# TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

**VIỆN KỸ THUẬT – CÔNG NGHỆ**

****

**PHÁT TRIỂN ỨNG DỤNG ĐA NỀN TẢNG**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ VÀ KẾT NỐI**

**CÁC THIẾT BỊ CHO NHÀ THÔNG MINH**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Trần Văn Hữu |
| **Sinh viên thực hiện:** | 1824801030053 – Dương Lê Phước Trung  1824801030015 – Nguyễn Ngọc Minh |

*Bình Dương, tháng 04 năm 2021*

# TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

**VIỆN KỸ THUẬT – CÔNG NGHỆ**

****

**PHÁT TRIỂN ỨNG DỤNG ĐA NỀN TẢNG**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ VÀ KẾT NỐI**

**CÁC THIẾT BỊ CHO NHÀ THÔNG MINH**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Trần Văn Hữu |
| **Sinh viên thực hiện:** | 1824801030053 – Dương Lê Phước Trung  1824801030015 – Nguyễn Ngọc Minh |

*Bình Dương, tháng 04 năm 2021*

**LỜI CẢM ƠN**

Để hoàn thành tốt đề tài nghiên cứu khoa học này trước hết chúng em xin gửi đến quý thầy, cô trong viện Kỹ Thuật - Công Nghệ trường Đại Học Thủ Dầu Một lời cảm ơn chân thành.

Chúng em xin cảm ơn đến ban lãnh đạo nhà trường đã tạo mọi điều kiện tốt nhất để chúng em có cơ hội được trình bày ý tưởng của mình, từ đó áp dụng kiến thức, kỹ năng mình có để đưa ý tưởng thành sản phẩm thực tế có thể áp dụng trong đời sống.

Đặc biệt, chúng em xin gửi đến thầy Trần Văn Hữu, người đã tận tình hướng dẫn, giúp đỡ trong việc nâng cao kiến thức chuyên môn, phát huy cho chúng em tính tự học, tự tìm hiểu, từ đó hình thành cho chúng em phương pháp nghiên cứu, tìm tòi, tiếp cận với khoa học công nghệ. Đồng thời trau dồi cho chúng em kỹ năng học tập năng động và sáng tạo, giúp cho chúng em tiếp cận các ứng dụng thông minh trong thực tế một cách thiết thực hơn cũng như hỗ trợ chúng em hoàn thành đề tài nghiên cứu khoa học này lời cảm ơn sâu sắc nhất.

Vì kiến thức bản thân còn hạn chế, nên trong quá trình xây dựng đề tài nghiên cứu khoa học này chúng em không tránh khỏi những sai sót, kính mong nhận được những ý kiến đóng góp từ quý thầy cô.

Chúng em xin chân thành cảm ơn!

**MỤC LỤC**

[DANH MỤC HÌNH i](#_Toc68681606)

[DANH MỤC BẢNG ii](#_Toc68681607)

[CHƯƠNG I. GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc68681608)

[1. Tên đề tài 1](#_Toc68681609)

[2. Lý do chọn đề tài 1](#_Toc68681610)

[3. Mục tiêu của đề tài 1](#_Toc68681611)

[4. Đối tượng, phạm vi nghiên cứu, cách tiếp cận và phương pháp nghiên cứu 1](#_Toc68681612)

[4.1. Đối tượng nghiên cứu 1](#_Toc68681613)

[4.2. Phạm vi nghiên cứu 1](#_Toc68681614)

[4.3. Cách tiếp cận 1](#_Toc68681615)

[4.4. Phương pháp nghiên cứu 1](#_Toc68681616)

[CHƯƠNG II. GIẢI PHÁP CÔNG NGHỆ 3](#_Toc68681617)

[1. Công nghệ và nền tảng 3](#_Toc68681618)

[2. Tổng quan về ngôn ngữ lập trình Java và thư viện Netty 3](#_Toc68681619)

[2.1. Ngôn ngữ lập trình Java là gì? 3](#_Toc68681620)

[2.2. Ưu điểm 4](#_Toc68681621)

[2.3. Nhược điểm 4](#_Toc68681622)

[2.4. Thư viện Netty là gì? 4](#_Toc68681623)

[2.5. Ưu điểm 5](#_Toc68681624)

[2.6. Nhược điểm 5](#_Toc68681625)

[3. Tổng quan về React Native và Firebase 5](#_Toc68681626)

[3.1. React Native là gì? 5](#_Toc68681627)

[3.2. Ưu điểm 5](#_Toc68681628)

[3.3. Nhược điểm 5](#_Toc68681629)

[4. Tổng quan về Nginx 6](#_Toc68681630)

[4.1. Nginx là gì? 6](#_Toc68681631)

[4.2. Ưu điểm 6](#_Toc68681632)

[4.3. Nhược điểm 6](#_Toc68681633)

[5. Tổng quan về Node.js 6](#_Toc68681634)

[5.1. Node.js là gì? 6](#_Toc68681635)

[5.2. Ưu điểm 7](#_Toc68681636)

[5.3. Nhược điểm 7](#_Toc68681637)

[6. Tổng quan về cơ sở dữ liệu PostgreSQL 7](#_Toc68681638)

[6.1. Cơ sở dữ liệu PostgreSQL là gì? 7](#_Toc68681639)

[6.2. Ưu điểm 8](#_Toc68681640)

[6.3. Nhược điểm 8](#_Toc68681641)

[7. Tổng quan ESP32 và ESP8266 8](#_Toc68681642)

[7.1. ESP32 và ESP8266 là gì? 8](#_Toc68681643)

[CHƯƠNG III. XÁC ĐỊNH YÊU CẦU VÀ XÂY DỰNG CHƯƠNG TRÌNH 10](#_Toc68681644)

[1. Yêu cầu tổng thể 10](#_Toc68681645)

[1.1. Phía người dùng 10](#_Toc68681646)

[2. Thiết kế Usecase 10](#_Toc68681647)

[2.1. Danh sách Actor 10](#_Toc68681648)

[2.2. Biểu đồ hệ thống 10](#_Toc68681649)

[2.3. Danh sách Usecase 11](#_Toc68681650)

[2.4. Biểu đồ Usecase 12](#_Toc68681651)

[2.5. Đặc tả Usecase 12](#_Toc68681652)

[2.5.1. Use case đăng nhập 12](#_Toc68681653)

[2.5.2. Use case đăng ký 13](#_Toc68681654)

[2.5.3. Use case đăng xuất 14](#_Toc68681655)

[2.5.4. Use case quên mật khẩu 14](#_Toc68681656)

[2.5.5. Use case thông tin cá nhân người dùng 15](#_Toc68681657)

[2.5.6. Use case thay đổi mật khẩu 16](#_Toc68681658)

[2.5.7. Use case thay đổi thông tin cá nhân 16](#_Toc68681659)

[2.5.8. Use case xem thông tin bảng điều khiển 17](#_Toc68681660)

[2.5.9. Use case xem thông tin phòng 18](#_Toc68681661)

[2.5.10. Use case xem thông tin thiết bị 18](#_Toc68681662)

[2.5.11. Use case bật hoặc tắt thiết bị 19](#_Toc68681663)

[2.5.12. Use case điều chỉnh cường độ thiết bị 20](#_Toc68681664)

[2.5.13. Use case điều khiển bằng giọng nói 20](#_Toc68681665)

[2.5.14. Use case thêm thiết bị 21](#_Toc68681666)

[2.5.15. Use case sửa thiết bị 22](#_Toc68681667)

[2.5.16. Use case xoá thiết bị 23](#_Toc68681668)

[2.6. Biểu đồ tuần tự các chức năng hệ thống 24](#_Toc68681669)

[2.6.1. Use case đăng nhập 24](#_Toc68681670)

[2.6.2. Use case đăng ký 24](#_Toc68681671)

[2.6.3. Use case đăng xuất 25](#_Toc68681672)

[2.6.4. Use case quên mật khẩu 25](#_Toc68681673)

[2.6.5. Use case thông tin cá nhân 26](#_Toc68681674)

[2.6.6. Use case thay đổi mật khẩu 26](#_Toc68681675)

[2.6.7. Use case thay đổi thông tin cá nhân 27](#_Toc68681676)

[2.6.8. Use case xem thông tin bảng điều khiển 27](#_Toc68681677)

[2.6.9. Use case xem thông tin phòng 28](#_Toc68681678)

[2.6.10. Use case xem thông tin thiết bị 28](#_Toc68681679)

[2.6.11. Use case bật hoặc tắt thiết bị 28](#_Toc68681680)

[2.6.12. Use case điều chỉnh cường độ thiết bị 29](#_Toc68681681)

[2.6.13. Use case điều khiển bằng giọng nói (Trợ lý ảo) 29](#_Toc68681682)

[2.6.14. Use case thêm thiết bị 30](#_Toc68681683)

[2.6.15. Use case sửa thiết bị 30](#_Toc68681684)

[2.6.16. Use case xoá thiết bị 31](#_Toc68681685)

[3. Thiết kế cơ sở dữ liệu 32](#_Toc68681686)

[3.1. Bảng users 32](#_Toc68681687)

[3.2. Bảng forwarding\_tokens 33](#_Toc68681688)

[3.3. Bảng flashed\_tokens 33](#_Toc68681689)

[3.4. Bảng purchase 33](#_Toc68681690)

[3.5. Bảng redeem 34](#_Toc68681691)

[3.6. Bảng cloned\_projects 34](#_Toc68681692)

[CHƯƠNG IV. GIAO DIỆN ỨNG DỤNG 35](#_Toc68681693)

[1. Giao diện màn hình đăng nhập 35](#_Toc68681694)

[2. Giao diện màn hình chính 36](#_Toc68681695)

[3. Giao diện màn hình xem chi tiết phòng 37](#_Toc68681696)

[4. Giao diện màn hình trợ lý ảo 38](#_Toc68681697)

[5. Giao diện màn hình thông tin 39](#_Toc68681698)

[CHƯƠNG V. KẾT LUẬN 40](#_Toc68681699)

[1. Kết quả đạt được 40](#_Toc68681700)

[2. Hạn chế 40](#_Toc68681701)

[3. Hướng phát triển của đề tài 40](#_Toc68681702)

[CHƯƠNG VI. TÀI LIỆU THAM KHẢO 41](#_Toc68681703)

[1. Tài liệu tiếng Việt 41](#_Toc68681704)

[2. Tài liệu tiếng Anh 41](#_Toc68681705)

DANH MỤC HÌNH

[Hình 1 - Ngôn ngữ lập trình Java 3](#_Toc68679922)

[Hình 2 - Thư viện Netty 4](#_Toc68679923)

[Hình 3 - Logo Framework React Native 5](#_Toc68679924)

[Hình 4 - Logo Nginx 6](#_Toc68679925)

[Hình 5 - Logo Node.js 6](#_Toc68679926)

[Hình 6 - Logo PostgreSQL 7](#_Toc68679927)

[Hình 7 - ESP32 8](file:///F:\TRUNG\11_BaoCao_XayDungUngDungQuanLyVaKetNoiCacThietBiChoNhaThongMinh%20(Repaired).docx#_Toc68679928)

[Hình 8 - ESP8266 8](file:///F:\TRUNG\11_BaoCao_XayDungUngDungQuanLyVaKetNoiCacThietBiChoNhaThongMinh%20(Repaired).docx#_Toc68679929)

[Hình 9 - Biểu đồ hệ thống 10](#_Toc68679930)

[Hình 10 - Biểu đồ Use case của người dùng cuối 12](#_Toc68679931)

[Hình 11 - Biểu đồ tuần tự use case đăng nhập 24](#_Toc68679932)

[Hình 12 - Biểu đồ tuần tự use case đăng ký 24](#_Toc68679933)

[Hình 13 - Biểu đồ tuần tự use case đăng xuất 25](#_Toc68679934)

[Hình 14 - Biểu đồ tuần tự use case quên mật khẩu 25](#_Toc68679935)

[Hình 15 - Biểu đồ tuần tự use case thông tin cá nhân 26](#_Toc68679936)

[Hình 16 - Biểu đồ tuần tự use case thay đổi mật khẩu 26](#_Toc68679937)

