

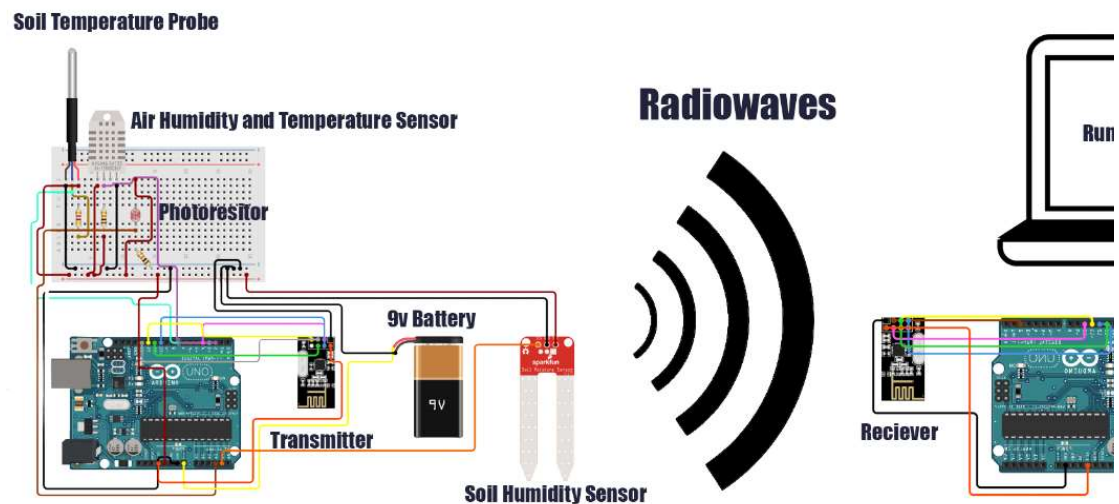
# Design

## Table of Content

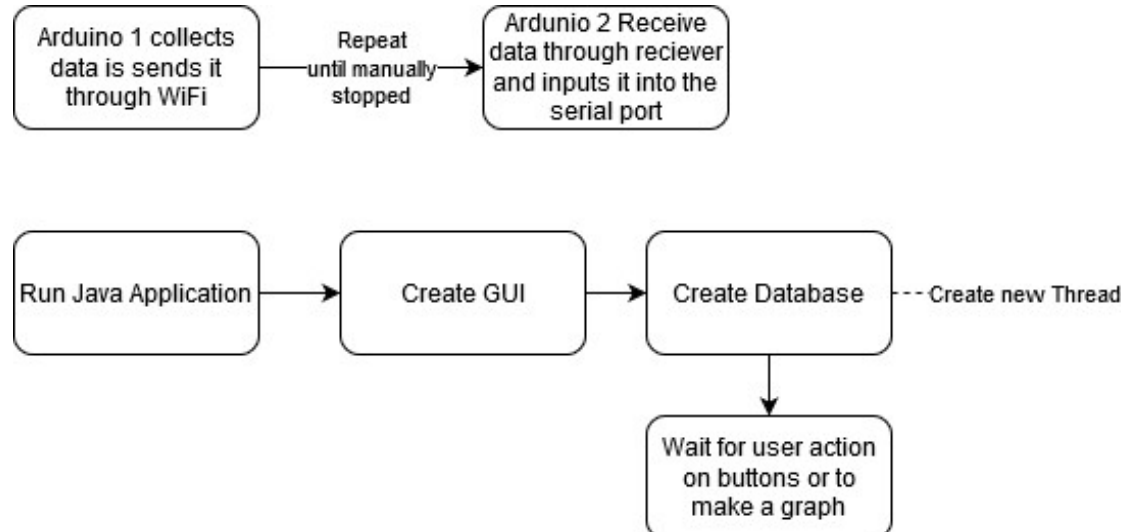
- Flow chart for the entire application
- Structure of Application
- GUI Design
- Database Design
- Key Features of the Application
  - Sending the data
  - Receiving Data
  - Reading the Serial Port
  - Graphing Data
  - Performing Calculations on Data
  - Saving Data onto a text file
  - Importing data from a text file
  - Reading and Working the Database
- Test Plan

- **Flow chart for entire Application**

The following flow chart maps out on a very basic level how the entire applications works.



