Project Progress Report 03

For

Safe Driver: AI powered Real Time Driver Monitoring and Accident Prevention System

Prepared by:

CodeCrafters Team

Team Members:

D. S. Sandeepa 221444563

R. U. Gonalagoda 621429318

P. S. Ekanayake 621429276

K. J. L. Perera 621429815

T. P. Denagamage 321429305

MAY 2026

Table of Contents

List of Tables

List of Figures

Document Revisions

Date	Version	Description	Author
28/04/2026	V1	Initial Progress Report 03	K.J.L. Perera
05/05/2026	V1.1	Reviewed Version	D.S. Sandeepa R.U. Gonalagoda

Table 1 : Document version history

Declaration

We hereby declare that this progress report 3 is our original work and does not include, without proper acknowledgment, any material previously published or submitted for a degree or diploma at any other university or institution of higher learning. To the best of our knowledge and belief, it does not contain any material written or published by another person except where such acknowledgement is made within the text.

Signature: ..... Date: 03.05.2026

D. S. Sandeepa

Signature: ..... Date: 03.05.2026

R. U. Gonalagoda

Signature: ..... Date: 03.05.2026

P. S. Ekanayake

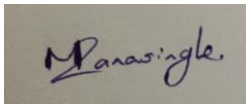
Signature: ..... Date: 03.05.2026

K. J. L. Perera

Signature: ..... Date: 03.05.2026

T. P. Denagamage

I have reviewed the progress report and confirm that it adheres to the provided progress report 3 outline. I hereby approve the report for submission.



Signature of the supervisor: ..... Date: 04.05.2026

Dr. D.D.M. Ranasinghe

Executive Summary

This Progress Report 3 presents the detailed design and test-readiness status of the SafeDriver system, an AI-powered, real-time driver monitoring and accident prevention solution for Sri Lankan public bus fleets. Building on the requirements and architectural design established in earlier reports, the project team has now completed a comprehensive database design, screen-level GUI prototypes, and a structured test plan with representative test cases. Collectively, these artefacts translate the conceptual system into an implementable blueprint that spans the embedded device, cloud backend, web dashboard, and passenger mobile application.

On the data side, a normalized logical model and ER diagram have been refined into a concrete schema and mapped to the Firebase Firestore platform used by the implementation teams. The GUI work has produced consistent, multilingual interfaces for drivers, passengers, fleet operators and transport authorities, validated through iterative prototyping. In parallel, the team has prepared detailed functional and non-functional tests aligned with the SRS and has continued to apply configuration-management and CMMI-style practices to control changes and document decisions. Overall, the project has reached approximately 80% completion, with design and planning activities largely finalized and the focus now shifting towards full implementation, integration, and pilot deployment.

1. Introduction

1.1 Problem Statement

Road traffic accidents remain one of the leading causes of death and injury worldwide, with driver fatigue and distraction being significant contributing factors. In Sri Lanka's public transportation sector, the lack of real-time driver behavior monitoring systems creates a critical gap in safety management. Current solutions either lack intelligence, require expensive infrastructure, or fail to provide actionable real-time alerts. Public transport passengers lack transparency regarding their safety, and transport authorities struggle to track compliance and incident patterns across their fleet.

1.2 Project Aims

- Develop an intelligent, edge-based driver monitoring system capable of detecting unsafe driver behaviors in real time.
- Provide immediate alerts to drivers and escalate critical incidents to transport authorities.
- Measure driving quality and other details by enabling passengers with transparent feedback options through a mobile application.
- Empower transport authorities with comprehensive fleet monitoring, analytics, and compliance reporting.

1.3 Objectives

- Detect drowsiness using Eye Aspect Ratio (EAR), blink frequency, head pose and yawning detection.
- Identify distractions (mobile phone usage) via CNN models.
- Implement a multi-level alert system (buzzer, voice, authority notifications).
- Integrate GPS-based hazard mapping for accident-prone zones.
- Develop a passenger transparency mobile application.
- Generate automated compliance reports for authorities.

The project has made strong progress toward its objectives. Detection of drowsiness using EAR, blink frequency, head pose, and yawning, as well as distraction detection via mobile phone usage models, is largely implemented and functioning as intended. The multi-level alert system with buzzer and voice feedback, together with the passenger transparency mobile application,

is also mostly complete, with core features available for testing and demonstration. GPS-based hazard mapping has been partially implemented, with basic integration working but further data collection and tuning still required to reach full functionality. Automated compliance reporting for authorities is currently in progress, with initial structures and data flows defined but not yet finalized or fully validated.

1.4 Purpose

The purpose of Progress Report 3 (PR3) is to document the outcomes of the detailed design and quality-assurance preparation phase of the SafeDriver project. While Progress Report 2 focused on system architecture, requirements, and high-level design, PR3 explains how these designs were refined into concrete database models, graphical user interfaces, and test artefacts that directly guide implementation. The report also evidences ongoing process maturity through configuration management and CMMI-style meeting minutes for this phase.

1.5 Project Overview

SafeDriver remains a modular, scalable driver-monitoring system targeting Sri Lankan public bus fleets, built on embedded computer-vision hardware, a cloud backend, and mobile and web applications. The system continuously analyses driver behavior to detect drowsiness and distraction and escalates alerts to drivers, authorities, and passengers via a hybrid edge-cloud architecture. In PR3, this architecture has been refined into specific data structures, screen flows, and testable behaviors ready for implementation across all four tiers: embedded device, edge analytics, cloud services, and user applications.

1.6 Development Progress Achieved

During the PR3 period the team concentrated on *detailed design* and *test readiness*:

- Completed a full database design, including ER diagrams, collections / documents definitions, and Firestore mappings for vehicles, drivers, routes, alerts, feedback, and passengers.
- Produced a working GUI prototype covering the authority dashboard, and passenger mobile application using Figma and early mobile/web implementations.
- Prepared a structured test plan with functional and non-functional test cases mapped to SRS requirements.
- Continued to apply configuration-management and CMMI practices with documented meetings, baselines, and code-review sessions.

1.7 Scope of Progress Report 3

The scope of PR3 is limited to:

- Detailed database design (data dictionary, ER diagrams, and Firebase mapping).
- GUI design and prototypes for both authority dashboard and passenger mobile application.
- Test plan and representative test cases.
- CMMI-compliant meeting minutes for the detailed-design and early implementation period.

Summary of Chapter 1

Chapter 1 restates the original problem; project aims and objectives from Progress Report 2 so that the SafeDriver vision and expected outcomes remain consistent across all phases. It then explains that, after PR1 documented the requirements and PR2 established the high-level architecture and design, Progress Report 3 specifically addresses the detailed design and quality-assurance preparation stage of the project. This chapter also clarifies that PR3 focuses on finalizing the database design, completing GUI prototypes for both authority dashboard and passenger mobile application, defining a structured test plan and recording CMMI-compliant process documentation to support the upcoming implementation and pilot activities.

2. Literature Review Summary

The literature on driver monitoring systems identifies four main technical approaches: image-based methods using facial features, biological-signal-based methods (EEG, ECG), vehicle-based measures (steering and lane behavior), and hybrid combinations of these. Image-based deep learning models such as 3D CNNs and CNN-LSTM networks have achieved over 90% accuracy for drowsiness detection in controlled environments, while specialized architectures can detect mobile-phone usage and other distractions with accuracy above 95%. These studies demonstrate that computer-vision and deep learning provide a strong technical foundation for real-time detection of risky behaviors, but they are often evaluated under ideal laboratory conditions.

Commercial driver monitoring products from international vendors show that integrated camera-based systems with multi-level alerts and cloud connectivity are technically mature. However, most of these solutions are designed for left-hand-drive vehicles, temperate climates, and high-income markets, leading to limitations when applied directly in Sri Lanka. Key gaps include limited support for Sinhala/Tamil, incomplete specifications for high-humidity tropical environments, and high per-vehicle costs that are difficult to sustain at national scale.

The Sri Lankan road safety context further emphasizes the need for a localized solution. National statistics show thousands of annual road fatalities, with bus accidents strongly linked to fatigue, distraction, and aggressive driving patterns in competitive private operations. The government's 2025 rapid deployment initiative for AI-powered driver monitoring systems confirmed the policy priority but relied primarily on imported commercial systems with limited localization. Studies and government reports highlight additional constraints such as intermittent rural connectivity, right-hand-drive configurations, and economic pressure on transport operators, all of which affect technology adoption.

From this review, several critical research gaps and practical challenges emerge lack of solutions optimized for right-hand-drive buses, insufficient environmental hardening for tropical climates, limited multilingual and culturally adapted user interfaces, and high-cost barriers for large-scale deployment. The SafeDriver system is positioned to address these gaps through locally developed, Raspberry-Pi-based hardware, TensorFlow Lite models optimized for edge deployment, offline-capable GPS hazard mapping, and multilingual interfaces for drivers, passengers, and authorities.

Summary of Chapter 2

This chapter reviews research and commercial solutions for driver monitoring, covering image-based, bio-signal, vehicle-based, and hybrid approaches along with state-of-the-art deep learning models. It highlights that existing systems are accurate but mostly evaluated in ideal conditions and are poorly adapted to Sri Lanka's right-hand-drive vehicles, tropical climate, language needs, and cost constraints. The chapter concludes that these gaps justify a localized, low-cost, Raspberry-Pi-based, multilingual SafeDriver system with edge AI and offline-capable features tailored to Sri Lankan public transport.

3. Methodology and Proposed Solution

3.1 Methodology

The project follows an iterative, Agile-inspired methodology that is aligned with the university's progress-report milestones and CMMI-oriented process expectations. Work began with stakeholder consultations, problem analysis and a comprehensive literature review, which were synthesised into the Software Requirements Specification (SRS) containing functional, non-functional and pilot-success criteria. Based on this specification, the team developed UML diagrams and a high-level four-tier architecture in Progress Report 2, then refined these into detailed database schemas, GUI designs and test artefacts during Progress Report 3.

Throughout the methodology, tasks are managed using GitHub projects and GitHub repositories, ensuring that requirements, design changes and implementation work are tracked and reviewed systematically. Regular meetings with the supervisor and within the team are documented through CMMI-compliant minutes, capturing decisions, risks and action items that feed back into subsequent iterations. Testing activities are planned from early stages, with unit, integration, system and pilot-level tests defined against the SRS, so that each prototype increment can be evaluated and improved in a controlled manner.

3.2 Proposed Solution

The proposed SafeDriver solution is implemented as a hybrid edge-cloud platform designed specifically for Sri Lankan public bus operations. At the edge, a Raspberry Pi-based monitoring unit equipped with a camera and GPS module runs Python, OpenCV, MediaPipe , TensorFlow Lite and YOLO to track facial landmarks, compute Eye Aspect Ratio, Mouth Aspect Ratio and related metrics, estimate head pose and classify distraction behaviours such as mobile-phone use, yawning and smoking in real time. When unsafe behaviour persists beyond configured thresholds, the unit issues local audio alerts to the driver and simultaneously sends structured alert events with timestamps, GPS coordinates and driver identifiers to the cloud backend.

The cloud layer is built on Firebase, using Firestore to store drivers, vehicles, routes, alerts, feedback and configuration, Authentication for secure access control, Cloud Functions for server-side processing, and Cloud Messaging for notifications. On top of this backend, a web-based authority dashboard allows fleet operators and regulators to monitor buses on a live map, inspect incident histories, analyse route-level safety trends, manage thresholds and hazard zones, and generate compliance reports. A Flutter-based passenger mobile application

complements this by providing route and driver safety indicators, enabling passengers to submit structured feedback with optional images and locations, and maintaining a points mechanism to encourage participation. Together, this methodology and solution ensure that SafeDriver evolves from requirements to a coherent, testable prototype that can be evaluated against technical, operational and safety objectives.

Summary of Chapter 3

Chapter 3 explains that the project uses an iterative, Agile-style process: starting from SRS and UML design and moving through detailed database, GUI and test-plan development, with tasks and decisions tracked formally. It then outlines the hybrid edge-cloud solution in which a Raspberry-Pi unit performs real-time driver monitoring and sends events to a Firebase backend, which powers an authority dashboard and passenger mobile app. The chapter concludes that this approach produces a traceable, testable prototype aligned with the project's safety and operational objectives.

4. Detailed Database Design

4.1 Overview

The SafeDriver backend uses Cloud Firestore/Realtime Database as its primary operational database, complemented by Firebase Authentication and Cloud Storage. To maintain academic rigor, the team first modeled a relational ER design, then mapped it to Firestore collections and documents while preserving all key entities and relationships. This section presents the collection structures, relationships, and the correspondence between relational tables and Firestore collections.

4.2 Logical Data Model (Relational View)

4.2.1 Main Entities

The core logical entities are:

- VEHICLES – buses equipped with SafeDriver devices.
- DRIVERS – registered bus drivers with biometric profiles and safety scores.
- ROUTES – public transport routes (start, end, distance, KPIs).
- USERS – authentication and global profile/roles (driver, passenger, admin, authority).
- PASSENGER_DETAILS – passenger-specific profile, preferences, favourites.
- FEEDBACK – structured passenger feedback on drivers and vehicles.
- ALERTS – real-time unsafe behaviour events detected on road.
- POINT_TRANSACTIONS – loyalty/safety point history for passengers.
- DEVICES – embedded devices mounted in buses and bound to a driver and vehicle.
- CONFIGS – model and system configuration, including AI thresholds.

Each entity is documented through structured table descriptions and HTML tables for inclusion in the report or system design chapter.

4.3 Relationships

Key relationships are:

- ROUTES to VEHICLES via routeId.
- DRIVERS to VEHICLES via driverId.
- USERS / PASSENGERS to PASSENGER_DETAILS via userId ↔ uid (one profile per user).
- USERS / PASSENGERS to POINT_TRANSACTIONS via userId.
- USERS / PASSENGERS to FEEDBACK via userId.
- VEHICLES to FEEDBACK via busId.
- DRIVERS to FEEDBACK via driverId.
- VEHICLES to ALERTS via busNumber.
- DRIVERS to ALERTS via driverId.
- DEVICES to ALERTS via deviceId.
- VEHICLES to DEVICES via busNumber (each vehicle has a single registered device).

4.4 Firebase Physical Data Model

Firestore collections and key example documents include:

- vehicles/{busNumberPlate} – stores VEHICLES table fields and runtime telemetry.
- drivers/{driverId} – DRIVERS with biometric and safety attributes.
- routes/{routeId} – ROUTES including performance KPIs and stops.
- devices/{deviceId} – DEVICES including assignment to driver and vehicle.
- alerts/{deviceId}/{date}/{alertId} – ALERTS, partitioned by date for efficient queries.
- feedback/{feedbackId} – FEEDBACK linked to user, bus and driver.
- passenger_details/{uid} – PASSENGER_DETAILS.
- point_transactions/{transactionId} – POINT_TRANSACTIONS.
- configs/model_configurations – system-wide AI configuration parameters.

