DBMS Lab Assignment 4

WEB APP DEVELOPMENT

HOSPITAL MANAGEMENT SYSTEM

Group Name :-	Brogrammers	
Group Members:-	Chenna Keshava Reddy	20CS10014
	Ragala Abhishek	20CS10048
	Sailada Vishnu Vardhan	20CS10051

System Design:

In this software we have developed some forms. The brief description about them is as follow:-

Front Desk Operator:

The operator module handles various functionalities for the Front Desk Operators such as registering patients, discharging patients etc

- Patient Registration
- Book Appointment
- Book Room
- Patient Discharge.
- Schedule Test

Admin:

This module handles all the administrative details such as deleting or adding new users, etc.

- Manage Users
- Manage Doctors
- Manage Operators

Doctor:

This module handles all the doctor details such as enquiring patient information etc.

- My Appointments
- Manage Patients
- Search Patient Info
- Record Prescriptions
- Past Appointments and Prescriptions

Data Entry Operator:

The operator module handles various functionalities for the Data Entry Operators such as testing patients,sending emails etc.

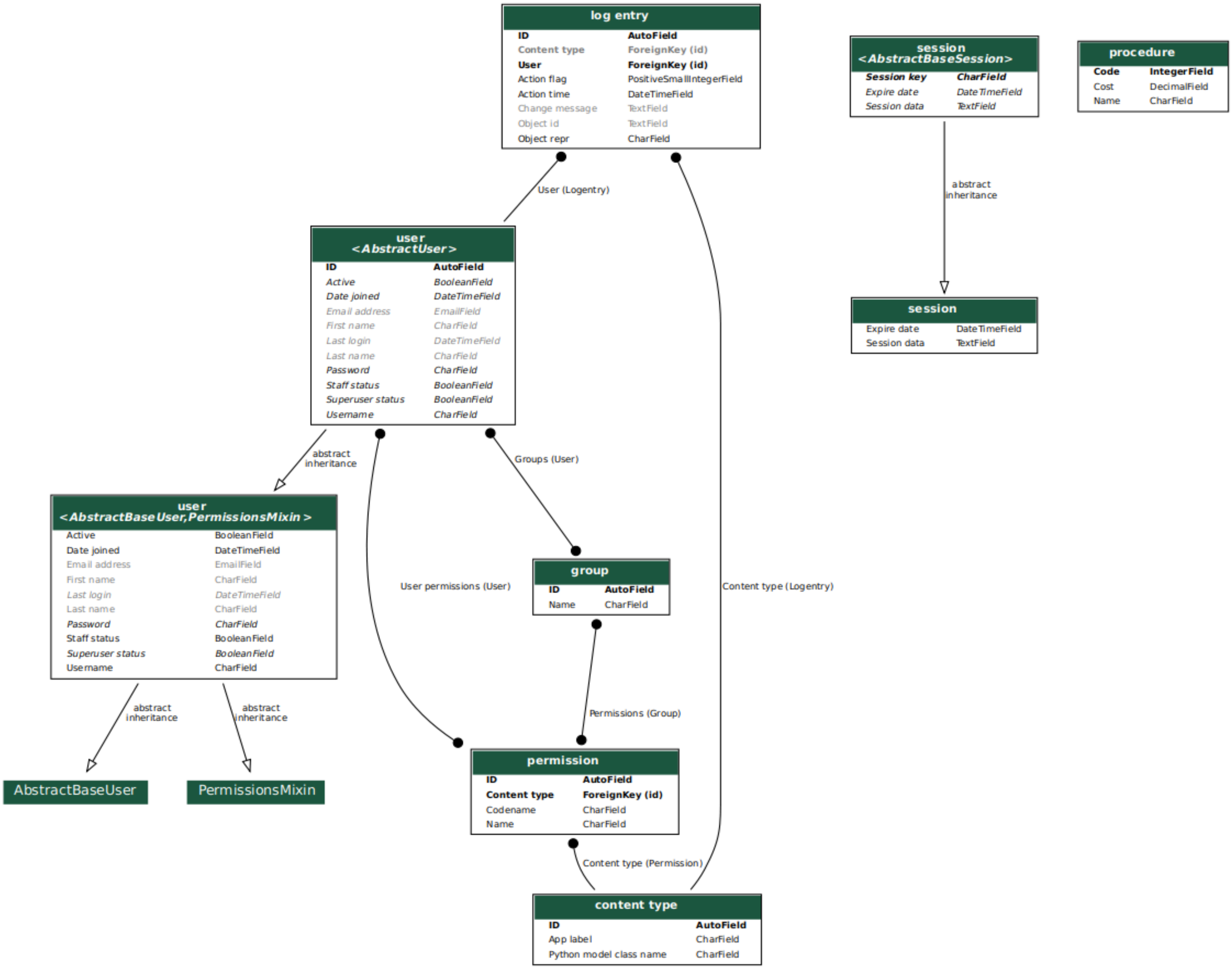
- Patient Testing
- Record Treatment Details
- Email Weekly Reports to Respective Doctors
- Immediate Email to Respective Doctors if Urgent

