Proof of Concept (POC)

This POC contains shows the working of backend as well as the android app (client)

The working prototype of the project can be found here and the details are explained as follows:

- The backend is built using Django v3.0.4 and using the djangorestframework v3.11.0

- The populate_database.py file acts as the main script of the project that fetches CDM details from various hospital websites parses them and stores them in the backend database
    - `fetch_original()` function fetches the original CDM details for Penn Presbyterian Medical Center, parses the details and adds to the Hospitals Model, only one real hospital is chosen just to show how the work can be done
    - `generate_fake_data()` function populates the database using 100 fake hospital names and 50 procedures along with fake charges details

- The urls.py file shows the API endpoint which is to be used by the android app. The endpoint

  ```
  api/hospitals/hospital_name=HOSPITAL_NAME&procedure_name=PROCEDURE_NAME/
  ```

  Sends back a JSON response, filtering from the database for `hospital_names` and `procedure_names` as provided while making the API call

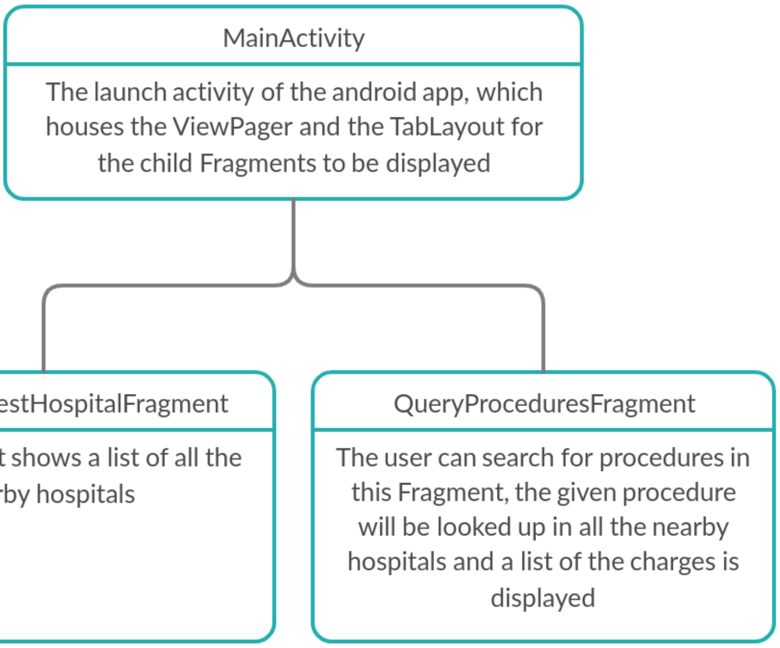
  A sample JSON response for `hospital_name="Penn"` and `procedure_name="heart"` is shown below

```
{
    "hospital_name": "Penn Presbyterian Medical Center",
    "procedure_name": "HC HEARTRAIL II GUIDE CATH",
    "estimated_cost": "362.00"
},
{
    "hospital_name": "Penn Presbyterian Medical Center",
    "procedure_name": "HC HEART IMAGE SPECT",
    "estimated_cost": "2244.00"
},
{
    "hospital_name": "Penn Presbyterian Medical Center",
    "procedure_name": "HC OAT OPEN HEART W/AUTOTRAN",
    "estimated_cost": "28647.00"
},
```

- The models.py file describes the model as will be stored in the database. For the mockup purpose, the following model is used

```
hospital_name = models.CharField(max_length=200)
procedure_name = models.CharField(max_length=200)
estimated_cost = models.DecimalField(decimal_places=2,
   max_digits=1000000)
```

- The overall architecture of the android app is shown below

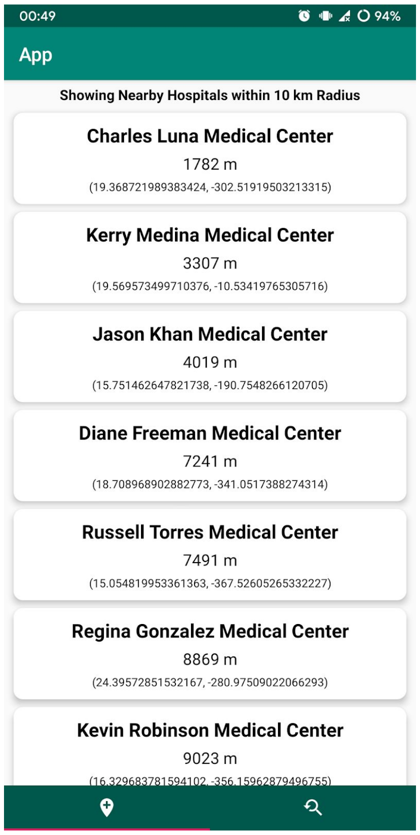

MainActivity
The launch activity of the android app, which houses the ViewPager and the TabLayout for the child Fragments to be displayed
DisplayNearestHospitalFragment
This Fragment shows a list of all the nearby hospitals
QueryProceduresFragment
The user can search for procedures in this Fragment, the given procedure will be looked up in all the nearby hospitals and a list of the charges is displayed

- The NearbyHospitals.java file demonstrates the work of location-based hospital searching. For the mockup purpose, `fetchNearbyHospitals(int n)` method randomly selects a few hospitals from the available list of all the hospitals and returns them as if they are the nearby located hospitals

