**Project Synopsis on**

**RealTime Ticket Booking System with Payment Integration**

**A Project Report**

***Submitted in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

***by***

Punita Gaba | Prashant Kaushik | Tushar jakhar

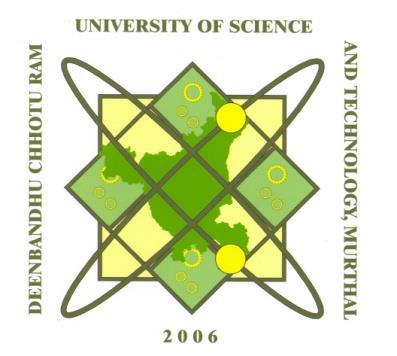
21001001087 | 21001001084 | 21001001126

**Under The Supervision of**

**Mr. Ajmer Singh**

**(Associate Professor, CSED)**

Session JulyDec 2024

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Sonepat131039**

**Index**

| **S. No.** | **Content** | **Page No.** | **Signature** | |
| --- | --- | --- | --- | --- |
| 1. | Introduction | 1-3 |  | |
| 2. | Modules in the Project | 4-6 |  | |
| 3. | Data Flow Diagrams | 7-10 |  | |
| 4. | | Software and Hardware Requirements | 11-12 |  |
| 5. | Development Plan | 13-14 |  | |
| 6. | References | 15 |  | |

**Introduction**

#### ****Significance of the Project****

In today’s digital era, the demand for convenience, efficiency, and realtime access to services is evergrowing. The RealTime Ticket Booking System with Payment Integration aims to provide users with a seamless, unified platform for booking tickets across various modes of travel, including trains, flights, and cabs, as well as reserving hotel accommodations. This project not only simplifies the booking process but also centralizes it, reducing the need for multiple apps or websites and enhancing the user experience with a secure, featurerich, and easily navigable system. By integrating a secure payment gateway, the project aims to create a reliable, all in one solution for booking and payment.

**Background**

The travel and hospitality industry has seen significant advancements with platforms like MakeMyTrip, Expedia, and Booking.com, which provide users with the convenience of booking multiple services—trains, flights, cabs, and hotels—within a single platform. These platforms have streamlined the user experience by consolidating travel services into one interface, allowing users to plan and book their entire trip in one place. Inspired by these existing platforms, this project aims to build a realtime ticket booking system that not only replicates the convenience of platforms like MakeMyTrip but also addresses challenges such as data consistency, concurrency, and scalability. The platform will integrate thirdparty APIs for dynamic services like train schedules and flight availability, ensuring that users always have access to the most uptodate information. By using trusted APIs for these timesensitive services, the platform will eliminate the need for manual data updates and ensure realtime accuracy of availability and pricing.

For services like cabs and hotels, the platform will directly utilize a MongoDB database to store information, ensuring that this more static data (e.g., available hotels, cab booking details, etc.) can be managed and updated more easily through an admin panel. The flexibility of MongoDB allows for quick updates to hotel and cab availability, and the admin panel will streamline data management, ensuring that the platform's backend is always uptodate.

An important aspect of the project is handling concurrency in realtime bookings, particularly when multiple users attempt to book the same seat, room, or service. The platform will address these issues through concurrency control mechanisms such as optimistic locking and transaction management in MongoDB. This will ensure that doublebooking is prevented, and the system remains consistent even under high user traffic.

In addition to providing a reliable, multimodal booking experience, the platform will incorporate a secure payment gateway for seamless transactions, supporting various payment methods and ensuring that user financial data is protected. An admin panel will allow administrators to manage and update both dynamic (APIbased) and static (MongoDBbased) data, making it easy to add, modify, or remove available services as needed.

By combining API integration for dynamic data and MongoDB for static data management, this project offers a scalable and userfriendly solution for booking trains, flights, cabs, and hotels in realtime. The platform will not only provide a seamless experience for users but also ensure data consistency, reduce concurrencyrelated issues, and create a reliable system for handling bookings and payments.

**How My Project Differs from Existing Platforms**

This project builds on the foundations of existing platforms but takes significant strides in improving data accuracy, concurrency handling, and both user and admin experience. By combining real-time API data integration with MongoDB for stable service management, advanced concurrency techniques, and a secure, easy-to-use admin panel, this system provides users with a seamless and reliable booking experience. Additionally, the user interface is designed to be straightforward and easy to navigate, ensuring that users of all experience levels can enjoy a smooth and accessible booking process. Ultimately, this solution not only matches the convenience of well-known booking platforms but also enhances data integrity, management ease, and user satisfaction.

**Key Proposed Features**

1. MultiService Booking

The platform will allow users to book a variety of services in one place, including train tickets, flight bookings, hotel reservations, and cab bookings. This integrated approach provides a seamless user experience for planning and executing all travelrelated bookings from a single platform.

1. Real-Time Data Integration

For train schedules and flight availability, the platform will utilize third-party APIs to fetch real-time data on available routes, timings, and pricing. To ensure quick response times and reduce the load on these external APIs, a caching mechanism will be implemented.

