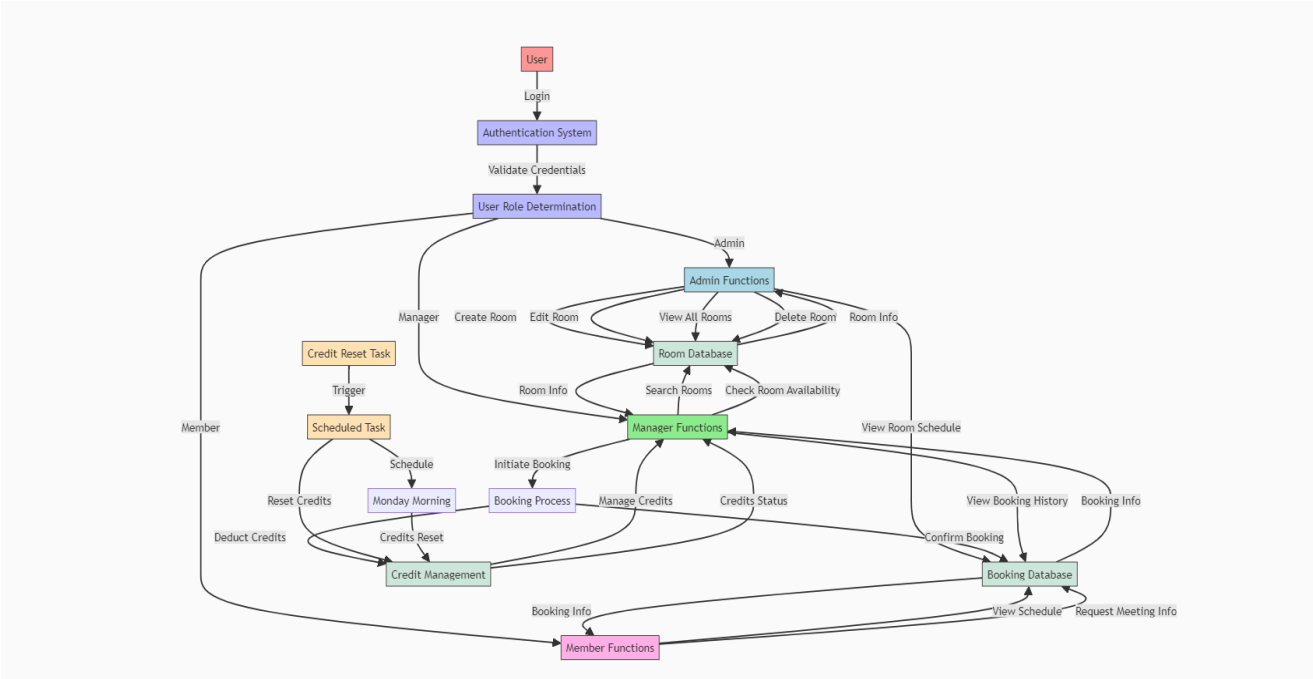
Design Documents

Data Flow Diagram:



