

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



THỰC TẬP ĐỒ ÁN ĐA NGÀNH

Mobile App

WAREHOUSE MANAGEMENT

TA:	Bùi Xuân Giang
SV:	Hồ Công Sơn 1712964
	Tô Phú Quý 1712892
	Phan Tân Quốc 1712855
	Hoàng Văn Sang 1712928
	Nguyễn Văn Sỹ 1712996

TP. HỒ CHÍ MINH, THÁNG 7/2020



Mục lục

1 Giới thiệu đề tài	2
2 Phân công công việc	3
3 Phân tích chức năng	5
3.1 Xem thông tin môi trường kho	5
3.2 Thiết lập thông số môi trường cơ bản của kho và tự động điều khiển thiết bị đầu ra	6
3.3 Các chức năng mở rộng	7
4 Tìm hiểu công nghệ	8
4.1 Front-end và Back-end: Java	8
4.2 Database: Firebase	8
4.3 Mô hình IOT	9
5 Thiết kế hệ thống	10
5.1 Người dùng	10
5.2 Ứng dụng di động	10
5.3 Kho nông sản	11
5.4 Server	11
5.5 Database	11
6 Mô tả chi tiết	12
6.1 Ứng dụng	12
6.1.1 Đăng nhập và đăng ký	12
6.1.2 Trang chủ	15
6.1.3 Xem và thiết lập môi trường kho	16
6.1.4 Xem lịch sử	18
6.1.5 Xem thông số gợi ý các loại nông sản	20
6.1.6 Xem thời tiết	21
6.1.7 Tuỳ chỉnh tài khoản	22
6.2 Server, input và output	24
6.2.1 Kết nối server	24
6.2.2 Input	28
6.2.3 Output	28
7 Tổng kết	29
7.1 Kết quả	29
7.1.1 Hạn chế	29
8 Hướng phát triển	30

1 Giới thiệu đề tài

Bảo quản nông sản tươi sau thu hoạch ở nước ta nói riêng và trên thế giới nói chung ngày càng là vấn đề được quan tâm hàng đầu. Đây được xem là công đoạn thiết yếu nhằm bảo đảm chất lượng đầu ra của sản phẩm cùng những vấn đề liên quan. Da số các loại nông sản như rau quả là loại nông sản khó bảo quản vì lượng nước trong rau quả cao, có thể chiếm 85%-95% khối lượng. Do đó loại thực phẩm này rất dễ bị hỏng trong quá trình bảo quản. Theo thống kê từ VCCI, mức độ tổn thất với rau quả và trái cây có thể lên tới 45%.

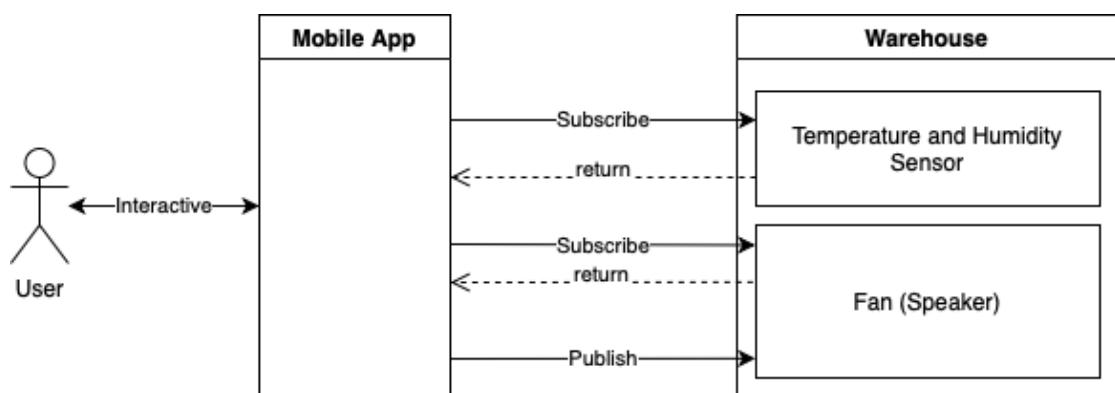
Vì nhu cầu nông sản cần được bảo quản tốt hơn và ít tốn chi phí cũng như thời gian nhất. Nên nhóm chúng em quyết định lên ý tưởng cho một ứng dụng IoT điều khiển kho nông sản. Đây sẽ là một kho nông sản thông minh được giám sát thông qua các cảm biến khác nhau được kết nối với mạng internet. Dữ liệu từ các cảm biến truyền về sẽ là dữ liệu thời gian thực. Ngoài việc quan sát ra, ta còn có thể điều khiển từ xa các thiết bị đầu ra tùy thuộc vào điều kiện hiện tại. Không những thế ứng dụng còn có nhiều tính năng mở rộng tối ưu hiệu suất sử dụng.

Cụ thể, trong dự án này nhóm chúng em mô phỏng một kho nông sản thực tế.

Thiết bị input của kho gồm 2 cảm biến là cảm biến nhiệt độ không khí và cảm biến độ ẩm không khí (Thiết bị demo là TempHumi). Thiết bị output là một động cơ quạt (Thiết bị demo là Speaker).

Người dùng sẽ tương tác với một ứng dụng di động đã kết nối với các thiết bị trong kho thông qua internet. Ứng dụng giúp người dùng xem thông số các thiết bị input theo thời gian thực và chủ động điều khiển thiết bị output theo cơ chế định sẵn hoặc theo cách thủ công.

Cơ chế tổng quát được trình bày trong hình dưới đây:



Hình 1: Cơ chế tổng quát



2 Phân công công việc

Tuần 1:

- Tìm nhóm và họp nhóm.
- Suy nghĩ đề tài.

Tuần 2:

- Thông nhất đề tài.
- Thông nhất môi trường phát triển và chọn thiết bị.
- Lên ý tưởng cho dự án.

Tuần 3:

Tên	MSSV	Công việc
Hồ Công Sơn	1712964	Viết UseCase "Xem thông tin thời tiết khu vực"
Tô Phú Quý	1712892	Viết UseCase "Xem thông tin dữ liệu môi trường trong kho đo được từ các cảm biến", viết báo cáo tuần
Phan Tân Quốc	1712855	Viết UseCase "Điều khiển hệ thống thiết bị Output"
Hoàng Văn Sang	1712928	Viết UseCase "Thông kê các thông số của kho hàng"
Nguyễn Văn Sỹ	1712996	Viết UseCase "Thiết lập các thông số cơ bản môi trường theo điều kiện cần của từng loại nông sản"

Tuần 4:

Tên	MSSV	Công việc
Hồ Công Sơn	1712964	Thiết kế giao diện xem thông tin thời tiết khu vực
Tô Phú Quý	1712892	Thiết kế giao diện login, trang chủ, xem thông tin dữ liệu môi trường kho. Viết báo cáo tuần
Phan Tân Quốc	1712855	Thiết kế giao diện điều khiển hệ thống thiết bị output
Hoàng Văn Sang	1712928	Thiết kế giao diện Thông kê các thông số của kho hàng
Nguyễn Văn Sỹ	1712996	Thiết kế Giao diện thiết lập các thông số cơ bản môi trường theo điều kiện cần của từng loại nông sản



Tuần 5 - 7:

Tên	MSSV	Công việc
Hồ Công Sơn	1712964	Nghiên cứu và thực hiện kết nối server, thực hiện subscribe và publish
Tô Phú Quý	1712892	Nghiên cứu xây dựng ứng dụng, thiết kế giao diện
Phan Tân Quốc	1712855	Nghiên cứu và thực hiện kết nối server, thực hiện subscribe và publish
Hoàng Văn Sang	1712928	Nghiên cứu xây dựng server
Nguyễn Văn Sỹ	1712996	Nghiên cứu xây dựng ứng dụng, thiết kế giao diện

Tuần 8:

- Demo giữa kỳ.

Tuần 9 - 10:

Tên	MSSV	Công việc
Hồ Công Sơn	1712964	Xây dựng backend cho ứng dụng
Tô Phú Quý	1712892	Xây dựng, thiết kế UX/UI cho ứng dụng
Phan Tân Quốc	1712855	Xây dựng, thiết kế UX/UI cho ứng dụng
Hoàng Văn Sang	1712928	Xây dựng thiết kế database
Nguyễn Văn Sỹ	1712996	Xây dựng, thiết kế UX/UI cho ứng dụng

Tuần 11:

- Tổng kết dự án.
- Viết báo cáo.

