REPUBLIQUE DU CAMEROON
PAIX-Travail-Patrie
MINISTRE DE
L'ENSEIGNEMENT SUPERIEUR
FACULTE D'INGINERIE
ET TECHGNOLOGIE

REPUBLIC OF CAMEROON
Peace-Work-Fatherland
MINISTER OF HIGHER EDUCATION

FACULTY OF ENGINEERING AND TECHNOLOGY

Software Requirements Specification (SRS)

Project: AI-based Car Fault Diagnosis Mobile Application (CarDiagApp)

Version: 1.1
Status: Approved

Prepared by: Group 10

ABANDA SERGIO	**FE22A131**
ABILATEZIE VIN-WILSON ANU	**FE22A132**
AGBOR EMMANUEL NCHENGE	**FE22A138**
ASHLEY TAN WACHE	**FE22A154**
ASHU DESLEY	**FE22A155**

CEF440: INTERNET PROGRAMMING AND MOBILE PROGRAMMING

INSTRUCTOR: Dr. VALERY NKEMENI

Date: May 19, 2025

Table of Contents

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document describes the functional and non-functional requirements for an AI-based mobile application that enables car users to diagnose faults using dashboard warning light recognition and engine sound analysis. It serves as the agreement between stakeholders and the development team and provides a basis for estimating costs, schedules, and validation criteria.

1.2 Scope

The application ("CarDiagApp") will allow users to:

- Register and login with role-based access (car users, administrators)
- Capture and interpret dashboard warning lights via camera
- Record and classify engine sounds via microphone
- Receive automated explanations, repair suggestions, maintenance alerts, and video tutorials
- Track maintenance history and receive reminders based on urgency levels
- Operate offline for core diagnostics and online for updated data
- Locate nearby mechanics using Google Maps (optional location sharing)
- Automatically save diagnostic reports for tracking purposes

The application will not:

- Replace professional diagnostic tools or mechanic expertise
- Interface directly with vehicle onboard computers (no OBD-II connection)
- Provide real-time monitoring while driving
- Guarantee 100% accuracy in all diagnostic scenarios
- Store car model or property information

1.3 Definitions, Acronyms, and Abbreviations

- **App:** The CarDiagApp mobile application
- **ML:** Machine Learning
- **UI/UX:** User Interface / User Experience

- **MVP:** Minimum Viable Product
- **OBD-II:** On-Board Diagnostics II (vehicle self-diagnostic standard)
- **CV:** Computer Vision
- **PII:** Personally Identifiable Information
- **API:** Application Programming Interface
- **2FA:** Two-Factor Authentication
- **RBAC:** Role-Based Access Control

1.4 References

- IEEE Std 830-1998: IEEE Recommended Practice for SRS
- Identify_Issues and Priority.docx: Requirements analysis and prioritization
- GROUP_10_FINAL_REPORT_TASK2.pdf: Report on requirement gathering
- ISO/IEC 25010:2011: Systems and software Quality Requirements and Evaluation (SQuaRE)
- ISO 26262: Road vehicles - Functional safety standard
- OWASP Mobile Application Security Verification Standard (MASVS)

1.5 Overview

The remainder of this document covers:

- Section 2: Overall description of the product, including perspective, functions, user characteristics, constraints, and assumptions
- Section 3: Specific requirements, including functional, interface, and non-functional requirements
- Section 4: System features with detailed descriptions of core functionality
- Section 5: Validation methods and risk mitigation strategies
- Section 6: Appendices with supplementary information

2. Overall Description

2.1 Product Perspective

CarDiagApp is a standalone mobile application designed for iOS and Android platforms using the Flutter cross-platform framework. It leverages on-device ML models (TensorFlow Lite) for image and audio classification, with optional cloud connectivity for data updates and enhanced features.

