Design Document

for

Nexa

Version 1.0

Prepared by

Group Members **Group Name: Arcade Nation**

Likith Sai Jonna	**220483**	jonnals22@iitk.ac.in
Rohit Verma	**220917**	rohitverma22@iitk.ac.in
Kollamoram Karthik	**220538**	kkarthik22@iitk.ac.in
K S U Rithwin	**220537**	ksupendra22@iitk.ac.in
Ritesh Hans	**220893**	riteshhans22@iitk.ac.in
Gottupulla Venkata Aman	**220413**	gvaman22@iitk.ac.in

Course: CS455
Institute: IIT Kanpur
Instructor: Prof. Priyanka Bagade
Date: September 4, 2025

Contents

1 Revisions **2**

2 Architecture Design **3**

3 Object Oriented Design **5**
3.1 Use Case Diagrams . . . 5
3.1.1 Use Case 1 - Registration & Profile Management . . . 5
3.1.2 Use Case 2 - Travel Search, Booking & Ticketing . . . 5
3.1.3 Use Case 3 - Booking Management and Travel Support . . . 6
3.1.4 Use Case 4 - Payment and Loyalty Management . . . 6
3.1.5 Use Case 5 - Admin Login & Security . . . 7
3.1.6 Use Case 6 - User & Provider Management . . . 7
3.1.7 Use Case 7 - Data & Reports . . . 8
3.1.8 Use Case 8 - Monitoring & Communication . . . 8
3.1.9 Use Case 9 - Registration & Profile Management . . . 9
3.1.10 Use Case 10 - Service Management . . . 9
3.1.11 Use Case 11 - Analytics & Reports . . . 10
3.1.12 Use Case 12 - Bookings & Financials . . . 10
3.2 Class Diagrams . . . 11
3.2.1 Authentication Schema . . . 12
3.2.2 User's Schema . . . 13
3.2.3 Services's Schema . . . 14
3.2.4 Booking's Schema . . . 15
3.2.5 Payment's Schema . . . 16
3.2.6 Notification Schema . . . 17
3.2.7 AI Content Schema . . . 18
3.2.8 Forum's Schema . . . 19
3.3 Sequence Diagrams . . . 20
3.3.1 Sequence 1 - Registration & Profile Management . . . 20
3.3.2 Sequence 2 - Travel Search, Booking & Ticketing . . . 20
3.3.3 Sequence 3 - Booking Management and Travel Support . . . 21
3.3.4 Sequence 4 - Payment and Loyalty Management . . . 21
3.3.5 Sequence 5 - Admin Login & Security . . . 22
3.3.6 Sequence 6 - User & Provider Management . . . 22
3.3.7 Sequence 7 - Data & Reports . . . 23
3.3.8 Sequence 8 - Monitoring & Communication . . . 23
3.3.9 Sequence 9 - Registration & Profile Management . . . 24
3.3.10 Sequence 10 - Service Management . . . 24
3.3.11 Sequence 11 - Analytics & Reports . . . 25
3.3.12 Sequence 12 - Bookings & Financials . . . 25
3.4 State Diagrams . . . 26
3.4.1 Overall state diagram . . . 26
3.4.2 Booking Sequence to Avoid Double Booking . . . 27

1. Revisions

Version	Primary Author(s)	Description of Version	Date Completed
v1.0	Likith Sai Jonna Rohit Verma Kollamoram Karthik K S U Rithwin Ritesh Hans Gottupulla Venkata Aman	First version of the Software Design Document	04/09/2025

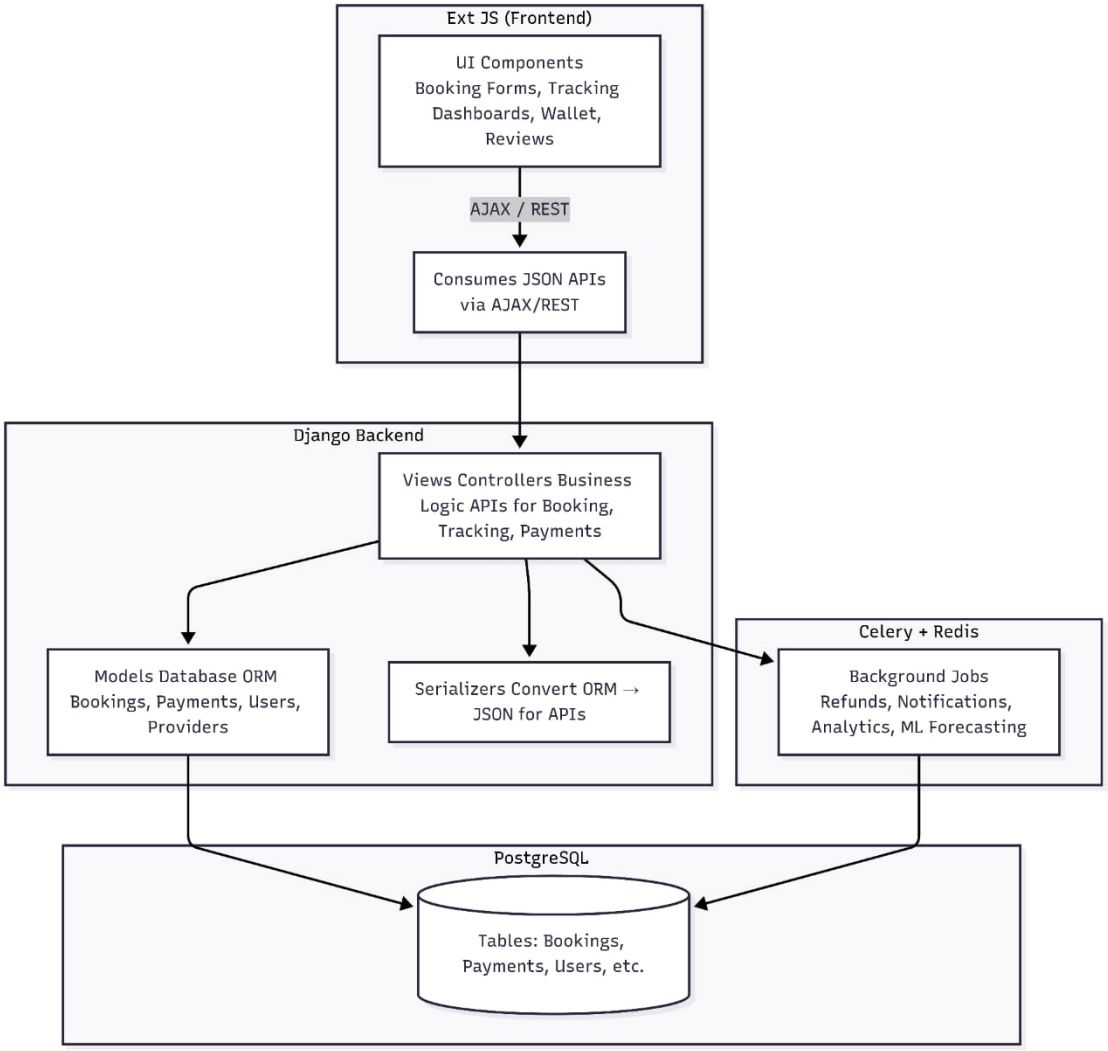
2. Architecture Design

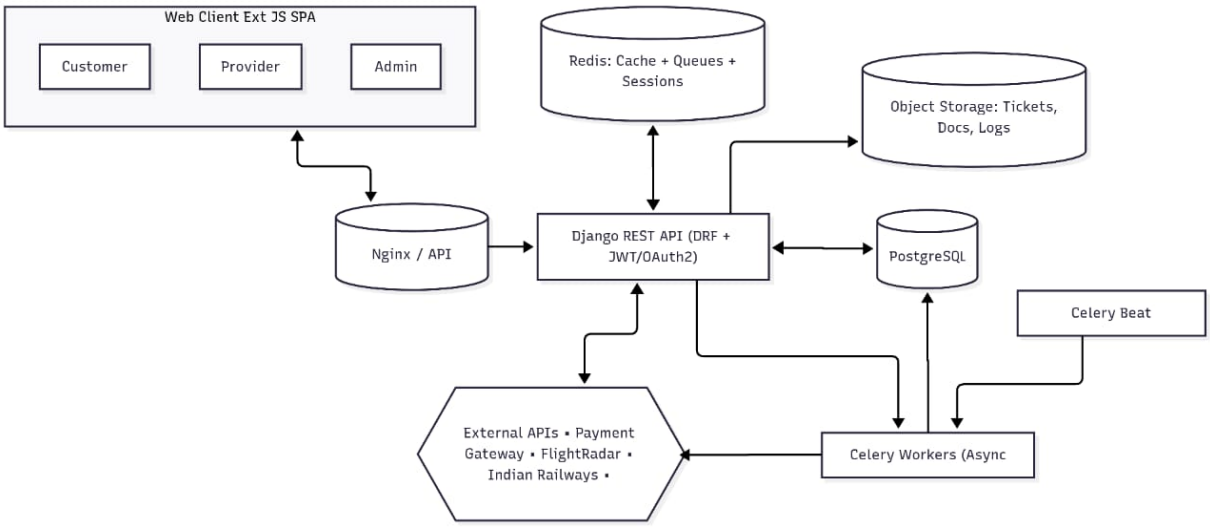
The overall architecture of our web application, Nexa, follows the Model-View-Controller (MVC) model. Given the complexity of the application, each functionality is subdivided into smaller tasks, which are achieved by leveraging the MVC architecture to ensure modularity, scalability, and maintainability.

The Model-View-Controller architecture is structured as follows:

- **Model**: The model serves as the data-centric component of the application, acting as the sole component directly interfacing with the database. Its primary role entails storing and retrieving data related to user profiles, bookings, service provider schedules, payments, and analytics. We plan to use PostgreSQL as our database, with Django as the backend framework to interact with the database efficiently.
- **View**: The view acts as the presentation component of the application, responsible for rendering data to the user and handling user input. For customers and service providers, the view will be a responsive web application built using Ext.js ensuring a seamless and adaptive user interface across desktops, tablets, and smartphones. Administrators will access a dedicated web interface with enhanced controls for managing providers, bookings, and analytics.
- **Controller**: The controller serves as the logic hub of the application, tasked with processing data and determining the application's flow. It will be developed using ext.js with django to handle business logic, API integrations, and request routing. The integration of external APIs, such as Google Maps API for route visualization and real-time tracking, and a dummy payment gateway (e.g., Stripe sandbox) for payment simulations, will be managed here. AI-based features, such as recommendation systems and chatbots, will also be integrated through the controller.

We have opted for the Model-View-Controller (MVC) model because it effectively accommodates the diverse data interaction needs of Nexa. This model supports various operations, such as searching for travel options, managing bookings, handling dynamic pricing, and providing administrative oversight. The MVC architecture also facilitates future enhancements, such as adding multilingual support or integrating additional transportation modes, making it the best fit for our scalable and user-centric platform.


Ext JS (Frontend)
UI Components Booking Forms, Tracking Dashboards, Wallet, Reviews
AJAX / REST
Consumes JSON APIs via AJAX/REST
Django Backend
Views Controllers Business Logic APIs for Booking, Tracking, Payments
Models Database ORM Bookings, Payments, Users, Providers
Serializers Convert ORM → JSON for APIs
Celery + Redis
Background Jobs Refunds, Notifications, Analytics, ML Forecasting
PostgreSQL
Tables: Bookings, Payments, Users, etc.

