ABSTRACT

Project topic is DrowsiShield that ensures drivers safety. The main aim of this system is to avoid accidents by instantly detecting the driver's level of drowsiness and sending alerts through audio and haptic stimulation. It also aims to promote road safety through oversight of driver vigilance and quick response to emergencies.

ACKNOWLEDGEMENT

I express my sincere gratitude to the Principal, Dr. Parag Ajagaonkar, and the Vice- Principal, CA Dr. Savita Desai, Prof. Geeta Desai & Prof. Heena Bhuva, of Narsee Monjee College of Commerce and Economics, for providing me with an enriching learning experience and opportunities to showcase my abilities. I would also like to extend my heartfelt thanks to my Head of Department, Prof. Anupama Jawale, for her unwavering support and guidance throughout the project.

I take this opportunity to express my profound gratitude and deep regards to my guide Dr. Neha Kushe, for exemplary guidance at each step of this project, monitoring and constant encouragement throughout the course of this project.

I would also like to thank the department of BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY), for their resources and training that made the implementation smoother. A special mention to all the faculties, lab assistants and the non-teaching staff of the department as well for their efforts and cooperation.

Lastly, I would like to thank the institute, SVKM NARSEE MONJEE COLLEGE OF COMMERCE AND ECONOMICS, for giving me this opportunity to explore and expand our horizons through this project and all our academic endeavors.

DECLARATION

I hereby declare that the project entitled, "DrowsiShield" done at Narsee Monjee College of Commerce and Economics, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

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

Name and Signature of the Student

Date :

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 : INTRODUCTION

1. Theoretical Background

IoT system architecture

IoT has revolutionized our lives by presenting the thought of objects communicating through an interconnected remote organize of sensors. These communicating things, gadgets or objects have the capacity to sense/perceive their environment through a set of coordinates sensors and act through a set of planned actuators. Sensors are utilized to gather information from the working environment of the gadget and send it to the microcontroller, which forms the information, analyses it and creates control activities accordingly. IoT has a few viewpoints concurring to the scope said. IoT can be clarified as some administrations given by gadgets communicating with a computer or with each other to enrich human life. IoT can too be portrayed as a world without impediments and boundaries, where it is conceivable to get to any gadget at any put and communicate at any time and effectively and timely. In conventional arrange frameworks where computers are associated through the Web, information is created and utilized by individuals, where in IoT, information is collected by some sensors and utilized by a few actuators that start a few activities in their work environment.

2. Objective of the Project

- To develop a robust hardware setup comprising an Arduino Nano microcontroller, eye blink sensor, buzzer, vibration motor, GSM module, and GPS module**.**
- To design and integrate feedback mechanisms utilizing the buzzer and vibration motor to promptly alert the driver upon detection of drowsiness.
- To develop a user-friendly device allowing the driver to manually deactivate the alert system using a switch button when necessary.
- To integrate the GSM module to enable automatic transmission of SMS alerts to predefined emergency contacts in the event of an accident and to incorporate the GPS module to provide real-time location data, facilitating precise localization of the vehicle during emergency situations.
- To conduct rigorous testing procedures as well as calibration processes under various operating conditions ensuring reliability.

3. Purpose, Scope & Availability of the Project

A. Purpose

The primary purpose of this project is to enhance driver's safety and reduce the risk of accidents that is caused by drowsiness or unconsciousness while driving. By integrating a range of hardware components like Arduino nano, buzzer, vibration motor, GSM module and GPS module, the system is used to designed to detect the drowsiness of driver in real-time. If the driver's eyes remained closed for longer period of time, then the system triggers a series of alert, such as audible and vibration warnings. If there is no response then it sends SMS alert and containing the driver's location to the designated contact. It underscores the significance of leveraging the technology to mitigate road accidents and also underscores the potential of hardware-based solutions to address safety concerns. By raising awareness about dangers of drowsy- driving and offering proactive mechanisms, this technology seeks to ensure safety and ultimately save lives.

B. Scope

a. Limitation:

1. Technical Complexity: The project involves integrating multiple hardware and software components which may result in technical challenges and limitations. This may include compatibility issues, stability problems, and accuracy limitations of the sensors used.

2. Cost: Implementing a wearable device with advanced features and capabilities such as sensory feedback, and emergency alerts can be expensive. This may limit the accessibility of the technology for some individuals.

3. User Adaptability: Some individuals may struggle to adapt to using a new technology, especially those with limited experience using wearable devices. This may require additional training and support to ensure a positive user experience.

4. Environmental Factors: Sensors may struggle in certain conditions such as in low light or bright light which can affect their accuracy.

b. Accessibility:

1. Affordability: The project uses readily available off-the-shell hardware components like Arduino Nano, buzzer, vibration motor, GSM Module, GPS Module, eye blink sensor which are relatively affordable.

2. Adaptability: The system is adaptable and can easily integrate with different types of vehicles including personal cars, commercial vehicles, buses and trucks, etc.

3. User Friendly Design: A switch button is used to deactivate the alerts, giving driver a sense of control and autonomy over the system.

4. Customizable Alerts: The alert system can be customized to send SMS notification to multiple contacts, ensuring that the driver’s safety is communicated to a network of concerned individuals.

5. Simplicity: The project design is very simple and straightforward, making it accessible to a wide of users.

6. Versatility: The system can be modified or expanded based on specific requirements and preferences, making it versatile solution for addressing driver drowsiness.

C. Applicability

The applicability of this project extends across a wide spectrum of scenarios and environments while driver’s safety is important. One of its key applications is in automobile industry, particularly in vehicles where long and continuous hours of driving are common, such as trucks, buses and other vehicles. Here the system can be incorporated into wearable devices for drivers, and ensures that they remain vigilant and alert during journeys.

Another significant application is in domain of transportation and logistics, where driver’s fatigue is major concern. By employing this system, fleet operators can ensure that their drivers are alert and responsive, minimizing the risk of accidents and improving overall operational efficiency.

Furthermore, the system’s integration with GPS technology enables it to be used in emergency response and rescue operations. In scenarios where drivers are involved in accidents or require immediate assistance, the system can transmit their precise location to rescue teams, expediting the response and potentially saving lives.

DIRECT APPLICATIONS:

1. Personal Vehicles: The system can be used in personal cars to alert individual drivers and their designated contact in case of drowsiness or emergency.

2. Commercial Fleets: Fleet operators can also use the system in commercial vehicles like buses and trucks to ensure safety of drivers and passengers.

3. Fleet Management Services: Companies offering fleet management can provide the system as an add-on feature to enhance the safety of their client’s safety.

4. Ride Share and Taxi Services: Companies offering rideshare or Taxi services can equip their vehicles with the system to enhance passengers’ safety.

INDIRECT APPLICATIONS:

1. Data Analysis for Road Safety Research: Analysis of eye-blink data patterns may lead to insights that help in the development of better road safety measures and policies.

2. Emergency Responses: The system's ability to automatically send alerts in case of emergency can facilitate their faster response time and potentially save lives in critical situations.

3. Insurance Premium Reduction: In additional to potential discounts, the adoption of safety technologies like this system may lead to reduced insurance premiums for drivers and fleet operators.

4. Public Awareness: The implementation of DrowsiShield can help raise awareness about the dangers of drowsy driving and encourage responsible driving behaviour.

4. Expected Achievements

The end goal of this project is to enhance driver's safety by detecting driver's drowsiness and unconsciousness while driving. The system's ability to alert drivers and external contacts in real-time and can significantly reduce the risk of accidents due to drowsy driving.

Expected achievements of this project include:

- Reduce accident rates by providing early warnings and alerting systems which is significant contributor to road accidents.
- Improved response time where the system will automatically send SMS notification and GPS location in case of emergencies.
- The project's implementation can also contribute to raising awareness about the dangers of driving while drowsy, encouraging responsible driving habits and potentially influencing border road safety initiatives.
- It allows scalability to be implemented in single vehicles or expanded to entire fleets, making it versatile solution for individuals, family and organizations.

In conclusion, the project aims to promote driver safety, reducing accident rates, improving emergency response, raising road safety awareness and providing an affordable, accessible solution for enhancing road safety.

CHAPTER 2 : SURVEY OF TECHNOLOGIES

1. Description of Available Technologies

There are various technologies available in the IoT domain that can be used to create device to detect driver's drowsiness. Some of them are:

• Microcontrollers: Microcontrollers such as Arduino, Raspberry Pi, and ESP8266 can be used to control the glasses and process the eye blink inputs.

• Sensors: Sensors such as proximity sensors, light sensors, and temperature sensors can be used to detect the wearer's environment and provide inputs to the microcontroller.

• Actuators: Actuators such as vibration motors, LED lights, and buzzers can be used to provide feedback to the wearer based on the eye blink inputs.

• Bluetooth: Bluetooth technology can be used to connect the glasses to a mobile application that can be used to input eye blink or change the glasses' settings.

• Machine Learning: Machine learning algorithms can be used to train the glasses to recognize specific eye blink patterns and provide customized feedback to the wearer.

• Cloud Computing: Cloud computing can be used to store and analyse data collected by the glasses, allowing for remote monitoring and analysis of the wearer's condition.

2. Comparative Analysis of Technologies in Chosen Area

Factors	**Arduino Nano**	**Arduino Uno**
Size	Smaller and compact	Bulkier and heavier
Processor	Microcontroller is ATmega328(older versions) ATmega328P (newer).	Microcontroller is ATmega328(older versions) ATmega328P (newer).
Voltage	Operating voltage is 5V	Operating voltage is 5V
Flash Memory	Flash Memory is 32KB	Flash Memory is 32KB
USB	USB Interface is Mini USB	USB Interface is USB-B
Cost	Less expensive than Uno	More expensive than Nano
Pins	Generally, have more analog pins than Uno.	Generally, have fewer analog pins than Nano.

In summary, Arduino Nano is smaller and compact board compared to Arduino Uno, making it suitable for projects where space is limited. It has the same microcontroller like Uno, so it offers similar performance but with a smaller footprint. The Nano is less expensive than Uno, making it cost- effective choice for budget conscious project. Nano is great choice for wearable tech, while Uno is better suited for general-purpose projects.

3. Chosen Project Domain

The chosen project domain is driver drowsiness detection, focusing on developing that utilizes hardware components such as Arduino nano, GSM Module, GPS Module, an eye blink sensor, a vibration motor and a buzzer to enhance driver safety. This system is designed to detect drowsiness in real-time and triggers alert to wake up the driver. In case of driver's accident or unresponsiveness, the system automatically sends an SMS containing the driver's location as well to designated contact.

