Bilkent University
CS 491 Senior Design Project I
Fall 2022-2023

Analysis Report

RoadVisor

Group: T2317

Group Members
Ahmet Faruk Ulutaş - 21803717
Ammaar Iftikhar - 21901257
Arda İçöz - 21901443
Emin Berke Ay - 21901780
Nurettin Onur Vural - 21902330

Supervisor
Asst. Prof. Dr. Hamdi Dibeklioğlu

Innovation Expert
Cem Çimenbiçer

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 informs 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 [7]. 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:

3. Proposed System

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 features 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.

6. Song Recommendation Based On Road: The feature is meant to be for the entertainment of the user. The feature will play a song based on the type of road that the user will be driving on. For example, it will play metal on the highway or jazz in a high-traffic area. The type of music in what circumstance can be configured by the user.

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.

3.5.1 Use Case Diagrams

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.

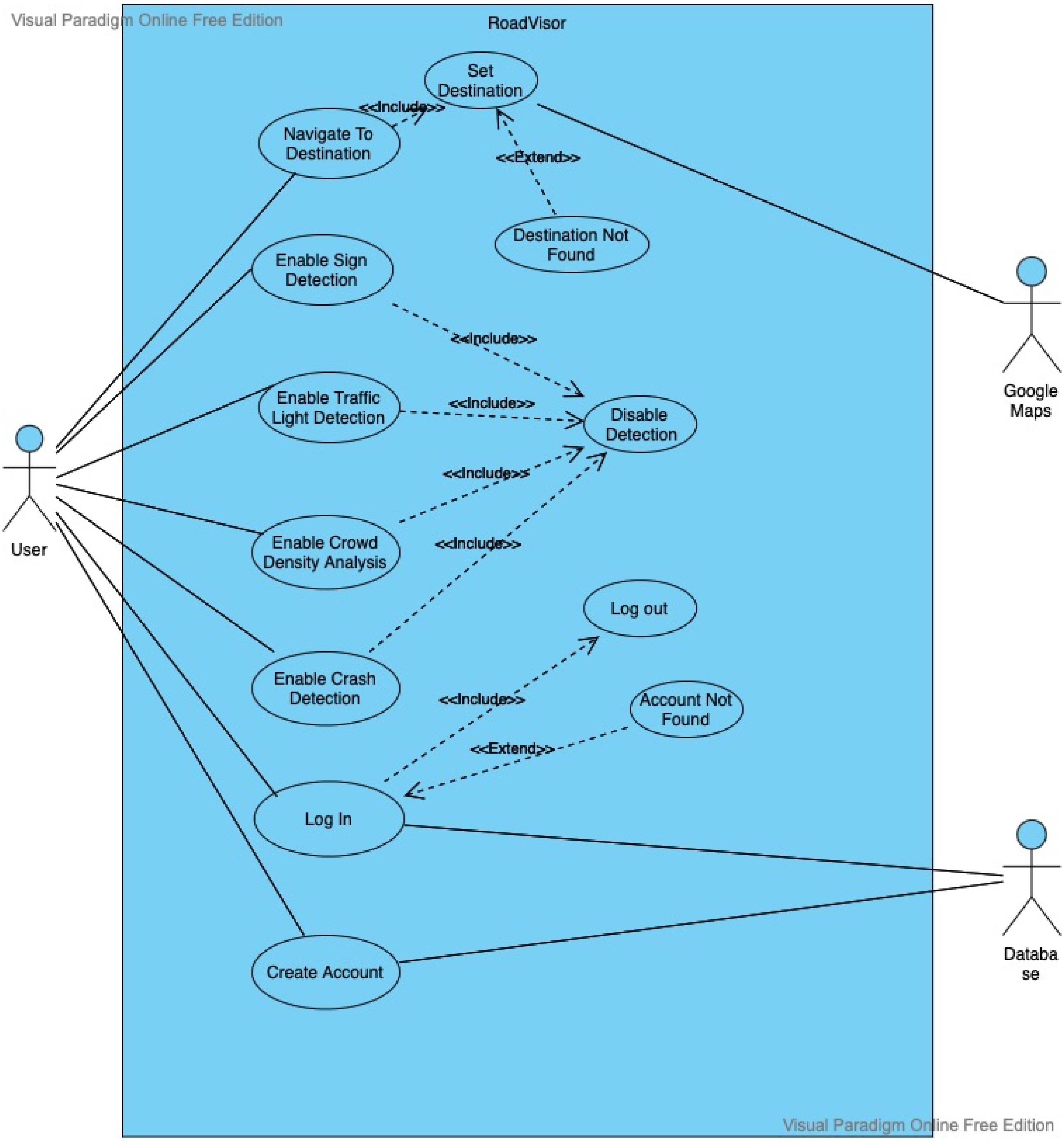

Fig 1. The Use Case diagram of application

Initial structure

3.5.3. Object and Class Model

3.5.3.1. Sequence Diagrams

Scenario 1: Enabling/Disabling Features

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.

