**Software Requirements Specification (SRS) – Wedding Planner**

**1. Introduction**

**1.1 Purpose**

At Wedding Planner, our mission is to transform your wedding dreams into an extraordinary reality.

"We do more than just plan weddings: - we tell stories, create unforgettable moments, and protect your special love story.

**1.2 Scope**

Our goal is simple but meaningful:- To create a wedding experience that mirrors your uniqueness, celebrates your love, and goes beyond all your expectations.

**1.3 Definitions, Acronyms, and Abbreviations**

* **Client/User:** End-user booking and managing weddings.
* **Admin:** System administrator with full access.
* **Organizer:** Manages wedding events and logistics.
* **Booking:** Reservation of a wedding event/package.
* **DFD:** Data Flow Diagram.
* **ERD:** Entity-Relationship Diagram.

**2. Overall Description**

**2.1 Product Perspective**

The Wedding Planner is a web-based application, modular in design, with clear separation between user management, booking, event management, and administration.

**2.2 Product Functions**

* User registration, login, and profile management
* Wedding booking with category selection
* Event, and gallery management for each booking
* Post-wedding activities
* Admin and organizer dashboards

**2.3 User Classes and Characteristics**

* **Client/User:** Books and manages their weddings.
* **Admin:** Manages all users, bookings, categories,

**2.4 Operating Environment**

* Web browser (Chrome)
* Java backend (Servlets, Hibernate)
* MySQL or compatible RDBMS

**2.5 Design and Implementation Constraints**

* Follows MVC architecture
* Uses Hibernate ORM for database access
* JSP for frontend

**2.6 Assumptions and Dependencies**

* Users have internet access
* Admins and organizers are assigned by the system

**3. System Features**

**3.1 User Management**

* Register, login, logout
* Profile management
* Access control (admin, client)

**3.2 Wedding Booking**

* Book a wedding with details (bride, groom, date, category)
* View and manage bookings

