Low cost Smart Home application

Table of Contents

1. Introduction

During the last several years, the use and development of mobile applications has skyrocketed. Nowadays, everyone in the modern world has the ability to obtain a mobile phone. As a result, developers have had to adapt and broaden their market to include a wider spectrum of consumers. You can discover everything from applications for kids to learn the alphabet to apps for seniors to manage their insulin levels in the Apple Store or Google Play.

This paper will also look at the relationship between smart devices and the Internet of Things (IoT), which is one of the hottest topics right now, which develops the concept of connecting smart devices through the Internet.

1.1. Objectives

The aim of this assignment is to develop a smart home mobile application that will improve people's standard of living at home. It will focus on making people's lives easier by sparing them time and money. The purpose of the project is to create a user-friendly application that will improve the user's interaction with smart devices.

1.2. Functionalities

In this part, I will go over the app's functionality:

- Create an account and login: The user will be able to create an account and sign up to the application to gain access to the facility. Social network logins, such as Facebook, should not be used in applications which require protection to control something private, such as a house, because they can endanger the user's data.

- Lights: The app will be capable of turning on and off the lights inside the house.

- Fan: the app will have a feature that allows you to control the fan.

- Temperature sensor: people who have the app installed will be able to check the temperature of the house at any time.

2. Background and Research

2.1 Internet of things

The concept of Internet of Things, or IoT, is gaining popularity. IoT does not have a formal definition. Although multiple groups of academics have defined this concept, Ashton (2009)[1] is credited with the first use of the word: “Our economy, society and survival are not based on ideas or information – they are based on things. Ideas and information are important, but things matter much more”. According to Aggarwal and Lal Das (2012)[2], the IoT can also be understood as just a worldwide network that permits interactions between humans, human-to-things, and things-to-things.

If it has the potential to connect to the internet, any object could be converted into an IoT device. According to Patel (2020)[3], because all systems are connected, it is easier to control various functions while also saving money, energy, and time. Therefore, the Internet of Things is becoming increasingly integrated into our daily lives. The graph below illustrates the variety of devices linked this year (30,73 billion) and the forecast for 2025, which is expected to exceed 75 billion devices.

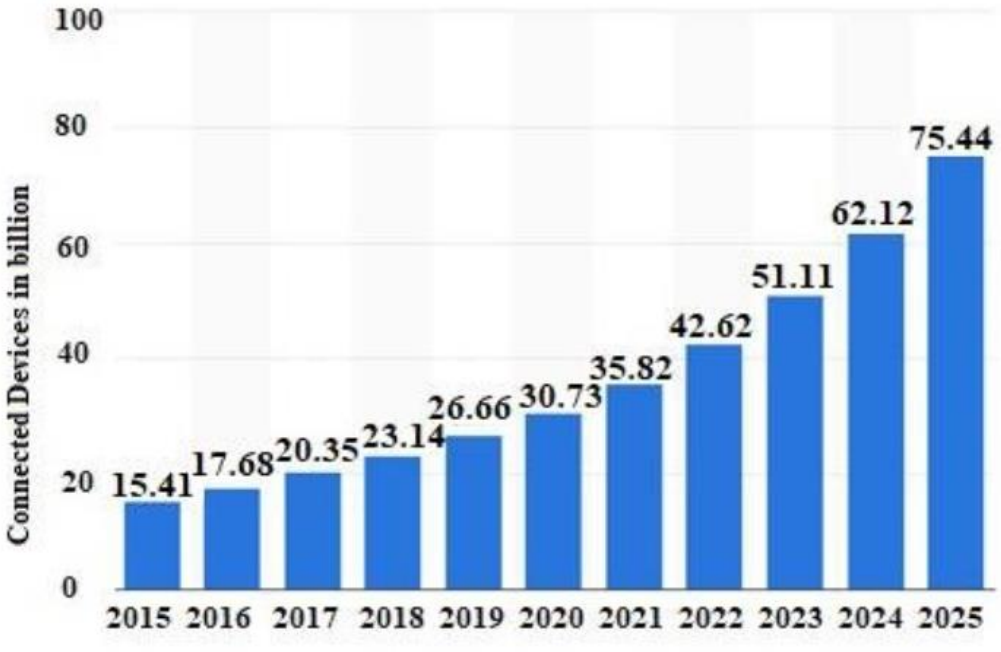

Figure 1: Internet of Things connected devices in billions. (ResearchGate 2018)

