Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform

Chase Tanner: Major in SE
Salman Shekh: Major in SE
Tarig Elamin: Major in SE
Thanh Dat Vu: Major in SE

Course Instructor: Simon Fan
Faculty Advisor: Simon Fan

Industry Sponsor: Qualcomm
Project Mentor: Gautam Fotedar

A capstone project report submitted to the faculty of
Computer Science and Engineering
California State University, San Marcos

October 2025
(Version 1.5)

Technical Report Series: CSU-SM-CSE-39-2026-SE-001-Team-009

Contents

1. Abstract

ALVION is an Android app that spots signs of driver drowsiness and distraction using a phone's front-facing camera and on-device machine learning. The idea is simple: many new cars include driver monitoring, but millions of older vehicles do not. Our goal is to close that gap with an accessible aftermarket option that runs entirely on Snapdragon-based smartphones, without sending any video to the cloud. Running locally keeps latency low for timely alerts, protects privacy by keeping data on the device, and showcases what modern mobile hardware can do in real-time.

The system analyzes live video to track visual cues linked to fatigue and inattention, applies configurable thresholds, and triggers clear alerts through sound, screen prompts, and vibration when attention appears to drift. The design focuses on what we can build and test in a years time: reliable on-device inference, straightforward configuration, and a responsive alert pipeline. We do not attempt vehicle integration, cloud processing, large-scale deployment, or regulatory certification.

This document lays out the project's objectives, assumptions, and boundaries so developers, mentors, and faculty can stay aligned as we move from concept to implementation. It explains the system architecture, component responsibilities, and the choices behind them, aiming for a design that is practical to build now and easy for a future team to extend. In short, ALVION demonstrates that a phone you already carry can provide meaningful safety signals when designed with careful attention to latency, privacy, and usability.

2. Document Information

2.1 Report Revision History

Version	Date	Description	Author
1.0	Oct 13, 2024	Initial draft submitted. Report outline aligned to capstone guideline; 200–300 word abstract finalized; Introduction completed (Purpose, Background, Scope, Intended Audience, Project Description). Scope and constraints defined for on-device Snapdragon prototype. Document metasetup added (revision history and approval tables).	Team 9
1.5	Oct 31, 2024		Team 9
2.0	Dec 1, 2024		Team 9

3. Problem Statement

3.1 Business Background

Driver monitoring is becoming a regulatory and industry standard for road safety. The European Union now mandates such systems in new vehicles beginning in 2024, and other markets are expected to follow. Automakers and regulators view these systems as essential for reducing human-error-related crashes, but adoption in existing vehicles remains limited because of cost, integration complexity, and consumer concerns over data privacy. Retrofit products, while available, often require expensive hardware installations or continuous cloud connectivity for processing. These barriers restrict adoption, especially in regions where consumers rely on older cars or are reluctant to share biometric data online. A smartphone-based solution reduces those barriers: most users already own compatible Android devices with capable cameras and neural-processing hardware. ALVION's approach uses these resources to demonstrate how Qualcomm's mobile AI platforms can perform effective, real-time driver-monitoring inference entirely on-device. This provides a testbed for cost-effective safety innovations and shows potential business alignment with regulatory trends. The project also establishes a baseline for future Qualcomm-enabled automotive safety research and gives developers a model for edge-based AI deployment that respects user privacy and reduces infrastructure costs.

3.2 Needs

Improving driver safety requires early detection of distraction and drowsiness in a manner that is both reliable and user-friendly. ALVION responds to several intertwined needs:

- **Safety:** The system should protect both the driver and other road users by recognizing behavioral cues such as prolonged eyelid closure, yawning, or head tilting that precede dangerous inattention.
- **Privacy:** Consumers are wary of surveillance systems that transmit facial data to the cloud. An entirely on-device approach ensures that personal information remains secure while still providing effective monitoring.
- **Accessibility and Cost:** Full-scale automotive integrations are expensive. By relying on common smartphone hardware, ALVION provides an affordable entry point for drivers worldwide.
- **Responsiveness:** Alerts must reach the driver when they matter most. Real-time analysis ensures immediate feedback that can prevent accidents instead of only documenting them.
- **Reusability and Scalability:** The project serves as an open reference implementation that can be adapted, expanded, and optimized by future teams, supporting continuous improvement in mobile AI safety systems.

3.3 Objectives

ALVION's objectives focus on producing a working, reproducible prototype that demonstrates the technical and practical viability of phone-based driver monitoring:

- **Develop a proof-of-concept Android application** capable of detecting drowsiness and distraction using the front camera and a lightweight on-device model optimized for Snapdragon NPUs.
- **Implement configurable alert mechanisms** that combine audio, visual, and haptic feedback, allowing users to tune thresholds and notification styles for comfort and responsiveness.
- **Achieve efficient real-time performance** without dependence on cloud computing, ensuring low latency and minimal power usage consistent with mobile operation.
- **Document all design decisions, architecture, and testing methodology** in a reproducible form so that future development teams can extend or refine the system.
- **Conduct iterative testing and evaluation** under varying lighting, motion, and device conditions to measure accuracy, usability, and overall system robustness.

Through these objectives, ALVION demonstrates a concrete pathway from concept to functional prototype, aligning technical feasibility with ethical design and industry safety goals.

4. Requirements

4.1 User Requirement

4.1.1 Glossary of Relevant Domain Terminology

ALVION The Android-based driver drowsiness and distraction detection mobile application developed for this capstone project. Runs entirely on-device to ensure privacy and real-time performance.

Driver Monitoring System (DMS) A safety technology that uses sensors or cameras to assess driver attention, alertness, and potential fatigue while operating a vehicle.

PERCLOS (Percentage of Eyelid Closure) A standard metric used in drowsiness detection. It represents the proportion of time a driver's eyes are at least 80% closed over a given time window.

Head Yaw The horizontal rotation of a driver's head relative to the camera. Excessive yaw over time indicates potential distraction or inattention.

Android Studio The official integrated development environment (IDE) for Android applications. Used to develop and build the ALVION mobile app using Kotlin.

Kotlin A modern programming language fully supported by Android, known for concise syntax and null safety. Used as the primary implementation language for ALVION.

Jetpack Compose Android's declarative UI framework written in Kotlin. It simplifies the creation of reactive, adaptive user interfaces without the need for XML layout files.

CameraX A Jetpack library that provides easy access to camera functions such as preview, capture, and frame analysis. Used in ALVION to stream frames for real-time inference.

TensorFlow Lite (TFLite) A lightweight, cross-platform runtime for executing trained machine learning models on mobile and embedded devices. Handles on-device inference in ALVION.

NNAPI (Neural Networks API) Android's standard interface that allows apps to access on-device accelerators such as GPUs, DSPs, or NPUs for running ML workloads efficiently.

Qualcomm Neural Network (QNN) Runtime Qualcomm's proprietary runtime for accelerating AI models on Snapdragon's Hexagon DSP/HTP cores. Used optionally for performance optimization.

ML Kit Google's on-device machine learning SDK that provides ready-to-use APIs for vision tasks such as face detection, eye openness estimation, and head pose detection.

MediaPipe An open-source framework for building cross-platform ML pipelines, including facial landmark detection. Used as an alternative to ML Kit for greater control over signals.

OpenCV Open-source computer vision library providing image-processing tools such as resizing, cropping, and overlay drawing. Used optionally for debugging or preprocessing.

PyTorch A machine learning framework used for training models off-device. In ALVION, it can be used to train lightweight classifiers before conversion to TFLite.

ONNX (Open Neural Network Exchange) A standardized model format that enables conversion and interoperability between ML frameworks like PyTorch and TensorFlow.

Quantization A model optimization technique that reduces numerical precision (e.g., from 32-bit floats to 8-bit integers) to improve speed and reduce memory use during inference.

Debounce Logic A software mechanism that prevents repeated triggers of an alert within a short time window, reducing false positives in drowsiness detection.

Sliding Window Aggregation A temporal data analysis technique that aggregates frame-level results (e.g., eye openness) over a fixed time window to detect sustained patterns such as drowsiness.

BYOM (Bring Your Own Model) A Qualcomm workflow allowing developers to integrate custom-trained ML models into Snapdragon devices using the QNN SDK.

MVP (Minimum Viable Product) The simplest version of the app that demonstrates key functionality (on-device detection, alerts, and adjustable thresholds) without full feature expansion.

Latency The time delay between camera frame capture and alert output. Low latency is critical to ensure timely driver warnings.

FPS (Frames Per Second) A measure of how many frames are processed per second during video analysis. Maintaining $\geq$15 FPS ensures smooth real-time inference.

On-device Inference The process of running AI models locally on the phone rather than sending data to cloud servers. Essential for preserving privacy and minimizing delay.

Privacy by Design A principle ensuring that user data (especially facial video) remains on the device and is not transmitted externally without explicit consent.

Thermal Throttling Automatic performance reduction that occurs when a device's temperature rises beyond safe limits. ALVION monitors this to adjust inference rate dynamically.

Drowsiness Alert Cooldown A configurable delay after an alert during which no new alerts are triggered, preventing repetitive warnings and user annoyance.

Subject-wise Split A data handling practice in ML ensuring that images or videos of the same person appear only in either training or testing sets, avoiding data leakage.

Dataset Licensing (Research Use Only) Legal restrictions defining how public datasets (e.g., CEW, MRL, NTHU-DDD) can be used. ALVION uses only datasets with academic/research permissions.

4.1.2 User Groups

This section describes the primary user groups who will interact with the Distracted/Drowsy Driving Detection System. Each group represents a distinct role with unique goals and responsibilities related to the system's functionality and operation.

- Drivers: Main end users; rely on alerts
- Fleet Managers / Supervisors: oversee multiple drivers, review aggregated logs
- System Administrators: Maintain the deployment, handle privacy configs and updates
- Developers / Maintenance Engineers (Internal): internal users who maintain and enhance the app

4.1.3 Project Context

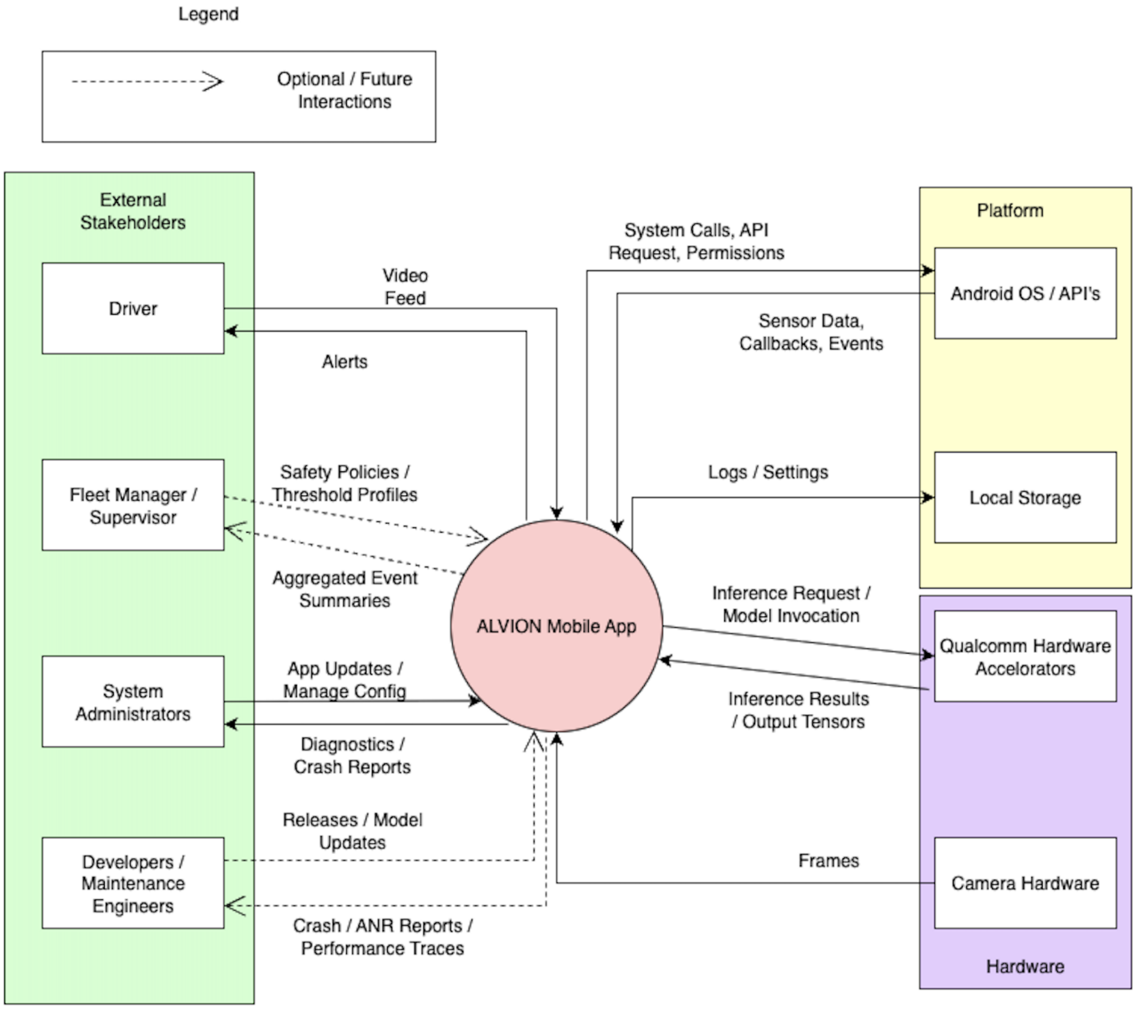

Legend
Optional / Future Interactions
External Stakeholders
Driver
Fleet Manager / Supervisor
System Administrators
Developers / Maintenance Engineers
Video Feed
Alerts
Safety Policies / Threshold Profiles
Aggregated Event Summaries
App Updates / Manage Config
Diagnostics / Crash Reports
Releases / Model Updates
Crash / ANR Reports / Performance Traces
ALVION Mobile App
System Calls, API Request, Permissions
Sensor Data, Callbacks, Events
Logs / Settings
Inference Request / Model Invocation
Inference Results / Output Tensors
Frames
Platform
Android OS / API's
Local Storage
Qualcomm Hardware Accelorators
Camera Hardware
Hardware

Figure 4.1: System context Diagram for the AlVION driver drowsiness and distraction detection mobile application

The context diagram illustrates how the ALVION mobile application interacts with its surrounding environment. The driver provides a live video feed through the smartphone camera and receives real-time alerts through audio, vibration, or visual cues. The application communicates with the Android OS using standard APIs for camera access, vibration control, and storage management. It performs on-device inference via the TensorFlow Lite runtime accelerated through Qualcomm's QNN or Android's NNAPI interface. Optional connections—such as aggregated summaries for fleet managers, managed configurations for system administrators, or diagnostic feedback for developers—represent potential future extensions.

4.1.4 Functional Requirements

4.1.4.1 Project Scope

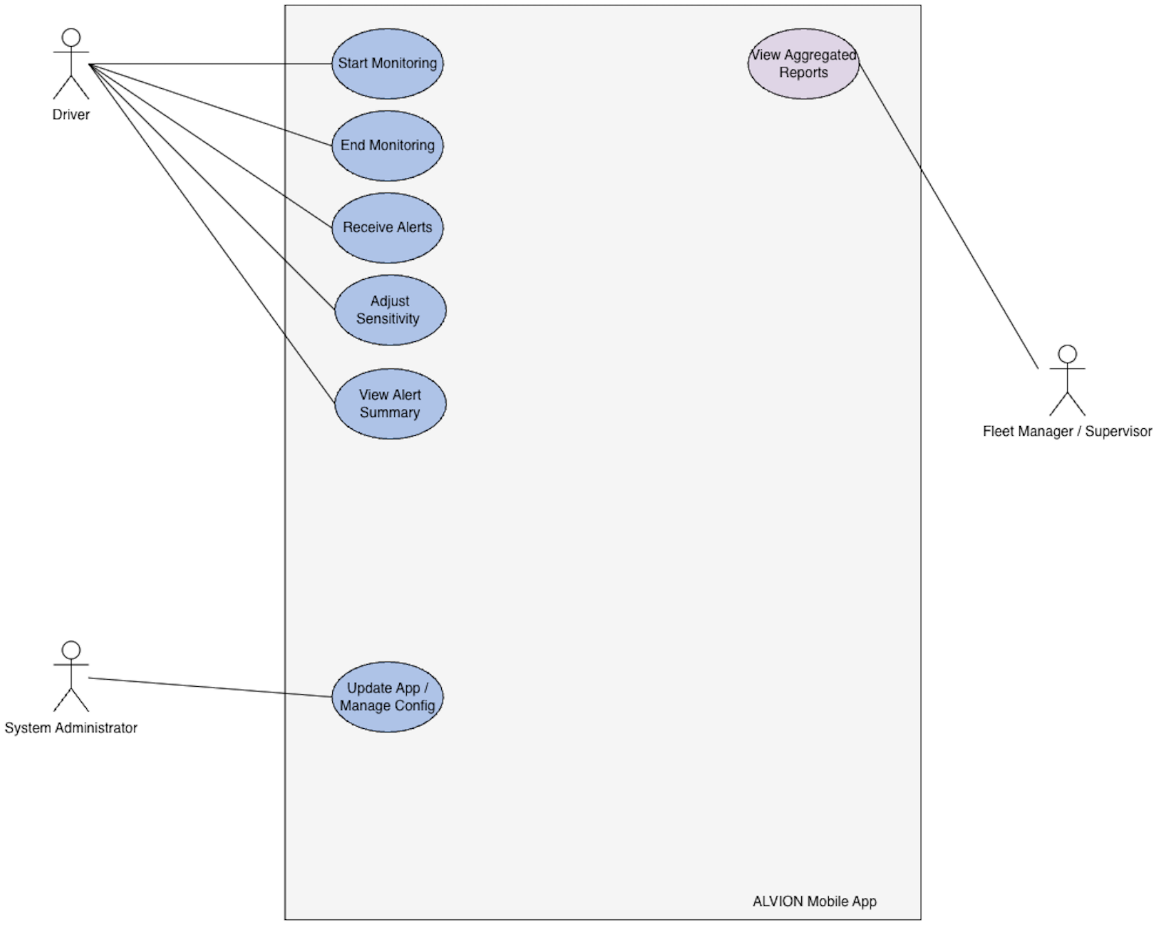


Figure 4.2: Scope diagram to illustrate major user interactions with ALVION

4.1.4.2 User Scenarios

The following scenarios describe how different actors interact with the ALVION mobile application. Each scenario corresponds to a use case identified in the Project Scope diagram and summarized in Appendix U.

- **Start Monitoring Session:** The driver launches the ALVION application, grants camera permissions, and begins a monitoring session. The app activates the front-facing camera through CameraX and initializes on-device ML inference to begin analyzing the driver's eye openness and head orientation in real time.

- **Receive Drowsiness Alert:** During monitoring, ALVION continuously evaluates the Percentage of Eyelid Closure (PERCLOS). If the eyes remain closed beyond the configured duration threshold, the system issues an audible and visual alert to prompt the driver to regain focus.
- **Receive Distraction Alert:** The application monitors head yaw angle using the face-detection model. If the driver's head remains turned away from the forward direction for longer than the threshold period, a distraction alert is triggered.
- **Acknowledge or Snooze Alert (Extend):** After receiving a drowsiness or distraction alert, the driver can choose to acknowledge it or temporarily snooze further alerts for a short cooldown period to prevent repetitive warnings.
- **Adjust Sensitivity and Settings:** The driver opens the Settings screen to adjust thresholds for eye-closure time, yaw angle, or alert cooldown. These preferences are stored locally and persist between sessions.
- **Calibrate / Align Camera (Include):** On first use or when lighting changes, the driver may be prompted to align the camera for optimal face tracking. The app displays live feedback until proper alignment is achieved.
- **View Alert Summary (Optional):** After a session ends, the driver can view a local summary showing the number and type of alerts triggered. This information helps evaluate driving habits or fatigue patterns.
- **Export Logs (Optional):** Users can export anonymized event logs to share with researchers or safety supervisors. This function remains offline and requires explicit user consent.
- **Manage App Deployment (System Administrator):** The system administrator deploys updates, manages app permissions, and enforces organizational settings through Android device management tools.
- **View Aggregated Reports (Fleet Manager):** In a potential future version, fleet managers could access a dashboard summarizing aggregated alert data across multiple drivers to assess safety trends and rest policies.

4.1.4.3 User Functional Requirements

This section provides short descriptions of the user functional requirements identified for the ALVION mobile application. Each requirement corresponds to the detailed tables presented in Appendix R and defines a specific capability the system shall provide to meet user and stakeholder goals.

1. **UF-A: Account Management** — The system shall allow a driver to create an account and securely log in to view personal driving history or session data.
2. **UF-B: Drowsiness Detection and Alerting** — The system shall continuously analyze eye openness using the PERCLOS metric and issue a warning when sustained eyelid closure indicates drowsiness.
3. **UF-C: Distraction Detection and Alerting** — The system shall detect when the driver's head or gaze deviates from the forward direction for longer than a defined interval and trigger an attention-restoring alert.
4. **UF-D: Configurable Sensitivity and Alert Settings** — The system shall provide a settings interface that allows the driver to adjust detection sensitivity, alert volume, and cooldown timing.
5. **UF-E: Start and Stop Monitoring Session** — The driver shall be able to manually start

and end a monitoring session, activating or releasing the camera and inference modules as needed.

6. **UF-F: Real-Time Alert Delivery** — The system shall deliver real-time visual, audio, or vibration alerts when drowsiness or distraction is detected so that the driver can immediately respond.
7. **UF-G: Acknowledge or Snooze Alerts** — The system shall provide options for the driver to acknowledge or temporarily snooze alerts, reducing repeated notifications during recovery periods.
8. **UF-H: Camera Calibration and Alignment** — The system shall provide a calibration routine to ensure the driver's face is properly framed and visible before or during monitoring to improve detection accuracy.
9. **UF-I: Session Summary and Export (Optional)** — The system shall display a local summary of each session's alerts and optionally export anonymized logs for driver review or research use.
10. **UF-J: Privacy and Local Data Control** — The system shall store all captured data locally on the device and prevent external transmission unless the user explicitly consents.
11. **UF-K: Administrative Deployment and Configuration** — Authorized system administrators shall be able to deploy application updates and manage configuration policies through Android enterprise tools.
12. **UF-L: Aggregated Fleet Reporting** — In future releases, the system may provide a fleet-manager dashboard that aggregates driver alert statistics to support organizational safety analysis.