**3.3 Event Management**

* Add, edit, delete events for a booking

**3.4 Gallery Management**

* Upload and manage images for a booking

**3.5 Post-Wedding Activities**

* Manage post-wedding events and information

**3.6 Admin/Organizer Features**

* Manage users, bookings, categories, and events

**4. External Interface Requirements**

**4.1 User Interfaces**

* Web-based UI (JSP, HTML, CSS, JS)
* Admin and organizer dashboards

**4.2 Hardware Interfaces**

* Standard PC

**4.3 Software Interfaces**

* Java, Hibernate, MySQL

**6. Data Model Overview**

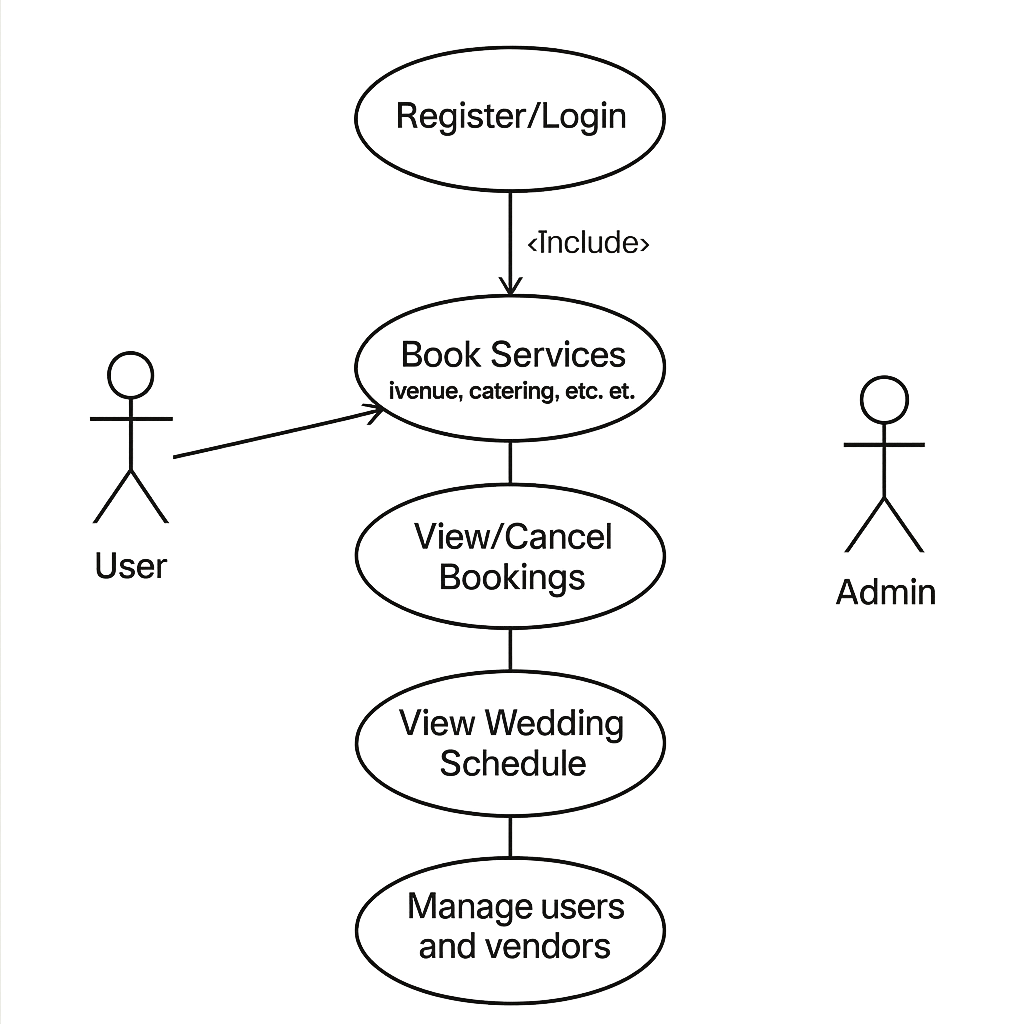
**Main Entities:**

* User
* WeddingBook
* WeddingCategory
* Events
* Gallery
* PostWedding
* Organizer

