

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN**



LUẬN VĂN TỐT NGHIỆP
ĐỀ TÀI: NGUYÊN CỨU TÍCH HỢP IOT VÀO HỆ
THỐNG QUẢN LÝ NHÀ TRỌ

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn: ThS. Vũ Đình Long

Sinh viên thực hiện: Nguyễn Quang Linh

MSSV: 1651120032 Lớp: CN16A

TP. Hồ Chí Minh, ngày 05 tháng 12 năm 2020

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

TP.HCM, ngày ... tháng ... năm 2020

Giảng viên hướng dẫn

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

TP.HCM, ngày ... tháng ... năm 2020

Giảng viên phản biện

LỜI CẢM ƠN

Trong quá trình học tập và tìm hiểu để thực hiện luận văn tốt nghiệp này, em đã nhận được rất nhiều sự giúp đỡ vô cùng quý báu từ quý thầy, cô giáo và các bạn. Em xin gửi lời cảm ơn chân thành đến quý thầy giáo, cô giáo trường Đại học Giao Thông Vận Tải TP.HCM đã tận tình giảng dạy, tạo điều kiện hết mực cho em trong suốt quá trình học tập, rèn luyện và phát huy hết khả năng tại trường. Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến Th.S Vũ Đình Long, người luôn đồng hành cùng em trong suốt quá trình làm luận văn này. Em cũng xin cảm ơn những người luôn theo dõi, động viên và tiếp sức tinh thần cho em vượt qua những khó khăn và thử thách để hoàn thành xong khóa luận văn cuối cùng. Một lần nữa em xin chân thành cảm ơn đến tất cả và xin chúc quý thầy cô giáo, các bạn thành công trong học tập cũng như tương lai của mình.

Mặc dù đã cố gắng hết khả năng của mình nhưng do về mặt thời gian, kiến thức cũng như kinh nghiệm nghiên cứu về luận văn tốt nghiệp còn hạn chế, nên không tránh được những thiếu sót. Em rất mong được sự nhận xét, đánh giá, đóng góp ý kiến của quý thầy cô giáo và các bạn độc giả để bài luận văn tốt nghiệp hoàn thiện hơn.

LỜI NÓI ĐẦU

Ngày nay, công nghệ thông tin đã có những bước chuyển biến phát triển mạnh mẽ, thiết yếu và có ảnh hưởng sâu rộng đến thói quen, sinh hoạt, giải trí của nhiều người.

Đứng trước vai trò của thông tin hoạt động cạnh tranh gay gắt thì các tổ chức và các doanh nghiệp đều tìm các biện pháp để xây dựng hệ thống thông tin của mình hoàn thiện hơn nhằm tin học hóa các tác vụ của đơn vị với nhiều phần mềm mã nguồn mở.

Thuật ngữ “phần mềm mã nguồn mở” có nghĩa tương đương với “mã nguồn mở” nhưng với độ bao hàm cao hơn. Ngày nay có rất nhiều dạng mở bao gồm: phần cứng, phần mềm nguồn mở, tài liệu thiết kế... Phần mềm nguồn mở đang phát triển với tốc độ khá cao, cho thấy nó có nhiều động lực hơn với mã đóng.

Ở Việt Nam mã nguồn mở cũng không còn xa lạ. Việc sử dụng và phát triển mã nguồn mở ở đất nước đang phát triển như đất nước ta đang rất được quan tâm, nó giúp giảm thiểu nguồn tài chính khi chi trả bản quyền và nhiều lợi ích khác. Hiện nay, có rất nhiều sản phẩm nguồn mở như: ruby on rails, joomla, wordpress ... trong đó ruby on rails đang được quan tâm nhất bởi tính đơn giản, dễ sử dụng và đặc biệt là hoàn toàn miễn phí.

Với thực trạng hiện nay, tại các thành phố lớn số lượng người di cư đến sống và làm việc ngày càng phát triển mạnh mẽ. Song liệu có bao nhiêu phần trăm người khi di cư đến có thể tự mua một căn nhà hay một miếng đất để xây dựng nhà. Điều đó vô cùng khó khăn đối với đại đa số còn lại. Vì vậy việc chọn một ngôi nhà trọ để thuê là một vấn đề tất yếu. Hãy đặt ra câu hỏi “ Nên thuê trọ như thế nào ? Giá tầm bao nhiêu để phù hợp với mình, an ninh khu vực ra sao, có an toàn hay không? Phòng có sạch sẽ và đầy đủ tiện nghi hay không? Và quan trọng hơn hết là gần chỗ làm việc hay học tập mình không nữa, v.v...” cũng là một vấn đề hết sức khó khăn.

Cũng là một người từ nơi khác chuyển vào để học tập thì em thấu hiểu được những sự bất tiện đó. Ví như mỗi tháng phải ghi giấy tính tiền trọ, thông báo nộp tiền, rồi sự lãng phí điện, nước khi sử dụng... Nhìn thấy sự bất tiện trong quản lý nhà trọ, em đã nảy sinh một ý tưởng: “ Nghiên cứu Iot về quản lý nhà trọ”. Với ý tưởng này sẽ giúp cho những chủ trọ

quản lý trợ của mình một cách thuận tiện, còn người dùng có thể sử dụng điện một cách tiết kiệm và có hiệu quả hơn.

MỤC LỤC

PHẦN 1: CƠ SỞ LÝ THUYẾT	12
CHƯƠNG 1: TÌM HIỂU VỀ RUBY ON RAILS TRÊN MÔI TRƯỜNG LINUX.....	12
I. Khái quát.....	12
1. Ruby on Rails là gì ?.....	12
2. Tại sao chọn Rails ?.....	12
3. Ưu và nhược điểm.....	13
II. Cài đặt ruby và Rails.....	13
1. Cài NodeJs và Yarn.....	13
2. Cài rbenv.....	13
3. Cài bundler.....	14
4. Cài Rails.....	14
III. Cấu trúc của Rails.....	15
1. Cấu trúc thư mục của Rails.....	15
2. Cách chạy Rails.....	17
CHƯƠNG 2: TÌM HIỂU VỀ REACT NATIVE.....	18
I. Khái niệm.....	18
1. React native là gì ?.....	18
2. Ưu và nhược điểm.....	19
3. Cấu trúc của react native.....	20
CHƯƠNG 3: TÌM HIỂU VỀ API.....	24
I. Khái niệm.....	24
II. API thường ứng dụng ở đâu ?.....	24
III. Web API là gì ?.....	25
IV. Web API hoạt động như thế nào ?.....	25
V. Ưu điểm và nhược điểm của Web API.....	26
CHƯƠNG 4: TÌM HIỂU VỀ REACT.....	27
I. Khái niệm.....	27
1. React là gì.....	27
2. Tại sao phải sử dụng React ?.....	27
3. Ưu và nhược điểm.....	28
II. Cách cài đặt React trên môi trường linux (ubuntu).....	28
1. Cách cài đặt môi trường.....	28
2. Thư mục trong trong react.....	30
CHƯƠNG 5: TÌM HIỂU VỀ IOT.....	31
I. Khái niệm.....	31
1. IoT là gì ?.....	31
2. Tầm quan trọng của IoT.....	31
3. Cài đặt.....	31
PHẦN 2: NỘI DUNG THỰC HIỆN.....	33
CHƯƠNG 1: ỨNG DỤNG CHẠY TRÊN WEBSITE.....	33
I. Phân tích hệ thống.....	33
1. Xác định yêu cầu.....	33
2. Đặc tả yêu cầu.....	33
3. Xây dựng biểu đồ chức năng hệ thống.....	33

4. Biểu đồ hoạt động.....	36
5. Xây dựng biểu đồ thực thể.....	53
II. Thiết kế hệ thống.....	60
1. Thiết kế cơ sở dữ liệu.....	60
III. API.....	81
1. Các API được sử dụng trong trang web.....	81
CHƯƠNG 2: ỨNG DỤNG NGƯỜI DÙNG CHẠY TRÊN APP.....	94
I. Phân tích hệ thống.....	94
1. Xác định yêu cầu.....	94
2. Đặc tả yêu cầu.....	94
3. Xây dựng biểu đồ chức năng.....	94
II. Thiết kế hệ thống.....	95
1. Mô tả chi tiết từng màn hình.....	95
CHƯƠNG 3: ỨNG DỤNG CỦA NGƯỜI DÙNG CHẠY TRÊN WEB.....	103
I. Phân tích yêu cầu.....	103
II. Các màn hình trên website của khách hàng.....	103
CHƯƠNG 4: INTERNET OF THINGS (IOT).....	107
I. Cách nạp code vào module ESP8266.....	107
1. Module ESP8266 là gì ?.....	107
2. Cách nạp code.....	107
3. Module ESP8266 kết nối wifi.....	110
4. Lưu trữ dữ liệu các thiết bị ở đâu ?	112
II. Mô hình lắp đặt.....	113
1. Cách lắp đặt.....	113
PHẦN 3: ĐÁNH GIÁ VỀ HƯỚNG PHÁT TRIỂN.....	115
I. Đánh giá và kết luận.....	115
II. Hướng phát triển.....	115
TÀI LIỆU KHAM KHẢO.....	117

