connect: Establishes a connection to the database using the JDBC DriverManager. Initializes the con (Connection) and stmt (Statement) objects.

close: Closes the stmt (Statement) and con (Connection) to release the database resources.

addNewPatient: Inserts a new patient record into the PATIENT table using the details from the Patient object. The method constructs an SQL INSERT query and executes it.

String query = "INSERT INTO PATIENTS VALUES (" + p.getPatient\_SSN() + ", " + p.getPatient\_ID()

+ ", '" + p.getDate\_Of\_Birth() + "', '" + p.getFirst\_Name() + "', '" +

p.getLast\_Name() + "', '" + p.getEmail() +"', '" + p.getPhone\_Number()

+ "', '" + p.getAddress() + "', '" + p.getGender() + "', '" +

p.getMedical\_History() + "')";

addNewDoctor: Inserts a new doctor record into the DOCTOR table using the details from the Doctor object. The method constructs an SQL INSERT query and executes it.

String query = "INSERT INTO DOCTOR VALUES (" + d.getDoctorId() + ", '" + d.getFirstName()

+ "', '" + d.getLastName() + "', '" + d.getEmail() + "', '" + d.getPhoneNumber()

+ "', '" + d.getTitle() + "', '" + d.getSpecialty() + "', " + d.getStartingYear()

+ ", " + d.getMedicalFacilityId()+ ")";

getDoctorsBySpecialty: Retrieves a list of doctor first names and last names based on a given specialty. Executes an SQL SELECT query and returns the results in an ArrayList.

String query = "SELECT First\_Name, Last\_Name FROM DOCTOR WHERE specialty = " + specialty;

getAllSpecialties: Retrieves a list of all unique specialties from the DOCTOR table. Executes an SQL SELECT query and returns the results in an ArrayList.

SELECT Specialty FROM DOCTOR

String q = "SELECT Specialty FROM DOCTOR";

getApptInfoFromDoctor: Retrieves all appointment information for a doctor based on their first name and last name. Executes an SQL SELECT query and returns the results in an ArrayList of Appointment objects.

String q = "SELECT \* FROM APPOINTMENT WHERE Appointment\_ID IN (SELECT Appointment\_ID FROM IS\_AVAILABLE NATURAL JOIN DOCTOR d WHERE d.First\_Name = '" + fn + "', AND Last\_Name = '" + ln + "';";

WHERE First\_Name = ? AND Last\_Name = ?

)

removeApptFromAvailability: Deletes an appointment from the IS\_AVAILABLE table based on the appointment ID. Executes an SQL DELETE query.

String q = "DELETE FROM IS\_AVAILABLE WHERE Appointment\_ID = " + appointmentId + ";";

addAppttoBookAppt: Inserts a new record into the BOOK\_APPOINTMENT table using the details from the Book\_Appointment object. The method constructs an SQL INSERT query and executes it.

String q = "INSERT INTO BOOK\_APPOINTMENT VALUES (" + a.getPatient\_SSN()

+ ", " + a.getAppointment\_ID() + ", '" + a.getReason() + "';";

getApptFromSSN: Retrieves all appointments associated with a specific patient's SSN. Executes an SQL SELECT query and returns the results in an ArrayList of Appointment objects.

String q = "SELECT \* FROM APPOINTMENT NATURAL JOIN BOOK\_APPOINTMENT WHERE Patient\_SSN = " + ssn + ";";

removeApptFromBooked: Deletes a booked appointment from the BOOK\_APPOINTMENT table based on the appointment ID. Executes an SQL DELETE query.

String q = "DELETE FROM BOOK\_APPOINTMENT WHERE Appointment\_ID = " + chosenAppointment.getAppointmentId() + ";";

addAppttoAvailable: Inserts a new appointment into the IS\_AVAILABLE table using the details from the Appointment object. The method constructs an SQL INSERT query and executes it.

exists: Checks if a patient with a given SSN exists in the PATIENTS table. Executes an SQL SELECT query and returns a boolean result based on the presence of the record.

SELECT \* FROM PATIENTS WHERE Patient\_SSN = ?

createNewInsurancePlan: Inserts a new insurance plan record into the INSURANCE\_PLAN table using the details from the Insurance\_Plan object. The method constructs an SQL INSERT query and executes it.