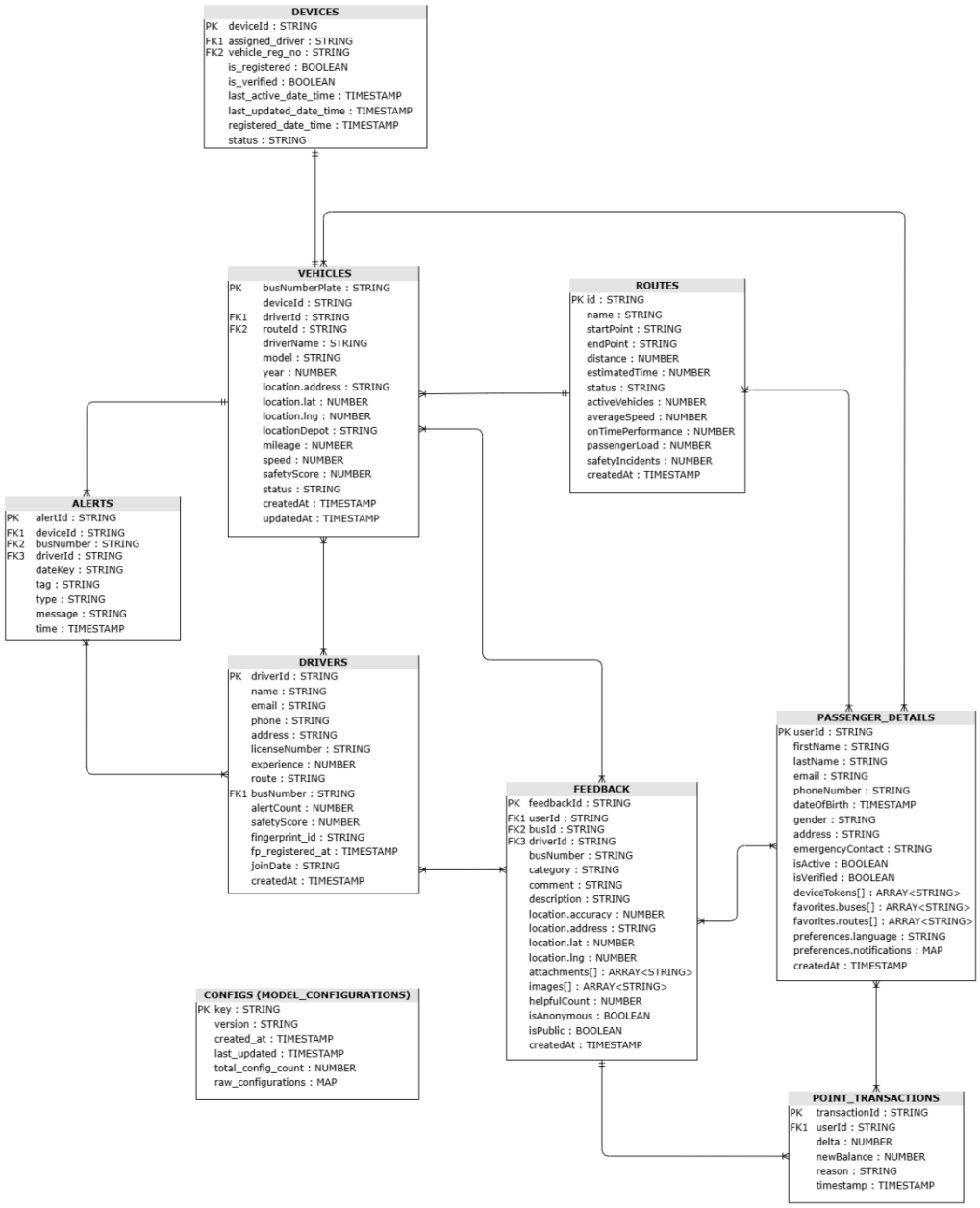


Figure 1 - Database Design

Summary of Chapter 4

This section shows that the data layer has moved from conceptual ideas to a complete, implementable model. It defines all key entities (vehicles, drivers, routes, passengers, alerts, feedback, devices, configs, points) and the relationships between them, then maps these cleanly onto Firebase Firestore collections. By the end of the section, both an ER diagram and a detailed data dictionary exist, so developers and supervisors can see exactly how operational data and analytics will be stored, linked, and validated.

5. GUI Detailed Design and Prototypes

Design Principles and Tools

The graphical user interface design follows the principles of clarity, safety-critical visualization, and multilingual usability described in the PR2 non-functional requirements. Figma was used as the main tool for wireframing and interactive prototyping, ensuring consistent style and reusable components across the embedded UI, authority dashboard, and passenger mobile application.

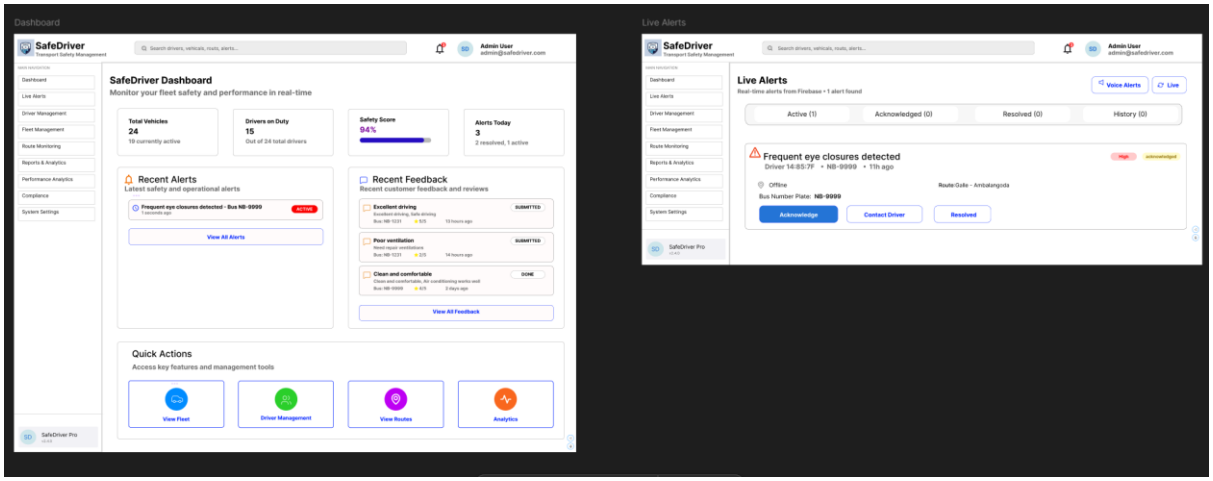
5.1 Authority Dashboard Screens

The web-based dashboard for fleet operators and transport authorities includes:

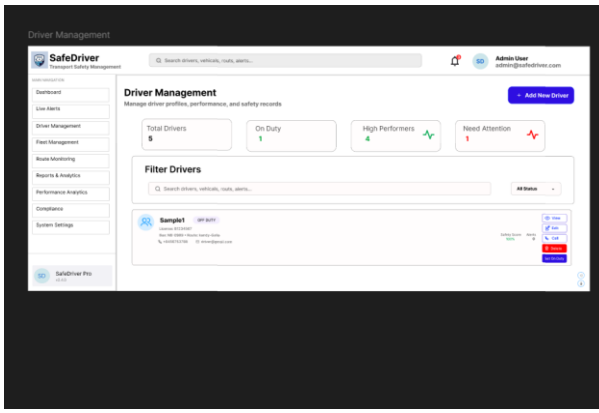
- **Login & access control** screens integrated with Firebase Authentication and role-based routing.
- A **Fleet Overview** homepage with map visualization of active buses, color-coded by safety status and route.
- **Driver Management** views for listing drivers, viewing safety scores, and checking biometric registration status.
- **Live Alert** pages with filterable tables by route, driver, alert type, and time range.
- **Route Monitoring** pages with charts for incidents per route, on time performance.
- **Configuration** views for managing alert thresholds, hazard zones, and notification recipients.

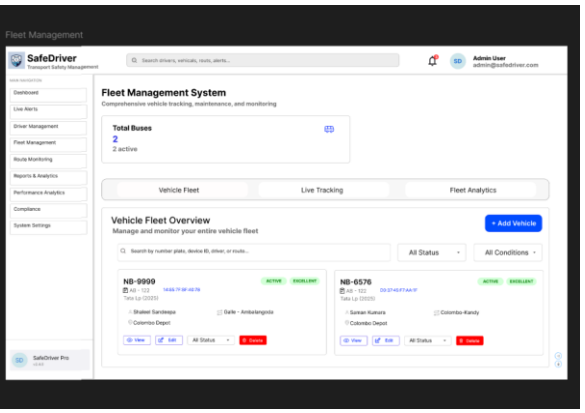
Figma Link:

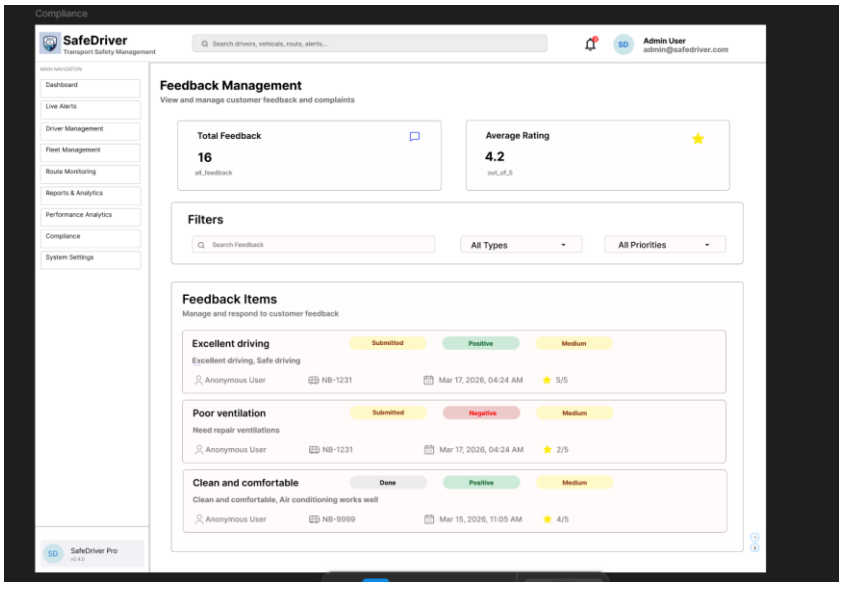
https://www.figma.com/design/ib3qEk3sChFRF4fi3UuQZ0/Untitled?node-id=0-1&t=rHqVNtVk0vdxBwrj-1


Dashboard
SafeDriver
SafeDriver Dashboard
Monitor your fleet safety and performance in real-time
Total Vehicles
24
Drivers on Duty
15
Safety Score
94%
Alerts Today
3
Recent Alerts
Recent Feedback
Quick Actions
Access key features and management tools
View Fleet
Driver Management
View Routes
Analytics

Live Alerts
SafeDriver
Live Alerts
Real-time alerts from Firebase • 1 alert found
Active (1)
Acknowledged (0)
Resolved (0)
History (0)
Frequent eye closures detected
Acknowledge
Contact Driver
Resolved


Driver Management
SafeDriver
Driver Management
Manage driver profiles, performance, and safety records
Add New Driver
Total Drivers
5
On Duty
1
High Performers
4
Need Attention
1
Filter Drivers
Sample1


Fleet Management
SafeDriver
Fleet Management System
Comprehensive vehicle tracking, maintenance, and monitoring
Total Buses
2
2 active
Vehicle Fleet
Live Tracking
Fleet Analytics
Vehicle Fleet Overview
Manage and monitor your entire vehicle fleet
Add Vehicle
NB-9999
NB-0576


Compliance
SafeDriver
Feedback Management
View and manage customer feedback and complaints
Total Feedback
16
Average Rating
4.2
Filters
All Types
All Priorities
Feedback Items
Manage and respond to customer feedback
Excellent driving
Submitted
Positive
Medium
Excellent driving, Safe driving
Poor ventilation
Negative
Need repair ventilations
Clean and comfortable
Done
Clean and comfortable, Air conditioning works well

5.2 Passenger Mobile Application Screens

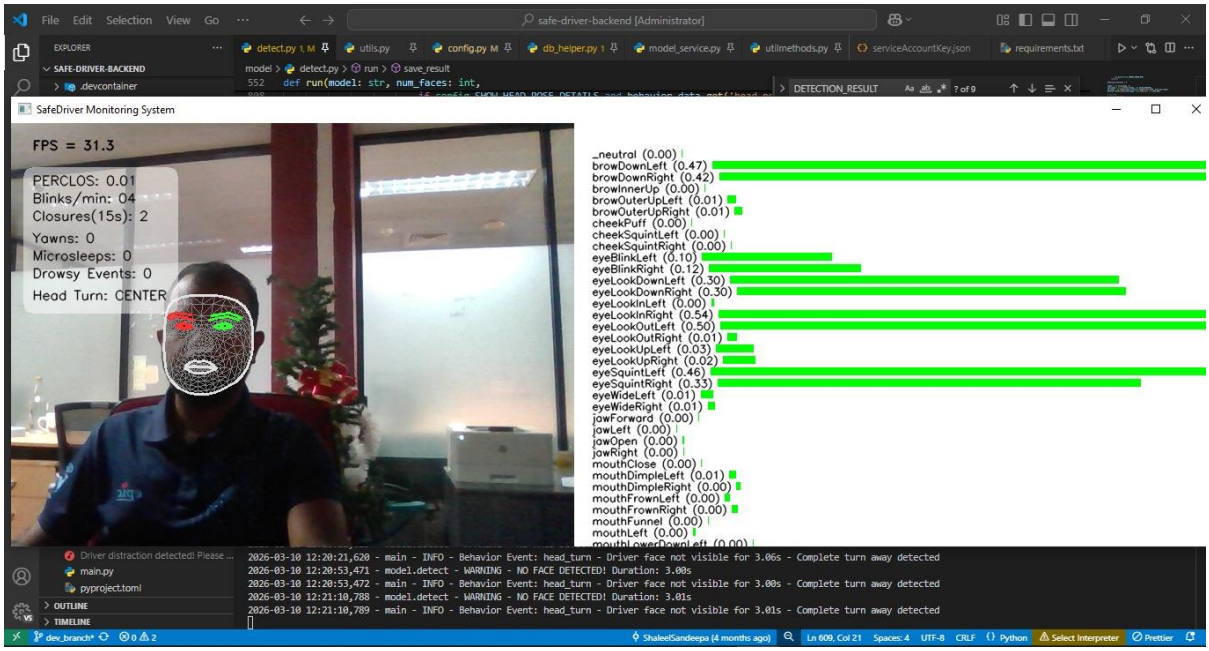
The passenger app, designed using Flutter and guided by PR2 requirements, includes:

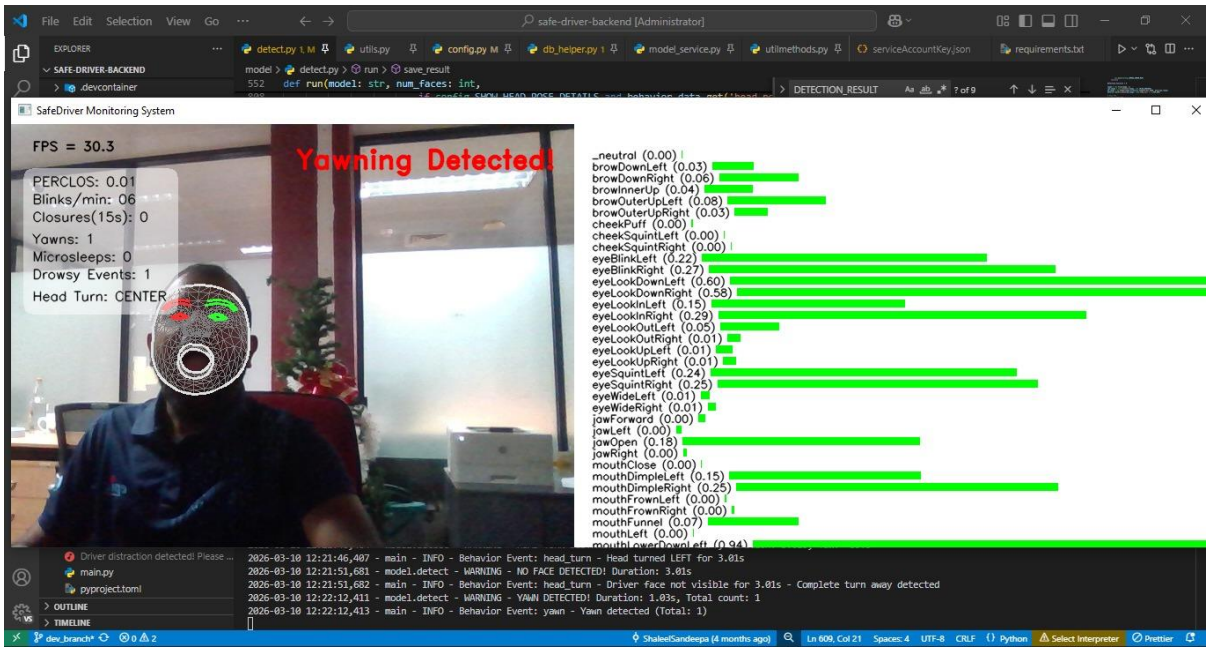
- Registration and login screens using email/OTP and storing profiles in passenger details.
- A **Home** screen that displays current route information, driver safety status, and last alert summaries for the selected bus.
- A **Feedback** form allowing passengers to submit comments, select categories, attach images, and optionally remain anonymous.
- A **Points Wallet** showing current points and a list of point transactions with reasons and timestamps.
- **Favorites** management for saving commonly used buses and routes for quick access.

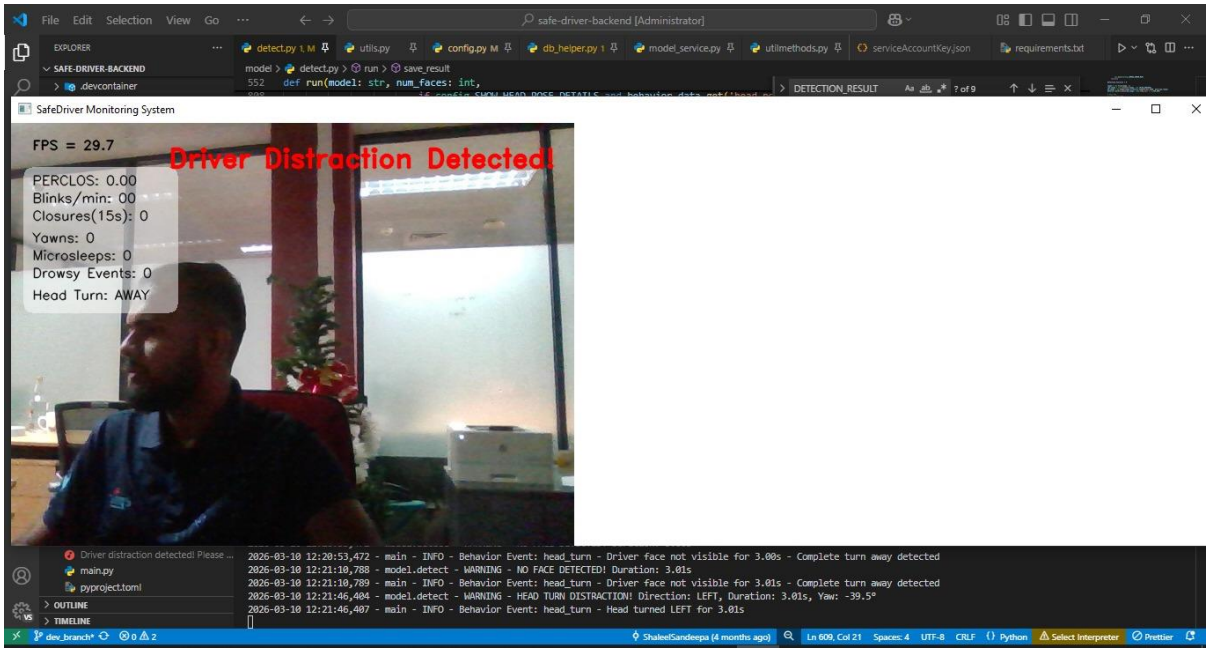
Figma Link: https://www.figma.com/design/X6BaOAPspOH2MycPN4qtFp/Untitled?node-id=7-23&t=krzd1UM5u4BTbwxn-1