4. Technologies to be used

The creation of DrowsiShield involves the use of various technologies in the IoT domain. The glasses may incorporate IR sensors modules and microcontrollers such as the Arduino Nano for processing data and controlling output devices such as buzzers and vibration motor. The emergency alerts and location of driver can be sent through GSM and GPS Modules. Driver's eye blink mechanism can be done through programming using the Arduino IDE or other integrated development environments.

A. Front End

Eye bink sensor module

B. Back End

- Arduino Nano
- Buzzer
- Vibration Motor
- GSM Module
- GPS Module

C. Other Development Tools

Arduino IDE and C++

5. Reason Supporting the use of above selected technologies

The reasons for using these components in the project are as follows:

- Arduino Nano - versatile and affordable microcontroller board that provides compatibility and expandability, making it an ideal choice for many IoT projects, including the driver drowsiness glasses project.

- Buzzer - The buzzer provides audio feedback and making the device more accessible and user-friendly.

- Vibration Motor - The vibration motor provides haptic feedback and making the device more accessible and user-friendly like buzzer.

- GSM Module – It allows the system to send the SMS notification to a designated contact in case of emergency.
- GPS Module – It allows the system to determine and communicate the driver's location to the emergency contact.

- Eye Blink Sensor - The eye blink sensor detects the eye blink input by the user and detects the drowsiness of the driver.

In conclusion, these components are chosen for their versatility, accessibility, and ease of use, allowing for the development of a functional and user-friendly device that meets the needs of individuals with drowsiness.

CHAPTER 3 : REQUIREMENT & ANALYSIS

1. Problem Statement and Problem Definition

Problem Statement :

To develop a Driver Drowsiness Detection & Accident Alert System that will ensure safety of the driver and prevent road accidents.

Problem Definition :

While caffeine intake, taking nap and rest may increase our alertness, but in some driving situations especially in late night or long-distance driving, these things are not enough to relieve our drowsiness and fatigue, and if the incident is not reported to emergency personnel or designated contact as soon as possible then it may take long time and may not provide immediate assistance to the driver. Therefore, we need an early warning system that can detect the sleep early and provide emergency alerts without any delay.

The problem definition for this is to ensure safety issue of drowsy driving, which can lead to accidents and fatality on the roads. Drowsy driving is a serious problem, affecting millions of drivers every year, and it poses significant risk to public safety. This project aims to develop a driver drowsiness detection using microcontroller and various sensor to detect drowsiness in real-time. The system will detect drowsiness and will trigger alert and cause vibrations including SMS notification and GPS location to ensure prompt assistance to driver in critical situations.

The challenge in this project is to develop a device that meets these requirements while incorporating key features such as eye blink input, real-time updates, and provide timely alerts without any delay.

By addressing this problem, this project ensures enhanced road safety and reduce the number of accidents caused by drowsy driving. In conclusion, the goal is to develop a smart glasses device that can detect drowsiness of the driver and provide timely alerts to avoid accidents while driving.

2. Requirement Specification :

To develop a DrowsiShield that uses hardware components such as microcontroller and sensors. The system must accurately monitor the driver's eye blink pattern to detect drowsiness in real-time. It should provide the timely alerts with vibration and audible warnings. In case of emergency or if the driver remains unresponsive, the system should automatically send an SMS and the driver's GPS location to the designated contact.

A. Functional Requirements

- Eye Blink Detection: The system must accurately detect the eye blink pattern using an eye blink sensor.
- Real-Time Alerting: The system must provide real-time alerts, such as audible alerts and vibrations, to wake up the driver when signs of drowsiness are detected.
- Emergency Notification: In case of emergency or if the driver remains unresponsive, the system must automatically send an SMS notification and GPS location to the designated contact.
- Adaptability: The system must be adaptable to different types of vehicle and user preferences, ensuring its accessibility to wide range of users.
- Reliability and Accuracy : The system mut be reliable and accurate, minimizing false alerts and ensuring timely and effective detection of drowsiness.
- Cost-effectiveness: The system must be cost-effective, using readily available and affordable components to ensure its accessibility and practicality to users.

B. <u>Non-Functional Requirement</u>

- Usability: The system should have intuitive interface, making it users for to configure settings and interpret alerts.
- Performance: The system should have low latency, ensuring real-time detection and alerting to minimize the risk of accidents.
- Reliability: The system should be robust and reliable, with minimal false alerts and high accuracy in detecting drowsiness.
- Scalability: The system should be scalable, capable of handling multiple users and accommodating future expansions and upgrades.
- Safety: The device does not emit any harmful radiation or interfere with other electronic devices.

C. <u>User Requirements</u>

- Reliable Detection: User expect the system to reliably detect drowsiness in real-time to prevent accidents caused by driver fatigue.
- Fast Response: User expect the system to respond quickly to drowsiness by providing alerts and interventions to wake up the driver.

- Ease of Use: User expect the system to be user-friendly, with simple interface for configuring settings and alerts.
- Adaptability: User expect the system to be adaptable to different types of vehicles and user preferences, ensuring its practicality and accessibility to wide range of users.
- Cost-effectiveness: User expect the system to be cost-effective, using affordable components to ensure its accessibility to users.
- Reliability : User expect the system to be reliable, with minimal false alerts and high accuracy in detecting drowsiness.

D. Hardware Requirements

- Microcontroller: Arduino Nano
- Eye Blink Sensor
- Buzzer
- Vibration Motor
- GSM Module
- GPS Module

E. Software Requirements

The software requirements for this project primarily revolve around the use of the Arduino Integrated Development Environment (IDE). The Arduino IDE serves as the central platform for writing, editing, and uploading code to the Arduino Nano microcontroller board. It provides a user-friendly interface with essential features such as code syntax highlighting and serial monitoring. Additionally, the Arduino IDE offers a vast library of pre-written code examples and libraries, making it easier to integrate various sensors and modules into the project. Its compatibility with Arduino boards, including the Arduino Nano used in this project, ensures seamless development and testing of the DrowsiShield. Overall, the Arduino IDE is essential for programming the Arduino Nano and implementing the necessary logic and functionality to detect drowsiness, trigger alerts, and communicate with external modules such as the eye blink sensor, GSM module, and GPS module.

3. Feasibility

The revolution of the Internet, communication protocols and the introduction of the Internet of Things technology (IoT), have enabled the generation, processing and storage of data on a large scale. It also enabled real-time services and initiated control actions based on immediate processing of data items. The field of IoT has recently evolved to improve the quality of our daily lives by enabling communication between various things that share our physical world over the Internet.

Most devices invented for users with drowsiness detection are not portable in nature. The complexity of implementing such devices and their limited usage scenarios also make them tailored to the specific needs of patients and accordingly trigger some limited actions. Furthermore, all invented devices should follow a standard communication language to create a single flexible environment suitable for all users with hearing impairments and to help them in their different situations.

A. Operational Feasibility

- Affordability : The cost of producing the device is reasonable and within the budget of the users.
- Ease of Use: The system should be user-friendly, with simple interfaces and intuitive controls, allowing drivers to operate it without extensive training or technical expertise.
- User Acceptance: The operational feasibility of the device using Driver's Drowsiness depends on user acceptance. The device should be designed to meet the needs of people with drowsiness, and the users should be willing to use the device.
- Reliability and Performance: The system should demonstrate reliable performance under various conditions, accurately detecting drowsiness and triggering alerts when necessary, while minimizing false alarms.

B. Technical Feasibility

- Hardware Requirements: The Morse glasses project will require hardware components such as a Microcontroller :
  a. Arduino Nano
  b. Buzzer
  c. Vibration Motor
  d. Switch Button
  e. GSM Module
  f. GPS Module

1. Availability of necessary hardware components: The project requires the use of a number of hardware components such as Arduino Nano, Buzzer, Vibration Motor, Switch Button, GSM Module and GPS Module. All of these components are readily available in the market and can be easily sourced.
2. Availability of necessary software tools: The software tools required for the project such as Arduino IDE that are freely available and can be downloaded from the respective website.
3. Technical expertise: The project requires expertise in programming, electronics, and IoT. The team should have the necessary skills and expertise to build and test the system.

The feasibility of the project depends on these components because these components are versatility, accessibility, and ease of use, allowing for the development of a functional and user-friendly device that meets the needs of individuals with drowsiness.

- Software Requirements: The project requires software to process the Drowsiness Detection & Accident Alert System, and detect drowsiness Additionally, custom software logic needs to be developed to implement the core functionalities of the system, including algorithms for detecting drowsiness, triggering alerts, and sending SMS notifications with GPS location data. Its compatibility with Arduino boards, including the Arduino Nano used in this project, ensures seamless development and testing of the DrowsiShield.

C. <u>Economic Feasibility</u>

- The Arduino Nano is relatively low-cost and widely available, making it an affordable component to use in the project.
- The cost of the additional components such as the eye blink sensor, buzzer, vibration motor, GSM Module and GPS Module can be kept low with careful selection of components.
- The cost of the project can be further optimized by using open-source libraries for the programming.
- The project has the potential to reduce the need for human intervention in monitoring and providing assistance to the drowsy people, which can have economic benefits in the long run.

1. <u>Planning and Scheduling</u>

Planning is a crucial step in creating any project, and it helps to ensure that the project is completed on time, within budget, and meets the required specifications. Here are some key steps in planning the IoT-based project for DrowsiShield:

- PROJECT SCOPE and OBJECTIVE:
  The scope of the project here is to develop a driver drowsiness detection system using Arduino Nano and various other sensors to accurately monitor and alert the driver and emergency contacts in case of drowsiness or an accident.

- PROJECT PLAN: The plan for the project started back in October, 2022 after the exams ended the planning steps were pretty simple and they were as follows:
  1. Research phase: Where we had to find the best way possible to give birth to our idea and to get as much knowledge as possible to make it easy for us.
  2. Hardware implementation: Finding the best pieces of hardware possible to be used out of many options present around in the modern era and choosing the one's that integrate well with each other.
  3. Software implementation: Implementing the code for the hardware selected.
  4. Designing: Phase where we designed all of our products and integrating the hardware as a functional product.
  5. Testing: Where the products are tested to see if they work properly and complement each other (i.e. hardware and software).
  6. Documentation: Where the black book is documented.

Scheduling is an important part of project management that helps ensure that the project is completed on time and within budget.
We scheduled our project at first to get the best out of the weeks we have to perfectly get the desired outcome and to get the best of both worlds hardware and software that we will be using and to get the optimum level of familiarity when using the product together.

