
MOBILE DATA COLLECTION AND ANALYSIS

May 8, 2019

Submitted To

Dr. Vikram Pudi
DSAC Labs ,IIIT Hyderabad

Submitted by

Chittaranjan Rath 2018201007
Nitish Srivastava 2018201012
Prakash Nath Jha 2018201013
Ishan Tyagi 2018201017

Contents

0.1	Open Source Projects Investigated	2
0.2	Purple Robot Application Overview	5
0.3	Data Base Design Implementation	8
0.4	Deployment	9
0.5	Issues Faced/Future Aspects	11
0.6	Dashboard-For Data Visualization	12

0.1 OPEN SOURCE PROJECTS INVESTIGATED

1. *RADAR CNS*:

Pros:

Well known platform for collecting and analysing sensor data with a in built analytics tool for analysing sensor data developed King's College and Janssen Pharmaceutica NV. Supports both passive and active data collection by collecting both sensor data and by a questionnaire app. Questionnaire app is can be used to do a survey.

Cons:

Lack of proper documentation available and no proper channel of communication with the RADAR CNS community to discuss the problem faced during installing the software. It not developer and individuals friendly as mentioned in https://en.wikipedia.org/wiki/Mobile_phone_based_sensing_software Available on only Android platform not on IOS. Captures fewer number of sensors data. No user or developer forum available. We also raise an issue in github but receive no good response. Does not provides encrypted data transfer.

reference link: <https://github.com/RADAR-base>

2. *Beiwe*:

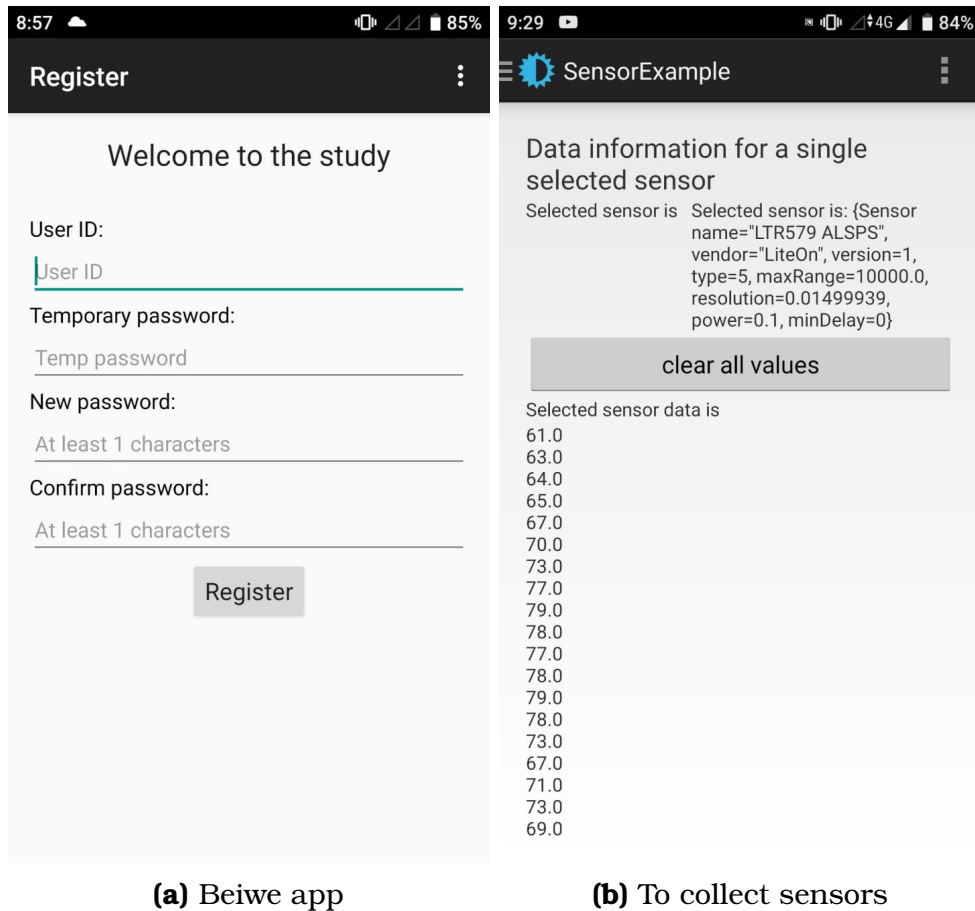
Pros:

Its developers, researchers, individual friendly developed by HSPH Onnela Lab with built in web dashboard, back-end and data analysis pipeline. Both Android and IOS app available. Optimised for battery drain problem. All data on phones, on the server, and in-transit use industry-standard encryption techniques

Cons:

Backend on AWS which makes it necessary to have AWS developer account with subsequent cost associated with it to host it on AWS servers.

reference link: <https://github.com/onnella-lab/beiwe-android>



3. Custom App:

Other than exploring open source projects we also attempted to build our own full stack system containing both back end and front end. Here's are few:

- **Battery Monitor App:** In this, we build a app which takes the information of your smart phone battery like Battery type, battery level, battery health etc.
- **Probe Sensors:** In this , we build a app which captures the different-different sensors data like light sensors, proximity sensors, Gravity sensors, Battery information and lot more.
- **To get Data of Selected Sensors:** In this, we can modify which sensor data we want to collect and moreover we can also modify the time interval of sending the data, this will significantly decrease the load the on Processor.
- **GPS tracking app:** In this, we collect the GPS location of user and it collect on time interval and collect only if there is significant change in location.

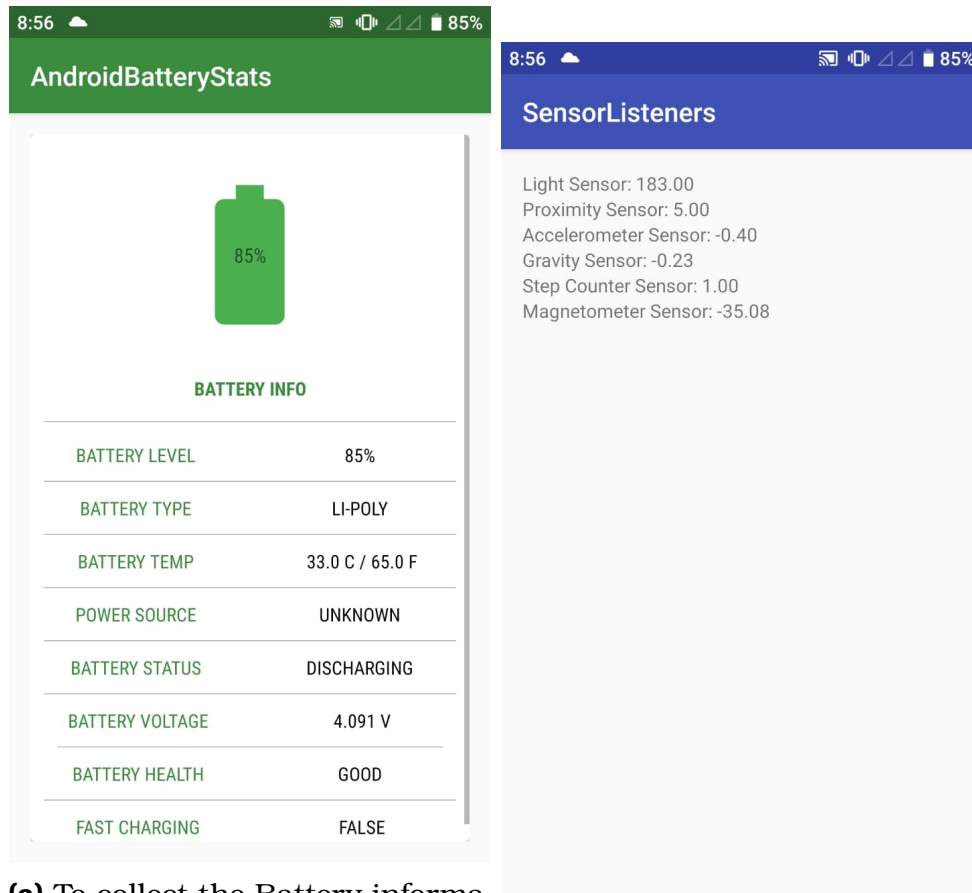
- **Questionnaire app:** In this, we just have questions and answer interface to ask direct questions to get users information.