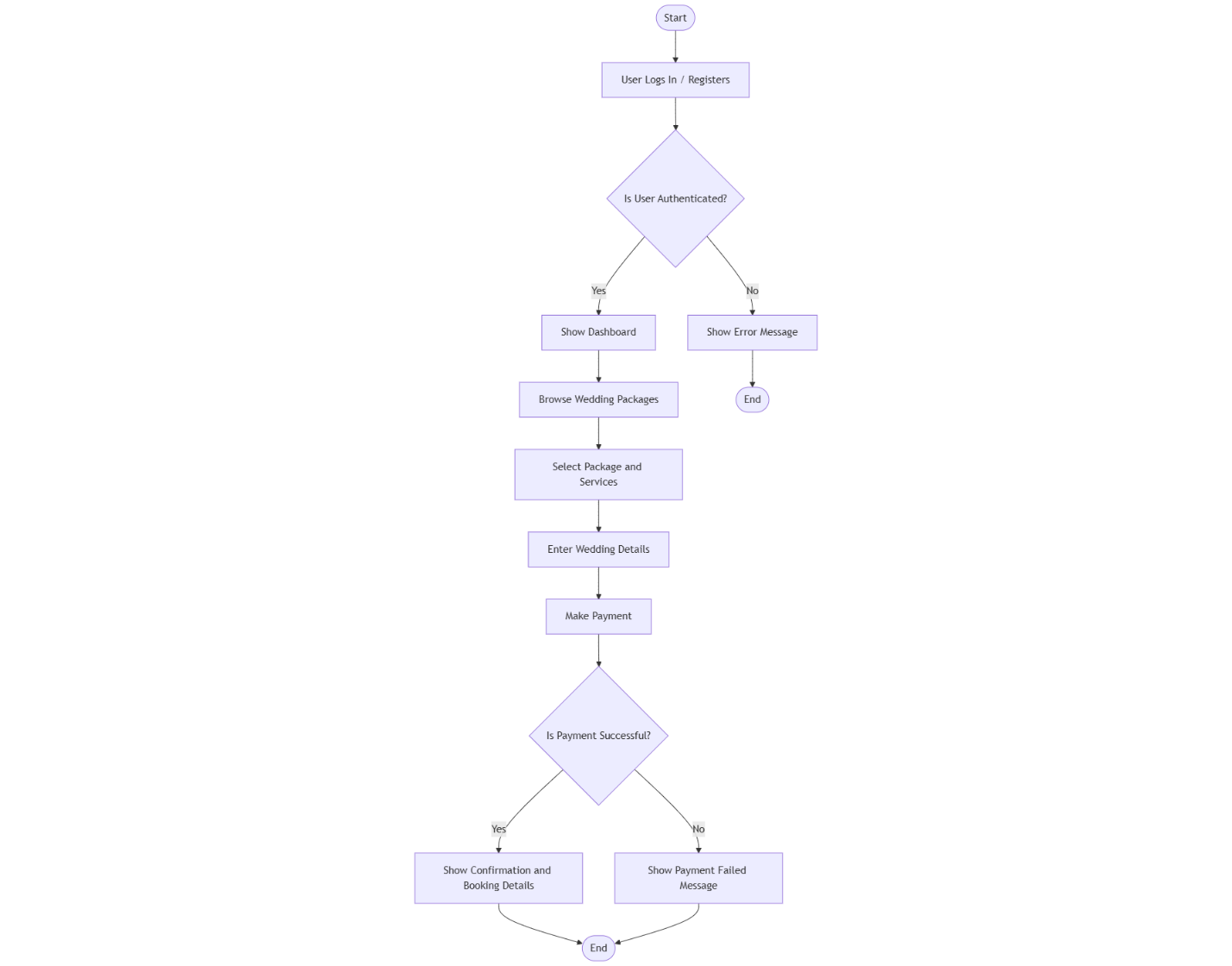
**7. Use Case Diagram**

### Use Cases:

* Register/Login
* Book Services (venue, catering, photography, etc.)
* View/Cancel Bookings
* View Wedding Schedule
* Admin manages users and vendors



