

**DROID NURSE v1.0**

**AN ANDROID APP**

**WITH MONGODB**

**A PROJECT REPORT**

*Submitted by*

**VENKATESH BALASUBRAMANIAN  
SANDEEP RAMAMOORTHY**

**COLLEGE OF COMPUTER AND INFORMATION SCIENCE**

**NORTHEASTERN UNIVERSITY, BOSTON**

**AUGUST 2014**

## **ABSTRACT:**

The main objective of this project is to create an interactive mobile application (Android) that collects user's prescriptions and health metrics to provide him/her with an overview of the medications consumed, status of the prescriptions, online prescription refill, track expenses, search for physician details and drugs thereby taking a step ahead in simplifying daily tasks.

## **TOOLS USED:**



PLATFORM

IDE            ANDROID STUDIO (BETA 0.82)

Language    Java

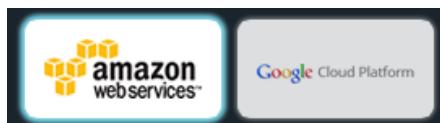


STORAGE



HOST

(SINGLE NODE DEVELOPMENT)



CLOUD PROVIDER

(AMAZON WEB SERVICES /  
GOOGLE CLOUD PLATFORM)

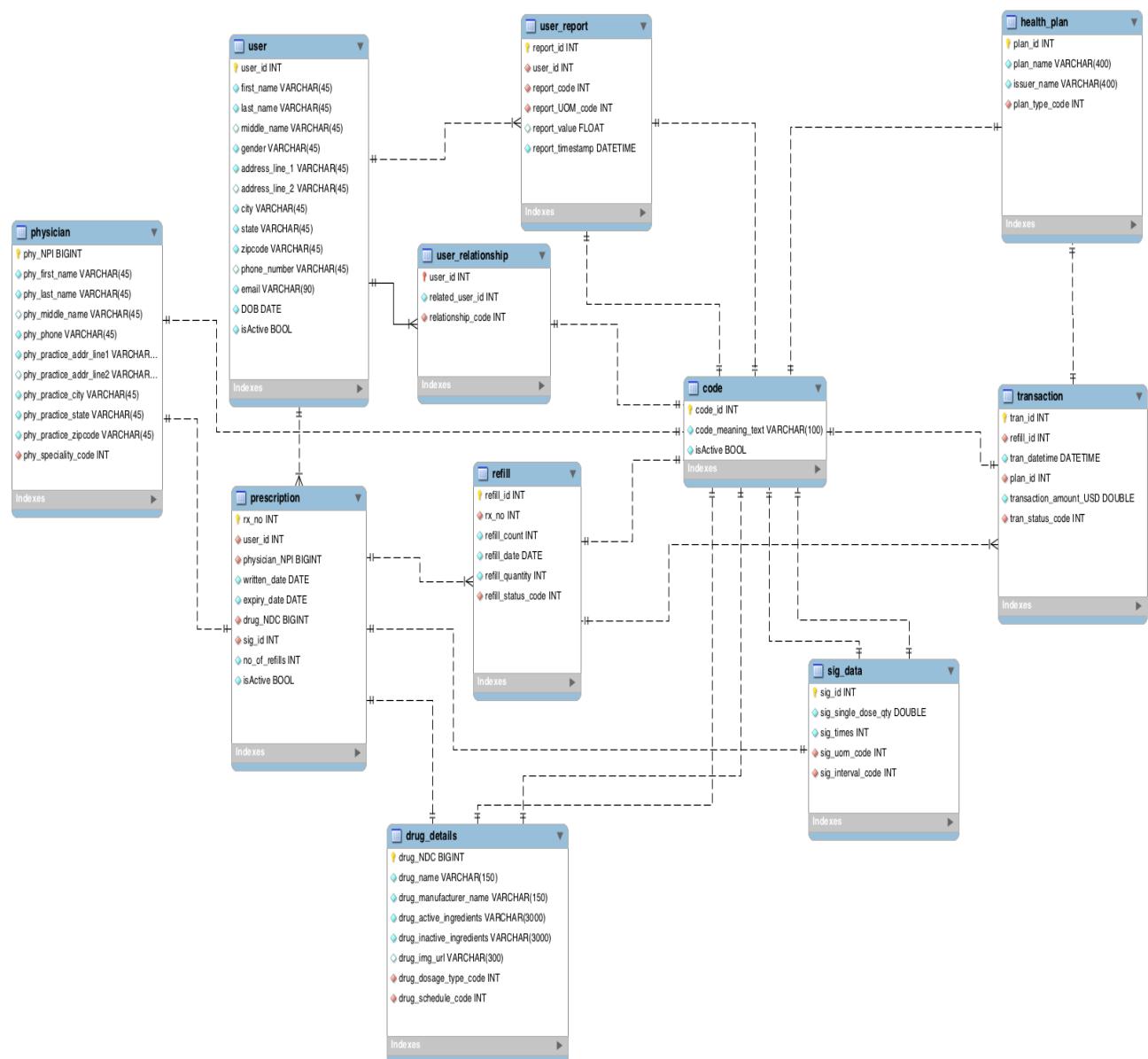
## SOURCE CODE LOCATION

**[HTTPS://GITHUB.COM/VENKATESH891/CS5200](https://github.com/Venkatesh891/cs5200)**

## **SYSTEM FEATURES:**

- Ability to refill medications with Walgreens API
- Generate expense report on the total health expenses of the user
- Notification alerts for the medication that needs to be taken by the user
- Facilitates the end user with the visual image of the pills to avoid confusion when there is more than one medication to be taken
- Facilitates the end user to input and track various metrics like blood pressure, blood glucose level, weight etc.
- Provides various information about doctors, physicians and health insurance plans based on the user queries
- Provides the end user to locate near by hospitals and pharmacies in real time based on his geo location

## DATABASE DIAGRAM:



## MONGODB COLLECTIONS:

A grouping of MongoDB documents. A collection is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection have a similar or related purpose. Below are the list of collections which has been designed to support our application needs:

The screenshot shows the MongoDB Compass interface with the 'Collections' tab selected. The top navigation bar includes tabs for 'Collections', 'Users', 'Stats', 'Backups', and 'Tools'. A button for 'Add collection' is located in the top right corner of the main area.

**Collections**

NAME	DOCUMENTS	CAPPED?	SIZE	
code	183	false	19.77 KB	
drug_details	32,470	false	4.87 MB	
health_plan	77,318	false	15.94 MB	
physician	878,798	false	97.41 MB	
prescription	4	false	9.11 KB	
refill	3	false	8.88 KB	
user_profile	50,002	false	17.54 MB	
user_report	14	false	9.95 KB	

