

**TRƯỜNG ĐH SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG**



**ĐỒ ÁN MÔN HỌC 1
XÂY DỰNG ỨNG DỤNG DI ĐỘNG ĐIỀU KHIỂN
THIẾT BỊ TRONG GIA ĐÌNH SỬ DỤNG ESP8266**

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

GVHD : ThS. TRƯỜNG NGỌC HÀ

Sinh viên : NGUYỄN THỊ NHẬT BĂNG

MSSV: 19119013

LÊ NGUYỄN HOÀNG VINH

MSSV: 19119148

TP. Hồ Chí Minh, tháng 05 năm 2023

TRƯỜNG ĐH SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG



ĐỒ ÁN MÔN HỌC 1
XÂY DỰNG ỨNG DỤNG DI ĐỘNG ĐIỀU KHIỂN
THIẾT BỊ TRONG GIA ĐÌNH SỬ DỤNG ESP8266

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

GVHD : ThS. TRƯỜNG NGỌC HÀ

Sinh viên : **NGUYỄN THỊ NHẬT BĂNG**

MSSV: 19119013

LÊ NGUYỄN HOÀNG VINH

MSSV: 19119148

TP. Hồ Chí Minh, tháng 05 năm 2023

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM



Độc lập - Tự do - Hạnh phúc

BẢNG NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ tên	MSSV	Đánh giá	Điểm
Nguyễn Thị Nhật Băng	19119013		
Lê Nguyễn Hoàng Vinh	19119148		

NHẬN XÉT CỦA GIÁO VIÊN

1. Về nội dung đề tài và khối lượng thực hiện

.....
.....
.....
.....
.....
.....
.....
.....

2. Thuyết trình

.....
.....
.....
.....
.....
.....
.....
.....

TP. Hồ Chí Minh, ngày... tháng 5 năm 2023

Giảng viên nhận xét
(Ký & ghi rõ họ tên)

LỜI NÓI ĐẦU

Với sự phát triển công nghệ dẫn đến sự ra đời của mạng Internet. Mạng Internet là một trong những sản phẩm có giá trị hết sức lớn lao và ngày càng trở nên một công cụ không thể thiếu, là nền tảng chính cho sự truyền tải, trao đổi thông tin trên toàn cầu. Giờ đây, các thiết bị tự động hóa ngày càng được con người áp dụng vào cuộc sống sinh hoạt hàng ngày: chỉ cần có máy tính hay Smartphone kết nối Internet thì bạn có thể điều khiển những thiết bị nơi bạn đang sinh sống. Do đó một ngôi nhà thông minh không còn là ước mơ của con người nữa mà nó đã trở thành hiện thực hóa.

Xuất phát từ những lý do trên, nhóm em thực hiện đề tài “Xây dựng ứng dụng di động điều khiển thiết bị trong gia đình sử dụng esp8266 ” để vận dụng những kiến thức mà nhóm đã trải qua suốt khoảng thời gian học tập lý thuyết lẫn thực hành để áp dụng vào đồ án môn học 1.

LỜI CẢM ƠN

“Lời đầu tiên, nhóm chúng tôi xin gửi lời cảm ơn chân thành đến Thầy Trương Ngọc Hà, người đã đồng hành cùng nhóm chúng tôi xuyên suốt học phần Đò Án 1 trong việc giúp đỡ, đánh giá và nhận xét những hạn chế trong đồ án này”.

Học phần Đò Án 1 là môn học thú vị, bổ ích và có tính thực tế cao. Đảm bảo cung cấp đầy đủ kiến thức, kỹ năng, giúp sinh viên có thể ứng dụng vào thực tế. Tuy nhiên, do khả năng tiếp thu thực tế còn nhiều hạn hẹp, kiến thức lý thuyết chưa sâu rộng nhưng đó là kinh nghiệm quý báu hành trình trang đẻ nhóm phát triển cho đồ án tốt nghiệp và xa hơn là trong công việc ,sự nghiệp tương lai sau này. Mặc dù bản thân chúng em đã cố gắng hết sức hàng tuần nhưng chắc chắn đồ án này khó tránh khỏi những thiếu sót, kính mong thầy xem xét và góp ý để bài tiểu luận của nhóm được hoàn thiện và tốt hơn.

Chúng em xin chân thành cảm ơn!

Nhóm sinh viên thực hiện

Nguyễn Thị Nhật Băng – 19119013

Lê Nguyễn Hoàn Vinh – 19119148

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI.....	1
1.1. Giới thiệu.....	1
1.2. Mục tiêu.....	1
1.3. Giới hạn đề tài.....	3
1.4. Phương pháp nghiên cứu.....	3
1.5. Đối tượng và phạm vi nghiên cứu.....	3
1.6. Bố cục quyển báo cáo.....	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	6
2.1. Phần mềm sử dụng.....	6
2.1.1 Android Studio.....	6
2.1.2 Figma.....	7
2.1.3. Arduino IDE.....	8
2.1.4. Sơ đồ giải thuật.....	8
2.2. Phần cứng hiển thị.....	9
2.2.1. NODEMCU ESP8266.....	9
2.2.2. DHT11.....	10
2.2.3. LED.....	11
CHƯƠNG 3 : THIẾT KẾ FIGMA, GIAO DIỆN NGƯỜI DÙNG VỚI ANDROID STUDIO.....	12
3.1. Thiết kế Figma.....	12
3.2. Giao diện người dùng với Android Studio.....	12
3.2.1. Giao diện đăng ký, đăng nhập	13
3.2.2. Giao diện room.....	16
3.2.3. Giao diện livingroom, kitchenroom, bedroom.....	20
3.2.4. Giao diện profife.....	24
CHƯƠNG 4 : CHỨC NĂNG CỦA ÚNG DỤNG.....	25
4.1. Đăng ký, đăng nhập qua firebase.....	25
4.1.1. Kết nối firebase với Android Studio.....	25
4.1.2. Chức năng đăng ký qua firebase.....	28
4.1.3. Chức năng đăng ký qua firebase.....	32
4.2. Hiển thị ngày giờ thực tế	35
4.3. Đọc giá trị nhiệt độ, độ ẩm qua DHT11, led.....	37
4.3.1. Kết nối firebase với Android Studio.....	37
4.3.2. Kết nối firebase với Arduino IDE.....	39
4.4. Điều khiển hình ảnh qua Realtime Database.....	43
4.5. Đánh giá những kết quả đã thực hiện được.....	46
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	47
5.1. Kết luận.....	47
5.2. Hướng phát triển.....	47
TÀI LIỆU THAM KHẢO.....	48

DANH MỤC HÌNH

Hình 2.1: Giao diện phần mềm Android studio.....	6
Hình 2.2: Giao diện phần mềm Figma.....	7
Hình 2.3: Giao diện Arduino IDE.....	8
Hình 2.4: Esp8266.....	9
Hình 2.5: Led.....	11
Hình 3.1: Thiết kế Figma.....	12
Hình 3.2: Tạo mới project.....	13
Hình 3.3: Chọn Empty Activity.....	13
Hình 3.4: Đường dẫn project.....	14
Hình 3.5: Layout login.....	14
Hình 3.6: Layout register.....	15
Hình 3.7: Layout room.....	16
Hình 3.8: Tạo menu.....	17
Hình 3.9: Đặt tên menu.....	18
Hình 3.10: Layout menu.....	19
Hình 3.11: Giao diện Layout menu.....	19
Hình 3.12: Layout livingroom.....	20
Hình 3.13: Layout kitchenroom.....	21
Hình 3.14: Layout bedroom.....	22
Hình 3.15: Layout thanh seekbar.....	22
Hình 3.16: Layout profile.....	24

Hình 4.1: Chức năng authentication.....	25
Hình 4.2: Cách thức để chọn phương thức firebase.....	26
Hình 4.3: Cách thức để chọn phương thức firebase.....	27
Hình 4.4: Ánh xạ, khai báo các biến sử dụng ở file register.java.....	28
Hình 4.5: Bắt sự kiện cho 2 nút nhấn.....	28
Hình 4.6: Khai báo các file java để chuyển trang.....	29
Hình 4.7: Xét điều kiện cho các ô.....	30
Hình 4.8: Xét điều kiện ô password bằng ô confirm password.....	31
Hình 4.9: Các button bắt sự kiện, và các hàm sử dụng.....	32
Hình 4.10: Hàm đăng nhập bằng phương thức Authentication của firebase.....	33
Hình 4.11: Hàm kiểm tra đăng nhập.....	33
Hình 4.12: File setup cho hàm kiểm tra đăng nhập.....	34
Hình 4.13: TextClock hiển thị giờ thực tế.....	35
Hình 4.14: Khai báo TextClock.....	35
Hình 4.15: Hiển thị thời gian.....	35
Hình 4.16: Cách thức để chọn phương thức firebase.....	37
Hình 4.17: Kết nối Android Studio với Realtime database.....	38
Hình 4.18: Định nghĩa thư viện, wifi, firebase, chân esp8266.....	39
Hình 4.19: Định nghĩa chân esp của led, kết nối wifi, firebase.....	40
Hình 4.20: Đọc giá trị DHT11 và LED.....	40
Hình 4.21: Phòng livingroom.....	41
Hình 4.22: Phòng kitchenroom.....	41
Hình 4.23: Phòng bedroom.....	42

Hình 4.24: Vòng lặp chạy.....	42
Hình 4.25: Khai báo biến và ánh xạ.....	43
Hình 4.26: Cấu trúc cây trên firebase.....	44
Hình 4.27: Code chuyển đổi hình ảnh	

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu

Trong cuộc cách mạng công nghiệp hóa đang ngày càng phát triển, cũng như sự phổ biến của internet, máy tính và điện thoại thông minh làm cho đời sống con người ngày càng hoàn thiện. Các thiết bị tự động hóa ngày càng được con người áp dụng vào cuộc sống sinh hoạt hàng ngày của mỗi con người. Do đó một ngôi nhà thông minh không còn là ước mơ của con người nữa mà nó đã trở thành hiện thực hóa.

Một phần mềm điều khiển thiết bị trong nhà từ xa là một căn nhà được trang bị các hệ thống cảm biến và các thiết bị đồ dùng được liên kết với một phần mềm xử lý trung tâm. Các thiết bị đồ dùng này được xử lý tình huống được lập trình trước, hoặc là được điều khiển và giám sát từ xa nhằm mục đích phục vụ cho cuộc sống ngày càng tiện nghi, an toàn và góp phần sử dụng hợp lí các nguồn tài nguyên. Nhờ các cảm biến như cảm biến nhiệt độ, độ ẩm,... thì người dùng có thể theo dõi các trạng thái bên trong ngôi nhà để đưa ra các quyết định giải quyết một cách phù hợp.

Cùng với sự phát triển của các thiết bị điện tử cá nhân như máy tính bảng và điện thoại thông minh... thì cho phép người dùng cài đặt phần mềm để giám sát hoặc điều khiển ngôi nhà ở bất cứ đâu

1.2. Mục tiêu đề tài

Đề tài “Xây dựng ứng dụng di động điều khiển thiết bị trong gia đình sử dụng esp8266 ” được thực hiện nhằm thiết kế hệ thống ứng dụng có các tính năng sau:

❖ Các tính năng cơ bản:

- ✓ Đăng ký tài khoản bằng: email, username, password, confirm password.
- ✓ Đăng nhập bằng: email và password.
- ✓ Đăng nhập thành công, chuyển sang trang room (lưu session).

❖ Trang room hiển thị:

- ✓ Ngày giờ thực tế.

- ✓ Độ ẩm, nhiệt độ (lấy thông tin từ HW).

➤ **Dùng dht11 và esp 8266**: để hiển thị nhiệt độ, độ ẩm ở trang room.

- ✓ Danh sách 3 phòng có các thiết bị thiết yếu trong gia đình (chuyển đổi hình ảnh qua firebase, ứng dụng)

❖ **Phòng khách: (4 thiết bị)**

- ✓ Camera, TiVi, quạt, đèn

❖ **Phòng ngủ: (3 thiết bị)**

- ✓ Điều hòa, đèn ngủ , TiVi

❖ **Phòng bếp: (4 thiết bị)**

- ✓ Tủ lạnh, đèn, quạt âm trần(hút khí thải), bếp

❖ **Thanh navigation bottom (3 button)**

- ✓ Nút Home để quay trở về room.
- ✓ Nút Person show thông tin 2 thành viên.
- ✓ Nút Log Out để xóa session quay trở về trang đăng nhập.

1.3. Giới hạn đề tài

Đề tài nhóm được thực hiện cơ bản đã hoàn thành giao diện giúp người dùng trải nghiệm vừa mang tính hiện đại vừa mang tính bắt mắt nhưng vẫn còn hạn chế về phần cứng và chưa được phổ biến rộng rãi nhiều người do cập nhật lên Google Play còn nhiều sự kiểm duyệt chặt chẽ

1.4. Phương pháp nghiên cứu

Các phương pháp được sử dụng để giải quyết đề tài: Phương pháp phân tích tổng kết kinh nghiệm, phương pháp thực nghiệm khoa học, phương pháp phân loại, phương pháp phân tích và tổng hợp lý thuyết.