MỤC LỤC ẢNH

Hình 1: Cấu trúc thư mục của Rails.....	14
Hình 2: Chạy lệnh rails s.....	16
Hình 3: Kiểm tra java version.....	19
Hình 4: Khởi tạo dự án mới với react native.....	19
Hình 5: Chạy react native trên terminal.....	20
Hình 6: Chạy react native trên thiết bị thật.....	20
Hình 7: Cấu trúc thư mục của react native.....	21
Hình 8: Trao đổi dữ liệu thông qua Api.....	22
Hình 9: Kiểm tra node, npm và yarn version.....	26
Hình 10: Tạo dự án mới với reactjs.....	27
Hình 11: Chạy reactjs trên terminal.....	27
Hình 12: Chạy reactjs trên trình duyệt chrome.....	28
Hình 13: Cấu trúc thư mục reactjs.....	28
Hình 14: Cách tải Arduino trên trang chủ Arduino.....	30
Hình 15: Biểu đồ phân rã chức năng quản trị viên và chủ тро.....	33

Hình 16: Biểu đồ hoạt động đăng nhập của quản trị viên và chủ trọ.....	35
Hình 17: Biểu đồ thêm một khách hàng mới.....	36
Hình 18: Biểu đồ hoạt động thêm dịch vụ mới.....	37
Hình 19: Biểu đồ hoạt động cập nhật dịch vụ, khách hàng chung phòng và chỉ số dịch vụ.....	38
Hình 20: Biểu đồ hoạt động gửi email.....	39
Hình 21: Biểu đồ hoạt động tạo nhắc nhở.....	40
Hình 22: Biểu đồ hoạt động gửi phản hồi.....	41
Hình 23: Biểu đồ hoạt động quản lý tài khoản.....	42
Hình 24: Biểu đồ hoạt động thanh toán tiền trọ qua mail.....	43
Hình 25: Biểu đồ hoạt động trả phòng.....	44
Hình 26: Biểu đồ hoạt động thêm mới một nhà.....	45
Hình 27: Biểu đồ hoạt động thêm phòng nhanh.....	46
Hình 28: Biểu đồ hoạt động thêm một phòng.....	47
Hình 29: Biểu đồ hoạt động điều khiển bằng giọng nói.....	48
Hình 30: Biểu đồ điều khiển thiết bị trong phòng.....	49
Hình 31: Biểu đồ hoạt động tạo slider cho app và website khách hàng.....	50
Hình 32: Biểu đồ hoạt động tạo một quy định mới.....	51
Hình 33: Biểu đồ lớp.....	58
Hình 34: Giao diện đăng nhập.....	59
Hình 35: Giao diện thay đổi mật khẩu.....	60
Hình 36: Giao diện quên mật khẩu.....	60
Hình 37: Giao diện hiển thị thông tin khách hàng đã đặt phòng.....	61
Hình 38: Giao diện khách hàng cùng phòng.....	62
Hình 39: Giao diện thêm một khách hàng mới.....	62
Hình 40: Giao diện cập nhật dịch vụ.....	63
Hình 41: Giao diện cập nhật khách hàng cùng phòng.....	64
Hình 42: Giao diện cập nhật chỉ số dịch vụ điện, nước.....	64
Hình 43: Giao diện gửi thanh toán hóa đơn về mail.....	65
Hình 44: Giao diện quá hạn trả tiền được gửi về mail.....	65
Hình 45: Giao diện quản lý thiết bị của chủ trọ.....	66
Hình 46: Giao diện quản lý phòng của chủ trọ.....	66
Hình 47: Giao diện tiếp nhận trả phòng.....	67
Hình 48: Giao diện hiển thị chi tiết các dụng vụ khách đang sử dụng của chủ trọ.....	67
Hình 49: Giao diện quản lý thông tin khách hàng của chủ trọ.....	68
Hình 50: Giao diện quản lý tài khoản của chủ trọ.....	68
Hình 51: Giao diện tạo nhắc nhở của chủ trọ.....	69
Hình 52: Giao diện tạo mới một nhắc nhở của chủ trọ.....	70
Hình 53: Giao diện tạo một nhắc nhở ngày đã được chọn của chủ trọ.....	71
Hình 54: Giao diện xóa một nhắc nhở của chủ trọ.....	72
Hình 55: Giao diện thống kê của chủ trọ.....	73
Hình 56: Giao diện xuất ra hóa đơn file txt.....	73
Hình 57: Giao diện quản lý phản hồi từ khách hàng của chủ trọ.....	74
Hình 58: giao diện gửi mail của chủ trọ.....	75
Hình 59: Giao diện tạo slider cho app và web bên khách hàng của quản trị viên.....	75
Hình 60: Giao diện quản lý tất cả các thiết bị ở tất cả khu vực của quản trị viên.....	76

Hình 61: Giao diện quản lý phòng của tất cả khu vực của quản trị viên.....	76
Hình 62: Giao diện tạo một nhà mới.....	77
Hình 63: Giao diện tạo một phòng mới.....	77
Hình 64: Giao diện tạo một dịch vụ mới của quản trị viên.....	78
Hình 65: Giao diện quản lý khách hàng.....	78
Hình 66: Giao diện quản lý tài khoản của quản trị viên.....	79
Hình 67: Giao diện quản lý chi phí của quản trị viên.....	79
Hình 68: Biểu đồ phân rã chức năng của khách hàng.....	94
Hình 69: Màn hình trang chủ chưa đăng nhập của khách hàng.....	95
Hình 70: Màn hình đăng nhập của khách hàng.....	96
Hình 71: Màn hình trang chủ của khách hàng sau khi đăng nhập.....	97
Hình 72: Màn hình cập nhật thông tin cá nhân khách hàng.....	98
Hình 73: Màn hình điều khiển thiết bị của khách hàng.....	99
Hình 74: Màn hình hiển thị dịch vụ và chi phí hàng tháng của khách hàng.....	100
Hình 75: Màn hình gửi phản hồi của khách hàng.....	101
Hình 76: Giao diện đăng nhập của khách hàng.....	102
Hình 77: Giao diện trang chủ của khách hàng.....	103
Hình 78: Giao diện điều khiển thiết bị của khách hàng.....	103
Hình 79: Giao diện xét thời gian cho thiết bị của khách hàng.....	104
Hình 80: Giao diện cập nhật thông tin cá nhân của khách hàng.....	104
Hình 81: Giao diện phản hồi của khách hàng.....	105
Hình 82: Module Esp8266.....	106
Hình 83: Cài đặt package cho arduino.....	107
Hình 84: Cài đặt thư viện trong arduino.....	107
Hình 85: Cách nạp code cho module Esp8266.....	108
Hình 86: Sau khi nạp code thành công.....	109
Hình 87: Hiển thị trạng thái kết nối với wifi thành công.....	110
Hình 88: Mô hình kết nối với led.....	113
Hình 89: Sơ đồ mạch điện mô hình thực tế.....	114

MỤC LỤC BẢNG

Bảng 1: Users.....	51
Bảng 2: Information.....	51
Bảng 3: Houses.....	52
Bảng 4: Cities.....	52
Bảng 5: Districts.....	52
Bảng 6: Wards.....	53
Bảng 7: Rooms.....	53
Bảng 8: Services.....	54
Bảng 9: Use_services.....	54
Bảng 10: Members.....	54
Bảng 11: Paytherents.....	54
Bảng 12: Reminders.....	55
Bảng 13: Reports.....	55
Bảng 14: Regulations.....	56

Bảng 15: Apps.....	56
Bảng 16: Api AppSlider.....	79
Bảng 17: Api đăng nhập.....	80
Bảng 18: Api cập nhật thông tin cá nhân.....	81
Bảng 19: Api hiển thị bảng điều khiển.....	82
Bảng 20: Api chế độ điều khiển thiết bị: Bật, tắt và hẹn giờ.....	84
Bảng 21: Api lấy thông tin dịch vụ đã đăng ký.....	85
Bảng 22: Api lấy dữ liệu thanh toán hàng tháng.....	86
Bảng 23: Api gửi phản hồi.....	87
Bảng 24: Api Lấy thông tin đã phản hồi.....	88
Bảng 25: Api xóa một phản hồi.....	89
Bảng 26: Api lấy thông tin nội quy.....	89

PHẦN 1: CƠ SỞ LÝ THUYẾT

CHƯƠNG 1: TÌM HIỂU VỀ RUBY ON RAILS TRÊN MÔI TRƯỜNG LINUX

I. Khái quát

1. Ruby on Rails là gì ?

Trước khi tìm hiểu về Ruby on Rails, chúng ta sẽ nói về ngôn ngữ Ruby. Ruby là một ngôn ngữ lập trình hướng đối tượng được sáng tạo năm 1993 bởi Yukihiro Matz Matsumoto. Với niềm tin rằng mục đích của cuộc sống là hạnh phúc, Ruby ra đời nhằm giúp công việc lập trình trở nên dễ dàng và vui hơn.

