

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**

**Nhóm 12**

**Nguyễn Long Vũ – 21522800**

**Lê Huy Hiệp – 21522067**

**Nguyễn Thanh Tuấn – 21522756**

**ĐỒ ÁN MÔN HỌC**  
**HỆ THỐNG NHÚNG MẠNG KHÔNG DÂY**

**GIẢNG VIÊN HƯỚNG DẪN**

**LÊ ANH TUẤN**

TP. HỒ CHÍ MINH, 2024

## Mục Lục

DANH MỤC HÌNH VẼ.....	6
DANH MỤC TỪ VIẾT TẮT.....	8
TÓM TẮT ĐỒ ÁN.....	9
MỞ ĐẦU.....	10
Chương 1. Lời Mở Đầu.....	12
1.1. Lý do chọn đề tài.....	12
1.2. Mục tiêu nghiên cứu.....	13
1.3. Phạm vi nghiên cứu.....	14
Chương 2. Tổng Quan Về Hệ Thống Cửa Thông Minh Trong Bệnh Viện.....	15
2.1. Giới thiệu về hệ thống cửa thông minh.....	15
2.2. Ứng dụng của hệ thống cửa thông minh trong bệnh viện.....	15
2.3. Lợi ích của việc áp dụng hệ thống cửa thông minh.....	16
Chương 3. Cơ Sở Lý Thuyết.....	17
3.1. Tổng quan về Raspberry Pi 3.....	17
3.2. Tổng quan về Arduino Uno.....	20
3.3. Firebase Realtime Database.....	22
3.3.1 Các đặc điểm chính:.....	22
3.3.2 Lợi ích:.....	22
3.3.3 Ứng dụng:.....	22
3.4. Các loại cảm biến, thiết bị được sử dụng trong hệ thống cửa thông minh.....	23
3.4.1 Cảm biến sóng âm HC-SR04.....	23
3.4.2 Server.....	24

3.4.3 Còi buzzer .....	25
3.4.4 Hệ điều hành .....	26
3.5. Chuẩn giao tiếp không dây .....	27
3.5.1 Một số chuẩn không dây phổ biến hiện nay.....	27
Các Công Nghệ Tiên Tiến trong Wi-Fi .....	31
Bảo Mật trong Wi-Fi .....	31
Chương 4. Thiết Kế Hệ Thống .....	35
4.1. Kiến trúc tổng thể của hệ thống .....	35
4.2. Thiết kế.....	36
4.2.1 Raspberry Pi 3 giao tiếp Firebase .....	36
4.2.2 Raspberry Pi 3 giao tiếp Arduino Uno.....	37
4.2.3 Các cảm biến và thiết bị điều khiển .....	38
4.3. Thiết kế phần mềm .....	39
4.3.1 Ứng dụng điện thoại.....	39
4.3.2 Giao diện web .....	42
Chương 5. Triển Khai Hệ Thống .....	46
5.1. Cài đặt và cấu hình Raspberry Pi 3 .....	46
5.2. Lập trình và cấu hình Arduino Uno .....	48
5.3. Phát triển ứng dụng điện thoại .....	50
5.4. Phát triển giao diện web .....	55
5.5. Kết Quả và Đánh Giá .....	58
5.6. Tích hợp và kiểm tra hệ thống.....	58
Chương 6. Kết Quả Thử Nghiệm.....	59
6.1. Thử nghiệm tính năng điều khiển từ xa .....	59

6.2.	Thử nghiệm tính năng mở/đóng cửa tự động .....	59
6.3.	Đánh giá hiệu quả hoạt động của hệ thống .....	59
Chương 7.	Kết Luận và Đánh Giá .....	60
7.1.	Phân tích ưu điểm của hệ thống .....	60
7.2.	Đánh giá hạn chế và khó khăn .....	61
7.3.	Khả năng mở rộng và phát triển trong tương lai .....	62
Tài Liệu Tham Khảo .....		64

## DANH MỤC HÌNH VẼ

Hình 1 Raspberry pi 3 (Model B) .....	18
Hình 2 Cấu tạo của Raspberry pi 3 (Model B) .....	18
Hình 3 Cấu tạo của Arduino.....	21
Hình 4: Cảm biến sóng âm HC-SR04 .....	23
Hình 5: động cơ servor.....	24
Hình 6: còi buzzer .....	25
Hình 7: Sơ đồ hệ thống .....	35
Hình 8: Sơ đồ hoạt động của Rasp với Firebase .....	36
Hình 9: sơ đồ hoạt động của rasp với Arduino .....	37
Hình 10: sơ đồ UserCase của ứng dụng .....	39
Hình 11: Sơ đồ UserCase của web.....	42
Hình 12: giao diện intro app.....	51
Hình 13: giao diện đăng nhập .....	52
Hình 14: giao diện đăng ký .....	53
Hình 15: giao diện home (quản lý) .....	54
Hình 16: giao diện đăng nhập sai cần đăng nhập lại.....	55
Hình 17: giao diện đăng nhập web.....	56
Hình 18: giao diện chính của web.....	57

## **DANH MỤC BẢNG**

Bảng 1: Sơ đồ chân HC-SR04.....	23
Bảng 2: sơ đồ chân động cơ server .....	24
Bảng 3: sơ đồ chân của còi buzzer.....	26
Bảng 4: Thông tin các chuẩn Wifi .....	32

## DANH MỤC TỪ VIẾT TẮT



## TÓM TẮT ĐỒ ÁN

Đồ án này tập trung vào việc thiết kế và triển khai một hệ thống cửa thông minh sử dụng Raspberry Pi 3 và Arduino Uno, có khả năng kết nối với ứng dụng điện thoại và giao diện web, nhằm tăng cường an ninh và tiện ích trong bệnh viện. Hệ thống được phát triển để quản lý và kiểm soát việc ra vào tại các vị trí quan trọng trong bệnh viện, đồng thời cung cấp tính năng nhận diện khuôn mặt và giám sát từ xa.

Cấu trúc của hệ thống bao gồm Raspberry Pi 3 làm trung tâm điều khiển, kết nối với các cảm biến và mạch điều khiển thông qua Arduino Uno. Thông tin từ các cảm biến và mạch điều khiển được gửi đến Firebase Realtime Database, từ đó có thể được truy cập và kiểm soát từ ứng dụng điện thoại và giao diện web.

Hệ thống này có thể giám sát và ghi nhận thông tin về việc ra vào, bao gồm thời gian và danh tính của người ra vào, đồng thời cung cấp tính năng mở cửa tự động dựa trên nhận diện khuôn mặt. Ngoài ra, người dùng cũng có khả năng điều khiển cửa từ xa thông qua ứng dụng điện thoại và giao diện web.

Kết quả thử nghiệm cho thấy hệ thống hoạt động một cách ổn định và chính xác, cung cấp một giải pháp hiệu quả để tăng cường an ninh và quản lý ra vào trong môi trường bệnh viện. Đồ án cũng đề xuất một số phương hướng cải tiến và mở rộng để nâng cao tính linh hoạt và khả năng ứng dụng của hệ thống trong tương lai.

## MỞ ĐẦU

Ngày nay, sự phát triển mạnh mẽ của khoa học – công nghệ đã góp phần giúp chất lượng cuộc sống con người không ngừng được nâng cao. Thêm vào đó, sự kết hợp giữa khả năng tích hợp của các vi xử lý nhúng và khả năng truyền thông không dây đã mở ra nhiều cơ hội cho việc ứng dụng công nghệ trong nhiều lĩnh vực khác nhau. Chúng em nhận thấy rằng việc áp dụng công nghệ nhúng mạng không dây vào hệ thống cửa thông minh trong bệnh viện không chỉ giúp tiết kiệm thời gian và công sức của nhân viên y tế mà còn mang lại hiệu quả cao hơn trong việc bảo vệ an ninh, quản lý ra vào và tăng cường tiện nghi cho cả bệnh nhân và nhân viên.

Vì vậy, đề tài "Xây dựng hệ thống cửa thông minh sử dụng Raspberry Pi 3 cho bệnh viện" được chúng em thực hiện nhằm nghiên cứu và phát triển một hệ thống nhúng mạng không dây kết hợp với cảm biến và các công nghệ khác nhau để xây dựng một hệ thống cửa thông minh trong bệnh viện. Hệ thống này không chỉ cung cấp các tính năng như giám sát, điều khiển từ xa, mở/đóng cửa tự động mà còn cho phép nhận diện khuôn mặt, quản lý ra vào của bệnh nhân, bác sĩ và nhân viên y tế, đồng thời thu thập dữ liệu quan trọng để phân tích và tối ưu hoá quy trình quản lý bệnh viện.

Cụ thể, hệ thống cửa thông minh bao gồm ứng dụng điện thoại và web để điều khiển và kết nối tới Firebase Realtime. Raspberry Pi 3 sẽ thực hiện nhận lệnh đóng/mở cửa hoặc tự động điều khiển thông qua mạch Uno ra lệnh cho các cảm biến. Hệ thống này sẽ giúp cải thiện an ninh, giảm thiểu rủi ro do việc ra vào không kiểm soát, đồng thời tạo ra một môi trường an toàn và tiện nghi hơn cho bệnh viện.

Hy vọng rằng đề tài này sẽ đóng góp vào việc phát triển và ứng dụng công nghệ nhúng mạng không dây trong lĩnh vực y tế, đồng thời cung cấp một cái nhìn đa chiều về tiềm năng và lợi ích của việc sử dụng Raspberry Pi 3 trong xây dựng các hệ thống nhúng thông minh.

Vì trình độ nghiên cứu và thời gian thực hiện còn hạn chế song song với đó sự tiếp nhận kiến thức vẫn còn tồn tại một số hạn chế nhất định. Do đó, trong quá trình hoàn thành đề án chắc chắn không tránh khỏi những thiếu sót. Bản thân nhóm em rất mong nhận được những góp ý đến từ thầy để báo cáo được hoàn thiện hơn.

Cuối cùng, chúng em xin gửi lời cảm ơn sâu sắc nhất đến Ths. Lê Anh Tuấn đã giúp đỡ và hướng dẫn chúng em hoàn thiện đề tài này. Kính chúc thầy thật nhiều sức khỏe, hạnh phúc và thành công trên con đường giảng dạy của mình.

Chúng em xin chân thành cảm ơn!

## **Chương 1. Lời Mở Đầu**

### **1.1. Lý do chọn đề tài**

Việc lựa chọn đề tài "Xây dựng hệ thống cửa thông minh trong bệnh viện" xuất phát từ một số lý do sau:

Nhu cầu cần thiết trong bệnh viện: Bệnh viện là một môi trường đòi hỏi sự quản lý chặt chẽ và an ninh cao do số lượng người ra vào lớn và nhu cầu bảo mật thông tin bệnh nhân. Việc sử dụng hệ thống cửa thông minh có thể giúp tăng cường an ninh và quản lý ra vào trong bệnh viện.

Ứng dụng của công nghệ trong y tế: Sự phát triển của công nghệ nhúng đã mở ra nhiều cơ hội để áp dụng trong lĩnh vực y tế. Việc sử dụng Raspberry Pi 3 và Arduino Uno để xây dựng hệ thống cửa thông minh không chỉ là một dự án thú vị mà còn là cơ hội để áp dụng kiến thức kỹ thuật vào thực tiễn.

Nhu cầu thực tiễn của đội ngũ y tế: Đội ngũ y tế cần có các công cụ và hệ thống hỗ trợ để làm việc hiệu quả hơn. Một hệ thống cửa thông minh có thể giúp họ tiết kiệm thời gian và nỗ lực trong việc quản lý ra vào, từ đó tập trung hơn vào việc chăm sóc bệnh nhân.

Thách thức trong quản lý an ninh: Bệnh viện đối mặt với nhiều thách thức trong việc quản lý an ninh, bao gồm việc kiểm soát người ra vào, ngăn chặn sự xâm nhập trái phép và bảo vệ thông tin bệnh nhân. Hệ thống cửa thông minh có thể giúp giải quyết một số thách thức này một cách hiệu quả.

Khả năng mở rộng và phát triển: Mô hình hệ thống cửa thông minh có thể được mở rộng và phát triển để tích hợp các tính năng mới như nhận diện khuôn mặt, điều khiển từ xa và giao tiếp với các hệ thống thông tin khác trong bệnh viện. Điều này tạo ra tiềm năng cho việc nghiên cứu và phát triển trong tương lai.

## 1.2. Mục tiêu nghiên cứu

Mục tiêu của đề tài là xây dựng và triển khai một hệ thống cửa thông minh trong bệnh viện, với các mục tiêu cụ thể sau:

Thiết kế hệ thống cửa thông minh: Phát triển kiến trúc tổng quan của hệ thống cửa thông minh, bao gồm các thành phần phần cứng và phần mềm cần thiết.

Kết nối với Firebase Realtime Database: Thiết lập kết nối giữa hệ thống và Firebase Realtime Database để lưu trữ và quản lý dữ liệu từ các cảm biến và mạch điều khiển.

Phát triển ứng dụng điện thoại: Xây dựng một ứng dụng di động để người dùng có thể kiểm soát hệ thống từ xa, bao gồm việc mở/đóng cửa, kiểm tra trạng thái và nhận thông báo.

Tạo giao diện web: Phát triển một giao diện web để truy cập và kiểm soát hệ thống từ bất kỳ thiết bị nào có kết nối internet, cung cấp tính năng tương tự như ứng dụng di động.

Hiện thực tính năng nhận diện khuôn mặt: Tích hợp và thử nghiệm tính năng nhận diện khuôn mặt để nhận dạng và xác thực người sử dụng hệ thống.

Thử nghiệm và đánh giá hiệu quả: Tiến hành thử nghiệm và đánh giá hiệu suất của hệ thống, bao gồm tính chính xác, độ tin cậy và khả năng mở rộng.

Đề xuất phương hướng cải tiến: Dựa trên kết quả thử nghiệm, đề xuất các phương hướng cải tiến và mở rộng để nâng cao tính năng và khả năng ứng dụng của hệ thống trong tương lai.