2.2. Technology

2.2.1. Android studio

Android works with millions of smart phones. Android has become the preferred platform for users and developers due to the freedom it offers. Simultaneously, the open system of Android makes it more vulnerable to hackers.

One of the most important features of Android Studio is that it comes with an emulator that could be used to test your app on a variety of Android devices. The emulator simulates practically all of the features of a real Android device, including calls and access to the Google Play Store, among other things.

2.2.2. Firebase

Firebase is a platform that helps in the development and support of Android apps and includes features like:

- Real time database: This functionality is simple to utilize. It utilizes data synchronization, which guarantees that every device that is connected receives data updates instantly (2020)[4]. Firebase real-time events continue to run even when the user is offline.

- Google Cloud Storage: images, videos and audios can be stored within the application. The information is highly secure, which means it will restart from where it left off if a network error happens.

2.2.3 Raspberry Pi

Raspberry Pi is a low-cost microcomputer that can connect to any HDMI or RCA video input device. It is inexpensive in comparison to other computers on the industry. One of the greatest things about the it is how many different things you can do with it.

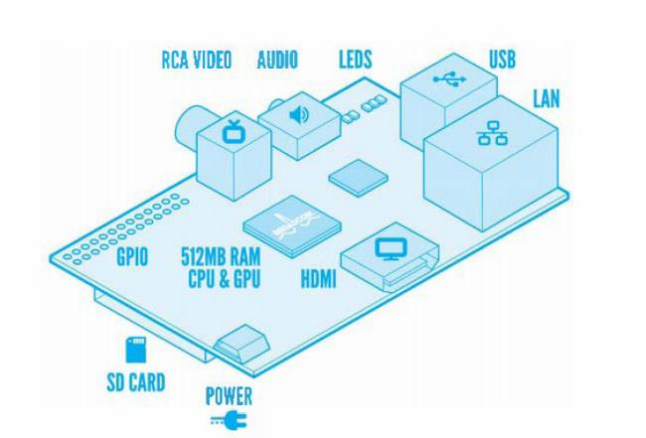

Figure 2: Raspberry pi model B (ieeexplore.ieee.org 2017)

You can play around with the product and make it into something new. The SD card may be simply exchanged, allowing you to modify its functionalities without having to update the software. This gadget may be charged with any mobile phone charger. Although the GPIO has no unique purpose, having it incorporated in the hardware allows the user to avoid having to collect any extra electronics. The Raspberry Pi also has an RCA video and audio jack, letting it to play videos with a quality of 1080p.

Despite the fact that the Raspberry Pi can perform multiple tasks at once, it has some limits owing to its technology. It can't run Windows since the Raspberry Pi doesn't support X86 operating systems. "The Raspberry Pi can serve as a personal computer, but it cannot replace it," Varguese (2017)[5] noted.

2.2.4 Python

Python is a powerful coding language that is known for being simple to learn since it allows you to construct programs with less lines of code than other programming languages.

3. Requirement and Analysis

The requirements have been separated into two groups based on their priority: (1) Requirements that must be met in order for the application to execute. (2) Optional needs: The program will function even if these prerequisites are not met.

	Requirement	**Priority**
1	A new user must complete the registration process on the register page.	1
2	Registered user must login to run the app.	1
3	If the user is logged in, the programme must launch immediately.	2
4	The temperature of your home must be displayed right away.	1
5	Lights must have the ability to turn on and off.	1
6	Fan must have the ability to turn on and off.	1
7	User must be able to sign out	1

Table 1: List of requirements

A flow chart has been added in the document to provide a better view of the application's performance. Flow chart is a diagram that includes rectangles, ovals and numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence (2020)[6].