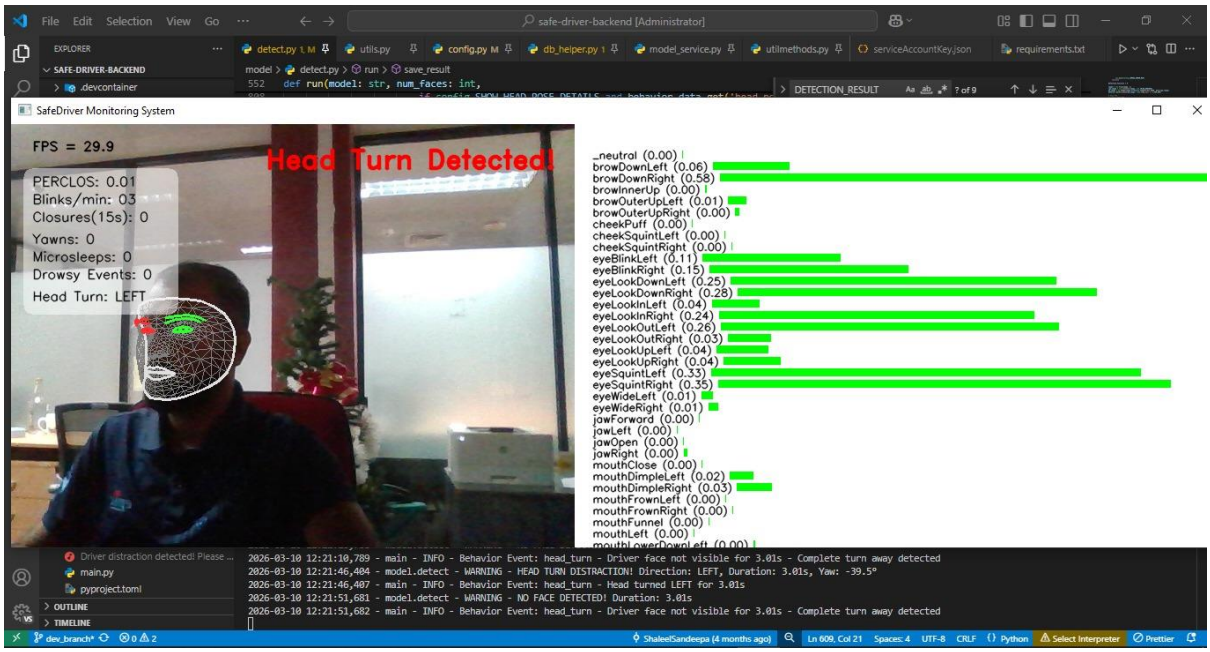
6. Implementation

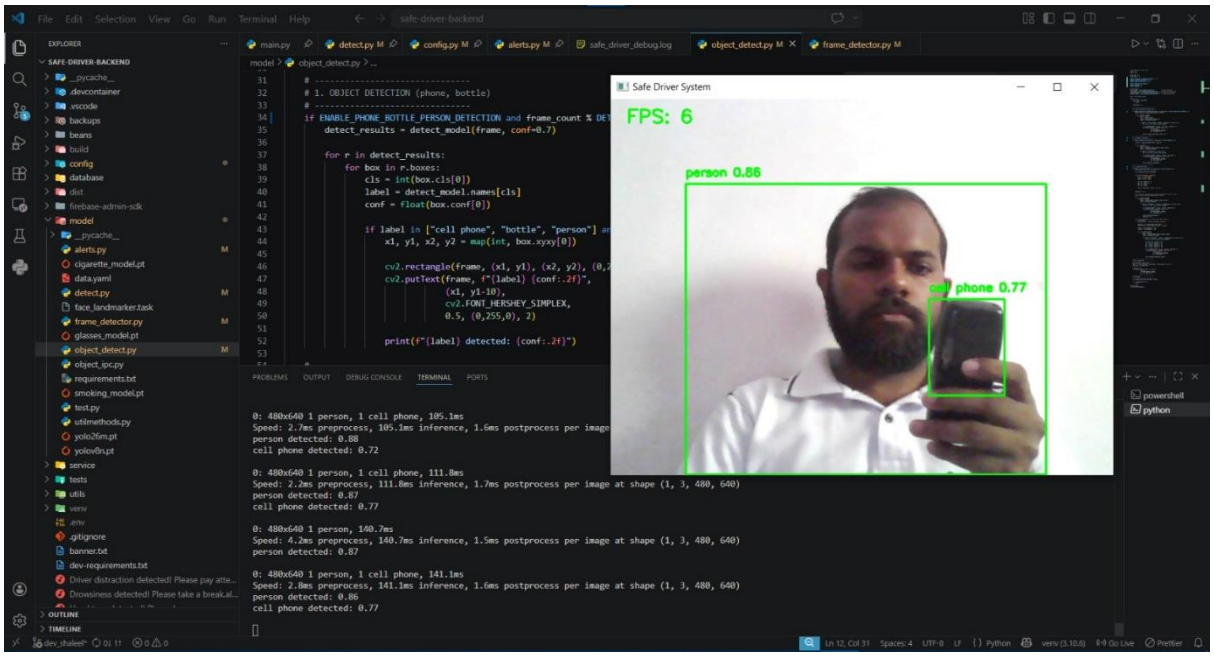
6.1 Monitoring System

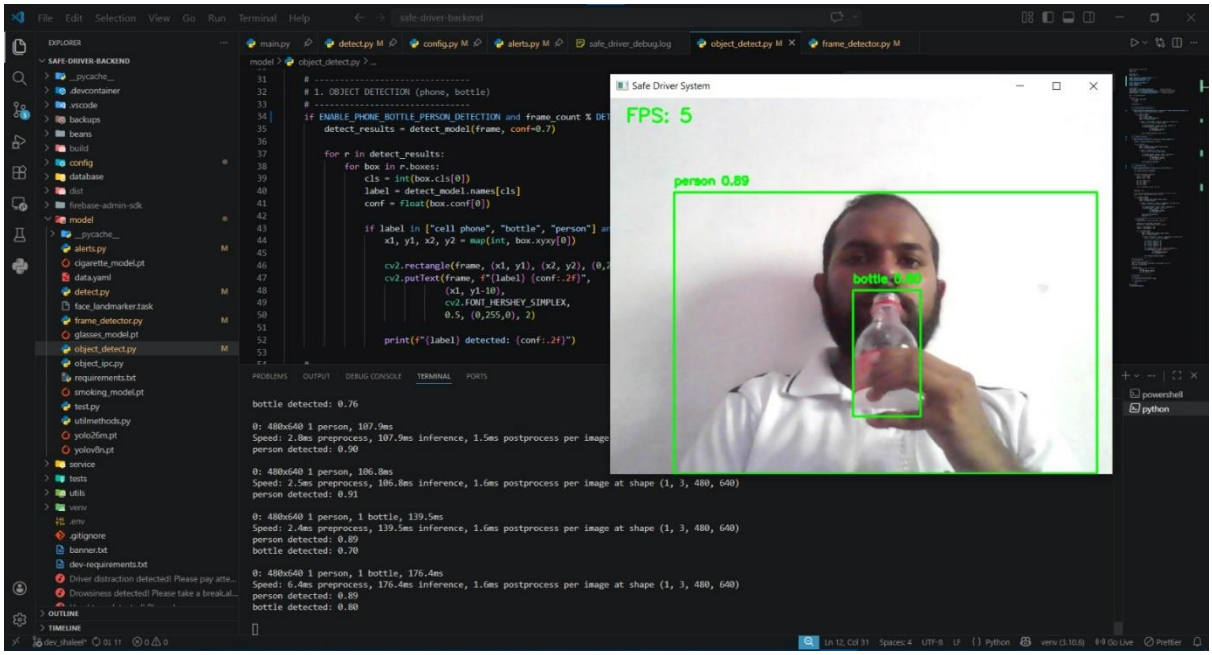


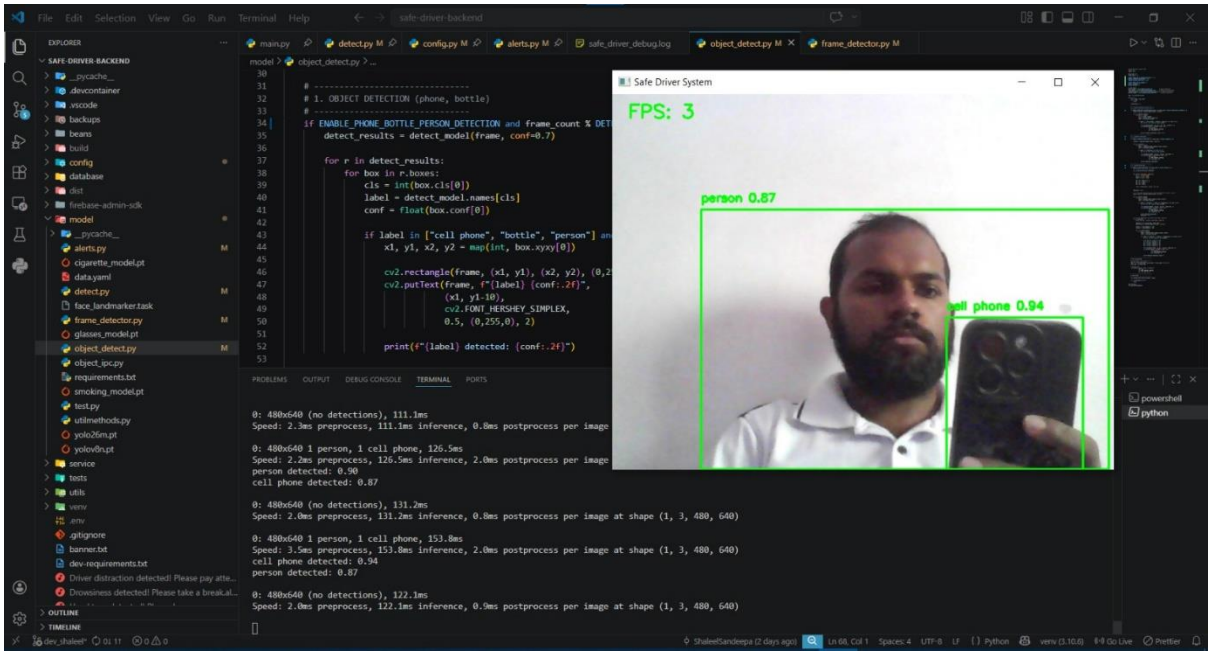


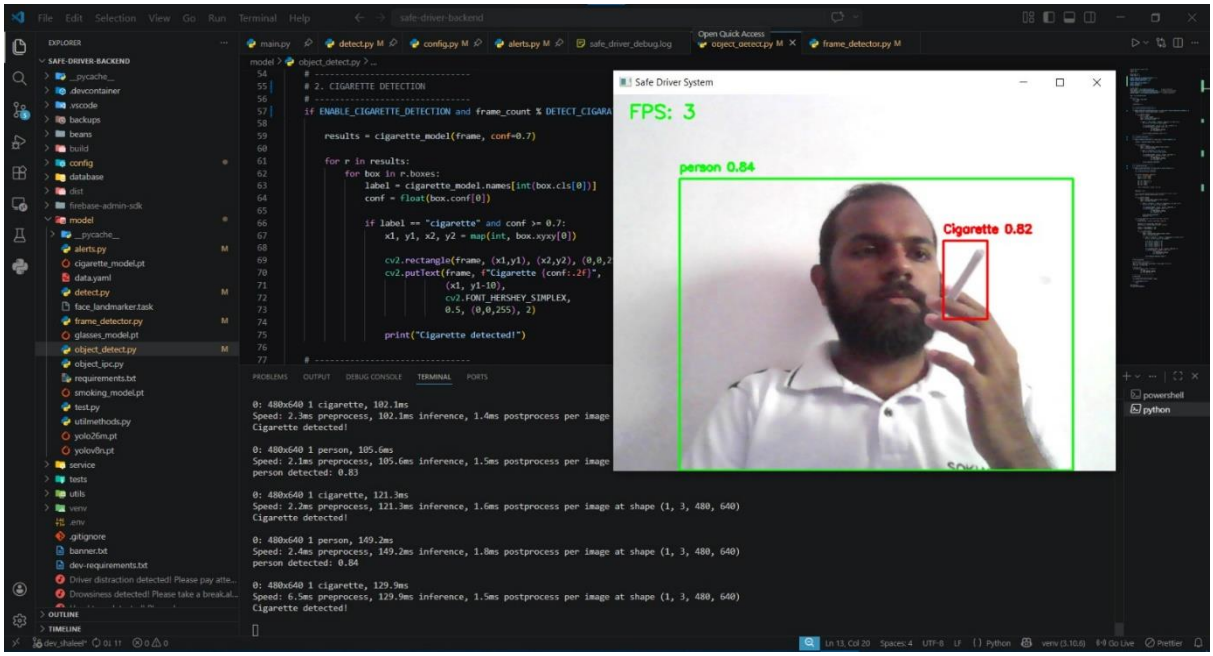

SafeDriver Monitoring System
FPS = 29.7
Driver Distraction Detected!
PERCLOS: 0.00
Blinks/min: 00
Closures(15s): 0
Yawns: 0
Microsleeps: 0
Drowsy Events: 0
Head Turn: AWAY
2026-03-10 12:20:53,472 - main - INFO - Behavior Event: head_turn - Driver face not visible for 3.00s - Complete turn away detected
2026-03-10 12:21:10,788 - model.detect - WARNING - NO FACE DETECTED! Duration: 3.01s
2026-03-10 12:21:10,789 - main - INFO - Behavior Event: head_turn - Driver face not visible for 3.01s - Complete turn away detected
2026-03-10 12:21:46,404 - model.detect - WARNING - HEAD TURN DISTRACTION! Direction: LEFT, Duration: 3.01s, Yaw: -39.5°
2026-03-10 12:21:46,407 - main - INFO - Behavior Event: head_turn - Head turned LEFT for 3.01s


SafeDriver Monitoring System
FPS = 29.9
Head Turn Detected!
PERCLOS: 0.01
Blinks/min: 03
Closures(15s): 0
Yawns: 0
Microsleeps: 0
Drowsy Events: 0
Head Turn: LEFT
_neutral (0.00)
browDownLeft (0.06)
browDownRight (0.58)
browInnerUp (0.00)
browOuterUpLeft (0.01)
browOuterUpRight (0.00)
cheekPuff (0.00)
cheekSquintLeft (0.00)
cheekSquintRight (0.00)
eyeBlinkLeft (0.11)
eyeBlinkRight (0.15)
eyeLookDownLeft (0.25)
eyeLookDownRight (0.28)
eyeLookInLeft (0.04)
eyeLookInRight (0.24)
eyeLookOutLeft (0.26)
eyeLookOutRight (0.03)
eyeLookUpLeft (0.04)
eyeLookUpRight (0.04)
eyeSquintLeft (0.33)
eyeSquintRight (0.35)
eyeWideLeft (0.01)
eyeWideRight (0.01)
jawForward (0.00)
jawLeft (0.00)
jawOpen (0.00)
jawRight (0.00)
mouthClose (0.00)
mouthDimpleLeft (0.02)
mouthDimpleRight (0.03)
mouthFrownLeft (0.00)
mouthFrownRight (0.00)
mouthFunnel (0.00)
mouthLeft (0.00)
2026-03-10 12:21:10,789 - main - INFO - Behavior Event: head_turn - Driver face not visible for 3.01s - Complete turn away detected
2026-03-10 12:21:46,404 - model.detect - WARNING - HEAD TURN DISTRACTION! Direction: LEFT, Duration: 3.01s, Yaw: -39.5°
2026-03-10 12:21:46,407 - main - INFO - Behavior Event: head_turn - Head turned LEFT for 3.01s
2026-03-10 12:21:51,681 - model.detect - WARNING - NO FACE DETECTED! Duration: 3.01s
2026-03-10 12:21:51,682 - main - INFO - Behavior Event: head_turn - Driver face not visible for 3.01s - Complete turn away detected

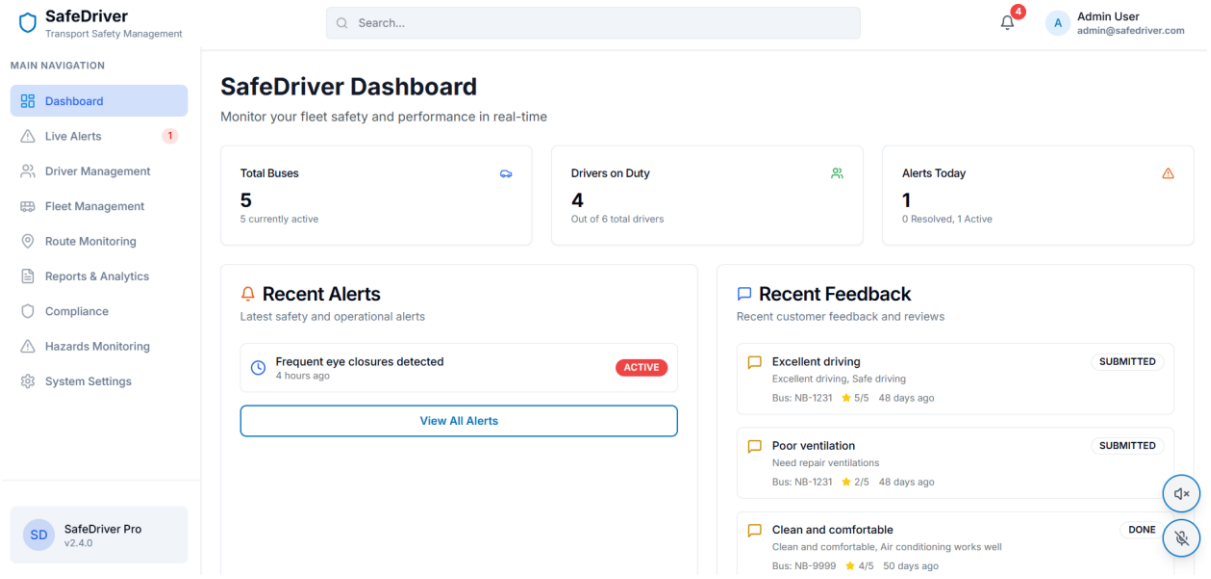

Safe Driver System
FPS: 6
person 0.86
cell phone 0.77
0: 480x640 1 person, 1 cell phone, 105.1ms
Speed: 2.7ms preprocess, 105.1ms inference, 1.6ms postprocess per image
person detected: 0.88
cell phone detected: 0.72
0: 480x640 1 person, 1 cell phone, 111.8ms
Speed: 2.2ms preprocess, 111.8ms inference, 1.7ms postprocess per image at shape (1, 3, 480, 640)
person detected: 0.87
cell phone detected: 0.77
0: 480x640 1 person, 140.7ms
Speed: 4.2ms preprocess, 140.7ms inference, 1.5ms postprocess per image at shape (1, 3, 480, 640)
person detected: 0.87
0: 480x640 1 person, 1 cell phone, 141.1ms
Speed: 2.8ms preprocess, 141.1ms inference, 1.6ms postprocess per image at shape (1, 3, 480, 640)
person detected: 0.86
cell phone detected: 0.77

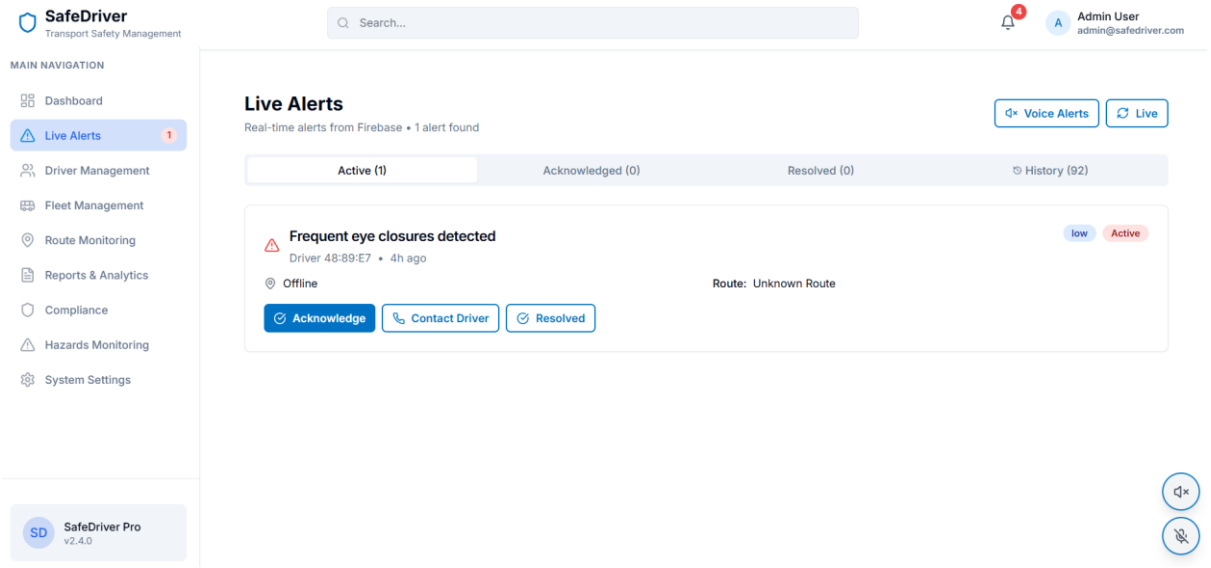

Safe Driver System
FPS: 5
person 0.89
bottle 0.89
bottle detected: 0.76
0: 480x640 1 person, 107.9ms
Speed: 2.8ms preprocess, 107.9ms inference, 1.5ms postprocess per image
person detected: 0.90
0: 480x640 1 person, 106.8ms
Speed: 2.5ms preprocess, 106.8ms inference, 1.6ms postprocess per image at shape (1, 3, 480, 640)
person detected: 0.91
0: 480x640 1 person, 1 bottle, 139.5ms
Speed: 2.4ms preprocess, 139.5ms inference, 1.6ms postprocess per image at shape (1, 3, 480, 640)
person detected: 0.89
bottle detected: 0.70
0: 480x640 1 person, 1 bottle, 176.4ms
Speed: 6.4ms preprocess, 176.4ms inference, 1.6ms postprocess per image at shape (1, 3, 480, 640)
person detected: 0.89
bottle detected: 0.80

