

National University of Computer and Emerging

Sciences



Voice Controlled Home Automation

Haroon Ahmad 13-4122

Zain Saleem 13-4338

Umar Iqbal 13-4027

Manaal Ayub 13-4073

Supervisor: Aamir Wali

B.S. Computer Science

Final Year Project: December 2016

Department of Computer Science

FAST NU, Lahore, Pakistan

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the:

**Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Is the sole contribution of the author(s) and no part hereof has been reproduced on **as it is** basis (cut and paste) which can be considered as **Plagiarism.** All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Student 1

Name: **Haroon Ahmad**

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student 2

Name: **Zain Saleem**

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student 3

Name: **Umar Iqbal**

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student 4

Name: **Manaal Ayub**

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Table of Contents

[Abstract 6](#_Toc489141599)

[1. Introduction 7](#_Toc489141600)

[1.1. Goals and Objectives 7](#_Toc489141601)

[1.2. Scope of the Project 7](#_Toc489141602)

[1.3. Functional Requirements 8](#_Toc489141603)

[2. Literature Survey 9](#_Toc489141604)

[2.1. The Framework of Home Remote Automation System Based on Smartphone [1] 9](#_Toc489141605)

[2.2. Voice Control Based Home Automation System [2] 10](#_Toc489141606)

[3. Requirements and design 12](#_Toc489141607)

[3.1. System Overview 12](#_Toc489141608)

[3.2. Design Considerations 12](#_Toc489141609)

[3.3. Goals and Guidelines 13](#_Toc489141610)

[3.4. Development Methods 14](#_Toc489141611)

[3.5. Architectural Strategies 14](#_Toc489141612)

[3.6. System Architecture 16](#_Toc489141613)

[3.6.1 General Framework 16](#_Toc489141614)

[3.6.2 Complete System Architecture 17](#_Toc489141615)

[3.6.3 Component Diagram 17](#_Toc489141616)

[4. Description of Prototype 20](#_Toc489141617)

[5. Implementation 21](#_Toc489141618)

[6. Test Cases 23](#_Toc489141619)

[7. Experimental Results and Analysis 33](#_Toc489141620)

[7.1. Algorithmic Experimentation: 33](#_Toc489141621)

[7.2. Hardware Level Experimentation: 33](#_Toc489141622)

[7.3. Experimentation Results: 33](#_Toc489141623)

[8. Conclusion 35](#_Toc489141624)

[9. References 36](#_Toc489141625)

List of Figures

|  |  |
| --- | --- |
| Fig 2.1: Framework of Home Remote Automation System Based on Smartphone | 9 |
| Fig 2.2: Model of User Input | 10 |
| Fig 2.2.2 Logic Diagram of Thread Algorithm | 11 |
| Fig 3.6.1: General Framework | 16 |
| Fig 3.6.2: Complete System Architecture | 17 |
| Fig 3.6.3: General Component Diagram | 17 |
| Fig 3.6.3.1: Client Component | 18 |
| Fig 3.6.3.2: Hub Component | 18 |
| Fig 3.6.3.3: Cloud Database Component | 19 |
| Fig 4.1: Basic Level Prototype for LED Control | 20 |
| Fig 5.1: Before Input Voice Command | 34 |
| Fig 5.2: After Input Voice Command | 34 |

List of Tables

|  |  |
| --- | --- |
| Fig 5.1:Create Profile | 23 |
| Fig 5.2: Log into Profile | 24 |
| Fig 5.3: Add a Room | 25 |
| Fig 5.4: Add a Device | 26 |
| Fig 5.5: Configure Bluetooth | 27 |
| Fig 5.6: Enter Command via Buttons | 28 |
| Fig 5.7: Enter Command via Voice | 29 |
| Fig 5.8: Receive Visual Feedback | 30 |
| Fig 5.9: Receive Current Feedback | 31 |
| Fig 5.10: Logout of Profile | 32 |

# **Abstract**

Home automation refers to the automation and control of electronic devices in a house. The idea is to reduce human effort to a bare minimum. This encompasses operations from turning on a small garage light from your bedroom to switching off the air conditioner while you are out of the house.

No longer a fiction sequence and with a significant market size, Smart Homes have and continue to attract a great deal of investors and engineers. Quite a few tech-giants have launched home automation systems of their own that, though powerful, are not without their fair share of limitations. These limitations include the control and operation of only certified devices that are compatible and endorsed. So to set up a smart home, a user not just buys the product itself but also has to buy devices that are compatible with it. Tedious installations and configuration settings make it too complex for a user.

Our project, voice controlled home automation, on the contrary, does not come with separate smart devices but instead makes existing devices smart. After an initial hardware setup, it relies on speech recognition to control devices via a relay attached to an Arduino development board. The result of voice controlled home automation would be the implementation of an Android application that allows users to control their homes remotely using voice commands.

# Introduction

The idea of controlling and monitoring your home through a handheld device has been around for some time now. And it certainly isn’t surprising to see a great deal of investment being put into it since the emergence of smartphones. However, some of the keys areas that still need attention are restrictions imposed by the hardware, absence of command feedback and tiresome configuration procedures. In our Final Year Project, we target such limitations by developing an Android application that uses simple hardware (an android based smartphone, arduino board, and wifi module), makes electrical devices simpler to add and control through command feedback by using **speech recognition**.

## Goals and Objectives

The goals and objectives of our project include:

* Development of a **generic framework** that allows the user to make and edit a software replica of their home architectural setup, inclusive of devices to be automated and their details.
* Mapping of **voice commands** to specified devices through speech recognition.
* Creation of two Android applications: **Home Hub and Personal.**
* **Home Hub**: Installed on a smartphone kept within home premises, this application would engage with the cloud and the actual devices to map each input command.
* **Personal**: This user-end application would be installed on each individual member’s smartphone and would be responsible for command recognition and processing.
* Inclusion of a **feedback** component that allows the user to detect change in voltage or current reading before and after input command.

## Scope of the Project

The possible limitations for our project, up till now, are as follows:

* Our project includes control of electrical appliances through Bluetooth when they are found within the standard range allowed for Bluetooth connectivity. Wi-Fi would be used when the actual difference between device and user is beyond this standard range.
* Also, while Bluetooth is prone to hacking, safe-guard implementation is still beyond the scope of our project.
* The scope for our project includes only speech recognition, meaning that the text of the command is recognized but not the voice of the user.
* User will be provided the functionality of assigning names to each device but that have to be valid English words as we are using Google’s Speech Recognition API based on standard American English dictionary.

## Functional Requirements

* Our mobile application will allow a user to :
  + Set up his profile.
  + Create a basic layout that acts as a blueprint to depict the number of rooms and devices installed.
  + Add electrical appliances within each room as specified in the blueprint layout.
  + Input commands to switch devices *On* or *Off*, through speech.
* Our Hub application will allow a user to :
  + Remotely control appliances by sending a notification to the always-ON phone or tablet that is set as the hub.
  + Keep track of current state (On/Off) of all appliances specified on the blueprint.
* Both the Hub and the mobile application will be in sync with one another by means of a common account maintained in a cloud database.
* Speech signals will be processed and valuable tokens extracted; a final decision will include three tokens: Command-Room-Device.

