VIETNAM NATIONAL UNIVERSITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
Faculty of Computer Science and Engineering

Multidisciplinary Project (CO3109)

Home Automation

by Android Application-based Remote Control

CC01 - Semester 2021-2022

Instructor:	Assoc. Prof. Dr. Pham Quoc Cuong	
Group Members:	Huynh Phuoc Thien	1952463
	Pham Minh Quang	1952410
	Dinh Huy Thien	1952462
	Doan Viet Tu	1952521
	Tran Quoc Duy	1952214

Ho Chi Minh City, 15th May 2022

Contents

Document history

Date	Ver.	Changes	Author
28/03/2022	1.0	- Create report template, initialize sections.	Doan Viet Tu
28/03/2022	1.1	- Write topic introduction and illustration.	Doan Viet Tu
01/04/2022	1.2	- Add and review devices' description.	Pham Minh Quang
08/04/2022	2.0	- Add use-case scenarios, use-case diagram.	Tran Quoc Duy
09/04/2022	2.1	- Add definition of MQTT, design pattern.	Tran Quoc Duy
16/04/2022	2.2	- Design the deployment view.	Dinh Huy Thien
17/04/2022	2.3	- Edit the deployment view.	Huynh Phuoc Thien
25/04/2022	3.0	- Describe the system requirements.	Doan Viet Tu
30/04/2022	3.1	- Design the activity diagram.	Dinh Huy Thien
30/04/2022	3.2	- Design the database overview.	Huynh Phuoc Thien
01/05/2022	3.3	- Update the finished product part.	Dinh Huy Thien
15/05/2022	4.0	- Review and edit the whole report.	Doan Viet Tu

Work distribution

Stu. ID	Full name	Tasks	Workload
1952463	Huynh Phuoc Thien	- Main UI/backend developer. - Design system diagrams.	100%
1952410	Pham Minh Quang	- Create MQTT between Micro:bit and the Adafruit IO server. - Deploy the Smart Temperature system.	100%
1952462	Dinh Huy Thien	- Main UI/backend developer. - Deploy the Smart Curtain system.	100%
1952521	Doan Viet Tu	- Handle the report, support scripting. - Deploy the Door Security system.	100%
1952214	Tran Quoc Duy	- Support UI/backend and report. - Editing the demo footages.	100%

1 INTRODUCTION

1.1 About the project

1.1.1 Brief

The main objective of this project is to develop a simple and cost efficient **home automation system** with an **Android application which would act as a remote** and make at most use of all the appliances that can be connected which would help to simplify daily life.

Basically, every single appliance in your house can be controlled using a remote device. Android based E-home is an application of Internet of Things system which integrates Android operating system, Adafruit server and the Micro:bit for the implementation of Smart Home. Any Android device can act as a transmitting device. The user can control any appliance through a user friendly mobile application built in Android platform. Also nowadays all the new homes are shifting towards LED lights so by using our designed app and system we can control the intensity of these lights and also apply it in other applications like curtain, fan, motor etc.

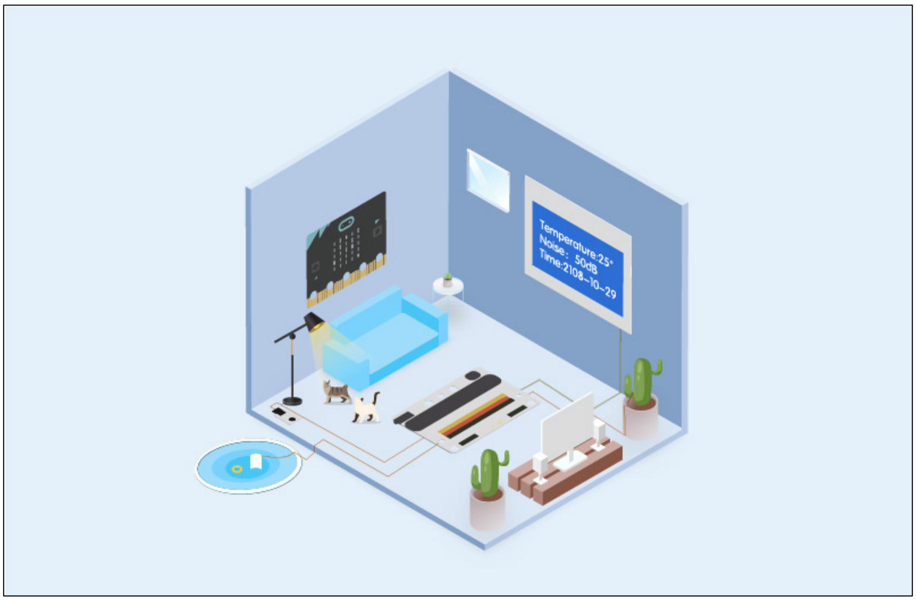


Figure 1: An illustration of our project. Source: www.elecfreaks.com

1.1.2 Expected outcomes

- Know how to design a Home Automation system with an Android application that can be controlled remotely.
- Practice those two aspects: IoT and Mobile application.
- Practice with Adafruit server as well as implement IoT gateway using Python.
- Get used to some methodologies: screen (UI/UX) design, MQTT and design pattern.
- Improve teamwork skills and research skills.

