Cairo University

Faculty of Computers and Artificial intelligence

Department of Software Engineering

Eventak

Supervised By

Dr/ Lamia Abo Zaid

Implemented by

20216055	**Doha Sherif**
20216140	**Mariam Ashraf**
20216141	**Mariam Emad**
20216142	**Antonios Elia**
20215059	**Youssef Osama**

Graduation Project

Academic Year 2024-2025

Table of Contents

Chapter 1: Introduction

1.1. Abstraction
1.2. Problem Definition
1.3. Eventak Solution

Chapter 2: Related Work

2.1. The Existing Similar Competitors

2.2. The main differences between competitors and Eventak

Chapter 3: Project Specifications

3.1. Stakeholders
3.2. Functional Requirements
3.3. Non-functional Requirements
3.4. Use Case Diagrams
3.5. Sequence Diagrams
3.6. Activity Diagrams
3.7. Component Diagram
3.8. Deployment Diagram
3.9. ERD Diagram
3.10. Class Diagram

Chapter 4: Prototype

Chapter 5: Work Plan

Chapter 1: Introduction

1.1. Abstract

People today increasingly want to celebrate important moments in their lives with special events. However, many lack the experience or knowledge to handle all the event preparations on their own. Even when they recognize the need to explore options, they often struggle to find suitable services and don’t know how to identify the best choices for their needs. On the other hand, many service providers across different event categories fail to reach enough customers simply because people are unaware they exist. Our project addresses these challenges by creating a centralized platform that offers everything a customer—whether an individual or an organization—might need to plan an event from start to finish. By seamlessly connecting customers with service providers through an easy-to-use, well-organized system, we simplify the entire event planning process.

1.2. Background

Event planning is a complex and time-consuming process that often requires coordinating multiple vendors, venues, and services across different platforms. Our project introduces an all-in-one event planning platform designed to simplify this experience by allowing users to book venues, select services (such as caterers, DJs, and photographers), manage event details, and send invitations—all in one centralized system. This eliminates the need for switching between apps, making phone calls, or handling separate payments. The platform aims to streamline event organization for customers while providing vendors and venue owners with greater visibility, efficient booking management, and digital tools to grow their businesses. Our motivation stems from the challenges faced by both event planners (who struggle with fragmented processes) and small vendors (who lack exposure and booking automation). By integrating these needs into a single, user-friendly platform, we enhance convenience for customers and business opportunities for service providers.

1.3. Problem Definition

Event planning is a huge process that often cause stress and inefficiency. Customers tend to visit multiple websites or contact different service providers individually to find the right venue, book services, and organize event details. This scattered process leads to confusion, delays, and sometimes missed bookings. On the other hand many vendors and venue owners struggle to attract enough customers because they lack a unified platform to show their services. They also have difficulty managing their availability, bookings, and payments in an organized way. In fact most places organize these things manually. Moreover, existing solutions often focus only on parts of the event planning process such as venue booking or service reservations but none provide a complete, integrated system that covers everything from searching and booking to event management and communication. Our project addresses these gaps by offering a single platform that connects customers, vendors, and admins, enabling smooth coordination, reliable bookings, and effective event management all in one place.

1.4. Eventak Solution

Eventak is our proposed solution to the challenges in event planning. It is a comprehensive digital platform that brings together customers, service providers, and venue owners into a single, streamlined ecosystem. The platform allows users to search for venues, compare services, manage event details, and complete bookings—all in one place. Eventak eliminates the need for multiple apps, calls, or manual coordination by offering an end-to-end system that simplifies the entire event planning journey. For vendors and venue owners, Eventak provides tools for listing services, managing availability, handling payments, and receiving bookings with minimal effort. By centralizing the process and improving visibility for service providers, Eventak not only enhances convenience for customers but also boosts business opportunities for vendors. Ultimately, it transforms event planning from a fragmented, stressful experience into a seamless and efficient process for all parties involved.

Chapter 2: Related Work

2.1. Existing Similar Competitors

1. Venu.ai

- Focus: Venue booking
- Strengths:
    - Filter venues by event type, price, capacity, and location.
    - Chat with hosts, book site visits, and view photos.
- Weaknesses:
    - No real-time availability or customer reviews.
    - Lacks event management features.

2. Eventplanner.net

- Focus: Event services + task management
- Strengths:
    - Search for venues/services by price, location, and capacity.
    - Task assignment and deadline tracking.
- Weaknesses:
    - Redirects users for bookings (no in-app payments).
    - Poor UX/UI design.

3. TLT Concepts

- Focus: Venue showcase
- Strengths: Displays past events for better visualization.
- Weaknesses: No online booking; limited venue options.

4. Plein Air (Egypt)

- Focus: Wedding packages
- Strengths: Pre-packaged deals.
- Weaknesses: No chat, direct booking, or modern UI.

5. Hafllah (Qatar)

- Focus: Vendor marketplace
- Strengths: Search by date/location; favorites system.
- Weaknesses: No venue booking or package deals; mobile-only.

6. Eventective (Global)

- Focus: Multi-service vendor platform
- Strengths: Wide vendor range; CMS event websites.
- Weaknesses: No real-time booking or planning tools.

2.2. Comparison

Feature	Eventak	Venu.ai	EventPlanner	Tlt concepts	Plien Air	Hafllah	Eventective
Direct Booking	✓	✓	✗	✗	✗	✓	✗
Chatbot	✓	✗	✓	✗	✗	✗	✗
Dashboard for Event Management	✓	✗	✓	✗	✗	✓	✓
Generate Event Website	✓	✗	✗	✗	✗	✗	✓
Blogs	✓	✗	✓	✗	✓	✗	✗
Rating & Reviews	✓	✗	✓	✗	✗	✗	✗
Vendor portfolio	✓	✗	✗	✗	✗	✓	✗
Customer Loyalty and Reward Systems	✓	✗	✗	✗	✗	✗	✗
Customized Services (not from the venue)	✓	✗	✓	✗	✗	✓	✓
Various Venue Booking Packages	✓	✓	✓	✗	✓	N\A	✓
Event Planning Tools	✓	✗	✓	✗	✗	✗	✓
Site Visit (No charge)	✓	✓	✓	✗	✗	✗	✗
Recommendations based on your requirements	✓	✗	✗	✗	✗	✓	✓
See available dates	✓	✗	✗	✗	✗	✓	✗
Map Navigation	✓	✗	✗	✗	✗	✗	✓

Chapter 3: Project Specifications

3.1. Stakeholders

The application's stakeholders can be divided into two main categories based on their level of engagement with the system.

Primary stakeholders include end-users , service providers ,and administrators:

1. **Customers (End Users)**

Role: Book and manage services through the platform.

2. **Service Providers**

Role: Offer and manage services (e.g., catering, cleaning, repairs).

3. **Administrators (System Operators)**

Role: Oversee platform operations and ensure smooth functioning.

Secondary stakeholders consist of external systems such as payment gateways and mapping APIs that enable critical functionalities, along with customer support teams that assist users.

4. **Developers (Technical Team)** Role: Build, maintain, and enhance the application.
5. **Third-Party API Providers**(e.g., MapsAPI, PaymentGateway)

3.2. Functional Requirements

The functional requirements define the core system capabilities needed to support administrators, customers, and vendors, ensuring all user roles can effectively use the platform.

1. **Admin Features:**
    - 1.1. **User Management**
        - 1.1.1. Manage user accounts (view, suspend)
    - 1.2. **Service Management**
        - 1.2.1. Manage service provider categories (caterers, photographers, DJs, etc.)
        - 1.2.2. Manage event categories (wedding, birthday, Seminars, etc.)
        - 1.2.3. Oversee packaged services
        - 1.2.4. Set and update cancellation policies
        - 1.2.5. Moderate reviews and ratings
        - 1.2.6. Manage automated invitation system templates
    - 1.3. **Booking & Reservation Management**
        - 1.3.1. View and manage all bookings
        - 1.3.2. Monitor booking statuses and transactions
    - 1.4. **Dashboard & Reporting**
        - 1.4.1. Access aggregated event and booking statistics
        - 1.4.2. Track customer loyalty and reward system usage
    - 1.5. **Blogs Management**
        - 1.5.1. Publish and manage blog content (tips, trends and guides)
2. **Customer Features:**
    - 2.1. **Venue Search & Booking**
        - 2.1.1. Search and filter venues by location, capacity, date availability, and price (including holiday pricing)
        - 2.1.2. View venue details (photos, max capacity, pricing, available dates)
        - 2.1.3. View nearby venues via map navigation

2.1.4. See immediate date availability without contacting host.

2.1.5. Make direct reservations via website with secure payment gateway .

2.1.6. Book free site visit

2.2. **Service Booking**

2.2.1. Book standalone services independently or with venue reservation

2.2.2. Choose from multiple service providers for services (caterers, DJs, photographers, etc.)

2.2.3. View packaged services and discounts from multiple providers.

2.3. **Event Management Tools**

2.3.1. Access personalized event dashboard per account.

2.3.2. Use event planning tools per event:

2.3.2.1. Auto-generated or customizable todo lists

2.3.2.2. Create and customize event website with RSVP system (CMS)

2.3.2.3. Generate printable event timeline

2.3.2.4. Send automated invitations to uploaded email/phone lists

2.3.2.5. Track budget (estimated vs actual expenses)

2.3.3. View sold services and venues added by the system automatically

2.4. **Communication & Support**

2.4.1. Use chatbot for event management recommendations as the chatbot can read the context of the event details in each tool.

2.5. **Feedback & Loyalty**

2.5.1. Rate and review services and service providers after event.

2.5.2. Participate in customer loyalty and reward programs.

2.6. **Account Management**

2.6.1. Manage personal profile.

2.6.2. View booking and payment history.

2.6.3. Cancel reservations following cancellation policies.

3. **Service Provider Features:**

3.1. **Profile & Portfolio Management**

3.1.1. Create and update vendor profile with service details, photos, pricing.

3.1.2. Upload and manage portfolio and activity history.

3.2. **Availability & Booking Management**

3.2.1. Update and manage venue or service availability calendar.

3.2.2. Track reservations for venues or service bookings.

3.2.3. Accept or decline bookings where applicable.

3.2.4. View booking details and payment status.

3.3. **Service providers management dashboard.**

3.3.1. Access dashboard to track reservations and service requests

3.3.2. View ratings, reviews, and feedback from customers

3.3.3. Manage packaged services and special offers.

3.4. **Communication**

3.4.1. Receive automated invitations for events if providing services

3.4.2. Respond to customer inquiries or booking requests

3.3. Non-functional Requirements

1. **Performance**

1.1. The system should handle **≥1,000 concurrent users** during peak times.

1.2. Page load times should be **<2 seconds** for 95% of requests.

1.3. Search results (venues/services) should load in **<1 second**.

2. **Scalability**

2.1. The system shall be designed to scale to accommodate future growth in user base and service listings

2.2. It should be easy to add new services, features, or categories without impacting system performance.

3. **Security**

   3.1. The system shall support secure user authentication using encrypted passwords (e.g., bcrypt).

   3.2. All sensitive data (such as payment and personal information) shall be transmitted over HTTPS using SSL/TLS.

   3.2. Role-based access control shall be implemented to prevent unauthorized access (e.g., admin, service providers roles).

