

DRIVER DROWSINESS DETECTION AND ANALYSIS

A PROJECT REPORT

Submitted by:

JASMINE JOSEPH

LLMC16MCA017

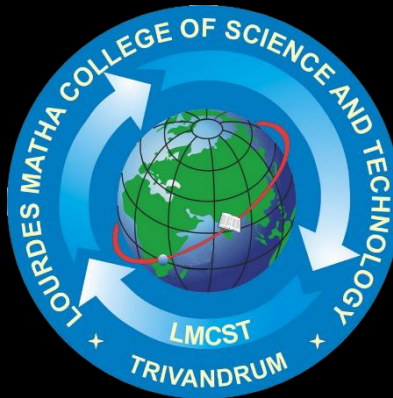
to

The APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications



Department of Computer Applications

**LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY
KUTTICAL, THIRUVANANTHAPURAM 695574**

MAY 2019

DRIVER DROWSINESS DETECTION AND ANALYSIS

A PROJECT REPORT

Submitted by:

JASMINE JOSEPH

LLMC16MCA017

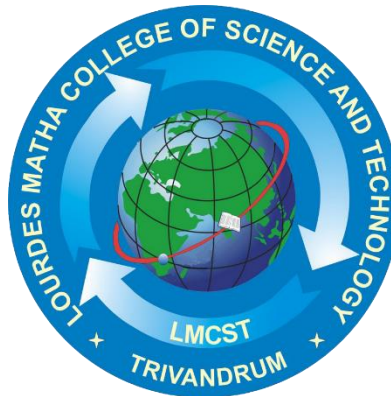
to

The APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications



Department of Computer Applications

**LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY
KUTTICAL, THIRUVANANTHAPURAM**

MAY 2019

DEPARTMENT OF COMPUTER APPLICATIONS
LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY
KUTICAL, THIRUVANANTHAPURAM



CERTIFICATE

This is to certify that the report entitled **DRIVER DROWSINESS DETECTION AND ANALYSIS** submitted by **JASMINE JOSEPH** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by her under my guidance and supervision.

Prof. Bismi K Charleys
(Internal Supervisor)

Prof. Neethu Mohan
(Project Co-ordinator)

Prof. Selma Joseph
(Head of the Dept.)

DECLARATION

I undersigned hereby declare that the project report “DRIVER DROWSINESS DETECTION AND ANALYSIS”, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Application of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Ms. Bismi K Charleys. This submission represents my ideas in my own words and, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University.

Thiruvananthapuram

30/05/19

JASMINE JOSEPH

CONTENTS

Contents	Page No.
ACKNOWLEDGEMENT	i
ABSTRACT	ii
Chapter 1. Introduction	
1.1 General Background	01
1.2 Objective	02
Chapter 2. Literature Survey	
2.1 Study of similar work	03
2.1.1 Existing System	04
Chapter 3. Overall Description	
3.1 Proposed System	05
3.2 Features of Proposed System	05
3.3 Functions of Proposed System	06
3.4 Requirements Specification	06
3.5 Feasibility Analysis	07
3.5.1 Technical Feasibility	08
3.5.2 Operational Feasibility	08
3.5.3 Economical Feasibility	08
3.5.4 Behavioral Feasibility	
Chapter 4. Operating Environment	
4.1 Hardware Requirements	10
4.2 Software Requirements	13
4.3 Tools and Platforms	13
4.3.1 An Overview of Front End	13
4.3.2 An Overview of Back End	14
Chapter 5. Design	
5.1 System Design	15
5.1.1 Activity Diagram and Flow chart	15
5.1.2 Project Diagram	18

5.2. Database Design	21
5.3. Input Design	22
5.4 Output Design	23
5.6 Program Design	23
Chapter 6. Functional and Non-Functional Requirements	
6.1 Functional requirements	25
6.2 Non-Functional Requirements	26
Chapter 7. Testing	
7.1 Testing Strategies	28
7.2 Unit Testing	30
7.3 Integration Testing	30
7.4 System Testing	30
7.5 Testing Results	31
Chapter 8. Results and Discussion	
8.1 Results	33
8.2 Screen Shots	34
Chapter 9. Conclusion	
9.1 System Implementation	40
9.2 Conclusion	42
9.3 Future Enhancement	43
References	I
1. Books	44
2. Websites	44
3. Journals and Publications	44
Appendices	II
1. SCRUM Board	45
2. List of Tables	49
3. List of Figures	50
4. Abbreviations and Notation	51
5. Coding	52

ACKNOWLEDGEMENT

An endeavor over a long can be successful only with advice and support of many well-wishers. I wish to place on record my profound indebtedness and gratitude to all those who have contributed directly or indirectly to make this project work a success. At the very onset, I express my gratitude to God Almighty, who sheltered me under his protective wings and showered on innumerable blessings throughout the period of this Master of Computer Application Course.

It is a great pleasure to express my sincere gratitude to **Rev. Dr.Tomy Joseph Padinjareveettil**, Director and **Prof. Dr. P.P Mohanlal**, Principal, Lourdes Matha College of Science and Technology for permitting to do this project with the fullest spirit. I am highly obliged to **Ms. Selma Joseph**, Head of the Department of Computer Applications, Lourdes Matha College of Science and Technology, for being the source of inspiration throughout the course and for her valuable guidance.

With heart full of thanks, I would like to take up this opportunity to wish my internal guide **Ms. Bismi K Charleys**, Assistant Professor and all staffs of department of computer applications for their endless support, encouragements and suggestions in various stages of the development of this project.

I thank my parents and friends for their moral support and encouragement for the successful completion of this project.

ABSTRACT

Nowadays, more and more professions require long-term concentration. Drivers must keep a close eye on the road, so they can react to sudden events immediately. Driver Fatigue is one of the most common reasons for fatal road accidents around the world. This shows that in the transportation industry especially, where a driver of a heavy vehicle is often exposed to hours of monotonous driving which causes fatigue without frequent rest period. Therefore, there is a need to develop the systems that will detect and notify a driver of her/him bad psychophysical condition, which could significantly reduce the number of fatigue-related accidents. In this method an efficient driver's drowsiness detection system is designed using yawn detection by taking eye detection into consideration so that road accidents can be avoided successfully. However, the development of such systems encounters many difficulties related to fast and proper recognition of a driver's fatigue symptoms. One of the technical possibilities to implement driver drowsiness detection systems is to use the vision-based approach. In the developed system, a webcam records the video and driver's face is detected in each frame employing image processing techniques. The camera must be placed at a distance of 40 cm to 50 cm from a driver. The camera must be placed at an angle of 45 degrees from the driver's face. At an angle between 35-50 degrees the face can be captured with perfection and ease. Speakers are used for the proposed system as it generates the output in terms of an alarm.

And it also includes an additional feature, which is driver alcohol detection. This system also really helps the detection of alcoholic intoxication. Computer vision, alcohol gas sensor application is combined to an embedded system to achieve this goal. This is achieved by integrating alcohol sensor with Arduino board. Arduino processor ATmega328 is able to handle more functions than conventional microcontrollers. The alcohol sensor used in this part is MQ3 which to detect the alcohol content in human breath. Since sensor has fine sensitivity range around 2 meters, it can suit to any vehicle and can easily be hidden from the suspects. This is fitted inside the vehicle. The project is designed for the safety of people sitting inside the vehicle. If the alcohol is detected, the information will send to the Motor Vehicle Department and Vehicle owner.

CHAPTER 1

INTRODUCTION

1.1 GENERAL BACKGROUND

Driver fatigue is a significant factor in a large number of vehicle accidents. Recent statistics estimate that annually 1,200 deaths and 76,000 injuries can be attributed to fatigue related crashes. Automotive population is increasing exponentially in the country. The biggest problem regarding the increased traffic is the raise in number of road accidents. Road accidents are undoubtedly a global menace in our country. The global status report on road safety published by the World Health Organization (WHO) identified the major causes of road accidents are due to driver errors and carelessness. Driver sleepiness, alcoholism and carelessness are the key players in accident scenario. The fatalities and associated expenses as a result of road accidents are very serious problems.