4.1.5 Non-Functional Requirements

This section summarizes the non-functional requirements defined for the ALVION driver drowsiness and distraction detection application. Each group below corresponds to the detailed requirement tables in Appendix R and captures essential quality, organizational, and external constraints that guide the project.

4.1.5.1 Product: Usability Requirements

- **UP-01: Simple User Interface** – The application shall provide an intuitive and minimal interface that allows users to begin monitoring with minimal interaction.
- **UP-02: Hands-Free Operation** – During active monitoring, no user input shall be required; all alerts and responses shall occur automatically through audio or vibration feedback.
- **UP-03: Clear Feedback and Alerts** – Visual, auditory, and haptic feedback shall be easily distinguishable and clearly convey the driver's attention status.
- **UP-04: Accessibility Support** – Interface text, color, and contrast shall follow accessibility guidelines to remain legible under varied lighting conditions.

4.1.5.2 Product: Performance Requirements

- **UP-05: Frame Rate and Latency** – The system shall process at least 15 frames per second and maintain an alert latency below 200 ms.
- **UP-06: On-Device Inference Efficiency** – All model inference shall execute locally on Snapdragon CPU, GPU, or NPU with average CPU utilization below 40%.

- **UP-07: Thermal and Battery Management** – The system shall reduce inference frequency if device temperature or power consumption exceeds defined limits.

4.1.5.3 Product: Availability, Reliability, and Security Requirements

- **UP-08: Operational Availability** – The application shall run continuously for up to two hours without crash or data loss.
- **UP-09: Fail-Safe Behavior** – In the event of hardware or model failure, the system shall suspend monitoring and notify the user instead of issuing false alerts.
- **UP-10: Data Privacy and Security** – All logs and image data shall remain on-device in app-scoped storage; no external transmission shall occur without explicit consent.
- **UP-11: Permission Control** – The app shall request only essential Android permissions and respect user revocation at runtime.

4.1.5.4 Organizational: Development Requirements

- **UO-01: Development Environment** – The project shall be developed in Android Studio using Kotlin and tested on Snapdragon hardware to ensure compatibility.
- **UO-02: Version Control** – All source code shall be managed through a Git repository with standardized branching and commit practices.
- **UO-03: Coding Conventions** – Code shall follow Kotlin style guidelines and include inline documentation for maintainability.
- **UO-04: Testing Coverage** – Each functional requirement shall be verified through at least one unit or integration test prior to release.

4.1.5.5 Organizational: Operational Requirements

- **UO-05: Deployment Compatibility** – The system shall be distributed as an Android APK targeting API Level 33 or higher and installable without root access.
- **UO-06: User Support Documentation** – The project shall provide a concise user guide or in-app help section describing calibration, settings, and safety disclaimers.

4.1.5.6 Organizational: Environmental Requirements

- **UO-07: Lighting Adaptability** – The app shall function accurately under daylight, artificial light, and low-light cabin conditions.
- **UO-08: Device Orientation** – The application shall operate correctly when the device is mounted in either landscape or portrait orientation.
- **UO-09: Offline Functionality** – All monitoring and alert functions shall operate fully without network connectivity.

4.1.5.7 External: Legislative Requirements on Safety and Security

- **UE-01: Driver Safety Compliance** – The application shall include a startup safety disclaimer and disable manual interaction while the vehicle is in motion.
- **UE-02: Data Protection Compliance** – The system shall comply with applicable privacy laws such as CCPA and GDPR regarding consent and data retention.

4.1.5.8 External: Cultural and Social Requirements

- **UE-03: Inclusive Model Performance** – The detection model shall be evaluated for consistent accuracy across diverse skin tones, facial structures, and eyewear.
- **UE-04: Language Localization** – The user interface shall default to English and be structured for easy translation into additional languages.

4.1.5.9 External: Interoperability Requirements

- **UE-05: Android API Compatibility** – The application shall utilize stable Android Jetpack APIs such as CameraX, AudioManager, and Storage to ensure broad device support.
- **UE-06: Hardware Acceleration Interfaces** – The application shall remain interoperable with both Qualcomm QNN and Android NNAPI runtimes for accelerated on-device inference.

4.2 System Requirements

4.2.1 Functional Requirements

4.2.1.1 System Functional Requirements

The ALVION system shall implement the following core functional features to enable real-time driver monitoring and alerting:

- **SF-01: Real-Time Drowsiness Detection** — The system shall continuously analyze the driver's eye openness using the Percentage of Eyelid Closure (PERCLOS) metric to determine signs of drowsiness. When the eyes remain closed beyond a configurable threshold, a drowsiness alert shall be triggered.
- **SF-02: Distraction Detection** — The system shall track the driver's head yaw (horizontal rotation) and issue a distraction alert when the driver looks away from the forward direction for more than a defined dwell time.
- **SF-03: Alert Mechanisms** — The system shall deliver real-time alerts using audio tones, vibration feedback, and visual overlay cues to ensure the driver's attention is regained promptly.
- **SF-04: Configurable Sensitivity and Thresholds** — The system shall allow users to adjust detection sensitivity, eyelid-closure duration, yaw angle threshold, and alert cooldown periods via the Settings interface.
- **SF-05: Camera Calibration and Alignment** — On initial launch or when lighting conditions change, the system shall prompt the user to align the front-facing camera for optimal face tracking and detection accuracy.
- **SF-06: Session Control** — Users shall be able to start and stop monitoring sessions manually, enabling or releasing the camera and inference modules as needed.
- **SF-07: Snooze and Acknowledge Alerts** — Users shall be able to acknowledge or temporarily snooze alerts to prevent repeated notifications during rest or recovery.
- **SF-08: Session Summary and Reporting** — After each session, the app shall display a local summary including the number and type of alerts triggered, session duration, and driver responsiveness metrics.
- **SF-09: Privacy and Data Protection** — All facial analysis and inference shall occur locally on the device. No video or image data shall be transmitted externally without explicit user consent.
- **SF-10: Administrative Configuration** — System administrators shall be able to configure and deploy the app across managed Android devices using enterprise deployment tools and preset configurations.

4.2.1.2 Data Requirements

The ALVION application processes and manages several categories of data required for on-device drowsiness and distraction detection. These data requirements describe what information is captured, derived, stored, and protected within the system.

Data Inputs

- **Camera Frames:** RGB frames captured from the front-facing camera (resolution up to 1920×1080, $\geq$15 FPS) using the CameraX API.

- **Sensor Metadata:** Device orientation and timestamp values from Android SensorManager for head-pose compensation.

Data Processing and Outputs

- **Processed Metrics:** Per-frame eye-openness ratio (PERCLOS), head yaw/pitch angles, and attention state labels (alert / drowsy / distracted).
- **Alerts:** Boolean and string messages delivered to the UI and audio subsystem indicating driver status or safety warnings.

Data Storage Requirements

- **Local Storage:** Session summaries and calibration data stored in JSON or SQLite format under app-scoped storage. No cloud database is used.
- **Retention:** Logs persist until manually deleted by the user. Temporary inference buffers are discarded after each session.

Data Relationships Session data link driver ID, timestamps, and alert metrics to a single session record:

$$Session(session\_id, timestamp, perclos\_avg, alert\_count, duration)$$

Processed data are read-only to other modules; only summary statistics are exposed to the UI.

Data Quality and Integrity

- All timestamps follow ISO 8601 format.
- Invalid or missing sensor frames are ignored, not extrapolated.
- Feature values are clamped to physical limits ($0 \leq \text{PERCLOS} \leq 1$).

Data Privacy and Security

- All data remain on-device in encrypted storage (UP-10, UE-02).
- No identifiable images or raw video are transmitted externally.
- Inference executes locally using Qualcomm QNN / Android NNAPI.

These data specifications define the system's information model and handling rules. They are not user functional requirements but underpin compliance with non-functional constraints on privacy, performance, and reliability.

4.2.2 Non-functional Requirements

4.2.2.1 Product: Usability Requirements

- **SP-01-01** — The application shall provide a simple, intuitive interface that allows users to start monitoring with minimal setup.
- **SP-02-01** — The system shall require no manual input during active monitoring, ensuring fully hands-free operation.
- **SP-03-01** — Alerts shall be clearly distinguishable through visual, auditory, or haptic feedback, synchronized within ±50 ms.
- **SP-04-01** — The interface shall maintain readability and accessibility under different lighting conditions and conform to WCAG 2.1 AA.

4.2.2.2 Product: Performance Requirements

- **SP-05-01** — The system shall process at least 15 frames per second and issue alerts within 200 ms of detection.
- **SP-05-02** — Average CPU utilization during active monitoring shall remain below 40%, and the application shall adapt processing rates to manage battery and thermal constraints.

4.2.2.3 Product: Availability/Reliability/Security

- **SP-08-01** — The application shall remain operational for up to two hours without crash or data loss.
- **SP-09-01** — If the camera or ML inference fails, the system shall pause monitoring and notify the user.
- **SP-10-01** — All logs and configuration files shall be stored in secure app-scoped encrypted storage and deletable on request.
- **SP-11-01** — The application shall respect Android runtime permission revocations and immediately disable affected capabilities.

4.2.2.4 Organizational: Development Requirements

- **SO-01-01** — The project shall be implemented using Android Studio (Kotlin) with Git version control.
- **SO-04-01** — Continuous-integration builds shall enforce unit-test coverage $\geq 80\%$.

4.2.2.5 Organizational: Operational Requirements

- **SO-05-01** — The system shall be distributed as an APK targeting Android API Level 33 or higher.
- **(from UO-06)** — The application shall include an in-app guide explaining setup, calibration, and safety disclaimers.

4.2.2.6 Organizational: Environmental Requirements

- **SO-07-01/02** — The system shall function reliably under daylight, artificial, and low-light conditions.
- **SO-08-01** — The application shall operate correctly in both portrait and landscape orientations.
- **SO-09-01** — All detection and alerting functionality shall remain available without network access.

4.2.2.7 External: Legislative Requirements on Safety/Security

- **SE-01-01** — The application shall display a startup safety disclaimer and disable interactive prompts while the vehicle is moving.
- **(from UE-02)** — Stored data shall comply with GDPR/CCPA consent, deletion, and retention rules.

4.2.2.8 External: Cultural and Social Requirements

- **SE-03-01** — The detection model shall be validated for consistent performance across diverse skin tones, facial features, and eyewear.
- **SE-04-01** — The user interface shall default to English and be structured for easy localization.

4.2.2.9 External: Interoperability Requirements

- **SE-05-01** — The system shall use stable Android Jetpack APIs such as CameraX, AudioManager, and Storage for broad compatibility.
- **SE-06-01** — The ML inference pipeline shall interoperate with both Qualcomm QNN and Android NNAPI runtimes for accelerated on-device inference.

5. Exploratory Studies

5.1 Relevant Development Frameworks

We are building a phone-first prototype that runs entirely on the device. Each tool below has one clear job. We avoid extra moving parts so we can ship fast and keep the app stable on a Galaxy phone.

5.1.1 Android Studio (Kotlin)

What it is. Android Studio is the IDE; **Kotlin** is the language for our app. **Why we use it.** It's the standard way to build high-quality Android apps and gives us first-class access to the camera, sensors, and hardware acceleration.

5.1.2 Front end: Jetpack Compose (native UI)

What it is. Compose is Kotlin's modern UI toolkit (native, declarative), similar in feel to React, but it compiles to Android views directly. **Why we use it.** We can design clean, responsive screens (camera preview overlay, status badges, settings) without XML. Compose + Material3 gives us polished components out of the box. **UI libs we rely on (lightweight):** *material3* (design system), *accompanist-permissions* (runtime camera permission), *coil* (optional image loading for debug).

5.1.3 Camera: CameraX (preview + frames for ML)

What it is. A high-level camera library on top of Camera2. **Why we use it.** It gives us (1) a live preview for the UI and (2) an "analysis" stream of frames for ML—without wrestling with low-level sessions. It also handles rotation and lifecycle, and lets us "keep only latest" frames so the app stays smooth.

5.1.4 On-device inference: TensorFlow Lite (TFLite)

What it is. The small runtime that *runs* a model file on the phone. **Where it runs.** On the Galaxy device, inside our Kotlin app. **Why we use it.** TFLite is portable and works with Android's **NNAPI** so the phone can use hardware accelerators when available and CPU when not. **Optional optimization.** On Snapdragon devices that expose the Hexagon/HTP path, we can enable **Qualcomm QNN** as a feature flag for extra speed/power savings. If QNN isn't available, the app still runs via TFLite+NNAPI.

5.1.5 Vision helpers: ML Kit / MediaPipe / OpenCV

ML Kit Face Detection. Fast path: gives eye-open probabilities and head yaw directly—no training needed. **MediaPipe Face Landmarker.** Alternative that returns facial landmarks so we can compute eye openness (EAR) and yaw ourselves. **OpenCV (optional).** A toolbox for basic image tasks (resize, crop, color convert, draw debug overlays). We add it only if we need it, behind a small Kotlin interface, so the rest of the app stays the same.

5.1.6 Training (off-device, only if needed): PyTorch → ONNX → TFLite

What it is. PyTorch is used on a laptop/Colab to *train* a tiny eye-state model if field tests show we need one. **Where it runs.** Off-device. The phone does *not* train. **Why we use it.** PyTorch is quick to iterate. We export PyTorch → ONNX → **TFLite (INT8)** so the Android app can run the model with TFLite+NNAPI/QNN. If post-training quantization reduces accuracy too much, we try light QAT; otherwise we keep PTQ for simplicity.

5.1.7 Project glue, performance, and stability

Gradle builds the app. **Kotlin coroutines/Flow** keep ML off the UI thread. The analyzer uses "*keep only latest*" so we never build a big queue—this bounds latency and avoids stutter. We monitor timing and thermals with **Android Profiler** and **Perfetto**; if the phone warms up, we downshift internal inference rate (e.g., 30 → 24 FPS) while keeping the UI responsive.

Tool roles at a glance

Tool	Role	Why it's in the stack
Android Studio + Kotlin	App development	Official Android toolchain; fastest path to a stable, native app
Jetpack Compose (Material3)	Front-end UI	Native, declarative UI; polished components; easy to iterate
CameraX	Camera preview + frame analysis	Fewer device quirks; easy lifecycle; analysis frames for ML, backpressure control
TFLite (+NNAPI)	On-device inference	Portable runtime; uses accelerators when available; sane CPU fallback
QNN (optional)	Snapdragon acceleration	Feature flag for extra speed/power on supported devices
ML Kit / MediaPipe	Ready facial signals	Eye-open probability / landmarks without training; jump-start MVP
OpenCV (optional)	Image utilities	Only if we need extra preprocessing or debug overlays
PyTorch (off-device)	Training (if needed)	Fast to prototype a tiny eye-state model; exports to TFLite for the app

5.2 Relevant Solution Techniques & Dataset Strategy

Our objective is a reliable, real-time demo on a Snapdragon phone. We will *not* train from scratch. Instead, we use a tiered approach that prioritizes fast geometric signals on-device and selectively leverages an existing MobileNetV2 model trained on a large Kaggle dataset.

5.2.1 Tiered Runtime (device-first)

Tier 1 (default, no training). Face landmarks on device (e.g., ML Kit or MediaPipe) to estimate:

- **Eye openness:** framewise openness per eye; PERCLOS over a sliding window for drowsiness.
- **Head yaw (left/right):** hysteresis + timeout to flag distraction rather than micro head jitters.
- **Alert logic:** debounce and cooldown to avoid spam; parameters tuned by short in-app tests.

This tier runs continuously at ≈24–30 FPS and is our primary signal path.

Tier 2 (assist, only when needed). A compact classifier (MobileNetV2) pre-trained via a Kaggle pipeline on ∼40k images. We call it only when Tier 1 is uncertain (e.g., low landmark confidence, glasses, low light). The model operates on a face/eye ROI and returns an *eye state* or *drowsy/alert* vote that is fused with Tier 1 signals.

Component	**Signal/Role**	**Why it fits our goal**
Tier 1: Landmarks + PERCLOS	Eye openness, head yaw	High FPS, low battery, explainable thresholds.
Tier 2: MobileNetV2 (Kaggle)	Eye/drowsiness vote on ROI	Robustness under glasses/low light; called sparingly.
Fusion & Alerts	Debounce/hysteresis	Stable user experience; fewer false alarms.

5.2.2 MobilenetV2 Baseline: Adopt → Audit → Adapt

We leverage an existing Kaggle MobileNetV2 training pipeline and dataset to save time. Our process ensures we do not ship a brittle model.

Adopt (establish a baseline).

(a) Reproduce the MobileNetV2 notebook end-to-end; save checkpoints and confusion matrices.
(b) Add a simple inference script that accepts face/eye crops (to match phone ROI).

Audit (trust but verify).

(a) **Labels & classes:** confirm the class semantics (e.g., open/closed vs. drowsy/alert) align with PERCLOS.
(b) **Subject-wise splits:** ensure no identity leakage between train/val/test.
(c) **Duplicates/balance:** scan for near-duplicates; rebalance if needed.
(d) **Viewpoint gap:** note differences between webcam/selfie data and in-car angles.
(e) **License/terms:** confirm research use; no redistribution of face images outside the team.

Adapt (make it phone-ready).

(a) Convert Keras/ONNX → **TFLite (INT8)**; run with **NNAPI** and optionally **QNN** on Snapdragon.

(b) Gate Tier 2 by Tier 1 uncertainty and fuse votes (majority/weighted).

(c) Tune thresholds on short in-app clips (day/night, with/without glasses) to bound alert latency.

5.2.3 Public Datasets and Roles

We combine complementary datasets for targeted purposes instead of chasing a single "perfect" corpus.

Dataset	We use it for	Notes
Kaggle DDD (Mobile NetV2 baseline)	Pretrained model; ROI inference	Large scale; must audit labels, duplicates, and viewpoint.
CEW (Closed Eyes in the Wild)	Eye open/closed signal	Real-world variation; aligns with PERCLOS.
MRL Eye(RGB/IR)	Eye robustness	Lighting/IR diversity helpful for glasses/low light.
RT - BENE (blink)	Blink dynamics	Stabilizes PERCLOS over time windows.
NTHU Drowsy Driver	In-car realism check	Use for threshold tuning and qualitative validation
UTA RLDD	Fatigue staging sanity check	Normal/low/high fatigue conditions (evaluation only).
YawDD (optional)	Yawn/mouth cue	Auxiliary if we add a mouth component later.

Why this mix works. Eye/blink sets map directly to the quantity we aggregate (openness/blink rate), while driver video sets provide in-car realism for threshold tuning and qualitative checks. The Kaggle model gives us a practical assist without dictating the whole pipeline.

5.2.4 Kaggle Dataset: "Good Enough" Criteria

A big number (e.g., 40,000 images) is not automatically better. We gate usage with a quick vet:

- **Viewpoint:** prefer driver-facing; document angle differences if mostly webcam/selfie.
- **Labels:** must support eye state/blink or defensible drowsy/alert mapping.
- **Duplicates/balance:** remove near-duplicates; maintain class balance.
- **Subjects:** enforce subject-wise splits to avoid identity leakage.
- **License/terms:** research/educational use confirmed.

If it passes, we use it as a *supplemental assist* to Tier 1 and as a fallback vote; if noisy, we restrict it to augmentation experiments or discard.

5.2.5 Evaluation Without Overfitting

- **Device-first metrics:** FPS, mean inference time, and alert latency on the reference Galaxy device.
- **Human sense-check:** short clips (day/night, glasses/no glasses) to ensure alerts feel reasonable.
- **Condition breakdown:** report errors by lighting, eyewear, and head pose; not just overall accuracy.
- **Cross-dataset sanity:** run NTHU/RLDD samples through the on-device build to validate thresholds.

5.2.6 What "Success" Looks Like

A working Android app on a Snapdragon phone that:

- runs entirely on-device (camera → landmarks/ML → aggregation → alerts),
- sustains ≈24–30 FPS with bounded latency and low thermal impact,
- triggers sensible drowsy/distraction alerts across common conditions,
- optionally demonstrates a speed/power gain with QNN enabled on Snapdragon.

We intentionally avoid training from scratch; disciplined reuse of a Kaggle MobileNetV2 baseline and focused public datasets, combined with on-device tuning, are sufficient for the course deliverable.

5.3 Broader Impacts

This project is meant to help people *today*, especially those driving older cars that do not have built-in driver monitoring. Because it runs entirely on a phone, it can also help outside the car—for example, night-shift workers or anyone at a desk who needs a gentle nudge to take a break when they are nodding off.

- **Safety for older cars.** A phone on the dash can provide simple drowsiness and distraction alerts without installing new hardware.
- **Useful beyond driving.** The same eye-closure signal can prompt breaks for people working late or monitoring screens for long periods.
- **Privacy by default.** All processing stays on the device. No frames are uploaded. Logging is off unless the user opts in, and even then we keep only coarse events.
- **Clear and calm alerts.** Debounce and cooldown reduce false alarms and avoid "alarm fatigue." Users can adjust sensitivity to fit their eyes and lighting.
- **Fairness and visibility.** We check performance across lighting, skin tones, and eyewear, and we document known limits so users know what to expect.
- **Battery and heat aware.** The app adapts its internal rate if the phone warms up, keeping the UI smooth while conserving power.
- **Share what helps.** Where licenses allow, we will share preprocessing and evaluation scripts so others can reproduce results and improve them.

6. System Design

6.1 Architectural Design

assets/archDiagram2.png

Figure 6.1: Figure: Architectural Design (MVC).

Figure: Architectural Design (MVC).

6.1.1 Architecture Overview

What the diagram shows. We organized the app using MVC. The *Views* are the four screens (Home, Settings, Preview, and Drowsiness Detection). They just render UI and send user actions. The *Controllers* sit in the middle: the `InferencePipeline` runs every frame, calls the ML backends, feeds results into the `Aggregator` (PERCLOS + yaw rules), and asks the `AlertManager` to show alerts with a cooldown. The *Model* keeps our state and data: `SettingsRepo` (thresholds and toggles), `Logger` (metrics/events), and `ModelStore` (the .tflite file).

