SOFTWARE ENGINEERING LABORATORY (22IS43) TASK EXECUTION SHEET

Name:Aaditya Rao	**USN: 1NT23IS003**	**Date:25/04/2025**
Lab Activity # / Task #: LA-04	**Document name:SRS**	**Submitted details:**

Project Blueprint: Safety-Critical Software System for Autonomous Vehicle Navigation and Decision-Making

1. Project Overview This project aims to develop a safety-critical software system for controlling the navigation and decision-making of autonomous vehicles while ensuring safety. Given the high level of criticality of the safety aspect in the automotive field/domain, the development process must comply with ISO 26262 standards and should incorporate rigorous verification and validation practices.

2. Selected Software Development Model: V-Model (Verification and Validation Model) The V-Model is best suited for any-purpose safety-critical systems due to the emphasis/stress on testing at each development stage, traceability of requirements, and ensuring compliance with industry standards.

---

3. Development Process, Activities, Milestones, and Deliverables

3.1. Requirements Engineering

- **Activity:** Implementing Capture system, safety, and functional requirements.
- **Deliverables:** System Requirements Specification (SRS), Safety Requirements Specification (SaRS).
- **Verification:** Requirements review and traceability matrix.
- **Milestone:** Requirements Sign-Off.

3.2. System Design

- **Activity:** Definition of high-level architecture, safety mechanisms, and component interactions.
- **Deliverables:** System Architecture Design Document (SADD).
- **Verification:** Reviewing the Design and checking architecture validation against requirements.
- **Milestone:** Approval of System Architecture

3.3. Detailed Design

SOFTWARE ENGINEERING LABORATORY (22IS43) TASK EXECUTION SHEET

Name:Aaditya Rao	USN: 1NT23IS003	Date:25/04/2025
Lab Activity # / Task #: LA-04	Document name:SRS	Submitted details:

- **Activity:** Developing a low-level module specifications.
- **Deliverables:** Software Design Description (SDD).
- **Verification:** Peer reviews, traceability to system design.
- **Milestone:** Detailed Design Completion

3.4. Implementation

- **Activity:** Code the modules based on design specifications.
- **Deliverables:** Source Code, Coding Standards Compliance Report.
- **Verification:** Static code analysis, unit testing.
- **Milestone:** Code Freeze

3.5. Unit Testing

- **Activity:** Verifying each module in isolation.
- **Deliverables:** Unit Test Plan, Unit Test Cases, Test Report.
- **Verification:** Code coverage analysis, results documentation.
- **Milestone:** Unit Testing Complete

3.6. Integration Testing

- **Activity:** Verifications of interactions among the modules.
- **Deliverables:** Integration of Test Plan, Test Cases and Test Report.
- **Verification:** Interface compliance, functional correctness.
- **Milestone:** Getting the Integration Verified

3.7. System Testing

- **Activity:** Completely validating the software system against requirements.
- **Deliverables:** System Test Plan, System Test Cases and the Test Report.
- **Verification:** Functional, performance, stress, and regression testing.
- **Milestone:** System Test Sign-Off

3.8. Acceptance Testing

SOFTWARE ENGINEERING LABORATORY (22IS43) TASK EXECUTION SHEET

Name:Aaditya Rao	USN: 1NT23IS003	Date:25/04/2025
Lab Activity # / Task #: LA-04	Document name:SRS	Submitted details:

- **Activity:** Customer validation of final product.
- **Deliverables:** Acceptance Test Plan, Test Report and the User Manual.
- **Verification:** Final compliance check and getting customer feedback.
- **Milestone:** Acceptance of the product by the customer.

3.9. Maintenance

- **Activity:** Post-deployment support and updates.
- **Deliverables:** Maintenance Logs, Change Requests.
- **Verification:** Regression testing, impact analysis.
- **Milestone:** Post-Deployment Review

---

4. Compliance and Safety Assurance

- Following ISO 26262 for functional safety.
- Use of certified tools for development and testing.
- Performing Hazard and Risk Analysis (HARA).
- Maintain traceability from requirements to implementation.

---

5. Quality Target and Quality Model Justification

Main Quality Target: Meet ISO 26262 ASIL-D standard.

- 99.999% reliability of decision-making logic.
- Zero tolerance for dangerous behavior.
- Full traceability from requirements to code.
- High maintainability for updates.

Cost Assured Product Quality:

- Use model-based development and automated test frameworks.
- V-Model for early defect detection.

SOFTWARE ENGINEERING LABORATORY (22IS43) TASK EXECUTION SHEET

Name:Aaditya Rao	USN: 1NT23IS003	Date:25/04/2025
Lab Activity # / Task #: LA-04	Document name:SRS	Submitted details:

- Certified toolchains (MISRA C, AUTOSAR).
- Unit and integration test coverage to minimize rework.

Quality Model: ISO/IEC 25010 Focus:

- Functional Suitability.
- Reliability.
- Maintainability.
- Safety.
- Performance Efficiency.

Quality Improvement Strategies:

- Continuous Integration & Testing.
- Code Reviews & Pair Programming.
- Formal Verification.
- Simulated & Real-world Testing.
- Retrospective Audits.

