DRIVER DROWSINESS DETECTION

A MINOR PROJECT REPORT

Submitted by

Malik Aurangzaib[RA2111003011070]
Saurabh Mittal [RA2111003011066]
Priyanshu Naskar[RA2111003011065]

Under the guidance of
Ms. HEMA M
(Asst Professor)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

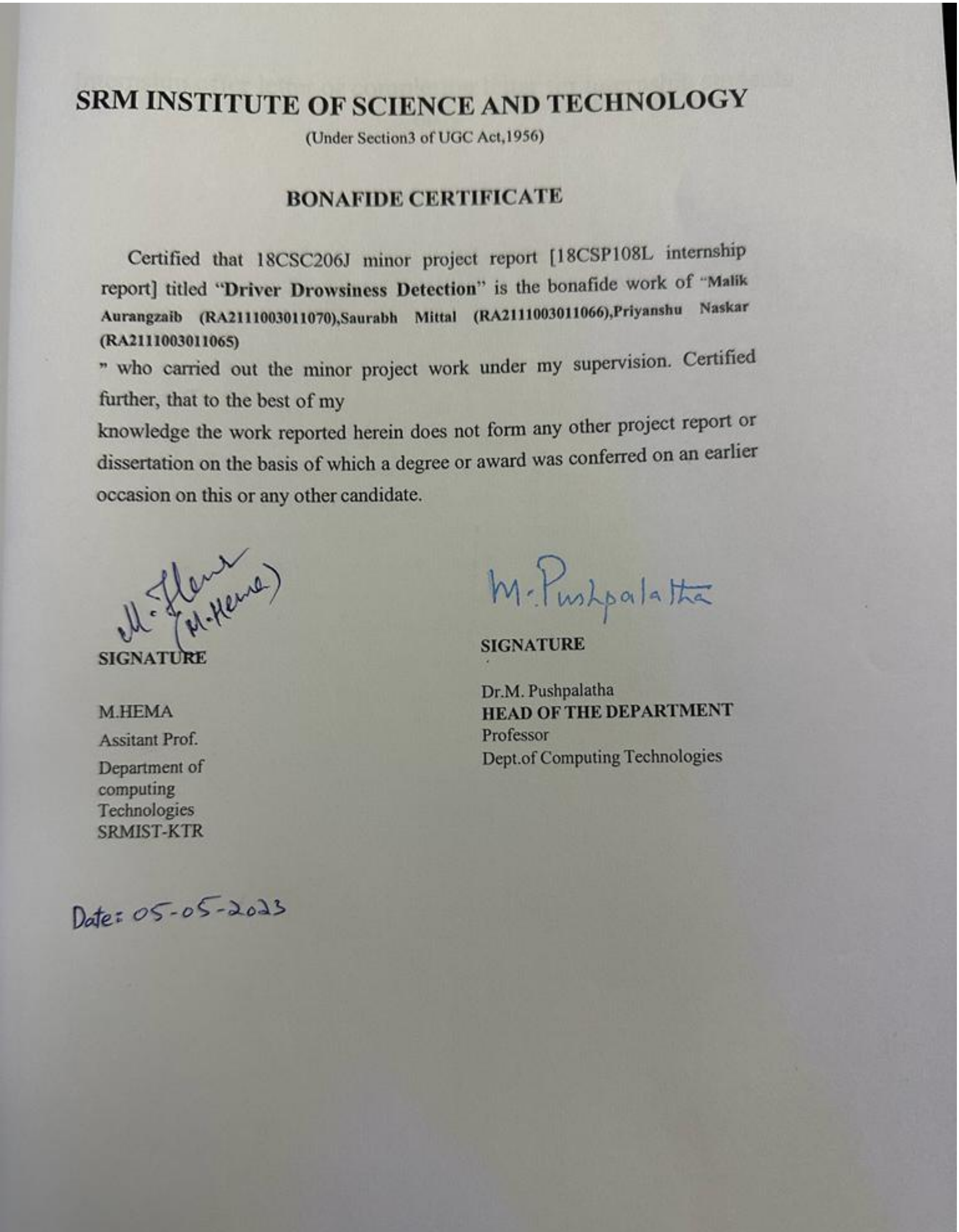
COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M.Nagar, Kattankulathur, Chengalpattu District

MAY 2023



SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section3 of UGC Act,1956)

BONAFIDE CERTIFICATE

Certified that 18CSC206J minor project report [18CSP108L internship report] titled "**Driver Drowsiness Detection**" is the bonafide work of "**Malik Aurangzaib (RA2111003011070),Saurabh Mittal (RA2111003011066),Priyanshu Naskar (RA2111003011065)**

" who carried out the minor project work under my supervision. Certified further, that to the best of my

knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

M. Hema (M.Hema)

SIGNATURE

M.HEMA
Assitant Prof.
Department of
computing
Technologies
SRMIST-KTR

M. Pushpalatha

SIGNATURE

Dr.M. Pushpalatha
HEAD OF THE DEPARTMENT
Professor
Dept.of Computing Technologies

Date: 05-05-2023

Internship offer letter or completion letter for internship students only

ABSTRACT

Drowsy Driver Detection System has been developed using a non-intrusive machine vision based concepts. The system uses a small monochrome security camera that points directly towards the driver’s face and monitors the driver’s eyes in order to detect fatigue. In such a case when fatigue is detected, a warning signal is issued to alert the driver. This report describes how to find the eyes, and also how to determine if the eyes are open or closed. The algorithm developed is unique to any currently published papers, which was a primary objective of the project. The system deals with using information obtained for the binary version of the image to find the edges of the face, which narrows the area of where the eyes may exist. Once the face area is found, the eyes are found by computing the horizontal averages in the area. Taking into account the knowledge that eye regions in the face present great intensity changes, the eyes are located by finding the significant intensity changes in the face. Once the eyes are located, measuring the distances between the intensity changes in the eye area determine whether the eyes are open or closed. A large distance corresponds to eye closure. If the eyes are found closed for 5 consecutive frames, the system draws the conclusion that the driver is falling asleep and issues a warning signal. The system is also able to detect when the eyes cannot be found, and works under reasonable lighting conditions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AES	Advanced Encryption Standard
ANN	Artificial Neural Network
CSS	Cascading Style Sheet
CV	Computer Vision
DB	Data Base
SQL	Structured Query Language
SVM	Support Vector Machine
UI	User Interface

CHAPTER1

INTRODUCTION

1.1 PURPOSE

Humans have always invented machines and devised techniques to ease and protect their lives, for mundane activities like traveling to work, or for more interesting purposes like aircraft travel. With the advancement in technology, modes of transportation kept on advancing and our dependency on it started increasing exponentially. It has greatly affected our lives as we know it. Now, we can travel to places at a pace that even our grandparents wouldn't have thought possible. In modern times, almost everyone in this world uses some sort of transportation every day. Some people are rich enough to have their own vehicles while others use public transportation. However, there are some rules and codes of conduct for those who drive irrespective of their social status. One of them is staying alert and active while driving.

Neglecting our duties towards safer travel has enabled hundreds of thousands of tragedies to get associated with this wonderful invention every year. It may seem like a trivial thing to most folks but following rules and regulations on the road is of utmost importance. While on road, an automobile wields the most power and in irresponsible hands, it can be destructive and sometimes, that carelessness can harm lives even of the people on the road. One kind of carelessness is not admitting when we are too tired to drive. In order to monitor and prevent a destructive outcome from such negligence, many researchers have written research papers on driver drowsiness detection systems. But at times, some of the points and observations made by the system are not accurate enough. Hence, to provide data and another perspective on the problem at hand, in order to improve their implementations and to further optimise the solution, this project has been done.

1.1 FACTS & STATISTICS

Our current statistics reveal that just in 2015 in India alone, 148,707 people died due to car related accidents. Of these, at least 21 percent were caused due to fatigue causing drivers to make mistakes. This can be a relatively smaller number still, as among the multiple causes that can lead to an accident, the involvement of fatigue as a cause is generally grossly underestimated. Fatigue combined with bad infrastructure in developing countries like India is a recipe for disaster. Fatigue, in general, is very difficult to measure or observe unlike alcohol and drugs, which have clear key indicators and tests that are available easily. Probably, the best solutions to this problem are awareness about fatigue-related accidents and promoting drivers to admit fatigue when needed. The former is hard and much more expensive to achieve, and the latter is not possible without the former as driving for long hours is very lucrative. When there is an increased need fora job, the wages associated with it increases leading to more and more people adopting it.
Such is the case for driving transport vehicles at night. Money motivates drivers to make unwise decisions like driving all night even with fatigue. This is mainly because the drivers are not themselves aware of the huge risk associated with driving when fatigued. Some countries have imposed restrictions on the number of hours a driver can drive at a stretch, but it is still not enough to solve this problem as its implementation is very difficult and costly.

1.3 Software Requirements Specification

3.1 Python:

- Python 3

3.2 Libraries

- Numpy
- Scipy
- Playsound
- Dlib
- Imutils
- opencv, etc.

3.3 Operating System

- Windows or Ubuntu

2 LITERATURE SURVEY

2.1 SYSTEM REVIEW

This survey is done to comprehend the need and prerequisite of the general population, and to do as such, we went through different sites and applications and looked for the fundamental data. Based on these data, we made an audit that helped us get new thoughts and make different arrangements for our task. We reached the decision that there is a need of such application and felt that there is a decent extent of progress in this field too.

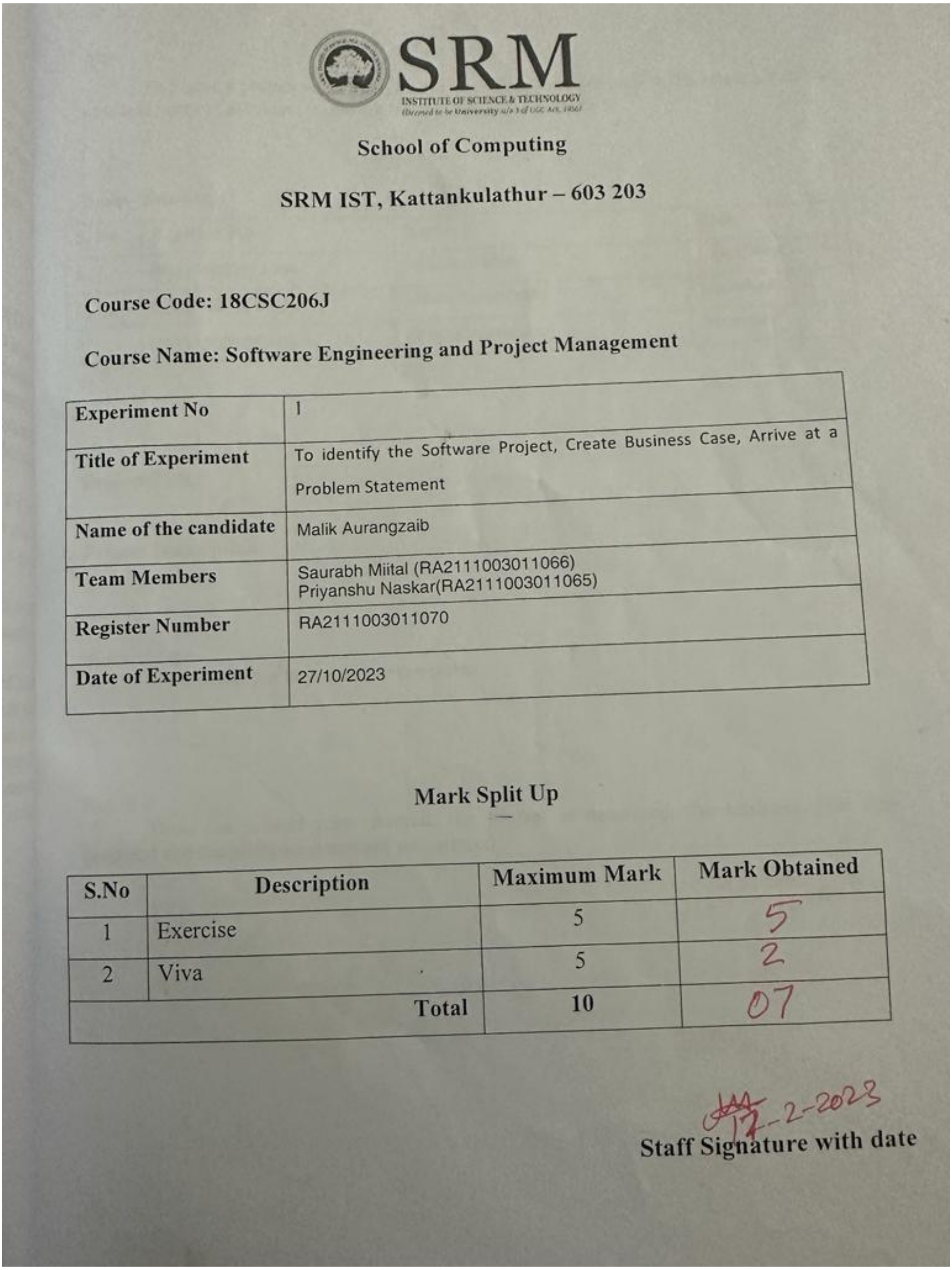
2.2 TECHNOLOGY USED

a. PYTHON - Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed AND supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

b. JUPYTER Lab - Project Jupyter is a nonprofit organization created to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

c. IMAGE PROCESSING - In computer science, digital image processing is the use of computer algorithms to perform image processing on digital images.

d. MACHINE LEARNING - Machine learning is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build
a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly told.



SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	1
Title of Experiment	To identify the Software Project, Create Business Case, Arrive at a Problem Statement
Name of the candidate	Malik Aurangzaib
Team Members	Saurabh Miital (RA2111003011066) Priyanshu Naskar(RA2111003011065)
Register Number	RA2111003011070
Date of Experiment	27/10/2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	5
2	Viva	5	2
	Total	**10**	07

17-2-2023

Staff Signature with date

Aim

To Frame a project team, analyze and identify a Software project. To create a business case and Arrive at a Problem Statement for the <title of the project>

Team Members:

S. No	Register No	Name	Role
1	RA2111003011066	Saurabh Mittal	**Lead/Rep**
2	RA2111003011070	Malik Aurangzaib	**Member**
3	RA2111003011065	Priyanshu Naskar	**Member**

Project Title:

Project Description

Business Case

<Incorporate the Business Case template>

Result

Thus, the project team formed, the project is described, the business case was prepared and the problem statement was arrived.

ONE PAGE BUSINESS CASE TEMPLATE

DATE	**28/10/2023**
SUBMITTED BY	**Saurabh Mittal (RA2111003011066)** **Malik Aurangzaib (RA2111003011070)** **Priyanshu Naskar(RA2111003011065)**
TITLE / ROLE	**Driver Drowsiness Detection System using Machine Learning**

THE PROJECT

1. In this project by monitoring Visual Behaviour of a driver with webcam and machine learning SVM (support vector machine) algorithm we are detecting Drowsiness in a driver.

2. The purpose of the drowsiness detection system is to aid in the prevention of accidents passenger and commercial vehicles.

THE HISTORY

Existing drowsiness detection methods include:
1. Carnegie-Mellon Research Institute:
PERCLOS SystemsPERCLOS (percentage closure) is defined as the measurement of the percentage of time the pupils of the eyes are 80%
or more occluded over a specified time interval. It has been found that PERCLOS is a reliable measure in detecting drowsiness.

2.Head position metrics:
Systems have been devised such that the head position of the driver is detected and when the head leaves the headrest past a certainthreshold percentage, the system alerts the driver.

LIMITATIONS

1. Space: The solution needs to be implemented in a space-efficient manner. It must not interfere with the existing controls of the car.

2. Power: There will be a limited power source so the solution needs to designed so that it can operate properly on limited power requirements.

APPROACH

1. Python: Python is the basis of the program that we wrote. It utilizes many of the python libraries.
2. Libraries:
- Numpy: Pre-requisite for Dlib
- Scipy: Used for calculating Euclidean distance between the eyelids.
- Playsound: Used for sounding the alarm
- Dlib: This program is used to find the frontal human face and estimate its pose using 68 face landmarks.
- Imutils: Convenient functions written for Opencv.
- Opencv: Used to get the video stream from the webcam, etc.

3. OS: Program is tested on Windows 10 build 1903 and PopOS 19.04
4. Laptop: Used to run our code.
5. Webcam: Used to get the video feed.