A. Gantt Chart

Fig 3.1 Gantt Chart

2. Preliminary Product Description

Our product is a wearable device that ensures driver's safety and prevents from accident as well as keeps awake. The device is equipped with eye blink sensor that detects the drowsiness based on eye blink patterns. Eye Blink Sensor then sends the data to the microcontroller Arduino Nano and then it causes alert mechanism through other sensors such as buzzer and vibration motor. In the event of persistent drowsiness or a high risk of an accident, the system activates an emergency alert feature, sending SMS notifications to designated contacts along with the vehicle's GPS location. Additionally, a manual override option allows the driver to disable the emergency alert feature in non-emergency situations.

a. Description:

- To provide a portable IoT device that can detect drowsiness of the driver.
- To alert the driver on time when the driver when drowsiness is detected.

b. Requirements:

- The device must be able to detect drowsiness from Eye Blink Sensor.
- Eye Blink Sensor must be able to send data to Arduino Nano in real-time for alert mechanisms.
- The device must perform alert warnings when driver eyes are closed and keep the driver alert and stay awake while driving.

3. Conceptual Model

Following the Prototype Model from the Software Development Life Cycle (SDLC) for the driver drowsiness detection project needs a phased approach focused on rapid prototyping and iterative development. Initially, the project begins with the identification of key requirements and objectives, outlining the desired functionality and features of the system. This is followed by the creation of an initial prototype, typically a basic version of the system that demonstrates core functionalities such as sensor integration, drowsiness detection, and alert triggering.

Once the initial prototype is developed, it undergoes extensive testing and evaluation to gather feedback from potential users. This feedback is then used to refine and improve the prototype in subsequent iterations, incorporating additional features, enhancing performance, and addressing any identified issues or limitations. Each iteration builds upon the previous one, gradually evolving the prototype into a more polished and robust solution.

Throughout the development process, collaboration and communication play a crucial role in ensuring the success of the prototype model. Regular testing, reviews, and demonstrations are conducted to facilitate continuous improvement and alignment with project goals.

By following the Prototype Model, the DrowsiShield project benefits from increased flexibility, reduced development time, and greater adaptability to changing requirements and user needs. This iterative approach allows for early validation of concepts and rapid experimentation with different design options. Ultimately, the Prototype Model enables to create a highly functional and user-centric solution that effectively addresses the problem of drowsy driving while minimizing risks and maximizing efficiency.

A. Process Model

DESIGN OF PROCESS MODEL:

The process model followed based on Prototype for the project is described as follows:

- The process involves iterative development and rapid prototyping.
- Project progresses through cycles of creating prototypes, testing, gathering feedback, and refinement.
- Initial focus on identifying key requirements and objectives.

- Development of an initial prototype emphasizing core functionalities like sensor integration and drowsiness detection.
- Subsequent iterations of the prototype are developed based on feedback, incorporating additional features and refinements.
- Iterative process continues until the final prototype meets desired functionality, usability, and performance standards.
- Deploy the final product.

REASON FOR CHOOSING PROTOTYPE DEVELOPMENT MODEL

1. Rapid Iterative Development: The project involves quick iterations and frequent refinements to ensure the effectiveness of the drowsiness detection system. The Prototype Model allows for rapid prototyping and iteration, enabling the team to experiment with different designs, functionalities, and algorithms in a timely manner.

2. Early Validation of Concepts: By developing prototypes early in the development process, the team can validate design concepts and functionalities before investing significant time and resources into full-scale implementation. This helps identify potential issues or shortcomings early on, allowing for timely adjustments and improvements.

3. User Involvement and Feedback: The Prototype Model emphasizes user involvement and feedback throughout the development process. By asking input from potential users, testers, and stakeholders during each iteration, the team can ensure that the drowsiness detection system meets user needs and expectations effectively.

4. Flexibility and Adaptability: The Prototype Model offers flexibility and adaptability, allowing the team to respond quickly to changing requirements, emerging technologies, and user feedback. This agility is essential in a project like driver drowsiness detection, where the effectiveness of the system may depend on real-world factors and evolving user preferences.

5. Reduced Risk of Costly Mistakes: Developing prototypes allows the team to explore different design options and functionalities without committing to full-scale implementation. This reduces the risk of costly mistakes or design flaws that may arise later in the development process, ultimately saving time and resources.

   ADVANTAGES:
   - Early validation of concepts and functionalities.
   - Rapid prototyping allows for quick iteration and refinement.
   - Enhances user involvement and feedback throughout the development process.
   - Facilitates flexibility and adaptability to changing requirements.

- Reduces the risk of costly mistakes by exploring design options early.
- Provides a tangible representation of the final product, aiding in communication and decision-making.

DISADVANTAGES:

- May lead to incomplete or inaccurate requirements gathering if not properly managed.
- Requires additional time and effort for prototyping and iteration.
- Possibility of diverging from initial project scope if too many changes are made during prototyping.
- Potential for confusion or misunderstanding if prototypes are not clearly documented or communicated.
- Increased risk of project delays if prototypes require significant rework or redesign.
- Can be challenging to estimate project timelines and costs accurately due to iterative nature.

B. <u>The goals of a process model are to be :</u>

- Descriptive :
  The goal of achieving a functional and highly reliable IoT system was achieved successfully with the help of the Prototype process model.

- Prescriptive :
  The goal of the Prototype process model for the project could be to deliver a functional and reliable IoT system that meets the user's requirements in an efficient and iterative manner, while maintaining flexibility to accommodate changes in requirements and feedback throughout the development process.

- Explanatory :
  Every process right from the very beginning i.e. planning till the last phase of testing and preparing the final test cases for the documentation and product/prototype deployment it was all perfect as prototype model helps us with the with every single unit of the project and provides us with a great overview of the hardware as well as software that we had chosen in developing the project.