---

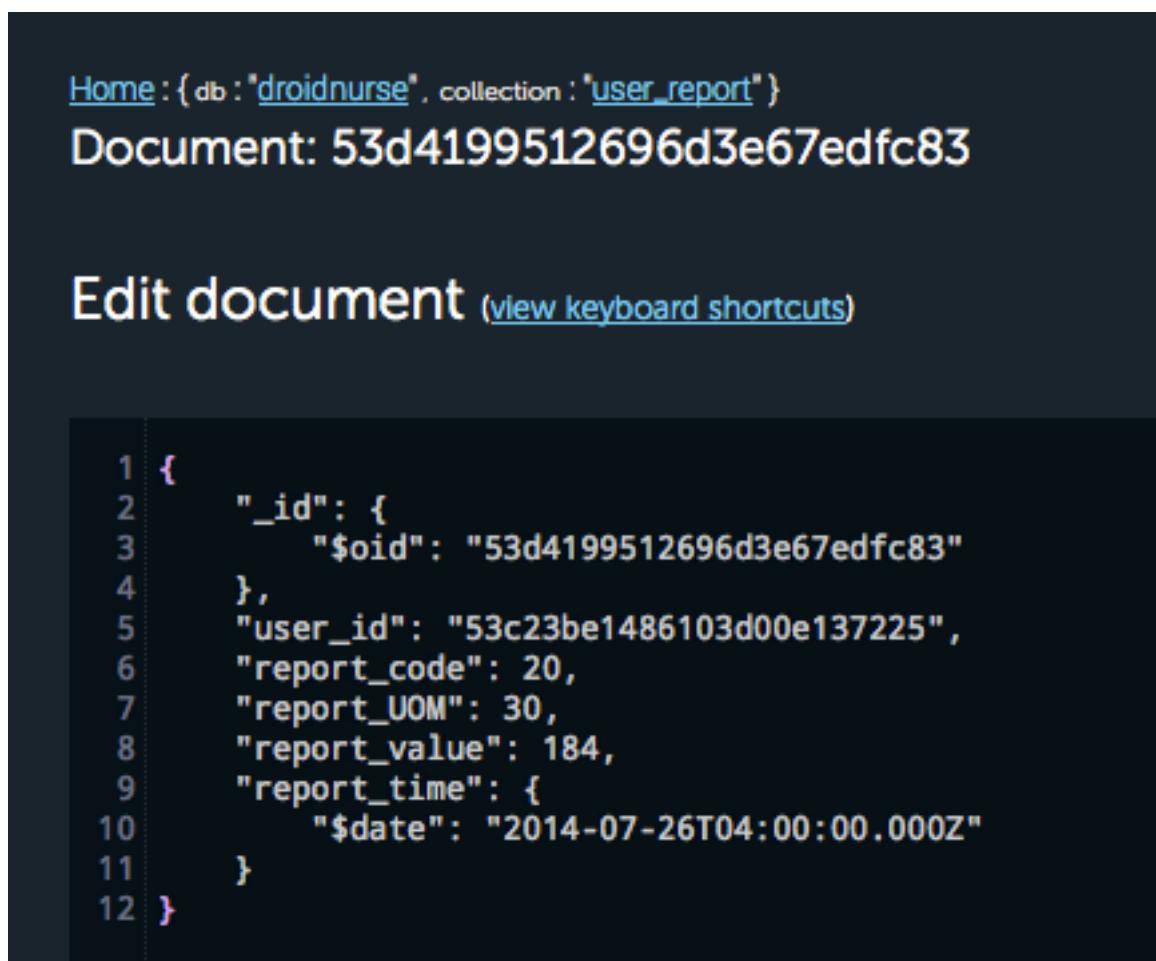
**System Collections**

NAME	DOCUMENTS	SIZE
system.indexes	12	1.00 KB
system.users	3	16.26 KB

## MONGODB DOCUMENTS:

A grouping of MongoDB documents. A collection is the equivalent of an RDBMS table. A record in a MongoDB collection and the basic unit of data in MongoDB. Documents are analogous to JSON objects but exist in the database in a more type-rich format known as BSON. The following screenshots would display the contents of various document:

### USER\_REPORT DOCUMENT



The screenshot shows a MongoDB document editor interface. At the top, it displays the URL: `Home : { db : "droidnurse", collection : "user_report" }` and the document ID: `Document: 53d4199512696d3e67edfc83`. Below this, there is a large button labeled `Edit document` with a link to `(view keyboard shortcuts)`. The main area contains the BSON representation of the document, with line numbers 1 through 12 on the left:

```
1 {
2   "_id": {
3     "$oid": "53d4199512696d3e67edfc83"
4   },
5   "user_id": "53c23be1486103d00e137225",
6   "report_code": 20,
7   "report_UOM": 30,
8   "report_value": 184,
9   "report_time": {
10     "$date": "2014-07-26T04:00:00.000Z"
11   }
12 }
```

## USER\_PROFILE DOCUMENT

```
1 {
2     "_id": {
3         "$oid": "53c23be1486103d00e137225"
4     },
5     "DOB": {
6         "$date": "1939-01-30T05:00:00.000Z"
7     },
8     "addr_line_1": "1765 Sherwood Circle",
9     "city": "Lafayette",
10    "email": "MarionLLittle@teleworm.us",
11    "first_name": "Marion",
12    "gender": "female",
13    "health_plans": [
14        {
15            "plan_object_id": "53c49473e49ef1ed10f27cc5",
16            "plan_type": 80
17        },
18        {
19            "plan_object_id": "53c49473e49ef1ed10f289c4",
20            "plan_type": 80
21        }
22    ],
23    "isActive": true,
24    "last_name": "Little",
25    "middle_name": "L",
26    "password": "a",
27    "phone": "337-207-8400",
28    "state": "LA",
29    "username": "a",
30    "zip": 70506
31 }
```

## CODE DOCUMENT

[Home](#) : { db : "droidnurse", collection : "code" }

Document: 1.0

[Edit document](#) ([view keyboard shortcuts](#))

```
1 {
2     "_id": 1,
3     "meaning_text": "Son",
4     "isActive": true
5 }
```

## DRUG\_DETAILS DOCUMENT

[Home](#) : { db : "droidnurse", collection : "drug\_details" }

Document: 53d935c50c69242187fdef0a

[Edit document](#) ([view keyboard shortcuts](#))

```
1 {
2     "_id": {
3         "$oid": "53d935c50c69242187fdef0a"
4     },
5     "NDC": "67296-0642",
6     "drug_name": "APAP 750 MG / hydrocodone bitartrate 7.5 MG Oral Tablet",
7     "hasImage": 0
8 }
```

## **HEALTH\_PLAN DOCUMENT**

[Home](#) : { db : "droidnurse", collection : "health\_plan" }  
Document: 53c49473e49ef1ed10f27cc5

**Edit document** ([view keyboard shortcuts](#))

```
1 {
2     "_id": {
3         "$oid": "53c49473e49ef1ed10f27cc5"
4     },
5     "Plan_id": "38344AK0570001",
6     "plan_name": "Blue Cross Blue Shield Plus 5250 HSA High, a Multi-State Plan",
7     "issuer_name": "Premera Blue Cross Blue Shield of Alaska",
8     "city": "ALEUTIANS EAST",
9     "state": "AK"
10 }
```

## **PYSICIAN DOCUMENT**

[Home](#) : { db : "droidnurse", collection : "physician" }  
Document: 53c49ec9e49ef1ed10f3aaca

**Edit document** ([view keyboard shortcuts](#))

```
1 {
2     "_id": {
3         "$oid": "53c49ec9e49ef1ed10f3aaca"
4     },
5     "npi": 1003000134,
6     "first_name": "THOMAS",
7     "last_name": "CIBULL"
8 }
```

## REFILL DOCUMENT

[Home](#) : { db : "droidnurse", collection : "refill" }

Document: 53ea37efd39a644b2a3afb88

Edit document [\(view keyboard shortcuts\)](#)

```
1  {
2      "_id": {
3          "$oid": "53ea37efd39a644b2a3afb88"
4      },
5      "refill_count": 1,
6      "refill_date": {
7          "$date": "2014-08-12T04:00:00.000Z"
8      },
9      "refill_quantity": 2,
10     "qty_remaining": 0,
11     "total_cost": 120,
12     "isActive": true,
13     "insurance_details": {
14         "plan_object_id": "53c49473e49ef1ed10f27cc5",
15         "plan_cover": 115,
16         "plan_copay": 5
17     },
18     "notification": {
19         "last_update_time": {
20             "$date": "2014-08-13T16:32:44.790Z"
21         },
22         "note_counter": 0
23     }
24 }
```

