St. Xavier's College

(Autonomous)

Department of Information Technology

Project Report

Jarvis

A Home Automation System

Submitted By

Jobin Lawrance

135069

Joanna Cerejo

135098

TY IT

March 2016

DECLARATION

We, Joanna Cerejo - UID no: 135098 and Jobin Lawrance -UID no: 135069 hereby declare that this project report titled:

"Jarvis-An Home Automation System"

which is being submitted in partial fulfillment of the Bachelors in Information Technology Examination conducted by St Xavier's College Mumbai (autonomous under Mumbai University), is the result of the work carried out by us under guidance of Prof. Shivranjani Gudibanda of St Xavier's College, Mumbai.

Joanna Cerejo

Jobin Lawrance

St. Xavier's College (Autonomous)

Mumbai – 400 001.

DEPARTMENT OF INFORMATION TECHNOLOGY CERTIFICATE

This is to certify that the project titled

"Jarvis-An Home Automation System"

undertaken by

Jobin Lawrance(UID:135069)

&

Joanna Cerejo(UID:135098)

has been submitted in partial fulfilment of the BSC IT degree.

It is further certified that they have completed all the required phases of the project.

Signature	Signature	Signature
(Project Guide)	(Internal Examiner)	(External Examiner)

Signature

Head of Department(IT) (College Seal)

Acknowledgements

We would like to take this opportunity to thank everyone who supported us throughout the course of this project. We are thankful for their aspiring guidance, invaluable constructive criticism and friendly advice during the project work.

We would also like to thank our project guide - ***Mrs. Shivranjani Gudibanda****, Prof. of Department of Information Technology, St Xavier's College - for her advice,wholehearted cooperation and constant motivation, without which this project would not have seen the light of day.*

We would also like to specially thanks ***Mr. Cecil Cerejo*** *for all the financial Assistance and* ***Mrs. Binu Lawrance*** *for all the support throughout the course of the project.*

Finally, we would like to thank our families who supported us through all the work, the late nights and who have been an important and indispensable source of spiritual support.
We are grateful to St. Xavier's College that has given us so much these three years. Thank you

Table of Contents

Jarvis

Preface

This project aims at the introduction to a Home automation system.This report contains the basic logic used for software and hardware development along with the diagrams so that the logic may be apprehended without difficulty. For detail information, screen layouts provided with the report can be viewed.

Although this report is prepared with utmost care, the project is subjected to further enhancement. The screen shots of various components may be different to the final product.

The various modules have been coded in the following components
C/C++, Java, Xml, Html, Css, JQuery, Ajax, Php, MySql.

Introduction

Houses of the 21st century will become more and more self-controlled and automated. Simple devices such as a timer to turn on one’s coffee maker in the morning have been around for many years, but much more sophisticated mechanisms will soon be prevalent in houses around the world. Imagine walking into your home and being greeted at the door with lights illuminating your path without you ever having to touch a lightswitch, with your favorite music streaming through the speakers in whichever room you enter (because your home recognized that it was you and not some other household member), all while having the peace of mind knowing that your home automation system took care of activating your security system. Furthermore, such a system could allow the user to schedule events to occur at recurring intervals (e.g., turn on light system at 4:30a.m. every Tuesday and Thursday). This report describes an approximation of such a home automation system that was designed and built as a final project for 6th semester at St Xavier's College.

This report documents the development of a range of systems that enable home automation systems to be integrated and exposed to the Web.

To achieve these goals, the system runs a user-defined program on a special-purpose processor, using real-world sensor inputs as operands. The sensors and other input mechanisms along with user-programmable event schedules allow the user to adjust and customize the home environment. Using sensors that measure temperature and infrared commands,this system will create a safe home atmosphere.

Design decisions, implementation details, and testing procedures are thoroughly discussed, and the resulting functional system is described.

Objective

The objective of the project is given below

- to control basic home appliances like fans,lights etc from inside and outside the house using internet.
- to allows multiple users to control the house while retaining the main control for single admin.
- to monitor the in-house temperature over a period of time.
- to secure the house.
- to monitor the appliance usage time, so that better power saving decisions can be made.
- to have an AI module that can control all of the above features.

Current Available System and its drawbacks

The current system available in the market that are similar to Jarvis is based on smart houses and is only compatible with smart houses which uses smart devices.Moreover the cost involved into the production, installation and maintenance of these systems is nothing less than ₹ 10,000.
The cheapest competitor to Jarvis is pert.me, which sells at ₹ 6000, but it only offers basic on/off functions for up to 8 devices.

Advantages of the proposed System

The Cost Factor

The cost of making Jarvis is less than ₹ 3000, which is comparatively cheaper than all its competitors with the same features.
Moreover more components can be added on top of the existing ones without inflating the cost at a noticeable level.

Convenience Through App based access to the home automation system

- A forgetful user will no longer have to worry about fan being left on, when he leaves for work. Simply by accessing the smartphone app, the status of the fan can be changed.
- Enabling the security mode allows the user to receive a push notification when any movement is sensed in the house, upon which the user can then alert the police, neighbours etc.
- Monitoring the in-house temperature with graphical representation.
- Directly typing in commands or using voice commands and the AI module executes the commands.
- Rich user-interface

Authentication and sessions maintained:

Different sessions are maintained for different users. All the users of the application are authenticated with use of login. Therefore only authorized users can change the state of the lights and fans within the house and not everyone. Thus increasing the security of the home.

Limitations:

Proprietorship - Many of the systems (TV, stereo, surveillance camera, etc.) mentioned are proprietary and as such each have their own programmatic interface that control them, or none at all. This introduces a high cost due to the amount of work required to generalize these systems. High cost means that home automation is less likely to become a common household system, unfortunate for both home automation vendors and households.

Standardization - To obtain an extensible system of home automation related devices, that system must provide an agreed upon standard for device communication.

System Study and Analysis

Problem Definition

Current Home automation systems face four main challenges, these are high cost of ownership, inflexibility, poor manageability, and difficulty in achieving security. The main objectives of this research is to design and implement a home automation system using Arduino Mega board that is capable of controlling and automating most of the house appliances through an easy manageable interface. The proposed system has a great flexibility by using internet technology to interconnect its distributed sensors to home automation server. This will decrease the deployment cost and will increase the ability of upgrading, and system reconfiguration.

Proposed System Feature

The proposed system is a distributed home automation system, consisting of a server, sensors and a microcontroller. The microcontroller checks the server for the sensor state and subsequently controls and monitors the corresponding sensors, and can be easily configured to handle more hardware modules (sensors) through the mobile application via the internet.

Feasibility Study

A. Technical Feasibility

The technical aspects for the development of the proposed project are well within the project team's capabilities to produce such a product. The project team has experience in all aspects of the technology to be used

C++	Arduino
Java	Android programming
Xml	Android Layout
Html,Css	Web layout and design
JQuery,Ajax	Client side Scripting
Php,	Server Side Scripting
MySql	RDBMS
JSON	Data Transfer from server to client.

The project is developed in such a manner that that we implement web pages within our android application.
The main Advantage of this process is that our application can be accesses on different operating system like the Apple's iOS and not necessarily android.

The web pages are developed with the help of bootstrap.The fonts used on the web application are similar to those used on an android app to give it the look and the feel of the same.

The database used is a MySql server. User documentation is provided for the operation of the database.

B. Economic Feasibility

The following are the costs associated with the project.

1. 1. Direct Costs
    a. Purchase of Arduino board plus the ethernet shield to develop project
    b. Sensors

2. Indirect Costs – NIL

3. Fixed Costs
    a. Server renewals
    b. Upgrade Costs

4. Operational Costs
    a. Maintenance Costs

5. Tangible Costs
    a. Correlates to the entire project expenditure.
    b. Any cost to which a monetary value can be assigned

6. Intangible Costs
    a. Costs if the app has to be published on the play store.
    b. Developing and publishing app on other platforms.
    c. Cost in the case of any system Failures or crashes.

Feasibility Study Report/Conclusion

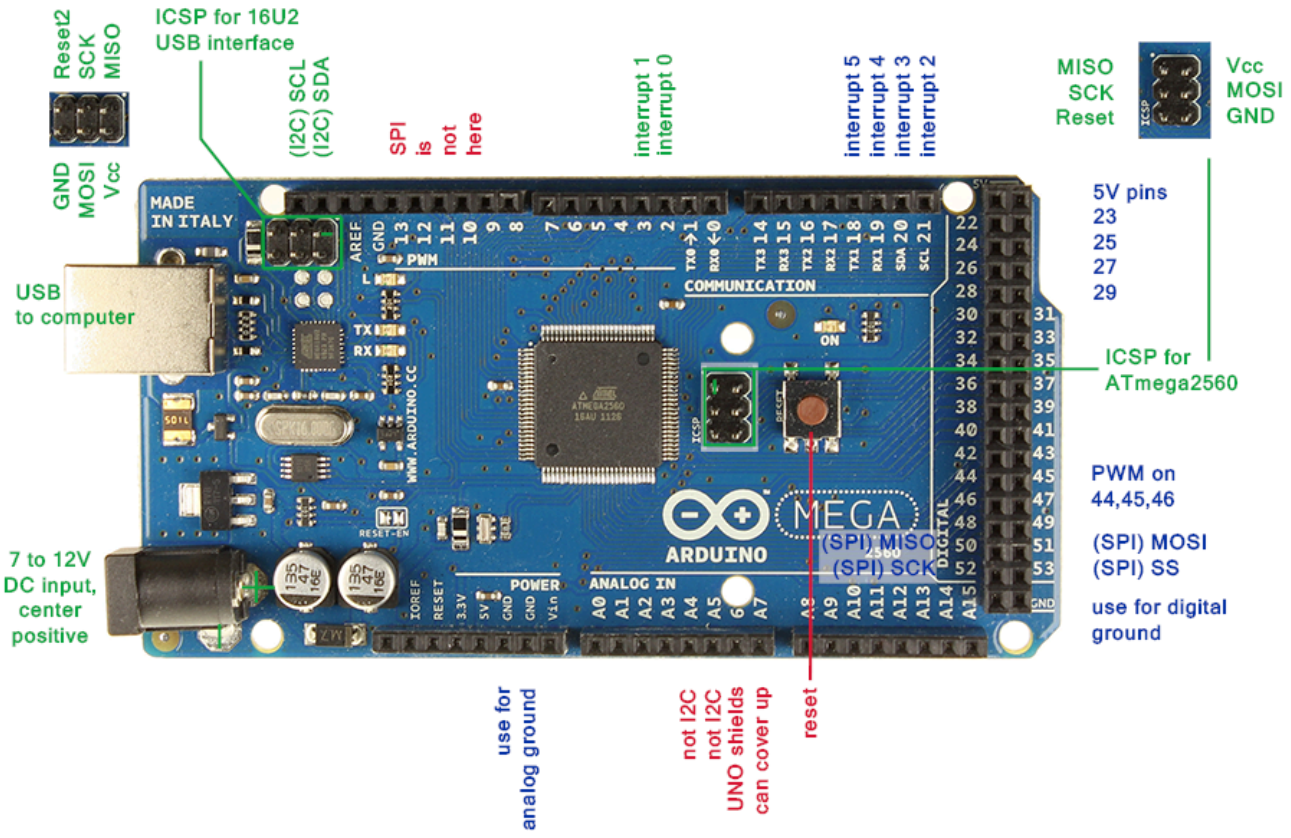
The result of the Feasibility Study provided the following facts:

- The system developed provides a user friendly environment.
- The proposed system should be simple to use, maintainable and incorporate all necessary services.
- All appliances are now accessible and only a click away
- Appliances can be accessed from within as well as from outside their respective homes.

System requirements

Arduino Mega 2560 :

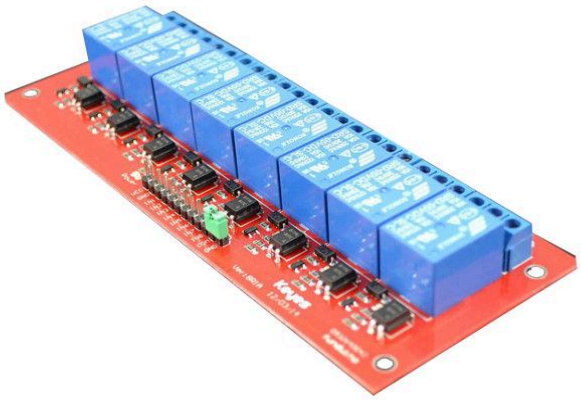
It is a board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.



Arduino Ethernet shield:

The Arduino Ethernet Shield connects your Arduino to the internet in mere minutes. Just plug this module onto your Arduino board, connect it to your network with an RJ45 cable (not included) and follow a few simple instructions to start controlling your world through the internet. As always with Arduino, every element of the platform – hardware, software and documentation – is freely available and open-source. This means you can learn exactly how it's made and use its design as the starting point for your own circuits. Operating voltage 5V (supplied from the Arduino Board)

The Relay board:



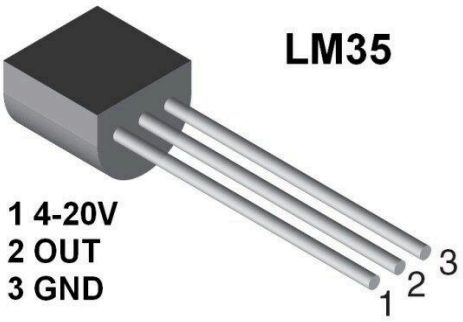
A Relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate the switch and provide electrical isolation between two circuits.
A can be operated by a relatively small electric current that can turn on or off a much larger electric current. The heart of a relay is an electromagnet (a coil of wire that becomes a temporary magnet when electricity flows through it).

Note:

We know that the arduino board can supply only 5v of current which is not sufficient to switch on the lights or fans therefore we need a relay that converts this current to a 220-240 volt to switch on the lights.Hence we use the relay board.

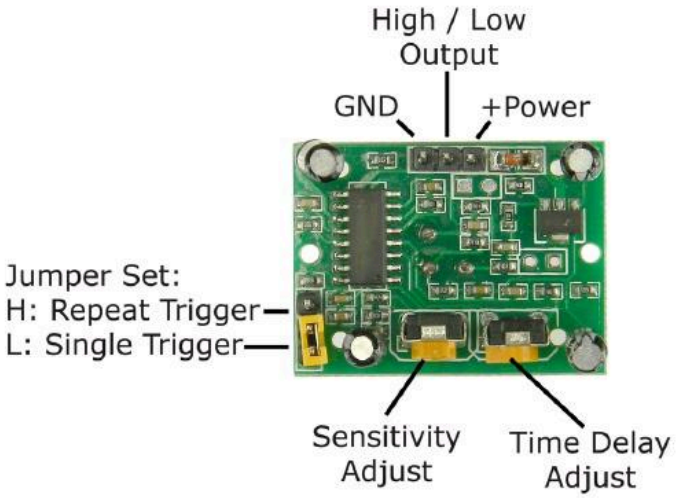
Temperature sensor LM35:

The LM35 is an integrated circuit sensor that can be used to measure temperature with an electrical output proportional to the temperature (in °C)



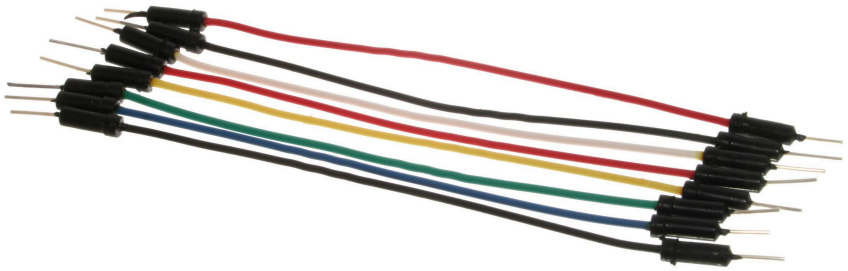
Pir sensor:

A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors.



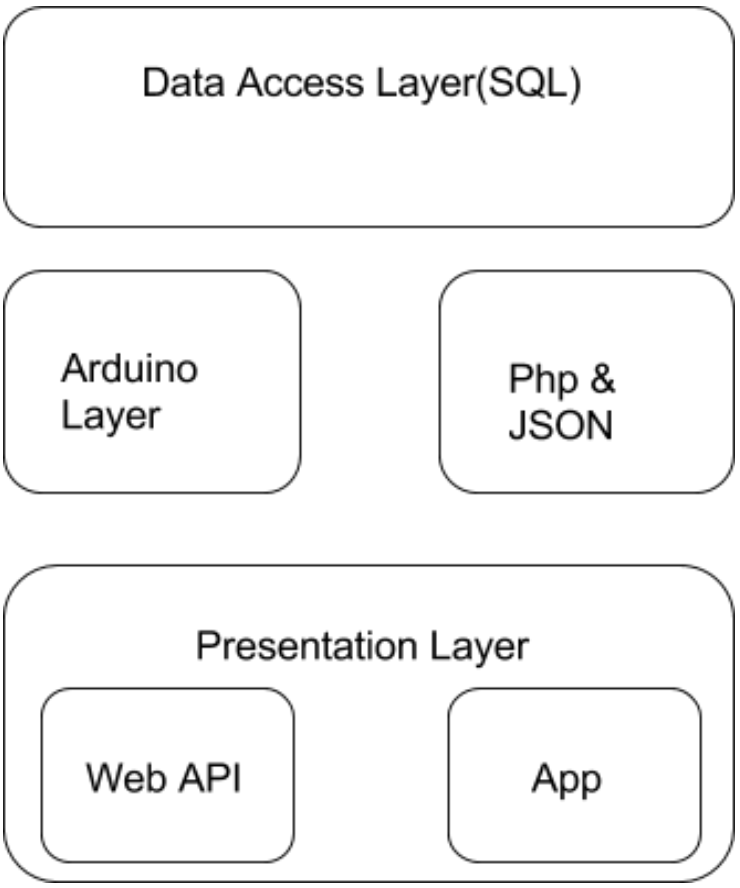
Jumper Wires:

Used to make all the electrical connections i.e from the arduino to all its respective components.



System Design

Architecture:



Data Access Layer:

This layer is where the SQL runs. We create a Database on the server to store all out SQL functions and table. It contains all the database operations and stores various values to maintain our records.

The Php Layer:

This layer is responsible for communicating between the database where our values are stored and the Application.
Whenever commands are fired from the app they are sent to the php pages which then execute the respective JSON,Javascript or the Ajax, thus making changes in the records of our database.

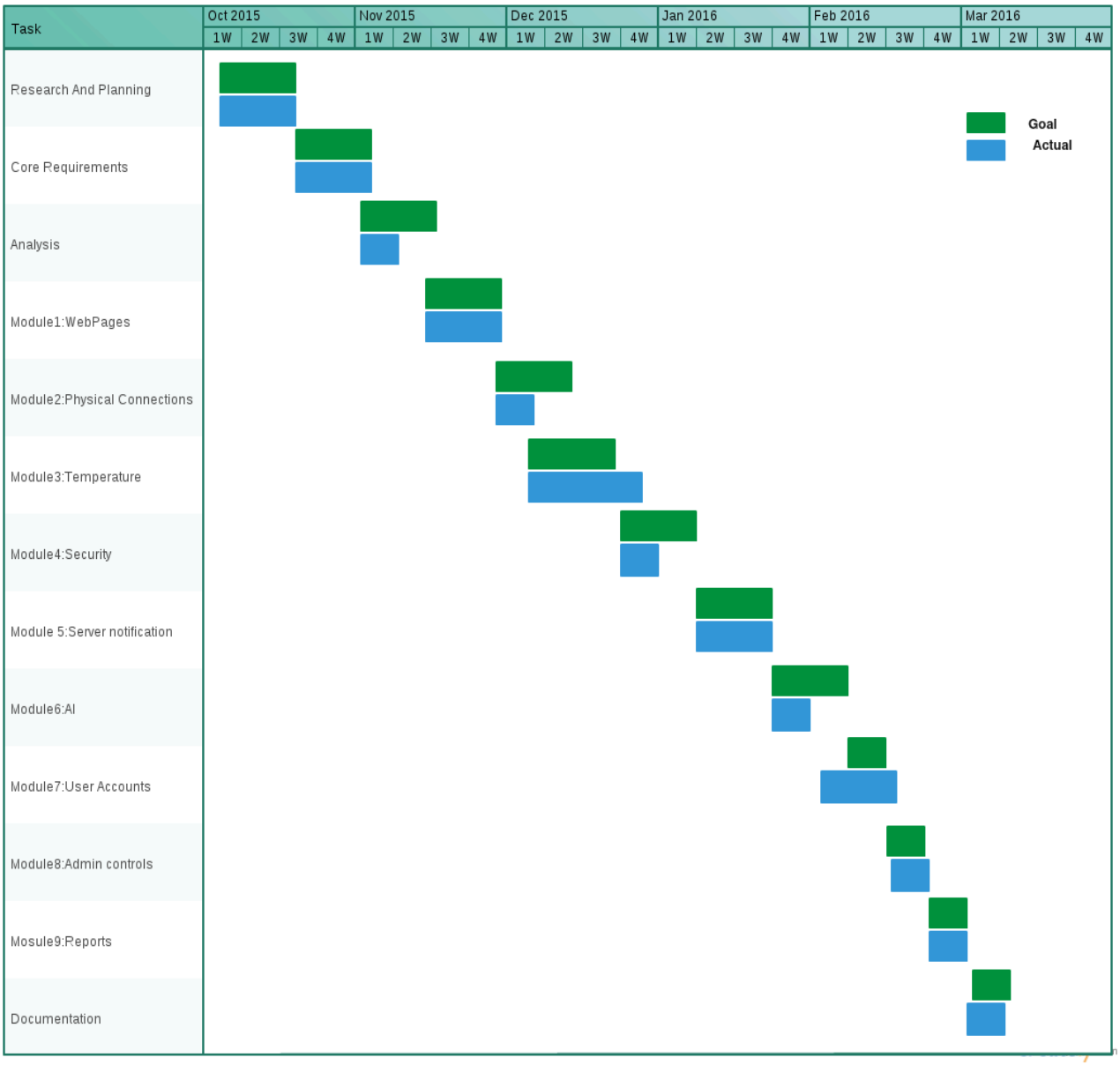
The arduino Layer:

The arduino is responsible to connect to the database within our server and continuously look for changes. This layer is responsible for physically switching on the light.

The Presentation Layer:

This is the top most layer visible to the end users.Through this layer the user interact with our Application and requests are made to the server.
This layer continuously makes use of the php layer for every request and thus dependent on it.

Gantt Chart:

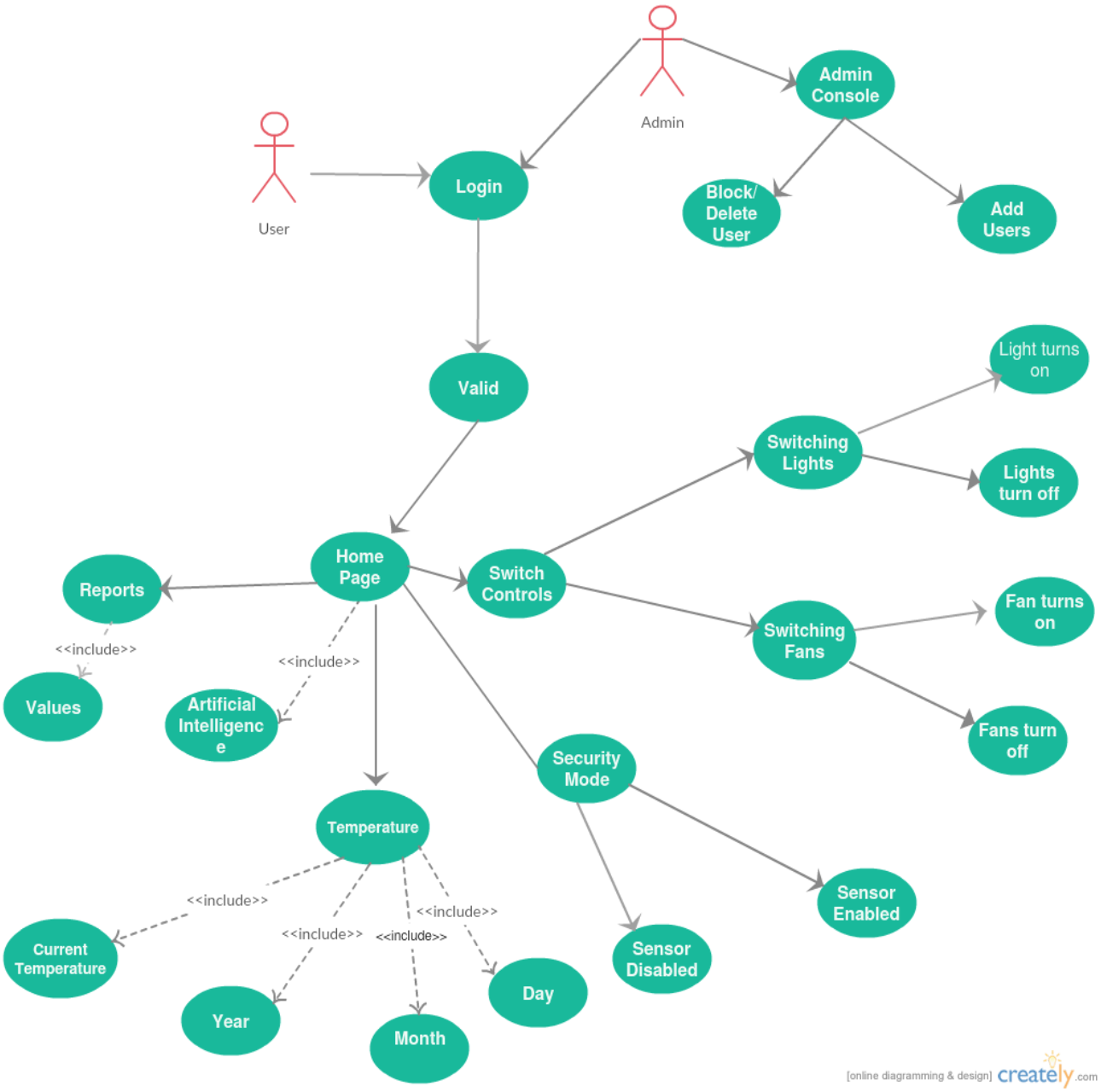

Task
Oct 2015
Nov 2015
Dec 2015
Jan 2016
Feb 2016
Mar 2016
1W 2W 3W 4W
Research And Planning
Core Requirements
Analysis
Module1:WebPages
Module2:Physical Connections
Module3:Temperature
Module4:Security
Module 5:Server notification
Module6:AI
Module7:User Accounts
Module8:Admin controls
Mosule9:Reports
Documentation
Goal
Actual

Event table:

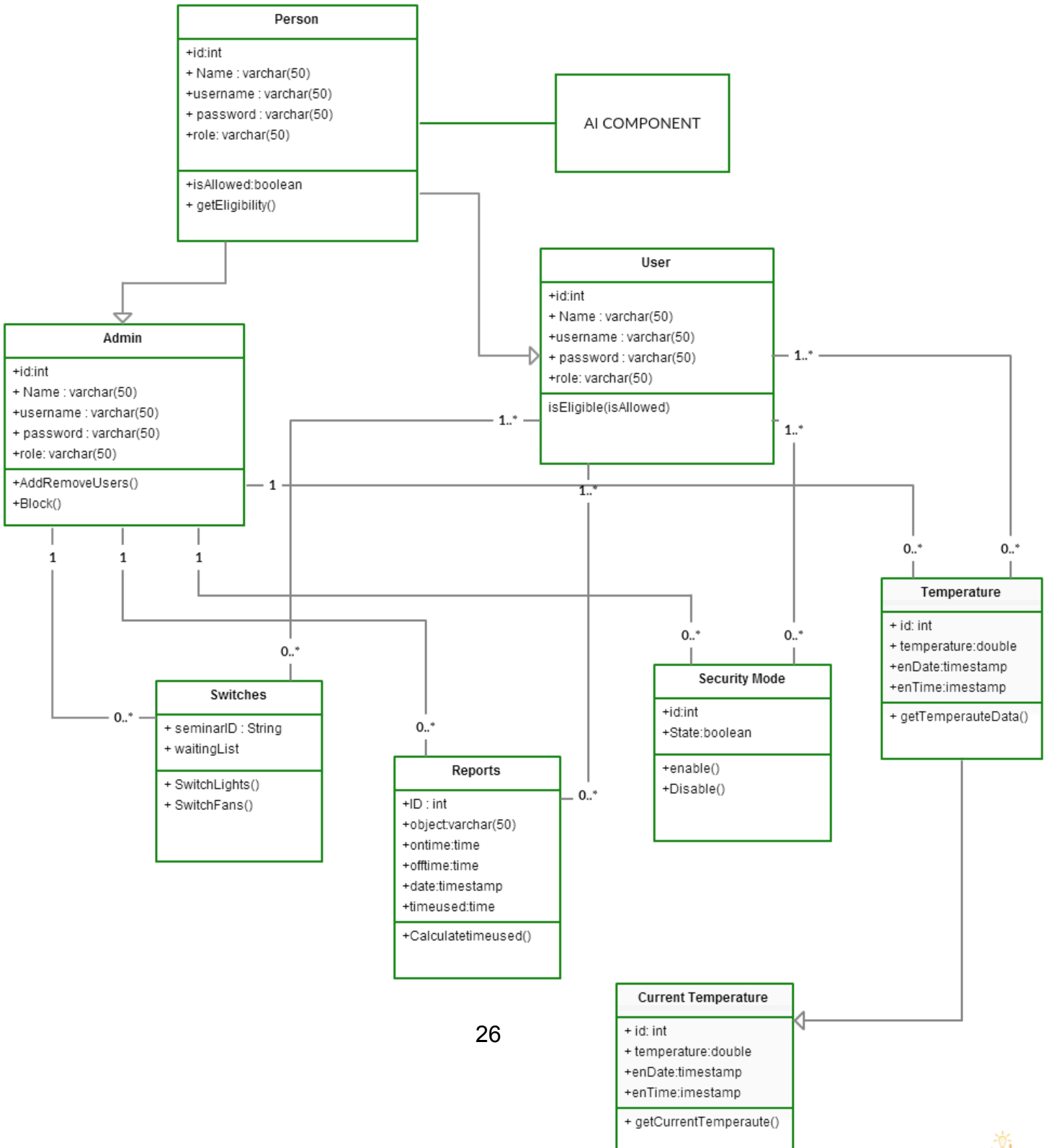
Event	Trigger	Source Actor	Usecase	Response	Destination Actor
User Logins	User enters username and password	User	Login	Valid--HomePage	User
Admin Logins	Admin enters username and password	Admin	Login	Valid--HomePage	Admin
User/Admin switches Lights	enables or disables the switch	User/Admin	Switches Controls	Lights turned on or off	User/Admin
User/Admin switches Fans	enables or disables the switch	User/Admin	Switches Controls	Fans turned on or off	User/Admin
User/Admin switches Security Mode	enables or disables the switch	User/Admin	Security Mode	Security (sensor)turned on or off	User/Admin
User/Admin requests for temperature	Selects temperature	User/Admin	temperature	Graph/Current Temperature	User/Admin

User/Admin sends a voice command	Taps on the mice icon on the main menu	User/Admin	Artificial Intelligence Module	The requested Page is loaded	User/Admin
Admin Adds Users	Selects Add users on the Admin console	Admin	Admin Console	User Added	Admin
Admin Deletes/ Blocks Users	Selects delete/block users on the Admin console	Admin	Admin Console	User deleted or blocked	Admin
User/Admin requests for reports	selects the reports tab	User/ Admin	Reports	Resultant data	User/Admin

Use Case Diagram

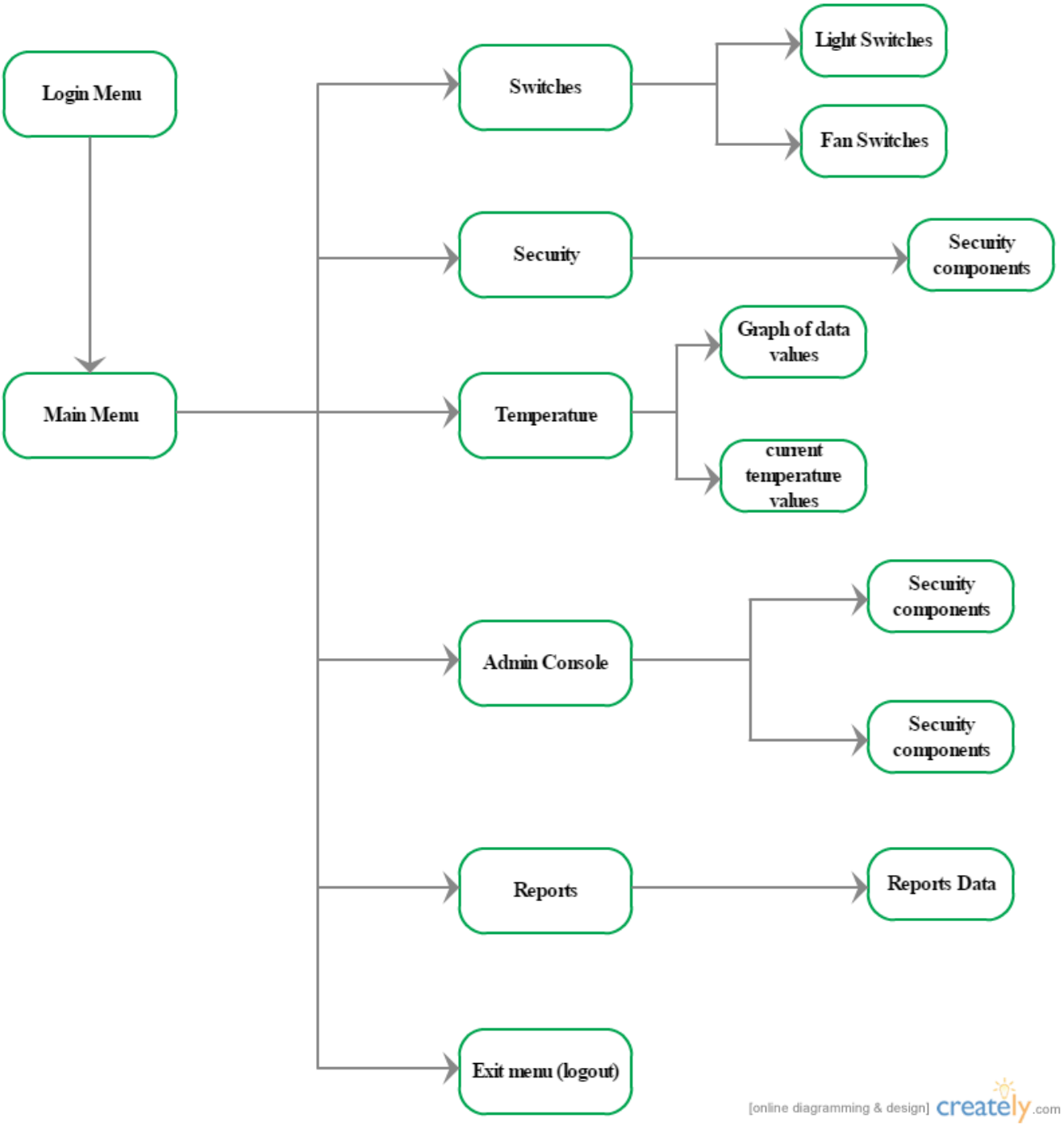


Class Diagram:

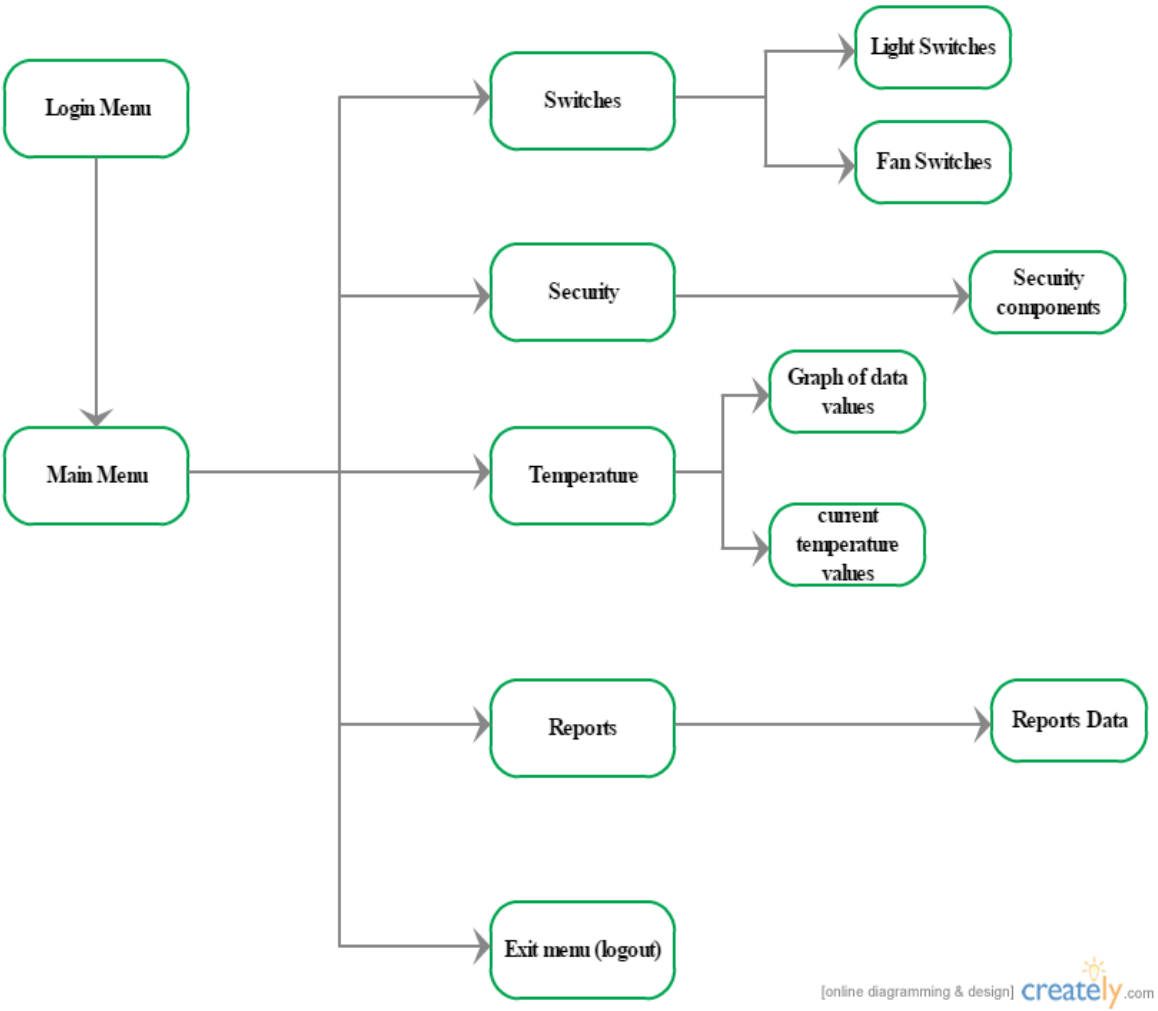


Menu tree Diagram

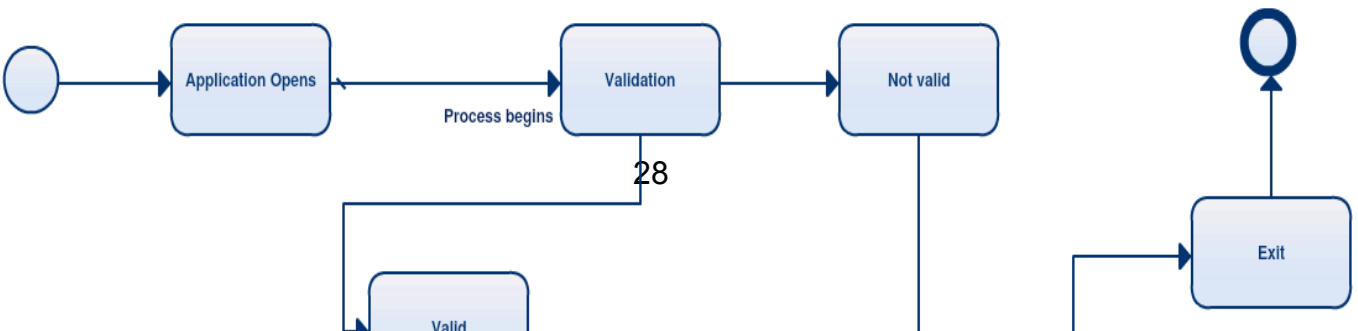
For Admin:



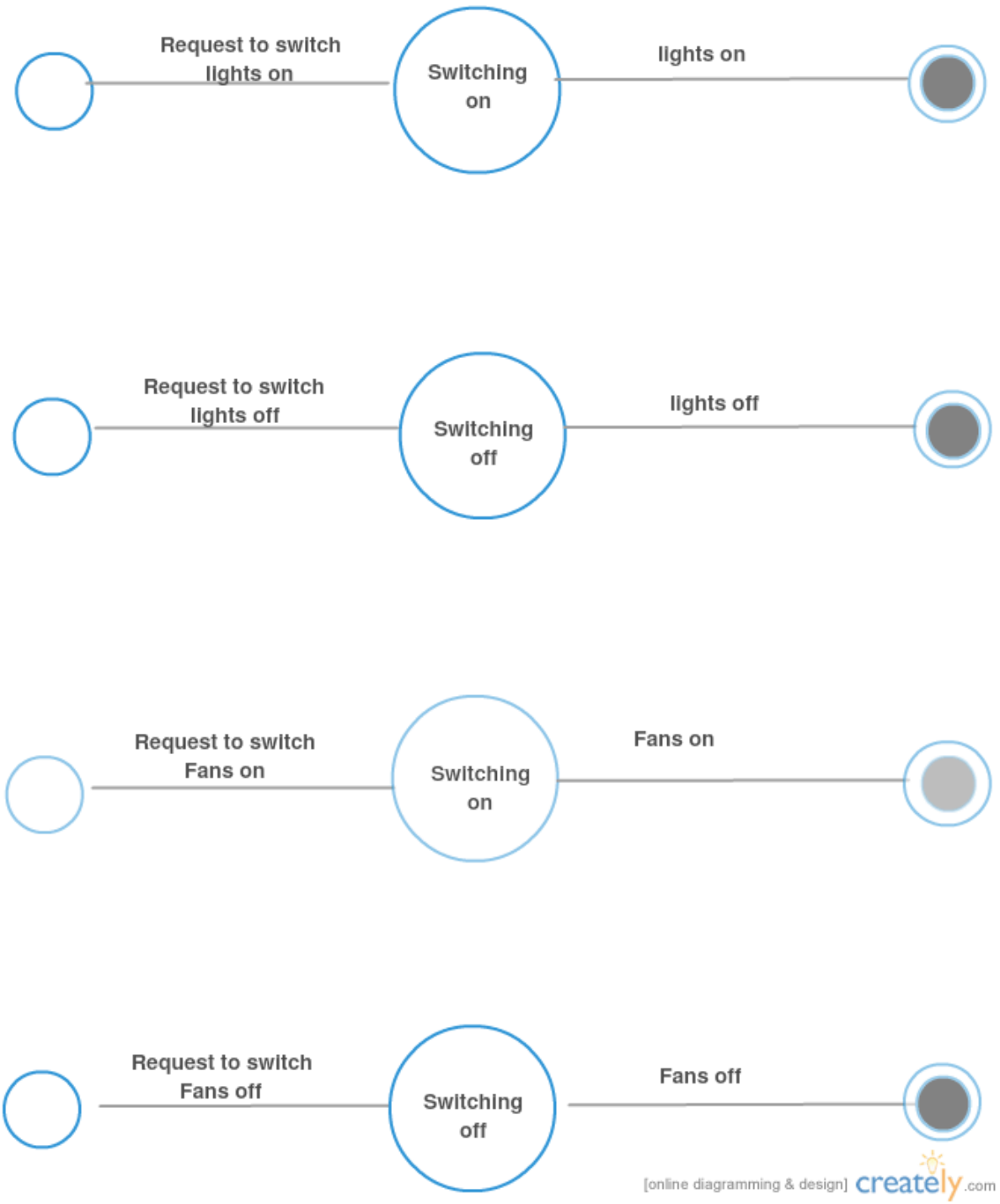
For normal users:



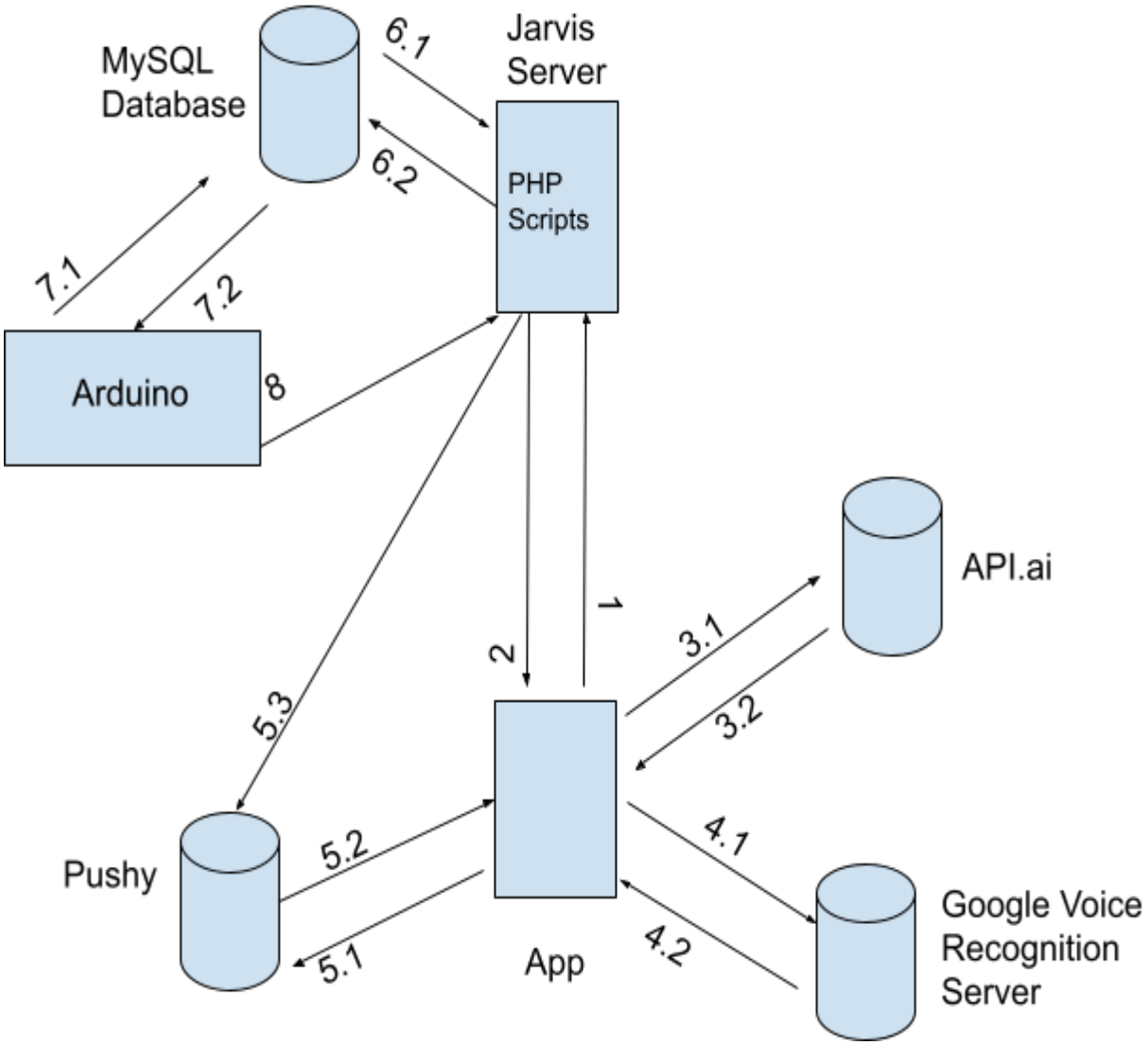
State Diagram



State Diagram for the appliances


Request to switch lights on
Switching on
lights on
Request to switch lights off
Switching off
lights off
Request to switch Fans on
Switching on
Fans on
Request to switch Fans off
Switching off
Fans off
[online diagramming & design] creately.com

The Flow of Control Layout:



Legend:

1. Ajax Request
2. HTTP Response,JSON object(only for temperature)

3.1 GET request

3.2 Json Object

4.1 Audio Input

4.2 Speech

5.1 GET request

5.2 Push notification from pushy

5.3 Http request for notification from server to pushy

6.1 Read request

6.2 Write request

7.1 SQL request

7.2 SQL response

8 Http request from the arduino to the php files on the server

*Note that the above figure does not describe the flow of data within the system but the processes included in it.

The Modules:

Module 1 : The web pages

We Start off by making simple web pages.These pages would contain the basic switches to switch the light and fans on/off.

Objective:

The main objective of this module was to create a web page that could manipulate the changes in the database thus enabling arduino to read these changes.
So when the user wishes to switch on or off a light or fan they can just simply enable the switch.
We use a toggle button for switches because the switches call have only two fixed values “on or off”.

Here in order to use a toggle button in html, a third party framework was needed. Bootstrap material design was the apt framework for it.

The files required are then included in the html file. That involves including the bootstrap css, javascript and jquery files.

The toggle button is added to the html page.

```
      <form>
                  <div class="togglebutton" >
                           <label>
                                    <input type="checkbox" id="light1">
Light 1
                           </label>
```

```
            </div>
      </form>
```

for this toggle button a jquery event handler is added.

```
<script type="text/javascript">
     $(document).ready(function() {

         $("#light1").change(function(){
            if($(this).prop("checked") == true){
               var light = 1;
                    $.ajax({
                       type: 'POST',
                        url: 'simon_says.php',
                        data: {light1:light},
                        success: function(data)
                           {
                              alert(data);
                            }
                      });
                          }
                   else  {
                         var light = 0;
                           $.ajax({
                           type: 'POST',
                          url: 'simon_says.php',
                          data: {light1:light},
                          success: function(data)
                             {
                           alert(data);
                          }
                          });
                       }
                    });
               });
          </script>
```

Here the handler checks when the checkbox is clicked that if it is enabled or disabled and accordingly using ajax sends the value of 1 or 0 to the php code on the server though post.

The php code looks for the value received through ajax and writes value to the corresponding table in the mysql database. Then it sends a ‘success’ message as a reply back if the connection and query execution is successful otherwise an error message is send back.
Ajax captures this message and displays an alert box with the message.
Thus we can add multiple switches in the same format.

Tables maintained within the Database

To keep a complete record of the switches we maintain a table called the States which contains the columns for the lights, fans and the security mode.
The values within the table reflect the following binary values:
1: switched on 0:Switched off

The arduino continually checks within this table for changes in the values.
now if a change in the values within the database would occur then the arduino would take up the corresponding action.But because it has to check continuously it would move into a stressful state and thus reduce its performance.

The solution to this problem is that we implement a checker variable within a checker table and fire a trigger to change the state of this checker variable.Thus the arduino would have to keep a tab only on one variable.
If it detects a change in the value of this variable only then will it search for the specific value that has changed in the states table.

the following is the code for the trigger to change the value of the checker variable:

```
CREATE TRIGGER `checker_trigger`
AFTER UPDATE ON `states`
FOR EACH ROW
UPDATE `checker` SET `state`=1 WHERE `id` = 0
```

The arduino code:

```
/*here we define the relay pins and make some change the
output*/


#include "SPI.h"
#include "Ethernet.h"
#include "sha1.h"
#include "mysql.h"
/* Setup for Ethernet Library */
byte mac_addr[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
```

The above code imports the various library files. The mysql.h is used to directly connect to the MySQL database.
The mac_addr[ ] is a byte array which defines the mac address of the arduino which will be used to connect to the internet.

The following lines of code are simple SQL querries and the connections to the internet are also included
It also includes the void loop() and the void setup()

Void setup() is used to set all the pins and their modes. It is also used to connect to the internet using the Ethernet.begin(Mac_addr,IP_addr,dns_adddr,subnet,gateway).The void setup calls the void loop after it is connect to the internet.

The void loop() is a continuous loop where all the main functions run. Through this loop the arduino can continuously check for changes in the database values for all the appliances.

```
const char checker_querry[] = "SELECT * FROM
malharfe_arduino.checker WHERE id=0";
const char states_querry[] = "SELECT state from
malharfe_arduino.states";
```

```
const char revert_querry[] = "UPDATE malharfe_arduino.checker
SET state = 0 WHERE id = 0 ";
/*Setup for the Connector/Arduino */
Connector my_conn; // The Connector/Arduino reference
char user[] = "malharfe_jojo";
char password[] = "******";
;
int fan1_current_val = 0;
int light1_current_val = 0;
int checker_state = 0;
int myArray[2];

int Relay1=7;
int Relay2=8;

void setup() {

 pinMode(Relay1, OUTPUT);
 pinMode(Relay2, OUTPUT);

// the ethernet.begin should also include the ip address, dns,
subnet and the gateway.

 Ethernet.begin(mac_addr);
 Serial.begin(9600);
 while (!Serial);
 delay(1000);
 Serial.println(Ethernet.localIP());
 Serial.println("Connecting...");
 if (my_conn.mysql_connect("www.malharfest.org", 3306, user,
password)){
 Serial.println("Success connecting to malharfest.org");


 } else {
 Serial.println("Connection failed.");

 }

}
```

```
void loop() {

 relay_switching();
 delay(500);
}
```

The void loop calls the following relay_switching() function which in turn checks the database

For example if the arduino find the value of light one=1 in the database then it performs a

digitalWrite(High,pin_no)

Here the pin_no is replaced with relay1. While realy1 is nothing but the pin that is connected to the relay board.

Thus we can perform all the other connections.

```
void relay_switching(){
  Serial.println("hello.. Is there anybody in there");
my_conn.cmd_query(checker_querry);
 my_conn.get_columns();

 row_values *row = NULL;
 do {
 row = my_conn.get_next_row();
 // this is to check the value of checker from the database
 if (row != NULL) {

 checker_state = atol(row->values[0]);
 }
 my_conn.free_row_buffer();
 } while (row != NULL);
 my_conn.free_columns_buffer();

 if(checker_state==1){

     my_conn.cmd_query(states_querry);
     my_conn.get_columns();
     row_values *row = NULL;
     int i = 0;
     do {
```

```
    row = my_conn.get_next_row();

    if (row != NULL) {
    myArray[i] = atol(row->values[0]);
    }
    i++;
    my_conn.free_row_buffer();
    } while (row != NULL);
    my_conn.free_columns_buffer();

    light1_current_val=myArray[1];
    fan1_current_val=myArray[0];

         if(light1_current_val==1){
         Serial.print("light1 = ");
         Serial.println(myArray[1]);
         Serial.println("Turning the Light On");
         digitalWrite(Relay1, HIGH);
         }
         else{
         digitalWrite(Relay1, LOW);
         Serial.println("Turning the Light Off");
         }

    my_conn.cmd_query(revert_querry);
  }
}
```

Module 2: Circuit connections

Objective:

maintaining and securing all the connections to the arduino.

Physical connections to the relay board:

Why to use relay for controlling AC light?

AC is alternating current 220v (india) which powers the ac lights. Arduino cannot control high volt amp, but a relay can do this job, which is the sole design of it. so we are using relay as switch to control high power devices.

What is NO NC and COM in relay?

C = Common Connection

NC = Normally Closed Connection

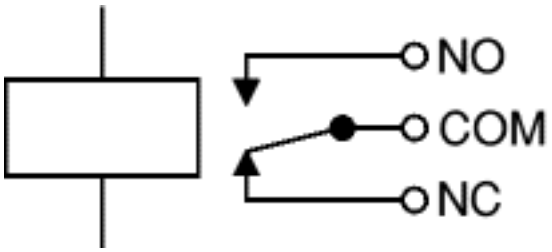
NO = Normally Open Connection

Thus the 5v current from the arduino is now amplified when it passes through the relay to produce 220-240 volts which is the current range used by most of the appliances at home.

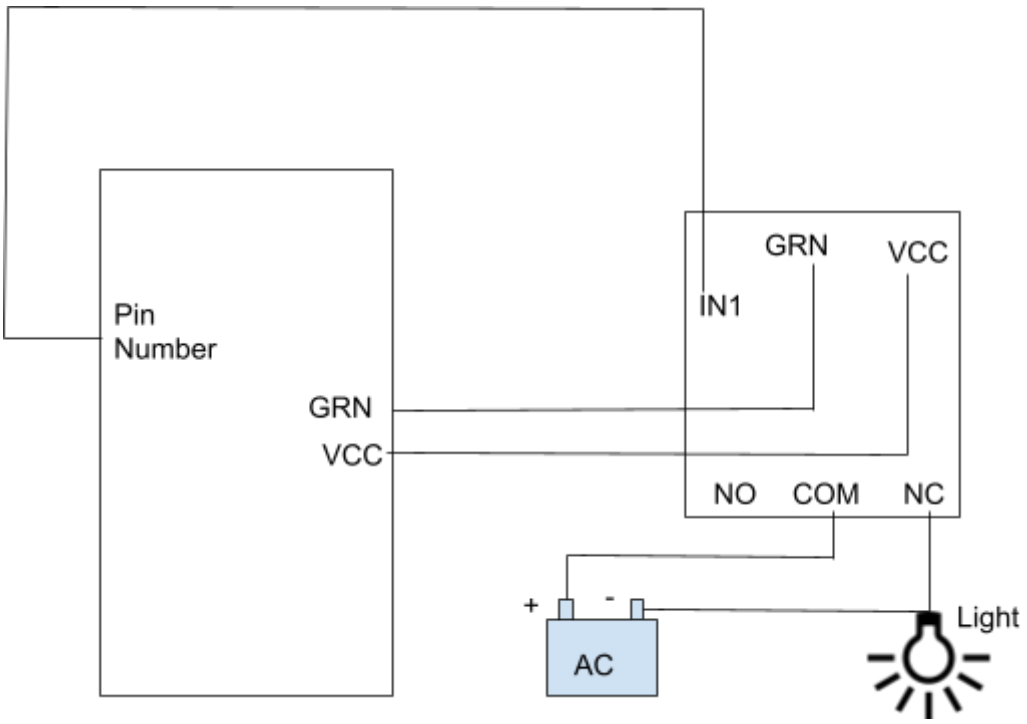
COM - Common connection--> it is the center terminal, It is hot as power to the load is connected at this terminal.

NO Normally open ---> It acts like a switch,since it is open - there will be no contact between COM and NO, When we trigger the relay module, it connects to COM by the electromagnet inside the relay and supply to the load is provided,which powers up the light.Thus the circuit is closed until we trigger the state to low in relay.

NC Normally closed---->It is always in contact with COM, even when relay is not powered.when we trigger the relay it opens the circuit, so the connection is lost. it behaves just opposite to NO.



Here we used Normally closed connection
The following is the complete circuit diagram to connect the arduino to the light through the relay.

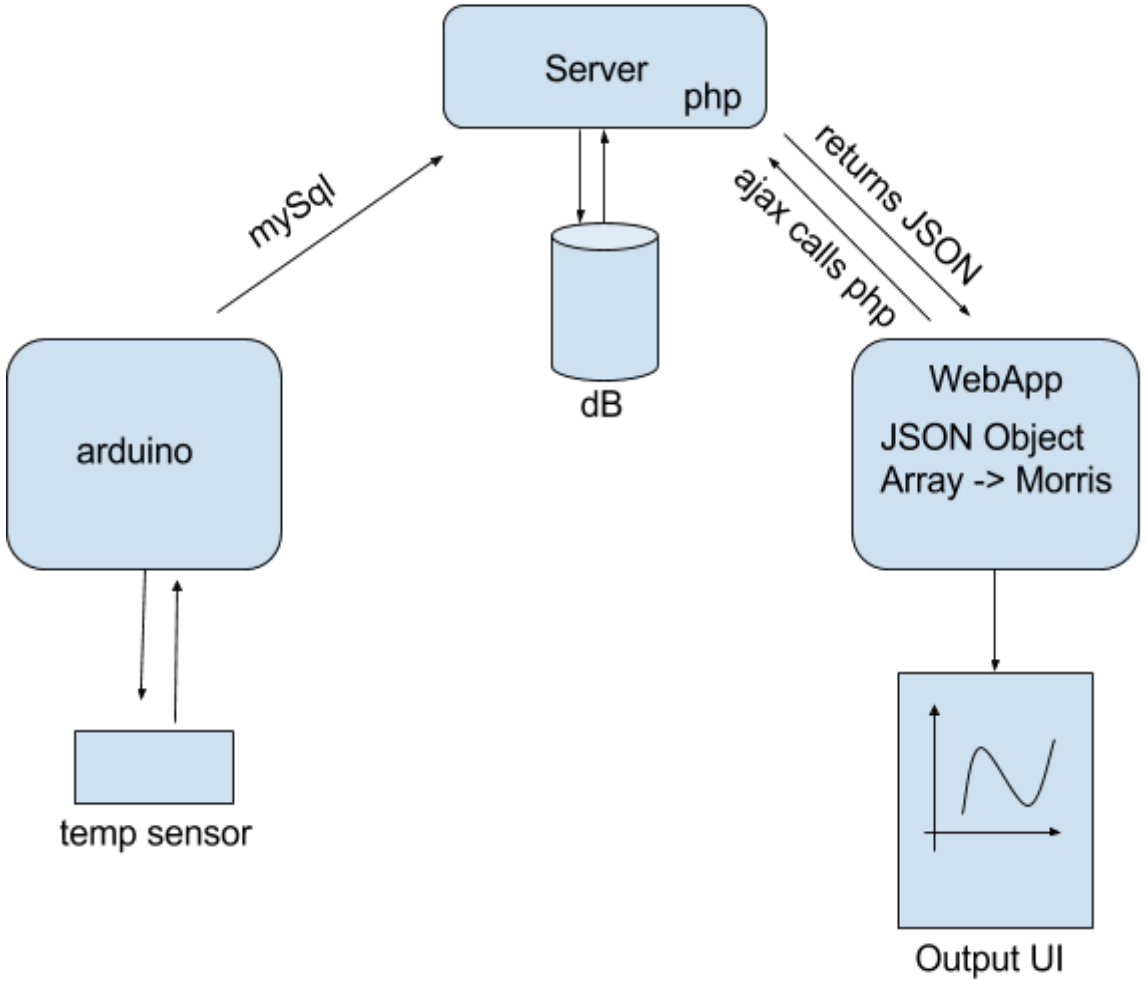


Module 3:The Temperature module

objective:

To setup the Temperature sensor and provide the end users with not only the current temperature but also a graph with the previous records of the temperature.

For temperature sensor the data flow happens in the following pattern.



So the arduino sensor reads the temperature in a periodic intervals and stores it in a mysql database on the server. The front end web application then accesses these data stored in database and then displays in a graph format. For displaying it in a graphical way, we use a jquery library called Morris js. It takes JSON array objects and displays it in a graphs.

Mysql database

We want the database to store the temperature and the date and time of creation.To make it easier for the arduino board , it will only send the temperature reading and we want the mysql server to store the date and time automatically.

Mysql server only supports TIMESTAMP as an automatic default value. TIMESTAMP stores both the time and date in ‘ YYYY-MM-DD HH:MM:SS ‘ format. Since the date and time is in the same column it makes the querry’s complicated. So to separately store date and time we use trigger instead. First we make a table temp with `id(primary key, AUTO_INCREMENT), temperature, enDate,enTime` . And then we add the following trigger to it.

```
CREATE TRIGGER `date_trigger` BEFORE INSERT ON `temp`
FOR EACH ROW if ( isnull(new.enDate) ) then
set new.enDate=curdate();
set new.enTime=curtime();
end if
```

Now when we store temperature , date and time is automatically stored.

Testing with test values

id	temperature	enDate	enTime
5	33	2016-02-08	12:11:53
6	35	2016-02-08	12:11:53
7	32	2016-02-08	12:12:28
8	36	2016-02-08	12:12:28
9	22	2016-01-04	*NULL*
10	23	2016-01-04	*NULL*
11	24	2016-01-05	*NULL*
12	22	2016-01-03	*NULL*
13	23.6	2016-01-03	*NULL*

sql query for temperature day-wise

```
SELECT ROUND(AVG(temperature),2) as temp , enDate as date From
temp GROUP BY `enDate
```

temp	date
22.80	2016-01-03
22.50	2016-01-04
24.00	2016-01-05
34.00	2016-02-08

Sql query for temperature month-wise

```
SELECT ROUND(AVG(temperature),2) as temp , DATE_FORMAT(`enDate`,
'%Y-%m') as date From temp GROUP BY DATE_FORMAT(`enDate`,
'%Y-%m')
```

temp	date
22.92	2016-01
34.00	2016-02

Once the database is ready , the front end web pages can be made.
We make temp.html that using Jquery's on load function calls the php page get_temp.php, which returns a json array object.

The HTML Page:

temp.html

we add the following morris library to the head

```
<link rel="stylesheet"
href="//cdnjs.cloudflare.com/ajax/libs/morris.js/0.5.1/morris.css">
<script
src="//cdnjs.cloudflare.com/ajax/libs/raphael/2.1.0/raphael-min.js"><
/script>
<script
src="//cdnjs.cloudflare.com/ajax/libs/morris.js/0.5.1/morris.min.js">
</script>
```

In the body we add an empty div element which will be transformed into a graph using morris script

```
<div class="row">
<div class="col-lg-12">
          <div id="morris-line-chart"></div>
     </div>
</div>
```

We then add the following script to display the chart in the above element.

```
<script type="text/javascript">
     $(document).ready(function() {
var months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
"Aug",
               "Sep","Oct", "Nov", "Dec"];
     //Initially on load set's the graph sorting on month basis
          $.ajax({
                 type: 'POST',
                 url: 'get_temp.php',
                 data: {sort_by:"month"},
                 dataType: 'json',
                 success: function(Json_data)
                  {
                    Morris.Line({
```

```
                    element: 'morris-line-chart',
                    data: Json_data,
                    xkey: 'date',
                    xLabels:'month',xLabels:'month',
                     xLabelFormat: function(x)
{
  var day = months[x.getMonth()];
        return day;
      },
ykeys: ['temperature'],
      labels: ['temperature'],
      ymin: 'auto',
      ymax: 'auto',
      postUnits : '°C'

         });

     }

 });
     });

</script>
```

The Code - Explanation

- When the page in ready ,the .ready jquery function uses ajax to calls get_temp.php and sends a variable sort_by with value month (to initially load the graph of month values) through Post action.
- Ajax then receives a JSON object from the php code (Json_data)
- Then using Morris.Line function we set the attributes for morris graph that should be displayed
    - element : selects the id of element in the body where the graph should be displayed.
    - data: The data to plot. This is an array of objects, containing x and y attributes as described by the xkey and ykeys options. In the above code we use Json_data as the objects array to be used.
    - xkey : A string containing the name of the attribute that contains date (X) values.

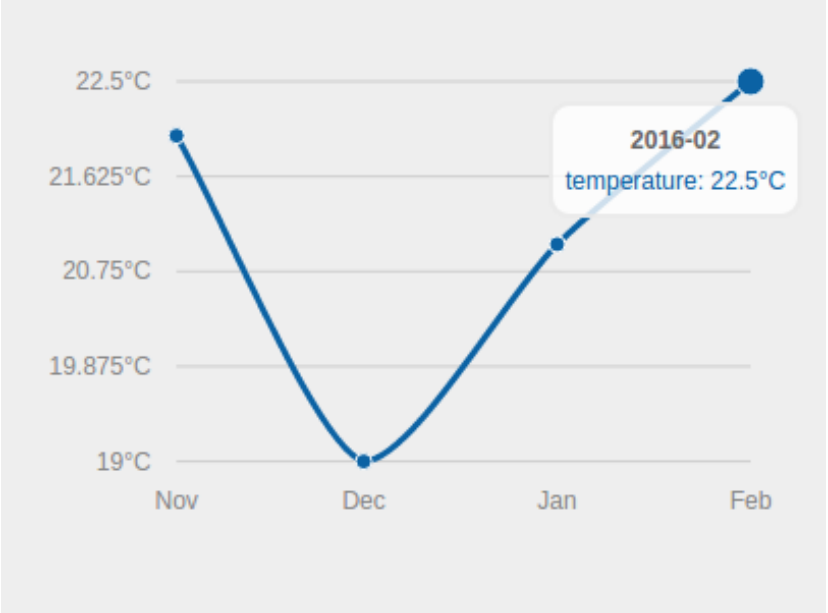
- ykeys : A list of strings containing names of attributes that contain Y values (one for each series of data to be plotted).
- labels : A list of strings containing labels for the data series to be plotted (corresponding to the values in the ykeys option).
- xLabels: Sets the x axis labelling interval. By default the interval will be automatically computed. The following are valid interval strings:
  - "decade"
  - "year"
  - "month"
  - "week"
  - "day"
  - "hour"
  - "30min"
  - "15min"
  - "10min"
  - "5min"
  - "minute"
  - "30sec"
  - "15sec"
  - "10sec"
  - "5sec"
  - "second"
- A function that accepts Date objects and formats them for display as x-axis labels. Overrides the default formatter chosen by the automatic labeller or the xLabels option. eg: function (x) { return x.toString(); }

PHP Code

```
$query_month = "SELECT ROUND(AVG(temperature),2) as temperature
, DATE_FORMAT(`enDate`, '%Y-%m') as date From temp GROUP BY
DATE_FORMAT(`enDate`, '%Y-%m')";
```

```
  $sort = $_POST[""];
if($sort=='month'){
           $query = $query_month;sort_by
         }

if ($result->num_rows > 0) {

               $data = array();
               while ($array = mysqli_fetch_assoc($result)) {
                    $data[] = $array;
               }

             }
echo json_encode($data);
```

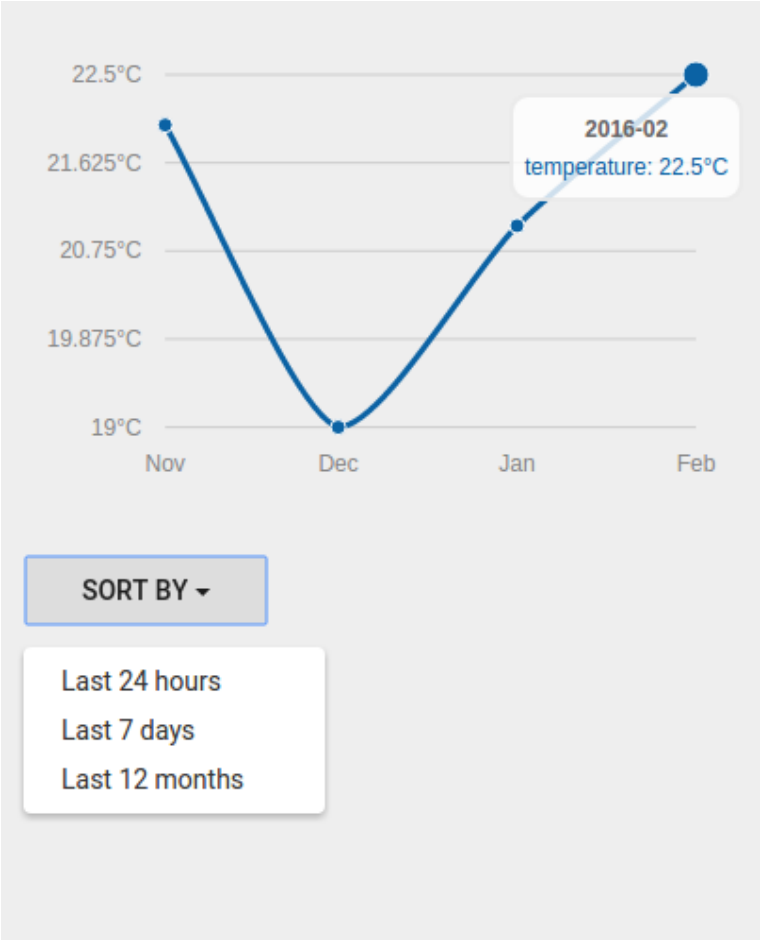
After adding a lot of test(test) values to test the implementation of the code with we get the following result on temp.html page



Now to add different sorting types such as monthly, daily,weekly etc we add a drop down menu

```
<div class="dropdown">
 <button class="btn" id="menu1" type="button"
data-toggle="dropdown">Sort By
```

```
        <span class="caret"></span></button>
           <ul class="dropdown-menu" role="menu"
aria-labelledby="menu1">
             <li><a href="#" id="action-day">Last 24 hours
hours</a></li>
             <li><a href="#" id="action-week">Last 7 days</a></li>
             <li><a href="#" id="action-month">Last 12
months</a></li>
           </ul>
</div>
```



Now to these elements an event listener needs to be added.

```
<script type="text/javascript">
     $("#action-week").click(function(e){
         // Code to be executed
```

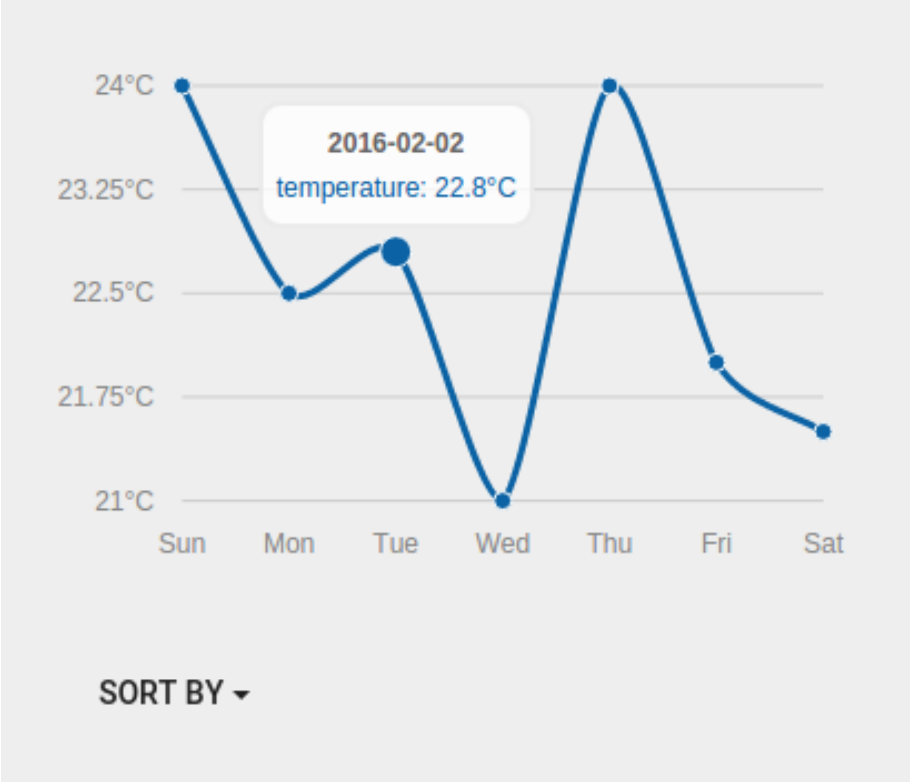
```
        e.preventDefault();
    });
</script>
```

Now for each click ,we will remove the div element that contains the graph , add a new div element with the same id name and use the previously used Morris function but with changing the data that is needed to be changed.