String q = "INSERT INTO INSURANCE\_PLAN VALUES (" + ip.getInsurancePlanId()

+ ", '" + ip.getCompanyProvider() + "', '" + ip.getInsuranceClass()

+ "', '" + ip.getIssuingDate() + "', '" + ip.getEndDate()

+ "', " + ip.getPatientSSN() + ");";

updateMedicalHistory: Updates the medical history of a patient in the PATIENTS table based on the patient's ID (MRN). Executes an SQL UPDATE query.

String q = "UPDATE PATIENTS SET Medical\_History = '" + MedHis

+ "' WHERE Patient\_ID = " + MRN;

getSSNFromMRN: Retrieves the SSN of a patient based on their medical record number (MRN). Executes an SQL SELECT query and returns the SSN.

String q = "SELECT Patient\_SSN FROM PATIENTS WHERE Patient\_ID = " + MRNs + ";";

int SSNN = 0;

getPatientFromMRN: Retrieves all information of a patient based on their medical record number (MRN). Executes an SQL SELECT query and returns the details in a Patient object.

String q = "SELECT \* FROM PATIENTS WHERE Patient\_ID = " + MRN;

updatePatientBySSN: Updates the details of a patient in the PATIENTS table based on their SSN. The method constructs an SQL UPDATE query and executes it.

String q = "UPDATE PATIENTS SET Patient\_SSN = " + p.getPatient\_SSN()

+ ", Patient\_ID = " + p.getPatient\_ID() + ", Date\_Of\_Birth = '" + p.getDate\_Of\_Birth()

+ "', First\_Name = '" + p.getFirst\_Name() + "', Last\_Name = '" +

p.getLast\_Name() + "', email = '" + p.getEmail() + "', Phone\_Number = '" +

p.getPhone\_Number() + "', Address = '" + p.getAddress() + "', Gender = '"

+ p.getGender() + "' "

+ "WHERE Patient\_SSN = " + p.getPatient\_SSN() + ";";

getInsuranceInfoFromMRN: This method retrieves insurance plan information for a patient identified by their MRN (Medical Record Number). It constructs a query that first finds the patient's SSN (Social Security Number) using the MRN, then retrieves all fields of the insurance plan associated with that SSN from the INSURANCE\_PLAN table. The method then maps the retrieved data to an Insurance\_Plan object and returns it.

String q = "SELECT \* FROM INSURANCE\_PLAN WHERE Patient\_SSN = "

+ "(SELECT PATIENT\_SSN FROM PATIENTS WHERE PATIENT\_ID = " + mrn

+ ");";

UpdateInsurancePlan : This method updates an existing insurance plan in the INSURANCE\_PLAN table. The update is performed based on the Insurance\_Plan\_ID provided in the Insurance\_Plan object. It updates all relevant fields (insurance plan ID, company provider, class, issuing date, end date, and patient SSN) with the values from the given Insurance\_Plan object.

String q = "UPDATE INSURANCE\_PLAN SET Insurance\_Plan\_ID = " + ip.getInsurancePlanId()

+ ", Company\_Provider = '" + ip.getCompanyProvider()

+ "', Class = '" + ip.getInsuranceClass() + "', Issuing\_Date = '"

+ ip.getIssuingDate() + "', End\_Date = '" + ip.getEndDate() + "', Patient\_SSN = "

+ ip.getPatientSSN()

+ " WHERE Insurance\_Plan\_ID = " + ip.getInsurancePlanId() + ";";

getPerformFromSurgery: This method retrieves the performance details of a surgery. It uses the surgery ID from the provided Surgery object to find the corresponding performance record in the PERFORM\_SURGERY table. The method then maps the retrieved data to a Perform\_Surgery object and returns it.

String q = "SELECT \* FROM PERFORM\_SURGERY WHERE Surgery\_ID = "

+ "(SELECT SURGERY\_ID FROM SURGERY WHERE SURGERY\_ID = "

+ s.getSurgeryId();

retrieveSurgeriesbyMRN: This method retrieves all surgeries associated with a patient identified by their MRN. It performs a natural join between the SURGERY, PERFORM\_SURGERY, and PATIENTS tables to gather all relevant information about the surgeries. The method then maps the retrieved data to Surgeries objects and adds them to an ArrayList, which it returns.

String q = "SELECT \* FROM SURGERY NATURAL JOIN PERFORM\_SURGERY NATUAL JOIN PATIENTS WHERE "

+ "Patient\_ID = " + mrn + ";";

getMRNFromSSN: This method retrieves the MRN (Medical Record Number) of a patient based on their SSN (Social Security Number). It queries the PATIENTS table to find the Patient\_ID corresponding to the given SSN and returns it.