Tuy nhiên, ngôn ngữ Ruby không được thiết kế sử dụng trên web. Hay nói cách khác là để viết các trang web hay ứng dụng web với Ruby thì cần một framework hỗ trợ.

Ruby on Rails là một web framework mã nguồn mở được phát triển các ứng dụng với ngôn ngữ Ruby và cho phép chạy được như một trang web. Rails gọi tắt của Ruby on Rails (ROR), thường được các developer sử dụng.

Rails sẽ giúp chúng ta viết ứng dụng web với Ruby một cách dễ dàng. Ngoài Rails còn có các framework ra đời để hỗ trợ Ruby ví dụ như Sinatra hay Hanami.

2. Tại sao chọn Rails ?

- Nhanh

Thiết kế website bằng Rails rất nhanh, Rails sẽ hỗ trợ tất cả về giao diện cũng như các thuộc tính cơ bản chỉnh sửa, xóa, tạo. Vì vậy khi sử dụng Rails chúng ta cần thao tác thêm bootstrap là có một trang web.

Ngoài ra, Rails cũng cung cấp nhiều plugin được gọi là Ruby Gems. Cùng với các công cụ này sẽ giúp các developer sử dụng một cách hiệu quả hơn.

- Linh động

Sau khi khởi chạy ứng dụng web của bạn, có thể chỉnh sửa một cách dễ dàng. Thêm tính năng mới hay thay đổi chuẩn dữ liệu thì thao tác thực hiện rất nhanh. Tiết kiệm cả thời gian và chi phí.

- Hữu ích

Rails là một framework khá dễ tiếp cận, về căn bản code của Rails rất dễ đọc - tính năng này rất hữu ích khi chuyển giao dự án cũng như nhiều bên tham gia vào dự án.

3. Ưu và nhược điểm

- Nhược điểm
 - Thời gian chạy: So với các ngôn ngữ khác như NodeJS hay GoLang, Rails có tốc độ chạy chậm hơn. Vấn đề thường gặp ở việc thiết kế server, thậm chí là cả cơ sở dữ liệu.
 - Tốc độ boot: Vấn đề chính gây ra khó chịu khi làm việc với Rails là tốc độ boot. Phụ thuộc vào số lượng gem và files. Mất thời gian để khởi động điều này ảnh hưởng trực tiếp đến hiệu suất làm việc.

II. Cài đặt ruby và Rails

1. Cài NodeJs và Yarn

- sudo apt install curl
- curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
- curl sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo aptkey add -
- echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list
- sudo apt-get update
- sudo apt-get install git-core zlib1g-dev build-essential libssl1-dev libreadline-dev libyaml-dev libsqlite3-dev sqlite3 libxml2-dev libxslt1-dev libcurl4-openssl-dev software-properties-common libffi-dev nodejs
- yarn

2. Cài rbenv

- cd
- git clone https://github.com/rbenv/rbenv.git ~/.rbenv
- echo 'export PATH="\$HOME/.rbenv/bin:\$PATH"' >> ~/.bashrc
- echo 'eval "\$(rbenv init -)"' >> ~/.bashrc
- exec \$SHELL

- git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-build
- echo ‘export PATH= “\$HOME/.rbenv/plugins/ruby-build/bin:\$PATH”’ >> ~/.bashrc
- exec \$SHELL
- rbenv install 2.5.1
- rbenv global 2.5.1
- ruby -v

3. *Cài bundler*

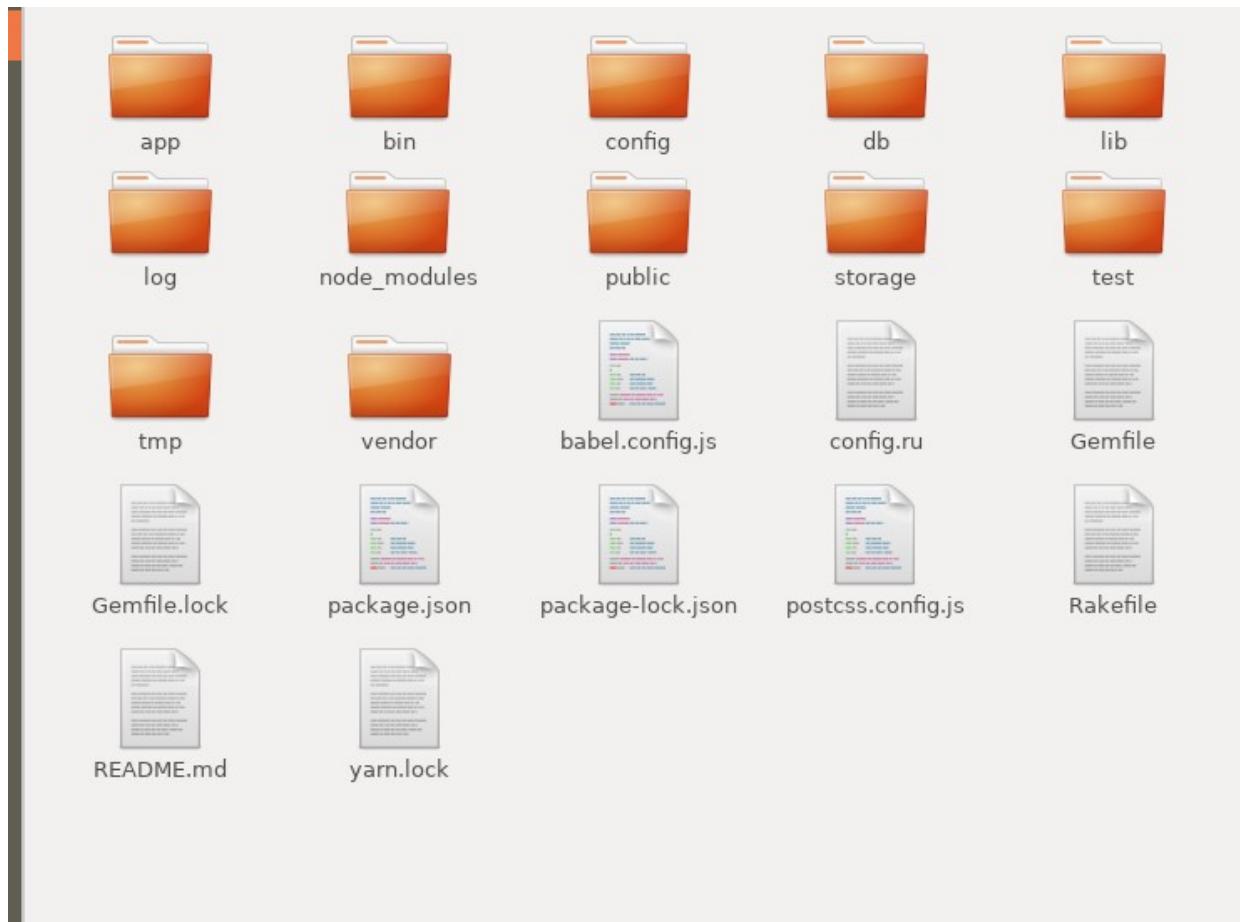
- gem install bundler

4. *Cài Rails*

- gem install rails -v 6.0.2.2
- rbenv rehash
- rails -v

III. Cấu trúc của Rails

1. Cấu trúc thư mục của Rails



Hình 1: Cấu trúc thư mục của Rails

Cấu trúc thư mục của Rails gồm:

- App: Nó tổ chức các thành phần chính ứng dụng và hầu hết code nằm ở thư mục này. Trong app chứa các thư mục con View, Controller, Model và một số thư mục khác.
 - app/assets: Chứa các file liên quan đến front-end như: Javascript, stylesheet, fonts, images
 - app/controllers: Là bộ não của controller trong mô hình MVC. Việc đặt tên của file này bắt buộc phải theo quy tắc snake_case nghĩa là tên model + “_controller” ví dụ users_controller.rb. Bạn có thể dùng generate script: rails g controller controller_name aciton_name

- app/helpers: Ứng với mỗi controller sẽ có một helper tương ứng với nó. Helper sẽ giúp chúng ta thực hiện dễ dàng và ngắn gọn sau đó sẽ show ra view.
- app/mailers: Mailer mục đích gửi mail. Nó cũng giống như controller và nó sẽ chứa ở file view : app/views/mailers_name. Để tạo mailer thì có thể dùng script: rails g mailer MailerName
- app/models: Models đóng vai trò như 1 đối tượng kết nối với database. Quy tắc đặt tên model sẽ viết các object table trong database ở dạng số ít.
- app/views: Người dùng sẽ nhìn thấy trên giao diện web hiện thị là những file được viết trong phần view này. Những file này được kết hợp giữa html và Ruby với cấu trúc file là file.html.erb và được tổ chức dựa trên controller. Ví dụ UsersController#index thì trong view sẽ có tương ứng với file app/view/users/index.html.erb
- Bin: Chứa các file binstubs cho Rails. Nó có thể thay cho bundle exec <command>. Ở đây có thể chạy các gem hoặc chứa các script của dự án. Mặc định sẽ có bundle, rails, rake, setup, spring.
- Config: Là nơi config đến đến database, routes, declare, assets, locales với nhằm mục đích hỗ trợ cho người sử dụng một cách hiệu quả để tương tác với các tùng bộ phận trong Rails.
- Db: db là nơi lưu trữ dữ liệu. Ngoài ra bạn có thể khởi tạo dữ liệu mẫu trong db/seeds.rb để khi chạy migrate dữ liệu đưa vào lưu trữ trong db
- Lib: Chứa những thư viện cho app. Có thể toàn source thư viện như là một gem chỉ dành riêng cho dự án của bạn.
 - lib/assets: Chứa các library assets có thể hiểu như là các images, scripts, stylesheets mà không có ở ứng dụng nói cách khác là những cái bên ngoài chúng ta đưa vào.
 - lib/task: Đây là nơi chứa các file rake. Có thể chạy lúc deploy hay trên server khi cần script thay đổi dữ liệu của app. Hoặc những việc chạy thường xuyên tự động theo giờ bằng crontab.

