Bilkent University
Department of Computer Engineering

Senior Design Project

Group: T2317 - RoadVisor

Analysis and Requirement Report

Group Members:
Ahmet Faruk Ulutaş - 21803717 - faruk.ulutas@ug.bilkent.edu.tr
Ammaar Iftikhar - 21901257 - ammaar.iftikhar@ug.bilkent.edu.tr
Arda İçöz - 21901443 - arda.icoz@ug.bilkent.edu.tr
Emin Berke Ay - 21901780 - berke.ay@ug.bilkent.edu.tr
Nurettin Onur Vural - 21902330 - onur.vural@ug.bilkent.edu.tr

Supervisor:
Asst. Prof. Dr. Hamdi Dibeklioğlu

Innovation Expert:
Cem Çimenbiçer

Course Instructors:
Erhan Dolak
Tağmaç Topal

13.11.2022
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.

Contents

Analysis and Requirement Report

RoadVisor

1 Introduction

Navigation in unknown areas is a difficulty faced by automobilists. The automobilist may be an experienced taxi driver or a novice. Navigators don't only face the hurdle of driving through unknown streets, but also face the task of keeping track of road information like traffic lights and street signs simultaneously. The presence of so much information for the driver to process might lead to increased mistakes or accidents [1]. There are other issues that pertain to the road, like boundaries and crowd density, that play a crucial role to inform the driver. If a greater amount of information is presented to the driver with more ease, the driver will be able to perform better.

While there are many applications that attempt to present the driver with directions to the destination, there aren't many that help simultaneously inform the driver of other necessary information simultaneously. These applications mostly focus on delivering the queried information to the driver like the location of certain places and the distance to the next turn. Information about processing real-time information for the navigator on the road is not prioritized by advanced applications like Google Maps. Our application attempts to fill the disparity between the user view and the application. RoadVisor assists the drivers in real-time using Machine Learning and Augmented Reality.

The main motivation behind RoadVisor is to assist the driver in processing the real-time events on the street fetched using the mobile phone camera while also providing the user with an improved street view and directions on their mobile phones. The ability to process information and the information fetched by the application is constrained by the processing capabilities of the devices. The application will use different Machine Learning methodologies, like Deep Neural Networks, and APIs, like Google Maps Platform API, to continuously inform the driver about road information and directions. Augmented Reality will try to ensure that the driver does not miss the road information while consulting the mobile for directions or other information.

The application will mitigate traffic violations, like violations of street signs and traffic lights, by drivers. The application will try to fill the void in the navigation application industry by presenting a viable and better alternative to the ones available in the market. Our application will be an attempt to assist the driver without distracting or bugging to decrease the fatalities caused by road accidents or car crashes [1]. The application will be built incrementally and optimized for usage on mobile devices.

RoadVisor aims to have a place in the Offering section and be a Product Performance focused application according to the 10 Types of Innovation Wheel by Doblin [2]. We believe that the AR-supported navigation feature distinguishes RoadVisor from other applications and provides better and more comfortable functionality. With these distinguishable features, RoadVisor will be an incremental innovation project and optimization will be the key element as the goal is to provide a better user experience.

2 Current System

There are existing systems that are able to provide AR navigation systems for driving assistance. But, as it is this feature is more exclusively used in very high-end car models and comes within their system. As an external application within Android that aims to serve a much broader range of audience, RoadVisor will come out as a strong candidate. There are obviously existing applications that promise to help the driver but this mostly branches into two categories, namely, ones that try to give a more defined picture with combined driving features with most of them lacking AR navigation while ones that provide AR navigation give the impression of being more experimental and more or less like functionality rather than being a true mobile application. In any case, current systems can be exemplified as:

Applications

- Google Maps AR Street View [3],
- Yandex AR [4],
- Mercedes AR Navigation [10]

Datasets

- Lane Detection [5],
- Road Boundary Detection [6],
- Traffic Sign Detection [7],
- Pedestrian Detection [8],
- Traffic Light Detection [9].

3 Proposed System

This is where you provide the details of the results of your analysis work.

3.1 Overview

RoadVisor will use Augmented Reality to assist the driver to navigate. The view provided to the user will include the road view and additional information that improves the information output to the driver. The features we intend to implement are:

1. Augmented Reality Supported Navigation: The feature will provide the user the directions to navigate by showing arrows in the augmented reality view. It will also display road boundaries and turns. The user will be able to view the road as well as the directions. The feature will be implemented using image processing and computer vision techniques. Deep Neural Network methodologies like, Road and road Boundary

detection Network (RBNet), are available for road boundary and region detection [2]. We will additionally use Google Maps API to fetch directions.

2. Traffic Light Detection: The feature will inform the driver about traffic light signals. The feature will help improve driver's awareness of traffic lights. The information utilized by the feature will be fetched using the back camera of the device. The feature is in line with our motive to help drivers process road information with ease. The feature will be implemented using Computer Vision, particularly Convolutional Neural Networks (CNNs) [3]. Such mechanisms are used in ADAS systems. The DriveU dataset is a candidate for use during the training and testing of the application [4]. The DriveU dataset is based on European traffic lights. There are some other candidates that can be used, but the problem with most of the options is their American nature, i.e, the American and European traffic light signals differ. Background noise like the presence of multiple traffic lights might, however, decrease the accuracy.

3. Sign Detection: The application will help users be aware of the traffic signs that might be on the road like 'school ahead" or "danger ahead." The signs the feature will recognize are expected to follow Vienna Convention of Road Signs. The constraint mentioned exists due to the dataset we will use. It is a dataset consisting of 80000 samples from 150+ classes [5]. The implementation of this feature will also utilize Convolutional Neural Networks. The feature assists the user to prevent violation of signs that might lead to other consequences like fines.

4. Crowd Density Analysis: The application will provide information to the user about the pedestrians and crowd in a particular location. The application will use crowd-counting techniques to implement this feature [6]. We can implement this feature using CNNs as well. The Shanghai Tech dataset and the UCF_CC_50 dataset are 2 possible datasets candidates that we might use for the training of the Neural Networks.

Our first goal is to finish implementing feature number 1 (see above). The features here are numbered based on the priority we have assigned to them. Therefore, the flow of implementation will also follow that. Our end goal is to finish the implementation of all the features we have listed above. However, this might change due to the long-developing process of the AR navigation system. The below-mentioned feature will only be implemented if we can successfully implement the above-mentioned features and the devices used to test the application can work effectively in the presence of other features. This latter constraint exists due to the presence of limited computational power of mobile phones and the fact that the entire application is based on Augmented Reality, which is heavier than other simple applications.

5. Crash Detection: Assisting our users during times of danger like car crashes is a feature that we want to include. The feature detects a car crash by using Computer vision. Then, it notifies the emergency service to respond to the car crash. The feature might be able to save the lives of people by immediately contacting the concerned

authorities. However, we need to consider the hardware limitations of mobile devices when implementing this feature. We are working on a possible feasible solution for this feature.

3.2 Functional Requirements

3.2.1. Road Navigation with Augmented Reality

- One of the key intentions behind RoadVisor is to provide the user with complete guidance during the entire road trip. Accordingly, the application not only aims to take the user to a selected location in an optimal way but also gets directly involved with showing the necessary movements of the vehicle.

- RoadVisor aims to achieve this by showing the direction that the vehicle has to follow with the help of Augmented Reality. In other words, the application uses the camera of the mobile phone to track the road while inserting arrows in accordance with the path that the driver has to follow to reach the desired location.

- To start this navigation service, the user needs to enter the destination location and afterwards, RoadVisor will get the footage of the road from the camera as an input to match it frame by frame with the route of the desired location to place the arrows onto the road.

- Arrows need to be constantly changing with respect to the road and furthermore when more drastic movements are required such as turning left or right, those will be emphasized on the screen with larger arrows popping up.

- The screen also has to show information about the current location and the remaining distance to the location.

3.2.2. Traffic Light and Sign Detection

- RoadVisor needs to use the live camera footage of the user to analyze it frame by frame.

- Using this camera footage, the application has to constantly make image analysis to be able to detect the presence of traffic lights and traffic signs visible.

- For traffic lights, RoadVisor needs to inform the user by showing the color of the light on the phone screen and giving a pop-up sign when it spots a red light.

- For traffic signs, RoadVisor needs to inform the user about any sign spotted and what the sign is about. Signs will be classified and shown on the phone screen where signs that require further attention such as stop signs can also have additional pop-up messages for the purpose of alerting.

3.2.3. Pedestrian Detection

- RoadVisor needs to use the phone camera of the user to analyze live camera footage frame by frame.

- By getting the camera footage as input, the application has to constantly make image analysis to be able to detect the presence of any pedestrian that is spotted on the road.

- When RoadVisor spots a pedestrian, it needs to mark it on the screen and furthermore show an accompanying warning message for alerting purposes.

3.2.4. Requesting Help and Crash Detection

- The user will be able to enter phone numbers as “urgent contact” information.

- When the user is in need, he/she will be able to press the SOS button of RoadVisor so that the application sends an automatic message to the urgent contact numbers stating that the user may be in a problematic situation together with the location of the user.