[Hình 17 - Biểu đồ tuần tự use case thay đổi thông tin cá nhân 27](#_Toc68679938)

[Hình 18 - Biểu đồ tuần tự use case xem thông tin bảng điều khiển 27](#_Toc68679939)

[Hình 19 - Biểu đồ tuần tự use case xem thông tin phòng 28](#_Toc68679940)

[Hình 20 - Biểu đồ tuần tự use case xem thông tin thiết bị 28](#_Toc68679941)

[Hình 21 - Biểu đồ tuần tự use case bật hoặc tắt thiết bị 28](#_Toc68679942)

[Hình 22 - Biểu đồ tuần tự use case điều chỉnh cường độ thiết bị 29](#_Toc68679943)

[Hình 23 - Biểu đồ tuần tự use case điều khiển bằng giọng nói 29](#_Toc68679944)

[Hình 24 - Biểu đồ tuần tự use case thêm thiết bị 30](#_Toc68679945)

[Hình 25 - Biểu đồ tuần tự use case sửa thiết bị 30](#_Toc68679946)

[Hình 26 - Biểu đồ tuần tự use case xoá thiết bị 31](#_Toc68679947)

[Hình 27 - Biểu đồ lớp cở sở dữ liệu chính 31](#_Toc68679948)

[Hình 28 - Biểu đồ lớp cở sở dữ liệu ghi nhật ký 32](#_Toc68679949)

[Hình 29 - Giao diện màn hình đăng nhập 35](#_Toc68679950)

[Hình 30 - Giao diện màn hình chính 36](#_Toc68679951)

[Hình 31 - Giao diện màn xem chi tiết phòng 37](#_Toc68679952)

[Hình 32 - Giao diện màn hình xem trợ lý ảo 38](#_Toc68679953)

[Hình 33 - Giao diện mà hình thông tin 39](#_Toc68679954)

DANH MỤC BẢNG

[Bảng 1 - Danh sách Actor 10](#_Toc68679846)

[Bảng 2 - Danh sách use case 11](#_Toc68679847)

[Bảng 3 - Bảng đặc tả use case đăng nhập 12](#_Toc68679848)

[Bảng 4 - Bảng đặc tả use case đăng ký 13](#_Toc68679849)

[Bảng 5 - Bảng đặc tả use case đăng xuất 14](#_Toc68679850)

[Bảng 6 - Bảng đặc tả use case quên mật khẩu 14](#_Toc68679851)

[Bảng 7 - Bảng đặc tả use case thông tin cá nhân của người dùng 15](#_Toc68679852)

[Bảng 8 - Bảng đặc tả use case Thay đổi mật khẩu của trang người dùng 16](#_Toc68679853)

[Bảng 9 - Bảng đặc tả use case thay đổi thông tin cá nhân 16](#_Toc68679854)

[Bảng 10 - Bảng đặc tả use case xem thông tin bảng điều khiển 17](#_Toc68679855)

[Bảng 11 - Bảng đặc tả use case xem thông tin phòng 18](#_Toc68679856)

[Bảng 12 - Bảng đặc tả use case xem thông tin thiết bị 18](#_Toc68679857)

[Bảng 13 - Bảng đặc tả use case bật hoặc tắt thiết bị 19](#_Toc68679858)

[Bảng 14 - Bảng đặc tả use case điều chỉnh cường độ thiết bị 20](#_Toc68679859)

[Bảng 15 - Bảng đặc tả use case điều khiển bằng giọng nói 20](#_Toc68679860)

[Bảng 16 - Bảng đặc tả use case thêm thiết bị 21](#_Toc68679861)

[Bảng 17 - Bảng đặc tả use case sửa thiết bị 22](#_Toc68679862)

[Bảng 18 - Bảng đặc tả use case xoá thiết bị 23](#_Toc68679863)

[Bảng 19 - Bảng users 32](#_Toc68679864)

[Bảng 20 - Bảng forwarding\_tokens 33](#_Toc68679865)

[Bảng 21 - Bảng flashed\_tokens 33](#_Toc68679866)

[Bảng 22 - Bảng purchase 33](#_Toc68679867)

[Bảng 23 - Bảng redeem 34](#_Toc68679868)

[Bảng 24 - Bảng cloned\_projects 34](#_Toc68679869)

# 

GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI

Tên đề tài

“Xây dựng ứng dụng quản lý và kết nối các thiết bị cho nhà thông minh”

Lý do chọn đề tài

- Hiện tại nhà thông minh đang được phát triển rất mạnh mẽ ở Việt Nam và cũng là xu hướng của thế giới. Đặc biệt có nhiều sự phát triển của các hãng lớn liên quan như Apple, Google, Amazon, … cũng đang tích phát triển. Bình Dương đã và đang xây dựng thành phố thông minh, trong đó nhà thông minh và sự kết nối, sự đồng bộ giữa các thiết bị trong nhà là một phần không thể thiếu. Vì vậy, việc xây dựng ứng dụng giúp kết nối đến các thiết bị thông minh trong gia đình, đồng bộ và tối ưu là một vấn đề cấp thiết cần được giải quyết.

Mục tiêu của đề tài

Mục tiêu chính của để tài:

Xây dựng ứng dụng có thể kết nối nhiều thiết bị thông minh trong nhà.

- Các tính năng như: Thêm thiết bị, Xóa thiết bị, Điều khiển thiết bị, Xem thông tin từ các cảm biến, Hẹn giờ/ Lập lịch bật tắt thiết bị, …

- Xây dựng các chức năng điều khiển từ xa, ra lệnh thông minh từ ứng dụng đến các thiết bị riêng biệt.

Đối tượng, phạm vi nghiên cứu, cách tiếp cận và phương pháp nghiên cứu

Đối tượng nghiên cứu

Người có nhu cầu sử dụng nhà thông minh

Phạm vi nghiên cứu

Những người có nhu cầu sử dụng nhà thông minh ở Bình Dương.

Cách tiếp cận

- Tìm hiểu về các ứng dụng đã có hiện nay

- Tìm hiểu về các ngôn ngữ lập trình để xây dựng ứng dụng

Phương pháp nghiên cứu

* Phương pháp nghiên cứu lý thuyết:
  + Nghiên cứu các thiết bị chưa có kết nối thông minh và đã có kết nối thông minh
  + Nghiên cứu cách kết nối các thiết bị với ứng dụng
  + Nghiên cứu các thiết bị hổ trợ kết nối
  + Nghiên cứu framework React Native
  + Nghiên cứu ngôn ngữ lập trình Android
  + Nghiên cứu các lỗi bảo mật để phòng tránh (SQL Injection, XSS Attack, CSRF Attack, …)
* Phương pháp nghiên cứu thực nghiệm:
  + Xây dựng ứng dụng quản lí và kết nối các thiết bị cho Nhà thông minh dựa trên framework React Native
  + Lập trình ứng dụng bằng Java
  + Xây dựng API RESTful
  + Xây dựng ứng dụng Android từ API

# 

GIẢI PHÁP CÔNG NGHỆ

Công nghệ và nền tảng

Hệ thống nhà thông minh được xây dựng dựa trên:

- Sử dụng Java 8 và Netty để lập trình phía máy chủ

- Sử dụng React Native và Firebase để lập trình ứng dụng di động

- Sử dụng Nginx để xây dựng Reverse Proxy

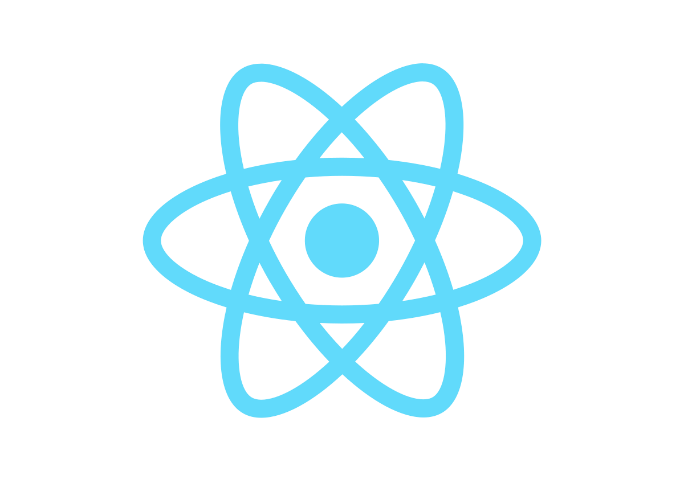
- Sử dụng Node.js và NLP.js để xây dựng mô đun xử lý ngôn ngữ tự nhiên Tiếng Việt

- Sử dụng cơ sở dữ liệu PostgreSQL để lưu trữ dữ liệu

- Thiết bị IoT được sử dụng là ESP32 và ESP8266 và các mô đun kèm theo

Tổng quan về React Native và Firebase

React Native là gì?



Hình - Logo của React.

React Native là một framework mã nguồn mở được sáng tạo bởi Facebook năm 2015. Nó được sử dụng để phát triển ứng dụng iOS, Android Web và UWP bằng các cho phép các nhà phát triển sử dụng React cùng với môi trường ứng dụng gốc (native).

Các nguyên tắc hoạt động của React Native gần như giống hệt với React ngoại trừ việc React Native không thao tác với DOM thông qua DOM ảo. Nó chạy một quá trình xử lý nền (phiên dịch JavaScript viết bởi các nhà phát triển) trực tiếp trên thiết bị đầu cuối và giao tiếp với nền tảng gốc qua một cầu trung gian, bất đông bộ và theo đợt. Các thành phần React bao bọc mã gốc và tương tác với API gốc qua mô hình UI khai báo và Javascript của React. Điều này giúp việc phát triển ứng dụng cho nhiều nền tảng nhanh hơn. React Native không sử dụng HTML. Thay vào đó, nó sử dụng các thành phần khác từ luồng Javascript.

React Native phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động. Chúng ta sẽ build được ứng dụng Native, và chúng ta cũng có thể build ứng dụng đó một cách đa nền tảng (multi-platform) chứ không phải là một “mobile web app”, không phải là “HTML5 app”, và cũng không phải là một “hybrid app” hay cũng không chỉ build trên iOS hay Android mà chúng ta build và chạy được cả hai hệ sinh thái luôn, sợ chưa!!! Một điểm hay ho nữa mà mình có đề cập là giảm chi phí recompile của Native bằng cách sử dụng Hot-Loading tức là bạn không cần phải build lại ứng dụng từ đầu nên việc chỉnh sửa diễn ra rất nhanh chóng. Giúp cho lập trình viên có thể thấy được những chỉnh sửa của họ một cách nhanh chóng trực quan, không còn phải bỏ quá nhiều thời gian trong việc build và run ứng dụng nữa.

Ưu điểm

##### Hiệu quả về mặt thời gian phát triển ứng dụng được rút ngắn.

##### Tiết kiệm chi phí nhân lực cho doanh nghiệp

##### Hiệu năng tương đối ổn định.

##### Cộng đồng phát triển mạnh.

##### Khả năng tái sử dụng code cao (75% – 80%)

##### Trải nghiệm người dùng rất tốt

##### Ứng dụng tin cậy và ổn định.

##### Xây dựng cho nhiều hệ điều hành khác nhau với ít native code nhất.

##### Trải nghiệm người dùng tốt hơn là hybrid app.

Nhược điểm

##### Vẫn đòi hỏi native code.

##### Hiệu năng sẽ thấp hơn với app thuần native code.

##### Chỉ hỗ trợ 2 nền tảng Android và iOS

##### Bảo mật không cao do dựa trên JS.

##### Quản lý bộ nhớ.

##### Bảo mật không cao

##### Khả năng tùy biến cũng không thực sự tốt đối với một vài module4

Google Service: Google Firebase Cloud Messaging



Hình - Google Firebase

Google Firebase là gì

##### - Firebase là platform do Google cung cấp, nhằm hỗ trợ việc tạo ra các web application, mobile application với chất lượng cao.

##### - Với việc sử dụng Firebase, developer có thể tập trung vào việc phát triển application mà không cần lo về việc sản phẩm của mình sẽ hoạt động và được quản lý thể nào ở phía Backend.

