# Assignment 1

# Business Requirements Document (BRD)

**Project Name:** Dairy Supply Chain Management System
**Project ID:** DSCMS-2025
**Version ID:** V1.0
**Author:** [KANIKA JAGTAP]
**Date:** [04/07/2025]

## 1. Document Revisions

| Date | Version | Document Changes |
|---|---|---|
| 04/07/2025 | 0.1 | Initial Draft Prepared |

## 2. Approvals

| Role | Name | Title | Signature | Date |
|---|---|---|---|---|
| Project Sponsor | Rajesh | Director | | 04/07/2025 |
| Business Owner | Anjali | Supply Chain Head | | 04/07/2025 |
| Project Manager | Sunil | PM | | 04/07/2025 |
| Architect | Aryan | System Architect | | 04/07/2025 |
| Development Lead | Priya | Dev Lead | | 04/07/2025 |

| | | | | |
|---|---|---|---|---|
| QA Lead | Tanmay | QA Lead | | 04/07/2025 |

## 3. RASCI Chart

| Name | Position | R | A | S | C | I |
|---|---|---|---|---|---|---|
| KANIKA | Business Analyst | ✓ | | | ✓ | ✓ |
| Sunil | Project Manager | | ✓ | | ✓ | ✓ |
| Priya | Dev Lead | | | ✓ | ✓ | |
| Tanmay | QA Lead | | | ✓ | ✓ | |
| Anjali | Business Owner | | ✓ | | | ✓ |

## 4. Introduction

### *4.1 Business Goals*

To develop an integrated system for:

- Real-time inventory management across plants and warehouses.
- Optimized delivery logistics for quickest customer order fulfillment.

### *4.2 Business Objectives*

- Track inventory levels across locations in real-time.
- Forecast demand based on historical data and trends.
- Suggest optimal delivery route and warehouse based on proximity and stock.
- Monitor vehicle movement and delivery time.
- Alert system for low stock and delays.

### *4.3 Business Rules*

- Inventory updates must be synchronized every 15 minutes.

- Orders above ₹10,000 require warehouse manager approval.
- Perishable items (ice-cream) have shelf-life constraints (7 days).
- Delivery must be assigned within 15 minutes of order confirmation.
- Real-time tracking of all outbound vehicles is mandatory.

### *4.4 Background*

The company operates multiple plants and warehouses across India. Manual inventory records, delayed dispatches, and delivery inefficiencies are leading to wastage and customer dissatisfaction.

### *4.5 Project Objective*

To deliver a digital platform that ensures stock accuracy and timely delivery, enabling competitive advantage and customer satisfaction. Target completion in 7 months within a ₹20 lakh budget.

## 5. Project Scope

### *5.1 In-Scope*

- Inventory Management Module (real-time stock tracking)
- Delivery Optimization Module (route suggestions, vehicle assignment)
- Order Management System
- Dashboard for plant/warehouse managers
- Alert Notifications and Stock Threshold Triggers
- Integration with GPS tracking system

### *5.2 Out-of-Scope*

- Mobile App for Customers (Phase 2)
- Hardware upgrades for logistics fleet
- Payroll and HR functions
- Vendor onboarding workflows

## 6. Assumptions

- Warehouses are GPS-enabled for tracking.
- Employees will be trained before go-live.
- All plants follow a standardized product coding system.
- Stakeholders will give timely input during sprints.

## 7. Constraints

- Budget capped at ₹20 lakhs.
- System must handle 5000 concurrent users.
- Deployment must be done in phases.
- System must be usable in Hindi and English.

