1. INTRODUCTION

"The beginning is the most important part of work"

— Plato

1.1 Home Automation

As you know Internet of Things (IoT) is the concept actually refers to objects and devices such as cars, watches, home appliances, etc. that have communication technology and capability to connect. These devices are commonly known as smart devices. Today, a large number of common home appliances such as lamps, switches, and door locks have become smart and can be easily controlled even if you are not at home with a home computer, laptop, tablet and smartphone. So, these Automated Houses are also called Smart Houses or Smart Homes.

1.2 Smart Switchboard

Smart Switchboard is electronic switchboard which directs power from one or more source of electricity to many regions or devices that can be controlled by mobile phone. With this tool, all electronic devices can be controlled via Bluetooth. The most convenient option for our connection between mobile phone and the device controller is to use Bluetooth technology. Because this technology is available in almost all mobile phones, it makes it possible to establish a secure and hassle-free wireless connection. Common radio door opener systems have low security and can be heard and detected by the passcode, but the nature of Bluetooth technology uses Frequency-hopping along with advanced encryption, which makes it extremely secure and impenetrable.

1.3 Objective

- Energy Saving is one of the significant benefits of the Smart Switchboard that can save users from high bills due to electric appliances waste energy while they are not required to be used.

- It will help those who have a physical disability and can't move properly, with the help of this Smart Switchboard, they can do their work alone with their smartphones at homes without any help from others.

- Convenience, Smart Switchboard will enable users either to use it physically or control it from their mobile phones. It is working in two way switching mechanism.

- It's all about one-time cost, investing money on Smart Switch Boards are saving future's bills.

- Smart Switchboard reduces the possibility of fire whenever the smoke or fire is detected inside the switch board it will stop the flow of electricity.

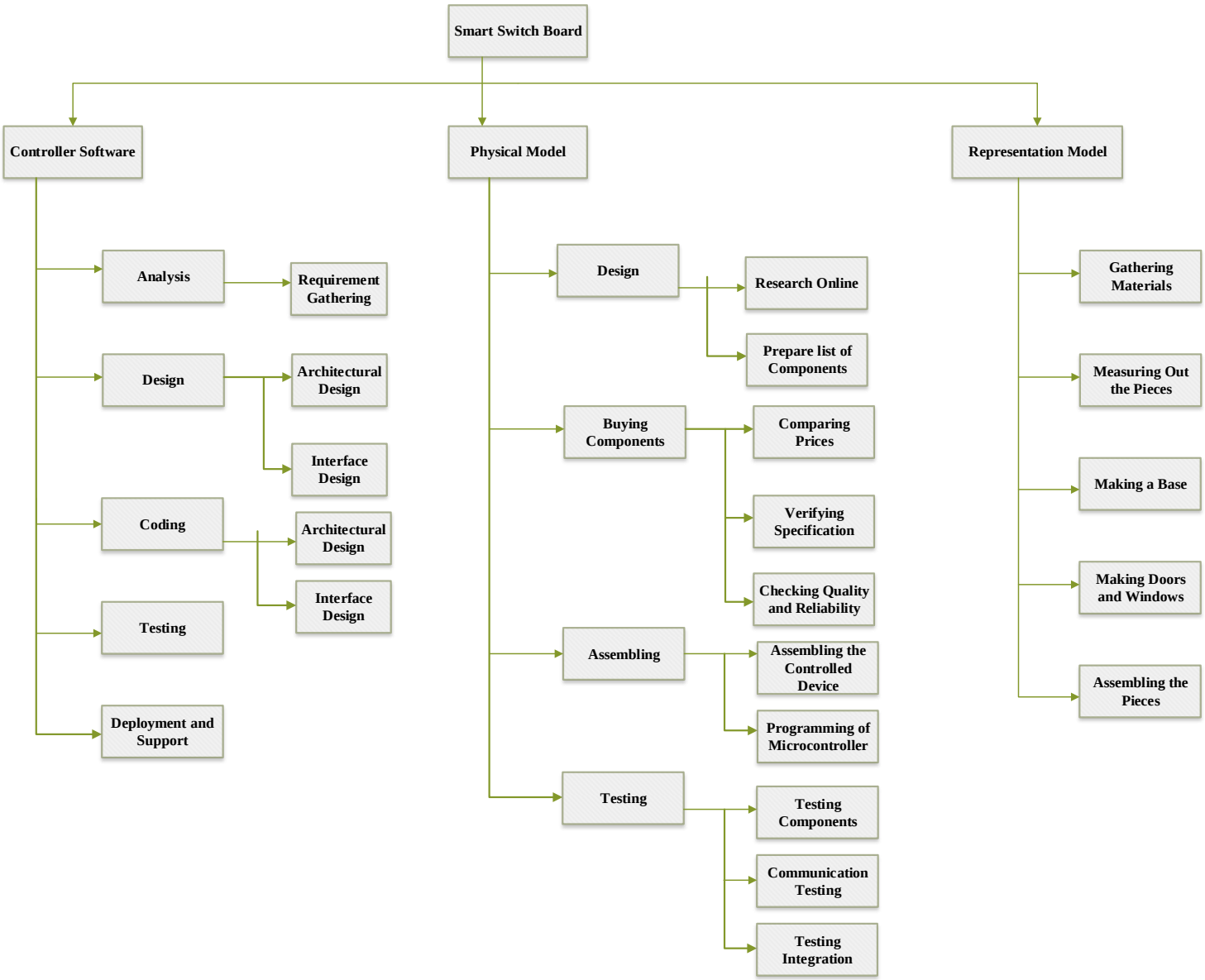
- It can also reduce the possibility of fire that happens due to overheating of appliances with the help of Smart Switchboard users can set the timer for how long the appliances will run.

2. PROJECT PLAN

Project planning is the process of writing a standard document to guide the control and execution of the project.

- To document and discuss shareholder products and project expectations
- To control work schedule and product delivery
- To calculate and control corresponding risks

2.1 Work Down Structure



2.2 Risk Analysis

Concerns associated with Smart Switchboard:

Smart Switches need to be installed instead of their traditional switches, which require a little knowledge and work with electrical wiring. It is not easy for many consumers to check the installation of a smart switch and it requires expertise.

Smart Switchboard needs to be connected with mobile devices through Bluetooth connection which can drain the battery of the device.

Smart Switchboard is working in a two-way switching mechanism so it would be difficult for the user to determine the state of switch whether it is On/Off.

2.3 Preliminary Investigation

In a world where everything is going digital and things and electronic devices are capable of doing wonders. Still after having so many resources we are still using manual switch boards in our homes for which we have move from one place to another in our room to turn on or off a bulb or fan or any electronic device connected to that. In the world where everything is getting connected digitally for making our lives easier and comfortable. So, using IoT we can solve the problem of manual switch boards and make it digital that can be controlled from a smartphone and while also sowing current time and temperature on the board too. Many times, we forget to turn off electrical appliances when leaving for office or any work place or going to bed. In this we can manually set the time to turn it off which will help in energy conservation at the same time.

- Control processes locally
- Gather, Monitor and process real-time data
- Switching traditional electrical appliances to Smart devices is very difficult task

2.4 Feasibility Study

Feasibility study or feasibility studies is the potential for evaluating and analyzing a proposed project and is based on research and studies that support the decision-making process.

2.4.1 Economic Feasibility

As we have studied and analyzed most of the Internet of Things (IoT) projects are feasible economically. The purpose of IoT projects is It should be cheap and affordable that people can afford it easily. In Smart Switchboard we have used the following components with its prices:

- Arduino Uno Board - 450Rs.
- 8 Channel 5V Relay Module 450Rs.
- HC-5 Bluetooth module 500Rs.
- MQ2 Smoke and Fire Detector 250Rs.
- Switch Board and all related components costs 300.
- Development tools and frameworks are open source and almost free of cost.
- Cardboard Model Costs 1000Rs for Representation purpose.
- Electronic cables and jumper wires and other small components costs in total 200Rs.

Note: The cost of testing and failure is not mentioned. For example, if some of the electronic components were damaged during the implementation of project.

2.4.2 Operational Feasibility

As operational feasibility is mainly concern about issues like whether the system will be used if it is developed and implemented successfully. As the technology is advancing day by day and home are emerging towards Smart Homes so we can say hopefully yes, the Smart Switchboard will have great demand in future in terms of usability and convince.

Our projects will be successful if we answer the following questions.

Are the users are being happy with the current manual system? Will Smart Switchboard reduce the time (operational time) considerably?

Will the Smart Switchboard system really benefits organizations? Will the system effect the customer's inconsiderable way?

Answer to the first question we can say the users are happy with the current manual system but not satisfied because they have limited control over it. They will be happy and satisfied if they acquire more control and customization on the Switch Board. Smart Switchboard can be customized and controlled conveniently.

It will reduce the operation time dramatically; everything will be in the fingertip of users. They will be able to control the whole board from their smartphone.

In addition to it, the proposed system will also benefit the organizations from overbilling and saving energy and cost. Organizations can bring more customization for their benefits if they want by re-coding the main board of smart switch to make limits on what time the Smart Switchboard starts working and on exactly what time It should stop working. It will increase the productivity and efficiency of organizations.

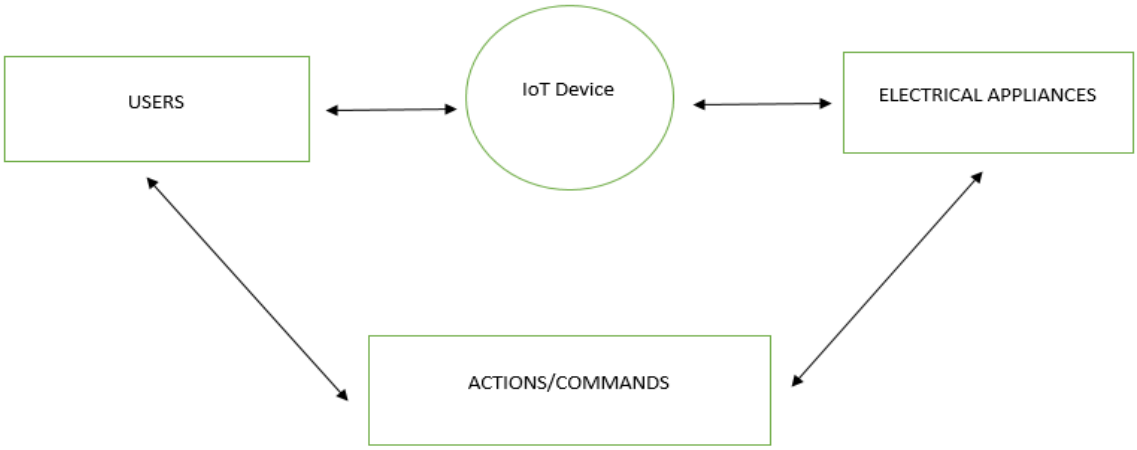
It will affect the customers of any organization if the organization save the cost of the product and services, they will afford their products and services at a lower price to its customers.

Ultimately, there is no need for special training of using of Smart Switchboard to its user. Almost everybody is familiar with the smartphone and they know how to turn on or off the flashlight of his/her smartphone.