Pros:

- **1** Having built the Entire system from scratch was that we would have entire control and understanding of the system.
- **2** System could have been flexible as per our requirements.
- **3** Using open source projects in some sense imposes some restrictions based on the initial system design of the open source projects.

Cons:

- **1** Building the entire system from scratch is time consuming and we have the time constraints and that may not be battery efficient which can be a serious problem going ahead.
- **2** Security is also one of the issue, because we are accessing the users information , we have to make sure that there is no threats to user's data.
- **3** Scalabilty is also one of the main reason.
- **4** We have to implement the lot of funnctionality to collect data, which makes our app heavy and in rural India ,generally people don't have smart phone with powerfull processor. So we need a very lite weight app.



(a) To collect the Battery information.

(b) To collect different sensor data.

0.2 PURPLE ROBOT APPLICATION OVERVIEW

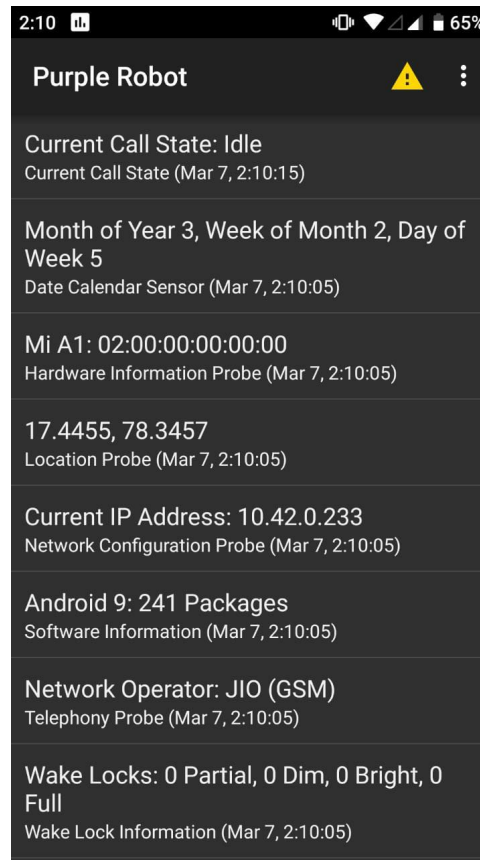
Pros:

Well structured and documented code available on github with built in dashboard and analytics tools with backend in Django and database as PostgreSQL. Its developers, researchers, individual friendly developed developed by CBITS, Northwestern University. Captures larger number of sensors and user data. Provides encrypted data transfer between the user hardware and server. Optimised for battery drain problem

Cons:

- 1 Available on only Android platform not on IOS.
- 2 Github repository is not very active

reference link: <https://github.com/cbitstech>



(a) Screenshot of PurpleRobot app

BUILDING PURPLE ROBOT

Purple Robot is a large application that includes a variety of service integration libraries as well as full language runtimes for both the JavaScript and Scheme programming languages.

Consequently, it cannot be built by traditional Android IDEs given its reliance on ProGuard to shrink installation below the 65k methods limitation imposed by Android.

The instructions below describe how to create an APK Android package for your own testing and development.

PREREQUISITES

Purple Robot depends upon the following packages for compilation and installation:

- **Git** : This may already be installed on your system.
- **Android SDK** : An SDK installation is included with Android Studio.
- **Gradle** : Install version 2.2.1.
- **Java** : This may already be installed.

After installing these prerequisites, open the Android SDK Manager and verify that the following packages are installed:

- Android 5.1.1 (API 22) SDK Platform
- Android SDK Build-tools 21.1.2
- Android Support Repository

Pull from GitHub

Once the prerequisites are installed, clone the Purple Robot Git repository at <https://github.com/cbitstech/Purple-Robot.git>

BUILDING PURPLE ROBOT FROM ANDROID STUDIO

Once Purple Robot can be built from the command line, you can import the project into Android Studio and use that IDE as your primary Purple Robot development environment.