- The SOS feature of RoadVisor can be also activated automatically if the application detects a sound that may resemble a car crash sound. In that case, the application asks the user if there is a problematic situation. If the user fails to respond in a given duration, RoadVisor will accept this as an urgent issue and automatically alert the urgent contact numbers.

3.3 Non-functional Requirements

3.3.1. Reliability

- It’s crucial for RoadVisor to give navigation information correctly. This means that the application has to be reliable in terms of fulfilling the expected necessities such as preparing the route accurately, placing the direction arrows correctly, showing true traffic light and sign information messages, and successfully sending the SOS messages with the location of the user.

- RoadVisor has to operate without significant interruptions and therefore provides navigation service and other promised services during the entire car trip duration starting from the moment that the destination is selected until the user arrives at it given that the necessary conditions are satisfied such as battery and connection.

3.3.2. Performance

- RoadVisor has to have a high standard in terms of performing well in frequent conditions and in a short duration. Since the application needs to constantly extract input from live camera feeds and analyze them frame by frame, the frame rate has to be above a tolerable level so that the application works without significant delay.

- The detection models used within RoadVisor have to perform with high accuracy rates for being considered reliable.

- The application needs to work smoothly with the touch screen which requires RoadVisor to have a fast response time.

- RoadVisor is designed mainly to work in a mobile environment. In this respect, the application has to perform well to satisfy its features without being overly affected by hardware limitations of mobile phones.

3.3.3. Usability

- RoadVisor is an application that aims to reach a very wide range of audience. Essentially, all drivers (especially ones who do not possess the latest technological tools in their cars) are targeted as potential users of our application. In this respect, the application needs to be understandable and easy to use by the target audience.

- RoadVisor needs to be compatible with a wide range of Android phones given that they are above a determined version.

- RoadVisor needs to have a highly responsive, simple-looking user interface. Given that the user will be busy with driving as well the UI has to be designed in such a manner that it will not have very complex operations for the user that take various steps. Instead, the parts that require user interaction must be large enough, easily locatable on the phone screen, and respond within a short amount of time (less than a minute).

- Since the user will be driving meanwhile, the application must not cause unnecessary distraction or irritate the user. This will be achieved by avoiding using irritating alert sounds, having a simple UI layout, and decreasing the need for user input (such as automatically closing pop-ups after a while).

3.3.4. Privacy

- Confidential information such as contact numbers, destination information, or exact route data will not be used with third parties to respect user privacy.
- To minimize the risk of exposure of personal information in a possible attack, confidential data will be kept encrypted.

3.4 Pseudo Requirements

- RoadVisor must have a high response time since it needs to accurately show the direction information. For this reason, making the necessary computations beforehand or within the mobile side without fetching it from the cloud environment may be considered.

- The application must consider the space and functionality limitations of mobile phones and must be able to operate as expected by acknowledging such limitations. The overall hardware systems in mobile phones must be taken into account in this respect.

- RoadVisor has to access the camera and other necessary device information. The application has to be compatible with Android devices.

- RoadVisor has to be compatible with the selected map application such as Yandex or Google Maps.

- The application has to be making synchronous computations and therefore selected ML and computer vision models need to be able to regard this issue.

3.5 System Models

3.5.1 Scenarios

Use case name	Navigate To Destination
Participating actors	Initiated by User Communicates with Google Maps
Flow of events	1. The User selects 'travel to a destination' option. 2. The User then enters the information about the destination. 3. If the destination is valid, the directions are fetched from Google Maps. 4. The directions are then displayed using Augmented Reality.

Entry Conditions	● The location info of the User is turned on. ● The User is logged in. ● The camera is turned on.
Exit Condition	● The user terminates the application or cancels navigation.

Use case name	Enable Sign Detection
Participating actors	Initiated by User
Flow of events	1. The User enables Sign Detection using the sign detection option. 2. The system uses the images sent by the camera to process the information.. 3. If traffic signs are detected, the System alerts the User through the screen.
Entry Conditions	● The camera of the User is turned on. ● The User is logged in.
Exit Condition	● The user terminates the application or selects the Disable button.

Use case name	Enable Traffic Light Detection
Participating actors	Initiated by User
Flow of events	1. The User enables Traffic Light Detection using the button. 2. The System uses the images sent by the camera to process the information.. 3. If Traffic Lights are detected, the System alerts the User.
Entry Conditions	● The camera of the User is turned on. ● The User is logged in.
Exit Condition	● The user terminates the application or selects the Disable button.

Use case name	Enable Crowd Density Analysis
Participating actors	Initiated by User
Flow of events	1. The User selects Enable Crowd Density Analysis. 2. The system checks if the camera is open, in case it will request the User to turn it on. 3. The System then notifies the User about the color of Traffic Lights when detected.
Entry Conditions	● The camera of the User is turned on. ● The User is logged in.
Exit Condition	● The user terminates the application or cancels navigation.

Use case name	Log In
Participating actors	Initiated by User Communicates with the remote Database
Flow of events	1. The User selects the Login option. 2. The System displays a screen to fetch User's username and password. 3. The User enters the info and selects the sign in option. 4. The System checks the entered information. If the information is validated, the user is signed in.
Entry Conditions	● The device has internet connection
Exit Condition	● The user terminates the application or cancels navigation.

