**BÁO CÁO PROJECT**

**Đề tài T12: Smart Home-Control and Monitoring System Using Smart Phone**

|  |  |
| --- | --- |
| **Người thực hiện:** | **Nguyễn Văn Tiệm Nguyễn Đức Trọng Nguyễn Văn Khoa** |

|  |  |
| --- | --- |
| **Mentor hướng dẫn:** | **Phùng Đức Vũ Nguyễn Tiến Tuyên Trần Trung Hiếu** |
|  |
|  | | |

Hà Nội, 06/2021

# LỜI NÓI ĐẦU

Trong thời buổi số hóa cả thế giới đang tiến tới cuộc cách mạng 4.0 việc ứng dụng công nghệ, máy tính vào hỗ trợ con người ngày càng trở nên phổ biến trong đó lĩnh vực kết nối vạn vật là một trong những lĩnh vực quan trọng và được chú trọng.

Với ưu điểm nhỏ gọn giá thành rẻ máy tính nhúng đang phát huy vai trò của mình đối với những ứng dụng không đòi hỏi tốc độ và bộ nhớ quá cao, máy tính nhúng đang phát huy vai trò nổi trội của mình trong việc hỗ trợ công việc cho con người trong việc theo dõi sức khỏe, nhà thông minh, kết nối các thiết bị nhằm làn cho cuộc sống con người trở nên tiện dụng.

Đi cùng với sự phát triển của công nghệ thời gian gần đây lĩnh vực phát triển nhà thông minh là một trong những lĩnh vực được quan tâm nó giúp cho người dùng có thể tương tác, quản lý ngôi nhà của mình tốt hơn từ đó nâng cao chất lượng cuộc sống.

Chúng tôi dựa trên những nhu cầu thực tế và xu thế trong tương lai chúng tôi đã quyết định lựa chọn đề tài Smart Home-Control and Monitoring System Using Smart Phone.

Trong giới hạn của đồ án, mặc dù tác giả đã cố gắng nhưng đề tài vẫn không thể tránh khỏi nhiều thiếu sót, rất mong nhận được sự góp ý của anh (chị)

# TÓM TẮT ĐỒ ÁN

Trong nội dung đồ án chúng tôi xác định công việc đối với việc phát triển nhà thông minh bao gồm:

Lựa chọn các thiết bị phần cứng nhúng để trực tiếp điều khiển các thiết trong nhà, từ đó xây dựng cơ cấu phần cứng và phát triển firmware cho thiết bị

Xây dựng một server quản lí thông tin tất cả các thiết bị cũng như tài khoản đã đăng kí đối với nhà cung cấp từ đó cung cấp thông tin cho người dùng tương tác với các thiết bị trong ngôi nhà của họ.

Xây dựng ứng dụng android cung cấp giao diên tiện lợi cho người dùng giúp họ kết nối và dễ điều khiển và quản lí thiết bị trong nhà của họ hơn.

Nội dung đồ án gồm 4 chương chính sau:

* Chương 1: Đặt vấn đề và hướng giải quyết.
* Chương 2: Xây dựng phần cứng và phát triển firmware thiết bị.
* Chương 3: Xây dựng Server và cơ sở dữ liệu quản lí
* Chương 4: Phân tích thiết kế và phát triển ứng dụng trên android
* Chương 5: Đánh giá kết quả

# 

# MỤC LỤC

[CHƯƠNG 1: ĐẶT VẤN ĐỀ VÀ GIẢI PHÁP 6](#_Toc73950286)

[1. Đặt vấn đề. 6](#_Toc73950287)

[2. Giải pháp. 6](#_Toc73950288)

[2.1. Giải pháp phần cứng. 6](#_Toc73950289)

[2.2. Giải pháp về giao thức giao tiếp giữa các modle 7](#_Toc73950290)

[3. Giới thiệu về Socket io 7](#_Toc73950291)

[CHƯƠNG 2: PHẦN CỨNG VÀ PHÁT TRIỂN FIRMWARE 9](#_Toc73950292)

[1. Hoạt động phần cứng. 9](#_Toc73950293)

[2. Phát triển firmware. 10](#_Toc73950294)

[2.1. Cấu hình thiết bị 11](#_Toc73950295)

[2.2. Giao tiếp socket io giữa thiết bị và server 11](#_Toc73950296)

[CHƯƠNG 3: XÂY DỰNG SERVER VÀ CƠ SỞ DỮ LIỆU 13](#_Toc73950297)

[1. Xây dựng server 13](#_Toc73950298)

[CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG ANDROID 19](#_Toc73950299)

[1. Chức năng chính của ứng dụng 19](#_Toc73950300)

[2. Thiết kế Activity 20](#_Toc73950301)

[3.1. Splash Screen 20](#_Toc73950302)

[3.2. Activity Đăng nhập/Đăng kí 20](#_Toc73950303)

[3.3. Activity chính 21](#_Toc73950304)

[3. Thiết kế Fragment 22](#_Toc73950305)

[3.1. Fragment Home 22](#_Toc73950306)

[3.2. Fragment phòng 23](#_Toc73950307)

[3.3. Fragment thêm phòng 24](#_Toc73950308)

[3.4. Fragment thiết bị 25](#_Toc73950309)

[3.5. Fragment thêm thiết bị 26](#_Toc73950310)

[3.6. Fragment thiết bị đang hoạt động 28](#_Toc73950311)

[3.7. Fragment yêu thích 29](#_Toc73950312)

[3.8. Fragment setting 29](#_Toc73950313)

[4. Navigation Graph 30](#_Toc73950314)

[CHƯƠNG 5: KẾT QUẢ ĐẠT ĐƯỢC VÀ ĐÁNH GIÁ 32](#_Toc73950315)

# CHƯƠNG 1: ĐẶT VẤN ĐỀ VÀ GIẢI PHÁP

1. Đặt vấn đề.

Như đã trình bày ở trên hệ thống bao gồm 3 module chính:

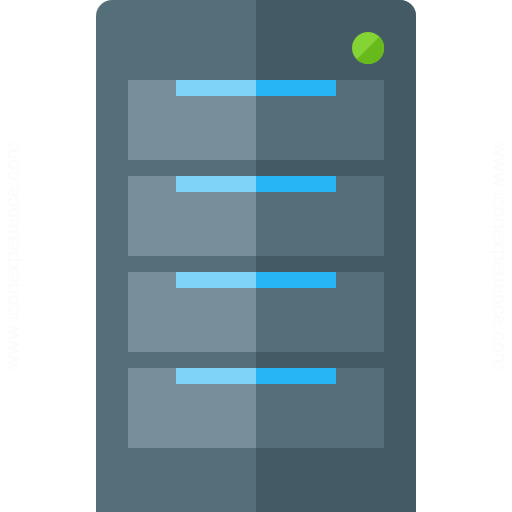
Home

Server

*Get*

*Push*

User





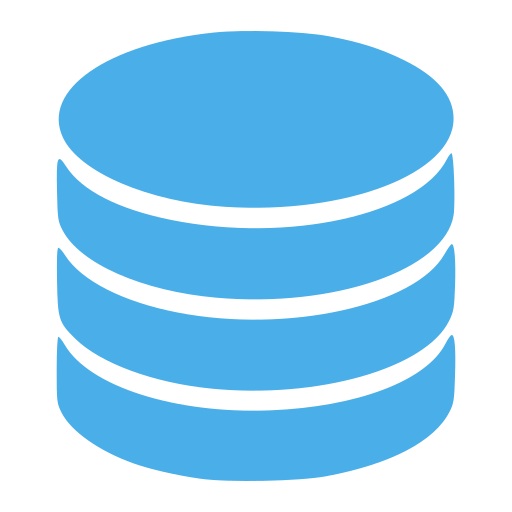


Decive

*Control*

get

App android



database

Với mục đích giúp cho người dùng quản lí ngôi nhà của mình mọi lúc mọi nơi thì hệ thống phải đáp ứng được những yêu cầu sau: các module phải giao tiếp thười gian thực , khi có sự thay đổi tại một thiết bị cả hệ thống sẽ được cập nhật, người dùng có thể điều khiển thiết bị trong nhà ở mọi nơi tức không bị phụ thuộc vào mạng cục bộ, cuối cùng hệ thống phải đảm bảo khả năng tiện lợi dễ sử dụng có tính mở rộng cao.

1. Giải pháp.

### Giải pháp phần cứng.

Sử dụng máy tính nhúng esp8266 để điều khiểu thiết bị trong nhà, với ưu điểm nhỏ gọn, giá thành rẻ, có khả năng kết nối giao tiếp qua wifi.

Thông số phần cứng:

* IC chính: ESP8266
* Phiên bản firmware: NodeMCU Lua
* Chip nạp và giao tiếp UART: CP2102.
* GPIO tương thích hoàn toàn với firmware Node MCU.
* Cấp nguồn: 5VDC MicroUSB hoặc Vin.
* GIPO giao tiếp mức 3.3VDC
* Tích hợp Led báo trạng thái, nút Reset, Flash.
* Tương thích hoàn toàn với trình biên dịch Arduino.
* Kích thước: 25 x 50 mm



### Giải pháp về giao thức giao tiếp giữa các modle

Phía người dùng giao tiếp với các thiết bị thông qua server các thông tin về thiết bị được server lưu trữ trong cơ sở dữ liệu. Server được phát triển bằng node js và sử dụng my sql để quản lí dữ liệu. Để đảm bảo tính thời gian thực các module giao tiếp dựa trên giao thức socket io.

1. Giới thiệu về Socket io

Socket io là gì ?

Nó được xem là 1 module trong Node.js được nhà sáng chế tạo ra và phát triển từ năm 2010. Mục đích lớn nhất của Socket io là để tạo môi trường giao tiếp thuận lợi trên Internet giúp trả về các giá trị thực ngay tại thời điểm giao tiếp giữa các bên với nhau (thường là giữa server và client).

Việc giao tiếp 2 chiều giữa máy khách và máy chủ được thực hiện bởi socket io khi và chỉ khi máy khách có module này trong trình duyệt và máy chủ cũng đã tích hợp sẵn gói socket io. Các ứng dụng sử dụng socket io thường đòi hỏi tốc độ phản hồi ngay lập tức. Một số ví dụ điển hình như xổ số, trực tiếp bóng đá, chat…

Socket io có những ưu điểm gì?

* Bảo mật cao

Socket io được xây dựng dựa trên Engine.IO. Nó sẽ khởi chạy phương thức long-polling trước nhất để kết nối. Sau đó nó mới sử dụng các phương thức giao tiếp tốt hơn như là Websocket chẳng hạn. Vì được thiết lập chặt chẽ như vậy nên khi socket io xuất hiện nó sẽ tự động tạo những kết nối bảo mật như là: proxy và cân bằng tải hoặc là tường lửa cá nhân và phần mềm chống vi rút.

