate. In Save for keeping track of the heart rate, there is no additional equipment needed thanks to the remote photoplethysmography (PPG).
- Emotimeter is a mobile application for emotion detection.
- UGV Driver Assistant is a more complete program compared to DriveAlert and Emotimeter but unlike Save its main focus is the road. Save focuses more on the driver's behavior while driving.

3.0 Proposed System

3.1 Overview

SAVE will be an Android application for smartphones that will assist drivers. The application will detect drowsiness, heart rate, and emotion of the driver and will send an emergency message in case of a collision. Drowsiness detection system will warn the driver if eyes are closed longer than a blink. Also, the system will track the number of eye blinks and yawns, and heart rate. If there is an abnormal situation, again the system will warn the driver of drowsiness. In addition, if the telephone is connected to Bluetooth and the driver confirms the system will open upbeat music to keep the driver awake. Heart rate will be tracked while driving and 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. Moreover, the application will track emotional state and in case of anger, the application may open a relaxing song to reduce the temper if the telephone is connected to the car via Bluetooth. Another feature of the application will be collision emergency message. By measuring G force in the car during the trip, the system may detect if a collision occurs by enormous change of G force. If the system detects any collision, it will ask the drivers for their situations. If the driver does not respond, the system will send an emergency message to the people on the emergency list which are determined by the users.

Additionally, if we have enough time, we are planning to add pedestrian warning system, traffic light assistant, distance tracking assistant and distraction detection system. Pedestrian warning system will warn the driver if there is a pedestrian on the sidewalk and the warning level will increase if pedestrian jumps onto the road. Traffic light assistant will warn the driver to move if the traffic light turns green. Distance tracking assistant will check the distance between driver and the car in front and if the driver is too close than the usual following distance according to the speed of the car. Distraction detection system will track eye motion of the driver and will warn the driver if the driver is not focusing on the road. All warnings given by the application will be in the form of audio and visual.

3.2 Functional Requirements

- 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 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.

3.3 Nonfunctional Requirements

3.3.1 Usability

- 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.

3.3.2 Reliability

- The application’s main functions such as drowsiness detection should be used without internet requirement.
- The application can be used during night driving on 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.

3.3.3 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.

3.3.4 Security

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

3.4 Pseudo Requirements

- Working environment of the application will be smartphones, thus the amount of space occupied must be sufficient to not block the telephone’s normal operations.
- Since it is not possible to reach internet connection everywhere while driving, driver related features like fatigue detection must work on the phone’s CPU and GPU instead of cloud. In other words, some features must work without internet connection.

3.5 System Models

3.5.1 Scenarios

Scenario #1

Use Case Name	Mute/Unmute Audio Warnings
Participating Actor	Initiated by Actor
Flow of Events	1 - Actor clicks the mute/unmute button on settings menu. 2 - State of voice warnings is changed.
Entry Condition	-The actor is in the settings menu.

Exit Condition	-The update on muting/unmuting the voice alert state is completed.
Quality Requirements	None

Scenario #2

Use Case Name	Adjust Volume
Participating Actor	Initiated by Actor
Flow of Events	1 - Actor drags the sound bar on settings menu. 2 - Level of sound is changed.
Entry Condition	-The actor is in the settings menu.
Exit Condition	The update on adjusting sound level is completed.
Quality Requirements	None

Scenario #3

Use Case Name	Respond Audio Message
Participating Actor	Initiated by Actor
Flow of Events	1 - System asks the state of the driver. 2.1 - Driver responses system does nothing. 2.2 - Driver does not responses system send an emergency message to people on the emergency list
Entry Condition	- The actor has an health issue or get involved in a collision.
Exit Condition	- The actor responds successfully. - The system sends emergency messages successfully.

Quality Requirements	None

Scenario #4

Use Case Name	Login
Participating Actor	Initiated by Actor
Flow of Events	1 - The actor opens the application 2 - The actor enters its username and password
Entry Condition	- The actor is in opening page - Telephone has internet connection
Exit Condition	- The actor successfully logged in its account.
Quality Requirements	None

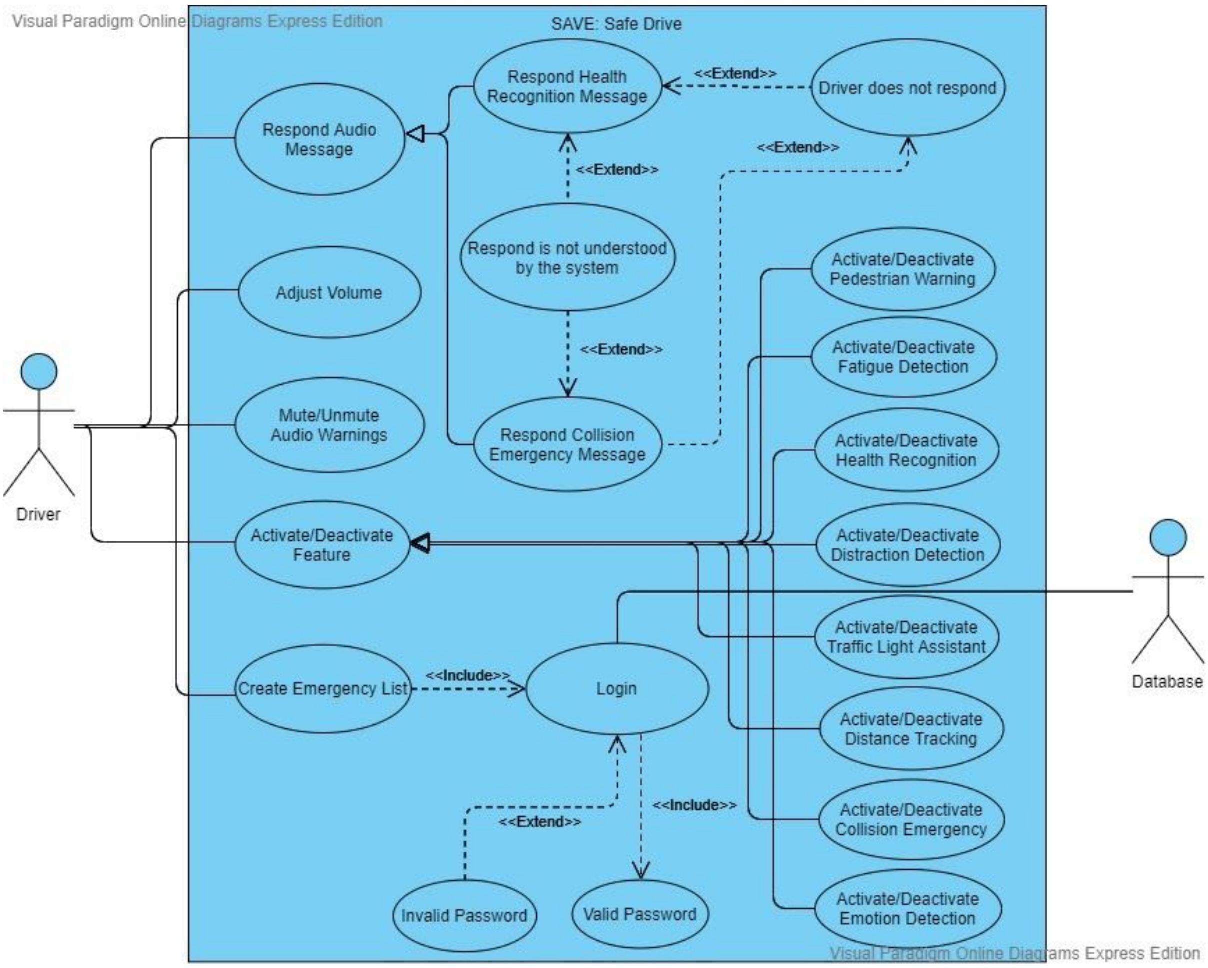
Scenario #5

Use Case Name	Activate/Deactivate Feature
Participating Actor	Initiated by Actor
Flow of Events	1 - Actor changes the state of the switch button near the features on the list. 2 - The state of the feature is changed
Entry Condition	- The actor is logged in - The actor is in settings menu
Exit Condition	- The change in the state of the feature is completed.
Quality Requirements	None

Scenario #6

Use Case Name	Create Emergency List
Participating Actor	Initiated by Actor
Flow of Events	1 - Login to personal account 2 - Actor visits emergency list on the settings menu 3 - Give information of people actor want to add to emergency list
Entry Condition	- The actor is logged in - The actor is in settings menu
Exit Condition	- The update on the emergency list is completed
Quality Requirements	None

