# Software Requirements Specification

**for**

# DRIVER DROWSINESS SYSTEM

**Version 1.0 approved**

**Prepared by**

1. **Vaishnavi Malshikare**

**ORGANIZATION - AISSMS IOIT**

**Date created –**

# Table of Contents



# Revision History

| Name | Date | Reason For Changes | Version |
|---|---|---|---|
| | | | |
| | | | |

# 1. Introduction

## 1.1 Purpose

According to surveys by the World Health Organization (WHO) on road accidents, about 1.3 million people die every year on road highways in 2018. Also, a survey report and research provided by the National Highway Traffic Safety Association (NHTS) on road accidents, about 795 people die from drowsy-driving and 91,000 people die from motor vehicle crashes involving drowsy driving in 2017. So, the drowsy driver is considered as one of the factors for road accidents.The purpose of the drowsiness detection system is to aid in the prevention of accidents passenger and commercial vehicles. The system will detect the early symptoms of drowsiness before the driver has fully lost all attentiveness and warn the driver that they are no longer capable of operating the vehicle safely.

## 1.2 Intended Audience

The different types of readers that can be influenced by our project are teachers, students, entrepreneurs, car manufacturing companies, etc.

## 1.3 Product Scope

There are many products out there that provide the measure of fatigue level in the drivers which are implemented in many vehicles. The driver drowsiness detection system provides the similar functionality but with better results and additional benefits. Also, it alerts the user on reaching a certain saturation point of the drowsiness measure.

Following are some of the objectives of this android based drowsiness detection system:

a. Research on the various system to detect drowsiness
b. Understand the working flow of the system
c. Implementation of an accurate algorithm for the motion of eye detection and produce alert sound after detecting drowsiness
d. Testing performance of the system
e. Prepare report based on the project

## 1.4 References

The lists of web sites are

1. https://graspcoding.com/driver-drowsiness-detection-system-ai-project/
2. https://en.wikipedia.org/wiki/Driver_drowsiness_detection
3. https://ieeexplore.ieee.org/document/6602353/
4. https://en.wikipedia.org/wiki/Face_detection#:~:text=Face%20detection%20is%20a%20computer,faces%20in%20a%20visual%20scene.
5. https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9

# 2. Overall Description

## 2.1 Product Perspective

This project is based on the driver drowsiness system. Drowsiness is a complex phenomenon which states that there is a decrease in alerts and conscious levels of the driver. Though there is no direct measure to detect the drowsiness but several indirect methods can be used for this purpose. Several measures are available for the measurement of drowsiness which includes the following:

1. Vehicle based measures - Vehicle-based measures survey path position, which monitors the vehicle's position as it identifies with path markings, to determine driver weakness, and accumulate steering wheel movement information to characterize the fatigue from low level to high level. In many research project, researchers have used this method to detect fatigue, highlighting the continuous nature of this non-intrusive and cost-effective monitoring technique.

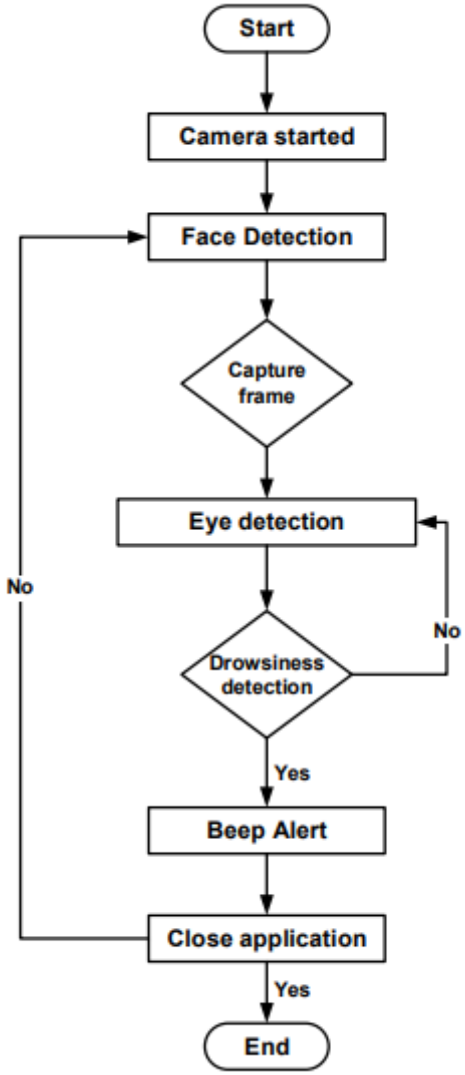
2. Physiological measures - Physiological measures are the objective measures of the physical changes that occur in our body because of fatigue. These physiological behaviours can be measured by by monitoring the heart beats, blood pressure, eye movements, etc.