1.5. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

❖ Phần mềm

- Thiết kế giao diện, thực hiện các chức năng :Figma , Android Studio Code (XML,Java)

❖ Phần cứng

- ESP8266 NodeMCU, Arduino IDE

❖ Database

- Firebase

Phạm vi nghiên cứu:

Xử lý các dữ liệu từ phần cứng gửi về Firebase và ngược lại, từ ứng dụng di động về Firebase và ngược lại cho phép tạo tài khoản mới dùng cho nhiều người trên cùng thiết bị di động

1.6. Bố cục quyển báo cáo

Quyển báo cáo gồm 5 chương:

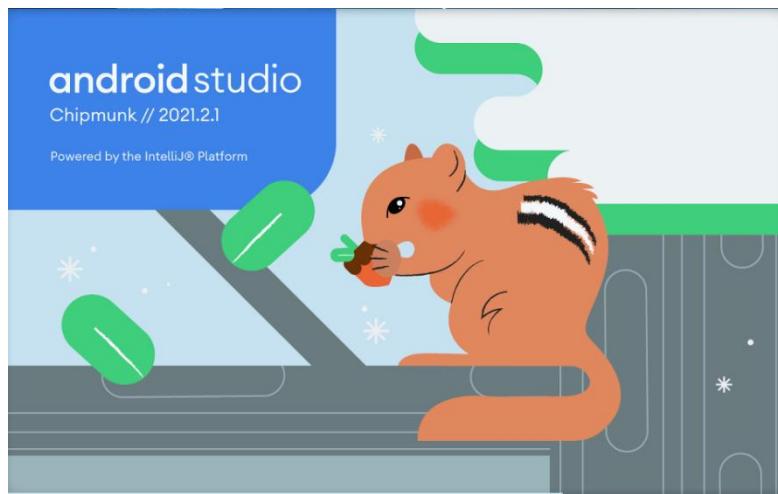
- ❖ **Chương 1: Giới thiệu về tài**
- ❖ **Chương 2: Cơ sở lý thuyết**
- ❖ **Chương 3: Thiết kế Figma, giao diện người dùng**
- ❖ **Chương 4: Chức năng của ứng dụng, đánh giá kết quả**
- ❖ **Chương 5: Kết luận và hướng phát triển**

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Phần Mềm Sử Dụng

2.1.1. *Android Studio*

Android Studio là IDE chính thức được sử dụng trong phát triển ứng dụng Android dựa trên IntelliJ IDEA. Chức năng chính của Android Studio là cung cấp các giao diện người dùng có thể tạo các ứng dụng và xử lý các công cụ file phức tạp phía sau. Ngôn ngữ lập trình được sử dụng trong Android Studio là Java hoặc Kotlin.



Hình 2.1 Giao diện phần mềm Android studio

2.1.2. Figma

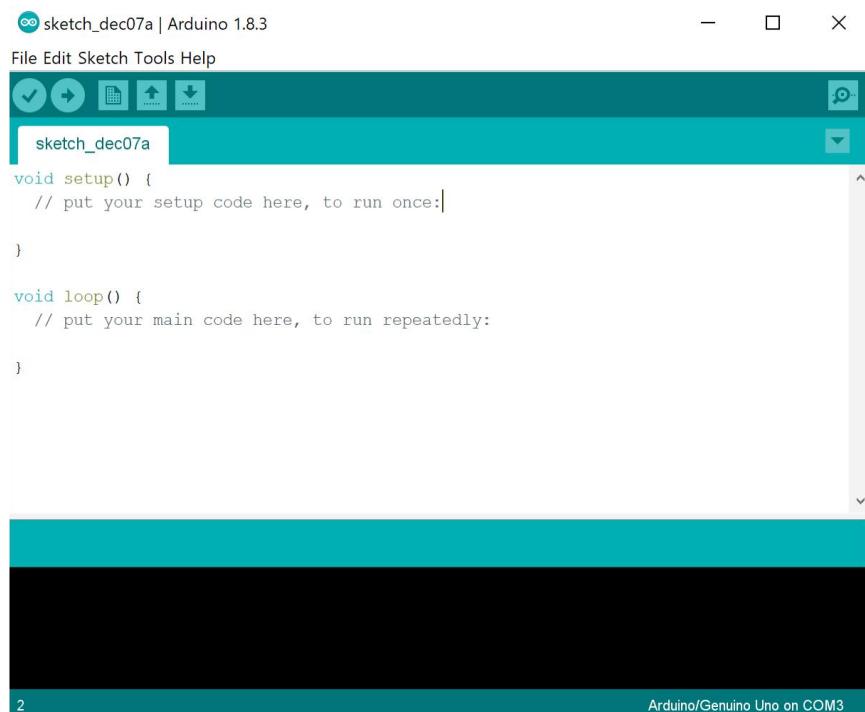
Figma là một công cụ thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX) dựa trên web Figma cho phép người dùng tạo ra các bản vẽ và prototype cho ứng dụng di động, trang web, hoặc các sản phẩm kỹ thuật số khác. Nó cung cấp một giao diện người dùng trực quan và dễ sử dụng, cho phép các nhà thiết kế tạo ra các thành phần, trang, và luồng làm việc bằng cách kéo và thả, vẽ và chỉnh sửa các yếu tố giao diện, và tạo ra các animation và hiệu ứng chuyển động. Một trong những đặc điểm nổi bật của Figma là tính năng hợp tác trực tiếp. Nó cho phép nhiều người dùng làm việc cùng một lúc trên cùng một thiết kế, xem các thay đổi ngay lập tức và trò chuyện trong thời gian thực.



Hình 2.2 Giao diện phần mềm Figma

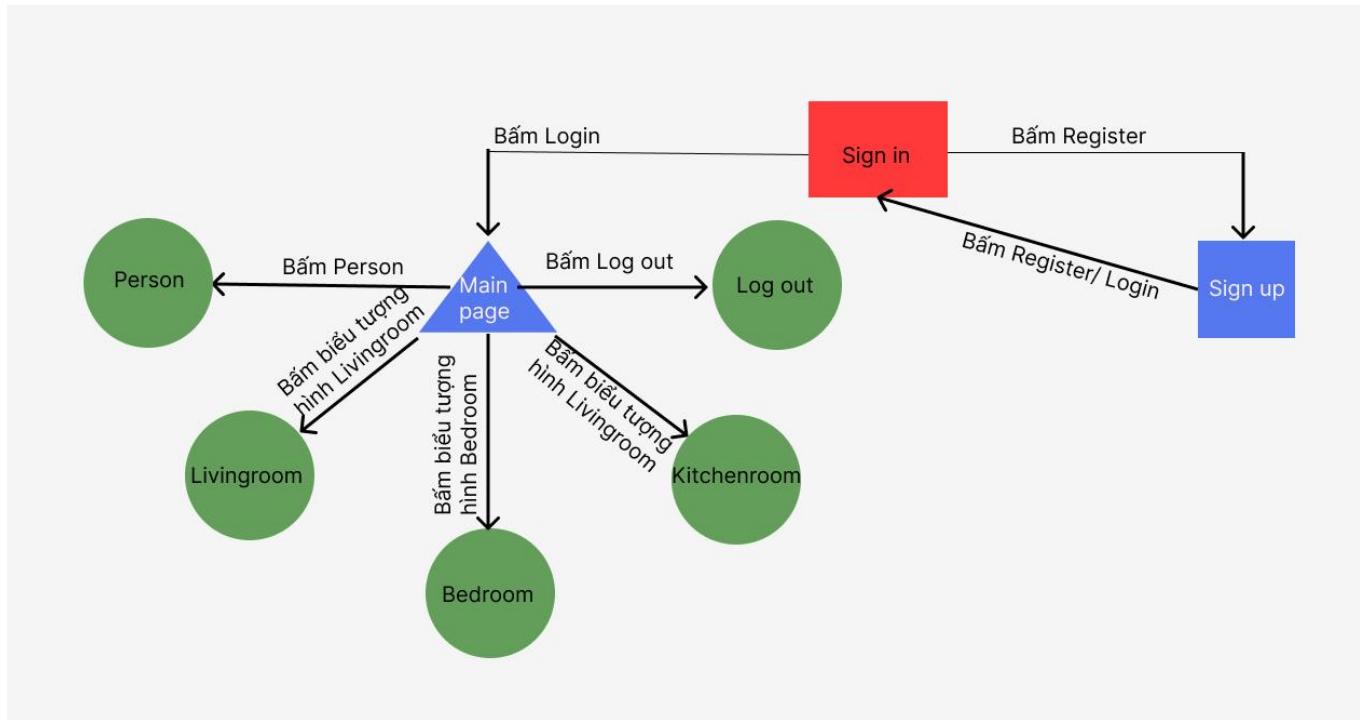
2.1.3. Arduino IDE

Arduino IDE (Integrated Development Environment) là một phần mềm được sử dụng để lập trình các board Arduino. Nó cung cấp một giao diện đồ họa và các công cụ hỗ trợ để viết, biên dịch và nạp chương trình cho các board Arduino. Arduino IDE giúp viết mã nguồn Arduino bằng ngôn ngữ lập trình dựa trên C/C++. Nó cung cấp các chức năng như kiểm tra cú pháp, tạo ra các chương trình, và nạp chương trình lên board Arduino.



Hình 2.3 Giao diện Arduino IDE

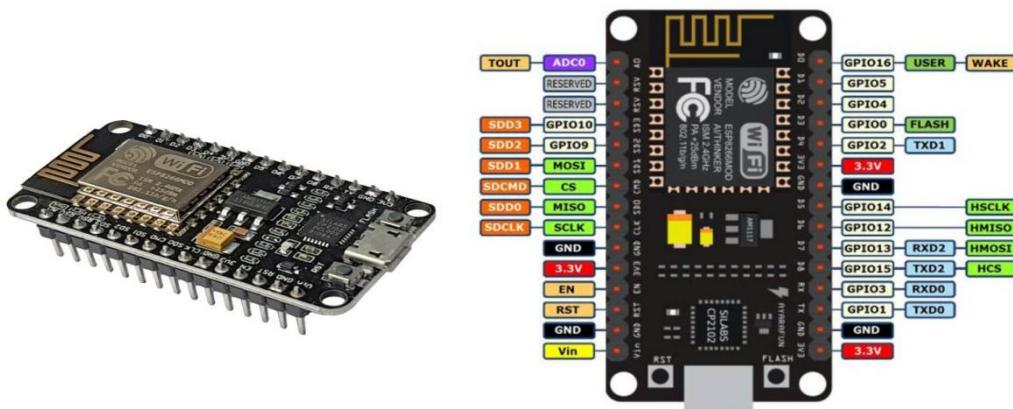
2.1.4. Sơ đồ giải thuật



2.2. Phần Cứng Hiện Thị

2.2.1 NODEMCU ESP8266

Module ESP266 là module wifi được đánh giá rất cao cho các ứng dụng liên quan đến Internet và Wifi cũng như các ứng dụng truyền nhận sử dụng thay thế cho các module RF khác với khoảng cách truyền lên tới 100 mét (với môi trường không có vật cản). Trên 400m với anten và router thích hợp.



Hình 2.4 Esp8266

Module nodeMCU ESP8266 có các chân dùng để cấp nguồn và thực hiện kết nối. Chức năng

của từng chân như sau:

- Tx : chân Tx của giao thức UART, kết nối với chân Rx của vi điều khiển.
- Rx : chân Rx của giao thức UART, kết nối với chân Tx của vi điều khiển.
- RST : chân reset
- 9 chân GPIO từ D0-D8, có chức năng PWM, IIC, giao tiếp SPI, I2C.

Các tính năng của nodeMCU ESP8266 là:

- Hỗ trợ chuẩn 802.11 b/g/n.
- Wifi 2.4 GHz, hỗ trợ WPA/WPA2.
- Chuẩn điện áp hoạt động : 3.3V
- Chuẩn giao tiếp nối tiếp UART với tốc độ Baud lên đến 115200.
- Tích hợp ngăn xếp giao thức TCP/IP
- Tích hợp chuyển đổi TR, LNA, bộ khêch đại công suất và phù hợp với mạng.
- Tích hợp PLL, bộ quản lí và các đơn vị quản lí điện năng.
- Công suất đầu ra +19.5dBm trong chế độ 802.11b/g/n.
- Tích hợp cảm biến nhiệt độ.
- Hỗ trợ nhiều loại anten.
- Chế độ chờ tiêu thụ điện năng <1.0mW (DTIM3).
- Hỗ trợ cả hai giao tiếp TCP và UDP.
- Làm việc như các máy chủ có thể kết nối với các máy con.

2.2.2 DHT11

DHT11 là một cảm biến kỹ thuật số giá rẻ để cảm nhận nhiệt độ và độ ẩm. Cảm biến này có thể dễ dàng giao tiếp với bất kì vi điều khiển nào như Arduino, Raspberry Pi,...để đo độ ẩm và nhiệt độ ngay lập tức.

Cảm biến DHT11 bao gồm một phần tử cảm biến độ ẩm điện dung và một điện trở nhiệt

để cảm nhận nhiệt độ. Tự điện cảm biến nhiệt độ có hai cực với chất nền giữ ẩm làm chất điện môi giữa chúng. Thay đổi giá trị điện dung xảy ra với sự thay đổi của các mức độ ẩm. IC đo, xử lý các giá trị điện trở đã thay đổi này và chuyển chúng thành dạng kỹ thuật số. Để đo nhiệt độ, cảm biến này sử dụng một nhiệt điện trở có hệ số nhiệt độ âm, làm giảm giá trị điện trở của nó khi nhiệt độ tăng. Để có được giá trị điện trở lớn hơn ngay cả đối với sự thay đổi nhỏ nhất của nhiệt độ, cảm biến này thường làm bằng gốm bán dẫn hoặc polymer.

Cảm biến DHT11 cấu tạo gồm ba chân:

- VCC : nối với nguồn
- Data : Đầu ra cả nhiệt độ và độ ẩm thông qua dữ liệu nối tiếp
- Ground : nối đất

Thông số kỹ thuật DHT11:

- Điện áp hoạt động: 3V -5V DC
- Dòng điện tiêu thụ: 2.5mA
- Phạm vi cảm biến độ ẩm: 20% -90% RH, sai số $\pm 5\%$ RH
- Phạm vi cảm biến nhiệt độ: 0°C ~50°C, sai số $\pm 2\%$ RH

2.2.3 LED

LED là ký hiệu viết tắt của cụm từ “Light Emitting Diode”, mang nghĩa là diot phát quang. Hay được hiểu theo nghĩa khác là các diot có khả năng phát sáng hoặc tia tử ngoại, tia hồng ngoại,..

