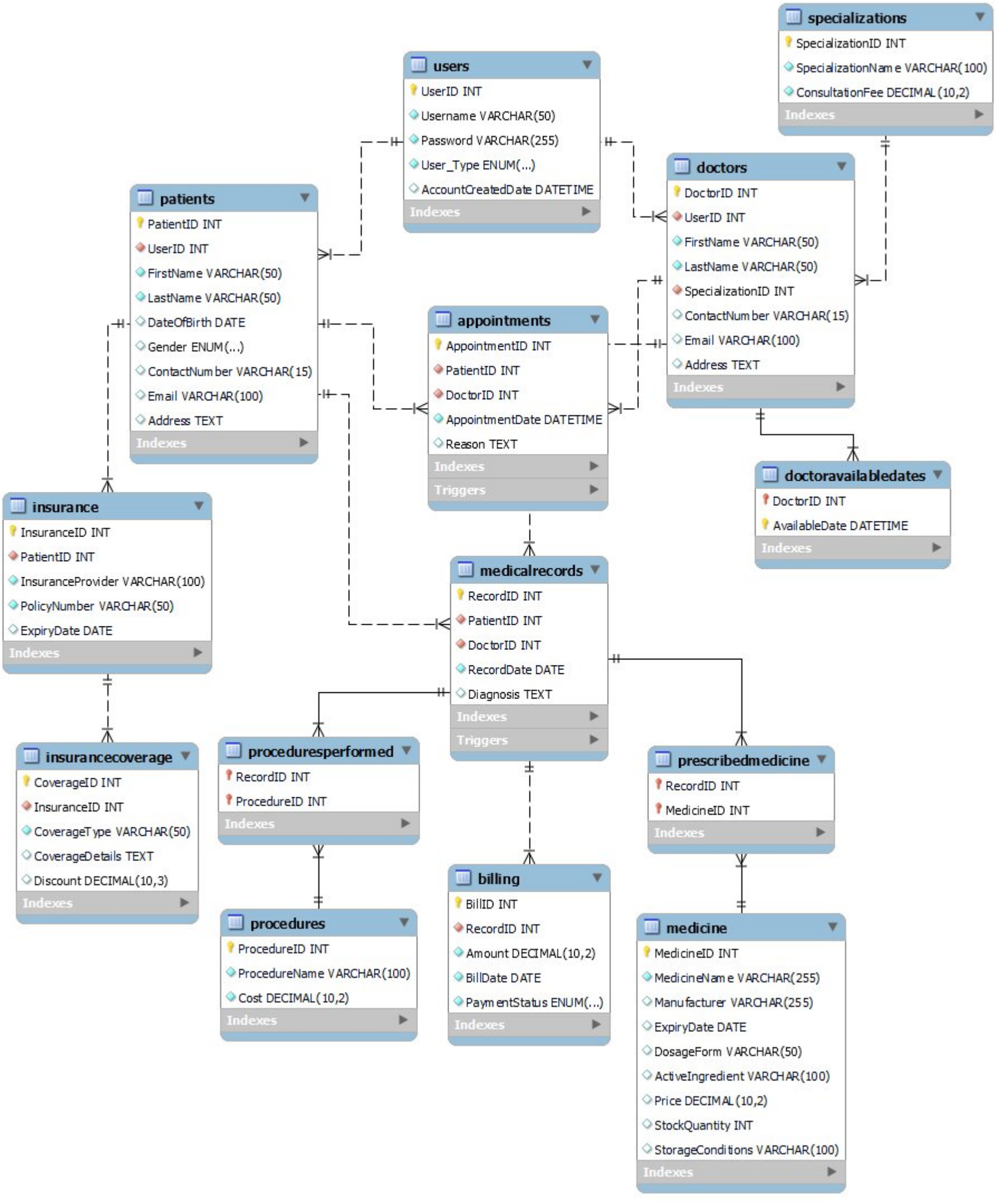

specializations
SpecializationID INT
SpecializationName VARCHAR(100)
ConsultationFee DECIMAL(10,2)
Indexes
users
UserID INT
Username VARCHAR(50)
Password VARCHAR(255)
User_Type ENUM(...)
AccountCreatedDate DATETIME
Indexes
doctors
DoctorID INT
UserID INT
FirstName VARCHAR(50)
LastName VARCHAR(50)
SpecializationID INT
ContactNumber VARCHAR(15)
Email VARCHAR(100)
Address TEXT
Indexes
patients
PatientID INT
UserID INT
FirstName VARCHAR(50)
LastName VARCHAR(50)
DateOfBirth DATE
Gender ENUM(...)
ContactNumber VARCHAR(15)
Email VARCHAR(100)
Address TEXT
Indexes
appointments
AppointmentID INT
PatientID INT
DoctorID INT
AppointmentDate DATETIME
Reason TEXT
Indexes
Triggers
doctoravailabledates
DoctorID INT
AvailableDate DATETIME
Indexes
insurance
InsuranceID INT
PatientID INT
InsuranceProvider VARCHAR(100)
PolicyNumber VARCHAR(50)
ExpiryDate DATE
Indexes
medicalrecords
RecordID INT
PatientID INT
DoctorID INT
RecordDate DATE
Diagnosis TEXT
Indexes
Triggers
insurancecoverage
CoverageID INT
InsuranceID INT
CoverageType VARCHAR(50)
CoverageDetails TEXT
Discount DECIMAL(10,3)
Indexes
proceduresperformed
RecordID INT
ProcedureID INT
Indexes
prescribedmedicine
RecordID INT
MedicineID INT
Indexes
billing
BillID INT
RecordID INT
Amount DECIMAL(10,2)
BillDate DATE
PaymentStatus ENUM(...)
Indexes
procedures
ProcedureID INT
ProcedureName VARCHAR(100)
Cost DECIMAL(10,2)
Indexes
medicine
MedicineID INT
MedicineName VARCHAR(255)
Manufacturer VARCHAR(255)
ExpiryDate DATE
DosageForm VARCHAR(50)
ActiveIngredient VARCHAR(100)
Price DECIMAL(10,2)
StockQuantity INT
StorageConditions VARCHAR(100)
Indexes

CS5200 - Database Management Systems - Fall 2023

Submitted by: Aayush Jaiswal, Kabi Shrestha
Team Name: JaiswalAShresthaK

Software used for this project: Python, MySQL Workbench
Libraries/Modules used in Python: pymysql, tkinter, datetime

DESCRIPTION:

We created a database application for a hospital management system. The application allows patients to schedule appointments with doctors, view their medical records, view their bills, and view their prescribed medicine. The application also allows doctors to view their upcoming appointments, cancel appointments, view their patient's medical records, and create new medical records for their patients.

PREREQUISITE:
- Open the file connection.py in your IDE and change the host, username, password, database name to the file with your MySQL Workbench details.
- In your terminal, run the command 'pip install pymysql' to install the pymysql module.
- Then also run the command 'pip install tkinter' to install the tkinter module.

FOR RUNNING THE APPLICATION:

1) Open the file numdDashboard.py in your IDE
2) Open dedicated terminal in your IDE and run the command 'python numdDashboard.py'
4) Now you have entered the application.
