Doctor Appointment Booking Web App

Presented By :
Zakariaa Bouzbita
Amina L'Bakali
Hajar Zouini
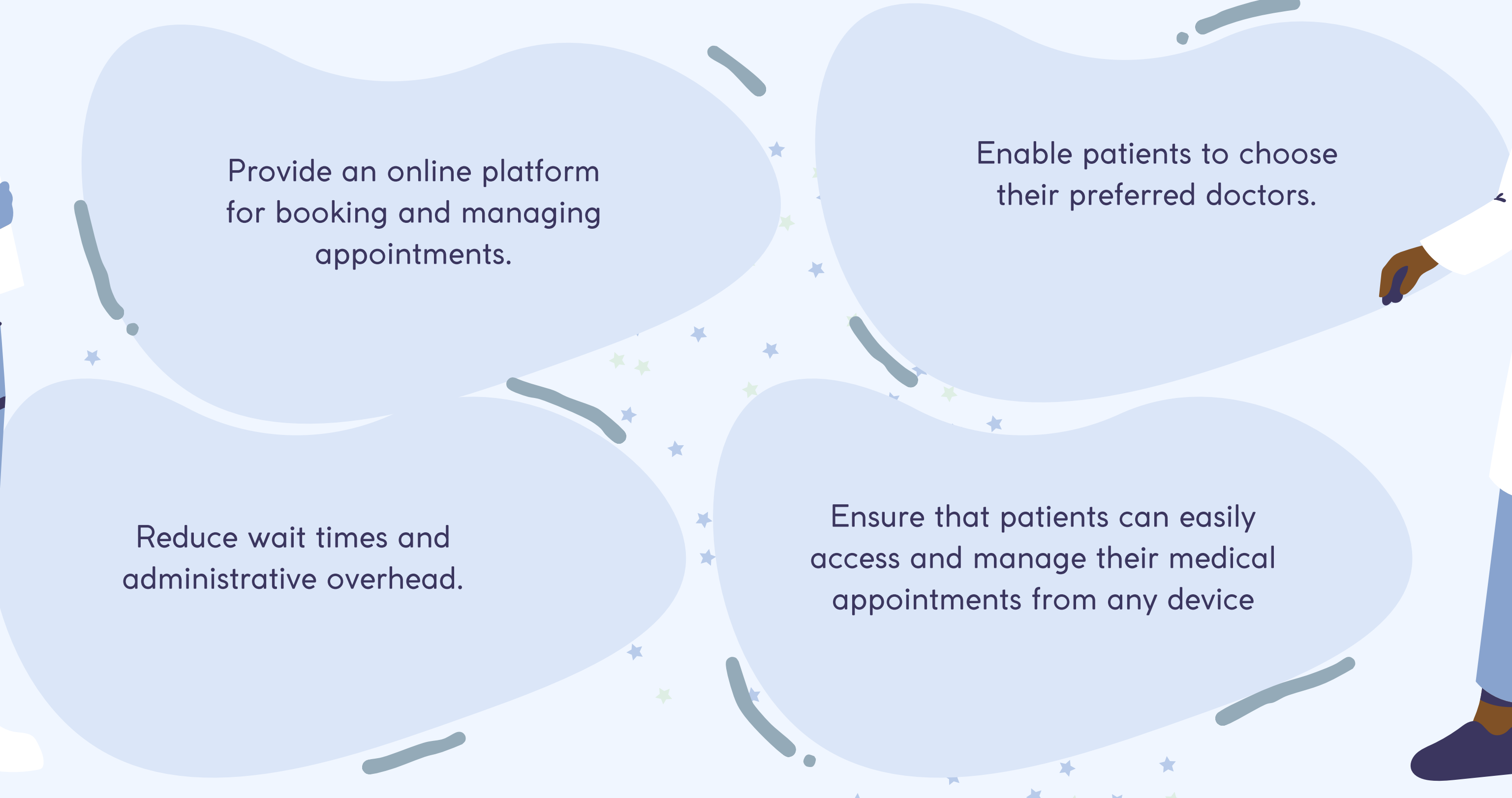
Introduction

What problem does our app solve ?

Traditional appointment scheduling in hospitals is often cumbersome and time-consuming, leading to long wait times and administrative inefficiencies DoctorCare simplifies the process by providing a seamless, online platform for patients to book, manage, and track their healthcare appointments.

Objectives

Provide an online platform for booking and managing appointments.

Enable patients to choose their preferred doctors.

Reduce wait times and administrative overhead.

