**COMSATS Institute of Information Technology,**
**Park Road, Chak Shahzad, Islamabad Pakistan**

# Third Eye

## (Driver Assistance System)

***By***

**Syed Ali Kazim** **CIIT/SP13-BSE-079/ISB**

**Abdullah Akber** **CIIT/SP13-BSE-002/ISB**

***Supervisors***

**Dr. Mubeen Ghafoor**

**Dr. Aimal Tariq Rextin**

***Bachelor of Science in Software Engineering (2013-2017)***

**The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.**

COMSATS Institute of Information Technology,
Park Road, Chak Shahzad, Islamabad Pakistan

# Third Eye

## (Driver Assistance System)

**A project presented to**
**COMSATS Institute of Information Technology, Islamabad**

**In partial fulfillment**
**of the requirement for the degree of**

***Bachelors of Science in Software Engineering (2013-2017)***

**By**

| | |
|---|---|
| **Syed Ali Kazim** | **CIIT/SP13-BSE-079/ISB** |
| **Abdullah Akber** | **CIIT/SP13-BSE-002/ISB** |

# DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Syed Ali Kazim                                        Abdullah Akber

--------------------------                            --------------------------

# CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (SE) Third Eye (Driver Assistance System) was developed by **Syed Ali Kazim (CIIT/SP13-BSE-079)** and **Abdullah Akber (CIIT/SP13-BSE-002)** under the supervision of Dr. Mubeen Ghafoor and co supervision of Dr. Aimal Tariq Rextin and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Software Engineering.

**--------------------------------------**

**Supervisor**

**--------------------------------------**

**External Examiner**

**--------------------------------------**

**Head of Department**

**(Department of Computer Science)**

# Executive Summary

Accident research shows that most of the accidents on roads are caused by driver’s negligence and his inability to foresee the situation that leads to the accident.

Driver assistance system analyses environment at real time such as oncoming cars and unexpected departure of lane. And gives reliable feedback to the driver to prevent any potential accident.

Third eye is developed with computer vision technologies to provide assistance while driving vehicles on roads. Third eye runs on image processing algorithms that provide reliable lane tracking. It recognizes lanes marking on the road and alerts the driver with visual and audio warning when the driver departs from his lane.

Third eye also monitors the road to detect and track vehicles ahead, when the oncoming vehicle reaches critical distance, the app generates audio and visual warning to alert the driver, so that he takes the necessary measures to prevent potential collision.

Third eye system has an additional application that can be used to track current location of driver using GPS. Remote app will ping the main application to request its location and current picture of roadway.

Demand for driver assistance system is expected to rise over the next decade as people have to commute long distance to reach to their office or education institute.

# Acknowledgement

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Dr. Mubeen Ghafoor” and our co-supervisor “Dr. Aimal Tariq Rextin”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Syed Ali Kazim Abdullah Akber

-------------------------- --------------------------

# Abbreviations

| | |
|---|---|
| **SRS** | Software Require Specification |
| **PC** | Personal Computer |
| **SDS** | Software Design Document |
| **GPS** | Global Positioning System |
| **OpenCV** | Open Source Computer Vision Library |
| **SMS** | Short Message Service |
| **OOP** | Object Oriented Programming |
| **SRS** | Software Requirements Specification |
| **SDS** | Software Design Specification |

# Table of Contents

# List of Figures

# 1 Introduction

This chapter will present and discuss the project. First, it will cover the introduction of the project, and then it will move on to relevance and background of this project. The latter part has covered the literature that was studied to gain domain knowledge of the system and the end of this chapter mentions the software development methodology for this project along with the rationale behind selected methodology.

## 1.1 Brief

Third eye is a smartphone based driver assistance system that makes use of rear camera of phone to capture real time video and process it to provide assistance to the driver of vehicle. The application is developed on Android Studio. The algorithm of the applications is developed and designed in MATLAB and implemented in C++/native to achieve enhanced performance and processing speed of this app.

## 1.2 Relevance to Course Modules

There are number of courses which came handy in development of this project. The knowledge that we gained from Software Requirements Engineering and Software Design and Architecture was very helpful in the documentation of the project. On the other hand, concepts of OOP and introduction to programming helped us to develop the Android application in native. Web engineering course gave us necessary information on how to implement web server.

## 1.3 Project Background

The idea behind this project was to process real time video feed to implement a driver assistance system. The project revolves around the computer vision algorithm that is used to process digital images. Basic aim of digital image processing is to extract information from images, after extracting the data from the images, this data can be used to make decisions, and can also apply operations on the extracted data.

## 1.4 Literature Review

Following are some existing systems that were studied for literature review:

### Drivea:

Drivea is an android app that monitors roads for unusual activities including potential accidents and lane departure, it also issues warnings if the driver exceeds speed limit. Another feature of Drivea is that it takes picture where the car is parked in a parking lot for the future reference of driver, this app is not perfect and it is not designed to replace driver’s vigilance from the road.

## Augmented driving:

Augmented driving is a real time object detection app and over speed avoidance app, it monitors the distance from the preceding vehicle and issues warning if the distance is critical, this app also supports multiple languages, if driver is stuck in traffic jam, this app detects moving off traffic in front of car and issues audio command for the driver.

## 1.5 Analysis from Literature Review

Vehicle detection and tracking applications are under research for many years now, the main objective of this field of study is to investigate vision based intelligent systems that can be used to extract useful and precise information about roadways and vehicles.

In the domain of digital image processing number of techniques are being used to process images in order to extract information. After extracting the data from the images, this data can be used to make decisions. Different operations can also be applied on the extracted data.

Monitoring roads and its surrounding by smartphone camera can give us useful advantages for road safety and driver assistance. Although it is an emerging field of study these days but as the growth in the development and research of digital image processing looks likely to continue, this technology will become increasingly more affordable and easier to use. The future seems rich for road safety and driver assistance applications.

## 1.6 Methodology and Software Lifecycle for This Project

Incremental software development methodology was followed to develop the Third Eye application. The system was divided into modules and each module passed through the phase of development and testing. Then the next module was pursued.

Object oriented design methodology is adapted for the development of Third Eye Application.

### 1.6.1 Rationale behind Selected Methodology

Incremental methodology helped to focus on one module at a time. It allowed detecting defects at an early stage, and upon development of each module it was very helpful to get reliable feedback from the supervisor and to track progress of the project.

Object oriented approach allows implementing interface with the help of already existing services in the library. This extent of flexibility is not possible with procedural language. It is easy to decouple interface with the functionality of the system.
Categorizing system into classes decreases the chances of complete shutdown of system if any part stops working. Each part of system is independent and decoupled from the rest of the

system. Since developing in JAVA which is pure OOP and using OpenCV which is also OOP based

# 2 Problem Definition

There is a need for a cheap solution that can monitor the road and alert driver to revert his attention to critical event on the road and to cater a situation that can lead to accident.

## 2.1 Problem Statement

Every day on average 15 people die in traffic accidents across Pakistan according to Pakistan Bureau of Statistics data from 2004 to 2013. These traffic accidents are majorly due to driver's negligence to provide undivided attention on the road and more importantly drivers are often distracted by moving and stationary objects on the roadside. The most common reason of driver's distraction is either visual (e.g. driver not looking on the road) or manual (e.g. texting on mobile phone) or cognitive (e.g. thinking about matters while answering a phone call). So, there is a need to keep the attention of the driver focused at the task of driving to ensure driver safety.

## 2.2 Deliverables and Development Requirements

### 2.2.1 Software Requirements Specification Document

Requirement elicitation and analysis was the first phase of this project. Requirements were collected from the drivers, supervisor and the existing applications. Considering all the requirements, a formal software requirements specification document was created. All the requirements were evaluated and accepted by the stakeholders, this SRS document served as blueprint of whole system.

### 2.2.2 Software Design Specification

Second deliverable was software design specification, which helped to understand interaction of different components of system. The SRS tell what to develop and this document explains how to develop it. It also serves to specify architecture of design and underlying processes.

### 2.2.3 Software Test Document

After development, each module was tested against its use cases. Once the use cases were validated, they were documented and presented to the stakeholders.

## 2.3 Current System

### iOnRoad:

iOnRoad is an app to help driver in long drives, using smartphone camera it provides number of features to assist the driver, most notable features of iOnRoad app are as follows:

- Collision warning
- Lane detection
- Black Box recording
- It also provides feedback to the driver including number of miles driven.

# 3 Requirement Analysis

## 3.1 Use Cases Diagram

### 3.1.1 Main Application

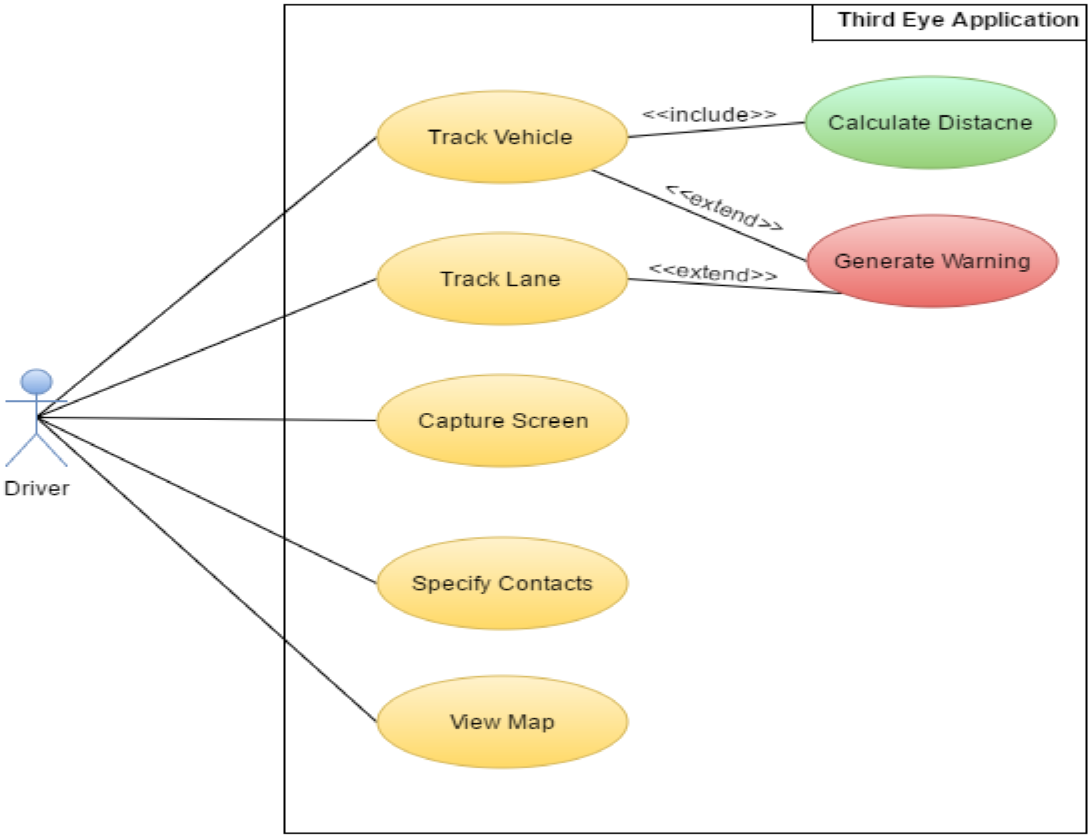


**Figure 1 Main App Use-case**

### 3.1.2 Remote Application

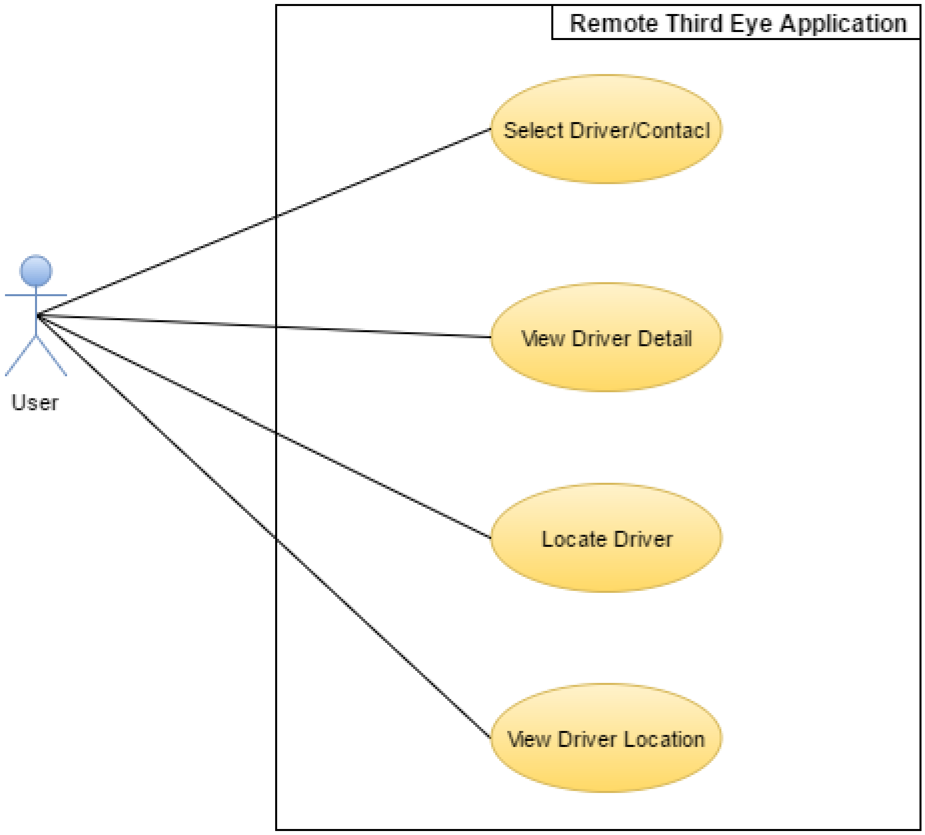


**Figure 2 Remote App Use-case**

## 3.2 Detailed Use Case

| Id | Primary Actor | Title |
|---|---|---|
| UC-001 | Driver | Start Application |
| UC-002 | Driver | Track Vehicle |
| UC-003 | Driver | Track Lane |
| UC-004 | Driver | Capture Screen |
| UC-005 | Driver | View Map |
| UC-006 | Driver | Specify Contacts |
| UC-007 | Driver | Calculate Distance |
| UC-008 | Driver | Generate Warning |
| UC-009 | Remote App User | Locate Driver |
| UC-010 | Remote App User | Display Driver Location |

## 3.2.1 Use Case Name

### Main Application

#### *3.2.1.1 Start/Launch Application*

| | |
|---|---|
| **Use Case ID:** | UC-001 |
| **Use Case Name:** | Start Application |
| **Actors:** | Driver |
| **Description:** | The user launches the application to start tracking vehicle and lanes and get driver assistance. |
| **Trigger:** | User taps on the application icon in the app drawer of the smartphone. |
| **Preconditions:** | 1. Driver has mounted the smartphone on the windshield of the car horizontally.<br>2. Application is installed in the smartphone. |
| **Post conditions:** | 1. Application starts tracking vehicles and lane from live video feed.<br>2. Application generates warnings related to vehicle and lane tracking to alert the driver. |
| **Normal Flow:** | 1. Driver taps on the application icon available in the app drawer of the smartphone.<br>2. System displays the startup activity.<br>3. User click on “Start Tracking” button.<br>4. System displays camera view of on main interface from rear camera of phone.<br>5. System captures the live video feed frame by frame and detects vehicle and lane.<br>6. System detects vehicle and lane in multiple consecutive frames for tracking.<br>7. System calculates distance of detected vehicle and displays it on the main interface.<br>8. System generates audio and visual warnings at critical distance or when lane changes. |
| **Alternative Flows:** | NA |
| **Exceptions:** | 6a If vehicle is detected and smartphone is on silent<br>1. System will not generate audio warning.<br>2. Smartphone will vibrate.<br>3. Use case terminates. |
| **Includes:** | Calculate Distance. |
| **Special** | NA |

| | |
|---|---|
| **Requirements:** | |
| **Assumptions:** | NA |
| **Notes and Issues:** | NA |