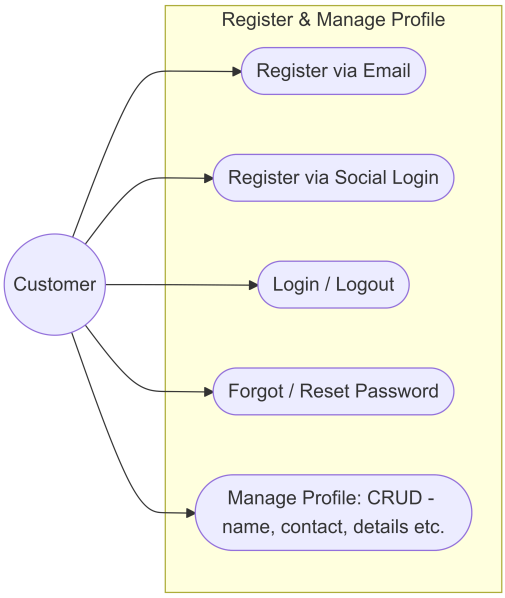

Web Client Ext JS SPA
Customer
Provider
Admin
Redis: Cache + Queues + Sessions
Object Storage: Tickets, Docs, Logs
Nginx / API
Django REST API (DRF + JWT/OAuth2)
PostgreSQL
Celery Beat
External APIs • Payment Gateway • FlightRadar • Indian Railways •
Celery Workers (Async

3. Object Oriented Design

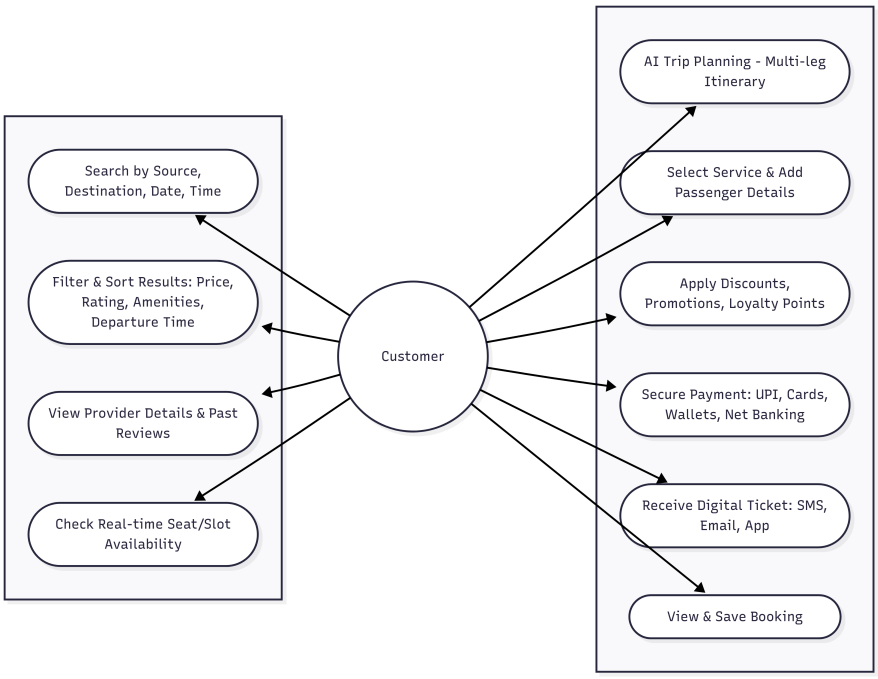
3.1 Use Case Diagrams

User Role : Customer

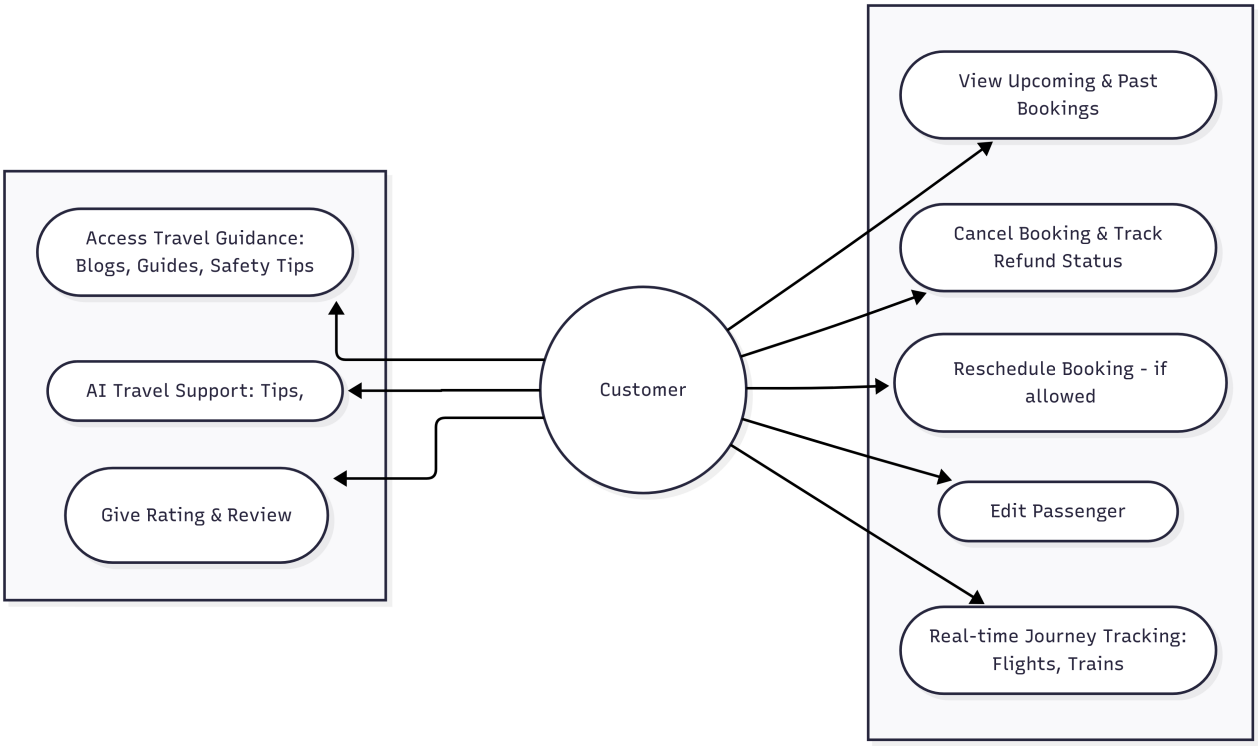
3.1.1 Use Case 1 - Registration & Profile Management



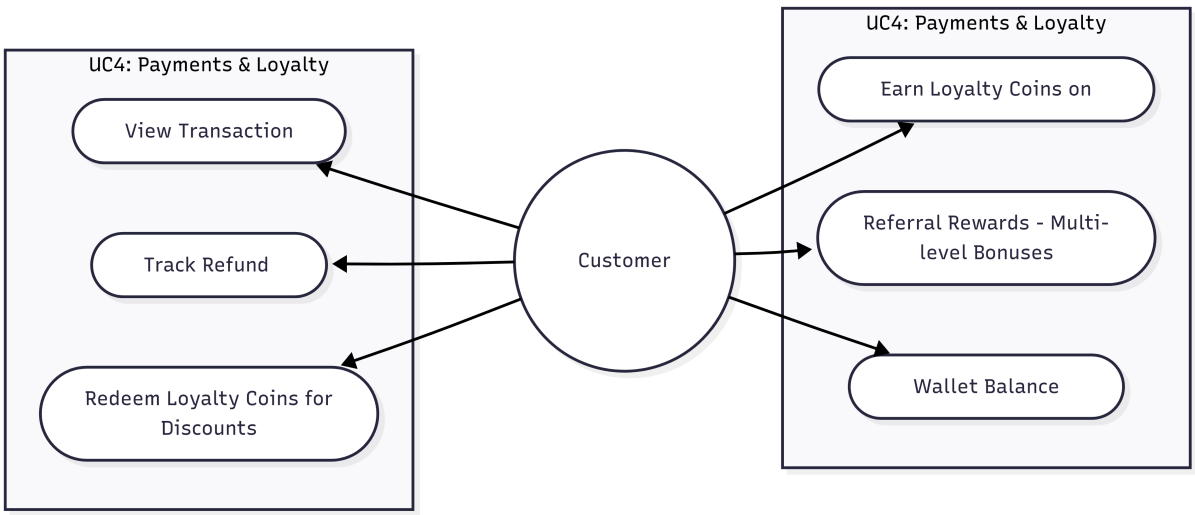
3.1.2 Use Case 2 - Travel Search, Booking & Ticketing



3.1.3 Use Case 3 - Booking Management and Travel Support

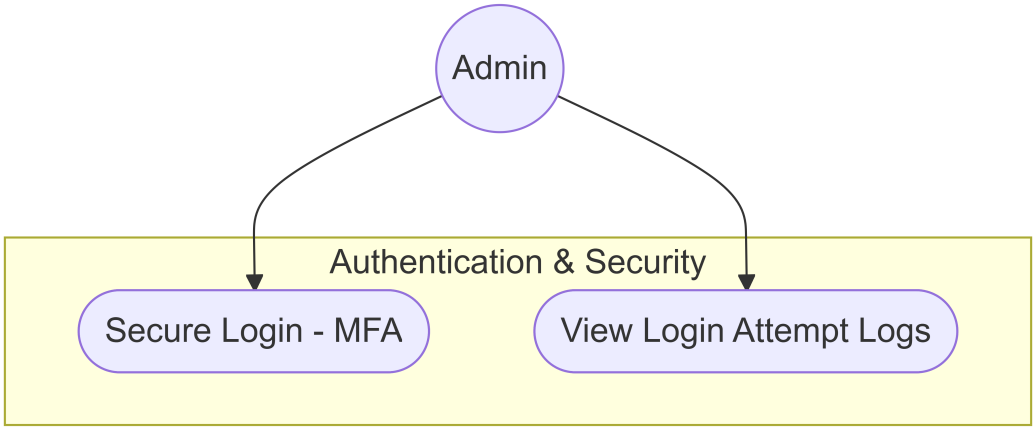


3.1.4 Use Case 4 - Payment and Loyalty Management

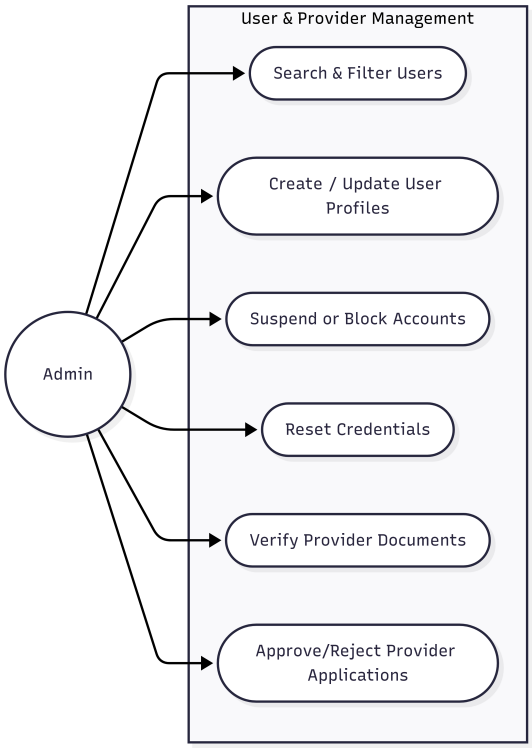


User Role : Admin

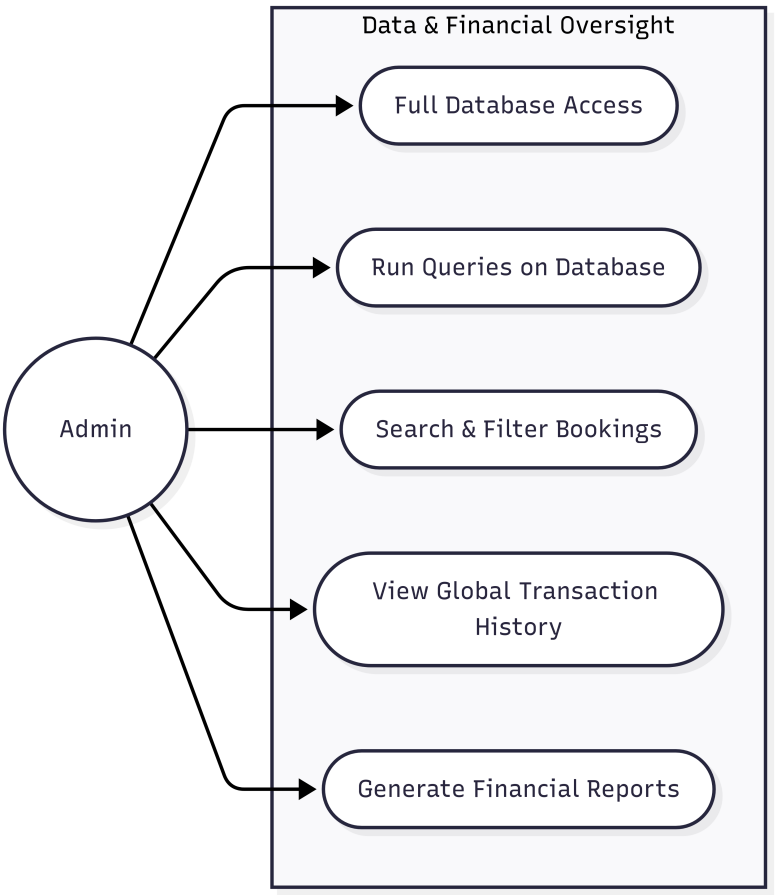
3.1.5 Use Case 5 - Admin Login & Security