* Kết nối tự động tới server

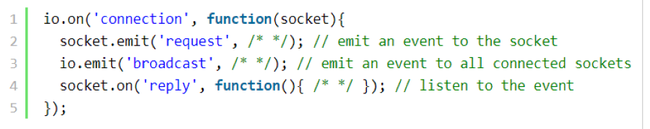
Đặc điểm tự động kết nối đến server của socket io là gì? Giả sử trong quá trình khởi chạy bị mất kết nối giữa client và server thì socket io sẽ tự động gắn kết nối mãi mãi cho đến khi nào server phản hồi lại. Và đây là tính năng có thể tùy chỉnh được nên bạn có quyền chọn không kết nối tự động đến bất kỳ server nào mà mình muốn.

* Mã hóa nhị phân

Socket io có thể hỗ trợ mã hóa nhị phân như ArrayBuffer và Blob trên trình duyệt hoặc là ArrayBuffer và Buffer trong Node.js.

* Cho phép tạo kênh và phòng

Đặc điểm cho phép tạo kênh và phòng của Socket.io là gì? Có thể nói đây là 1 tính năng khá nổi bật khi mà socket io có thể tạo ra mối quan hệ giữa các phần hoặc các module riêng lẻ bằng cách tạo ra những kênh riêng biệt khác nhau. Ngoài việc tạo kênh, nó còn hỗ trợ tạo phòng cho các clients tham gia với mục đích gửi thông báo đến 1 nhóm người dùng được kết nối với 1 số thiết bị nào đó chẳng hạn. Trong đó có thể kể đến 1 số các API đơn giản như hình dưới đây.



# CHƯƠNG 2: PHẦN CỨNG VÀ PHÁT TRIỂN FIRMWARE

1. Hoạt động phần cứng.

Với vấn đề xây dựng phần cứng trước hết cần phải trả lời câu hỏi các thiết bị trong nhà gồm những thiết bị nào? Mục đích nhu cầu và cách thức hoạt động ra sao? Trong dự án này nhóm xác định gồm các thiết bị sau: lamp, light bulb, fluorescent lamp, air conditioner, lock, temperature sensor, humidity sensor,fan, camera, assistant, door, window, other.

Từ đó có thể xác định các định cơ chế điều khiển chủ yếu là Bật/tắt, đọc digital (không sử dụng các chuẩn giao tiếp đặc biệt nào khác).

Đối với thiế bị cảm biến như : cảm biến nhiệt độ, độ ẩm, báo cháy, khí ga …

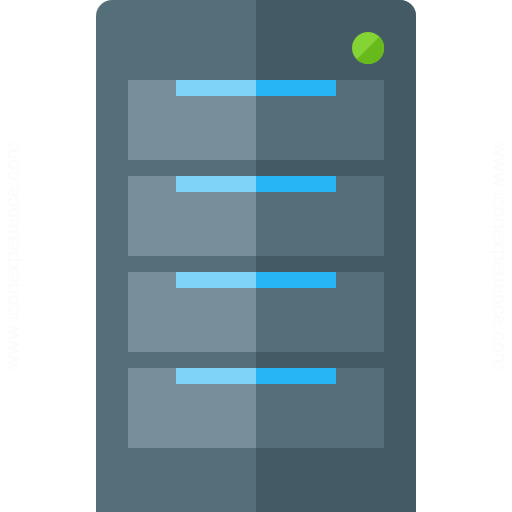
Thiết bị sẽ thu thập dữ liệu và gửi về server thời gian thực sử dụng giao thức socket io



*Push repeat 2s*

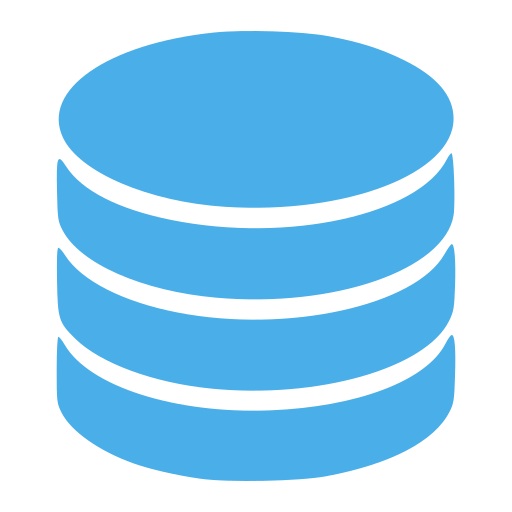
*Pin4*

Senor



*Update*

*Thu thập dữ liệu*

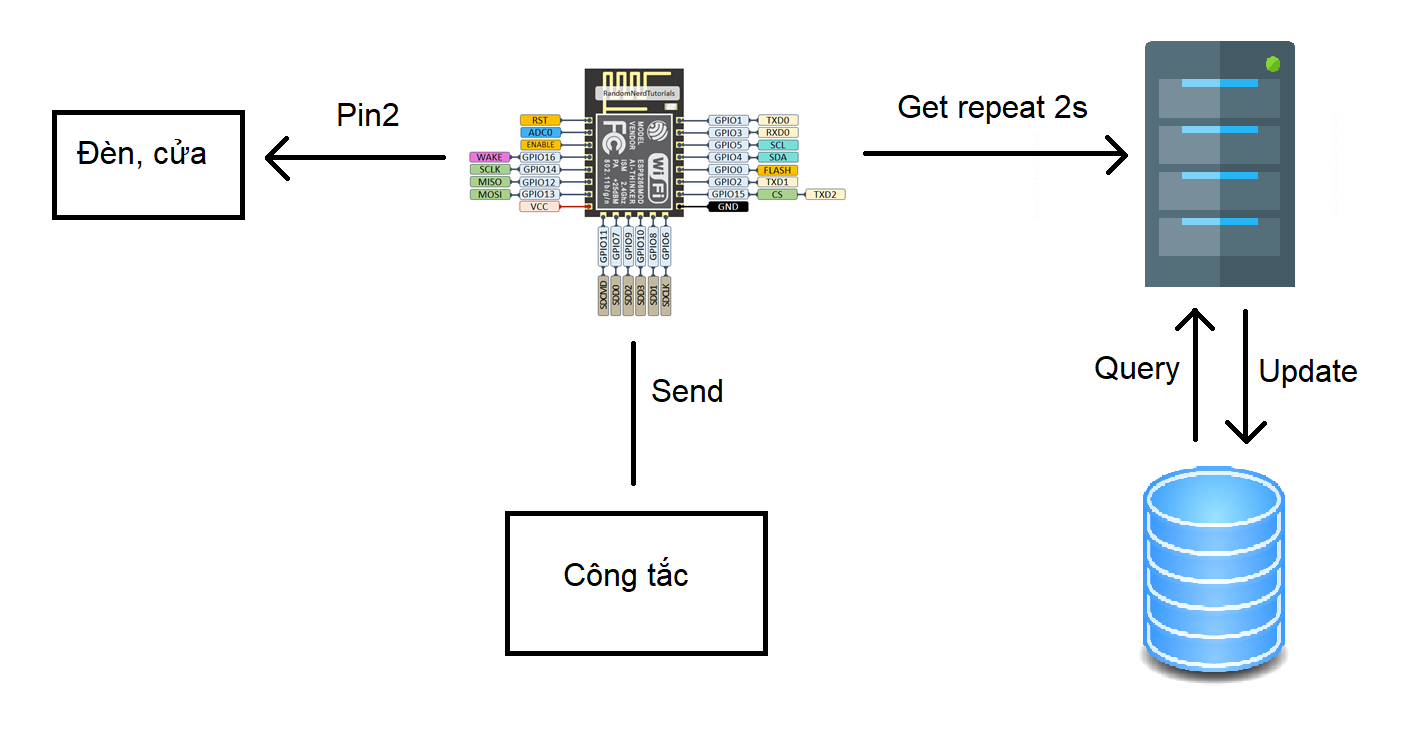


Thiết bị sẽ sử dụng 1 pin để lấy dữ liệu từ cảm biến ở đây dùng pin 4 việc pin sẽ được nhà sản xuất quyết định trong quá trình tích hợp sản xuất thiết bị. Sau khi có được dữ liệu về cảm biến thiết bị sẽ đóng gói và push lên server với gói tin gồm ID thiết bị và giá trị đo được, sau khi server nhận được dữ liệu sẽ dựa vào ID thiết bị và cập nhật giá trị vào database.

Đối với thiết bị điều khiển: thiết bị điều khiển có thể điều khiển thủ công tại nhà bằng công tắc và có thể điều khiển qua ứng dụng,

Trong tình huống bình thường thiết bị sẽ lấy dữ liệu từ server và từ đó lấy ra thông tin thiết bị và điều khiển và cập nhật trạng thái của công tắc.

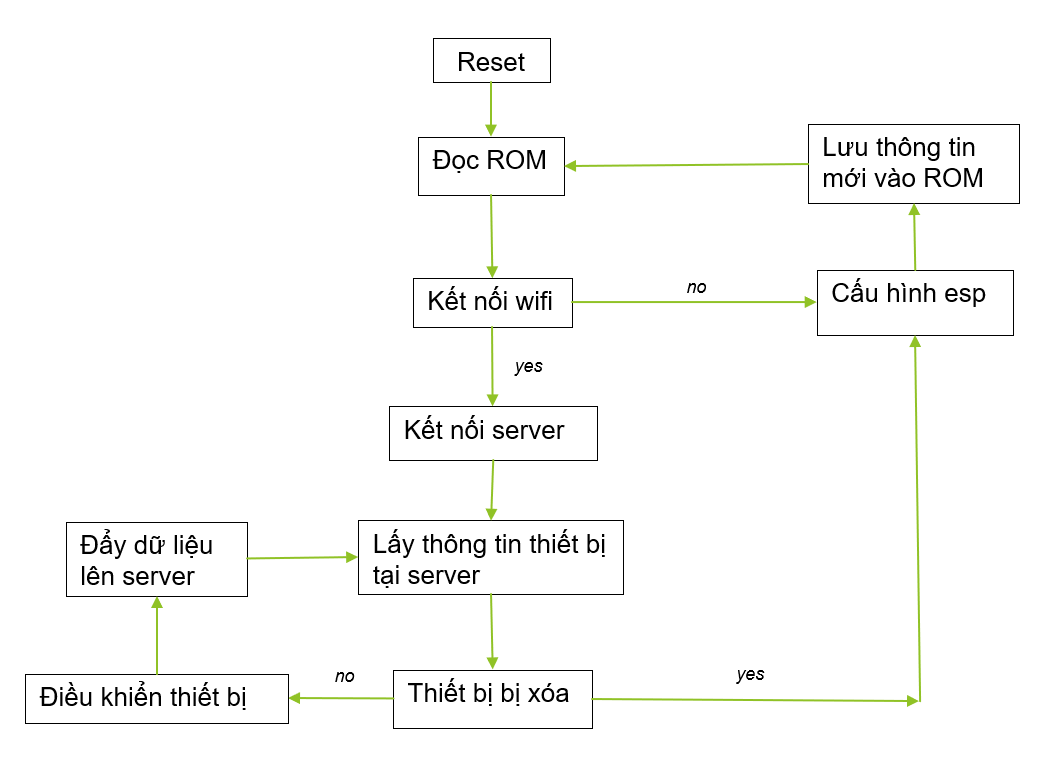
Trong trường hợp người dùng tắt thiết bị thủ công bằng công tắc tại nhà thiết bị dựa vào trạng thái hiện tại của đèn và thay đổi trạng thái tương ứng sau đó push trạng thái mới của thiết bị lên server, server nhận được sẽ update trạng thái đó vào database.