3.5.2 Use Case Model



3.5.3 Object and Class Model

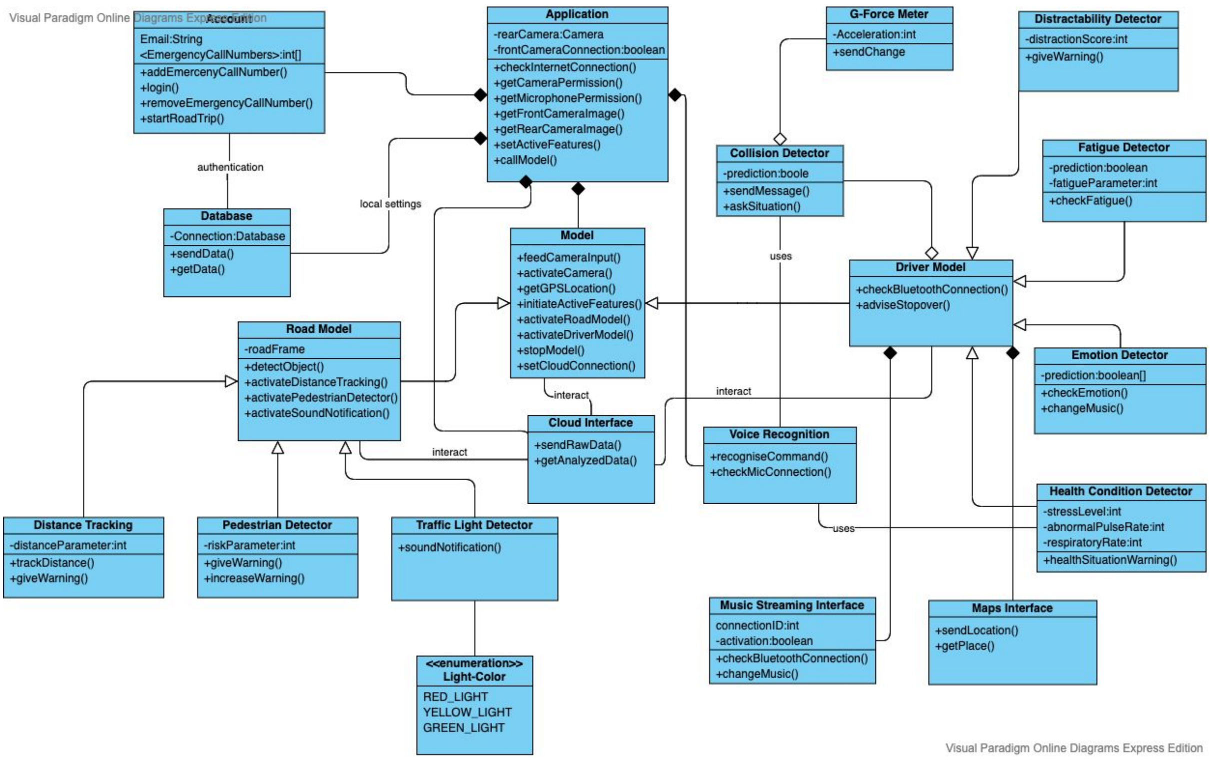


Figure 1: Object and Class Model Diagram

3.5.4 Dynamic Models

3.5.4.1 Sequence Diagram

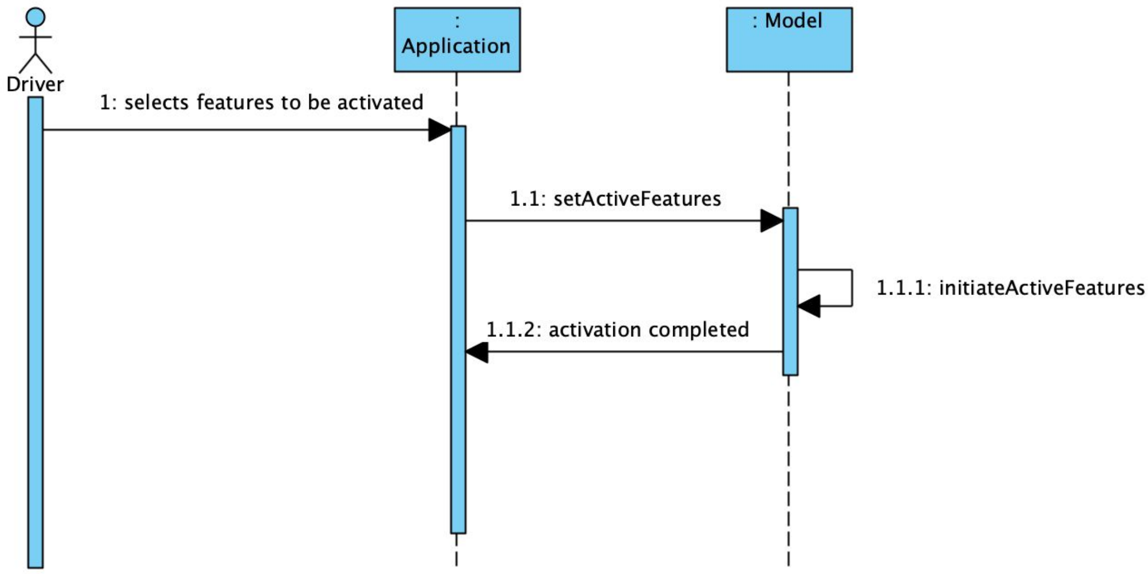


Figure 2: Application Model Sequence Diagram for scenario 1

Scenario 1: Feature Activation/Deactivation
Actor: Driver
Driver visits the settings menu to select the desired features of the application to work or not. Application sets the active features based on the selection of the user. Model initiates only the desired features and returns a boolean to indicate the process is completed.

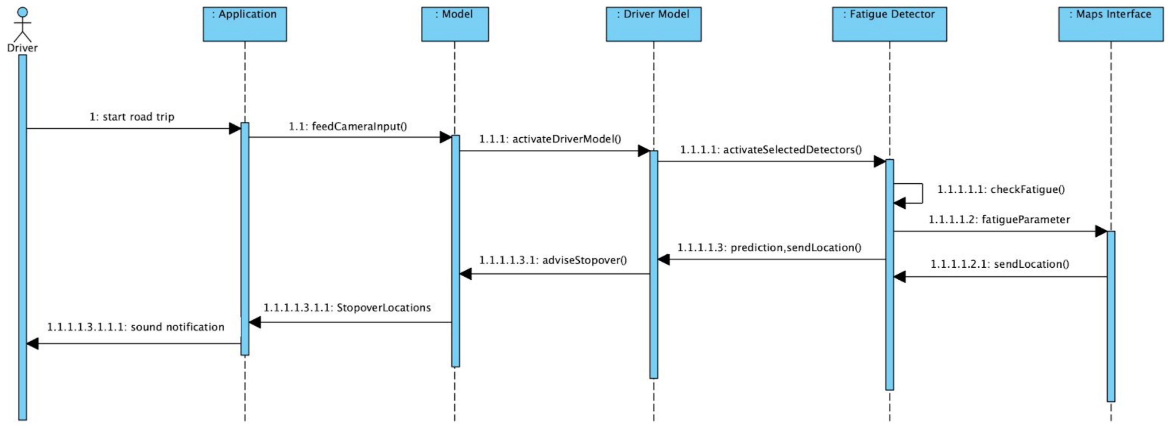


Figure 3: Fatigue Detector Sequence Diagram for scenario 2

Scenario 2: Stopover Location Suggestion
Actor: Driver
After the driver starts the application, if the fatigue detection feature is activated, the application will give camera input to the model. The Driver Model will activate Fatigue Detector class and it will continuously check for fatigue of the driver. Fatigue parameters will be calculated and if the system detects a risky situation, it will connect with Google Maps Interface and take proposed stopover locations from that. Finally, these stopover locations are sent to the driver by voice output.

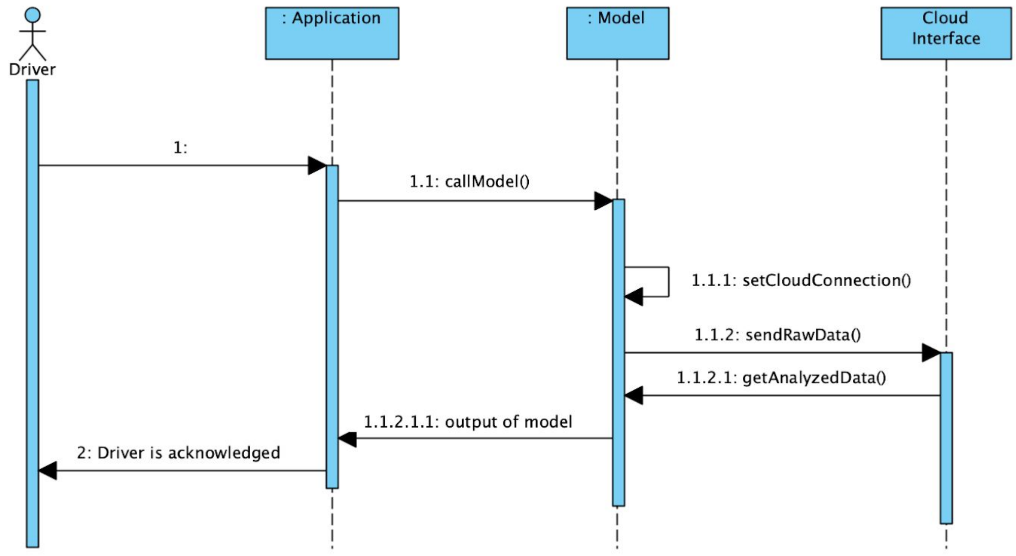


Figure 4: Cloud Interface Sequence Diagram for scenario 3

<u>Scenario 3: Cloud Connection</u>
Actor: Driver
In general, SAVE applications will do a significant amount of image processing and related calculations continuously. To reduce the workload of phones, some computations are planned to be done on cloud. Cloud interface will talk with cloud servers and enable applications to send/receive data back and forth.

3.5.4.2 State Diagram

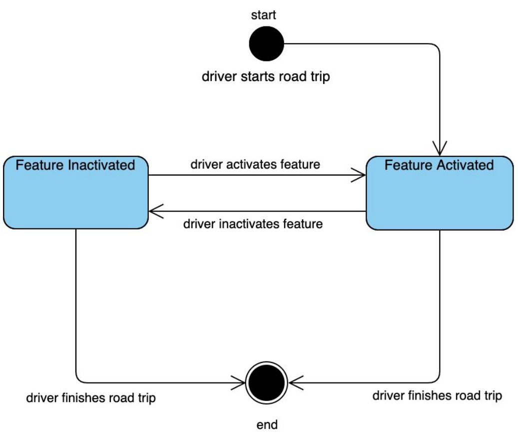


Figure 5: State Diagram

Any feature of a SAVE application has 2 states decided by activation/deactivation settings that are previously selected by the user. When the application is initiated for the first time, by default the features are at active state. If the user disables the usage of a feature from settings page, it switches into the inactivated state and does not take any action.