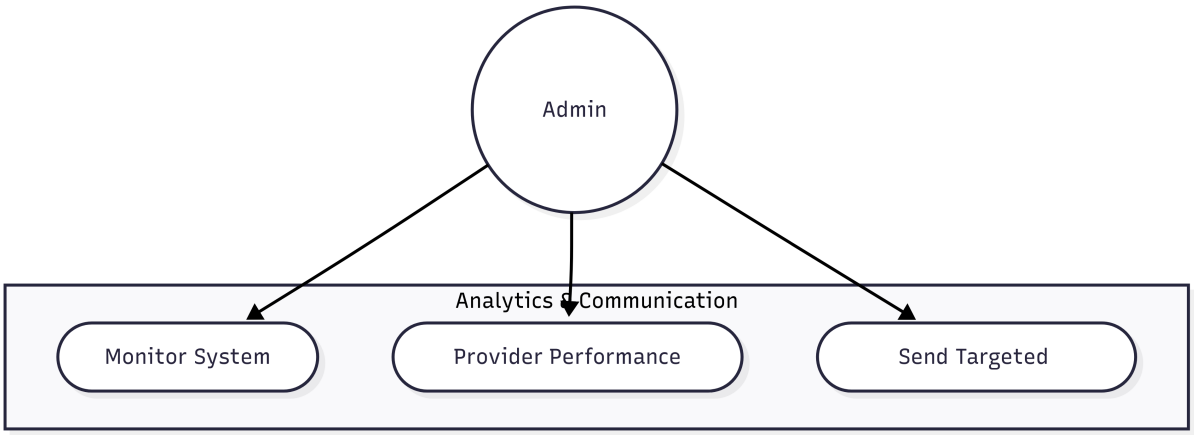
3.1.6 Use Case 6 - User & Provider Management



3.1.7 Use Case 7 - Data & Reports

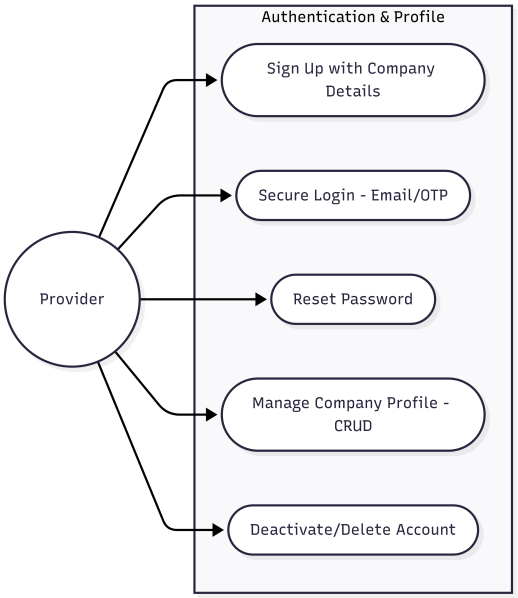


3.1.8 Use Case 8 - Monitoring & Communication

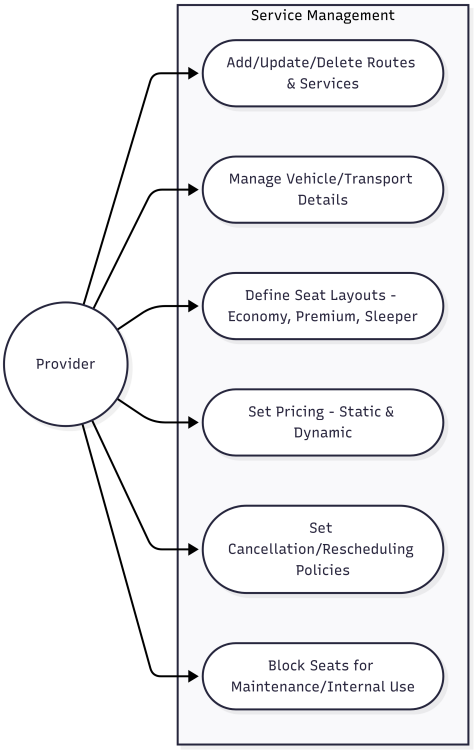


User Role : Travel Service Provider

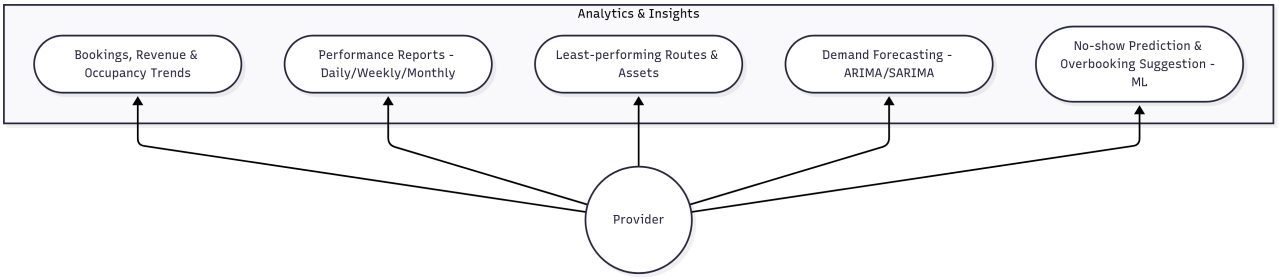
3.1.9 Use Case 9 - Registration & Profile Management



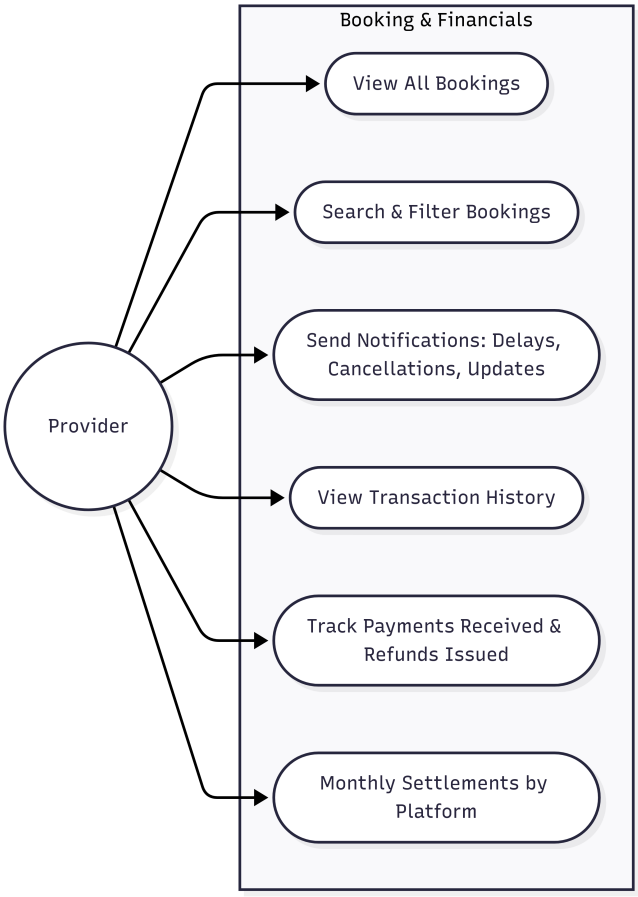
3.1.10 Use Case 10 - Service Management



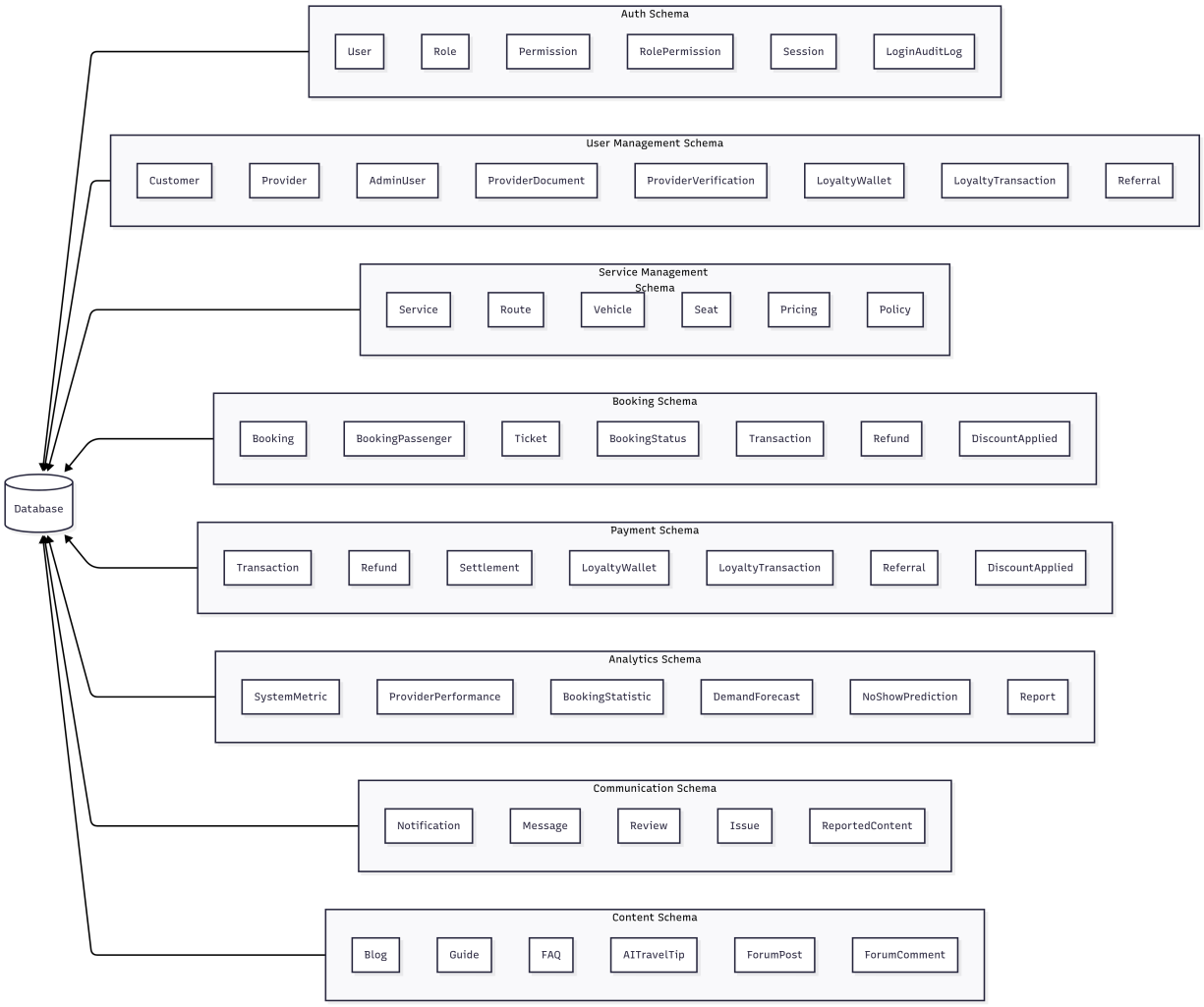
3.1.11 Use Case 11 - Analytics & Reports