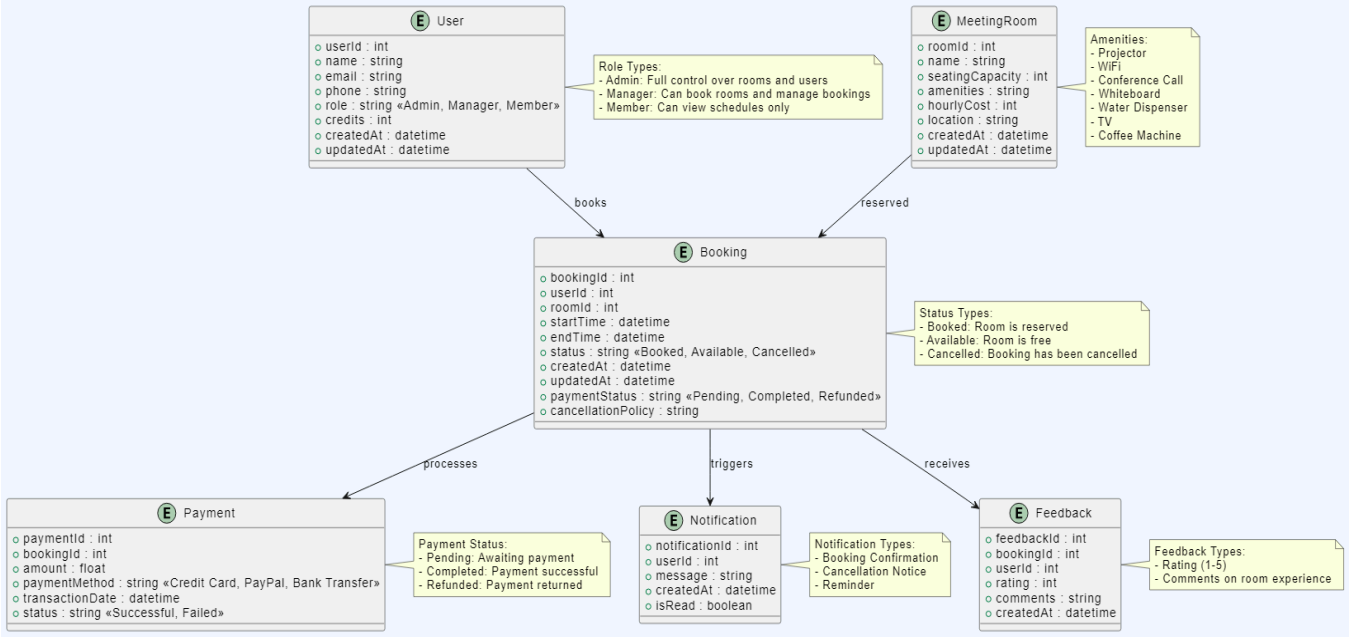
This image depicts a flowchart that outlines the architecture and functionality of the Automated Meeting Room Booking System. Here’s a breakdown of the components:

Key Components:

- User Authentication:
    - Users must log in to access the system.
    - The system validates credentials and determines the user role (Admin, Manager, or Member).
- User Roles:
    - Admin Functions:
        - Admins can create, edit, delete, and view all meeting rooms.
        - They interact with the Room Database to manage room information.
    - Manager Functions:
        - Managers can initiate bookings, check room availability, and manage credits.
        - They have access to view room schedules and booking history.
    - Member Functions:
        - Members can view schedules and request meeting information but cannot book rooms.
- Credit Management:
    - A scheduled task resets manager credits every Monday morning.
    - Managers can deduct credits when booking rooms.
- Booking Process:
    - The flow includes searching for rooms, checking availability, and confirming bookings.
    - Booking information is stored in the Booking Database.

- Scheduled Tasks:
    - The system includes a credit reset task that runs on a defined schedule.

ER diagram:



This image depicts a ER class diagram for a booking system, showcasing the relationships and attributes of various entities involved in the system. Here's a breakdown of the key components:

Entities:

- User
    - Attributes: `userId`, `name`, `email`, `phone`, `role`, `credits`, `createdAt`, `updatedAt`.
    - Role Types:
        - Admin: Full control over rooms and users.
        - Manager: Can book rooms and manage bookings.
        - Member: Can view schedules only.
- MeetingRoom
    - Attributes: `roomId`, `name`, `seatingCapacity`, `amenities`, `hourlyCost`, `location`, `createdAt`, `updatedAt`.
    - Amenities include: Projector, WiFi, Conference Call, Whiteboard, Water Dispenser, Coffee Machine.
- Booking
    - Attributes: `bookingId`, `userId`, `roomId`, `startTime`, `endTime`, `status`, `createdAt`, `updatedAt`, `paymentStatus`, `cancellationPolicy`.
    - Status Types: Booked, Available, Canceled.

- Payment
    - Attributes: `paymentId`, `bookingId`, `amount`, `paymentMethod`, `transactionDate`, `status`.
    - Payment Methods include: Credit Card, PayPal, Bank Transfer.
    - Payment Status: Pending, Completed, Refunded.

- Notification
    - Attributes: `notificationId`, `userId`, `message`, `createdAt`, `isRead`.
    - Notification Types: Booking Confirmation, Cancellation Notice, Reminder.