3.5.2 Use-Case Model

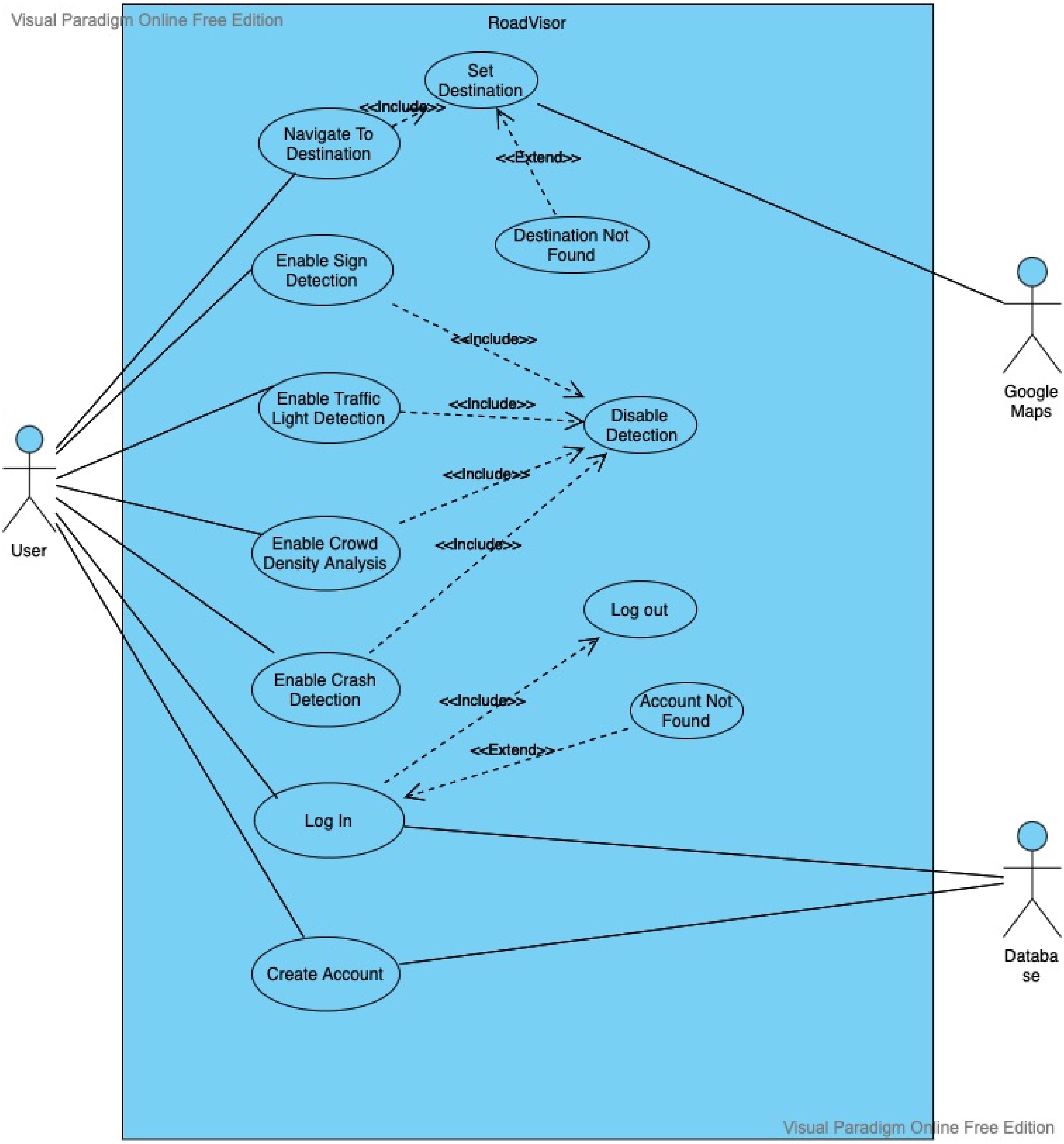

Figure 3.1 *Use Case Diagram of RoadVisor*

3.5.3 Object and Class Model

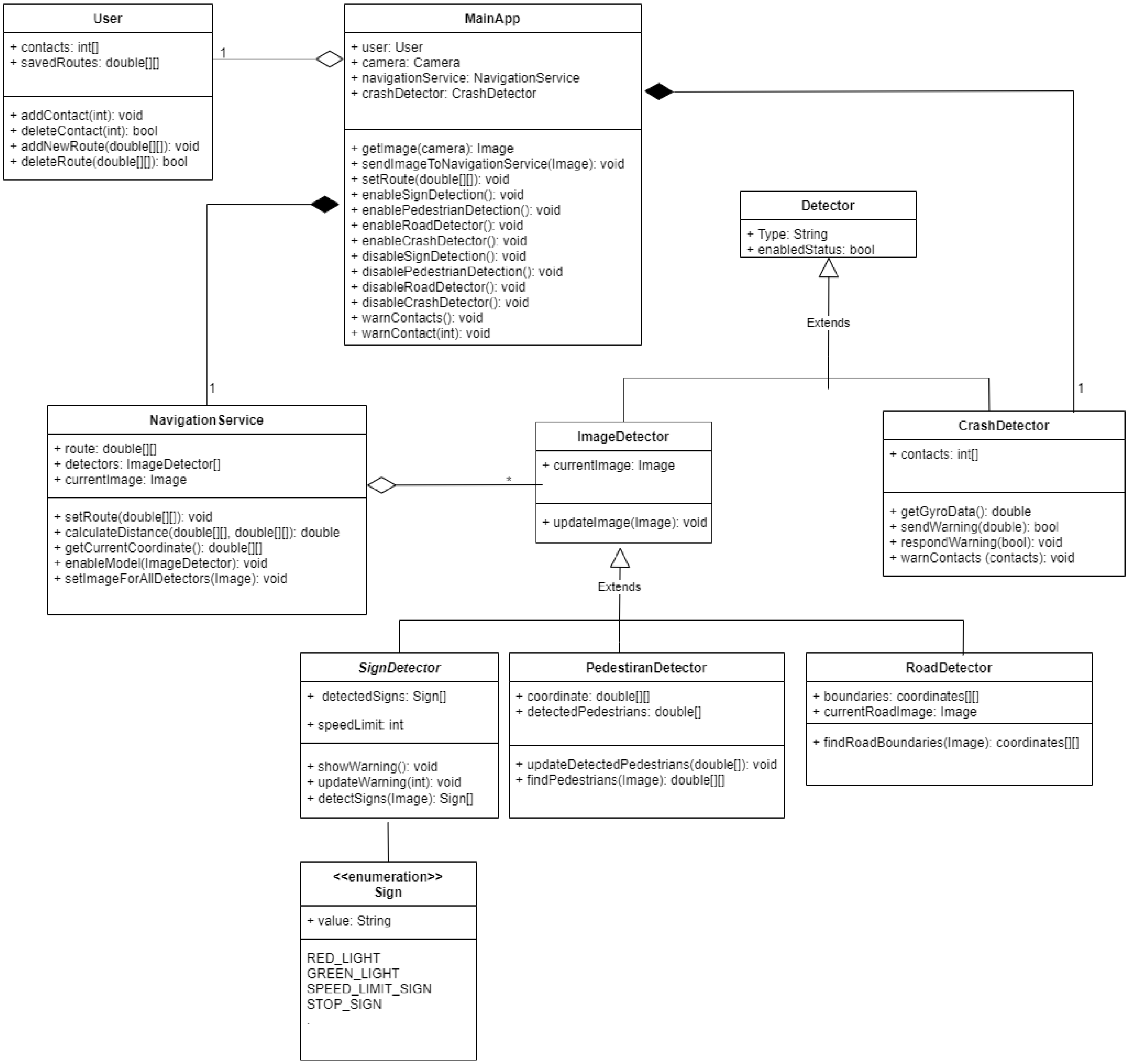

Figure 3.2 *Object and Class Diagram of RoadVisor*

3.5.4 Dynamic Models

3.5.4.1 Sequence Diagrams

<u>Scenario 1: Enabling/Disabling Features</u>

In this scenario, the actor is the user. The user toggles the activation for the desired feature of the application such as enabling or disabling the traffic light detection. In accordance with the user's input, the system starts or stops processing that functionality.

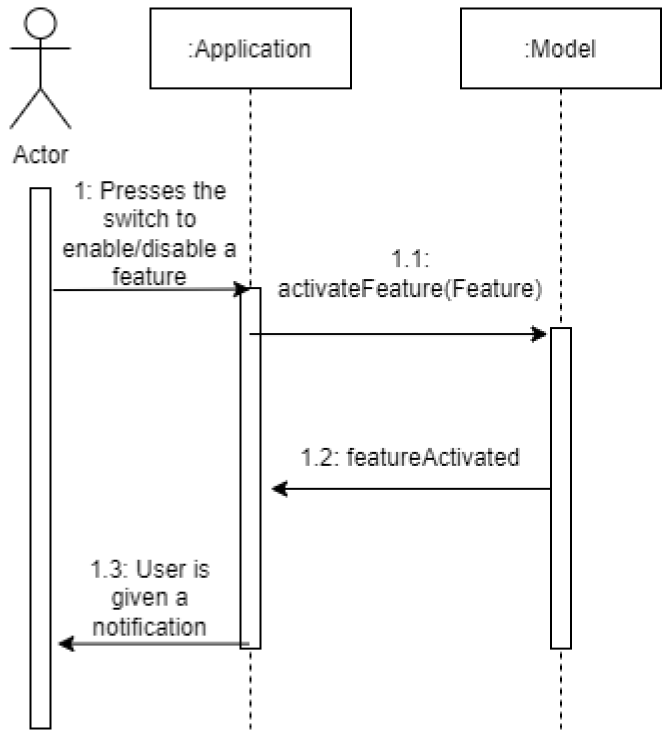

Figure 3.3 *Sequence Diagram for Scenario 1.*

Scenario 2: Setting Up Navigation to Destination

In this scenario, the actor is the user. The user sets a destination through the application and the application navigates the user to the set destination using Google Maps. If the given destination is not found, the application throws an error.

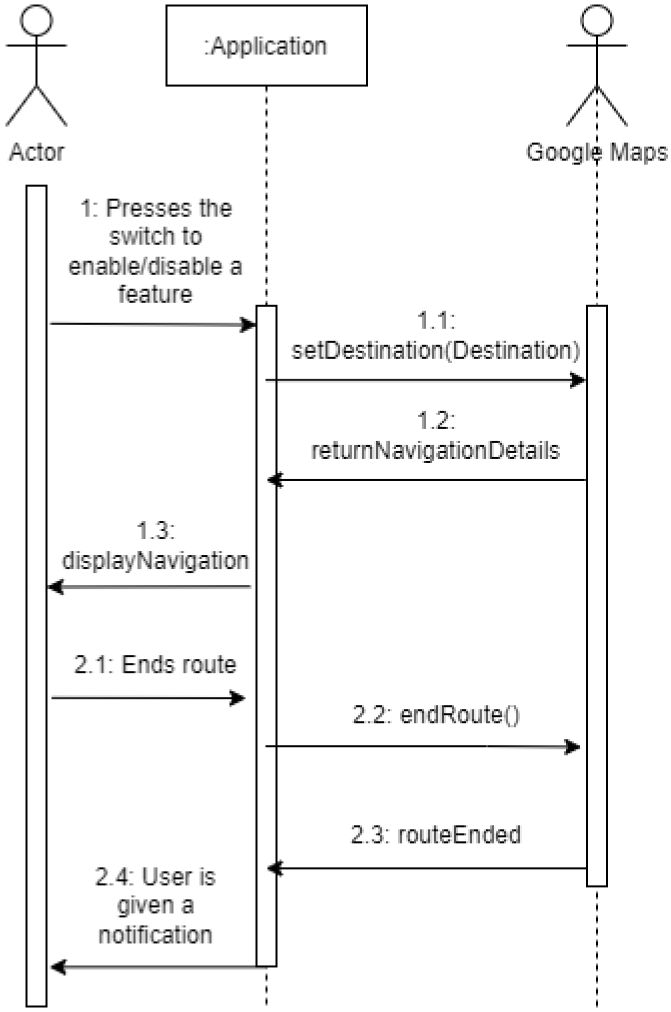

Figure 3.4 *Sequence Diagram for Scenario 2.*

Scenario 3: Car Crash Detection

In this scenario, the actor is the user. When the user gets involved in a crash, the application detects it and then sends the user a notification if there is a media player playing in the background, the application pauses the media player. If the user does not respond to the notification the application sends in a designated amount of time, the application alerts the emergency services.