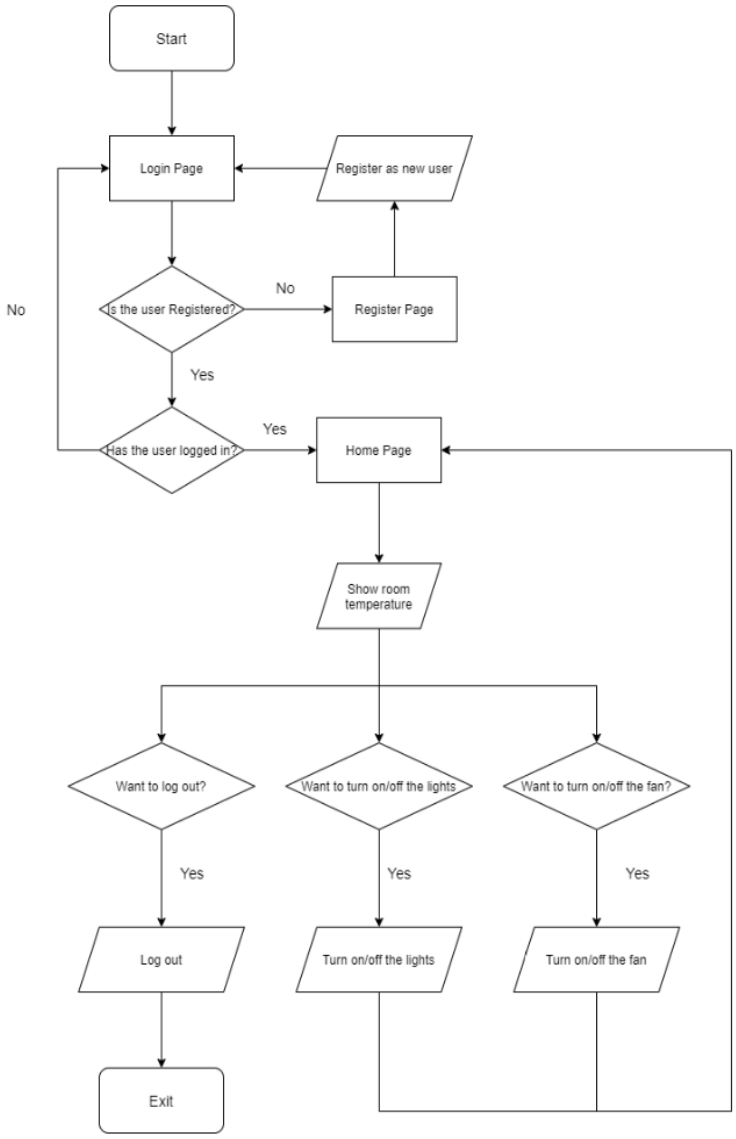

Figure 3: Flow chart

4. Architecture of project

4.1 Hardware

First, I didn't have all the materials I needed for my project, so I had to order the majority of them online. The most difficult part was getting it into the Raspberry Pi with the necessary wires, components, and resistors. After a long search, I discovered a wealth of tutorials at https://www.freenove.com/. This website allows you to purchase Raspberry Pi components as well as detailed instructions on how to connect the hardware pieces based on your project.