### **1.3. Phạm vi nghiên cứu**

Phạm vi nghiên cứu của đề tài "Xây dựng hệ thống cửa thông minh trong bệnh viện" được xác định như sau:

Phạm vi Đối Tượng:

Nghiên cứu sẽ tập trung vào việc áp dụng hệ thống cửa thông minh trong một bệnh viện cụ thể.

Đối tượng nghiên cứu là các thành phần của hệ thống, bao gồm Raspberry Pi 3, Arduino Uno, các cảm biến và mạch điều khiển, cùng với ứng dụng điện thoại và giao diện web.

Người sử dụng cuối cùng của hệ thống là nhân viên và bệnh nhân trong bệnh viện.

Phạm Vi Kỹ Thuật:

Nghiên cứu sẽ tập trung vào việc phát triển và triển khai phần cứng và phần mềm của hệ thống cửa thông minh.

Phạm vi kỹ thuật bao gồm việc thiết kế, lập trình, tích hợp và kiểm tra các thành phần của hệ thống.

Đặc biệt, tính năng nhận diện khuôn mặt sẽ được hiện thực và thử nghiệm trong phạm vi của đề tài.

Phạm Vi Thời Gian:

Thời gian nghiên cứu được giới hạn trong một khoảng thời gian nhất định do hạn chế về tài nguyên và thời gian.

Việc triển khai và thử nghiệm hệ thống sẽ được thực hiện trong khoảng thời gian nhất định được quy định từ đầu.

Phạm Vi Khảo Sát:

Nghiên cứu sẽ không bao gồm việc thăm dò ý kiến hoặc khảo sát người dùng về mức độ hài lòng hoặc sự chấp nhận của họ đối với hệ thống cửa thông minh.

Tuy nhiên, đánh giá định lượng và định tính về hiệu suất và tính chất kỹ thuật của hệ thống sẽ được thực hiện.

## **Chương 2. Tổng Quan Về Hệ Thống Cửa Thông Minh Trong Bệnh Viện**

### **2.1. Giới thiệu về hệ thống cửa thông minh**

Hệ thống cửa thông minh là một ứng dụng của công nghệ nhúng và trí tuệ nhân tạo vào việc kiểm soát và quản lý cửa ra vào một cách tự động và hiệu quả. Nó kết hợp các cảm biến, vi xử lý nhúng, giao thức truyền thông và phần mềm để cung cấp các tính năng như nhận diện người dùng, điều khiển từ xa, và ghi lại lịch sử ra vào.

### **2.2. Ứng dụng của hệ thống cửa thông minh trong bệnh viện**

Trong môi trường bệnh viện, hệ thống cửa thông minh có thể được áp dụng vào nhiều mục đích khác nhau, bao gồm:

Quản Lý Ra Vào Hiệu Quả: Hệ thống cửa thông minh giúp quản lý và kiểm soát việc ra vào tại các khu vực như phòng mổ, phòng chăm sóc đặc biệt,

hoặc phòng cấp cứu. Điều này giúp đảm bảo chỉ những người có quyền truy cập được phép vào các khu vực nhạy cảm này.

**Xác Thực Danh Tính:** Thông qua tính năng nhận diện khuôn mặt, hệ thống có thể xác định và xác thực danh tính của nhân viên y tế và bệnh nhân trước khi cho phép họ ra vào. Điều này giúp ngăn chặn sự xâm nhập trái phép và bảo vệ thông tin cá nhân của bệnh nhân.

**Tăng Cường An Toàn:** Hệ thống cửa thông minh có thể tích hợp với các hệ thống an ninh khác như hệ thống camera giám sát và hệ thống báo động để tăng cường an toàn cho bệnh viện. Việc tự động ghi lại thông tin ra vào cũng giúp quản lý an ninh và giải quyết tranh chấp một cách dễ dàng hơn.

### **2.3. Lợi ích của việc áp dụng hệ thống cửa thông minh**

Việc áp dụng hệ thống cửa thông minh trong bệnh viện mang lại nhiều lợi ích quan trọng như sau:

**Tăng Cường An Ninh:** Hệ thống giúp ngăn chặn sự xâm nhập trái phép và bảo vệ an toàn cho bệnh nhân, nhân viên y tế và tài sản của bệnh viện.

**Tiết Kiệm Thời Gian:** Quản lý ra vào tự động giảm thiểu thời gian và công sức của nhân viên y tế, giúp họ tập trung vào công việc chăm sóc bệnh nhân.

**Tăng Cường Tiện Ích:** Người dùng có thể kiểm soát cửa từ xa thông qua điện thoại di động hoặc máy tính, tạo ra sự tiện ích và linh hoạt trong việc quản lý ra vào.

**Ghi Nhận Dữ Liệu:** Hệ thống ghi lại thông tin về ra vào, giúp quản lý dễ dàng kiểm tra lịch sử và phân tích hoạt động của cửa, từ đó cải thiện quản lý và ra quyết định dựa trên dữ liệu.



## **Chương 3. Cơ Sở Lý Thuyết**

### **3.1. Tổng quan về Raspberry Pi 3**

Raspberry Pi là một máy tính bo mạch đơn – single-board computer (SBC) (hay còn gọi là máy tính nhúng) được phát triển bởi Raspberry Pi Foundation. Được giới thiệu lần đầu vào năm 2012, Raspberry đã nhanh chóng trở thành biểu tượng trong cộng đồng điện tử tự chế và người học lập trình trên toàn thế giới. Thiết bị này không chỉ là một máy tính nhỏ gọn, mà còn là một nền tảng mạnh mẽ để khám phá công nghệ và thực hiện các dự án sáng tạo.

Raspberry Pi 3 B+ (RP3) cũng là một mô hình phổ biến được phát triển bởi Raspberry Pi Foundation. Được giới thiệu vào năm 2018, Raspberry Pi 3 B+ đã mang lại nhiều tính năng mới và cải tiến so với các phiên bản trước đó, làm cho nó trở thành một lựa chọn hấp dẫn cho các dự án điều khiển thông minh và các ứng dụng khác.

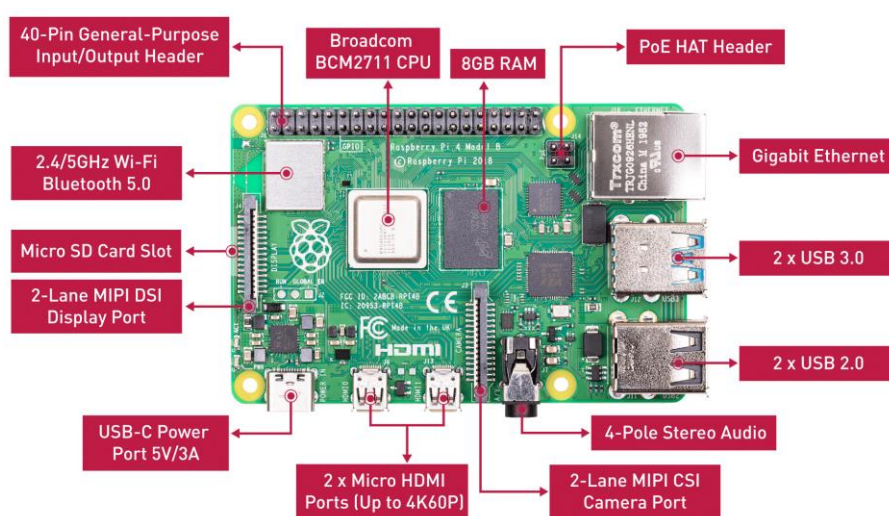
Raspberry Pi 3 B+ được trang bị SoC Broadcom BCM2837B0, bao gồm một bộ vi xử lý ARM Cortex-A53 64-bit 4 lõi tốc độ 1.4 GHz và một GPU VideoCore IV. Mặc dù không có nhiều tùy chọn bộ nhớ RAM như Raspberry Pi 4, Raspberry Pi 3 B+ thường được cung cấp với 1 GB RAM, đủ để chạy nhiều ứng dụng phổ biến.

Các tính năng kết nối của Raspberry Pi 3 B+ bao gồm cổng HDMI cho độ phân giải Full HD, cổng Ethernet 10/100, cổng USB 2.0, cũng như kết nối không dây Wi-Fi và Bluetooth tích hợp.

Dù không có các tính năng mạnh mẽ như Raspberry Pi 4, Raspberry Pi 3 B+ vẫn là một nền tảng mạnh mẽ và linh hoạt cho nhiều dự án sáng tạo và ứng dụng điều khiển thông minh.



*Hình 1 Raspberry pi 3 (Model B)*



*Hình 2 Cấu tạo của Raspberry pi 3 (Model B)*

Raspberry Pi được hỗ trợ bởi nhiều hệ điều hành khác nhau dựa trên nhân Linux.

Một số hệ điều hành phổ biến được sử dụng chuyên cho Raspberry Pi 3 là:

Raspbian Buster (hệ điều hành “chính hãng” của Raspberry Pi 3), LibreELEC, ArchLinux ARM, Kali Linux, Ubuntu.

Với hiệu suất mạnh mẽ, khả năng xử lý đa nhiệm tốt hơn và khả năng đồ họa nâng cao, Raspberry Pi 3 trở thành một lựa chọn hấp dẫn cho các dự án yêu cầu khả năng tính toán cao, xử lý video và đồ họa, cũng như các ứng dụng IoT phức tạp. Đồng thời, việc hỗ trợ các cổng kết nối nhanh và mạnh mẽ cũng mang lại khả năng mở rộng và tích hợp với các thiết bị và mạng ngoại vi một cách linh hoạt.

**Cổng mạng Gigabit Ethernet:** Cổng Mạng Gigabit Ethernet trên Raspberry Pi 3 không chỉ cung cấp khả năng kết nối với Internet mà còn cho phép thiết lập một mạng LAN (Local Area Network) với tốc độ truyền dữ liệu đáng kể nhanh hơn so với các phiên bản Raspberry Pi trước đó. Với tốc độ Gigabit, cổng Ethernet trên Raspberry Pi 3 cung cấp khả năng truyền và nhận dữ liệu với tốc độ lên tới 1 Gigabit mỗi giây (1000 megabit mỗi giây). Điều này cho phép truyền dữ liệu lớn nhanh chóng và hiệu quả.

**Cổng USB:**

2 cổng USB 3.0

2 cổng USB 2.0

**Chuẩn 40 chân GPIO (General-purpose input/output Header):** là một tập hợp các chân được thiết kế để kết nối với nhiều loại thiết bị ngoại vi khác nhau. Chúng cung cấp khả năng tương tác với nhiều loại linh kiện điện tử, cảm biến, mô-đun, màn hình và nhiều thiết bị khác thông qua giao diện GPIO, cho phép Raspberry Pi thực hiện các nhiệm vụ đa dạng và linh hoạt trong các dự án và ứng dụng khác nhau.

**Cổng ra màn hình:**

Hỗ trợ 2 cổng ra màn hình chuẩn Micro HDMI với độ phân giải lên tới 4K, cho phép kết nối với màn hình hoặc TV có hỗ trợ HDMI.

**Cổng MIPI CSI:** cổng MIPI CSI (MIPI Camera Serial Interface) trên Raspberry Pi 3 Model B được sử dụng để kết nối camera với bo mạch Raspberry Pi. Cổng này có thể chuyển dữ liệu hình ảnh từ camera trực tiếp vào bo mạch, cho phép người dùng thực hiện các ứng dụng liên quan đến việc chụp ảnh, quay video, xử lý hình ảnh và nhận dạng hình ảnh.

**Cổng MIPI DSI:** Cổng MIPI DSI (Display Serial Interface) trên Raspberry Pi có chức năng chính là kết nối với màn hình hoặc hiển thị điện tử tuân theo chuẩn giao tiếp MIPI DSI. Đây là một chuẩn giao tiếp dành riêng cho việc truyền dữ liệu từ

Raspberry Pi đến màn hình, điều khiển hiển thị và hiển thị dữ liệu như hình ảnh, video hoặc thông tin khác lên màn hình.

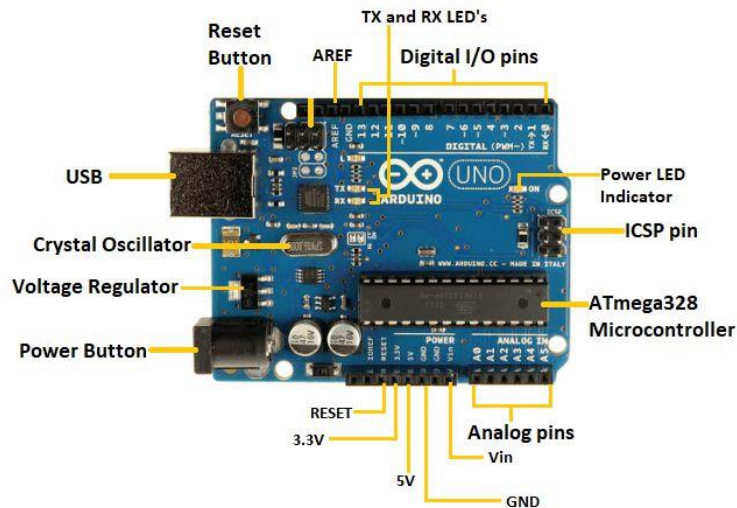
Cổng AV 4 chân: Cổng này thường được gọi là cổng AV TRRS (Tip, Ring, Ring, Sleeve) có 4 chân và thường được sử dụng để kết nối với màn hình hoặc TV thông qua cáp AV composite. Cổng có thể sử dụng để truyền tín hiệu video composite và âm thanh analog.

PoE HAT Header: cung cấp khả năng kết nối với các module mở rộng (HAT) hoặc board mở rộng có hỗ trợ POE (Power over Ethernet). POE là một công nghệ cho phép truyền dữ liệu và cung cấp nguồn điện qua cùng một dây cáp Ethernet (CAT5 hoặc CAT6) được sử dụng để kết nối thiết bị với mạng Ethernet.