The application relates to existing products in the following ways:

- Unlike professional OBD-II scanners, it doesn't require physical connection to the vehicle
- Compared to existing automotive apps, it uniquely combines visual and audio analysis
- Integrates with third-party content platforms (YouTube) for educational materials
- Incorporates Google Maps API for nearby mechanic location services

2.2 Product Functions

- **User Authentication:** Registration, login, and profile management with role-based access
- **Dashboard Light Scanner:** Detect and classify multiple warning lights in a single image
- **Engine Sound Analyzer:** Record, preprocess, and classify engine noises (knocking, squealing, hissing, etc.)
- **Diagnosis Report:** Provide cause, urgency level, repair tips, and video links
- **Maintenance Tracker:** Log past diagnostics and schedule reminders
- **Knowledge Base:** Access to common car problems and solutions
- **Offline Mode:** Core features without internet; **Online Mode:** Fetch latest models and tutorials
- **Nearby Mechanics Locator:** Find mechanics in the vicinity using Google Maps

2.3 User Classes and Characteristics

- **Primary: Car Users (Non-Technical)**
    - Demographics: Ages 18-65, varying technical expertise
    - Goals: Quick diagnosis, basic understanding of car issues
    - Technical expertise: Limited automotive knowledge
    - Usage frequency: Occasional, when warning lights appear
    - Access level: Standard diagnostic and maintenance features

- **Secondary: Mechanics (as Car Users)**
    - Goals: Quick pre-checks, client education tool
    - Technical expertise: Professional automotive knowledge
    - Usage frequency: Regular, multiple vehicles
    - Access level: Same diagnostic features as regular car users
- **Tertiary: Content Creators**
    - Goals: Provide and manage tutorial content
    - Technical expertise: Automotive expertise, content creation skills
- **System Administrators**
    - Goals: Manage backend, update models, monitor performance
    - Technical expertise: IT/ML expertise
    - Usage frequency: Regular maintenance and monitoring
    - Access level: Full system access, user management, analytics dashboard

2.4 Operating Environment

- **Mobile Devices:**
    - Android 8.0 (API level 26) or higher
    - iOS 12.0 or higher
    - Minimum 2GB RAM, 100MB free storage
    - Camera with minimum 5MP resolution
    - Microphone with noise cancellation capabilities (preferred)
    - GPS for location services (optional)
- **Network:**
    - Internet connectivity for online features (Wi-Fi or cellular)
    - Offline capability for core features
- **Backend Services:**
    - Cloud-based user authentication system
    - Cloud-based model training infrastructure
    - Content delivery network for tutorials and model updates
    - Secure database for user profiles and preferences
    - Google Maps API integration

2.5 Design and Implementation Constraints

- **Technical Constraints:**
  - On-device ML model size must be <10MB
  - Use TensorFlow Lite for ML inference
  - Flutter framework for cross-platform development
  - Latency: Diagnostic feedback within 5 seconds
  - Battery usage: <5% per diagnostic session
- **Business Constraints:**
  - Comply with privacy regulations: no personal data storage beyond app scope
  - No advertisements in core diagnostic functions
  - App size not to exceed 50MB for initial download
- **Regulatory Constraints:**
  - Must include disclaimer about diagnostic accuracy
  - Must comply with data protection regulations (GDPR, CCPA)
  - Cannot claim to replace professional diagnosis
  - Authentication must comply with industry security standards
  - Location services must be optional with clear user consent

2.6 User Documentation

The following user documentation will be provided:

- In-app tutorial for first-time users
- FAQ section covering common use cases
- Online help center with troubleshooting guides
- Video demonstrations for key features
- Tooltips for interface elements
- Role-specific guidance for different user types
- Audio explanation of diagnosis results (under consideration)

2.7 Assumptions and Dependencies

- Users grant camera and microphone permissions
- Access to YouTube API for tutorial links

- Availability of labeled audio and image datasets for model training
- Users have basic familiarity with smartphone operation
- Adequate lighting conditions for dashboard photo capture
- Vehicle engine accessible for sound recording
- Secure authentication service availability
- Google Maps API availability for nearby mechanic location