Tuần 12:

- Báo cáo tổng kết.

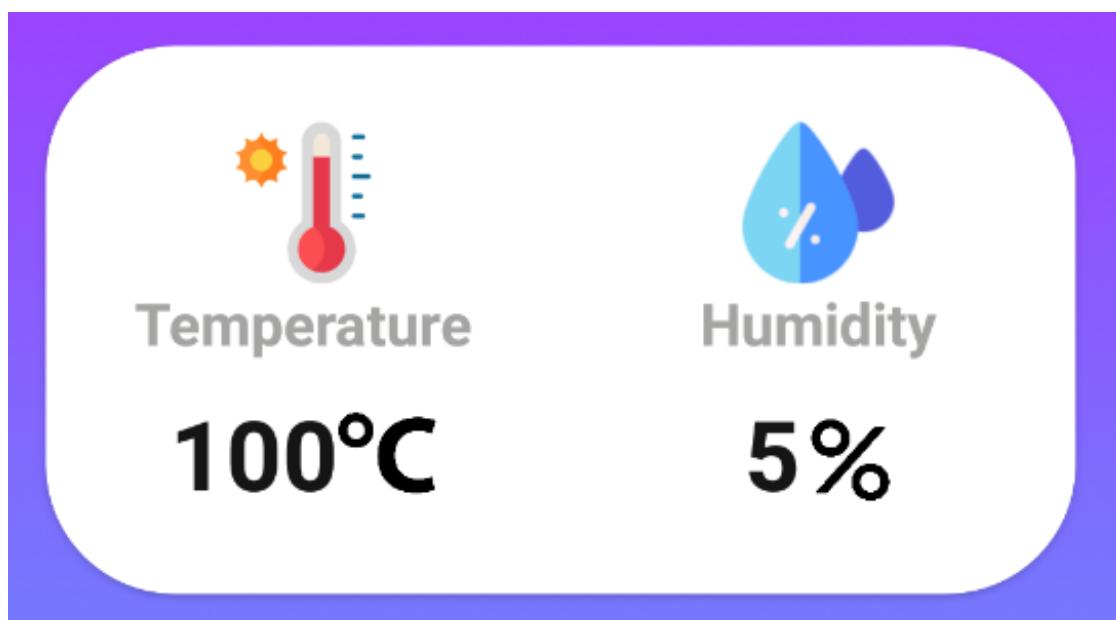
3 Phân tích chức năng

3.1 Xem thông tin môi trường kho

Với đề tài quản lý kho bảo quản nông sản, nhóm lựa chọn thiết bị in put là cảm biến nhiệt độ và độ ẩm không khí (TempHumi).

Thiết bị input này gồm 2 thông số là nhiệt độ và độ ẩm.

Ứng dụng sẽ lấy thông số từ cảm biến theo thời gian thực và hiển thị ra màn hình trong tính năng môi trường kho như hình dưới:



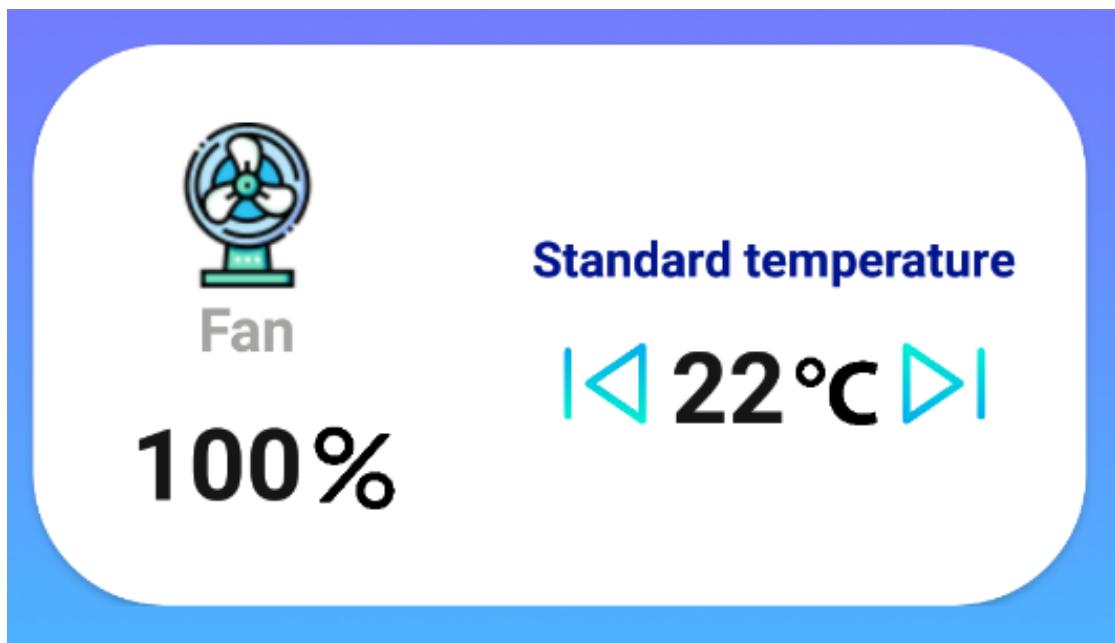
Hình 2: Nhiệt độ và độ ẩm được hiển thị ra màn hình ứng dụng

3.2 Thiết lập thông số môi trường cơ bản của kho và tự động điều khiển thiết bị đầu ra

Tương tự với thiết bị input, ứng dụng cũng sẽ lấy thông số từ cảm biến theo thời gian thực và hiển thị ra màn hình trong tính năng môi trường kho.

Ngoài ra ứng dụng còn có thể điều khiển thiết bị output theo cơ chế:

- Đầu tiên, người dùng chọn nhiệt độ tiêu chuẩn cho kho (Nhiệt độ tuỳ chỉnh bởi người dùng theo từng loại nông sản của kho đó và theo từng mùa trong năm). Nhiệt độ tiêu chuẩn sẽ giới hạn từ 0 đến 29 độ.
- Hệ thống sẽ so sánh nhiệt độ thực tế với nhiệt độ tiêu chuẩn. Nếu nhiệt độ thực tế cao hơn nhiệt độ tiêu chuẩn, quạt sẽ chạy. Nhiệt độ càng cao hơn nhiệt độ tiêu chuẩn thì quạt sẽ chạy càng mạnh (Cơ chế tự động này chúng ta sẽ cùng tìm hiểu ở phần Mô tả chi tiết).