4. **Reliability & Availability**

   4.1. The system shall be available 99.5% of the time, excluding scheduled maintenance

   4.2. Critical operations like bookings and payments shall be logged and confirmed with transactional integrity.

5. **Usability**

   5.1. Achieve **≥85% satisfaction** in user testing for core workflows.

   5.2. Mobile-responsive design (support **iOS/Android + modern browsers**).

   5.3. The system shall provide a user-friendly interface that can be used by individuals with basic computer and mobile literacy.

6. **Compatibility**

   6.1. Support **Chrome, Safari, Firefox, Edge** (latest 2 versions).

   6.2. API compatibility with major payment gateways (**Stripe, PayPal**).

7. **Maintainability**

   7.1. The system architecture shall be modular to support easier maintenance.

   7.2. Code should follow standard coding practices and include documentation to simplify future developer involvement.

8. **Legal & Compliance**

   8.1. Clear **terms of service** and **cancellation policy** displays.

3.4. Use Case Diagrams

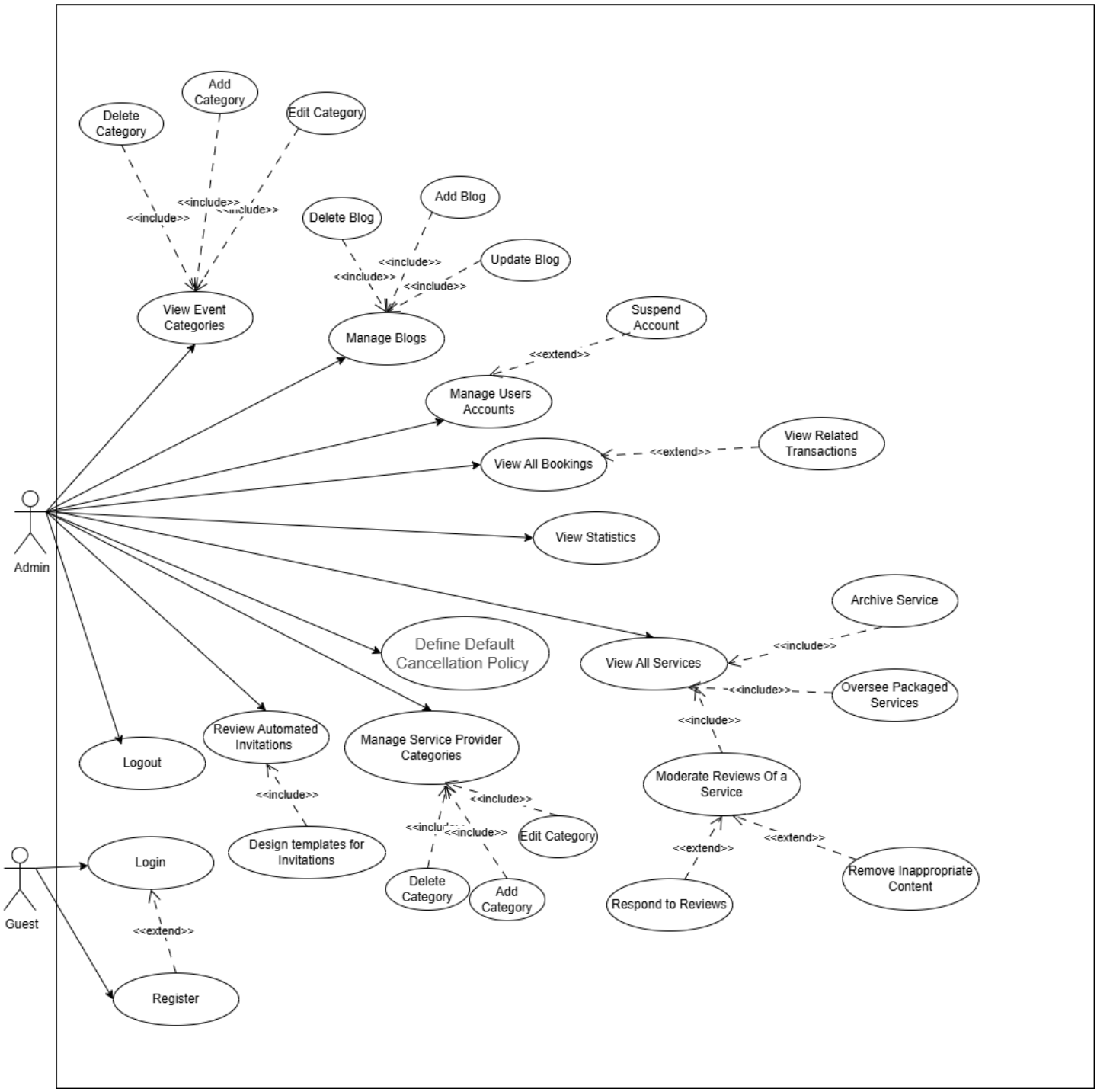


Figure 1 : Admin Use Case Diagram