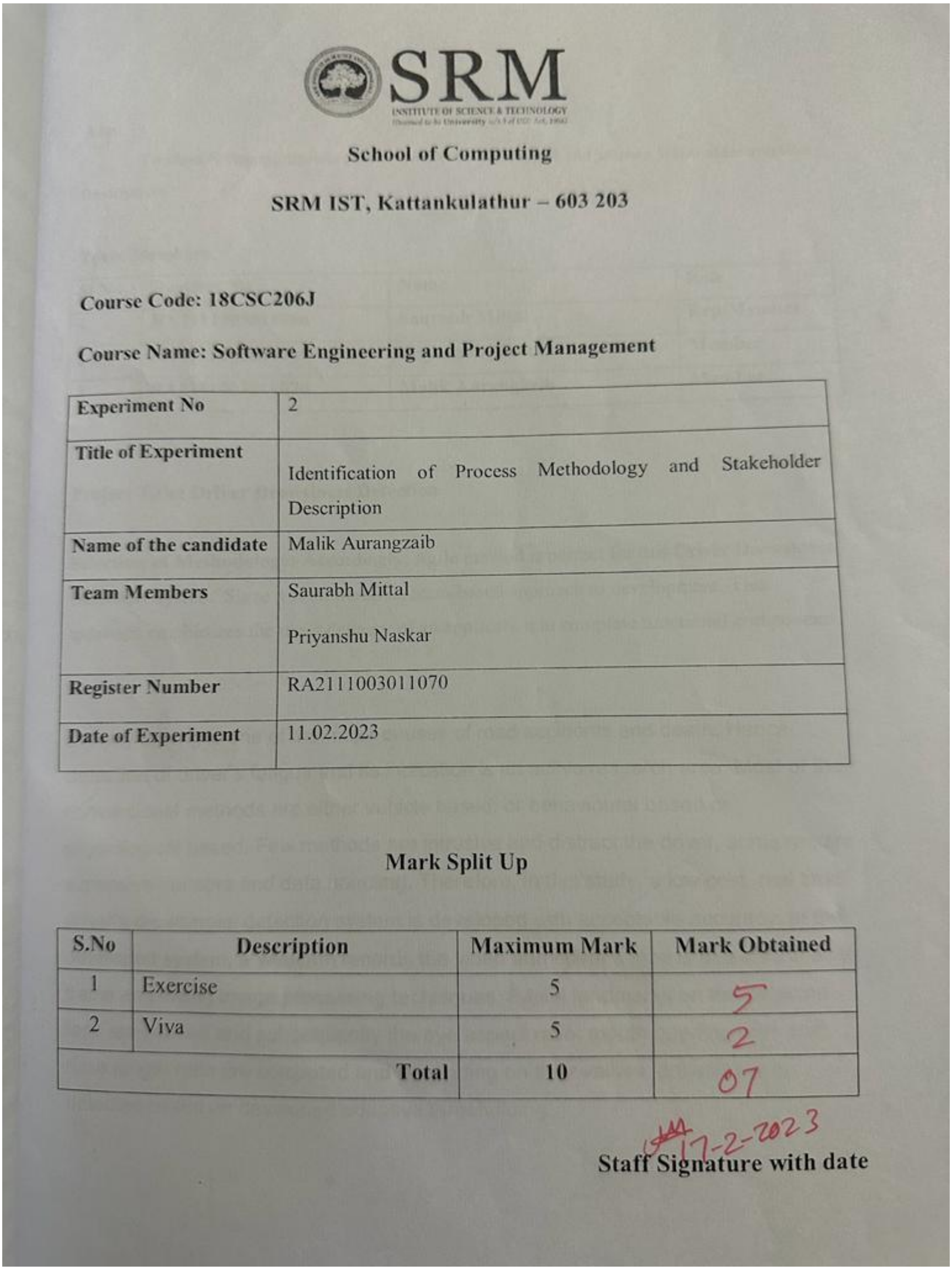
BENEFITS

1. Low Cost: As no expensive items or products are being used so it can available at a very low cost or even Free.

2. Easy to Use: Component establishes interface with other drivers very easily.

3. Safety: Life of the driver can be saved by alerting him using the alarm system.

4. Real-Time: As it provides the result in real time that will help the user to make fastrer decisions.
.
5. Practically applicable: 4. This method is practically applicable.



SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	2
Title of Experiment	Identification of Process Methodology and Stakeholder Description
Name of the candidate	Malik Aurangzaib
Team Members	Saurabh Mittal Priyanshu Naskar
Register Number	RA2111003011070
Date of Experiment	11.02.2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	5
2	Viva	5	2
	Total	**10**	07

17-2-2023

Staff Signature with date

Aim

To identify the appropriate Process Model for the project and prepare Stakeholder and User Description.

Team Members:

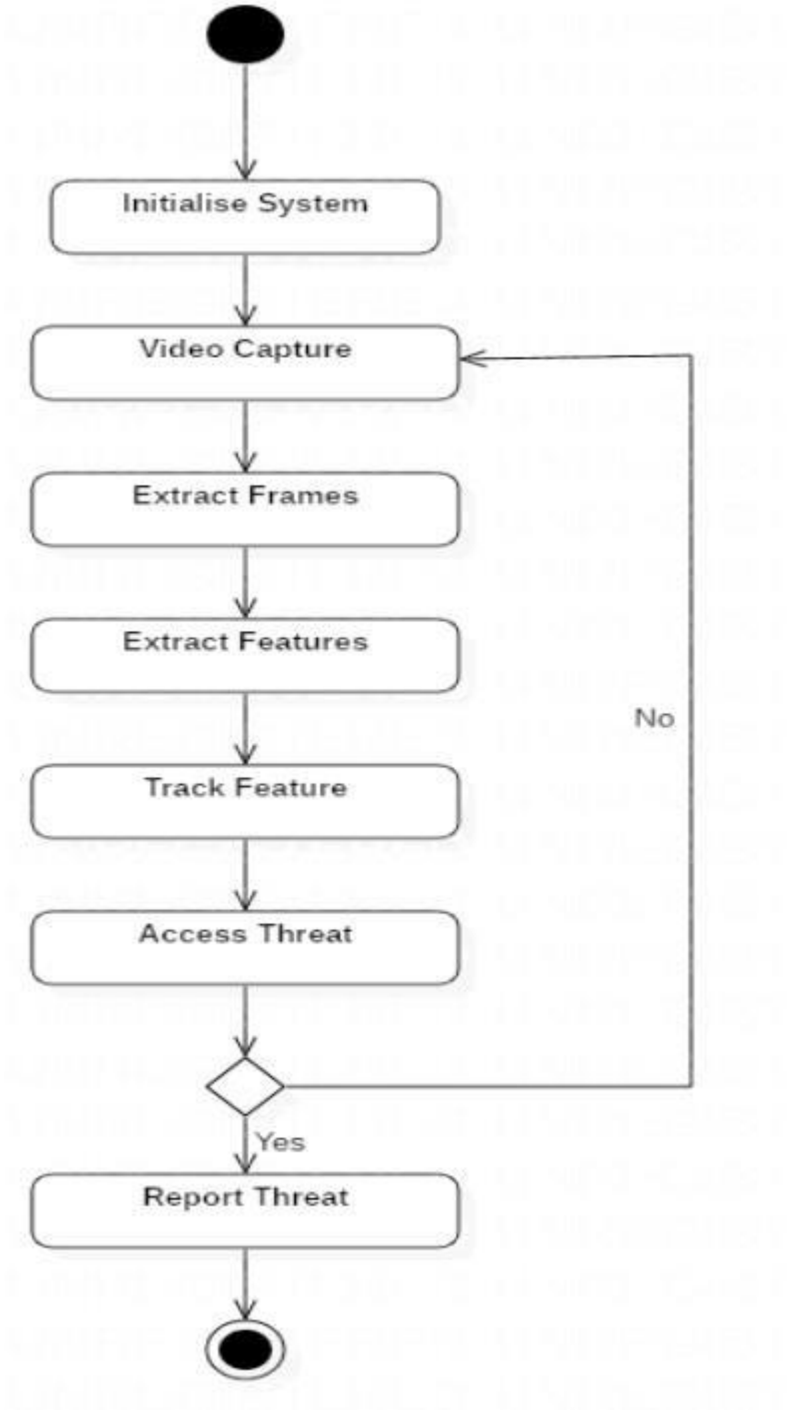
Sl No	Register No	Name	Role
1	RA2111003011066	Saurabh Mittal	Rep/Member
2	RA2111003011065	Priyanshu Naskar	Member
3	RA2111003011070	Malik Aurangzaib	Member

Project Title: Driver Drowsiness Detection

Selection of Methodology: Accordingly, Agile method is perfect for this Driver Drowsiness Detection System. Since it is an iterative, team-based approach to development. This approach emphasizes the rapid delivery of an application in complete functional components.

Drowsy driving is one of the major causes of road accidents and death. Hence, detection of driver's fatigue and its indication is an active research area. Most of the conventional methods are either vehicle based, or behavioural based or physiological based. Few methods are intrusive and distract the driver, some require expensive sensors and data handling. Therefore, in this study, a low cost, real time driver's drowsiness detection system is developed with acceptable accuracy. In the developed system, a webcam records the video and driver's face is detected in each frame employing image processing techniques. Facial landmarks on the detected face are pointed and subsequently the eye aspect ratio, mouth opening ratio and nose length ratio are computed and depending on their values, drowsiness is detected based on developed adaptive thresholding.

Case Diagram

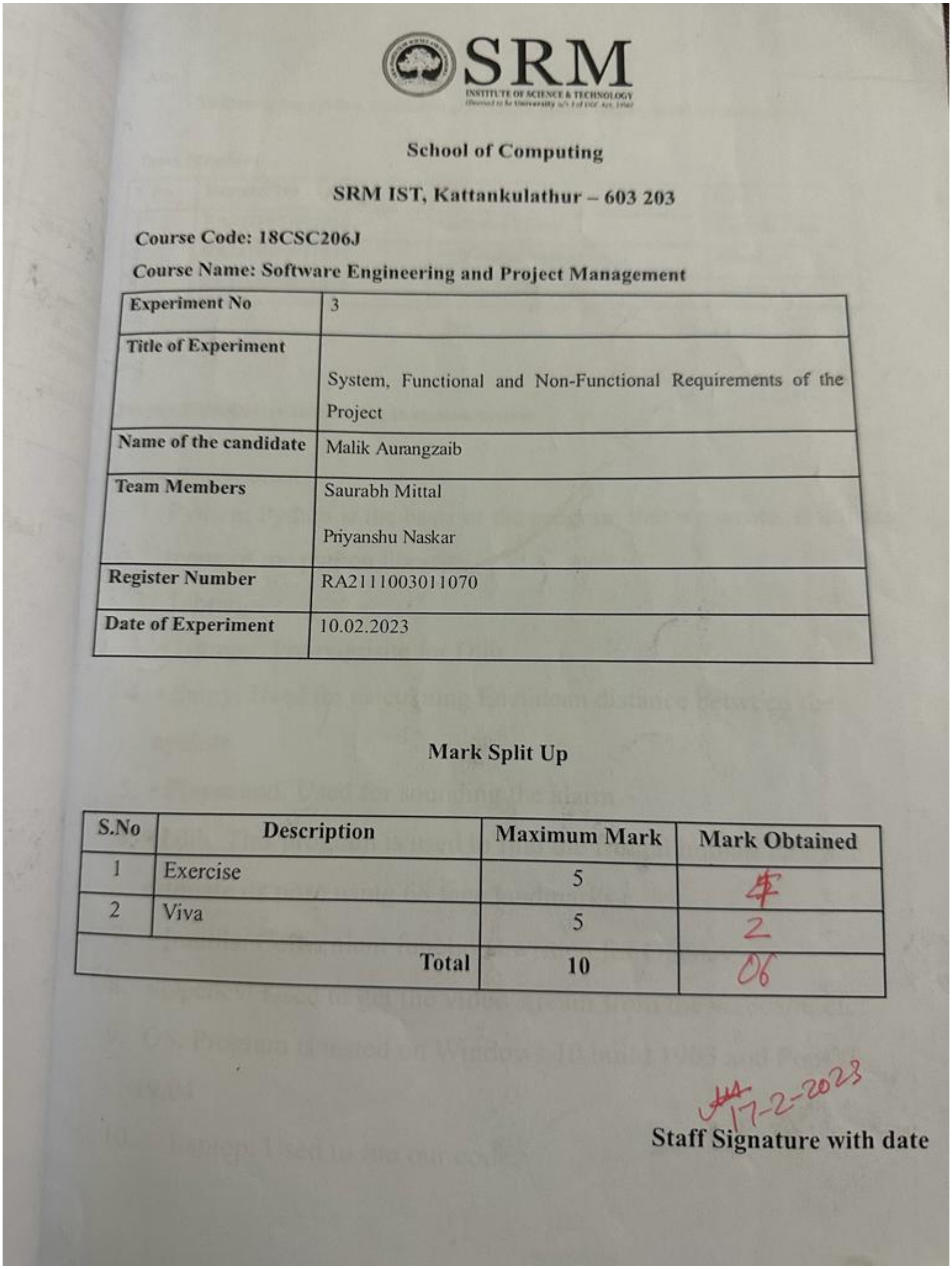


Incorporate information to below table regarding stakeholders of the project [Make use of below examples]

Stakeholder Name	Activity/ Area /Phase	Interest	Influence	Priority (High/ Medium/ Low)
Owner	Ensure product & its functionability.	High	High	High
Leader	Guides the team accordingly & is a responsible mentor of the project & team	High	High	High
Sponsorship	Supports the project financially	High	High	High
Team Members	Main working heads behind the project	High	High	High
Resource Managers	Assigns tasks to team members according to their skills	High	Low	Medium
End Users	Main critic of the product & help us with their feedback	Low	High	Medium

Result:

Thus the Project Methodology was identified and the stakeholders were described.



SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	3
Title of Experiment	System, Functional and Non-Functional Requirements of the Project
Name of the candidate	Malik Aurangzaib
Team Members	Saurabh Mittal Priyanshu Naskar
Register Number	RA2111003011070
Date of Experiment	10.02.2023

Mark Split Up

S.No	**Description**	**Maximum Mark**	**Mark Obtained**
1	Exercise	5	4
2	Viva	5	2
	Total	**10**	06

17-2-2023

Staff Signature with date

Aim

To identify the system, functional and non-functional requirements for the project.

Team Members:

S No	Register No	Name	Role
1	RA2111003011066	Saurabh Mittal	Rep/Member
2	RA2111003011065	Priyanshu Naskar	Member
3	RA2111003011070	Malik Aurangzeb	Member

Project Title: Driver Drowsiness Detection System

System Requirements:

1. Python: Python is the basis of the program that we wrote. It utilizes many of the python libraries.
2. Libraries:
3. • Numpy: Pre-requisite for Dlib
4. • Scipy: Used for calculating Euclidean distance between the eyelids.
5. • Playsound: Used for sounding the alarm
6. • Dlib: This program is used to find the frontal human face and estimate its pose using 68 face landmarks.
7. • Imutils: Convenient functions written for Opencv.
8. • Opencv: Used to get the video stream from the webcam, etc.
9. OS: Program is tested on Windows 10 build 1903 and PopOS 19.04
10. Laptop: Used to run our code.

11. Webcam: Used to get the video feed

Functional Requirements:

- Lighting conditions. Frequent and drastic change in darkness or brightness of a scene (or part of it), which may happen even during the shortest driving intervals, have been proven to be a significant challenge for many computer vision algorithms.
- Camera motion. Poor road conditions as well as a more aggressive style of driving can introduce significant amount of vibrations and discomfort to the driving experience. Those vibrations can be passed onto the camera and cause distortion in the images which can significantly skew the results and decrease the overall performance of the system.
- Relative positioning of device. The camera must be positioned within a certain range from the driver and within a certain viewing angle. Every computer vision algorithm has a “comfort zone” in which it performs the best and most reliably. If that comfort zone is left, performance can be dropped significantl

Non-Functional Requirements

Scalability assesses the highest workloads under which the system will still meet the performance requirements. There are two ways to enable your system scale as the workloads get higher: horizontal and vertical scaling.

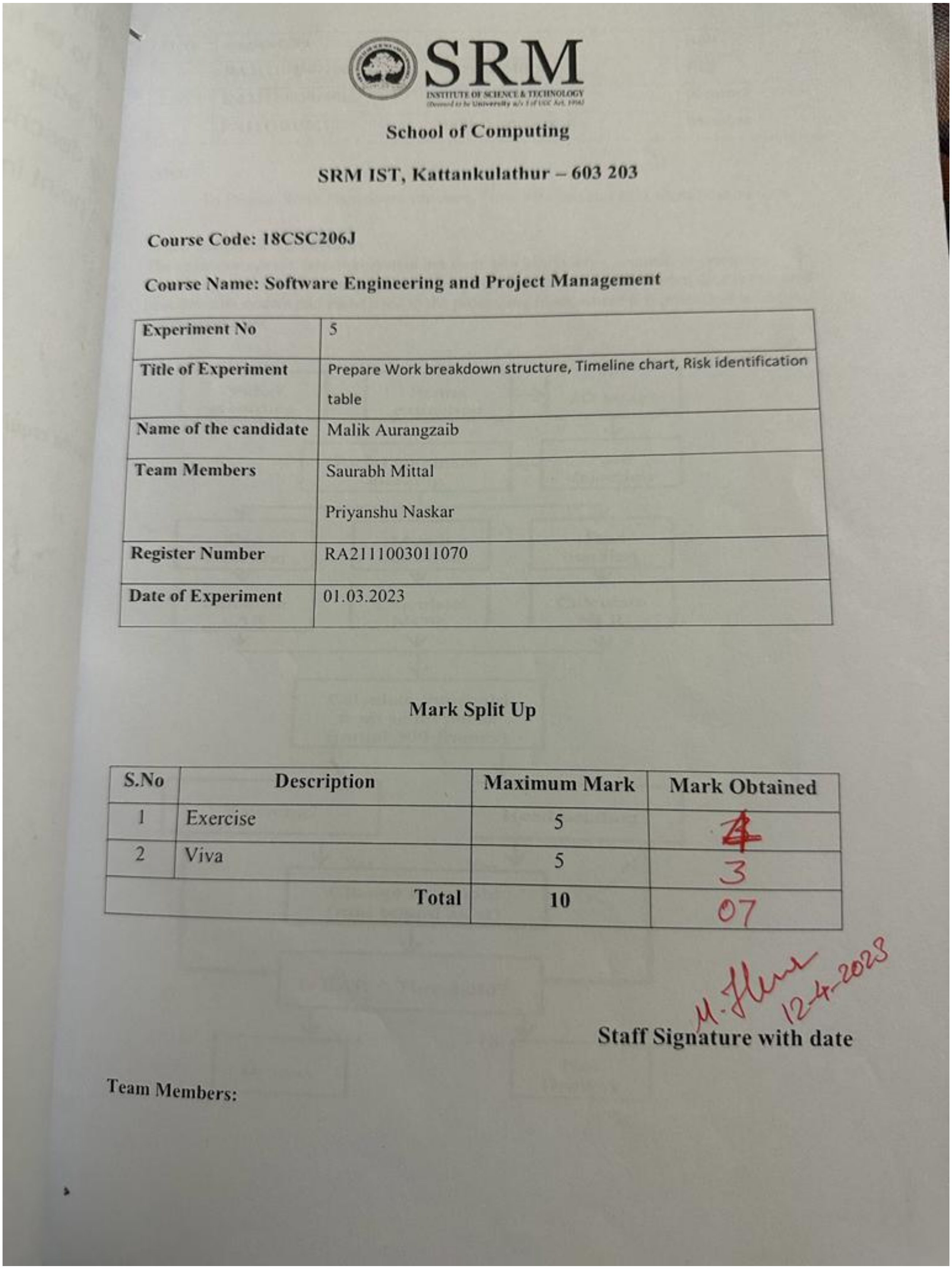
Portability determines how a system or its element can be launched within one environment or another.

Compatibility, as an additional aspect of portability, defines how a system can coexist with another system in the same environment. Maintainability defines the time required for a solution or its component to be fixed, changed to increase performance or other qualities, or adapted to a changing environment.

