USER MANUAL FOR PURPLE ROBOT

May 8, 2019

Submitted to :

Dr. Vikram Pudi
DSAC Labs,IIIT Hyderabad

Team Members:

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

Contents

0.1 Introduction

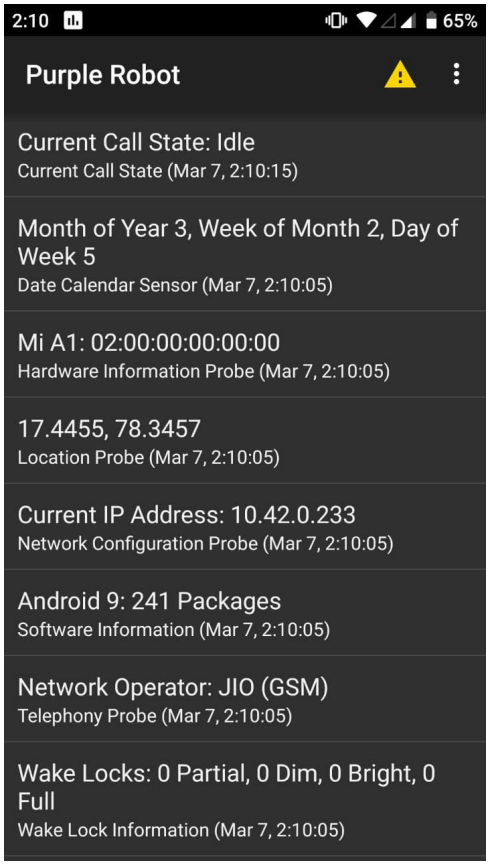
Pros:
Well structured and documented code available on github with built in dasboard and analytics tools with backend in Django and database as PostgreSQL. Its developers, researchers, individual friendly developed developed by CBITS, Northwestern University. Captures larger number of sensors and user 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.

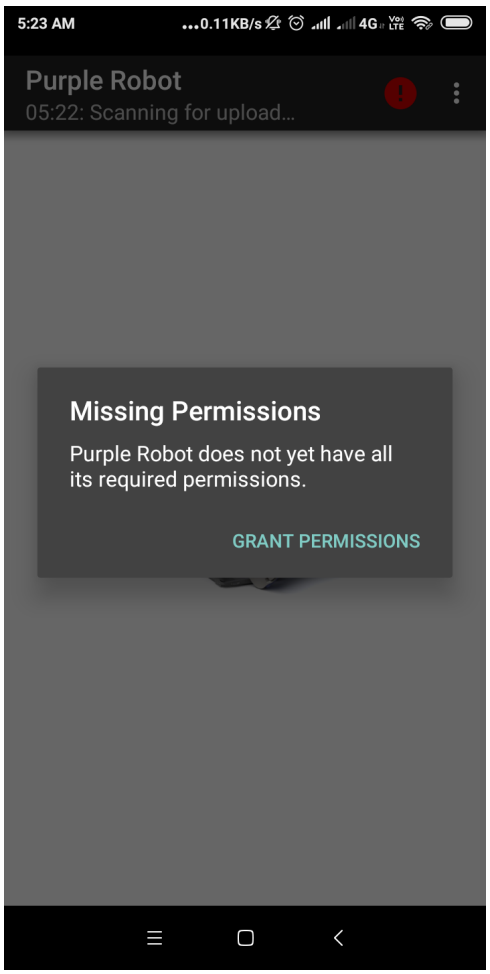
Once Android Studio has successfully synced with the project, you should switch the build type to release, build the app's APK, and install it on your device

0.2 Step by Step Setup

Post installation of android apk this page shows that app has been successfully installed on mobile phone.

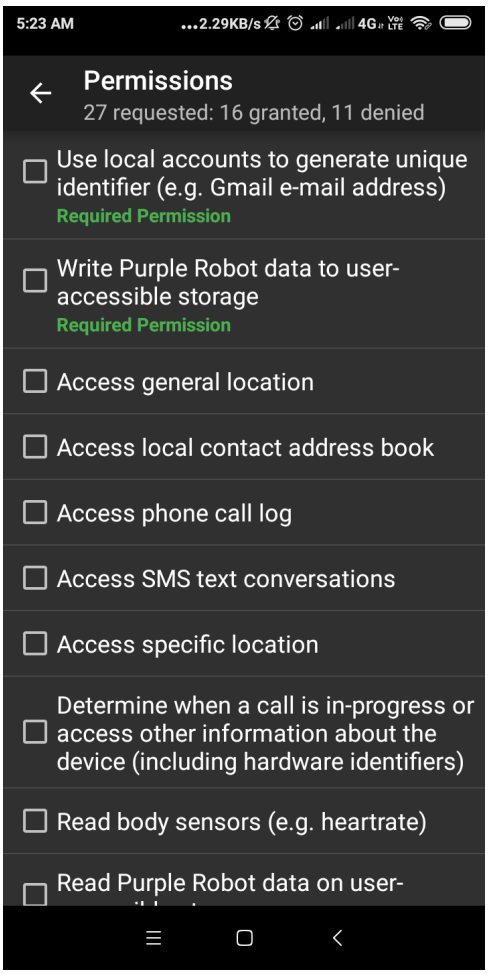
(a) StartPage

Page is requesting for pre-requisite sensors access . Without this request grant you will not be able to proceed further.