3.5.4.2 Overall Activity Diagram

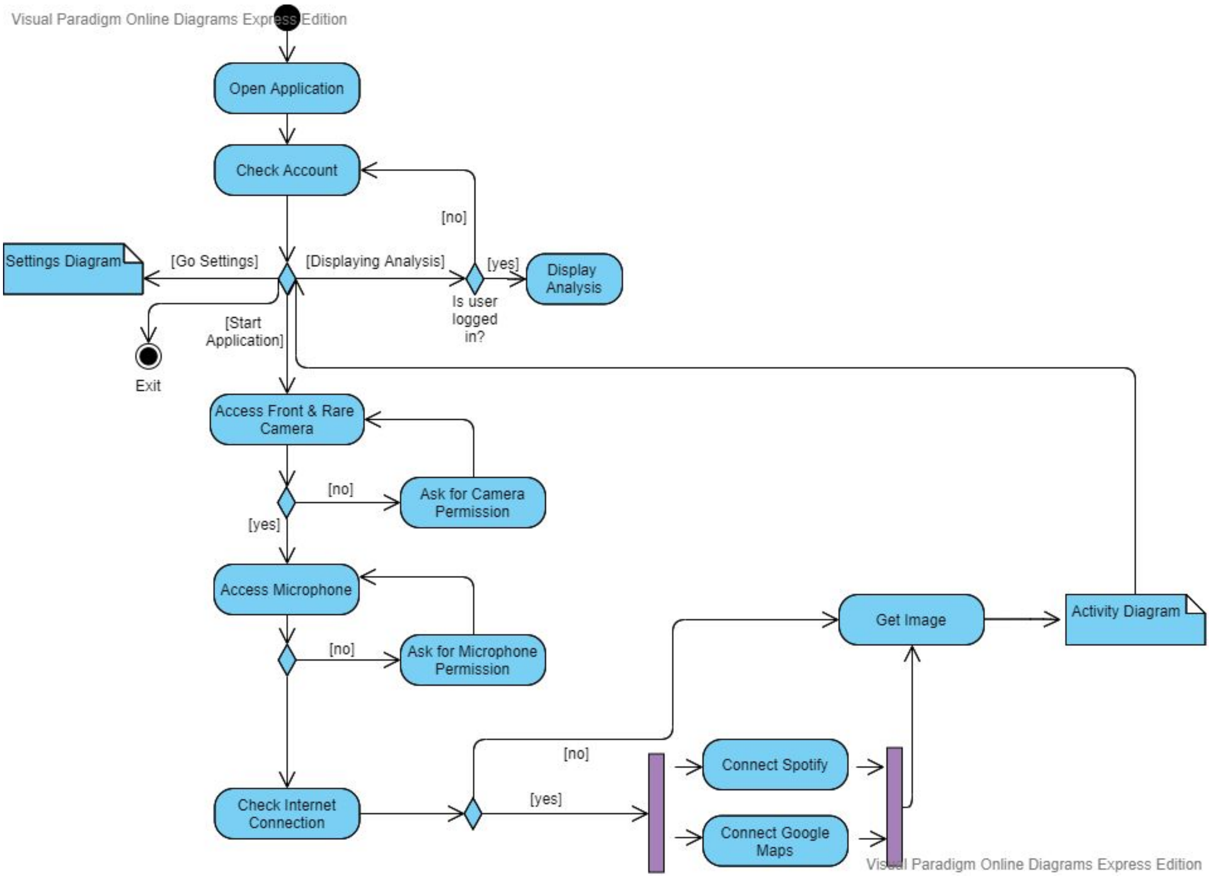


Figure 6: Overall Activity Diagram

3.5.4.2 Feature Based Activity Diagram

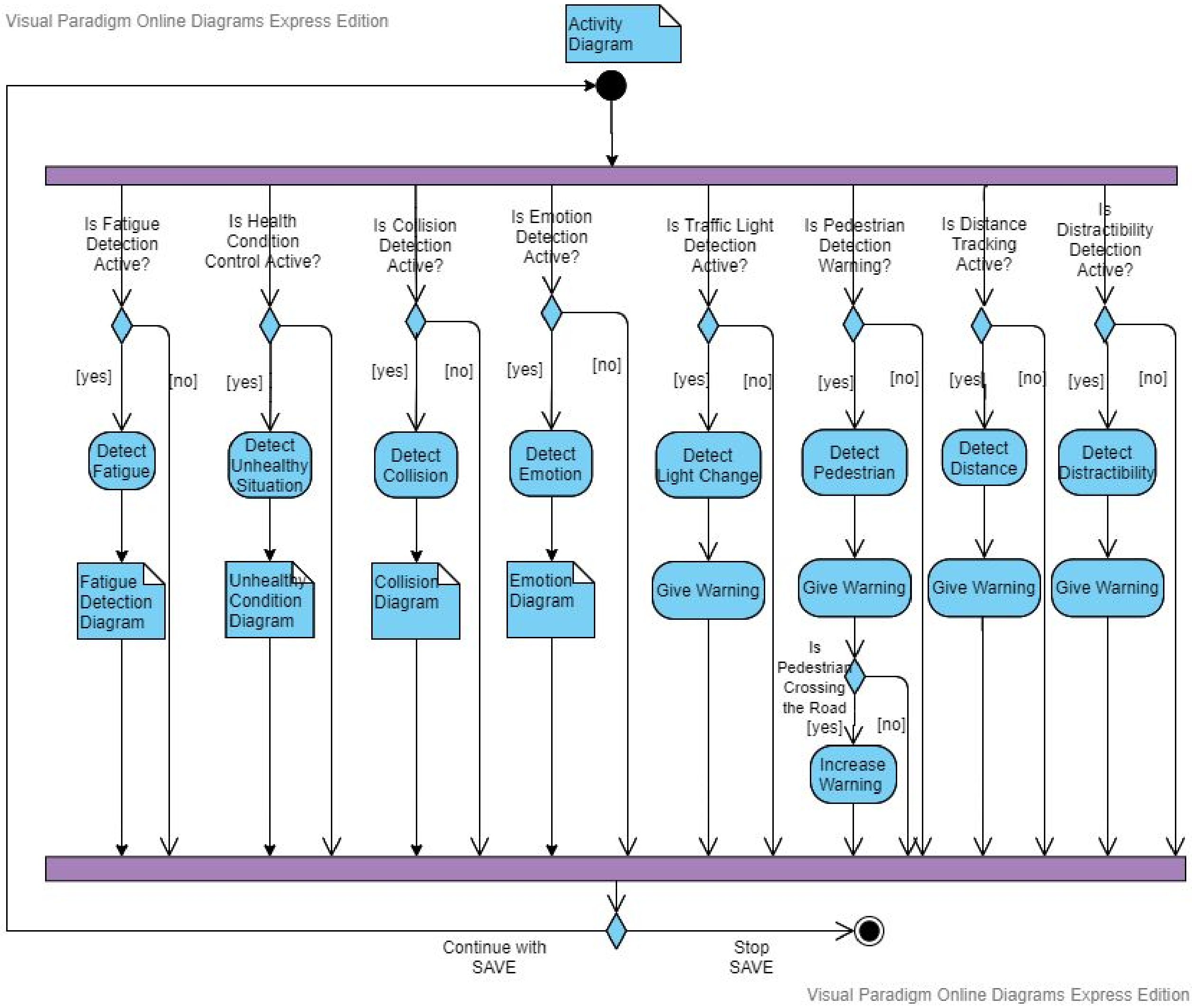


Figure 7 : Feature Based Activity Diagram

3.5.4.3 Fatigue Detection Activity Diagram

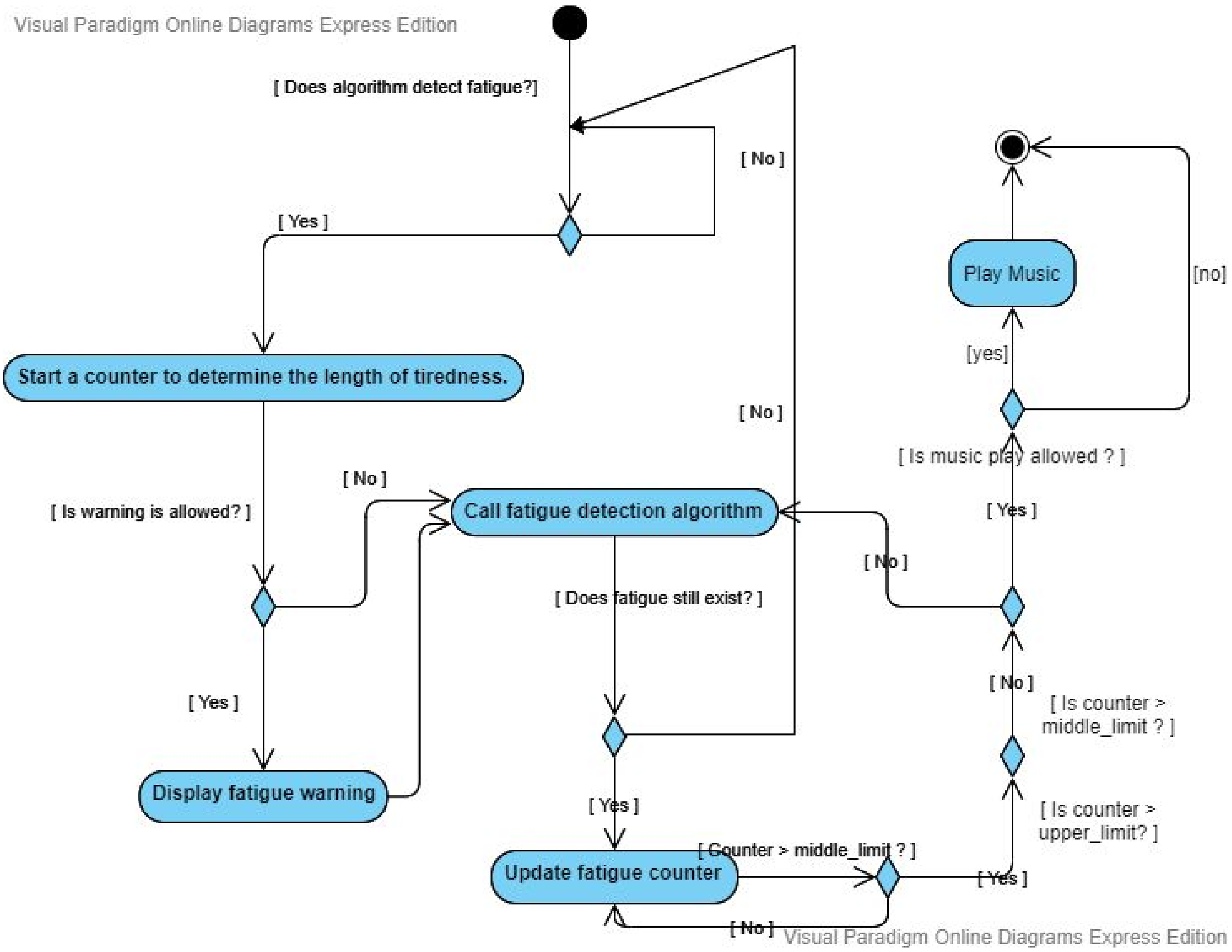


Figure 8: Fatigue Detection Activity Diagram

3.5.4.4 Unhealthy Condition Activity Diagram

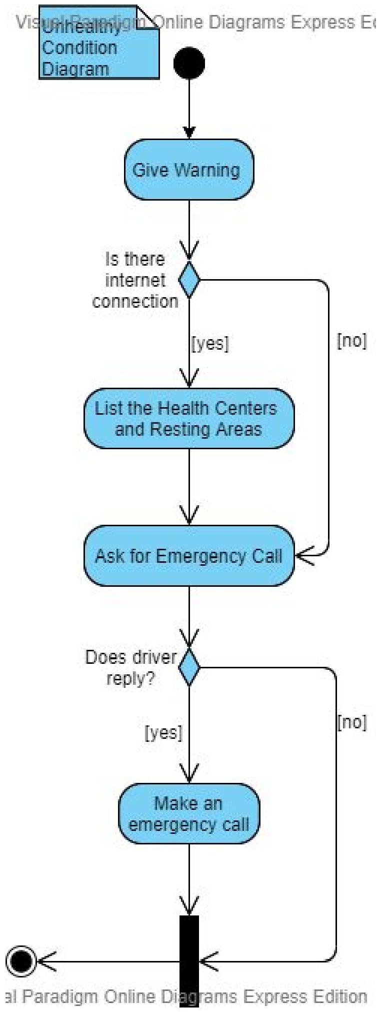


Figure 9: Unhealthy Condition Activity Diagram

3.5.4.5 Collision Activity Diagram

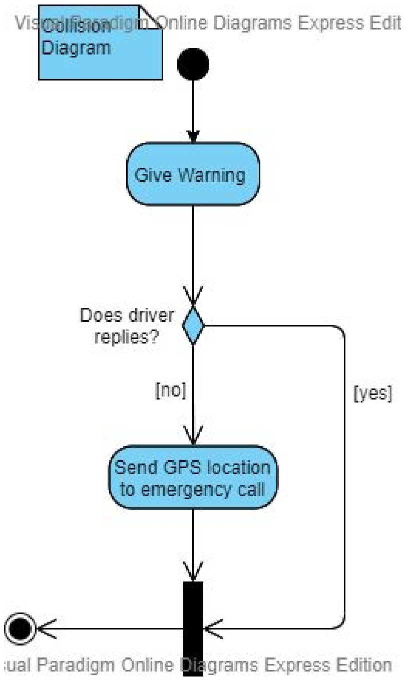


Figure 10: Collision Activity Diagram

3.5.4.6 Emotion Activity Diagram

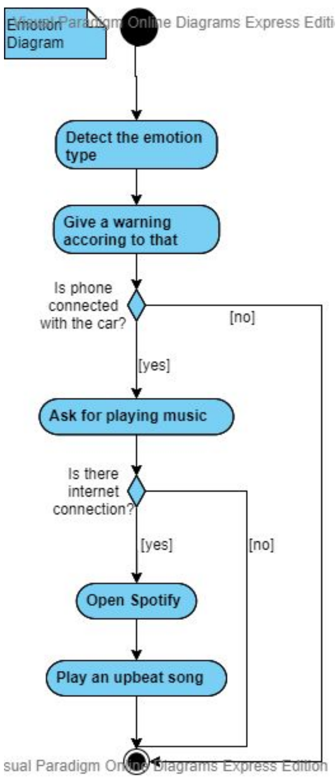


Figure 11: Emotion Activity Diagram

3.5.5.7 Settings Activity Diagram

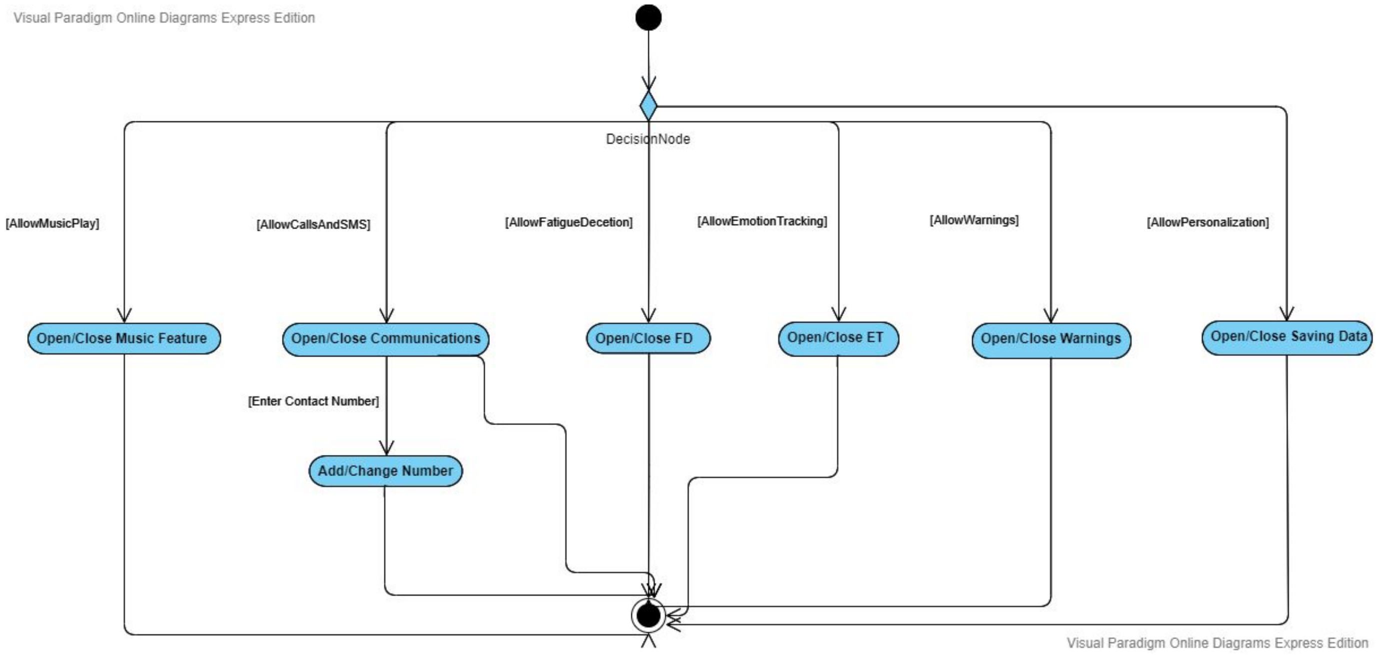


Figure 12: Settings Activity Diagram

3.5.5 User Interface - Navigational Paths and Screen Mock-ups

3.5.5.1 Sign-up

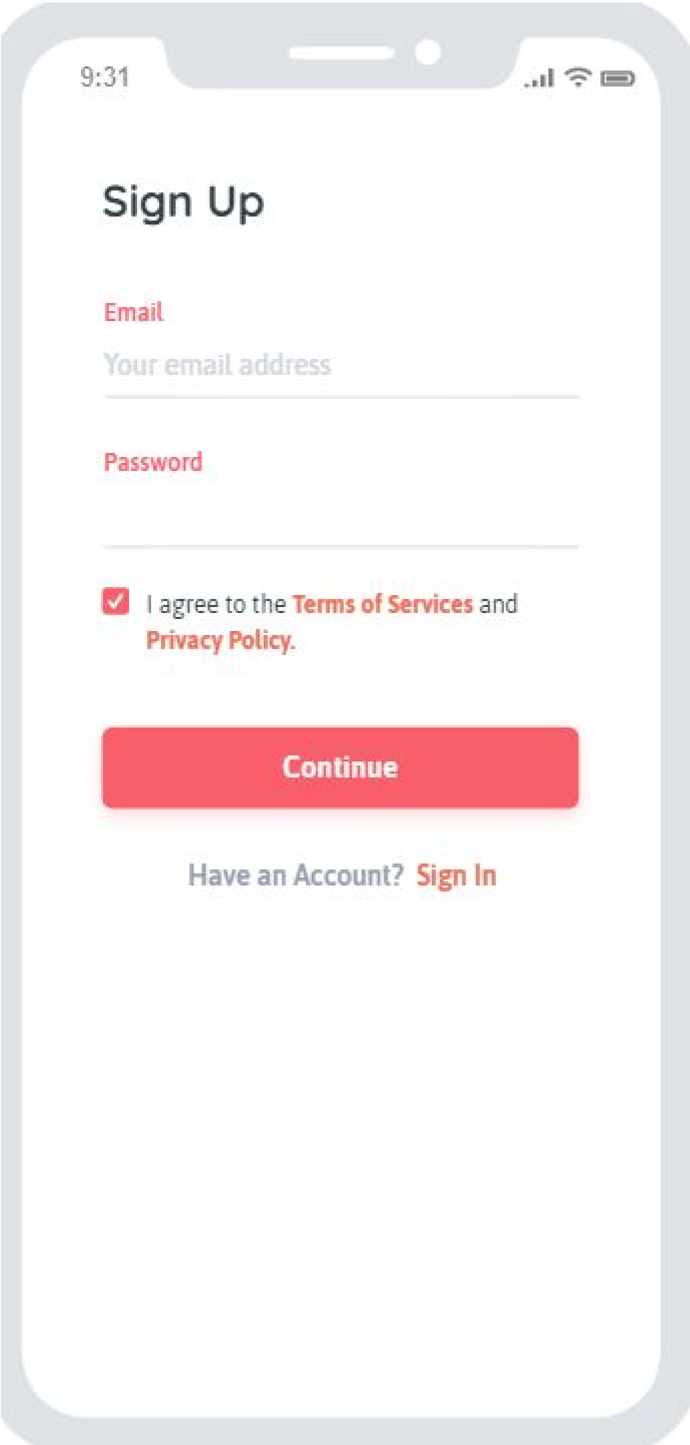


Figure 13: Sign up Screen

3.5.5.2 Sign-in

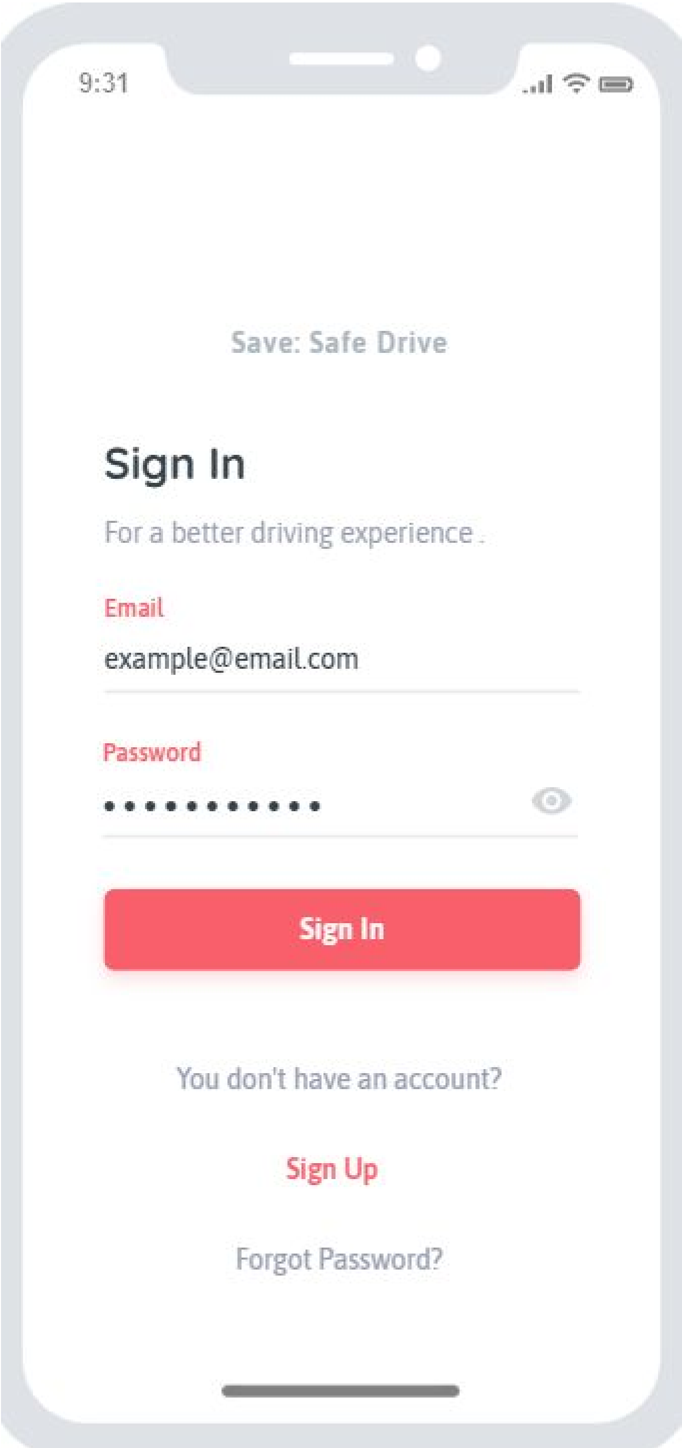