3. Custom Database for Cabs and Hotels

While dynamic services like trains and flights will use APIs, cabs and hotels will be managed directly through a custom MongoDB database. This will enable the platform to store and manage details such as available cabs, hotels, pricing, and availability, which can be easily updated via the admin panel.

4. Concurrency Control for RealTime Bookings

The system will implement advanced concurrency control mechanisms such as optimistic locking and transaction management to handle simultaneous booking requests and prevent doublebooking. This ensures that bookings are consistent and reliable even when multiple users are trying to book the same service in realtime.

5. User Authentication and Secure Login

The platform will include a secure user authentication system that allows users to register, log in, and manage their profiles. This will ensure that each user’s data, booking history, and payment details are securely stored and easily accessible across sessions.

6. Payment Gateway Integration

The platform will integrate a secure payment gateway to facilitate safe and smooth financial transactions. Users will be able to make payments for their bookings using various payment methods, including credit/debit cards, wallets, and other digital payment systems.

7. Admin Panel for Service Management

An intuitive admin panel will allow administrators to manage the backend of the platform, including the addition of new cabs, hotels, and services. Admins can also manage user accounts, booking histories, and other platform data, ensuring smooth operations and easy updates to the available services.

8. Booking History and Notifications

Users will have access to their booking history, including previous and upcoming reservations. The platform will also send realtime notifications to users regarding their booking status, payment confirmation, and upcoming travel details.

9. Search and Filter Options

Users will be able to search for available services based on various criteria such as destination, date, price, service type, and availability. Advanced filter options will enable users to easily find the best options suited to their preferences.

10. Responsive UI/UX Design

The platform will be designed with a responsive user interface that works seamlessly across various devices, including desktops, tablets, and smartphones. The design will focus on providing an easytouse, intuitive experience for users to quickly navigate and make bookings.

11. Ratings and Reviews System

Users will be able to rate and review the services they use, such as hotels, cabs, and trains, helping future users make informed decisions. Reviews will be a key feature for improving the credibility of services listed on the platform.

12. Admin Analytics and Reporting

The platform will include an analytics dashboard for admins to track user behavior, popular services, revenue generation, and booking trends. This data will help in optimizing platform operations and offering tailored services to users.

**Modules in the Project**

1. User Authentication Module

Description: This module will handle user registration, login, and authentication. It will use JWT (JSON Web Tokens) or sessionbased authentication to securely manage user login sessions.

Key Features:

User Registration (signup with email, phone, and password)

Login and Logout functionality

Password recovery and reset functionality

Profile management (view and edit user details)

2. Booking Module

Description: This module is the core functionality of the system, allowing users to book train tickets, flight tickets, hotels, and cabs.

Key Features:

Search functionality for available services (train, flights, cabs, and hotels)

Display of service details (timings, prices, availability)

Booking of selected services (train, flight, hotel, cab)

Booking confirmation and summary

Ability to cancel or modify bookings

Concurrency handling to prevent doublebooking (locking mechanism)

3. Admin Panel Module

Description: This module provides an administrative interface for managing services (train schedules, flights, hotels, cabs) on the platform.

Key Features:

CRUD (Create, Read, Update, Delete) operations for trains, flights, cabs, and hotel data

Ability to update pricing, availability, and service details

Manage user accounts and bookings

Data analytics dashboard for monitoring user activity and service performance

4. Payment Integration Module

Description: This module will integrate a payment gateway to allow users to make secure payments for their bookings.

Key Features:

Integration with thirdparty payment systems (e.g., Stripe, PayPal, Razorpay)

Handling of different payment methods (credit/debit cards, wallets)

Payment confirmation and receipt generation

Secure transaction processing (SSL encryption)

5. Search and Filter Module

Description: This module allows users to search for available services (trains, flights, hotels, cabs) and apply filters based on different criteria.

Key Features:

Search by destination, date, service type, price, etc.

Filter options to narrow down search results (e.g., nonstop flights, available cabs, hotel ratings)

Sorting of search results by price, availability, or user ratings

6. Notification Module

Description: This module will handle notifications to inform users about booking status, payment confirmations, and other updates.

Key Features:

Push notifications for booking status updates (e.g., booking confirmed, payment successful)

Email notifications for important booking details (e.g., reminders, cancellations)

Realtime alerts for service availability or special offers

7. Ratings and Reviews Module

Description: This module allows users to rate and review the services they book (trains, flights, hotels, cabs).

Key Features:

Users can rate services on a scale (e.g., 1 to 5 stars)

Ability to write reviews for each service (train, flight, hotel, cab)

Display of ratings and reviews for each service to help other users make informed decisions

8. Booking History and User Profile Module

Description: This module will maintain a history of all the bookings made by the user, as well as manage user profile information.

Key Features:

View and manage booking history (past and upcoming bookings)

User profile (personal details, payment methods, preferences)

Option to rebook or cancel past bookings

Integration with the User Authentication Module for profile updates

9. Data Analytics and Reporting Module (Admin)

Description: This module is designed for administrators to monitor platform usage, service performance, and financial reports.

