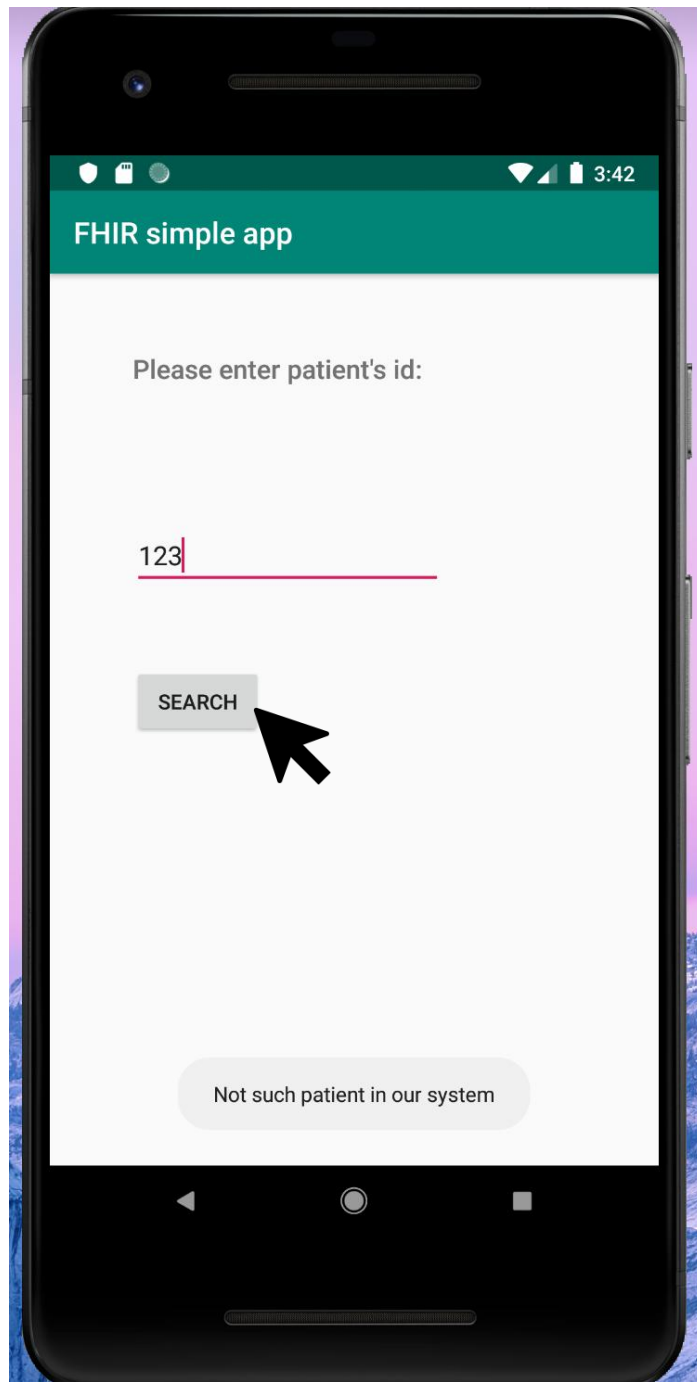


FIREBASE: store the patient/
practitioner who has logged in the
app; store the message sent
between patient & practitioner