## 

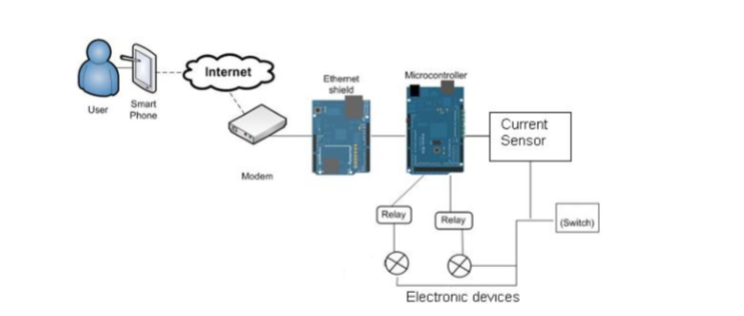
The rest of the report is organized as follows. Chapter 2 presents a literature review of the work done so far. Chapter 3 contains the implementation details of the system such as the overall design of the system, different modules, design considerations, and the methodologies that were used. Chapter 4 refers to the description of the prototype being developed including details of individual components as well as the complete system. Finally, Chapter 5 contains the conclusion.

# Literature Survey

This chapter comprises of the literature survey done both prior and during development of our project. And while we continue to come across various variants for home automation, the ones that share the most similarity are mentioned below.

## The Framework of Home Remote Automation System Based on Smartphone [1]

This section deals with a research paper published by the International Journal of Smart Home in 2015. Students from the Bina Nusantara University, School of Computer Science, in Jakarta-Indonesia worked on creating a framework (shown below) that allows users to automate devices within their home.



**Fig 2.1 Framework of Home Remote Automation System Based on Smartphone**

This research paper targeted two main tasks, namely controlling devices and providing current information.

**2.1.1 Hardware:**

Using an Arduino microcontroller that connects to a router in the house and is controlled by an Android application, signals of ON/OFF are sent to the device through a relay and current information is retrieved with the help of a current sensor.

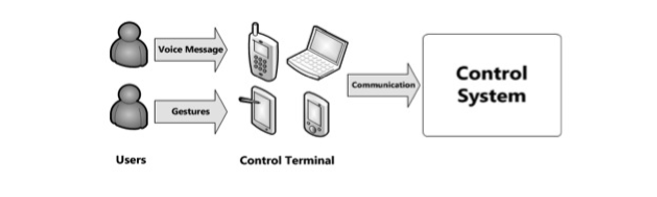
**2.1.2 Software:**

Within software development, the provided framework for Arduino, (Arduino 1.0.3) was used. For the mobile application, Titanium software with multi-platform settings was relied on. And as far as the web server was concerned, PHP language was used to create a database with MySQL programming.

While this research paper relates to development of an Android application like ours, it still relies on user commands through the application interface and hence is devoid of any form of speech recognition.

## Voice Control Based Home Automation System [2]

This project included two forms of input: Voice Message and Gestures. Students from The State University of New Jersey Rutgers worked on experimentation of two input mediums to achieve the task of automating devices around the user.



**Fig 2.2 Model of User Input**

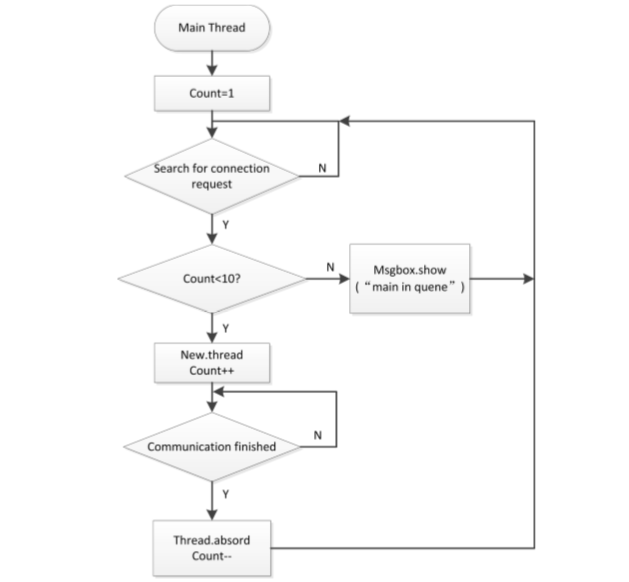
**2.2.1Domain of Voice Based Home Automation**

An Android application that takes voice command will be used to control devices and these devices controlled or monitored may include electronic ovens, motion sensors, music players etc. The Android platform will have the ability to communicate with server via LTE network or Wi-Fi. Along with this Socket communication and a multi-threaded server will also be relied on.

**2.2.2 Thread Algorithm for the Server:**

Thread algorithm allows the server to communicate with multiple clients at the same time.

In general, if the server has only one thread and once the server is in communication with a client, the thread is taken. To avoid such a problem, multi-thread method for building the server was employed.



**Fig 2.2.2 Logic Diagram of Thread Algorithm**

Perusing through the project above proved to be immensely helpful since it allowed us to understand areas of research that have been paid attention too and issues that still remain to be solved. Also, while many have dwelled within home automation, not many have relied on bringing speech recognition, together with an Android application to create one complete product. Even with the above Voice Based Home Automation project from The State University of New Jersey, certain aspects like command feedback that allowed the user to see the actual effect of their spoken command and maintenance of a cloud database for remote access are absent.

On the whole, our literature survey is not just limited to projects from student teams like ours but also includes keeping track of commercial products within the arena of Home Automation like Amazon Echo [3], Apple Homekit [4] or Google’s Nest [5]. Mainstream products such as these indicate the demand for Smart Homes within the global market, reviews for which help highlight actual limitations people come across when using them.

# Requirements and design

This chapter refers to all functional and non-functional requirements of the system as well as an overview of how the system functions and how each functionality is divided and assigned to a particular component.

## System Overview

To map voice commands to specified household devices, our application needs to be active on two platforms: one would be the personal smartphone application of the user and the second would be the Hub (another Android smartphone or tablet dedicated to accessing the cloud). The first application platform would be responsible for taking input command and processing it to extract valuable tokens. With the Hub being responsible for forwarding this processed command to the device in actual, the cloud database will be accessed both prior to execution to get current states as well as after execution of command to update current states.

## Design Considerations

Following are the major issues that were addressed or resolved for development of the complete design.

**3.2.2 Assumptions and Dependencies**

* Availability of internet.
* User educated enough to use and understand the application.
* Availability of the online cloud for database access.
* Bluetooth connectivity enabled in the smartphone.
* Relatively noise free environment expected for speech recognition to perform well.
* Smartphones with at least 4.4 KitKat Android OS.