1. Phát triển firmware.

Thiết bị ban đầu được phát triển là thiết bị trắng sau khi người dùng mua về sẽ tiến hành cấu hình thành thiết bị mình muốn sử dụng

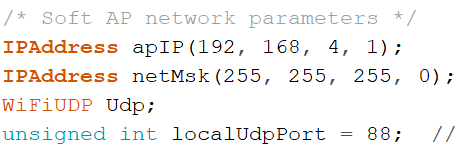
Trong trường hợp thiết bị không kết nối được wifi hoặc đã bị xóa thiết bị sẽ tự trở về trạng thái cấu hình, các thông tin cấu hình sẽ được lưu trong ROM :



Trong việc phát triển firmware cho thiết bị có 2 chức năng cần chú ý: cấu hình thiết bị và việc giao tiếp giữa thiết bị và server .

### Cấu hình thiết bị

Trong chế độ cấu hình thiết bị sẽ ở chế độ **Access Point** với địa chỉ ip cấu hình tĩnh sẵn . các thiết bị sẽ kết nối vào tiến hành cấu hình cho thiết bị bằng giao thức UDP



Các thông tin cần thiết cho việc cấu hình là tên Wifi, mật khẩu wifi, link đến server, tên thiết bị cần cấu hình, id thiết bị, loại thiết bị

### Giao tiếp socket io giữa thiết bị và server

Giao thức socket io là việc truyền dữ liệu real time vì vậy các hoạt động giao tiếp diễn ra bất đồng bộ . phía bên nhận sẽ mở trạng thái nghe một sự kiện khi có dữ liệu gửi đến sẽ tiến hành thực hiện một chức năng nào đó bằng hàm được cung cấp bởi thư viện socket io: socket.on(“event”,”message”).

Đối với thiết bị chỉ cần lắng nghe ở sự kiện get

Để gửi một sự kiện đến server đến server sử dụng hàm socket.emit(“event”,”message”) đối với thiết bị cần gửi 2 sự kiện là get và push.

# CHƯƠNG 3: XÂY DỰNG SERVER VÀ CƠ SỞ DỮ LIỆU

1. Xây dựng server

Server sử dụng node js để phát triển đối sử dụng các thư viện socket io để sử dụng giao thức socket io, express ejs và http để xây dựng web, mysql để quản lý cơ sở dữ liệu. Đối với server local ,sử dụng port 3000, đối với server ngoài mạng local sử dụng port 80 theo chuẩn http bình thường.

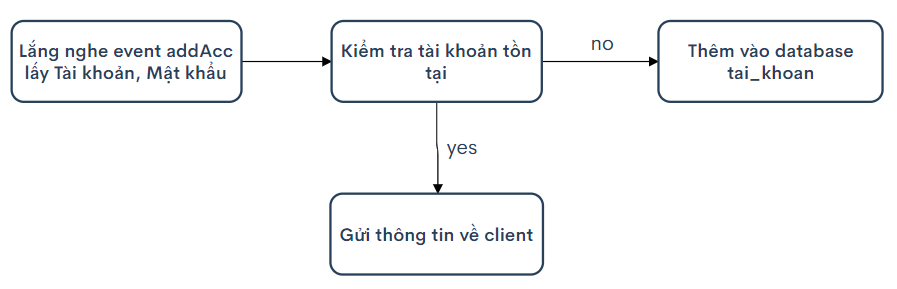


Xác định các chức năng chính: Server là nơi trung gian để người dùng tương tác với ngôi nhà của mình ngoài ra cũng có thể tùy biến ngôi nhà của mình như thêm sửa xóa các phòng, thêm sửa xóa các thiết bị.

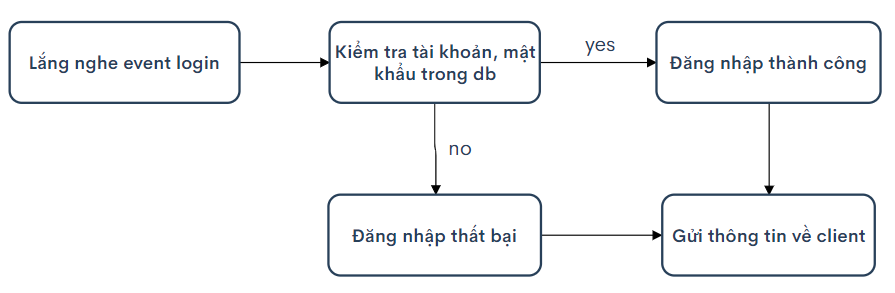
Các chức năng quản lí: thêm tài khoản, xóa tài khoản, login, logout, thêm phòng, xóa phòng, thêm thiết bị, xóa thiết bị, vào nhà, vào phòng, get thông tin thiết bị, điều khiển thiết bị.

Chức năng thêm tài khoản:

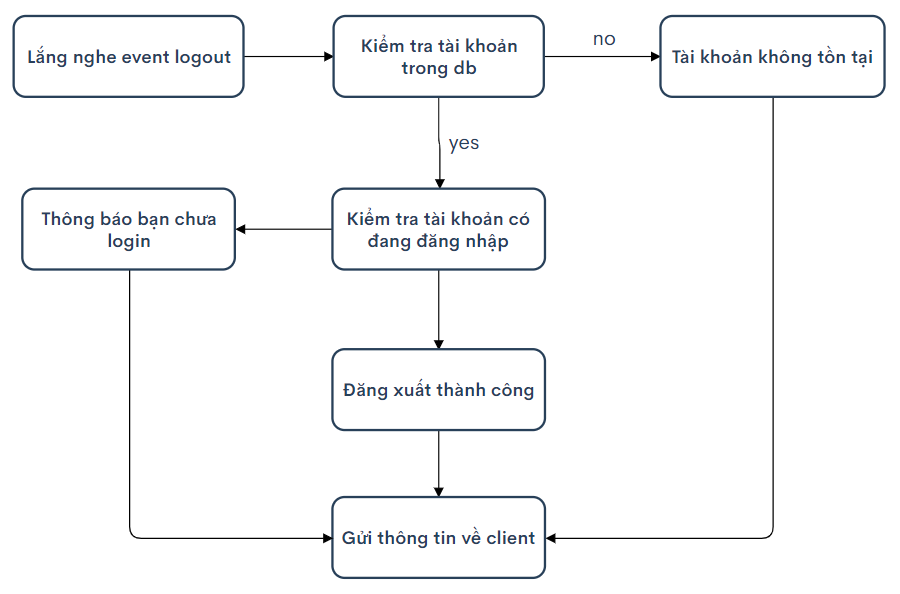
Người dùng có thể thêm tài khoản

**

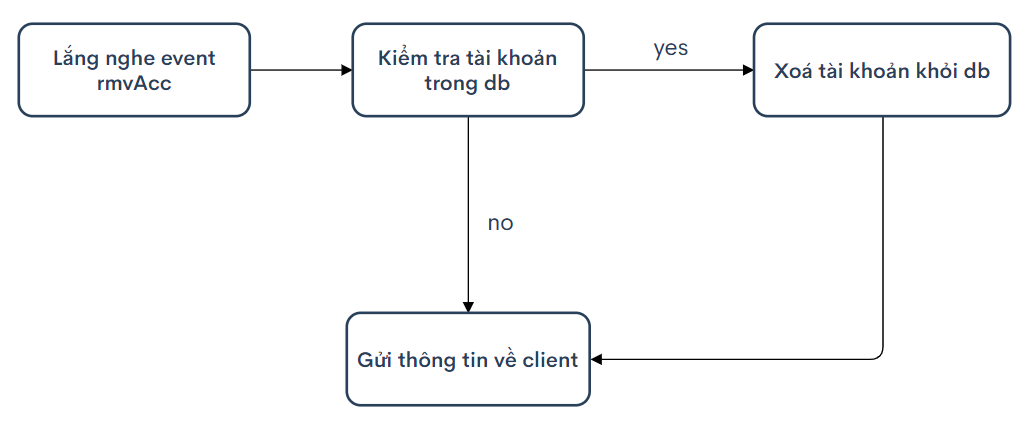
Chức năng login:



Chức năng logout:



Chức năng xóa tài khoản:

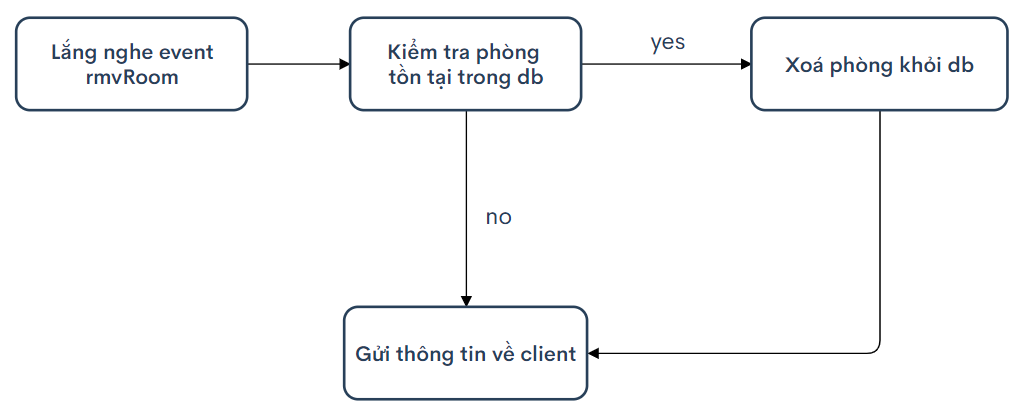


Chức năng thêm phòng:

Đối với chức năng thêm phòng chỉ cần trong trạng thái login là có thể thêm được phòng.

