HRS

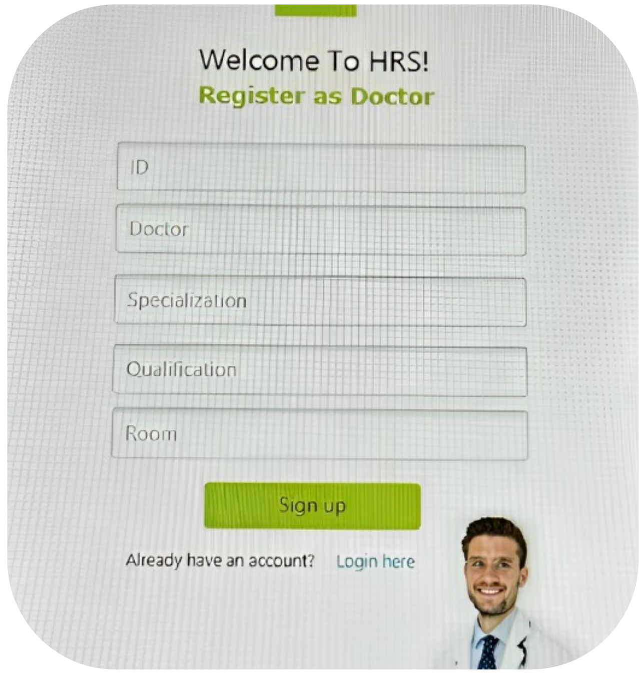
PRESENTATION

Hospital Reservation System

Team Members:

Aiperi Zhenishova
Aizirek Ibraimova
Aigerim Nuralieva
Aigerim Naizabekova

HRS



Welcome To HRS!
Register as Doctor

ID

Doctor

Specialization

Qualification

Room

Sign up

Already have an account? Login here

About HRS

Hospital Reservation System(HRS) - is an innovative healthcare solution revolutionizing patient-doctor interactions. With a modular codebase, database integration, and a vision for future enhancements, HRS is poised to redefine healthcare management.

Significance:

- Improve patient experience
- Optimize resource allocation
- Ensure timely appointments

Our Goal:

Enhance healthcare service efficiency through a streamlined reservation system.

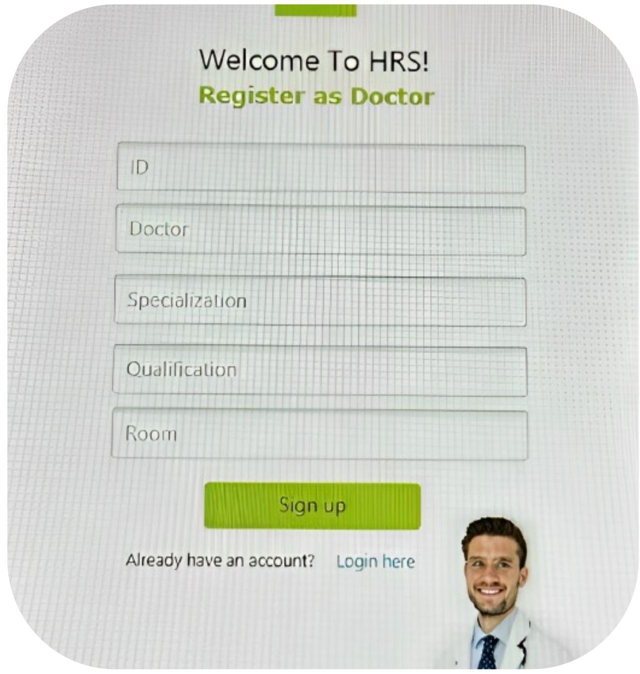
HRS?

This is an online appointment with a specialist, in which the admin books a patient for an appointment by his id, assigning him to a doctor by id and selecting a date.

HRS

About HRS

Hospital Reservation System(HRS) - is an innovative healthcare solution revolutionizing patient-doctor interactions. With a modular codebase, database integration, and a vision for future enhancements, HRS is poised to redefine healthcare management.



Welcome To HRS!
Register as Doctor

ID
Doctor
Specialization
Qualification
Room

Sign up

Already have an account? Login here

Significance:

- Improve patient experience
- Optimize resource allocation
- Ensure timely appointments

Our Goal:

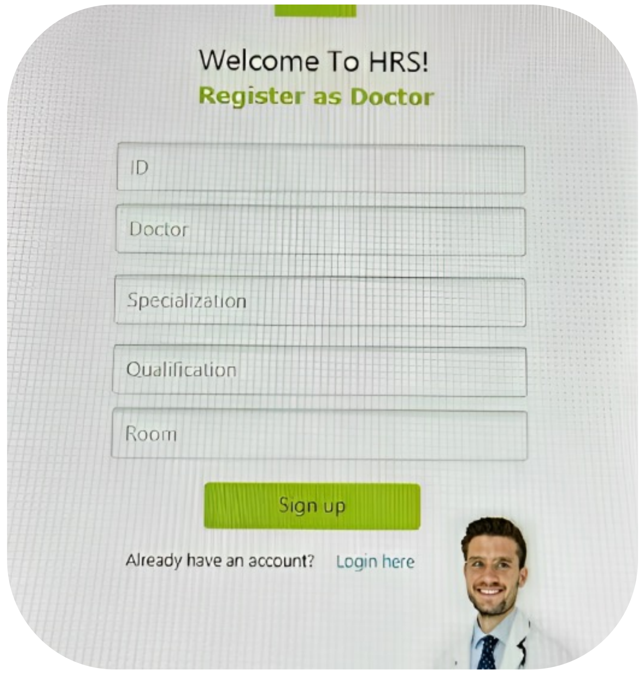
Enhance healthcare service efficiency through a streamlined reservation system.

HRS?

This is an online appointment with a specialist, in which the admin books a patient for an appointment by his id, assigning him to a doctor by id and selecting a date.

About HRS

Hospital Reservation System(HRS) - is an innovative healthcare solution revolutionizing patient-doctor interactions. With a modular codebase, database integration, and a vision for future enhancements, HRS is poised to redefine healthcare management.



Welcome To HRS!