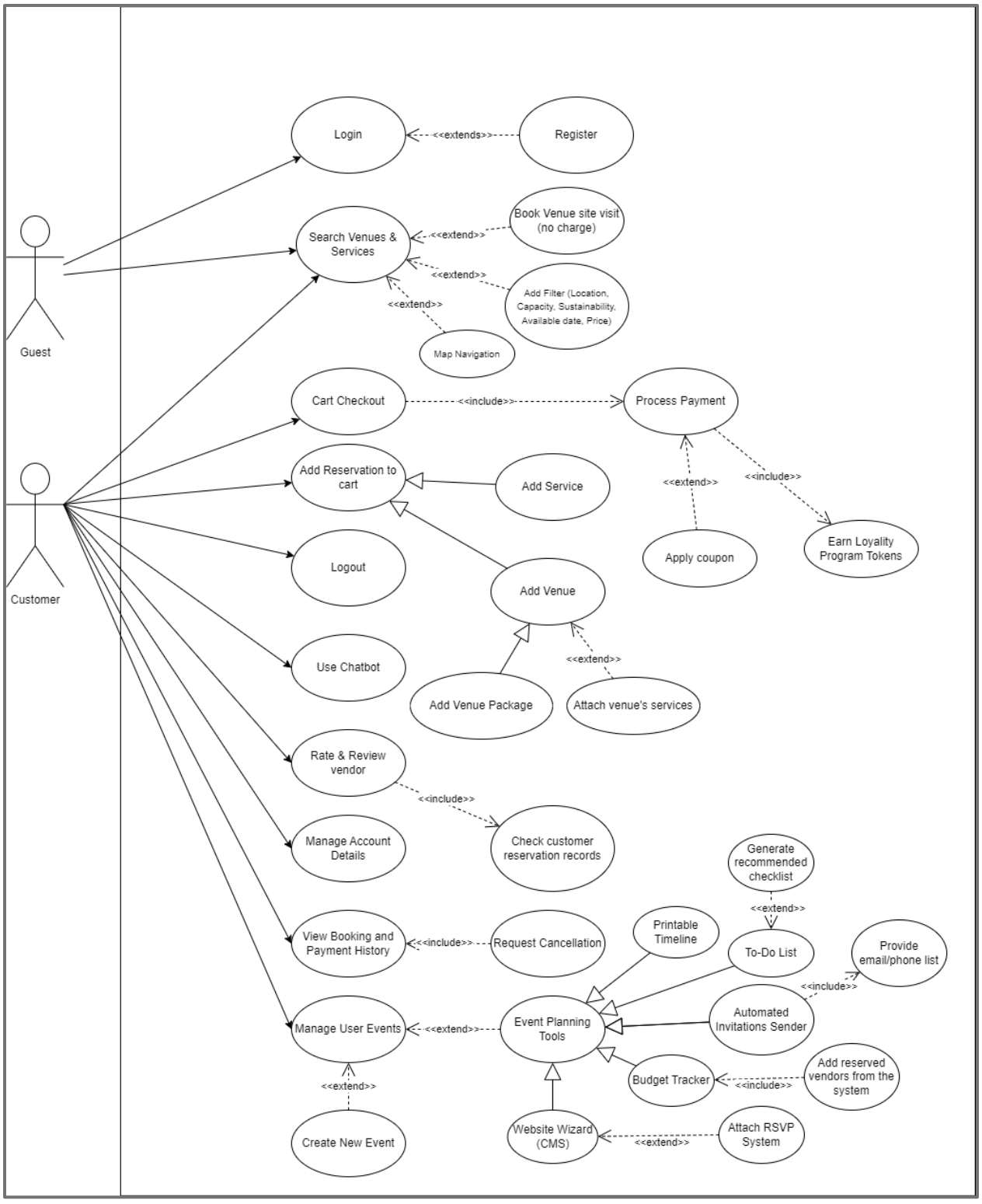


Figure 2 : Customer Use Case Diagram

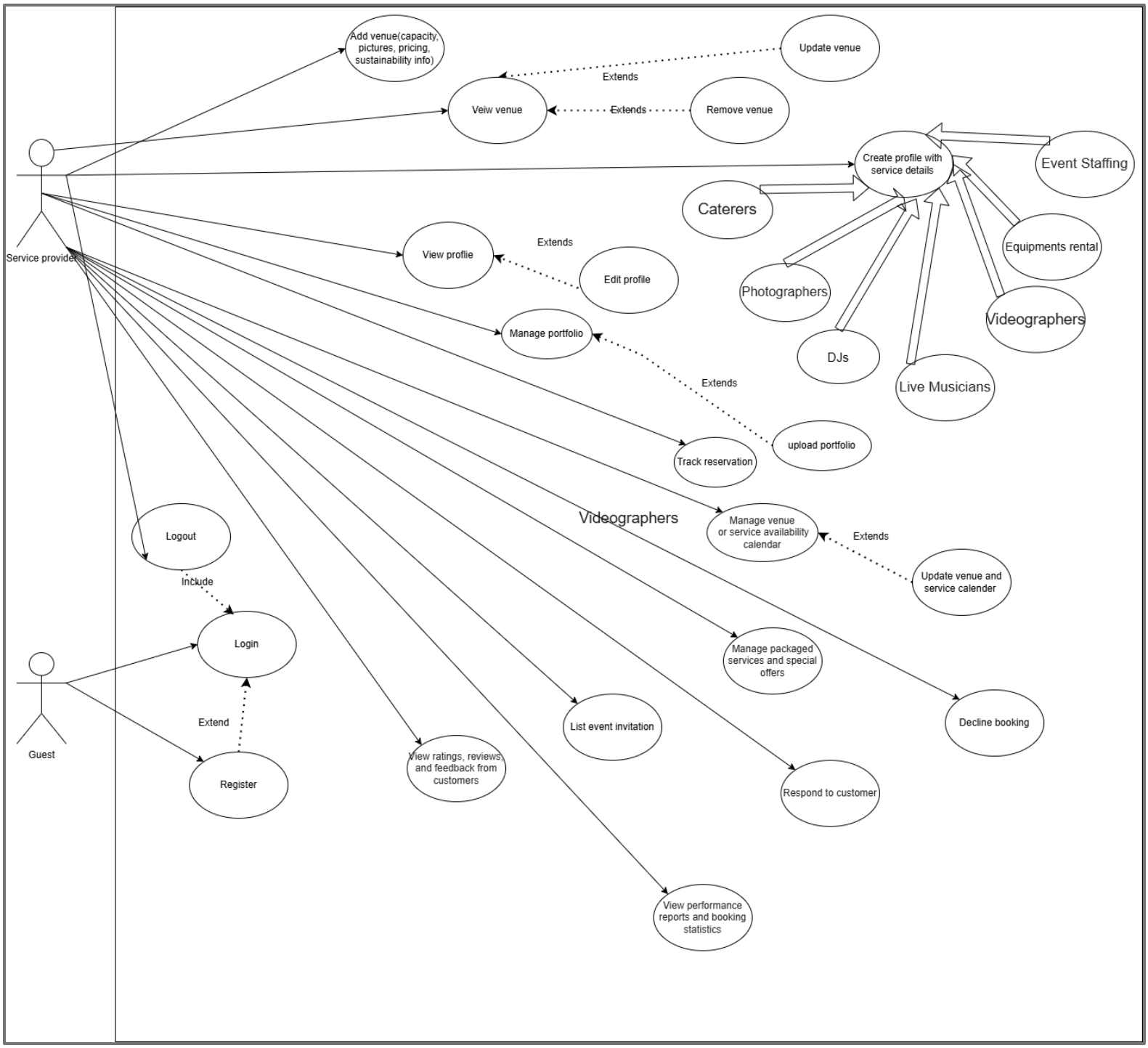


Figure 3: Service Provider Use Case Diagram

3.5. Sequence Diagrams

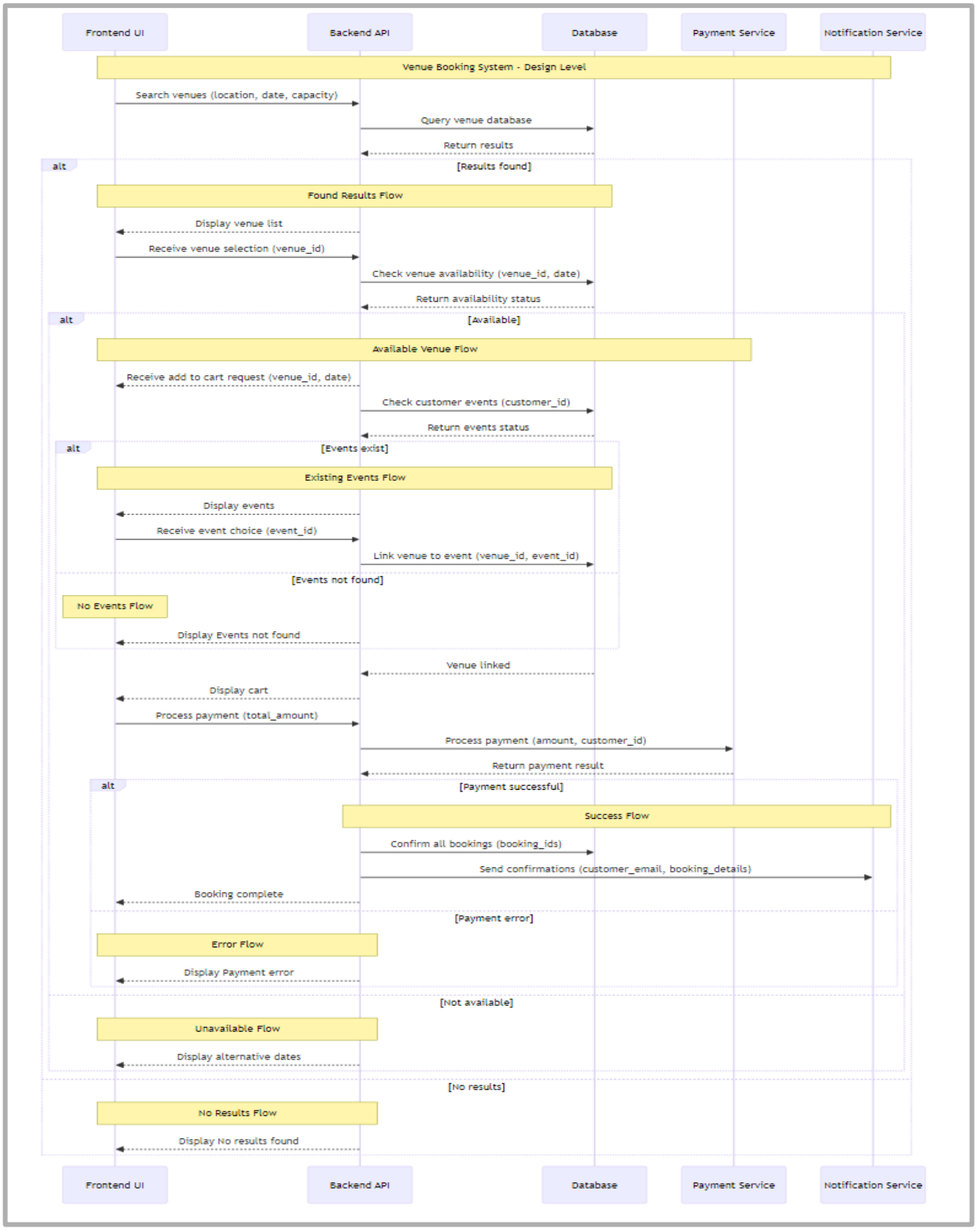


Figure 4: Book Venue _ Sequence Diagram

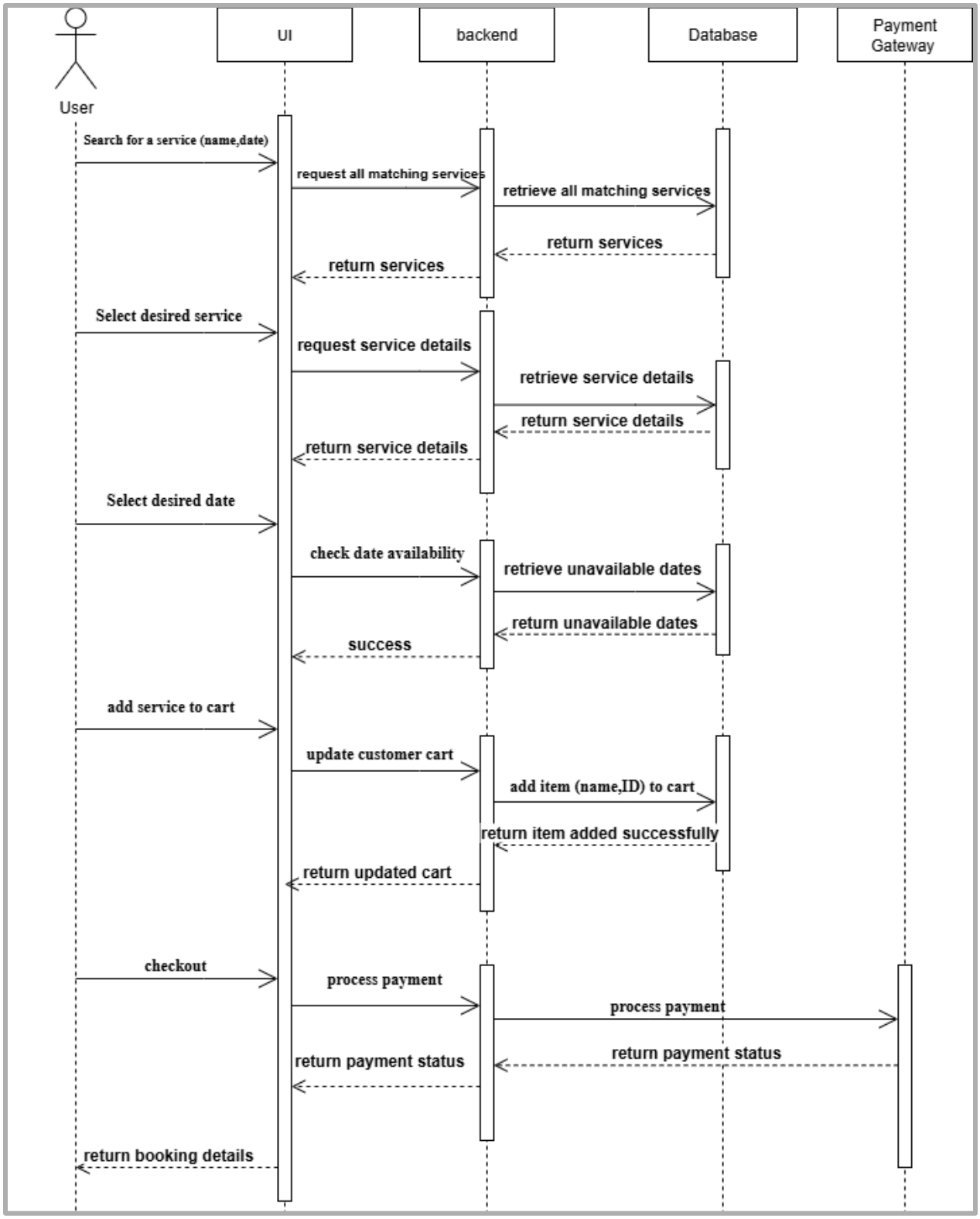


Figure 5: Book Additional Service _ Sequence Diagram

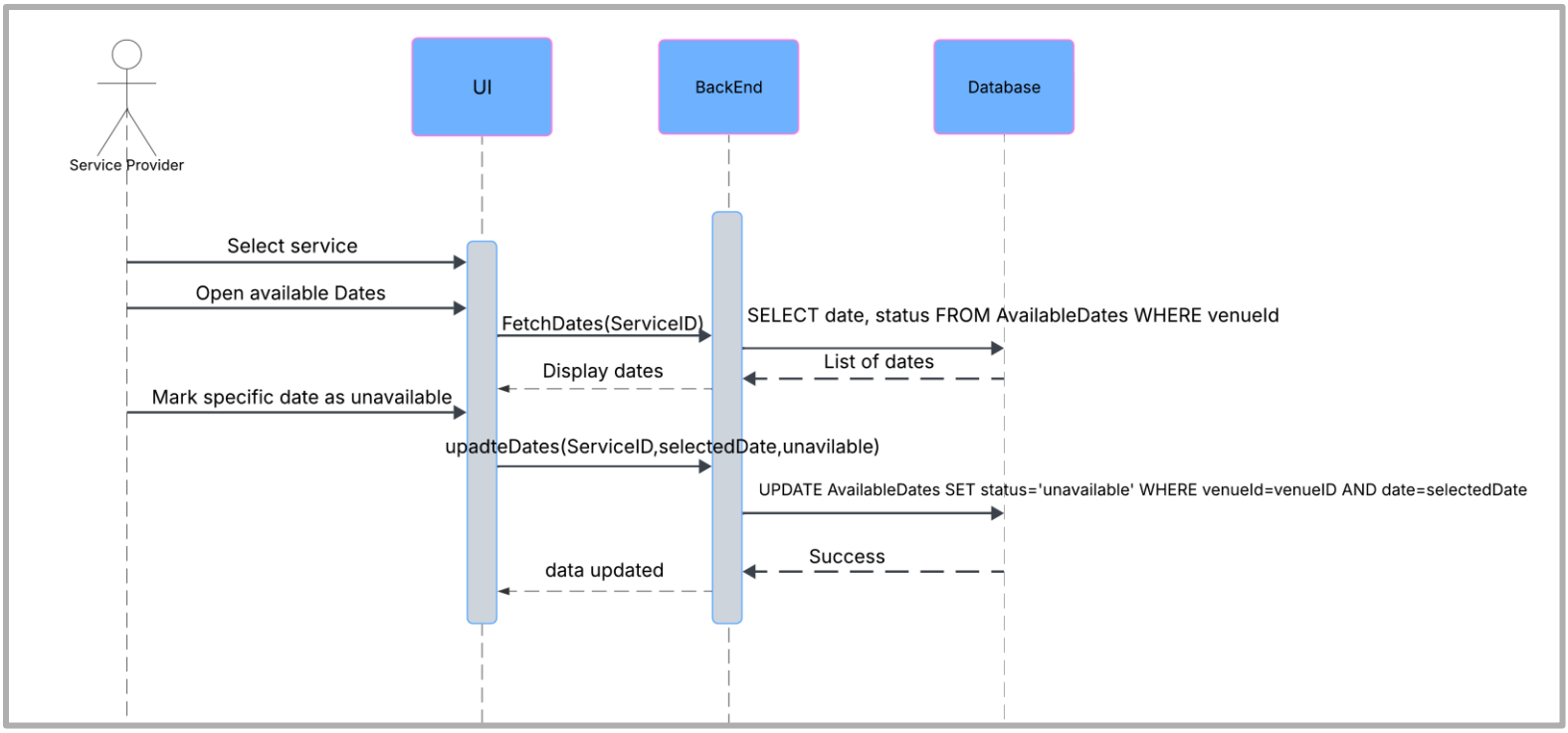


Figure 6: Update available dates of existing service _ Sequence Diagram

3.6. Activity Diagrams

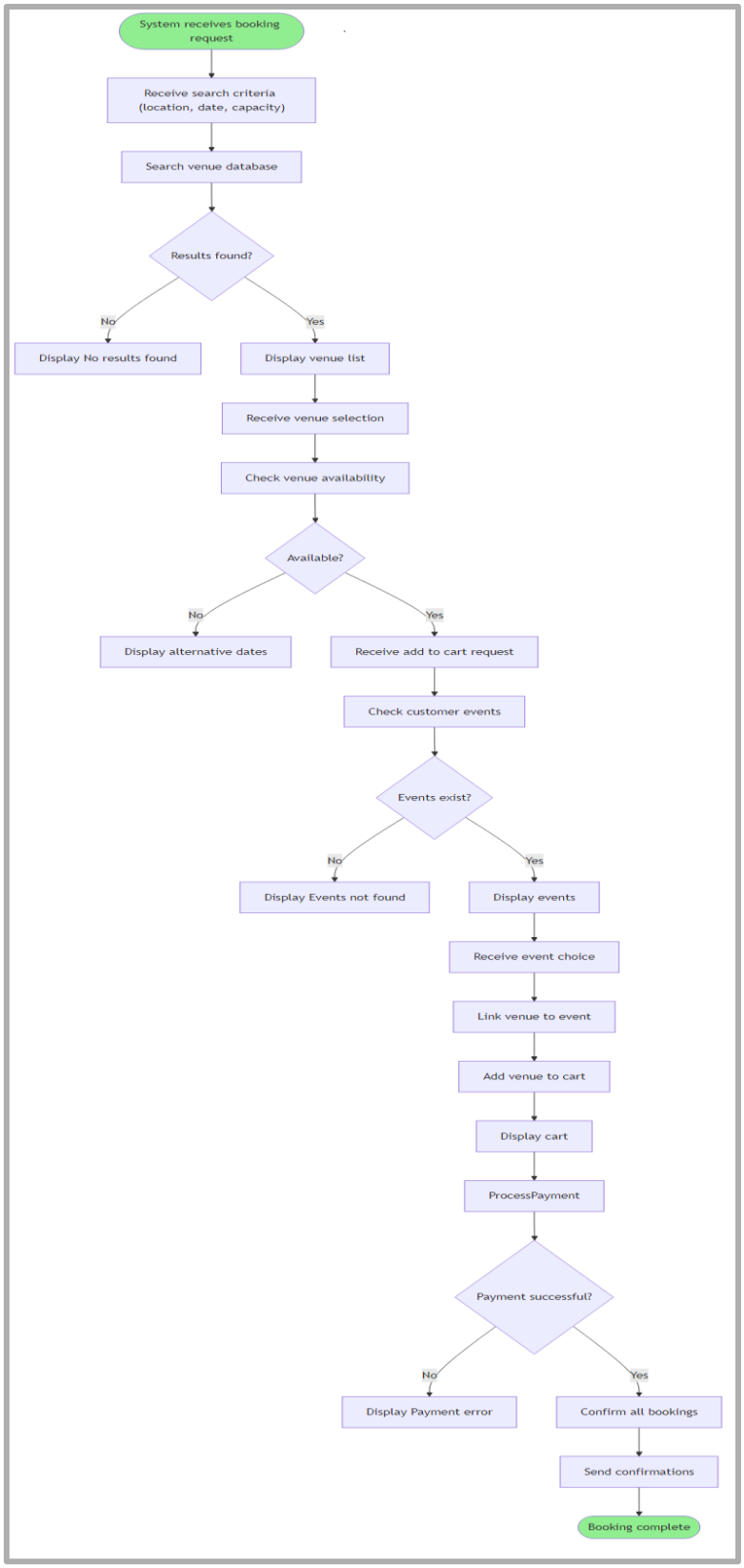