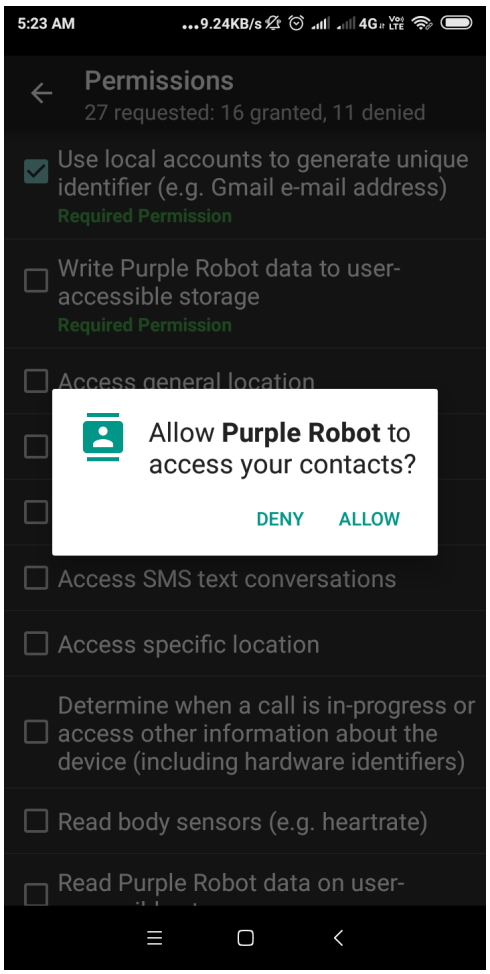
(a) Grant the Missing Permissions.

Shows all the list of Permission which user can give . Specific permission is granted whenever user will check the respective check box.



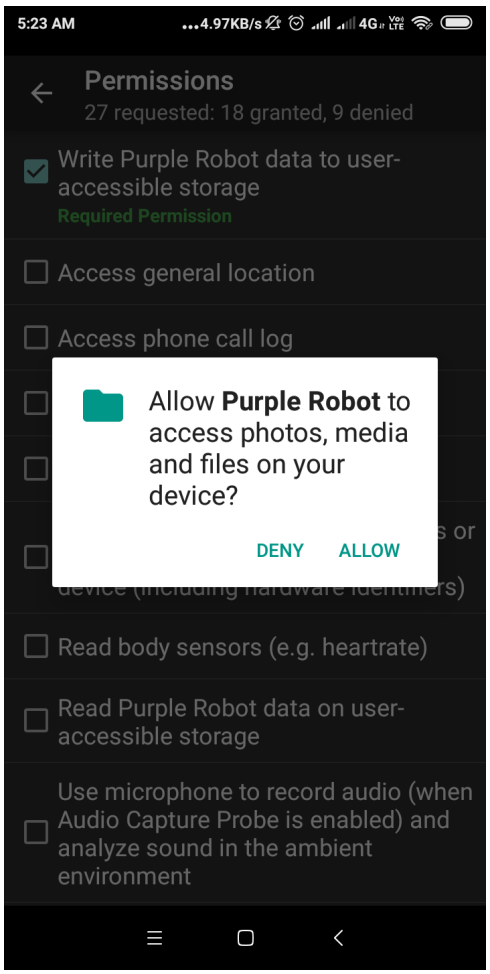
(a) List of Permissions.

As User proceed with granting permission , confirmation box appear to alert user and request an re-confirmation for allowing the access.



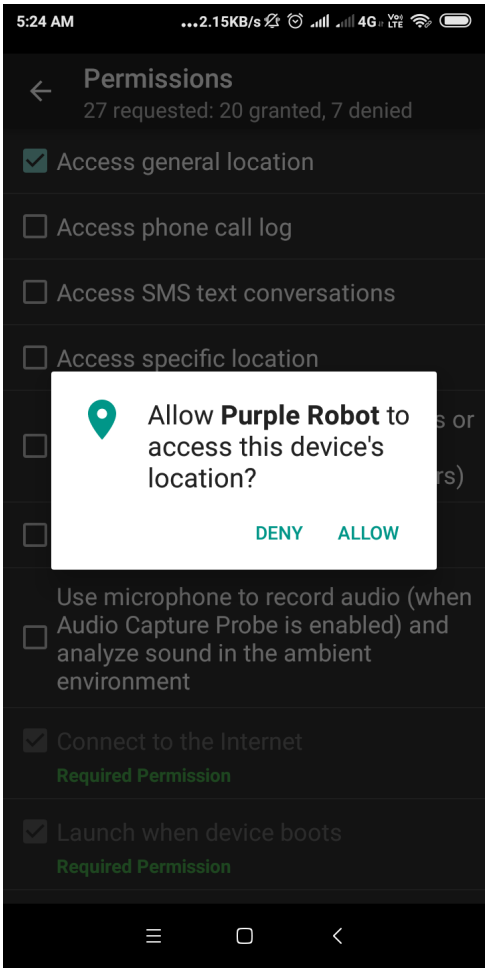
(a) UserAccessControl.