Chức năng xóa phòng:

Chức năng xóa phòng chỉ hoạt động khi người dùng đang dăng nhập



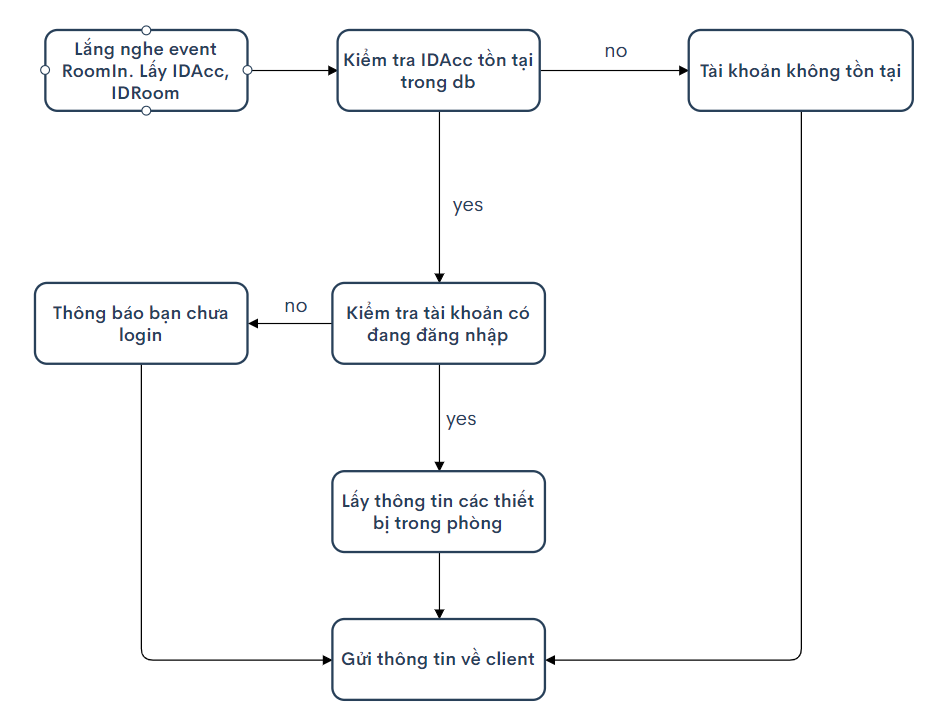
Chức năng thêm thiết bị:

Khi thiết bị ở đã login và ở trong phòng thì có thể thêm thiết bị qua sự kiện: addDecv.

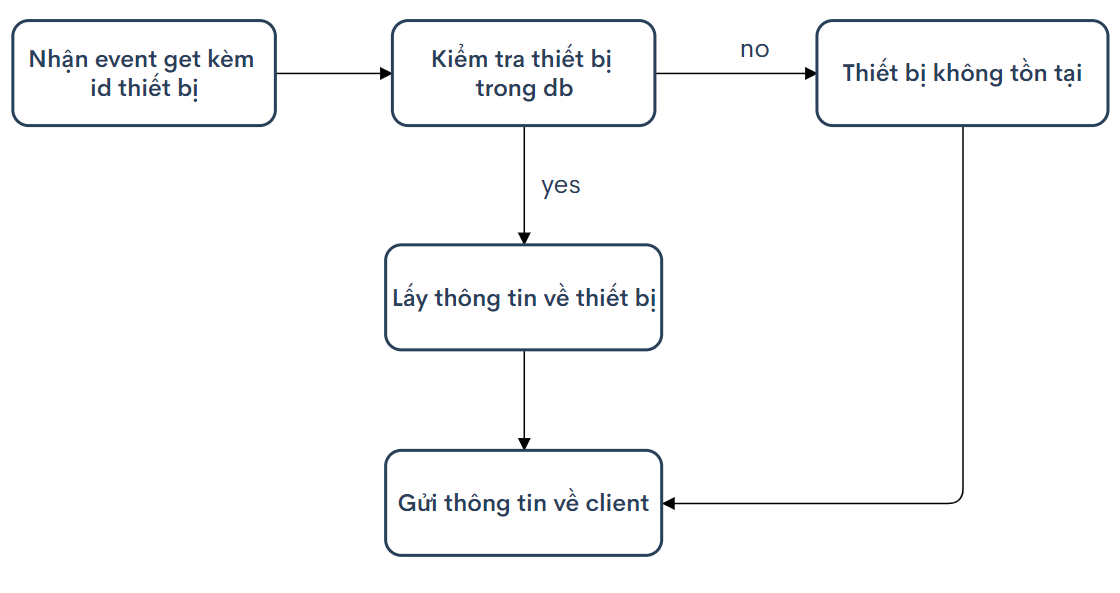
Chức năng xóa thiết bị: Tương tự xóa phòng

Chức năng vào nhà: sau khi login khi client gửi sự kiện hoomIn với message là định danh tìa khoản, server sẽ tiến hành lấy tất cả các phòng có IDAcc bằng định danh tài khoản có trong cơ sở dữ liệu phong gửi về client

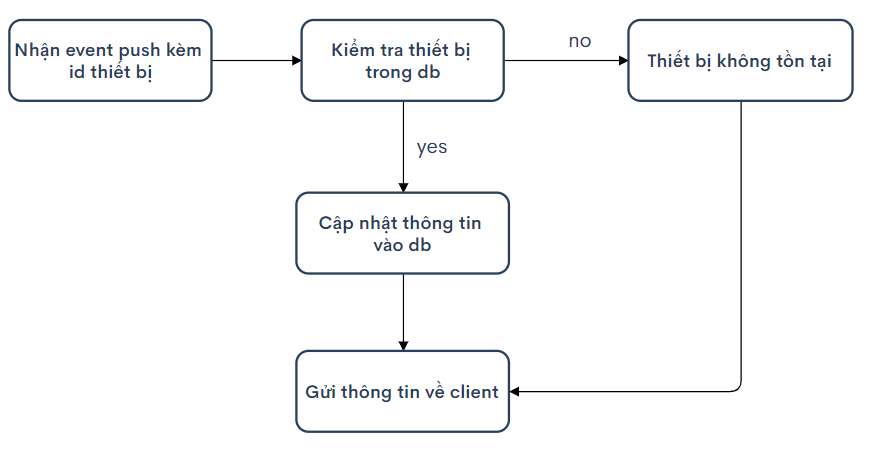
Chức năng vào phòng: sau khi login và vào nhà khi client gửi sự kiện roomIn với message là định danh phòng, server sẽ lấy tất cả các thiết bị có trong cơ sở dữ liệu thiet\_bi thỏa mãn IDRoom bằng với định danh phòng và gửi lại cho client.



Chức năng get:



Chức năng push:



Gửi thông tin về client

1. **Xây dựng cơ sở dữ liệu**

Server quản lí thông tin tài khoản, thông tin nhà, thông tin phòng và thông tin thiết bị.

Tài khoản gồm ID, tên tài khoản, mật khẩu.

Nhà gồm ID nhà, ID tài khoản sở hữu nhà, địa chỉ nhà

Phòng gồm, ID tài khoản và nhà sở hữu phòng đó, tên phòng, loại của phòng, phòng gồm các loại sau: bed room, living room, kitchen, kid room, dining room, library, yard, toilet, media room, garden, bath room, ware house, garage, gaming, other. Được đánh thứ tư tương ứng theo thứ tự từ 1.

Thiết bị gồm ID thiết bị, ID tài khoản và nhà và phòng sở hữu thiết bị, tên thiết bị, yêu thích, giá trị thiết bị, trjang thái hoạt động và loại thiết bị, thiết bị bao gồm những loại sau: lamp, light bulb, fluorescent lamp, air conditioner, lock, temperature sensor, humidity sensor, fan, camera, assistant, door, window, other*.* Được đánh số thứ tự tương ứng từ 1.

Vì vậy cần triển khai bốn bảng sau:

Bảng “*tai\_khoan” :* *ID tài khoản, tên tài khoản, mật khẩu*

|  |  |  |
| --- | --- | --- |
| ID | Name | Password |
|  |  |  |
|  |  |  |
|  |  |  |

Bảng “*nha” : ID nhà, ID tài khoản sở hữu nhà, địa chỉ nhà*

|  |  |  |
| --- | --- | --- |
| ID | IDAcc | Adress |
|  |  |  |
|  |  |  |
|  |  |  |

Bảng “*phong”*: ID phòng, ID tài khoản và nhà sở hữu phòng đó, tên phòng, loại của phòng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | IDHome | IDAcc | Name | Type |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

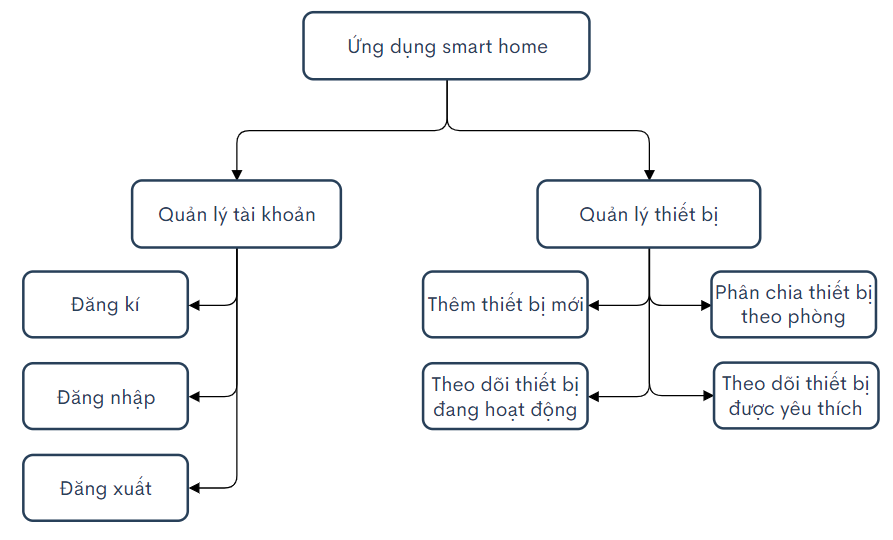
Bảng “*thiet\_bi”*: ID thiết bị , ID nhà , ID tài khoản, ID phòng chứa thiết bị tên thiết bị, loại thiết bị, yêu thích, trạng thái bật/tắt, giá trị khác(cảm biến)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | IDRoom | IDHome | IDAcc | Name | Type | Favorite | Active | Value |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

# CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG ANDROID

1. Chức năng chính của ứng dụng

Ứng dụng cung cấp các thao tác cơ bản, chia thành 2 loại thao tác chính: Quản lý tài khoản và Quản lý thiết bị.



