Doctor appointment application

Problem Statement:

Most of the health care personnels and organisations today follow the appointment system and ditched the first come first serve system and the way of booking appointments are still telephonic and the processing of appointments are also pretty much processed by humans by maining records in paper. Instead we propose a new system which can automate the process of booking appointments, viewing doctor profiles and matching patients or customers with doctors.

Technology to be used (3 Tiered application):

Backend API service: The backend functions as the interface between the frontend and the data source, which is often database management software like MySQL. The Java Spring boot framework is to be used to create the backend. An open source Java framework called Spring Boot uses Java Servlets to enable developers to build stand-alone, high-quality apps. The DAO (Data access object) paradigm, which is used to segregate the business logic from the data layer, is the foundation of a typical Spring Boot application (Database connection). An industry-grade REST API (Representational State Transfer) built on top of "HTTP" is produced by a spring boot application (Hyper Text Transfer Protocol).

Frontend Application: As the presentation layer for the entire application, the frontend plays the most significant role in a tiered application. The user interacts with the application at this layer. ***ReactJS***, a browser native JavaScript framework, is to be used to create the frontend layer. It provides a painless experience in constructing frontend applications that are lightning fast and simple to scale even if the requirements change in the future. React makes it simpler to write HTML in JavaScript by utilising JSX (JavaScript Extended). The frontend development process is sped up thanks to the ability to create reusable JSX elements and components thanks to the component-based approach and DRY (don't repeat yourself) pattern.

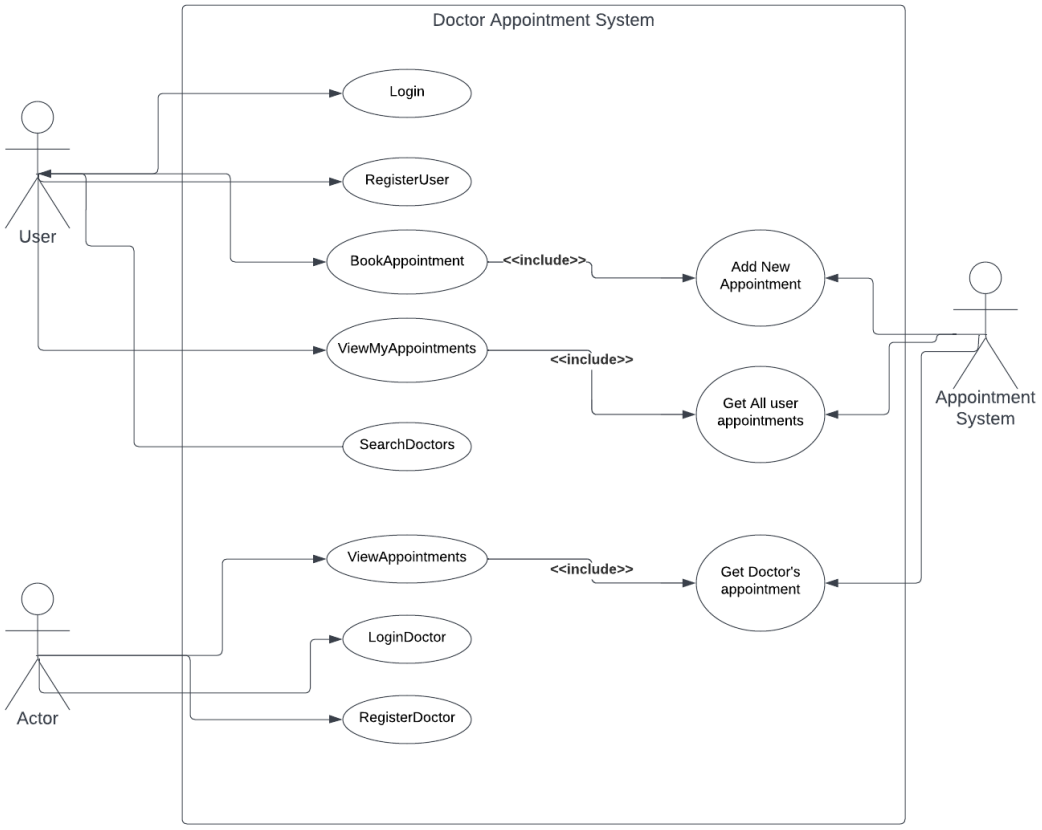
Database: Database layer represents the persistent data layer (Data is not erased even if the user exists the application). Database layer is to be made with the help of rigid, relational, table based database management system named ***MySQL.*** It is based on the SQL language. In MySQL, all the data are stored as Databases → Tables → Records → Fields.