## 8. Risks

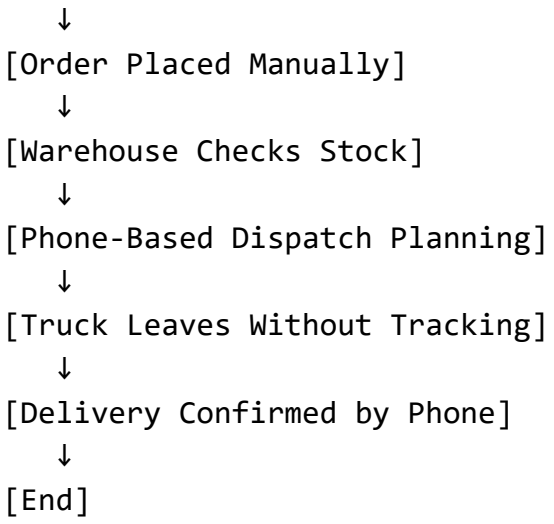
| Type | Description | Likelihood | Impact | Mitigation |
|---|---|---|---|---|
| Operational | Network issues in remote warehouses | Medium | High | Offline sync backup setup |
| Technical | API failure with GPS system | Low | High | Use fallback routing logic |
| Requirement | Scope creep due to evolving expectations | High | High | Scope freeze post-SRS |
| Data | Incorrect stock updates | Medium | High | Implement auto reconciliation |

## 9. Business Process Overview

### *9.1 AS-IS Process (Manual & Delayed)*

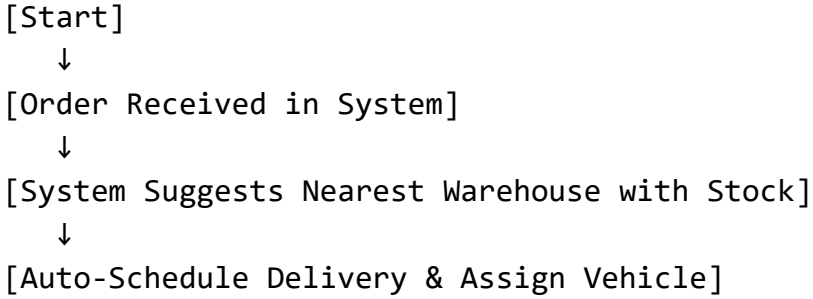
The current **AS-IS process** of the dairy supply chain is largely **manual, uncoordinated, and delayed**, resulting in inefficiencies and lack of real-time visibility. Orders are placed manually, often via phone or

paperwork, leading to miscommunication and delays. The warehouse manually checks stock availability, and dispatch planning is done over the phone, lacking optimization. Trucks leave without any GPS tracking, and delivery confirmation is also handled through phone calls, making it difficult to monitor the supply chain or address issues proactively.




### *9.2 TO-BE Process (Automated & Tracked)*

In contrast, the **TO-BE process** proposes an **automated, streamlined, and technology-driven** system that significantly enhances operational efficiency. Orders are captured digitally in the system, which automatically checks stock levels across warehouses and suggests the nearest warehouse with available stock. Delivery is scheduled by the system, and vehicles are auto-assigned, integrated with **real-time GPS tracking**. Customers receive automatic updates, including **Estimated Time of Arrival (ETA)** and delivery status. Stock is automatically deducted and updated in the system, ensuring accurate inventory management and full end-to-end visibility of the supply chain.



```
   ↓
[GPS Tracking Enabled for Vehicle]
   ↓
[Customer Gets ETA and Delivery Status]
   ↓
[Stock Auto-Deducted & Updated]
   ↓
[End]
```

## 10. Business Requirements

| ID | Description | Priority | Category |
|---|---|---|---|
| BR001 | Real-time inventory visibility | High | Functional |
| BR002 | Automated order routing based on location & stock | High | Functional |
| BR003 | GPS-based delivery tracking | High | Functional |
| BR004 | Alerts for low stock & delays | Medium | Functional |
| BR005 | Integration with existing ERP for order sync | High | Integration |
| BR006 | Dashboards for stock levels, delivery stats | High | Reporting |
| BR007 | Role-based access control | Medium | Non-Functional |
| BR008 | Secure login & multilingual support | Medium | Non-Functional |

## 11. Appendices

### *11.1 Acronyms*

| Acronym | Full Form |
|---|---|
| ERP | Enterprise Resource Planning |

| | |
|---|---|
| GPS | Global Positioning System |
| ETA | Estimated Time of Arrival |

### *11.2 Glossary*

- **Perishable Inventory**: Products with a limited shelf life like milk or ice-cream.
- **Inventory Reconciliation**: Matching physical stock with digital records.