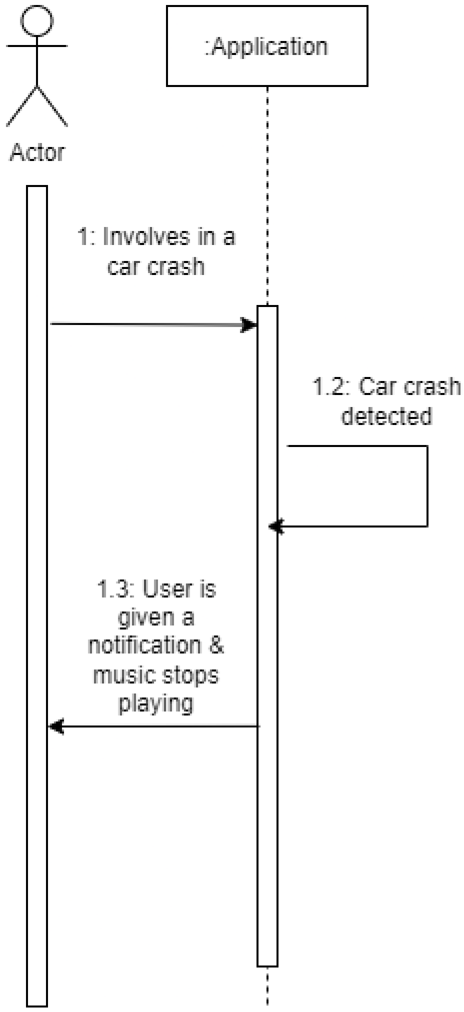

Figure 3.5 *Sequence Diagram for Scenario 3.*

3.5.4.2. State Diagrams

Scenario 1: Enabling/Disabling Features

This scenario depicts a generalization for activation and deactivation of available features. Since almost all of our features only need to user to activate or deactivate features (as the user will be driving as well and wanting complex permission from the user would be unsafe), this generalized state diagram captures the essence of all basic features where the states correspond to that functionality being active such as detecting pedestrians. All features start as enabled when the user first launches the app. All features terminate on quitting the app as depicted in the diagram

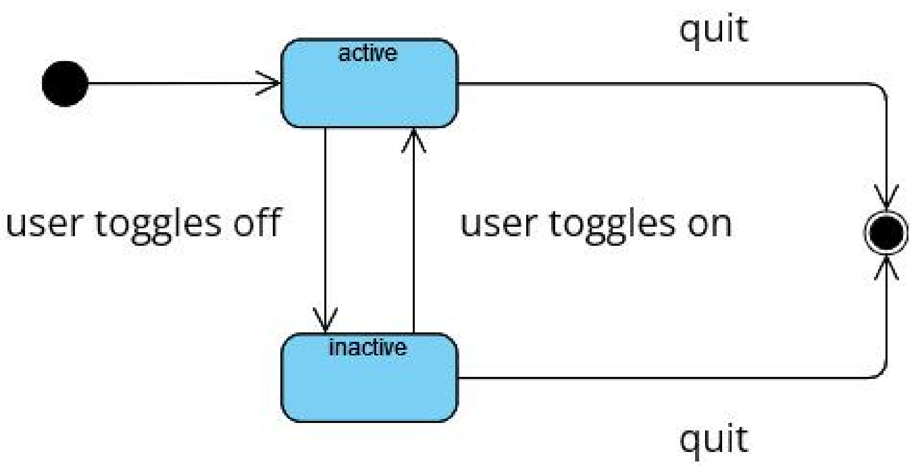

Figure 3.6 *State Diagram for Scenario 1.*

Scenario 2: Setting Up Navigation to Destination

This scenario depicts the AR navigation feature overall. When the feature is activated, the application will request the destination information from the user as described in the waiting condition in Route Selection state. The user may either disable the feature for termination state or select a route. After the route being selected by the user, navigation will be active until the user reaches the destination or disables the feature.

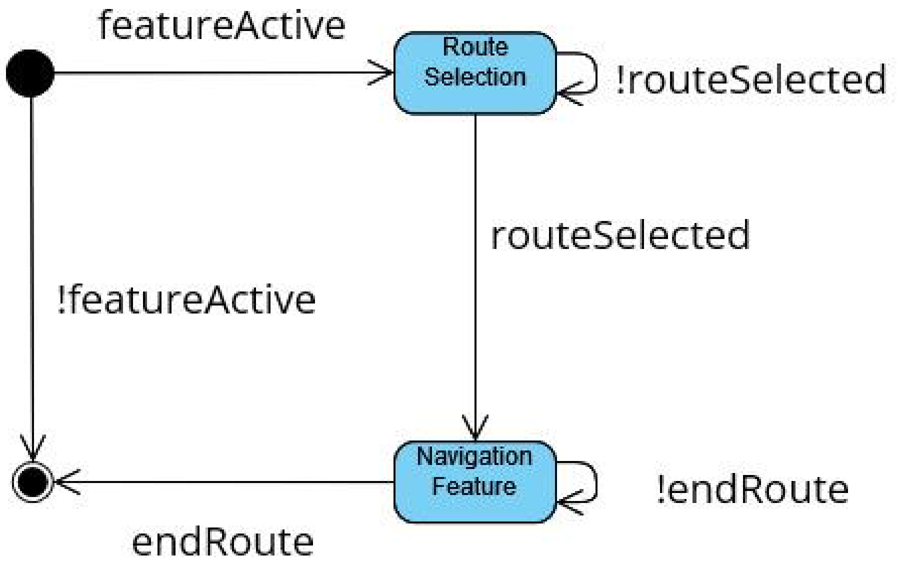

Figure 3.7 *State Diagram for Scenario 2.*

3.5.4.3. Activity Diagrams

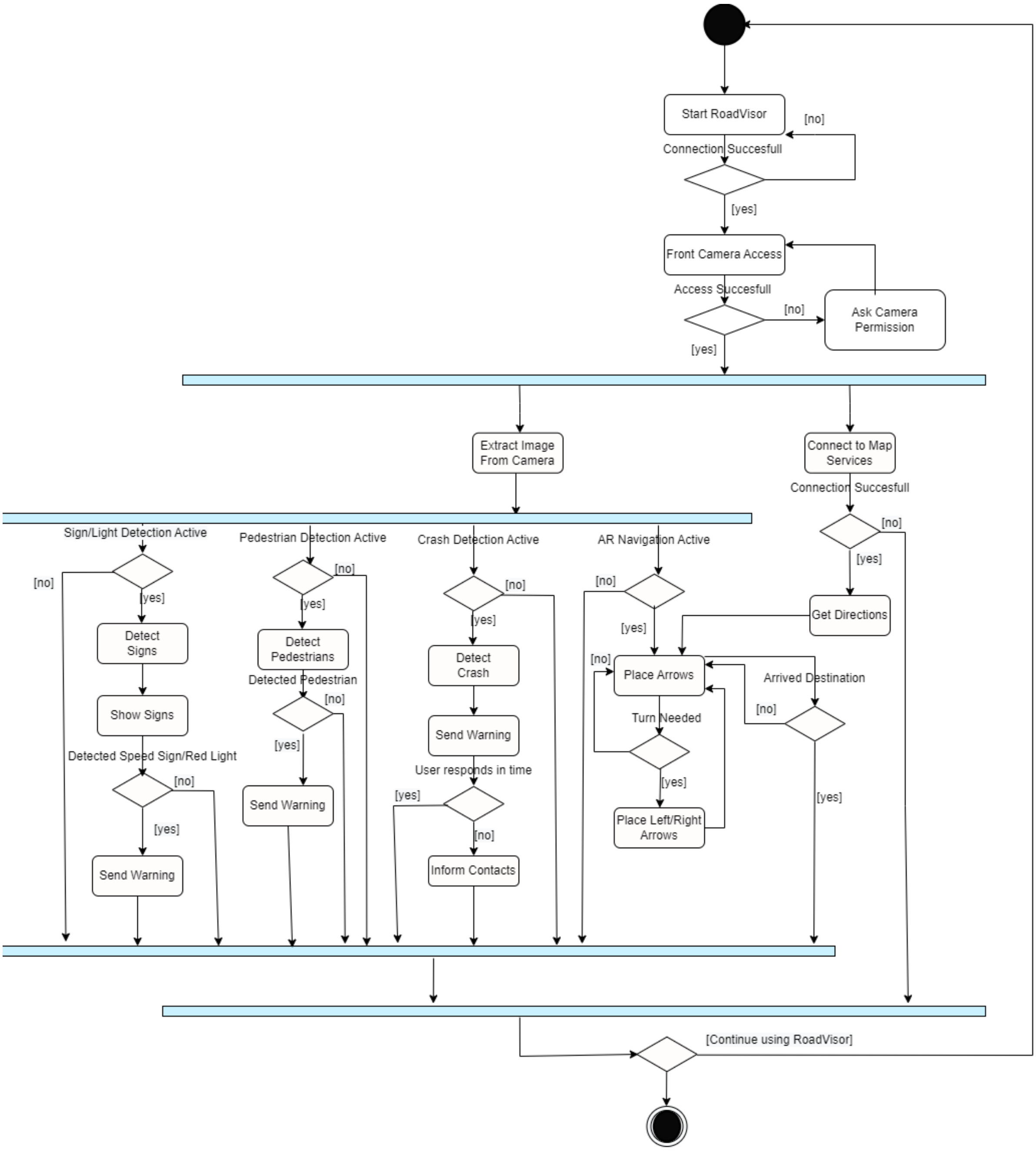

Figure 3.8 *Activity Diagram for RoadVisor*

3.5.5 User Interface

The following user interface screen mock-ups show the screens for each feature:

Figure 3.9 *Augmented Reality Navigation [11]*

This is an example screen when the augmented reality navigation feature will be used. The screen will be simple as possible to ensure there are no distractions. Therefore only required information will be shown, such as the AR elements itself and the location information.

