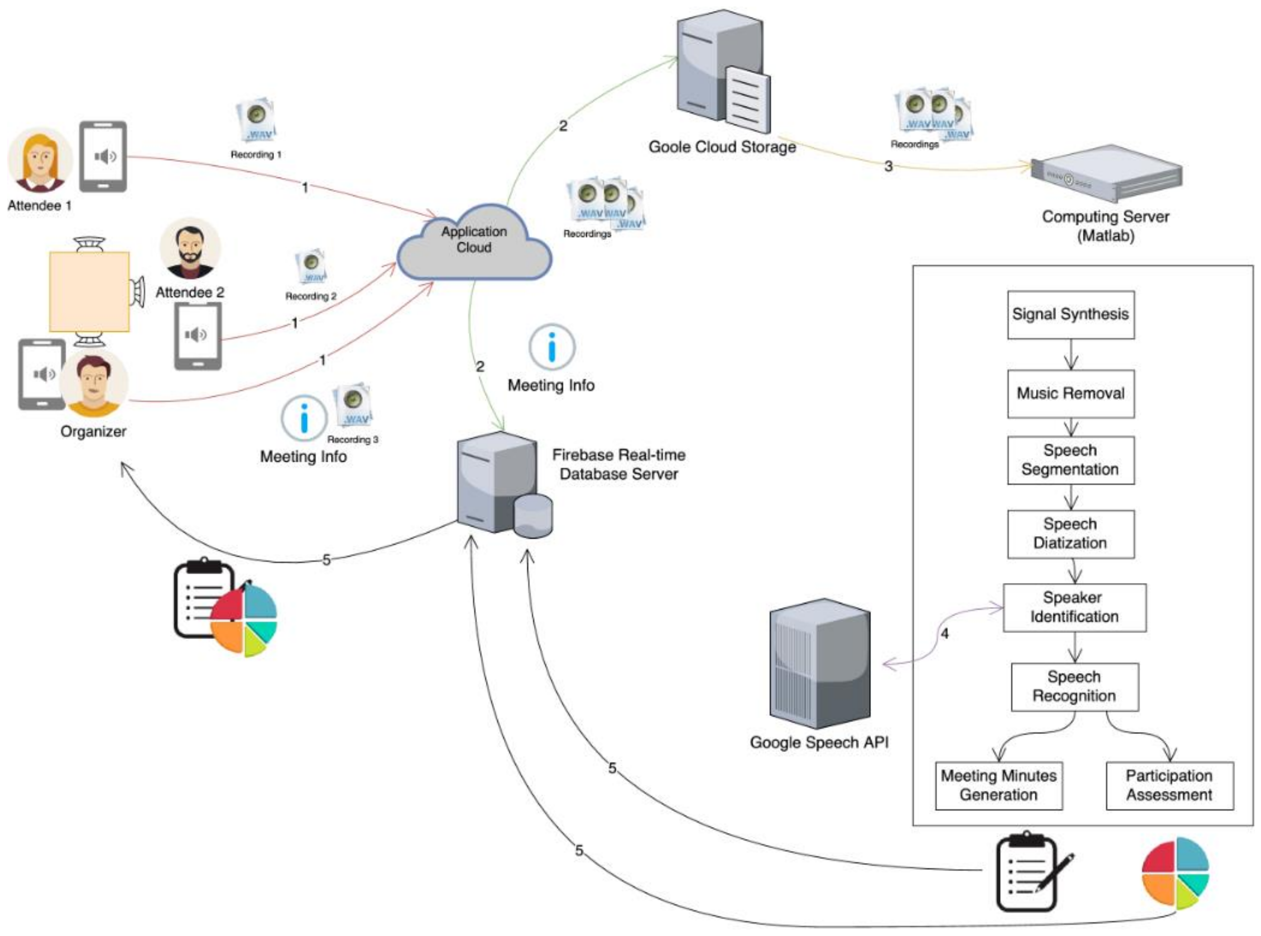
Design Tools Catalogue

- **System Overview Diagram:**

We recommend using draw.io for developing your system overview diagram. Your diagram should be descriptive and well labelled.

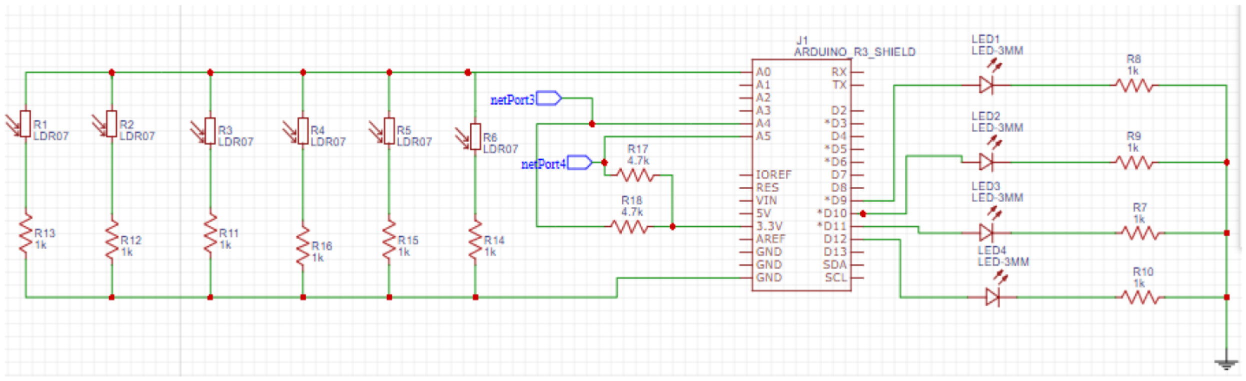


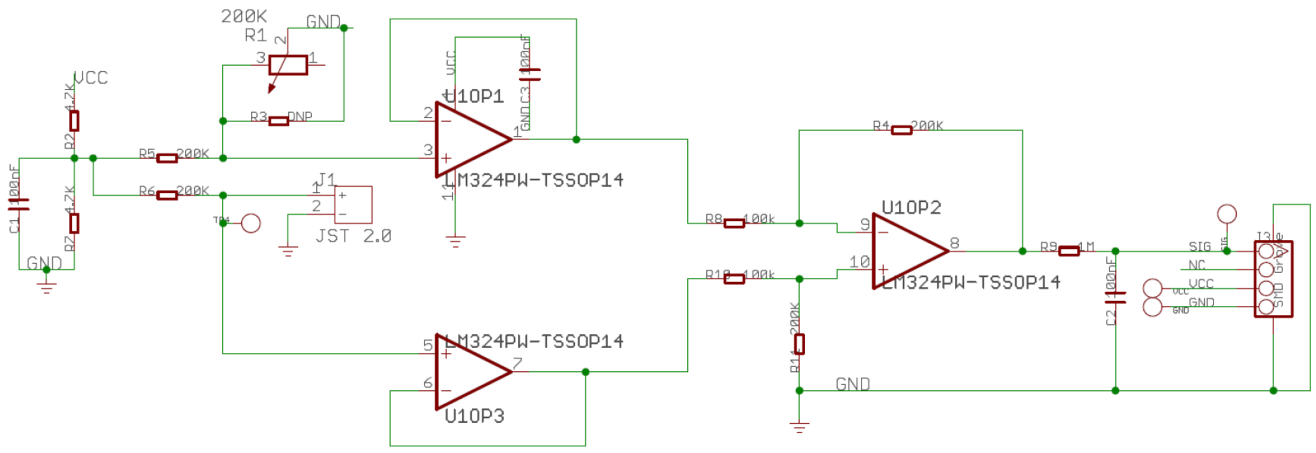
Our proposed system uses speech segmentation for participation assessment and feedback. The system overview is shown above. The process begins with the meeting organizer or teacher creating or recalling an existing pre-scheduled meeting or class. This happens using digital assistants, web portals, email clients, or smart phones. A meeting identification number is auto-generated by hashing meeting and user data and sharing it visually or verbally with participants. Simultaneously and not necessarily in sync, the smart phones of participants begin recording the meeting audio signals. We record multiple audio for two purposes. First, we align the recording and merge them to produce an enhanced signal, which improves the quality of the subsequent steps. Second, we capitalize on the relative high availability of idle smart phones. The system works with a single recording, but depending on the distance and volume of user speech signals and location of recording smart phones, the performance in terms of participation assessment accuracy improves with additional recordings.

Meeting meta data, Meeting Inf, in the figure is simultaneously kept in a non-relational real-time database. The uploaded recordings are retrieved by a processing server. Audio analysis begins by synthesizing an audio-enhanced signal. It moves to remove background music if it exists and proceeds to segment the speech signal. Speech diarization is performed and the result is used to segment the audio signal into pieces to be assigned to speakers. Low-level audio features collected from the segmented audio signal and the ones from training data are used to identify the speaker. The digital identity of the speakers is maintained and collected by the application itself as part of the setup stage as well (e.g., profile information). Segmented speech is used for participation assessment as the percentage of time the audio is assigned to a particular speaker, which is used to build a pie-chart. The meeting organizer creates the mapping between the identity of the speaker and the real-world identity (e.g., speaker 1 is John Doe). Speech identification technology can also be used, but it requires collecting speech data from meeting participants in the setup stage to train a classifier. Cloud-based speech recognition services can also be used on the segmented audio to convert the speech to text for automating the process of taking meeting minutes.

