**Project Details**

Team Members

1. C070 - Ronit Shetty
2. C073 - Viha Shukla
3. C079 - Christopher Dsouza
4. C125 - Shrunjal Sanghvi

**Travel Itinerary Generator - Microservices Platform**

**Introduction:** The Travel Itinerary Generator is a microservices-based platform designed to help users seamlessly plan, book, and manage personalized travel itineraries. The system offers dynamic recommendations, real-time updates, and a structured approach to travel planning. Built using Spring Boot, the platform ensures modularity, scalability, and efficiency by breaking functionalities into independent microservices.

**Objectives:**

* To develop a scalable and modular travel itinerary management system.
* To provide personalized activity recommendations based on user preferences.
* To enable users to create, update, delete, and share travel itineraries.
* To facilitate real-time booking and itinerary updates
* To ensure secure authentication and seamless user experience.

**Scope of the Project:**

* The system will allow users to register, log in, and manage their profiles.
* Users can create itineraries, add destinations, and book activities.
* Activities will be suggested based on user preferences and location.
* The platform will handle booking management with real-time status updates.

**System Overview:** The platform consists of three core microservices: **User Service, Itinerary Service, and Activity Service**. These services communicate via REST APIs and are managed using an API Gateway integrated with Eureka Server for service discovery.

1. **User Service:** Manages user profiles, authentication, and travel preferences. Users can create accounts, update their details, and store travel preferences that influence itinerary recommendations.
2. **Itinerary Service:** Handles travel plans, allowing users to create, modify, delete, and share itineraries. It supports multiple destinations, trip durations, and status updates.
3. **Activity Service:** Provides a curated list of activities based on destination and user preferences. Users can browse, book, and manage activities within their itineraries.

**Technologies Used:**

* **Backend:** Spring Boot, Spring Cloud, Java
* **Database:** MySQL
* **API Gateway & Service Discovery:** Spring Cloud Gateway, Eureka Server
* **Security:** Spring Security, JWT Authentication
* **Communication:** REST APIs, Feign Clients
* **Infrastructure & Deployment:** Docker, Kubernetes, AWS

**System Workflow:**

1. Users register and log in via the **User Service**.
2. They create an itinerary using the **Itinerary Service**.
3. The **Activity Service** suggests activities based on preferences and location.
4. Users book activities, and bookings are managed under the **Itinerary Service**.
5. Updates and notifications are provided in real-time, ensuring a smooth user experience.

**ER diagram**

**User**

| **Column Name** | **Constraints** | **Description** |
| --- | --- | --- |
| **user\_id** | Primary Key | Unique User ID |
| name | NOT NULL | User’s full name |
| email | UNIQUE, NOT NULL | User’s email |
| password | NOT NULL | password |
| preferences | NULLABLE | Travel preferences |
| created\_at | DEFAULT CURRENT TIMESTAMP | Registration time |