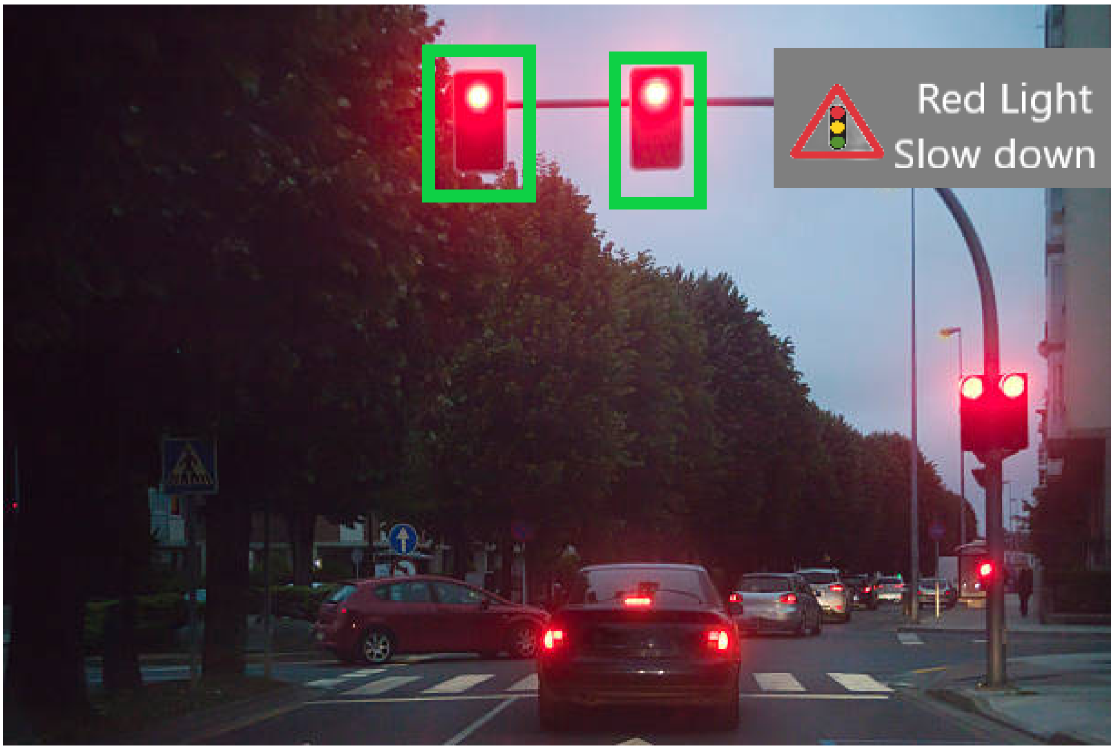

Figure 3.10 *Traffic Light Detection [12]*

RoadVisor detects traffic lights and warns drivers to slow down. A warning message is shown to confirm the lights are detected, and a sound is played. This process happens in a similar way for traffic signs also.

Figure 3.11 *Traffic Sign Detection [13]*

Again, a message is displayed, and a sound is played to warn the driver.

For pedestrian detection, the application will warn the user about possible danger and mark the person on the screen. The warning message will be shown on the top right (similar to other features), and a sound will be played.

These mock-up screens do not represent the final work of the RoadVisor. They are given as a concept and to demonstrate a possible user interface. In the final work the user interface can be changed as new elements can be added or deleted.

4 Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

As a project that aims to bring an innovative solution to a real and substantial problem, it is necessary to analyze the RoadVisor's effect on both people and the world itself. As a candidate application to assist drivers by using augmented reality in navigation, one of the important aspects that came to mind is the safety and therefore the health of the drivers. Additionally, any new innovation might cause change globally. This can be represented in sustainable and social aspects. Therefore, in this analysis process, public health, public safety, public welfare, global factors, cultural factors and social factors were the main aspects that were essential to be considered. Each of these factors is analyzed in detail in their own subtopics. Lastly, a table is provided to show the effect level and a brief description of every factor.

4.1.1 Public Health and Public Safety

RoadVisor contains features that can affect public health and safety. Traffic lights and sign detection feature plays an important role if the driver does not be careful about these signs during driving the car. Sign detection mainly focuses on the “STOP” signs. If the driver passes these signs without stopping or approaches very fast, they need to be warned. Similarly, if the driver does not decrease their speed after seeing the red light or passes through the red light this cause a threat to public safety and health where accidents may happen.

The Pedestrian/Crowd detection feature will be beneficial to warn the drivers if there is a person or group of people on the road. By giving appropriate warnings, it is aimed that the driver becomes more cautious about their road surroundings.

However, while applying these features, RoadVisor must not distract the driver's attention as the main priority of the driver should be concentrating on the road. Any element that can divert the driver's attention may cause accidents to happen. Therefore, we noticed that public health and public safety factors are one of the most important factors in our analysis process.

4.1.2 Public Welfare

Main goal of the RoadVisor is to bring Augmented Reality Navigation and other important infotainment features to mobile phones. Augmented Reality Navigation is the key feature of the RoadVisor and this feature is only available on high-end cars today [10]. Therefore, if the user of this application does not own a car that is in high segments, they can still use these features without getting a new car. We believe that this will save users money as the features will be available on their mobile phones. Hence, RoadVisor has an important part in protecting the public welfare.

4.1.3 Global Factors

Providing an Augmented Reality supported Navigation feature will help drivers to find their exact path. This is different from 2D navigation applications that are currently used widely. In our analysis, we saw that drivers sometimes can misinterpret the user interface of these navigation applications, therefore, missing the path they were supposed to go as a result. Because the RoadVisor combines computer vision elements with the real world, this misinterpretation can be eliminated. In the end, drivers can reach their destinations faster. This leads to fewer carbon emissions caused by cars, contributing to global climate change positively.

4.1.4 Cultural Factors

Each country has different traffic laws. This situation also affects the types of traffic signs. There might be additional signs that are not available in Turkey or vice versa. If these additional signs are commonly used in these countries, it is also important for RoadVisor to detect these signs with high accuracy. Additionally, road lines might have slight differences between countries as well. Lastly, the signs and any road indicator related to RoadVisor can be written in different languages. In order for the RoadVisor to properly work in different countries. These cultural factors have to be emphasized in the development process.

4.1.5 Social Factors

As stated in the global factors, it is possible that the drivers can reach their destinations quickly with RoadVisor. This means less time is spent on the road and this leads to a decrease in the time lost while travelling. Therefore the drivers and passengers can allocate more time to their daily lives. Additionally, with the driver assistant features, RoadVisor can help to decrease car accidents.

Table 4.1: Factors that can affect analysis and design.

	Effect level	Effect
Public Health and Public Safety	9	Detection algorithms should work with high accuracy. This can protect public health and safety by ensuring less accidents.
Public Welfare	8	RoadVisor’s proposed features currently exist in high-end cars. Users don’t need to spend their money on such expensive cars for these features.
Global Factors	6	It is important for the augmented reality navigation feature to provide an understandable UI as it can decrease travelling times and carbon emissions.
Cultural Factors	5	Different traffic signs/indicators and different languages in these signs require additional analysis.
Social Factors	7	Augmented reality navigation feature can be important in user’s daily life as it can decrease travelling times.

4.2 Risks and Alternatives

In our meetings with both our advisor and innovation expert we recognized possible major risks. In the analysis process we discussed about their possible alternatives and came up with a Plan B. Below topics list all these risks with our alternative plan and there is a table to sum up all these factors.

4.2.1 Hardware Limitations of the Mobile Phones

The main bottleneck we discovered with this project is the hardware limitations of mobile phones. Our project requires the execution of various machine learning and neural network models. Normally, these processes work best if they are run using a GPU because it allows parallel computing. However, mobile phone processors execute instructions consecutively, and this decreases application performance.

4.2.2 Connection with the Maps Application API

RoadVisor will require a connection with the Maps API (Google Maps or Yandex Maps) in order to get the direction and navigation information. However, it is not guaranteed that the mobile connection works in every place, so users can experience connection problems which can cause the augmented reality navigation feature not to work. Additionally, these APIs require payment after a certain amount of request queries. The risk here is that if RoadVisor requires lots of information frequently, a payment has to be made by the developers. This is an economic burden for the developers, and it directly affects the application usability.