As User proceed with granting permission for media details , confirmation box appear to alert user and request an re-confirmation for allowing the access.



(a) Allow for Media Details.

As User proceed with granting permission for Location details , confirmation box appear to alert user and request an re-confirmation for allowing the access. This is an optional permission.

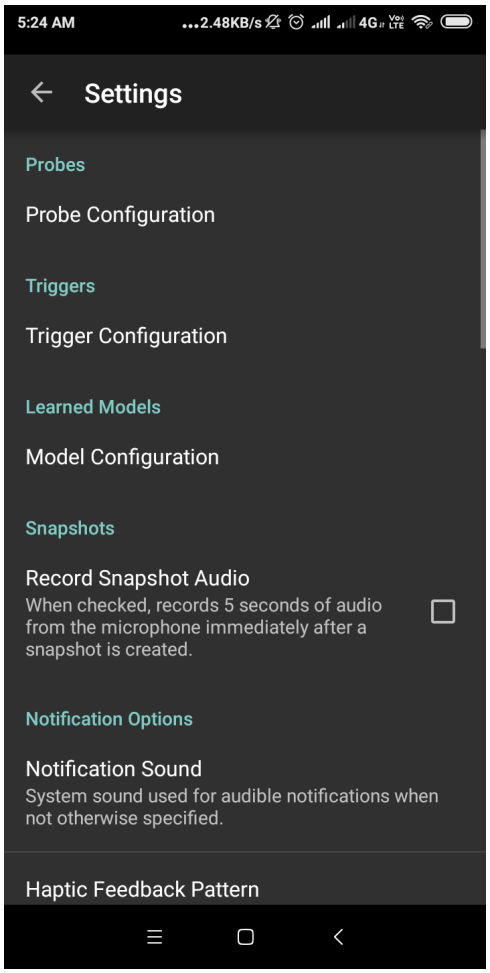


(a) User Access For Location .

Now as we have grant the permissions to acquire data, we need to perform certain settings to enable the probes(data) for collecting data.

Probe configuration need to be checked for the same.

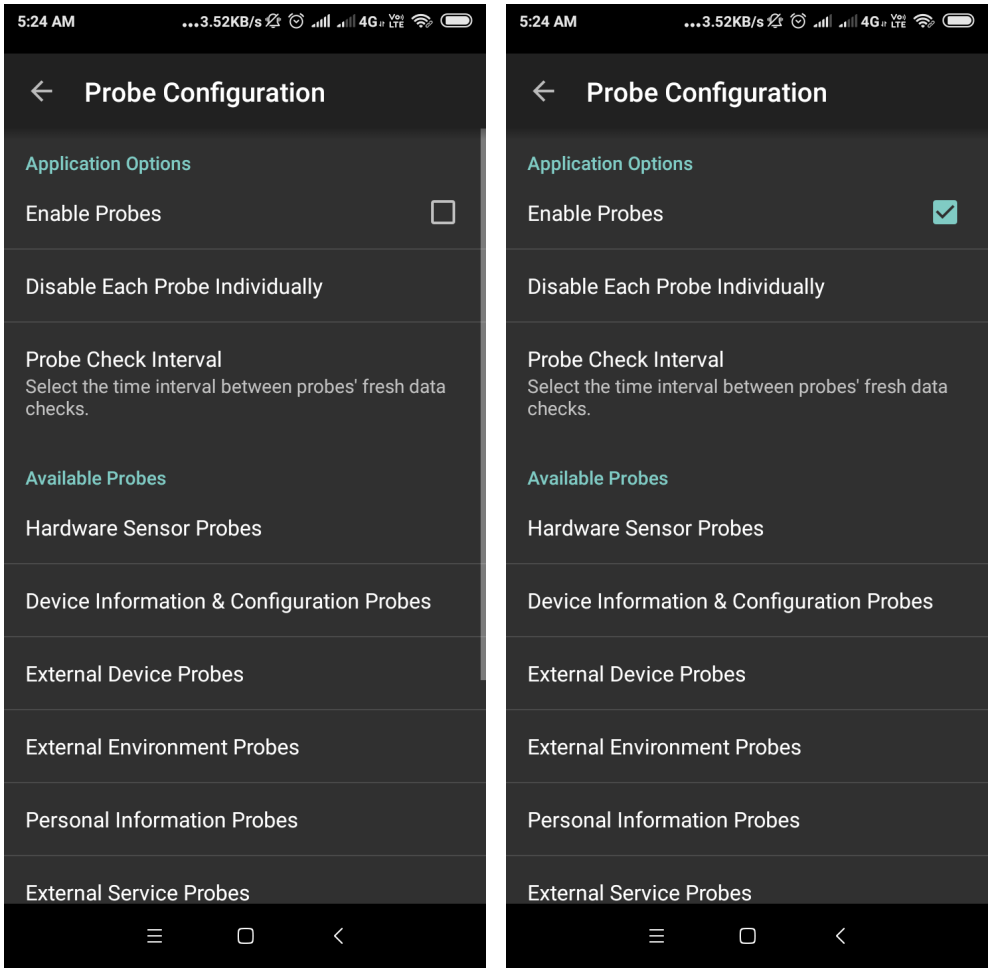
Trigger configuration is used for sending the Probe when any particular type of trigger is initiated.