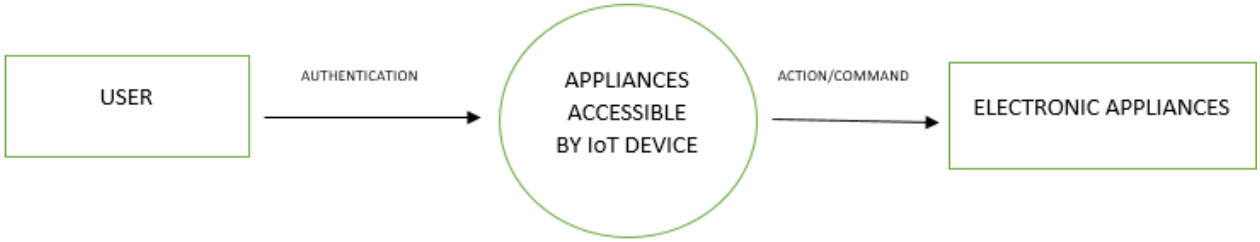
3. DFD, ER DIAGRAM, TABLE STRUCTURE, DATA STRUCTURE

3.1 DFD (Level 0,1,2)

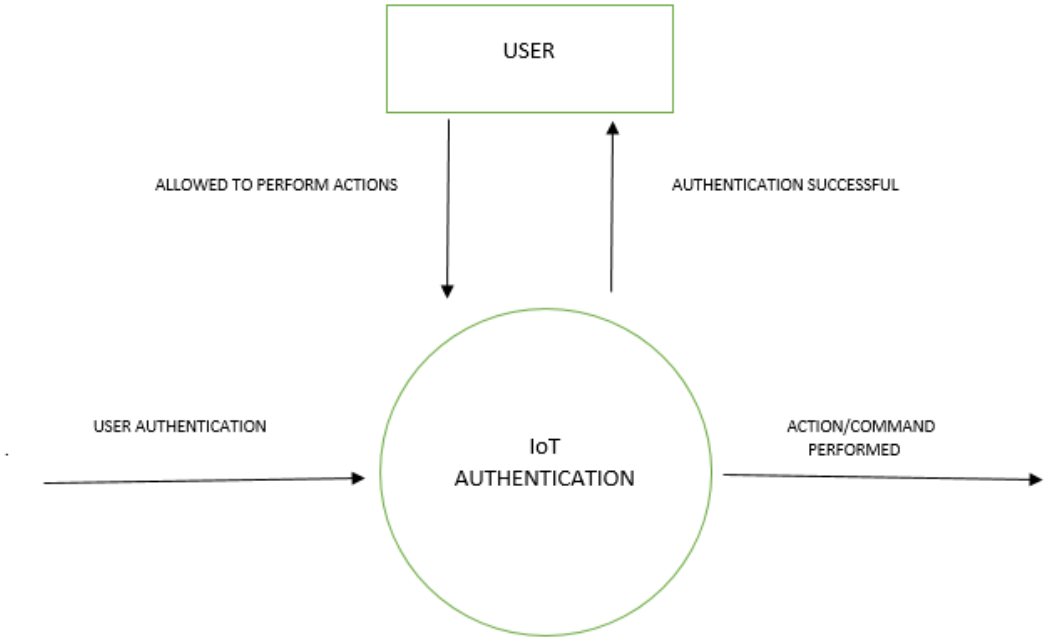
LEVEL 1



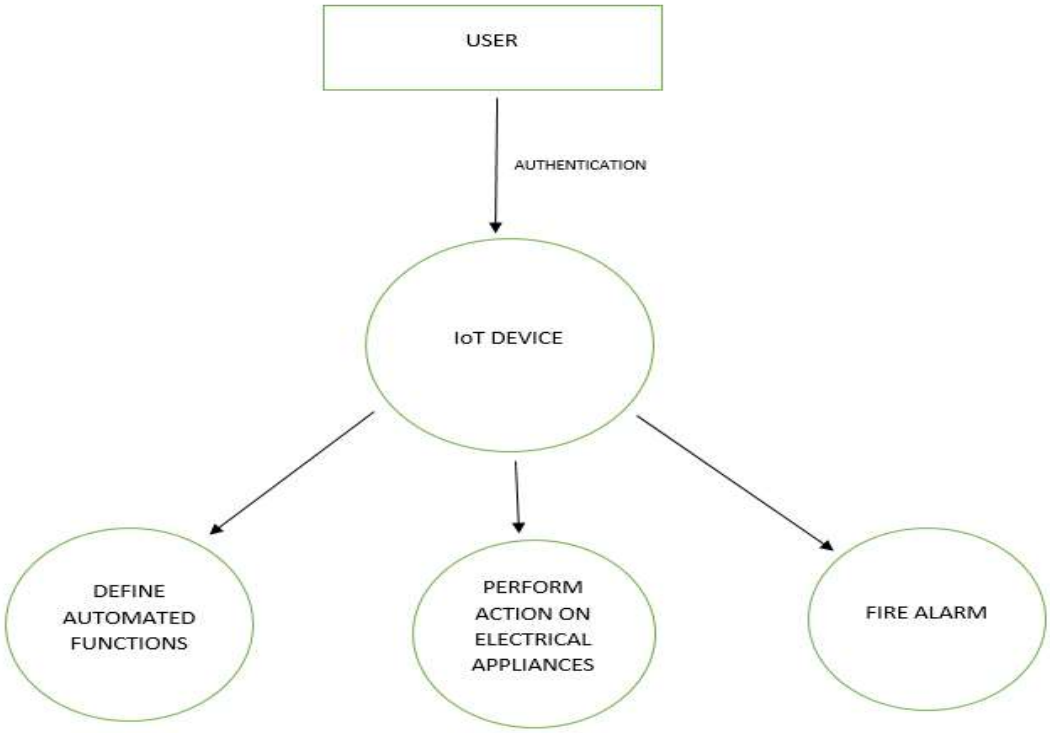
LEVEL 2



LEVEL 3


USER
ALLOWED TO PERFORM ACTIONS
AUTHENTICATION SUCCESSFUL
USER AUTHENTICATION
IoT
AUTHENTICATION
ACTION/COMMAND
PERFORMED

3.2 ER Diagram



4. SOFTWARE AND HARDWARE REQUIREMENTS

4.1 Software Requirements

Operating System	: Android
Technology	: Bluetooth
IDE	: MIT App Inventor, Arduino IDE
Programming Lang	: C++, JAVA
Designing and DFD	: Microsoft Visio

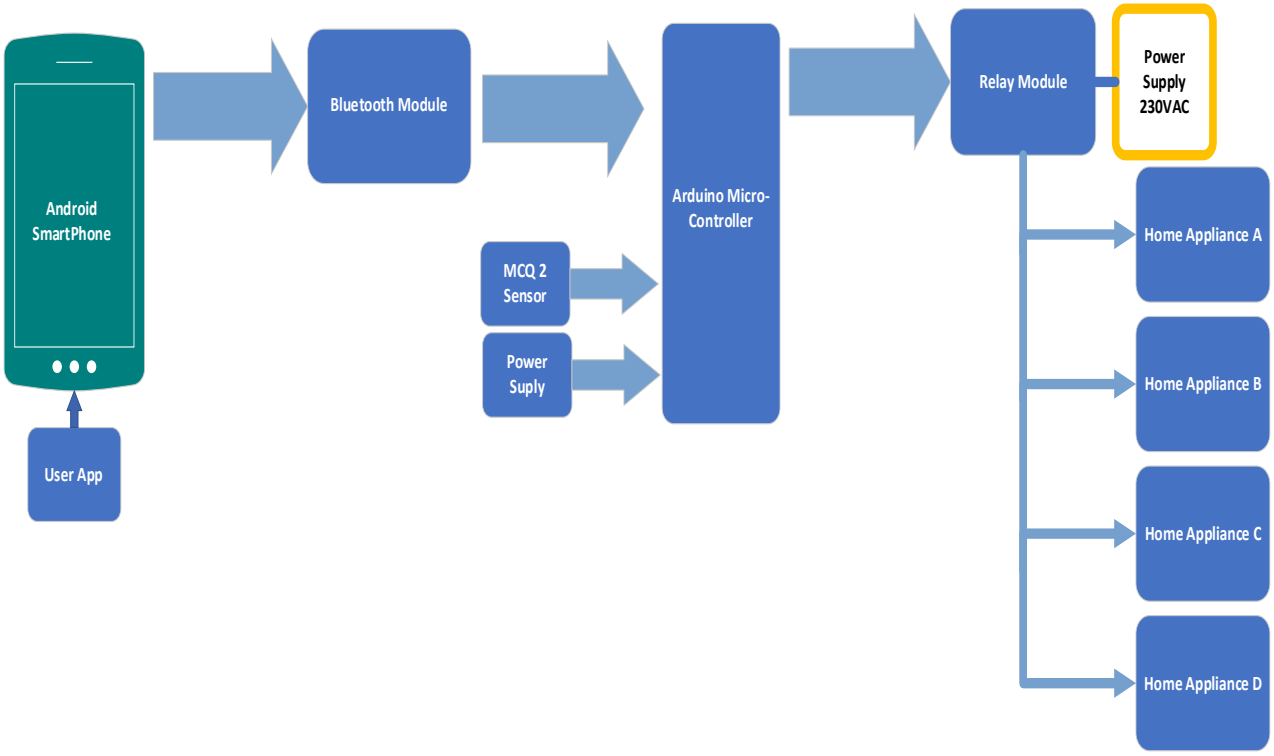
4.2 Hardware Requirements

Microcontroller Board	: Arduino Uno
Communication Module	: HC-05 Bluetooth Module
Relay Module	: 8 Channel 5V Relay Module
Sensor	: MQ-2 Smoke/Fire Detector
Switch Board	: 2 Way Electric Switchboard
Wiring	: Jumper wires, Electric wires
Controller	: Android based Smartphone
Demonstration	: Card Board Model, LCD Bulbs
Communication Media	: Wireless (Bluetooth)
Power Source	: AC-to-DC adapter/USB/Battery

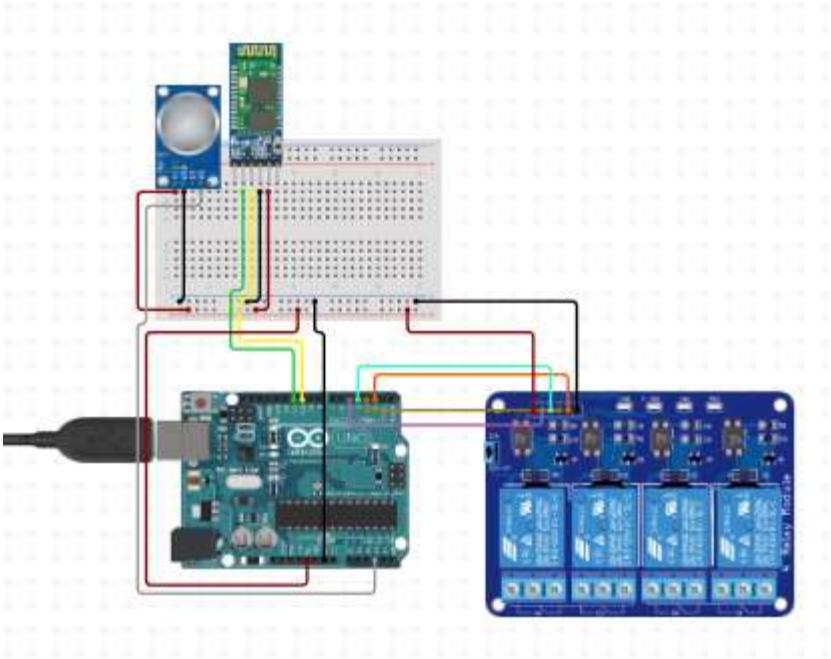
5. SYSTEM DESIGN

System design is a process by which a system's model, architecture, modules is defined and created, which is intended to achieve goals, making use of a set of elementary elements and subject to restrictions.

