AI-POWERED REAL TIME DRIVER MONITORING & ACCIDENT PREVENTION SYSTEM

CodeCrafters Team

D. S. Sandeepa	**221444563**
R. U. Gonalagoda	**621429318**
P. S. Ekanayaka	**621429276**
K. J. L. Perera	**621429815**
T. P. Denagamage	**321429305**

29th May 2025

TABLE OF CONTENTS

1. Executive Summary

Road traffic accidents remain a leading cause of death and injury worldwide, with human error identified as the primary factor of most crashes. In Sri Lanka, public buses account for a significant proportion of road incidents, largely due to driver fatigue, distraction, and improper behavior. Despite this, most buses lack automated systems that can continuously monitor driver alertness or prevent accidents proactively.

SafeDriver addresses this gap by harnessing state-of-the-art artificial intelligence (AI) techniques combined with embedded hardware to monitor driver behavior in real-time. Using camera-based eye tracking and deep learning models, it detects driver drowsiness, mobile phone usage, and distraction. The system then issues immediate multimodal alerts to the driver (audio, buzzer, voice) and escalates warnings to transport authorities if unsafe behavior persists.

Additionally, SafeDriver incorporates GPS tracking and hazard mapping to correlate environmental risks with driver behavior, offering a comprehensive safety solution. Passengers can access a mobile app that delivers live updates on bus safety status, increasing transparency and accountability.

This intelligent intervention is expected to provide scalable technology tailored to the public transport infrastructure in Sri Lanka and other similar developing countries.

2. Introduction

2.1 Background

Road safety is a complex challenge globally, exacerbated in developing countries where infrastructure, enforcement, and awareness may be inadequate. In Sri Lanka, over 2000 road fatalities are reported annually, with bus accidents representing a large fraction [1]. According to the National Transport Commission (NTC), the leading causes include driver fatigue, distractions due to mobile phone use, and inattentiveness to driving conditions [2].

Traditional safety measures in Sri Lanka rely heavily on reactive processes: accident reports, manual inspections, or driver self-reporting. These approaches fail to provide real-time interventions, which have been proven critical in reducing accident risk [3].

Advancements in computer vision, machine learning, and embedded systems offer new possibilities. Systems that use cameras and AI to monitor drivers have demonstrated effectiveness in detecting early warning signs such as prolonged eye closure (PERCLOS), yawning, and head pose changes indicative of drowsiness [4], [5]. Deep learning models have shown promise in recognizing distracted behaviors, including mobile phone use, even under constrained hardware environments like the Raspberry Pi [6], [7].

Moreover, integrating these driver monitoring systems with GPS data and cloud infrastructure allows for hazard mapping and timely alerts to both drivers and authorities, improving accident prevention and emergency response [8].

2.2 Problem Statement

The specific problem statement is “Sri Lanka’s public transport system currently lacks a proactive and continuous monitoring mechanism for the safety of the driver and passengers while on the move”.

Key challenges include:

- Absence of real-time detection of driver fatigue and distraction, leading to unaddressed risky behavior during operation.
- Delays in accident prevention such as warnings or interventions come post-incident.
- Limited communication pathways between drivers, authorities, and passengers regarding safety status.
- Lack of scalable, cost-effective AI-enabled monitoring solutions tailored to local conditions.

2.3 Aim

The SafeDriver system aims to revolutionize public transport safety in Sri Lanka by providing a comprehensive, AI-driven monitoring solution that bridges the gap between reactive accident reporting and proactive safety intervention. This system will serve as a technological backbone for modernizing Sri Lanka's public transport infrastructure while ensuring scalability and cost-effectiveness.

3. Key System Features

3.1 Real-Time Driver Alertness Monitoring

Continuous surveillance of driver behavior through advanced eye-tracking algorithms, monitoring eye aspect ratio (EAR), blink frequency, and head pose to detect early signs of fatigue and drowsiness before they lead to accidents [4], [9].

3.2 Intelligent Distraction Detection

AI-powered recognition system that identifies mobile phone usage and other distracting behaviors using lightweight convolutional neural networks optimized for embedded platforms, ensuring immediate intervention [6], [7].

3.3 Multi-Modal Alert System

Graduated response mechanism featuring immediate audio buzzers and voice warnings for drivers, with automatic escalation to transport authorities through SMS and cloud notifications when unsafe behavior persists [10].

3.4 GPS-Integrated Hazard Mapping

Location-aware safety system that correlates real-time vehicle position with accident-prone zones, providing context-sensitive alerts when entering high-risk areas while driver attention is compromised [8].