(a) Perform Settings

Enables Probes check Box , is mandatory to be checked.

Probe Categories As per the type of sensors, the respective probes can be configure(Enable/Disable) accordingly.



(a) Probe configuration.(Before) **(b)** Probe configuration.(After)

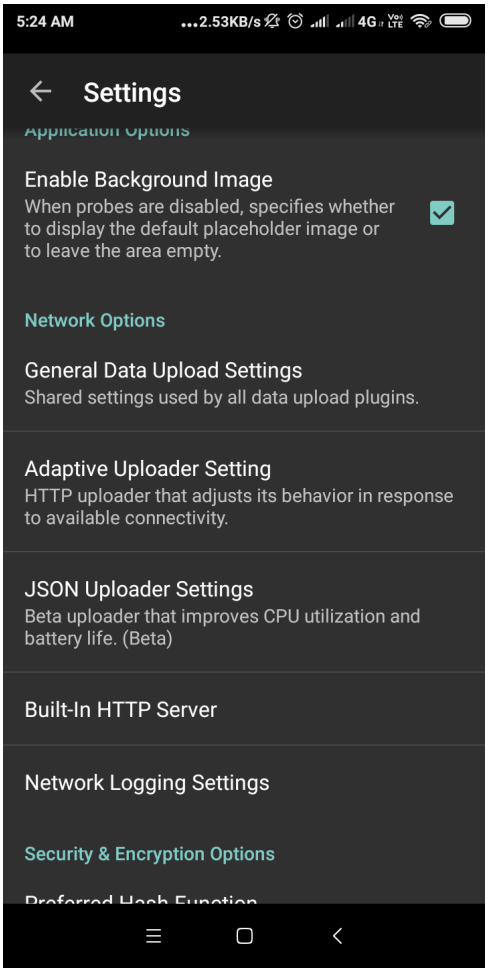
The following screen is responsible for

Uploader settings

Configuration of interval

Type of data transfer medium

Enable/Disable Security

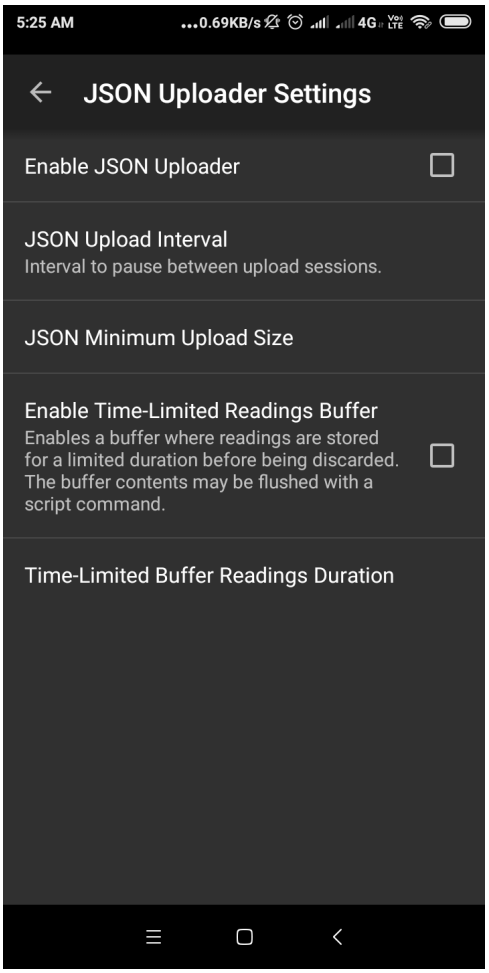


(a) Settings

The Following screen is responsible for transfer the data in JSON format . Below are some criteria while sending the data

Frequency This indicates , at what interval app will send the data

Upload size We are providing the flexibility to the user to limit the amount of data size to get transferred

