Software Requirements Specification

for

<Driver Drowsiness Detection System Using Python>

Version 1.0 approved

Prepared by M. Mohamed Faiz

29/04/2024

Table of Contents

1. Introduction

1.1 Purpose

Road conditions that are unpleasant, bad weather, and driver fatigue have all contributed to many accidents. Enforcement officers manning the main thoroughfares and highways in this area claim that drivers who are sleep deprived are nevertheless accountable for about 40% of all collisions in the road. An Internet of Things (IoT) technology aims to avoid numerous accidents caused by fatigued drivers' behavioural and psychological changes by tracking the driver's eye movements. Our effort aims to help solve real-world problems at a reasonable cost. If the driver closes his eyes for more than a second due to fatigue, the buzzer will ring.

1.2 Intended Audience and Reading Suggestions

Countless accidents have occurred as a result of driver exhaustion, tiresome road conditions, and adverse weather conditions. According to enforcement personnel monitoring the motorways and major roads here, sleep-deprived drivers are still responsible for around 40% of all traffic accidents. "This is especially relevant to exhausted drivers who fall asleep at the wheel on our roadways between midnight and 5 a.m.," stated B. Shefiq, Joint Regional Transport Officer, Perumbavur. Only expert drivers, according to B. G. Sreedevi, Chief Scientist at the National Transportation Planning and Research Centre (Natpac) of the Kerala State Council for Science, Technology, and Environment ensures that they get enough sleep before embarking on nighttime travels.

1.3 Product Scope

The Driver Drowsiness Detection System aims to develop a software solution capable of monitoring driver behavior in real-time to detect signs of drowsiness and issue alerts to mitigate potential risks of accidents due to driver fatigue.

Real-time Monitoring: Implement a system to continuously monitor the driver's facial features and behavior using video feed from an in-car camera.

Drowsiness Detection: Utilize machine learning algorithms to analyze facial cues indicative of drowsiness, such as eye closure, head nodding, or changes in facial expression.

Alerting Mechanism: Design an alerting mechanism to notify the driver promptly upon detection of drowsiness, ensuring timely intervention to prevent accidents.

Adaptability: Develop the system to adapt to varying environmental conditions and driver characteristics, ensuring robust performance across different scenarios.

1.4 References

● Arefnezhad, S.; Samiee, S.; Eichberger, A.; Nahvi, A. Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection. Sensors 2019, 19, 943. https://doi.org/10.3390/s19040943
● V B Navya Kiran, Raksha R, Anisoor Rahman, Varsha K N, Dr. Nagamani N P, 2020, Driver Drowsiness Detection, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCAIT – 2020 (Volume 8 – Issue 15),
● Hashemi, M., Mirrashid, A. & Beheshti Shirazi, A. Driver Safety Development: Real-Time Driver Drowsiness Detection System Based on Convolutional Neural Network. SN COMPUT. SCI. 1, 289 (2020).
● Anil Kumar Biswal, Debabrata Singh, Binod Kumar Pattanayak, Debabrata Samanta, Ming-Hour Yang, "IoT-Based Smart Alert System for Drowsy Driver Detection", Wireless Communications and Mobile Computing, vol. 2021, Article ID 6627217, 13 pages, 2021.
● Ashwini, Veda, M., Smitha, S., Krishna, D.H., & Talekar, P. (2020). IOT BASED DRIVER DROWSINESS AND HEALTH MONITORING SYSTEM.
● V B Navya Kiran, Raksha R, Anisoor Rahman, Varsha K N, Dr. Nagamani N P, 2020, Driver Drowsiness Detection, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCAIT – 2020 (Volume 8 – Issue 15),
● Awais M, Badruddin N, Drieberg M. A Hybrid Approach to Detect Driver Drowsiness Utilizing Physiological Signals to Improve System Performance and Wearability. Sensors (Basel). 2017 Aug 31;17(9):1991. doi: 10.3390/s17091991. PMID: 28858220; PMCID: PMC5620623

2. Overall Description

2.1 Product Perspective

Product Origin:
The Enhanced Driver Drowsiness Detection System is a new, self-contained product designed to address a critical safety issue in automotive transportation — driver fatigue. This product is not a follow-on to an existing product family but a novel solution that integrates advanced technological innovations to improve road safety.

Motivation:
The development of this product is motivated by the high incidence of road accidents attributed to driver drowsiness. Existing systems have limitations, particularly in terms of detection accuracy under diverse conditions and integration of multimodal data. This system aims to leverage advancements in machine learning, image processing, and sensor technology to provide a more reliable and effective solution.