### 3.2.1.2 Track Vehicle

| | |
|---|---|
| **Use Case ID:** | UC-002 |
| **Use Case Name:** | Track Vehicle |
| **Actors:** | Driver |
| **Description:** | The user can select or deselect to track vehicle from settings. |
| **Trigger:** | User taps on the settings icon on the main interface. |
| **Preconditions:** | 1. Driver has mounted the smartphone on the windshield of the car horizontally.<br>2. Application is running. |
| **Post conditions:** | 1. Application starts tracking vehicle from live video feed.<br>2. Application generates warnings related to vehicle tracking. |
| **Normal Flow:** | 1. Driver taps on the settings icon given on the main interface at the top left corner.<br>2. System displays the settings screen.<br>3. User selects or enables the track vehicle option by sliding the toggle switch.<br>4. User taps the back button to go back to main interface.<br>5. System displays the main interface and starts tracking vehicle. |
| **Alternative Flows:** | 3a. In step 3 of the normal flow, if the track vehicle option is already enabled<br>1. User disables the track vehicle option by sliding toggle switch.<br>2. User taps the back button to go back to main interface.<br>3. Use case terminates.<br><br>3b. In step 3 of the normal flow, if the track vehicle option is already enabled<br>1. Use case resumes on step 4. |
| **Exceptions:** | None |
| **Includes:** | Calculate Distance. |
| **Special Requirements:** | NA |
| **Assumptions:** | NA |
| **Notes and Issues:** | NA |

### 3.2.1.3 Track Lane

| | |
|---|---|
| **Use Case ID:** | UC-003 |
| **Use Case Name:** | Track Lane |
| **Actors:** | Driver. |
| **Description:** | The user can select or deselect to track road lane from settings. |
| **Trigger:** | User taps on the settings icon on the main interface. |
| **Preconditions:** | 1. Driver has mounted the smartphone on the windshield of the car horizontally.<br>2. Application is running. |
| **Post conditions:** | 1. Application starts tracking road lane from live video feed.<br>2. Application generates warnings related to lane tracking. |
| **Normal Flow:** | 1. Driver taps on the settings icon given on the main interface at the top left corner.<br>2. System displays the settings screen.<br>3. User selects or enables the track lane option by sliding the toggle switch.<br>4. User taps the back button to go back to main interface.<br>5. System displays the main interface and starts tracking road lane. |
| **Alternative Flows:** | 3a. In step 3 of the normal flow, if the track lane option is already enabled<br>1. User deselects or disables the track lane option by sliding toggle switch.<br>2. User taps the back button to go back to main interface.<br>3. Use case terminates.<br>3b. In step 3 of the normal flow, if the track lane option is already enabled<br>1. Use case resumes on step 4. |
| **Exceptions:** | NA |
| **Includes:** | NA |
| **Special Requirements:** | NA |
| **Assumptions:** | NA |
| **Notes and Issues:** | NA |

### 3.2.1.4 Capture Screen

| | |
|---|---|
| **Use Case ID:** | UC-004 |
| **Use Case Name:** | Capture Screen |
| **Actors:** | Driver |
| **Description:** | The user can capture the screen of the main interface while the application is tracking vehicle and lane |
| **Trigger:** | User taps on the camera icon given on the main interface. |
| **Preconditions:** | 1. Driver has mounted the smartphone on the windshield of the car horizontally.<br>2. Application is running. |
| **Post conditions:** | 1. The application captures the screen and stores it in user's smartphone gallery. |
| **Normal Flow:** | 1. User taps on the camera icon given on the main interface at the middle left side.<br>2. System captures the image of the main interface screen.<br>3. System stores the captured image in smartphones image gallery.<br>4. System notifies the user with a toast "Captured and Saved". |
| **Alternative Flows:** | NA |
| **Exceptions:** | NA |
| **Includes:** | NA |
| **Special Requirements:** | NA |
| **Assumptions:** | NA |
| **Notes and Issues:** | NA |

### 3.2.1.5 View Map

| | |
|---|---|
| **Use Case ID:** | UC-005 |
| **Use Case Name:** | View Map |
| **Actors:** | Driver |
| **Description:** | The user can view his location on Google map. |
| **Trigger:** | User taps on the map icon on the main interface screen. |
| **Preconditions:** | 1. Driver has mounted the smartphone on the windshield of the car horizontally.<br>2. Application is running.<br>3. GPS of the smartphone is enabled. |
| **Post conditions:** | 1. System displays location of driver’s car on Google Map. |
| **Normal Flow:** | 1. User taps on the map icon given on the main interface at the bottom left corner.<br>2. System checks if GPS of the smartphone is enabled.<br>3. System captures smartphones location coordinates via GPS.<br>4. System displays location of driver on Google Map using location coordinates. |
| **Alternative Flows:** | NA |
| **Exceptions:** | 3a. In step 3 of the normal flow, if the smartphone’s GPS is not enabled.<br>1. System fails to acquire location coordinates.<br>2. System notifies the user with a toast “GPS not enabled”.<br>3. Use case terminates |
| **Includes:** | NA |
| **Special Requirements:** | Smartphones GPS should be enabled. |
| **Assumptions:** | NA |
| **Notes and Issues:** | NA |

### 3.2.1.6 Specify Contacts

| | |
|---|---|
| **Use Case ID:** | UC-006 |
| **Use Case Name:** | Specify Contacts |
| **Actors:** | Driver |
| **Description:** | The user can specify people who can ping the application to acquire location coordinates of the smartphone. |
| **Trigger:** | User taps on the settings icon on the main interface. |
| **Preconditions:** | 1. Application is running. |
| **Post conditions:** | 1. Contact is specified successfully. |
| **Normal Flow:** | 1. User taps on the settings icon given on the main interface screen at the top left corner.<br>2. System displays settings menu.<br>3. User taps on the specify contacts option given in settings menu list.<br>4. System display Specify contact activity.<br>5. User taps on “Add Contact” option<br>6. System displays all the contacts present in the phone.<br>7. User selects the contact.<br>8. System adds the contact in shared preference of System. |
| **Alternative Flows:** | NA |
| **Exceptions:** | 5a. In step 5 of the normal flow, if the user’s smartphone has no contacts saved in it<br>1. System notifies the user via toast “No contacts saved in the smartphone”.<br>2. Use case terminates. |
| **Includes:** | NA |
| **Special Requirements:** | NA |
| **Assumptions:** | NA |
| **Notes and Issues:** | NA |

### 3.2.1.7 Calculate Distance

| | |
|---|---|
| **Use Case ID:** | UC-007 |
| **Use Case Name:** | Calculate Distance [Included Use Case] |
| **Actors:** | Driver |
| **Description:** | After the system has detected the vehicle ahead, the system calculates detected vehicle's distance from the driver's car. |
| **Trigger:** | System detects vehicle ahead. |
| **Preconditions:** | 1. Driver has mounted the smartphone on the windshield of the car horizontally.<br>2. Application is running. |
| **Post conditions:** | 1. System calculates and displays the distance of the detected vehicle ahead. |
| **Normal Flow:** | 1. Each time system detects vehicle, system will calculate its distance from driver's car by using calibration method.<br>2. System will display this calculated distance on the main interface above the tracked car using graphics. |
| **Alternative Flows:** | NA |
| **Exceptions:** | 1a. In step 1 of the normal flow, if the detected object is not a vehicle.<br>1. System will not calculate the detected object's distance.<br>2. Use case terminates. |
| **Includes:** | NA |
| **Special Requirements:** | NA |
| **Assumptions:** | NA |
| **Notes and Issues:** | NA |

### 3.2.1.8 Generate Warning

| | |
|---|---|
| **Use Case ID:** | UC-009 |
| **Use Case Name:** | Warning Generation[Extended Usecase] |
| **Actors:** | Driver |
| **Description:** | System shall generate audio and visual warnings |
| **Trigger:** | Vehicle ahead reaches a critical distance. |
| **Preconditions:** | 1. Driver has mounted the smartphone on the windshield of the car horizontally.<br>2. Application is running. |
| **Post conditions:** | 1. Driver will be alerted through audio and visual warnings. |
| **Normal Flow:** | 1. The detected vehicle ahead reaches critical distance.<br>2. System generates audio warning by playing different alarm sounds.<br>3. System generates visual warnings by showing collision warning signs on the main interface. |
| **Alternative Flows:** | NA |
| **Exceptions:** | 2a. In step 2 of the normal flow. If the system is in silent mode.<br>1. System will not generate audio warnings.<br>Use case resumes on step 3. |
| **Includes:** | Calculate Distance |
| **Special Requirements:** | NA |
| **Assumptions:** | NA |
| **Notes and Issues:** | NA |

### 3.2.2 Remote Application:

#### *3.2.2.1 Locate Driver*

| | |
|---|---|
| **Use Case ID:** | UC-010 |
| **Use Case Name:** | Locate Driver |
| **Actors:** | Remote Application User |
| **Description:** | The user can inquire location of driver |
| **Trigger:** | User taps on the locate icon present under contact name in remote application. |
| **Preconditions:** | 1. Application is running. |
| **Post conditions:** | 1. Application receives location coordinates and current screen of driver. |
| **Normal Flow:** | 1. System displays specified contacts on the main interface.<br>2. User taps on Locate icon present next to contact name.<br>3. Application pings driver phone for location. |
| **Alternative Flows:** | NA |
| **Exceptions:** | |
| **Includes:** | Display Driver Location |
| **Special Requirements:** | NA |
| **Assumptions:** | NA |
| **Notes and Issues:** | NA |

### *3.2.2.2 Display Driver Location*

| | |
|---|---|
| **Use Case ID:** | UC-011 |
| **Use Case Name:** | Display Driver Location (Included Use Case) |
| **Actors:** | Remote Application User |
| **Description:** | The user can view location and current screen of driver's smartphone. |
| **Trigger:** | System receives SMS with location Co-ordinates. |
| **Preconditions:** | 1. User has already requested driver location.<br>2. User receives a SMS.<br>3. GPS of the smartphone is enabled. |
| **Post conditions:** | 1. System displays received image and location of driver on Google map. |
| **Normal Flow:** | 1. System receives SMS from Third-Eye application.<br>2. System performs check if contact is already specified for location request.<br>3. System displays a toast "Location Received"<br>4. System displays received image of driver's smartphones current screen.<br>5. System uses received location coordinates in SMS to displays current location of driver on Google Maps. |
| **Alternative Flows:** | 2a. If at step 2. The contact is not specified.<br>1. System displays a toast "Contact not Specified" |
| **Exceptions:** | NA |
| **Includes:** | NA |
| **Special Requirements:** | GPS of the smartphone running remote application should be enabled. |
| **Assumptions:** | User has already inquired for driver's location. |
| **Notes and Issues:** | NA |

## 3.3 Functional Requirements

| ID | Description |
|---|---|
| Req-001 | The system shall display camera view on the main interface from smartphones rear using “Camera” available class in Android. |
| Req-002 | The system shall capture live video from smartphones rear camera frame by frame using “Camera” class in Android Studio. |
| Req-003 | The system shall detect vehicle in each video frame using edge detection. e.g. canny edge detection or Mean shift. |
| Req-004 | The system shall track detected vehicle ahead in multiple consecutive frames from live video feed using camshaft algorithm. |
| Req-005 | The system shall detect lane in each video frame using edge and line detection e.g. canny edge detection or Hough transform. |
| Req-006 | The system shall track detected lane of driver’s in multiple consecutive frames from live video feed using Hough transform and line detection algorithms. |
| Req-007 | The system shall calculate the distance, of the detected vehicle ahead, from the smartphone’s rear camera using triangulation method. |
| Req-008 | The system shall display the calculated distance on the main interface above the detected vehicle using graphics. |
| Req-009 | The system shall generate (refer to Req-025, Req-026) audio and visual collision warnings when vehicle detected ahead reaches a critical distance. |
| Req-010 | The system shall generate (refer to Req-025, Req-026) audio and visual warnings saying “Lane Changing” when driver’s vehicle changes its lane. |
| Req-011 | The system shall display settings button at top left corner on the main interface screen. |
| Req-012 | The user shall tap on settings icon given at top left corner on the main interface screen. |
| Req-013 | The system shall display settings menu containing (track vehicle, track lane, critical distance, specify contacts) as a list of options. |
| Req-014 | The system shall display toggle switch “Track Vehicle” as an option in the settings menu list. |
| Req-015 | The user shall enable or disable vehicle tracking by sliding the toggle switch. |
| Req-016 | The system shall display toggle switch “Track Lane” as an option in the settings menu list. |
| Req-017 | The user shall enable or disable lane tracking by sliding the toggle switch. |
| Req-018 | The system shall display a clickable link “Specify Contacts” as an option in the settings menu list. |
| Req-019 | The user shall tap on the specify contacts link given in settings menu list. |
| Req-020 | The system shall display a list of contacts stored in user’s smartphone. |
| Req-021 | The user shall select the desired contact by tapping on it. |
| Req-022 | The system shall display all the specified contacts. |
| Req-023 | The system shall display different confirmation message s, that would include (‘Contacts specified successfully’, ‘No contacts saved in the smartphone’, ‘Image Captured Successfully’), using a toast. |
| Req-024 | The system shall display a camera icon on the main interface. |

| | |
|---|---|
| Req-025 | The user shall tap on the camera icon given on the main interface at the middle left side. |
| Req-026 | The system shall capture image of main interface screen. |
| Req-027 | The system shall store the captured image in smartphone image gallery. |
| Req-028 | The system shall show map icon on main interface at bottom left corner. |
| Req-029 | The user shall tap on the map icon given on the main interface at the bottom left corner |
| Req-030 | The system shall capture smartphone’s location coordinates via GPS. |
| Req-031 | The system shall display current location of driver on Google map. |
| Req-032 | The system shall notify user with toast “GPS not Available”. |
| Req-033 | The system shall generate audio warning by playing hazard tone. |
| Req-034 | The system shall display hazard sign on main interface of the screen.. |
| Req-035 | The system shall send GPS coordinates via SMS |
| Req-036 | The system shall upload current frame on Server. |
| Req-037 | The system shall check shared preference for contact. |
| Req-038 | The system shall display list of contacts for location inquiry on the main interface of the application. |
| Req-039 | The user shall tap on the contact name given in the contacts list on the main interface. |
| Req-040 | The system shall display contact name on specify contact activity. |
| Req-041 | The system shall display a button “Locate” along with a globe icon next to the contact name. |
| Req-042 | The user shall tap “Locate” button with a globe icon next to contact name. |
| Req-043 | The system shall request driver for his location and current screen of Third Eye application on driver’s smartphone via GSM service e.g. SMS. |
| Req-044 | The user wants to view current screen Third Eye application and location of driver on Google map. |
| Req-045 | The system shall display current screen of driver and display it on the upper half of screen under the name of driver. |
| Req-046 | The system shall use received coordinates in SMS to display current location of driver on lower half of the screen using Google Map. |

### 3.3.1 Display Camera View

| Identifier | Req-001 |
|---|---|
| Title | Display Camera View |
| Requirement | The system shall display camera view on the main interface from smartphones rear camera using “Camera” class available in Android. |
| Source | Supervisor |
| Rationale | System captures video for tracking vehicle and lane. |
| Restrictions and Risk | NA |
| Dependencies | NA |

| | |
|---|---|
| Priority | High |

### 3.3.2 Capture Video

| Identifier | Req-002 |
|---|---|
| Title | Capture Video |
| Requirement | The system shall capture live video from smartphones rear camera frame by frame using “Camera” class in Android Studio. |
| Source | Supervisor |
| Rationale | System captures video for tracking vehicle and lane. |
| Restrictions and Risk | NA |
| Dependencies | Req-001 |
| Priority | High |

### 3.3.3 Detect Vehicle

| Identifier | Req-003 |
|---|---|
| Title | Detect vehicle |
| Requirement | The system shall detect vehicle in each video frame using edge detection algorithm. |
| Source | Supervisor |
| Rationale | System detects vehicle ahead for tracking. |
| Restrictions and Risk | NA |
| Dependencies | Req-002. |
| Priority | High |

### 3.3.4 Track Vehicle

| Identifier | Req-004 |
|---|---|
| Title | Track vehicle |
| Requirement | The system shall track detected vehicle ahead in multiple consecutive frames from live video feed using image tracking algorithm. |
| Source | Supervisor |
| Rationale | System tracks vehicle ahead to assist driver. |
| Restrictions and Risk | NA |
| Dependencies | Req-002, Req-003. |
| Priority | High |

### 3.3.5 Detect Lane

| Identifier | Req-005 |
|---|---|
| Title | Detect Lane |
| Requirement | The system shall detect lane in each video frame using edge and line detection algorithms e.g. Hough transform |
| Source | Supervisor |

| | |
|---|---|
| Rationale | System detects lane of driver's car. |
| Restrictions and Risk | NA |
| Dependencies | Req-002. |
| Priority | High |

### 3.3.6 Track Lane

| Identifier | Req-006 |
|---|---|
| Title | Track lane |
| Requirement | The system shall track detected lane of driver's in multiple consecutive frames from live video feed using Hough transform and line detection algorithms. |
| Source | Supervisor |
| Rationale | System tracks road lane to assist driver. |
| Restrictions and Risk | NA |
| Dependencies | Req-002, Req-005. |
| Priority | High |

