Software Requirements Specification for Vans Tracking System

Toqa Ezzatly/D$_H$ub

December 17, 2024

Contents

1 Introduction

1.1 Purpose of This Document

This document specifies the requirements for the Vans Tracking System. It is intended for use by the system developers, testers, and stakeholders. It will serve as a guide for the design, development, and validation of the system.

1.2 Scope of the Project

The Vans Tracking System is a real-time system designed to track and manage a fleet of vans. The system will provide features such as location monitoring, speed tracking, payload tracking, and mileage tracking. The scope of the project includes software for web and mobile platforms.

1.3 Definitions, Acronyms, and Abbreviations

- **GPS**: Global Positioning System.
- **SRS**: Software Requirements Specification.
- **UI**: User Interface.

1.4 References

- IEEE Std 830-1998: Recommended Practice for Software Requirements Specifications.

2 Overall Description

2.1 Product Perspective

The Vans Tracking System will be a standalone application, integrating GPS hardware and software. It will interact with the following external interfaces:

- GPS Devices (for location data).
- Database (for data storage).
- Web Browser (for management UI).
- Mobile App (for driver access).

2.2 Product Functions

The system will perform the following major functions:

1. Real-time location tracking of vans.
2. Display van location on a map.
3. Monitor and report van speed.

4. Monitor and record van payload.
5. Record and report van mileage.
6. Provide a user-friendly interface for administrators and drivers.
7. Generate reports on van usage and performance.

2.3 User Classes and Characteristics

- **Fleet Managers**: Responsible for managing the fleet and monitoring all vans, they require the web interface.
- **Drivers**: Responsible for the van, requires the mobile app to view their current location, payload and mileage.

2.4 Operating Environment

- The system must be accessible from any device with a web browser.
- The mobile application will need to be installed on a mobile device, must be on at least android 8.
- The GPS data must be updated at least every 5 minutes.
- The database will be located on a remote server.

2.5 Design and Implementation Constraints

- The system must comply with all data privacy regulations and standards.
- System must be developed in Python using a PostgreSQL database.

3 Specific Requirements

3.1 Functional Requirements

3.1.1 Location Tracking

- The system shall track and record each van's location using GPS at least every 5 minutes.
- The system shall display the current location of each van on a map interface.
- The system shall provide a history of van locations.

3.1.2 Speed Tracking

- The system shall track the speed of each van.
- The system shall record the maximum speed achieved by each van.
- The system shall alert the user if a van exceeds the maximum set speed.

3.1.3 Payload Tracking

- The system shall track the load of each van.
- The system shall display the current load of each van.
- The system shall prevent loading if the maximum capacity is exceeded.

3.1.4 Mileage Tracking

- The system shall track and record each van's mileage using GPS.
- The system shall record the cumulative mileage of each van.

3.2 Non-Functional Requirements

3.2.1 Usability

- The UI shall be user-friendly and intuitive.
- The web interface shall be compatible with all modern web browsers.

3.2.2 Performance

- The system shall update van location on the map within 2 seconds of receiving the GPS data.
- The system shall handle concurrent requests from multiple users without performance degradation.

3.2.3 Security

- The system shall protect user data using encryption protocols.
- The system shall require user authentication for access to administrative functionalities.

3.2.4 Reliability

- The system shall be available 99.9% of the time.
- The system shall have a backup and recovery mechanism to prevent data loss.

4 Future Enhancements

- Predictive routing based on historical data.
- Integration with third-party APIs for weather updates.
- Automated alerts for scheduled maintenance based on mileage.