4.2.3 Accuracy of the Models in Different Conditions

Implementing machine learning and neural networks models requires datasets. These datasets are often acquired in optimal conditions where there is no rain, there is daylight and roads have proper lines. However, not all roads have lines. Some may deteriorate over time, and some do not have it from the beginning. Additionally, the application may be required to use in rainy conditions or at night. These conditions are far away from the optimal conditions and the application accuracy (detection of signs and pedestrians, proper placement of AR elements in navigation) can be affected negatively.

4.2.4 Phone Placement and Camera Quality

In order for the RoadVisor to work properly, the camera of the mobile phone needs to see the road at a proper angle and this requires proper phone placement in the car. Additionally, the camera quality has to ensure that the image quality is clean enough to be processed. These situations are directly linked to the user. If the user cannot place the phone accordingly or if the phone does not have enough requirements, the application may not work properly.

4.2.5 Plan B

With these risks are considered, the Plan B must be planned to eliminate these risks without risking the driver’s health and safety. These are the proposed methods to ensure this requirement:

- At most two features will be usable at the same time. With the usage of machine learning and neural network models, there is a high chance that these models cannot be optimized within the given project deadline. Without lightweight models, executing every feature will decrease performance drastically. Affected performance might also affect the response time of other features which is dangerous for the driver. This proposed method might also be extended to allow only one feature to be used simultaneously.

- When the connection is lost with the API, RoadVisor's primary goal is not to put the driver and the passengers in danger. Therefore not showing incorrectly positioned AR elements is crucial. Some APIs restrict the map information to be cached beforehand. If this is not the case with the chosen API, the directions will be saved and GPS information will be used to track the process. If it is the case, lastly created AR element will be stored in a cache. After that, the driver will be warned about the connection being lost.
- To work in different conditions, RoadVisor team will train datasets with the same or different conditions. However, augmented reality navigation may perform poor in these conditions as it will require semantic segmentation, a process that requires enough light. The application can limit the usage of this feature if the certain conditions are not met. Datasets for the other features in these conditions are more easy to find but if the accuracy is not enough, the limitation can be extended to these features as well to ensure the safety and health of the driver and passengers.
- If the phone placement is not done correctly the application will warn the user and give instructions to help the user to place it properly. Additionally, if the camera quality is not enough for the features, the features again can be limited with the priority of safety.

Table 4.2: Risks and Alternatives.

Risk	Effect on the Project	Plan B
Hardware Limitations of Mobile Phones	All the proposed features might not work properly simultaneously.	Limit the number of features to work at the same time to two. If the performance is not enough, limit it to one.
Connection with the Maps Application API	Augmented Reality Navigation feature loses its direction info and the API requires payment.	Cache the direction information beforehand or some AR elements. Warn the user about the connection.
Accuracy of the Models in Different Conditions	ML and neural networks models might not work accurately in some conditions.	Train different datasets. If the accuracy and performance are not enough, limit affected features to ensure safety.
Phone Placement and Camera Quality	Application might not work accurately if the phone is not placed properly and if the camera quality is bad.	Warn and give instructions to the user for proper placement. If the camera quality is not enough, limit some or all features to ensure safety.

4.3 Project Plan

Table 4.2: List of work packages

WP#	Work package title	Leader	Members involved
WP1	Project Specification Report	Arda İçöz	Emin Berke Ay, Nurettin Onur Vural, Ahmet

			Faruk Ulutaş, Ammaar Iftikhar
WP2	Analysis Report	Emin Berke Ay	Arda İçöz, Nurettin Onur Vural, Ahmet Faruk Ulutaş, Ammaar Iftikhar
WP3	Detailed Design Report	Nurettin Onur Vural	Emin Berke Ay, Arda İçöz, Ahmet Faruk Ulutaş, Ammaar Iftikhar
WP4	Final Report	Ammaar Iftikhar	Emin Berke Ay, Nurettin Onur Vural, Ahmet Faruk Ulutaş, Arda İçöz
WP5	Computer Vision Model Development & Testing	Ammaar Iftikhar	Nurettin Onur Vural, Ahmet Faruk Ulutaş
WP6	Traffic Sign Detection Model Development	Emin Berke Ay	Nurettin Onur Vural, Arda İçöz
WP7	Traffic Light Detection Model Development	Ahmet Faruk Ulutaş	Ammaar Iftikhar, Arda İçöz
WP8	Crowd Density Analysis Model Development	Nurettin Onur Vural	Emin Berke Ay, Ammaar Iftikhar
WP9	Crash Detection Model Development	Arda İçöz	Nurettin Onur Vural, Ahmet Faruk Ulutaş
WP10	UI/UX Design	Emin Berke Ay	Arda İçöz
WP11	Backend Development	Ahmet Faruk Ulutaş	Nurettin Onur Vural, Ammaar Iftikhar
WP12	Frontend Development	Arda İçöz	Emin Berke Ay
WP13	Maps Integration	Ammaar Iftikhar	Ahmet Faruk Ulutaş, Emin Berke Ay
WP14	Component and Unit Testing	Ahmet Faruk Ulutaş	Ammaar Iftikhar, Arda İçöz
WP15	Computer Vision Model Integration	Nurettin Onur Vural	Ahmet Faruk Ulutaş, Ammaar Iftikhar

WP 1: *Project Specification Report*			
Start date: *Week 1* **End date:** *Week 4*			
Leader:	*Arda İçöz*	**Members involved:**	*Emin Berke Ay, Nurettin Onur Vural, Ahmet Faruk Ulutaş, Ammaar Iftikhar*
Objectives: *Project Specification Report needs to be completed. The required information needs to be provided in the report under relevant sections and subsections.*			

Tasks: ***Task 1.1 <Task Division Among Team Members>*** **:** *Tasks should be divided equally among the team members.* ***Task 1.2 <Report Writing>*** **:** *Each team member should write their part of the report.* ***Task 1.3 <Proofreading>*** **:** *Proofreading needs to be done before submitting the report*
Deliverables ***D1.1:*** *Project Specification Report* ***D1.2:*** *Innovation Expert Form*

WP 2: *Analysis Report*			
Start date: *Week 5* **End date:** *Week 8*			
Leader:	*Emin Berke Ay*	**Members involved:**	*Arda İçöz, Nurettin Onur Vural, Ahmet Faruk Ulutaş, Ammaar Iftikhar*
Objectives: *Analysis Report needs to be completed. The required information needs to be provided in the report under relevant sections and subsections.*			
Tasks: ***Task 1.1 <Task Division Among Team Members>*** **:** *Tasks should be divided equally among the team members.* ***Task 1.2 <Report Writing>*** **:** *Each team member should write their part of the report.* ***Task 1.3 <Proofreading>*** **:** *Proofreading needs to be done before submitting the report*			
Deliverables ***D1.1:*** *Project Specification Report*			

WP 3: *Detailed Design Report*			
Start date: *Week 8* **End date:** *Week 18*			
Leader:	*Nurettin Onur Vural*	**Members involved:**	*Arda İçöz, Emin Berke Ay, Ahmet Faruk Ulutaş, Ammaar Iftikhar*
Objectives: *Detailed Design Report needs to be completed. The required information needs to be provided in the report under relevant sections and subsections.*			
Tasks: ***Task 1.1 <Task Division Among Team Members>* :** *Tasks should be divided equally among the team members.* ***Task 1.2 <Report Writing>* :** *Each team member should write their part of the report.* ***Task 1.3 <Proofreading>* :** *Proofreading needs to be done before submitting the report*			
Deliverables ***D1.1:*** *Detailed Design Report*			

WP 4: *Final Report*			
Start date: *Week 19* **End date:** *Week 34*			
Leader:	*Ammaar Iftikhar*	**Members involved:**	*Arda İçöz, Nurettin Onur Vural, Ahmet Faruk Ulutaş, Emin Berke Ay*
Objectives: *Final Report needs to be completed. The required information needs to be provided in the report under relevant sections and subsections.*			
Tasks:			

Task 1.1 <Task Division Among Team Members> **:** *Tasks should be divided equally among the team members.* ***Task 1.2 <Report Writing>*** **:** *Each team member should write their part of the report.* ***Task 1.3 <Proofreading>*** **:** *Proofreading needs to be done before submitting the report*
Deliverables ***D1.1:*** *Final Specification Report*

WP 5: *Computer Vision Model Development & Testing*			
Start date: *Week 8* **End date:** *Week 22*			
Leader:	*Ammaar Iftikhar*	**Members involved:**	*Nurettin Onur Vural,* *Ahmet Faruk Ulutaş*
Objectives: *Various Computer Vision models need to be developed and tested. The most accurate and efficient models are to be used during the actual implementation.*			
Tasks: ***Task 1.1 <Task Division Among Team Members>*** **:** *Tasks should be divided equally among the team members.* ***Task 1.2 <Computer Vision Model Testing>*** **:** *All of the models need to be tested.*			
Deliverables ***D1.1:*** *A brief summary about all of the models tested.* ***D2.2:*** *The source codes of models.*			