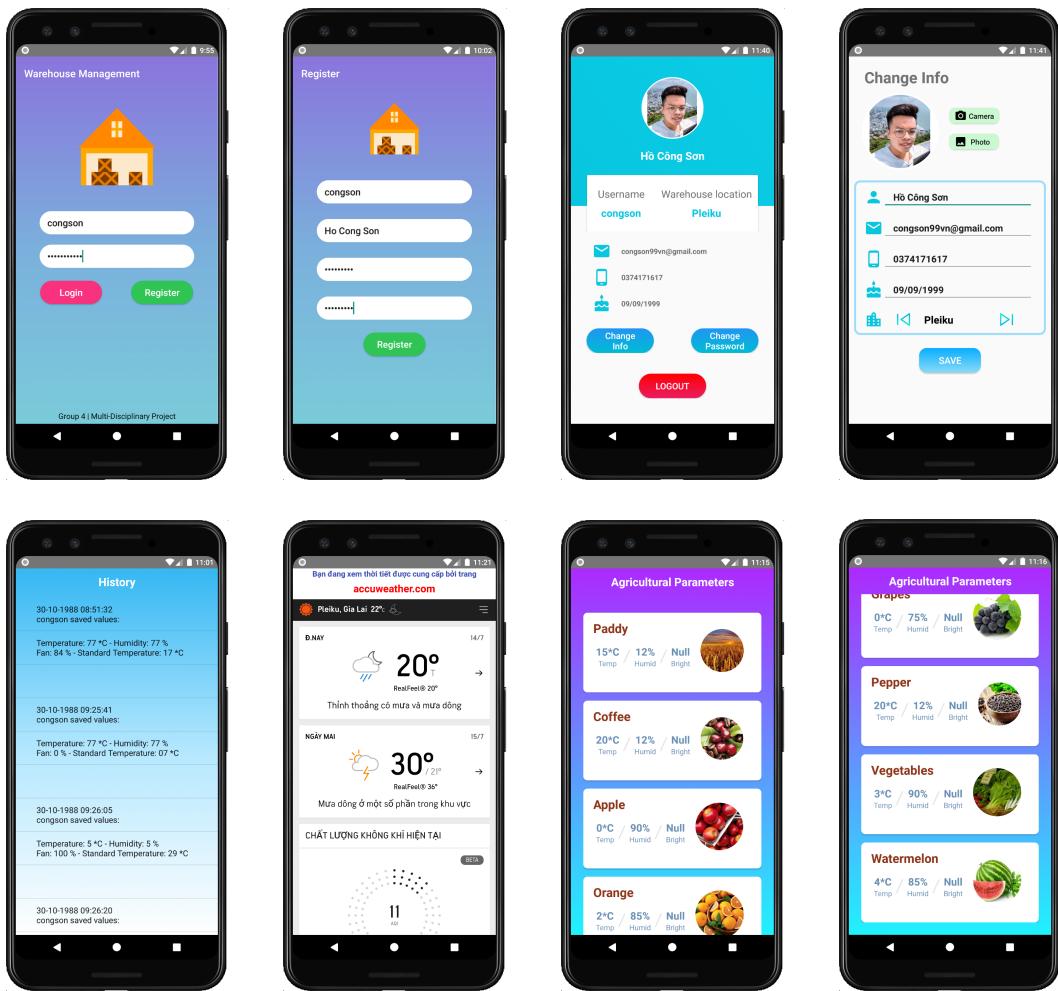


Hình 3: Công suất quạt và nhiệt độ tiêu chuẩn được hiển thị ra màn hình ứng dụng

3.3 Các chức năng mở rộng

Ngoài ra, ứng dụng còn có các tính năng mở rộng:

- Login, logout, tạo và tùy chỉnh tài khoản.
- Lưu lịch sử môi trường.
- Xem thời tiết.
- Xem thông số bảo quản tiêu chuẩn của từng loại nông sản.



Hình 4: Các chức năng mở rộng

Chúng ta sẽ tìm hiểu chi tiết các chức năng này ở phần Mô tả chi tiết.



4 Tìm hiểu công nghệ

4.1 Front-end và Back-end: Java

Java là một ngôn ngữ lập trình hướng đối tượng phổ biến và được sử dụng rộng rãi trong nhiều lĩnh vực.

Nói đến Java, thì không thể không nói đến sự dễ dàng trong việc tìm hiểu cũng như trong việc xây dựng một ứng dụng cho thiết bị di động. Khi mà độ phổ biến của nó là gần như được chạy trên toàn bộ các thiết bị smartphone hiện nay.

Java cung cấp cho ta một kho tàng tài liệu phong phú cũng như được trang bị vô số các API mã nguồn mở tiện lợi cho việc lập trình. Ngoài ra Java được đánh giá cao về mặt hiệu suất cũng như là bảo mật.

Nhóm em quyết định chọn Java là vì độ phổ biến và dễ dàng lập trình trong ứng dụng Android. Cộng thêm việc sử dụng IDE là Android Studio, việc tạo ra sản phẩm là vô cùng dễ dàng và tiện lợi.

4.2 Database: Firebase

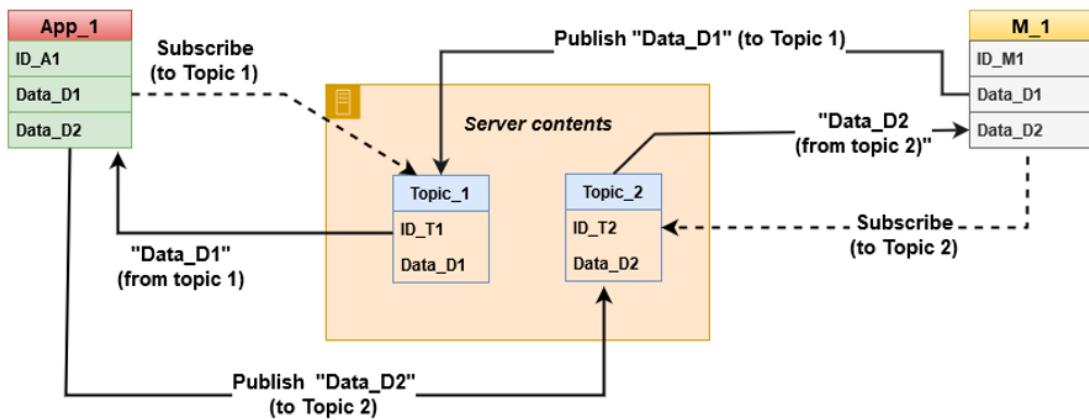
Firebase là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây được Google cung cấp cho các ứng dụng mobile. Nhằm đơn giản hóa các thao tác với cơ sở dữ liệu thông qua các API đơn giản.

Nó cung cấp cho bạn các chức năng như phân tích, báo cáo, ... để bạn có thể dễ dàng nhanh chóng phát triển các ứng dụng mà không cần tốn nhiều thời gian, nhân lực trong việc quản lý hệ thống và cơ sở dữ liệu.

Uy tín chất lượng là không cần bàn cãi do được hỗ trợ và đảm bảo bởi Google cùng với quy mô sử dụng từ hàng triệu người dùng và các tập đoàn lớn trên khắp thế giới. Thêm một điểm cộng cho Firebase nữa là nhờ giao diện website đơn giản, cộng thêm khả năng ổn định khi sử dụng.

Nhóm em quyết định sử dụng Firebase cho đồ án này là vì sự đơn giản trong các thao tác đối với cơ sở dữ liệu.

4.3 Mô hình IOT



Hình 5: Sơ đồ triển khai mô hình IoT

Mô tả tổng quan

Server contents: dùng để truyền các thông điệp (messages) giữa nhiều Client (Publisher và Subscriber). Trong đó nó là điểm giao của tất cả các kết nối đến từ client. Nhiệm vụ của Server Contents là nhận message từ publisher và xếp các message theo hàng đợi rồi chuyển chúng tới một địa chỉ cụ thể.

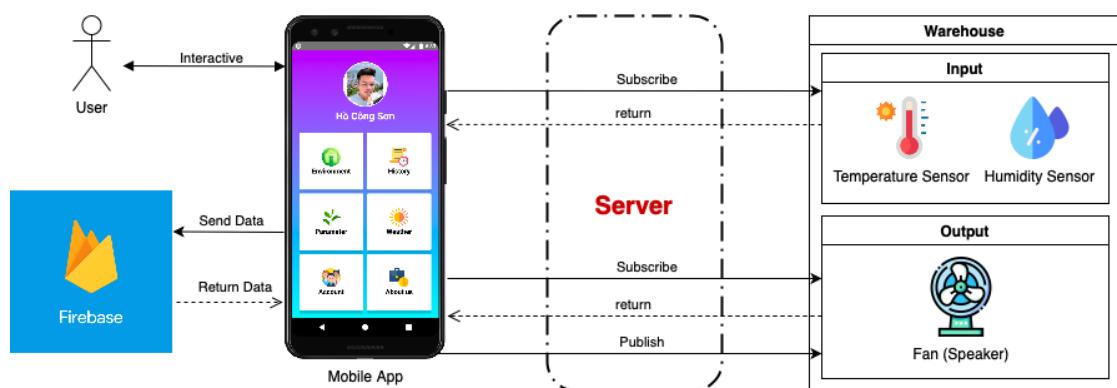
Publisher và Subscriber làm một trong 2 việc là publish các message lên một topic cụ thể hoặc subscribe một topic nào đó để nhận message từ topic này.

5 Thiết kế hệ thống

Hệ thống dự án gồm có 5 thành phần sau:

- Người dùng
- Ứng dụng
- Server
- Kho nông sản
- Database (Firebase)

Các thành phần sẽ kết hợp với nhau như hình dưới đây để tạo thành một hệ thống hoàn chỉnh.



Hình 6: Sơ đồ tổng quát hệ thống

5.1 Người dùng

Người dùng đóng vai trò quan trọng trong hệ thống, là nhân tố đầu cuối mà hệ thống phục vụ, theo dõi và điều khiển toàn bộ hệ thống thông qua việc tương tác với ứng dụng.

