DROWSINESS DETECTION SYSTEM

A Project Report

Submitted in partial fulfilment of the
requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

MAURYA VIVEK

Seat Number: 18302A0025

Under the esteemed guidance of

Mrs. Maitreyi Joglekar

Assistant Professor, Department of Information Technology

DEPARTMENT OF INFORMATION TECHNOLOGY

VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY

(Affiliated to University of Mumbai)

MUMBAI, 400 037

MAHARASHTRA

2020 - 2021

VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY

(Affiliated to University of Mumbai)

MUMBAI-MAHARASHTRA-400037

DEPARTMENT OF INFORMATION TECHNOLOGY

VSIT | Vidyalankar School of Information Technology

CERTIFICATE

This is to certify that the project entitled, **" Drowsiness Detection System."** is bonafide work of **MAURYA VIVEK** bearing Seat No: 18302A0025 submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide **Coordinator**

Internal Examiner **External Examiner**

Date:

College Seal **Principal**

ABSTRACT

This paper presents a real-time driver drowsiness detection system for driving safety. Based on computer vision techniques, the driver's face is located from a colour video captured in a car. Then, face detection is employed to locate the regions of the driver's eyes, which are used as the templates for eye tracking in subsequent frames. Finally, the tracked eye's images are used for drowsiness detection to generate warning alarms. The proposed approach has three phases: Face, Eye detection and drowsiness detection. The role of image processing is to recognize the face of the driver and then extracts the image of the eyes of the driver for detection of drowsiness. The face detection algorithm takes captured frames of image as input and then the detected face as output. Next, dlib is used to tracking eyes from the detected face. If the eyes are closed for a predefined period, the eyes of the driver will be considered closed and hence an alarm will be started to alert the driver. The proposed system was tested on a Raspberry pi 4 . The experimental results appear quite encouraging and promising. The system could reach more than 20 frames per second for face and eye tracking, and the average correct rate for eye location and tracking could achieve 99.0% on some test videos. Thus, it can be concluded that the proposed approach is a low cost and effective solution method for a real-time of driver drowsiness detection.

Keywords: Smart Detection System, companion, social, secure, reliable.

ACKNOWLEDGEMENT

We would like to express our special thanks and gratitude to ur project guide **Mrs. Maitreyi Joglekar** for guiding us to do the project work on time and giving us all support and guidance, which made complete our project duly. We are extremely thankful to her for providing such nice support and guidance.

We are also thankful for and fortunate enough to get constant encouragement, support, and guidance from the teachers of information Technology who helped us in successfully completing our project work.

DECLARATION

I hereby declare that the project entitled, "**Drowsiness Detection System.**" done at Vidyalankar School of Information Technology, has not been in any case duplicated to submit to any other universities for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Name and Signature of the Student

Contents

CHAPTER 1: INTRODUCTION 8
1.1 Background 8
1.2 Objectives 9
1.3 Purpose, Scope and Applicability 9
1.3.1 Purpose 9
1.3.2 Scope 10
1.3.3 Applicability 10
CHAPTER 2: SURVEY OF TECHNOLOGY 13
2.1 Technologies Used: 13
2.1.1 Programming language Python: 13
2.1.2 Introduction to Dlib: 13
2.1.3 Dlib Face landmark Detector: 13
2.1.4 Open CV: 14
2.1.5 Raspberry Pi 4: 14
2.1.6 Raspbian OS: 15
2.1.7 Display (3.5" touch display): 16
2.2 Comparison of Technology: 16
2.2.1 Python VS C/C++: 16
2.2.1 Raspberry pi VS Arduino: 17
CHAPTER 3: REQUIREMENTS AND ANALYSIS 19
3.1 Problem Definition 19
3.2 Requirement Specification 20
3.3 Planning and Scheduling 20
3.4. Software and Hardware Requirements 21
3.4.1 Software Requirement 21
3.4.2 Hardware Requirement 22
3.5 Preliminary Product Description: 27
3.6 Conceptual Model 27
CHAPTER 4: SYSTEM DESIGN 30
4.1 Basic Modules: 30
4.2 Data Design: 30
4.3 Diagrams 30
4.3.1 Use Case Diagram: 30
4.3.2 Activity Diagram: 31

4.3.3 Block Diagram: 32
4.3.4 Sequence Diagram: 33
4.3.5 Data Flow Diagram: 34
4.3.6 Circuit Diagram: 35
4.3.7 Event Table: 35
4.3.8 User Interface Design: 35
4.3.9 Test Cases Design: 37

CHAPTER 1: INTRODUCTION

1.1 Background

Basically, our project is based on drowsiness which name as Drowsiness Detection System.

In our project we are using OV5647 5MP 1080P IR camera which takes video input and pass it to the raspberry pi, after that our raspberry pi play an important role and with the help of program it detects drowsiness and passes instructions to Buzzer ,LED and Motor accordingly, simultaneously data of how many times alarm is beeping will be stored in the file and that can be send to another device for further calculation to give ratings accordingly as per the efficiency of the driver .

This project we had chosen because in day to day life we observe Drowsy driving is one of the major causes behind fatal road accidents. One of the recent studies shows that one out of five road accidents are caused by drowsy driving which is roughly around 21% of road accidents, and this percentage is increasing every year as per global status report on road safety 2015.

The drowsiness cannot be detected by traffic police like alcohol detection before the accident takes place. Hence to prevent this problem we are placing OV5647 5MP 1080P IR camera which continuously monitor person eye once the car started. So, if the person tries to operate the car if he is sleepy then the drowsiness will be detected.

1.2 Objectives

The target of this project is to give an idea and inventive method for avoiding drowsy driving of a car. Likewise, to permit a man who is not drowsy to drive the same car. To broaden this thought with more innovative headways and make it accessible in a financially effective way. We need to plan a sort of framework which can recognize the drowsy person in the cars to prevent drowsy driving.

This system is used for preventing the car accidents using OV5647 5MP 1080P IR camera by taking video input and processing through the help of raspberry pi4 by using python programming language. This system uses number of features of dlib (get_frontal_face_detector,68 facial landmark predictor) and open computer vision for processing the video data and detecting drowsiness. Once the drowsiness of a person is detected the alarm will be beep and if the person is awake he will stop the alarm process with the help of reset button if not the process will go ahead and dc motor will slow down simultaneously the LED which is behind the car will glow.