Use cases:

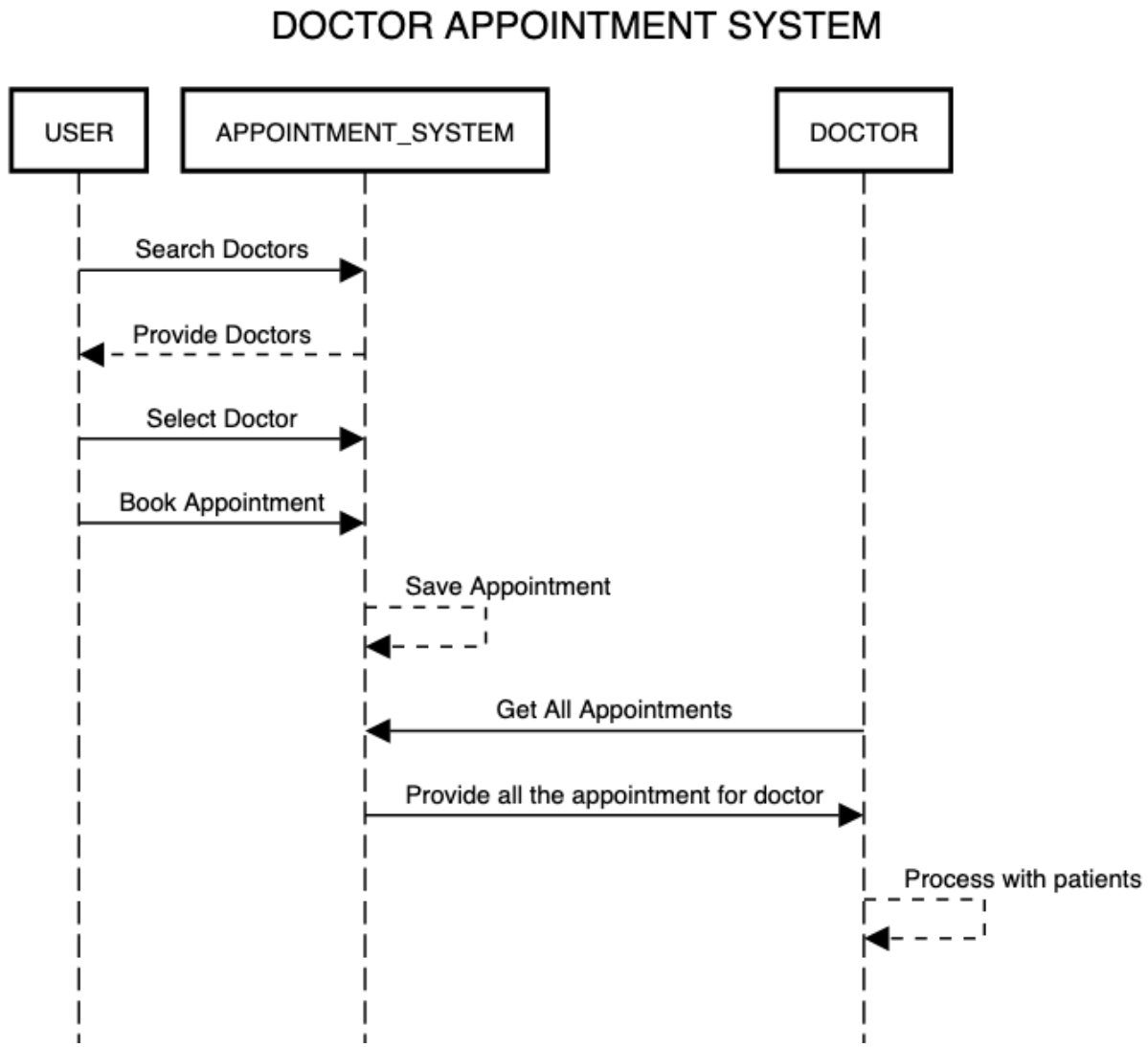
Use Case	**Description**

RegisterUser	Create a new user in the application and store in DB
RegisterDoctor	Create a new doctor in the application and store in DB
LoginUser	Check if user exists and his/her password is correct
LoginDoctor	Check if doctor exists and admin password is correct
SearchDoctors	Search for doctors based on preferences and departments (eg. Cardiology, Diabetic etc)
BookAppointment	Enter needed details and make a booking for an appointment with the doctor.
ViewAppointments	Doctor or his/her assistant can view the appointment sorted by time with patient's symptoms
ViewMyAppointments	User or Patient is able to track his previously made appointments
ViewDoctorProfile	User is able to view doctor's profile