Once Android Studio is installed, select "Import Non-Android Studio Project" and browse to the Purple Robot directory in downloaded repository.

If prompted to use a different SDK, select Android Studio's default SDK. This will update the local.properties file.

When prompted whether to use the Gradle wrapper, select Cancel and select the location of your Gradle 2.2.1 installation. Android Studio will sync the Purple Robot build files.

Once Android Studio has successfully synced with the project, you should switch the build type to release, build the app's APK, and install it on your device

0.3 DATA BASE DESIGN IMPLEMENTATION

Application is capable of holding multiple Sensor details. But as part our project we have only stored some specific relevant Sensor details. Following are the table name wrt each Sensor information we stored:

Table 1: Sensor Information w.r.t Database Schema

Sensor Name	Database Table Name
Battery	Battery
WifiDetails	Wifi Access Points
SunriseSunset	SunriseSunsetInformation
Call State	CallState
Network Details	Network
Well Known Places	PopularLocations
Wake Lock Details	Wake Lock Information
Time & Month	Date Calendar

· **Table Battery :**

Stores Battery level , Scale , Temperature , Voltage , Time stamp

· **Table Users Location :**

Stores the latitude and longitude along with time stamp.

· **Table Call state :**

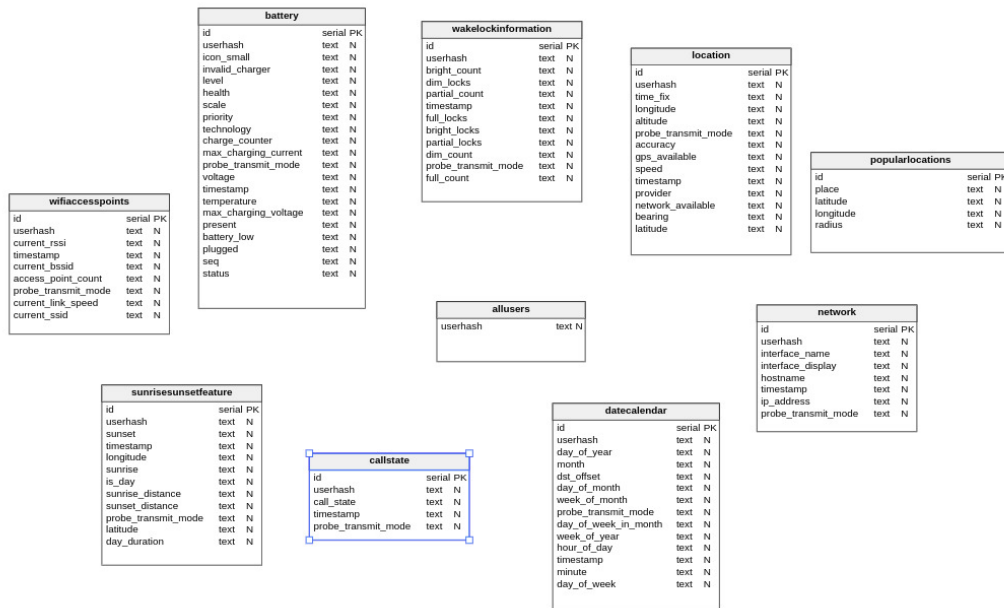
It shows how much users talk on phone or not on basis of current time stamp.

· **Table Network information :**

It shows the connectivity information of the user with internet. Collects Host Name , Host IP of the user.

· **Table Wifi Information :**

It shows the connectivity information of the user with Wifi. Collects Access Points Count , Link Speed , Mac Address of the access point.