Availability describes how likely the system is accessible to a user at a given point in time

Result

Thus the requirements were identified and accordingly described.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	5
Title of Experiment	Prepare Work breakdown structure, Timeline chart, Risk identification table
Name of the candidate	Malik Aurangzaib
Team Members	Saurabh Mittal Priyanshu Naskar
Register Number	RA2111003011070
Date of Experiment	01.03.2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	4
2	Viva	5	3
	Total	**10**	07

12-4-2023

Staff Signature with date

Team Members:

Team Members:

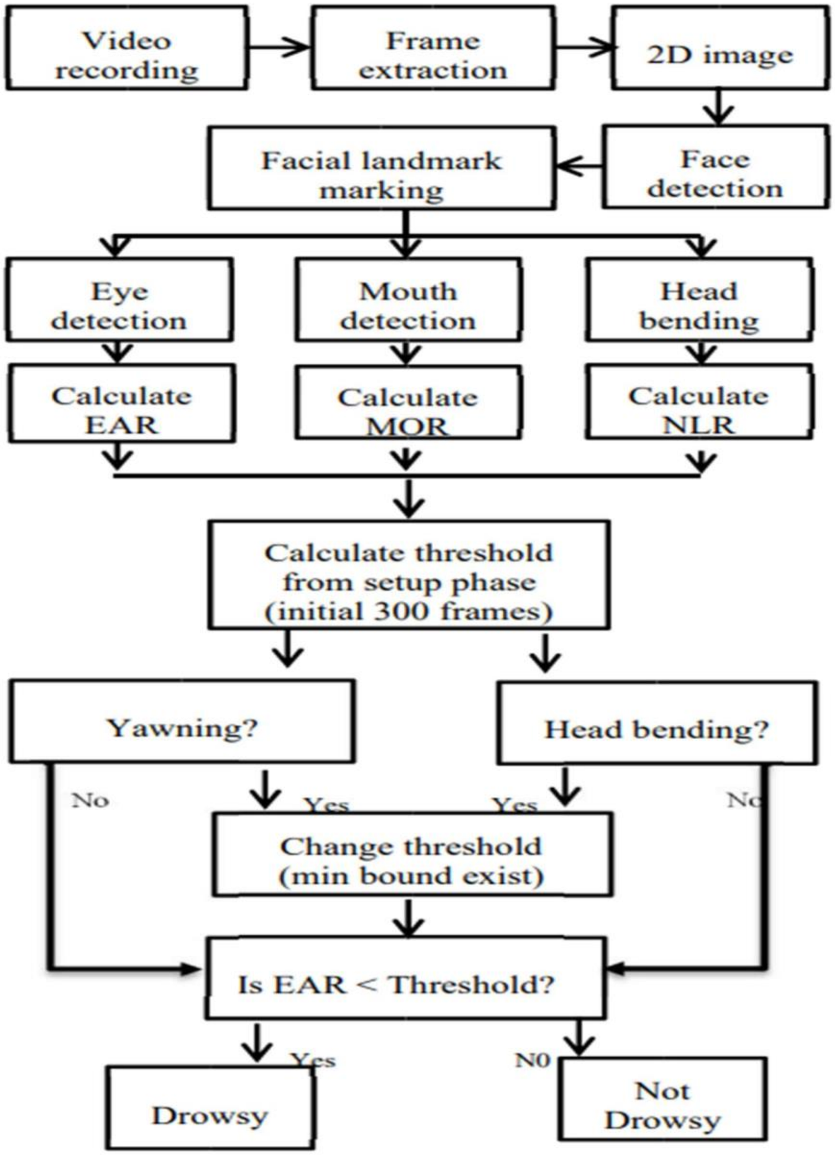
Sl No	Register No	Name	Role
1	RA2111003011066	Saurabh Mittal	Rep
2	RA2111003011065	Priyanshu Naskar	Member
3	RA2111003011070	Malik Aurangzaib	Member

Aim:

To Prepare Work breakdown structure, Timeline chart and Risk identification table

Work Breakdown Structure:

The basic drowsiness detection system has three blocks/modules; acquisition system, processing system and warning system. Here, the video of the drivers frontal face is captured in acquisition system and transferred to the processing block where it is processed online to detect drowsiness.



A block diagram of the proposed driver drowsiness monitoring system has been depicted.

- At first, the video is recorded using a webcam. The camera will be positioned in front of the driver to capture the front face image.
- From the video, the frames are extracted to obtain 2- D images. Face is detected in the frames using histogram of oriented gradients (HOG) and linear support vector machine (SVM) for object detection
- After detecting the face, facial landmarks like positions of eye, nose, and mouth are marked on the images.
- From the facial landmarks, eye aspect ratio, mouth opening ratio and position of the head are quantified and using these features and machine learning.

Timeline- Gantt Chart: A timeline is a chart that depicts how a set of resources are used over time. If you're managing a software project and want to illustrate who is doing what and when, or if you're organizing a conference and need to schedule meeting rooms, a timeline is often a reasonable visualization choice.

TASK	START DATE	END DATE	DURATION
Project Planning	01.04.2023	14.04.2023	2 weeks
Data collection & Processing	15.04.2023	15.05.2023	4 weeks
Process Selection & feature Engineering	16.05.2023	30.05.2023	2 weeks
Model Selection & Optimization	31.05.2023	14.06.2023	2 weeks
Setting Sensor	15.06.2023	30.06.2023	2 weeks
Feedback	01.07.2023	28.07.2023	4 weeks
Maintainance & model Monitering	29.07.2023	30.09.2023	8 weeks

RISK IDENTIFICATION:

RISK ANALYSIS

SWOT analysis is a commonly used method to analyze risk in various projects. Through this method, strength, weakness, opportunities, and threats may be determined.

Strengths:

□ Platform independent

□ Fast and Reliable

□ Accuracy

Weakness:

□ Cannot detect eyes if the user is wearing glasses or sunglasses.

□ New technology emerging daily.

Opportunity:

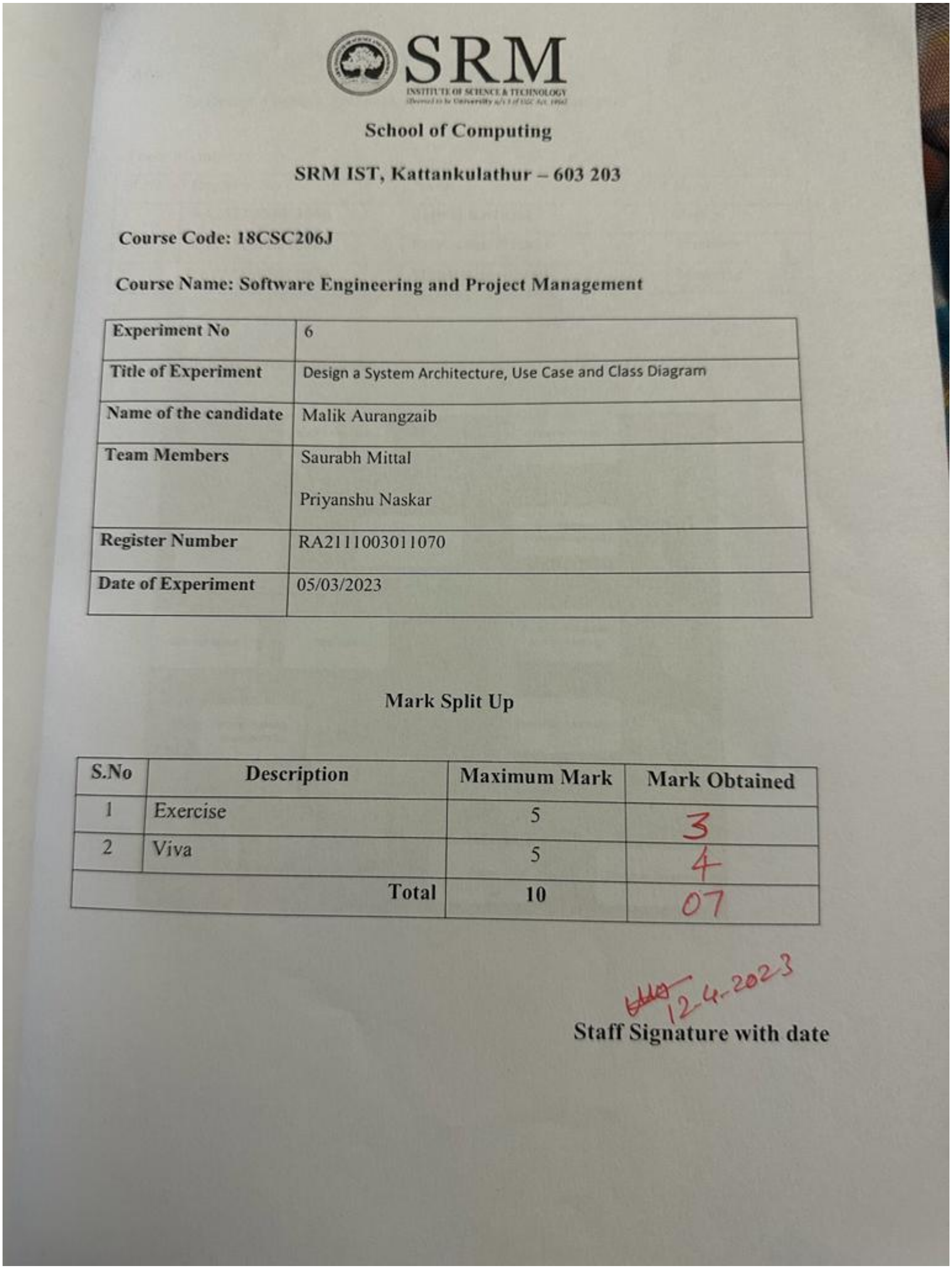
□ No such application is currently being used.

Threats:

□ New technology emerging daily

Result:

Thus, the work breakdown structure with timeline chart and risk table were formulated successfully.



SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	6
Title of Experiment	Design a System Architecture, Use Case and Class Diagram
Name of the candidate	Malik Aurangzaib
Team Members	Saurabh Mittal Priyanshu Naskar
Register Number	RA2111003011070
Date of Experiment	05/03/2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	3
2	Viva	5	4
	Total	**10**	07

12-4-2023

Staff Signature with date

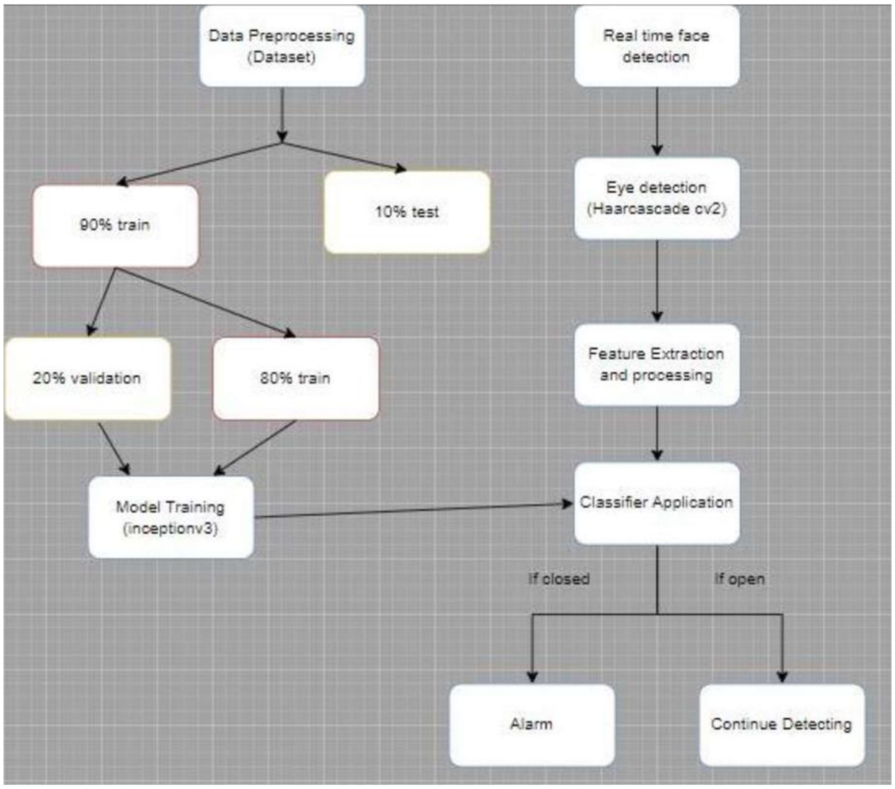
Aim

To Design a System Architecture, Use case and Class Diagram

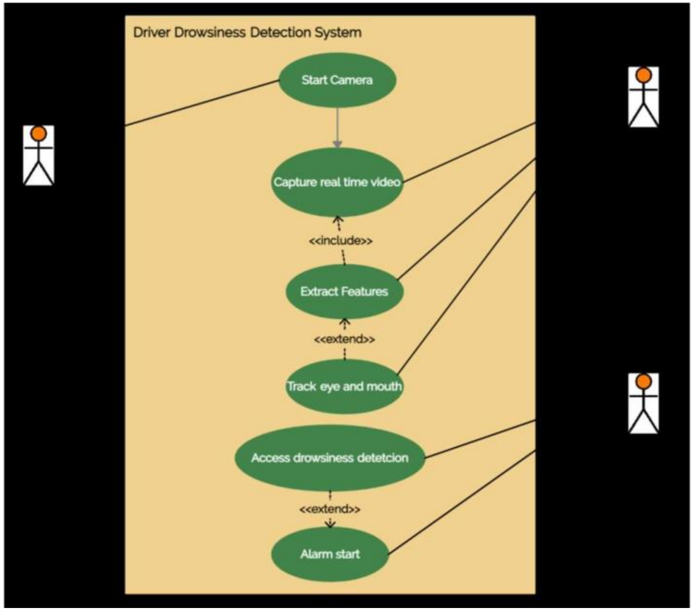
Team Members:

Sl No	Register No	Name	Role
1	RA2111003011066	Saurabh Mittal	Rep
2	RA2111003011065	Priyanshu Naskar	Member
3	RA2111003011070	Malik Aurangzaib	Member

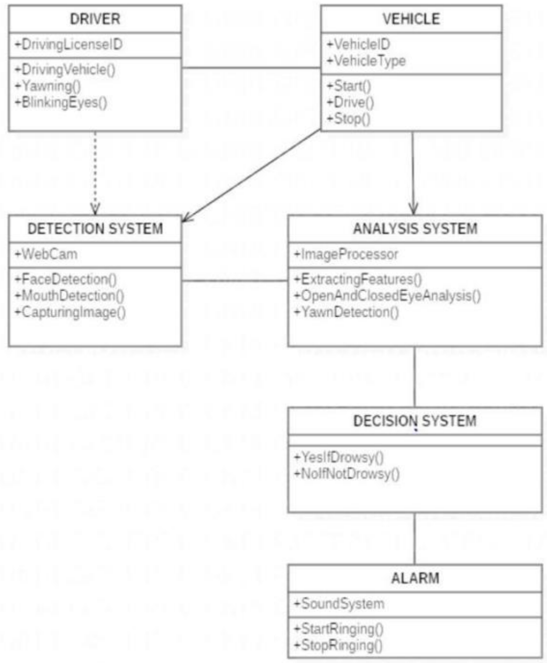
System Architecture Diagram:



Use Case Diagram:

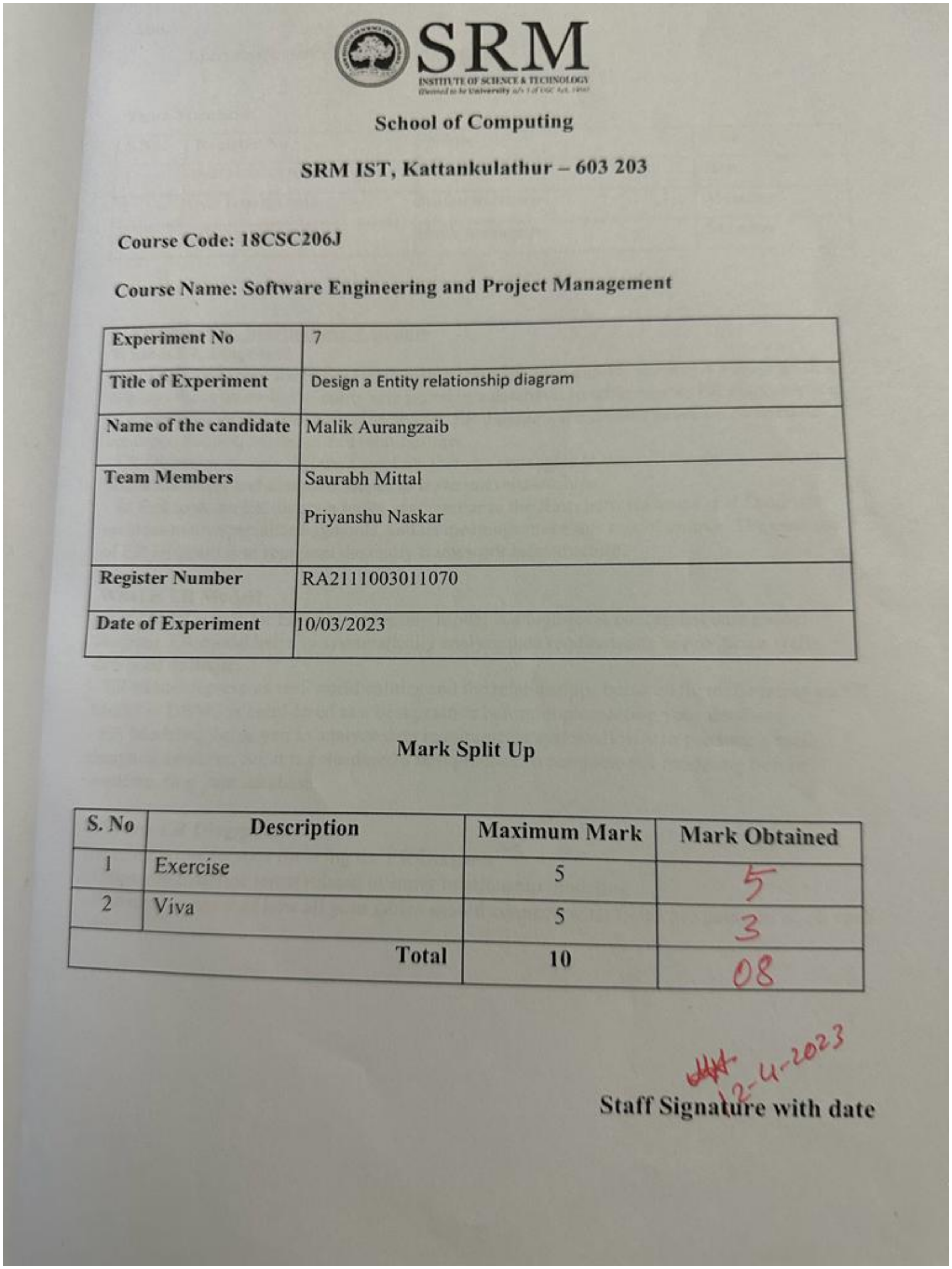


Class Diagram:



Result:

Thus, the system architecture, use case and class diagram created successfully.



SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	7
Title of Experiment	Design a Entity relationship diagram
Name of the candidate	Malik Aurangzaib
Team Members	Saurabh Mittal Priyanshu Naskar
Register Number	RA2111003011070
Date of Experiment	10/03/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	5
2	Viva	5	3
	Total	**10**	08

12-4-2023

Staff Signature with date

Aim

To create the Entity Relationship Diagram

Team Members:

S No	Register No	Name	Role
1	RA2111003011066	Saurabh Mittal	**Rep**
2	RA2111003011065	Priyanshu Naskar	**Member**
3	RA2111003011070	Malik Aurangzaib	**Member**

*/ ER Diagram, Notation and Example

What is ER Diagram?

- ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.
- ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.
- At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

What is ER Model?

- ER Model stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyze data requirements to produce a well-designed database.
- ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered as a best practice before implementing your database.
- ER Modeling helps you to analyze data requirements systematically to produce a well-designed database. So, it is considered a best practice to complete ER modeling before implementing your database.

Why use ER Diagrams?

Here, are prime reasons for using the ER Diagram

- Helps you to define terms related to entity relationship modeling
- Provide a preview of how all your tables should connect, what fields are going to be on each table

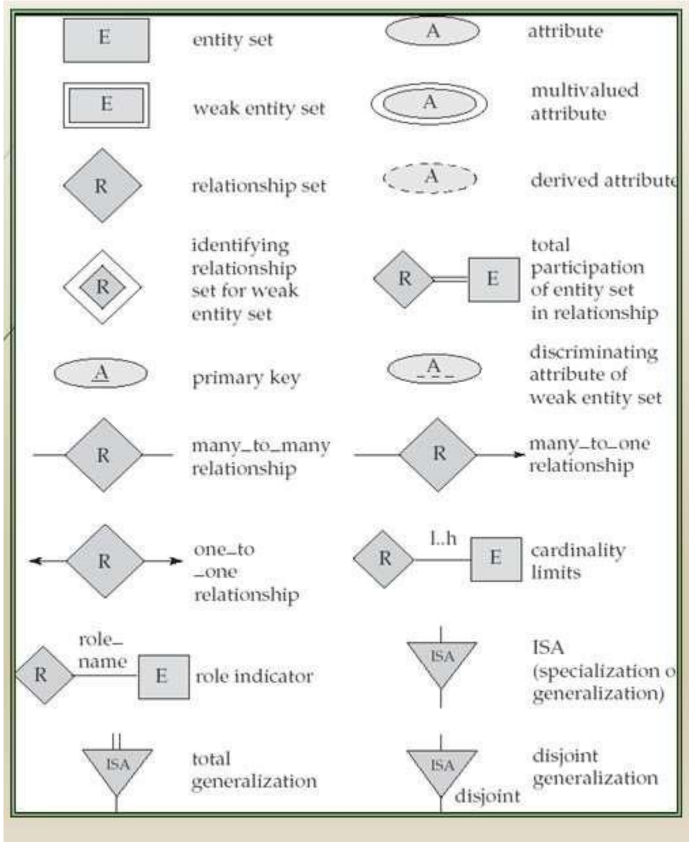
- Helps to describe entities, attributes, relationships
- ER diagrams are translatable into relational tables which allows you to build databases quickly
- ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
- The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram
- ERD Diagram allows you to communicate with the logical structure of the database to users

Components of the ER Diagram

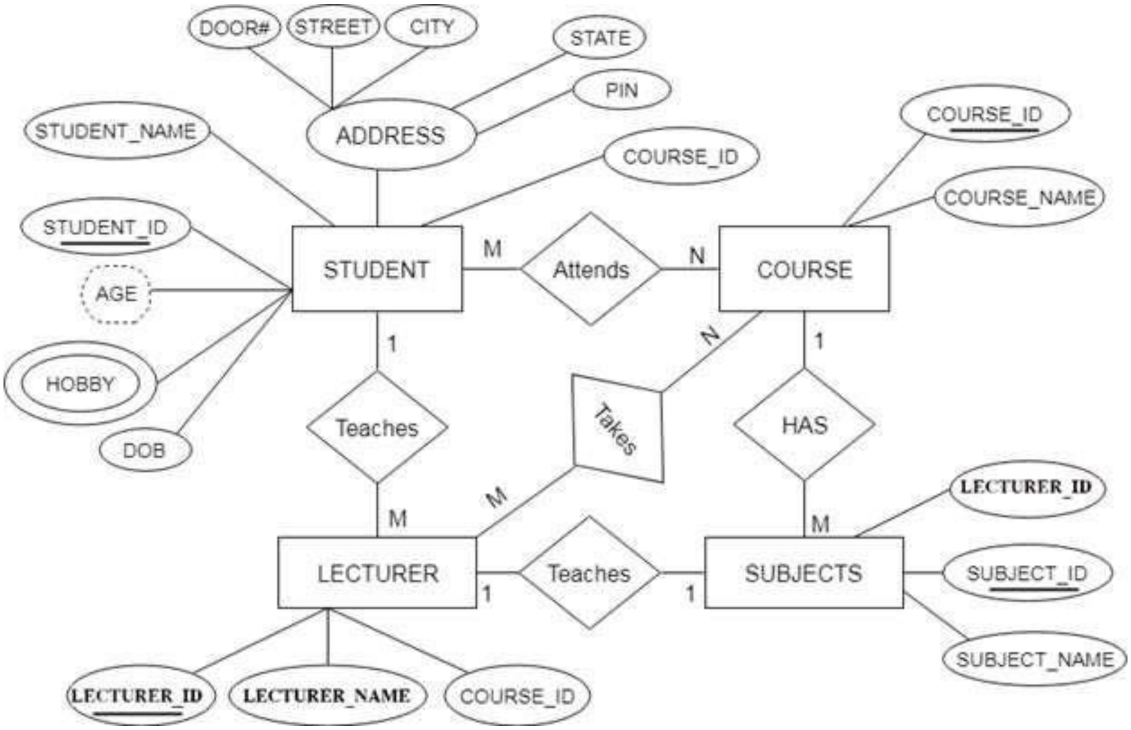
This model is based on three basic concepts: Entities, Attributes, Relationships

ER Diagram – Notations

- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
- Double ellipses represent multivalued attributes.
- Dashed ellipses denote derived attributes.
- Underline indicates primary key attributes



ER Diagram of University Database



ADDITIONAL NOTES

- A database can be modeled as a collection of entities, relationship among entities.
- An entity is an object that exists and is distinguishable from other objects.

Example: specific person, company, event, plant

- Entities have attributes.

Example: people have names and addresses

- An entity set is a set of entities of the same type that share the same properties.

Example: set of all persons, companies, trees, holidays

- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- We express cardinality constraints by drawing either a directed line (->), signifying “one,” or an undirected line (—), signifying “many,” between the relationship set and the entity set.

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

Example: customer = (customer-id, customer-name, customer-street, customer-city)
loan = (loan-number, amount)

- Domain – the set of permitted values for each attribute
- Attribute types:

1. Simple and composite attributes.
2. Single-valued and multi-valued attributes

E.g. multivalued attribute: phone-numbers
3. Derived attributes-Can be computed from other attributes
E.g. age, given date of birth

Cardinality
- For a binary relationship set the mapping cardinality must be one of the following types:
1. One to one
A customer is associated with at most one loan via the relationship borrower. A loan is associated with at most one customer via borrower
2. One to many
A loan is associated with at most one customer via borrower, a customer is associated with several (including 0) loans via borrower
3. Many to one
A loan is associated with several (including 0) customers via borrower, a customer is associated with at most one loan via borrower
4. Many to many
A loan is associated with several (including 0) customers via borrower, a customer is associated with several loans (including 0) via borrower

Weak Entity Set
- An entity set that does not have a primary key is referred to as a weak entity set and represented by double outlined box in E-R diagram.
Example : Consider the entity set payment which got three attributes : payment_number, payment_date and payment_amount. Payment numbers are sequential starting from 1 generally separately for each loan. Although each payment entity is distinct, payments for different loans may share the same payment number. Thus this entity set does not have a primary key.

Discriminator
- The discriminator (or partial key) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set
Example: discriminator of weak entity set payment is the attribute payment_number since for each loan a payment number uniquely identifies one single payment for that loan.

Specialization-Generalization-ISA
- E-R model provides means of representing these distinctive entity groupings
- Process of designating subgroupings within an entity set is called specialization depicted by triangle component labelled ISA (“is a”)
- Bottom up design process in which multiple entity sets are synthesized into higher level entity set - Generalization
- ISA relationship may also be referred to as superclass-subclass relationship
- Higher and lower level entity sets are designated by the terms superclass and subclass.
- Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.

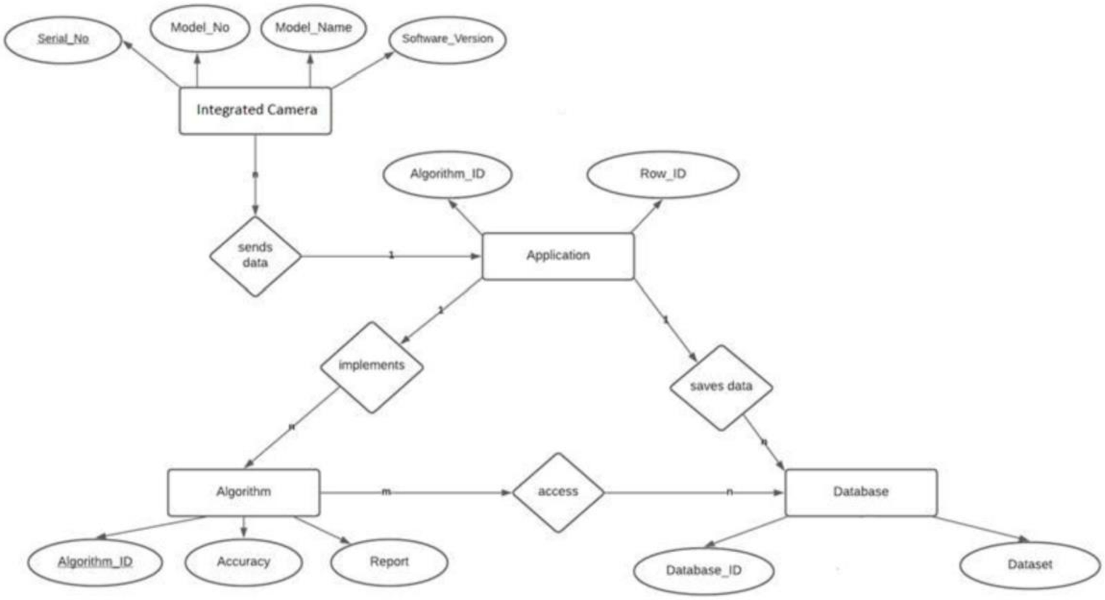
Total & Partial Participation
- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set
E.g. participation of loan in borrower is total, every loan must have a customer associated to it via borrower

- Partial participation: some entities may not participate in any relationship in the relationship set
Example: participation of customer in borrower is partial

Cardinality limits
- Cardinality limits can also express participation constraints
- Minimum and maximum cardinality is expressed as l..h where l is the minimum and h is the maximum cardinality
- Minimum value of 1 indicates total participation of entity set in relationship set
- Maximum value of 1 indicates entity participates in atmost one relationship set.
- Maximum value of * indicates no limit

Role indicator
- Entity sets of a relationship need not be distinct
- The labels “manager” and “worker” are called roles; they specify how employee entities interact via the works-for relationship set.
- Roles are indicated in E-R diagrams by labeling the lines that connect diamonds to rectangles.
- Role labels are optional, and are used to clarify semantics of the relationship

Disjoint Generalization
- Disjointness constraint requires that an entity belong to more than one lower level entity set.
Example: account entity can satisfy only one condition for account_type attribute ; entity can either be savings or chequing account but not both.



Result:

Thus, the entity relationship diagram was created successfully.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	8
Title of Experiment	Develop a Data Flow Diagram (Process-Up to Level 1)
Name of the candidate	Malik Aurangzaib
Team Members	Saurabh Mittal Priyanshu Naskar
Register Number	RA2111003011070
Date of Experiment	14/03/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	2.5
2	Viva	5	2.5
	Total	**10**	5

28/04/23

Staff Signature with date

Aim

To develop the data flow diagram up to level 1 for the Driver Drowsiness Detection.

Team Members:

S No	**Register No**	**Name**	**Role**
1	RA2111003011066	Saurabh Mittal	**Rep**
2	RA2111003011065	Priyanshu Naskar	**Member**
3	RA2111003011070	Malik Aurangzaib	**Member**

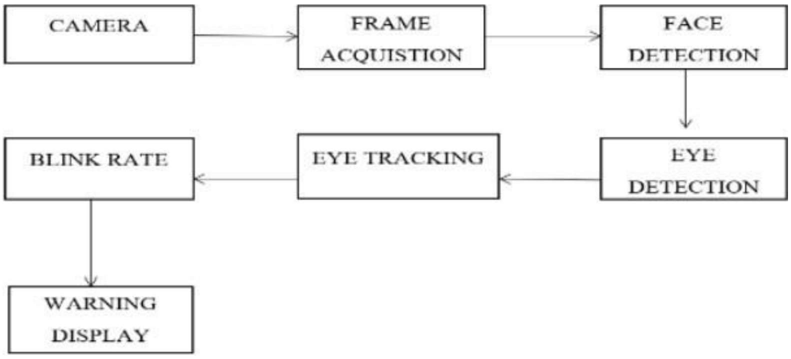
Data Flow Diagram

The DFD takes an input-process-output view of a system. That is, data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software. Data objects are represented by labeled arrows, and transformations are represented by circles (also called bubbles). The DFD is presented in a hierarchical fashion. That is, the first data flow model (sometimes called a level 0 DFD or context diagram) represents the system as a whole. Subsequent data flow diagrams refine the context diagram, providing increasing detail with each subsequent level.

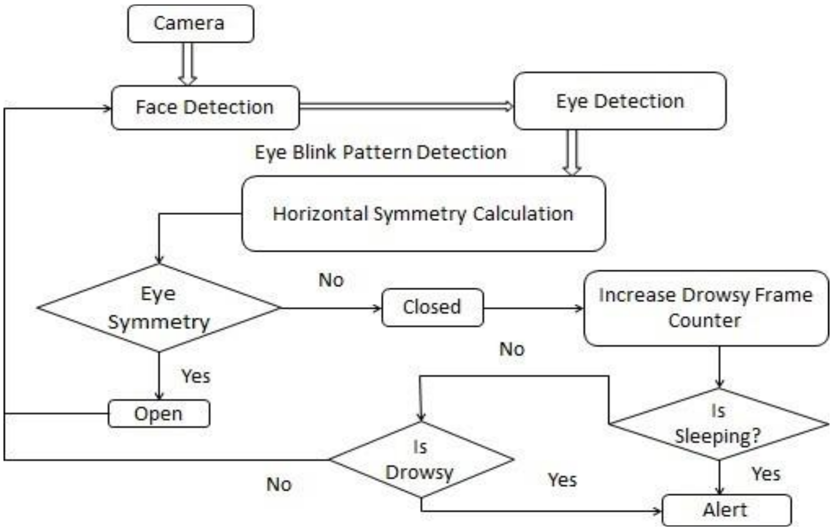
The data flow diagram enables you to develop models of the information domain andfunctional domain. As the DFD is refined into greater levels of detail, you perform animplicit functional decomposition of the system. At the same time, the DFD refinementresults in a corresponding refinement of data as it moves through the processes that embody theapplication. A few simple guidelines can aid immeasurably during the derivation of a data flow diagram:
(1) Level 0 data flow diagram should depict the software/system as a
single bubble;
(2) Primary input and output should be carefully noted;
(3) Refinement should begin by isolating candidate processes, data objects, and data stores to be represented at the next level;
(4) All arrows and bubbles should be labeled with meaningful names;
(5) Information flow continuity must be maintained from level to level and

(6) One bubble at a time should be refined. There is a natural tendency to overcomplicate the data flow diagram. This occurs when you attempt to show too much detail too early or represent procedural aspects of the software in lieu of information flow.