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

Driver drowsiness detection is a car safety technology which helps prevent accidents caused by the driver getting drowsy. Various studies have suggested that around 21% of all road accidents are fatigue-related, up to 50% on certain roads. Our project is being done for preventing the car accident. It has a real time application system. Our project reduces the number of drossy drive cases, it is useful for public safety as well as it has positive impact on surrounding. Due to our project the traffic rules will be maintain and the moto "**Alert today – Alive tomorrow**".

1.3.2 Scope

The system we are developing majorly aims on providing a safety system to prevent road accidents. We applied many methodologies by interfacing OV5647 5MP 1080P IR camera, Buzzer, DC motor with raspberry pi. We have also include led for showing car taillight.

The limitations of our project: It can be used in only cars.

To build a car-control system that restricts driver from operating the car if he is declared drowsy. To develop this proposed system, we required minimal number of tools, the entire system will be relying on these major components, OV5647 5MP 1080P IR camera, Raspberry pi4, Buzzer(5v), LED(5v).

This system has user friendly interface which will provide selection for old and new user options which will help to calculate the eye aspect ratio of the driver.

1.3.3 Applicability

Technical Feasibility Study

Features of Raspberry pi 4: -

1.Open Source

2.Easy to Implement

3.Fast processor

4.More Amt of RAM

5.On Board Wi-Fi

6.Various Pre-Installed IDEs

♦ Why not another board?

We have another option also available for our project which is Arduino Uno, which is not as powerful as Raspberry pi 4, but it can be great for quick setups. It is a minicomputer. The Arduino Uno runs one program at a time on the other hand raspberry pi 4 runs multiple programs at a time.

The Raspberry Pi 4 can connect to Bluetooth devices and the Internet right out of the box using Ethernet or by connecting to Wi-Fi. The Arduino Uno cannot do that without a Shield that adds Internet or Bluetooth connectivity. HATS and Shields help with this.

Difference between Raspberry pi 4 & Arduino Uno

	Raspberry Pi 4	**Arduino Uno**
Price	5400 INR	400 INR
Size	8.8cm*5.8cm*1.95cm	7.6cm*1.9cm*6.4cm
Memory	4 GB	0.002mb
Clock Speed	1500 MHz	16MHz
Multitasking	Yes	No
Input Voltage	5v	7 to12v
Operating System	Linux distribution	None
IDE	Scratch, Thonny Python, anything Linux supports	Arduino

USB	Two USB 2.0, Two USB 3.0, Two Micro HDMI ports, Ethernet port etc.	One, Input only

Operational Feasibility Study

In this phase, we studied the operational implementation of the system. It is the Activity stream which can help the user.

- **Old User**

The old user can see their name which is already stored in the system. After choosing the name the system will use the associated data to predict the drowsiness.

- **New User**

The new user will enter the name and EAR (Eye Aspect Ratio) will be calculated and stored in the system. This data will be used to predict the drowsiness.

- **Reset**

It can be clicked by the user to reset the alarm count if user wakes up after the first alarm.

♦ **Benefits are as follows: -**

1. Detects drowsiness.
2. Decreasing road accidents.
3. Optimum use of Resources.
4. Saves human life.
5. Works in low light.
6. Help to rate the driver's efficiency.

CHAPTER 2: SURVEY OF TECHNOLOGY

2.1 Technologies Used:

2.1.1 Programming language Python:

According to the latest Programming Community Index, Python is one of the top 10 most popular programming languages of 2017.You can use Python for developing desktop GUI applications, also websites and web applications. Python can be used in various fields even IOT where you can create simple syntax and work on your device. Python is a simple programming language which can be easily used by anyone and everyone. Choosing Python over other programming languages gives you ease and faster working on it. It is the official language to be used with Raspberry Pi. Python is a very easy language to master and easy to use and deploy, so you do not need to spend too much of your time in learning the formatting standards or compiling processes. Python can be used in embedded systems, it is Portable, and Expandable. It supports a variety of single board computer systems, no matter what the architecture and operating system is. Python includes a lot of help and libraries online which provides help for the language. Nowadays most of the microcontrollers are using Python. For example, versions like Micro Python boards.

2.1.2 Introduction to Dlib:

Dlib is a general purpose cross-platform software library written in the programming language C++.Its design is heavily influenced by ideas from design by contract and component-based software engineering. Thus it is, first and foremost, a set of independent software components. It is open-source software

2.1.3 Dlib Face landmark Detector:

The face detector we use is made using the classic Histogram of Oriented Gradients (HOG) feature combined with a linear classifier, an image pyramid, and sliding window detection

scheme. The pose estimator was created by using dlib's implementation of the paper, One Millisecond Face Alignment with an Ensemble of Regression Trees by Vahid Kazemi and Josephine Sullivan, CVPR 2014 and was trained on the iBUG 300-W face landmark dataset (see https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/):

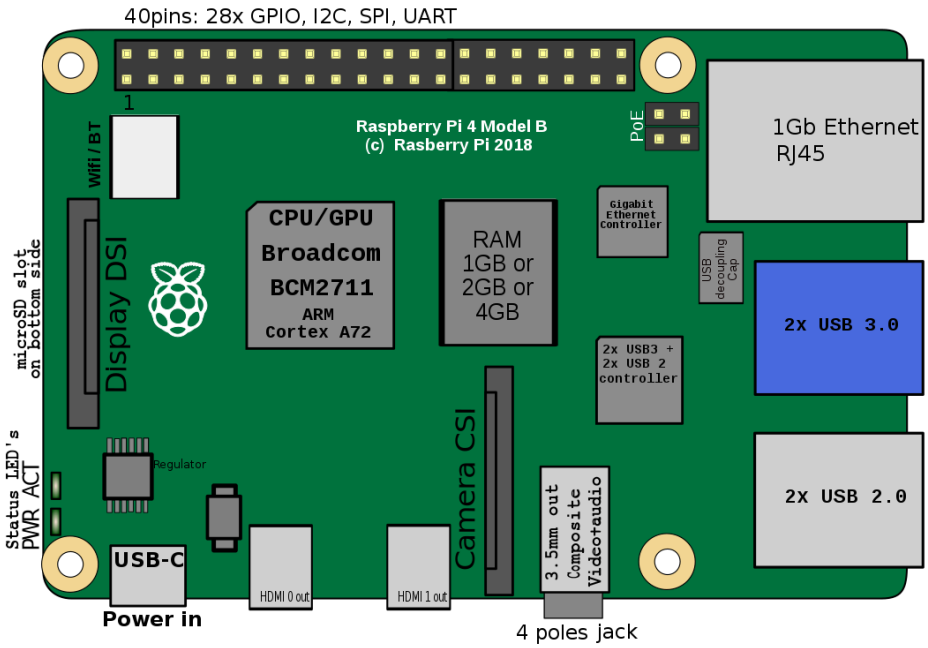
2.1.4 Open CV:

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use.