##### - Khái niệm Backend nhắc tới trong câu trên để chỉ việc xử lý và lưu trữ dữ liệu trong service, ở vùng "phía sau" mà người dùng không nhìn thấy.

##### - Firebase là một trong những BaaS (Backend as a service).

##### - Serverless architecture được chú ý như một giải pháp đáp ứng yêu cầu release một dịch vụ trong thời gian ngắn, trong bối cảnh đó, BaaS ra mắt người dùng. BaaS hướng đến thiết bị chủ yếu là mobile nên còn được gọi với một tên gọi khác là MBaaS.

3.2. Firebase Cloud Messaging

3.2.1. Khái niệm



Hình - Firebase Cloud Messaging

Firebase Cloud Messaging (FCM), trước đây gọi là Google Cloud Messaging (GCM), là một giải pháp đám mây đa nền tảng dành cho tin nhắn và thông báo cho các ứng dụng web Android, iOS và kể từ năm 2021 có thể được sử dụng miễn phí. Nhắn tin qua đám mây Firebase cho phép các nhà phát triển ứng dụng bên thứ ba gửi thông báo hoặc tin nhắn từ máy chủ do GCM lưu trữ tới người dùng nền tảng hoặc người dùng cuối. Dịch vụ được cung cấp bởi Firebase, một công ty con của Google. Vào ngày 21 tháng 10 năm 2014, Firebase thông báo rằng họ đã được Google mua lại với số tiền không được tiết lộ. Trang web chính thức của Google Cloud Messaging chỉ ra Firebase Cloud Messaging (FCM) là phiên bản mới của GCM. Firebase là một nền tảng di động hỗ trợ người dùng phát triển các ứng dụng web và di động. Nhắn tin qua đám mây của Firebase là một trong nhiều sản phẩm thuộc nền tảng Firebase. Trên nền tảng này, người dùng có thể tích hợp và kết hợp các tính năng Firebase khác nhau trong cả ứng dụng web và ứng dụng di động.

##### Firebase Cloud Messaging (FCM) là một phần của nền tảng Firebase, là một mô hình dịch vụ đám mây tự động phát triển phần phụ trợ hoặc Dịch vụ phụ trợ (BaaS). Sau khi công ty Firebase được Google mua lại vào năm 2014, một số sản phẩm hoặc công nghệ nền tảng Firebase đã được tích hợp với các dịch vụ hiện có của Google. Dịch vụ thông báo dành cho thiết bị di động của Google là Google Cloud Messaging (GCM) đã được thay thế bằng FCM vào năm 2016. Vào ngày 10 tháng 4 năm 2018, Google đã xóa GCM và vào ngày 29 tháng 5 năm 2019, API ứng dụng và máy chủ GCM không còn được dùng nữa. FCM đã trở thành sự thay thế cho GCM. Tuy nhiên, FCM tương thích với Bộ phát triển phần mềm (SDK) hiện có của Google.

Tổng quan về ngôn ngữ lập trình Java và thư viện Netty

Ngôn ngữ lập trình Java là gì?



Hình - Ngôn ngữ lập trình Java

Java là một một ngôn ngữ lập trình hiện đại, bậc cao, hướng đối tượng, bảo mật và mạnh mẽ. và là một Platform.

Platform: Bất cứ môi trường phần cứng hoặc phần mềm nào mà trong đó có một chương trình chạy, thì được hiểu như là một Platform. Với môi trường runtime riêng cho mình (JRE) và API, Java được gọi là Platform.

Ngôn ngữ lập trình Java ban đầu được phát triển bởi Sun Microsystems do James Gosling khởi xướng và phát hành vào năm 1995. Phiên bản mới nhất của Java Standard Edition là Java SE 8. Với sự tiến bộ của Java và sự phổ biến rộng rãi của nó, nhiều cấu hình đã được xây dựng để phù hợp với nhiều loại nền tảng khác nhau. Ví dụ: J2EE cho các ứng dụng doanh nghiệp, J2ME cho các ứng dụng di động.

Các phiên bản J2 mới đã được đổi tên thành Java SE, Java EE và Java ME. Phương châm của java là "Write Once, Run Anywhere" - viết một lần chạy nhiều nơi, nghĩa là bạn chỉ cần viết một lần trên window chẳng hạn, sau đó vẫn chương trình đó bạn có thể chạy trên Linux, Android, các thiết bị J2ME…

Ưu điểm

- Hướng đối tượng: Trong Java, mọi thứ đều là một Object. Java có thể dễ dàng mở rộng và bảo trì vì nó được xây dựng dựa trên mô hình Object.

- Nền tảng độc lập: Không giống nhiều ngôn ngữ lập trình khác bao gồm cả C và C ++, khi Java được biên dịch, nó không được biên dịch thành ngôn ngữ máy nền tảng cụ thể, thay vào mã byte - nền tảng độc lập. Mã byte này được thông dịch bởi máy ảo (JVM) trên nền tảng nào đó mà nó đang chạy.

- Hiệu năng cao: Với việc sử dụng trình biên dịch Just-In-Time, Java cho phép thực hiện hiệu năng cao.

- Bảo mật: Với tính năng an toàn của Java, nó cho phép phát triển các hệ thống không có virus, giả mạo. Các kỹ thuật xác thực dựa trên mã hoá khóa công khai.

Nhược điểm

- Trình biên dịch Java chưa được tối ưu hóa tốt so với C ++.

- Không có sự tách biệt đặc điểm kỹ thuật khi triển khai.

- Quản lý bộ nhớ, với Java, là khá tốn kém.

- Việc thiếu các template có thể hạn chế khả năng của Java để tạo ra các cấu trúc dữ liệu chất lượng cao.

Thư viện Netty là gì?



Hình - Thư viện Netty

Netty là 1 framwork client/server với việc hỗ trợ mạnh mẽ về Java network ẩn đi mức độ phức tạp như nguyên bản của nó và cung cấp các APIs dễ sử dụng. Netty gửi thông tin với mức high performance và scalability, nó giúp chúng ta chỉ cần tập trung vào nghiệp vụ trong ứng dụng.

Thời điểm trước vấn đề phát triển về network, nó thực hiện mất rất nhiều thời gian để học sockets, addressing, … coding với thư viện socket sử dụng ngôn ngữ C, và cần thực hiện các cách sử lý khác nhau đối với mỗi hệ điều hành.

Version đầu tiên của Java (1995-2002) giới thiệu đầy đủ về object-oriented ẩn đi độ phức tạp, nhưng để thực hiện client-server protocols hoàn chỉnh thì cần phải soạn thảo rất nhiều mã lệnh (và rất nhiều monitoring để đảm bảo sự chắc chắn trong hoạt động).

Ra đời sớm nhất phải nói đến Java APIs (java.net) chỉ cung cấp gọi các chức năng “blocking” thông qua thư viện native socket. 1 ví dụ đơn giản

Ưu điểm

- Netty gửi thông tin với mức high performance và scalability

- Netty trong việc giải quyết các vấn đề về high-concurrency network

Nhược điểm

- Là một thư viện bên ngoài nên phải cài vào bằng tay hoặc các phần mềm như Maven, Gradle

Tổng quan về Nginx

Nginx là gì?



Hình - Logo Nginx

NGINX, đọc là “engine-ex” là một phần mềm web server mã nguồn mở nỗi tiếng. Ban đầu nó dùng để phục vụ web HTTP. Tuy nhiên, ngày nay nó cũng được dùng làm reverse proxy, HTTP load balancer và email proxy như IMAP, POP3, và SMTP.

NGINX xuất bản chính thức vào tháng 10 năm 2004. Nhà sáng lập của phần mềm này là Igor Sysoev, triển khai dự án từ năm 2002 để giải quyết vấn đề C10k. C10k là giới hạn của việc xử lý 10 ngàn kết nối cùng lúc. Ngày nay, có nhiều web server còn phải chịu nhiều kết nối hơn vậy để xử lý. NGINX sử dụng kiến trúc hướng sự kiện (event-driven) không đồng bộ (asynchronous). Tính năng này khiến NGINX server trở nên đáng tin cậy, tốc độ và khả năng mở rộng lớn nhất.

Ưu điểm

- Hỗ trợ hầu hết hệ điều hành cho máy chủ

- Khả năng bảo mật dữ liệu tuyệt vời

- Tài liệu đầy đủ và cộng đồng hỗ trợ rộng lớn

- Linh hoạt, nó không chỉ là Web Server mà còn là Reverse Proxy, …

Nhược điểm

- Tương đối khó cấu hình với người mới bắt đầu

- Bản thương mại cần có bản quyền để có nhiều chức năng hơn

Tổng quan về Node.js

Node.js là gì?



Hình - Logo Node.js

NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux.

Ưu điểm

Đầu tiên là ưu điểm về tốc độ thực thi và khả năng mở rộng. Node.js có tốc độ rất nhanh. Đó là một yêu cầu khá quan trọng khi bạn là một startup đang cố gắng tạo ra một sản phẩm lớn và muốn đảm bảo có thể mở rộng nhanh chóng, đáp ứng được một lượng lớn người dùng khi trang web của bạn phát triển lên.

Node.js có thể xử lý hàng ngàn kết nối đồng thời. Nodejs sử dụng cơ chế bất đồng bộ để xử lý, nên có thể xử lý cực kỳ nhanh chóng, mang lại một hiệu suất đáng kinh ngạc.

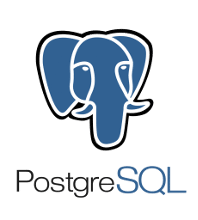
Nhược điểm

- Tương đối khan hiếm hosting để chạy được Node.js

- Thay đổi liên tục nên việc tương thích giữa các thư viện rất khó, các thư viện sẽ bị lỗi thời nhanh chóng nếu không được cập nhật liên tục.

Tổng quan về cơ sở dữ liệu PostgreSQL

Cơ sở dữ liệu PostgreSQL là gì?



Hình - Logo PostgreSQL

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng (object-relational database management system) có mục đích chung, hệ thống cơ sở dữ liệu mã nguồn mở tiên tiến nhất hiện nay.

PostgreSQL được phát triển dựa trên POSTGRES 4.2 tại phòng khoa học máy tính Berkeley, Đại học California.

PostgreSQL được thiết kế để chạy trên các nền tảng tương tự UNIX. Tuy nhiên, PostgreSQL sau đó cũng được điều chỉnh linh động để có thể chạy được trên nhiều nền tảng khác nhau như Mac OS X, Solaris và Windows.

PostgreSQL là một phần mềm mã nguồn mở miễn phí. Mã nguồn của phần mềm khả dụng theo license của PostgreSQL, một license nguồn mở tự do. Theo đó, bạn sẽ được tự do sử dụng, sửa đổi và phân phối PostgreSQL dưới mọi hình thức.

PostgreSQL không yêu cầu quá nhiều công tác bảo trì bởi có tính ổn định cao. Do đó, nếu bạn phát triển các ứng dụng dựa trên PostgreSQL, chi phí sở hữu sẽ thấp hơn so với các hệ thống quản trị dữ liệu khác.

Ưu điểm

PostgreSQL là hệ thống quản lý cơ sở dữ liệu đầu tiên triển khai tính năng kiểm soát đồng thời nhiều phiên bản (MVCC) trước cả Oracle. Tính năng MVCC cũng tương tự với các snapshot riêng biệt trong Oracle.

Là hệ thống quản lý cơ sở dữ liệu quan hệ đối tượng, PostgreSQL cho phép thêm vào các tính năng tùy chỉnh được phát triển bằng các ngôn ngữ chương trình khác nhau như C/C, Java, …

Bên cạnh đó, khi bạn có bất kỳ vấn đề gì cần hỗ trợ, luôn có một cộng đồng nhiệt tình sẵn sàng để trợ giúp. Bạn luôn có thể tìm thấy câu trả lời từ cộng đồng của PostgreSQL về các vấn đề có thể gặp phải khi làm việc với PostgreSQL. Ngoài ra, cũng có rất nhiều công ty cung cấp các dịch vụ hỗ trợ trong trường hợp bạn thực sự cần đến.