C. <u>Diagrams to be included in the design phase are as follows :</u>

1. <u>Use Case Diagram</u>

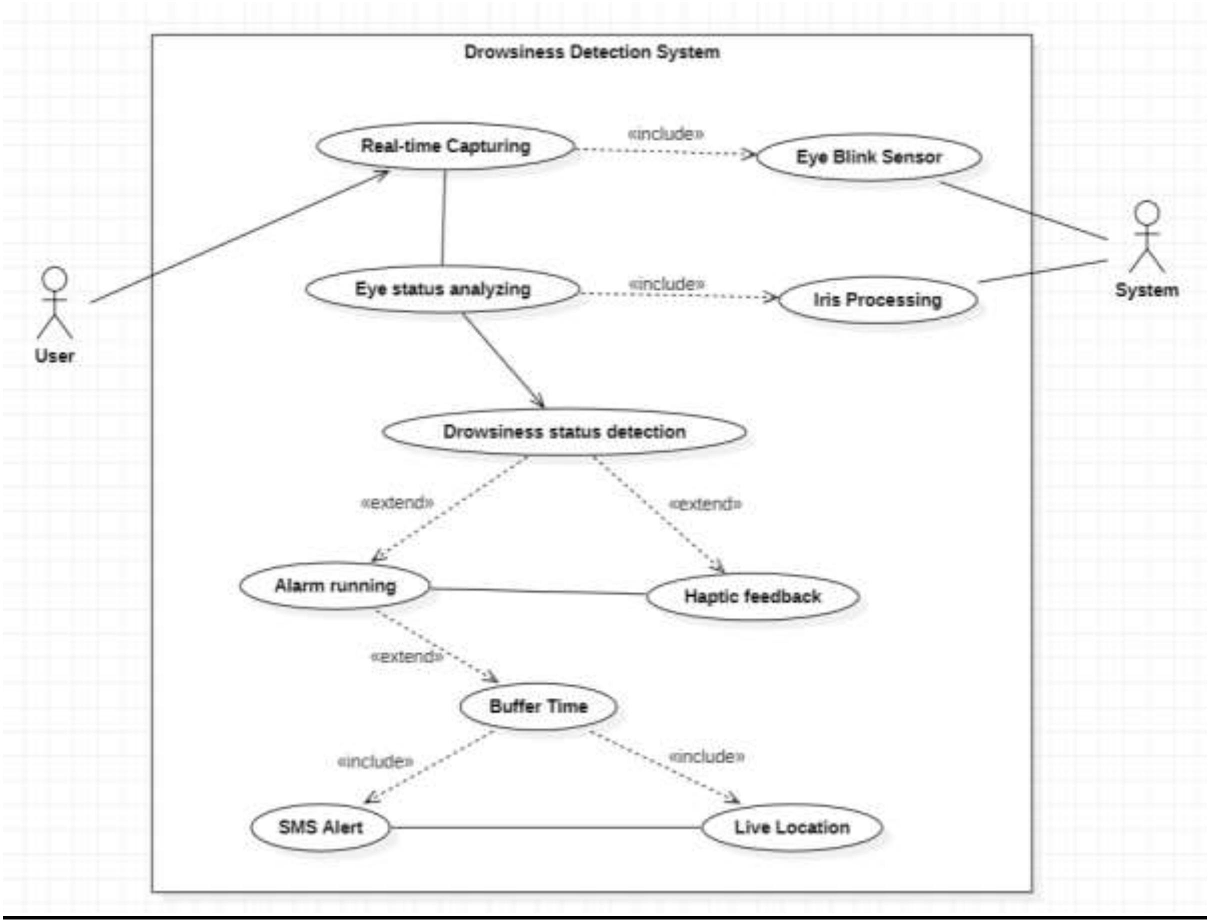

Fig 3.2 Use Case Diagram

2. Activity Diagram

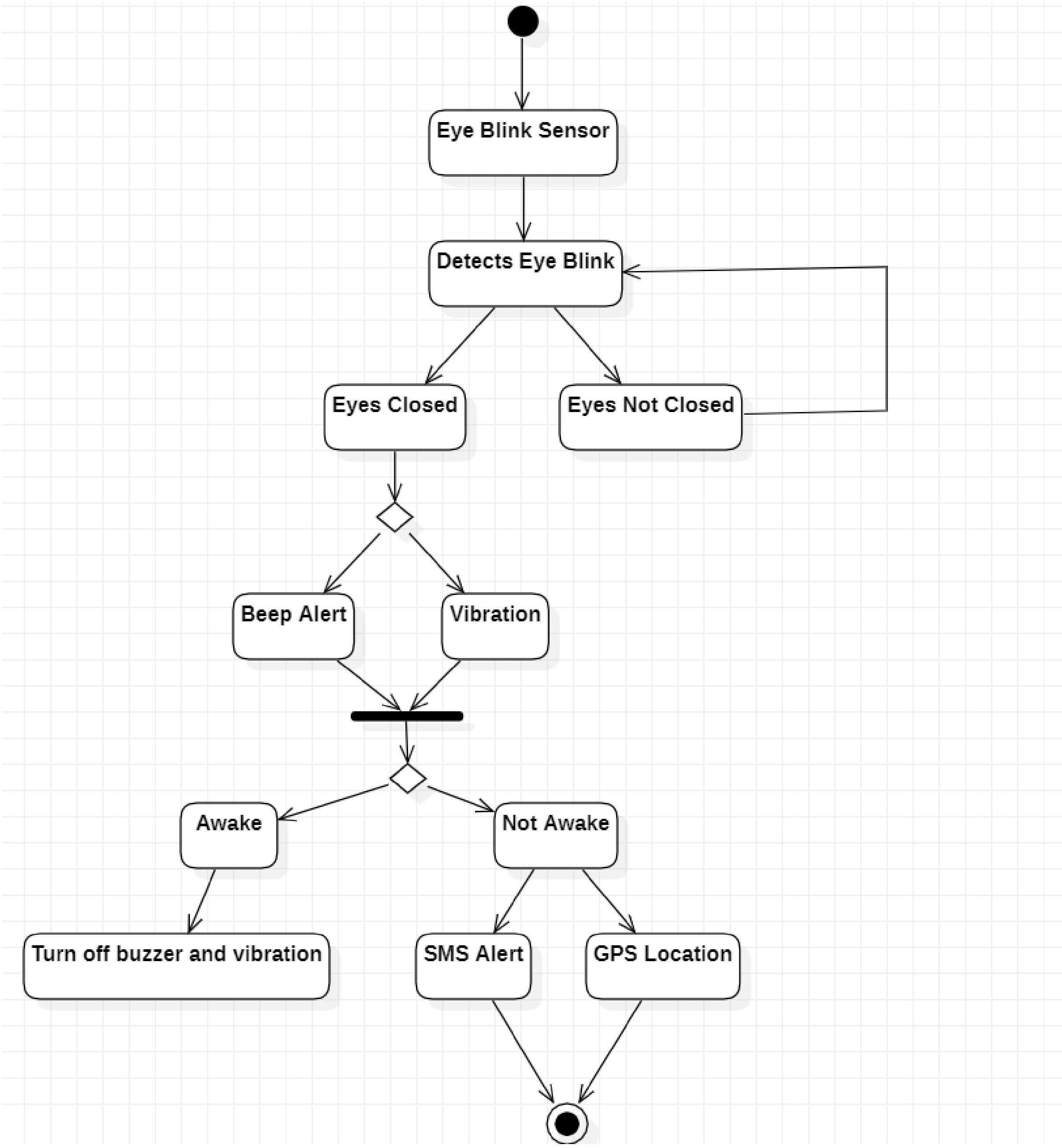

Fig 3.3 Activity Diagram

3. State Chart Diagram

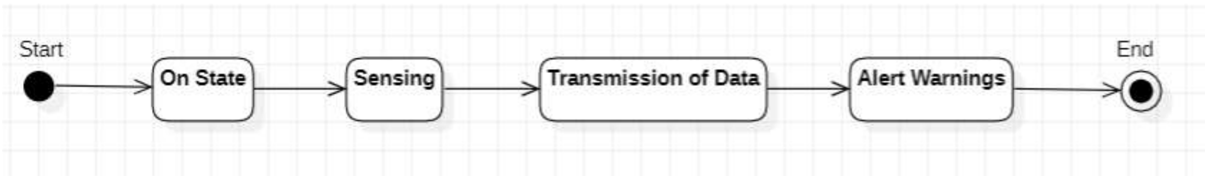

Fig 3.4 State Chart Diagram

4. **<u>Class Diagram</u>**

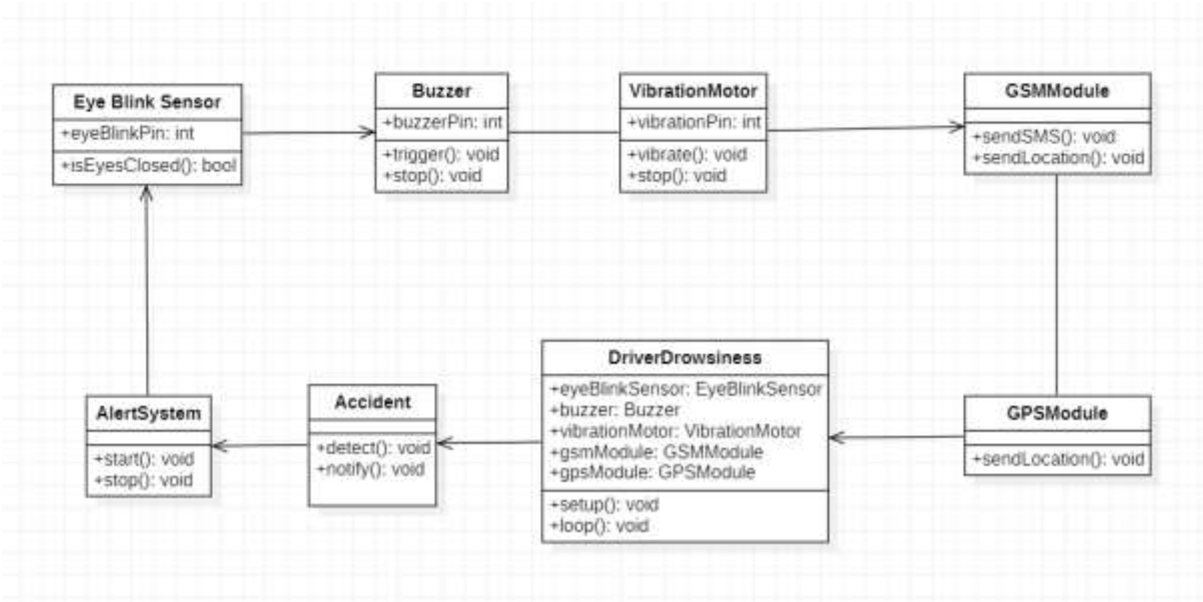

Fig 3.5 Class Diagram

CHAPTER 4 : SYSTEM DESIGN

1. Basic Modules

Modules used in creating our IoT Project for Hardware are as follows:

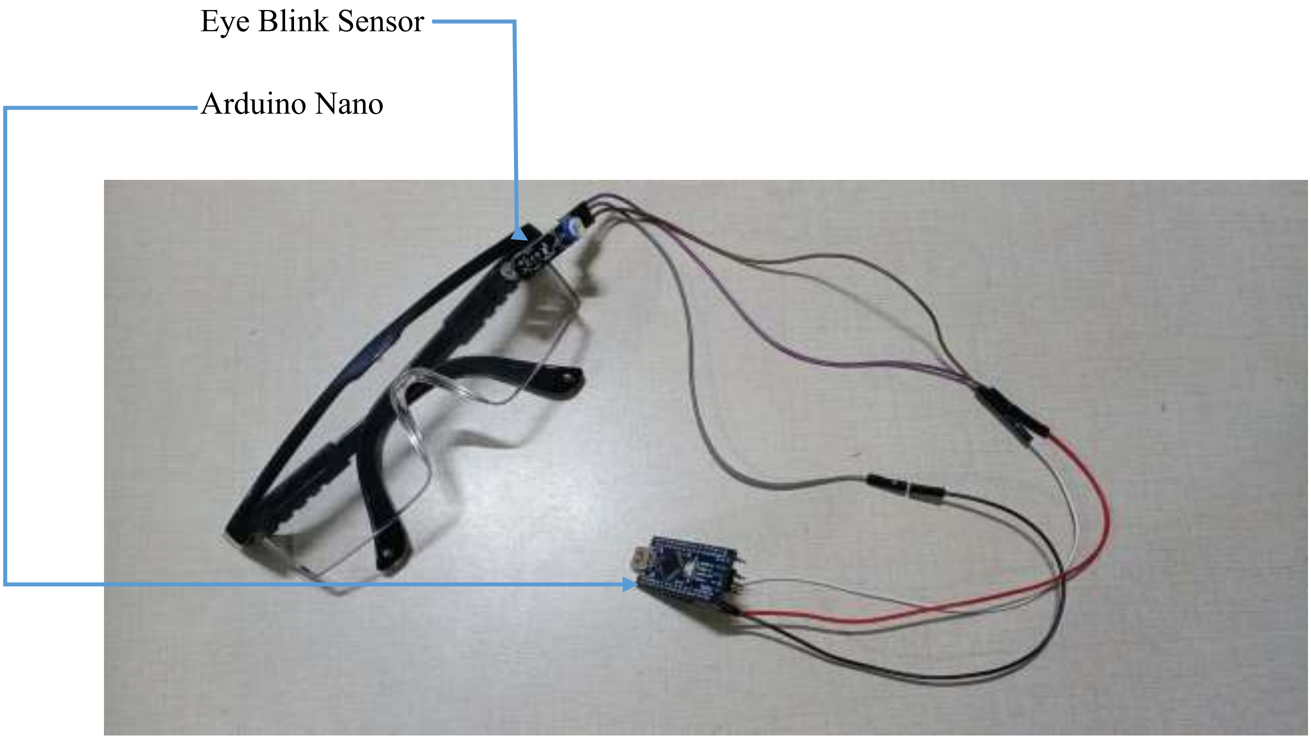

Fig 4.1 Hardware Module

A. Description of Desired Modules

Arduino Nano:

- Microcontroller ATMega328P
- Operating Voltage
- Input Voltage (Recommended): 7-12V
- Input Voltage (Limits) : 6-20V
- Digital I/O Pins : 14
- PWM Pins : 6
- Analog Input Pins : 8
- DC Current Per I/O Pin : 40mA
- DC Current Per I/O Pin : 50mA
- Flash Memory 32KB (ATMega328P) of which 0.5KB used by Bootloader
- SRAM 2KB (ATMega328P)
- EEPROM 1KB (ATMega328P)
- Clock Speed 16 MHz
- Dimensions 18mm x 45mm
- USB: micro-USB port for power, programming and debugging
- Processor ATMega328P 16MHz

- Weight :7g
- Width : 18mm
- Length : 45mm

Eye Blink Sensor:

- Operating Voltage
- Output : Digital TTL output, high level when eyes are closed, low level when eyes are open
- Detection Angle : 15º – 30º
- Sensitivity : Adjustable, typically set to detect blink durations of 100ms or more
- Dimensions : Small form factor, typically less than 2cm in diameter and 1cm in height
- Power Consumption : Low power consumption, typically less than 1mA
- Interface : Easy to use, can be directly connected to microcontroller or Arduino Board
- Response Time : Fast response time, typically less than milliseconds
- Operating Temperature : Wide operating temperature range, typically -20ºC to 80ºC
- Environmental Conditions : Minimal environmental impact, suitable for use in indoor outdoor environments

B. Description of Desired Features

Arduino Nano

Process all the data collected by the eye blink sensor properly and transmit the data correctly to the other sensors and modules such as vibration motor, buzzer, GSM Module and GPS Module.