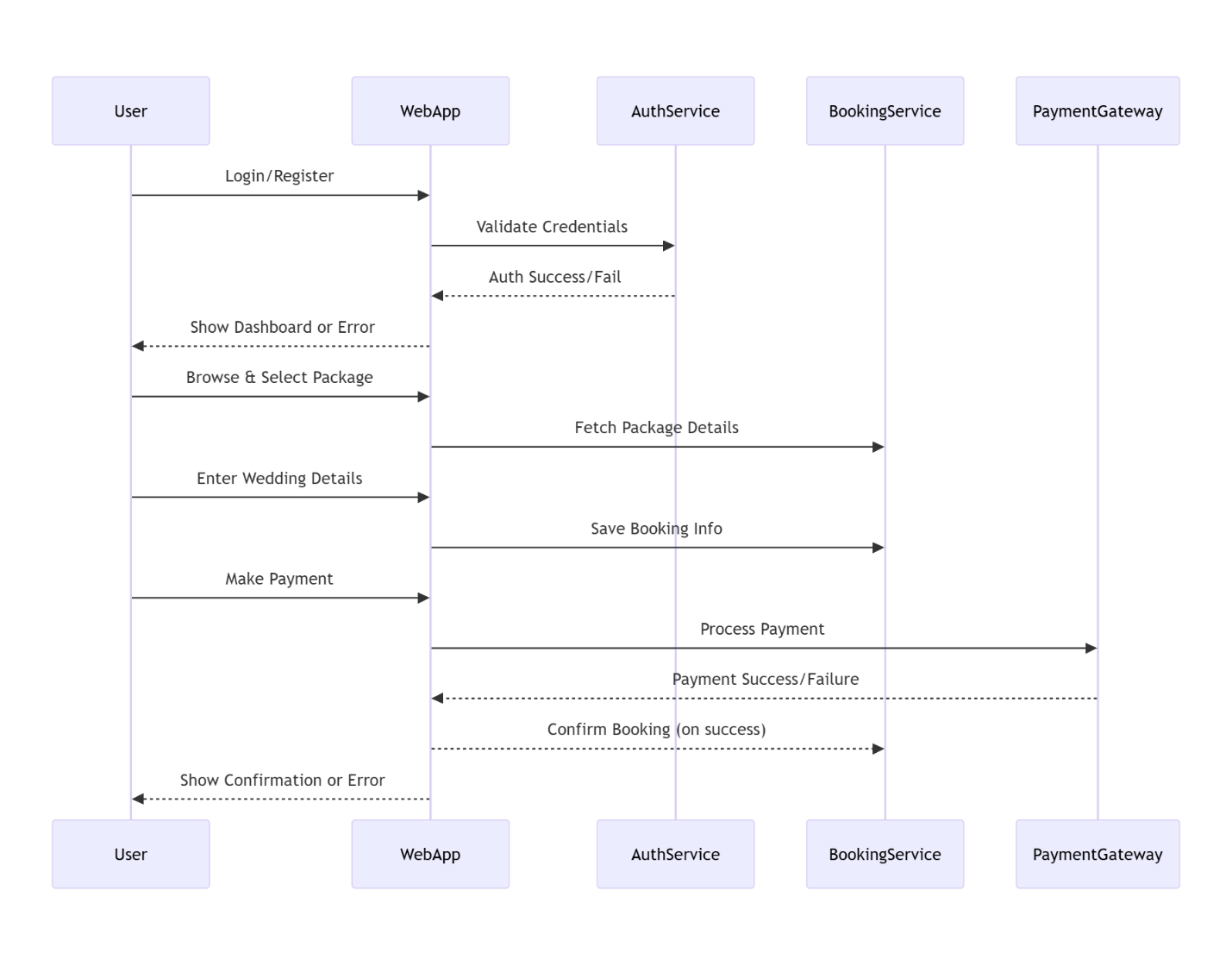
## ****Activity Diagram****



## ****State Machine Diagram****

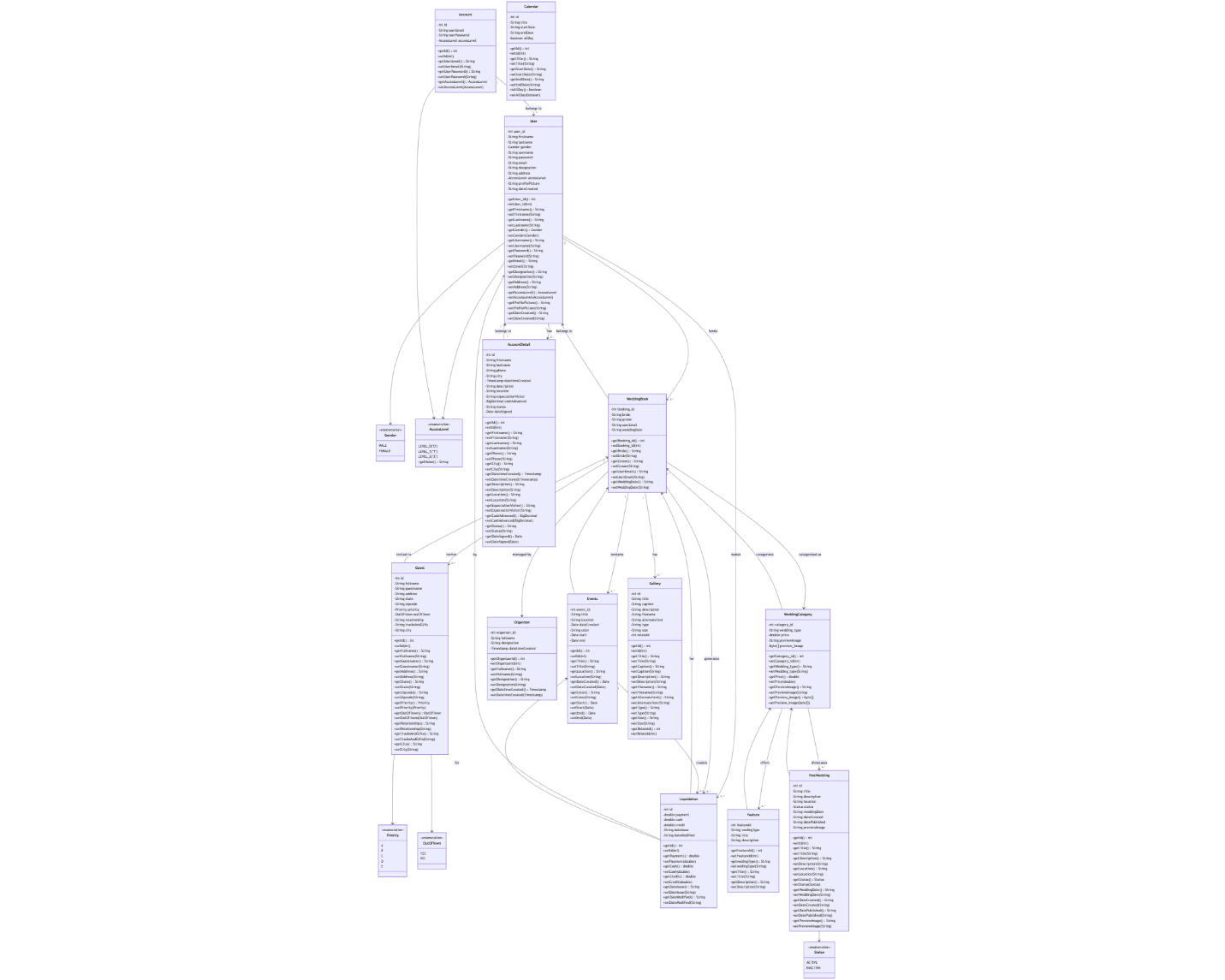