Figure 14: Login Screen

3.5.5.3 Main Screen

Figure 15: Navigation Screen

3.5.5.4 Menu

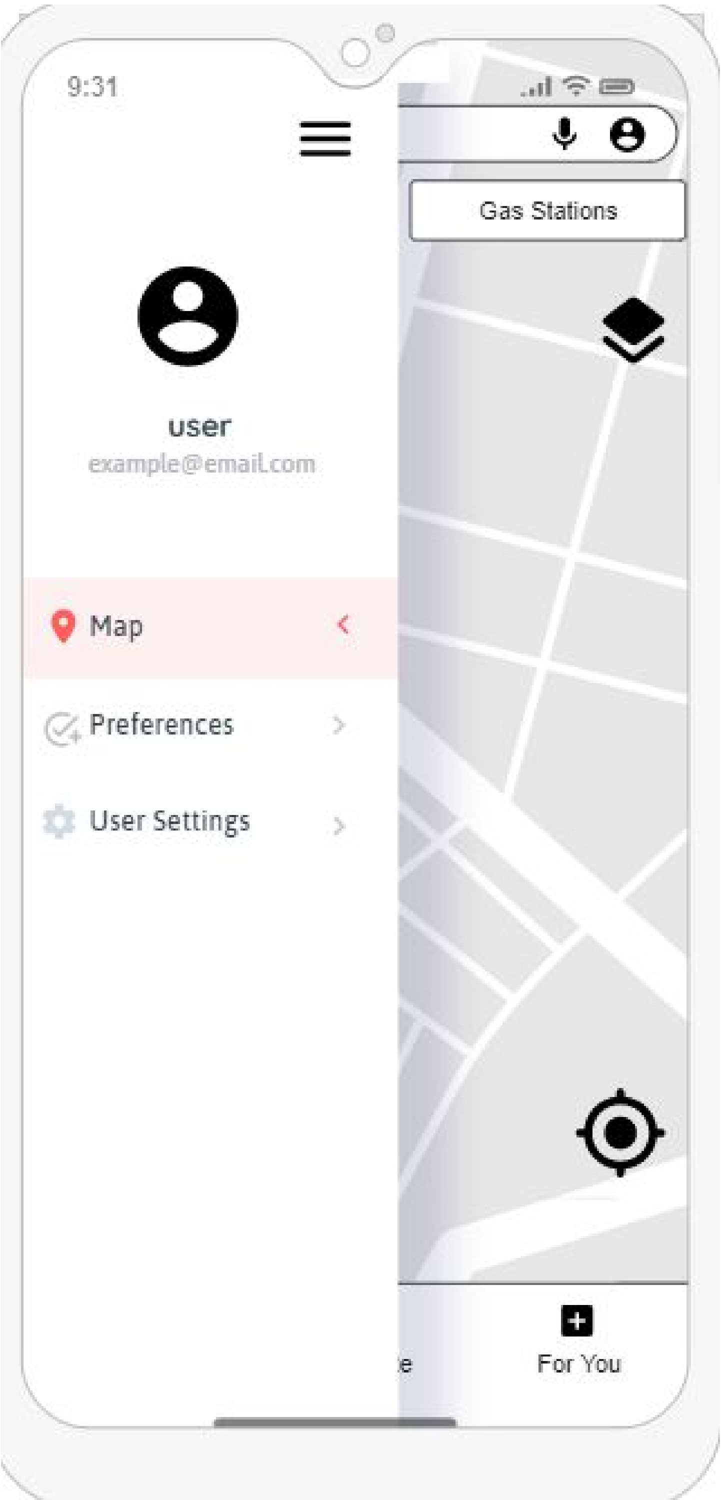


Figure 16: Menu opened screen

3.5.5.5 Fatigue Detection Alert

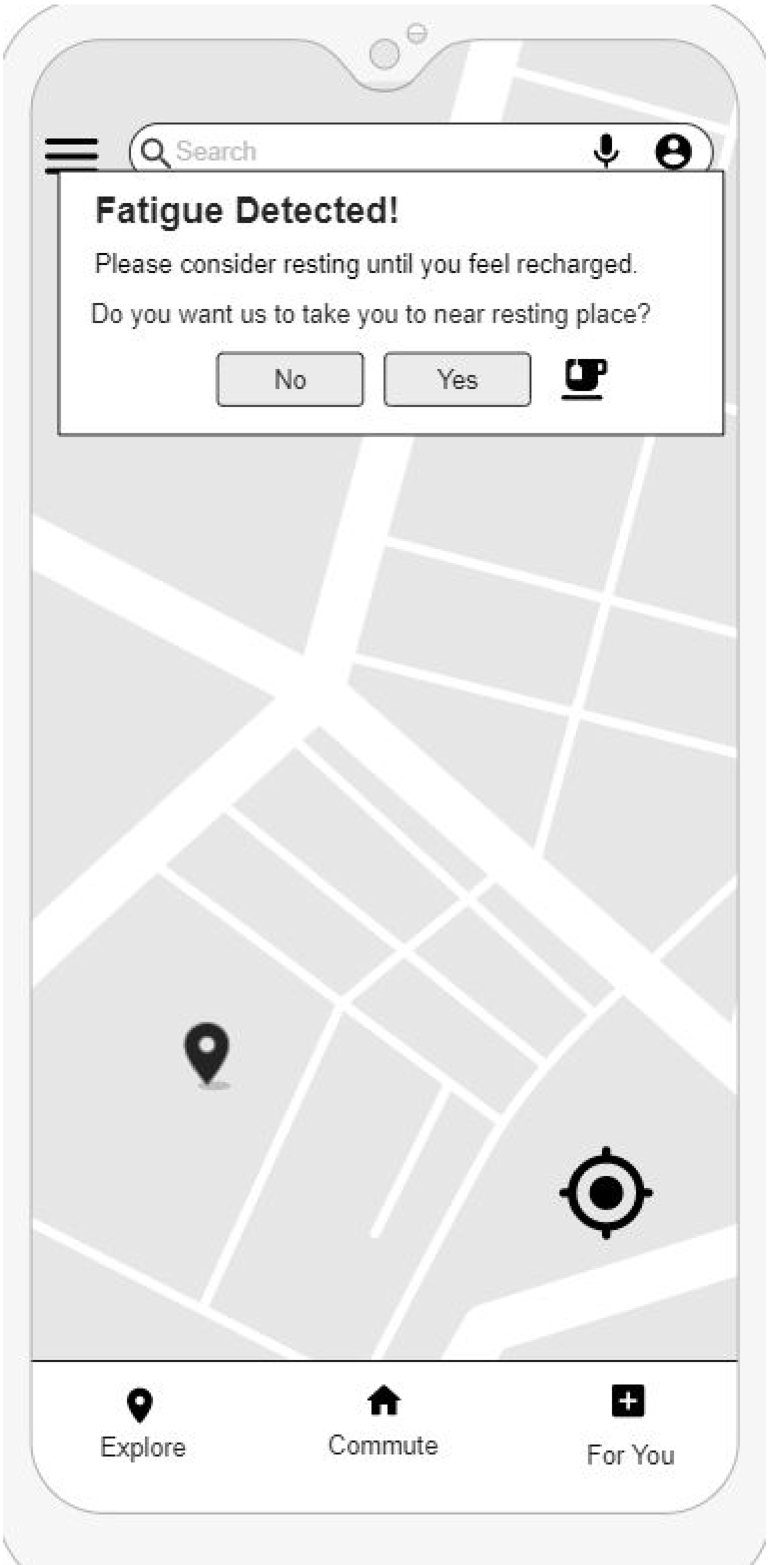


Figure 17: Fatigue detected Screen

3.5.5.6 Health Condition Alert

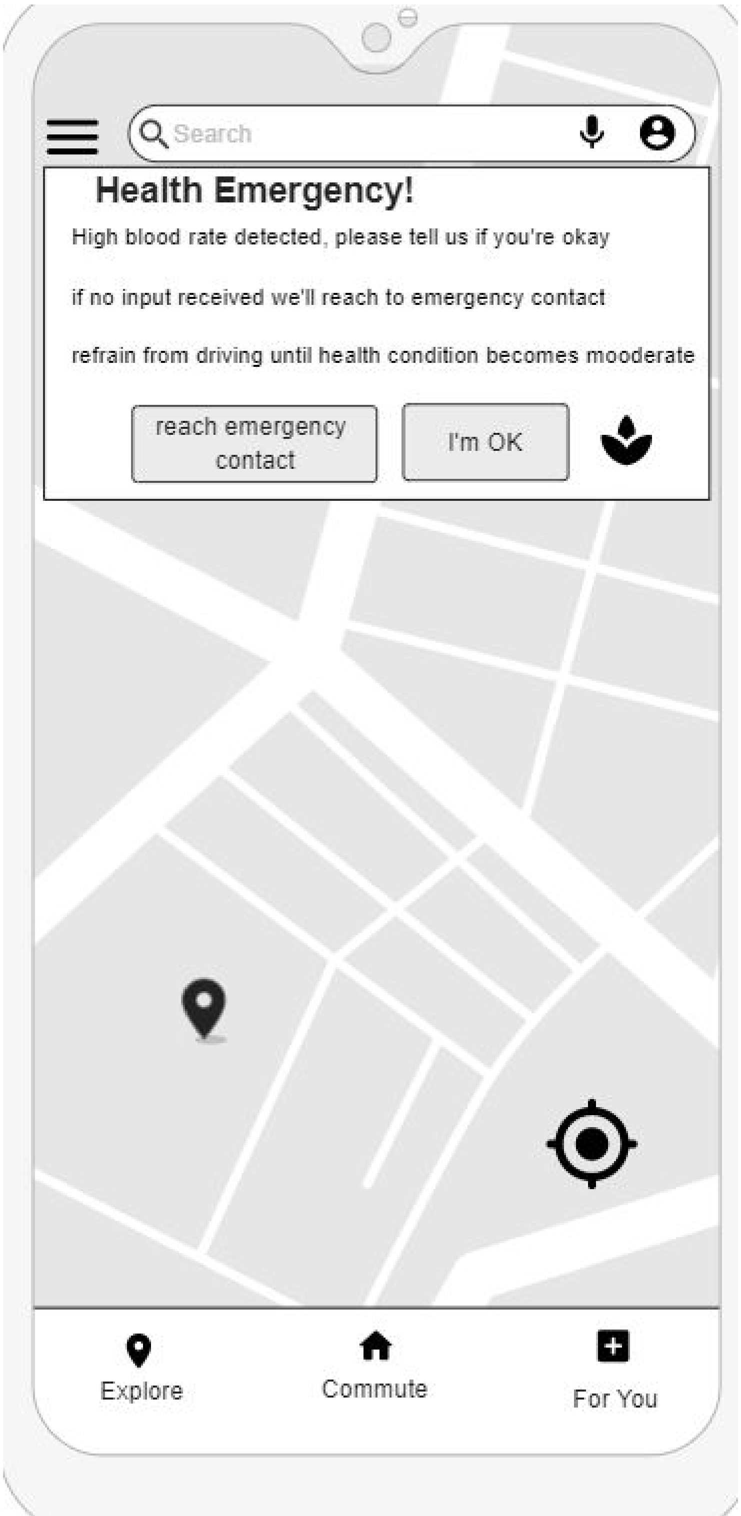


Figure 18: Health emergency screen

3.5.5.7 Search Destination

Figure 19: Search destination from navigation screen

3.5.5.8 User Settings

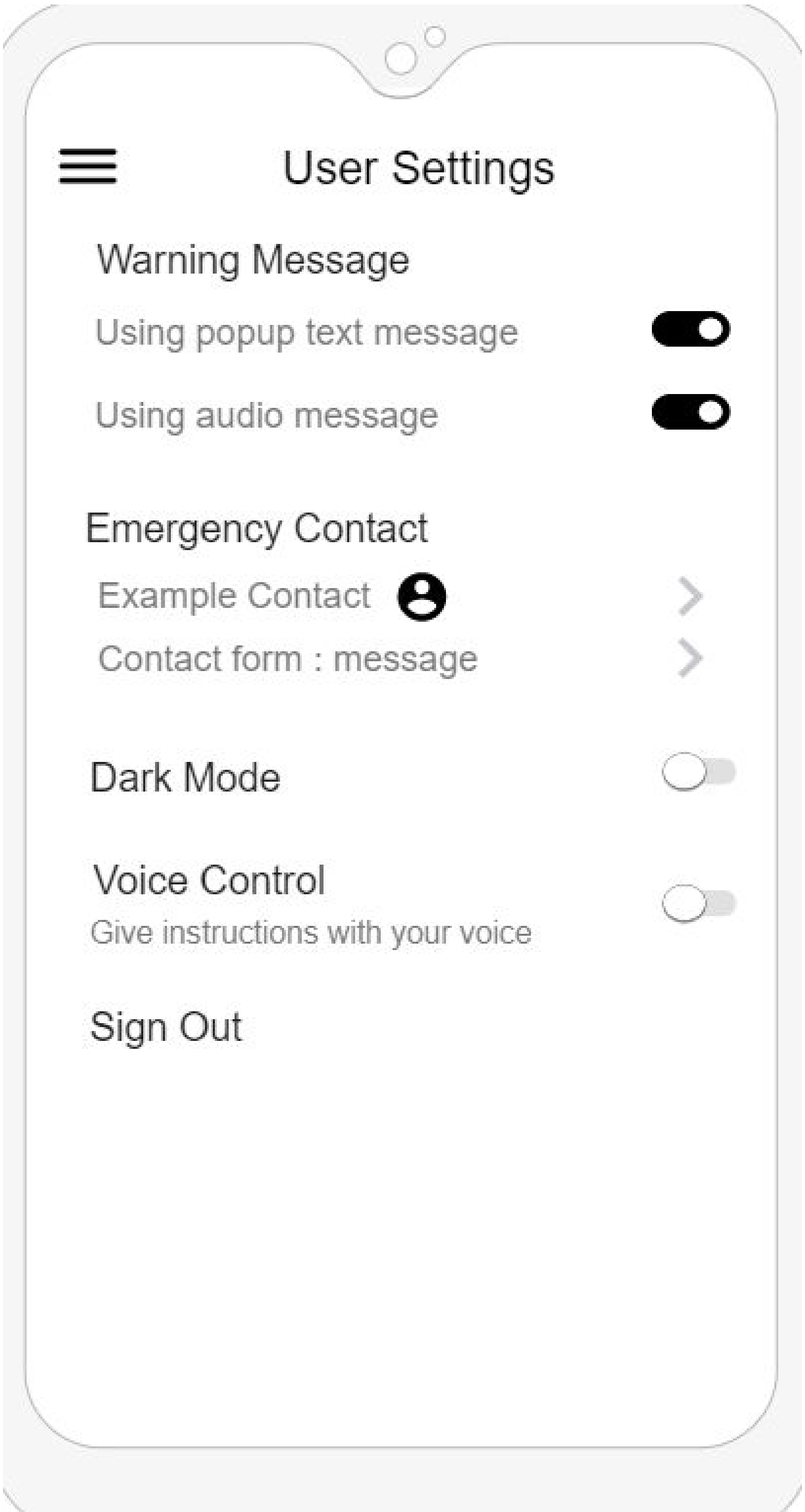


Figure 20: User Settings Screen

3.5.5.9 Preferences

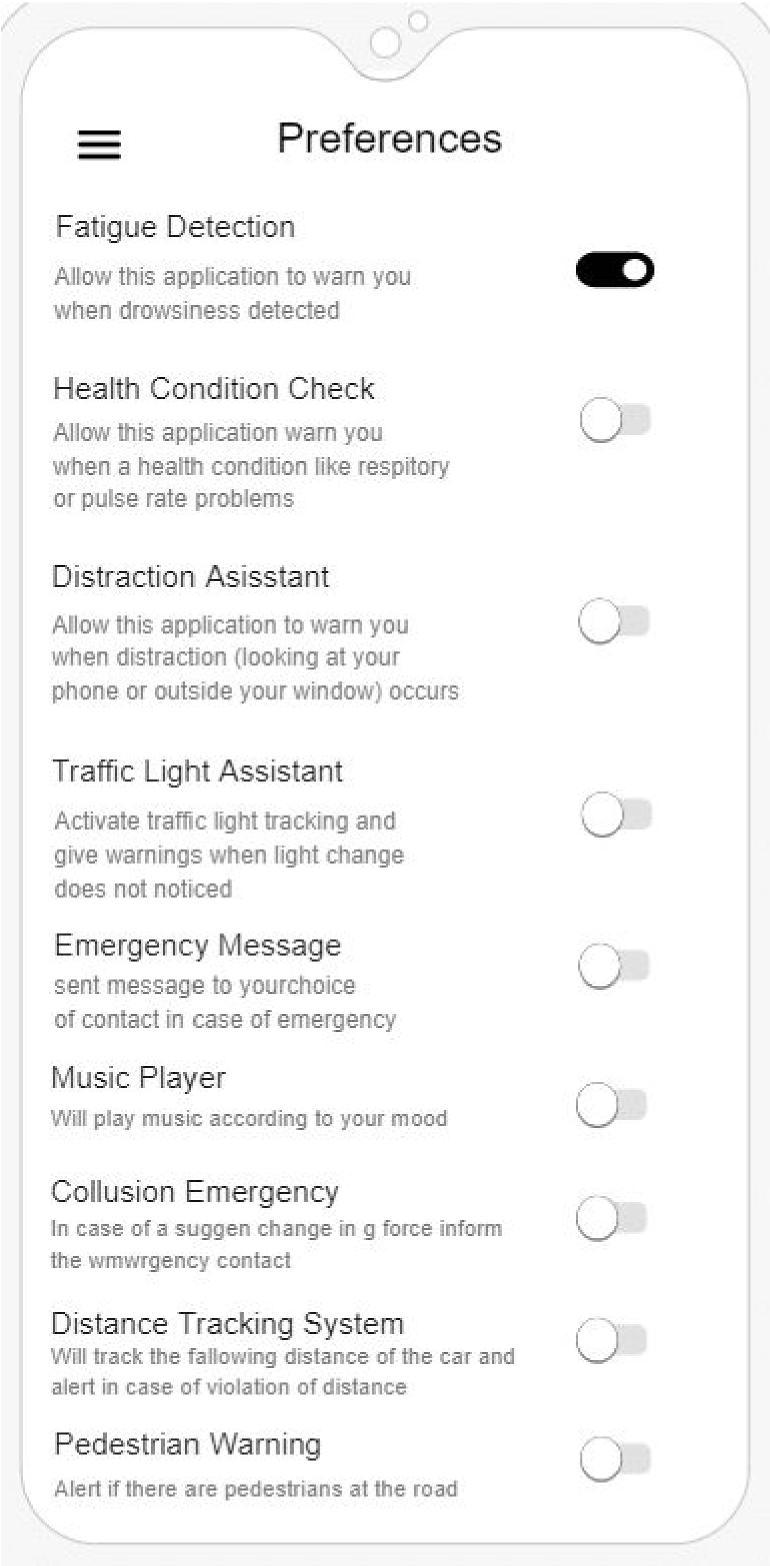


Figure 21: User preferences screen

3.5.5.10 Screen Navigation

The project will use stack navigation properties to keep the pages and return to the previously opened page with the phones return key. When the application opens for the first time, it will open the login page, Figure 13, and ask for login credentials. If the user isn't registered in the system, the application will direct them to the signup page, Figure 