**3.2.3 General Constraints**

* **Hardware and software environment:**

Mobile based applications have a restricted application size as compared to a desktop applications because they are stored on a smartphone. Hence our intention was to build an application that efficiently used the system’s hardware or network resources, minimizing the application size. Hardware specifications for adequate performance of our application had to be set at 1GB RAM, 4.4 KitKat OS and a working Bluetooth and Internet connection adapter. Switchboards also needed to be configured prior to use, which included setting up the Bluetooth based Arduino setup with each switchboard.

* **End-user environment:**

For the end user, it was imperative to be within the Bluetooth range. Coherent speech was a pre-requisite for effective speech recognition and only valid English words recognized by the Google Speech API could be used as voice commands. Also a command was considered complete and valid only if it contained 3 token of information: **room**, **device** and **action to be performed (ON/OFF).**

* **Availability or volatility of resources:**

Our Android application required enough memory space on the client smartphone and good latency, ping and bandwidth speed from the internet connection was also required for reliable and speedy communication between the cloud and the Hub.

* **Standards compliance:**

Resources were managed effectively within all mobile activities as per performance standards to ensure that the application was not killed by the OS.

* **Data repository and distribution requirements:**

Amazon Web Services were used to maintain cloud database and this allowed access to information without any dependence on local smartphone storage.

* **Memory and other capacity limitations:**

Smartphone needed enough memory to store GUI, Local Database and other required resources of the application.

* **Performance requirements:**

The performance of our application highly depended on the internet facility as all the data was to be retrieved from the online cloud database so high internet speed was required.

* **Verification and validation requirements:**

A login system authenticated the user was present to confirm from the online database whether that person was registered to the application or not.

## Goals and Guidelines

Following are the goals and the guidelines kept in mind while developing the application:

* **Simple and Easy to Use:**

Our goal was to keep things as simple as we could. We tried to incorporate most principles previously studied in Human Computer Interaction so that the user interface was easy and interesting enough to involve the user.

* **Performance:**

We had a substantial amount of data that needed to be processed in real time and since that data had to be used by all applications and processes of different users running at the same time, so we opted for efficient ways of access. The activity access and load time was efficient enough which is why the user did not find it too slow or bulky.

* **Flexibility in Voice Commands:**

As mentioned earlier, a valid voice command required the user to enter room, device and action information and our aim was not to restrict the user to any specific format and instead provide enough flexibility to the user. However, there had to be certain assumptions taken within our speech processing module that included the fact that the user would provide the complete necessary information (Room, Device, and Action).

* **Feedback for Commands:**

Digital current or voltage feedback was incorporated to allow the user to observe the effect of the voice command even when the device was not in front of the user or when the user was not at home.

## Development Methods

We divided our application into different modules, each assigned to a different member, so as to develop it concurrently in a small period of time. We targeted the hardware module first and worked on establishing connection between actual sample hardware (LED’s and Arduino) and our mobile application. Whilst doing this, we also worked on our speech processing module and this mode of development allowed us to work on modules concurrently according to their importance under the umbrella of **Agile Development Model.** Such a development model not only speeded our testing but also allowed us to divide work amongst all team members.

## Architectural Strategies

* **Cloud Services**

Android applications usually require a cloud backend database for the permanent storage of data. In our case, not only did our cloud database include the login information of the users but also the registered technical blueprint of the house and the current states of the devices installed, were maintained as well.

* **Reuse of existing software components to implement various features of the application:**

Most Web Services provide a built in basic security and performance features which was used to offer a secure and efficient experience for the users logged into the application. The Google and Google Play services included in the Android studio package were used as well. Along with that the Google Voice Search API for American English language was relied on to send voice commands from the application.

* **Future plans for extending or enhancing the software:**

A possible extension for our application could include work on commands spoken to in Urdu. Further extension could involve using WIFI along with Bluetooth to improve connectivity beyond the Bluetooth range. Along with this, sensors could be incorporated to include security tracking too.

* **User interface paradigms:**

The interface of the application was simple, easy to use and to our best effort in accordance with the paradigms of Human Computer Interaction.

* **Data/Memory and management policies:**

We used parsing pre-processing techniques on the voice commands to minimize the data storage in the application so as to keep the application memory efficient. A temporary local database was also maintained in the mobile in order to hybridize the application to a small extent. As our application was highly dependent on getting the information from third party sources like the electronic devices attached, hence we only acquired the required information from them to minimize the access time from other sources as well as the data cost.

* **Distributed system:**

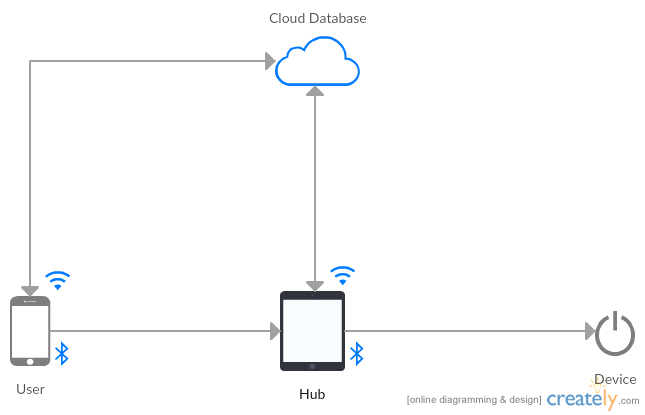
Multiple users were able to access the application in real time to manage the electronic devices in the registered house.

## System Architecture

This section includes a diagrammatic overview of the complete home automation system.

## 3.6.1 General Framework

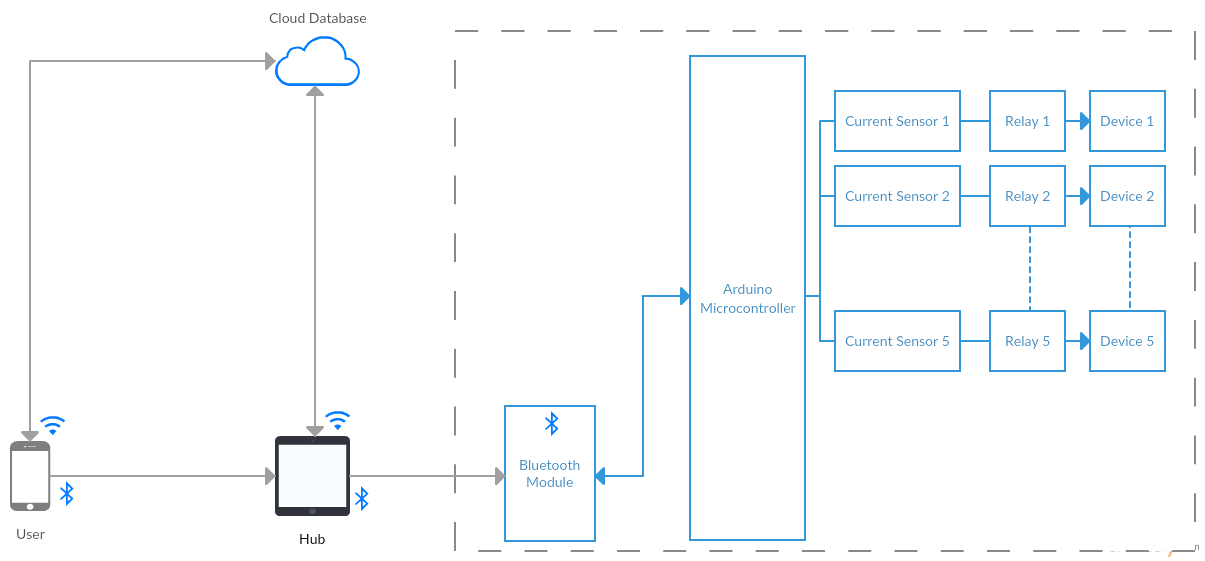
The following image shows the connectivity between all components of the system.