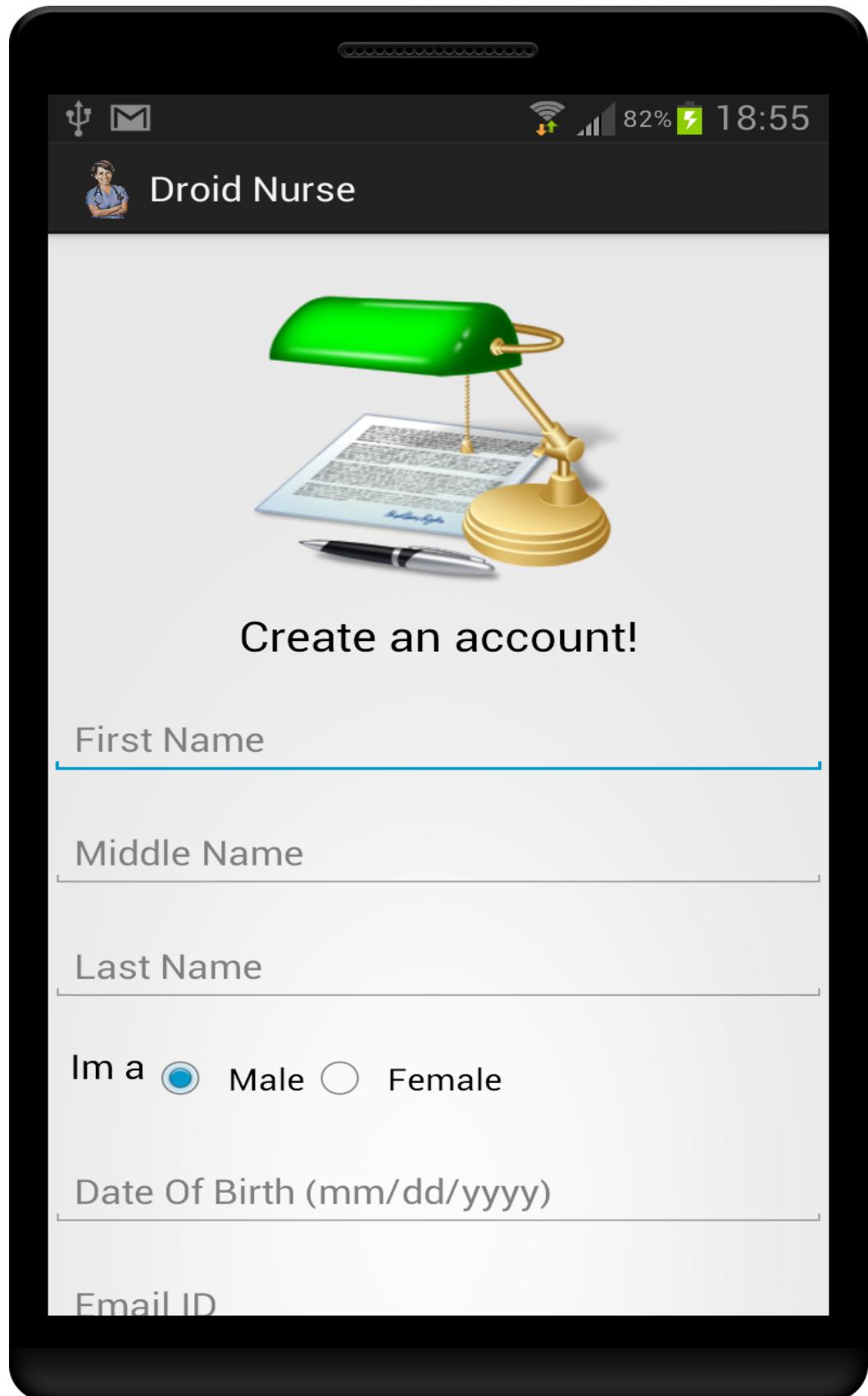
## **API CONSUMED:**

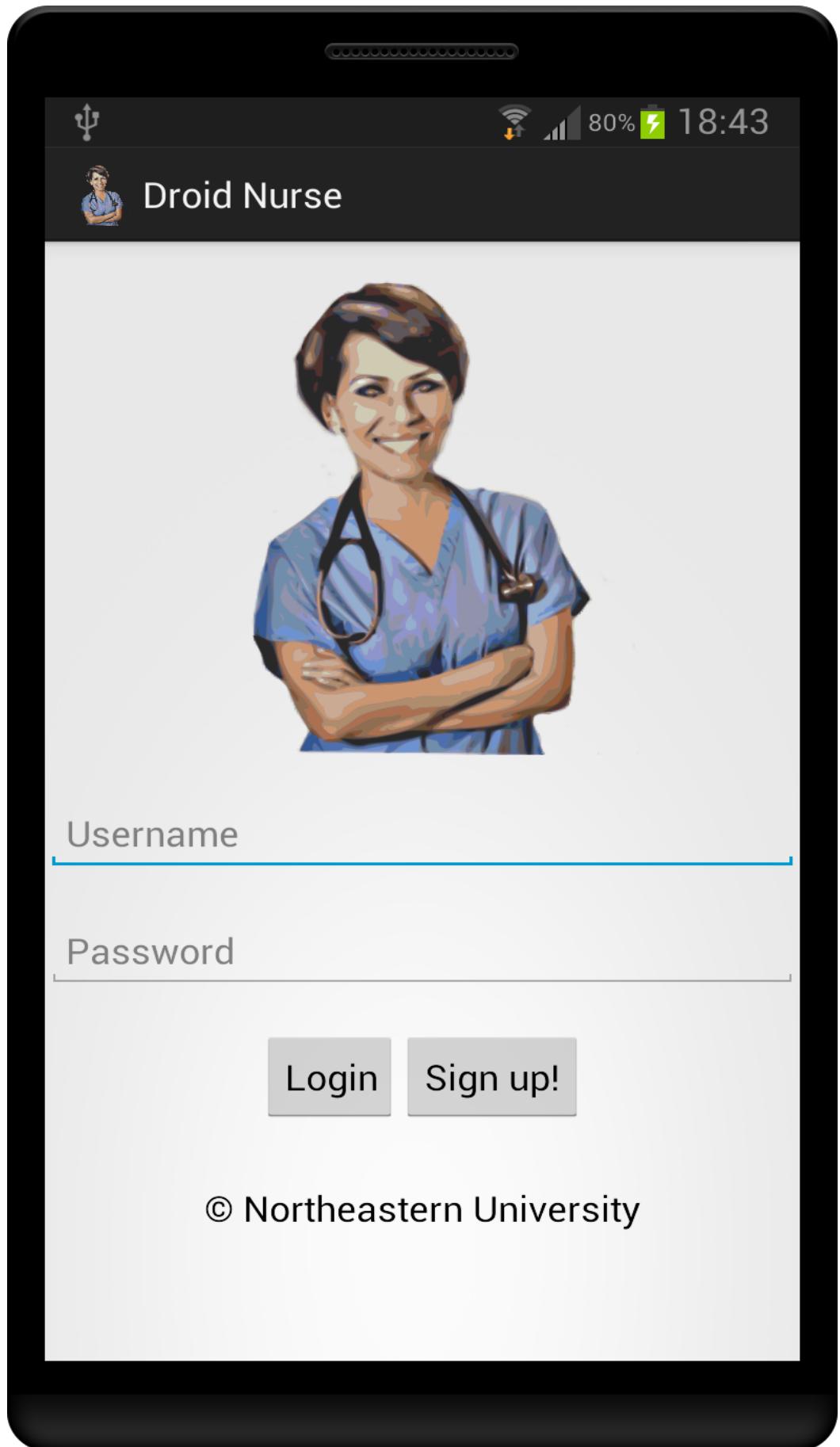
Apart from the data which is present in our MongoDB, droid nurse application consumes various REST API for detailed information. Below are the various API which was serving as back bone for running our application:

API	USAGE	REFERENCE
Walgreens Prescription API	Allows the user to refill prescription online at Walgreen pharmacy	<a href="#"><u>Walgreens API Documentation</u></a>
PillBox API	Provide details about various drugs along with images	<a href="#"><u>PillBox API Documentation</u></a>
Socrata Physician API	Provide details about registered physician within United States	<a href="#"><u>Socrata Physician API</u></a>
Socrata Health Plan API	Provide details about health insurance plans for the states that are participating in the Federal Health Insurance Market Place	<a href="#"><u>Socrata Health Plan API</u></a>
Google Maps API	3D rendering of Google Maps with customized markers and location service	<a href="#"><u>Google Maps API Android</u></a>
Google Places API	Provides information about places with geographic locations	<a href="#"><u>Google Places API</u></a>

## **SIGNUP & LOGIN FEATURE:**

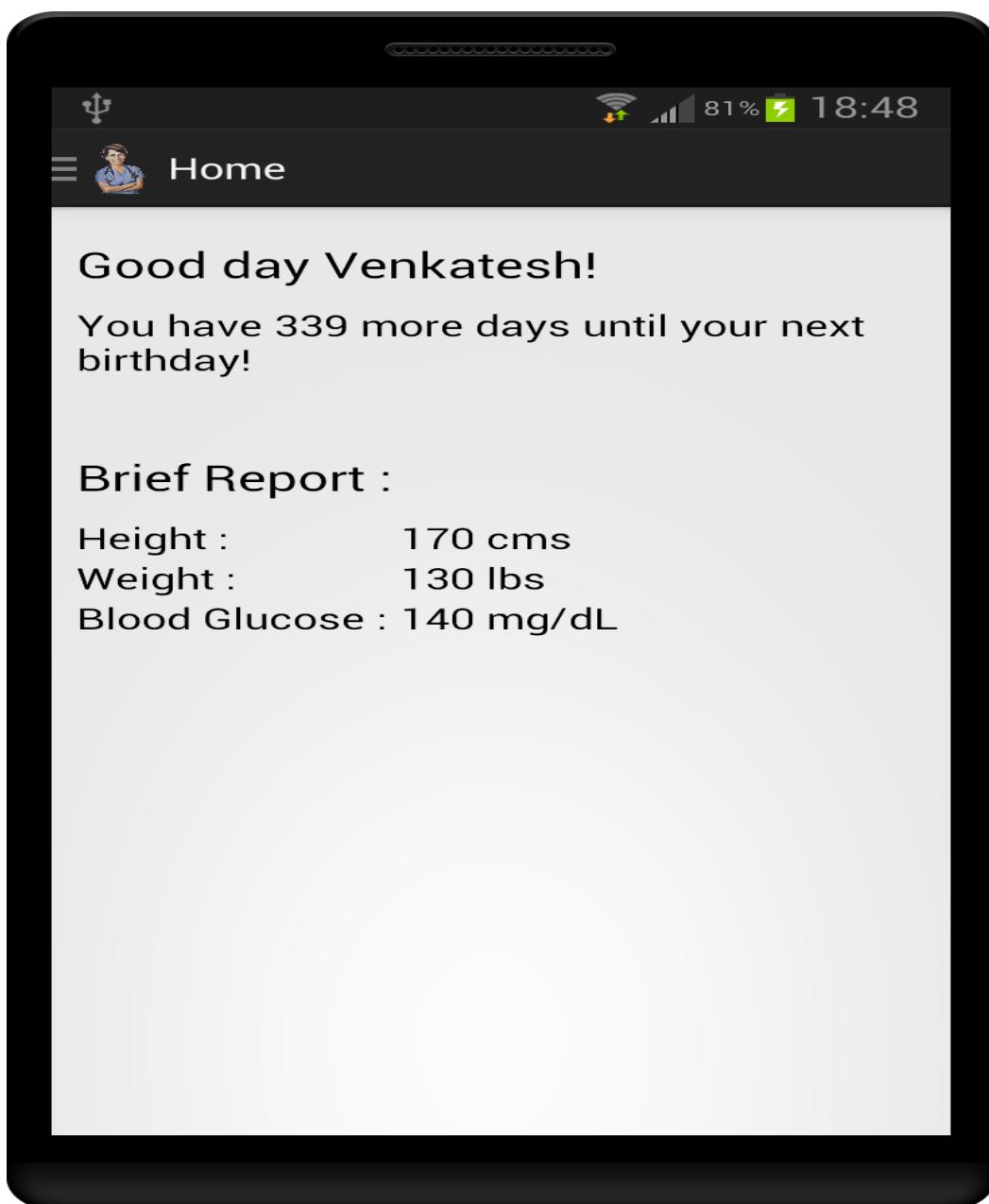
After successful installation of the android package (.apk) file on any android smart phone, the end user has to be registered user to access the features of the application. Simple sign up process makes it to be a registered user for our application. Below are the screenshot of sign up and login screen:

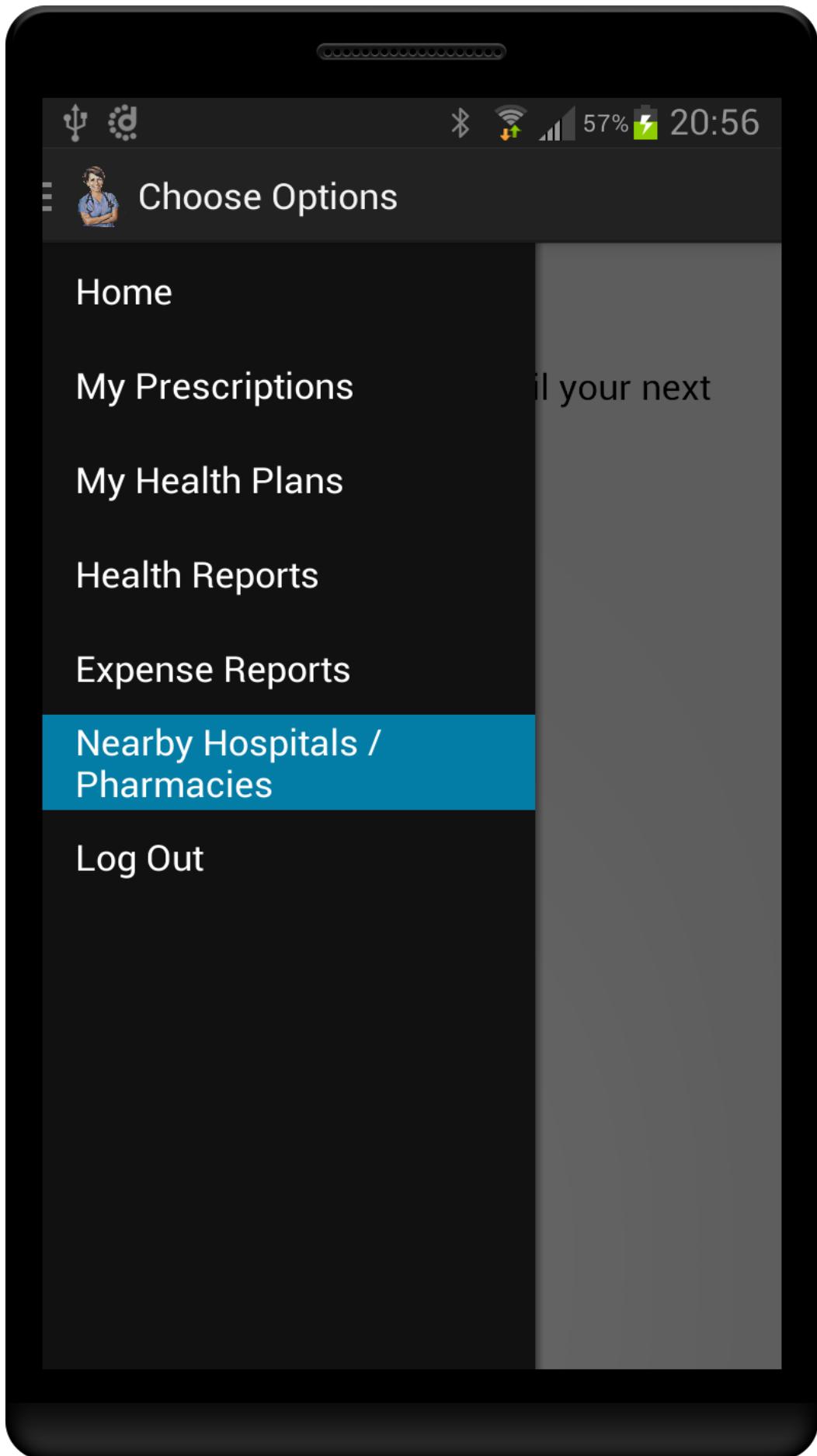




## **HOME SCREEN & MENU OPTION:**

After successful login into our application, the home page would contain a welcome message with a brief health report along with the number of days until your next birthday. The home screen is an android activity where in all the remaining user option runs as android fragments. Upon clicking on the top left option in the screen displays the various menu which are the system features.





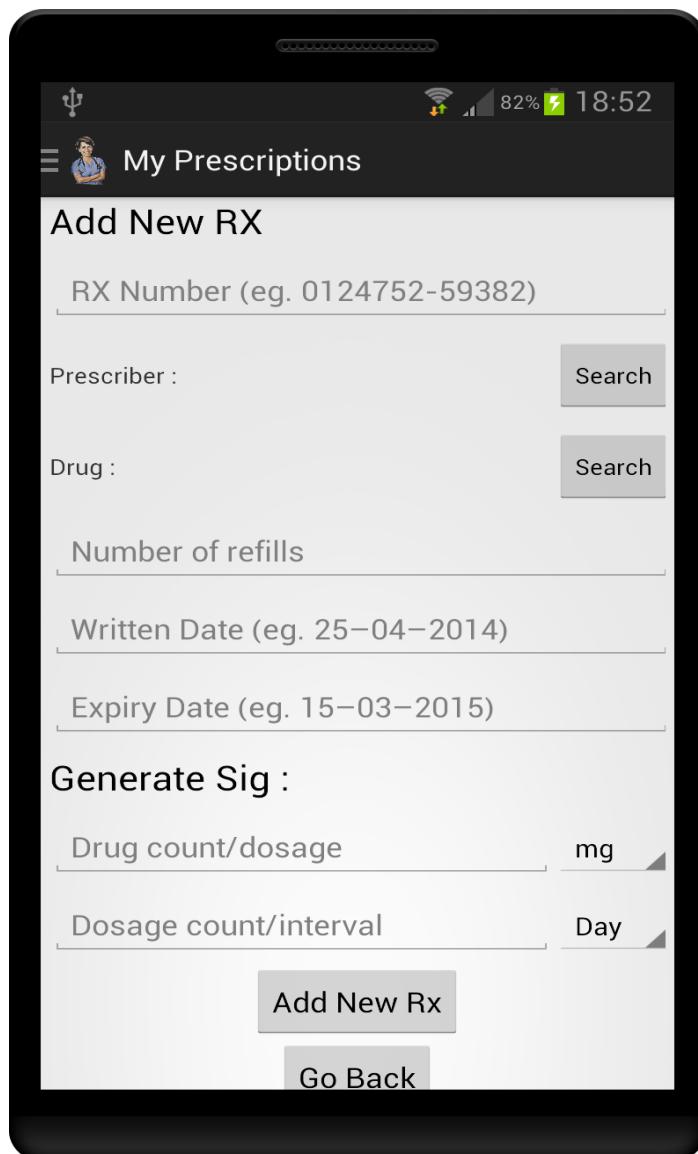
## MY PRESCRIPTION:

Prescriptions, shortly called as Rx which indicates “to take”. A prescription is a health care programme that governs the plan of care for an individual patient and is implemented by a qualified practitioner. The below screenshot displays the list of Rx which has been added earlier by the user and also he/she has the ability to add new Rx information.



## 1) Add New Rx:

Add new Rx screen enables the user to feed in the medication details, which was prescribed by the physician. The Add New Rx screen captures the important information, which are: Rx Number, prescriber name, drug name, the number of refills that could be done with this prescription, prescription written date, prescription expiry date and the sig details. The sig details are required to provide notifications on hourly basis.