Key Features:

Dashboard for tracking active users, booking trends, and revenue

Reports on popular routes, services, and user preferences

Monthly/quarterly financial reports

User behavior insights (e.g., service searches, booking trends)

10. Concurrency Control Module

Description: This module will ensure that simultaneous booking requests for the same resource (e.g., seat, hotel room) are handled properly to prevent conflicts and doublebooking.

Key Features:

Locking mechanisms to prevent doublebooking (optimistic or pessimistic locking)

Transaction management to ensure data consistency during concurrent operations

Conflict resolution (if two users attempt to book the same seat or service)

11. Admin Authentication and Security Module

Description: This module will manage admin access, ensuring only authorized users can access the admin panel.

Key Features:

Admin user roles and permissions

Secure login for admins (using multifactor authentication)

Access control for different sections of the admin panel

**Future Enhancements and Improvements**

1. Discounts and Offers Module

Description: This module will manage discounts, offers, and promo codes available for users, either based on special occasions, membership, or partnerships with service providers.

Features:

Userspecific discounts based on booking history or loyalty.

Promo codes for special discounts on selected services.

Display and management of active offers for services like flights, hotels, or trains.

2. GeoLocation and Map Integration Module

Description: This module can add geolocation functionality for cab bookings and for displaying the proximity of available services.

Features:

Realtime location tracking for cabs and hotels.

Map view for locating nearby services (e.g., hotels, train stations, cab pickups).

Route suggestions for better travel planning.

3. MultiLanguage and MultiCurrency Support

Description: For users from different regions, you could add support for multiple languages and currencies, improving accessibility.

Features:

Language selection for global users (English, Spanish, French, etc.).

Currency conversion and display of prices in local currency.

Support for international payment methods (PayPal, international debit/credit cards).

4. Search History and Suggestions Module

Description: This module will store search history and suggest personalized results based on user preferences and past searches.

Features:

Suggested services based on user’s previous search and bookings.

Recently searched services (trains, flights, hotels, cabs).

Autocomplete search bar for quicker bookings.

5. Customer Support Chatbot or Live Chat

Description: An AIpowered or live chat support system to help users with booking issues, inquiries, or cancellations.

Features:

Automated responses for common queries (e.g., booking status, payment issues).

Option for users to connect with a live support agent.

24/7 support for urgent issues or inquiries.

**Data Flow Diagram**

Context Diagram: This will show the system as a single process and the external entities interacting with it.

Decomposition Diagram: This breaks down the system into its primary components and shows how data flows between them.

**Context Diagram**

In this diagram, we have shown the RealTime Ticket Booking System as a single process, and the interactions with external entities like the User, Admin, and Payment Gateway.

Entities:

User: A person who books tickets (train, flight, cab, hotel) and makes payments.

Admin: The administrator who manages services, bookings, and user data.

Payment Gateway: The external payment system (e.g., Stripe, PayPal) for processing payments.

Diagram Description:

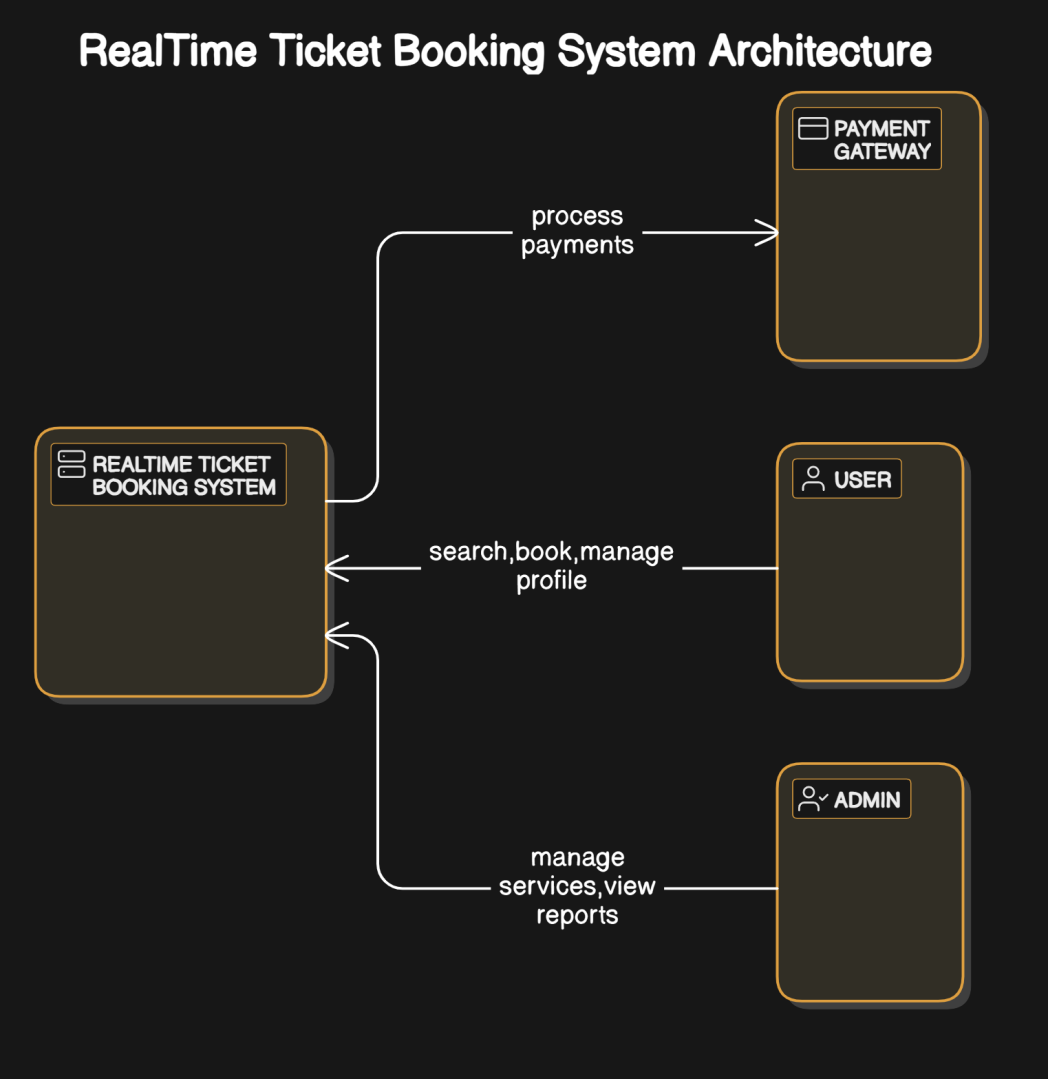
System: RealTime Ticket Booking System (depicted as a single process)