3.5 Passenger Transparency Platform

Dedicated mobile applications provide passengers with real-time safety status updates, fostering accountability and enabling community participation in transport safety monitoring [11].

3.6 Automated Compliance Reporting

An intelligent reporting system generating daily and incident-based reports for transport authorities, enabling data-driven policy decisions and targeted safety interventions [12].

4. Project Objectives

1. **Develop robust computer vision algorithms** for drowsiness detection through eye aspect ratio, blink frequency, and head pose estimation.
2. **Create and deploy mobile phone and distraction detection models** based on convolutional neural networks optimized via TensorFlow Lite for efficient embedded execution.
3. **Design a real-time alerting system** combining audio buzzers and voice warnings to the driver, plus SMS or cloud notifications for transport authorities.
4. **Integrate GPS-based hazard detection**, mapping accident-prone zones and alerting drivers when entering such areas.
5. **Develop a passenger-facing mobile application** using Flutter and Firebase, providing transparency and live safety updates.
6. **Conduct iterative testing and calibration** under realistic conditions to minimize false positives/negatives and improve system robustness.
7. **Generate automated daily and event-triggered reports** for authorities to track driver safety compliance and intervene when necessary.

5. Methodology

5.1 Driver Monitoring Subsystem

- Utilize the Raspberry Pi 4 as the core embedded platform, coupled with a Raspberry Pi Camera Module 2 NoIR positioned to capture the driver's face unobtrusively.
- Employ OpenCV and MediaPipe for efficient facial landmark detection, tracking eyes and estimating head poses in real time [16], [17].
- Compute metrics such as Eye Aspect Ratio (EAR) and blink rate to detect drowsiness, supported by state-of-the-art algorithms [4], [9].
- Implement TensorFlow Lite-based convolutional neural networks to detect mobile phone usage with high accuracy while maintaining low latency on embedded hardware [6], [13].

5.2 Alerting and Communication

- Develop an alert system that triggers immediate buzzer sounds and synthesized voice warnings when fatigue or distraction thresholds are exceeded.
- Configure escalation logic to send SMS or push notifications to the National Transport Commission (NTC) if alerts are ignored or repeated [10].
- Use Firebase Cloud Messaging to facilitate real-time communication between the device, authorities, and passenger apps [14].

5.3 Hazard Detection and GPS Integration

- Integrate a GPS module with the Raspberry Pi to collect real-time location and speed data.
- Maintain a cloud-based database of hazardous locations (accident hotspots, sharp curves, etc.) to correlate with driver behavior and preemptively alert drivers [8].
- Enable logging and analytics to assist authorities in understanding risk patterns [12].

5.4 Passenger Mobile Application

- Develop a cross-platform app with Flutter, connected to Firebase Real-time Database to receive and display live alerts and bus status [11], [14].
- Features include real-time location tracking, safety status notifications, and historical driver behavior reports to increase passenger confidence and participation in safety monitoring.

5.5 Testing and Calibration

- Conduct extensive field tests under varying lighting and traffic conditions.
- Gather feedback from drivers after completing their rides, as well as from relevant authorities to fine-tune sensitivity thresholds, minimizing false alarms while maintaining safety. [15].
- Use Agile methodologies to iteratively improve model performance and system usability.

6. Technical Approach

The technical approach for the SafeDriver system focuses on the integration of computer vision, embedded hardware, real-time alerting, and cloud connectivity to create a robust, scalable, and cost-effective driver monitoring and accident prevention platform. The system architecture is modular, enabling flexibility in development and ease of maintenance.

6.1 System Architecture Overview

The SafeDriver system employs a distributed architecture combining edge computing for real-time processing with cloud integration for data analytics and communication. This hybrid approach ensures minimal latency for critical safety functions while enabling comprehensive monitoring and reporting capabilities.