The aim of this project is to develop a prototype drowsiness detection system. The focus will be placed on designing a system that will accurately monitor the open or closed state of the driver's eyes in real-time. By monitoring the eyes, it is believed that the symptoms of driver fatigue can be detected early enough to avoid a vehicle accident. Detection of fatigue involves a sequence of images of a face, and the observation of eye movements and blink patterns. And also develop a prototype alcohol detection system. It is useful to detect the driver alcoholic intoxication while on driving.

1.2. OBJECTIVE

The aim of this thesis is to contribute to the study of driver behavior while driving, through the development and evaluation of a drowsiness driver model system. Non-intrusive is chosen as a method due to comfort to the drivers. The result from the research will be integrated to produce the systems that can be efficient in detecting the drowsiness level at an early stage by giving a warning to them about their lack of attention due to drowsiness or other factors. Driver drowsiness detection is a safety technology which helps to save the life of the driver by preventing accidents when the driver is getting drowsy. The main objective is to first design a system to detect driver's drowsiness by continuously monitoring retina of the eye. The system works in spite of driver wearing spectacles and in various lighting conditions. To alert the driver on the detection of drowsiness by using buzzer or alarm. And also helps to identify the driver alcoholic intoxication.

CHAPTER 2

LITERATURE SURVEY

2.1 STUDY OF SIMILAR WORKS

In 2013, Osama Rizwan and Hamza Rizwan described “**Development of an efficient system for vehicle accident warning**”. They proposed an efficient accident avoidance system has been a great need since the invention of motor vehicles. We propose a vehicle accident warning system based on image processing techniques. It uses two criteria for enabling accident warning. These are driver drowsiness and front vehicle distance from our vehicle. Drowsiness, especially in long distance journeys, is a key factor in traffic accidents. We use visually observable facial behaviors as indicators of driver drowsiness. For face tracking, we use Viola-Jones face detection algorithm. The eyes region is found using a novel approach to check whether they are close or open. Other image processing techniques are used to calculate distance of front vehicles. Along with this software design, an easy to build hardware is used to complete the said system to be used in real time. To evaluate the effectiveness of proposed system, a number of drowsy persons were tested and evaluated. Experimental results show high accuracy in each section which makes the system efficient and reliable for accident warning.

In 2014, Syed Imran Ali, Prashant Singh and Sameer Jain described “**An efficient system to identify user attentiveness based on fatigue detection**”. They proposed a user alertness identification system based on fatigue detection. The user may be a person like computer operator, operating critical operations on distant machines, hands free interaction with computational devices/machines, controlling heavy machineries like cranes or performing time critical operations like air traffic controlling etc. Same system can also be employed to detect and notifying the driver

vigilance level and hence to avoid possibility of road accidents. The proposed system uses the time-efficient image processing techniques to measure eyes closer count, blinking rate of eye and user yawning as the parameters to conclude user's fatigue. The proposed system continuously captures the image of the subject on site using web camera and detects face region, then focuses on eyes and lips using efficient image processing techniques to monitor their behavior. If abnormality either in behavior of eyes or mouth is detected, it indicates that the subject is falling asleep or having state of drowsiness therefore fatigue is detected and a warning alarm is generated.

2.1.1 Existing System

Recently most of the accident occur due to drowsiness of drivers in cars and trucks or In other words, drowsiness or a fatigue of a driver is the major reason for road accidents. Annually 1200 deaths and 76000 injured. Therefore, there is a need to develop the systems that will detect and notify a driver of her/him bad psychophysical condition, which could significantly reduce the number of fatigue-related car accidents. This approaches include analysis of police reported crash data, in-depth on-site investigations immediately following a crash of the general driving population.

Currently there is no existing system, only paper works are established.

- There is no way to control the road accidents and alerting him on real time by driver fatigue.
- There is no real time detection of alcoholic intoxication of driver and send information to Motor Vehicle Department.

CHAPTER 3

OVERALL DESCRIPTION

3.1 PROPOSED SYSTEM

The proposed system is to develop nonintrusive system which will detect the fatigue or drowsiness of driver and will issue a warning with the help of alarm. As most of the accidents are caused due to drowsiness so this project will help to decrease the crashes or accidents. In this project we will detect the eye blinking with the help of webcam. If the eyes of the person are closed for more interval of time then this will result into the warning in the form of sound. An additional feature is added that is, alcohol detection system. It is useful to detect the driver alcoholic intoxication while on driving.

3.2 FEATURES OF PROPOSED SYSTEM

- Warns the driver of drowsiness and the risk of a micro sleep.
 - Compliance with driver warnings helps to avoid crashes caused by fatigue.
 - The driver drowsiness monitoring using Eye detection.
 - The additional feature of this system also real helps the detection of alcoholic intoxication.
- Computer vision, alcohol gas sensor application is combined to an embedded system to achieve this goal.
- If the alcohol is detected, the information will send to the Motor Vehicle Department and Vehicle owner.

3.3 FUNCTIONS OF PROPOSED SYSTEM

- Drowsy detection
- Real time warning alert to driver
- Avoid crashes
- Controlling vehicle accidents caused by driver fatigue
- Alcohol detection
- Send information to Motor Vehicle Department if driver is alcoholic.

3.3 REQUIREMENT SPECIFICATION

System analyst talk to a variety of persons to gather details about the business process and their opinions of why things happen as they do and their ideas for changing the process. These can be done through questionnaire', detailed investigation, observation, collection of samples etc. As the details are collected, the analyst study the requirements data to identify features the new system must have, including both the information the system should produce and operational features such as processing controls, response times and input-output methods.

Requirements specification simply means, "Figuring out what is to be made before making it." It determines what people need before starting to develop a product for them. Requirement definition is the activity of translating the information gathered in to a document that defines a set of requirements.

The requirements for an effective drowsy driver detection system are as follows:

- A non-intrusive monitoring system that will not distract the driver.
- A real-time monitoring system, to insure accuracy in detecting drowsiness.

- A system that will work in both daytime and nighttime conditions.
- Additionally, an alcohol sensor MQ-3 is used to detect the driver alcohol intoxication.

The above requirements are subsequently the aims of this project. The project will consist of a concept level system that will meet all the above requirements.

3.4 FEASIBILITY ANALYSIS

The initial investigation points to be question whether the project is feasible. The feasibility study concerns with the considerations made to verify whether the system fit to be developed in all terms. Once the idea to develop software is put forward, the question that rises first will pertain to be the feasibility aspects. The prime objective of feasibility study is to ensure that the problem is worth to be solved. At the stage a cost benefit analysis is performed to assertion that the benefit from the system will over rule the cost association with the whole analysis, design and development of the new system. An important outcome of the preliminary investigation determining whether the system required is feasible.

3.4.1 Technical Feasibility

The main objective of feasibility study is to test the technical, social and economic feasibility of developing a system. Investing the existing system in the area under investigation and generating ideas about the new system does this. Feasibility study has been done to gather required information. Training, experience and commonsense are required for collection of the information. Data was gathered and checked for completeness and accuracy. Analyzing the data involved identification of the components of the system and their interrelationship and identified the strength and weakness of the system.

My system is developed by using front end as Java and back end as MS SQL. The font end is technically feasible and it has lot of features. So the technical part of this project is very secure. So my system is technically feasible.

3.4.2. Operational Feasibility

There is no difficulty in implementing the system. The proposed system is effective, user friendly and functionally. The user of the system must be completely unaware of the internal working of the system so that the users will not face any problem running the system. The system thus reduces the responsive time of computer thereby, the system is found to be operationally feasible.

Design is the only ways that can accurately translate user registration into finished software or system. Without software design, the risk of building an unstable system exists. System design provides the procedural details necessary for implementing the system recommended in the feasibility study.

3.4.3 Economical Feasibility

Economic and Financial analysis is used for evaluating the effectiveness of the system. The project is technically and operationally feasible.

MS SQL is a free software so no cost is needed to buy the backend. My system is economically feasible because the project completed in few months. So less resources are used.

