Requirement Classification and Specifications

# Purpose of Classifying Requirements

Classifying requirements into functional and non-functional categories ensures clear understanding of what the system must do (functional) versus how well it must do it (non-functional). This helps:  
- Guide the design and implementation process.  
- Ensure all user needs and technical constraints are addressed.  
- Facilitate testing, validation, and system maintenance.  
- Aid in resource allocation and prioritization.

# Functional Requirements Specifications

|  |  |  |
| --- | --- | --- |
| FR# | Functional Requirement | Specification |
| FR1 | User Registration | User registration with email verification step to confirm identity. |
| FR2 | Role-based Authentication | Role-based login access (Car Owner, Mechanic, Administrator). |
| FR3 | Password Recovery | Secure password recovery via email or security questions. |
| FR4 | Session Management | User sessions with inactivity timeout to ensure security. |
| FR5 | Image Capture | Users can take dashboard photos; app detects warning icons. |
| FR6 | Icon Interpretation | App explains detected icons with urgency labels (e.g., critical). |
| FR7 | Audio Recording | Record engine sounds for up to 10 seconds using device microphone. |
| FR8 | Sound Classification | Audio analysis classifies sound into predefined fault categories. |
| FR9 | Text Explanation | Faults include explanations and text-based repair suggestions. |
| FR10 | Tutorial Links | Diagnostic results include links to relevant tutorial videos. |
| FR11 | History Storage | Diagnostic history saved locally with timestamps. |
| FR12 | Maintenance Reminders | Sends reminders for maintenance based on faults or schedule. |
| FR13 | Offline Diagnostics | Diagnostics (image/audio) work offline using preloaded models. |
| FR14 | Online Updates | Updates fault database/tutorials when internet is available. |
| FR15 | Vehicle Info Input | Vehicle details (make/model/year) influence diagnostic accuracy. |
| FR16 | Multiple Icon Detection | Detects and interprets multiple warning icons in one image. |
| FR17 | Urgency Estimation | Assigns urgency levels (Immediate, Soon, Monitor) to issues. |
| FR18 | Report Sharing | Users can save/share diagnostics reports. |
| FR19 | Maintenance Tips | Provides preventive tips based on identified problems. |
| FR20 | Mechanic Notes | Mechanics can add notes to diagnostic reports (restricted to role). |
| FR21 | User Management | Admins manage users (add, remove, edit profiles). |
| FR22 | Profile Management | Users can edit profile and set preferences. |

# Summary of Non-Functional Requirements

## Performance

* Fast diagnostics: ≤5s (image), ≤7s (audio)
* Fast UI and authentication: ≤3s
* High throughput: 50 sessions/day/user, 100+ concurrent auth requests
* Resource limits: <200MB RAM, <30% CPU, <1% battery during standby

## Reliability

* Uptime: 99% online, 100% offline core features
* MTBF: <5 crashes per 1000 sessions
* Strong error handling and recovery for session loss, denied permissions, network failure

## Usability

* Simple UI for non-technical users; max 3 taps to any function
* Fast onboarding: <2 mins registration, <3 mins first diagnosis
* Accessibility: screen readers, voice commands, high contrast, WCAG 2.1 AA compliance

## Security

* Multi-factor auth, secure storage, role-based access
* Local processing of sensitive data, opt-in for analytics
* Explicit permission requests with clear justification
* Integrity checks, audit trails, anti-tampering

## Maintainability & Portability

* Modular, documented Flutter codebase with layered architecture
* Support for Android/iOS, including tablets
* OTA updates, versioned migrations, isolated auth module

## Quality Attributes

* Accuracy: ≥90% (visual), ≥85% (audio), <5% false positives
* Efficient: ≤5s total diagnosis, low battery and data use
* Privacy: no tracking, data anonymized, compliant with regulations
* Lightweight: optimized for low bandwidth, compressed data transfers