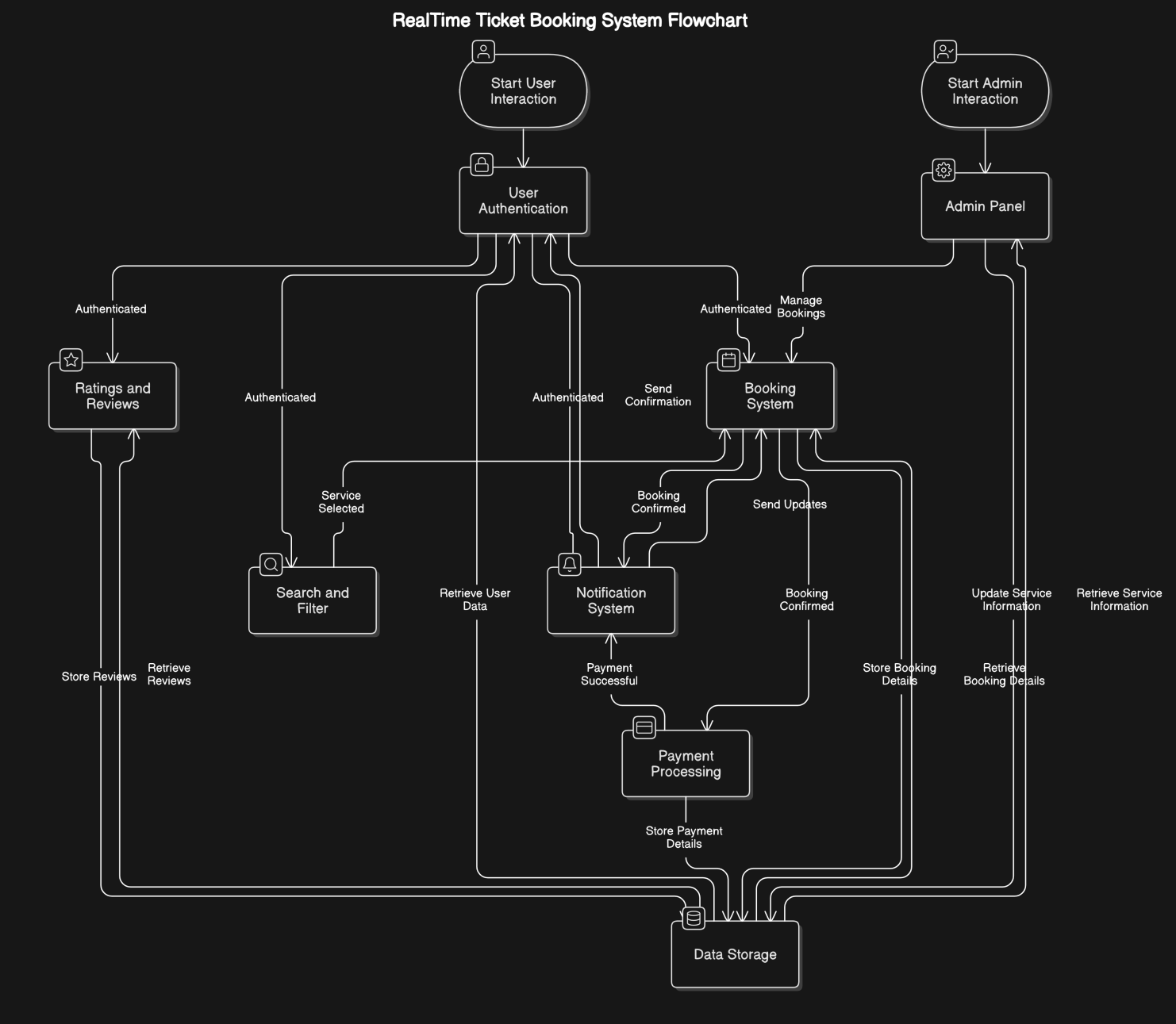
Interactions:

Users interact with the system to search for and book services, make payments, and manage their profile.