Nhược điểm

PostgreSQL không thuộc sở hữu của một tổ chức. Vì vậy, nó đã gặp khó khăn khi đưa tên của mình ra khỏi đó mặc dù có đầy đủ tính năng và có thể so sánh với các hệ thống DBMS khác.

Những thay đổi được thực hiện để cải thiện tốc độ đòi hỏi nhiều công việc hơn MySQL vì PostgreSQL tập trung vào khả năng tương thích.

Nhiều ứng dụng nguồn mở hỗ trợ MySQL, nhưng có thể không hỗ trợ PostgreSQL

Về số liệu hiệu suất, nó chậm hơn MySQL.

Tổng quan ESP32 và ESP8266

ESP32 và ESP8266 là gì?

Hình 7 - ESP32

ESP8266 là một vi mạch Wi-Fi giá rẻ, có hỗ trợ bộ giao thức TCP/IP và có thể tích hợp vào thành phần của vi điều khiển, được sản xuất bởi hãng Espressif Systems ở Thượng Hải, Trung Quốc.

Hình 8 - ESP8266

Chip ESP8226 lần đầu tiên được các nhà sản xuất phương Tây chú ý vào tháng 8 năm 2014 với module ESP-01, do nhà sản xuất bên thứ ba là Ai-Thinker sản xuất. Module này cho phép các vi điều khiển kết nối với mạng Wi-Fi và thực hiện các kết nối TCP/IP đơn giản bằng cách sử dụng các lệnh kiểu Hayes (tập lệnh AT). Tuy nhiên, ban đầu hầu như không có tài liệu tiếng Anh nào về chip và các tập lệnh của nó. Vì mức giá rất thấp với rất ít thành phần bên ngoài khác trên module, các module ESP8266 đã thu hút nhiều hacker khám phá nó và các phần mềm trên đó, cũng như việc dịch thuật các tài liệu tiếng Trung Quốc của chip.

ESP8285 là một chip ESP8266 với 1 MiB bộ nhớ flash được tích hợp, cho phép người dùng có thể sản xuất các thiết bị có khả năng kết nối với Wi-Fi chỉ với một chip đơn.

Dòng chip kế thừa ESP8266 là ESP32, bao gồm cả vi điều khiển ESP32-C3 tương thích chân.

XÁC ĐỊNH YÊU CẦU VÀ XÂY DỰNG CHƯƠNG TRÌNH

Yêu cầu tổng thể

Phía người dùng

- Màn hình đăng nhập

- Màn hình đăng ký

- Màn hình bảng điều khiển

- Màn hình điều khiển thiết bị

- Màn hình trợ lý ảo

- Màn hình chỉnh sửa thiết bị

- Màn hình thông tin cá nhân

- Màn hình cập nhật thông tin cá nhân

Thiết kế Usecase

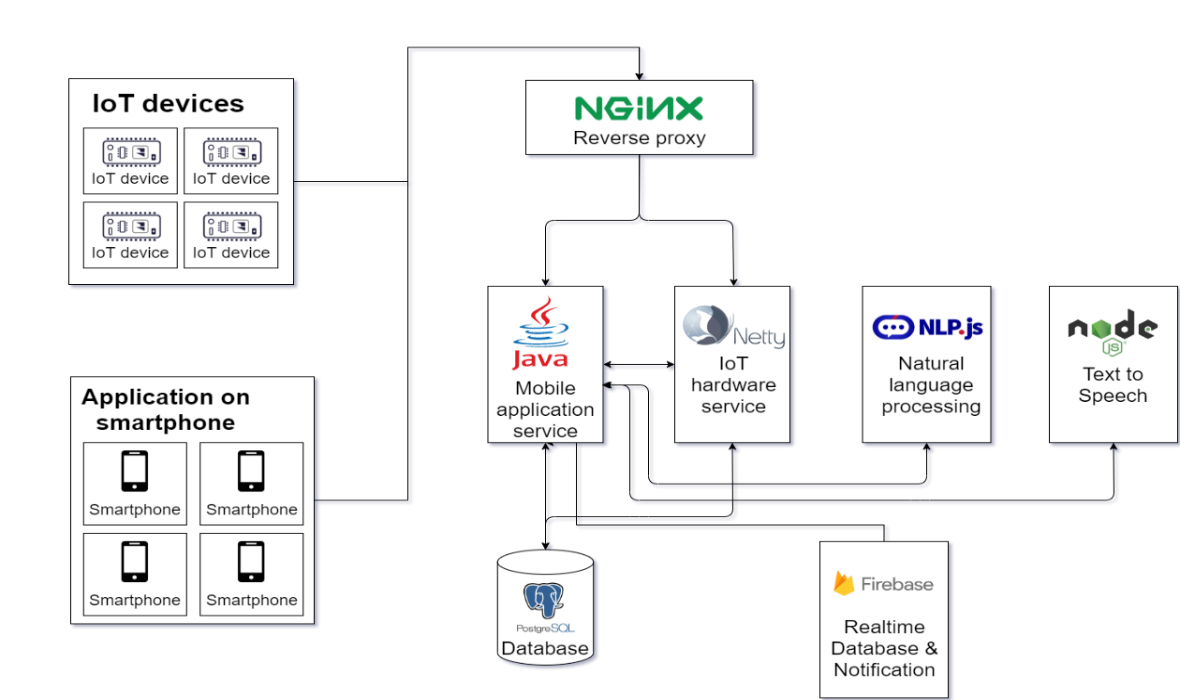
Danh sách Actor

Một actor hay tác nhân ngoài là một vai trò của một hay nhiều người hay vật thể trong sự tương tác với hệ thống.

Bảng - Danh sách Actor

|  |  |  |
| --- | --- | --- |
| **STT** | **Tác nhân** | **Mô tả** |
| 1 | Người dùng cuối | Là người sử dụng hệ thống, có quyền truy cập vào hệ thống |

Biểu đồ hệ thống



Hình - Biểu đồ hệ thống

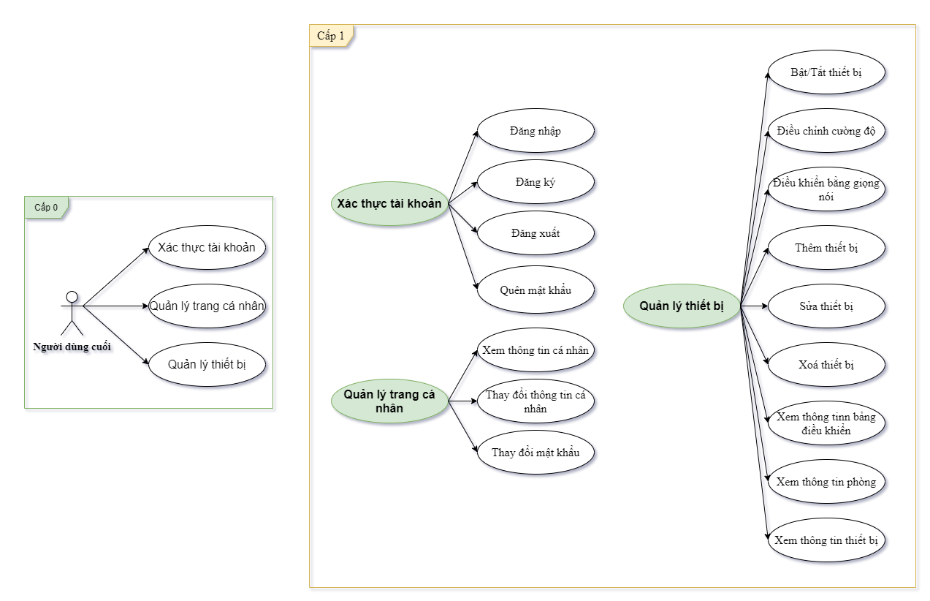
Danh sách Usecase

Bảng - Danh sách use case

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Actor** | **Use case** | **Ý nghĩa** |
| 1 | Người dùng cuối | Đăng nhập | Người dùng đăng nhậo vào ứng dụng |
| 2 | Đăng ký | Người dùng đăng ký tài khoản |
| 3 | Đăng xuất | Người dùng đăng xuất khỏi ứng dụng |
| 4 | Quên mật khẩu | Người dùng quên mật khẩu tài khoản |
| 5 | Bật/Tắt thiết bị | Người dùng bật hoặc tắt thiết bị |
| 6 | Điều chỉnh cường độ | Người dùng điều chỉnh cường độ thiết bị |
| 7 | Điều khiển bằng giọng nói | Người dùng ra lệnh bằng giọng nói |
| 8 | Thêm thiết bị | Người dùng thêm một thiết bị mới |
| 9 | Sửa thiết bị | Người dùng sửa thông tin thiết bị |
| 10 | Xoá thiết bị | Người dùng xoá thiết bị đã thêm |
| 11 | Thông tin cá nhân người dùng | Người dùng xem thông tin cá nhân của mình trên ứng dụng |
| 12 | Thay đổi mật khẩu | Người dùng thay đổi mật khẩu cá nhân |
| 13 | Thay đổi thông tin cá nhân | Người dùng thay đổi thông tin cá nhân |
| 14 | Xem thông tin bảng điều khiển | Người dùng xem thông tin bảng điều khiển |
| 15 | Xem thông tin phòng | Người dùng xem thông tin phòng |
| 16 | Xem thông tin thiết bị | Người dùng xem thông tin thiết bị |

### 

Biểu đồ Usecase



Hình - Biểu đồ Use case của người dùng cuối

Đặc tả Usecase

Use case đăng nhập

Bảng - Bảng đặc tả use case đăng nhập

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case mô tả cách người dùng đăng nhập vào hệ thống |
| 2 | Actor | Người dùng cuối |
| 3 | Dòng sự kiện | **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng cuối muốn đăng nhập vào ứng dụng.  + Người dùng cuối nhập tên đăng nhập và mật khẩu.  + Hệ thống kiểm tra thông tin đăng nhập, nếu tài khoản và mật khẩu nhập chính xác thì qua màn hình điều khiển.  **Dòng sự kiện khác:** Nếu trong dòng sự kiện chính, người dùng nhập sai tên đăng nhập, mật khẩu thì hệ thống sẽ hiển thị thông báo lỗi. Người dùng có thể chọn trở về đầu dòng sự kiện chính hoặc hủy bỏ việc đăng nhập, lúc này use case kết thúc. |
| 4 | Các yêu cầu đặc biệt | Không có. |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đứng ở màn hình đăng nhập, các chức năng không được hiển thị. |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công actor có thể sử dụng các chức năng của màn hình bảng điều khiển |
| 7 | Điểm mở rộng | Không có |

Use case đăng ký

Bảng - Bảng đặc tả use case đăng ký

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case mô tả cách người dùng đăng ký tài khoản |
| 2 | Actor | Người dùng cuối |
| 3 | Dòng sự kiện | **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng cuối muốn đăng ký tài khoản  + Người dùng cuối nhập đầy đủ các thông tin cần thiết trong biểu mẫu đăng ký tài khoản  + Hệ thống kiểm tra thông tin đăng ký, nếu thông tin hợp lệ thì chuyển qua màn hình điều khiển.  **Dòng sự kiện khác:** Nếu trong dòng sự kiện chính, người dùng nhập sai thông tin thì hệ thống sẽ hiển thị thông báo lỗi. Người dùng có thể chọn trở về đầu dòng sự kiện chính hoặc hủy bỏ việc đăng ký, lúc này use case kết thúc. |
| 4 | Các yêu cầu đặc biệt | Không có. |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đứng ở màn hình đăng nhập, các chức năng không được hiển thị. |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công actor có thể sử dụng các chức năng của màn hình bảng điều khiển |
| 7 | Điểm mở rộng | Không có |

Use case đăng xuất

Bảng - Bảng đặc tả use case đăng xuất

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng đăng xuất khỏi ứng dụng |
| 2 | Actor | Người dùng cuối |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn đăng xuất khỏi ứng dụng   + Người dùng chọn nào nút đăng xuất trên ứng dụng.  + Hệ thống hủy bỏ phiên đăng nhập của tài khoản và quay trở về màn hình đăng nhập của trang người dùng.  - **Dòng sự kiện khác:** Không có. |
| 4 | Các yêu cầu đặc biệt | Không có. |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở giao diện của màn hình thông tin người dùng |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công actor sẽ quay trở về màn hình đăng nhập của người dùng. |
| 7 | Điểm mở rộng | Không có |