### 3.3.7 Calculate Distance

| Identifier | Req-007 |
|---|---|
| Title | Calculate Distance |
| Requirement | The system shall calculate the distance, of the detected vehicle ahead, from the smartphone's rear camera. |
| Source | Supervisor |
| Rationale | The application tracks vehicle ahead for collision warnings. |
| Restrictions and Risk | NA |
| Dependencies | Req-002, Req-003. |
| Priority | High |

### 3.3.8 Display Distance

| Identifier | Req-008 |
|---|---|
| Title | Display Distance |
| Requirement | The system shall display the calculated distance on the main interface above the detected vehicle using graphics in Android. |
| Source | Supervisor |
| Rationale | The application tracks vehicle ahead for collision warnings. |
| Restrictions and Risk | NA |
| Dependencies | Req-007 |
| Priority | High |

### 3.3.9 Critical Distance Warning

| Identifier | Req-009 |
|---|---|
| Title | Critical Distance Warning |
| Requirement | The system shall generate (refer to Req-025, Req-026) audio and visual collision warnings when vehicle detected ahead reaches a critical distance. |
| Source | Supervisor |
| Rationale | The application alerts the driver at critical distance. |
| Restrictions and Risk | NA |
| Dependencies | Req-035 |
| Priority | High |

### 3.3.10 Lane Change Warning

| Identifier | Req-0010 |
|---|---|
| Title | Lane Change Warning |
| Requirement | The system shall generate (refer to Req-025, Req-026) audio and visual warnings saying “Lane Changing” when driver’s vehicle changes its lane. |
| Source | Supervisor |
| Rationale | The application alerts the driver when vehicle’s lane changes. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | High |

### 3.3.11 Display Settings Icon

| Identifier | Req-011 |
|---|---|
| Title | Display Settings Icon |
| Requirement | The system shall display settings button at top left corner on the main interface screen. |
| Source | Supervisor |
| Rationale | The user wants to open the settings menu. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | High |

### 3.3.12 Open Settings

| Identifier | Req-012 |
|---|---|
| Title | Open Settings |
| Requirement | The user shall tap on settings icon given at top left corner on the main interface screen. |
| Source | Supervisor |
| Rationale | The user wants to open the settings menu. |

| | |
|---|---|
| Restrictions and Risk | NA |
| Dependencies | Req-011 |
| Priority | High |

### 3.3.13 Display Settings Menu

| Identifier | Req-013 |
|---|---|
| Title | Display Settings Menu |
| Requirement | The system shall display settings menu containing (track vehicle, track lane, specify contacts) as a list of options. |
| Source | Supervisor |
| Rationale | The user wants to configure the application. |
| Restrictions and Risk | NA |
| Dependencies | Req-012 |
| Priority | High |

### 3.3.14 Track Vehicle Toggle Switch

| Identifier | Req-014 |
|---|---|
| Title | Track vehicle toggle switch |
| Requirement | The system shall display toggle switch “Track Vehicle” as an option in the settings menu list. |
| Source | Supervisor |
| Rationale | The user wants to enable or disable track vehicle feature of the application. |
| Restrictions and Risk | NA |
| Dependencies | Req-013 |
| Priority | Medium |

### 3.3.15 Slide Track Vehicle Toggle Switch

| Identifier | Req-015 |
|---|---|
| Title | Slide  Track Vehicle Toggle Switch |
| Requirement | The user shall enable or disable vehicle tracking by sliding the toggle switch. |
| Source | Supervisor |
| Rationale | The user wants to enable or disable track vehicle feature of the application. |
| Restrictions and Risk | NA |
| Dependencies | Req-014 |
| Priority | Medium |

### 3.3.16 Track Lane Toggle Switch

| Identifier | Req-016 |
|---|---|
| Title | Track lane toggle switch |
| Requirement | The system shall display toggle switch “Track Lane” as an |

|  |  |
|---|---|
|  | option in the settings menu list. |
| Source | Supervisor |
| Rationale | The user wants to enable or disable track lane feature of the application. |
| Restrictions and Risk | NA |
| Dependencies | Req-013 |
| Priority | Medium |

### 3.3.17 Slide Track Lane Toggle Switch

| Identifier | Req-017 |
|---|---|
| Title | Slide Track Lane Toggle Switch |
| Requirement | The user shall enable or disable lane tracking by sliding the toggle switch. |
| Source | Supervisor |
| Rationale | The user wants to enable or disable lane tracking. |
| Restrictions and Risk | NA |
| Dependencies | Req-016 |
| Priority | Medium |

### 3.3.18 Specify Contacts Link

| Identifier | Req-018 |
|---|---|
| Title | Specify Contacts Link |
| Requirement | The system shall display a clickable link “Specify Contacts” as an option in the settings menu list. |
| Source | Supervisor |
| Rationale | The user wants to specify contacts that can ping the application to acquire location coordinate. |
| Restrictions and Risk | NA |
| Dependencies | Req-013 |
| Priority | Medium |

### 3.3.19 Tap Specify Contacts Link

| Identifier | Req-019 |
|---|---|
| Title | Tap Specify Contacts Link |
| Requirement | The user shall tap on the specify contacts link given in settings menu list. |
| Source | Supervisor |
| Rationale | The user wants to specify contacts that can ping the application to acquire location coordinate. |
| Restrictions and Risk | NA |
| Dependencies | Req-018 |
| Priority | Medium |

### 3.3.20 Display Contacts List

| Identifier | Req-020 |
|---|---|
| Title | Contacts list |
| Requirement | The system shall display a list of contacts stored in user’s smartphone. |
| Source | Supervisor |
| Rationale | The user wants to specify contacts that can ping the application to acquire location coordinate. |
| Restrictions and Risk | NA |
| Dependencies | Req-019 |
| Priority | Medium |

### 3.3.21 Select Contacts

| Identifier | Req-021 |
|---|---|
| Title | Select Contacts |
| Requirement | The user shall select the desired contact by tapping on it. |
| Source | Supervisor |
| Rationale | The user wants to specify contacts that can ping the application to acquire location coordinate. |
| Restrictions and Risk | NA |
| Dependencies | Req-020 |
| Priority | Medium |

### 3.3.22 Display Specified Contacts

| Identifier | Req-022 |
|---|---|
| Title | Display Done Button |
| Requirement | The system shall display all the specified contacts. |
| Source | Supervisor |
| Rationale | The user wants to save the selected contacts that can ping the application to acquire location coordinate. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | Low |

### 3.3.23 Confirmation Message

| Identifier | Req-023 |
|---|---|
| Title | Confirmation Message |
| Requirement | The system shall display different confirmation message s, that would include (‘Contacts specified successfully’, ‘No contacts saved in the smartphone’, ‘Image Captured Successfully’), using a toast. |
| Source | Supervisor |

| | |
|---|---|
| Rationale | System displays different messages to assure user that requested action has been performed or not. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | Low |

### 3.3.24 Display Image Capture Button

| Identifier | Req-024 |
|---|---|
| Title | Display Image Capture Button |
| Requirement | The system shall display a camera icon on the main interface. |
| Source | Supervisor |
| Rationale | The application captures screenshot of main interface screen. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | Medium |

### 3.3.25 Tap Image Capture Button

| Identifier | Req-025 |
|---|---|
| Title | Tap Image Capture Button |
| Requirement | The user shall tap on the camera icon given on the main interface at the middle left side. |
| Source | Supervisor |
| Rationale | The user wants to capture the image of current roadway. |
| Restrictions and Risk | NA |
| Dependencies | Req-024 |
| Priority | Medium |

### 3.3.26 Capture Image

| Identifier | Req-026 |
|---|---|
| Title | Capture Image |
| Requirement | The system shall capture image of main interface screen. |
| Source | Supervisor |
| Rationale | The application captures screenshot of main interface screen. |
| Restrictions and Risk | NA |
| Dependencies | Req-025 |
| Priority | Medium |

### 3.3.27 Save Captured Image

| Identifier | Req-027 |
|---|---|
| Title | Save Captured Image |

| | |
|---|---|
| Requirement | The system shall store the captured image in smartphone image gallery. |
| Source | Supervisor |
| Rationale | The application captures screenshot of main interface screen. |
| Restrictions and Risk | NA |
| Dependencies | Req-026 |
| Priority | Medium |

### 3.3.28 Display Map Icon

| Identifier | Req-028 |
|---|---|
| Title | Map Icon |
| Requirement | The system shall show map icon on main interface at bottom left corner. |
| Source | Supervisor |
| Rationale | The application shows current location of driver. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | Medium |

### 3.3.29 Tap Map Icon

| Identifier | Req-029 |
|---|---|
| Title | Tap Map Icon |
| Requirement | The user shall tap on the map icon given on the main interface at the bottom left corner |
| Source | Supervisor |
| Rationale | User wants to see his location on the Google Map. |
| Restrictions and Risk | NA |
| Dependencies | Req-028 |
| Priority | Medium |

### 3.3.30 Capture Location

| Identifier | Req-030 |
|---|---|
| Title | Capture Location |
| Requirement | The system shall capture smartphone’s location coordinates via GPS. |
| Source | Supervisor |
| Rationale | The application shows current location of driver. |
| Restrictions and Risk | GPS of the smartphone should be enabled. |
| Dependencies | Req-029 |
| Priority | Medium |

### 3.3.31 Display Location

| Identifier | Req-031 |
|---|---|
| Title | Display Location |
| Requirement | The system shall display current location of driver on Google map. |
| Source | Supervisor |
| Rationale | Driver wants to see his current location on Map. |
| Restrictions and Risk | NA |
| Dependencies | Req-030 |
| Priority | Medium |

### 3.3.32 GPS not Enabled Notification

| Identifier | Req-032 |
|---|---|
| Title | GPS not enabled. |
| Requirement | The system shall notify user about connectivity with toasts like “GPS not Available” and “No Network Coverage”. |
| Source | Supervisor |
| Rationale | The application failed to capture GPS co-ordinates. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | Low |

### 3.3.33 Audio Warning Generation

| Identifier | Req-033 |
|---|---|
| Title | Audio Warning |
| Requirement | The system shall generate audio warning by playing hazard tone. |
| Source | Supervisor |
| Rationale | The detected vehicle ahead reaches critical distance. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | High |

### 3.3.34 Visual Warning Generation

| Identifier | Req-034 |
|---|---|
| Title | Visual Warning |
| Requirement | The system shall display hazard sign on main interface of the screen. |
| Source | Supervisor |
| Rationale | The detected vehicle ahead reaches critical distance. |
| Restrictions and Risk | NA |
| Dependencies | NA |

| | |
|---|---|
| Priority | High |

### 3.3.35 Send GPS Coordinates

| Identifier | Req-035 |
|---|---|
| Title | Send GPS Coordinates. |
| Requirement | The system shall send longitude and latitude values via SMS. |
| Source | Supervisor |
| Rationale | The remote app want to see driver location. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | Medium |

### 3.3.36 Upload Frame on Server

| Identifier | Req-036 |
|---|---|
| Title | Upload frame on server. |
| Requirement | The system shall upload frame on server. |
| Source | Supervisor |
| Rationale | The remote app want to see driver location. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | Medium |

### 3.3.37 Check Shared Preference

| Identifier | Req-037 |
|---|---|
| Title | Check shared preference for contact |
| Requirement | The system shall check shared preference to check if contact exists for location inquiry. |
| Source | Supervisor |
| Rationale | The remote user wants to see driver location. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | Low |

### 3.3.38 Contact List

| Identifier | Req-038 |
|---|---|
| Title | Contact List |
| Requirement | The system shall display list of contacts who have granted permission for location inquiry on the main interface of the application. |
| Source | Supervisor |
| Rationale | The user wants to locate driver. |
| Restrictions and Risk | NA |

| | |
|---|---|
| Dependencies | NA |
| Priority | High |

### 3.3.39 Open Contact Details

| Identifier | Req-039 |
|---|---|
| Title | Open Contact Detail |
| Requirement | The user shall tap on the contact name given in the contacts list on the main interface. |
| Source | Supervisor |
| Rationale | The user wants to locate driver. |
| Restrictions and Risk | NA |
| Dependencies | Req-038 |
| Priority | High |

### 3.3.40 Display Contact Details

| Identifier | Req-040 |
|---|---|
| Title | Contact Details |
| Requirement | The system shall display contact name in specify contact activity. |
| Source | Supervisor |
| Rationale | The user wants to locate driver. |
| Restrictions and Risk | NA |
| Dependencies | Req-039 |
| Priority | Medium |

### 3.3.41 Locate Driver

| Identifier | Req-041 |
|---|---|
| Title | Locate Driver |
| Requirement | The system shall display a button “Locate” along with a globe icon next to contact name. |
| Source | Supervisor |
| Rationale | The user wants to locate driver. |
| Restrictions and Risk | NA |
| Dependencies | Req-040 |
| Priority | High |

### 3.3.42 Tap Locate Driver

| Identifier | Req-042 |
|---|---|
| Title | Tap Locate Driver |
| Requirement | The user shall tap “Locate” button with a globe icon next to contact name. |
| Source | Supervisor |
| Rationale | The user wants to locate driver. |

| | |
|---|---|
| Restrictions and Risk | NA |
| Dependencies | Req-041 |
| Priority | High |

### 3.3.43 Ping Driver

| Identifier | Req-043 |
|---|---|
| Title | Ping Driver |
| Requirement | The system shall request driver for his location and current screen of Third Eye application on driver’s smartphone via GSM service e.g. SMS. |
| Source | Supervisor |
| Rationale | The user wants to inquire location of driver. |
| Restrictions and Risk | NA |
| Dependencies | Req-042 |
| Priority | Medium |

### 3.3.44 Open SMS

| Identifier | Req-044 |
|---|---|
| Title | Open SMS |
| Requirement | The system shall open SMS received from driver’s smartphone using remote application. |
| Source | Supervisor |
| Rationale | The user wants to view current screen Third Eye application and location of driver on Google map. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | Medium |

### 3.3.45 Display Image

| Identifier | Req-045 |
|---|---|
| Title | Display Image |
| Requirement | The system shall receive current screen of Third Eye app on driver’s smartphone and shall display it on the upper half of screen. |
| Source | Supervisor |
| Rationale | The user wants to view the received image. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | Medium |

### 3.3.46 Display Location on Map

| Identifier | Req-046 |
|---|---|
| Title | Location on Map |

| | |
|---|---|
| Requirement | The system shall use received coordinates in SMS to display current location of driver on lower half of the screen using Google Map. |
| Source | Supervisor |
| Rationale | The user wants to view location of driver. |
| Restrictions and Risk | NA |
| Dependencies | NA |
| Priority | Medium |

## 3.4 Non-Functional Requirements

### 3.4.1 Usability

#### *3.4.1.1 Usability Requirements*

| ID | Description |
|---|---|
| UR-1 | Since a driver assistance system, the GUI will be simple following the principle of Human Computer Interaction (HCI) e.g. Show system statuses, Familiar metaphors and language, Error prevention, Recognition over recall, Aesthetics and minimalist design, Flexibility and efficiency. |
| UR-2 | It will take only 10 minutes for a power user to understand the functionality of the system and perform the basic tasks. The normal user may take 15 minutes to completely understand and use the system this will be tested through trainings and controlled experimentations. |

### 3.4.2 Reliability

#### *3.4.2.1 Reliability Requirements*

| ID | Description |
|---|---|
| R-1 | The system shall be maintained and evolved by providing regular updates. |
| R-2 | The maximum time between failures will be approximately 25 to 30 seconds that’s the time it must take to restart an application after crashing. |
| R-3 | The system shall provide at least greater than 75% accurate results. |

### 3.4.3 Performance

#### *3.4.3.1 Performance Requirements*

| ID | Description |
|---|---|
| P-1 | The response time of the application for each task shall be minimum 5 seconds, and it shall not exceed 6 seconds. |
| P-2 | The application shall find location of the smartphone in 3-5 seconds upon request. |

### 3.4.4 Supportability

#### *3.4.4.1 Supportability Requirements*

| ID | Description |
|---|---|
| SU-1 | Some of coding standard or class libraries that will be used in developing this system are :<br>OpenCV |
| SU-2 | Following software will be used to build this system:<br>Android Studio for Android Application development |
| SU-3 | The system shall be compatible with Android versions 4.4 and above. |
| SU-4 | The system shall be compatible with a smartphone of at least 1GB ram, Quad Core processor and 8mp (mega pixel) camera. |

# 4 Design and Architecture

## 4.1 System Architecture

The Architectural pattern that will be followed to implement Third Eye driver assistance system is MVC (Model-View-Controller). MVC is best for GUI applications in Android, as it decouples user interface and its representation from the actual functionality of the system. MVC enhances the understandability of the code. Even in the late stages of software development, new features can be added quite easily if MVC is adapted while development.