3.1.12 Use Case 12 - Bookings & Financials



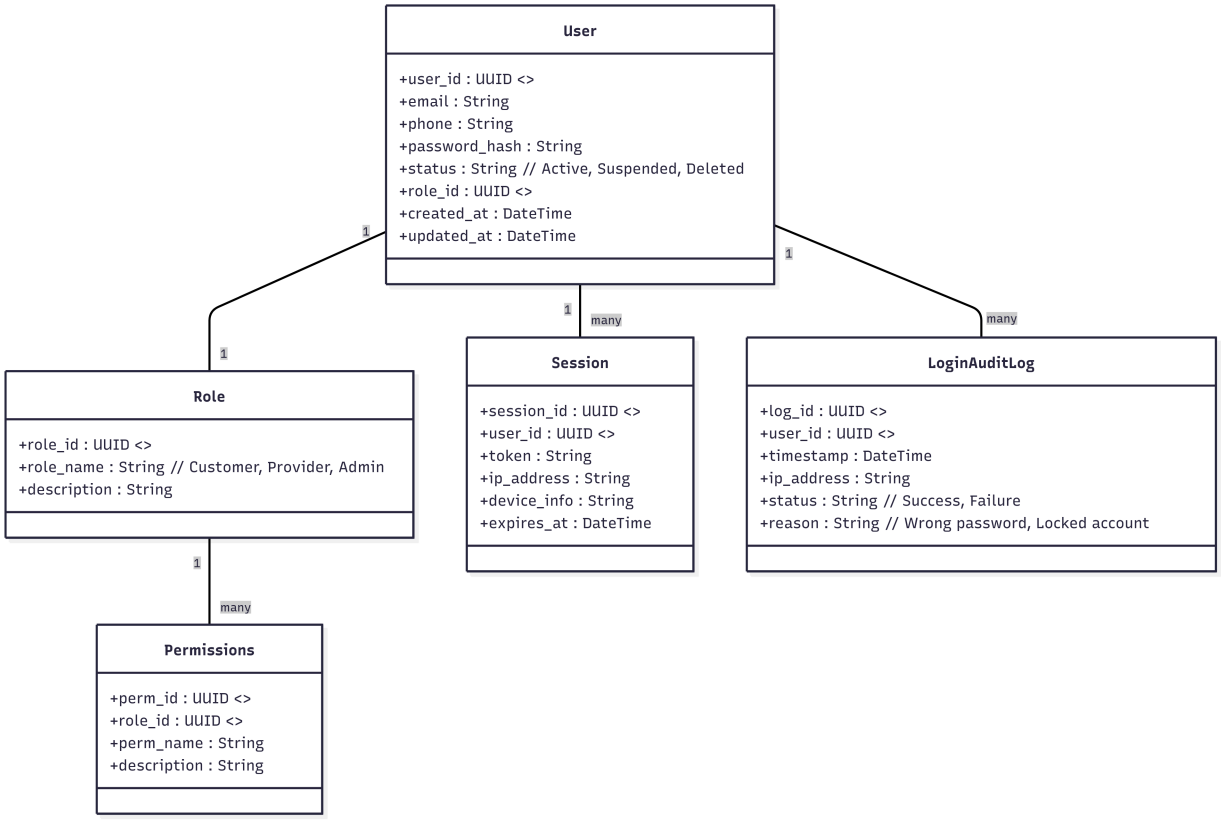
3.2 Class Diagrams



Class Schema Overview of Nexa

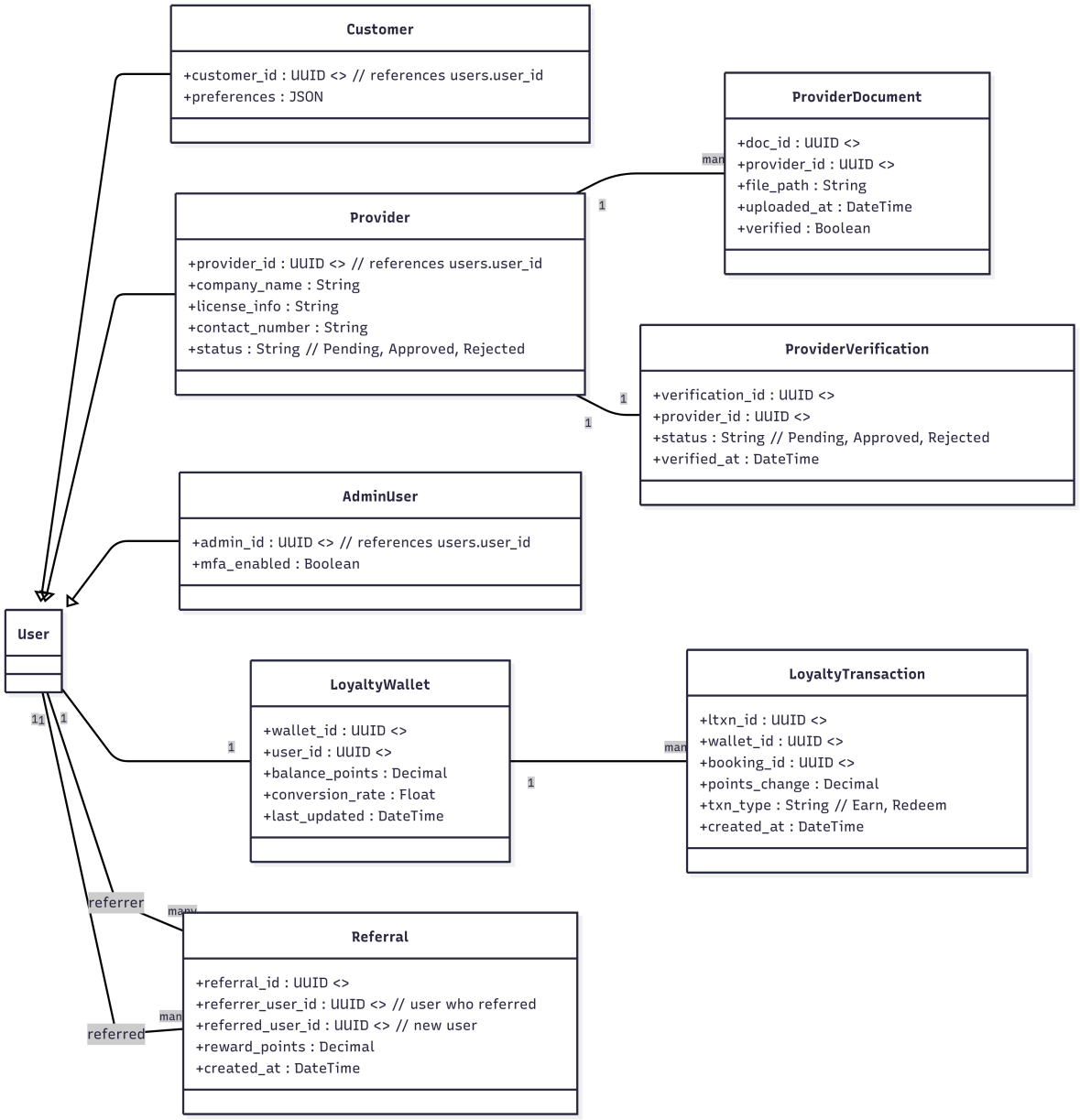
This is an overview of all the schema in the DB.