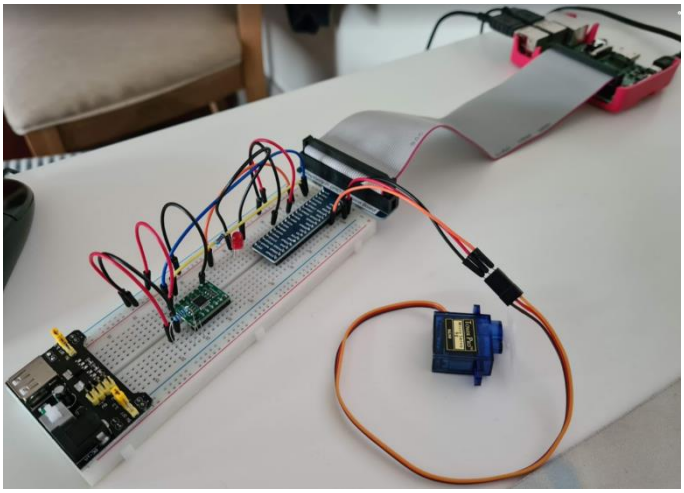

Figure 4: Raspberry Pi

4.2 Software

This year, Android Studio has become one of my favourite software components. I realised that it is truly adapted for all Android devices, that it is open-source software, and that it is supported by a large group of other Android developers.

With Firebase, you can link Android Studio and Python to the same database at the same time. This revelation got me thinking: there's a chance that if I alter the database settings in the Application, the values in my Python code (which runs within the Raspberry Pi) will change as well, allowing the Pi to function.

Connectivity is another significant advantage. Because Firebase is already talking with the App through real-time database, we can guarantee that each customer's Raspberry Pi device will be automatically connected to it.

```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

from multiprocessing import Process

cred = credentials.Certificate('/home/pi/cred.json')

firebase_admin.initialize_app(cred,{
'databaseURL': "https://login-dle09.firebaseio.com/"
})
```

Figure 5: Linking Firebase in Python

```
user = FirebaseAuth.getInstance().getCurrentUser();
reference = FirebaseDatabase.getInstance().getReference( path: "Users");
userID = user.getUid();
```

Figure 6: Linking Firebase in Android Studio

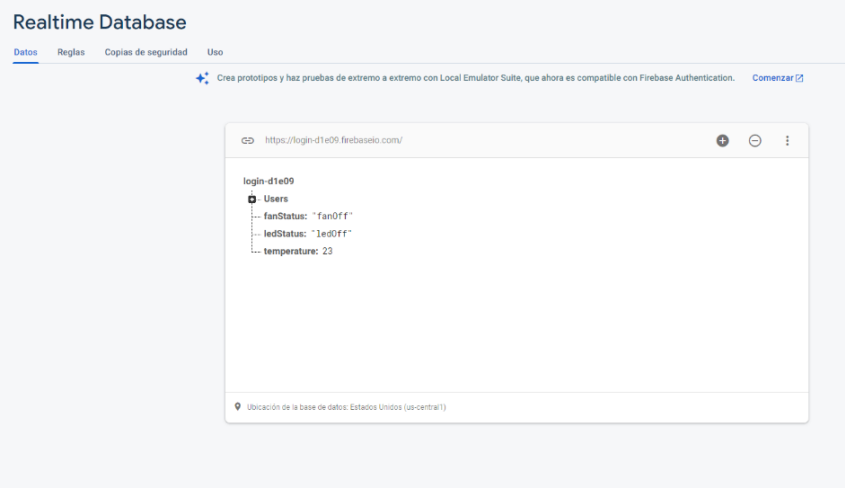

Figure 7: Realtime Database

My Realtime database is made up of four different sources of data: user information, light and fan status, and the temperature of my Raspberry Pi, that is refreshed each second.

4.3 Function of the Light

We have the capacity to switch on or off the lights from our App's Home Page. Whenever we push the buttons, the status of the light in our database is updated, and "ledOn" or "ledOff" is recorded. While the python program on the Raspberry Pi is functioning, it will read the changes from the database and adjust the message accordingly.

```
led_on.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myref = database.getReference( path: "ledStatus");
        myref.setValue("ledOn");
    }
});

led_off.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myref = database.getReference( path: "ledStatus");
        myref.setValue("ledOff");
    }
});
```

Figure 8: Android studio - light

```
def ledLoop():
    while True:
        if ref.get() == 'ledOn':
            GPIO.output(ledPin,True)
            #print('Led is on')
        elif ref.get() == 'ledOff':
            GPIO.output(ledPin,False)
            #print('Led is off')
```