Our project is based on drowsiness detection system, so we are using OpenCV to take video input and find facial landmarks and detect whether person is drowsy or not.

2.1.5 Raspberry Pi 4:

Our project is based on OV5647 5MP 1080P IR camera it helps to get video input from the user and then passes to the raspberry pi. Once the video gets to the raspberry pi then it is divided into the frames, Raspberry pi here play an important role which acts as a brain of our project and detects drowsiness of a person.



Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. It is the smallest and cheapest CPU available, Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz Broadcom Video Core, USB ports: 4 and 40 GPIO pins. It has its own Operating system "RASPBIAN". It is integrated with Python IDE. It is probably the best option available when it comes to IOT, but its cost is approximately 5400(Raspberry Pi 4).

2.1.6 Raspbian OS:

- Raspbian is the free operating system based on the Debian optimized for the Raspberry Pi hardware.
- OS is the set of most basic programs and the utilities that make your Raspberry Pi run.
- It is more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.
- Raspbian uses PIXEL, Pi improved X-windows Environment Lightweight as its main desktop environment as of the latest update.

- It is composed of a modified LXDE environment and the Open box stacking window manager with a new theme.

2.1.7 Display (3.5” touch display):

Designed for Raspberry Pi Model B/B+, 3.5inch RPi LCD (A) is an ideal alternative solution for HDMI monitor. Combined with portable power this display module is the most convenient men-machine interface for Raspberry Pi. It supports Raspbian system, enabling your system to; play videos (supports multi formats, MP4 and so on) and take photos by touching (up to 17 camera modes). Also, it supports software keyboard aiding system interaction without keyboard or mouse. It can be even used in your customized Raspbian system directly.

2.2 Comparison of Technology:

2.2.1 Python VS C/C++:

- **Python**

Python is the most popular introductory programming language according to a

study by the Barr Group, eight of the top ten CS departments currently use Python to teach coding. It is a simple fact that the most widely understood language in the hiring pool of recent graduates is Python. It is more likely that a recent graduate will understand how to code in Python than in C/C++. Python is not only most popular language for introductory CS programs, Its also the fastest-growing language for embedded computing. The industry with experienced programming drones, robots, or other projects frequently have an Arduino or Raspberry Pi background.

- **C/C++**

  While C++ is slow to write, error-prone, and frequently unreadable, & Python is known for its writability, error reduction, and readability. This design of python outclasses C/C++. In the todays Agile environment design reuse can be the difference between staying ahead or falling behind the competition.

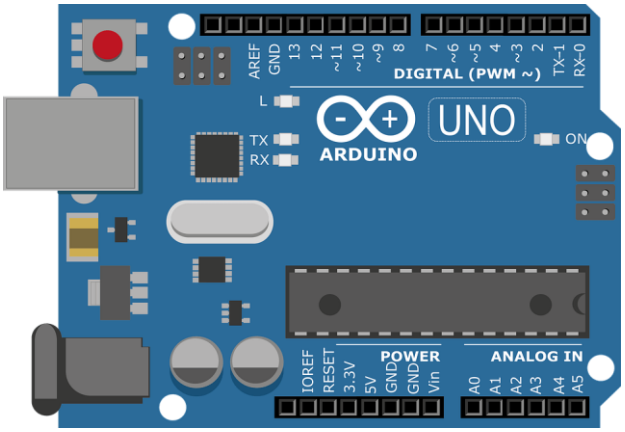
2.2.1 Raspberry pi VS Arduino:

- **Raspberry pi**

  A Raspberry Pi is a general-purpose computer, usually used with a Linux operating system, and the ability to run multiple programs. It is more complicated than Arduino.

- **Arduino**

  An Arduino is a microcontroller motherboard. A microcontroller is a simple computer will run one program at a time, repeatedly. It is very easy to use.


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM ~)
L
TX
RX
ARDUINO
UNO
ON
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

The main purpose of this project is to develop prototype Drowsiness Detection System. Nowadays more accident occurs in trucks and cars than vehicles due to drowsiness. Nearly 97% of crashes of vehicles happen due to drowsiness of driver. It results into loss. for e.g.: human loss, money loss, medical loss.

The accident or crashes not only affect the internal system but also to outside world. 70% injury occurs in internal system and 30% injury happen to the external system. Environmental loss is one of the disadvantages of accident. Accidents results in human as well as non-human loss.

At present time, drowsy driving has become one of the major issues of the traffic collision. According to statistics. Many road accidents occur due to drowsy driving which results in severe injuries and deaths. For this reason, various studies were done in designing systems that can examine the driver fatigue and alert him beforehand, thus preventing him to fall asleep behind the wheel and cause an accident.

Some traditional approaches used vehicle-based measures to design their system, however, such measurements are highly influenced by the structure of the road, type of vehicle and the driving skill. Other approaches used psychological measures for their system that tend to provide better accuracy in monitoring the drowsiness of the driver.

Our proposed system used the eye closure ratio as input parameter to detect the drowsiness of the driver. If the eye closure ratio deteriorates from the standard ratio, the driver

is alerted with the help of a buzzer. For our system, a camera is used to capture the images of the driver's eye and the entire system is incorporated using Raspberry-Pi.

Transportation Safety: - Drowsy and fatigue driving is a major transportation safety concern and is responsible for thousands of accidents and numerous fatalities every year. The resulting harms of drowsy/fatigue driving could be even higher among commercial vehicles. Drowsy driving crashes are usually of high severity due to the drivers' significant loss of control, often leading to unpredicted vehicle trajectory and no braking response. Reliable safety systems are needed to mitigate these crashes. The most important challenge is to detect the driver's condition sufficiently early, prior to the onset of sleep, to avoid collisions.

3.2 Requirement Specification

The Requirement Specification provide complete information about our system called "Drowsiness Detection system".

The Requirements for our system are as follows:

Functional Requirements:

- User should be able to power On/Off the System.
- Device should do minimum computation on its own.
- Device should be able to drive the actuators.
- User should be able to Add New User.
- User should be able to capture video through the mounted camera.
- Device should be able to Turn On/Turn Off Buzzer and LED.
- Device should be able to capture the user yawned.
- Device should be able to alert the user using alarm and LED.