Core Design Principles:

- **Edge-First Processing**: Critical safety decisions made locally to eliminate network dependency
- **Modular Architecture**: Independent subsystems allowing flexible deployment and maintenance
- **Scalable Integration**: Cloud backend supporting fleet-wide deployment and management
- **Fault-Tolerant Design**: Multiple redundancy layers ensuring system reliability

6.2 Advanced Driver Monitoring Implementation

Computer Vision Pipeline:

- **Facial Landmark Detection**: MediaPipe framework for robust face tracking under varying lighting conditions [17]
- **Eye State Analysis**: Advanced PERCLOS (Percentage of Eyelid Closure) calculation using temporal filtering to reduce noise [4], [9]

- **Head Pose Estimation**: 3D head orientation tracking to detect attention deviation and nodding patterns [5]
- **Blink Pattern Analysis**: Frequency and duration analysis to distinguish between normal blinking and microsleep episodes [18]

Deep Learning Integration:

- **Mobile Phone Detection**: Custom CNN architecture trained on diverse datasets, optimized through TensorFlow Lite quantization [6], [13]
- **Behavioral Classification**: Multi-class detection system identifying phone calls, texting, and other distracting activities [7]
- **Adaptive Learning**: System calibration based on individual driver baseline behaviors to reduce false positives [15]

6.3 Intelligent Alert Escalation System

Multi-Tiered Response Protocol:

- **Level 1 - Immediate Intervention:** Audio buzzer and voice alerts (0-5 seconds)
- **Level 2 - Persistent Warning:** Increased alert intensity with visual indicators (5-15 seconds)
- **Level 3 - Authority Notification:** Automated SMS/push notifications to NTC (15+ seconds) [10]
- **Level 4 - Emergency Response:** GPS location sharing with emergency services for severe cases

Smart Threshold Management:

- Dynamic sensitivity adjustment based on time of day, route difficulty, and weather conditions
- Driver-specific calibration to account for individual behavioral patterns [15]

- Environmental context awareness (traffic density, road conditions) for alert prioritization.

6.4 Advanced GPS and Environmental Integration

Hazard Intelligence System:

- **Dynamic Risk Mapping**: Real-time integration of traffic data, weather conditions, and historical accident data [8]
- **Predictive Analytics**: Machine learning models predicting high-risk scenarios based on multiple environmental factors
- **Geofencing Technology**: Automatic alert activation when entering predefined dangerous zones

6.5 Cloud Infrastructure and Mobile Application

Backend Architecture:

- **Firebase Realtime Database**: Instant data synchronization across all system components [14]
- **Cloud Functions**: Serverless processing for data analytics and report generation
- **Machine Learning Pipeline**: Continuous model improvement using aggregated anonymous data
- **Security Framework**: End-to-end encryption and secure authentication protocols

Mobile Application Features:

- **Real-Time Safety Dashboard**: Live visualization of driver alertness status [11]
- **Historical Analytics**: Trend analysis of safety metrics over time
- **Community Reporting**: Passenger feedback system for safety concerns
- **Emergency Contacts**: Quick access to emergency services and transport authorities

6.6 Implementation Tools and Technologies

Layer/Module	Technology/Tool	Purpose
Hardware	Raspberry Pi 4, Raspberry Pi Camera Module 2 NoIR, GPS Module, Buzzer	Embedded processing and sensing hardware
Computer Vision	OpenCV, MediaPipe	Face/eye tracking and head pose estimation
Deep Learning	TensorFlow Lite	Mobile phone usage detection on embedded platform
Backend & Cloud	Firebase Realtime DB, Firebase Cloud Messaging	Real-time data storage and communication
Mobile Application	Flutter	Cross-platform passenger app development
Programming Languages	Python	Driver monitoring and system integration scripts

6.7 Data Analytics and Reporting Framework

Intelligent Analytics Engine:

- **Pattern Recognition**: Identification of recurring safety issues and temporal patterns [12]
- **Predictive Modeling**: Forecasting potential safety risks based on historical data
- **Performance Metrics**: Comprehensive KPIs for driver safety and system effectiveness
- **Compliance Monitoring**: Automated tracking of safety regulation adherence

Reporting Capabilities:

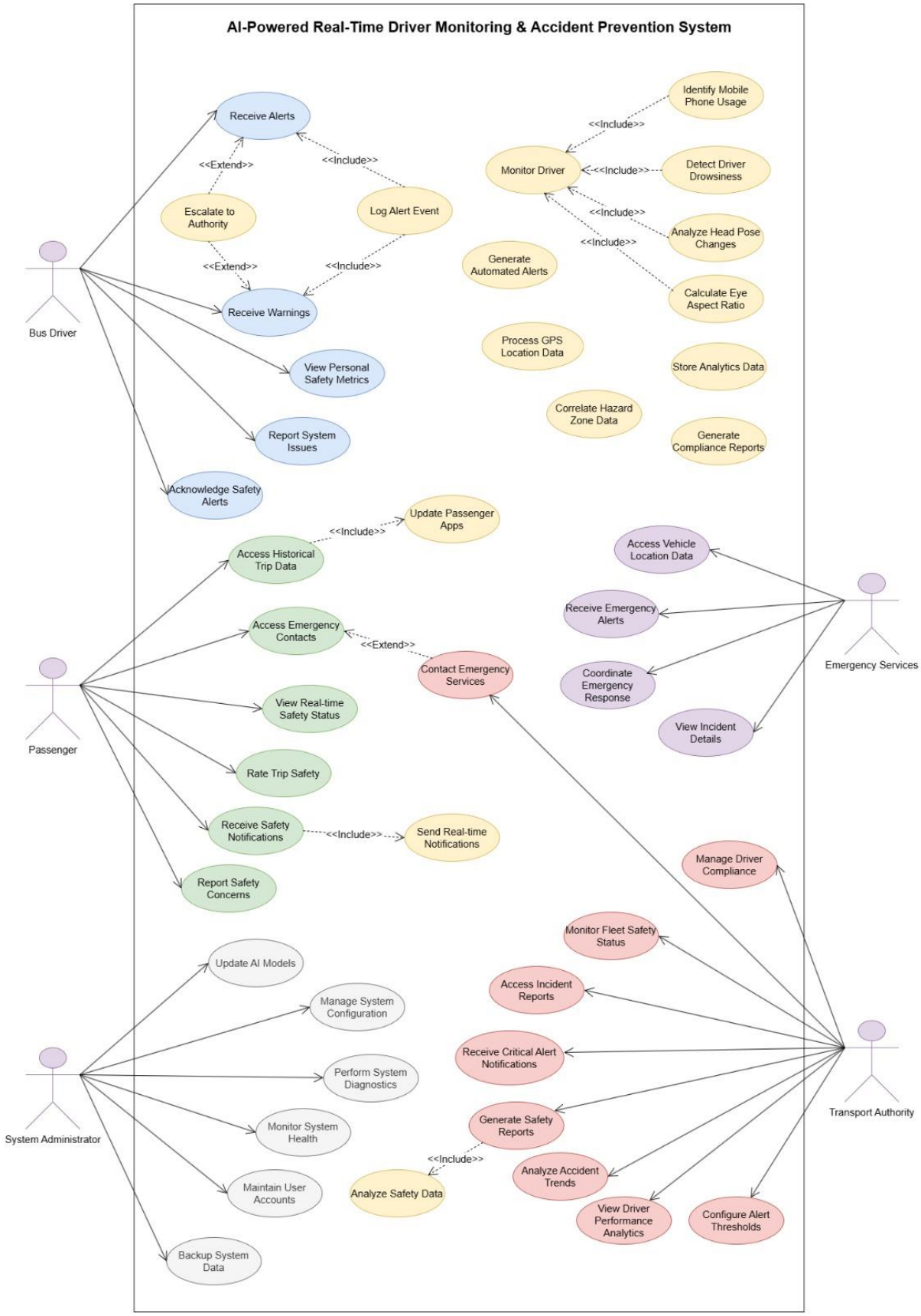
- **Real-Time Dashboards**: Live monitoring interfaces for transport authorities
- **Automated Reports**: Daily, weekly, and monthly safety summaries
- **Incident Analysis**: Detailed forensic reports for accident investigation
- **Fleet Management**: Comparative analysis across different routes and drivers