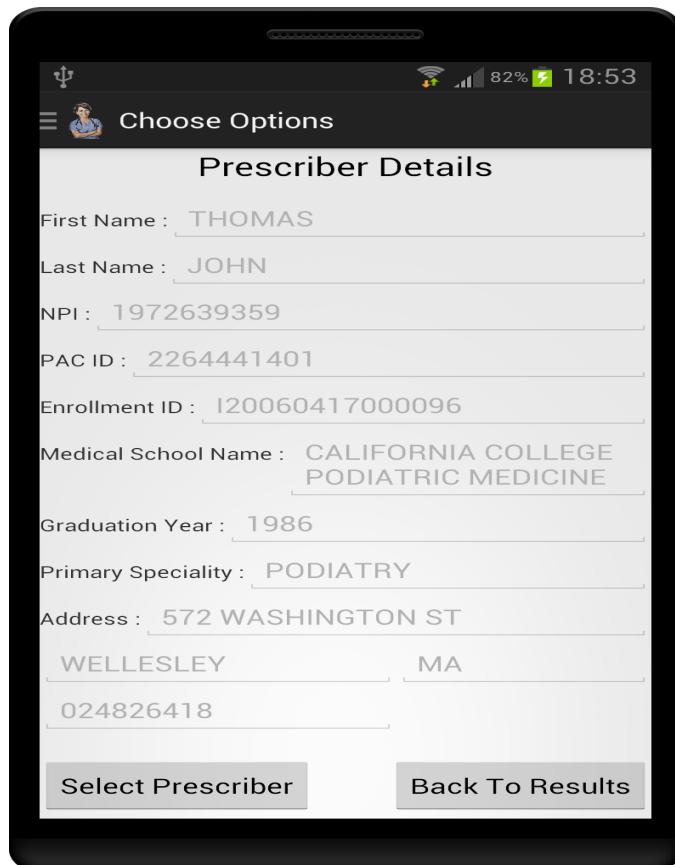
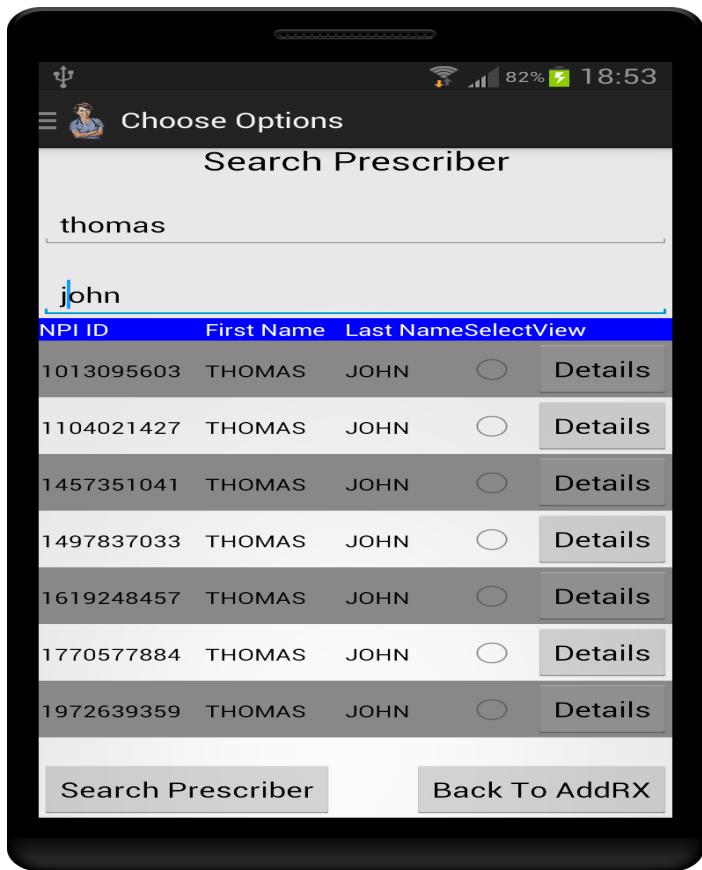


After providing all the required details, hitting “Add New Rx” button would capture these details and store them at MongoDB “prescription” collection and appends Rx Number to the array list present within the “user\_profile” collection.

**1.1) Search Prescriber:** The search prescriber screen allows the user to look for physician who has prescribed the particular Rx. The search is based on first / last name of the physician, which is then queried from 8,78,798 documents located in the “physician” collection of Mongo DB. The physician collection of the mongo db holds limited details about the physician, if more details are requested for, and then we make an HttpGet request from the REST API provided by Socrata and those additional details are displayed after parsing the response json data.

Below mentioned is the URL, which is framed to obtain details about a particular physician. Every physician would have a unique NPI number with which we would be able to get his/her details.

**<http://data.medicare.gov/resource/s63f-csi6.json?where=npi='<NPI NUMBER>'>**



1.2) **Search Drug:** The search drug screen allows the user to look for drug details, which the physician has prescribed. The search is based on name of the drug, regular expression are consumed for providing efficient search results. The drug\_details collection of the mongo db holds limited details about the drug, if more details are requested then we make an HttpGet request from the REST API provided by PillBox and those additional details are displayed after parsing the response json data. Below mentioned is the URL, which is framed to obtain details about a particular drug. Every drug would have a unique NDC number with which we would be able to get the details along with the drug image too.

**<http://pillbox.nlm.nih.gov/PHP/pillboxAPIService.php?prodcode=<NDC NUMBER>&key=<API KEY>>**

Choose Options

Search Drug

gem

Search Drug Back To AddRX

NDC	Select	View
53808-0368	<input type="radio"/> Gemfibrozil 600 MG Oral Tablet	Details
0904-5379	<input type="radio"/> Gemfibrozil 600 MG Oral Tablet	Details
100-10-000	<input type="radio"/> Gemfibrozil 600 MG Oral Tablet	Details

Choose Options

SELECTED (TYPE 1)  
HYPMELLOSES /  
MAGNESIUM STEARATE  
/ METHYLCELLULOSE  
(15 CPS) /  
POLYETHYLENE  
GLYCOLS / TITANIUM  
DIOXIDE

Has Image : Image Preview Available For This Drug !!