3.2.1 Authentication Schema



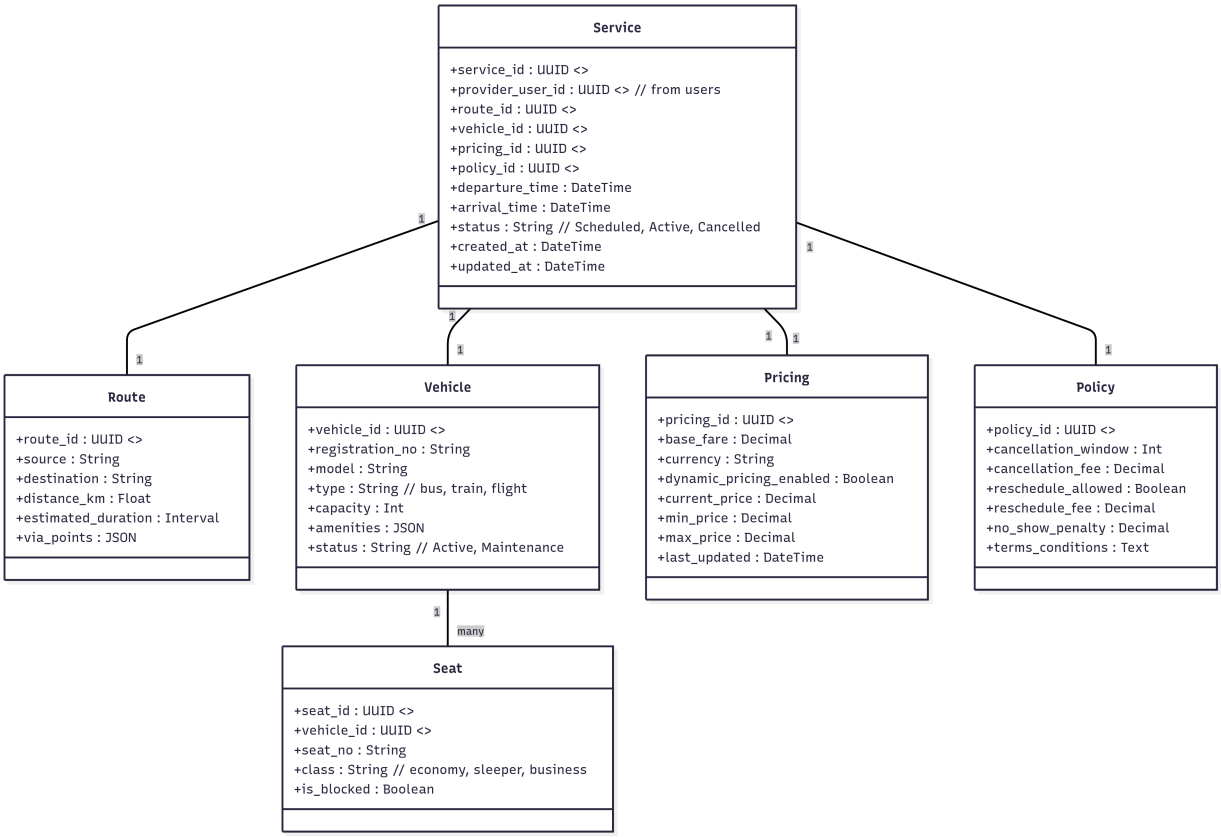
Authentication Schema Interactions

3.2.2 User's Schema



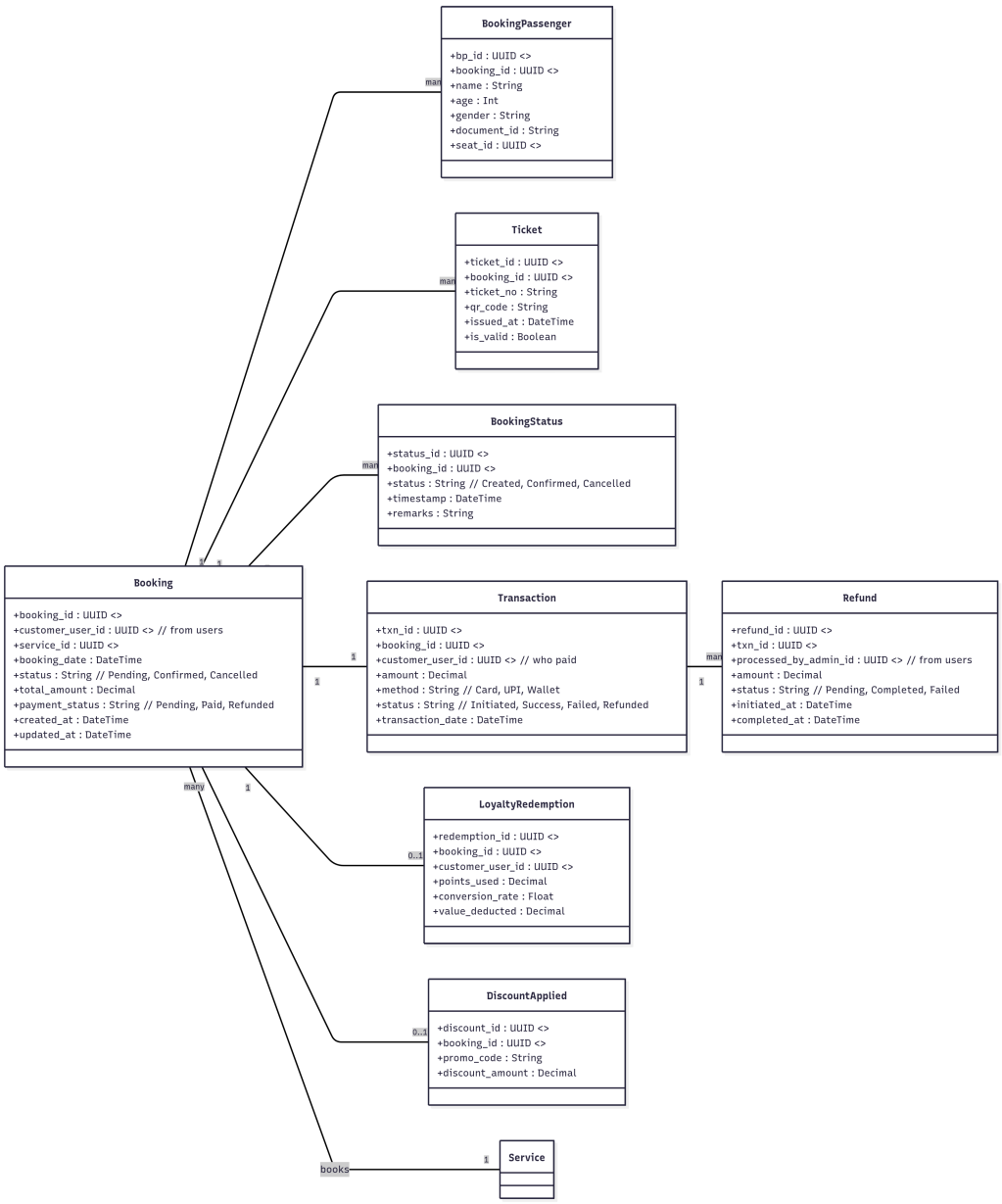
User Schema Interactions

3.2.3 Services's Schema



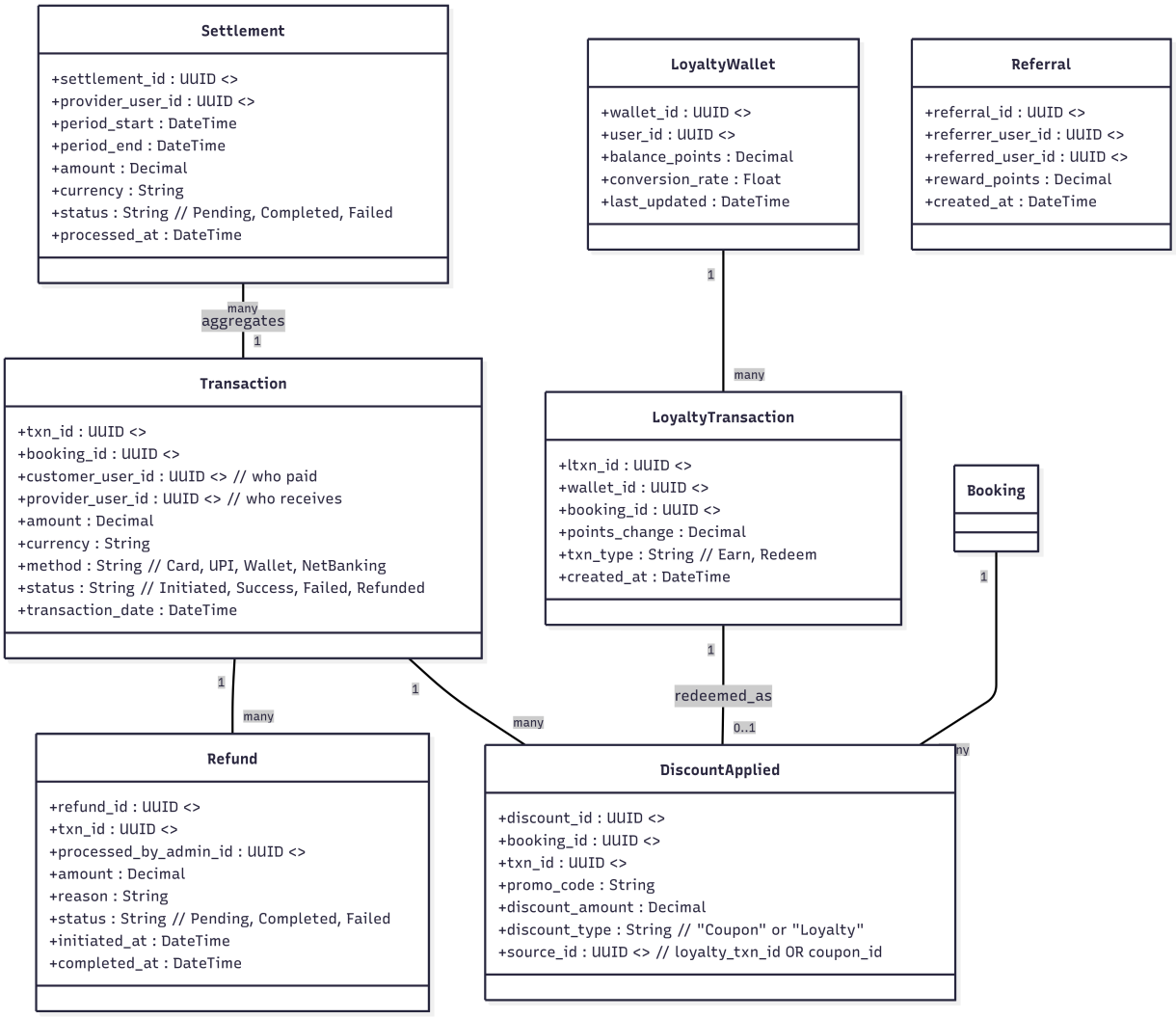
Services Schema Interactions

3.2.4 Booking's Schema



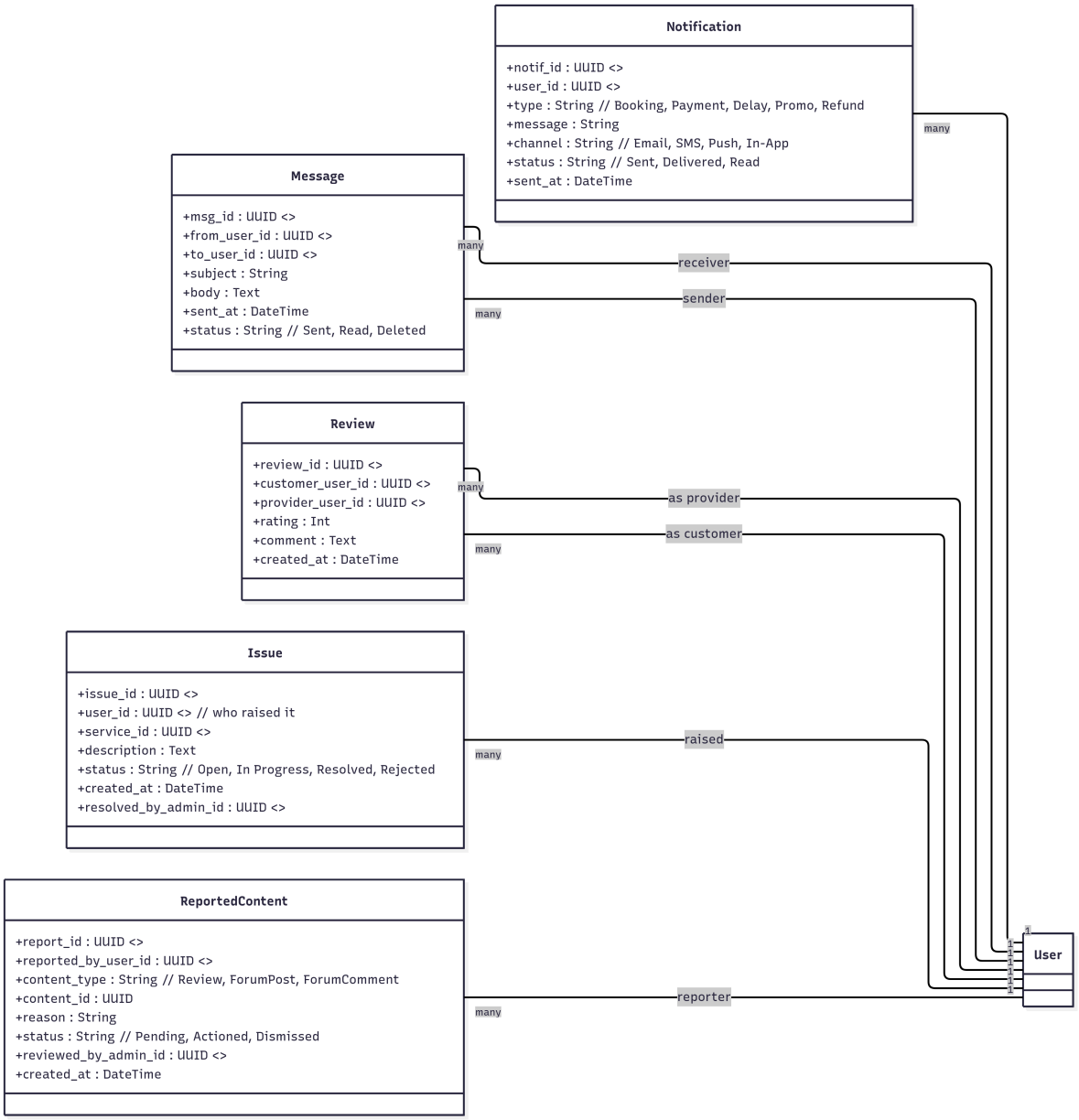
Booking Schema Interactions

3.2.5 Payment's Schema



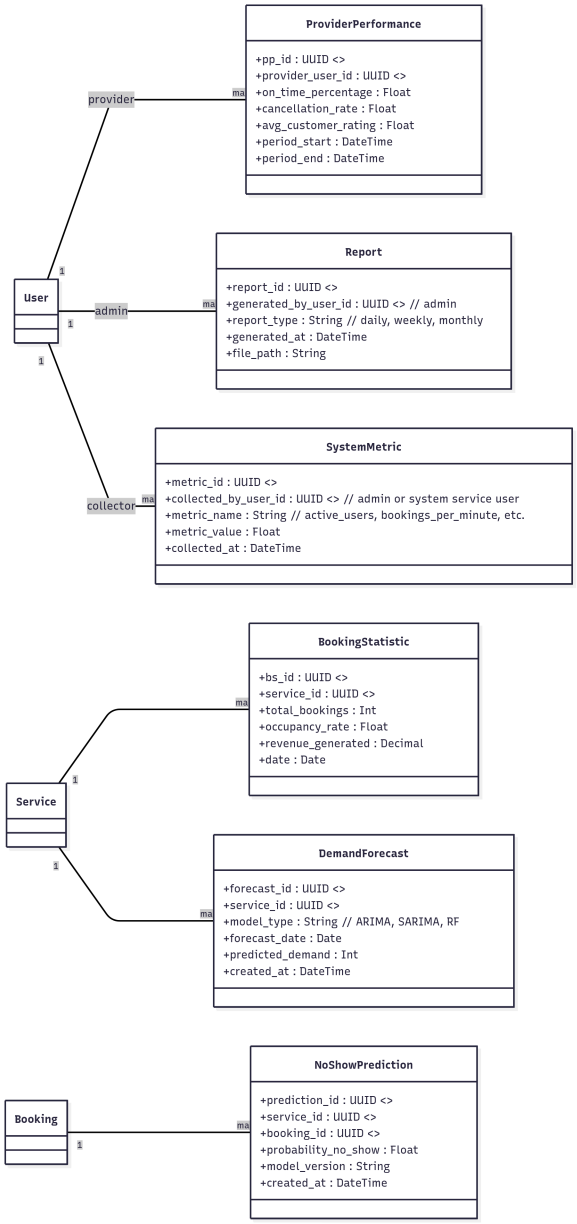
Payment Schema Interactions

3.2.6 Notification Schema