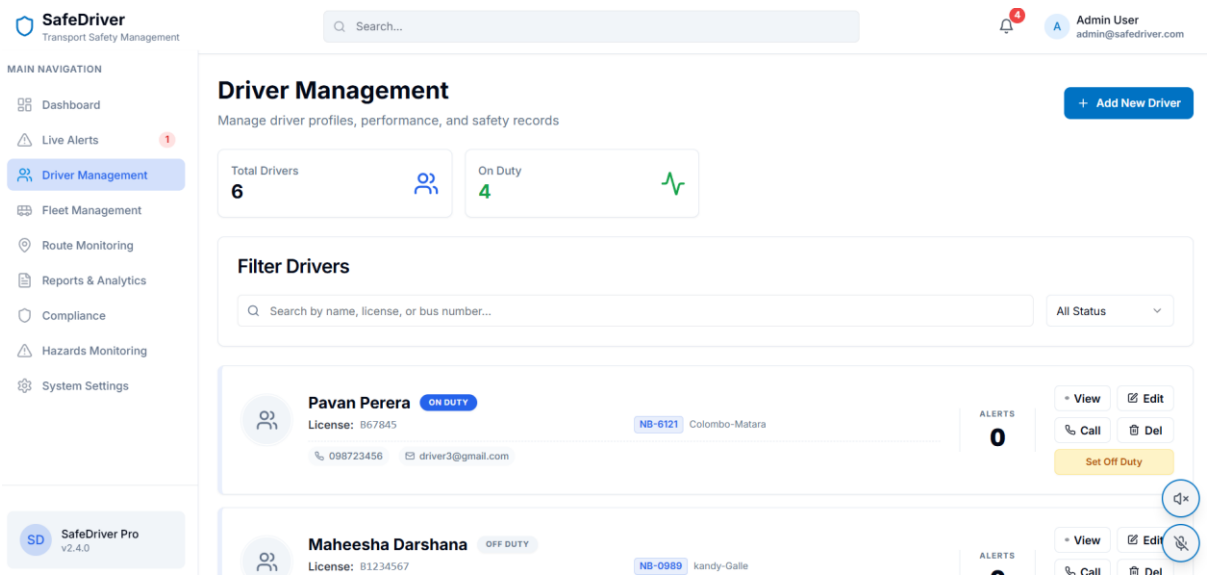

Safe Driver System
FPS: 3
person 0.87
cell phone 0.94

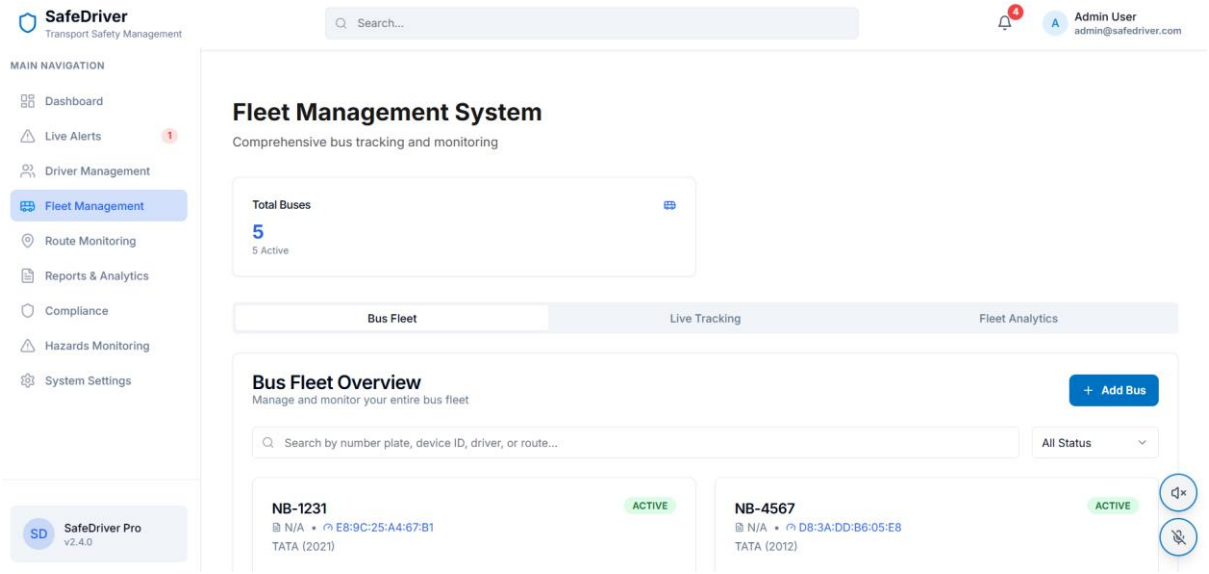

Safe Driver System
FPS: 3
person 0.84
Cigarette 0.82

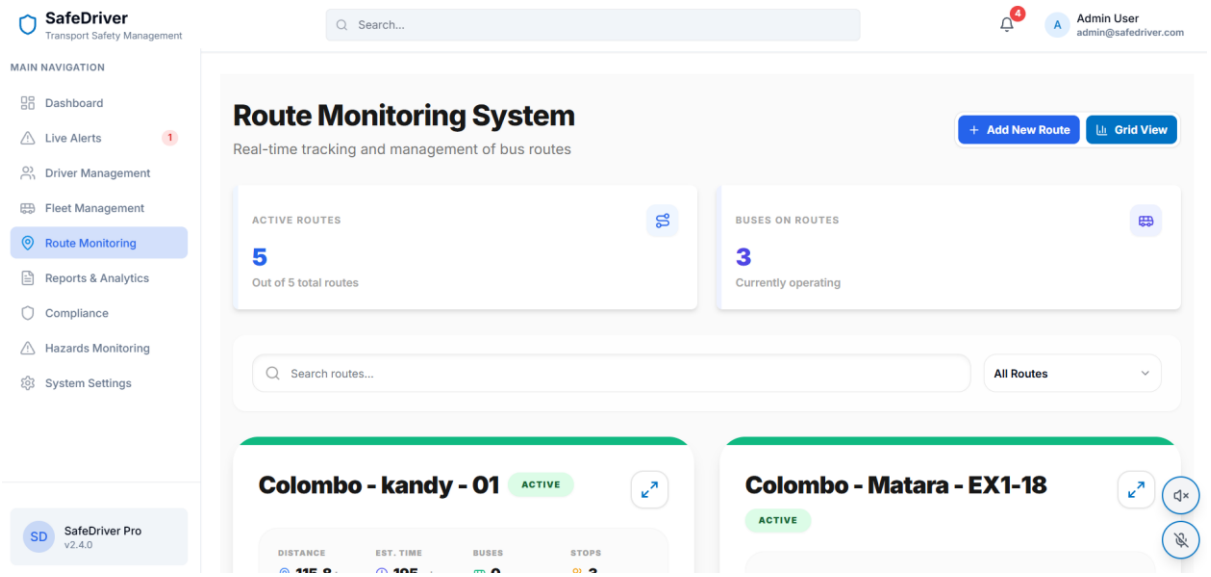
6.2 Authority Dashboard

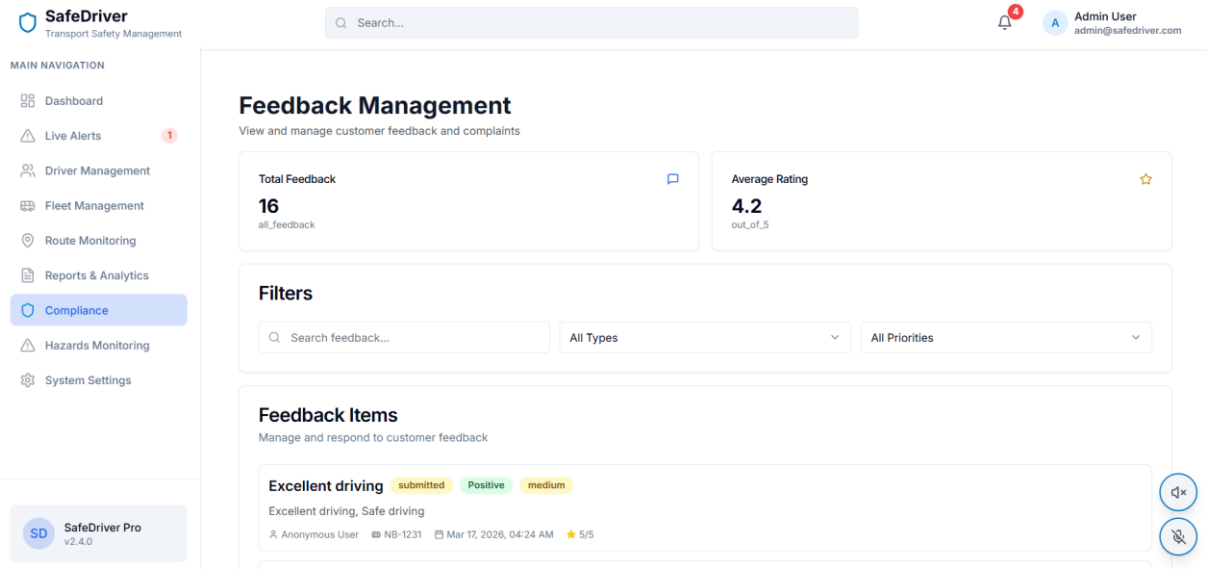


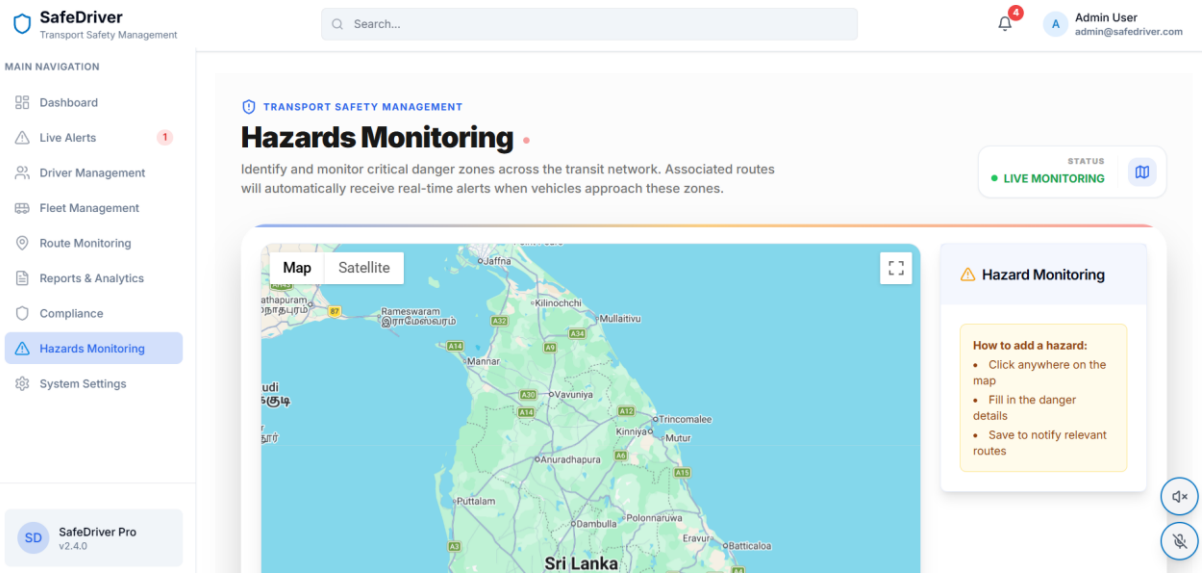



SafeDriver
Transport Safety Management
Search...
Admin User
admin@safedriver.com
MAIN NAVIGATION
Dashboard
Live Alerts
Driver Management
Fleet Management
Route Monitoring
Reports & Analytics
Compliance
Hazards Monitoring
System Settings
SafeDriver Pro
v2.4.0
Driver Management
Manage driver profiles, performance, and safety records
Add New Driver
Total Drivers
6
On Duty
4
Filter Drivers
Search by name, license, or bus number...
All Status
Pavan Perera
ON DUTY
License: B67845
NB-6121
Colombo-Matara
098723456
driver3@gmail.com
ALERTS
0
View
Edit
Call
Del
Set Off Duty
Maheesha Darshana
OFF DUTY
License: B1234567
NB-0989
kandy-Galle

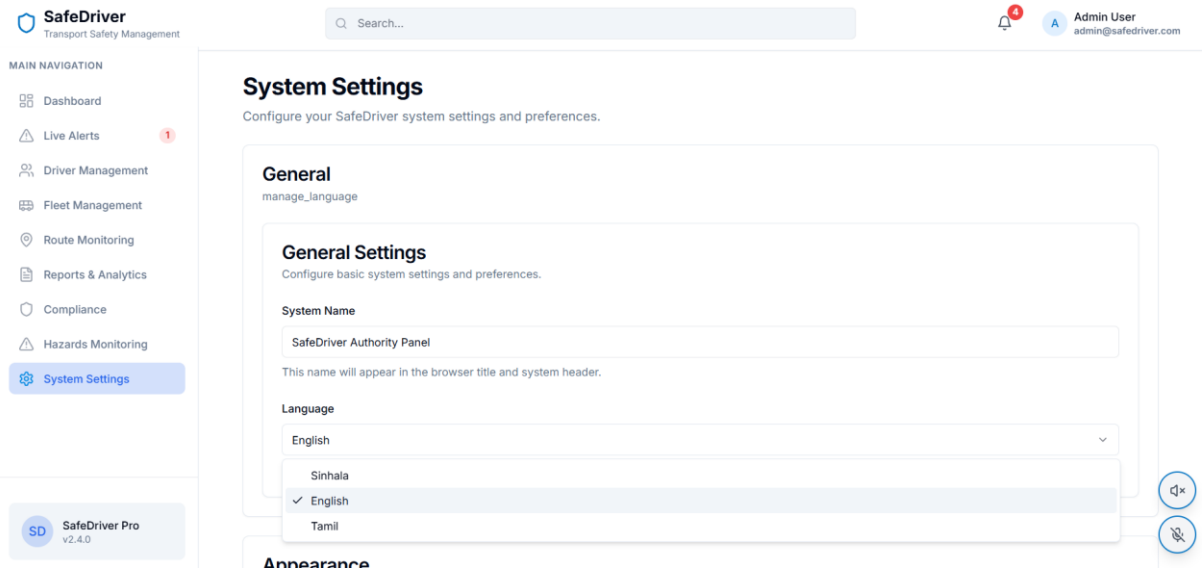

SafeDriver
Transport Safety Management
Search...
Admin User
admin@safedriver.com
MAIN NAVIGATION
Dashboard
Live Alerts
Driver Management
Fleet Management
Route Monitoring
Reports & Analytics
Compliance
Hazards Monitoring
System Settings
SafeDriver Pro
v2.4.0
Fleet Management System
Comprehensive bus tracking and monitoring
Total Buses
5
5 Active
Bus Fleet
Live Tracking
Fleet Analytics
Bus Fleet Overview
Manage and monitor your entire bus fleet
Add Bus
Search by number plate, device ID, driver, or route...
All Status
NB-1231
ACTIVE
N/A
E8:9C:25:A4:67:B1
TATA (2021)
NB-4567
ACTIVE
N/A
D8:3A:DD:B6:05:E8
TATA (2012)