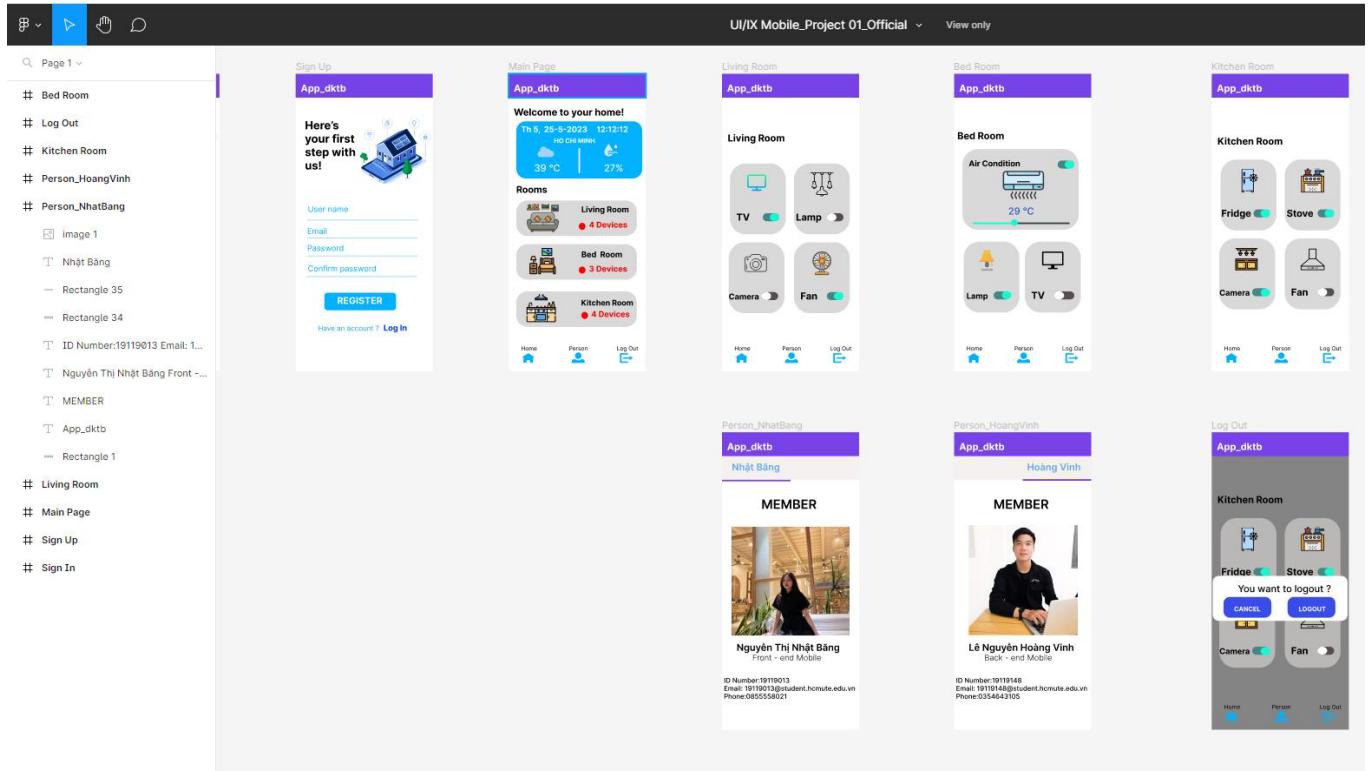


Hình 2.5 Led

Cũng giống như diode, LED được cấu tạo từ một khối bán dẫn loại p ghép với khối bán dẫn loại n. Dây dài là cực dương (Anode) và dây ngắn là cực âm (Cathode).

CHƯƠNG 3: THIẾT KẾ FIGMA, GIAO DIỆN NGƯỜI DÙNG VỚI ANDROID STUDIO

3.1. Thiết kế Figma

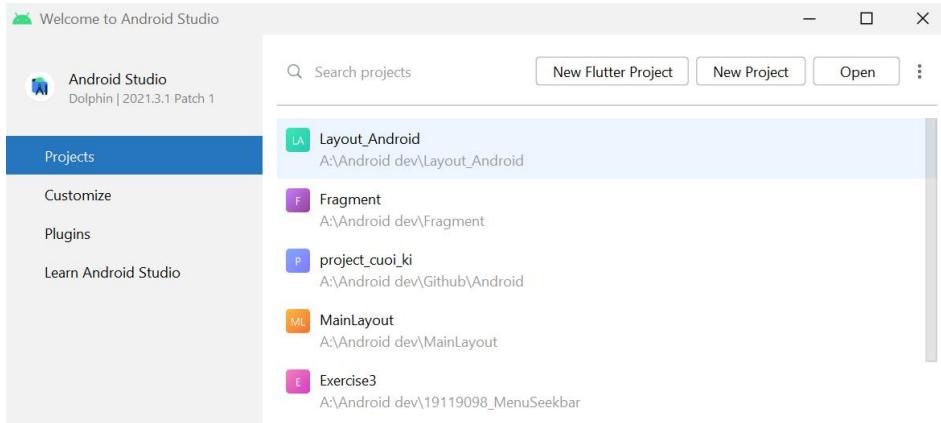


Hình 3.1 Thiết kế Figma

3.2. Giao diện người dùng với Android Studio

3.2.1 Giao diện đăng ký, đăng nhập

Bước 1: mở ứng dụng Android Studio, chọn “New Project” để tạo project.



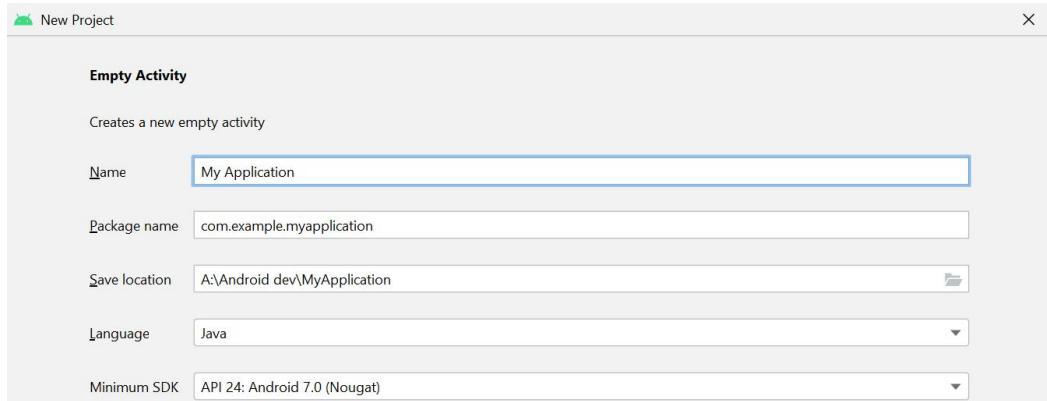
Hình 3.2. Tạo mới project

Bước 2: chọn “Empty Activity” rồi nhấn Next



Hình 3.3 Chọn Empty Activity

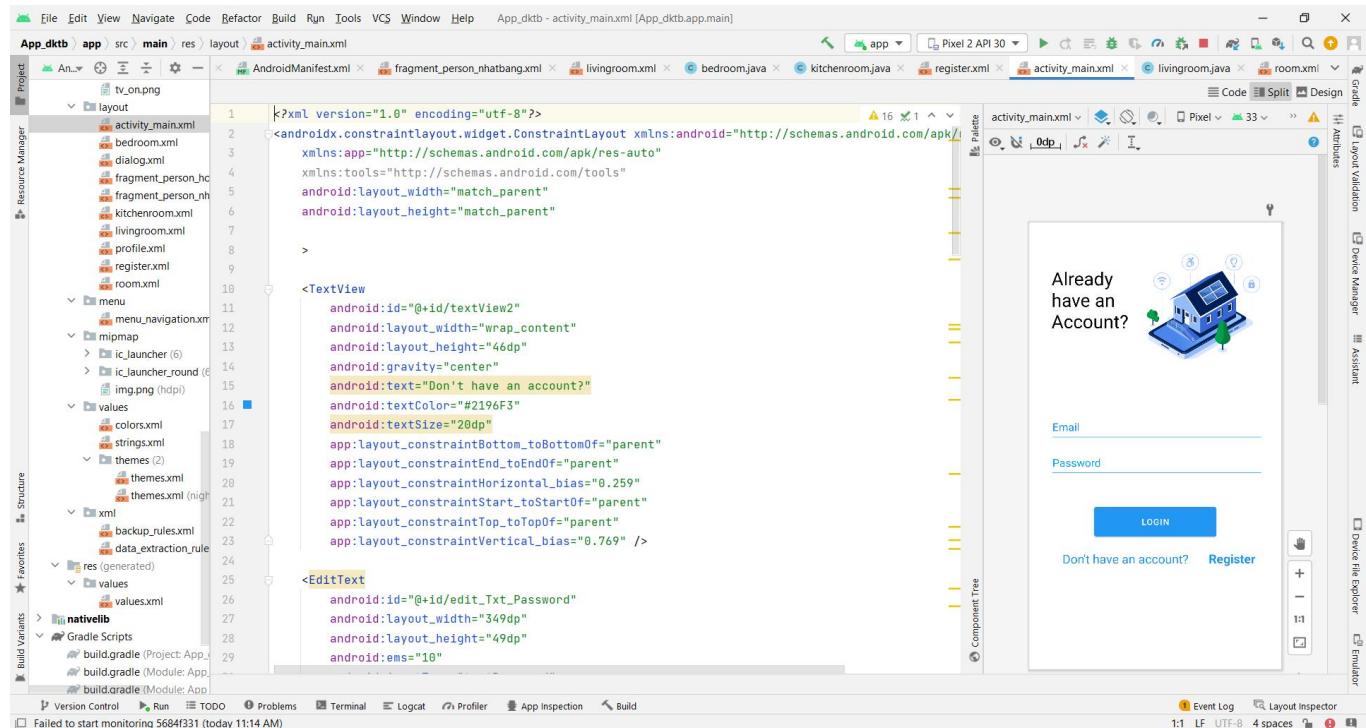
Bước 3: đặt tên cho Project, điền các lựa chọn và thế là bạn đã có 1 project android studio



Hình 3.4 Đường dẫn project

- Name: đặt tên cho ứng dụng.
- Pakage name: tên package để chứa code.
- Save location: nơi lưu trữ Project.
- Language: ngôn ngữ sử dụng – Java.
- Minium SDK: phiên bản nhỏ nhất mà ứng dụng chạy được.

Layout đăng nhập sẽ nằm ở file activity_main.xml



Hình 3.5 Layout login

Layout đăng ký sử dụng ConstraintLayout với width và height là “match_parent” (theo phần tử cha).

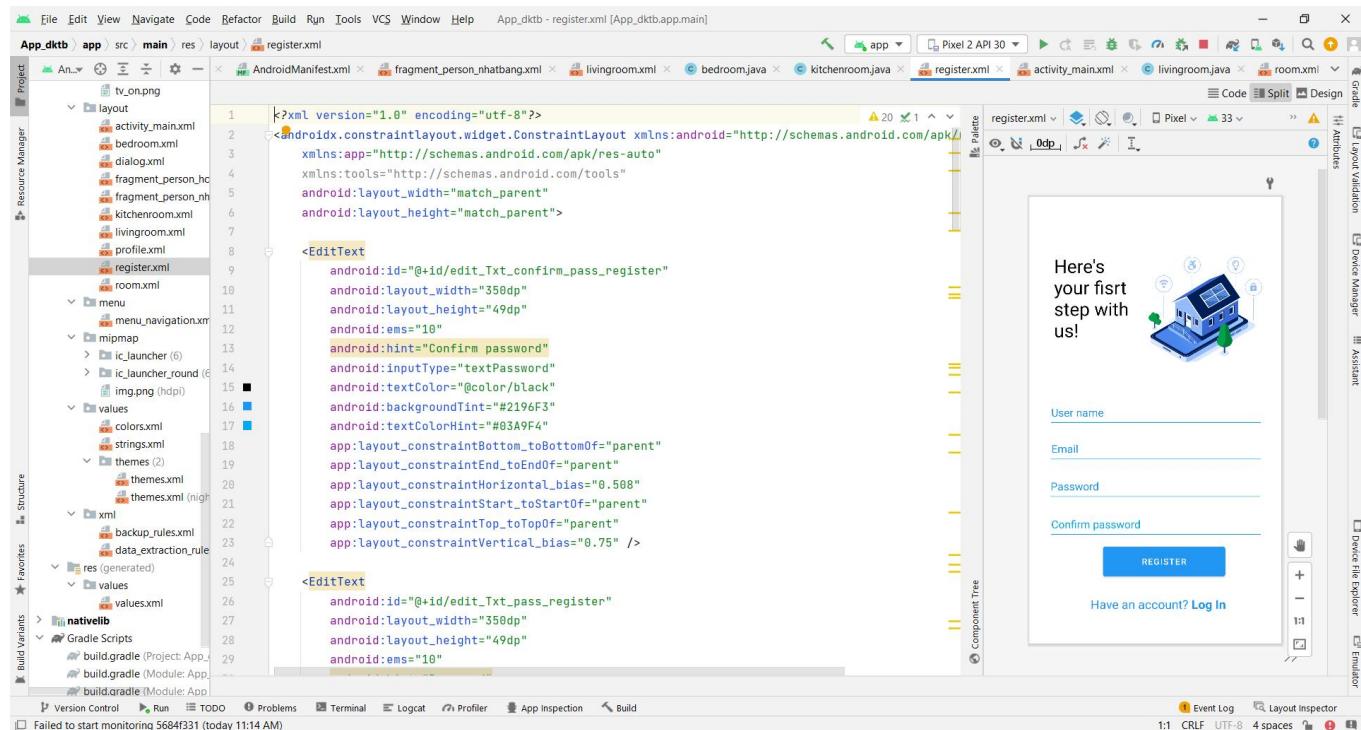
Logo của ứng dụng ta sử dụng thẻ ImageView. Email, Password ta sử dụng thẻ EditText, LOGIN sử dụng thẻ Button và còn lại là các thẻ TextView

Các thuộc tính quan trọng cần chú ý ở đây là:

- gravity= “center”: thuộc tính căn giữa màn hình của các phần tử
- fontFamily: thay đổi font chữ của ứng dụng
- gravity= “center”: thuộc tính căn giữa màn hình của các phần tử
- margin: căn chỉnh các phần tử qua trái, phải, trên dưới.
- id: đây là thuộc tính rất quan trọng, ứng dụng sẽ thông qua id để giao tiếp, tương tác với nhau.

