Names: Mahima Roy Karmakar and Manal Jkini

Maven Programming Pattern Project

Scenario:

We have chosen to do a project about clinic management system. The clinic management system is used in the clinic to allow the management of the database of a client which is the patient. This management system is used by the doctors to perform tasks related to the user's and client's needs. The application operates in hospitals where doctors need quick access to patient histories, appointment, and treatment plans. The doctors can schedule an appointment by contacting the patient. They can also cancel an appointment if the patient cannot come to the appointment.

Doctors can give prescriptions, diagnoses or treatments for various medical conditions of patients. Each treatment has a price for which the clients have to do the payment. But the payment is not taken by the system. The system will only inform the user about the total cost of their treatments.

Doctors can view and update patient's medical records which contain the name, address, phone number, doctor name, appointments, treatments, diagnoses, prescriptions and payment information. This keeps track of the medical history of each patient to provide continuous care. They can make necessary changes in the list of data like adding a new patient or removing a patient that will no longer be registered in that clinic.

Design Paradigms

Here are the functionalities that we plan to demonstrate depending on the user.

Doctors:

1) Create appointment: Doctors can create new appointments for patients, specifying the date, time, and reason for the visit.

2) View appointment: Doctors can view the appointments of patients.

3) Cancel or reschedule appointments: In case a patient or doctor needs to cancel an appointment, the system provides functionality to remove an appointment from the system and update it.

4) Add patient: This functionality to register a new patient with information: name, address, phone number, assigned doctor, scheduled appointments, medical history, prescribed treatments and payment details.

5) View who are their patients and their information.

6) Remove patient: This functionality will remove patients who are no longer receiving services at the clinic.

7) Access medical reports.

8) Create new medical reports.

9) Create treatment plans

10) View treatment plans (diagnosis, treatment and price)

11) Cost: This will calculate the cost of the patient's treatment at the clinic.

12) Login System: This functionality will make secure login system where users must authenticate themselves before accessing the system with password and name. If the user is not found by the system, it will reject access.

13) Appointment availability check: This will show available time for each doctor, which will help to schedule appointments more easily by avoiding conflicts.

14) Doctors can switch between French and English.

Patients:

1) Create appointment.

2) View appointment.

3) Reschedule appointment.

4) Cancel appointment.

5) Access medical reports

6) View treatment plans

7) The application can be switched between French and English.

8) Can view their doctors

Expected Output:

We expect that our output will be an application created with graphical user interface that the doctor can use. First, the application will ask the user to choose a language like English or French that will translate the system to the language chosen. We can make this happen using I18N Java classes. Then, the application will ask the user to enter a password and the user's name. If the login is not validated, it will reject the access to the medical records. If it is validated, it will ask the user to choose between schedule appointments, cancel appointments, view available appointment, add patient, remove patient or view medical record. These options are represented with buttons If a wrong option is chosen, it will show an error message and ask the user to choose again.