Code to delete and add the div element again

```
var child = document.getElementById("morris-line-chart");
var parent = document.getElementById("graph_parent");
parent.removeChild(child);

var div = document.createElement('div');
div.id = 'morris-line-chart';
parent.appendChild(div);
$("#morris-line-chart").css("height","250px");
```

After adding the proper morris code we get the below result when we click last 7 days



Same for last 12/24 hours , we add the following query in our php file

```
SELECT `temperature`, CONCAT_WS(' ',`enDate`,`enTime`) as time
FROM `temp` WHERE `enDate` = curtime(),
```

But since we are using test data values use `enDate` = '2016-02-07' , or any date that has timely values.

temperature	time
19	2016-02-07 06:00:00
20	2016-02-07 10:00:00
22	2016-02-07 14:00:00
21	2016-02-07 18:00:00

Time zone error in server

The problem with having a shared hosting server is that the time zone is set by the the operating system on which the the server runs and the OS time depends on where the server is located. Changing the server timezone requires root privileges. Hence we have to manually change the time- zone each time we fire a query that uses functions like curtime() , curdate() and now().

Therefore the query to be used is as Follows:

`SET SESSION time_zone = "+05:30"` // for India

then followed by the specific query is to be executed;

So as an example we'll change the above query as follows

```
char INSERT_DATA[] = "SET SESSION time_zone = "+05:30" ;Insert
into malharfe_arduino.temp_current(temperature) Values(%s)" ;
```

```
SET SESSION time_zone = "+05:30" ;INSERT INTO
`temp_current`(`temperature`) VALUES ('23');
```

Procedure solution

The temp_current table becomes very deep as values are recorded every 5 minutes. Therefore to maintain a more systematic data repository we maintain a fixed amount of records in the table.Thus we delete/replace the the older records with the newer ones.

But when we use the before and after insert triggers on the table an error is thrown.
It is because within a stored function or trigger, it is not permitted to modify a table that is already being used (for reading or writing) by the statement that invoked the function or trigger. Accordingly, we try the following solution:
we create a stored procedure, that inserts into and deletes from the target table.

```
DELIMITER //
CREATE PROCEDURE current_temp(IN temp VARCHAR(255))
```

```
BEGIN
      SET SESSION time_zone = "+05:30" ;
      INSERT INTO malharfe_arduino.temp_current(temperature)
VALUES(temp);
      DELETE FROM malharfe_arduino.temp_current ORDER BY `id` LIMIT
1;
END //
DELIMITER ;
```

Now arduino can call the procedure directly and the code can be simpler like

```
char INSERT_DATA[] = "call current_temp(%s)" ;
```

This code works well as long as the id of the new entry is more than the old one. But what if the auto increment reaches it's max number to increment (let's say 100) and then it starts from 1 again. This will cause us a problem because according to our given definition for procedure it will delete the row with id 1, even though it is the new entry at this point as it the smaller than the existing row with id of 100.Thus the procedure get's stuck in a state where it keeps on deleting the new value thinking it is an old value.

Probable solution -

we could try to use timestamp to compare the old and new values instead of keeping a check on the id as the current time entry would always be greater that the previous entry.

Maintaining the temperature values(The Timer):

When a user requests for temperature, we not only display the current temperature values but also the graph of the previous recorded values.
To maintain a record of these previous values we continuously read the temperature from the sensor after every 5 mins and after every 4 hours in different tables namely current_temp and temp resp.

We download a 3rd party timer library to schedule a specific event , like in this case updating the temperature periodically

Therefore we introduce a timer to keep a record of time

We include 2 new libraries which are as follows:

```
#include <Time.h>
#include <TimeAlarms.h>
```

The Alarm library is a companion to the Time library that makes it easy to perform tasks at specific times or after specific intervals.

The entire code for arduino is as follows:

```
#include "SPI.h"
#include "Ethernet.h"
#include "sha1.h"
#include "mysql.h"
#include <Time.h>
#include <TimeAlarms.h>

float temp;
int tempPin = 0;

char INSERT_CURRENT_TEMP[] = "CALL
malharfe_arduino.current_temp(%s)" ;
char INSERT_TEMP[] = "CALL malharfe_arduino.temp(%s)" ;
char query[128];
char temperature[10];


void setup()
{
Serial.begin(9600);
  Ethernet.begin(mac_addr);
 delay(1000);
 Serial.println("Connecting...");
 Serial.println(Ethernet.localIP());
 Serial.println("Connecting...");
 if  (my_conn.mysql_connect("www.malharfest.org", 3306, user,
password)){
```

```
 Serial.println("Success connecting to malharfest.org");

 Alarm.timerRepeat(300, readTemp_current);
 Alarm.timerRepeat(900, readTemp);
 readTemp();
 readTemp_current();

  } else {
 Serial.println("Connection failed.");

 }

}

void loop()

{
Alarm.delay(0);

}
void readTemp_current(){
temp = analogRead(tempPin);
Serial.print("Inside the readTempCurrent : ");
temp = temp * 0.48828125;

dtostrf(temp, 1, 1, temperature);
 sprintf(query, INSERT_CURRENT_TEMP,temperature);
 Serial.print("every 1 min " );
 Serial.println  (query);
 my_conn.cmd_query(query);

 }

void readTemp(){
temp = analogRead(tempPin);
Serial.print("Inside the readTemp : ");
temp = temp * 0.48828125;

dtostrf(temp, 1, 1, temperature);
 sprintf(query, INSERT_TEMP,temperature);
```

```
Serial.print("every 5 min " );
Serial.println  (query);
my_conn.cmd_query(query);

}
```

Explanation is bits:

The queries are the respective calls to the created procedures.

```
char INSERT_CURRENT_TEMP[] = "CALL
malharfe_arduino.current_temp(%s)" ;
char INSERT_TEMP[] = "CALL malharfe_arduino.temp(%s)" ;
```

where %s specifies that the values has to be obtained in a string format.
we then check the connection to the server.

The following lines is where the library comes into the picture.

```
Alarm.timerRepeat(300, readTemp_current);
```

The timer thus created triggers a task or a function (in our code readTemp_current) after every x seconds(here 300 seconds).
And similarly the timer triggers read_temp after every 900 seconds.

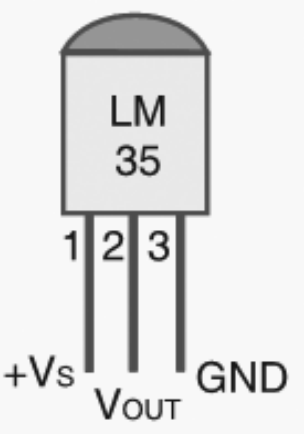
```
Alarm.timerRepeat(900, readTemp);
```

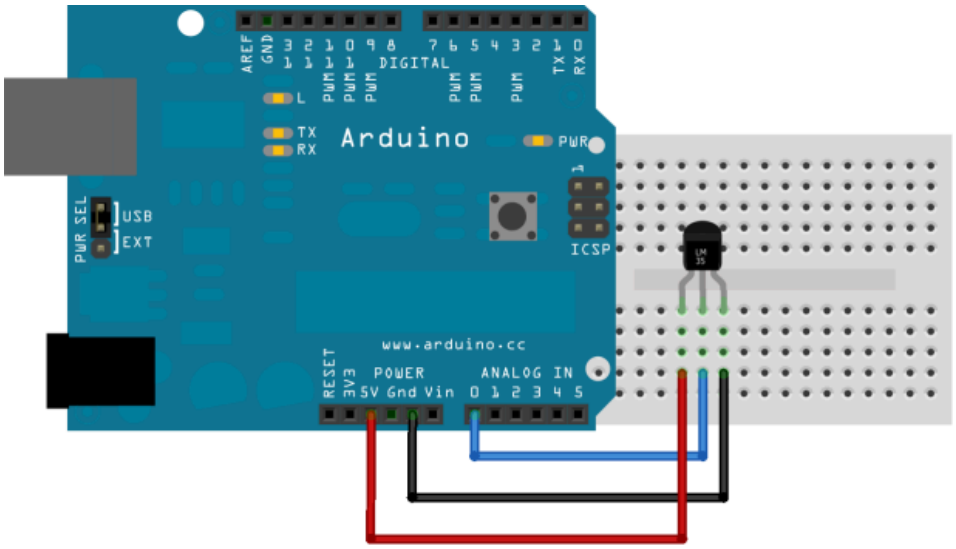
The functions readTemp_current and read_temp calculate the temperature and record the corresponding values in the database.

```
void loop(){
     Alarm.delay(0);
     }
```

Task scheduling is handled in the Alarm.delay function.Tasks are monitored and triggered from within the Alarm.delay call so Alarm.delay(Int value to delay in seconds) should be called whenever a delay is required in your sketch.
If your sketch waits on an external event (for example, a sensor change), make sure you repeatedly call Alarm.delay while checking the sensor.
You can call Alarm.delay(0) if you need to service the scheduler without a delay.
The void loop keeps the timer running.

The physical connection of the Sensor to the arduino:
We used a LM35 sensor which has an accuracy of 98% and can withstand a wide range of temperatures ranging between -50℃ to 150℃
The following is the circuit diagram to connect the arduino to the sensor.
We make use of the breadboard to secure the sensor.





Tables maintained within the database.

There are 2 tables maintained with the database namely temp and current_temp

The temp table:

This table maintains a complete record of the temperature with a timestamp after every 30 mins
since the user has the liberty to choose the temperature graph of the previous reading on a hourly monthly and yearly basis the values for the graph are retrieved from this table

The current_temp table:

the current temperature at that very moment is also given to the user when the user selects the temperature module or requests for the temperature which we maintain a current_temp table.
It records data after every 10 mins and maintains a timestamp as well. it keeps overriding the temperature value thus maintaining only one row.

The JSON obect fetches the values of this table for the current temperature

At trigger called date_trigger_current is created to override the values within the current_temp table in the following manner:

```
CREATE TRIGGER `date_trigger_current`
BEFORE INSERT ON `temp_current`
FOR EACH ROW if ( isnull(new.enDate) ) then
set new.enDate=curdate();
set new.enTime=curtime();
end if
```

Once all the connections have been made and they are secured then we can test out sensor.
Ideally the values read by the sensor should be entered with the database and once the user requests for the temperature a graph along with the current recorded temperature would be displayed. The user can then sort the data according to his convenience.

Module 4:Security module

objective:

The main objective of this module is to make sure the house is secured from intruders. The concept involves a motion sensor connected to the arduino and a security mode. When the user need to sure his house he just needs to enable the security mode. This process activates the sensor which is places at a prime location in the house (For example the entrance or the hall).

The sensor continuously checks if any movement is detected. Once a motion is detected it sends a signal to the arduino and a Push notification is sent to the mobile application.

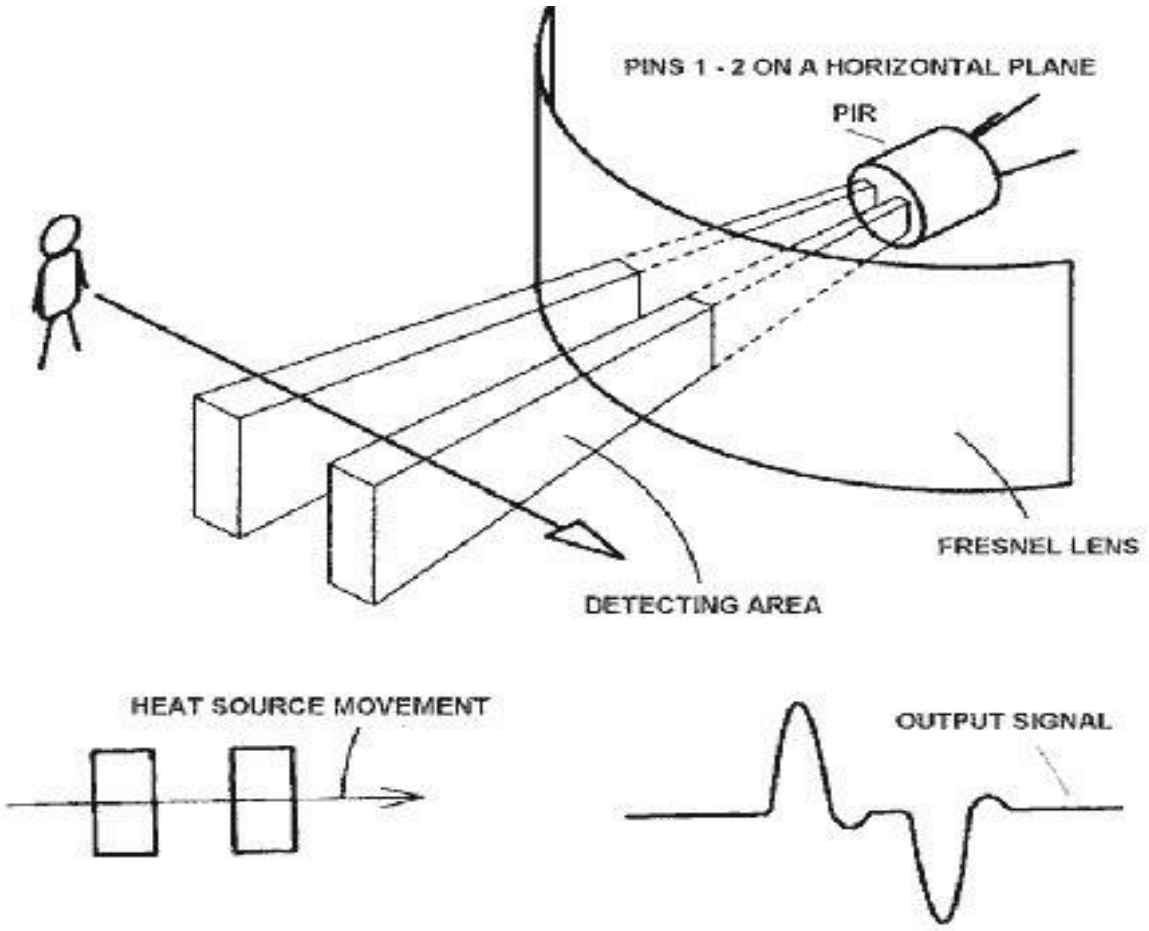
The Motion Sensor.

we'll build a motion-sensing arduino alarm using a PIR (passive infrared) sensor and an Arduino microcontroller.

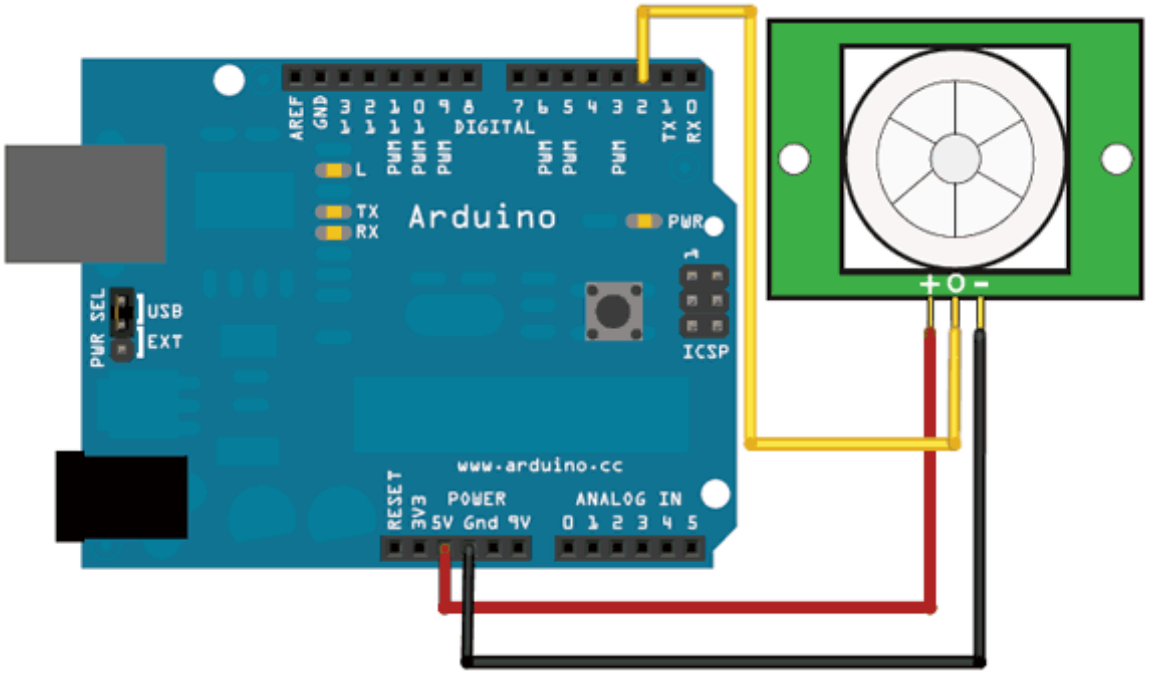
The PIR sensor:
PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors, which can detect levels of infrared radiation.

The working:
Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

The PIR sensor itself has two slots in it, each slot is made of a special material that is sensitive to IR. When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors. When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a positive differential change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected.



Connection to the Arduino:



The connections:

Arduino	**Motion sensor**
5V	**positive end**
GND	**negative end**
Pin_number (x)	**Center Open connection**

The Code for audino is as follows:

```
int inputPin = 2;

int pirState = LOW;
int val = 0;                    // variable for reading the pin
status

void setup() {
pinMode(inputPin, INPUT);     // declare sensor as input

Serial.begin(9600);
}
```

```
void loop(){
val = digitalRead(inputPin);  // read input value
if (val == HIGH) {            // check if the input is HIGH
    if (pirState == LOW) {
    // we have just turned on
    Serial.println("Motion detected!");
    // We only want to print on the output change, not state
    pirState = HIGH;
  }
} else {

  if (pirState == HIGH){
    // we have just turned of
    Serial.println("Motion ended!");
    // We only want to print on the output change, not state
    pirState = LOW;
  }
}
}
```

Explanation in bits:

```
int inputPin = 2;
int pirState = LOW;
int val = 0;
```

Choose the input pin (for PIR sensor here we choose pin number 2).We then start, assuming no motion detected, this is done by setting the value of the pirState variable to LOW .We select a variable named val for reading the pin status

```
void loop(){
val = digitalRead(inputPin);  // read input value
if (val == HIGH) {            // check if the input is HIGH
    if (pirState == LOW) {
    // we have just turned on
    Serial.println("Motion detected!");
    // We only want to print on the output change, not state
    pirState = HIGH;
```

```
  }
} else {

  if (pirState == HIGH){
    // we have just turned of
    Serial.println("Motion ended!");
    // We only want to print on the output change, not state
    pirState = LOW;
  }
}
```

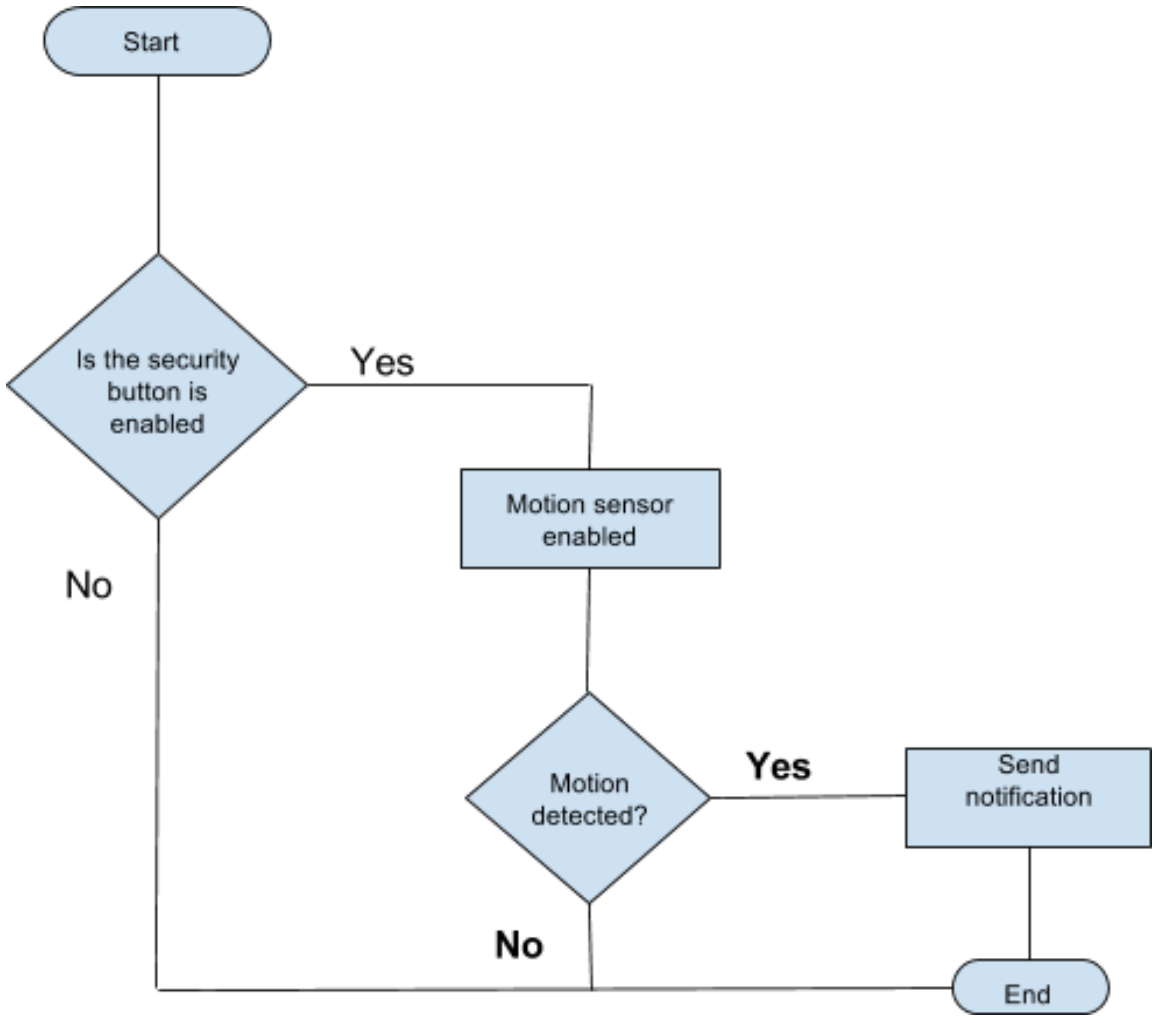
Within the void loop we check if the motion sensor has detected any changes and thus display them in accordance.

The motion sensor works Hand in Hand with Pushy(A Third party application which we shall see in detail) which is used to send a Server side push notification to the mobile device.This message is sent when motion is detected by the sensor irrespective if the app is open or not.Thus the user can now know if there has been any breach in security.

Module 5:

The Server Notification(security message)

A notification is a message you can display to the user outside of your application's normal UI.
With the working motion sensor our main motive of a push notification is the check if the security of the house is breached.That is when one enables the security mode on our mobile app the sensor is then activated. any sort of human activity is then detected by the sensor and a push notification is sent to the mobile phone of the user regardless if the application is on or not.
The flow of data is given from the flowchart below



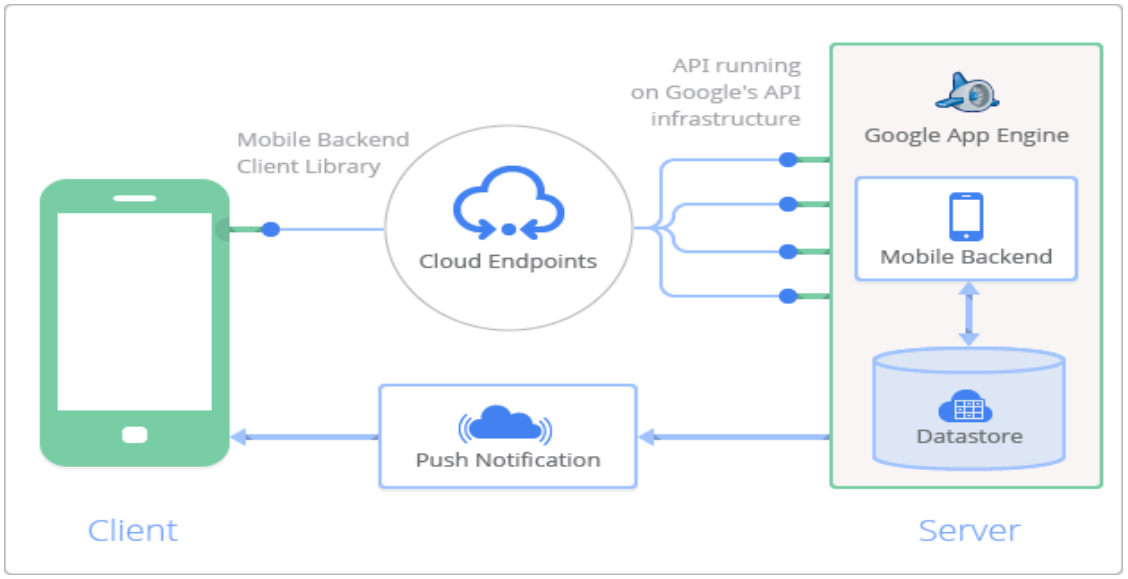
The problem arises when we need to send the notification to the mobile phone
We therefore can not implement normal application notifications because we need to send a message from our server database and not from within our application.

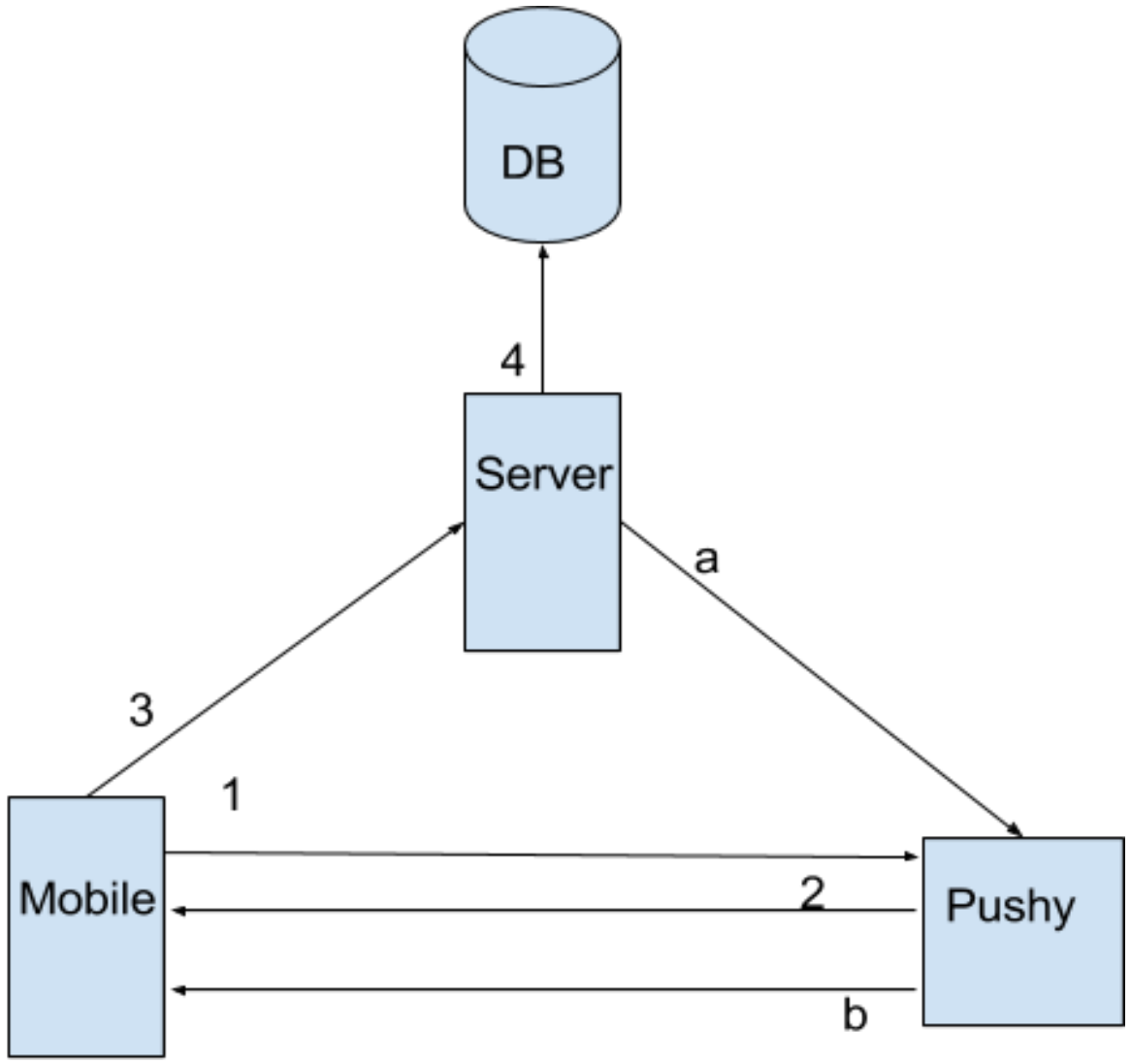
Therefore instead of using normal applications in which the the notification is sent from within the app we make use of a third party server known as Pushy. This solves our above problem.

Pushy

Pushy is a push notification gateway, perfect for real-time applications.It notifies our users immediately. Devices maintain a background connection to Pushy using the MQTT protocol(MQTT is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol.It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.), an extremely lightweight pub/sub protocol that consumes very little network bandwidth and battery, which makes it perfect for mobile.

The Working:





Keynotations:

1. Android device sends sender ID, application ID to the Cloud Service pushy for registration
2. upon Successful registration the server issues registration id to Android devices.
3.after receiving registration id, device will send registration id to our server.
4. our server will Store registration id in the database for later usage.

a.Whenever push notification is needed our server sends a message to the clouds server pushy along with device registration id (which is stored earlier in the database)

b. pushy server will deliver that message to the respected mobile device using that particular registration id

The following are the steps to integrate Pushy :

Steps 1:We get started by signing up for free by providing the package name and receiving an API key.

Step 2: download the pushy SDK version 1.0.7
we then copy the .jar file to Android project libs a folder

Step3:Import the SDK into our Android project
In Gradle, we have to make sure the following line is present in your build.gradle (it is present by default):

```
     dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
}
```

Step 4:Register Devices for Push Notifications
Devices need to be uniquely identified to receive push notifications. Every device is assigned a unique registration ID that you can use to push it at any given time. Once the device has been assigned a registration ID, it should be stored in your application's backend database. When you're ready to push the device, you'll send us this registration ID via our API, and we'll deliver the push to the corresponding device.
Step 5:simply add the following code to your application when you're ready to register the device:

```
            new registerForPushNotificationsAsync().execute();
```

Step 6:Modify Launcher Activity
You must call the following function in your launcher activity's onCreate() so that Pushy's socket service will restart in case the user force-closed the application and caused the service to be killed by Android.
We recommend placing this code in your main launcher activity.
Find: super.onCreate(savedInstanceState);

Add Below: Pushy.listen(this);

Step7:Add Pushy Permissions to AndroidManifest.xml
Pushy requires the following permissions to maintain a background socket connection, to prevent Wifi from disconnecting when the device enters sleep mode, to check for an Internet connection, to persist the registration ID on the device, and to connect automatically when the device boots.
Add the following lines to your AndroidManifest.xml, inside the <manifest> tag:

```
<!-- Pushy Permissions - Added in v1.0.0 -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission
android:name="android.permission.RECEIVE_BOOT_COMPLETED" />

<!-- Pushy Permissions - Added in v1.0.1 (optional) -->
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

Step 8:Add Pushy Services and Receivers to AndroidManifest.xml
Pushy needs to listen for broadcasts in order to restart the listener service when our app gets updated or when the device wakes up, and to run a background service that maintains an idle socket connection. Also, Pushy needs to know where to send your pushes when they arrive.
Add the following lines to your AndroidManifest.xml, inside the <application> tag:

```
<!-- Pushy Declarations -->

<!-- Pushy Notification Receiver -->
<!-- Incoming pushes will be redirected to the following
BroadcastReceiver class -->
<receiver android:name=".PushReceiver" >
    <intent-filter>
        <!-- Do not modify this -->
        <action android:name="pushy.me" />
    </intent-filter>
</receiver>

<!-- Pushy Update Receiver -->
```

```
<!-- Do not modify - internal BroadcastReceiver that restarts
the listener service -->
<receiver
android:name="me.pushy.sdk.receivers.PushyUpdateReceiver">
    <intent-filter>
        <action
android:name="android.intent.action.PACKAGE_REPLACED" />
        <data android:scheme="package" />
    </intent-filter>
</receiver>

<!-- Pushy Boot Receiver -->
<!-- Do not modify - internal BroadcastReceiver that restarts
the listener service -->
<receiver
android:name="me.pushy.sdk.receivers.PushyBootReceiver" >
    <intent-filter>
        <action
android:name="android.intent.action.BOOT_COMPLETED"/>
    </intent-filter>
</receiver>

<!-- Pushy Socket Service -->
<!-- Do not modify - internal socket background service -->
<service
android:name="me.pushy.sdk.services.PushySocketService"/>

<!-- End Pushy Declarations -->
```

Step 9:Create BroadcastReceiver to Handle Push Notifications
Create the class that will receive our push notifications when they arrive, and execute the relevant action in response, such as issuing a notification, playing a sound, vibrating the phone, etc.
Step 10:Implement Backend API Call
To send push notifications via Pushy, we need to implement a backend API request to our Push API. Send us the device registration IDs along with the push notification payload, and we'll do the rest.
The back end android code can be found at the end of this book.

Tables maintained within the database:

Since the pushy sends us a unique registration id for every new device connected we need to maintain this record of data.

We maintain a table called device_registration within the database which plainly contains the registration ids of the devices and the timestamp i.e the date and the time for which the device first opened our application.

Module 6:

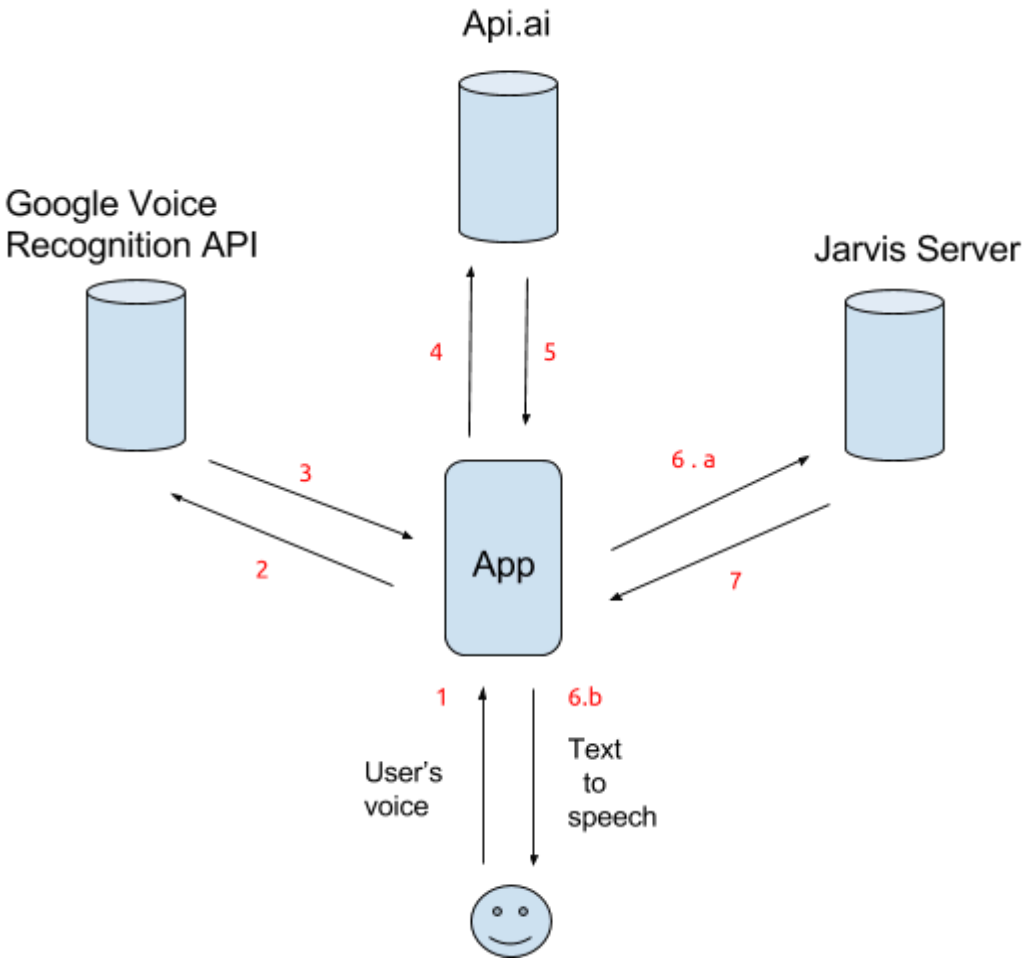
The Artificial Intelligence Module

Objective:

The main goal of this module is to implement voice recognition and make process of home automation more user friendly.
Thus the user can just directly talk to the device and the action would be carried out automatically.

This module is used as our home screen for the application.So once the user has logged onto the application this is the first page he is redirected to.
We make use of a third party tools called the API.ai and the google text to speach and vice versa.
Api.ai provides developers and companies with the advanced tools they need to build voice interfaces for apps and hardware devices.The Api.ai platform lets developers seamlessly integrate intelligent voice command systems into their products to create consumer-friendly voice-enabled user interfaces.
Google Text-to-Speech is a screen reader application developed by Google Inc for its Android operating system. It powers applications to read aloud (speak) the text on the screen.And the google speech to text allows users to speak directly and converts it into text.

The flow of control is given in the following diagram:



Steps

1 - The Android Recognizerintent captures the audio spoken by the user .
2 - This audio is sent to Goolge API to process.
3 - It then sends the text result back to the user.
4 - This received text is then sent to Api.ai php to for Natural Language Procesing.
5 - The api makes sense of the sent text and returns the meaning in simpler terms like parameters and its values.
6.a - These parameters are then processed by the app and accordingly it send request to the Jarvis Server.
6.b - Simultaneously in a background thread, the reply text returned by api.ai is converted from to Speech using google/pico TTS service

7. The reply from the Jarvis server is then outputted on the group.

Android SDK for api.ai

The API.AI Android SDK makes it easy to integrate speech recognition with API.AI natural language processing API on Android devices.
Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction.
API.AI allows using voice commands and integration with dialog scenarios defined for a particular agent in API.AI.
Two permissions are required to use the API.AI Android SDK:

- android.permission.INTERNET for internet access
- android.permission.RECORD_AUDIO for microphone access

We add this dependencies to the project to use SDK