### **3.2. Tổng quan về Arduino Uno**

Arduino Uno là một bo mạch vi điều khiển phổ biến được sử dụng rộng rãi trong các dự án điện tử và lập trình nhúng. Đây là một trong những sản phẩm của dòng Arduino, nổi tiếng nhờ tính dễ sử dụng và khả năng mở rộng. Arduino Uno sử dụng vi điều khiển ATmega328P của hãng Atmel và được thiết kế để dễ dàng lập trình và kết nối với các cảm biến, thiết bị điều khiển và các module khác.

Ở đồ án này, vì Raspberry Pi 3 không có khả năng đọc tín hiệu analog trực tiếp từ các cảm biến mà không cần các phụ kiện ngoại vi bổ sung. Nên phải cần dùng thêm Arduino Uno để đọc dữ liệu từ các cảm biến (bao gồm cả cảm biến analog) và sau đó chuyển dữ liệu này đến Raspberry Pi 3 để xử lý.



*Hình 3 Cấu tạo của Arduino*

Thông số kỹ thuật:

- Vi điều khiển: ATmega328 họ 8bit
- Điện áp hoạt động: 5V DC (chỉ được cấp qua cổng USB)
- Tần số hoạt động: 16 MHz
- Dòng tiêu thụ: khoảng 30mA
- Điện áp vào khuyến dùng: 7-12V DC
- Điện áp vào giới hạn: 6-20V DC
- Số chân Digital I/O: 14 (6 chân hardware PWM)
- Số chân Analog: 6 (độ phân giải 10bit)
- Dòng tối đa trên mỗi chân I/O: 30 mA
- Dòng ra tối đa (5V): 500 mA
- Dòng ra tối đa (3.3V): 50 mA
- Bộ nhớ flash: 32 KB (ATmega328) với 0.5KB dùng bởi bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)

### 3.3. Firebase Realtime Database

Firebase Realtime Database là một dịch vụ cơ sở dữ liệu đám mây thuộc Google, được thiết kế để lưu trữ và đồng bộ hóa dữ liệu theo thời gian thực. Firebase Realtime Database cho phép các ứng dụng lưu trữ và đồng bộ hóa dữ liệu giữa các người dùng theo thời gian thực, bất kể họ đang ở đâu trên thế giới.

#### 3.3.1 Các đặc điểm chính:

**Cơ sở dữ liệu thời gian thực (Realtime Database):** Cho phép dữ liệu được đồng bộ hóa ngay lập tức giữa tất cả các khách hàng kết nối. Mọi thay đổi dữ liệu trên một thiết bị sẽ được cập nhật ngay lập tức đến các thiết bị khác đang kết nối với cơ sở dữ liệu.

**Kiến trúc NoSQL:** Firebase sử dụng mô hình dữ liệu NoSQL, với cấu trúc dữ liệu dạng JSON, cho phép lưu trữ các đối tượng phức tạp và linh hoạt.

**Khả năng mở rộng (Scalability):** Firebase Realtime Database được thiết kế để dễ dàng mở rộng, phù hợp với các ứng dụng từ nhỏ đến lớn.

**Bảo mật (Security):** Firebase cung cấp các quy tắc bảo mật tùy chỉnh (Firebase Security Rules) để kiểm soát quyền truy cập và chỉnh sửa dữ liệu.

**Dễ dàng tích hợp:** Firebase cung cấp SDK cho nhiều nền tảng khác nhau như iOS, Android, và Web, giúp việc tích hợp trở nên đơn giản và nhanh chóng.

#### 3.3.2 Lợi ích:

- **Đồng bộ hóa thời gian thực:** Giúp các ứng dụng có thể phản hồi nhanh chóng và mượt mà với các thay đổi dữ liệu.
- **Quản lý trạng thái ngoại tuyến (Offline):** Firebase có thể lưu trữ dữ liệu cục bộ trên thiết bị, cho phép ứng dụng hoạt động ngay cả khi không có kết nối mạng và sẽ đồng bộ hóa dữ liệu khi kết nối được khôi phục.
- **Dịch vụ quản lý toàn diện:** Firebase không chỉ cung cấp cơ sở dữ liệu mà còn nhiều dịch vụ khác như Authentication, Cloud Messaging, và Analytics, giúp xây dựng ứng dụng toàn diện và mạnh mẽ.

#### 3.3.3 Ứng dụng:

Firebase Realtime Database rất phù hợp cho các ứng dụng cần cập nhật dữ liệu liên tục như:

Ứng dụng chat và nhắn tin tức thời.

Các ứng dụng theo dõi vị trí thời gian thực.

Ứng dụng quản lý dữ liệu chia sẻ giữa nhiều người dùng.

### 3.4. Các loại cảm biến, thiết bị được sử dụng trong hệ thống cửa thông minh

#### 3.4.1 Cảm biến sóng âm HC-SR04

Nguyên lý hoạt động: Cảm biến này sử dụng sóng siêu âm để đo khoảng cách từ cảm biến đến vật thể và tính toán khoảng cách dựa trên thời gian sóng siêu âm đi và trở về.



Hình 4: Cảm biến sóng âm HC-SR04

Cấu trúc chân:

Số chân	Tên chân	Mô tả
1	Vcc	Nguồn điện, thường là 5V
2	Ground	Kết nối với đất
3	Trigger	Gửi tín hiệu để kích hoạt sóng siêu âm
4	Echo	Nhận tín hiệu trả về từ vật thể

Bảng 1: Sơ đồ chân HC-SR04

Thông số kỹ thuật:

Khoảng cách hoạt động: Từ 2cm đến 400cm.

Điện áp hoạt động: 5V.

Độ chính xác: Cao và độ phân giải tốt.

Tiêu thụ dòng điện: Thấp.

Ứng dụng: HC-SR04 được sử dụng rộng rãi trong các ứng dụng như robot tự lái, hệ thống đo khoảng cách tự động, hệ thống an ninh, và các ứng dụng đo lường và phát hiện vật thể khác.

### 3.4.2 Servor

Nguyên lý hoạt động: Servo là một loại cảm biến được sử dụng để kiểm soát vị trí góc của một cánh quạt hoặc cơ cấu cơ khí khác thông qua một tín hiệu điều khiển. Nó hoạt động bằng cách nhận tín hiệu điều khiển PWM và di chuyển cánh quạt hoặc cơ cấu tương ứng đến vị trí mong muốn.



Hình 5: động cơ servor

Cấu trúc chân:

Số chân	Tên chân	Mô tả
1	Vcc	Nguồn điện, thường là từ 4.8V đến 6V
2	Ground	Kết nối với đất
3	Signal	Nhận tín hiệu điều khiển PWM từ vi điều khiển

Bảng 2: sơ đồ chân động cơ servor



Thông số kỹ thuật:

Phạm vi góc hoạt động: Thường từ 0 độ đến 180 độ, tùy thuộc vào loại servo.

Điện áp hoạt động: Thường từ 4.8V đến 6V, nhưng một số loại servo có thể hoạt động ở điện áp cao hơn.

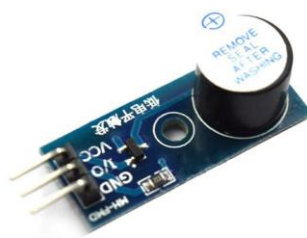
Dòng hoạt động: Phụ thuộc vào loại servo và tải đang được chịu đựng.

Tín hiệu điều khiển: Servo nhận tín hiệu PWM để định vị vị trí của cánh quạt.

Ứng dụng: Servo được sử dụng rộng rãi trong nhiều ứng dụng như robot, máy in 3D, máy bay điều khiển từ xa, và các hệ thống tự động khác cần điều khiển vị trí hoặc góc quay của cơ cấu cơ khí.

### 3.4.3 Còi buzzer

Nguyên lý hoạt động: Còi buzzer là một loại cảm biến âm thanh được sử dụng để tạo ra âm thanh cảnh báo hoặc thông báo trong các hệ thống điều khiển hoặc báo động. Nó hoạt động bằng cách tạo ra các âm thanh khi được kích hoạt bằng tín hiệu điều khiển.



Hình 6: còi buzzer

Cấu trúc chân:

Số chân	Tên chân	Mô tả
1	Vcc	Nguồn điện, thường là từ 3V đến 12V
2	Ground	Kết nối với đất
3	Signal	Nhận tín hiệu điều khiển để kích hoạt còi buzzer

*Bảng 3: sơ đồ chân của còi buzzer*

Thông số kỹ thuật:

Điện áp hoạt động: Thường từ 3V đến 12V, tùy thuộc vào loại còi buzzer.

Dòng hoạt động: Thấp, thường chỉ một vài miliampe.

Tần số âm thanh: Phụ thuộc vào loại còi buzzer, thường từ vài kHz đến vài chục kHz.

Độ ồn: Thường được đo bằng decibel (dB), độ ồn của còi buzzer có thể khác nhau tùy thuộc vào môi trường và điều kiện hoạt động.

Ứng dụng: Còi buzzer được sử dụng rộng rãi trong các hệ thống báo động, hệ thống điều khiển tự động, và các thiết bị điện tử cần cảnh báo âm thanh như hệ thống báo cháy, báo động cửa, báo động ô tô, và nhiều ứng dụng khác.

#### 3.4.4 Hệ điều hành

Hệ điều hành được sử dụng là Raspberry Pi OS 64 Bit. Đây là một phiên bản của hệ điều hành Raspberry Pi OS dành cho các bo mạch Raspberry Pi có kiến trúc 64 bit. Nó được phát triển để tối ưu hiệu suất và sử dụng tài nguyên mạnh mẽ của các bo mạch Raspberry Pi mới nhất, từ Raspberry Pi 3 trở lên.

**Sự cải tiến và các tính năng chính:** Với kiến trúc 64 bit, hệ điều hành này giúp tận dụng toàn bộ tiềm năng của các bo mạch Raspberry Pi mới nhất, mang lại hiệu suất cải thiện đáng kể trong xử lý dữ liệu và tác vụ đòi hỏi nhiều tài nguyên.

**Tính tương thích và hỗ trợ thiết bị:** Raspberry Pi OS 64 Bit được xây dựng để tối ưu hóa khả năng tương thích với các bo mạch Raspberry Pi mới và cũ, hỗ trợ nhiều thiết bị và tính năng mới, nhằm cung cấp trải nghiệm người dùng đồng đều và mạnh mẽ.

**Hiệu suất và thử nghiệm thực tế:** Raspberry Pi OS 64 Bit mang lại hiệu suất cải thiện trung bình khoảng 48%, được đánh giá thông qua nhiều thử nghiệm, bao gồm đồ họa, trí tuệ nhân tạo, mã hóa âm thanh và stress test. Qua nhiều trải nghiệm thực tế cho thấy sự ổn định và khả năng đáp ứng tốt trong các tác vụ văn phòng hàng ngày.

### 3.5. Chuẩn giao tiếp không dây

#### 3.5.1 Một số chuẩn không dây phổ biến hiện nay

##### 3.5.1.1 Bluetooth

Bluetooth là một công nghệ truyền thông không dây ngắn hạn cho phép các thiết bị trao đổi dữ liệu trong khoảng cách gần. Công nghệ này được phát triển bởi Bluetooth Special Interest Group (SIG) và sử dụng băng tần ISM (Industrial, Scientific, and Medical) 2.4 GHz. Dưới đây là các khía cạnh lý thuyết chi tiết về Bluetooth:

#### **Kiến trúc Bluetooth:**

Bluetooth sử dụng kiến trúc điểm-điểm (point-to-point) hoặc điểm-đa điểm (point-to-multipoint) để kết nối các thiết bị. Một hệ thống Bluetooth cơ bản bao gồm:

- **Piconet:** Một mạng Bluetooth nhỏ gồm một thiết bị chính (master) và một hoặc nhiều thiết bị phụ (slave). Một piconet có thể chứa tối đa 7 thiết bị phụ.
- **Scatternet:** Một tập hợp của nhiều piconet. Một thiết bị có thể là thành viên của nhiều piconet, do đó tạo ra một mạng scatternet.

#### **Các phiên bản Bluetooth:**

Bluetooth đã trải qua nhiều phiên bản phát triển, mỗi phiên bản cải thiện tốc độ, bảo mật, và tính năng mới:

- **Bluetooth 1.0 và 1.0B:** Ra mắt vào năm 1999, cung cấp tốc độ truyền dữ liệu lên đến 721 kbps, nhưng gặp phải nhiều vấn đề về khả năng tương thích và bảo mật.
- **Bluetooth 2.0 + EDR (Enhanced Data Rate):** Ra mắt vào năm 2004, tăng tốc độ truyền dữ liệu lên đến 3 Mbps bằng cách sử dụng điều chế PSK (Phase Shift Keying).
- **Bluetooth 3.0 + HS (High Speed):** Ra mắt vào năm 2009, sử dụng giao thức 802.11 để đạt tốc độ truyền dữ liệu lên đến 24 Mbps.
- **Bluetooth 4.0:** Ra mắt vào năm 2010, giới thiệu Bluetooth Low Energy (BLE), một giao thức tiết kiệm năng lượng dành cho các ứng dụng yêu cầu pin lâu dài. Tốc độ truyền dữ liệu BLE tối đa là 1 Mbps.
- **Bluetooth 4.1:** Ra mắt vào năm 2013, cải thiện khả năng tương tác với các giao thức mạng khác và tăng hiệu suất của BLE.
- **Bluetooth 4.2:** Ra mắt vào năm 2014, tăng cường bảo mật và hỗ trợ cho các địa chỉ IPv6 thông qua 6LoWPAN.
- **Bluetooth 5.0:** Ra mắt vào năm 2016, tăng phạm vi hoạt động lên gấp 4 lần, tốc độ truyền dữ liệu lên đến 2 Mbps, và cải thiện khả năng tương tác với IoT.
- **Bluetooth 5.1:** Ra mắt vào năm 2019, giới thiệu tính năng định vị chính xác thông qua AoA (Angle of Arrival) và AoD (Angle of Departure).
- **Bluetooth 5.2:** Ra mắt vào năm 2020, hỗ trợ LE Audio, cải thiện chất lượng âm thanh và khả năng truyền tải dữ liệu âm thanh.