### *11.3 Related Documents*

- Legacy Process Audit Report
- Inventory Master Sheet (Current)
- Route Mapping Plan (Initial)

### *Key Features:*

- Inventory Management Module
- Order Management System
- Delivery Route Optimizer (Google Maps API Integration)
- Warehouse-Plant Synchronization
- Role-Based Dashboards
- Mobile App for Delivery Personnel

### *Functional Requirements:*

- Real-time inventory updates
- Barcode scanning for products
- Auto re-order thresholds
- Delivery time estimation
- Order placement, modification & tracking
- Customer notifications via SMS/email

### *Non-Functional Requirements:*

- High availability
- Scalable architecture
- Secure access control

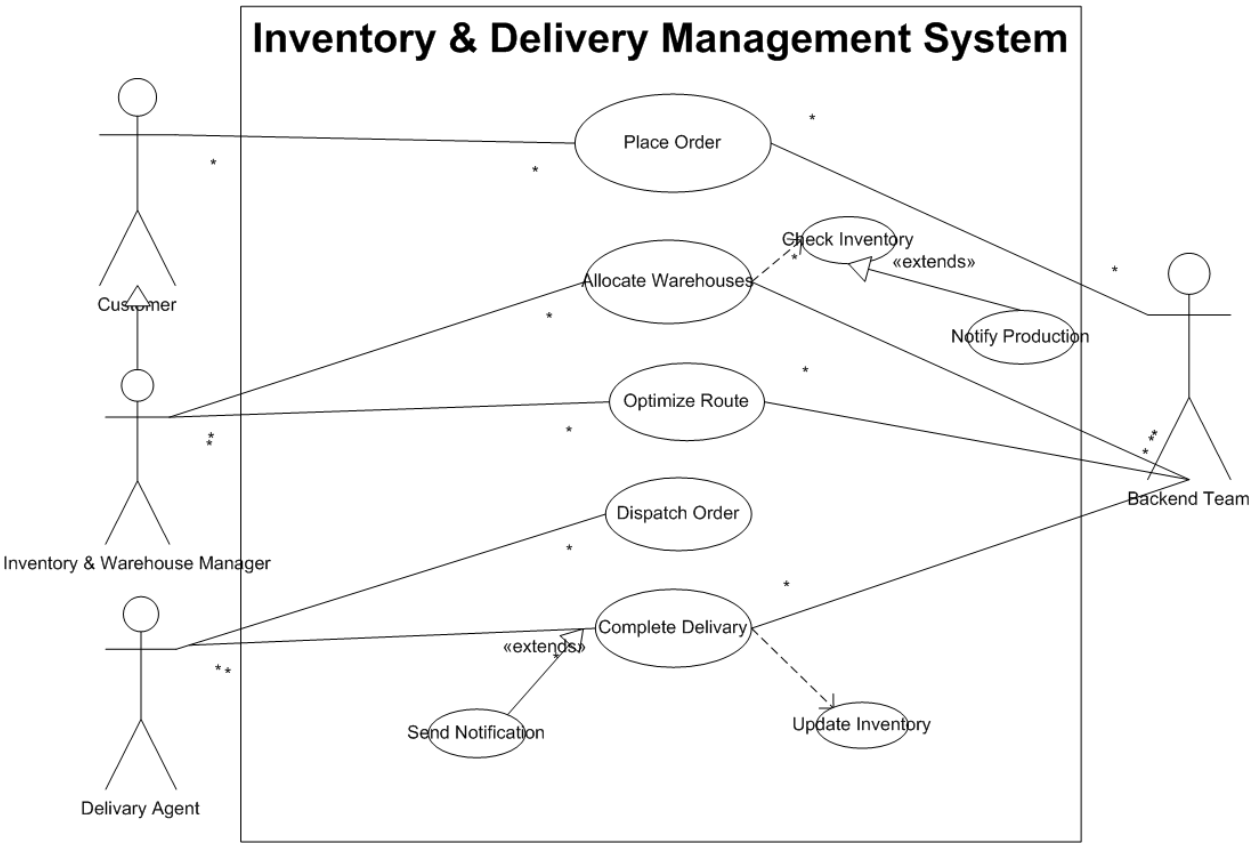
- Integration with existing ERP

### *Technology Stack:*

- Backend: Node.js / Python (Django)
- Frontend: React / Angular
- Database: PostgreSQL / MongoDB
- Mobile App: Flutter
- APIs: Google Maps, Twilio (notifications), ERP connectors

### *Development Plan:*

| Phase | Timeline (Weeks) | Resources |
|---|---|---|
| Requirement Analysis | 2 | 1 BA, 1 PM |
| Design | 2 | 1 UI/UX Designer, 1 Architect |
| Development Phase I | 4 | 3 Devs (Frontend+Backend), 1 QA |
| Development Phase II | 3 | 2 Devs, 1 QA |
| Testing & UAT | 2 | 2 QA, 1 BA |
| Deployment | 1 | 1 DevOps |
| Support & Training | Ongoing | 1 Trainer, 1 Support Engineer |




# ☑ Assignment 2

## 1. Introduction Letter

**Subject:** Introduction – Business Analyst for Your Project

Dear Pankaj,

I hope this message finds you well. My name is Kanika Jagtap, and I have been assigned as the Business Analyst for your upcoming project. I'll be working closely with you and your team to understand your business objectives, gather detailed requirements, and ensure that we build a solution that aligns perfectly with your vision.

I'm looking forward to beginning our discovery sessions and helping translate your ideas into a robust, scalable, and efficient system.