5.2 Ứng dụng di động

Đây là thành phần chủ chốt của hệ thống, với các vai trò:

- Tương tác với người dùng, show các thông số và nhận lệnh từ người dùng.
- Thực hiện subscribe đến các thiết bị để lấy thông số (Thông qua server).
- Xử lý số liệu cũng như nhận lệnh để publish đến thiết bị output (Thông qua server).
- Lấy dữ liệu từ Database về xử lý và lưu dữ liệu mới lên Database.



5.3 Kho nông sản

Đây là thành phần thực hiện chức năng chính của hệ thống, đó chính là bảo quản nông sản.

Trong mỗi kho có các cảm biến nhiệt độ và ánh sáng thu nhận thông tin để gửi về ứng dụng thông qua server.

Ngoài ra còn có động cơ quạt thực hiện chức năng của nó khi có yêu cầu từ ứng dụng thông qua server.

5.4 Server

Đây chính là thành phần trung gian đảm bảo việc gửi và nhận giữa ứng dụng và các thiết bị nhà kho.

5.5 Database

Database nhóm sử dụng là Firebase.

Database lưu trữ dữ liệu từ ứng dụng gửi lên và trả về khi có yêu cầu.

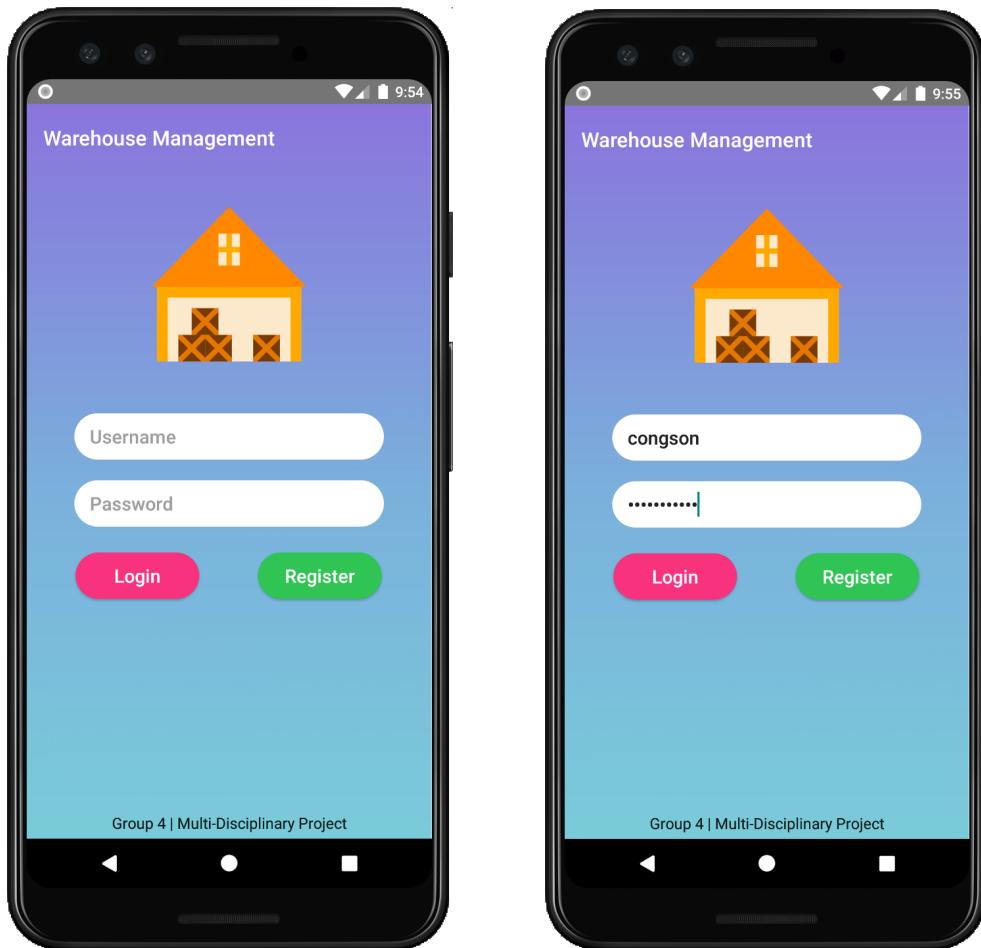
6 Mô tả chi tiết

6.1 Ứng dụng

6.1.1 Đăng nhập và đăng ký

Ngay khi mở ứng dụng, ta sẽ được đưa tới màn hình đăng nhập. Nếu đã có tài khoản, ta sẽ nhập tài khoản và nhấn đăng nhập.

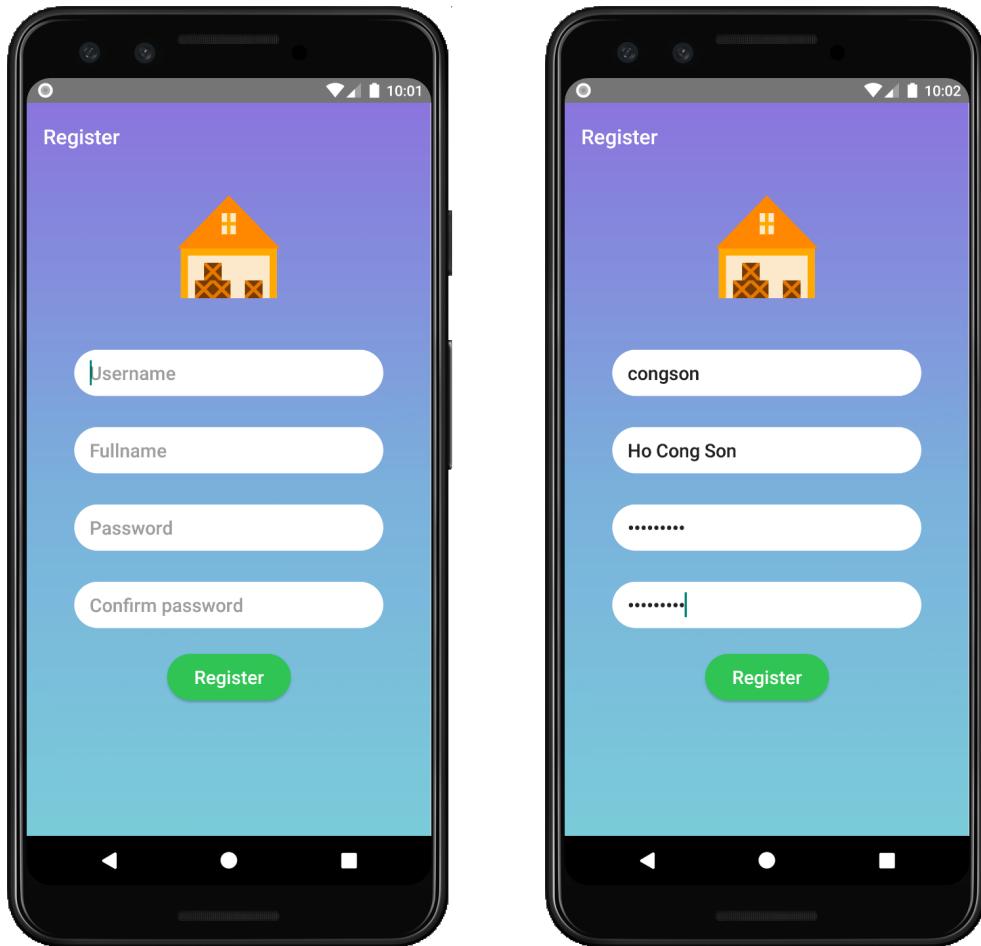
Ứng dụng sẽ kiểm tra xem ta đã nhập đúng yêu cầu chưa. Nếu đúng ứng dụng sẽ truy cập vào Database và kiểm tra xem tài khoản và mật khẩu có đúng hay không. Nếu đúng, đăng nhập thành công. Nếu sai, ứng dụng sẽ báo lỗi cụ thể để người dùng kiểm tra.



Hình 7: Giao diện đăng nhập



Nếu chưa có tài khoản, ta sẽ bấm nút đăng kí và ứng dụng sẽ chuyển đến màn hình đăng kí.



Hình 8: Giao diện đăng kí

Sau khi được đưa đến màn hình đăng kí, ta thực hiện nhập các thông tin và nhấn đăng kí.

Ứng dụng sẽ kiểm tra xem ta đã nhập đúng yêu cầu hay chưa. Nếu đúng ứng dụng sẽ truy cập Database và kiểm tra xem Username yêu cầu đăng kí đã tồn tại hay chưa. Nếu đã tồn tại, ứng dụng sẽ báo lỗi và yêu cầu người dùng chọn Username khác. Nếu chưa tồn tại ứng dụng sẽ tạo tài khoản với thông tin tương ứng và lưu vào Database.