Register as Doctor

ID

Doctor

Specialization

Qualification

Room

Already have an account? Login here

Significance:

- Improve patient experience
- Optimize resource allocation
- Ensure timely appointments

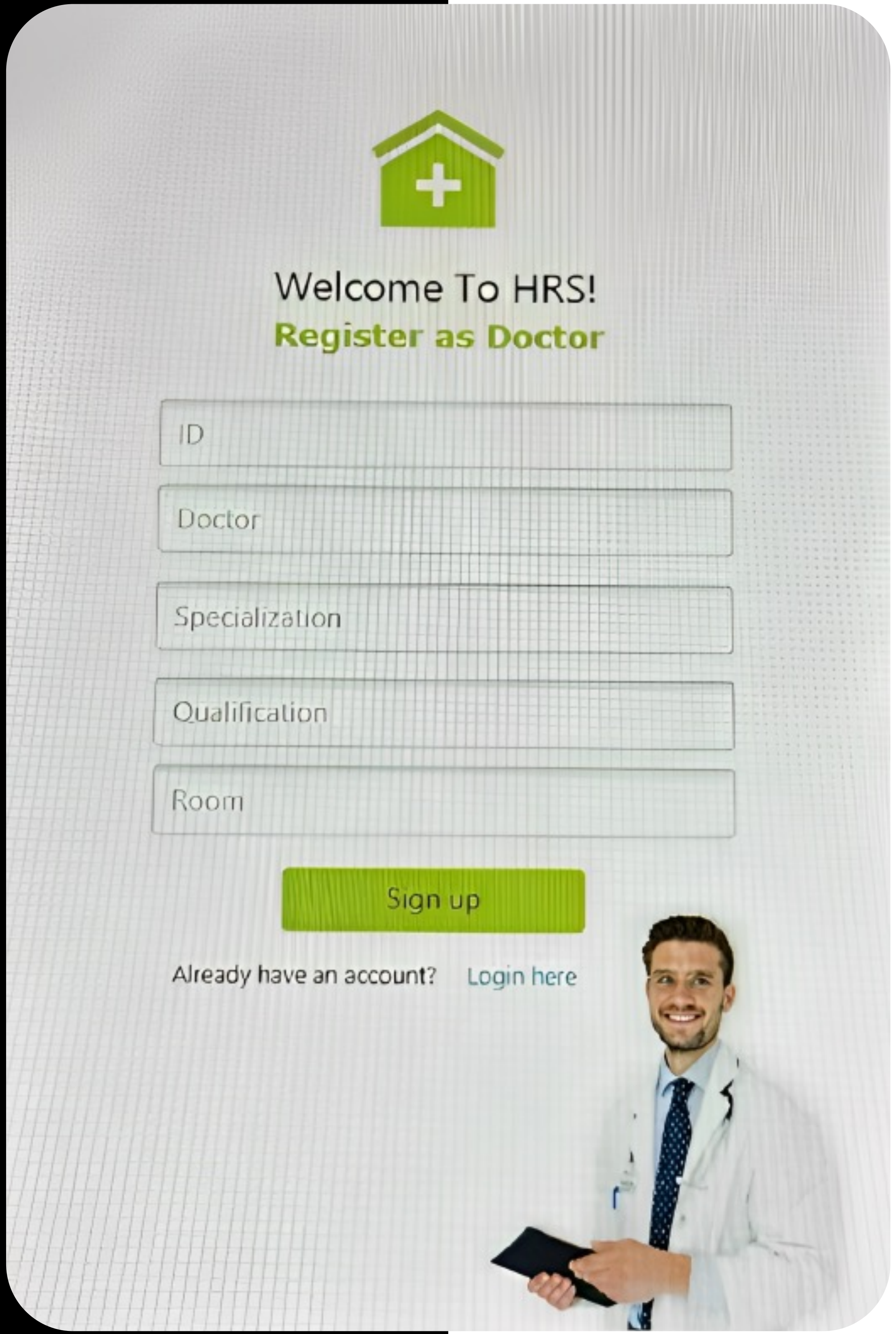
Our Goal:

Enhance healthcare service efficiency through a streamlined reservation system.

HRS?

This is an online appointment with a specialist, in which the admin books a patient for an appointment by his id, assigning him to a doctor by id and selecting a date.

HRS



Welcome To HRS!

Register as Doctor

ID

Doctor

Specialization

Qualification

Room

Already have an account? Login here

Requirements

User Registration and Login

- Patients and doctors are able to register and log in securely.

Patient Management:

- Patients can add their information, such as name, surname, phone number, reason for visiting, and specify a preferred room.
- Patients can view their information, including past appointments.

Doctor Management:

- Doctors can view their schedule and patient appointments.
- The system display a list of all doctors, their specializations, qualifications, and assigned rooms.

Reservation system(Appointment):

- Patients can book appointments with available doctors.

