A **Software Requirements Specification (SRS)** document outlines the functionality, design, and other details of a system. Here’s an example outline for an SRS document tailored to a **Bus Ticket Reservation System**:

**1. Introduction**

1.1 **Purpose**  
This document defines the software requirements for the Bus Ticket Reservation System. The system allows users to search for available buses, book tickets, cancel bookings, and manage their reservations. It also enables the admin to manage routes, bus schedules, and bookings.

1.2 **Scope**  
The system will support both user and admin interfaces. The main functionalities will include searching for buses, booking tickets, payment processing, user profile management, and admin dashboard for bus and route management.

1.3 **Definitions, Acronyms, and Abbreviations**

* **User**: A customer booking bus tickets.
* **Admin**: A system administrator who manages bus schedules and other operational tasks.
* **Reservation**: A booked seat on a bus, with a specific departure time.

1.4 **Overview**  
This SRS document provides the functional and non-functional requirements for the Bus Ticket Reservation System. It describes the system's purpose, usage, and intended operations.

**2. Overall Description**

2.1 **Product Perspective**  
The system will operate as a web-based application that allows users to interact with the reservation platform and book bus tickets. The system will include an admin interface for management of the buses, schedules, and reservations.

2.2 **Product Functions**

* **User Side**:
  + Register and log in.
  + Search for buses based on date, time, and route.
  + View bus details (bus number, departure time, available seats).
  + Book tickets.
  + View and manage reservations.
  + Cancel bookings and request refunds.
* **Admin Side**:
  + Add, update, or delete bus routes and schedules.
  + View and manage user reservations.
  + Monitor the status of bookings.

2.3 **User Classes and Characteristics**

* **End Users**: Customers seeking to book bus tickets.
* **Admin Users**: System administrators managing routes, schedules, and bookings.

2.4 **Operating Environment**

* Web-based system compatible with all modern browsers (Chrome, Firefox, Safari, Edge).
* Database for storing user and booking information.

**3. System Features**

3.1 **User Registration and Login**

* Users must be able to register with an email, password, and personal details.
* After registration, users can log in to their accounts.

3.2 **Search Bus Availability**

* Users can search for available buses based on departure location, destination, and date.
* Search results display the list of buses with available seats and departure times.

3.3 **Book Tickets**

* Users can select a bus, choose seats, and book tickets.
* A user can pay for the tickets via integrated payment gateways (e.g., PayPal, Stripe).

3.4 **Cancel or Modify Booking**

* Users can view and manage existing bookings.
* Users can cancel or modify their reservations, subject to cancellation policies.

3.5 **Admin Management**

* Admin can add, edit, or remove bus routes and schedules.
* Admin can view and manage all user bookings.
* Admin can generate reports on system usage and reservations.

**4. External Interface Requirements**

4.1 **User Interfaces**

* Web-based user interface designed for easy navigation.
* Clean, intuitive design with search filters, booking forms, and confirmation dialogs.

4.2 **Hardware Interfaces**

* The system will be accessible on personal computers, laptops, and mobile devices.

4.3 **Software Interfaces**

* Integration with payment gateways for processing transactions.
* Database interface for storing and retrieving reservation data.

4.4 **Communications Interfaces**

* The system will use HTTPS for secure communication between the client and server.

**5. System Features**

5.1 **Booking Confirmation and Notifications**

* Users will receive a confirmation email or SMS after a successful booking.
* Notifications will be sent for booking updates, cancellations, or reminders.

5.2 **Payment Gateway Integration**

* The system will support payments via credit/debit cards and third-party payment processors.

**6. Non-Functional Requirements**

6.1 **Performance Requirements**

* The system should handle a load of up to 10,000 simultaneous users.
* Response time for booking actions should not exceed 2 seconds under normal conditions.

6.2 **Security Requirements**

* All user data must be encrypted in the database.
* Strong password policies for user authentication.
* Admin access should be restricted with role-based access control.

6.3 **Reliability and Availability**

* The system must be operational 99.5% of the time.
* The system should have failover mechanisms for high availability.

6.4 **Usability**

* The interface should be intuitive and easy to use, with minimal training required for end users.
* Mobile-friendly design to support users on different devices.

**7. Other Requirements**

7.1 **Backup and Recovery**

* Regular backups of user and transaction data will be performed.
* A disaster recovery plan should be in place to restore the system in case of failures.

7.2 **Compliance**

* The system will comply with data protection regulations, such as GDPR for handling personal user information.

**8. Appendix**

* Any supplementary information or additional details can be included here.

This is just a template to get you started. Depending on the complexity of your system, you might need to adjust or expand sections accordingly. Let me know if you need further details for any section!