CSCI 3308 Milestone 4

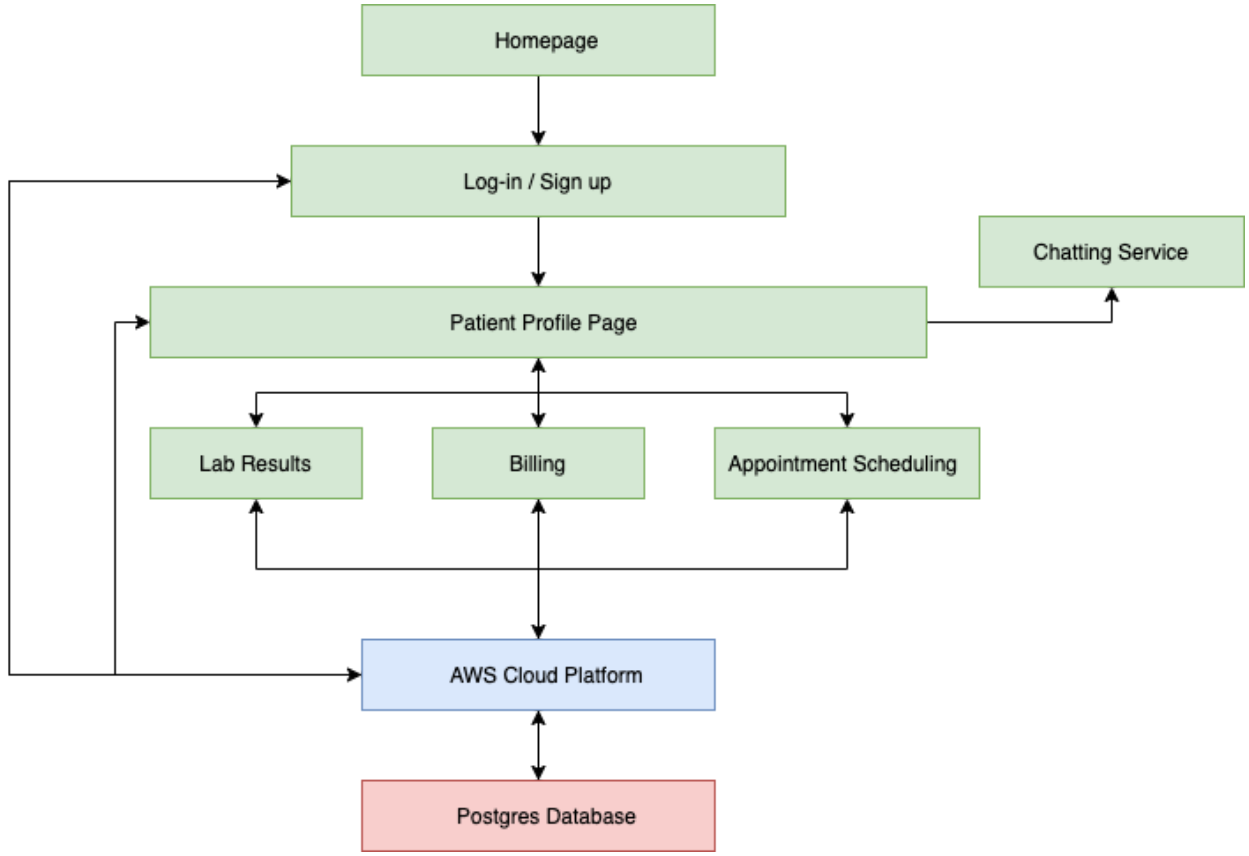
Team Number: 103-2
Team Name: Team = [hip, hip, array];
Team Members: Nagisa Her, Tobi Jacobson, Andrew King, and Sahand Setareh

---

Revised List of Features: (Remained unchanged)

- User login
- Appointment Scheduling
- Lab Results
- Patient-to-Provider Chatting Service
- Prescription Management and Ordering
- Billing and Insurance