SafeDriver
Transport Safety Management
Search...
Admin User
admin@safedriver.com
MAIN NAVIGATION
Dashboard
Live Alerts
Driver Management
Fleet Management
Route Monitoring
Reports & Analytics
Compliance
Hazards Monitoring
System Settings
SafeDriver Pro
v2.4.0
Route Monitoring System
Real-time tracking and management of bus routes
Add New Route
Grid View
ACTIVE ROUTES
5
Out of 5 total routes
BUSES ON ROUTES
3
Currently operating
Search routes...
All Routes
Colombo - kandy - 01
ACTIVE
DISTANCE
EST. TIME
BUSES
STOPS
Colombo - Matara - EX1-18
ACTIVE

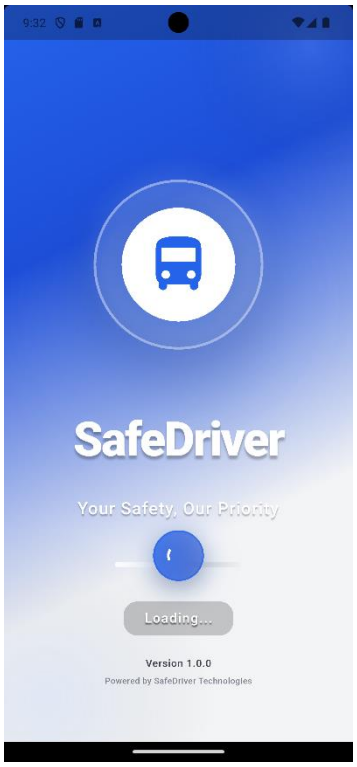

SafeDriver
Transport Safety Management
Search...
Admin User
admin@safedriver.com
MAIN NAVIGATION
Dashboard
Live Alerts
Driver Management
Fleet Management
Route Monitoring
Reports & Analytics
Compliance
Hazards Monitoring
System Settings
SafeDriver Pro
v2.4.0
Feedback Management
View and manage customer feedback and complaints
Total Feedback
16
all_feedback
Average Rating
4.2
out_of_5
Filters
Search feedback...
All Types
All Priorities
Feedback Items
Manage and respond to customer feedback
Excellent driving
submitted
Positive
medium
Excellent driving, Safe driving
Anonymous User
NB-1231
Mar 17, 2026, 04:24 AM
5/5

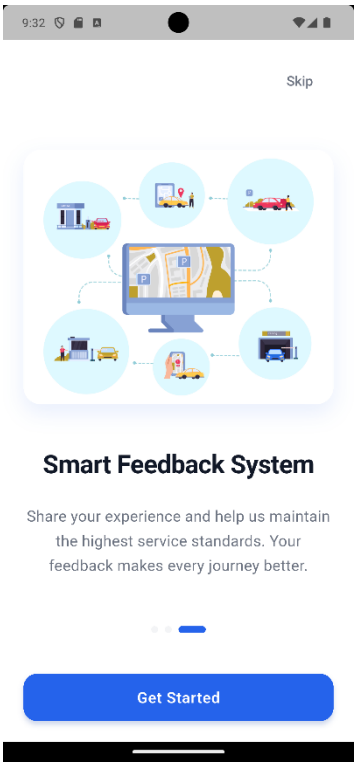

SafeDriver
Transport Safety Management
Search...
Admin User
admin@safedriver.com
MAIN NAVIGATION
Dashboard
Live Alerts
Driver Management
Fleet Management
Route Monitoring
Reports & Analytics
Compliance
Hazards Monitoring
System Settings
SafeDriver Pro
v2.4.0
TRANSPORT SAFETY MANAGEMENT
Hazards Monitoring
Identify and monitor critical danger zones across the transit network. Associated routes will automatically receive real-time alerts when vehicles approach these zones.
STATUS
LIVE MONITORING
Map
Satellite
Jaffna
Kilinochchi
Mullaitivu
Rameswaram
Mannar
Vavuniya
Trincomalee
Kinniya
Mutur
Anuradhapura
Puttalam
Dambulla
Polonnaruwa
Batticaloa
Sri Lanka
Hazard Monitoring
How to add a hazard:
Click anywhere on the map
Fill in the danger details
Save to notify relevant routes

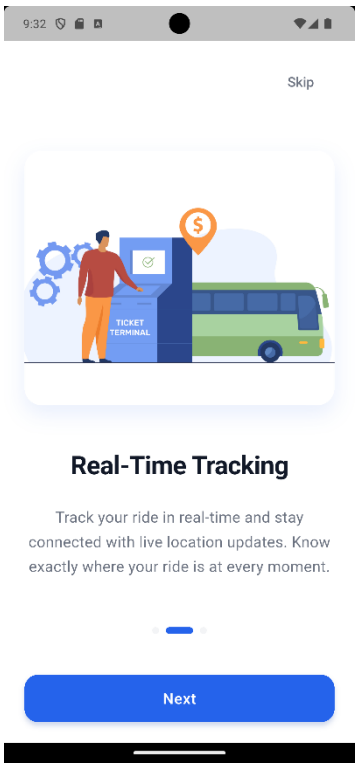
,

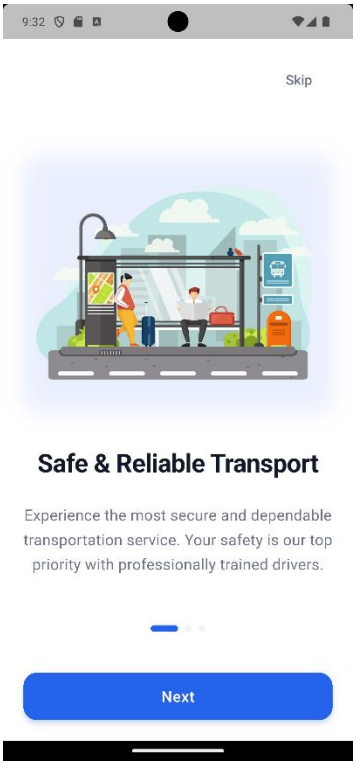

SafeDriver
Transport Safety Management
Search...
Admin User
admin@safedriver.com
MAIN NAVIGATION
Dashboard
Live Alerts
Driver Management
Fleet Management
Route Monitoring
Reports & Analytics
Compliance
Hazards Monitoring
System Settings
SafeDriver Pro
v2.4.0
System Settings
Configure your SafeDriver system settings and preferences.
General
manage_language
General Settings
Configure basic system settings and preferences.
System Name
SafeDriver Authority Panel
This name will appear in the browser title and system header.
Language
English
Sinhala
English
Tamil
Appearance

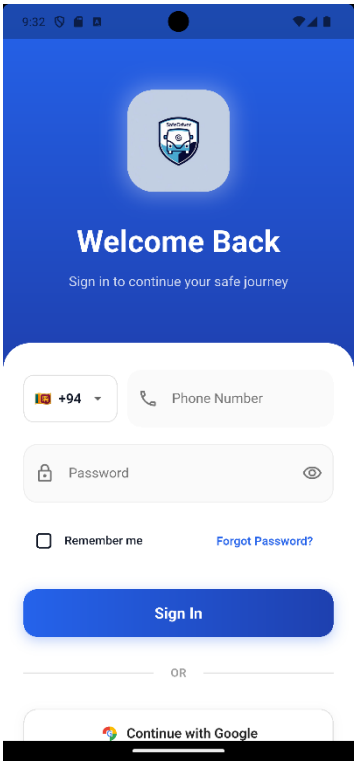
6.3 Passenger Mobile App

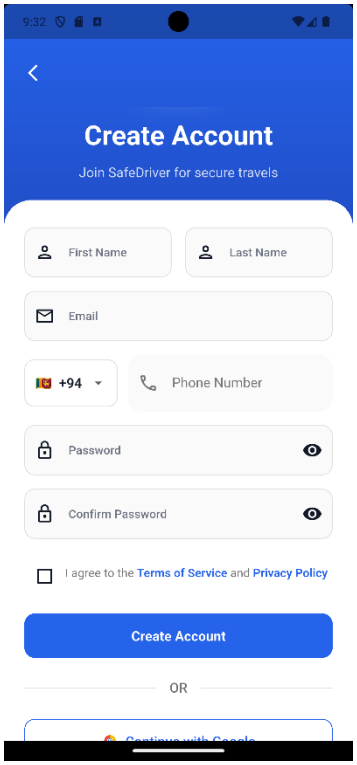


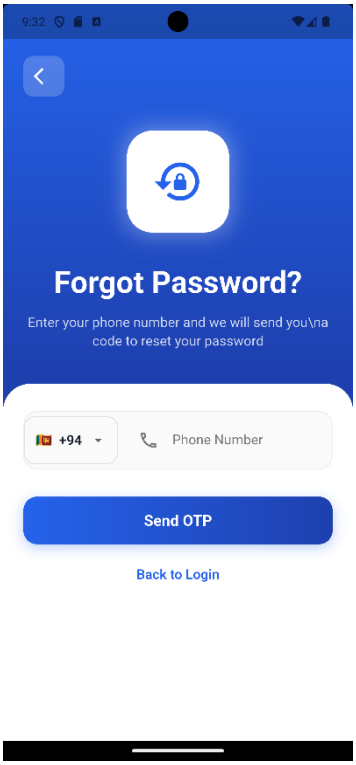


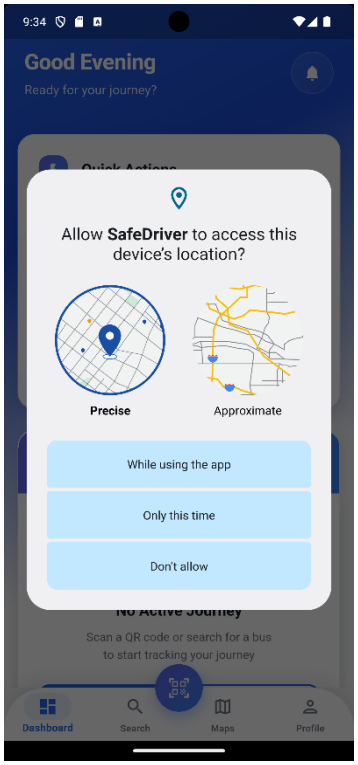


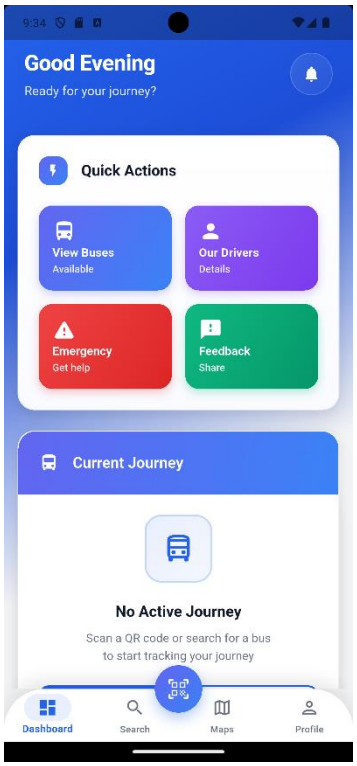


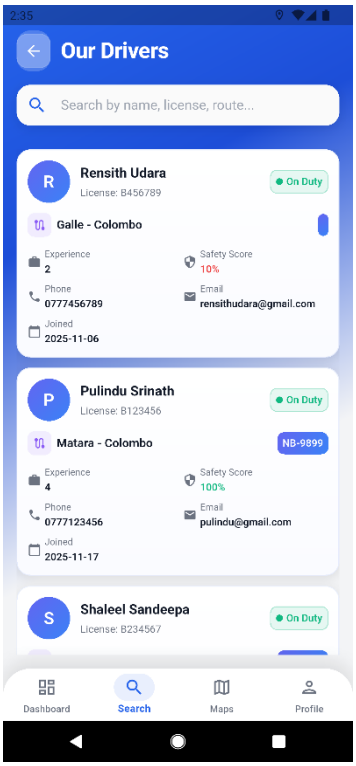


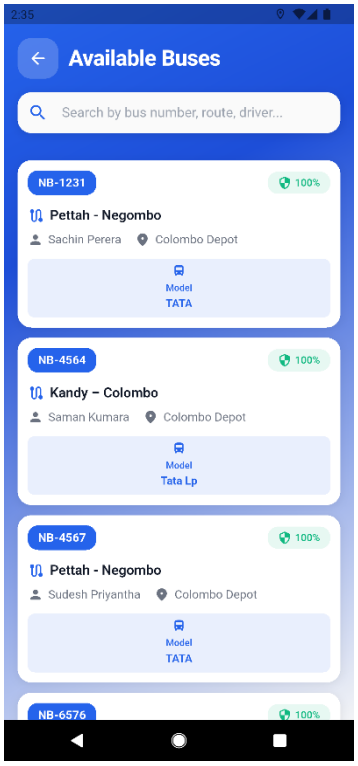

Create Account
Join SafeDriver for secure travels
First Name
Last Name
Email
+94
Phone Number
Password
Confirm Password
I agree to the Terms of Service and Privacy Policy
Create Account
OR

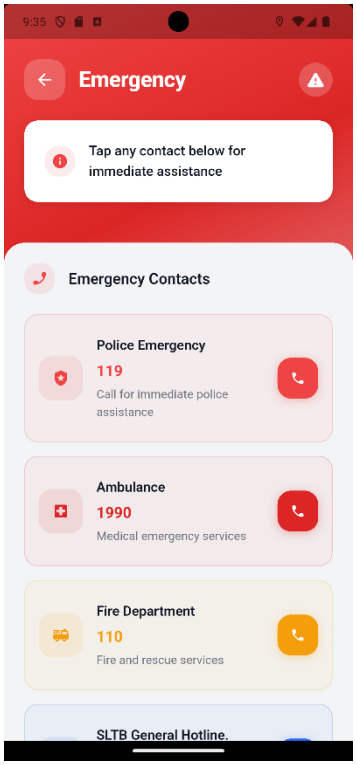

Forgot Password?
Enter your phone number and we will send you\na code to reset your password
+94
Phone Number
Send OTP
Back to Login