Use case quên mật khẩu

Bảng - Bảng đặc tả use case quên mật khẩu

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng khôi phục lại mật khẩu |
| 2 | Actor | Người dùng cuối |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn khôi phục mật khẩu   + Người dùng chọn nút quên mật khẩu trên màn hình đăng nhập ứng dụng.  + Người dùng điền thông tin vào biểu mẫu quên mật khẩu.  + Hệ thống kiểm tra thông tin, và gửi lại mật khẩu mới vào địa chỉ email của người dùng  - **Dòng sự kiện khác:** Nếu người dùng nhập thông tin không chính xác thì hệ thống sẽ trả về thông báo lỗi. |
| 4 | Các yêu cầu đặc biệt | Không có. |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở giao diện của màn hình thông tin người dùng |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công actor sẽ quay trở về màn hình đăng nhập của người dùng. |
| 7 | Điểm mở rộng | Không có |

Use case thông tin cá nhân người dùng

Bảng - Bảng đặc tả use case thông tin cá nhân của người dùng

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng xem thông tin cá nhân của mình |
| 2 | Actor | Ngươi dùng |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn xem chi tiết thông tin cá nhân của tài khoản.   + Người ấn vào nút thông tin cá nhân  + Hệ thống kiểm tra và hiển thị thông tin cá nhân của người dùng.  - **Dòng sự kiện khác:** Không có |
| 4 | Các yêu cầu đặc biệt | Không có |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở giao diện của màn hình người dùng |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công actor sẽ chuyển qua màn hình thông tin tài khoản |
| 7 | Điểm mở rộng | Không có |

Use case thay đổi mật khẩu

Bảng - Bảng đặc tả use case Thay đổi mật khẩu của trang người dùng

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng thay đổi mật khẩu tài khoản của mình |
| 2 | Actor | Người dùng |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn thay đổi mật khẩu cá nhân.   + Người dùng ấn và nút thay đổi mật khẩu ở mà hình thông tin người dùng.  + Sau khi nhập đầy đủ thông tin và người dùng chọn cập nhật.  + Hệ thống kiểm tra và lưu thông tin vào cơ sở dữ liệu rồi thông báo với người dùng.  - **Dòng sự kiện khác:** Nếu trong quá trình thực thi xảy ra lỗi thì hệ thống sẽ trỡ về giao diện người dùng và thông báo với người dùng là có lỗi xảy ra. |
| 4 | Các yêu cầu đặc biệt | Mật khẩu cũ phải đúng Giá trị trong trường “Nhập lại mật khẩu mới” phải giống với giá trị trong trường “Nhật khẩu mới”. |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở giao diện của trang thay đổi mật khẩu |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công actor sẽ quay trở về trang Người dùng và thông báo cập nhật thành công. |
| 7 | Điểm mở rộng | Không có |

Use case thay đổi thông tin cá nhân

Bảng - Bảng đặc tả use case thay đổi thông tin cá nhân

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng sửa thông tin cá nhân |
| 2 | Actor | Người dùng |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn thay đổi thông tin cá nhân.   + Người dùng chọn Thay đổi thông tin cá nhân  + Hệ thống hiễn thị biểu mẫu thông tin cá nhân cho người dùng.  + Sau khi người dùng sửa xong thì ấn và nút cập nhập.  + Hệ thống kiểm tra và lưu thông tin vào cơ sở dữ liệu rồi thông báo với người dùng.  - **Dòng sự kiện khác:** Nếu trong quá trình thực thi xảy ra lỗi thì hệ thống sẽ trỡ về giao diện chi tiết tài khoản và thông báo với người dùng là có lỗi xảy ra. |
| 4 | Các yêu cầu đặc biệt | Một số thông tin nhập vào bắt buộc phải có. |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở giao diện của trang người dùng |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công actor sẽ quay trở về trang người dùng và thông báo cập nhật thành công. |
| 7 | Điểm mở rộng | Không có |

Use case xem thông tin bảng điều khiển

Bảng - Bảng đặc tả use case xem thông tin bảng điều khiển

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng xem thông tin bảng điều khiển |
| 2 | Actor | Người dùng |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn xem thông tin bảng điều khiển   + Người dùng chọn tab bảng điều khiển (hoặc mặc định màn hình người dùng sau khi đăng nhập).  + Hệ thống lấy kết quả từ trong cơ sở dữ liệu và hiển thị lên màn hình.  - **Dòng sự kiện khác:** Không có. |
| 4 | Các yêu cầu đặc biệt | Không có. |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở trong giao diện của màn hình người dùng. |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công kết quả sẽ hiển thị trực tiếp trên màn hình bảng điều khiển |
| 7 | Điểm mở rộng | Không có |

Use case xem thông tin phòng

Bảng - Bảng đặc tả use case xem thông tin phòng

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng xem thông tin chi tiết của một phòng nhất định |
| 2 | Actor | Người dùng |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn xem thông tin của một phòng nhất định   + Người dùng chọn tab bảng điều khiển (hoặc mặc định màn hình người dùng sau khi đăng nhập).  + Người dùng chọn phòng cần xem  + Hệ thống lấy kết quả từ trong cơ sở dữ liệu và hiển thị lên màn hình.  - **Dòng sự kiện khác:** Không có. |
|  | Các yêu cầu đặc biệt | Không có. |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở trong giao diện của màn hình người dùng. |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công kết quả sẽ hiển thị trực tiếp trên màn hình bảng điều khiển |
| 7 | Điểm mở rộng | Không có |

Use case xem thông tin thiết bị

Bảng - Bảng đặc tả use case xem thông tin thiết bị

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng xem thông tin một thiết bị nhất định |
| 2 | Actor | Người dùng |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn xem thông tin chi tiết của một thiết bị nhất định   + Người dùng chọn thiết bị muốn xem từ danh sách phòng  + Hệ thống lấy kết quả từ trong cơ sở dữ liệu và hiển thị lên màn hình.  - **Dòng sự kiện khác:** Không có. |
| 4 | Các yêu cầu đặc biệt | Không có. |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở trong giao diện của màn hình người dùng. |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công kết quả sẽ hiển thị trực tiếp trên màn hình xem thông tin chi tiết thiết bị |
| 7 | Điểm mở rộng | Không có |

Use case bật hoặc tắt thiết bị

Bảng - Bảng đặc tả use case bật hoặc tắt thiết bị

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng bật hoặc tắt thiết bị |
| 2 | Actor | Người dùng |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn bật tắt thiết bị   + Người dùng chọn vào nút Bật/Tắt ở màn hình thiết bị  + Ứng dụng gửi yêu cầu lên hệ thống  + Hệ thống kiểm tra và lưu thông tin vào cơ sở dữ liệu rồi thông báo với người dùng.  + Hệ thống gửi yêu cầu xuống phần cứng  - **Dòng sự kiện khác:** Không có |
| 4 | Các yêu cầu đặc biệt | Không có |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở giao diện của màn hình thiết bị |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công thiết bị sẽ được bật hoặc tắt |
| 7 | Điểm mở rộng | Không có |

Use case điều chỉnh cường độ thiết bị

Bảng - Bảng đặc tả use case điều chỉnh cường độ thiết bị

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng điều chỉnh cường độ thiết bị (VD: độ sáng đèn, tốc độ quạt, nhiệt độ máy lạnh, …) |
| 2 | Actor | Người dùng |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn điều chỉnh cường độ thiết bị   + Người dùng trượt thanh trượt ở màn hình thiết bị sao cho phù hợp  + Ứng dụng gửi yêu cầu lên hệ thống  + Hệ thống kiểm tra và lưu thông tin vào cơ sở dữ liệu rồi thông báo với người dùng.  + Hệ thống gửi yêu cầu xuống phần cứng  - **Dòng sự kiện khác:** Không có |
| 4 | Các yêu cầu đặc biệt | Không có |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở giao diện của màn hình thiết bị |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công thiết bị sẽ được điều chỉnh cường độ giống như trên ứng dụng |
| 7 | Điểm mở rộng | Không có |

Use case điều khiển bằng giọng nói

Bảng - Bảng đặc tả use case điều khiển bằng giọng nói

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng điều khiển các thiết bị bằng giọng nói Tiếng Việt |
| 2 | Actor | Người dùng |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn giao tiếp với ứng dụng bằng giọng nói Tiếng Việt   + Người dùng chọn vào biểu tượng trợ lý ảo tại màn hình chính  + Trợ lý ảo xuất hiện và người dùng giao tiếp với trợ lý ảo  + Đoạn ghi âm giọng nói sẽ được gửi lên máy chủ xử lý  + Máy chủ xử lý sẽ trả về văn bản của đoạn ghi âm  + Văn bản sau đó sẽ được xử lý bằng máy chủ xử lý ngôn ngữ tự nhiên và trả về kết quả là câu trả lời bằng văn bản  + Đồng thời sẽ gửi lệnh điều khiển sang thiết bị IoT (lệnh bật/tắt, điều chỉnh cường độ, …) nếu có  + Đoạn văn bản trả lời sẽ được gửi sang máy chủ chuyển văn bản thành giọng nói  + Tập tin âm thanh đã được chuyển đổi sẽ được gửi ngược trở về ứng dụng  + Ứng dụng phát tập tin âm thanh nhận được  - **Dòng sự kiện khác:** Không có |
| 4 | Các yêu cầu đặc biệt | Không có |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở giao diện của màn hình thiết bị |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công ứng dụng sẽ phát âm thanh giống như một trợ lý ảo giao tiếp, đồng thời thiết bị cũng được điều khiển theo yêu cầu |
| 7 | Điểm mở rộng | Không có |

Use case thêm thiết bị

Bảng - Bảng đặc tả use case thêm thiết bị

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng thêm thiết bị mới vào hệ thống |
| 2 | Actor | Người dùng |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn thêm thiết bị   + Người dùng chọn biểu tượng dấu cộng tại màn hình xem phòng  + Giao diện thêm thiết bị hiện ra và người dùng điền đầy đủ thông tin và chọn thêm  + Hệ thống kiểm tra và lưu thông tin vào cơ sở dữ liệu rồi thông báo với người dùng.  + Hệ thống gửi yêu cầu xuống phần cứng  - **Dòng sự kiện khác:** Nếu người dùng nhập thông tin không chính xác, thì yêu cầu nhập lại và thông báo lỗi. |
| 4 | Các yêu cầu đặc biệt | Không có |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở giao diện của màn hình thiết bị |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công thiết bị sẽ được thêm vào hệ thống |
| 7 | Điểm mở rộng | Không có |

Use case sửa thiết bị

Bảng - Bảng đặc tả use case sửa thiết bị

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng sửa thiết bị ở trong hệ thống |
| 2 | Actor | Người dùng |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn sửa thiết bị   + Người dùng chọn biểu tượng bánh răng tại màn hình xem phòng  + Giao diện thêm thiết bị hiện ra và người dùng điền đầy đủ thông tin và chọn sửa  + Hệ thống kiểm tra và lưu thông tin vào cơ sở dữ liệu rồi thông báo với người dùng.  + Hệ thống gửi yêu cầu xuống phần cứng  - **Dòng sự kiện khác:** Nếu người dùng nhập thông tin không chính xác, thì yêu cầu nhập lại và thông báo lỗi. |
| 4 | Các yêu cầu đặc biệt | Không có |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở giao diện của màn hình thiết bị |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công thiết bị sẽ được sửa ở trong hệ thống |
| 7 | Điểm mở rộng | Không có |