5 Diagrams

Figure 1: Data Flow Diagram

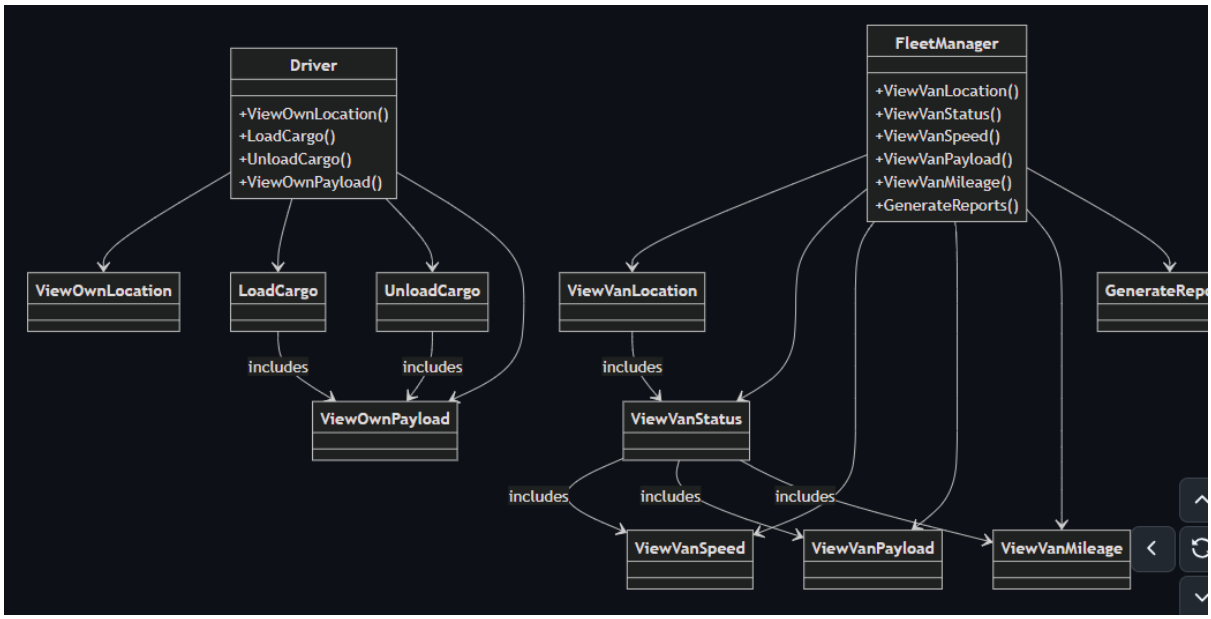

Figure 2: Use Case Diagram

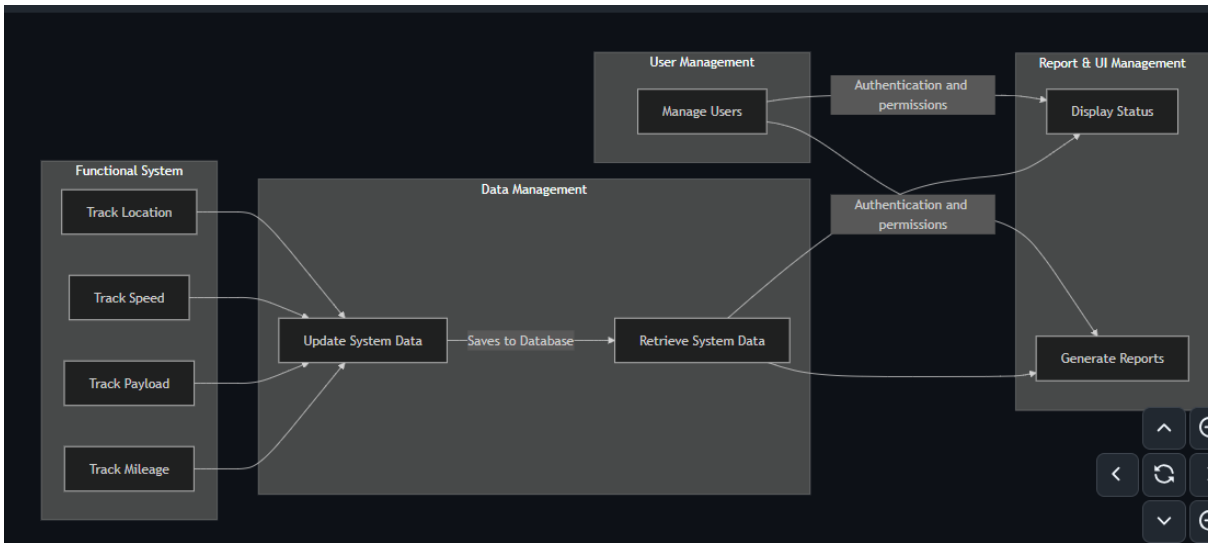