```
compile 'ai.api:sdk:1.7.6@aar'
// api.ai SDK dependencies
compile 'com.android.support:appcompat-v7:22.0.0'
compile 'com.google.code.gson:gson:2.3'
compile 'commons-io:commons-io:2.4'
```

Currently, speech recognition is performed using Google's Android SDK, either on the client device or in the cloud. Recognized text is passed to the API.AI through HTTP requests.

The Authentication is accomplished through setting the client access token and subscription key when initializing an AIConfiguration object. The client access token specifies which agent will be used for natural language processing, subscription key used for managing subscription info.

Using our own speech recognition

in this session we can use our own speech recognition or the text that we have text that we want to process as natural language. Once we've added the SDK library, we follow these steps:

1. Add this permission into the AndroidManifest:
    - android.permission.INTERNET
2. Create an instance of AIConfiguration, specifying the access token, locale, and recognition engine. You can specify any recognition engine, since that value will not be used.
3. Create an AIDataService instance using the configuration object.
4. Create the empty AIRequest instance. Set the request text using the method setQuery.
5. Send the request to the API.AI service using the method aiDataService.request(aiRequest).
6. Process the response.

Troubleshooting

If you get an error when trying to install app that says "INSTALL_FAILED_OLDER_SDK", then check you have Android SDK 19 and build tools 19.1 installed.

API.ai

Introduction And Concepts:

Api.ai is built on a number of concepts. They are as follows:

Agents correspond to applications. Once you train and test an agent, you can integrate it with your app or device.
Entities represent concepts that are often specific to a domain as a way of mapping natural language phrases to canonical phrases that capture their meaning.
Intents represent a mapping between what a user says and what action should be taken by your software.

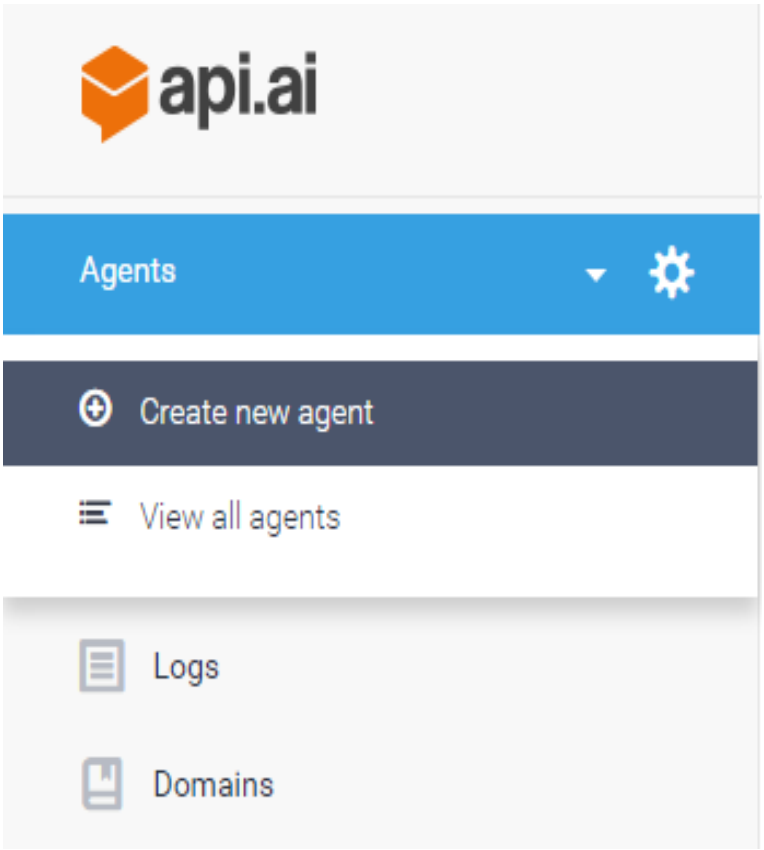
Contexts are strings that represent the current context of the user expression. This is useful for differentiating phrases which might be vague and have different meaning depending on what was spoken previously.
Aliases. When an entity is used in an expression, it will also have an alias, which acts like a variable name so that it can be referenced later.

Implementation:

Our goal is to make the process of creating and integrating sophisticated voice interfaces.
Here are the Steps to do so:

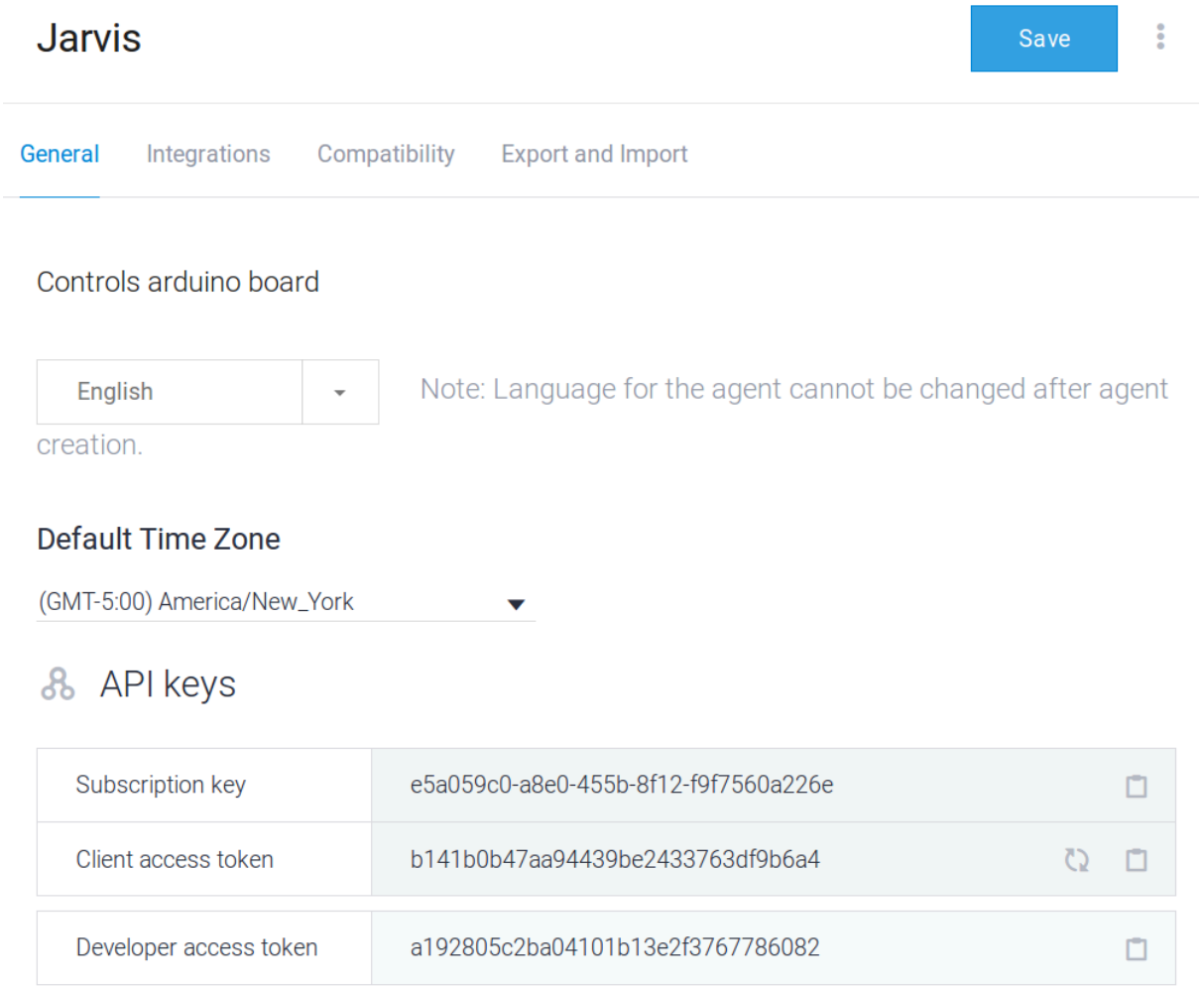
Step 1: Create agent

An Api.ai agent corresponds to an application or device you're voice-enabling.
You'll be prompted to create an agent right after you register at Api.ai.



After entering the name of your new agent, you will need to choose the language you want your app to communicate in. You can choose from a selection of 15 languages.

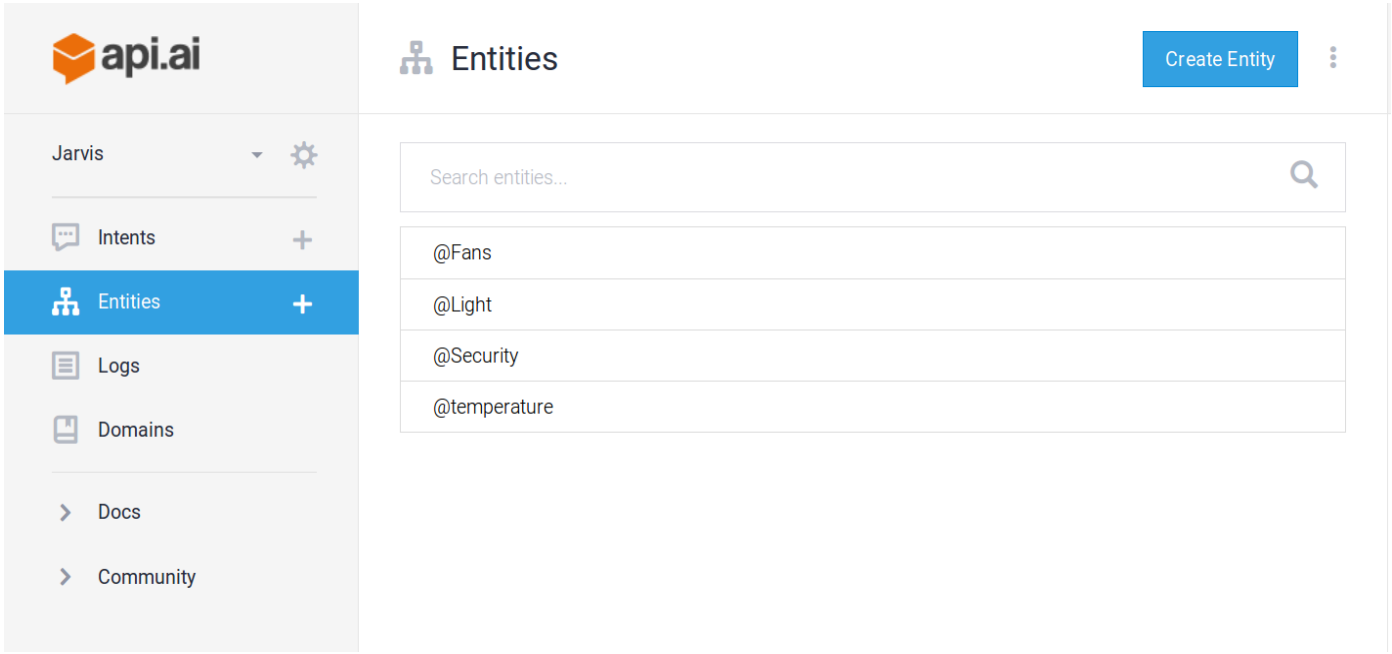
NOTE: The language you choose for your agent cannot be changed after creation. However, you can always create a new agent if you wish to work in an additional language.



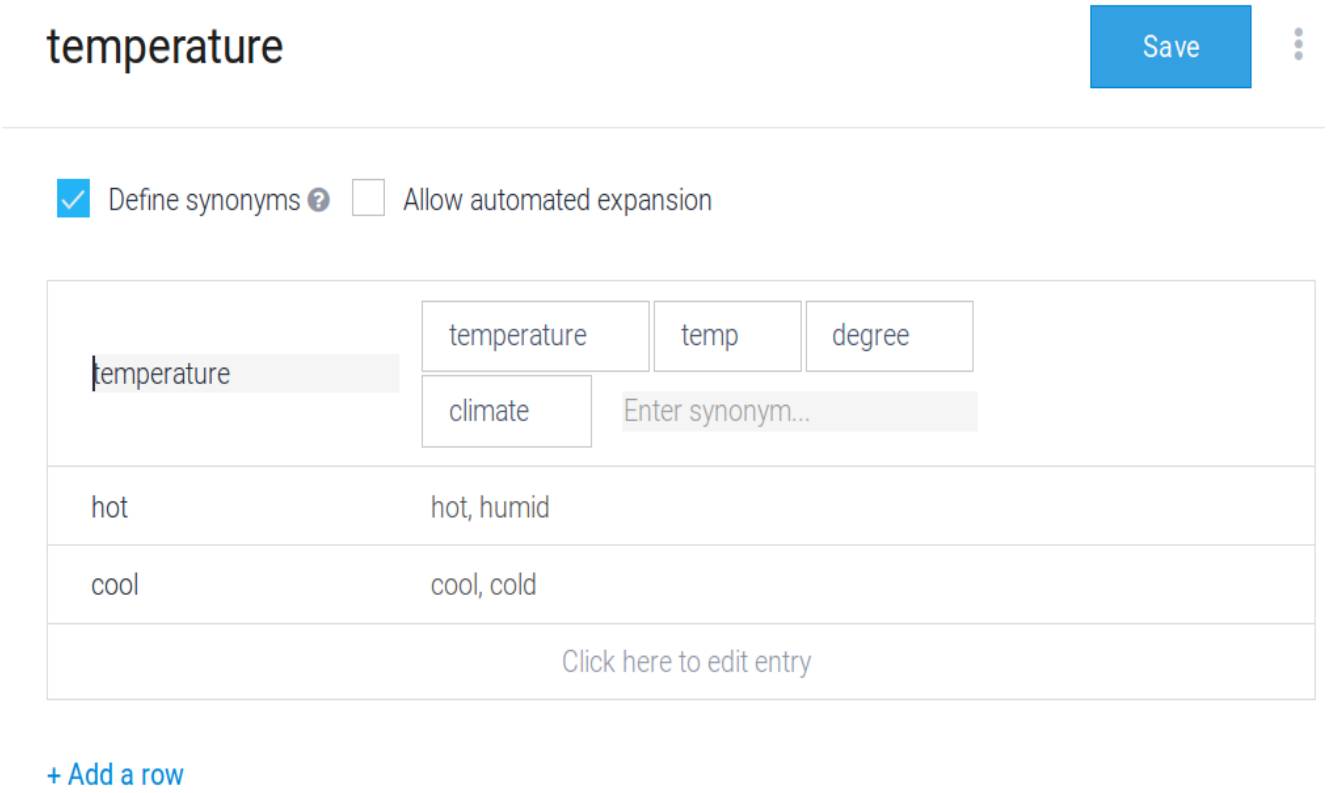
Once you have named your agent and selected a language, click the Save button.

Step 2: Create entities

Once you've created your agent, we would want to create a few entities to go with it. These will be stored in the Entities tab, located on the left of the Api.ai developer console.Here the figure shows us the Entity tab and the following Entities created by us.

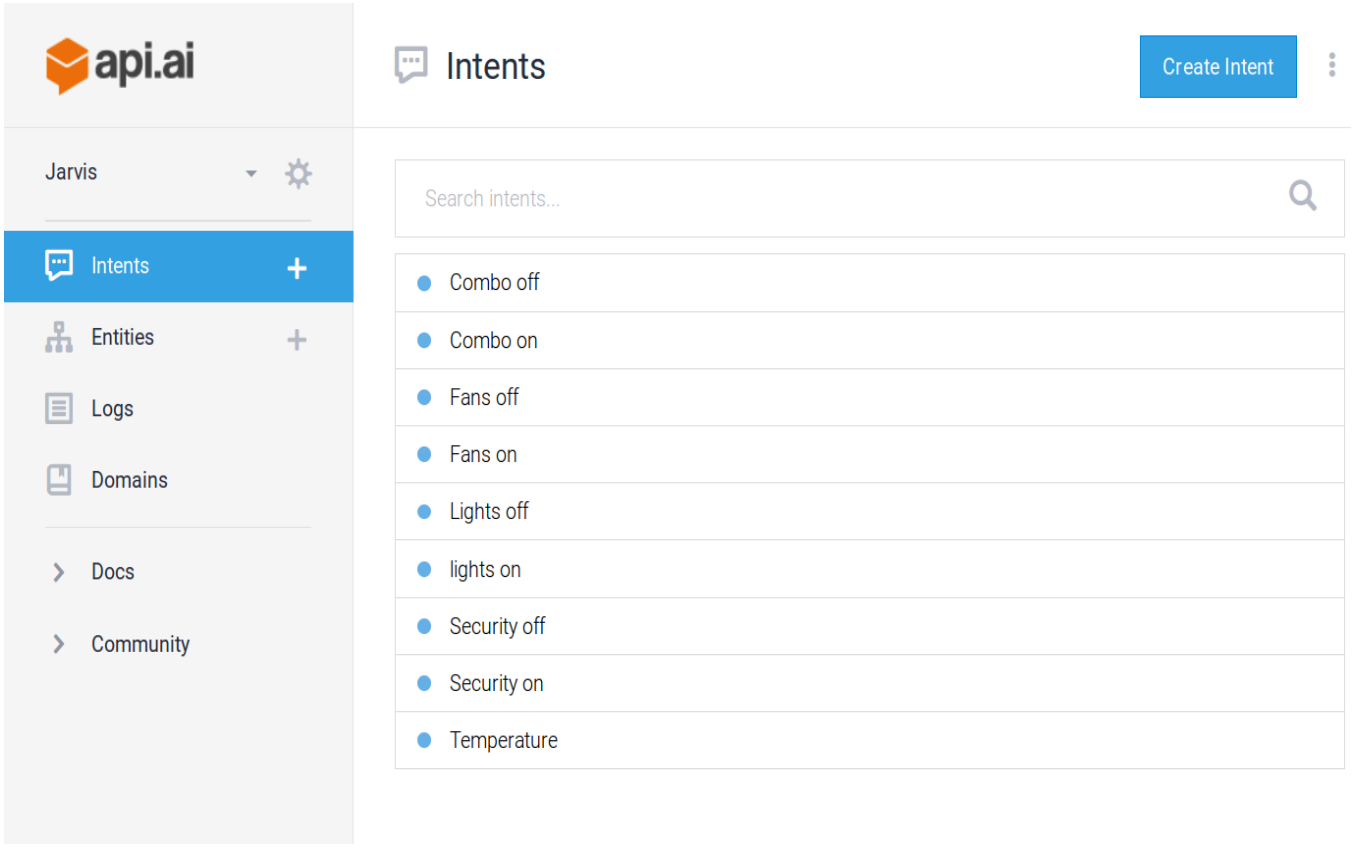


Entities are objects that are often specific to a domain as a means of mapping natural language phrases to the canonical phrases that capture their meaning.
The Entity titled “temperature” represents different types of temperature components and their synonyms. It defines the numerous way in which the user can request for the temperature and also the definition and components that describe the temperature.

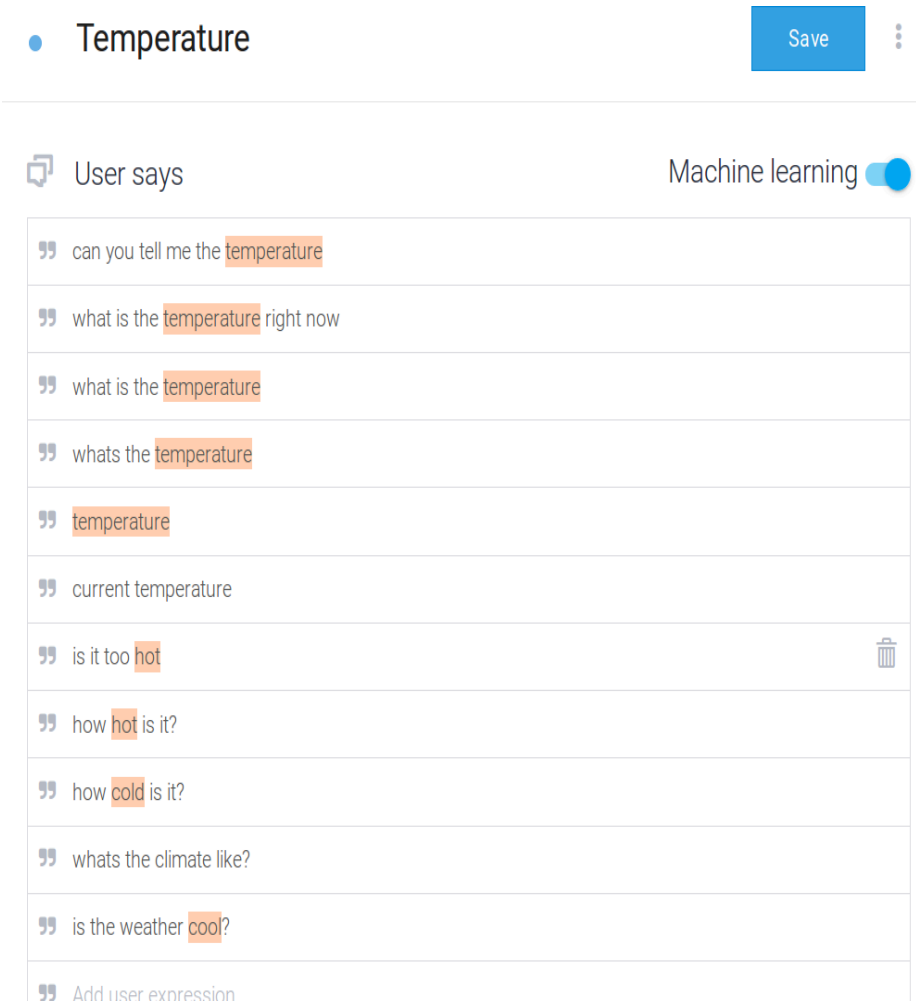


Step 3: Create intents

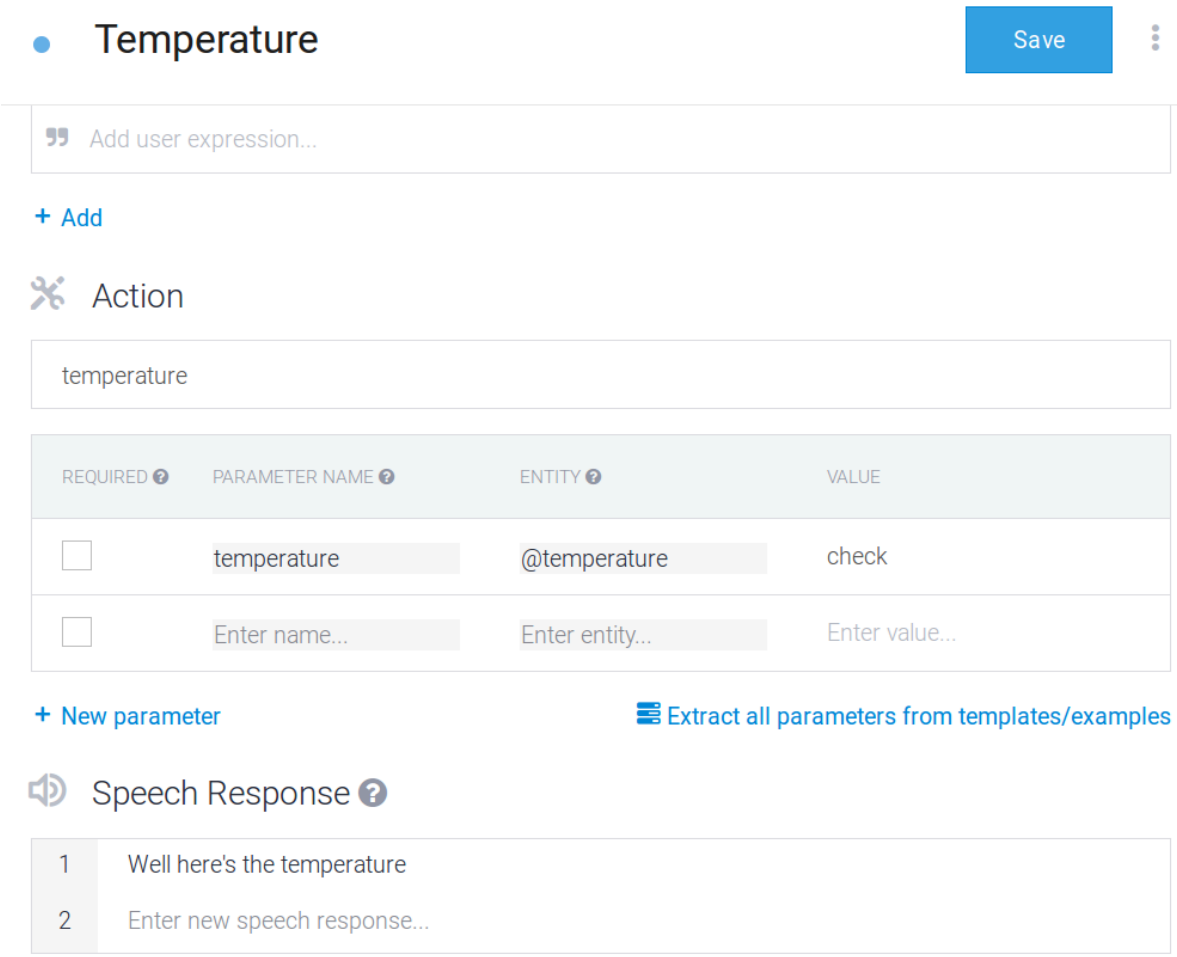
Once you created one or more entities, proceed to the Intents tab.
An intent maps all kinds of user requests to an action.
Here are all the different intents created for the backend processing which carries out an action.



Below is the example intent , which is designed to understand some basic requests from the user.It contains the possible inputs to the system.
The name of the intent is Temperature.All the highlighted keywords indicate that they belong to the @temperature entity.The user says tab contains all the possible requests the user can enter and their various combinations.

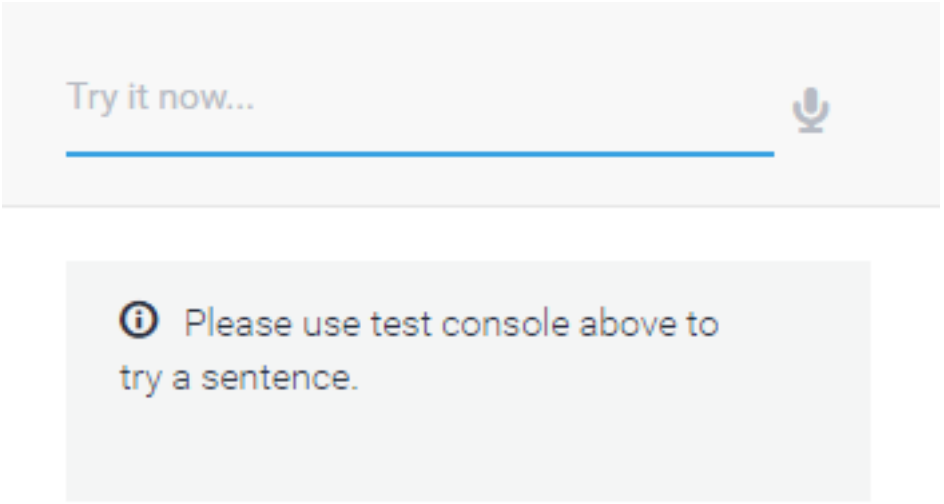


We then record the corresponding actions required to be taken and the different responses we would like once the action is completed.With the action we specify the different parameters we require and the values that have to be passed.
We also mention the entity that the Intent belongs to.(Here ite entity is @temperature the parameter specified is temperature and the value is checked)
The value of the parameter is then caught by the javascript within our webpage which can perform the respective action.

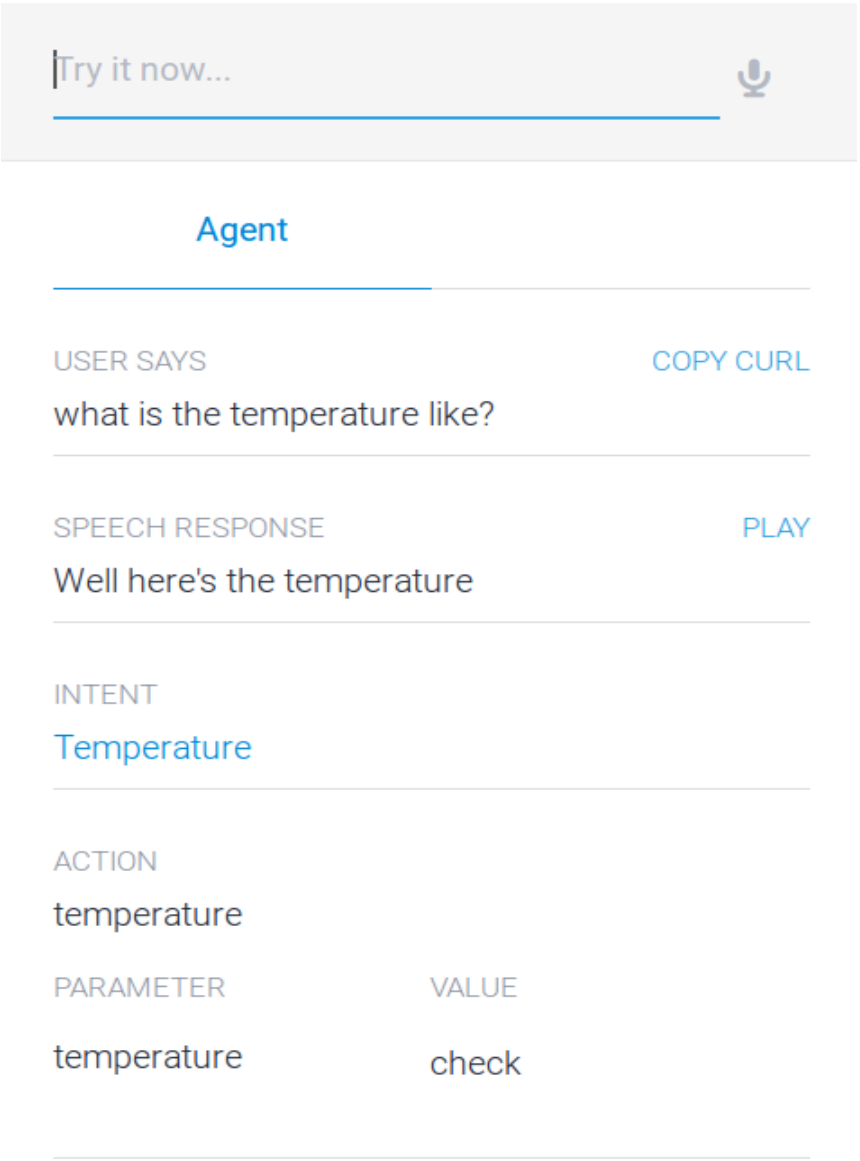

Temperature
Save
Add user expression...
+ Add
Action
temperature
REQUIRED
PARAMETER NAME
ENTITY
VALUE
temperature
@temperature
check
Enter name...
Enter entity...
Enter value...
+ New parameter
Extract all parameters from templates/examples
Speech Response
1
Well here's the temperature
2
Enter new speech response...

Step 4: Test and train your agent

In the top right corner of the Api.ai developer console, you can find the test console, where you can try how well your agent understands you already.



If your request is recognized, you'll see which intent was used to process it and what information was extracted.



If your request was not processed, you can add it to one of your intents right away or use the Logs feature later to find all the unrecognized inputs and use them to train your agent

Step 5: Integrate

Once your agent is ready, integrate it into your app or device. We have SDK's for all most popular platforms.

The response of the API.ai

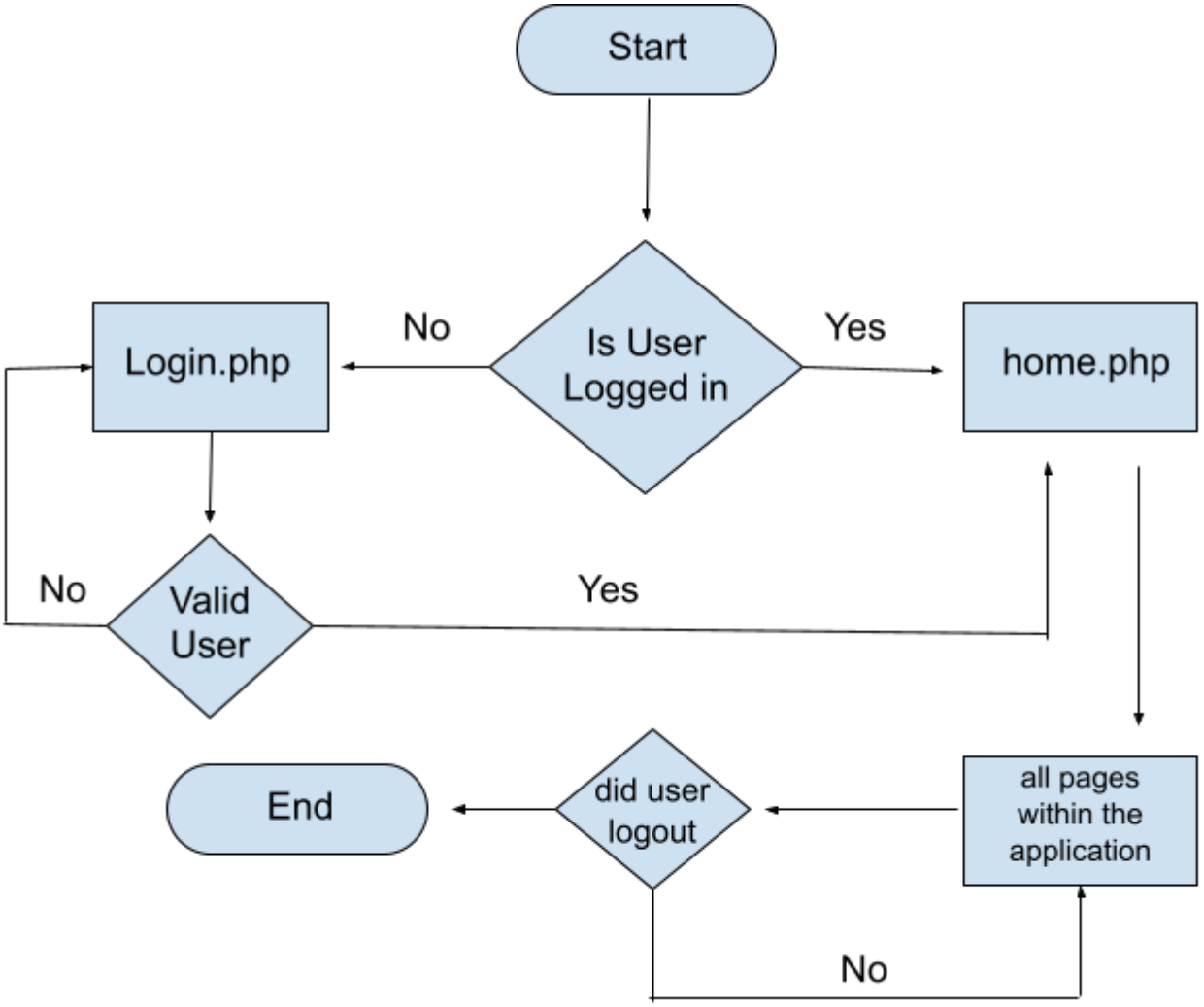
The api returns a JSON object which can then be read by the javascript within our web application.
The response in the form of the JSON object for temperature is as follows:

```
{
  "id": "98dab057-f230-4e97-b6c6-9207c0007213",
  "timestamp": "2016-02-28T14:55:42.443Z",
  "result": {
    "source": "agent",

//users request
    "resolvedQuery": "what is the temperature like?",/    "action":
"temperature",
    "actionIncomplete": false,
    "parameters": {
      "temperature": "check"
    },
    "contexts": [],
    "metadata": {
      "intentId": "79824460-b5cd-469b-874e-b67e1c3dd956",
      "intentName": "Temperature"
    },
    "fulfillment": {
//The response
      "speech": "Well here's the temperature"
    },
    "score": 0.3333333333333333
  },
  "status": {
    "code": 200,
    "errorType": "success"
  }
}
```

The fulfillment attribute contains the speech response which is caught by the javascript after the query has been executed successfully.

The back end android code can be found at the end of this book.

Module 7:User based Customization in Jarvis



Objective :

The objective of making user login is to differentiate between different users who controls the same house and in order to have a admin user who can add new users or block existing ones hence disallowing them from controlling the house anymore. The block is temporary and can be removed at any time when the admin pleases to.

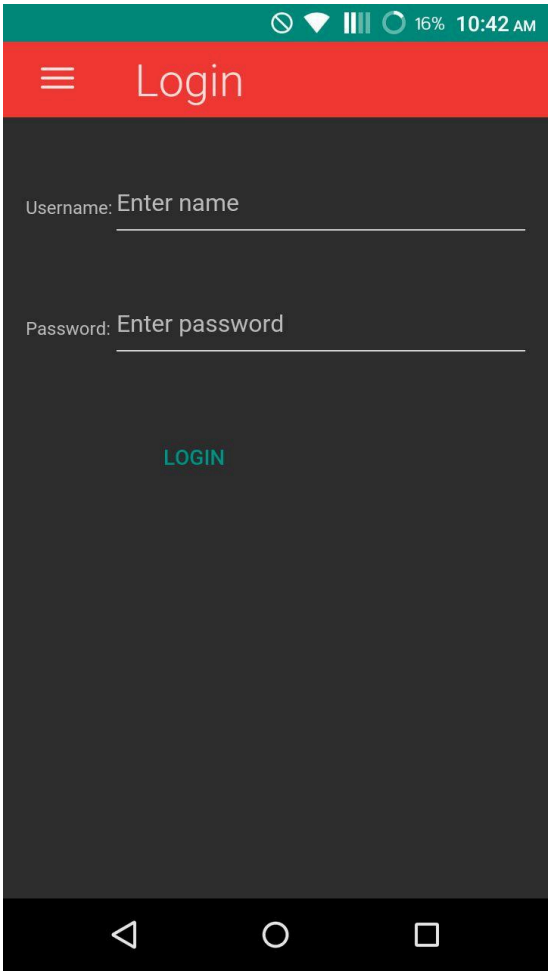
The main motive was to create different user accounts which was accomplished with the help of cookies.

A Cookie is a small piece of data sent from a website and stored in the user's web browser while the user is browsing. Every time the user loads the website, the browser sends the cookie back to the server to notify the user's previous activity.
Moreover Cookies provide a method by which web servers to know whether the user is logged in or not.

When the app is first open, by default the home page (AI module) is loaded. The home page checks if a cookie that was left by it exists. If not, it redirects the user to a login page. If the user is authenticated, two cookies with user's name and role is sent to the app and then the user is forwarded to the home page, now since the home page can find the cookie , it validates the user and allows access. From here all the other pages that the user visits checks if the cookie exists otherwise the user is redirected to the login page

Code used in all the pages including the home page

```
<?php
   if(!isset($_COOKIE["jcook_name"]))
     header("Location : login.php");
?>

Login Page code

<?php
   $username = $_POST['username'];
   $password = $_POST['password'];
//code to check these variables with
user database

$query = "Select * FROM user_account
WHERE username='".$post_username."'
AND password='".$post_password."'";
$result=mysqli_query($con,$query);
$row =  mysqli_fetch_array($result);
```



```
if(!empty($row['username'])&&!empty($row['password']))
{
  setcookie("jcook_name", $row['name'], time() + (86400 * 30), "/");
  setcookie("jcook_role", $row['role'], time() + (86400 * 30), "/");
  header("Location: home.php");
  exit;
}
else
{
      $login_status = 0;
}

?>
```

Now the user has the option to logout from any of the pages, this deletes the cookies that was previously left.
Therefore every time a user logs in into the application a new cookie is created else existing cookies are used.

The database that the php checks to verify if the user exists look like below

id	name	username	password	role	isAllowed
1	Jobin	jobinlawrance	696969	admin	1
2	Joanna	joannacerejo	969696	user	1
3	Jibin	jibinlawrnace	526272	user	0

Here the php page checks if the user exists and if he does then the user's name and role is stored in the cookies.

Now the admin has the option to block a specific user (explained in detail in admin module) upon which the user's switch control page becomes disabled, disallowing him from making any changes.

This is done by checking the " isAllowed" variable specific to the particular user and then disabling the buttons using jquery.

Therefore the switch control page consists of the following

```
<?php
include_once 'configure_db.php';
 $name = $_COOKIE["jcook_name"];
 $con = mysqli_connect($host,$user,$pass,$databaseName);
if (mysqli_connect_errno())
       {
       echo "Failed to connect to MySQL: " .
mysqli_connect_error();
       }
 $query = "Select `isAllowed` FROM user_account WHERE
name='".$name."'";
 $result=mysqli_query($con,$query);
 $row =  mysqli_fetch_array($result);
?>
```

we first check if the cookie exist, if it does we check the value of the “isAllowed” variable to that specific user. If it is 0, then we disable the button’s using the following jquery

```
var x = <?php echo $row['isAllowed'];?>;
if(!x)
{
          $("#fan1").attr('disabled','disabled');
          $("#light1").attr('disabled','disabled');
          Android.showToast("You have been blocked by admin");
      }
```

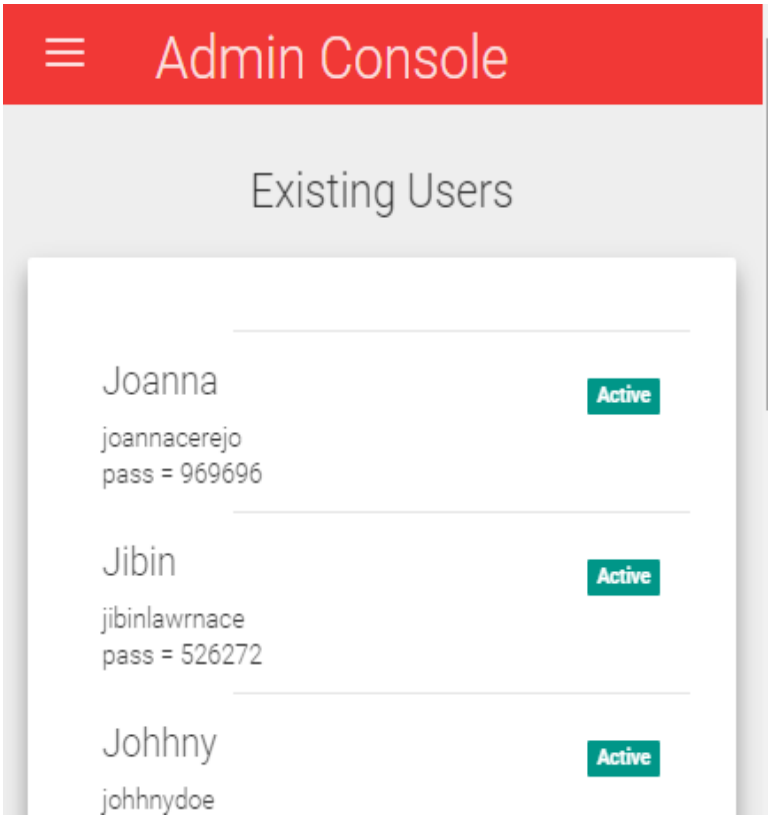
This similar process is applied to the security module so that the admin can also keep a track of the users who are eligible to switch on or off the security of the house.

Module 8: The Admin Console

objective :

The main purpose of this module is to create a master Slave relationship between the Admin and all its user where the Admin is the Master and all the other users are its Slaves.

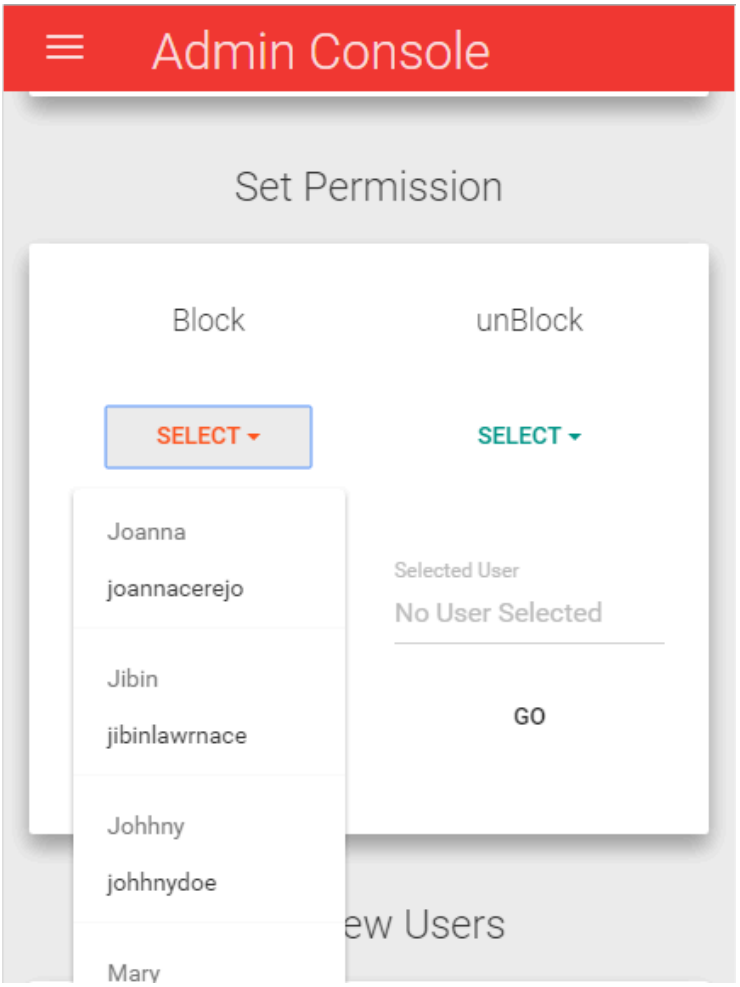
The admin can perform management of its users within the Admin console.
The admin console is a tab in the navigation menu that is only available to the admin.
Once the admin opens the application He/She can login to the application using the admin username and password.
Once authenticated the admin can then move to the Admin Console which is shown below:



The admin can now see a list of all the users of the application along with their usernames and password. An Active Icon is also present next to all the users indicating that they are currently active and can use the application controls.

The Admin Controls:

Block/ Delete a user:
if for some reason the admin feels the need to block a user he can do so by selecting the name of the user within the Set Permission screen which is shown below:



It Consists of 2 options namely block and unblock and a dropdown list.
within the block tab, the dropdown list contains the name of only those users who are active.
If the admin wished to block any user he can do so by simply selecting the name of the user and then selecting go.
Within the unlock tab, the dropdown list contains the name of users who have been blocked only.

The drop down lists for the block menu and the unblock menu are created in the following manner:

```
<div class="row ">
      <div class="col-xs-6 text-center">
     <h4>Block</h4>
  <br>
      <?php
   $query="SELECT `name`,`username` FROM `user_account` WHERE name
NOT IN ('".$_COOKIE['jcook_name']."') AND `isAllowed`=1";
                                  $result=mysqli_query($con,$query);

 ?>
<div class="dropdown">
<button class="btn btn-warning" id="blck_menu" type="button"
data-toggle="dropdown">Select
<span class="caret"></span></button>
  <ul class="dropdown-menu blck_menu" role="menu"
aria-labelledby="blck_menu">

<?php

while($row=mysqli_fetch_array($result))
{
  echo '<li class="dropdown-header"><h5>'.$row["name"].'</h5></li>';
echo '<li><a>'.$row["username"].'</a></li>';
echo '<li class="divider"></li>';
}
?>
</ul>
    </div>

                                     <form
action="setUser_permission.php" method="post">
                                        <div class="form-group
text-left">
 <label class="control-label" for="disabledInput">Selected
User</label>
 <input class="form-control blck_in" id="disabledInput"  type="text"
name="uname" placeholder="No User Selected" readonly>
   </div>
<input type="hidden" name="type" value="blck" />
  <button class="btn btn-default">Go</button>
```

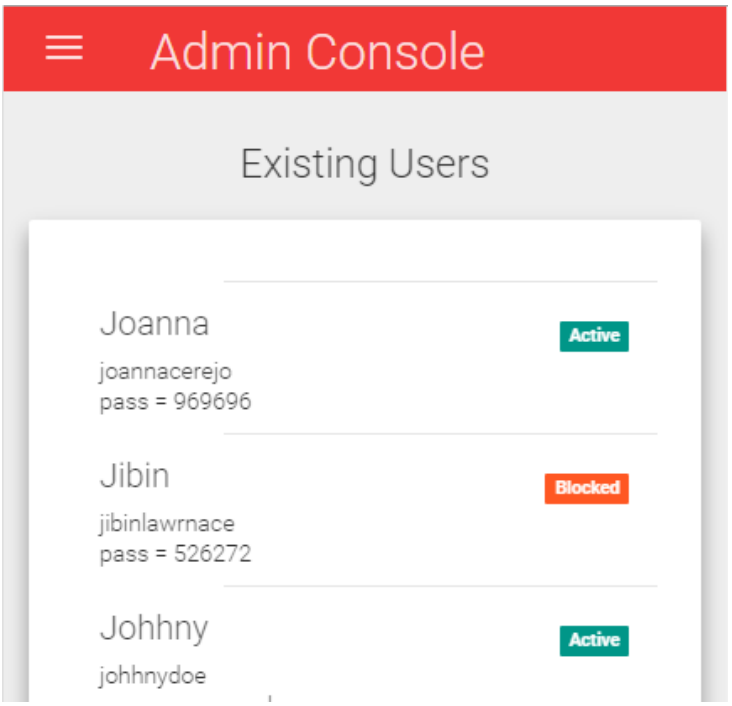
```
   </form>
```

The unblock Html code is as follows:

```
</div>
<div class="col-xs-6 text-center">
<h4>unBlock</h4>
<br>
<?php
                                                    $query="SELECT
`name`,`username` FROM `user_account` WHERE name NOT IN
('".$_COOKIE['jcook_name']."') AND `isAllowed`=0    ";

$result=mysqli_query($con,$query);
?>

<div class="dropdown">
<button class="btn btn-primary" id="ublck_menu" type="button"
data-toggle="dropdown">Select
<span class="caret"></span></button>
<ul class="dropdown-menu ublck_menu" role="menu"
aria-labelledby="ublck_menu">

     <?php
  while($row=mysqli_fetch_array($result)){
                                                                  echo
'<li class="dropdown-header"><h5>'.$row["name"].'</h5></li>';
                                                                  echo
'<li><a>'.$row["username"].'</a></li>';
     echo '<li class="divider"></li>';
      }
?>
</ul>
</div>
<form action="setUser_permission.php" method="post">
   <div class="form-group text-left">
<label class="control-label" for="disabledInput">Selected
User</label>
 <input class="form-control ublck_in" id="disabledInput"
name="uname"type="text" placeholder="No User Selected" readonly="">
</div>
<input type="hidden" name="type" value="ublck" />
   <button class="btn btn-default">Go</button>
                </form>
           </div>
```

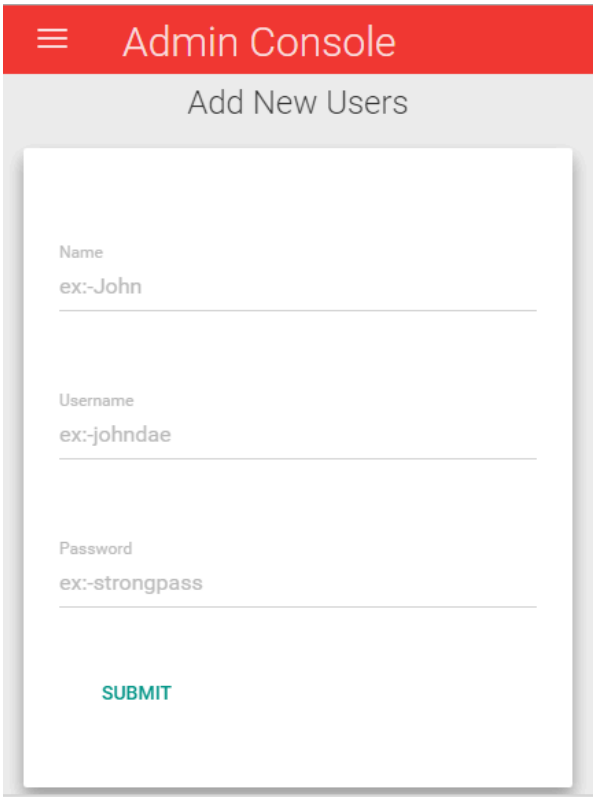
```
    </div>
</div>
```

Once the Admin blocks a user the Existing users List reflects the changes automatically.
For example when we block jibin the icon next to his name on the Existing users List changes automatically.
Once a user is blocked he/she can no longer make changes to the states of the appliances(switch on lights or fans).All the switches are blocked to this specific user.



Adding new Users:

The Admin also possesses the power to add new users ie new users who can use the application.This is down by adding the basic details of the users on the Add new Users within the console screen.
The following code consists of a simple form within a div tag containing 3 different input fields that have to be filled out by the admin and a submit button.
A screenshot of the following is given below



The code for the following is given below:

```
<div class="well well-lg">
<div class="row">
<div class="col-lg-12">
<form class="form-horizontal" id="sub_form" action="adduser.php"
method="post">                <div class="form-group">
   <label for="inputname" class="col-md-2
control-label">Name</label>

               <div class="col-md-10">
                           <input type="text"
name="name"class="form-control" id="inputname"
placeholder="ex:-John">
              </div>
          </div>
     <div class="form-group"
<label for="inputun" class="col-md-2
control-label">Username</label>
```

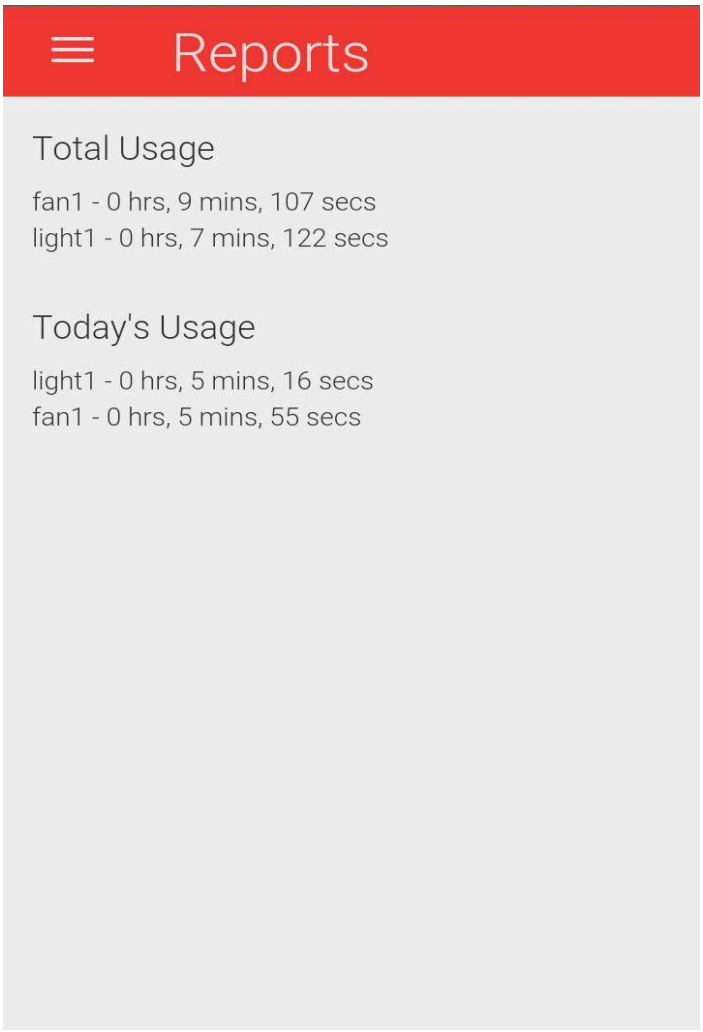
```
        <div class="col-md-10">
  <input type="text" name="uname"class="form-control"
id="inputun" placeholder="ex:-johndae">
 </div></div>
               <div class="form-group">
           <label for="inputps" class="col-md-2
control-label">Password</label>

            <div class="col-md-10">
  <input type="text" name="pass"class="form-control"
id="inputps" placeholder="ex:-strongpass">
           </div>
          </div>
          <br>
<button type="submit" class="btn btn-primary"
id="sub_btn">Submit</button>
</form>
```

Module 9:Reports

objective:

The main objective of this module is to create productive data that the user can use this data to analyze the amount of time for which the a particular object was kept on.

A new tab is created within the main menu during the course of this module.
This tab is known as a reports tab and it can be accessed by all users as well as the admin.



This tab informs all its users about the total amount of time for which an appliance was switched on thus helping the users to maintain a track of the current spent on every appliance.

The Reports tab shows the total usage of all the appliances and well as the usage on a daily basis.

The code to display the total usage is as follows:

```
<?php
                        $con =
mysqli_connect($host,$user,$pass,$databaseName);
                        if (mysqli_connect_errno())
                                {
                                echo "Failed to connect to MySQL: " .
mysqli_connect_error();
                                }
                        $query = "SELECT  `object` , `date` , SUM(
`hr` ) AS hr, SUM(  `mn` ) AS mn, SUM(  `sc` ) AS sc
                                        FROM  `timelog` GROUP BY
`object`";
                        $result=mysqli_query($con,$query);

                        echo '<div class="row">';
                             echo '<div class="col-xs-12">';
                                  echo "<h4>Total Usage</h4>";
                                    while ($row =
mysqli_fetch_array($result)) {
                                        echo $row['object']." -
".$row['hr']." hrs, ".$row['mn']." mins, ".$row['sc']." secs";
                                        echo "<br>";
                                    }

                             echo"</div>";
                        echo"</div>"

      ?>
```

The query is executed using the sum and group by functions
The sum calculates the sum of that particular object ie total time and the groupby then groups together the values in accordance to the date.
The code for daily reports is as follow:

```
<div class="row">
                        <div class="col-xs-12">
                             <h4>Today's Usage</h4>

                             <?php
                                  $query = "SELECT  `object` , `date`
, SUM(  `hr` ) AS hr, SUM(  `mn` ) AS mn, SUM(  `sc` ) AS sc FROM
`timelog` WHERE object = 'light1' GROUP BY  `date` ORDER BY  `date`
DESC LIMIT 1";
```

```
                                $result=mysqli_query($con,$query);
                                $row =  mysqli_fetch_array($result);
                                echo $row['object']." -
".$row['hr']." hrs, ".$row['mn']." mins, ".$row['sc']." secs";
                                    echo "<br>";
                                                                $query =
"SELECT  `object` , `date` , SUM(  `hr` ) AS hr, SUM(  `mn` ) AS mn,
SUM(  `sc` ) AS sc FROM  `timelog` WHERE object = 'fan1' GROUP BY
`date` ORDER BY  `date` DESC LIMIT 1";

