IFS DriverAlert

Quality Assurance Document

Date: Apr. 03, 2025

Table of Contents

1.0 Introduction

The purpose of this Quality Assurance (QA) document is to make sure that the IFS-DriverAlert system meets its intended functional and non-functional requirements. QA plays a critical role in confirming that our system operates as expected, as it provides consistent and accurate detection of driver drowsiness in real-time.

IFS-DriverAlert is a capstone project for our software systems engineering degree, and it aims to improve road safety by detecting three signs of drowsiness (closed eyes, yawning, and looking away). The system uses a Raspberry Pi 5, NEO-6M GPS module, camera, and speaker. Upon detecting signs of drowsiness, the system plays a progressive audio alert to warn the driver. The GPS module supports Auto mode by enabling detection only when the vehicle is moving above 19 km/h.

This document outlines the QA strategies used to validate the software’s performance, functionality, usability, and safety under various conditions. Also, it provides context for QA activities, documents known limitations, and highlights opportunities for future improvements.

2.0 QA Plan

The QA plan defines the specific functional and non-functional requirements that our system must fulfill to be considered successful. That includes validating the system's ability to detect drowsiness accurately, alert the driver effectively, and operate reliably under different conditions.

2.1 Functional Requirements to Validate:

- The system must detect closed eyes, yawning, and looking away.
- The detection should trigger a progressive audio alert with increasing volume and varied messages based on repeat detections.
- The GPS module must correctly monitor vehicle speed and enable Auto mode only when speed is above 19 km/h.
- In Manual mode, detection should run continuously regardless of GPS data.
- The system must play an audio reminder every 10 minutes if Auto mode is active but no speed is detected.
- The user should be able to toggle the system on/off through a physical switch.

2.2 Non-Functional Requirements to Validate:

- The system must operate offline without needing internet connectivity.
- Detection and alerting must occur with minimal latency (real-time).
- The Raspberry Pi 5 must handle the processing load without crashing or lagging.
- The system must function reliably in different lighting conditions (day, night, dim, etc.).
- No images or video frames should be saved to ensure user privacy.
- The system should be easy to install and require minimal user interaction after setup.

3.0 QA Activities

In order to validate that our system meets its requirements, different types of testing were performed, which included code testing, document-based testing (boundary value, equivalence testing and integration testing), and usability testing. These methods helped verify the system's functionality, integration, and user experience under real-world scenarios.

The sections below outline the QA goals, the main validation tasks we performed, and a record of any notable outcomes or feedback that influenced the project.

3.1 QA Testing Goals

The primary goals of the QA process for the our system are:

- Confirm that the system accurately detects closed eyes, yawning, and looking away.
- Confirm that the progressive audio alert system plays the appropriate message at the right volume level based on repeat detections.
- Validate the effectiveness of both Auto mode (GPS-based) and Manual mode (always on).
- Demonstrate real-time detection performance without noticeable delays.
- Confirm privacy protection by ensuring no image data is stored or transmitted.
- Verify offline capability and system reliability during vehicle movement.

3.2 QA Tasks

- The detection of all three signs in various lighting conditions was tested.
- The detection of all three signs was tested with different drivers (age, gender, different skin, etc).
- The progressive alerts based on repeat detections were verified.
- The GPS-based Auto mode behavior at speeds above and below 19 km/h was tested.
- The system reminder in Auto mode if no speed is detected for 10 minutes was tested (We tried this inside a house where the GPS signal was lost).
- The offline operation and privacy policy (no data storage) was verified.

3.3 QA Records

Design Evolution: Based on user feedback, yawning and looking away detection were added to improve overall effectiveness. The audio alert system was also enhanced to include varied messages and volumes based on user feedback.

Minor Delay Noted: After crossing the 19 km/h threshold in Auto mode, the detection system starts after 4–6 seconds. This is due to system startup time on the Raspberry Pi. Once detection starts, detection runs in real-time and no further issues are found.

Camera Positioning Insight: Testing revealed that if the camera is angled (not parallel to the driver's line of sight), the looking-away detection may produce false positives due to incorrect face angle calculation. To mitigate this, a specific camera setup instruction was added to the user manual, which explained to users how to position the camera straight and parallel to the driver’s view.

No Other Issues: The system performed reliably across all test cases with no additional mitigations required.

4.0 Limitations and Future Considerations

4.1 Current Limitations

Audio Alert Limitations: The effectiveness of the alert system may be reduced if the car’s music is too loud. Also, the system might not be effective for drivers with hearing impairments, because it currently relies solely on audible feedback.

Camera Angle Sensitivity: Looking-away detection relies on calculating the face angle in the frame which is captured through the camera. If the camera is positioned at a tilt or is not parallel to the driver's line of sight, false positives may occur. This issue is addressed in the user manual with detailed setup instructions.

Startup Delay: When the detection system is being called to start, it takes approximately 4–6 seconds to begin detection due to hardware limitations on the Raspberry Pi. Once initialized, detection operates in real time.

4.2 Future Considerations

Adding Haptic Feedback: To improve accessibility for deaf drivers or noisy environments, a small vibration motor could be added to the steering wheel to provide tactile alerts along with the speaker alerts.

Audio Control Integration: Pausing or lowering the car's music volume automatically when an alert is triggered could be a good enhancement for the driver to hear the warning clearly.

5.0 Summary and Conclusion

The Quality Assurance process for the IFS-DriverAlert project was designed to make sure that the system meets its intended goals of detecting drowsiness through closed eyes, yawning, and looking away, and alerting the driver through a progressive audio alerts. Both functional and non-functional requirements were validated through structured testing under various real-world conditions.

Through the QA activities, we confirmed that the system operates reliably in real time, maintains user privacy, functions offline, and handles typical road conditions. Minor limitations, such as camera angle sensitivity and startup delay, were identified and addressed where possible through user instructions.