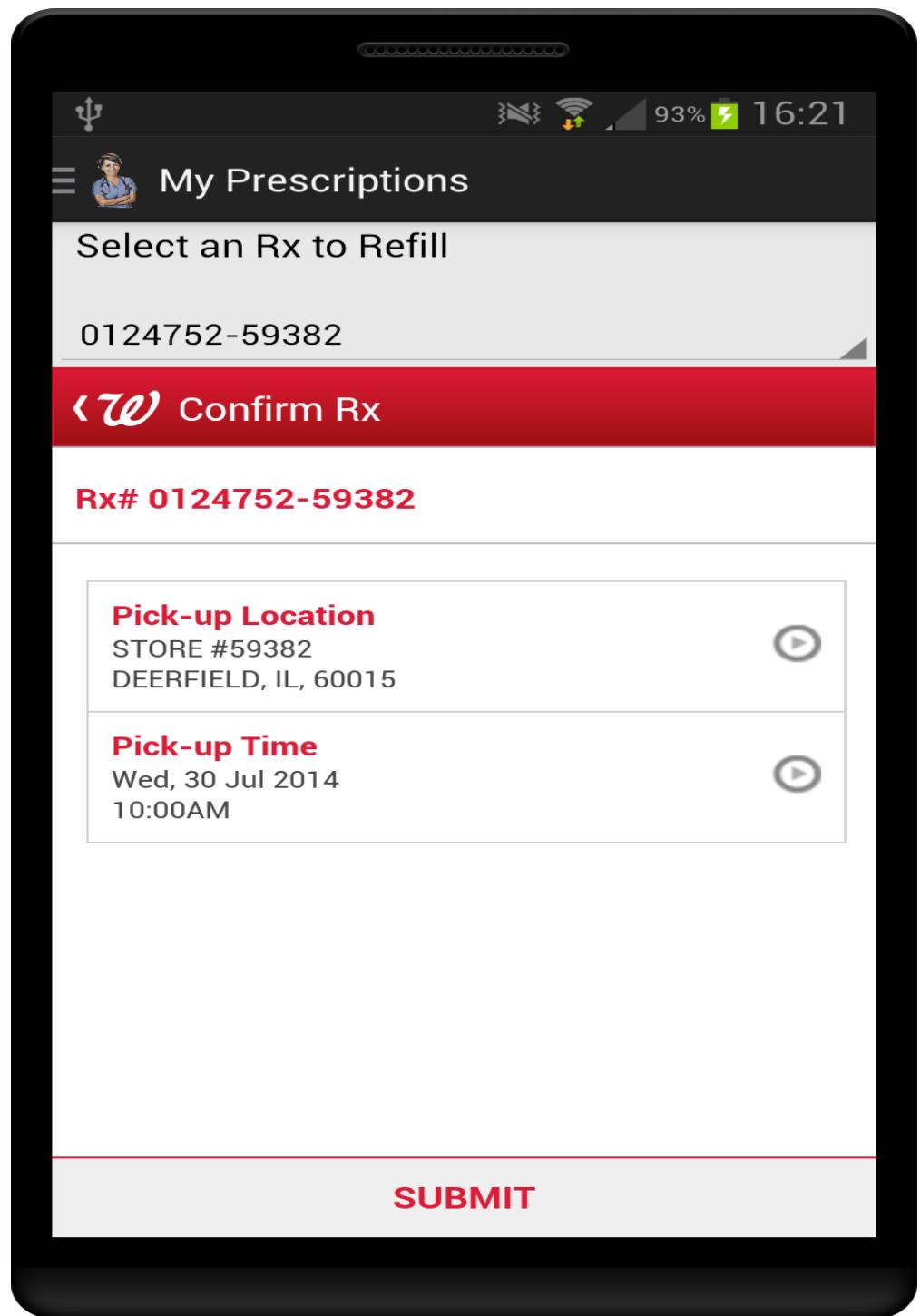


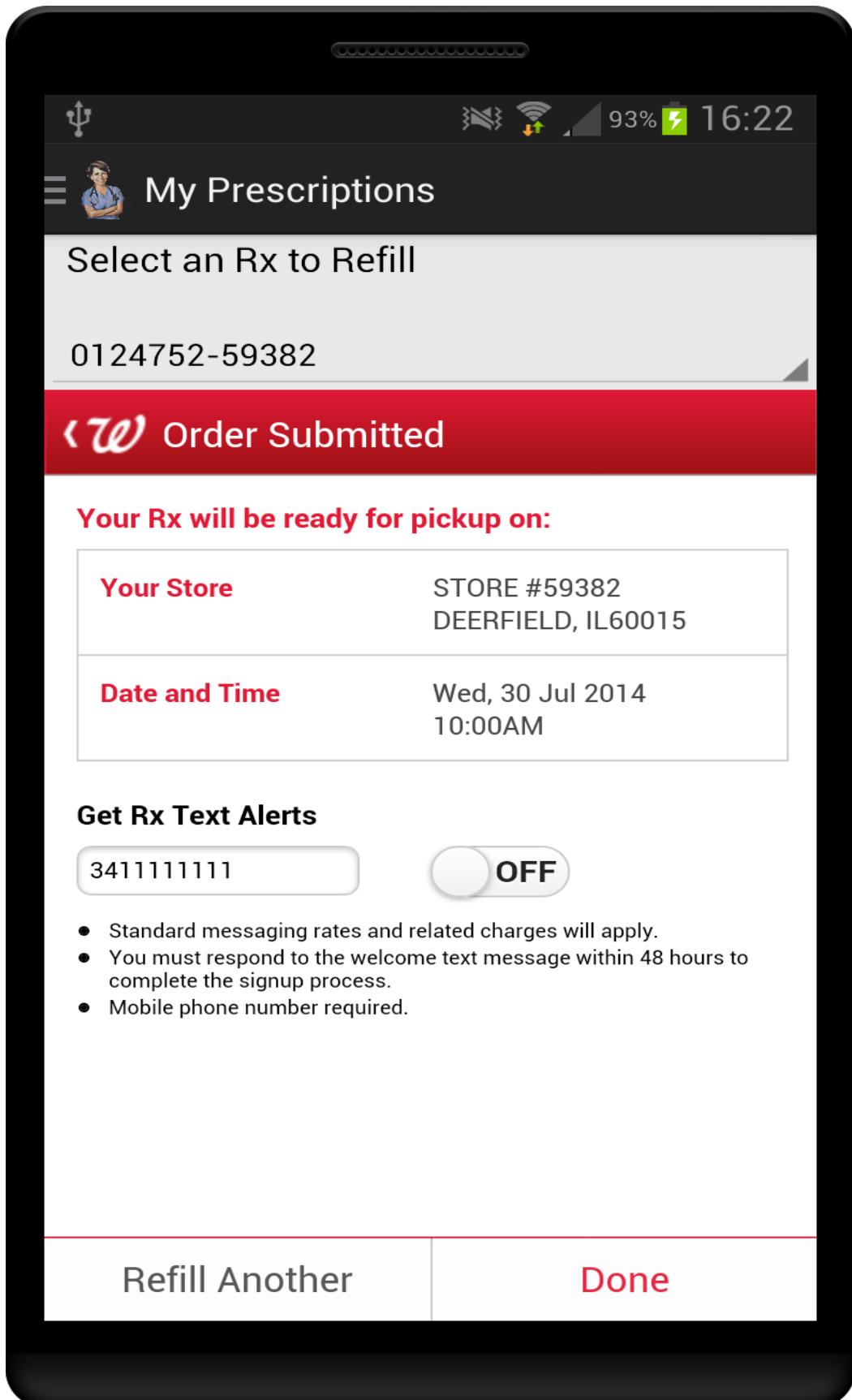
Select Drug Back To Results

## **2) Refill From Walgreens:**

Once when the medications are over, the user would go for a refill if he/she is gonna continue the same medication. The Walgreens Pharmacy Prescription API allows users to quickly order refills of prescriptions originally filled at one of the 8,000+ Walgreens pharmacies. Application users can order refills in seconds, as well as select the option to opt-in to receive a text alert when their prescription order is ready for pick up.

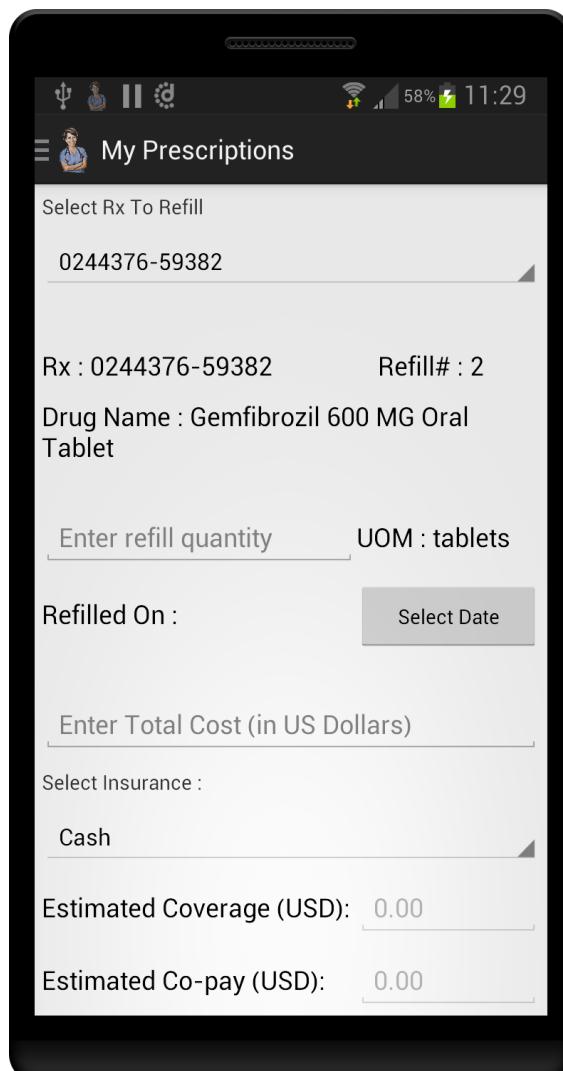
The health care focused Pharmacy Prescription API is designed to increase prescription compliance and aid in personal health management through automated refill alerts and a streamlining of the refill process. By offering easy prescription refills through this health management API, Walgreens hopes to further increase health care adherence by reaching a wider audience through third party health care apps.





### 3) Record Refill:

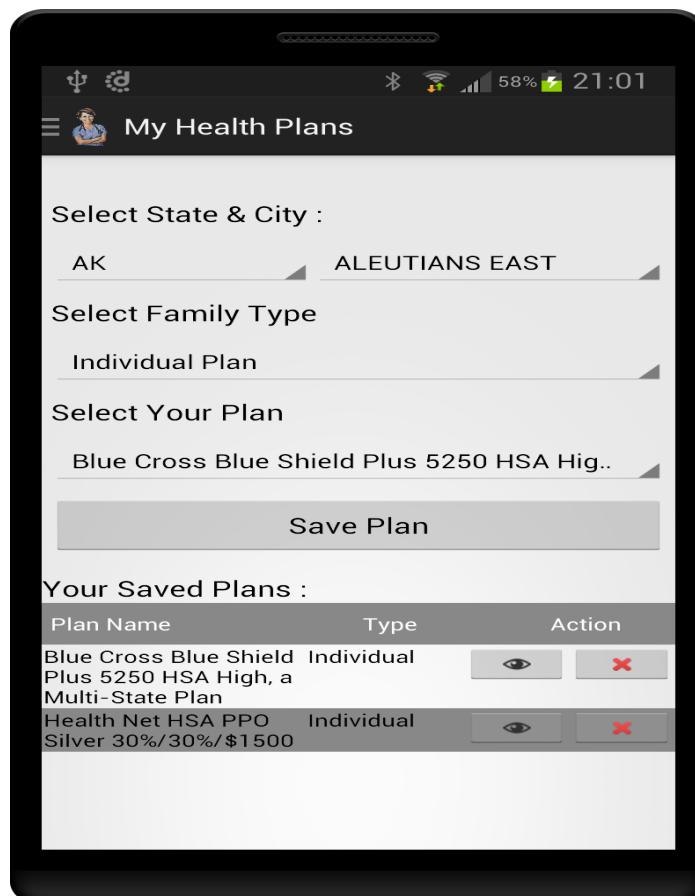
After successfully refilling the Rx with Walgreens, we capture the expense involved in the transaction. We record information's such as refill quantity, refill date, insurance plan that covers the refill expenses, plan coverage and plan copay details in our mongo db under "refill" collection. All these components are required for the expense report generation.