- Log: Chứa toàn bộ log file của dự án. Sẽ khác nhau tùy theo môi trường dự án.
- Public: Thường dùng để làm sitemap handle một số request ví dụ như 404, 402, 500 và được tạo ra cùng với favicon và một robots.txt
- Test: Đây là folder chứa source test cho dự án hoặc bạn dùng Rspec thì sẽ có một folder spec
- Tmp: Chứa toàn bộ cache như build assets, mail.
- Vender/assets: Đây là thư viện bên thứ 3 như là javascript và css. Những file ở đây sẽ thành một phần của assets pipeline một cách tự động.

2. *Cách chạy Rails*

- Mở terminal
- Khởi tạo dự án mới: rails new webser
- Đi vào thư mục dự án: cd webser
- Chạy dự án: rails s

```
=> Booting Puma
=> Rails 6.0.3.2 application starting in development
=> Run `rails server --help` for more startup options
/usr/lib/ruby/vendor_ruby/rubygems/defaults/operating_system.rb:10: warning: con
stant Gem::ConfigMap is deprecated
/usr/lib/ruby/vendor_ruby/rubygems/defaults/operating_system.rb:10: warning: con
stant Gem::ConfigMap is deprecated
Puma starting in single mode...
* Version 4.3.5 (ruby 2.5.1-p57), codename: Mysterious Traveller
* Min threads: 5, max threads: 5
* Environment: development
* Listening on tcp://127.0.0.1:3000
* Listening on tcp://[::1]:3000
Use Ctrl-C to stop
```

Hình 2: Chạy lệnh rails s

- Mở trình duyệt chrome: localhost:3000
- Tương tác với database ví dụ tạo bảng users: rails g scaffold user email:string name:string
- rails db:migrate
- Trên chrome: localhost:3000/users

CHƯƠNG 2: TÌM HIỂU VỀ REACT NATIVE

I. Khái niệm

1. React native là gì ?

React native là một framework do công ty công nghệ nổi tiếng của facebook phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybird và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động.

React native có thể build ứng dụng đa nền tảng chứ không phải là một “mobile web app”, không phải là “HTML5 app”, và cũng không pahir là một “Hybird app” hay cũng không chỉ build chỉ trên IOS hay Android mà chúng ta build và chạy trên cả 2 hệ sinh thái.

Điểm nổi bật của React native là việc sử dụng Hot-Loading tức là sau khi lưu thì chúng ta không cần build lại app mà nó tự động chạy những sự thay đổi trong quá trình code.

Native App là các ứng dụng được build và phát triển trên chính công cụ hỗ trợ mà nhà phát triển IOS hay Android cung cấp cho người lập trình viên ở đây là XCode và Android.

Hybird App là sự kết hợp giữa ứng dụng web và ứng dụng trên mobile. Đồng thời như ứng dụng web được xây dựng HTML + CSS + JS. Nhược điểm của Hybird App đó chính là vẫn đề hiệu năng sẽ bị ảnh hưởng đáng kể cũng như không tương tác được kết hợp với tài nguyên hệ thống phần mềm cũng như phần cứng.

2. **Ưu và nhược điểm**

Ưu điểm:

- Hiệu quả về mặt thời gian khi mà bạn muốn phát triển một ứng dụng một cách nhanh chóng.
- Hiệu năng tương đối ổn định.
- Cộng đồng phát triển mạnh mẽ.
- Tiết kiệm chi phí.
- Team phát triển nhỏ.
- 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 hybird app.

Nhược điểm:

- Vẫn đòi hỏi native code.
- Hiệu năng sẽ thấp hơn với app thuần native code.
- Bảo mật không cao do dựa trên JS
- Quản lý bộ nhớ.
- Khả năng tùy biến cũng không thật sự tốt đối với một vài module.

a) Cài đặt môi trường

- Reactive native có thể chạy ở tất cả các môi trường như Windows, Linux, MacOs.

Trong dự án này chỉ hướng dẫn cài trên hệ điều hành linux (ubuntu):

- Mở terminal

- Cài đặt node và npm:
 - Cài NVM:
 - sudo apt-get update
 - sudo apt-get install build-essential libssl-dev
 - Cài: curl -o https://raw.githubusercontent.com/creationix/nvm/v0.33.8/install.sh | bash
 - reload bash: source ~/.bashrc
 - Kiểm tra: nvm --version
 - Cài Node:
 - Cài đặt version mới nhất: nvm install node
 - Đặt mặc định cho node: nvm alias default node
 - Kiểm tra phiên bản của node và npm: node -v, npm -v
 - Cài SDK:
 - sudo apt update
 - sudo apt install default-jdk
 - java -version

```
lihn@lihn-x556ur:~$ java -version
openjdk version "1.8.0_252"
OpenJDK Runtime Environment (build 1.8.0_252-8u252-b09-1~18.04-b09)
OpenJDK 64-Bit Server VM (build 25.252-b09, mixed mode)
lihn@lihn-x556ur:~$
```

Hình 3: Kiểm tra java version

- Cài đặt react-native
 - Mở terminal
 - npm install -g react-native-cli

3. Cấu trúc của react native

- a. Tạo một ứng dụng với react-native
 - Mở terminal
 - react-native init MyProject

```
lihn@lihn-x556ur:~$ react-native init MyProject
This will walk you through creating a new React Native project in /home/lihn/MyProject
Using yarn v1.22.4
Installing react-native...
yarn add v1.22.4
warning ..\package.json: No license field
info No lockfile found.
[1/4] Resolving packages...
warning react-native > fbjs > core-js@2.6.11: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning react-native > fbjs-scripts > core-js@2.6.11: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning react-native > @react-native-community/cli > @hapi/joi@15.1.1: joi is leaving the @hapi organization and moving back to 'joi' (https://github.com/sideway/joi/issues/2411)
warning react-native > @react-native-community/cli > metro > metro-babel-register > core-js@2.6.11: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning react-native > @react-native-community/cli > @hapi/joi > @hapi/address@2.1.4: This version has been deprecated and is no longer supported or maintained
warning react-native > @react-native-community/cli > @hapi/joi > @hapi/bourne@1.3.2: This version has been deprecated and is no longer supported or maintained
warning react-native > @react-native-community/cli > @hapi/joi > @hapi/topo@3.1.6: This version has been deprecated and is no longer supported or maintained
warning react-native > @react-native-community/cli > @hapi/joi > @hapi/hoek@8.5.1: This version has been deprecated and is no longer supported or maintained
warning react-native > @react-native-community/cli > @hapi/joi > @hapi/topo > @hapi/hoek@8.5.1: This version has been deprecated and is no longer supported or maintained
warning react-native > @react-native-community/cli > metro > jest-haste-map > fsevents@1.2.13: fsevents 1 will break on node v14+ and could be using insecure binaries. Upgrade to fsevents 2.
, kind-of@^5.0.0 ]
```

Hình 4: Khởi tạo dự án mới với react native

- cd MyProject
 - Vào thư mục android tạo file local.properties với nội dung: sdk.dir = /home/<USER>/Android/Sdk/
 - Chạy các packager: react-native start