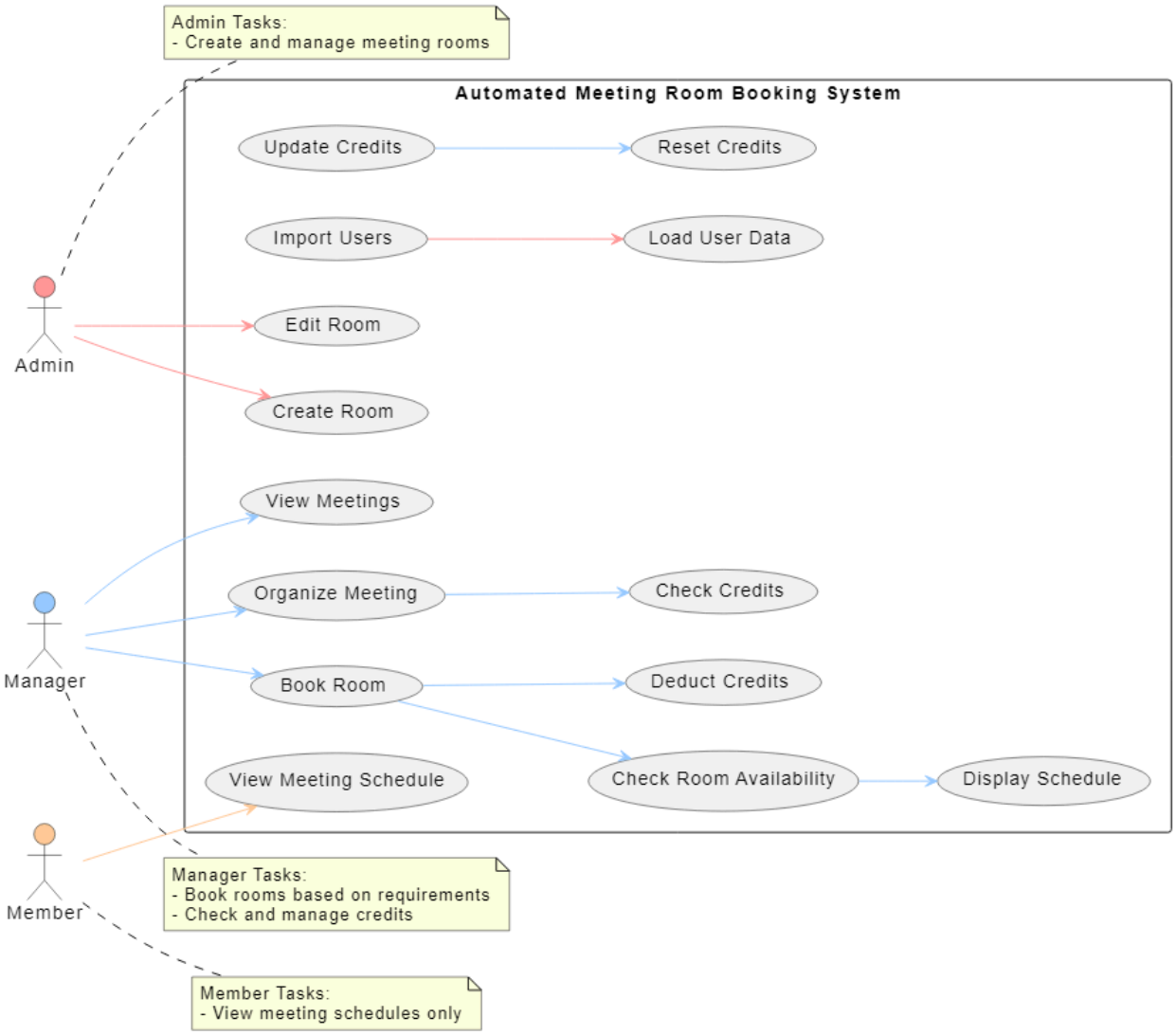
- Feedback
    - Attributes: `feedbackId`, `bookingId`, `userId`, `rating`, `comments`, `createdAt`.
    - Feedback Types: Rating (1-5), Comments on room experience.

Relationships:

- User books MeetingRoom through Booking.
- Booking is processed through Payment.
- Notifications are triggered based on Booking events.
- Feedback is associated with specific Bookings.

This diagram effectively illustrates the structure and interactions of the components within the booking system, providing a clear overview of how users, rooms, bookings, payments, notifications, and feedback are interconnected.

USE CASE Diagram:



The image depicts a diagram for an "Automated Meeting Room Booking System," outlining the roles and tasks associated with three types of users: Admin, Manager, and Member.

Key Components:

- Admin Tasks:
    - Create and manage meeting rooms.

