BỘ GIÁO DỤC VÀ ĐÀO TẠO

**ĐẠI HỌC PHENIKAA**

----

A logo on a black background

AI-generated content may be incorrect.

**LẬP TRÌNH ĐIỀU KHIỂN TRÊN THIẾT BỊ DI ĐỘNG**

TÊN ĐỀ TÀI

HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ- ĐỘ ẨM VÀ ĐIỀU KHIỂN ĐÈN, QUẠT THÔNG GIÓ TRONG TRANG TRẠI

Sinh viên thực hiện: Trần Sơn Dương

Mã sinh viên: 21011224

Lớp K15-ĐK&TĐH1

Giảng viên hướng dẫn: ThS.Đào Tô Hiệu

Khoa: Điện – Điện tử

*Hà Nội, tháng 7 năm 2025*

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**ĐẠI HỌC PHENIKAA**

----

A logo on a black background

AI-generated content may be incorrect.

**LẬP TRÌNH ĐIỀU KHIỂN TRÊN THIẾT BỊ DI ĐỘNG**

TÊN ĐỀ TÀI

HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ- ĐỘ ẨM VÀ ĐIỀU KHIỂN ĐÈN, QUẠT THÔNG GIÓ TRONG TRANG TRẠI

Sinh viên thực hiện: Trần Sơn Dương

Mã sinh viên: 21011224

Lớp K15-ĐK&TĐH1

Giảng viên hướng dẫn: ThS.Đào Tô Hiệu

Khoa: Điện – Điện tử

*Hà Nội, tháng 7 năm 2025*

**LỜI CAM ĐOAN**

Chúng em xin cam đoan đề tài: “Hệ thống giám sát nhiệt độ- độ ẩm và điều khiển đèn, quạt thông gió trong trang trại” là một công trình nghiên cứu độc lập không có sự sao chếp, thuê mượn, copy của người khác. Đề tài là một sản phẩm mà chúng tôi đã nỗ lực nghiên cứu trong quá trình học tập tại trường. Trong quá trình viết bài có sự tham khảo một số tài liệu có nguồn gốc rõ ràng, dưới sự hướng dẫn của thầy ThS. Đào Tô Hiệu – Giảng viên Khoa Điện- Điện tử, Trường Kỹ thuật Phenikaa. Chúng em xin cam đoan sẽ chịu hoàn toàn trách nhiệm nếu có vấn đề xảy ra.

Sinh viên thực hiện

*(Ký và ghi rõ họ tên)*

**RUBIC BÁO CÁO/TIỂU LUẬN**

**Môn học**: Lập trình điều khiển trên thiết bị di động **Mã học phần**: EEE703049

**Giảng viên**: ThS. Đào Tô Hiệu **Hình thức**: Bài tập lớn/tiểu luận

**Thời gian**: Học kỳ III, 2024-2025 **Bậc học**: Đại học **Số tín chỉ**: 3

| **Mức độ đạt chuẩn quy định** | | | | | | | **CĐR** | **Trọng số** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tiêu chí đánh giá** | | **Điểm**  **(0-3.9)** | **Điểm**  **(4.0-5.4)** | **Điểm**  **(5.5-6.9)** | **Điểm**  **(7.0-8.4)** | **Điểm**  **(8.5-10)** |
| **Điều kiện tiên quyết** | | Báo cáo, sản phẩm phải do bản thân sinh viên/nhóm sinh viên thực hiện. Nếu có dấu hiệu copy, thuê làm Đồ án hoặc đạt được sản phẩm nhưng không nắm được nội dung/kết quả, hoặc không đạt được tiến độ (TC2<4.0 điểm) thì sinh viên bị 0 điểm Thi KTHP. | | | | |  |  |
| TC1 | Hình thức trình bày báo cáo | * Không trình bày hình thức báo cáo theo yêu cầu của 1 báo cáo khoa học. * Có lỗi chính tả. * Chưa trình bày logic nội dung liên quan lĩnh vực đề tài. | | * Trình bày hình thức báo cáo theo yêu cầu. * Có lỗi chính tả. * Chưa trình bày logic nội dung liên quan lĩnh vực đề tài. | | * Hoàn thiện về hình thức báo cáo. * Không có lỗi chính tả. * Trình bày logic các nội dung liên quan lĩnh vực thuộc đề tài. |  | 20% |
| TC2 | Tiến độ thực hiện | * SV không đạt tiến độ về nội dung thực hiện được giao. | * Sinh viên nộp quyển báo cáo đúng hạn. * Có sản phẩm theo yêu cầu. * Sản phẩm không vận hành được | * SV nộp báo cáo đúng hạn. * Có sản phẩm vận hành được. * Sản phẩm không đúng theo 100% yêu cầu. | * SV nộp báo cáo đúng hạn. * Có sản phẩm vận hành được. * Sản phẩm không đúng theo 100% yêu cầu. * Sản phẩm chưa ổn định | * SV nộp báo cáo đúng hạn. * Có sản phẩm vận hành được. * Sản phẩm đạt 100% yêu cầu. * Sản phẩm vận hành ổn định. |  | 20% |
| TC3 | Kỹ năng nhóm | * Không tham gia nhóm. * Không giải quyết được các vấn đề thuộc phạm vi đề tài dựa trên làm việc nhóm. | * Dưới 40% thành viên tham gia thực hiện đề tài. * Giải quyết chưa tốt các vấn đề thuộc phạm vi đề tài dựa trên làm việc nhóm | * Từ 41% đến dưới 60% thành viên tham gia thực hiện đề tài. * Giải quyết cơ bản các vấn đề thuộc phạm vi đề tài dựa trên làm việc nhóm. | * Từ 60% đến dưới 85% thành viên tham gia thực đề tài. * Giải quyết tương đối tốt các vấn đề thuộc phạm vi đề tài dựa trên làm việc nhóm. | * 100% thành viên tham gia thực hiện. * Giải quyết được 100% các vấn đề thuộc phạm vi đề tài | 3.1 | 20% |
| TC4 | Cơ sở lý thuyết | * Giải thích được các khái niệm cốt lõi của IoT và các công nghệ hỗ trợ (50%). * Mô tả được nguyên lý thiết kế và phát triển hệ thống IoT và ứng dụng (50%). | | | | | 1.1 | 20% |
| TC5 | Chất lượng sản phẩm | - Thiết kế được các ứng dụng Android trên cơ sở các đối tượng Activity, View, XML, Intent, Dialog, các công cụ thiết kế giao diện… (20%)  - Lập trình điều khiển phần cứng/phần mềm qua truyền thông (20%).  - Sản phẩm có đầy đủ các thành phần hệ thống theo mô tả (20%).  - Làm rõ được những bài toán cần lập trình đề cần giải quyết các vấn đề thuộc phạm vi đề tài (giao thức truyền thông, ứng dụng thực tiễn…) (20%).  - Sản phẩm đề tài có khả năng vận hành ổn định dựa trên kết quả lập trình ứng dụng (20%). | | | | | 2.1 | 20% |

|  |  |
| --- | --- |
| **KHOA ĐIỆN- ĐIỆN TỬ** | **GIẢNG VIÊN** |

**TIẾN ĐỘ THỰC HIỆN**

**I) Thông tin sinh viên**

|  |  |  |
| --- | --- | --- |
| **STT** | **Hình ảnh** | **Thông tin cá nhân** |
| 1 |  | - Họ và tên: Trần Sơn Dương  - Mã SV: 21011224  - Lớp: K15 ĐK&TĐH1  - Trường Kỹ thuật Phenikaa.  - SĐT: 0359036307  - Nơi ở: Hà Nội |

**II) Nhiệm vụ**

|  |  |  |
| --- | --- | --- |
| **Thành viên** | **Nhiệm vụ** | **Tự đánh giá** |
| Trần Sơn Dương | Hoàn thiện toàn bộ project từ việc thiết kế app, thiết kế hệ thống, vẽ mạch, làm mạch in, viết báo cáo | **Đạt** |

**II) Tiến độ thực hiện**

|  |  |  |
| --- | --- | --- |
| **Nội dung** | **Thời gian thực hiện** | **Kết quả** |
| Tìm hiểu về thông tin môn học, kiến thức đã học và chọn đề tài phù hợp | 23/04/2025- 20/06-2025 | Hoàn thành |
| Thiết kệ hệ thống, lập trình app, thiết kế mạch in | 21/06/2025 – 21/07/2025 | Hoàn thành |
| Làm mạch cứng,hoàn thiện báo cáo | 22/07/2025- 29/07/2025 | Hoàn thành |

**Nhóm sinh viên thực hiện**

*(Ký, ghi rõ họ tên)*

**III, Đánh giá tiến độ thực hiện**

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**GIẢNG VIÊN**

(Ký và ghi rõ họ tên)

**MỤC LỤC**

[CHƯƠNG I: CƠ SỞ LÝ THUYẾT 1](#_Toc204772448)

[1.1.Các thành phần chính của dự án 1](#_Toc204772449)

[1.1.1.Android Studio 1](#_Toc204772450)

[1.1.2.Xampp 1](#_Toc204772451)

[1.1.3.Arduino IDE 2](#_Toc204772452)

[1.2.Mục tiêu 3](#_Toc204772453)

[1.3.Các thành phần cấu tạo ứng dụng trên thiết bị di động 4](#_Toc204772454)

[1.3.1.Các layout 4](#_Toc204772455)

[1.3.2.Các lớp Activity 5](#_Toc204772456)

[1.3.3.Class và các thành phần khác 5](#_Toc204772457)

[1.3.4.Giao thức truyền thông HTTP 6](#_Toc204772458)

[1.4.Kết luận chương 8](#_Toc204772459)

[CHƯƠNG II: THIẾT KẾ HỆ THỐNG 9](#_Toc204772460)

[2.1.Mô tả hệ thống 9](#_Toc204772461)

[2.1.1.Kiến trúc hệ thống 9](#_Toc204772462)

[2.1.2.Tính năng 12](#_Toc204772463)

[2.2.Triển khai hệ thống 15](#_Toc204772464)

[2.2.1.Phần cứng 15](#_Toc204772465)

[2.2.2.Cài đặt XAMPP và thiết lập server 19](#_Toc204772466)

[2.2.3.Lập trình trên Arduino IDE 22](#_Toc204772467)

[2.2.4.Thiết lập project trên Android Studio 26](#_Toc204772468)

[2.2.5.Lưu đồ thuật toán 43](#_Toc204772469)

[2.3.Chức năng ứng dụng 49](#_Toc204772470)

[2.3.1.Hiển thị dữ liệu 49](#_Toc204772471)

[2.3.2.Tương tác và điều khiển 51](#_Toc204772472)

[2.3.3.Các chức năng khác 53](#_Toc204772473)

[2.3.3.Phương pháp đánh giá kiểm thử 55](#_Toc204772474)

[2.4.Kết luận chương 56](#_Toc204772475)

[CHƯƠNG III: KẾT QUẢ VÀ ĐÁNH GIÁ 58](#_Toc204772476)

[3.1.Tiến độ thực hiện 58](#_Toc204772477)

[3.2.Kết quả đạt được 58](#_Toc204772478)

[3.3.Kết luận 58](#_Toc204772479)

[3.3.1.Ưu điểm 58](#_Toc204772480)

[3.3.2.Nhược điểm 58](#_Toc204772481)

[3.3.3.Giải pháp khắc phục 58](#_Toc204772482)

[3.4.Hướng phát triển 58](#_Toc204772483)

[TÀI LIỆU THAM KHẢO 60](#_Toc204772484)

[1.Tài liệu tham khảo tiếng Việt 60](#_Toc204772485)

[2.Tài liệu tham khảo tiếng Anh 60](#_Toc204772486)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1: Android Studio phiên bản 2023.1.1 1](#_Toc204772404)

[Hình 1.2: Giao diện của XAMPP 2](#_Toc204772405)

[Hình 1.3: Arduino IDE 3](#_Toc204772406)

[Hình 1.4: Sơ đồ hoạt động của HTTP 7](#_Toc204772407)

[Hình 2.1: Sơ đồ khối hệ thống 9](#_Toc204772408)

[Hình 2.2: Sơ đồ mô tả các lớp và các chức năng chính 12](#_Toc204772409)

[Hình 2.3: Module Esp32 15](#_Toc204772410)

[Hình 2.4: Cảm biến nhiệt độ, độ ẩm DHT11 16](#_Toc204772411)

[Hình 2.5: Đèn sợi đốt 17](#_Toc204772412)

[Hình 2.6: Quạt tản nhiệt 17](#_Toc204772413)

[Hình 2.7: Module Relay 18](#_Toc204772414)

[Hình 2.8: Sơ đồ nguyên lý mạch 19](#_Toc204772415)

[Hình 2.9: Khởi động XAMPP chạy quyền quản trị viên 20](#_Toc204772416)

[Hình 2.10: Các file đuôi php đã được gửi vào tệp htdocs/BTL\_DHT11 20](#_Toc204772417)

[Hình 2.11: Database mydata\_dht11 22](#_Toc204772418)

[Hình 2.12: Thiết kế giao diện tại tab LoginActivity 39](#_Toc204772419)

[Hình 2.13: Thiết kế giao diện cho tab MainActivity 40](#_Toc204772420)

[Hình 2.14: Thiết kế giao diện cho tab BluetoothActivity 41](#_Toc204772421)

[Hình 2.15: Thiết kế giao diện cho tab DataActivity 41](#_Toc204772422)

[Hình 2.16: Thiết kế giao diện cho tab AdminActivity 42](#_Toc204772423)

[Hình 2.17: Thiết kế giao diện cho tab RegisterActivity 43](#_Toc204772424)

[Hình 2.18: Lưu đồ thuật toán con StartActivity 43](#_Toc204772425)

[Hình 2.19: Lưu đồ thuật toán con LoginActivity 44](#_Toc204772426)

[Hình 2.20: Lưu đồ thuật toán con RegisterActivity 44](#_Toc204772427)

[Hình 2.21: Lưu đồ thuật toán con ForgotPasswordActivity 45](#_Toc204772428)

