

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING  
DELHI TECHNOLOGICAL UNIVERSITY

SOFTWARE REQUIREMENT  
SPECIFICATION FOR  
CAB BOOKING SYSTEM

Submitted to:  
Dr. Manoj Sethi  
C0301 - Software  
Engineering.

Submitted by:  
Ayush Tandon  
2K22/C0/133  
(A2 batch - C0)

## TABLE OF CONTENTS

1. Introduction
  - Problem Statement
  - 1.1 Purpose
  - 1.2 Product Scope
  - 1.3 Definitions, Acronyms and Abbreviations.
  - 1.4 References
2. Overall Description.
  - 2.1 Product Perspective
  - 2.2 Product Functions
  - 2.3 User classes and Characteristics
  - 2.4 Operating Environment
  - 2.5 Design & Implementation Constraints
  - 2.6 User Documentation
  - 2.7 Assumption & Dependencies.
3. External Interface Requirements
  - 3.1 User Interface
  - 3.2 Hardware Interface
  - 3.3 Software Interface
  - 3.4 Communication Interface
4. System Features
  - 4.1 User Registration and Authentication
  - 4.2 Cab Booking and Ride Tracking
  - 4.3 In-App Payment Processing
  - 4.4 Driver Rating and feed back
  - 4.5 Driver Assignment and Details.
5. Other Non Functional Requirements
  - 5.1 Performance Requirements
  - 5.2 Safety Requirements
  - 5.3 Security Requirements
  - 5.4 Software Quality Attributes
  - 5.5 Business Rules
6. Other Requirements.

# SRS For Cab Booking System.

— Ayush Tandon

## 1. Introduction

This document presents a comprehensive description of the software requirements for cab booking system.

This booking feature will help a user to select a ride according to his schedule, comforts, availability, affordability. Document will provide a detailed account of system's functionality and specifies the requisite standards for development and implementation.

### Problem Statement

Cabs booking service often face issues like long wait times, inefficient ride allocation, unclear pricing, and limited communication between customer and driver. These problems lead to poor customer experience, especially during peak hours or in remote areas. Our cab booking system aims to address these challenges by providing real time tracking, seamless communication, secure cashless payments, with efficient booking process, improving overall user satisfaction for both passenger and drivers.

#### 1.1 Purpose

The purpose of this document is to specify functional and non-functional requirements for development of cab booking system. The system allows user to book cabs, track rides, make payments, and provide feedback.

## 1.2 Scope

The system is designed for passengers and drivers, providing a seamless way for users to book rides and for drivers to accept and complete trips. Aim is to automate system, calculate fare, collect fare, collect all necessary information of the client and then serve client. Data will be stored in a central database to make better decisions in future, enhancing user experience.

## 1.3 Definitions, Acronyms and Abbreviations

- User who books cab is a Passenger.
- Driver: User who drives the cab.
- OTP : one time password.
- ETA : Estimated Time of arrival

## 1.4 References

- Google maps API documentation
- Cab booking design guidelines
- Driver Laws Rule book.
- Payment Gateway Operations.
- IEEE SRS (Software Requirement Specification) template

## 2. Overall Description

### 2.1 Product Perspective

The Cab booking system is a standalone mobile application designed for passengers and drivers. It integrates with Google Maps for navigating and stripe for payment processing. The system is designed for high availability and accessibility.

## 2.2 Product functions

- 1. User registration and authentication
- 2. Cab booking and ride tracking
- 3. Driver assignment and notification
- 4. In-app payments and bill generation
- 5. Choice of vehicle and fare estimation.
- 6. Real time updates of driver's status.
- 7. Feedback and complaint portal

## 2.3 User Classes and Characteristics

- Passenger: User who book and track rides
- Driver: Users who accept ride requests and provide transport services.
- Admin: User managing system data and overseeing operations.

## 2.4 Operating Environment

- Mobile devices → Android/iOS for passengers and drivers
- Web Application. → for admin use.
- Dependencies → Requires internet access, GPS functionality, support for third party services (API calling, Payment gateways)

## 2.5 Design and Implementation Constraints

- Compliance with data privacy laws.
- Secure handling of payment information (PCI-DSS)
- Reliable GPS data for accurate tracking.