6.8 Summary of Workflow

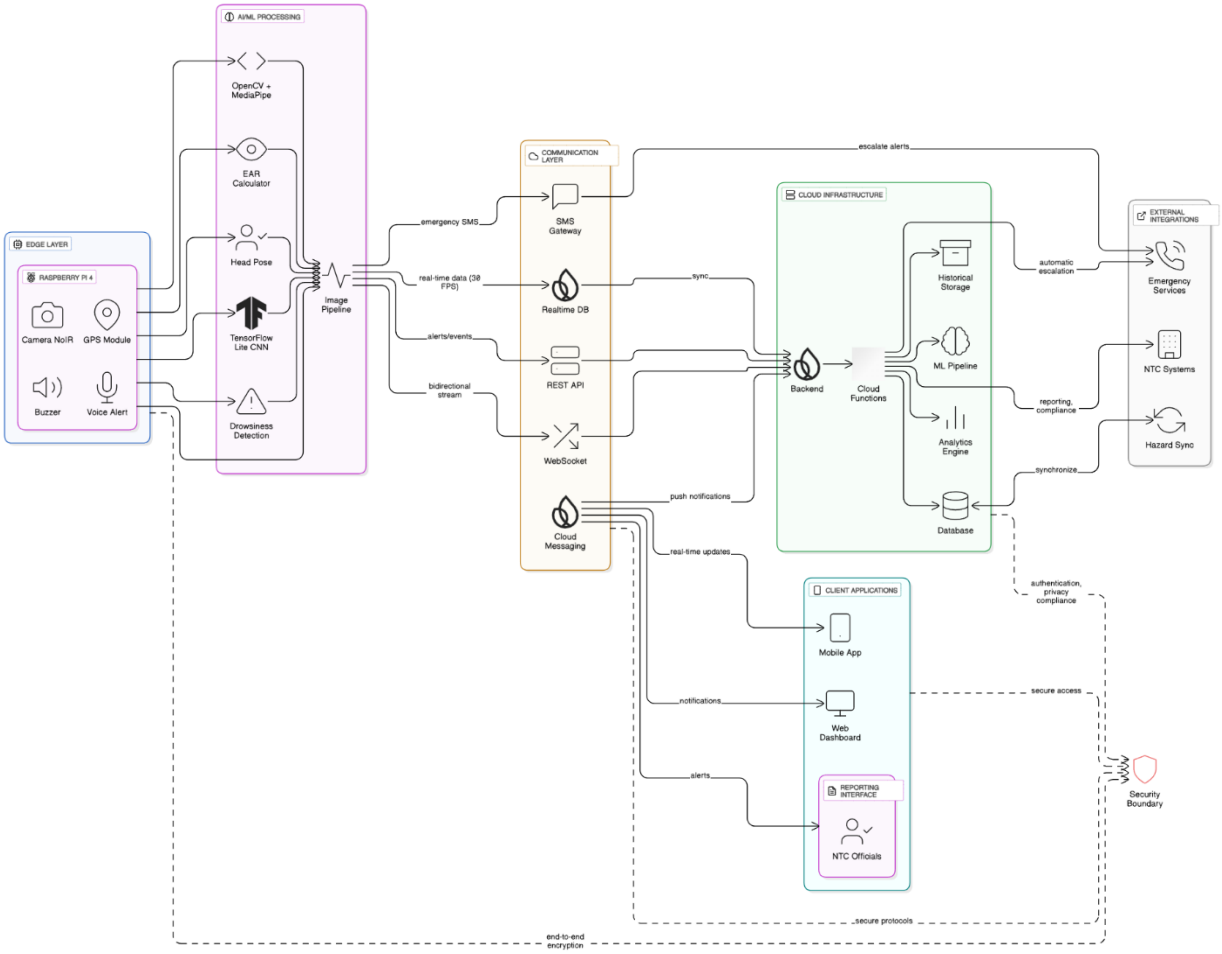
1. The Raspberry Pi Camera Module 2 NoIR continuously captures video frames of the driver.
2. Frames are processed locally using OpenCV and MediaPipe to detect facial landmarks and calculate fatigue metrics [16], [17].
3. The mobile phone detection CNN analyzes images for distraction signs [6], [7].
4. On detecting unsafe behavior, the system immediately triggers audible alerts.
5. GPS data is continuously monitored and combined with driver status for hazard evaluation [8].
6. If warnings are ignored or behavior persists, alerts escalate to transport authorities via cloud notifications [10].
7. Passengers receive live safety updates through the mobile app [11].