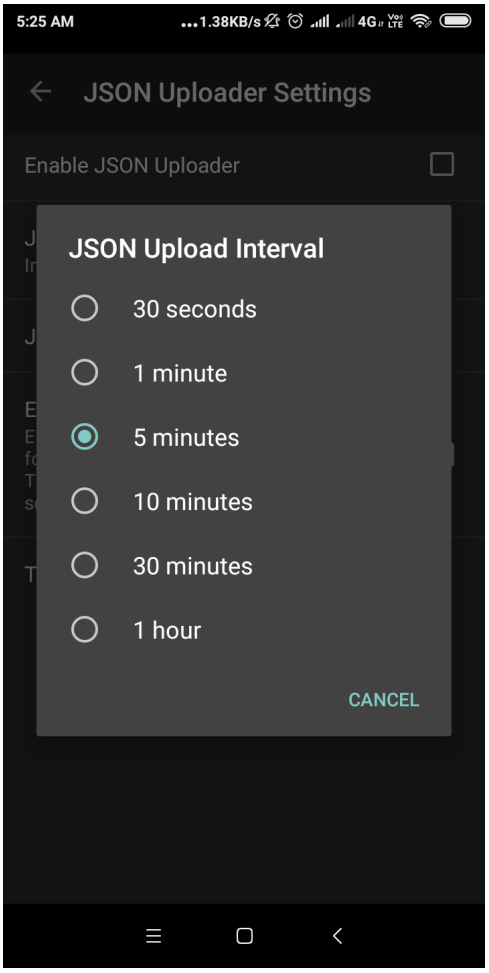


(a) Json Uploader Settings

Frequency This indicates , at what interval app will send the data

Selection From the given options, user can select the frequency of sending the data to server.

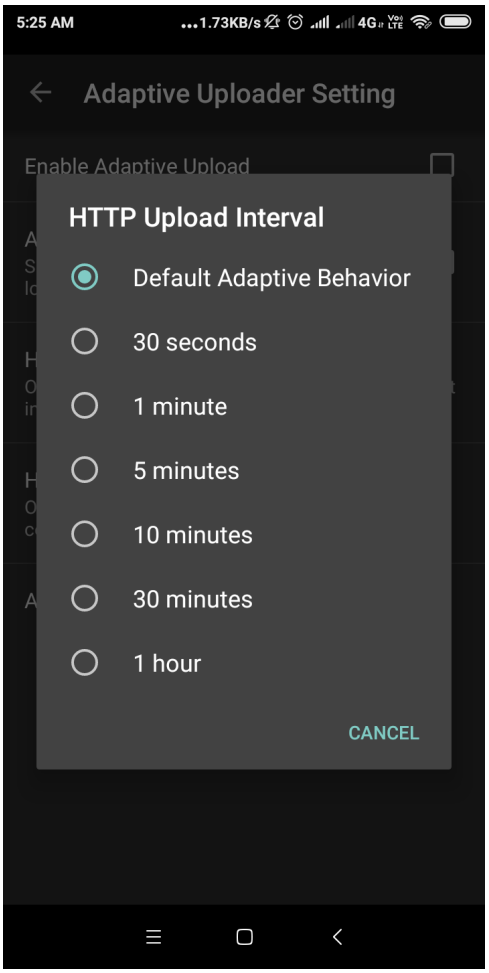
By default 5 min is considered, as per implementation.



(a) Options to select the frequency

Upload interval via Http Selection of default adaptive behaviour indicates, that the probes will be send only if user send the data explicitly by clicking on **Upload data** on right corner of Home Screen.

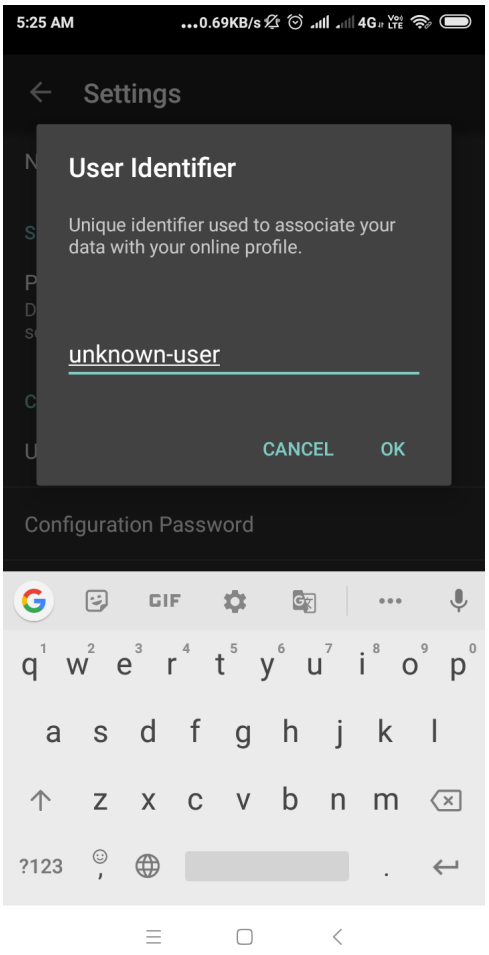
Selection From the given options, user can select the frequency of sending the probes.



(a) Upload data via Httpdata.