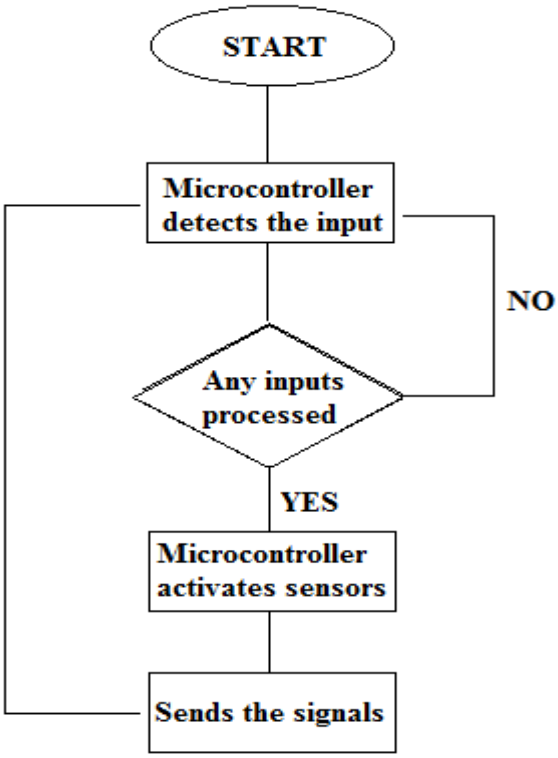
System design of Smart Switchboard is divided into two parts the physical model and the software model. In physical model we have all the electronic components and circuitry the way in which all those components are integrated. Similarly, in Software model we have two part the first part is User App by which user will be able to control the Smart Switchboard and program for the Arduino Microcontroller Board through which the board all the integrated components to the board will be controlled.



Circuit Diagram

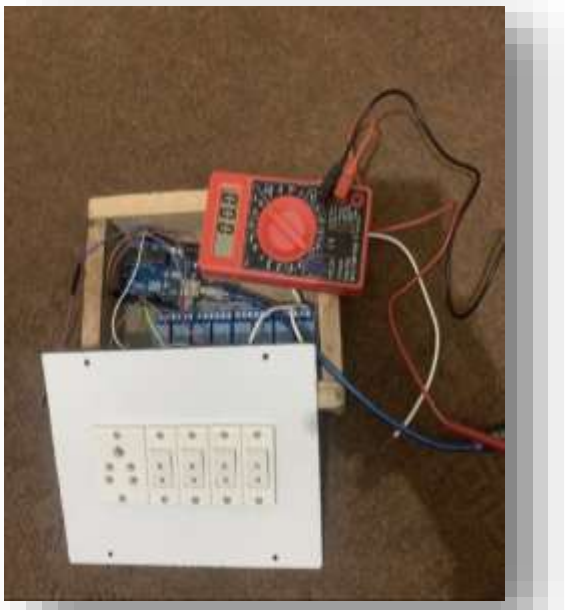


Flow chart of Microcontroller Board



User Interface Design of User App

Phyical Design of Model in Development Stage



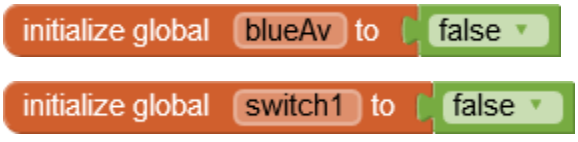
6. SCREENSHOTS WITH CODING

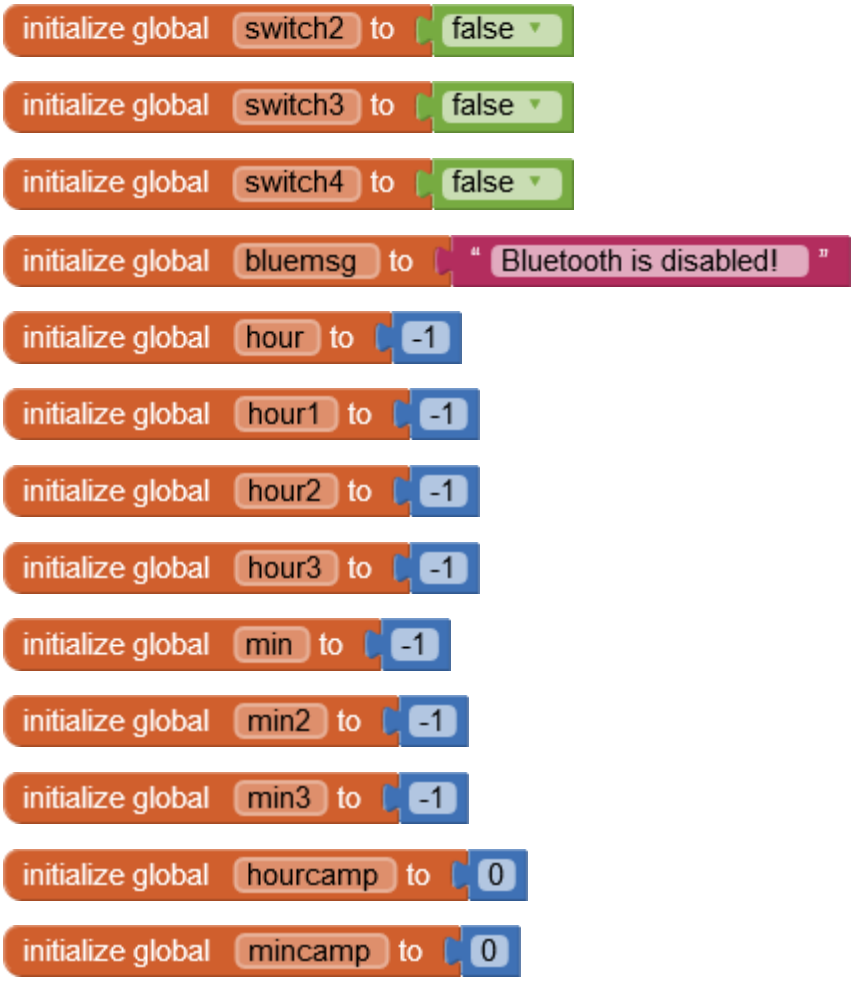
Microcontroller Board's Program

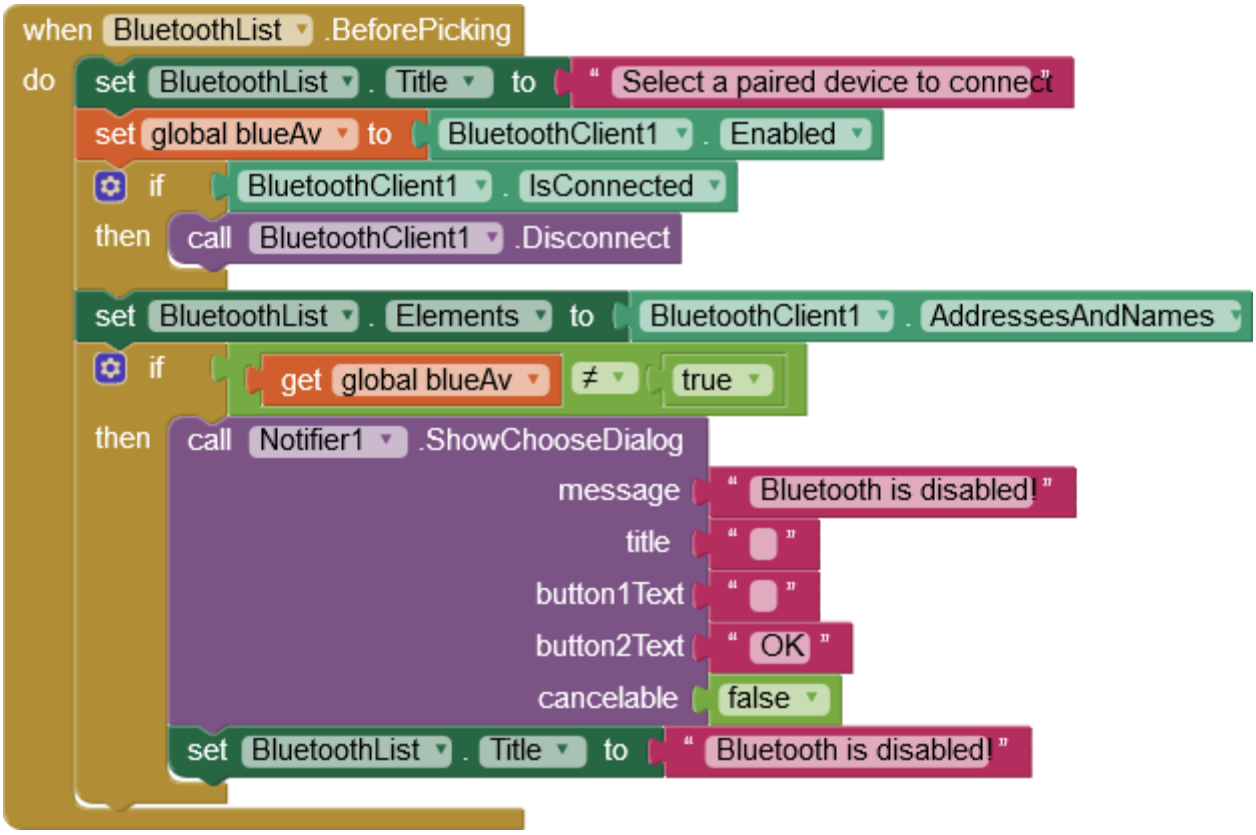
```
#define relay1 8 //Connect relay1 to pin 8
#define relay2 9 //Connect relay2 to pin 9
#define relay3 10 //Connect relay3 to pin 10
#define relay4 11 //Connect relay4 to pin 11
void setup() {
Serial.begin(9600);
pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
pinMode(relay3, OUTPUT);
pinMode(relay4, OUTPUT);

digitalWrite(relay1, LOW); //Switch relay1 off
digitalWrite(relay2, LOW); //Swtich relay2 off
digitalWrite(relay3, LOW); //Switch relay3 off
digitalWrite(relay4, LOW); //Swtich relay4 off

 }
void loop() {
  // put your main code here, to run repeatedly:
 if(Serial.available()>0)
   {
     char data= Serial.read(); // reading the data received from the bluetooth module
```

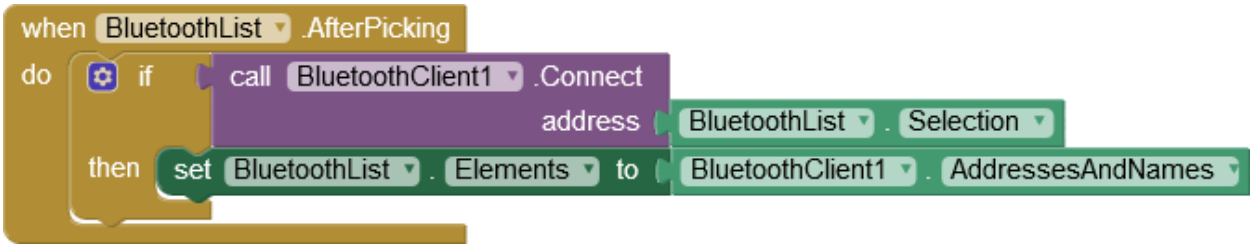
```
    switch(data)
    {
      case 'A': digitalWrite(relay2, LOW);break; //
      case 'a': digitalWrite(relay2, HIGH);break; //

      case 'B': digitalWrite(relay2, LOW);break; //
      case 'b': digitalWrite(relay2, HIGH);break; //

      case 'C': digitalWrite(relay3, LOW);break; //
      case 'c': digitalWrite(relay3, HIGH);break; //

      case 'D': digitalWrite(relay4, LOW);break; //
      case 'd': digitalWrite(relay4, HIGH);break; //

      default : break;
    }
    Serial.println(data);
  }
  delay(50);
}
```