$result=mysqli_query($con,$query);
                                                                $row =
mysqli_fetch_array($result);
                                                                echo
$row['object']." - ".$row['hr']." hrs, ".$row['mn']." mins,
".$row['sc']." secs";
                                                                     echo
"<br>";
                            ?>
                        </div>
                    </div>
```

within this piece of code the query remains the same for the daily report.The only thing that changes is that we make use of another function called the order by which orders the result set in accordance with the date.

Table List:

All the tables are Listed below:

1. states
2. checker
3. device_registration
4. temp
5. temp_current
6. user_account
7. timelog

Table Name: states

Purpose: To maintain the states of the Lights, Fans and security mode.

column_name	Datatype
id	varchar(50)
state	int(11)

Table Name: Checker

Purpose: To check if any changes were made within the table states

column_name	Datatype
id	int(11)
state	int(11)

Table Name: device_registration

Purpose: To maintain a record of all the mobile devices and their registration ids.

column_name	Datatype
reg_id	varchar(200)
datetime	timestamp

Table Name: temp

Purpose: To maintain a complete record of temperature values after every 30 minutes.

column_name	Datatype
id	varchar(50)
temperature	float
enDate	date
enTime	time

Table Name: temp_current

Purpose: To fetch the current value of the temperature.

column_name	Datatype
id	varchar(50)
temperature	float
enDate	date
enTime	time

Table Name:user_account

Purpose: To maintain the details of all the user accounts as well as the admin details. It also maintains the record of which of the user accounts are ligible to use the controls.

column_name	Datatype
id	int(11)
name	varchar(50)
username	varchar(50)
password	varchar(50)
role	varchar(50)

isAllowed	tiny Int

Table Name: timelog

Purpose: To maintain data for the Record tab.

column_name	**Datatype**
id	int
object	varchar(50)
onTime	time
offTime	time
date	timestamp
timeused	time
hr	int(11)
mm	int(11)
sc	int(11)

Data entry form:

In our Application the users have the option to enter data in the following formats:

1. Text
2. Direct speech
3. Selections through buttons

1.Text:
When the user logs into the app he is then directed to a homepage which consists of a text box where the user can enter the text command.For example : Please switch on the lights, and the app would directly switch on the lights and respond back with a message.

2.Direct speech
The homepage also contains an icon through which a user can directly talk to the app.The app then recognizes the speech and then converts it into a text message automatically. Once this step is executed the lights are turned on.

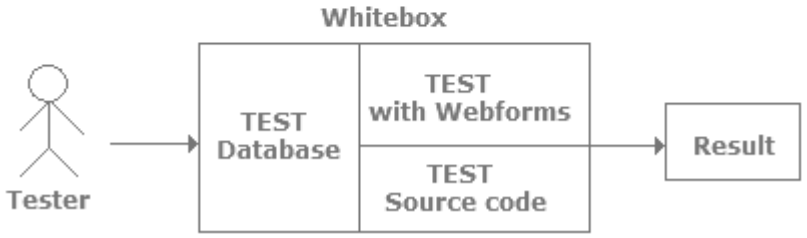
3.Selections through buttons
The user can physically navigate through the application and select the switch control tab.
Within this tab the user can then select the appliances he/she wants to switch on.

System testing:

We undertook white box testing method .

All tests were carried out after completion of each individual module and finally a complete integration test was carried out by combining all the codes.

Module 1 Testing:Web page testing



Once the web pages were created with the switches the basic functionality and the different changes made within the database were tested.
The main objective was to check if the web pages could directly make changes within the database using a simple php code.

State maintenance test:
Normally when the user switches on the light, exited the webpage and then loaded it again he would loose the previous state of all the appliances.
But this was not the case with our allpication.It still maintaied all the previos states even after an exit.

Module 2 testing physical connections

When the values of the given tables were found to be in accordance with the expected output a hardware test was performed.
It comprised of the testing the connections between the arduino board, the relay and the physical light bulb.
The hardware testing comprised of the connection of the arduino board and ethernet shield to the internet.It also consisted tests that were performed to check the connection between the board and our server on which the database is stored.

Thus The serial monitor was used for the purpose of displaying an appropriate output message on the screen to ensure that we have created a successful connection with the database

Module 3: Temperature

The main objective was to test if the webpage could fetch the data from the database and produce the results in the form of a graph
Another aspect on which test were performed was the sensitivity of the temperature sensor.
We also tested all the tables within the database

Module 4:Security

Similar to the previous module the motion sensor was first tested and then the database values.
The sensor was setup, its connections to the arduino board were then tested using the serial monitor to display a message.

Module 5:Pushy

Within this module we send a server side (message) notification and not a usual in-app notification using a third party server called Pushy.
Therefore we also had to test the reliability of Pushy.
The time required for the message to reach the mobile phone was also tested and found out to be ranging from few milliseconds to 2 seconds depending on the internet connection.

Module 6: AI Testing

Within this module we make use of a third party server called the API.ai which tries to make sence of the data sent to it with the help of the details(entities, intents, different vocabilary combinations of the users) provided by us and sends back a message to our application.
Thus all possible inputs that the user could request for from the application were tested weather it is in speech or text format.

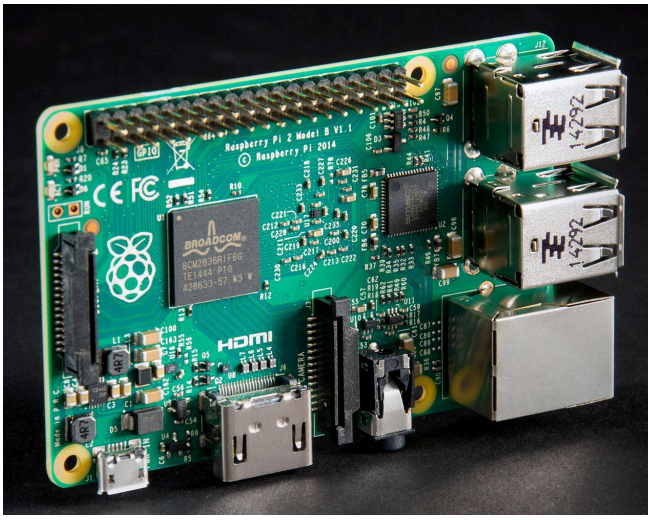
Module 7 and 8 :Different User Accounts and Admin Account

There are 2 types of accounts the user account and the admin account.
The only difference between the 2 accounts is that the admin can view and monitor all user accounts and also exempt users from using the application. ie blocking a user.The admin can also add new users.
Therefore the test were performed in such a manner that they spread through both the account types.Initial test were performed to check the authentication of the different users.
The admins capabilities were also tested separately.

Future Enhancements

Combining Arduino with Raspberry Pi

Even though the Arduino Mega is powerful and offers a plethora of I/O ports , it is limited by the fact that it is just a microcontroller at the end of the day. Here we can combine it with a small , cheap computer like the Raspberry Pi.
The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse.



But even though the Raspberry pi is more powerful than the arduino in terms of processor clock speed and memory, it cannot be used as a stand-alone device for home automation projects because it is not as flexible as the arduino when it comes to low level programming and usage of sensors. Arduino has a 'real-time' and 'analog' capability that the Pi does not: This flexibility allows it to work with just about any kind of sensor or chips while the raspberry pi allows complexity and high level programming.

So we can use Raspberry Pi to do all the complex programming part of the project and use Arduino to deal with the hardware sensors. We can further use the Raspberry Pi to host a local server that runs parallel with the hosted server. So when the user is at home the local server can be used, hence allowing control capabilities even without the Internet. It can be used to further broaden the AI capabilities like telling Jarvis to play a song from internet/youtube etc on the bluetooth speakers or start a movie on the screen connected to the Raspberry pi from Netflix etc. Hence the possibilities are more when both Raspberry Pi and Arduino are combined.

Separating Android App and Web Components.

Even though using Webview makes app development easy in Android by combining HTML and CSS for layout, it comes at a cost of execution speed. The app has to load the web page each time it opens and all the other pages has to loaded when used, this causes a speed issue when the internet speed is less. Moreover it can also cause security issues in login, since it is merely done with cookies. All this can be fixed by using the native code for layout and logic instead of webview. This drastically increases the speed, because each time the app is loaded only the button states(on/off) has to asynchronously taken from the server. The layout will be done using the native app libraries with XML. It also allows us to exclude the use of Javascript interface, the bridge that allows javascript to call native android functions, which further decreases the security risks.

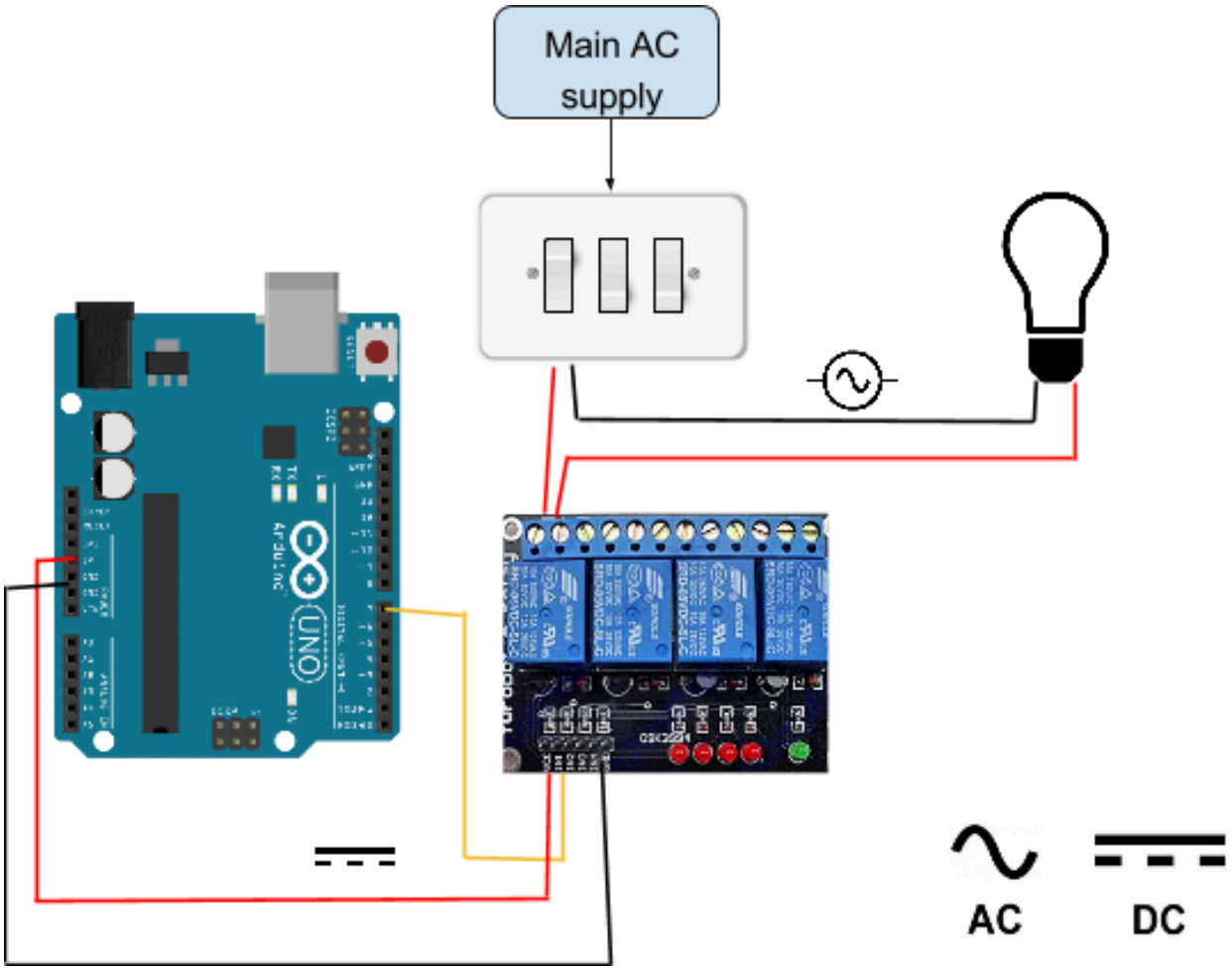
Using a PHP framework

According to the current situation the server side php files are all in folder without proper differentiation in terms of its purpose. We can use a PHP framework like Symfony 2 to properly organize all the files and properly deal with security. The main benefits of a framework are

- Organising Files And Codes - When we setup a PHP Framework, it already has a certain folder structure. Most of the Framework uses MVC (Model-View-Controller) guideline or structure for managing the code. MVC pattern allows to separate business logics and presentation layer of your website, making its structure consistent and maintainable.
- Enforcing of a good coding standard - Another effect of using Framework is maintaining coding standard in an efficient manner. Coding conventions are a set of guidelines for a specific programming language that recommend programming style, practices and methods for each aspect of a piece program written in this language. Using Framework makes it so much easier to code as you should.
- Utilities and Libraries - PHP is a great language for web development and provides countless number of tools and libraries.These frameworks contains a lot of libraries to make our job easier. All top PHP frameworks come with certain Libraries and Helpers, that can help us with:
    - Form Validation
    - SOAP/REST API
    - Caching Mechanism
    - Input/Output Filtering
    - Data Handle With JSON

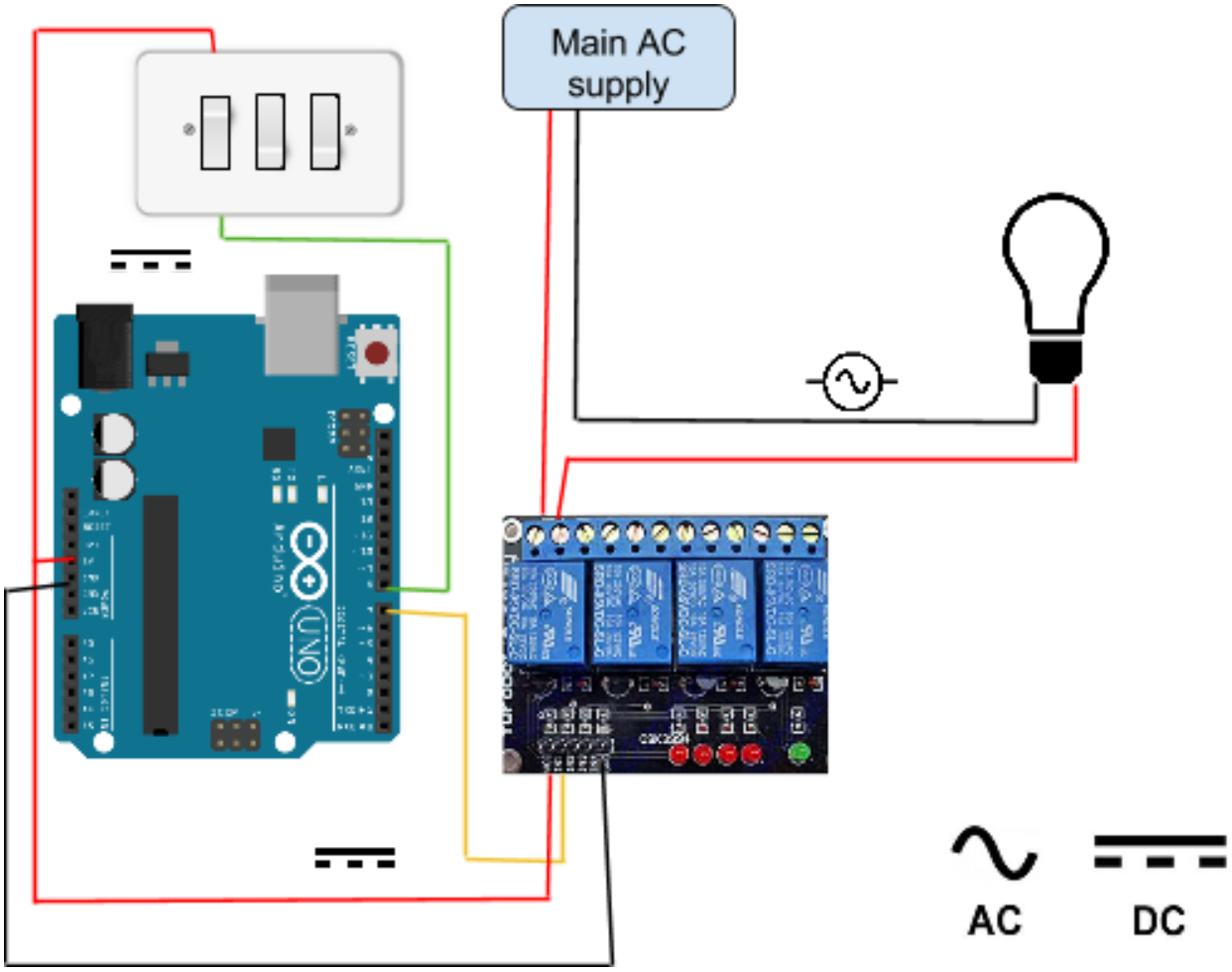
    - Database Abstraction
    - Session and Cookie Handling
    - Email, Calendar, Pagination, etc…
- Security Enhancements - With a framework, most of the work for code security can be done automatically. For example:
    - Central Authentication
    - Any value passed to database object gets filtered against SQL injection attacks
    - Central input validation and filtering
    - All HTML generating functions, such as form helpers and URL helpers filter the output automatically
    - Cross Site Request Forgery (CSRF) Protection
    - Session Management Security
    - Cross Site Scripting (XSS) Protection
    - Encrypting cookies automatically is only a matter of changing a config option and lot more…

Properly using the physical switches



The above diagram shows the existing circuit diagram for the switch control module, as it can be seen the relay board acts as a middle layer between the bulb and the physical switch. Hence the home automation works as long as the physical switch for the particular component is turned on.
But if the physical switch is turned off, the home automation system ceases to exist in terms of switch control. This is both a boon and curse at the same time. It gives the physical switch an upper hand in having more control over the state of the components. And if the components are connected in always-close port in the relay , the components can be controlled using the physical switches even if arduino is dead.
But the home automation system becomes dependant on the physical switches. To be fair to the automation system, there is a workaround to the existing circuitry. We can redesign the model in such a way that the arduino becomes the main control system

and the physical switches becomes one of the way to manipulate the states of the components along with the android app.



In the redesigned circuit, the physical switches are detached from the main supply and the components, and instead used as a DC switch for the arduino. Here the relay board is directly connected to the main AC supply and the physical switch is connected to the arduino board

We change the arduino code in such a way that now it also reads the physical switch values connected to its digital input pins. So now when the the physical switch is turned on, it sends a digital signal to the arduino and it subsequently switches the relay on, turning the appropriate component on. This way the physical switch acts like an alternate way to control the components apart from the mobile app.

There are some issues associated with this design though.

1. For every new component that we add, two digital pins will be required(one for the physical switch input and the other one for the relay output) on the arduino

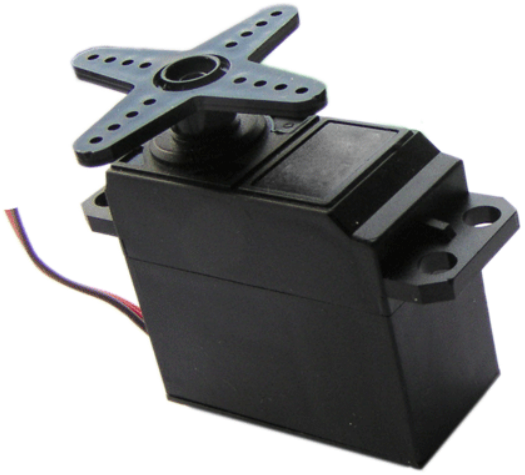
board. But in the case of Arduino Mega the number of digital pins are more compared to the Uno.So this won't be much of a problem to a great extent.

2. Since the arduino is the main controller , if it dies all of the components will either be on or off depending on which relay port it is connected (Always-open/Always-closed).That is none of the component can be controlled as long as the arduino is replaced or fixed.
3. We also need to change the security module control such that arduino keeps listening to the physical switches without missing the PIR sensor readings, otherwise with the existing code, the physical switches will also become useless when the security mode is enabled.

Other Functionalities:

1. Servo Motors for curtains

The basic concept is that we could manuputate the curtains on our window through the application according to our desire.To open or close the curtains we would require a motor that would be attached to the rim of the curtains thus allowing us to pull or push them.



A normal motor , either AC or DC does not run in position mode i.e. its position cannot be controlled. If you supply AC voltage to an AC motor (or DC voltage to a DC motor) , the motor will start rotating , and when you remove the voltage , the motor will stop. But the exact point at which it will stop cannot be controlled.

While a servomotor is a rotary actuator that allows for precise control of angular position, velocity and acceleration.The enabling us to control the precise amount of rotations.

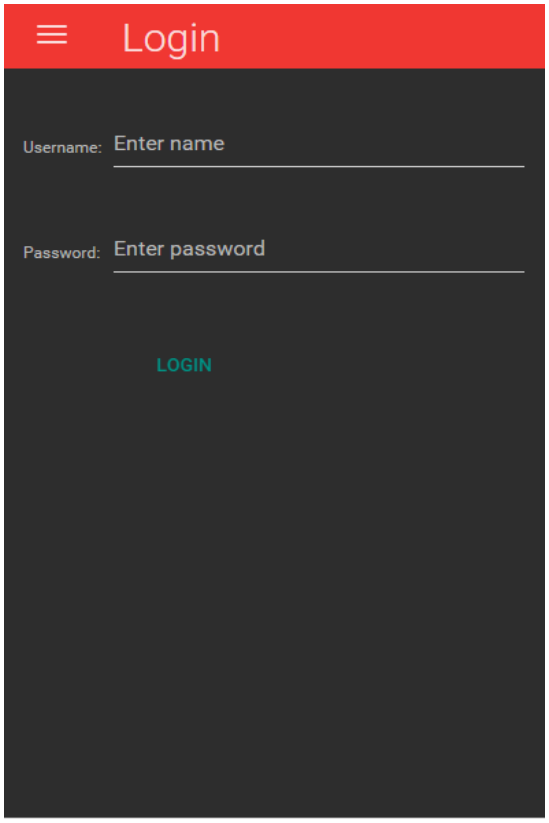
2.Gas Sensor MQ2

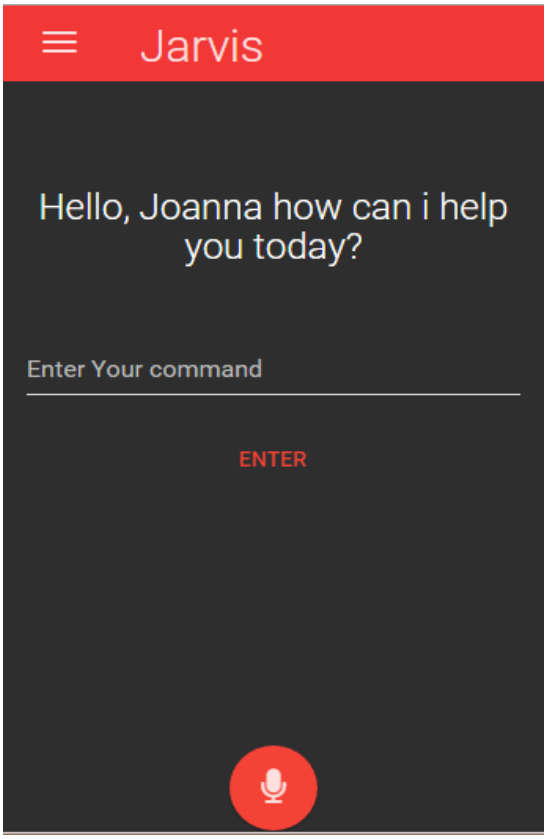
The main purpose to install this sensor is to alert the users in case of an emergency like a fire or a gas leak.
We could use a MQ2 sensor. This is a sensor that is not only sensitive to smoke, but also to flammable gas.

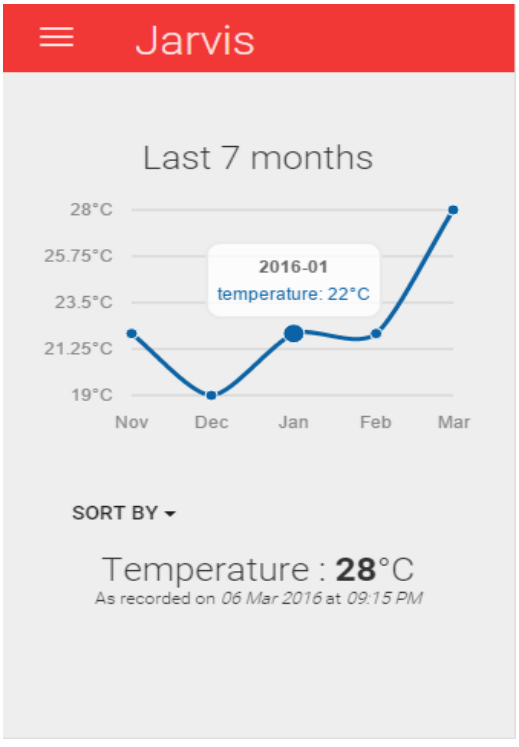
The MQ series of gas sensors use a small heater inside with an electrochemical sensor. They are sensitive for a range of gasses and are used indoors at room temperature.

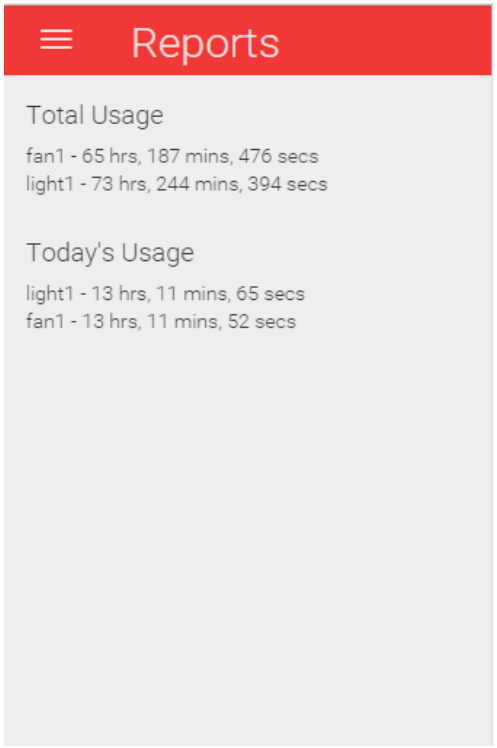
We could set up a notification to all the user or an alarm that would go out within the house once the smoke detector detects the presence of smoke or gas.
The output of this sensor is a analog output and thus it can be easily read by the analog input of the of the arduino making the sensor more suitable and compatible with the arduino.

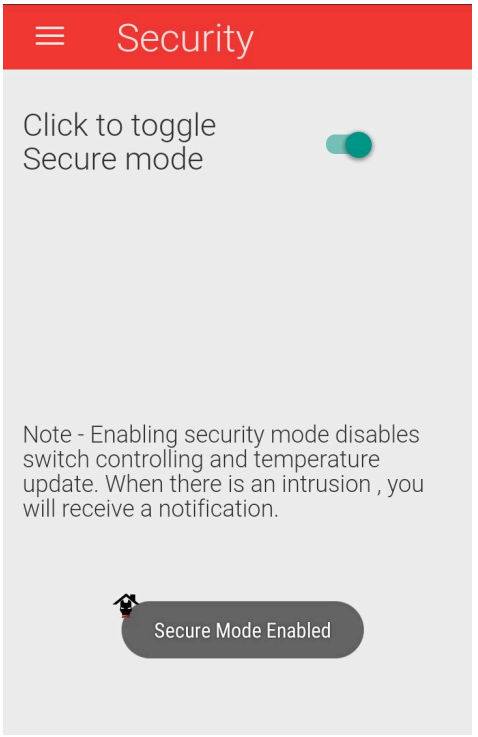
Screen Shots:

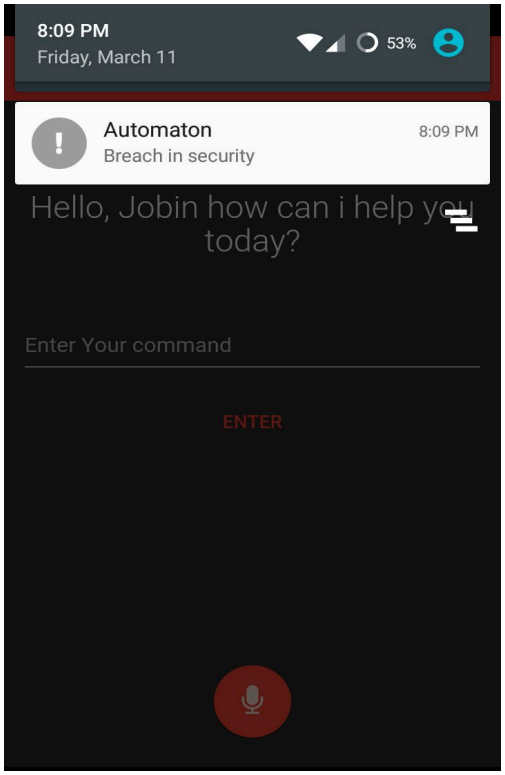


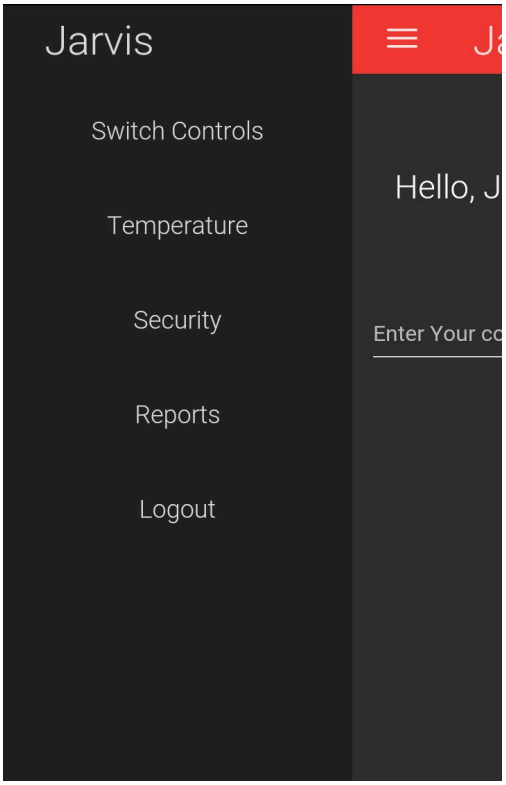




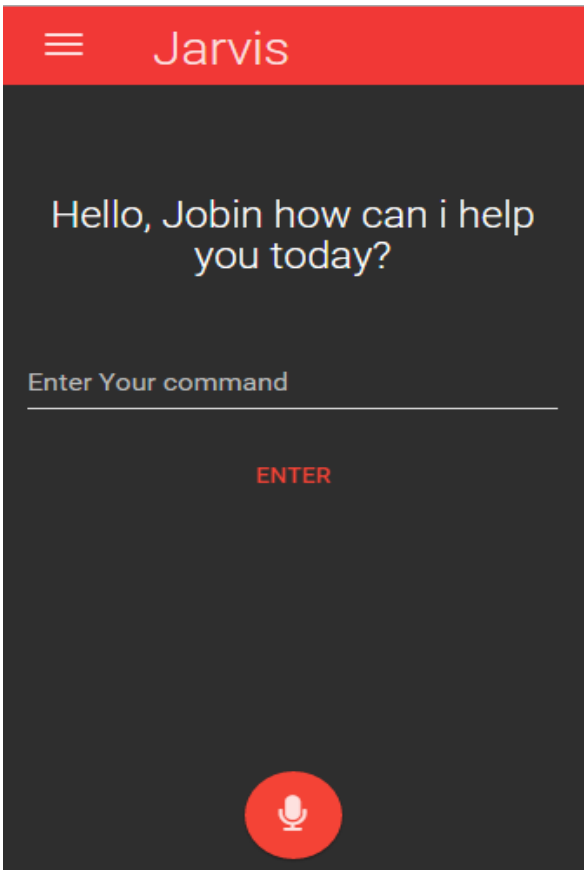

Reports
Total Usage
fan1 - 65 hrs, 187 mins, 476 secs
light1 - 73 hrs, 244 mins, 394 secs
Today's Usage
light1 - 13 hrs, 11 mins, 65 secs
fan1 - 13 hrs, 11 mins, 52 secs

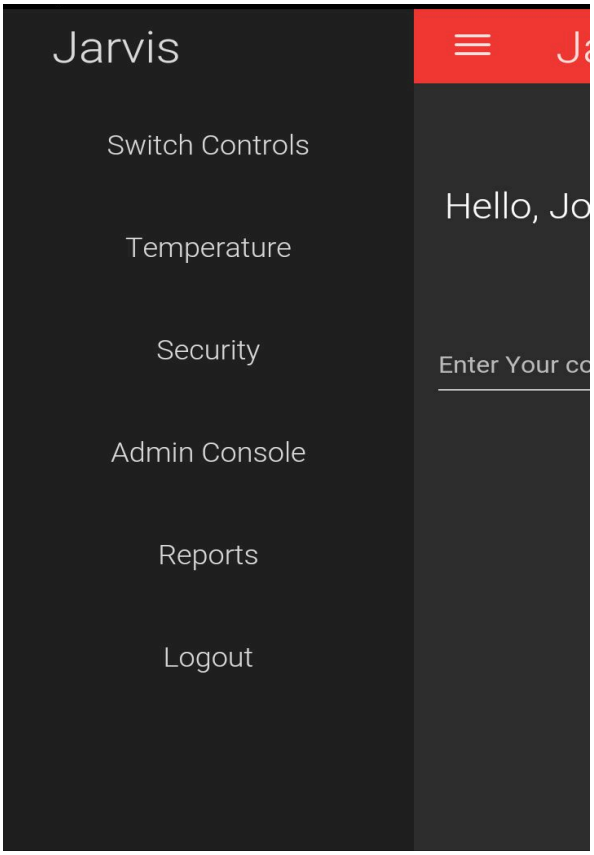

Security
Click to toggle
Secure mode
Note - Enabling security mode disables switch controlling and temperature update. When there is an intrusion , you will receive a notification.
Secure Mode Enabled

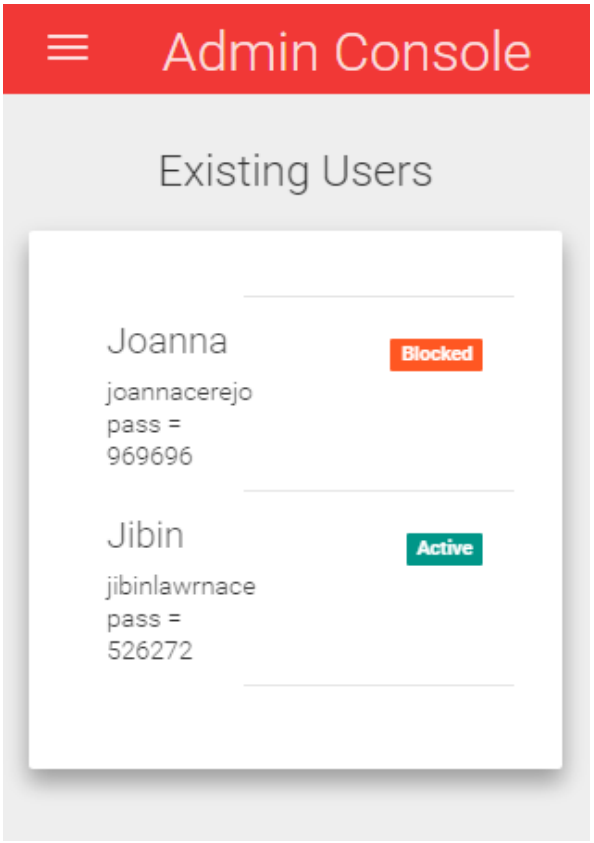

8:09 PM
Friday, March 11
53%
Automaton
Breach in security
8:09 PM
Hello, Jobin how can i help you today?
Enter Your command
ENTER

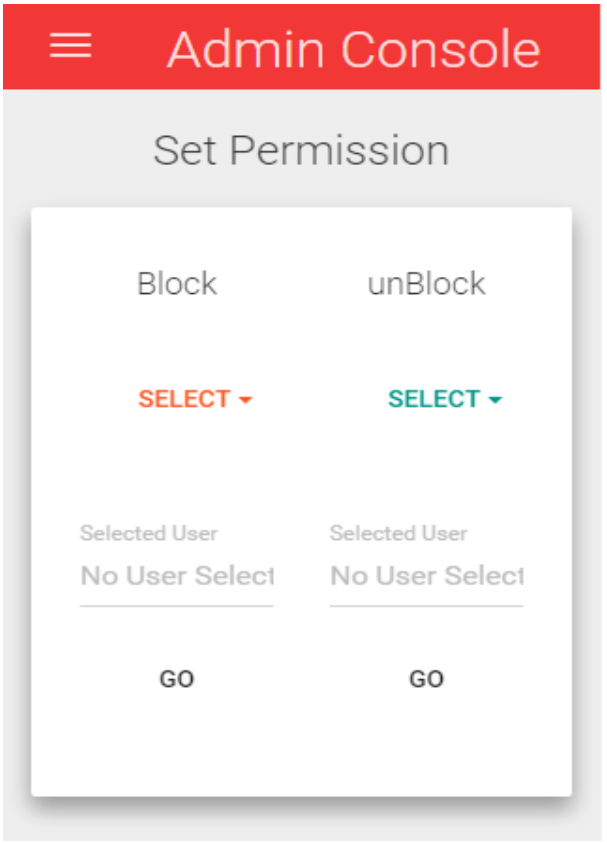

Jarvis
Switch Controls
Temperature
Security
Reports
Logout
Hello, J
Enter Your co

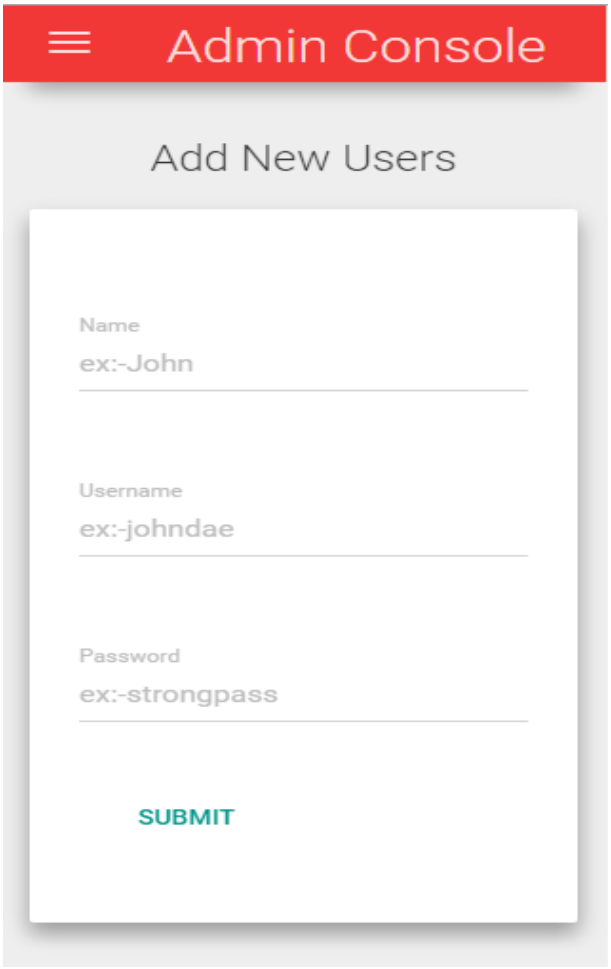
Admin console:






Admin Console
Existing Users
Joanna
Blocked
joannacerejo
pass =
969696
Jibin
Active
jibinlawrnace
pass =
526272


Admin Console
Set Permission
Block
unBlock
SELECT
SELECT
Selected User
No User Select
Selected User
No User Select
GO
GO


Admin Console
Add New Users
Name
ex:-John
Username
ex:-johndae
Password
ex:-strongpass
SUBMIT

Appendix:Source Code

Arduino code:

```
/*here we define the relay pins and make some change the output*/



#include "SPI.h"
#include "Ethernet.h"
#include "sha1.h"
#include "mysql.h"
#include <Time.h>
#include <TimeAlarms.h>



/* Setup for Ethernet Library */
byte mac[] = { 0xB4, 0x52, 0x7D, 0x1B, 0x1E, 0xF6 };
IPAddress ip(192,168,0,235);
IPAddress dns_serv(59,185,3,10);
IPAddress subnet(255,255,255, 0);
IPAddress gateway(192, 168, 0, 254);

const char checker_querry[] = "SELECT * FROM
   malharfe_arduino.checker WHERE id=0";
const char states_querry[] = "SELECT state from
   malharfe_arduino.states";
const char revert_querry[] = "UPDATE malharfe_arduino.checker SET
   state = 0 WHERE id = 0 ";
const char security_querry[] = "SELECT state from
   malharfe_arduino.states WHERE id='secutiy'";

/*Setup for the Connector/Arduino */
Connector my_conn; // The Connector/Arduino reference
char user[] = "malharfe_jojo";
```

```
char password[] = "jojo69";
;ge the output*/


#include "SPI.h"
#include "Ethernet.h"
#include "sha1.h"
#include "mysql.h"
#include <Time.h>
#include <TimeAlarms.h>


/* Setup for Ethernet Library */
byte mac[] = { 0xB4, 0x52, 0x7D, 0x1B, 0x1E, 0xF6 };
IPAddress ip(192,168,0,235);
IPAddress dns_serv(59,185,3,10);
IPAddress subnet(255,255,255, 0);
IPAddress gateway(192, 168, 0, 254);

const char checker_querry[] = "SELECT * FROM
   malharfe_arduino.checker WHERE id=0";
const char states_querry[] = "SELECT state from
   malharfe_arduino.states";
const char revert_querry[] = "UPDATE malharfe_arduino.checker SET
   state = 0 WHERE id = 0 ";
const char security_querry[] = "SELECT state from
   malharfe_arduino.states WHERE id='secutiy'";

/*Setup for the Connector/Arduino */
Connector my_conn; // The Connector/Arduino reference
char user[] = "malharfe_jojo";
char password[] = "jojo69";
;
```

```
int fan1_current_val = 0;
int light1_current_val = 0;

int fan1_prev_val = 0;
int light1_prev_val = 0;

int checker_state = 0;
int myArray[2];

int Relay1=7;
int Relay2=8;

float temp;
int tempPin = 0;

int Pir_Pin = 2;                // choose the input pin (for PIR
   sensor)
int pirState = LOW;              // we start, assuming no motion
   detected
int m_sense_val = 0;                      // variable for reading
   the pin status
int isSecure_state = 0;

char INSERT_CURRENT_TEMP[] = "CALL
   malharfe_arduino.current_temp(%s)" ;
char INSERT_TEMP[] = "CALL malharfe_arduino.temp(%s)" ;
char query[128];
char temperature[10];



EthernetClient client;
void setup() {
```

```
pinMode(Relay1, OUTPUT);
pinMode(Relay2, OUTPUT);
pinMode(Pir_Pin, INPUT);     // declare sensor as input

Serial.begin(9600);
while (!Serial);
Ethernet.begin(mac, ip, dns_serv,gateway,subnet);
delay(5000);

Serial.println(Ethernet.localIP());
Serial.println("Connecting...");
if (my_conn.mysql_connect("www.malharfest.org", 3306, user,
   password)){
Serial.println("Success connecting to malharfest.org");
Alarm.timerRepeat(15, readTemp_current);
Alarm.timerRepeat(30, readTemp);
readTemp();
readTemp_current();
} else {
Serial.println("Connection failed.");
}
}

void loop() {
Alarm.delay(0);
checker_check();
delay(250);

}

void  checker_ch
int fan1_current_val = 0;
int light1_current_val = 0;
```

```
int fan1_prev_val = 0;
int light1_prev_val = 0;

int checker_state = 0;
int myArray[2];

int Relay1=7;
int Relay2=8;

float temp;
int tempPin = 0;

int Pir_Pin = 2;                // choose the input pin (for PIR
   sensor)
int pirState = LOW;              // we start, assuming no motion
   detected
int m_sense_val = 0;                      // variable for reading
   the pin status
int isSecure_state = 0;

char INSERT_CURRENT_TEMP[] = "CALL
   malharfe_arduino.current_temp(%s)" ;
char INSERT_TEMP[] = "CALL malharfe_arduino.temp(%s)" ;
char query[128];
char temperature[10];



EthernetClient client;
void setup() {

pinMode(Relay1, OUTPUT);
pinMode(Relay2, OUTPUT);
```

```
pinMode(Pir_Pin, INPUT);     // declare sensor as input

Serial.begin(9600);
while (!Serial);
Ethernet.begin(mac, ip, dns_serv,gateway,subnet);
delay(5000);

Serial.println(Ethernet.localIP());
Serial.println("Connecting...");
if (my_conn.mysql_connect("www.malharfest.org", 3306, user,
   password)){
Serial.println("Success connecting to malharfest.org");
Alarm.timerRepeat(15, readTemp_current);
Alarm.timerRepeat(30, readTemp);
readTemp();
readTemp_current();
} else {
Serial.println("Connection failed.");
}
}

void loop() {
Alarm.delay(0);
checker_check();
delay(250);

}

void  checker_check(){
    Serial.println("Checking ...");
    my_conn.cmd_query(checker_querry);
    my_conn.get_columns();
```

```
  row_values *row = NULL;
  do {
  row = my_conn.get_next_row();
  // this is to check the value of checker from the database
  if (row != NULL) {

  checker_state = atol(row->values[0]);
  }
  my_conn.free_row_buffer();
  } while (row != NULL);
  my_conn.free_columns_buffer();
if(checker_state==1){
     my_conn.cmd_query(states_querry);
     my_conn.get_columns();
     row_values *row = NULL;
     int i = 0;
     do {

     row = my_conn.get_next_row();

     if (row != NULL) {
     myArray[i] = atol(row->values[0]);
     }
     i++;
     my_conn.free_row_buffer();
     } while (row != NULL);
     my_conn.free_columns_buffer();
     Serial.println(myArray[0]);
     Serial.println(myArray[1]);
     Serial.println(myArray[2]);
     secure_mode();
     relay_switching();

     my_conn.cmd_query(revert_querry);
```

```
  }
}

void secure_mode(){
 if(myArray[2]== 1){
   isSecure_state = 1;
   Serial.println("Inside motion sensor");
     while(true){

       m_sense_val = digitalRead(Pir_Pin);  // read input value
       if (m_sense_val == HIGH) {            // check if the input
   is HIGH

       if (pirState == LOW) {
       // we have just turned on
       Serial.println("Motion detected!");
       // We only want to print on the output change, not state
           if (client.connect("tedxstxaviersmumbai.com", 80)) {
             Serial.println("connected");
             // Make a HTTP request:
             client.println("GET /Jarvis/send_pushy.php
   HTTP/1.1");
             client.println("Host: tedxstxaviersmumbai.com");
             client.println("User-Agent: Mozilla/4.0 (compatible;
   MSIE 7.0; Windows NT 5.1)");
             client.println();
         }
           else {
           // kf you didn't get a connection to the server:
            Serial.println("connection failed");
         }
        pirState = HIGH;
       }
       } else {
```

```
  if (pirState == HIGH){
  // we have just turned of
  Serial.println("Motion ended!");
  // We only want to print on the output change, not state
  pirState = LOW;
  }
}
  my_conn.cmd_query(states_querry);
my_conn.get_columns();
row_values *row = NULL;
int i = 0;
do {

row = my_conn.get_next_row();

if (row != NULL) {
myArray[i] = atol(row->values[0]);
}
i++;
my_conn.free_row_buffer();
} while (row != NULL);
my_conn.free_columns_buffer();

 my_conn.cmd_query(states_querry);
my_conn.get_columns();
row_values *row1 = NULL;
int j = 0;
do {

row1 = my_conn.get_next_row();

if (row1 != NULL) {
myArray[j] = atol(row1->values[0]);
```

```
        }
        j++;
        my_conn.free_row_buffer();
        } while (row1 != NULL);
        my_conn.free_columns_buffer();
           if(myArray[2]==0){
             Serial.println("Out of security mode");
             break;
           }
           delay(2000);
      }
 }

}

void relay_switching(){


        light1_current_val=myArray[1];
        fan1_current_val=myArray[0];

          // for light1
         if(light1_current_val==1 && light1_prev_val==0){
           Serial.print("light1 = ");
           Serial.println(myArray[1]);
           Serial.println("Turning the Light On");
           digitalWrite(Relay1, HIGH);
           }
          else if(light1_current_val==0 && light1_prev_val==1){
            digitalWrite(Relay1, LOW);
            Serial.println("Turning the Light Off");
          }
          //for fan1
          if(fan1_current_val==1 && fan1_prev_val==0){
```

```
            Serial.print("fan1 = ");
            Serial.println(myArray[0]);
            Serial.println("Turning the FAN On");
            digitalWrite(Relay2, HIGH);
            }
           else if(fan1_current_val==0 && fan1_prev_val==1){
             digitalWrite(Relay2, LOW);
             Serial.println("Turning the fan Off");
           }


       fan1_prev_val = fan1_current_val;
       light1_prev_val = light1_current_val;



}

void readTemp_current(){
temp = analogRead(tempPin);
Serial.print("Inside the readTempCurrent : ");
temp = temp * 0.48828125;

dtostrf(temp, 1, 1, temperature);
sprintf(query, INSERT_CURRENT_TEMP,temperature);
Serial.print("every 1 min " );
Serial.println  (query);
my_conn.cmd_query(query);
}

void readTemp(){
temp = analogRead(tempPin);
Serial.print("Inside the readTemp : ");
temp = temp * 0.48828125;
```

```
dtostrf(temp, 1, 1, temperature);
sprintf(query, INSERT_TEMP,temperature);
Serial.print("every 5 min " );
Serial.println  (query);
my_conn.cmd_query(query);
}
```

HTML and PHP Codes:

home.php

```
<?php
    if(isset($_COOKIE["jcook_name"])){}
       else header('Location: login.php');

   ?>
   <html>
       <head>
         <title>Jarvis</title>
         <meta charset="utf-8">
         <meta http-equiv="X-UA-Compatible" content="IE=edge">

         <!-- Mobile support -->
         <meta name="viewport" content="width=device-width,
   initial-scale=1">