WP 6: *Traffic Sign Detection Model Development*
Start date: *Week 8* **End date:** *Week 22*

Leader:	*Emin Berke Ay*	**Members involved:**	*Nurettin Onur Vural, Arda İçöz*
Objectives: *Computer Vision models regarding the traffic sign detection need to be developed and tested. The most accurate and efficient models are to be used during the actual implementation.*			
Tasks: ***Task 1.1 <Task Division Among Team Members>* :** *Tasks should be divided equally among the team members.* ***Task 1.2 <Computer Vision Model Testing>* :** *All of the models regarding the traffic sign detection need to be tested.*			
Deliverables ***D1.1:*** *A brief summary about all of the models tested.* ***D2.2:*** *The source codes of models.*			

WP 7: *Traffic Light Detection Model Development*			
Start date: *Week 8* **End date:** *Week 22*			
Leader:	*Ahmet Faruk Ulutaş*	**Members involved:**	*Ammaar Iftikhar, Arda İçöz*
Objectives: *Computer Vision models regarding the traffic light detection need to be developed and tested. The most accurate and efficient models are to be used during the actual implementation.*			
Tasks: ***Task 1.1 <Task Division Among Team Members>* :** *Tasks should be divided equally among the team members.*			

***Task 1.2 <Computer Vision Model Testing>* :** *All of the models regarding the traffic light detection need to be tested.*
Deliverables ***D1.1:*** *A brief summary about all of the models tested.* ***D2.2:*** *The source codes of models.*

WP 8: *Crowd Density Analysis Model Development*			
Start date: *Week 8* **End date:** *Week 22*			
Leader:	*Nurettin Onur Vural*	**Members involved:**	*Emin Berke Ay, Ammaar Iftikhar*
Objectives: *Computer Vision models regarding the crowd density analysis need to be developed and tested. The most accurate and efficient models are to be used during the actual implementation.*			
Tasks: ***Task 1.1 <Task Division Among Team Members>* :** *Tasks should be divided equally among the team members.* ***Task 1.2 <Computer Vision Model Testing>* :** *All of the models regarding the crowd density analysis need to be tested.*			
Deliverables ***D1.1:*** *A brief summary about all of the models tested.* ***D2.2:*** *The source codes of models.*			

WP 9: *Crash Detection Analysis Model Development*

Start date: *Week 8* **End date:** *Week 22*			
Leader:	*Arda İçöz*	**Members involved:**	*Nurettin Onur Vural,* *Ahmet Faruk Ulutaş*
Objectives: *Computer Vision models regarding the crash detection analysis need to be developed and tested. The most accurate and efficient models are to be used during the actual implementation.*			
Tasks: ***Task 1.1 <Task Division Among Team Members>*** **:** *Tasks should be divided equally among the team members.* ***Task 1.2 <Computer Vision Model Testing>*** **:** *All of the models regarding the crash detection analysis need to be tested.*			
Deliverables ***D1.1:*** *A brief summary about all of the models tested.* ***D2.2:*** *The source codes of models.*			

WP 10: *UI/UX Design*			
Start date: *Week 8* **End date:** *Week 34*			
Leader:	*Emin Berke Ay*	**Members involved:**	*Arda İçöz*
Objectives: *UI/UX Design of the mobile application needs to be drawn. Detailed drawings are required at this stage in order to start implementing the frontend layer of the mobile application. This stage can take up to week 34 to finish since at any point, an improvement can be made to the designs and this design change can dramatically affect the frontend layer of the application.*			
Tasks:			

Task 1.1 <Task Division Among Team Members> **:** *Tasks should be divided equally among the team members.* ***Task 1.2 <Design Drawing>*** **:** *UI/UX designs should be drawn using third party software.* ***Task 1.3 <Approving the Designs>:*** *All of the designs need to be approved.*
Deliverables ***D1.1:*** *UI/UX Designs of the mobile application.*

WP 11: *Backend Development*			
Start date: *Week 8* **End date:** *Week 34*			
Leader:	*Ahmet Faruk Ulutaş*	**Members involved:**	*Nurettin Onur Vural,* *Ammaar Iftikhar*
Objectives: *Backend layer of the application needs to be developed. All of the necessary functions, classes, components need to be implemented.*			
Tasks: ***Task 1.1 <Task Division Among Team Members>*** **:** *Tasks should be divided equally among the team members.* ***Task 1.2 <Implementation>*** **:** *Backend layer of the application needs to be implemented.* ***Task 1.3 <Approving the Code>:*** *All of the code needs to be approved by other team members.*			
Deliverables ***D1.1:*** *Source code for the backend layer of the application.*			

WP 12: *Frontend Development*			
Start date: *Week 8* **End date:** *Week 34*			
Leader:	*Arda İçöz*	**Members involved:**	*Emin Berke Ay*
Objectives: *Frontend layer of the application needs to be developed. All of the necessary functions, classes, components need to be implemented with the help of UI/UX designs.*			
Tasks: ***Task 1.1 <Task Division Among Team Members>*** **:** *Tasks should be divided equally among the team members.* ***Task 1.2 <Implementation>*** **:** *Frontend layer of the application needs to be implemented with the help of UI/UX designs.* ***Task 1.3 <Approving the Code>:*** *All of the code needs to be approved by other team members.*			
Deliverables ***D1.1:*** *Source code for the frontend layer of the application.*			

WP 13: *Maps Integration*			
Start date: *Week 8* **End date:** *Week 34*			
Leader:	*Ammaar Iftikhar*	**Members involved:**	*Ahmet Faruk Ulutaş,* *Emin Berke Ay*
Objectives: *Map application of choice needs to be implemented into the mobile application. It will serve as the navigation application and the augmented reality feature will rely on the chosen map application to properly display the arrows on the screen.*			
Tasks:			

Task 1.1 <Task Division Among Team Members> **:** *Tasks should be divided equally among the team members.* ***Task 1.2 <Integration>*** **:** *Maps integration of the mobile application needs to be implemented with.* ***Task 1.3 <Approving the Code>:*** *All of the code needs to be approved by other team members.*
Deliverables ***D1.1:*** *Source code for the frontend layer of the application and the successful integration of the maps application of choice to the mobile application.*

WP 14: *Component and Unit Testing*			
Start date: *Week 11* **End date:** *Week 34*			
Leader:	*Ahmet Faruk Ulutaş*	**Members involved:**	*Ammaar Iftikhar, Arda İçöz*
Objectives: *Component and Unit testing of the mobile application needs to be done. This is a crucial step of the process since a successful execution of this step will ensure that the application is working as intended.*			
Tasks: ***Task 1.1 <Task Division Among Team Members>*** **:** *Tasks should be divided equally among the team members.* ***Task 1.2 <Testing>*** **:** *Various component, unit and also bug testing needs to be done thoroughly.*			
Deliverables ***D1.1:*** *Approval of the team members on mobile application working as intended.*			

D2.2: *If needed, fixes on required functions, components of files.*

WP 15: *Computer Vision Model Integration*			
Start date: *Week 11* **End date:** *Week 34*			
Leader:	*Nurettin Onur Vural*	**Members involved:**	*Ahmet Faruk Ulutaş,* *Ammaar Iftikhar*
Objectives: *Chosen computer vision models need to be integrated into the mobile application. This work package should start after the WP5 since in WP5 we plan on developing and testing various computer vision models.*			
Tasks: ***Task 1.1 <Task Division Among Team Members>*** **:** *Tasks should be divided equally among the team members.* ***Task 1.2 <Integration>*** **:** *Chosen computer vision models need to be implemented.*			
Deliverables ***D1.1:*** *Source code of the mobile application with computer vision models integrated.*			

