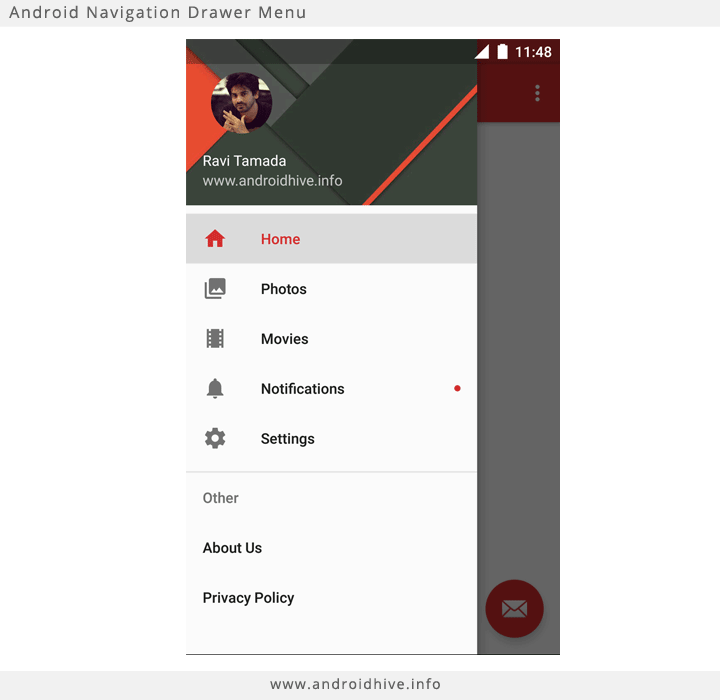
**Requirements Document for Firebase connection and App layout.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Description** |
| 9th Jan 19 | 1.0 | Chetan Prasad | Initial Version |
| 24th Jan 19 | 2.0 | Chetan Prasad | Updated Requirements for database. Check for requirements highlighted in Yellow. |

# **App requirements**

1. **App shall provide a splash screen.**
2. **App shall provide Firebase login. (I have already implemented this feature. I can provide the code for you)**
3. **The app shall check if internet connection is present.**
4. **The app shall access the GPS location of the device and shall be able to store in the database when required.**
5. **App shall have Navigation Drawer**

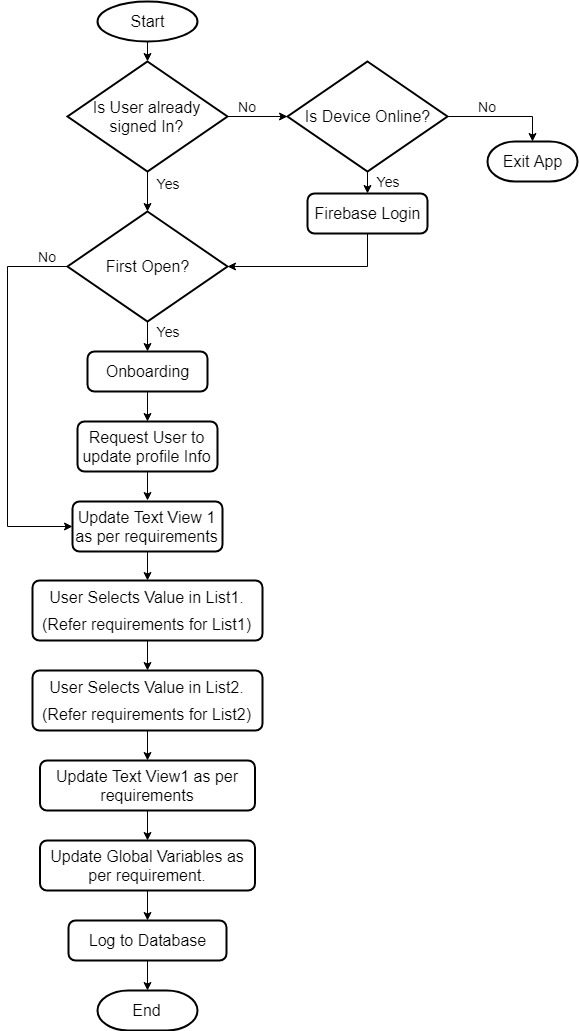


It shall have the following items.

* Name and email id (profile photo not required)
* OnBoarding
* Basic Electronics Tutorial
* Link to Website
* Link to Youtube channel, facebook and instagram
* About Us
* Privacy Policy

1. **Flowchart for the app**

The flowchart below gives the flow for Main Activity.



1. **The App shall provide onboarding screens to help user to get started.**

* The onboarding screens shall be customisable.
* Onboarding screens should be able to play Gif file format.
* Onboarding screen shall be displayed only the first time the app is opened.
* There shall be an option to view onboarding screens from the Navigation drawer.
* Reference for Onboarding screens: <https://blog.iamsuleiman.com/onboarding-android-viewpager-google-way/>

1. **Profile Information**

* The app shall provide an activity to update the profile information.
* The activity shall be accessible from the profile section of the Navigation Drawer.
* When the app opened for the first, Profile activity shall start after the onboarding screen is complete.
* The app shall never allow user to proceed if profile information is incomplete.
* Everytime the app is opened, a check shall be done to make sure all fields in profile information are updated.
* Profile Activity shall contain the following fields
  + Name (Automatically updated from Firebase login info)
  + Email (Automatically updated from Firebase login info)
  + College (Text box : Only Characters)
  + Semester (Text box : Only numbers from 1 to 8)