5) You'll notice a window pop up asking for the user to sign up or sign in. Click on sign up.
6) You'll be asked for a new username and password and the user type - Patient or Doctor
7) Fill out details and click submit
8) If user was created successfuly, you'll be asked to log in again with your new credentials.
9) After log in is successful, a new window will ask you to finish registering as the user type you chose.
10) If you selected PATIENT:
    a. You'll be asked to fill out first name, last name, date of birth, phone number, email, address, and gender.
    b. Enter details and click submit
    c. If successful you can see your patient dashboard wth options to click on button: View Appointments, View Medical Report, Prescribed Medicine, and Sign Out.
    d. Go to step 12 for further information on each button.

11) If you selected DOCTOR:
a. You'll be asked to fill out first name, last name, phone number, email, address, and specialization.
b. Enter details and click submit
c. If successful you can see your doctor dashboard
d. From here you can click on button: View Appointments, View Patient Records, and Sign Out.
e. Go to step 16 for further information on each button.

PATIENT BUTTONS:

12) View Appointments:
a. It displays a table of all upcoming appointments for the patient. Along with the patient's information,
it also displays the doctor's name, the date and time of the appointment, and the reason for the appointment.
b. Each row of the table is an appointment and sorted by earliest upcoming appointment to latest upcoming appointment.
To select an appointment, simply click on the row.
c. 'Schedule New Appointment' will open a new winow which will ask the user to fill out a form with Specialization of
the doctor, the requested date, and reason of visit. Submitting this form will pass these values to the procedure,
CreateAppointment which will query for a doctor that has the requested specialization and is available on that day of the
week as requested appointment date. For instance, if the user requests an appointment for Monday, December 12, 2023, then the
doctor must be available on Modays. If the doctor is available, then the appointment will be created on the doctor's available
time on Monday.
d. 'Cancel Appointment' to cancel a seleted upcoming appointment. If no appointment is selected,
it will display an error message. Otherwise, it will delete the appointment from the view (and the database).
e. 'Reschdeule Appointment' to reschdeule a selected upcoming appointment. If no appointment is selected, it will display
an error message. If an appointment is selected, it will open a new window and have the same behavior as
'Schedule New Appointment' button. While rescheduling, the procedure will exclude the current appointment date and time
to avoid conflict with the new appointment date and time.