Please feel free to reach out to me with any questions or to schedule our initial discussion.

Best regards,
Kanika Jagap
Business Analyst
KJ1234@gmail.com

9826587952

## 2. BRD + SRS – Online Ticketing System

## BRD – Online Ticketing System

## Business Requirements Document (BRD)

**Project Name**: Online Ticketing System
**Project ID**: OTS-2025
**Version ID**: V1.0
**Author**: [Kanika Jagtap]
**Date**: [07/07/2025]

## 1. Document Revisions

| Date | Version | Document Changes |
|---|---|---|
| 07/07/2025 | 0.1 | Initial Draft Prepared |

## 2. Approvals

| Role | Name | Title | Signature | Date |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Project Sponsor | Meera | Operations Director | | 07/07/2025 |
| Business Owner | Rahul | Ticketing Head | | 07/07/2025 |
| Project Manager | Neha | PM | | 07/07/2025 |
| Architect | Karan | System Architect | | 07/07/2025 |
| Development Lead | Sneha | Dev Lead | | 07/07/2025 |
| QA Lead | Tanmay | QA Lead | | 07/07/2025 |

## 3. RASCI Chart

| Name | Position | R | A | S | C | I |
|---|---|---|---|---|---|---|
| Kanika Jagtap | Business Analyst | ✓ | | | ✓ | ✓ |
| Neha | Project Manager | | ✓ | | ✓ | ✓ |
| Sneha | Dev Lead | | | ✓ | ✓ | |
| Tanmay | QA Lead | | | ✓ | ✓ | |
| Rahul | Business Owner | | ✓ | | | ✓ |

## 4. Introduction

### *4.1 Business Goals*

To build a centralized and user-friendly online platform for ticket booking, enabling real-time booking, cancellation, and tracking across entertainment and transport sectors.

### *4.2 Business Objectives*

- Enable users to book, cancel, and modify tickets online.

- Show real-time seat availability.
- Integrate with payment gateways for secure transactions.
- Provide e-tickets via email/SMS.
- Allow partner onboarding (cinemas, buses, events).

### *4.3 Business Rules*

- Ticket booking closes 30 minutes before the event/start time.
- One user can book up to 6 tickets per event/trip.
- Refunds applicable only before 24 hours of event time.
- Payment confirmation must occur within 5 minutes or the session expires.