String q = "SELECT Patient\_ID FROM PATIENTS WHERE Patient\_SSN = " + p.getPatient\_SSN() + ";";

retrieveLabbyMRN: This method retrieves all lab tests associated with a patient identified by their MRN. It performs a natural join between the LAB\_TEST, MEDICAL\_FILE, and PATIENTS tables to gather all relevant information about the lab tests. The method then maps the retrieved data to Lab\_Test objects and adds them to an ArrayList, which it returns.

String q = "SELECT \* FROM LAB\_TEST NATURAL JOIN MEDICAL\_FILE NATURAL JOIN PATIENTS "

+ "WHERE Patient\_ID = " + mrnOfPatient;

retrieveRadiologybyMRN: This method retrieves all radiology reports associated with a patient identified by their MRN. It performs a natural join between the RADIOLOGY and MEDICAL\_FILE tables to gather all relevant information about the radiology reports. The method then maps the retrieved data to Radiology objects and adds them to an ArrayList, which it returns.

String q = "SELECT \* FROM RADIOLOGY NATURAL JOIN MEDICAL\_FILE WHERE MEDICAL\_FILE\_ID = " + mrnOfPatient + ";";

retrieveTreatmentsbyMRN: This method retrieves all treatments associated with a patient identified by their MRN. It performs a natural join between the TREATMENT, MEDICAL\_FILE, and PATIENTS tables to gather all relevant information about the treatments. The method then maps the retrieved data to Treatment objects and adds them to an ArrayList, which it returns.

String q = "SELECT \* FROM TREATMENT NATURAL JOIN MEDICAL\_FILE NATURAL JOIN PATIENTS WHERE Patient\_ID = " + mrnOfPatient + ";";

getMedicalFileIdFromSSN: This method retrieves the medical file ID associated with a patient's SSN. It queries the MEDICAL\_FILE table to find the Medical\_File\_ID corresponding to the given SSN and returns it.

String q = "SELECT Prescription FROM MEDICAL\_FILE NATURAL JOIN PATIENTS WHERE "

+ "Patient\_ID = " + mrnOfPatient + ";";

retrievePrescriptionFromMedicalFile: This method retrieves the prescription information associated with a patient's MRN (Medical Record Number). It performs a natural join between the MEDICAL\_FILE and PATIENTS tables to obtain the prescription details and returns it as a string.

String q = "SELECT Prescription FROM MEDICAL\_FILE NATURAL JOIN PATIENTS WHERE "

+ "Patient\_ID = " + mrnOfPatient + ";";

createTheirMedicalFile: This method creates a new medical file for a patient by inserting a new record into the MEDICAL\_FILE table. The record includes the patient's ID, the current date as the date of creation, and the patient's SSN.

String q = "INSERT INTO MEDICAL\_FILE (Medical\_File\_ID, Date\_Of\_Creation, Patient\_SSN) " +

"VALUES (" + p.getPatient\_ID() + ", '2024-07-22', '" + p.getPatient\_SSN() + "');";

createApptInAppointment: This method creates a new appointment by inserting a record into the APPOINTMENT table. The record includes the appointment day, start time, and end time.

String q = "INSERT INTO APPOINTMENT (Day, Start\_Time, End\_Time) VALUES ('"

+ a.getDay() + "', '" + a.getStartTime() + "', '" + a.getEndTime() + "')";

getDocIDFromApptID: This method retrieves the doctor ID associated with a specific appointment ID. It queries the IS\_AVAILABLE table to find the Doctor\_ID for the given Appointment\_ID and returns it.

String q = "SELECT Doctor\_ID FROM IS\_AVAILABLE WHERE Appointment\_ID = " + appointmentId;

addAppttoAvailable: This method adds an appointment to the IS\_AVAILABLE table, associating it with a specific doctor ID. It inserts the appointment ID and doctor ID into the table.

String q = "INSERT INTO IS\_AVAILABLE (Appointment\_ID, Doctor\_ID) VALUES ("

+ chosen.getAppointmentId() + ", " + doc\_ID + ")";

getDocIDFromDoc: This method retrieves the doctor ID based on the doctor's first and last name. It queries the DOCTOR table to find the Doctor\_ID for the given first and last name and returns it.