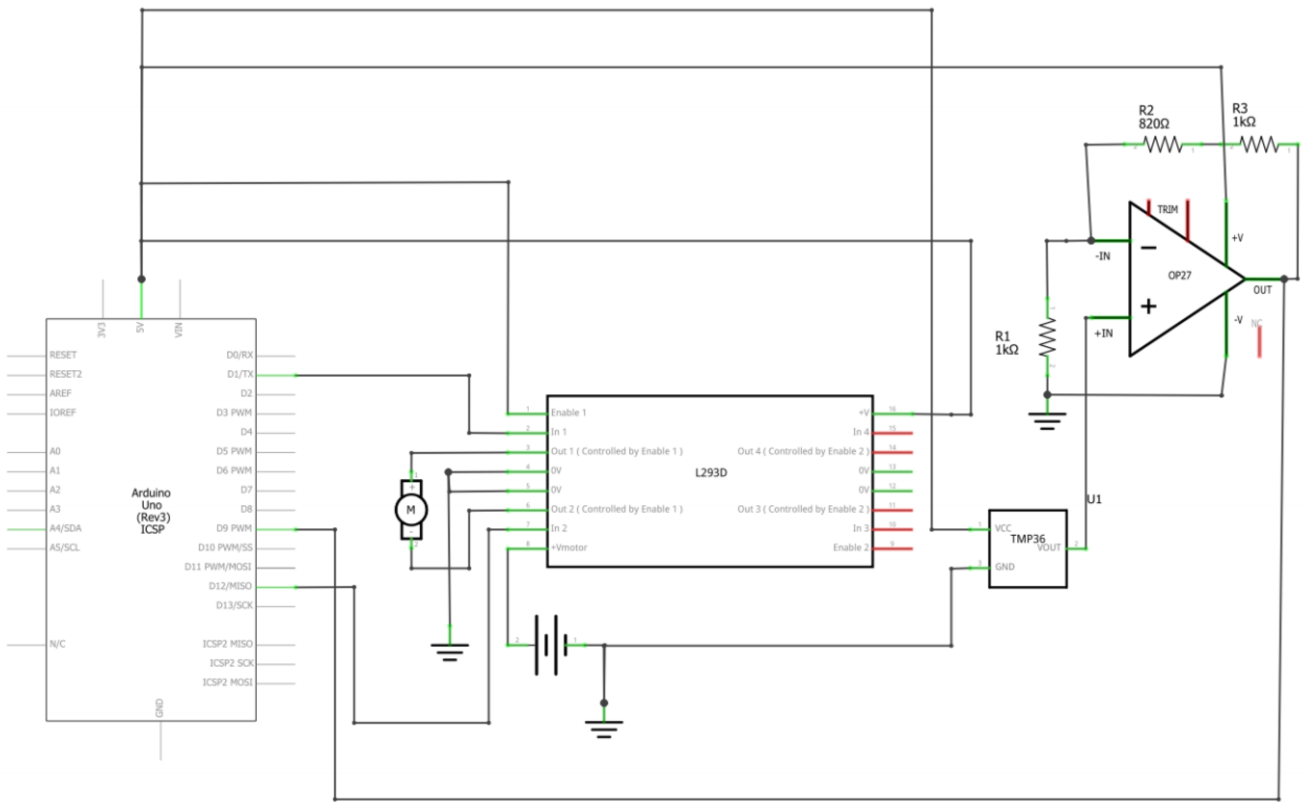
- **Hardware Schematic(s):**

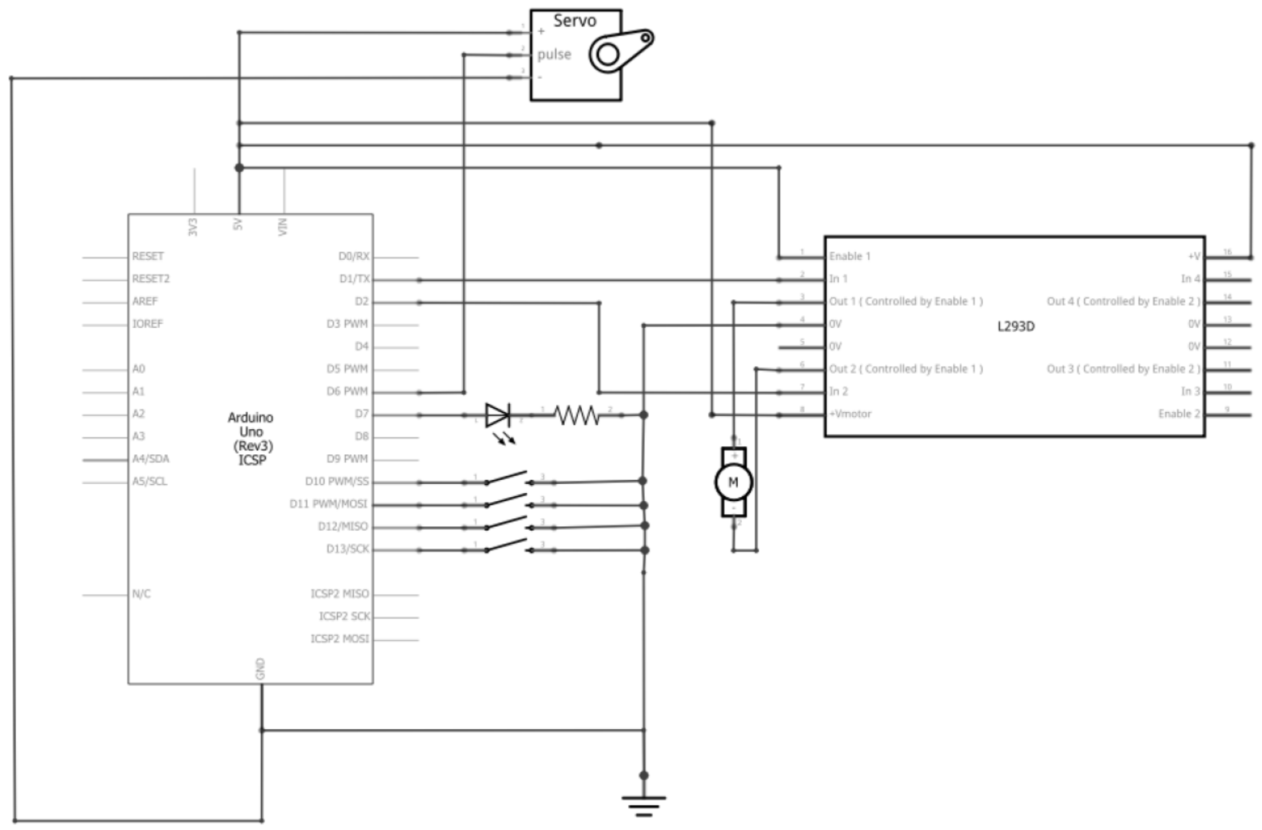
The hardware schematics can be designed using any software/tool such as fritzing, Multisim, Tinkercad, etc.

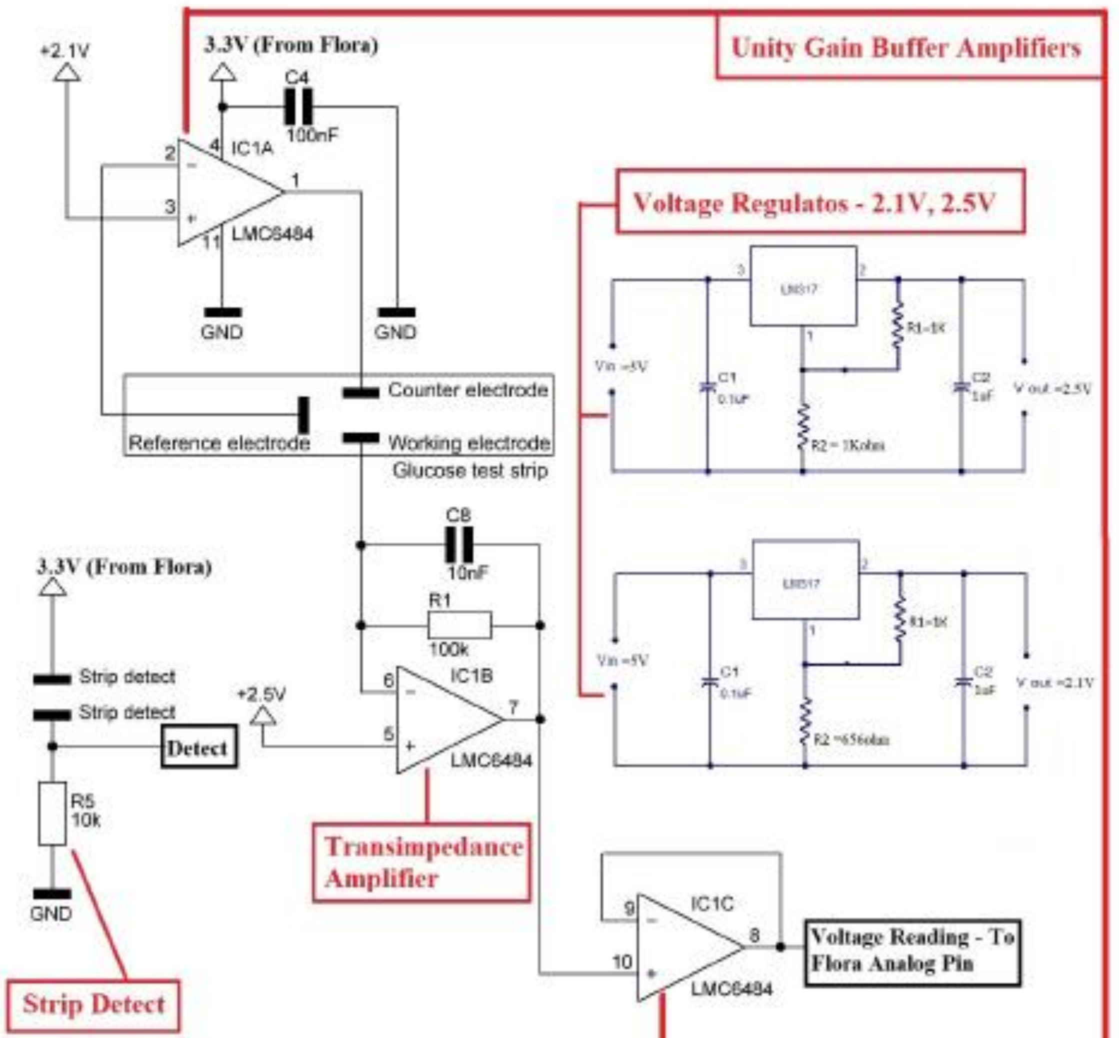




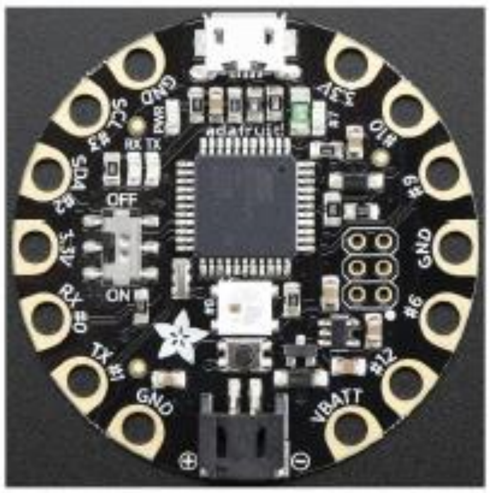
Circuit diagram of the Grove GSR sensor



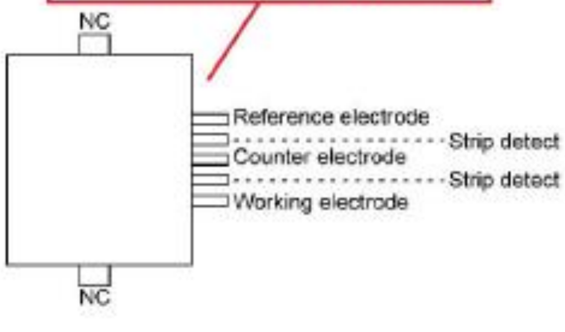

Servo
+
pulse
-
3V3
5V
VIN
RESET
RESET2
AREF
IOREF
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
Arduino
Uno
(Rev3)
ICSP
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
ICSP2 MISO
ICSP2 SCK
ICSP2 MOSI
GND
M
L293D
Enable 1
In 1
Out 1 (Controlled by Enable 1)
0V
0V
Out 2 (Controlled by Enable 1)
In 2
+Vmotor
+V
In 4
Out 4 (Controlled by Enable 2)
0V
0V
Out 3 (Controlled by Enable 2)
In 3
Enable 2