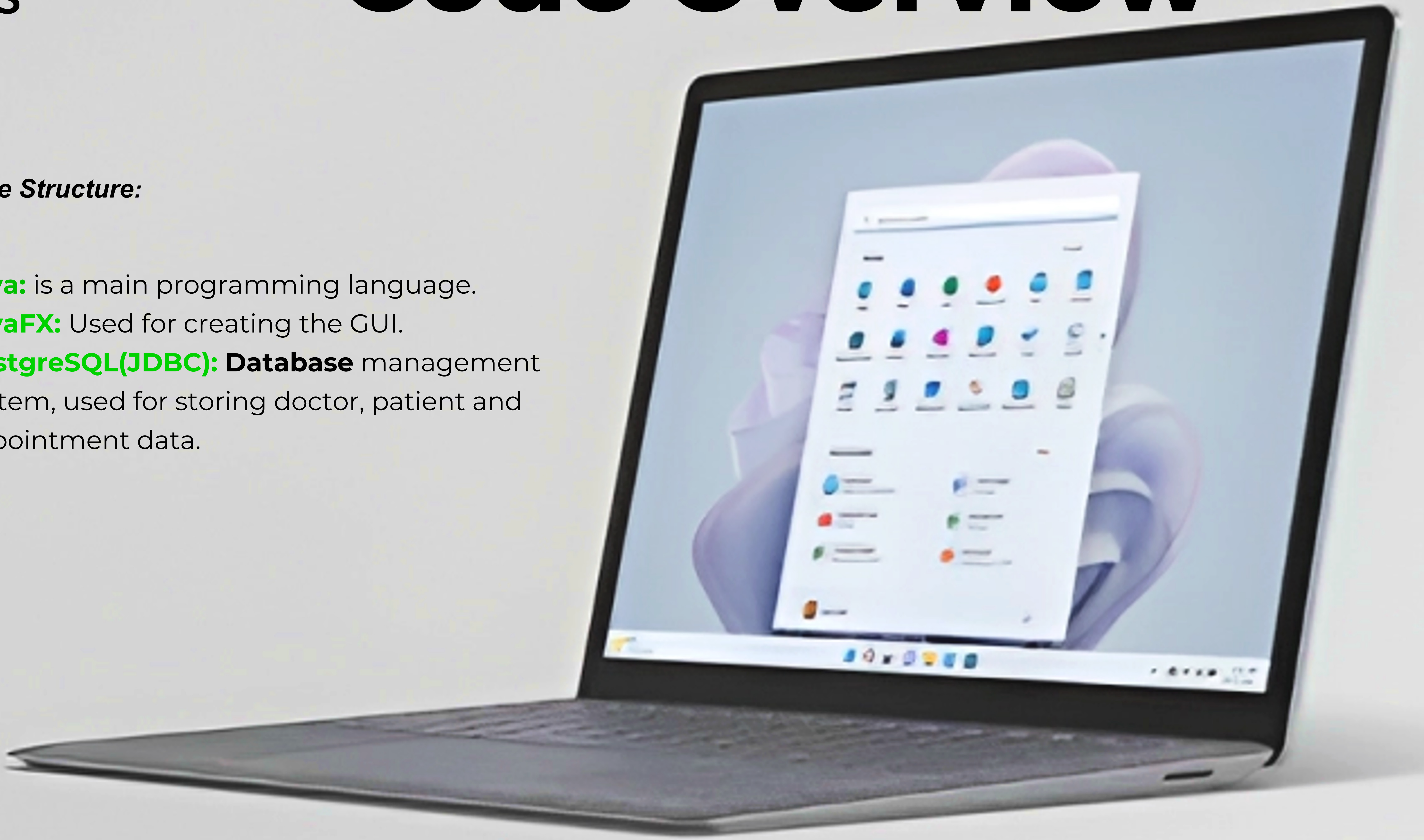
Code Overview

HRS

Code Overview

Code Structure:

- **Java:** is a main programming language.
- **JavaFX:** Used for creating the GUI.
- **PostgreSQL(JDBC): Database** management system, used for storing doctor, patient and appointment data.



HRS

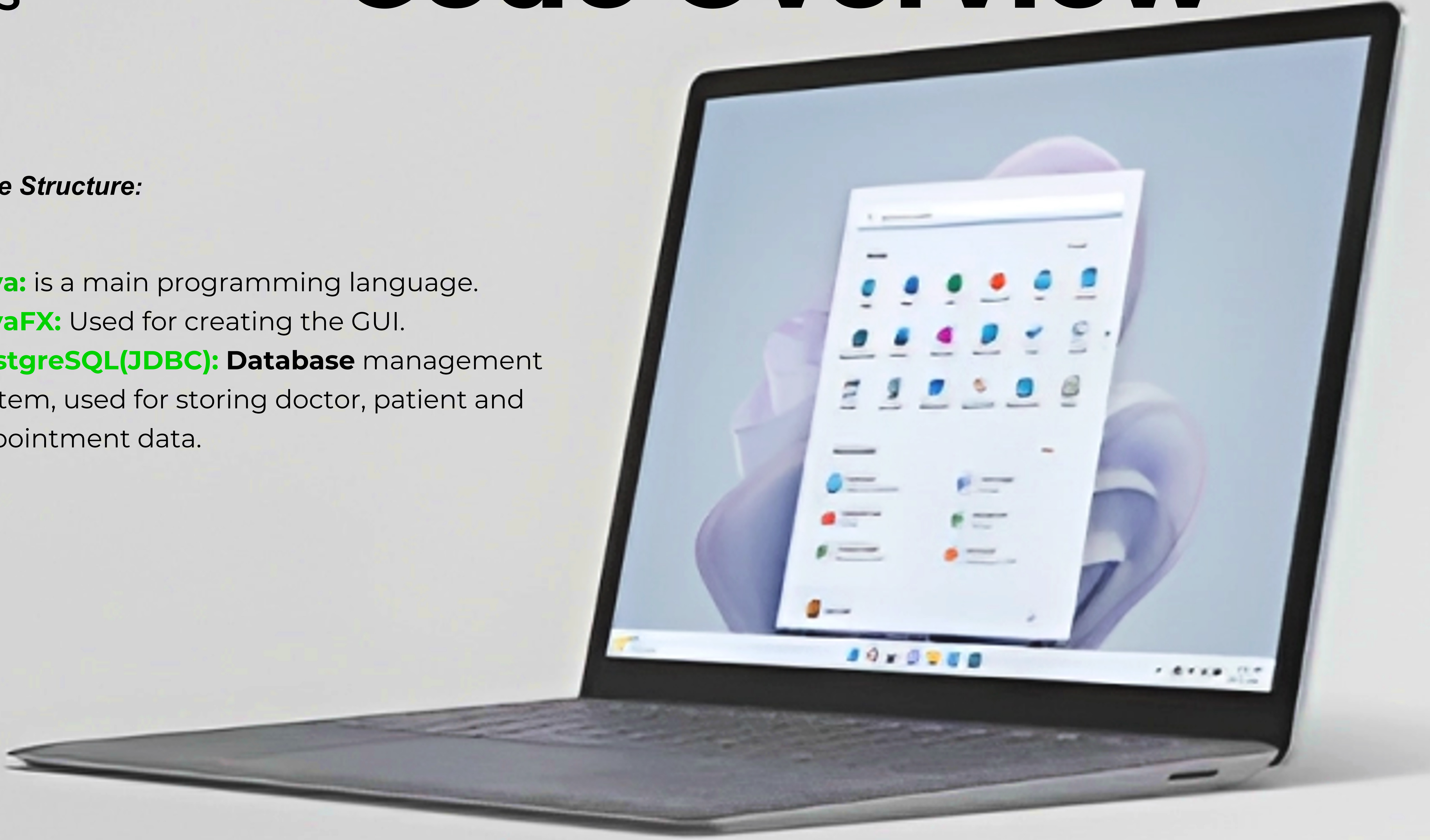
Code Overview

Code Structure:

Java: is a main programming language.

JavaFX: Used for creating the GUI.

PostgreSQL(JDBC): Database management system, used for storing doctor, patient and appointment data.



HRS

Code Overview

Main Modules/Components*:*

HelloApplication

HelloController

HRS(Main Class) class:

- . Processes user input, directs data flow within the application, and integrates patient and doctor functions.
- Includes a method for booking appointments and checking doctor availability.