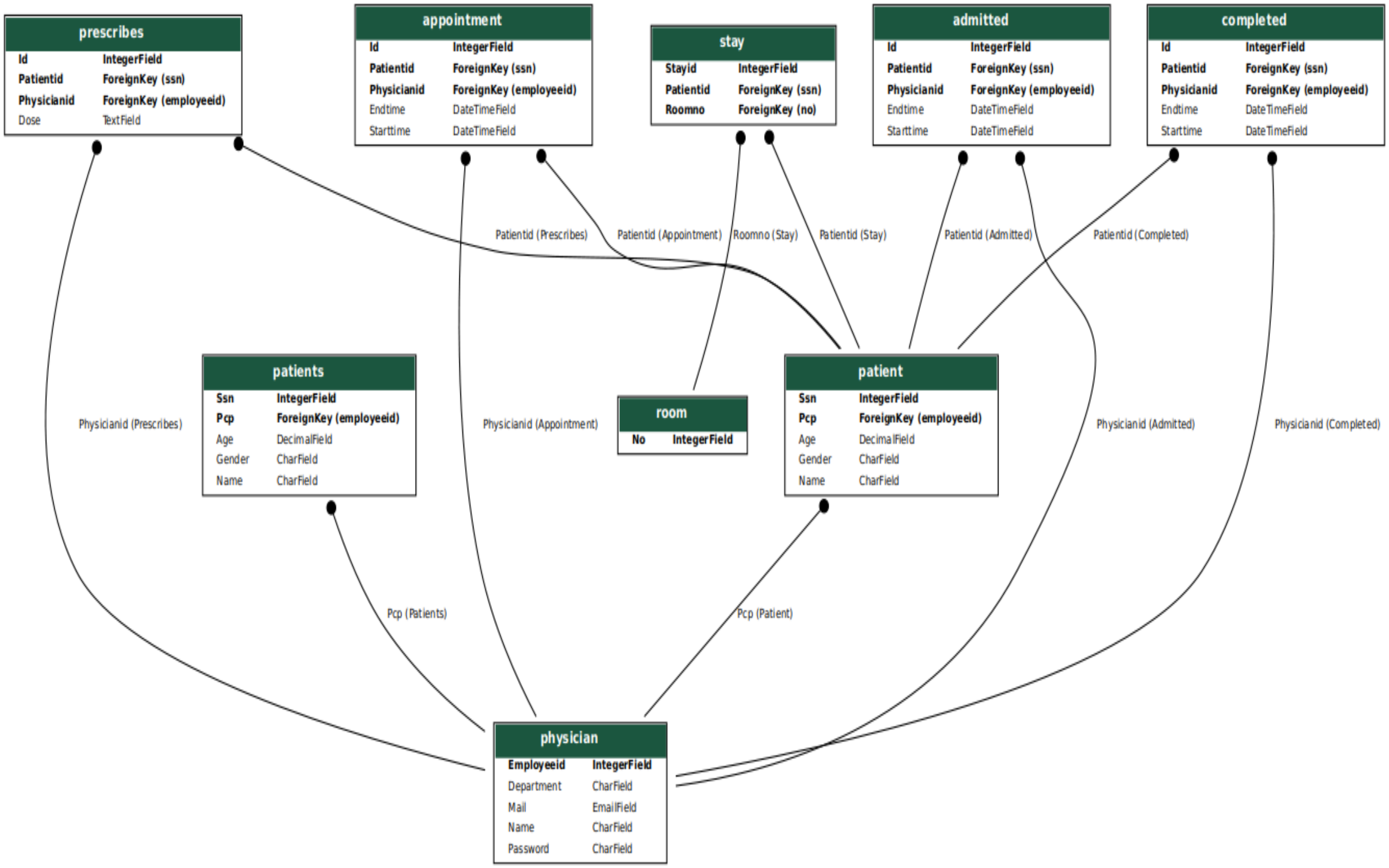
Software Requirements:

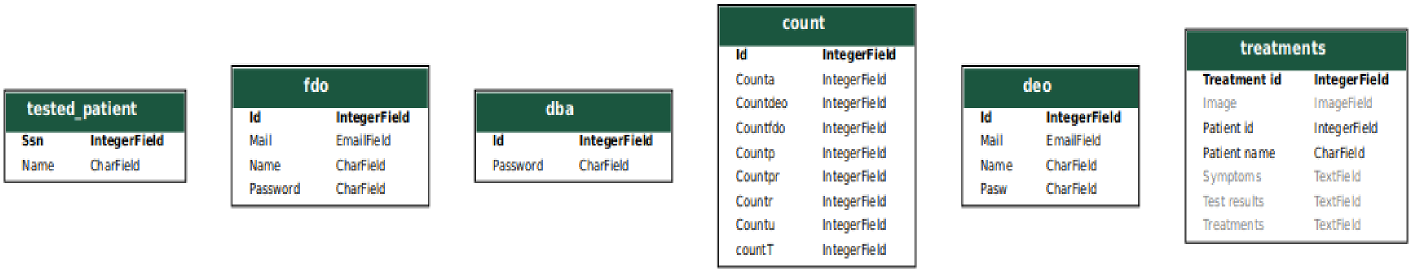
Web Technologies	:	HTTP
Language	:	HTML, JavaScript, CSS (Front End) Django (Back End)
Database	:	PostgresSQL
Operating System	:	WINDOWS, Linux, MacOS