Relationship to Larger Systems:
While the Enhanced Driver Drowsiness Detection System is designed as a standalone product, it is also intended to be compatible with existing automotive safety systems. It can enhance the capabilities of systems such as Advanced Driver Assistance Systems (ADAS) by providing additional safety features and enriching the vehicle's safety ecosystem.

2.2 Product Functions

The project aims to develop a sophisticated driver drowsiness detection system using a hybrid approach that incorporates advanced image processing, machine learning, and computational intelligence algorithms. This system is designed to detect signs of driver fatigue through both visual indicators and physiological data, providing real-time alerts to prevent accidents caused by drowsy driving.

Major Functions:
The system is expected to perform the following major functions:

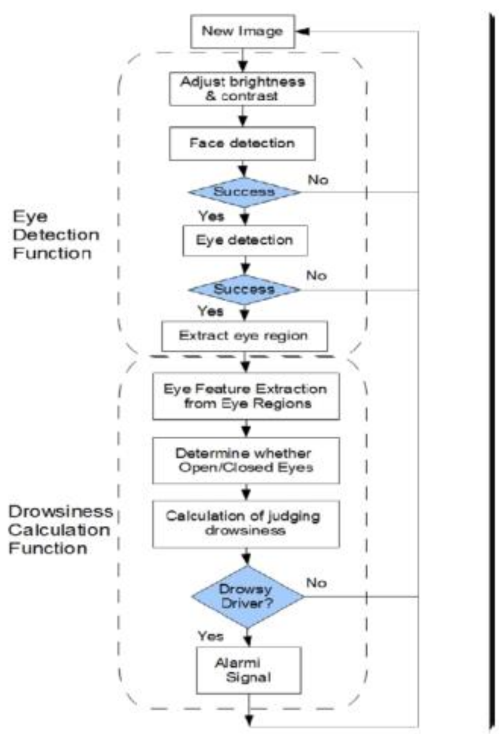
Real-time Drowsiness Detection: Utilize machine learning algorithms and image processing techniques to detect early signs of drowsiness such as eye closure, yawning, and head position changes.

Multi-Modal Data Integration: Incorporate data from dual cameras and heartbeat sensors to enhance detection accuracy, ensuring reliable performance under various lighting and driving conditions.

Alert Generation: Provide timely and effective alerts to the driver through both visual (e.g., flashing lights) and auditory (e.g., beeps) warnings based on the detected level of drowsiness.

Monitoring and Feedback: Offer continuous feedback on the driver's state, including time since last break and recommended rest periods, to encourage safer driving practices.

System Adaptability: Ensure the system is adaptable to different vehicles and driver preferences, including calibration options for sensitivity and alert types.



2.3 User Classes and Characteristics

User Class and it's Characteristics:

- **Drivers:**

  Characteristics:

  Primary users of the system.
  Operators of vehicles equipped with the drowsiness detection system.
  May vary in driving experience, age, and preferences.

  Needs:

  Require a reliable and unobtrusive system that effectively alerts them to signs of drowsiness.
  Value user-friendly interfaces for system interaction and customization.

- **Passengers:**

  Characteristics:

  Individuals traveling in vehicles equipped with the drowsiness detection system.
  Can include family members, friends, or colleagues.

  Needs:

  Concerned about their safety while traveling with a drowsy driver.
  May benefit from system alerts to intervene or provide assistance to the driver.

- **Automotive Manufacturers:**

  Characteristics:

  Companies involved in the design, production, and distribution of vehicles.
  Seek to enhance vehicle safety features and customer satisfaction.

  Needs:

  Interested in integrating drowsiness detection systems into their vehicles to offer advanced safety features.
  Require reliable and cost-effective solutions that can be easily integrated into existing vehicle systems.

- **Regulatory Authorities:**

  Characteristics:

  Government agencies responsible for setting and enforcing safety regulations in the automotive industry.
  Ensure compliance with standards and regulations related to driver monitoring systems.

Needs:

Require systems that meet specified safety and performance standards.
Need documentation and evidence of compliance with regulatory requirements.