[Hình 2.22: Lưu đồ thuật toán con BluetoothActivity 45](#_Toc204772429)

[Hình 2.23: Lưu đồ thuật toán MainActivity 46](#_Toc204772430)

[Hình 2.24: Lưu đồ thuật toán con AdminActivity 46](#_Toc204772431)

[Hình 2.25: Lưu đồ thuật toán con SensorMonitorService 47](#_Toc204772432)

[Hình 2.26: Lưu đồ thuật toán của app 48](#_Toc204772433)

[Hình 2.27: Giao diện hiển thị dữ liệu dưới dạng đồ thị 49](#_Toc204772434)

[Hình 2.28: Giao diện tab chính MainActivity 50](#_Toc204772435)

[Hình 2.29: Giao diện tab BluetoothActivity 51](#_Toc204772436)

[Hình 2.30: Tương tác và điều khiển thiết bị 52](#_Toc204772437)

[Hình 2.31: Mạch khi chạy thực tế 53](#_Toc204772438)

[Hình 2.32: Giao diện tab Setting Activity 54](#_Toc204772439)

[Hình 2.33: Giao diện thiết kế tab AdminActivity 55](#_Toc204772440)

**DANH MỤC BẢNG BIỂU**

[Bảng 2.1: Cấu trúc giao diện người dùng 10](#_Toc204772441)

[Bảng 2.2: Cấu trúc chức năng cho Admin 10](#_Toc204772442)

[Bảng 2.3: Liên kết với hệ thống IoT 11](#_Toc204772443)

[Bảng 2.4: Giao tiếp với ESP32 14](#_Toc204772444)

[Bảng 2.5: Lưu trữ và giao tiếp 15](#_Toc204772445)

[Bảng 2.6: Thông số kĩ thuật của quạt tản nhiệt 18](#_Toc204772446)

[Bảng 2.7: Chức năng chính của các file PHP 22](#_Toc204772447)

# CHƯƠNG I: CƠ SỞ LÝ THUYẾT

## 1.1.Các thành phần chính của dự án

### 1.1.1.Android Studio

Android Studio là môi trường tích hợp IDE để viết code và phát triển ứng dụng trên nền tảng hệ điều hành Android, dành cho Google Android Development được ra mắt vào ngày 16 tháng 5 năm 2013, trong sự kiện I/O 2013 của Google.

Android Studio chứa tất cả các công cụ để hỗ trợ việc phát hành ứng dụng Android như: thiết kế, kiểm tra, gỡ lỗi và cấu hình ứng dụng. Android Studio sử dụng Gradle để quản lý dự án của bạn.[[1]](#footnote-1)

A cartoon hedgehog in a dark room

AI-generated content may be incorrect.

Hình 1.1: Android Studio phiên bản 2023.1.1

### 1.1.2.Xampp

XAMPP là viết tắt của 5 module được tích hợp bên trong nó bao gồm là Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) và Perl (P). XAMPP là một phần mềm nguồn mở và miễn phí dùng để tạo web server trên máy tính cá nhân (Localhost), XAMPP tương thích với các hệ điều hành phổ biến như : Linux, MacOS, Windows,..

Ưu điểm lớn nhất của XAMPP là mã nguồn mở và tính dễ sử dụng, tương đối đơn giản, gọn nhẹ nên được sử dụng ngày càng phổ biến hiện nay[[2]](#footnote-2).

XAMPP được sử dụng rộng rãi từ người dùng phổ thông đến các lập trình viên chuyên nghiệp, Xampp còn được sử dụng để vận hành cũng như phát triển các website dùng ngôn ngữ lập trình PHP như: WordPress, Magento, Joomla!,..

A screenshot of a computer

AI-generated content may be incorrect.

Hình 1.2: Giao diện của XAMPP

Với XAMPP, người dùng có thể phát triển và kiểm thử trang web mà không cần kết nối internet. Đây là được coi là một tiện ích tuyệt vời cho việc phát triển và thử nghiệm ứng dụng web trước khi triển khai thực tế. XAMPP còn cung cấp giao diện điều khiển đơn giản để phục vụ cho việc quản lý các dịch vụ web và cơ sở dữ liệu.

### 1.1.3.Arduino IDE

Arduino IDE là một phần mềm với một mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã vào module Arduino. Nó bao gồm phần cứng và phần mềm. Phần cứng chứa đến 300,000 board mạch được thiết kế sẵn với các cảm biến, linh kiện. Phần mềm giúp bạn có thể sử dụng các cảm biến, linh kiện ấy của Arduino một cách linh hoạt phù hợp với mục đích sử dụng.[[3]](#footnote-3)



Hình 1.3: Arduino IDE

Đây là phần mềm mã nguồn mở, miễn phí với tất cả mọi người. Sử dụng ngôn ngữ lập trình C/C++ thân thiện với các lập trình viên, với đa dạng các thư viện phong phú, hỗ trợ nhiều loại bo mạch như arduino, esp,..Đồng thời nó cũng có giao diện dễ dùng, hỗ trợ đa nền tảng từ Window đến Linux, MacOs.

## 1.2.Mục tiêu

Dự án này nhằm phát triển một ứng dụng Android để giám sát và điều khiển môi trường trong trang trại, với các mục tiêu:

* Giám sát nhiệt độ và độ ẩm, xây dựng ứng dụng cho phép theo dõi nhiệt độ và độ ẩm trong trang trại theo thời gian thực, hiển thị dữ liệu rõ ràng để người dùng dễ dàng nắm bắt tình hình.
* Điều khiển đèn và quạt thông gió, cung cấp chức năng điều khiển từ xa các thiết bị đèn và quạt thông gió thông qua ứng dụng, giúp người dùng bật/tắt thiết bị một cách linh hoạt.
* Điều khiển tự động, tích hợp tính năng tự động điều chỉnh đèn và quạt dựa trên dữ liệu từ cảm biến nhiệt độ và độ ẩm, đảm bảo môi trường trong trang trại luôn ở trạng thái tối ưu.
* Lưu trữ và hiển thị dữ liệu lịch sử theo thời gian, đồng thời cung cấp giao diện để người dùng xem lại dữ liệu lịch sử và phân tích xu hướng.
* Cảnh báo điều kiện bất thường, gửi thông báo đến người dùng khi nhiệt độ hoặc độ ẩm vượt quá ngưỡng cho phép, hỗ trợ phản ứng nhanh chóng trước các tình huống bất lợi.

## 1.3.Các thành phần cấu tạo ứng dụng trên thiết bị di động

### 1.3.1.Các layout

Các tệp bao gồm:

* activity\_about\_us.xml: Giao diện cho màn hình "Giới thiệu về chúng tôi".
* activity\_admin.xml: Giao diện cho màn hình Quản lý tài khoản dưới quyền admin
* activity\_create\_user.xml: Giao diện để tạo người dùng mới dưới quyền admin
* activity\_data.xml: Giao diện hiển thị dữ liệu dưới dạng đồ thị của nhiệt độ, độ ẩm
* activity\_edit\_profile.xml: Giao diện chỉnh sửa hồ sơ người dùng.
* activity\_forgot\_password.xml: Giao diện khôi phục mật khẩu.
* activity\_login.xml: Giao diện đăng nhập.
* activity\_logout.xml: Giao diện đăng xuất.
* activity\_main.xml: Giao diện màn hình chính của ứng dụng, hiển thị nhiệt độ độ ẩm, đồng thời hiển thị nút bật tắt đèn và quạt.
* activity\_notifications.xml: Giao diện hiển thị thông báo.
* activity\_register.xml: Giao diện đăng ký người dùng.
* activity\_settings.xml: Giao diện cài đặt.
* activity\_start.xml: Giao diện màn hình khởi động
* activity\_bluetooth.xml : Giao diện bật tắt Bluetooth, hiển thị thiết bị đã được kết nối, thiết bị đang hoạt động đồng thời hiển thị nhiệt độ, độ ẩm

### 1.3.2.Các lớp Activity

Các lớp activity được sử dụng trong đề tài:

* AboutUsActivity: Xử lý màn hình "Về chúng tôi", những thông tin người dùng có thể liên hệ với chúng tôi thông qua các cách thức liên lạc như email, số điện thoại, địa chỉ
* AdminActivity: Xử lý màn hình quản trị viên, quản lý danh sách tài khoản
* CreateUserActivity: Xử lý chức năng tạo người dùng mới dưới quyền admin
* DataActivity: Xử lý màn hình hiển thị dữ liệu để giám sát nhiệt độ, độ ẩm
* EditProfileActivity: Xử lý chỉnh sửa thông tin tài khoản
* ForgotPasswordActivity: Xử lý khôi phục mật khẩu.
* LoginActivity: Xử lý đăng nhập tài khoản bằng kết nối Wifi, truy cập kết nối Bluetooth
* BluetoothActivity: Bật tắt Bluetooth, kết nối các thiết bị thông qua Bluetooth, nhận dữ liệu từ ESP32 về nhiệt độ, độ ẩm để hiển thị.
* LogoutActivity: Xử lý đăng xuất.
* MainActivity: Màn hình chính của ứng dụng, nơi hiển thị nhiệt độ, độ ẩm, đồng thời điều khiển đèn và quạt
* NotificationsActivity: Xử lý màn hình thông báo.
* RegisterActivity: Xử lý đăng ký người dùng.
* SettingsActivity: Xử lý màn hình cài đặt.
* StartActivity: Xử lý màn hình khởi động.

### 1.3.3.Class và các thành phần khác

* AccountAdapter: Adapter hỗ trợ hiển thị danh sách tài khoản.
* Account: Định nghĩa đối tượng tài khoản, hỗ trợ truyền, xử lý và hiển thị thông tin người dùng trong app Android
* CustomMarkerView: Lớp tùy chỉnh cho việc hiển thị marker, dùng để hiển thị biểu đồ cho tab DataActivity
* NotificationAdapter: Adapter hỗ trợ hiển thị danh sách thông báo.
* SensorMonitorService: Dịch vụ giám sát cảm biến, chạy nền khi tắt app, đảm bảo trong trường hợp nhiệt độ, độ ẩm quá cao hoặc quá thấp
* BluetoothConnectionManager: Giữ một BluetoothSocket dùng chung trong toàn app, đồng thời cung cấp các hàm tiện ích để quản lý kết nối

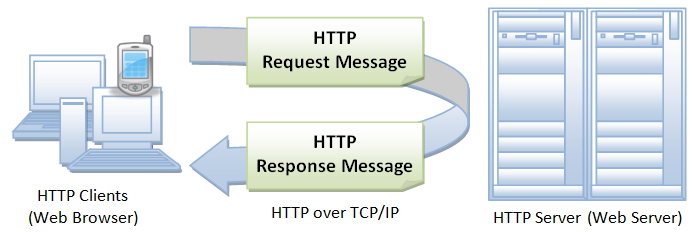
### 1.3.4.Giao thức truyền thông HTTP

#### 1.3.4.1.HTTP là gì?

HTTP (HyperText Transfer Protocol - Giao thức truyền tải siêu văn bản) là một trong các giao thức chuẩn về mạng Internet, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (Web server) và Máy sử dụng dịch vụ (Web client), là giao thức Client/Server dùng cho World Wide Web – WWW

HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP (các giao thức nền tảng cho Internet). [[4]](#footnote-4)

#### 1.3.4.2.Sơ đồ hoạt động của HTTP



Hình 1.4: Sơ đồ hoạt động của HTTP[[5]](#footnote-5)

HTTP hoạt động dựa trên mô hình Client – Server. Trong mô hình này, các máy tính của người dùng sẽ đóng vai trò làm máy khách (Client). Sau một thao tác nào đó của người dùng, các máy khách sẽ gửi yêu cầu đến máy chủ (Server) và chờ đợi câu trả lời từ những máy chủ này.

HTTP là một stateless protocol. Hay nói cách khác, request hiện tại không biết những gì đã hoàn thành trong request trước đó.

HTTP cho phép tạo các yêu cầu gửi và nhận các kiểu dữ liệu, do đó cho phép xây dựng hệ thống độc lập với dữ liệu được truyển giao.[[6]](#footnote-6)

#### 1.3.4.3.HTTP trong XAMPP

HTTP (HyperText Transfer Protocol) là giao thức tiêu chuẩn để truyền tải dữ liệu trên mạng, đặc biệt là trong việc phục vụ các trang web, và nó đóng vai trò quan trọng trong XAMPP. XAMPP là một gói phần mềm mã nguồn mở bao gồm máy chủ Apache HTTP Server, cùng với MariaDB, PHP và Perl, được sử dụng để tạo môi trường phát triển web cục bộ.

Trong XAMPP, máy chủ Apache sử dụng HTTP làm giao thức mặc định để phục vụ các tệp web (như HTML, PHP) từ thư mục \xampp\htdocs. Khi khởi động Apache, có thể truy cập các tệp này qua địa chỉ HTTP, ví dụ: http://localhost hoặc http://127.0.0.1, thường trên cổng 80.

HTTP hoạt động theo mô hình yêu cầu-đáp ứng, nơi trình duyệt hoặc ứng dụng gửi yêu cầu (request) đến máy chủ, và máy chủ trả về dữ liệu (response). Trong dự án này, HTTP có thể được dùng để gửi dữ liệu cảm biến ( nhiệt độ, độ ẩm) và lệnh điều khiển (như bật/tắt đèn, quạt) thông qua các API hoặc trang web cục bộ.

XAMPP sử dụng HTTP theo mặc định, nhưng có thể cấu hình thêm để hỗ trợ HTTPS (HTTP Secure) bằng cách bật SSL/TLS trong Apache, yêu cầu chứng chỉ bảo mật.

Với đề tài "Hệ thống giám sát nhiệt độ - độ ẩm và điều khiển đèn, quạt thông gió trong trang trại", HTTP trong XAMPP có thể giúp kết nối ứng dụng Android với máy chủ cục bộ để xử lý dữ liệu và điều khiển thiết bị, đặc biệt khi phát triển và kiểm thử.