3.4.4 Behavioral feasibility

The behavioral feasibility depends upon whether the system performed in the expected way or not. Feasibility study is a test of system proposal according to it workability, impact on

organization, ability to meet the user's need and effective use of resources. However, a feasibility study provides a useful starting point for full analysis.

My system is behaviorally feasible because of the effective use of the resources and also the system satisfied the user needs and the system is user friendly.

CHAPTER 4

OPERATING ENVIRONMENT

4.1 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application.

System	: PENTIUM IV 2.4 GHZ
Hard Disk	: 250 GB
Floppy Drive	: 1.44 MB
Monitor	: 15 VGA COLOUR
Mouse	: LOGITECH.
Keyboard	: 110 KEYS ENHANCED
RAM	: 1 GB
External Hardware	: Arduino UNO, Alcohol Sensor-MQ3

4.1.1 Arduino UNO

The arduino board is an open source programmable circuit board. The arduino uno is the microcontroller board based on the ATmega 328. It is a programmable microcontroller for prototyping electromechanical devices. it has 14 digital inputs/output pins (of which 6 can be used as PWM output), 6 analog inputs, a 16 MHz ceramic resonator. The arduino differs from all preceding boards in that it does not use the FTDI USB to serial driver chip. Arduino that can

be integrated into a wide variety of makerspace projects both simple and complex. This board contains a microcontroller which is able to be programmed to sense and control objects in the physical world. By responding to sensors and inputs, the Arduino is able to interact with a large array of outputs such as LEDs, motors and displays. Because of its flexibility and low cost, Arduino has become a very popular choice for makers and makerspaces looking to create interactive hardware projects.



4.1.2 Alcohol Sensor (MQ3)

The analog gas sensor- MQ3 is suitable for alcohol detecting, this sensor can be used in a breath analyzer. It has a high sensitivity to alcohol and small sensitivity to benzene. The sensitivity can be adjusted by the potentiometer sensitive material of MQ3 gas sensor is SnO_2 , which with lower conductivity in clean air. When the target alcohol gas exist, the sensors conductivity is higher along with the gas concentration rising, use of simple electro circuit, convert change of conductivity to correspond output signal of gas concentration.



4.2 SOFTWARE REQUIREMENTS

A software requirements specification (SRS) is a description of a software system to be developed. The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view.

Software	: JAVA
Operating System	: Windows 95/98/2000/NT4.0.
Application Server	: Apache Tomcat
Language	: Java.
Database	: MS SQL

4.3 TOOLS AND PLATFORMS

4.3.1 An Overview of Front End

i. Java

Java is a high-level language with which to write programs that can execute on a variety of platforms. Part of Java's novelty arises from its new approach to portability. In previous High-level languages, the portable element was the source program. Once the source program is compiled into executable form for a specific instruction set architecture and bound to a library of hardware-dependent I/O, timing and related operating system services, portability is lost. The resultant executable form of the program runs only on platforms having that specific ISA and OS. Thus, if a program is to run on several different platforms, it has to be recompiled and

relinked for each platform. And if a program is sent to a remote target for execution, the sender must know in advance the exact details of the target to be able to send the correct version. With Java, source statements can be compiled into *machine-independent*, "virtual instructions" that are interpreted at execution time.

4.3.2 An Overview of Back End

ii. MS SQL

MS SQL Server is a relational database management system (RDBMS) developed by Microsoft. This product is built for the basic function of storing retrieving data as required by other applications. It can be run either on the same computer or on another across a network. This tutorial explains some basic and advanced concepts of SQL Server such as how to create and restore data, create login and backup, assign permissions, etc. Each topic is explained using examples for easy understanding.

CHAPTER 5

DESIGN

5.1 SYSTEM DESIGN

System design involves translating system requirements and conceptual design into technical specifications and general flow of processing. After the system requirements have been identified, information has been gathered to verify the problem and after evaluating the existing system, a new system is proposed.

System design is the process of planning of new system or to replace or complement an existing system. It must be thoroughly understood about the old system and determine how computers can be used to make its operations more effective.

5.1.1 Activity Diagram and Flow Chart

i. Activity Diagram

An activity diagram is a UML behavior diagram that represents the workflow of stepwise activities of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. UML models basically three types of diagrams, namely, structure diagrams, interaction diagrams, and behavior diagrams. An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram is used by developers to understand the flow of programs on a high level.

➤ Components of Activity Diagram



Start Point/initial state



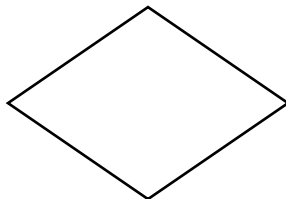
Activity



Action flow



Class/object



Decision/branching

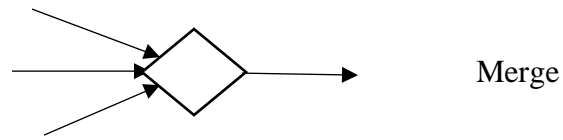
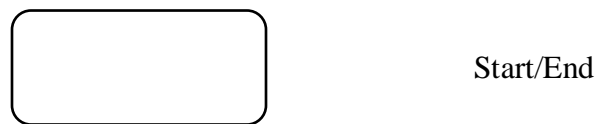


Fig 5.1.1 Components of activity diagram

ii. Flowchart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

➤ Components of Flowchart



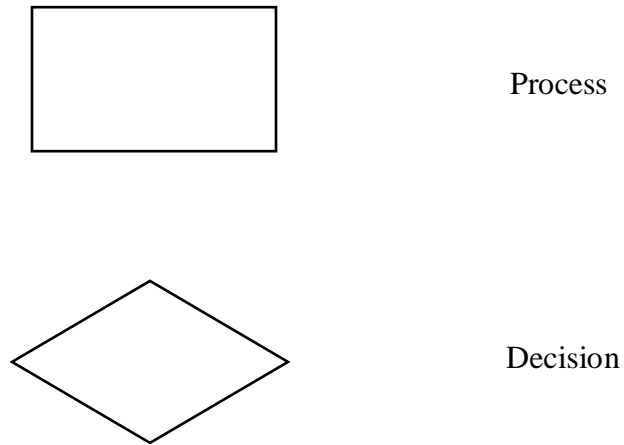
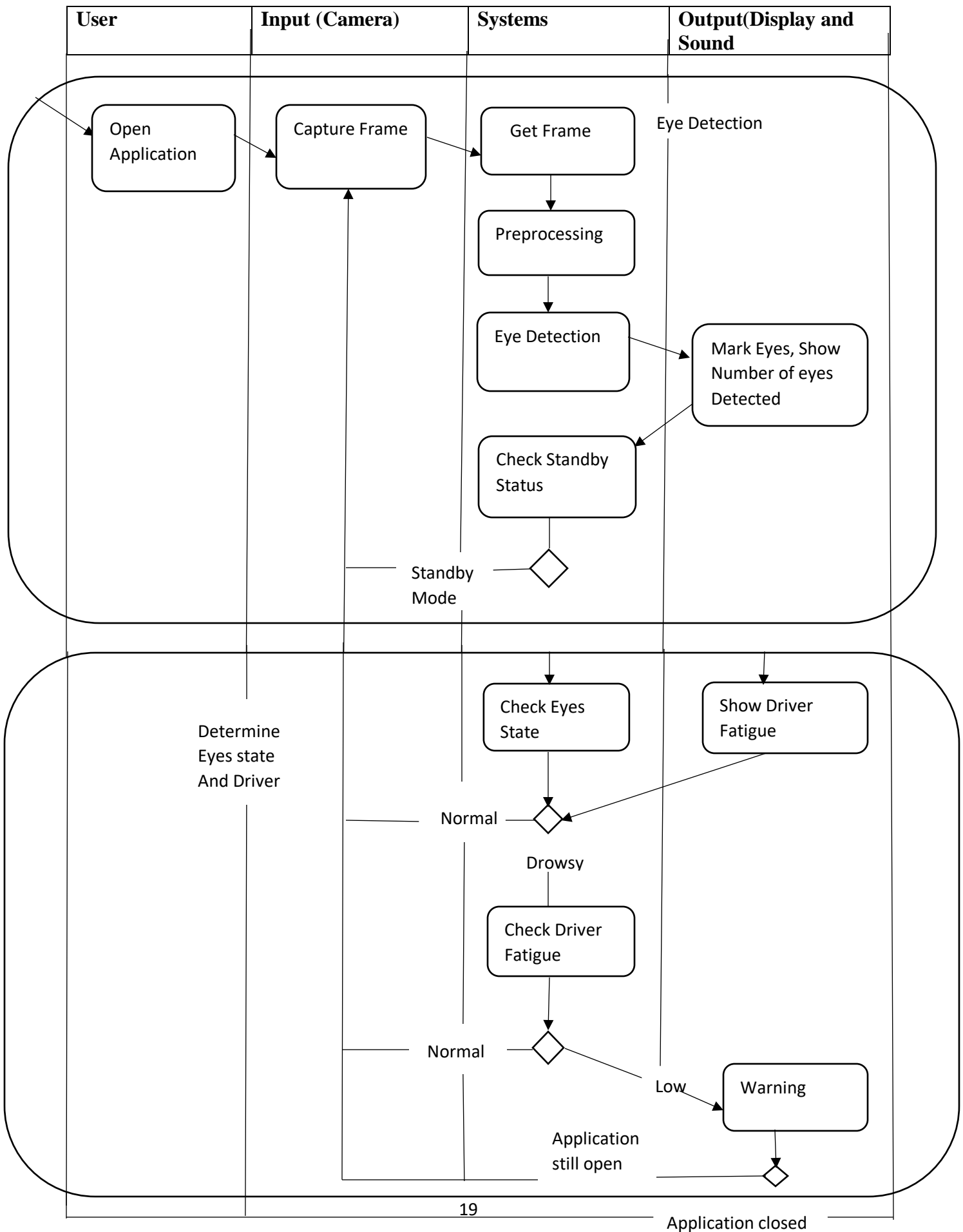