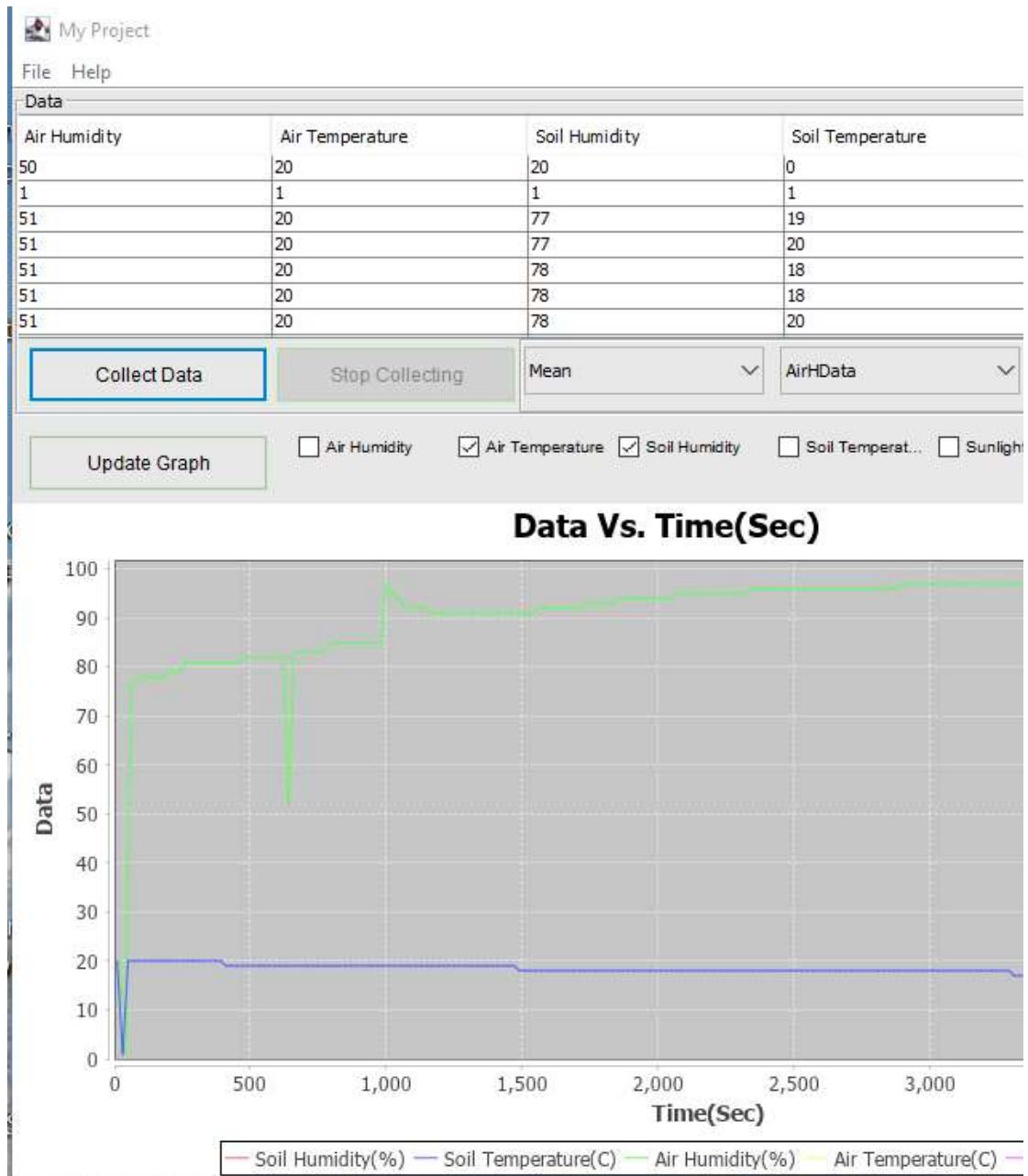
(Diagram 1: Illustration of the two Arduinos)



(Diagram 2: Flow chart for the Application)

- **Graphical User Interface Design**

Next will be the Graphical User Interface (GUI) with labels showing what everything is.