3. Specific Requirements

3.1 Functional Requirements

ID	Requirement	Priority	Source
FR1	The app shall provide user registration functionality with email verification.	Must-Have	Security Standards
FR2	The app shall support role-based user authentication (car user and administrator).	Must-Have	Project Revision
FR3	The app shall allow users to recover forgotten passwords securely.	Must-Have	Security Standards
FR4	The app shall maintain user sessions with appropriate timeout controls.	Must-Have	Security Standards
FR5	The app shall allow users to capture dashboard images and detect warning light icons.	Must-Have	Survey, Interviews
FR6	The app shall interpret each detected icon and display its meaning and urgency level.	Must-Have	Survey, Expert Interviews
FR7	The app shall record engine sound clips up to 10 seconds in length.	Must-Have	Survey, Interviews

ID	Requirement	Priority	Source
FR8	The app shall classify recorded engine sounds into predefined categories (e.g., knocking, squealing).	Must-Have	Requirements Analysis
FR9	The app shall provide text-based explanations and repair suggestions for detected faults.	Must-Have	Survey, Interviews
FR10	The app shall include links to video tutorials relevant to the diagnosed issue.	Should-Have	Survey, User Feedback
FR11	The app shall store past diagnostics locally with timestamps and results.	Should-Have	Survey, Interviews
FR12	The app shall send maintenance reminders based on fault urgency (red: immediate alert, yellow: 1-hour alert, monitoring: 2-hour alert).	Should-Have	Requirements Analysis
FR13	The app shall operate offline for image and audio diagnostics using preloaded models.	Must-Have	Survey, Interviews
FR14	The app shall fetch updated fault databases and tutorial metadata when online.	Should-Have	Survey, Interviews
FR15	The app shall detect multiple warning lights simultaneously in a single image.	Must-Have	Project Description
FR16	The app shall estimate maintenance urgency on a scale (immediate, soon, monitoring).	Must-Have	Project Description
FR17	The app shall allow users to share diagnostic reports.	Could-Have	User Feedback

ID	Requirement	Priority	Source
FR18	The app shall provide administrators with user management capabilities.	Must-Have	Project Revision
FR19	The app shall allow users to manage their profile information and preferences.	Should-Have	User Feedback
FR20	The app shall automatically save diagnostic reports for tracking purposes.	Must-Have	Project Revision
FR21	The app shall enable users to locate nearby mechanics using Google Maps.	Should-Have	Project Revision
FR22	The app shall make location sharing optional for users.	Must-Have	Project Revision

3.2 External Interface Requirements

3.2.1 User Interfaces

- **Authentication Screens:**
  - Registration form with email verification
  - Login screen with forgot password option
  - Profile management screen
- **Home Screen:**
  - Buttons for "Scan Lights" and "Record Sound"
  - Quick access to history and settings
  - User-specific features based on role
  - Nearby mechanic locator option
- **Dashboard Scanner Screen:**
  - Camera viewfinder with guide overlay
  - Capture button and flash control

    - Preview and retake options
- **Sound Recorder Screen:**
    - Recording button with timer
    - Sound level visualization
    - Instructions for optimal recording
- **Results Screen:**
    - Displays identified warning icons with labels
    - Sound classification results
    - Explanations, urgency rating, and tutorial links
    - Share options
    - Option to locate nearby mechanics
- **History Screen:**
    - List of past diagnostics with date/time and summary
    - Filtering and search functionality
    - Option to compare multiple diagnoses
- **Admin Dashboard:**
    - User management interface
    - System performance metrics
    - Model update controls
- **Settings Screen:**
    - Notification preferences
    - Data usage controls
    - Authentication settings
    - Language (French / English)
    - Location sharing preferences

3.2.2 Hardware Interfaces

- **Camera:**
    - Access to rear-facing camera with autofocus
    - Flash control for low-light conditions
    - Minimum 5MP resolution