## 

## ****Sequence Diagram****



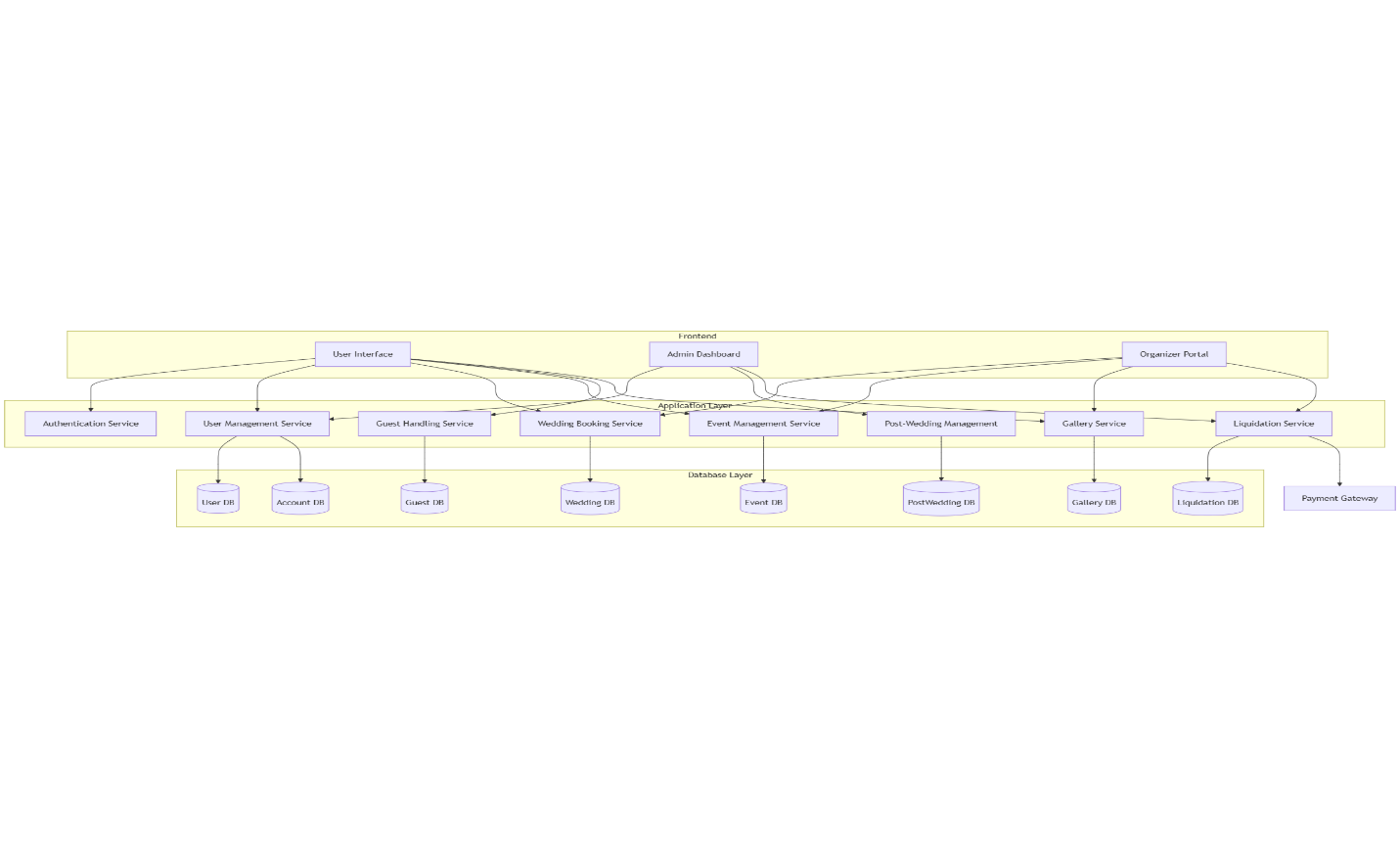
## ****Class Diagram****

Classes like:

* User