Eye Blink Sensor

Sense and detect the signals of eye winks and send the data properly to the Arduino Nano.

2. Data Design

In this phase of designing, the project is divided and explained well in terms of how the product is actually going to work and how the data is collected, and then transmitted over to the other sensors and modules where it will wake up the driver using vibrations and audible alerts and in case of unresponsiveness it will send an SMS alert and drivers live location to the designated contact.

This can be understood better with respect to the data diagram, state chart diagram, sequence diagram present and explain above in chapter:3 and the architectural diagram present and explained in the further procedural design.

3. Procedural Design

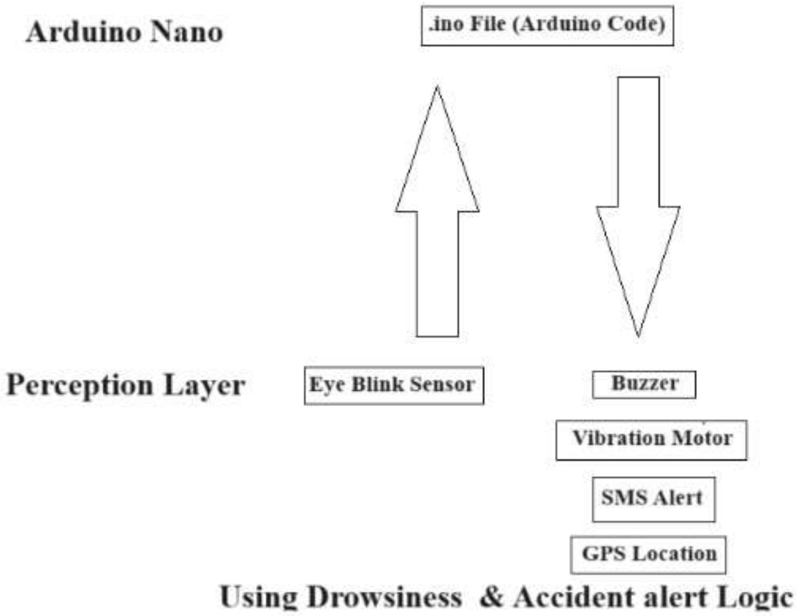

Fig 4.2 Architecture Diagram

4. User Interface Design

There is no app created as this is an Internet of Things (IoT) device. We have built a prototype device where the user will use Eye Blink Sensor and Switch Button to communicate.

Fig 4.3 Eye Blink Sensor Pinout

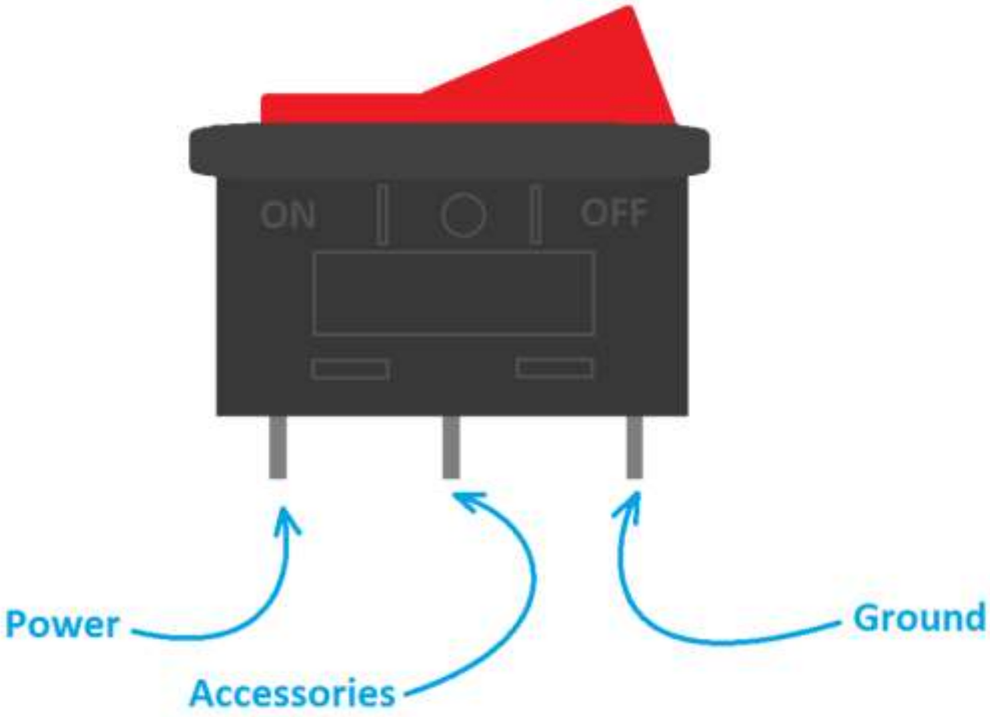

Fig 4.4 Switch Button

5. Security Issues

I. List of Security Issues:

- Data privacy: The system collects sensitive data about the driver's physiological state and location. It is important to ensure that this data is secretly stored and transmitted, protecting the driver's privacy.
- Software Vulnerabilities: The system relies on software to process sensor data and trigger alerts. Identifying and patching software vulnerabilities is crucial to prevent from attackers.
- Emergency Alarm Abuse: The system's emergency alert features, such as sending SMS notification, could be abused if not properly secured. Implementing strict access controls and usage policies can prevent misuse of these features.
- False Alarms: The system must be designed to minimize false alerts, as frequent false alerts can lead to complacency and reduce the system's effectiveness.

II. Plan of Action to treat these issues:

- Data Privacy: Implement end-to-end encryption for data transmission, ensuring that sensitive information such as physiological state and location is securely encrypted and protected during transmission and storage.
- Software Vulnerabilities: Regularly update the system's software and firmware to patch known vulnerabilities and ensure the latest security patches are applied.

- Emergency Alert Abuse: Implement strict access controls and usage policies to prevent unauthorized use of the system’s emergency alert features.
- False Alarms: Implement algorithms and filters to minimize false alarms, such as learning models to detect patterns of drowsiness and filter out false patterns.

CHAPTER 5 : IMPLEMENTATION AND TESTING

1. Implementation Approaches

The implementation approaches that are used for developing this IoT project are as follows:

1. Hardware Development: The first approach is to develop the hardware needed for the project. This includes designing the electronic circuits and selecting the necessary components such as sensors, microcontrollers, actuators, and wireless communication modules.
2. Software Development the software for the Arduino Nano to interface with hardware components and implement the necessary algorithms for driver drowsiness detection and emergency alerting. This involves programming C/C++ using the Arduino IDE or other development environments.
3. Integration and Testing: Integrate the hardware and software components, and conduct thorough testing to ensure the system functions as intended. Thus includes the testing accuracy of the eye blink sensor, the responsiveness of the alerting system, and the reliability of the emergency notification feature.
4. User-Centred Design: The fifth approach is to use a user-centred design approach. This involves designing the project with the user in mind and gathering feedback from users throughout the development process to ensure that the project meets their needs and expectations.

A. Define the implementation plan

- Application used for the project:
  We have created an assistive device that will help people to stay awake and ensure safety while driving. This will reduce the chances of accidents while driving. It helps people to drive safely and stay awake while driving.
- Hardware components:
  This part of the project was quite challenging at first as we were trying to create a product that were to be worn be people for a major amount of the day and at the same time find the hardware that was capable and compatible with other hardware components that were to be used. Now seeing all these challenges in front of us when starting the project, we first went for an Arduino UNO as we were quite used to the module and the hardware itself cause of IoT as a subject that was taught to us in Semester-5, and to go with It Arduino Nano, Eye Blink Sensor to get data from simple winks of the eye, Vibration Motor and Buzzer as actuators to give output for the drowsiness of the driver. But then we had to switch the Micro Controller to Arduino Nano due to some challenges faced while building a prototype for the working glasses which is explained further in the document.

  LIST OF HARDWARE USED:

  Arduino Nano:

Fig 5.1

Eye Blink Sensor:

Fig 5.2

Vibration Motor:

Fig 5.4

Switch Button:

Fig 5.3

Buzzer:

Fig 5.4

Switch Button:

Fig 5.5

GSM Module:

Fig 5.6

GPS Module:

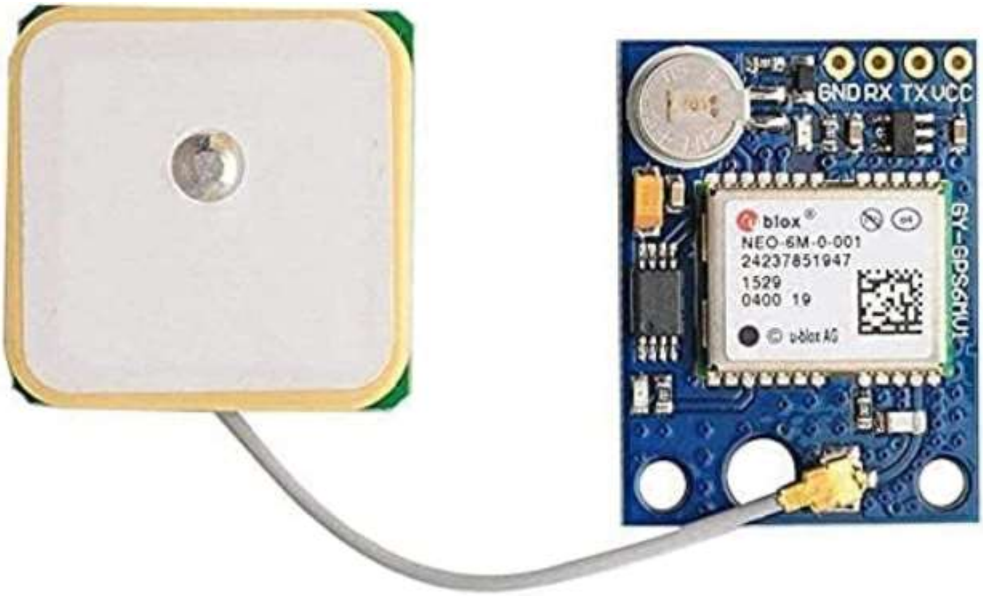

Fig 5.7

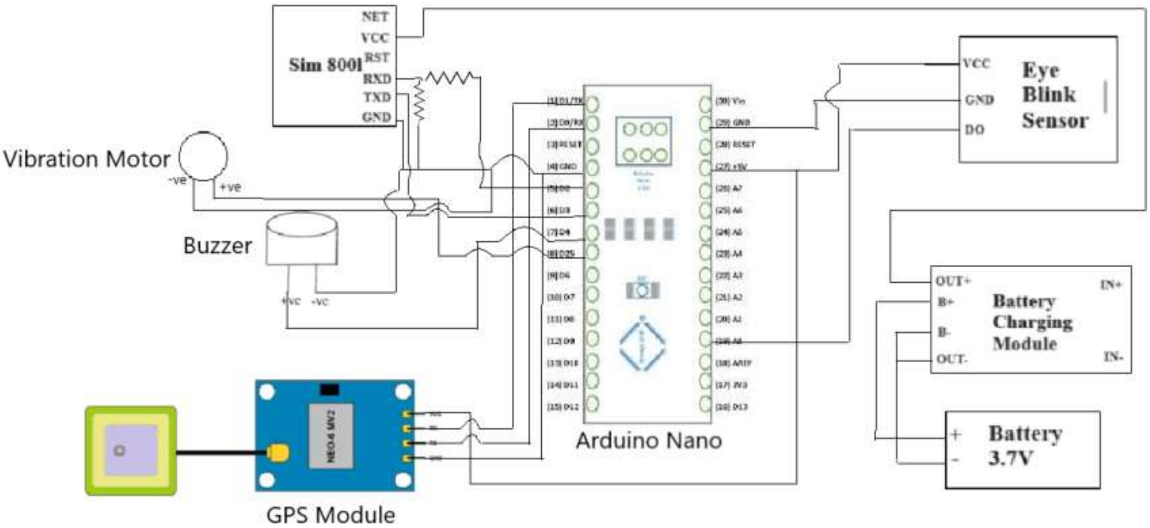

Fig 5.8 Circuit Diagram

This is the circuit diagram for the project that is made by me, there were no such difficulties faced once the hardware was specified and all the connections were simple and not that complicated.

- Product Prototype

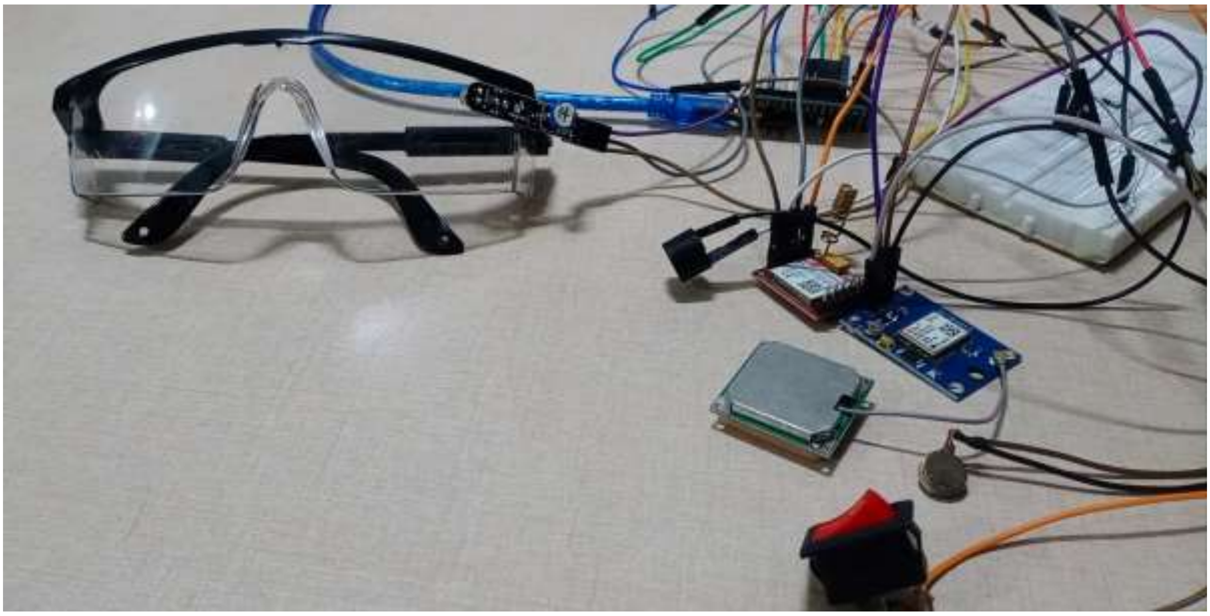

Fig 5.9 Final Prototype

The product prototype building was a little more challenging as it was to be worn as a normal pair of glasses and it could not be very heavy which would lead to other issues such as headaches, etc. The major challenge here at first was to fit a whole Arduino UNO R3 MCU, a 9v battery etc. to be fitted which made the glasses heavier and not very comfortable to wear for more than a few minutes. So then to solve the problem and having to go through a lot of research we chose to switch to Arduino Nano from Arduino UNO R3 which drastically changed the way our glasses looked, weighed, felt, and improved the user experience by a margin. This made it easier as well to fit and design the prototype way more systematically.

B. State the standards and protocols used in implementation

- **Serial Communication Protocol (UART):** UART is commonly used for serial communication between the Arduino Nano and peripheral devices such as the eye blink sensor, GSM module, and GPS module. It facilitates the transmission of data in a serial format, allowing for reliable communication between different components of the system.
- **Global System for Mobile Communications (GSM)**: GSM is a standard protocol for cellular communication used for sending SMS notifications in the event of an emergency. The GSM module interfaces with the Arduino Nano to send SMS alerts to designated contacts, providing real-time notification of potential accidents.
- **Global Positioning System (GPS):** GPS is a satellite-based navigation system used to determine the geographical location of the vehicle. The GPS module interfaces with the Arduino Nano to provide accurate location data, which is included in the SMS alerts sent during emergencies, enabling recipients to pinpoint the vehicle's location.

5. Coding Details and Code Efficiency

The code for hardware is written in C++ and coded on Arduino IDE.

```cpp
#include <SoftwareSerial.h>
#include <TinyGPS++.h>

#define GSM_RX 2
#define GSM_TX 3
#define BUZZER_PIN 4
#define VIBRATION_PIN 5
#define EYE_BLINK_PIN 6
#define SWITCH_PIN 7

SoftwareSerial gsmSerial(GSM_RX, GSM_TX);

TinyGPSPlus gps;
bool isDriverAwake = true;
bool isSwitchOn = false;
unsigned long lastBlinkTime = 0;
unsigned long lastVibrateTime = 0;

void setup() {
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(VIBRATION_PIN, OUTPUT);
  pinMode(EYE_BLINK_PIN, INPUT);
  pinMode(SWITCH_PIN, INPUT_PULLUP);

  Serial.begin(9600);
  gsmSerial.begin(9600);
}

void loop() {
  checkEyeBlink();
  checkAccident();
  checkSwitch();
}

void checkEyeBlink() {
  if (digitalRead(EYE_BLINK_PIN) == LOW) {
    if (!isDriverAwake) {
      isDriverAwake = true;
      turnOffAlert();
    }
  } else {
    if (isDriverAwake) {
      isDriverAwake = false;
      sendAlert();
    }
  }
}

void checkAccident() {
  if (!isDriverAwake) {
    if (millis() - lastVibrateTime > 15000) {
      sendLocation();
    }
  }
}

void checkSwitch() {
  if (digitalRead(SWITCH_PIN) == LOW) {
    isSwitchOn = true;
    turnOffAlert();
  } else {
    isSwitchOn = false;
  }
}

void turnOffAlert() {
  digitalWrite(BUZZER_PIN, LOW);
  digitalWrite(VIBRATION_PIN, LOW);
}
```

```
void sendAlert() {
  digitalWrite(BUZZER_PIN, HIGH);
  digitalWrite(VIBRATION_PIN, HIGH);

  if (!isSwitchOn) {
    gsmSerial.println("AT+CMGF=1");
    delay(1000);
    gsmSerial.println("AT+CMGS=\"+919321092120\"");
    gsmSerial.println("Alert: Driver is not responding!");
    delay(1000);
    gsmSerial.println((char)26);
    delay(1000);
  }
}

void sendLocation() {
  if (gps.location.isValid()) {
    String latitude = String(gps.location.lat(), 6);
    String longitude = String(gps.location.lng(), 6);
    String googleMapsUrl = "https://www.google.com/maps?q=" + latitude + "," + longitude;

    if (!isSwitchOn) {
      gsmSerial.println("AT+CMGF=1");
      delay(1000);
      gsmSerial.println("AT+CMGS=\"+919321092120\"");
      delay(1000);
      gsmSerial.println(googleMapsUrl);
      delay(1000);
      gsmSerial.println((char)26);
      delay(1000);
    }
  }
}

void serialEvent() {
  while (Serial.available() > 0) {
    if (gps.encode(Serial.read())) {
      sendLocation();
    }
  }
}
```

A. Code of Main Logic

This code is for a project that uses a Arduino Nano to detect drowsiness of driver using eye blink sensor and then display the corresponding message on an LCD screen. Here is a brief explanation of the logic of the code:

- Libraries and Definitions: The sketch includes the necessary libraries (SoftwareSerial and TinyGPS++) and defines pin numbers for GSM communication, buzzer, vibration motor, eye blink sensor, and switch.
- Global Variables: Global variables are declared to track the driver's state (isDriverAwake), the switch state (isSwitchOn), and the last blink and vibration times.
- Setup Function: Initializes the pin modes and begins serial communication for both the Arduino and the GSM module.
- Loop Function: Contains three main functions called in a loop: checkEyeBlink(), checkAccident(), and checkSwitch().
- Check Eye Blink: Monitors the state of the eye blink sensor. If the driver is detected as awake, it turns off any alerts. If the driver is detected as asleep, it triggers an alert.
- Check Accident: Checks for an accident scenario (driver is asleep) and sends the location if there hasn't been a recent vibration.

- Check Switch: Monitors the state of the switch. If the switch is on, it turns off any alerts.
- Turn Off Alert: Function to turn off the buzzer and vibration motor.