Database tài khoản người dùng sẽ có dạng như sau:

```
Account
  +-- congson
    |   +-- Avatar: "iVBORw0KGgoAAAANSUhEUgABC8AAQCCAIAAAAD4UWCAAA..."
    |   +-- DOB: "09/09/1999"
    |   +-- Email: "congson99vn@gmail.com"
    |   +-- Name: "Hồ Công Sơn"
    |   +-- Pass: "Son123"
    |   +-- Phone: "0374171617"
  +-- test
    +-- Avatar: "None"
    +-- DOB: "DOB"
    +-- Email: "Email"
    +-- Name: "Tesst"
    +-- Pass: "abc"
    +-- Phone: "Phone Number"
```

Hình 9: Database tài khoản người dùng

Khi vừa tạo tài khoản, tài khoản sẽ được lưu vào nhánh có key là username và 2 giá trị là Pass và Name được điền khi đăng ký, các giá trị còn lại sẽ được khởi tạo giá trị mặc định, người dùng sẽ bổ sung ở phần thay đổi thông tin tài khoản.

Ở hình trên, test là tài khoản vừa được tạo với các thông số mặc định còn congson là tài khoản đã được thêm các giá trị và avatar (hình ảnh đại diện sẽ được ứng dụng đưa về dạng byte và lưu vào Database).



6.1.2 Trang chủ

Sau khi đăng nhập thành công, người dùng sẽ được đưa đến giao diện trang chủ, tại màn hình này người dùng sẽ lựa chọn đi tới tính năng mong muốn.

Trang chủ sẽ hiển thị Avatar, tên người dùng và đầy đủ các chức năng của ứng dụng.



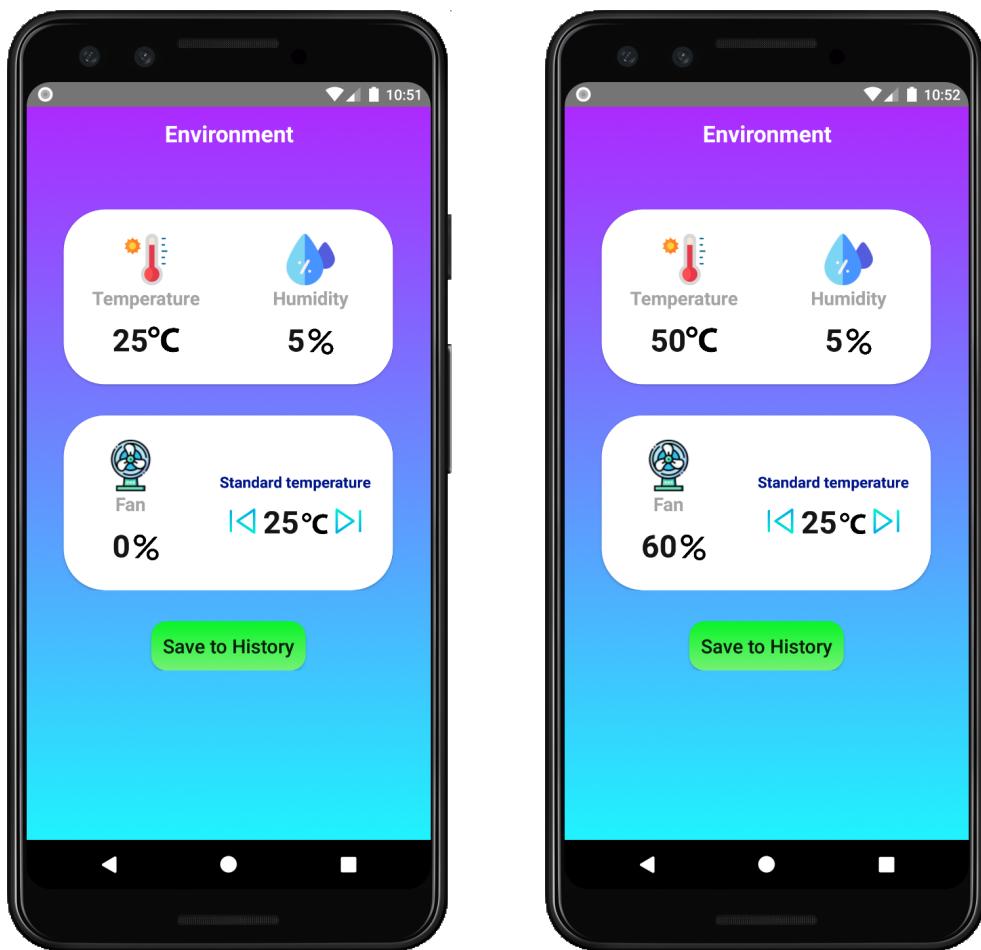
Hình 10: Giao diện trang chủ

6.1.3 Xem và thiết lập môi trường kho

Dây chính là phần trọng tâm của dự án.

Tại đây người dùng sẽ theo dõi thông số từ các thiết bị theo thời gian thực.

Ngoài ra ngoài dùng còn có thể thiết lập "nhiệt độ tiêu chuẩn" để quạt chạy với cơ chế như sau. Nếu nhiệt độ thực tế lớn hơn nhiệt độ tiêu chuẩn x độ quạt sẽ chạy với công suất ($2x$)% và công suất tối đa của quạt là 100%.



Hình 11: Giao diện xem môi trường kho

Ở hình ảnh trên, ban đầu nhiệt độ là 25 bằng với nhiệt độ tiêu chuẩn nên quạt sẽ không chạy. Nhưng sau đó nhiệt độ tăng lên 50 và quạt đã chạy với công suất tương ứng.

Ngoài ra ta còn có một nút "save to history" để lưu các thông số tại thời điểm hiện tại vào lịch sử. Tính năng xem lịch sử sẽ được giới thiệu trong phần tiếp theo.



Để thực hiện tự động bật tắt quạt, ứng dụng gọi đến hàm startMQTTTempHumi để lấy thông số cảm biến và thực hiện tính toán. Sau đó sẽ gọi hàm sendDataToMQTT (hàm này sẽ được trình bày ở phần kết nối server) để thay đổi thông số quạt (Speaker).

Listing 1: “startMQTTTempHumi”

```
private void startMQTTTempHumi(String ID, String topic, final TextView a,final TextView b, final TextView cel){
    mqttHelper = new MQTTHelper(getApplicationContext(), ID, topic);
    mqttHelper.setCallBack(new MqttCallbackExtended() {
        @Override
        public void connectComplete(boolean reconnect, String serverURI) {
        }
        @Override
        public void connectionLost(Throwable cause) {
        }
        @Override
        public void messageArrived(String topic, MqttMessage message) throws Exception {
            System.out.println(message.toString());
            JSONArray jsonArray = new JSONArray(message.toString());
            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                String device_id = jsonObject.getString("device_id");
                String location = jsonObject.getString("values");

                JSONArray arr_value = new JSONArray(location);
                a.setText(arr_value.getString(0));
                b.setText(arr_value.getString(1));
                float longitude = Float.parseFloat(a.getText().toString());
                if (!a.getText().toString().equals("") &&
                    !cel.getText().toString().equals("")){
                    if(longitude > Integer.parseInt(cel.getText().toString()) + 50){
                        sendDataToMQTT("Speaker", "1", "5000");
                    }else {
                        if(longitude <= Integer.parseInt(cel.getText().toString())){
                            sendDataToMQTT("Speaker", "0", "1");
                        }else {
                            String temp =
                                String.valueOf((Integer.parseInt(a.getText().toString()) -
                                Integer.parseInt(cel.getText().toString()))*2*50);
                            sendDataToMQTT("Speaker", "1", temp);
                        }
                    }
                }
            }
        }
        @Override
        public void deliveryComplete(IMqttDeliveryToken token) {
        }
    });
}
```



6.1.4 Xem lịch sử

Sau khi nhấn lưu lịch sử ở màn hình môi trường kho, ứng dụng sẽ đọc lại thông tin môi trường kho tại thời điểm đó. Nếu các thông số hợp lệ ứng dụng sẽ lưu lại vào Database kèm theo thời gian và tài khoản lưu. Đồng thời thông báo đến người dùng đã lưu thành công.