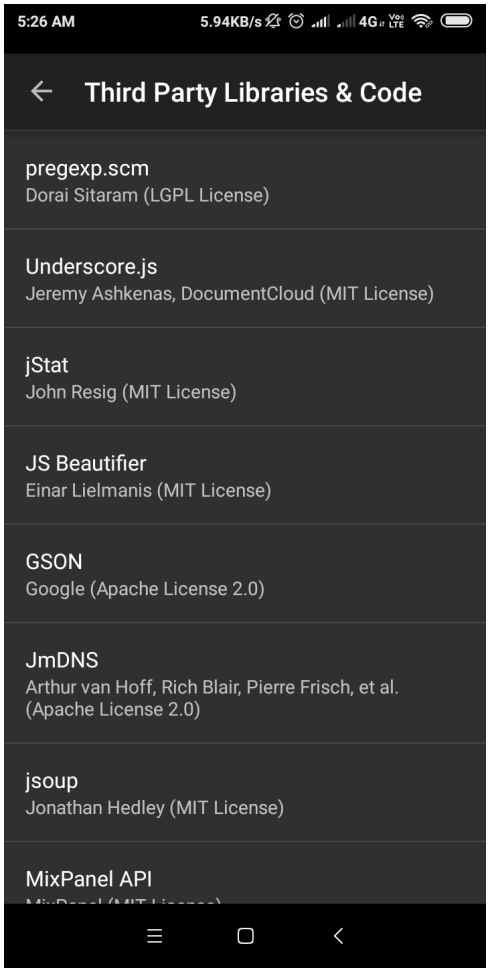
User Identifier Id For every user ,a hash is created in system for each corresponding userId provided.

Unique Key The hash generated above will be unique for each users.

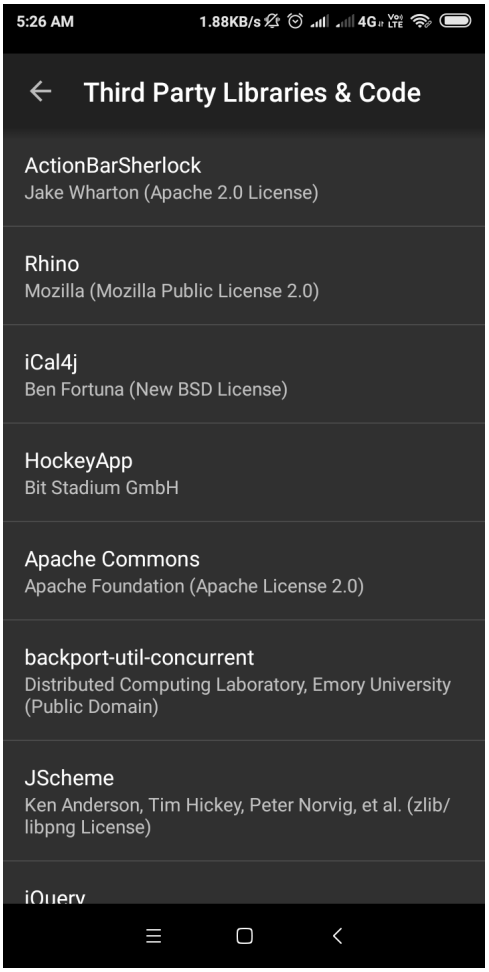


(a) User Identifier Screen.

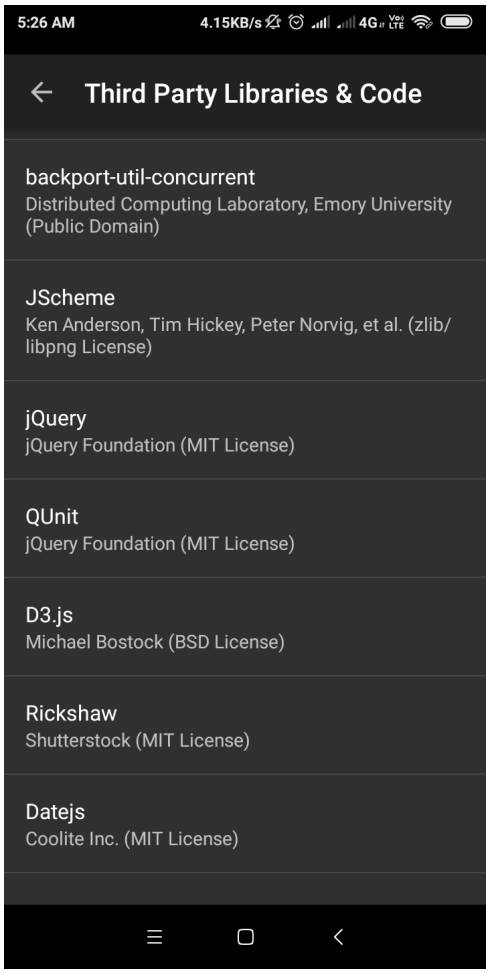
External Library App used the Third Party packages to do certain specific functions