Hình 4.0.1: Các chức năng chính trên ứng dụng Android

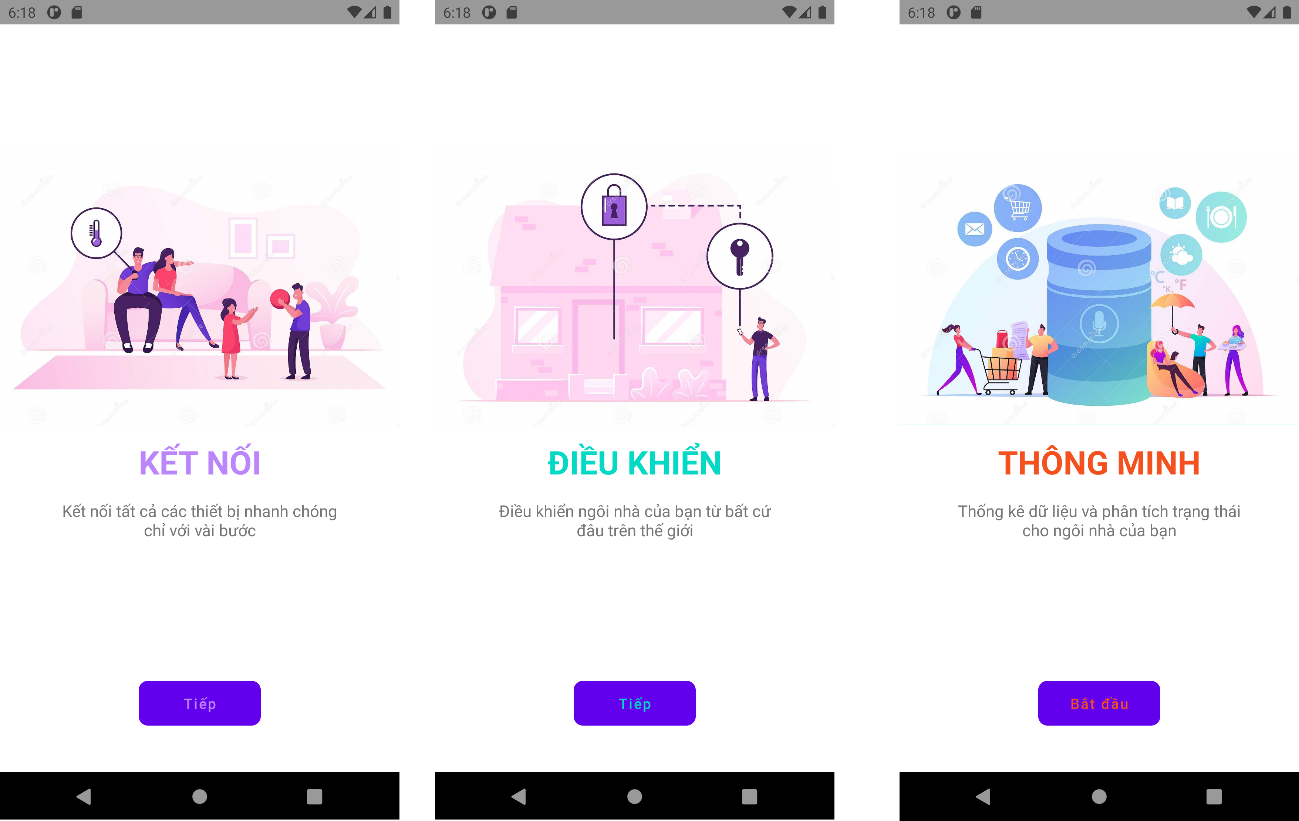
Mô tả sơ lược:

* Đăng kí/Đăng nhập/Đăng xuất: Người dùng tạo tài khoản và đăng nhập để sử dụng ứng dụng
* Thêm thiết bị mới: Kết nối với thiết bị IoT mới
* Phân chia thiết bị theo phòng: Các thiết bị được sắp xếp theo các phòng để người dùng dễ dàng quản lí
* Theo dõi các thiết bị đang hoạt động: Liệt kê nhanh các thiết bị đang hoạt động
* Theo dõi các thiết bị được yêu thích: Một số thiết bị có thể được sử dụng thường xuyên hơn các thiết bị khác. Người dùng có thể thêm các thiết bị này vào danh sách yêu thích để truy cập nhanh hơn vào các thiết bị này

1. Thiết kế Activity

### Splash Screen

Đây là màn hình giới thiệu khi lần đầu người dùng sử dụng ứng dụng, sẽ cung cấp những cái nhìn tổng quát về chức năng của ứng dụng.



Hình 4.0.2 Màn hình splash

### Activity Đăng nhập/Đăng kí

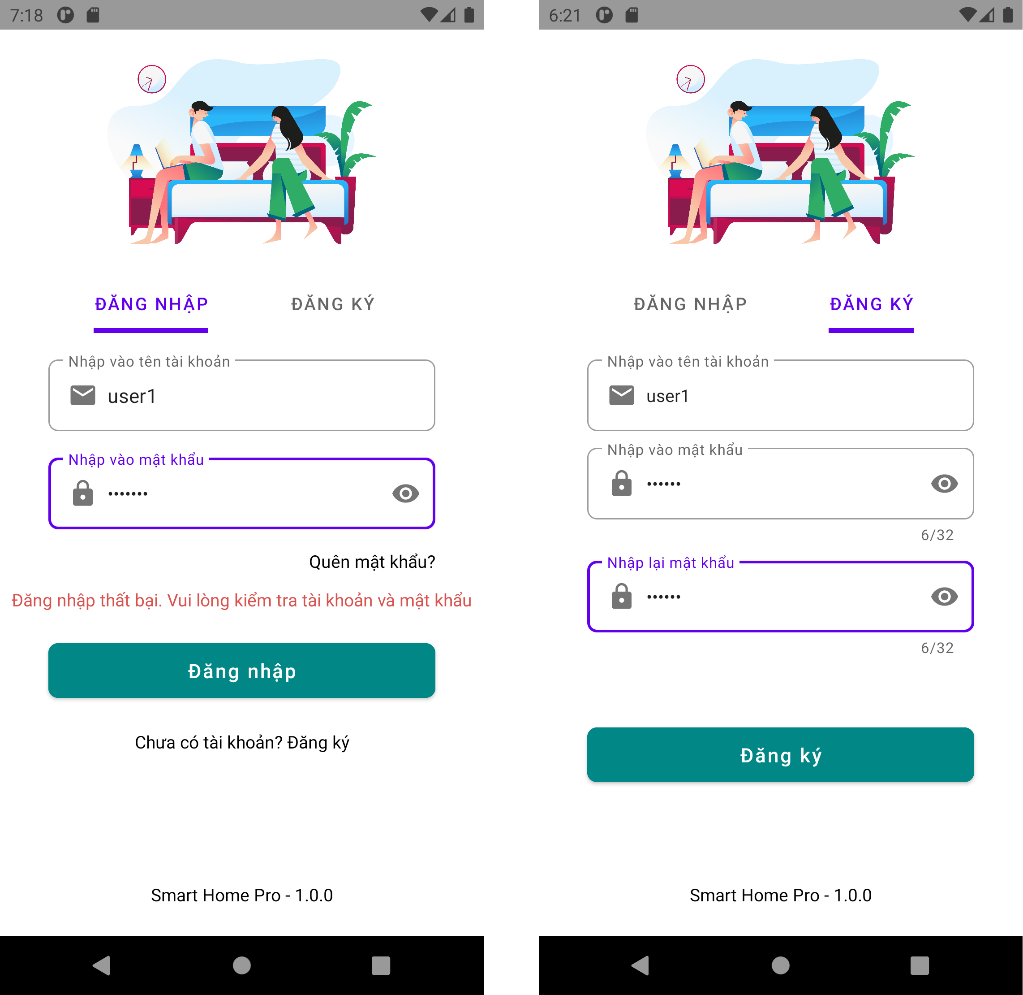
Hiển thị giao diện đăng kí, đăng nhập. Giao diện này cần có khả năng đưa ra thông báo khi người dùng nhập dữ liệu không hợp lệ hoặc khi không thể hoàn thành thao tác. Ngoài ra, sau khi người dùng đăng nhập thành công, ứng dụng sẽ lưu lại thông tin tài khoản và mật khẩu để tự động đăng nhập cho lần mở ứng dụng tiếp theo

Các trường dữ liệu chính của giao diện đăng kí người dùng cần nhập:

|  |  |  |
| --- | --- | --- |
| Trường dữ liệu | Kiểu dữ liệu | Ghi chú |
| Tên đăng nhập | Email |  |
| Mật khẩu | Password |  |
| Nhập lại mật khẩu | Password | Giống với mật khẩu đã nhập trước đó |

Các trường dữ liệu của giao diện đăng nhập người dùng cần nhập:

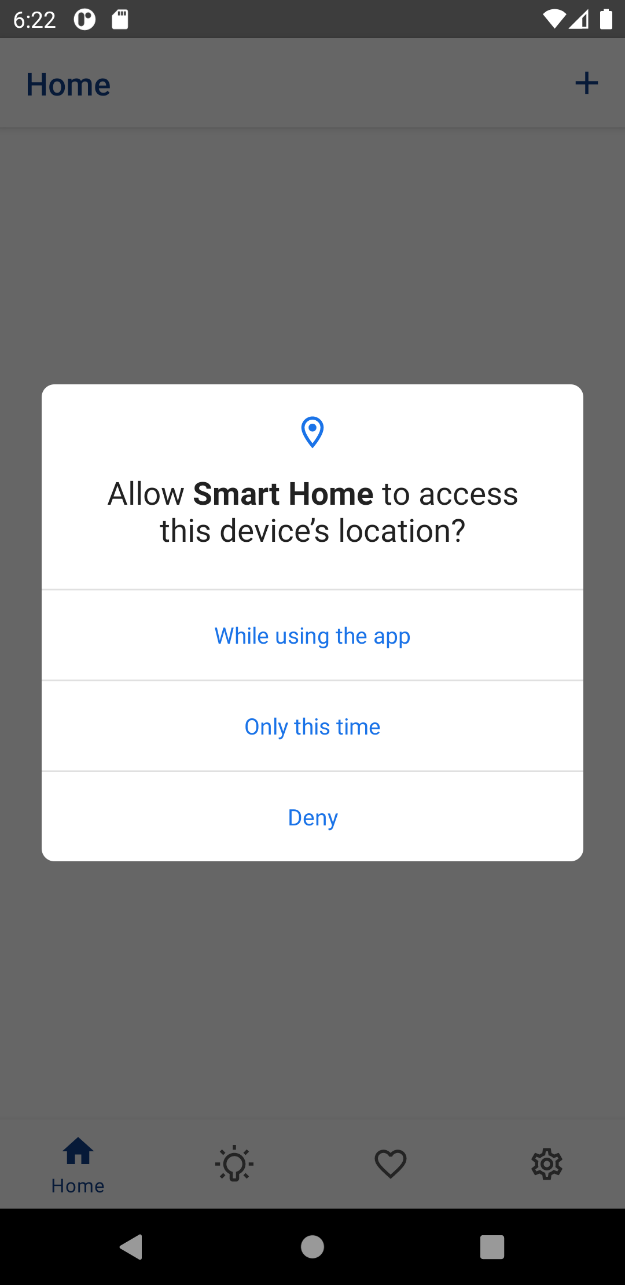
|  |  |  |
| --- | --- | --- |
| Trường dữ liệu | Kiểu dữ liệu | Ghi chú |
| Tên đăng nhập | Email |  |
| Mật khẩu | Password |  |



Hình 4.0.3. Giao diện đăng kí và đăng nhập (với thông báo lỗi)

### Activity chính

Activity này chứa toàn bộ các giao diện quản lí phòng, thiết bị,… và được hiển thị ngay sau khi người dùng đăng nhập thành công. Các màn hình khác nhau được hiển thị bằng cách thay thế fragment trong Activity này. Activity chính gồm 3 khu vực: Menu Bar (hiển thị tên màn hình hiện tại và các phím chức năng), Fragment container (chứa fragment đang hiển thị), Bottom Navbar (chuyển đổi giao diện giữa các tab chính). Có 4 tab trên Bottom Navbar mà người dùng có thể sử dụng: Home, Thiết bị đang hoạt động, Thiết bị được yêu thích, Cài đặt.