Layout đăng ký sẽ nằm ở file register.xml

Tương tự như layout đăng nhập, ở layout đăng ký cũng sử dụng các thẻ TextView, EditText, ImageView và Button.

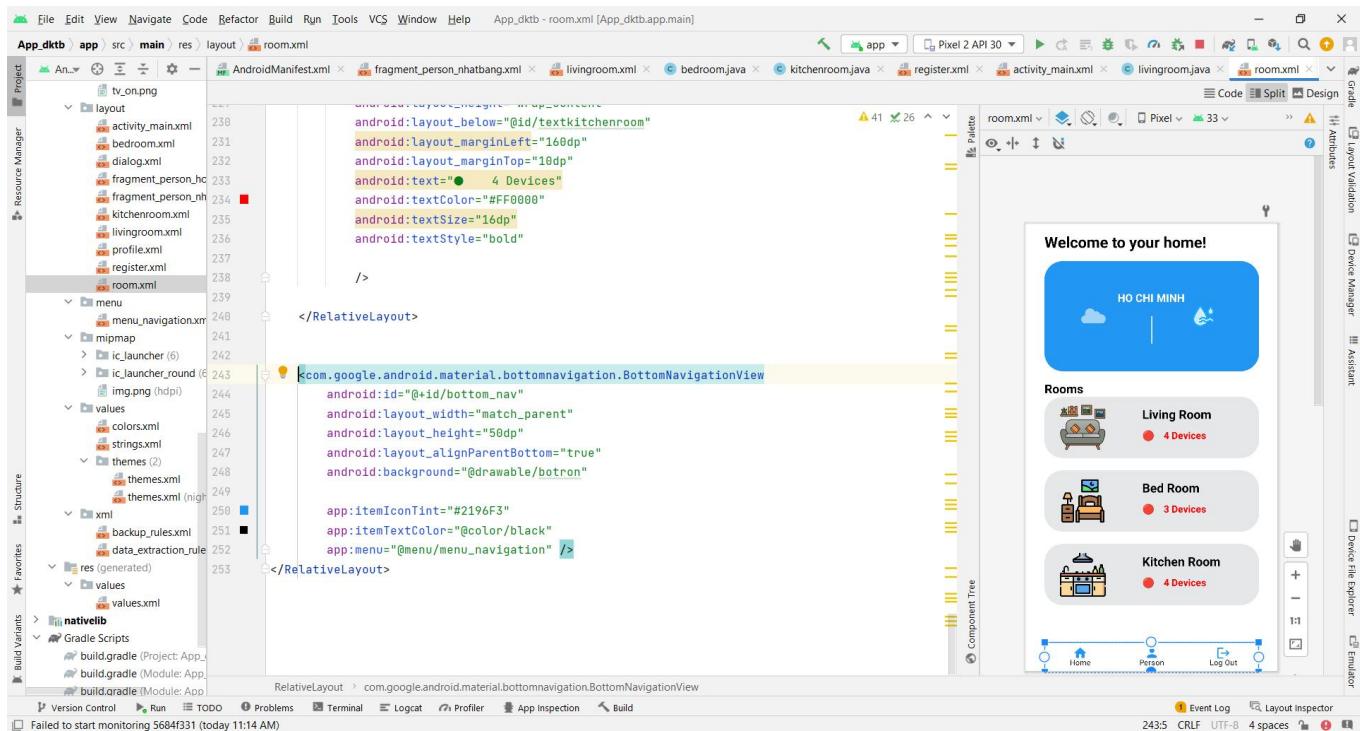


Hình 3.6 Layout register

3.2.2 Giao diện room

Layout room sẽ nằm ở file room.xml

Layout room sử dụng Relativelayout với width và height là “match_parent” (theo phần tử cha) và sử dụng paddingLeft và paddingRight để chống tràn viền.



Hình 3.7 Layout room

Trong Layout room sử dụng 4 RelativeLayout lân lượt là thông tin nhiệt độ, độ ẩm, realtime về căn nhà, Living Room, Bed Room và KitchenRoom. Và cuối cùng là bottom navigation hiển thị home, person và logout.

Các thuộc tính quan trọng cần chú ý ở đây là:

- layout_below: phần tử sử dụng thuộc tính này sẽ nằm dưới id trong layout_below
- backgroundTint: thay đổi background của RelativeLayout

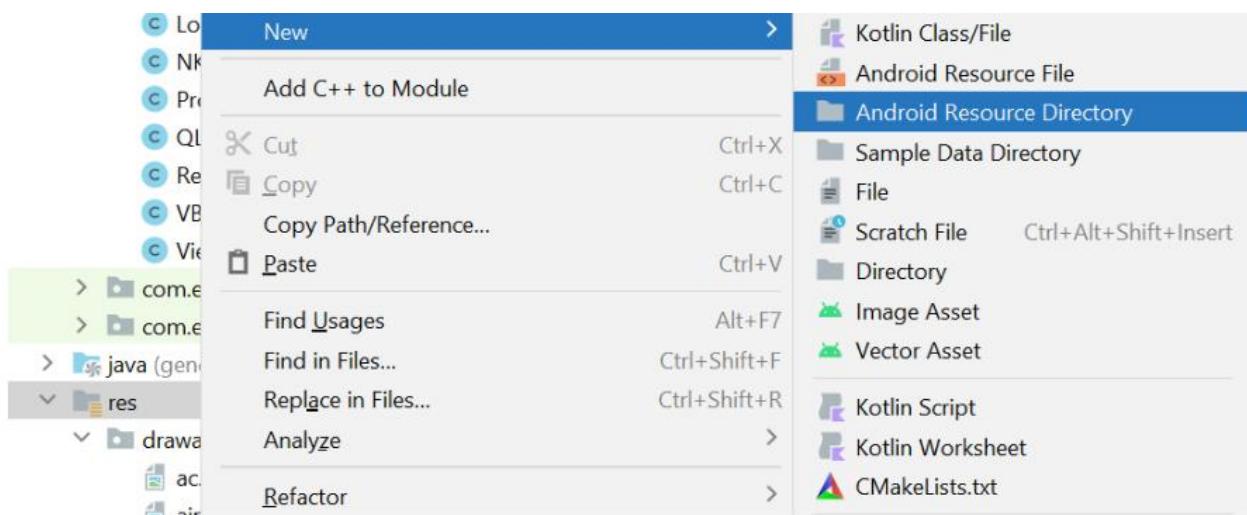
- radius: bo tròn các RelativeLayout

- src: thêm hình theo đường dẫn nằm vào layout

- id: đây là thuộc tính rất quan trọng, ứng dụng sẽ thông qua id để giao tiếp, tương tác với nhau.

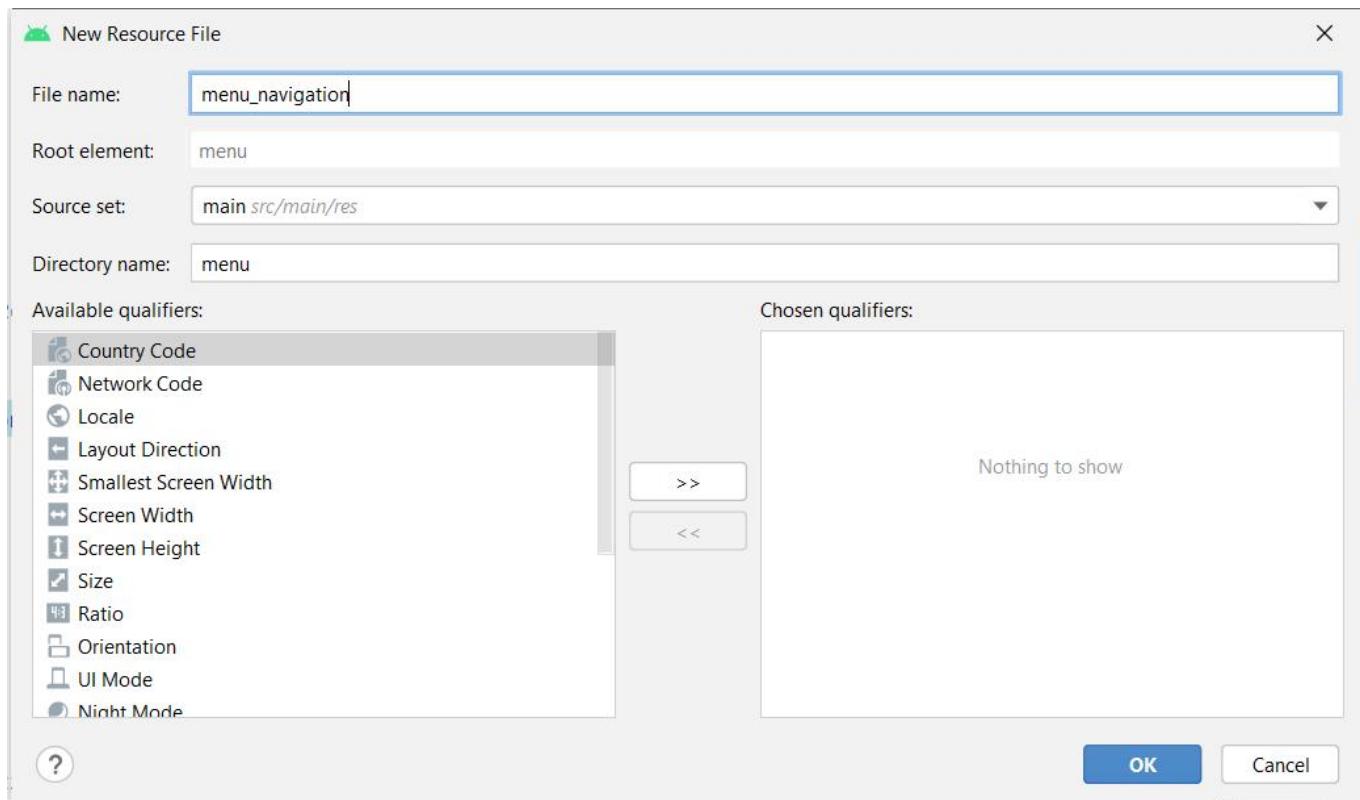
Ở bottom navigation ta phải tạo một menu_bottom_nav.xml nằm trong folder menu (đây là folder tự tạo).

Click chuột phải vào folder res => new => Android Resource Directory



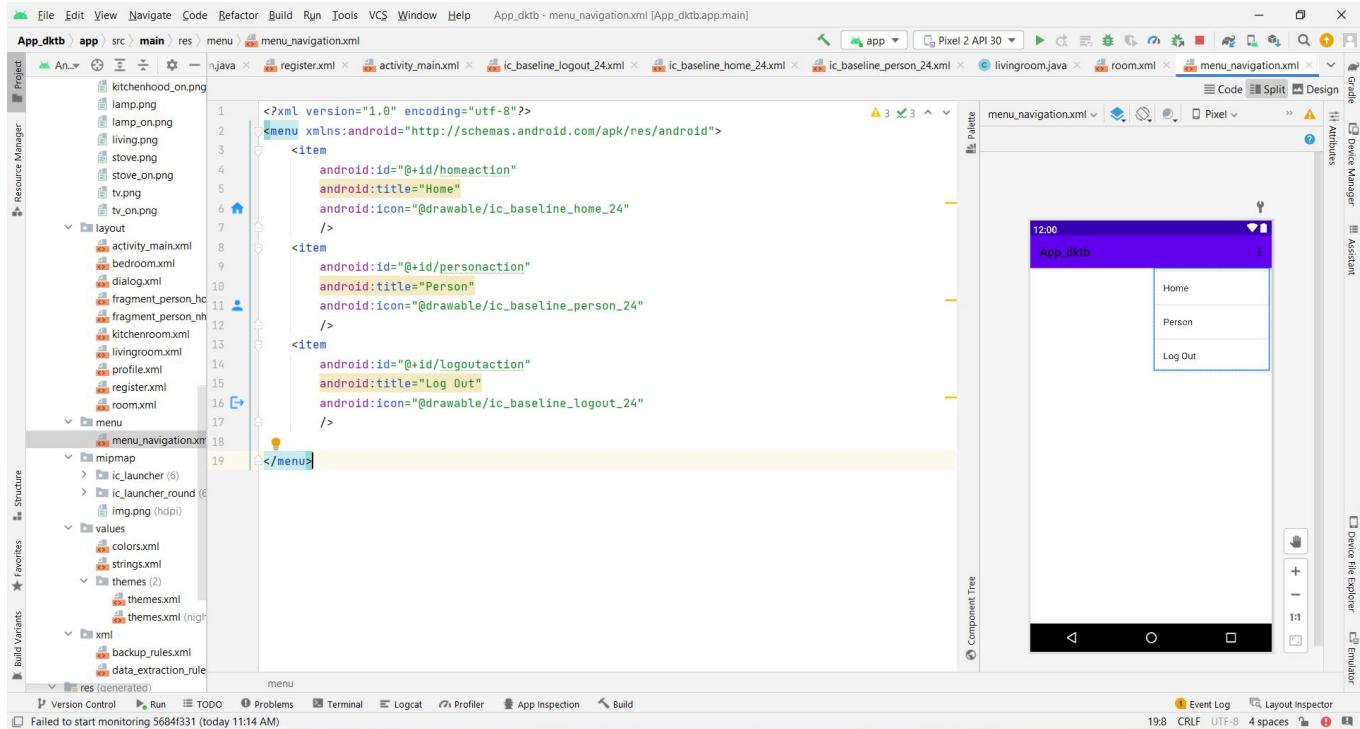
Hình 3.8 Tạo menu

Đặt tên cho file là menu_navigation và chọn Resource type là menu sau đó nhấn ok là xong.



Hình 3.9 Đặt tên menu

Vào file menu_bottom_nav.xml để tạo layout.



