Project Title:

Venue Management System

Group Members:

1. Nowshin Nure Azad Shihab	ID: 2131330642
2. Sazzad Hossain Shawon	ID: 2211415042

Course Code: CSE 327 (Software Engineering)

Section: 02

Instructor: [AKM Iqtidar Newaz]

Description:

The **Venue Management System** is a C++-based software application designed to streamline the process of finding, managing, and booking venues for events such as weddings, corporate meetings, birthdays, and conferences. In the current manual system, customers rely on physical visits, phone calls, or word-of-mouth to find suitable venues, which often leads to time-consuming and inefficient experiences.

This project aims to solve these problems by creating a centralized and automated platform that enables users to **search, view, and book venues** according to their preferences (e.g., location, price, capacity, date). It will also allow **venue owners** to list and manage their venues, update availability, and respond to booking requests. An **admin** role will be responsible for monitoring and controlling listings, users, and bookings to maintain system integrity.

The core goals of this system include:

- Reducing the effort of finding available venues.
- Providing a unified platform for venue owners and customers.
- Ensuring smooth booking operations with proper notifications.
- Applying real-world software engineering principles .

---

2. End Users of the Project

The system is designed for three distinct types of users:

A. Customers / Event Organizers

- Individuals or businesses planning events.
- Can browse available venues using filters.
- Can request and confirm bookings.

B. Venue Owners

- Owners of event halls, rooftops, convention centers, or meeting rooms.
- Can register their venues with relevant details.

- Can manage availability and respond to booking requests.

C. Admin

- Oversees the platform to ensure smooth operation.
- Approves or rejects new venue listings.
- Manages users, resolves conflicts, and handles misuse.

Each user role will have access to specific features based on their permissions, ensuring a secure and streamlined experience.

---

3. Functional Requirements (FRs)

Functional requirements define the core behavior and features of the system. These are directly tied to what the system must do.

FR Code	Requirement Description
FR1	The system shall allow users to register and log in based on role (Admin, Owner, Customer).
FR2	Customers shall be able to search venues using filters such as location, price, capacity, and availability.
FR3	Customers shall be able to book venues by selecting a date and submitting a booking request.
FR4	Owners shall be able to list new venues with attributes like name, location, price, and capacity.
FR5	Owners shall be able to edit or delete their venue listings.
FR6	Admin shall be able to approve or reject venue listings.
FR7	The system shall notify users (email/SMS style) upon successful booking (Observer Pattern).
FR8	Admin shall be able to add or remove venues using command-based actions (Command Pattern).
FR9	The system shall dynamically apply different search strategies (Strategy Pattern).
FR10	A factory shall be used to create different types of bookings such as wedding or conference bookings.

---

4. Non-Functional Requirements (NFRs)

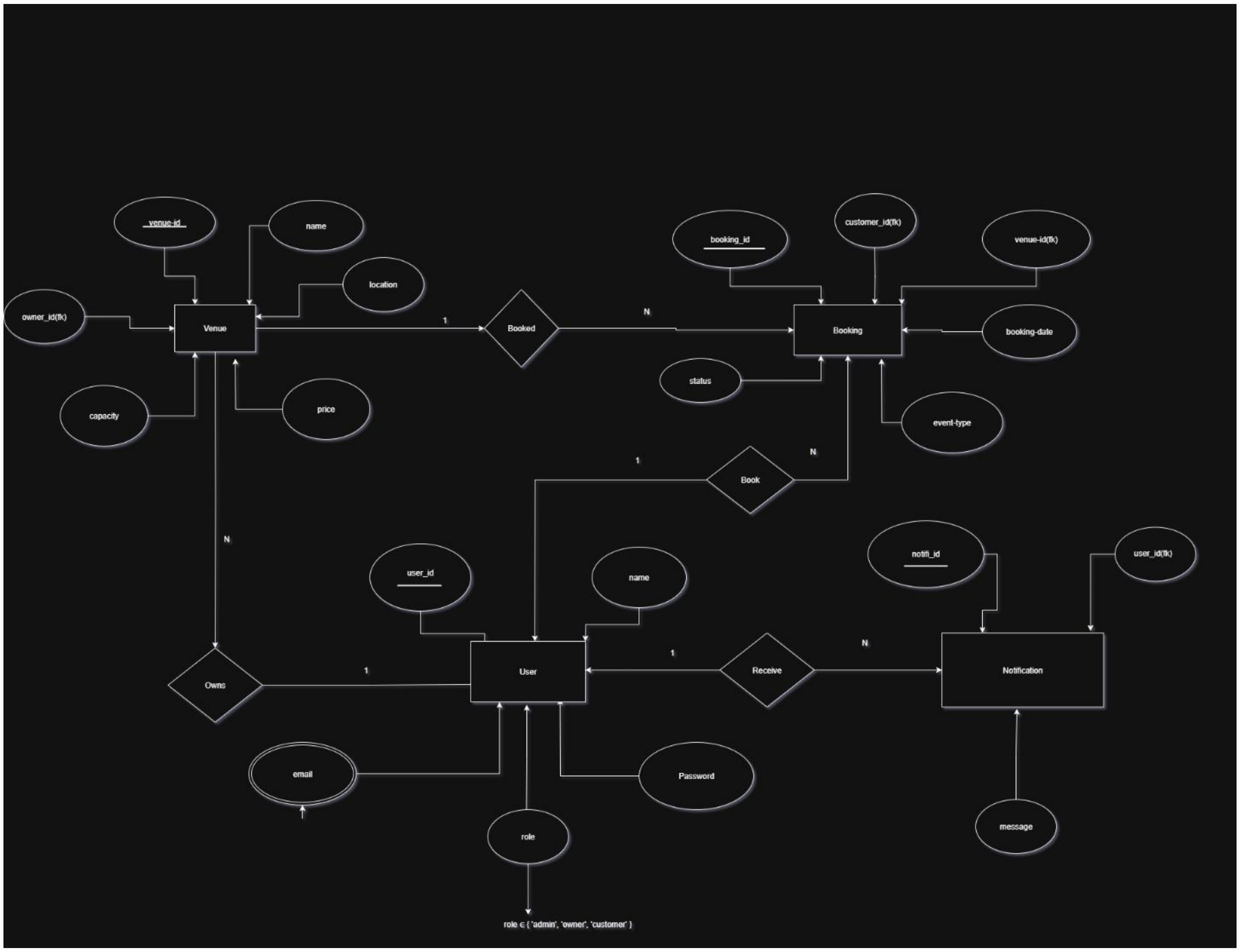
Non-functional requirements define how the system performs, rather than what it does.

