**Vehicle Service Management System - Requirements Document**

## Table of Contents

1. \*\*Introduction\*\*

- Purpose

- Scope

- Definitions

- References

2. \*\*Overall Description\*\*

- Product Perspective

- Product Functions

- User Classes and Characteristics

- Operating Environment

- Design and Implementation Constraints

3. \*\*Functional Requirements\*\*

- Vehicle Management

- Component Management

- Issue Management

- Payment Processing

- Reporting and Analytics

4. \*\*Non-functional Requirements\*\*

- Performance Requirements

- Security Requirements

- Usability Requirements

5. \*\*Assumptions and Dependencies\*\*

6. \*\*Acceptance Criteria\*\*

---

## 1. Introduction

### Purpose

The purpose of this document is to outline the requirements for the Vehicle Service Management System, which will facilitate vehicle management, issue tracking, payment processing, and reporting.

### Scope

The system will support users in managing vehicle services, tracking issues, processing payments, and generating revenue reports. It will include a web-based interface for both users and administrators.

### Definitions

- \*\*User\*\*: Any person who utilizes the system, including customers and administrators.

- \*\*Vehicle\*\*: Any automobile that is registered in the system.

-\*\*Component\*\*:Any component used for a vehicle registered in the system.

- \*\*Issue\*\*: A problem reported by the user regarding a vehicle.

- \*\*Payment\*\*: The monetary transaction made by the user for services rendered.

### References

- [Recharts Documentation](https://recharts.org/en-US/)

- [Django REST Framework](https://www.django-rest-framework.org/)

---

## 2. Overall Description

### Product Perspective

The Vehicle Service Management System will be a standalone web application. It will be developed using Django for the backend and React for the frontend. The system will be hosted on a cloud server.

### Product Functions

- Component management (add, update components)

- Vehicle management (add, update vehicles)

- Issue tracking (log issues related to vehicles)

- Payment processing (process payments based on logged issues)

- Generate revenue reports (daily, monthly, yearly)

### Operating Environment

- Web browsers (latest versions of Chrome, Firefox, Safari, and Edge)

- Mobile devices (responsive design)

### Design and Implementation Constraints

- The application must adhere to security best practices.

- Performance optimizations for faster loading times are required.

---

## 3. Functional Requirements

### 3.1 Vehicle Management

- Users can add new components with details.

- Users can update component information.

### 3.2 Vehicle Management

- Users can add new vehicles with details

- Users can update vehicle information.

### 3.3 Issue Management

- Users can log issues for their vehicles.

- Issues can be marked as repairable or non-repairable.

- Each issue can be associated with multiple components.

### 3.4 Payment Processing

- Users can view the total price based on the logged issues.

- Users can process payments for services rendered.

- Payment history should be tracked and accessible by users.

### 3.5 Reporting and Analytics

- The system should generate reports on revenue (daily, monthly, yearly).

- Graphs for revenue should be responsive and user-friendly.

---

## 4. Non-functional Requirements

### 4.1 Performance Requirements

- The application should load within 3 seconds under normal conditions.

- The system should handle up to 100 concurrent users.

### 4.2 Security Requirements

- All data should be transmitted over HTTPS.

- User data should be protected and not exposed to unauthorized access.

### 4.3 Usability Requirements

- The user interface should be intuitive and easy to navigate.

- The system should be responsive and usable on mobile devices.

---

## 5. Assumptions and Dependencies

- Users will have access to the internet.

- Users will have devices compatible with modern web browsers.

---

## 6. Acceptance Criteria

- Ability to manage components (add, update).

- Ability to manage vehicles (add, update).

- Ability to log and track issues related to vehicles.

- Successful payment processing and display of payment history.

- Generation of accurate revenue reports.

---