- **Microphone:**
    - Access to device microphone
    - Support for external microphones
    - Audio sampling at 44.1kHz, 16-bit depth
- **Storage:**
    - Access to local storage for saving diagnostic history
    - Cache management for model data
    - Secure storage for authentication tokens
- **Display:**
    - Support for various screen sizes and resolutions
    - Landscape and portrait orientation support
    - Accessibility considerations for text size and contrast
- **GPS/Location Services:**
    - Access to device location for mechanic locator feature
    - Support for both high and low-accuracy location modes

3.2.3 Software Interfaces

- **Authentication Service:**
    - Firebase Authentication
    - OAuth 2.0 integration for social logins (optional)
    - Token-based authentication system
- **TensorFlow Lite:**
    - On-device ML inference using .tflite models
    - Model version tracking and updates
- **YouTube Data API:**
    - Tutorial video retrieval and embedding
    - Search functionality based on diagnostic results
- **Google Maps API:**
    - Nearby mechanic search functionality
    - Distance calculation and routing
- **Local Database:**

- SQLite for diagnostic history storage
- Secure storage for user credentials

- **Push Notification Service:**
  - Firebase Cloud Messaging for maintenance reminders
  - Authentication alerts
  - Urgency-based notification scheduling

3.2.4 Communications Interfaces

- **Network:**
  - HTTPS/TLS for secure data fetch and API calls
  - REST API for backend services
  - Secure communication for authentication
- **File Transfer:**
  - Secure download of model updates
  - Upload of anonymized feedback data (with user consent)
  - Encrypted data transmission

3.3 Non-Functional Requirements

3.3.1 Performance Requirements

- **Response Time:**
  - Maximum 5 seconds for image classification
  - Maximum 7 seconds for audio classification
  - Maximum 2 second for UI transitions
  - Authentication processes complete within 3 seconds
  - Nearby mechanic search results within 5 seconds
- **Startup Time:**
  - App launches within 5 seconds on supported devices
  - Models load within 5 seconds after launch
  - Authentication state verification within 2 second
- **Throughput:**
  - Support for up to 50 diagnostic sessions per day per user
  - Database queries complete in <100ms

- Handle 100+ concurrent authentication requests

- **Resource Utilization:**
  - Memory usage <200MB during operation
  - CPU usage <30% during analysis
  - Battery consumption <1% during standby
  - GPS usage only when actively searching for mechanics

3.3.2 Reliability Requirements

- **Availability:**
  - 99% uptime for online features
  - 100% availability for offline core features
  - Authentication services available 99.9% of time
- **MTBF (Mean Time Between Failures):**
  - Less than 5 crash per 1000 user sessions
  - Authentication system failure rate <0.5%
- **Error Handling:**
  - Graceful message if camera/mic/location access is denied
  - Appropriate fallbacks for network failures
  - User-friendly error messages with recovery options
  - Authentication failure handling with clear user guidance
- **Recovery:**
  - Auto-save partial diagnostic data during system interruptions
  - Automatic retry for failed network operations
  - Session recovery after unexpected termination

3.3.3 Usability Requirements

- **UI Complexity:**
  - Suitable for users with no technical background
  - Maximum 3 taps to reach any main function
  - Consistent design language and navigation patterns
  - Clear role-specific interfaces
- **Learnability:**

- First-time users can complete registration in <2 minutes
- First-time users can complete a diagnosis in <3 minutes
- Interactive onboarding tutorial for new users
- Role-specific guidance for different user types

- **Accessibility:**
  - Support for screen readers and large fonts
  - Color schemes with adequate contrast ratios
  - Voice commands for key functions
  - Compliance with WCAG 2.1 AA standards
  - Optional audio explanations of diagnostic results

3.3.4 Security Requirements

- **Authentication:**
  - Multi-factor authentication option
  - Password complexity requirements
  - Account lockout after multiple failed attempts
  - Secure credential storage
- **Data Privacy:**
  - Audio/image data processed locally; no PII stored or transmitted
  - Anonymized usage data only with explicit opt-in
  - Secure storage of diagnostic history
  - User data segregation and access controls
  - Optional location sharing with clear consent mechanism
- **Permissions:**
  - Explicit runtime requests for camera, microphone, and location
  - Granular permission management
  - Clear explanation of permission usage
  - Role-based access control for features
- **Data Integrity:**
  - Checksum verification for model updates
  - Prevention of diagnostic history tampering