Admin interacts with the system to manage available services, view reports, and monitor bookings.

Payment Gateway handles the payment processing for user bookings.





**Decomposition Diagram**

This level decomposes the RealTime Ticket Booking System into its key modules to show data flows between them.

Key Modules:

1. User Authentication: Handles user login, registration, and authentication.

2. Booking System: Handles ticket booking (train, flight, cab, hotel) logic.

3. Admin Panel: Manages services, booking management, and adminspecific functionalities.

4. Payment Processing: Handles the payment system integration for bookings.

5. Search and Filter: Allows users to search available services based on various parameters.

6. Notification System: Manages notifications related to booking status, confirmations, and updates.

7. Ratings and Reviews: Allows users to rate and review services after booking.

8. Data Storage (Database): Stores user data, booking details, and service information (MongoDB).

Data Flows:

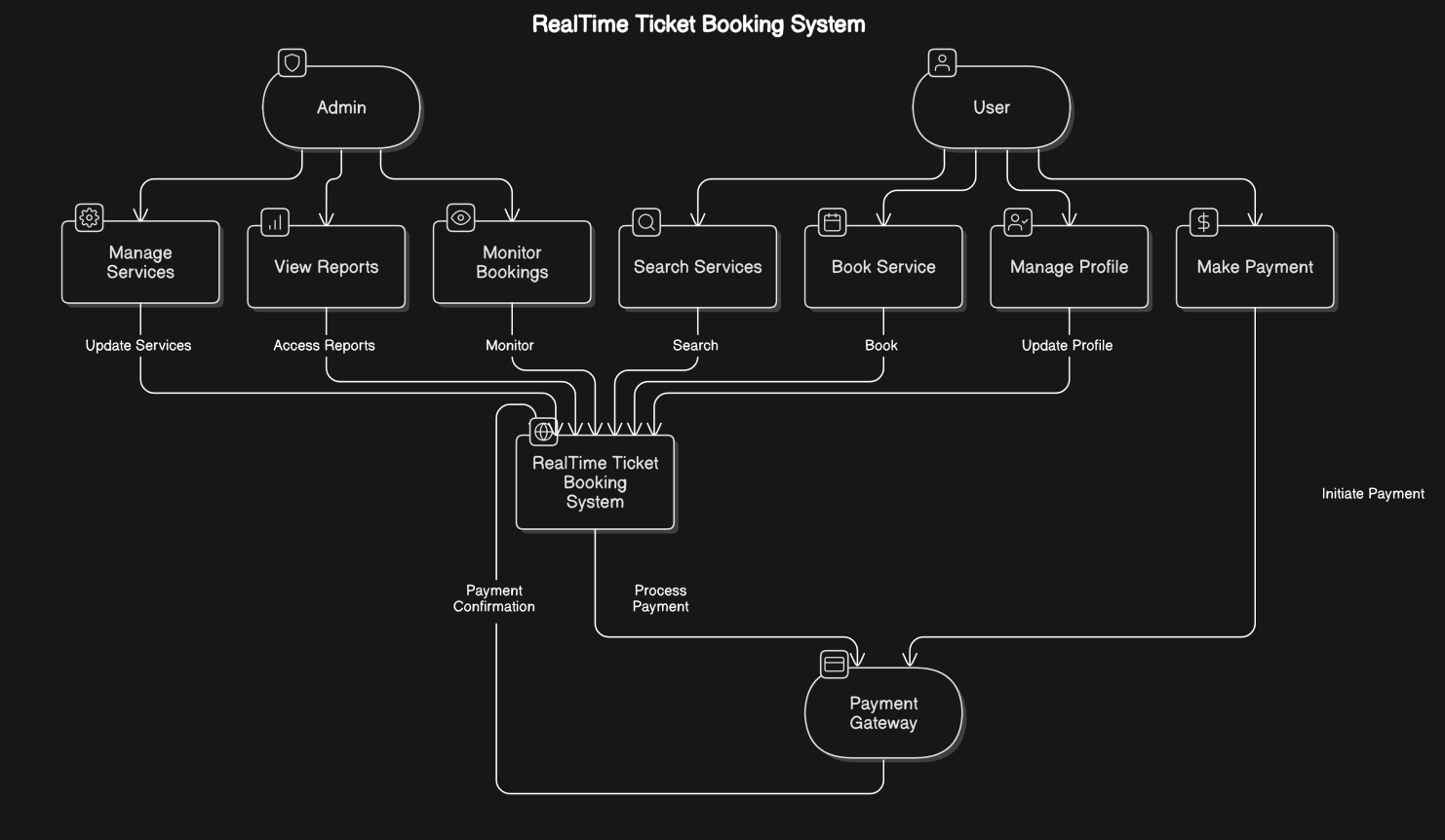
User Authentication sends user credentials to the User Profile Management module to validate login.