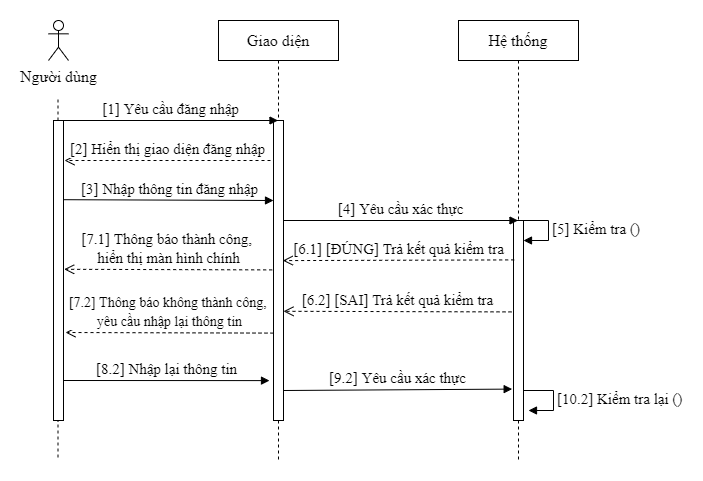
Use case xoá thiết bị

Bảng - Bảng đặc tả use case xoá thiết bị

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Nội dung** |
| 1 | Tóm tắt | Use case này mô tả cách người dùng xoá thiết bị trong hệ thống |
| 2 | Actor | Người dùng |
| 3 | Dòng sự kiện | * **Dòng sự kiện chính:** Use case này bắt đầu khi người dùng muốn xoá thiết bị   + Người dùng chọn biểu tượng thùng rác tại màn hình xem phòng  + Giao diện hỏi lại người dùng lần cuối có chắc chắn muốn xoá hay không  + Hệ thống kiểm tra và lưu thông tin vào cơ sở dữ liệu rồi thông báo với người dùng.  + Hệ thống gửi yêu cầu xuống phần cứng  - **Dòng sự kiện khác:** Không có |
| 4 | Các yêu cầu đặc biệt | Không có |
| 5 | Trạng thái hệ thống trước khi thực hiện use | Đang đứng ở giao diện của màn hình thiết bị |
| 6 | Trạng thái hệ thống sau khi thực hiện use | Nếu use case thành công thiết bị xoá khỏi hệ thống |
| 7 | Điểm mở rộng | Không có |

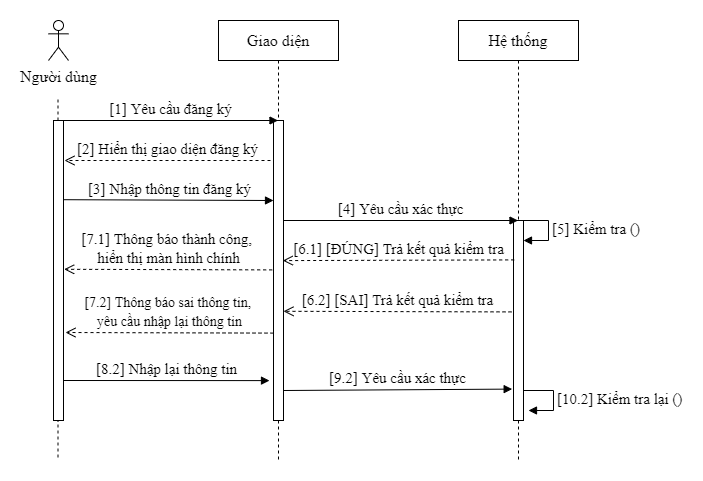
Biểu đồ tuần tự các chức năng hệ thống

Use case đăng nhập



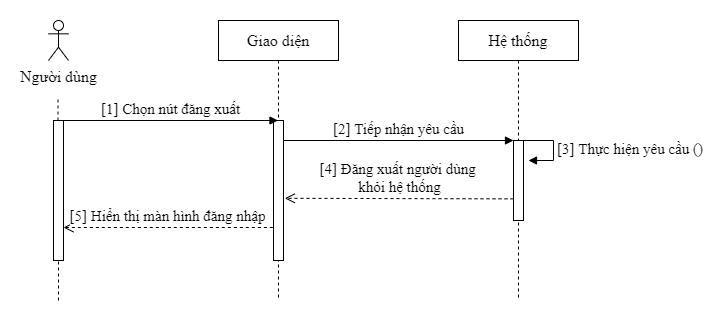
Hình - Biểu đồ tuần tự use case đăng nhập

Use case đăng ký



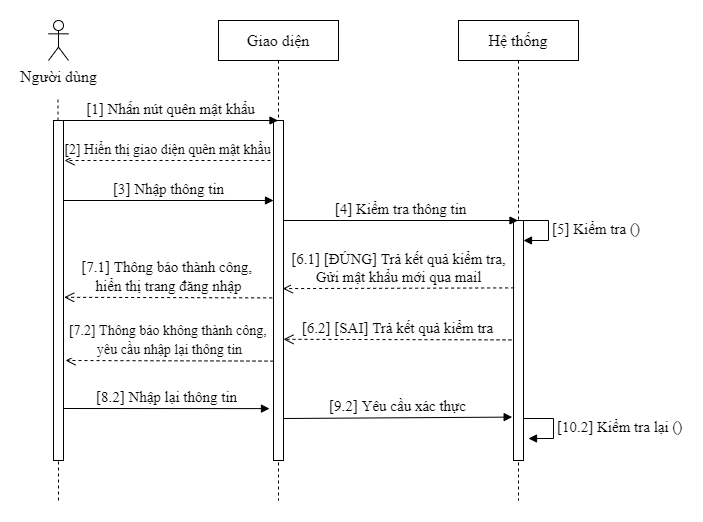
Hình - Biểu đồ tuần tự use case đăng ký

Use case đăng xuất



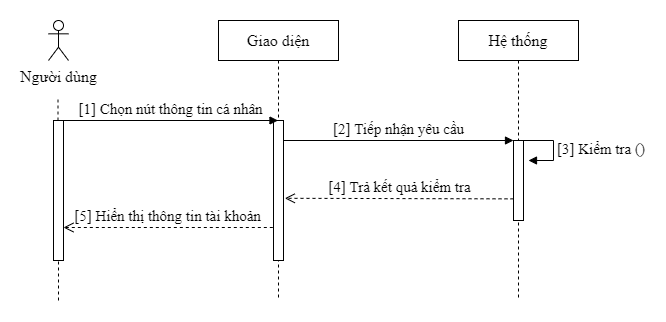
Hình - Biểu đồ tuần tự use case đăng xuất

Use case quên mật khẩu



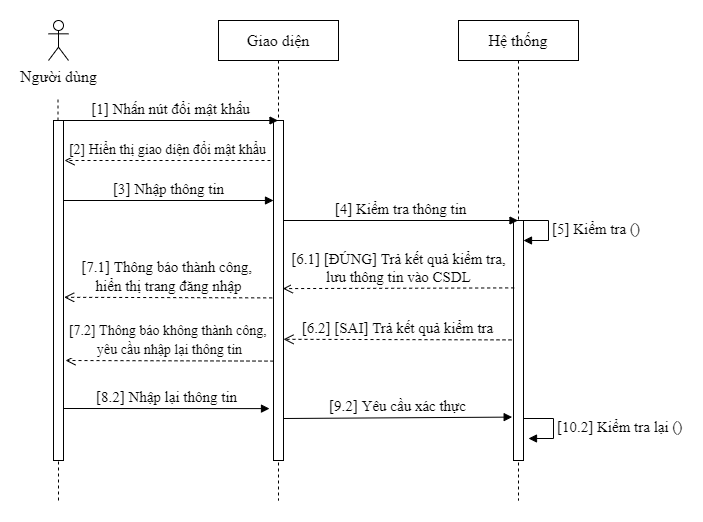
Hình - Biểu đồ tuần tự use case quên mật khẩu

Use case thông tin cá nhân



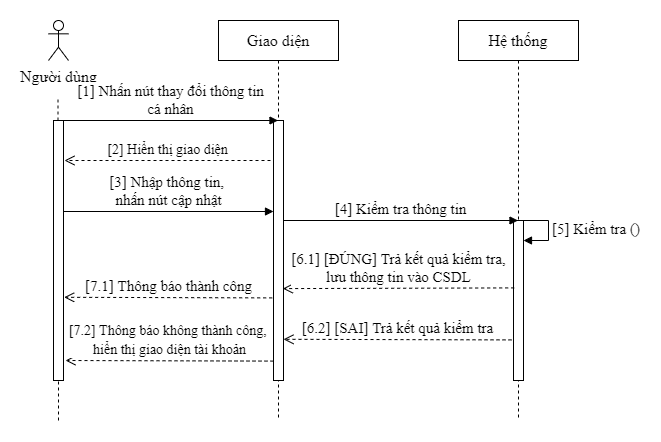
Hình - Biểu đồ tuần tự use case thông tin cá nhân

Use case thay đổi mật khẩu



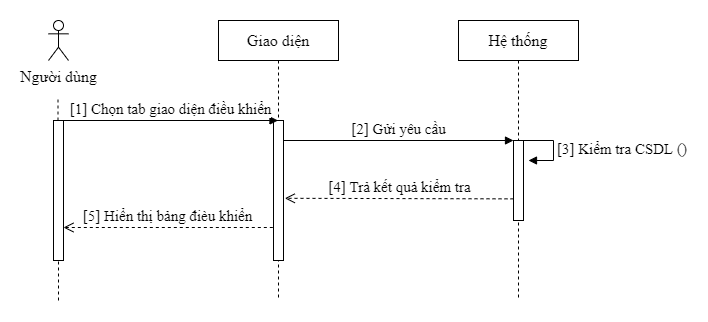
Hình - Biểu đồ tuần tự use case thay đổi mật khẩu

Use case thay đổi thông tin cá nhân



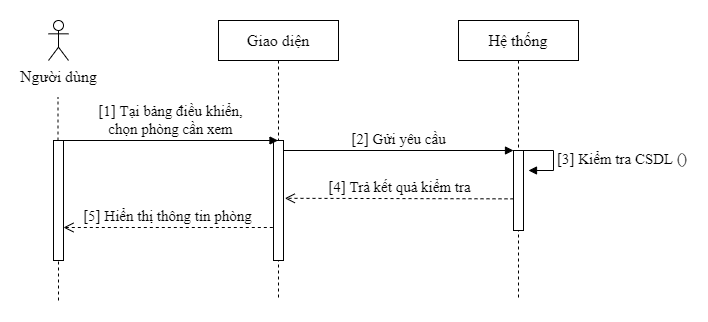
Hình - Biểu đồ tuần tự use case thay đổi thông tin cá nhân

Use case xem thông tin bảng điều khiển



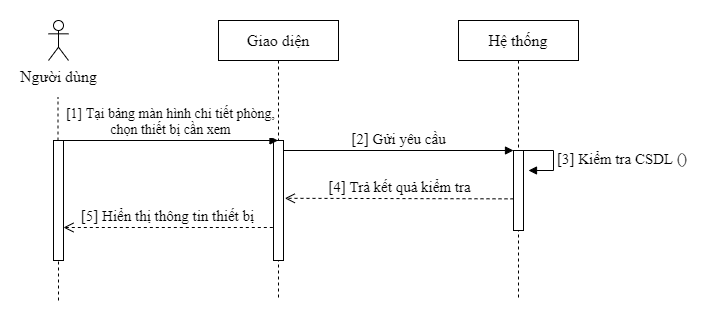
Hình - Biểu đồ tuần tự use case xem thông tin bảng điều khiển

Use case xem thông tin phòng



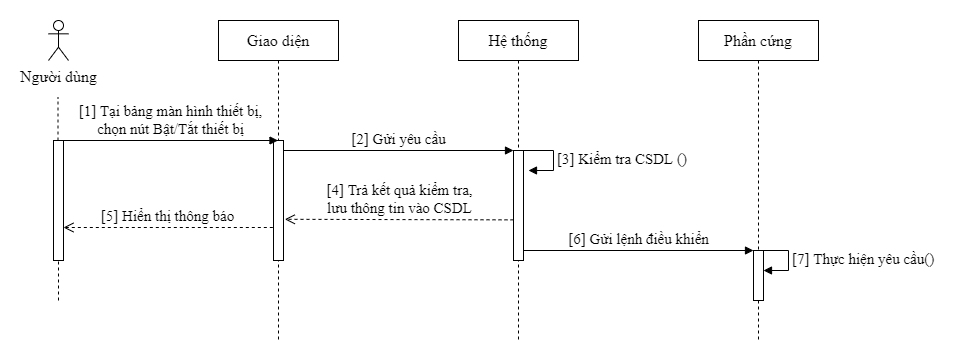
Hình - Biểu đồ tuần tự use case xem thông tin phòng