(a) Database Schema

0.4 DEPLOYMENT

SPECIFICATIONS

- * **Operating System** CentOS release 6.10 (Final)
- * **FrameWorks** Programming Language : Java , Python
 IDE : Android Visual Studio , Visual Studio
 Web FrameWork : Flask , Decorator
 Visualization : Plotly
- * **Database** PostGreSQL 8.9 (and above)

STEPS

- * **App Development • Configure End Point** : Required Configuration is to be made in configuration (code) of Purple Robot application in order to redirect the data to respective server of database.
- * **Create APK file** : Post Configuration in the above step, we can proceed with apk file creation.

- **Installation of APK** : Respective APK file can be used for installation on the mobile phones.
- **Set up for Mobile App** : Please refer to the "User Case Documentation" for the step-by-step procedure for configuring and using app in bestway.

* **Web Development • Software Packages** :

Python for server implementation.

JSON for data transfer.

Plotly for data visualization.

Flask for web development.

OpenSSL for user hash generation.

Cryptography techniques for securing the data.

- **Start Server** : Run app.py file inside frontend folder to start the front end application.
- **Verification of Application Running** : Please check website <http://dsac.msensui.iiit.ac.in> for accessing user interface over internet.

0.5 ISSUES FACED/FUTURE ASPECTS

ISSUES ENCOUNTERED

- a)** How to Learn to Configure a Mobile App :
<https://github.com/thehyve/RADAR-AndroidApplication>
- b)** Challenges faced during Plotly Implemntation
- c)** Allowing of port numbers and challenges faced with different Python version
<https://linuxconfig.org/how-to-enable-disable-firewall-on-ubuntu-18-04-bionic-beaver-linux>
- d)** Set up problems and resolution from following references :
<https://stackoverflow.com/questions/7695962/postgresql-password-authentication-failed-for-user-postgres>
<https://askubuntu.com/questions/831262/how-to-install-pgadmin-4-in-desktop-mode-on-ubuntu>

FUTURE ASPECTS OF PROJECT

- * The widespread use of sensor-packed mobile devices, including smartphones, provides us with an unprecedented opportunity to study and develop applications for people's daily lives.
- * Collected data can be used to provide device security and theft prevention by identifying the last known location from application.
- * Identification can also be used to automatically personalize mobile device settings after identifying the current user of the device and his/her current activity.
- * Applications that recognize activities and adapt phones as a result (such as selecting a certain playlist or sending calls to voicemail while running) can encourage healthy behavior. Moreover, the records of a user's activity can be tracked and reported over time, enabling health and fitness applications for users.

RELATED LINKS

- * **Github Projectlink**

<https://github.com/ConnectNitish/CapturingSensorData>

- * **DashBoard url** <http://dsac.msensui.iiit.ac.in>

0.6 DASHBOARD-FOR DATA VISUALIZATION

It shows the description of data which was collected from all the users of purple robot app. It has user-friendly interface which shows various information regarding the users and allows us to analysis the data. It shows the following information:

- **Battery Information :**

It shows how the battery varies for the particular users

- **Users Location :**

It shows the location of user according to their latitude and longitude.

- **Call state :**

It shows how much users talk on phone or not.

- **Network information :**

It shows the location of user

- **Wifi Information :**

It shows the speed of user's net connection and others things.

Dashboard is divided into major parts described below :

- **All User Details :**

Shows the details of all the data for most frequent / active users.