### Các chế độ hoạt động:

Bluetooth có hai chế độ hoạt động chính:

- **Classic Bluetooth:** Chế độ này dành cho các ứng dụng yêu cầu tốc độ truyền dữ liệu cao, chẳng hạn như truyền tải âm thanh và video. Nó hỗ trợ các cấu hình như A2DP (Advanced Audio Distribution Profile), HFP (Hands-Free Profile), và nhiều hơn nữa.
- **Bluetooth Low Energy (BLE):** BLE được tối ưu hóa cho các ứng dụng yêu cầu năng lượng thấp và tốc độ truyền dữ liệu thấp, như cảm biến, thiết bị đeo và các thiết bị IoT. BLE sử dụng một loạt các kênh nhỏ để truyền dữ liệu, tiết kiệm năng lượng hơn so với Classic Bluetooth.

### Bảo mật Bluetooth:

Bluetooth cung cấp nhiều cơ chế bảo mật để bảo vệ dữ liệu:

- **Pairing:** Quá trình ghép nối hai thiết bị Bluetooth với nhau. Có nhiều phương pháp ghép nối như Just Works, Passkey Entry, và Numeric Comparison để đảm bảo tính bảo mật.
- **Authentication:** Xác thực các thiết bị để đảm bảo rằng chúng được phép kết nối với nhau.
- **Encryption:** Mã hóa dữ liệu được truyền giữa các thiết bị để bảo vệ khỏi sự can thiệp và truy cập trái phép. Bluetooth sử dụng E0 stream cipher và AES-CCM (Advanced Encryption Standard - Counter with CBC-MAC) cho mã hóa.

### Các ứng dụng của Bluetooth:

Bluetooth được sử dụng rộng rãi trong nhiều ứng dụng khác nhau:

- **Thiết bị âm thanh:** Tai nghe không dây, loa, hệ thống âm thanh xe hơi.
- **Thiết bị đeo:** Đồng hồ thông minh, vòng tay thể dục.
- **Nhà thông minh:** Đèn thông minh, khóa cửa, cảm biến.
- **Y tế:** Máy đo nhịp tim, máy đo huyết áp.
- **Truyền dữ liệu:** Chia sẻ tập tin, đồng bộ hóa thiết bị.

Bluetooth tiếp tục phát triển và mở rộng khả năng của mình để đáp ứng nhu cầu kết nối không dây trong cuộc sống hiện đại và công nghiệp IoT.

#### 3.5.1.2 Wi-Fi (IEEE 802.11)

Wi-Fi là một tập hợp các chuẩn truyền thông không dây được thiết kế để cung cấp khả năng kết nối mạng tốc độ cao thông qua sóng vô tuyến. Các chuẩn này được định nghĩa bởi IEEE (Institute of Electrical and Electronics Engineers) trong bộ tiêu chuẩn 802.11. Dưới đây là các khía cạnh lý thuyết và cơ chế hoạt động chi tiết của Wi-Fi.

## Cấu Trúc Mạng Wi-Fi:

Wi-Fi hoạt động dựa trên hai kiến trúc mạng chính:

- **Infrastructure Mode:** Bao gồm một hoặc nhiều thiết bị không dây (stations) kết nối với một điểm truy cập (Access Point - AP) trung tâm. AP đóng vai trò là cầu nối giữa các thiết bị không dây và mạng có dây (Ethernet).
- **Ad-Hoc Mode:** Còn được gọi là mạng ngang hàng (peer-to-peer), nơi các thiết bị không dây kết nối trực tiếp với nhau mà không cần AP. Thường sử dụng trong các mạng tạm thời hoặc các thiết lập nhanh chóng.

## Basic Service Set (BSS) và Extended Service Set (ESS)

- **BSS:** Một nhóm các thiết bị không dây kết nối với một AP duy nhất.
- **ESS:** Gồm nhiều BSS liên kết với nhau thông qua một hệ thống phân phối (DS - Distribution System), thường là mạng có dây.

## Cơ Chế Hoạt Động của Wi-Fi:

### Quản Lý Tần Số và Kênh:

Wi-Fi hoạt động trên hai băng tần chính:

- **2.4 GHz:** Có 14 kênh, trong đó chỉ có 3 kênh không trùng lặp (1, 6, 11). Băng tần này dễ bị nhiễu từ các thiết bị khác như lò vi sóng và điện thoại không dây.
- **5 GHz:** Có nhiều kênh không trùng lặp hơn, ít nhiễu hơn nhưng phạm vi ngắn hơn so với 2.4 GHz. Wi-Fi 6E và Wi-Fi 7 mở rộng lên băng tần 6 GHz, cung cấp thêm nhiều kênh và tốc độ cao hơn.

### Truyền Tải Dữ Liệu:

- **PHY (Physical Layer):** Xử lý việc mã hóa và truyền dữ liệu trên sóng vô tuyến. Bao gồm các phương pháp điều chế như OFDM, DSSS, và QAM.
- **MAC (Medium Access Control) Layer:** Quản lý quyền truy cập vào kênh truyền thông, điều khiển luồng dữ liệu và xử lý các khung dữ liệu (frames).

## CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)

Wi-Fi sử dụng CSMA/CA để tránh xung đột trong việc truyền dữ liệu:

- **Carrier Sensing:** Thiết bị kiểm tra kênh trước khi truyền dữ liệu.
- **Collision Avoidance:** Thiết bị gửi tín hiệu cảnh báo trước khi truyền để tránh xung đột. Nếu kênh bận, thiết bị sẽ chờ một khoảng thời gian ngẫu nhiên trước khi thử lại.

### **ACK (Acknowledgment)**

Mỗi khung dữ liệu truyền đi phải được xác nhận bởi thiết bị nhận. Nếu không nhận được ACK, khung dữ liệu sẽ được truyền lại.

## **Các Công Nghệ Tiên Tiến trong Wi-Fi**

### **MIMO (Multiple Input Multiple Output)**

- **Single-User MIMO (SU-MIMO):** Sử dụng nhiều ăng-ten để truyền và nhận dữ liệu giữa hai thiết bị, tăng tốc độ truyền và phạm vi.
- **Multi-User MIMO (MU-MIMO):** Cho phép nhiều thiết bị truyền và nhận dữ liệu đồng thời từ AP, tăng hiệu suất mạng.

### **Beamforming**

Kỹ thuật tập trung sóng vô tuyến trực tiếp vào thiết bị nhận, thay vì phát sóng ra mọi hướng, cải thiện hiệu suất và giảm nhiễu.

### **OFDMA (Orthogonal Frequency Division Multiple Access)**

Kỹ thuật chia kênh thành nhiều sóng con nhỏ hơn để truyền tải dữ liệu từ nhiều thiết bị đồng thời, tăng hiệu quả sử dụng kênh.

## **Bảo Mật trong Wi-Fi**

### **WEP (Wired Equivalent Privacy)**

- **Ra mắt:** 1997
- **Mã hóa:** RC4
- **Nhược điểm:** Bảo mật yếu, dễ bị tấn công.

### **WPA (Wi-Fi Protected Access)**

- **Ra mắt:** 2003

- **Mã hóa:** TKIP (Temporal Key Integrity Protocol)
- **Ưu điểm:** Bảo mật tốt hơn WEP.

## WPA2

- **Ra mắt:** 2004
- **Mã hóa:** AES (Advanced Encryption Standard)
- **Ưu điểm:** Bảo mật mạnh mẽ, hiện là chuẩn phổ biến.

## WPA3

- **Ra mắt:** 2018
- **Tính năng mới:** SAE (Simultaneous Authentication of Equals), bảo vệ chống lại tấn công brute-force, tăng cường bảo mật cá nhân và doanh nghiệp.

### So sánh các chuẩn wifi hiện nay

Phiên Bản	Tên Gọi	Năm Ra Mắt	Kênh Hoạt Động	Tốc Độ Đơn Luồng (tối đa)	Trạng Thái
IEEE 802.11a	Không có	1999	5 GHz	54 Mbps	Ít được sử dụng
IEEE 802.11b	Không có	1999	2.4 GHz	11 Mbps	Ít được sử dụng
IEEE 802.11g	Wi-Fi 3	2003	2.4 GHz	54 Mbps	Ít được sử dụng
IEEE 802.11n	Wi-Fi 4	2009	2.4 GHz và 5 GHz	600 Mbps	Được sử dụng
IEEE 802.11ac	Wi-Fi 5	2014	5 GHz	1300 Mbps	Được sử dụng
IEEE 802.11ax	Wi-Fi 6	2019	2.4 GHz và 5 GHz	10.Gbps	Được sử dụng

Bảng 4: Thông tin các chuẩn Wifi



**Wibree:** Wibree là một công nghệ mạng không dây được phát triển để hoạt động trong khoảng cách ngắn. Nó cung cấp khả năng truyền thông tin giữa các thiết bị tiêu thụ ít năng lượng và có phạm vi hoạt động tương đối ngắn. Wibree thường được sử dụng trong các ứng dụng IoT (Internet of Things) và các thiết bị di động. Tuy nhiên, Wibree không phổ biến như các công nghệ khác như Bluetooth hay Zigbee, và có thể đã được thay thế bởi các tiêu chuẩn khác trong thời gian gần đây.

**Zigbee:** Zigbee là một công nghệ truyền thông ngắn khoảng cách, hỗ trợ truyền thông đến nhiều thiết bị (mạng lưới mesh) so với Bluetooth. Phạm vi hoạt động của Zigbee nâng cao từ 75m đến vài trăm mét. Các thông số kỹ thuật ZigBee 3.0 dựa trên IEEE802.15.4 bao gồm tần số 2.4 GHz, phạm vi 10-20m và tốc độ truyền dữ liệu 250 kbp/s. 19

- Zigbee Coordinator (ZC): Thiết bị gốc quản lý cấu trúc mạng, định địa chỉ và lưu trữ bảng địa chỉ. Mỗi mạng có một ZC và là thiết bị "nói chuyện" với các mạng khác.
- Zigbee Router (ZR): Định tuyến trung gian trong truyền dữ liệu, tự động phát hiện và lập bản đồ nút xung quanh, điều khiển các nút hoạt động.
- Zigbee End Device (ZED): Thiết bị điểm cuối giao tiếp với ZC và ZR gần nhất. Chúng đọc thông tin từ thành phần vật lý, thường ở trạng thái nghỉ và chỉ làm việc khi cần.

**NFC:** Khi hai thiết bị đều có kết nối NFC, bạn có thể chạm chúng vào nhau để kích hoạt tính năng này và nhanh chóng truyền tập tin gồm danh bạ, nhạc, hình ảnh, video, ứng dụng hoặc địa chỉ website... Ở các nước phát triển, NFC còn được xem là chiếc ví điện tử khi có thể thanh toán trực tuyến, tiện lợi và nhanh chóng. Thiết bị NFC chỉ có thể truyền không dây vài kilobit dữ liệu trong phạm vi vài cm, do đó nó đảm bảo an toàn khi người sử dụng muốn trao đổi thông tin nhạy cảm.

**Wireless USB:** có tính năng tương tự USB nói chung nhưng không cần dây cáp. Nó hỗ trợ máy in, máy ảnh, ổ cứng rời... kết nối không dây với máy tính. Trên thực tế, USB không dây có thể được đưa vào hệ thống dưới dạng tích hợp sẵn hoặc qua các

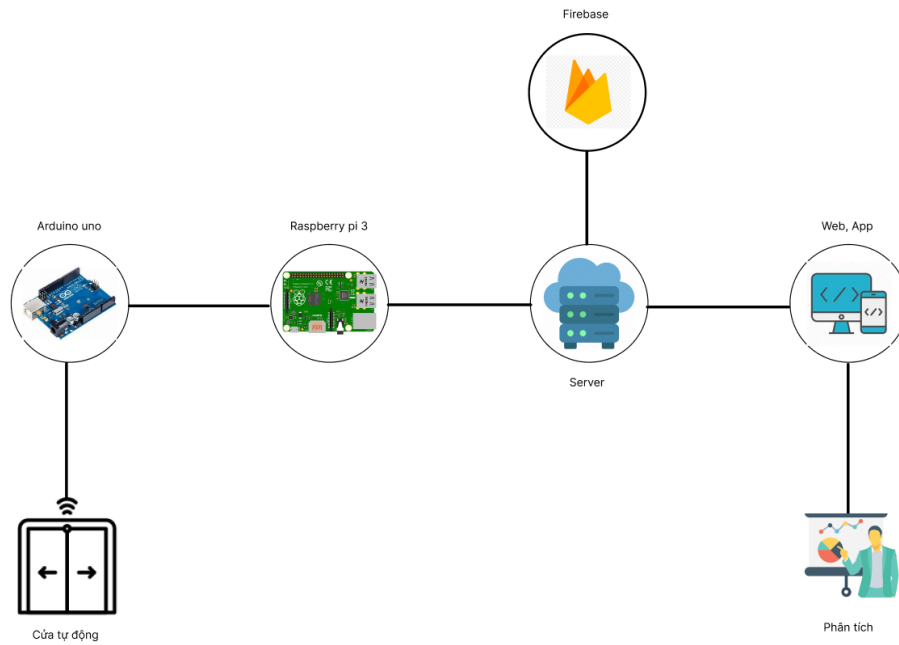
adapter lắp ngoài (tương tự như Bluetooth). Về mặt lý thuyết, USB không dây có thể cho phép 127 thiết bị truy cập cùng lúc trên một anten đơn (ít hơn so với mức 256 thiết bị / cổng của USB 2.0 thông thường). Tuy con số có phần khiêm tốn nhưng thế mạnh của WUSB là không cần dùng tới hub như USB.

Wireless USB sử dụng nền tảng UWB (ultra-wideband) với tốc độ lên đến 2 GB/giây, hoạt động trong dải tần UWB (Ultra Wide Band) với tần số từ 3.1GHz đến 10.6 GHz. Việc phân tán tín hiệu trên diện rộng như vậy giúp WUSB tránh được nhiều hiện tượng nhiễu so với các công nghệ không dây khác trên máy tính hiện nay (ví dụ như Bluetooth, Wifi sử dụng băng tần 2.4 GHz).