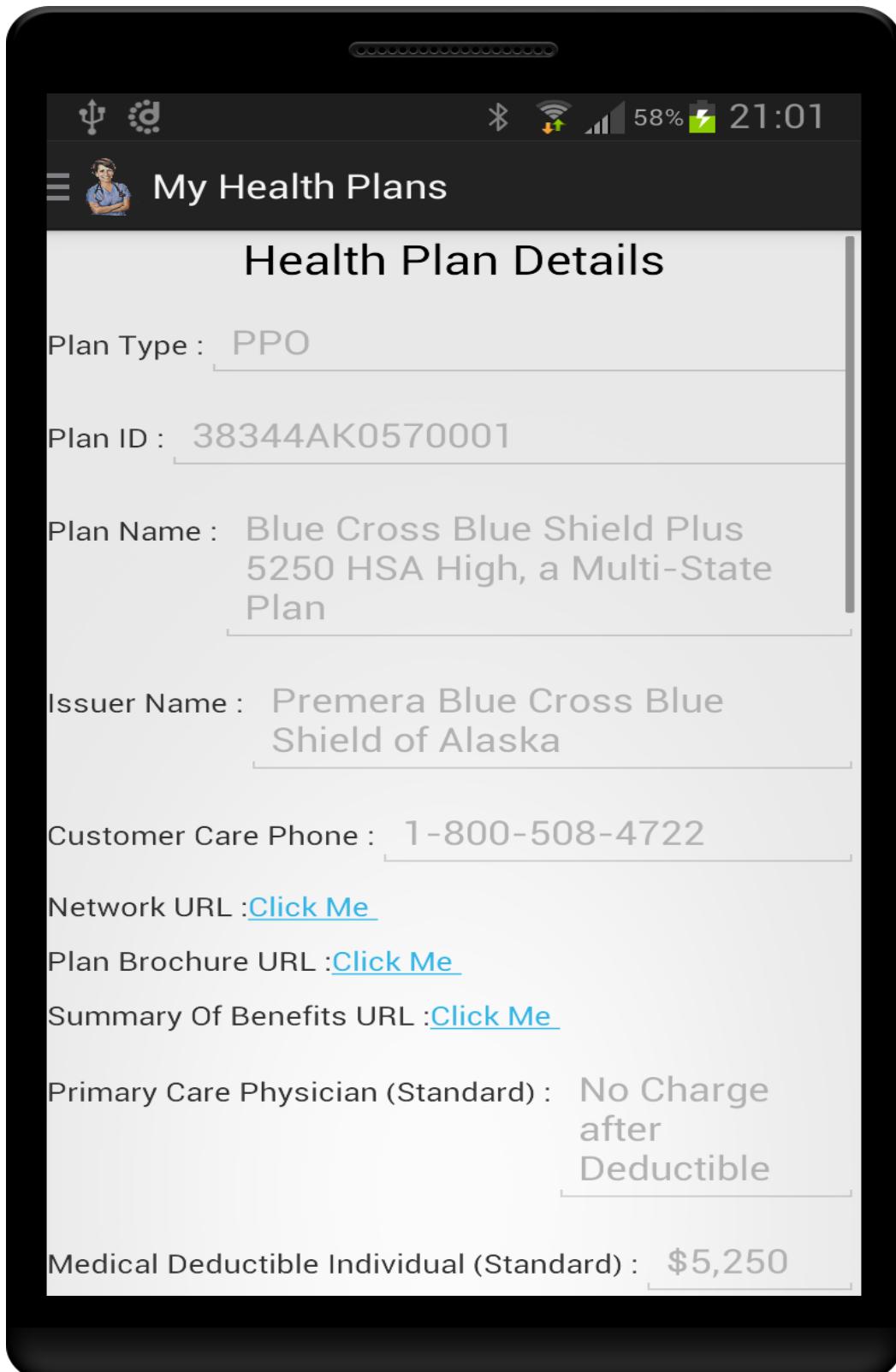


## MY HEALTH PLANS:

The end user has the ability to add his/her health plans to our application. The user has the facility to search for plans from our database which holds around 77,318 health plans. If the user would like to get detailed information about a health plan, at that instance of time we make an API call with the below mentioned URL and obtain the response data in JSON format. This json data is then parsed and appropriate values are displayed in the front end.

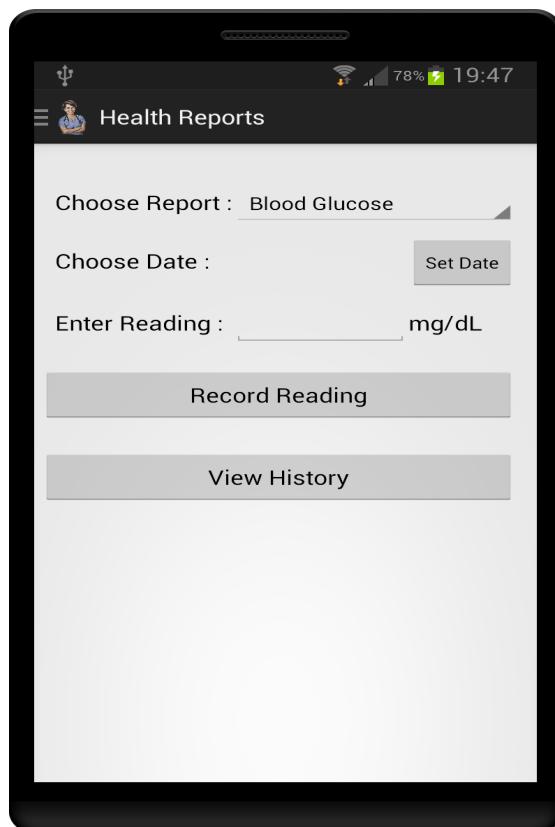
[http://data.healthcare.gov/resource/b8inzsz6k.json?&\\$select=state,county,issuer\\_name,plan\\_id\\_standard\\_component,plan\\_marketing\\_name,plan\\_type,metal\\_level,customer\\_service\\_phone\\_number\\_toll\\_free,network\\_url&&\\$where=plan\\_id\\_standard\\_component=<PLAN\\_ID>AND state =<STATE> AND county=<COUNTY>](http://data.healthcare.gov/resource/b8inzsz6k.json?&$select=state,county,issuer_name,plan_id_standard_component,plan_marketing_name,plan_type,metal_level,customer_service_phone_number_toll_free,network_url&&$where=plan_id_standard_component=<PLAN_ID>AND state =<STATE> AND county=<COUNTY>)





## **HEALTH REPORTS:**

The report generator purpose is to take data from a source such as database, XML, spreadsheet and use it to produce results which satisfies a particular human readership. The user has the ability to feed in his/her health report details, as of now we have space to accomdate only the following 3 reports: Blood Glucose, Height and Weight report. After hitting “Record Reading” button, the values are loaded into “user\_report” collection in our MongoDB. The “View History” provides a tabular view of data recorded so far with respect to the report type. A line graph as well is depicted to have better readability on the health report.