- **Specific User Details :**

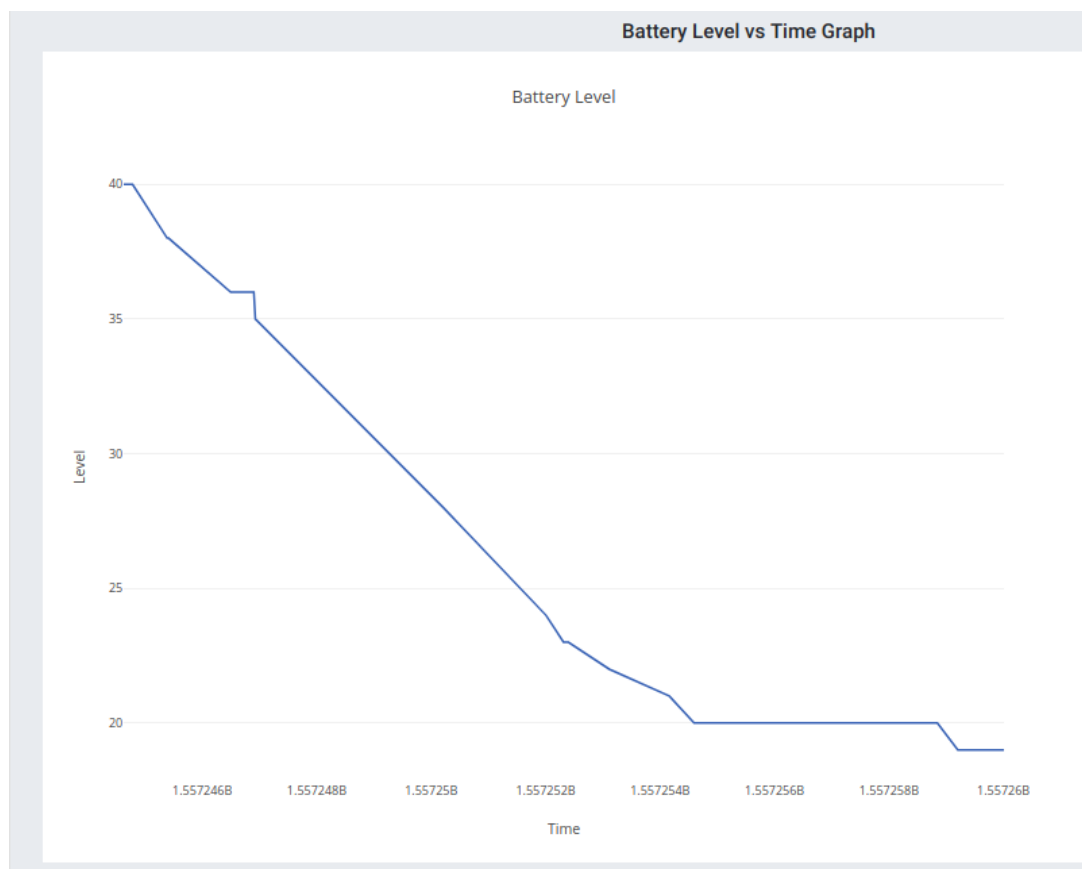
User interface is responsible for showing the details of the Particular user.

- **Location Specific Details :**

Provide the details of the user who has visited to a specific place (w.r.t Latitude and Longitude of the information shared to the system.

BATTERY INFORMATION:

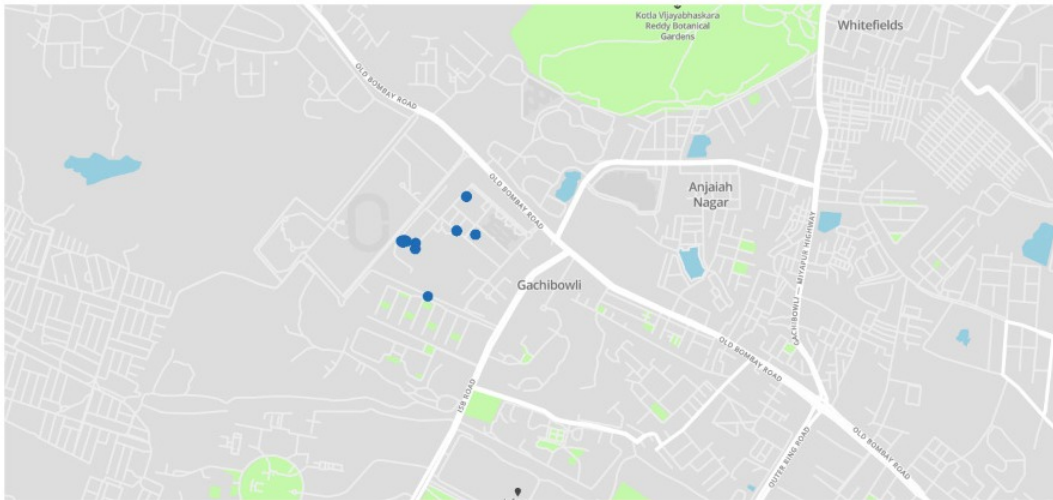
- (a) It shows how the battery varies for the particular users w.r.t time.
- (b) It can give us the information that ,at what time users is using their most of battery and we can use to location at that time to get inference from the battery.
- (c) It has the battery level vs time Graph.