Booking System interacts with the Search and Filter and Database to fetch and book services.

Payment Processing handles payment requests from the Booking System and sends transaction status back.

Admin Panel updates and manages service information in the Database.

Notification System sends booking updates to users after payment and booking.



Data Flow Overview:

1. User Authentication:

The user inputs credentials (email/password) and the system verifies them. The User Authentication module communicates with the Database to validate and authenticate users.

2. Booking System:

The user searches for available services (train, flights, hotels, cabs) via the Search & Filter module. It fetches the data from the Database.

After booking a service, the Booking System generates a booking and interacts with the Payment Processing system for the payment transaction.

3. Payment Processing:

The Payment Processing module sends payment details to the Payment Gateway. Once the payment is processed, the status is returned to the Booking System, which updates the booking status.

4. Admin Panel:

The Admin manages the system's backend through the Admin Panel module. They add, update, or delete services (train schedules, flight details, hotel rooms, cabs) in the Database.

5. Notification System:

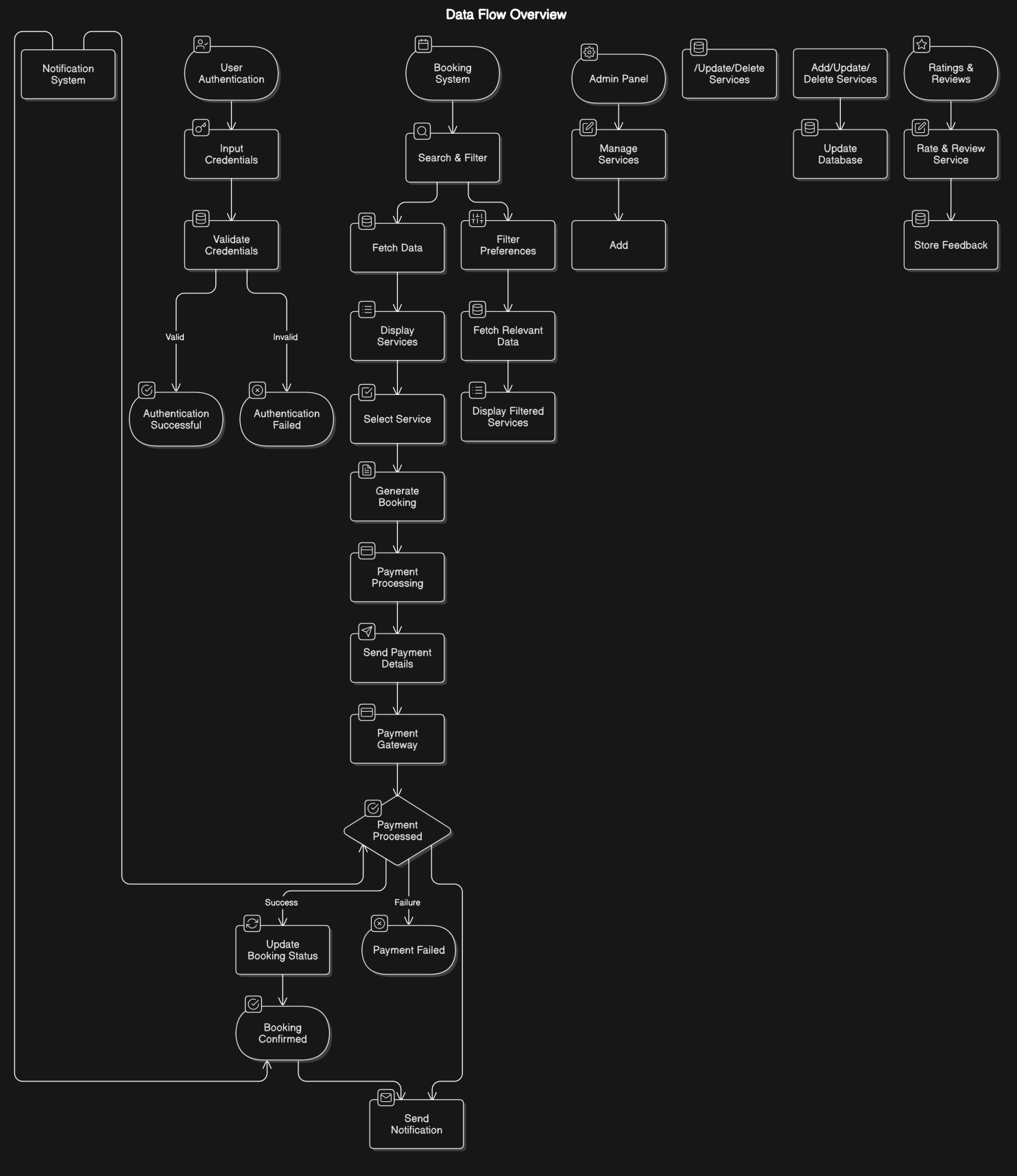
Once a booking is confirmed or a payment is processed, the Notification System sends notifications (via email, SMS) to the user.