Following is the detailed description of how MVC will be adapted for the development of Third Eye driver assistance system.

In Android Studio, each Activity.xml (UI) represents a view. As any event is generated on the activity, it activates controller of that event which is present in the Activity.java file. Controller looks for methods assigned to each event in its respective Model class and passes argument. Using the model classes, controller performs the required action and updates the Activity.
Model is basically repository of all the classes and predefined functions. Model represents data of controllers. View can sometimes access the models but views cannot modify model, only controller can modify models. Controller is event handler and purely java based. Controller is activated by events on views. It uses methods from model classes processes events and displays processed data on View.

Following are the MVC components of the system:
**Models**

- Services Class
- Intents Class
- Android Studio Libraries
- Graphics Class
- OpenCV Library
- Camera Class
- MMSService Class
- GoogleMap Class
- Tracker Class
- SpecifyContact Class

**Views**

- Main Activity.
- Settings Activity.
- Specify Contacts Activity.
- App Start up Activity.
- Google Map Activity

**Controllers**

- Event handler class of each activity.xml file.

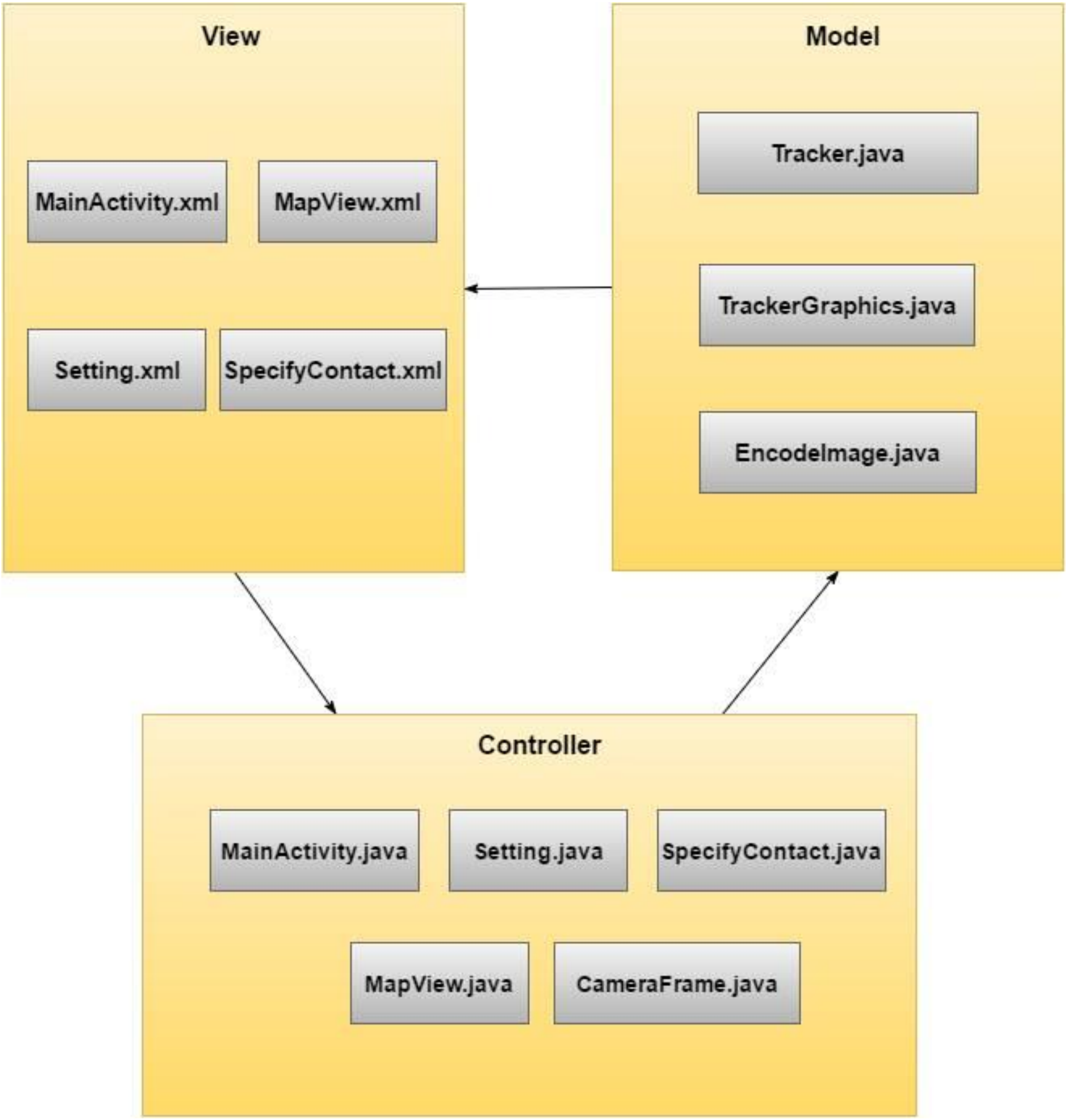


**Figure 3 MVC Architecture**

## 4.2 Data Representation

Third Eye driver assistance system is not data intensive; the only need is to store and fetch contact details and critical distance. Since most of the computations and processing shall be performed in real time and on real time data. So instead of using a database in the form of SQLite we will use shared preferences of Android operating system.

Shared preferences save the data in the internal storage of system. It allows you to retrieve and store data in key/value pair. For Instance, you just have to make object of shared preferences class, call its function "getSharedPrefrences()". This function will return the value stored against the key that is present in the shared preferences data storage. This method has two parameters key and mode. Key works as a pointer to the data stored in the storage and mode defines operation to be performed on the data present against the value key is pointing at.

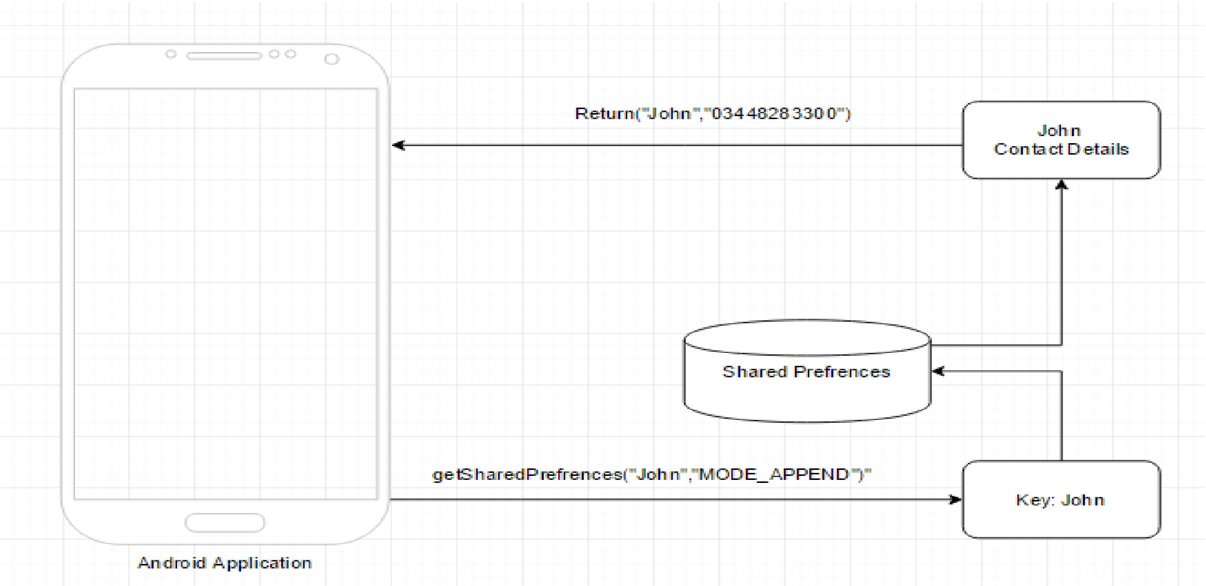

Return("John","034 4828 3300")
John
Contact Details
Shared Prefrences
getSharedPrefrences("John","MODE_APPEND")"
Key: John
Android Application

