SOFTWARE REQUIREMENT SPECIFICATION

for

Fleet Management System iOS Application

Version History

Name	Date	Version
Virag Bardiya	03/02/2025	1.0
Prasad B S	29/12/2025	2.0

1. INTRODUCTION

1.1. APP VISION

This document specifies the software requirements for a Fleet Management System (FMS) that facilitates the management and tracking of a fleet of vehicles. The FMS will cater to the needs of fleet managers, drivers, and maintenance personnel, enabling efficient vehicle utilization, maintenance scheduling, and cost control.

1.2. PURPOSE OF THE APP

This project is not being developed for specific client, however it will be extensible enough to be adapted and customisable for deployment and integration into any logistics network. The objectives of the apps are:

- Provide a centralized platform for managing vehicle information and maintenance schedules.
- Facilitate real-time tracking of vehicle locations and usage.
- Optimize vehicle routing and dispatching.
- Track fuel consumption and other operational costs.
- Enhance communication and collaboration among fleet managers, drivers, and maintenance personnel.
- AI-driven predictive maintenance, fuel optimization, intelligent routing.

The target audience includes transportation companies, logistics providers, delivery services, and any organization managing a fleet of vehicles.

2. SCOPE OF THE PROJECT

Definitions:

Vehicle: Any vehicle managed by the FMS (cars, trucks, vans, etc.).

Driver: An individual authorized to operate a vehicle in the fleet.

Maintenance: Scheduled or unscheduled maintenance activities performed on vehicles.

Route: A planned path for a vehicle to travel.

Trip: A single journey made by a vehicle, often as part of a route.

2.1. HIGH LEVEL FEATURES

(Add Min 5 features per user type identified) Sample features shall be as below

2.1.1.Admin Management:

2.1.1.1.User Management: System administration, driver and maintenance personnel account management with differentiated access controls.

2.1.1.2.Vehicle Management: Adding, editing, and deleting vehicle information (make, model, VIN, etc.).

2.1.1.3.Maintenance Management: Setting up maintenance schedules, AI-powered predictive maintenance, tracking maintenance history, and generating maintenance reports.

2.1.1.4.Reporting and Analytics: Generating reports on vehicle usage, fuel consumption, maintenance costs, and other key metrics, fuel consumption optimization, automated compliance alerts.

2.1.1.5.Geofencing: Defining virtual boundaries for vehicle tracking and alerts.

2.1.2.Driver Management:

2.1.2.1.Trip Management: Intelligent route suggestions based on traffic and fuel efficiency,

2.1.2.2. Voice-based trip logging, recording trip details, including start/end times, locations, and mileage.

2.1.2.3.Vehicle Inspection: Reporting vehicle defects or maintenance needs.

2.1.2.4.Communication: Communicating with fleet managers and maintenance personnel.

2.1.2.5.Route Navigation: Accessing assigned routes and navigation tools.

2.1.3.Maintenance Personnel Management:

2.1.3.1.Maintenance Scheduling: Viewing and managing scheduled maintenance tasks.

2.1.3.2.Maintenance Tracking: Recording completed maintenance activities, including parts used and labor costs.

2.1.3.3.Inventory Management: Managing inventory of spare parts and supplies. AI-driven spare parts forecasting

2.1.3.4.Communication: Communicating with fleet managers and drivers regarding maintenance issues.

2.1.3.5.Work Order Management: Creating and managing work orders for vehicle repairs and maintenance, automated work order prioritization.

2.2. TYPES OF USERS

(Min 3 types of users to be identified)

1. Fleet Manager: Manages user accounts, system settings, vehicle information, and generates reports. Requires strong technical skills and fleet management knowledge.

2. Driver: Operates vehicles, records trip details, reports vehicle issues, and communicates with fleet managers. Requires basic computer literacy and driving skills.

3. Maintenance Personnel: Performs vehicle maintenance, tracks parts usage, and communicates with fleet managers. Requires technical skills related to vehicle maintenance.

2.3. TECHNICAL SPECIFICATIONS

Devices: iPhone, iPad, OS: iOS 26+.

Frameworks: SwiftUI, Core ML, Vision, App Intents.

Architecture: MVVM with Swift Concurrency.

Security: Passkeys, encryption, role-based access, GDPR compliance.

3. NON-FUNCTIONAL REQUIREMENTS

3.1. PERFORMANCE

- Load Time: The application will be responsive and load quickly, even on slow network connections
- The application will be able to handle high volumes of data (xxx MB) and support large numbers of users (1000s).
- Zero memory leaks
- Zero constraint issues (warnings and conflict errors)

3.2. SECURITY

- The application will use secure login protocols, such as password protection and two-factor authentication, to ensure the privacy and security of user information. Should have role based authentication for accessing functional modules,
- The application will use secure data storage and encryption to protect sensitive information from unauthorised access.
- The application will comply with relevant data privacy regulations

3.3. USABILITY

- The application will have a user-friendly interface that is easy to navigate and understand.

3.4. SCALABILITY

- The application should be scalable to accommodate growth and changes in user needs.
- The application should be able to support new features and functionalities as needed.

3.5. RELIABILITY

- The application should be reliable and available at all times, with minimal downtime.
- The application should be able to recover from failures and maintain data consistency in case of errors.
- Zero leaks, zero constraint warnings, zero warnings

3.6. ACCESSBILITY

- How easily people with the widest range of capabilities can use the system.

4. DETAILED FUNCTIONAL REQUIREMENTS (SAMPLE)

4.1. Fleet Manager:

4.1.1.Vehicle Management:

4.1.1.1.Add new vehicles with detailed information (make, model, year, VIN, license plate).

4.1.1.2.Edit existing vehicle information.

4.1.1.3.Assign vehicles to drivers.

4.1.2.Maintenance Management:

4.1.2.1.Create and schedule preventive maintenance tasks.

4.1.2.2.Track maintenance history for each vehicle.

4.1.2.3.Generate maintenance reports.

4.2. Driver:

4.2.1.Trip Management:

4.2.1.1.Start and end trips, recording mileage and location.

4.2.1.2.Report delays or issues during trips.

4.2.1.3.View assigned routes.

4.2.2.Vehicle Inspection:

4.2.2.1.Perform pre-trip and post-trip vehicle inspections.

4.2.2.2.Report vehicle defects or maintenance needs.

4.3. Maintenance Personnel:

4.3.1.Maintenance Tracking:

4.3.1.1.Record completed maintenance tasks.

4.3.1.2.Update vehicle maintenance history.

4.3.1.3.Manage parts inventory.

5. SUBMISSIONS

5.1. Code base

5.2. App Video demo

5.3. Memory Profile Screenshot

5.4. Flow diagram