String q = "SELECT Doctor\_ID FROM DOCTOR WHERE First\_Name = '" + d.getFirstName()

+ "' AND Last\_Name = '" + d.getLastName() + "'";

getAllMedicalFacilities: This method retrieves all medical facilities from the MEDICAL\_FACILITY table and returns them as an ArrayList of Medical\_Facility objects. Each facility object is populated with the facility's ID, name, and location.

String q = "SELECT \* FROM MEDICAL\_FACILITY";

getSurgeriesFromMRN: This method retrieves all surgeries associated with a patient's MRN (Medical Record Number). It performs a natural join between the SURGERY and PERFORM\_SURGERY tables, and uses the Patient\_SSN to fetch relevant surgery details. The method returns an ArrayList of Surgeries objects.

String q = "SELECT \* FROM SURGERY NATURAL JOIN PERFORM\_SURGERY WHERE Patient\_SSN = (SELECT Patient\_SSN FROM PATIENTS WHERE Patient\_ID = " + mrnOfPatient + ");";

createNewRadiologyOnMedicalFile: This method creates a new radiology record for a patient by inserting a record into the RADIOLOGY table. The record includes the radiology name, date, report, reason, and the medical file ID associated with the patient.

String query = "INSERT INTO RADIOLOGY (Radiology\_Name, Date, Report, Reason, Medical\_File\_ID) VALUES (?, ?, ?, ?, ?)";

addNewSurgery: This method adds a new surgery record for a patient by inserting a record into the PERFORM\_SURGERY table. The record includes the doctor ID, medical facility ID, patient SSN, surgery ID, whether the surgery was successful, and the date of the surgery.

String query = "INSERT INTO PERFORM\_SURGERY (Doctor\_ID, Medical\_Facility\_ID, Patient\_SSN, Surgery\_ID, Successful, Date) VALUES (?, ?, ?, ?, ?, ?)";

addNewTreatment: This method adds a new treatment record for a patient by inserting a record into the TREATMENT table. The record includes the treatment name, reason, start date, end date, and the medical file ID associated with the patient.

String query = "INSERT INTO TREATMENT (Treatment\_Name, Reason, Start\_Date, End\_Date, Medical\_File\_ID) VALUES (?, ?, ?, ?, ?)";

addNewLabTest: This method adds a new lab test record for a patient by inserting a record into the LAB\_TEST table. The record includes the test name, date, report, reason, and the medical file ID associated with the patient.

String query = "INSERT INTO LAB\_TEST (Test\_Name, Date, Report, Reason, Medical\_File\_ID) VALUES (?, ?, ?, ?, ?)";

getLabTestsOfPatientsByMRN: This method retrieves all lab tests associated with a patient's MRN (Medical Record Number). It queries the LAB\_TEST table to find all records that match the given Medical\_File\_ID and returns them as an ArrayList of Lab\_Test objects.

String query = "SELECT \* FROM LAB\_TEST WHERE Medical\_File\_ID = ?";

updateMedicalFileWithNewPrescriptionUsingMRN: This method updates a patient's medical file with a new prescription based on the provided medical record number (MRN). The method constructs an SQL query to update the Prescription field in the MEDICAL\_FILE table for the specified Medical\_File\_ID. The connect method is called to establish a connection to the database, the update query is executed using the stmt object, and the connection is closed in the finally block to ensure that resources are released even if an exception occurs.

String q = "UPDATE MEDICAL\_FILE SET Prescription = '" + prescription + "' WHERE Medical\_File\_ID = " + patientMRN;

addEmergencyContactToMRN: This method adds a new emergency contact for a patient identified by their medical record number (MRN). The method constructs an SQL query to insert the details of the emergency contact (first name, last name, phone number, email, relationship) into the EMERGENCY\_CONTACT table, associating it with the patient's Medical\_File\_ID. The connect method is called to establish a connection to the database, the insert query is executed using the stmt object, and the connection is closed in the finally block.

String q = "INSERT INTO EMERGENCY\_CONTACT VALUES ('"

+ ec.getPhoneNumber() + "', '"

+ ec.getName() + "', '"

+ ec.getRelationship() + ", "

+ ec.getPatientSSN() + ");";

getApptsBySpecialty: This method retrieves all available appointments for a specific medical specialty. It constructs an SQL query to select all appointments from the APPOINTMENT, IS\_AVAILABLE, and DOCTOR tables where the doctor's specialty matches the provided specialty and the appointment is not yet booked (i.e., Patient\_SSN is null). The method establishes a connection to the database, executes the query, iterates over the result set to create Appointment objects, adds them to an ArrayList, and closes the connection.