Fora V3 Microcontroller

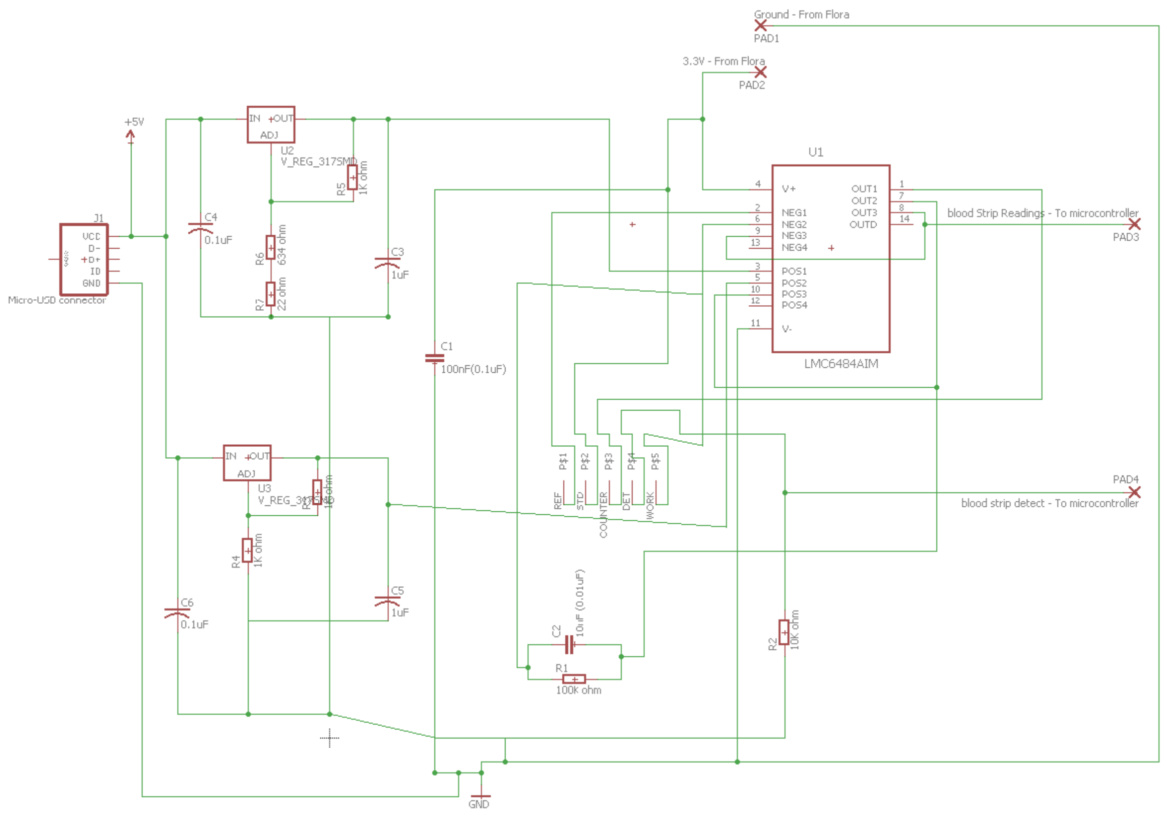


OneTouch Ultra Blood Strip Connector

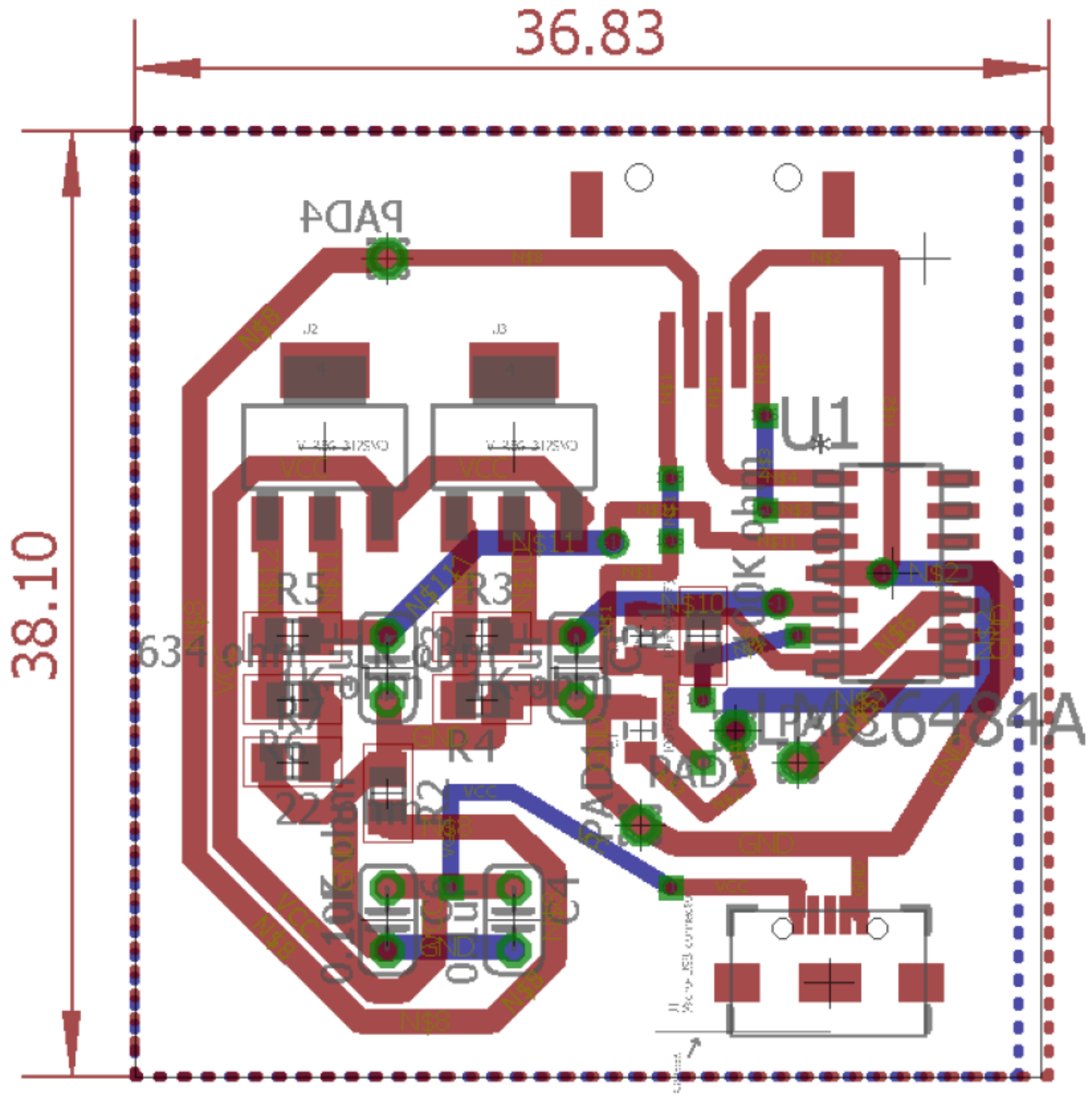


- **PCB:**

The below figure shows the overall schematic design of our final PCB. We connected all of the components based on the hardware schematic. We used the block that we designed for the blood strip connector to connect the working electrode, counter electrode, and the reference electrode. We are powering the amplifier IC using separate pins that will be connected to the Flora 3.3V regulated voltage using pin headers. To power up the 2.5V and 2.1V regulators we are using a micro-USB connector which is shown on the left side of the figure. The amplifier IC contains individual amplifiers that are used in our design such as the unity gain amplifiers and the transimpedence amplifier located on the right side of our design. In addition, PAD1 will be connected to the ground of Flora V3 microcontroller, while PAD2 will be connected to the 3.3V pin from the Flora V3 microcontroller. PAD3 is the blood readings which is the output of the transimpedance amplifier, and PAD4 is the blood strip detect pin which is coming from the blood strip detect circuit. PAD3 will be connected to D6 (A7) pin on the Flora micro-controller and PAD4 will be connected to the D12 of the Flora micro-controller.



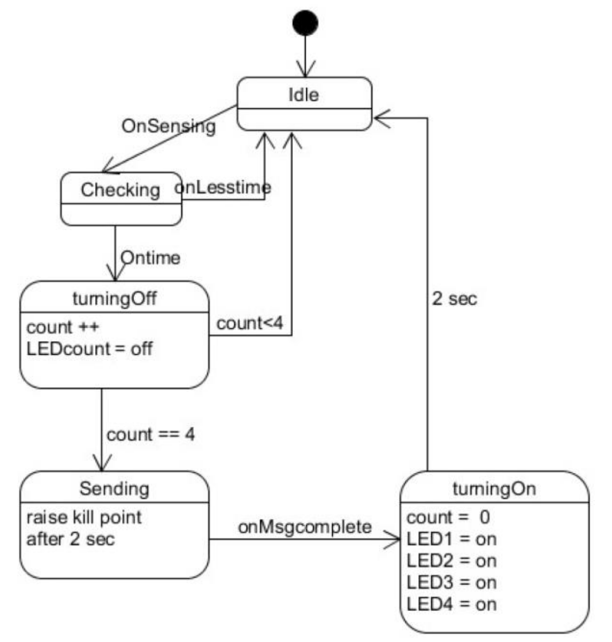
After finalizing the schematic design, we moved to the second stage of designing the PCB, which is the routing design. Due to our small PCB design, we need to design the routs manually because the auto router uses a big space to establish all of the routs without any overlaps between wires. The below figure shows the PCB board design. The micro-USB connector is located on the bottom-right corner and the OneTouch Ultra blood strip connector is located on the top-right corner. The amplifier IC is located under the blood strip connector and the two voltage regulators (2.5V and 2.1V) are located on the top-left corner of the PCB board design. The amplifier IC has 14 pins. Pin number 4 and pin number 11 are connected to the ground and 3.3V. The first three pins belong to the first amplifier, pins 5, 6, and 7 belong to the second amplifier, and pins 8, 9, and 10 belong to the third amplifier. The fourth amplifier is not used as we only require three amplifiers, so its pins are not connected to anything as shown below. The overall dimensions of the PCB are 35.83mm X 38.1mm (3.683cm X 3.81cm) which is fairly small and fits on the wrist to look like a watch perfectly. The following table shows all of the PCB components names and values. It shows each component used in the PCB and its corresponding actual component name and its value.



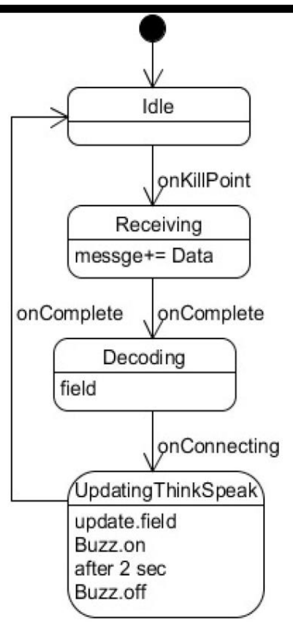
PCB Components Names and Values

PCB Component Name	**Actual Component Used - Value - Name**
U1	LMC6484AIM/NOPB Amplifier IC
U2	Voltage Regulator [LM317AEMP/NOPB]
U3	Voltage Regulator [LM317AEMP/NOPB]
J1	Micro USB connector
C1	100nF (0.1uF) Multilayer Ceramic Capacitor
C2	10nF (0.01uF) Multilayer Ceramic Capacitor
C3	1uF – Electrolyte Capacitor
C4	0.1uF – Electrolyte Capacitor
C5	1uF – Electrolyte Capacitor
C6	0.1uF – Electrolyte Capacitor
R1	100k ohms
R2	10k ohms
R3	1k ohms
R4	1k ohms
R5	1k ohms
R6	634 ohms
R7	22 ohms
PAD1	Ground From Flora Microcontroller
PAD2	3.3V From Flora Microcontroller
PAD3	Blood Readings – Flora Analog Input D6 (A7)
PAD4	Blood strip detect – Flora Digital Input (D12)

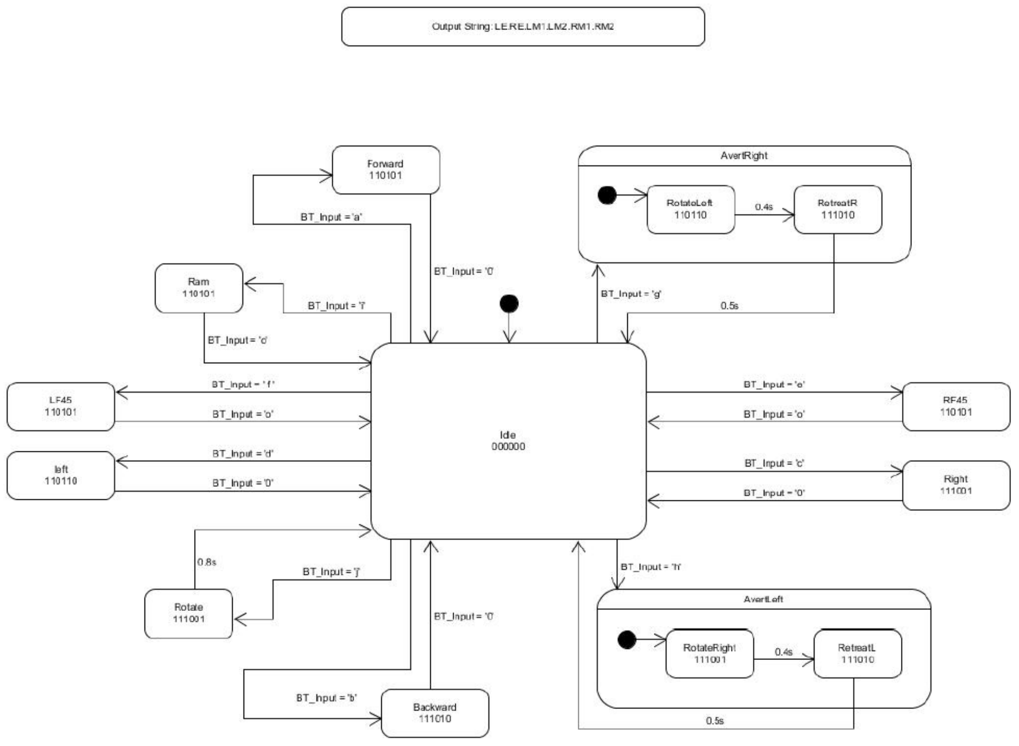
- **UML State chart(s) for embedded systems:**



Master's FSM.