Figure 7: Book Venue Activity Diagram

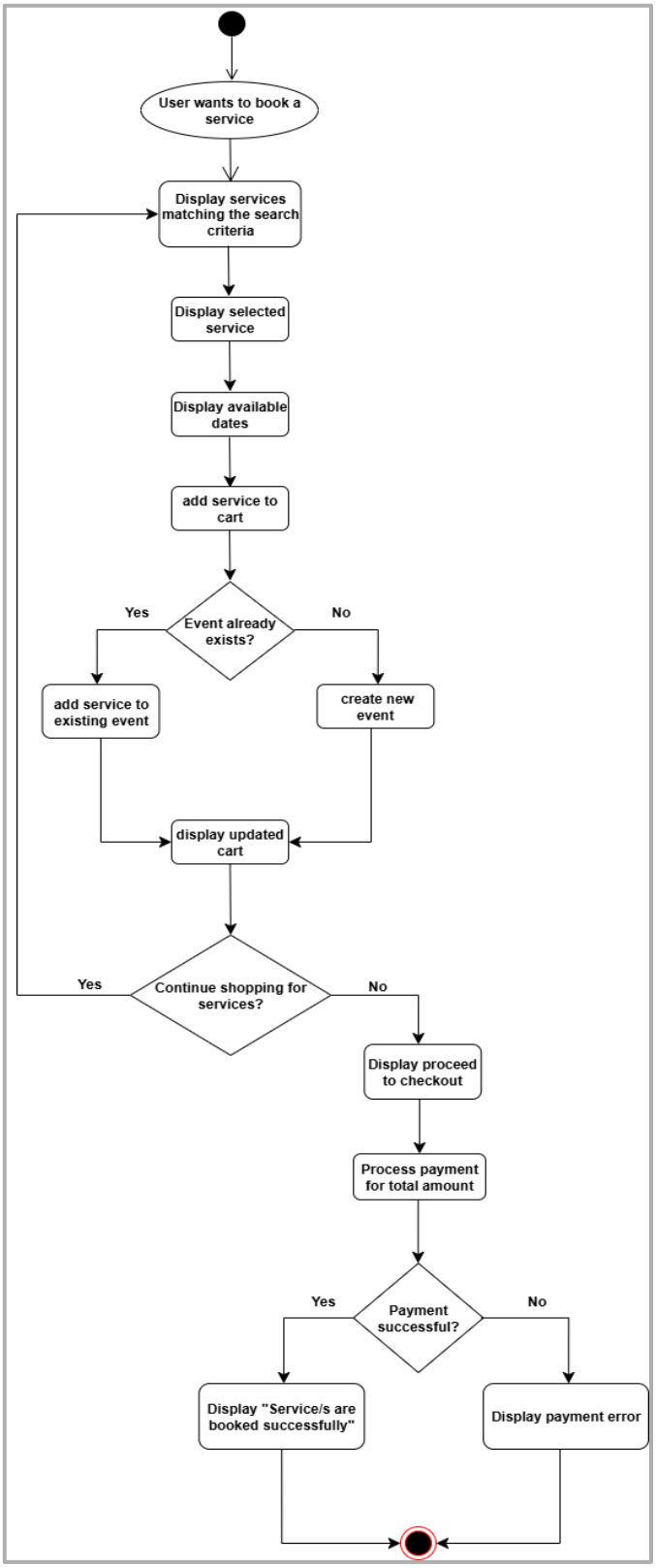


Figure 8: Book Additional Service Activity Diagram

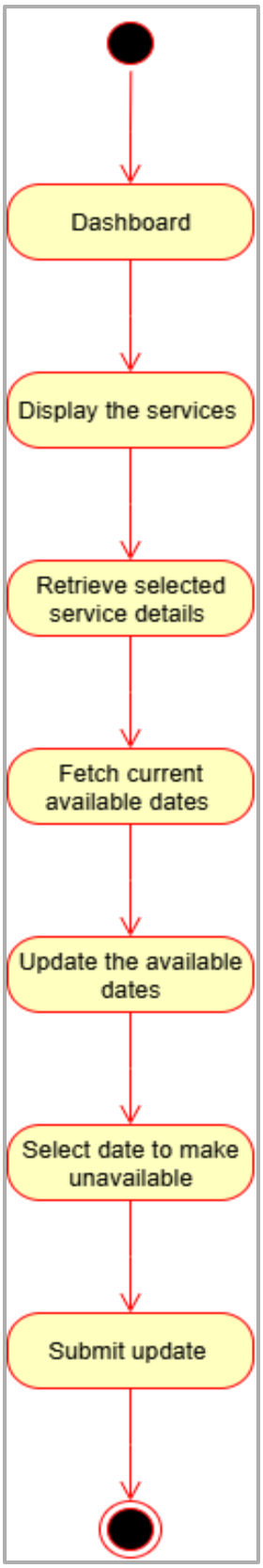


Figure 9: Update available dates Activity Diagram

3.7. Component Diagram

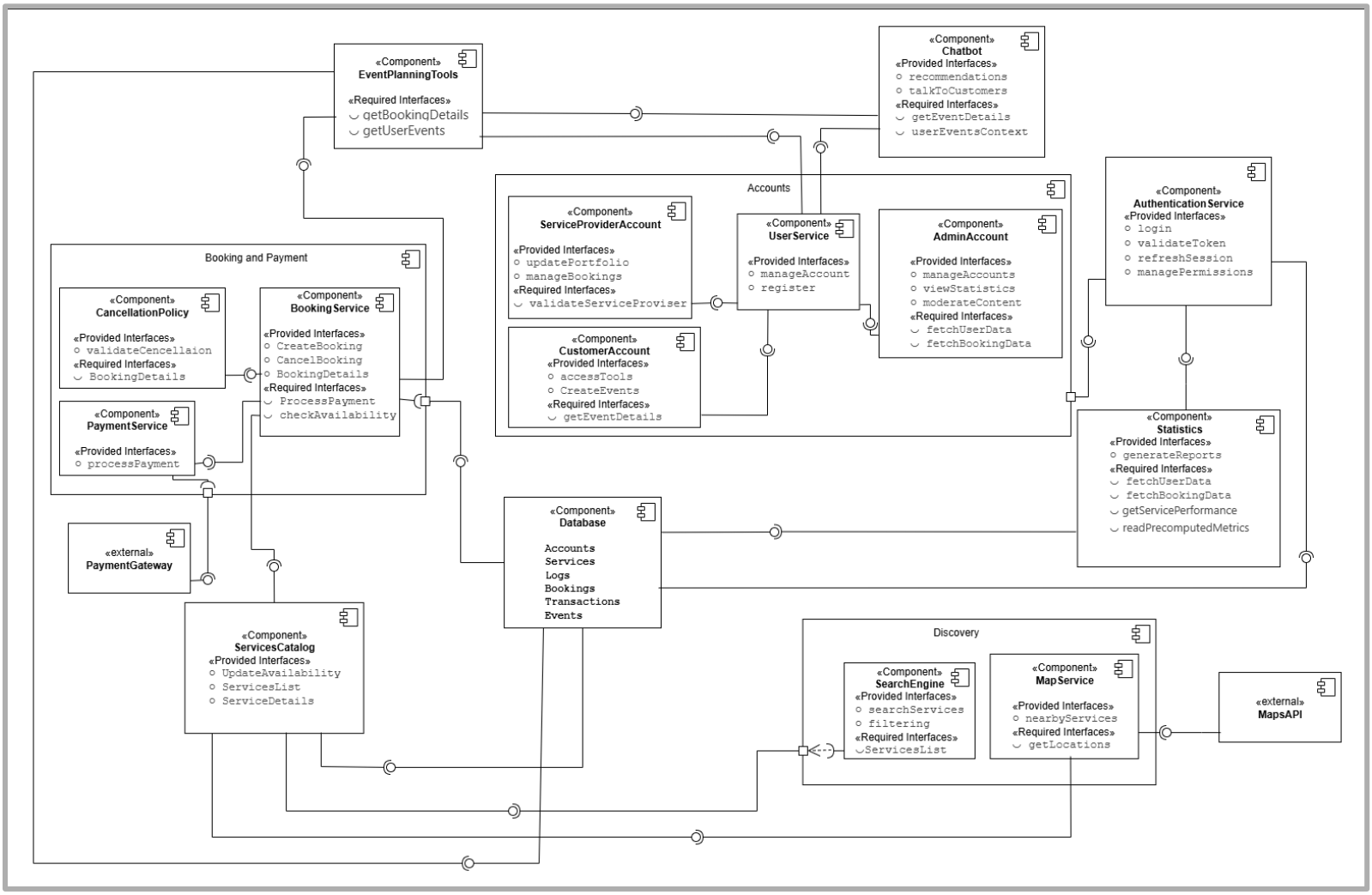


Figure 10: Component Diagram

3.8. Deployment Diagram

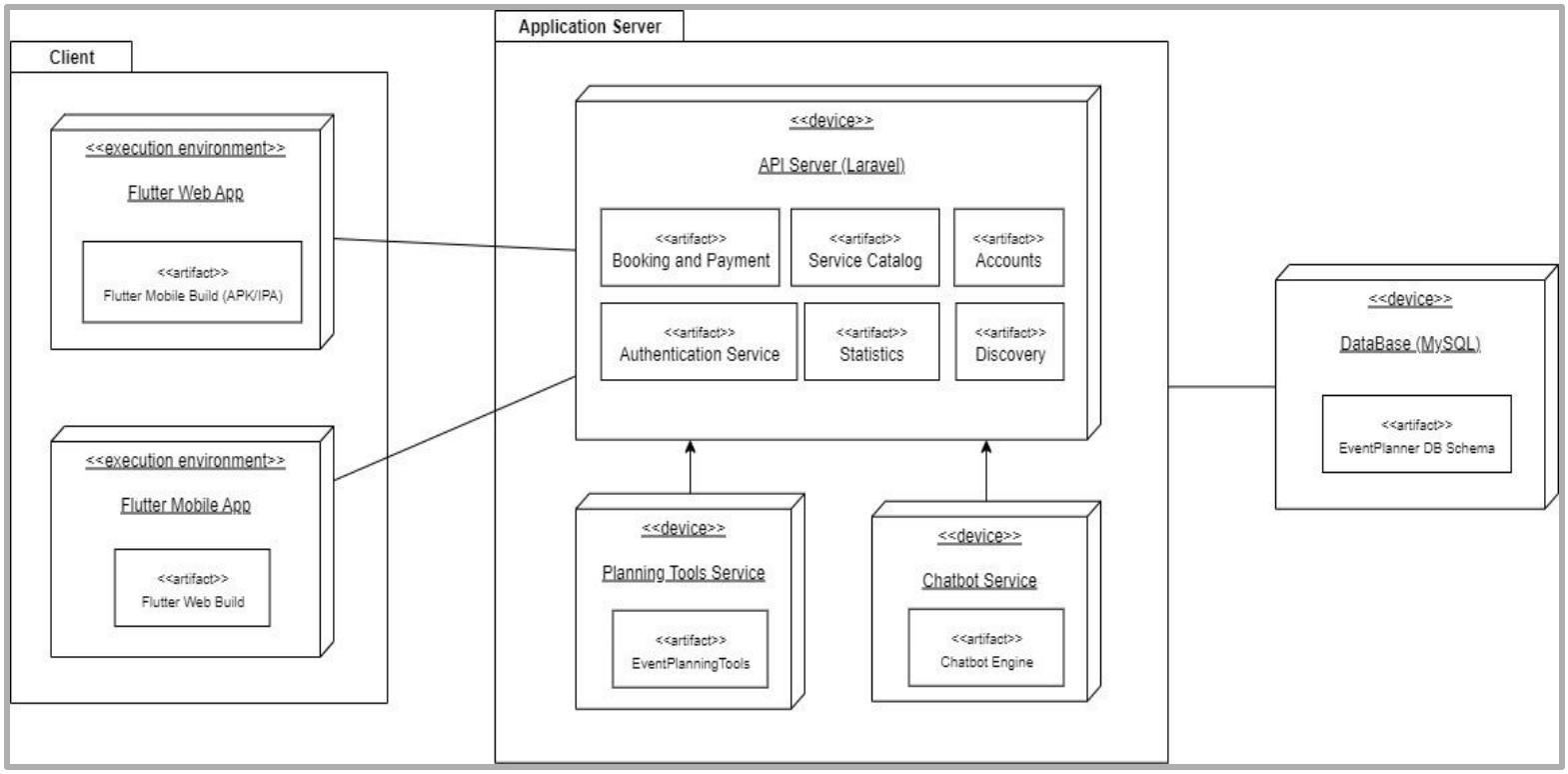


Figure 11: Deployment Diagram

Used Technologies:

- Frontend: Flutter mobile app (main client), web optional
- Backend: Laravel monolith (API, logic, orchestration)