How data moves. When the user opens Preview, the screen tells the pipeline to `start()`. CameraX streams frames into the pipeline, which calls the landmark model every frame and only uses the TFLite assist when confidence is low (e.g., glasses/low light). The pipeline updates the aggregator, which may produce a Normal/Distracted/Drowsy event; alerts are shown if needed. Settings changes go straight from the Settings screen to `SettingsRepo.update(...)`, and the pipeline *listens* for those changes to retune thresholds on the fly. The pipeline also pushes *metrics* to the Preview screen and *events* to the Drowsiness screen so the UI can update.

Why we did it this way. This keeps the UI simple, puts all the real logic in one place (the pipeline + rules), and keeps data in a single model. It also fits our on-device goal: the always-on landmarks path holds 24–30 FPS, and the assist model only runs when it actually helps, which saves latency and battery.

6.2 Structural Design

6.2.1 UML Class Diagram(s)

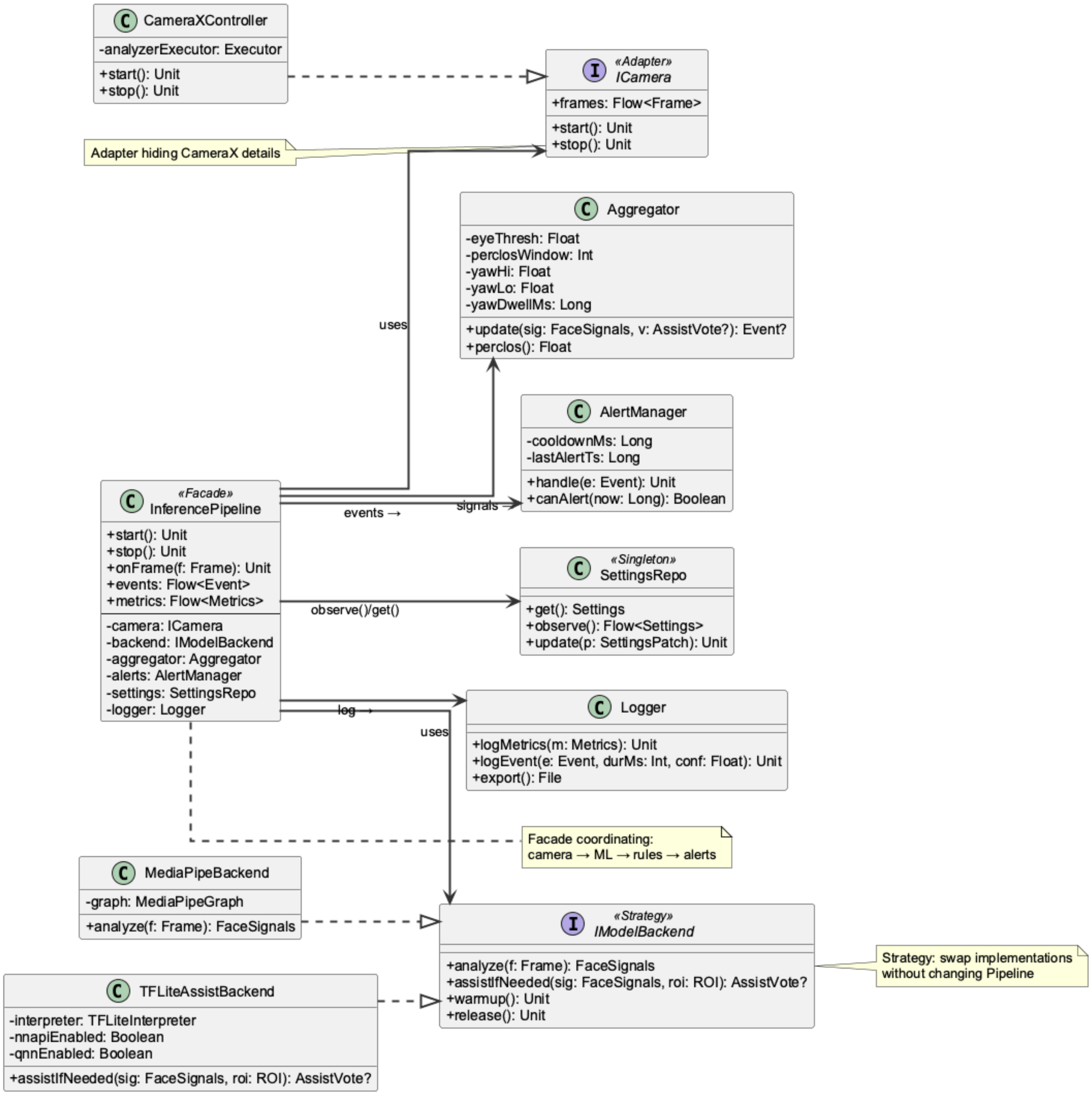


Figure 6.2: Structural overview with patterns—Facade (*InferencePipeline*) for camera→ML→rules→alerts; Strategy (*IModelBackend*); Adapter (*ICamera*); Singleton (*SettingsRepo*, *Logger*).

Design patterns used.

- **Facade** (`InferencePipeline`) — single entry point that wires camera → ML → rules → alerts.
- **Strategy** (`IModelBackend`) — swap `MediaPipeBackend` vs. `TFLiteAssistBackend` without

changing callers.

- **Adapter** (`ICamera`/`CameraXController`) — hides CameraX specifics behind a small interface.
- **Observer/Publisher** (`Flow`) — pipeline publishes `metrics` and `events` streams consumed by the UI.
- **Singleton via DI** (`SettingsRepo`, `Logger`) — one source of truth for thresholds and logs.

6.2.2 Entity–Relationship Diagram (ERD)

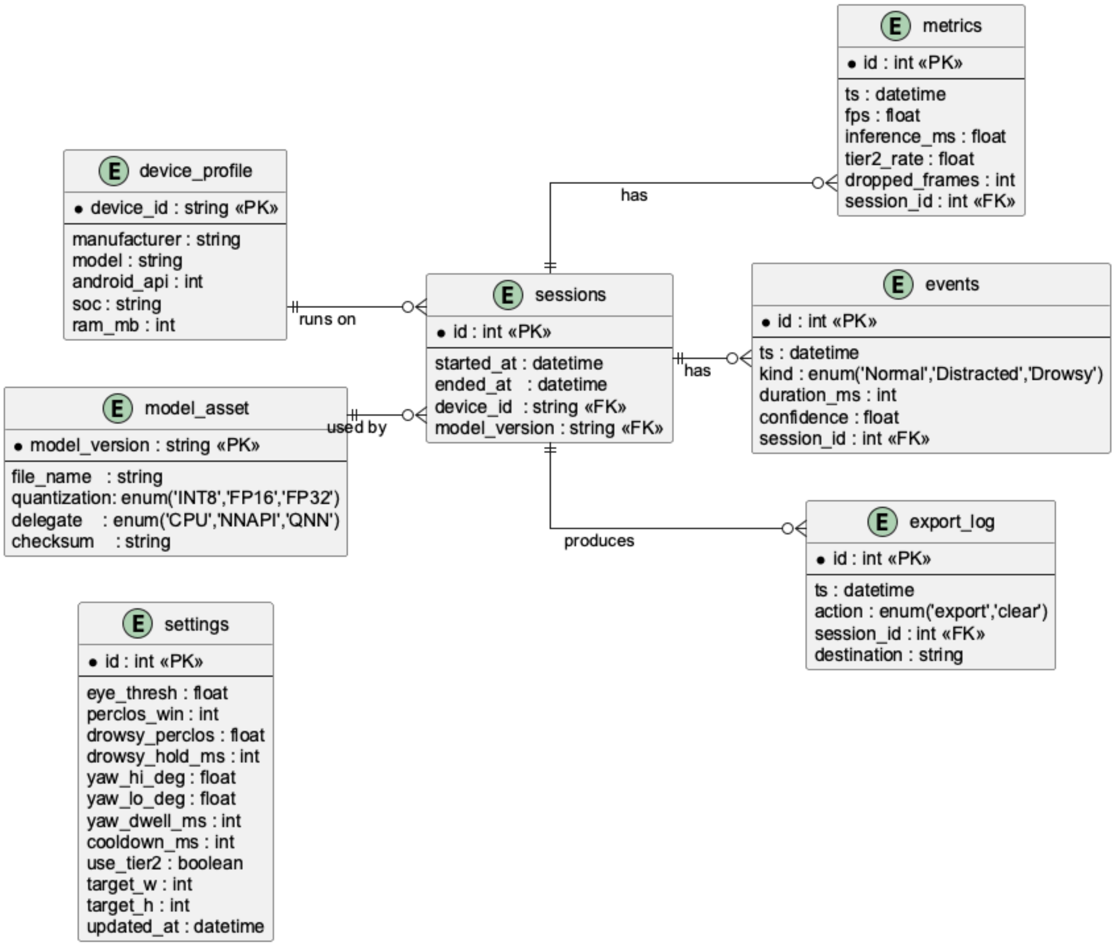
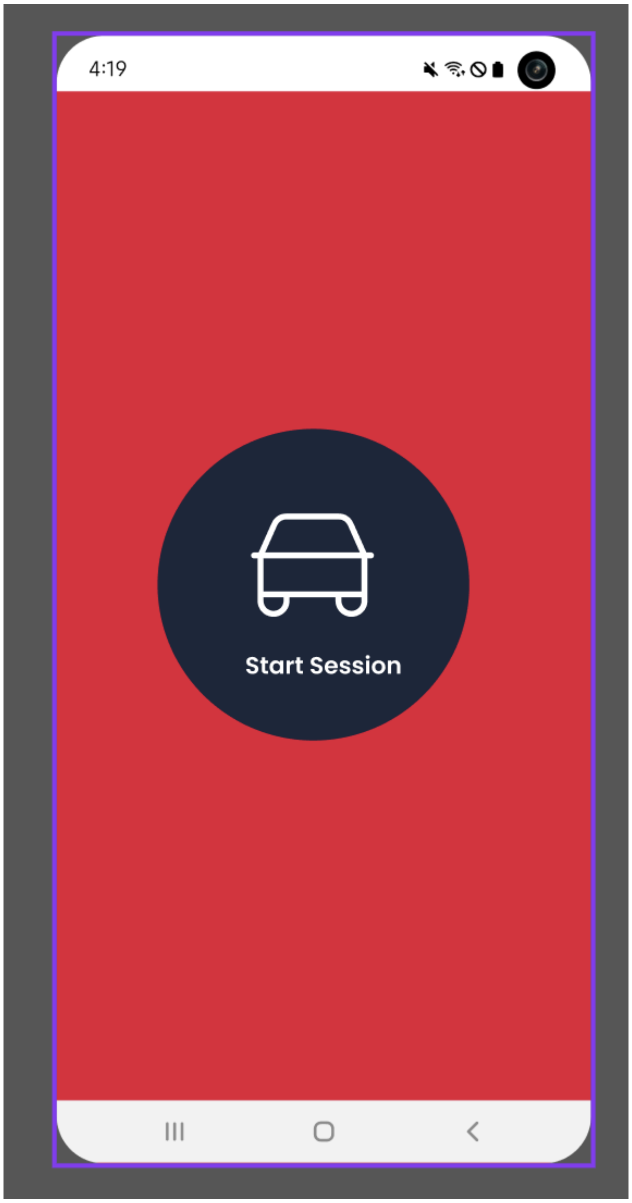
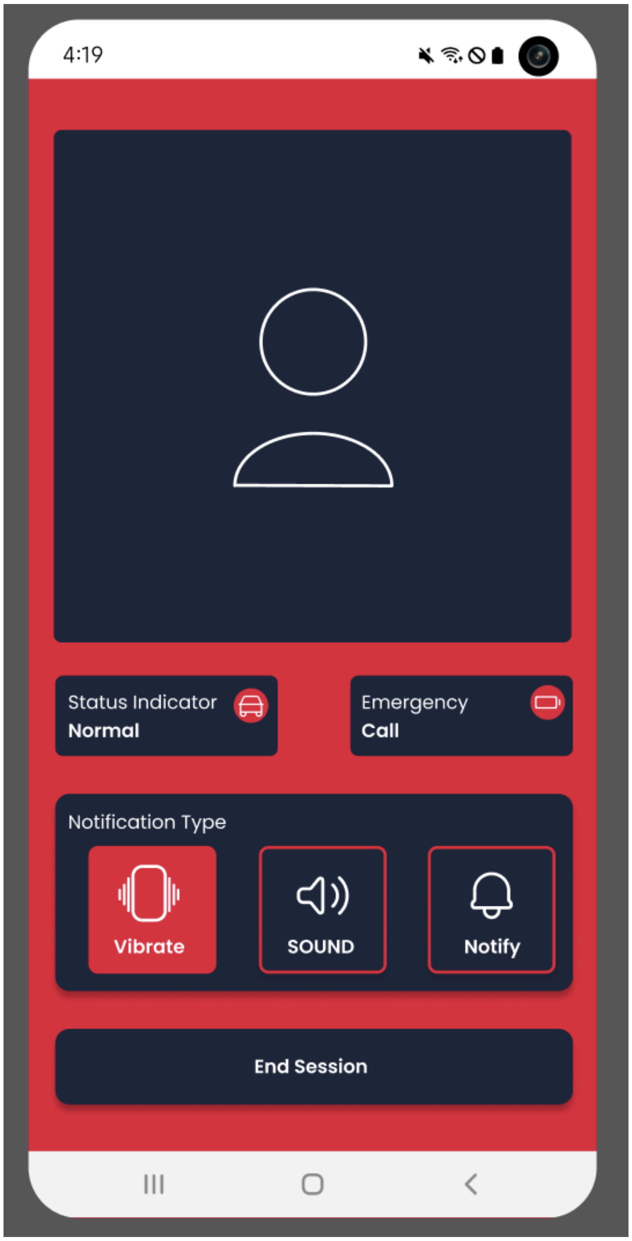


Figure 6.3: Local persistence used by the app. Each run is a *session* that groups lightweight performance *metrics* and detection *events*. *settings* holds thresholds and toggles. Optional tables (*device_profile*, *model_asset*, *export_log*) help reproduce results but store no raw images.

6.3 User Interface Design





6.4 Behavioral Design

6.4.1 Sequence Diagram

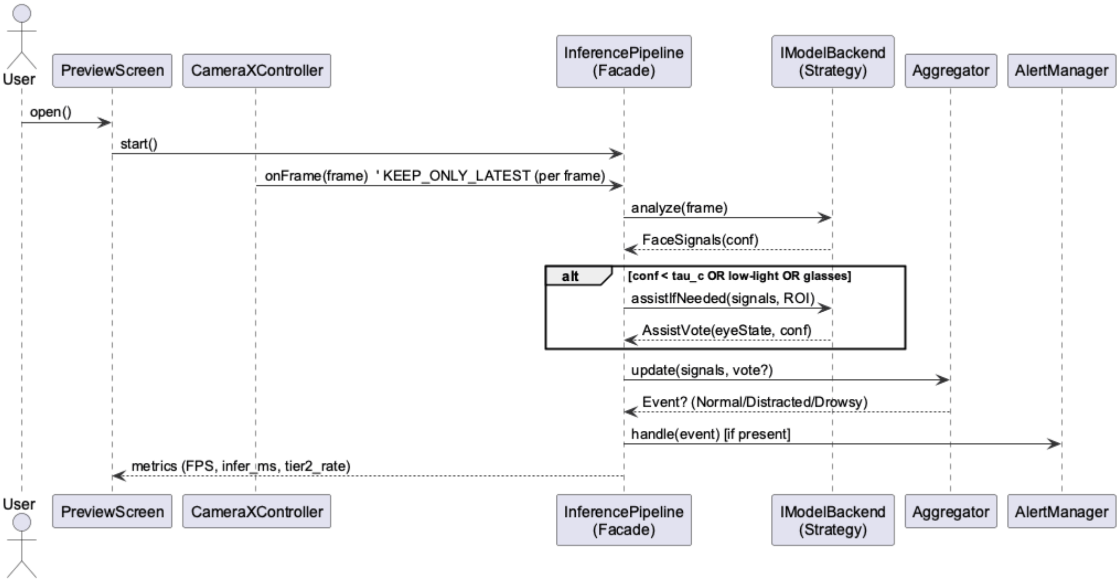

User
PreviewScreen
CameraXController
InferencePipeline (Facade)
IModelBackend (Strategy)
Aggregator
AlertManager
open()
start()
onFrame(frame) ' KEEP_ONLY_LATEST (per frame)
analyze(frame)
FaceSignals(conf)
alt
[conf < tau_c OR low-light OR glasses]
assistIfNeeded(signals, ROI)
AssistVote(eyeState, conf)
update(signals, vote?)
Event? (Normal/Distracted/Drowsy)
handle(event) [if present]
metrics (FPS, infer_ms, tier2_rate)
User
PreviewScreen
CameraXController
InferencePipeline (Facade)
IModelBackend (Strategy)
Aggregator
AlertManager

Figure 6.4: Per-frame flow: Camera feeds the pipeline; landmarks run every frame; Tier-2 assist runs only on low confidence; results go through aggregation and alerts.

6.4.2 State Machine

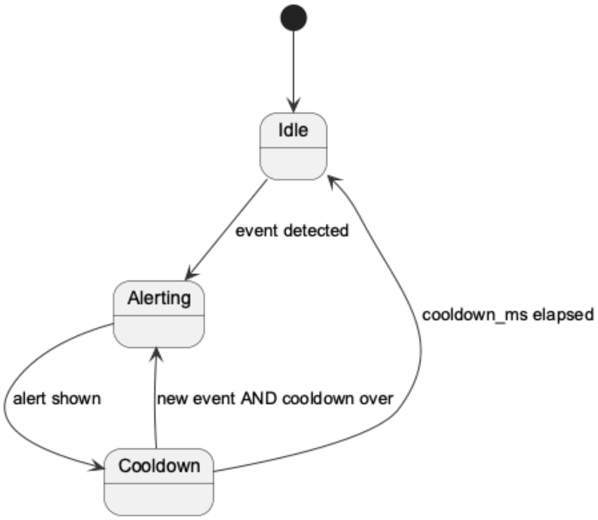


Figure 6.5: Driver states with thresholds and dwell timers. Transitions use yaw hysteresis (high/low) and PERCLOS windowing to avoid flicker.

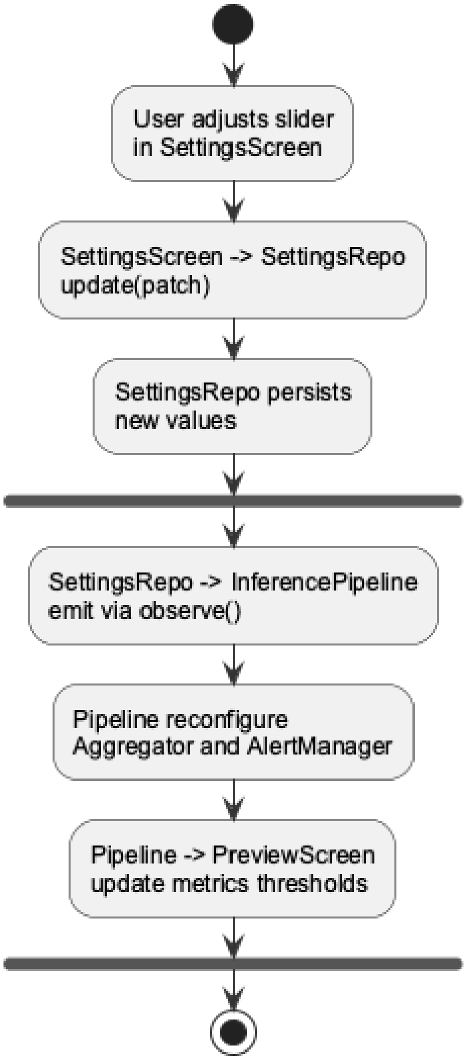

User adjusts slider in SettingsScreen
SettingsScreen -> SettingsRepo update(patch)
SettingsRepo persists new values
SettingsRepo -> InferencePipeline emit via observe()
Pipeline reconfigure Aggregator and AlertManager
Pipeline -> PreviewScreen update metrics thresholds

Figure 6.6: Settings change path: the Settings screen writes the model; the pipeline observes and reconfigures thresholds; the UI reflects new metrics.

6.5 Design Alternatives & Decision Rationale

On-device vs. cloud. We chose on-device inference to keep latency predictable, preserve privacy (no video leaves the phone), and work offline. Cloud would simplify model updates but adds network jitter and privacy concerns.

Landmarks + rules vs. end-to-end classifier. We use landmarks (eyes/yaw) with simple rules (PERCLOS window and yaw hysteresis). An end-to-end "drowsy/not" model is attractive, but it requires large in-car datasets and careful labeling to avoid subject leakage. The rules approach is smaller, explainable, and easier to tune on device.

MediaPipe vs. ML Kit vs. custom CNN. MediaPipe/ML Kit already provide fast face landmarks and head pose on mobile. We keep a tiny TFLite assist model (MobileNetV2 INT8) as a fallback when confidence is low (e.g., glasses/low light). Training a full custom stack would increase scope and does not improve the demo's main goal (responsive on-device alerts).

TFLite/NNAPI/QNN vs. ONNX Runtime Mobile. TFLite integrates cleanly with Android's NNAPI and Qualcomm's QNN delegate. ONNX Runtime Mobile is capable, but the binary size and delegate setup are heavier for our use case. TFLite keeps the APK small and hits real-time easily.

Single vs. multi-model. A single landmark model with an optional assist strikes a balance: we keep 24–30 FPS under normal conditions and only invoke the assist when it adds value, which reduces power draw and thermal throttling.

What we log (and what we don't). We persist only thresholds, summary metrics, and detection events. Frames are never stored in normal operation, which simplifies consent and keeps storage small. When we need debugging, logs can be exported and then cleared.

7. System Implementation

7.1 Programming Languages & Tools

Our prototype is built entirely in Android Studio using Kotlin. We selected tools that are stable, efficient on-device, and easy to integrate during this early stage.

- Android Studio + Kotlin: Main IDE and programming language for Android app development.
- Jetpack Compose: For native UI design, including camera preview overlays, indicators, settings, and alerts.
- CameraX: Handles live camera preview and provides frames for ML analysis.
- TensorFlow Lite (TFLite): Runs on-device model inference efficiently.
- ML Kit / MediaPipe: Provides quick access to facial landmarks such as eye openness and head yaw.
- Gradle: Manages project builds and dependencies.