(a) Battery information

USERS LOCATION:

- (a) It shows the location of particular user according to their latitude and longitude.
- (b) It can give us the information that ,at what time users is present where , this will very helpful when we use this information with others result.



(a) Location information

Wifi Information:

- (a) It shows the wifi information of particular user.
- (b) It gives us various information like link speed , Count of AccessPoint and BSSID.
- (c) With the help of this ,we can infer various information like in which place internet speed is good etc.

Wifi Information			
It shows the various wifi regarding network of User			
UserHash	Link Speed	Access Point Count	BSSID
34237272481aa6a02ea94799695a6982	72	5	38e60a:df31:a6
34237272481aa6a02ea94799695a6982	72	6	7c7635:dd55:95
34237272481aa6a02ea94799695a6982	72	5	f894:c224:547d
34237272481aa6a02ea94799695a6982	72	2	7c7635:dd55:95
34237272481aa6a02ea94799695a6982	72	8	7c7635:dd55:95
34237272481aa6a02ea94799695a6982	72	4	f894:c224:547d

(a) Wifi information

NETWORK INFORMATION:

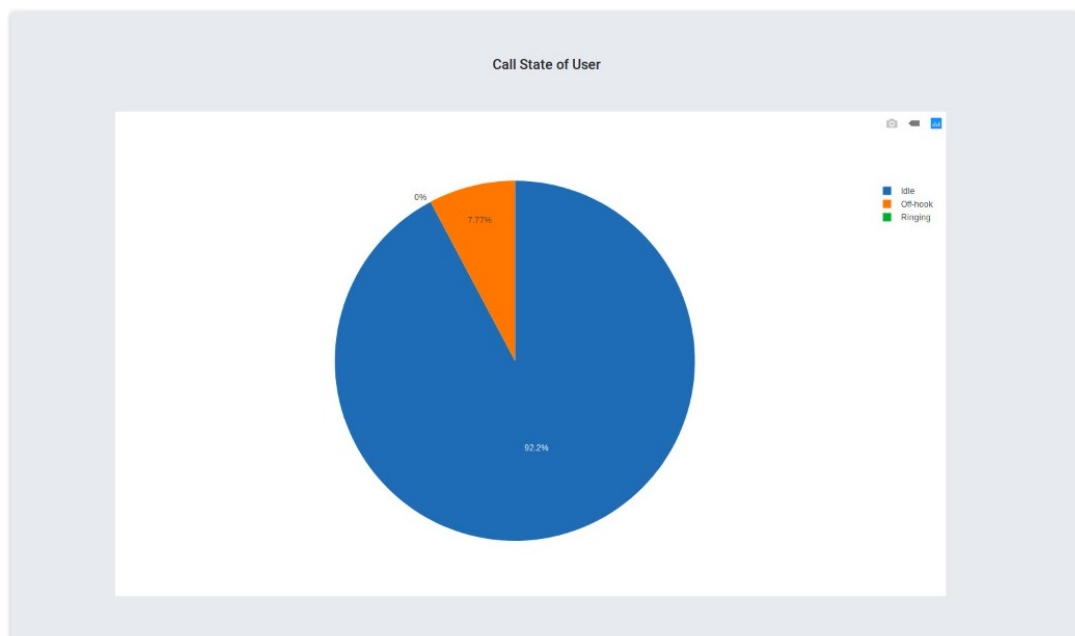
- (a) It shows the Network information of particular user.
- (b) It gives us various information like IP and Hostname of user's phone
- (c) With the help of this ,we can infer that which mobile is user are using .

