Name of Testers:

1. Hrushikesh
2. Harshvardhan
3. Asha
4. Shilpa
5. Vinith
6. Jayavardhini

Test Plan for Lane Management Service with diagnostic report with Diagnostic error code analysis

Version: 1.0

Created: 05/21/2024

Last Updated: 05/21/2024

Status: Currently working on the "Diagnostic Error Code Analysis"

Table of Contents

1. Introduction

1.1. Purpose
The purpose of this test plan is to outline the testing strategy for the Lane Management Service (LMS) integrated with a car, ensuring it meets the functional, integration, and performance requirements within the AUTOSAR (AUTomotive Open System ARchitecture) layered architecture.

1.2. Project Overview
The project involves integrating a Lane Management Service into a car's existing architecture. This includes lane departure warning, lane keeping assist, and related diagnostics, all implemented using the AUTOSAR layered architecture.

1.3. Audience
This document is intended for project managers, developers, testers, and stakeholders involved in the development and deployment of the Lane Management Service.

2. Test Strategy

2.1. Test Objectives
- Validate the functional requirements of the Lane Management Service.
- Ensure seamless integration with other vehicle systems.
- Test the system's performance under various conditions.
- Confirm the system's compliance with AUTOSAR standards.

2.2. Test Assumptions
- The hardware components are pre-tested and functioning correctly.
- The test environment mimics the real-world operating conditions of the car.
- Access to all necessary documentation and specifications.

2.3. Test Principles
- Adherence to AUTOSAR standards.
- Comprehensive coverage of all functional and non-functional requirements.
- Systematic and repeatable testing processes.

2.4. Data Approach
- Use of realistic data sets for lane management scenarios and diagnostics.
- Test data should cover a range of use cases, including edge cases.

2.5. Scope and Levels of Testing

2.5.1. Unit Testing
- Application Layer:
  - Verify individual functions of the lane management application.
  - Validate user interface responses to inputs.
- Runtime Environment (RTE):
  - Check RTE configuration and data exchange between application and BSW.
  - Validate task scheduling and timing requirements.
- Basic Software (BSW) Modules:
  - Test communication services (COM, PDU Router).
  - Validate memory services (NvM, MemStack).
  - Check diagnostic services (DEM, DCM).
- Microcontroller Abstraction Layer (MCAL):
  - Verify peripheral drivers (CAN, Ethernet).
  - Test microcontroller-specific configurations.

2.5.2. Integration Testing
- Communication Interfaces:
  - Validate CAN and Ethernet communication between lane management components.
  - Test data exchange between head unit and external devices (e.g., diagnostic tools).
- Sensor Integration:
  - Verify functionality of lane detection sensors and cameras.
  - Check integration with vehicle's braking and steering systems.
- User Interface Integration:
  - Verify display and alerts for lane departure and lane keeping assist.
  - Check responsiveness and accuracy of user controls.

2.5.3. System Testing
- System Boot and Shutdown:
  - Verify system start-up and shutdown sequences.
  - Check for error-free initialization and graceful shutdown.
- Functional Scenarios:
  - Test common use cases (e.g., lane departure warning, lane keeping assist).
  - Validate multi-functionality (simultaneous operation of multiple systems).
- Stress and Load Testing:
  - Assess system performance under high load conditions.
  - Validate system behavior under extended operation.
- Failover and Recovery:
  - Test system response to component failures.
  - Validate recovery mechanisms and data integrity.

2.5.4. User Acceptance Testing (UAT)
- Conduct UAT with real users to ensure the system meets their needs and expectations.
- Collect feedback and make necessary adjustments.

2.5.5. Performance Testing
- Measure response times for user inputs and system commands.
- Assess data transfer rates and communication latency.
- Monitor CPU, memory, and network usage.
- Validate system performance under varying resource availability.

2.5.6. Security Testing
- Ensure the lane management system is secure from external threats.
- Test for vulnerabilities and potential breaches.

2.5.7. Regression Testing
- Re-run previous test cases to ensure new updates do not break existing functionalities.
- Validate bug fixes and new features.

2.6. Test Effort Estimate
- Exploratory Testing: 5 days
- Functional Testing: 15 days
- Integration Testing: 10 days
- System Testing: 12 days
- UAT: 8 days
- Performance Testing: 5 days
- Security Testing: 5 days
- Regression Testing: 8 days

3. Test Acceptance Criteria

- All high and medium severity defects are resolved.
- Functional test cases achieve a pass rate of 95%.
- Integration and system test cases achieve a pass rate of 98%.
- Performance benchmarks for response time, lane detection accuracy, and user interface responsiveness are met.
- Security tests show no vulnerabilities.