String q = "SELECT \* FROM APPOINTMENT a NATURAL JOIN IS\_AVAILABLE NATURAL JOIN DOCTOR"

+ "WHERE Specialty = '" + chosenSpecialty + "' AND a.Patient\_SSN IS NULL;";

getApptFromSSN: This method retrieves all appointments for a patient identified by their Social Security Number (SSN). It constructs an SQL query to select all appointments from the APPOINTMENT table where the Patient\_SSN matches the provided SSN. The method establishes a connection to the database, executes the query, iterates over the result set to create Appointment objects, adds them to an ArrayList, and closes the connection.

String q = "SELECT \* FROM APPOINTMENT WHERE Patient\_SSN = " + SSN + ";";

updateAppointment: This method updates an appointment to set the Patient\_SSN field to NULL, effectively marking the appointment as unbooked. It constructs an SQL query to update the APPOINTMENT table where the Appointment\_ID matches the provided appointment's ID. The method establishes a connection to the database, executes the update query, and closes the connection in the finally block.

String q = "UPDATE APPOINTMENT SET Patient\_SSN = NULL WHERE Appointment\_ID = " + a.getAppointmentId() + ";";

getBookedAppointments: This method retrieves all booked appointments for a specific doctor identified by their doctor ID. It constructs an SQL query to select all appointments from the APPOINTMENT and IS\_AVAILABLE tables where the Doctor\_ID matches the provided ID and the appointment is booked (i.e., Patient\_SSN is not null). The method establishes a connection to the database, executes the query, iterates over the result set to create Appointment objects, adds them to an ArrayList, and closes the connection.

String q = "SELECT \* FROM APPOINTMENT NATURAL JOIN IS\_AVAILABLE WHERE"

+ "Doctor\_ID = " + DocID + " AND Patient\_SSN IS NOT NULL;";

getPatientFromAppt: This method retrieves the patient information for a given appointment. It constructs an SQL query to select all patient details from the PATIENT table where the Patient\_SSN matches the SSN associated with the given appointment's ID. The method establishes a connection to the database, executes the query, checks if a result is found, creates a Patient object with the retrieved information, and closes the connection.

String q = "SELECT \* FROM PATIENT WHERE Patient\_SSN = (SELECT"

+ "Patient\_SSN FROM APPOINTMENT WHERE Appointment\_ID = " + ap.getAppointmentId() + ";";

Patient p = null;

NewApptID: This method retrieves the maximum appointment ID from the APPOINTMENT table to determine the most recently created appointment's ID. It constructs an SQL query to select the highest value of Appointment\_ID

String q = "SELECT MAX(Appointment\_ID) AS Appointment\_ID FROM APPOINTMENT";

getSurgeriesFromSSN: This method retrieves all surgeries performed on a patient based on their Social Security Number (SSN). It constructs and executes an SQL query to select all relevant surgeries from the SURGERY and PERFORM\_SURGERY tables.

String q = "SELECT \* FROM SURGERY NATURAL JOIN PERFORM\_SURGERY WHERE Patient\_SSN = " + SSnOfPatient + ";";

EnsureEmail: This method checks if the provided email address matches the email address associated with a specific doctor in the database.

String q = "SELECT email FROM DOCTOR WHERE Doctor\_ID = " + iDofDoctor + ";";

EnsureEmailofPatient:

This method checks if the provided email address matches the email address associated with a specific patient in the database.

String q = "SELECT email FROM PATIENTS WHERE Patient\_ID = " + parseInt + ";";

bookAppt: This method updates the details of an appointment in the database, setting the patient assigned to the appointment and the reason for the appointment.

String q = "UPDATE APPOINTMENT SET Patient\_SSN = " + a.getPatient\_SSN() + ", REASON = '" + a.getReason() + "' WHERE Appointment\_ID = " + a.getAppointmentId();

getDoctors: This method retrieves all doctors from the database and returns them as an ArrayList of Doctor objects.

String query = "SELECT \* FROM DOCTOR;";

addDRtoMedicalFile: This method assigns a specified doctor to a patient's medical file by updating the Doctor\_ID field in the MEDICAL\_FILE table.

String q = "UPDATE MEDICAL\_FILE SET Doctor\_ID = " + chosenDR

+ " WHERE Medical\_File\_ID = " + mrnOfPatient + ";";