(Diagram 3: The GUI)

1. Data Received from Arduino displayed on a table
2. Button that starts collecting data from the Arduino
3. Button to stop collecting data
4. Drop down menu with: Mean, Mode, Max, Min and Range. The user selects which one they want to calculate
5. Drop down menu with: AirHData, AirTData, SoilHData, SoilTData, Sunlight. The user selects which data set they want the calculation to be performed on
6. Button that calculates the calculation from 4 and 6, then displays a text box with the

answer

7. Button to update graph with the data select in 8.
8. Checkboxes with every data set. The user selects which data they want displayed on the graph. More than one can be selected
9. The graph that displays the data
10. A menu which allows the user to save or upload data

- **Database**

The database is very simple with only one table and no relationships between entities.

Database Name: Data

Table name: allData

Entities:

Name	Type
AirHdata	Integer
SoilHata	Integer
AirTdata	Integer
SoilTdata	Integer
Sunlight	Integer

- **Structure of the Application**

All methods are present in the same class.

Method Name	Method Purpose
Main(String[] args): void	-Set properties for the GUI -Call BuildGUI
BuildGUI(): Constructor	-Generates all the components for the GUI (Buttons, Text field, graphs etc...)
CollectDataB(): Object(button)	-When clicked, calls CollectDataT()
StopCollectB(): Object(button)	-when clicked, stops the collecting of new Data
UpdateGraphB(): Object(button)	-When clicked, calls updateGraph()
CalculateB(): Object(button)	-When clicked, calls doCalculation()
UpdateTable(): void	-Updates the table

doCalculation(): void	-Checks two dropdown boxes' option and performs the appropriate calculation
updateGraph(): void	-Check checkboxes for which one is ticked and displays the data of the specified type on the graph
generateMenu(): void	-Generates the menu at the top of the GUI
infoBox(String infoMessage, String titleBar): void	-Called with a message and a title then will display a message box with the text
getData(): void	-Opens the serial port and collects the data -Every 5 data points collect it calls insertData()
CollectDataT(): void	-Creates a second thread that runs getData
insertData(): void	-Take data recived from getData() then inputs them into the database
openConn(): void	-connects to the database, if that fails it calles createDatabase()
closeConn(): void	-disconnects from the database
createDatabase(): void	-creates a database with the appropriate tables
updateTable(): void	-connects to the database and reads it while insterting the values into the table on the GUI
saveButton(): void	-starts saving the data by calling WriteTextFile()
WriteTextFile(): void	-connects to the database and inserts its values into a text file created
ReadTextFile(): void	-user selects a text file and then it reads the data and replaces the current data in the database with the data in the text file

### • Key Features

Next key features of the application will be outlined in order to give an idea of how they will function.