**Figure 4 Fetching Value from Shared Preferences**

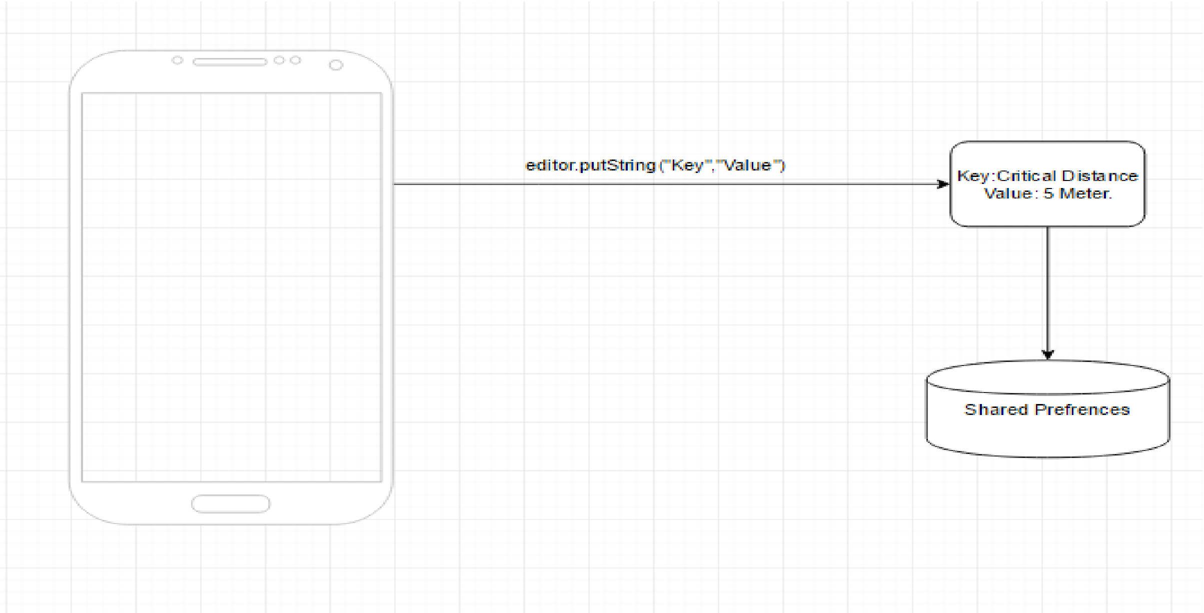


**Figure 5 Updating Value in Shared Preferences**

## 4.3 Process Flow/Representation

### 4.3.1 Main Activity:

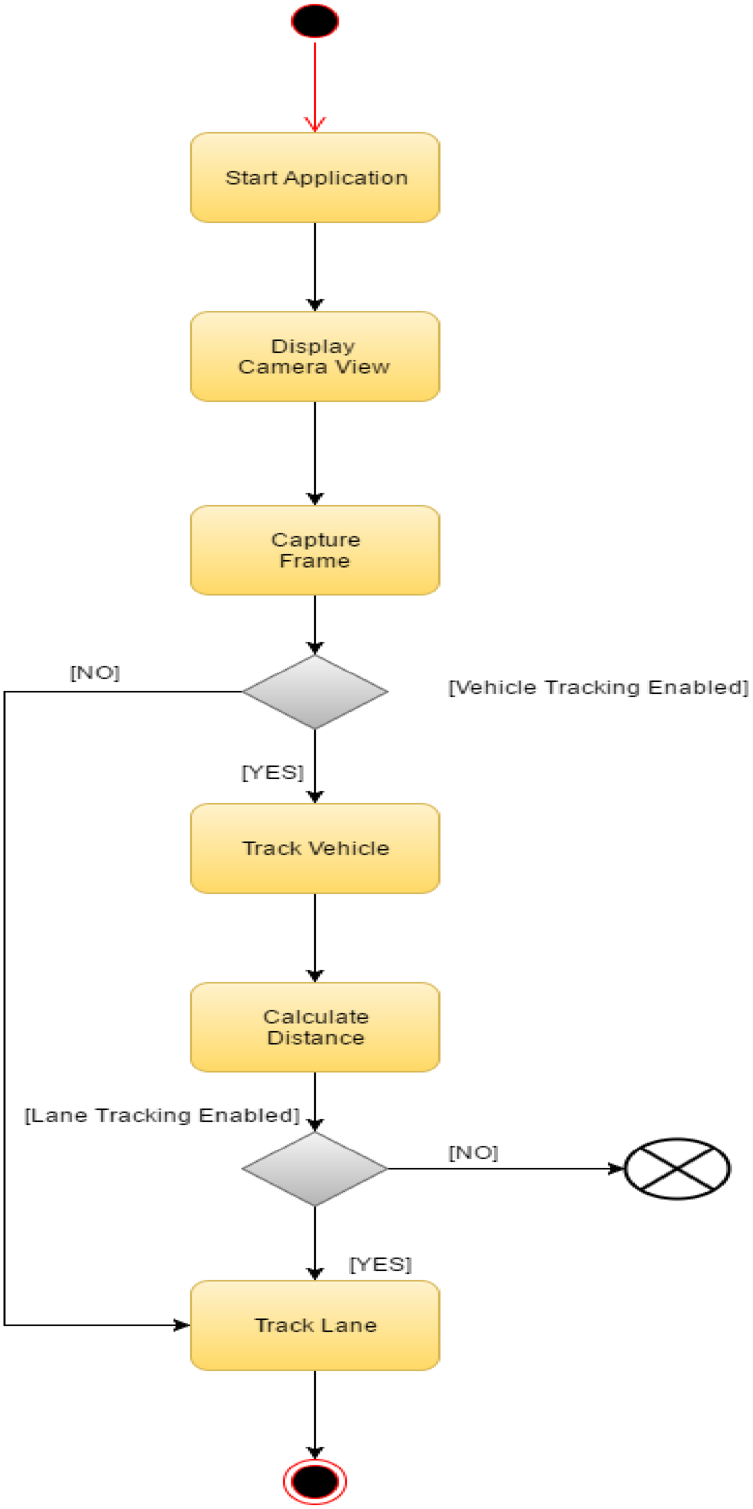


**Figure 6 Activity Main Flow**

### 4.3.2 Capture Screen:

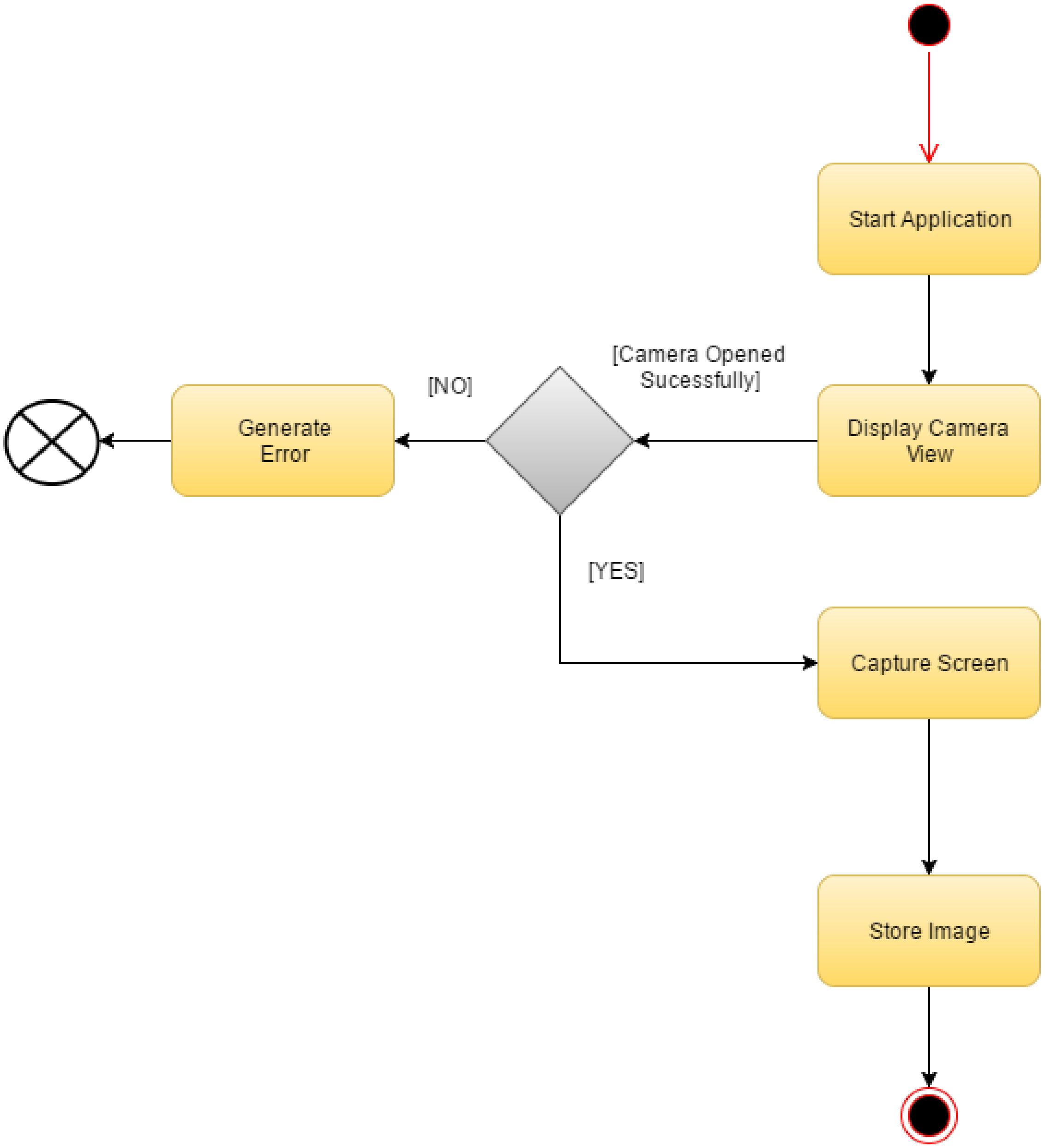


**Figure 7 Activity Capture Screen**

### 4.3.3 View Map:

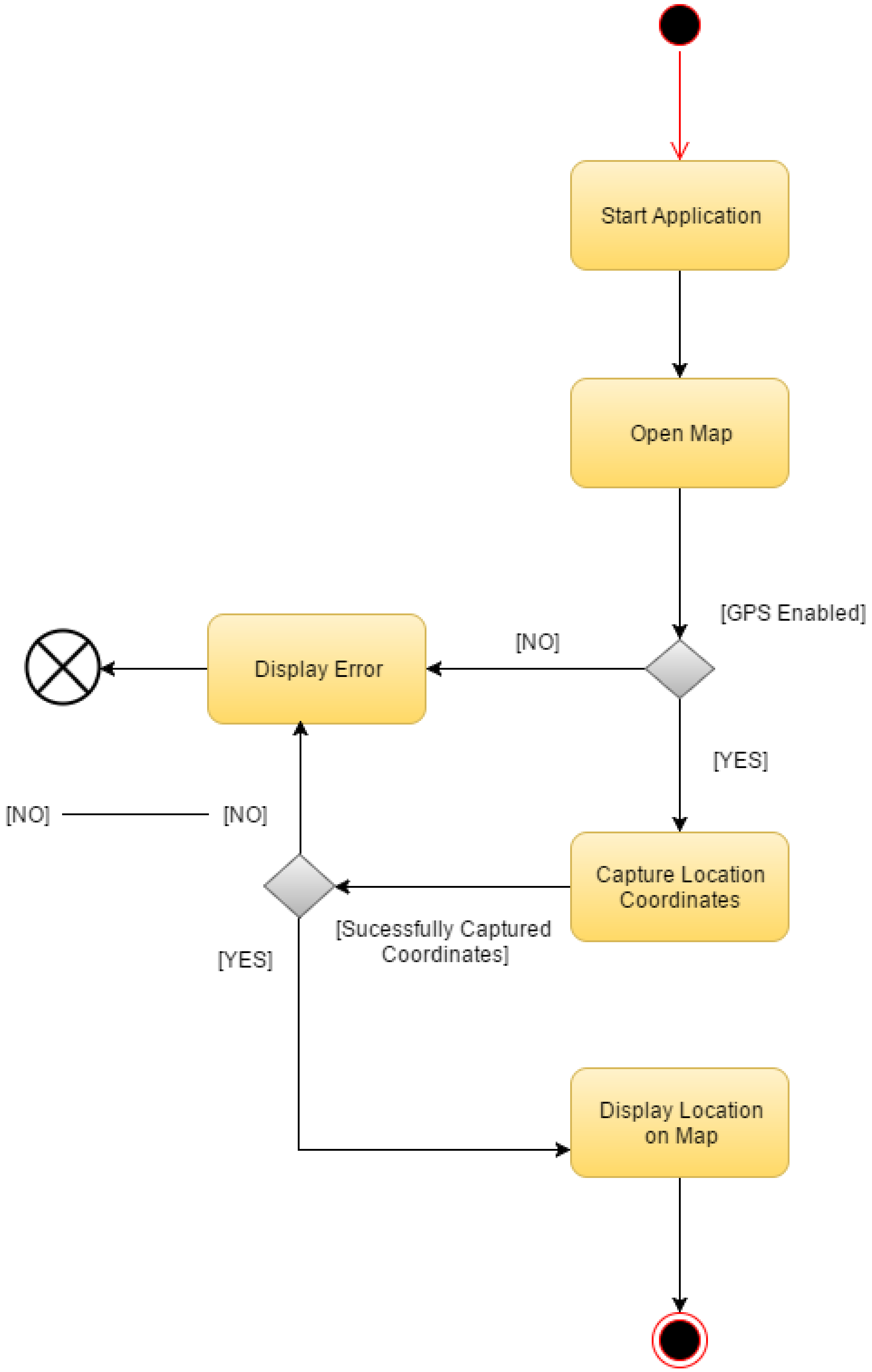


**Figure 8 Activity View Map**

### 4.3.4 Track Vehicle:

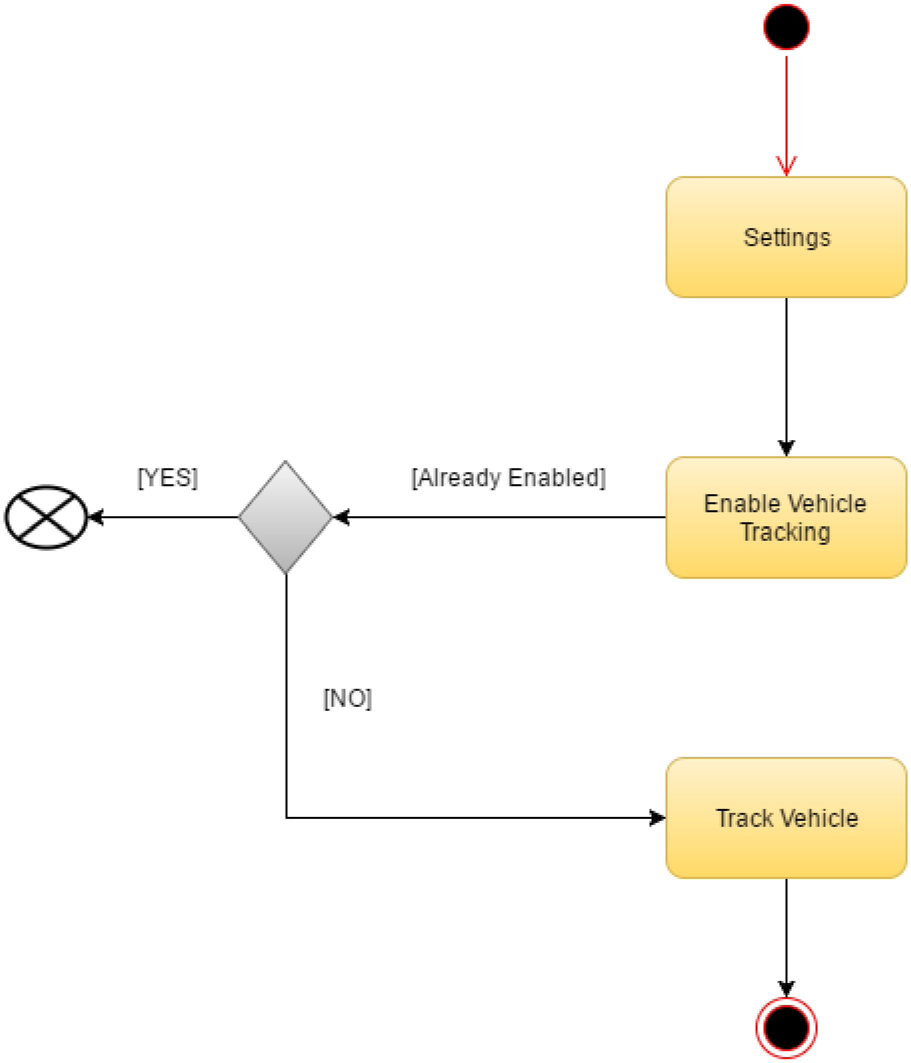


**Figure 9 Activity Enable Track Vehicle**

### 4.3.5 Track Lane:

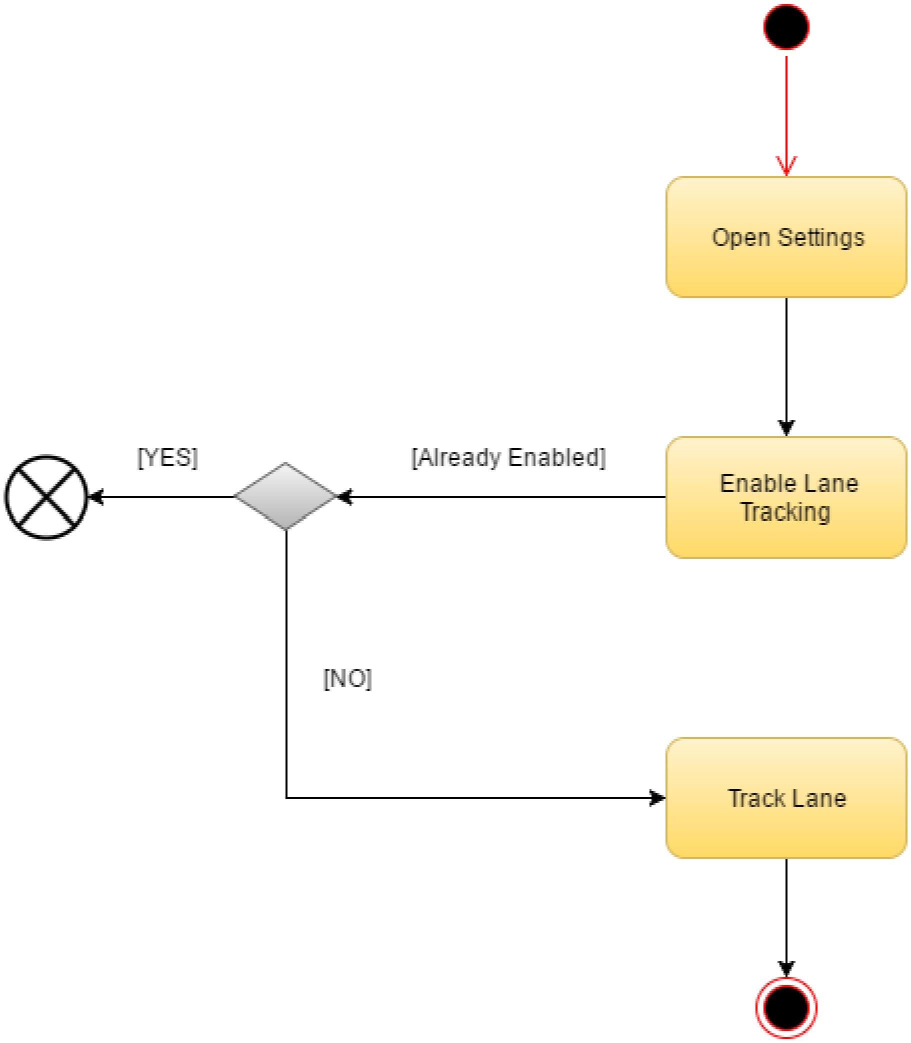


**Figure 10 Activity Track Lane**

### 4.3.6 Specify Contacts:

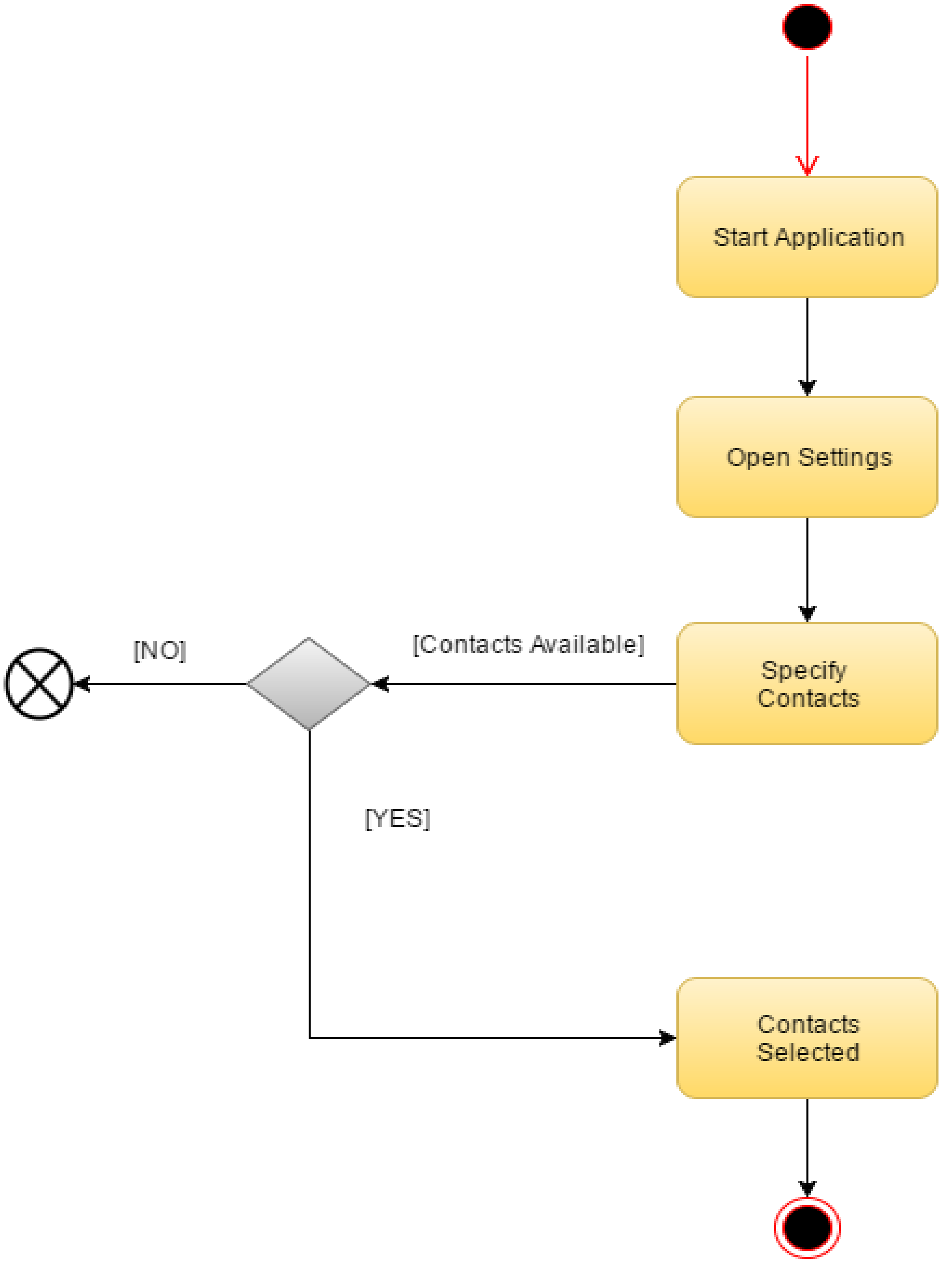


**Figure 11 Activity Specify Contacts**

### 4.3.7 Generate Critical Distance Warning:

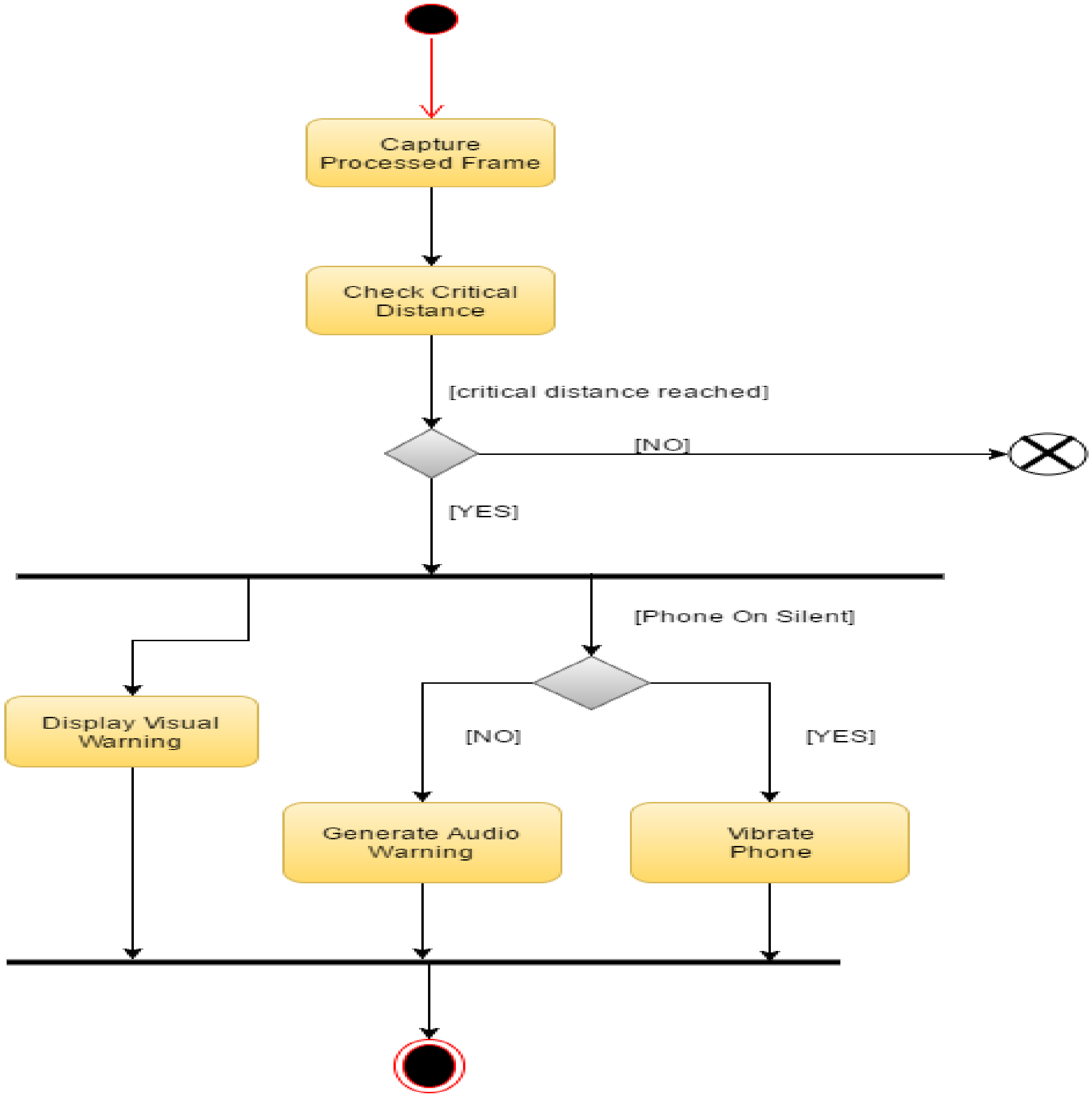


**Figure 12 Activity Generate Critical Distance Warning**

### 4.3.8 Generate Lane Change Warning:

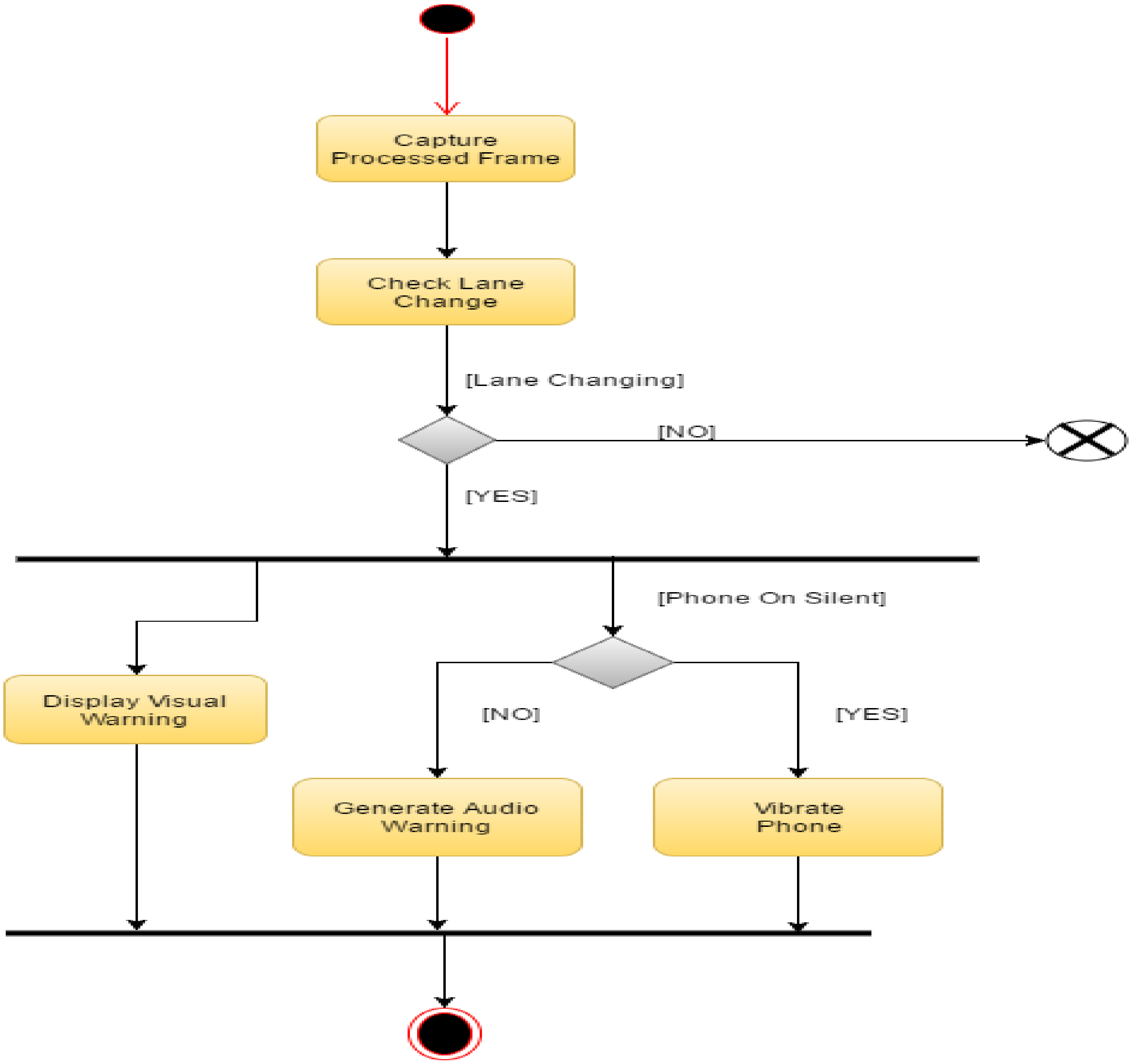


**Figure 13 Activity Generate Lane Warning**

### 4.3.9 Locate Driver:

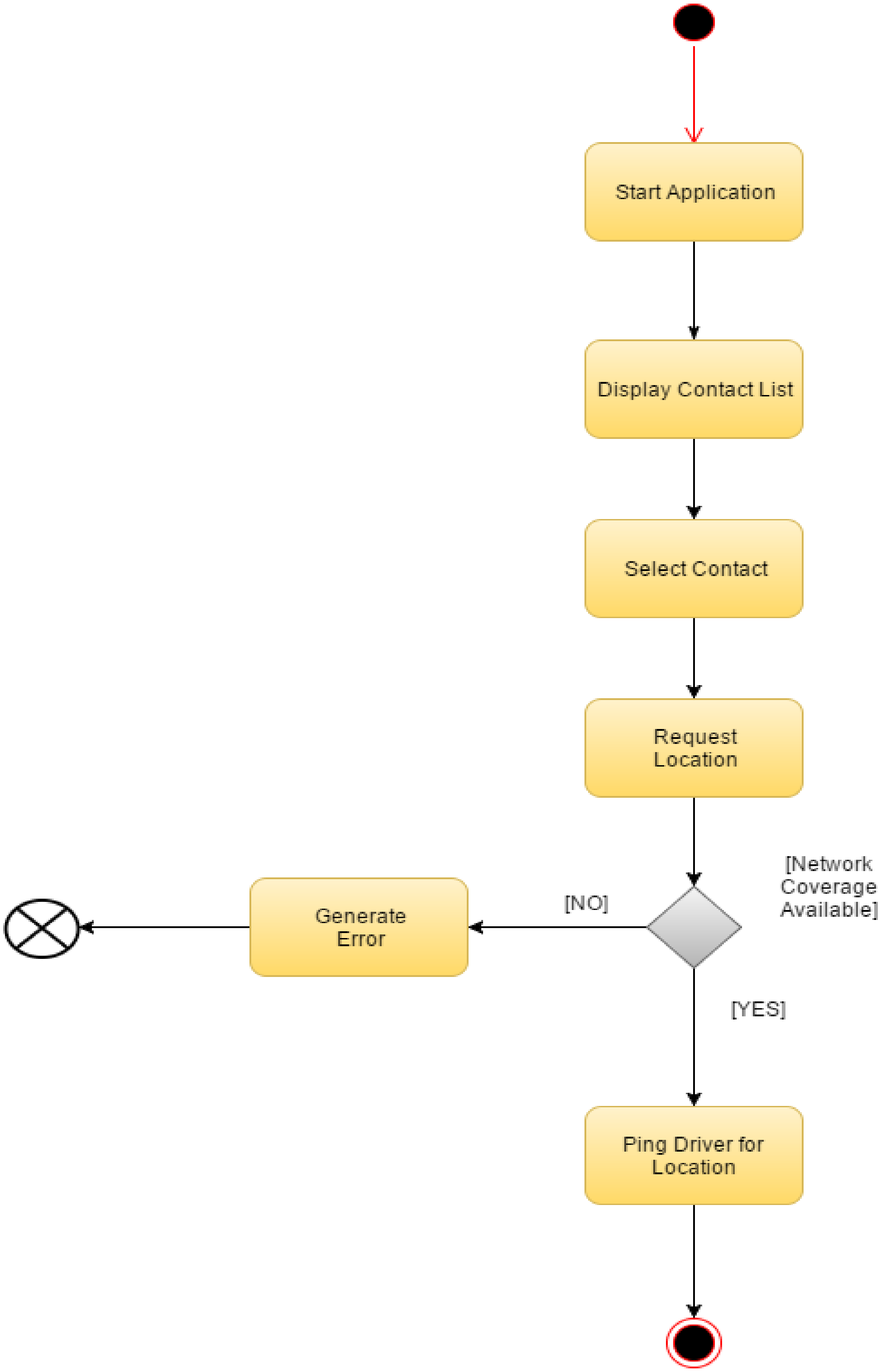