3.9. ERD Diagram

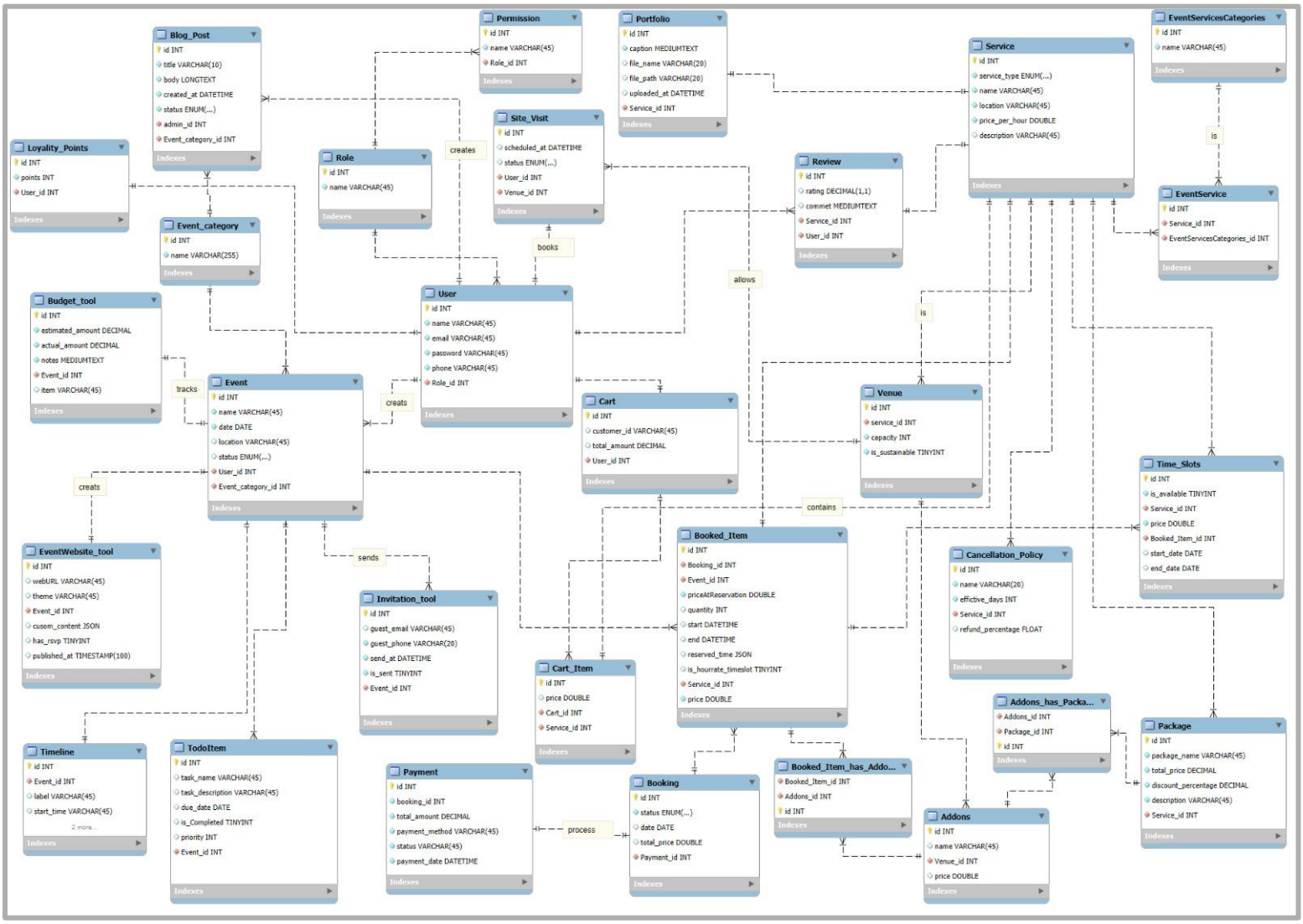


Figure 12: ERD Diagram

3.10. Class Diagram

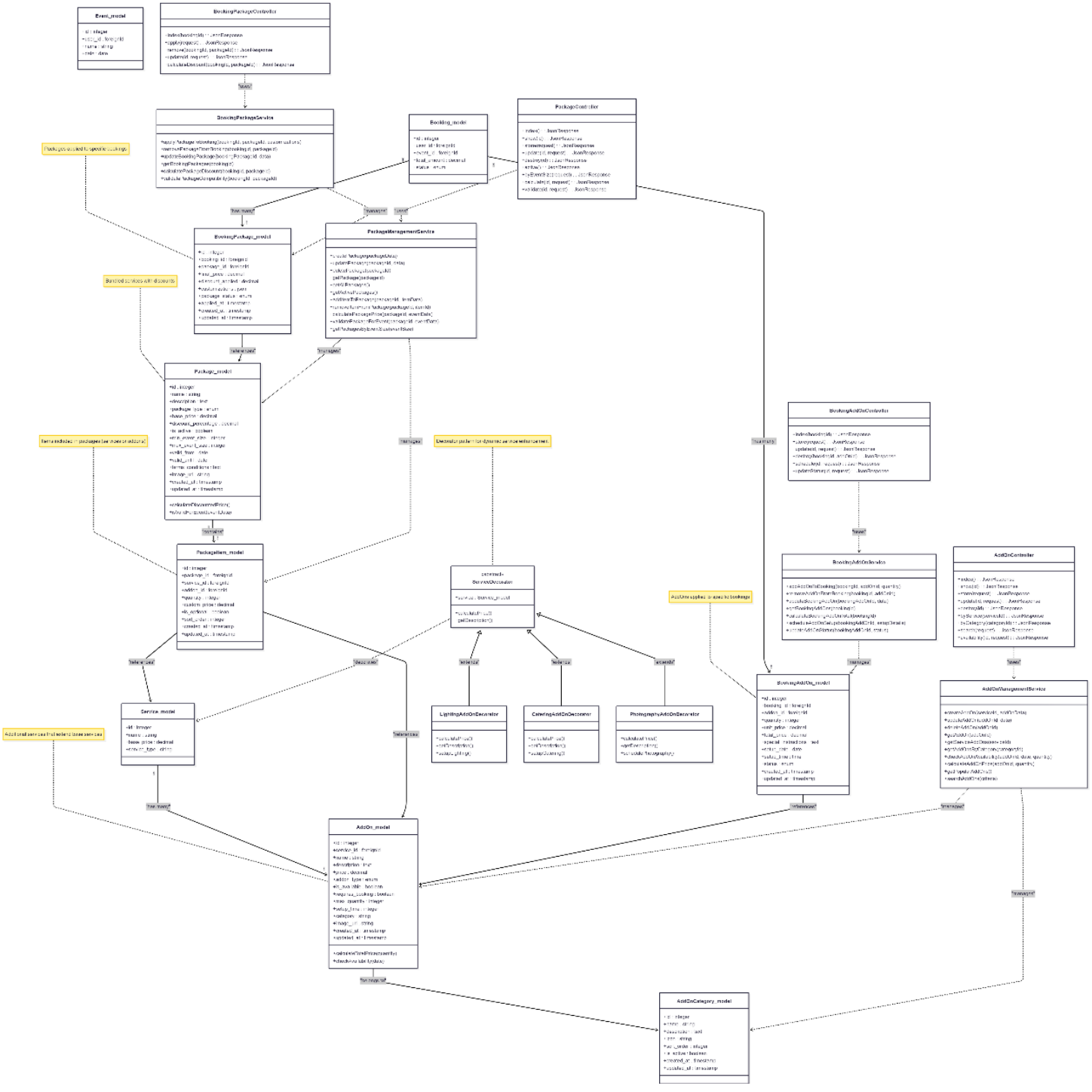


Figure 13: class diagram1 (addon package system)

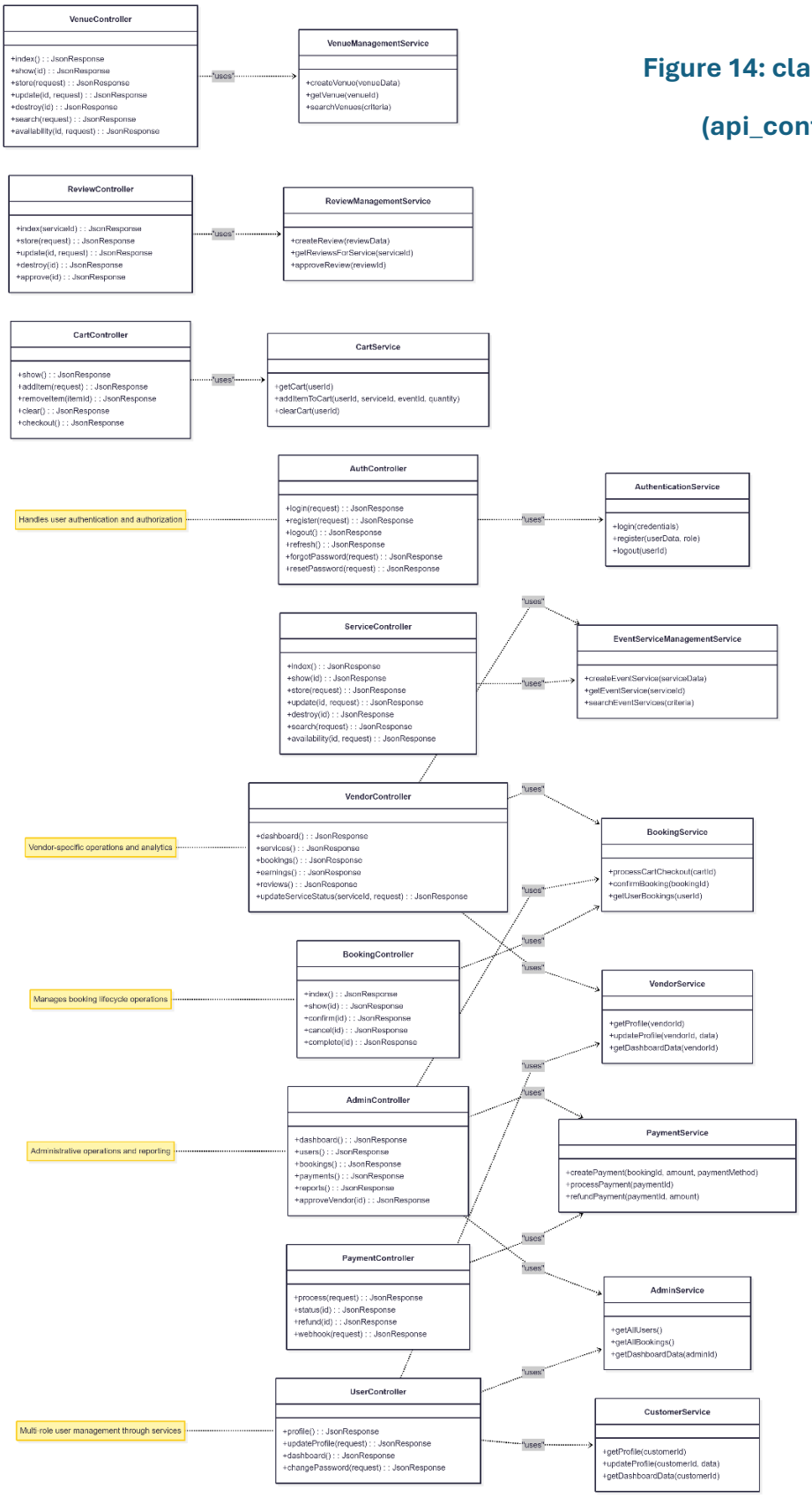


Figure 14: class diagram 2 (api_controllers)

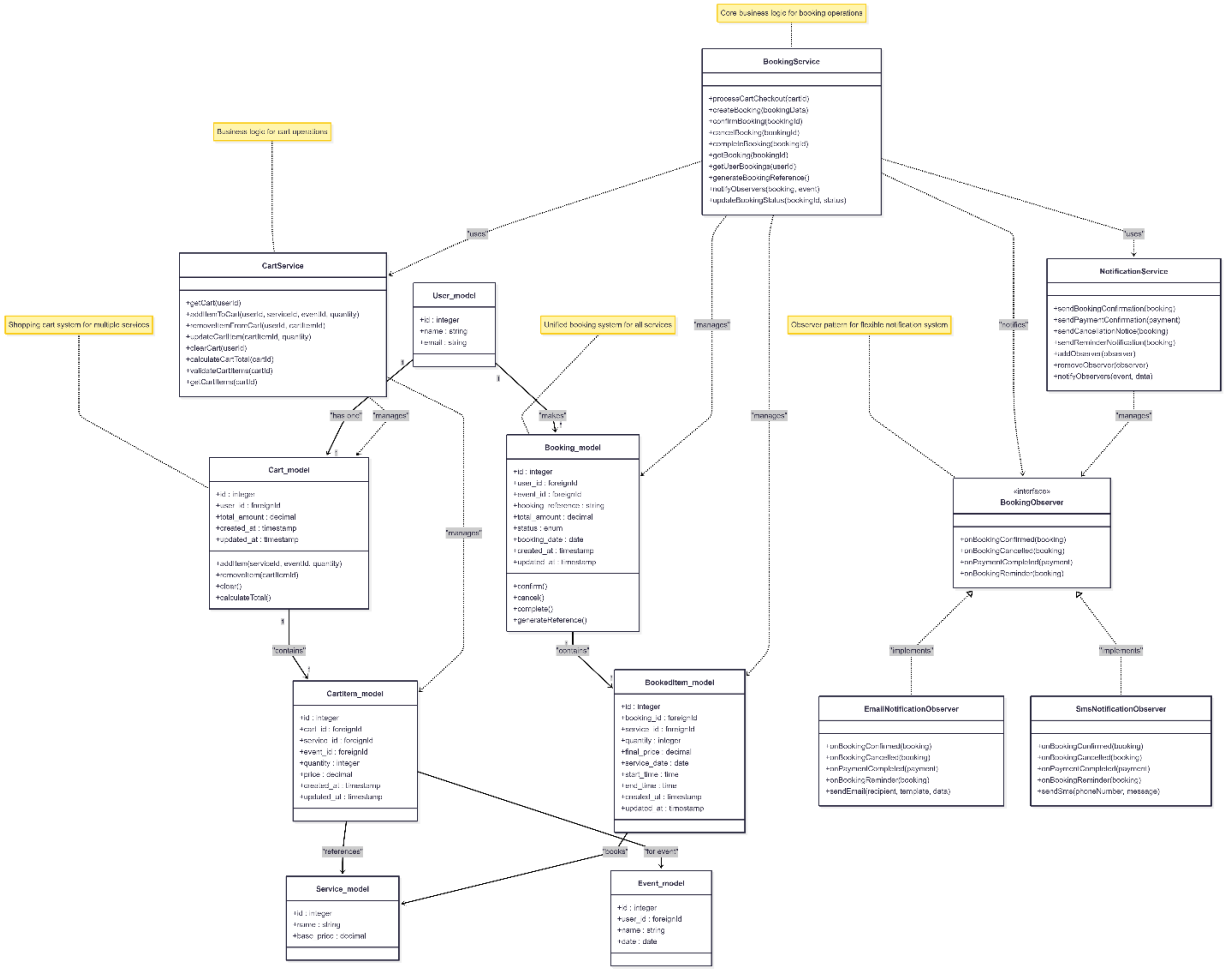


Figure 15: class diagram 3 (cart_booking_notifications)