Hình 5: Chạy react native trên terminal

- Mở 1 màn hình terminal khác chạy trên android: react-native run-android

```

Info Running jetifier to migrate libraries to AndroidX. You can disable it using "--no-jetifier" flag.
Jetifier found 1227 file(s) to forward-jetify. Using 4 workers...
Info JS server already running.
* daemon not running; starting now at tcp:5037
* daemon started successfully
Info Installing the app...
Starting a Gradle Daemon (subsequent builds will be faster)

> Task :app:stripDebugDebugSymbols UP-TO-DATE
Compatible side by side NDK version was not found.

> Task :app:installDebug
09:57:50 V/ddms: execute: running am get-config
09:57:52 V/ddms: execute 'am get-config' on '920133c2a0877441' : EOF hit. Read: -1
09:57:52 V/ddms: execute: returning
Installing APK 'app-debug.apk' on 'SM-G532G - 6.0.1' for app:debug
09:57:52 D/app-debug.apk: Uploading app-debug.apk onto device '920133c2a0877441'
09:57:52 D/Device: Uploading file onto device '920133c2a0877441'
09:57:52 D/ddms: Reading file permission of /home/lthn/webser_app/android/app/build/outputs/apk/debug/app-
debug.apk as: rw-r--r--
09:57:59 V/ddms: execute: running pm install -r -t "/data/local/tmp/app-debug.apk"
09:58:41 V/ddms: execute 'pm install -r -t "/data/local/tmp/app-debug.apk"' on '920133c2a0877441' : EOF h
it. Read: -1
09:58:41 V/ddms: execute: returning
09:58:41 V/ddms: execute: running rm "/data/local/tmp/app-debug.apk"
09:58:42 V/ddms: execute 'rm "/data/local/tmp/app-debug.apk"' on '920133c2a0877441' : EOF hit. Read: -1
09:58:42 V/ddms: execute: returning
Installed on 1 device.

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.0.1/userguide/command\_line\_interface.html#sec:command\_line\_warnings

BUILD SUCCESSFUL in 1m 36s
207 actionable tasks: 2 executed, 205 up-to-date
Info Connecting to the development server...
adb server version (41) doesn't match this client (39); killing...
* daemon started successfully
Info Starting the app on "920133c2a0877441"...
Starting: Intent { cmp=com.webser_app/.MainActivity }

```

Hình 6: Chạy react native trên thiết bị thật

- Lưu ý: Bạn có thể build ở thiết bị điện thoại của bạn hoặc có thể sử dụng máy ảo. Ở đây mình build trên thiết bị thật là SamSung SG-G532G.

b. Xây dựng cấu trúc thư mục

```
File Edit View Search Terminal Help
=====
" ===== Netrw Directory Listing ===== (netrw v156)
" /home/lhn/webser_app
" Sorted by name
" Sort sequence: [ \ / ] $ , <core\%(\.\d+\)\>=\>, \.h$, \.c$, \.cpp$, \~\=*$,* , \.o$, \.obj$, \.info$, \.swp$, \.b
" Quick Help: <F1>:help -:go up dir D:delete R:rename s:sort-by x:special
" =====
[ / .
./
.git/
__tests__/
android/
components/
images/
ios/
node_modules/
.buckconfig
.eslintrc.js
.flowconfig
.gitattributes
.gitignore
.prettierrc.js
.watchmanconfig
App.js
README.md
app.json
babel.config.js
index.js
metro.config.js
package-lock.json
package.json
yarn.lock
.App.js.swp
~
~
```

Hình 7: Cấu trúc thư mục của react native

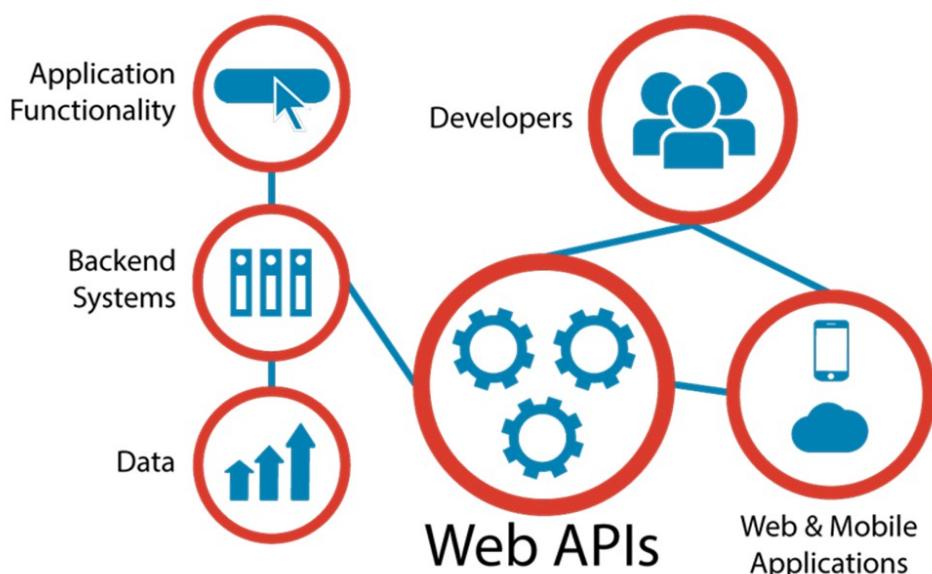
Bao gồm các phần chính:

- android/ios: Là ứng dụng chạy trên nền tảng android hay ios hoặc cả 2.
- components: Là nơi chứa các màn hình ví dụ như màn hình LoginComponent.js, UsersComponent.js....
- Images: Là nơi lưu trữ các hình ảnh.
- node_modules: Nơi chứa các thư viện hỗ trợ trong quá trình sử dụng.
- App.js: Kết nối các màn hình lại với nhau.
- Package: Là nơi lưu trữ các thư viện đã được cài đặt.

CHƯƠNG 3: TÌM HIỂU VỀ API

I. Khái niệm

API là các phương thức, giao kết nối với các thư viện và ứng dụng khác. Nó viết tắt của Application Programming Interface - giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó trao đổi dữ liệu giữa các ứng dụng với nhau.



Hình 8: Trao đổi dữ liệu thông qua API

II. API thường ứng dụng ở đâu ?

Web API: Là hệ thống các API được sử dụng ở trong web. Và có tất cả các website đều ứng dụng đến Web API cho phép kết nối, lấy dữ liệu hoặc truy cập vào cơ sở dữ liệu. Ví dụ như chức năng login bằng các tài khoản Facebook, Google, Github ... Điều này có nghĩa là đang gọi đến API để sử dụng hoặc là những ứng dụng chạy trên điện thoại đều lấy dữ liệu thông qua API.

API trên hệ điều hành: Trên các hệ điều hành Window hoặc Linux có rất nhiều API, nhà phát triển cung cấp các tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức kết nối. Nó giúp lập trình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành.

API của thư viện phần mềm hay framework: API mô tả và quy định các hành động mong muốn và các thư viện cung cấp. Một API có thể có nhiều cách khai thác khác nhau và nó cung cấp cho một chương trình hoặc một ngôn ngữ có thể sử dụng thư viện được viết bằng một ngôn ngữ khác. Ví dụ có thể dùng Ruby on Rails để yêu cầu thư viện viết bằng ngôn ngữ C++.

API hiện nay đều tuân thủ theo tiêu chuẩn REST và HTTP, tạo sự thân thiện để sử dụng với nhà phát triển. Giúp người dùng dễ dàng truy cập. Web API hiện đại dùng các đối tượng cụ thể như mobile developer với document, version khác nhau.

III. Web API là gì ?

Web API là một phương thức để cho phép các ứng dụng khác có thể giao tiếp trao đổi dữ liệu qua lại. Dữ liệu được Web API trả lại thường ở các dạng như JSON hoặc XML thông qua giao thức HTTP hoặc HTTPS.

Web API hỗ trợ restful đầy đủ các loại phương thức: Get/Post/Put/Delete dữ liệu. Nó sẽ giúp xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Đồng thời cũng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content format.

Tự động hóa sản phẩm: Web API sẽ tự động hóa quản lý công việc, cập nhật luồng công việc, giúp tăng xuất và tạo hiệu quả công việc cao hơn.

Tích hợp linh động: API cho phép lấy nội dung bất kỳ website hoặc ứng dụng nào một cách dễ dàng nếu được cho phép. Ngoài ra việc chia sẻ thông tin được chọn nhưng vẫn tránh được các yêu cầu không mong muốn.

IV. Web API hoạt động như thế nào ?

Xây dựng URL API để bên thứ ba có thể gửi request để trao đổi dữ liệu thông qua giao thức HTTP hoặc HTTPS.

Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.

Server sẽ trả kết quả dưới dạng JSON hoặc XML thông qua HTTP hoặc HTTPS.

Tại nơi yêu cầu request (di động hoặc ứng dụng web) dữ liệu JSon/XML sẽ được parse để lấy data. Sau đó sẽ thực hiện hiển thị dữ liệu hoặc lưu dữ liệu ...

V. Ưu điểm và nhược điểm của Web API

Ưu điểm

- Web API được sử dụng hầu hết trên các ứng dụng desktop, ứng dụng mobile và ứng dụng website.
- Linh hoạt với các dạng dữ liệu trả về phía client: JSon, XML hay định dạng khác.
- Xây dựng nhanh HTTP server: URI, request/response headers, caching, versioning, content formats và có thể host trong ứng dụng hoặc trên IIS.
- Mã nguồn mở, hỗ trợ chức năng RESTFUL.
- Hỗ trợ đầy đủ các thành phần MVC: routing, controller, action result, filter, model binder,...
- Giao tiếp hai chiều được xác nhận trong các giao dịch, đảm bảo độ tin cậy cao.