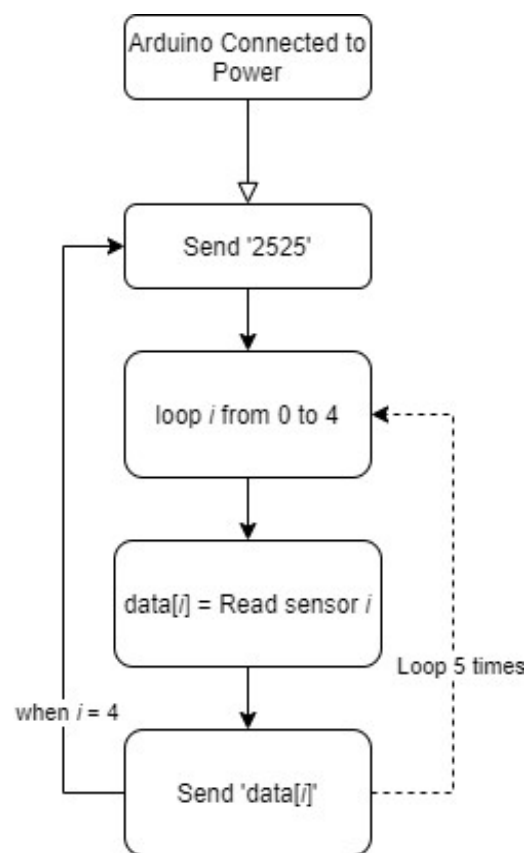
I will need:

- Arduino Sketch to collect and send data via radio transmission

- Arduino Sketch to receive radio transmission and send it to the PC
- Java Method to Read the Serial Port and insert the data into a database
- Java method to graph the data
- Java method to perform calculation on the data
- Java method allowing the user to save and import data.

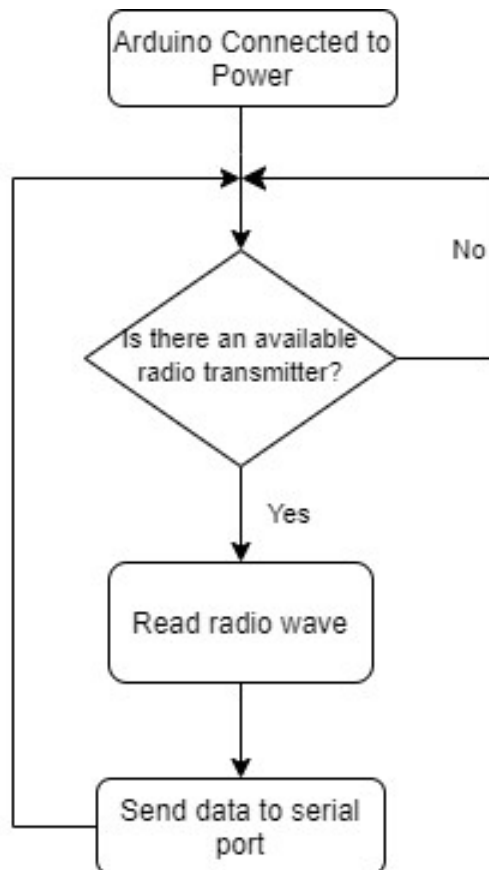
### 5.1 Collecting and sending data

This is how data is sent from Arduino 1. The 2525 serves as a flag to indicate the start of the data stream so the application knows when the data is starting.



(Diagram 4: Flow chart for Sending the collecting and sending data)

