Overview

This is a medical appointment scheduling system designed to facilitate the booking, management, and tracking of healthcare consultations between patients and medical professionals. The application provides secure authentication and authorization mechanisms to ensure appropriate access control, enables patients to schedule appointments with doctors for specific medical conditions, and allows healthcare providers to manage their appointment schedules effectively. The system includes automated confirmation notifications, appointment status tracking throughout the consultation lifecycle, and comprehensive appointment management capabilities including viewing, updating, and monitoring scheduled medical consultations within a healthcare facility or practice.

Key Personas

- Doctor
- Admin
- Patient

Key Functionalities

Patient:

- Authenticate identity with valid token to securely access appointment scheduling system
- Validate appointment request fields to ensure complete and accurate information
- Check appointment slot availability to avoid scheduling conflicts
- Create new appointment record with doctor and medical details
- Receive automated email confirmation with appointment details

Doctor:

- View paginated appointment listings to manage and review scheduled appointments
- Update appointment details including patient, doctor, disease, and date information
- Manage appointment status through valid transitions to track lifecycle from pending to completion

Admin:

- Verify user role-based permissions to ensure only authorized users can perform appointment operations
- Access full appointment management without filtering restrictions
- Update appointment details and manage appointment status with administrative privileges

Activity 1: Manage Medical Appointment Scheduling

User Action 1: Authenticate User Access

Description:
As a Patient, I want to authenticate my identity with a valid token so that I can securely access the appointment scheduling system.

- System validates JWT token from Authorization header when user makes a request
- System returns 401 Access Denied error when no token is provided in the request
- System returns 401 Invalid Token error when an invalid or expired token is provided
- System attaches verified user information to the request when token validation succeeds
- System allows authenticated users to proceed to the next step in the appointment workflow

User Action 2: Verify User Permissions

Description:
As an Admin, I want the system to verify user role-based permissions so that only authorized users can perform appointment operations.

- System validates JWT token from Authorization header before processing any appointment request
- System retrieves user information from verified token and attaches it to the request
- System returns 401 Access Denied error when no authorization token is provided
- System returns 401 Invalid Token error when JWT token verification fails
- System checks user role against required permissions for appointment operations
- System allows access when user role is included in the allowed roles list (doctor, admin)
- System returns 401 Permission Denied error when user role is not authorized for the requested operation
- System applies role-based filtering where patients can only access their own appointments

- System applies role-based filtering where doctors can only access appointments assigned to them
- System grants full appointment access to admin users without filtering restrictions

User Action 3: Validate Appointment Request

Description:
As a Patient, I want the system to validate all required appointment fields so that I can ensure my appointment request is complete and accurate.

- System validates that patient field is provided and not empty
- System validates that doctor field is provided and not empty
- System validates that disease field is provided and not empty
- System validates that appointmentDate field is provided and not empty
- System returns 400 status with message 'All inputs are required' when any required field is missing
- System converts appointment date string to proper Date object format
- System checks for existing appointments at the requested date and time
- System returns 409 status with message 'Appointment already existing' when time slot is unavailable
- System allows appointment creation to proceed when all required fields are valid and time slot is available

User Action 4: Check Appointment Availability

Description:
As a Patient, I want to check if my desired appointment slot is available so that I can avoid scheduling conflicts.

- System validates that no existing appointment exists for the requested date and time slot
- System returns 409 conflict error when appointment slot is already taken
- System allows appointment creation to proceed when appointment slot is available
- System checks appointment availability by querying the database for existing appointments with the same converted date
- System provides clear feedback to the patient about slot availability status

User Action 5: Create Appointment Record

Description:
As a Patient, I want to create a new appointment record with my doctor and medical details so that I can schedule my medical consultation.

- System validates that all required appointment fields (patient, doctor, disease, appointmentDate) are provided before creating the record
- System converts the appointment date string to a proper Date object for consistent database storage
- System checks that no existing appointment exists for the requested date and time slot
- System creates a new appointment record in the database with patient, doctor, disease references and sets status to 'pending' by default
- System populates the created appointment with complete patient, doctor, department, and disease details
- System sends an automated confirmation email to the patient with appointment details including date, department, and doctor information
- System returns a success response with status 'success' and message 'Appointment created' when the appointment is successfully created
- System returns a 400 error with message 'All inputs are required' when any required field is missing
- System returns a 409 conflict error with message 'Appointment already existing' when the requested time slot is already taken

User Action 6: Send Confirmation Email

Description:
As a Patient, I want to receive an automated email confirmation with my appointment details so that I have a record of my scheduled appointment.

- System sends automated email notification to patient after appointment creation
- Email contains patient's full name extracted from user profile
- Email includes complete appointment date and time information
- Email displays department name for the scheduled appointment
- Email is sent to the patient's registered email address
- Email confirmation is triggered automatically upon successful appointment record creation

- System retrieves comprehensive appointment details with populated patient, doctor, and department information before sending email

User Action 7: View Appointment Listings

Description:
As a Doctor, I want to view paginated appointment listings so that I can manage and review all scheduled appointments efficiently.

- System displays paginated appointment listings with 10 appointments per page
- Appointment listings include patient details excluding department and password fields
- Doctor information is populated with associated department details and password excluded
- Disease information is populated with associated department details
- Current page parameter is accepted from query string with default value of 1
- Role-based filtering is applied where doctors see only their assigned appointments
- Appointment listings return comprehensive appointment details with all related entities populated
- System handles pagination parameters correctly for efficient data retrieval
- Access is restricted to users with doctor or admin roles only
- Appointment data includes appointment date, status, and all referenced entities

User Action 8: Update Appointment Details

Description:
As a Doctor, I want to update appointment information including patient, doctor, disease, and date so that I can modify appointments as needed.

- User has valid permissions for appointment operations
- All required appointment fields including patient, doctor, disease, and appointment date are provided
- Appointment ID exists in the system and can be located for update
- Updated appointment date is converted to proper Date object format
- No existing appointment conflicts with the new appointment date and time

- Appointment record is successfully updated with new patient, doctor, disease, and date information
- System returns success status with confirmation message after successful update
- Role-based access control allows only doctors and admins to update appointments
- Updated appointment details are retrieved with populated patient, doctor, department, and disease information

User Action 9: Manage Appointment Status

Description:
As a Doctor, I want to update appointment status through valid transitions so that I can track the appointment lifecycle from pending to completion.

- System validates that the user has doctor or admin role permissions before allowing appointment status updates
- System accepts only valid appointment status values from the defined enum options
- System successfully updates appointment status when provided with valid status transition
- System returns appropriate error message when invalid status value is provided
- System maintains appointment lifecycle integrity during status transitions
- System provides confirmation response when appointment status is successfully updated
- System handles appointment status changes from pending to open appropriately
- System manages appointment completion workflow when status is updated to completed