**WiMax:** WiMax là viết tắt của Worldwide Interoperability for Microwave Access - Khả năng tương tác trên toàn thế giới cho Truy cập Vi sóng. Công nghệ không dây này cho phép truyền dữ liệu với tốc độ 30-40 MB/giây. Thuật ngữ này đề cập cụ thể đến việc triển khai tương thích của họ không dây IEEE 802.16. Công nghệ này từng được một số hãng di động sử dụng, đặc biệt là Sprint, để cung cấp dữ liệu không dây cho khách hàng của mình. Sprint, cùng với nhiều nhà cung cấp dịch vụ khác đã sử dụng công nghệ này, rồi từ đó chuyển sang sử dụng mạng 4G LTE để truyền dữ liệu nhanh hơn. 20 Tên đề tài: Hệ thống khu vườn thông minh sử dụng Raspberry Pi 4 WiMax Forum sẽ xác nhận thiết bị trước khi bán chúng cho người tiêu dùng hoặc doanh nghiệp. Công nghệ này có thể được sử dụng cả trong nhà và ngoài trời. Tuy nhiên, các thiết bị WiMax thường có tín hiệu tốt hơn khi được sử dụng bên ngoài hoặc qua một cửa sổ.

## Chương 4. Thiết Kế Hệ Thống

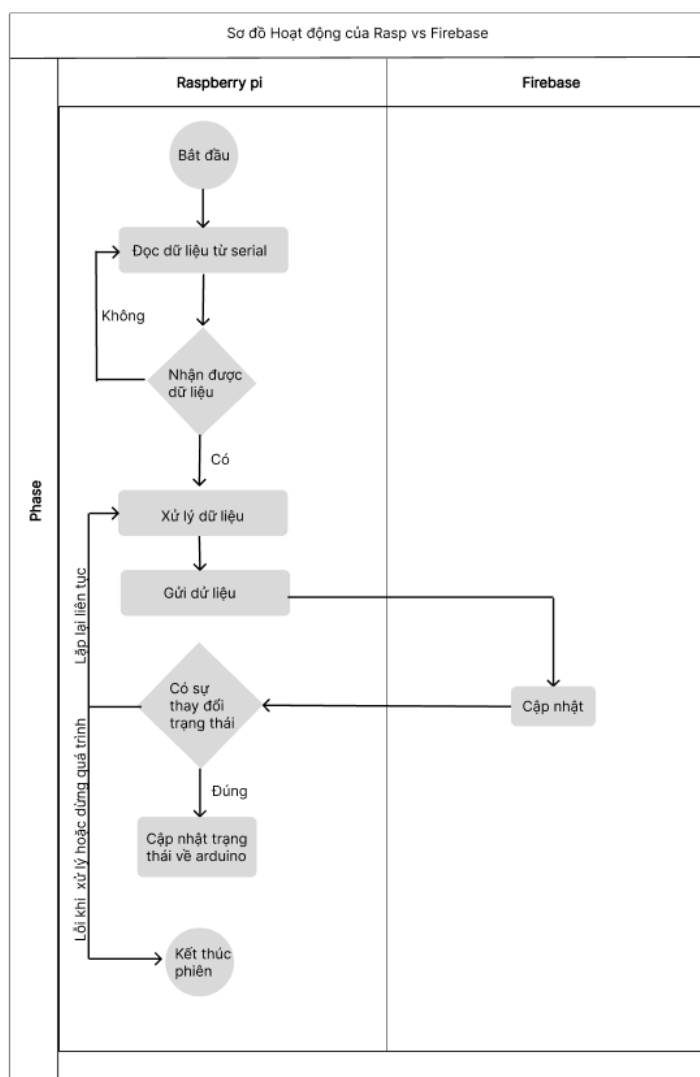
### 4.1. Kiến trúc tổng thể của hệ thống



Hình 7: Sơ đồ hệ thống

## 4.2. Thiết kế

### 4.2.1 Raspberry Pi 3 giao tiếp Firebase

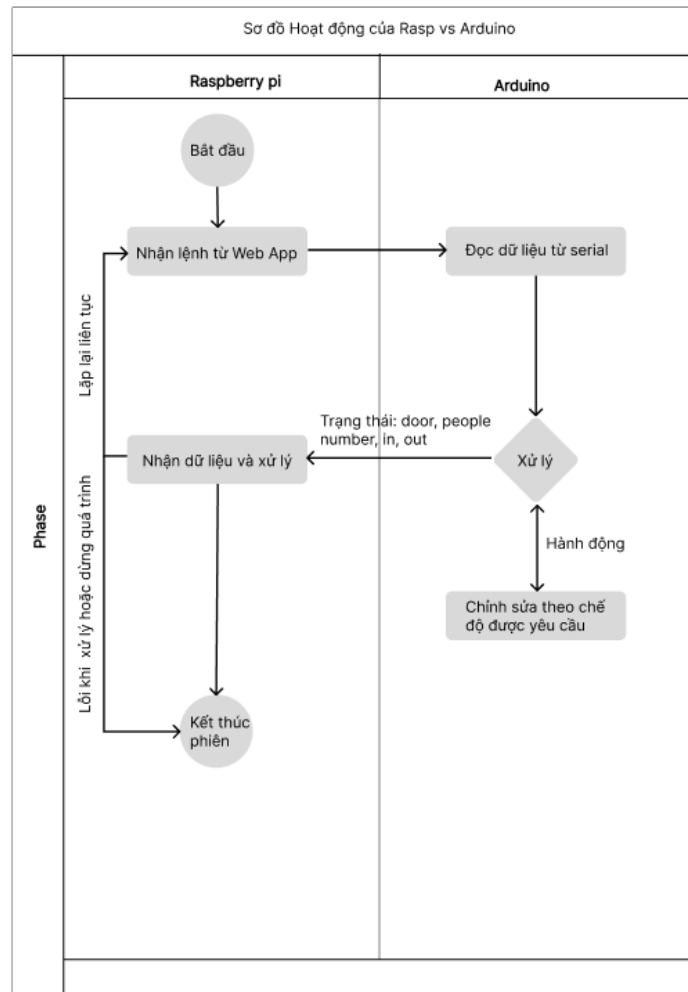


Hình 8: Sơ đồ hoạt động của Rasp với Firebase

Hệ thống Raspberry Pi kết hợp với Firebase hoạt động theo trình tự như sau: Quá trình bắt đầu khi Raspberry Pi được khởi động và liên tục đọc dữ liệu từ cổng serial, thường từ một thiết bị như Arduino. Khi nhận được dữ liệu, Raspberry Pi sẽ xử lý dữ liệu này và gửi lên Firebase. Nếu có sự thay đổi trạng thái, hệ thống sẽ cập nhật trạng thái mới lên Firebase. Trong trường hợp xảy ra lỗi khi xử lý hoặc cần dừng quá trình, trạng thái lỗi sẽ được gửi ngược lại về Arduino để thông báo. Quá trình

này lặp lại liên tục để đảm bảo dữ liệu luôn được cập nhật và hệ thống hoạt động bình thường.

#### 4.2.2 Raspberry Pi 3 giao tiếp Arduino Uno



Hình 9: sơ đồ hoạt động của rasp với Arduino

Hệ thống Raspberry Pi và Arduino hoạt động theo trình tự như sau: Quá trình bắt đầu khi cả Raspberry Pi và Arduino được khởi động. Raspberry Pi nhận lệnh điều khiển từ một ứng dụng web (Web App) và sau đó đọc dữ liệu từ cổng serial. Quá trình này được lặp lại liên tục. Arduino sẽ nhận dữ liệu từ Raspberry Pi và xử lý các lệnh, bao gồm các trạng thái như "door", "people number", "in", và "out". Sau khi

xử lý, Arduino thực hiện hành động tương ứng và chỉnh sửa theo chế độ được yêu cầu. Nếu xảy ra lỗi khi xử lý hoặc cần dừng quá trình, hệ thống sẽ ghi nhận và xử lý các lỗi này. Quá trình này tiếp diễn để đảm bảo hoạt động liên tục và ổn định của hệ thống.

#### **4.2.3 Các cảm biến và thiết bị điều khiển**

Hệ thống bao gồm các cảm biến và thiết bị điều khiển sau:

Servo điều khiển cửa:

Servo được sử dụng để mở và đóng cửa tự động. Góc đóng của servo được định nghĩa là GOC\_DONG (0 độ) và góc mở là GOC\_MO (90 độ).

Điều khiển servo thông qua chân servoPin (chân số 4).

Còi buzzer:

Còi buzzer được sử dụng để thông báo khi có người đi qua cửa. Khi phát hiện có người vào hoặc ra, còi buzzer sẽ kêu trong một khoảng thời gian ngắn.

Điều khiển còi buzzer thông qua chân buzzerPin (chân số 7).

LED thấp sáng khi có người trong phòng:

Một đèn LED được sử dụng để thấp sáng khi có người trong phòng. Đèn LED này sẽ bật khi có ít nhất một người trong phòng và tắt khi không có ai.

Điều khiển đèn LED thông qua chân ledPin (chân số 6).

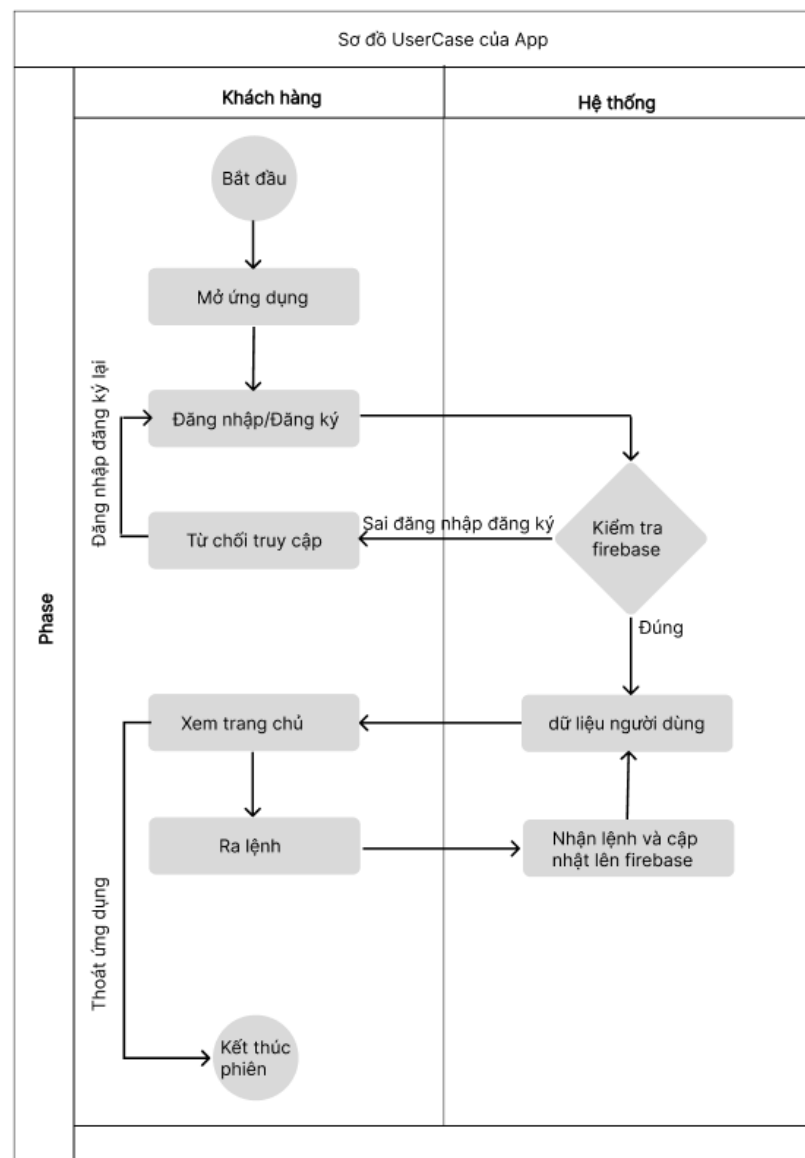
LED hiển thị trạng thái đóng cửa:

Một đèn LED khác được sử dụng để hiển thị trạng thái đóng cửa. Đèn LED này sẽ sáng khi cửa đang đóng và tắt khi cửa mở.

Điều khiển đèn LED trạng thái này thông qua chân ledPin\_close (chân số 8).

### 4.3. Thiết kế phần mềm

#### 4.3.1 Ứng dụng điện thoại



Hình 10: sơ đồ UserCase của ứng dụng

Ứng dụng Smart Home hoạt động theo trình tự như sau: Khi khách hàng mở ứng dụng, họ sẽ được yêu cầu đăng nhập hoặc đăng ký. Thông tin đăng nhập hoặc đăng ký được gửi đến hệ thống và kiểm tra với cơ sở dữ liệu Firebase. Nếu thông tin hợp lệ, khách hàng sẽ được cấp quyền truy cập và chuyển đến trang chủ, nơi hiển thị dữ liệu thời gian thực như số người trong cửa hàng, số người vào/ra, và trạng thái cửa. Từ đây, khách hàng có thể điều khiển trạng thái cửa (mở, đóng, tự động). Lệnh của khách hàng được hệ thống nhận và cập nhật lên Firebase để thay đổi trạng thái cửa. Khi khách hàng thoát ứng dụng, phiên làm việc kết thúc. Hệ thống đảm bảo thông tin luôn được cập nhật kịp thời và chính xác, mang lại trải nghiệm người dùng tốt nhất.

#### a) Giới Thiệu Chung

Ứng dụng di động Smart Home được phát triển nhằm cung cấp một nền tảng trực quan và thân thiện với người dùng. Ứng dụng này giúp người dùng dễ dàng theo dõi các hoạt động trong cửa hàng thông qua biểu đồ và thông tin thời gian thực, cũng như điều khiển trạng thái cửa ra vào.

#### b) Thành Phần Giao Diện

##### b.1) Intro và Đăng Nhập/Đăng Xuất

Intro: Màn hình giới thiệu ứng dụng với logo của hệ thống Smart Home, kèm theo tên và thông tin ngắn gọn về các tính năng của ứng dụng.