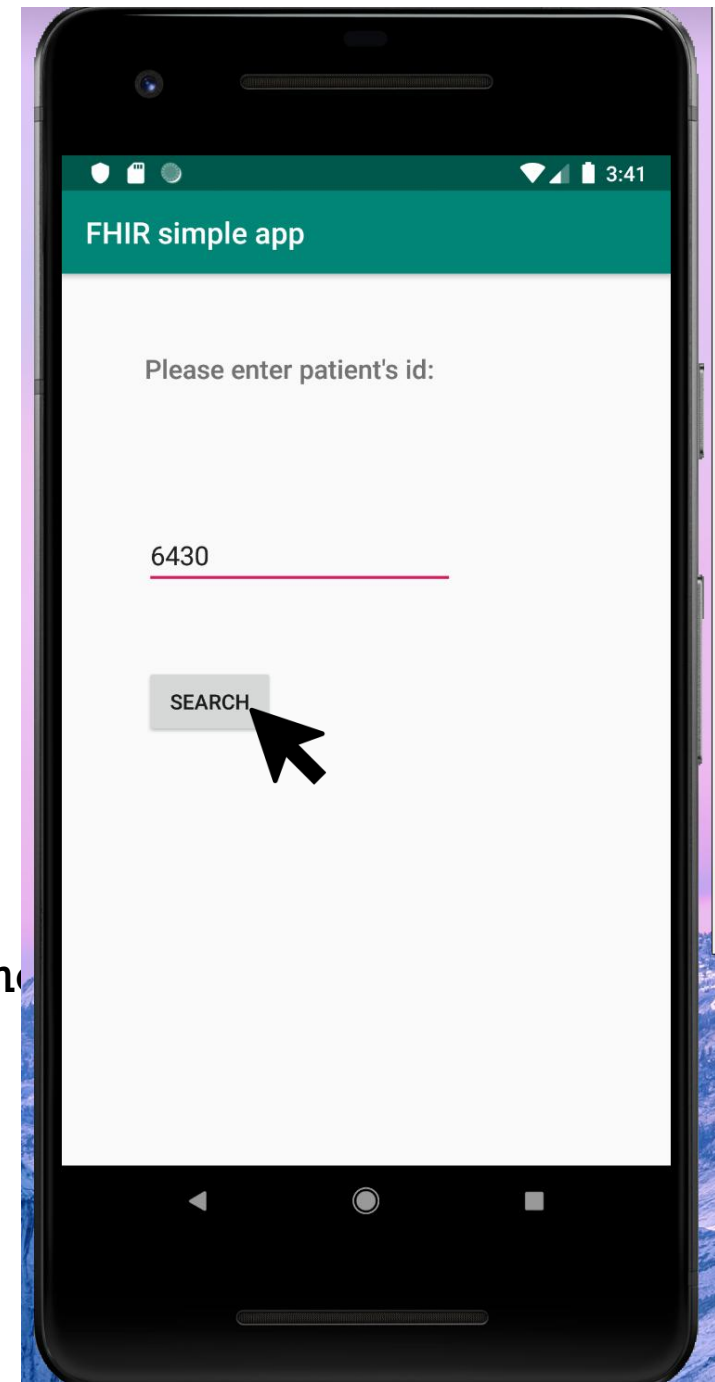


1. CURRENT APP



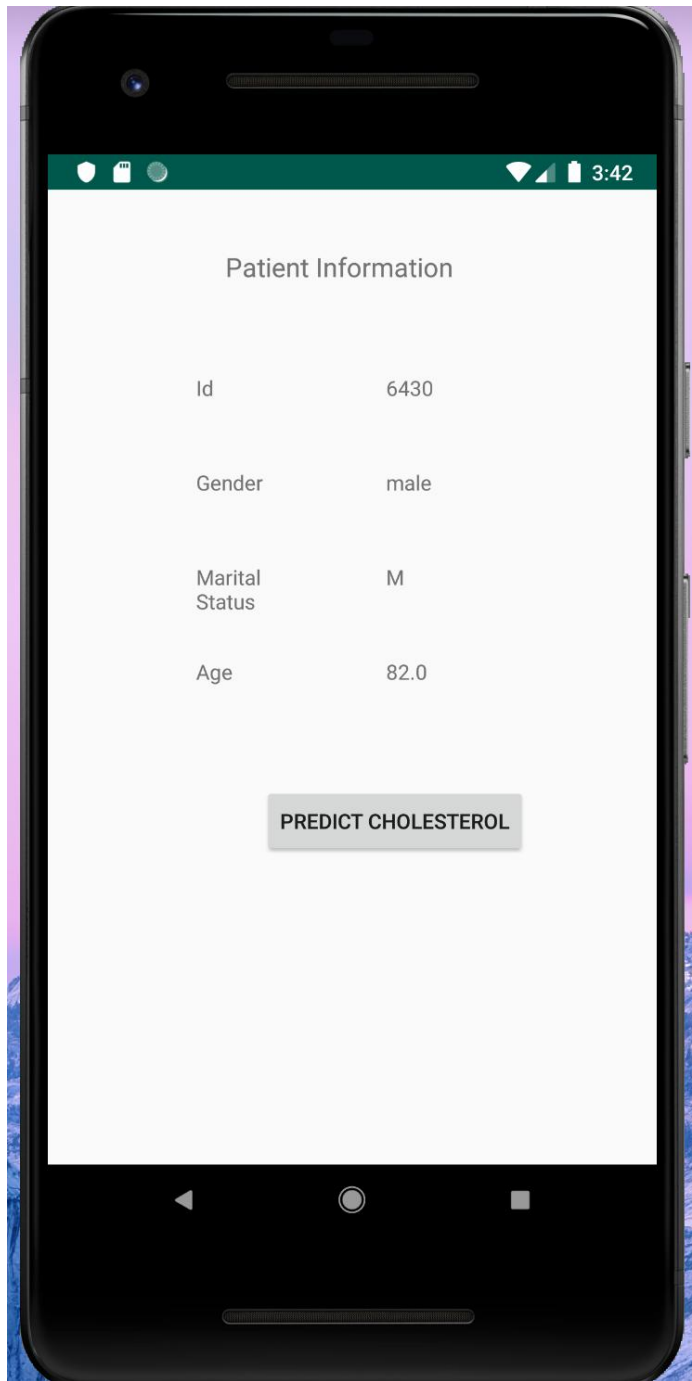


1st Activity
Patient's id not
found



1st Activity
Patient's id found

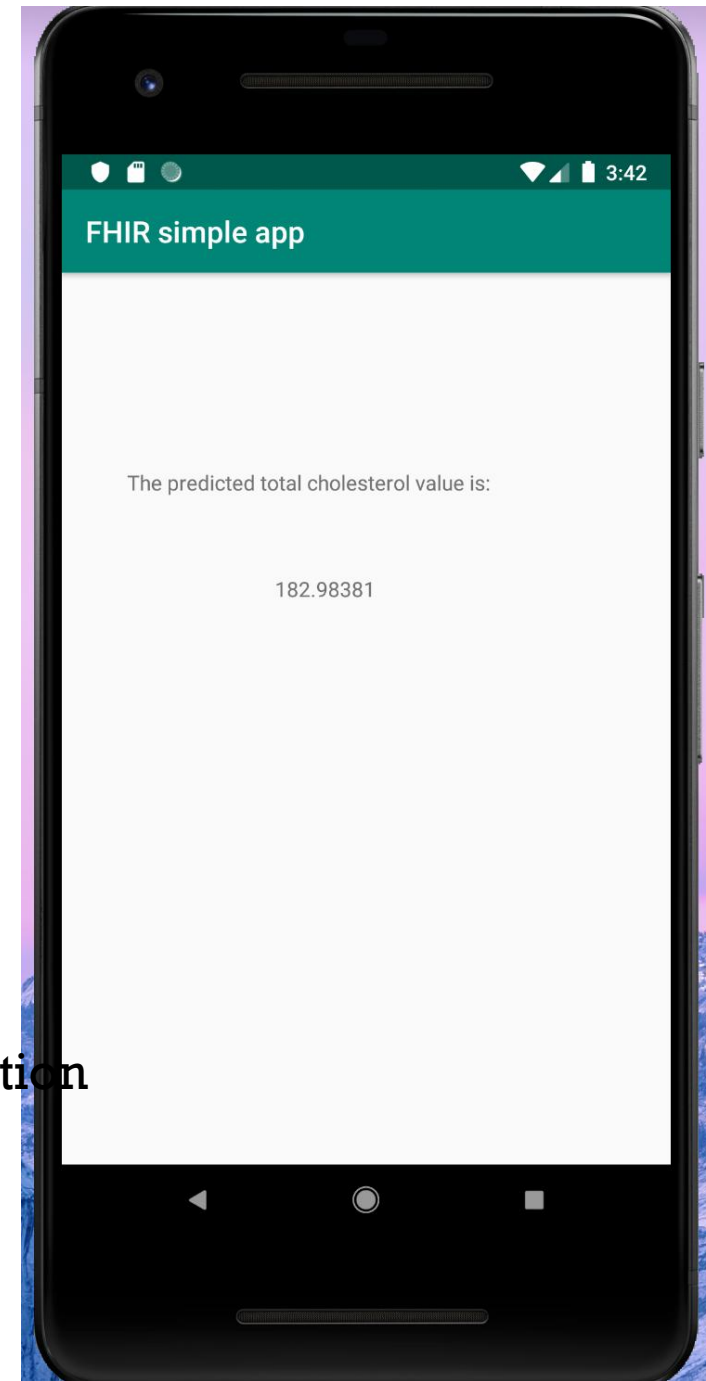




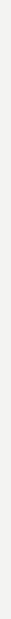
2nd Activity
Show
patient's
information



3rd Activity
Show prediction



2. TRAIN DATA WITH TENSORFLOW



COLLECT DATA & STORE IN PANDAS DATA FRAME

In [8]: fhirdata

Out[8]:

	patientid	gender	birthDate	marituaStatus	totalCholesterol	Triglycerides	lowDensity	highDensity	issued
95351	95351	male	1951-08-08	M	198.715371	149.923347	93.341805	75.388896	2016-11-02
105329	105329	female	1979-05-22	M	220.848174	190.973841	130.625559	52.027846	2016-05-10
95533	95533	male	1955-02-10	S	162.250776	173.231002	124.408832	33.037192	2019-03-18
106002	106002	female	1951-10-15	M	155.319703	151.206764	96.470399	72.357836	2019-02-05
106399	106399	female	1979-02-03	M	162.709395	131.943518	68.071220	68.249472	2018-12-01
...
86123	86123	female	1974-08-13	M	161.073698	147.065740	67.712838	63.947711	2015-10-27