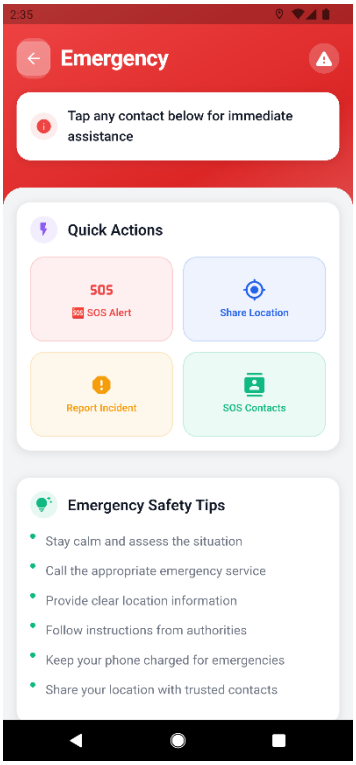

Good Evening
Ready for your journey?
Allow SafeDriver to access this device's location?
Precise
Approximate
While using the app
Only this time
Don't allow
Scan a QR code or search for a bus to start tracking your journey
Dashboard
Search
Maps
Profile

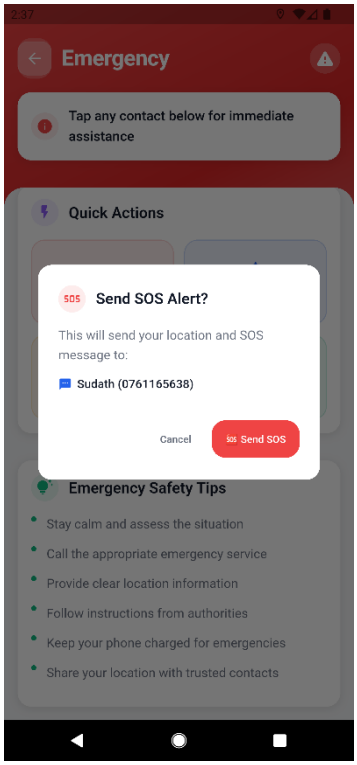

Good Evening
Ready for your journey?
Quick Actions
View Buses
Available
Our Drivers
Details
Emergency
Get help
Feedback
Share
Current Journey
No Active Journey
Scan a QR code or search for a bus to start tracking your journey
Dashboard
Search
Maps
Profile

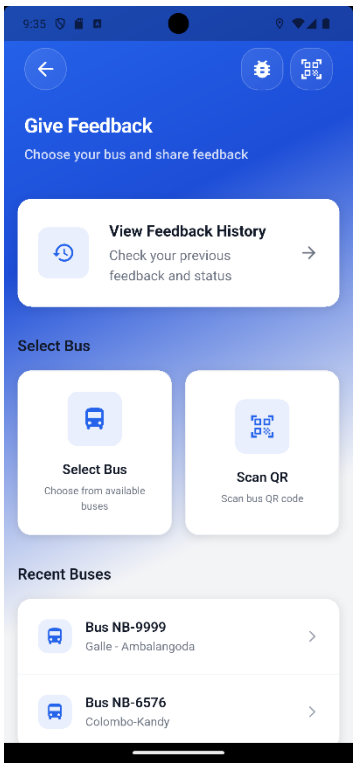

Our Drivers
Search by name, license, route...
Rensith Udara
License: B456789
On Duty
Galle - Colombo
Experience
2
Safety Score
10%
Phone
0777456789
Email
rensithudara@gmail.com
Joined
2025-11-06
Pulindu Srinath
License: B123456
On Duty
Matara - Colombo
NB-9899
Experience
4
Safety Score
100%
Phone
0777123456
Email
pulindu@gmail.com
Joined
2025-11-17
Shaleel Sandeepa
License: B234567
On Duty
Dashboard
Search
Maps
Profile

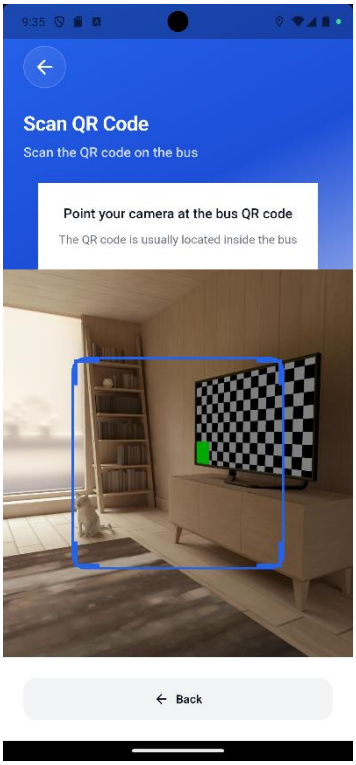

Available Buses
Search by bus number, route, driver...
NB-1231
100%
Pettah - Negombo
Sachin Perera
Colombo Depot
Model
TATA
NB-4564
100%
Kandy – Colombo
Saman Kumara
Colombo Depot
Model
Tata Lp
NB-4567
100%
Pettah - Negombo
Sudesh Priyantha
Colombo Depot
Model
TATA
NB-6576
100%

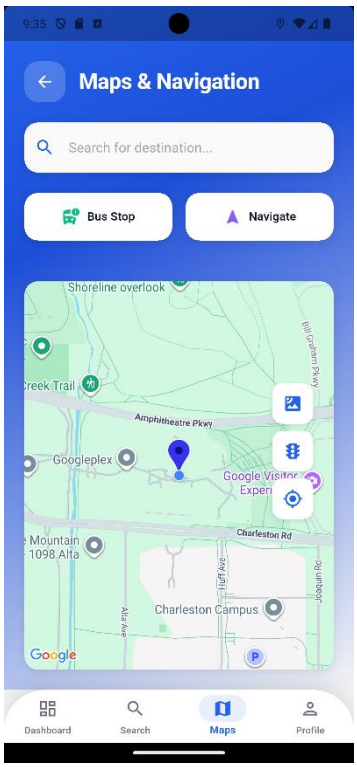

Emergency
Tap any contact below for immediate assistance
Emergency Contacts
Police Emergency
119
Call for immediate police assistance
Ambulance
1990
Medical emergency services
Fire Department
110
Fire and rescue services
SLTB General Hotline.

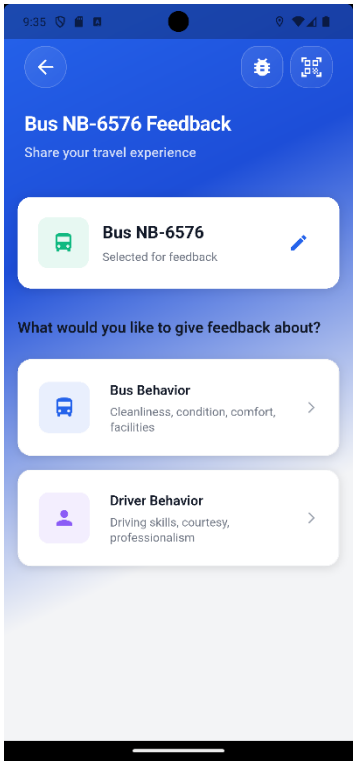

Emergency
Tap any contact below for immediate assistance
Quick Actions
SOS
SOS Alert
Share Location
Report Incident
SOS Contacts
Emergency Safety Tips
Stay calm and assess the situation
Call the appropriate emergency service
Provide clear location information
Follow instructions from authorities
Keep your phone charged for emergencies
Share your location with trusted contacts

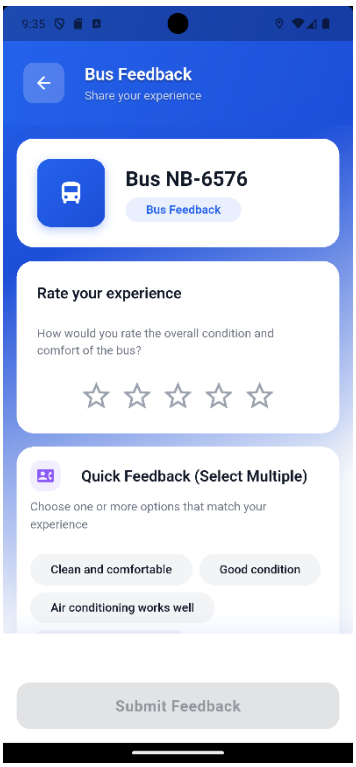

Emergency
Tap any contact below for immediate assistance
Quick Actions
Send SOS Alert?
This will send your location and SOS message to:
Sudath (0761165638)
Cancel
Send SOS
Emergency Safety Tips
Stay calm and assess the situation
Call the appropriate emergency service
Provide clear location information
Follow instructions from authorities
Keep your phone charged for emergencies
Share your location with trusted contacts

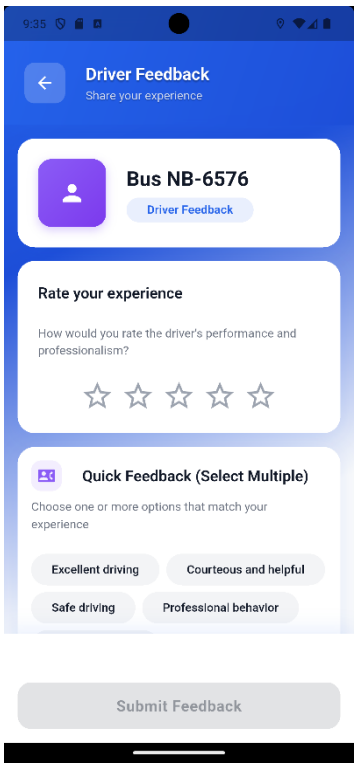

Give Feedback
Choose your bus and share feedback
View Feedback History
Check your previous feedback and status
Select Bus
Select Bus
Choose from available buses
Scan QR
Scan bus QR code
Recent Buses
Bus NB-9999
Galle - Ambalangoda
Bus NB-6576
Colombo-Kandy

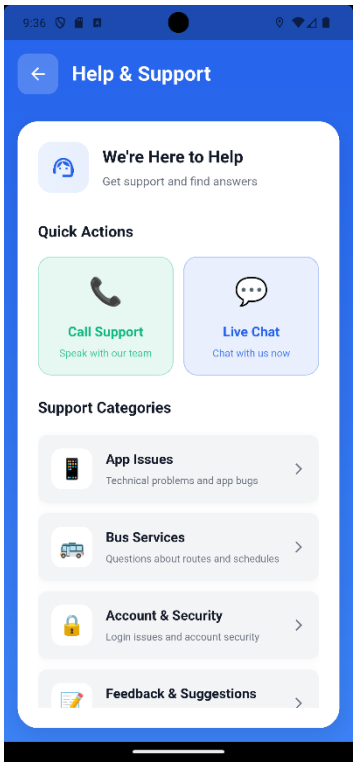

Scan QR Code
Scan the QR code on the bus
Point your camera at the bus QR code
The QR code is usually located inside the bus
Back

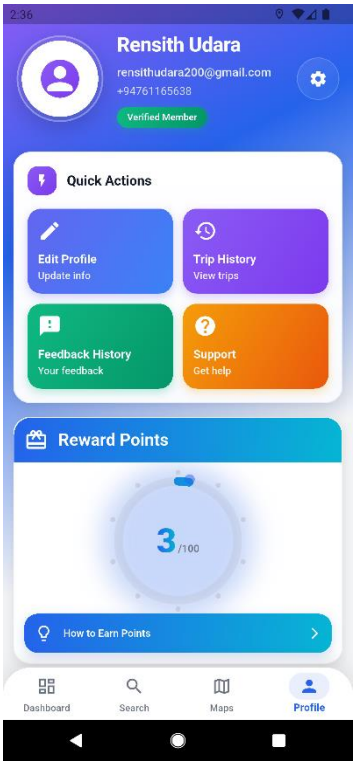

Maps & Navigation
Search for destination...
Bus Stop
Navigate
Shoreline overlook
Amphitheatre Pkwy
Googleplex
Charleston Rd
Charleston Campus
Google
Dashboard
Search
Maps
Profile


Bus NB-6576 Feedback
Share your travel experience
Bus NB-6576
Selected for feedback
What would you like to give feedback about?
Bus Behavior
Cleanliness, condition, comfort, facilities
Driver Behavior
Driving skills, courtesy, professionalism


Bus Feedback
Share your experience
Bus NB-6576
Bus Feedback
Rate your experience
How would you rate the overall condition and comfort of the bus?
Quick Feedback (Select Multiple)
Choose one or more options that match your experience
Clean and comfortable
Good condition
Air conditioning works well
Submit Feedback


Driver Feedback
Share your experience
Bus NB-6576
Driver Feedback
Rate your experience
How would you rate the driver's performance and professionalism?
Quick Feedback (Select Multiple)
Choose one or more options that match your experience
Excellent driving
Courteous and helpful
Safe driving
Professional behavior
Submit Feedback


Help & Support
We're Here to Help
Get support and find answers
Quick Actions
Call Support
Speak with our team
Live Chat
Chat with us now
Support Categories
App Issues
Technical problems and app bugs
Bus Services
Questions about routes and schedules
Account & Security
Login issues and account security
Feedback & Suggestions


Rensith Udara
rensithudara200@gmail.com
+94761165638
Verified Member
Quick Actions
Edit Profile
Update info
Trip History
View trips
Feedback History
Your feedback
Support
Get help
Reward Points
3/100
How to Earn Points
Dashboard
Search
Maps
Profile

Summary of Chapter 6

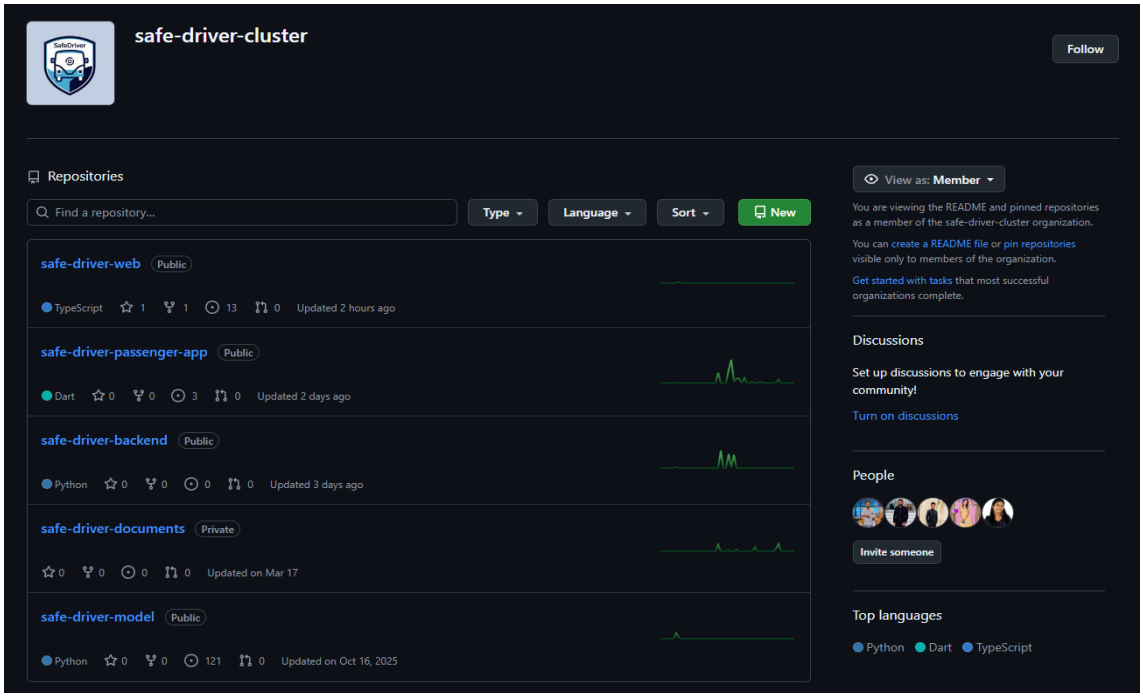
This chapter details the current implementation of the embedded monitoring system, authority dashboard, and passenger mobile app. The Raspberry Pi-based monitoring prototype uses Python, OpenCV, MediaPipe, YOLO and TensorFlow Lite to achieve real-time detection of drowsiness, distraction, yawning, and smoke with high accuracy and low alert latency. The chapter also describes the NextJS-Firebase authority dashboard with real-time KPIs and fleet tools, and the Flutter-based passenger app that provides tracking, safety status, feedback, and emergency features, demonstrating an end-to-end working prototype.