3. Behavioral measures - Certain behavioral changes take place during drowsing like
    - Yawning
    - Amount of eye closure
    - Eye blinking
    - Head position

## 2.2 Product Functions

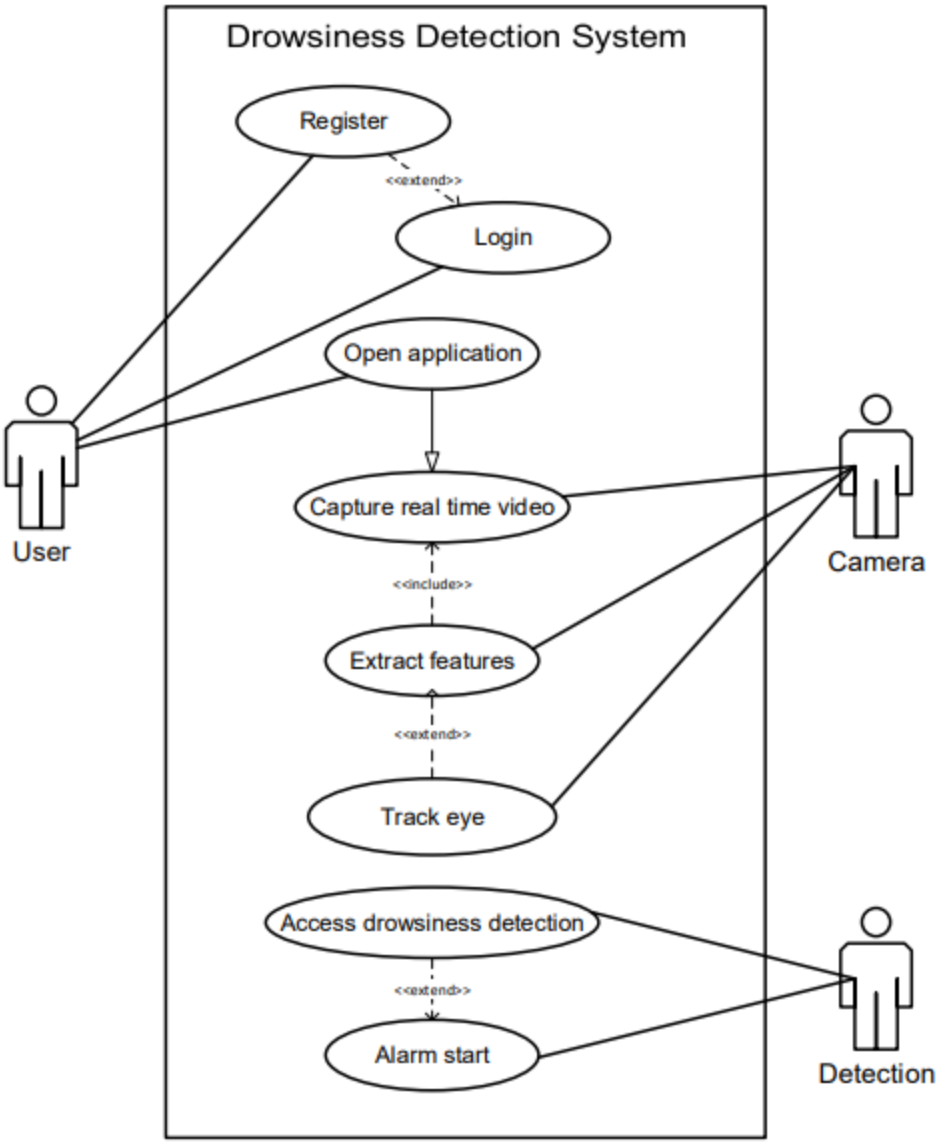
Some of the technical functions the system will perform are:

- Facial Detection
- Facial Recognition
- Alert System




## 2.3 User Classes and Characteristics

Software team needs to provide a simple and user friendly interface that is easy to cope with. A midlevel phone user must comprehend the system just with reading simple instruction manual about command list. Command list should not be very complex. An average user should easily understand and memorize commands. A user with small speaking flaws must be tolerated by the system.




## 2.4 Operating Environment

Software Platform

- It will operate on Windows operating system
- Python 3
- OpenCV Library
- Python IDLE

Hardware Platform

- Laptop with basic requirements to run python environment
- Web cam
- Raspberry Pi

## 2.5 Design and Implementation Constraints

1. The device should measure what it is intended to, operationally E.g. eye blinks, heart rate, jaw movement and conceptually e.g., alertness.
2. The device should monitor driver behavior in real time
3. The device should be consistent in its measurement over time, and it should measure the same event for all drivers.
4. The device should be able to operate accurately and reliably in both day time and night time illuminations conditions.