1.1.3 Functionality

1. Entrance Security system.

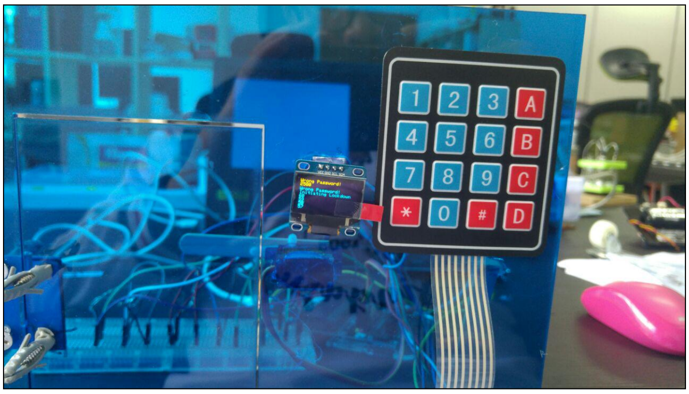


Figure 2: An illustration of this feature. Source: tinkercademy.com

2. Smart Curtain & Lighting system.

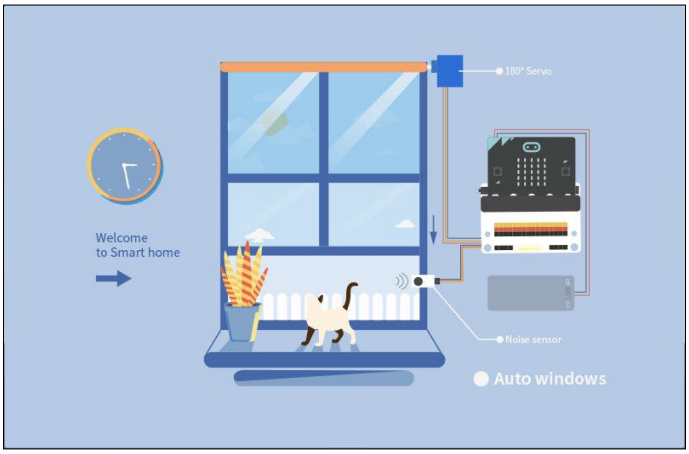


Figure 3: An illustration of this feature. Source: www.elecfreaks.com

3. Smart Temperature system.

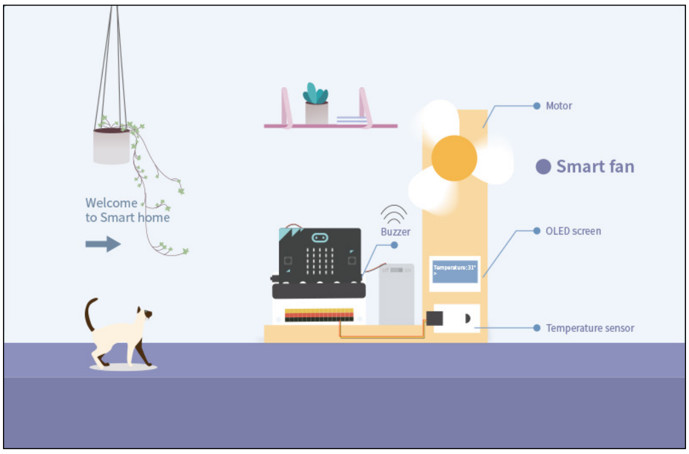


Figure 4: An illustration of this feature. Source: www.elecfreaks.com

1.2 Micro:bit

The Micro Bit (also referred to as BBC Micro Bit, stylized as Micro:bit) is an open source hardware ARM-based embedded system designed by the BBC for use in computer education in the United Kingdom. It was first announced on the launch of BBC's Make It Digital campaign on 12 March 2015 with the intent of delivering 1 million devices to pupils in the UK.

The device is described as half the size of a credit card and has an ARM Cortex-M0 processor, accelerometer and magnetometer sensors, Bluetooth and USB connectivity, a display consisting of 25 LEDs, two programmable buttons, and can be powered by either USB or an external battery pack. The device inputs and outputs are through five ring connectors that form part of a larger 25-pin edge connector. In October 2020, a physically nearly identical v2 board was released that features a Cortex-M4F microcontroller, with more memory and other new features.

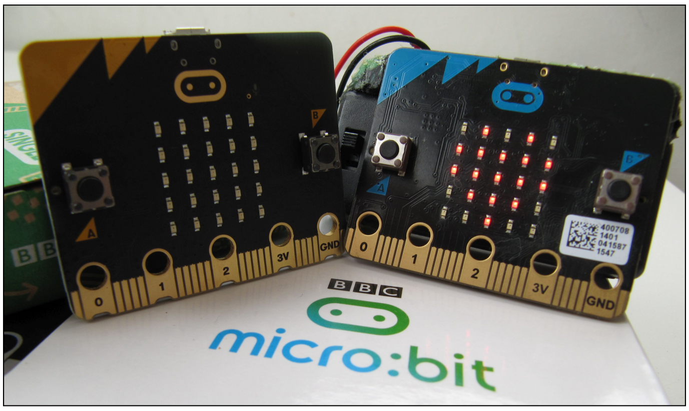


Figure 5: Micro:bit v1 (right) and v2 (left)

1.3 Android

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google. Android applications are usually developed in the Java language using the Android Software Development Kit (SDK). Once developed, Android applications can be packaged easily and sold out either through a store such as Google Play, SlideME, Opera Mobile Store, Mobango, F-droid and the Amazon Appstore.

Why Android?

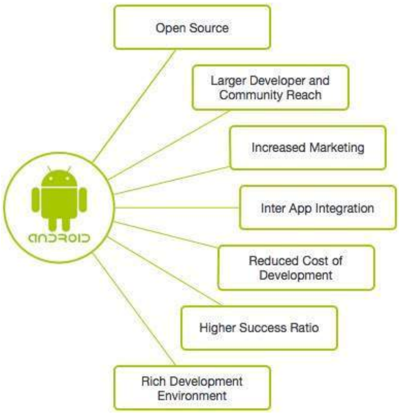

Open Source
Larger Developer and Community Reach
Increased Marketing
Inter App Integration
Reduced Cost of Development
Higher Success Ratio
Rich Development Environment
android

Figure 6: Advantages of Android platform

2 SYSTEM REQUIREMENTS

2.1 Functional

2.1.1 Internet of Things aspect

- Can open the entrance door by typing right password, but trigger the alarm when the input is wrong for more than three times.
- Can automatically close the entrance door when it is opened for a long period of time.
- Can change the password to only-number, only-character or both but not special characters.
- Can detect the sound of breaking windows then trigger the alarm, but only for testing as the real device has a limitation.
- Can measure the temperature and humidity in the room, then display the status in real-time to the LCD.
- Can base on the light of the room and also the temperature to adjust the curtain system.

2.1.2 Mobile application aspect

- Allow user to manually control (to open or close) the entrance door and the curtain system.
- Can change the password of the door.
- Display on the app the current data of temperature and humidity measurements.
- Can compare the data in real-time to trigger some automatic features.
- Can show the system performance and data analysis.

2.2 Non-functional

2.2.1 Internet of Things aspect

- Can handle data from sensors.
- Can avoid system corruption when the data is overloaded.
- The server can work for 24/24.
- The system should be general enough and extendable for applying in other mobile operating systems.

2.2.2 Mobile application aspect

- User shall be able to authenticate via a securely-stored account password.
- The password should be at least 6 characters.
- Connection with server should take under 2 seconds while basic interface interaction should take under 500ms.
- UI must be friendly and also easily operable.
- Support local data storage of up to 100MB as well as on database server.
- Run on Android.