7. Project progress and deliverables

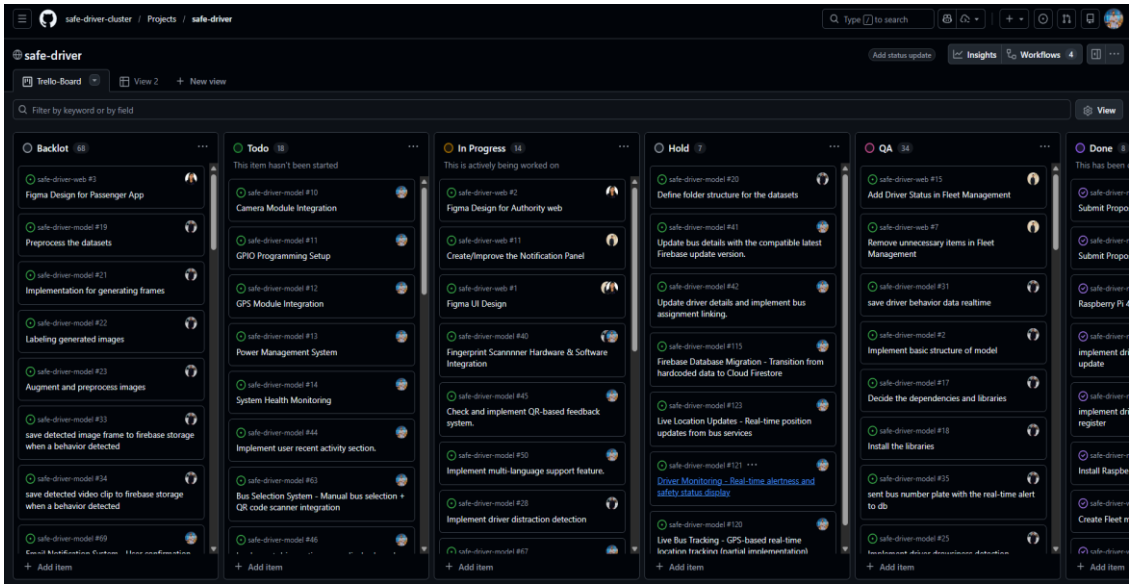
7.1 Development Infrastructure

GitHub Repository: safe-driver-cluster organization with repositories for documents, embedded system, backend, web dashboard, mobile app, and ML models

Link: [safe-driver-cluster repositories]



Project Management: Trello boards for task tracking organized by status (Backlog, In-Progress, Review, Testing, Completed) Link: [https://github.com/orgs/safe-driver-cluster/projects/1/views/1]



- Documentation: Version control with tracked revisions and CMMI-compliant meeting minutes
- CI/CD Pipeline: Initial setup for automated testing and deployment

7.2 Team Collaboration

Regular team meetings conducted via MS Teams and Zoom, with documented agendas and action items. Supervisor consultations scheduled bi-weekly to review designs, validate architectural decisions, and provide feedback. All discussions captured in CMMI-compliant meeting minutes for accountability and traceability.

Summary of Chapter 7

This chapter summarizes design-phase achievements, including a finalized four-tier architecture, complete UML diagrams, and Figma UI/UX designs for the dashboard and mobile app. It documents the established infrastructure such as GitHub repositories, Trello boards, meeting minutes, and initial CI/CD pipelines that support structured development. Overall, the chapter shows that the team has laid a solid technical and organizational foundation for subsequent implementation and testing activities.

8. Test Plan and Test Cases

8.1 Test Objectives and Levels

The test plan aims to verify that SafeDriver meets the functional and non-functional requirements defined in the SRS and PR2 documentation. Testing is organised into unit, integration, system, and pilot/acceptance levels, focusing particularly on safety-critical driver-monitoring and alerting functions.

8.2 Test Environment and Tools

The planned environment reuses the hardware and platforms identified in earlier documents:

- Raspberry Pi 4 with camera, GPS, buzzer and local storage for the embedded monitoring unit.
- Firebase project with Firestore, Cloud Functions, Cloud Messaging, and Authentication.
- Android and iOS devices for mobile apps and modern browsers for the web dashboard.
- GitHub Actions and local runners for automated unit and integration tests on selected components.

8.3 Functional Test Cases

Representative functional test cases include:

- **Driver monitoring tests** to validate EAR-based drowsiness detection, distraction classification by TensorFlow Lite models, and alert escalation timelines.
- **Alert pipeline tests** to ensure that unsafe events create records in Firestore, trigger notifications to the dashboard and, at higher levels, notify authorities.
- **Feedback and points tests** verifying that passenger feedback submissions create feedback documents, upload attachments, and generate the correct point transactions.
- **User management tests** for role-based access to dashboard pages and app features across drivers, passengers, admins, and authorities.

Each test case is documented with ID, description, preconditions, detailed steps, expected results, and traceability to SRS requirement IDs.

8.4 Non-Functional Test Cases

Non-functional tests focus on performance, security, reliability, and usability defined in PR2:

- **Performance tests** simulate multiple buses and passengers to confirm that backend response times and alert latency stay within required thresholds.
- **Security tests** verify TLS usage, authentication flows, role-based access control, and data anonymization for passenger data.
- **Reliability tests** check device recovery after network interruptions and power cycles, validating offline buffering and reconnection logic.
- **Usability tests** evaluate whether drivers, passengers, and administrators can complete critical tasks quickly and without confusion on prototypes.

8.5 Test Plan and Test Case Document

Driver Monitoring System

- Test Case Document
- Test Plan Document

Authority Dashboard

- Test Case Document
- Test Plan Document

Passenger Mobile App

- Test Case Document
- Test Plan Document

Summary of Chapter 8

This section establishes how the team will prove that SafeDriver works as specified. It explains the testing strategy (unit, integration, system, and pilot tests), the target environments (Pi hardware, Firebase backend, apps), and gives representative functional and non-functional test cases tied back to requirements. With this plan and sample tests, the project has a clear, structured approach for validating safety-critical behaviour detection, alerting, performance, security, and usability during the upcoming implementation phase.

9. Conclusion

The work completed during the PR3 period demonstrates that SafeDriver has successfully transitioned from high-level design to detailed, implementation-ready specifications. The database structures and ER relationships ensure that safety-critical data such as alerts, feedback, and driver performance can be captured, linked and analysed reliably across the entire system. The availability of full GUI prototypes for the authority dashboard, and passenger mobile app confirms that user interactions have been thoroughly considered and can now be implemented without major ambiguity.

The defined test plan and initial set of test cases provide a clear framework for verifying functional correctness, performance, security, and usability once implementation progresses. At the same time, continued use of configuration-management procedures and CMMI-compliant meeting documentation shows that the team is managing complexity and change in a controlled manner. In conclusion, the project is on track: the detailed design baseline established in this report gives a solid foundation for the remaining development, integration testing, and pilot evaluation required to deliver a robust, deployable SafeDriver solution.

10. References

[1] International Organization for Standardization. (2013). ISO/IEC 27001: Information security management. Retrieved from https://www.iso.org/standard/54534.html

[2] International Organization for Standardization. (2018). ISO 26262: Road vehicles – Functional safety. Retrieved from https://www.iso.org/standard/68383.html

[3] Institute of Electrical and Electronics Engineers. (2008). IEEE 829: Standard for Software and System Test Documentation. Retrieved from https://standards.ieee.org/standard/829-2008.html

[4] National Transport Commission, Sri Lanka. (n.d.). Road Transport Regulations and Safety Standards. https://www.ntc.gov.lk

[5] Ministry of Transport, Sri Lanka. (n.d.). Public Transport Policy Framework. http://www.transport.gov.lk

[6] Google. (n.d.). TensorFlow Lite Documentation. Retrieved from https://www.tensorflow.org/lite

[7] Google. (n.d.). Firebase Documentation. Retrieved from https://firebase.google.com/docs

[8] Google. (n.d.). MediaPipe Documentation. Retrieved from https://mediapipe.dev

[9] OpenCV. (n.d.). OpenCV Documentation. Retrieved from https://opencv.org

[10] Flutter. (n.d.). Flutter Documentation. Retrieved from https://flutter.dev/docs

[11] Amazon Web Services. (n.d.). AWS CloudFormation Documentation. AWS. https://docs.aws.amazon.com/cloudformation

Initial Discussion to Prepare for Progress 3

Meeting Information			
Meeting Date/Time	04/04/2026 - 19:00:00		
Participants	D. S. Sandeepa R. U. Gonalagoda T. P. Denagamage K. J. L. Perera P. S. Ekanayaka		
Estimated Time	45min	**Actual Time**	45min
Special Notes	Discuss PR3 requirements, identify detailed-design tasks, plan documentation.		
Call/Location Information	MS Teams		
Supported Documents	PR3 rubric, project schedule		

Agenda:

- Get an overall idea about Progress Report 3 and list required artefacts (DB design, GUI prototype, test plan, CMMI minutes).

Notes/Clarifications:

Meeting Minutes:

- Discussed PR3 requirements, identified all detailed-design and documentation tasks, and agreed on responsibilities and internal deadlines.
- Confirmed that database design, GUI prototype, test plan and CMMI minutes will be the main deliverables prepared for this milestone.

Action Items:

Action item	Decision made by
Review PR3 rubric and mark required sections	All participants
Finalize ER diagram and Firebase mapping	R. U. Gonalagoda

Finalize the Abstract

Meeting Information			
Meeting Date/Time	28/04/2026 – 21:00:00		
Participants	D. S. Sandeepa R. U. Gonalagoda T. P. Denagamage K. J. L. Perera P. S. Ekanayaka		
Estimated Time	1h	Actual Time	1h
Special Notes	Final review and polishing of project abstract.		
Call/Location Information	MS Teams		
Supported Documents	Draft abstract, SRS, PR2		

Agenda:

- Review and finalise the abstract for submission and reuse in PR3.

Notes/Clarifications:

Meeting Minutes:

- Reviewed the draft project abstract against SRS and PR2 and refined wording to highlight AI-based monitoring and hybrid edge-cloud architecture.
- Agreed on the final abstract text for faculty submission and reuse in PR3 and future publications.

Action Items:

Action item	Decision made by
Apply final text changes to abstract	R. U. Gonalagoda
Submit approved abstract to faculty portal	D. S. Sandeepa

Progress 3 Discussion with Supervisor

Meeting Information			
Meeting Date/Time	01/05/2026 – 11:00:00		
Participants	Dr. D.D.M. Ranasinghe (Supervisor) D. S. Sandeepa R. U. Gonalagoda K. J. L. Perera P. S. Ekanayaka		
Estimated Time	1h	Actual Time	1h
Special Notes	Review presentation and finalization		
Call/Location Information	Zoom		
Supported Documents	ER diagram, GUI prototypes, draft test plan		

Agenda

- Present detailed database design, GUI prototypes and planned test approach for PR3.

Notes/Clarifications

Meeting Minutes

- Presented updated database design, ER diagram, GUI prototypes and initial test-plan outline to the supervisor and captured feedback.
- Supervisor requested clarifications on relationships and alert visualization; team agreed to adjust diagrams, UI colors and test-case traceability accordingly.

Action Items:

Action item	Decision made by
Update ER diagram and clarify driver–vehicle–device links	R. U. Gonalagoda

Route Mapping Discussion

Meeting Information			
Meeting Date/Time	01/05/2026 – 12:30:00		
Participants	D. S. Sandeepa R. U. Gonalagoda P. S. Ekanayaka T. P. Denagamage K. J. L. Perera		
Estimated Time	1h	Actual Time	1h
Special Notes	Internal follow-up on supervisor feedback		
Call/Location Information	MS Teams		
Supported Documents			

Agenda:

- Review supervisor comments from the Progress 3 meeting.
- Plan changes to documentation, GUI and test plan.

Notes/Clarifications:

Meeting Minutes:

- Discussion on design corrections, dashboard behavior and PR3 documentation tasks.

Action Items:

Action item	Decision made by
Update PR3 database-design	**R.U. Gonalagoda**
Implement route-mapping changes in dashboard prototype	**P.S. Ekanayake**
Compile updated CMMI meeting-minutes appendix for PR3	**K.J.L. Perera**

Route Mapping Discussion

Meeting Information			
Meeting Date/Time	17/06/2025 20:00:00		
Participants	D. S. Sandeepa R. U. Gonalagoda P. S. Ekanayaka		
Estimated Time	1h	**Actual Time**	2h
Special Notes	Finalization of route-mapping features in dashboard.		
Call/Location Information	MS Teams		
Supported Documents			

Agenda:

- Discuss and finalize route mapping and hazard-zone visualization on web dashboard.

Notes/Clarifications:

Meeting Minutes:

- Discussed and finalized the route-mapping design, including how buses, routes and hazard zones will be visualized on the web dashboard.
- Confirmed data fields and responsibilities for implementing the updated route-mapping functionality in the dashboard prototype.

Action Items:

Action item	Decision made by
Implement route-mapping changes in dashboard prototype	**P.S. Ekanayake**