        - Additional functionalities like updating credits, importing users, editing rooms, and viewing meetings.
- Manager Tasks:
    - Book rooms based on requirements.
    - Check and manage credits.
    - Organize meetings and view meeting schedules.
- Member Tasks:
    - View meeting schedules only.

Functionalities:

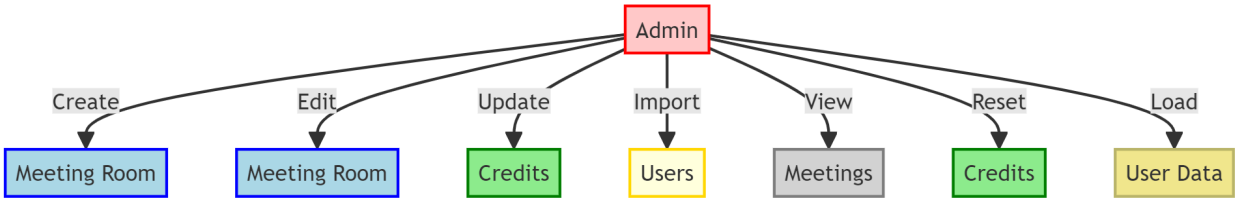
- The diagram illustrates various actions that can be performed by each user type, such as:
    - Admin: Can reset credits, load user data, and manage room-related tasks.
    - Manager: Can check credits, book rooms, and check room availability.
    - Member: Limited to viewing schedules.

Visual Structure:

- The roles are represented by icons (Admin, Manager, Member) with connecting lines to their respective tasks, showcasing the flow of actions and responsibilities within the system.

This diagram serves as a visual representation of user roles and their associated tasks within the booking system, helping to clarify the system's functionality and user interactions.

1. Admin Use Case Diagram



Overview

The Admin role is crucial for managing the overall system. Admins have comprehensive control over meeting rooms and user management.

Use Cases

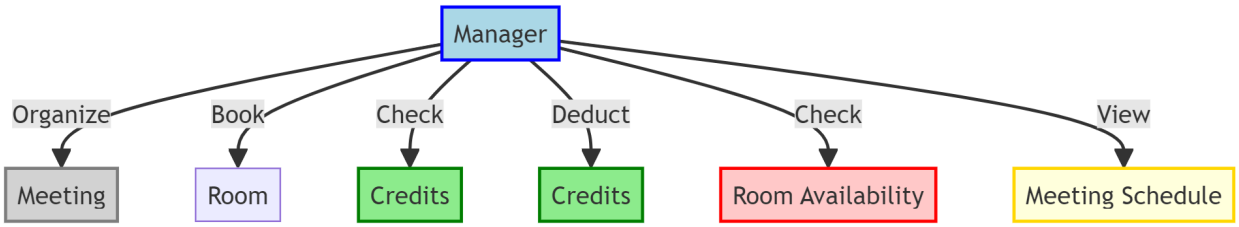
- Create Meeting Room: Admins can set up new meeting rooms with various configurations.
- Edit Meeting Room: They can modify existing room details, such as amenities and seating capacity.
- Update Credits: Admins manage the credits assigned to managers, ensuring they have enough for bookings.
- Import Users: They can load user data from an XML file, facilitating easy user management.
- View Meetings: Admins can access the schedule of all meetings held in the organization.
- Reset Credits: They have the authority to reset credits for managers, typically on a weekly basis.

- Load User Data: Admins can import user data from a specified file path on the server.

Visual Representation

In the diagram, each use case is connected to the Admin role, showcasing the various functionalities available to them. The use cases are color-coded for clarity, with different colors representing different categories of tasks (e.g., room management, user management).

2. Manager Use Case Diagram



Overview

The Manager role is focused on organizing and booking meeting rooms. Managers are responsible for ensuring that meetings are scheduled efficiently and within budget.