### 5.2 Procedure for Receiving Data

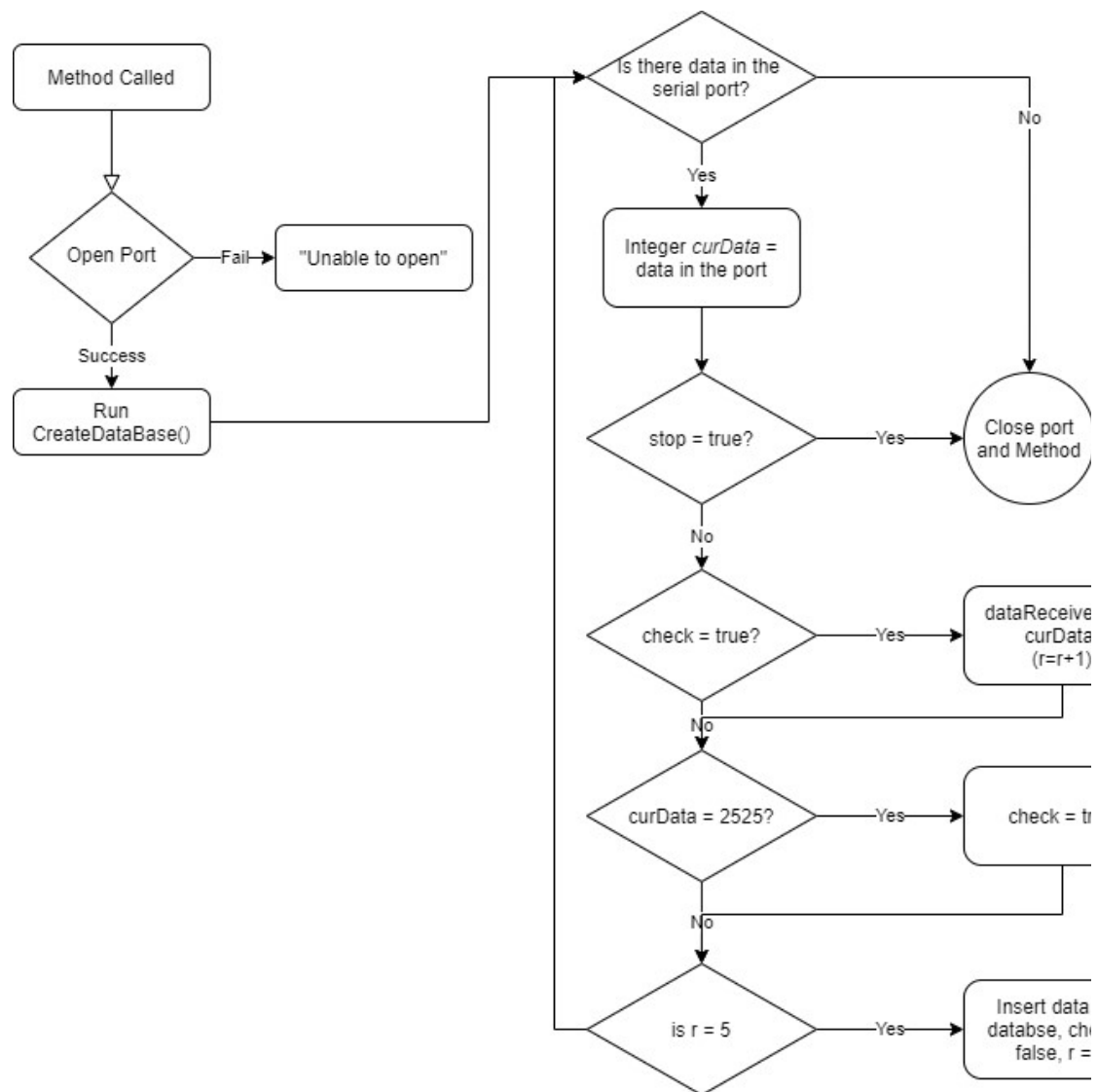


*(Diagram 5:Flowchart for Receiving Data)*

### **5.3 Procedure for Reading serial port and collecting data**

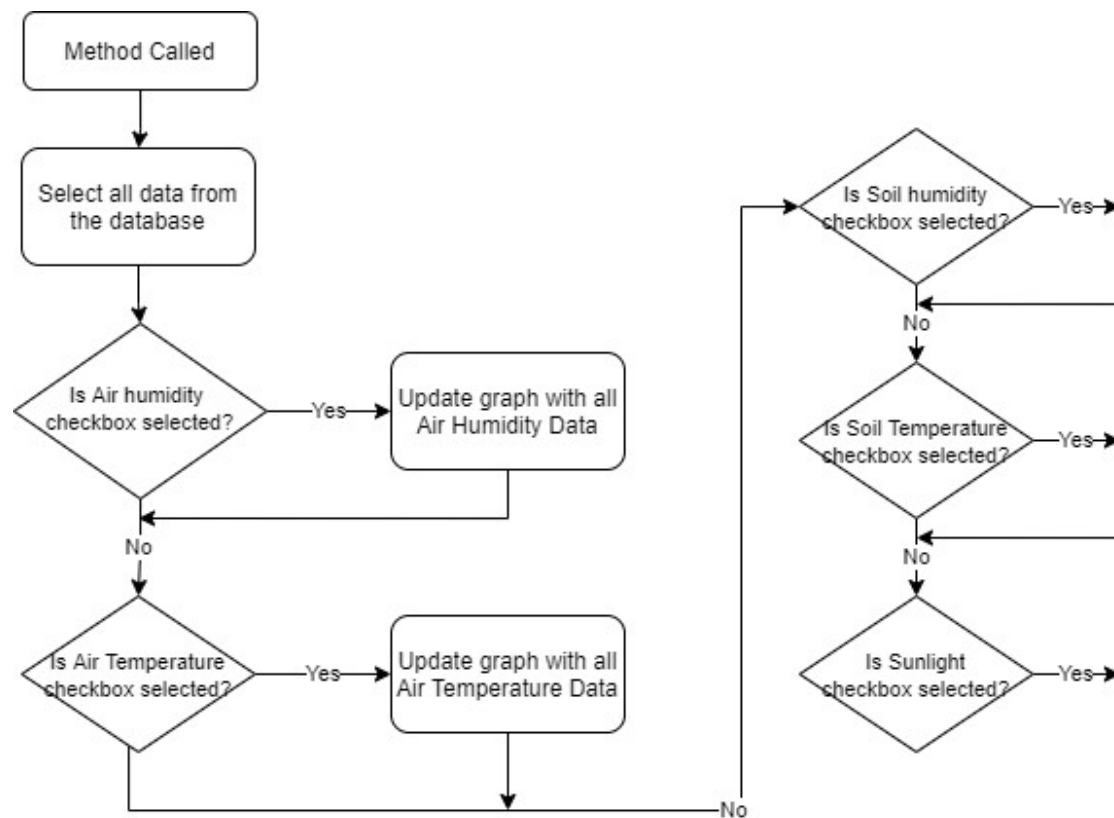