6.9 Diagrams

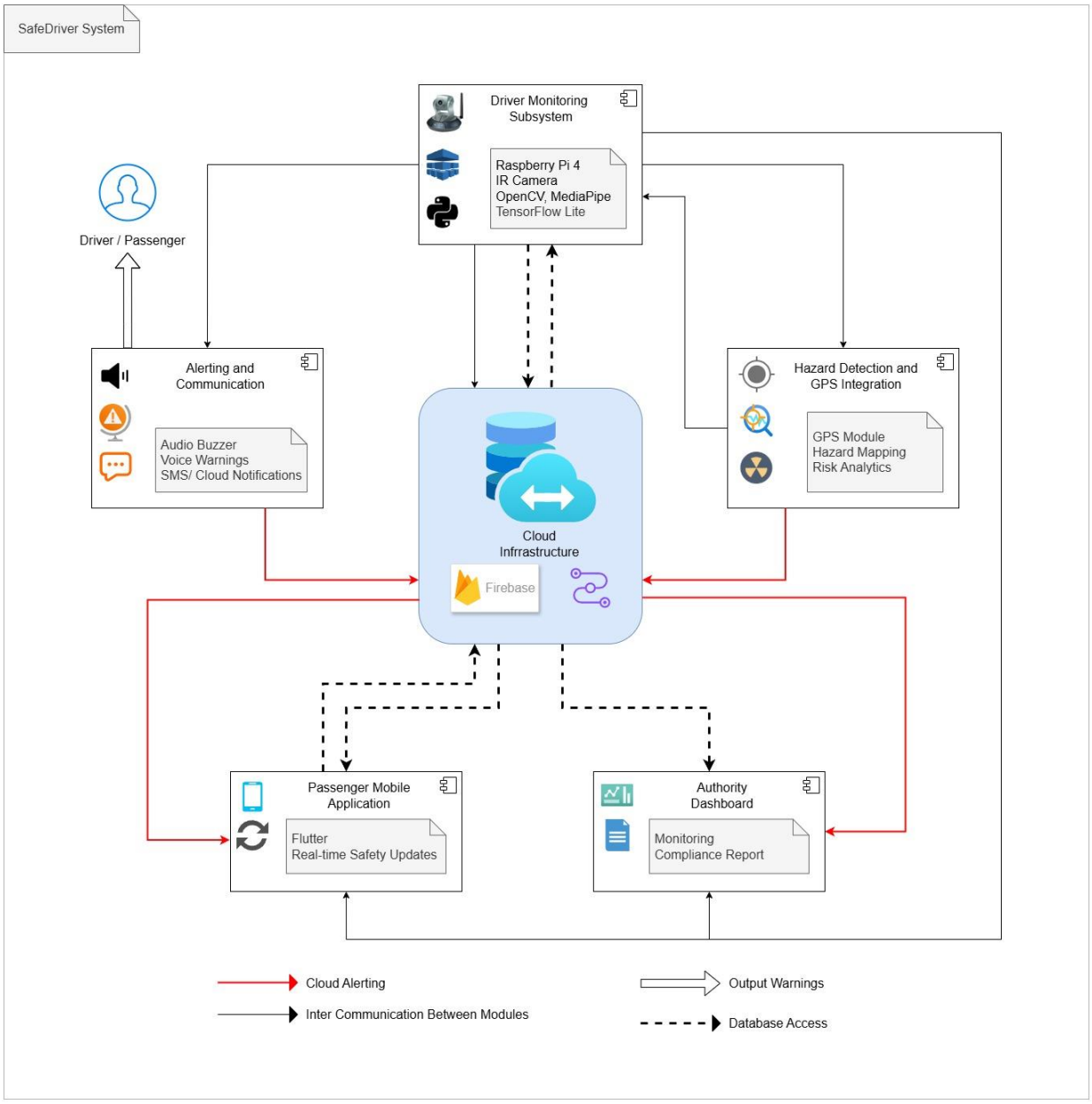
6.9.1 Use Case Diagram



6.9.2 System Architecture Diagram



6.9.3 System Block Diagram



7. Project Plan

Phase	Activities	Duration	Dates
Initial Meeting & Discussion	Discussion with Head & Staff	Half day	10/05/2025
Requirements Gathering	Stakeholder meetings, specification definition	2 weeks	11/05/2025 – 25/05/2025
System Design & Architecture	Architectural diagrams, component selection	2 weeks	26/05/2025 – 08/06/2025
Final Proposal Submission	Preparation and submission of final proposal	1 day	08/06/2025
AI Model Development	Data collection, training, optimization	6 weeks	09/06/2025 – 20/07/2025
Progress Report 1 Submission	Submit progress report 1	1 day	20/07/2025
Hardware Integration	Component assembly, sensor integration	8 weeks	21/07/2025 – 28/09/2025
Progress Report 2 Submission	Submit progress report 2	1 day	28/09/2025
Testing & Calibration	Real-world testing, threshold tuning	10 weeks	29/09/2025 – 12/12/2025
Progress Report 3 Submission	Submit progress report 3	1 day	12/12/2025
Final Draft Report Preparation	Compilation and review of final draft	3 months	13/12/2025 – 31/03/2026

Final Draft Report Submission	Submit final draft report	1 day	31/03/2026
Final Report Preparation	Final report editing and preparation	3 weeks	01/04/2026 – 23/04/2026
Final Report Submission	Submit final report	1 day	24/04/2026
Reflective Log Submission	Submit reflective log	1 day	24/04/2026