### *4.4 Background*

Currently, customers rely on different platforms or offline modes to book tickets. The need for a unified, scalable, and mobile-friendly system has emerged to meet growing digital expectations.

### *4.5 Project Objective*

To develop a reliable and secure online ticketing system accessible via web and mobile, completed within 6 months and a budget of ₹25 lakhs.

## 5. Project Scope

### *5.1 In-Scope*

- Ticket booking module
- Seat selection and availability checker
- Secure payment integration
- E-ticket generation and email/SMS notification
- Partner dashboard for managing events/shows
- Admin panel for overall system control

### *5.2 Out-of-Scope*

- Offline booking at physical counters
- Transportation logistics or event management services
- Loyalty and rewards program (Phase 2)

## 6. Assumptions

- All partners will provide updated schedule and pricing.
- Internet access is available for all users.
- SMS and email gateways will be integrated via third-party APIs.
- Users will self-register and maintain their profiles.

## 7. Constraints

- Budget limited to ₹25 lakhs.
- System must handle 10,000+ concurrent users.
- Must support web, Android, and iOS platforms.
- Go-live deadline in 6 months.

## 8. Risks

| Type | Description | Likelihood | Impact | Mitigation |
|---|---|---|---|---|
| Operational | Partner delay in sharing schedule | Medium | High | Agreement with SLA |
| Technical | Payment gateway API failure | Low | High | Backup payment provider setup |
| Functional | User overload during peak bookings | High | High | Load balancing and scaling |
| Data | Incorrect seat allocation | Medium | High | Real-time lock mechanism on seats |

## 9. Business Process Overview

### *9.1 AS-IS Process (Manual / Fragmented)*

The current **AS-IS process** of booking tickets is **manual, fragmented, and time-consuming**, leading to a poor user experience. Users often need to visit multiple booking platforms or physical counters to find available events or shows. Seat availability is not shown in real-time, requiring users to manually inquire or refresh multiple pages. Payment may be done offline or through unreliable gateways, and ticket confirmation is sent manually via email, often causing delays or errors in communication.

```
[Start]
  ↓
[User visits multiple booking sites]
  ↓
[Manual seat checking]
  ↓
[Offline or online payment]
  ↓
[Manual confirmation sent via email]
  ↓
[End]
```

### *9.2 TO-BE Process (Unified & Digital)*

The proposed **TO-BE process** transforms the system into a **unified, seamless, and fully digital platform**. Users can log in to a centralized Online Ticketing Portal, search for events, shows, or trips, and instantly view **real-time seat availability**. They can then select their preferred seat, complete payment through secure online gateways, and immediately receive a system-generated e-ticket. The system automatically updates the seat status, ensuring no overbooking or manual intervention is needed. This end-to-end digital process enhances user convenience, accuracy, and operational efficiency.

```
pgsql
CopyEdit
```

```
[Start]
  ↓
[User logs into Online Ticketing Portal]
  ↓
[Searches for Event/Show/Trip]
  ↓
[System shows real-time seat availability]
  ↓
[User selects seat and pays online]
  ↓
[Ticket generated and sent to user]
  ↓
[Seat status auto-updated]
  ↓
[End]
```

## 10. Business Requirements

| ID | Description | Priority | Category |
|---|---|---|---|
| BR001 | Real-time seat availability | High | Functional |
| BR002 | Online booking with seat selection | High | Functional |
| BR003 | Secure payment gateway integration | High | Integration |
| BR004 | E-ticket generation and SMS/email notifications | High | Functional |
| BR005 | User registration and profile management | Medium | Functional |
| BR006 | Role-based access for admins and partners | Medium | Non-Functional |
| BR007 | Multilingual support (English/Hindi) | Medium | Non-Functional |
| BR008 | Booking history and refund tracking | Medium | Reporting |

## 11. Appendices

### *11.1 Acronyms*

| Acronym | Full Form |
|---|---|
| SMS | Short Message Service |
| ETA | Estimated Time of Arrival |
| API | Application Programming Interface |

### *11.2 Glossary*

- **E-ticket**: Digitally issued ticket sent via email or SMS.
- **Refund Window**: Timeframe in which user can cancel and get a refund.
- **Seat Lock**: Temporary hold on selected seat until payment is done.