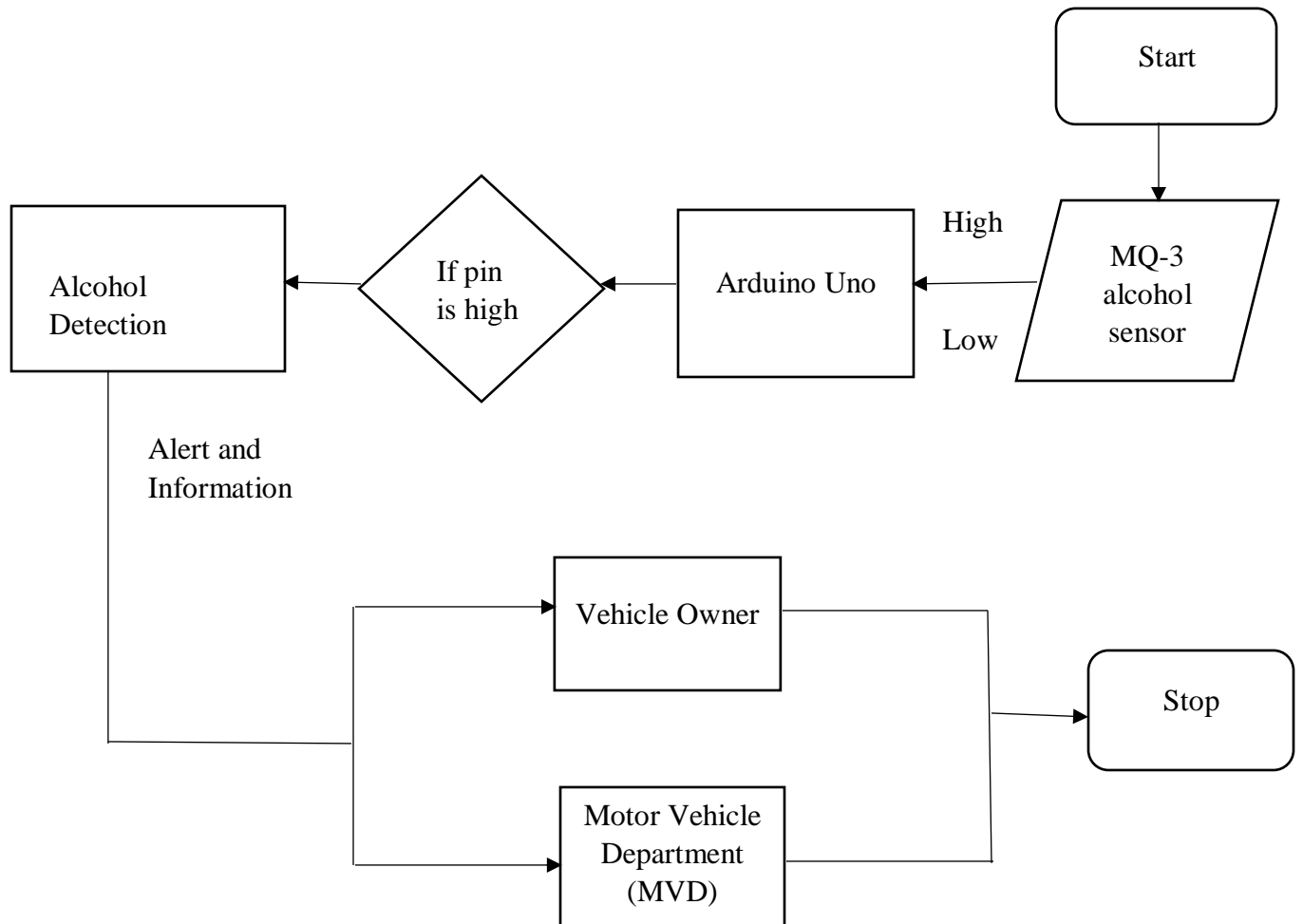
Figure 5.1.2 Components of flow chart

5.1.2 Project Activity Diagram and Flow Chart

- i. Fig 5.1.2.1 Activity Diagram- Driver Drowsiness Detection



ii. Fig 5.1.2.2 Flow Chart- Driver Alcohol Detection



5.2 DATABASE DESIGN

The most important aspect of building software systems is database design. The highest level in the hierarchy is the database. It is a set of inter-related files for real time processing. It contains the necessary data for problem solving and can be used by several users accessing data concurrently. The general objective of database design is to make the data access easy, inexpensive and flexible to the user. Database design is used to define and then specify the structure of business used in the client/server system. A business object is nothing but information that is visible to the users of the system.

Database Name: WebCam

Table Name : Image

Description : Stores the image of driver

Table 5.2.1 – Image table

Sl.no	Fieldname	Datatype	Constraints	Description
1.	id	int	Primary key	Image id
2.	vehicle number	nvarchar(MAX)		Vehicle number
3.	message	nvarchar(MAX)		The alert message
4.	image	varbinary(MAX)		Driver image

5.4 INPUT DESIGN

The input design is the process of converting the user oriented inputs in to the computer based format. The goal of designing input data is to make automation as easy and free from errors as possible. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right message and help for the user at right time are also considered for the development of the project. The input design is the link between the information system and the user.

➤ **Hardware inputs are:**

- Camera : Image Processing and Feature Extraction.
- Alcohol Sensor and Camera : Detection of driver's alcohol intoxication and capture the driver image.

➤ **Input form:**

- Login page : System administrator can activate the system with Username, Password and Vehicle number.

5.4 OUTPUT DESIGN

The output generally refers to the results and information that are generated by the system. A major form of the output is the display of the information generated by the system and servicing the user requests to the system. In this project the necessary outputs are;

➤ The device outputs are:

- Alarm : It will alert the driver when the drowsiness is detected.
- Notification : It will send a notification to (vehicle number and driver image) to Motor Vehicle Department and the vehicle owner if the driver is intoxicated.

5.4 PROGRAM DESIGN

i. Camera

Step 1: If the application is active, the camera will detect or capture the image.

Step 2: The captured frame will be preprocessed for the eye detection.

Step 3: If eye is not detected, then go to step2.

Step 4: Otherwise, the number of eyes will detect and calculate the blinking states.