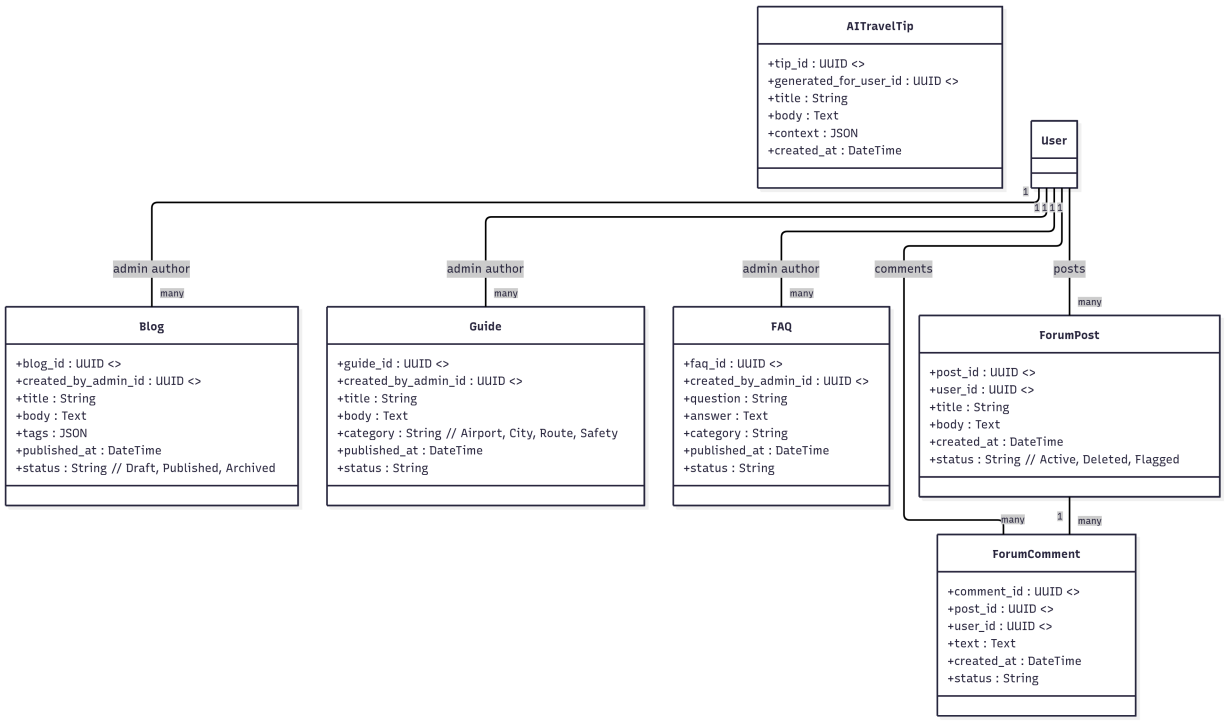
Notification Schema Interactions

3.2.7 AI Content Schema



AI Content Schema Interactions

3.2.8 Forum's Schema

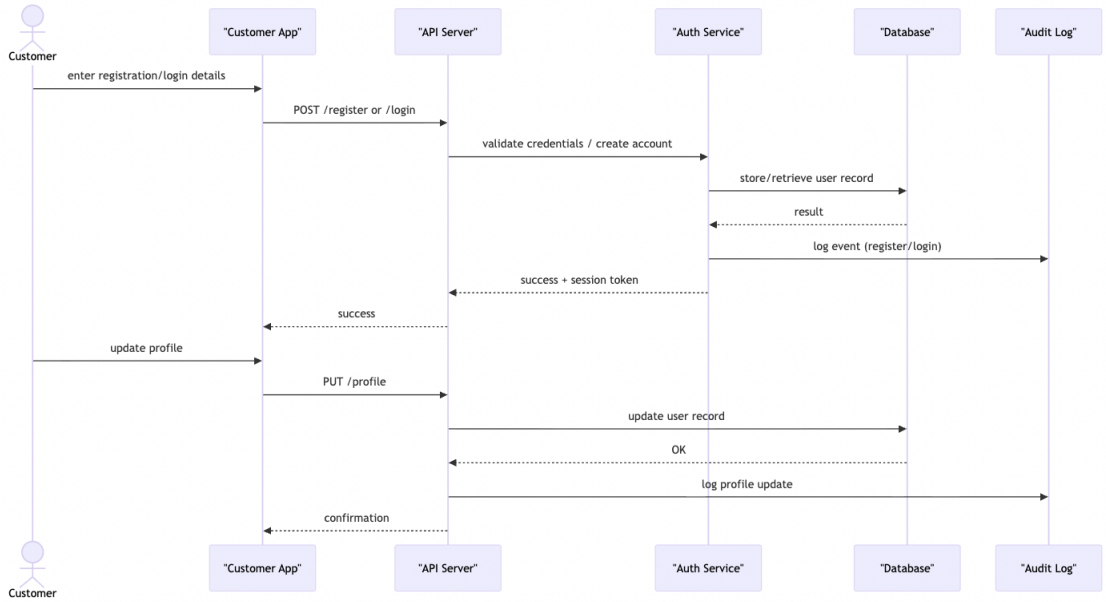


Forum's Schema Interactions

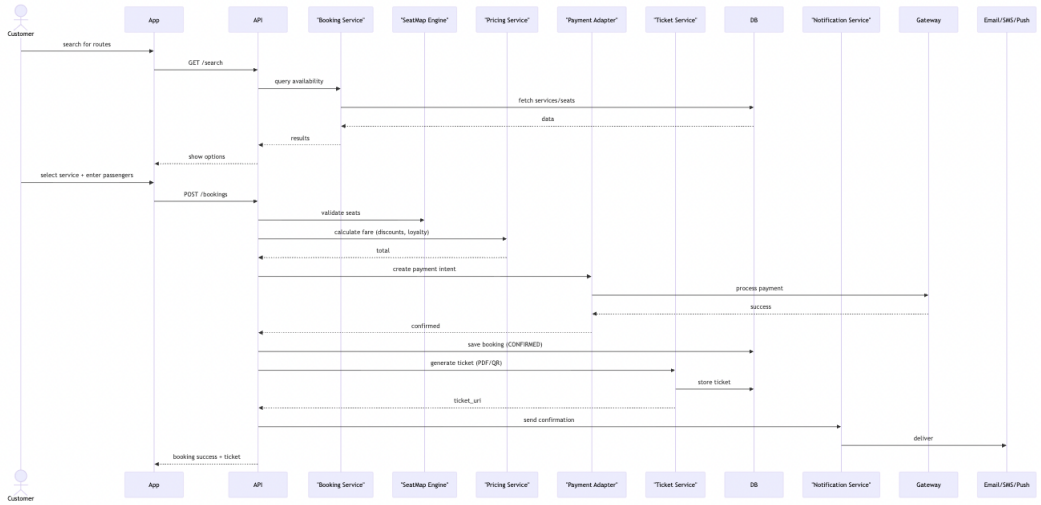
3.3 Sequence Diagrams

User Role : Customer

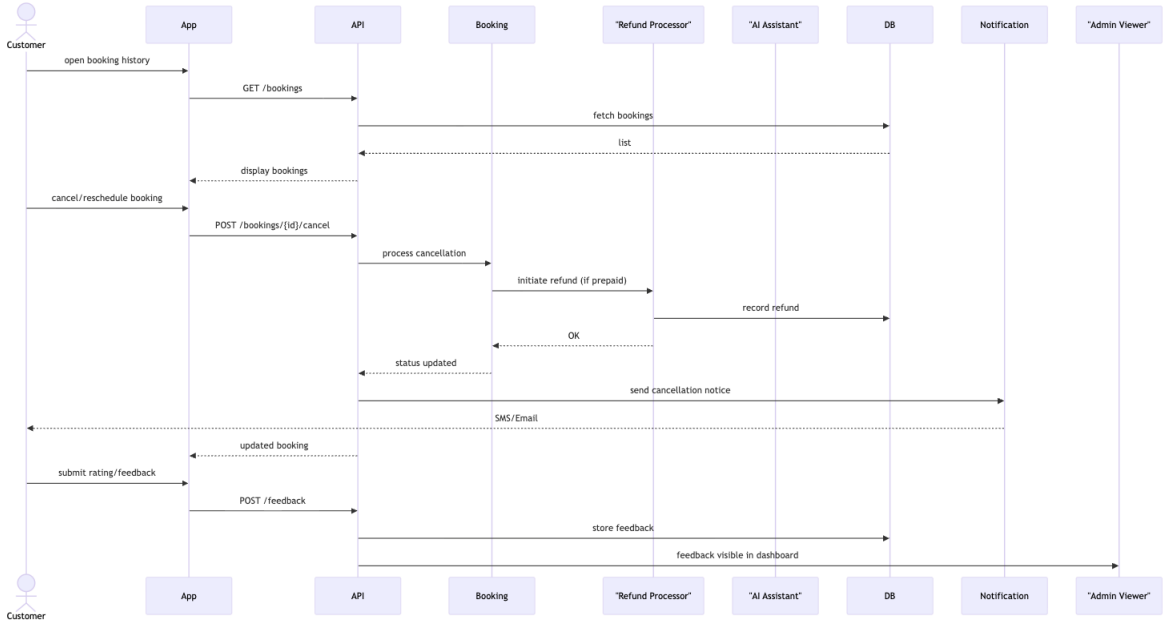
3.3.1 Sequence 1 - Registration & Profile Management



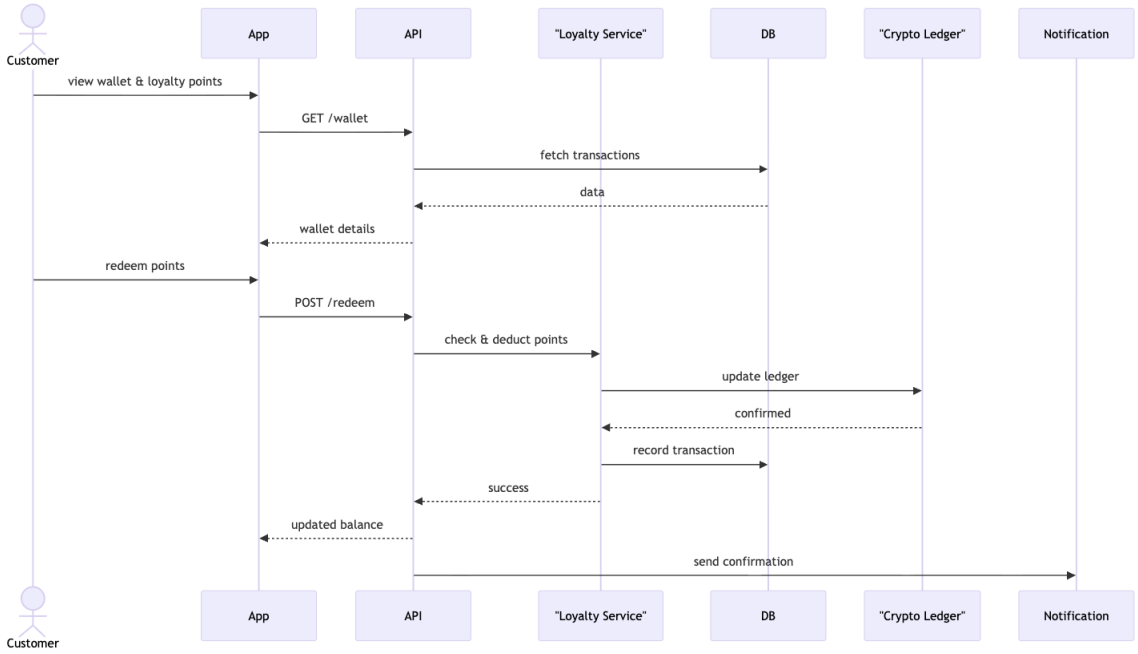
3.3.2 Sequence 2 - Travel Search, Booking & Ticketing