Level 0:



Level 1:



Result:

Thus, the data flow diagrams have been created for the Driver Drowsiness Detection.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	9
Title of Experiment	Design a Sequence and Collaboration Diagram
Name of the candidate	Malik Aurangzaib
Team Members	Priyanshu Naskar (RA2111003011065) Saurabh Mittal (RA2111003011066)
Register Number	RA2111003011070
Date of Experiment	12.04.2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	2.5
2	Viva	5	2.5
	Total	**10**	5

20/4/23

Staff Signature with date

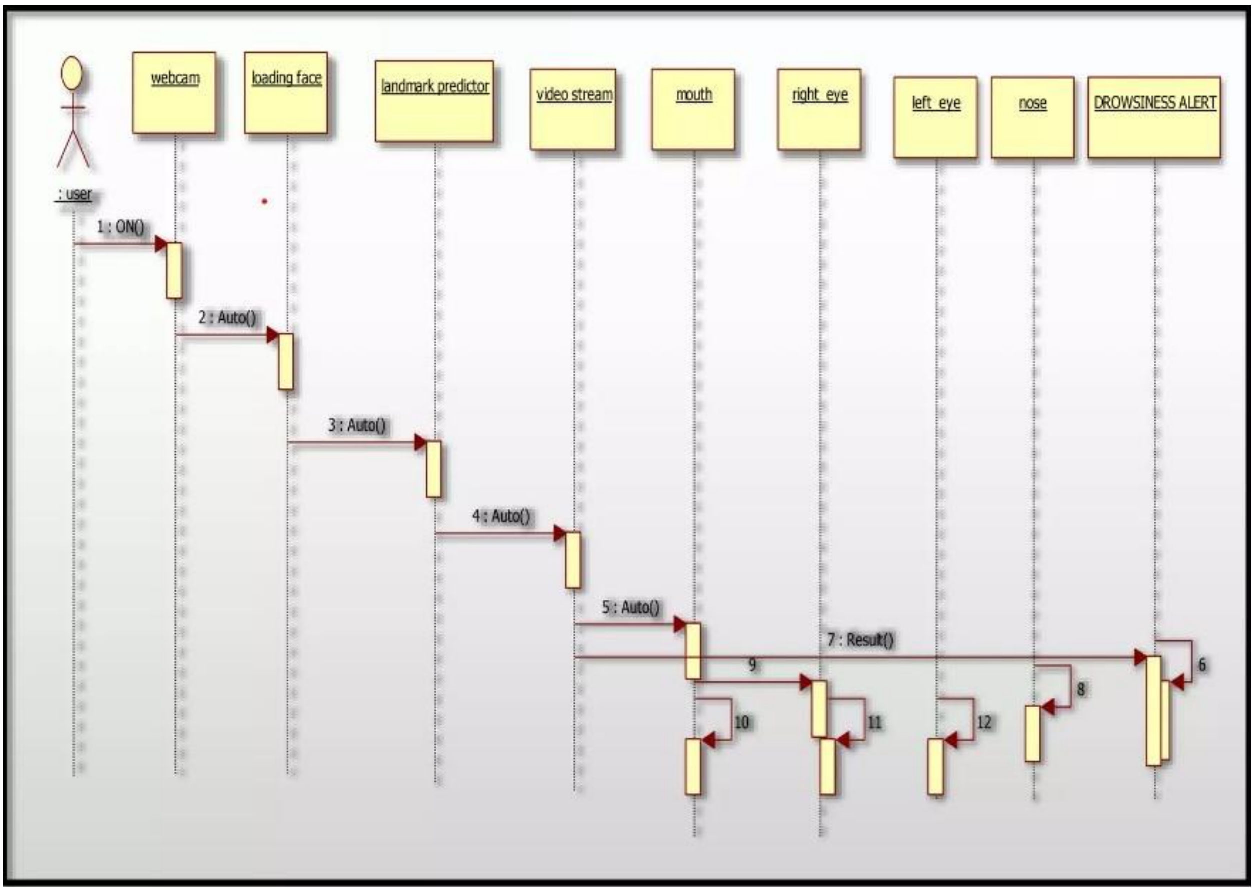
Aim

To create the sequence and collaboration diagram for the Driver Drowsiness Detection.

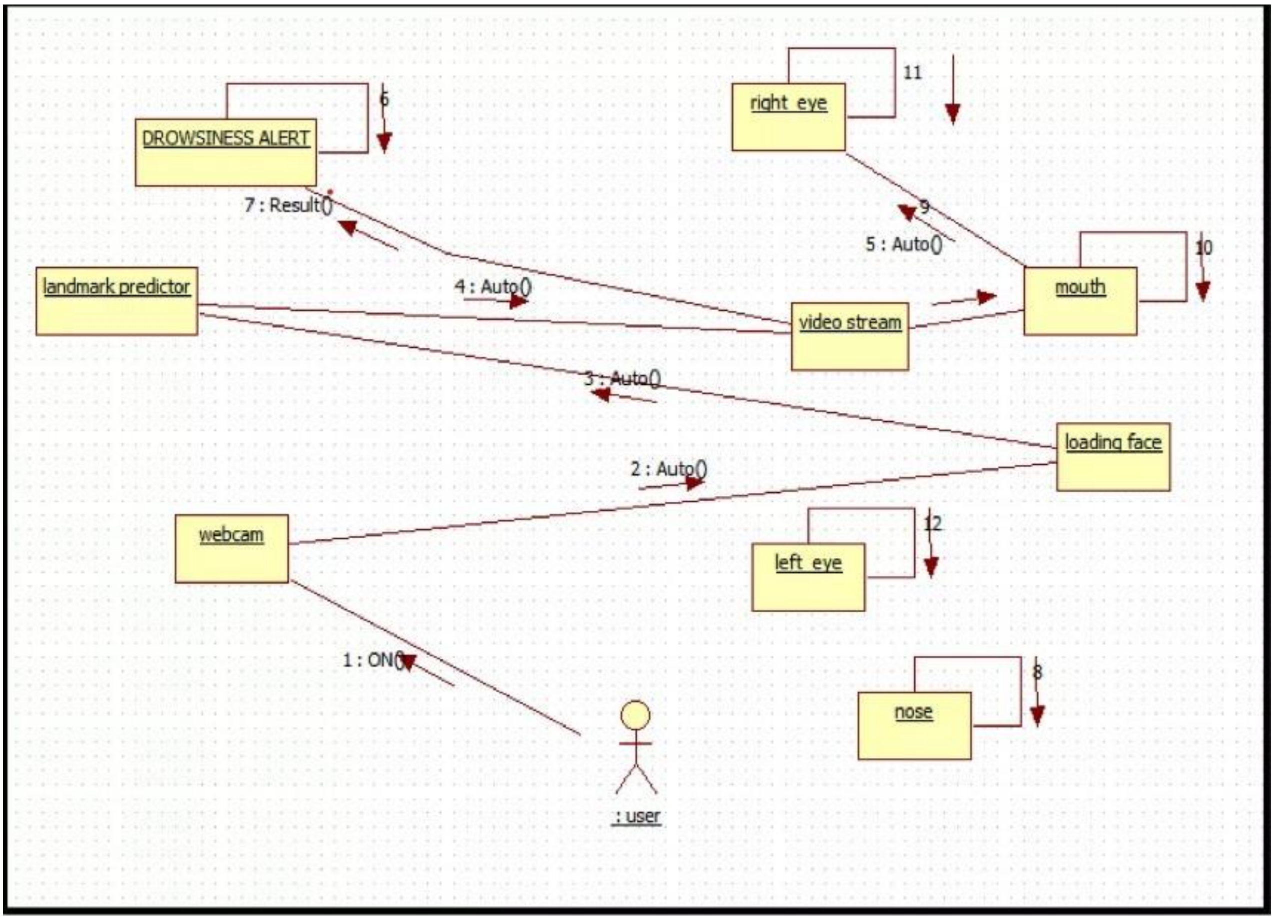
Team Members:

S No	Register No	Name	Role
1	RA2111003011066	Saurabh Mittal	**Rep/Member**
2	RA2111003011065	Priyanshu Naskar	**Member**
3	RA2111003011070	Malik Aurangzaib	**Member**

Sequence Diagram

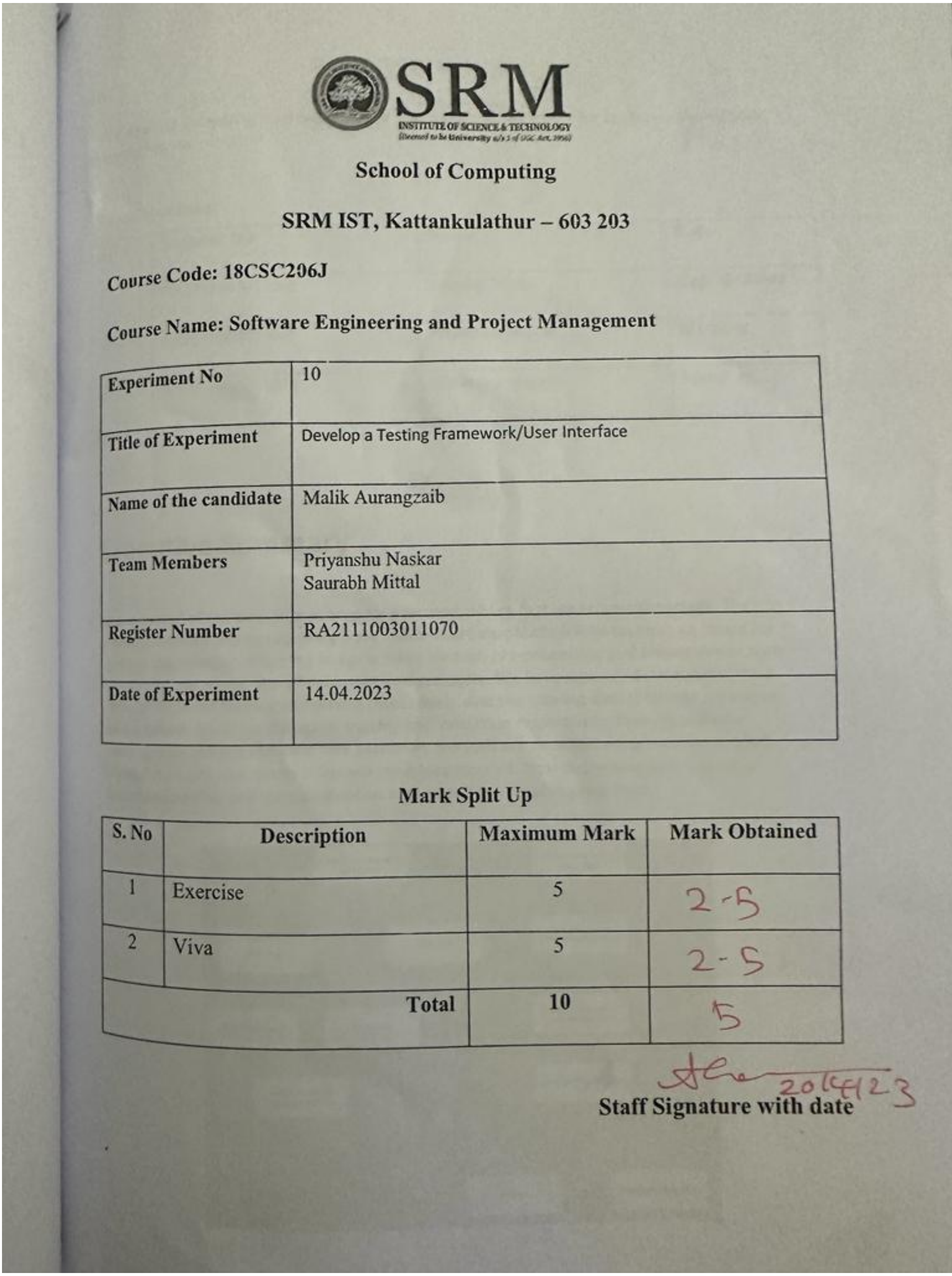


Collaboration Diagram



Result:

Thus, the sequence and collaboration diagrams were created for the <project name>.



SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	10
Title of Experiment	Develop a Testing Framework/User Interface
Name of the candidate	Malik Aurangzaib
Team Members	Priyanshu Naskar Saurabh Mittal
Register Number	RA2111003011070
Date of Experiment	14.04.2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	2.5
2	Viva	5	2.5
	Total	**10**	5

20/4/23

Staff Signature with date

Aim:

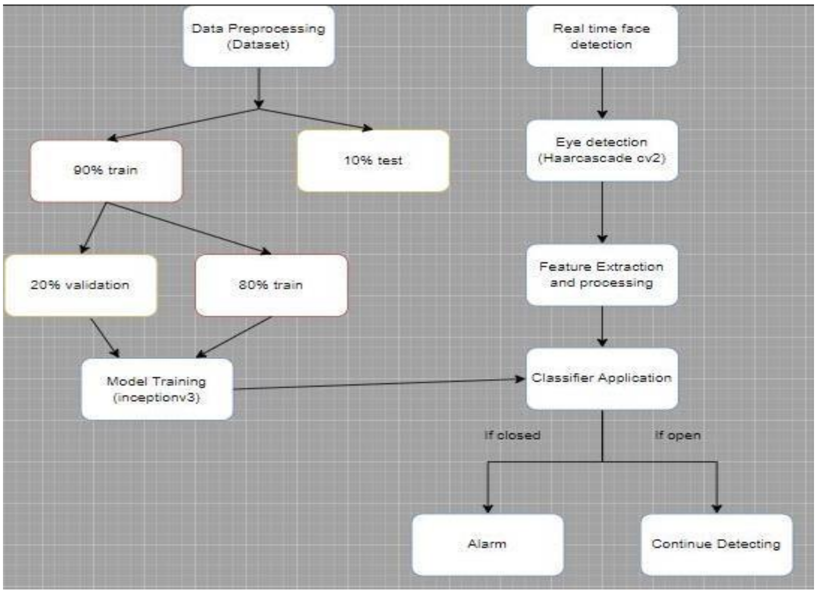
To develop the testing framework and/or user interface framework for the Driver Drowsiness Detection System.

Team Members:

S No	Register No	Name	Role
1	RA2111003011066	Saurabh Mittal	**Rep/Member**
2	RA2111003011065	Priyanshu Naskar	**Member**
3	RA2111003011070	Malik Aurangzaib	**Member**

Executive Summary

It consists of all the step-wise procedure executed, where first we acquire the image. The camera installed will be capturing a live video where snapshots will be taken as an image for further classification. Then the image is taken through pre-processing and enhancement such as resizing and conversion of the image into grayscale. We have separated the datasetinto a ratio of 90:10 for training and testing respectively. And the training data is further separated into a ratio of 80:20 for the actual training and validation respectively. Then through the training data and the real time face detection, the eyes are detected using Haarcascade cv2. Using the results, the model is trained using Inception V3. Now the testing data is used for feature extraction and the classification is done and the result is presented.



Test Plan

Scope of Testing

Module 1

We acquired the dataset which consists of eye images of 33 Males and 4 Females with atotal of 84,898 images. The images consist of the following variations:

I. With and without glasses
II. Open and closed eyes
III. None, small or big reflections
IV. Good and bad lighting conditions

Images capturedby3differentsensors:

1. Intel RealSense RS 300 sensor with 640 x 480 resolution
2. IDS Imaging sensor with 1280 x 1024 resolution
3. Aptina sensor with 752 x 480 resolution

We further classified the data into train, test and validation data and performed transfer learning wherein a pre-trained model (InceptionV3) in Keras (Neural Network Library thatis part of TensorFlow) is used as the starting point for our model due to the large size of the dataset.
IMPLEMENTATION:

- Divided the dataset into closed and open eyes
- Divided the dataset into 90% train data and 10% test data (8,008) and resized all theimages
- Further divided the train data into 80% train data (61,513) and 20% validation data (15,377)
- Used InceptionV3 transfer learning model to use the pre-trained layers of the modelexcept the last fully connected layer that is specific to ImageNet
- Modified the model based on our requirements such as flattening the output layer to1 dimension, adding a fully connected layer with 64 hidden units and ReLU activation (default activation function as it is easier to train the model and increases performance), adding a drop rate of 0.5 and adding a final softmax layer for classification
- Created checkpoints to save only the best model i.e; models with the leastvalidation loss

Module 2

Module 2 includes the following:

- Model Training for 50 epochs wherein only the models with lower validation lossare saved

- Detection of a face in a real time video captured through the primary camera of thedevice.

- Followed by detection of the face the eyes are detected.

- The images are passed through the CNN model and classified as open or closed.

- Generation of the score for every frame to trigger the alarm

IMPLEMENTATION:

- The CNN model is trained for 50 epochs with a batch size of 8 for every epoch
- Categorical cross entropy is used to evaluate the loss after each epoch
- We have used Haarcascade to identify the face followed by identification of theeyes

- Haarcascade_frontalface_default.xml has a coded list of decision trees that help categorize an image into a face or not and correspondingly haarcascade_eye.xmlthat categorizes the eyes

- We begin by capturing a real time video through our primary camera and creating aframe of the face by using a min of 3 neighbors.

- Rectangle is created around the face to further detect the eyes and process it

- This model has an approximate latency of 40ms where the eyes are categorized asclosed if it says closed for 0.3secs or more and categorized as open if it stays openfor 0.9secs or more.
- A score is generated for every frame captured
- If the score exceeds 15 an alarm sound is played (0.3secs x15)
- That is if the eyes remain closed for a period of 4.5 seconds the alarm is played

Types of Testing, Methodology, Tools

Category	Methodology	Tools Required
Eye detection	Preset	1. Intel RealSense RS 300 sensor with 640 x 480 resolution 2. IDS Imaging sensor with 1280 x 1024 resolution 3. Aptina sensor with 752 x 480 resolution
Machine Learning Model	Pre-Trained Layers	InceptionV3
Face and Eye Detection	Coded Lists	Haarcascade

Result:

Thus, the testing framework/user interface framework has been created for the Driver drowsiness detection System.

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	11
Title of Experiment	Test Cases
Name of the candidate	Malik Aurangzaib
Team Members	Saurabh Mittal\ Priyanshu Naskar
Register Number	RA2111003011070
Date of Experiment	10.04.2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	4
2	Viva	5	3
	Total	**10**	07

4-5-23

Staff Signature with date

Aim

To develop the test cases manual for Driver Drowsiness Detection System

Team Members:

S No	Register No	Name	Role
1	RA2111003011066	Saurabh Mittal	Rep
2	RA2111003011065	Priyanshu Naskar	Member
3	RA2111003011070	Malik Aurangzaib	Member

Test Case

Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1	Test the system under different lighting conditions, such as bright sunlight, low light, and night-time driving.	The system should be tested under different conditions such as rain, fog, and snow. bright sunlight, low light, and night-time driving.	If any defects or issues were identified. They should be documented and prioritized based on their severity and impact on the system's performance.	All identified issues and defects should be resolved and the system should be retested to ensure that they have been fixed.	If the system meets the required specifications and performs as expected during testing, the actual outcome would be considered successful.	Pass	Success
2	Test the system with different facial expressions, including wearing	Analyze the test results and determine the issues and audio alert to the driver when	If defects are found, report them to the development team and retest	If the system meets these requirements and no defects or issues are identified during	If defects are found, they should be reported to the development team and resolved before retesting the system.	Pass	Success

	sunglasses, wearing a face mask, and facial hair.	drowsiness is detected, regardless of facial expression	after they are resolved.	the testing process, the test case is considered successful as expected.			

Non-Functional Test Cases

1. Performance testing

- Verify the response time of the system to detect drowsiness and issue an alert
- Check the system's accuracy in detecting drowsiness under different lightingconditions, such as low light and bright light.
- Test the system's performance when there are multiple passengers in the car.

2. Usability Testing

- Test the system's user interface and ensure that it is easy to understand and use.
- Check if the system provides clear and understandable instructions to the driver incase of an alert.
- Test if the system can differentiate between a drowsy driver and a driver who isfocused on the road.

3. Compatibility Testing

- Test if the system works with different types of vehicles and car models.
- Verify if the system is compatible with different operating systems, hardware, andsoftware.
- Check if the system works with different languages and cultures.

4. Reliability Testing

- Test the system's ability to perform accurately and consistently over a long periodof time.
- Verify that the system does not produce false alerts or misses real drowsinessevents.
- Test the system's fault tolerance and its ability to recover from system crashes orfailures.