Figure 9: Raspberry Pi - light

4.4 Function of the Fan

The fan and light functions are comparable. The app will update the database, and the Raspberry Pi will be notified of the changes.

```
fan_on.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myref = database.getReference( path: "fanStatus");
        myref.setValue("fanOn");
    }
});

fan_off.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myref = database.getReference( path: "fanStatus");
        myref.setValue("fanOff");
    }
});
```

Figure 10: Android Studio - fan

```
def fanLoop():
    while True:
        if ref2.get() == 'fanOn':
            for angle in range(0, 181, 1):   # make serv
                servoWrite(angle)
                time.sleep(SERVO_DELAY_SEC)
            time.sleep(0.5)
            for angle in range(180, -1, -1): # make serv
                servoWrite(angle)
                time.sleep(SERVO_DELAY_SEC)
            time.sleep(0.5)
            #print('Fan is on')
        elif ref2.get() == 'fanOff':
            p.start(0)
            #print('fan is off')
```

Figure 11: Raspberry Pi - fan

4.5 Function of the Temperature

The temperature feature is more difficult to understand than the other two. The data from the sensor will be sent to the Raspberry Pi. It varies every second and adjusts itself based on the temperature surrounding it. This information is updated in our database, which then transmits it to our app on our home page.

```
def tempLoop():
    while True:
        value = adc.analogRead(0)        # read ADC value A0 pin
        voltage = value / 255.0 * 3.3        # calculate voltage
        Rt = 10 * voltage / (3.3 - voltage)    # calculate resistance value of thermistor
        tempK = 1/(1/(273.15 + 25) + math.log(Rt/10)/3950.0) # calculate temperature (Kelvin)
        tempC = tempK -273.15        # calculate temperature (Celsius)
        print ('ADC Value : %d, Voltage : %.2f, Temperature : %.2f'%(value,voltage,tempC))
        time.sleep(0.01)
        ref3.set(tempC)
```

Figure 12: Raspberry Pi - temperature

```
DatabaseReference myref = FirebaseDatabase.getInstance().getReference();
myref.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        temp = snapshot.child("temperature").getValue().toString();
        tempData.setText(temp);
    }
```

Figure 13: Android Studio - temperature

5. Conclusion

This paper looked at the overall increase in the use of smart devices in this period, as well as the current systems on the industry. Offered a solution in the form of a mobile application that will allow customers on a budget to experience the benefits of enquiring a Smart House. The App delivered a practical advancement of the requirements described in this report.

For future work, this project could develop indefinitely. Because we made possible the connection between Raspberry Pi and Android Studio through the real-time database, any new sensor or new component could be added to the Raspberry. For more automation feeling for customers we can also include push notifications in the future to alert about the energy consumption from the components if we wanted to save money.

5. Bibliography

1. Ashton, K. (2009) *That 'Internet of Things' Thing.* RFID Journal. http://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf Accessed 2 November 2021.

2. Aggarwal, R. and Lal Das, M. (2012) *RFID Security in the Context of "Internet of Things"*. ACM Digital Library Home. http://dx.doi.org/10.1145/2490428.2490435 Accessed 2 November 2021.

3. Patel, R. (2020) *IoT and home automation: What does the future hold?* IOTNOW. https://www.iot-now.com/2020/06/10/98753-iot-home-automation-future-holds/ Accessed 6 November 2021.

4. Anonymous (2020) *Firebase Realtime Database.* Firebase. https://firebase.google.com/docs/database Accessed 18 November 2021.

5. Varguese, E. M. (2017) *Raspberry pi technology.* Yeklo Mar Basilios College. Page 11. https://es.slideshare.net/EldhoMVarghese/raspberry-pi-technology-73011685 Accessed 18 December 2021.

6. Anonymous (2020) *What is a Flowchart?* Lucidchart. https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial Accessed 21 December 2021.