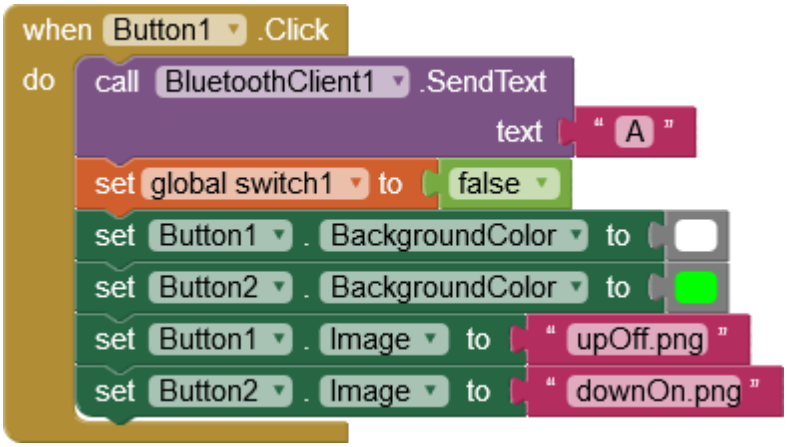
User App blocks-based Coding

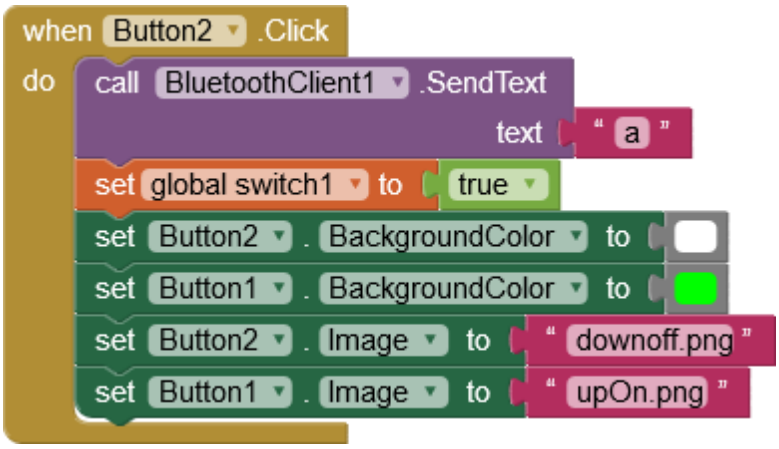


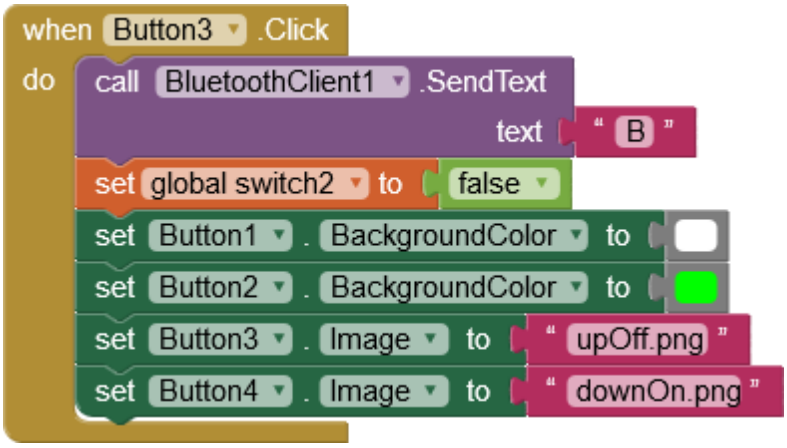

initialize global switch2 to false
initialize global switch3 to false
initialize global switch4 to false
initialize global bluemsg to " Bluetooth is disabled! "
initialize global hour to -1
initialize global hour1 to -1
initialize global hour2 to -1
initialize global hour3 to -1
initialize global min to -1
initialize global min2 to -1
initialize global min3 to -1
initialize global hourcamp to 0
initialize global mincamp to 0

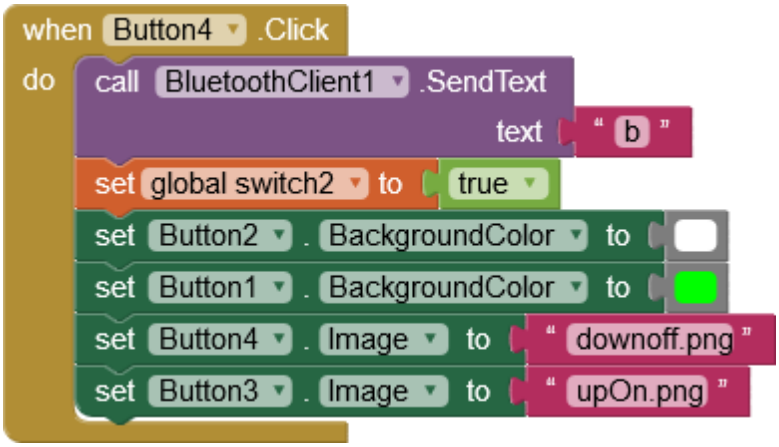

when BluetoothList .BeforePicking
do set BluetoothList . Title to " Select a paired device to connect"
set global blueAv to BluetoothClient1 . Enabled
if BluetoothClient1 . IsConnected
then call BluetoothClient1 .Disconnect
set BluetoothList . Elements to BluetoothClient1 . AddressesAndNames
if get global blueAv ≠ true
then call Notifier1 .ShowChooseDialog
message " Bluetooth is disabled! "
title " "
button1Text " "
button2Text " OK "
cancelable false
set BluetoothList . Title to " Bluetooth is disabled! "

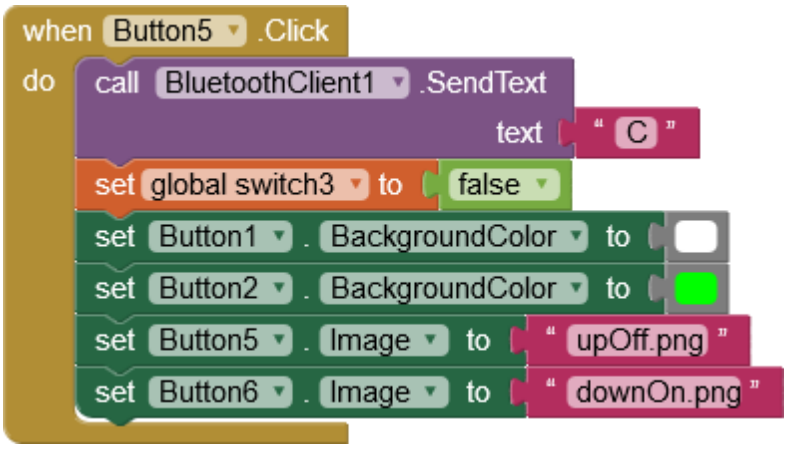

when BluetoothList .AfterPicking
do if call BluetoothClient1 .Connect
address BluetoothList . Selection
then set BluetoothList . Elements to BluetoothClient1 . AddressesAndNames

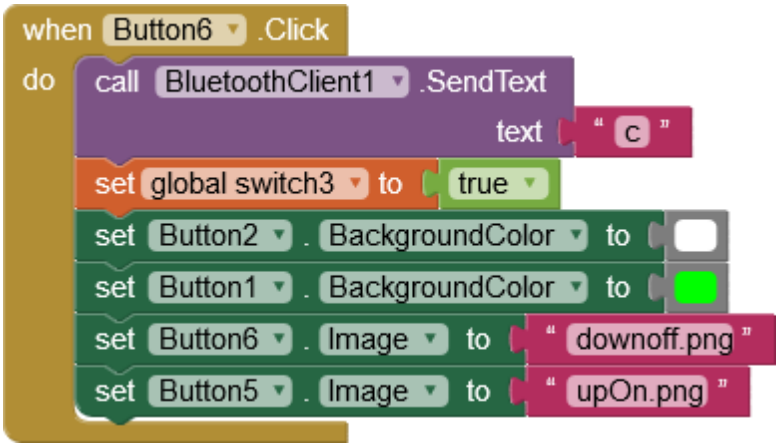

when Button1 .Click
do call BluetoothClient1 .SendText
text " A "
set global switch1 to false
set Button1 . BackgroundColor to
set Button2 . BackgroundColor to
set Button1 . Image to " upOff.png "
set Button2 . Image to " downOn.png "

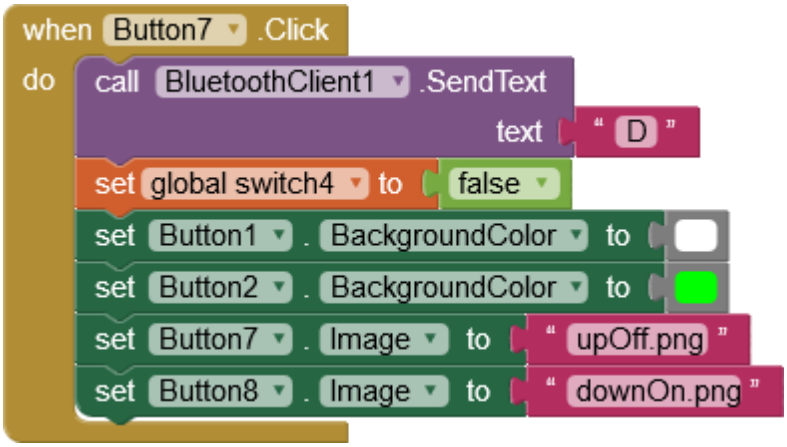

when Button2 .Click
do call BluetoothClient1 .SendText
text " a "
set global switch1 to true
set Button2 . BackgroundColor to
set Button1 . BackgroundColor to
set Button2 . Image to " downoff.png "
set Button1 . Image to " upOn.png "

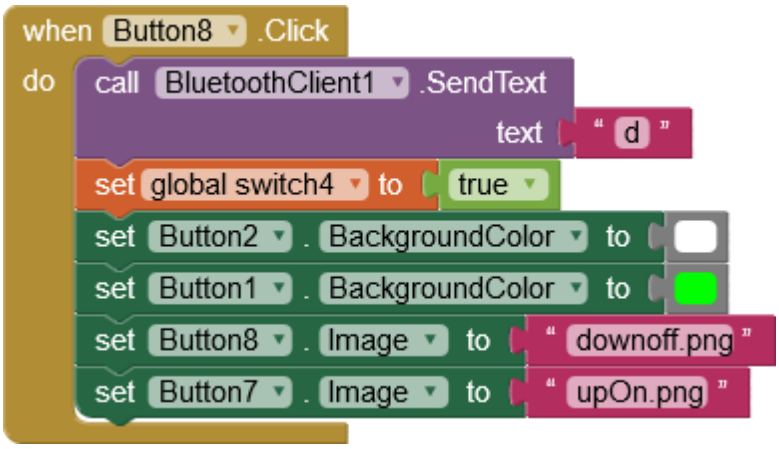

when Button3 .Click
do call BluetoothClient1 .SendText
text " B "
set global switch2 to false
set Button1 . BackgroundColor to
set Button2 . BackgroundColor to
set Button3 . Image to " upOff.png "
set Button4 . Image to " downOn.png "