Result:

Thus, the test case manual has been created for the Driver Drowsiness DetectionSystem.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	12
Title of Experiment	Manual Test Case Reporting
Name of the candidate	Malik Aurangzaib
Team Members	Saurabh Mittal Priyanshu Naskar
Register Number	RA2111003011070
Date of Experiment	16.04.2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	4
2	Viva	5	4
	Total	**10**	08

4-5-23

Staff Signature with date

Aim

To prepare the manual test case report for the Driver Drowsiness Detection System

Team Members:

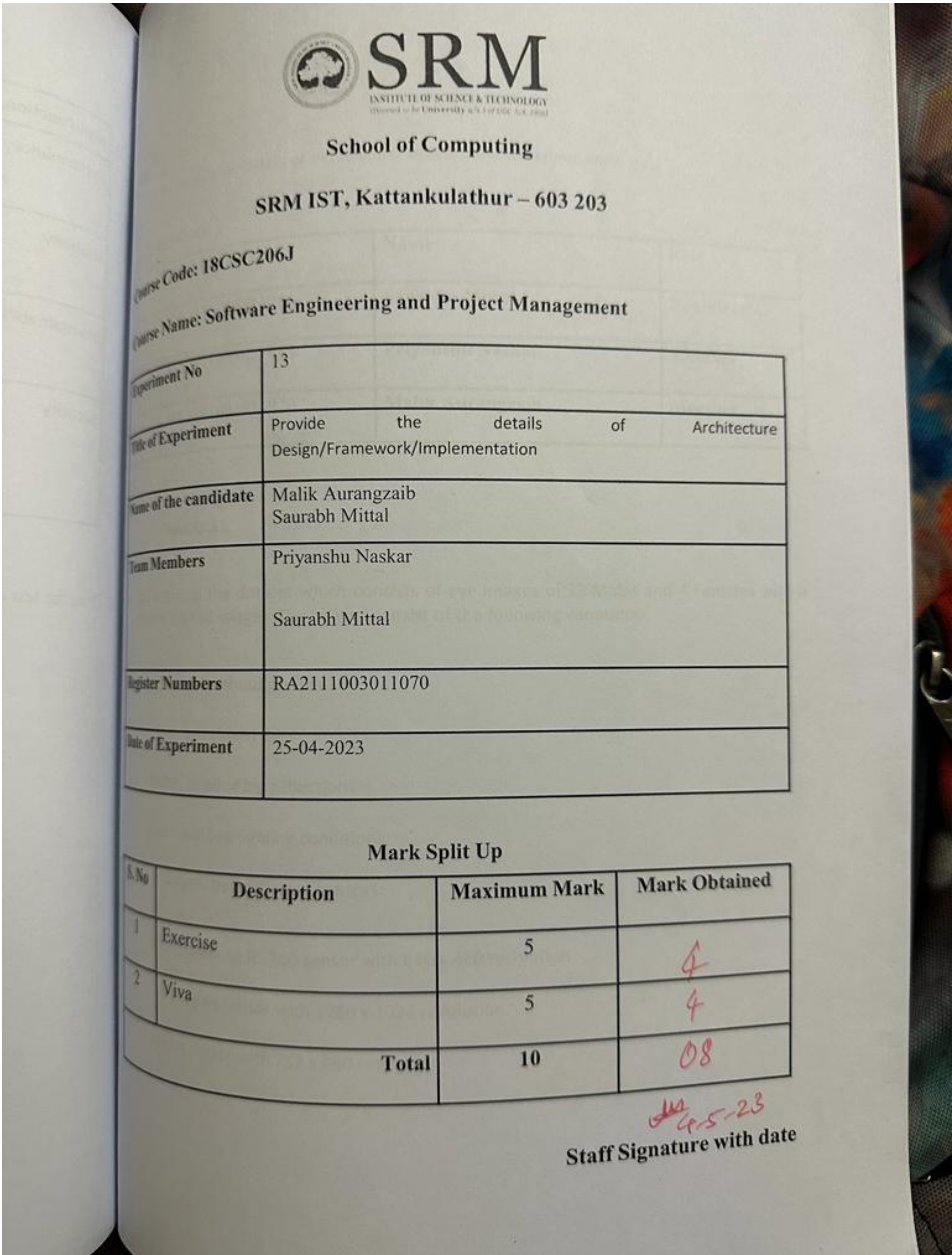
S No	Register No	Name	Role
1	RA2111003011066	Saurabh Mittal	**Rep/Member**
2	RA2111003011065	Priyanshu Naskar	**Member**
3	RA2111003011070	Malik Aurangzaib	**Member**

Functional Testing	Result Analysis	Status
Test case creation: Test cases are designed to simulate real-world driving conditions and scenarios, including various levels of driver drowsiness.	The results of the test cases are analyzed to determine if the system met the expected results and if there are any defects or issues that need to be addressed.	Successful
Defect reporting: Any defects or issues identified during the testing process are reported to the development team for resolution.	Once defects have been resolved, the system is retested to verify that the fixes are implemented correctly and that the system functions as intended	Successful

Functional	Test Case Coverage (%)	Status
User Interface	70%	In-Progress

Non-Functional:	**Actions/Results:**
Performance:	The system's response time is tested under different loads to ensure that it can process data and provide alerts in real-time without any significant delays.
Usability:	Testing can include evaluating the system's user interface, its ability to provide clear and concise alerts, and the overall user experience.
Maintainability:	The system is tested to ensure that it is easy to maintain and modify. This includes evaluating the system's architecture, documentation, and ease of debugging and troubleshooting.
Security:	The system is tested to ensure that it is secure and protected from potential threats, including hacking and unauthorized access. This testing includes evaluating the system's authentication and authorization mechanisms, data encryption, and access controls.

Result: Thus, the test case report has been created for the Driver Drowsiness Detection System



SRM INSTITUTE OF SCIENCE & TECHNOLOGY

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	13
Title of Experiment	Provide the details of Architecture Design/Framework/Implementation
Name of the candidate	Malik Aurangzaib Saurabh Mittal
Team Members	Priyanshu Naskar Saurabh Mittal
Register Numbers	RA2111003011070
Date of Experiment	25-04-2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	4
2	Viva	5	4
	Total	**10**	08

4-5-23

Staff Signature with date

Aim

To provide the details of architectural design/framework/implementation

Team Members:

S No	Register No	Name	Role
1	RA2111003011066	Saurabh Mittal	Rep/Member
2	RA2111003011065	Priyanshu Naskar	Member
3	RA2111003011070	Malik Aurangzaib	Member

MODULE 1

We acquired the dataset which consists of eye images of 33 Males and 4 Females with atotal of 84,898 images. The images consist of the following variations:

- With and without glasses
- Open and closed eyes
- None, small or big reflections
- Good and bad lighting conditions

Images captured by 3 different sensors:

1. Intel RealSense RS 300 sensor with 640 x 480 resolution
2. IDS Imaging sensor with 1280 x 1024 resolution
3. Aptina sensor with 752 x 480 resolution

We further classified the data into train, test and validation data and performed transfer learning wherein a pre-trained model (InceptionV3) in Keras (Neural Network Library thatis part of TensorFlow) is used as the starting point for our model due to the large size of thedataset.

IMPLEMENTATION:

· Divided the dataset into closed and open eyes

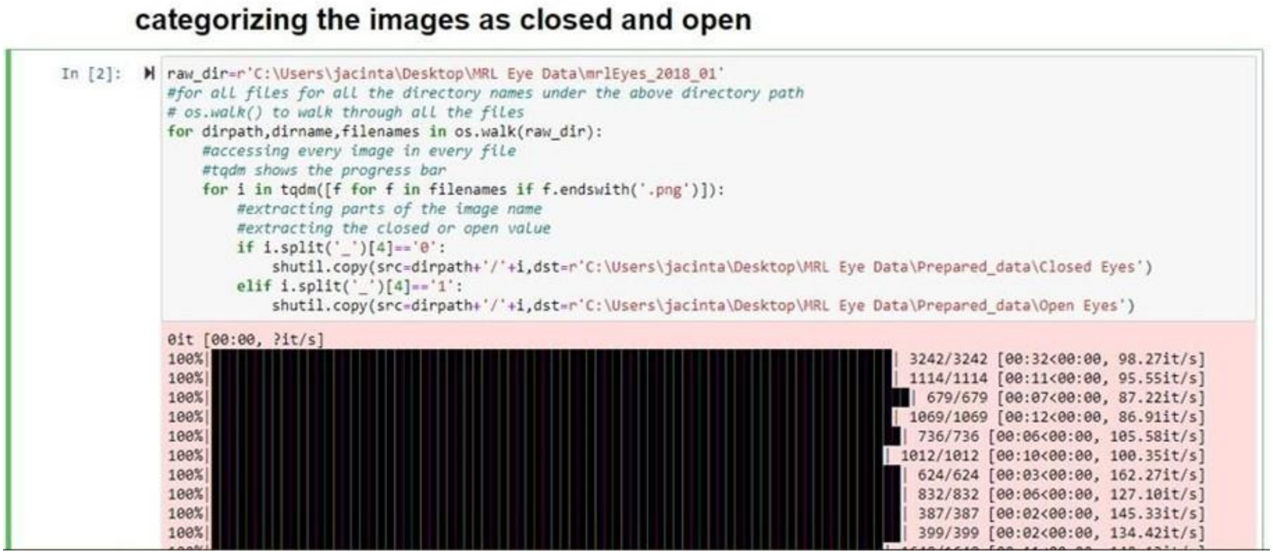
· Divided the dataset into 90% train data and 10% test data(8,008) and resized all the images.

· Further divided the train data into 80% train data(61,513) and 20% validation data(15,377)

· Used InceptionV3 transfer learning model to use the pre-trained layers of the model except the last fully connected layer that is specific to ImageNet

· Modified the model based on our requirements such as flattening the output layer to 1 dimension, adding a fully connected layer with 64 hidden units and ReLU activation(default activation function as it is easier to train the model and increases performance), adding a drop rate of 0.5 and adding a final softmax layer for classification

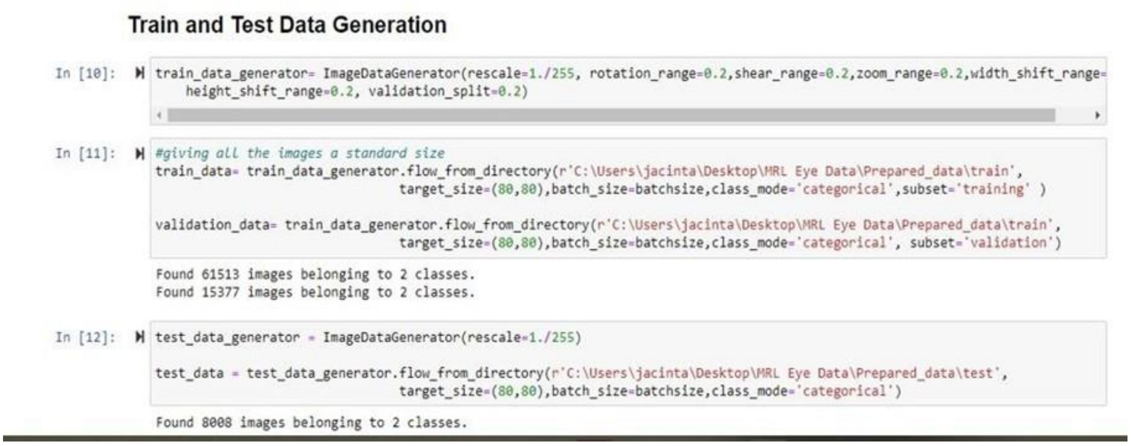
· Created checkpoints to save only the best model i.e; models with the least validation loss



categorizing the images as closed and open

```
In [2]:  raw_dir=r'C:\Users\jacinta\Desktop\MRL Eye Data\mrlEyes_2018_01'
         #for all files for all the directory names under the above directory path
         # os.walk() to walk through all the files
         for dirpath,dirname,filenames in os.walk(raw_dir):
             #accessing every image in every file
             #tqdm shows the progress bar
             for i in tqdm([f for f in filenames if f.endswith('.png')]):
                 #extracting parts of the image name
                 #extracting the closed or open value
                 if i.split('_')[4]=='0':
                     shutil.copy(src=dirpath+'/'+i,dst=r'C:\Users\jacinta\Desktop\MRL Eye Data\Prepared_data\Closed Eyes')
                 elif i.split('_')[4]=='1':
                     shutil.copy(src=dirpath+'/'+i,dst=r'C:\Users\jacinta\Desktop\MRL Eye Data\Prepared_data\Open Eyes')
```

```
0it [00:00, ?it/s]
100%|██████████████████████████████████████| 3242/3242 [00:32<00:00, 98.27it/s]
100%|██████████████████████████████████████| 1114/1114 [00:11<00:00, 95.55it/s]
100%|████████████████████████████████████████| 679/679 [00:07<00:00, 87.22it/s]
100%|██████████████████████████████████████| 1069/1069 [00:12<00:00, 86.91it/s]
100%|███████████████████████████████████████| 736/736 [00:06<00:00, 105.58it/s]
100%|█████████████████████████████████████| 1012/1012 [00:10<00:00, 100.35it/s]
100%|███████████████████████████████████████| 624/624 [00:03<00:00, 162.27it/s]
100%|███████████████████████████████████████| 832/832 [00:06<00:00, 127.10it/s]
100%|███████████████████████████████████████| 387/387 [00:02<00:00, 145.33it/s]
100%|███████████████████████████████████████| 399/399 [00:02<00:00, 134.42it/s]
```