Đăng Nhập: Giao diện đăng nhập bao gồm các trường để người dùng nhập email và mật khẩu. Sau khi đăng nhập thành công, người dùng sẽ được chuyển đến màn hình chính.

Đăng Xuất: Nút đăng xuất được đặt ở vị trí dễ tìm thấy trên màn hình chính hoặc trong menu, cho phép người dùng đăng xuất khỏi hệ thống một cách nhanh chóng và an toàn.



## b.2) Màn Hình Home

Logo: Logo của Smart Home được đặt ở đầu trang.

Thông tin: số người ra vào, chế độ hoạt động, số người trong nhà.

Chart: biểu đồ cho thấy sự tương quan của số lượng người ra vào.

Button: thực hiện lệnh close, open, auto cửa.

## c) Tính Năng

Cập Nhật Thời Gian Thực: Ứng dụng liên tục cập nhật dữ liệu từ cơ sở dữ liệu Firebase để đảm bảo thông tin hiển thị luôn mới nhất.

Thông Báo Trạng Thái: Hiển thị thông báo khi người dùng thực hiện các hành động như mở cửa, đóng cửa, hoặc thay đổi trạng thái cửa.

Trực Quan Hóa Dữ Liệu: Sử dụng thư viện `com.github.PhilJay:MPAndroidChart.1.0` để trực quan hóa dữ liệu, giúp người dùng dễ dàng hiểu và phân tích thông tin.

## d) Công Nghệ Sử Dụng

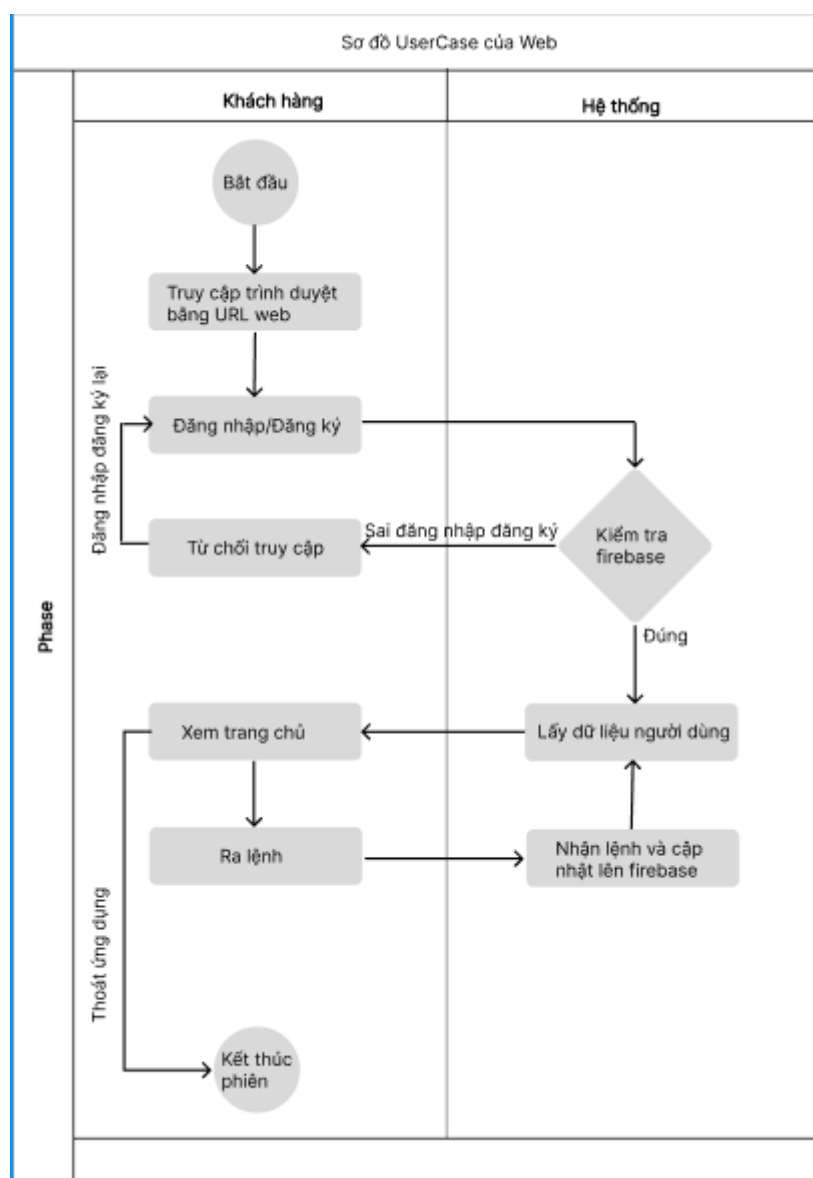
Firebase: Được sử dụng để quản lý dữ liệu thời gian thực và xác thực người dùng.

ProGuard: Được sử dụng để tối ưu hóa và bảo mật mã nguồn của ứng dụng.  
`com.github.PhilJay:MPAndroidChart.1.0`: Được sử dụng để vẽ biểu đồ và trực quan hóa dữ liệu.

## e) Đánh Giá

Ứng dụng di động được thiết kế với tiêu chí dễ sử dụng, trực quan và cung cấp đầy đủ thông tin cần thiết cho người dùng. Tuy nhiên, trong quá trình sử dụng thực tế, các cải tiến có thể được thực hiện để tối ưu hóa trải nghiệm người dùng hơn nữa.

### 4.3.2 Giao diện web



Hình 11: Sơ đồ UserCase của web

Cách hoạt động của ứng dụng web Smart Home bắt đầu khi khách hàng truy cập trình duyệt và nhập URL của ứng dụng. Sau đó, hệ thống yêu cầu người dùng đăng nhập hoặc đăng ký tài khoản. Nếu thông tin đăng nhập hoặc đăng ký không chính xác, hệ thống từ chối truy cập. Hệ thống kiểm tra thông tin đăng nhập với Firebase. Nếu thông tin đúng, người dùng sẽ được chuyển đến trang chủ, nơi hệ thống lấy dữ liệu người dùng từ Firebase và hiển thị trên giao diện. Người dùng có thể ra lệnh điều khiển trạng thái cửa, và hệ thống sẽ nhận lệnh và cập nhật lên Firebase. Khi người dùng thoát khỏi trình duyệt, phiên làm việc kết thúc.

#### a) Giới Thiệu Chung

Giao diện web của hệ thống Smart Home được thiết kế với mục đích cung cấp một nền tảng trực quan và thân thiện với người dùng. Giao diện này giúp người dùng dễ dàng theo dõi các hoạt động trong cửa hàng thông qua các biểu đồ và thông tin thời gian thực, cũng như điều khiển trạng thái cửa ra vào.

#### b) Thành Phần Giao Diện

##### b.1) Tiêu Đề và Chào Mừng

Tiêu Đề: "Smart Home" được đặt ở đầu trang với kiểu chữ đậm và màu sắc nổi bật, giúp người dùng dễ dàng nhận diện hệ thống.

Thông Điệp Chào Mừng: Hiển thị thông điệp chào mừng kèm theo email của người dùng đăng nhập, tạo cảm giác thân thiện và cá nhân hóa trải nghiệm.

##### b.2) Khối Dữ Liệu Cửa Hàng

Cơ Sở Dữ Liệu: "Home Data" được hiển thị rõ ràng để phân biệt với các phần khác của giao diện.

Thông Tin Thời Gian Thực:

Số Người Trong Cửa Hàng: Hiển thị số lượng người hiện tại đang có mặt trong cửa hàng.

Số Người Vào: Hiển thị số người đã vào cửa hàng (input).

Số Người Ra: Hiển thị số người đã ra khỏi cửa hàng (output).

Trạng Thái Cửa: Hiển thị trạng thái hiện tại của cửa (mở, đóng, tự động).

### b.3) Biểu Đồ

Biểu Đồ Số Người Trong Cửa Hàng: Sử dụng biểu đồ đường để hiển thị số lượng người trong cửa hàng theo thời gian, giúp người dùng theo dõi sự thay đổi số lượng người một cách trực quan.

Biểu Đồ Trạng Thái Cửa: Sử dụng biểu đồ cột để hiển thị số lần cửa ở mỗi trạng thái (mở, đóng, tự động), cung cấp cái nhìn tổng quan về tần suất sử dụng các trạng thái cửa.

### b.4) Điều Khiển Cửa

Nút Điều Khiển:

Mở Cửa: Nút để mở cửa.

Đóng Cửa: Nút để đóng cửa.

Cửa Tự Động: Nút để chuyển cửa sang chế độ tự động.

Nút Chuyển Đổi Cập Nhật: Nút "Toggle Update Interval" cho phép người dùng bật hoặc tắt chế độ cập nhật tự động.

#### b.5) Đăng Xuất

Nút Đăng Xuất: Cho phép người dùng đăng xuất khỏi hệ thống, đảm bảo an toàn và bảo mật thông tin cá nhân.

#### c) Tính Năng

Cập Nhật Thời Gian Thực: Giao diện cập nhật liên tục dữ liệu từ cơ sở dữ liệu Firebase để đảm bảo thông tin hiển thị luôn là mới nhất.

Thông Báo Trạng Thái: Hiển thị thông báo khi người dùng thực hiện các hành động như mở cửa, đóng cửa, hoặc thay đổi trạng thái cửa.

Trực Quan Hóa Dữ Liệu: Sử dụng Chart.js để trực quan hóa dữ liệu, giúp người dùng dễ dàng hiểu và phân tích thông tin.

#### d) Công Nghệ Sử Dụng

HTML và CSS: Để cấu trúc và tạo kiểu cho giao diện người dùng.

JavaScript: Để xử lý logic phía khách hàng và tương tác với Firebase.

Firebase: Để quản lý dữ liệu thời gian thực và xác thực người dùng.

Chart.js: Để vẽ biểu đồ và trực quan hóa dữ liệu.

#### e) Đánh Giá

Giao diện web của Smart Home đã được thiết kế với tiêu chí dễ sử dụng, trực quan và cung cấp đầy đủ thông tin cần thiết cho người dùng. Tuy nhiên, trong quá trình sử dụng thực tế, các cải tiến có thể được thực hiện để tối ưu hóa trải nghiệm người dùng hơn nữa.

## Chương 5. Triển Khai Hệ Thống

### 5.1. Cài đặt và cấu hình Raspberry Pi 3

#### Nhận dữ liệu từ cổng serial của arduino

```
# Cổng serial của Arduino
arduino_port = '/dev/ttyACM0'
baud_rate = 9600

arduino = serial.Serial(arduino_port, baud_rate, timeout=1)
```

Thiết lập cổng cần kết nối đến là "ttyACM0"

Tốc độ truyền 9600

Tạo một đối tượng Arduino thuộc thư viện serial với các thông số đã được khai báo. Đối tượng này giúp tương tác với cổng serial, cho phép gửi và nhận dữ liệu giữa Raspberry pi 3 với Arduino.

#### Kết nối đến firebase

```
# Cấu hình Firebase
cred = credentials.Certificate("key.json")
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://smart-store-iot-default-rtdb.asia-southeast1.firebaseio.com'
})

# Tham chiếu đến các nút trong Firebase
user_ref = db.reference('users/L1mfXWVYZMasBvjXwPJSrL4Q7od2/MyHome')
input_people_ref = user_ref.child('input')
output_people_ref = user_ref.child('output')
number_people_ref = user_ref.child('number_people')
status_door_ref = user_ref.child('status_door')
```

Thư viện Firebase Admin được sử dụng để tương tác giữa Raspberry Pi 3 với Firebase

Đầu tiên tạo một đối tượng chứng chỉ cred bằng cách đọc file "key.json". File "key.json" chứa thông tin xác thực cần thiết để kết nối với Firebase project. File này được tải xuống từ Firebase console.

Dòng thứ 2 khởi tạo đối tượng Firebase với thông tin xác thực và URL tới Realtime Database.

Tạo một tham chiếu đến nút Myhome trong Database thông qua thương phức `db.reference()` và đưa vào đối tượng `user_ref`

Sau đó khởi tạo đối tượng tham chiếu đến các nút con nằm trong Myhome để thuận tiện cho việc đọc và ghi các chỉ số input (số người vào); output (số người ra); `number_people` (số lượng người còn lại trong bệnh viện); `statusdoor` (trạng thái cửa open, close, auto).

## Cập nhật dữ liệu từ Arduino lên Firebase

```
while True:
    # Đọc dữ liệu từ cổng serial của Arduino
    data = arduino.readline().decode().strip()

    if data:
        try:
            # Phân tích chuỗi JSON
            data_json = json.loads(data)
            print("Dữ liệu từ Arduino:", data_json) # Hiển thị dữ liệu
            input_value = data_json.get("input")
            output_value = data_json.get("output")
            # Cập nhật số lượng người lên Firebase
            if input_value is not None and output_value is not None:
                input_number = float(input_value)
                output_number = float(output_value)

                # Thực hiện phép tính trừ
                count = input_number - output_number
                input_people_ref.set(input_value)
                output_people_ref.set(output_value)
                number_people_ref.set(count)
                print("Số lượng người:", count)
            else:
                print("Không thể trích xuất dữ liệu từ tệp JSON.")

        except (ValueError, json.JSONDecodeError) as e:
            print("Lỗi phân tích dữ liệu:", e)
```

Sử dụng vòng lặp liên tục để đọc dữ liệu từ Arduino

Đọc dữ liệu từ cổng serial, giải mã từ byte sang string, loại bỏ khoảng trắng đầu cuối bằng `arduino().readline().decode()` sau đó gán cho biến `data`

Phân tích chuỗi Json từ arduino bằng cách chuyển nó sang một đối tượng của python thông qua hàm `json.loads()`

Sau đó tiếp tục thực hiện đọc số liệu người vào ra từ arduino sau đó tính toán số người còn lại trong bệnh viện và cập nhật dữ liệu lên Firebase thông qua phương thức .set()

### **Cập nhật lại trạng thái cửa về Arduino nếu có sự thay đổi**

```
# Lấy trạng thái cửa từ Firebase
door_status = status_door_ref.get()

# Kiểm tra xem trạng thái cửa có thay đổi không
if door_status != current_door_status:
    current_door_status = door_status
    door_status_changed = True

# Nếu có sự thay đổi trong trạng thái cửa
if door_status_changed:
    print("Trạng thái cửa từ Firebase:", door_status)

    # Gửi lệnh tương ứng đến Arduino
    if door_status == "open":
        arduino.write(b'2')
    elif door_status == "close":
        arduino.write(b'1')
    elif door_status == "auto":
        arduino.write(b'3')

    # Đặt lại biến cờ
    door_status_changed = False
```

Lấy trạng thái cửa từ firebase từ tham chiếu status\_door\_def thông qua phương thức get()

Sau đó sẽ kiểm tra trạng thái hiện tại và trạng thái hiện tại và trạng thái cửa mới, nếu có thay đổi thì sẽ cập nhật về lại Arduino.