8. Project Management

8.1 Team Structure and Responsibilities

D. S. Sandeepa

- **Primary Responsibilities:**
    - Overall project coordination and timeline management
    - System architecture design and integration oversight
    - Computer vision algorithm development (drowsiness detection)
    - Deep learning model training and optimization
    - TensorFlow Lite model conversion and deployment
    - API development for mobile and embedded communication
- **Secondary Tasks:**
    - Integration testing with embedded systems
    - Hardware troubleshooting and maintenance
    - Performance optimization for embedded deployment

R. U. Gonalagoda

- **Primary Responsibilities:**
    - Raspberry Pi hardware setup and configuration
    - Camera module integration and optimization
    - GPIO programming for buzzers and sensors
    - Power management and system reliability
    - Cloud messaging and notification systems
- **Secondary Tasks:**
    - Database optimization and scaling
    - Mobile app deployment and distribution

P. S. Ekanayaka

- **Primary Responsibilities:**
    - Stakeholder communication and requirement gathering

        - User interface design and user experience optimization
        - Real-time data visualization and dashboards
- **Secondary Tasks:**
    - Backend API development assistance
    - Testing protocol design
    - Data collection and dataset preparation
    - Field testing coordination
    - User acceptance testing coordination

K. J. L. Perera

- **Primary Responsibilities:**
    - Research and implementation of the latest AI techniques
    - Firebase backend architecture and setup
    - Real-time database design and implementation
    - Flutter mobile application development
    - Passenger-facing features implementation
- **Secondary Tasks:**
    - Security implementation and authentication
    - Cloud infrastructure monitoring

T. P. Denagamage

- **Primary Responsibilities:**
    - Risk assessment and mitigation strategies
    - Technical documentation and report preparation.
    - Performance benchmarking and accuracy validation
    - Data analytics and reporting systems
    - App testing and debugging
- **Secondary Tasks:**
    - Budget management and resource allocation
    - Model documentation and training guides

- Frontend documentation and user guides

8.2 Collaboration Framework

Weekly Sprint Planning (Every Saturday)

- Sprint goal definition and task allocation
- Progress review and impediment identification
- Resource requirement assessment
- Timeline adjustment if necessary

Daily Stand-ups (Monday, Wednesday, Friday)

- Individual progress updates (15 minutes maximum)
- Blocker identification and resolution planning
- Cross-team collaboration requirements
- Next-day priority setting

Bi-weekly Integration Meetings

- Module integration testing and validation
- System-wide compatibility checks
- Performance benchmarking and optimization
- Documentation updates and reviews

8.3 Quality Assurance and Testing Strategy

Individual Module Testing

- Each team member is responsible for unit testing their components
- Code review process with at least one other team member
- Performance benchmarking against defined metrics
- Documentation of test cases and results

Integration Testing (Shared Responsibility)

- **Week 1-2**: AI models integration with embedded systems (R.U. Gonalagoda + P.S.Ekanayaka)
- **Week 3-4**: Backend integration with mobile app (K.J.L. Perera + T.P. Denagamage)
- **Week 5-6**: End-to-end system testing (All members)
- **Week 7-8**: Field testing and calibration (All members)

User Acceptance Testing

- **Passenger App Testing**: T.P. Denagamage (Lead), D.S. Sandeepa (Support)
- **Driver Interface Testing**: P.S. Ekanayaka (Lead), R.U. Gonalagoda (Support)
- **Authority Dashboard Testing**: K.J.L. Perera (Lead), D.S. Sandeepa (Support)

8.4 Risk Management and Mitigation

Technical Risks

- **Hardware Failures**: Backup components procured, P.S. Ekanayaka maintains inventory
- **Model Performance Issues**: R.U. Gonalagoda to maintain alternative algorithm implementations
- **Connectivity Problems**: K.J.L. Perera to implement offline fallback mechanisms
- **Integration Challenges**: Weekly integration sprints to identify issues early

Project Management Risks

- **Timeline Delays**: Buffer time included in each phase, weekly progress monitoring
- **Resource Constraints**: Cross-training among team members for critical skills
- **Commuation Issues**: Structured meeting schedule and documentation requirements
- **Scope Creep**: D.S. Sandeepa maintains requirements baseline and change control

8.5 Communication and Documentation Standards

Internal Communication

- Primary: Microsoft Teams for daily communication
- Secondary: WhatsApp group for urgent notifications
- Code Repository: GitHub with branch protection and review requirements
- Documentation: Shared Google Workspace with version control