3 DEVICES

3.1 DHT11

- **Application:** Measure temperature and report to the IoT server. Temperature is a vital statistic for the smart house such as controlling or maintain a comfortable temperature for the owners.
- **Input:** Air temperature - the sensor is mounted in the house.
- **Output:** The temperature value, which is displayed on the LCD and on IoT server.

3.2 Light sensor

- **Application:** Measure sunshine intensity and detect daylight. Sunshine intensity also affects the heat. Daylight detection on the other hand dictate whether to close the curtain or not.
- **Input:** Light density, the sensor is placed near the window, exposed to the sunlight.
- **Output:** Return light density level and display on screen.

3.3 Buzzer

- **Application:** To alarm the home owner about intruders when the input password is incorrectly typed for 3 times or when the door is opened for a long period of time.
- **Input:** State 0 or 1, which indicates the buzzer being silent or active, respectively. The input can also be from 0 to 1023, which is the value for the sound level of buzzer.
- **Output:** The buzzer makes sound based on the input conditions.

3.4 LCD I2C

- **Application:** Displaying the temperature and humidity measurements.
- **Input:** Temperature and humidity data from the Micro:bit and its sensors.
- **Output:** The string-type information that the user want to show.

3.5 Push button/Touch button

- **Application:** To replace the numpad and serve the purpose of typing in password (combined with the built-in buttons on Micro:bit).
- **Input:** Press or Not press.
- **Output:** 0 or 1.

3.6 Magnetic switch

- **Application:** To detect opening door.
- **Input:** The door is closed or opened.
- **Output:** 0 or 1.

3.7 Mini motor and propeller as a fan

- **Application:** Reduce the temperature of the room.
- **Input:** Temperature value that goes through certain threshold value that had been set for the temperature.
- **Output:** The temperature is controlled and will be comfortable for the owner.

3.8 Sound sensor

- **Application:** Detect the noise of breaking windows. When the lock switch is active, which means the house owner is away, this sensor will work as an alarm to notify when the burglar tries to break in.
- **Input:** The sound around the window.
- **Output:** The level of the noise, which triggers the buzzer if the level too high.

3.9 RC Servo 590

- **Application:** There will be two servo devices used for this project, one for operating the main door and the other for the windows.
- **Input:** Angle data from MQTT.
- **Output:** Open or close the main door or windows.

3.10 Microbit

- The hub for all sensors. It receives sensor readings through the expansion circuit board and perform preliminary processing (such as unit conversion, threshold and the light).
- Pre-processed data are then sent to the server for logging and broadcasting to all smartphones currently connected to it.

3.11 Real-time clock

- **Application:** Tracking the time of the house and transfer the data to the controller which will help it to define which time to close or open the curtain.
- **Input:** GMT+7 time zone.
- **Output:** Tell the current time to detect if it is day or night.

4 USE-CASE DETAILS

4.1 Use-case scenarios

4.1.1 Door Security system

Use-case Name	Door Security
Actors	User, System, Mobile app.
Description	Use the password to lock the door and a buzzer to alarm unsual cases.
Preconditions	The system is active with Internet connection.
Normal Flow	1. User enters a correct password on the door. 2. System turns off the magnetic switch on the door. 3. System opens the door.
Exceptions	Exception 1: at step 1 1a: The input password is wrong for more than 3 times. Exception 2: at step 3 3a: The door is left open for more than 5 seconds.
Alternative Flows	Alternative 1: at step 1 1a: System turns on the buzzer and sends notification to the mobile app. Alternative 2: at step 3 3a: System turns on the buzzer and sends notification to the mobile app. Then system automatically closes the door.
Postconditions	The door is successfully opened after entering the correct password. The alarm system works fine.

Table 1: Door Security Use-case table

4.1.2 Smart Curtain system

Use-case Name	Smart Curtain based on light
Actors	User, System, Mobile app.
Description	The user can manually open or close the curtain. Otherwise, it will be automatically controlled based on the light density in the room.
Preconditions	The system is active with Internet connection.
Normal Flow	1. User clicks the on/off button on the Mobile app. 2. The system opens or closes the curtain.
Exceptions	Exception 1: at step 1 1a. User chooses to do nothing.
Alternative Flows	Alternative 1: at step 1 1a. The system detects the light density and control the curtain accordingly.
Postconditions	The curtain is successfully controlled both manually and automatically.

Table 2: Smart Curtain Use-case table

4.1.3 Smart Temperature system

Use-case Name	Temperature and Humidity control
Actors	User, System, Mobile app.
Description	Record the data of temperature and humidity in the room.
Preconditions	The system is active with Internet connection.
Normal Flow	1. System collects data from sensors and send to server. 2. The application receives data from server and categorize into illustrations. 3. User clicks the statistics screen on Mobile app. 4. User sees the line chart of temperature and humidity measurements over time.
Exceptions	Exception 1: at step 1 1a. The data is over the limitation.
Alternative Flows	Alternative 1: at step 1 1a: System sends notification to the mobile app.
Postconditions	The line chart is updated automatically with the shape changed over time.