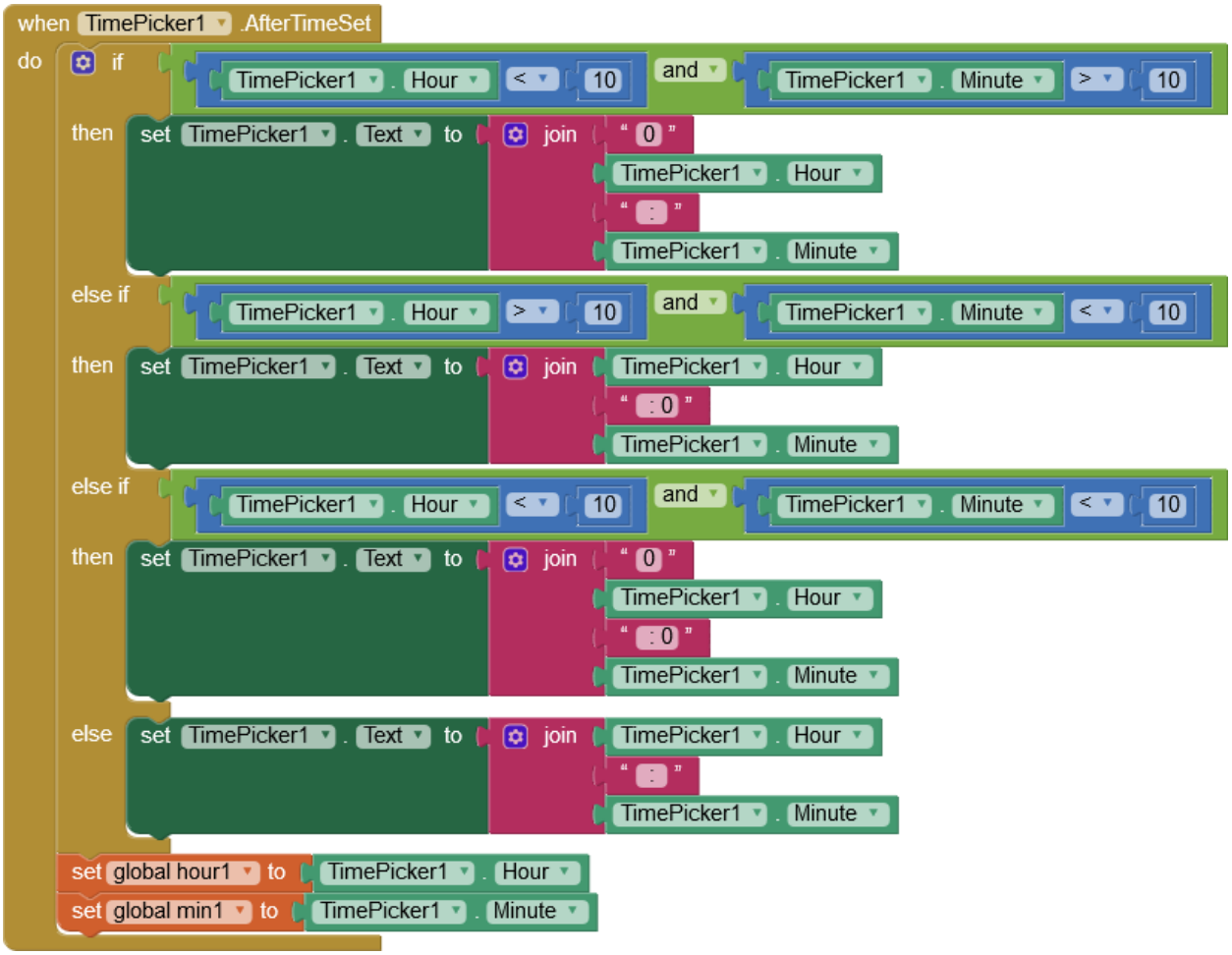

when Button4 .Click
do call BluetoothClient1 .SendText
text " b "
set global switch2 to true
set Button2 . BackgroundColor to
set Button1 . BackgroundColor to
set Button4 . Image to " downoff.png "
set Button3 . Image to " upOn.png "


when Button5 .Click
do call BluetoothClient1 .SendText
text " C "
set global switch3 to false
set Button1 . BackgroundColor to
set Button2 . BackgroundColor to
set Button5 . Image to " upOff.png "
set Button6 . Image to " downOn.png "

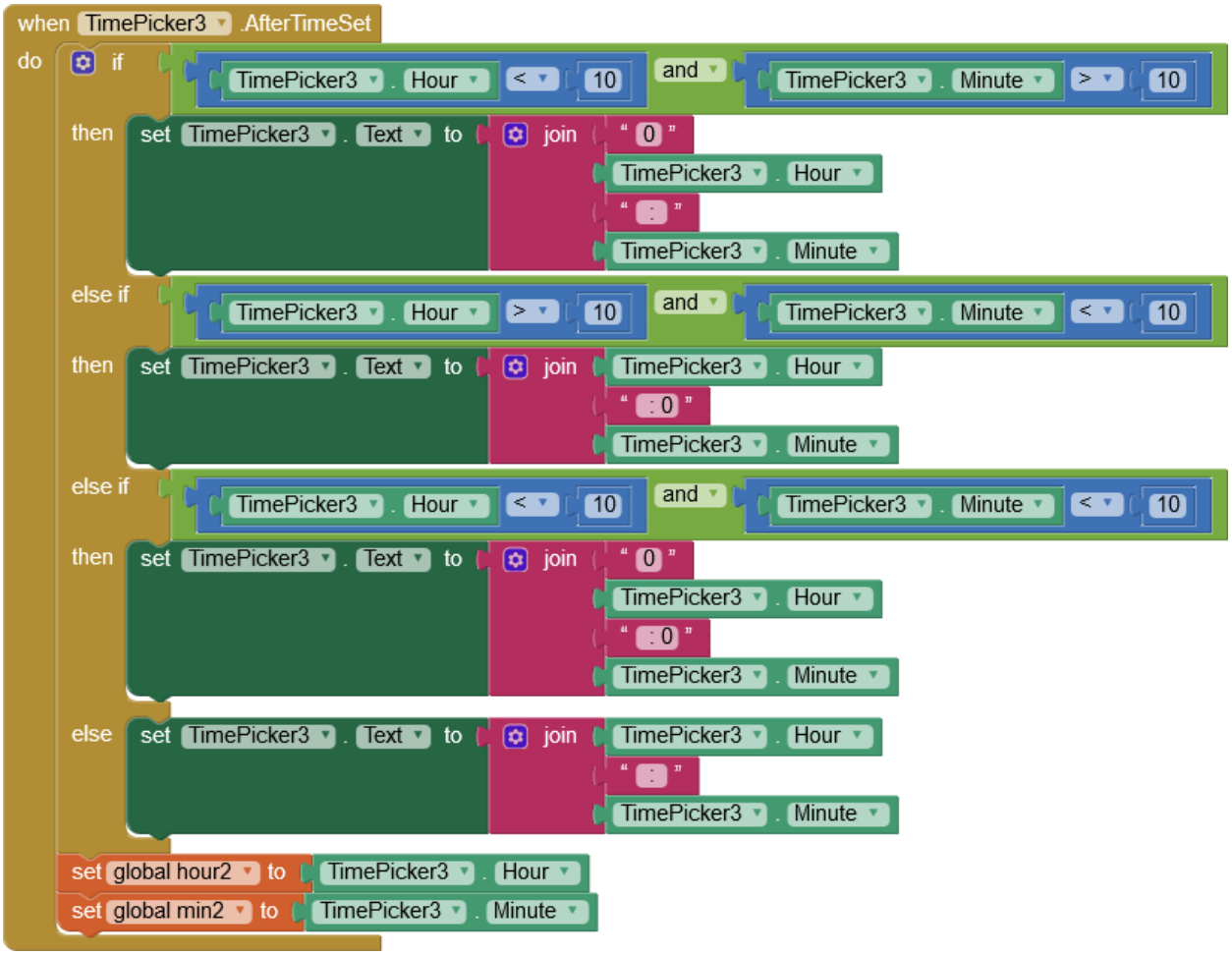

when Button6 .Click
do call BluetoothClient1 .SendText
text " c "
set global switch3 to true
set Button2 . BackgroundColor to
set Button1 . BackgroundColor to
set Button6 . Image to " downoff.png "
set Button5 . Image to " upOn.png "


when Button7 .Click
do call BluetoothClient1 .SendText
text " D "
set global switch4 to false
set Button1 . BackgroundColor to
set Button2 . BackgroundColor to
set Button7 . Image to " upOff.png "
set Button8 . Image to " downOn.png "

