

***An Automated Meeting Room Booking System***

# 1. Introduction

This design document outlines the structure, functionality, and design considerations for the Automated Meeting Room Booking System. The system allows users to manage rooms, amenities, meetings, and user accounts. It supports different user roles such as Admin, Manager, and Member, each with specific capabilities.

2. System Architecture

The system follows a layered architecture with separation of concerns. It consists of the following layers:

1. **Domain Layer**: Represents the core business entities.
2. **Service Layer**: Contains business logic and operations.
3. **DAO Layer**: Manages data access and persistence.
4. **UI Layer**: User interface (console-based in our case).

**2.1 Domain Layer**

The domain layer includes the core business entities with their attributes and methods. These entities are:

* **Amenities**
* **Meetings**
* **Rooms**
* **Users**

**2.2 Service Layer**

The service layer contains business logic and provides methods to interact with the domain objects. The services include:

* **UserService**
* **LoginService**
* **RoomService**
* **AmenityService**
* **MeetingService**

**2.3 DAO Layer**

The DAO (Data Access Object) layer is responsible for CRUD operations on the database. This layer interacts with the database and provides methods to fetch or update data.

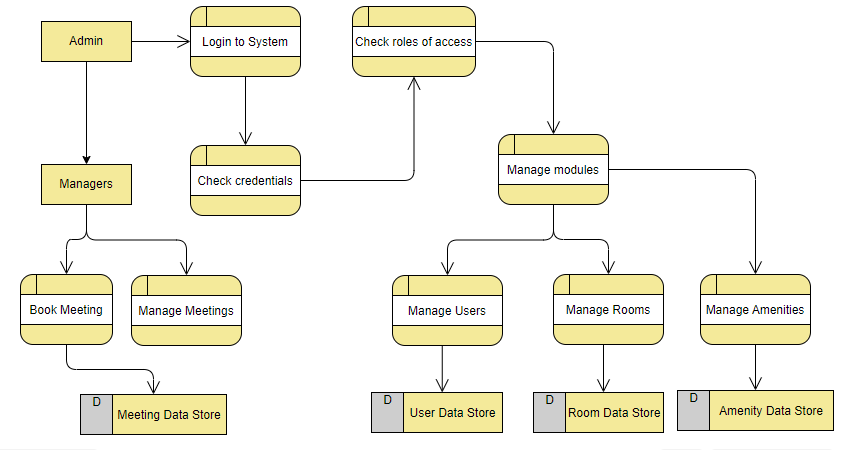
**DAO Interfaces**:

* UserDAO
* LoginDAO
* RoomDAO
* AmenityDAO
* MeetingDAO

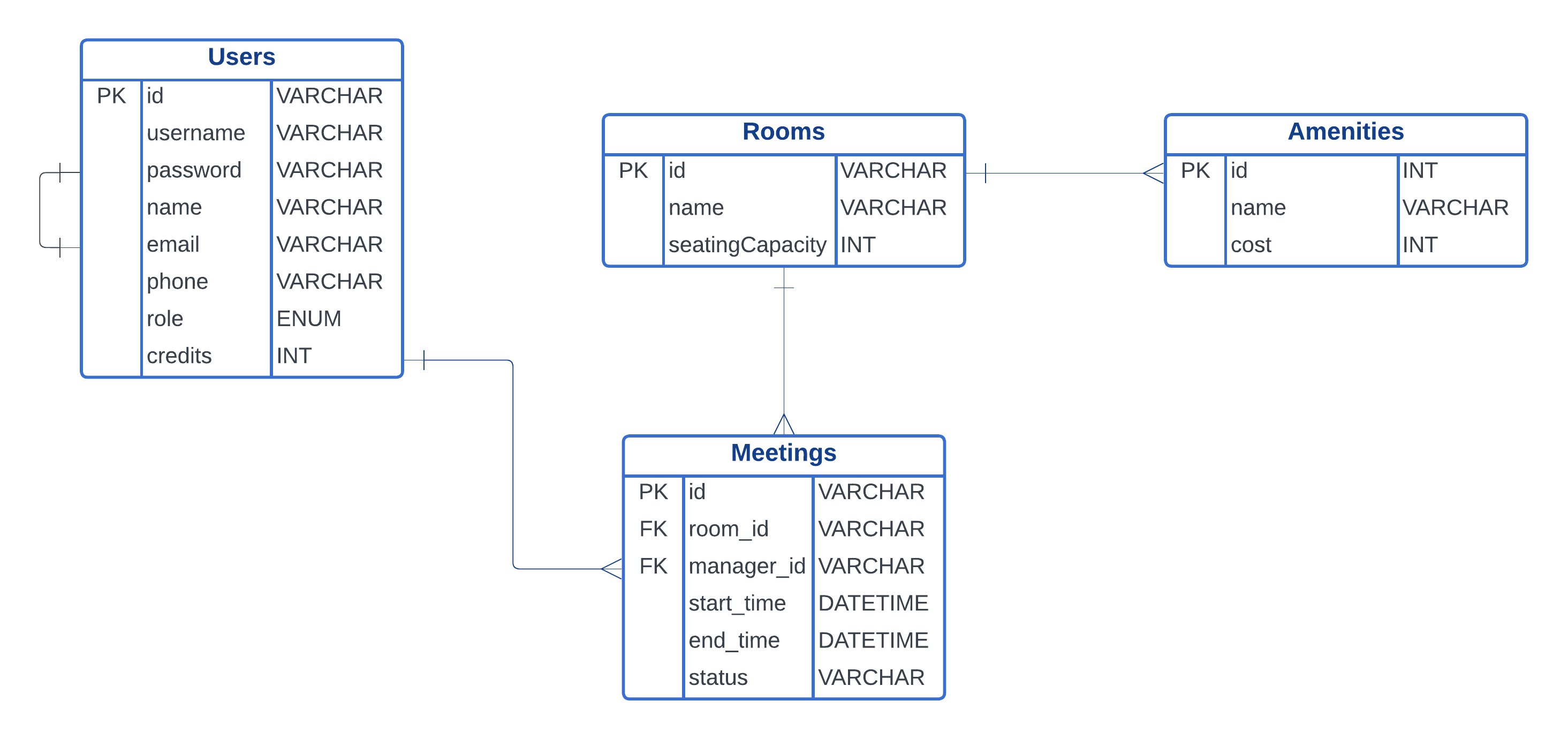
**2.4 UI Layer**

The presentation layer in this system is a console-based interface. It uses a MainApp class to provide a menu-driven interface for users to interact with the system. The main functionalities include authentication and user-specific menus.

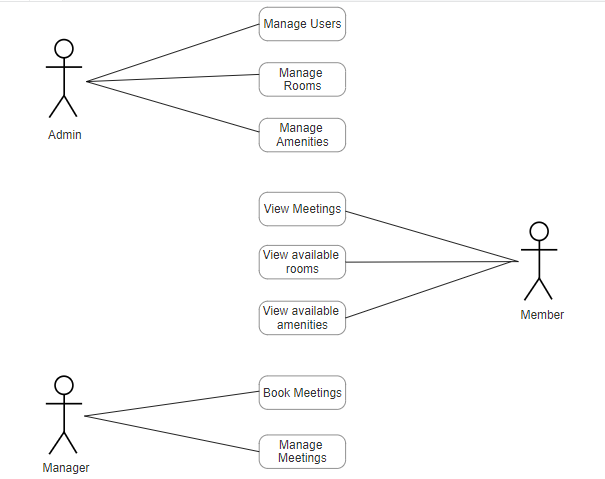
3. Data Flow Diagram (DFD)



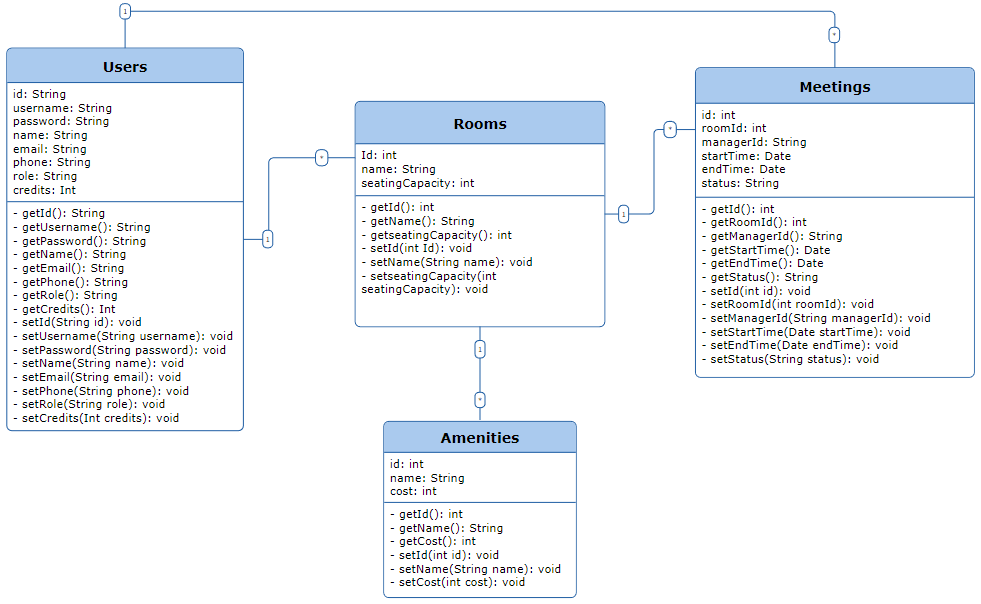
4. Entity-Relationship (ER) Diagram

****

5. Use Case Diagram



6. Class Diagram (UML)

****