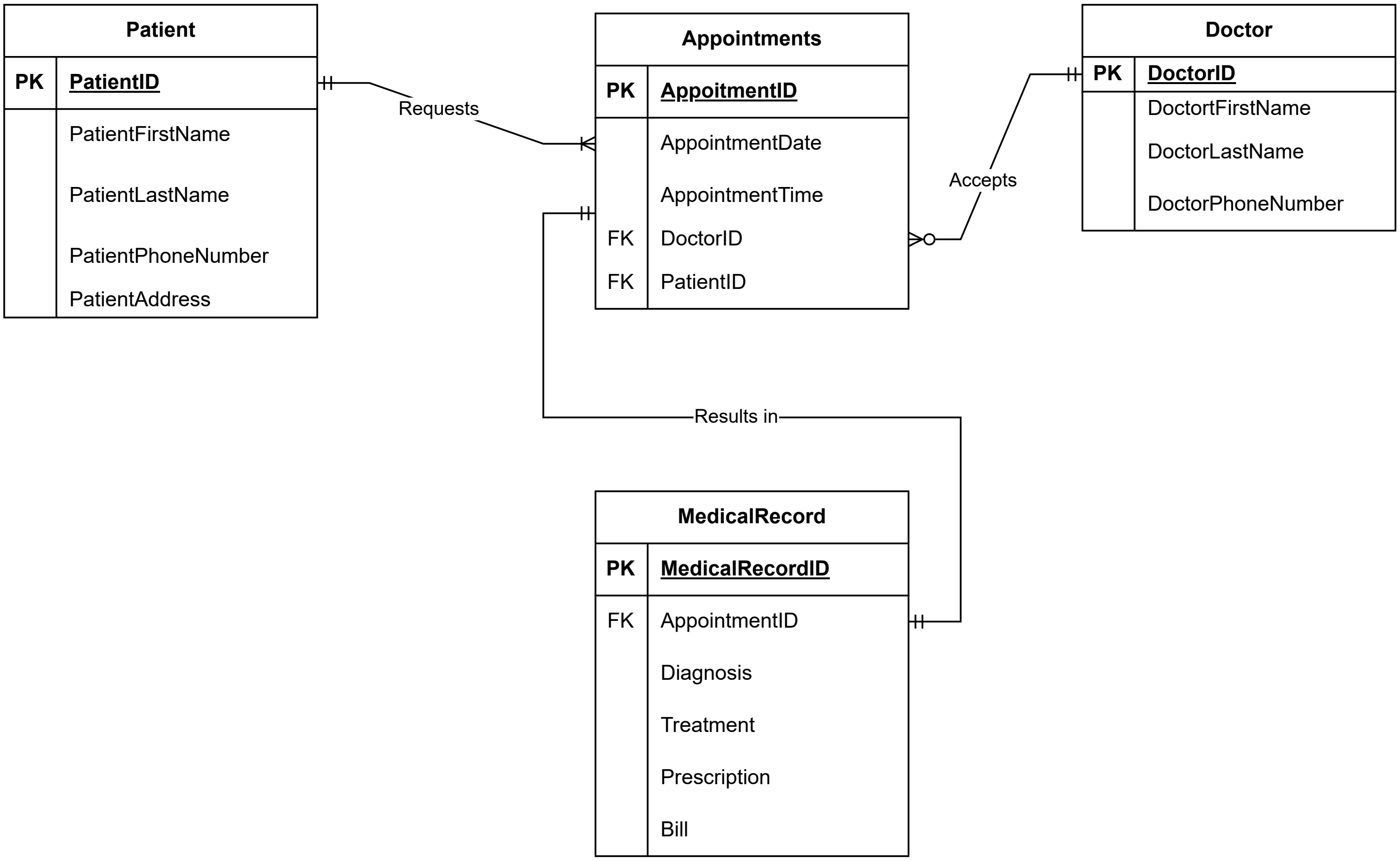
When the schedule appointment is chosen, the application will ask the user to enter the date of appointment for specific client by entering their name. If the user wants to remove an appointment, it will also ask the date of the appointment to remove. If the users do not know which the days available to schedule an appointment are, the user can go back and view available appointment. After each task, there will be confirmation message to ensure that the task is done successfully. You also have a button for patients records where you can view either all of them at once or filter them by checking the options that you want e.g.: only the patients that have a specific diagnosis that you specify can be selected as well as their information or if the patients have more than three appointments they will be selected. There is also a button where you can create new medical record for a patient.

When the user chooses to add a patient, the application will ask the user to enter important confidential information of the client like name, address, phone number, assigned doctor, scheduled appointments, medical history, prescribed treatments and payment details. If the user wants to remove a client, the application will ask to enter the patient's name, and the system will remove all the information history of that client. After this task, it will ask the user to confirm the change to update the system.