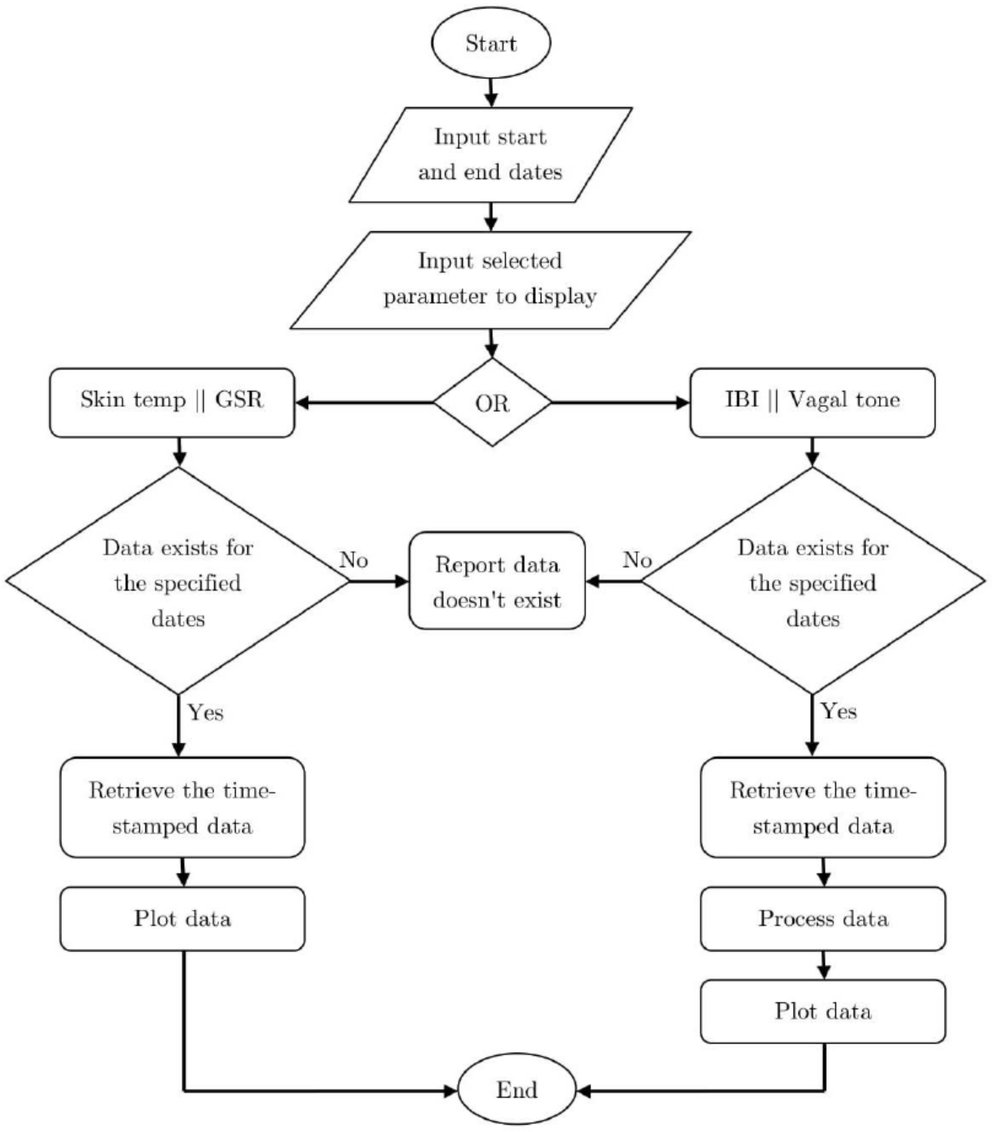
Slave's FSM.



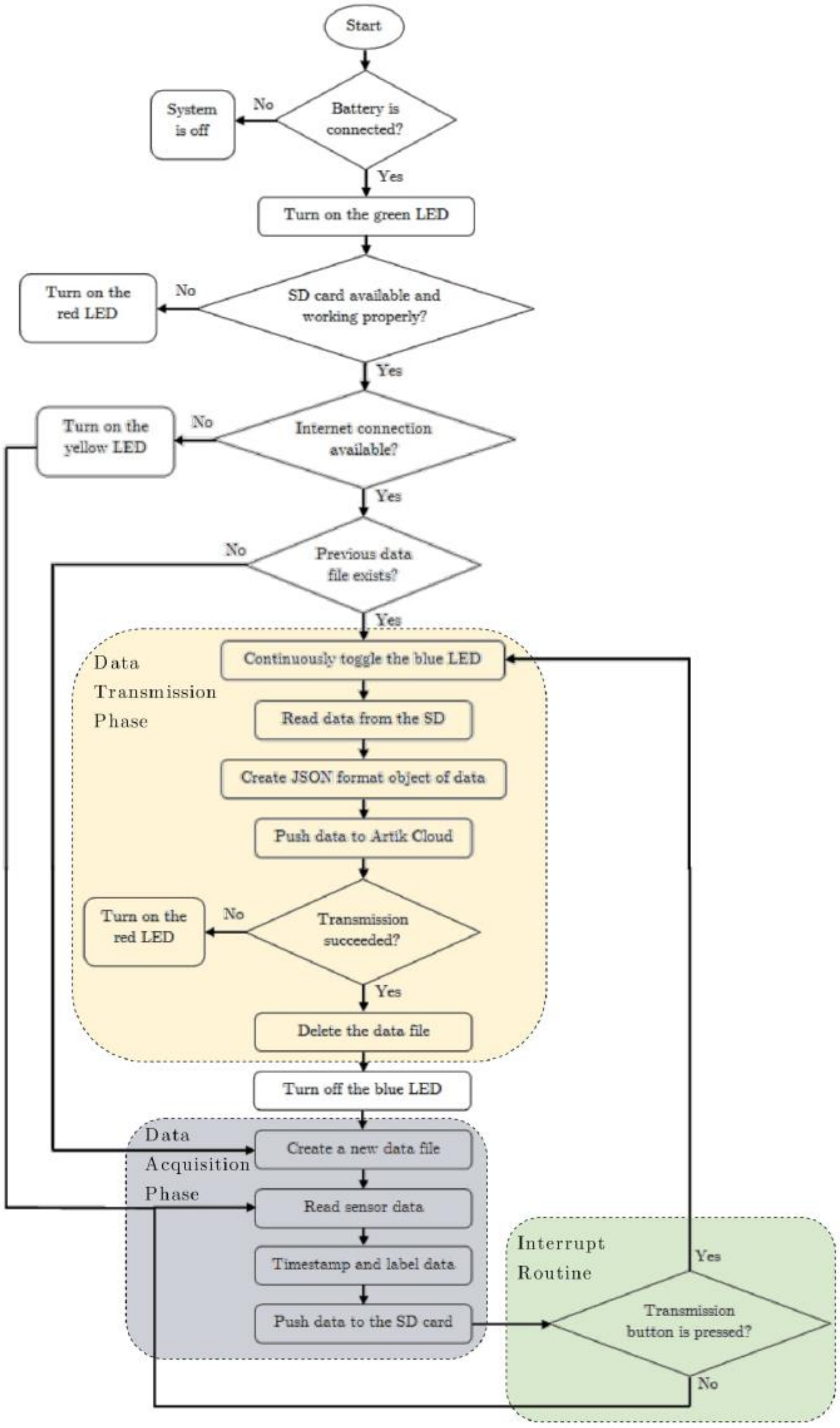
The State Chart of the Robot System.

For more information, please visit:
https://www.tutorialspoint.com/uml/uml_statechart_diagram.htm

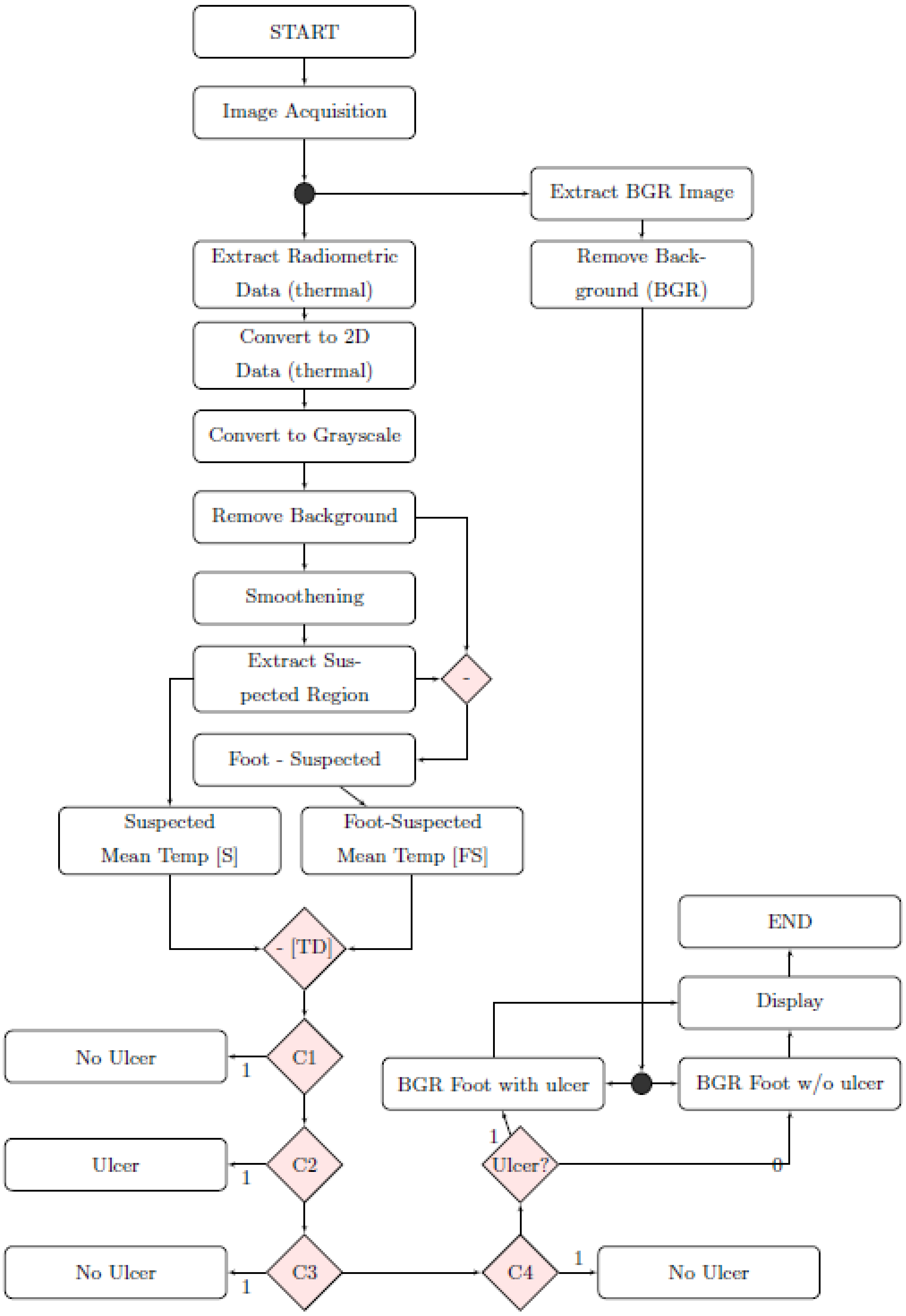
- **Flowcharts(s) for all codes and algorithms:**


Start
Input start and end dates
Input selected parameter to display
OR
Skin temp || GSR
IBI || Vagal tone
Data exists for the specified dates
No
Report data doesn't exist
No
Data exists for the specified dates
Yes
Yes
Retrieve the time-stamped data
Retrieve the time-stamped data
Plot data
Process data
Plot data
End

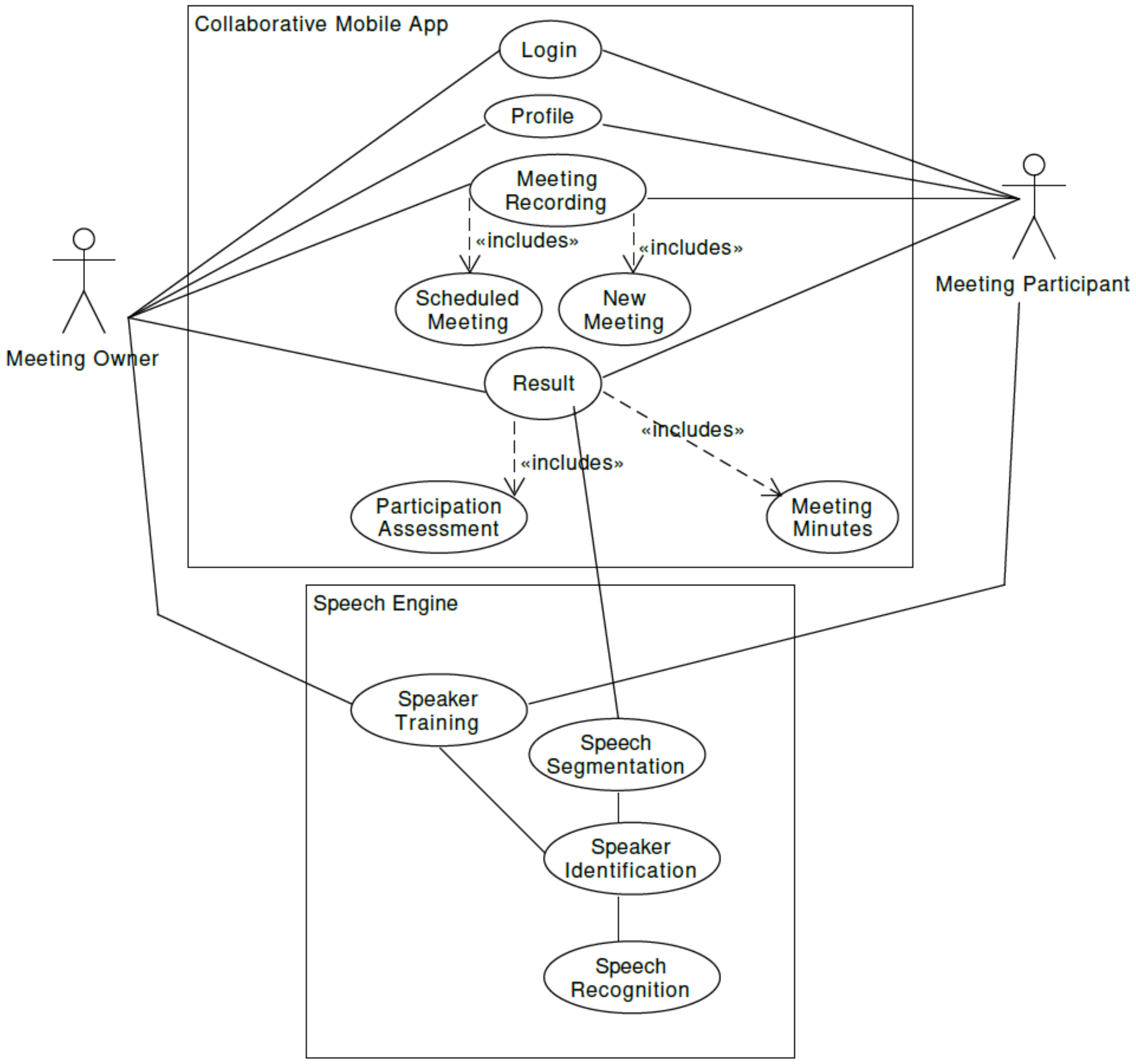
Data retrieval and data representation algorithm



