Software Requirements Specification (SRS)

---

1. Introduction

1.1 Purpose

This document specifies the requirements for a **web-based Bus Booking and Tracking System**.
The purpose of the system is to enable:

- **Users** can register, browse available bus routes, make and manage bookings, and track buses in real-time.
- **Admins** to manage routes, assign drivers to buses, monitor system analytics.
- **Drivers** to view assigned routes and update real-time status information.

The system will provide live bus location updates using GPS integration, send notifications for changes or delays.

1.2 Scope

The Bus Booking and Tracking System will:

- Allow **Users** to:
    - Sign up and log in.
    - View all available bus routes, stops, timings, and prices.
    - Book rides and manage existing bookings.
    - Cancel or modify bookings only if done ≥ **2 hours before scheduled departure**.
    - Receive notifications (e.g., booking confirmations, delays, cancellations).
    - Track buses in real-time..
- Allow **Admins** to:
    - Add, edit, or remove routes and stops.
    - Assign drivers to buses.
    - View all users and booking history.
    - Monitor bus locations in real-time.
    - Payments and Notifications Oversight & Monitoring.
    - View analytics (e.g., peak booking times, cancellations, revenue).
- Allow **Drivers** to:
    - View assigned routes, stops, and schedules.
    - Update trip status (start, delay, completed).
    - Send delay or incident updates directly to the system.

The system will integrate with **Google Maps API** (with provision to switch APIs easily) for route visualization and live tracking.

---

2. Overall Description

2.1 Product Perspective

The system will be a **web-based application** with role-based access control.
It will:

- Use a central database to store user, route, booking, and driver information.
- Connect to GPS tracking devices or a mobile tracking app for drivers.
- Use API integration (e.g., Google Maps API) for live tracking and route maps.
- Be accessible from both desktop and mobile devices.

2.2 Product Functions

The system will support:

- **User Account Management** (registration, login, profile updates).
- **Route and Schedule Viewing** with filtering by date, time, and location.
- **Booking and Payment Processing** (secure online payments).
- **Booking Modification Rules** (changes/cancellations only ≥ 2 hours before departure).
- **Admin Route & Driver Management** (including driver allocation).
- **Live Bus Tracking** (real-time location updates every ≤ 5 seconds).
- **Notification System** (push notifications).
- **Analytics & Reporting** (admin access).

2.3 User Classes and Characteristics

- **Users**:
    - Familiar with basic web navigation.
    - Responsible for their own bookings.
    - Require access to live bus tracking and notifications.
- **Admins**:
    - Skilled in managing operational data.
    - Have full control over routes, drivers, payments, and system settings.
- **Drivers**:
    - Require simple mobile interface for schedule viewing and status updates.
    - Responsible for accurate trip status updates.

2.4 Assumptions and Dependencies

- GPS hardware or mobile tracking apps will be available in each bus.
- Users will have stable internet access
- Notification service providers (email, SMS) will be integrated.
- Google Maps API or equivalent will be available for live tracking.
- Payment gateway integration will be supported.

---

3. Specific Requirements

3.1 Functional Requirements

3.1.1 User Management

- Users can register, log in, and edit their profile.
- Users can view available bus routes, stops, timings, and prices.
- Users can make bookings with secure payment.
- Users can modify or cancel bookings only ≥ **2 hours before departure**.

3.1.2 Admin Dashboard

- View, add, edit, and delete routes and stops.
- Assign drivers to buses.
- View and export booking and payment data.
- Monitor real-time bus locations.
- Send system-wide notifications.
- Access analytics on bookings, cancellations, and revenue.

3.1.3 Driver Dashboard

- View assigned route and schedule.
- Update trip status: “Starting,” “In Transit,” “Delayed,” “Completed.”
- Report delays or incidents directly.

3.1.4 Booking & Payment System

- Display route availability in real time.
- Enforce booking modification restrictions.
- Process cancellations with refund rules.
- Integrate secure online payments (UPI).

3.1.5 Live Tracking

- Show real-time bus location on a map for both Admins and Users.
- Update ETA dynamically.
- Automatically notify users if the bus is delayed or rerouted.

3.1.6 Notification System

- Send booking confirmations, reminders, and cancellation notices.
- Send live status updates (delays, route changes).
- Notifications via push alerts.

---

4. External Interface Requirements

4.1 User Interfaces

- **Login/Signup Page** (common for all roles, redirect based on role).
- **User Dashboard**: bookings, routes, live tracking.
- **Admin Dashboard**: route management, driver assignments, analytics.
- **Driver Dashboard**: assigned routes, status updates.

4.2 Hardware Interfaces

- GPS tracking devices or smartphones for buses.

4.3 Software Interfaces

- **Database**: MongoDB.
- **Frontend**: React.js.
- **Backend**: Node.js.
- **Mapping API**: Google Maps API (with modular implementation to allow easy switching).
- **Payment Gateway**: Stripe, PayPal, or similar.

---

5. Non-Functional Requirements

5.1 Performance

- Handle up to **10,000 concurrent users**.
- Live location updates every ≤ **5 seconds**.

5.2 Security

- Encrypted passwords and authentication.
- Role-based access control.
- Secure payment transactions.

5.3 Availability

- **99.9% uptime**.
- Scheduled maintenance alerts.

5.4 Maintainability

- Modular code structure for easy updates.
- Clear API abstraction to switch mapping services.

5.5 Scalability

- Add new routes, buses, or cities without downtime.