Hình 4.0.4. Giao diện màn hình chính sau khi đăng nhập

1. Thiết kế Fragment

### 3.1. Fragment Home

Fragment chứa danh sách các phòng đã được người dùng thêm vào. Tại giao diện này, người dùng có thể thêm phòng mới từ phím chức năng trên Menu Bar

Ảnh có chứa văn bản, thiết bị điện tử

Mô tả được tạo tự động

Hình 4.0.5. Giao diện home

Các thao tác trên màn hình home:

* Ấn vào phòng: truy cập màn hình hiển thị các thiết bị có trong phòng
* Ấn vào biểu tượng dấu cộng trên Menu Bar: Thêm phòng mới

### 3.2. Fragment phòng

Hiển thị danh sách các thiết bị đã được thêm trong phòng, có thể bật tắt nhanh thiết bị qua nút switch trên thiết bị tương ứng. Người dùng có thể thêm thiết bị mới hoặc xoá phòng từ các phím chức năng trên thanh Menu Bar

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết bị điện tử

Mô tả được tạo tự động

Hình 4.0.6. Giao diện phòng

Các thao tác trên giao diện phòng:

* Ấn vào thiết bị: truy cập vào màn hình hiển thị thông tin chi tiết thiết bị
* Ấn vào phím switch trên mỗi thiết bị: bật/tắt thiết bị tương ứng
* Ấn vào biểu tượng thùng rác trên Menu Bar: xoá phòng hiện tại
* Ấn vào biểu tượng dấu cộng trên Menu Bar: thêm thiết bị mới vào phòng hiện tại

### 3.3. Fragment thêm phòng

Giao diện thêm phòng mới. Người dùng cần nhập các loại thông tin sau:

|  |  |  |
| --- | --- | --- |
| Trường dữ liệu | Kiểu dữ liệu | Ghi chú |
| Tên phòng | Text |  |
| Loại phòng | Selection | Lựa chọn trong danh sách có sẵn |

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4.0.7. Giao diện thêm phòng

Các thao tác trên giao diện:

* Ấn vào nút Thêm phòng: Nếu người dùng nhập thiếu thông tin, hiển thị thông báo lỗi. Nếu đủ thông tin, tạo phòng mới
* Ấn vào nút Huỷ: Quay trở lại màn hình trước đó

### 3.4. Fragment thiết bị

Hiển thị thông tin về thiết bị. Người dùng có thể bật/tắt, thêm hoặc bỏ yêu thích thiết bị, xoá thiết bị từ giao diện này

Ảnh có chứa bàn

Mô tả được tạo tự động

Hình 4.0.8. Giao diện thiết bị

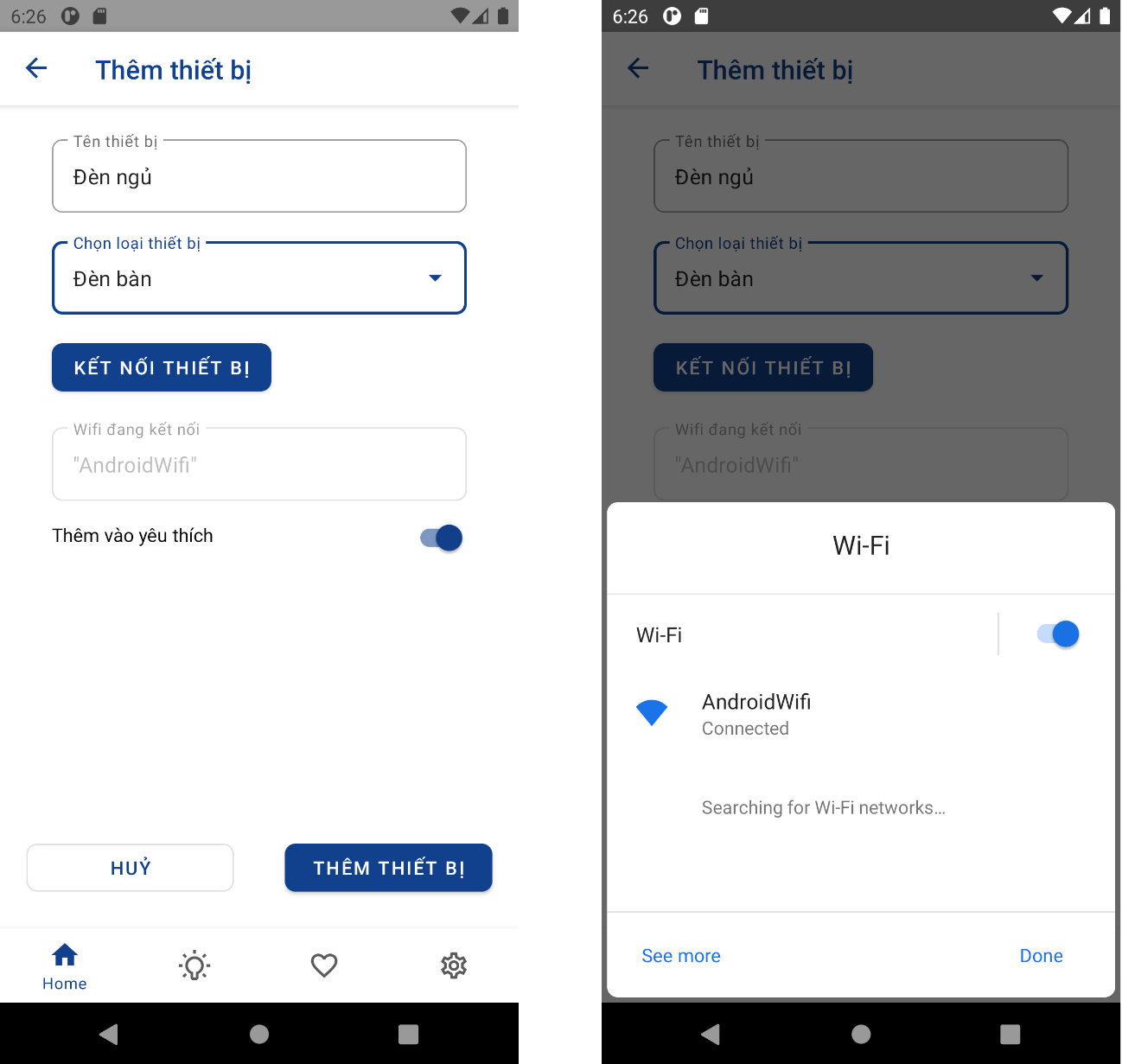
Các thao tác trên giao diện:

* Ấn vào nút Bật/Tắt: bật/tắt thiết bị
* Ấn vào nút Yêu thích/Bỏ yêu thích: Thêm/bỏ thiết bị trong mục yêu thích
* Ấn vào biểu tượng Thùng rác trên Menu Bar: Xoá thiết bị

### 3.5. Fragment thêm thiết bị

Kết nối thiết bị IoT mới và thêm vào phòng đang chọn. Người dùng cần nhập các thông tin sau đây:

|  |  |  |
| --- | --- | --- |
| Trường dữ liệu | Kiểu dữ liệu | Ghi chú |
| Tên thiết bị | Text |  |
| Loại thiết bị | Selection | Lựa chọn trong danh sách có sẵn |
| Có thêm vào yêu thích hay không | Boolean |  |
| Kết nối với thiết bị IoT | Button | Kết nối với thiết bị từ danh sách wifi |



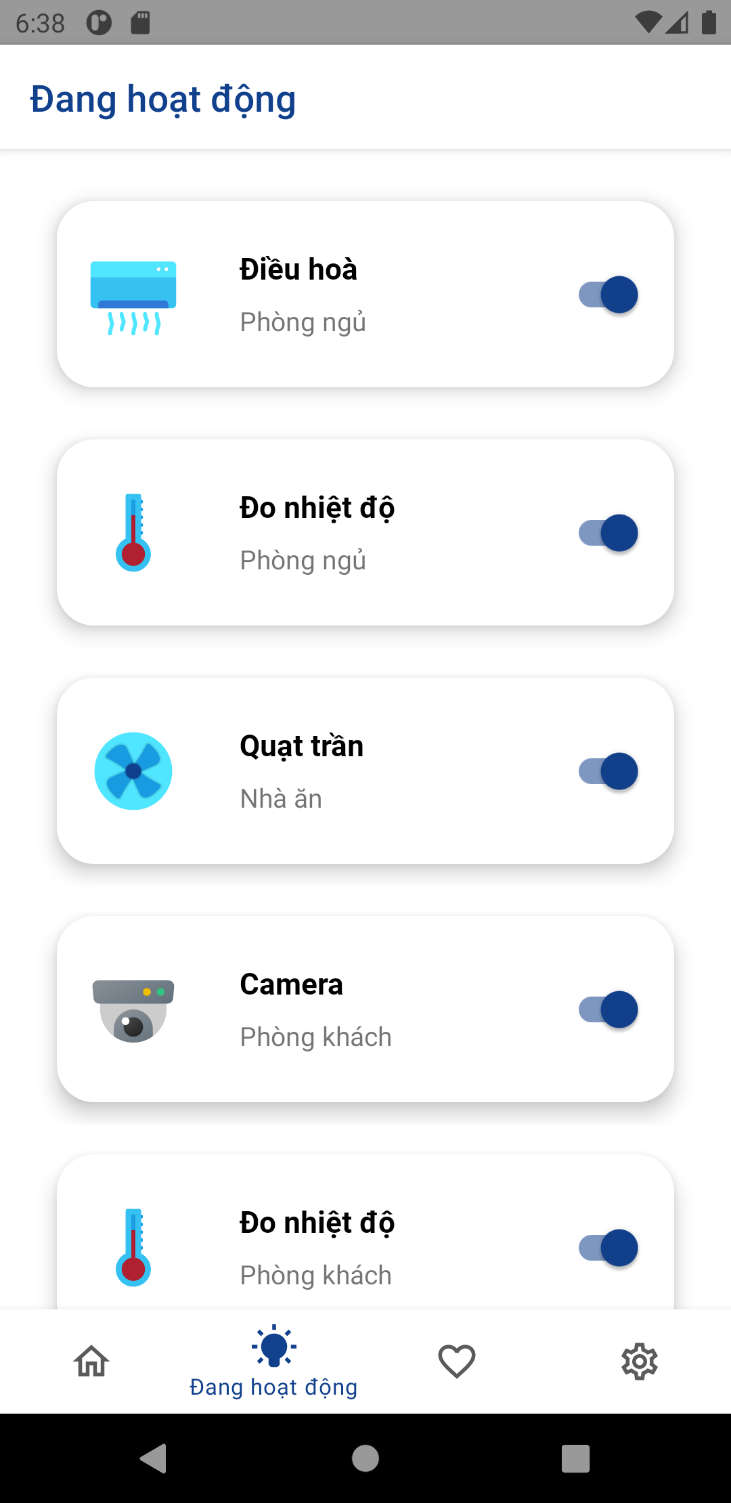
Hình 4.0.9. Giao diện thêm thiết bị mới và chọn kết nối thiết bị qua wifi

Các thao tác trên giao diện:

* Ấn nút kết nối thiết bị: hiển thị popup danh sách wifi có thể kết nối
* Ấn vào nút thêm thiết bị: Thêm thiết bị với thông tin người dùng vừa nhập
* Ấn vào nút huỷ: Quay trở lại màn hình trước đó

### 3.6. Fragment thiết bị đang hoạt động

Hiển thị danh sách các thiết bị đang hoạt động.