Step 5: Check the stand by status. If not detected, go to step 2.

Step 6: If the blinking state is correct, then check the eye curve.

Step 7: Both are normal again go to step 2. Otherwise, the drowsiness is detected and sounds the alarm.

ii. Alcohol Sensor(MQ-3)

Step 1: If the application is active, the alcohol sensor will detect the alcohol content.

Step 2: Then the camera will capture the driver image.

Step 3: And the information will send to both Motor Vehicle Department (MVD) and the
Vehicle owner.

Step 4: Otherwise, go to step1.

CHAPTER 6

FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

6.1 FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Generally, functional requirements are expressed in the form “system must do requirement”. Functional requirements for each of the uses cases described below:

- Descriptions of data to be entered into the system
- Descriptions of operations performed by each screen
- Descriptions of work-flows performed by the system
- Descriptions of system reports or other outputs
- Who can enter the data into the system
- How the system meets applicable regulatory requirements

6.2 NON-FUNCTIONAL REQUIREMENTS

In addition to the obvious features and functions that you will provide in your system, there are other requirements that don't actually DO anything, but are important characteristics nevertheless. These are called "non-functional requirements" or sometimes "Quality Attributes." For example, attributes such as performance, security, usability, compatibility are not a "feature" of the system, but are a required characteristic. You can't write a specific line of code to implement them, rather they are "emergent" properties that arise from the entire solution. The specification needs to describe any such attributes the customer requires. You must decide the kind of requirements that apply to your project and include those that are appropriate.

Some of the non-functional requirements are mentioned below

i. **Performance requirements**

Requirements about resources required, response time, transaction rates, throughput, benchmark specifications or anything else having to do with performance.

ii. **Operating constraints**

List any run-time constraints. This could include system resources, people, needed software, etc.

iii. **Platform constraints**

Discuss the target platform. Be as specific or general as the user requires. If the user doesn't care, there are still platform constraints.

iv. **Accuracy and Precision**

Requirements about the accuracy and precision of the data. Beware of 100% requirements; they often cost too much.

v. **Modifiability**

Requirements about the effort required to make changes in the software. Often, the measurement is personnel effort (person- months).

vi. **Portability**

The effort required to move the software to a different target platform. The measurement is most commonly person-months or % of modules that need changing.

vii. **Reliability**

Requirements about how often the software fails. The measurement is often expressed in MTBF (mean time between failures). The definition of a failure must be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error detection, and a strategy for correction.

viii. **Security**

One or more requirements about protection of your system and its data. The measurement can be expressed in a variety of ways (effort, skill level, time) to break into the system. Do not discuss solutions (e.g. passwords) in a requirements document.

ix. **Usability**

Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

CHAPTER 7

TESTING

7.1 TESTING STRATEGIES

An engineered product can be tested in one of the two ways. These testing strategies include:

- **Black box Testing**
- **White box Testing**

These testing strategy checks the correctness of every statement in the program and results in execution of every instruction in the program module.

7.1 Black box testing:

Knowing the specified function that a product has been designed to perform, test can be conducted that each function is fully operational. Black box test is carried out to test that input function is properly accepted and output 3 correctly produced. This test examines some aspects of system with little regard for the internal structure of the software.

In our project we use black box testing, we have no clear idea about the coding and we just test the external structure. That is we just try to test that the correct output will come based on the input we give.

Errors found through black box testing are:

- Incorrect or missing function.
- Interface errors
- Errors in database structure or external database access.
- Performance errors.

- Initialization and termination errors.

7.2 White box testing

White box test of software is predicted on a close examination of procedural detail. The status of the project may be tested at various points to determine whether the expected or asserted status is corresponding to the actual status. We use white box testing, because in this testing all the internal functions and operations are tested. In our project, we have three modules. We separately test these three modules and its internal structure.

Using these following test cases can be derived:

- Exercise all logical conditions on their true or false side.
- Exercise all loops within their boundaries and their operation bounds.
- Exercise internal data structure to ensure their validity.

7.2 UNIT TESTING

Unit testing focuses on verification effort on the smallest limit of software design. Using the unit test plan prepared in the design phase of the system, important control paths are tested to uncover the errors within the module. This testing was carried out during the coding itself. In this testing step each module is going to be working satisfactorily as the expected output from the module.

7.3 INTEGRATION TESTING

It is the systematic technique for constructing the program structure to uncover errors associated with the interface. The objective is to take unit-tested module and built the program structure that has been dictated by design. All modules are combined in this step. Then the entire program is tested as a whole. If a set of errors is encountered connection is difficult because the isolation of causes is complicated by vastness of the entire program. Using this test plan preparing the design phase of the system, the integration was carried out. All the errors found in the system were corrected for the next testing step.

7.4 SYSTEM TESTING

In system testing, the software and other system elements are tested as a whole. System testing is actually a series of different test whose primary purpose is to fully exercise the computer-based system.

7.5 TESTING RESULTS

7.5.1 Test results

Sl No	Test Description	User	Input	Expected Output	Actual Output	Result
1.	Application is open	Driver	Camera	Get frame	Captured frame	Pass
2.	Image processing	Driver	Camera	Get binary image	Binary image	Pass
3.	Face detection	Driver	Camera	Get actual image	Actual image	Pass
4.	Feature Extraction	Driver	Camera	Divided into 4 quadrants	Upper and lower quadrants	Pass
5.	Feature Extraction	Driver	Camera	Get upper quadrants	Eyes	Pass
6.	Feature Extraction	Driver	Camera	Get upper quadrants	Lips	Pass
7.	Eye detection	Driver	Camera	Bezier curve of left eye	Bezier curve of left eye detected	Pass
8.	Eye detection	Driver	Camera	Bezier curve of right eye	Bezier curve of right eye detected	Pass
9.	Eye detection	Driver	Camera	Bezier curve of right eye	Bezier curve of right eye detected	Pass
10.	Mouth detection	Driver	Camera	Bezier curve of mouth	Bezier curve of Mouth Mouth detected	Pass
11.	Drowsiness detection	Driver	Camera	Sounds alarm	Alarm	Pass

12.	Drowsiness detection	Driver	Camera	Sounds alarm	No sound	Fail
13.	Device login	Admin	Username, Password, Vehicle number	Device on/ Successful login	Device on/ Successful login	Pass
14.	Device login	Admin	Username, Password, Vehicle number	Device on/ Successful login	Device off	Fail
15.	Alcohol detection, image capturing	Driver	Camera	Capture image and stored in db	Capture image and stored in db	Pass
16.	Alcohol detection, face detection	Driver	MQ-3 alcohol sensor	Capture image and stored in db	Image not captured	Fail
17.	Alcohol detection	Driver	MQ-3 alcohol sensor	Alert and information will send to both Motor Vehicle Department (MVD) and vehicle owner	Alert and information will send to both Motor Vehicle Department (MVD) and vehicle owner	Pass
18.	Alcohol detection	Driver	MQ-3 alcohol sensor	Alert and information will send to both Motor Vehicle Department (MVD) and vehicle owner	No alert and no information will passed	Fail

CHAPTER 8

RESULTS AND DISCUSSION

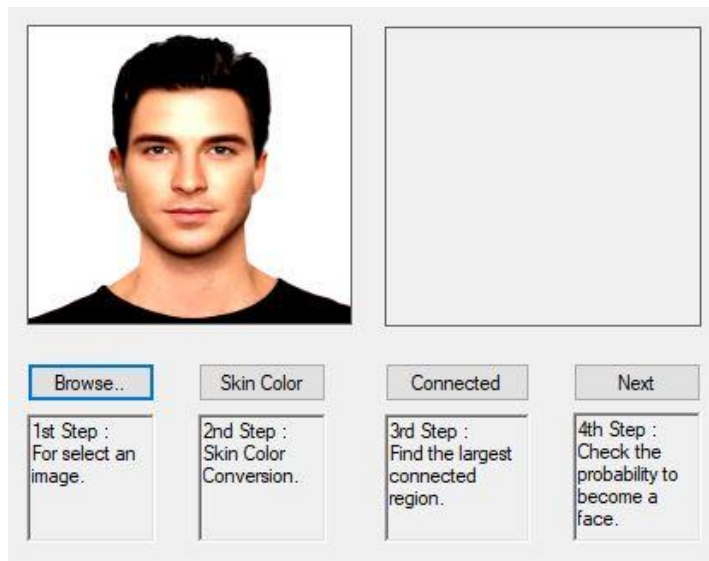
8.1 RESULTS

- Warns the driver of drowsiness and the risk of a micro sleep.
- Compliance with driver warnings helps to avoid crashes caused by fatigue.
- The driver drowsiness monitoring using Eye detection.
- The detection of alcoholic intoxication. The alcohol sensor used in this part is MQ3 which to detect the alcohol content in human breath.
- If the alcohol is detected, the information will send to the Motor Vehicle Department and Vehicle owner