- Audit trails for administrative actions

3.3.5 Maintainability and Portability

- **Codebase Structure:**
  - Modular design in Flutter
  - Separation of UI, business logic, and data layers
  - Comprehensive code documentation
  - Authentication module isolation
- **Platform Support:**
  - iOS and Android with identical feature sets
  - Tablet optimization for larger screens
  - Consistent authentication experience across platforms
- **Upgradability:**
  - Support for over-the-air model updates
  - Database migration strategy for version updates
  - Authentication system upgradable without user disruption

3.3.6 Quality Attributes

- **Accuracy:**
  - ≥90% diagnostic accuracy for common warning lights
  - ≥85% accuracy for engine sound classification
  - False positive rate <5%
  - Zero authentication false positives
- **Efficiency:**
  - ≤5s total response time for complete diagnosis
  - Battery consumption <1% per diagnostic session
  - Data usage <1MB per online session (excluding video playback)
  - Authentication tokens with appropriate expiration periods
  - Location services used only when needed
- **Privacy:**
  - User data anonymized and secure
  - No tracking of user location or behavior without consent

    - Compliance with global privacy regulations
    - Separation of authentication and diagnostic data
- **Data Usage:**
    - Optimized for low bandwidth environments
    - Data compression for model updates
    - Wi-Fi only option for large downloads
    - Minimal data usage for authentication processes

4. System Features

4.1 User Authentication System

4.1.1 Description

The user authentication system provides secure registration, login, and role-based access control for different user types (car users and administrators), ensuring appropriate feature access and data security.

4.1.2 Functional Requirements

- FR1: User registration with email verification
- FR2: Role-based authentication
- FR3: Password recovery
- FR4: Session management
- FR18: Administrator user management
- FR19: Profile management

4.1.3 Technical Implementation

- Firebase Authentication or equivalent backend service
- Token-based authentication
- Secure credential storage using industry-standard encryption
- Role-based access control (RBAC) system
- Email verification workflow
- Password policy enforcement

4.1.4 User Workflow

1. **Registration:**
    - User selects "Register" from login screen
    - User provides email, password, and selects role
    - Email verification sent to confirm address
    - User confirms email and completes profile setup
2. **Login:**
    - User enters credentials on login screen
    - System authenticates and determines appropriate role
    - User directed to role-specific home screen
3. **Password Recovery:**
    - User selects "Forgot Password" from login screen
    - User provides registered email
    - System sends password reset link
    - User creates new password
4. **Profile Management:**
    - User accesses profile from settings
    - Can update personal information and preferences
    - Changes saved securely to user profile

4.2 Dashboard Warning Light Scanner

4.2.1 Description

The dashboard warning light scanner allows users to capture images of their vehicle's dashboard and automatically identifies and interprets warning lights that are active.

4.2.2 Functional Requirements

- FR5: Image capture functionality
- FR6: Warning light interpretation
- FR15: Multi-light detection capability

4.2.3 Technical Implementation

- Computer vision model using CNN architecture

- Preprocessing steps for image enhancement in various lighting conditions
- Database of 100+ common warning light symbols across major manufacturers
- Real-time highlighting of detected symbols

4.2.4 User Workflow

1. User selects "Scan Dashboard" from home screen
2. Camera interface appears with framing guide
3. User captures dashboard image
4. App processes image and highlights detected warning lights
5. User confirms or retakes image
6. App displays detailed information about each detected light

4.3 Engine Sound Analyzer

4.3.1 Description

The engine sound analyzer records audio of engine operation and uses machine learning to classify sounds associated with common mechanical issues.