3.3 Planning and Scheduling

	Semester V													
	June		July				August			September			October	
Activities	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
Project Idea Finalization														
Requirements														
Survey of data/ need (Literature Review)														
Feasibility and need validation														
Scope Freezing														
Requirements Detailing														
Use Case Diagrams														
Static User Interface Prototype														
Design														
Database Design/ Block Diagram (ER Diagram, Key Data Structures)														
Other UML Diagrams (Sequence, Activity, Flow Chart etc.)														
Class Diagrams														
Hardware Design - [for embedded/ IoT projects]														
Evaluate Technology options														
Prototype														
Key Technical issue defination														
Build basic Working Prototype														
Planning & Review														
Overall Project Plan														
Weekly Review/ Discussion with Guide														

3.4. Software and Hardware Requirements

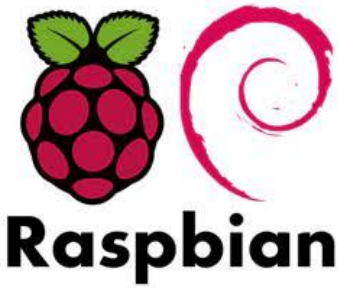
3.4.1 Software Requirement

Python: -

Python is simple, easy to learn syntax efficient and provides readability and therefore reduces the cost of program maintenance. Developing python program requires less time compared to some of the other languages.

Operating System: -

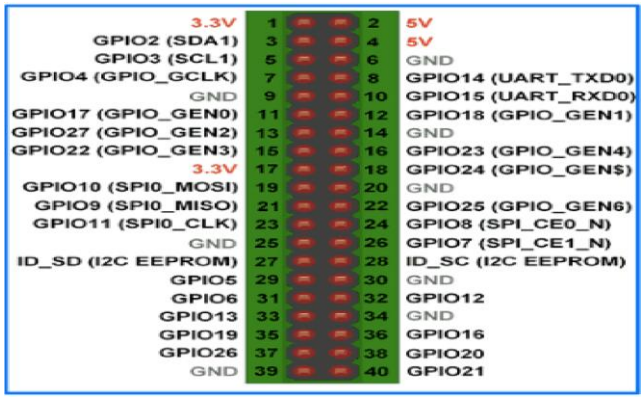
Raspbian OS: -



A free Debian-based OS optimized for Raspberry Pi's hardware, Raspbian comes with all the basic programs and utilities you expect from a general-purpose operating system.

3.4.2 Hardware Requirement

Raspberry Pi 4 -



The **Raspberry Pi 4** Model B is the latest version of the low-cost Raspberry Pi computer. The Pi isn't like your typical device; in its cheapest form it doesn't have a case, and is simply a credit-card sized electronic board -- of the type you might find inside a PC or laptop, but much smaller.

Camera for Raspberry Pi 4 –

- OV5647 5MP 1080P IR-Cut Camera for Raspberry Pi 3/4 with Automatic Day Night Mode Raspberry Pi Camera, supports all revisions of the Pi.
- Embedded IR-CUT filter, eliminating colour distortion in the daylight.
- Comes with infrared LED, supports night vision.
- 5-megapixel OV5647 sensor.
- Can Attach IR LEDs if Night Vision mode is required.

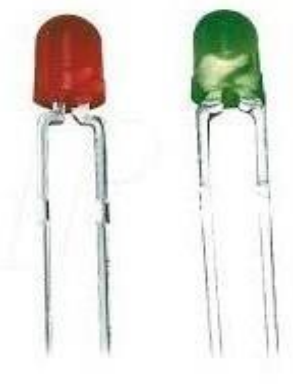
Buzzer –

buzzers are easier to use and allow us to use them on their own, even when you just apply steady DC power. So, we'll use an buzzer. With a continuous DC voltage, it will buzz at a predefined frequency of about 2300Hz. Ideally, the buzzer operates at 5V. As the output voltage of the GPIO pins of our Raspberry Pi are only 3.3V, it looks a little too low for our 5V buzzer. But the buzzer is functioning at 3.3V too. Nevertheless, at 3.3V the volume of the

produced sound is less strong. So, if you have a NPN-transistor, you'll be able to power the buzzer with 5V and it's preferable to use it.

3mm LED'S –

3mm LEDs can be used anywhere and everywhere where you just need low power, high- intensity reliable lighting or indication. They go easily into a breadboard and will add that extra zing to your project.



Jumper Wires –

Jumper wires are wires that simply have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are used with breadboards and other prototyping tools to make it easy to change a circuit as needed.

DC Motor (5V)–

The main principle in controlling a **DC Motor** with **Raspberry Pi** lies with the **Motor** Driver. A **Motor** Driver is a special circuit or IC that provides the necessary power (or rather the current) to the **motor** for smooth and safe operation. Even a small 5V **DC Motor** draws a high initial current of around 300 – 400 mA.

330R/ 330-ohm Resistor Colour Code: -

- Value:330 Ω.
- Type:4 Band Colour Codee System.
- Colour Code: Orange, Orange, Brown, Gold.
- Multiplier: Brown,10.
- Tolerance: Gold Band ±5%.

Display (3.5” touch display): -

This small 3.5-inch touch screen Raspberry Pi Display module is designed especially for Raspberry Pi, using the latest Linux Core system. This is ideal for DIY anywhere, anytime and does not require any separate power source or case to hold it. The module sits right on top of Pi and an ideal alternative solution for HDMI monitors. The screen also comes with a stylus to interact with the small screen.

- Supports all revision of Raspberry Pi (directly pluggable models).
- Works with Raspbian/Ubuntu directly.
- Comes with a full set of screws and nuts for assembly.
- 320×480 resolution, better display.
- Lightweight and easy to install.