USB  57% 20:58

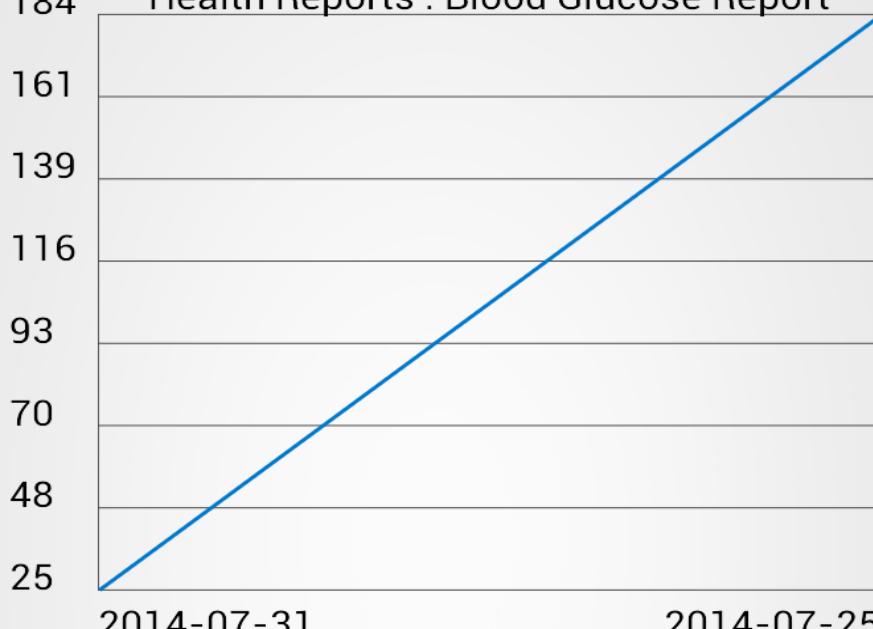
≡  Health Reports

## Health Reports : Blood Glucose Report

Date	Value	Delete
2014-07-31	25.0 mg/dL	X
2014-07-25	184.0 mg/dL	X

[Go Back](#)

184      **Health Reports : Blood Glucose Report**

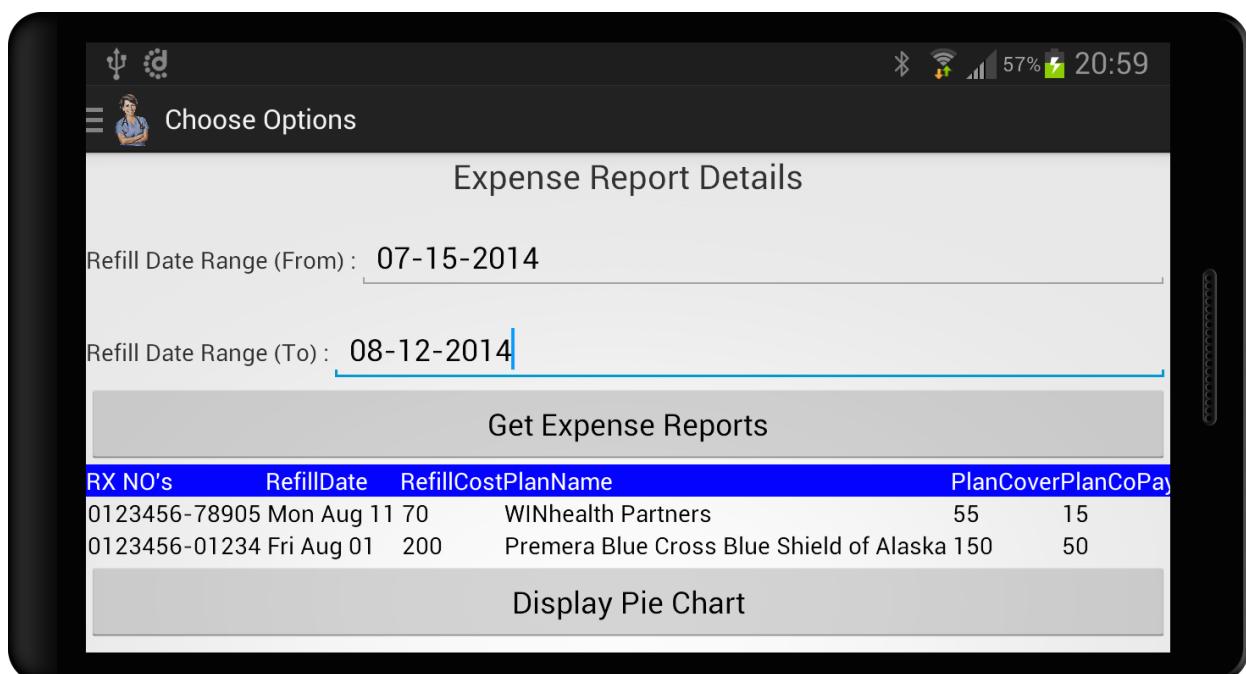


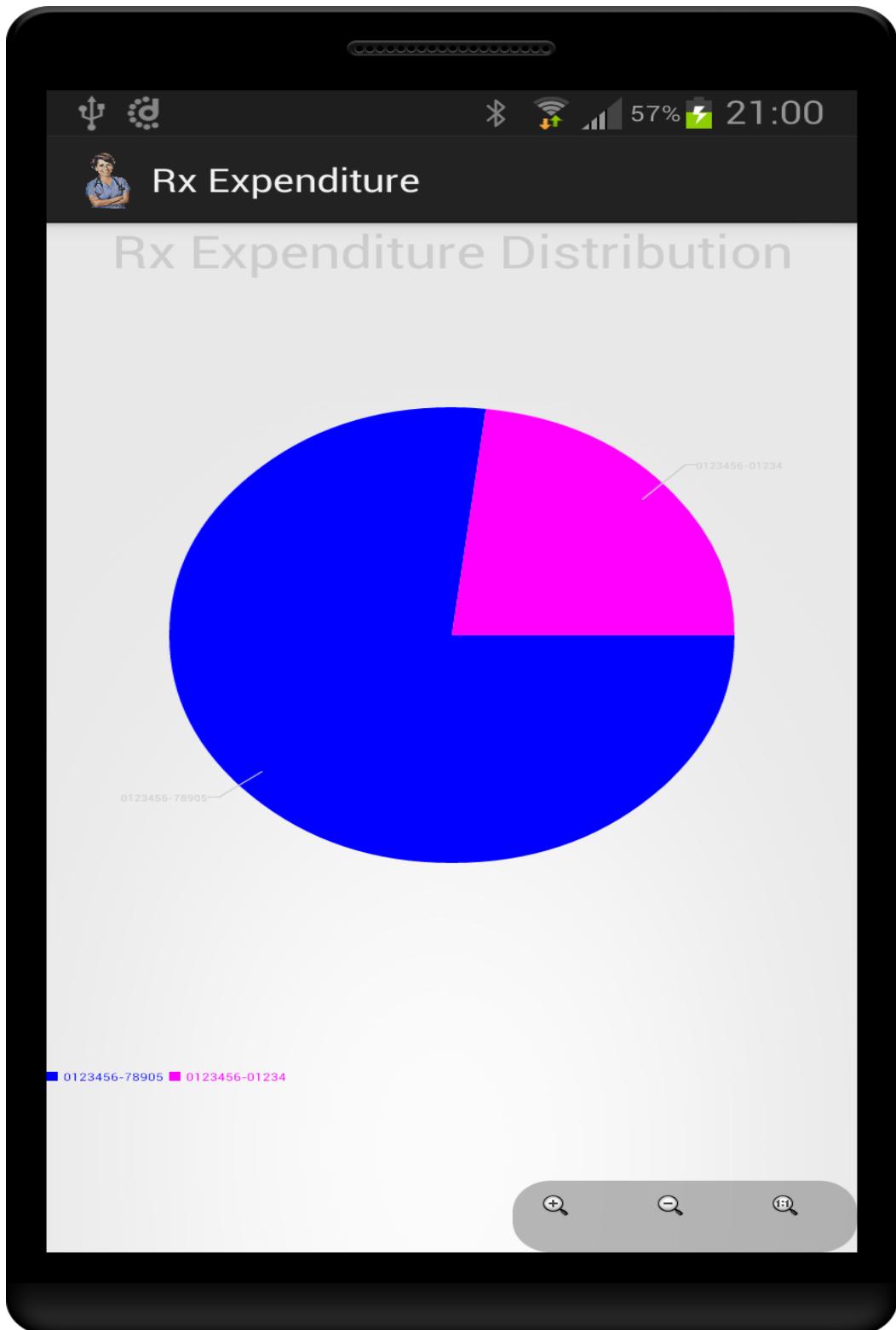
The graph displays a single data series representing blood glucose levels over time. The Y-axis represents the glucose level in mg/dL, ranging from 25 to 184 with major grid lines every 15 units. The X-axis represents the date, with markers for 2014-07-31 and 2014-07-25. A blue line connects two data points: (2014-07-31, 25.0) and (2014-07-25, 184.0). The line has a positive slope, indicating a significant increase in glucose levels between the two dates.

Date	Glucose Level (mg/dL)
2014-07-31	25.0
2014-07-25	184.0

## **EXPENSE REPORT:**

Based on the data which was gathered at the record refill screen, the application has the capability to generate expense report for the specified time period. The tabular view of the expense report would get generated upon hitting “Generate Expense Report” button. The table of data holds detailed information like Rx Number, Refill Date, Refill Expense, Health Plan, Plan Coverage and Plan CoPay. To have better understanding on the expense report, pie chart as well is generated.





## NEAR BY HOSPITALS / PHARMACIES:

The near by hospital and pharmacy locator is dynamic in nature. How it works?? The current geolocation of the end user is identified using GPS / Network provider and then the position is plotted over google maps. Google places API is consumed to identify the information about hospital and pharmacies with its latitude and longitude. The response data is of the json format, which is then parsed to obtain the specific details. Customized markers are then used to plot these obtained values.

<https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=42.339243,-71.090078&radius=2500&types=hospital&key=<API KEY>>

