SOFTWARE ARCHITECTURE DOCUMENT

Project Title: SeeAnyGP

Project Description

You are a patient trying to book an appointment with your GP (general practitioner - also family doctor) but you keep getting disappointed after queuing for ages on the surgery appointments line as all appointments have been booked. You think - what if I could get an appointment with any GP without having to join the queue on the phone? This is where SeeAnyGP comes in: a service where you can see/access available GPs in your location and book appointments to fit your schedule.

Overview

Due to staff shortages, it is increasingly becoming difficult to get an appointment with a GP. There are many people in the queue before you when you ring your surgery at 08:01! You may end up not getting an appointment on that day and have to queue again the next day. Most times you don't see the same GP as before when you get an appointment. This can be frustrating. Considering this, SeeAnyGP was developed to make the appointment process easier and to allow you to choose to see the same GP as much as possible if you wish.

In this project, there will be three people interacting with the system: the patients(users), doctors and the admin. The patients will register and log in using their saved login details. They will be prompted to fill in any necessary information or attach documents needed for their profile. They will be shown a list of available GPs in their area so that they can book appointment slots with their chosen GP. The admin will add doctors(GPs) who work with SeeAnyGP to the application. They can amend appointments, and manage patient and doctor profiles as required. GPs can view their appointments and manage their profiles.

Terms:

1. **Patient:** The person who wants to make an appointment with a GP.

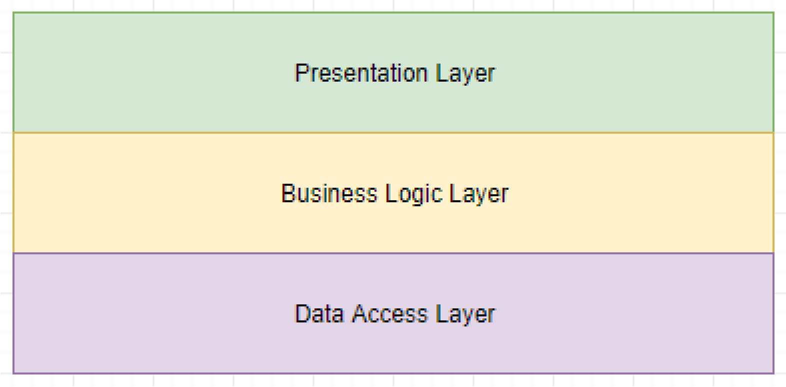
2. **GP:** The doctor who sees the patient during the appointment.

FUNCTIONAL REQUIREMENT & MINIMUM VIABLE PRODUCT (MVP)

- **The patient** registers on the platform.
- Then submits details to complete their profile.
- They can view available doctors and book appointments with them.
- **The GP/doctor** is onboarded internally on the platform and is a working member of the company.
- They can view the list of patients booked with them and their profiles.