         <!-- Material Design fonts -->
         <link rel="stylesheet"
   href="http://fonts.googleapis.com/css?family=Roboto:300,400,500
   ,700" type="text/css">
         <link
   href="https://fonts.googleapis.com/icon?family=Material+Icons"
   rel="stylesheet">

         <!-- Bootstrap -->
         <link
   href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.m
   in.css" rel="stylesheet">


         <!-- Bootstrap Material Design -->
         <link href="dist/css/bootstrap-material-design.css"
```

```
rel="stylesheet">
      <link href="dist/css/ripples.min.css" rel="stylesheet">

      <!-- Dropdown.js -->
      <link
href="//cdn.rawgit.com/FezVrasta/dropdown.js/master/jquery.drop
down.css" rel="stylesheet">

      <!-- General Page style -->
      <link href="index.css" rel="stylesheet">
      <link rel="stylesheet" href="slidebars/slidebars.css">
      <!-- Custom Page style-->
      <link href="custom_css/jarvis.css" rel="stylesheet">

      <!-- Font Icon Style   -->
       <link href="fonts/css/fontello.css" rel="stylesheet">

        <!-- Font Icon Style   -->
       <link href="fonts/css/animation.css" rel="stylesheet">

      <!-- jQuery -->
      <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery
.min.js"></script>

      <meta name="viewport" content="width=device-width,
initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0,
user-scalable=no">

      <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/fo
nt-awesome.min.css">
    </head>
    <body>
```

```
        <nav class="navbar navbar-fixed-top custom-navbar
sb-slide border" role="navigation">
          <div class="container bg-nav">
            <button type="button" class="navbar-toggle
sb-toggle-left">

              <span class="icon-bar"></span>
              <span class="icon-bar"></span>
              <span class="icon-bar"></span>
            </button>
            <a href=""class="navbar-brand"><h2>Jarvis</h2></a>
            <!--/.nav-collapse -->
          </div>
          <!--/.container -->
        </nav>
        <div id="sb-site" class="bg-body">
             <div class="container">

                  <br>
                  <br>
                  <br>
                  <br>
                  <br>
                  <br>

                  <div class="row text-center white-text">
                       <div class="col-sm-12">
                            <h3>Hello, <?php echo
$_COOKIE["jcook_name"]?> how can i help you today?</h3>
                       </div>
                  </div>
                  <br>
```

```
                    <div class="row ">
                         <div class="col-sm-10">
                              <div class="form-group
label-floating">
                                <label class="control-label"
for="query">Enter Your command</label>
                                <input class="form-control"
id="query" type="text">
                              </div>
                         </div>
                         <div class="col-sm-2 text-center">
                              <input type="button" class="btn
btn-danger " id="text-btn" value="Enter">
                         </div>
                    </div>
                    <br><br>
                    <br><br>
                    <br><br>
                    <br><br>
                    <div class="row text-center">
                              <div class="col-xs-12">

                                   <button class="btn btn-danger
btn-fab"id="mic-btn">
                                        <i class="fa
fa-microphone"></i>
                                   </button>
                              </div>
                         </div>
               </div>

          </div>

          <div class="sb-slidebar sb-left sb-style-push">
```

```
          <!-- Your left Slidebar content. -->
          <div class="container">
             <nav class="navbar navbar-default"
role="navigation">
               <!-- Brand and toggle get grouped for better
mobile display -->
                <div class="navbar-header ">
                 <a class="navbar-brand text-center"
href=""><h2>Jarvis</h2></a>
                </div>

                <!-- Collect the nav links, forms, and other
content for toggling -->

                 <ul class="nav navbar-nav">
                   <li class=" text-center"><a
href="jarvis.php"><h4>Switch Controls</h4></a></li>
                   <li class="text-center"><a
href="temp.php"><h4>Temperature</h4></a></li>
                   <li class="text-center"><a
href="security.php"><h4>Security</h4></a></li>

                   <?php
                       if($_COOKIE['jcook_role']=="admin"){
                            echo '<li class="text-center"><a
href="admin.php"><h4>Admin Console</h4></a></li>';
                       }
                   ?>
                   <li class="text-center"><a
href="reports.php"><h4>Reports</h4></a></li>
                   <li class="text-center"><a
href="logout.php"><h4>Logout</h4></a></li>
                 </ul>
```

```
            </nav>
          </div>
        </div>


    <!-- Twitter Bootstrap -->
    <script
src="//cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.6/j
s/bootstrap.min.js"></script>

    <!-- Material Design for Bootstrap -->
    <script src="dist/js/material.js"></script>
    <script src="dist/js/ripples.min.js"></script>

    <!--Slide bars-->
    <script src="slidebars/slidebars.js"></script>


    <script>
      $.material.init();
      $.slidebars();
    </script>
    <script type="text/javascript">
                        $(document).ready(function() {


                                $("#mic-btn").click(function(){


                                  alert("works");
                                  Android.showSpeech();

                                });
```

```
    $("#text-btn").click(function(){

                                        if($('#query').val()!=null
    ||$('#query').val()!="")

    Android.chat_API($('#query').val());

                                    });


                            });
                        </script>

        <!-- Sliders -->
        <script
    src="//cdnjs.cloudflare.com/ajax/libs/noUiSlider/6.2.0/jquery.n
    ouislider.min.js"></script>

        <!-- Dropdown.js -->
        <script
    src="https://cdn.rawgit.com/FezVrasta/dropdown.js/master/jquery
    .dropdown.js"></script>
        <script>
          $("#dropdown-menu select").dropdown();
        </script>
</body>
</html>
```

Jarvis.php

```
<?php
include_once 'configure_db.php';
```

```
$uname = $_COOKIE["jcook_uname"];
$con = mysqli_connect($host,$user,$pass,$databaseName);
if (mysqli_connect_errno())
     {
     echo "Failed to connect to MySQL: " . mysqli_connect_error();
     }
$query = "Select `isAllowed` FROM user_account WHERE
   name='".$uname."'";
$result=mysqli_query($con,$query);
$row =  mysqli_fetch_array($result);
?>
<html>
   <head>
     <title>Jarvis</title>
     <meta charset="utf-8">
     <meta http-equiv="X-UA-Compatible" content="IE=edge">

     <!-- Mobile support -->
     <meta name="viewport" content="width=device-width,
   initial-scale=1">

     <!-- Material Design fonts -->
     <link rel="stylesheet"
   href="http://fonts.googleapis.com/css?family=Roboto:300,400,500
   ,700" type="text/css">
     <link
   href="https://fonts.googleapis.com/icon?family=Material+Icons"
   rel="stylesheet">

     <!-- Bootstrap -->
     <link
   href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.m
   in.css" rel="stylesheet">
```

```
  <!-- Bootstrap Material Design -->
  <link href="dist/css/bootstrap-material-design.css"
rel="stylesheet">
  <link href="dist/css/ripples.min.css" rel="stylesheet">

  <!-- Dropdown.js -->
  <link
href="//cdn.rawgit.com/FezVrasta/dropdown.js/master/jquery.drop
down.css" rel="stylesheet">

  <!-- General Page style -->
  <link href="index.css" rel="stylesheet">
  <link rel="stylesheet" href="slidebars/slidebars.css">
  <!-- Custom Page style-->
  <link href="custom_css/jarvis.css" rel="stylesheet">

  <!-- Font Icon Style   -->
<link href="fonts/css/fontello.css" rel="stylesheet">

<!-- Font Icon Style   -->
<link href="fonts/css/animation.css" rel="stylesheet">

  <!-- jQuery -->
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery
.min.js"></script>

  <meta name="viewport" content="width=device-width,
initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0,
user-scalable=no">
```

```
</head>
<body>


 <nav class="navbar navbar-fixed-top custom-navbar sb-slide"
role="navigation">
   <div class="container bg-nav">
    <button type="button" class="navbar-toggle sb-toggle-left">

    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    </button>
    <a href=""class="navbar-brand "><h2>Switch
Controls</h2></a>
    <!--/.nav-collapse -->
   </div>
   <!--/.container -->
 </nav>
 <div id="sb-site">
     <div class="container ">

          <br>
          <br>
          <br>
          <br>
          <div class="row text-center">
               <div class = "col-xs-4">
                    <div class="togglebutton" >

                      <label>
                    <input type="checkbox" id="light1">
                    </label>
```

```
            </div>
                </div>
                <div class = "col-xs-4">
                      <h3>Light 1 </h3>
                </div>
                <div class = "col-xs-4">
                     <h2><span class="icon-lightbulb"
id="light1_icon"></span></h2>
                </div>
           </div>
           <br>
           <div class="row text-center">
                <div class = "col-xs-4">
                     <div class="togglebutton" >
                    <label>
                    <input type="checkbox" id="fan1">
                    </label>
                </div>
                 </div>

                 <div class = "col-xs-4">
                       <h3>Fan 1 </h3>
                 </div>
                 <div class = "col-xs-4">
                      <h2><span class="icon-fan"
id="fan1_icon"></span></h2>
                 </div>
           </div>

                <br>


     </div>
```

```
 </div>

 <div class="sb-slidebar sb-left sb-style-push">
   <!-- Your left Slidebar content. -->
   <div class="container">
     <nav class="navbar navbar-default" role="navigation">
       <!-- Brand and toggle get grouped for better mobile
display -->
         <div class="navbar-header ">
         <a class="navbar-brand text-center"
href="home.php"><h2>Jarvis</h2></a>
         </div>

         <!-- Collect the nav links, forms, and other content
for toggling -->

         <ul class="nav navbar-nav">
         <li class="active text-center"><a ><h4>Switch
Controls</h4></a></li>
         <li class="text-center"><a
href="temp.php"><h4>Temperature</h4></a></li>
         <li class="text-center"><a
href="security.php"><h4>Security</h4></a></li>

         <?php
               if($_COOKIE['jcook_role']=="admin"){
                    echo '<li class="text-center"><a
href="admin.php"><h4>Admin Console</h4></a></li>';
               }
         ?>
         <li class="text-center"><a
href="logout.php"><h4>Logout</h4></a></li>
         </ul>
```

```
      </nav>
    </div>
  </div>


 <!-- Twitter Bootstrap -->
 <script
 src="//cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.6/j
 s/bootstrap.min.js"></script>

 <!-- Material Design for Bootstrap -->
 <script src="dist/js/material.js"></script>
 <script src="dist/js/ripples.min.js"></script>

 <!--Slide bars-->
 <script src="slidebars/slidebars.js"></script>


 <script>
   $.material.init();
   $.slidebars();
 </script>

<script type="text/javascript">
               $(document).ready(function() {

                    var x = <?php echo $row['isAllowed'];?>;

                    if(!x){

 $("#fan1").attr('disabled','disabled');

 $("#light1").attr('disabled','disabled');
```

```
                        Android.showToast("You have been
blocked by admin");
                    }

                     $.ajax({
                        type: 'POST',

                        url: 'getval.php',

                        data: {id:1},

                        dataType: 'json',
                        success: function(data)
                        {

                             if(data[1])
                             {
                              $('#light1').prop('checked',
true);
                             }else {
                              $('#light1').prop('checked',
false);
                             }
                        }

                        });


                        $("#light1").change(function(){

   if($(this).prop("checked") == true){
                var light = 1;
                $.ajax({
                    type: 'POST',
```

```
            url: 'simon_says.php',                  //the
 script to call to get data
                                                    data:
 {light1:light},                          //you can insert url
 argumnets here to pass to api.php

     //for example "id=5&parent=6"
                                 //data format
                                                    success:
 function(data)          //on recieve of reply
                                                    {


 Android.showToast(data);

 $("#light1_icon").css("text-shadow","2px 2px 3px #ffef1a");
          }

});
     }else  {
                var light = 0;
               $.ajax({
                type: 'POST',
                 url: 'simon_says.php',
          //the script to call to get data
                    data: {light1:light},
     //you can insert url argumnets here to pass to api.php

     //for example "id=5&parent=6"

     //data format
          success: function(data)       //on recieve of reply
     {
```

```
    Android.showToast(data);

    $("#light1_icon").css("text-shadow","");
              }

              });
         }

    });

    $("#fan1").change(function(){

      if($(this).prop("checked") == true){
                   var fan = 1;
                  $.ajax({
                  type: 'POST',
                  url: 'simon_says.php',
//the script to call to get data
                        data: {fan1:fan},
    //you can insert url argumnets here to pass to api.php

        //for example "id=5&parent=6"

        //data format
        success: function(data)        //on recieve of reply
              {

    $("#fan1_icon").addClass("animate-spin");

    Android.showToast(data);
              }
```

```
    });
    }else  {
       var fan = 0;
        $.ajax({
        type: 'POST',
        url: 'simon_says.php',                    //the script to
    call to get data
      data: {fan1:fan},                           //you can insert
    url argumnets here to pass to api.php

        //for example "id=5&parent=6"

        //data format
             success: function(data)       //on recieve of reply
        {



    $("#fan1_icon").removeClass("animate-spin");

    Android.showToast(data);
        }

        });
    }

});
});
            </script>

    <!-- Sliders -->
    <script
    src="//cdnjs.cloudflare.com/ajax/libs/noUiSlider/6.2.0/jquery.n
    ouislider.min.js"></script>
```

```
    <!-- Dropdown.js -->
    <script
    src="https://cdn.rawgit.com/FezVrasta/dropdown.js/master/jquery
    .dropdown.js"></script>
    <script>
      $("#dropdown-menu select").dropdown();
    </script>
    </body>
</html>
```

Simon_Says.php

```
<?php


    //---------------------------------------------------------------
    -------------
 // Example php script for fetching data from mysql database

    //---------------------------------------------------------------
    -------------
 include_once 'configure_db.php';
 $tableName = "states";
 $light1 = getParam('light1');
 $fan1 = getParam('fan1');
 //inorder to handle both get and post requests

 function getParam($key){
    switch (true) {
    case isset($_GET[$key]):
    return $_GET[$key];
    break;
    case isset($_POST[$key]):
```

```
   return $_POST[$key];
   break;
   default:
   return null;
   break;
   }
 }

 date_default_timezone_set("Asia/Calcutta");
 $con = mysqli_connect($host,$user,$pass,$databaseName);


// Check connection
if (mysqli_connect_errno())
 {
 echo "Failed to connect to MySQL: " . mysqli_connect_error();
 }
$query = "SELECT `state` FROM `states` WHERE `id` = 'fan1'";
$result = mysqli_query($con,$query);
$row = mysqli_fetch_array($result);
$fan1_prev = $row['state'];

$query = "SELECT `state` FROM `states` WHERE `id` = 'light1'";
$result = mysqli_query($con,$query);
$row = mysqli_fetch_array($result);
$light1_prev = $row['state'];

if($light1!=null){
   $query = "UPDATE  `states` SET  `state` =".$light1. " WHERE
   `id`='light1'";
   mysqli_query($con,$query);
   if($light1=="1"&&$light1_prev=="0") {
```

```
$query = "INSERT INTO `timelog`(`object`, `onTime`,`offTime`,
`date`) VALUES
('light1','".date("H:i:s")."','".date("H:i:s")."','".date("Y-m-
d")."') ";
mysqli_query($con,$query);
echo "Lights are now on";
}
elseif($light1=="0"&&$light1_prev=="1") {
$query = "SELECT * FROM `timelog` WHERE `object`= 'light1'
ORDER BY `id` DESC LIMIT 1";
$result = mysqli_query($con,$query);
$row = mysqli_fetch_array($result);
$date = $row['date'];
$date = date_create($date);

$cur_date = date("Y-m-d");
$cur_date = date_create($cur_date);
$diff=date_diff($date,$cur_date);

$diff = $diff->days;

    if($diff>0){
         $query = "UPDATE `timelog` SET `offTime` = '23:59:59'
WHERE `timelog`.`id` = ".$row['id']." ";
         mysqli_query($con,$query);
         if($diff>1){
         while($date!=$cur_date){

date_add($date,date_interval_create_from_date_string("1
days"));
         $diff--;
         if($diff==0)break;
         $query = "INSERT INTO `timelog`(`object`, `onTime`,
`offTime` ,`date`) VALUES
```

```
    ('light1','00:00:00','23:59:59','".date_format($date,"Y-m-d")."
    ') ";
              mysqli_query($con,$query);
              echo date_format($date,"Y-m-d");
              }

              }
              $query = "INSERT INTO `timelog`(`object`, `onTime`,
    `offTime` ,`date`) VALUES
    ('light1','00:00:00','".date("H:i:s")."','".date_format($cur_da
    te,"Y-m-d")."') ";
              mysqli_query($con,$query);
         }else{
              $query = "UPDATE `timelog` SET `offTime` =
    '".date("H:i:s")."' WHERE `timelog`.`id` = ".$row['id']." ";
              mysqli_query($con,$query);
         }

    echo "Lights are now off";
    }
    elseif($light1=="0"&&$light1_prev=="0"){
    echo "Lights are already off";
    }
    elseif ($light1=="1"&&$light1_prev=="1") {
    echo "Lights are already on";
    }
  }



if($fan1!=null){
    $query = "UPDATE  `states` SET  `state` =".$fan1. " WHERE
    `id`='fan1'";
    mysqli_query($con,$query);
    if($fan1=="1"&&$fan1_prev=="0") {
```

```
$query = "INSERT INTO `timelog`(`object`, `onTime`,`offTime`,
`date`) VALUES
('fan1','".date("H:i:s")."','".date("H:i:s")."','".date("Y-m-d"
)."') ";
mysqli_query($con,$query);
echo "Fans are now on";
}
elseif ($fan1=="0"&&$fan1_prev=="1") {

$query = "SELECT * FROM `timelog` WHERE `object`= 'fan1' ORDER
BY `id` DESC LIMIT 1";
$result = mysqli_query($con,$query);
$row = mysqli_fetch_array($result);
$date = $row['date'];
$date = date_create($date);

$cur_date = date("Y-m-d");
$cur_date = date_create($cur_date);
$diff=date_diff($date,$cur_date);

$diff = $diff->days;

    if($diff>0){
    $query = "UPDATE `timelog` SET `offTime` = '23:59:59' WHERE
`timelog`.`id` = ".$row['id']." ";
    mysqli_query($con,$query);
    if($diff>1){
         while($date!=$cur_date){

date_add($date,date_interval_create_from_date_string("1
days"));
         $diff--;
         if($diff==0)break;
```

```
            $query = "INSERT INTO `timelog`(`object`, `onTime`,
    `offTime` ,`date`) VALUES
    ('fan1','00:00:00','23:59:59','".date_format($date,"Y-m-d")."')
    ";
            mysqli_query($con,$query);
            echo date_format($date,"Y-m-d");
        }

        }
        $query = "INSERT INTO `timelog`(`object`, `onTime`,
    `offTime` ,`date`) VALUES
    ('fan1','00:00:00','".date("H:i:s")."','".date_format($cur_date
    ,"Y-m-d")."') ";
        mysqli_query($con,$query);
        }else{
            $query = "UPDATE `timelog` SET `offTime` =
    '".date("H:i:s")."' WHERE `timelog`.`id` = ".$row['id']." ";
            mysqli_query($con,$query);
        }
    echo "Fans are now off";

    }
    elseif($fan1=="0"&&$fan1_prev=="0"){
    echo "Fans are already off";
    }
    elseif ($fan1=="1"&&$fan1_prev=="1") {
    echo "Fans are already on";
    }
}

?>
```

getval.php

```
<?php

   include_once 'configure_db.php';

   $con = mysqli_connect($host,$user,$pass,$databaseName);

   $query = "SELECT state from states ";

   if (mysqli_connect_errno())
     {
     echo "Failed to connect to MySQL: " . mysqli_connect_error();
     }

 $result=mysqli_query($con,$query);
 $row=mysqli_fetch_row($result);
 echo $row[1];
?>
```

temp.php

```
<html>
       <head>
         <title>Jarvis</title>
         <meta charset="utf-8">
         <meta http-equiv="X-UA-Compatible" content="IE=edge">

         <!-- Mobile support -->
         <meta name="viewport" content="width=device-width,
   initial-scale=1">

         <!-- Material Design fonts -->
         <link rel="stylesheet"
   href="http://fonts.googleapis.com/css?family=Roboto:300,400,500
   ,700" type="text/css">
```

```
      <link
href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">

      <!-- Bootstrap -->
      <link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.m
in.css" rel="stylesheet">

      <!-- Bootstrap Material Design -->
      <link href="dist/css/bootstrap-material-design.css"
rel="stylesheet">
      <link href="dist/css/ripples.min.css" rel="stylesheet">

      <!-- Dropdown.js -->
      <link
href="//cdn.rawgit.com/FezVrasta/dropdown.js/master/jquery.drop
down.css" rel="stylesheet">

      <!-- Page style -->
      <link href="index.css" rel="stylesheet">

      <!-- jQuery -->
      <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery
.min.js"></script>

      <!-- Script for Morris js-->
      <link rel="stylesheet"
href="//cdnjs.cloudflare.com/ajax/libs/morris.js/0.5.1/morris.c
ss">
      <script
src="//cdnjs.cloudflare.com/ajax/libs/raphael/2.1.0/raphael-min
.js"></script>
```

```
      <script
src="//cdnjs.cloudflare.com/ajax/libs/morris.js/0.5.1/morris.mi
n.js"></script>

    </head>
    <body>


         <div class="container">
              <br>
                   <div class="row" >
                        <h3 id="sort_title"
class="text-center">Last 7 months</h3>
                   </div>
                   <div class="row">
                        <div class="col-lg-12">
                             <div id="graph_parent">
                                  <div id="morris-line-chart"
style="height: 200px;"></div>
                             </div>
                        </div>
                   </div>
                   <br>
                   <div class="row">
                        <div class="col-lg-12">
                                  <div class="dropdown">
                                      <button class="btn"
id="menu1" type="button" data-toggle="dropdown">Sort By
                                      <span
class="caret"></span></button>
                                      <ul class="dropdown-menu"
role="menu" aria-labelledby="menu1">
                                              <li><a href="#"
id="action-day">Last 12 hours</a></li>
```

```
                                        <li><a href="#"
id="action-week">Last 7 days</a></li>
                                    <li><a href="#"
id="action-month">Last 12 months</a></li>
                                </ul>
                            </div>

                        </div>
                    </div>

                    <div class="row" >
                        <div class="col-lg-12">
                            <h3
class="text-center">Temperature : <strong><span
id="cur_temp"></span></strong>°C</h3>
                            <p class="text-center"><small>As
recorded on <cite><span id="cur_date"></span></cite> at
<cite><span id="cur_time"></span></cite></small></p>
                        </div>
                    </div>



        <script type="text/javascript">

            $(document).ready(function() {
                        var months = ["Jan", "Feb", "Mar",
"Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"];

                        //Initially on load set's the graph
sorting on month basis
                        $.ajax({
                          type: 'POST',
```

```
                        url: 'get_temp.php',
                        data: {sort_by:"month"},
                        dataType: 'json',
                        success: function(Json_data)
                        {
                           Morris.Line({
                                // ID of the element in which
to draw the chart.
                                element: 'morris-line-chart',

                                // Chart data records -- each
entry in this array corresponds to a point
                                // on the chart.
                                data: Json_data,
                                // The name of the data record
attribute that contains x-values.
                                  xkey: 'date',
                                  xLabels:'month',
                                  xLabelFormat: function(x) {
// <--- x.getMonth() returns valid index
                                      var day =
months[x.getMonth()];
                                      return day;
                                    },

                                  // A list of names of data
record attributes that contain y-values.
                                  ykeys: ['temperature'],
                                  // Labels for the ykeys --
will be displayed when you hover over the
                                  // chart.
                                  labels: ['temperature'],
                                  ymin: 'auto',
                                  ymax: 'auto',
```

```
                    postUnits : '°C',
                    autosize:true
                });

            }

        });

        $.ajax({
            type: 'POST',
            url: 'get_temp.php',
            data: {sort_by:"last"},
            dataType: 'json',
            success: function(Json_data)
            {
                var temp=Json_data[0];
                var dt=Json_data[1];
                var t=Json_data[2];

                $('#cur_temp').html(temp);
                $('#cur_date').html(dt);
                $('#cur_time').html(t);

            }

        });


    });
</script>

<script type="text/javascript">
//for week
    $("#action-week").click(function(e)
```

```
        {
              //removing the existing div
              var child =
document.getElementById("morris-line-chart");
              var parent =
document.getElementById("graph_parent");
              parent.removeChild(child);

              //creating the div again
              var div = document.createElement('div');
              div.id = 'morris-line-chart';
              parent.appendChild(div);
              $("#morris-line-chart").css("height","200px");
              $('#sort_title').html("Last 7 Days");

              $.ajax({
                type: 'POST',
                url: 'get_temp.php',
                data: {sort_by:"week"},
                dataType: 'json',
                success: function(Json_data)
                {       var days = ["Sun",
"Mon","Tue","Wed","Thu","Fri","Sat"];
                    Morris.Line({
                        // ID of the element in which to draw
the chart.
                        element: 'morris-line-chart',

                        // Chart data records -- each entry in
this array corresponds to a point
                        // on the chart.
                        data: Json_data,
                        // The name of the data record attribute
that contains x-values.
```

```
                        xkey: 'date',
                        xLabels:'day',
                        xLabelFormat: function(x) { // <---
x.getMonth() returns valid index
                            var day = days[x.getDay()]

                            return day;
                          },

                        // A list of names of data record
attributes that contain y-values.
                        ykeys: ['temperature'],
                        // Labels for the ykeys -- will be
displayed when you hover over the
                        // chart.
                        labels: ['temperature'],
                        ymin: 'auto',
                        ymax: 'auto',
                        postUnits : '°C',
                        autosize:true
                    });

                  }

              });
              e.preventDefault();
              });
         //for month
              $("#action-month").click(function(e)
         {
              //removing the existing div
              var child =
document.getElementById("morris-line-chart");
              var parent =
```

```
document.getElementById("graph_parent");
              parent.removeChild(child);

              //creating the div again
              var div = document.createElement('div');
              div.id = 'morris-line-chart';
              parent.appendChild(div);
              $("#morris-line-chart").css("height","200px");
              $('#sort_title').html("Last 7 Months");

              var months = ["Jan", "Feb", "Mar", "Apr", "May",
"Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"];

              $.ajax({
                type: 'POST',
                url: 'get_temp.php',
                data: {sort_by:"month"},
                dataType: 'json',
                success: function(Json_data)
                {
                   Morris.Line({
                        // ID of the element in which to draw
the chart.
                        element: 'morris-line-chart',

                        // Chart data records -- each entry in
this array corresponds to a point
                        // on the chart.
                        data: Json_data,
                        // The name of the data record attribute
that contains x-values.
                          xkey: 'date',
                          xLabels:'month',
                          xLabelFormat: function(x) { // <---
```

```
x.getMonth() returns valid index
                              var day = months[x.getMonth()];
                              return day;
                            },

                           // A list of names of data record
attributes that contain y-values.
                           ykeys: ['temperature'],
                           // Labels for the ykeys -- will be
displayed when you hover over the
                           // chart.
                           labels: ['temperature'],
                           ymin: 'auto',
                           ymax: 'auto',
                           postUnits : '°C',
                           autosize:true
                       });

                     }

                );

                e.preventDefault();
                });
          //for day
                $("#action-day").click(function(e)
          {
                //removing the existing div
                var child =
document.getElementById("morris-line-chart");
                var parent =
document.getElementById("graph_parent");
                parent.removeChild(child);
```

```
            //creating the div again
            var div = document.createElement('div');
            div.id = 'morris-line-chart';
            parent.appendChild(div);
            $("#morris-line-chart").css("height","200px");
            $('#sort_title').html("Last 12 hours");



            $.ajax({
              type: 'POST',
              url: 'get_temp.php',
              data: {sort_by:"day"},
              dataType: 'json',
              success: function(Json_data)
              {
                 Morris.Line({
                      // ID of the element in which to draw
the chart.
                      element: 'morris-line-chart',

                      // Chart data records -- each entry in
this array corresponds to a point
                      // on the chart.
                      data: Json_data,
                      // The name of the data record attribute
that contains x-values.
                        xkey: 'time',
                        xLabels:'hour',


                        // A list of names of data record
attributes that contain y-values.
```

```
                        ykeys: ['temperature'],
                        // Labels for the ykeys -- will be
displayed when you hover over the
                        // chart.
                        labels: ['temperature'],
                        ymin: 'auto',
                        ymax: 'auto',
                        postUnits : '°C',
                        hideHover:true
                    });

                  }

                });


                e.preventDefault();
                });
            </script>
            </div>


    <!-- Twitter Bootstrap -->
    <script
src="//cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.6/j
s/bootstrap.min.js"></script>

    <!-- Material Design for Bootstrap -->
    <script src="dist/js/material.js"></script>
    <script src="dist/js/ripples.min.js"></script>
    <script>
      $.material.init();
    </script>
```

```
        <!-- Sliders -->
        <script
    src="//cdnjs.cloudflare.com/ajax/libs/noUiSlider/6.2.0/jquery.n
    ouislider.min.js"></script>

        <!-- Dropdown.js -->
        <script
    src="https://cdn.rawgit.com/FezVrasta/dropdown.js/master/jquery
    .dropdown.js"></script>
        <script>
          $("#dropdown-menu select").dropdown();

        </script>
        </body>
    </html>
```

get_temp.php

```
<?php


//-------------------------------------------------------------------
----------
  // Example php script for fetching data from mysql database

//-------------------------------------------------------------------
----------
  $host = "localhost";
  $user = "root";
  $pass = "root";

  $databaseName = "arduino";
```

```
//-----------------------------------------------------------------
----------
  // 1) Connect to mysql database

//-----------------------------------------------------------------
----------
  $con = mysqli_connect($host,$user,$pass,$databaseName);



// Check connection
  if (mysqli_connect_errno())
          {
            echo "Failed to connect to MySQL: " .
mysqli_connect_error();
          }


  $query_month = "SELECT ROUND(AVG(temperature),0) as temperature
, DATE_FORMAT(`enDate`, '%Y-%m') as date From temp GROUP BY
DATE_FORMAT(`enDate`, '%Y-%m')";
  $query_week= "SELECT ROUND(AVG(temperature),1) as temperature ,
enDate as date From temp GROUP BY `enDate` Order by `enDate`DESC
LIMIT 7";
  $query_day="SELECT  `temperature`, CONCAT_WS('
',`enDate`,`enTime`) as time FROM `temp` WHERE `enDate` =
'2016-02-07'";
  $query_last="SELECT `temperature`, DATE_FORMAT(`enDate`,'%d %b
%Y'), TIME_FORMAT(enTime, '%h:%i %p') FROM `temp` ORDER BY
`temp`.`enDate` DESC, `temp`.`enTime` DESC LIMIT 1";
```

```
  $sort = $_POST["sort_by"];

          if($sort=='last'){
            $query = $query_last;
            $result = mysqli_query($con,$query);          //query
            $array = mysqli_fetch_row($result);
//fetch result

            //---------------echo json_encode($array);echo
json_encode($array);echo json_encode($array);echo
json_encode($array);echo json_encode($array);echo
json_encode($array);echo json_encode($array);echo
json_encode($array);echo json_encode($array);echo
json_encode($array);echo json_encode($array);echo
json_encode($array);echo
json_encode($array);-----------------------------------------------
-------------
            // 3) echo result as json

//-------------------------------------------------------------------
----------
            echo json_encode($array);
          } else{

             if($sort=='month'){
               $query = $query_month;
             }
             elseif ($sort=='week') {
               $query = $query_week;
             }
             elseif($sort=='day'){
               $query = $query_day;
```

```
            }

           $result = mysqli_query($con,$query);
           $total_rows =  $result->num_rows;
           if($result)
           {
               $rows = mysqli_fetch_all($result, MYSQLI_ASSOC);
           }

            echo json_encode($rows);
       }



?>
```

Pushyapi.php

```
<?php

class PushyAPI
{
   static public function sendPushNotification( $data, $ids )
   {
   // Your Pushy API key
   $apiKey =
   'fd954fa796cd9e209d55a9821fb6b74ac1162bacf2327c381257042308fb75
   8a';

   // Define URL to Pushy endpoint
   $url = 'https://pushy.me/push?api_key=' . $apiKey;

   // Set post variables
   $post = array
```

```
(
        'registration_ids'  => $ids,
        'data'              => $data,
);

// Set Content-Type since we're sending JSON
$headers = array
(
        'Content-Type: application/json'
);

// Initialize curl handle
$ch = curl_init();

// Set URL to Pushy endpoint
curl_setopt( $ch, CURLOPT_URL, $url );

// Set request method to POST
curl_setopt( $ch, CURLOPT_POST, true );

// Set our custom headers
curl_setopt( $ch, CURLOPT_HTTPHEADER, $headers );

// Get the response back as string instead of printing it
curl_setopt( $ch, CURLOPT_RETURNTRANSFER, true );

// Set post data as JSON
curl_setopt( $ch, CURLOPT_POSTFIELDS, json_encode( $post ) );

// Actually send the push
$result = curl_exec( $ch );

// Display errors
```

```
    if ( curl_errno( $ch ) )
    {
            echo curl_error( $ch );
    }

    // Close curl handle
    curl_close( $ch );

    // Debug API response
    //echo $result;
    }
}

?>
```

Pushy_register.php

```
<?php
 include_once 'configure_db.php';
 $con = mysqli_connect($host,$user,$pass,$databaseName);

 if (mysqli_connect_errno())
 {
 echo "Failed to connect to MySQL: " . mysqli_connect_error();
 }
 if(!null){
     $query = "INSERT INTO `device_registration`(`reg_id`) VALUES
    ('".$regId."')";
    $result=mysqli_query($con,$query);
    if(!result){
    $response = array('status'=>0,'message'=>'Unable to register');
    }
    else {
```

```
   $response = array('status'=>1,'message'=>'Succesfully
   registered');
   }
   print json_encode($response);
 }

?>
```

send_pushy.php

```
<?php
require('pushyapi.php');
include_once 'configure_db.php';
 $tableName = "device_registration";
 $message = "Breach in security";
 $con = mysqli_connect($host,$user,$pass,$databaseName);

 if (mysqli_connect_errno())
 {
 echo "Failed to connect to MySQL: " . mysqli_connect_error();
 }
 $result = mysqli_query($con,"select reg_id from
   device_registration");

 if($result)
 {

   while ($row = mysqli_fetch_array($result)) {
   $ids[] = $row['reg_id'];
    $device_count++;
     }

 }
```

```
 $data = array( 'message' => $message );
 PushyAPI::sendPushNotification( $data, $ids);

 echo "Done";
?>
```

login.php

```
<?php
$login_status=true;
if(isset($_COOKIE["jcook_name"])) header('Location: home.php');
else if(isSet($_POST['submit']))
{

include_once 'do_login.php';
}

?>

<html>
   <head>
     <title>Jarvis</title>
     <meta charset="utf-8">
     <meta http-equiv="X-UA-Compatible" content="IE=edge">

     <!-- Mobile support -->
     <meta name="viewport" content="width=device-width,
   initial-scale=1">

     <!-- Material Design fonts -->
```

```
  <link rel="stylesheet"
href="http://fonts.googleapis.com/css?family=Roboto:300,400,500
,700" type="text/css">
  <link
href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">

  <!-- Bootstrap -->
  <link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.m
in.css" rel="stylesheet">


  <!-- Bootstrap Material Design -->
  <link href="dist/css/bootstrap-material-design.css"
rel="stylesheet">
  <link href="dist/css/ripples.min.css" rel="stylesheet">

  <!-- Dropdown.js -->
  <link
href="//cdn.rawgit.com/FezVrasta/dropdown.js/master/jquery.drop
down.css" rel="stylesheet">

  <!-- General Page style -->
  <link href="index.css" rel="stylesheet">
  <link rel="stylesheet" href="slidebars/slidebars.css">
  <!-- Custom Page style-->
  <link href="custom_css/jarvis.css" rel="stylesheet">

  <!-- Font Icon Style   -->
<link href="fonts/css/fontello.css" rel="stylesheet">

<!-- Font Icon Style   -->
```

```
<link href="fonts/css/animation.css" rel="stylesheet">

  <!-- jQuery -->
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery
.min.js"></script>

  <meta name="viewport" content="width=device-width,
initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0,
user-scalable=no">

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/fo
nt-awesome.min.css">
  <script type="text/javascript">
     $(document).ready(function(){
          var x = <?php echo $login_status; ?>;

          if(x==0){
               Android.showToast("Wrong Username or Password");
          }
     });
  </script>

</head>
<body>


 <nav class="navbar navbar-fixed-top custom-navbar sb-slide
border" role="navigation">
   <div class="container bg-nav">
    <button type="button" class="navbar-toggle sb-toggle-left">
```

```
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    </button>
    <a href=""class="navbar-brand"><h2>Login</h2></a>
    <!--/.nav-collapse -->
   </div>
   <!--/.container -->
 </nav>
 <div id="sb-site" class="bg-body">
     <div class="container">

          <br>
          <br>
          <br>

          <form class="form-horizontal" role="form"
method="post" action="login.php">
            <div class="form-group">
              <label class="control-label col-xs-2"
for="uname">Username:</label>
              <div class="col-xs-10">
              <input type="text" class="form-control"
id="uname" placeholder="Enter name" name="username">
              </div>
            </div>
            <div class="form-group">
              <label class="control-label col-xs-2"
for="pwd">Password:</label>
              <div class="col-xs-10">
              <input type="password" class="form-control"
id="pwd" placeholder="Enter password"name="password">

