IIT(ISM) Dhanbad

Software Requirements Specification

PROJECT DETAILS

PROJECT NAME		
Vehicle Tracking System		
CREATOR		
● *Shyam Sunder* ● *Shubham Saurav* ● *Shresth Gupta*		
Instructed By : **Prof Saurabh Srivastava**		
Under guidance of : **Mr Rahul Soren**		
DOCUMENT NO.	**DATE**	**VERSION NO.**
VTS02	**05/02/2025**	**V01**

Table of Contents

1. Introduction

1.1. Purpose

This document aims to provide a detailed Software Requirements Specification (SRS) for the Vehicle Tracking System (VTS) to ease the process of building the desired application system. The desired system will allow administrators, vehicle owners, and general users(passengers users and bulk users) to interact with a real-time tracking and monitoring platform that ensures efficiency, security, and ease of use.

1.2 Scope

The desired VTS will enable:

- Vehicle owners to register and monitor their vehicles in real-time
- Passenger Users to check active vehicles nearby their devices
- Bulk Users to track location of their shipment
- Provide alerts and notifications to users and vehicle owners
- Administrators to manage the platform and oversee all activities

1.3 Intended Audience and Reading Suggestions

- Software Development Team
- Client
- Anyone possessing knowhow of Software Engg.

1.4 Definitions, Acronyms, and Abbreviations

- **GPS**: Global Positioning System
- **SRS :** Software Requirements Specification
- **BRD :** Business Requirements Document

- **VTS :** Vehicle Tracking System
- **FCM**: Firebase Cloud Messaging
- **GMap**: Google Maps API
- **API**: Application Programming Interface
- **DB**: Database
- **ISM**: Indian School of Mines
- **iOS**: iPhone Operating System
- **OAuth**: Open Authorization

1.5 References

- BRD for the VTS → Document No. : VTS01, Version : V01
- React Native Documentation
- Firebase Authentication and Realtime Database Documentation
- Google Maps API Documentation

1.6 Overview

This document details the system's functional and non-functional requirements, as well as its constraints, assumptions, and potential risks associated with the project.

---

2. Overall Description

2.1 Product Perspective

The system is a standalone native solution that leverages React Native for application framework, Firebase for authentication and real-time database storage, Google Maps API for location tracking, and FCM for notifications.

2.2 Product Functions

1. User Management:
    - Admin can manage vehicle owners and approve or reject registrations.
    - Vehicle owners can register and monitor their vehicles.
    - Both types of users can track vehicles and receive notifications.
2. Real-Time Tracking:
    - GPS-based vehicle tracking.
    - Interactive map visualization.
3. Notifications:
    - Alerts for overspeeding, shipment delivery and ride completion
4. Data Privacy & Security:
    - Role-based access control.
    - End-to-end encryption for data transmission.

2.3 Operating Environment

The software system :

- Works on Android and iOS.
- Tech Stacks: React Native for frontend, Firebase for backend. GMap API will be used as a map service.

2.4 User Characteristics

- Administrators: Tech-savvy individuals managing user accounts and data.
- Vehicle Owners: Drivers and fleet owners monitoring their vehicles.
- General Users: Individuals tracking vehicles for convenience.
- Bulk Users : They can track the live location of their shipment and will receive notifications about their delivery.

2.5 Constraints

- Internet connectivity required for real-time tracking.
- Initial deployment limited to ISM campus.
- GPS signal delays due to network issues.

2.6 Assumptions

- Users have compatible smartphones with GPS enabled.
- Users will have an active internet connection.

---

3. Specific Requirements

3.1 Functional Requirements

3.1.1 User Management

- *Admin View:*
  - Admins open the application and manage vehicle owner accounts.
  - Approve/reject vehicle owner registrations.
- *Vehicle Owner View:*
  - For first time users (Registration): Vehicle Owners will register themselves by adding their Name, Age, License No, Vehicle Registration No, Phone Number and on click of register button they get registered. After registration they come to the HomePage.
  - For existing users: Vehicle owners will login using their Phone Number and OTP. After login they will be transferred to HomePage.
- *User View:*
  - On opening the app they directly come to HomePage and they get to know nearby vehicle locations.

    - They can change their personal information. They will also have the option to anonymize their data.

3.1.2 Tracking and Monitoring

- Real-time GPS tracking of registered vehicles.
- Display vehicle location and speed using Google Maps API on user's view.

3.1.3 Data Privacy

- Ensure data is safe and under their control.
- Implement role-based access control.
- Use Firebase Auth and Firebase Realtime DB to store driver information.

---

4. Non-Functional Requirements

4.1 Performance

- Support up to 1,000 concurrent users with <5 seconds response time.
- Real-time tracking updates every *10-15 seconds*.

4.2 Scalability

- System should scale to *10,000+ vehicles* in future expansions.

4.3 Security

- Implement **end-to-end encryption** for data transmission.
- Use **OAuth 2.0** for secure authentication.

4.4 Availability

- Ensure **2 nines uptime** with robust failover mechanisms.

4.5 Usability

- Intuitive UI for all user roles.
- Responsive design compatible with Android & iOS.

4.6 Maintainability

- Modular architecture for easy updates.
- Well-documented codebase and system design.

---

5. Other Considerations

5.1 Risks

- GPS signal transmission delays due to poor network coverage.
- Possible misuse of system data if access controls are compromised.
- Resistance from users unfamiliar with digital tracking platforms.

5.2 Future Enhancements

- Expansion beyond ISM campus.
- Integration with additional third-party APIs for enhanced tracking.

---

6. Appendices

- Firebase Documentation
- Google Maps API Documentation
- React Native Documentation