Characteristics:

1. **Safety-Critical:**
    - The project addresses safety concerns related to driver fatigue and drowsiness, making it crucial for accident prevention and mitigation.
2. **Real-Time Processing:**
    - The system requires real-time processing capabilities to monitor and detect signs of drowsiness promptly.
3. **Machine Learning Integration:**
    - Incorporates machine learning algorithms for facial feature analysis, necessitating data processing and model training components.
4. **Hardware Integration:**
    - Involves integration with existing in-car camera hardware, requiring compatibility and interoperability considerations.
5. **User Interface Design:**
    - Requires intuitive user interfaces for system configuration, monitoring, and interaction, catering to users' varying levels of technical expertise.
6. **Compliance Requirements:**
    - Must adhere to relevant safety standards and regulations governing driver monitoring systems in automotive environments.
7. **Scalability:**
    - Should be designed with scalability in mind to accommodate future updates, improvements, and integration with additional features or systems.

2.4 Operating Environment

1. **Power Supply:**

   Ensure sufficient power supply for all hardware components, considering the power requirements of the in-car camera, Arduino board, sensors, and actuators.
   Use appropriate voltage regulators or power management circuits to regulate and distribute power effectively.

2. **Integration with Vehicle Systems:**

   Integrate the Driver Drowsiness Detection system seamlessly with existing vehicle systems, ensuring compatibility and minimal interference with other functionalities.

3. **Robustness and Reliability:**

   Design the system to withstand the harsh operating conditions of a vehicle environment, including temperature variations, vibrations, and electromagnetic interference.
   Implement error handling and recovery mechanisms to ensure system robustness and reliability.

4. **User Interaction and Feedback:**

   Design intuitive user interfaces for system configuration and monitoring, considering the limited display and input options available in a vehicle environment.
   Provide clear and timely feedback to the driver through visual, auditory, and tactile cues to effectively communicate alerts and warnings.

2.5 Design and Implementation Constraints

<u>Design Constraints:</u>

1. **Hardware Limitations:**

   The system must operate within the constraints of the hardware components available, such as processing power, memory, and sensor capabilities.
   Compatibility with existing in-car camera hardware and microcontroller platforms (e.g., Arduino) may impose limitations on system design.

2. **Power Consumption:**

   The system should minimize power consumption to ensure compatibility with the vehicle's electrical system and prolong battery life.
   Efficient power management techniques should be employed to optimize energy usage without compromising system performance.

3. **Space Limitations:**

   Physical space constraints within the vehicle may limit the size and placement of hardware components, such as sensors, actuators, and computing units.
   Compact and space-efficient designs are necessary to integrate the system seamlessly into the vehicle's interior without obstructing visibility or passenger comfort.

4. **Regulatory Compliance:**

   The system must comply with regulatory standards and requirements governing automotive safety and driver monitoring systems.
   Compliance with standards such as ISO 26262 (Functional Safety) and regional regulations (e.g., FMVSS in the United States) may impose constraints on system design and implementation.

<u>Implementation Constraints:</u>

1. **Software Dependencies:**

   The choice of programming languages, libraries, and frameworks may be constrained by compatibility with hardware platforms and existing software infrastructure.
   Limited support for certain programming languages or libraries on embedded systems (e.g., Arduino) may influence implementation decisions.

2. **Real-Time Processing Requirements:**

   The system must meet real-time processing requirements to detect and respond to drowsiness events promptly.
   Implementation constraints related to algorithm efficiency, computational complexity, and latency must be addressed to ensure timely detection and alerting.

3. **Data Privacy and Security:**

   The system should adhere to data privacy regulations and ensure the secure handling of sensitive information, such as video footage of drivers.
   Implementation constraints related to data encryption, access control, and secure communication protocols may impact system design and development.

4. **Testing and Validation:**

   Comprehensive testing and validation procedures are necessary to ensure the accuracy, reliability, and safety of the system.
   Implementation constraints related to resource availability, testing environments, and access to real-world driving data may influence the scope and depth of testing efforts.

5. **Cost Constraints:**

   Budgetary constraints may limit the resources available for system development, including hardware components, software licenses, and personnel expenses.
   Implementation decisions should balance cost considerations with the need to achieve desired performance, functionality, and reliability.

2.6 Assumptions and Dependencies

<u>Assumptions:</u>

1. **In-car Camera Availability:**

   It is assumed that vehicles equipped with in-car cameras are readily available or can be easily retrofitted with compatible cameras.

2. **Driver Cooperation:**

   It is assumed that drivers will cooperate with the system and adhere to any alerts or warnings issued by the drowsiness detection system.

3. **Adequate Lighting Conditions:**

   It is assumed that adequate lighting conditions are present within the vehicle cabin to facilitate accurate facial feature detection and analysis.