**Fig 3.6.1: General Framework**

## 3.6.2 Complete System Architecture

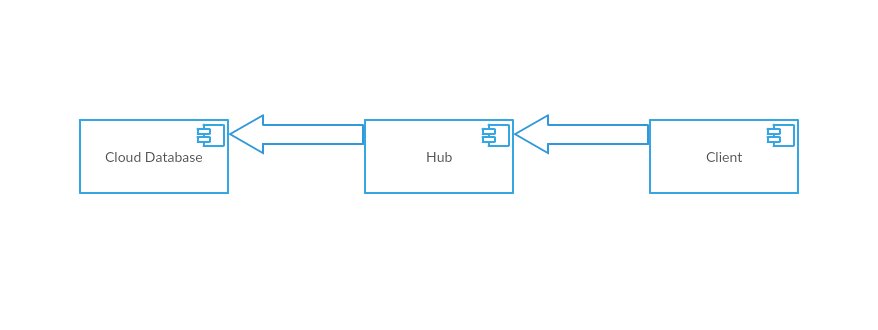
The following image refers to the inner details of the electronic components involved.

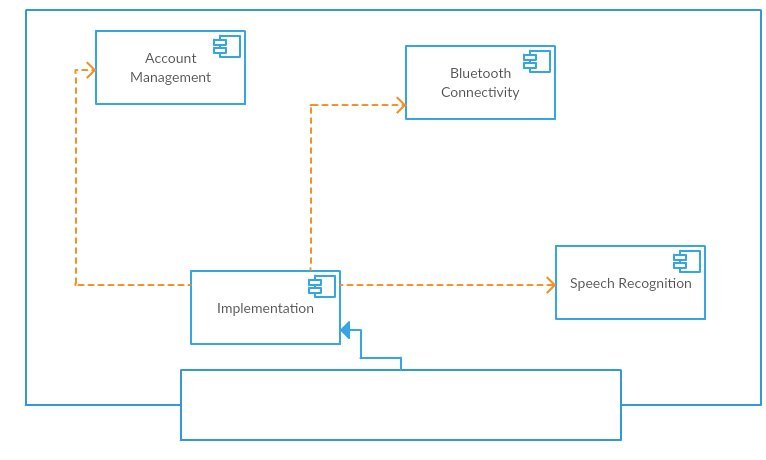


**Fig 3.6.2: Complete System Architecture**

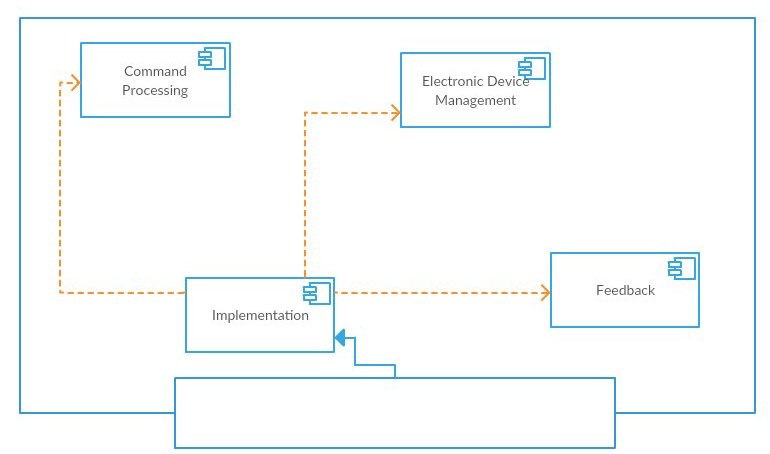
## 3.6.3 Component Diagram

Fig 3.6.3 depicts the flow of information between all components of the system with the diagrams following ahead showing the inner details.

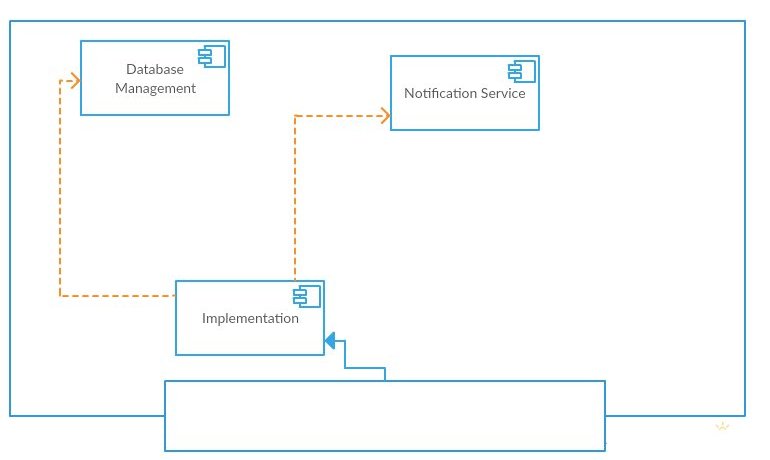
 **Fig 3.6.3: General Component Diagram**



**Fig 3.6.3.1: Client Component**



**Fig 3.6.3.2: Hub Component**

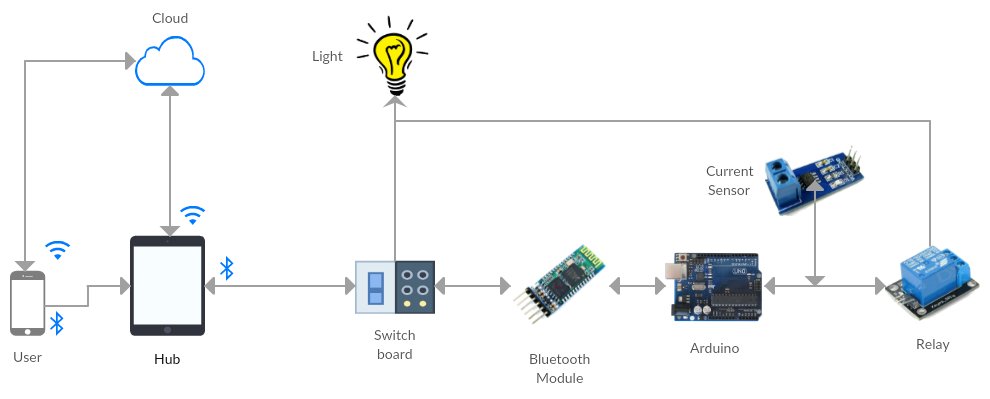


**Fig 3.6.3.3: Cloud Database Component**

To conclude this chapter, it could be summarized that the architectural design and analysis relied on for the development of the complete Android application was outlined here.