14, with the touchable text that says “Signup”. After login is done the application will save the user and won’t ask them the sign in information until the user chooses to sign out.

The user will be directed to the main page of the application, Figure 15, when they open the application if they’re signed in. The main page will have a navigation system that the user can interact with. The navigation will be taken from an existing navigation system application like Google Maps API, it will have features of searching and setting a destination, and will give the optimal route for it (Figure 19).

At the left-top of the navigation page there exists an icon of three lines. This icon opens the side menu of the application (Figure 16), with the first option being the navigation app page called “Map” (Figure 15). Second option is to open the “Preferences” page, Figure 21, which will take the user to a page where they can decide to activate the features the application provides. The third option in the menu, which is called “User Settings”, will take the user to the settings page, Figure 20, where they can decide some of the properties of the application like voice control or emergency contacts.

4.0 Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

At the consideration of various factors, the external factors of the environment that the system works in will be considered. The SAVE application that is being proposed attempts to provide a better driving experience for drivers from different ranges that are not able to afford high levels of technological service.

4.1.1 Data Privacy

There is no use of any personal data against GDPR however, these regulations tend to change with improving data network in global. Because the application’s main focus is the driver, driver related features may violate the data privacy issues in the future in case changes occur in definition of the private data. The features related to data privacy are fatigue detection, emotion detection, health recognition and distraction detection. Any of these features make any video recording to further analysis or sharing.