Use case xem thông tin thiết bị



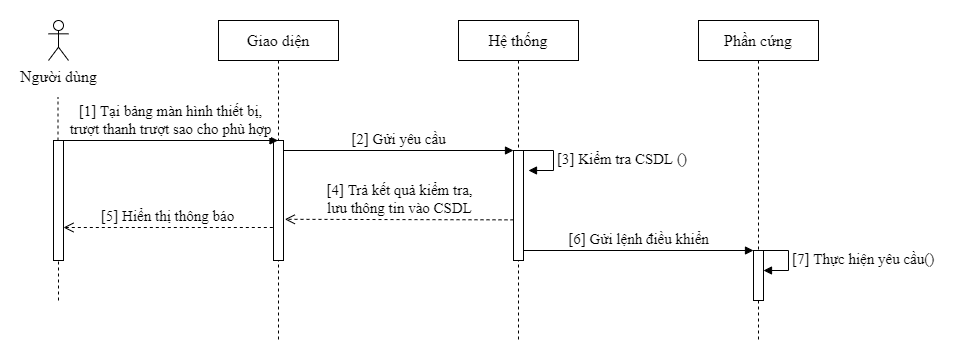
Hình - Biểu đồ tuần tự use case xem thông tin thiết bị

Use case bật hoặc tắt thiết bị



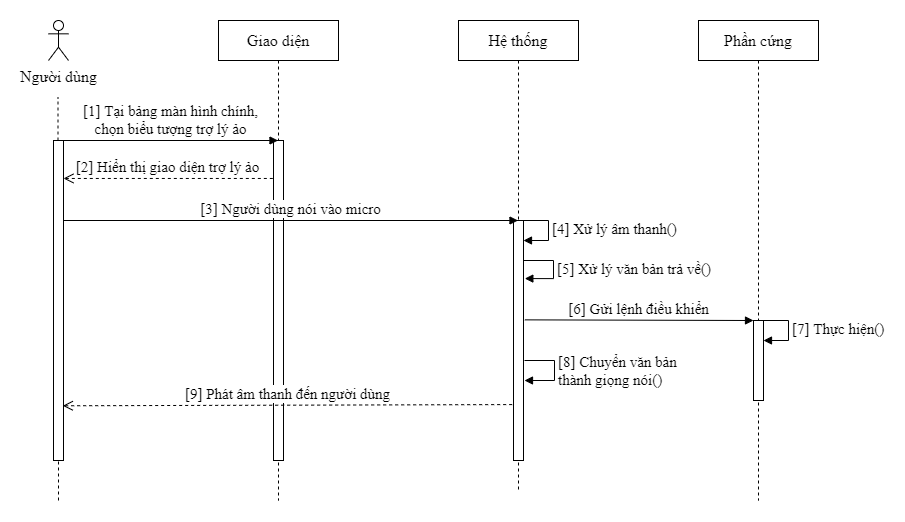
Hình - Biểu đồ tuần tự use case bật hoặc tắt thiết bị

Use case điều chỉnh cường độ thiết bị



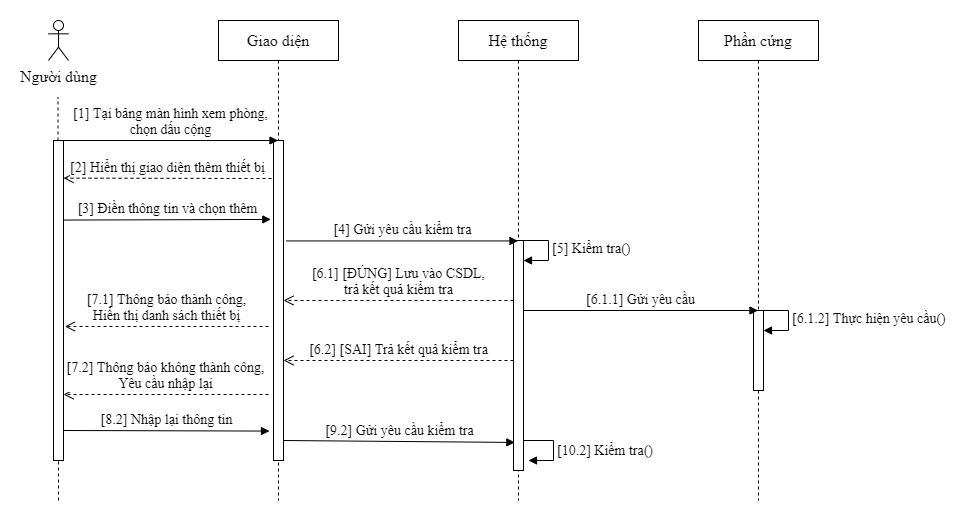
Hình - Biểu đồ tuần tự use case điều chỉnh cường độ thiết bị

Use case điều khiển bằng giọng nói (Trợ lý ảo)



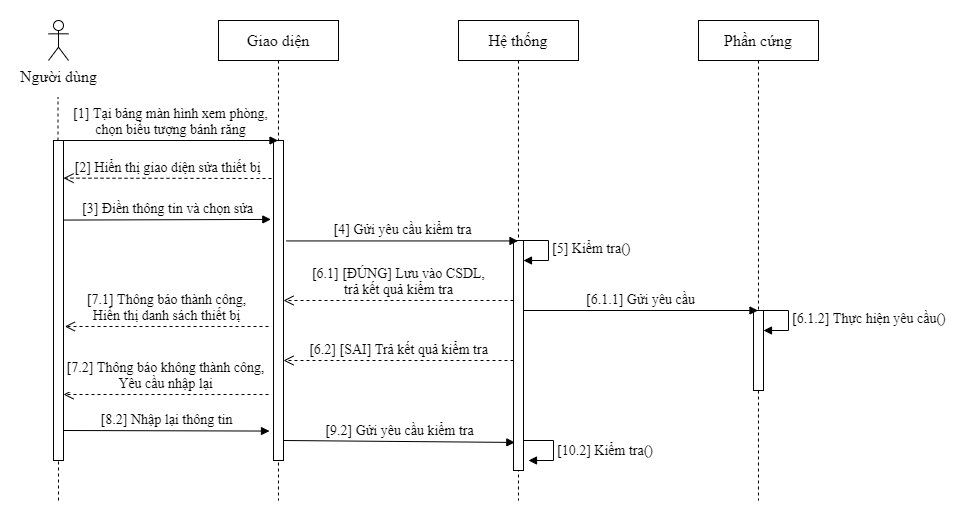
Hình - Biểu đồ tuần tự use case điều khiển bằng giọng nói

Use case thêm thiết bị



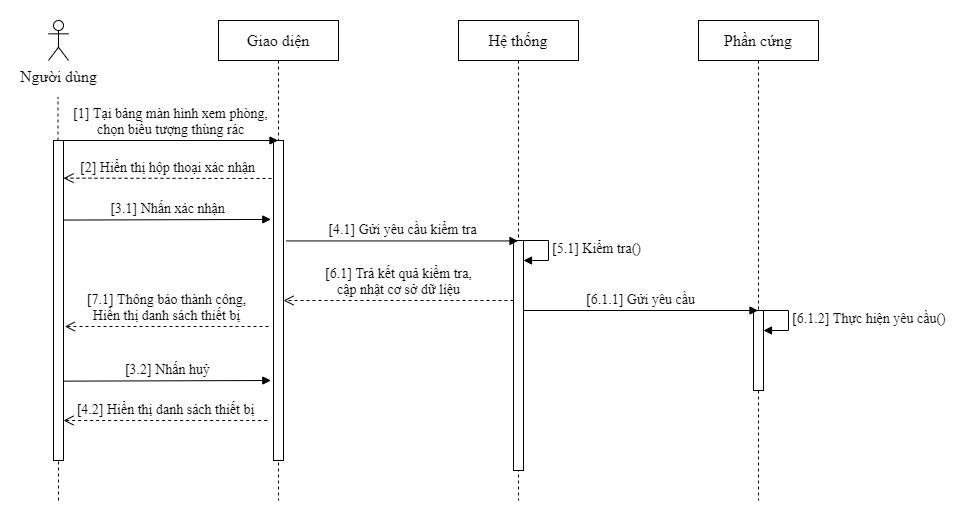
Hình - Biểu đồ tuần tự use case thêm thiết bị

Use case sửa thiết bị



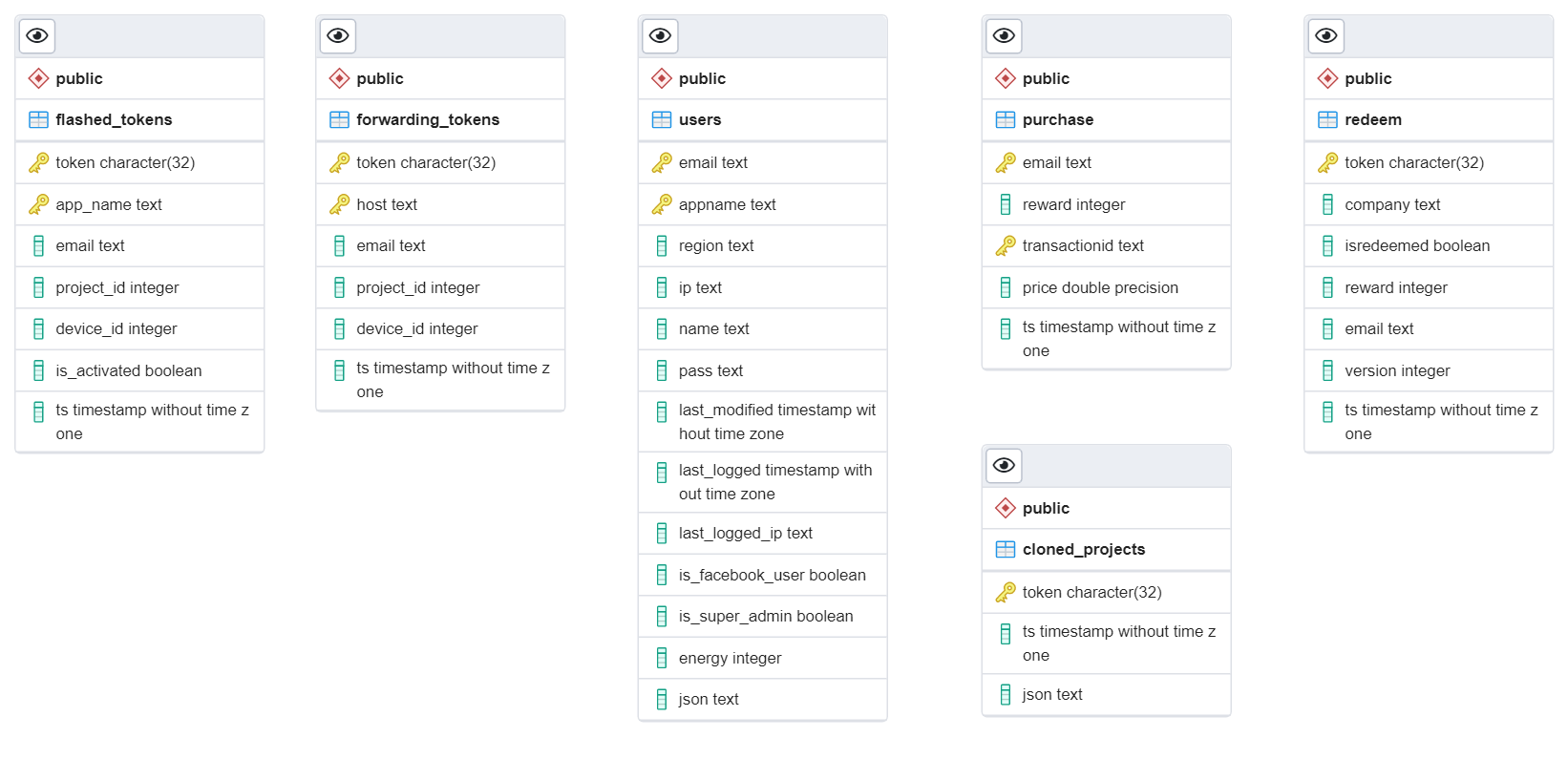
Hình - Biểu đồ tuần tự use case sửa thiết bị