# Description of Prototype

This chapter describes the final prototype inclusive of hardware components as well as their connection with the software being developed.



**Fig 4.1: Basic Level Prototype for LED Control**

As mentioned earlier, Voice Controlled Home Automation is an electronic arrangement that lets the user control electronic devices through an Android smartphone.

The product consists of two parts.

* Android application.
* Electronic architecture.

To understand the figure above, the connection amongst various components shown needs to be understood first. A brief description is as follows:

* The user speaks the command into the application on his Android phone.
* This command is parsed into tokens and sent to the hub which controls the devices.
* The hub is another Android device that acts as a main server and is connected to the switchboard via Bluetooth.
* The parsed command is sent to the Arduino microcontroller which allows the current to flow through to the relay which amplifies the low power signal.
* The current sensor acts as a feedback module that assures the user if any current is flowing through it.

This description above refers to the simplified version of the data flow existing amongst all components, making the prototype explanation far easier to understand. What needs to be noted is the fact that the pictorial representation above is only for a particular device type i.e. an LED.

# Implementation

For this chapter, details of the algorithms that were implemented along with the platform and APIs used are discussed along with an explanation for their merit contribution.

* 1. **Hub App Module:**

Like any other substantially extensive Android application, our work also incorporated the following core Android components.

1. **Activities and Intents**
2. **Custom Views**
3. **Fragments**
4. **XML Layouts**
5. **Resources**
6. **Manifest and Permissions**
7. **Adapters**
8. **Intent Services**
9. **Asynchronous Threads**
10. **Broadcast Receivers**
11. **Local Database**
12. **Cloud Database with Firebase Cloud Services Integration**
    1. **User App Module:**

This module work was completed once the Hub App was fully integrated with the cloud. It, too, employed the same design components listed above.

* 1. **Firebase Cloud Integration:**

Firebase Cloud Messaging (FCM) is a cross-platform messaging solution that allowed us to reliably deliver messages at no cost. Our FCM implementation included two main components for sending and receiving:

1. An environment such as Cloud Functions for Firebase to build, target and send messages.
2. An Android client app that received the sent messages.
   1. **Speech Recognition Module:**

Cloud Speech API enabled easy integration of Google speech recognition technologies into our Smart home application. Extracting audio to receive a text transcription was accomplished with the Cloud Speech API service. Using the most advanced deep learning neural network algorithms, the API was able to convert the user's audio to text with significant accuracy. The long term benefit is that the Speech API accuracy improves over time as Google improves the internal speech recognition technology used by all Google products.

To customize the recognition to our needs, we also created a dictionary that maintained the names of all rooms and devices that the user added. Mapping to this dictionary was made possible with the **Minimum Edit Distance Algorithm.**

As per a Wikipedia definition, Edit distance is a way of quantifying how dissimilar two strings are to one another by counting the minimum number of operations required to transform one string into the other. And this proved to helpful since with this, we were able to map a recognized word to the closest mapping that existed in our dictionary. This decision was based on a score threshold and allowed us to make a reasonable guess of what the user meant even if it was not recognized correctly.

* 1. **Development with Arduino Development Boards, Bluetooth and Wi-Fi Module:**

Arduino UNO Development boards with embedded microcontrollers acted as an abstraction for real life switchboards and were programmed efficiently through personalized Arduino IDE. Furthermore, an algorithm was developed and implemented on said Arduino boards to automate mechanical switches. So, when in function, the mechanical switches are not controlled by 220V input supply but the Arduino boards.

Bluetooth HC-05 modules were used to establish Bluetooth based communication channel between smart phones and Arduino boards. For simplification, they were also programmed to be renamed via Arduino development boards.

Relays were installed to control the circuit using our Arduino’s low powered signal to extend our algorithm from 3.5V LEDs to 220V electronic devices.

For Wi-Fi based communication between the smartphone and Arduino board, we used ESP 866 Wi-Fi modules.

* 1. **Current Sensor Feedback:**

A current feedback system was implemented to provide confirmation that the device has turned on or off.

Current Sensor ACS712 30A current sensors were used to sense the instantaneous amount of current flowing in the circuit. The current sensor was installed in series with the relay which allowed us to read the amount of current flowing from the relay at a particular moment in time.

Arduino UNO board consistently calculates the value of current for all sensors attached to it and returns the value of current on command.

This chapter is concluded with the intention that the development procedures and components used are mentioned with substantial detail.

# Test Cases

Test cases mentioned below are based not only on the application flow but also on the integral features that need to be present to ensure the smooth working of our mobile application. The idea is to ascertain the credibility of these very features by running the following test cases.

**Note**: Corresponding Use Case ID’s mentioned for each test case match the Use Case ID’s specified in Functional Specifications Document submitted previously.

**5.1 Create Profile.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | 5.1 | | QA Test Engineer | | Manaal Ayub |
| **Test case Version:** | | 1 | | **Reviewed By** | | Zain Saleem |
| **Test Date:** | | 4-03-2017 | | **Use Case Reference(s)** | | UC 1.1 |
| **Revision History** | | None | | | | |
| **Objective** | | View successful completion of task | | | | |
| **Product/Ver/Module** | | Hub/User App Module | | | | |
| **Environment:** | | Android Version > KitKat | | | | |
| **Assumptions:** | | Application must have a reliable internet connection. | | | | |
| **Pre-Requisite:** | | User is not associated with an existing Firebase instance. | | | | |
| **Step No.** | Execution description | | | | **Procedure result** | |
| *1.* | User enters family email address, password to create family account. | | | | System checks Firebase for any existing similar data. Lack of such an occurrence leads to creation of a new family account:  Proceeds to Step 2.  However, upon finding the email address already in use:  Proceeds to Step 3. | |
| *2.* | User enters family email address, user email address, and password along with phone number to create a user account, once the family account is created. | | | | System checks Firebase for a matching family account. Upon finding a match, creation of a new user account is completed.  Proceeds to Step 4. | |
| *3.* |  | | | | Corresponding error shown to user. Proceeds to Step 1. | |
| *4.* | User successfully logs into account. | | | |  | |
| Comments  System response was as documented. | | | | | | |
|  | | | *Passed* *Failed* *Not Executed* | | | |