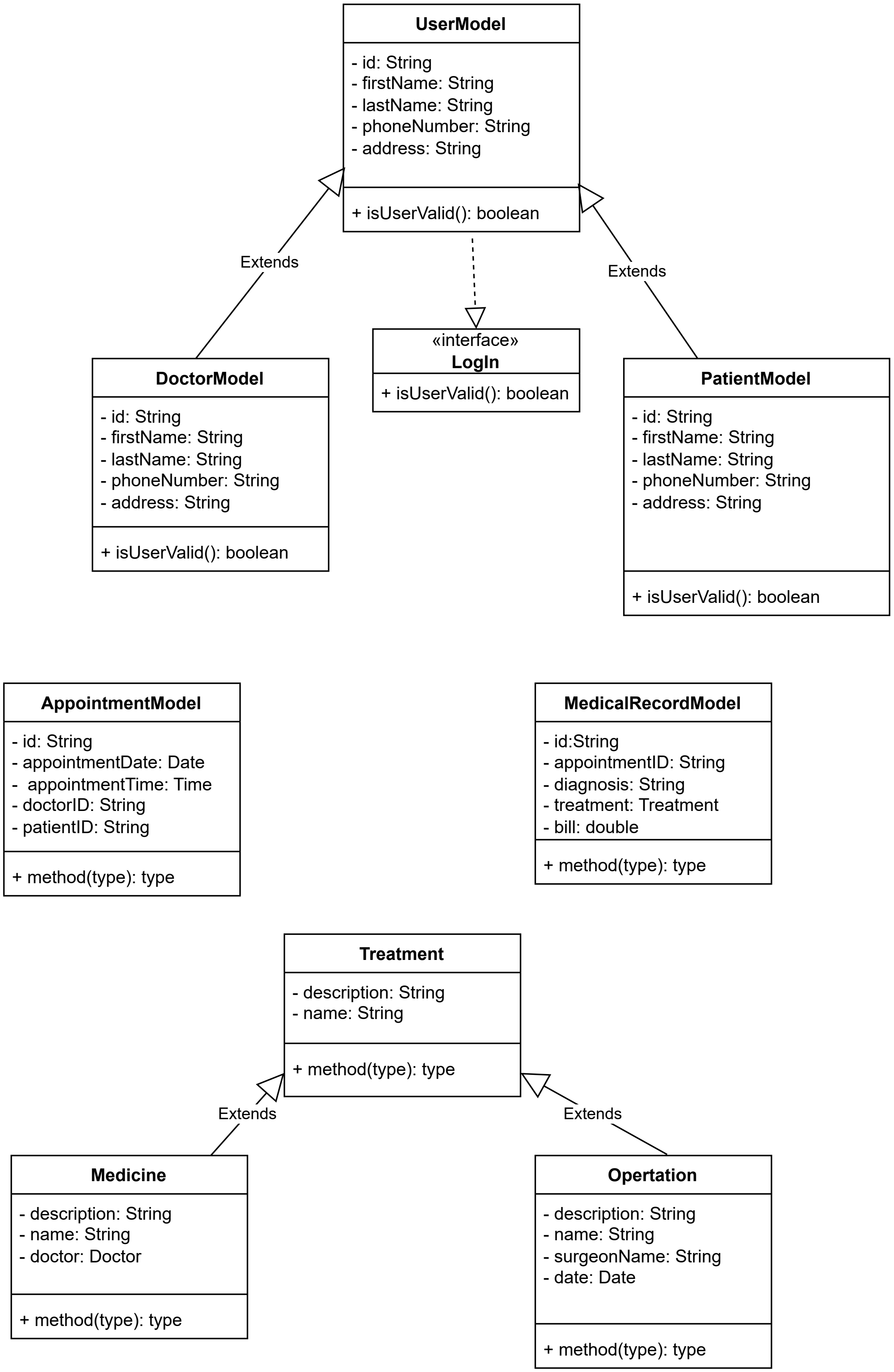
When the user chooses to view a client's medical record, the application will ask the user to enter the name of the client of which the user wants to view records. Then, it will show all the information of the client like the confidential information, assigned doctor, scheduled appointments, medical history, prescribed treatments and payment details. The user is also able to change those information's and save it. For example, the changes could be adding or removing treatments, diagnoses or prescribing new medications. The changes made will be updated by the system.

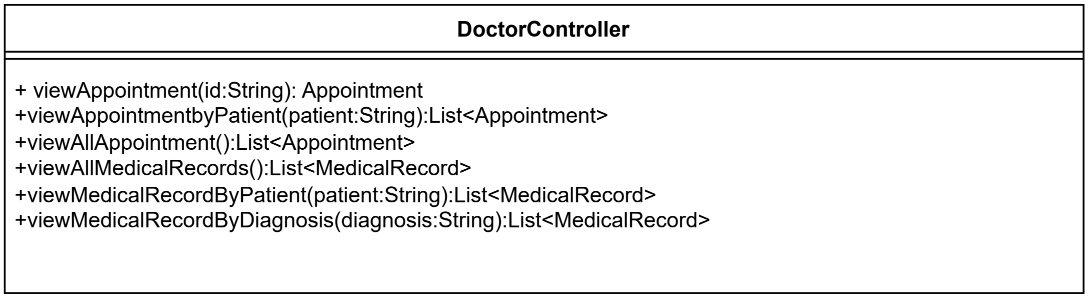
If you sign in as a patient, you click on the appointment button that gives you option to view all your appointments, cancel or reschedule them. You also have a button to view your medical records where you can view either all of them at once or filter them by checking the options that you want e.g.: view only the medical records that was done by a specified doctor. There is also a button to check your bills.