              </div>
```

```
            </div>



            <div class="form-group">
              <div class="col-xs-offset-2 col-xs-10">
              <input id="submit" type="submit" name="submit"
value="Login" class="btn btn-primary">

              </div>
            </div>

     </div>

 </div>

 <div class="sb-slidebar sb-left sb-style-push">
   <!-- Your left Slidebar content. -->
   <div class="container">
     <nav class="navbar navbar-default" role="navigation">
       <!-- Brand and toggle get grouped for better mobile
display -->
         <div class="navbar-header ">
         <a class="navbar-brand text-center"
href="#"><h2>Automate Me</h2></a>
         </div>

         <!-- Collect the nav links, forms, and other content
for toggling -->

         <ul class="nav navbar-nav">
         <li class="text-center"><a ><h4>Home</h4></a></li>
```

```
          <li class="text-center"><a ><h4>Switch
Controls</h4></a></li>
          <li class="text-center"><a
><h4>Temperature</h4></a></li>
          <li class="text-center"><a ><h4>Security</h4></a></li>
          <li class="text-center"><a ><h4>Logout</h4></a></li>

          </ul>

      </nav>
   </div>
 </div>


<!-- Twitter Bootstrap -->
<script
src="//cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.6/j
s/bootstrap.min.js"></script>

<!-- Material Design for Bootstrap -->
<script src="dist/js/material.js"></script>
<script src="dist/js/ripples.min.js"></script>

<!--Slide bars-->
<script src="slidebars/slidebars.js"></script>


<script>
  $.material.init();
  $.slidebars();
</script>
```

```
    <!-- Sliders -->
    <script
    src="//cdnjs.cloudflare.com/ajax/libs/noUiSlider/6.2.0/jquery.n
    ouislider.min.js"></script>

    <!-- Dropdown.js -->
    <script
    src="https://cdn.rawgit.com/FezVrasta/dropdown.js/master/jquery
    .dropdown.js"></script>
    <script>
      $("#dropdown-menu select").dropdown();
    </script>
    </body>
</html>
```

do_login.php

```
<?php
    include_once 'configure_db.php';
    $con = mysqli_connect($host,$user,$pass,$databaseName);
    if (mysqli_connect_errno())
          {
          echo "Failed to connect to MySQL: " .
    mysqli_connect_error();
          }

    // declare post fields

    $post_username = trim($_POST['username']);
    $post_password = trim($_POST['password']);

    $query = "Select * FROM user_account WHERE
    username='".$post_username."' AND
    password='".$post_password."'";
```

```
$result=mysqli_query($con,$query);
$row =  mysqli_fetch_array($result);

if(!empty($row['username'])&&!empty($row['password']))
{

    setcookie("jcook_name", $row['name'], time() + (86400 *
30), "/");
    setcookie("jcook_uname", $row['username'], time() + (86400
* 30), "/");
    setcookie("jcook_role", $row['role'], time() + (86400 *
30), "/");
    header("Location: home.php");
    exit;
}
else
{
    $login_status = 0;
}
?>
```

logout.php

```
<?php
    setcookie("jcook_name", "", time() - (86400 * 30), "/");
setcookie("jcook_role", "", time() - (86400 * 30), "/");

?>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery
```

```
    .min.js"></script>
    <script type="text/javascript">
        $(document).ready(function(){

                        Android.showToast("Successfully Logged
    out");
                        window.location.href="login.php";

             });
    </script>
    </head>
```

adduser.php

```
<?php
 include_once 'configure_db.php';
$con = mysqli_connect($host,$user,$pass,$databaseName);
if (mysqli_connect_errno())
{
      echo "Failed to connect to MySQL: " .
mysqli_connect_error();
}

 $name=$_POST['name'];
 $uname=$_POST['uname'];
 $pass=$_POST['pass'];
 $query = "INSERT INTO `user_account` ( `name`, `username`,
`password`,`role`) VALUES
('".$name."','".$uname."','".$pass."','user')";
```

```
 if(mysqli_query($con,$query)){
  header("Location:admin.php");
 }else{
  echo "Error: " . $sql . "<br>" . mysqli_error($con);
 }

?>
```

admin.php

```
<?php
 if(isset($_COOKIE["jcook_name"])){}
    else header('Location: login.php');
  include_once 'configure_db.php';
  $con = mysqli_connect($host,$user,$pass,$databaseName);
  if (mysqli_connect_errno())
        {
        echo "Failed to connect to MySQL: " .
mysqli_connect_error();
        }
?>
<html>
  <head>
    <title>Jarvis</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <!-- Mobile support -->
    <meta name="viewport" content="width=device-width,
initial-scale=1">

    <!-- Material Design fonts -->
    <link rel="stylesheet"
```

```
href="http://fonts.googleapis.com/css?family=Roboto:300,400,500,70
0" type="text/css">
    <link
href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">

    <!-- Bootstrap -->
    <link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.
css" rel="stylesheet">


    <!-- Bootstrap Material Design -->
    <link href="dist/css/bootstrap-material-design.css"
rel="stylesheet">
    <link href="dist/css/ripples.min.css" rel="stylesheet">

    <!-- Dropdown.js -->
    <link
href="//cdn.rawgit.com/FezVrasta/dropdown.js/master/jquery.dropdow
n.css" rel="stylesheet">

    <!-- General Page style -->
    <link href="index.css" rel="stylesheet">
    <link rel="stylesheet" href="slidebars/slidebars.css">
    <!-- Custom Page style-->
    <link href="custom_css/jarvis.css" rel="stylesheet">

    <!-- Font Icon Style   -->
     <link href="fonts/css/fontello.css" rel="stylesheet">

      <!-- Font Icon Style   -->
     <link href="fonts/css/animation.css" rel="stylesheet">
```

```
    <!-- jQuery -->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.mi
n.js"></script>

    <meta name="viewport" content="width=device-width,
initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0,
user-scalable=no">

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-
awesome.min.css">
  </head>
  <body>


       <nav class="navbar navbar-fixed-top custom-navbar sb-slide
border" role="navigation">
         <div class="container bg-nav">
           <button type="button" class="navbar-toggle
sb-toggle-left">

             <span class="icon-bar"></span>
             <span class="icon-bar"></span>
             <span class="icon-bar"></span>
           </button>
           <a href=""class="navbar-brand"><h2>Admin
Console</h2></a>
           <!--/.nav-collapse -->
         </div>
         <!--/.container -->
       </nav>
       <div id="sb-site" >
            <br>
```

```
            <br>
            <br>
            <br>
            <div class="container">
                 <div class="row">
                      <div class="col-lg-12 text-center">
                           <h3 >Existing Users</h3>

                      </div>
                   </div>
                   <br>
                 <div class="well well-lg">

                   <div class="row">
                      <div class="col-lg-12 ">


                           <div class="list-group">
                           <div
class="list-group-separator"></div>
                             <?php
                             $query="SELECT * FROM `user_account`
WHERE name NOT IN ('".$_COOKIE['jcook_name']."')";
                             $result=mysqli_query($con,$query);

while($row=mysqli_fetch_array($result)){
                             echo '<div class="list-group-item">';
                             echo '<div class="row-content">';
                             if($row["isAllowed"]){
                                echo '<div
class="action-secondary"><span class="label
label-primary">Active</span>
                                     </div>';
                             }else{
```

```
                                    echo '<div
class="action-secondary"><span class="label
label-warning">Blocked</span>
                                    </div>';
                                }
                                echo '<h4
class="list-group-item-heading">'.$row["name"].'</h4>';
                                echo '<p
class="list-group-item-text">'.$row["username"].'</p>';
                                echo '<p
class="list-group-item-text"> pass = '.$row["password"].'</p>';
                                echo '</div>';
                                echo '</div>';
                                echo '<div
class="list-group-separator"></div>';
                            }
                              ?>


                            </div>

                        </div>
                    </div>
                </div>
                <br>
                <div class="row">
                    <div class="col-lg-12 text-center">
                        <h3 >Set Permission</h3>

                    </div>
                </div>
                  <br>
                 <div class="well well-lg">
                    <div class="row ">
```

```
                        <div class="col-xs-6 text-center">
                            <h4>Block</h4>
                            <br>
                            <?php

                                $query="SELECT
`name`,`username` FROM `user_account` WHERE name NOT IN
('".$_COOKIE['jcook_name']."') AND `isAllowed`=1";

$result=mysqli_query($con,$query);

                            ?>
                            <div class="dropdown">
                                <button class="btn
btn-warning" id="blck_menu" type="button"
data-toggle="dropdown">Select
                                <span
class="caret"></span></button>
                                <ul class="dropdown-menu
blck_menu" role="menu" aria-labelledby="blck_menu">

                                        <?php

while($row=mysqli_fetch_array($result)){
                                                echo '<li
class="dropdown-header"><h5>'.$row["name"].'</h5></li>';
                                                echo
'<li><a>'.$row["username"].'</a></li>';
                                                echo '<li
class="divider"></li>';
                                            }
                                        ?>
                                </ul>
                            </div>
```

```
                                    <form
action="setUser_permission.php" method="post">
                                        <div class="form-group
text-left">
                                          <label
class="control-label" for="disabledInput">Selected User</label>
                                          <input class="form-control
blck_in" id="disabledInput"  type="text" name="uname"
placeholder="No User Selected" readonly>
                                        </div>
                                        <input type="hidden"
name="type" value="blck" />
                                        <button class="btn
btn-default">Go</button>
                                    </form>
                                </div>
                                <div class="col-xs-6 text-center">
                                    <h4>unBlock</h4>
                                                <br>
                                                <?php

                                                    $query="SELECT
`name`,`username` FROM `user_account` WHERE name NOT IN
('".$_COOKIE['jcook_name']."') AND `isAllowed`=0    ";

$result=mysqli_query($con,$query);

                                                <?>
                                                <div
class="dropdown">
                                                    <button
class="btn btn-primary" id="ublck_menu" type="button"
data-toggle="dropdown">Select
```

```
                                                    <span
class="caret"></span></button>
                                                    <ul
class="dropdown-menu ublck_menu" role="menu"
aria-labelledby="ublck_menu">

                                                            <?php

while($row=mysqli_fetch_array($result)){

  echo '<li
class="dropdown-header"><h5>'.$row["name"].'</h5></li>';

  echo '<li><a>'.$row["username"].'</a></li>';

  echo '<li class="divider"></li>';
                                                                }
                                                            ?>
                                                    </ul>
                                                </div>
                                                <form
action="setUser_permission.php" method="post">
                                                    <div
class="form-group text-left">
                                                        <label
class="control-label" for="disabledInput">Selected User</label>
                                                        <input
class="form-control ublck_in" id="disabledInput"
name="uname"type="text" placeholder="No User Selected"
readonly="">
                                                    </div>
                                                    <input
type="hidden" name="type" value="ublck" />
                                                    <button
```

```
class="btn btn-default">Go</button>
                                                    </form>
                          </div>
                     </div>
                 </div>
                 <br>
                <div class="row">
                     <div class="col-lg-12 text-center">
                          <h3 >Add New Users</h3>

                     </div>
                  </div>
                  <br>
                <div class="well well-lg">
                     <div class="row">
                     <div class="col-lg-12">
                          <form class="form-horizontal"
id="sub_form" action="adduser.php" method="post">
                               <div class="form-group">
                                    <label for="inputname"
class="col-md-2 control-label">Name</label>

                                    <div class="col-md-10">
                                      <input type="text"
name="name"class="form-control" id="inputname"
placeholder="ex:-John">
                                    </div>
                               </div>
                               <div class="form-group">
                                    <label for="inputun"
class="col-md-2 control-label">Username</label>

                                    <div class="col-md-10">
                                      <input type="text"
```

```
name="uname"class="form-control" id="inputun"
placeholder="ex:-johndae">
                                        </div>
                                    </div>
                                    <div class="form-group">
                                        <label for="inputps"
class="col-md-2 control-label">Password</label>

                                        <div class="col-md-10">
                                          <input type="text"
name="pass"class="form-control" id="inputps"
placeholder="ex:-strongpass">
                                        </div>
                                    </div>
                                    <br>
                                    <button type="submit" class="btn
btn-primary" id="sub_btn">Submit</button>
                                </form>
                            </div>
                        </div>
                    </div>
                    <br>

            </div>
        </div>

        <div class="sb-slidebar sb-left sb-style-push">
          <!-- Your left Slidebar content. -->
          <div class="container">
             <nav class="navbar navbar-default" role="navigation">
               <!-- Brand and toggle get grouped for better mobile
display -->
                <div class="navbar-header ">
                 <a class="navbar-brand text-center"
```

```
href="#"><h2>Jarvis</h2></a>
                </div>

                <!-- Collect the nav links, forms, and other
content for toggling -->

                 <ul class="nav navbar-nav">
                   <li class="text-center"><a href="#"><h4>Switch
Controls</h4></a></li>
                   <li class="text-center"><a
href="temp.html"><h4>Temperature</h4></a></li>
                   <li class="text-center"><a
href="security.html"><h4>Security</h4></a></li>
                   <li class="active  text-center"><a ><h4>Admin
Console</h4></a></li>
                   <li class="text-center"><a
href="logout.php"><h4>Logout</h4></a></li>

                 </ul>

              </nav>
           </div>
        </div>


  <!-- Twitter Bootstrap -->
  <script
src="//cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.6/js/b
ootstrap.min.js"></script>

  <!-- Material Design for Bootstrap -->
  <script src="dist/js/material.js"></script>
  <script src="dist/js/ripples.min.js"></script>
```

```
  <!--Slide bars-->
  <script src="slidebars/slidebars.js"></script>


  <script>
    $.material.init();
    $.slidebars();
  </script>
  <script type="text/javascript">
                      $(document).ready(function() {
                           $('.blck_menu li >
a').click(function(){
                                $('.blck_in').val($(this).text());
                           });
                           $('.ublck_menu li >
a').click(function(){

$('.ublck_in').val($(this).text());
                           });
                      });
                </script>

  <!-- Sliders -->
  <script
src="//cdnjs.cloudflare.com/ajax/libs/noUiSlider/6.2.0/jquery.noui
slider.min.js"></script>

  <!-- Dropdown.js -->
  <script
src="https://cdn.rawgit.com/FezVrasta/dropdown.js/master/jquery.dr
opdown.js"></script>
  <script>
    $("#dropdown-menu select").dropdown();
  </script>
```

```
  </body>
</html>
```

setuser_permission.php

```
<?php
include_once 'configure_db.php';
$con = mysqli_connect($host,$user,$pass,$databaseName);
if (mysqli_connect_errno())
      {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }
$username = $_POST['uname'];
$type = $_POST['type'];
$isAllowed = null;
if($type=="blck"){
$isAllowed = 0;
}
else if ($type=="ublck"){
$isAllowed = 1;
}
 $query = "UPDATE `user_account` SET `isAllowed` =
'".$isAllowed."' WHERE `user_account`.`username`
='".$username."'";
 if(mysqli_query($con,$query)){
  header("Location:admin.php");
 }else{
  echo "Error: " . $sql . "<br>" . mysqli_error($con);
 }
?>
```

reports.php

```
<?php
if(isset($_COOKIE["jcook_name"])){
   include_once 'configure_db.php';
}
   else header('Location: login.php');

?>
<html>
   <head>
     <title>Jarvis</title>
     <meta charset="utf-8">
     <meta http-equiv="X-UA-Compatible" content="IE=edge">

     <!-- Mobile support -->
     <meta name="viewport" content="width=device-width,
initial-scale=1">

     <!-- Material Design fonts -->
     <link rel="stylesheet"
href="http://fonts.googleapis.com/css?family=Roboto:300,400,500,70
0" type="text/css">
     <link
href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">

     <!-- Bootstrap -->
     <link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.
css" rel="stylesheet">
```

```
    <!-- Bootstrap Material Design -->
    <link href="dist/css/bootstrap-material-design.css"
rel="stylesheet">
    <link href="dist/css/ripples.min.css" rel="stylesheet">

    <!-- Dropdown.js -->
    <link
href="//cdn.rawgit.com/FezVrasta/dropdown.js/master/jquery.dropdow
n.css" rel="stylesheet">

    <!-- General Page style -->
    <link href="index.css" rel="stylesheet">
    <link rel="stylesheet" href="slidebars/slidebars.css">
    <!-- Custom Page style-->
    <link href="custom_css/jarvis.css" rel="stylesheet">

    <!-- Font Icon Style   -->
  <link href="fonts/css/fontello.css" rel="stylesheet">

      <!-- Font Icon Style   -->
  <link href="fonts/css/animation.css" rel="stylesheet">

    <!-- jQuery -->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.mi
n.js"></script>

    <meta name="viewport" content="width=device-width,
initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0,
user-scalable=no">
```

```
     <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-
awesome.min.css">
   </head>
   <body>


   <nav class="navbar navbar-fixed-top custom-navbar sb-slide
border" role="navigation">
     <div class="container bg-nav">
       <button type="button" class="navbar-toggle sb-toggle-left">

       <span class="icon-bar"></span>
       <span class="icon-bar"></span>
       <span class="icon-bar"></span>
       </button>
       <a href=""class="navbar-brand"><h2>Reports</h2></a>
       <!--/.nav-collapse -->
     </div>
     <!--/.container -->
   </nav>
   <div id="sb-site" >
        <div class="container">
             <br><br><br>
             <?php
                  $con =
mysqli_connect($host,$user,$pass,$databaseName);
                  if (mysqli_connect_errno())
                        {
                        echo "Failed to connect to MySQL: " .
mysqli_connect_error();
                        }
                  $query = "SELECT  `object` , `date` , SUM(  `hr`
) AS hr, SUM(  `mn` ) AS mn, SUM(  `sc` ) AS sc
```

```
                                   FROM  `timelog` GROUP BY
`object`";
                  $result=mysqli_query($con,$query);

                  echo '<div class="row">';
                       echo '<div class="col-xs-12">';
                            echo "<h4>Total Usage</h4>";
                              while ($row =
mysqli_fetch_array($result)) {
                                   echo $row['object']." -
".$row['hr']." hrs, ".$row['mn']." mins, ".$row['sc']." secs";
                                   echo "<br>";
                                }

                       echo"</div>";
                  echo"</div>"


             ?>
             <br>
             <div class="row">
                  <div class="col-xs-12">
                       <h4>Today's Usage</h4>

                       <?php
                            $query = "SELECT  `object` , `date` ,
SUM(  `hr` ) AS hr, SUM(  `mn` ) AS mn, SUM(  `sc` ) AS sc FROM
`timelog` WHERE object = 'light1' GROUP BY  `date` ORDER BY
`date` DESC LIMIT 1";
                            $result=mysqli_query($con,$query);
                            $row =  mysqli_fetch_array($result);
                            echo $row['object']." - ".$row['hr']."
hrs, ".$row['mn']." mins, ".$row['sc']." secs";
                                   echo "<br>";
```

```
                                                    $query =
"SELECT  `object` , `date` , SUM(  `hr` ) AS hr, SUM(  `mn` ) AS
mn, SUM(  `sc` ) AS sc FROM  `timelog` WHERE object = 'fan1' GROUP
BY  `date` ORDER BY  `date` DESC LIMIT 1";

$result=mysqli_query($con,$query);
                                                    $row =
mysqli_fetch_array($result);
                                                    echo
$row['object']." - ".$row['hr']." hrs, ".$row['mn']." mins,
".$row['sc']." secs";
                                                        echo
"<br>";
                        ?>
                    </div>
                </div>
            </div>

    </div>

    <div class="sb-slidebar sb-left sb-style-push">
      <!-- Your left Slidebar content. -->
      <div class="container">
         <nav class="navbar navbar-default" role="navigation">
           <!-- Brand and toggle get grouped for better mobile
display -->
             <div class="navbar-header ">
             <a class="navbar-brand text-center"
href="home.php"><h2>Jarvis</h2></a>
             </div>

             <!-- Collect the nav links, forms, and other content
for toggling -->
```

```
            <ul class="nav navbar-nav">
            <li class=" text-center"><a
href="jarvis.php"><h4>Switch Controls</h4></a></li>
            <li class="text-center"><a
href="temp.php"><h4>Temperature</h4></a></li>
            <li class="text-center"><a
href="security.php"><h4>Security</h4></a></li>

            <?php
                  if($_COOKIE['jcook_role']=="admin"){
                        echo '<li class="text-center"><a
href="admin.php"><h4>Admin Console</h4></a></li>';
                  }
            ?>
            <li class="text-center
active"><a><h4>Reports</h4></a></li>
            <li class="text-center"><a
href="logout.php"><h4>Logout</h4></a></li>
            </ul>

         </nav>
      </div>
   </div>


   <!-- Twitter Bootstrap -->
   <script
src="//cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.6/js/b
ootstrap.min.js"></script>

   <!-- Material Design for Bootstrap -->
   <script src="dist/js/material.js"></script>
   <script src="dist/js/ripples.min.js"></script>
```

```
    <!--Slide bars-->
    <script src="slidebars/slidebars.js"></script>


    <script>
      $.material.init();
      $.slidebars();
    </script>
    <script type="text/javascript">
                    $(document).ready(function() {









                    });
                </script>

    <!-- Sliders -->
    <script
src="//cdnjs.cloudflare.com/ajax/libs/noUiSlider/6.2.0/jquery.noui
slider.min.js"></script>

    <!-- Dropdown.js -->
    <script
src="https://cdn.rawgit.com/FezVrasta/dropdown.js/master/jquery.dr
opdown.js"></script>
    <script>
      $("#dropdown-menu select").dropdown();
    </script>
    </body>
</html>
```

CODE FOR ANDROID:

MainActivity.java

```
package com.jarvis.automaton;

import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.KeyEvent;
import android.view.View;
import android.webkit.CookieManager;
import android.webkit.JavascriptInterface;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Switch;
import android.widget.Toast;
```

```
import com.google.gson.JsonElement;
import com.jarvis.automaton.utils.AppConfig;

import org.json.JSONException;
import org.json.JSONObject;
import org.json.JSONTokener;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;
import java.util.Locale;
import java.util.Map;

import ai.api.AIConfiguration;
import ai.api.AIDataService;
import ai.api.AIServiceException;
import ai.api.model.AIRequest;
import ai.api.model.AIResponse;
import ai.api.model.Result;
import me.pushy.sdk.Pushy;
import me.pushy.sdk.exceptions.PushyException;

public class MainActivity extends AppCompatActivity {
    WebView myWebView;
    String RegistrationID;
    Dialog match_text_dialog;
    ListView textlist;
    ArrayList<String> matches_text;
    AIDataService aiDataService;
    TextToSpeech t1;
```

```
private static final int REQUEST_CODE = 1234;
ProgressDialog load;



@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

//Webview compo nent0
myWebView = (WebView) findViewById(R.id.webview);
WebSettings webSettings = myWebView.getSettings();
myWebView.getSettings().setJavaScriptEnabled(true);
//Inject WebAppInterface methods into Web page by having
Interface name 'Android'
myWebView.addJavascriptInterface(new WebAppInterface(this),
"Android");
myWebView.getSettings().setSaveFormData(false);
myWebView.setWebViewClient(new WebViewClient() {
         @Override
         public boolean shouldOverrideUrlLoading(WebView view,
String url) {
         view.loadUrl(url);


         return true;
         }
         @Override
         public void onPageFinished(WebView view,String url) {
         view.clearFormData();
         view.clearHistory();
         }
});
```

```
// Show main page

CookieManager.getInstance().setAcceptCookie(true);
myWebView.loadUrl("http://192.168.0.103/Jarvis/home.php");

//Everything Pushy related
// Restart the socket service, in case the user force-closed
Pushy.listen(this);
RegistrationID = AppConfig.getPref("PushyRegistrationID",
MainActivity.this);

if (RegistrationID == null) // not
// registered
// on
// server
{
         // Register up for push notifications (will return
existing token if
         // already registered before)
         new RegisterForPushNotifications().execute();
}else{

Toast.makeText(getApplicationContext(),RegistrationID,Toast.LEN
GTH_SHORT).show();
}



//Api.ai related
final AIConfiguration config = new
AIConfiguration("b141b0b47aa94439be2433763df9b6a4",
"e5a059c0-a8e0-455b-8f12-f9f7560a226e",
         AIConfiguration.SupportedLanguages.English,
         AIConfiguration.RecognitionEngine.System);
```

```
aiDataService = new AIDataService(this,config);

//Text to speech
t1=new TextToSpeech(getApplicationContext(), new
TextToSpeech.OnInitListener() {
         @Override
         public void onInit(int status) {
         if(status != TextToSpeech.ERROR) {
              t1.setLanguage(Locale.UK);
         }
         }
});
load = new ProgressDialog(MainActivity.this);
load.setMessage("Processing request");
load.setCancelable(false);
}

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
// Check if the key event was the Back button and if there's
history
if ((keyCode == KeyEvent.KEYCODE_BACK) &&
myWebView.canGoBack()) {
         myWebView.goBack();
         return true;
}
// If it wasn't the Back key or there's no web page history,
bubble up to the default
// system behavior (probably exit the activity)
return super.onKeyDown(keyCode, event);
}
```

```
public class WebAppInterface {
Context mContext;

/**
    * Instantiate the interface and set the context
    */
WebAppInterface(Context c) {
        mContext = c;
}

/**
    * Show Toast Message
    *
    * @param toast
    */
@JavascriptInterface
public void showToast(String toast) {
        Toast.makeText(mContext, toast,
Toast.LENGTH_SHORT).show();
}

@JavascriptInterface
public void showSpeech(){

        Intent intent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
             RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        startActivityForResult(intent, 1234);
}

@JavascriptInterface
```

```
public void chat_API(String text){
        AI_processing(text);
}



}


private class RegisterForPushNotifications extends
        AsyncTask<String, Void, String> {
ProgressDialog mLoading;

public RegisterForPushNotifications() {
        // Create progress dialog and set it up
        mLoading = new ProgressDialog(MainActivity.this);
        mLoading.setMessage("Registring Device");
        mLoading.setCancelable(false);

        // Show it
        mLoading.show();
}

@Override
protected String doInBackground(String... params) {
        // Temporary string that will hold the registration
result
        String result;

        try {
        // Get registration ID via Pushy
        result = Pushy.register(MainActivity.this);
        RegistrationID = result;
        } catch (PushyException exc) {
```

```
            // Show error instead
            result = exc.getMessage();
            }
            // Write to log
            Log.d("Pushy", "Registration result: " + result);

            // Return result
            return result;
}

@Override
protected void onPostExecute(String result) {
            // Activity died?
            if (isFinishing()) {
            return;
            }

            // Hide progress bar
            mLoading.dismiss();

            // Display it



Toast.makeText(getApplicationContext(),result,Toast.LENGTH_SHOR
T).show();

            new RegisterIDOnServer().execute();
}
}

private class RegisterIDOnServer extends
            AsyncTask<String, Void, JSONObject> {
```

```
Exception mException = null;

@Override
protected void onPreExecute() {
        super.onPreExecute();
        this.mException = null;
}

public RegisterIDOnServer() {

}

@Override
protected JSONObject doInBackground(String... params) {
        HttpURLConnection urlConnection = null;
        URL url = null;
        JSONObject object = null;
        try {
        String urlString = AppConfig.WEB_SERVICE_URL +
"pushy_register.php?id="+RegistrationID;
        url = new URL(urlString);
        urlConnection = (HttpURLConnection)
url.openConnection();
        urlConnection.setRequestMethod("POST");
        urlConnection.setRequestProperty("Content-Type",
                "application/json");
        urlConnection.setDoOutput(true);
        urlConnection.setDoInput(true);
        urlConnection.connect();
        InputStream inStream = null;
        inStream = urlConnection.getInputStream();
        BufferedReader bReader = new BufferedReader(
                new InputStreamReader(inStream));
```

```
          String temp, response = "";
          while ((temp = bReader.readLine()) != null)
               response += temp;
          bReader.close();
          inStream.close();
          urlConnection.disconnect();
          object = (JSONObject) new
JSONTokener(response).nextValue();
          } catch (Exception e) {
          this.mException = e;
          }

          return (object);
}

@Override
protected void onPostExecute(JSONObject result) {
          try {
          String status = result.getString("status");
          String message = result.getString("message");

          if (status.equals("1")) {
               AppConfig.putPref("PushyRegistrationID",
RegistrationID,
                         MainActivity.this);
          }

          Toast.makeText(MainActivity.this, message,
Toast.LENGTH_LONG)
                    .show();

          } catch (JSONException e) {
          // TODO Auto-generated catch block
```

```
        e.printStackTrace();
        }
}
}


protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
if (requestCode == REQUEST_CODE && resultCode == RESULT_OK) {

        match_text_dialog = new Dialog(MainActivity.this);

match_text_dialog.setContentView(R.layout.dialog_matches_frag);
        match_text_dialog.setTitle("Select Matching Text");
        textlist = (ListView)
match_text_dialog.findViewById(R.id.list);
        matches_text = data

.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
        ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this,
             android.R.layout.simple_list_item_1,
matches_text);
        textlist.setAdapter(adapter);
        textlist.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View
view,
                                int position, long id) {

             match_text_dialog.hide();
             AI_processing(matches_text.get(position));
```

```
        }
        });
        match_text_dialog.show();

}
super.onActivityResult(requestCode, resultCode, data);
}



public void AI_processing(String query){

final AIRequest aiRequest = new AIRequest();
aiRequest.setQuery(query);

new AsyncTask<AIRequest, Void, AIResponse>() {


        @Override
        protected AIResponse doInBackground(AIRequest...
requests) {
        final AIRequest request = requests[0];

        try {
             final AIResponse response =
aiDataService.request(aiRequest);
             return response;
        } catch (AIServiceException e) {
        }
        return null;
        }
        @Override
        protected void onPostExecute(AIResponse response) {
```

```
        if (response != null) {
            // process aiResponse here
            Result result = response.getResult();

            // Get parameters
            String param_name="",param_val="";
            if (result.getParameters() != null &&
!result.getParameters().isEmpty()) {
                for (final Map.Entry<String, JsonElement>
entry : result.getParameters().entrySet()) {
                    param_name += entry.getKey();
                    param_val += entry.getValue();
                }

            }

            char[] c = param_val.toCharArray();
            param_val="";
            for(char x:c){
                if(x!='\"')
                    param_val +=x;
            }
            String reply =
result.getFulfillment().getSpeech();
            switch (param_name){
                case "light":{

                    if(param_val.equals("on")){
                        new
php_switch("simon_says.php","light1","1").execute();}
                    else if(param_val.equals("off")){
```

```
                              new
php_switch("simon_says.php","light1","0").execute();}

                         t1.speak(reply,
TextToSpeech.QUEUE_FLUSH, null);
                         break;

                    }
                    case "fan":{

                         if(param_val.equals("on")){
                              new
php_switch("simon_says.php","fan1","1").execute();}
                         else if(param_val.equals("off")){

                              new
php_switch("simon_says.php","fan1","0").execute();}
                         t1.speak(reply,
TextToSpeech.QUEUE_FLUSH, null);
                         break;
                    }
                    case "temperature":{
                         Log.i("blah", "Inside Switch");
                         new
php_temp("get_currtemp.php").execute();
                         break;
                    }

               }
```

```
        }

        }
}.execute(aiRequest);
}

public class php_switch extends AsyncTask<String ,Void,String>
{
String php_name,param_name,param_value;
public php_switch(String url,String name,String value)
{
        php_name=url;
        param_name=name;
        param_value=value;
}
@Override
protected String doInBackground(String... params) {
        HttpURLConnection urlConnection = null;
        URL url = null;
        try{
        String urlString = AppConfig.WEB_SERVICE_URL +
php_name+"?"+param_name+"="+param_value;
        url = new URL(urlString);
        urlConnection = (HttpURLConnection)
url.openConnection();
        urlConnection.setDoOutput(true);
        urlConnection.setDoInput(true);
        urlConnection.connect();
        InputStream inStream = null;
        inStream = urlConnection.getInputStream();
```

```
        BufferedReader bReader = new BufferedReader(
                  new InputStreamReader(inStream));
        StringBuffer sb = new StringBuffer("");
        String line="";

        while ((line = bReader.readLine()) != null) {
             sb.append(line);
             break;
        }
        bReader.close();
        inStream.close();
        return sb.toString();
        }
        catch(Exception e)
        {
        return null;
        }



}

@Override
protected void onPostExecute(String s) {

Toast.makeText(getApplicationContext(),s,Toast.LENGTH_SHORT).sh
ow();
}
}
public class php_temp extends AsyncTask<String ,Void,String>
{
String php_name;
public php_temp(String url)
{
```

```
        php_name=url;

}
@Override
protected String doInBackground(String... params) {
        HttpURLConnection urlConnection = null;
        URL url = null;
        try{
        String urlString = AppConfig.WEB_SERVICE_URL +
php_name;
        url = new URL(urlString);
        urlConnection = (HttpURLConnection)
url.openConnection();
        urlConnection.setDoOutput(true);
        urlConnection.setDoInput(true);
        urlConnection.connect();
        InputStream inStream = null;
        inStream = urlConnection.getInputStream();
        BufferedReader bReader = new BufferedReader(
                  new InputStreamReader(inStream));
        StringBuffer sb = new StringBuffer("");
        String line="";

        while ((line = bReader.readLine()) != null) {
             sb.append(line);
             break;
        }
        bReader.close();
        inStream.close();
        return sb.toString();
        }
        catch(Exception e)
        {
        return null;
```

```
            }


    }

    @Override
    protected void onPostExecute(String s) {
            String talk = "The current temperature in your house
    is, " + s + " degree celsius";
            float c = Float.parseFloat(s);
            double f = c* Float.parseFloat("1.8") + 32;
            f =  (double) Math.round(f * 100.0) / 100.0;
            String show = "Temp = " + s + " °C / " + f + " °F";

    Toast.makeText(getApplicationContext(),show,Toast.LENGTH_SHORT)
    .show();
            t1.speak(talk, TextToSpeech.QUEUE_FLUSH, null);
    }
    }
}
```

PushyReceiver.java

```
package com.jarvis.automaton;

/**
* Created by jobin on 21/2/16.
*/


    import android.app.Notification;
```

```
    import android.app.NotificationManager;
    import android.app.PendingIntent;
    import android.content.BroadcastReceiver;
    import android.content.Context;
    import android.content.Intent;

public class PushyReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
    String notificationTitle = "Automaton";
    String notificationDesc = "Notification";

    // Attempt to grab the message property from the payload
    if ( intent.getStringExtra("message") != null )
    {
             notificationDesc = intent.getStringExtra("message");
    }

    // Set notification click intent
    PendingIntent notificationIntent =
    PendingIntent.getActivity(context, 0, new Intent(context,
    MainActivity.class), 0);

    Notification.Builder builder = new
    Notification.Builder(context);
    builder.setContentIntent(notificationIntent)
             .setSmallIcon(R.drawable.ic_stat_warrning)
             .setWhen(System.currentTimeMillis())
             .setAutoCancel(true)
             .setContentTitle(notificationTitle)
             .setContentText(notificationDesc);
```

```
    Notification notification1 =  builder.build();

    NotificationManager mNotificationManager =
    (NotificationManager)
    context.getSystemService(Context.NOTIFICATION_SERVICE);
    mNotificationManager.notify(0, notification1);

    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

    tools:showIn="@layout/activity_main"
    tools:context=".MainActivity">

    <WebView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/webview"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true"
    />

</RelativeLayout>
```

dialod_matches_frag.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical" >

   <ListView
   android:id="@+id/list"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   />
</LinearLayout>
```

build.gradle file

```
apply plugin: 'com.android.application'

android {
   compileSdkVersion 23
   buildToolsVersion "23.0.2"

   defaultConfig {
   applicationId "com.jarvis.automaton"
   minSdkVersion 19
   targetSdkVersion 23
   versionCode 1
   versionName "1.0"
```

```
    }
    buildTypes {
    release {
             minifyEnabled false
             proguardFiles
    getDefaultProguardFile('proguard-android.txt'),
    'proguard-rules.pro'
    }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.1.1'
    compile 'com.android.support:design:23.0.1'
    compile 'ai.api:sdk:1.7.6@aar'
    // api.ai SDK dependencies
    compile 'com.google.code.gson:gson:2.3'
    compile 'commons-io:commons-io:2.4'
}
```

Bibliography and References:

http://www.tutorialspoint.com

https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwijwIe5h6LLAhWCUY4KHWHIALYQjRwIBw&url=http%3A%2F%2Fwiznetmuseum.com%2Fportfolio-items%2Fdetecting-pir-motion-from-wizwiki-w7500%2F&psig=AFQjCNFOTIn9Bulp5RrMqXTKo9cfzo3VpA&ust=1457009895175911

www.instructables.com/id/Arduino-Home-automation/

https://www.arduino.cc/
https://api.ai/
https://www.pushy.me/

https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjErsnTiaLLAhUCS44KHSpvBmUQjRwIBw&url=http%3A%2F%2Fforum.arduino.cc%2Findex.php%3Ftopic%3D125908.0&bvm=bv.115339255,d.dGY&psig=AFQjCNH7KfcsN7y2-QFVIrm3OoAWWC9bFQ&ust=1457010559690179

http://wiznetmuseum.com/wp/wp-content/uploads/2014/10/06.Electronics-Ethernet-Shield-1.jpg

https://github.com/FezVrasta/bootstrap-material-design
http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf

www.softwaretestingclass.com

http://creately.com/