External Communication

- **Supervisor Meetings**: Dr. D.D.M. Ranasinghe (Supervisor)
- **Stakeholder Updates**: D.S. Sandeepa (Lead), relevant team member based on topic
- **Progress Reports**: Collaborative writing with D.S. Sandeepa as final reviewer
- **Technical Documentation**: Each member documents their respective modules

9. Budget Breakdown

Item	Description	Cost (LKR)
Software Development	AI, app, backend services	Rs. 100000
Hardware	Raspberry Pi 4, sensors, peripherals	Rs. 75000
Cloud Services	Firebase, SMS gateway	Rs. 25000
Testing & Miscellaneous	Field trials, materials	Rs. 10000
Total		Rs. 210000

10. Conclusion

SafeDriver offers a transformational leap in public transport safety by integrating AI-powered real-time monitoring with GPS hazard tracking and passenger engagement. By proactively detecting driver fatigue and distractions and involving transport authorities and passengers, the system aims to reduce accidents, save lives, and foster a culture of safety and transparency in Sri Lanka's public transport.

Its modular, scalable design and use of affordable embedded hardware ensure that SafeDriver can be widely deployed across fleets, positioning Sri Lanka as a leader in intelligent transportation systems in the developing world.

11. References

[1] Ministry of Transport and Civil Aviation Sri Lanka, "National Road Safety Strategic Plan 2023-2027," Colombo, 2023.

[2] National Transport Commission of Sri Lanka, "Annual Road Safety Statistics Report," Colombo: NTC Publications, 2024.

[3] World Health Organization, "Global Status Report on Road Safety," Geneva: WHO Press, 2023.

[4] T. Srivastava et al., "A Real-time Light-weight Computer Vision Application for Driver's Drowsiness Detection," *International Journal of Engineering and Manufacturing*, vol. 14, no. 2, pp. 12-23, 2024.

[5] A. M. Rahman et al., "Real-Time Machine Learning-Based Driver Drowsiness Detection Using Visual Features," *Journal of Imaging*, vol. 9, no. 5, article 91, 2023.

[6] C. Jin, Z. Zhu, Y. Bai, G. Jiang, and A. He, "A deep-learning-based scheme for detecting driver cell-phone use," *IEEE Access*, vol. 8, pp. 18580-18589, 2020.

[7] N. K. Vaegae, K. K. Pulluri, K. Bagadi, and O. O. Oyerinde, "Design of an efficient distracted driver detection system: Deep learning approaches," *IEEE Access*, vol. 10, pp. 116087-116097, 2022.

[8] M. F. B. Amin, I. J. Khan, F. Akter, A. Ahmed, and M. M. Islam, "SafeTrax: Smart collision prediction and alert system using IoT for sustainable traffic safety," *IEEE Access*, vol. 13, pp. 6667-6684, 2025.

[9] S. K. Ahmed et al., "Drowsiness detection in real-time via convolutional neural networks and transfer learning," *Journal of Engineering and Applied Science*, vol. 71, article 457, 2024.

[10] A. Oguntimilehin et al., "Internet of Things (IoT) enabled automobile accident detection and reporting system," in *Proc. 2022 5th Information Technology for Education and Development (ITED)*, 2022, pp. 15-19.

[11] R. K. Peddarapu, L. Basuthkar, M. Desireddy, and S. L. Kompella, "Raspberry Pi-based driver drowsiness detection," in *Proc. 2024 IEEE Int. Conf. on Computing, Power and Communication Technologies (IC2PCT)*, 2024, pp. 205-210.

[12] H. A. Khalil, S. A. Hammad, H. E. Abdelmunim, and S. A. Maged, "Low-cost driver monitoring system using deep learning," *IEEE Access*, vol. 13, pp. 14151-14164, 2025.

[13] TensorFlow Team, "TensorFlow Lite for Mobile and Edge Devices," Available: https://www.tensorflow.org/lite, 2024.

[14] Firebase Team, "Firebase Real-time Database Documentation," Available: https://firebase.google.com/docs/database, 2024.

[15] C. N. G. Babu, K. T. Chandrashekhara, J. Verma, and M. Thungamani, "Real-time alert system to prevent car accidents," in *Proc. 2021 Int. Conf. on Forensics, Analytics, Big Data, Security (FABS)*, 2021, pp. 1-6.

[16] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[17] C. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," *arXiv preprint arXiv:1906.08172*, 2019.

[18] T. Soukupová and J. Čech, "Eye blink detection using facial landmarks," in *21st Computer Vision Winter Workshop*, 2016.