Hình 4.0.10. Giao diện Đang hoạt động

Các thao tác trên giao diện:

* Ấn vào thiết bị: truy cập vào màn hình thiết bị
* Ấn vào switch trên mỗi thiết bị: Tắt thiết bị tương ứng

### 3.7. Fragment yêu thích

Hiển thị các thiết bị trong danh sách yêu thích.

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết bị điện tử

Mô tả được tạo tự động

Hình 4.0.11. Giao diện Yêu thích

Các thao tác trên giao diện:

* Ấn vào thiết bị: truy cập vào màn hình thiết bị
* Ấn vào switch trên mỗi thiết bị: Bật/tắt thiết bị tương ứng

### 3.8. Fragment setting

Hiển thị các lựa chọn cài đặt ứng dụng cho người dùng



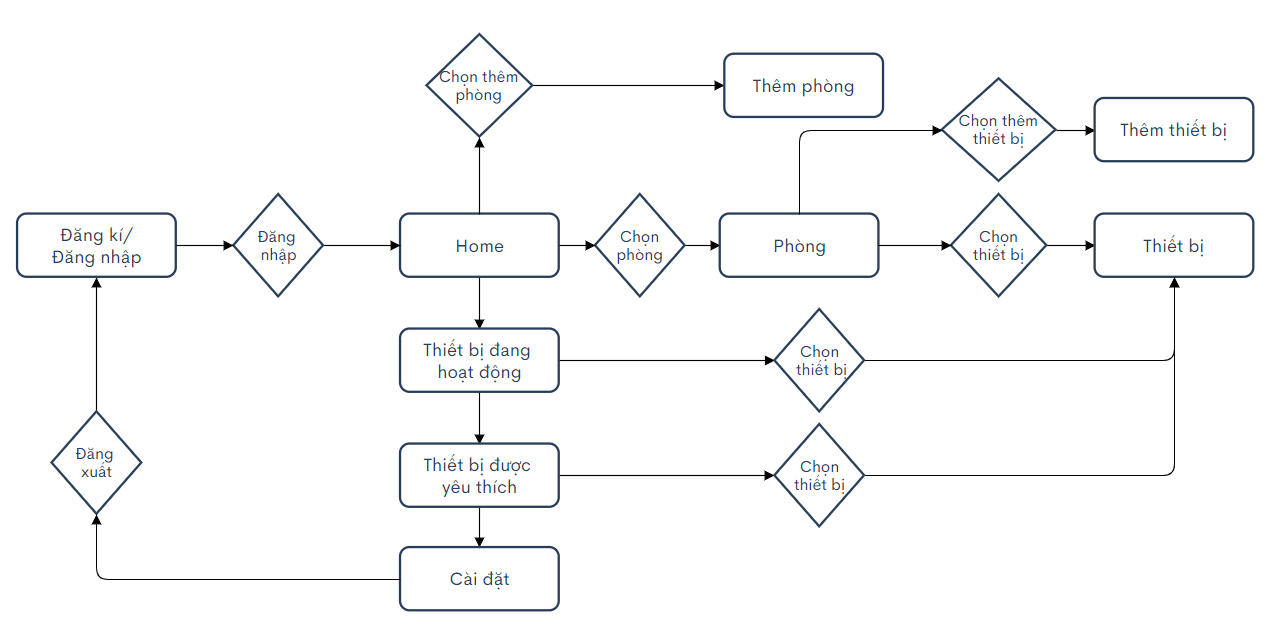
Hình 4.12. Giao diện Cài đặt

Các thao tác trên giao diện:

* Ấn vào đăng xuất: đăng xuất tài khoản

1. Navigation Graph

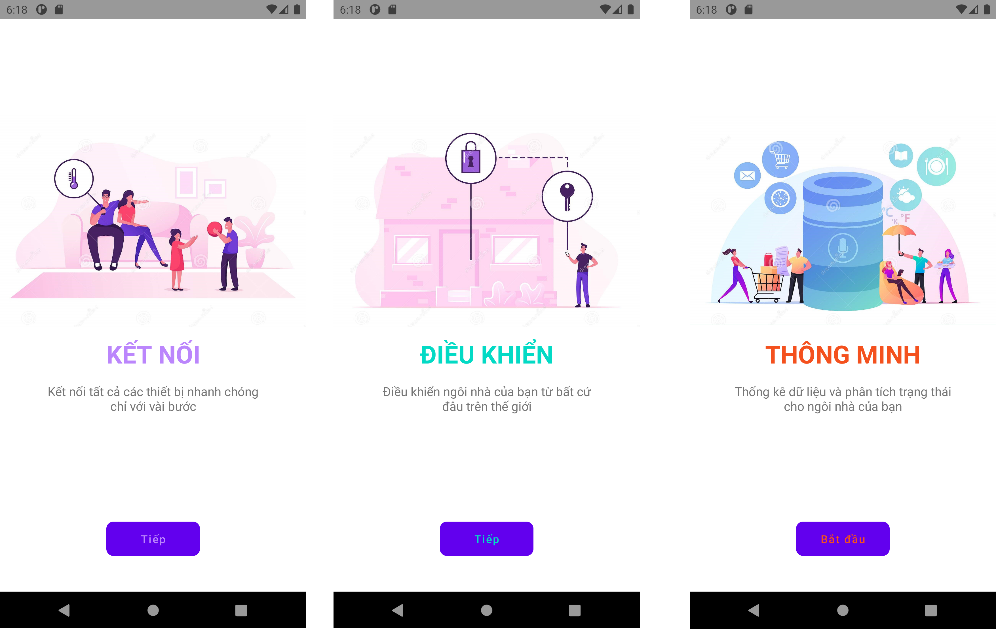
Đồ thị điều hướng giao diện

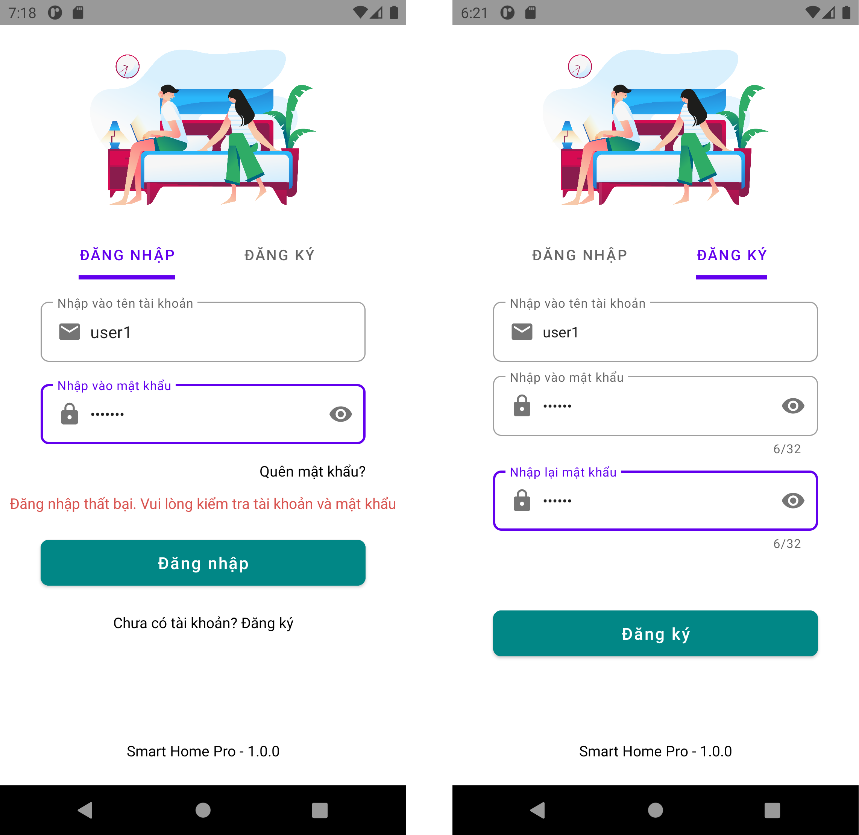


# CHƯƠNG 5: KẾT QUẢ ĐẠT ĐƯỢC VÀ ĐÁNH GIÁ

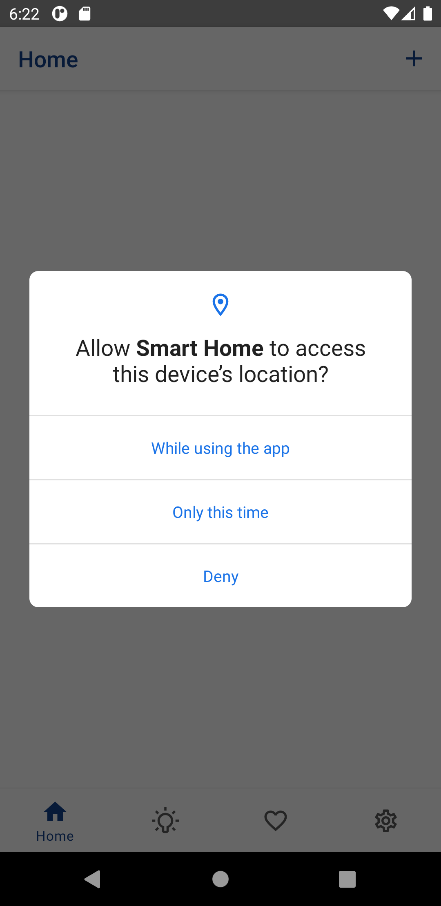
Trong dự án này về cơ bản chúng tôi đã phát triển được hệ thống điều khiển và theo dõi nhà thông minh với thời gian thực, mọi thay đổi của các thiết bị trong nhà đều được cập nhật sau 2 giây tại server và trên ứng dụng.