4.1.2 Distraction Factors

In order for application to function optimally, telephone must be located in a place where both the rear and front camera will see the area of functioning. Since the location of the telephone will be in the sight of the driver, visual and audio notifications from the other application may attract the driver. As a result, the focus of the driver may slide from road to telephone. Additionally, the warnings application give to the driver also have the possibility to distract the driver. To make this distraction minimum sound volume of the warnings are adjustable. Driver can change the sound level to minimize the distraction. Furthermore, to minimize the physical interaction between the driver and the telephone, user interaction related features provided both audio and touch controllable such as activate/deactivate a specific feature.

4.1.3 External Factors

Since the application analyzes the visual input from the driver's face, functioning of the features may be affected by the physical factors. The fatigue detection system program detects fatigue according to eye aspect ratio(EAR), mouth aspect ratio(MAR). EAR value changes person to person. Even though the threshold value of EAR for the fatigue alert defined for an average value, eye aperture of some people may be below the threshold and the system may not function well. Although wearing glasses do not affect the functioning of the application, usage of the sunglasses blocking the view of eyes from the camera and application could not detect the eyes and face properly. Unfortunately, with the Covid-19 pandemic people use face masks for protection. However, similar to wearing sunglasses, using face masks blocking the view of the mouth from the camera and application could not detect the mouth properly. As a result, related features may not work properly.

4.1.4 Upcoming Mobile Device Designs

Constant mobile device development affects the application's working style. Development in the CPU and GPU technologies increases the capability of the processors while decreasing the physical size. Parallel to these developments capabilities of the telephones increases every year. As a result, features work on cloud servers may work on telephone's own hardware. Position and the resolution of the cameras also open the change over the years. Telephone location may be required to calibrate for optimum performance in case of telephone change. Telephones are getting bigger in time. New telephones may not fit to old phone holders. Users may purchase a new phone holder to locate the telephone.

4.1.5 Evaluation of the Constraints

	Effect Weight (out of 10)	Effect to the Application
Data Privacy	5	Does not create a problem with the current GDPR. Possible data privacy definition change may require further action in the implementation.
Distraction Factors	4	Distraction level changes driver to driver. More customizable user interfaces may be developed.
External Factors	9	External factors may heavily affect the functioning of the application.

Upcoming Mobile Device Designs	6	Additional hardware purchase may be required. (Additional cost for user)

4.2 Risks and Alternatives

SAVE application has a significant dependency on computer vision and image processing applications as mentioned above. Due to the fact that these computational abilities require high performance of the device it works on, it has a risk factor that application may not work for all devices at the same level or requiring high energy consumption with short battery life. Position of the cell phone camera is an issue to consider as well as computational performance of the phone.

4.2.1 Computational Performance of the Cell Phone

As the application proposed requires intensive computation of visual models with different tasks occuring at the same time, as well as requiring GPS tracking and voice recognition features, hardware of the user has a risk of being an obstacle against the functionality of the application. SAFE application has various features which can be switched on and off as user desires. Although users can switch desired features off in order to reduce the computational burden on the device or reduce the energy consumption, our aim is to provide users with the most of what their hardware can achieve and the best user experience.

In this regard, our aim is to make most intensive computations that do not require instant responses on the cloud server. Although this will reduce the computation done on device, it will require constant data transaction between servers which means high internet performance.

4.2.2 High Power Consumption

As mentioned in the previous sections the application will use several computational models at the same time. Most of the features we proposed require high CPU & GPU usage. An application that affects battery life of a phone in a negative way is not a result we want to achieve due to the fact that it creates an unfavorable user experience.

According to the issues stated above our alternative plan is expected to solve these issues by reducing the extensive workload on the phone and avoiding any pitfalls that affect customer satisfaction. These solutions will be proposed at the later stages of the report.

4.2.3 Data Privacy Issues

Our application gathers significant amounts of personal information such as facial image, respiratory rate, health condition, stress level, taking footage of the user continuously etc. Due to the fact that related data are considered as private information, our application creates a discourse on privacy. In order to avoid any violation of law regulations we are going to propose our solutions on processing, and protecting sensitive information of the user.

4.2.4 Lightning of The Environment

SAFE mobile application requires image capturing of the user and visual connection between the environment it works in to function properly. One of the obstacles we are expecting to face is the performance of cell phone cameras in low light. Considering the fact that fatigue happens mostly on night drives and it is proposed as a main feature of our application, performance of our application on low light is an important topic to have a discourse in order to provide the best user experience and achieve most of our proposed functionality.

Considering all of these factors, performance of application on low light and low visibility requires an alternative plan to solve proposed issues which we will discuss in the next section.

4.2.5 Proposed B Plan Solutions

Considering the discussions stated above, our plan B is developed considering the elimination of the risks that will reduce user satisfaction such as computational workload, sensitive data issues, low visibility, and reduced battery life.

Proposed B Plan solutions mentioned are the following:

- Features of the application that does not require instant response but long term calculations will work on the cloud server in an optimized way that will reduce the computational workload of the phone, decreasing any latency and high energy consumption.
- We expect our features to work on external light inside the city. Rather than that, urban areas still remain as a problem. An external infrared camera can be used for thermal imaging and night vision. Such a camera can be obtained for reasonable prices. We are not going to require purchasing such tools. Users can have additional infrared support as desired.
- For data privacy, we will not store any private image or sensitive information such as facial images of the user. Only the mandatory information will be stored such as an emergency call list. Computations including sensitive information will be handled internally rather than sending to the cloud server. If any information is required to send, it will be encrypted and will not be shared with any third party source.

Table 1: Summary of the Risk Factors and Related Alternative Plans

Risk Factor	Likelihood	Effect on the project	B Plan Summary
Computational Performance of the Cell Phone	Medium	Reduction of functionality, high latency and unfavorable user experience	Limitation on the functionality of the features of the application. Handling intensive and long term

			calculations on the server
High Power Consumption	Medium	Adversely affected customer experience	Limitation on the functionality of the features of the application.
Data Privacy Issues	Low	Sensitive data obtained from user creating legal enforcement and privacy issues	Application won't store any private information in any part of the system. Private data is going to be encrypted and work on the internal system.
Low Visibility due to Lack of Lightning	Medium	Key functionalities adversely affected. Decrease practicality of the application	Users are supported to purchase any night vision/infrared camera.

4.3 Project Plan

The final goal of our project is to implement a complete Android application that will work as a driver assistant and notify the user about drowsiness and emotions that can affect the driving performance and will provide solutions such as a break location.

With respect to the CS 491/2 structure, our project is divided into two parts. During the first semester, we will implement our core features: fatigue detection, emotion detection, and collision emergency. In the second semester, we will implement our other features: traffic light assistant, pedestrian warning system, distance tracking assistant, distractibility detection.

After clearly indicating the steps of our project and the jobs that need to be done, we divide the project into three parts: structural development, technologies and models, and reports.

Structural development includes the general design of the project, namely, back-end design (Java Spring) , database, cloud services, and front-end design (user interface). Technologies and models include the models that we are going to use such as Remote PPG models, their implementation and integration. Reports WP includes reports that must be written for CS491/2 courses and the design of the website of our project.

WP#	Work Package Title	Leader	Members Involved
WP1	Structural Development	Elif Demir	Elif Demir, Münevver Uslukılıç, Büşra Ünver
WP2	Technologies and Models	Utku Kalkanlı	Utku Kalkanlı, Celal Bayraktar
WP3	Reports	Münevver Uslukılıç	Elif Demir, Utku Kalkanlı, Büşra Ünver, Celal Bayraktar, Münevver Uslukılıç

4.3.1 Work Packages

In this chapter, work packages are explained in more detail.

<table>
<tr><td colspan="4">WP 1: Structural Development</td></tr>
<tr><td colspan="4">Start date: November 22, 2020 End date: April 18, 2021</td></tr>
<tr><td>Leader:</td><td>Elif Demir</td><td>Members involved:</td><td>Münevver Uslukılıç, Büşra Ünver</td></tr>
<tr><td colspan="4">Objectives: This package makes sure that a well working environment will make sure that models can work properly, and the user has an easy to use application. Mainly, it will provide a secure and fast database, a trustable cloud system, easy to use user interface, and a Java Spring implementation that will be the core of others.</td></tr>
<tr><td colspan="4">Tasks:

Task 1.1 Implementation of basic user interface : The non complete user interface implementation that will be required to show our first part features (remote PPG, emotion-fatigue detection, collision emergency) for the first demo.

Task 1.2 Repository-Entity-Model : Implementation of database related classes such as User. Since users will have some personal information that will be stored such as application setting preferences, emergency call number, a well designed</td></tr>
</table>

encrypted database will be used and proper representation of it in the Java Spring back-end will be done in this task.

Task 1.3 Database Connection**:** *Database connection with Java Spring will be implemented. It will be tested via our user interface, or Postman if the user interface is not ready at the moment.*

Task 1.4 Exceptions**:** *It is important to provide custom exceptions in order to protect the system and increase the user experience quality in the system. Therefore, custom exceptions will be implemented.*

Task 1.5 Configuration**:** *If the user has a personal account, they can access it with username and password. Their credentials will be checked from the database, and using Spring Java features the security will be provided.*

Task 1.6 Cloud Implementation**:** *Cloud system will be understood and integrated to our system.*

Task 1.7 Cloud Testing**:** *Cloud system will be tested to understand whether it can work properly or not.*

Task 1.8 Back-end testing I**:** *The back-end implementation for semester I will be controlled with our tests. These tests will be conducted by the members of WP1 since we do not think it is necessary to conduct another team for tests.*

Task 1.9 Complete User Interface**:** *The complete user interface of our application that we will show at the end of the year will be implemented.*

Task 1.10 Back-end testing II**:** *The back-end implementation for semester II will be controlled with our tests. These tests will be conducted by the members of WP1 since we do not think it is necessary to conduct another team for tests.*

Task 1.11 Voice Control Implementation**:** *Our system provides voice control feature to provide non distracting communication between driver and the application, therefore; this system will be implemented within backend.*

Deliverables

D1.1: *Java Spring back-end system*

D1.2: *User interface*

D1.3: *Cloud system*

D1.4: *Database*

WP 2: *Technologies and Models*

<table>
<tr><td colspan="4">Start date: November 22, 2020 End date: April 18, 2021</td></tr>
<tr><td>Leader:</td><td>Utku Kalkanlı</td><td>Members involved:</td><td>Celal Bayraktar
Büşra Ünver
Elif Demir
Münevver Uslukılıç</td></tr>
<tr><td colspan="4">Objectives: This work package includes the technologies and model libraries that are going to be implemented and integrated to the application to do the process of the features and the testing whether if they all work properly.</td></tr>
<tr><td colspan="4">Tasks:

Task 2.1 Remote PPG Implementation and Integration: Implementing the health condition tracker libraries to the application in order to do the health analysis.

Task 2.2 Fatigue Detection Implementation and Integration: Implementing the fatigue detector libraries to the application in order to track the fatigueness level of the driver, so that application can give a warning when there is an ongoing safeless driving for the driver.