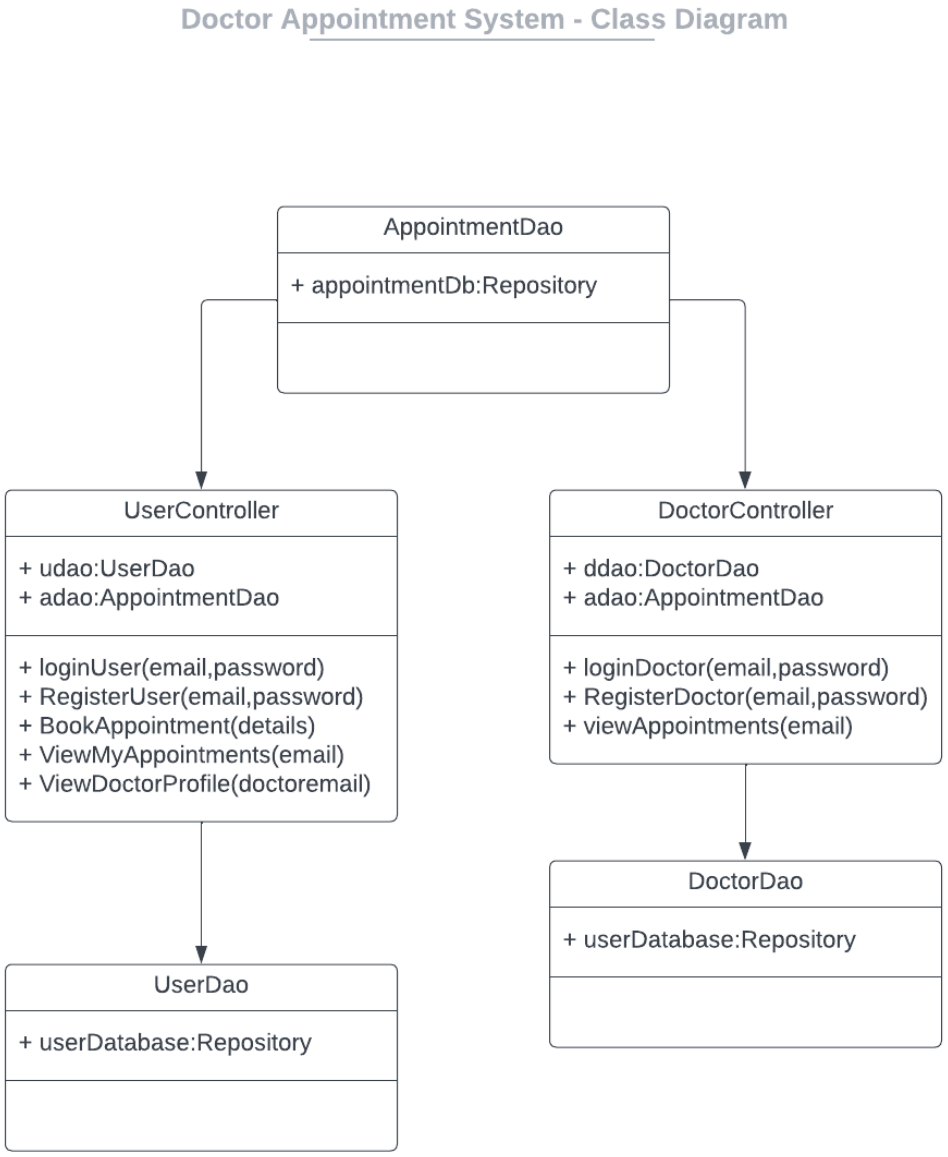
Use Case



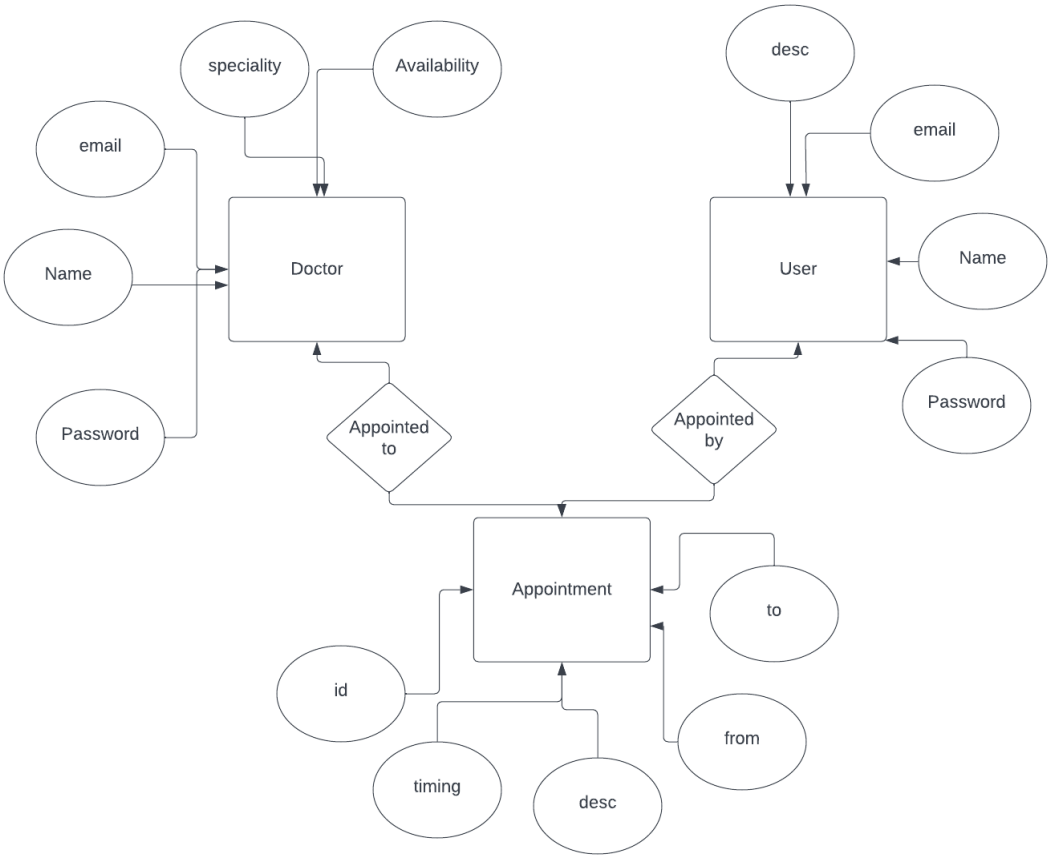
Sequence Diagram