Clinic Management ER Diagram



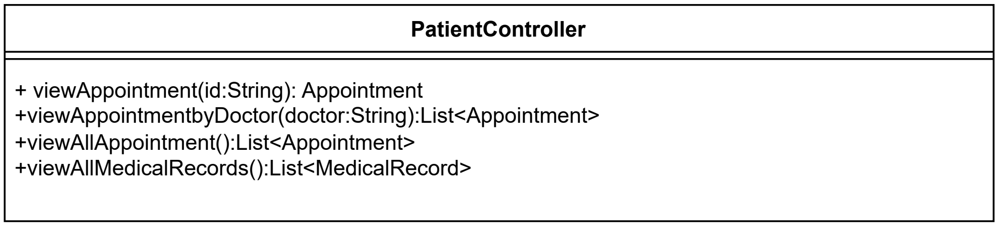
Clinic Management Class Diagram





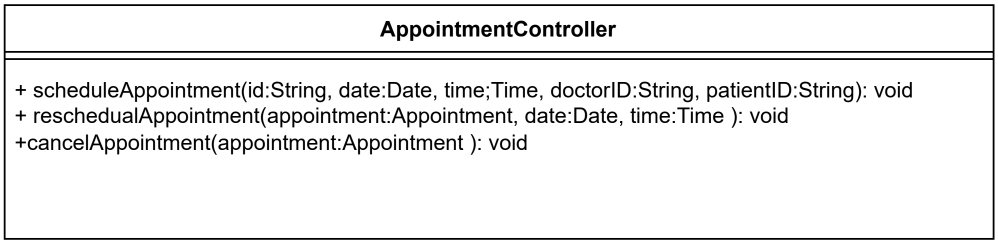
DoctorController

```
+ viewAppointment(id:String): Appointment
+viewAppointmentbyPatient(patient:String):List<Appointment>
+viewAllAppointment():List<Appointment>
+viewAllMedicalRecords():List<MedicalRecord>
+viewMedicalRecordByPatient(patient:String):List<MedicalRecord>
+viewMedicalRecordByDiagnosis(diagnosis:String):List<MedicalRecord>
```



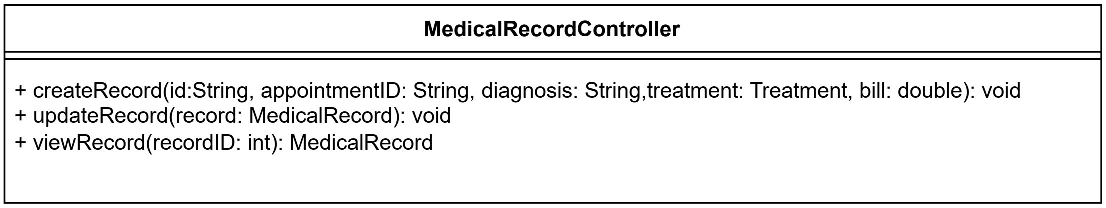
PatientController

```
+ viewAppointment(id:String): Appointment
+viewAppointmentbyDoctor(doctor:String):List<Appointment>
+viewAllAppointment():List<Appointment>
+viewAllMedicalRecords():List<MedicalRecord>
```