1. **The Main activity shall have the layout as shown below.**

List 1

List 2

Button 1

Button 2

Button 3

Button 4

Text View 1

* The activity shall be a scrollable activity so that when new buttons are added it shall be easy to view.
* The Button size shall remain consistent both in landscape and portrait mode.
* The Buttons color shall change on button press to “Blue”.
  1. **“List 1” Requirements**
* **“List 1”** shall be a drop down box.
* **“List 1”** shall read the values from the database. All the names in the array “Colleges” shall be shown in the drop down list. Refer “Database Design” section for the database structure.

|  |  |  |
| --- | --- | --- |
| **Collection** | **Document** | **Field** (Array) |
| Colleges | College\_List | Colleges |

* **“List 1”** shall work in the same way even if array size of Colleges change.
  1. **“List 2” Requirements**
* **“List 2”** shall be a drop down box.
* The values displayed in “**List 2**” depends on the value entered in “**List 1**”,
* If “**Demo**” has been selected in “**List 1**”, the values for “**List 2**” shall be retrieved from the array **demo\_modules** as shown below.

|  |  |  |
| --- | --- | --- |
| **Collection** | **Document** | **Field** (Array) |
| Colleges | Demo | demo\_modules |

* If “**Individual**” has been selected in “**List 1**”, the app shall search the stored email id (as obtained from Firebase sign in) in the database in the path below.

|  |  |  |
| --- | --- | --- |
| **Collection** | **Document** | **Field** (map) |
| Colleges | Individual | Email id as obtained from Firebase |

If email id is found in the database, the app shall get the module name from the path below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Collection** | **Document** | **Field** (map) | **Field** (String) |
| Colleges | Individual | Email id as obtained from Firebase | module |

If email id is not found in the database, the app shall display the message “**You have not Subscribed**” in the “**Text View 1**” box.

* If any value other than “**Individual**” and “**Demo**” is selected in “**List 1**”, the values for “**List 2**” shall be retrieved from the array **college\_modules** as shown below.

|  |  |  |
| --- | --- | --- |
| **Collection** | **Document** | **Field** (Array) |
| Colleges | Value as in “List 1” | college\_modules |

Ex: If “List 1” = RNSIT then the app shall retrieve values from Colleges -> RNSIT -> college\_modules

* **“List 2”** shall work in the same way even if array size of college\_modules change.
  1. **“Text View 1” Requirements**
* Before the selection of **“List 1”** and **“List 2”, “Text View 1”** shall display the text “**Select the Correct Module to Connect**”.
* After the selection of **“List 1”** and **“List 2”, “Text View 1”** shall display the text “**App will connect to Module xxx**” where XXX is the value selected in “List 2”.
* If the value selected in “**List 1**” is “**Individual**” and the email id was **not found** in the database, the app shall display the message “**You have not Subscribed**” in the “**Text View 1**” box.
* If the value selected in “**List 1**” is “**Individual**” and the email id was **found** in the database, the app shall read the validity date from the database path as shown below and display in “**Text View 1**” box “ Subscription valid untill : XXX”. where XXX is the date read from database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Collection** | **Document** | **Field** (map) | **Field** (timestamp) |
| Colleges | Individual | Email id as obtained from Firebase | validity |

* The background color of “Text View1” shall change to RED while displaying error messages.
  1. **Button Requirements**
* All 4 buttons shall open an empty activity.
* All 4 buttons shall be Image and Text buttons.
* User shall be able to open the activities in Button1 and Button2 only if proper selection has been done in “List1” and “List2”.
* If User presses “Button1” or “Button2” **without** correct selection of “List1” and “List2”, there shall be a Toast message “**Select Correct Module**” and also the same message shall be displayed in “Text View1” box.
* When User Presses “Button1” or “Button2” **with** correct selection of “List1” and “List2”, the corresponding activity shall create database entry (Refer Writing to Database requirements).

1. **Global Variables Requirement**

* The app shall have a class called “Global\_Var” which extends Application. This helps in accessing the members of this class globally in different activities.
* The “Global\_Var” class shall have the following members
  + String **mUsername; -> Got from Firebase Login**
  + String **mEmailId**; **-> Got from Firebase Login**
  + String **college\_name**; -> from profile info
  + String **semester**; -> from profile info
  + String **category;**  -> value of List1
  + **int current\_session**=0; -> no need to update
  + **int** **Error\_Code**; -> no need to update
  + **String** **Connection\_status ->** Check Requirements
  + **GPS Location** (dont know what datatype) -> Last best Location
  + String **MAC\_Address**; -> From Database based on value of List 2
  + String **Module**\_**Name**; -> From Database based on value of List 2
  + **int INA1\_Calibration**; -> From Database based on value of List 2
  + **int INA2\_Calibration**; -> From Database based on value of List 2
* The class shall provide get and set methods for all the members.
* The variable “Connection\_status” shall have the following values.
  + “**App Opened**” when Main\_Activity is created.
  + “**Connected**” after Button1 or Button2 is pressed.
  + “**Disconnected**” after it comes back to Main\_Activity from Button1 or Button2 activity.

1. **Writing to Database**

* The app shall provide a class to write to database.
* Member function of this class shall write to database.
* In addition to the Global variables, this class shall also update Timestamp while writing to database.
* Database write location shall be as shown below

|  |  |  |
| --- | --- | --- |
| **Collection** | **Document** | **Field** |
| Usage\_Info | Value in global var “Category” | Key value pairs as given below. |

* The following variables shall be part of the key value pairs
  + **mUsername**
  + **mEmailId**
  + **College\_name**
  + **Semester**
  + **Connection\_status**
  + **Location**
  + **Module**\_**Name**
  + **TimeStamp** -> update with current time