**Project Design Phase**

**Solution Architecture**

| Date | 15 April 2025 |
| --- | --- |
| Team ID | Team-145269 |
| Project Name | Flight Finder |
| Maximum Marks | 4 Marks |

## Solution Architecture Overview: Flight Booking System

Solution Architecture is a strategic process that ensures our technology choices effectively support the core functionalities and user needs of the Flight Booking System. This architecture aims for a scalable, efficient, and user-friendly platform for searching, booking, and managing flights.

**Purpose of Solution Architecture in Flight Booking System:**

* **Identify Best Tech Solution:** Leverage a robust and scalable architecture, potentially incorporating technologies like React for a dynamic frontend, a suitable backend framework (like Spring Boot for Java, Django for Python, or Node.js with Express), and a reliable database (like PostgreSQL or MySQL) to manage flight data, bookings, and user information.
* **Communicate with Stakeholders:** Clearly illustrate how users interact with the system (searching, selecting flights, booking), how flight and booking data is managed in the backend, and the role of APIs in connecting different components.
* **Define Features & Phases:** Structure the development process into manageable sprints, focusing on core features like flight search and display, user authentication, booking management, and payment gateway integration.
* **Deliver Specifications:** Provide technical documentation detailing API endpoints for flight data and booking operations, database schema, authentication and authorization mechanisms, and user interface/user experience (UI/UX) flows.

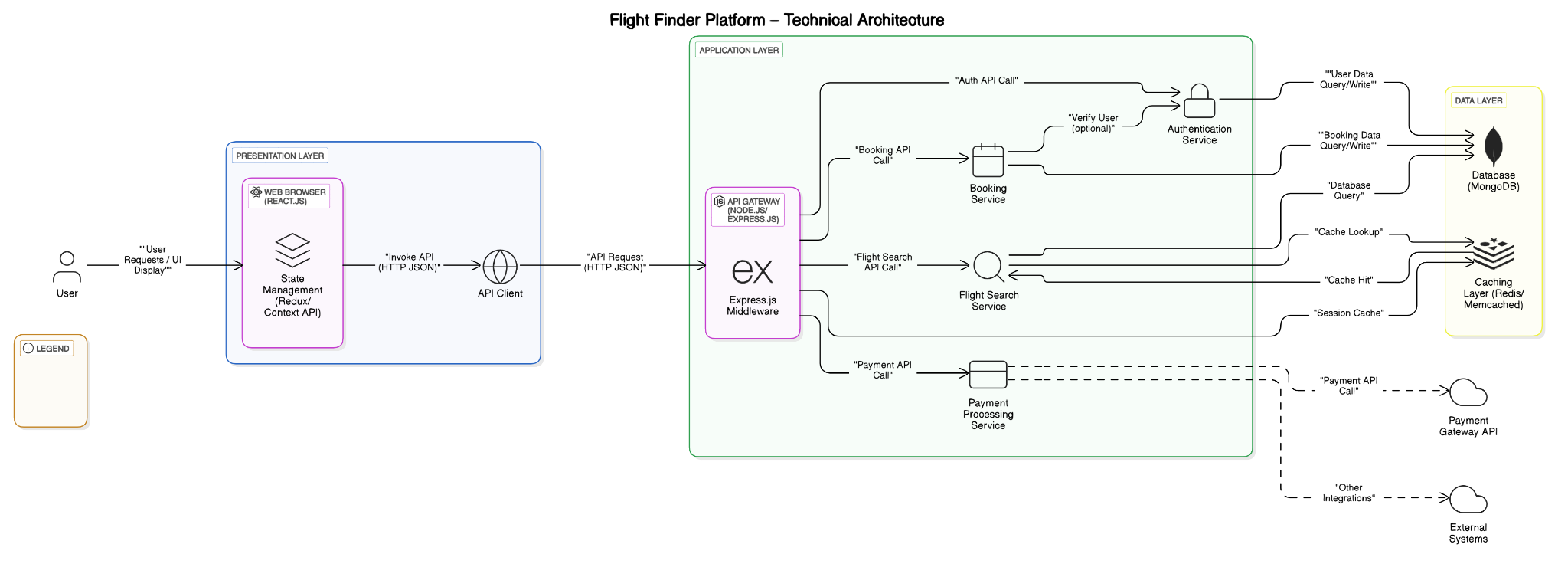
### **Key Components of the Solution Architecture**

| Component | Description |
| --- | --- |
| Frontend (React.js) | Delivers an interactive and responsive user interface for searching flights, viewing flight details, managing bookings, and handling user accounts. |
| Backend (e.g., Spring Boot/Django/Node.js + Express) | Handles API requests for flight data, user authentication, booking creation and management, payment processing integration, and communication with the database. |
| Database (e.g., PostgreSQL/MySQL) | Stores critical data including flight schedules, airline information, airport details, user profiles, booking records, and payment information securely and efficiently. |
| Flight Data Service | Responsible for fetching and providing real-time or near real-time flight information (availability, schedules, pricing) potentially from external APIs or a dedicated flight data management system. |
| Booking Management Service | Manages the entire booking lifecycle, including seat selection, passenger details, confirmation generation, and cancellation/modification processes. |
| User Authentication & Authorization | Securely manages user registration, login, and access control using industry-standard practices like OAuth 2.0 or JWT to protect user data and booking information. |
| Payment Gateway Integration | Facilitates secure online payments through integration with third-party payment processors (e.g., Stripe, PayPal) to handle transactions for flight bookings. |
| Search & Filtering Module | Enables users to efficiently search for flights based on various criteria such as origin, destination, date, time, airline, and price range. |
| Notifications Service | Provides real-time updates and notifications to users regarding booking confirmations, flight changes, or other relevant information (e.g., via email, SMS, or in-app notifications). |
| Deployment | Hosted on cloud platforms (e.g., AWS, Google Cloud, Azure) for scalability, reliability, and continuous availability. |

**Development Phases:**

* **Sprint 1: User Registration, Login, and Flight Search: Implement user registration and secure authentication (e.g., JWT). Develop the basic flight search functionality allowing users to find flights based on origin, destination, and dates.**
* **Sprint 2: Displaying Flight Details and Booking Initiation: Enhance the search results display with detailed flight information. Implement the initial steps of the booking process, including passenger details input and seat selection (if applicable).**
* **Sprint 3: Payment Gateway Integration and Booking Confirmation: Integrate with a payment gateway to process booking payments securely. Implement the booking confirmation process and generate booking details for the user.**
* **Sprint 4: Booking Management and Notifications: Develop features for users to view, modify, or cancel their bookings. Implement a notification system to keep users informed about their bookings.**
* **Sprint 5 (and beyond): Additional Features and Enhancements: Implement features like loyalty programs, baggage handling options, integration with other travel services (e.g., hotels, car rentals), advanced search filters, and performance optimization.**

**Example - Solution Architecture Diagram:**

****

*Figure 1: Architecture and data flow*