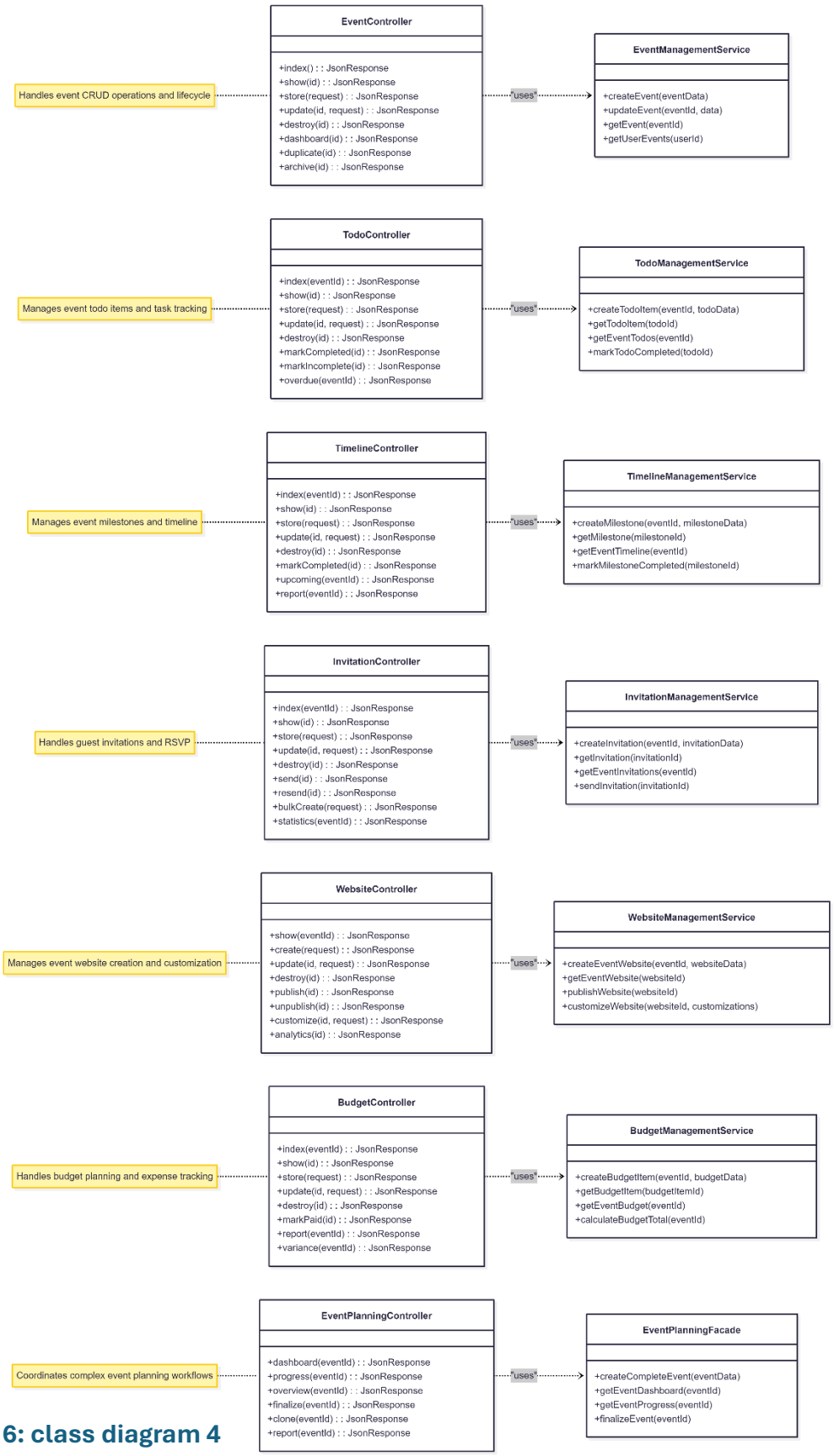


Figure 16: class diagram 4 (event_planning_controllers)

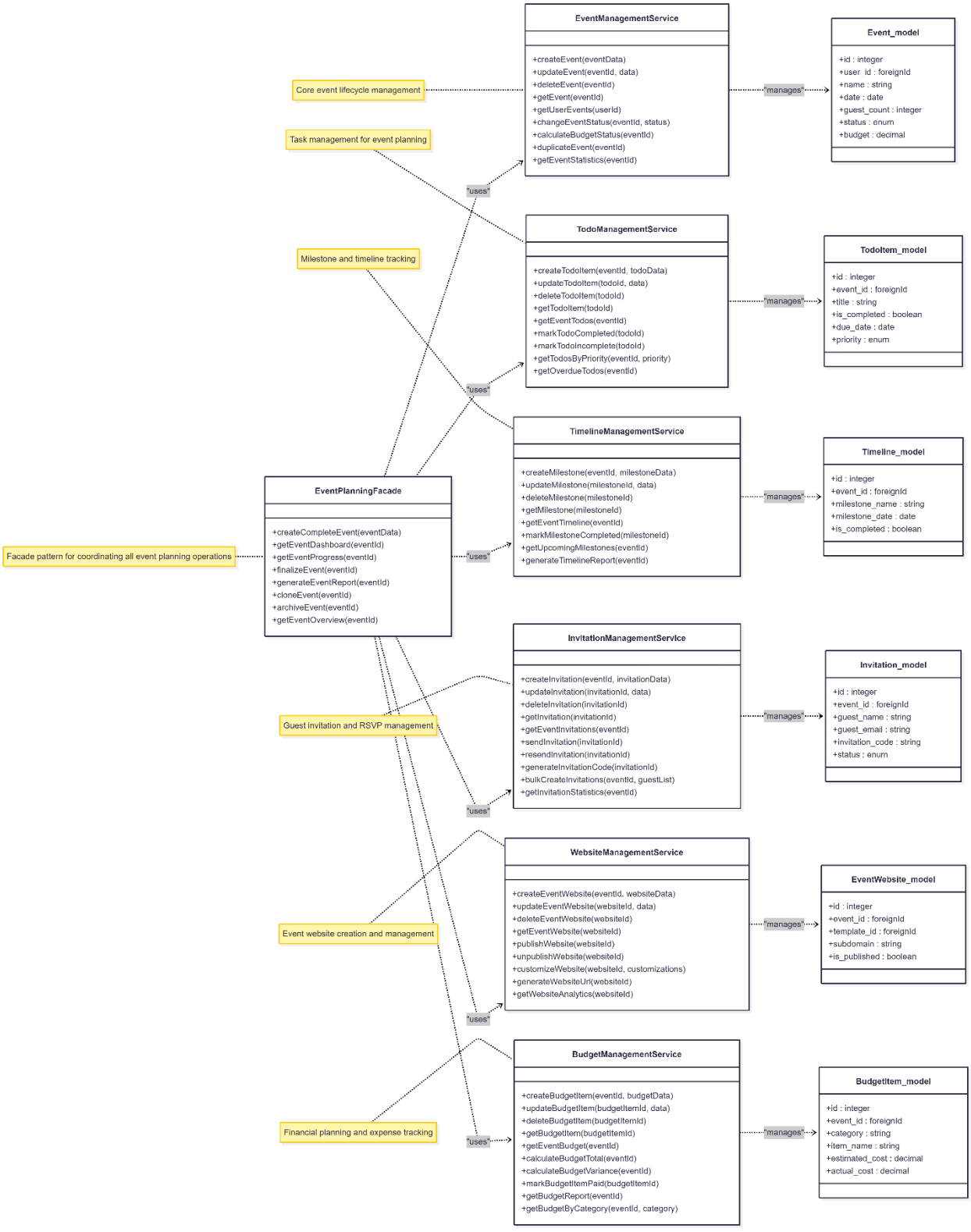


Figure 17: class diagram 5 (event_planning_services)

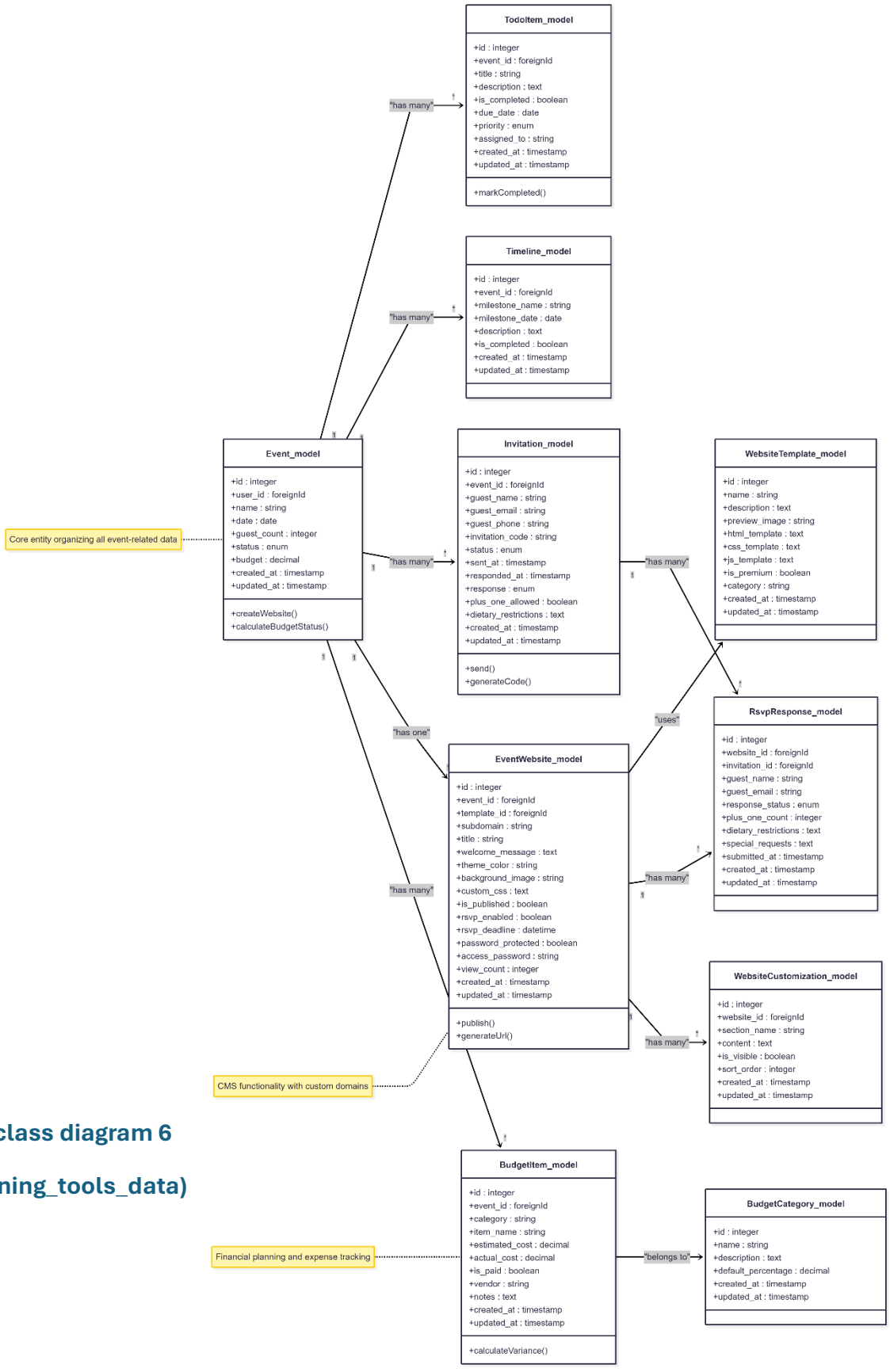


Figure 18: class diagram 6

(event_planning_tools_data)