- The file JsonPlaceHolderApi.java sets up the template for the REST API calls using Retrofit. The API call is made with two params, **hospital_name** and **procedure_name**, the **hospital_name** is selected based on the user's location and the **procedure_name** is queried by the user

```
@GET("hospitals/hospital_name={hospital_name}&procedure_name={procedure
_name}/")
Call<List<HospitalModel>> getHospitalsList(
       @Path("hospital_name") String hospitalName,
       @Path("procedure_name") String procedureName);
```

- The following screenshots demonstrate the working of the android app
    - DisplayNearestHospitalFragment
      This Fragment fetches the list of nearby hospitals, i.e for mock up purpose, randomly selected few hospitals are displayed, along with their distance (in meters) and the (latitude, longitude) information. The latitude and longitude information can be further used to show the direction to the user for navigation



Showing Nearby Hospitals within 10 km Radius

Charles Luna Medical Center
1782 m
(19.368721989383424, -302.51919503213315)

Kerry Medina Medical Center
3307 m
(19.569573499710376, -10.53419765305716)

Jason Khan Medical Center
4019 m
(15.751462647821738, -190.7548266120705)

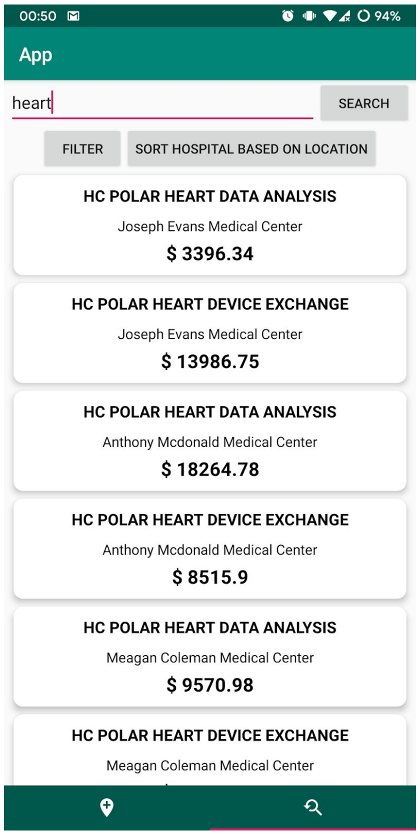
Diane Freeman Medical Center
7241 m
(18.708968902882773, -341.0517388274314)

Russell Torres Medical Center
7491 m
(15.054819953361363, -367.52605265332227)

Regina Gonzalez Medical Center
8869 m
(24.39572851532167, -280.97509022066293)

Kevin Robinson Medical Center
9023 m
(16.329683781594102, -356.15962879496755)

- QueryProceduresFragment
  In this fragment, the API call to the server is made for every nearby hospital including the procedure as queried by the user



HC POLAR HEART DATA ANALYSIS
Joseph Evans Medical Center
$ 3396.34

HC POLAR HEART DEVICE EXCHANGE
Joseph Evans Medical Center
$ 13986.75

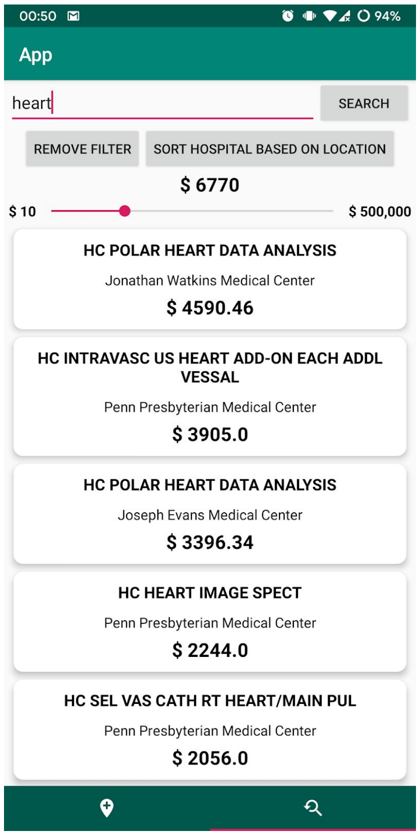
HC POLAR HEART DATA ANALYSIS
Anthony Mcdonald Medical Center
$ 18264.78

HC POLAR HEART DEVICE EXCHANGE
Anthony Mcdonald Medical Center
$ 8515.9

HC POLAR HEART DATA ANALYSIS
Meagan Coleman Medical Center
$ 9570.98

HC POLAR HEART DEVICE EXCHANGE
Meagan Coleman Medical Center

The filter options are there, either to filter hospitals based on location or on procedure charges. For the mockup purpose, the filter is shown based on the procedure charges



00:50 94%

App

heart SEARCH

REMOVE FILTER SORT HOSPITAL BASED ON LOCATION

$ 6770

$ 10 $ 500,000

HC POLAR HEART DATA ANALYSIS
Jonathan Watkins Medical Center
$ 4590.46

HC INTRAVASC US HEART ADD-ON EACH ADDL VESSAL
Penn Presbyterian Medical Center
$ 3905.0

HC POLAR HEART DATA ANALYSIS
Joseph Evans Medical Center
$ 3396.34

HC HEART IMAGE SPECT
Penn Presbyterian Medical Center
$ 2244.0

HC SEL VAS CATH RT HEART/MAIN PUL
Penn Presbyterian Medical Center
$ 2056.0

- The dependencies used in the mockup android app are listed below
    - Retrofit v2.7.2 A type-safe HTTP client for android, all the RESTFUL API calls made to the backend server are consumed using Retrofit Library. This library makes it relatively easy to retrieve and upload JSON via a REST-based web service
    - androidx.recycler-view:1.1.0 used to display a large number of items on the screen, with improved performance

The demonstration and working of the android app can be viewed here