## 1.4.Kết luận chương

Chương 1 đã cung cấp nền tảng lý thuyết vững chắc cho dự án "Hệ thống giám sát nhiệt độ - độ ẩm và điều khiển đèn, quạt thông gió trong trang trại" thông qua việc giới thiệu các công cụ và công nghệ cốt lõi. Android Studio được xác định là môi trường phát triển chính, với các thành phần của dự án như các Activity, layout cùng với các class, hỗ trợ việc thiết kế và lập trình ứng dụng một cách hiệu quả. XAMPP, với máy chủ Apache sử dụng giao thức HTTP, đóng vai trò quan trọng trong việc tạo môi trường cục bộ để phục vụ dữ liệu và điều khiển thiết bị, đặc biệt thông qua các tệp PHP. Arduino IDE là phần mềm mã nguồn mở được sử dụng để đọc dữ liệu từ cảm biến và gửi lên server local thông qua giao thức HTTP.

# CHƯƠNG II: THIẾT KẾ HỆ THỐNG

## 2.1.Mô tả hệ thống

### 2.1.1.Kiến trúc hệ thống

#### 2.1.1.1.Sơ đồ khối hệ thống

A diagram of a computer system

AI-generated content may be incorrect.

Hình 2.1: Sơ đồ khối hệ thống

Hệ thống bao gồm 3 khối chính: Thiết bị IoT, Server và Ứng dụng. Cụm thiết bị IoT sẽ đọc dữ liệu từ cảm biến DHT11 sau đó gửi lên server thông qua giao thức HTTP. Các dữ liệu được lưu trên server local, sau đó ứng dụng sẽ lấy dữ liệu từ server để hiển thị trên app. Server có chức năng lưu trữ dữ liệu, xác thực tài khoản và xử lý các yêu cầu từ App Android.Dữ liệu được lưu dưới dạng tham số, ESP32 sẽ liên tục lấy dữ liệu yêu cầu bật tắt thiết bị từ trên server, từ đó điều khiển bật tắt đèn hoặc quạt theo mong muốn.

#### 2.1.1.2.Cấu trúc ứng dụng Android

Giao diện người dùng:

|  |  |
| --- | --- |
| **Activity** | **Chức năng chính** |
| StartActivity | Màn hình khởi động, kiểm tra đăng nhập |
| LoginActivity | Người dùng đăng nhập hoặc truy cập Bluetooth |
| RegisterActivity | Người dùng mới đăng ký |
| MainActivity | Trang chính: hiển thị nhiệt độ, độ ẩm, điều khiển đèn/quạt |
| DataActivity | Biểu đồ dữ liệu nhiệt độ, độ ẩm |
| NotificationsActivity | Hiển thị cảnh báo quá ngưỡng |
| SettingsActivity | Cài đặt tài khoản |
| EditProfileActivity | Đổi username/mật khẩu (có kiểm tra mật khẩu cũ) |
| ForgotPasswordActivity | Khôi phục mật khẩu |
| LogoutActivity | Xử lý đăng xuất |
| AboutUsActivity | Thông tin liên hệ |
| BluetoothActivity | Bật tắt Bluetooth, kết nối Bluetooth với các thiết bị, hiển thị nhiệt độ, độ ẩm |

Bảng 2.1: Cấu trúc giao diện người dùng

Chức năng cho tài khoản Admin:

|  |  |
| --- | --- |
| **Màn hình/Adapter** | **Chức năng** |
| AdminActivity | Quản lý danh sách tài khoản |
| CreateUserActivity | Tạo user mới (guest/admin) |
| AccountAdapter | Adapter hiển thị danh sách user |

Bảng 2.2: Cấu trúc chức năng cho Admin

Biểu đồ và cảnh báo:

* CustomMarkerView: Marker tùy chỉnh khi chạm vào điểm dữ liệu trong biểu đồ (DataActivity).
* NotificationAdapter: Adapter hiển thị danh sách cảnh báo.
* SensorMonitorService: Dịch vụ chạy nền, định kỳ lấy dữ liệu từ server và cảnh báo nếu nhiệt độ/độ ẩm vượt ngưỡng.
* NotificationService: Tạo thông báo foreground có rung và lưu nội dung cảnh báo.

#### 2.1.1.3.Liên kết với hệ thống IoT

Dữ liệu của cảm biến được gửi từ ESP32 lên server thông qua file send\_data.php

|  |  |
| --- | --- |
| **Thành phần Android** | **Giao tiếp với Server / IoT** |
| SensorMonitorService | Gửi request đến get\_data.php để đọc nhiệt độ/độ ẩm |
| MainActivity | Điều khiển đèn/quạt qua control.php |
| DataActivity | Lấy dữ liệu biểu đồ từ get\_all\_data.php |
| LoginActivity,RegisterActivity,  CreateUserActivity | Gửi POST đến login.php, register.php, create\_user.php |
| EditProfileActivity | Gửi POST đến update\_profile.php |
| ForgotPasswordActivity | Gửi POST đến reset\_password.php |

Bảng 2.3: Liên kết với hệ thống IoT

#### 2.1.1.4.Sơ đồ mô tả các lớp và các chức năng chính của ứng dụng

A diagram of a diagram

AI-generated content may be incorrect.

Hình 2.2: Sơ đồ mô tả các lớp và các chức năng chính

### 2.1.2.Tính năng

#### 2.1.2.1. Đăng ký - Đăng nhập - Phân quyền

Đăng ký (RegisterActivity)

* Người dùng điền username, email, password.
* Gửi thông tin đến server thông qua register.php.
* Mật khẩu được mã hóa trước khi lưu vào CSDL.

Sau khi đăng ký thành công, chuyển về trang LoginActivity.

Đăng nhập (LoginActivity)

* Người dùng nhập tài khoản và mật khẩu, nếu thiết bị không có Wifi thì truy cập BluetoothActivity
* Gửi thông tin đến login.php, kiểm tra tài khoản.
* Nếu hợp lệ: Lưu thông tin đăng nhập (username, role) vào SharedPreferences.
* Chuyển đến MainActivity.
* Phân quyền: tùy theo role (admin, guest) để điều hướng phù hợp.

#### 2.1.2.2.Màn hình chính (MainActivity)

* Hiển thị nhiệt độ, độ ẩm hiện tại (từ get\_data.php).
* Điều khiển Đèn/Quạt thông qua nút nhấn:
* Gửi lệnh POST đến control.php (với trạng thái bật/tắt).
* Tự động lấy dữ liệu mỗi vài giây/lần bằng Handler.

#### 2.1.2.3.Thông báo cảnh báo cảm biến

SensorMonitorService

* Chạy nền (background service) khi mở ứng dụng.
* Chu kỳ 3 giây, gửi request đến get\_data.php.

Nếu:

Nhiệt độ > 35°C hoặc < 10°C

Độ ẩm > 70% hoặc < 20%

Thì gọi NotificationService để hiển thị thông báo nguy hiểm.

NotificationService

* Hiển thị thông báo rung & biểu tượng cảnh báo.
* Ghi lại thông báo vào SharedPreferences (theo từng tài khoản).
* Khi người dùng mở NotificationsActivity sẽ thấy toàn bộ thông báo của mình.

#### 2.1.2.4. Xem dữ liệu biểu đồ (DataActivity)

Dữ liệu lấy từ get\_all\_data.php.

Dùng thư viện MPAndroidChart để hiển thị:

* Biểu đồ nhiệt độ
* Biểu đồ độ ẩm

Tương tác tốt, có MarkerView hiển thị giá trị tại điểm chạm.

#### 2.1.2.5. Quản lý người dùng (AdminActivity)

Chức năng dành riêng cho tài khoản admin:

* Danh sách tài khoản (get\_all\_user.php)
* Hiển thị toàn bộ danh sách người dùng.
* Sử dụng AccountAdapter với ListView hoặc RecyclerView.

Tạo người dùng (CreateUserActivity)

* Admin tạo tài khoản mới (admin hoặc guest).
* Gửi thông tin qua create\_user.php.
* Trường created\_by lưu lại ai là người tạo (theo logged\_in\_user).

Xóa tài khoản (delete\_user.php)

* Admin có thể xóa tài khoản bất kỳ (trừ admin gốc).

Cập nhật quyền tài khoản (update\_user\_role.php)

* Chuyển đổi quyền admin ↔ guest.

#### 2.1.2.6. Chỉnh sửa tài khoản (EditProfileActivity)

Người dùng nhập:

* Mật khẩu cũ (để xác thực).
* Tên đăng nhập mới, mật khẩu mới (được mã hóa).

Gửi yêu cầu POST tới update\_profile.php.

Chỉ thực hiện nếu xác thực mật khẩu cũ thành công.

#### 2.1.2.7.Quên mật khẩu (ForgotPasswordActivity)

Khi người dùng nhập email, app sẽ gửi request đến file reset\_password.php, khi đó hệ thống sẽ thực hiện:

* Tạo mật khẩu mới ngẫu nhiên.
* Gửi về email người dùng.
* Lưu lại mật khẩu mới đã mã hóa vào CSDL.

#### 2.1.2.8.Kết nối Bluetooth (BluetoothActivity)

Người dùng bật tắt Bluetooth của điện thoại thông qua nút switch trên app, với các thiết bị từ Android 12 trở đi, muốn tắt Bluetooth thì app sẽ trực tiếp truy cập cài đặt Bluetooth trên điện thoại để người dùng tắt.

Các thiết bị từng được kết nối Bluetooth và các thiết bị đang bật Bluetooth sẽ được hiển thị. Muốn gỡ kết nối với các thiết bị đã kết nối thì người dùng cần nhấn giữ để gỡ thiết bị, muốn kết nối với bất kì thiết bị nào đang hoạt động thì người dùng cũng cần nhấn giữ để kết nối với thiết bị.

Dữ liệu nhiệt độ, độ ẩm được gửi từ ESP32 thông qau giao tiếp Bluetooth và hiển thị lên trên app

#### 2.1.2.9.Giao tiếp với ESP32 (IoT)

|  |  |
| --- | --- |
| Thành phần | Tương tác |
| ESP32 | Gửi nhiệt độ, độ ẩm thông qua send\_data.php |
| Nhận lệnh điều khiển từ get\_control.php |
| App Android | Lấy dữ liệu từ get\_data.php |
| Gửi lệnh điều khiển lên control.php |

Bảng 2.4: Giao tiếp với ESP32

ESP32 hoạt động như Client IoT, gửi thông tin theo định kỳ và điều khiển thiết bị thật (đèn/quạt) qua chân GPIO.

#### 2.1.2.10. Lưu trữ & Giao tiếp

|  |  |
| --- | --- |
| **Dữ liệu** | **Cách lưu/Giao tiếp** |
| Thông tin tài khoản | CSDL My SQL (bảng users) |
| Dữ liệu cảm biến | Bảng dht11 |
| Trạng thái thiết bị | Bảng device\_control |
| Thông báo người dùng | Lưu local (SharedPreferences) |
| Thông tin đăng nhập | Lưu local (MyAppPrefs) |

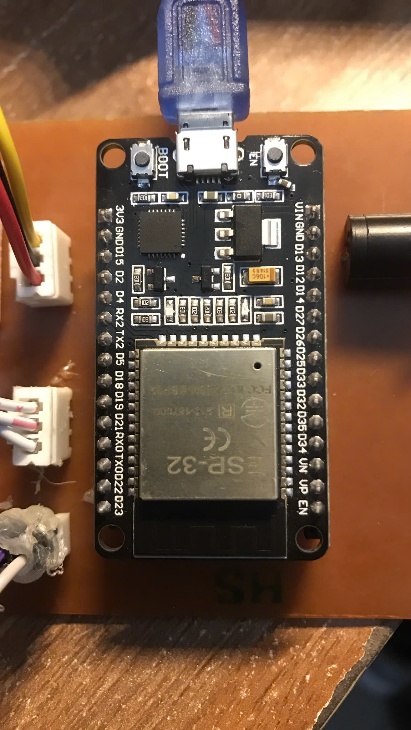
Bảng 2.5: Lưu trữ và giao tiếp

## 2.2.Triển khai hệ thống

### 2.2.1.Phần cứng

#### 2.2.1.1.Module ESP32

Module ESP32 được Espressif phát triển, với khả năng kết nối WiFi và Bluetooth không dây cũng như bộ xử lý lõi kép. Đây là sản phẩm kế thừa của ESP8266 với nhiều tính năng mới hơn.