Data acquisition and data transmission algorithm


START
Image Acquisition
Extract BGR Image
Extract Radiometric Data (thermal)
Remove Back-ground (BGR)
Convert to 2D Data (thermal)
Convert to Grayscale
Remove Background
Smoothening
Extract Sus-pected Region
-
Foot - Suspected
Suspected Mean Temp [S]
Foot-Suspected Mean Temp [FS]
- [TD]
END
Display
C1
No Ulcer
1
BGR Foot with ulcer
BGR Foot w/o ulcer
C2
Ulcer
1
1
Ulcer?
0
C3
No Ulcer
1
C4
1
No Ulcer

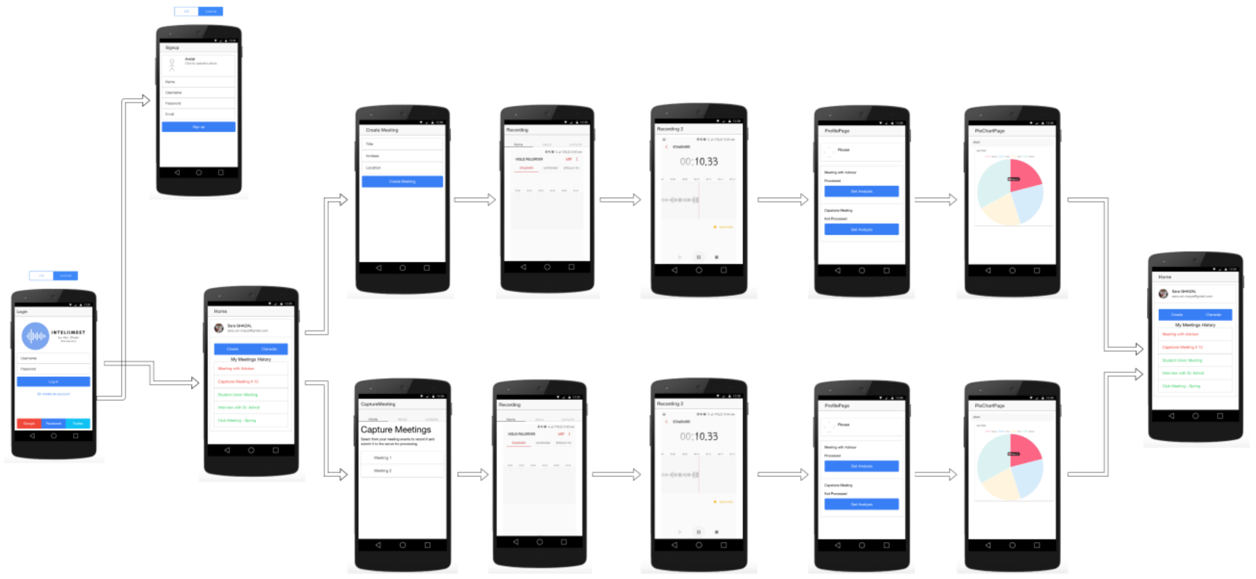
- **UML Use Case Diagrams:**

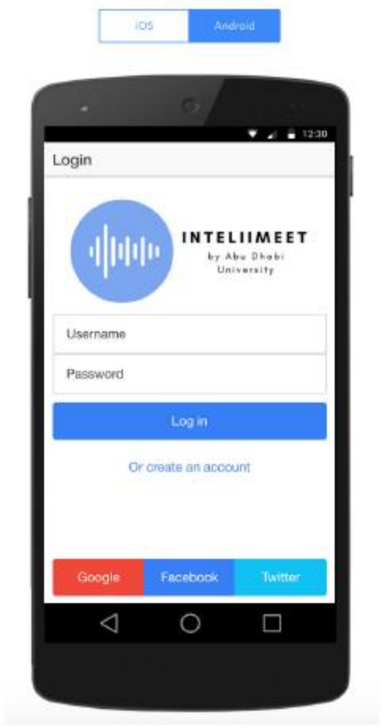


We designed the use cases of our app using a UML use case diagram, which can be seen in the figure. We have two actors in the system, the meeting owner and the meeting participants. The meeting owner is the person who will schedule the meeting. They view the meeting in their calendar or receive the meeting ID from the meeting owner. They then participate by recording the meeting and uploading the recording to the database and the Cloud storage. There is a main system for IntelliMeet: the Collaborative Mobile App, which has processes for Login, Profile, and Meeting Recordings whether Scheduled Meeting or New Meeting and Results including Participation Assessment and Meeting Minutes. The actors of the system also interact with the speech engine by providing speech data for speaker training. This data is used as training data in the speaker identification. The result process in the Collaborative Mobile App interacts with the Speech Engine starting with the Speech Segmentation and ending with the Speech Recognition.

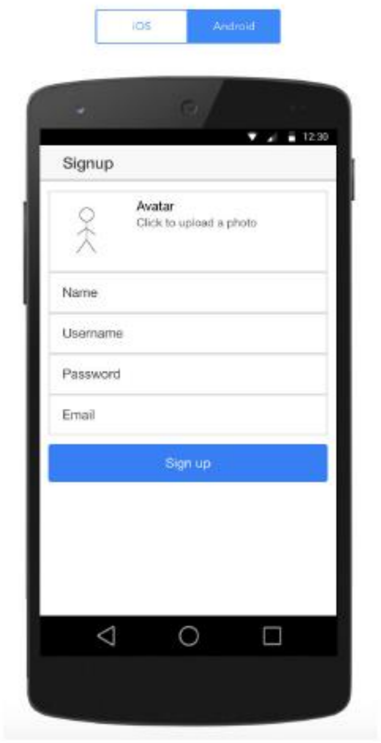
- **User journey and view designs for all mobile applications:**

To design the app, we developed an interactive prototype for it based on client requirements and shared it with the client. We then received the feedback from the client and used it to improve the prototype until it was accepted. The focus on our design was simplicity and reducing the number of clicks. We also made sure the design is visually consistent. One important feature is using both new meeting information and existing meeting information from the mobile phone calendar. This allows the user to schedule meetings using any mobile app, web app, desktop app, or even digital assistant like Siri. These meetings will be retrieved by the user to choose which meeting to record. The user journey and the design of each screen of the app are shown in the figures below respectively, where the prototype can be accessed from this link: https://creator.ionic.io/share/d340b428dad2.

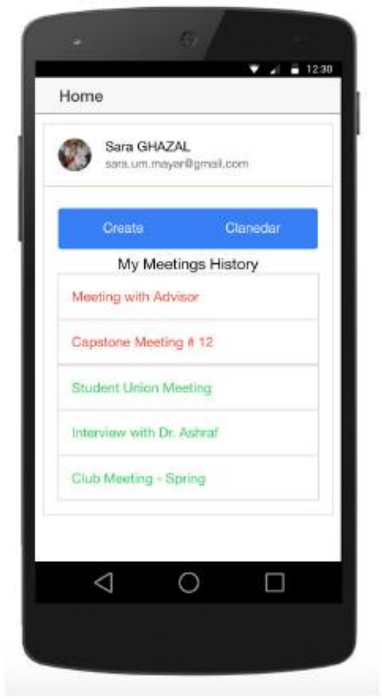




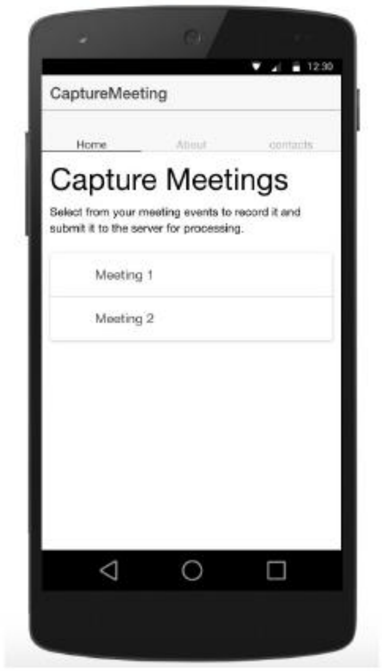
(a) Login View



(b) Sign Up View

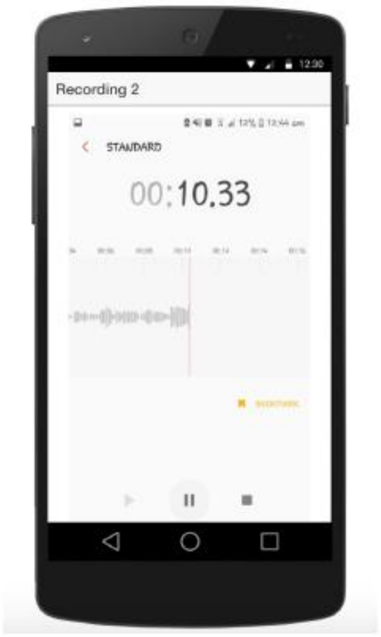


(c) Profile View

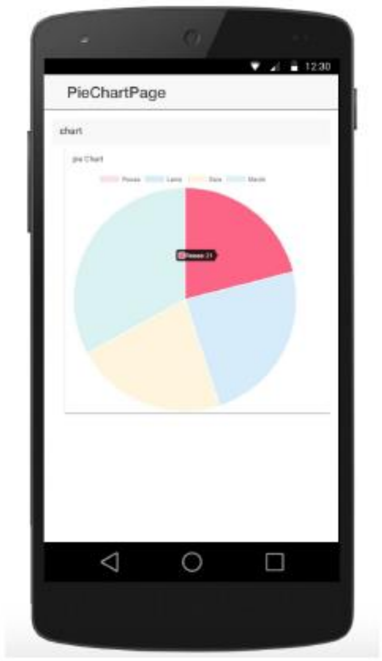


Select from your meeting events to record it and submit it to the server for processing.