NFR Code	Requirement Description
NFR1	**Performance**: The system shall respond to all user actions within 2 seconds.
NFR2	**Security**: Only registered users shall be able to log in. Admin actions shall require elevated permissions.
NFR3	**Usability**: The system shall have a user-friendly and intuitive interface for both terminal and optional GUI.
NFR4	**Maintainability**: The system shall be modular and follow OOP principles for easy updates and debugging.
NFR5	**Scalability**: The system shall support an increasing number of users and venue listings without performance loss.
NFR6	**Reliability**: The system shall validate user inputs and handle incorrect data gracefully (e.g., invalid booking dates).
NFR7	**Portability**: The application shall be able to run on any modern operating system with a C++ compiler.
NFR8	**Extensibility**: The system shall allow for future features such as online payment integration, user ratings, and reviews.

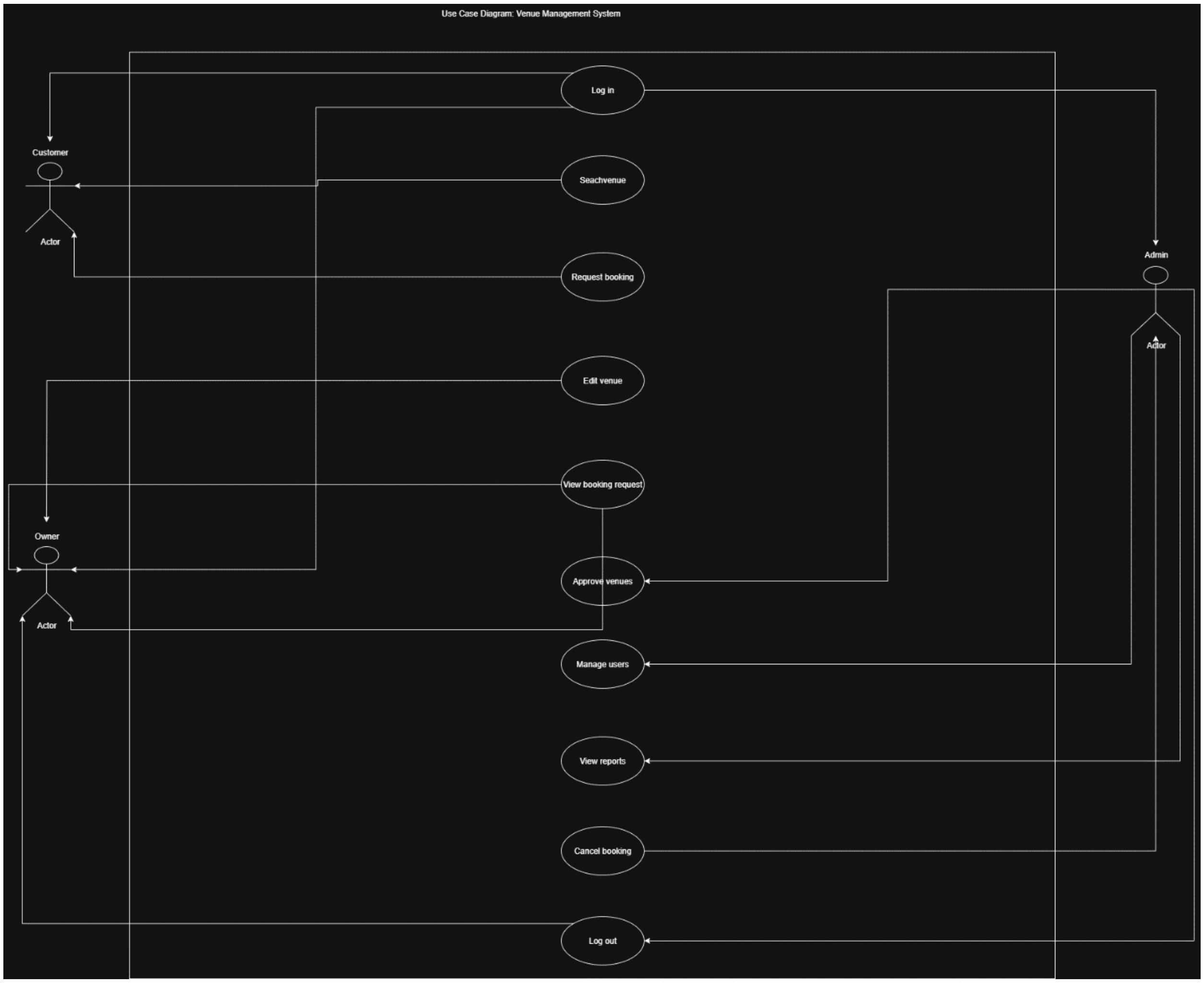
Conclusion

The **Venue Management System** will serve as a complete solution for users seeking a fast, transparent, and efficient way to find and book venues. By combining clean architecture with key design patterns and strong user role management, the project will be a strong demonstration of software engineering principles. It will not only address a real-world problem but also be a robust and extendable system for future growth.

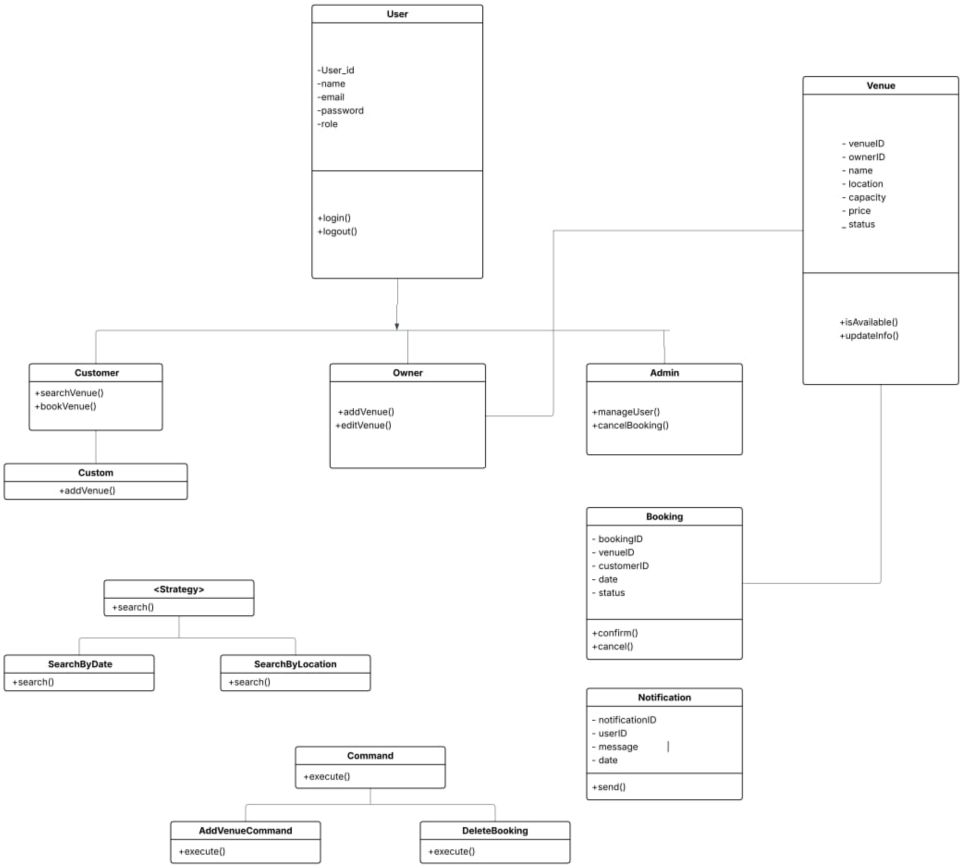
ER Diagram:

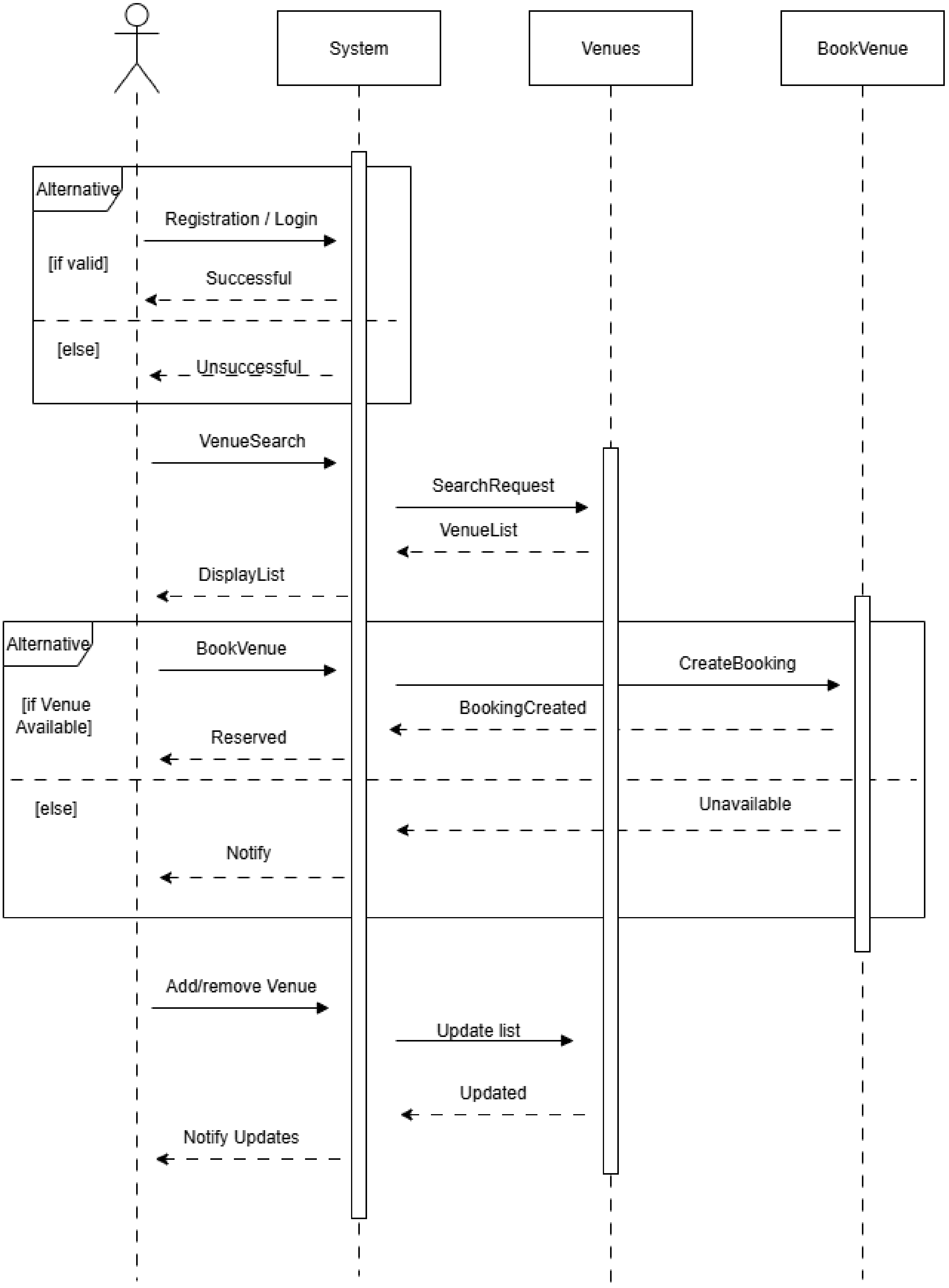

venue-id
name
location
owner_id(fk)
Venue
capacity
price
1
Booked
N
booking_id
customer_id(fk)
venue-id(fk)
Booking
booking-date
status
event-type
1
Book
N
N
notifi_id
user_id(fk)
user_id
name
Owns
1
User
1
Receive
N
Notification
email
Password
role
message
role ∈ { 'admin', 'owner', 'customer' }

Use-Case Diagram:



Class Diagram:




System
Venues
BookVenue
Alternative
Registration / Login
[if valid]
Successful
[else]
Unsuccessful
VenueSearch
SearchRequest
VenueList
DisplayList
Alternative
BookVenue
CreateBooking
[if Venue Available]
BookingCreated
Reserved
[else]
Unavailable
Notify
Add/remove Venue
Update list
Updated
Notify Updates