This procedure would be found within the Java application and would read the serial port and collect the data.





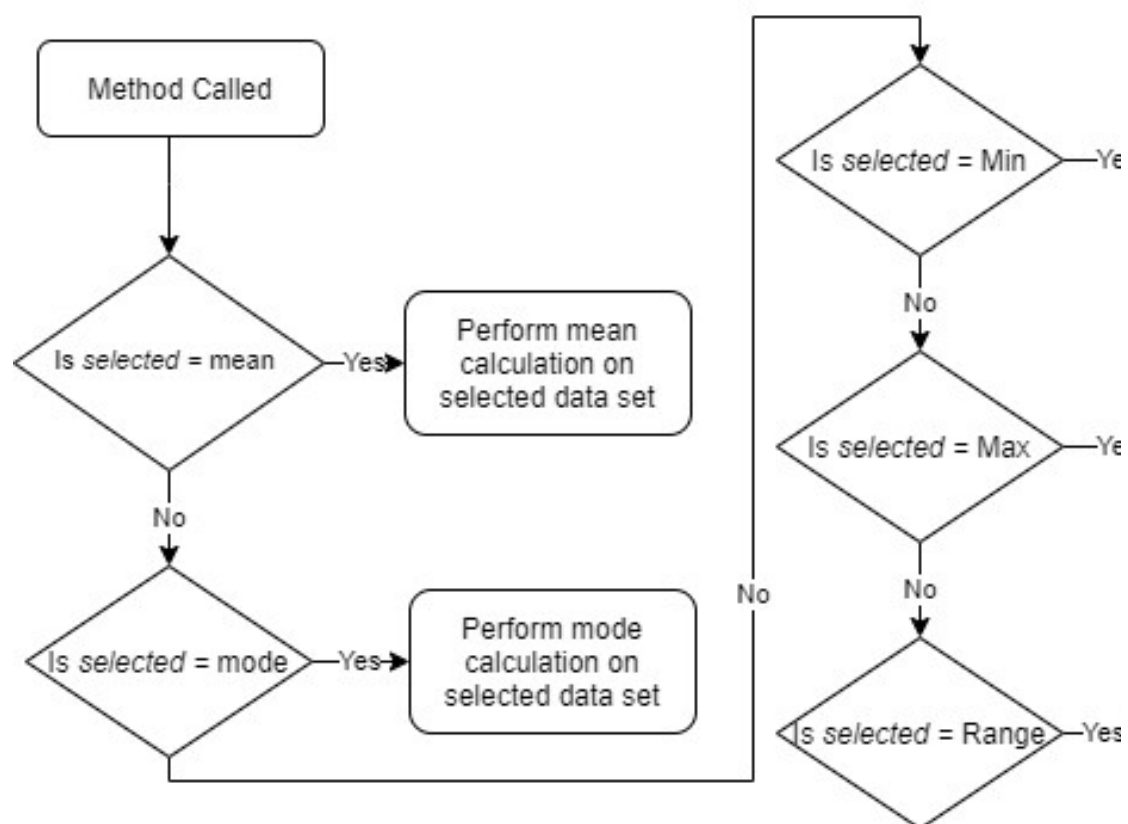
(Diagram 6: Flowchart for reading the seral port)