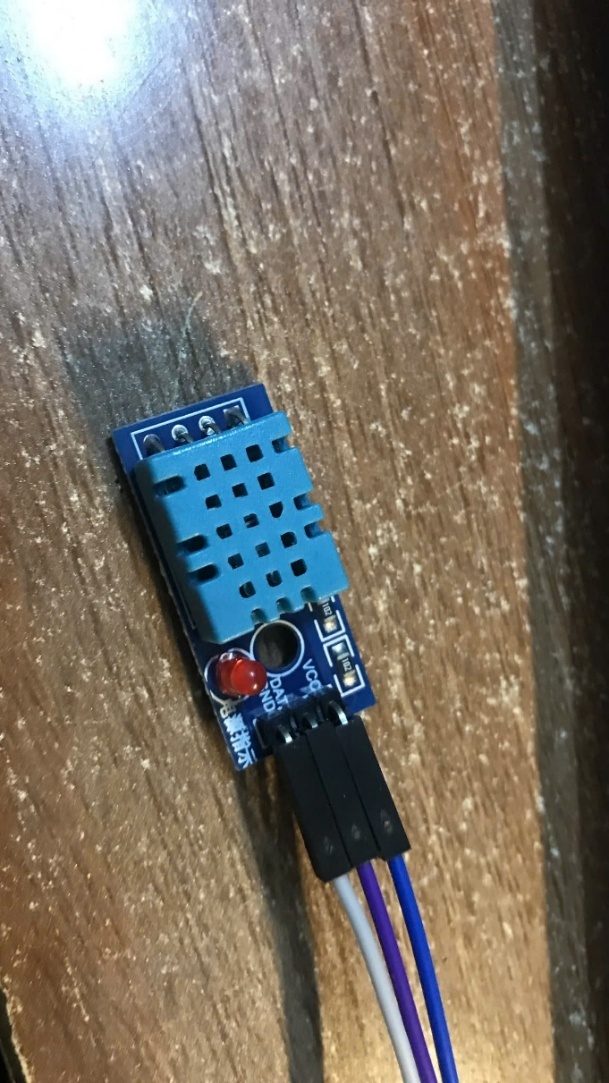
Hình 2.3: Module Esp32

Thông số kĩ thuật:

* Chip điều khiển: ESP32, CPU dual-core Xtensa® 32-bit LX6, tần số lên đến 240 MHz.
* Bộ nhớ: 520 KB SRAM trên chip; hỗ trợ Flash ngoài lên đến 16 MB và RAM ngoài lên đến 8 MB.
* Wi-Fi: Chuẩn 802.11 b/g/n, 2.4 GHz.
* Bluetooth: Bluetooth v4.2 BR/EDR và Bluetooth LE.
* Số chân GPIO: 34 chân có thể lập trình, trong đó 30 chân ra ngoài cho DevKit V1.
* Điện áp hoạt động: 3.3V (phạm vi từ 2.3V đến 3.6V).
* Chân ADC: ADC 12-bit với 18 kênh.
* Chân DAC: 2 kênh DAC 8-bit.
* Chân cảm biến cảm ứng: 10 kênh cảm biến điện dung.
* Giao tiếp ngoại vi: SPI, I2C, UART, PWM[[7]](#footnote-7)

#### 2.2.1.2.DHT11

DHT11 Là cảm biến nhiệt độ, độ ẩm rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1-wire ( giao tiếp digital 1-wire truyền dữ liệu duy nhất). Cảm biến được tích hợp bộ tiền xử lý tín hiệu giúp dữ liệu nhận về được chính xác mà không cần phải qua bất kỳ tính toán nào.



Hình 2.4: Cảm biến nhiệt độ, độ ẩm DHT11

Đặc điểm:

* Điện áp hoạt động : 3V - 5V (DC)
* Dải độ ẩm hoạt động : 20% - 90% RH, sai số ±5%RH
* Dải nhiệt độ hoạt động : 0°C ~ 50°C, sai số ±2°C
* Tần số lấy mẫu tối đa: 1 Hz
* Khoảng cách truyển tối đa: 20m
* Cảm biến DHT11 gồm 2 chân cấp nguồn, và 1 chân tín hiệu.[[8]](#footnote-8)

#### 2.2.1.3. Đèn sợi đốt

Để mô hình cho đề tài, nên em đã chọn đèn sợi đốt 12V/21W để làm



Hình 2.5: Đèn sợi đốt

Thông số sản phẩm:

* Điện áp: 12V
* Công suất: 35 / 35w
* Tăng sáng: +30%
* Tuổi thọ: 400 giờ[[9]](#footnote-9)

#### 2.2.1.4.Quạt tản nhiệt 5V



Hình 2.6: Quạt tản nhiệt

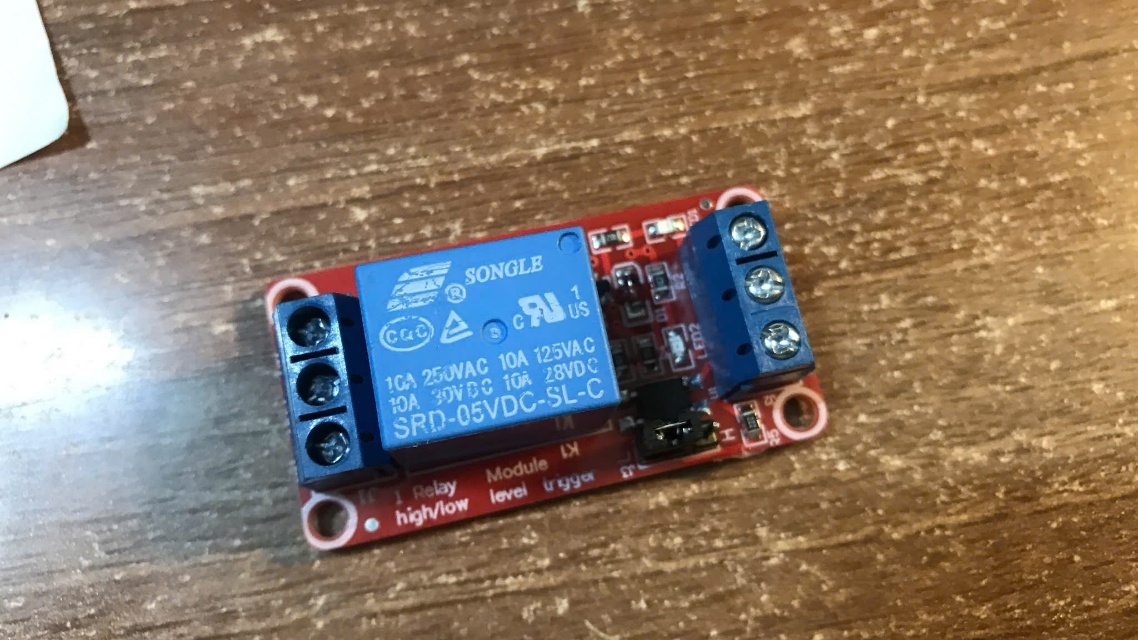
Dưới đây là bảng thông số kĩ thuật của thiết bị:

|  |  |
| --- | --- |
| **Điện áp** | 5VDC |
| **Cường độ dòng** | 0.1A |
| **Công suất** | 0.5W |
| **Tốc độ** | 7000 vòng/phút |
| **Kích thước (DàixRộngxCao)** | 3x3x1cm |
| **Tuổi thọ** | 30.000 giờ |
| **Nhiệt độ hoạt động** | -10 ~ 70\*C |
| **Chất liệu** | Nhựa cao cấp |
| **Phụ kiện tương thích** | Tấm bảo vệ quạt tản nhiệt 3cm |

Bảng 2.6: Thông số kĩ thuật của quạt tản nhiệt [[10]](#footnote-10)

#### 2.2.1.5.Module Relay

Module relay 5V là một thiết bị điện tử dùng để đóng/ngắt các thiết bị điện công suất cao bằng tín hiệu điều khiển điện áp thấp 5V từ vi điều khiển như Arduino, Raspberry Pi...



Hình 2.7: Module Relay

Thông số kỹ thuật của Module Relay

* Điện áp tải maximum: AC 250V/10A, DC 30V/10A
* Điện áp hoạt động: 5V
* Dòng kích Relay: 5mA
* Relay tích cực mức thấp
* Bốn lỗ Via để bắt ốc: đường kính 3.1mm
* Kích thước: 50x26x18.5mm[[11]](#footnote-11)

#### 2.2.1.6.Sơ đồ nguyên lý của mạch

A diagram of a computer chip

AI-generated content may be incorrect.

Hình 2.8: Sơ đồ nguyên lý mạch

Mạch sử dụng bộ xử lý trung tâm là module ESP32 , sử dụng các module relay 5V đóng vai trò một công tắc điều khiển bằng điện, cho phép đóng/ngắt mạch điện công suất lớn (AC 220V, quạt, đèn, động cơ…) bằng tín hiệu điều khiển từ ESP32. Từ đó giúp mạch luôn đảm bảo hoạt động ổn định cho thiết bị, chân tín hiệu của đèn được nối với chân D5, chân tín hiệu của quạt được nối với chân D18. Cảm biến DHT11 được sử dụng để đọc tín hiệu nhiệt độ, độ ẩm, chân tín hiệu được nối với chân D4 của ESP32. Ngoài ra mạch còn thiết kế một JACK DC cái để cấp nguồn hoạt động cho module ESP32

### 2.2.2.Cài đặt XAMPP và thiết lập server

Khởi động Apache và MySQL

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.9: Khởi động XAMPP chạy quyền quản trị viên

Gửi các file PHP vào tệp htdocs/BTL\_DHT11

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.10: Các file đuôi php đã được gửi vào tệp htdocs/BTL\_DHT11

|  |  |
| --- | --- |
| **Tên file PHP** | **Chức năng chính** |
| db.php | Kết nối cơ sở dữ liệu MySQL. Là file được include trong hầu hết các file khác. |
| login.php | Xử lý đăng nhập người dùng (kiểm tra username, password, trả về kết quả JSON). |
| register.php | Xử lý đăng ký tài khoản mới, lưu vào CSDL với quyền mặc định (guest hoặc admin). |
| create\_user.php | Cho phép tài khoản admin tạo tài khoản khác (admin hoặc guest). |
| delete\_user.php | Xóa tài khoản khỏi CSDL, chỉ cho phép admin xóa. |
| get\_all\_user.php | Lấy toàn bộ danh sách tài khoản (username, quyền) để hiển thị trong AdminActivity. |
| update\_user\_role.php | Cập nhật quyền của người dùng (từ guest sang admin và ngược lại). |
| update\_profile.php | Cho phép người dùng đổi username hoặc password (nếu nhập đúng mật khẩu cũ |
| reset\_password.php | Gửi lại mật khẩu mới ngẫu nhiên đến email người dùng (dùng thư viện PHPMailer). |
| logout.php | Xử lý đăng xuất (kết thúc phiên hoặc xác thực logout). |
| get\_data.php | Lấy dữ liệu nhiệt độ, độ ẩm từ CSDL để hiển thị trên biểu đồ. |
| get\_all\_data.php | Trả về toàn bộ dữ liệu cảm biến từ ESP32 (theo bảng log hoặc biểu đồ). |
| send\_data.php | ESP32 gửi dữ liệu cảm biến (nhiệt độ, độ ẩm) lên server qua HTTP POST. |
| control.php | App gửi yêu cầu điều khiển thiết bị (bật/tắt đèn, quạt) lên server. |
| get\_control.php | ESP32 lấy lệnh điều khiển hiện tại từ server (đèn/quạt đang ở trạng thái nào). |
| hash\_admin.php | |  | | --- | |  |  |  | | --- | | Tạo mật khẩu admin đầu tiên có mã hóa (thường dùng một lần). | |

Bảng 2.7: Chức năng chính của các file PHP

Import database từ mydata\_dht11.sql ( qua phpMyAdmin)

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.11: Database mydata\_dht11

### 2.2.3.Lập trình trên Arduino IDE

Với phần lập trình trên Arduino IDE để module esp32 đọc dữ liệu từ cảm biến DHT11, sau đó gửi dữ liệu lên serve sử dụng file send\_data.php, đồng thời lấy dữ dữ liệu điều khiển từ trên server xuống để bật tắt quạt và đèn bằng file get\_control.php

Khai báo thư viện sử dụng:

#include <WiFi.h>

#include <HTTPClient.h>

#include <ArduinoJson.h>

#include <DHT.h>

#include "BluetoothSerial.h"

Định nghĩa các phần cứng và biến, đồng thời khởi tạo cảm biến DHT11 và thiết lập đối tượng giao tiếp Bluetooth

BluetoothSerial SerialBT;

#define DHTPIN 4

#define DHTTYPE DHT11

DHT dht11(DHTPIN, DHTTYPE);

#define LED\_PIN  5

#define FAN\_PIN 18

Cấu hình Wifi cục bộ để ESP kết nối , đồng thời cấu hình URL chỉ đến server xử lý nhận dữ liệu DHT11

String URL = "http://172.20.10.8/BTL\_DHT11/send\_data.php";

const char\* ssid = "Duong Tran";

const char\* password = "11111111";

Tại hàm setup(), khởi tạo Serial, cảm biến, chân I/O và Bluetooth, đồng thời gọi hàm connectWiFi() để kết nối WiFi.

void setup() {

  Serial.begin(115200);

  dht11.begin();

  pinMode(LED\_PIN, OUTPUT);

  pinMode(FAN\_PIN, OUTPUT);

  // Bắt đầu Bluetooth với tên thiết bị

  SerialBT.begin("Duong\_Tran\_ESP");

  Serial.println("Bluetooth đã bật. Đang chờ kết nối...");

  connectWiFi();

}

Tại vòng lặp chính loop(), thực hiện đọc dữ liệu từ DHT11, kiểm tra và kết nối lại Wifi, gửi dữ liệu về nhiệt độ, độ ẩm lên server khi có Wifi, gửi dữ liệu về nhiệt độ, độ ẩm khi có kết nối Bluetooth.

void loop() {

  Load\_DHT11\_Data();

// Nếu WiFi chưa kết nối, thử kết nối lại

  if (WiFi.status() != WL\_CONNECTED) {

    connectWiFi();

  }

  // Nếu có WiFi thì mới gửi dữ liệu và lấy lệnh

  if (WiFi.status() == WL\_CONNECTED) {

    String postData = "temperature=" + String(temperature) + "&humidity=" + String(humidity);

    HTTPClient http;

    http.begin(URL);

    http.addHeader("Content-Type", "application/x-www-form-urlencoded");

    int httpCode = http.POST(postData);

    String payload = http.getString();

    Serial.print("URL : "); Serial.println(URL);

    Serial.print("Data: "); Serial.println(postData);

    Serial.print("httpCode: "); Serial.println(httpCode);

    Serial.print("payload : "); Serial.println(payload);

    Serial.println("--------------------------------------------------");

    getControlCommand(); // lấy lệnh điều khiển từ server

    http.end();

  } else {

    Serial.println("Không có kết nối WiFi. Đang chỉ hiển thị dữ liệu cảm biến...");

  }

  // Gửi dữ liệu qua Bluetooth để app Android nhận

  String btData = "Temp:" + String(temperature) + "\*C, Humi:" + String(humidity) + "%";

  SerialBT.println(btData);

  delay(500); // Delay để tránh đọc quá nhanh

}

Thiết lập hàm Load\_DHT11\_Data() để đọc và kiểm tra dữ liệu từ cảm biến, đồng thời in ra Serial Monitor

void Load\_DHT11\_Data() {

  temperature = dht11.readTemperature(); //Celsius

  humidity = dht11.readHumidity();