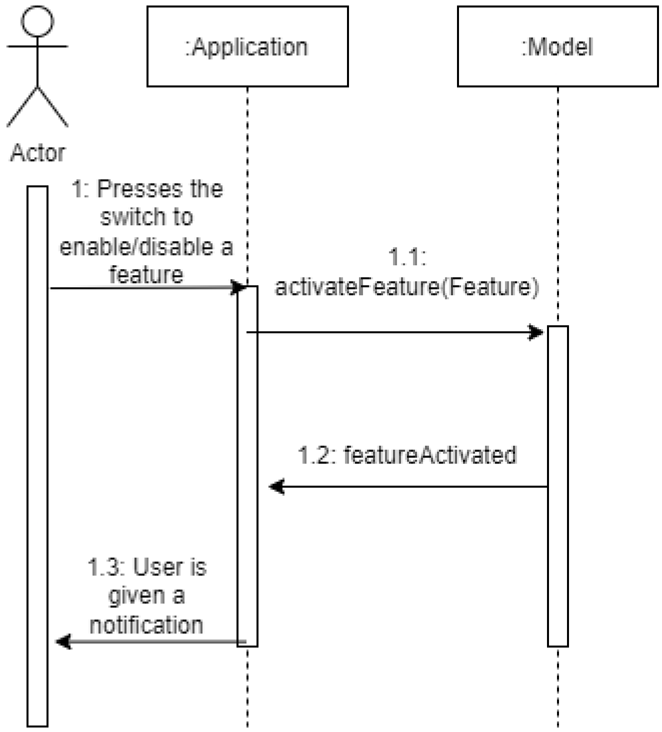

Figure 2. *Sequence Diagram for Scenario 1.*

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.

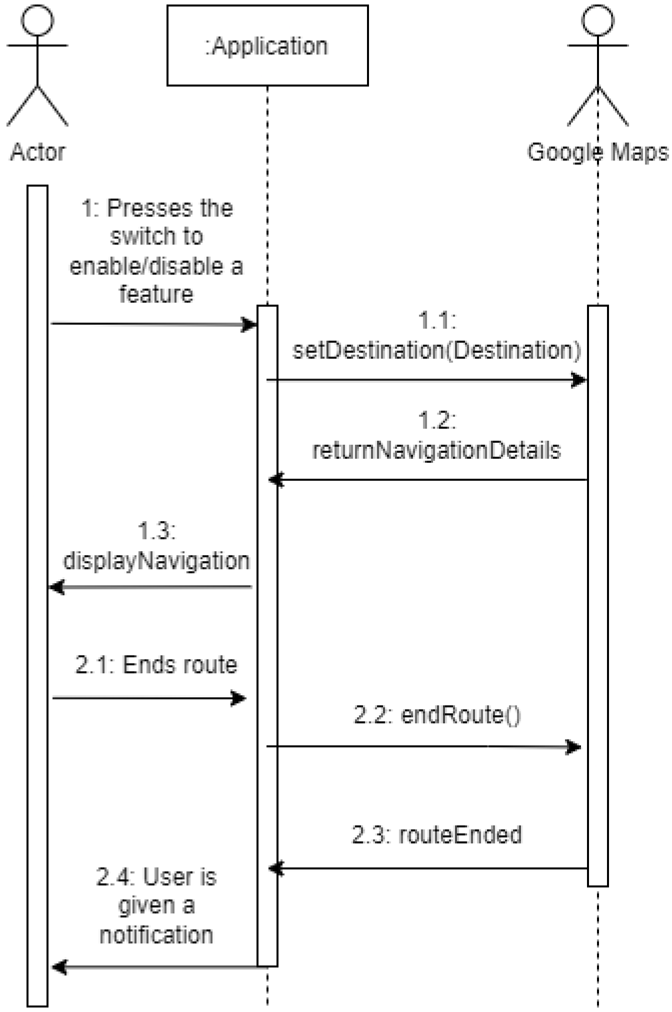

Figure 3. *Sequence Diagram for Scenario 2.*

3.5.3.2. State Diagrams

3.5.4. Dynamic Models

3.5.5. User Interface - Navigational Paths and Screen Mock-ups

4. Other Analysis Elements

4.1. Consideration of Various Factors

4.1.1 Data Privacy

4.1.2 Economic Factors

4.1.3. Effect Of Cultural Differences

4.1.4. Health and Distraction Concerns

4.1.5. Upcoming Mobile Device Designs

4.1.6. Evaluation of the Constraints

4.2. Risks and Alternatives

4.3. Project Plan

4.4. Ensuring Proper Team Work

4.5. Ethics and Professional Responsibilities

4.6. Planning for New Knowledge and Learning Strategies

5. Glossary

6. 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.