4. Test Deliverables

- Test Plan Document
- Test Cases and Test Scripts
- Test Execution Reports
- Defect Logs and Resolution Reports
- UAT Feedback Reports
- Final Test Summary Report

5. Risk Management

- Identify Potential Risks: Hardware failures, software bugs, communication breakdowns.

- Mitigation Strategies:

  - Implement robust error-handling and recovery mechanisms.

  - Conduct thorough pre-testing of individual components.

  - Regularly update and review test plans based on feedback and findings.

  - Allocate additional time for unforeseen issues during test phases.

6. AUTOSAR Layered Architecture Explanation

AUTOSAR (AUTomotive Open System ARchitecture) is a standardized automotive software architecture that provides a framework for developing and integrating complex automotive software systems. It divides the software into several layers to simplify development, enhance modularity, and ensure interoperability.

Application Layer:
- Contains the functional software components that implement specific application functionalities, such as lane management.

Runtime Environment (RTE):
- Acts as a middleware layer, providing communication services between software components and between software components and the Basic Software (BSW) layer.

Basic Software (BSW):
- Includes standardized software modules that provide services for communication, memory management, diagnostic services, and operating system functionality.

Microcontroller Abstraction Layer (MCAL):
- Provides a hardware abstraction layer that allows the higher layers to be hardware-independent. It includes drivers for peripherals like CAN, Ethernet, and other communication interfaces.

7. BOD, UDS, and DoIP Explanation

BOD (Base Operating Diagnostic):
- A foundational diagnostic service that provides basic diagnostic functionalities, such as error code reading and clearing, and basic system health checks.

UDS (Unified Diagnostic Services):
- A set of standard diagnostic services defined in ISO 14229. UDS provides a comprehensive set of diagnostic and communication services, including fault code management, data streaming, and system reset functions.

DoIP (Diagnostics over Internet Protocol):
- A diagnostic protocol defined in ISO 13400 that allows diagnostic communication over IP networks. DoIP facilitates remote diagnostics and software updates by leveraging standard IP networking technologies.

8. Diagnostics of the Defects

User Interface Diagnostics:
- Error Display:
  - Check the diagnostic information displayed on the user interface for clarity and accuracy.
  - Ensure that users are informed of system errors and their severity.
- Troubleshooting Guidance:
  - Validate that the user interface provides troubleshooting steps or suggestions for resolving errors.
  - Ensure the diagnostic interface is user-friendly and accessible.

Diagnostic Logging and Reporting:
- Log Generation:
  - Ensure the system generates detailed diagnostic logs for all detected errors and anomalies.
  - Validate the accuracy and completeness of diagnostic logs.
- Reporting:
  - Test the system's ability to compile and report diagnostic

 information to external systems or maintenance personnel.
  - Verify that the reporting format is consistent with industry standards.

Periodic Self-Checks:
- Automated Diagnostics:
  - Validate the system's ability to perform periodic self-checks and diagnostics.
  - Ensure that these self-checks cover all critical components and functions of the lane management system.
- Alert Generation:
  - Test the system's ability to generate alerts for any issues detected during self-checks.
  - Ensure timely notification of potential problems to the user or maintenance system.

Making Diagnostics Easier with AUTOSAR:
- Standardized Interfaces:
  - AUTOSAR provides standardized diagnostic interfaces and services, which simplifies the integration and interoperability of diagnostic tools.
- Modularity:
  - The layered architecture allows for modular diagnostics, making it easier to isolate and diagnose specific components or functions.
- Scalability:
  - AUTOSAR's scalable framework supports a wide range of diagnostic capabilities, from basic error reporting to advanced remote diagnostics.

9. Diagnostic Error Code Analysis

Error Code Structure:
- Diagnostic trouble codes (DTCs) are structured codes that identify specific issues within the lane management system. These codes follow a standardized format, making it easy to interpret and address errors.

Common DTCs for Lane Management:
- U0010: Communication error between the lane management module and the central control unit.
- C1100: Lane detection sensor malfunction.
- B1240: User interface display error.
- P2560: Lane keeping assist system failure.

Analysis Process:
- Detection:
  - The system continuously monitors for faults and logs DTCs when issues are detected.
- Logging:
  - DTCs are stored in the system's memory and can be accessed through diagnostic tools.
- Reporting:
  - Diagnostic reports compile these DTCs and provide detailed information about the nature and severity of the issues.
- Resolution:
  - Maintenance personnel use the diagnostic reports to identify and resolve issues, ensuring the lane management system operates correctly.

Example:
- If the lane management system detects a malfunction in the lane detection sensor, it logs a DTC such as C1100. This code is then displayed on the user interface, along with troubleshooting steps to resolve the issue. The diagnostic log is generated and can be reviewed by maintenance personnel, who can use the standardized DTC format to quickly understand and fix the problem.

THANK YOU