3.5 Preliminary Product Description:

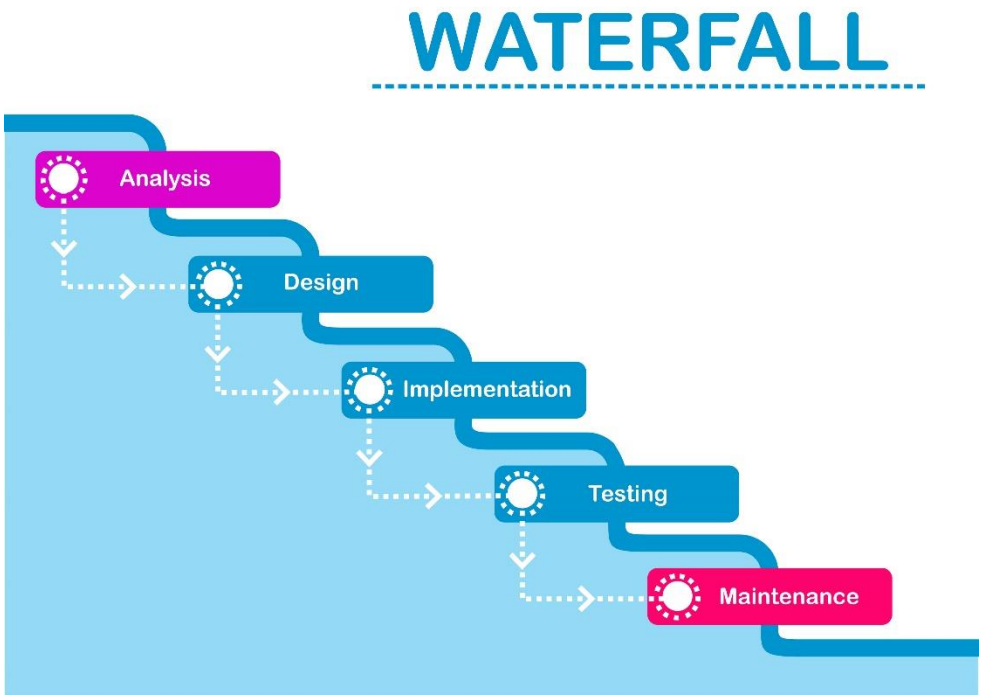
The human eye in a normal physical state has a different size, so that when a person who has large eyes when drowsy then the large opening of his eyes will be different from the size of the eyes of someone who has small eye size. Therefor in this project the system was designed by taking a unique eye reference per person. Preliminary data retrieval is taken when the driver starts driving a vehicle assuming the initial condition of the driver is not drowsy. This initial data will be used as a reference for threshold decision making in the process of determining drowsiness in subsequent experiments. In this project using the EAR value of each driver from the initialization process, and the EAR value is also used as a baseline for stating a driver's eyes are closed or open based on the EAR threshold. If the EAR value (from driving process) is below the threshold for several consecutive times (with a reference of 1.5 seconds) the driver will be detected drowsy by the system. If detected drowsiness, it will be given a response in the form of alarm sound. The drowsiness detection system has been designed using the proposed EAR threshold method from each ratio of eyes each individual driver. From this research get the EAR driver is 0.03-0.69. EAR min for each driver is in the range: 0.03-0.18 and EAR max for each driver in the range: 0.35-0.69. The minimum value of the EAR threshold is 0.20 and the maximum value of the EAR threshold is 0.41. From the four times the drowsiness detection system testing, the system successfully to detect drowsiness from each driver.

3.6 Conceptual Model

Software process model deals with the model which we are going to use for the development of the project. There are many software processes models available but while choosing it we should choose it according to the project size that is whether it is industry scale project or big scale project or medium scale project. Accordingly, the model which we choose should be suitable for the project as the software process model changes the cost of the project also changes because the steps in each software process model varies. This

software is build using the waterfall mode. This model suggests work cascading from step to step like a series of waterfalls. It consists of the following steps in the following manner.

WATERFALL MODEL:



Analysis Phase:

A problem is attacked by breaking it into modules. The purpose of the analysis is to determine what needs to be done to solve the problem. Typically, the logical elements of the system (its limitations, processes, and data) are defined during analysis.

Design Phase:

The purpose of the design is to determine how the problem will be resolved. During design the analyst's focus shifts from logical to physical.

Coding Phase:

The system is built during this phase. Programs are coded, debugged, documented, and tested. New hardware is selected and ordered. Procedures are written and tested.

Test Phase:

Once the system is developed, it is tested to ensure that it does what it was designed to do. After the system passes its final test and any remaining problems are corrected, the system is implemented and released to use.

CHAPTER 4: SYSTEM DESIGN

4.1 Basic Modules:

- Display.
- Face Detection.
- Drowsiness Detection.
- Distraction Detection.
- Alert Meter.
- Buzzer.
- LED.
- DC Motor.

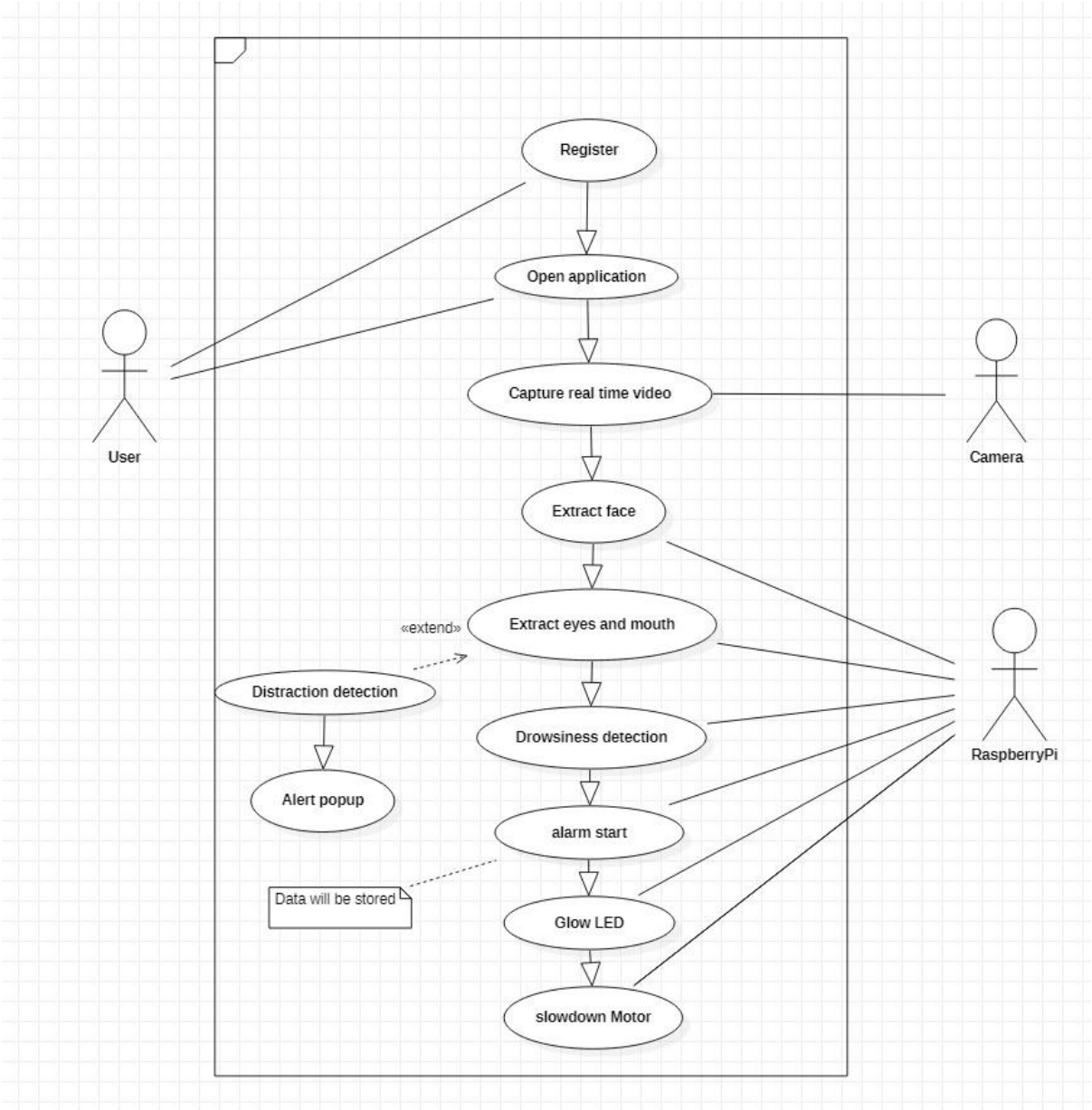
4.2 Data Design:

Hardware component which is fixed device basically consists of three main units: Raspberry Pi, camera and display. Display is connected to raspberry pi via GPIO. Camera is connected to raspberry pi via CSI port through 15 pin ribbon cable. Video input taken by camera is sent to the raspberry Pi for detecting drowsiness, Then RaspberryPi sends instructions to buzzer, LED and motor.

4.3 Diagrams

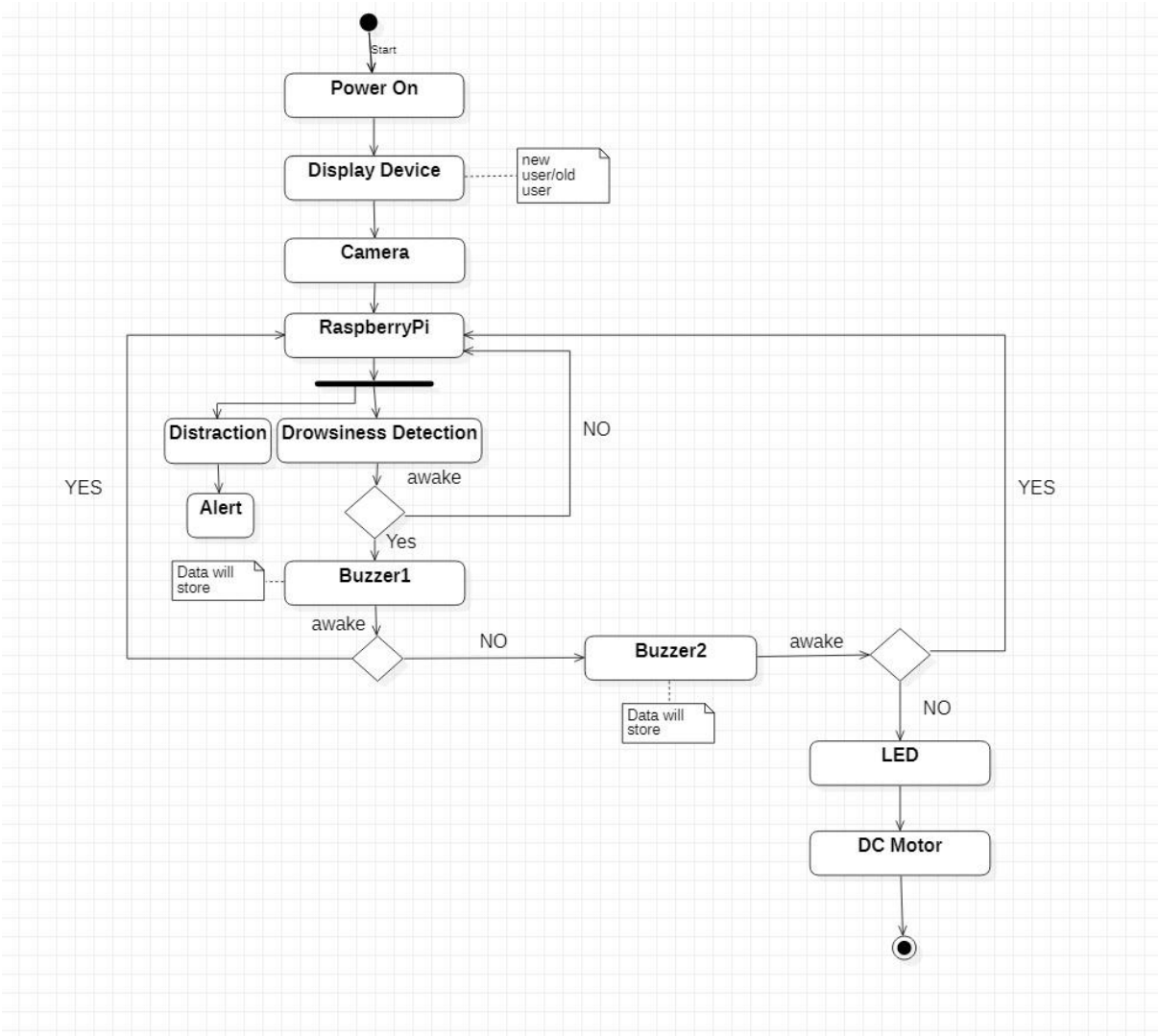
4.3.1 Use Case Diagram:

This use case diagram shows the working of each and every components and modules of our project.