Dữ liệu sẽ được lưu vào nhánh History ở Database với key là thời gian và user người gửi. value là các thông số môi trường kho tại thời điểm đó.

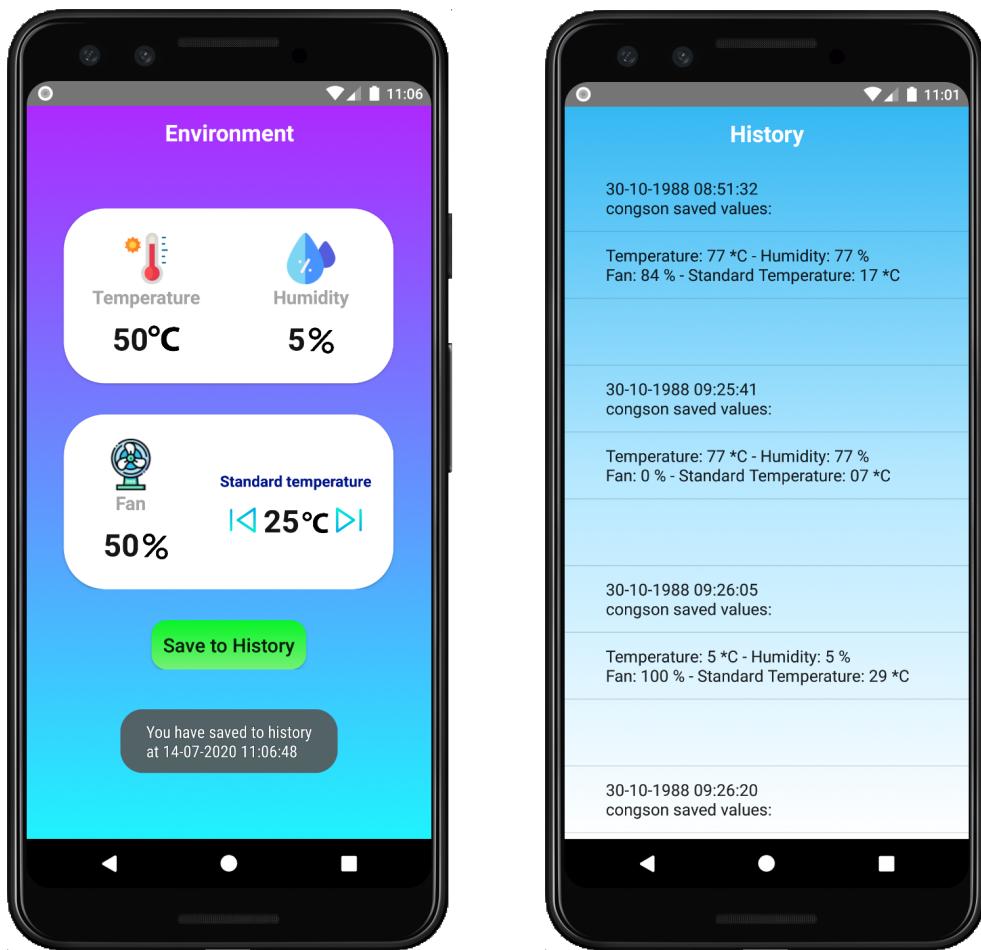
The screenshot shows a web browser window with the following details:

- URL: <https://warehouse-management-279915.firebaseio.com/Main/History>
- Page Title: warehouse-management-279915 > Main > History
- Content Area:
 - Section: History
 - List of data entries (each entry is a timestamp followed by a JSON object):
 - 1594222692548congson saved values:: "Temperature: 77 *C - Humidity: 77 %\nFan: 84 % - ..."
 - 1594224741218congson saved values:: "Temperature: 77 *C - Humidity: 77 %\nFan: 0 % - ..."
 - 1594224765302congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 100 % - ..."
 - 1594224780045congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 60 % - S..."
 - 1594224795272congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 40 % - S..."
 - 1594224808839congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 20 % - S..."
 - 1594224848233congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 0 % - St..."
 - 1594276647004congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 6 % - St..."
 - 1594301913747congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 100 % - ..."
 - 1594459849695congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 74 % - S..."
 - 1594459860099congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 74 % - S..."
 - 1594459937025congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 74 % - S..."
 - 1594460033532congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 74 % - S..."
 - 1594461381419hihih saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 74 % - S..."
 - 1594547737124sy saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 100 % - ..."
 - 1594742798647congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 50 % - S..."
 - 1594742808362congson saved values:: "Temperature: 5 *C - Humidity: 5 %\nFan: 50 % - S..."

Hình 12: Database lịch sử



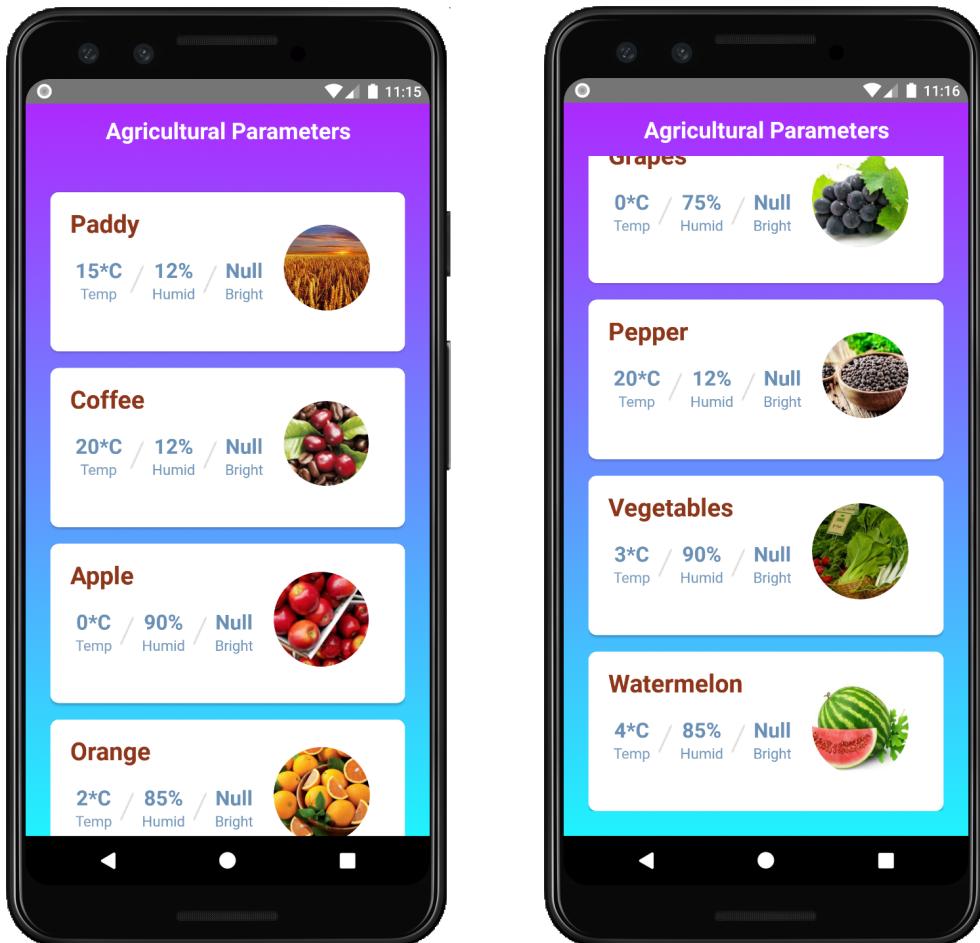
Khi người dùng vào giao diện Xem lịch sử sẽ thấy được toàn bộ lịch sử đã lược lưu dưới dạng list và sắp xếp theo thời gian.



Hình 13: Giao diện xem lịch sử

6.1.5 Xem thông số gợi ý các loại nông sản

Tại màn hình xem thông số gợi ý các loại nông sản, người dùng có thể thấy thông số nhiệt độ và độ ẩm phù hợp cho từng loại nông sản, từ đó có thể điều chỉnh thông số hợp lý cho kho bảo quản của mình.



Hình 14: Giao diện xem thông số gợi ý các loại nông sản



6.1.6 Xem thời tiết

Tại màn hình này, người dùng sẽ được xem thời tiết thực tế tại khu vực của kho bảo quản từ đó có thể đưa ra các thông số phù hợp, kịp thời thay đổi, kiểm tra hoặc ứng phó với những biến cố thời tiết.