- Send Alert: Function to trigger an alert (buzzer and vibration) and send an SMS alert to the designated contact if the switch is off.
- Send Location: Function to send the GPS location to the designated contact if the switch is off.
- Serial Event: Event handler that runs when data is received on the Arduino's serial port. It decodes the GPS data and sends the location.

B. Code Efficiency

The code written overall is very efficient as it meets up with the requirement specifications of the project and the prototype created, once the user gets a well hold on the device.

How well does the code meets with the Requirement Specifications?
Yes, I would say that the code used to program the hardware is truly in its most optimum as it meets the specific requirements of the project to be helpful to the one’s in need and the speed, ease, etc to understand and learn the product is clarified through the code.

Drawbacks of the code :
The delay statements in the code could be improved, as they block the program execution for a fixed amount of time, which could lead to unresponsive behaviour if the delays are too long. It may be better to use non-blocking techniques, such as the millis() function, to manage timing in the code. This change especially is due to a major reason that using millis() instead of delay() allows the program to continue executing other tasks while waiting for a specific amount of time to pass.
We will surely improve this in the future assessments.

6. Testing Approach

For Real time Driver Drowsiness Detection and Accident Alert System, it is important to do manual testing and hence, I have gone with manual testing. The Most important aspect to think about when testing real time detection is that we should cover up all the possible scenarios possible in the process.

A. Functional Testing

Functional testing is a type of testing that is performed to ensure that a software system or application behaves according to its functional requirements. It involves testing the features and functionality of the system by providing input values and verifying the corresponding output values. The main goal of functional testing is to verify that the software system is performing its intended functions correctly and meets the requirements specified by the client or end-user. Functional testing can be performed manually or using automated testing tools, and it can be done at various levels of the software development life cycle, such as unit testing, integration testing, system testing, and acceptance testing.

For our project the major functional testing were conducted to ensure that the testing focuses on evaluating each functional aspect of the system, including drowsiness detection, alert triggering, SMS notification, and GPS location tracking. During functional testing, the system is subjected to various scenarios and conditions to assess its performance under different circumstances. Similarly, the alert triggering mechanism is tested to ensure that alerts are activated promptly and effectively when drowsiness is detected. Additionally, functional testing involves verifying the reliability and accuracy of SMS notifications, ensuring that alerts are sent to designated contacts with the correct information, including the vehicle's GPS location.

1. **User Acceptance Testing or Beta Testing**

   This testing was done by 3 candidates including me and 2 more unbiased personals where the major role was to test the functional requirements are met to the requirement specifications set by us and to get the best result off our project.

2. **Unit Testing**

   The Unit Testing was conducted after every step of the way, to see if every single component of the project is working fine without any issues:

- Eye Blink Sensor was tested for the range and responsiveness.
- Vibration Motor and Buzzer were tested to see if the respond well with the eye blink sensor.
- The logic of the code was tested on laptop to determine whether the code works effectively and accurately.
- Arduino Nano was checked to determine whether it receives the eye blink patterns from Eye Blink Sensor for drowsiness detection.
- Both Eye Blink Sensor and Arduino Nano were checked if they received sufficient power from batteries.
- And just like that every module of the project was tested individually to see if there are any errors or problems before integrating them together.

3. **Integration Testing**

   In Integration testing, the test was conducted by assembling all the modules of the project and integrating them together with each other and see if the code if working fine and if not then improvising the code and making changes as per the needs so that we get the most efficient code and meet our requirements with ease.

   Firstly, some minor twerks and changes were made to the code after figuring out that we had to replace the Arduino UNO with Arduino Nano, as the prototype was getting bulkier due to all the excess weight added by the huge MCU such as UNO. We integrated the code in Arduino Nano and checked whether it runs without any error. After we connected Eye Blink Sensor with Arduino Nano and ran the code to check whether it detects eye blink. After that we integrated the Eye Blink Sensor and Arduino Nano with buzzer and vibration motor for alert warnings. But there was issue in vibration motor while integrating which then the problem was resolved later by resoldering the microcontroller Arduino Nano.Then we integrated GSM and GPS Module with Arduino Nano and other sensor as well as the switch button to provide SMS alert notification and GPS location.

Integrating Arduino Nano with the hardware and the software part of the project was way more simpler and the prototype of the product was also on the underside of the weight that we were expecting the product to be.

B. Non-Functional Testing

Non-functional testing is a type of software testing that verifies the non-functional requirements of a system, such as performance, reliability, security, usability, and scalability. The main purpose of non-functional testing is to evaluate how well a system meets its nonfunctional requirements and to identify any issues or defects that may impact the system's quality or user experience. For our project we have focused on non-functional test such as load testing, stress testing, performance testing, usability testing and compatibility testing to see that the project from our end is at its best when provided or supplied to the users.

1. Performance Testing

Here various tests were performed in different cases to see that the performance outcome of the product is as required by our standards.
Different test scenarios tested under performance testing are as follows:

- Load testing: This test, simulate a large number of users or data to assess the system's response and determine its capacity.
- Stress testing: This Test tests, the system under extreme conditions, such as high traffic or heavy load, to assess its behaviour and response time.
- Volume testing: This Test tests, the system with a large amount of data to assess how it handles data storage, processing, and retrieval.
- Endurance testing: This Test tests, the system under extended use to ensure its stability and performance over time.
- Concurrency testing: This Test tests, the system's ability to handle multiple users and requests simultaneously.
- Latency testing: This Test tests, the system's response time to requests to ensure that it meets performance requirements.
- Throughput testing: This Test tests, the system's ability to process and deliver data to ensure that it meets performance requirements.

2. **Scalability Testing**

Scalability testing is a type of non-functional testing that evaluates a system's ability to perform well under an increased workload or when the system is scaled up. It is used to determine how well a system can handle a growing number of users, transactions, or data volumes. The goal of scalability testing is to ensure that a system can scale up and handle a large amount of work without crashing or degrading in performance.

Scalability testing can be performed in two ways: vertical and horizontal. Vertical scalability testing involves adding more resources, such as RAM, CPU, or storage, to a single machine. Horizontal scalability testing involves adding more machines to a system to distribute the load. This type of testing is critical for systems that need to handle increasing user demands or growth in the volume of data or transactions.

And through this we got an outcome that the project that we developed is highly scalable both horizontally and vertically and can be made more specific depending on the updates we get from the users over time.

3. **Portability Testing**

Portability testing is the type of testing that is conducted to determine the system's ability to be moved from one environment to another without any difficulty. It is a type of non-functional testing that evaluates whether the software can function efficiently in different environments, platforms, or devices. The objective of this testing is to verify that the application can be easily and quickly installed, configured, and run in different environments. It is essential to perform portability testing to ensure that the application is accessible and usable on different devices and operating systems, making it easy for users to switch to new devices or platforms without any difficulty.

This was a major task at hand as our project is a glasses that are to be worn by the user to get the best out of it and at the same time for us to get a good client feedback. We had to make sure that the project we are creating is highly portable both in hardware and prototypical manner as well as software and OS compatible manner which we were able to achieve at the end after getting the testing results.

Black Box Testing

Black box testing was conducted on the given code by testing the functionality of the code without any knowledge of its internal workings. This was done by providing inputs and verifying the outputs against the expected results.

The functioning of the Buzzer, vibration motor, GSM Module, GPS Module and the eye blink sensor were tested by providing the appropriate inputs.

C. **White Box Testing**

White-box testing involves testing the internal workings of the code. Here are some possible test cases for white-box testing of the provided code:

• We verified that all pins are set up as expected.

• We verified that the correct buzzer pins are used.

• We verified that the correct eye sensor pins are used.

• We verified that the correct vibration motor pins are used.

• We verified that the correct alert mechanisms are triggered when both eyes are closed.

• We verified that the SMS alert and GPS location is sent to the designated contact.

• We verified that the delay function is used appropriately to avoid unnecessary blocking.

Black box testing and white box testing are two different approaches for testing software applications. Black box testing is focused on testing the functionality of the software without any knowledge of the internal code, while white box testing involves analysing the internal code and structure of the application to test it.

In the case of the given code, black box testing involves analysing the behaviour of the code when the input and output parameters are provided. In this approach, the code is treated as a black box and testing is carried out to check if the output of the code is correct for a given input.

In the analysis performed on the given code, black box testing was used to identify the functional behaviour of the software by testing its inputs and outputs. This helped to verify that the software met the requirements of the specifications and performed its intended functions. White box testing, on the other hand, was used to analyse the software's internal structure and workings to identify the defects in the code, architecture, and logic. This helped to ensure that the software's internal components and modules were functioning correctly and met the requirements of the specifications.

In general, black box testing is more focused on the end-user's perspective and the software's external behaviour, while white box testing is more focused on the software developer's perspective and the software's internal structure and workings. Both methods are important for ensuring the quality of software and can be used together to provide comprehensive software testing.

• In the case of our code, black box testing could be used to ensure that the input from the sensors is properly handled, and that the expected output is produced based on that input. White box testing could be used to ensure that the code is structured and written correctly, with proper error handling and optimal performance.

• Both types of testing are important for ensuring the quality and reliability of the code. Black box testing helps ensure that the code meets the requirements and works as intended, while white box testing helps ensure that the code is written correctly and is efficient.

7. Test Cases

Test cases are a set of pre-defined inputs, conditions, and expected outcomes that are used to verify that a system, software application, or product is functioning as intended. The purpose of creating test cases is to ensure that the software product or system is working as per the specified requirements and that it meets the quality standards before it is delivered to the end-users. Test cases help in identifying defects or issues in the software early in the development process, which reduces the cost and time required to fix them.

Here are a few test cases where each unit and process was tested while developing the project.