Table 3: Smart Temperature Use-case table

4.2 Use-case diagram

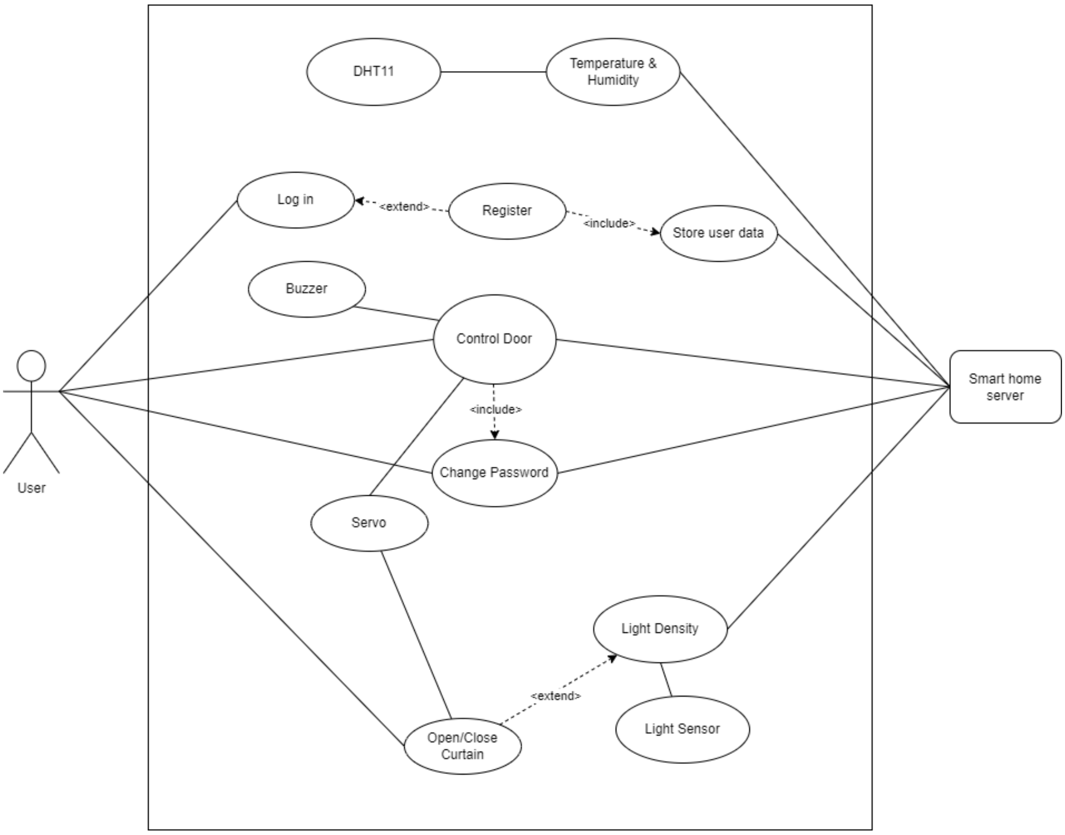


Figure 7: The Use-case diagram of the whole system

5 GENERAL DESIGN

5.1 MQTT protocol

MQTT, or message queue telemetry transport, is a protocol for device communication that Adafruit IO supports. Using a MQTT library or client you can publish and subscribe to a feed to send and receive feed data.

"MQTT is a Client Server publish/subscribe messaging transport protocol. It is light weight, open, simple, and designed so as to be easy to implement. These characteristics make it ideal for use in many situations, including constrained environments such as for communication in Machine to Machine (M2M) and Internet of Things (IoT) contexts where a small code footprint is required and/or network bandwidth is at a premium." - Citation from the official MQTT specification.

In this project, subsystems share the same MQTT and database backends. The MQTT backend takes care of all connections to MQTT servers (concurrently) and abstracts this from the rest of the logic by exposing a simple per-device subscription data streams and publish functions, i.e. the irrigation subsystem upon receiving data from the moisture sensor's data stream (from the backend) will act by calling the publish function of the backend, specifying which device and what data to send.

5.2 Design pattern

In this project, we choose the MVC pattern, in which MVC stands for **Model - View - Controller**, as the backbone architecture for the mobile application.

- **Model:**
  The Model component represents all the data-related logic that the user engages with. This can be the data being transmitted between the View and Controller components or any other data related to business logic.

- **View:**
  The View component is used to handle all of the application's UI logic.

- **Controller:**
  Controllers act as a link between the Model and View components, processing all business logic and incoming requests, manipulating data using the Model, and interacting with Views to render the final output.

6 TECHINICAL VIEW

6.1 Deployment view

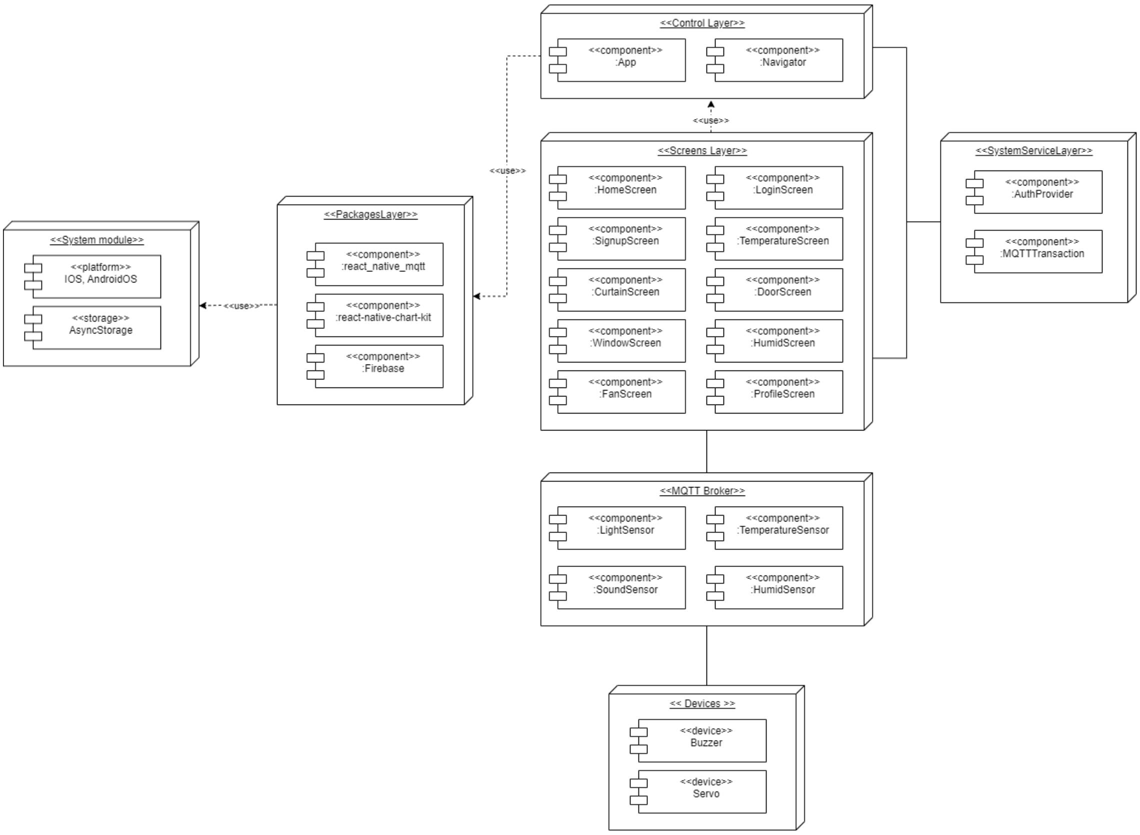

<<Control Layer>>
<<component>> :App
<<component>> :Navigator
<<use>>
<<use>>
<<Screens Layer>>
<<component>> :HomeScreen
<<component>> :LoginScreen
<<component>> :SignupScreen
<<component>> :TemperatureScreen
<<component>> :CurtainScreen
<<component>> :DoorScreen
<<component>> :WindowScreen
<<component>> :HumidScreen
<<component>> :FanScreen
<<component>> :ProfileScreen
<<SystemServiceLayer>>
<<component>> :AuthProvider
<<component>> :MQTTTransaction
<<PackagesLayer>>
<<component>> :react_native_mqtt
<<component>> :react-native-chart-kit
<<component>> :Firebase
<<use>>
<<System module>>
<<platform>> IOS, AndroidOS
<<storage>> AsyncStorage
<<MQTT Broker>>
<<component>> :LightSensor
<<component>> :TemperatureSensor
<<component>> :SoundSensor
<<component>> :HumidSensor
<< Devices >>
<<device>> Buzzer
<<device>> Servo