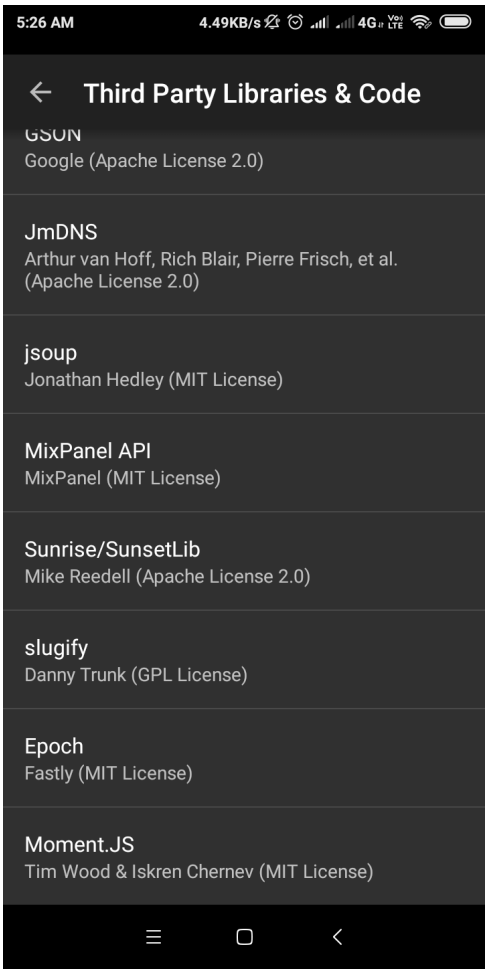
(a) Third Parties Packages.



(b) Third Parties Packages.



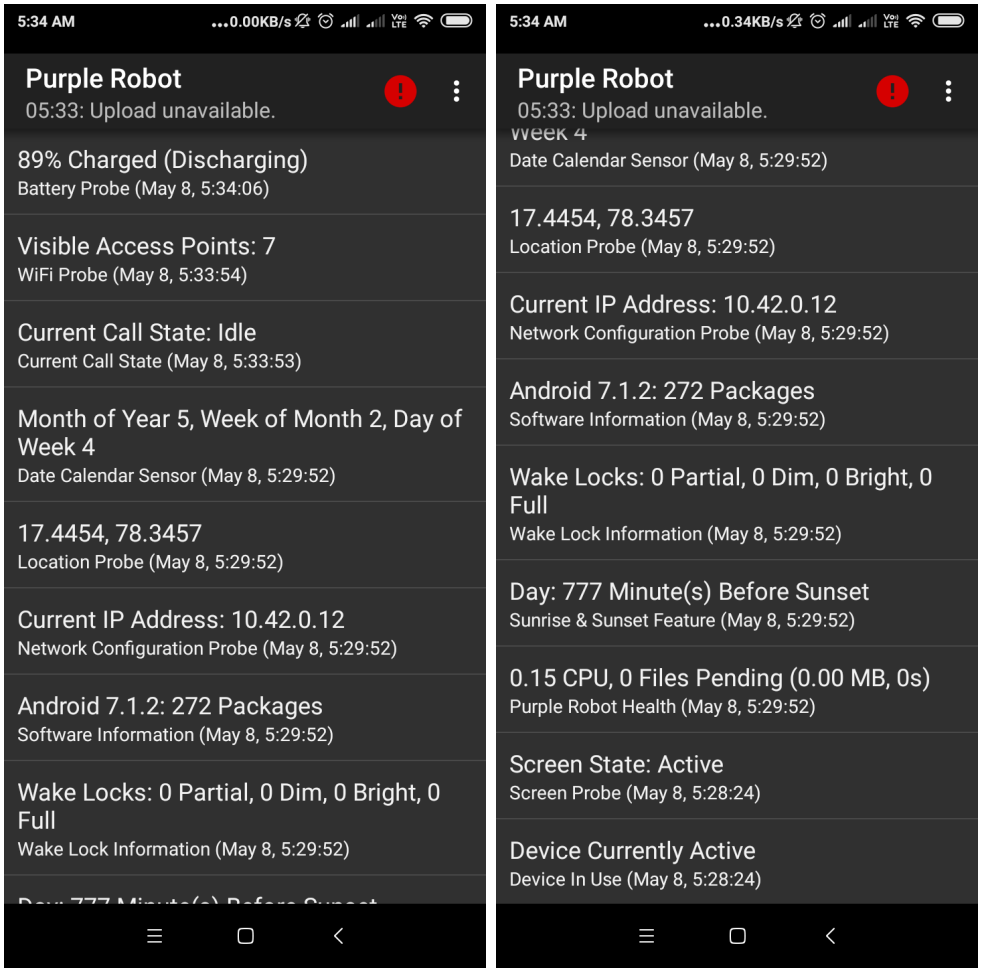
(a) Third Parties Packages.



(b) Third Parties Packages.

Different Probes Heart of Application which is showing all the probes which is enabled by user.