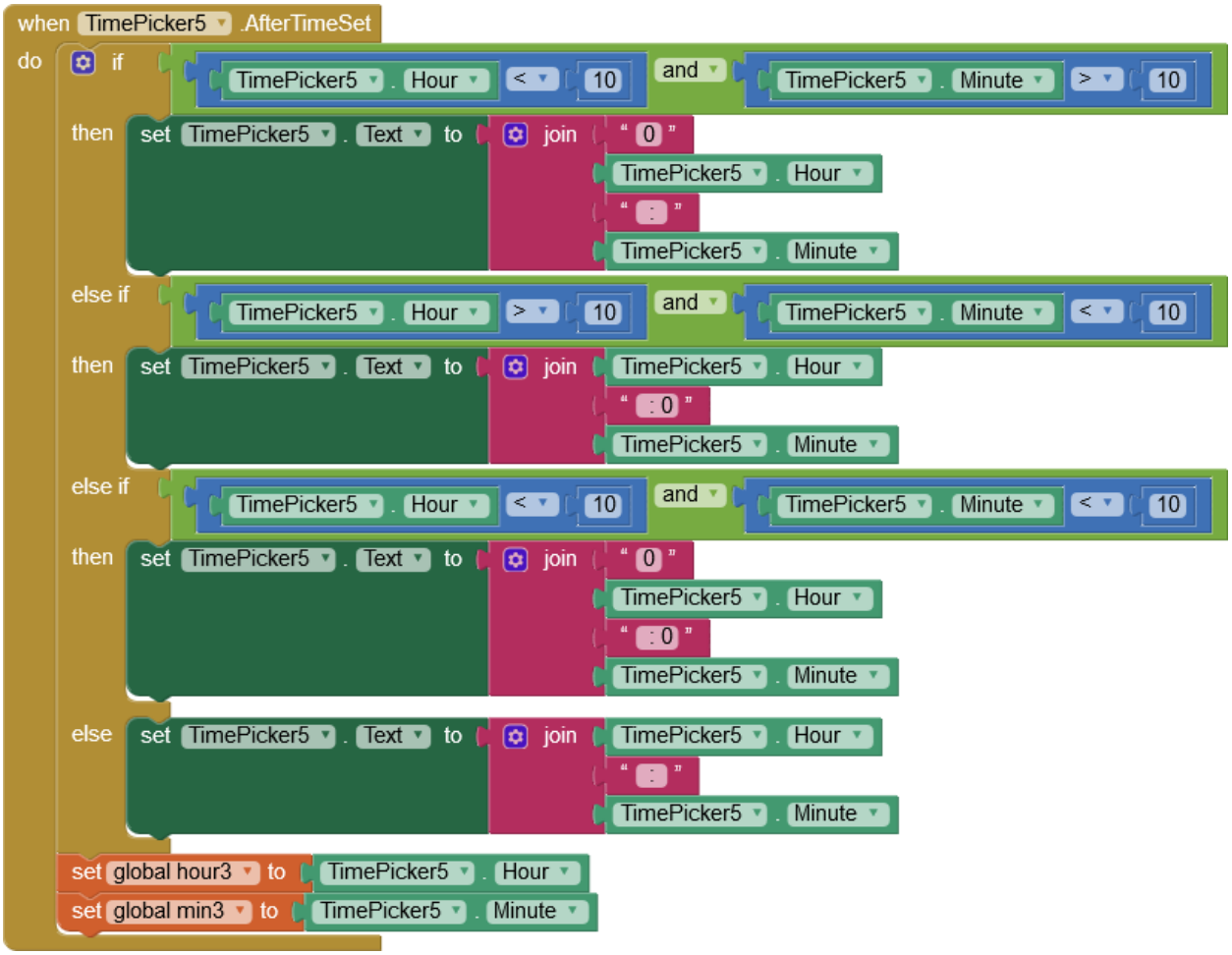

when Button8 .Click
do call BluetoothClient1 .SendText
text " d "
set global switch4 to true
set Button2 . BackgroundColor to
set Button1 . BackgroundColor to
set Button8 . Image to " downoff.png "
set Button7 . Image to " upOn.png "


when TimePicker1 .AfterTimeSet
do if TimePicker1 . Hour < 10 and TimePicker1 . Minute > 10
then set TimePicker1 . Text to join " 0 " TimePicker1 . Hour " : " TimePicker1 . Minute
else if TimePicker1 . Hour > 10 and TimePicker1 . Minute < 10
then set TimePicker1 . Text to join TimePicker1 . Hour " : 0 " TimePicker1 . Minute
else if TimePicker1 . Hour < 10 and TimePicker1 . Minute < 10
then set TimePicker1 . Text to join " 0 " TimePicker1 . Hour " : 0 " TimePicker1 . Minute
else set TimePicker1 . Text to join TimePicker1 . Hour " : " TimePicker1 . Minute
set global hour1 to TimePicker1 . Hour
set global min1 to TimePicker1 . Minute

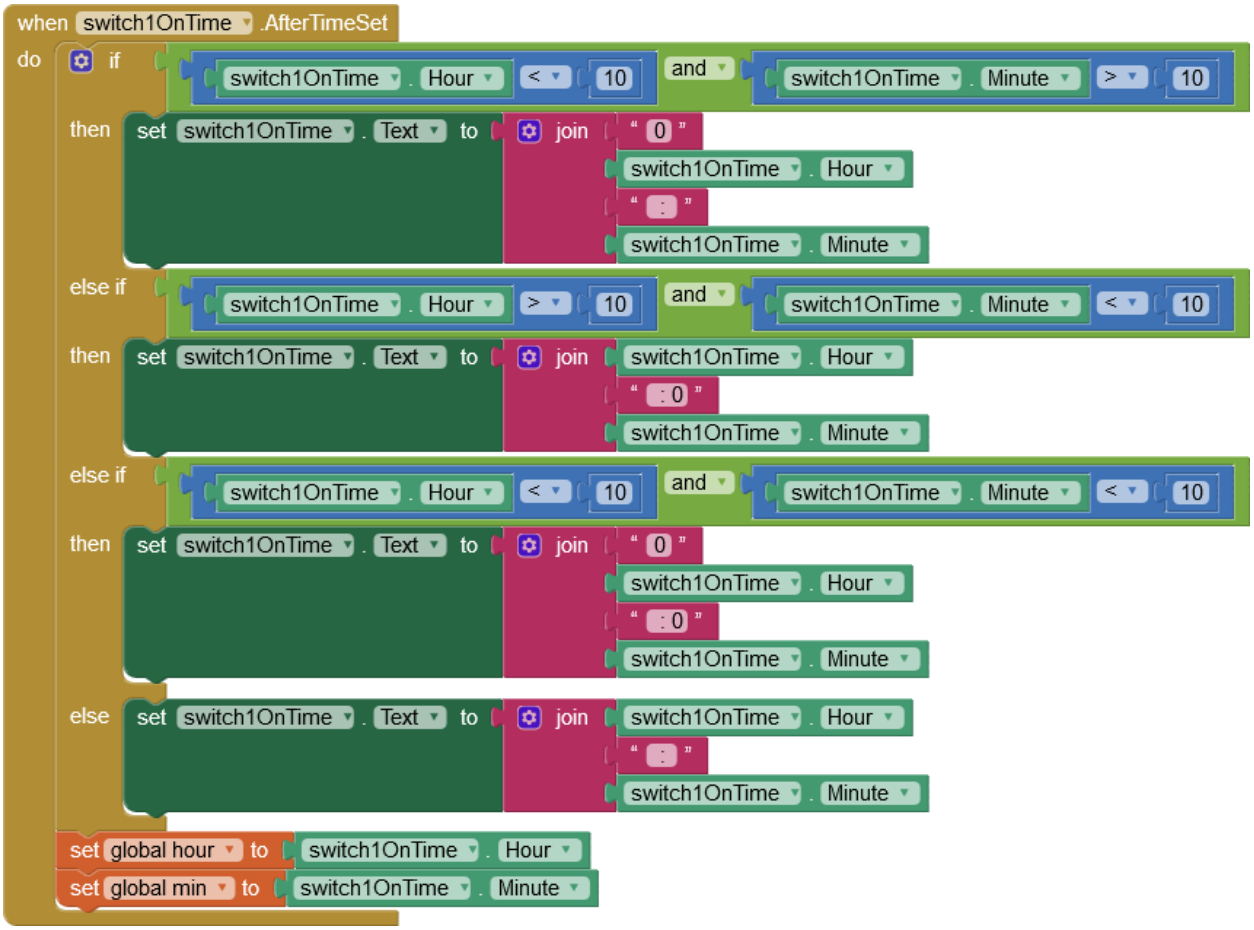

when TimePicker1 .TouchDown
do set TimePicker1 . Text to " 00 : 00 "
set global hour1 to -1
set global min1 to -1


when TimePicker3 .AfterTimeSet
do if TimePicker3 . Hour < 10 and TimePicker3 . Minute > 10
then set TimePicker3 . Text to join " 0 " TimePicker3 . Hour " : " TimePicker3 . Minute
else if TimePicker3 . Hour > 10 and TimePicker3 . Minute < 10
then set TimePicker3 . Text to join TimePicker3 . Hour " : 0 " TimePicker3 . Minute
else if TimePicker3 . Hour < 10 and TimePicker3 . Minute < 10
then set TimePicker3 . Text to join " 0 " TimePicker3 . Hour " : 0 " TimePicker3 . Minute
else set TimePicker3 . Text to join TimePicker3 . Hour " : " TimePicker3 . Minute
set global hour2 to TimePicker3 . Hour
set global min2 to TimePicker3 . Minute

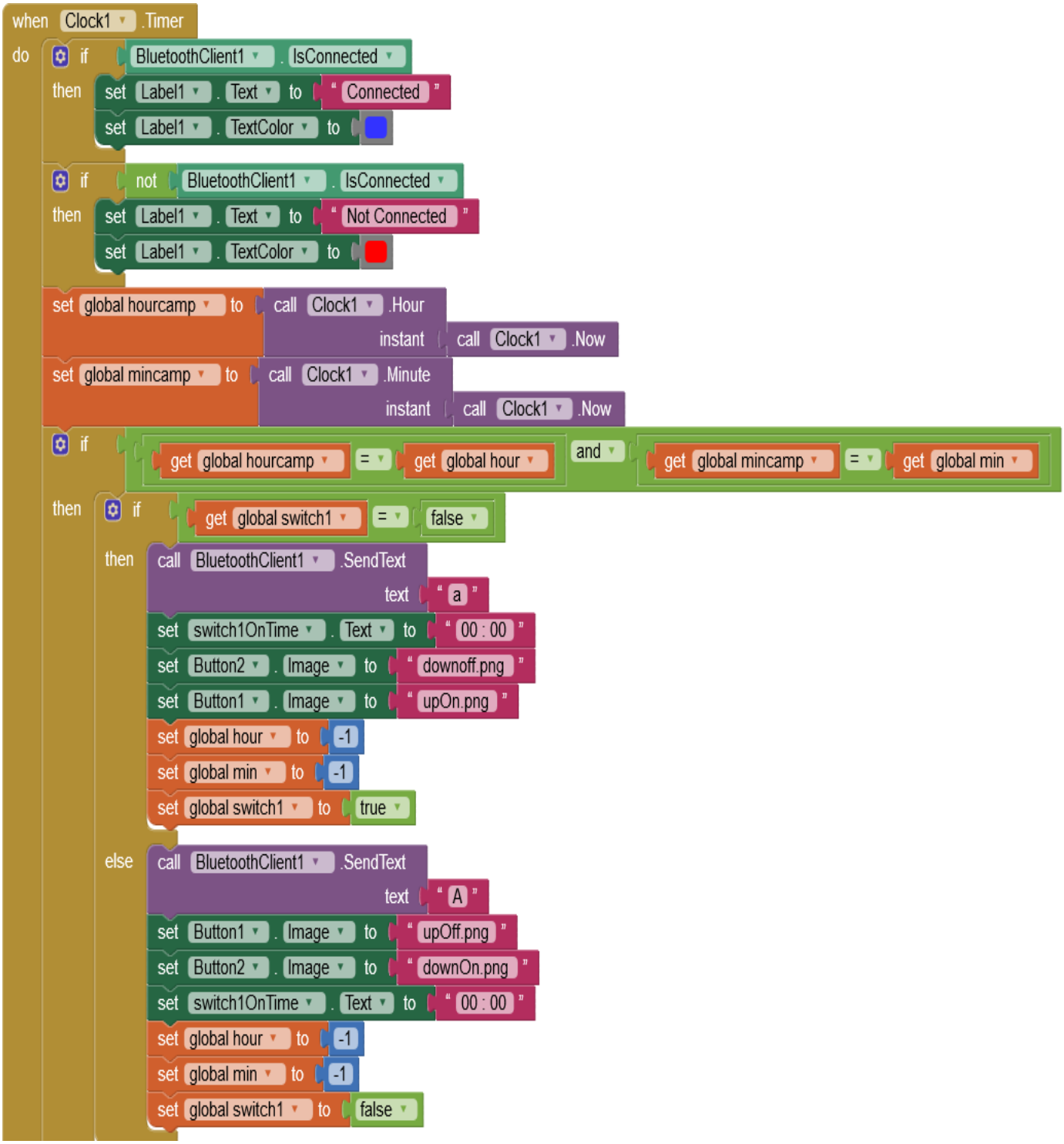

when TimePicker3 .TouchDown
do set TimePicker3 . Text to " 00 : 00 "
set global hour2 to -1
set global min2 to -1