### *11.3 Related Documents*

- UI Wireframes (Initial)
- Payment Gateway Integration Plan
- Event Partner Onboarding Guide

## Key Features:

- Event, Movie, and Travel Ticket Booking
- Real-time Seat Lock and Selection
- Payment Gateway Integration (Razorpay/Paytm)
- Partner Dashboards
- Notification via Email/SMS
- Booking Reports and Admin Controls

## Functional Requirements:

- Search, filter, and book tickets

- View booking history and cancel/refund
- Partner event management dashboard
- Auto seat allocation and lock
- Alerts for booked/available seats
- Multilingual interface

## Non-Functional Requirements:

- High system availability (99.9%)
- Data encryption and secure login
- Scalability to handle 10k+ users
- Cross-platform compatibility
- Backup and disaster recovery

## Technology Stack:

- **Frontend**: React / Angular
- **Backend**: Node.js / Python (Django)
- **Database**: PostgreSQL / MySQL
- **Mobile App**: Flutter
- **APIs**: Razorpay/Paytm (Payments), Twilio (SMS), Mailgun (Email)

## Development Plan:

| Phase | Timeline (Weeks) | Resources |
|---|---|---|
| Requirement Analysis | 2 | 1 BA, 1 PM |
| UI/UX & Architecture | 2 | 1 Designer, 1 Architect |
| Development Phase I | 4 | 3 Devs (Frontend + Backend), 1 QA |
| Development Phase II | 3 | 2 Devs, 1 QA |

| | | |
|---|---|---|
| Testing & UAT | 2 | 2 QA, 1 BA |
| Deployment | 1 | 1 DevOps |
| Support & Training | Ongoing | 1 Trainer, 1 Support Engineer |

***Objective:***

To build an online platform for users to book event tickets (movies, concerts, sports, etc.) seamlessly.

***Business Needs:***

- Sell event tickets online
- Integrate with payment gateway
- Mobile-friendly user experience
- Admin panel to manage events

***Stakeholders:***

- Customers (ticket buyers)
- Event organizers
- Admins
- Payment gateway providers

***Scope:***

- Event Listing & Filtering
- Seat Selection (if applicable)
- Payment & Ticket Generation
- Email/SMS Confirmation
- Refund & Cancellation

# SRS – Online Ticketing System

## ☑ Project Title: Online Ticket Booking System

**Version:** 1.0
**Prepared By:** Kanika Jagtap (Business Analyst)
**Date:** 04-July-2025

## 1. Introduction

### *1.1 Purpose*

The purpose of this document is to define the software requirements for an **Online Ticket Booking System**, which allows users to browse, select, and book tickets for movies, events, or transportation services.

### *1.2 Scope*

The system will offer real-time booking, seat selection, online payment, ticket generation, and cancellation options for end-users, while providing admins and partners with tools to manage events, inventory, and reports.

### *1.3 Definitions, Acronyms and Abbreviations*

| Term | Description |
|---|---|
| OTP | One-Time Password |
| UI | User Interface |
| DB | Database |
| API | Application Programming Interface |

### *1.4 References*

- ISO 25010 Software Quality Standards

- Payment Gateway API Documentation (Razorpay/Paytm)

### *1.5 Overview*

This document outlines the functional and non-functional requirements, interface descriptions, and system models for the ticket booking platform.

## 2. Overall Description

### *2.1 Product Perspective*

The system will be a web and mobile-responsive platform, integrated with third-party payment services, real-time seat inventory, and event management systems.