Dưới đây là kết quả trên app :

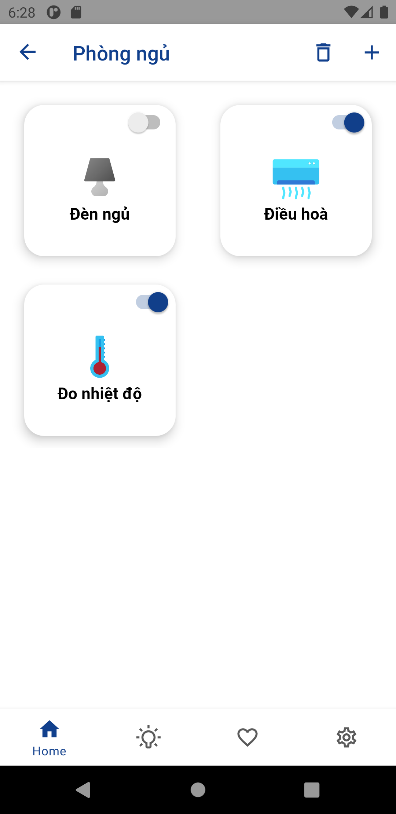


Khởi động ứng dụng

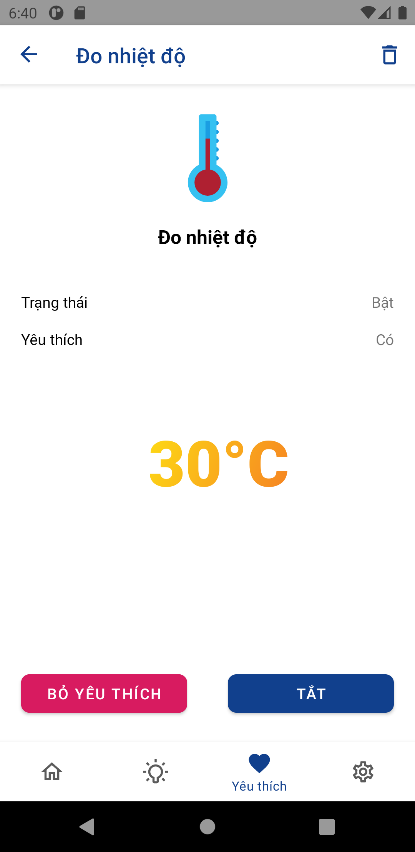
Màn hình đăng kí, đăng nhập



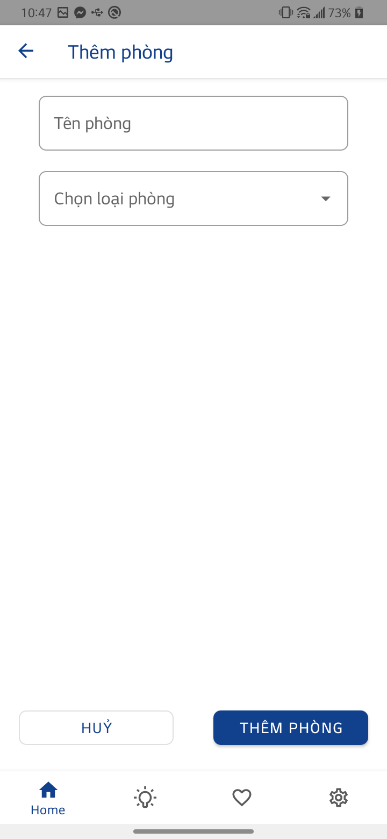
Sau khi đăng nhập thành công (vào nhà)



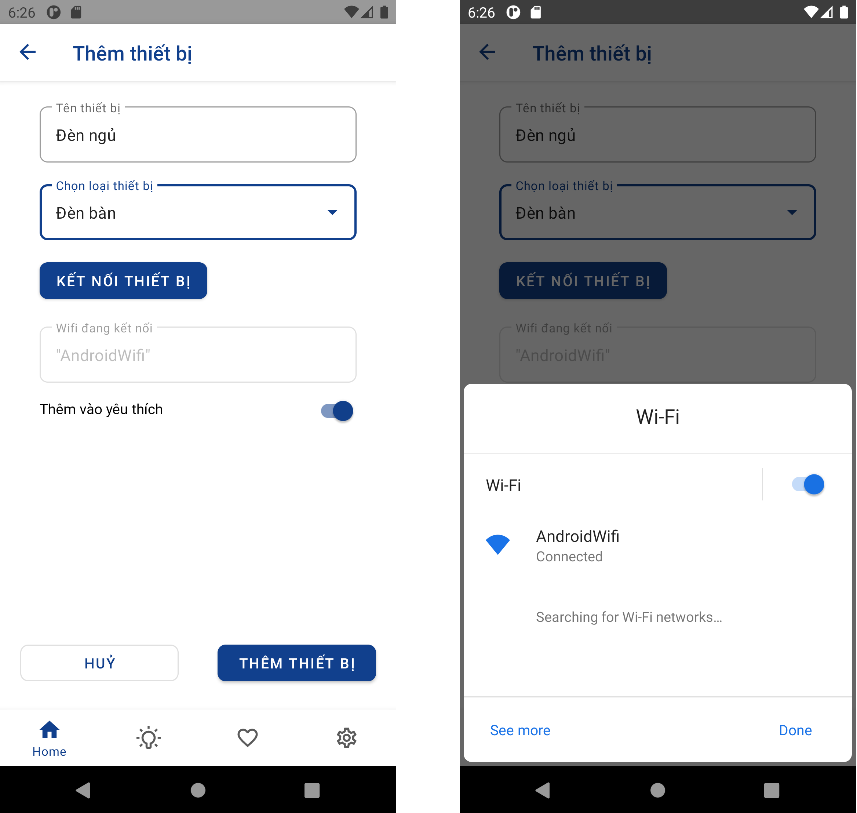
Sau khi vào phòng thành công



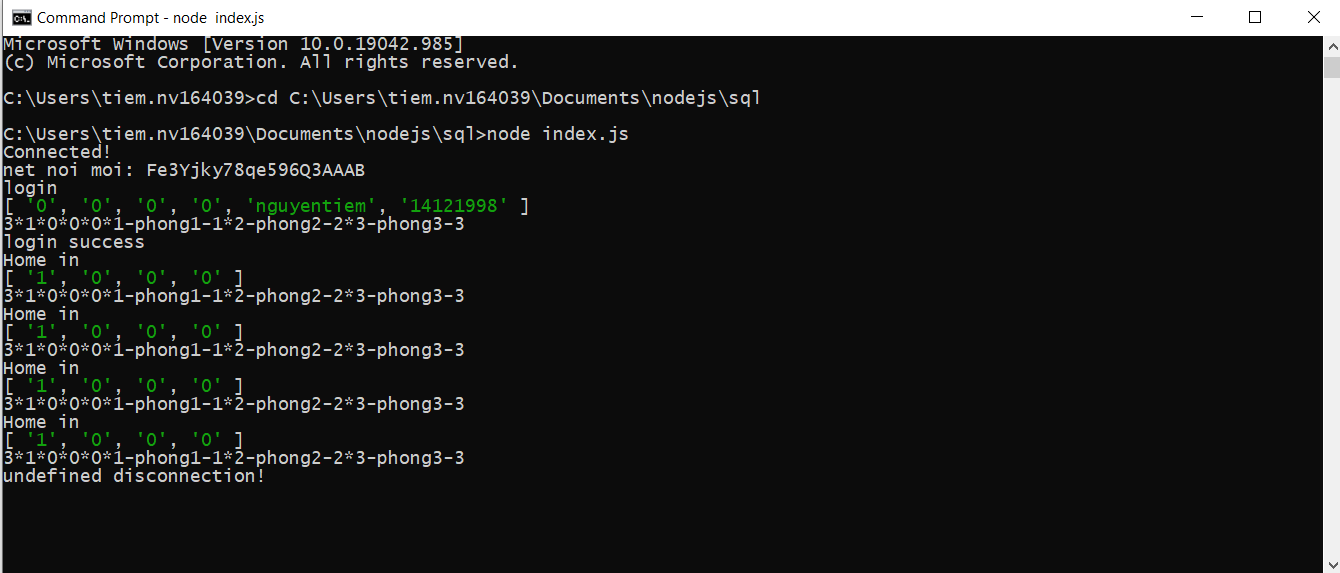
Màn hình thông tin thiết bị (cảm biến nhiệt độ)



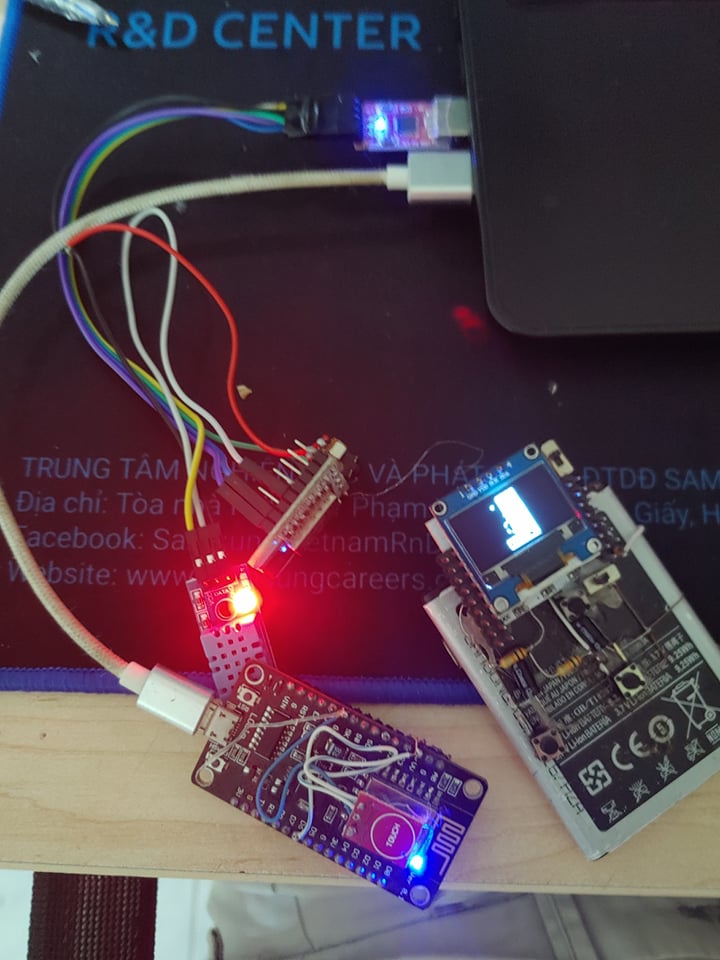
Màn hình chức năng thêm phòng



Màn hình chức năng thêm thiết bị



Server ở chức năng đăng nhập



Các thiết bị khi hoạt động