---

6. Assurance and Achievement of Defined Quality Targets

1. Safety Assurance (ASIL-D Compliance)

- Conduct Hazard Analysis and Risk Assessment (HARA)
- Implement safety mechanisms (redundancy, fail-safes)
- Develop safety cases per ISO 26262

2. Requirements Traceability

- Use traceability matrix from requirements to tests
- Apply requirements tools (e.g., IBM DOORS)
- Conduct trace reviews

3. Verification & Validation Activities

SOFTWARE ENGINEERING LABORATORY (22IS43) TASK EXECUTION SHEET

Name:Aaditya Rao	USN: 1NT23IS003	Date:25/04/2025
Lab Activity # / Task #: LA-04	Document name:SRS	Submitted details:

- Apply V-Model at each phase
- Use static analysis tools (Polyspace, QAC, SonarQube)
- Perform complete test coverage (unit, integration, system)

4. Continuous Quality Control

- CI/CD pipelines
- Code coverage (100% MC/DC for safety components)
- Monitor metrics (complexity, maintainability index)

5. Peer Review and Inspections

- Design/code reviews with safety-focused checklists
- Pair programming for critical logic

6. Simulated and Real-World Testing

- Use HIL setups and simulators
- Perform stress, boundary, and environmental testing

7. Post-Deployment Quality Control

- Capture real-time feedback and logs
- Conduct root cause analysis (RCA)
- Apply OTA updates with regression checks

---

7. Software Requirements Specification (SRS) – Based on IEEE 830

1. Introduction

- **Purpose:** Define software specs for autonomous driving functions under ISO 26262.
- **Scope:** Route planning, obstacle detection, emergency actions
- **Definitions:** ASIL-D, HARA, HIL, ECU
- **References:** ISO 26262, IEEE 830, AUTOSAR, Design Docs
- **Overview:** Structured to guide full lifecycle dev and testing

SOFTWARE ENGINEERING LABORATORY (22IS43) TASK EXECUTION SHEET

Name:Aaditya Rao	USN: 1NT23IS003	Date:25/04/2025
Lab Activity # / Task #: LA-04	Document name:SRS	Submitted details:

2. Overall Description

- **Product Perspective:** Works with ADAS and embedded ECUs
- **Product Functions:** Route planning, emergency logic, real-time obstacle avoidance
- **User Characteristics:** Developers, testers, regulatory bodies
- **Constraints:** Real-time execution, MISRA C compliance, ISO 26262

3. Specific Requirements

- **FR-1:** Detection of obstacles within 20m using RADAR and Camera.
- **FR-2:** Computaton of safe route in 100 ms after route change
- **FR-3:** Should enter fail-safe within 50 ms after fault detection
- **Performance:** <100 ms latency
- **Safety:** ASIL-D safety recovery
- **Interfaces:** CAN, Ethernet, LIDAR, GPS, camera
- **Maintainability:** Modular code, portable to multiple MCUs

4. Traceability Matrix

Req. ID	Description	Source	Test Method
FR-1	Obstacle detection	Stakeholder	Simulation
FR-2	Route decision	Use Case 3	HIL Testing
FR-3	Fail-safe mode	Safety Workshop	Fault Injection

5. V&V Plan

- Unit Testing → correctness
- Integration Testing → interfaces
- System Testing → end-to-end logic
- Acceptance Testing → regulatory approval

6. Appendices

SOFTWARE ENGINEERING LABORATORY (22IS43) TASK EXECUTION SHEET

Name:Aaditya Rao	USN: 1NT23IS003	Date:25/04/2025
Lab Activity # / Task #: LA-04	Document name:SRS	Submitted details:

- Glossary
- Use Case Diagrams
- FMEA and HARA Summary
- Supporting Scenarios

---

8. Requirements Validation Process

In order to ensure that all the requirements will be meeting the stakeholder's needs and ensure the compliance with the safety standards, these following steps will be taken in order to validate the Software Requirements Specification (SRS):

Step 1: Identification of Stakholders

- Identifying all key stakeholders. For example :
    - Automotive engineers,
    - Safety compliance officers,
    - Software developers,
    - End-users (fleet operators),
    - Regulatory authorities.

Step 2: Requirement Review Sessions

- Conducting of regular formal review meetings involving the stakeholders.
- Walk through each functional and non-functional requirement.
- Use checklists to verify the following :
    - Consistency
    - Completeness
    - Feasibility
    - Testability
    - Compliance with ISO 26262

Step 3: Use Case Validation

SOFTWARE ENGINEERING LABORATORY (22IS43) TASK EXECUTION SHEET

Name:Aaditya Rao	USN: 1NT23IS003	Date:25/04/2025
Lab Activity # / Task #: LA-04	Document name:SRS	Submitted details:

- Present documented use cases and scenarios which are derived/provided by stakeholder input.
- Validate that requirements align with expected system behavior in real-world contexts and scenarios.

Step 4: Traceability Verification

- Demonstrate traceability from each requirement to:
  - Original stakeholder input
  - Design components
  - Corresponding test cases

Step 5: Simulated Demonstrations

- Use of early simulation models or prototypes to visualize the requirement implementation.
- Gather feedback from stakeholders on functional adequacy and safety response.

Step 6: Formal Sign-Off

- After reviews and adjustments, get formal validation and sign-off from all key stakeholders.

Step 7: Documentation and Updates

- Record validation feedback, decisions, and sign-offs.
- Update the SRS document to reflect any changes or clarifications post-validation.

SOFTWARE ENGINEERING LABORATORY (22IS43) TASK EXECUTION SHEET

Name:Aaditya Rao	**USN: 1NT23IS003**	**Date:25/04/2025**
Lab Activity # / Task #: LA-04	**Document name:SRS**	**Submitted details:**