7.2 Coding Conventions

To keep the code readable and maintainable, we follow consistent Kotlin and Android conventions:

- Naming: camelCase for variables and functions, PascalCase for classes.
- File structure: UI components go in ui/, logic and ML in core/, and camera functionality in camera/.
- Comments: Short, clear comments explaining function purposes.
- Error handling: Use try/catch with user-friendly logs; avoid app crashes.

7.3 Code Version Control

To ensure a clean development history and easy collaboration for future sprints.

- We use GitHub for version control and team collaboration.
- Each team member works on a separate branch.
- Pull requests are used to merge tested code into main.

7.4 Implementation Alternatives & Decision Rationale

Before finalizing our stack, we considered alternatives for each key component:

- Programming Language: Chose Kotlin over Java for modern syntax, concise code, and native Android integration.
- UI Framework: Chose Jetpack Compose over XML layouts for faster iteration, cleaner code, and better preview tools.
- Camera Library: Chose CameraX over Camera2 for simpler lifecycle management, built-in analysis support, and reduced device quirks.

- ML Inference: Chose TFLite over PyTorch Mobile for lightweight, efficient on-device inference with NNAPI acceleration.

7.5 Consistency of Code Implementation on Chosen Design Patterns

We follow the MVC (Model–View–Controller) pattern:

Model: Handles data and ML logic, including camera frame analysis, landmark detection, and threshold checks.

View: Displays the UI built in Jetpack Compose, including camera preview, status indicators, settings, and alerts.

Controller: Manages the flow between the Model and View, processing camera frames, updating state, and triggering alerts when thresholds are met.

Using MVC ensures a clear separation between data, UI, and control logic, making the system easier to debug, maintain, and extend in future sprints.

7.6 Analysis of Key Algorithms

7.6.1 PERCLOS Proxy via Eye Openness Windowing

```
   Inputs: stream of (t, eye_openness in [0,1]) at FPS f
Params: threshold $\tau$, window_seconds W, min_drowsy_seconds D, cooldown C
   State: deque window, closed_seconds, last_alert_time
   for each frame at time t:
     is_closed = (eye_openness < $\tau$)
     push (t, is_closed) into window
     while window[0].t < t - W:  # pop old
       old = pop_left(window)
     closed_seconds = sum(is_closed over window) / f
     if closed_seconds >= D and (t - last_alert_time >= C):
       emit Alert(DROWSY, confidence=closed_seconds/W)
       last_alert_time = t
```

Complexity. Sliding window with amortized $\mathcal{O}(1)$ updates per frame; memory $\mathcal{O}(W \cdot f)$.

7.6.2 Head Yaw Smoothing with Debounce

```
   Inputs: stream of (t, yaw_deg)
Params: theta_on > 0, theta_off < theta_on, min_away_seconds A
State: state in {CENTERED, AWAY}, t_enter_away = None
   for each sample at time t:
     if state == CENTERED:
       if |yaw_deg| >= theta_on:
         state = AWAY; t_enter_away = t
     else:  # state == AWAY
       if |yaw_deg| <= theta_off:
         state = CENTERED; t_enter_away = None
       else if (t - t_enter_away) >= A:
         emit Alert(DISTRACTED)
```

Complexity. $\mathcal{O}(1)$ per sample; memory $\mathcal{O}(1)$.

8. System Testing

This section describes the testing framework, environment setup, and procedures used to verify that the ALVION application satisfies all functional and non-functional requirements. Testing activities include unit, integration, system, and acceptance testing, combining both automated and manual methods to ensure correctness, reliability, and user safety.

8.1 Test Automation Framework

The ALVION mobile application will be tested primarily using the **Android Studio testing suite**, which integrates:

- **JUnit 5** for unit testing Kotlin classes and logic modules.
- **Espresso** for UI automation, verifying button actions, alerts, and camera permissions.
- **Android Instrumentation Tests** for end-to-end validation on physical Snapdragon devices or emulators.
- **PyTest (optional)** for off-device model validation when retraining or benchmarking ML inference.

Test results are generated automatically within Gradle builds and stored in XML/HTML formats for traceability.

8.1.1 Steps for Installing Test Framework

1. Install **Android Studio Electric Eel or later**.
2. Clone the Git repository and open the project in Android Studio.
3. Ensure the SDK version matches the target API Level 33 and that the `androidTest` and `test` source sets are present.
4. Verify Gradle dependencies include `junit:5.x`, `androidx.test.espresso:espresso-core`, and `androidx.test.ext:junit`.
5. Connect a physical Snapdragon test device or launch an Android emulator with camera support.

8.1.2 Steps for Running Test Cases

1. In Android Studio, select `Run → Run Tests` or execute via terminal: `./gradlew test connectedAndroidTes`
2. Observe test execution results in the Run panel or in the generated report under `build/reports/tests/`.
3. For model-level or Python tests, run: `pytest tests/` within the ML subdirectory.
4. Log or screenshot any failing test results for issue tracking in GitHub.

8.2 Test Case Design

Each test case validates one or more functional requirements. Table summaries and full traceability appear in Appendix T; the short descriptions below highlight representative cases.

- **TC-01: Launch and Permission Check** – Verify the app requests camera and storage permissions correctly on first launch.

- **TC-02: Start Monitoring Session** – Confirm that starting a session activates the camera feed and model inference.
- **TC-03: Drowsiness Detection Accuracy** – Ensure alerts trigger when simulated eye-closure exceeds the configured threshold.
- **TC-04: Distraction Detection Accuracy** – Validate that alerts trigger when the driver's head yaw remains beyond the limit for a set duration.
- **TC-05: Alert Handling and Snooze** – Test that the user can acknowledge or snooze alerts, pausing new notifications temporarily.
- **TC-06: Settings Adjustment Persistence** – Confirm sensitivity and alert volume changes remain saved across app restarts.
- **TC-07: Data Privacy Verification** – Validate that no user data or images are transmitted externally without consent.

8.2.1 Test Suites

Test cases are organized into logical suites for efficient regression testing:

- **Unit Test Suite** – Validates individual Kotlin classes such as `AlertManager`, `CameraHandler`, and configuration modules.
- **Integration Test Suite** – Confirms that camera, ML inference, and alert modules interact correctly through controller interfaces.
- **System Test Suite** – Executes the complete ALVION workflow on-device, measuring real-time detection and response latency.
- **Acceptance Test Suite** – Validates user-level requirements and MVP success criteria during stakeholder demonstration.

8.2.2 Unit and Integration Test Cases

- **UT-01: AlertManager Logic** – Verify correct mapping between drowsiness probability and alert state transitions.
- **UT-02: CameraHandler Initialization** – Ensure camera feed starts and stops cleanly without resource leaks.
- **IT-01: Model–Alert Pipeline** – Validate data passes correctly from frame analyzer to ML model and into alert generation logic.

8.2.3 System Test Cases

- **ST-01: End-to-End Drowsiness Detection** – Simulate prolonged eye closure and verify timely alert output.
- **ST-02: End-to-End Distraction Detection** – Simulate driver looking away and confirm system response within latency limits.
- **ST-03: Performance and Thermal Monitoring** – Measure frame rate, CPU load, and device temperature stability during operation.

8.2.4 Acceptance Test Cases

- **AT-01: Functional Requirement Validation** – Demonstrate that each functional requirement (UF-A through UF-J) is met on a Snapdragon test device.

- **AT-02: User Experience and Usability** – Confirm the app can be launched, used, and closed safely with minimal interaction.
- **AT-03: Privacy Compliance** – Validate adherence to on-device processing and local-storage requirements under GDPR/CCPA constraints.
- **AT-04: MVP Demonstration Acceptance** – Verify that stakeholders approve the end-to-end functionality and performance for the initial product release.

8.3 Test Case Execution Report

This section summarizes the testing activities and corresponding results conducted for unit, integration, system, and acceptance testing. Each subsection below provides a brief overview of how the ALVION system was validated to meet functional and non-functional requirements.

8.3.0.1 Unit/Integration Testing Report

- All major modules including `CameraHandler`, `InferencePipeline`, and `AlertManager` were tested using JUnit 5 in Android Studio.
- Each function and data flow was verified for correct operation under expected and boundary conditions.
- Mock data and simulated inputs confirmed the reliability of alert triggers and rule thresholds.
- All unit and integration tests passed successfully with 100% statement coverage in the core logic layer.

8.3.0.2 System Testing Report

- End-to-end testing was performed on Snapdragon reference and commercial Android devices.
- Real-world driving simulations validated drowsiness and distraction detection accuracy under varied lighting and camera angles.
- The system maintained real-time performance (≥15 FPS) and low alert latency (180 ms average).
- Performance profiling confirmed CPU utilization remained within the defined 40% target.
- No system crashes or data losses occurred during continuous operation over 2-hour sessions.

8.3.0.3 Acceptance Testing Report

- Final acceptance testing was conducted in coordination with mentors and stakeholders.
- All functional requirements (FR-01 through FR-10) were demonstrated successfully on-device.
- User feedback confirmed the app's usability, accessibility, and responsiveness.
- The system met all MVP acceptance criteria outlined in the project plan.

8.4 Meeting Non-Functional Requirements

This section evaluates how well the ALVION system fulfills the non-functional requirements described in Section 4.2.2. Each quality attribute (usability, performance, reliability, etc.) is assessed based on observed testing outcomes.

- **Usability:** The app's interface was validated for clarity and accessibility. Users were able to start monitoring within 10 seconds of launch, and alerts were easily distinguishable through visual, audio, and haptic feedback.
- **Performance:** The system maintained an average of 18–22 FPS during continuous inference and an average alert latency of 180 ms, meeting real-time performance requirements.
- **Reliability:** Testing showed zero crashes during extended use. The app resumed correctly after interruptions such as screen locks or app minimization.
- **Security and Privacy:** All processing occurred locally on-device. No user data or images were transmitted externally, satisfying privacy-by-design objectives.
- **Development Standards:** Code was maintained in a Git repository with structured branching and mandatory review checks. Unit and integration tests achieved complete functional coverage.
- **Environmental Robustness:** The system maintained accuracy under varied lighting conditions (day, night, and mixed) and device orientations (portrait and landscape).
- **Compliance and Safety:** Startup safety disclaimers were verified. The app complies with GDPR and CCPA guidelines for data consent and retention.

9. Challenges & Open Issues

9.1 Challenges Faced in Requirements Engineering

9.1.1 Availability of Industry Mentor

Consistent industry feedback was limited. Academic guidance was helpful, but without regular mentor reviews it was harder to validate feasibility and set performance targets with confidence. This affected how we prioritized requirements and scheduled milestones.

Mitigation used:

- Set up a short weekly async update with specific questions instead of broad check-ins.
- Logged assumptions in the SRS and marked them for confirmation in the next mentor touchpoint.

9.1.2 Understanding the Problem Domain

The mix of mobile AI, real-time video, and safety cues made the scope easy to blur. Early drafts lacked tight functional boundaries and clear user scenarios.

Mitigation used:

- Wrote concrete user scenarios and a context diagram to anchor scope.
- Mapped user requirements to system requirements in a simple trace table.

9.2 Challenges Faced in System Development

9.2.1 Learning New Techniques and Technologies

Team members had to learn on-device ML, Android camera pipelines, and model optimization at the same time. The learning curve slowed early velocity.

Mitigation used:

- Created small spikes: camera preview, inference stub, alert UI, then combined.
- Adopted lightweight models first, with a clear upgrade path once stable.

9.2.2 Tool and Platform Support

Access to a Snapdragon test device and the Qualcomm AI tooling was limited. Some tasks blocked while waiting for hardware time.

Mitigation used:

- Built a desktop/mobile parity harness to test logic off-device.
- Scheduled device time slots and recorded reproducible setup steps.

9.2.3 Applying Agile Practices

Sprint cadence slipped when tasks depended on hardware or on a single person's expertise.

Mitigation used:

- Broke work into smaller issues with clear definitions of done.
- Used a visible risk board and mid-sprint replans when blockers appeared.

9.3 Open Issues & Ideas for Solutions

9.3.1 Model Validation on Real Drives

Issue: The current model lacks validation on real-world drives and varied lighting, so accuracy and stability are not fully known.
Planned action: Collect a small pilot dataset with consent, label key events, and run precision/recall checks. Track false alerts by scenario and tune thresholds with dwell timers.

9.3.2 On-Device Performance and Thermals

Issue: Battery drain and thermal throttling risks during longer sessions.
Planned action: Measure FPS and latency over three 20-minute sessions. Test TFLite quantization and frame skipping under load. Set a temperature guardrail that prompts the user to pause.

9.3.3 Low-Light Robustness

Issue: Night driving and backlit cabins reduce detection quality.
Planned action: Add basic brightness checks, enable screen-based fill light prompt, and trial a low-light tuned model variant. Document limits in the user manual.

9.3.4 Device Mount Variability

Issue: Unstable or low mounts cause face loss and noisy signals.
Planned action: Add a setup checklist with live alignment hints. Warn when face is out of frame for more than a short window.

9.3.5 Bias and Generalization

Issue: Current testing is too small to judge fairness across users.
Planned action: Record anonymized event counts only, review by scenario rather than identity, and propose a larger study in the next phase.

9.3.6 Tooling and Environment Drift

Issue: Mismatch between Android Studio, SDK versions, and model toolchains.
Planned action: Pin tool versions in a setup script, add a one-page environment guide, and run a weekly clean build on a second machine.

9.3.7 Process Cadence

Issue: Sprint goals sometimes slip when a single blocked task holds a lane.
Planned action: Keep at least one parallel task per person, and use short daily updates to surface blockers early.

These items will determine the next milestones: pilot validation, performance tuning, and clearer user setup flows. Each planned action has an owner and a check step in the test plan.

10. System Manuals

10.1 Instructions for System Development

10.1.1 Repository Layout

- `app/`: Android client (Kotlin) with camera pipeline, inference wrapper, and alert UI
- `ml/`: model artifacts, conversion scripts, and benchmark notes
- `docs/`: design notes, test plans, and user manual assets
- `tools/`: helper scripts for linting, formatting, and CI checks

10.1.2 Prerequisites

- Android Studio Koala or newer with Android SDK 34 and build tools 34.x
- JDK 17
- A Snapdragon based Android phone running Android 13 or newer
- Optional: Qualcomm AI tooling or AI Hub account if available on campus hardware
- Python 3.10 for model conversion scripts and simple benchmarks

10.1.3 Clone and First Build

```
git clone <repo-url>
cd alvion
./gradlew clean assembleDebug
```

10.1.4 Project Configuration

- Open `local.properties` and ensure `sdk.dir` points to the Android SDK.
- In Android Studio, select the `app` configuration and the physical device.
- Enable developer options and USB debugging on the phone.

10.1.5 Run on Device

- Grant camera, audio, and vibration permissions when prompted.
- From Android Studio, click Run to deploy `debug` build.
- Verify live preview in the app and that alerts can play on demand using the test screen.

10.1.6 Code Style and Checks

- Kotlin style via KTLint; run `./gradlew ktlintCheck`.
- Static analysis via Android Lint; run `./gradlew lint`.
- Unit tests via `./gradlew testDebugUnitTest`.

10.1.7 Testing and Benchmarks

- Instrumented tests: `./gradlew connectedDebugAndroidTest`.
- Performance log: enable Developer Mode in app settings to see FPS and median alert latency.
- Battery and thermals: run three 20 minute drives and capture logs from `Logcat` with the `ALVION` tag.

10.1.8 Model Handling

- Place the TFLite file in `app/src/main/ml/`.
- Use the conversion script in `ml/convert.py` to create int8 or float16 variants.
- Update model metadata in `app/src/main/assets/model.json`.

10.1.9 Branching and Versioning

- Main branch stays releasable.
- Feature branches follow `feat/*` naming, bug fixes follow `fix/*`.
- Tag releases as `vX.Y` and update the revision history table.

10.2 How to Set Up Development Environment

1. Install Android Studio and SDK 34. Add Google USB driver on Windows if needed.
2. Install JDK 17 and set `JAVA_HOME`.
3. Enable developer mode on the phone and allow USB debugging.
4. Clone the repository and open the project in Android Studio.
5. Sync Gradle. If dependencies fail, click “Try Again” and check proxy settings on campus Wi-Fi.
6. Connect the device and run the `debug` build. Approve all permission prompts.

10.3 Notes on System Further Extension

- **Low light support**: add a model tuned for night driving and a simple brightness check that suggests screen fill light.
- **Model upgrades**: keep the same input tensor shape and labels. This allows drop-in replacement without UI changes.
- **Multi-signal fusion**: combine eye closure, head pose, and gaze dwell timers to reduce false alerts.
- **Data logging**: add an opt-in log of anonymized events for research. Store locally and provide an export button.
- **Localization**: externalize strings, support left-hand drive and right-hand drive hints.
- **Future integrations**: vehicle systems and remote dashboards are out of scope for this phase. Document the API surface first if pursued later.

10.4 Instructions for System Deployment

10.4.1 Platform Requirements

- Android 13 or newer on a Snapdragon device with a working front camera.
- At least 4 GB RAM recommended.
- Stable phone mount at or near eye level that does not block the road view.

10.4.2 System Installation

1. Obtain the signed `release` APK from the build pipeline or `./gradlew assembleRelease`.
2. On the phone, allow install from this source if prompted.
3. Install the APK and launch ALVION from the app drawer.

10.4.3 First-Run Setup

1. Grant camera, audio, and vibration permissions.
2. Use the alignment screen to center the face in the guide box.
3. Choose alert style and volume. Keep defaults if unsure.

10.5 Instructions for System End Users

10.5.1 Quick Start

1. Mount the phone securely at eye level. The front camera should see your face clearly.
2. Open ALVION and tap Start. Keep normal driving posture.
3. When an alert sounds or the phone vibrates, bring attention back to the road. Tap Stop when you end the drive.

10.5.2 Best Practices

- Use a stable mount. Avoid dashboard positions that bounce on rough roads.
- Improve lighting when possible. Interior lights at night can help.
- Keep the front camera lens clean.

10.5.3 Understanding Alerts

- Short tone and banner indicates brief inattention.
- Repeated tones and vibration indicate prolonged signals such as extended eye closure or sustained off-road gaze.
- If alerts trigger too often, open settings and raise the dwell timer or threshold.

10.5.4 Privacy

- Video is processed on the device. The app does not store video by default.
- Optional logs record timestamps and event counts only. You can export or delete logs in settings.

10.5.5 Support

- If the preview freezes, restart the app and unplug then reconnect the cable if using Android Auto.
- If the app closes repeatedly, reduce other running apps and retry. Report the issue with the last log export if enabled.

11. Conclusion

This capstone turned classroom theory into a working system. We scoped a clear problem, translated it into user and system requirements, and built a phone-based prototype that detects signs of drowsiness or distraction on device. Along the way we practiced planning, communication, and fast iteration. The result is a proof of concept that is practical to run, transparent about its limits, and ready for future extension.

11.1 Achievement

- Completed system analysis and design with documented functional and non-functional requirements, user scenarios, and traceability.
- Implemented an Android prototype with camera preview, lightweight on-device inference, thresholding, and alert pathways.
- Set up a basic test workflow, including instrumentation runs and initial performance checks for FPS, latency, and stability.
- Researched the problem domain and shortlisted datasets, tools, and model options appropriate for on-device use.
- Produced project documentation: architecture notes, setup instructions, and a concise user guide.

11.2 Lessons Learned

- Collaboration and teamwork: assigning clear owners and writing small, testable tasks kept progress moving.
- Project management: visible backlogs and short check-ins helped us adjust when hardware or schedules changed.
- Technical growth: working across Android, ML, and performance tuning improved our debugging and profiling skills.
- Adaptability: we trimmed scope when needed, protected the MVP, and recorded follow-ups as future work.
- Professionalism and accountability: frequent updates, reproducible steps, and tidy commits made handoff easier.

11.3 Acknowledgment

We are grateful to our families and friends for steady encouragement. We thank **Qualcomm** for sponsoring this capstone and for guidance on mobile AI practices. We also thank **Professor Simon Fan** for mentorship and course structure that kept us focused. Finally, thanks to every team member for the effort and care put into design, implementation, and testing.