(d) Calendar View

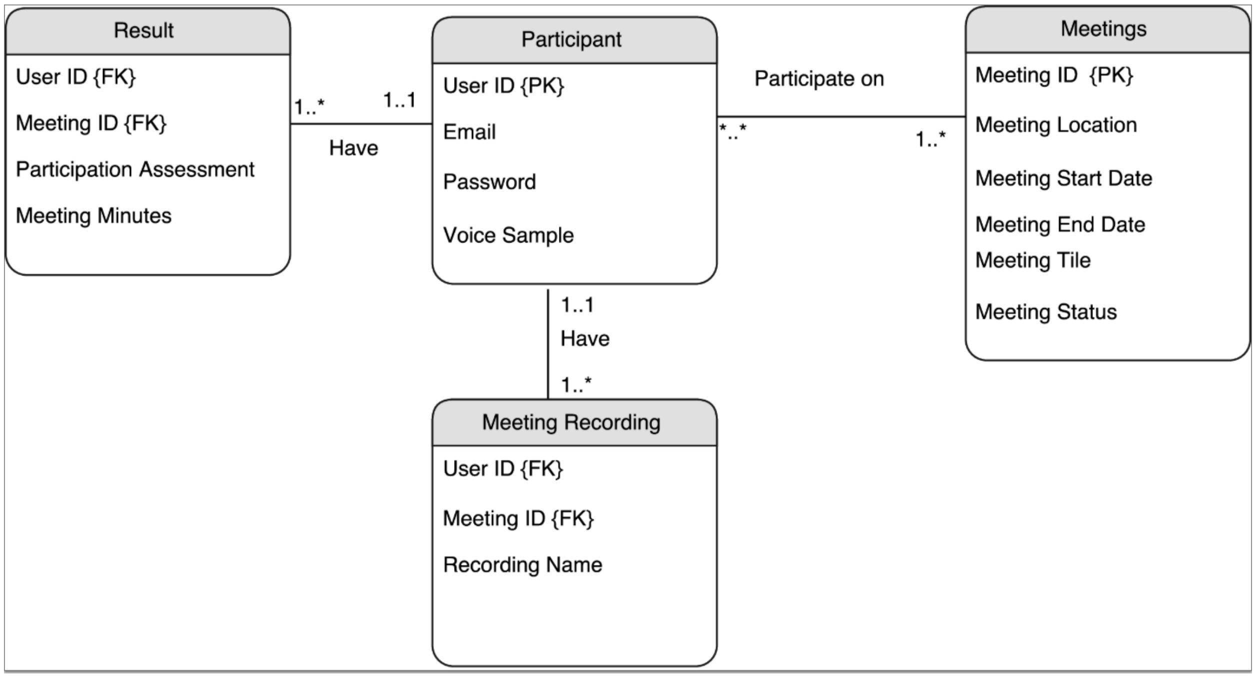


(e) Recording View



(f) Analysis View

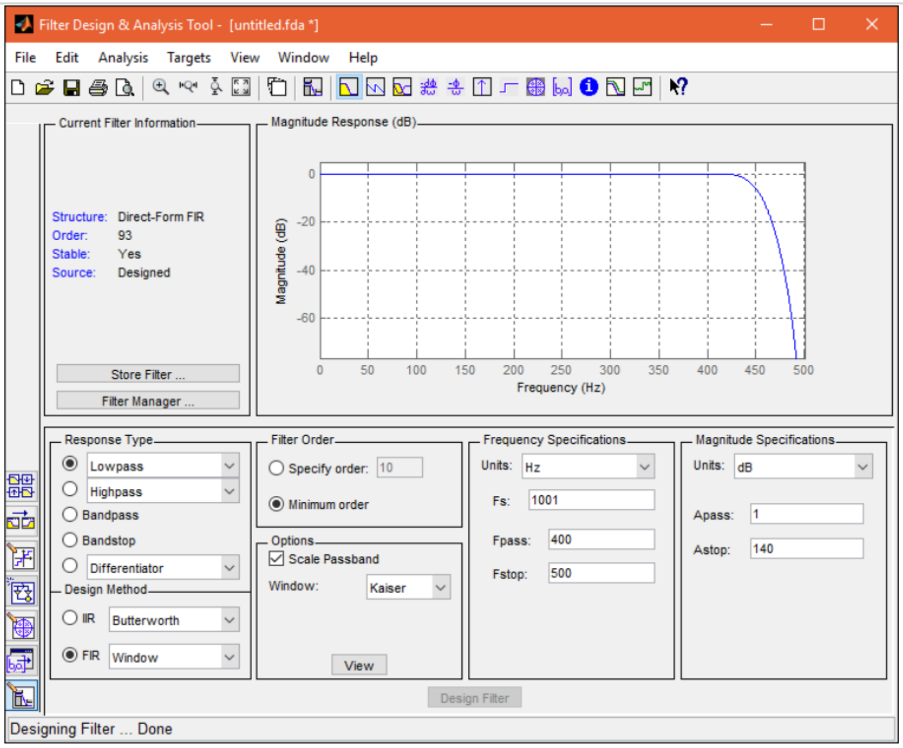
- **Entity Relationship diagrams [for projects with databases]:**



The Entity Relationship of our system consists of four parts as in the figure. Firstly, we have the list of participants where each one has to login to the application with her/his email and password to be authenticated. However, each participant will be having a user id on Firebase, where his/her information and recordings will be saved under that user id. On the other hand, each participant has to provide a sample of his voice as a training data that will be used later for speaker identification process. Moreover, each meeting will be having a meeting ID, title, status, and start and end time. Furthermore, the user can participate in more than one meeting in the list. Furthermore, each participant needs to record the meeting so that s/he will receive the participation percentages for each attendee plus the meeting minutes.

- **Filter design artifacts if filters are involved:**

1. To use the serial plotter on Arduino, we are required to have the latest Arduino software. Therefore, the first step we did was downloading Arduino Software IDE version 1.8.3.
2. We have generated the low pass filter coefficients by inserting the following specifications in Matlab fdatool as shown below:
    - 1 dB ripple in the pass-band
    - -140 dB minimum Attenuation in the stopband
    - Band pass frequency wp = 400 Hz
    - Band stop frequency ws = 500 Hz
    - Sampling frequency fs = 1001 Hz



Current Filter Information

Structure: Direct-Form FIR
Order: 93
Stable: Yes
Source: Designed

Response Type: [✓ Lowpass] Highpass Bandpass Bandstop Differentiator

Design Method: IIR Butterworth [✓ FIR Window]

Filter Order: Specify order: 10 [✓ Minimum order]

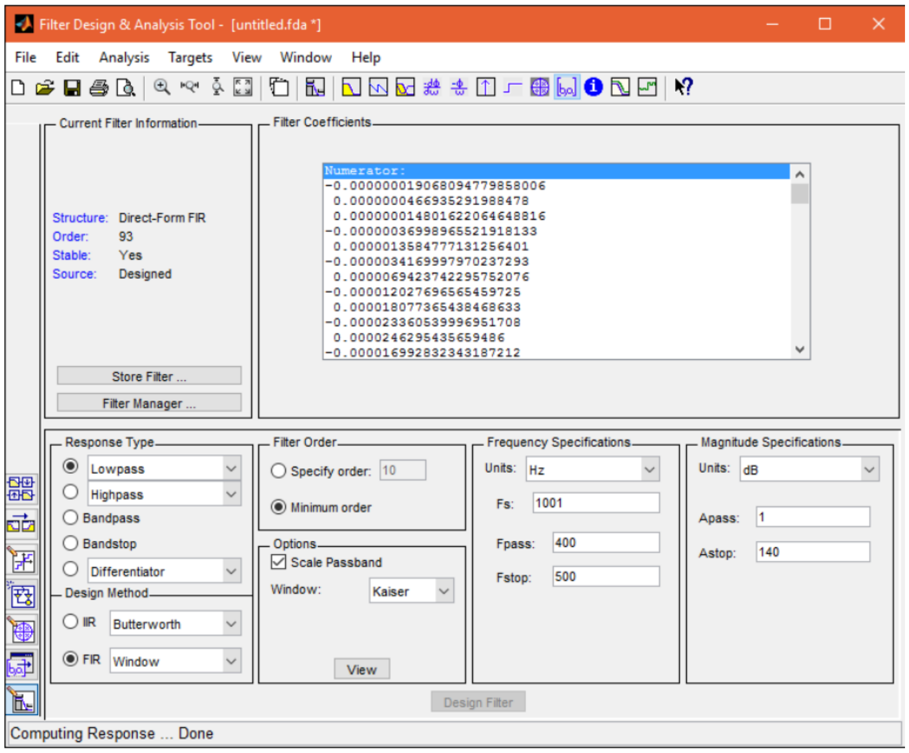
Options: [✓ Scale Passband]
Window: Kaiser

Frequency Specifications
Units: Hz
Fs: 1001
Fpass: 400
Fstop: 500

Magnitude Specifications
Units: dB
Apass: 1
Astop: 140

Designing Filter ... Done

Specifications of the Low Pass Filter.


Filter Design & Analysis Tool - [untitled.fda *]
File Edit Analysis Targets View Window Help
Store Filter ...
Filter Manager ...
View
Design Filter

Current Filter Information

Structure: Direct-Form FIR
Order: 93
Stable: Yes
Source: Designed

Filter Coefficients

```
Numerator:
-0.000000019068094779858006
 0.0000000466935291988478
 0.000000014801622064648816
-0.00000036998965521918133
 0.0000013584777131256401
-0.0000034169997970237293
 0.0000069423742295752076
-0.000012027696565459725
 0.000018077365438468633
-0.000023360539996951708
 0.000024629543565948 6
-0.000016992832343187212
```

Response Type: [✓ Lowpass] Highpass Bandpass Bandstop Differentiator

Design Method: IIR Butterworth [✓ FIR Window]

Filter Order: Specify order: 10 [✓ Minimum order]

Options: [✓ Scale Passband]
Window: Kaiser

Frequency Specifications
Units: Hz
Fs: 1001
Fpass: 400
Fstop: 500

Magnitude Specifications
Units: dB
Apass: 1
Astop: 140

Computing Response ... Done

Low Pass Filter Coefficients Generation.

3. The minimum attenuation value inputted in the fdatool is 140 dB instead of -140 dB because fdatool does not accept a negative As value.
4. The designing method chosen for the low pass filter LPF is window method; specifically, kaiser window option is selected.
5. Afterwards, the coefficients have been exported in the form of coefficient File (ASCII) in decimal format and inserted in the Arduino Code.

- **Design calculations:**

1.1 Determining Battery Size

1. Assuming All loads are A.C. and Inverter Efficiency of 90%.

$$TotalLoad = 28784 * 0.9 = 32000Wh$$

2. System Voltage is chosen according to Figure 1.

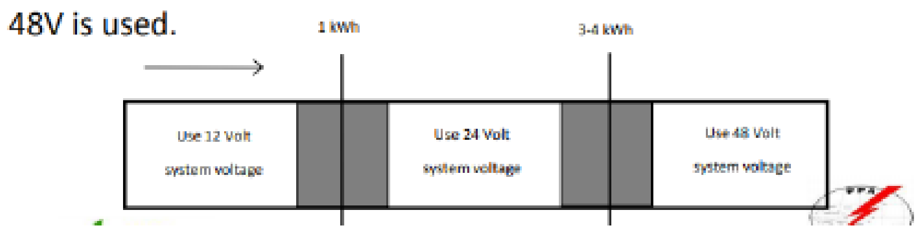

48V is used.
1 kWh
3-4 kWh
Use 12 Volt system voltage
Use 24 Volt system voltage
Use 48 Volt system voltage

Figure 1: Determining System Voltage

3. The daily Ah demand will be:

$$Ah = \frac{32000}{48} = 666.7Ah$$

4. Adjusted battery capacity assuming 70% DOD and 2 days of autonomy.

$$Battery\ Capacity = \frac{666.7 * 2}{0.7} = 1904.86Ah$$

1.2 Sizing PV Arrays Standard Controller

1. Daily Array Requirement assuming 90% battery efficiency and 6 PSH:

$$PV\,Array\ Output = \frac{666.7}{0.9 * 6} = 123.5Ah$$

2. Adjusted Array output current with 10% over-size:

$$Adjusted\ Array\ Output = 123.5 * 1.1 = 135.85A$$

3. De-rated current of the module assuming 95% output tolerance and 95% dirt de-rating:

$$Derated\ Current = 4.75 * 0.95 * 0.95 = 4.29A$$

4. Number of modules in a series string with 48V system voltage and 12V nominal module voltage:

$$No.Modules\ In\ Series = 48/12 = 4$$

5. Number of strings in parallel:

$$No.\ String = \frac{135.85}{4.29} = 31$$

6. Therefore, the number of modules in the array will be:

$$No.Modules\ In\ Array = 4 * 31 = 124$$

7. Peak Rating of Array:

$$Peak\ Rating = 80 * 124 = 9920W_p$$

2.1 Yearly Yield & Performance Ratio

The average yearly energy yield can be determined as follows:

$$E_{sys} = P_{array\_STC} * f_{temp} * f_{mm} * f_{dirt} * H_{tilt} * \eta_{pv\_inv} * \eta_{inv} * \eta_{inv-sb}$$

1. Assuming 5% manufacturer tolerance, 5% dirt losses, 15% temperature losses, PSH of 5, 24 panels, 3% DC losses, 96% inverter efficiency and 1% AC losses, then the final energy yield per year is:

$$Energy\,Yield = 160 * 24 * 0.85 * 0.95 * 0.95 * 5 * 0.97 * 0.96 * 0.99 * 365 = 4956kWh/year$$