**Figure 14 Activity Locate Driver**

## 4.4 Design Models [along with descriptions]

### 4.4.1 Class Diagram

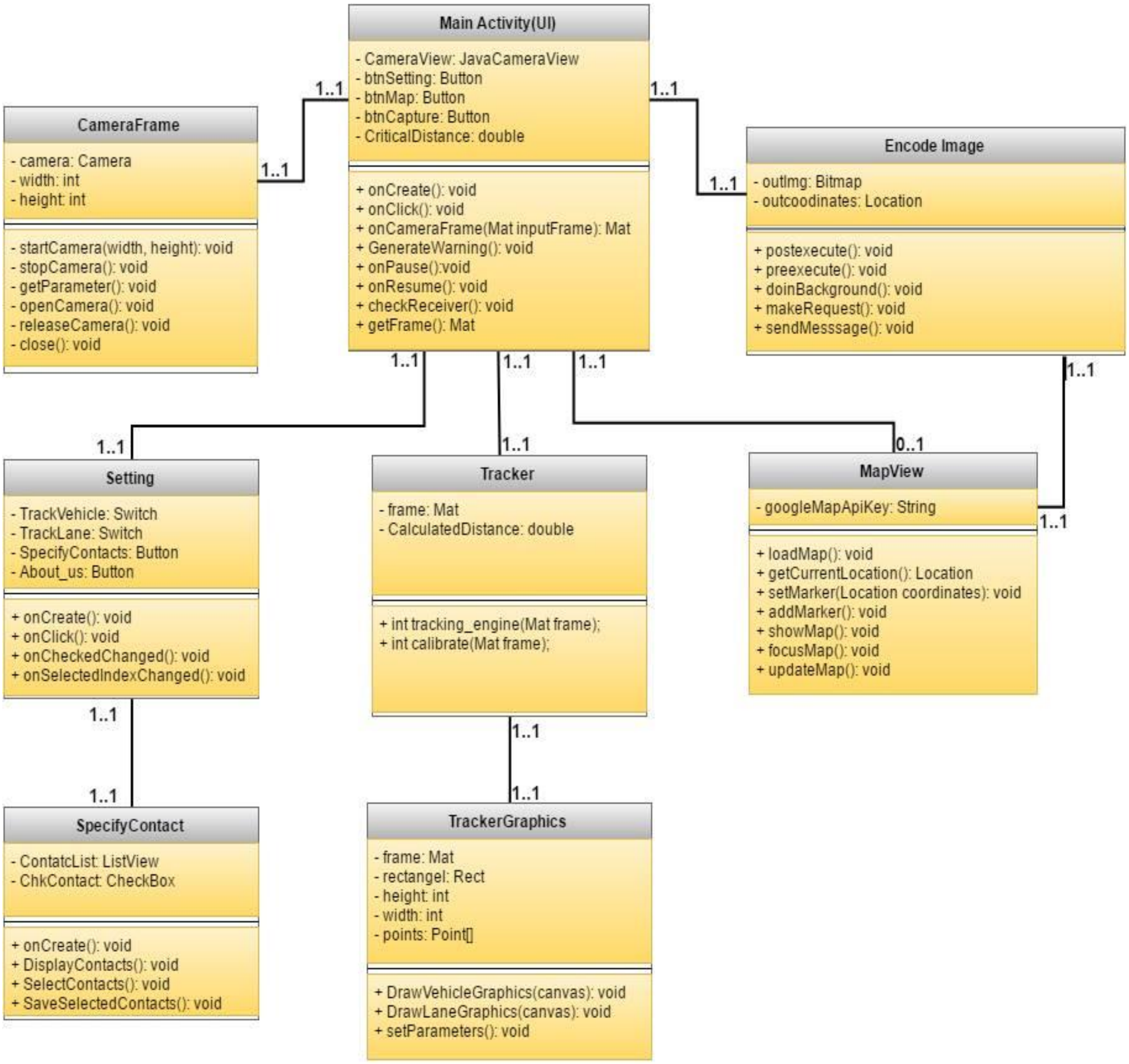


**Figure 15 Class Diagram Main Application**

| Class Name | Description |
|---|---|
| **Main Activity** | This is the main interface of system. This class will display CameraView on the screen. It will provide user an interface to interact with system. Tracked vehicle and lane will be also displayed on the screen using this class. |
| **Tracker** | This class will detect and track vehicle and lane in a frame. It also has functions that will calculate distance of tracked vehicle. |
| **MapView** | This class will be used to access Google Map. It will locate and mark location of driver on Map. |
| **CameraFrame** | This class will be used to access camera of the phone. It captures live video feed and sends it to main activity. |
| **Settings** | This class is used to display and change settings of the application |
| **SpecifyContact** | This Class will allow user to specify contacts who will have permission to acquire location of driver. |