AppointmentController

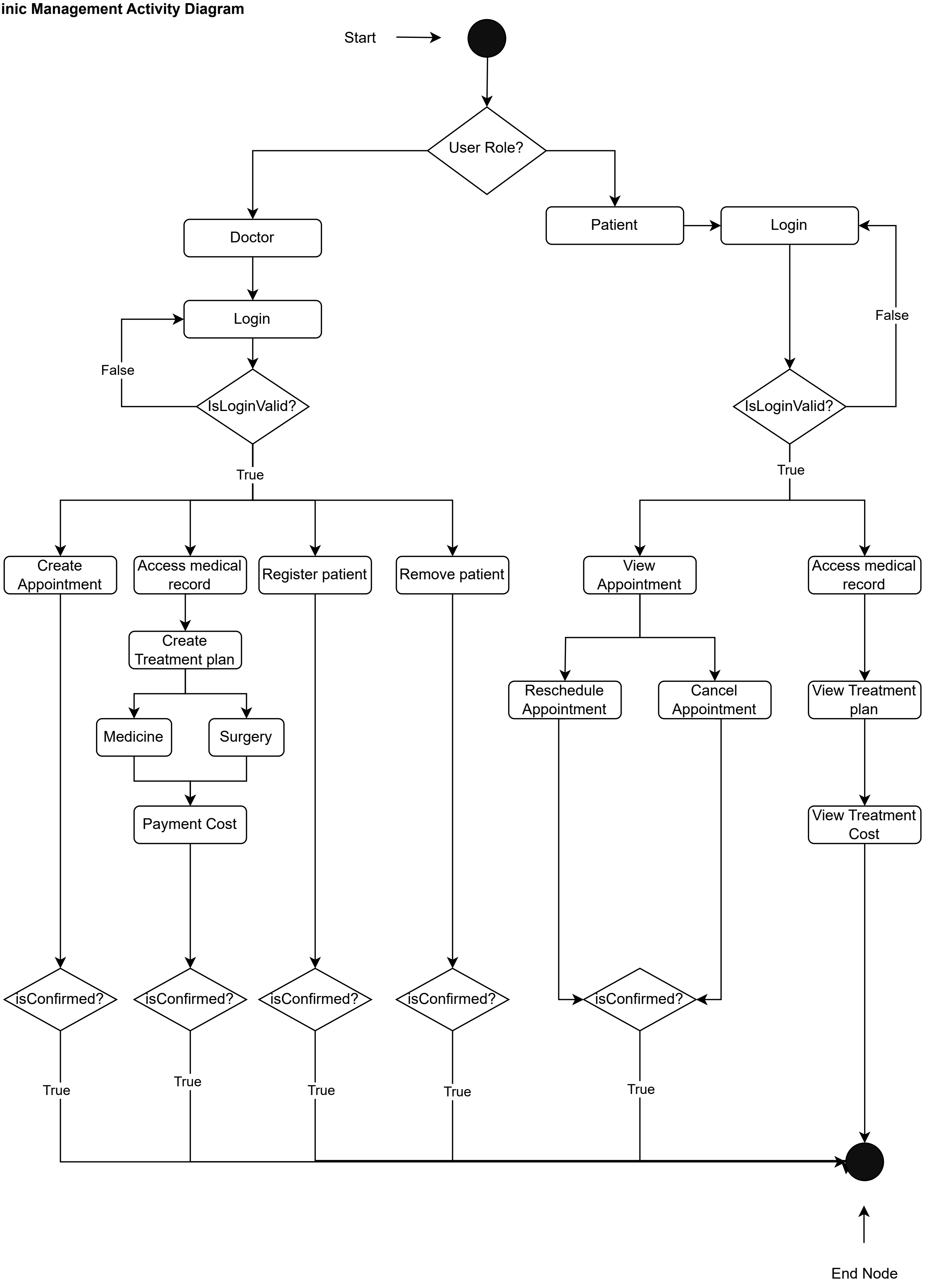
```
+ scheduleAppointment(id:String, date:Date, time;Time, doctorID:String, patientID:String): void
+ reschedualAppointment(appointment:Appointment, date:Date, time:Time ): void
+cancelAppointment(appointment:Appointment ): void
```



MedicalRecordController

```
+ createRecord(id:String, appointmentID: String, diagnosis: String,treatment: Treatment, bill: double): void
+ updateRecord(record: MedicalRecord): void
+ viewRecord(recordID: int): MedicalRecord
```

Clinic Management Activity Diagram


Start
User Role?
Doctor
Login
False
IsLoginValid?
True
Create Appointment
Access medical record
Register patient
Remove patient
Create Treatment plan
Medicine
Surgery
Payment Cost
isConfirmed?
isConfirmed?
isConfirmed?
isConfirmed?
True
True
True
True
Patient
Login
False
IsLoginValid?
True
View Appointment
Access medical record
Reschedule Appointment
Cancel Appointment
View Treatment plan
View Treatment Cost
isConfirmed?
True
End Node