APPLICATION ARCHITECTURE

(3) Three-tier Architecture

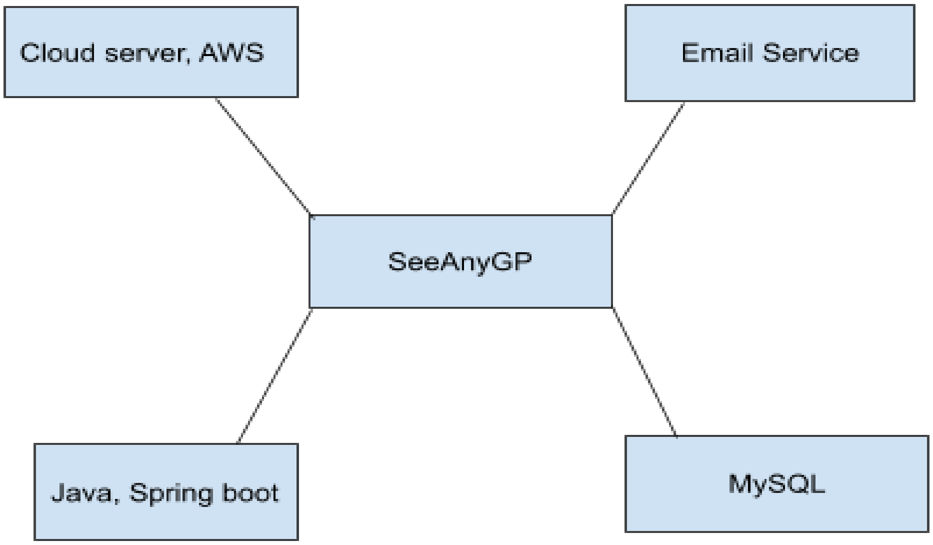


Presentation Layer: The layer where users interact with the application and where data will be made visible to the users. It acts as an interface between the user and the application and it will employ a frontend programming framework.

Business Logic: It acts as an intermediate between the Presentation and the Data Access Layer. It uses Java Runtime and Spring boot framework.

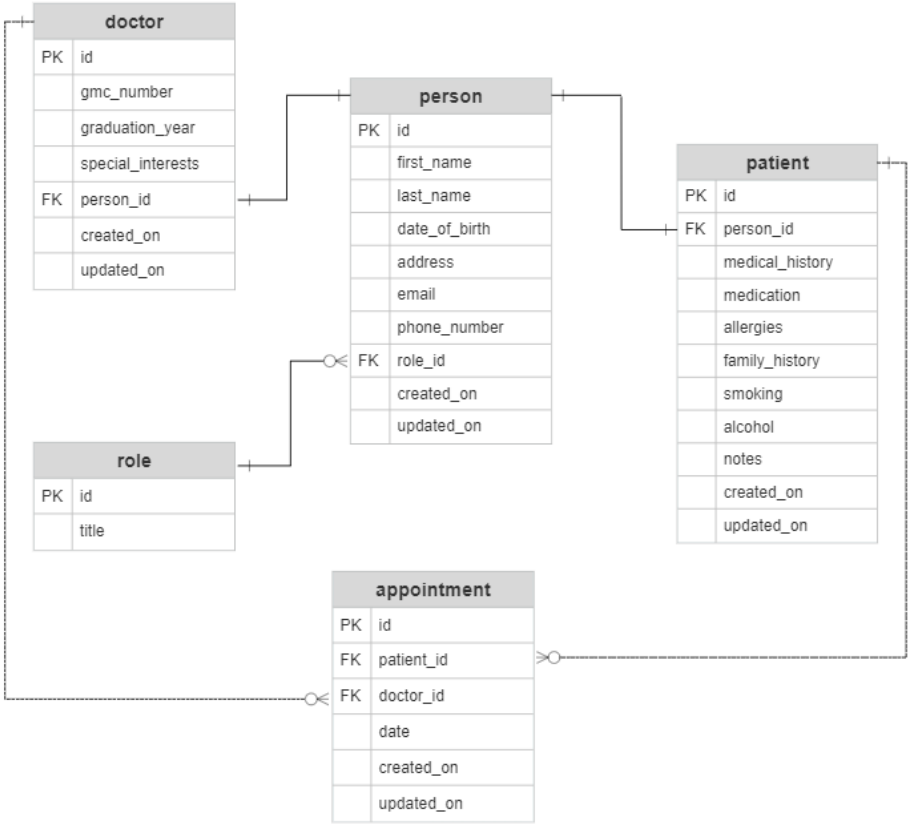
Data Repository and Access: The layer where the data is persisted & managed. It uses MySQL.

SYSTEM CONTEXT DIAGRAM



SEQUENCE DIAGRAM...............