ANALYSIS

DESIGN

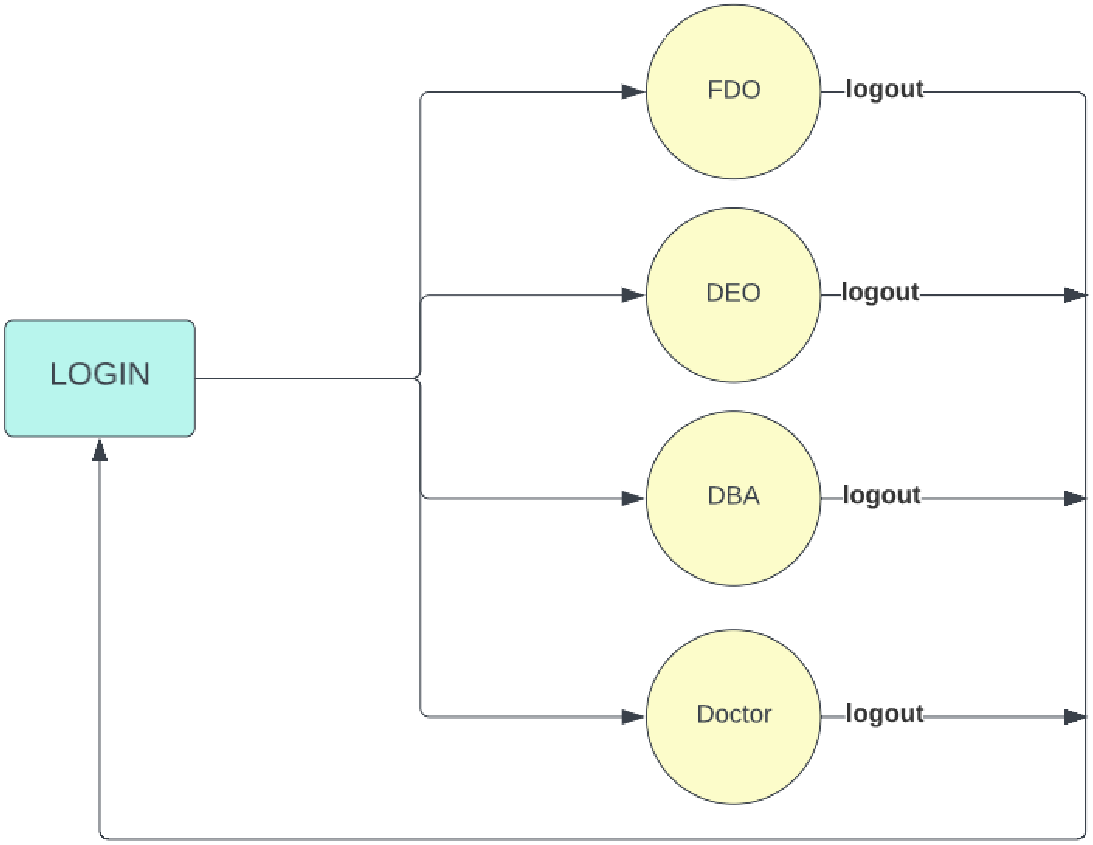



prescribes
Id IntegerField
Patientid ForeignKey (ssn)
Physicianid ForeignKey (employeeid)
Dose TextField
appointment
Id IntegerField
Patientid ForeignKey (ssn)
Physicianid ForeignKey (employeeid)
Endtime DateTimeField
Starttime DateTimeField
stay
Stayid IntegerField
Patientid ForeignKey (ssn)
Roomno ForeignKey (no)
admitted
Id IntegerField
Patientid ForeignKey (ssn)
Physicianid ForeignKey (employeeid)
Endtime DateTimeField
Starttime DateTimeField
completed
Id IntegerField
Patientid ForeignKey (ssn)
Physicianid ForeignKey (employeeid)
Endtime DateTimeField
Starttime DateTimeField
Patientid (Prescribes)
Patientid (Appointment)
Roomno (Stay)
Patientid (Stay)
Patientid (Admitted)
Patientid (Completed)
patients
Ssn IntegerField
Pcp ForeignKey (employeeid)
Age DecimalField
Gender CharField
Name CharField
room
No IntegerField
patient
Ssn IntegerField
Pcp ForeignKey (employeeid)
Age DecimalField
Gender CharField
Name CharField
Physicianid (Prescribes)
Physicianid (Appointment)
Physicianid (Admitted)
Physicianid (Completed)
Pcp (Patients)
Pcp (Patient)
physician
Employeeid IntegerField
Department CharField
Mail EmailField
Name CharField
Password CharField