4. **System Accuracy:**

   It is assumed that the drowsiness detection system achieves a reasonable level of accuracy in identifying signs of driver drowsiness.

5. **Compliance with Regulations:**

   It is assumed that the system will comply with relevant safety regulations and standards governing driver monitoring systems in automotive environments.

Dependencies:

1. **Availability of Hardware Components:**

   The project depends on the availability of hardware components such as in-car cameras, microcontrollers (e.g., Arduino), sensors, and actuators.

2. **Software Libraries and Frameworks:**

   The project depends on the availability and compatibility of software libraries and frameworks for image processing, machine learning, and communication protocols.

3. **Access to Testing Environments:**

   The project depends on access to testing environments, including vehicles equipped with in-car cameras and simulated driving scenarios.

4. **Regulatory Approval:**

   The project depends on obtaining regulatory approval for the drowsiness detection system from relevant authorities.

5. **User Acceptance:**

   The project depends on user acceptance and adoption of the drowsiness detection system by drivers and automotive manufacturers.

3. External Interface Requirements

3.1 User Interfaces

Overview:
The user interface (UI) of the Enhanced Driver Drowsiness Detection System is designed to provide an intuitive, user-friendly experience that allows drivers to easily interact with the system while minimizing distraction. The interface will adhere to automotive industry standards for safety and usability.

Components Requiring User Interface:

Alert Notifications Panel: Displays real-time alerts and warnings about the driver's state.
System Status Screen: Shows the operational status of the system, including camera and sensor activity.
Settings Menu: Allows drivers to customize settings such as sensitivity levels and types of alerts.
User Interface Characteristics:

Screen Layout Constraints:
The main information must be visible within the top two-thirds of the display to ensure it is within the driver's natural line of sight.
Critical alerts must be prominently displayed with larger fonts and distinctive colors to catch immediate attention.
Standard Buttons and Functions:
Home Button: Returns to the main dashboard from any screen.
Help Button: Accessible from all screens, providing quick tips and FAQs on system use.
Settings Button: Allows customization of system settings.
Each screen will feature a consistent header and footer layout for familiarity.
GUI Standards:
Follow the "Automotive User Interface Design Principles" focusing on minimalism, high contrast for readability, and non-distracting color schemes.
Use of large, easy-to-press buttons to accommodate a moving vehicle environment.
Keyboard Shortcuts:
Shortcuts for enabling/disabling the system (e.g., Ctrl + E/D).
Quick access to settings (e.g., Ctrl + S).
Emergency mute or pause alerts (e.g., Ctrl + M).
Error Message Display Standards:
Error messages must be concise, avoiding technical jargon, and provide clear instructions on how to resolve issues.
Use red or orange colors for error messages to ensure they are easily distinguishable from other alerts.
Alert Notification Panel: A visual and auditory warning icon appears, with text such as "Drowsiness Detected! Please consider taking a break."
System Status Screen: Displays a green checkmark or a red cross to indicate the operational status of cameras and sensors.
Settings Menu: Options displayed in a list format, each with a toggle or slider for adjustment.

3.2 Hardware Interfaces

1. In-Car Camera:

An in-car camera is used to capture real-time video feed of the driver's face.
It may be a standalone camera or integrated into existing vehicle systems.

2. Microcontroller (Arduino):

Arduino serves as the hardware platform for interfacing with sensors and actuators.
It processes data from sensors, controls actuators, and communicates with the main computing unit (e.g., Raspberry Pi) for drowsiness detection.

3. Sensors:

Various sensors may be used to collect data relevant to driver drowsiness detection, such as:

i) Accelerometer or gyroscope to detect vehicle motion and vibrations.
ii) Heart rate sensor to monitor physiological indicators of drowsiness.
iii) Temperature sensor to detect changes in cabin temperature, which may indicate discomfort or fatigue.

4. Actuators:

Actuators are used to provide feedback to the driver when drowsiness is detected, such as:

i) LED indicators to provide visual alerts.
ii) Buzzer or speaker to emit audible alarms.
iii) Haptic feedback devices (e.g., vibrating motors) to alert the driver through tactile sensations.

3.3 Software Interfaces

1. Python Scripts:

Python is used for high-level programming tasks, including:

i) Image processing and facial recognition algorithms for drowsiness detection.
ii) Integration with machine learning libraries (e.g., OpenCV, TensorFlow) for model training and inference.
iii) Communication with the Arduino microcontroller to send/receive data and control actuators.