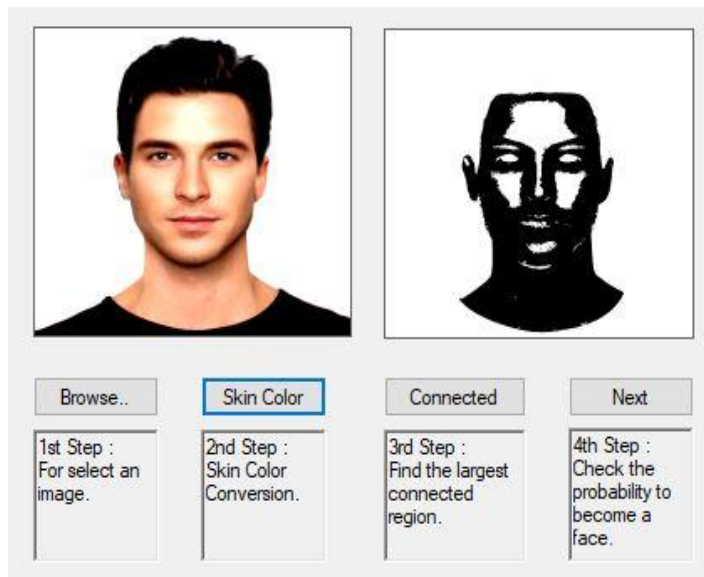
8.2 SCREENSHOTS

Drowsiness detection

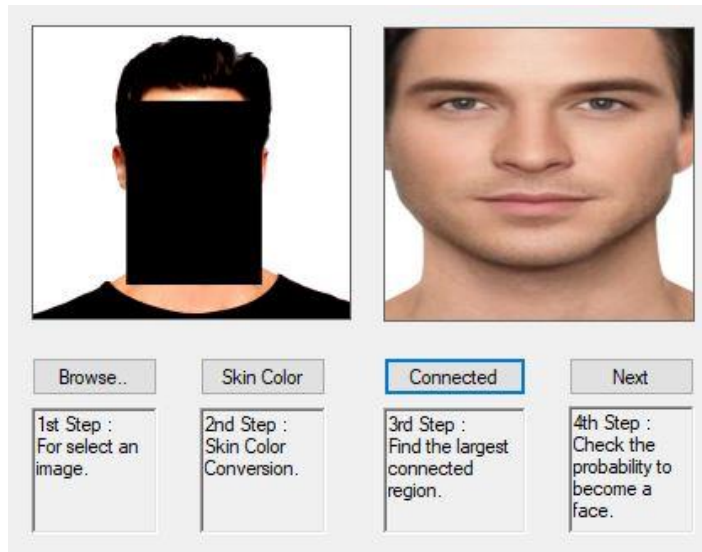
- i. Captured image



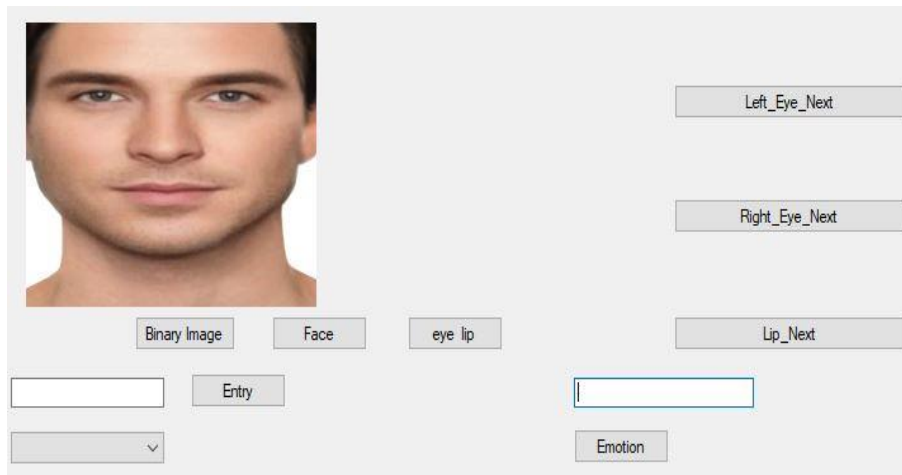
- ii. Convert into gray scale image



iii. Detection of largest connected region



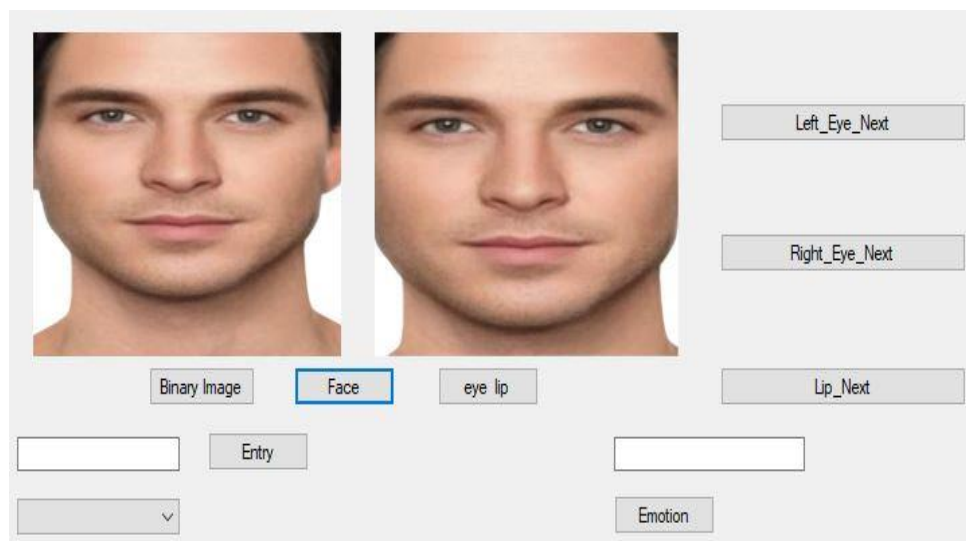
iv. Check the Probability to become a face