Ensure that patients can easily access and manage their medical appointments from any device

Features

The features that our app provides are :

Feature 01

Users can easily view available time slots and book appointments with their preferred doctors.
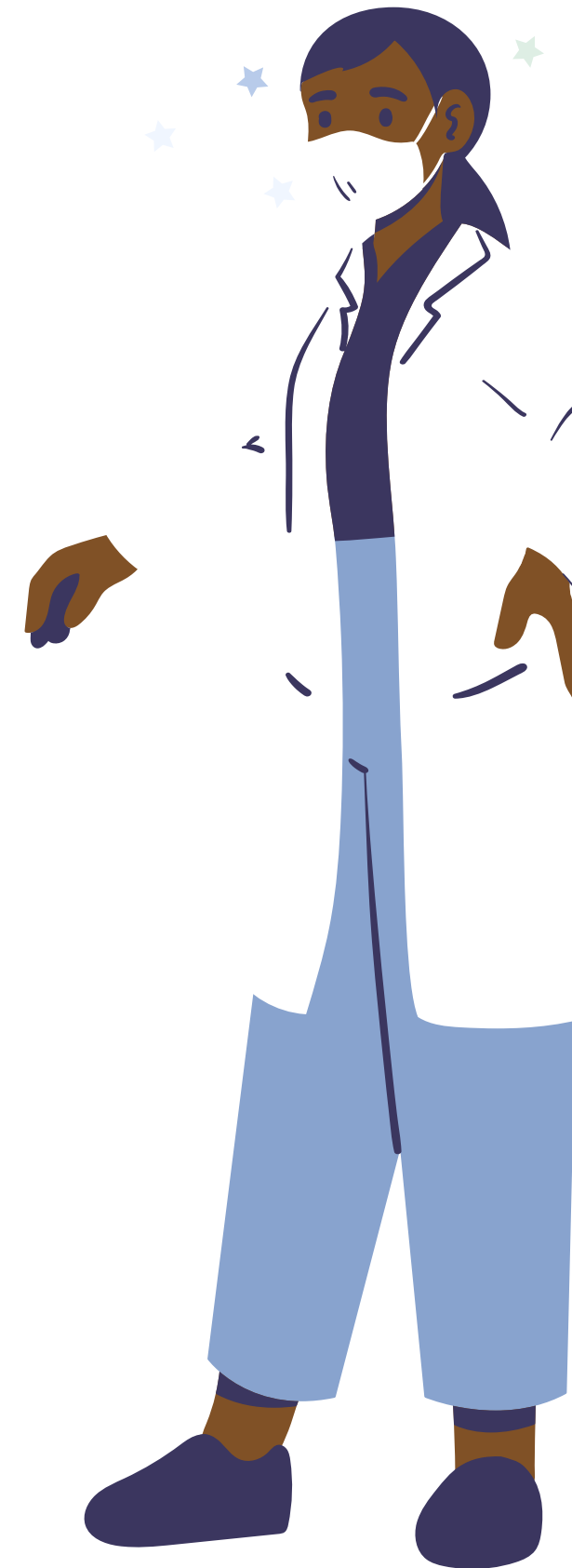
Feature 02

Detailed profiles of doctors, including their specialties, qualifications, and patient reviews

Feature 03

Patients can leave reviews and rate their experience, aiding others in selecting the best healthcare providers.

Target Users

Patients
Need an easy and efficient way to book and manage appointments

Doctors
Require a system to manage their schedules and communicate with patients

Staff
Need tools to oversee appointment bookings and cancellations

Technology Stack

The MERN stack includes MongoDB, Express.js, React.js, and Node.js. MongoDB provides flexible, scalable data storage. Express.js builds robust server-side applications. React.js creates dynamic, responsive user interfaces. Node.js powers the back end with efficient JavaScript execution. This stack enables full-stack development with JavaScript, ensuring smooth integration between front-end and back-end.
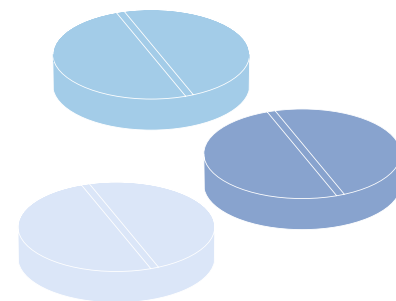
UML Diagrams

Why make UML Diagrams ?