#### 5.4 Procedure for Graphing Data



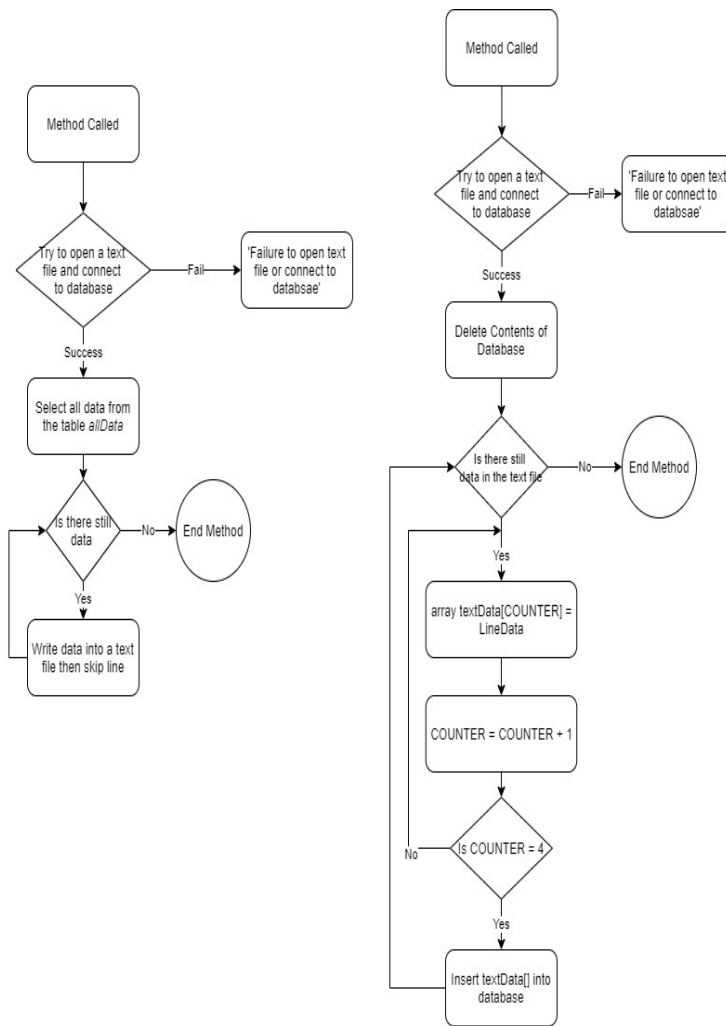
(Diagram 7: Flowchart for graphing the data)