### *2.2 Product Functions*

- Search and filter shows/events
- Real-time seat layout and selection
- User registration and login
- Payment integration
- Ticket confirmation and download
- Admin backend to manage listings

### *2.3 User Classes and Characteristics*

| User Type | Description |
|---|---|
| Guest User | Can view events without login |
| Registered User | Can book, cancel, and view tickets |
| Admin | Can add events, manage bookings, and view reports |
| Vendor/Partner | Manages their event listings |

### *2.4 Operating Environment*

- Web: Chrome, Firefox, Edge (Desktop & Mobile)

- Backend: Linux/Windows Server with MySQL & Node.js/Django
- Frontend: HTML5, React/Angular

### *2.5 Design Constraints*

- GDPR-compliant personal data handling
- Payment gateway PCI-DSS compliance
- Max 3s page load time

### *2.6 User Documentation*

- Online help (FAQs, How-To Videos)
- User guide PDF
- Admin manual

### *2.7 Assumptions and Dependencies*

- Payment gateway integration is functional and available
- Event providers supply real-time seat inventory
- Users have a stable internet connection

## 3. System Features (Functional Requirements)

| ID | Feature | Description | Priority |
|---|---|---|---|
| SF001 | User Registration/Login | Users register via email/OTP or social login | High |
| SF002 | Event Search | Users can browse and filter events by date, type, or location | High |
| SF003 | Seat Selection | Real-time seat map and selection for movie/event | High |
| SF004 | Payment Integration | Secure online payment using gateway APIs | High |
| SF005 | Ticket Generation | Generates a digital ticket after booking | High |
| SF006 | Cancellation & Refund | Users can cancel tickets and apply refund policy | Medium |

| | | | |
|---|---|---|---|
| SF007 | Admin Panel | Admin can add/edit/delete events, prices, and venues | High |
| SF008 | Email & SMS Notification | Confirmation sent post-booking | Medium |

## 4. External Interface Requirements

### *4.1 User Interfaces*

- Responsive Web UI
- Mobile-first design
- Accessibility compliant (font scaling, ARIA tags)

### *4.2 Hardware Interfaces*

- Not applicable (web-based)

### *4.3 Software Interfaces*

| System | Interface | Description |
|---|---|---|
| Payment Gateway | REST API | For processing online payments |
| Email/SMS Gateway | SMTP/REST | For sending confirmations |
| Ticketing DB | MySQL/PostgreSQL | Stores user and event data |

### *4.4 Communication Interfaces*

- HTTPS protocol for secure communication
- REST APIs for internal microservices

## 5. Non-Functional Requirements

| Type | Description |
|---|---|
| Performance | System should support 10,000+ concurrent users |
| Availability | 99.9% uptime required |

| | |
|---|---|
| Security | SSL encryption, input validation, token-based authentication |
| Usability | User-friendly interface with fewer than 3 clicks to book |
| Maintainability | Modular code for easy enhancements |
| Localization | Supports English, Hindi, and regional languages |
| Scalability | Horizontal scaling support on cloud platforms |

## 6. System Models and Diagrams

- **Use Case Diagram**
  Actors: Guest, Registered User, Admin
  Use Cases: Browse Events, Register/Login, Book Ticket, Make Payment, Generate Ticket, Cancel Ticket, Admin Event Management
- **Activity Diagram**
  Flow:
  [Start] → Search Event → View Seats → Select Seat → Make Payment → Generate Ticket → [End]
- **ER Diagram**
  Entities: User, Event, Booking, Payment, Ticket, Venue

## 7. Appendices

### *7.1 Glossary*

- **Booking ID**: Unique reference number for a transaction
- **Event Provider**: Organization hosting the event/show
- **Refund Window**: Time period in which cancellation is allowed

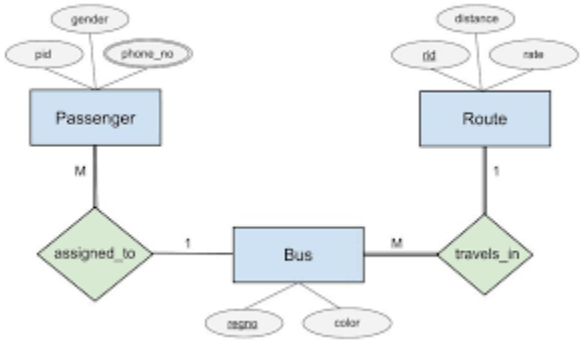
### *7.2 Acronyms*

| Acronym | Full Form |
|---|---|
| OTP | One-Time Password |
| UI | User Interface |
| DB | Database |

