**Smart Appointment Booking System**

**Problem Statement:**

Developing a smart appointment booking application for users to book an appointment with specific doctors and services, ensuring doctor’s schedules aren’t double booked, confirmations and reminders are sent automatically, and access is role based (Patient/Doctor/Admin) also preventing excessive login attempts.

**Objectives:**

* Provide self service appointment booking for patients with real time availability.
* Allow doctors to manage their schedules and appointment states.
* Give admins control over patients, services and policies.
* Send confirmation emails asynchronously and reliably.
* Enforce rate limiting and login throttling to reduce abuse and ensure fairness.
* Expose well documented APIs (Swagger).

**Function Requirements:**

1. User Management:

* Register/Login (JWT)
* Roles – DOCTOR, ADMIN, PATIENT
* CRUD for users

1. Appointment:

* Create Appointment – Patient chooses doctor, service, Date & Time from available slots.
* Prevent double booking.
* States: PENDING -> CONFIRMED -> COMPLETED OR CANCELLED
* Patient can reschedule/cancel appointments within policy window. Doctors can confirm/complete/cancel appointments with reason.

1. Doctor Schedule and Availability:

* Doctors have defined working hours, preventing patients from booking appointments after their working hours.
* Derived availability time slots by service duration.

1. Patient Services:

* Patients can sign up, login to book an appointment with a doctor.
* Patients can update their details and fetch their current details.
* Patients can cancel their appointments which are still not completed.

1. Admin:

* Admin can add/update/remove doctors and patients.
* Add, update or remove medical services.
* View and manage all appointments.
* Cancel appointments in case of conflicts or policy violations.

1. Notifications:

* Email confirmation on creation/update/cancellation.
* Reminder emails (e.g. 24hrs before start, 2hr before start)

**Non-Functional Requirements:**

* Simplicity – Clear separation of services.
* Reliability – Ensure no double booking of appointments.
* Security – Protect endpoints using Spring Security with JWT and hashed passwords.
* Scalability – Microservices should be independently deployable and scalable.

**High Level Architecture:**

A diagram of a software application

AI-generated content may be incorrect.

**Tech Stack Justification:**

* Spring Boot - Rapid development, support for microservices.
* MySQL – Relational database per service.
* Spring Security + JWT – For Role based access control.
* Spring cloud gateway – Centralized routing and rate limiting.
* RabbitMQ – Async notifications.
* Gmail SMTP – For sending mail notifications.
* Swagger – API Documentation.
* Docker – For running RabbitMQ

Github Repository link - https://github.com/utkarshmalik19/smart-appointment-booking-system