Địa điểm của kho được thiết lập và thay đổi trong phần thay đổi thông tin sẽ được trình bày ở phần tiếp theo.

Thời tiết được lấy từ trang www.accuweather.com.

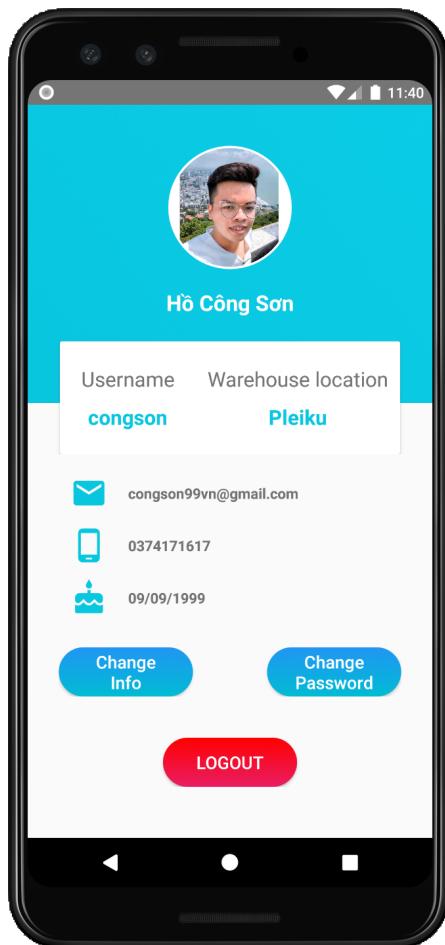


Hình 15: Giao diện xem thời tiết



6.1.7 Tuỳ chỉnh tài khoản

Khi vừa vào tính năng này, người dùng sẽ thấy thông tin tài khoản của mình cùng 3 nút chức năng là thay đổi thông tin, thay đổi mật khẩu và đăng xuất.

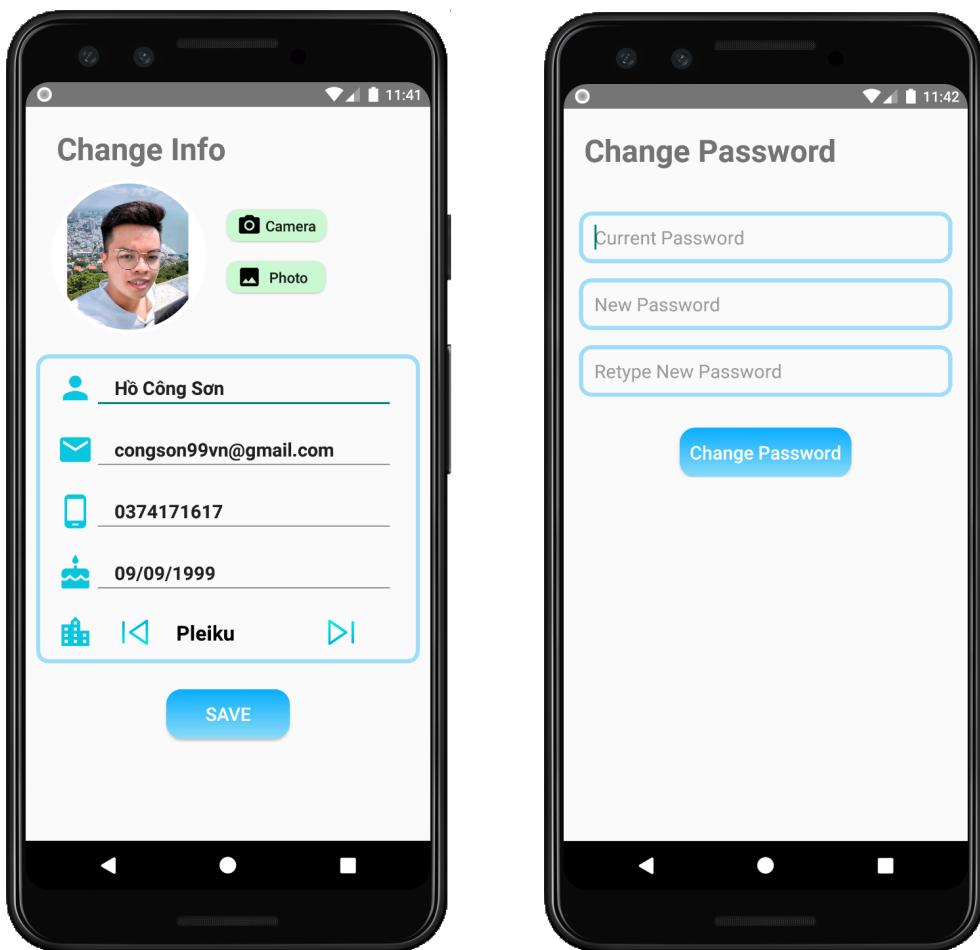


Hình 16: Giao diện tùy chỉnh tài khoản



Nếu nhấn vào nút thay đổi thông tin, ứng dụng sẽ đưa người dùng đến màn hình thay đổi thông tin.

Tại đây người dùng có thể thay đổi ảnh đại diện bằng cách chụp ảnh hoặc chọn ảnh từ thiết bị và thay đổi các thông tin khác bằng cách nhập. Riêng ô điền tên người dùng dùng sẽ không được để trống, nếu không ứng dụng sẽ báo lỗi và không lưu thay đổi. Ảnh đại diện sẽ được đưa về dạng byte trước khi lưu vào Database.



Hình 17: Giao diện thay đổi thông tin tài khoản và mật khẩu

Tương tự khi người dùng nhấn vào thay đổi mật khẩu, ứng dụng sẽ đưa người dùng đến giao diện thay đổi mật khẩu. Tại đây người dùng sẽ nhập mật khẩu hiện tại và mật khẩu mới. Ứng dụng sẽ kiểm tra xem các giá trị nhập có hợp lệ không. Nếu hợp lệ, mật khẩu sẽ được thay đổi và lưu lại vào Database.



6.2 Server, input và output

6.2.1 Kết nối server

Dầu tiên, chúng ta sử dụng file hỗ trợ MQTTHelper để kết nối với server giúp thực hiện các chức năng Publisher và Subscriber.

Listing 2: “Class MQTTHelper”

```
public class MQTTHelper {  
    final String serverUri = "tcp://52.187.125.59";  
  
    final String username = "BKvm";  
    final String password = "Hcmut_CSE_2020";  
  
    private String clientId, subscriptionTopic;  
  
    public MqttAndroidClient mqttAndroidClient;  
  
    public MQTTHelper(Context context, String Id, String Topic){  
        clientId = Id;  
        subscriptionTopic = Topic;  
        mqttAndroidClient = new MqttAndroidClient(context, serverUri, clientId);  
        mqttAndroidClient.setCallback(new MqttCallbackExtended() {  
            @Override  
            public void connectComplete(boolean reconnect, String serverURI) {  
                Log.w("Mqtt", serverURI);  
            }  
  
            @Override  
            public void connectionLost(Throwable cause) {  
            }  
  
            @Override  
            public void messageArrived(String topic, MqttMessage mqttMessage) throws  
                Exception {  
                Log.w("Mqtt", mqttMessage.toString());  
            }  
  
            @Override  
            public void deliveryComplete(IMqttDeliveryToken token) {  
            }  
        });  
  
        connect();  
    }  
  
    public void setCallBack(MqttCallbackExtended callBack){  
        mqttAndroidClient.setCallback(callBack);  
    }  
  
    private void connect(){  
        MqttConnectOptions mqttConnectOptions = new MqttConnectOptions();
```



```
mqttConnectOptions.setAutomaticReconnect(true);
mqttConnectOptions.setCleanSession(false);
mqttConnectOptions.setUserName(username);
mqttConnectOptions.setPassword(password.toCharArray());

try{
    mqttAndroidClient.connect(mqttConnectOptions, null, new
        IMqttActionListener() {
        @Override
        public void onSuccess(IMqttToken asyncActionToken) {
            DisconnectedBufferOptions disconnectedBufferOptions = new
                DisconnectedBufferOptions();
            disconnectedBufferOptions.setBufferEnabled(true);
            disconnectedBufferOptions.setBufferSize(100);
            disconnectedBufferOptions.setPersistBuffer(false);
            disconnectedBufferOptions.setDeleteOldestMessages(false);
            mqttAndroidClient.setBufferOpts(disconnectedBufferOptions);
            subscribeToTopic();
        }

        @Override
        public void onFailure(IMqttToken asyncActionToken, Throwable exception) {
            Log.w("Mqtt", "Failed to connect to: " + serverUri +
                exception.toString());
        }
    });
} catch (MqttException ex){
    Log.w("Exception", "Exception");
    ex.printStackTrace();
}

private void subscribeToTopic(){
    try {
        mqttAndroidClient.subscribe(subscriptionTopic, 0, null, new
            IMqttActionListener() {
            @Override
            public void onSuccess(IMqttToken asyncActionToken) {
                Log.w("Mqtt", "Subscribed!!!");
            }

            @Override
            public void onFailure(IMqttToken asyncActionToken, Throwable exception) {
                Log.w("Mqtt", "Subscribed fail!");
            }
        });
    } catch (MqttException ex) {
        System.err.println("Exception subscribing");
        ex.printStackTrace();
    }
}
}
```