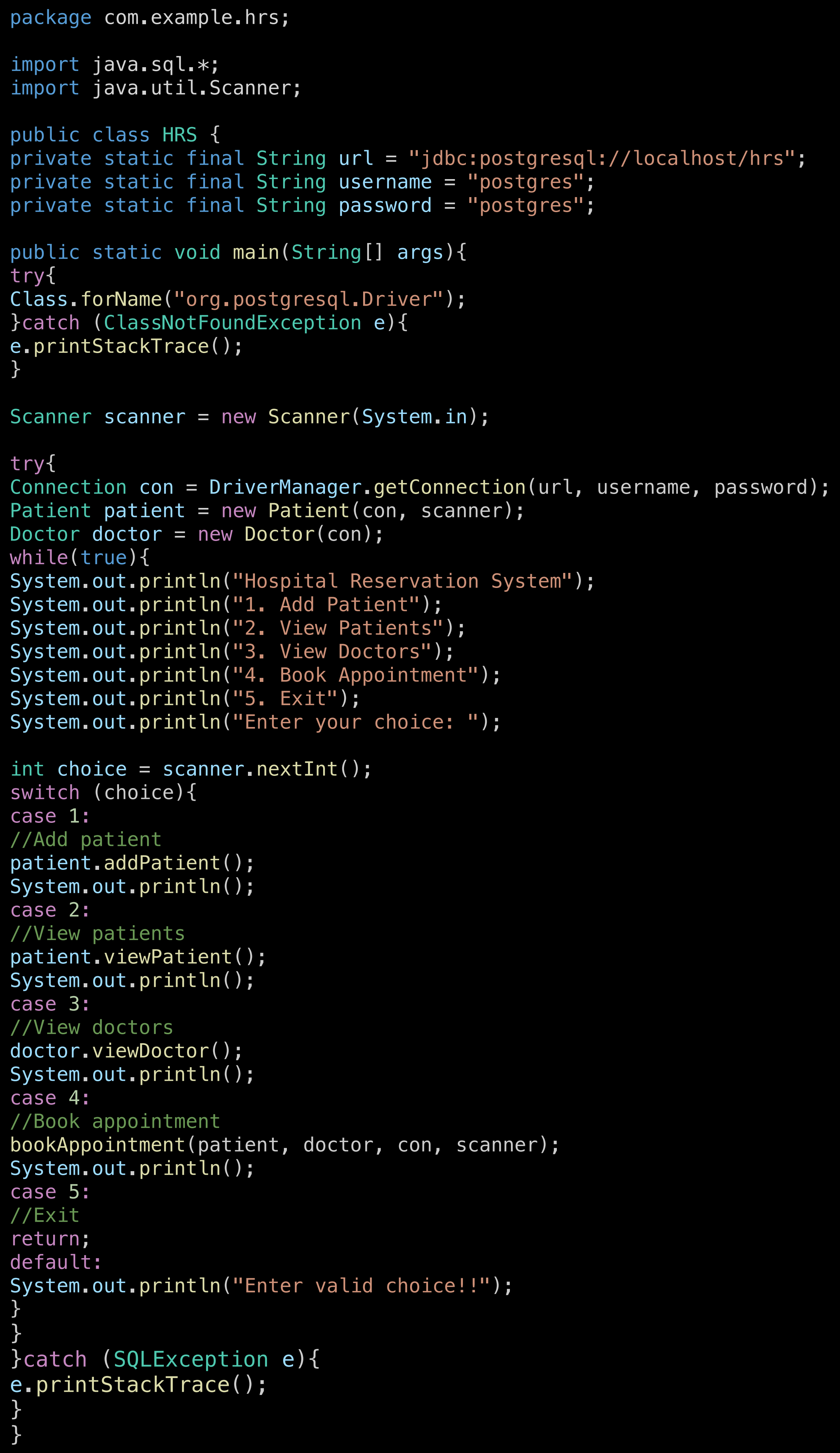
Doctors class:

- Manages viewing all doctors and checking the availability of a specific doctor.

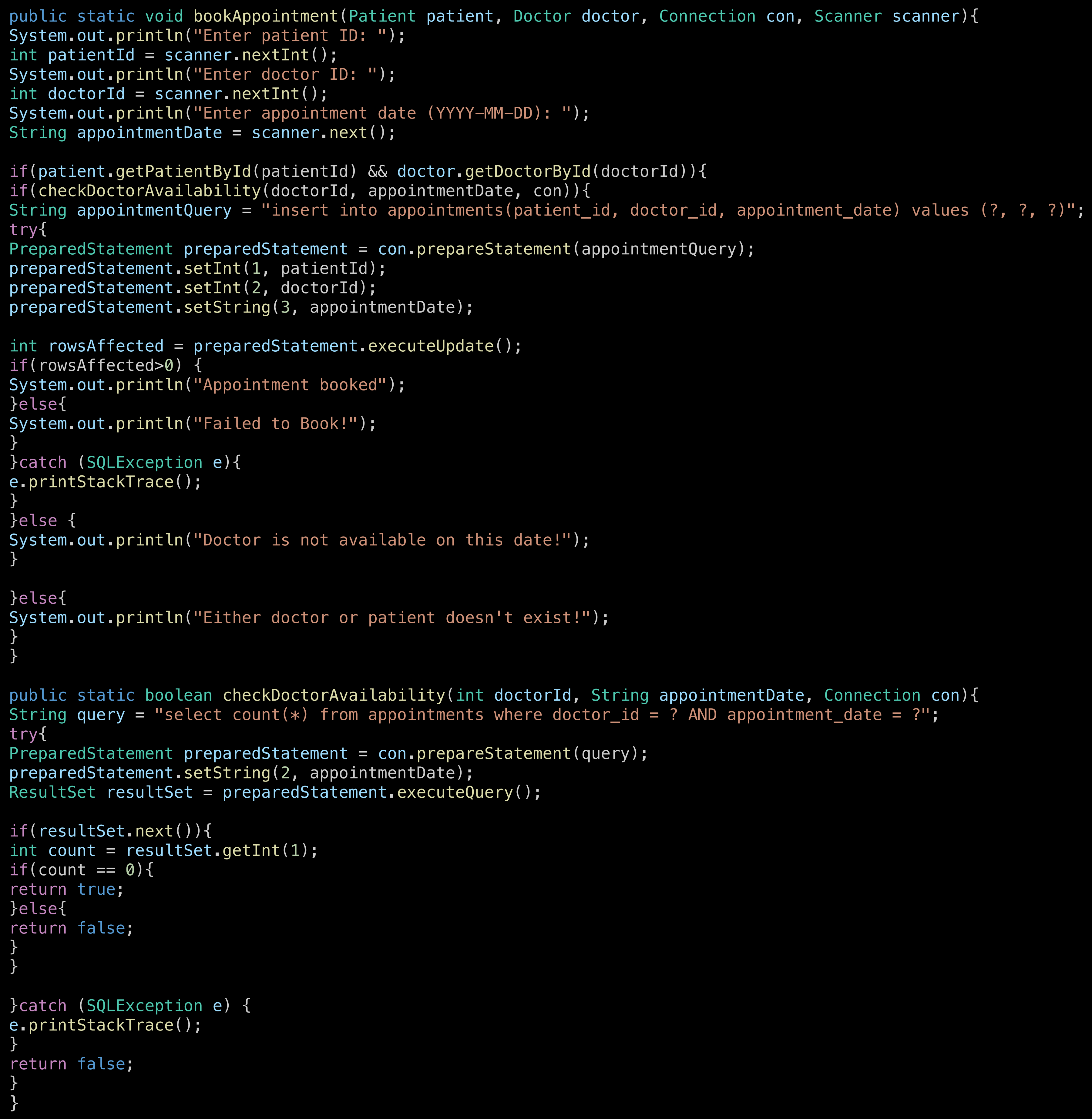
Patients class:

- Functionality is adding patients, viewing patient info and checking the existence of patient ID.

HRS class



```java
package com.example.hrs;

import java.sql.*;
import java.util.Scanner;

public class HRS {
private static final String url = "jdbc:postgresql://localhost/hrs";
private static final String username = "postgres";
private static final String password = "postgres";

public static void main(String[] args){
try{
Class.forName("org.postgresql.Driver");
}catch (ClassNotFoundException e){
e.printStackTrace();
}

Scanner scanner = new Scanner(System.in);

try{
Connection con = DriverManager.getConnection(url, username, password);
Patient patient = new Patient(con, scanner);
Doctor doctor = new Doctor(con);
while(true){
System.out.println("Hospital Reservation System");
System.out.println("1. Add Patient");
System.out.println("2. View Patients");
System.out.println("3. View Doctors");
System.out.println("4. Book Appointment");
System.out.println("5. Exit");
System.out.println("Enter your choice: ");

int choice = scanner.nextInt();
switch (choice){
case 1:
//Add patient
patient.addPatient();
System.out.println();
case 2:
//View patients
patient.viewPatient();
System.out.println();
case 3:
//View doctors
doctor.viewDoctor();
System.out.println();
case 4:
//Book appointment
bookAppointment(patient, doctor, con, scanner);
System.out.println();
case 5:
//Exit
return;
default:
System.out.println("Enter valid choice!!");
}
}
}catch (SQLException e){
e.printStackTrace();
}
}
```



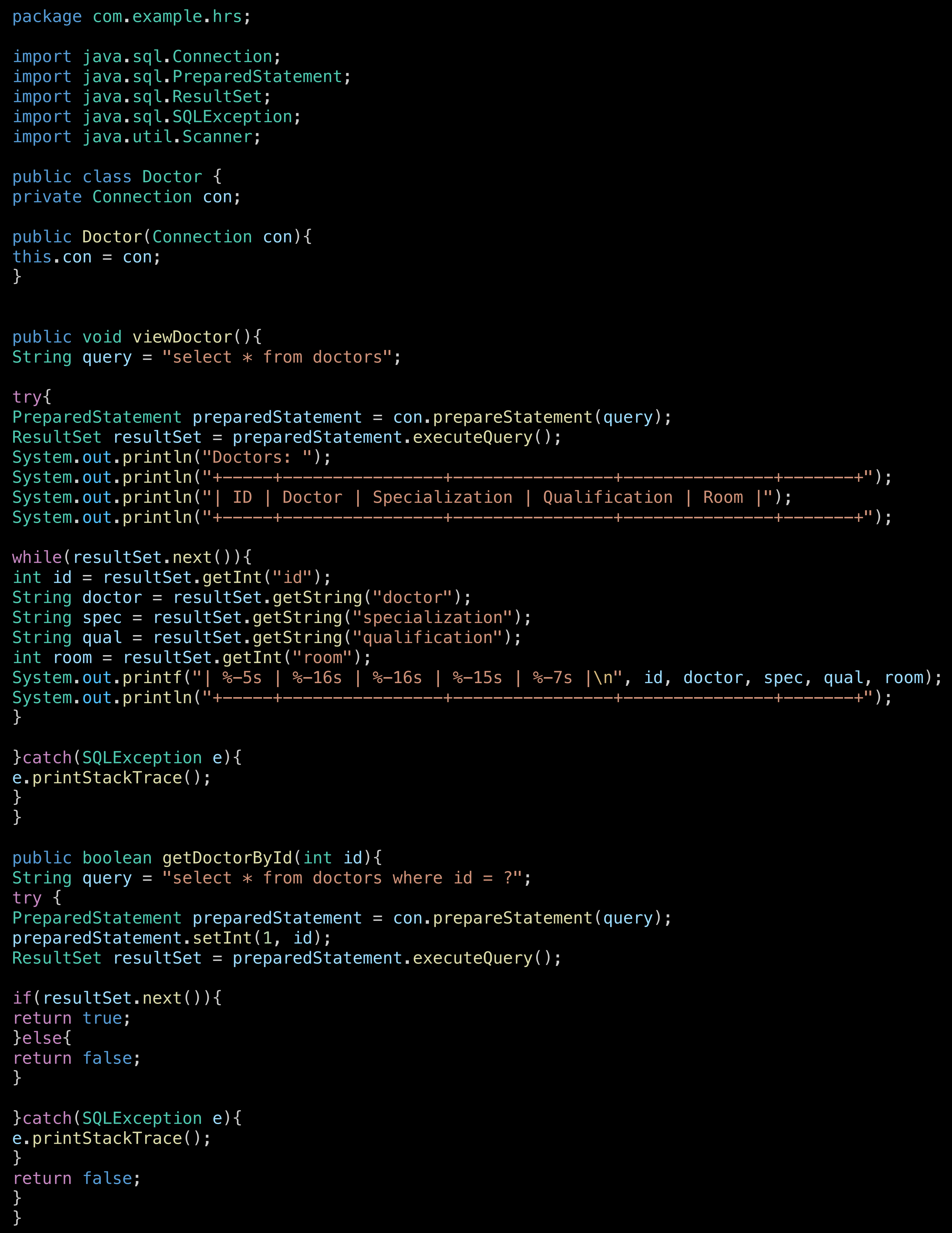
```java
public static void bookAppointment(Patient patient, Doctor doctor, Connection con, Scanner scanner){
System.out.println("Enter patient ID: ");
int patientId = scanner.nextInt();
System.out.println("Enter doctor ID: ");
int doctorId = scanner.nextInt();
System.out.println("Enter appointment date (YYYY-MM-DD): ");
String appointmentDate = scanner.next();

if(patient.getPatientById(patientId) && doctor.getDoctorById(doctorId)){
if(checkDoctorAvailability(doctorId, appointmentDate, con)){
String appointmentQuery = "insert into appointments(patient_id, doctor_id, appointment_date) values (?, ?, ?)";
try{
PreparedStatement preparedStatement = con.prepareStatement(appointmentQuery);
preparedStatement.setInt(1, patientId);
preparedStatement.setInt(2, doctorId);
preparedStatement.setString(3, appointmentDate);

int rowsAffected = preparedStatement.executeUpdate();
if(rowsAffected>0) {
System.out.println("Appointment booked");
}else{
System.out.println("Failed to Book!");
}
}catch (SQLException e){
e.printStackTrace();
}
}else {
System.out.println("Doctor is not available on this date!");
}

}else{
System.out.println("Either doctor or patient doesn't exist!");
}
}

public static boolean checkDoctorAvailability(int doctorId, String appointmentDate, Connection con){
String query = "select count(*) from appointments where doctor_id = ? AND appointment_date = ?";
try{
PreparedStatement preparedStatement = con.prepareStatement(query);
preparedStatement.setString(2, appointmentDate);
ResultSet resultSet = preparedStatement.executeQuery();

if(resultSet.next()){
int count = resultSet.getInt(1);
if(count == 0){
return true;
}else{
return false;
}
}

}catch (SQLException e) {
e.printStackTrace();
}
return false;
}
}
```