13) View Medical Report:
a. Clicking on this button will open a new window with a table of all created medical records for the patient. Along with patient's
details and doctor details, it will display the date of the record date, and the diagnosis by the doctor.
b. Each row of the table is a medical record and to select a medical record, simply click on the row.
c. 'View Billing' will open a new window showing the total amount and the itemized bill at the bottom for the selected medical record.
d. 'Make Payment' allows the selected bill to be paid. Status of the bill will be changed to 'Paid'.

14) Prescribed Medicine:

a. Allows patient to view all their prescribed medicine. It will display the name of the medicine, the manufacturer, expiration date, sotrage conditions on each row.

15) Sign Out: will sign out the user and go back to the sign in page.

DOCTOR BUTTONS:

16) View Appointments:
a. It displays all upcoming appointments for the patient. Along with the patient's information, it also displays the doctor's name, the date and time of the appointment, and the reason for the appointment.
b. Each row of the table is an appointment and sorted by earliest upcoming appointment to latest upcoming appointment. To select an appointment, simply click on the row.
c. 'Cancel Appointment' to cancel a seleted upcoming appointment. If no appointment is selected, it will display an error message. Otherwise, it will delete the appointment from the view (and the database).

17) View Patient Records:
a. Shows a table of the patients the doctor has seen. Displays the patientID, patient's name, and their phone number. To select a patient, simply click on the row.
b.'Create Medical Record' will open a new window which will ask the doctor to fill out a form which contaiins the diagnois, list of procedures, and list of medicines prescribed. Submitting this form will pass these values to the procedure, CreateMedicalRecord which will first insert a new medical record into teh database and fetch the new recordID which will be used to insert the list of procedures and list of medicines prescribed into the database. Also look at FORMAT FOR CREATING MEDICAL RECORDS at step 19 for more information.
c. 'View Patient History' allows the doctor to access the selected patient's medical records. This allows the doctor to view any previous medical record that was created for that patient in our database for a comprehensive diagnosis.

18) Sign Out: will sign out the user and go back to the sign in page.

FORMAT FOR CREATING MEDICAL RECORDS:

19) Has 3 fields: Diagnosis, Procedures, and Medicines Prescribed
a. Diagnosis: free text field. Doctor can enter any diagnosis they want. For example: "Patient has a cold, runny nose, and fever of 102 degrees. Blood Test indicates that patient has a bacterial infection"
b. Procedures: Enter a comma separated list of medical procedures/tests performed on the patient. For example: "X-Ray, Blood Test, Physical Examination"
c. Medicines Prescribed: Enter a comma separated list of medicines prescribed to the patient. For example: "Tylenol, Advil".

Note: the medicine must already exist in the database. If the medicine does not exist, the procedure will fail and the medical record will not be created.

LESSONS LEARNED:

a. We learned how to use Python to connect to a MySQL database and perform queries on the database, along with creating procedures, functions, and triggers in the database.
b. We gained experience in using the tkinter module to create a GUI for our application and how to use the grid layout manager to organize the widgets in our frontend.
c. We realized that it was more useful to create database objects like procedures, functions, and triggers after having a basic idea of how our GUI was supposed to work.
d. We learned about the conceptual design and logical design of a database management system and how to implement it in MySQL Workbench.
e. We learned to desing and implement a frontend for our dataabse application.