Hình 3.10 Layout menu

Các thẻ quan trọng cần chú ý ở đây là:

- Thẻ menu: chứa 3 item của bottom navigation.
- Thẻ item: gồm các thuộc tính về id (để thao tác), title (đặt tựa đề cho item) và icon (icon hiển thị của item được lấy từ icon có sẵn trong thư viện của ứng dụng).

Cuối cùng thêm vào file dashboard.xml để sử dụng bottom navigation.



Hình 3.11 Giao diện Layout menu

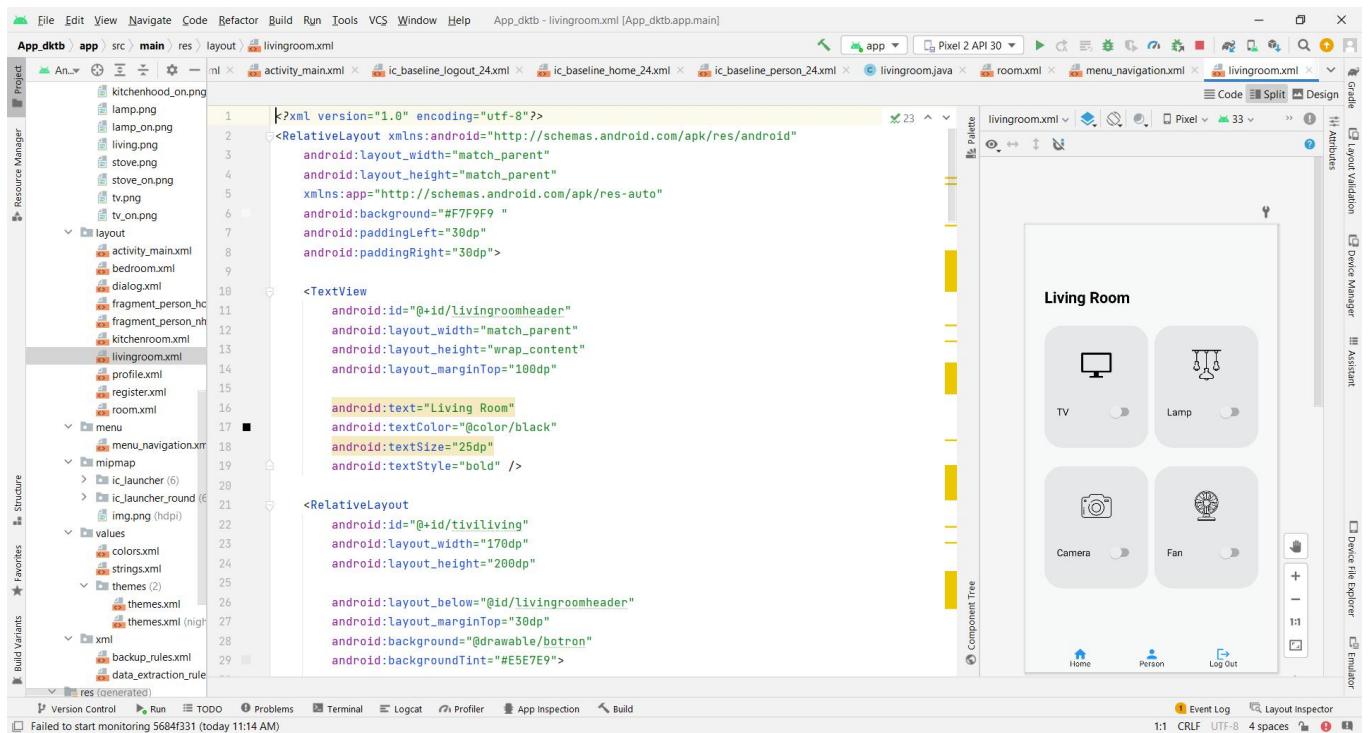
Các thuộc tính quan trọng cần chú ý ở đây là:

- menu: thêm file menu_bottom_nav vừa tạo trước đó vào.
- layout_alignParentBottom= “true”: đặt bottom navigation vào dưới cùng của layout cha.
- itemIconTint: đổi màu cho icon.
- itemTextColor: đổi màu text của icon.

3.2.3 Giao diện LivingRoom, BedRoom và Kitchen Room

Layout LivingRoom sẽ nằm ở file livingroom.xml

Layout LivingRoom sử dụng Relativelayout với width và height là “match_parent” (theo phần tử cha) và sử dụng paddingLeft và paddingRight để chống tràn viền.



Hình 3.12 Layout livingroom

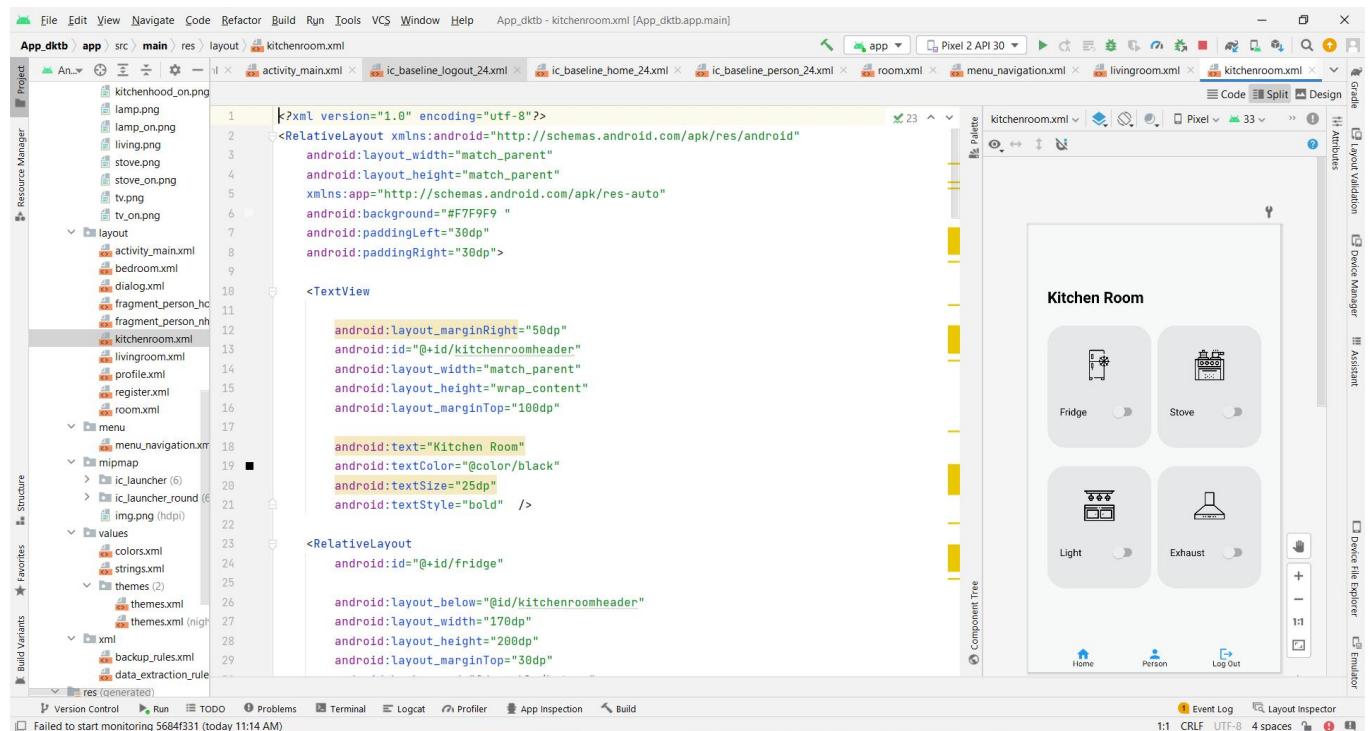
Trong Layout LivingRoom sử dụng 4 RelativeLayout để hiển thị 4 thiết bị bao gồm TV, Lamp, Camera và Fan

Thẻ quan trọng cần chú ý ở đây là:

- Thẻ Switch: tạo nút chuyển đổi on off cho các thiết bị.

Layout KitchenRoom sẽ nằm ở file kitchenroom.xml

Tương tự Layout LivingRoom, layout KitchenRoom sử dụng Relativelayout với width và height là “match_ parent” (theo phần tử cha) và sử dụng paddingLeft và paddingRight để chống tràn viền.

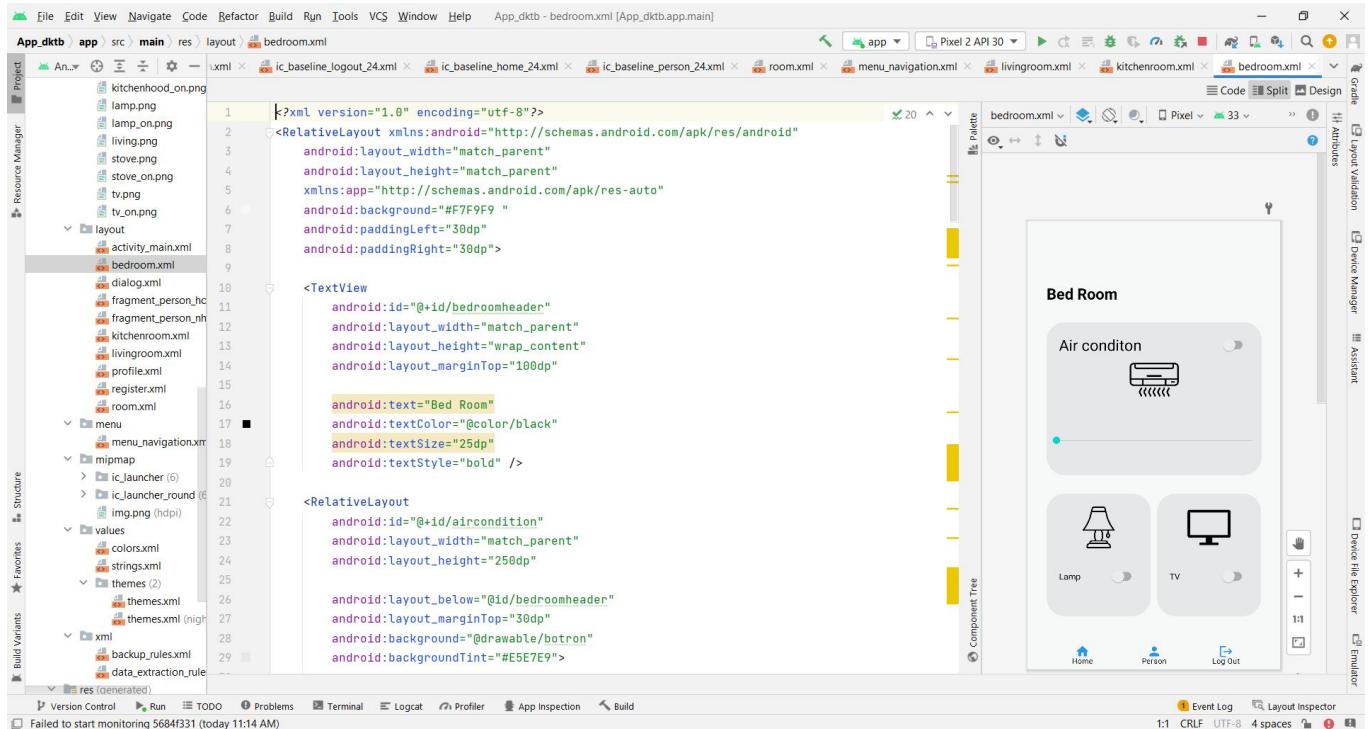


Hình 3.13 Layout kitchenroom

Trong Layout BedRoom sử dụng 4 RelativeLayout để hiển thị 4 thiết bị bao gồm Fridge, Stove, Light, Exhaust.

Layout BedRoom sẽ nằm ở file bedroom.xml

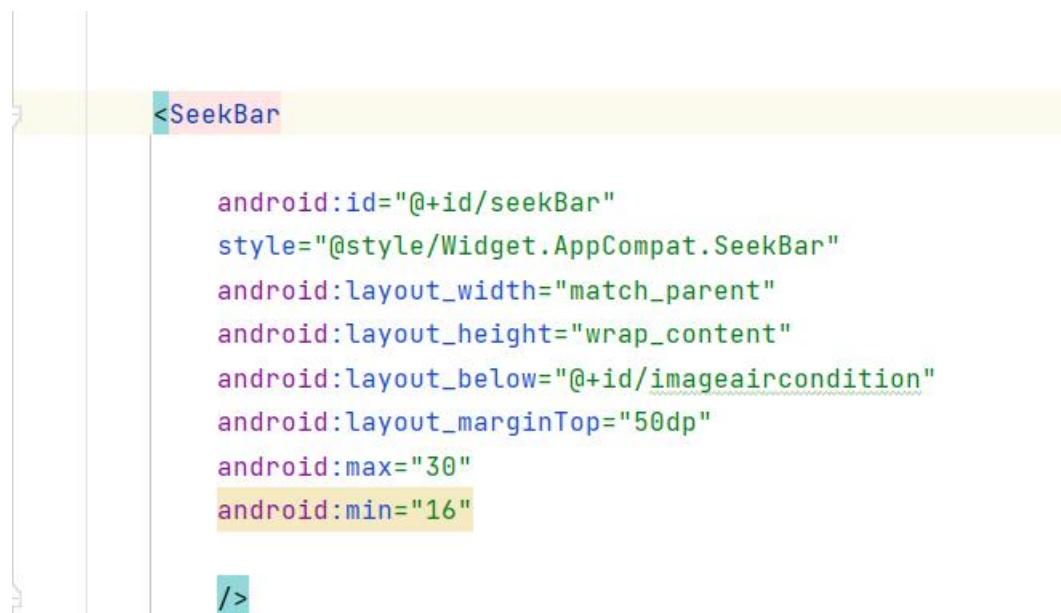
Layout BedRoom sử dụng Relativelayout với width và height là “match_ parent” (theo phần tử cha) và sử dụng paddingLeft và paddingRight để chống tràn viền.