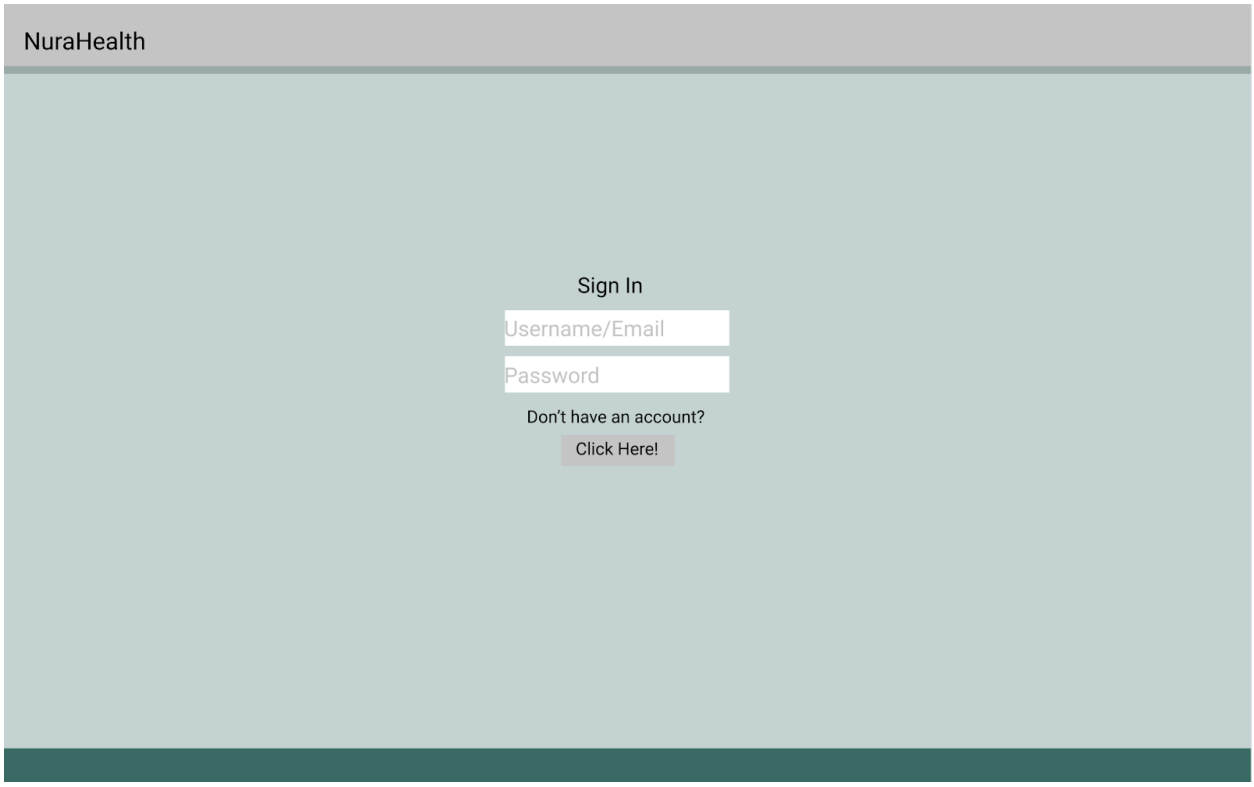
Architecture Diagram:



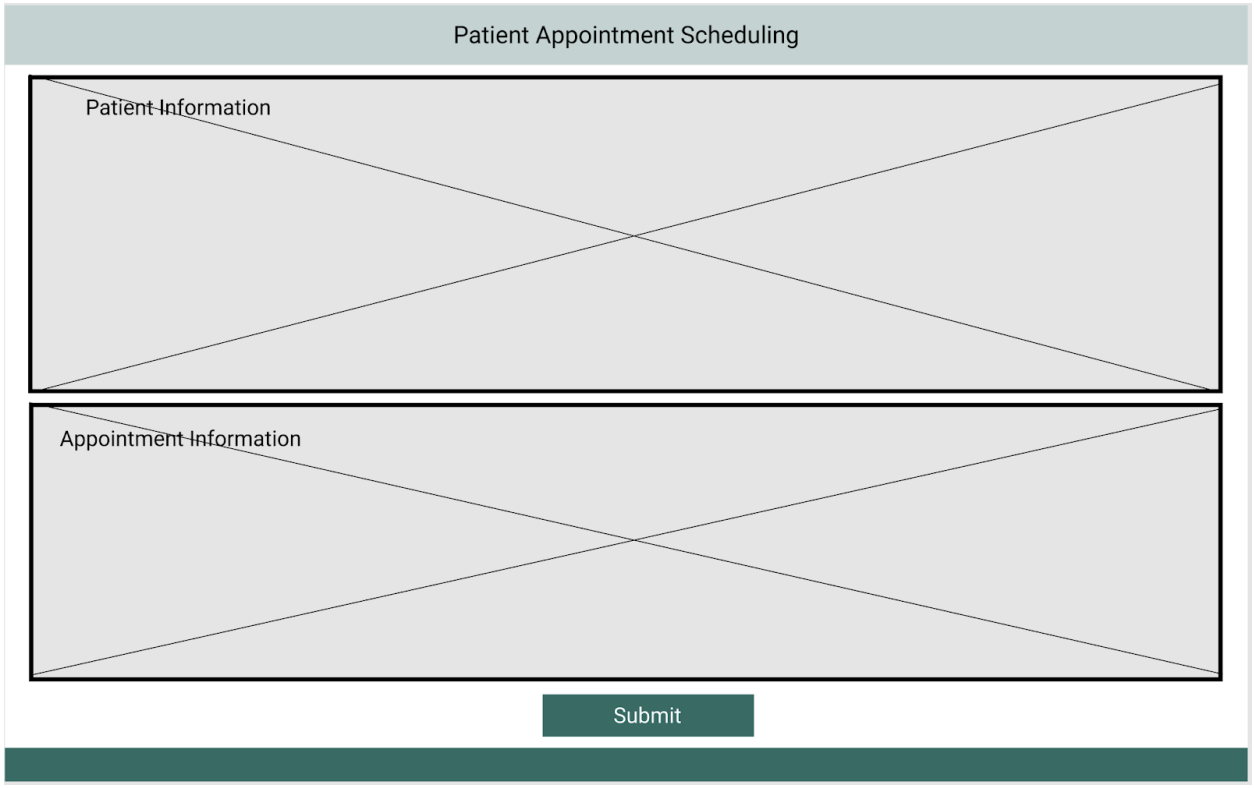
- We will be hosting our front-end, integration layer, and backend via AWS IoT Core cloud services.
- Protocols used by AWS IoT Core:
    - HTTP: HTTP allows intermediate network elements to enable communications between clients and servers.
    - MQTT: Is a lightweight protocol for transmitting data between machines.
    - WebSockets: Websocket is a computer communications protocol that operates between HTTP ports 80 and 443 as well as supporting HTTP proxies, hence it works in unison with HTTP.
    - TLS: Transport Layer Security are cryptographic protocols used for communication security between client and a server.

Front End Design:

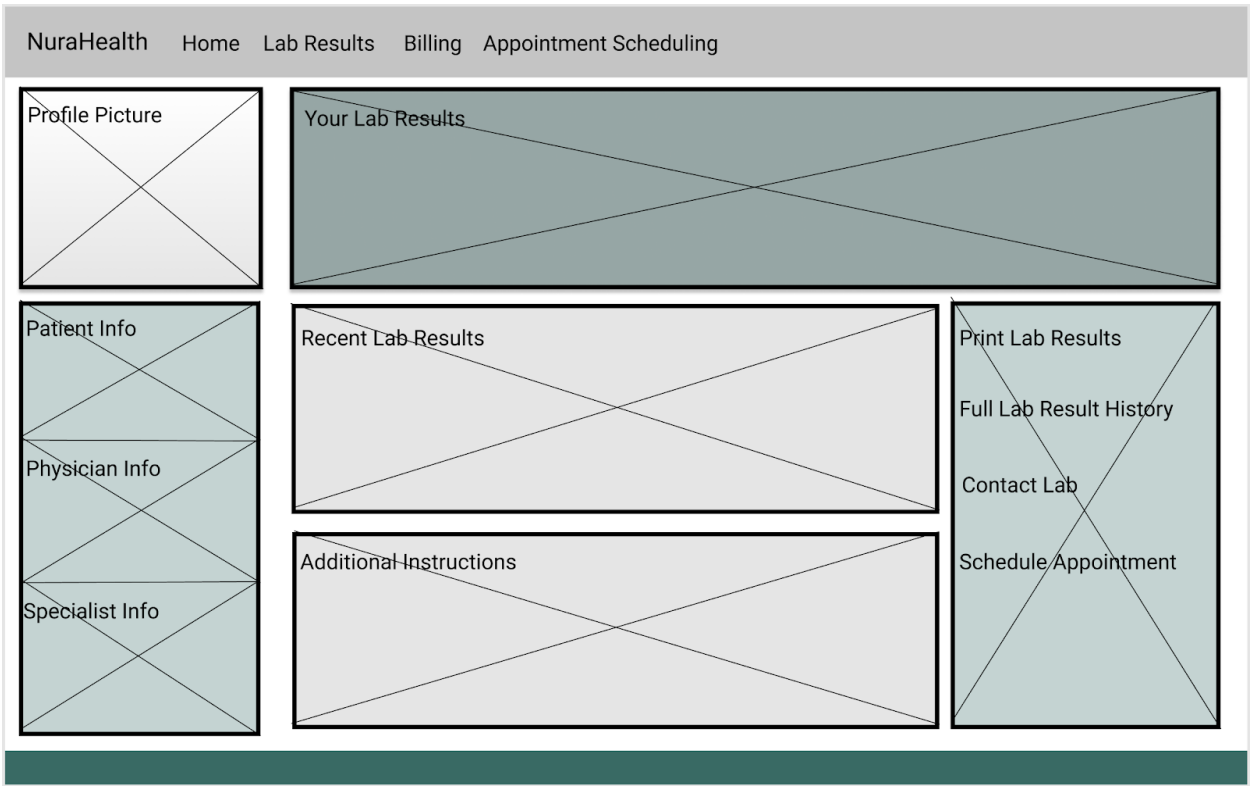
- *User Login*



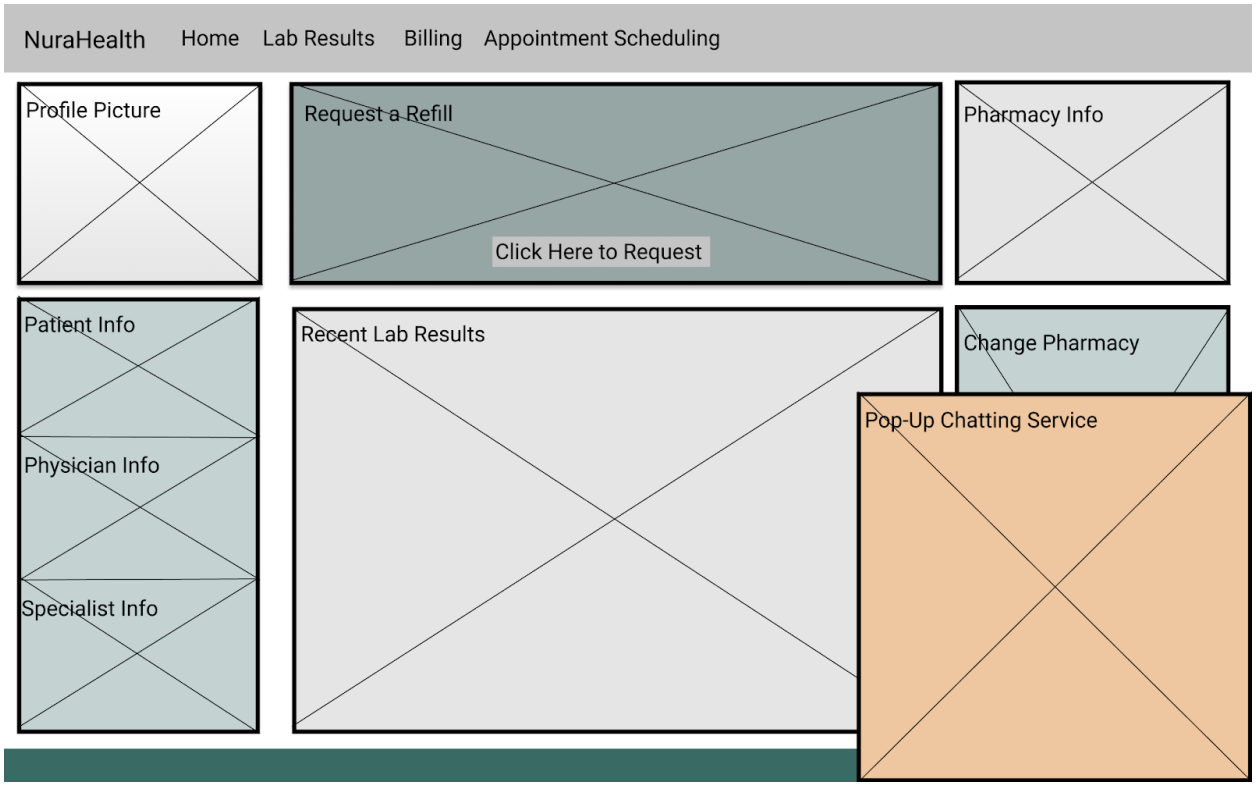
- *Appointment Scheduling*