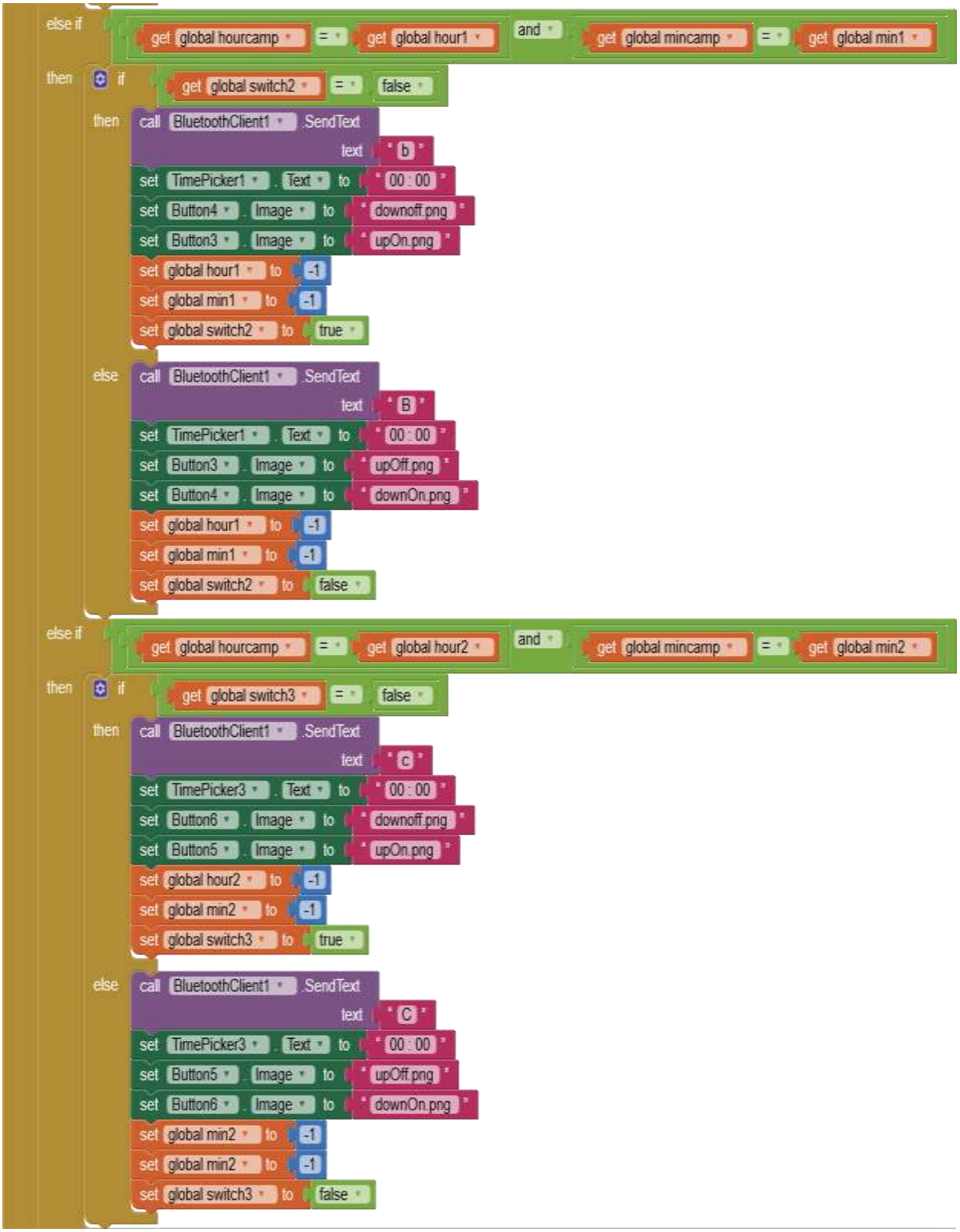

when TimePicker5 .AfterTimeSet
do if TimePicker5 . Hour < 10 and TimePicker5 . Minute > 10
then set TimePicker5 . Text to join " 0 " TimePicker5 . Hour " : " TimePicker5 . Minute
else if TimePicker5 . Hour > 10 and TimePicker5 . Minute < 10
then set TimePicker5 . Text to join TimePicker5 . Hour " : 0 " TimePicker5 . Minute
else if TimePicker5 . Hour < 10 and TimePicker5 . Minute < 10
then set TimePicker5 . Text to join " 0 " TimePicker5 . Hour " : 0 " TimePicker5 . Minute
else set TimePicker5 . Text to join TimePicker5 . Hour " : " TimePicker5 . Minute
set global hour3 to TimePicker5 . Hour
set global min3 to TimePicker5 . Minute

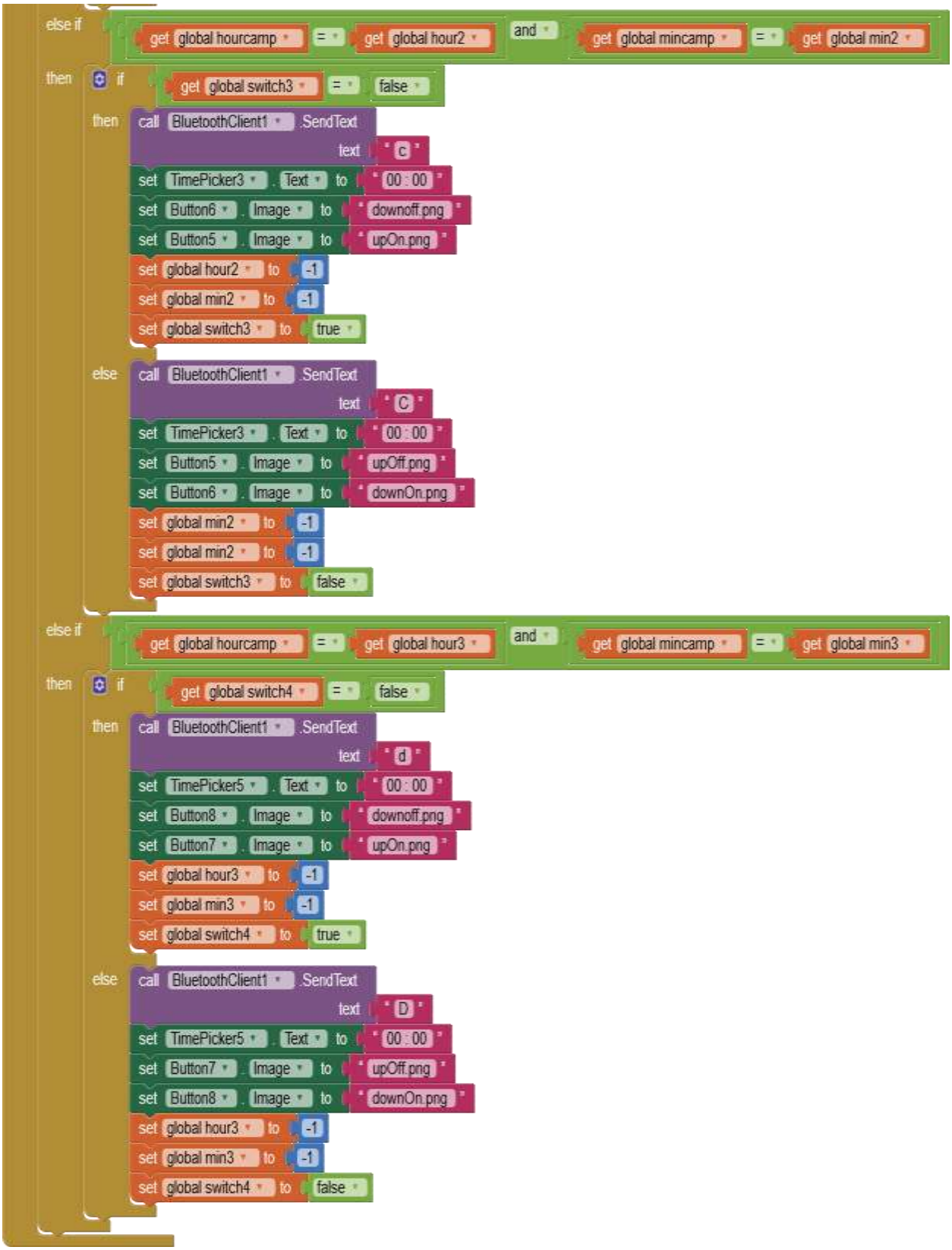

when TimePicker5 .TouchDown
do set TimePicker5 . Text to " 00 : 00 "
set global hour3 to -1
set global min3 to -1


when switch1OnTime .AfterTimeSet
do if switch1OnTime . Hour < 10 and switch1OnTime . Minute > 10
then set switch1OnTime . Text to join " 0 " switch1OnTime . Hour " : " switch1OnTime . Minute
else if switch1OnTime . Hour > 10 and switch1OnTime . Minute < 10
then set switch1OnTime . Text to join switch1OnTime . Hour " : 0 " switch1OnTime . Minute
else if switch1OnTime . Hour < 10 and switch1OnTime . Minute < 10
then set switch1OnTime . Text to join " 0 " switch1OnTime . Hour " : 0 " switch1OnTime . Minute
else set switch1OnTime . Text to join switch1OnTime . Hour " : " switch1OnTime . Minute
set global hour to switch1OnTime . Hour
set global min to switch1OnTime . Minute

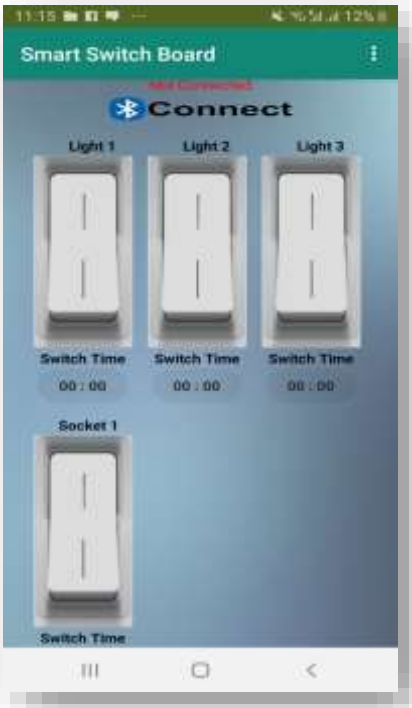

when switch1OnTime .TouchDown
do set switch1OnTime . Text to " 00 : 00 "
set global hour to -1
set global min to -1


when Clock1 .Timer
do if BluetoothClient1 . IsConnected
then set Label1 . Text to " Connected "
set Label1 . TextColor to
if not BluetoothClient1 . IsConnected
then set Label1 . Text to " Not Connected "
set Label1 . TextColor to
set global hourcamp to call Clock1 .Hour
instant call Clock1 .Now
set global mincamp to call Clock1 .Minute
instant call Clock1 .Now
if get global hourcamp = get global hour and get global mincamp = get global min
then if get global switch1 = false
then call BluetoothClient1 .SendText
text " a "
set switch1OnTime . Text to " 00 : 00 "
set Button2 . Image to " downoff.png "
set Button1 . Image to " upOn.png "
set global hour to -1
set global min to -1
set global switch1 to true
else call BluetoothClient1 .SendText
text " A "
set Button1 . Image to " upOff.png "
set Button2 . Image to " downOn.png "
set switch1OnTime . Text to " 00 : 00 "
set global hour to -1
set global min to -1
set global switch1 to false