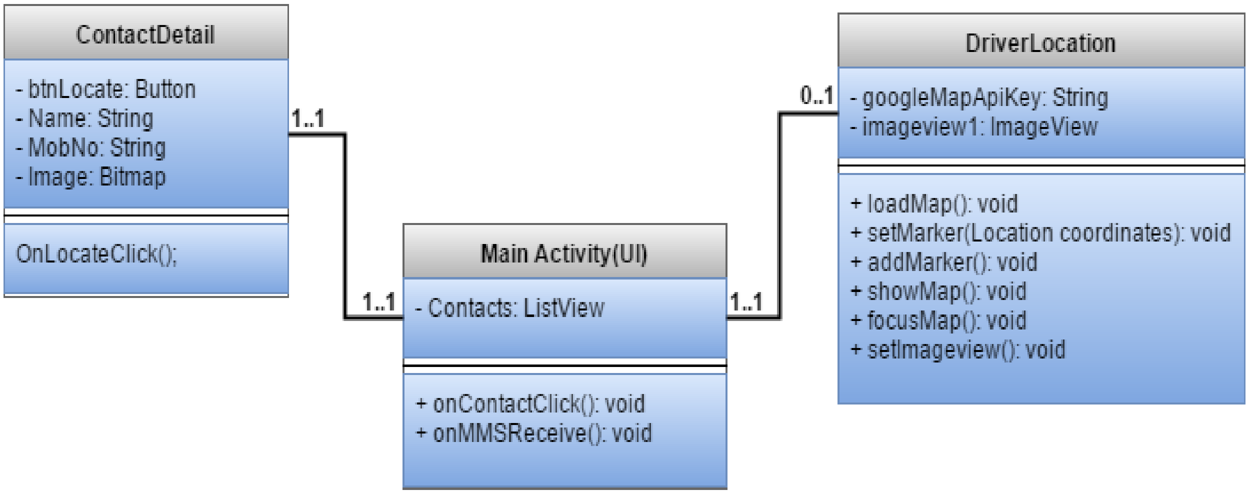


**Figure 16 Class Diagram Remote Application**

| 3.3 **Class Name** | 3.4 **Description** |
|---|---|
| **MainActivity** | Main activity class will display a list of driver's whose location can be inquired. |
| **ContactDetails** | It will display name of driver and his details along with a button to |

|  |  |
|---|---|
|  | initiate driver location inquiry. |
| **Driver Location** | This class will display the received location of the driver on Google Map. It will also display current screen of driver’s phone. |

### 4.4.2 Sequence Diagram

#### 4.4.2.1 Track Vehicle and Lane:

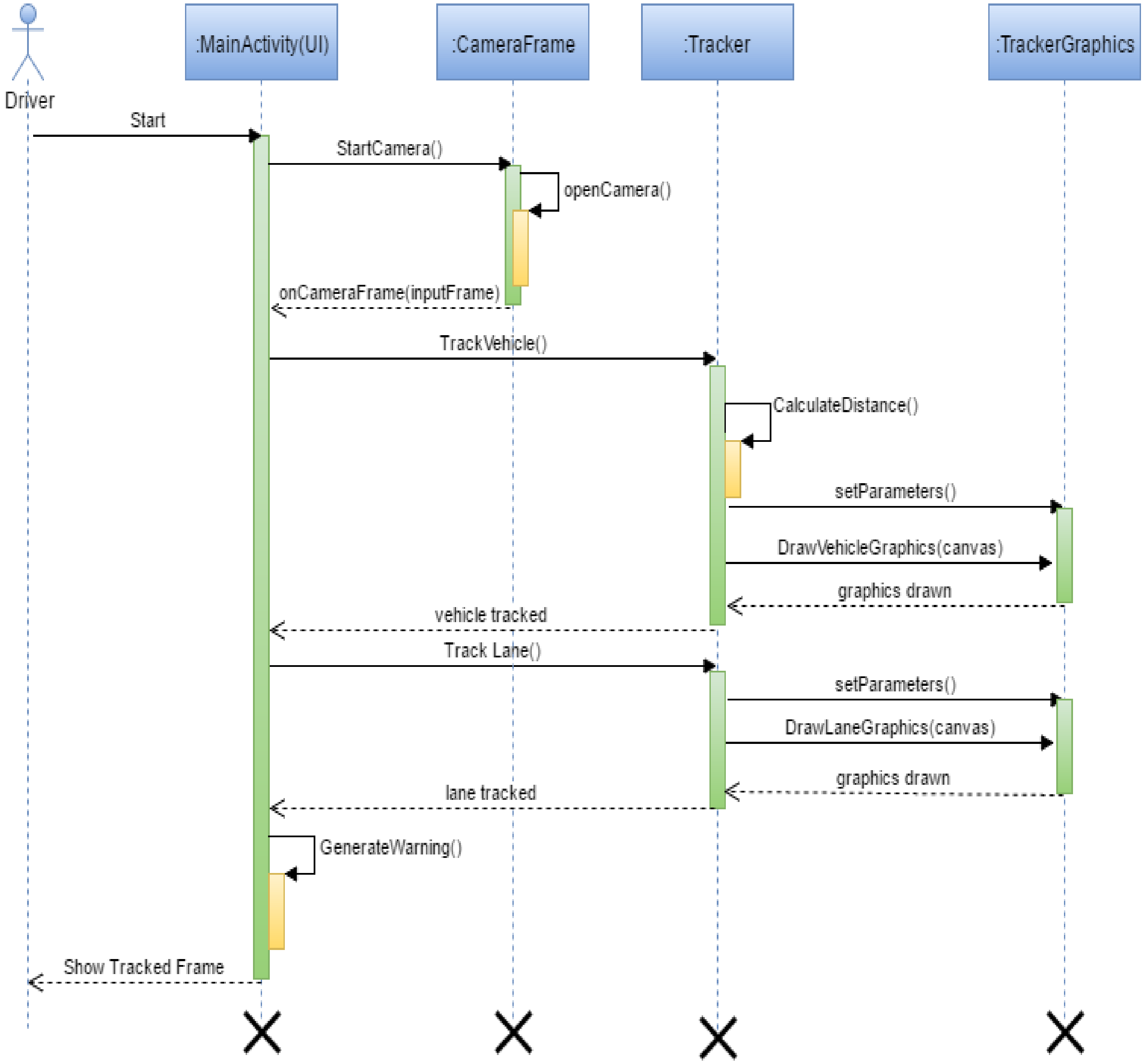


**Figure 17 Sequence Diagram Track Vehicle and Lane**

### 4.4.2.2 View Map:

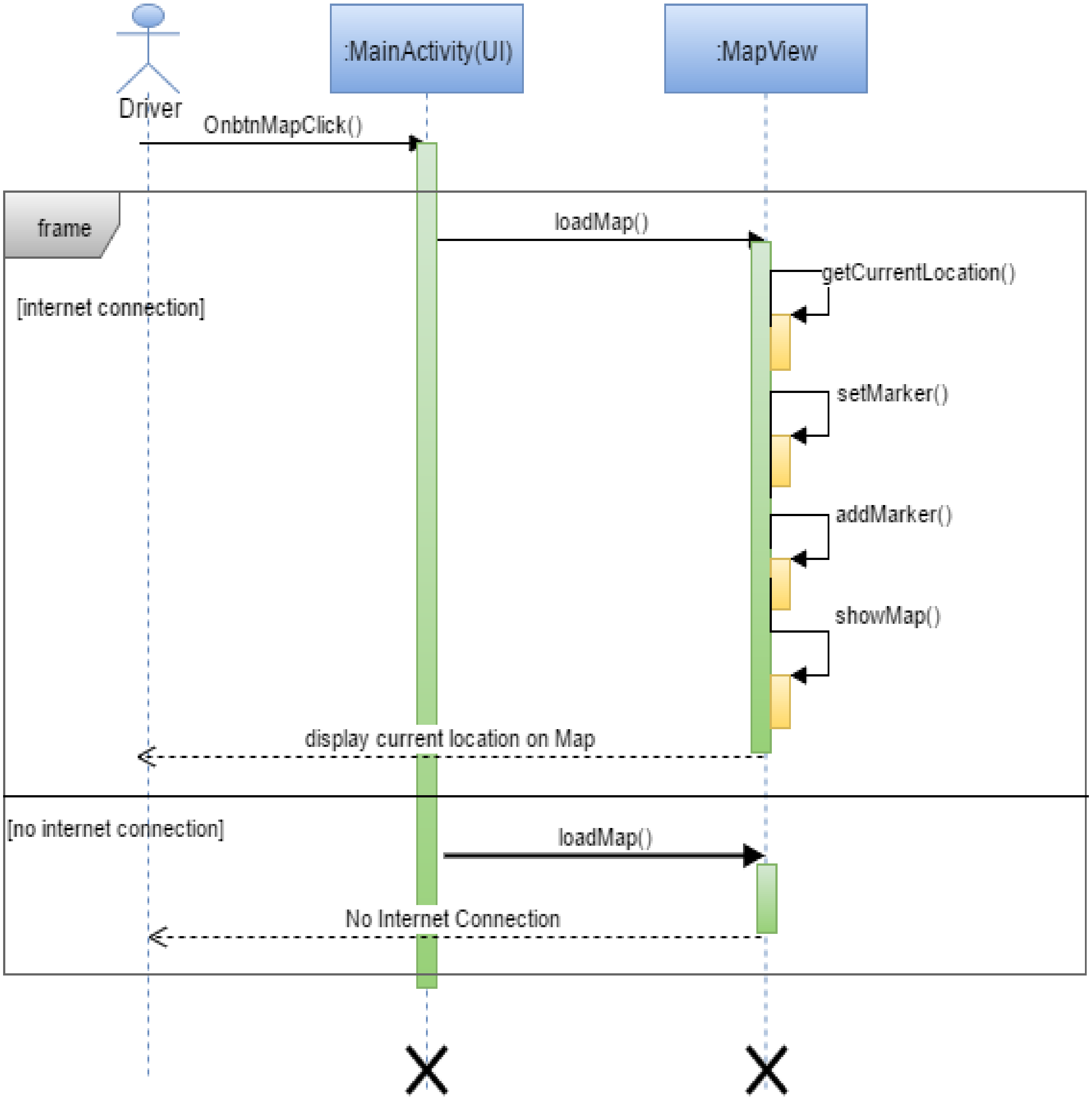


**Figure 18 Sequence Diagram View Map**

### 4.4.2.3 Specify Contact:

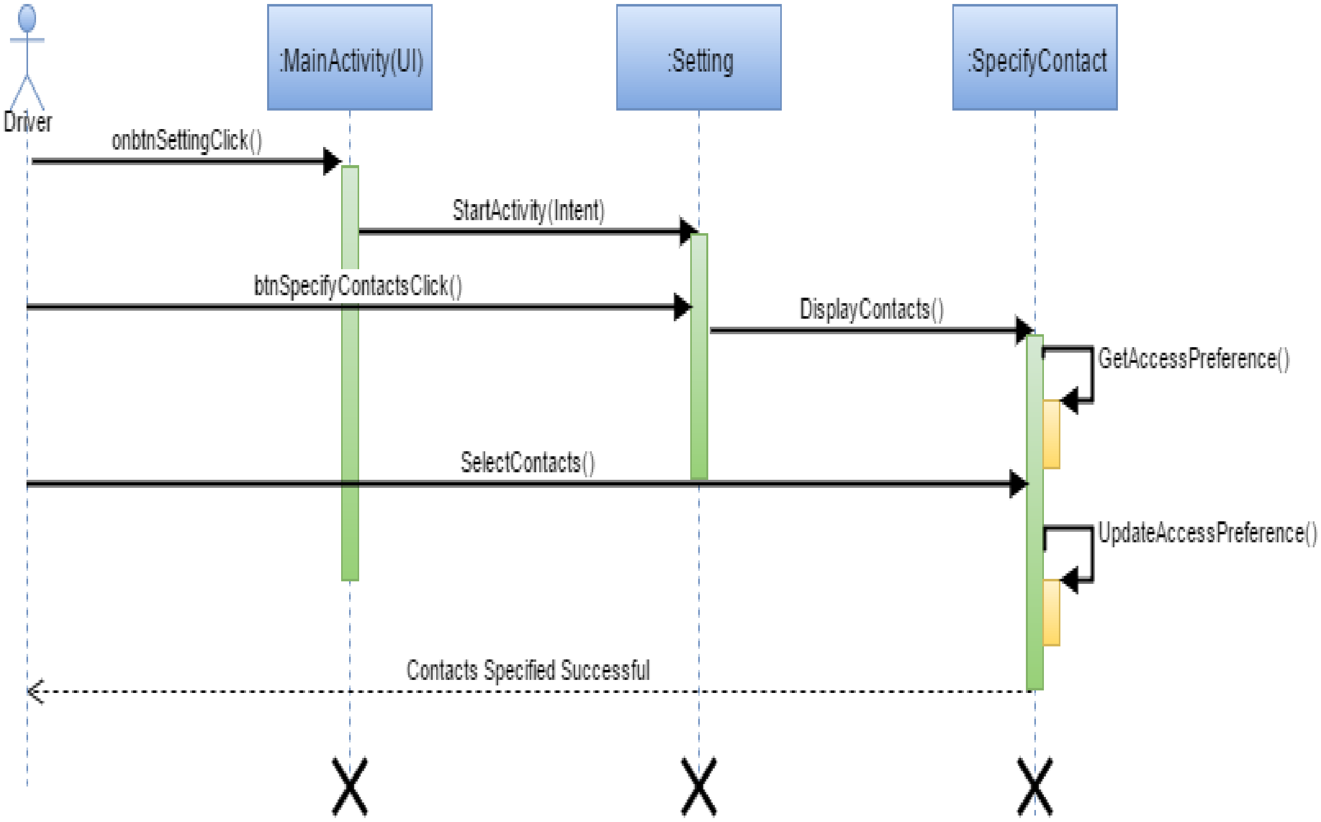


**Figure 19 Sequence Diagram Specify Contacts**

### 4.4.2.4 Remote Third Eye Sequence Diagram:

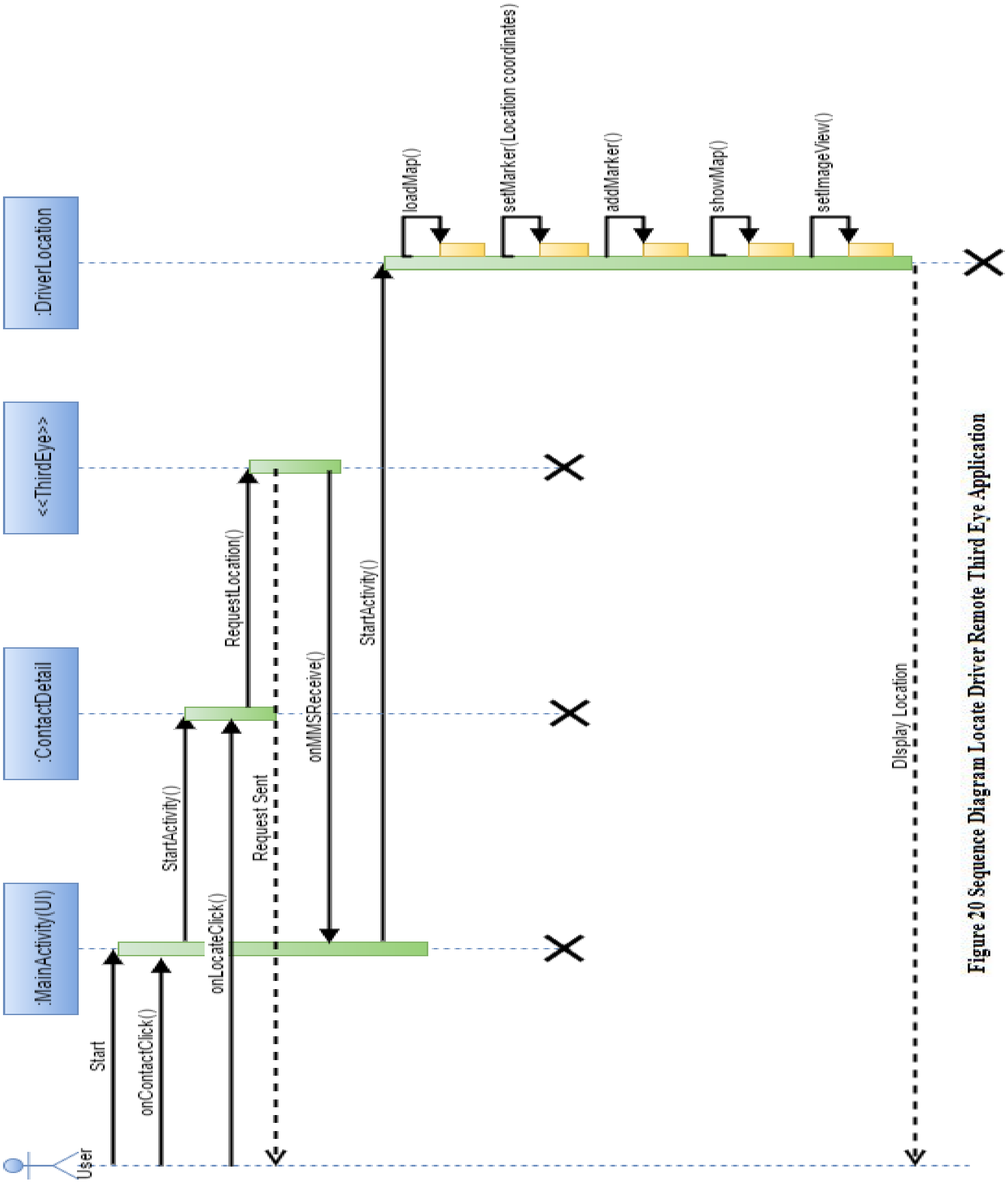


Figure 20 Sequence Diagram Locate Driver Remote Third Eye Application

Figure 20 Sequence Diagram Remote Application

# 5 Implementation

## 5.1 Algorithm

The algorithm developed to implement Third Eye (driver assistance system) includes following steps.