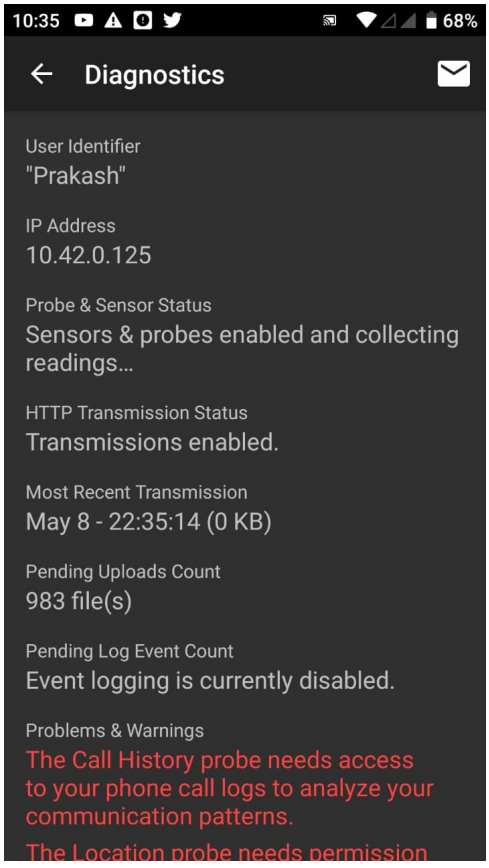
Visualization The data collected from the probes is used for Visualization and analysis .



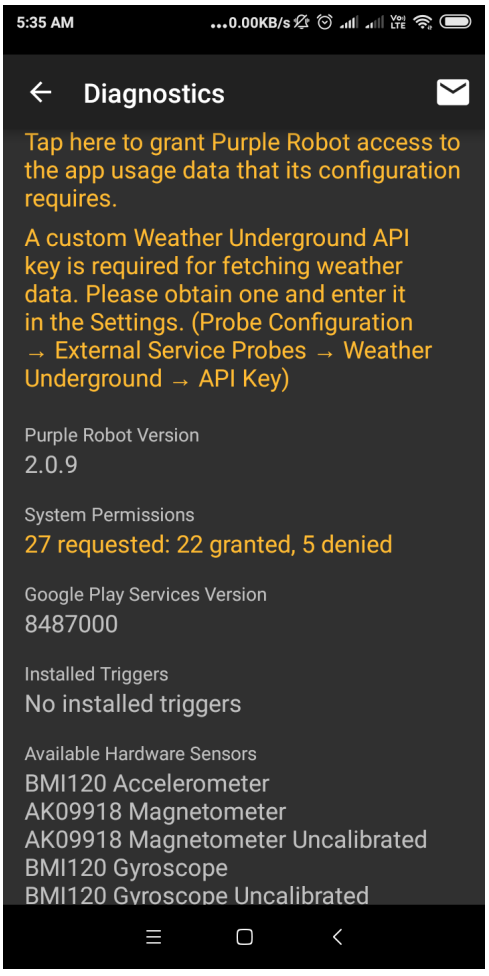
(a) Probe Screen **(b)** Probe Screen contd...

Diagnostics The screen is showing , the data which is collected by app. By looking at it, user can see his own data and accordingly ,user can enable/disable the respective probes

Notification Diagnostics can help in warning the user , in case of hazardous things occur in his device like temperature overheating , accessing the phone call logs etc.
Notification are highlighted in red.



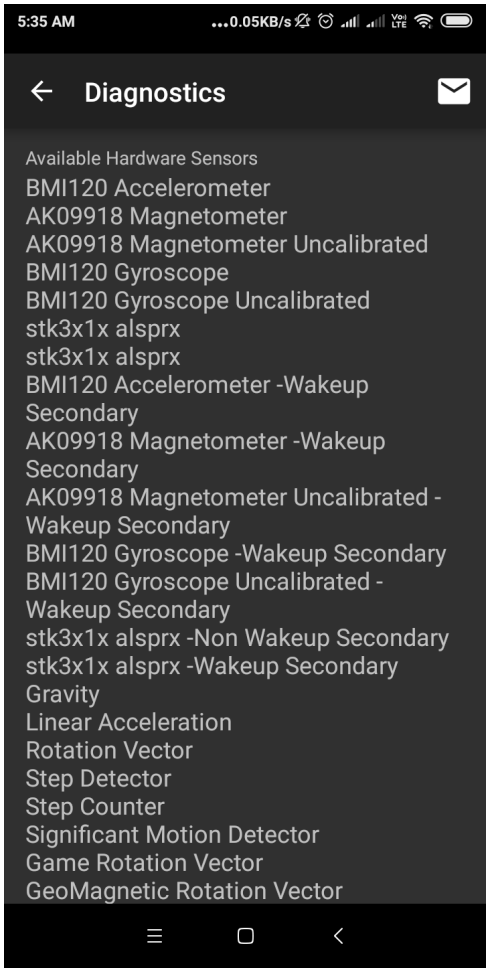
(a) Diagnostics Screen.



(b) Diagnostics Screen.

Available sensors Following Screen will show all the available sensors present on device of user.

Selection On the basis of above data, user can reached to respective user setting for enabling/Disabling the Probes.



(a) All Available sensors Screen.

0.3 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