ENTITY RELATION DIAGRAM (ERD)



doctor	
PK	id
	gmc_number
	graduation_year
	special_interests
FK	person_id
	created_on
	updated_on

person	
PK	id
	first_name
	last_name
	date_of_birth
	address
	email
	phone_number
FK	role_id
	created_on
	updated_on

patient	
PK	id
FK	person_id
	medical_history
	medication
	allergies
	family_history
	smoking
	alcohol
	notes
	created_on
	updated_on

role	
PK	id
	title

appointment	
PK	id
FK	patient_id
FK	doctor_id
	date
	created_on
	updated_on

List of Tables

- person
- role
- patient
- doctor
- appointment

person: The person table stores the personal details of new users (both patients and doctors(GPs) on registration. It has a one-to-one relationship with the patient and doctor tables.

role: This table details the list of enumerated or predefined roles or privileges that can be assigned to a user, e.g patient, doctor or admin. It has a one-to-many with the person table because many persons can have one role.

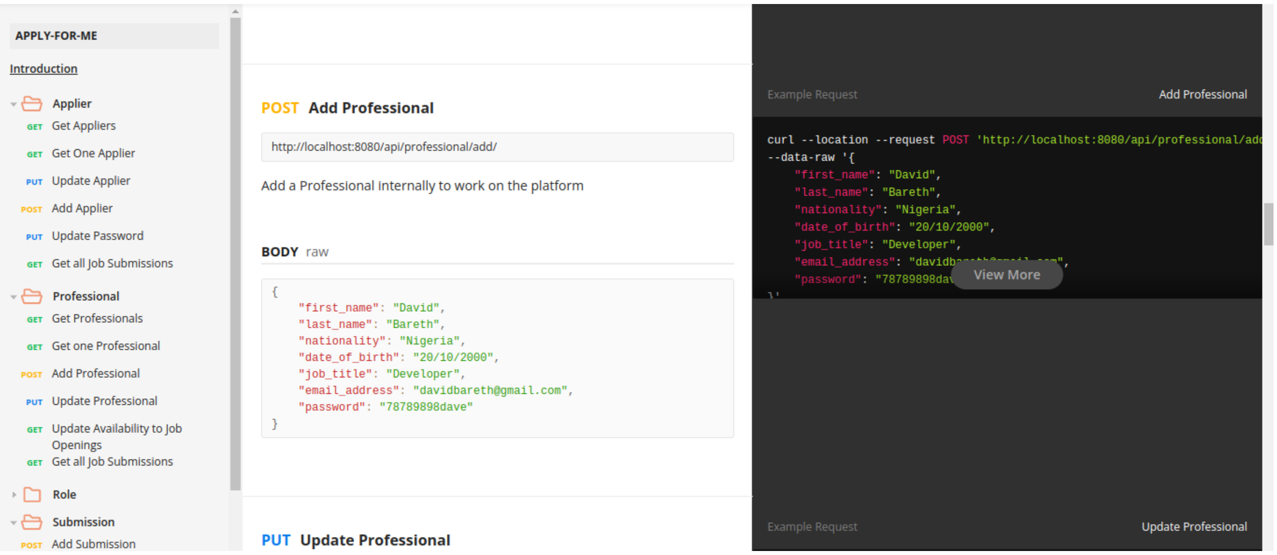
patient: This table holds all relevant medical details of the patient. Information stored in this table is used by the doctor to know the medical background of the patient. It has a one-to-one relationship with the person table.

doctor: This table holds all relevant details of the GP. Information stored in this table is used by patients to know the GP's qualifications. It has a one-to-one relationship with the person table.

appointment: This table holds information about all appointments. It has a many-to-one relationship with the patient and doctor tables.

API Documentation (To be updated)

Postman Documentation:



POST **Add Professional**

http://localhost:8080/api/professional/add/

Add a Professional internally to work on the platform

BODY raw

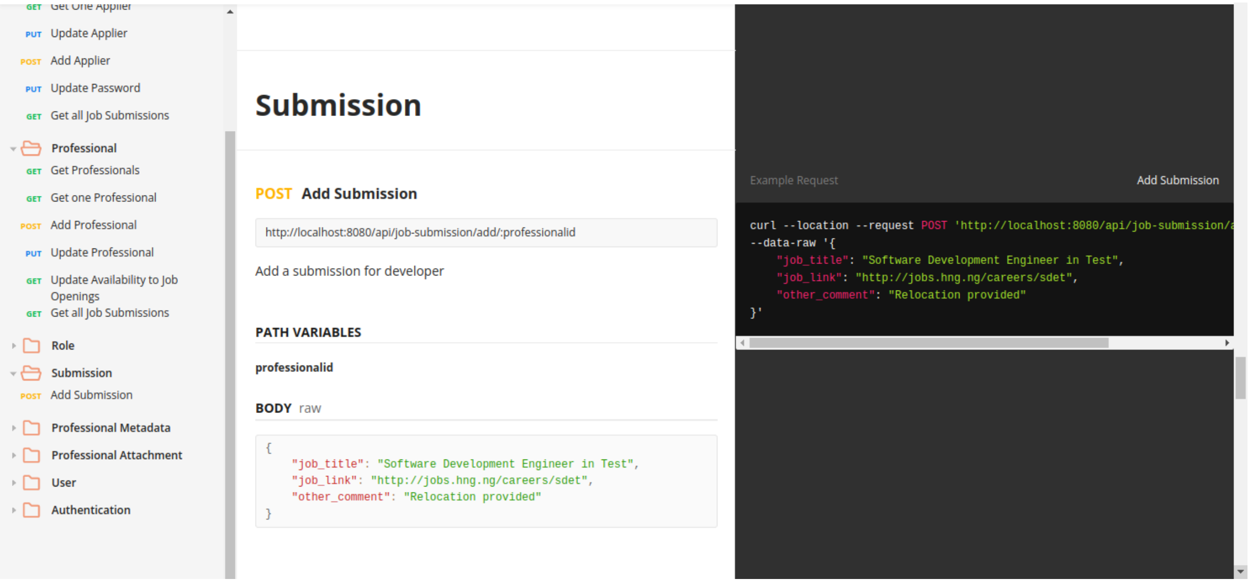
```
{
    "first_name": "David",
    "last_name": "Bareth",
    "nationality": "Nigeria",
    "date_of_birth": "20/10/2000",
    "job_title": "Developer",
    "email_address": "davidbareth@gmail.com",
    "password": "78789898dave"
}
```

PUT **Update Professional**

Example Request — Add Professional

```
curl --location --request POST 'http://localhost:8080/api/professional/add
--data-raw '{
    "first_name": "David",
    "last_name": "Bareth",
    "nationality": "Nigeria",
    "date_of_birth": "20/10/2000",
    "job_title": "Developer",
    "email_address": "davidbareth@gmail.com",
    "password": "78789898dav
}'
```

Example Request — Update Professional



Submission

POST **Add Submission**

http://localhost:8080/api/job-submission/add/:professionalid

Add a submission for developer

PATH VARIABLES

professionalid

BODY raw

```
{
    "job_title": "Software Development Engineer in Test",
    "job_link": "http://jobs.hng.ng/careers/sdet",
    "other_comment": "Relocation provided"
}
```

Example Request — Add Submission

```
curl --location --request POST 'http://localhost:8080/api/job-submission/a
--data-raw '{
    "job_title": "Software Development Engineer in Test",
    "job_link": "http://jobs.hng.ng/careers/sdet",
    "other_comment": "Relocation provided"
}'
```

Patient’s Journey

1. Amina is a patient struggling to get an appointment with her GP. She registers on the SeeAnyGP platform. During the registration, she provides the following details:
    - First name
    - Last name
    - Date of birth
    - Address
    - Email
    - Phone number
    - Username and Password
2. Amina then proceeds to log in using her username and password as login credentials.
3. She submits her profile details. After which they get saved to the database.
    - Medical history
    - Medication
    - Allergies
    - Family history
    - Smoking status
    - Alcohol
4. She is then able to view doctors' profiles, and availability and book appointments.

GP/Doctor’s Journey

Bash is a GP who has a list of appointment slots that can be booked.

1. Bash registers on the SeeAnyGP platform.
2. He logs in using his username and password as login credentials.
3. He is able to see a list of patients booked with him. He contacts the patients at the right time for their appointments.
4. He logs out after seeing all his patients.