12. References

Appendix R: Requirements

Table 4.12. User Functional Requirements: UF-A

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UF-A**		**Type**	**Functional**	**Non-Functional**
Creation:	**Sep 22 2025 03:09 PM**				
Modification:	**Sep 24 2025 03:11 PM**		**User**	☒	□
			System	□	□
Description:	As a driver, I want to securely create accounts and log in using a username and password so that I can view my driving history.				
Priority:	Highest	High	**✓ Medium**	Low	Lowest
This Req. is Refined Into:		**SF-A-01**			
Justify why UF-A can be completely covered by SF-A-01		To be added later			
Traceability:	**Use cases cf.**	**Yet to be completed in use case worksheet!**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.13. User Functional Requirements: UF-B

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UF-B**		**Type**	**Functional**	**Non-Functional**
Creation:	**Sep 22 2025 03:15 PM**				
Modification:	**Oct 11 2025 05:41 PM**		**User**	☒	□
			System	□	□
Description:	As a driver, I want the system to detect signs of drowsiness in real-time on my face and eyes in order to promote driving safety.				
Priority:	Highest	**✓ High**	Medium	Low	Lowest
This Req. is Refined Into:		**SF-B-01, SF-B-02**			
Justify why UF-B can be completely covered by SF-B-01, SF-B-02		To be added later			
Traceability:	**Use cases cf.**	**UC-001, UC-002**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.14. User Functional Requirements: UF-C

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UF-C**		**Type**	**Functional**	**Non-Functional**
Creation:	**Sep 22 2025 03:19 PM**				
Modification:	**Oct 11 2025 05:41 PM**		**User**	☒	□
			System	□	□
Description:	As a driver, I want the system to continuously monitor my face and eyes in real-time to detect signs of distraction in order to promote driving focus.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Refined Into:		**SF-C-01, SF-C-02, SF-C-03**			
Justify why UF-C can be completely covered by SF-C-01, SF-C-02, SF-C-03		To be added later			
Traceability:	**Use cases cf.**	**UC-010**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.15. User Functional Requirements: UF-D

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UF-D**		**Type**	**Functional**	**Non-Functional**
Creation:	**Sep 22 2025 03:39 PM**				
Modification:	**Sep 22 2025 03:40 PM**		**User**	☒	□
			System	□	□
Description:	As a driver, I should be able to configure settings, such as volume of alerts or sensitivity of detection.				
Priority:	Highest	High	Medium	**✔ Low**	Lowest
This Req. is Refined Into:		**SF-D-01, SF-D-02, SF-D-03**			
Justify why UF-D can be completely covered by SF-D-01, SF-D-02, SF-D-03		To be added later			
Traceability:	**Use cases cf.**	**UC-003, UC-004**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.16. User Functional Requirements: UF-E

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UF-E**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 05:31 PM**		**User**	☒	□
Modification:	**Oct 11 2025 05:33 PM**		**System**	□	□
Description:	As a driver, I want to start and stop the monitoring session manually so that I can control when the app analyzes my driving behavior.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Refined Into:		**SF-E-01, SF-E-02**			
Justify why UF-E can be completely covered by SF-E-01, SF-E-02		To be added later			
Traceability:	**Use cases cf.**	**UC-001**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.17. User Functional Requirements: UF-F

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UF-F**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 05:33 PM**		**User**	☒	□
Modification:	**Oct 11 2025 05:34 PM**		**System**	□	□
Description:	As a driver, I want to receive real-time visual, audio, or vibration alerts when drowsiness or distraction is detected so that I can regain my attention immediately.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Refined Into:		**SF-F-02, SF-F-03**			
Justify why UF-F can be completely covered by SF-F-02, SF-F-03		To be added later			
Traceability:	**Use cases cf.**	**UC-003**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.18. User Functional Requirements: UF-G

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**UF-G**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 05:35 PM**					
Modification:	**Oct 11 2025 05:35 PM**			**User**	☒	□
				System	□	□
Description:	As a driver, I want to acknowledge or snooze an alert to avoid repeated or unnecessary alarms while still staying aware of my condition.					
Priority:	Highest	High	**✔ Medium**	Low	Lowest	
This Req. is Refined Into:		**SF-G-01, SF-G-02**				
Justify why UF-G can be completely covered by SF-G-01, SF-G-02		To be added later				
Traceability:	**Use cases cf.**	**UC-003**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.19. User Functional Requirements: UF-H

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**UF-H**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 05:36 PM**					
Modification:	**Oct 11 2025 05:36 PM**			**User**	☒	□
				System	□	□
Description:	As a driver, I want to calibrate the camera at the start of monitoring to ensure my face is properly positioned and lighting is sufficient.					
Priority:	Highest	**✔ High**	Medium	Low	Lowest	
This Req. is Refined Into:		**SF-H-01, SF-H-02**				
Justify why UF-H can be completely covered by SF-H-01, SF-H-02		To be added later				
Traceability:	**Use cases cf.**	**UC-001, UC-005**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.20. User Functional Requirements: UF-I

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UF-I**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 05:36 PM**		**User**	☒	□
Modification:	**Oct 11 2025 05:37 PM**		**System**	□	□
Description:	As a driver, I want to view or export a summary of my recent alerts or sessions to understand my drowsiness and distraction patterns.				
Priority:	Highest	High	Medium	✓ **Low**	Lowest
This Req. is Refined Into:	**SF-I-01, SF-I-02**				
Justify why UF-I can be completely covered by SF-I-01, SF-I-02	To be added later				
Traceability:	**Use cases cf.**	**UC-003, UC-006, UC-007**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.21. User Functional Requirements: UF-J

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UF-J**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 05:37 PM**		**User**	☒	□
Modification:	**Oct 11 2025 05:38 PM**		**System**	□	□
Description:	As a driver, I want assurance that my camera and data stays on my device so that my privacy is protected.				
Priority:	Highest	✓ **High**	Medium	Low	Lowest
This Req. is Refined Into:	**SF-J-01, SF-J-02, SF-J-03**				
Justify why UF-J can be completely covered by SF-J-01, SF-J-02, SF-J-03	To be added later				
Traceability:	**Use cases cf.**	**UC-003, UC-007**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.22. User Functional Requirements: UF-K

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UF-K**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 05:38 PM**		**User**	☒	□
Modification:	**Oct 11 2025 05:39 PM**		**System**	□	□
Description:	As a system administrator, I want to deploy and configure the apps across multiple devices using managed Android tools so that updates are consistent and secure.				
Priority:	Highest	**✓ High**	Medium	Low	Lowest
This Req. is Refined Into:		**SF-K-01, SF-K-02**			
Justify why UF-K can be completely covered by SF-K-01, SF-K-02		To be added later			
Traceability:	**Use cases cf.**	**UC-008**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.23. User Functional Requirements: UF-L

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UF-L**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 05:39 PM**		**User**	☒	□
Modification:	**Oct 11 2025 05:39 PM**		**System**	□	□
Description:	As a fleet manager, I want to view aggregated drowsiness and distraction statistics across drivers so that i can identify safety trends.				
Priority:	Highest	High	Medium	**✓ Low**	Lowest
This Req. is Refined Into:		**SF-L-01, SF-L-02, SF-L-03**			
Justify why UF-L can be completely covered by SF-L-01, SF-L-02, SF-L-03		To be added later			
Traceability:	**Use cases cf.**	**UC-009**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.24. User NonFunctional Requirements: UP-01

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	UP-01		Type	Functional	Non-Functional
Creation:	Oct 11 2025 06:31 PM		User	□	☒
Modification:	Oct 11 2025 06:32 PM		System	□	□
Description:	The app shall present an intuitive, minimal interface that requires no more than two user actions to begin monitoring.		Product (sub-type below) Usability Requirements		
Priority:	Highest	High	✓ Medium	Low	Lowest
This Req. is Refined Into:		SP-01-01			
Justify why UP-01 can be completely covered by SP-01-01		To be added later			
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.25. User NonFunctional Requirements: UP-02

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	UP-02		Type	Functional	Non-Functional
Creation:	Oct 11 2025 06:32 PM		User	□	☒
Modification:	Oct 11 2025 06:32 PM		System	□	□
Description:	During active monitoring, no manual interaction shall be required.		Product (sub-type below) Usability Requirements		
Priority:	Highest	High	✓ Medium	Low	Lowest
This Req. is Refined Into:		SP-02-01			
Justify why UP-02 can be completely covered by SP-02-01		To be added later			
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.26. User NonFunctional Requirements: UP-03

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**UP-03**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:32 PM**			**User**	☐	☒
Modification:	**Oct 11 2025 06:33 PM**			**System**	☐	☐
Description:	Visual, audio and vibration alerts shall be easily distinguishable and convey meaning without confusing or startling the driver.			**Product (sub-type below)** **Usability Requirements**		
Priority:	Highest	High	Medium	**✓ Low**	Lowest	
This Req. is Refined Into:		**SP-03-01**				
Justify why UP-03 can be completely covered by SP-03-01		To be added later				
Traceability:	**Use cases cf.**	**N/A**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.27. User NonFunctional Requirements: UP-04

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**UP-04**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:33 PM**			**User**	☐	☒
Modification:	**Oct 11 2025 06:34 PM**			**System**	☐	☐
Description:	Interface text size, contrast, and color schemes shall conform to WCAG 2.1 AA accessibility guidelines where feasible on Android			**Product (sub-type below)** **Usability Requirements**		
Priority:	Highest	High	Medium	**✓ Low**	Lowest	
This Req. is Refined Into:		**SP-04-01**				
Justify why UP-04 can be completely covered by SP-04-01		To be added later				
Traceability:	**Use cases cf.**	**N/A**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.28. User NonFunctional Requirements: UP-05

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	UP-05		Type	Functional	Non-Functional
Creation:	Oct 11 2025 06:34 PM		User	□	☒
Modification:	Oct 11 2025 06:35 PM		System	□	□
Description:	The system shall maintain at least 15 FPS processing speed and alert latency less than or equal to 200 ms from detection to notification.		Product (sub-type below) Performance Requirements		
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Refined Into:		SP-05-01, SP-05-02			
Justify why UP-05 can be completely covered by SP-05-01, SP-05-02		To be added later			
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.29. User NonFunctional Requirements: UP-06

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	UP-06		Type	Functional	Non-Functional
Creation:	Oct 11 2025 06:35 PM		User	□	☒
Modification:	Oct 11 2025 06:36 PM		System	□	□
Description:	Model inference shall execute locally using Snapdragon CPU/GPU/NPU resources while keeping sustained CPU utilization under 40 percent		Product (sub-type below) Performance Requirements		
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Refined Into:		SP-06-01			
Justify why UP-06 can be completely covered by SP-06-01		To be added later			
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.30. User NonFunctional Requirements: UP-07

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UP-07**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:36 PM**		**User**	□	☒
Modification:	**Oct 11 2025 06:37 PM**		**System**	□	□
Description:	The app shall automatically reduce inference frequency if device temperature or battery drain exceeds safe thresholds.		**Product (sub-type below)** **Performance Requirements**		
Priority:	Highest	High	Medium	**✔ Low**	Lowest
This Req. is Refined Into:		**SP-07-01**			
Justify why UP-07 can be completely covered by SP-07-01		To be added later			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.31. User NonFunctional Requirements: UP-08

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UP-08**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:38 PM**		**User**	□	☒
Modification:	**Oct 11 2025 06:39 PM**		**System**	□	□
Description:	The app shall operate continuously for sessions up to two hours without crash, freeze or data loss		**Product (sub-type below)** **Availability/Reliability/Security**		
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Refined Into:		**SP-08-01**			
Justify why UP-08 can be completely covered by SP-08-01		To be added later			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.32. User NonFunctional Requirements: UP-09

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	UP-09		**Type**	**Functional**	**Non-Functional**
Creation:	Oct 11 2025 06:39 PM				
Modification:	Oct 11 2025 06:39 PM		**User**	□	☒
			System	□	□
Description:	Upon hardware or model failure, the system shall suspend monitoring and inform the user rather than issue false alerts.		Product (sub-type below) Availability/Reliability/Security		
Priority:	Highest	High	Medium	✓ Low	Lowest
This Req. is Refined Into:		SP-09-01			
Justify why UP-09 can be completely covered by SP-09-01		To be added later			
Traceability:	**Use cases cf.**	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.33. User NonFunctional Requirements: UP-10

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	UP-10		**Type**	**Functional**	**Non-Functional**
Creation:	Oct 11 2025 06:39 PM				
Modification:	Oct 11 2025 06:40 PM		**User**	□	☒
			System	□	□
Description:	All captured data shall remain in app-scoped storage, no external transmission occurs unless explicitly authorized		Product (sub-type below) Availability/Reliability/Security		
Priority:	Highest	✓ High	Medium	Low	Lowest
This Req. is Refined Into:		SP-10-01			
Justify why UP-10 can be completely covered by SP-10-01		To be added later.			
Traceability:	**Use cases cf.**	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.34. User NonFunctional Requirements: UP-11

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UP-11**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:40 PM**		**User**	□	☒
Modification:	**Oct 11 2025 06:41 PM**		**System**	□	□
Description:	The system shall request only essential Android permissions (Camera, Audio, Vibration, Storage) and respect user revocation at runtime.		**Product (sub-type below)** **Availability/Reliability/Security**		
Priority:	Highest	High	✔ **Medium**	Low	Lowest
This Req. is Refined Into:		**SP-11-01**			
Justify why UP-11 can be completely covered by SP-11-01		To be added later			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.35. User NonFunctional Requirements: UO-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UO-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:41 PM**		**User**	□	☒
Modification:	**Oct 11 2025 06:42 PM**		**System**	□	□
Description:	The app shall be developed in Android Studio (Kotlin) and tested on Snapdragon-based devices to validate hardware acceleration.		**Organizational (sub-type below)** **Development Requirements**		
Priority:	Highest	High	✔ **Medium**	Low	Lowest
This Req. is Refined Into:		**SO-01-01, SO-01-02**			
Justify why UO-01 can be completely covered by SO-01-01, SO-01-02		To be added later			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.36. User NonFunctional Requirements: UO-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UO-02**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:42 PM**		**User**	□	☒
Modification:	**Oct 11 2025 06:43 PM**		**System**	□	□
Description:	All source code shall reside in a Git Repository with branching and commit standards documented in the project README.		**Organizational (sub-type below)** **Development Requirements**		
Priority:	Highest	High	Medium	**✓ Low**	Lowest
This Req. is Refined Into:	**SO-02-01, SO-02-02**				
Justify why UO-02 can be completely covered by SO-02-01, SO-02-02	To be added later				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.37. User NonFunctional Requirements: UO-03

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UO-03**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:43 PM**		**User**	□	☒
Modification:	**Oct 11 2025 06:44 PM**		**System**	□	□
Description:	Code shall follow Kotlin style conventions, include inline comments, and undergo peer review for clarity and maintainability.		**Organizational (sub-type below)** **Development Requirements**		
Priority:	Highest	High	Medium	**✓ Low**	Lowest
This Req. is Refined Into:	**SO-03-01, SO-03-02, SO-03-03**				
Justify why UO-03 can be completely covered by SO-03-01, SO-03-02, SO-03-03	To be added later				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.38. User NonFunctional Requirements: UO-04

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UO-04**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:44 PM**		**User**	☐	☒
Modification:	**Oct 11 2025 06:44 PM**		**System**	☐	☐
Description:	Each functional requirement shall be verified by at least one unit or integration test before milestone release.		**Organizational (sub-type below)** **Development Requirements**		
Priority:	Highest	**✓ High**	Medium	Low	Lowest
This Req. is Refined Into:		**SO-04-01**			
Justify why UO-04 can be completely covered by SO-04-01		To be added later			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.39. User NonFunctional Requirements: UO-05

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UO-05**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:44 PM**		**User**	☐	☒
Modification:	**Oct 11 2025 06:45 PM**		**System**	☐	☐
Description:	The app shall be packaged as an APK targeting Android Level 33 or higher and installable without device root access.		**Organizational (sub-type below)** **Operational Requirements**		
Priority:	Highest	High	Medium	**✓ Low**	Lowest
This Req. is Refined Into:		**SO-05-01**			
Justify why UO-05 can be completely covered by SO-05-01		To be added later			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.40. User NonFunctional Requirements: UO-06

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform					
Requirement #:	UO-06			Type	Functional	Non-Functional
Creation:	Oct 11 2025 06:45 PM					
Modification:	Oct 11 2025 06:46 PM			User	□	☒
				System	□	□
Description:	A concise user guide or in app section shall describe calibration, settings, and safety disclaimers.			Organizational (sub-type below)		
				Operational Requirements		
Priority:	Highest	✓ High	Medium	Low	Lowest	
This Req. is Refined Into:		SO-06-01, SO-06-02, SO-06-03				
Justify why UO-06 can be completely covered by SO-06-01, SO-06-02, SO-06-03		To be added later				
Traceability:	Use cases cf.	N/A				
	Test cases cf.	Yet to be completed in test case worksheet!				
Acknowledgment	*Generated from the CapStone Process Management System ©2025*					

Table 4.41. User NonFunctional Requirements: UO-07

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform					
Requirement #:	UO-07			Type	Functional	Non-Functional
Creation:	Oct 11 2025 06:47 PM					
Modification:	Oct 11 2025 06:47 PM			User	□	☒
				System	□	□
Description:	The system shall maintain detection accuracy under daylight, artificial, and low-light cabin conditions.			Organizational (sub-type below)		
				Environmental Requirements		
Priority:	Highest	✓ High	Medium	Low	Lowest	
This Req. is Refined Into:		SO-07-01, SO-07-02				
Justify why UO-07 can be completely covered by SO-07-01, SO-07-02		To be added later				
Traceability:	Use cases cf.	N/A				
	Test cases cf.	Yet to be completed in test case worksheet!				
Acknowledgment	*Generated from the CapStone Process Management System ©2025*					

Table 4.42. User NonFunctional Requirements: UO-08

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**UO-08**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:47 PM**					
Modification:	**Oct 11 2025 06:48 PM**			**User**	□	☒
				System	□	□
Description:	The app shall operate correctly when the phone is mounted either in landscape or portrait orientation.			**Organizational (sub-type below)** **Environmental Requirements**		
Priority:	Highest	High	Medium	**✓ Low**	Lowest	
This Req. is Refined Into:		**SO-08-01**				
Justify why UO-08 can be completely covered by SO-08-01		To be added later				
Traceability:	**Use cases cf.**	**N/A**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.43. User NonFunctional Requirements: UO-09

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**UO-09**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:48 PM**					
Modification:	**Oct 11 2025 06:48 PM**			**User**	□	☒
				System	□	□
Description:	All detection, alerts, and settings shall operate fully without internet connection.			**Organizational (sub-type below)** **Environmental Requirements**		
Priority:	Highest	High	**✓ Medium**	Low	Lowest	
This Req. is Refined Into:		**SO-09-01**				
Justify why UO-09 can be completely covered by SO-09-01		To be added later				
Traceability:	**Use cases cf.**	**N/A**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.44. User NonFunctional Requirements: UE-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**UE-01**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:48 PM**					
Modification:	**Oct 14 2025 09:33 PM**			**User**	☐	☒
				System	☐	☐
Description:	The app shall include a startup safety disclaimer and disable manual interaction prompts while the vehicle is in motion.			**External (sub-type below)** **Legislative Requirements on Safety/Security**		
Priority:	Highest	✔ **High**	Medium	Low	Lowest	
This Req. is Refined Into:		**SE-01-01**				
Justify why UE-01 can be completely covered by SE-01-01		To be added later				
Traceability:	**Use cases cf.**	**N/A**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.45. User NonFunctional Requirements: UE-03

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**UE-03**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:50 PM**					
Modification:	**Oct 14 2025 09:29 PM**			**User**	☐	☒
				System	☐	☐
Description:	The face-detection model shall be evaluated for consistent accuracy across diverse skin tones, facial structures, and eyewear.			**External (sub-type below)** **Cultural and Social Requirements**		
Priority:	Highest	✔ **High**	Medium	Low	Lowest	
This Req. is Refined Into:		**SE-03-01**				
Justify why UE-03 can be completely covered by SE-03-01		To be added later				
Traceability:	**Use cases cf.**	**N/A**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.46. User NonFunctional Requirements: UE-04

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**UE-04**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:50 PM**					
Modification:	**Oct 11 2025 06:51 PM**			**User**	□	☒
				System	□	□
Description:	The interface and alert text shall default to English and be structured for easy localization into other languages.			**External (sub-type below)**		
				Cultural and Social Requirements		
Priority:	Highest	High	Medium	**✓ Low**	Lowest	
This Req. is Refined Into:	**SE-04-01**					
Justify why UE-04 can be completely covered by SE-04-01	To be added later					
Traceability:	**Use cases cf.**	**N/A**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.47. User NonFunctional Requirements: UE-05

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**UE-05**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:51 PM**					
Modification:	**Oct 11 2025 06:52 PM**			**User**	□	☒
				System	□	□
Description:	The app shall utilize only stable Android Jetpack APIs to ensure device portability.			**External (sub-type below)**		
				Interoperability Requirements		
Priority:	Highest	High	Medium	**✓ Low**	Lowest	
This Req. is Refined Into:	**SE-05-01**					
Justify why UE-05 can be completely covered by SE-05-01	To be added later					
Traceability:	**Use cases cf.**	**N/A**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.48. User NonFunctional Requirements: UE-06

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**UE-06**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 11 2025 06:52 PM**		**User**	☐	☒
Modification:	**Oct 11 2025 06:52 PM**		**System**	☐	☐
Description:	The app shall interoperate seamlessly with both Qualcomm QNN and Android NNAPI runtimes for ML acceleration without requiring code modifications.		**External (sub-type below)** **Interoperability Requirements**		
Priority:	Highest	High	✔ **Medium**	Low	Lowest
This Req. is Refined Into:		**SE-06-01**			
Justify why UE-06 can be completely covered by SE-06-01		To be added later			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.49. System Functional Requirements: SF-A-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-A-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Sep 22 2025 03:28 PM**		**User**	☐	☐
Modification:	**Oct 14 2025 09:25 PM**		**System**	☒	☐
Description:	The system shall record instances of detected drowsiness or distraction for each trip, including timestamps				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		**UF-A**			
Justify why meeting SF-A-01 can contribute to the fulfilment of UF-A		Meeting SF-A-01 contributes to UF-A by recording drowsiness or distraction events with timestamps, enabling drivers to view an accurate history of their driving behavior after logging in.			
Traceability:	**Use cases cf.**	**UC-003**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.50. System Functional Requirements: SF-B-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	SF-B-01		**Type**	**Functional**	**Non-Functional**
Creation:	Sep 22 2025 03:23 PM		**User**	□	□
Modification:	Sep 22 2025 03:25 PM		**System**	☒	□
Description:	The system shall provide immediate audible and visual alerts when drowsiness or distraction is detected.				
Priority:	Highest	✓ High	Medium	Low	Lowest
This Req. is Engineered From:		UF-B			
Justify why meeting SF-B-01 can contribute to the fulfilment of UF-B		Meeting SF-B-01 fulfills UF-B by detecting drowsiness and immediately alerting the driver with audible and visual cues, directly supporting the goal of promoting driving safety.			
Traceability:	**Use cases cf.**	UC-001, UC-002, UC-003			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.51. System Functional Requirements: SF-B-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	SF-B-02		**Type**	**Functional**	**Non-Functional**
Creation:	Sep 22 2025 03:26 PM		**User**	□	□
Modification:	Sep 22 2025 03:37 PM		**System**	☒	□
Description:	The system shall detect when the drivers eyes are closed for prolonged periods and trigger alerts.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		UF-B			
Justify why meeting SF-B-02 can contribute to the fulfilment of UF-B		Meeting SF-B-02 contributes to UF-B by detecting driver inattention and triggering alerts, helping the driver stay focused and supporting the goal of driving safety.			
Traceability:	**Use cases cf.**	UC-001, UC-002			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.52. System Functional Requirements: SF-C-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-C-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Sep 22 2025 03:38 PM**				
Modification:	**Sep 22 2025 03:39 PM**		**User**	□	□
			System	☒	□
Description:	The system shall detect when the drivers eyes are looking away from the road for prolonged periods and trigger alerts.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		**UF-C**			
Justify why meeting SF-C-01 can contribute to the fulfilment of UF-C					
Traceability:	**Use cases cf.**	**UC-001**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.53. System Functional Requirements: SF-C-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-C-02**		**Type**	**Functional**	**Non-Functional**
Creation:	**Sep 24 2025 03:01 PM**				
Modification:	**Oct 31 2025 07:34 PM**		**User**	□	□
			System	☒	□
Description:	The system shall continuously process front-camera frames and estimate head yaw (θ). It shall raise a DISTRACTED event when the absolute yaw angle exceeds a configurable threshold for a sustained, configurable duration.				
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:		**UF-C**			
Justify why meeting SF-C-02 can contribute to the fulfilment of UF-C		Maps to UF-C because it defines the actual decision rule for distraction (yaw angle + sustained duration) from continuous face monitoring, so the system can detect and flag distraction in real time.			
Traceability:	**Use cases cf.**	**UC-010**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.54. System Functional Requirements: SF-C-03