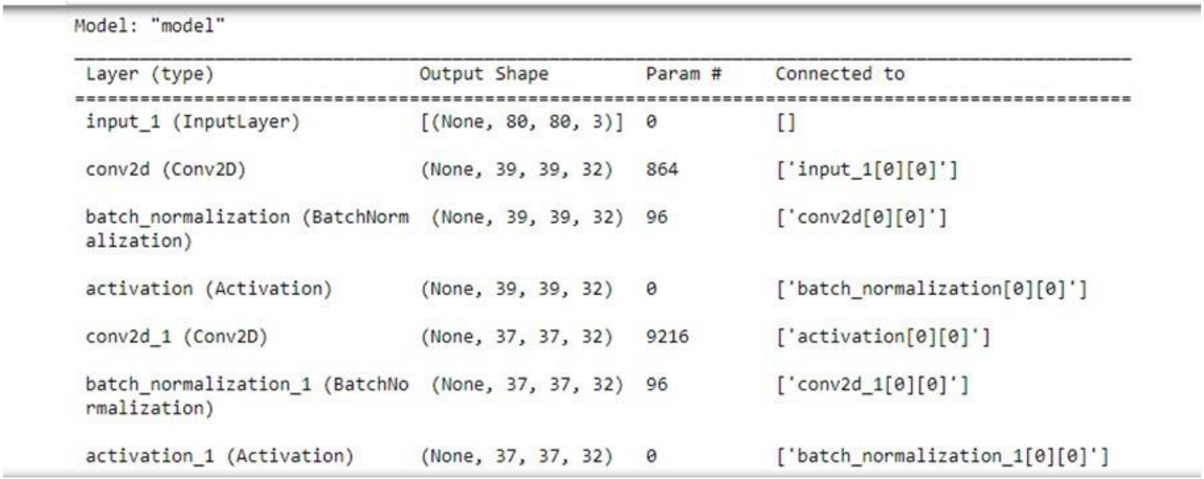


Train and Test Data Generation

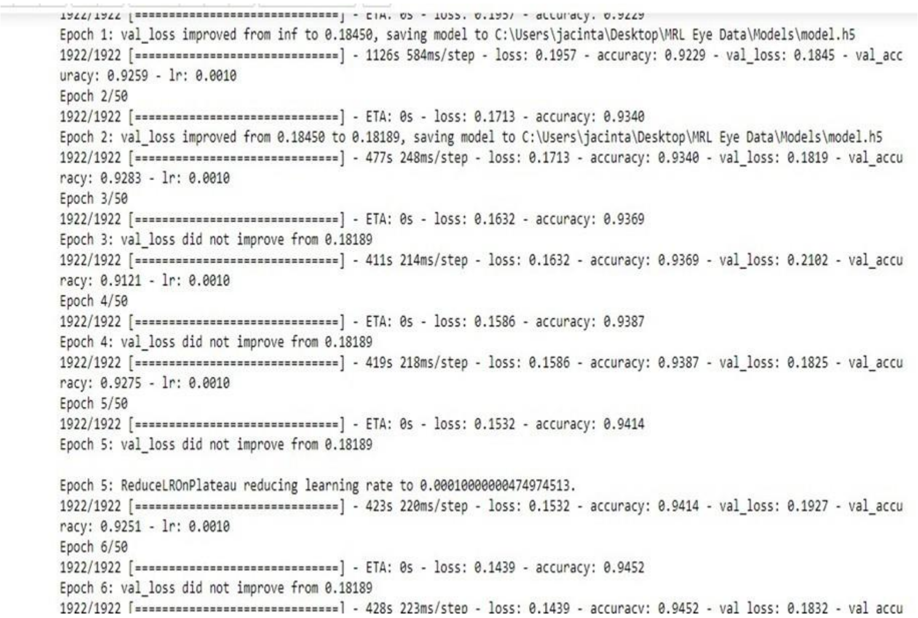
```
In [10]:  train_data_generator= ImageDataGenerator(rescale=1./255, rotation_range=0.2,shear_range=0.2,zoom_range=0.2,width_shift_range=
              height_shift_range=0.2, validation_split=0.2)
```

```
In [11]:  #giving all the images a standard size
          train_data= train_data_generator.flow_from_directory(r'C:\Users\jacinta\Desktop\MRL Eye Data\Prepared_data\train',
                                          target_size=(80,80),batch_size=batchsize,class_mode='categorical',subset='training' )

          validation_data= train_data_generator.flow_from_directory(r'C:\Users\jacinta\Desktop\MRL Eye Data\Prepared_data\train',
                                          target_size=(80,80),batch_size=batchsize,class_mode='categorical', subset='validation')

          Found 61513 images belonging to 2 classes.
          Found 15377 images belonging to 2 classes.
```

```
In [12]:  test_data_generator = ImageDataGenerator(rescale=1./255)

          test_data = test_data_generator.flow_from_directory(r'C:\Users\jacinta\Desktop\MRL Eye Data\Prepared_data\test',
                                          target_size=(80,80),batch_size=batchsize,class_mode='categorical')

          Found 8008 images belonging to 2 classes.
```



```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 80, 80, 3)]  0           []

 conv2d (Conv2D)                (None, 39, 39, 32)   864         ['input_1[0][0]']

 batch_normalization (BatchNorm  (None, 39, 39, 32)  96          ['conv2d[0][0]']
 alization)

 activation (Activation)        (None, 39, 39, 32)   0           ['batch_normalization[0][0]']

 conv2d_1 (Conv2D)              (None, 37, 37, 32)   9216        ['activation[0][0]']

 batch_normalization_1 (BatchNo  (None, 37, 37, 32)  96          ['conv2d_1[0][0]']
 rmalization)

 activation_1 (Activation)      (None, 37, 37, 32)   0           ['batch_normalization_1[0][0]']
```



```
1922/1922 [==============================] - ETA: 0s - loss: 0.1957 - accuracy: 0.9229
Epoch 1: val_loss improved from inf to 0.18450, saving model to C:\Users\jacinta\Desktop\MRL Eye Data\Models\model.h5
1922/1922 [==============================] - 1126s 584ms/step - loss: 0.1957 - accuracy: 0.9229 - val_loss: 0.1845 - val_acc
uracy: 0.9259 - lr: 0.0010
Epoch 2/50
1922/1922 [==============================] - ETA: 0s - loss: 0.1713 - accuracy: 0.9340
Epoch 2: val_loss improved from 0.18450 to 0.18189, saving model to C:\Users\jacinta\Desktop\MRL Eye Data\Models\model.h5
1922/1922 [==============================] - 477s 248ms/step - loss: 0.1713 - accuracy: 0.9340 - val_loss: 0.1819 - val_accu
racy: 0.9283 - lr: 0.0010
Epoch 3/50
1922/1922 [==============================] - ETA: 0s - loss: 0.1632 - accuracy: 0.9369
Epoch 3: val_loss did not improve from 0.18189
1922/1922 [==============================] - 411s 214ms/step - loss: 0.1632 - accuracy: 0.9369 - val_loss: 0.2102 - val_accu
racy: 0.9121 - lr: 0.0010
Epoch 4/50
1922/1922 [==============================] - ETA: 0s - loss: 0.1586 - accuracy: 0.9387
Epoch 4: val_loss did not improve from 0.18189
1922/1922 [==============================] - 419s 218ms/step - loss: 0.1586 - accuracy: 0.9387 - val_loss: 0.1825 - val_accu
racy: 0.9275 - lr: 0.0010
Epoch 5/50
1922/1922 [==============================] - ETA: 0s - loss: 0.1532 - accuracy: 0.9414
Epoch 5: val_loss did not improve from 0.18189

Epoch 5: ReduceLROnPlateau reducing learning rate to 0.00010000000474974513.
1922/1922 [==============================] - 423s 220ms/step - loss: 0.1532 - accuracy: 0.9414 - val_loss: 0.1927 - val_accu
racy: 0.9251 - lr: 0.0010
Epoch 6/50
1922/1922 [==============================] - ETA: 0s - loss: 0.1439 - accuracy: 0.9452
Epoch 6: val_loss did not improve from 0.18189
1922/1922 [==============================] - 428s 223ms/step - loss: 0.1439 - accuracy: 0.9452 - val_loss: 0.1832 - val_accu
```

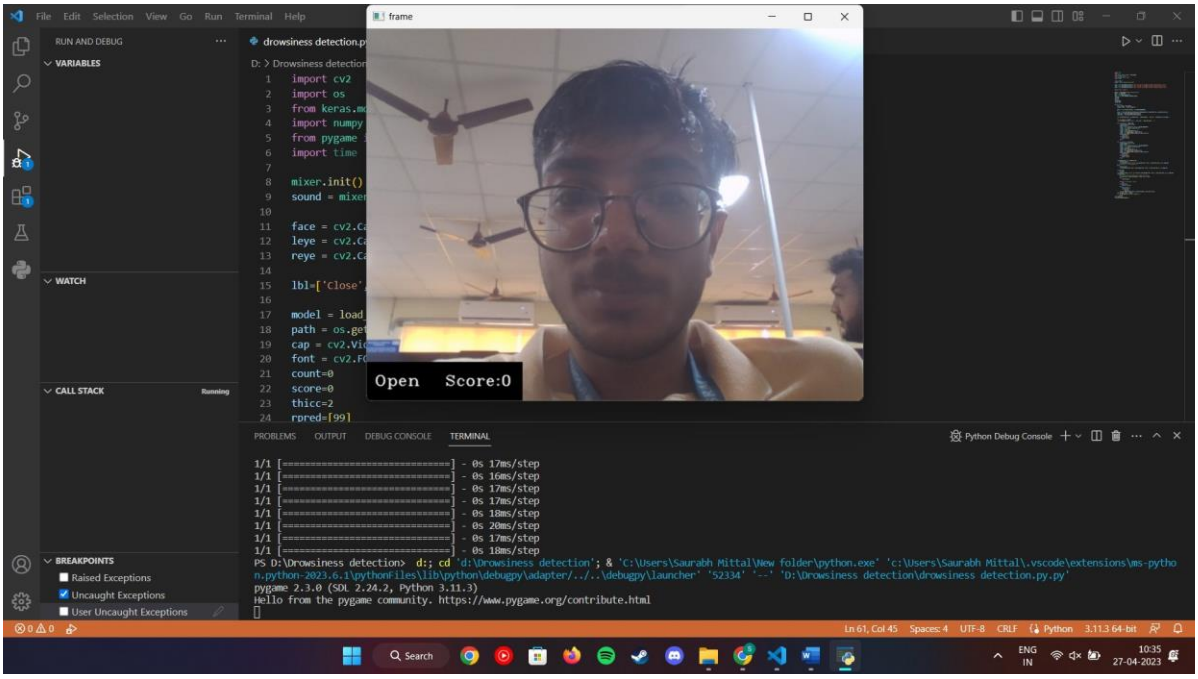
MODULE 2

Module 2 includes the following

- Model Training for 50 epochs wherein only the models with lower validation loss are saved
- Detection of a face in a real time video captured through the primary camera of the device.
- Followed by detection of the face the eyes are detected.
- The images are passed through the CNN model and classified as open or closed.
- Generation of the score for every frame to trigger the alarm

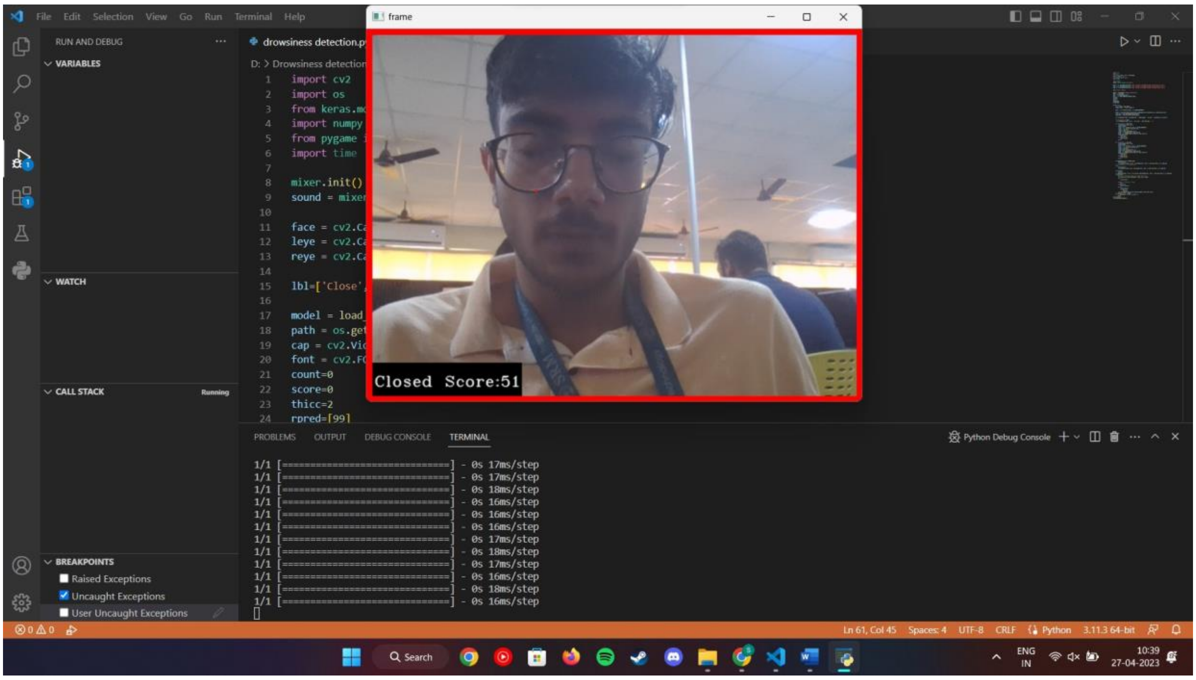
IMPLEMENTATI
ON:

- The CNN model is trained for 50 epochs with a batch size of 8 for every epoch
- Categorical crossentropy is used to evaluate the loss after each epoch
- We have used Haarcascade to identify the face followed by identification of the eyes
- Haarcascade_frontalface_default.xml has a coded list of decision trees that help categorize an image into a face or not and correspondingly haarcascade_eye.xml that categorizes the eyes
- We begin by capturing a real time video through our primary camera and creating a frame of the face by using a min of 3 neighbours
- Rectangle is created around the face to further detect the eyes and process it
- This model has an approximate latency of 40ms where the eyes are categorized as closed if it says closed for 0.3secs or more and categorized as open if it stays open for 0.9secs or more.
- A score is generated for every frame captured
- If the score exceeds 15 an alarm sound is played (0.3secs x15)
- That is if the eyes remain closed for a period of 4.5 seconds the alarm is played


File Edit Selection View Go Run Terminal Help
RUN AND DEBUG
VARIABLES
WATCH
CALL STACK
Running
BREAKPOINTS
□ Raised Exceptions
☑ Uncaught Exceptions
□ User Uncaught Exceptions
drowsiness detection.py
D: > Drowsiness detection
frame
Open Score:0
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Debug Console
Ln 61, Col 45 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit
Search
ENG IN
10:35
27-04-2023

```
import cv2
import os
from keras.mo
import numpy
from pygame i
import time

mixer.init()
sound = mixer

face = cv2.Ca
leye = cv2.Ca
reye = cv2.Ca

lbl=['Close',

model = load_
path = os.get
cap = cv2.Vic
font = cv2.FC
count=0
score=0
thicc=2
rpred=[99]
```

```
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
PS D:\Drowsiness detection>  d:; cd 'd:\Drowsiness detection'; & 'C:\Users\Saurabh Mittal\New folder\python.exe' 'c:\Users\Saurabh Mittal\.vscode\extensions\ms-pytho
n.python-2023.6.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '52334' '--' 'D:\Drowsiness detection\drowsiness detection.py.py'
pygame 2.3.0 (SDL 2.24.2, Python 3.11.3)
Hello from the pygame community. https://www.pygame.org/contribute.html
```


File Edit Selection View Go Run Terminal Help
RUN AND DEBUG
VARIABLES
WATCH
CALL STACK
Running
BREAKPOINTS
□ Raised Exceptions
☑ Uncaught Exceptions
□ User Uncaught Exceptions
drowsiness detection.py
D: > Drowsiness detection
frame
Closed Score:51
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Debug Console
Ln 61, Col 45 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit
Search
ENG IN
10:39
27-04-2023

```
import cv2
import os
from keras.mo
import numpy
from pygame i
import time

mixer.init()
sound = mixer

face = cv2.Ca
leye = cv2.Ca
reye = cv2.Ca

lbl=['Close',

model = load_
path = os.get
cap = cv2.Vic
font = cv2.FC
count=0
score=0
thicc=2
rpred=[99]
```

```
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
```

Result:

Thus, the details of architectural design/framework/implementation along with the screenshots were provided.

4.1 System Description

4.1.1 Eye Detection

Images or the real time video is captured from the camera installed in front of the driver’s face. This video is converted into number of frames. OpenCV face OpenCV-classifier is loaded.Each frame is compared with the pre-defined features of the OpenCV-classifiers. When the features are matched the face is detected and a rectangle is drawn around the face. Using feature extraction we estimate the position of the eyes. By comparing with the OpenCV eye-OpenCV classifier, the eyes are detected and rectangles are drawn around left and right eye. In the system we have used facial landmark prediction for eye detection Facial landmarks are used to localize and represent salient regions of the face, such as: • Eyes • Eyebrows • Nose

- Mouth
- Jawline

Facial landmarks have been successfully applied to face alignment, head pose estimation, face swapping, blink detection and much more. In the context of facial landmarks, our goal is detecting important facial structures on the face using shape prediction methods. Detecting facial landmarks is therefore a two step process:

• Localize the face in the image. • Detect the key facial structures on the face ROI. Localize the face in the image: The face image is localized by Haar feature-based cascade classifiers which was discussed in the first step of our algorithm i.e. face detection. Detect the key facial structures on the face ROI: There are a variety of facial landmark detectors, butall methods essentially try to localize and label the following facial regions:

- Mouth
- Right eyebrow
- Left eyebrow
- Right eye
- Left eye
- Nose7

The facial landmark detector included in the dlib library is an implementation of theOneMillisecond Face Alignment with an Ensemble of Regression Trees paper by Kazemi and Sullivn. This method starts by using:

1. A training set of labeled facial landmarks on an image. These images are manually labeled, specifying specific (x, y)-coordinates of regions surrounding each facial structure. 2. Priors, of more specifically, the probability on distance between pairs of input pixels.The pre-trained facial landmark detector inside the dlib library is used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the face.

4.1.2 Technology Used

TensorFlow: IT is an open-source software library for dataflow programming across a range of tasks. It is a symbolic math library, and is also used for the machine learning applications such as neural networks. It is used for both research and production. TensorFlow computations are expressed as tensor. 2.5.2 Machine learning: Machine learning is the kind of programming which gives computers the capability to automatically learn from data without being explicitly programmed. This means in other words that these programs change their behaviour by learning from data. Python is clearly one of the best languages for machine learning. Python does contain special libraries for machine learning namely scipy, pandas and numpy which great for linear algebra and getting to know kernel methods of machine learning. The language is great to use when working with machine learning algorithms and has easy syntax relatively

OpenCV: OpenCV stands for Open Source Computer Vision. It's an OpenSourceBSD licensed library that includes hundreds of advanced Computer Vision algorithms that are optimized to use hardware acceleration. OpenCV is commonly used for machine learning, 4 image processing, image proc. manipulation, and much more. OpenCV has a modular structure. There are shared and static libraries and a CV Namespace. In short, OpenCV is used in our application to easily load bitmap files that contain landscaping pictures and perform a blend operation between two pictures so that one picture can be seen in the background of another picture. This image manipulation is easily performed in a few lines of code using OpenCV versus other meths.

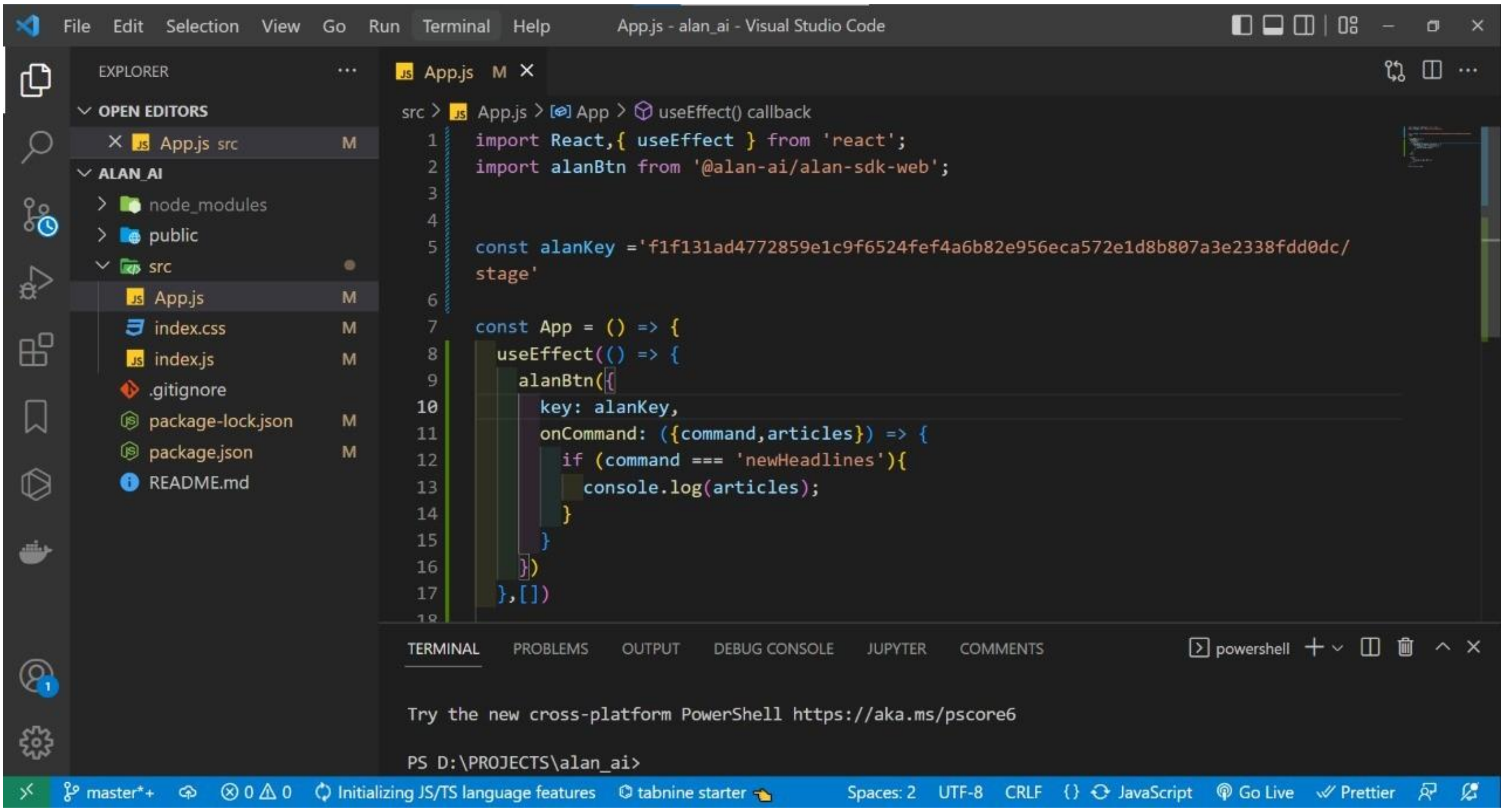
TOOLS AND IMAGE PROCESSING LIBRARIES

Following optimized tools and image processing libraries are used by author for implementation of presented algorithm. Open CV: OpenCV (Open-source Computer Vision) is the Swiss Army knife of computer vision.