Patient class

```java
package com.example.hrs;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class Patient {
private Connection con;
private Scanner scanner;

public Patient(Connection con, Scanner scanner){
this.con = con;
this.scanner = scanner;
}

public void addPatient(){
System.out.println("Enter patient name: ");
String name = scanner.next();
System.out.println("Enter patient surname: ");
String surname = scanner.next();
System.out.println("Enter patient phone number: ");
int p_number = scanner.nextInt();
System.out.println("Enter patient reason of visiting: ");
String reason = scanner.next();
System.out.println("Specify patient room: ");
int room = scanner.nextInt();

try{
String query =
"INSERT INTO patients(name, surname, phone number, reason, room) VALUES (?, ?, ?,
?, ?)";
PreparedStatement preparedStatement = con.prepareStatement(query);
preparedStatement.setString(1, name);
preparedStatement.setString(2, surname);
preparedStatement.setInt(3, p_number);
preparedStatement.setString(4, reason);
preparedStatement.setInt(5, room);

int affectedRows = preparedStatement.executeUpdate();
if(affectedRows>0){
System.out.println("Patient added succesfully!");
}else{
System.out.println("Failed to add Patient");
}
}catch(SQLException e){
e.printStackTrace();
}
}
```

```java
public void viewPatient(){
String query = "select * from patients";

try{
PreparedStatement preparedStatement = con.prepareStatement(query);
ResultSet resultSet = preparedStatement.executeQuery();
System.out.println("Patients: ");
System.out.println("+-----+--------------+-------------+-------------+-------------+------+");
System.out.println("| ID | Name | Surname | Phone number| Reason | Room ");
System.out.println("+-----+--------------+-------------+-------------+-------------+------+");

while(resultSet.next()){
int id = resultSet.getInt("id");
String name = resultSet.getString("name");
String surname = resultSet.getString("surname");
int p_number = resultSet.getInt("phone number");
String reason = resultSet.getString("reason");
int room = resultSet.getInt("room");
System.out.printf("| %-5s | %-14s | %-13s | %-13s | %-13s | %-7s |");
System.out.println("+-----+--------------+-------------+-------------+-------------+------+");
}

}catch(SQLException e){
e.printStackTrace();
}
}

public boolean getPatientById(int id){
String query = "select * from patients where id = ?";
try {
PreparedStatement preparedStatement = con.prepareStatement(query);
preparedStatement.setInt(1, id);
ResultSet resultSet = preparedStatement.executeQuery();

if(resultSet.next()){
return true;
}else{
return false;
}

}catch(SQLException e){
e.printStackTrace();
}
return false;
}
}
```