Hình 3.14. Layout bedroom

Trong Layout BedRoom sử dụng 3 Relativelayout để hiển thị 3 thiết bị gồm Air condition, Lamp và TV.

Thiết bị Lamp và TV tương tự như ở 2 phòng khác chỉ sử dụng switch. Nhưng ở thiết bị Air condition có sử dụng thêm SeekBar để điều chỉnh được nhiệt độ.



Hình 3.15 Layout thanh seekbar

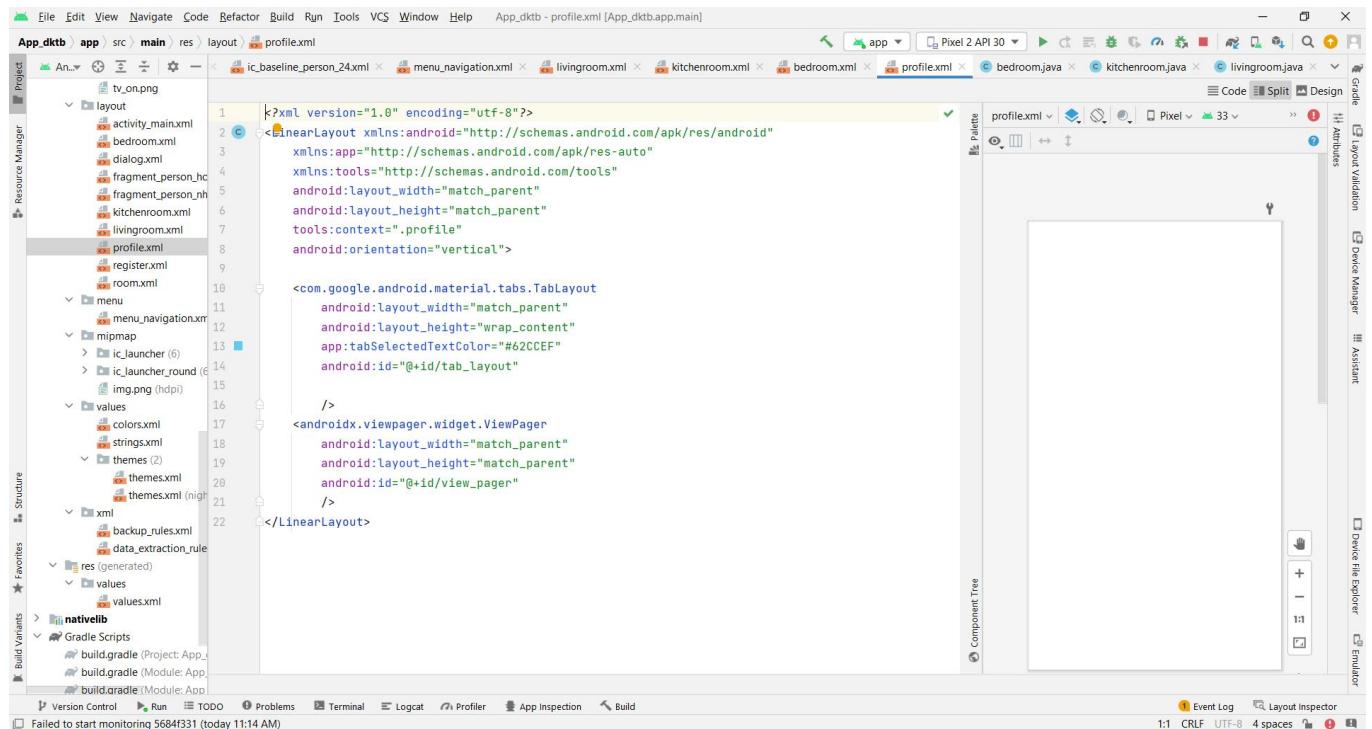
Thuộc tính cần chú ý ở Seekbar:

- min và max: điều chỉnh giá trị min, max của Seekbar.
- style: sử dụng style có sẵn của thư viện ứng dụng.

3.2.4 Giao diện Profile

Layout Profile sẽ nằm ở file profile.xml và 3 file fragment để hiển thị về thông tin của 3 thành viên thực hiện

Profile sử dụng LinearLayout với width và height là “match_parent”.



Hình 3.1 Layout profile

Layout Profile với 2 thẻ chính là TabLayout và ViewPager

- Thẻ TabLayout hiển thị thanh công cụ để án chuyên đổi qua lại giữa các fragment.
- Thẻ ViewPager hiển thị thông tin của các fragment.

CHƯƠNG 4: CHỨC NĂNG CỦA ỦNG DỤNG

4.1. Đăng kí, đăng nhập qua firebase

4.1.1 Kết nối firebase với Android Studio

Đầu tiên, kích hoạt chức năng authentication của firebase, chọn phương thức đăng kí, đăng nhập bằng email.



Hình 4.1: Chức năng authentication với phương thức đăng kí, đăng nhập của firebase

Kết nối firebase với Android Studio. Mở phần mềm Android Studio, thanh công cụ chọn Tools sau đó chọn Firebase

The screenshot shows the Firebase documentation website. At the top, there are tabs for 'Assistant' and 'Firebase', with 'Firebase' being the active tab. Below the tabs is the Firebase logo and the word 'Firebase'. A main heading states: 'Firebase gives you the tools and infrastructure from Google to help you develop, grow and earn money from your app. Learn more'. There are three main sections: 'Analytics', 'Authentication', and 'Realtime Database'. The 'Analytics' section has a brief description and a link to 'More info'. The 'Authentication' section is expanded, showing several sub-links under the heading 'Sign in and manage users with ease using popular login providers like Google, Facebook, and others. You can even use a custom authentication system. More info'. These sub-links include: 'Authenticate using Google', 'Authenticate using Google [Java]', 'Authenticate using Facebook Login', 'Authenticate using Facebook Login [Java]', 'Authenticate using a custom authentication system', and 'Authenticate using a custom authentication system [Java]'. A red arrow points to the 'Authenticate using Google' link. The 'Realtime Database' section has a brief description.

Assistant Firebase

Firebase

Firebase gives you the tools and infrastructure from Google to help you develop, grow and earn money from your app. Learn more

▶ **Analytics**

Measure user activity and engagement with free, easy, and unlimited analytics. [More info](#)

▶ **Authentication**

Sign in and manage users with ease using popular login providers like Google, Facebook, and others. You can even use a custom authentication system. [More info](#)

▶ **Realtime Database**

Store and sync data with this cloud-hosted NoSQL database. Data is synced across all clients in realtime and

Hình 4.2 Cách thức để chọn phương thức firebase

Sau đó chọn Authenticate using Google.

The screenshot shows the Firebase console's Authentication screen. At the top, there are tabs for Assistant and Firebase, with Firebase selected. Below the tabs, it says "← Firebase > Authentication". The main title is "Authenticate using Google". A sub-instruction says "You can let your users authenticate with Firebase using their Google Account". There is a "Launch in browser" button. Below this, a numbered list starts with "1 Connect your app to Firebase", which has a "Connect to Firebase" button. A green checkmark indicates "1 of 2 modules completed". The second step, "2 Add the Firebase Authentication SDK to your app", has a corresponding button. A note at the bottom says "NOTE: After adding the SDK, here are some other helpful configurations".

Hình 4.3 Cách thức để chọn phương thức firebase

Bấm kích hoạt từng bước (từ bước 1 tới bước 3) là hoàn thành các bước thực hiện kết nối firebase với android studio.

4.1.2 Chức năng đăng kí qua firebase

Đầu tiên, ánh xạ các biến với layout và khai báo các biến cần sử dụng



```
1 package com.example.app_dktb;
2
3 import ...
26
27 public class register extends AppCompatActivity {
28     //Trang đăng kí
29     EditText edit_Txt_Username_register, edit_Txt_email_register, edit_Txt_pass_register, edit_Txt_confirm_pass_register;
30     Button btn_register;
31     TextView btn_login_toggle;
32     private FirebaseAuth mAuth;
33
34     @Override
35     protected void onCreate(Bundle savedInstanceState) {
36         super.onCreate(savedInstanceState);
37         setContentView(R.layout.register);
38         mAuth = FirebaseAuth.getInstance();
39         //Trang đăng kí
40         edit_Txt_Username_register=findViewById(R.id.edit_Txt_Username_register);
41         edit_Txt_email_register=findViewById(R.id.edit_Txt_email_register);
42         edit_Txt_pass_register=findViewById(R.id.edit_Txt_pass_register);
43         edit_Txt_confirm_pass_register=findViewById(R.id.edit_Txt_confirm_pass_register);
44         btn_register=findViewById(R.id.btn_register);
45         btn_login_toggle=findViewById(R.id.btn_login_toggle);
46     }
}
```

Hình 4.4: Ánh xạ, khai báo các biến sử dụng ở file register.java

Từ dòng 40 tới 45: ánh xạ các biến sẽ sử dụng với file xml (file layout).

Dòng 32: khai báo phương thức Authencation của firebase.

Dòng 38: nhận và gửi thông tin từ firebase.



```
47
48
49
50     btn_register.setOnClickListener(new View.OnClickListener() {
51         @Override
52         public void onClick(View view) { register(); }
53     );
54
55     btn_login_toggle.setOnClickListener(new View.OnClickListener() {
56         @Override
57         public void onClick(View v) { login_toggle(); }
58     );
59
60
61 }
```

Hình 4.5 Bắt sự kiện cho 2 nút nhấn

Từ dòng 48 tới 53: bắt sự kiện cho nút nhấn đăng kí sử dụng hàm register.

Từ dòng 56 tới 60: bắt sự kiện cho nút nhấn chuyển từ trang đăng kí sang trang đăng nhập sử dụng hàm login_toggle.

Phương thức intent cho phép chuyển đổi từ trang đăng kí sang trang đăng nhập (nhưng các bạn phải khai báo trong file AndroidManifest.xml).

```

<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/img"
    android:label="App_dktb"
    android:roundIcon="@mipmap/img"
    android:supportsRtl="true"
    android:theme="@style/Theme.App_dktb"
    tools:targetApi="31">
    <activity
        android:name=".login"
        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=".register"/>
    <activity android:name=".room"/>
    <activity android:name=".livingroom"/>
    <activity android:name=".bedroom"/>
    <activity android:name=".kitchenroom"/>
    <activity android:name=".profile"/>
</application>

```

Hình 4.6 Khai báo các file java để chuyển trang

Với trang đầu tiên là trang đăng nhập nhờ xét dữ kiện từ dòng 16 tới dòng 24.

```

62     private void register(){
63         String email, password, confirm_password, username;
64         email = edit_Txt_email_register.getText().toString();
65         password = edit_Txt_pass_register.getText().toString();
66
67         username = edit_Txt_Username_register.getText().toString();
68         confirm_password = edit_Txt_confirm_pass_register.getText().toString();
69
70         if(TextUtils.isEmpty(username)){
71             Toast.makeText( context: this, text: "Vui lòng nhập tên người dùng", Toast.LENGTH_SHORT).show();
72             return;
73         }
74
75         if(TextUtils.isEmpty(email)){
76             Toast.makeText( context: this, text: "Vui lòng nhập email", Toast.LENGTH_SHORT).show();
77             return;
78         }
79
80         if(TextUtils.isEmpty(password)){
81             Toast.makeText( context: this, text: "Vui lòng nhập mật khẩu", Toast.LENGTH_SHORT).show();
82             return;
83         }else{
84             if (password.length() < 6) {
85                 Toast.makeText( context: this, text: "Vui lòng nhập 6 ký tự cho mật khẩu", Toast.LENGTH_SHORT).show();
86                 return;
87             }
88         }
89         if(TextUtils.isEmpty(confirm_password)){
90             Toast.makeText( context: this, text: "Vui lòng nhập mật khẩu xác nhận", Toast.LENGTH_SHORT).show();
91             return;
92         }

```

Hình 4.7 Xét điều kiện cho các ô

Từ dòng 67 tới 70: xét điều kiện cho ô nhập user name (nếu ô user name trống hiển thị thông báo).

Từ dòng 71 tới 77 và 84 tới 86: tương tự với user name xét điều kiện cho ô email, confirm password và password .

Từ dòng 79 tới 82: xét thêm điều kiện bắt người dùng nhập đủ 6 kí tự.

```

93     if(password.equals(confirm_password)){
94         Task<AuthResult> authResultTask = mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
95             @Override
96             public void onComplete(@NonNull Task<AuthResult> task) {
97                 if (task.isSuccessful()) {
98                     Toast.makeText(getApplicationContext(), text: "Tạo tài khoản thành công", Toast.LENGTH_SHORT).show();
99                     login_toggle();
100                } else {
101                    Toast.makeText(getApplicationContext(), text: "Tạo tài khoản thất bại", Toast.LENGTH_SHORT).show();
102                }
103            });
104        });
105    }else{
106        Toast.makeText(context: this, text: "Mật khẩu bạn nhập không trùng khớp", Toast.LENGTH_SHORT).show();
107        return;
108    }
109 }
110
111 private void login_toggle(){
112     Intent intent=new Intent( packageContext: register.this,login.class);
113     startActivity(intent);
114 }
115
116
117
118 }
119

```

Hình 4.8 Xét điều kiện ô password bằng ô confirm password

Dòng 89: cú pháp thực hiện tạo tài khoản lên firebase.

Dòng 92 tới 94: đăng ký thành công thì hiển thị thông báo và chuyển về trang đăng nhập

Dòng 95 tới 97: các trường hợp khi đăng ký thất bại như là: điện thoại không có mạng dữ liệu, wifi; email nhập không đúng format.

4.1.3 Chức năng đăng nhập qua firebase

Tương tự với chức năng đăng kí, chức năng đăng nhập cũng khai báo các biến sử dụng và ánh xạ dữ liệu.



```
37 // sự kiện nút nhấn đăng nhập
38 btn_login.setOnClickListener(new View.OnClickListener() {
39     @Override
40     public void onClick(View view) { login(); }
41 });
42 // nút nhấn register tại giao diện register.xml
43 btn_register_toggle.setOnClickListener(new View.OnClickListener() {
44     @Override
45     public void onClick(View v) { register(); }
46 });
47
48 CheckLogin();
49 }
```

The code shows event listeners for two buttons: 'btn_login' and 'btn_register_toggle'. The 'btn_login' listener calls the 'login()' method. The 'btn_register_toggle' listener calls the 'register()' method. Finally, the 'CheckLogin()' method is called.