3.3.3 Sequence 3 - Booking Management and Travel Support

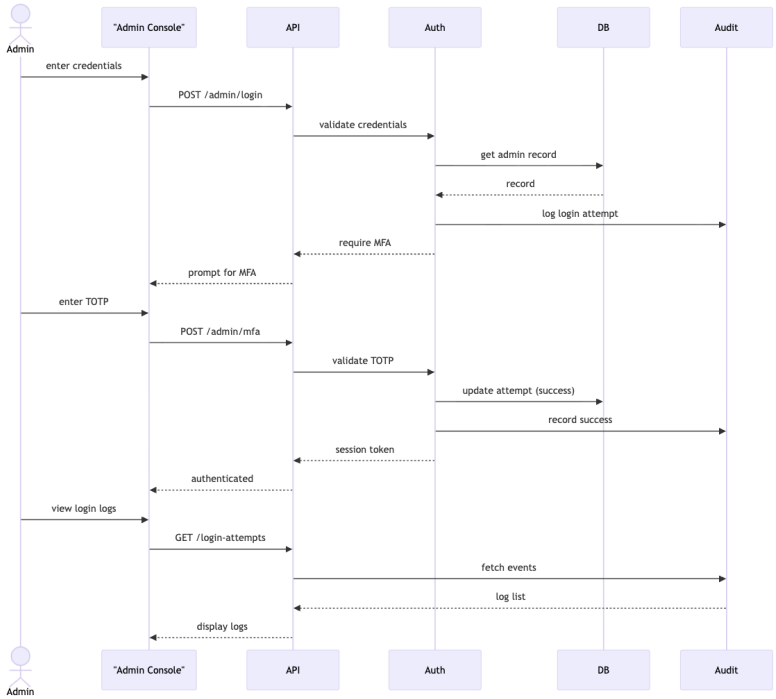


3.3.4 Sequence 4 - Payment and Loyalty Management

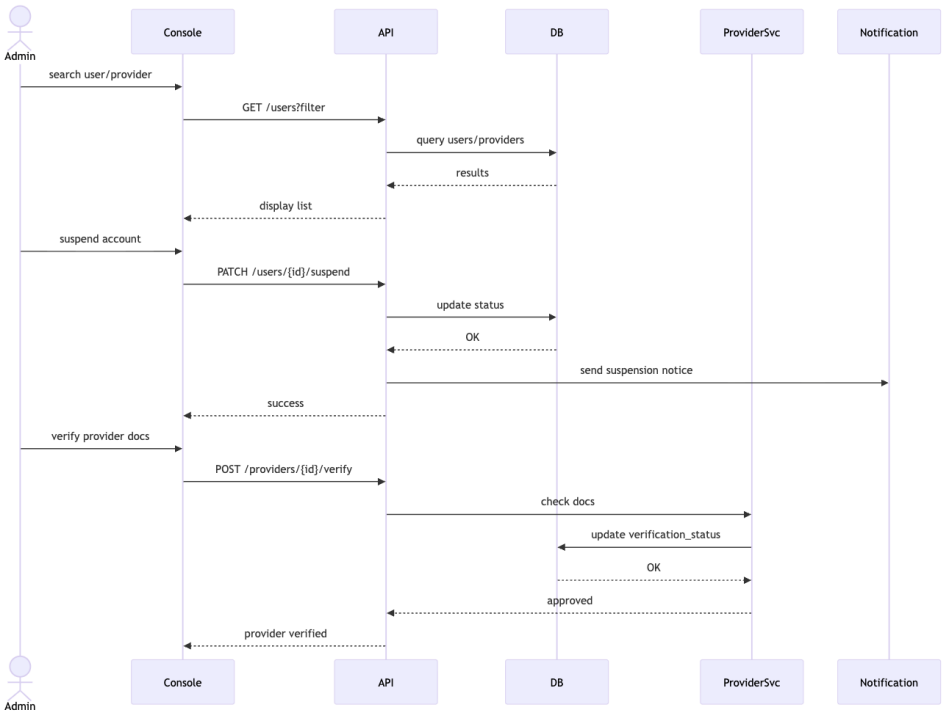


User Role : Admin

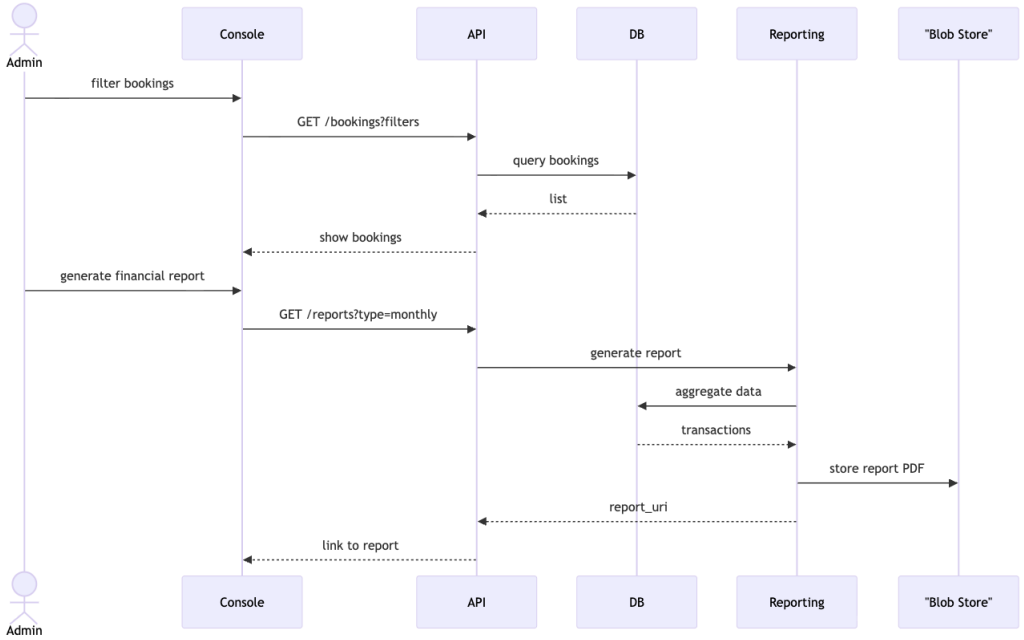
3.3.5 Sequence 5 - Admin Login & Security



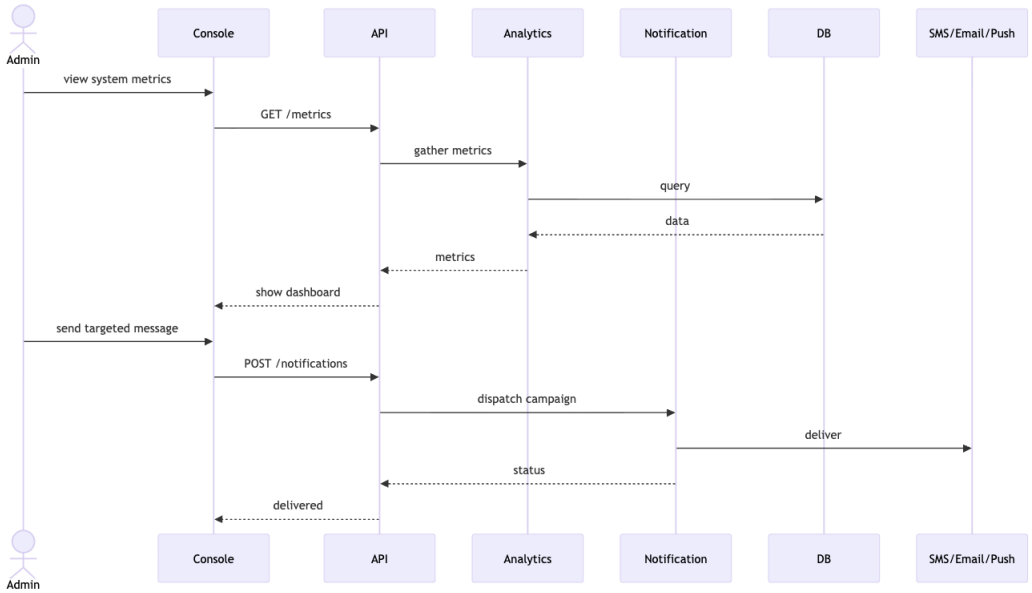
3.3.6 Sequence 6 - User & Provider Management