### 5.5 Procedure to perform calculations



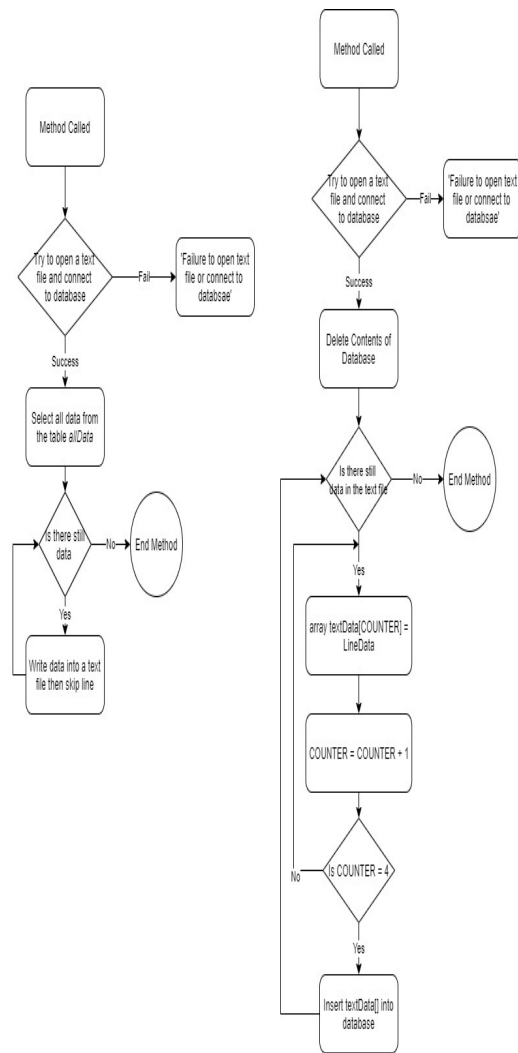
(Diagram 8: Flowchart for performing the various calculations)

### 5.6 Java Method to save data onto a text file



(Diagram 9:Flowchart for saving data to a text file)

## 5.7 Method to Import Data from a text file



(Diagram 10: Flowchart for importing data from a text file)

- **Reading data base and working the Data**

These are the SQL queries I will need for the application.

Store Data	INSERT INTO allData(AirHdata,AirTdata, SoilHdata, SoilTdata, Sunlightdata) VALUES "X1,X2,X3,X4,X4"
Fetch Data	SELECT <i>entities</i> FROM allData
Average	SELECT AVG <i>entities</i> FROM allData
Minimum & Maximum	SELECT MIN/MAX <i>entities</i> FROM allData
Range	SELECT MAX <i>entities</i> MIN <i>entities</i> FROM allData MAX – MIN
Mode	SELECT <i>entities</i> COUNT(*) FROM allData GROUP BY <i>entities</i> ORDER BY COUNT(*) DESC
Deleting Database Data	DELETE * FROM allData

- **Test Plan**

<b>The action</b>	<b>How it is tested</b>
The data is being correct sent through radio waves.	<ul style="list-style-type: none"> <li>• Send a specific data point from Arduino 1 to Arduino 2</li> <li>• Check the serial monitor to see if the data received is correct</li> </ul>
Is the Java Program correctly reading the port	<ul style="list-style-type: none"> <li>• When reading the serial port print it onto the Java terminal</li> <li>• Check if correct data is displayed</li> </ul>
Check if database can be correctly worked with	<ul style="list-style-type: none"> <li>• Perform all the necessary queries and check if they provide the correct value</li> </ul>
Check if Graph and Table are displaying data correctly	<ul style="list-style-type: none"> <li>• Manually input data and check if the graph and table is correctly displaying the data</li> </ul>
Check if program correctly saves and reads text files	<ul style="list-style-type: none"> <li>• Manually put data into the database and save it to a text file. Check if text file has correct values</li> <li>• Make a text file with custom data. Read it from the application. Check if application correctly read it.</li> </ul>

**The client as seen in Appendix 3 Line 65 has approved design Development**