  // Check if any reads failed.

  if (isnan(temperature) || isnan(humidity)) {

    Serial.println("Failed to read from DHT sensor!");

    temperature = 0;

    humidity = 0;

  }

  Serial.printf("Temperature: %d °C\n", temperature);

  Serial.printf("Humidity: %d %%\n", humidity);

}

Thiết lập hàm connectWiFi() thực hiện tắt WiFi, rồi bật lại và kết nối với mạng được chỉ định, nếu sau 5 giây chưa kết nối, báo lỗi.

void connectWiFi() {

  WiFi.mode(WIFI\_OFF);

  delay(1000);

  WiFi.mode(WIFI\_STA);

  WiFi.begin(ssid, password);

  Serial.println("Đang kết nối WiFi...");

  unsigned long startTime = millis();

  const unsigned long timeout = 5000; // 5 giây

  while (WiFi.status() != WL\_CONNECTED && millis() - startTime < timeout) {

    delay(500);

    Serial.print(".");

  }

  if (WiFi.status() == WL\_CONNECTED) {

    Serial.println("\n Đã kết nối WiFi!");

    Serial.print("IP: "); Serial.println(WiFi.localIP());

  } else {

    Serial.println("\n Kết nối WiFi thất bại.");

  }

}

Tạo hàm getControlCommand() để gửi HTTP GET đến server để lấy trạng thái thiết bị, phân tích chuỗi JSON trả về, từ đó điều khiển trạng thái chân LED và FAN:

void getControlCommand() {

  HTTPClient http;

  http.begin("http://172.20.10.8/BTL\_DHT11/get\_control.php");

  int httpCode = http.GET();

  if (httpCode == 200) {

  String payload = http.getString();

  Serial.print("Control Payload: ");

  Serial.println(payload);

  // Phân tích JSON

  DynamicJsonDocument doc(256);  //512

  DeserializationError error = deserializeJson(doc, payload);

  if (!error) {

    int ledState = doc["led"];

    int fanState = doc["fan"];

    digitalWrite(LED\_PIN, ledState == 1 ? HIGH : LOW);

    digitalWrite(FAN\_PIN, fanState == 1 ? HIGH : LOW);

    Serial.printf("LED: %d, FAN: %d\n", ledState, fanState);

  } else {

    Serial.println("Lỗi phân tích JSON");

  }

}

  http.end();

}

### 2.2.4.Thiết lập project trên Android Studio

#### 2.2.4.1.Cấp quyền và khai báo các thành phần chính trong ứng dụng

Ứng dụng sử dụng nhiều quyền hệ thống để:

* Truy cập Internet và kiểm tra mạng (INTERNET, ACCESS\_NETWORK\_STATE, CHANGE\_NETWORK\_STATE) nhằm gửi dữ liệu cảm biến và nhận lệnh từ server.
* Quản lý Bluetooth (BLUETOOTH\_\*) để quét, kết nối và trao đổi dữ liệu với thiết bị ESP32.
* Gửi thông báo (POST\_NOTIFICATIONS) và rung (VIBRATE) khi có cảnh báo môi trường.
* Cấp quyền cho dịch vụ nền (FOREGROUND\_SERVICE) nhằm duy trì khả năng theo dõi liên tục.

<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.CHANGE\_NETWORK\_STATE" />  
<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />  
<uses-permission android:name="android.permission.FOREGROUND\_SERVICE" />  
<uses-permission android:name="android.permission.VIBRATE" />  
<uses-permission android:name="android.permission.POST\_NOTIFICATIONS" />  
<uses-permission android:name="android.permission.FOREGROUND\_SERVICE\_DATA\_SYNC" />  
<uses-permission android:name="android.permission.BLUETOOTH\_ADMIN"/>  
<uses-permission android:name="android.permission.BLUETOOTH"/>  
<uses-permission android:name="android.permission.BLUETOOTH\_CONNECT"/>  
<uses-permission android:name="android.permission.BLUETOOTH\_SCAN"/>  
<uses-permission android:name="android.permission.BLUETOOTH\_SCAN"/>  
<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />  
<uses-permission android:name="android.permission.BLUETOOTH\_BOND" />

Khai báo các activity phục vụ các chức năng chính như:

* Đăng nhập, đăng ký, quên mật khẩu, chỉnh sửa tài khoản.
* Kết nối Bluetooth độc lập, hiển thị dữ liệu cảm biển
* Điều khiển thiết bị qua giao diện chính.
* Quản trị hệ thống người dùng (AdminActivity).
* Hiển thị dữ liệu cảm biến, thông báo và tùy chỉnh ứng dụng.
* Ứng dụng sử dụng hai service chạy nền gồm NotificationService để hiển thị thông báo khi có điều kiện cảnh báo và SensorMonitorService để giám sát dữ liệu cảm biến liên tục.

<activity  
 android:name=".BluetoothActivity"  
 android:exported="true">  
 </activity>  
  
 <service  
 android:name=".NotificationService"  
 android:enabled="true"  
 android:exported="false" />  
 <service  
 android:name=".SensorMonitorService"  
 android:exported="false" />  
  
 <activity  
 android:name=".CreateUserActivity"  
 android:exported="true"/>  
 <activity  
 android:name=".ForgotPasswordActivity"  
 android:exported="true" />  
 <activity  
 android:name=".LogoutActivity"  
 android:exported="true" />  
 <activity  
 android:name=".AboutUsActivity"  
 android:exported="true" />  
 <activity  
 android:name=".NotificationsActivity"  
 android:exported="true" />  
 <activity  
 android:name=".EditProfileActivity"  
 android:exported="true" />  
 <activity  
 android:name=".SettingsActivity"  
 android:exported="true" />  
 <activity  
 android:name=".DataActivity"  
 android:exported="true" />  
 <activity  
 android:name=".LoginActivity"  
 android:exported="true" />  
 <activity  
 android:name=".RegisterActivity"  
 android:exported="false" />  
 <activity  
 android:name=".AdminActivity"  
 android:exported="true" />  
 <activity  
 android:name=".StartActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <activity  
 android:name=".MainActivity"  
 android:exported="true" />  
</application>

#### 2.2.4.2.Các Activity chính

LoginActivity là Activity chịu trách nhiệm xử lý đăng nhập cho người dùng, kết nối với server để xác thực thông tin tài khoản

* Nhập tên đăng nhập và mật khẩu bằng cách lầy thông tin từ các EditText

String username= etUsername.getText().toString().trim();  
String password= etPassword.getText().toString().trim();

* Chuyển đến các Activity khác như RegisterActivity, ForgotPasswordActivity và BluetoothActivity.

tvRegister.setOnClickListener(v->{  
 Intent intent = new Intent(LoginActivity.this,RegisterActivity.class);  
 startActivity(intent);  
});  
tvForgotPassword.setOnClickListener(v->{  
 Intent intent= new Intent(LoginActivity.this, ForgotPasswordActivity.class);  
 startActivity(intent);  
});  
BLE.setOnClickListener(v->{  
 Intent intent= new Intent(LoginActivity.this,BluetoothActivity.class);  
 startActivity(intent);  
});

* Gửi yêu cầu đăng nhập đến server bằng HTTP POST (sử dụng thư viện OkHttp).

JSONObject json= new JSONObject();  
try {  
 json.put("username",username);  
 json.put("password",password);  
} catch (JSONException e){  
 e.printStackTrace();  
}  
RequestBody body= RequestBody.*create*(json.toString(), MediaType.*parse*("application/json"));  
Request request= new Request.Builder()  
 .url(loginUrl)  
 .post(body)  
 .build();  
OkHttpClient client= new OkHttpClient();  
client.newCall(request).enqueue(new Callback() {...});

* Xử lý phản hồi từ server và lưu thông tin đăng nhập vào SharedPreferences

SharedPreferences prefs = getSharedPreferences("MyAppPrefs", *MODE\_PRIVATE*);  
SharedPreferences.Editor editor = prefs.edit();  
editor.putString("logged\_in\_user", username); // <- lưu username  
editor.putString("user\_role",role); //giu dang nhap sau khi dong app  
editor.putBoolean("isLoggedIn", true);  
editor.apply();

* Chuyển sang MainActivity nếu đăng nhập thành công

Intent intent = new Intent(LoginActivity.this, MainActivity.class);  
intent.putExtra("username", username);  
intent.putExtra("role", role); // truyền role (admin hoặc guest)  
startActivity(intent);

MainActivity là trung tâm của ứng dụng, nơi hiển thị dữ liệu cảm biến nhiệt độ - độ ẩm và cho phép người dùng điều khiển thiết bị từ xa như LED và quạt thông qua server.

* Lấy dữ liệu nhiệt độ, độ ẩm từ server (PHP)

Request request = new Request.Builder()  
 .url(SERVER\_URL)  
 .build();  
  
// Gửi request bất đồng bộ  
client.newCall(request).enqueue(new Callback() {...});

* Bật/tắt thiết bị (LED, quạt) thông qua Switch.

switchLed.setOnCheckedChangeListener((buttonView, isChecked) ->{  
 ivBulb.setImageResource(isChecked ? R.drawable.*ic\_bulb\_on* :R.drawable.*ic\_bulb\_off*);  
 controlDevice("led",isChecked ?1:0);  
});  
switchFan.setOnCheckedChangeListener((buttonView, isChecked) -> {  
 ivFan.setImageResource(isChecked ? R.drawable.*fan\_on* : R.drawable.*fan\_off*);  
 controlDevice("fan", isChecked ? 1 : 0);  
});

* Tự động điều khiển thiết bị dựa theo ngưỡng (logic thông minh)

boolean shouldTurnOnFan = temp > *TEMP\_HIGH\_THRESHOLD* || humid > *HUMID\_HIGH\_THRESHOLD*;  
boolean shouldTurnOnLed = temp > *TEMP\_HIGH\_THRESHOLD* || temp < *TEMP\_LOW\_THRESHOLD*;  
  
// Bật/tắt quạt  
if (shouldTurnOnFan) {  
 if (!switchFan.isChecked()) {  
 switchFan.setChecked(true); // sẽ kích hoạt listener để gửi lệnh  
 }  
} else {  
 if (switchFan.isChecked()) {  
 switchFan.setChecked(false);  
 }  
}  
  
// Bật/tắt đèn  
if (shouldTurnOnLed) {  
 if (!switchLed.isChecked()) {  
 switchLed.setChecked(true);  
 }  
} else {  
 if (switchLed.isChecked()) {  
 switchLed.setChecked(false);  
 }  
}

* Gửi thông báo khi vượt ngưỡng an toàn

if (temp > *TEMP\_HIGH\_THRESHOLD*) {  
 message = "Nhiệt độ quá cao: " + temp + "°C";  
} else if (temp < *TEMP\_LOW\_THRESHOLD*) {  
 message = "Nhiệt độ quá thấp: " + temp + "°C";  
} else if (humid > *HUMID\_HIGH\_THRESHOLD*) {  
 message = "Độ ẩm quá cao: " + humid + "%";  
} else if (humid < *HUMID\_LOW\_THRESHOLD*) {  
 message = "Độ ẩm quá thấp: " + humid + "%";  
}  
  
if (message != null) {  
 sendNotification(message);  
}

* Chuyển trang đến các Activity khác (Settings, Data, Admin)

navHome.setOnClickListener(v->{  
 //MainActivity  
});  
navSetting.setOnClickListener(v ->{  
 Intent intent= new Intent(MainActivity.this,SettingsActivity.class);  
 startActivity(intent);  
});  
navData.setOnClickListener(v ->{  
 Intent intent = new Intent(MainActivity.this, DataActivity.class);  
 startActivity(intent);  
});  
navManageUsers.setOnClickListener(v -> {  
 // mở màn hình quản lý tài khoản hoặc fragment tương ứng  
 Intent intent = new Intent(MainActivity.this, AdminActivity.class);  
 startActivity(intent);  
});

* Chạy SensorMonitorService để xử lý nền

Intent serviceIntent = new Intent(this, SensorMonitorService.class);  
startService(serviceIntent);

BluetoothActivity là Activity cho phép người dùng bật/tắt Bluetooth, quét thiết bị lân cận, kết nối với thiết bị ESP32 qua Bluetooth, đọc dữ liệu cảm biến nhiệt độ/độ ẩm, và hiển thị dữ liệu đó lên giao diện

- Quy trình bật tắt Bluetooth

switchBluetooth.setChecked(bluetoothAdapter.isEnabled());  
switchBluetooth.setOnCheckedChangeListener((buttonView, isChecked) -> {  
 if (isChecked) {  
 if (!bluetoothAdapter.isEnabled()) {  
 Intent enableBtIntent = new Intent(BluetoothAdapter.*ACTION\_REQUEST\_ENABLE*);  
 startActivityForResult(enableBtIntent, REQUEST\_ENABLE\_BT);  
 }  
 } else {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*TIRAMISU*) {  
 Intent intent = new Intent(Settings.*ACTION\_BLUETOOTH\_SETTINGS*);  
 startActivityForResult(intent, *REQUEST\_DISABLE\_BT*);  
 Toast.*makeText*(this, "Vui lòng tắt Bluetooth thủ công, sau đó quay lại", Toast.*LENGTH\_LONG*).show();  
 } else {  
 bluetoothAdapter.disable();  
 disableBluetooth();  
 }  
 }  
});

* Quét và hiển thị thiết bị khả dụng

bluetoothAdapter.startDiscovery();

registerReceiver(bluetoothReceiver, new IntentFilter(BluetoothDevice.*ACTION\_FOUND*));

- Kết nối đến thiết bị khi người dùng chọn

bluetoothSocket = device.createRfcommSocketToServiceRecord(*MY\_UUID*);  
bluetoothAdapter.cancelDiscovery();  
bluetoothSocket.connect();  
BluetoothConnectionManager.*setSocket*(bluetoothSocket);

- Đọc dữ liệu từ thiết bị Bluetooth (ESP32)

InputStream inputStream = bluetoothSocket.getInputStream();

String incomingMessage = new String(buffer, 0, bytes);

- Tách và hiển thị nhiệt độ/độ ẩm

String[] parts = fullData.split(",");  
String tempPart = parts[0].split(":")[1].replace("\*C", "").trim();  
String humidPart = parts[1].split(":")[1].replace("%", "").trim();

- Hủy ghép nối với thiết bị đã paired

java.lang.reflect.Method method = device.getClass().getMethod("removeBond");  
method.invoke(device);

DataActivity là Activity có nhiệm vụ hiển thị biểu đồ nhiệt độ và độ ẩm dựa trên dữ liệu thu được từ server thông qua cảm biến DHT11 gắn trên ESP32. Các bước xử lý chính của Activity này gồm:

* Gửi yêu cầu HTTP GET tới server PHP (get\_all\_data.php). Nhận dữ liệu JSON chứa các giá trị nhiệt độ, độ ẩm và thời gian.