Use Cases

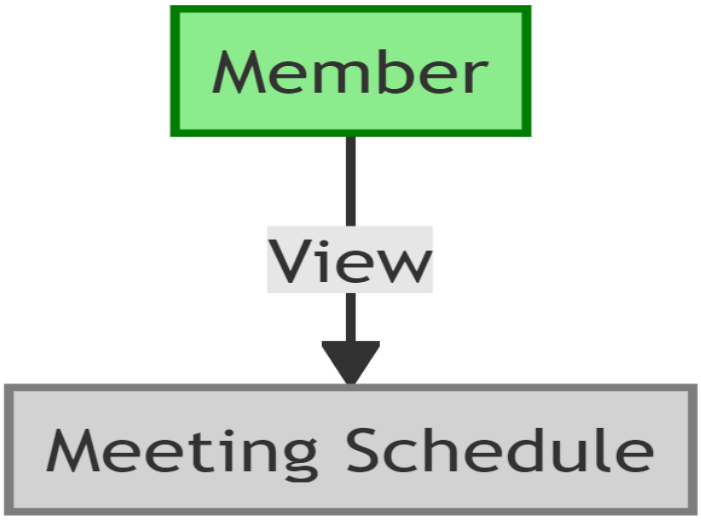
- Organize Meeting: Managers can plan and arrange meetings, specifying details like time and participants.

- Book Room: They can reserve meeting rooms based on their requirements, such as seating capacity and amenities.
- Check Credits: Managers can verify their available credits before making a booking.
- Deduct Credits: When a room is booked, the corresponding credits are deducted from their account.
- Check Room Availability: Managers can check if a room is available for the desired time slot before booking.
- View Meeting Schedule: They can access the schedule of meetings they have organized, allowing for better planning.

Visual Representation

In this diagram, the Manager role is linked to all relevant use cases, illustrating their capabilities. The use cases are also color-coded to differentiate between various tasks related to meeting organization and room management.

3. Member Use Case Diagram



Overview

The Member role is limited in functionality compared to Admins and Managers. Members primarily have access to view scheduled meetings.