**5.2 Log into Profile.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | 5.2 | | QA Test Engineer | | Manaal Ayub |
| **Test case Version:** | | 1 | | **Reviewed By** | | Zain Saleem |
| **Test Date:** | | 4-03-2017 | | **Use Case Reference(s)** | | UC 1.2 |
| **Revision History** | | None | | | | |
| **Objective** | | View successful completion of task | | | | |
| **Product/Ver/Module** | | Hub/User App Module | | | | |
| **Environment:** | | Android Version > KitKat | | | | |
| **Assumptions:** | | Application must have a reliable internet connection. | | | | |
| **Pre-Requisite:** | | User is not associated with an existing Firebase instance. | | | | |
| **Step No.** | Execution description | | | | **Procedure result** | |
| *1.* | User enters email address and password to log into account. | | | | System checks Firebase for any existing similar data.  Upon finding a match:  Proceeds to Step 3.  Upon lack of such an occurrence:  Proceeds to Step 2. | |
| *2.* |  | | | | Corresponding error shown to user. Proceeds to Step 1. | |
| *3.* | User successfully logs into account. | | | |  | |
| Comments  System response was as documented. | | | | | | |
|  | | | *Passed Failed Not Executed* | | | |

**5.3 Add a Room.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | 5.3 | | QA Test Engineer | | Manaal Ayub |
| **Test case Version:** | | 1 | | **Reviewed By** | | Zain Saleem |
| **Test Date:** | | 5-03-2017 | | **Use Case Reference(s)** | | UC 1.6 |
| **Revision History** | | None | | | | |
| **Objective** | | View successful completion of task | | | | |
| **Product/Ver/Module** | | Hub/User App Module | | | | |
| **Environment:** | | Android Version > KitKat | | | | |
| **Assumptions:** | | Application must have a reliable internet connection. | | | | |
| **Pre-Requisite:** | | User is not associated with an existing Firebase instance. | | | | |
| **Step No.** | Execution description | | | | **Procedure result** | |
| *1.* | User clicks on Add Room and chooses a room type from a list of types. | | | | A new activity fragment is triggered that takes in details of the new room. | |
| *2.* | User enters a room name that will be associated with that room. | | | | If the name entered is valid and does not already exist:  Proceeds to Step 4.  Else:  Proceeds to Step 3 | |
| *3.* |  | | | | User is prompted to enter a new room name.  Proceeds to Step 2. | |
| *4.* | User successfully adds a new room. | | | |  | |
| Comments  System response was as documented. | | | | | | |
|  | | | *Passed Failed Not Executed* | | | |

**5.4 Add a Device.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | 5.4 | | QA Test Engineer | | Manaal Ayub |
| **Test case Version:** | | 1 | | **Reviewed By** | | Zain Saleem |
| **Test Date:** | | 5-03-2017 | | **Use Case Reference(s)** | | UC 1.6 |
| **Revision History** | | None | | | | |
| **Objective** | | View successful completion of task | | | | |
| **Product/Ver/Module** | | Hub/User App Module | | | | |
| **Environment:** | | Android Version > KitKat | | | | |
| **Assumptions:** | | Application must have a reliable internet connection. | | | | |
| **Pre-Requisite:** | | User is not associated with an existing Firebase instance. | | | | |
| **Step No.** | Execution description | | | | **Procedure result** | |
| *1.* | User clicks on Add Device after selecting an existing switchboard or clicking on Add Switchboard. | | | | A new activity fragment is triggered that takes in details of the new device. | |
| *2.* | User enters a device name that will be associated with that room along with a port number. | | | | If the name and port number entered is valid and does not already exist:  Proceeds to Step 4.  Else:  Proceeds to Step 3 | |
| *3.* |  | | | | User is prompted to enter a new device name.  Proceeds to Step 2. | |
| *4.* | User successfully adds a new device. | | | |  | |
| Comments  System response was as documented. | | | | | | |
|  | | | *Passed Failed Not Executed* | | | |

**5.5 Configure Bluetooth.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | 5.5 | | QA Test Engineer | | Manaal Ayub |
| **Test case Version:** | | 1 | | **Reviewed By** | | Zain Saleem |
| **Test Date:** | | 5-03-2017 | | **Use Case Reference(s)** | | UC 1.7 |
| **Revision History** | | None | | | | |
| **Objective** | | View successful completion of task | | | | |
| **Product/Ver/Module** | | Hub/User App Module | | | | |
| **Environment:** | | Android Version > KitKat | | | | |
| **Assumptions:** | | Application must have a reliable internet connection. | | | | |
| **Pre-Requisite:** | | User is not associated with an existing Firebase instance. | | | | |
| **Step No.** | Execution description | | | | **Procedure result** | |
| *1.* | User clicks on Configure Bluetooth after switching on mobile Bluetooth. | | | | A new activity fragment is triggered that shows the user a list of Bluetooth devices available. | |
| *2.* | User clicks on the Bluetooth module that is associated with the room that he wants to configure Bluetooth module for. | | | | Mac Address of chosen Bluetooth module is added with the corresponding switchboard and room. | |
| *3.* | User successfully configures a Bluetooth module for a corresponding switchboard. | | | |  | |
| Comments  System response was as documented. | | | | | | |
|  | | | *Passed Failed Not Executed* | | | |

**5.6 Enter Command Via Buttons.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | 5.6 | | QA Test Engineer | | Manaal Ayub |
| **Test case Version:** | | 1 | | **Reviewed By** | | Zain Saleem |
| **Test Date:** | | 5-03-2017 | | **Use Case Reference(s)** | | UC 1.5 |
| **Revision History** | | None | | | | |
| **Objective** | | View successful completion of task | | | | |
| **Product/Ver/Module** | | Hub/User App Module | | | | |
| **Environment:** | | Android Version > KitKat | | | | |
| **Assumptions:** | | Application must have a reliable internet connection. | | | | |
| **Pre-Requisite:** | | User is not associated with an existing Firebase instance. | | | | |
| **Step No.** | Execution description | | | | **Procedure result** | |
| *1.* | User chooses a room from the list of rooms created. | | | | A new activity fragment is triggered that shows the user a list of switchboards present within the chosen room. | |
| *2.* | User chooses a switchboard from the list of existing switchboards. | | | | A new activity fragment is triggered that shows the user a list of devices associated with a particular room. | |
| *3.* | User chooses device and clicks on the button to successfully power on or power off the socket. | | | |  | |
| Comments  System response was as documented. | | | | | | |
|  | | | *Passed Failed Not Executed* | | | |

**5.7 Enter Command Via Voice.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | 5.7 | | QA Test Engineer | | Manaal Ayub |
| **Test case Version:** | | 1 | | **Reviewed By** | | Zain Saleem |
| **Test Date:** | | 5-03-2017 | | **Use Case Reference(s)** | | UC 1.4 |
| **Revision History** | | None | | | | |
| **Objective** | | View successful completion of task | | | | |
| **Product/Ver/Module** | | Hub/User App Module | | | | |
| **Environment:** | | Android Version > KitKat | | | | |
| **Assumptions:** | | Application must have a reliable internet connection. | | | | |
| **Pre-Requisite:** | | User is not associated with an existing Firebase instance. | | | | |
| **Step No.** | Execution description | | | | **Procedure result** | |
| *1.* | User clicks on visible microphone and enters voice command. | | | | Voice command is interpreted and upon finding it meaningful:  Proceeds to Step 3.  Upon finding no meaningful match for the given voice command:  Proceeds to Step 2. | |
| *2.* |  | | | | User is prompted to enter the voice command again. | |
| *3.* | User voice command is successfully mapped to correct device socket. | | | |  | |
| Comments  System response was as documented. | | | | | | |
|  | | | *Passed Failed Not Executed* | | | |