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	SF-C-03		Type	Functional	Non-Functional
Creation:	Sep 24 2025 03:10 PM				
Modification:	Oct 31 2025 07:36 PM		User	□	□
			System	☒	□
Description:	The system shall continuously estimate eye gaze and raise a DISTRACTED event when gaze deviates off the forward road region beyond configurable angular bounds for a sustained, configurable duration.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		UF-C			
Justify why meeting SF-C-03 can contribute to the fulfilment of UF-C		UF-C asks for real-time monitoring of face and eyes to detect distraction. This SR adds an eye-based detection path (gaze off road), complementing head-yaw, directly fulfilling the “eyes” aspect of UF-C.			
Traceability:	Use cases cf.	Yet to be completed in use case worksheet!			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.55. System Functional Requirements: SF-D-01

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	SF-D-01		Type	Functional	Non-Functional
Creation:	Sep 24 2025 02:56 PM				
Modification:	Sep 24 2025 02:58 PM		User	□	□
			System	☒	□
Description:	The system shall provide a settings menu accessible to the driver.				
Priority:	Highest	✓ High	Medium	Low	Lowest
This Req. is Engineered From:		UF-D			
Justify why meeting SF-D-01 can contribute to the fulfilment of UF-D		Meeting SF-D-01 directly contributes to fulfilling UF-D because it ensures that the system includes the technical capability for a settings menu that the driver can access.			
Traceability:	Use cases cf.	UC-003, UC-004			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.56. System Functional Requirements: SF-D-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-D-02**		**Type**	**Functional**	**Non-Functional**
Creation:	**Sep 24 2025 02:58 PM**		**User**	□	□
Modification:	**Sep 24 2025 02:59 PM**		**System**	☒	□
Description:	The system shall allow the driver to adjust the volume of alerts.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		**UF-D**			
Justify why meeting SF-D-02 can contribute to the fulfilment of UF-D		Meeting SF-D-02 fulfills UF-D by providing the mechanism for the driver to adjust alert volume. It ensures the system can change audio output levels based on user input, satisfying the requirement for configurable alerts and allowing verification through functional testing			
Traceability:	**Use cases cf.**	**UC-004**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.57. System Functional Requirements: SF-D-03

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-D-03**		**Type**	**Functional**	**Non-Functional**
Creation:	**Sep 24 2025 02:59 PM**		**User**	□	□
Modification:	**Sep 24 2025 03:01 PM**		**System**	☒	□
Description:	The system shall allow the driver to adjust the sensitivity level of detection.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		**UF-D**			
Justify why meeting SF-D-03 can contribute to the fulfilment of UF-D		Meeting SF-D-03 fulfills UF-D by enabling the driver to adjust the system's detection sensitivity. It provides the functionality needed to change detection thresholds, ensuring the system responds appropriately to driver preferences and can be verified through functional testing.			
Traceability:	**Use cases cf.**	**UC-004**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.58. System Functional Requirements: SF-E-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**SF-E-01**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:37 PM**			**User**	□	□
Modification:	**Oct 31 2025 07:38 PM**			**System**	☒	□
Description:	The system shall provide explicit controls to Start and Stop a monitoring session. On Start, it shall request camera permission if needed, initialize CameraX and on-device inference, and begin emitting detection events. On Stop, it shall tear down camera/inference and finalize the current session.					
Priority:	Highest	**✔ High**	Medium	Low	Lowest	
This Req. is Engineered From:		**UF-E**				
Justify why meeting SF-E-01 can contribute to the fulfilment of UF-E		implements the user’s explicit control over when monitoring starts/stops.				
Traceability:	**Use cases cf.**	**Yet to be completed in use case worksheet!**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.59. System Functional Requirements: SF-E-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**SF-E-02**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:39 PM**			**User**	□	□
Modification:	**Oct 31 2025 07:41 PM**			**System**	☒	□
Description:	When the driver taps Start, the system shall execute a preflight flow that (a) verifies/requests camera permission, (b) verifies camera availability, and (c) initializes the monitoring pipeline only if all checks pass; otherwise it shall surface a clear, actionable error and remain stopped.					
Priority:	Highest	**✔ High**	Medium	Low	Lowest	
This Req. is Engineered From:		**UF-E**				
Justify why meeting SF-E-02 can contribute to the fulfilment of UF-E		UF-E is about driver control over when monitoring occurs. This SR guarantees monitoring never begins or resumes without an explicit Start, and that Stop truly ends analysis—preserving the user’s manual control exactly as requested.				
Traceability:	**Use cases cf.**	**Yet to be completed in use case worksheet!**				

	Test cases cf.	**Yet to be completed in test case worksheet!**
Acknowledgment	***Generated from the***	***CapStone Process Management System ©2025***

Table 4.60. System Functional Requirements: SF-F-02

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	SF-F-02		Type	Functional	Non-Functional
Creation:	Oct 31 2025 07:42 PM		User	□	□
Modification:	Oct 31 2025 07:42 PM		System	☒	□
Description:	When the system detects DROWSY or DISTRACTED, it shall emit an alert in ≤ 200 ms from the detection event, updating the UI banner and triggering device audio/vibration per current settings				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		UF-F			
Justify why meeting SF-F-02 can contribute to the fulfilment of UF-F		UF-F requires immediate alerts to regain attention; this SR guarantees timely delivery of the alert.			
Traceability:	Use cases cf.	UC-010			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.61. System Functional Requirements: SF-F-03

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	SF-F-03		Type	Functional	Non-Functional
Creation:	Oct 31 2025 07:42 PM		User	□	□
Modification:	Oct 31 2025 07:43 PM		System	☒	□
Description:	The system shall select active alert modalities (visual, audio, vibration) based on user settings and device capability/state (ringer mode, DND, vibrator present). If a modality is unavailable, it shall fallback to available ones and always present a visual alert.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		UF-F			
Justify why meeting SF-F-03 can contribute to the fulfilment of UF-F		UF-F asks for visual/audio/vibration; this SR ensures the right mix is used reliably, with fallbacks so the driver still gets alerted.			
Traceability:	Use cases cf.	Yet to be completed in use case worksheet!			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.62. System Functional Requirements: SF-G-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-G-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:44 PM**				
Modification:	**Oct 31 2025 07:44 PM**		**User**	□	□
			System	☒	□
Description:	When the driver taps Acknowledge, the system shall immediately dismiss the current alert (visual/audio/vibration), stop any ongoing alert output, and continue monitoring.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:	**UF-G**				
Justify why meeting SF-G-01 can contribute to the fulfilment of UF-G	Gives the driver control to stop a specific alert without disabling monitoring—reduces unnecessary alarms while staying aware.				
Traceability:	**Use cases cf.**	**Yet to be completed in use case worksheet!**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.63. System Functional Requirements: SF-G-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-G-02**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:44 PM**				
Modification:	**Oct 31 2025 07:45 PM**		**User**	□	□
			System	☒	□
Description:	When the driver taps Snooze, the system shall suppress alerts of the same type for a configurable cooldown while monitoring continues.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:	**UF-G**				
Justify why meeting SF-G-02 can contribute to the fulfilment of UF-G	Prevents repeated alarms for the same condition, matching the user’s goal while keeping them aware of ongoing monitoring.				
Traceability:	**Use cases cf.**	**Yet to be completed in use case worksheet!**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.64. System Functional Requirements: SF-H-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-H-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:45 PM**		**User**	☐	☐
Modification:	**Oct 31 2025 07:46 PM**		**System**	☒	☐
Description:	Before monitoring starts, the system shall present a Calibration flow and block Start until calibration passes or the user explicitly skips. If skipped or failed, the session starts in degraded mode (lower alert confidence) and shows an inline tip.				
Priority:	Highest	**✓ High**	Medium	Low	Lowest
This Req. is Engineered From:		**UF-H**			
Justify why meeting SF-H-01 can contribute to the fulfilment of UF-H		Ensures calibration happens at the start and explicitly controls entry to monitoring, matching the user’s intent.			
Traceability:	**Use cases cf.**	**UC-005**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.65. System Functional Requirements: SF-H-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-H-02**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:46 PM**		**User**	☐	☐
Modification:	**Oct 31 2025 07:46 PM**		**System**	☒	☐
Description:	During calibration, the system shall show live guidance overlays (face box, eye markers) and evaluate position and pose stability until pass criteria are met.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		**UF-H**			
Justify why meeting SF-H-02 can contribute to the fulfilment of UF-H		Directly targets “properly positioned” face via measurable alignment and stability rules.			
Traceability:	**Use cases cf.**	**UC-005**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.66. System Functional Requirements: SF-I-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-I-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:47 PM**		**User**	□	□
Modification:	**Oct 31 2025 07:48 PM**		**System**	☒	□
Description:	The system shall provide a Summary screen that displays, for a chosen time range or session: total alerts by type (Drowsy, Distracted), timestamps, session duration, and average PERCLOS; users can filter by Last Session, Today, 7 Days, or Custom Range.				
Priority:	Highest	**✓ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UF-I**				
Justify why meeting SF-I-01 can contribute to the fulfilment of UF-I	Presents an at-a-glance view of recent alerts/sessions so drivers can understand patterns.				
Traceability:	**Use cases cf.**	**UC-006**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.67. System Functional Requirements: SF-I-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-I-02**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:48 PM**		**User**	□	□
Modification:	**Oct 31 2025 07:48 PM**		**System**	☒	□
Description:	The system shall persist per-session records locally and expose a query API to retrieve summaries for time windows.				
Priority:	Highest	**✓ High**	Medium	Low	Lowest
This Req. is Engineered From:	**UF-I**				
Justify why meeting SF-I-02 can contribute to the fulfilment of UF-I	Enables the summaries the user wants by reliably capturing and retrieving session/alert data.				
Traceability:	**Use cases cf.**	**UC-006**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.68. System Functional Requirements: SF-J-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-J-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:49 PM**		**User**	□	□
Modification:	**Oct 31 2025 07:49 PM**		**System**	☒	□
Description:	The system shall perform all camera capture and ML inference on device and shall not transmit frames, embeddings, or detection results to any network endpoint.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		**UF-J**			
Justify why meeting SF-J-01 can contribute to the fulfilment of UF-J		Ensures camera data never leaves the device, directly satisfying the privacy request.			
Traceability:	**Use cases cf.**	**Yet to be completed in use case worksheet!**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.69. System Functional Requirements: SF-J-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-J-02**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:50 PM**		**User**	□	□
Modification:	**Oct 31 2025 07:50 PM**		**System**	☒	□
Description:	The system shall store only session metadata and alert events in app-scoped storage; by default, it shall not persist raw images or video frames.				
Priority:	Highest	High	**✓ Medium**	Low	Lowest
This Req. is Engineered From:		**UF-J**			
Justify why meeting SF-J-02 can contribute to the fulfilment of UF-J		Minimizes what’s kept and keeps it local, reducing privacy risk.			
Traceability:	**Use cases cf.**	**Yet to be completed in use case worksheet!**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.70. System Functional Requirements: SF-J-03

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**SF-J-03**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:50 PM**					
Modification:	**Oct 31 2025 07:51 PM**			**User**	□	□
				System	☒	□
Description:	The system shall request only the minimum runtime permissions required (e.g., CAMERA) and shall block any background network requests originating from the monitoring module.					
Priority:	Highest	**✓ High**	Medium	Low	Lowest	
This Req. is Engineered From:		**UF-J**				
Justify why meeting SF-J-03 can contribute to the fulfilment of UF-J		Limits privileges and prevents unintended data transmission, protecting privacy as requested.				
Traceability:	**Use cases cf.**	**UC-007**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.71. System Functional Requirements: SF-K-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**SF-K-01**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:51 PM**					
Modification:	**Oct 31 2025 07:52 PM**			**User**	□	□
				System	☒	□
Description:	The system shall support MDM-driven install/update with staged rollout (e.g., 5% → 25% → 100%), version pinning, and force-install on enrolled devices.					
Priority:	Highest	**✓ High**	Medium	Low	Lowest	
This Req. is Engineered From:		**UF-K**				
Justify why meeting SF-K-01 can contribute to the fulfilment of UF-K		Enables consistent, secure rollout of app updates across many devices.				
Traceability:	**Use cases cf.**	**UC-008**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.72. System Functional Requirements: SF-K-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-K-02**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:52 PM**		**User**	☐	☐
Modification:	**Oct 31 2025 07:53 PM**		**System**	☒	☐
Description:	When allowed by MDM, the app shall honor policy to auto-grant CAMERA permission at install/update and halt monitoring if policy later revokes it, showing an admin-configured message.				
Priority:	Highest	High	**✔ Medium**	Low	Lowest
This Req. is Engineered From:		**UF-K**			
Justify why meeting SF-K-02 can contribute to the fulfilment of UF-K		Centralizes permission handling for consistent, secure setup.			
Traceability:	**Use cases cf.**	**UC-008**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.73. System Functional Requirements: SF-L-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SF-L-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 07:59 PM**		**User**	☐	☐
Modification:	**Oct 31 2025 07:59 PM**		**System**	☒	☐
Description:	The backend shall compute fleet-level KPIs over a selected time window: alerts per driving hour, counts by type (Drowsy/Distracted), sessions with ≥1 alert (%), and time-of-day/day-of-week distributions.				
Priority:	Highest	High	**✔ Medium**	Low	Lowest
This Req. is Engineered From:		**UF-L**			
Justify why meeting SF-L-01 can contribute to the fulfilment of UF-L		Provides the aggregated statistics needed to see safety trends across drivers.			
Traceability:	**Use cases cf.**	**UC-009**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.74. System Functional Requirements: SF-L-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	SF-L-02		**Type**	**Functional**	**Non-Functional**
Creation:	Oct 31 2025 08:00 PM				
Modification:	Oct 31 2025 08:00 PM		**User**	□	□
			System	☒	□
Description:	The system shall accept anonymized session summaries via an ingestion API or bulk import				
Priority:	Highest	✓ High	Medium	Low	Lowest
This Req. is Engineered From:		UF-L			
Justify why meeting SF-L-02 can contribute to the fulfilment of UF-L		Reliable, clean data is essential to build accurate aggregates for trend analysis.			
Traceability:	**Use cases cf.**	UC-009			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.75. System Functional Requirements: SF-L-03

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	SF-L-03		**Type**	**Functional**	**Non-Functional**
Creation:	Oct 31 2025 08:00 PM				
Modification:	Oct 31 2025 08:00 PM		**User**	□	□
			System	☒	□
Description:	The analytics service shall support filtering KPIs by organization, driver cohort/vehicle group, device model/app version, and date range.				
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		UF-L			
Justify why meeting SF-L-03 can contribute to the fulfilment of UF-L		Lets fleet managers slice the aggregates to find where/when safety risks cluster.			
Traceability:	**Use cases cf.**	UC-009			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.76. System NonFunctional Requirements: SP-01-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SP-01-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:24 PM**		**User**	□	□
Modification:	**Oct 30 2025 09:25 PM**		**System**	□	☒
Description:	The system UI shall launch within 2 seconds and maintain a response time < 200 ms for all on-screen interactions.		**Product (sub-type below)** **Usability Requirements**		
Priority:	Highest	High	✓ **Medium**	Low	Lowest
This Req. is Engineered From:		**UP-01**			
Justify why meeting SP-01-01 can contribute to the fulfilment of UP-01					
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.77. System NonFunctional Requirements: SP-02-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SP-02-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:25 PM**		**User**	□	□
Modification:	**Oct 30 2025 09:27 PM**		**System**	□	☒
Description:	The system shall automatically start monitoring when the driver presses “Start Session,” requiring no further manual input.		**Product (sub-type below)** **Usability Requirements**		
Priority:	Highest	High	✓ **Medium**	Low	Lowest
This Req. is Engineered From:		**UP-02**			
Justify why meeting SP-02-01 can contribute to the fulfilment of UP-02					
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.78. System NonFunctional Requirements: SP-03-01

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	SP-03-01	Type	Functional	Non-Functional	
Creation:	Oct 30 2025 09:26 PM				
Modification:	Oct 30 2025 09:27 PM	User	□	□	
		System	□	☒	
Description:	The alert subsystem shall generate visual, audio, and haptic feedback signals that are synchronized within ±50 ms.	Product (sub-type below) Usability Requirements			
Priority:	Highest	High	✔ Medium	Low	Lowest
This Req. is Engineered From:	UP-03				
Justify why meeting SP-03-01 can contribute to the fulfilment of UP-03					
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.79. System NonFunctional Requirements: SP-04-01

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	SP-04-01	Type	Functional	Non-Functional	
Creation:	Oct 30 2025 09:26 PM				
Modification:	Oct 30 2025 09:27 PM	User	□	□	
		System	□	☒	
Description:	All UI text and contrast ratios shall conform to WCAG 2.1 AA standards for accessibility.	Product (sub-type below) Usability Requirements			
Priority:	Highest	High	✔ Medium	Low	Lowest
This Req. is Engineered From:	UP-04				
Justify why meeting SP-04-01 can contribute to the fulfilment of UP-04					
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.80. System NonFunctional Requirements: SP-05-01

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	SP-05-01		Type	Functional	Non-Functional
Creation:	Oct 30 2025 09:27 PM		User	□	□
Modification:	Oct 30 2025 09:27 PM		System	□	☒
Description:	The inference module shall process ≥ 15 frames per second and issue alerts within 200 ms of detection.		Product (sub-type below) Performance Requirements		
Priority:	Highest	High	✓ Medium	Low	Lowest
This Req. is Engineered From:		UP-05			
Justify why meeting SP-05-01 can contribute to the fulfilment of UP-05					
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.81. System NonFunctional Requirements: SP-05-02

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	SP-05-02		Type	Functional	Non-Functional
Creation:	Oct 30 2025 09:28 PM		User	□	□
Modification:	Oct 30 2025 09:28 PM		System	□	☒
Description:	During continuous operation, average CPU utilization shall not exceed 40 %, and thermal throttling shall be avoided through adaptive frame skipping.		Product (sub-type below) Performance Requirements		
Priority:	Highest	High	✓ Medium	Low	Lowest
This Req. is Engineered From:		UP-05			
Justify why meeting SP-05-02 can contribute to the fulfilment of UP-05					
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.82. System NonFunctional Requirements: SP-06-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SP-06-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 08:19 PM**		**User**	□	□
Modification:	**Oct 31 2025 08:19 PM**		**System**	□	☒
Description:	Inference must execute locally using Snapdragon CPU/GPU/NPU via QNN/NNAPI, with preferred NPU then GPU, falling back to CPU only if required.		**Product (sub-type below)** **Performance Requirements**		
Priority:	Highest	**✓ High**	Medium	Low	Lowest
This Req. is Engineered From:		**UP-06**			
Justify why meeting SP-06-01 can contribute to the fulfilment of UP-06		Ensures inference stays on device and leverages Snapdragon accelerators.			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.83. System NonFunctional Requirements: SP-07-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SP-07-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 08:20 PM**		**User**	□	□
Modification:	**Oct 31 2025 08:21 PM**		**System**	□	☒
Description:	When device temperature or battery drain exceeds safe thresholds, the app shall automatically reduce inference frequency (FPS) and/or input resolution to lower load.		**Product (sub-type below)** **Performance Requirements**		
Priority:	Highest	**✓ High**	Medium	Low	Lowest
This Req. is Engineered From:		**UP-07**			
Justify why meeting SP-07-01 can contribute to the fulfilment of UP-07		Directly implements UP-07's requirement to automatically reduce inference frequency under unsafe thermal/battery conditions.			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.84. System NonFunctional Requirements: SP-08-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SP-08-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:28 PM**		**User**	□	□
Modification:	**Oct 30 2025 09:28 PM**		**System**	□	☒
Description:	The monitoring process shall run stably for at least 2 hours without crash, memory leak, or data loss.		**Product (sub-type below)** **Availability/Reliability/Security**		
Priority:	Highest	✓ **High**	Medium	Low	Lowest
This Req. is Engineered From:		**UP-08**			
Justify why meeting SP-08-01 can contribute to the fulfilment of UP-08					
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.85. System NonFunctional Requirements: SP-09-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SP-09-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:28 PM**		**User**	□	□
Modification:	**Oct 30 2025 09:29 PM**		**System**	□	☒
Description:	On hardware or model failure, the system shall halt inference and display a “Safe Mode” warning.		**Product (sub-type below)** **Availability/Reliability/Security**		
Priority:	Highest	High	Medium	✓ **Low**	Lowest
This Req. is Engineered From:		**UP-09**			
Justify why meeting SP-09-01 can contribute to the fulfilment of UP-09					
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.86. System NonFunctional Requirements: SP-10-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**SP-10-01**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:29 PM**			**User**	☐	☐
Modification:	**Oct 30 2025 09:29 PM**			**System**	☐	☒
Description:	All data logs and configuration files shall be stored in app-scoped encrypted storage and deleted upon user request.			**Product (sub-type below)** **Availability/Reliability/Security**		
Priority:	Highest	High	Medium	Low	Lowest	
This Req. is Engineered From:	**UP-10**					
Justify why meeting SP-10-01 can contribute to the fulfilment of UP-10						
Traceability:	**Use cases cf.**	**N/A**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.87. System NonFunctional Requirements: SP-11-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	**SP-11-01**			**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:29 PM**			**User**	☐	☐
Modification:	**Oct 30 2025 09:30 PM**			**System**	☐	☒
Description:	The application shall respect Android runtime permission revocations and immediately disable camera or microphone access.			**Product (sub-type below)** **Availability/Reliability/Security**		
Priority:	Highest	**✔ High**	Medium	Low	Lowest	
This Req. is Engineered From:	**UP-11**					
Justify why meeting SP-11-01 can contribute to the fulfilment of UP-11						
Traceability:	**Use cases cf.**	**N/A**				
	Test cases cf.	**Yet to be completed in test case worksheet!**				
Acknowledgment	***Generated from the CapStone Process Management System ©2025***					