v. Get binary image



vi. Face detection



vii. Feature extraction



viii. Get Bezier curve



Alcohol Detection

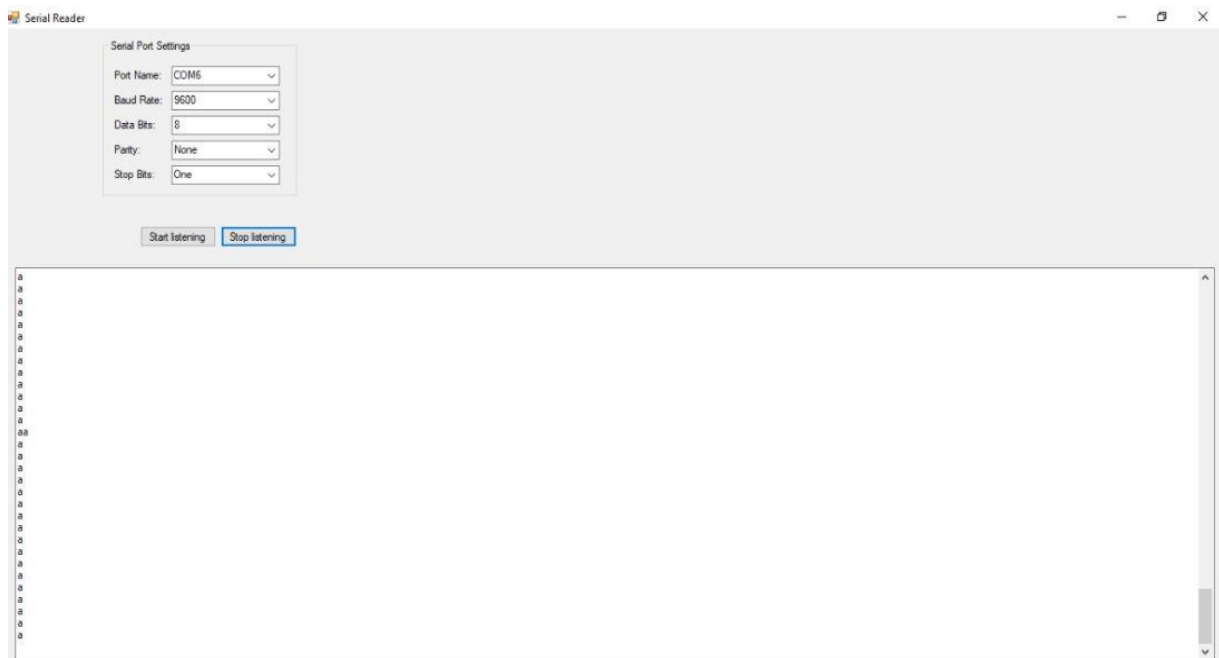
i. Device login

The screenshot shows a window titled "Login Page" with a light blue background. In the top right corner, there are two buttons labeled "MVD" and "Owner". In the center of the window, there are three input fields labeled "User Name", "Password", and "Vehicle Number". Below these fields are two buttons labeled "Login" and "Clear".

ii. Serial reader

The screenshot shows a "Serial Port Settings" dialog box. It contains five dropdown menus: "Port Name" (set to COM3), "Baud Rate" (set to 9600), "Data Bits" (set to 8), "Parity" (set to None), and "Stop Bits" (set to One). Below these settings are two buttons labeled "Start listening" and "Stop listening".

iii. Alcohol Detection



iv. Image Capturing



CHAPTER 9

CONCLUSION

9.1 SYSTEM IMPLEMENTATION

Implementation means converting a new design into iteration .During implementation there should be a strong interaction between the developer of the software and the users. Implementation involves installing hardware terminals and training the operating staff.

In this phase, user training is critical for minimizing reluctance to change and giving the new system a chance to prove its worth. The new system may be totally new replacing the existing system, or it may be the modifications of existing system. In either case proper implementation is essential to provide a reliable system to meet organizational requirements.

Major steps in the implementation of the system are as follows:

Installation of the hardware required for “DRIVER DROWSINESS DETECTION AND ANALYSIS”, is required special hardware (Arduino UNO, MQ-3Alcohol sensor), which is to be installed for the working of both software and hardware.

This can be worked on any Server Configured PC's, which works on Wamp Server with a Oracle and MS SQL. Since the application is being developed with a server configured machine with a machine language, it is necessary to change address from local host to unique domain name.

This case study is comparatively easy to implement. There are some technical risks in hardware. The specific hardware is used for detecting eye and alcohol. As a prerequisite you have to study the server scripting (Java) and working of web servers (apache). Using separate include files/classes for manipulating data base and other commonly used functionality will reduce our work and make the case study a beautiful one.

9.2 CONCLUSION

Drowsiness affects mental alertness, decreasing an individual's ability to operate a vehicle safely and increasing the risk of human error that could lead to fatalities and injuries. Furthermore it has been shown to slow reaction time, decreases awareness, and impairs judgment. Long hours behind the wheel in monotonous driving environments make truck drivers particularly prone to drowsy-driving crashes successfully addressing the issue of driver drowsiness in the commercial motor vehicle industry is a formidable and multi-faceted challenge. Since a large number of road accidents occur due to the driver drowsiness, this system will be helpful in preventing many accidents, and consequently save money and reduce personal suffering. This system will monitor the driver eyes using a camera and by developing an algorithm we can detect symptoms of driver fatigue early enough to avoid an accident. So this project will be helpful in detecting driver fatigue in advance and will give warning output in form of sound and seat belt vibration whose frequency will vary between 50 to 60 Hz. And also helps to identify the driver alcoholic intoxication.

9.3 FUTURE ENHANCEMENT

- This project can be implemented in the form of mobile application to reduce the cost of hardware.
- This project can be integrated with car, so that automatic speed control can be imparted if the driver is found sleeping.
- And also implementing automatic speed control if the driver is found alcoholic.
- Yawning detection based on the changes in the mouth geometric features.

REFERENCES

1. BOOKS

- Java 2 –the complete reference
- Fundamentals of software engineering

2. WEBSITES

- www.tutorialspoint.com/java
- <https://www.tutorialspoint.com/arduino>
- <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html>

3. JOURNALS AND PUBLICATIONS

- **Mohammad Amin Assari and Mohammad Rahmati (2011)** “Driver drowsiness detection using face expression recognition”, IEEE publications.
- **Osama Rizwan, Hamza Rizwan, Mudassar Ejaz (2013)** “Development of an efficient system for vehicle accident warning”, 2013 IEEE 9th International Conference on Emerging Technologies (ICET).
- **Garima Turan, Sheifali Gupta**, “Road Accidents Prevention system using Driver’s Drowsiness Detection”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 11, November 2013.
- **Syad Imran Ali, Prasanth Singh and Sameer Jain**, “An efficient system to identify user attentiveness based on fatigue detection”, March 2014 International Conference on Information Systems and Computer Networks (ISCON).

APPENDICES

1. SCRUM BOARD

i. Git

Git is a version-control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source-code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision-control system, it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

ii. Git Repositories

A Git repository contains the history of a collection of files starting from a certain directory. The process of copying an existing Git repository via the Git tooling is called cloning. After cloning a repository the user has the complete repository with its history on his local machine. Of course, Git also supports the creation of new repositories. If you want to delete a Git repository, you can simply delete the folder which contains the repository. If you clone a Git repository, by default, Git assumes that you want to work in this repository as a user. Git also supports the creation of repositories targeting the usage on a server.

iii. Scrum

Scrum is an agile way to manage a project, usually software development. Agile software development with Scrum is often perceived as a methodology; but rather than viewing Scrum as methodology, think of it as a framework for managing a process. In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented.

In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented.

Within agile development, Scrum teams are supported by two specific roles. The first is a Scrum Master, who can be thought of as a coach for the team, helping team members use the Scrum process to perform at the highest level. The product owner (PO) is the other role, and in Scrum software development, represents the business, customers or users, and guides the team toward building the right product.

iv.Git History

Branch: master ▾


New pull request

Create new file
















Upload files

Find File

Clone or download ▾

 jasminejoseph123 driver eye ...