**5.8 Receive Visual Feedback.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | 5.8 | | QA Test Engineer | | Manaal Ayub |
| **Test case Version:** | | 1 | | **Reviewed By** | | Zain Saleem |
| **Test Date:** | | 5-03-2017 | | **Use Case Reference(s)** | | UC 1.5,1.4 |
| **Revision History** | | None | | | | |
| **Objective** | | View successful completion of task | | | | |
| **Product/Ver/Module** | | Hub/User App Module | | | | |
| **Environment:** | | Android Version > KitKat | | | | |
| **Assumptions:** | | Application must have a reliable internet connection. | | | | |
| **Pre-Requisite:** | | User is not associated with an existing Firebase instance. | | | | |
| **Step No.** | Execution description | | | | **Procedure result** | |
| *1.* | User chooses a room from the list of rooms created. | | | | A new activity fragment is triggered that shows the user a list of switchboards present within the chosen room. | |
| *2.* | User chooses a switchboard from the list of existing switchboards. | | | | A new activity fragment is triggered that shows the user a list of devices associated with a particular room. | |
| *3.* | User chooses device and clicks on the button to successfully power on or power off the socket. | | | | Device figure changes turns bright yellow upon powering on. And turns white upon powering off. | |
| Comments  System response was as documented. | | | | | | |
|  | | | *Passed Failed Not Executed* | | | |

**5.9 Receive Current Feedback.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | 5.9 | | QA Test Engineer | | Manaal Ayub |
| **Test case Version:** | | 1 | | **Reviewed By** | | Zain Saleem |
| **Test Date:** | | 5-03-2017 | | **Use Case Reference(s)** | | UC 1.5,1.4 |
| **Revision History** | | None | | | | |
| **Objective** | | View successful completion of task | | | | |
| **Product/Ver/Module** | | Hub/User App Module | | | | |
| **Environment:** | | Android Version > KitKat | | | | |
| **Assumptions:** | | Application must have a reliable internet connection. | | | | |
| **Pre-Requisite:** | | User is not associated with an existing Firebase instance. | | | | |
| **Step No.** | Execution description | | | | **Procedure result** | |
| *1.* | User chooses a room from the list of rooms created. | | | | A new activity fragment is triggered that shows the user a list of switchboards present within the chosen room. | |
| *2.* | User chooses a switchboard from the list of existing switchboards. | | | | A new activity fragment is triggered that shows the user a list of devices associated with a particular room. | |
| *3.* | User chooses device and clicks on the button to successfully power on or power off the socket. | | | | New device current reading is shown to user. | |
| Comments  System response was as documented. | | | | | | |
|  | | | *Passed Failed Not Executed* | | | |

**5.10 Logout of Profile.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | 5.9 | | QA Test Engineer | | Manaal Ayub |
| **Test case Version:** | | 1 | | **Reviewed By** | | Zain Saleem |
| **Test Date:** | | 5-03-2017 | | **Use Case Reference(s)** | | UC 1.5,1.4 |
| **Revision History** | | None | | | | |
| **Objective** | | View successful completion of task | | | | |
| **Product/Ver/Module** | | Hub/User App Module | | | | |
| **Environment:** | | Android Version > KitKat | | | | |
| **Assumptions:** | | Application must have a reliable internet connection. | | | | |
| **Pre-Requisite:** | | User is not associated with an existing Firebase instance. | | | | |
| **Step No.** | Execution description | | | | **Procedure result** | |
| *1.* | User clicks on logout. | | | | A new activity fragment is triggered that shows the user the home page of the application. | |
| *2.* | User successfully logs out. | | | |  | |
| Comments  System response was as documented. | | | | | | |
|  | | | *Passed Failed Not Executed* | | | |

Conclusively, while the above test-cases may not provide full coverage testing, they do assure the smooth working of the important functionalities. And while other cases, for instance, boundary conditions and exceptions have been tested too, they have not been documented for the sake of brevity.

# Experimental Results and Analysis

This chapter deals with algorithmic as well as hardware level experimentation that we have engaged in, during the duration of both FYP-I and FYP-II.

## Algorithmic Experimentation:

Information Retrieval techniques including parsing, tokenization, stop-wording, stemming and dictionary maintenance are being used along with the implementation of Minimum Edit Distance Algorithm to process input commands retrieved from the speech recognition module. While previously our work led to the conclusion that other algorithms for auto-correction will need to be experimented, we did reach a consensus that the best choice would be Minimum Edit Distance since it did suit our requirements as opposed to other options available.

## Hardware Level Experimentation:

The initial part of our project was exploration of Arduino **IDE** as well as configuration of the **Bluetooth module** and **Arduino Development Board** with our Android application. Our analysis of the work done for now, highlights the need for improved robustness which seems to be deficient when working with the current Bluetooth module, which is why we are now working on the use of WIFI module as an extra support for increased range communication. Our work with current sensors and relays, however, was as expected, and we were able to use them effectively.

## Experimentation Results:

Results for both hardware and algorithmic experimentation can only be reported through pictorial representation of an actual voice command mapping. The figures ahead attempt to depict the voice command mapping for switching on an LED.

For the current developed software and hardware model, the following steps portray the input and output against a particular voice command.