String dataUrl="http://172.20.10.8/BTL\_DHT11/get\_all\_data.php";

private void fetchChartData(){  
 new Thread(()->{  
 try {  
 URL url= new URL(dataUrl);  
 HttpURLConnection conn= (HttpURLConnection) url.openConnection();  
 conn.setRequestMethod("GET");  
 BufferedReader in= new BufferedReader(new InputStreamReader(conn.getInputStream()));  
 StringBuilder response= new StringBuilder();  
 String line;  
 while ((line = in.readLine()) != null) response.append(line);  
 in.close();  
  
 JSONObject json= new JSONObject(response.toString());  
 if (json.getString("status").equals("success")){  
 JSONArray dataArray = json.getJSONArray("data");  
  
 ArrayList<Entry> tempEntries = new ArrayList<>();  
 ArrayList<Entry> humEntries = new ArrayList<>();  
  
 int total = dataArray.length();  
 int startIndex = Math.*max*(0, total - 20); // ✅ Chỉ lấy 20 điểm cuối  
 List<String> timestamps = new ArrayList<>();  
  
 for (int i = startIndex; i < total; i++) {  
 JSONObject obj = dataArray.getJSONObject(i);  
 float temp = (float) obj.getDouble("temperature");  
 float hum = (float) obj.getDouble("humidity");  
 String time = obj.getString("timestamp");  
 timestamps.add(time); // dùng cho marker  
  
 tempEntries.add(new Entry(i - startIndex, temp)); // index từ 0  
 humEntries.add(new Entry(i - startIndex, hum));  
 }  
  
 runOnUiThread(() -> {  
 showLineChart(lineChartTemp, tempEntries, "Nhiệt độ (°C)", "#FF5722", timestamps);  
 showLineChart(lineChartHum, humEntries, "Độ ẩm (%)", "#2196F3", timestamps);  
 });  
 }else {  
 showToast("Không có dữ liệu!");  
 }  
 } catch (Exception e){  
 e.printStackTrace();  
 showToast("Lỗi lấy dữ liệu từ server");  
 }  
 }).start();  
}

* Chuyển dữ liệu thành Entry để vẽ biểu đồ bằng thư viện MPAndroidChart.

for (int i = startIndex; i < total; i++) {  
 JSONObject obj = dataArray.getJSONObject(i);  
 float temp = (float) obj.getDouble("temperature");  
 float hum = (float) obj.getDouble("humidity");  
 String time = obj.getString("timestamp");  
 timestamps.add(time); // dùng cho marker  
  
 tempEntries.add(new Entry(i - startIndex, temp)); // index từ 0  
 humEntries.add(new Entry(i - startIndex, hum));  
}

* Tự động cập nhật biểu đồ mỗi 5 giây.

handler.postDelayed(this, 5000);

* Cung cấp nút quay lại màn hình chính (MainActivity).

btnBack.setOnClickListener(v->{  
 Intent intent = new Intent(DataActivity.this,MainActivity.class);  
 startActivity(intent);  
});

AdminActivity là Activity chỉ dành cho tài khoản admin, cho phép quản lý danh sách tài khoản, cấp/quay lại quyền admin và xóa tài khoản (trừ admin gốc).

* Định nghĩa đối tượng Account với các thuộc tính id, username, password, role, email, từ đó dùng để ánh xạ dữ liệu JSON trả về từ server thành đối tượng Java.

JSONArray users = resJson.getJSONArray("data");  
accountList.clear();  
for (int i = 0; i < users.length(); i++) {  
 JSONObject u = users.getJSONObject(i);  
 Account account = new Account(  
 u.getInt("id"),  
 u.getString("username"),  
 "", // Không có password trả về từ JSON  
 u.getString("role"),  
 u.optString("email", "")  
 );  
 accountList.add(account);  
}

* AccountAdapter.java dùng cho RecyclerView.Adapter hiển thị danh sách tài khoản. Đồng thời xử lý từng tài khoản với các tác vụ như xóa tài khoản gửi id lên server qua delete\_user.php.Cập nhật quyền, gửi id và quyền mới (admin hoặc guest) lên update\_user\_role.php.
* Các API PHP sử dụng, file get\_all\_user.php giúp trả về danh sách toàn bộ tài khoản dưới dạng JSON, hàm update\_user\_role.php để Nhận id, role, và cập nhật trên cơ sở dữ liệu, hàm delete\_user.php để Nhận id, xóa tài khoản khỏi bảng users.
* Xử lý hạn chế đặc biệt như không cho phép tài khoản admin gốc (id là admin) bị xóa. Nếu tài khoản hiện tại là admin thường (không phải admin gốc) thì không thể xóa hoặc hạ cấp tài khoản admin khác.
* Khi vào AdminActivity, ứng dụng gọi API get\_all\_user.php và hiển thị danh sách. Người dùng admin có thể bấm Tạo tài khoản để mở CreateUserActivity, chọn Cập nhật quyền để luân phiên giữa admin và guest, bấm Xóa để xóa tài khoản (nếu đủ quyền).Sau khi thao tác, dữ liệu được cập nhật lại để hiển thị trạng thái mới nhất.

#### 2.2.4.3.Các class hỗ trợ

Account.java giúp định nghĩa mô hình dữ liệu cho tài khoản người dùng gồm id, username, password, role, email.

public Account(int id, String username, String password, String role, String email) {  
 this.id = id;  
 this.username = username;  
 this.password = password;  
 this.role = role;  
 this.email = email;  
}

AccountAdapter.java là Adapter hiển thị danh sách tài khoản trong AdminActivity bằng RecyclerView, cho phép xử lý sự kiện như cập nhật quyền, xóa tài khoản.

@Override  
public void onBindViewHolder(@NonNull AccountViewHolder holder, int position) {  
 Account account = accountList.get(position);  
  
 holder.tvUsername.setText("Tên đăng nhập: " + account.getUsername());  
 holder.tvPassword.setText("Mật khẩu: " + account.getPassword());  
 holder.tvRole.setText("Quyền: " + account.getRole());  
  
 holder.btnToggleRole.setText(account.getRole().equals("admin") ? "Hạ quyền" : "Cấp quyền");  
  
 // Không cho phép admin gốc xóa chính mình  
 boolean isAdminRoot = "admin".equals(currentAdminUsername);  
  
 boolean isOtherAdmin = account.getRole().equals("admin") && !account.getUsername().equals("admin");  
  
 holder.btnDelete.setEnabled(isAdminRoot || !account.getRole().equals("admin"));  
  
 holder.btnToggleRole.setOnClickListener(v -> {  
 if (account.getUsername() != null && !"admin".equals(account.getUsername())) {  
 listener.onToggleRole(account);  
 } else {  
 Toast.*makeText*(context, "Không thể thay đổi quyền admin gốc", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 holder.btnDelete.setOnClickListener(v -> {  
 if (!account.getUsername().equals("admin")) {  
 listener.onDelete(account);  
 } else {  
 Toast.*makeText*(context, "Không thể xóa tài khoản admin gốc", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
}

BluetoothConnectionManager.java là Class singleton trung gian lưu và quản lý BluetoothSocket khi kết nối thành công, phục vụ truyền nhận dữ liệu giữa các Activity.

NotificationService là class cung cấp thông báo cảnh báo (nhiệt độ, độ ẩm…) cho người dùng, đồng thời lưu lại thông báo tương ứng theo tài khoản đăng nhập vào SharedPreferences

* Tạo foreground notification có nội dung cảnh báo.

startForeground(1, notification);

* Lưu thông báo dưới dạng văn bản theo từng tài khoản giúp NotificationsActivity hiển thị lịch sử thông báo theo từng người dùng.

if (username != null && message != null) {  
 SharedPreferences prefs = getSharedPreferences("MyNotifications", *MODE\_PRIVATE*);  
 String existing = prefs.getString(username, "");  
 String updated = existing + "- " + message + "\n";  
 prefs.edit().putString(username, updated).apply();  
}

* Kích hoạt rung thiết bị để cảnh báo.

Vibrator v = (Vibrator) getSystemService(*VIBRATOR\_SERVICE*);  
if (v != null && v.hasVibrator()) {  
 v.vibrate(VibrationEffect.*createOneShot*(1000, VibrationEffect.*DEFAULT\_AMPLITUDE*));  
}

* Tự động dừng service sau khi gửi thông báo.

stopSelf();

NotificationAdapter là Adapter cho RecyclerView giúp hiển thị danh sách các thông báo dưới dạng danh sách cuộn (RecyclerView) trong NotificationsActivity.

SensorMonitorService là loại Foreground Service. Cứ mỗi 3 giây, Service gửi yêu cầu HTTP GET để lấy dữ liệu nhiệt độ và độ ẩm từ ESP32 gửi lên. Dữ liệu được phân tích và nếu phát hiện vượt ngưỡng nguy hiểm, hệ thống sẽ tự động gọi NotificationService để gửi cảnh báo đến người dùng. Service này vẫn chạy kể cả khi ứng dụng bị đóng nhờ startForeground(), đảm bảo quá trình giám sát liên tục và ổn định.

Class CustomMarkerView là một lớp tùy chỉnh kế thừa từ MarkerView trong thư viện MPAndroidChart. Lớp này dùng để hiển thị chi tiết giá trị nhiệt độ/độ ẩm kèm theo thời gian tương ứng khi người dùng tương tác với biểu đồ. Thông tin được hiển thị dưới dạng popup nhỏ, được đặt chính xác tại điểm dữ liệu đang được chạm vào. Đây là một tính năng nâng cao giúp tăng khả năng tương tác và trực quan hóa dữ liệu trong ứng dụng.

@Override  
public void refreshContent(Entry e, Highlight highlight) {  
 int index = (int) e.getX();  
 String time = (index < timestamps.size()) ? timestamps.get(index) : "N/A";  
 String value = String.*format*("%.1f", e.getY());  
  
 tvValue.setText("Giá trị: " + value + "\nThời gian: " + time);  
 super.refreshContent(e, highlight);  
}

#### 2.2.4.4.Thiết kế các layout, giao diện chính

activity\_login.xml được thiết kế cho tab LoginActivity, bao gồm mục để điền thông tin tài khoản người dùng, các chức năng như quên mật khẩu, đăng nhập, bluetooth, quên mật khẩu.

A screenshot of a phone

AI-generated content may be incorrect.

Hình 2.12: Thiết kế giao diện tại tab LoginActivity

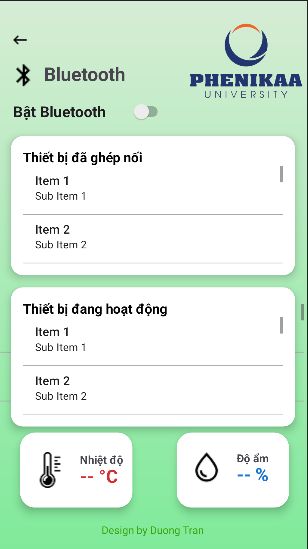
activity\_main.xml được thiết kế cho tab chính MainActivity, bao gồm hiển thị nhiệt độ, độ ẩm, điều khiển switch bật tắt đèn và quạt, hiển thị thanh chọn tab phía dưới.

Screens screenshot of a phone

AI-generated content may be incorrect.

Hình 2.13: Thiết kế giao diện cho tab MainActivity

activity\_bluetooth.xml được thiết kế cho tab BluetoothActivity, có switch bật tắt bluetooth, hiển thị thiết bị đã kết nối và thiết bị đang hoạt động, đồng thời sử dụng switch bật tắt đèn và quạt



Hình 2.14: Thiết kế giao diện cho tab BluetoothActivity

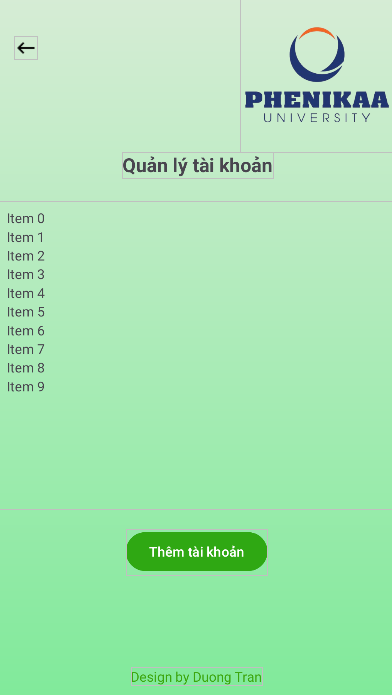
Activity\_data.xml được thiết kế cho tab DataActivity, nhằm mục đích hiển thị biểu đồ hiển thị nhiệt độ, độ ẩm.

A screenshot of a phone

AI-generated content may be incorrect.

Hình 2.15: Thiết kế giao diện cho tab DataActivity

activity\_admin.xml được thiết kế cho tab AdminActivity, hiển thị danh sách các tài khoản, cùng với nút button để thêm tài khoản



Hình 2.16: Thiết kế giao diện cho tab AdminActivity

activity\_register.xml được thiết kế dành cho tab RegisterActivity, có các mục nhập thông tin tên đăng nhập, mật khẩu và email, đồng thời có nút button nhấn Đăng ký, nếu đã có tài khoản thì nhấn “Đã có tài khoản” để trở về tab LoginActivity

A screenshot of a phone

AI-generated content may be incorrect.