UML diagrams improve communication, aid in planning and design, and help identify potential issues early in development.
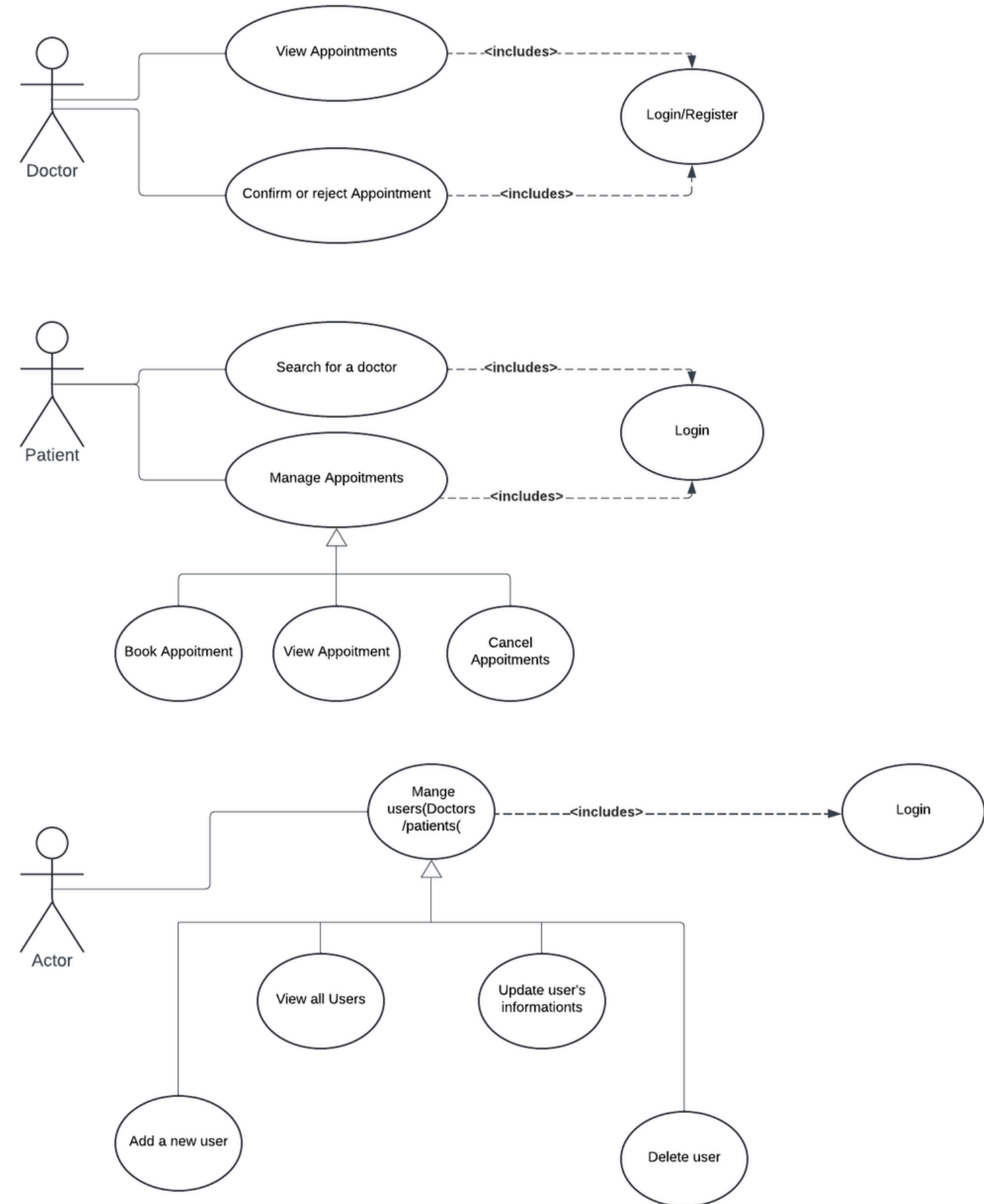
Use case diagrams clarify user interactions and system functionality. **Class diagrams** show the structure, attributes, methods, and relationships of system classes.

Use Case diagram

Doctor
- View Appointments --- <includes> ---> Login/Register
- Confirm or reject Appointment --- <includes> ---> Login/Register

Patient
- Search for a doctor --- <includes> ---> Login
- Manage Appoitments --- <includes> ---> Login
  - Book Appoitment
  - View Appoitment
  - Cancel Appoitments

Actor
- Mange users(Doctors/patients(--- <includes> ---> Login
  - View all Users
  - Update user's informationts
  - Add a new user
  - Delete user

Class Diagram

Doctor
- id : int
- first_name : String
- last_name : String
- age : int
- gender : String
- address : String
- phone_number : String
- cin: string
- specialization: String
- available: boolean