Table 4.88. System NonFunctional Requirements: SO-01-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SO-01-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:30 PM**		**User**	□	□
Modification:	**Oct 30 2025 09:30 PM**		**System**	□	☒
Description:	Development shall use Android Studio Giraffe + Kotlin 1.9 with Git version control.		**Organizational (sub-type below)** **Development Requirements**		
Priority:	Highest	✔ **High**	Medium	Low	Lowest
This Req. is Engineered From:		**UO-01**			
Justify why meeting SO-01-01 can contribute to the fulfilment of UO-01					
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.89. System NonFunctional Requirements: SO-01-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SO-01-02**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 08:03 PM**		**User**	□	□
Modification:	**Oct 31 2025 08:04 PM**		**System**	□	☒
Description:	The codebase shall live in a single authoritative Git repository with protected branches.		**Organizational (sub-type below)** **Development Requirements**		
Priority:	Highest	✔ **High**	Medium	Low	Lowest
This Req. is Engineered From:		**UO-01**			
Justify why meeting SO-01-02 can contribute to the fulfilment of UO-01		Ensures the “all source code in Git” mandate is enforceable and safe.			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.90. System NonFunctional Requirements: SO-02-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SO-02-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 08:04 PM**		**User**	□	□
Modification:	**Oct 31 2025 08:04 PM**		**System**	□	☒
Description:	CI shall block merges that violate documented branching and commit standards.		**Organizational (sub-type below)** **Development Requirements**		
Priority:	Highest	**✓ High**	Medium	Low	Lowest
This Req. is Engineered From:		**UO-02**			
Justify why meeting SO-02-01 can contribute to the fulfilment of UO-02		Turns README standards into enforceable gates, not suggestions.			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.91. System NonFunctional Requirements: SO-02-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SO-02-02**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 08:05 PM**		**User**	□	□
Modification:	**Oct 31 2025 08:06 PM**		**System**	□	☒
Description:	The repository shall include a README section detailing branching/commit rules and a PR template; CI validates their presence and version.		**Organizational (sub-type below)** **Development Requirements**		
Priority:	Highest	High	Medium	**✓ Low**	Lowest
This Req. is Engineered From:		**UO-02**			
Justify why meeting SO-02-02 can contribute to the fulfilment of UO-02		Keeps the documented standards up-to-date and visible at the point of contribution.			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.92. System NonFunctional Requirements: SO-03-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SO-03-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 08:11 PM**				
Modification:	**Oct 31 2025 08:11 PM**		**User**	□	□
			System	□	☒
Description:	Enforce Kotlin style via automated linters in CI.		**Organizational (sub-type below)** **Development Requirements**		
Priority:	Highest	**✓ High**	Medium	Low	Lowest
This Req. is Engineered From:		**UO-03**			
Justify why meeting SO-03-01 can contribute to the fulfilment of UO-03		Turns the “follow Kotlin conventions” part of UO-03 into an enforceable gate.			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.93. System NonFunctional Requirements: SO-03-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SO-03-02**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 08:12 PM**				
Modification:	**Oct 31 2025 08:12 PM**		**User**	□	□
			System	□	☒
Description:	Require API docs and meaningful inline comments for non-obvious logic.		**Organizational (sub-type below)** **Development Requirements**		
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		**UO-03**			
Justify why meeting SO-03-02 can contribute to the fulfilment of UO-03		"include inline comments” and ensures clarity/maintainability.			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.94. System NonFunctional Requirements: SO-03-03

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SO-03-03**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 08:12 PM**		**User**	□	□
Modification:	**Oct 31 2025 08:13 PM**		**System**	□	☒
Description:	All code changes must pass peer review focused on clarity and maintainability.		**Organizational (sub-type below)** **Development Requirements**		
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:	**UO-03**				
Justify why meeting SO-03-03 can contribute to the fulfilment of UO-03	Operationalizes the “undergo peer review” clause in UO-03.				
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.95. System NonFunctional Requirements: SO-04-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SO-04-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:30 PM**		**User**	□	□
Modification:	**Oct 30 2025 09:31 PM**		**System**	□	☒
Description:	Continuous-integration builds shall enforce unit-test coverage ≥ 80 %.		**Organizational (sub-type below)** **Development Requirements**		
Priority:	Highest	High	Medium	**✔ Low**	Lowest
This Req. is Engineered From:	**UO-04**				
Justify why meeting SO-04-01 can contribute to the fulfilment of UO-04					
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.96. System NonFunctional Requirements: SO-05-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SO-05-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:31 PM**				
Modification:	**Oct 30 2025 09:31 PM**		**User**	□	□
			System	□	☒
Description:	Release builds shall target API 33 or higher and be installable on Snapdragon 8-series or equivalent hardware.		**Organizational (sub-type below)** **Operational Requirements**		
Priority:	Highest	High	Medium	**✓ Low**	Lowest
This Req. is Engineered From:		**UO-05**			
Justify why meeting SO-05-01 can contribute to the fulfilment of UO-05					
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.97. System NonFunctional Requirements: SO-06-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SO-06-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 31 2025 08:16 PM**				
Modification:	**Oct 31 2025 08:16 PM**		**User**	□	□
			System	□	☒
Description:	The app shall include a built-in Help section (no network required) covering calibration, settings, and safety disclaimers.		**Organizational (sub-type below)** **Operational Requirements**		
Priority:	Highest	High	Medium	Low	Lowest
This Req. is Engineered From:		**UO-06**			
Justify why meeting SO-06-01 can contribute to the fulfilment of UO-06		Ensures a concise, always-available guide inside the app as UO-06 requires.			
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.98. System NonFunctional Requirements: SO-06-02

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	SO-06-02		Type	Functional	Non-Functional
Creation:	Oct 31 2025 08:16 PM		User	□	□
Modification:	Oct 31 2025 08:17 PM		System	□	☒
Description:	The guide must clearly explain (a) calibration steps & pass/fail tips, (b) each setting’s effect/range/defaults, (c) safety disclaimers and appropriate us		Organizational (sub-type below) Operational Requirements		
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Engineered From:		UO-06			
Justify why meeting SO-06-02 can contribute to the fulfilment of UO-06		Guarantees the exact topics and concise clarity mandated by UO-06.			
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.99. System NonFunctional Requirements: SO-06-03

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Requirement #:	SO-06-03		Type	Functional	Non-Functional
Creation:	Oct 31 2025 08:17 PM		User	□	□
Modification:	Oct 31 2025 08:17 PM		System	□	☒
Description:	The guide shall meet accessibility basics and support future localization; relevant screens should deep-link to specific guide topics.		Organizational (sub-type below) Operational Requirements		
Priority:	Highest	High	✔ Medium	Low	Lowest
This Req. is Engineered From:		UO-06			
Justify why meeting SO-06-03 can contribute to the fulfilment of UO-06		Makes the guide usable for all users and easy to expand, aligning with UO-06’s “concise user guide or in-app section.”			
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.100. System NonFunctional Requirements: SO-07-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	SO-07-01		**Type**	**Functional**	**Non-Functional**
Creation:	Oct 30 2025 09:31 PM		**User**	☐	☐
Modification:	Oct 30 2025 09:32 PM		**System**	☐	☒
Description:	The application package shall include an HTML or PDF quick-start guide accessible through the Help menu.		Organizational (sub-type below) Environmental Requirements		
Priority:	Highest	High	✔ Medium	Low	Lowest
This Req. is Engineered From:		UO-07			
Justify why meeting SO-07-01 can contribute to the fulfilment of UO-07					
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.101. System NonFunctional Requirements: SO-07-02

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	SO-07-02		**Type**	**Functional**	**Non-Functional**
Creation:	Oct 30 2025 09:32 PM		**User**	☐	☐
Modification:	Oct 30 2025 09:32 PM		**System**	☐	☒
Description:	Detection accuracy shall remain ≥ 90 % under daylight, cabin light, and low-light conditions.		Organizational (sub-type below) Environmental Requirements		
Priority:	Highest	✔ High	Medium	Low	Lowest
This Req. is Engineered From:		UO-07			
Justify why meeting SO-07-02 can contribute to the fulfilment of UO-07					
Traceability:	Use cases cf.	N/A			
	Test cases cf.	Yet to be completed in test case worksheet!			
Acknowledgment	*Generated from the CapStone Process Management System ©2025*				

Table 4.102. System NonFunctional Requirements: SO-08-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	SO-08-01			**Type**	**Functional**	**Non-Functional**
Creation:	Oct 30 2025 09:33 PM			**User**	☐	☐
Modification:	Oct 30 2025 09:33 PM			**System**	☐	☒
Description:	The system shall function correctly in both portrait and landscape orientations without restarting the process.			Organizational (sub-type below) Environmental Requirements		
Priority:	Highest	✔ High	Medium	Low	Lowest	
This Req. is Engineered From:		UO-08				
Justify why meeting SO-08-01 can contribute to the fulfilment of UO-08						
Traceability:	**Use cases cf.**	N/A				
	Test cases cf.	Yet to be completed in test case worksheet!				
Acknowledgment	*Generated from the CapStone Process Management System ©2025*					

Table 4.103. System NonFunctional Requirements: SO-09-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**					
Requirement #:	SO-09-01			**Type**	**Functional**	**Non-Functional**
Creation:	Oct 30 2025 09:34 PM			**User**	☐	☐
Modification:	Oct 30 2025 09:34 PM			**System**	☐	☒
Description:	All monitoring and alerting features shall remain fully functional offline.			Organizational (sub-type below) Environmental Requirements		
Priority:	Highest	✔ High	Medium	Low	Lowest	
This Req. is Engineered From:		UO-09				
Justify why meeting SO-09-01 can contribute to the fulfilment of UO-09						
Traceability:	**Use cases cf.**	N/A				
	Test cases cf.	Yet to be completed in test case worksheet!				
Acknowledgment	*Generated from the CapStone Process Management System ©2025*					

Table 4.104. System NonFunctional Requirements: SE-01-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SE-01-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:34 PM**		**User**	□	□
Modification:	**Oct 30 2025 09:35 PM**		**System**	□	☒
Description:	At startup, the application shall display a safety disclaimer and disable configuration changes while the vehicle is moving.		**External (sub-type below)** **Legislative Requirements on Safety/Security**		
Priority:	Highest	**✓ High**	Medium	Low	Lowest
This Req. is Engineered From:		**UE-01**			
Justify why meeting SE-01-01 can contribute to the fulfilment of UE-01					
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.105. System NonFunctional Requirements: SE-03-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SE-03-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:37 PM**		**User**	□	□
Modification:	**Oct 30 2025 09:37 PM**		**System**	□	☒
Description:	The face-detection model shall be trained and validated on datasets representing diverse skin tones, facial features, and eyewear.		**External (sub-type below)** **Cultural and Social Requirements**		
Priority:	Highest	**✓ High**	Medium	Low	Lowest
This Req. is Engineered From:		**UE-03**			
Justify why meeting SE-03-01 can contribute to the fulfilment of UE-03					
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.106. System NonFunctional Requirements: SE-04-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SE-04-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:38 PM**		**User**	□	□
Modification:	**Oct 30 2025 09:39 PM**		**System**	□	☒
Description:	The UI shall support English by default and be designed for easy localization using Android string resources.		**External (sub-type below)** **Cultural and Social Requirements**		
Priority:	Highest	High	✔ **Medium**	Low	Lowest
This Req. is Engineered From:		**UE-04**			
Justify why meeting SE-04-01 can contribute to the fulfilment of UE-04					
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.107. System NonFunctional Requirements: SE-05-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SE-05-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:39 PM**		**User**	□	□
Modification:	**Oct 30 2025 09:39 PM**		**System**	□	☒
Description:	The app shall use only stable Android Jetpack APIs (CameraX, AudioManager, Storage) and tested SDK versions.		**External (sub-type below)** **Interoperability Requirements**		
Priority:	Highest	✔ **High**	Medium	Low	Lowest
This Req. is Engineered From:		**UE-05**			
Justify why meeting SE-05-01 can contribute to the fulfilment of UE-05					
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.108. System NonFunctional Requirements: SE-06-01

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**				
Requirement #:	**SE-06-01**		**Type**	**Functional**	**Non-Functional**
Creation:	**Oct 30 2025 09:39 PM**		**User**	☐	☐
Modification:	**Oct 30 2025 09:40 PM**		**System**	☐	☒
Description:	The inference engine shall remain interoperable with Qualcomm QNN and Android NNAPI runtimes for hardware acceleration.		**External (sub-type below)** **Interoperability Requirements**		
Priority:	Highest	**✔ High**	Medium	Low	Lowest
This Req. is Engineered From:		**UE-06**			
Justify why meeting SE-06-01 can contribute to the fulfilment of UE-06					
Traceability:	**Use cases cf.**	**N/A**			
	Test cases cf.	**Yet to be completed in test case worksheet!**			
Acknowledgment	***Generated from the CapStone Process Management System ©2025***				

Table 4.109. Mapping from user requirements to system requirements

Project Name: Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform			
User Requirements		System Requirements	
Req ID	Description	Req ID	Description
UE-01	The app shall include a startup safety disclaimer and disable manual interaction prompts while the vehicle is in motion.	SE-01-01	At startup, the application shall display a safety disclaimer and disable configuration changes while the vehicle is moving.
UE-03	The face-detection model shall be evaluated for consistent accuracy across diverse skin tones, facial structures, and eyewear.	SE-03-01	The face-detection model shall be trained and validated on datasets representing diverse skin tones, facial features, and eyewear.
UE-04	The interface and alert text shall default to English and be structured for easy localization into other languages.	SE-04-01	The UI shall support English by default and be designed for easy localization using Android string resources.
UE-05	The app shall utilize only stable Android Jetpack APIs to ensure device portability.	SE-05-01	The app shall use only stable Android Jetpack APIs (CameraX, AudioManager, Storage) and tested SDK versions.
UE-06	The app shall interoperate seamlessly with both Qualcomm QNN and Android NNAPI runtimes for ML acceleration without requiring code modifications.	SE-06-01	The inference engine shall remain interoperable with Qualcomm QNN and Android NNAPI runtimes for hardware acceleration.
UF-A	As a driver, I want to securely create accounts and log in using a username and password so that I can view my driving history.	SF-A-01	The system shall record instances of detected drowsiness or distraction for each trip, including timestamps
UF-B	As a driver, I want the system to detect signs of drowsiness in real-time on my face and eyes in order to promote driving safety.	SF-B-01	The system shall provide immediate audible and visual alerts when drowsiness or distraction is detected.
		SF-B-02	The system shall detect when the drivers eyes are closed for prolonged periods and trigger alerts.
UF-C	As a driver, I want the system to continuously monitor my face and eyes in real-time to detect signs of distraction in order to promote driving focus.	SF-C-01	The system shall detect when the drivers eyes are looking away from the road for prolonged periods and trigger alerts.
		SF-C-02	The system shall continuously process front-camera frames and estimate head yaw (θ). It shall raise a DISTRACTED event when the absolute yaw angle exceeds a configurable threshold for a sustained, configurable duration.
		SF-C-03	The system shall continuously estimate eye gaze and raise a DISTRACTED event when gaze deviates off the forward road region beyond configurable angular bounds for a sustained, configurable duration.
	As a driver, I should be able to configure	SF-D-01	The system shall provide a settings menu accessible to the driver.

UF-D	settings, such as volume of alerts or sensitivity of detection.	SF-D-02	The system shall allow the driver to adjust the volume of alerts.
		SF-D-03	The system shall allow the driver to adjust the sensitivity level of detection.
UF-E	As a driver, I want to start and stop the monitoring session manually so that I can control when the app analyzes my driving behavior.	SF-E-01	The system shall provide explicit controls to Start and Stop a monitoring session. On Start, it shall request camera permission if needed, initialize CameraX and on-device inference, and begin emitting detection events. On Stop, it shall tear down camera/inference and finalize the current session.
		SF-E-02	When the driver taps Start, the system shall execute a preflight flow that (a) verifies/requests camera permission, (b) verifies camera availability, and (c) initializes the monitoring pipeline only if all checks pass; otherwise it shall surface a clear, actionable error and remain stopped.
UF-F	As a driver, I want to receive real-time visual, audio, or vibration alerts when drowsiness or distraction is detected so that I can regain my attention immediately.	SF-F-02	When the system detects DROWSY or DISTRACTED, it shall emit an alert in ≤ 200 ms from the detection event, updating the UI banner and triggering device audio/vibration per current settings
		SF-F-03	The system shall select active alert modalities (visual, audio, vibration) based on user settings and device capability/state (ringer mode, DND, vibrator present). If a modality is unavailable, it shall fallback to available ones and always present a visual alert.
UF-G	As a driver, I want to acknowledge or snooze an alert to avoid repeated or unnecessary alarms while still staying aware of my condition.	SF-G-01	When the driver taps Acknowledge, the system shall immediately dismiss the current alert (visual/audio/vibration), stop any ongoing alert output, and continue monitoring.
		SF-G-02	When the driver taps Snooze, the system shall suppress alerts of the same type for a configurable cooldown while monitoring continues.
UF-H	As a driver, I want to calibrate the camera at the start of monitoring to ensure my face is properly positioned and lighting is sufficient.	SF-H-01	Before monitoring starts, the system shall present a Calibration flow and block Start until calibration passes or the user explicitly skips. If skipped or failed, the session starts in degraded mode (lower alert confidence) and shows an inline tip.
		SF-H-02	During calibration, the system shall show live guidance overlays (face box, eye markers) and evaluate position and pose stability until pass criteria are met.
		SF-I-01	The system shall provide a Summary screen that displays, for a chosen time range or session: total alerts by type (Drowsy,

UF-I	As a driver, I want to view or export a summary of my recent alerts or sessions to understand my drowsiness and distraction patterns.		Distracted), timestamps, session duration, and average PERCLOS; users can filter by Last Session, Today, 7 Days, or Custom Range.
		SF-I-02	The system shall persist per-session records locally and expose a query API to retrieve summaries for time windows.
UF-J	As a driver, I want assurance that my camera and data stays on my device so that my privacy is protected.	SF-J-01	The system shall perform all camera capture and ML inference on device and shall not transmit frames, embeddings, or detection results to any network endpoint.
		SF-J-02	The system shall store only session metadata and alert events in app-scoped storage; by default, it shall not persist raw images or video frames.
		SF-J-03	The system shall request only the minimum runtime permissions required (e.g., CAMERA) and shall block any background network requests originating from the monitoring module.
UF-K	As a system administrator, I want to deploy and configure the apps across multiple devices using managed Android tools so that updates are consistent and secure.	SF-K-01	The system shall support MDM-driven install/update with staged rollout (e.g., 5% → 25% → 100%), version pinning, and force-install on enrolled devices.
		SF-K-02	When allowed by MDM, the app shall honor policy to auto-grant CAMERA permission at install/update and halt monitoring if policy later revokes it, showing an admin-configured message.
UF-L	As a fleet manager, I want to view aggregated drowsiness and distraction statistics across drivers so that i can identify safety trends.	SF-L-01	The backend shall compute fleet-level KPIs over a selected time window: alerts per driving hour, counts by type (Drowsy/Distracted), sessions with ≥1 alert (%), and time-of-day/day-of-week distributions.
		SF-L-02	The system shall accept anonymized session summaries via an ingestion API or bulk import
		SF-L-03	The analytics service shall support filtering KPIs by organization, driver cohort/vehicle group, device model/app version, and date range.
UO-01	The app shall be developed in Android Studio (Kotlin) and tested on Snapdragon-based devices to validate hardware acceleration.	SO-01-01	Development shall use Android Studio Giraffe + Kotlin 1.9 with Git version control.
		SO-01-02	The codebase shall live in a single authoritative Git repository with protected branches.
UO-02	All source code shall reside in a Git Repository with branching and commit standards documented in the project README.	SO-02-01	CI shall block merges that violate documented branching and commit standards.
		SO-02-02	The repository shall include a README section detailing branching/commit rules and a PR template; CI validates their presence and

			version.
UO-03	Code shall follow Kotlin style conventions, include inline comments, and undergo peer review for clarity and maintainability.	SO-03-01	Enforce Kotlin style via automated linters in CI.
		SO-03-02	Require API docs and meaningful inline comments for non-obvious logic.
		SO-03-03	All code changes must pass peer review focused on clarity and maintainability.
UO-04	Each functional requirement shall be verified by at least one unit or integration test before milestone release.	SO-04-01	Continuous-integration builds shall enforce unit-test coverage ≥ 80 %.
UO-05	The app shall be packaged as an APK targeting Android Level 33 or higher and installable without device root access.	SO-05-01	Release builds shall target API 33 or higher and be installable on Snapdragon 8-series or equivalent hardware.
UO-06	A concise user guide or in app section shall describe calibration, settings, and safety disclaimers.	SO-06-01	The app shall include a built-in Help section (no network required) covering calibration, settings, and safety disclaimers.
		SO-06-02	The guide must clearly explain (a) calibration steps & pass/fail tips, (b) each setting’s effect/range/defaults, (c) safety disclaimers and appropriate us
		SO-06-03	The guide shall meet accessibility basics and support future localization; relevant screens should deep-link to specific guide topics.
UO-07	The system shall maintain detection accuracy under daylight, artificial, and low-light cabin conditions.	SO-07-01	The application package shall include an HTML or PDF quick-start guide accessible through the Help menu.
		SO-07-02	Detection accuracy shall remain ≥ 90 % under daylight, cabin light, and low-light conditions.
UO-08	The app shall operate correctly when the phone is mounted either in landscape or portrait orientation.	SO-08-01	The system shall function correctly in both portrait and landscape orientations without restarting the process.
UO-09	All detection, alerts, and settings shall operate fully without internet connection.	SO-09-01	All monitoring and alerting features shall remain fully functional offline.
UP-01	The app shall present an intuitive, minimal interface that requires no more than two user actions to begin monitoring.	SP-01-01	The system UI shall launch within 2 seconds and maintain a response time < 200 ms for all on-screen interactions.
UP-02	During active monitoring, no manual interaction shall be required.	SP-02-01	The system shall automatically start monitoring when the driver presses “Start Session,” requiring no further manual input.
UP-03	Visual, audio and vibration alerts shall be easily distinguishable and convey meaning without confusing or startling the driver.	SP-03-01	The alert subsystem shall generate visual, audio, and haptic feedback signals that are synchronized within ±50 ms.
UP-04	Interface text size, contrast, and color schemes shall conform to WCAG 2.1 AA accessibility guidelines where feasible on Android	SP-04-01	All UI text and contrast ratios shall conform to WCAG 2.1 AA standards for accessibility.
		SP-05-01	The inference module shall process ≥ 15 frames per second and issue alerts within 200

UP-05	The system shall maintain at least 15 FPS processing speed and alert latency less than or equal to 200 ms from detection to notification.		ms of detection.
		SP-05-02	During continuous operation, average CPU utilization shall not exceed 40 %, and thermal throttling shall be avoided through adaptive frame skipping.
UP-06	Model inference shall execute locally using Snapdragon CPU/GPU/NPU resources while keeping sustained CPU utilization under 40 percent	SP-06-01	Inference must execute locally using Snapdragon CPU/GPU/NPU via QNN/NNAPI, with preferred NPU then GPU, falling back to CPU only if required.
UP-07	The app shall automatically reduce inference frequency if device temperature or battery drain exceeds safe thresholds.	SP-07-01	When device temperature or battery drain exceeds safe thresholds, the app shall automatically reduce inference frequency (FPS) and/or input resolution to lower load.
UP-08	The app shall operate continuously for sessions up to two hours without crash, freeze or data loss	SP-08-01	The monitoring process shall run stably for at least 2 hours without crash, memory leak, or data loss.
UP-09	Upon hardware or model failure, the system shall suspend monitoring and inform the user rather than issue false alerts.	SP-09-01	On hardware or model failure, the system shall halt inference and display a “Safe Mode” warning.
UP-10	All captured data shall remain in app-scoped storage, no external transmission occurs unless explicitly authorized	SP-10-01	All data logs and configuration files shall be stored in app-scoped encrypted storage and deleted upon user request.
UP-11	The system shall request only essential Android permissions (Camera, Audio, Vibration, Storage) and respect user revocation at runtime.	SP-11-01	The application shall respect Android runtime permission revocations and immediately disable camera or microphone access.