3.3.7 Sequence 7 - Data & Reports

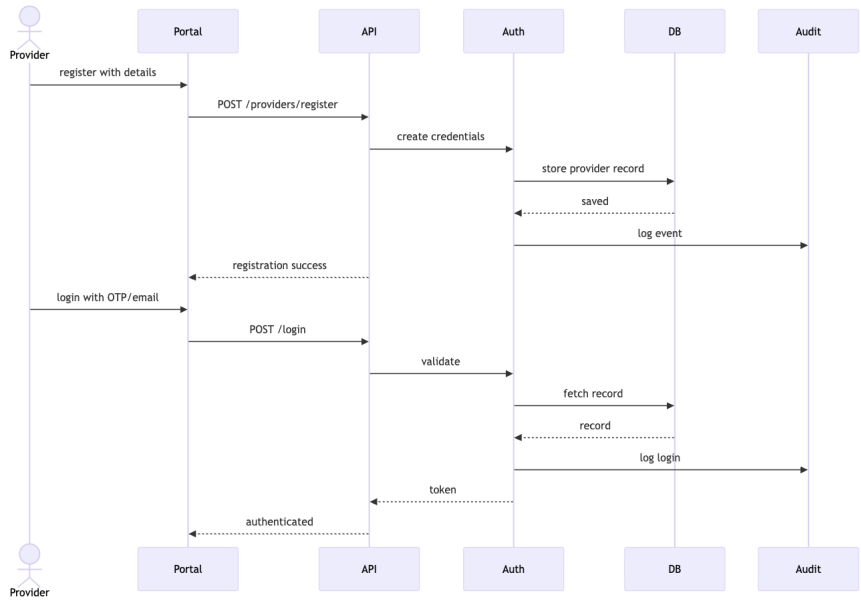


3.3.8 Sequence 8 - Monitoring & Communication

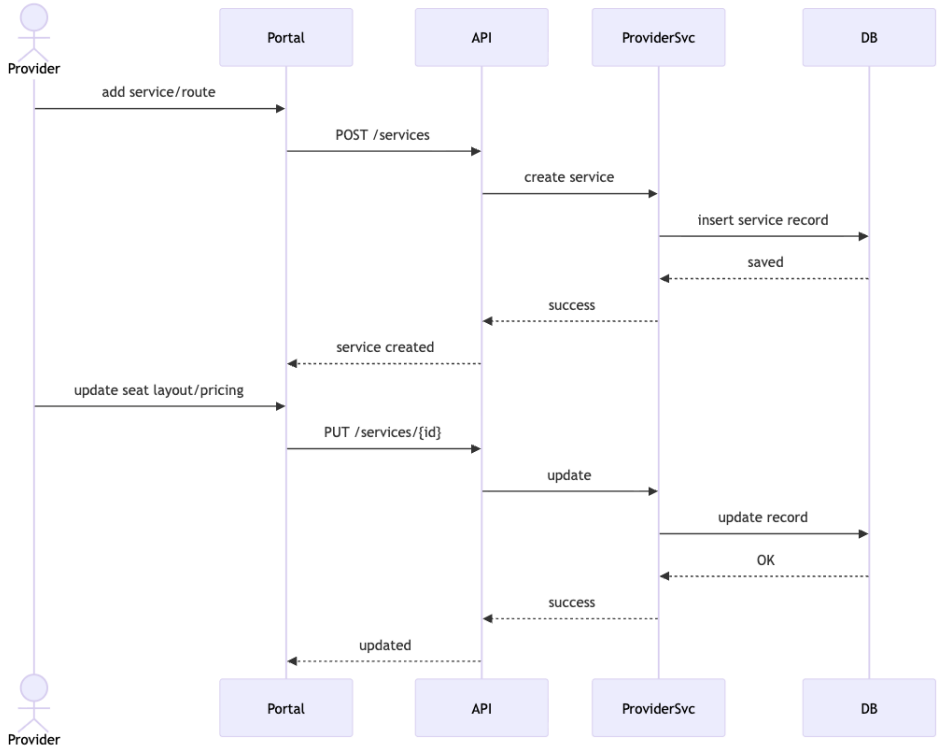


User Role : Travel Service Provider

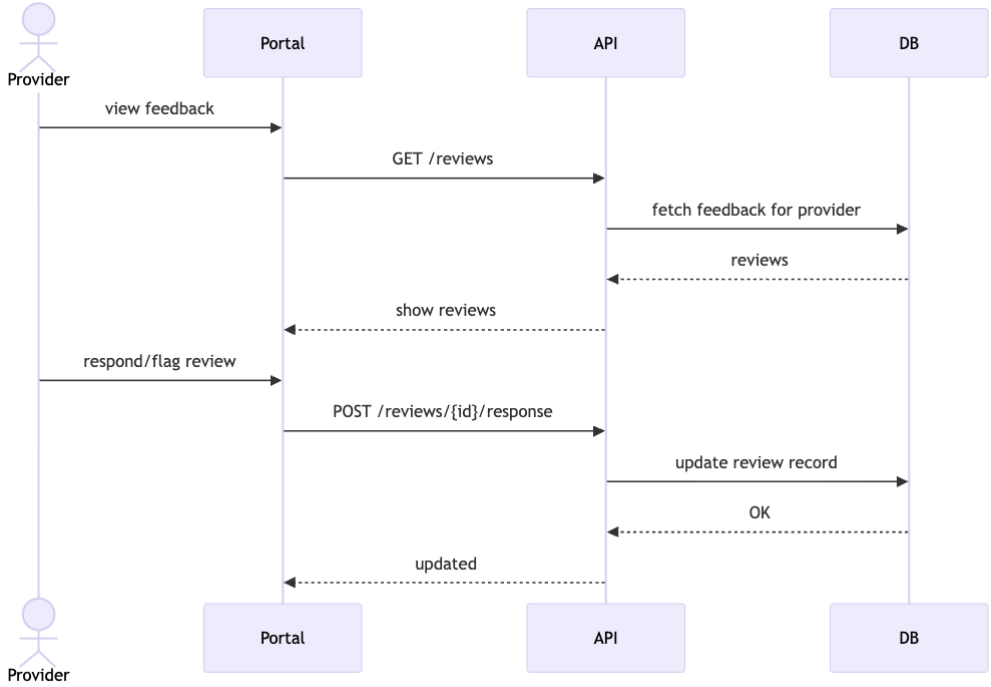
3.3.9 Sequence 9 - Registration & Profile Management



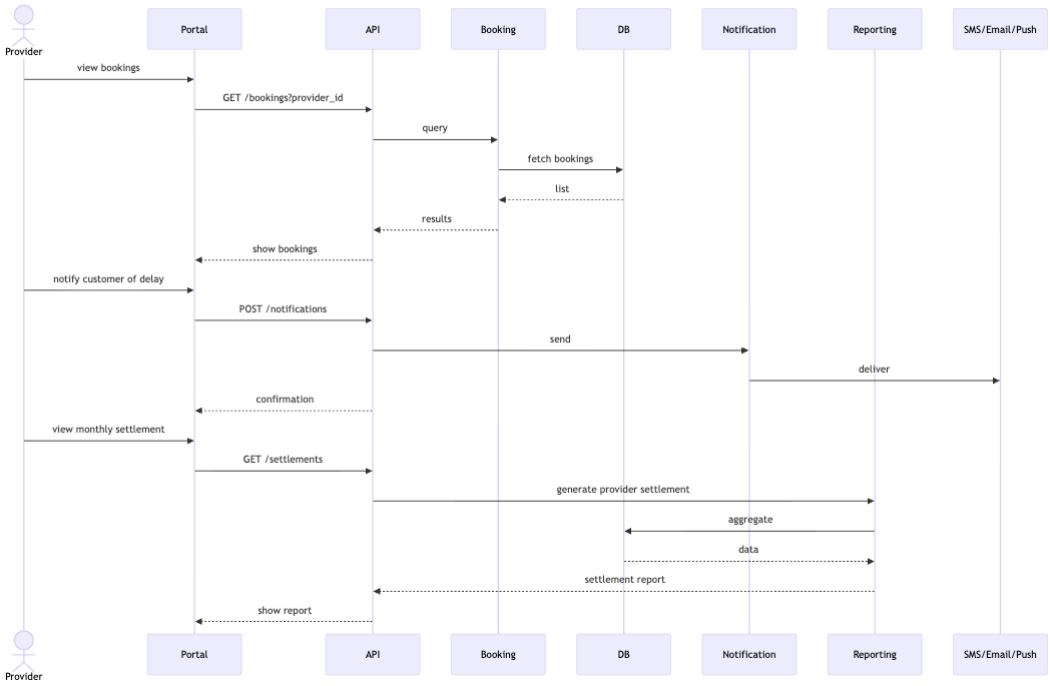
3.3.10 Sequence 10 - Service Management



3.3.11 Sequence 11 - Analytics & Reports

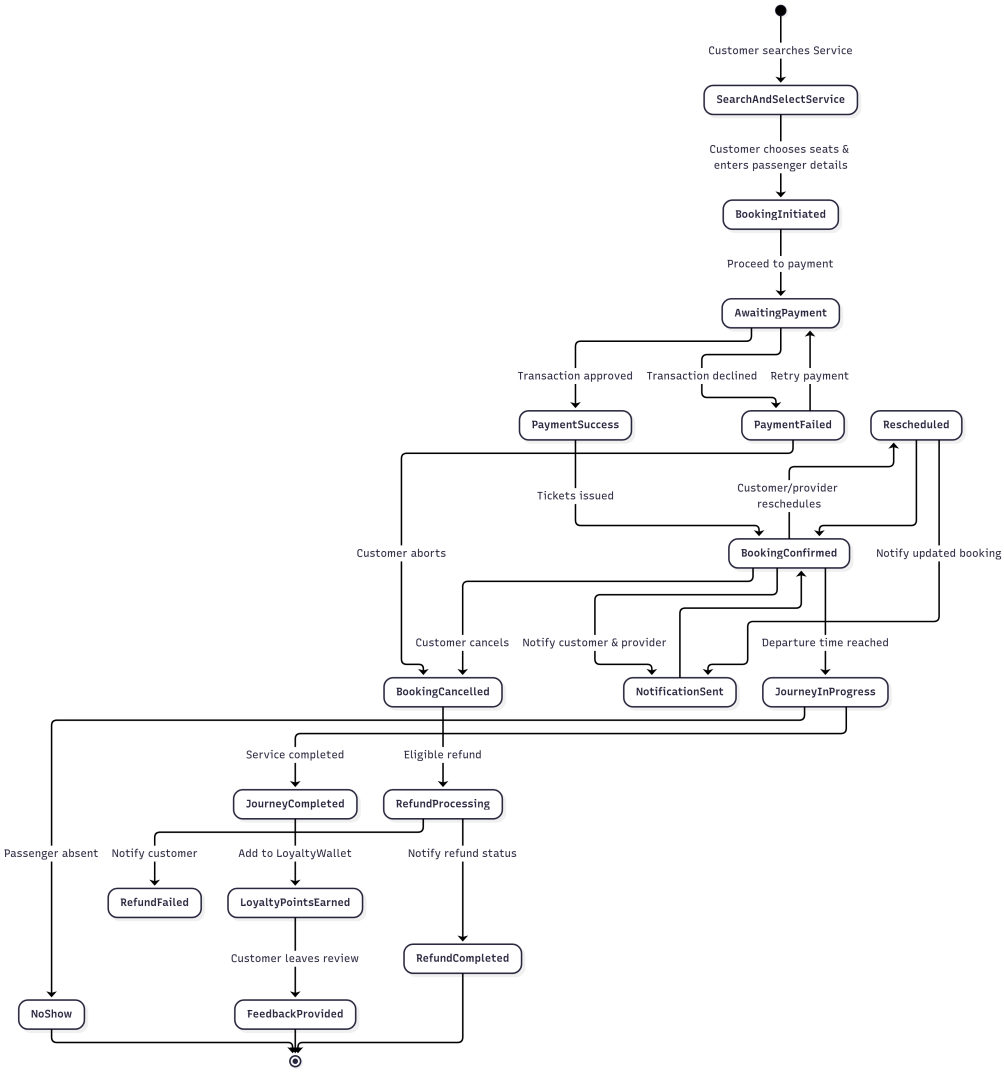


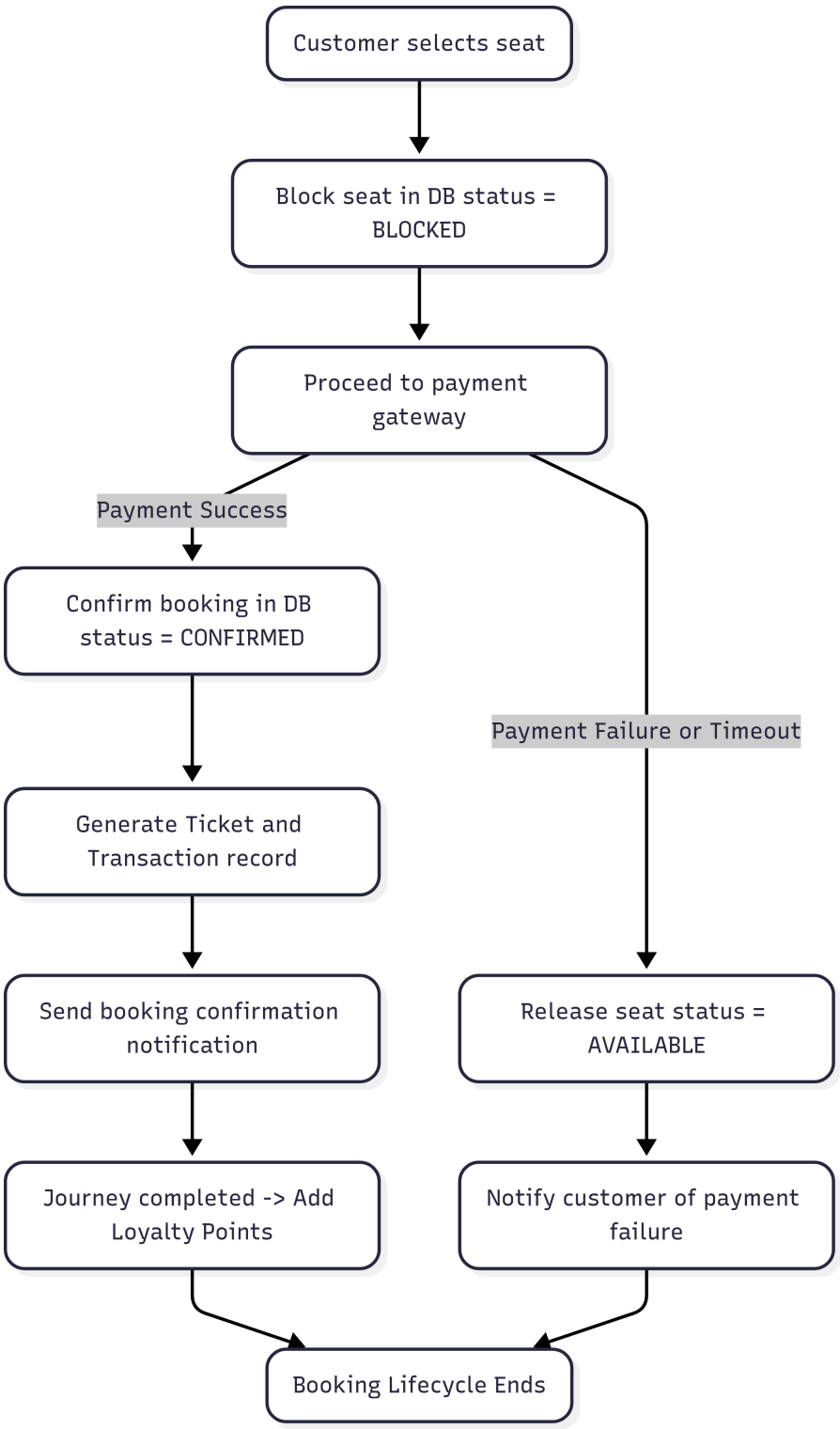
3.3.12 Sequence 12 - Bookings & Financials



3.4 State Diagrams

3.4.1 Overall state diagram



The image describes the high-level state diagram of the application from a user perspective. The customer can either be a subscribed customer or an unsubscribed customer.

3.4.2 Booking Sequence to Avoid Double Booking