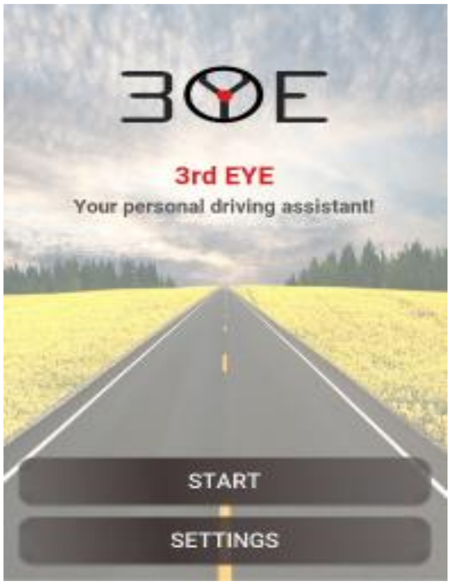
1. The algorithm will capture a frame from the live video feed via smartphone camera
2. Convert the captured frame to gray scale
3. Compute the edge image from the gray scale image
4. Compute edge angles using the edge image
5. Lane detection using Hough Transform
6. Vehicle Detection using HOG
7. Compute detected vehicle's distance

## 5.2 External APIs

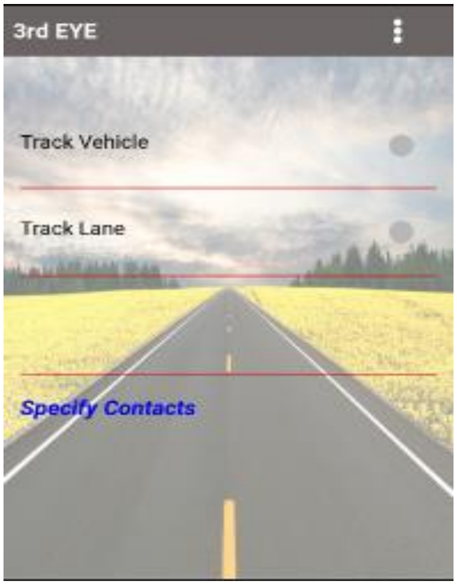
Describe the APIs used in the following table.

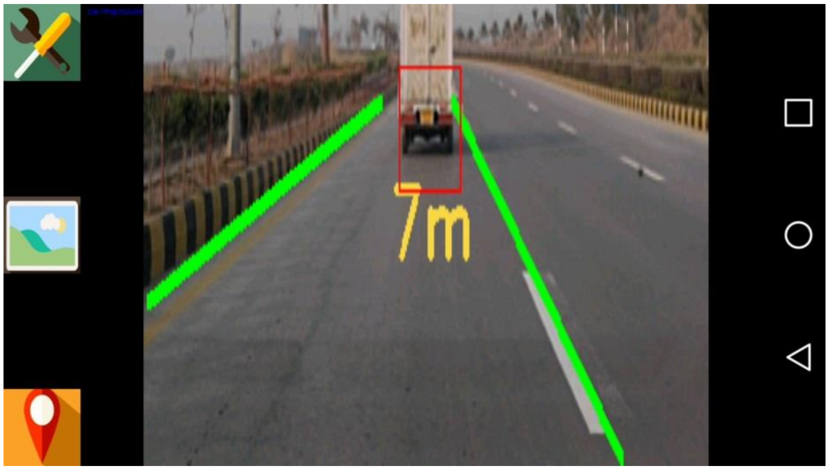
| **Name of API** | **Description of API** | **Purpose of usage** | **List down the function/class name in which it is used** |
|---|---|---|---|
| **Google Maps** | Displays Location on Google Maps | To Display Driver's location on Google Maps | 1. Maps Activity.<br>**2.** Remote App. |

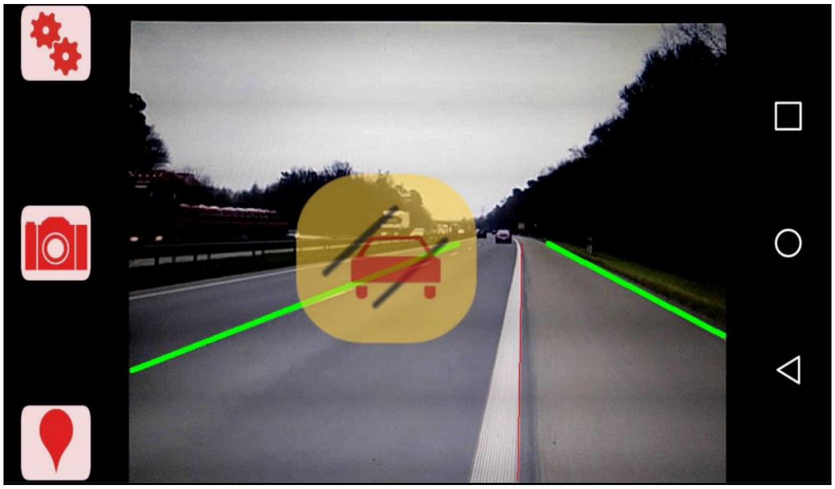
## 5.3 User Interface



Your personal driving assistant!







# 6. Testing and Evaluation

## 6.1 Selected testing methodology

Use-case based technique is used for the testing of Third Eye (driver assistance system).

### 6.1.1 Use-case based testing

#### *6.1.1.1* Start Application

| | | |
|---|---|---|
| **Test Case ID** | **TC-01** | |
| **Test Case Name** | Start Application | |
| **Testing Environment** | Android Smartphone. | |
| **Tested By: Abdullah Akbar** | | **Date: 24th October,2016** |
| **Test Data** | | |
| **Pre-Condition** | 1. Application is downloaded and installed on the phone. | |
| **Test Verification** | This test verifies that the application start successfully. | |
| **Actions** | | **System Response** |
| 1. User taps on the “Third Eye” icon present in the app drawer of the phone. | | 1. System opens startup activity of Application.<br>2. System displays interface of startup activity. |
| **Result:** | **Pass.** The application runs on Android phone successfully | |

#### *6.1.1.2* Capture Image

| | | |
|---|---|---|
| **Test Case ID** | **TC-02** | |
| **Test Case Name** | Capture Image | |
| **Testing Environment** | | |
| **Tested By: Abdullah Akbar** | | **Date: 24th October,2016** |
| **Test Data** | Frames being displayed on the main interface. | |
| **Pre-Condition** | 1. Application is running.<br>2. Application is displaying home interface. | |

| | |
|---|---|
| **Test Verification** | This test case verifies application captures current frame upon click of button. |
| **Actions** | **System Response** |
| 1. User taps on the camera icon present on the main interface at the left side of screen. | 1 System captures the frame being displayed on the interface.<br>2 System stores the captured frame as image in smartphones image gallery.<br>3 System notifies the user with a toast "Captured and Saved". |
| **Result:** | **Pass.** The application captures and stores the current frame as image. |

### *6.1.1.3* View Current Location on Map

| | |
|---|---|
| **Test Case ID** | **TC-03** |
| **Test Case Name** | View Current Location on Map |
| **Testing Environment** | |
| **Tested By: Abdullah Akbar** | **Date: 24th October,2016** |
| **Test Data** | Location Coordinates |
| **Pre-Condition** | 1. Application is running<br>2. Application is displaying home interface.<br>3. Location services is on. |
| **Test Verification** | This test case verifies that application displays current location of user on by mapping longitude and latitude coordinates on Google Maps. |
| **Actions** | **System Response** |
| 1. User taps on the map icon present on the main interface. | 5. System derives latitude and longitude of its current location.<br>6. System maps the location coordinates on Google Maps to display current location.<br>7. System displays current location of user on Google Maps. |
| **Result:** | **Pass.** User's location displayed on Google Maps successfully. |

### *6.1.1.4* Specify Contact

| | |
|---|---|
| **Test Case ID** | **TC-04** |
| **Test Case Name** | Specify Contact. |
| **Testing Environment** | |
| **Tested By: Abdullah Akbar** | **Date: November 1st, 2016** |

| | |
|---|---|
| **Test Data** | Contacts present in the contacts app. |
| **Pre-Condition** | Phone has contacts in the contacts apps. |
| **Test Verification** | The test verifies that the system has added contact for locate request. |
| **Actions** | **System Response** |
| 1 User taps on the settings icon present at the top left corner of main interface. | 9. System displays settings menu.<br>10. User taps on the specify contacts option given in settings menu list.<br>11. System displays specify contact activity.<br>12. User taps on specify contact icon present on the action bar.<br>13. System opens contact book of phone.<br>14. User selects the contact that is to be added.<br>15. System adds the contact into the list of people whose location can be requested.<br>16. System has successfully added a contact to request location. |
| **Result:** | **Pass.** Contact has been successfully added. |

### *6.1.1.5* Request Driver Location

| | |
|---|---|
| **Test Case ID** | **TC-05** |
| **Test Case Name** | Request Driver Location |
| **Testing Environment** | |
| **Tested By: Abdullah Akbar** | **Date: 24th October,2016** |
| **Test Data** | |
| **Pre-Condition** | Remote Application is running. |
| **Test Verification** | This test case verifies that system has successfully pinged the driver for his current location. |
| **Actions** | **System Response** |
| 1. User opens the Remote Application.<br>2. User taps on locate icon present next to the contact name on the main interface. | System sends a message to the selected contact to request his location. |
| **Result:** | **Pass.** Selected contact has been successfully pinged for location coordinates and current screen image. |

### *6.1.1.6* Display Driver Location

| | |
|---|---|
| **Test Case ID** | **TC-06** |

| | |
|---|---|
| **Test Case Name** | Display driver location |
| **Testing Environment** | |
| **Tested By: Abdullah Akbar** | **Date: November 1st, 2016** |
| **Test Data** | Image and location coordinates received from the driver. |
| **Pre-Condition** | 1. Remote application user has inquired the driver for his location.<br>2. Remote application is connected to the internet. |
| **Test Verification** | This test case verifies that system displays the received image on interface. And the system maps the received location coordinates on the Google map. |
| **Actions** | **System Response** |
| .Third Eye app sends a text containing location coordinates to Remote app. | 1. System accesses the server to retrieve the image of current screen of driver.<br>2. System maps the location coordinates on Google Map.<br>3. System displays the retrieved image on image view.<br>4. System maps the received location coordinates on Google Maps.<br>5. System displays the current frame being tracked on remote application interface. |
| **Result:** | **Pass.** Location of driver is successfully mapped on Google Maps and current screen image of driver is successfully displayed on the image view. |

### *6.1.1.7* Track Vehicle

| | |
|---|---|
| **Test Case ID** | **TC-07** |
| **Test Case Name** | Track Vehicle |
| **Testing Environment** | **Roads** |
| **Tested By: Abdullah Akbar and Syed Ali Kazim** | **Date: 1st November, 2016** |
| **Test Data** | |
| **Pre-Condition** | 1. Smartphone is mounted on the windshield of car horizontally.<br>2. Application is running. |
| **Test Verification** | Application identifies and tracks vehicles present in the live video feed.<br>Application generates warning when the preceding car is at critical distance. |
| **Actions** | **System Response** |
| User taps on Start tracking button on startup activity. | Application opens new activity and starts tracking vehicles. |
| **Result:** | **Pass.** Application is successfully tracking vehicles from live video feed. |

### *6.1.1.8* Track Lane

| | |
|---|---|
| **Test Case ID** | **TC-08** |
| **Test Case Name** | Track Lane |
| **Testing Environment** | **Roads** |
| **Tested By: Abdullah Akbar and Syed Ali Kazim** | **Date: 1st November, 2016.** |
| **Test Data** | |
| **Pre-Condition** | 1. Smartphone is mounted on the windshield of car horizontally.<br>2. Application is running |
| **Test Verification** | Application detects and tracks lanes present on the road. If car leaves its lane. Warning is generated. |
| **Actions** | **System Response** |
| User taps on Start tracking button on startup activity. | Application opens a new activity and starts tracking lanes of road. |
| **Result:** | **Pass:** Application successfully detects and tracks the lanes present on the road. |

### *6.1.1.9* Calculate Distance

| | |
|---|---|
| **Test Case ID** | **TC-09** |
| **Test Case Name** | Calculate Distance |
| **Testing Environment** | **Roads.** |
| **Tested By: Abdullah Akbar and Syed Ali Kazim** | **Date: 1st November, 2016.** |
| **Test Data** | |
| **Pre-Condition** | Application is running and vehicle is detected by the app. |
| **Test Verification** | Application calculates and displays the distance of detected vehicle ahead. |
| **Actions** | **System Response** |
| User will tap on Start Tracking button present on the startup activity. | 1. System will detect vehicles.<br>2. System will calculate and display distance of detected vehicle. |
| **Result:** | **Pass.** System successfully calculates and displays distance of the detected vehicle ahead. |

#### *6.1.1.10* Generate Warning

| | | |
|---|---|---|
| **Test Case ID** | **TC-10** | |
| **Test Case Name** | Generate Warning | |
| **Testing Environment** | **Roads.** | |
| **Tested By: Abdullah Akbar and Syed Ali Kazim** | | **Date: 1st November, 2016.** |
| **Test Data** | | |
| **Pre-Condition** | Application is running. | |
| **Test Verification** | 1. Application generates warning if preceding vehicle is at critical distance<br>2. Application generates warning if car leaves its lane. | |
| **Actions** | | **System Response** |
| User will tap on Start Tracking button present on the startup activity. | | 1. System will detect vehicles and lanes on the road.<br>2. System will generates warning if preceding vehicle is at critical distance or if car leaves its lane. |
| **Result:** | **Pass.** System successfully generates warning if vehicle ahead is at critical distance or the driver's car leaves its lane on the road. | |

## 6.2 Tools used

| Tool Name | Tool Description | Applied on [list of related test cases/FR/NFR] | Results |
|---|---|---|---|
| MATLAB | Tool for designing image processing algorithms | Track vehicle, track lane | Excellent |
| Visual Studio | Implementation of MATLAB algorithm in C++ using OpenCV | Track vehicle, track lane | Excellent |
| Android Studio | Implmentation of C++ algorithm in native and other android app features | Track vehicle, track lane, generate warning, specify contacts, calculate distance, display/request/view driver location | Excellent |

# 7 Conclusion and Future Work

## 7.1 Conclusion

The main features of this system include lane tracking and vehicle tracking. There are existing applications which provide number of features for driver assistance to ensure driver safety like speed calculation of driver’s car or speech recognition to control application. However these features come at a cost and makes use of expensive sensors. There are cars with built-in lane and vehicle detection systems. But these driver assistance systems are only present in expensive luxury cars like Mercedes, Audi, Maserati and BMW.

To conclude, the implemented system that is capable of providing driver assistance on a small scale through a smartphone camera without the use of heavy sensors and expensive radar systems.

## 7.2 Future Work

The idea is to take the system to product level by deploying it on a robot, in order to make the robot autonomous. The system will allow the robot to be mobile without any remote controls.

We also plan to make a finished product out of this system so that it can be used as a built-in system or can be embed in Pakistani automobiles. Through this the cheap Pakistani vehicles like Suzuki Mehran and Alto would have luxury driver assistance features like vehicle tracking, lane tracking and collision warnings.

# 8 References

1. Raad Ahmed Hadi, Ghazali Sulong and Loay Edwar George, "Vehicle detection and tracking techniques: A concise review," in Signal & Image Processing: An International Journal (SIPIJ) Vol.5, No.1, February 2014.
2. Pooja Sharma,Gurpreet Singh, Amandeep Kaur, "Different techniques edge detection in digital image processing," in International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 3, May-Jun 2013, pp.458-461.
3. Gayathiri Somasundaram, Kavitha, K.I.Ramachandran, "Lane change detection and tracking for a safe-lane approach in real time vision based navigation systems,".
4. Eduardo Romera, Luis M. Bergasa, Roberto Arroyo, "A real-time multi-scale vehicle detection and tracking approach for smartphones," IEEE 18th International Conference on Intelligent Transportation Systems, 2015.
5. Kungwon Chnag, Byung-Hun Oh and Kwang-Seok Hong, "An implementation of smartphone based driver assistance system using front and rear camera," in Institute of electrical and electronics engineers - Oct 8,2014.