Figure 8: Deployment view of the whole system

6.2 Implementation view

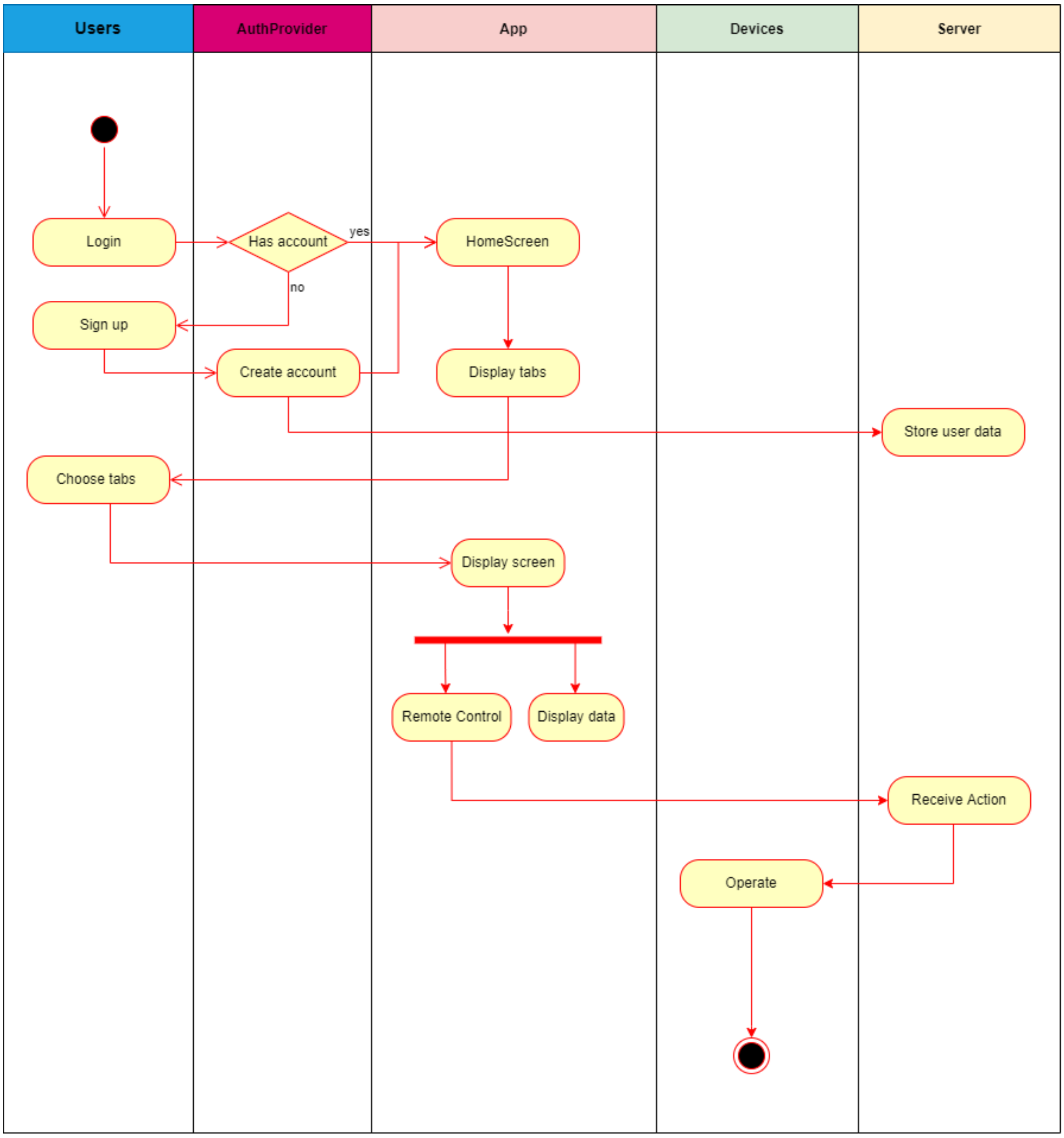


Figure 9: Activity diagram of the whole system

6.3 Database

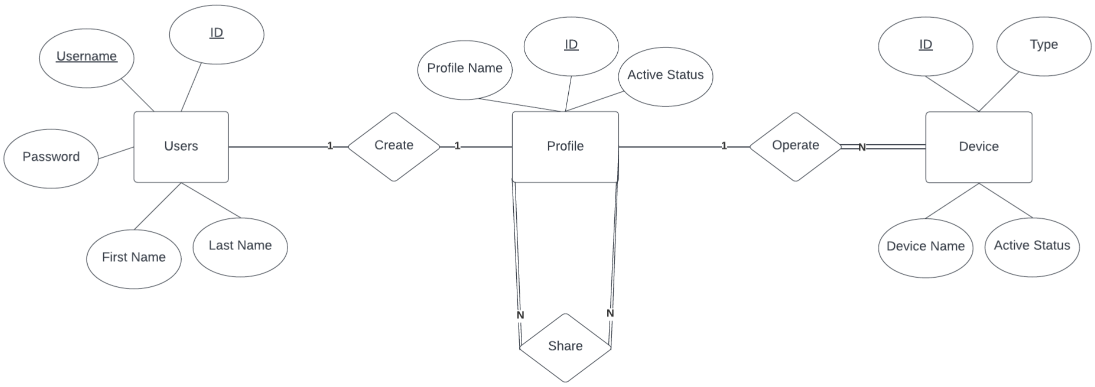
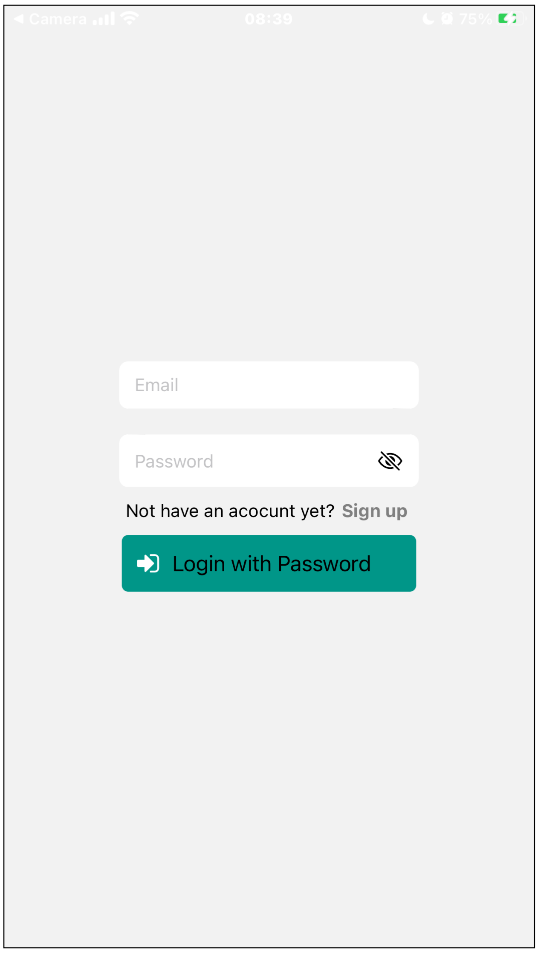


Figure 10: Database overview of the whole system

7 FINISHED PRODUCT

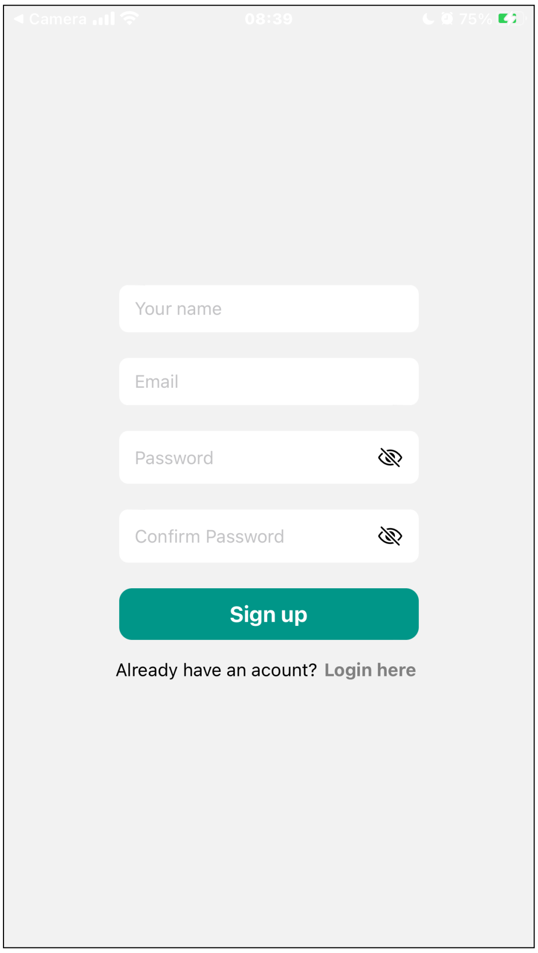
7.1 Authentication

Before getting access to our application, the user need to authenticate their identity. Our application also provides security to the users' data. When a new profile is created, the information will only be stored at our