Sau đó gọi và thực hiện Subscriber bằng hàm startMQTT. Ví dụ như hàm Subscriber vào Topic/Speaker dưới đây:

Listing 3: “startMQTTSpeaker”

```
private void startMQTTSpeaker(String ID, String topic, final TextView a){  
    mqttHelper = new MQTTHelper(getApplicationContext(), ID, topic);  
    mqttHelper.setCallBack(new MqttCallbackExtended() {  
        @Override  
        public void connectComplete(boolean reconnect, String serverURI){  
  
        }  
  
        @Override  
        public void connectionLost(Throwable cause) {  
  
        }  
  
        @Override  
        public void messageArrived(String topic, MqttMessage message) throws Exception {  
            System.out.println(message.toString());  
            JSONArray jsonArray = new JSONArray(message.toString()); for (int i = 0; i  
                < jsonArray.length(); i++) {  
                JSONObject jsonObject = jsonArray.getJSONObject(i);  
                String device_id = jsonObject.getString("device_id");  
                String location = jsonObject.getString("values");  
  
                JSONArray arr_value = new JSONArray(location);  
                if (arr_value.getString(0).equals("0") ||  
                    arr_value.getString(1).equals("0")){  
                    a.setText("0");  
                }  
                else {  
                    a.setText(String.valueOf(Integer.parseInt(arr_value.getString(1))/50));  
                }  
            }  
        }  
  
        @Override  
        public void deliveryComplete(IMqttDeliveryToken token) {  
  
        }  
    });  
}
```



Thực hiện Publish bằng hàm sendDataToMQTT.

Listing 4: “sendDataToMQTT”

```
private void sendDataToMQTT(String ID, String value1, String value2) throws
JSONException {

    MqttMessage msg = new MqttMessage();
    msg.setId(1234);
    msg.setQos(0);
    msg.setRetained(true);

    String x = "\\".substring(0,1);
    JSONObject payload = new JSONObject();
    payload.put("device_id", ID);
    payload.put("values", "");

    String list = "["+x+value1+x+","+x+value2+x+"]";
    String a = "[" + payload.toString().substring(0,payload.toString().length()-3) +
               list + ")]";
    byte[] b = a.getBytes(Charset.forName("UTF-8"));
    msg.setPayload(b);

    try {
        mqttHelper.mqttAndroidClient.publish("Topic/" + ID, msg);
    }catch (MqttException ignored){
    }
}
```



6.2.2 Input

Thông tin input nhận được sẽ có dạng JSON như sau.

Listing 5: “input”

```
[  
  { "device_id": "TempHumi ",  
    "values": ["60", "72"]  
  }  
]
```

Trong đó, 60 là nhiệt độ và 72 là độ ẩm ("values": ["Nhiệt độ", "Độ ẩm không khí"]), TempHumi là tên thiết bị.

6.2.3 Output

Thông tin output nhận được sẽ có dạng JSON như sau.

Listing 6: “output”

```
[  
  { "device_id": "Speaker",  
    "values": ["1", "100"]  
  }  
]
```

Trong đó, thiết bị output sẽ có 2 trạng thái là “OFF” ứng với giá trị 0” và “ON” ứng với giá trị 1” kèm theo đó là giá trị về cường độ.

Miền giá trị cường độ âm thanh: [0, 5000]

Khi thực hiện Publish ta gửi lên server file JSON có định dạng tương tự đã được chuyển về String.



7 Tổng kết

7.1 Kết quả

Sau 12 tuần làm việc cùng nhau, nhóm đã đạt được những kết quả đúng như mong đợi, cụ thể như sau:

- Nhóm đã hoàn thành hoàn chỉnh ứng dụng, giống tới 95% mục tiêu đề ra lúc đầu.
- Thực hiện demo thành công, kết nối và điều khiển được các thiết bị.
- Có thêm kiến thức chuyên môn về IOT, thiết kế ứng dụng,... và nhiều kỹ năng mềm như làm việc nhóm, tổ chức phát triển dự án.

7.1.1 Hạn chế

Tuy nhiên vẫn còn tồn tại một vài hạn chế đến từ dự án và hoạt động nhóm chưa được khắc phục:

- Một số ít tính năng chưa được như mong muốn vì giới hạn năng lực và sự đầu tư chưa được tối đa.
- Việc demo chỉ mang tính tương đối vì giới hạn về thiết bị.
- Nhóm đôi lúc trì trệ công việc và phân bổ thời gian chưa thực sự tốt.



8 Hướng phát triển

Sau khi hoàn thành dự án, nhóm nhận thấy đây là dự án rất có tiềm năng, mang nhiều lợi ích thực tế và thị trường rất rộng lớn.

Vì vậy, nhóm đã đề ra rất nhiều hướng phát triển và hoàn thiện dự án. Cụ thể như sau:

- Trước hết phát triển ứng dụng trên đa nền tảng chứ không chỉ android để mọi người dùng đều có thể sử dụng (IOS, PC, Web).
- Thiết kế xây dựng lại cấu trúc quản lý kho, đa người dùng để phục vụ cho nhiều người dùng với kho riêng của mình.
- Xây dựng server để quản lý giúp việc đa kho, đa người dùng trở nên dễ dàng hơn.
- Tăng cường thêm nhiều tính năng, nhiều tùy chọn tự động hóa.
- Liên kết với công ty cung cấp thiết bị để tối ưu hóa hệ thống IOT cũng như dễ dàng lắp đặt cho người dùng cuối.
- Tích hợp các công nghệ mới như AI để phân tích số liệu, định hướng phát triển, phát triển tính năng dựa trên thói quen người dùng hoặc dựa trên hướng phát triển chung của ngành.
- Mở rộng dự án sang các lĩnh vực liên quan có thể tích hợp như camera giám sát an ninh cho kho bảo quản, quản lý nông sản và gởi ý đầu tư, quản lý nông trại,...



Tài liệu

- [1] Tài liệu được cung cấp từ môn học
 - Manual-for-Android-device.docx
 - Sơ đồ triển khai mô hình IoT.docx
 - WorkflowData-format.docx
- [2] Hệ thống quản lý kho
https://vi.wikipedia.org/wiki/Hệ_thống_quản_lý_kho
- [3] Internet of things
https://en.wikipedia.org/wiki/Internet_of_things
- [4] Lập trình Android
<https://www.youtube.com/playlist?list=PLzrVYRai0riSRJ3M3bifVWWRq5eJMu6tv>
- [5] Design UI
 - <https://www.youtube.com/watch?v=LD2zsCAAVXw>
 - <https://www.youtube.com/watch?v=L0cd1evBcSA>
 - <https://www.youtube.com/watch?v=2pirZvqXza0>
 - <https://www.youtube.com/watch?v=Uur6-64KqxI>
 - <https://www.youtube.com/watch?v=fqZDL8XsWHc>
- [6] Firebase on Android
 - <https://firebase.google.com/docs/database/android/start>
 - https://www.youtube.com/playlist?list=PLzrVYRai0riTLPLclyGuByHvZ8_tDZZIr
- [7] Icon
<https://www.flaticon.com>
- [8] Tiêu chuẩn nhiệt độ các loại thực phẩm
<https://kholanhnambac.com/nhung-tieu-chuan-ve-nhiet-do-bao-quan-trong-kho-lanh>
- [9] Thời tiết
<https://www.accuweather.com>