86286	86286	male	1951-06-26	M	197.052226	105.095023	108.634595	67.398626	2016-06-14
104921	104921	male	1961-09-21	M	187.491146	143.170621	82.711031	76.145990	2012-04-05
105167	105167	female	1977-10-05	M	170.827770	135.898424	80.234724	63.413361	2017-09-13
117745	117745	female	1970-08-22	M	165.700146	142.332017	60.379737	76.854006	2016-09-10

306 rows × 9 columns



PRE-PROCESSING- GENDER/MARITAL/AGE

```
dataset3.head()
```

Out[21]:

	patientid	birthDate	totalCholesterol	issued	female	male	Married	Single	age
0	95351	1951-08-08	198.715371	2016-11-02	0	1	1	0	65.0
1	105329	1979-05-22	220.848174	2016-05-10	1	0	1	0	37.0
2	95533	1955-02-10	162.250776	2019-03-18	0	1	0	1	64.0
3	106002	1951-10-15	155.319703	2019-02-05	1	0	1	0	68.0
4	106399	1979-02-03	162.709395	2018-12-01	1	0	1	0	39.0



SPLIT DATA – TRAINING/TEST DATA

```
In [29]: train_dataset2 = train_dataset.drop(columns = ['patientid', 'issued', 'birthDate'])  
test_dataset2 = test_dataset.drop(columns = ['patientid', 'issued', 'birthDate'])  