database for functioning. Also, note that the length of the password should be at least 6 characters. After successfully logging in, the user will be directed to the home page.



Not have an acocunt yet? Sign up

(a) Log-in Screen

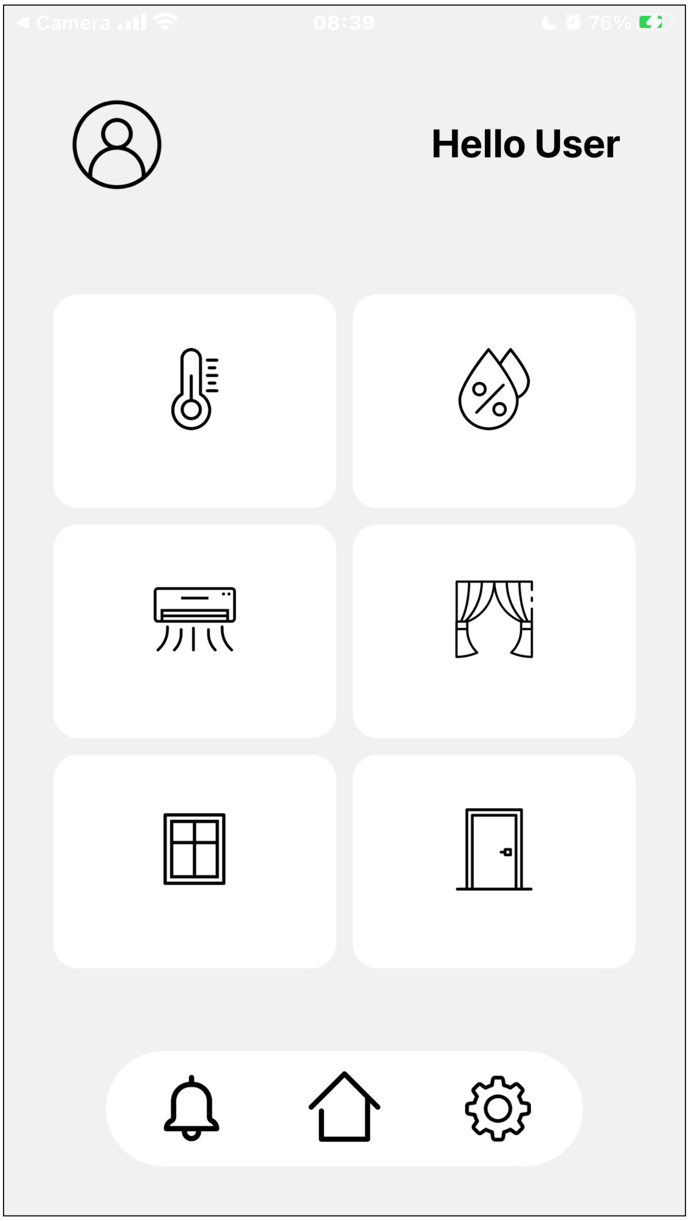


Already have an acount? Login here

(b) Sign-up Screen

7.2 Home page

In this screen, the application provides the user a wide range of choices. There are six main features of our application. The first two tabs are used for data analysis, they provide the illustration of the data. The others are used to control the devices in the house.