Hình 4.9 Các button bắt sự kiện, và các hàm sử dụng

Ở chức năng đăng nhập sẽ có 3 phần chính là:

- Button chuyển đổi đăng kí (tương tự như button chuyển đổi trang đăng nhập).
- Button đăng nhập sử dụng hàm login.
- Cuối cùng là hàm CheckLogin (để kiểm tra session đăng nhập).

```

54     public void login(){
55         String email, password;
56         email = edit_Txt_Email_login.getText().toString();
57         password = edit_Txt_Password.getText().toString();
58         if(TextUtils.isEmpty(email)){
59             Toast.makeText( context: this, text: "Vui lòng nhập email", Toast.LENGTH_SHORT).show();
60             return;
61         }
62         if(TextUtils.isEmpty(password)){
63             Toast.makeText( context: this, text: "Vui lòng nhập mật khẩu", Toast.LENGTH_SHORT).show();
64             return;
65         }
66         mAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
67             @Override
68             public void onComplete(@NonNull Task<AuthResult> task) {
69                 if(task.isSuccessful()){
70                     sessionManager.SetLogin( isLogin: true);
71                     Toast.makeText(getApplicationContext(), text: "Đăng nhập thành công", Toast.LENGTH_SHORT).show();
72                     Intent intent = new Intent( packageContext: login.this, room.class);
73                     startActivity(intent);
74                 }else{
75                     Toast.makeText(getApplicationContext(), text: "Đăng nhập không thành công", Toast.LENGTH_SHORT).show();
76                 }
77             }
78         });
79     }

```

Hình 4.10 Hàm đăng nhập bằng phương thức Authentication của firebase

Hàm login cũng sẽ kiểm tra 2 ô email và password nếu người dùng không nhập dữ liệu mà bấm button login sẽ hiển thị thông báo.

Nếu 2 ô **đã nhập thông tin** trùng khớp với **thông tin đã đăng kí** trước đó, thì sẽ cho biến SetLogin thành true để lưu session đăng nhập, thông báo thành công và sau đó chuyển sang trang dashboard.

Nếu 2 ô **đã nhập thông tin** không trùng khớp với **thông tin đã đăng kí** trước đó, thì sẽ thông báo cho người dùng.

```

86     private void CheckLogin(){
87         if(!sessionManager.Check()){
88             Toast.makeText( context: this, text: "Mời bạn đăng nhập", Toast.LENGTH_SHORT).show();
89         }else {
90             Intent intent = new Intent(getApplicationContext(),room.class);
91             startActivity(intent);
92             finish();
93         }
94     }

```

Hình 4.11 Hàm kiểm tra đăng nhập

Nếu không có session đăng nhập, thông báo cho người dùng. Ngược lại, có session đăng nhập chuyển thẳng sang trang dashboard.

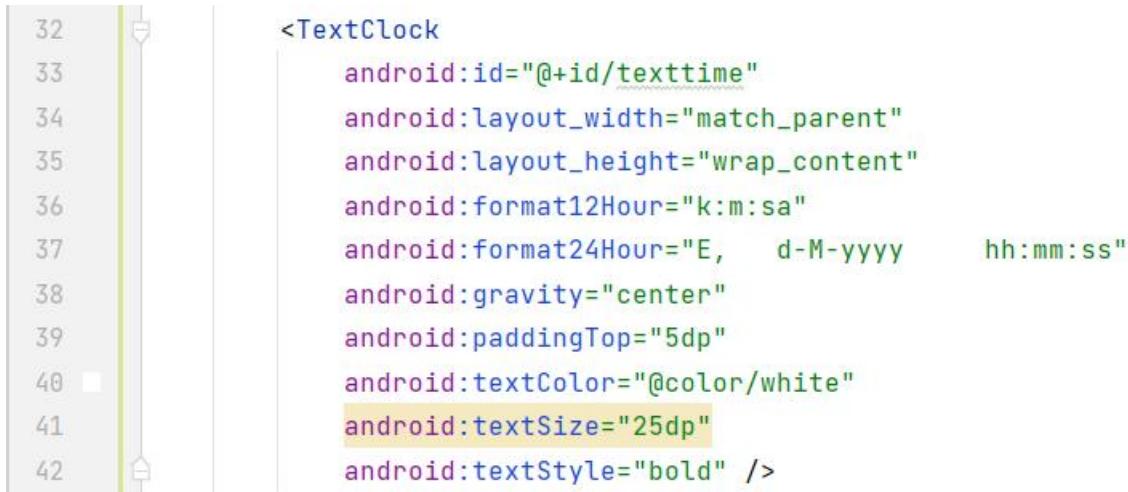
```
1 package com.example.app_dktb;
2
3 import ...
4
5
6 public class SessionManager {
7     private static String TAG = SessionManager.class.getName();
8
9     SharedPreferences preferences;
10    Context context;
11    SharedPreferences.Editor editor;
12    private static final String NAME = "android_session";
13    private static final String KEY_LOGIN = "islogin";
14
15    @ ...
16    public SessionManager(Context context){
17        Log.i(TAG, msg: "SessionManager: eqwwwww "+ TAG);
18        this.context = context;
19        preferences = context.getSharedPreferences(NAME, MODE_PRIVATE);
20        editor = preferences.edit();
21    }
22
23    public void SetLogin(boolean isLoggedIn){
24        editor.putBoolean(KEY_LOGIN, isLoggedIn);
25        editor.commit();
26    }
27
28    public boolean Check() { return preferences.getBoolean(KEY_LOGIN, defValue: false); }
29
30
31 }
```

Hình 4.12 File setup cho hàm kiểm tra đăng nhập

File setup này sử dụng phương thức SharedPreferences, hàm SessionManager giúp lưu session đăng nhập, hàm SetLogin bắt kiểu dữ liệu boolean true hoặc false. Nếu đã đăng nhập SetLogin = true, chưa đăng nhập SetLogin = false, hàm Check để kiểm tra session bên hàm CheckLogin bên file Login.java.

4.2. Hiển thị ngày, giờ thực tế

Đầu tiên, ta cần khai báo một TextClock trong file .xml là nơi cần hiển thị các giá trị ngày giờ thực tế.



Hình 4.13 TextClock hiển thị giờ thực tế

Thực hiện khai báo TextClock:

```
38     TextClock time;  
39     private static final String TIME_FORMAT_24 = "HH:mm:ss";  
40
```

Hình 4.14 Khai báo TextClock

Dòng 39: khai báo định dạng giờ có thể theo 2 loại là 12h hoặc 24h.

```
121     // hiển thị thời gian  
122     String formatdate = "E, d-M-yyyy hh:mm:ss";  
123     time.setFormat24Hour(formatdate);  
124
```

Hình 4.15 Hiển thị thời gian

Dòng 122: khai báo các giá trị cần hiển thị trên TextClock gồm:

- E : là thứ
- d : là ngày
- M: là tháng
- y : là năm
- h : là giờ

- m: là phút

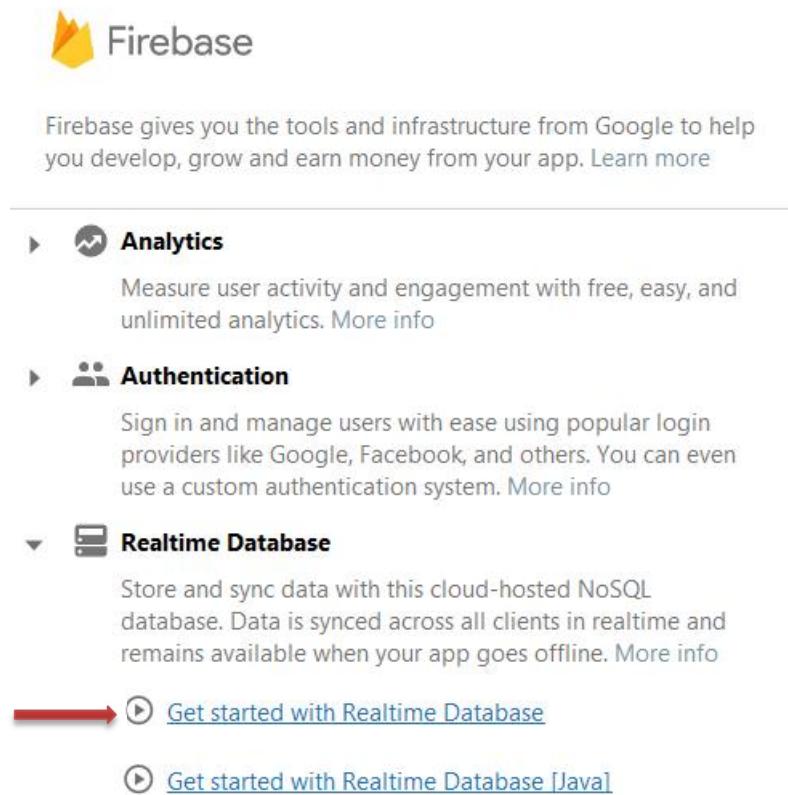
- s :là giây

Dòng 123: thời gian được cài đặt theo kiểu 24h

4.3. Đọc giá trị nhiệt độ, độ ẩm qua DHT11, led

4.3.1 Kết nối firebase với Android Studio

Mở phần mềm Android Studio, thanh công cụ chọn Tools sau đó chọn Firebase



Hình 4.16 Cách thức để chọn phương thức firebase

The screenshot shows the Firebase Realtime Database setup interface. At the top, there's a navigation bar with 'Assistant' and 'Firebase' tabs, and a settings icon. Below the navigation is a breadcrumb trail: '← Firebase > Realtime Database'. The main title is 'Get started with Realtime Database'. A descriptive text states: 'The Firebase Realtime Database is a cloud-hosted database. Data is stored and synchronized in realtime to every connected client.' Below this is a 'Launch in browser' button. The first step, '① Connect your app to Firebase', is shown with a 'Connect to Firebase' button and a green checkmark indicating '1 of 2 modules completed'. The second step, '② Add the Realtime Database to your app', is shown with a 'Add the Realtime Database SDK to your app' button and a green checkmark indicating '4 of 8 modules have dependencies added'. A note below says: 'NOTE: After adding the SDK, here are some other helpful configurations...'. There's also a section about managing library versions.

Hình 4.17 Kết nối Android Studio với Realtime database

Kết nối firebase phương thức realtime database với Android Studio để hiển thị giá trị nhiệt độ, độ ẩm của DHT11.

4.3.2 Kết nối firebase với Arduino IDE



```
dktb | Arduino 1.8.19
File Edit Sketch Tools Help
dktb
1 #include <ESP8266WiFi.h> // esp8266 library
2 #include <FirebaseESP8266.h> // firebase library
3 #include <DHT.h> // dht11 temperature and humidity sensor library
4 #define FIREBASE_HOST "https://appdktb-default-rtdb.firebaseio.com/" // the project name address from firebase id
5 #define FIREBASE_AUTH "yxirOHZRTNazXhmZEzgpRt3rQ4bxm2x5gtgRmqa" // the secret key generated from firebase
6
7 #define WIFI_SSID "TIEU THANH 95" // input your home or public wifi name
8 #define WIFI_PASSWORD "hoichizay" //password of wifi ssid
9
10#define DHTPIN D4 //chân gpio 2 = D4 // what digital pin we're connected to
11#define DHTTYPE DHT11 // select dht type as DHT 11 or DHT22
12
13
14 DHT dht(DHTPIN, DHTTYPE);
15 FirebaseData firebaseData;
16 int Den_liv = D0;
17 int Den_bed = D2;
18 int Den_kit = D3;
19 float h,t;
```

Hình 4.18 Định nghĩa thư viện, wifi, firebase, chân esp8266

Từ dòng 1 tới 3: thêm 3 thư viện của esp, dht và firebase.

Dòng 4, 5: Kết nối firebase

Dòng 7,8: Kết nối wifi và password

Dòng 10,11,14: Định nghĩa chân cho DHT11, chân data output nối với chân gpio2 là chân D4 trên ESP8266. (chân + nối với chân 3v3 của esp; chân - nối với GND của esp).

Dòng 16-18: Định nghĩa chân led để cảm esp 8266

Dòng 19: Định nghĩa nhiệt độ độ ẩm

```

20 void setup() {
21     // put your setup code here, to run once:
22     Serial.begin(9600);
23     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
24     while (WiFi.status() != WL_CONNECTED) {
25         delay (500);
26         Serial.print(".");
27     }
28     dht.begin();
29     Serial.println("");
30     Serial.println("connected");
31     Serial.println(WiFi.localIP());
32     Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
33     pinMode(Den_liv, OUTPUT);
34     pinMode(Den_bed, OUTPUT);
35     pinMode(Den_kit, OUTPUT);
36 }
37
38
39 }
40

```

Hình 4.19 Định nghĩa chân esp của led, kết nối wifi, firebase

Dòng 22 đến 33: Kết nối wifi

Dòng 35-37: Khai báo OUTPUT của led, led1, led2

Dòng 34: Kết nối firebase

```

49 void DocCamBien()
50 {
51     h = dht.readHumidity();
52     t = dht.readTemperature();
53
54     if (isnan(h) || isnan(t))
55     {
56         Serial.println("Failed to read from DHT sensor!");
57         return;
58     }
59
60     Serial.print("Nhiệt độ: ");
61     Serial.print(t);
62     Serial.print("*C ");
63     Serial.print("Độ ẩm: ");
64     Serial.print(h);
65     Serial.println("% ");
66     Firebase.setFloat (firebaseData ,"/DHT11/Nhiệt độ", t);
67     Firebase.setFloat (firebaseData ,"/DHT11/Độ ẩm", h);
68     delay(200);
69 }
70

```

Hình 4.20 Đọc giá trị DHT11 và LED

Dòng 51,52: Khai báo giá trị của DHT đọc được.