It has a wide range of modules that can help us with a lot of computer vision problems. But perhaps the most useful part of OpenCV is its architecture and memory management. It provides you with a framework in which you can work with images and video in any way you want, using OpenCV's algorithms or your own, without worrying about allocating and reallocating memory for your images. Open CV libraries and functions are highly optimized and can be used for real time image and video processing. OPENCV's highly optimized image processing function are used by author for real time image processing of live video feed from camera. DLib: Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge. Open Source Dib library is used by author for implementation of CNN(Neural Networks). Highly optimized Pre-learned facial shape predictor anddetectorsfunctions are used by author for detection of facial landmarks. Facial landmarks were further used for extracting eye coordinates.

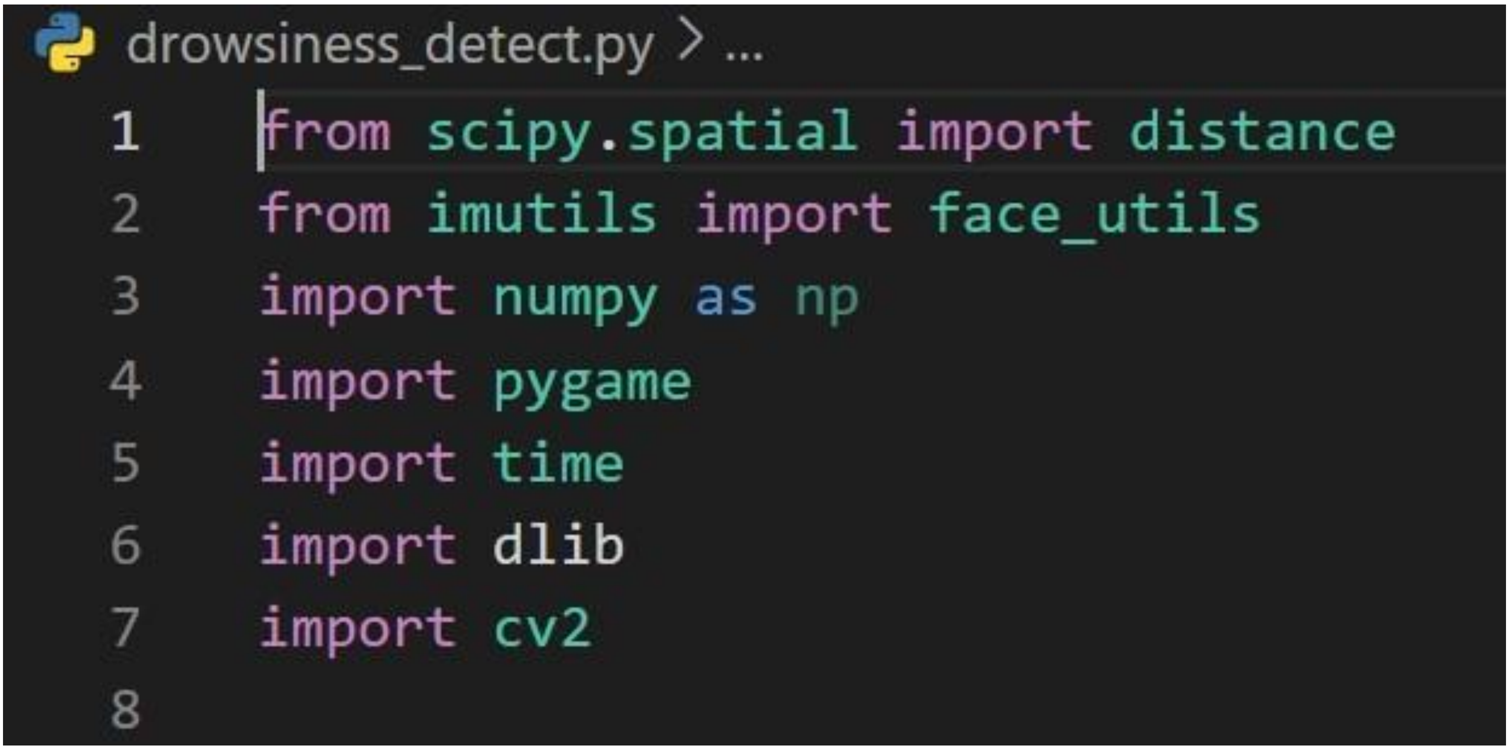
Python: Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping in the complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability. Python language is used by author due to his cross platform compatibility as main coding language for algorithm. OpenCV and Dliblibraries are integrated in python interpreter for using readymade optimized functions.

5 CODING AND TESTING

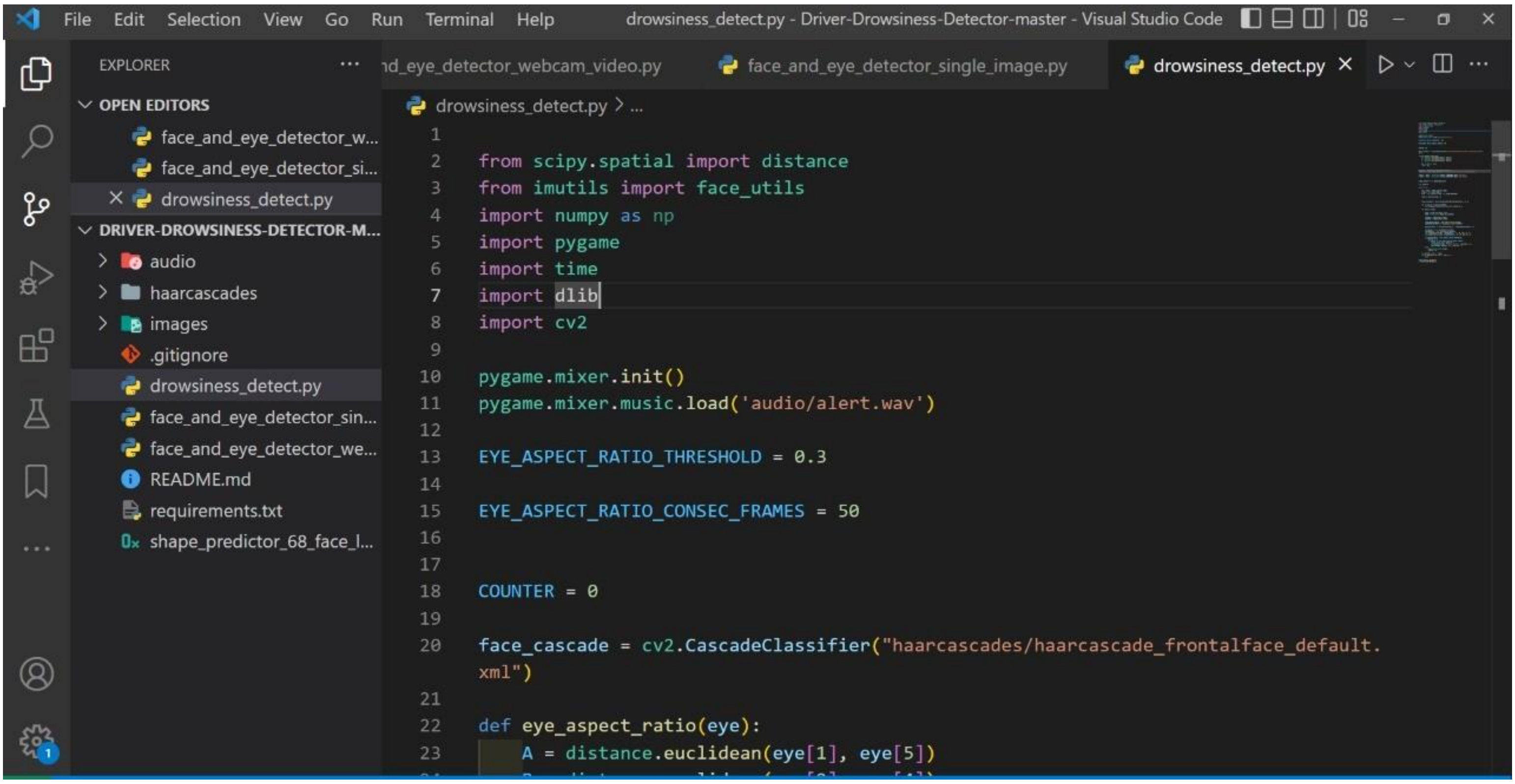


```
import React,{ useEffect } from 'react';
import alanBtn from '@alan-ai/alan-sdk-web';


const alanKey ='f1f131ad4772859e1c9f6524fef4a6b82e956eca572e1d8b807a3e2338fdd0dc/stage'

const App = () => {
  useEffect(() => {
    alanBtn({
      key: alanKey,
      onCommand: ({command,articles}) => {
        if (command === 'newHeadlines'){
          console.log(articles);
        }
      }
    })
  },[])
```

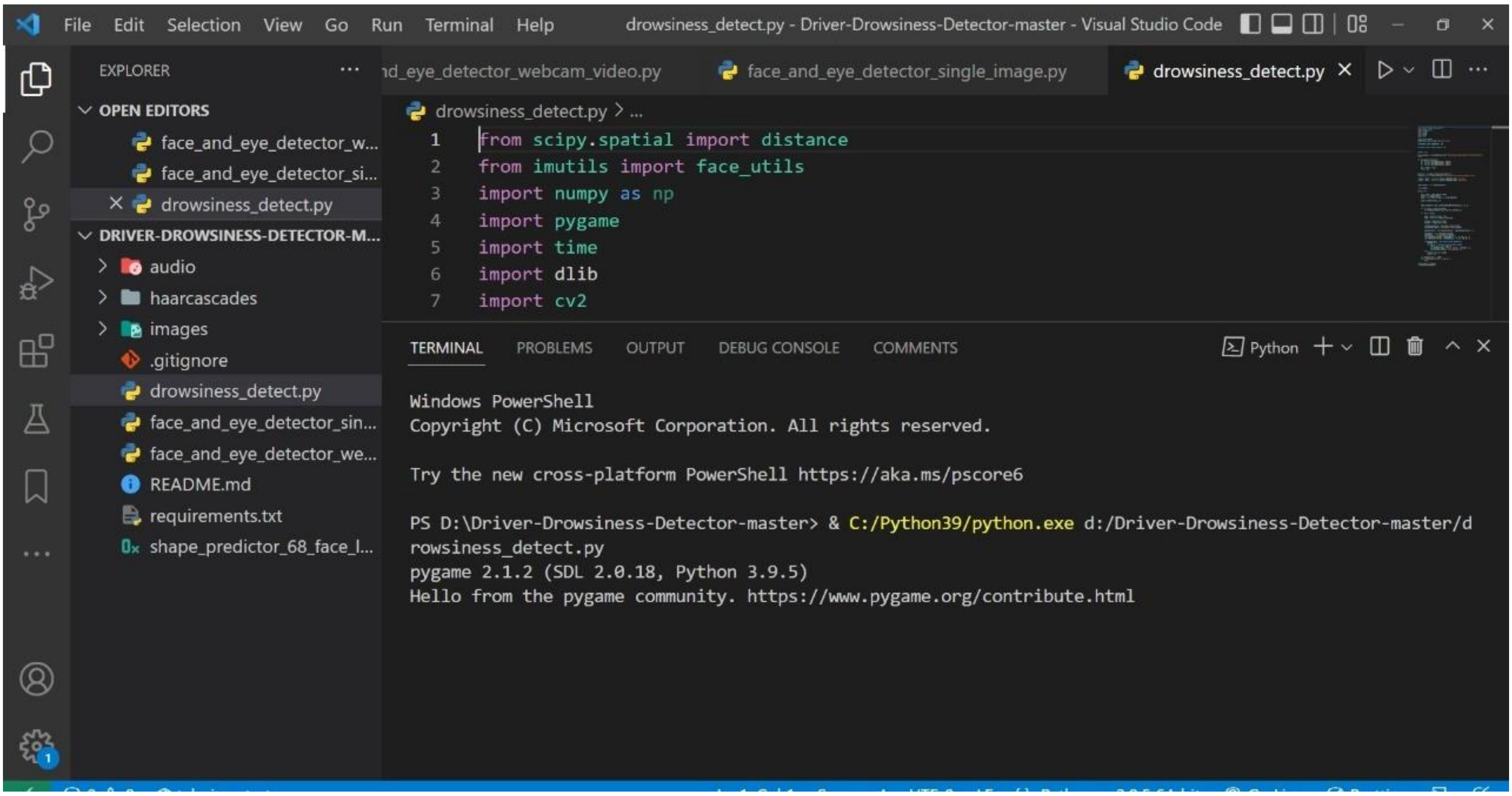
```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\PROJECTS\alan_ai>
```



```python
from scipy.spatial import distance
from imutils import face_utils
import numpy as np
import pygame
import time
import dlib
import cv2

```


File
Edit
Selection
View
Go
Run
Terminal
Help
drowsiness_detect.py - Driver-Drowsiness-Detector-master - Visual Studio Code
EXPLORER
OPEN EDITORS
face_and_eye_detector_w...
face_and_eye_detector_si...
drowsiness_detect.py
DRIVER-DROWSINESS-DETECTOR-M...
audio
haarcascades
images
.gitignore
drowsiness_detect.py
face_and_eye_detector_sin...
face_and_eye_detector_we...
README.md
requirements.txt
shape_predictor_68_face_l...
nd_eye_detector_webcam_video.py
face_and_eye_detector_single_image.py
drowsiness_detect.py
drowsiness_detect.py > ...

```

from scipy.spatial import distance
from imutils import face_utils
import numpy as np
import pygame
import time
import dlib
import cv2

pygame.mixer.init()
pygame.mixer.music.load('audio/alert.wav')

EYE_ASPECT_RATIO_THRESHOLD = 0.3

EYE_ASPECT_RATIO_CONSEC_FRAMES = 50


COUNTER = 0

face_cascade = cv2.CascadeClassifier("haarcascades/haarcascade_frontalface_default.xml")

def eye_aspect_ratio(eye):
    A = distance.euclidean(eye[1], eye[5])
```


File
Edit
Selection
View
Go
Run
Terminal
Help
drowsiness_detect.py - Driver-Drowsiness-Detector-master - Visual Studio Code
EXPLORER
OPEN EDITORS
face_and_eye_detector_w...
face_and_eye_detector_si...
drowsiness_detect.py
DRIVER-DROWSINESS-DETECTOR-M...
audio
haarcascades
images
.gitignore
drowsiness_detect.py
face_and_eye_detector_sin...
face_and_eye_detector_we...
README.md
requirements.txt
shape_predictor_68_face_l...
ıd_eye_detector_webcam_video.py
face_and_eye_detector_single_image.py
drowsiness_detect.py
drowsiness_detect.py > ...
TERMINAL
PROBLEMS
OUTPUT
DEBUG CONSOLE
COMMENTS
Python

```python
from scipy.spatial import distance
from imutils import face_utils
import numpy as np
import pygame
import time
import dlib
import cv2
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Driver-Drowsiness-Detector-master> & C:/Python39/python.exe d:/Driver-Drowsiness-Detector-master/d
rowsiness_detect.py
pygame 2.1.2 (SDL 2.0.18, Python 3.9.5)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

6 CONCLUSION AND FUTURE ENHANCEMENT

This project tells how to build a drowsiness detector using OpenCV, Python, and Dlib opensource Libraries. The first step in building a blink detector is to perform facial landmark detection to localize the eyes in a given frame from a video stream. The eye aspect ratio for each eye can be calculated using Euclidian distance functions of OPEN CV, which is a singular value, relating the distances between the vertical eye landmark points to the distances between the horizontal landmark points. Once the eye aspect ratio calculated, algorithm can threshold it to determine if a person is blinking the eye aspect ratio will remain approximately constant when the eyes are open and then will rapidly approach zero during a blink, then increase again as the eye opens. The duration of blink further provide estimation of microsleep. The proposed algorithm has been tested on personal car driver for testing purposes. For authentic results, the camera position was focused on the driver's face. Further, the algorithm has been tested in day time driving and Night time using IR camera. The results are discussed in Result section and found satisfactory. The proposed algorithm focused solely on using the eye aspect ratio as a quantitative metric to determine if a person has blinked in a video stream. However, due to noise in a video stream, subpar facial landmark detections, or fast changes in viewing angle, a simple threshold on the eye aspect ratio could produce a false-positive detection, reporting that a blink had taken place when in reality the person had not blinked. To make our blink detector more robust to these challenges further following improvements can be implemented Computing the eye aspect ratio for the Nth frame, along with the eye aspect ratios for N - 6 and N + 6 frames, then concatenating these eye aspect ratios to form a 13 dimensional feature vector.

REFERENCES

- “Road traffic injuries,” World Health Organisation, 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ road-traffic-injuries
- G. Li and W.-Y. Chung, “Detection of driver drowsiness using wavelet analysis of heart rate variability and a support vector machine classifier,” Sensors, vol. 13, no. 12, pp. 16 494–16 511,2013. [Online]. Available: https://www.mdpi.com/1424-8220/13/12/16494
- A. M. Williamson and A. M. Feyer, “Moderate sleep deprivation produces impairments in cognitiveand motor performance equivalent to legally prescribed levels of alcohol intoxication,” Occupational and environmental medicine, vol. 57, no. 10, pp. 649–655, 2000. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/10984335https: //www.ncbi.nlm.nih.gov/pmc/articles/ PMC1739867/
- M. D. Mulhall, T. L. Sletten, M. Magee, J. E. Stone, S. Ganesan, A. Collins, C. Anderson, S. W. Lockley, M. E. Howard, and S. M. W. Rajaratnam, “Sleepiness and driving events in shift workers:the impact of circadian and homeostatic factors,” Sleep, vol. 42, no. 6, 2019. [Online]. Available: https://doi.org/10.1093/sleep/zsz074

CARRS-Q, “Sleepiness and fatigue,” 2018. [Online]. Available: https://research.qut.edu.au/carrsq/wp-content/uploads/sites/296/ 2020/12/Sleepiness-and-fatigue.pdf