TEST CASE			
System Name :	DRIVER DROWSINESS DETECTION		
Module Code :	HW001 - HARDWARE		
Pass	5	Pending	0
Fail	1	Number of test cases:	6

ID	Test Case Description	Test Case Procedure	Expected Output	Acutal Output	Test date	Result	Note
HW1	Test Eye Blink Sensor functionality	Check whether the LED is blinking or not	LED is blinking	Same as expected	21-02-2024	Pass	
HW2	Test Vibration Motor functionality	Run a simple code for vibration motor on Arduino IDE	Vibration Motor vibrates	Same as expected	21-02-2024	Pass	
HW3	Test Buzzer functionality	Run a simple code for buzzer on Arduino IDE	Triggers the Buzzer	Same as expected	22-02-2024	Pass	
HW4	Test GSM Module functionality	Run a simple code for GSM SMS alert on Arduino IDE	Checking AT Commands	Same as expected	24-02-2024	Pass	
HW5	Test GPS Module functionality	Run a simple code for GPS Location on Arduino IDE	LED is blinking	Same as expected	26-02-2024	Pass	
HW6	Test Eye Blink Sensor with varying blink speed	Run a simple code for Eye Blink Sensor on Arduino IDE	Sensor output changes with eye blinking	Detects false eye blinks sometimes	26-02-2024	Fail	Eye Blink Sensor somrtimes detects eye blink with varying blink speed

Table 5.1 Test Module - HW001

TEST CASE			
System Name :	SIGN LANGUAGE DETECTION		
Module Code :	SW002 - SOFTWARE		
Pass	5	Pending	0
Fail	2	Number of test cases:	7

ID	Test Case Description	Test Case Procedure	Expected Output	Acutal Output	Test date	Result	Note
SW1	Test eye blink detection algorithm	Run a simple code for Eye Blink Sensor on Arduino IDE	System accurately detects eye closure events	Same as expected	23-02-2024	Pass	
SW2	Test drowsiness threshold logic	Run a simple code for Eye Blink Sensor with threshold logic on Arduino IDE	System triggers alert (vibration and/or buzzer) when eye closure exceeds a set threshold	Same as expected	25-02-2024	Pass	
SW3	Test SMS sending functionality	Run a simple code for SMS alert on Arduino IDE	System identifies and displays signs from each individual	Same as expected	27-02-2024	Pass	
SW4	Test drowsiness threshold logic with gradual eye closure	Run a simple code for Eye Blink Sensor with exceeding threshold logic on Arduino IDE	System triggers alert only after exceeding the threshold	Same as expected	27-02-2024	Pass	
SW5	Test SMS functionality with weak cellular signal	Run a simple code for GPS Module on Arduino IDE	System attempts to send location despite potential delays	System fails to send location entirely	28-02-2024	Fail	Issue because not connected to satellite signal as it needs open ground area to receive signals from satellite
SW6	Test SMS functionality with weak cellular signal	Run a simple code for GSM Module on Arduino IDE	System attempts to send SMS despite potential delays	System fails to send SMS entirely	29-02-2024	Fail	Issue because of network connectivity through GSM Module
SW7	Test software behavior with unexpected sensor data	Run a simple code for Eye Blink Sensor on Arduino IDE	System identifies and handles invalid sensor data gracefully (e.g., error message)	Same as expected	29-02-2024	Pass	

Table 5.2 Test Module - SW002

TEST CASE							
System Name :	SIGN LANGUAGE DETECTION						
Module Code :	IT003 - INTEGRATION						
Pass	7	Pending	0				
Fail	2	Number of test cases	9				

ID	Test Case Description	Test Case Procedure	Expected Output	Acutal Output	Test Date	Result	Note
IT1	Testing the overall system		System detects drowsiness and triggers alert (vibration and/or buzzer) as well as send sends SMS and location	Same as expected	02-03-2024	Pass	
IT2	Simulate eye closure with Eye Blink Sensor		System detects drowsiness and triggers alert (vibration and/or buzzer)	Same as expected	03-03-2024	Pass	
IT3	Simulate prolonged drowsiness scenario		System detects drowsiness, triggers alert, and sends SMS with GPS coordinates	Same as expected	03-03-2024	Pass	
IT4	Test system functionality under different lighting conditions (bright light, low light)		System accurately detects drowsiness regardless of lighting conditions	System maintains the functionality but the accuracy falls under stress test	04-03-2024	Fail	Due to inadequate cooling solutions, leading to instability
IT5	Test system functionality with different users (people wearing glasses, different eye		System accurately detects drowsiness for various users	Same as expected	05-03-2024	Pass	
IT6	Test system response to hardware failures (e.g., disconnected		System detects hardware failure and provides an appropriate error message	Same as expected	05-03-2024	Pass	
IT7	Simulate eye closure with object in front of sensor (e.g., hand)		System does not trigger drowsiness alert	System triggers drowsiness alert even with object blocking sensor	06-03-2024	Fail	Due to intensive computational tasks, causing thermal buildup beyond the device's
IT8	Test system response to sudden light changes (bright		System recovers quickly and resumes normal operation	Same as expected	06-03-2024	Pass	
IT9	Test system with multiple users consecutively		System accurately detects drowsiness for each user	Same as expected	08-03-2024	Pass	

Table 5.3 Test Module - IT003

8. Modification and Expected Improvements

The only modification that had to be done to improve the project was to eye detection algorithm using machine learning and to replace separate components for a MCU, i.e ESP32. Which automatically helped us a lot in creating the prototype and making it easier and to detect drowsiness of multiple users to wear for a long day as well as helped us to complete the project with way less hardware components and easier coding with libraries present for the open source ESP module.

CHAPTER 6 : RESULTS AND DISCUSSIONS

1. Test Reports :

TEST REPORT					
	Date :	Date of creating test report			
	No	**Module code**	**Pass**	**Fail**	**Total Number of test cases**
	1	HW001 - HARDWARE	5	1	6
	2	SW002 - SOFTWARE	5	2	7
	3	IT003 - INTEGRATION	7	2	9
		Sub total	17	5	22
		Test coverage	**100.00**	%	
		Test successful coverage	**77.27**	%	

Table 6.1 Overall Test Report

2. User Documentation :

User Manual for DrowsiShield:

Introduction:

DrowsiShield is an IOT device that helps people with drowsiness problems. This device helps them stay awake and prevent from accidents.

1. **Components:**

- Arduino Nano
- Switch Button
- Eye Blink Sensor
- Buzzer and vibration motor
- GSM and GPS Module

2. **Operation**

- Connect battery to Arduino Nano.
- Wear glasses with eye blink sensor for drowsiness detection.
- System monitors driver's eye movements for drowsiness.
- Upon detecting drowsiness, activates buzzer and vibrating motor for alert.
- Continues alerting if driver remains unresponsive.
- Activates emergency alert, sending SMS with GPS location if high risk of accident.
- Includes manual override option to disable emergency alert.

3. **Maintenance:**

- Charge the glasses after use.
- Avoid exposing the device to water or high temperatures.
- Keep the device clean and dry.

4. Troubleshooting:

- If the device fails to operate, you can try the following:
- Make sure the device is connected to battery.
- Restart the device.

3. <u>Cost Estimation :</u>

Cost estimation models are mathematical algorithms or parametric equations used to estimate the costs of a product or project. The results of the models are typically necessary to obtain approval to proceed, and are factored into business plans, budgets, and other financial planning and tracking mechanisms.

- **The Development Model**

  COCOMO (Constructive Cost Model) is a regression model based on LOC viz. number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.

- **Key Parameter**
  a. Efforts - measured in person months units
  b. Schedule - measured in span of months or weeks

  To estimate the effort and development time, COCOMO uses the same equations but with different coefficients (a, b, c, d in the effort and schedule equations) for each development mode. Types are as follows :

- Organic System
- Semi – detached System
- Embedded System

The basic COCOMO equations take the form

- Effort Applied (E) = ab (KLOC) bb [person-months]
- Development Time (D) = cb (Effort Applied) db [months]
- People Required (P) = Effort Applied / Development time [count]

Where, KLOC is the estimated number of delivered lines (expressed in thousands) of code for a project. The coefficient ab, bb, cb and db are given in the following table:

Software Project	a_b	b_b	c_c	d_d
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

COCOMO Model for “DrowsiShield”

a. Effort : 3.07

b. Time for development :

Where,

Effort = Number of staff months

(SM) Size = Number of source

lines of code

Time = Total number of months required to complete the project

The Project Code for DrowsiShield contains 200 Lines of code

Since, we know that 1000 Lines of Code = 1 KLOC (K - Kilo -

10^3) Therefore, the project consists of 1.265 KLOC.

Effort = 3.07 SM

Time for development = 2.5 * $(0.494)^{0.35}$ = 1.93 Months » 20 Months

Cost per Month = Rs.10000/-

Total Cost of the Project = Cost per Month * Time required for the development project

= 10000* 1.9

= Rs. 19,000

Additional cost of Hardware = 1800

Total = 19000+1800

= Rs. 20,800

CHAPTER 7 : CONCLUSIONS

1. Conclusion

In conclusion, the development of a driver drowsiness detection system incorporating hardware components such as the Arduino Nano, eye blink sensor, GSM module, and GPS module represents a significant advancement in enhancing road safety and mitigating the risks associated with drowsy driving. By leveraging innovative technologies and intelligent algorithms, this system aims to detect signs of driver fatigue in real-time, alerting the driver and designated contacts to prevent potential accidents. Through a comprehensive approach encompassing software development, hardware integration, and rigorous testing, the system offers a practical solution to address the pressing issue of drowsy driving, ultimately saving lives and reducing the economic and societal costs of road accidents. Furthermore, the economic feasibility analysis indicates promising returns on investment, with potential cost savings outweighing the initial investment in system development and deployment. As we move forward, continued research, refinement, and adoption of such systems hold the promise of a safer and more secure future for all road users.

2. Limitations

- False Positives: The system may occasionally trigger alerts due to factors other than driver drowsiness, such as sudden head movements or environmental conditions, leading to unnecessary interruptions and potential driver distraction.
- Sensor Accuracy: The accuracy and reliability of the sensors, particularly the eye blink sensor, may vary depending on factors such as lighting conditions, driver behaviour, and sensor calibration, impacting the effectiveness of drowsiness detection.
- Limited comfort: Wearing the glasses for extended periods can be uncomfortable, particularly if the wearer has a condition that causes eye strain or fatigue.

3. Future Scope of the Project

The future improvement of our system is including a machine learning module to and artificial intelligence techniques that can improve the accuracy and reliability of drowsiness detection, enabling more precise and proactive alerting mechanisms. Also developing the mobile application for user interface where driver can easily turn off the SMS alert and GPS location as well as the individual data of the driver is maintained based on the behavioural pattern of the driver.

REFERENCES

[1] Detection of Driver Drowsiness Using Eye Blink Sensor - https://ijsrset.com/IJSRSET2051043

[2] Detection of Driver Drowsiness using Eye Blink Sensor - https://www.researchgate.net/publication/326699270_Detection_of_Driver_Drowsiness_using_Eye_Blink_Sensor

[3] DRIVER DROWSINESS DETECTION BASED ON EYE BLINK - https://core.ac.uk/download/pdf/288218625.pdf

[4] A WOW project with Eye blink sensor and Arduino | Arduino Project - https://youtu.be/XpfINvWDnCo?si=hLoaBW36t0g-wH7L(youtube.com)