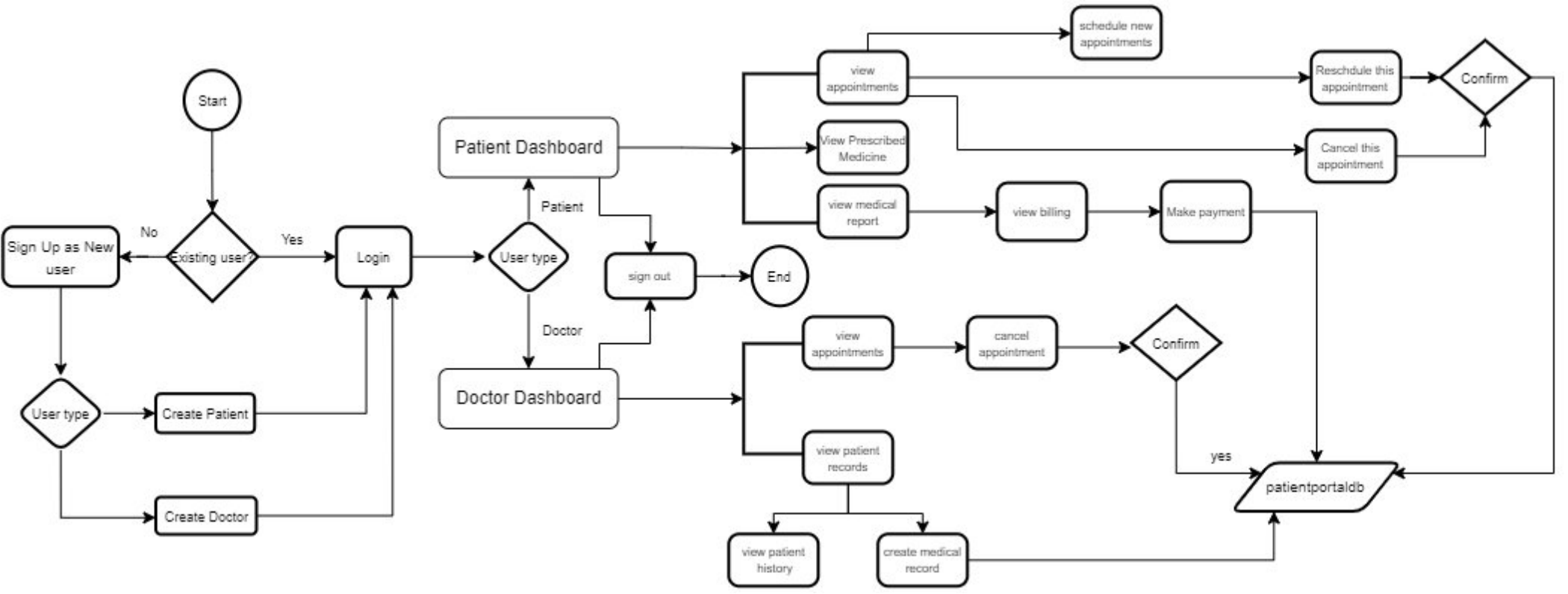
ISSUES:

a. Multiple instances of the application can be run at the same time which can result in other users being able to access another user's ID and type. This can cause UI elements to be displayed that would otherwise not be displayed to the user.
b. Display format issue when viewing the table of results where information is not displayed in a uniform manner.
c. When patient schedules or reschdeules an appointment they have to

---

FUTURE WORK:

a. We plan on adding the payment process for the patient to pay their bill.
b. We plan on making the GUI more user friendly and adding more features to the GUI like being able to message the doctor, being able to view the doctor's schedule, etc.
c. We plan on adding functionality for other hospital staff like nurses, technicians, etc.
d. To improve our appliaction, we intend to restrict it to only one instance of the application running at a time. We also intend to restrict the application to only one instance of the user logged in at a time.


Start
Existing user?
No
Yes
Sign Up as New user
Login
User type
Create Patient
Create Doctor
User type
Patient
Doctor
Patient Dashboard
Doctor Dashboard
sign out
End
view appointments
schedule new appointments
Reschdule this appointment
Confirm
Cancel this appointment
View Prescribed Medicine
view medical report
view billing
Make payment
view appointments
cancel appointment
Confirm
yes
view patient records
view patient history
create medical record
patientportaldb

Users

PK	UserID
AK	Username
	SHA2(Password)
	User_Type
	AccountCreatedDate

Specializations

PK	SpecializationID
	SpecializationName
	ConsultationFee

Patients

PK	PatientID
	FirstName
	LastName
	DateOfBirth
	Gender
	ContactNumber
	Email
	Address

Doctors

PK	DoctorID
	FirstName
	LastName
	AvailableDates[...]
	ContactNumber
	Email
	Address

Insurance

PK	InsuranceID
	InsuranceProvider
AK	PolicyNumber
	InsusranceCoverage[1..*]
	ExpiryDate

Appointments

PK	AppointmentID
	AppointmentDate
	Reason

MedicalRecords

PK	RecordID
	RecordDate
	Diagnosis

Procedures

PK	ProcedureID
	ProcedureName
	Cost

Billing

PK	BillID
	Amount
	BillDate
	PaymentStatus

Medicine

PK	MedicineID
	MedicineName
	Manufacturer
	ExpiryDate
	DosageForm
	ActiveIngredient
	Price
	StockQuantity
	StorageConditions

HasRole 1 — 0..1

1 HasRole 0..1

1 — 0..1 Specializes

1 — 0..* HealthInsurance

1 — 0..* Schedules

1 — 0..* Requests

1 — 1 Writes

1 — 0..* IsRecorded

1..* — 1..*

ProceduresProformed 0..* — 0..*

1 — 1 GetsBilled

0..* — 0..* PrescribedMedicine