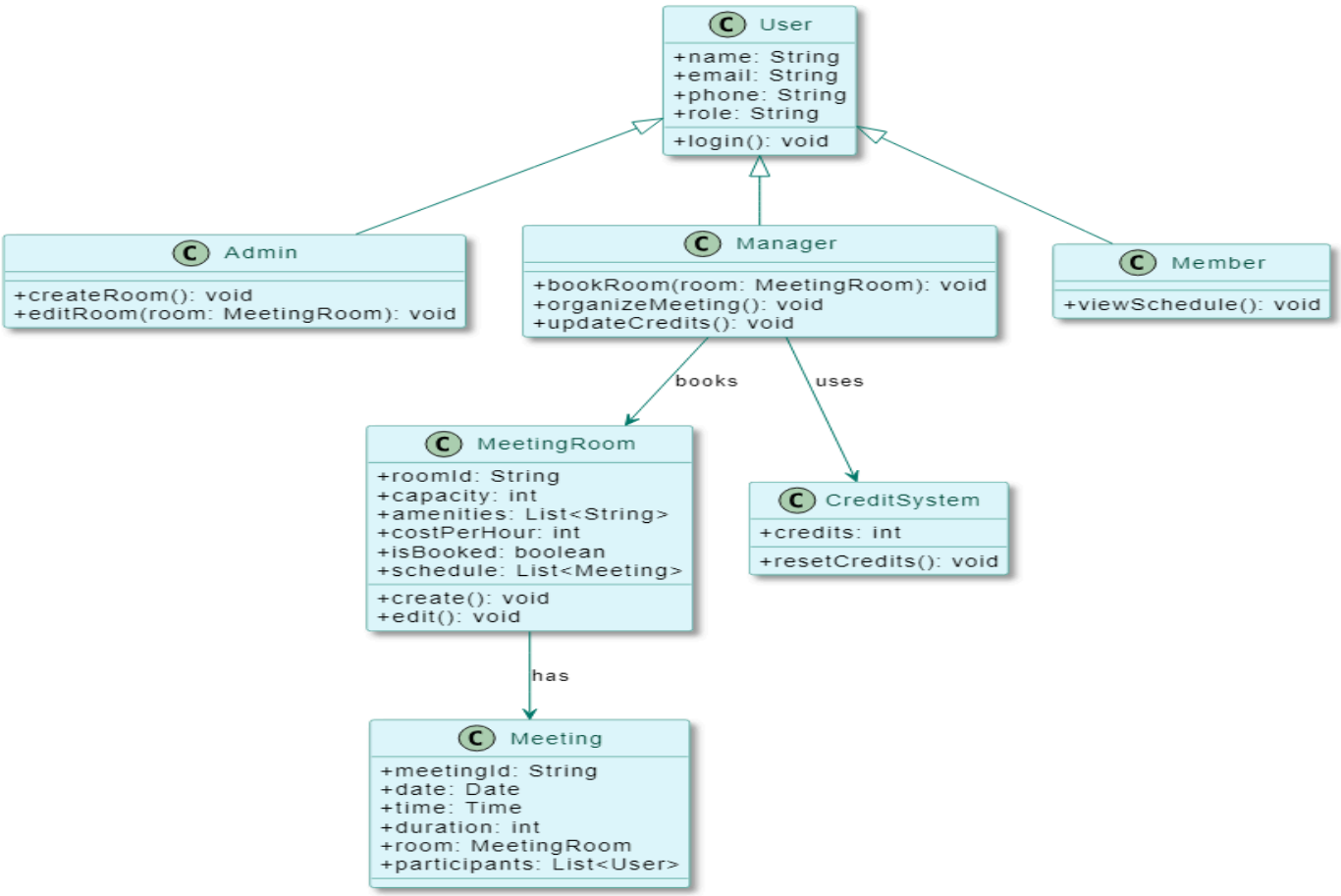
Use Cases

- View Meeting Schedule: Members can access the schedules of meetings they are invited to or can attend. This allows them to stay informed about upcoming meetings.

Visual Representation

In the Member use case diagram, there is a single use case connected to the Member role. The simplicity of this diagram reflects the limited scope of actions available to members, focusing solely on viewing meeting schedules.

Class (UML) Diagram



This UML class diagram represents a system for managing users, meeting rooms, and meetings. Here's a breakdown of its components:

Classes and Relationships

- User:
  - Attributes:
    - `name`: String
    - `email`: String
    - `phone`: String
    - `role`: String
  - Methods:

        - `login()`: Method to log the user into the system.
    - Relationships:
        - Inherited by `Admin`, `Manager`, and `Member`.
- Admin (inherits from User):
    - Methods:
        - `createRoom()`: Method to create a new meeting room.
        - `editRoom(room: MeetingRoom)`: Method to edit an existing meeting room.
- Manager (inherits from User):
    - Methods:
        - `bookRoom(room: MeetingRoom)`: Method to book a meeting room.
        - `organizeMeeting()`: Method to organize a meeting.
        - `updateCredits()`: Method to update user credits.
- Member (inherits from User):
    - Methods:
        - `viewSchedule()`: Method to view the member's meeting schedule.
- MeetingRoom:
    - Attributes:
        - `roomId`: String
        - `capacity`: int
        - `amenities`: List<String>
        - `costPerHour`: int
        - `isBooked`: boolean
        - `schedule`: List<Meeting>
    - Methods:
        - `create()`: Method to create a meeting room.
        - `edit()`: Method to edit meeting room details.