### *7.3 References*

- Event Provider API Docs
- Payment Gateway Docs

## 3. ERD – Ticketing Lifecycle




**Entities:**

- Passenger(UserID, Name, Email)
- Bus (Name, DateTime)
- **Venue**(VenueID, Name, Location)
- **Seat**(SeatID, VenueID, SeatNumber)
- **Ticket**(TicketID, UserID, EventID, SeatID, QRCode, Status)
- **Payment**(PaymentID, TicketID, Amount, Timestamp, Status)

# 🎯 User Stories with Acceptance Criteria – Online Ticketing System

## 1. User Registration

**As a** new user,
**I want to** register on the platform using my email and password,
**So that** I can access the ticket booking features.

**Acceptance Criteria:**

- Registration form must include name, email, mobile number, and password.
- Email should be unique.
- Password should follow strong format rules (min 8 characters, etc.).
- A confirmation email is sent upon successful registration.

## 2. User Login

**As a** registered user,
**I want to** log in using my email and password,
**So that** I can book tickets and view my bookings.

**Acceptance Criteria:**

- User can log in with valid credentials.
- System should display an error message on invalid credentials.
- After login, user is redirected to the dashboard.

## 3. Browse Events

**As a** user,
**I want to** browse available events by category, location, or date,
**So that** I can explore tickets I might be interested in.

**Acceptance Criteria:**

- Events are listed with title, location, date, and price.
- Filter options for category (e.g., movie, concert), city, and date.
- Sorting by price or popularity is available.

## 4. View Event Details

**As a** user,
**I want to** view complete details of a selected event,
**So that** I can decide whether to buy tickets.

**Acceptance Criteria:**

- Event details must include description, date/time, venue, ticket price, and seating layout (if applicable).
- Option to select the number of tickets.

## 5. Select Seats

**As a** user,
**I want to** select my preferred seats from a seating chart,
**So that** I can choose where I want to sit.

**Acceptance Criteria:**

- Available seats are shown in a visual layout.
- Selected seats are highlighted.
- Already booked seats are marked as unavailable.
- Seats cannot be held for more than 10 minutes without checkout.

## 6. Add to Cart

**As a** user,
**I want to** add selected event tickets to my cart,
**So that** I can book them after reviewing.

**Acceptance Criteria:**

- Tickets added to cart remain available for a limited time.
- Cart displays event name, date, seat numbers, and total cost.
- Option to update or remove items from cart.

## 7. Checkout and Payment

**As a** user,
**I want to** pay securely for the tickets I've selected,
**So that** I can confirm my booking.

**Acceptance Criteria:**

- Payment options include credit/debit card, UPI, and net banking.
- On successful payment, booking is confirmed.
- Failed transactions display a proper error message.
- Users receive a booking confirmation email/SMS.

## 8. Download E-Ticket

**As a** user,
**I want to** download my e-ticket after payment,
**So that** I can use it to attend the event.

**Acceptance Criteria:**

- User can download ticket in PDF format.
- Ticket must contain QR code, event details, and seat info.
- User receives a copy of the ticket in their registered email.

## 9. Cancel Booking

**As a** user,
**I want to** cancel my ticket before the event date,
**So that** I can get a refund (if allowed).

**Acceptance Criteria:**

- Tickets can be cancelled up to 24 hours before the event.

- Cancellation policy is displayed.
- Refund processed as per terms (full/partial).
- User gets confirmation of cancellation via email.

## 10. View Booking History

**As a** user,
**I want to** view my past bookings,
**So that** I can keep track of events I've attended or cancelled.

**Acceptance Criteria:**

- List of all past bookings with date, event name, and status.
- Option to filter by date or event type.
- Link to download ticket (if still valid) or view cancellation details.