Outlook

RE: Database structure for Meeting room

From Tuhin Das <tkdas@yotta.com>
Date Mon 6/23/2025 1:17 PM
To Suhaas Nandulamattam <i_snandulamattam@yotta.com>

S.No	Module	Method	Endpoint	Description
✓ 1	User	✓ POST	/users/register	Register a new user
		✓ POST	/users/login	Authenticate and get token
		✓ GET	/users/me	Get current user info
		✓ GET	/users/:id	Get user by ID (admin only)
		✓ GET	/users	List all users (admin only)
		✓ PUT	/users/me	Update current user
		✓ DELETE	/users/me	Delete current user
✓ 2	Location	✓ GET	/locations	List all locations
		✓ GET	/locations/:id	Get location by ID
		✓ POST	/locations	Create a new location (admin only)
		✓ PUT	/locations/:id	Update location (admin only)
		✓ DELETE	/locations/:id	Delete location (admin only)
✓ 3	Meeting Room	✓ GET	/meeting-rooms	List all meeting rooms
		✓ GET	/locations/:locationId/rooms	List rooms by location
		✓ GET	/meeting-rooms/:id	Get room by ID
		✓ POST	/locations/:locationId/rooms	Create a new room (admin only)
		✓ PUT	/meeting-rooms/:id	Update room (admin only)
		✓ DELETE	/meeting-rooms/:id	Delete room (admin only)
4	✓ Room Features & Room Feature Mapping	✓ GET	/room-features	List all available features
		✓ POST	/room-features	Add new feature (admin only)
		✓ GET	/meeting-rooms/:roomId/features	List features of a room
		✓ POST	/meeting-rooms/:roomId/features	Add feature to room (admin only)
		✓ DELETE	/meeting-rooms/:roomId/features/:featureId	Remove feature from room
✓ 5	Booking	✓ GET	/bookings	List all bookings (admin only)
		✓ GET	/bookings/my	List current user's bookings
		✓ GET	/bookings/:id	Get booking details
		✓ POST	/meeting-rooms/:roomId/bookings	Book a room
		✓ PUT	/bookings/:id	Edit booking (if owned or admin)
		✓ DELETE	/bookings/:id	Cancel booking
✓ 6	Participants	✓ GET	/bookings/:bookingId/participants	List participants
		✓ POST	/bookings/:bookingId/participants	Add participant(s)
		✓ PUT	/bookings/:bookingId/participants/:userId	Update participant status
		✓ DELETE	/bookings/:bookingId/participants/:userId	Remove participant
✓ 7	Meeting Minutes	✓ GET	/bookings/:bookingId/minutes	Get meeting minutes
		✓ POST	/bookings/:bookingId/minutes	Create meeting minutes
		✓ PUT	/bookings/:bookingId/minutes	Update meeting minutes
		✓ DELETE	/bookings/:bookingId/minutes	Delete meeting minutes
8	Recurring Meeting	POST ✓	/bookings/recurring	Create recurring booking
		GET	/bookings/:parentId/instances	List all instances of a recurring meeting
		DELETE	/bookings/:parentId/instances/:instanceId	Delete specific instance
✓ 9	Audit Log	✓ GET	/audit-logs	List audit logs (admin only)
		✓ GET	/audit-logs/user/:userId	Filter logs by user

From: Tuhin Das
Sent: 23 June 2025 11:00
To: Suhaas Nandulamattam <i_snandulamattam@yotta.com>
Subject: RE: Database structure for Meeting room

Recurring Bookings - users can create recurring bookings (daily, weekly, etc.).

- RecurringRule_ID (PK)
- Booking_ID (FK)
- Frequency ENUM('Daily', 'Weekly', 'Monthly')
- End_Date
- Repeat_On (e.g., "Monday,Wednesday")

User preferences - Stores default preferences for users (e.g., preferred location, auto-email settings).

- Preference_ID (PK)
- User_ID (FK)
- Default_Location_ID (FK)
- Receive_Email_Notifications BOOLEAN
- TimeZone

Audit Log - Tracks changes made by users for compliance or debugging.

- Log_ID (PK)
- User_ID (FK)
- Action_Type (e.g., "Booking Created", "Room Updated")
- Description
- Timestamp
- IP_Address

From: Tuhin Das
Sent: 23 June 2025 10:33
To: Suhaas Nandulamattam <i_snandulamattam@yotta.com>
Subject: RE: Database structure for Meeting room

Relationship	Description
Users ↔ MeetingRooms	A user creates a room (one-to-many)
Locations ↔ MeetingRooms	One location has many rooms (one-to-many)
MeetingRooms ↔ RoomFeatures	Many-to-many via RoomFeatureMapping
Users ↔ Bookings	A user creates a booking (one-to-many)
Bookings ↔ Participants	One booking can have many participants (one-to-many)
Bookings ↔ MeetingMinutes	One booking can have one set of minutes (one-to-one)

From: Tuhin Das
Sent: 23 June 2025 10:28
To: Suhaas Nandulamattam <i_snandulamattam@yotta.com>
Subject: Database structure for Meeting room

Functionality List:

- Create a Meeting Room
- Add Features to Meeting Rooms
- Search Meeting Rooms by Location & Availability
- Book a Meeting Room
- Edit or Cancel a Booking
- Add Participants to a Meeting
- Send Email Notifications
- Store Meeting Minutes

User Table - Stores user information (admins, employees, etc.)

- User_ID (PK)
- First_Name
- Last_Name
- Email (unique)
- Password_Hash
- Role (e.g., Admin, User)
- Created_At

Meeting Room - Stores meeting room details.

- Room_ID (PK)
- Name
- Location_ID (FK)
- Description
- Capacity
- Created_At
- Updated_At

Location - Represents different office locations or buildings.

- Location_ID (PK)
- Name
- Address
- City
- Postal_Code

Room Feature - List of possible features a room can have.

- Feature_ID (PK)
- Name (e.g., "Whiteboard", "Video Conference", "Projector")

Room Feature Mapping - Many-to-many mapping between rooms and features.

- Room_ID (FK)
- Feature_ID (FK)
- Primary Key (Room_ID, Feature_ID)

Booking - Tracks all room bookings.

- Booking_ID (PK)
- Room_ID (FK)
- User_ID (FK, creator of booking)
- Title
- Start_Time (datetime)
- End_Time (datetime)
- Status (e.g., Confirmed, Cancelled)
- Created_At
- Updated_At

Participants - List of participants in a meeting.

- Participant_ID (PK)
- Booking_ID (FK)
- User_ID (FK)
- Invitation_Status (e.g., Pending, Accepted, Declined)
- Notification_Sent BOOLEAN

Meeting Minutes - Stores minutes of each meeting.

- Minute_ID (PK)
- Booking_ID (FK)
- Notes_Text
- Attachments_Path
- Created_By (User_ID)
- Created_At

Email Log - Track sent emails related to meetings.

- Email_ID (PK)
- Booking_ID (FK)
- Recipient_Email
- Subject
- Body
- Sent_At
- Status (e.g., Sent, Failed)