Latest commit dea35b1 7 minutes ago


 MicroDVR2.0.exe	device login	2 days ago
 README.md	README.md	26 days ago
 admin.java	adminhome	26 days ago
 adminhome.java	adminhome	26 days ago
 alcohol.ino	Alcohol	2 days ago
 alcohol_with_beep.ino	alcohol beep	2 days ago
 alsensor.ino	Alcoholse	11 hours ago
 connection.txt	connection	2 days ago
 driver dwos left eye.txt	driver eye	7 minutes ago
 login.java	login	26 days ago
 owner.java	owner	26 days ago
 owner.txt	owner	2 days ago
 serialconnection.txt	serial connection	2 days ago
 sketch_may09al.ino	alcohol detection	2 days ago
 sobel filtering.exe	drowsy	11 hours ago


Commits on Jun 3, 2019

driver eye ...

jasminejoseph123 committed 11 hours ago

Verified

 dea35b1





Commits on Jun 2, 2019

drowsy ...

jasminejoseph123 committed 21 hours ago

Verified


 a96f6f0




Alcoholse ...

jasminejoseph123 committed 21 hours ago

Verified

 18d461b





Commits on Jun 1, 2019

serial connection

jasminejoseph123 committed 2 days ago

Verified


 3768f07




owner ...

jasminejoseph123 committed 2 days ago

Verified


 767adf4




connection ...

jasminejoseph123 committed 2 days ago

Verified


 1051feb




device login ...







jasminejoseph123 committed 2 days ago

Verified











 e6d1902











Commits on May 31, 2019





alcohol detection ... jasminejoseph123 committed 3 days ago	Verified		d26c140	
alcohol beep ... jasminejoseph123 committed 3 days ago	Verified		68e497f	
Alcohol ... jasminejoseph123 committed 3 days ago	Verified		ddb7a21	

Commits on May 7, 2019

README.md jasminejoseph123 committed 27 days ago	Verified		af308f1	
owner ... jasminejoseph123 committed 27 days ago	Verified		fff2a18	
admin ... jasminejoseph123 committed 27 days ago	Verified		1ef0dfb	
adminhome ... jasminejoseph123 committed 27 days ago	Verified		92e2904	
login ... jasminejoseph123 committed 27 days ago	Verified		a6261e0	

owner ... jasminejoseph123 committed 27 days ago	Verified		fff2a18	
admin ... jasminejoseph123 committed 27 days ago	Verified		1ef0dfb	
adminhome ... jasminejoseph123 committed 27 days ago	Verified		92e2904	
login ... jasminejoseph123 committed 27 days ago	Verified		a6261e0	

Commits on May 1, 2019

Delete proj2.java jasminejoseph123 committed on May 1	Verified		186692b	
login page jasminejoseph123 committed on May 1	Verified		c40e9a2	

2. LIST OF TABLES

Table no	Table name	Page no
Table 5.2.1	Image table	21
Table 7.5.1	Test results	31

1. LIST OF FIGURES

Figure no	Figure name	Page no
Figure 4.1.1	Arduino Uno	11
Figure 4.1.2	MQ-3 Alcohol Sensor	12
Figure 5.1.1	Components of activity diagram	16
Figure 5.1.2	Components of flow chart	17
Figure 5.1.2.1	Activity Diagram- Driver drowsiness detection	19
Figure 5.1.2.2	Flow chart- Alcohol detection	20

2. ABBREVIATIONS AND NOTATIONS

i. MQ-3

MQ3 is a type of gas sensor which actually has its full form in Chinese. M stands for ‘*mingan*’ which means sensitive and Q stands for ‘*Qǐ lái*’ which means gas to, i.e., it is sensitive to gas.

ii. MS SQL

Microsoft SQL (Structured Query Language) Server is a relational database management system developed by Microsoft.

iii. MVD

Motor Vehicle Department

3. CODING

i. Drowsiness detection

```
public Bitmap left_eye(Bitmap b)
{

    b = black_white(b);
    int i, j, max, x, h, w, y;
    max = 0;
    x = 0;

    h = b.Height;
    w = b.Width;
    y = h - 1;
    for (i = 2; i < 2 * h / 3; i++)
    {
        for (j = 0; j < w; j++)
        {
            if (b.GetPixel(j, i).R == 0 && b.GetPixel(j, i).G == 0 && b.GetPixel(j,
            i).B == 0)
            break;
        }
        if (j == w)
        j = w - 1;
        if (max <= j && j != 0)
        {
            max = j;
            x = i;
        }
    }
}
```

```

if (i >= h / 2 && j < w / 3)
break;
}
for (i = h - 1; i >= h / 2; i--)
{
int count = 0;
for (j = w / 4; j <= 3 * w / 4; j++)
if (b.GetPixel(j, i).R == 0 && b.GetPixel(j, i).G == 0 && b.GetPixel(j,
i).B == 0)
count++;
if (count * 2 >= w / 2)
break;
}
if (i != h - 1)
y = i + 1;

int le_1 = 0, ri_1 = w - 1, ttt = 0;
ttt = 0;
for (j = 0; j < w; j++)
{
for (i = 0; i < h; i++)
if (b.GetPixel(j, i).R == 0 && b.GetPixel(j, i).G == 0 && b.GetPixel(j,
i).B == 0)
{
ttt = 1;
break;
}
}
if (ttt == 1)
{

```

```

le_1 = j;
break;
}
}
ttt = 0;
for (j = w - 1; j >= 0; j--)
{
for (i = 0; i < h; i++)
if (b.GetPixel(j, i).R == 0 && b.GetPixel(j, i).G == 0 && b.GetPixel(j,
i).B == 0)
{
ttt = 1;
break;
}
if (ttt == 1)
{
ri_1 = j;
break;
}
}

ttt = 0;
for (i = x; i < y; i++)
{
for (j = w / 4; j < w - w / 4; j++)
if (b.GetPixel(j, i).R == 0 && b.GetPixel(j, i).G == 0 && b.GetPixel(j,
i).B == 0)
{
ttt = 1;

```

```

break;
}
if (ttt == 1)
{
x = i;
break;
}
}
ttt = 0;
for (i = y; i >= x; i--)

{
for (j = w / 4; j < w - w / 4; j++)
if (b.GetPixel(j, i).R == 0 && b.GetPixel(j, i).G == 0 && b.GetPixel(j,
i).B == 0)
{
ttt = 1;
break;
}
if (ttt == 1)
{
y = i;
break;
}
}

b = new Bitmap(pictureBox3.Image);
Bitmap BB = new Bitmap(w - le_1, h - x);

```

```

for (i = le_1; i < w; i++)
for (j = x; j < h; j++)
BB.SetPixel(i - le_1, j - x, b.GetPixel(i, j));

return BB;

}

public Bitmap right_eye(Bitmap b)
{

b = black_white(b);

int i, j, max, x, h, w, y;
max = 0;
x = 0;

h = b.Height;
w = b.Width;
y = h - 1;
for (i = 2; i < 2 * h / 3; i++)
{
for (j = w - 1; j >= 0; j--)
{
if (b.GetPixel(j, i).R == 0 && b.GetPixel(j, i).G == 0 && b.GetPixel(j,
i).B == 0)
break;
}
j = w - j;
if (j == w)

```



```

j = w - 1;
if (max <= j && j != 0)
{
max = j;
x = i;
}
if (i >= h / 2 && j < w / 3)
break;
}

for (i = h - 1; i >= h / 2; i--)
{
int count = 0;
for (j = w / 4; j <= 3 * w / 4; j++)
if (b.GetPixel(j, i).R == 0 && b.GetPixel(j, i).G == 0 && b.GetPixel(j,
i).B == 0)

count++;
if (count * 2 >= w / 2)
break;
}
if (i != h - 1)
y = i + 1;

int le_1 = 0, ri_1 = w - 1, ttt = 0;
ttt = 0;
for (j = 0; j < w; j++)
{
for (i = 0; i < h; i++)

```

```

if (b.GetPixel(j, i).R == 0 && b.GetPixel(j, i).G == 0 && b.GetPixel(j,
i).B == 0)
{
    ttt = 1;
    break;
}
if (ttt == 1)
{
    le_1 = j;
    break;
}
}
ttt = 0;
for (j = w - 1; j >= 0; j--)
{
    for (i = 0; i < h; i++)
    if (b.GetPixel(j, i).R == 0 && b.GetPixel(j, i).G == 0 && b.GetPixel(j,
i).B == 0)
    {
        ttt = 1;
        break;

    }
    if (ttt == 1)
    {
        ri_1 = j;
        break;
    }
}
}

```

ii. Alcohol Detection

```
int Input = A5;
int SensorVal = 0;
int Check = 0;
void setup() {
  Serial.begin(9600);
  pinMode(Input, INPUT);

  // Serial.println("Design by www.TheEngineeringProjects.com");
  //Serial.println();
}

void loop() {

  SensorVal = analogRead(Input);
  // Serial.println(SensorVal);
  if((SensorVal > 170) && (Check == 1))
  {
    Serial.println("a");
    Check = 0;
  }

  if((SensorVal < 170) && (Check == 0))
  {
    //Serial.println("All Clear . . .");
    Check = 1;
  }
  //Serial.println(SensorVal);
  delay(500);
}
```