Appointement
- id : int
- date: date
- diagnosis : String

Patient
- id : int
- first_name : String
- last_name : String
- age : int
- gender : String
- address : String
- phone_number : String
- cin: string
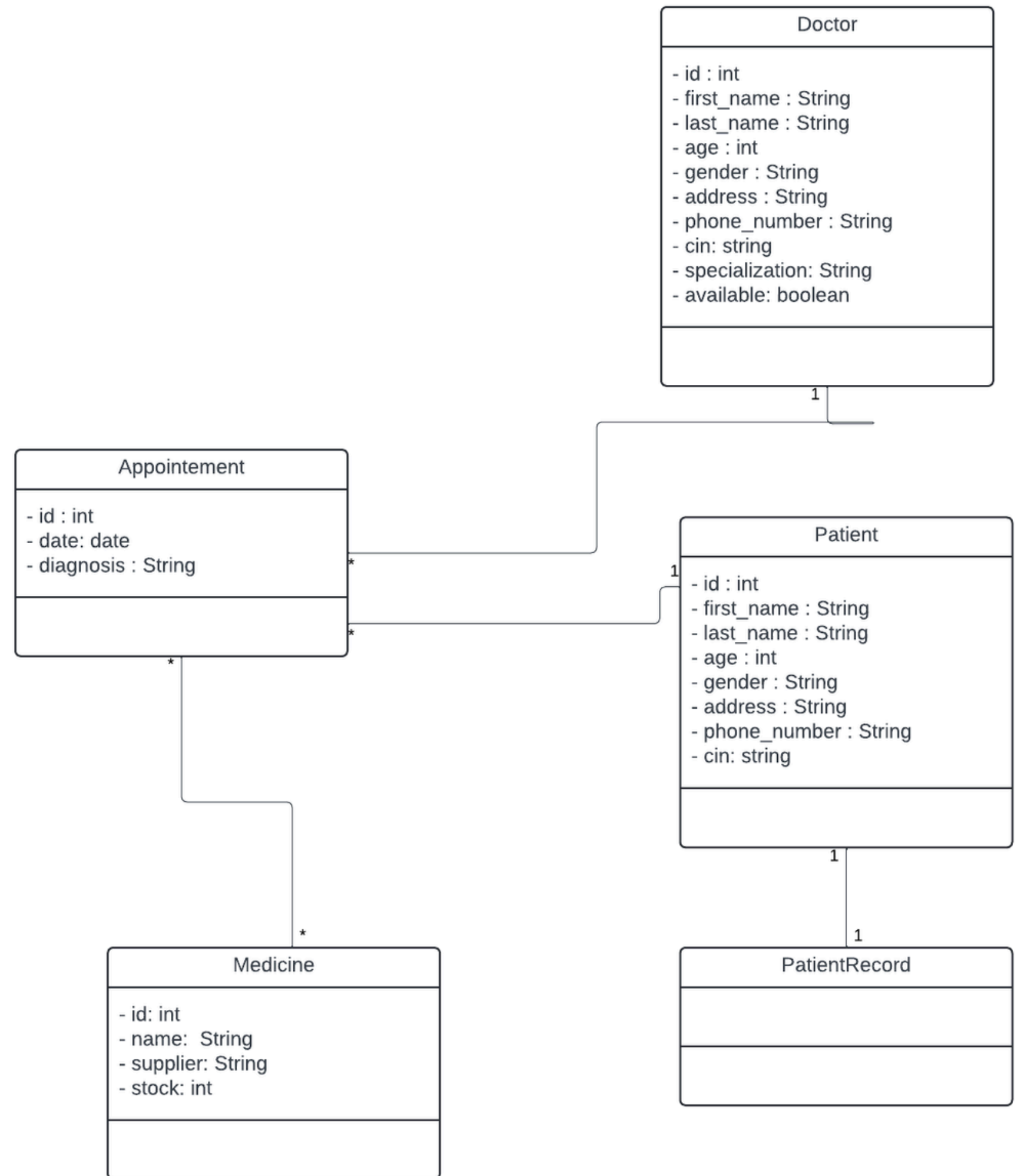
Medicine
- id: int
- name: String
- supplier: String
- stock: int

PatientRecord

Figma link

Explore the visual blueprint of our innovative DoctoreCare app by clicking on the following link.

[DoctorCare Figma design](#)

Future Enhancments

Health Records
Integration of electronic health records

Payment Integration
Adding secure payment gateways for online consultations.

Telemedicine
Video consultation feature

Demo

[Github Repo Link](#)

Conclusion

Our app streamlines the appointment process, reduces wait times, and improves patient-doctor communication. The app enhances the healthcare experience by making it more accessible and efficient for both patients and doctors.