Network Information		
It shows the various information regarding network of User		
User Hash	Hostname	IP Address
13b03abb6901f35dc4edbec044198af5	10.42.0.125	10.42.0.125
23859457968188f76ba17fe738d10d5d	10.42.0.125	10.42.0.125
34237272481aa6a02ee94799695a6982	10.42.0.12	10.42.0.12
34237272481aa6a02ee94799695a6982	192.168.43.187	192.168.43.187
34237272481aa6a02ee94799695a6982	10.42.0.25	10.42.0.25
34237272481aa6a02ee94799695a6982	10.42.0.125	10.42.0.125

(a) Network information

CALL STATE INFORMATION:

- (a) It collect the user's call state information.
- (b) It has 3 states
 - **offhook** : When the User is in progress of calling state
 - **Idle** : User is in normal (no calling) state.
 - **Ringling** : User is in progress of a call.
- (c) Below Figure show the contribution of Call State of the user for last 24 hours.
Contribution of each field is shown in percentage.



(a) Call State Piechart on percentage basis

NAVIGATE TO SPECIFIC USER DETAILS:

- As shared initially , in order to preserve Integrity of user data , only User-Hash instead of Actual User Name is stored in application.
- Below Figure shows all the details of User-Hash, when a click action is performed on "Users" button in navigation bar.
- Each Button displayed , on click of which User will be navigated to other "User Details" page , which shows specific details for the user.

The screenshot displays a mobile application interface. At the top, there is a table with four columns: UserHash, Device Name, Version, and Location. Below this table, there is a section titled "Wifi Information" with a subtitle "It shows the various wifi regarding network of User". This section contains a table with four columns: UserHash, Link Speed, Access Point Count, and BSSID. To the right of the "Wifi Information" table, there is a vertical list of 20 blue buttons, each containing a UserHash. At the bottom right of the screen, there is a red button labeled "CLOSE".

UserHash	Device Name	Version	Location
a269f1041ae5073302d70beac5137ae2	redmi6pro-redmi	10.42.0.25	
b50f1867e01021d3134b0057ecb7990d	redmi6pro-redmi	10.42.0.25	
d9ffaca46d5990ec39501bcd722ee7a1	redmi5-nitish	10.42.0.12	
e051a758734abbefed3347192e64ed9f		10.42.0.125	10.42.0.125

UserHash	Link Speed	Access Point Count	BSSID
34237272481aa6a02ee94799695a6982	72	5	38:e6:0a:e0:7c:76:33:a6
34237272481aa6a02ee94799695a6982	72	6	7c:76:33:a6:18:94:c2:29
34237272481aa6a02ee94799695a6982	72	5	7c:76:33:a6:18:94:c2:29
34237272481aa6a02ee94799695a6982	72	2	7c:76:33:a6:18:94:c2:29
34237272481aa6a02ee94799695a6982	72	8	7c:76:33:a6:18:94:c2:29
34237272481aa6a02ee94799695a6982	72	4	18:94:c2:29:7c:76:33:a6

068FD0571F7CC52704983D35F0D90809
FDE6295AF023AB40685ECCEEE22C402E
B50F1867E01B21D3134B0097ECB7990D
13B03AB86901F35DC4ADBEC044198AF5
73803249C6667C5AF2D51C0DEDEFAE487
D9FFACA46D5990EC39501BCDF22EE7A1
B284B54D674CE5F1D6776318177FC965
5B632C70B59FE3539C6B354FC1448A38
F38F55A32E342A6219002F8F1AA98031
E251A758734ABBEFE63347192E64ED9F
A269F1B41AE8073302D70BAAC6127AE2
34237272481AA6A02EA94799695A6982
541C787D9DEFCC152FAF2A222AA96FE2
67D978D85BF1F619274EF32034FADE19
93E2F4935402DA5361FCAEB636FC9CD2
2385945796818BF76BA17FE738D10D5D
5037284550221285C91DE86262EA0A0D

CLOSE

(a) To select the user by their userHash

REFERENCES

https://en.wikipedia.org/wiki/Mobile_phone_based_sensing_software

<https://github.com/cbitstech>

<https://github.com/cbitstech/Purple-Robot>

<https://plot.ly/javascript/>

<https://developer.android.com/docs>