Hình 2.17: Thiết kế giao diện cho tab RegisterActivity

### 2.2.5.Lưu đồ thuật toán

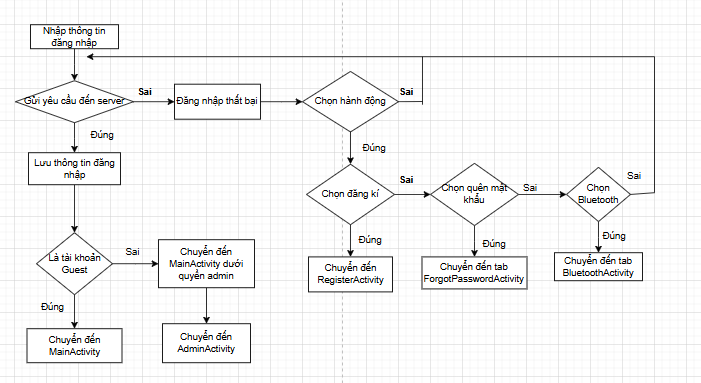
Lưu đồ thuật toán con StartActivity

A diagram of a process

AI-generated content may be incorrect.

Hình 2.18: Lưu đồ thuật toán con StartActivity

Lưu đồ thuật toán con LoginActivity



Hình 2.19: Lưu đồ thuật toán con LoginActivity

Lưu đồ thuật toán con RegisterActivity

A diagram of a computer program

AI-generated content may be incorrect.

Hình 2.20: Lưu đồ thuật toán con RegisterActivity

Lưu đồ thuật toán con ForgotPasswordActivity

A diagram of a computer program

AI-generated content may be incorrect.

Hình 2.21: Lưu đồ thuật toán con ForgotPasswordActivity

Lưu đồ thuật toán con BluetoothActivity

A diagram of a bluetooth connection

AI-generated content may be incorrect.

Hình 2.22: Lưu đồ thuật toán con BluetoothActivity

Lưu đồ thuật toán con MainActivity

A diagram of a system

AI-generated content may be incorrect.

Hình 2.23: Lưu đồ thuật toán MainActivity

Lưu đồ thuật toán con AdminActivity

A diagram of a work flow

AI-generated content may be incorrect.

Hình 2.24: Lưu đồ thuật toán con AdminActivity

Lưu đồ thuật toán con SensorMonitorService

A diagram of a system

AI-generated content may be incorrect.

Hình 2.25: Lưu đồ thuật toán con SensorMonitorService

Lưu đồ thuật toán này mô tả cách hoạt động của App, khi mở App sẽ thực hiện bắt đầu chương trình.

Người dùng chọn chế độ kết nối:

* Hệ thống sẽ kiểm tra trạng thái hoạt động của WiFi, nếu WiFi không hoạt động thì đến bước chọn tab Bluetooth.Nếu WiFi đang hoạt động, tiếp tục các bước sau.
* Nếu chọn tab Bluetooth, hệ thống bật kết nối Bluetooth và hiển thị nhiệt độ, độ ẩm từ cảm biến. Nếu không thì hệ thông sẽ quay lại bước kiểm tra trạng thái hoạt động của Wifi

Chọn hình thức đăng nhập:

* Nếu người dùng chưa có tài khoản, chọn đăng ký, nhập thông tin để đăng ký tài khoản mới.
* Nếu quên mật khẩu, chọn tính năng "Lấy lại mật khẩu" để hệ thống gửi mật khẩu mới.

Nhập thông tin đăng nhập:

* Hệ thống sẽ kiểm tra thông tin đăng nhập, nếu sai thì sẽ yêu cầu nhập lại, nếu đúng thì hệ thống sẽ xác định quyền của người dùng.

Phân quyền tài khoản:

* Nếu là tài khoản admin sẽ truy cập vào màn hình quản trị để thêm tài khoản mới, xóa tài khoản, cấp quyền admin cho user khác
* Nếu là tài khoản guest, truy cập vào màn hình chính.

Màn hình chính - Đọc dữ liệu từ server:

* Hệ thống đọc dữ liệu nhiệt độ và độ ẩm từ cơ sở dữ liệu mà ESP32 đã gửi trước đó.

Kiểm tra điều kiện nhiệt độ, độ ẩm:

* Nếu nhiệt độ ngoài ngưỡng cho phép < 10°C hoặc > 35°C hoặc độ ẩm ngoài ngưỡng < 20% hoặc > 70% thì rung điện thoại, gửi thông báo về app.
* Nếu trong ngưỡng an toàn thì cho phép người dùng bật/tắt đèn/quạt thủ công.

Kết thúc tiến trình.

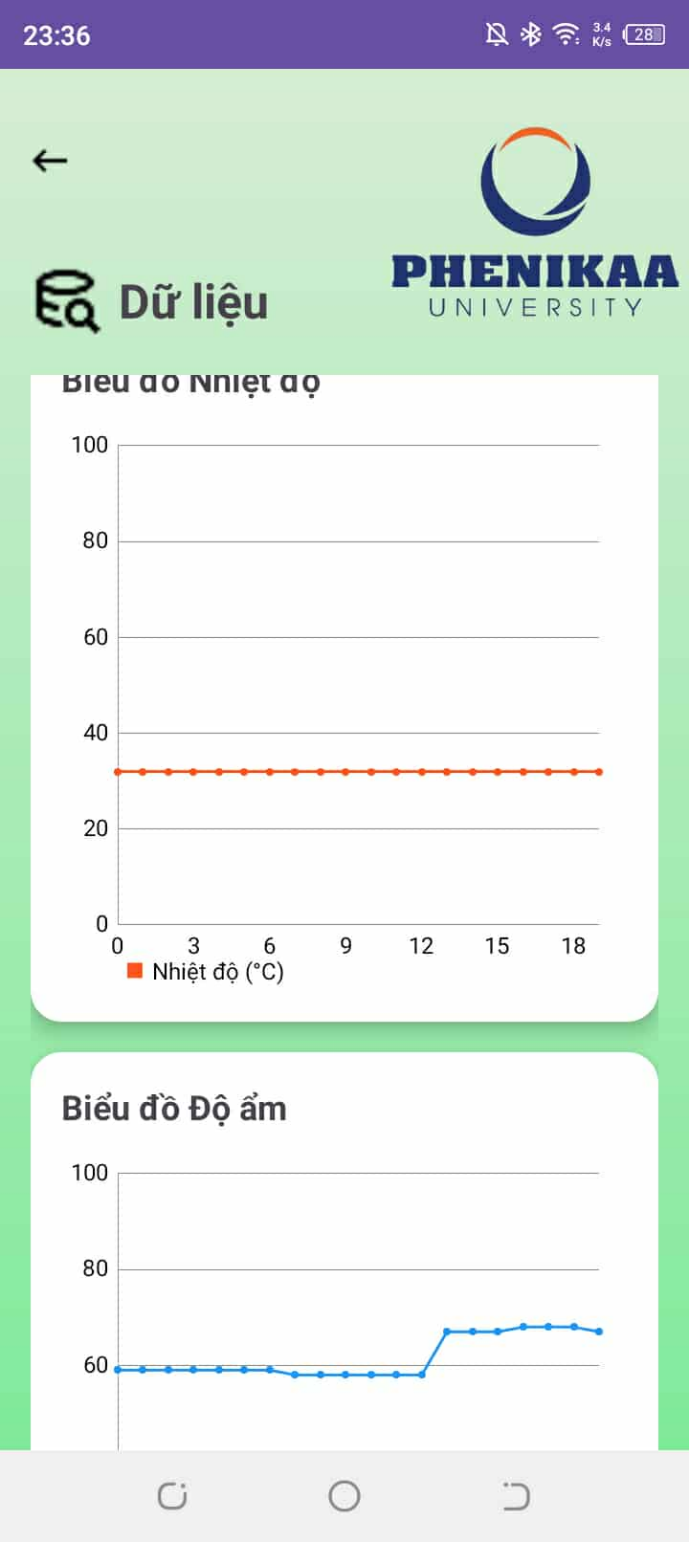
A diagram of a flowchart

AI-generated content may be incorrect.

Hình 2.26: Lưu đồ thuật toán của app

## 2.3.Chức năng ứng dụng

### 2.3.1.Hiển thị dữ liệu



Hình 2.27: Giao diện hiển thị dữ liệu dưới dạng đồ thị

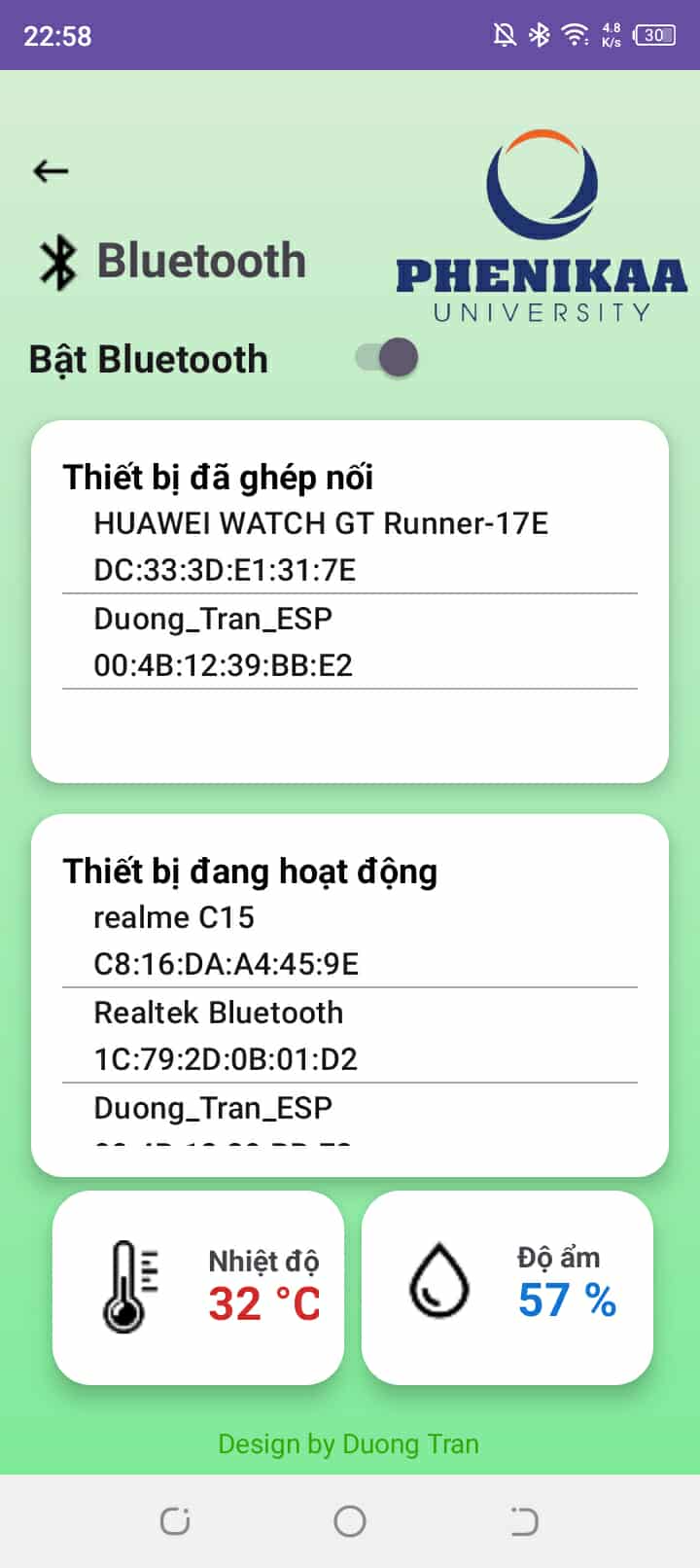
Dữ liệu đo được từ DHT11 sẽ được hiển thị realtime trên đồ thị giúp trực quan hóa dữ liệu, ngoài ra tại tab MainActivity sẽ hiển thị dữ liệu dưới dạng thông số cho người dùng quan sát

A screenshot of a device

AI-generated content may be incorrect.

Hình 2.28: Giao diện tab chính MainActivity

Ngoài ra việc hiển thị dữ liệu về nhiệt độ, độ ẩm còn được thể hiện trên tab BluetoothActivity thông qua kết nối Bluetooth. Tab BluetoothActivity hiển thị các thiết bị Bluetooth đã ghép nối và các thiết bị đang hoạt động.



Hình 2.29: Giao diện tab BluetoothActivity

### 2.3.2.Tương tác và điều khiển

Cho phép người dùng bật tắt đèn và quạt trên app

A screenshot of a device

AI-generated content may be incorrect.

Hình 2.30: Tương tác và điều khiển thiết bị

A circuit board with wires and lights

AI-generated content may be incorrect.