Doctor class



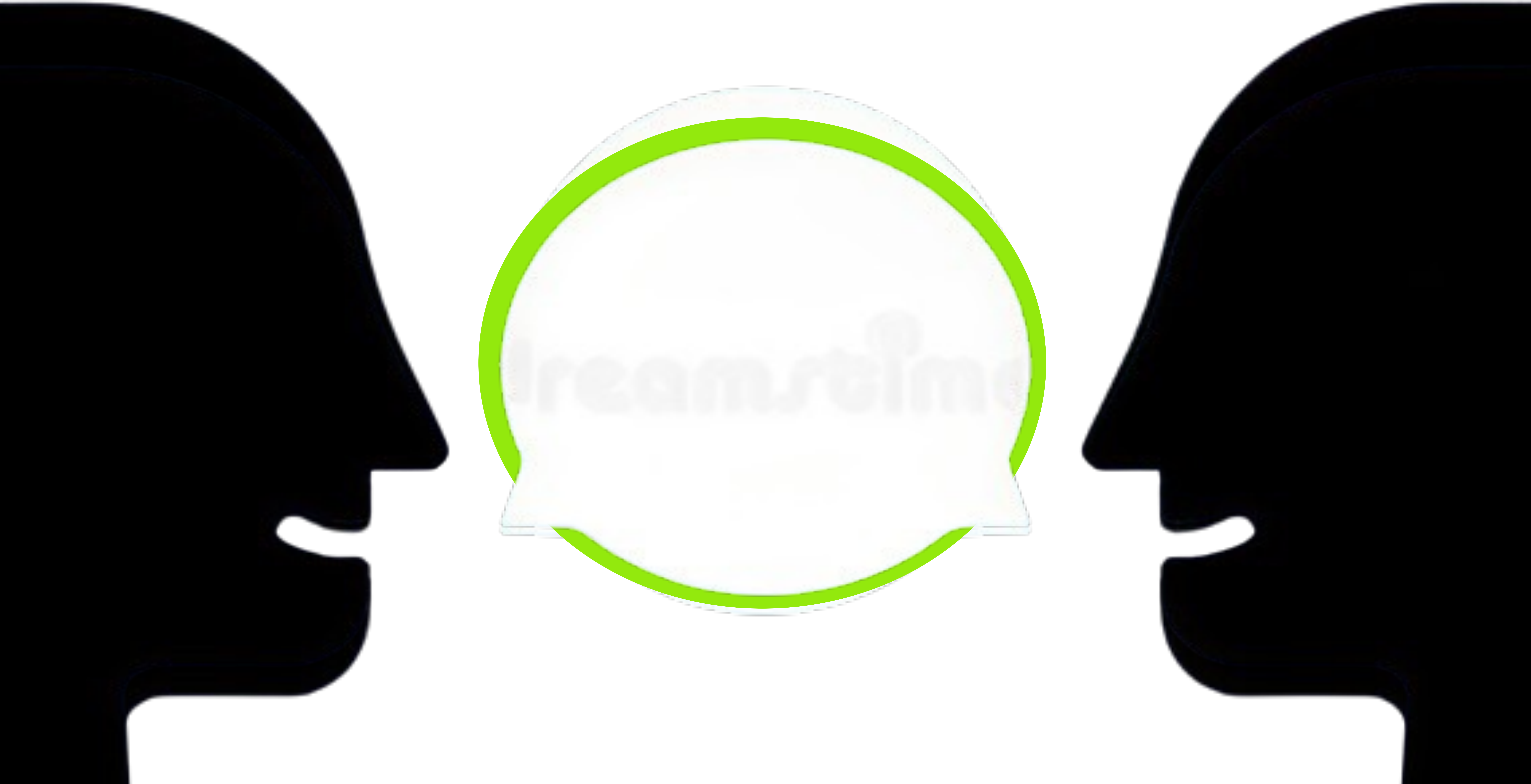
```
package com.example.hrs;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class Doctor {
private Connection con;

public Doctor(Connection con){
this.con = con;
}


public void viewDoctor(){
String query = "select * from doctors";

try{
PreparedStatement preparedStatement = con.prepareStatement(query);
ResultSet resultSet = preparedStatement.executeQuery();
System.out.println("Doctors: ");
System.out.println("+-----+----------------+----------------+---------------+-------+");
System.out.println("| ID | Doctor | Specialization | Qualification | Room |");
System.out.println("+-----+----------------+----------------+---------------+-------+");

while(resultSet.next()){
int id = resultSet.getInt("id");
String doctor = resultSet.getString("doctor");
String spec = resultSet.getString("specialization");
String qual = resultSet.getString("qualification");
int room = resultSet.getInt("room");
System.out.printf("| %-5s | %-16s | %-16s | %-15s | %-7s |\n", id, doctor, spec, qual, room);
System.out.println("+-----+----------------+----------------+---------------+-------+");
}

}catch(SQLException e){
e.printStackTrace();
}
}

public boolean getDoctorById(int id){
String query = "select * from doctors where id = ?";
try {
PreparedStatement preparedStatement = con.prepareStatement(query);
preparedStatement.setInt(1, id);
ResultSet resultSet = preparedStatement.executeQuery();

if(resultSet.next()){
return true;
}else{
return false;
}

}catch(SQLException e){
e.printStackTrace();
}
return false;
}
}
```

HRS

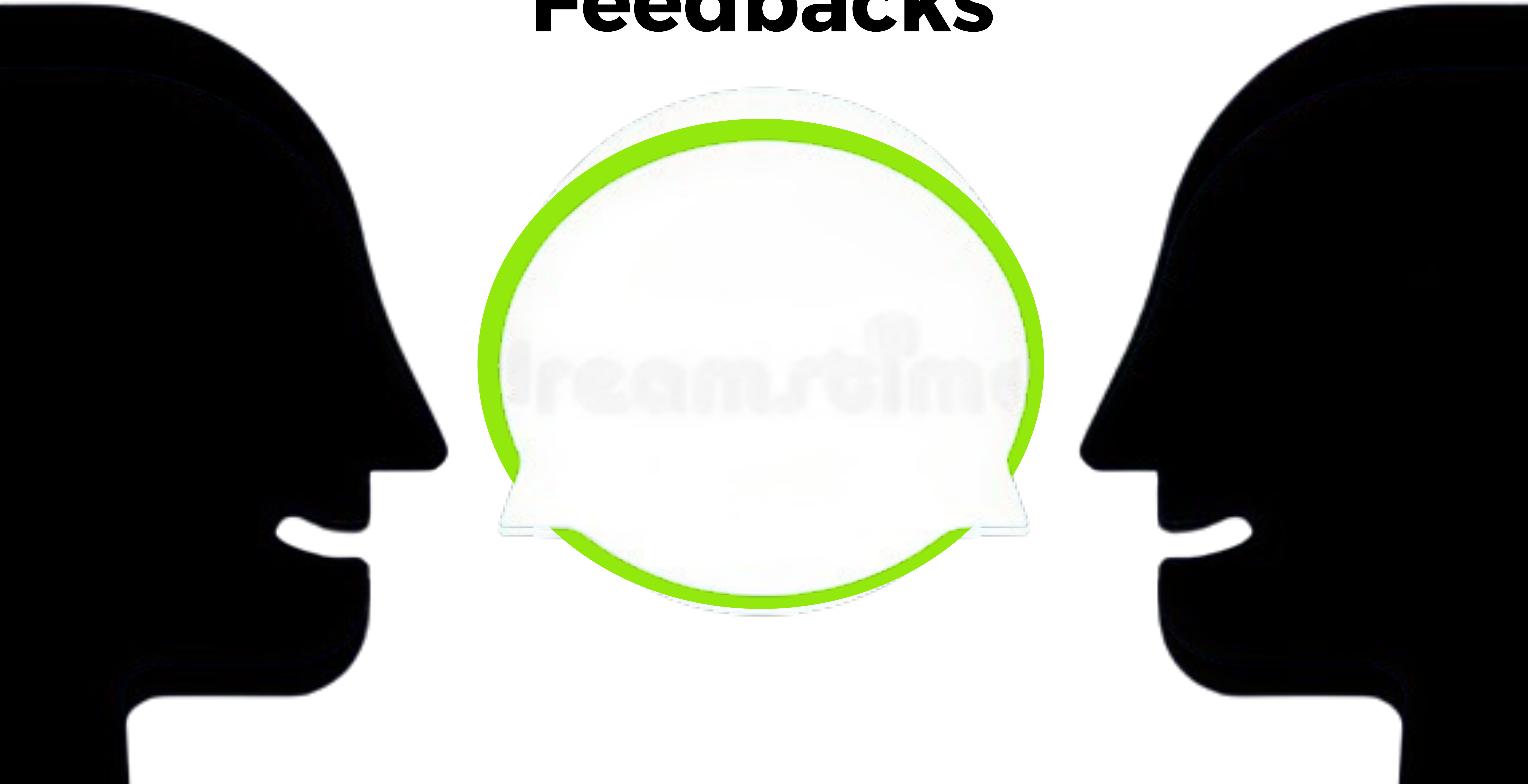
Code Overview

Further plans for Improving:

- Developing a **mobile version** of the application to increase accessibility for both patients and doctors.
- **Localization** features for users with different regions(to make available everywhere).
- Assigning different users like (**admin, receptionist**) with different level of access and permissions.
- Enabling a **calendar** view for easier date selection.
- Implementing **reminders and notifications** for patients about upcoming appointments.