### **Itinerary**

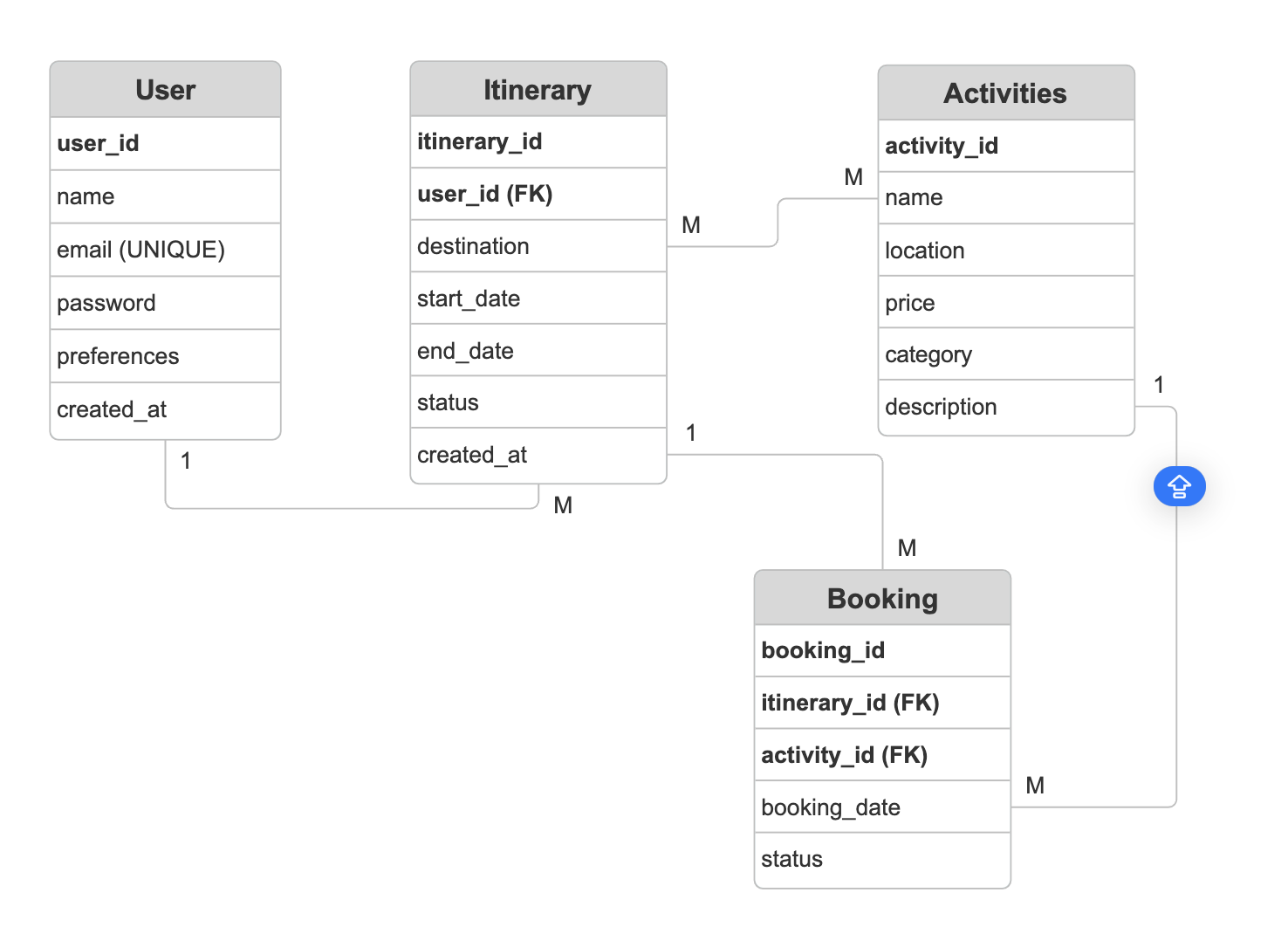
| **Column Name** | **Constraints** | **Description** |
| --- | --- | --- |
| **itinerary\_id** | Primary Key | Unique Itinerary ID |
| **user\_id** | Foreign Key (users) | Owner of the itinerary |
| destination | NOT NULL | Travel destination |
| start\_date | NOT NULL | Trip start date |
| end\_date | NOT NULL | Trip end date |
| status | DEFAULT 'Planned' | Planned, In Progress, Completed |
| created\_at | DEFAULT CURRENT\_TIMESTAMP | Itinerary creation time |

### **Activities**

| **Column Name** | **Constraints** | **Description** |
| --- | --- | --- |
| **activity\_id** | Primary Key | Unique Activity ID |
| name | NOT NULL | Activity Name |
| location | NOT NULL | Activity Location |
| price | NOT NULL DEFAULT 0.0 | Activity Cost |
| category | NOT NULL | Type (Adventure, Culture, etc.) |
| description | NULLABLE | Activity Details |

### **bookings (Stores user bookings for activities)**

| **Column Name** | **Constraints** | **Description** |
| --- | --- | --- |
| **booking\_id** | Primary Key | Unique Booking ID |
| **itinerary\_id** | Foreign Key (itineraries) | Reference to Itinerary |
| **activity\_id** | Foreign Key (activities) | Reference to Activity |
| booking\_date | DEFAULT CURRENT\_TIMESTAMP | When booking was made |
| status | DEFAULT 'Pending' | Pending, Confirmed, Canceled |



**Relationships**

* **User → Itinerary:** One-to-Many (A user can have multiple itineraries).
* **Itinerary → Booking:** One-to-Many (An itinerary can include multiple bookings).
* **Activities → Booking:** One-to-Many (A single activity can have multiple bookings).
* **Itinerary → Activities (Mapping Table):** Many-to-Many (An itinerary can include multiple activities, and an activity can be part of multiple itineraries)