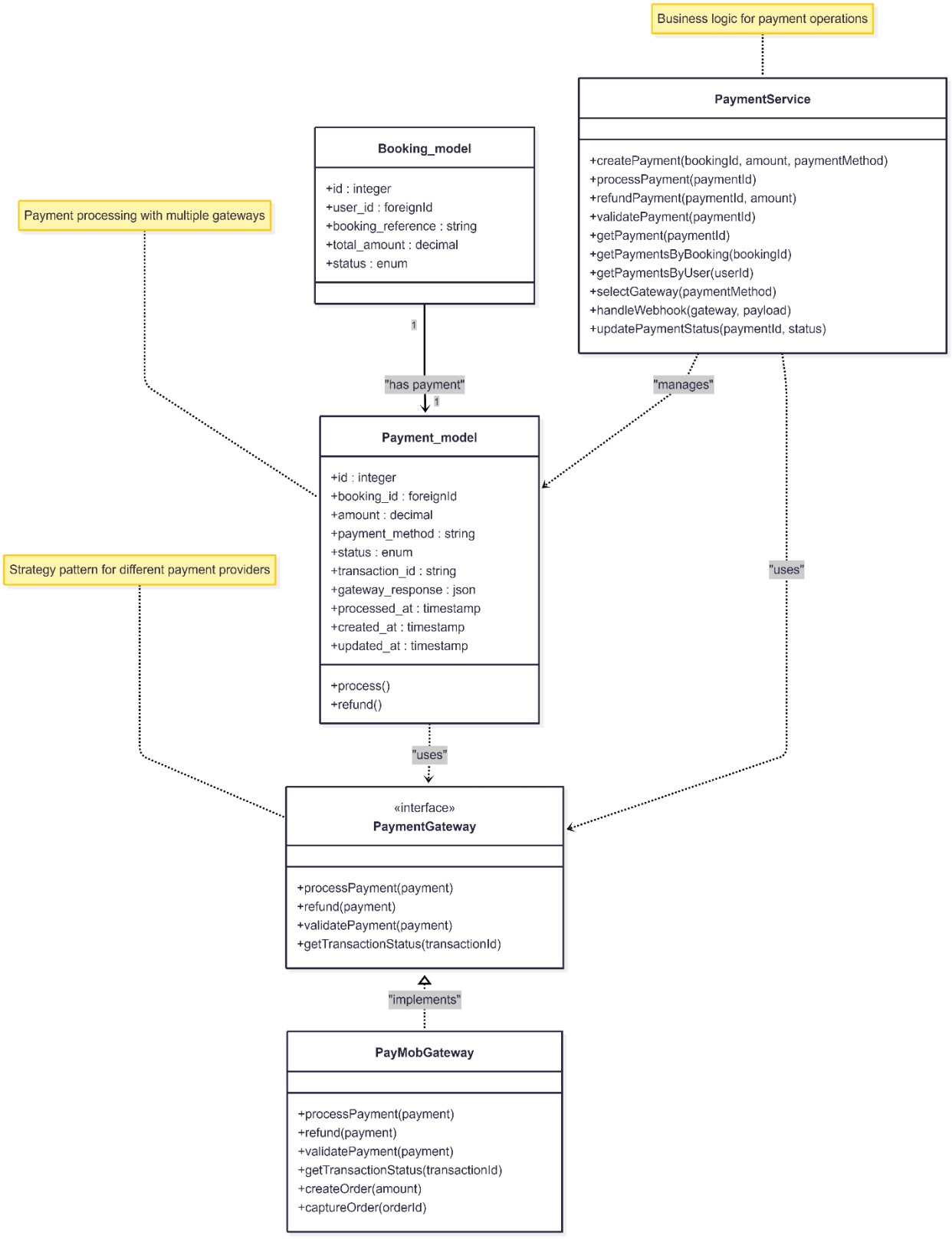


Figure 19: class diagram 7 (payment_system)

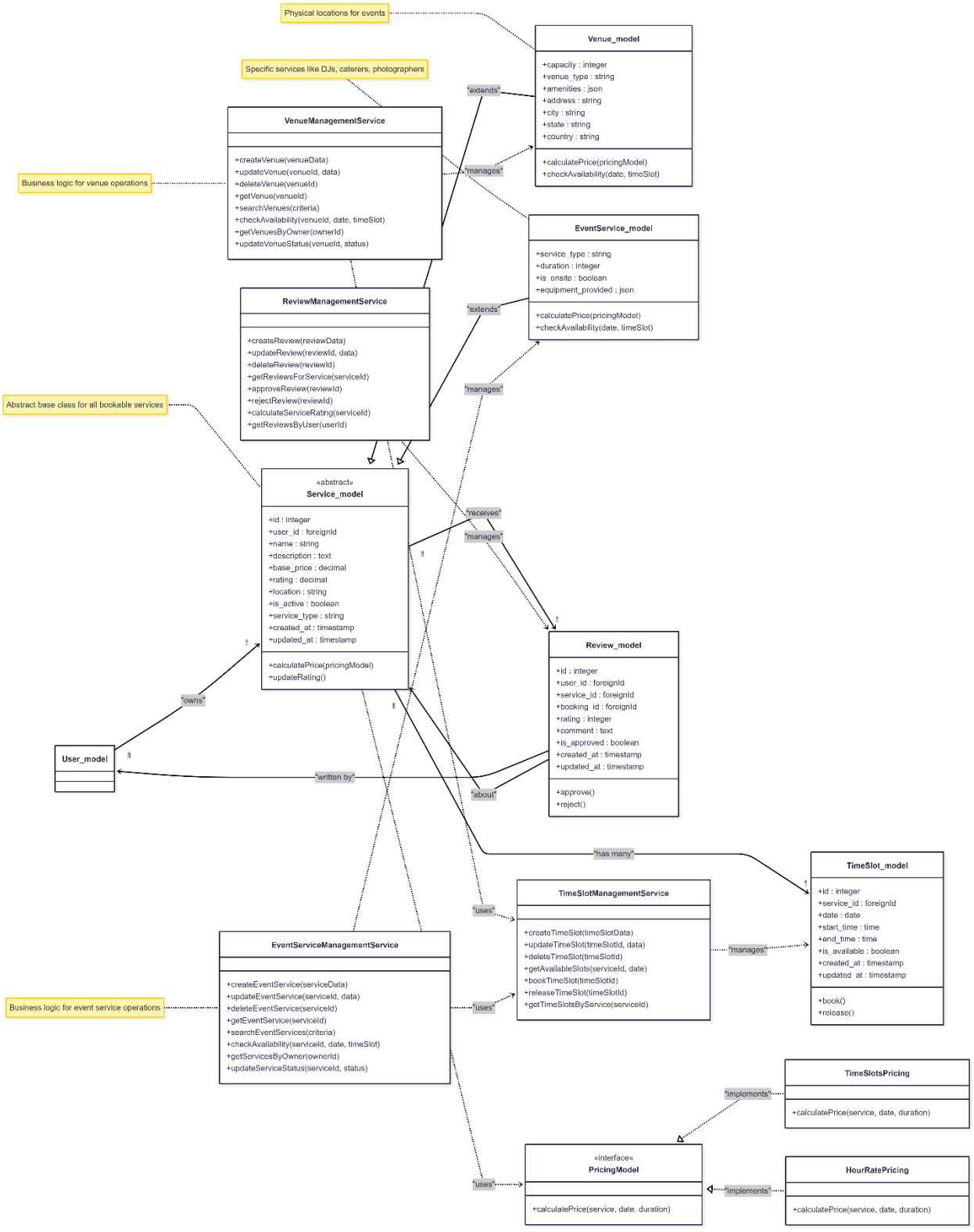
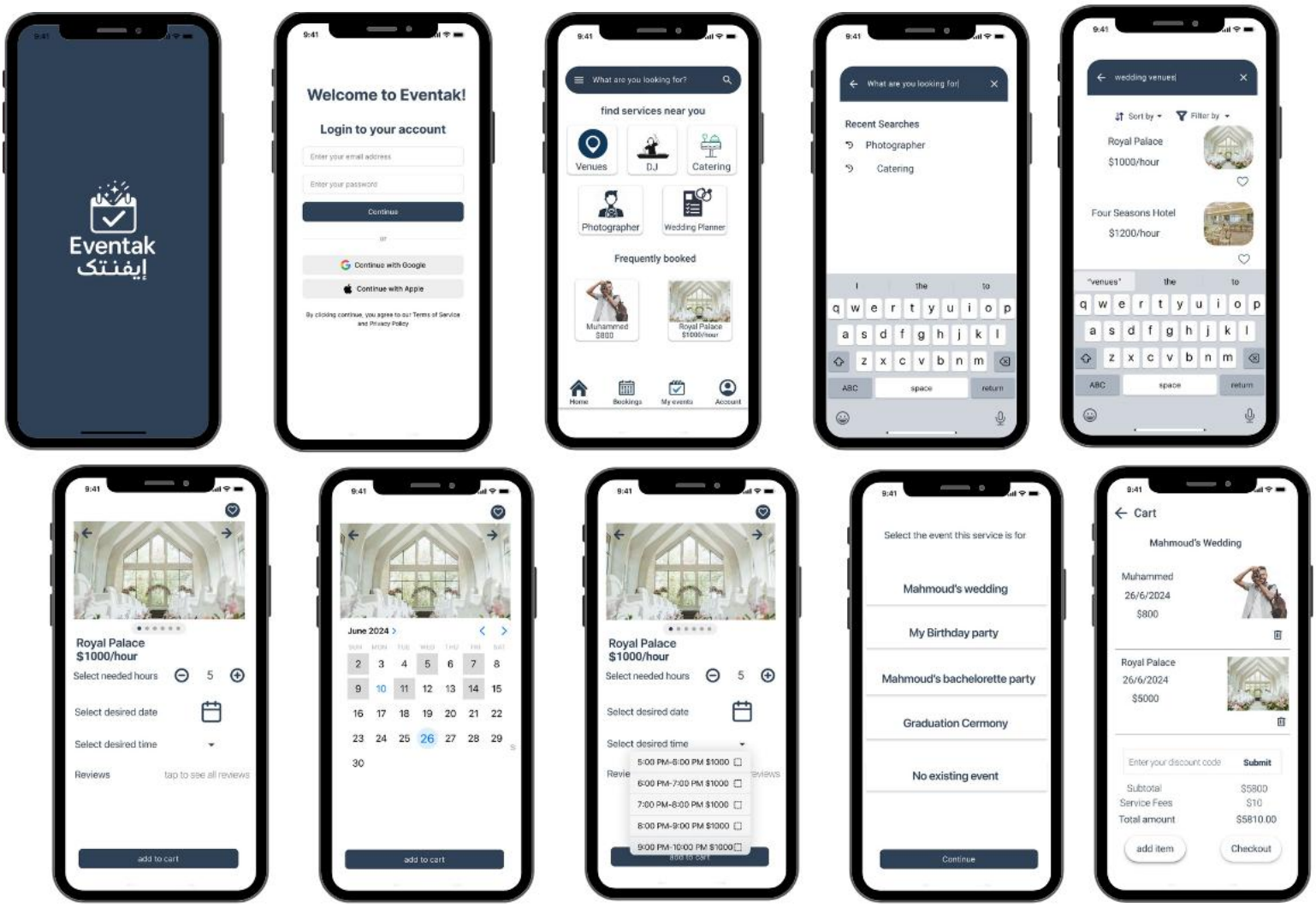


Figure 20: class diagram 8 (service_module)

Chapter 4: Prototype



Chapter 5: Work Plan

Task	Task Title	Description	Task status
1	Idea Selection	Brainstorming and Collecting ideas for the project.	Completed
2	Search for Related Works	Collecting data for the competitors and doing the market research	Completed
3	Define Functional Requirements	Focusing on the main features that the system will provide	Completed
4	Creating Diagrams	Creating UML diagrams and ERD to have better overview about the system flow	Completed
5	Design Database	Creating ER diagrams	Completed
6	Prototype	Making visual vision of the application	In Progress
7	First Documentation	Collecting all the diagrams and handling the documents	Completed
8	Frontend Development	Planned to be in Flutter	Not Started
9	Backend Development	Planned to be in Laravel	Not Started
10	Testing	Conducting unit and integration tests.	Not Started
11	Final Documentation	Writing final report and code documentation.	Not Started