Dòng 54 tới 58: Khi không được giá trị DHT.

Dòng 60 tới 65: Hiển thị giá trị, thông tin trên Serial Monitor.

Dòng 66, 67: Đọc giá trị theo đường dẫn trên database.

```

71 void PhongLiv()
72 {
73     if(Firebase.get(firebaseData,"/Living_Room/Lamp"))
74     {
75         if(firebaseData.dataType() == "string")
76         {
77             String den_liv = firebaseData.stringValue();
78             if(den_liv == "ON")
79             {
80                 Serial.println("den mo");
81                 digitalWrite(Den_liv,HIGH);
82             }
83             else if (den_liv == "OFF")
84             {
85                 Serial.println("den tat");
86                 digitalWrite(Den_liv,LOW);
87             }
88         }
89     }
90 }
91 }
```

Hình 4.21 Phòng livingroom

Từ dòng 71 tới 86: Kiểm tra giá trị trên database, nếu giá trị LED trên database “ON” thì đây giá trị chân D4 lên HIGH (đèn sáng), else if thì ngược lại, else hiển thị khi không đọc được giá trị database ở phòng khách

```

92
93 void PhongKit()
94 {
95     if(Firebase.get(firebaseData,"/Kitchen_Room/Light"))
96     {
97         if(firebaseData.dataType() == "string")
98         {
99             String den_kit = firebaseData.stringValue();
100            if(den_kit == "ON")
101            {
102                Serial.println("den mo");
103                digitalWrite(Den_kit,HIGH);
104            }
105            else if (den_kit == "OFF")
106            {
107                Serial.println("den tat");
108                digitalWrite(Den_kit,LOW);
109            }
110        }
111    }
112 }
113 }
114 }
```

Hình 4.22 Phòng kitchenroom

Từ dòng 93 tới 108: Kiểm tra giá trị trên database, nếu giá trị LED trên database “ON” thì đây giá trị chân D4 lên HIGH (đèn sáng), else if thì ngược lại, else hiển thị khi không đọc được giá trị database ở phòng bếp

```

115
116 void PhongBed()
117 {
118     {
119         if(Firebase.get(firebaseData, "/Bed_Room/Lamp"))
120     {
121         if(firebaseData.dataType() == "string")
122         {
123             String den_bed = firebaseData.stringData();
124             if(den_bed == "ON")
125             {
126                 Serial.println("den mo");
127                 digitalWrite(Den_bed, HIGH);
128             }
129             else if (den_bed == "OFF")
130             {
131                 Serial.println("den tat");
132                 digitalWrite(Den_bed, LOW);
133             }
134         }
135     }
136 }
137
138 }

```

Hình 4.23 Phòng bedroom

Từ dòng 116 tới 132: Kiểm tra giá trị trên database, nếu giá trị LED trên database “ON” thì đẩy giá trị chân D4 lên HIGH (đèn sáng), else if thì ngược lại, else hiển thị khi không đọc được giá trị database ở phòng bếp

```

40
41 void loop()
42 {
43     DocCamBien();
44     PhongLiv();
45     PhongBed();
46     PhongKit();
47 }
48

```

Hình 4.24 Vòng lặp chạy

4.4. Điều khiển hình ảnh qua realtime database

Đầu tiên, ta cần có những file .xml hiển thị thiết bị để có thể thay đổi hình ảnh khi bật ON/OFF. Sau đó ta tiến hành khai báo và ánh xạ nó:

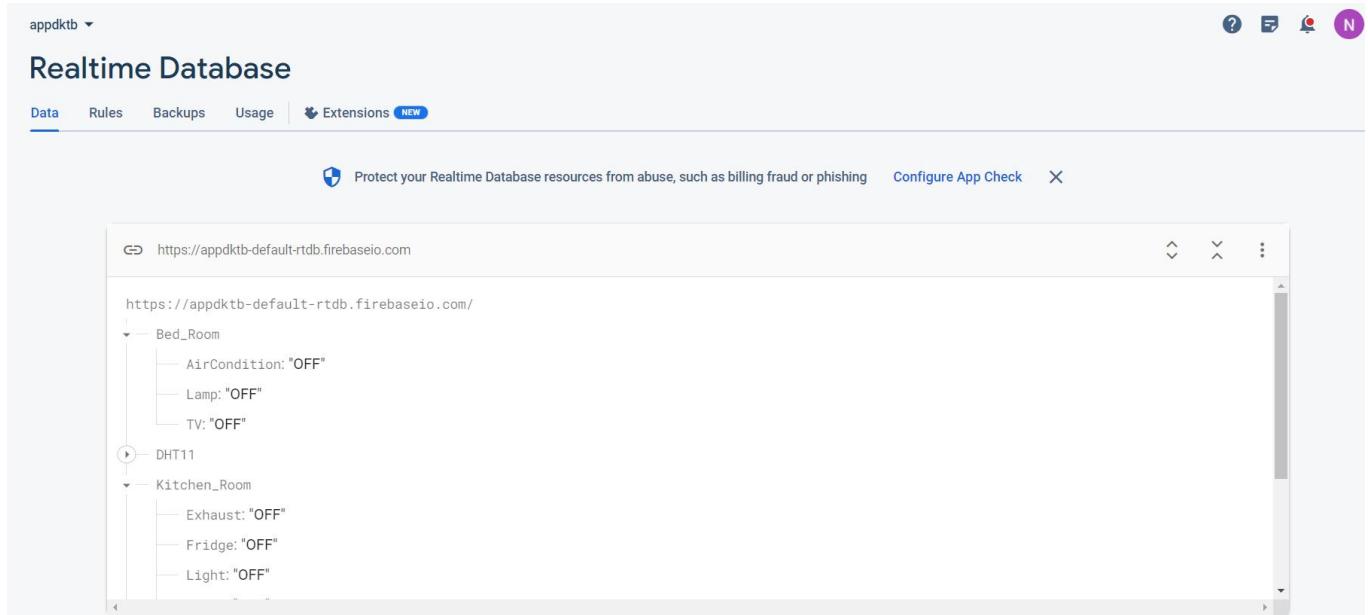
```
29     ImageView TV_icon,Lamp_icon,Camera_icon,Fan_icon;
30     Switch sw_TV, sw_Lamp, sw_Camera,sw_Fan;
31     boolean stateTV=false;
32     boolean stateLamp=false;
33     boolean stateCamera=false;
34     boolean stateFan=false;
35     @Override
36     protected void onCreate(Bundle savedInstanceState) {
37         super.onCreate(savedInstanceState);
38         setContentView(R.layout.livingroom);
39         TV_icon=findViewById(R.id.tiviicon);
40         Lamp_icon=findViewById(R.id.lampicon);
41         Camera_icon=findViewById(R.id.cameraicon);
42         Fan_icon=findViewById(R.id.fanicon);
43         sw_TV=findViewById(R.id.swTivi);
44         sw_Lamp=findViewById(R.id.swLamp);
45         sw_Camera=findViewById(R.id.swCamera);
46         sw_Fan=findViewById(R.id.swFan);
47
48         FirebaseDatabase database=FirebaseDatabase.getInstance();
49         DatabaseReference databaseReference = database.getReference();
```

Hình 4.25 Khai báo biến và ánh xạ

Ở đây, mỗi 1 thiết bị cần điều khiển ON/ OFF thay đổi hình ảnh thì cần nơi hiển thị ảnh (ví dụ các ImageView: TV_icon, Lamp_icon, Camera_icon, Fan_icon) và nút nhấn Switch để thực hiện ON/OFF và các trạng thái của thiết bị.

Dòng 48: kết nối tới firebase app qua .getInstance

Dòng 49: biến databaseReference được lấy từ phần realtime database trên firebase.



Hình 4.26 Cấu trúc cây trên firebase

Hình trên là phần thiết lập realtime database trên firebase của nhóm mình gồm 4 nhánh chính là:

- DHT11 : gồm Nhiệt do và Độ ẩm
- Living_Room: gồm 4 thiết bị là Camera, Fan, Lamp, TV
- Bed_Room: gồm 3 thiết bị là AirCondition, Lamp, TV
- Kitchen_Room: gồm 4 thiết bị là Exhaust, Fridge, Light, Stove

Sau đây là phần chi tiết kết nối ON/OFF thay đổi hình ảnh trên App và thay đổi giá trị trên FireBase. Do cách thiết lập của các thiết bị là giống nhau nên em sẽ trình bày về 1 thiết bị TV trong phòng Living_Room.

```
53 databaseReference.child("Living_Room").child("TV").addValueEventListener(new ValueEventListener
54 {
55     @Override
56     public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
57         String value= dataSnapshot.getValue().toString();
58         if(value.equals("ON")){
59             stateTV=true; //bật TV
60             TV_icon.setImageResource(R.drawable.tv_on);
61             sw_TV.setChecked(true);
62         }
63         else if(value.equals("OFF")){
64             stateTV=false; //tắt TV
65             TV_icon.setImageResource(R.drawable.tv);
66             sw_TV.setChecked(false);
67         }
68         //switch thay đổi tv on off
69         sw_TV.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
70             @Override
71             public void onCheckedChanged(CompoundButton compoundButton, boolean isChecked) {
72                 if(isChecked) {
73                     stateTV=true;
74                     TV_icon.setImageResource(R.drawable.tv_on);
75                     databaseReference.child("Living_Room").child("TV").setValue("ON");
76                 }
77                 else {
78                     stateTV=false;
79                     TV_icon.setImageResource(R.drawable.tv);
80                     databaseReference.child("Living_Room").child("TV").setValue("OFF");
81                 }
82             }
83         });
84     }
85 }
```

Hình 4.27 Code chuyển đổi hình ảnh

Dòng 53: từ biến databaseReference ta trả đến địa chỉ con “Living_Room” và trả tiếp đến địa chỉ con “TV” để tiến hành đọc giá trị của nó trong phần realtime database.

Dòng 55: khai báo hàm thay đổi giá trị trong Firebase ảnh hưởng tới app

Dòng 56: khai báo biến ‘value’ bằng giá trị lấy từ địa chỉ con “TV” dạng string.

Dòng 57: hàm điều kiện value là chữ ‘ON’

Dòng 58: gán stateTV =true

Dòng 59: TextView của TV là TV_icon hiển thị hình ảnh TV bật

Dòng 60: trạng thái của Switch là .isChecked(true) tức bật

Dòng 62 -65: điều kiện value là chữ ‘OFF’ thì trạng thái của TV là false, TextView của TV là TV_icon hiển thị hình ảnh TV tắt, trạng thái của Switch là .isChecked(true) tức tắt.

Dòng 68: điều khiển bật tắt của Switch ở App làm thay đổi các giá trị trên Firebase

Dòng 70: hàm OnCheckedChanged là hàm set sự thay đổi của Switch

Dòng 71-75: hàm điều kiện isChecked tức Switch bật thì trạng thái của TV là true, TextImage của TV là TV_icon hiển thị hình ảnh TV bật và đặt giá trị value của địa chỉ con “TV” là ‘ON’.

Dòng 76-80: hàm điều kiện không checked tức Switch tắt thì trạng thái của TV là false, TextImage của TV là TV_icon hiển thị hình ảnh TV tắt và đặt giá trị value của địa chỉ con “TV” là ‘OFF’.

Vậy là ta đã hoàn thành điều khiển ON/OFF thiết bị trên App làm thay đổi hình ảnh thiết bị và giá trị trên FireBase đồng thời khi thay đổi giá trị ON/OFF trên FireBase thì thiết bị trên App cũng thực hiện ON/OFF thay đổi hình ảnh.

4.5 Đánh giá những kết quả đã thực hiện được

- Thiết lập kết nối mạng Wifi thành công: Đã thiết kế và triển khai một mạng Wi-Fi ổn định để kết nối ESP8266 với các thiết bị trong nhà. Điều này cũng đảm bảo rằng ESP8266 của nhóm có thể kết nối và giao tiếp với ứng dụng điều khiển từ xa.

- Xây dựng ứng dụng điều khiển thiết bị: Đã phát triển một ứng dụng điều khiển thiết bị thông qua ESP8266. Ứng dụng này cho phép người dùng từ xa điều khiển các thiết bị trong nhà thông qua giao diện người dùng đơn giản và thuận tiện. Các chức năng điều khiển như bật/tắt thiết bị đèn, quạt, cảm biến nhiệt độ/độ ẩm và theo dõi trạng thái đã được triển khai và hoạt động đúng như mong đợi.

- Kiểm thử và đánh giá: Đã tiến hành kiểm thử hệ thống để đảm bảo tính ổn định. Các kịch bản kiểm thử đã được thực hiện để kiểm tra chức năng, tương thích và đáng tin cậy của hệ thống

-> Về cơ bản nhóm tự nhận xét phần mềm của nhóm đã giải quyết được hầu như các yêu cầu cơ bản mà đồ án đã đề ra. Còn về phần cứng có phần hạn chế về các thiết bị

Video demo kết quả: <https://www.youtube.com/watch?v=iP462x5GdNk>

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Nhóm nhận thấy qua đồ án 1 đã đóng góp vào việc xây dựng một hệ thống điều khiển thiết bị trong nhà thông qua ESP8266 mang lại sự tiện ích và linh hoạt cho người dùng trong thời buổi công nghiệp hóa đang diễn ra

5.2 Định hướng phát triển

Trong tương lai, hệ thống sẽ được phát triển theo hướng khắc phục những hạn chế hiện tại nêu trên. Bên cạnh đó sẽ bổ sung thêm nhiều chức năng hơn nữa để tăng tính hiệu quả và giúp cho người dùng sử dụng an toàn bảo mật hơn

TÀI LIỆU THAM KHẢO

- [1]- Visual Studio Code: <https://visualstudio.microsoft.com/fr/#vscode-section>
- [2]- Video học lập trình Android kênh Khoa Phạm:

https://www.youtube.com/playlist?list=PL5uqQAwS_KDjAgLGiaCakwJV1f4vRnTLS