Nhược điểm

- Để sử dụng cần phải có kiến thức chuyên sâu, có kinh nghiệm làm việc với backend tốt.
- Tốn thời gian và chi phí cho việc phát triển, nâng cấp và vận hành.
- Có thể gặp vấn đề bảo mật khi hệ thống bị tấn công nếu không giới hạn điều kiện kỹ.

CHƯƠNG 4: TÌM HIỂU VỀ REACT

I. Khái niệm

1. React là gì

React hay còn gọi là Reactjs là thư viện JavaScript phổ biến nhất hiện nay để xây dựng giao diện người dùng. Tốc độ xử lý tuyệt vời khi user nhập liệu bằng cách sử dụng các phương pháp mới để render ra trang web.

React được phát triển bởi Facebook. Được ra mắt vào năm 2013 như một công cụ mã nguồn mở JavaScript. Hiện tại ngôn ngữ này đã vượt mặt các đối thủ khác như Angular, Bootstrap.

2. Tại sao phải sử dụng React ?

React được sử dụng hàng trăm công ty lớn trên thế giới: Netflix, Facebook, WhatsApp, Instagram... Với sự sử dụng rộng rãi như vậy cho thấy ngôn ngữ này

- Dễ sử dụng
 - React là một thư viện GUI nguồn mở JavaScript tập trung và hoàn thành nhiệm vụ UI một cách hiệu quả. Nó chỉ có “V” trong mô hình MVC (Model-View-Controller)
 - Đối với những lập trình viên hiểu biết về JavaScript thì sẽ dễ dàng tạo một ứng dụng với react trong thời gian ngắn.
- Hỗ trợ Reusable Component trong java
 - React cho phép sử dụng lại các components đã được phát triển thành các ứng dụng khác có cùng chức năng. Đây là một lợi thế cho các lập trình viên tái sử dụng lại các components.
- Viết các Components dễ dàng
 - React component dễ viết vì nó sử dụng JSX, mở rộng cú pháp tùy chọn cho sự kết hợp giữa HTML và JavaScript. Nó làm rõ toàn bộ quá trình viết cấu trúc của trang web. Ngoài ra, nó sẽ giúp render nhiều sự lựa chọn dễ dàng hơn.
- Hiệu suất tốt hơn với Virtual DOM

- Trong quá trình phát triển tạo dựng dự án. Nếu có sự cập nhật trong quá trình phát triển dự án thì virtual sẽ thay đổi ngay lập tức. Với ưu thế này sẽ giúp cho việc tìm lỗi hoặc tốc độ của ứng dụng không bị dán đoạn.

3. Ưu và nhược điểm

Ưu điểm(giống phần tính dễ sử dụng phần 2)

Nhược điểm

- React nó chỉ có tầm View chứ nó không phải là một framework hoàn chỉnh có cả Model và Controller mà chúng phải kết hợp với các thư viện khác.
- Nếu tích hợp React vào các framework MVC thì yêu cầu phải cấu hình lại.
- Yêu cầu phải có kiến thức về JavaScript.

II. Cách cài đặt React trên môi trường linux (ubuntu)

1. Cách cài đặt môi trường

Tương tự như React native chúng ta đã cài 2 thư viện cần thiết là Nodejs và npm. Bật terminal gõ node -v và npm -v hoặc yarn -v



```
lihn@lihn-x556ur:~$ node -v
v12.14.1
lihn@lihn-x556ur:~$ npm -v
6.14.5
lihn@lihn-x556ur:~$ yarn -v
1.22.4
lihn@lihn-x556ur:~$ █
```

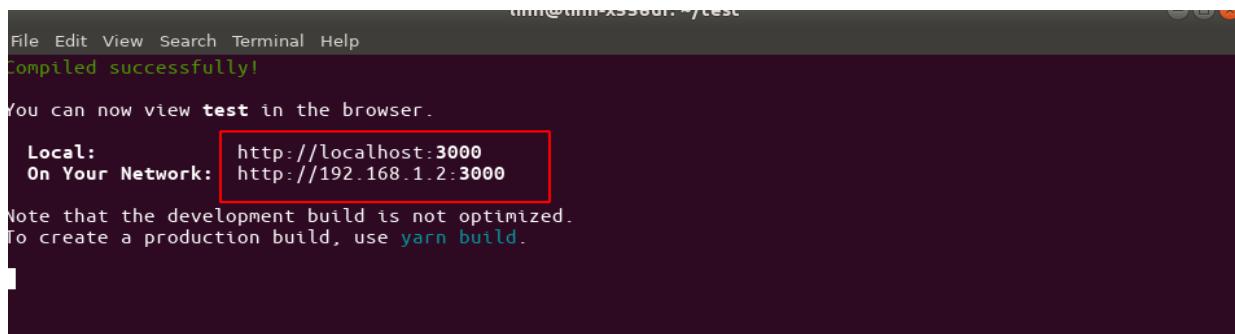
Hình 9: Kiểm tra node, npm và yarn version

- Khởi tạo dự án

```
success Uninstalled packages.  
Done in 7.77s.  
  
Success! Created test at /home/lihn/test  
Inside that directory, you can run several commands:  
  
  yarn start  
    Starts the development server.  
  
  yarn build  
    Bundles the app into static files for production.  
  
  yarn test  
    Starts the test runner.  
  
  yarn eject  
    Removes this tool and copies build dependencies, configuration files  
    and scripts into the app directory. If you do this, you can't go back!  
  
We suggest that you begin by typing:  
  
  cd test  
  yarn start  
  
Happy hacking!  
lihn@lihn-X556UR:~$
```

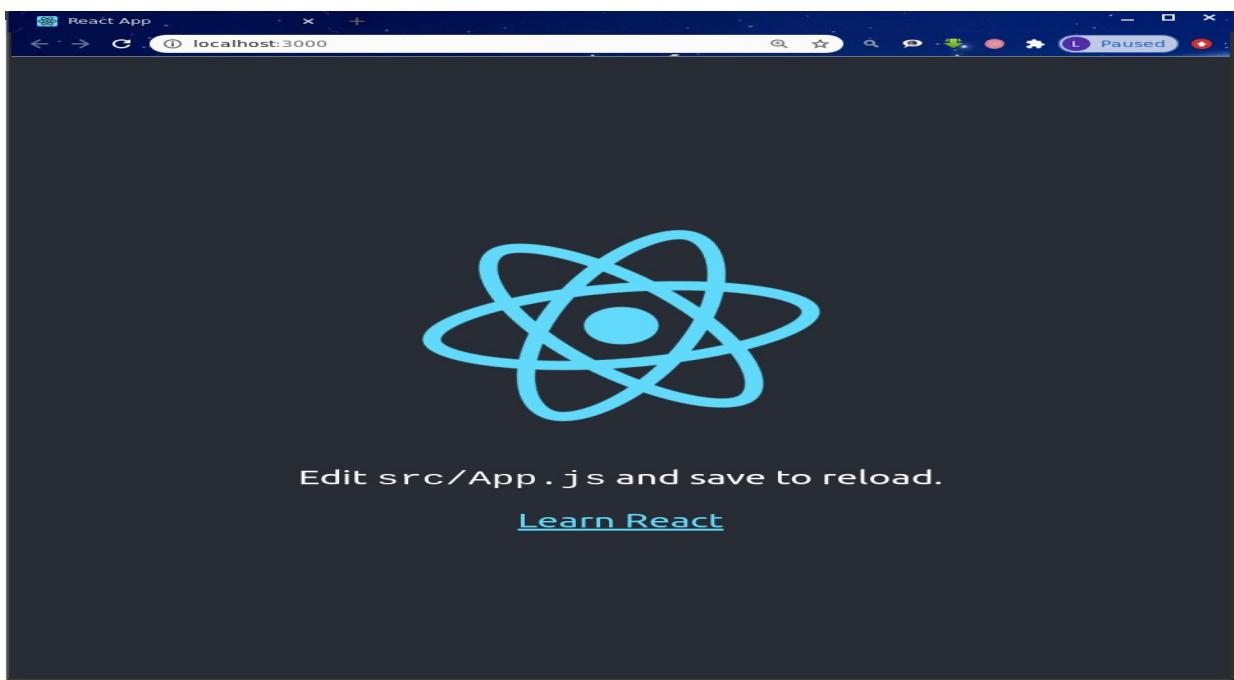
Hình 10: Tạo dự án mới với reactjs

- Tạo dự án test: npx create-react-app cd test
- yarn start



Hình 11: Chạy reactjs trên terminal

- Sau khi xong trên trình duyệt chúng ta sẽ thấy được dự án test đang



Hình 12: Chạy reactjs trên trình duyệt chrome

2. *Thư mục trong trong react*

```
lihn@lihn-x556ur:~/test$ ls
node_modules  package.json  public  README.md  src  yarn.lock
lihn@lihn-x556ur:~/test$
```

Hình 13: Cấu trúc thư mục reactjs

- src: Chứa những đoạn mã mà chúng ta viết hay gọi là các components.
- public: Chứa các file css, ảnh, js, ...
- node_modules: Các module cài tự động từ khi tạo react-app và những thư viện khác lúc bạn phát triển cài đặt.
- package.json và package-lock.json: Chứa thông tin các module cần thiết đã cài đặt.
- Ngoài ra còn các thư mục khác nhưng mà ta chỉ tập trung vào những phần chính.