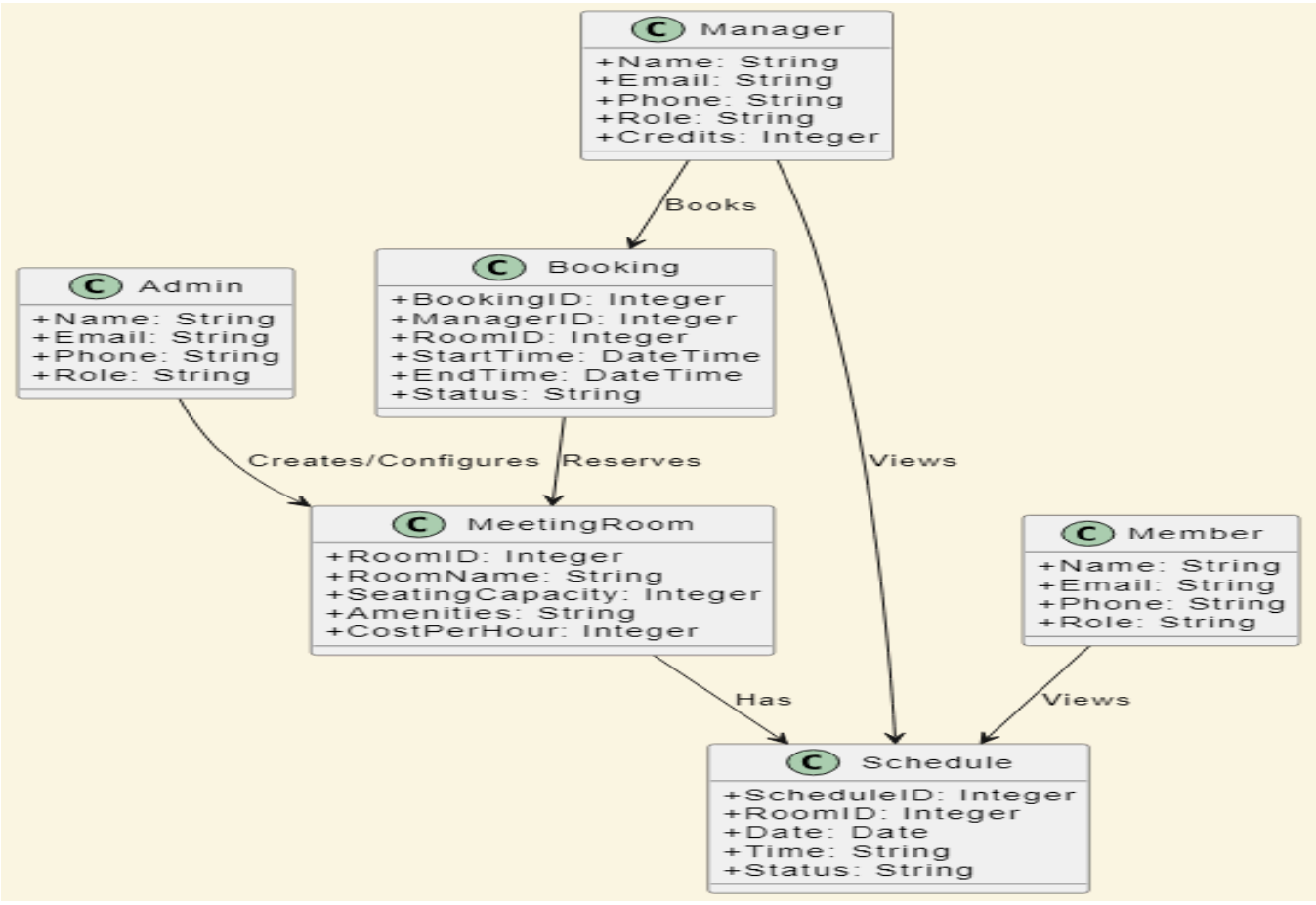
    - Relationships:
        - Can be booked by the Manager and is associated with the Meeting.
- Meeting:
    - Attributes:
        - `meetingId`: String
        - `date`: Date
        - `time`: Time
        - `duration`: int
        - `room`: MeetingRoom
        - `participants`: List<User>
    - Represents a scheduled meeting.

- CreditSystem:
    - Attributes:
        - `credits`: int
    - Methods:
        - `resetCredits()`: Method to reset the credits for the user.
    - Relationships:
        - Used by `Manager` for managing credits.

Relationships Explained

- Inheritance: `Admin`, `Manager`, and `Member` are specialized types of `User`, inheriting its attributes and methods.
- Association:
    - `Manager` can book and manage `MeetingRoom`.
    - `MeetingRoom` has a schedule consisting of multiple `Meeting` instances.
    - `Meeting` has participants, which are users.

RDBMS table structure with simple ER diagram



This is Relational database management system (RDBMS) table structure represented in the ER diagram:

Entities and Attributes

- Manager
    - Attributes: Name, Email, Phone, Role, Credits
    - Description: Represents a manager who oversees bookings.
- Admin
    - Attributes: Name, Email, Phone, Role
    - Description: Represents an administrator who manages the system.
- Booking
    - Attributes: BookingID, ManagerID, RoomID, StartTime, EndTime, Status
    - Description: Represents a booking made for a meeting room, linked to a manager and a specific room.
- MeetingRoom
    - Attributes: RoomID, RoomName, SeatingCapacity, Amenities, CostPerHour
    - Description: Represents the meeting rooms available for booking, including their features and costs.
- Member
    - Attributes: Name, Email, Phone, Role
    - Description: Represents a member who can view bookings and schedules.

- Schedule
    - Attributes: ScheduleID, RoomID, Date, Time, Status
    - Description: Represents the schedule of a meeting room for specific dates and times.

Relationships

- Books: The Manager can create bookings.
- Creates/Configures: The Admin can create and configure bookings.
- Reserves: Members can reserve meeting rooms through bookings.
- Has: Meeting rooms have schedules associated with them.
- Views: Members and Managers can view bookings and schedules