else if
get global hourcamp = get global hour1 and get global mincamp = get global min1
then if get global switch2 = false
then call BluetoothClient1 .SendText
text " b "
set TimePicker1 . Text to " 00 : 00 "
set Button4 . Image to " downoff.png "
set Button3 . Image to " upOn.png "
set global hour1 to -1
set global min1 to -1
set global switch2 to true
else call BluetoothClient1 .SendText
text " B "
set TimePicker1 . Text to " 00 : 00 "
set Button3 . Image to " upOff.png "
set Button4 . Image to " downOn.png "
set global hour1 to -1
set global min1 to -1
set global switch2 to false
else if
get global hourcamp = get global hour2 and get global mincamp = get global min2
then if get global switch3 = false
then call BluetoothClient1 .SendText
text " c "
set TimePicker3 . Text to " 00 : 00 "
set Button6 . Image to " downoff.png "
set Button5 . Image to " upOn.png "
set global hour2 to -1
set global min2 to -1
set global switch3 to true
else call BluetoothClient1 .SendText
text " C "
set TimePicker3 . Text to " 00 : 00 "
set Button5 . Image to " upOff.png "
set Button6 . Image to " downOn.png "
set global min2 to -1
set global min2 to -1
set global switch3 to false


else if
get global hourcamp = get global hour2 and get global mincamp = get global min2
then if get global switch3 = false
then call BluetoothClient1 .SendText
text " c "
set TimePicker3 . Text to " 00 : 00 "
set Button6 . Image to " downoff.png "
set Button5 . Image to " upOn.png "
set global hour2 to -1
set global min2 to -1
set global switch3 to true
else call BluetoothClient1 .SendText
text " C "
set TimePicker3 . Text to " 00 : 00 "
set Button5 . Image to " upOff.png "
set Button6 . Image to " downOn.png "
set global min2 to -1
set global min2 to -1
set global switch3 to false
else if
get global hourcamp = get global hour3 and get global mincamp = get global min3
then if get global switch4 = false
then call BluetoothClient1 .SendText
text " d "
set TimePicker5 . Text to " 00 : 00 "
set Button8 . Image to " downoff.png "
set Button7 . Image to " upOn.png "
set global hour3 to -1
set global min3 to -1
set global switch4 to true
else call BluetoothClient1 .SendText
text " D "
set TimePicker5 . Text to " 00 : 00 "
set Button7 . Image to " upOff.png "
set Button8 . Image to " downOn.png "
set global hour3 to -1
set global min3 to -1
set global switch4 to false

User/Controller App’s



Card Board Model for demonstration purpose

7. Validation Checks

The term validation means checking. Whenever any person is going to develop any kinds of technology. There are many tools and organization, which are working to validate the work, which any one did. The purpose of validation means to check the source code of any project and determines the formatting and standard, which must be followed during the development of the project. When the project is not following the standards and formatting according to the standards of the organization. The validation of the project would have failed and it will not validate.

Validation can help to get better in troubleshooting. When there are, any errors found in the code of the project it would affect the performance and make a big impact on the project. If the code is found invalid, it means that the project does not follow the rules and the project may not be validating.

Validation Practices

When the project having the standard and compliant code is the best practice for the validation. It encourages the best practices. While most of the organization are going to create the error-free, code and make few validation errors, and most of the developer make more errors. The validation practices can help most of the developer that learn from their mistakes.

Multiple Device Validation

According to our project, it is the most important validation. If we compare recent time with the last 15 years, there are big changes come in the technology field. In our project, we remove the device validation check. Because most of the users are going to use different devices like smartphones, tablets and much newer technology and they are accessing the internet through these different devices. Our project is related to the IoT based, so any users can access Smart Switchboard from any Android device and anywhere but he/she must have access to the Bluetooth enabled Smartphone.

- **User App Validation Checks**

  - Android Phone is required
  - Enable Bluetooth is required
  - HC-5 Bluetooth Module is required in paired list
  - Bluetooth connectivity must be in range
  - Accurate Time zone is required for Switch Timer
  - Android 4.1 or above is required for App to run
  - User should exist

- **Physical Module Validation Checks**

  - AC Power Adapter Regulated 9 Volt 2 Ampere (2.1mm x 5.5mm Barrel) Charger is required for Arduino Board Power Supply
  - The DC (Direct Current) Adaptor is required not AC (Alternating Current)
  - The operating voltage between 9V and 12V recommended
  - Make sure all the components are integrated properly through jumper wires
  - HC-5 Bluetooth Module is required in range
  - Relay Board should be Attached to the 230VAC Power Supply properly
  - Make sure inside of Smart Switchboard is clean for MQ-2 Sensor
  - All the physical Switches must work in 2-way mechanism

8. IMPLEMENTATION AND MAINTENANCE

8.1 Implementation

Implementation is the stage where the theoretical design is change to the practical system. First, we are going to gather the requirements. Before starting the implementation all, the requirements about your project must know. After the requirements gathering, we are going for the designing stage of our project where we are going to make the theoretical design of our project. We specify all the modules of our project and their relationship between the modules. Once the designing stage is complete then we start the implementation stage. In implementation, we are going to start the coding part of our project. In implementation, we convert the theoretical design of our project to the practical system. It is the most critical stage in which a new system is going to implement for achieving the goal successfully and it will be confirming that the system will work effectively. In this stage, we give the new system to the user or client for practical use.

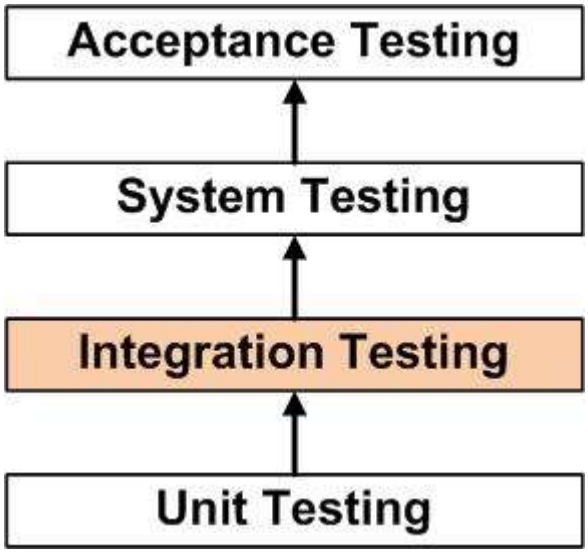
8.2 Maintenance

After implementation stage, the next phase is the maintenance stage. In maintenance, we are going to take care of the system. When the system is given to the client for the practical use, after some times the user finds some bugs or the system is not capable to provide the desirable output for the client. The system is going to the maintenance stage. The developer tests the system again if it is possible that the bugs are solve in maintenance. If the bugs are not solving in maintenance, then the developer again go back to the designing stage. They are working on the designing stage once they fix the error in that stage. After fixing it there then again, the implementation stage is going to start and they bring changes in the system. After completing the implementation, the system in given to the client for the practical use. After some time, the system will need some updates or upgrade so in maintenance stage all these issues are going to be solve in the future.

9. SYSTEM TESTING

The main objective of testing is to find the bugs within the system. Testing is a lengthy and costly phase of system development. The system must be tested whether the user requirements have been full filled or not? whether the expected outputs of the system are the same as the actual outputs or not?

It takes different aspect in consideration like checking, the functionality of different modules in the system the, dependency of each module, the overall interoperability of subunits. Initially, we start testing process from unit testing, then we integrate the units and performing integration testing later on functional testing at the end testing process is going to end with overall system testing.



Definition by ISTQB (International Software Testing Qualifications Board)

There are different methodologies of testing a work product and types of the testing process on work product.

9.1 Unit testing

In unit testing, the testing is done by the developer itself rather than tester because an individual component or module of the system/software will be tested. It will perform basic tests on the small parts of the system whether its functions are accurate or not. That is why it is done by the developer he produces the system unit by unit.

9.2 Integration testing

After each integration of unit, the system should be tested to verify the system is working as one piece or not. The objective of this test is to find out the fault in communication or interaction among different modules of the system.

9.3 Functional testing

In the functional testing phase, the functional requirements of the are going to check. In this part testing, we work on test design. We are going to design test cases and test suits. Like:

Functional testing is determined on the basis of the below items:

Valid Input : classes for the valid must be defined and accepted.

Invalid Input: classes for the invalid must be defined and rejected.

Functions : functions must be well defined and exercised.

Output : classes of application outputs identified must be exercised.

Systems/Procedures: interfacing procedures or systems must be called.

9.4 System Test

In this phase of testing the tester ensures the entire system comprises of different parts or modules meet the requirements. It tests the complete functionality of the system to ensure known and predictable outputs.

9.5 Acceptance Testing

It is the last stage of testing which is done by the clients or the user of the system. It verifies whether the business requirements are fulfilled or not. The clients will accept the system or software only when all the functions and features are working as it was expected.

9.6 Project Testing Overview

1. After we have completed the designing and coding phases, we start through the project testing. In which we followed all the above testing procedures.
2. We start from the User App Buttons are working or not. Whether all the buttons of the App are? We did the same for all the available user controls on the Application.
3. We checked all the Switch timer can set the time properly and can it trigger action on the specific or not.
4. We've tested whether on touch down of Switch timer will it reset the switch timer or not.
5. We've tested the Bluetooth connectivity is it working in range or not.
6.
7. We've tested the Smart Switchboard whether is it working in 2-way mechanism or not.
8. We've insured in case of power failure/down the board should reset all relays to down state.
9. We've insured 2-way switching mechanism from both Physical Model and User controller App.
10. We've insured the reliability of MQ-2 smoke/fire sensor.
11. We've insured the in case of fire/smoke inside Smart Switchboard the power supply to all Home Appliances should be down.
12. We have done the Compatibility testing or responsiveness of the User App how it will look like in different Android devices screen view.
13. We have tested the overall functionality of the Smart Switchboard whether it will work as we have expected in the design phase or not.

10. SYSTEM SECURITY MEASURES

As the Smart Switchboard project is IoT based in which involvement of user data and privacy is not involved so It decreases the security risks to the potential of losing user data and privacy.

Similarly, the User App does not require any advance or user related permission like access to the contacts or phone storage, photo gallery however it requires only Bluetooth permission which is an encrypted technology can't be interrupted easily.

Nevertheless, they are safety concerns due to the dependency of the project on the Alternate Current or we can electricity so we have to follow all safety guidelines and measures related to the electricity.

Concerns regarding Smart Switchboard

Smart Switchboard need to be installed instead of your traditional switches, which require a little knowledge and work with electrical wiring. It is not easy for many consumers to go by the installation of a Smart Switchboard and it requires expertise.

Smart Switchboard must be used for devices/appliances that fall within the specified rated voltage and power limits. Installation must be done in cool and dry places. Moisture may cause malfunctioning of the Board.

11. FUTURE SCOPE OF THE PROJECT

The idea of having a smart home dates back centuries. It can be said that human beings have been thinking about a comfortable life since they entered this world. They worked hard for this convenience until he invented the smartphone. Since then, the world has undergone dramatic changes that we can hear with our own fingers. Additionally, we can see dramatic changes in recent years in this field.

The future of Home Automation is Broad so is the future of Smart Switchboard. It saves energy and economy and many believe in saving. Smart Switchboard is a convenient choice for those who think smart, want more control and customization over their electronic appliances.

It will also help those who are physically challenged in doing repetitive tasks which disturbs them by asking help like switching on or off light bulb, fan or other electrical gadgets to stop them from over charging. Finally, we can see the broader picture of the Smart Switchboard.