## 2.6 Assumptions and Dependencies

*While researching and building a project, fatigue is observed using image recognition and computer vision techniques. These fatigues are dependent on:*

a. Keeping the user's head down causes fatigue detection inaccuracy
b. There is no dual user usability available
c. The eye-detection algorithm which plays an important role in detecting drowsiness creates a high degree of misunderstanding when tested with different positions of eyes.
d. It is only prevention, not a solution
e. If there is less light then there is variability and errors in face detection and recognition.

# 3. External Interface Requirements

## 3.1 User Interfaces

The web cam will be mounted near the steering wheel of the car. The web cam will be connected to the ignition of the car. Once, the car starts, the live feed will start and real time facial detection and recognition will be performed. If the driver is sleepy, then the system detects the state and will raise an alarm or signal in form of beeps and lights. The driver will be provided an external switch to turn off the beeping manually.

## 3.2 Hardware Interfaces

Software will run on all the platforms i.e. windows, mac os, andriod, ios, linux, and web. The sytem will be downloaded and stored in the device itself.

## 3.3 Software Interfaces

For the driver drowsiness system, an interactive python IDLE will be required which will use the OpnenCV library for facial detection and recognition.

## 3.4 Communications Interfaces

The system will warn with a beeping and a red light to the user if the user is sleepy. There will be a manual button provided to the user so that he/she can turn off the alarm and reset it once it starts beeping.

# 4. System Features

## 4.1 Face Detection

### 4.1.1 Description and Priority

Face Detection is one of the pillars in driver drowsiness detection. It is of extreme high priority. We will detect the drowsiness of the drives using face recognition methods.

### 4.1.2 Stimulus/Response Sequences

As soon as the user enters the car and turns on the ignition, our camera and live feed will start. As soon as the car starts, the face of the driver will be detected in a live stream.

### 4.1.3 Functional Requirements

REQ-1: Load image as input - The system shall receive as input an image or picture file from the user. User shall be able to load the image from a folder or file. This image shall be loaded into the system and used as a source for the subsequent face detection task.

REQ-2: Detect human face(s) in input image - The system shall be able to detect the presence of human face(s) contained in the input image. The detected human face(s) will be automatically marked with rectangle on the original image. The system will redisplay/reproduced this image as output.

## 4.2 Eye Detection and State of Eye

### 4.2.1 Description and Priority

After successful face detection, eyes should be detected for further processing. This is one of the important features as the state of drowsiness of the driver would be detected using his eyes. In our method, eyes would be the decision parameter for finding the state of the driver.

### 4.2.2 Stimulus/Response Sequences

It will detect the eye pupil and eye lid movement of the driver. Based on which, it the system will predict if the driver is sleepy or not. If sleepy, an alert will be raised.

### 4.2.3 Functional Requirements

In this stage, we find the actual state of the eye that if it is closed or open or semi closed or open. The identification of eyes status is most important requirement. It is achieved by an algorithm which will be clarified in the later parts. We channelize a warning message if we obtain that the eyes are in open state or semi open state up to a particular threshold value. If the system detects that the eyes are open then the steps are repeated again and again until it finds a closed eye

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

The performance requirements are:

- The web camera should be of good quality so that there are minimum errors in the facial detection as well as drowsiness prediction.
- The processor (like raspberry pi) should have high processing speed so that there is minimum delay in prediction the state of driver.

## 5.2 Safety Requirements

The beeping and the warning lights displayed will be cautiously installed so that there is no sudden distraction for the driver if he is in a sleepy state. This will help the driver to take the wheel more effectively once the warning is triggered.

## 5.3 Security Requirements

- Confidentiality: The software does not store any kind of live feed. The camera will neither record nor click any image. There will be no breakage of privacy and proprietary information.
- Integrity: The personal information or live feed of the user will not be stored or copied by the software.

## 5.4 Software Quality Attributes

1. Adaptability – The software can easily adapt to any car. It will not be dependent on the size or model or company of the car.
2. Correctness – The software will provide effective and fast results of the state of the driver.
3. Flexibility – It will be very easy to install and update additional features to the system.
4. Maintainability – It will require minimum maintenance as there will not be any regular or constant updates to it and it will not get corrupted easily.
5. Portability – It can be easily installed in any car and can be easily uninstalled from a particular car.