4.3.2 Functional Requirements

- FR7: Audio recording capability
- FR8: Sound classification functionality

4.3.3 Technical Implementation

- Audio preprocessing pipeline including noise reduction
- Feature extraction using MFCC (Mel-frequency cepstral coefficients)
- Classification model using CNN or LSTM architecture
- Detection of at least 10 common problematic engine sounds
- Optional audio explanations of results (under consideration)

4.3.4 User Workflow

1. User selects "Record Engine Sound" from home screen
2. App provides instructions for optimal recording
3. User positions device and starts recording (5-10 seconds)
4. App processes audio and performs classification

5. Results display with confidence scores for detected issues

4.4 Diagnosis and Reporting

4.4.1 Description

The diagnosis system combines results from warning light and sound analysis to provide comprehensive fault information, repair suggestions, and educational content.

4.4.2 Functional Requirements

- FR9: Explanation and repair suggestions
- FR10: Video tutorial links
- FR16: Urgency estimation
- FR17: Report sharing
- FR20: Automatic saving of diagnostic reports

4.4.3 Technical Implementation

- Rule-based system for combining visual and audio diagnostic results
- Urgency algorithm based on safety implications and potential damage
- Curated database of repair/maintenance suggestions
- Integration with YouTube API for relevant tutorial retrieval
- Automatic report saving system with timestamps

4.4.4 User Workflow

1. After completing scan/recording, results page displays
2. Each issue shows detailed description, potential causes, and urgency level
3. Video tutorials appear as embedded previews or external links
4. User can share report or find nearby mechanics
5. System automatically saves diagnostic report

4.5 Maintenance Tracking

4.5.1 Description

The maintenance tracking system stores diagnostic history and provides reminders for regular maintenance and follow-ups on detected issues based on urgency levels.

4.5.2 Functional Requirements

- FR11: Diagnostic history storage
- FR12: Maintenance reminders based on urgency levels
- FR17: Report sharing
- FR20: Automatic saving of diagnostic reports

4.5.3 Technical Implementation

- Local SQLite database for diagnostic history
- Cloud synchronization for registered users
- Notification system for scheduled maintenance based on urgency:
    - Red (immediate): Immediate alert
    - Yellow (soon): 1-hour alert
    - Monitoring: 2-hour alert
- Export functionality for reports (PDF format)
- Secure sharing via standard OS mechanisms

4.5.4 User Workflow

1. User accesses history from home screen
2. Chronological list of past diagnostics displayed
3. User can filter by date, issue type, or urgency
4. Detailed view available for each past diagnosis
5. Options to set reminders or share with mechanic

4.6 Online/Offline Functionality

4.6.1 Description

The application provides core functionality without internet connection while enabling enhanced features and updates when online.

4.6.2 Functional Requirements

- FR13: Offline operation for core diagnostics
- FR14: Online updates for databases and models

4.6.3 Technical Implementation

- Preloaded ML models for offline operation
- Incremental model updates to minimize data usage
- Version management system for models and databases
- Offline authentication caching mechanism

4.6.4 User Workflow

1. App functions with core capabilities in offline mode for authenticated users
2. Background checks for updates when connectivity restored
3. Notification of available updates with size information
4. Option to schedule updates during Wi-Fi connection

4.7 Nearby Mechanics Locator

4.7.1 Description

The nearby mechanics locator feature enables users to find mechanics in their vicinity using Google Maps integration, with optional location sharing.

4.7.2 Functional Requirements

- FR21: Locate nearby mechanics using Google Maps
- FR22: Optional location sharing

4.7.3 Technical Implementation

- Google Maps API integration
- Geolocation services with permission management
- Mechanic database with filtering options
- Distance calculation and routing algorithms
- Clear opt-in/opt-out process for location sharing

4.7.4 User Workflow

1. User selects "Find Nearby Mechanics" from results screen or menu
2. App requests location permission if not already granted
3. Map displays with mechanics in vicinity
4. User can filter by distance, services, or ratings

5. Selecting a mechanic shows details and contact information
6. Option to get directions or share diagnostic report

5. Validation and Risk Mitigation

5.1 Stakeholder Engagement

The requirements were gathered and validated through:

- User surveys targeting car users of various expertise levels
- Interviews with automotive technicians and mechanics
- Review sessions with project stakeholders
- Competitive analysis of existing diagnostic tools

5.2 Identified Risks & Mitigation

Risk	Impact	Likelihood	Mitigation Strategy
Authentication system breach	Critical	Low	Implement industry-standard security protocols and regular security audits
Inaccurate fault classification	High	Medium	Iterative model retraining and user feedback loop
Limited datasets availability	Medium	High	Use public datasets and crowd-source anonymized data
Audio interference in real-world recordings	Medium	Medium	Include noise-reduction preprocessing pipeline
Model size exceeds mobile limits	High	Low	Optimize using model quantization and pruning
Low adoption due to technical complexity	High	Medium	Focus on UI/UX simplicity and education tools

Risk	Impact	Likelihood	Mitigation Strategy
Battery drain during extended use	Medium	Medium	Optimize algorithms and implement power-saving modes
False sense of security from diagnosis	High	Medium	Clear disclaimers and urgency indicators
Privacy concerns with location data	High	Medium	Make location sharing optional with clear consent
User resistance to registration	Medium	High	Offer limited guest mode functionality

5.3 Testing Strategy

- **Security Testing:** Authentication system penetration testing and vulnerability assessment
- **Unit Testing:** Individual components (authentication, camera, audio recording, classification)
- **Integration Testing:** Combined diagnostic workflows
- **Performance Testing:** Response times, resource usage
- **Usability Testing:** With representative user groups across different roles
- **Beta Testing:** Limited release to gather real-world feedback
- **Model Validation:** Using labeled test datasets for accuracy metrics
- **Location Services Testing:** Accuracy and performance across different environments

6. Appendices

A. Glossary

- **Authentication:** The process of verifying a user's identity
- **RBAC:** Role-Based Access Control - restricting system access based on user roles
- **Diagnosis:** Process of identifying mechanical or electrical faults
- **Model Inference:** Running ML model on input to get predictions
- **Warning Light:** Dashboard indicator that signals vehicle system status

- **OBD System:** On-Board Diagnostics system in modern vehicles
- **Engine Knocking:** Abnormal combustion sound indicating potential issues
- **False Positive:** Incorrect identification of a non-existent issue
- **Model Quantization:** Technique to reduce ML model size for mobile devices
- **Token:** A piece of data used in authentication processes to represent authorization
- **Geolocation:** Identification of real-world geographic location of a device

B. Analysis Models

- **Use Case Diagram**: Captures major use cases: Register, Login, ScanLights, RecordSound, ViewResults, ViewHistory, FindMechanics
- **Data Flow Diagram:** Illustrates flow: Authentication → Input → Preprocessing → Inference → Output
- **Class Diagram:** Shows relationships between major software components
- **Sequence Diagram:** Details interactions during authentication and diagnostic process
- **State Diagram:** Represents application states during user session
- **Context Diagram:** Shows the CarDiagApp system in relation to external entities
- **Deployment Diagram:** Illustrates the physical architecture of the system

C. Issue Tracking

ID	Issue	Status	Resolution
ISSUE-001	Authentication token expiration handling	In Progress	Implementing refresh token mechanism
ISSUE-002	Model accuracy in low light	Open	Investigating preprocessing enhancements
ISSUE-003	Audio classification with background noise	In Progress	Adding noise isolation algorithms

ID	Issue	Status	Resolution
ISSUE-004	Battery consumption during scanning	Resolved	Optimized camera operations
ISSUE-005	Support for international warning symbols	Open	Expanding training dataset
ISSUE-006	Role transition process for users	Open	Designing approval workflow
ISSUE-007	Tutorial relevance for specific issues	In Progress	Refining metadata matching algorithm
ISSUE-008	Location permission handling	Open	Implementing clearer consent flow
ISSUE-009	Urgency-based notifications	In Progress	Configuring notification scheduling system