## **5.2. Lập trình và cấu hình Arduino Uno**

**Tự động đóng mở cửa :**



```

// Code tự động điều khiển cửa
if (door_auto == true) {
  if (distanceIn < Max && distanceIn > Min) {
    if (distanceIn > 25 && distanceIn < 35) {
      if (distanceIn > distanceOld && isUpdate == false) {
        output++;
        beep(500);
        isUpdate = true;
      } else if (distanceIn < distanceOld && isUpdate == false) {
        input++;
        beep(500);
        isUpdate = true;
      }
    } else {
      isUpdate = false;
    }
    servo.write(GOC_MO);
  } else {
    servo.write(GOC_DONG);
  }
  distanceOld = distanceIn;
  if ((input - output) > 0) {
    digitalWrite(ledPin, HIGH); // Tắt đèn LED.
  } else {
    digitalWrite(ledPin, LOW); // Bật đèn LED.
  }
}

```

Kiểm tra khoảng cách để điều khiển tự động cửa mở/đóng dựa trên sự hiện diện của người trong phạm vi cảm biến. Cập nhật số người vào/ra, và điều khiển đèn LED dựa trên số người hiện tại trong cửa hàng.

### Thực hiện lắng nghe lệnh từ serial:

```

// Code điều khiển servo và đèn LED
if (Serial.available() > 0) {
  command = Serial.read();

  if (command == '1') {
    servo.write(GOC_DONG); // đóng cửa khi nhận '1' từ Serial
    door_auto = false;
    door_status = 1;
    digitalWrite(ledPin_close, HIGH);
  } else if (command == '2') {
    servo.write(GOC_MO); // mở cửa khi nhận '2' từ Serial
    door_auto = false;
    door_status = 2;
    digitalWrite(ledPin_close, LOW);
  } else if (command == '3') {
    door_auto = true;
    door_status = 3;
    digitalWrite(ledPin_close, LOW);
  }
}

```

Kiểm tra lệnh nhận được từ Serial và điều khiển cửa mở/đóng hoặc chuyển sang chế độ tự động.

### Cập nhật dữ liệu lên serial dưới dạng json theo thời gian:

```
unsigned long currentMillis = millis();
if (currentMillis - previousSensorMillis >= sensorInterval) {
    previousSensorMillis = currentMillis;

    // Cập nhật giá trị lên serial
    char jsonBuffer[100];
    sprintf(jsonBuffer, "{
        \"input\": %d,
        \"output\": %d,
        \"door_status\": %d,
        \"distance\": %d
    }",
            input, output, door_status, distanceIn);

    // In chuỗi JSON ra Serial để debug
    Serial.println(jsonBuffer);
}
```

### 5.3. Phát triển ứng dụng điện thoại

Trong quá trình phát triển ứng dụng điện thoại cho hệ thống Smart Home, mục tiêu là tạo ra một ứng dụng thân thiện với người dùng, dễ dàng truy cập và cung cấp các tính năng quản lý và giám sát thời gian thực. Quá trình phát triển bao gồm các bước thiết kế giao diện, lập trình, và kiểm thử để đảm bảo ứng dụng hoạt động mượt mà và chính xác. Dưới đây là các giao diện chính của ứng dụng:

#### Giao diện intro app:

Giao diện intro của ứng dụng cung cấp một cái nhìn tổng quan về các tính năng chính và lợi ích của ứng dụng. Đây là điểm khởi đầu giúp người dùng làm quen với ứng dụng trước khi bắt đầu sử dụng.

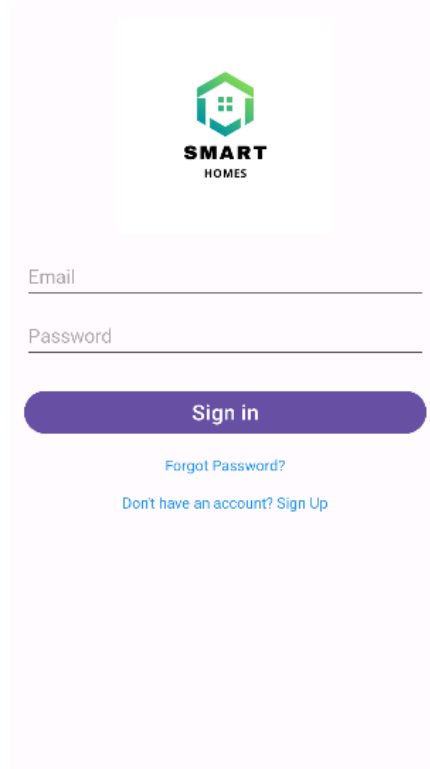


GET STARTED!

*Hình 12: giao diện intro app*

**Giao diện đăng nhập:**

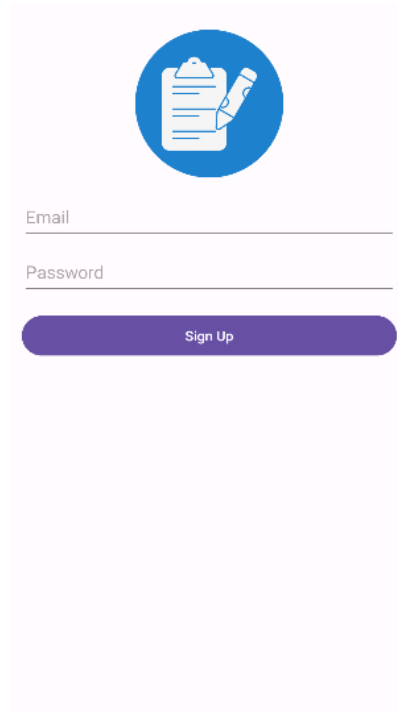
Giao diện đăng nhập cho phép người dùng nhập thông tin tài khoản để truy cập vào hệ thống. Giao diện này đảm bảo tính bảo mật và dễ sử dụng, giúp người dùng nhanh chóng đăng nhập vào ứng dụng.

The image shows a login interface for 'SMART HOMES'. At the top center is a logo consisting of a green house icon with a white window, above the word 'SMART' in bold black letters and 'HOMES' in smaller black letters below it. Below the logo are two input fields: 'Email' and 'Password', each with a horizontal line for text entry. Under the password field is a purple rounded rectangular button with the text 'Sign in' in white. Below the button are two links in blue text: 'Forgot Password?' and 'Don't have an account? Sign Up'.

*Hình 13: giao diện đăng nhập*

### **Giao diện đăng ký:**

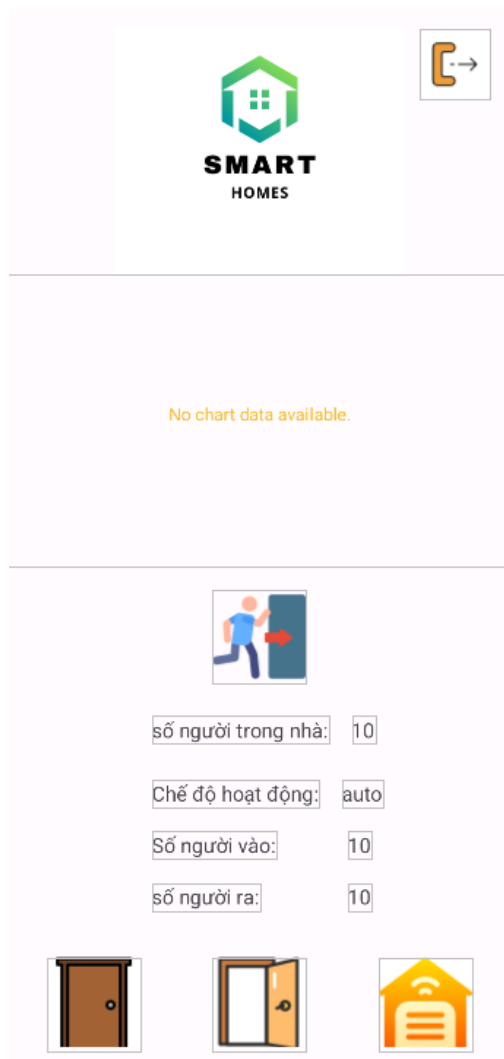
Giao diện đăng ký được thiết kế để người dùng mới có thể tạo tài khoản một cách dễ dàng. Các trường thông tin cần thiết được hiển thị rõ ràng và hướng dẫn cụ thể để đảm bảo quá trình đăng ký diễn ra thuận lợi.

A sign-up form with a light pink background. At the top center is a blue circular icon containing a white clipboard and pencil. Below the icon are two input fields: the first is labeled 'Email' and the second is labeled 'Password'. Both labels are in a small, light gray font. Below the password field is a purple rounded rectangular button with the text 'Sign Up' in white.

*Hình 14: giao diện đăng ký*

### **Giao diện Home:**

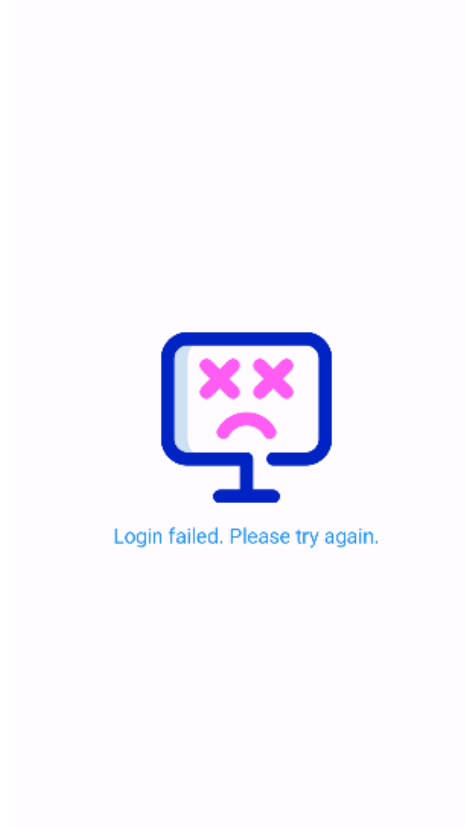
Giao diện home là nơi người dùng quản lý và giám sát các hoạt động của hệ thống Smart Store Monitor. Từ đây, người dùng có thể xem thông tin thời gian thực về số lượng người trong cửa hàng, trạng thái cửa, và các biểu đồ trực quan khác.



Hình 15: giao diện home (quản lý)

### **Giao diện đăng nhập sai cần đăng nhập lại:**

Giao diện đăng nhập sai sẽ xuất hiện khi người dùng nhập sai thông tin đăng nhập. Giao diện này cung cấp thông báo lỗi và yêu cầu người dùng thử lại, đảm bảo an toàn và bảo mật cho hệ thống.



*Hình 16: giao diện đăng nhập sai cần đăng nhập lại*

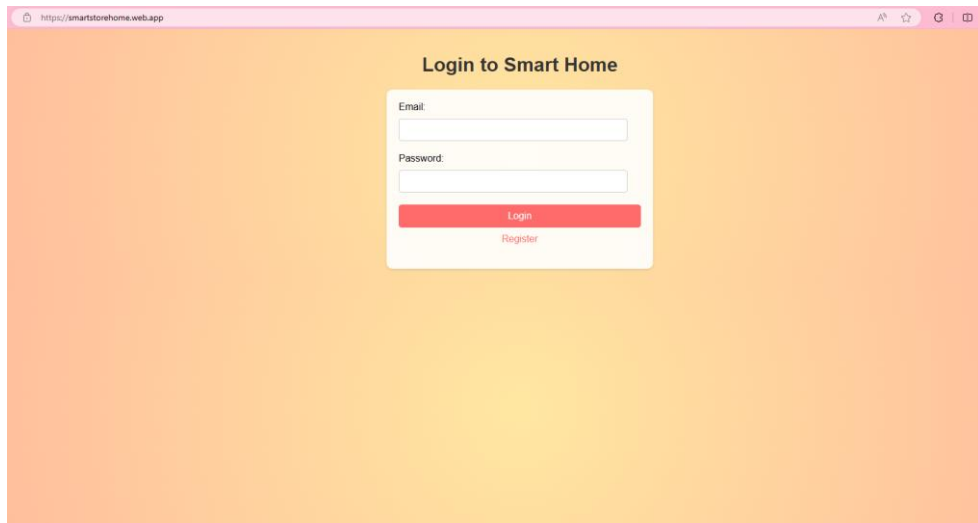
#### **5.4. Phát triển giao diện web**

Quá trình phát triển giao diện web cho hệ thống Smart Home tập trung vào việc tạo ra một nền tảng trực quan, dễ sử dụng và đáp ứng nhu cầu quản lý thời gian thực cho người dùng. Phát triển giao diện web bao gồm các giai đoạn từ thiết kế giao diện, lựa chọn công nghệ, phát triển và kiểm thử, đảm bảo sản phẩm cuối cùng hoạt động mượt mà và chính xác.