## 2.6. User Documentation.

- User guide for passengers and drivers → include all acceptance to company's terms and conditions as per country law and labour laws.
- Admin manual detailing system controls and monitoring tools. All technology framework management and feature updation, rolling of updates successfully to customers.

## 2.7. Assumptions and Dependencies

- Users must have access to mobile Internet.
- GPS data is available of customer & driver.
- Seamless working of payment gateways

# 3. External Interface Requirements.

## 3.1. User Interface

- UI for passenger : A mobile app with booking, payment and tracking functionalities. User will register and login, fill all personal details, choose pickup location, choose drop point and vehicle type.
- UI for Driver : A dedicated interface for Drivers. Here, driver will register all car details, his/her personal info, all necessary valid identification details. The app will allow driver to connect with passengers in proximity with ready to travel.

- Admin UI

A web based interface for managing all passenger and driver data and monitoring for future reference and maintenance. Administrator will be responsible for providing seamless user experience to passengers and drivers. Keep updated record of driver and ranking driver's performance based on their ratings by customers. Includes management of payment and keeping all essential components connected and secured.

### 3.2 Hardware Interfaces

- The mobile app will interface with GPS hardware for location tracking.
- Satellite communication maintained by Google Maps will be used for detecting trip start and end.

### 3.3 Software Interfaces

- Integration with Google Maps API for navigating and location tracking.
- Integration with Paypal / Stripe for processing payments. Handle payments between accurate passengers. Processing refunds after verification. Fee deduction for ride cancelling. Transaction management with refunds and disputes. Maintaining security and encryption during transactions. PCI DSS standards.
- Integration with e-mail and SMS services Managing SMS and e-mail for passenger trips, after ride booking, payment slip and promotions.

### 3.4 Communication Interfaces

- Internet Protocols : HTTP/ HTTPS for communication between mobile devices and the server. In-app chat services for query and chats. Real time messaging & message history maintenance. Quick response with less delay.
  - Data Encryption : For secure data transmission between user and the server. Security and firewall layers with encrypted data to secure privacy and prevent hacking and data loss.
- 

## 4. System Features

### 4.1 Feature 1 : User Registration and Authentication

#### 4.1.1. Description and Priority

Users must register using their mobile phone or e-mail ID. A high priority feature as it is the first interaction with system.

#### 4.1.2 Stimulus / Response Sequences

- User submits registration details
- System sends OTP for verification.
- User enters OTP to complete registration

#### 4.1.3 Functional Requirements

Req. ① System must send OTP for verification.

Req. ② User must authenticate using their email ID or mobile phone number.

## 4.2 Feature 2 : Cab Booking & Ride Tracking

### 4.2.1 Description and Priority

User can book cab by entering pick up and drop off locations. The system assigns the nearest neighbouring driver. This feature is critical to system functionality.

### 4.2.2 Stimulus / Response Sequences

- User selects pick up location.
- User tells drop off location / area.
- User selects vehicle type, bike, auto, Sedan, minil.
- As per traffic and time, distance, fare estimate is displayed.
- User is asked and booking is found.
- System notifies nearest driver.
- Driver accepts request for trip and confirms.
- User can track live location and car details.

### 4.2.3 Functional Requirements

Req 3 System must show real time location of cab on the map.

Req 4. System must allow users to cancel ride before driver reaches pickup point. Cancellation charges shall be applied.

## 4.3 In-App Payment Processing : Feature 3

### 4.3.1 Description and Priority.

Users must be able to make payments through various methods including credit / debit cards, digital wallets and online payment gateways (Paypal, stripe). This is a high-priority feature for completing transactions seamlessly.

### 4.3.2 Stimulate / Response Sequences

- User selects payment method after ride completion.
- System processes the payment & confirms.
- A receipt is generated and sent to the user.

### 4.3.3 Functional Requirements.

Req 5 : System must support multiple payment options (cards, wallets, banking)

Req 6 : If transaction fails, a notification should be provided to passenger for transaction failure.

Req 7 : All payment transactions should be done through secure, PCI DSS compliant gateways.

## 4.4 Feature 4: Driver Rating and Feedback

### 4.4.1. Description and Priority

After completing a ride, passengers can rate the driver and provide feedback. Drivers can also rate passengers. This is a medium priority feature but critical for maintaining service quality.