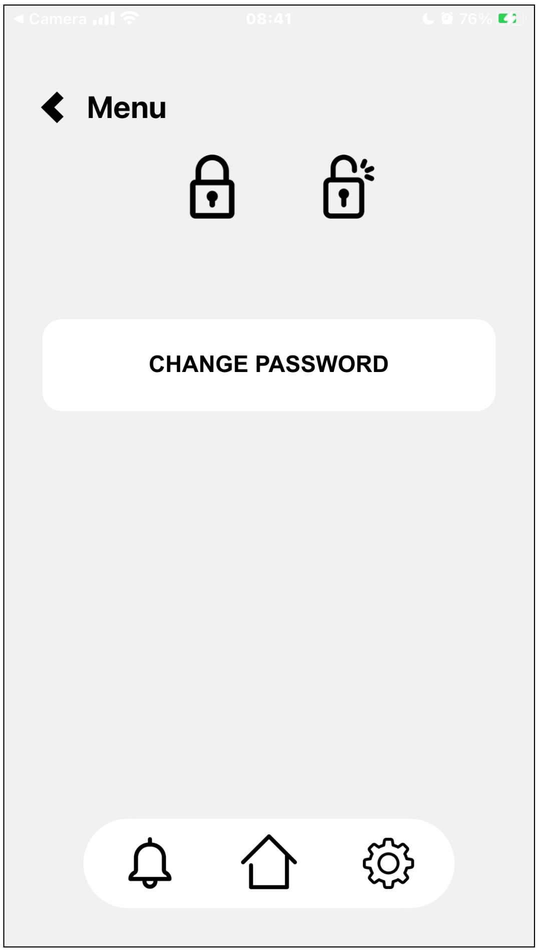


(c) Home page

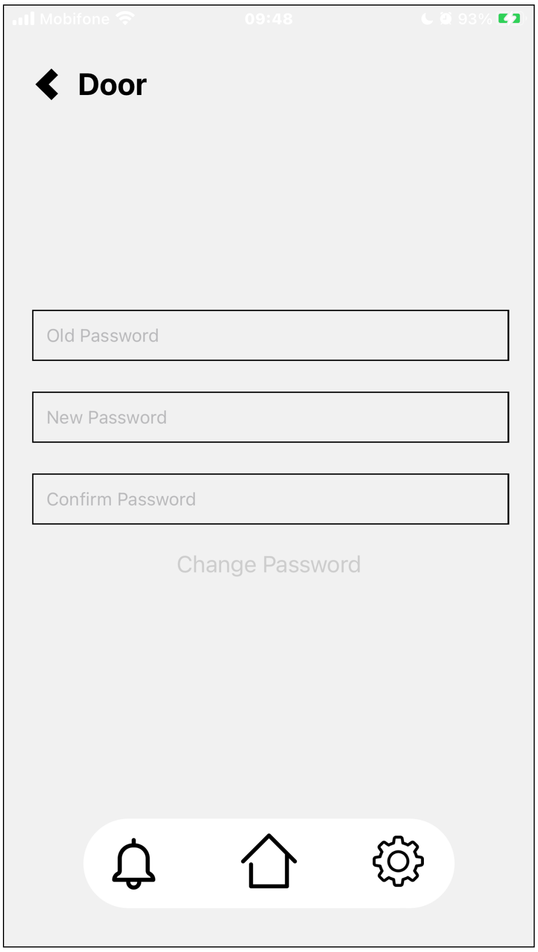
7.3 Features

7.3.1 Door control and passwords

For this feature, the user can open or close the entrance door using the two buttons on screen (a). Moreover, the password of the door can be changed at any time for the demand of security.