Feedbacks

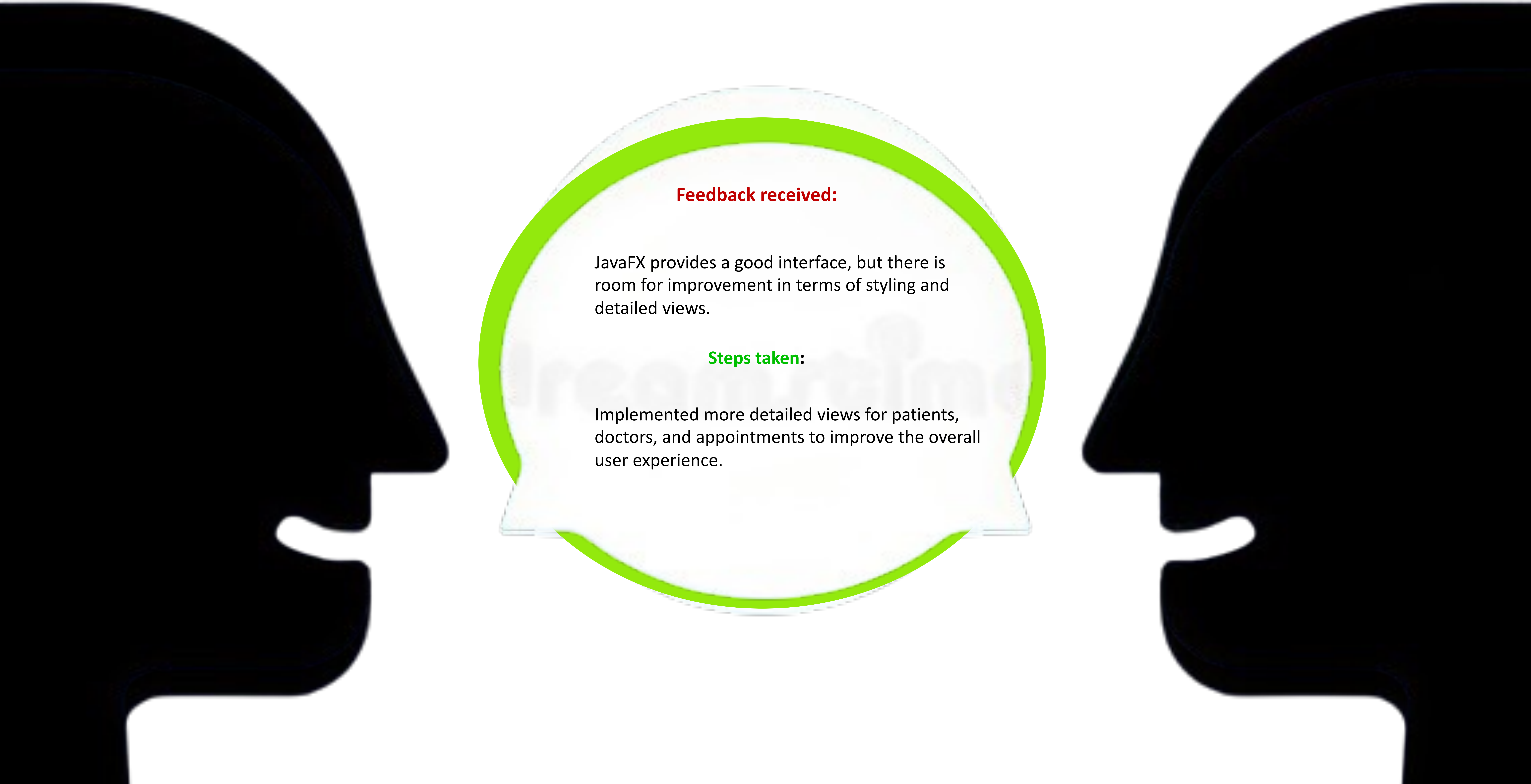


Feedback received:

Code structure is clear, but some sections lack comments, affecting readability.

Steps taken:

Implemented additional comments and documentation in areas where code clarity was lacking..



Feedback received:

JavaFX provides a good interface, but there is room for improvement in terms of styling and detailed views.

Steps taken:

Implemented more detailed views for patients, doctors, and appointments to improve the overall user experience.

Feedback received:

Limited error handling was observed, and the application could be enhanced to provide more informative error messages to users.

Steps taken:

Implemented more robust error handling mechanisms throughout the code, especially in user input scenarios.

HRS

Conclusion

. Let’s highlight the key takeaways:

- Successful implementation of a Hospital Reservation System using Java and PostgreSQL.
- The system provides functionality for adding patients, viewing patients and doctors, and booking appointments
- Utilization of JavaFX for the user interface, enhancing the user experience.
- Effective integration of a relational database (PostgreSQL) for storing and managing data.
- Continuous improvement through feedback, resulting in a more robust and user-friendly system.

With enhanced error handling, a more intuitive appointment booking process, and plans for additional features, HRS is not only user-friendly but also adaptive. The journey doesn’t end here: it’s a commitment to continuous improvement and a vision for a healthcare solution that truly meets the expectations of its users.

Phone
+123-456-7890

Address
22 Ankara St., Bishkek

Website
www.HRS@gmail.com

HRS

Thank You

Fundamentally, a hospital booking system is not just a technology solution; it is a catalyst for positive change in healthcare management. Using technology, we strive to improve the patient experience, increase the efficiency of healthcare providers and contribute to the overall improvement of healthcare services. Thank you for your attention. We are open to any questions and feedback.

www.HRS@gmail.com