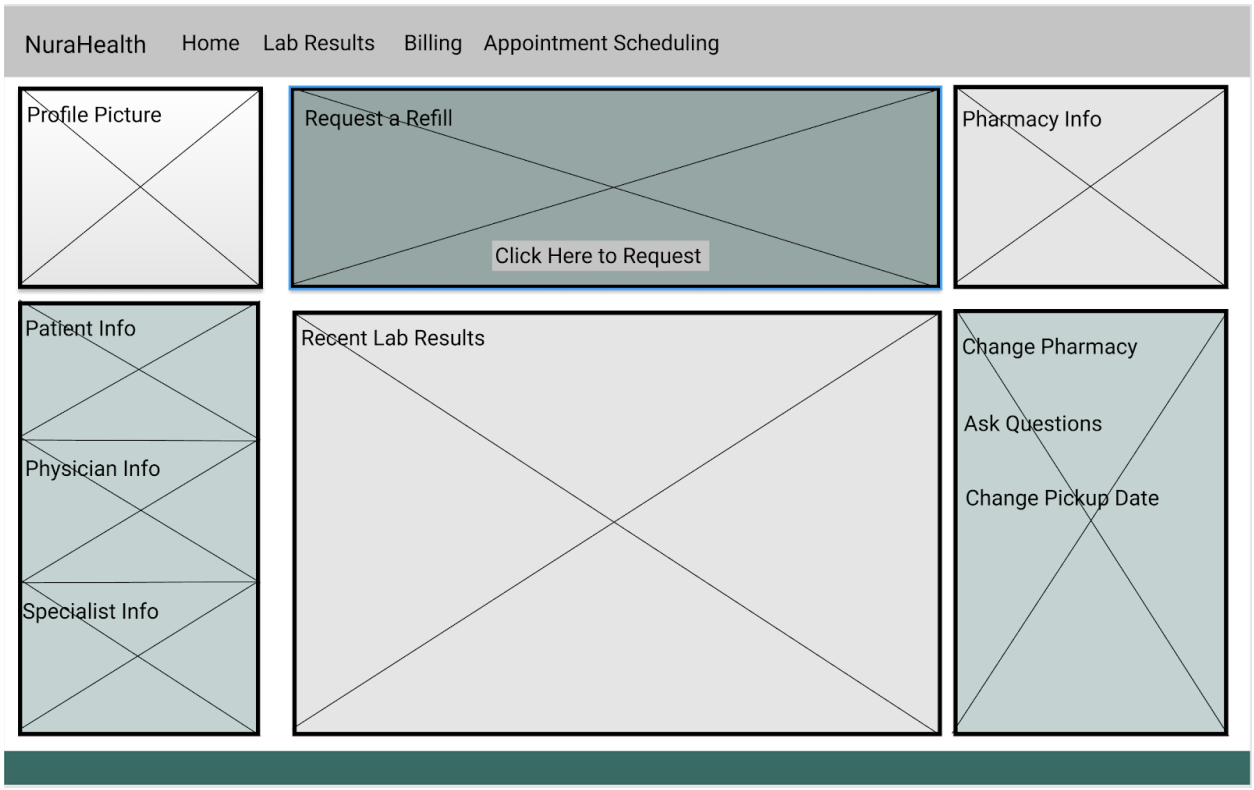
- *Lab Results*



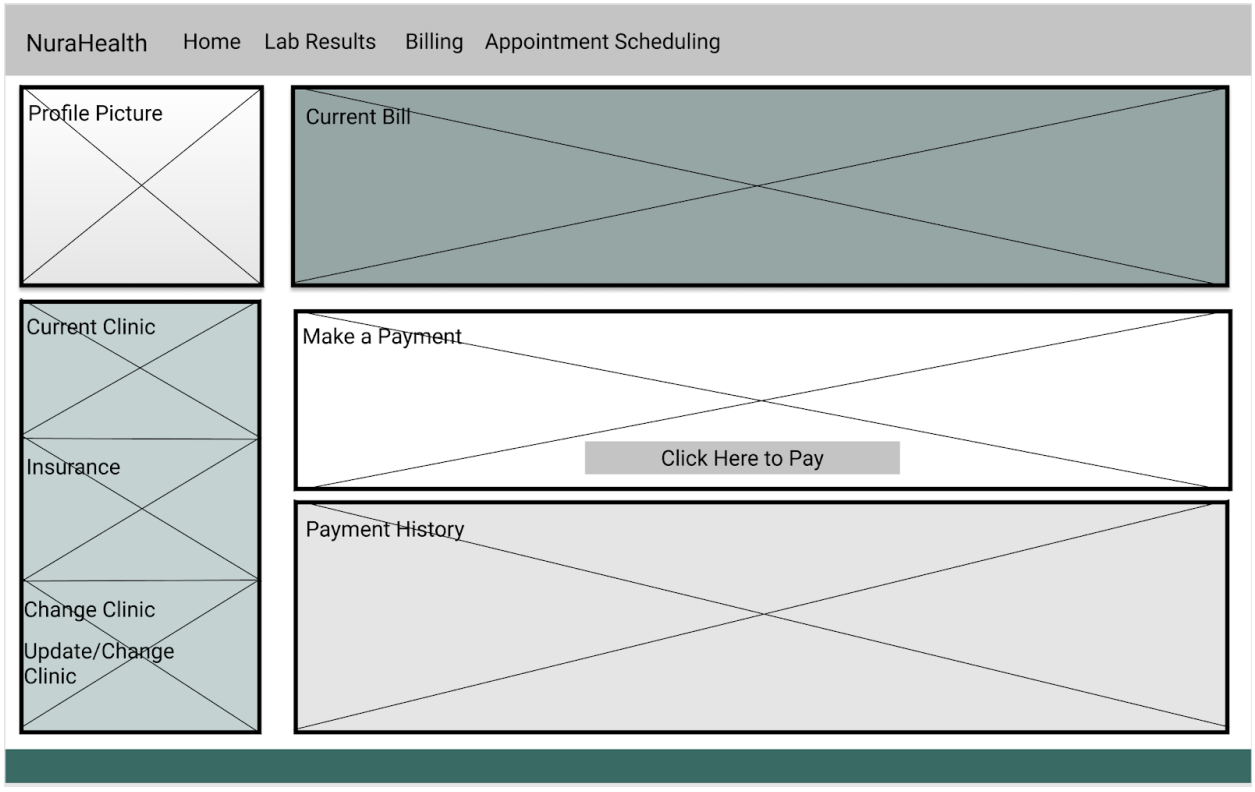
- Patient-to-Provider Chatting Service



- Prescription Management and Ordering



- Billing and Insurance



Web Service Design:

Our website will employ Amazon Web Services and associated APIs in order to pull and present information from our Postgres Database. We aren't too familiar with how this will entirely function in scope, but we anticipate that AWS will be a perfect fit to host our website and allow information that the user profile consists of, such as billing and insurance, address, lab results, appointment scheduling, profile pictures, etc to be seamlessly pulled from our database and presented on the front end. User log-in and credentials will also be managed through these API gateways.

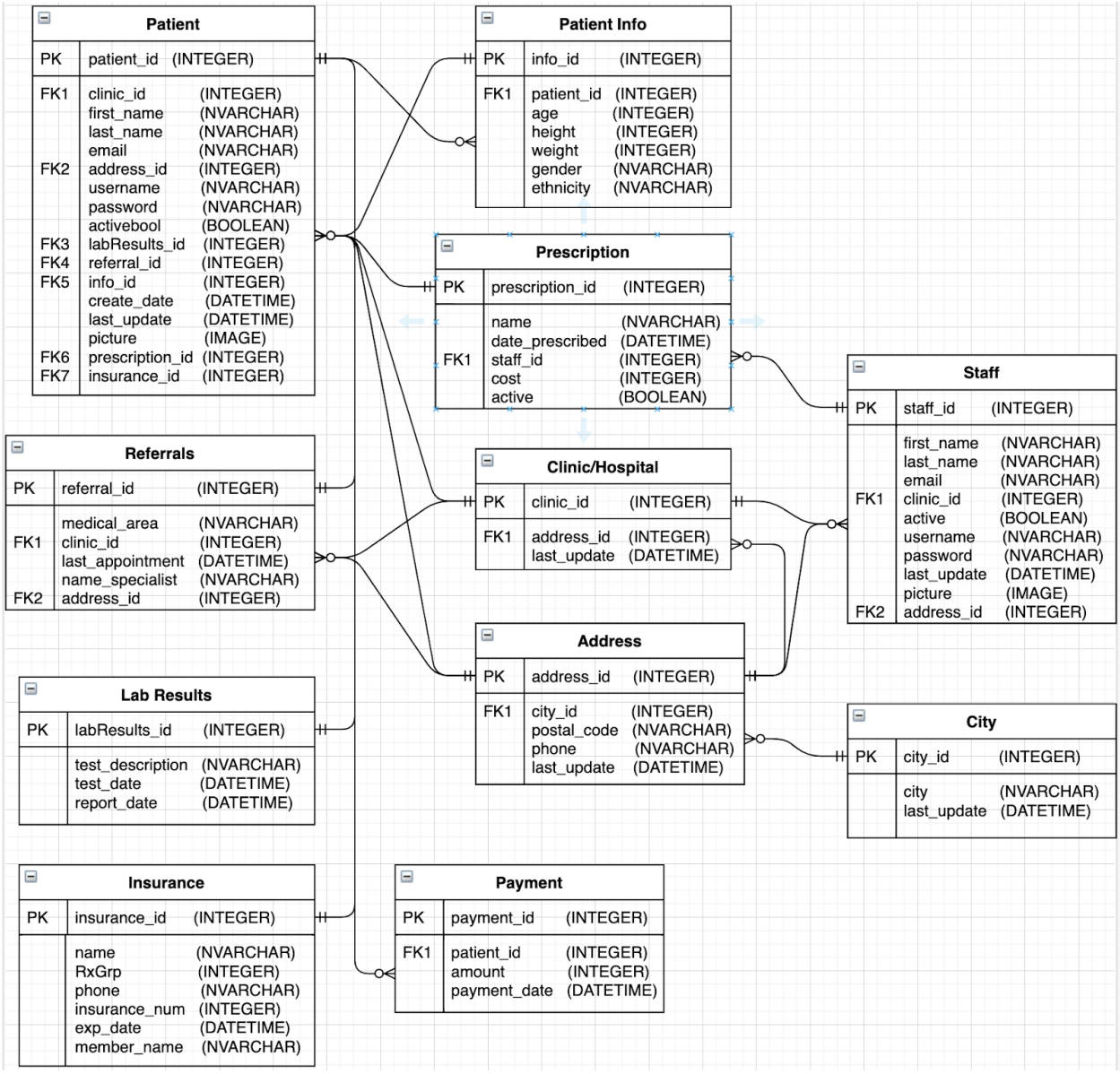
Database Design:

The specific DBMS technology that is being used to store the application data is PostgreSQL. The application's database will hold various tables containing patient information as well as hospital and provider information. Examples of the specific information stored would be patient age or hospital address. The application will pull from these databases in order to populate each page in the application based on the user that signs in. In addition to this, the database will also take in new data when a patient signs up for the application service.

The types of data being stored in the database are:

- INTEGER
- NVARCHAR
- BOOLEAN
- DATETIME
- IMAGE

Below is schema showing the relationship between the tables in the database:


Patient
PK patient_id (INTEGER)
FK1 clinic_id (INTEGER)
first_name (NVARCHAR)
last_name (NVARCHAR)
email (NVARCHAR)
FK2 address_id (INTEGER)
username (NVARCHAR)
password (NVARCHAR)
activebool (BOOLEAN)
FK3 labResults_id (INTEGER)
FK4 referral_id (INTEGER)
FK5 info_id (INTEGER)
create_date (DATETIME)
last_update (DATETIME)
picture (IMAGE)
FK6 prescription_id (INTEGER)
FK7 insurance_id (INTEGER)
Patient Info
PK info_id (INTEGER)
FK1 patient_id (INTEGER)
age (INTEGER)
height (INTEGER)
weight (INTEGER)
gender (NVARCHAR)
ethnicity (NVARCHAR)
Prescription
PK prescription_id (INTEGER)
name (NVARCHAR)
date_prescribed (DATETIME)
FK1 staff_id (INTEGER)
cost (INTEGER)
active (BOOLEAN)
Staff
PK staff_id (INTEGER)
first_name (NVARCHAR)
last_name (NVARCHAR)
email (NVARCHAR)
FK1 clinic_id (INTEGER)
active (BOOLEAN)
username (NVARCHAR)
password (NVARCHAR)
last_update (DATETIME)
picture (IMAGE)
FK2 address_id (INTEGER)
Referrals
PK referral_id (INTEGER)
medical_area (NVARCHAR)
FK1 clinic_id (INTEGER)
last_appointment (DATETIME)
name_specialist (NVARCHAR)
FK2 address_id (INTEGER)
Clinic/Hospital
PK clinic_id (INTEGER)
FK1 address_id (INTEGER)
last_update (DATETIME)
Address
PK address_id (INTEGER)
FK1 city_id (INTEGER)
postal_code (NVARCHAR)
phone (NVARCHAR)
last_update (DATETIME)
City
PK city_id (INTEGER)
city (NVARCHAR)
last_update (DATETIME)
Lab Results
PK labResults_id (INTEGER)
test_description (NVARCHAR)
test_date (DATETIME)
report_date (DATETIME)
Insurance
PK insurance_id (INTEGER)
name (NVARCHAR)
RxGrp (INTEGER)
phone (NVARCHAR)
insurance_num (INTEGER)
exp_date (DATETIME)
member_name (NVARCHAR)
Payment
PK payment_id (INTEGER)
FK1 patient_id (INTEGER)
amount (INTEGER)
payment_date (DATETIME)