

UNIVERSITY OF BUEA



REPUBLIC OF CAMEROON

PEACE-WORK-FATHERLAND

P.O. Box 63,  
Buea, South West Region  
CAMEROON  
Tel : (237) 3332 21 34/3332 26 90  
Fax: (237) 3332 22 72

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF COMPUTER ENGINEERING**

## **Software Requirements Specification (SRS)**

**Project Title:** Car Fault Diagnosis Mobile Application

**By Group 2:**

**BERINYUY CLETUS FE21A148  
EFUETLATEH JOAN FE22A197  
ETAPE NGABE FE22A210  
ETIENDEM PEARL FE22A211  
TATA GLEN FE22A309**

**2024/2025 Academic Year**

## Table of Contents

1	Introduction .....	3
1.1	Purpose .....	3
1.2	Scope .....	3
1.3	Definitions, Acronyms, and Abbreviations .....	3
2	Overall Description .....	3
2.1	Product Perspective .....	3
2.2	Product Features .....	3
2.3	User Characteristics .....	3
2.4	Constraints .....	4
2.5	Assumptions and Dependencies .....	4
3	Specific Requirements .....	4
3.1	Functional Requirements .....	4
3.2	Non-Functional Requirements .....	6
4	External Interface Requirements .....	7
4.1	User Interface .....	7
4.2	Hardware Interface .....	7
5	System Features .....	7
6	Other Requirements .....	8
7	Conclusion .....	8

# 1 Introduction

## 1.1 Purpose

This document outlines the software requirements for a mobile application that diagnoses car faults based on dashboard warning lights and engine sounds. It serves as a reference for developers, testers, and stakeholders.

## 1.2 Scope

The app enables car users to detect potential issues with their vehicle using image and sound input. It provides real-time feedback, suggestions for corrective action, and historical records. Target users include new and experienced drivers, as well as mechanics.

## 1.3 Definitions, Acronyms, and Abbreviations

- **SRS:** Software Requirements Specification
- **UI:** User Interface
- **DB:** Database
- **UX:** User Experience

# 2 Overall Description

## 2.1 Product Perspective

This application is standalone, but future updates may allow integration with cloud services or external diagnostic devices.

## 2.2 Product Features

- Scan dashboard lights using a phone camera.
- Record and analyze engine sounds.
- Provide fault diagnosis and recommendations.
- Save diagnostic history.
- Collect feedback from users.

## 2.3 User Characteristics

- Car owners/drivers with basic smartphone skills
- Mechanics

- Users aged 18–65 with varying tech experience

## 2.4 Constraints

- Android-only (minimum version 8.0)
- Limited access to external hardware (e.g., OBD scanners)
- Privacy concerns: must minimize data collection and avoid cloud uploads unless permitted.

## 2.5 Assumptions and Dependencies

- Users have access to a smartphone with camera and microphone.
- Data used for training the diagnostic system is reliable.

# 3 Specific Requirements

## 3.1 Functional Requirements

These define what the system should do, and the core functionalities that the app must support to meet user expectations:

### 1. User Registration (Sign-Up)

- The system shall allow users (car owners and mechanics) to register using their name and password.
- The system shall display appropriate error messages for invalid inputs.

### 2. User Authentication (Login)

- The system shall allow registered users to log in using their credentials.

### 3. Dashboard Indicator Scanner

- Allow users to scan dashboard lights using their phone camera.
- Use computer vision to recognize and classify the lights.

### 4. Engine Sound Analysis

- Record engine sounds through the phone's microphone.

- Analyze sounds using AI to detect patterns linked to faults (e.g., knocking, squealing).

## **5. Fault Detection and Interpretation**

- Display the meaning of each detected symbol/sound
- Include possible causes and the urgency of the issue

## **6. Multiple Fault Detection**

- Recognize multiple warning lights in a single scan
- Provide maintenance tips.

## **7. Recommendations and Repairs**

- Offer suggested repairs or next steps.
- Provide maintenance tips.

## **8. Video Tutorials Integration**

- Display embedded or linked YouTube videos from certified experts.

## **9. Offline and Online Modes**

- Basic features (e.g., recognition of common lights/sounds) should work offline.
- Online mode should allow access to updated fault databases and videos.

## **10. User Interface**

- Simple and intuitive UI.
- Allow users to navigate diagnostics and solutions easily.

## **11. Mechanic Contact Retrieval**

- The system shall retrieve a list of the nearest available mechanics based on the car's current location.

## **12. History Log**

- The app shall store previous diagnosis records for offline viewing.

## **13. Survey and Feedback Form**

- The app shall include a feedback system to gather user responses for improvement.

## 3.2 Non-Functional Requirements

These describe how the system should perform and the quality attributes it must meet:

### 1. Performance

- Fast image and sound processing with minimal lag.
- Quick diagnosis within seconds.

### 2. Accuracy

- High accuracy in recognition of warning lights and sound anomalies.
- Continuous model training to improve predictions.

### 3. Scalability

- Ability to expand the database of sounds, lights, and issues.

### 4. Security

- Protect user data such as recordings and vehicle history.
- Adhere to privacy policies and permissions.

### 5. Reliability and Availability

- Ensure the app works under varying phone conditions (e.g., low battery, poor lighting, poor connectivity).
- Minimal crashes or bugs.

### 6. Compatibility

- Compatible with a wide range of Android and iOS devices.

### 7. Maintainability

- Code should be modular and easily updatable.
- AI models should be upgradable.

## 8. Usability

- The app should be intuitive and easy to navigate, even for users with minimal tech skills.
- Language support or visual aids for better understanding.

# 4 External Interface Requirements

## 4.1 User Interface

- Home screen with navigation to diagnosis and history
- Camera and sound input interfaces
- Result display with readable icons and suggestions
- Feedback form

## 4.2 Hardware Interface

- Camera (to scan dashboard lights)
- Microphone (to record engine sounds)
- Local device storage

# 5 System Features

Feature	Description
Dashboard Light Scanner	Uses the phone camera to identify warning symbols
Engine Sound Analyzer	Uses sound pattern to detect specific issues
Fault Report	Summarizes faults and suggests possible fixes
History Log	Keeps previous reports for future reference
Feedback	Allows users to send suggestions and comments

## 6 Other Requirements

- The system should be scalable to support future modules like OBD scanner integration.
- App performance must remain optimal on budget smartphones.
- Accessibility features (voice prompts, large icons) are encouraged.

## 7 Conclusion

This Software Requirements Specification (SRS) serves as a comprehensive blueprint for the development of the Car Fault Diagnosis Mobile Application. It outlines the essential functional and non-functional requirements, user interactions, system features, and design constraints that will guide the design, implementation, and testing phases of the project. By providing a clear and structured representation of user needs and system expectations, the SRS ensures that the development team and stakeholders are aligned on the system objectives. As development progresses, this document will be revisited and refined to accommodate any changes or discoveries, ensuring that the final product is both effective and user-centric.