##### **a. Phân Tích Yêu Cầu**

- **Yêu Cầu Người Dùng:** Tìm hiểu các yêu cầu cụ thể từ phía người dùng, bao gồm khả năng theo dõi số lượng người trong cửa hàng, kiểm soát trạng thái cửa và xem các biểu đồ trực quan.
- **Yêu Cầu Kỹ Thuật:** Đảm bảo tính tương thích với các thiết bị và trình duyệt khác nhau, khả năng cập nhật dữ liệu thời gian thực và tính bảo mật cao.

## b. Thiết Kế Giao Diện



*Hình 17: giao diện đăng nhập web*





Hình 18: giao diện chính của web

### c. Lựa Chọn Công Nghệ

- **Frontend:** Sử dụng HTML, CSS và JavaScript để xây dựng giao diện người dùng.
- **Thư Viện và Frameworks:** Sử dụng Chart.js để tạo biểu đồ, Firebase để quản lý cơ sở dữ liệu và xác thực người dùng.
- **Công Cụ Phát Triển:** Sử dụng các công cụ như Visual Studio Code, Git để quản lý mã nguồn và cộng tác.

### d. Phát Triển

- **Tạo Cấu Trúc HTML:** Thiết kế các phần chính của trang web như tiêu đề, khối dữ liệu, biểu đồ và nút điều khiển.
- **Tạo Kiểu với CSS:** Sử dụng CSS để tạo kiểu cho trang web, bao gồm các hiệu ứng chuyển động và responsive để trang web hiển thị tốt trên mọi thiết bị.
- **Lập Trình JavaScript:** Viết mã JavaScript để xử lý logic, tương tác với Firebase, cập nhật dữ liệu thời gian thực và vẽ biểu đồ.

### e. Kiểm Thử

- **Kiểm Thử Chức Năng:** Đảm bảo tất cả các chức năng như cập nhật dữ liệu, hiển thị biểu đồ và điều khiển cửa hoạt động chính xác.
- **Kiểm Thử Giao Diện:** Đảm bảo giao diện hiển thị đúng và đẹp trên các trình duyệt và thiết bị khác nhau.
- **Kiểm Thử Hiệu Năng:** Đảm bảo trang web tải nhanh và không gặp sự cố khi xử lý dữ liệu thời gian thực.

## 5.5. Kết Quả và Đánh Giá

Sau quá trình phát triển, giao diện web của hệ thống Smart Home đã đạt được các mục tiêu đề ra, bao gồm:

- **Giao Diện Trực Quan và Thân Thiện:** Người dùng dễ dàng truy cập và sử dụng các chức năng của hệ thống.
- **Cập Nhật Thời Gian Thực:** Dữ liệu từ cửa hàng được cập nhật liên tục, giúp người dùng luôn nắm bắt được tình hình hiện tại.
- **Tính Bảo Mật Cao:** Đảm bảo an toàn thông tin người dùng và dữ liệu cửa hàng.
- **Hiệu Năng Tốt:** Trang web tải nhanh và hoạt động mượt mà ngay cả khi xử lý nhiều dữ liệu.

## 5.6. Tích hợp và kiểm tra hệ thống

## **Chương 6. Kết Quả Thử Nghiệm**

### **6.1. Thử nghiệm tính năng điều khiển từ xa**

Trong thử nghiệm tính năng điều khiển từ xa, chúng tôi đã sử dụng ứng dụng web và app để gửi các lệnh điều khiển đến hệ thống Raspberry Pi và Arduino. Hệ thống đã thành công trong việc nhận và thực hiện các lệnh điều khiển từ xa, bao gồm việc cập nhật trạng thái thiết bị và gửi phản hồi về ứng dụng web. Kết quả thử nghiệm cho thấy độ trễ trong việc thực hiện lệnh là nhỏ, đảm bảo tính thời gian thực trong điều khiển.

### **6.2. Thử nghiệm tính năng mở/đóng cửa tự động**

Chúng tôi đã tiến hành thử nghiệm tính năng mở/đóng cửa tự động dựa trên các cảm biến phát hiện người. Khi cảm biến nhận diện có người đến gần cửa, lệnh mở cửa được gửi từ Arduino và cửa tự động mở. Ngược lại, khi không có người trong phạm vi cảm biến, lệnh đóng cửa được thực hiện. Tính năng này hoạt động ổn định và chính xác, với thời gian phản hồi nhanh, đảm bảo sự tiện lợi và an toàn cho người sử dụng.

### **6.3. Đánh giá hiệu quả hoạt động của hệ thống**

Hệ thống kết hợp giữa Raspberry Pi và Arduino đã được đánh giá dựa trên các tiêu chí: độ tin cậy, độ chính xác, và hiệu suất hoạt động. Qua các thử nghiệm, hệ thống đã chứng minh được độ tin cậy cao với rất ít lỗi phát sinh. Độ chính xác của các lệnh điều khiển và cảm biến cũng rất tốt, đảm bảo các hành động được thực hiện đúng như mong đợi. Về hiệu suất, hệ thống có khả năng xử lý và phản hồi nhanh chóng, đáp ứng tốt yêu cầu của các tính năng điều khiển từ xa và tự động. Kết quả

đánh giá cho thấy hệ thống hoạt động hiệu quả, đáp ứng được các yêu cầu đặt ra trong thiết kế và triển khai.

## **Chương 7. Kết Luận và Đánh Giá**

### **7.1. Phân tích ưu điểm của hệ thống**

#### **Tính linh hoạt và khả năng mở rộng**

**Arduino:** Sử dụng Arduino để kết nối với các cảm biến mang lại sự linh hoạt cao vì Arduino có thể dễ dàng tương thích với nhiều loại cảm biến khác nhau.

**Raspberry Pi 3:** Đóng vai trò trung tâm xử lý, giúp dễ dàng mở rộng hệ thống bằng cách thêm các chức năng mới thông qua các thư viện phần mềm.

#### **Khả năng kết nối mạnh mẽ**

**Kết nối Serial:** Raspberry Pi kết nối với Arduino thông qua cổng serial, đảm bảo việc truyền dữ liệu nhanh chóng và đáng tin cậy giữa hai thiết bị.

**WiFi:** Raspberry Pi 3 tích hợp sẵn WiFi, cho phép kết nối Internet dễ dàng và gửi dữ liệu đến Firebase một cách hiệu quả.

#### **Khả năng lưu trữ và quản lý dữ liệu trực tuyến**

**Firebase:** Là một nền tảng đám mây mạnh mẽ, Firebase cung cấp dịch vụ cơ sở dữ liệu thời gian thực, giúp dữ liệu từ các cảm biến được cập nhật liên tục và đồng bộ hóa ngay lập tức trên các thiết bị khách (điện thoại, trang web).

#### **Điều khiển từ xa và giao diện người dùng tiện lợi**

Ứng dụng di động và trang web: Việc xây dựng ứng dụng và trang web cho phép người dùng theo dõi và điều khiển hệ thống từ xa một cách thuận tiện. Người dùng có thể nhận thông báo, xem dữ liệu thời gian thực và thay đổi trạng thái cửa (open, close, auto) dễ dàng từ bất kỳ đâu.

## **Tính bảo mật và an toàn**

**Firestore:** Cung cấp nhiều tính năng bảo mật như xác thực người dùng và quản lý quyền truy cập, giúp bảo vệ dữ liệu và đảm bảo rằng chỉ những người dùng được ủy quyền mới có thể thay đổi trạng thái của hoặc truy cập dữ liệu.

## **Phân tích và giám sát dữ liệu hiệu quả**

**Phân tích dữ liệu:** Firestore có thể tích hợp với các công cụ phân tích khác như Google Analytics, giúp bạn thu thập và phân tích dữ liệu người dùng và hành vi sử dụng một cách hiệu quả, từ đó đưa ra các cải tiến cho hệ thống.

## **Tiết kiệm chi phí và thời gian phát triển**

Phần cứng và phần mềm mã nguồn mở: Sử dụng Arduino và Raspberry Pi, cả hai đều là các nền tảng mã nguồn mở và chi phí thấp, giúp tiết kiệm chi phí cho việc phát triển hệ thống. Ngoài ra, có một cộng đồng lớn hỗ trợ, giúp bạn dễ dàng tìm kiếm giải pháp cho các vấn đề phát sinh.

## **Dễ dàng bảo trì và nâng cấp**

**Modular System:** Hệ thống có tính modular (module hóa), dễ dàng thay thế và nâng cấp từng thành phần mà không cần phải thay đổi toàn bộ hệ thống. Điều này giúp giảm thiểu thời gian chết và chi phí bảo trì.

## **7.2. Đánh giá hạn chế và khó khăn**

Trong quá trình phát triển và triển khai hệ thống Smart Home, chúng tôi đã gặp phải một số hạn chế và khó khăn cần được đánh giá và ghi nhận. Dưới đây là các vấn đề chính:

### **Hạn chế về Phần cứng**

Hiệu suất của Raspberry Pi: Mặc dù Raspberry Pi là một thiết bị mạnh mẽ và linh hoạt, nó có giới hạn về hiệu suất so với các máy chủ hoặc thiết bị

chuyên dụng khác. Điều này có thể ảnh hưởng đến khả năng xử lý dữ liệu lớn hoặc phức tạp trong thời gian thực.

**Độ chính xác của cảm biến:** Các cảm biến được sử dụng để phát hiện người và theo dõi trạng thái cửa có thể không hoàn toàn chính xác trong mọi tình huống, đặc biệt là trong các môi trường có nhiều nhiễu.

### Khó khăn về Kết nối và Truyền thông

**Độ trễ mạng:** Hệ thống phụ thuộc vào kết nối mạng để truyền dữ liệu giữa các thành phần. Độ trễ mạng hoặc mất kết nối có thể ảnh hưởng đến hiệu suất và độ tin cậy của hệ thống.

**Khả năng mở rộng:** Việc mở rộng hệ thống để quản lý nhiều cửa hàng hoặc thêm nhiều cảm biến và thiết bị mới có thể gặp khó khăn về mặt kỹ thuật và tài chính.

## 7.3. Khả năng mở rộng và phát triển trong tương lai

### Cải thiện khả năng nhận diện và kiểm soát truy cập:

- **Nhận diện khuôn mặt và vân tay:** Tích hợp các công nghệ nhận diện khuôn mặt và vân tay để tăng cường bảo mật và thuận tiện cho nhân viên y tế, bệnh nhân và khách.
- **Thẻ RFID:** Sử dụng thẻ RFID để kiểm soát truy cập, đảm bảo chỉ những người có thẩm quyền mới có thể vào các khu vực quan trọng như phòng phẫu thuật, phòng thuốc, v.v.

### Tích hợp các cảm biến và thiết bị y tế:

- **Cảm biến nhiệt độ và chất lượng không khí:** Giám sát nhiệt độ và chất lượng không khí trong phòng để đảm bảo môi trường luôn sạch sẽ và an toàn, đặc biệt quan trọng trong các phòng mổ và khu vực chăm sóc đặc biệt.
- **Tích hợp với hệ thống báo động y tế:** Kết nối với hệ thống báo động y tế để tự động mở cửa khi có sự cố khẩn cấp hoặc khi cần di tản bệnh nhân nhanh chóng.

### Tăng cường khả năng tương tác và thông báo:

- **Màn hình hiển thị thông tin:** Đặt màn hình hiển thị ở cửa ra vào để cung cấp thông tin hữu ích như hướng dẫn y tế, thông báo quan trọng hoặc thông tin phòng.
- **Thông báo đẩy:** Sử dụng thông báo đẩy qua ứng dụng điện thoại để thông báo cho nhân viên y tế khi có người đến hoặc khi cửa cần được bảo trì.

### **Tối ưu hóa hệ thống điều khiển và giám sát từ xa:**

- **Điều khiển bằng giọng nói:** Tích hợp công nghệ nhận diện giọng nói để nhân viên y tế có thể mở hoặc đóng cửa mà không cần tiếp xúc trực tiếp, giảm nguy cơ lây nhiễm.
- **Giám sát từ xa:** Phát triển các tính năng giám sát từ xa qua camera để quản lý và giám sát hoạt động tại các khu vực quan trọng từ phòng điều khiển trung tâm.

### **Cải thiện tính năng tự động và thông minh:**

- **Học máy (Machine Learning):** Áp dụng công nghệ học máy để hệ thống có thể học và dự đoán các mô hình di chuyển của nhân viên và bệnh nhân, từ đó tối ưu hóa hoạt động mở/đóng cửa.
- **Chế độ tiết kiệm năng lượng:** Triển khai các chế độ tiết kiệm năng lượng cho cửa tự động khi không có người qua lại, giúp giảm tiêu thụ điện năng và chi phí vận hành.

### **Tăng cường an toàn và phòng chống lây nhiễm:**

- **Hệ thống khử trùng tự động:** Tích hợp hệ thống khử trùng tự động trên tay nắm cửa hoặc vùng tiếp xúc để giảm nguy cơ lây nhiễm.
- **Cửa kháng khuẩn:** Sử dụng vật liệu kháng khuẩn cho bề mặt cửa và tay nắm để đảm bảo vệ sinh tối ưu.

### **Tích hợp với các hệ thống khác của bệnh viện:**

- **Hệ thống quản lý bệnh viện (HIS):** Kết nối với hệ thống quản lý bệnh viện để quản lý thông tin bệnh nhân và nhân viên hiệu quả hơn.
- **Hệ thống quản lý cơ sở vật chất (FMIS):** Tích hợp với hệ thống quản lý cơ sở vật chất để theo dõi và bảo trì cửa tự động kịp thời.

### **Phân tích dữ liệu và cải tiến liên tục:**

- **Phân tích dữ liệu sử dụng:** Thu thập và phân tích dữ liệu sử dụng cửa để tối ưu hóa hoạt động, phát hiện các vấn đề tiềm ẩn và cải tiến hiệu suất.
- **Phản hồi người dùng:** Tạo kênh thu thập phản hồi từ nhân viên y tế và bệnh nhân để liên tục cải tiến hệ thống dựa trên nhu cầu thực tế.

## **Tài Liệu Tham Khảo**

[802.11 Standards Explained: 802.11ax, 802.11ac, 802.11b/g/n, 802.11a \(lifewire.com\)](#)

[https://community.fs.com/article/802-11-standards-explained.html](#)