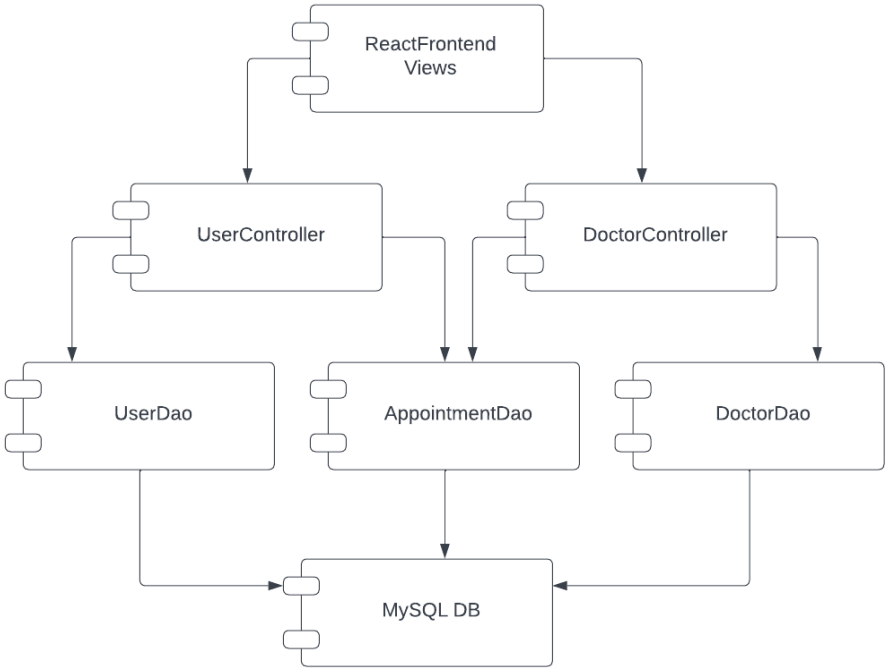
Class Diagram



Entity Relationship Diagram



Component Diagram



Deployment Diagram