2. Arduino Sketches:

Arduino sketches (code) are written to interface with sensors and actuators:

i) Read sensor data (e.g., accelerometer readings) to assess driver behavior and vehicle dynamics.
ii) Control actuators (e.g., LEDs, buzzer) to provide alerts to the driver based on input from Python scripts.

3. **Serial Communication:**

a) Serial communication protocols (e.g., UART, SPI) are used for bidirectional data exchange between Python scripts running on the main computing unit and Arduino microcontroller.
b) Python serial libraries (e.g., pySerial) are utilized to establish communication channels and send/receive data packets.

3.4 Communications Interfaces

1. **Inter-Component Communication:**

Requirement: Enable communication between different components of the system, such as the main computing unit, Arduino microcontroller, sensors, and actuators.
Python scripts running on the main computing unit communicate with Arduino sketches via serial communication to exchange data and control signals.

Protocol: Serial communication protocols like UART, SPI can be used for direct communication between the main computing unit and the Arduino microcontroller.

2. **Data Transmission:**

Requirement: Transmit data from sensors to the main computing unit for processing and analysis. Video data from the in-car camera is transmitted to the main computing unit via USB interface for real-time processing and facial feature analysis.

Protocol: Standard communication protocols such as USB, I2C, or GPIO can be used for connecting sensors to the main computing unit.

3. **Alerting Mechanism:**

Requirement: Provide alerts and notifications to the driver when signs of drowsiness are detected. Visual alerts can be implemented using LED indicators, auditory alerts using a buzzer or speaker, and tactile alerts using haptic feedback devices.

Protocol: Visual, auditory, and tactile communication methods can be used for alerting the driver.
Example: (e.g., vibrating motors).

4. **User Interface:**

Requirement: Implement a user interface for system configuration, monitoring, and interaction. A GUI application developed using Python's Tkinter or PyQt libraries provides a user-friendly interface for configuring system settings and viewing drowsiness detection status.

Protocol: Graphical user interfaces (GUIs) or command-line interfaces (CLIs) can be used for user interaction.

5. <u>Remote Monitoring and Control:</u>

Requirement: Allow remote monitoring and control of the system, potentially enabling fleet management or centralized monitoring solutions.
A web-based dashboard accessible via HTTP allows administrators to monitor the status of multiple vehicles equipped with the drowsiness detection system and receive alerts in real-time.

Protocol: Network communication protocols such as HTTP, WebSocket, or MQTT can be used for remote communication.

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 Real-time Drowsiness Detection

4.1.1 Description and Priority

This feature uses image processing and machine learning algorithms to detect signs of drowsiness in real-time. The priority is High, with a benefit rating of 9 due to its direct impact on driver safety, a penalty rating of 9 if absent, a cost rating of 6 reflecting the technological investment required, and a risk rating of 3 considering the mature nature of the underlying technologies.

4.1.2 Stimulus/Response Sequences

Stimulus: The system continuously monitors the driver through cameras and sensors.

Response: Upon detecting signs of drowsiness, the system alerts the driver visually and audibly.

4.1.3 Functional Requirements

REQ-1: The system must analyze facial cues such as eye closure and yawning.

REQ-2: Upon detection of drowsiness, the system must issue auditory and visual alerts within 2 seconds.

REQ-3: The system should log all drowsiness detection events for further analysis.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system must process images and generate alerts in real-time, with a maximum latency of 2 seconds under normal operating conditions.

5.2 Safety Requirements

The system must include fail-safe measures that ensure it does not distract the driver or impede vehicle operation.
Must comply with automotive safety regulations such as ISO 26262 for functional safety.

5.3 Security Requirements

The system must securely store and transmit data, ensuring that driver privacy is maintained.
Must comply with GDPR and other relevant privacy regulations concerning data handling and user consent.

5.4 Software Quality Attributes

Reliability: The system must perform consistently under various conditions without failure.
Usability: The interface must be intuitive and easily navigable even under driving conditions.
Maintainability: Code and system architecture must be documented to allow easy updates and maintenance.

5.5 Business Rules

Only the driver or authorized personnel may adjust critical system settings.
The system must not be disabled or overridden during active driving without explicit confirmation from the driver.

Appendix A: Glossary

Drowsiness Detection: The process of identifying signs of fatigue or sleepiness in a person.

GDPR: General Data Protection Regulation, a legal framework that sets guidelines for the collection and processing of personal information.