6. Search & Filter:

The Search & Filter module enables the user to filter available services based on their preferences (destination, price, date). It fetches relevant data from the Database.

7. Ratings & Reviews:

Users can rate and review services after completing a booking, storing this feedback in the Database.



**Software And Hardware Requirements**

**Software Requirements**

1. Frontend Development:

Node.js: JavaScript runtime used for building the frontend and backend (using Express.js).

React: JavaScript library for building the user interface.

Redux (optional): State management tool for handling application state (useful for larger apps).

Axios: HTTP client to interact with the backend API (for making API requests).

React Router: Library for navigating between different pages in a singlepage application.

Bootstrap or MaterialUI: CSS frameworks for responsive and modern user interface design.

2. Backend Development:

Node.js: JavaScript runtime environment for developing the serverside of the application.

Express.js: Minimal and flexible Node.js web application framework for building APIs.

MongoDB: NoSQL database for storing user data, service details, and bookings. It is highly scalable and flexible for this type of application.

Mongoose: Object Data Modeling (ODM) library for MongoDB and Node.js, used to interact with

MongoDB.

JWT (JSON Web Token): Used for user authentication and secure tokenbased communication.

Bcrypt.js: Library for hashing passwords for secure authentication.

Nodemailer: Used for sending emails (e.g., booking confirmations, user registration, etc.).

Payment Gateway API (e.g., Stripe/PayPal): For payment processing and integration into the system (if you are using free alternatives).

Socket.io (Optional): For realtime updates (e.g., booking status, availability of seats) and notifications.

3. Development Environment:

Visual Studio Code: Lightweight code editor with support for JavaScript, Node.js, and React development.

Postman: For testing API endpoints during development.

Git: Version control system to track changes in the codebase.

GitHub/GitLab: For hosting the code repository and collaborating with other developers (if applicable).

npm (Node Package Manager): For managing project dependencies.

4. Operating System:

Windows / MacOS / Linux: Any of these operating systems can be used for development, depending on the user's preference.

5. Browser:

Google Chrome or Mozilla Firefox: Recommended for testing and debugging the web application.

**Hardware Requirements**

1. Development Machine:

Processor: 2 GHz or faster (Intel i5 or equivalent recommended).

RAM: Minimum 8 GB (16 GB recommended for smoother development experience).

Storage: At least 100 GB of free disk space (SSD recommended for faster read/write speeds).

Display: 15inch screen with at least 1080p resolution (higher resolution is preferred for multitasking).

Network: Stable internet connection for accessing online resources and APIs.

2. Server (for Hosting the Application, if deploying):

Processor: Dualcore or better (e.g., Intel Xeon or equivalent).

RAM: Minimum 2 GB (4 GB or more recommended for handling larger traffic).

Storage: At least 20 GB of disk space (SSD preferred for faster server operations).

Network: At least 1 Mbps download/upload speed (depending on the expected user load, higher speeds might be needed).

3. Additional Tools (Optional but Helpful):

Database Hosting (if cloudbased, like MongoDB Atlas for database hosting):

MongoDB Atlas offers a freetier for smallscale applications. No paid services are required unless traffic increases.

Cloud Hosting Platforms (e.g., Heroku, Vercel, Netlify for frontend, or DigitalOcean for fullstackdeployment) offer free tiers that should suffice for initial testing and development.

**Development Plan**

Development Plan for RealTime Ticket Booking System with Payment Integration. Since our timeline is from 1st November to 29th November, we break down the tasks into key milestones for efficient project development.

Project Development Plan (1st Nov - 29th Nov)

| **S. No.** | **Task** | **Duration** | **Start Date** | **End Date** |
| --- | --- | --- | --- | --- |
| 1 | **Project Planning and Requirement Analysis** | 2 days | 1st Nov | 2nd Nov |
| 2 | **Database Design & Structure Setup** | 2 days | 3rd Nov | 4th Nov |
| 3 | **Setting up Development Environment** | 1 day | 5th Nov | 5th Nov |
| 4 | **Frontend (UI) Design and Structure** | 4 days | 6th Nov | 9th Nov |
| 5 | **Backend Setup (API, Server, Authentication)** | 5 days | 10th Nov | 14th Nov |
| 6 | **Admin Panel Development** | 3 days | 15th Nov | 17th Nov |
| 7 | **Integrating Train, Flight, Hotel, and Cab Data** | 4 days | 18th Nov | 21st Nov |
| 8 | **Payment Integration Setup** | 3 days | 22nd Nov | 24th Nov |
| 9 | **Concurrency Management (Booking Locking Mechanism)** | 3 days | 25th Nov | 27th Nov |
| 10 | **Testing & Debugging (Frontend & Backend)** | 2 days | 28th Nov | 29th Nov |

Detailed Breakdown of Tasks:

1. Project Planning and Requirement Analysis (1st Nov 2nd Nov)

Finalize project features.