Task 2.3 Emotion Detection Implementation: Implementation of the emotion detecting system to track if the user is in an uncomfortable emotion situation.

Task 2.4 Collision Emergency System Implementation: Implementation of the collision emergency message system to detect the dramatical change in the G force and send the GPS location to the emergency call number.

Task 2.5 Traffic Light Assistant Implementation: Implementation of the traffic light color detecting system in order to give warning to the user when the light has changed.
Task 2.6 Pedestrian Warning System Implementation: Implementation of the pedestrian warning system in order to give warning to the user when there is a pedestrian standing on the sidewalk.

Task 2.7 Distance Tracking System Implementation: Implementation of the distance tracking assistant to track the distance between the car that is in front of the user's car.

Task 2.8 Distractibility Assistant Implementation: Implementation of the distractibility that tracks the user if any distractibility situation is detected.

Task 2.9 Remote PPG and Emotion Detection Integration with Cloud System: Integrating these two features to the cloud to make application analyse the unhealthy situation and emotion detection in the cloud server.</td></tr>
</table>

Task 2.10 Testing the prior features: Testing remote PPG, fatigue and emotion detection and collision emergency message system to check whether the analysis and the results of them are correct. ***Task 2.11 Testing the other features:*** Testing traffic light assistant, pedestrian warning system and distractibility assistant features to check whether the analysis and the results of them are correct.
Deliverables ***D2.1:*** *Remote PPG & Fatigue Detection* ***D2.2:*** *Emotion Detection* ***D2.3:*** *Collision emergency message system* ***D2.4:*** *Traffic light assistant* ***D2.5:*** *Pedestrian warning system* ***D2.6:*** *Distance tracking assistant* ***D2.7:*** *Distractibility assistant*

WP 3: *Reports*			
Start date: *October 1, 2020* **End date:** *April 30, 2021*			
Leader:	*Münevver Uslukılıç*	**Members involved:**	*Elif Demir,* *Utku Kalkanlı,* *Büşra Ünver,* *Celal Bayraktar*
Objectives: *To make sure that all the requirements excluding the implementation of the project such as reports, and webpage will be prepared before their deadlines.*			
Tasks: ***Task 3.1 Website design of the project*** **:** The project website is going to be designed. It is going to include the reports of the project. ***Task 3.2 Writing Project Specification Report*** **:** Determining the specification and requirements of the project.			

Task 3.3 Writing Analysis Report: Determining the analysis of the project. It consists of system models, mockups, project and team working plans. ***Task 3.4 Writing High-Level Design Report:*** Determining the design goals of the project and having the plan for decomposition so that we can divide the project systems into subsystems and work more efficiently. It also includes software / hardware platforms and other technologies, system requirements, boundaries and other conditions. ***Task 3.5 Writing Low-Level Design Report:*** Stating the structure of the implementation as packages and classes. ***Task 3.6 Writing Final Report:*** *The report of the complete system includes revised requirements, design details, development and implementation details and process, project and team work features, maintenance plan and test details.* ***Task 3.7 Presenting Demo I:*** The demonstration that is going to be completed on the last day of the first semester in order to present the major points of the project. ***Task 3.8 Presenting Demo II:*** The demonstration that is going to be completed on the last day of the second semester in order to present the completed work of the project.
Deliverables ***D3.1:*** *Website* ***D3.2:*** *Project Specification Report* ***D3.3:*** *Analysis Report* ***D3.4:*** *High-Level Design Report* ***D3.5:*** *Low-Level Design Report* ***D3.6:*** *Final Report* ***D3.7:*** *Demo I* ***D3.8:*** *Demo II*

4.3.2 Software Management Tools

In order to organize the task and work, we started to use Trello at the beginning of the semester. However, since our works get complicated over time, Trello becomes inadequate to manage our work. Therefore, we decided to switch to another project management tool. Since we are a small team, we do not want to use a complicated management tool. Therefore, we started to investigate easy to use and free tools that can also provide the required features to us. We first try nTask Manager. It is easy to use; however; it does not provide a Gantt chart and 6 team members. Since we will also add our supervisor, we need access for 6 people. Therefore, we continue our search. We found another project management tool that is called Freedcamp. It is more complicated than the nTask Manager; however, we get used to it over time and we decided to use it since it provides us all the features except Gantt chart. For the gantt chart we use TeamGantt.

We created our webpage via GitHub and we will use it in our implementation as our main shared implementation tool.

4.3.3 Gantt Chart

Mentioned work packages and tasks with their estimated start and end dates are shown in a Gantt chart in Figure 22. Numbers on the grey part indicates MM/YY.

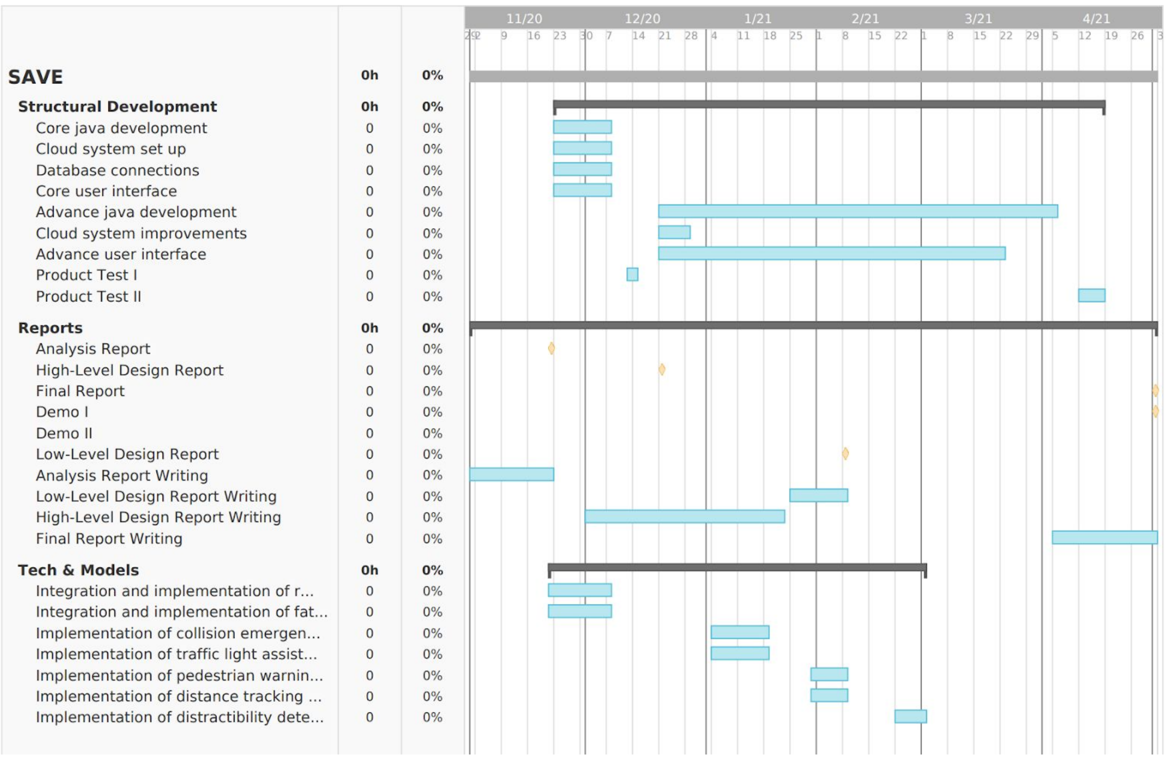


Figure 22: The Gantt chart for SAVE

4.4 Ensuring Proper Teamwork

During the progress of our project our team priorities communication between the group members since we know from older experiences with several projects during our university years that the lack in communication could cause the most severe damage to the project. To ensure the communication effectiveness and to be able to manage our time properly, we decided to form a regular schedule dedicated to the project where everybody could focus on the project. Also to track our tasks and our progress records, we decided to use a tool specifically for project management which is described in detail in the section 4.3.2 Software Management Tools. Sharing our individual work within the project members helps us to review the work and get feedback accordingly.

4.5 Ethics and Professional Responsibilities

Our application observes the user, takes their personal information like image or voice as an input to process. Since we collect our user's confidential information it is our responsibility to protect them. Our project will not collect any user information that our features don't need to function with and the information collected will not be shared with any 3rd party organizations or individuals. Nobody in the project team will have access to the information collected from the user and our system will preserve those information in encrypted manner, and take the proper measurements. Encryption will be considered in early stages of the implementation. We will provide a privacy policy that is considered data protection laws of the Department of Health & Human Services (HHS) [4], easy to follow and informs the user how our team ensures the data preservation following the regulations in a transparent manner.

4.6 Planning for New Knowledge and Learning Strategies

In the span of our project duration, the team members have to learn new skills and should be able to apply them in harmony with other features. After studying the technologies and getting familiar with it our teams want to move on to more active learning strategies that require retrieval of the studied information. First method to apply it should be self practice and then our team members should share what they have learned to other members and elaborate on the topics which also ensures the knowledge we obtained is truly learned and other members could understand the topic.

5.0 Glossary

Remote Photoplethysmography	Remote Photoplethysmography (PPG) is a contactless way to measure human cardiovascular activity by measuring the reflection variations of the skin registered by a video camera [5]. It only requires video recording with a high-resolution camera[6].
Eye aspect ratio	Eye aspect ratio process is the combination of eye localization, analyzing the whites of eyes and determining the period of time that white region of the eye disappears to indicate the blink rate of the human by using facial landmarks[7].
Mouth aspect ratio	Mouth aspect ratio The horizontal and vertical distance of the mouth by using 2D facial landmark locations [8].

Both EAR and MAR are calculated by computing the Euclidean distance between the landmarks using SciPy library [8].

6.0 References

[1] *Fatigue statistics*, Transport Accident Commision. Accessed on: Nov. 21 2020. [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*. Accessed on: Nov. 21 2020. [Online]. Available: https://drivision.wordpress.com/

[4] *Records, Computers and the Rights of Citizens (HHS)*, 14 June 2016. Accessed on: Nov. 21 2020. [Online]. Available: https://aspe.hhs.gov/report/records-computers-and-rights-citizens

[5] Verkruysse W., Svaasand L. O., Nelson J. S., "Remote plethysmographic imaging using ambient light," Opt. Express 16(26), 21434–21445 (2008).10.1364/OE.16.021434

[6] *What is RPPG?*, Noldus. 2020. Accessed on: Nov. 21 2020. [Online]. Available: https://www.noldus.com/blog/what-is-rppg

[7] Rosebrock, A., 2020. *Eye Blink Detection With Opencv, Python, And Dlib - Pyimagesearch*. PyImageSearch. Accessed on: Nov. 21 2020. [Online]. Available: https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/

[8] Kir Savaş, Burcu & Becerkli, Yaşar. (2018). Real Time Driver Fatigue Detection Based on SVM Algorithm. 1-4. 10.1109/CEIT.2018.8751886.