Hình 2.31: Mạch khi chạy thực tế

### 2.3.3.Các chức năng khác

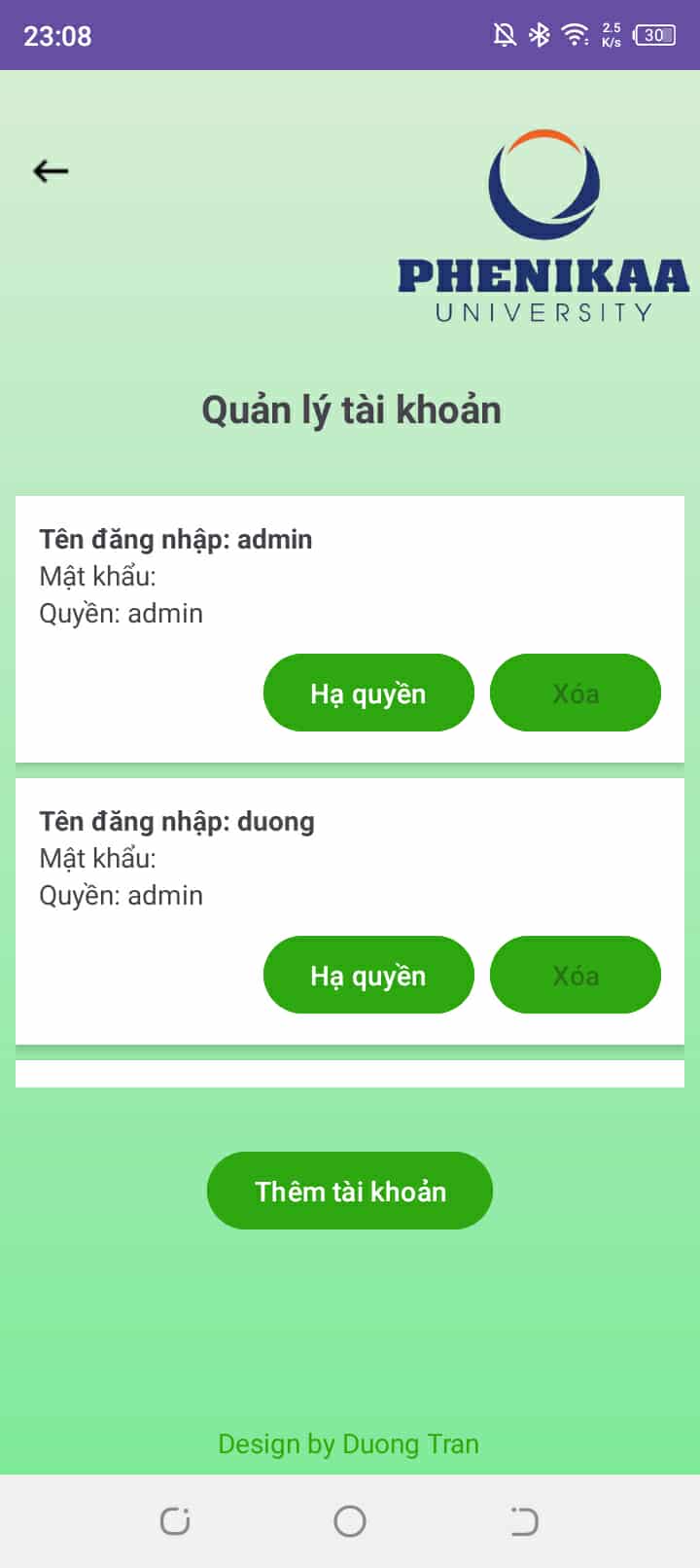
Tab SettingActivity hiển các cài đặt người dùng cần thiết

A screenshot of a phone

AI-generated content may be incorrect.

Hình 2.32: Giao diện tab Setting Activity

Tab AdminActivity là nơi để thực hiện quyền quản trị viên



Hình 2.33: Giao diện thiết kế tab AdminActivity

### 2.3.3.Phương pháp đánh giá kiểm thử

#### 2.3.3.1.Kiểm thử chức năng

Kiểm tra các phương thức xử lý dữ liệu cảm biến từ ESP32 (nhiệt độ, độ ẩm) để đảm bảo hiển thị đúng trên biểu đồ.

Giao diện người dùng tương thức được với các giao diện màn hình điện thoại android cơ bản

Kiểm tra luồng kết nối từ Android đến ESP32 qua Wifi và Bluetooth, từ đó nhận dữ liệu DHT11 và hiển thị lên giao diện của ứng dụng

Kiểm thử sự tương tác giữa các module trong ứng dụng, các phần có sự hoạt động tốt, kết nối với nhau như thiết kế

#### 2.3.3.2.Kiểm thử hiệu năng

Load Testing:

* Mô phỏng việc gửi liên tục nhiều lệnh điều khiển từ app đến ESP32 và quan sát khả năng xử lý phản hồi của thiết bị
* Kiểm tra khả năng nhận và xử lý dòng xử liệu cảm biến liên tục từ ESP32 mà không gây lag ứng dụng

Stress Testing:

* Tạo điều kiện gửi/nhận dữ liệu với tần suất cao hoặc kéo dài để kiểm tra xem app có bị đơ, treo hoặc gặp lỗi khi bị quá tải không

#### 2.3.3.3.Kiểm thử độ ổn định

Long-Run Testing:

* Chạy ứng dụng liên tục trong vòng vài tiếng để theo dõi việc cập nhật dữ liệu cảm biến và trạng thái thiết bị, kiểm tra có xuất hiện memory leak hoặc treo app không.

#### 2.3.3.4.Kiểm tra tính tương thích

Device Testing:

* Kiểm thử trên nhiều thiết bị Android từ Android 6.0 đến Android 13, cả điện thoại có cấu hình thấp và cao, đảm bảo app vẫn hoạt động ổn định.

Screen Size Testing:

* Kiểm tra hiển thị của ứng dụng trên các kích thước màn hình khác nhau (smartphone nhỏ, lớn, máy tính bảng), đảm bảo biểu đồ, nút điều khiển không bị lệch giao diện hoặc tràn lề.

## 2.4.Kết luận chương

Chương II đã cơ bản trình bày về thiết kế hệ thống, từ lý thuyết đến triển khai đề tài, từ sơ đồ khối, đến cấu trúc ứng dụng, tính năng của ứng dụng. Đồng thời nêu rõ các thiết bị phần cứng mà dự án sử dụng, các phần mềm mà dự án áp dụng. Lưu đồ thuật toán và chức năng ứng dụng, đồng thời kiểm thử ứng dụng đã được nêu ra. Tiếp theo là chương III sẽ trình bày về kết quả và đánh giá của toàn dự án

# CHƯƠNG III: KẾT QUẢ VÀ ĐÁNH GIÁ

## 3.1.Tiến độ thực hiện

Dự án đạt đúng kịp tiến độ đã đề ra, thiết kế các chức năng cơ bản của app. Tuy nhiên, app vẫn còn thiếu một số tính năng nâng cao do thời gian thực hiện dự án còn ngắn.

## 3.2.Kết quả đạt được

Kết quả: ứng dụng đã thực hiện cập nhật dữ liệu trực tiếp từ cảm biến, đồng thời người dùng có thể điều khiển thủ công, tự động các thiết bị IoT. Ngoài ra còn có các chức năng thông báo khi nhiệt độ, độ ẩm vượt ngưỡng, các chức năng đăng kí tài khoản, quên mật khẩu, tài khoản admin, kết nối Bluetooth riêng

So sánh kết quả với mục tiêu đề ra: sử dụng thuật toán PID trực tiếp trên app

## 3.3.Kết luận

### 3.3.1.Ưu điểm

Giao tiếp dễ dàng qua HTTP, không cần cấu hình phức tạp

Giá thành thấp: ESP32 (~100.000đ), DHT11 (~20.000đ)

Ứng dụng Android có giao diện dễ dùng, chia quyền rõ ràng, thông báo trực quan

Có thể mở rộng nhiều cảm biến hoặc kết nối Internet

Bảo mật cơ bản, dùng SharedPreferences, kiểm tra mật khẩu cũ khi đổi tài khoản

Kết nối Bluetooth

### 3.3.2.Nhược điểm

Chưa có xác thực mạnh (JWT, token...), dùng SharedPreferences đơn giản, không có mã hóa nâng cao

Không truy cập từ xa (Internet), hệ thống chỉ hoạt động trong mạng LAN/local WiFi

Cảnh báo phụ thuộc chạy nền Android, một số dòng máy có thể giới hạn dịch vụ chạy nền (battery saving)

### 3.3.3.Giải pháp khắc phục

Tích hợp đăng nhập có mã hóa, dùng JWT hoặc Firebase Authentication

Kết hợp Firebase Realtime DB, hoặc mở port public qua Cloud Server

## 3.4.Hướng phát triển

Đưa hệ thống lên Internet , dùng Firebase hoặc hosting MySQL trên server cloud như Hostinger, AWS...

Lưu và phân tích dữ liệu lịch sử, cho phép người dùng xem biểu đồ theo ngày/tháng

Tích hợp AI dự đoán, dùng machine learning để cảnh báo sớm nguy cơ quá nhiệt, quá ẩm

Bảo mật cao hơn, sử dụng HTTPS, mã hóa SHA256 hoặc JWT, xác thực hai lớp

Hỗ trợ nhiều cảm biến hơn, mở rộng để đọc pH nước, ánh sáng, chuyển động... phục vụ nông nghiệp

Tự động điều khiển thiết bị, tự bật quạt, bật đèn khi vượt ngưỡng nhiệt độ/độ ẩm

# TÀI LIỆU THAM KHẢO

## 1.Tài liệu tham khảo tiếng Việt

[1] Lê Chi, “Android Studio là gì? Hướng dẫn tải và cài đặt Android Studio” 2023. [Trực tuyến]. Available: https://fptshop.com.vn/tin-tuc/thu-thuat/android-studio-la-gi-150412. [Accessed 29 07 2025].

[2] "Xampp là gì? Cách sử dụng và cài đặt XAMPP" Top Dev, [Online]. Available: https://topdev.vn/blog/cai-dat-xampp/. [Accessed 29 07 2025].

[3] "Phần mềm Arduino IDE là gì? Chi tiết nhất" Got It, [Online]. Available: https://vn.got-it.ai/blog/phan-mem-arduino-ide-la-gi-chi-tiet-nhat. [Accessed 29 07 2025].

[4] Đỗ Đạt, "Tìm hiểu về HTTP (HyperText Transfer Protocol)" Viblo, 17 09 2018. [Online]. Available: https://viblo.asia/p/tim-hieu-ve-http-hypertext-transfer-protocol-bJzKmgewl9N. [Accessed 29 07 2025].

[5] "Đèn Halogen Philips HS1 35w" [Online]. Available: https://ledt2.com/halogen-philips-hs1-35w?. [Accessed 29 07 2025].

[6] "Quạt mini 30mm DC 5v 0.2A dùng cho cả Orange Pi, Raspberry Pi" Orangepi.vn, [Online]. Available: https://orangepi.vn/shop/quat-mini-24mm-dc-5v-0-2a-dung-cho-ca-orange-pi-raspberry-pi. [Accessed 29 07 2025].

## 2.Tài liệu tham khảo tiếng Anh

[7] Espressif Systems (Shanghai) Co., Ltd., "esp32\_datasheet\_en.pdf" 2025. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32\_datasheet\_en.pdf. [Accessed 29 07 2025].

[8] ETC, "DHT11 datasheet (PDF)," [Online]. Available: https://www.alldatasheet.vn/datasheet-pdf/pdf/1440068/ETC/DHT11.html. [Accessed 29 07 2025].

[9] Handson Technology, "1 Channel 5V Optical Isolated Relay Module," [Online]. Available: https://handsontec.com/dataspecs/relay/1Ch-relay.pdf. [Accessed 29 07 2025].

1. Lê Chi, “Android Studio là gì? Hướng dẫn tải và cài đặt Android Studio” 2023. [Trực tuyến]. Available: https://fptshop.com.vn/tin-tuc/thu-thuat/android-studio-la-gi-150412. [Đã truy cập 29 07 2025]. [↑](#footnote-ref-1)
2. "Xampp là gì? Cách sử dụng và cài đặt XAMPP" Top Dev, [Online]. Available: https://topdev.vn/blog/cai-dat-xampp/. [Accessed 29 07 2025]. [↑](#footnote-ref-2)
3. "Phần mềm Arduino IDE là gì? Chi tiết nhất" Got It, [Online]. Available: https://vn.got-it.ai/blog/phan-mem-arduino-ide-la-gi-chi-tiet-nhat. [Accessed 29 07 2025]. [↑](#footnote-ref-3)
4. Đỗ Đạt, "Tìm hiểu về HTTP (HyperText Transfer Protocol)" Viblo, 17 09 2018. [Online]. Available: https://viblo.asia/p/tim-hieu-ve-http-hypertext-transfer-protocol-bJzKmgewl9N. [Accessed 29 07 2025]. [↑](#footnote-ref-4)
5. Đỗ Đạt, "Tìm hiểu về HTTP (HyperText Transfer Protocol)" Viblo, 17 09 2018. [Online]. Available: https://viblo.asia/p/tim-hieu-ve-http-hypertext-transfer-protocol-bJzKmgewl9N. [Accessed 29 07 2025]. [↑](#footnote-ref-5)
6. Đỗ Đạt, "Tìm hiểu về HTTP (HyperText Transfer Protocol)" Viblo, 17 09 2018. [Online]. Available: https://viblo.asia/p/tim-hieu-ve-http-hypertext-transfer-protocol-bJzKmgewl9N. [Accessed 29 07 2025]. [↑](#footnote-ref-6)
7. [7] Espressif Systems (Shanghai) Co., Ltd., "esp32\_datasheet\_en.pdf" 2025. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32\_datasheet\_en.pdf. [Accessed 29 07 2025]. [↑](#footnote-ref-7)
8. ETC, "DHT11 datasheet (PDF)," [Online]. Available: https://www.alldatasheet.vn/datasheet-pdf/pdf/1440068/ETC/DHT11.html. [Accessed 29 07 2025]. [↑](#footnote-ref-8)
9. "Đèn Halogen Philips HS1 35w" [Online]. Available: https://ledt2.com/halogen-philips-hs1-35w?. [Accessed 29 07 2025]. [↑](#footnote-ref-9)
10. "Quạt mini 30mm DC 5v 0.2A dùng cho cả Orange Pi, Raspberry Pi" Orangepi.vn, [Online]. Available: https://orangepi.vn/shop/quat-mini-24mm-dc-5v-0-2a-dung-cho-ca-orange-pi-raspberry-pi. [Accessed 29 07 2025]. [↑](#footnote-ref-10)
11. Handson Technology, "1 Channel 5V Optical Isolated Relay Module," [Online]. Available: https://handsontec.com/dataspecs/relay/1Ch-relay.pdf. [Accessed 29 07 2025]. [↑](#footnote-ref-11)