train_dataset2
```

Out[29]:

	female	male	Married	Single	age
274	1	0	1	0	68.0
66	0	1	1	0	59.0
258	0	1	1	0	71.0
210	1	0	1	0	52.0
159	0	1	1	0	72.0
...
279	1	0	0	1	42.0
142	0	1	1	0	64.0
262	0	1	1	0	60.0
91	1	0	1	0	50.0
82	1	0	1	0	69.0

245 rows × 5 columns



BUILD MODEL

```
In [30]: def build_model():  
    model = keras.Sequential([  
        layers.Dense(64, activation='relu', input_shape=[len(train_dataset2.keys())]),  
        layers.Dense(64, activation='relu'),  
        layers.Dense(1)  
    ])  
  
    optimizer = tf.keras.optimizers.RMSprop(0.001)  
  
    model.compile(loss='mse',  
                  optimizer=optimizer,  
                  metrics=['mae', 'mse'])  
  
    return model
```



EXPORT MODEL AS TENSOR FLOW LITE FILE

```
new_model= tf.keras.models.load_model(filepath="/Users/shunyang/FHIR_coding/simple_model3.h5")
tflite_converter = tf.lite.TFLiteConverter.from_keras_model(new_model)
tflite_model = tflite_converter.convert()
open("tf_lite_model.tflite", "wb").write(tflite_model)
```