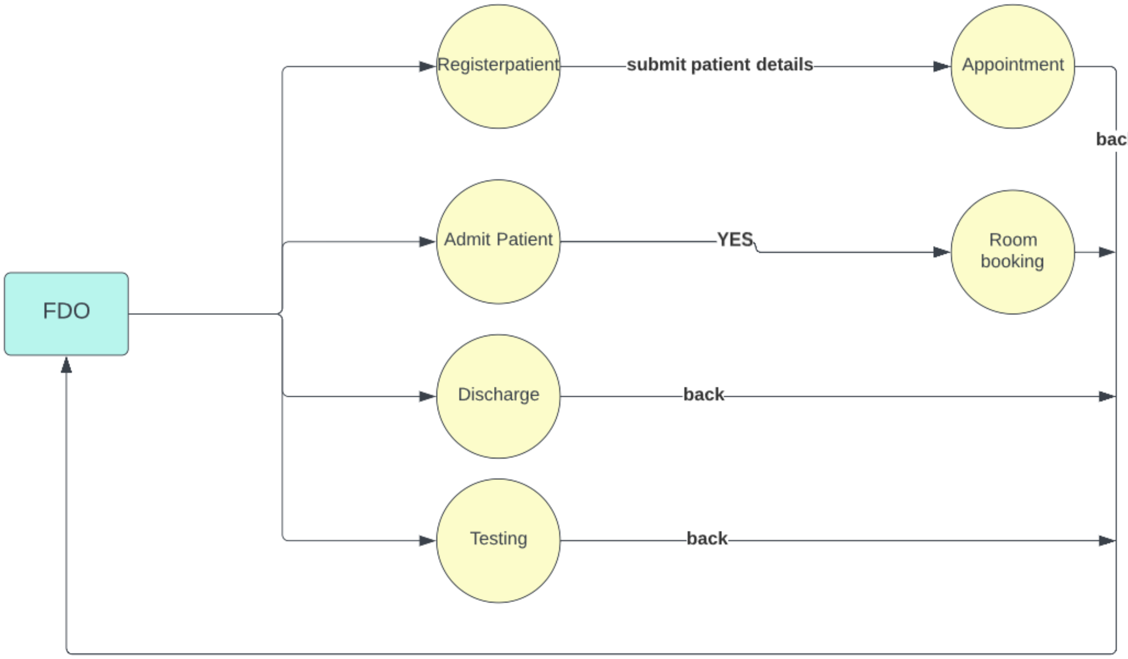

tested_patient
Ssn IntegerField
Name CharField
fdo
Id IntegerField
Mail EmailField
Name CharField
Password CharField
dba
Id IntegerField
Password CharField
count
Id IntegerField
Counta IntegerField
Countdeo IntegerField
Countfdo IntegerField
Countp IntegerField
Countpr IntegerField
Countr IntegerField
Countu IntegerField
countT IntegerField
deo
Id IntegerField
Mail EmailField
Name CharField
Pasw CharField
treatments
Treatment id IntegerField
Image ImageField
Patient id IntegerField
Patient name CharField
Symptoms TextField
Test results TextField
Treatments TextField

Work Flows:

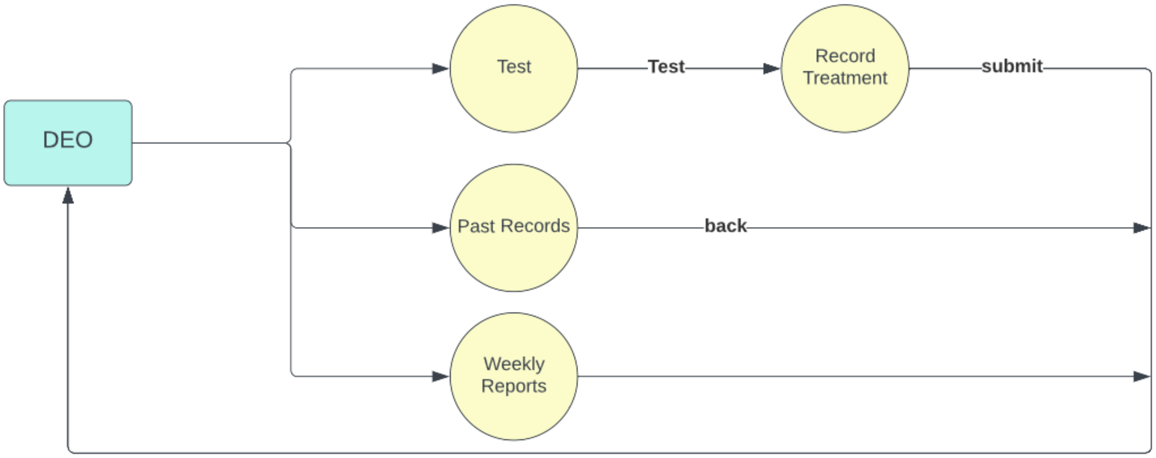
Login:



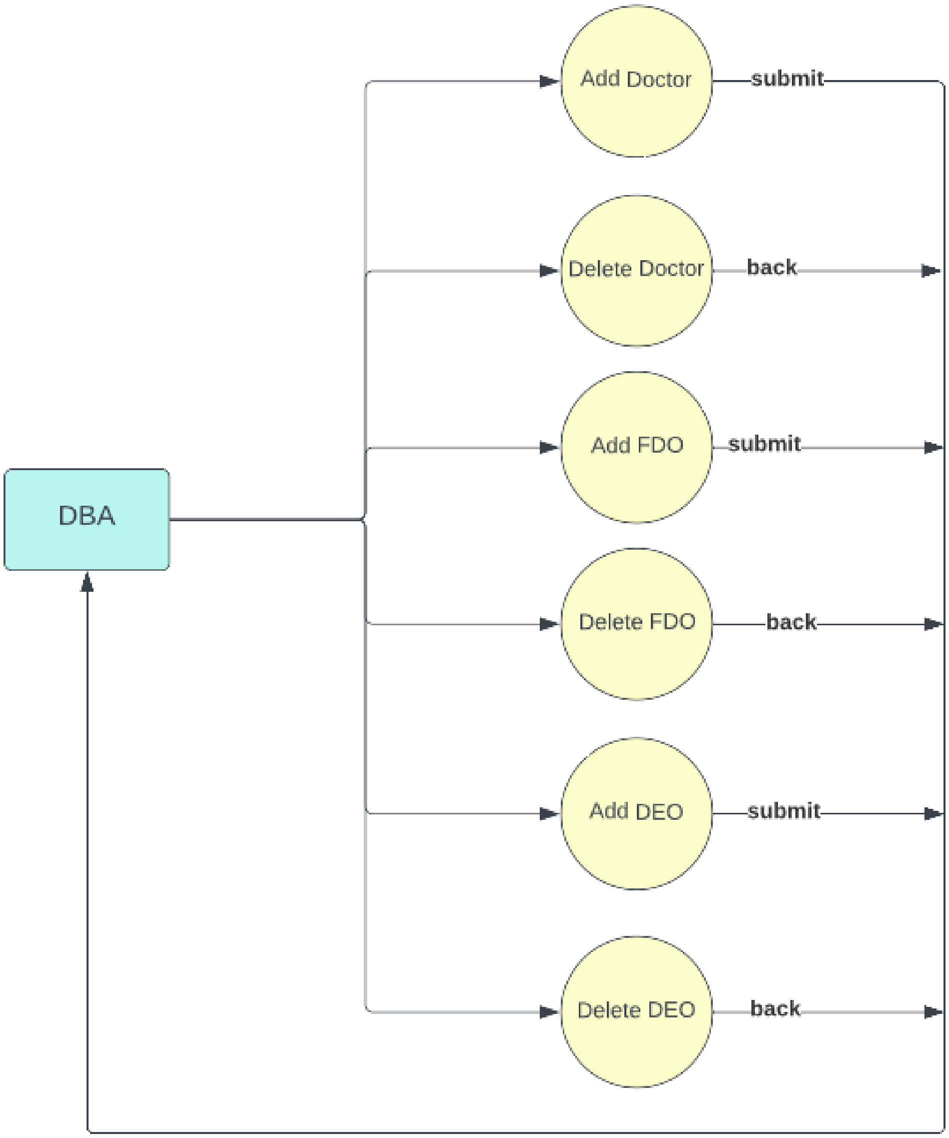
Front_Desk_Operator:



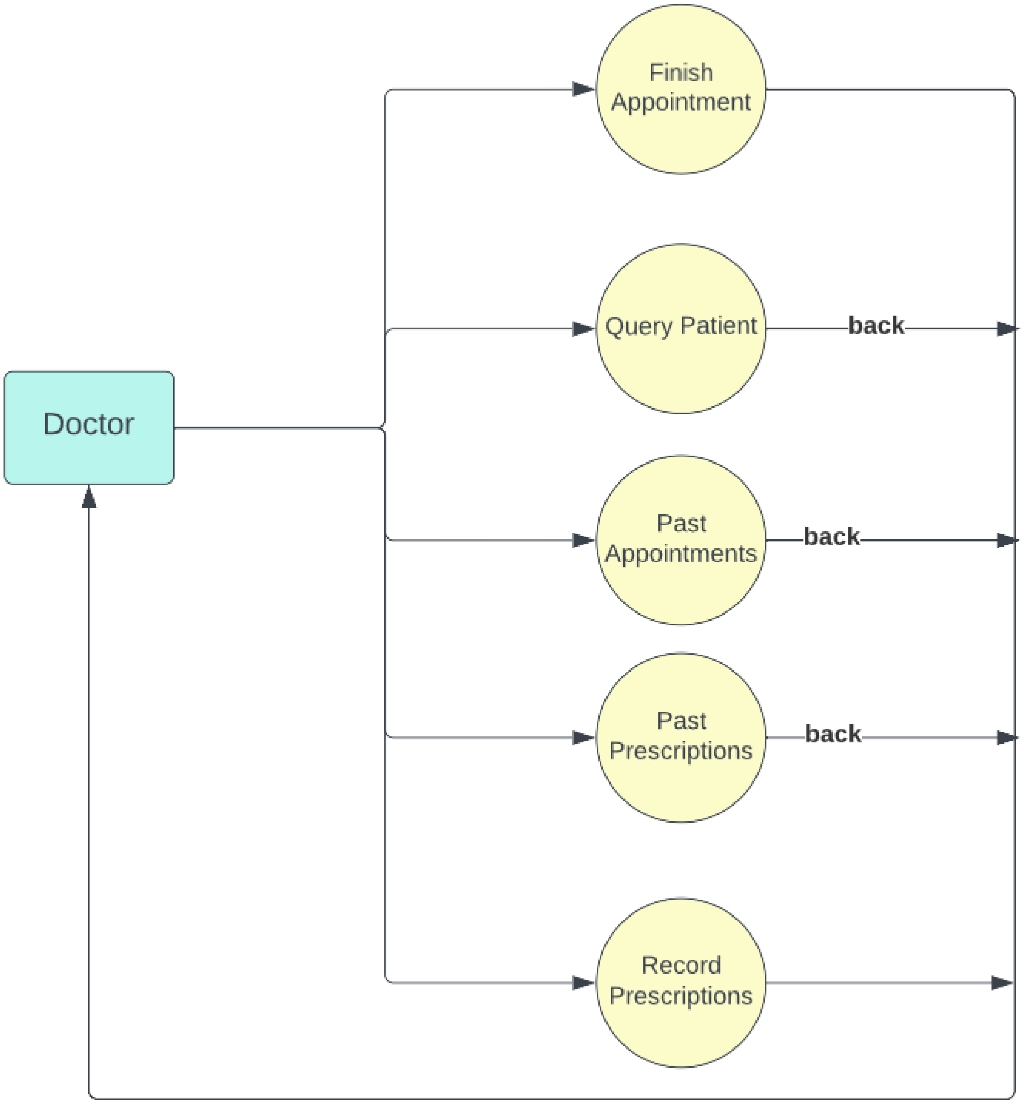
Data_Entry_Operator:



Data_Base_Administrator:



Doctor:


Doctor
Finish Appointment
Query Patient
back
Past Appointments
back
Past Prescriptions
back
Record Prescriptions

OUTPUT SCREENS

Home Page:

Forgot Password:

Reset Password:

FDO login:

Patient Registration:

Patient Discharge:

Patient Test Scheduling:

Patient Room booking:

DEO Login:

Past Records:

Weekly Reports:

Testing :

Urgent email

Save record

DBA Login:

Add doctor:
Delete Doctor:
Add DEO:
Delete DEO:
Add FDO:
Delete FDO:

Doctor Login:

Past appointments:
Past prescriptions:
Record prescription:
Query Patient:
Present appointments:

Django_Workflow:

Button in html page:

```
<form action="{% url 'testing_patient' %}">
    <button class="btn btn-primary" type="submit">Schedule_test</button>
</form>
```

Action for the above button:

```
def testing_patient(request):
    result = patients.objects.all()
    return render(request,'testing_patient.html',{'data': result})
```

Url to link this action and button:

```
path('testing_patient/',views.testing_patient,name = 'testing_patient'),
```

Models in Django:

```
class procedure(models.Model):
    code = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=100)
    cost = models.DecimalField(max_digits=10, decimal_places=2)
    class Meta:
        db_table = "procedure"
```

Accessing and Updating Psql Database:

Create a new object in tested_patient Class:

```
p = tested_patient(
       ssn = sta.ssn,
       name = sta.name,
   )
   p.save()
```

Update a object in a class:

```
pasw = request.POST.get('password')
obj = physician.objects.get(employeeid = us)
obj.password = pasw
obj.save()
```

Delete a object in a class:

```
sta = patients.objects.get(ssn = result_id)
sta.delete()
```

Sending a Email:

```
           s = 'patient_id’

           message = 'The user details are: {}'.format(s)

           email = EmailMessage(

           'User details',

           message,

           settings.EMAIL_HOST_USER,

           [id2.mail],

           # reply_to=['another@example.com'],

           headers={'Message-ID': 'foo'},

           )

           email.send()
```

Opening Image:

```
def view_xray(request):

   id = request.POST.get('result_id')

   print(id)

   T =  treatments.objects.get(treatment_id=id)

   response = HttpResponse(content_type='image/jpeg')

   response.write(T.image.read())

   return response
```

Storing Image:

```
image_file = request.FILES.get('x_ray')

STR = 'image'+str(C.countT)+'.jpg'

if image_file:

        image_data = BytesIO(image_file.read())

        uploaded_file = InMemoryUploadedFile(image_data, None, STR,
'image/jpeg', image_data.tell(), None)


 treatment_obj = treatments(

           treatment_id = C.countT,

           patient_id=patient_id,

           patient_name=patient_name,

           symptoms=symptoms,

           treatments=treatment,

           test_results=test_results,

           image=uploaded_file

       )

treatment_obj.save()
```