CHƯƠNG 5: TÌM HIỂU VỀ IOT

I. Khái niệm

1. IoT là gì ?

Internet of Things hay IoT là một thiết bị vật lý được kết nối với internet thu nhập và chia sẻ dữ liệu. Nhờ bộ xử lý bên trong cùng mạng không dây, bạn có thể biến mọi thứ trở nên chủ động hơn và một cách thông minh hơn.

Chúng ta có thể bắt gặp IoT từ hệ thống của hàng tự động, xe tự động hay máy bay tự lái... rất nhiều và nó ngày càng phổ biến hơn và được áp dụng trong tất cả các lĩnh vực.

2. Tầm quan trọng của IoT

Khi bất cứ thiết bị nào kết nối được với internet có nghĩa là nó có thể chia sẻ và nhận thông tin hoặc ngược lại. Với việc chia sẻ và nhận thông tin nó sẽ giúp mọi thứ trở nên nhanh chóng và thông minh luôn là điều hướng đến trong thời kỳ công nghệ 4.0

Điện thoại thông minh là một điển hình thực tế nhất. Ngày bây giờ bạn có thể nghe bất kỳ bài hát nào trên khắp thế giới, không phải vì điện thoại bạn có thể lưu trữ được tất cả bài hát mà là những bài hát đó được lưu trữ ở một nơi khác. Điện thoại có thể gửi yêu cầu (yêu cầu bài hát) và sau đó thông tin sẽ được tiếp nhận và chuyển về cho điện thoại của bạn và có thể nghe được bài hát đó.

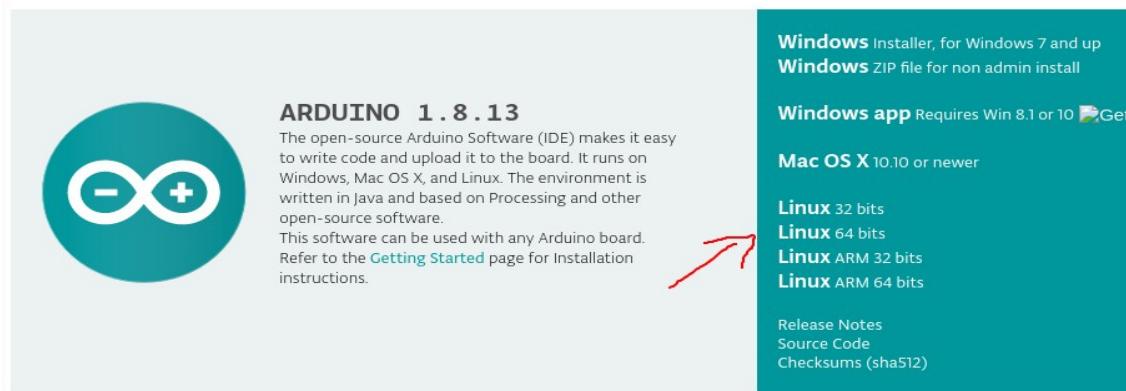
Internet of Things có thể chia thành 3 loại:

- Loại thu thập thông tin và sau đó gửi: Như các thiết bị cảm ứng nhiệt độ, cảm biến chuyển động, cảm biến ánh sáng ... Những thiết bị này đã được kết nối với internet, sau khi thu nhập thông tin từ môi trường nó sẽ gửi đến một nơi nào đó và sau đó show ra kết quả cho chúng ta.
- Loại nhận được thông tin và sau đó hành động: Như máy in. Sau khi tiếp nhận thông tin từ thiết bị khác như máy tính thì nó sẽ tự thực hiện hành động in.
- Thực hiện cả hai: Ví dụ trong ngành nông nghiệp. Việc sử dụng cảm biến nhiệt độ, độ ẩm tự động và việc tự động điều khiển tưới nước là một hành động thông minh và tiết kiệm thời gian cho người nông dân và tăng chi phí thu hoạch.

3. Cài đặt

- a. Cài đặt môi trường làm việc

- Lên trang chủ arduino để download về cho linux



Hình 14: Cách tải Arduino trên trang chủ Arduino

- Sau khi download bạn vào thư mục Downloads và giải nén.
- Bạn mở terminal
- Gõ cd/Downloads/arduino-1.X.X-linuxYY
- Gõ cd arduino-1.X.X
- Gõ sudo ./install.sh
- Sau khi cài xong bạn mở vào ứng dụng sẽ thấy Arduino IDE

PHẦN 2: NỘI DUNG THỰC HIỆN

CHƯƠNG 1: ỨNG DỤNG CHẠY TRÊN WEBSITE

I. Phân tích hệ thống

1. Xác định yêu cầu

- Chủ nhà trọ quản lý, theo dõi và điều khiển các thiết bị điện trong mỗi phòng trọ.
- Chủ nhà trọ theo dõi, quản lý quá trình thuê và vận hành các phòng trọ tại khu vực mình phụ trách.
- Người quản trị theo dõi, quản lý quá trình thuê và vận hành các phòng trọ tại tất cả các khu vực.

2. Đặc tả yêu cầu

a. Yêu cầu chức năng

- Khi khách hàng đến thuê phòng trọ thì người quản lý sẽ dẫn đi xem các phòng trống,
- Sau khi xem xong nếu muốn thuê phòng thì người quản lý sẽ thực hiện các bước sau:
 - Thêm khách hàng vừa đặt phòng.
 - Cập nhật dịch vụ tại phòng khách đang ở.
 - Cập nhật thành viên ở chung.
 - Cập nhật giá điện, giá nước của phòng.

b. Yêu cầu phi chức năng

- Website phải có dung lượng không quá lớn, tốc độ xử lý nhanh.
- Công việc tính toán phải chính xác, không chấp nhận sai sót.
- Thông tin khách hàng phải bảo mật, an toàn.
- Đảm bảo dữ liệu an toàn khi chạy trên server.
- Sử dụng Rails 6.0.3.2, Ruby 2.5.1.

3. Xây dựng biểu đồ chức năng hệ thống

a. Tác nhân

Có 2 tác nhân tham gia và tương tác với hệ thống: Quản trị viên và chủ trọ.

b. Chức năng hệ thống

Chủ trọ:

- Đăng nhập
- Đăng xuất

- Quản lý thiết bị: Bật, tắt và hẹn thời gian thiết bị.
- Quản lý:
 - Phòng: Chỉnh sửa phòng (giá, tên), xác nhận trả phòng.
 - Dịch vụ: Thêm, sửa, xóa dịch vụ.
 - Thành viên: Thêm, sửa, xóa thành viên ở chung với khách hàng.
 - Tài khoản khách hàng (dùng để đăng nhập điều khiển đèn trên app): mở hoặc khóa tài khoản.
 - Phản hồi từ khách hàng: Trả lời phản hồi từ khách hàng.
 - Thanh toán: Cập nhật thanh toán phòng hàng tháng.
 - Quy định của nhà trọ: thêm, sửa, xóa.
- Tạo nhắc nhở: Thêm, sửa, xóa nhắc nhở.
- Thống kê phòng và khách hàng phòng.
- Gửi mail cho khách hàng: Tại 1 khu vực.
- Gửi mail tự động khi khách hàng chưa đóng
- Điều hướng web + điều khiển đèn bằng giọng nói.

Quản trị viên:

- Đăng nhập
- Đăng xuất
- Quản lý thiết bị: Bật, tắt và hẹn giờ thiết bị tắt cả các khu vực.
- Quản lý:
 - Nhà: Thêm, sửa, xóa
 - Phòng: Thêm một hay nhiều phòng cùng một lúc, sửa, xóa
 - Dịch vụ: Thêm, sửa, xóa
 - Tài khoản: Thêm người quản lý từng khu vực.
 - Phản hồi: Tất cả các khu vực.
 - Thanh toán: Tất cả các khu vực.
 - Gửi mail cho khách hàng: Tất cả các khu vực.