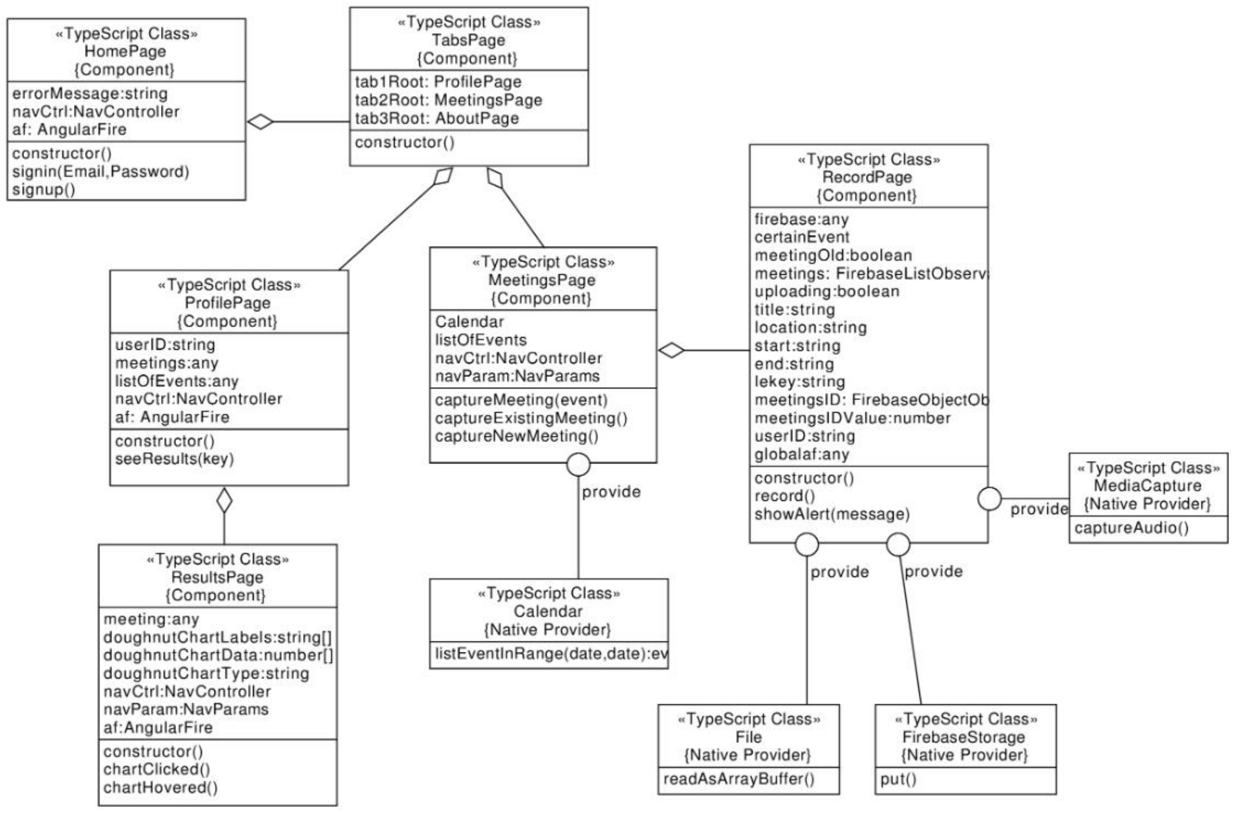
2. Accordingly, the specific energy yield is:

$$SY = \frac{4956}{3.84} = 1290.625kWh\ per\ kW_p$$

3. And the performance Ratio is:

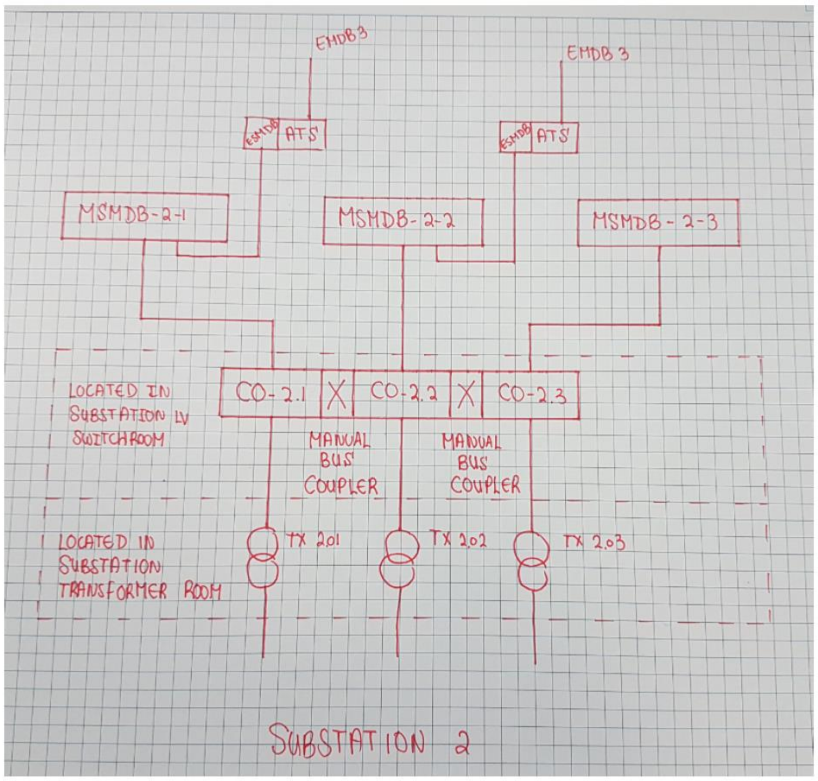
$$PR = \frac{4956}{5 * 365 * 3.84} = 0.71$$

- **UML Class Diagrams for all classes:**

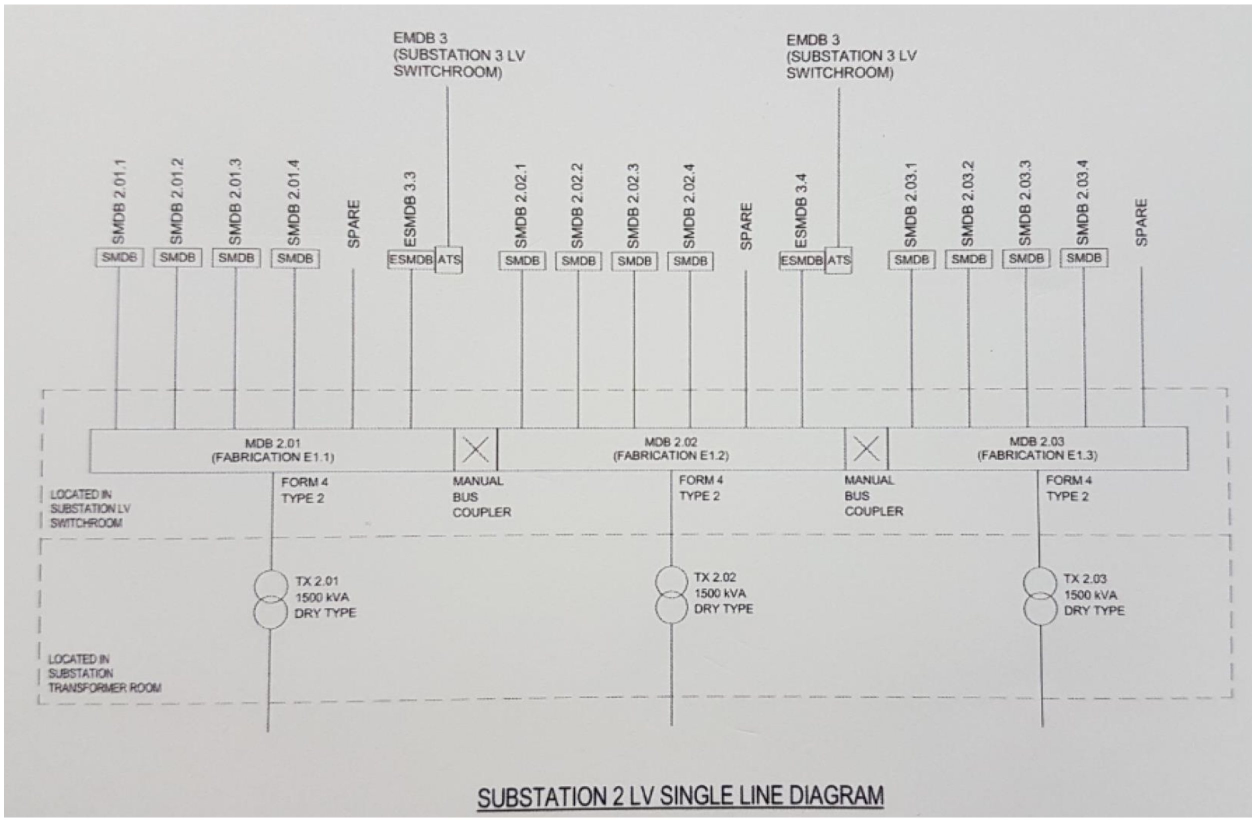


The mobile application is made up of six classes as seen in the figure. Each class represents the functionality in one page. The first class, or the HomePage class, has functions to handle signing in and signing up procedures. It has member functions to display error messages in cases of unsuccessful logins. The HomePage class has a TabsPage object that it calls when the login is successful. The TabsPage class has a MeetingsPage and a ProfilePage objects assigned to each of its tabs. It is mainly used for navigating the app. The MeetingPage class uses the Calendar service to retrieve a list of all calendar events. It also has the biggest class in our code, which is the RecordPage class. The RecordPage uses multiple services including the MediaCapture service to record the audio signal in .m4a format, the File service to access the recording after it is completed, and the FirebaseStorage service to upload the .m4a recording to the server. The ProfilePage has a ResultsPage object used for displaying the results.

- **Single Line Diagrams:**

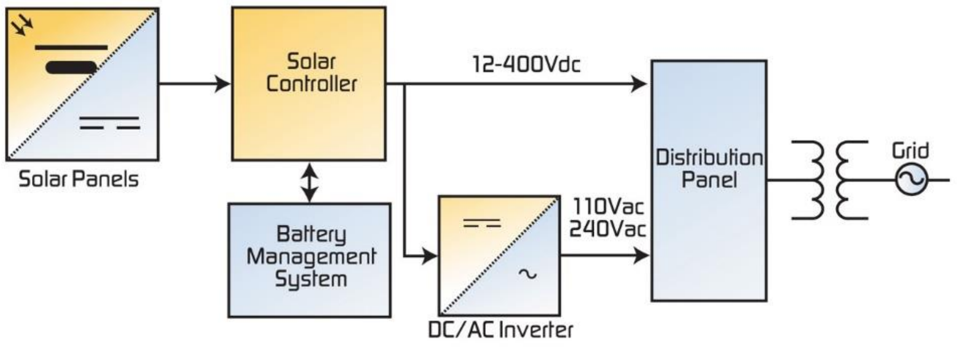


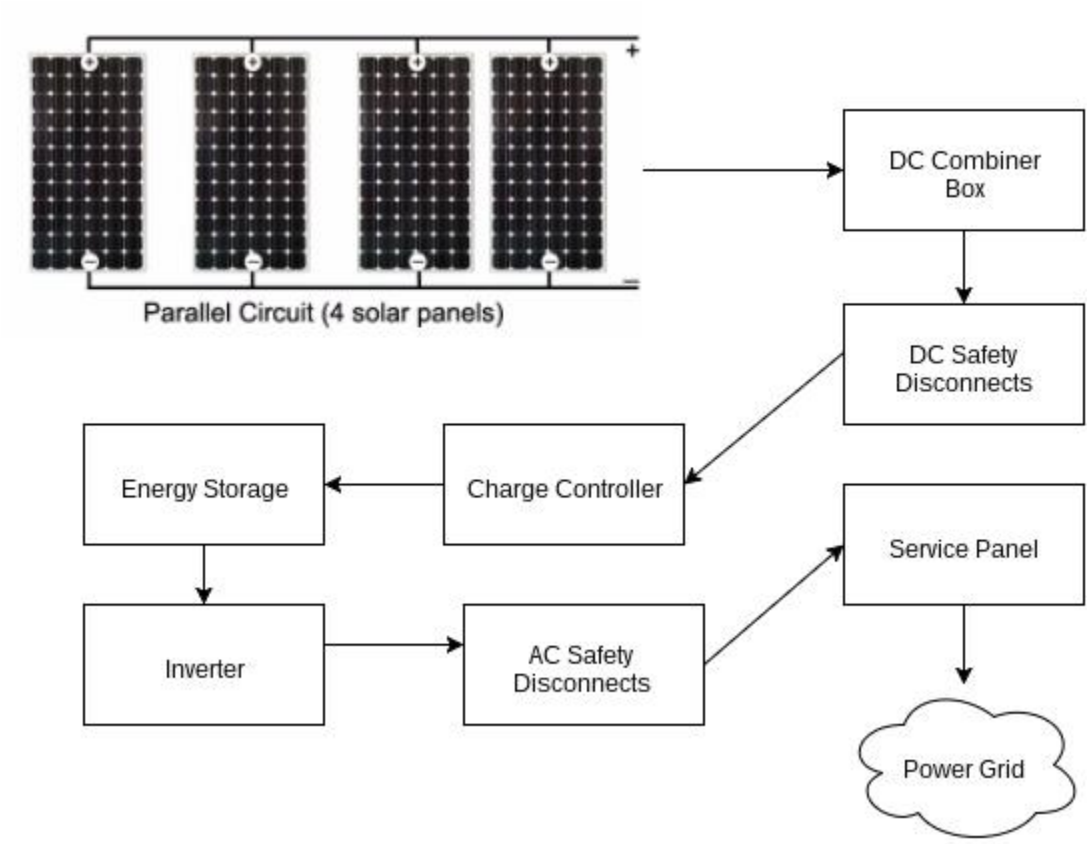
Single Line Diagram for Substation 2



EMDB 3 (SUBSTATION 3 LV SWITCHROOM)
EMDB 3 (SUBSTATION 3 LV SWITCHROOM)
SMDB 2.01.1
SMDB 2.01.2
SMDB 2.01.3
SMDB 2.01.4
SPARE
ESMDB 3.3
SMDB 2.02.1
SMDB 2.02.2
SMDB 2.02.3
SMDB 2.02.4
SPARE
ESMDB 3.4
SMDB 2.03.1
SMDB 2.03.2
SMDB 2.03.3
SMDB 2.03.4
SPARE
SMDB
ESMDB
ATS
MDB 2.01 (FABRICATION E1.1)
MDB 2.02 (FABRICATION E1.2)
MDB 2.03 (FABRICATION E1.3)
FORM 4 TYPE 2
MANUAL BUS COUPLER
LOCATED IN SUBSTATION LV SWITCHROOM
TX 2.01 1500 kVA DRY TYPE
TX 2.02 1500 kVA DRY TYPE
TX 2.03 1500 kVA DRY TYPE
LOCATED IN SUBSTATION TRANSFORMER ROOM
SUBSTATION 2 LV SINGLE LINE DIAGRAM