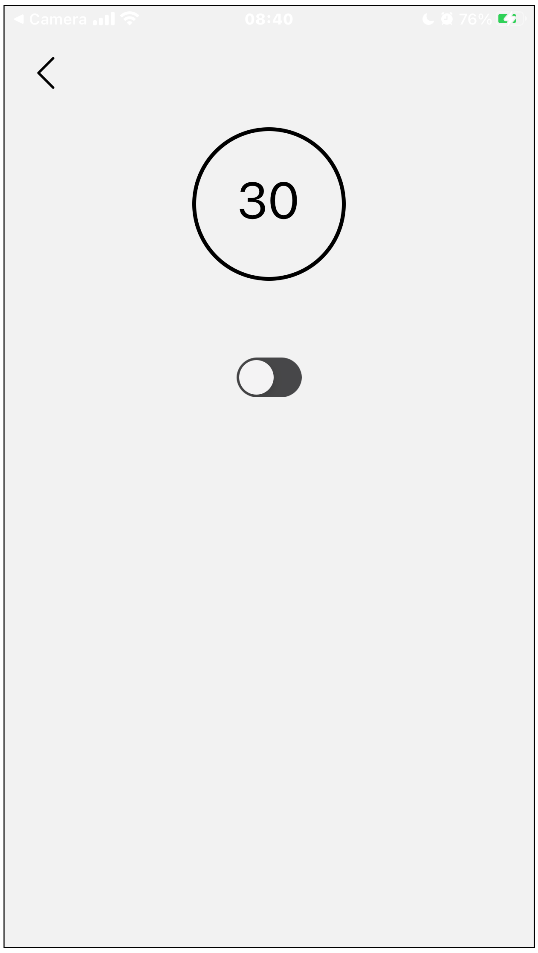
(d) Open or close the door



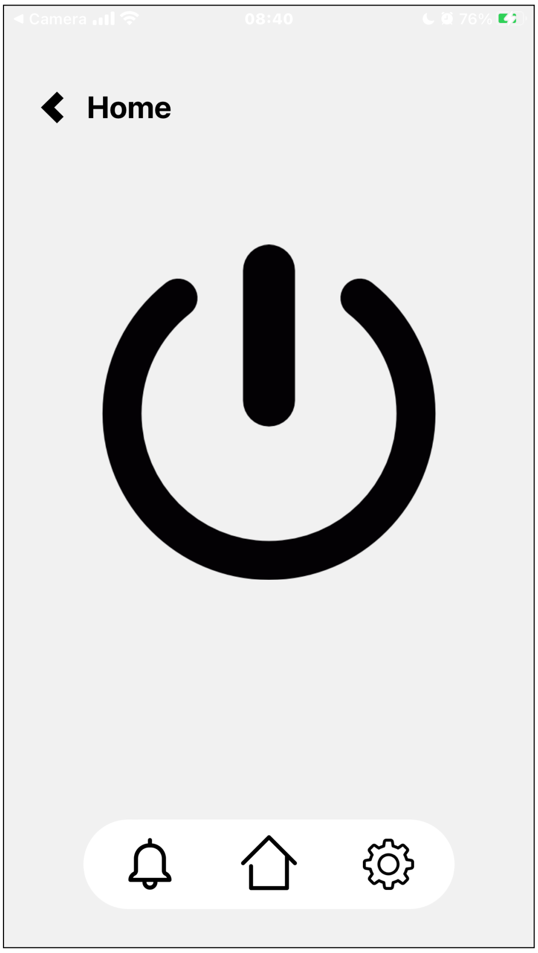
(e) Password settings

7.3.2 Curtain control based on light

For this feature, the user can close the curtain manually. Otherwise, the curtain will be closed automatically based on the light density of the room. Whenever the sensor records the change in the density of the environment, it sends a message to the application through MQTT broker. Thus, the app compares the data with the default density to decide the next action.



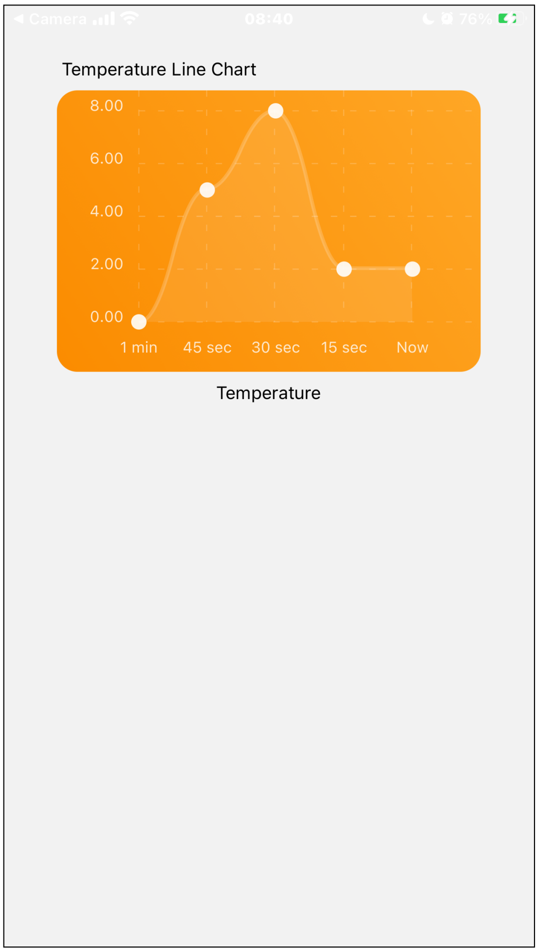
(f) Light density



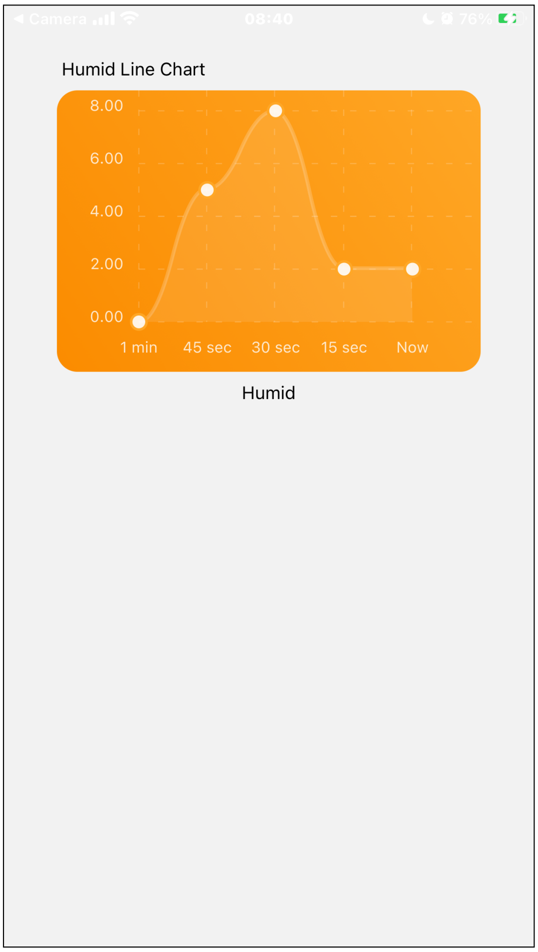
(g) Close the curtain manually

7.3.3 Statistics screen

The statistics screen can be reached by the user. Then the user can see the data of the temperature and the humidity over time. The application receives the data from the Adafruit IO server. Whenever the server invokes new data, it will send a message to the application and change the shape of the chart. About the chart, the five latest changes will be illustrated, hence, the older data will be eliminated.



(h) Temperature measurement



(i) Humidity measurement

8 DEMO

- **Final demo:** Home Automation with Android remote control | Youtube
- **Source:** GitHub

References

[1] https://www.cytrontech.vn/c-micro-bit-kits/

[2] https://io.adafruit.com/api/docs/mqtt.html#adafruit-io-mqtt-api

[3] https://reactnative.dev/docs/getting-started

[4] https://reactnavigation.org/docs/getting-started/

[5] https://firebase.google.com/docs/auth

[6] https://viblo.asia/p/firebase-login-and-registration-authentication-aWj53VWol6m

[7] https://www.toptal.com/firebase/role-based-firebase-authentication

[8] https://firebase.google.com/docs/firestore

[9] https://rnfirebase.io/firestore/usage

[10] Michael Burton (2015). Android App Development For Dummies. Apress.

[11] Akshat Paul, Abhishek Nalwaya (2019). React Native for Mobile Development: Harness the Power of React Native to Create Stunning iOS and Android Applications. Apress.

[12] Adam D. Scott (2020). JavaScript Everywhere: Building Cross-Platform Applications with GraphQL, React, React Native, and Electron. O'Reilly Media.

[13] Horton J. (2015). Android Programming for Beginners: Learn all the Java and Android skills you need to start making powerful mobile applications. PACKT Publishing.