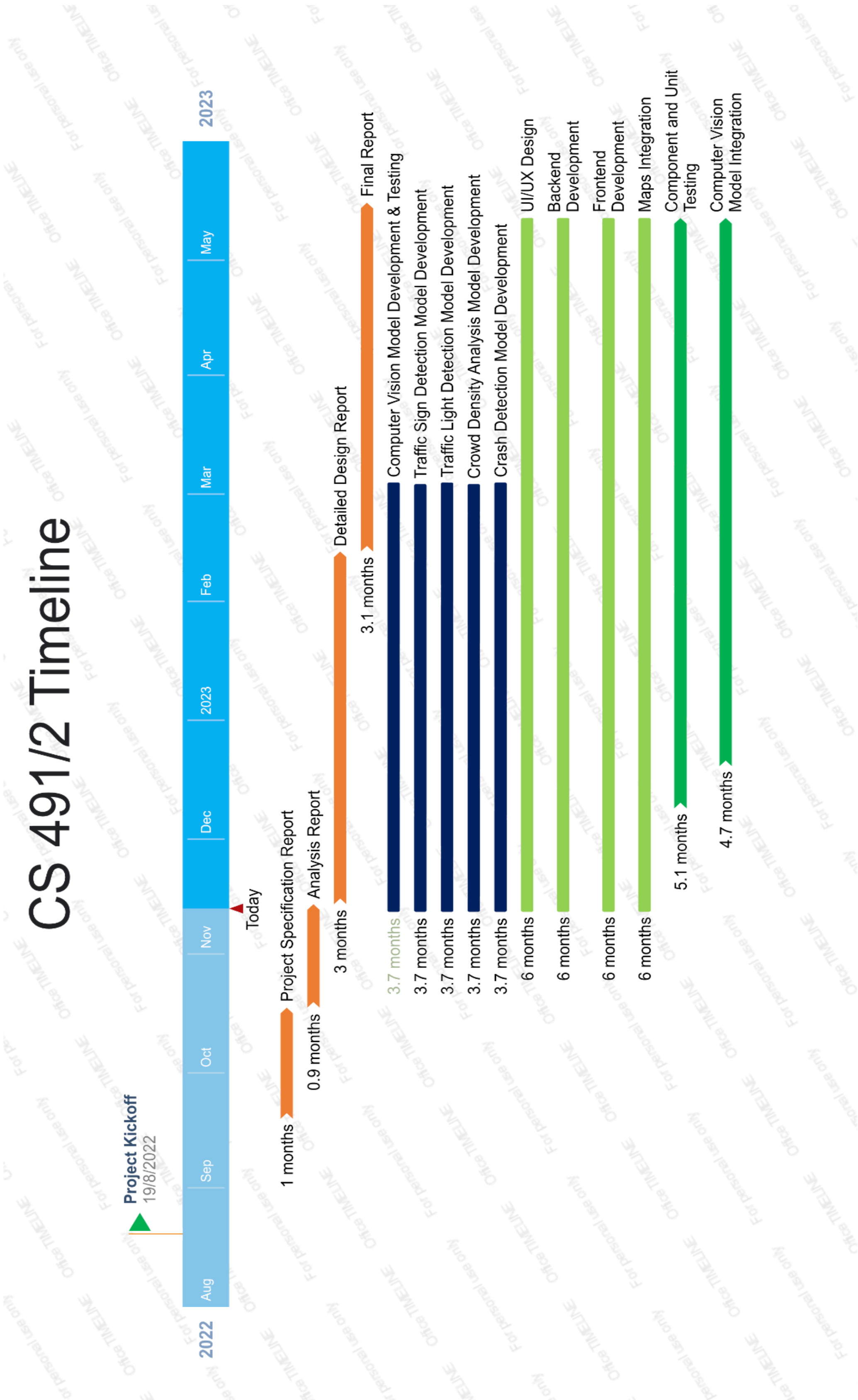

***Figure 4.1** Gantt Diagram of RoadVisor.*

4.4 Ensuring Proper Teamwork

The first thing we did to create a working environment in which every member of the group can participate equally was to open a Discord server. In this Discord server, we created text channels with different topics. Especially when finding a project topic, every member came up with an idea and posted his idea with enough detail in these channels. The same process happened in writing reports or doing research for a specific topic. We decided to use Discord as our digital communication method as it helped us to create an equal environment. It is also very beneficial for us to store our findings as the text messages are stored in a timeline. So, it is easy to find and it is possible to access these findings on every platform.

For report writing, we created a Google Drive directory. We divided the sections of the reports among group members. In this Drive, a Google Docs file is opened for every report and group members wrote their part. If a member was unable to continue their part, we quickly communicated and rearranged the topics. We will continue using Google Docs as our main report-writing platform because it stores the files in the cloud and allows edits on the files simultaneously for every group member.

For the implementation, we created work packages with different tasks. We emphasized the equal distribution of group leaders among these packages so that everybody has equal or very similar workloads. We will use GitHub as our base platform for version control. GitHub is a reliable platform for such work and every group member is familiar with how Git system and GitHub work.

Lastly, as our course instructors requested, every group member has opened a personal logbook. In these logbooks, every member writes the things they did for the project. It can be reading an academic paper or just thinking about a problem. Every logbook represents a physical evidence for the process of every group member. This makes it easy for both course instructors and our supervisor to track our process.

4.5 Ethics and Professional Responsibilities

- RoadVisor will require location services in order to provide accurate navigation. This location information will be shared directly with the navigation application. RoadVisor will only need this information to place the augmented reality elements in its navigation feature. The location of the user will not be stored anywhere and thus used later. Consent of the user will be required initially for the above location service requirements. The application will not work otherwise.
- Images retrieved from the camera of the phone will only be used for AR an computer vision tasks. The images will not be stored and therefore shared with any third party.
- The navigation information provided by the application and the placement of the AR elements might not be accurate depending on the location and condition. RoadVisor does not guarantee that AR elements are supported in every road type and location. Therefore the driver is responsible for their and the passenger’s safety.
- Pedestrian detection, traffic lights/signs detection and crash detection features might not be accurate and work in every condition. The driver is responsible for their and the passenger's safety.
- The application is recommended to be used while the phone is charging. Users will be informed about this effect on battery life.

4.6 Planning for New Knowledge and Learning Strategies

The project we are doing involves technologies and concepts not every member of team is familiar with and comfortable. In order to successfully implement our project, every team member need to learn something new along the course of the project.

First of all, every team member needs to get comfortable with computer vision models. Some of us have experience with machine learning and artificial intelligence before, but some of us have no experience in those subjects. Therefore, a threshold level of knowledge and experience level should be set, and every team member needs to surpass that level of knowledge.

Next, since we are going to develop an application for Android devices, experience with Android devices and Android Studio is required. For this purpose, every team member should know the basic principles of mobile application development and be familiar with common learning strategies for this purpose. Similarly, team members should be comfortable with some level of both backend and frontend development.

Our main learning strategy will be learning via online resources. We will utilise websites, videos, and documentations of the technologies we plan to use. To support our newly gathered knowledge, we will utilise papers in the literature world. By reading and discussing them, we will have a better understanding of the context.

5 References

[1] T. Shang, H. Lu, P. Wu, and Y. Wei, "Eye-tracking evaluation of exit advance guide signs in highway tunnels in familiar and unfamiliar drivers," International Journal of Environmental Research and Public Health, vol. 18, no. 13, p. 6820, 2021.

[2] "Ten types," Doblin, 21-Apr-2020. [Online]. Available: https://doblin.com/ten-types. [Accessed: 16-Oct-2022].

[3]"Ar Home," *Google AR & VR | AR home*. [Online]. Available: https://arvr.google.com/ar/. [Accessed: 12-Nov-2022].

[4] "Augmented reality," *Yandex.Maps for mobile. Help*. [Online]. Available: https://yandex.com/support/m-maps/concept/ar.html. [Accessed: 12-Nov-2022].

[5] Amusi, "Amusi/Awesome-lane-Detection: A paper list of Lane detection.," *GitHub*. [Online]. Available: https://github.com/amusi/awesome-lane-detection. [Accessed: 12-Nov-2022].

[6] afkim14, "Road-boundary-detection/FINAL_PROJECT_DATA at master · AFKIM14/road-boundary-detection," *GitHub*. [Online]. Available: https://github.com/afkim14/Road-Boundary-Detection/tree/master/final_project_data. [Accessed: 12-Nov-2022].

[7] Dineshresearch, "Dineshresearch/novel-deep-learning-model-for-traffic-sign-detection-using-capsule-networks: Capsule Networks that achieves outstanding performance on the German Traffic Sign Dataset," *GitHub*. [Online]. Available: https://github.com/dineshresearch/Novel-Deep-Learning-Model-for-Traffic-Sign-Detection-Using-Capsule-Networks. [Accessed: 12-Nov-2022].

[8] N. J. Karthika and S. Chandran, "Addressing the false positives in pedestrian detection," *Lecture Notes in Electrical Engineering*, pp. 1083–1092, 2020.

[9] Alex-Lechner, "Alex-Lechner/traffic-light-classification: A detailed tutorial on how to build a traffic light classifier with tensorflow for the capstone project of Udacity's self-driving car engineer nanodegree program.," *GitHub*. [Online]. Available: https://github.com/alex-lechner/Traffic-Light-Classification. [Accessed: 12-Nov-2022].

[10] "Benz EQC: Mbux augmented reality navigation," *Mercedes*. [Online]. Available: https://www.mercedes-benz.co.uk/passengercars/mercedes-benz-cars/models/eqc/design.pi.html/mercedes-benz-cars/models/eqc/design/comfort-gallery/augmented-video. [Accessed: 13-Nov-2022].

[11] M. K. Sharma, “Augmented reality navigation,” *International Journal of Engineering Research and*, vol. V9, no. 06, 2020.

[12] “Street view , red traffic and car lights in the foreground, city...,” *iStock*. [Online]. Available: https://www.istockphoto.com/tr/foto%C4%9Fraf/street-view-red-traffic-lights-city-life-gm544795666-98020471?utm_medium=organic&utm_source=google&utm_campaign=iptcurl. [Accessed: 13-Nov-2022].

[13] Carmen, “Intersections: Pedestrian-controlled traffic lights just confuse everyone,” *BC Driving Blog*, 30-Mar-2021. [Online]. Available: https://drivinginstructorblog.com/intersection-navigator-pedestrian-controlled-traffic-lights/. [Accessed: 13-Nov-2022].