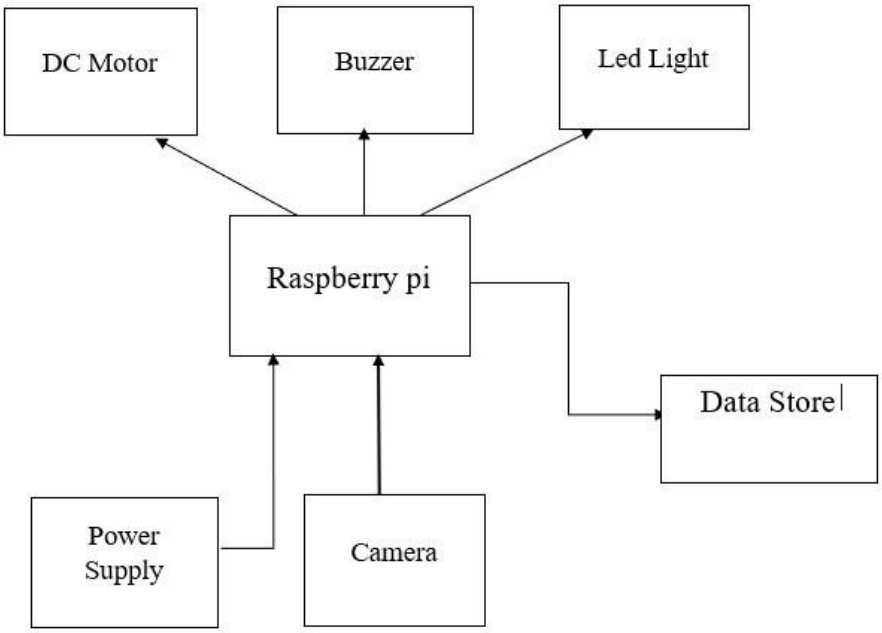
4.3.2 Activity Diagram:

This activity diagram shows the process of working in a very simple way. It is a stepwise activity of workflows of this project.



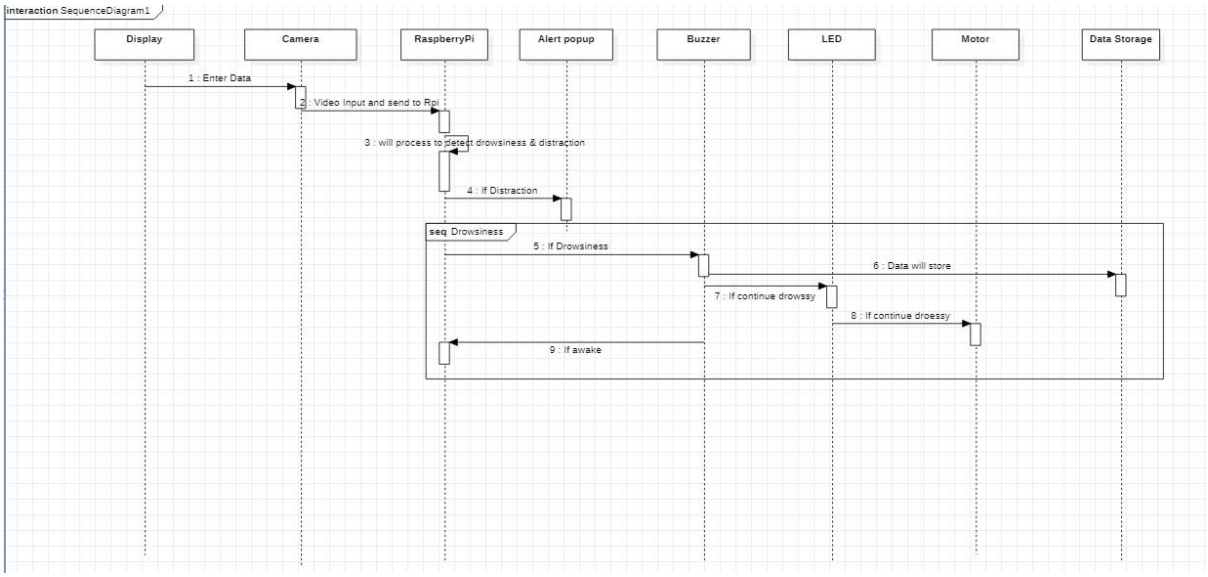
4.3.3 Block Diagram:

This block diagram shows, how raspberry pi is connecting to all the components. Figure 1 shows that raspberry pi is the hub of our project.



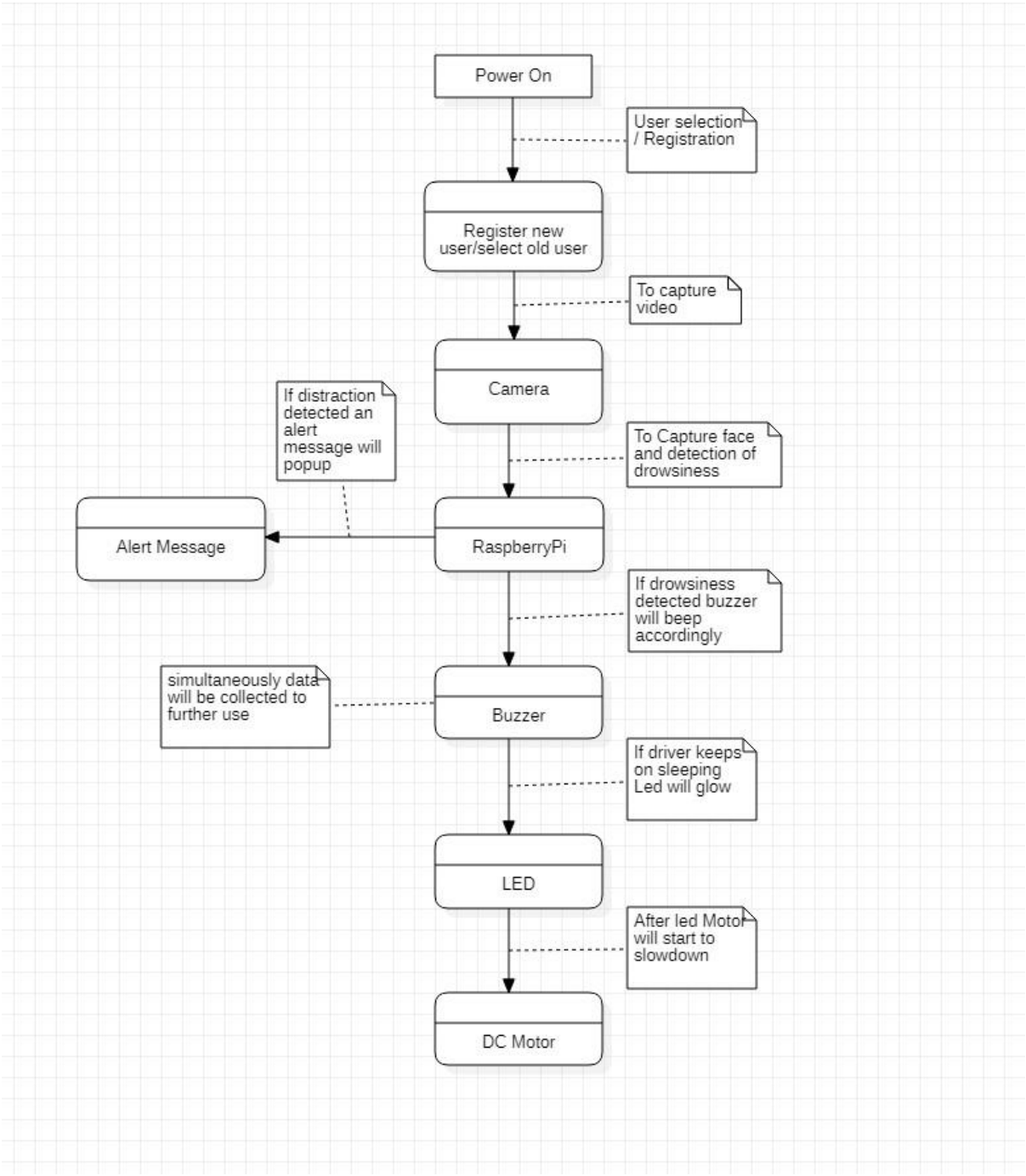
4.3.4 Sequence Diagram:

This sequence diagram is a type of interaction diagram which shows how and in what manner/order all the things are keep interacting with each other.



4.3.5 Data Flow Diagram:

A data-flow diagram is a way of representing a flow of data through a process or a system.



4.3.6 Circuit Diagram:

It is a graphical representation of electrical circuit diagram that shows interconnections between the electrical circuit.

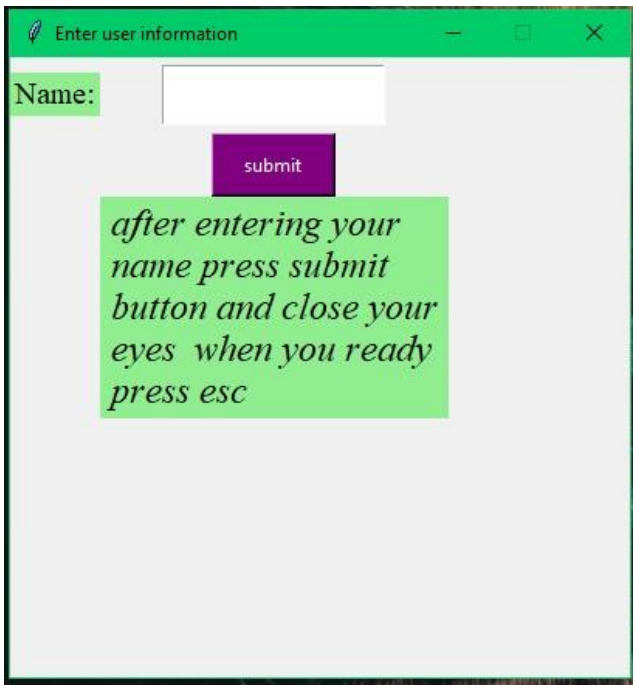
4.3.7 Event Table:

Event	Trigger	Source	Use Case	Response	Destination
User Input	Display	User	New User/Old User	Data Input	User
Watching Face	Camera	User	Taking Video	Detects face	User
Detecting Drowsiness	Raspberry Pi	User	Detection	Detects drowsiness and work accordingly	User
Distraction Detection	Raspberry Pi	User	Detection	Detects distraction and work accordingly	User
Yawning	Raspberry Pi	User	Detection And Alert	Detects yawning and efficiency	User
Beep	Raspberry Pi	Alarm	Alert	Driver Alert	User
LED glow	Raspberry Pi	LED	Alert	Cars that are behind will alert	Other Vehical Drivers
Speed Down	Raspberry Pi	Dc Motor	Slow Down Car	Car will Slow Down	User

4.3.8 User Interface Design:

Start Page :

New User Page:



Old User Page:

4.3.9 Test Cases Design:

Level of Testing

To uncover the errors, present in different phases we have the concept of levels of testing. The basic levels of testing are:

- Client Needs Acceptance Testing.
- Requirements System Testing.
- Design integration Testing.
- Code unit Testing.

A series of Testing is done for the proposed system before the system is ready for the user acceptance testing.

The steps involved in Testing are:

1. **Unit Testing:**

Unit Testing focuses verification efforts on the smallest unit of the software design, the module. This is also known as "Module Testing". The modules are tested separately. This

Testing carried out during the programming stage itself. In this Testing, each module found to be working satisfactorily as regards to the expected output from the module.

2. Integration Testing:

Data can be grossed across an interface; one module can have adverse efforts on another. Integration Testing's systematic Testing for construction the program structure while at the same time conducting tests to uncover errors associated within the interface. The objectives to take unit tested modules and build a program structure. All the modules are combined and tested. Here corrections difficult because of the isolation of causes complicate by the vast expense of the entire program.

3. System Testing:

After integration, the whole program will again be tested. Case studies will again be applied with integrated software.

4. The methodology used for testing:

The testing methodology implemented for testing of the project is Black Box testing. In the black box testing, the internal logic of the system under testing is not considered and the test cases are decided to form the specification or the requirements. It is often called functional testing. It aims to test functionality according to the requirements. Equivalence class portioning, boundary value analysis, and because effecting graphing are examples of methods for selecting test cases for black box testing. State-based testing is another approach in which the system is modelled as a state machine and then this model is used to select test cases using some transaction or path-based coverage criteria, state-based testing can also be viewed as grey-box testing in that it often requires more information than just the requirements. The testing methodology implemented for testing of the project is the black box testing. Refer to the above paragraph for explanation.

5. **Test Cases:**

The forms were tested for their functionality and error messages displayed wherever the input does not meet the required requirements. For the project, we need testing to make it successful. If each component works properly in all respect and gives desired output for all kind of inputs, then projects said to be successful.

Each hardware parts will be tested for their functionality according to the test cases given below:

Modules	**Pre-Condition**	**Test Data**	**Priority**
Display	The Module should be on.	The old user will select the name from the list and new user will enter new data and start the process.	High
Face Detection	The camera should be connected with Raspberry Pi.	It will detect the face in the video.	High
Drowsiness Detection	Face should be detected.	It will detect drowsiness.	High
Distraction Detection	Face should be detected.	It will detect distraction (if user is not concentrated on the road).	High

Alert Meter	Display should be on.	It will Show the efficiency of the user.	High
Buzzer	Drowsiness should be detected.	Alert the user.	High
LED	Drowsiness should be detected.	Alert the driver behind the car.	High
DC Motor	LED should be on.	Car will slow down.	High