END-TO-END PROJECT SRS DOCUMENT

Team Members:	Project Guide
21BQ1A05D3 – TM1 21BQ1A05E5 – TM2 21BQ1A05F2 – TM3 21BQ1A05G0 – TM4 21BQ1A05G3 – TM5 21BQ1A05G6 – TM6 21BQ1A05G7 – TM7 21BQ1A05H3 – TM8 21BQ1A05H8– TM9 21BQ1A05I2 – TM10 21BQ1A05I7—TM11 21BQ1A05I8 --- TM12 22BQ5A0515 -- TM13	Dr.T. K. Rao

TABLE OF CONTENTS:

4.External Interface Requirements

4.1 User Interfaces

4.2 Hardware Interfaces

4.3 Software Interfaces

4.4 Communication Interfaces

5. System Architecture

6.System and Data Architecture

6.1 System Architecture

6.2 Data Architecture

6.3 Database Schema

7. Validation and Verification

- 7.1 Testing Requirements
- 7.2 Acceptance Criteria

1.INTRODUCTION:

1.1 Purpose

The purpose of this document is to provide a detailed description of the requirements for the "HAPPY STAY" Hotel Reservation System. This SRS outlines the functionalities, features, and constraints of the system, serving as a guide for the development team and stakeholders to ensure that the final product meets the desired objectives.

1.2 Scope

The "HAPPY STAY" Hotel Reservation System is a web-based application designed to facilitate the booking process for guests and streamline hotel management operations. The system will allow users to register, log in, browse hotel rooms with filters, book rooms, pay online, manage bookings, view facilities, and contact hotel management. The hotel admin will have the ability to manage bookings, update room information, and handle cancellations.

1.3 Overview

This document is organized to provide an overview of the system, followed by detailed descriptions of the requirements, system architecture, design, and validation criteria.

2. Overall Description

2.1 Product Perspective

The "HAPPY STAY" Hotel Reservation System is a standalone web application. It interacts with external payment gateways (e.g., Paytm) and email services for communication with hotel management. The system is designed to be scalable and flexible, supporting multiple user roles and providing real-time updates on room availability

2.2 Product Features

- User registration and login
- Room browsing with advanced filters (check-in/out dates, room type, price range)
- Online room booking
- Payment integration with Paytm gateway
- Booking management (view, modify, cancel)
- Facility viewing
- Direct communication with hotel management via email
- Admin functionalities: booking management, order acceptance, room details management

2.3User Classes and Characteristics

Guests: Users who book rooms, manage their reservations, and communicate with the hotel.
Hotel Admin: Users responsible for managing the hotel's operations, including booking management and room updates.

2.4Operating Environment

Web application accessible via modern browsers (Chrome, Firefox, Edge)

Server hosting the application (Localhost)
Database server (MySQL)

2.5 Design and Implementation Constraints

- The system must comply with data protection regulations (e.g., GDPR).
- The Paytm payment gateway integration must adhere to the gateway's API specifications.
- The system should be optimized for performance and handle concurrent users.

2.6 Assumptions and Dependencies

- The hotel will provide updated room and facility information for the system.
- The Paytm gateway and email services will remain operational and available.

3. Specific Requirements

3.1 Functional Requirements

- User Registration and Login: Users must be able to create accounts and log in using valid credentials.
- Room Browsing and Filters: The system should allow users to search and filter rooms based on specific criteria.
- Booking Process: Users must be able to select a room, enter booking details, and confirm the booking.
- Payment Processing: The system should integrate with Paytm for secure online payments.
- Booking Management: Users should be able to view, modify, or cancel their bookings.
- Admin Features: Admins should be able to view all bookings, accept or cancel reservations, and update room details.

3.2 Non-Functional Requirements

- Performance: The system should load pages within 3 seconds under normal conditions.
- Usability: The interface should be intuitive and accessible for users with varying levels of technical expertise.
- Security: The system must use HTTPS for all communications and secure sensitive data.
- Reliability: The system should have an uptime of at least 99.5%.
- Scalability: The system should be able to handle up to 1000 concurrent users.

3.3 Interface Requirements

- User Interface: The system should have a responsive design, compatible with desktops, tablets, and smartphones.
- Admin Interface: Admins should have access to a dashboard with an overview of bookings, room status, and user inquiries.

3.4 System Features

- Real-time availability of room status
- Automatic email notifications for bookings and cancellations
- Detailed room descriptions and facility information

4. External Interface Requirements

4.1 User Interfaces

- Web Interface: Interactive GUI for users and admins, featuring forms, search filters, and data displays.
- Mobile Interface: Optimized mobile view with touch-friendly controls**.**

4.2 Hardware Interfaces

- Server: The application will be hosted on a server with at least 16 GB RAM, 4 CPUs, and SSD storage.
- User Devices: The system will be accessible via any device with internet connectivity.

4.3 Software Interfaces

- Database: MySQL for data storage and retrieval.
- Payment Gateway: Integration with Paytm API for processing payments.
- Email Service: Integration with SMTP servers for sending emails.

4.4 Communication Interfaces

- HTTP/HTTPS: All client-server communications will occur over these protocols.

5. System Architecture

- Presentation Layer: User interfaces, including web and mobile views.
- Business Logic Layer: Handles booking, payment processing, and management operations.
- Data Access Layer: Manages database interactions.
- External Interfaces: Integrates with Paytm and email services.

6. System Design

6.1 Database Design

- User Table: Stores user credentials and profiles.
- Booking Table: Stores booking details, including dates, room type, and payment status.
- Room Table: Stores room details, including availability, pricing, and features.

6.2 Security Requirements

- **Authentication**: User credentials must be encrypted and securely stored.
- **Authorization**: Different access levels for users and admins.
- **Data Protection**: All sensitive data must be encrypted in transit and at rest.

6.3 Performance Requirements

- The system must support fast retrieval and updating of room and booking data.
- The response time for any user action should not exceed 3 seconds.

7. Validation and Verification

7.1 Testing Requirements

- Unit Testing: Ensure individual components function correctly.
- Integration Testing: Verify that all components work together as expected.
- System Testing: Conduct end-to-end testing of the entire system.
- User Acceptance Testing (UAT): Validate the system with real users to ensure it meets requirements.

7.2 Acceptance Criteria

- The system must pass all defined test cases.
- Users and admins should be able to perform all listed functions without errors.
- The system must handle concurrent users without performance degradation.