What Platform Will You Use?

We will be using Python and Postgresql to build the SQL database. The platform is going to be built using the Django Framework.

What Functionality Will the Final Application Have?

- **Deliverable 1: Identify the different kinds of users that your application will service.**

  The users are (Staff - doctors, nurses, lab technicians) and patients

- **Deliverable 2: You need to have at least one INSERT statement.**

Staff

- Make query to Insert scheduled time for themselves

Patient:

- Make query to Insert Self to Patient

- **Deliverable 3: You need to have at least one DELETE statement.**

Staff

- Make query to Delete Schedule

- **Deliverable 4: You need to have at least one UPDATE statement.**

Patient:

- Make query to Update Info of patient

- **Deliverable 5: At least one meaningful query must join 3 or more tables.**

Nurse: Make query to get appointment information of their doctor. Need to join Nurse, Doctor, Appointment

Staff: Make query to select schedules according to given staff id.
Then we need to join Weeklyschedule, Staff, Scheduled_time.

Doctor,Patient: Want view their prescription, medicine info
Join Treats,Prescription,Contains,Medicine

- **Deliverable 6: At least one other meaningful query needs to join 2 or more tables.**

Patient:

  - Make query to get doctor's name, appointment details for patient
    Then we need to join Appointment, Doctor, Staff(to get Doctor name), Patient(for patient info)

- **Deliverable 7: At least one query must be an interesting GROUP BY query (aggregation). Describe it.**

Doctor:

  - Make query to GroupBy Medicine by Brand, then show decreasing order of count
  - Make query to GroupBy Medicine by Ingredient, then show decreasing order of count

  Description:
  Doctors want to analyze the statistical fact of the medicine they prescribe to patient. Then they will use this function.

- **Deliverables 8-10: Describe the other queries you plan to have (these can be simpler queries), so that you have at least 10 SQL statements overall.**

Staff

  - Make query to View Schedule, view info, view prescription, view appointment

Doctor:

  - Make query to View Nurse's schedule, info

Lab Technician:

  - Make query to View facility booking they work in.

Patient:

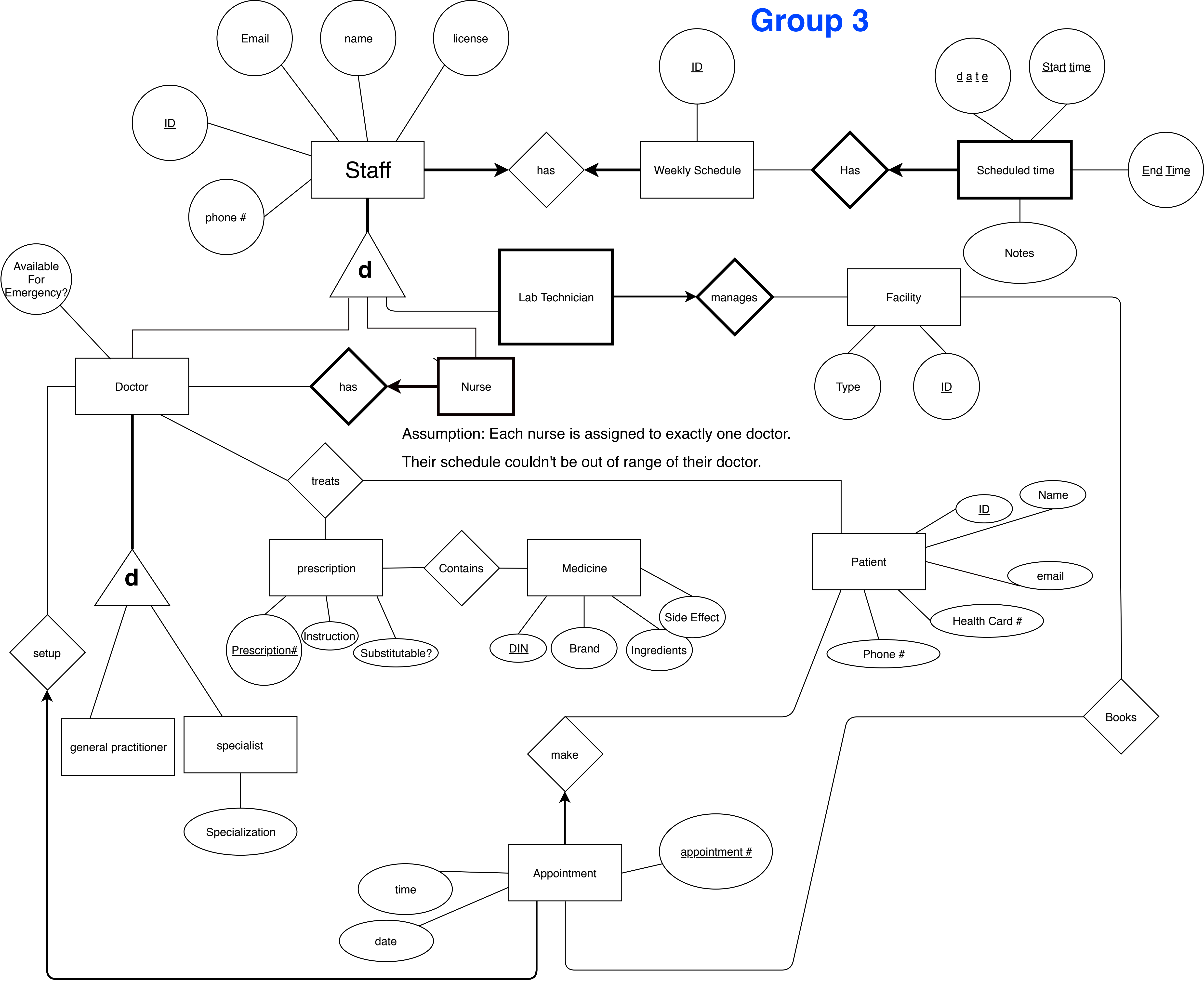
  - Make query to View info. View emergency doctor info (using select statement to select in view that we made by create view)