Analyze the architecture and technical stack (MERN stack).

Plan the project layout, database structure, and API endpoints.

2. Database Design & Structure Setup (3rd Nov 4th Nov)

Set up the MongoDB database.

Design collections for users, bookings, trains, flights, hotels, cabs, and payment details.

Set up data models and relations.

3. Setting up Development Environment (5th Nov)

Install necessary tools (Node.js, MongoDB, React, Express).

Set up the project repository (e.g., on GitHub).

Configure local development environment for both backend and frontend.

4. Frontend (UI) Design and Structure (6th Nov 9th Nov)

Design the UI for the homepage, search functionality, results display, and booking form.

Implement components for user signup, login, booking details, and payment.

Implement routing using React Router.

5. Backend Setup (API, Server, Authentication) (10th Nov 14th Nov)

Set up Express server and MongoDB integration.

Implement user authentication (signup, login, JWT tokenbased system).

Create API routes for user data, ticket search, and booking.

6. Admin Panel Development (15th Nov 17th Nov)

Develop an admin panel to manage train, flight, hotel, and cab data.

Implement functionalities to add, edit, and delete services.

Create admin authentication and authorization.

7. Integrating Train, Flight, Hotel, and Cab Data (18th Nov 21st Nov)

Add the necessary data for trains, flights, cabs, and hotels.

Integrate custom MongoDB data or external APIs (as required).

Implement the search, availability checks, and booking features.

8. Payment Integration Setup (22nd Nov 24th Nov)

Set up payment gateway (Stripe/PayPal) for payment processing.

Integrate payment features for booking confirmation.

Implement handling of successful and failed transactions.

9. Concurrency Management (Booking Locking Mechanism) (25th Nov 27th Nov)

Implement concurrency handling to prevent multiple users from booking the same seat (use locking mechanisms, such as optimistic/pessimistic locking).

Test and ensure data consistency during concurrent bookings.

10. Testing & Debugging (Frontend & Backend) (28th Nov 29th Nov)

Conduct thorough testing of the booking process (train, flights, cabs, hotels).

Test payment gateway functionality.

Debug issues and optimize code for better performance.

Write unit and integration tests to ensure functionality.

**Referances**

1. MakeMyTrip Official Website

MakeMyTrip. "MakeMyTrip: Travel & Tourism." Available at: (https://www.makemytrip.com/)

This platform served as inspiration for the design and functionality of the ticket booking system, providing insight into the realtime booking process, payment gateway integration, and multiservice options like flight, hotel, and cab bookings.

2. MongoDB Documentation

MongoDB, Inc. "MongoDB Documentation." MongoDB. Available at: (https://www.mongodb.com/docs)

Official MongoDB documentation to support the database design and data management for the ticketing system.

3. React Documentation

React Team. "React – A JavaScript Library for Building User Interfaces." React. Available at: [https://reactjs.org/](https://reactjs.org/)

The official React documentation for creating a dynamic and interactive user interface.

4. Express.js Documentation

Express Team. "Express Fast, Unopinionated, Minimalist Web Framework for Node.js." Express. Available at:(https://expressjs.com/)

The official Express documentation for building the backend and API logic for the booking system.

5. Stripe Payment Integration

Stripe, Inc. "Stripe API Reference." Stripe. Available at: (https://stripe.com/docs)

API documentation for integrating Stripe for secure payment processing within the application.

6. JWT Authentication

Auth0. "JWT Authentication Tutorial." Auth0. Available at: (https://auth0.com/docs/quickstart/backend/nodejs/01authorization)

A tutorial to help integrate JWT authentication for secure user login and authorization.

7. Concurrency Control in Distributed Systems

Silberschatz, Abraham, Korth, Henry F., and Sudarshan, S. Database System Concepts. 7th ed., McGrawHill, 2019.

A textbook that covers key concepts on concurrency control, crucial for preventing multiple users from booking the same seat at the same time.

8. Mongoose Documentation

Mongoose Team. "Mongoose.js Documentation." Mongoose. Available at: (https://mongoosejs.com/)

The Mongoose documentation used for creating the data models that interface with MongoDB.

9. Booking System Scalability and Performance

Gupta, Rajiv. "Scalable Booking Systems for High Traffic." International Journal of Software Architecture. 2021, vol. 29, no. 5, pp. 110123.

An academic reference discussing the challenges of scaling booking systems for high traffic volumes and the strategies to optimize performance.

10. Flight and Hotel Booking System APIs

Amadeus for Developers. "Amadeus API Documentation." Amadeus for Developers. Available at: (https://developers.amadeus.com/)

Documentation on Amadeus API, which offers realtime access to flight, hotel, and travel booking information, used as a reference for building your own realtime booking system.