IMPORT AND RUN MODEL IN ANDROID STUDIO

```
tflite = new Interpreter(loadModelFile());
float[][] inputVal = new float[1][5];

inputVal[0][0] = female;
inputVal[0][1] = male;
inputVal[0][2] = single;
inputVal[0][3] = married;
inputVal[0][4] = (float) age;

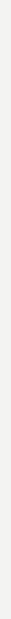
float[][] outVal = new float[1][1];
tflite.run(inputVal, outVal);

String cholesterolString = Float.toString(outVal[0][0]);
```

```
103 private MappedByteBuffer loadModelFile() throws IOException {
104     AssetFileDescriptor fileDescriptor = this.getAssets().openFd( fileName: "no_normalized_model.tflite");
105     FileInputStream inputStream = new FileInputStream(fileDescriptor.getFileDescriptor());
106     FileChannel fileChannel = inputStream.getChannel();
107     long startOffset = fileDescriptor.getStartOffset();
108     long declaredLength = fileDescriptor.getDeclaredLength();
109     return fileChannel.map(FileChannel.MapMode.READ_ONLY, startOffset, declaredLength);
110 }
```



3. FETCH JSON DATA FROM FHIR SERVER



QUERY DATA IN PYTHON

```
In [7]: next_page = True
next_url = base_url + 'DiagnosticReport'
count_pages = 0
count_patient_with_cholesterol = 0

while next_page == True:
    diagnosticReport_data = requests.get(url=next_url).json()
    next_page = False
    links = diagnosticReport_data['link']
    for i in range(len(links)):
        link = links[i]
        if link['relation'] == 'next':
            next_page = True
            next_url = link['url']
            print(next_url)
            count_pages +=1
    print(count_pages)

reports = diagnosticReport_data['entry']

for i in range(len(reports)):
    patient_list=[]
    results = reports[i]['resource']['result']
    cholesterol = False
    for result in results:
        if result['display'] == 'Total Cholesterol':
            cholesterol = True
    if cholesterol == True:
        count_patient_with_cholesterol +=1
    # print(patient_list)
```



QUERY DATA IN ANDROID STUDIO

```
48 protected PatientSample doInBackground(String... params){
49     PatientSample patient = null;
50     try{
51         String selectedPatient = params[0];
52         URL webServiceEndPoint = new URL( spec: "http://hapi-fhir.erc.monash.edu:8080/baseDstu3/Patient/" + selectedPatient);
53
54         //create connection
55         HttpURLConnection myConnection = (HttpURLConnection) webServiceEndPoint.openConnection();
56         if(myConnection.getResponseCode() == 200){
57
58             patient = new PatientSample();
59             patient.setPatientid(Integer.parseInt(selectedPatient));
60
61             //open a stream to it and get a reader
62             InputStream responseBody = myConnection.getInputStream();
63             InputStreamReader responseBodyReader = new InputStreamReader(responseBody, charsetName: "UTF-8");
64
65             //now use a JSON parser to decode data
66             JsonReader jsonReader = new JsonReader(responseBodyReader);
67             jsonReader.beginObject();//consume array's opening JSON brace
68             String keyName;
69             while(jsonReader.hasNext()){//process key/value pairs inside the current object
70                 keyName = jsonReader.nextName();
71                 if(keyName.equals("gender")){
72                     patient.setGender(jsonReader.nextString());
73                 }else if(keyName.equals("birthDate")){
74                     String birthDate = jsonReader.nextString();//1939-08-08
75                     Date today = new Date();
76                     int age = today.getYear()+1900 - Integer.parseInt(birthDate.substring(0,4));
77                     patient.setAge((double) age);
78                 }else if(keyName.equals("maritalStatus")){
79                     jsonReader.beginObject();
80                     while(jsonReader.hasNext()){
81                         keyName = jsonReader.nextName();
82                         if(keyName.equals("text")){
83                             patient.setMaritalStatus(jsonReader.nextString());
84                         }else{
85                             jsonReader.skipValue();
86                         }
87                     }
88                 }
89             }
90         }
91     } catch (Exception e) {
92         // Handle exception
93     }
94 }
```