Use case xoá thiết bị



Hình - Biểu đồ tuần tự use case xoá thiết bị

### Biểu đồ cơ sở dữ liệu



Hình - Biểu đồ lớp cở sở dữ liệu chính

Table

Description automatically generated

Hình - Biểu đồ lớp cở sở dữ liệu ghi nhật ký

Thiết kế cơ sở dữ liệu

Bảng users

Bảng - Bảng users

|  |  |  |
| --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Ý nghĩa** |
| 1 | email | Lưu trữ địa chỉ email, đồng thời là khoá chính |
| 2 | appname | Lưu trữ tên ứng dụng, đồng thời là khoá chính |
| 3 | region | Lưu trữ khu vực |
| 4 | ip | Lưu trữ địa chỉ IP |
| 5 | name | Lưu trữ tên |
| 6 | pass | Lưu trữ mật khẩu |
| 7 | last\_modified | Lưu trữ lần cuối cùng chỉnh sửa |
| 8 | last\_logged | Lưu trữ lần cuối cùng đăng nhập |
| 9 | last\_logged\_ip | Lưu trữ địa chỉ IP đăng nhập cuối cùng |
| 10 | is\_facebook\_user | Lưu trữ liên kết tài khoản facebook |
| 11 | is\_super\_admin | Lưu trữ có phải là quản trị hay không |
| 12 | energy | Lưu trữ năng lượng còn lại |
| 13 | json | Lưu trữ những dữ liệu khác |

Bảng forwarding\_tokens

Bảng - Bảng forwarding\_tokens

|  |  |  |
| --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Ý nghĩa** |
| 1 | token | Mã truy cập của thiết bị |
| 2 | host | Cụm máy chủ xử lý |
| 3 | email | Địa chỉ email của chủ sở hữu |
| 4 | project\_id | Mã số dự án |
| 5 | device\_id | Mã số thiết bị |
| 6 | ts | Ngày tạo |

Bảng flashed\_tokens

Bảng - Bảng flashed\_tokens

|  |  |  |
| --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Ý nghĩa** |
| 1 | token | Mã truy cập của thiết bị |
| 2 | app\_name | Tên ứng dụng |
| 3 | email | Địa chỉ email của chủ sở hữu |
| 4 | project\_id | Mã số dự án |
| 5 | device\_id | Mã số thiết bị |
| 6 | is\_activated | Thiết bị có được kích hoạt chưa |
| 7 | ts | Thời gian tạo |

Bảng purchase

Bảng - Bảng purchase

|  |  |  |
| --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Ý nghĩa** |
| 1 | email | Địa chỉ email người mua |
| 2 | reward | Số năng lượng nhận được |
| 3 | transcationid | Mã giao dịch trên cửa hàng |
| 4 | price | Giá tiền (tính bằng USD) |
| 5 | ts | Thời gian tạo |

Bảng redeem

Bảng - Bảng redeem

|  |  |  |
| --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Ý nghĩa** |
| 1 | token | Mã truy cập |
| 2 | company | Tên công ty |
| 3 | isredeemed | Đã nhận chưa? |
| 4 | reward | Năng lượng nhận được |
| 5 | email | Địa chỉ email người nhận |
| 6 | version | Phiên bản app |
| 7 | ts | Thời gian tạo |

Bảng cloned\_projects

Bảng - Bảng cloned\_projects

|  |  |  |
| --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Ý nghĩa** |
| 1 | token | Mã truy cập |
| 2 | ts | Thời gian tạo |
| 3 | json | Dữ liệu project được nhân bản |

GIAO DIỆN ỨNG DỤNG

Giao diện màn hình đăng nhập

Graphical user interface, application

Description automatically generated

Hình - Giao diện màn hình đăng nhập

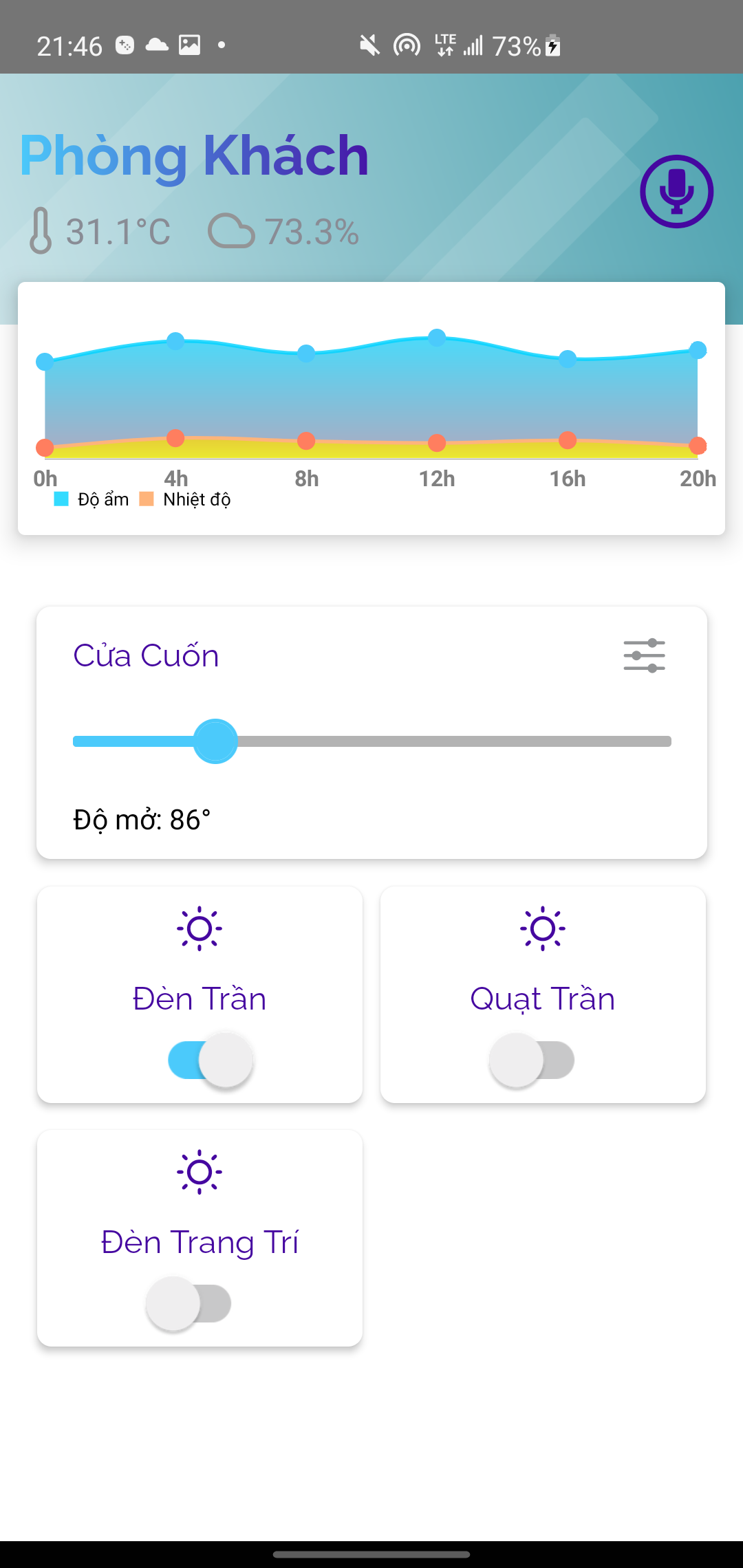
Giao diện màn hình chính

Graphical user interface, application

Description automatically generated

Hình - Giao diện màn hình chính

Giao diện màn hình xem chi tiết phòng



Hình - Giao diện màn xem chi tiết phòng

Giao diện màn hình trợ lý ảo

Text

Description automatically generated with low confidence

Hình - Giao diện màn hình xem trợ lý ảo

Giao diện màn hình thông tin

Graphical user interface, application

Description automatically generated

Hình - Giao diện mà hình thông tin

KẾT LUẬN

Kết quả đạt được

- Ứng dụng có thể điều khiển thiết bị

- Ứng dụng có thể xem, báo cáo các thông tin trong nhà

- Ứng dụng có thể điều khiển ở bất kì đâu chỉ cần có Internet

- Dễ dàng sử dụng

- Có thể điều khiển ra lệnh bằng giọng nói

- Tích hợp trợ lý ảo

- Có thể chịu tải cao do sử dụng kiến trúc Microservices

- Dễ dàng mở rộng, nâng cấp và thêm các mô đun mới trong tương lai

Hạn chế

- Ứng dụng vẫn còn hạn chế về số lượng tính năng do thời gian thực hiện dự án là có hạn.

- Chưa hỗ trợ đa ngôn ngữ đa quốc gia.

Hướng phát triển của đề tài

- Tích hợp thêm nhiều mô đun mới

- Mở rộng đề tài ở nhiều lĩnh vực khác như: trồng trọt và chăn nuôi

- Tối ưu hoá hiệu năng khi có số lượng lớn kết nối đồng thời

TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt

1. Tìm hiểu về nhà thông minh - <https://smarthomekit.vn/nha-thong-minh-tu-a-z/>
2. Dự án nhà thông minh tại Việt Nam - <https://lumi.vn/smart-home/>
3. Arduino đọc dữ liệu từ cảm biến nhiệt độ DHT11 - <http://arduino.vn/bai-viet/91-doc-nhiet-do-do-am-va-xuat-ra-man-hinh-lcd>
4. Kiến trúc phần mềm Microservices - <https://topdev.vn/blog/phat-trien-phan-mem-theo-kien-truc-microservice/>
5. Tối ưu kích thước Docker image - <https://viblo.asia/p/docker-tipstricks-lam-the-nao-de-toi-uu-hoa-kich-thuoc-cua-mot-docker-image-bWrZnm0pKxw>
6. Kỹ thuật tối ưu tốc độ bằng cache - <https://viblo.asia/p/caching-dai-phap-2-cache-the-nao-cho-hop-ly-ByEZkawE5Q0>

Tài liệu tiếng Anh

1. Nginx documents - <https://docs.nginx.com/>
2. Java documents - <https://docs.oracle.com/en/java/>
3. Netty documents - <https://netty.io/wiki/user-guide-for-4.x.html>
4. PostgreSQL documents - <https://www.postgresql.org/docs/>
5. Node.js documents - <https://nodejs.org/en/docs/>
6. NLP.js GitHub repository - <https://github.com/axa-group/nlp.js#readme>
7. Firebase documents - <https://firebase.google.com/docs>
8. ESP32 Schematic - <https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>
9. Arduino Programing Documents - <https://www.arduino.cc/en/main/docs>
10. ESP Async WiFiManager library documents - <https://github.com/khoih-prog/ESPAsync_WiFiManager>
11. Introduction Netty in Java - <https://www.baeldung.com/netty>
12. Microservices Architecture - <https://microservices.io/patterns/microservices.html>
13. Configure Nginx Reverse Proxy - <https://www.scaleway.com/en/docs/how-to-configure-nginx-reverse-proxy/>
14. Docker documents - <https://docs.docker.com/>
15. Microservices deployment in Docker - <https://developer.okta.com/blog/2019/02/28/spring-microservices-docker>
16. Smart home trends in 2021 - <https://www.techbuddy.co/post/smart-home-trends-2021>
17. React Native Performance documents - <https://reactnative.dev/docs/performance>
18. Redux documents - <https://react-redux.js.org/introduction/getting-started>
19. Android Java Socket documents - <https://developer.android.com/reference/java/net/Socket>
20. Natural Language Processing - <https://en.wikipedia.org/wiki/Natural_language_processing>
21. Convolutional Neural Networks - <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>