* Booking
* Payment
* Admin
* 

## ****Component Diagram****

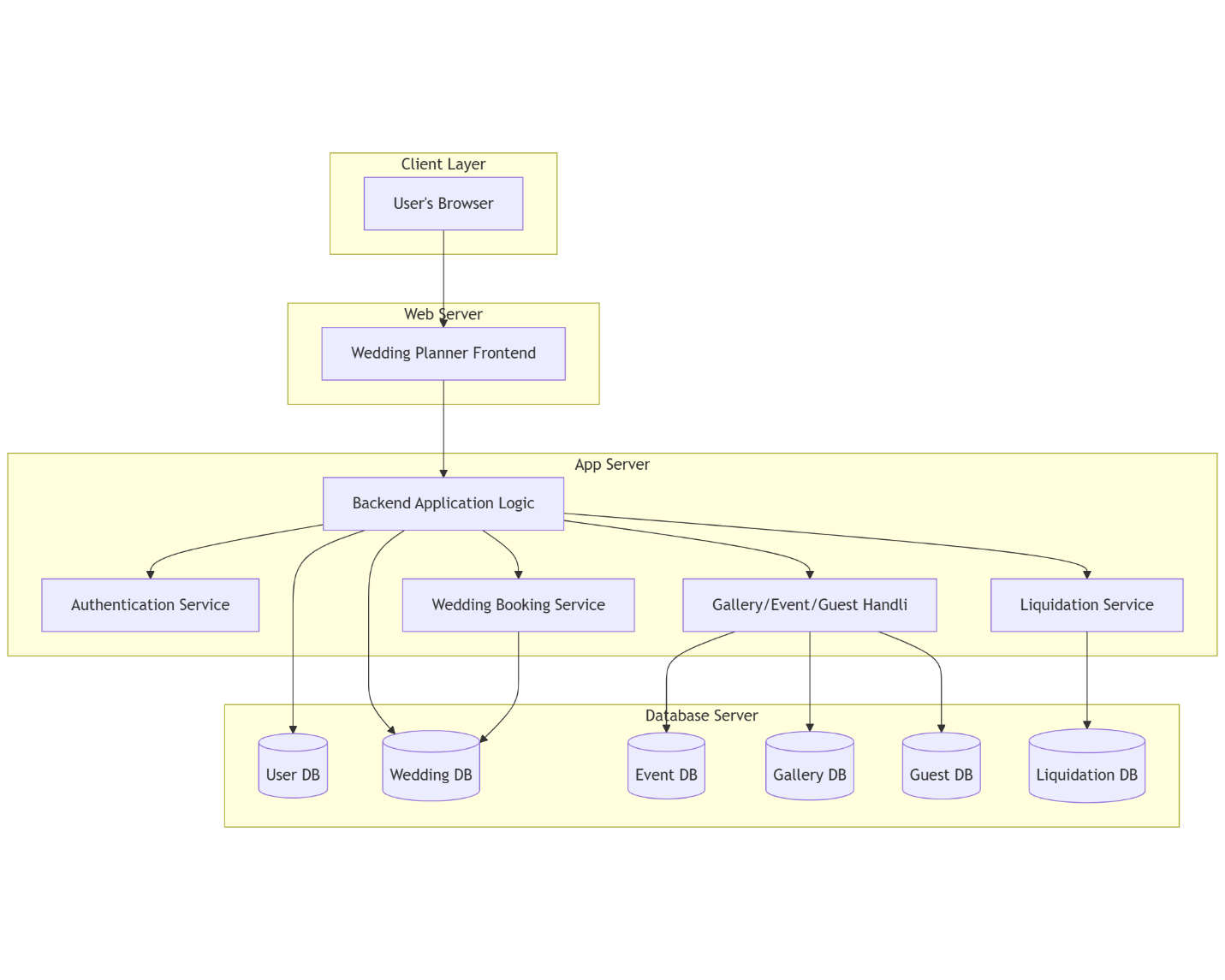
Shows how different modules like:

* User Interface
* Vendor Service
* Payment Service
* Database

## ****Deployment Diagram****

Shows deployment of:

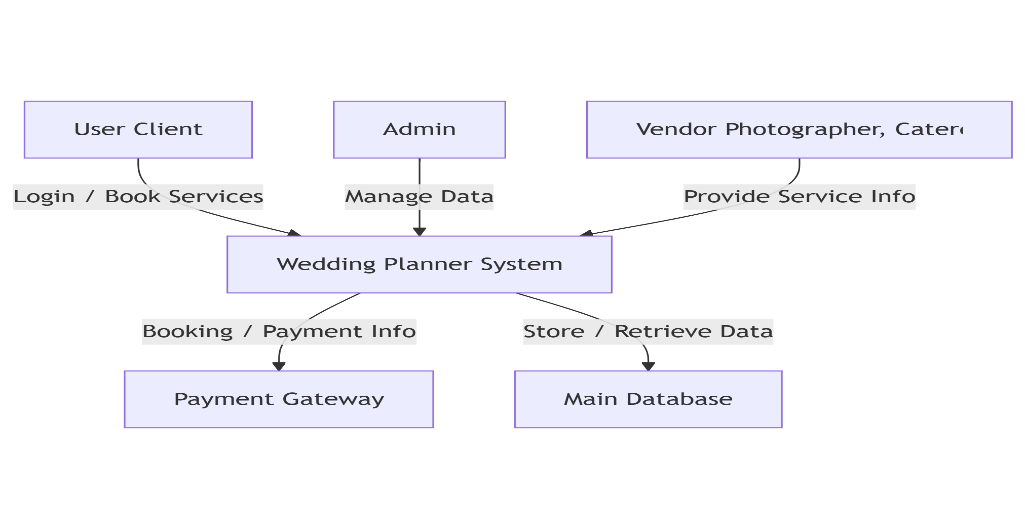
* Web Server
* App Server
* Database Server



## ****Data Flow Diagrams (DFD)****

You need **2 levels**:

* **Level 0 (Context Diagram)**: Overall system interaction



* **Level 1**: Break down major processes (e.g., vendor booking, payment)