#### 4.4.2 Stimulus / Response Sequence.

- After trip completion, the passenger is prompted to rate driver on a scale of 1-10.
- The driver is also prompted to rate the passenger.
- Both can leave optional comments.

#### 4.4.3 Functional Requirements.

- Req 8: Both passenger & driver must be allowed to rate each other.
- Req 9: System must store and keep record of ratings of each driver for future reference.
- Req 10: System must notify admin if a user receives consistently low ratings.

### 4.5 Driver Assignment and Details

#### 4.5.1 Description and Priority

When a user books a ride, the system must automatically assign the nearest available driver by first looking in range of 5 km radius. This is a critical, high priority feature for ensuring quick and efficient ride assignments.

#### 4.5.2 Stimulus / Response Sequences

- User confirms ride request.
- System identifies and notifies nearby drivers.
- Driver either accept or decline the ride.
- System again searches for other nearby drivers.
- If accepted, give details to customer and show live location.

#### 4.5.3 functional Requirements

- Req 11: System must assign nearest driver to passenger's pickup location.
- Req 12: If customer cancels ride after driver acceptance, then cancellation charges must be paid by the customer. Same goes for driver after acceptance if ride is cancelled.
- Req 13: System must reassign ride to another driver if driver cancels ride or does not respond in 30 seconds.

### 5. Other non-functional Requirements

#### 5.1 Performance Requirements

- NFR-1 : The system must handle up to 10,000 concurrent users with no degradation in response time.
- NFR-2 : Cab booking and confirmation should be completed within 5 seconds under normal conditions.
- NFR 3 : The system must be able to scale dynamically during peak times to handle increased user demand without downtime.
- NFR 4: Real time updates should be provided, refreshed in every 2 minutes.

#### 5.2 Safety Requirements.

- NFR 5: System must ensure secure handling of user data including personal information, payment details, location history, to prevent unauthorized access,

- NFR 6: Emergency alerts or options should be available within the app, allowing users to contact authorities in case of unsafe situations during ride.

### 5.3 Security Requirements

- NFR 7: All data transferred between client and the server must be encrypted using SSL.
- NFR 8: User authentication must follow industry standards for securing passwords and OTP validation.

### 5.4 Software Quality Attributes

- Reliability: The system must have 99.9% uptime.
- Usability: The user interface must be intuitive, requiring no more than 3 clicks to book a ride.
- Scalability: The system must scale to accommodate a growing user base, both in terms of numbers of users and geographic regions.
- Adaptability: The system should be adaptable to future requirements, such as adding new lines of features like ride-sharing or vehicle rental service.
- Testability: Automated and manual testing should be possible, ensuring bugs and issues can be detected early in the development cycle.

## 5.5 Business Rules

- Ride Cancellation: Passengers can cancel the ride within 5 minutes of booking without penalties.
- Driver Commission: Drivers receive 80% of the fare, with the remaining 20% going to the platform.
- Rating: Drivers with average Rating below 5 must be temporarily suspended or required to attend review session.
- Cancelling Ride: If customer cancels ride after booking, then cancellation charge will be applicable.

## 6. Other Requirements

### 6.1 Database Requirements

- System must store all user data, ride history, payment details and all other data in a relational database for efficient querying and management.
- Data base must be able to handle minimum of 1 million ride transactions per month.

### 6.2 Internationalized Requirements

- The system must support multiple languages, including English and Hindi, with ability to expand to additional languages as needed.

- The app's user interface should dynamically adjust for local specific time date formats and currency.
  - Regional payment methods like UPI in India must be supported for users based in those regions.
- 

#### 6.3 Legal and Regulatory Requirements

- The system must comply with country IT act laws to protect user data privacy and security.
  - System must meet local transportation regulations and licensing requirements, ensuring all drivers on platform are properly licensed & verified.
  - All transactions and contracts between the platform and users must be legally binding and recorded for future reference.
- 

#### 6.4 Reusability Objectives

- The code base must be designed to support future enhancements like ride sharing, package delivery services with modular components for easy use.
- Components like payment gateways and location services must be designed as independent services, allowing for their integration with other systems or platforms.

## Appendix A : Glossary

- GPS : Global Positioning System
- OTP : One Time Password
- ETA : Estimated Time of Arrival
- SMS : Short Messaging Service

## Overview of System