Figure 3: Functional Diagram

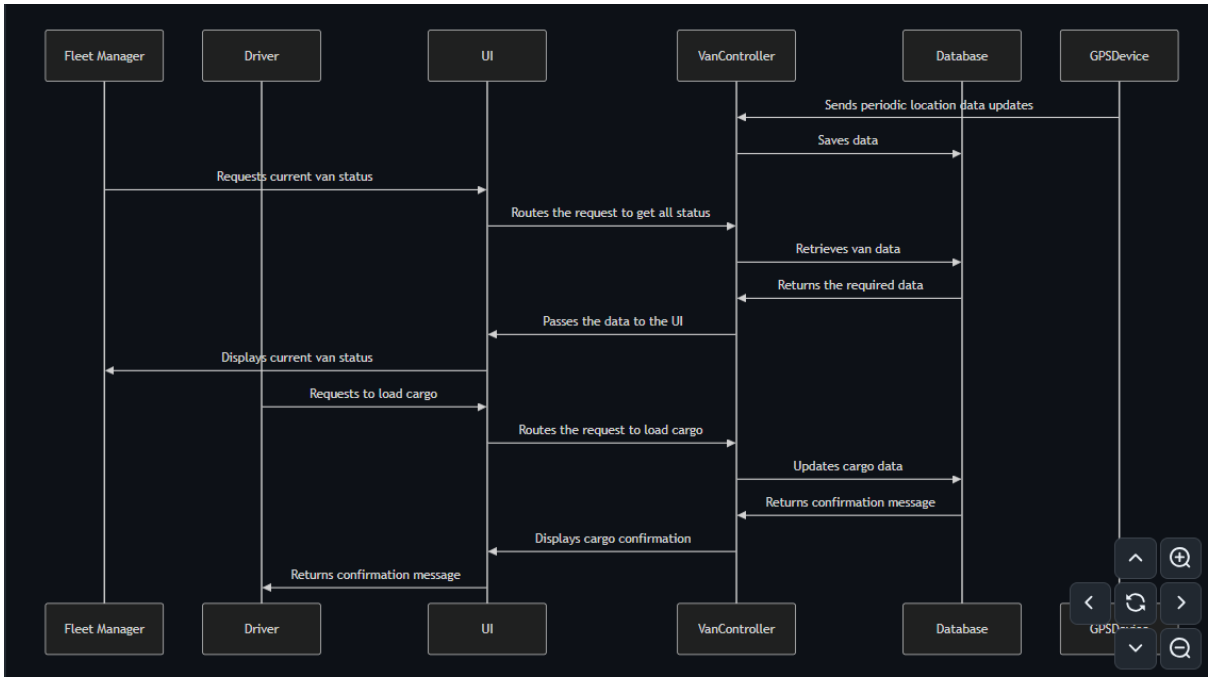

Figure 4: Sequence Diagram

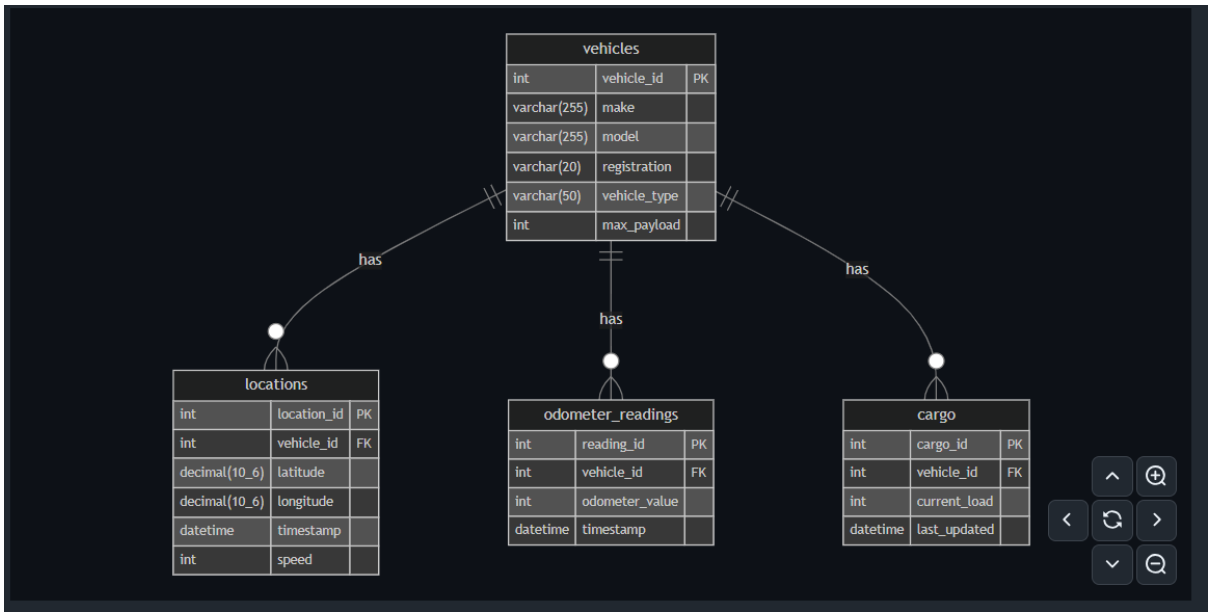

Figure 5: Entity Relational Diagram