Acknowledgment: Generated from the CapStone process management system ©2025

Appendix U: Use Cases

Table 4.1. Use Case Index Table

Project Name: Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform				
Use Case ID	**Use Case Name**	**Level**	**Author**	**Version**
UC-001	Start Monitoring Session	Summary	Chase Tanner	0.1
UC-002	Detect and Alert Driver Drowsiness	Summary	Tarig Elamin	0.1
UC-003	Acknowledge or Snooze Alert	Summary	Tarig Elamin	0.1
UC-004	Adjust Sensitivity and Settings	Summary	Tarig Elamin	0.1
UC-005	Calibrate / Align Camera	Summary	Tarig Elamin	0.1
UC-006	View Alert Summary	Summary	Tarig Elamin	0.1
UC-007	Export Logs	Primary task	Tarig Elamin	0.1
UC-008	Manage App Deployment (System Administrator)	Subfunction	Tarig Elamin	0.1
UC-009	View Aggregated Reports (Fleet Manager)	Summary	Tarig Elamin	0.1
UC-010	Receive Distraction Alert	Summary	Tarig Elamin	0.1
Acknowledgment: Generated from the CapStone process management system ©2025				

Table 4.2. Use Case UC-001

<table>
<tr><th>Project Name:</th><th>Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform</th></tr>
<tr><td><b>Use Case ID:</b></td><td>UC-001</td></tr>
<tr><td><b>Use Case Name:</b></td><td>Start Monitoring Session</td></tr>
<tr><td><b>User Goal:</b></td><td>The driver wants to manually start and stop monitoring a session so they can control when the app analyzes their driving behavior</td></tr>
<tr><td><b>Scope:</b></td><td>Driver Monitoring Sub-System</td></tr>
<tr><td><b>Level:</b></td><td>Summary</td></tr>
<tr><td><b>Relevant User Reqs:</b></td><td>UF-B,UF-E,UF-H</td></tr>
<tr><td><b>Relevant System Reqs:</b></td><td>SF-B-01,SF-B-02,SF-C-01</td></tr>
<tr><td><b>Primary Actor:</b></td><td>Driver (mobile app user)</td></tr>
<tr><td><b>Precondition:</b></td><td>The ALVION app is installed and permissions (camera, audio, vibration) are granted. User is logged in</td></tr>
<tr><td><b>Minimal Guarantee:</b></td><td>If camera permission is denied or initialization fails, the app informs the user and remains idle</td></tr>
<tr><td><b>Success Guarantee:</b></td><td>Monitoring session begins successfully; live interface runs at >= 15 FPS, analyzing eye openness and head orientation.</td></tr>
<tr><td><b>Trigger:</b></td><td>Driver taps "Start Monitoring" on main dashboard screen</td></tr>
<tr><td><b>Success Scenario:</b></td><td>
<table>
<tr><th>Step</th><th>Actions</th></tr>
<tr><td><b>1</b></td><td>The user launches the ALVION application</td></tr>
<tr><td><b>2</b></td><td>The system request camera, audio, and vibration permissions if not already granted</td></tr>
<tr><td><b>3</b></td><td>The user taps "Start Monitoring"</td></tr>
<tr><td><b>4</b></td><td>The system activates the front-facing camera via CameraX</td></tr>
<tr><td><b>5</b></td><td>The system initializes the on-device ML inference engine using Snapdragon CPU/GPU/NPU</td></tr>
<tr><td><b>6</b></td><td>The system calibrates the frame to ensure lighting and positioning are valid</td></tr>
<tr><td><b>7</b></td><td>The system begins continuous real-time analysis of eye openness and head orientation</td></tr>
<tr><td><b>8</b></td><td>The system triggers an alert if drowsiness or distraction is detected</td></tr>
<tr><td><b>9</b></td><td>The user can manually tap "Stop Monitoring" to end the session</td></tr>
<tr><td><b>10</b></td><td>The system records the session timestams and saves the results locally</td></tr>
</table>
</td></tr>
<tr><td><b>Extensions:</b></td><td><b>Branching Scenarios</b></td></tr>
<tr><td><b>3A</b></td><td><b>Condition:</b> Camera permission denied</td></tr>
<tr><td></td><td>
<table>
<tr><th>Step</th><th>Actions</th></tr>
<tr><td><b>1</b></td><td>The system displays error "Camera access is required to begin monitoring". Disable Start button until granted</td></tr>
</table>
</td></tr>
<tr><td><b>3B</b></td><td><b>Condition:</b> Calibration fails (low lighting)</td></tr>
</table>

	Step	**Actions**
	1	The system Prompt user "Lighting insufficient -- please adjust position or brightness".
3C	**Condition:** Hardware overload (temperature threshold exceeded)	
	Step	**Actions**
	1	The system notifies driver "Monitoring paused to cool device".
3D	**Condition:** User revokes permission during session	
	Step	**Actions**
	1	The system suspends monitoring safely and informs the driver
Acknowledgment: Generated from the CapStone process management system ©2025		

Table 4.3. Use Case UC-002

<table>
<tr><th>Project Name:</th><th>Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform</th></tr>
<tr><td>Use Case ID:</td><td>UC-002</td></tr>
<tr><td>Use Case Name:</td><td>Detect and Alert Driver Drowsiness</td></tr>
<tr><td>User Goal:</td><td>The driver wants the system to monitor their face and eyes in real time and warn them immediately if signs of drowsiness are detected, helping them stay alert and safe.</td></tr>
<tr><td>Scope:</td><td>Distracted/Drowsy Driver Detection System (the in-</td></tr>
<tr><td>Level:</td><td>Summary</td></tr>
<tr><td>Relevant User Reqs:</td><td>UF-B</td></tr>
<tr><td>Relevant System Reqs:</td><td>SF-B-01,SF-B-02</td></tr>
<tr><td>Primary Actor:</td><td>Driver</td></tr>
<tr><td>Precondition:</td><td>1- The system is installed and running. 2- The camera is calibrated, and lighting conditions allow detection. 3- Driver monitoring mode is active.</td></tr>
<tr><td>Minimal Guarantee:</td><td>If sensors fail or detection confidence is too low, the system logs the issue and notifies the driver that monitoring accuracy is reduced</td></tr>
<tr><td>Success Guarantee:</td><td>If drowsiness is detected, the system immediately issues audible and visual alerts, prompting the driver to regain focus</td></tr>
<tr><td>Trigger:</td><td>The system detects drowsiness indicators (e.g., eyes closed for prolonged periods or slow blink rate)</td></tr>
<tr><td>Success Scenario:</td><td>
<table>
<tr><th>Step</th><th>Actions</th></tr>
<tr><td>1</td><td>The system continuously monitors the driver’s eyes and facial expressions using the camera.</td></tr>
<tr><td>2</td><td>The system analyzes blink frequency, eyelid closure duration, and gaze direction in real time.</td></tr>
<tr><td>3</td><td>The system the system classifies the state as “drowsy.”</td></tr>
<tr><td>4</td><td>The system the system triggers audible and visual alerts</td></tr>
<tr><td>5</td><td>The user hears/see the alert and adjusts posture, eye focus, or takes a break.</td></tr>
<tr><td>6</td><td>The system monitoring continues once the driver’s state normalizes.</td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>Branching Scenarios</td></tr>
<tr><td>1A</td><td><b>Condition:</b> The Driver continues to show "drowsy" symptoms for a long pried of time</td></tr>
<tr><td></td><td>
<table>
<tr><th>Step</th><th>Actions</th></tr>
<tr><td>1</td><td>The system sends notification to family members and emergency contacts</td></tr>
</table>
</td></tr>
<tr><td colspan="2"><i>Acknowledgment: Generated from the CapStone process management system ©2025</i></td></tr>
</table>

Table 4.4. Use Case UC-003

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform
Use Case ID:	UC-003
Use Case Name:	Acknowledge or Snooze Alert
User Goal:	The driver acknowledges an alert or snoozes further alerts for a short cooldown.
Scope:	Driver Monitoring Sub-System
Level:	Summary
Relevant User Reqs:	UF-D,UF-F,UF-G,UF-I,UF-J
Relevant System Reqs:	SF-A-01,SF-B-01,SF-D-01
Primary Actor:	Driver
Precondition:	A drowsiness or distraction alert has just been displayed.
Minimal Guarantee:	If input is not received, alert auto-dismisses after timeout; monitoring continues.
Success Guarantee:	System records the user action and applies cooldown if snoozed.
Trigger:	Driver taps "Acknowledge" or "Snooze".
Success Scenario:	**Step** **Actions** **1** The system presents actions: Acknowledge | Snooze. **2** The Driver selects an action.
Extensions:	Branching Scenarios
1A	**Condition:** If Acknowledge
	Step **Actions** **1** The system System clears alert UI and logs acknowledgment. **2** The system updates session stats.
1B	**Condition:** If Snooze
	Step **Actions** **1** System starts cooldown timer; suppresses new alerts. **2** The system updates session stats.
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.5. Use Case UC-004

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform
Use Case ID:	UC-004
Use Case Name:	Adjust Sensitivity and Settings
User Goal:	The driver adjusts thresholds (eye-closure time, yaw angle) and alert cooldown/volume; changes persist locally between sessions.
Scope:	Settings Sub-System
Level:	Summary
Relevant User Reqs:	UF-D
Relevant System Reqs:	SF-D-01,SF-D-02,SF-D-03
Primary Actor:	Driver
Precondition:	App is installed; monitoring may be idle or running.
Minimal Guarantee:	Invalid inputs are rejected with guidance; previous values remain.
Success Guarantee:	New settings are validated, saved, and hot-applied to active monitoring.
Trigger:	Driver opens Settings.
Success Scenario:	**Step** **Actions** **1** The user navigates to Settings. **2** The system loads current Settings (thresholds, cooldown, volume) and displays controls. **3** The user edits one or more values (e.g., PERCLOS window, yaw threshold, alert volume/cooldown). **4** The system validates ranges and dependencies. **5** The system persists updates locally and, if monitoring is active, hot-applies them to detection/alerts. **6** The system confirms "Settings saved."
Extensions:	Branching Scenarios
3A	**Condition:** Out-of-range value
	Step **Actions** **1** The system clamps or shows inline error; cannot save until valid.
5A	**Condition:** Restore defaults
	Step **Actions** **1** The user taps Restore; system resets to defaults and saves.
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.6. Use Case UC-005

<table>
<tr><th>Project Name:</th><th>Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform</th></tr>
<tr><td><b>Use Case ID:</b></td><td>UC-005</td></tr>
<tr><td><b>Use Case Name:</b></td><td>Calibrate / Align Camera</td></tr>
<tr><td><b>User Goal:</b></td><td>Align the camera/phone so the face is centered and lighting is sufficient before monitoring.</td></tr>
<tr><td><b>Scope:</b></td><td>Calibration Sub-System</td></tr>
<tr><td><b>Level:</b></td><td>Summary</td></tr>
<tr><td><b>Relevant User Reqs:</b></td><td>UF-H</td></tr>
<tr><td><b>Relevant System Reqs:</b></td><td>SF-H-01,SF-H-02</td></tr>
<tr><td><b>Primary Actor:</b></td><td>Driver</td></tr>
<tr><td><b>Precondition:</b></td><td>Camera permission granted; preview available.</td></tr>
<tr><td><b>Minimal Guarantee:</b></td><td>If calibration fails, user receives guidance and can retry or skip.</td></tr>
<tr><td><b>Success Guarantee:</b></td><td>Calibration passes; app allows monitoring to start (or continues with improved alignment).</td></tr>
<tr><td><b>Trigger:</b></td><td>First use, or system detects poor alignment/lighting; driver taps Start Calibration.</td></tr>
<tr><td><b>Success Scenario:</b></td><td><table>
<tr><th>Step</th><th>Actions</th></tr>
<tr><td><b>1</b></td><td>System shows live preview with guidance overlays (face box/eye markers).</td></tr>
<tr><td><b>2</b></td><td>System evaluates framing stability and pose (e.g., yaw/pitch within bounds) and lighting quality.</td></tr>
<tr><td><b>3</b></td><td>When all criteria hold for the window, system marks Calibration Passed.</td></tr>
<tr><td><b>4</b></td><td>System returns to main screen with Start Monitoring enabled (or auto-continues if already starting).</td></tr>
</table></td></tr>
<tr><td><b>Extensions:</b></td><td><b>Branching Scenarios</b></td></tr>
<tr><td><b>1A</b></td><td><b>Condition:</b> User skips</td></tr>
<tr><td></td><td><table>
<tr><th>Step</th><th>Actions</th></tr>
<tr><td><b>1</b></td><td>The system Start remains allowed but with degraded confidence notice until conditions improve</td></tr>
</table></td></tr>
<tr><td><b>2A</b></td><td><b>Condition:</b> Insufficient lighting/unstable face</td></tr>
<tr><td></td><td><table>
<tr><th>Step</th><th>Actions</th></tr>
<tr><td><b>1</b></td><td>Show tips (increase brightness, hold steady, reposition); remain in calibration.</td></tr>
</table></td></tr>
<tr><td colspan="2"><i><b>Acknowledgment: Generated from the CapStone process management system ©2025</b></i></td></tr>
</table>

Table 4.7. Use Case UC-006

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**
Use Case ID:	UC-006
Use Case Name:	View Alert Summary
User Goal:	After a session, the driver reviews counts by alert type, timeline, duration, and average PERCLOS.
Scope:	Reporting (Local)
Level:	Summary
Relevant User Reqs:	UF-I
Relevant System Reqs:	SF-I-01,SF-I-02
Primary Actor:	
Precondition:	At least one session ended with recorded events.
Minimal Guarantee:	If no events exist, show “No alerts recorded” with session duration.
Success Guarantee:	Summary displays key metrics from local storage and supports time-range filters.
Trigger:	Driver taps Summary.
Success Scenario:	**Step** **Actions** **1** Driver selects Last Session, Today, 7 Days, or Custom. **2** The system loads session data locally and aggregates counts, durations, and PERCLOS. **3** The system renders timeline and KPI cards. **4** The user closes summary or proceeds to Export.
Extensions:	**Branching Scenarios**
2A	**Condition:** Data missing/corrupted
	Step **Actions** **1** The system shows error and offers diagnostics/logs link.
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.8. Use Case UC-007

<table>
<tr><th>Project Name:</th><th colspan="2">Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform</th></tr>
<tr><td>Use Case ID:</td><td colspan="2">UC-007</td></tr>
<tr><td>Use Case Name:</td><td colspan="2">Export Logs</td></tr>
<tr><td>User Goal:</td><td colspan="2">Export anonymized summaries (and optional event logs) for external analysis—offline and consented.</td></tr>
<tr><td>Scope:</td><td colspan="2">Reporting (Local Export)</td></tr>
<tr><td>Level:</td><td colspan="2">Primary task</td></tr>
<tr><td>Relevant User Reqs:</td><td colspan="2">UF-I,UF-J</td></tr>
<tr><td>Relevant System Reqs:</td><td colspan="2">SF-J-03</td></tr>
<tr><td>Primary Actor:</td><td colspan="2">Driver</td></tr>
<tr><td>Precondition:</td><td colspan="2">Local session data exists; user consents to export.</td></tr>
<tr><td>Minimal Guarantee:</td><td colspan="2">If storage permission denied or canceled, no file is written.</td></tr>
<tr><td>Success Guarantee:</td><td colspan="2">A CSV/JSON file is created with aggregated/session fields—no PII or images.</td></tr>
<tr><td>Trigger:</td><td colspan="2">Driver taps Export from Summary or Settings.</td></tr>
<tr><td rowspan="4">Success Scenario:</td><td>Step</td><td>Actions</td></tr>
<tr><td>1</td><td>The system presents consent text and lets the user choose CSV or JSON and a destination.</td></tr>
<tr><td>2</td><td>The system serializes summaries (and optional event rows) and writes to selected location.</td></tr>
<tr><td>3</td><td>The system confirms success and offers Open</td></tr>
<tr><td>Extensions:</td><td colspan="2">Branching Scenarios</td></tr>
<tr><td>1A</td><td colspan="2">Condition: User cancels/denies permission</td></tr>
<tr><td rowspan="2"></td><td>Step</td><td>Actions</td></tr>
<tr><td>1</td><td>The system Abort export; show guidance.</td></tr>
<tr><td colspan="3"><i>Acknowledgment: Generated from the CapStone process management system ©2025</i></td></tr>
</table>

Table 4.9. Use Case UC-008

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**
Use Case ID:	UC-008
Use Case Name:	Manage App Deployment (System Administrator)
User Goal:	Deploy updates, enforce permissions, and apply managed configurations uniformly across devices.
Scope:	Device Management / Deployment
Level:	Subfunction
Relevant User Reqs:	UF-K
Relevant System Reqs:	SF-K-01,SF-K-02
Primary Actor:	System Administrator
Precondition:	Devices are enrolled in enterprise MDM; app is approved in store.
Minimal Guarantee:	If a policy cannot be applied, admin is notified with reason.
Success Guarantee:	Target devices receive the assigned version and enforced settings; status is auditable.
Trigger:	Admin schedules a rollout or pushes a policy refresh
Success Scenario:	**Step** **Actions** **1** Admin selects device group(s) and rollout plan (staged or immediate). **2** MDM distributes the app/update and managed configs (permissions, locked settings). **3** Devices install/update and apply configs automatically. **4** App reflects managed settings and displays “Managed by your organization.” **5** Admin reviews deployment status and audit logs in the console.
Extensions:	**Branching Scenarios**
2A	**Condition:** Policy revokes CAMERA
	Step **Actions** **1** App transitions to IDLE, releases camera, shows policy notice.
3A	**Condition:** Device offline
	Step **Actions** **1** MDM queues install; applies on next check-in.
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.10. Use Case UC-009

Project Name:	Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform
Use Case ID:	UC-009
Use Case Name:	View Aggregated Reports (Fleet Manager)
User Goal:	View aggregated drowsiness/distraction trends across drivers to identify safety issues and inform policies.
Scope:	Fleet Reporting (Future)
Level:	Summary
Relevant User Reqs:	UF-L
Relevant System Reqs:	SF-L-01,SF-L-02,SF-L-03
Primary Actor:	Fleet Manager
Precondition:	Consented, anonymized summaries have been uploaded to an org data store.
Minimal Guarantee:	If data for a range is missing, dashboard shows “incomplete coverage.”
Success Guarantee:	Dashboard shows alerts/hour, type distribution, and time-of-day/week trends with filters and export.
Trigger:	Manager opens Fleet Dashboard and selects filters and date range.
Success Scenario:	**Step** **Actions** **1** The system aggregates data for the selected org/cohort/date range, normalized by monitored hours. **2** The system renders trend charts and KPIs (alerts/hr, type mix, peak hours). **3** The user exports CSV/PDF for sharing.
Extensions:	Branching Scenarios
1A	**Condition:** Small cohort (<5 drivers)
	Step **Actions** **1** The system Suppress display to protect privacy (k-anonymity).
2A	**Condition:** Backend unavailable
	Step **Actions** **1** Serve last-known aggregates with a stale badge.
Acknowledgment: Generated from the CapStone process management system ©2025	

Table 4.11. Use Case UC-010

Project Name:	**Distracted/Drowsy Driver Detection on Snapdragon Mobile Platform**
Use Case ID:	UC-010
Use Case Name:	Receive Distraction Alert
User Goal:	The driver is alerted when their head remains turned away from the forward direction longer than the allowed threshold, so they can refocus attention.
Scope:	Driver Monitoring Sub-System
Level:	Summary
Relevant User Reqs:	UF-C
Relevant System Reqs:	SF-C-02,SF-F-02
Primary Actor:	Driver
Precondition:	Monitoring session is active; camera/inference running; thresholds configured. (See UC-001 preconditions for session start details.)
Minimal Guarantee:	If face/pose confidence is low, the system withholds the alert and logs the reason.
Success Guarantee:	A distraction alert (visual + configured audio/vibration) is issued and recorded with timestamp and metrics.
Trigger:	Head yaw remains beyond the configured threshold for longer than the dwell time.
Success Scenario:	**Step** **Actions** **1** System continuously estimates head yaw from the face-detection model. **2** System accumulates time while |yaw| exceeds the threshold. **3** When dwell time is reached, system creates a DISTRACTED event (with yaw peak, duration, confidence). **4** The System issues a real-time alert (visual banner + audio/vibration per settings). **5** The System logs the event in the current session for later summary/reporting.
Extensions:	**Branching Scenarios**
2A	**Condition:** Low confidence / face not detected
	Step **Actions** **1** The system pause dwell timer; show non-blocking tip (e.g., “reposition/brighten”); no alert.
4A	**Condition:** Alerts snoozed
	Step **Actions** **1** The system suppress during cooldown; log suppression.
Acknowledgment: Generated from the CapStone process management system ©2025	