- **Solar Power System Design Diagram:**





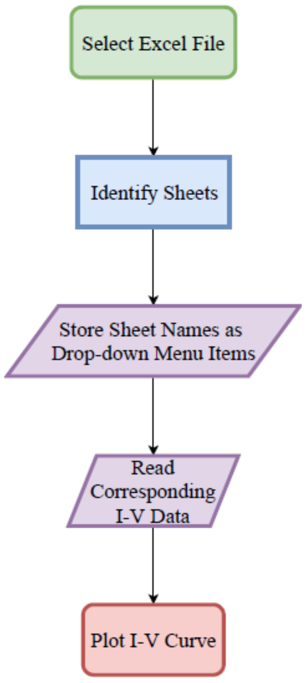
- **Graphical User Interface (GUI) Design:**

When you design your GUI, you need to explain all its functionality in forms of text, block diagrams, or flowcharts, etc. It is necessary to include both the layout design and the functionality design. Also, include segments of the major function codes in your GUI. A sample of such is presented here.

The GUI should do the following:

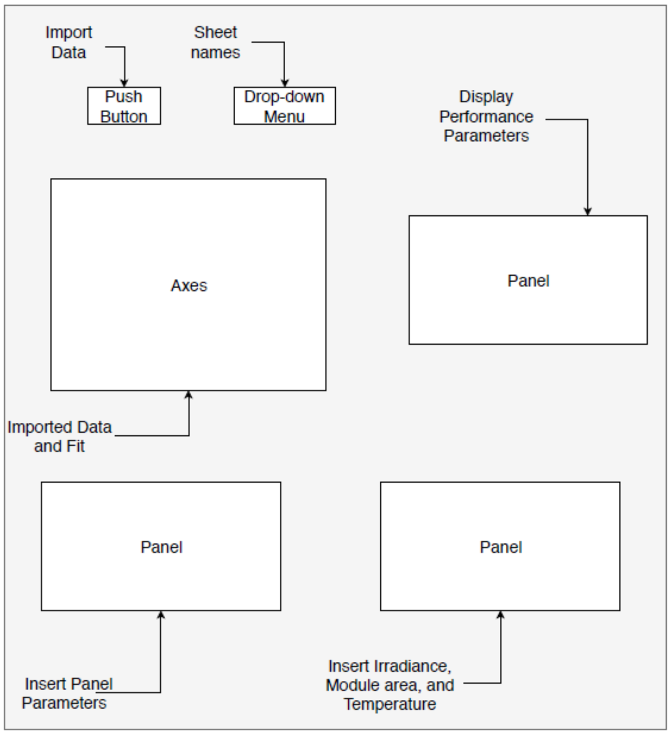
1. Ask the user to select the Excel file.
2. Identify the sheets within the Excel file.
3. Display sheet names in a drop-down menu where the user can select a sheet.
4. Plot the I-V curve for the selected sheet from the data.
5. Fit the data and plot the fit result alongside the imported data.
6. Calculate and display the performance parameters.

When the user clicks the button, the GUI asks for the Excel file, which contains the data then automatically detects the sheets. The sheets are stored as items for the drop-down menu; after that, the user can choose from the drop-down menu which sheet's current and voltage data to be plotted. The below figure shows a flowchart explaining the functionality of the "import data" button.

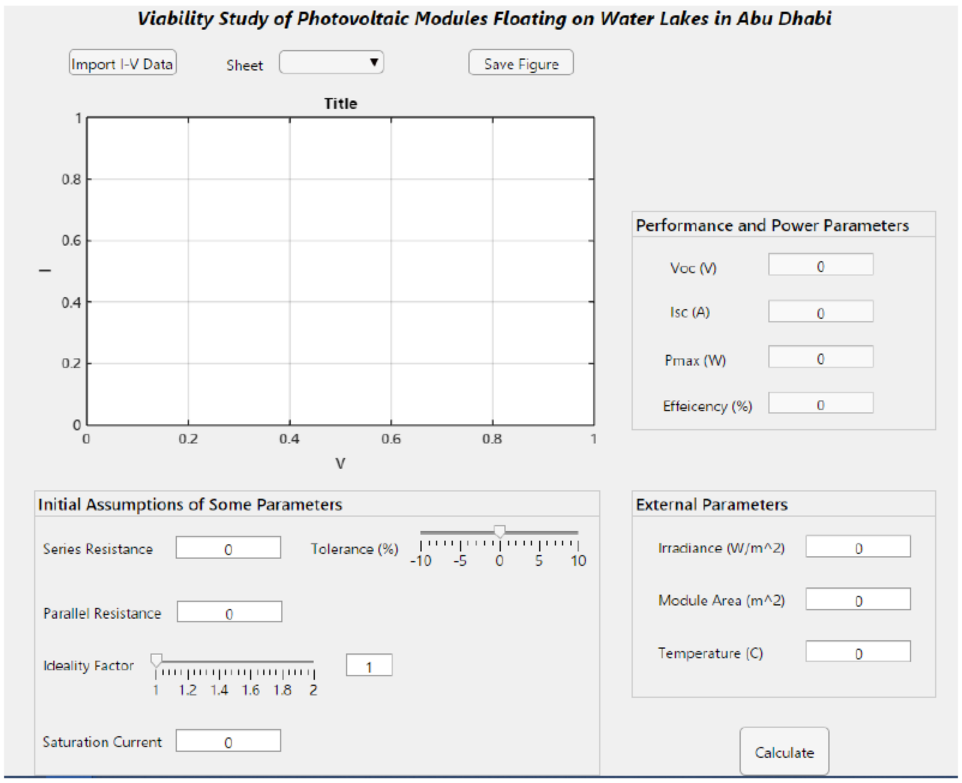

Select Excel File
Identify Sheets
Store Sheet Names as Drop-down Menu Items
Read Corresponding I-V Data
Plot I-V Curve

Importing Data Flowchart

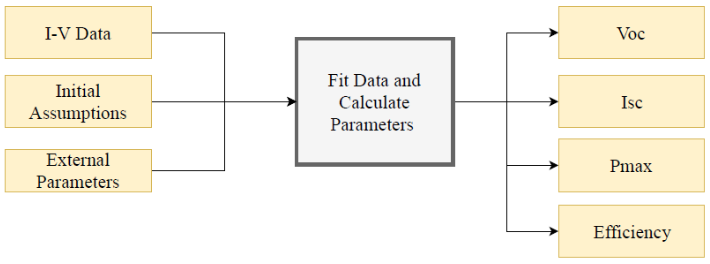
The following figures show the final planning for the layout and the final design for the GUI, respectively. A quick search shows that the ideality factor n is between 1 and 2; therefore, a slider can be used to optimize the initial assumption for this parameter. Another can be used with R_s for tolerance.



Final Planning for GUI Layout.

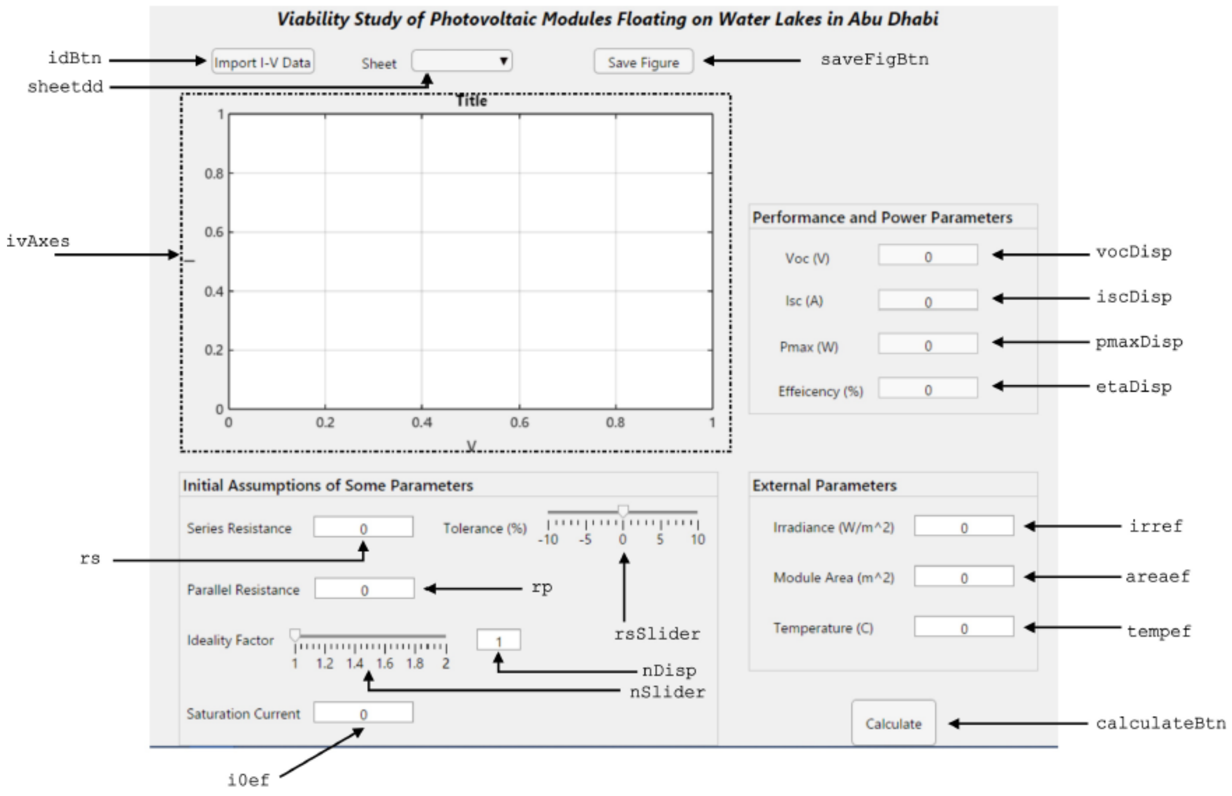


The following is a block diagram explaining what happens when the user presses the "calculate" button.



Overall GUI Functionality Block Diagram.

When a component is created in App Designer, it's automatically given a default variable name which can be changed in the Component Browser window in App Designer. All these variables are save as a property in the "app" variable. If we wish to get a handle for a button named "my_button" we have to write "app.my_button". The below figure shows the GUI components and their corresponding variable names.



The following Listing shows the function "importData" which is called when the "idBtn" is pushed.

```matlab
function [x_lim,y_lim] = plotIV(app)
    value = app.sheetdd.Value;
    V_measured = app.T(value).Data.V;
    I_measured = app.T(value).Data.I;
    app.Isc = app.T(value).Data.I(1);
    plot(app.ivAxes,V_measured,-I_measured,'--d');
    x_lim = app.ivAxes.XLim;
    y_lim = app.ivAxes.YLim;
    app.ivAxes.Title.String = ['I-V Characteristics for Sheet' app.sheetdd.Items(
    value)];
end
```