- **<u>Deliverable 11</u>: You need to have at least one view for your database, created using the CREATE VIEW statement in SQL. It should be a proper subset of a table. When the user wants to display all the data from this view, they will get all the columns specified by the view, but not all of the columns in the underlying table.**

Patient:

- Make query to create view Available from Doctors Where Available For Emergency =True
- We separate two types of viewing, when patient first click the link it just show doctor's name and ID but if patient agree to privacy statement, then it shows phone number too.

Group 3



Assumption: Each nurse is assigned to exactly one doctor.

Their schedule couldn't be out of range of their doctor.

Staff(ID : int, Email : char(20), name : char(20), license : char(8), phone # : char(10))

Primary Key: ID

Alternate Key: phone #, Email, license

Doctor(**ID** : int, AvailableForEmergency : bool)

Primary Key: ID

Foriegn Key: ID

ID references Staff

GeneralPracticioner(**ID** : int)

Primary Key: ID

Foriegn Key: ID

ID references Doctor

Specialist(**ID** : int, specialization : char(20))

Primary Key: ID

Foriegn Key: ID

ID references Doctor

Nurse(**ID** : int, **DID** : int)

Primary Key: ID, DiD

Foreign Key: ID, DID

ID references Staff

DID references Doctor

Lab_Technician(**ID** : int, **FID**: int)

Primary Key: ID, FID

Foreign Key: ID, FID

FID references Facility

Facility(ID: int, Type: char(20))

Primary Key: ID

WeeklySchedule(ID:int, **SID:** int)

Primary key: ID

Foreign Key: SID

SID references Staff

SID needs to be unique and not a NULL.

Scheduled_Time(Date: char(8), Start Time: char(8), End Time: char(8), **WID**: int, Notes: char(60))

Primary Key: Date, Start Time, End Time, WID

Foreign Key: WID

WID references WeeklySchedule

Books(**FID:** int, **appointment #**: int)

Primary Key: FID, appointment #

Foreign Key: FID, appointment #

FID references Facility

Appointment # references Appointment

Appointment(appointment #: int, date: char(20), time:char(10), **PID**: int, **DID:** int)

Primary Key: appointment #

Foreign Key:

PID, references Patient

DID references Doctor

PID, DID cannot be NULL

treats(**DID**: int, **PID**: int, **Prescription#**: int)

Primary Key: DID, PID, Prescription #

Foreign Key:

DID references Doctor

PID references Patient

Prescription# references Prescription

Patient(ID: int, email: char(20), Name: char(20), HealthCard#: char(20), phone#: char(10))

Primary Key: ID

Alternate Key: email, HealthCard#

Prescription(Prescription#: int, instruction: char(30), substitutable?: bool)

Primary Key: Prescription#

Contains(**Prescription#**: int, **DIN**: int)

Primary Key: Prescription#, DIN

Foreign Key: Prescription#, DIN

- **Prescription**# references Prescription
- **DIN** references Medicine

Medicine(DIN: int, brand: char(20), ingredients: char(20), side effect: char(80))

Primary Key: DIN