**Step 1**: Pairing of Bluetooth Module with Smartphone Application.

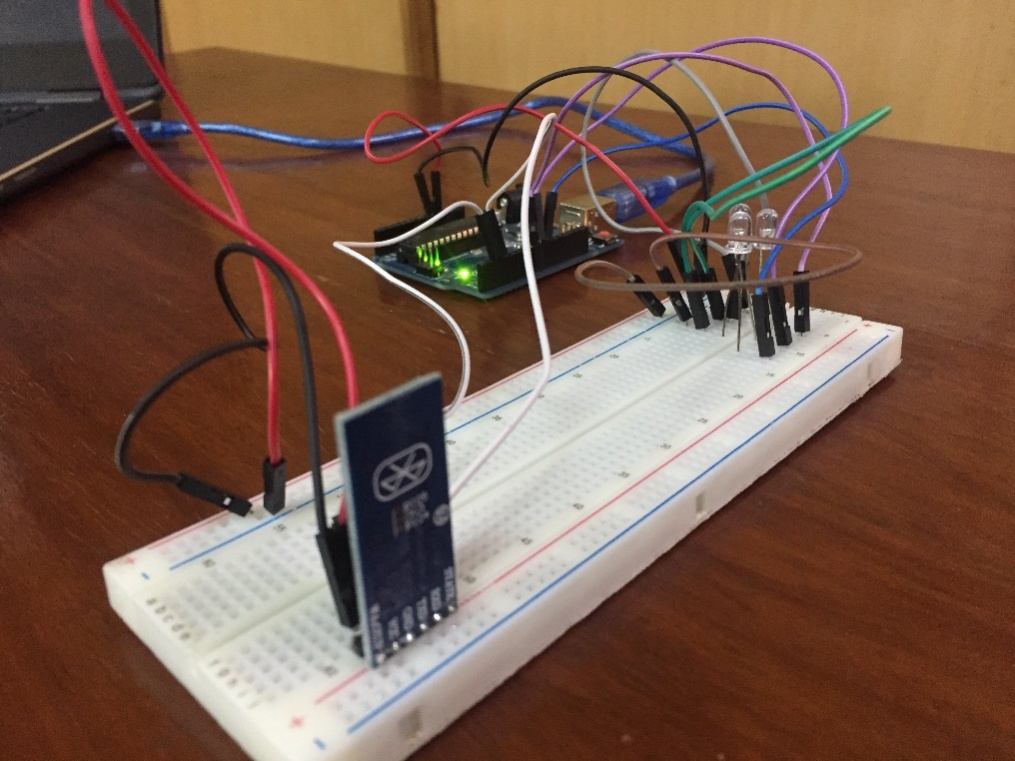
**Step 2**: Inclusion of all port numbers for LED’s to be automated.

**Step 3**: Input voice command: “LED One On”.

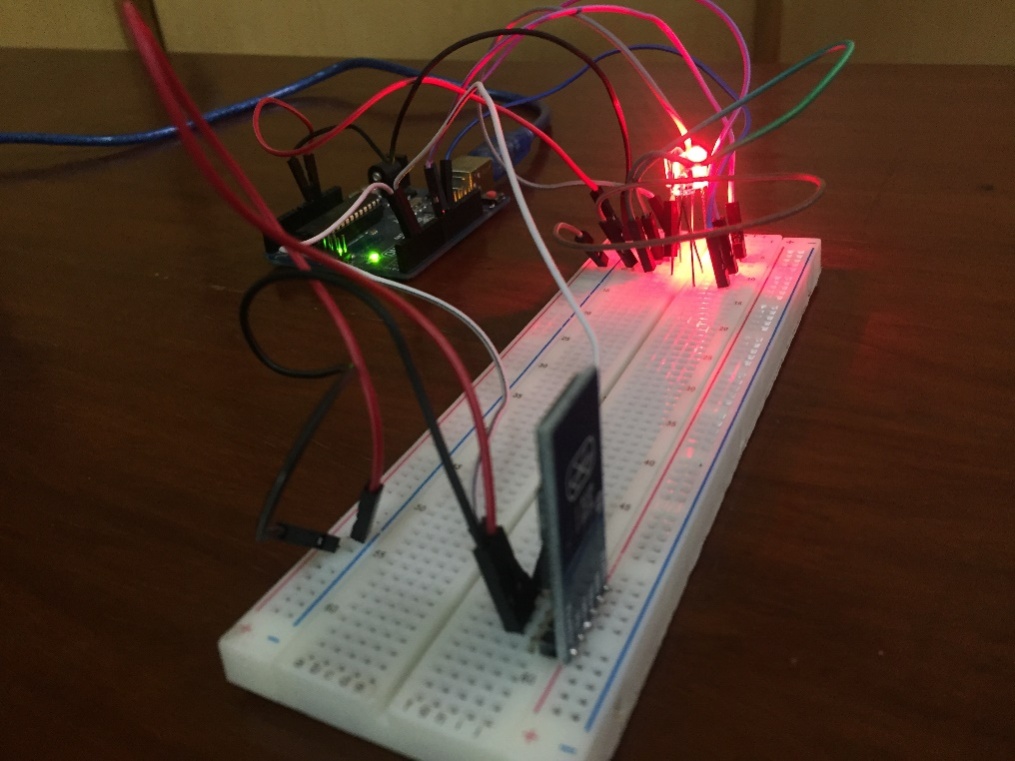
**Step 4**: This command results in LED One switching on.

Hardware Shown in the Figures includes:

* Arduino Development Board.
* Bluetooth Module.
* LED’s.
* Jumper Wires.
* Breadboard.



**Fig 5.1: Before Input Voice Command**



**Fig 5.2: After Input Voice Command**

Continued experimentation and analysis has helped us improve our development with each iteration. And while new improvements bring about new issues, this cycle of experimentation is on-going and continues to play a key role in many of our decisions regarding new and better algorithms or additional hardware components.

# Conclusions

This chapter revisits the work completed and gives the gist of the complete project.

To sum it all up, within our project, we worked on the following core areas:

* Completion of the Hub Android Application.
* Development of Personal Android Application
* Extension to high power electronic devices.
* Maintenance of a Cloud Server.

**Completion of Hub and Personal Application Development:**

As mentioned earlier, our development included two Android applications: one of which was the Hub and other being the Personal one, for each member of the family.

**Extension to High Power Electronic Devices:**

The idea was to extend our work from control of low-voltage devices to control and operate real life high power electronic devices like lights or fans etc.

**Firebase Cloud Server Setup:**

A cloud server that stores all the information about the user and the respective electronic arrangement and their states, was setup and communication between our two apps was synchronised via this online database.

Finally, if we are to encapsulate the complete idea of our final year project in just a few lines, it could be said that we attempted to create a cost-effective version of a smart home application that makes an effort to reduce overwhelming configuration protocols and device brand restrictions to bring forth an IoT based framework that allows for remote access via voice commands. Our work also led us to other areas of interest, like security of the application and robustness that could be pursued in the future for a more well-rounded approach. While software development based on the Android operating system remained an integral area throughout the course of our work, automation of electronic hardware and circuitry was an interesting area to explore as well, since it permitted us to acquire a level of understanding for embedded systems. Conclusively, we investigated the cost restriction challenge that needs to be addressed by developers of smart home applications so as to make IoT for household appliances a more accessible and well-received concept.

# References

**Proceedings paper**

1. Akbar Satria, Muhammad Luthfi Priadi, Lili Ayu Wulandhari and Widodo Budiharto, “The Framework of Home Remote Automation System Based on Smartphone”, in *International Journal of Smart Home*, Vol. 9, No. 1 (2015), pp. 53-60.

**Technical Report**

1. Li Xinyu, Gu Yue, Guo Jiaqi, Nayyar Vidur, Shu Chang, Zhang Lufan, “Voice Control Based Home Automation System,” The State University of New Jersey Rutgers, Department of Electrical and Computer Engineering, Technical Report, Fall 2013.

**World Wide Web**

1. “Amazon Echo - Alexa Enabled”,

[*https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E*](https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E)*.*

[4] “Home – Nest”, <https://nest.com/>

[5] “iOS Apple Home” , http://www.apple.com/ios/home/