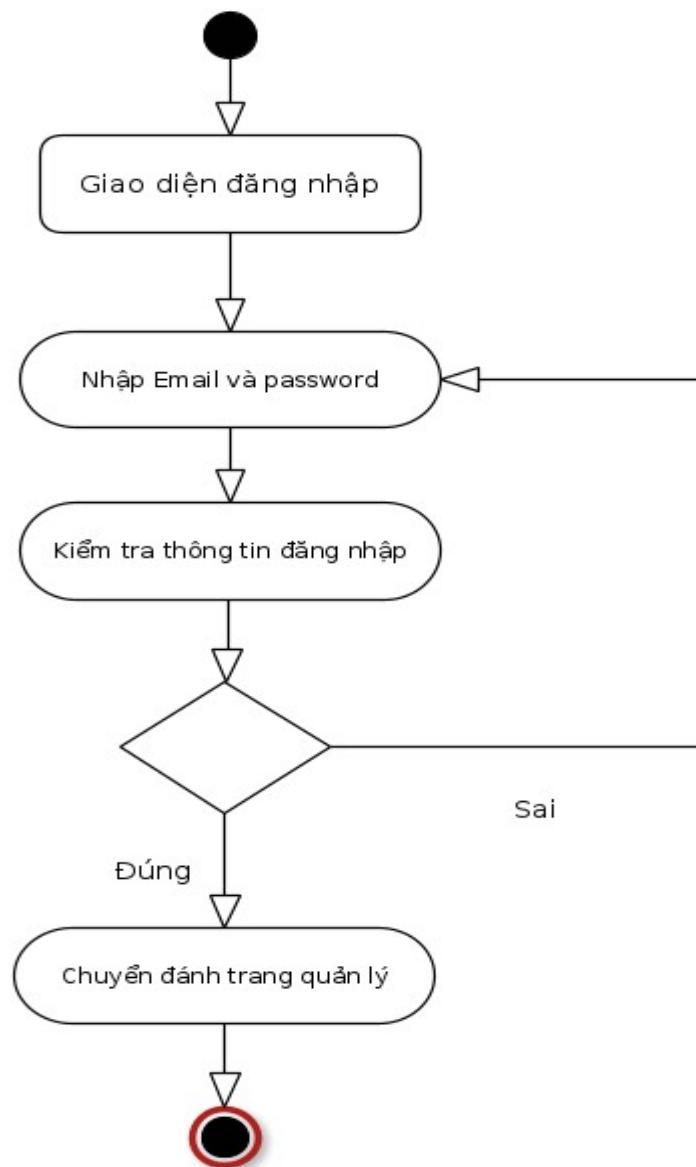
- Thống kê: Tiền trọ từng tháng, tất cả khách hàng, phòng trống hay không trống.
 - Điều hướng web + điều khiển đèn bằng giọng nói.
 - Slides chạy trên app: thêm, sửa xóa
- Từ việc phân tích các đối tượng ở trên ta có mô hình phân ra chức năng như hình 1



4. Biểu đồ hoạt động

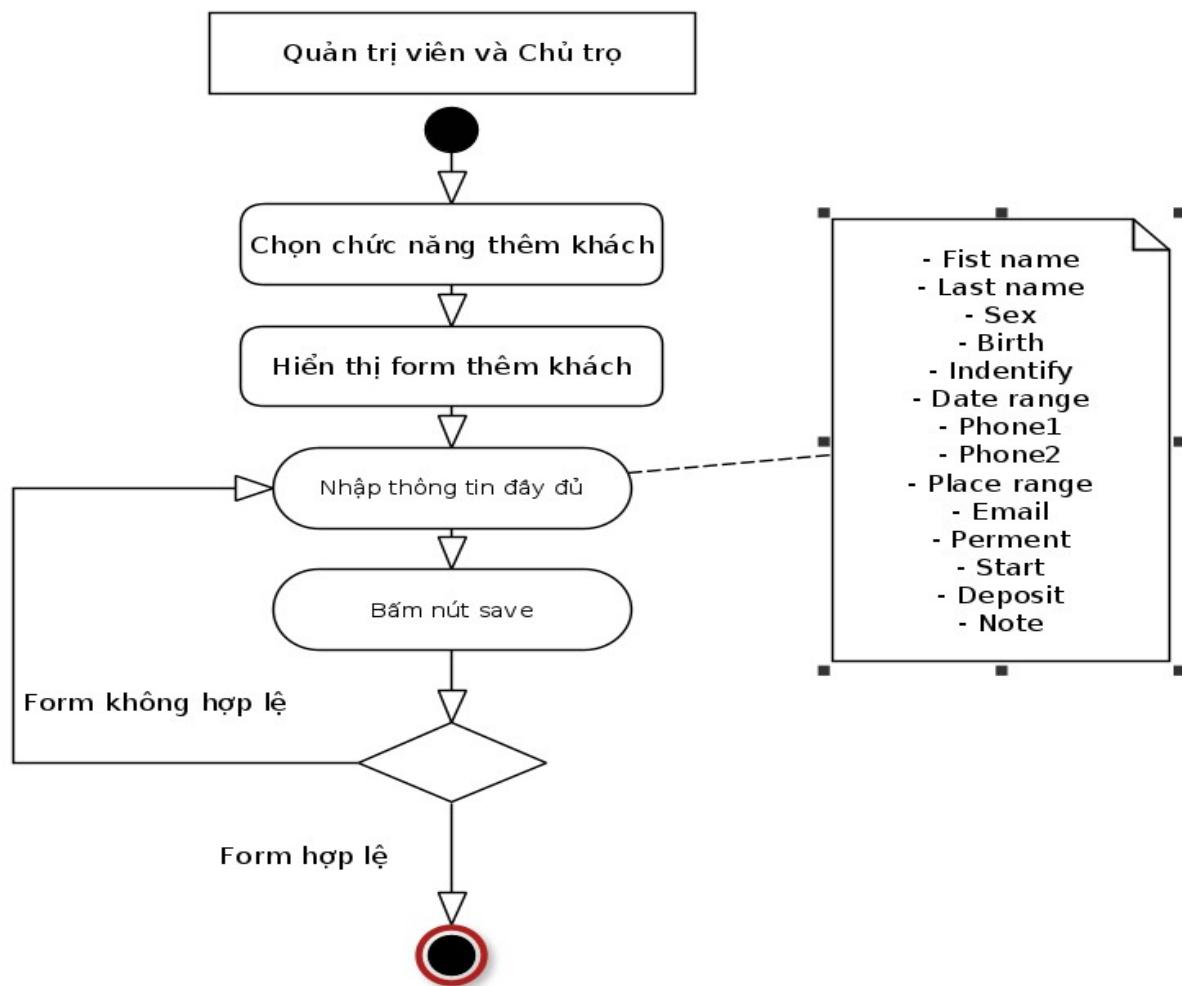
- a. Biểu đồ hoạt động cho chức năng đăng nhập

Để thực hiện trong việc quản lý nhà trọ thì chủ trọ cũng như là quản trị viên phải đăng nhập vào hệ thống.



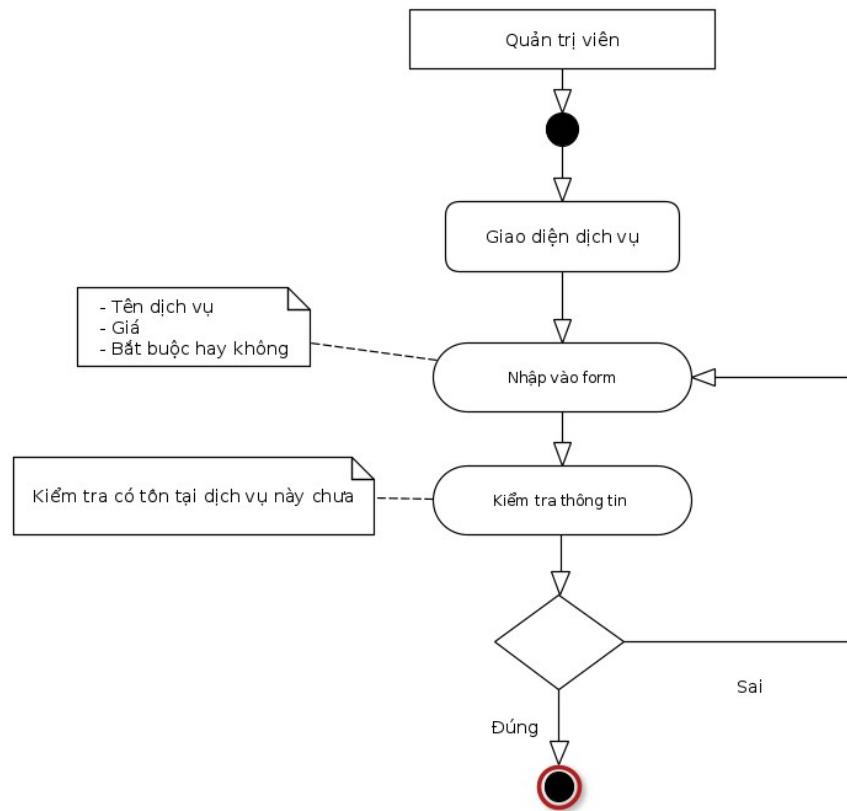
Hình 16: Biểu đồ hoạt động đăng nhập của quản trị viên và chủ trọ

b. Biểu đồ cho chức năng thêm một khách hàng mới



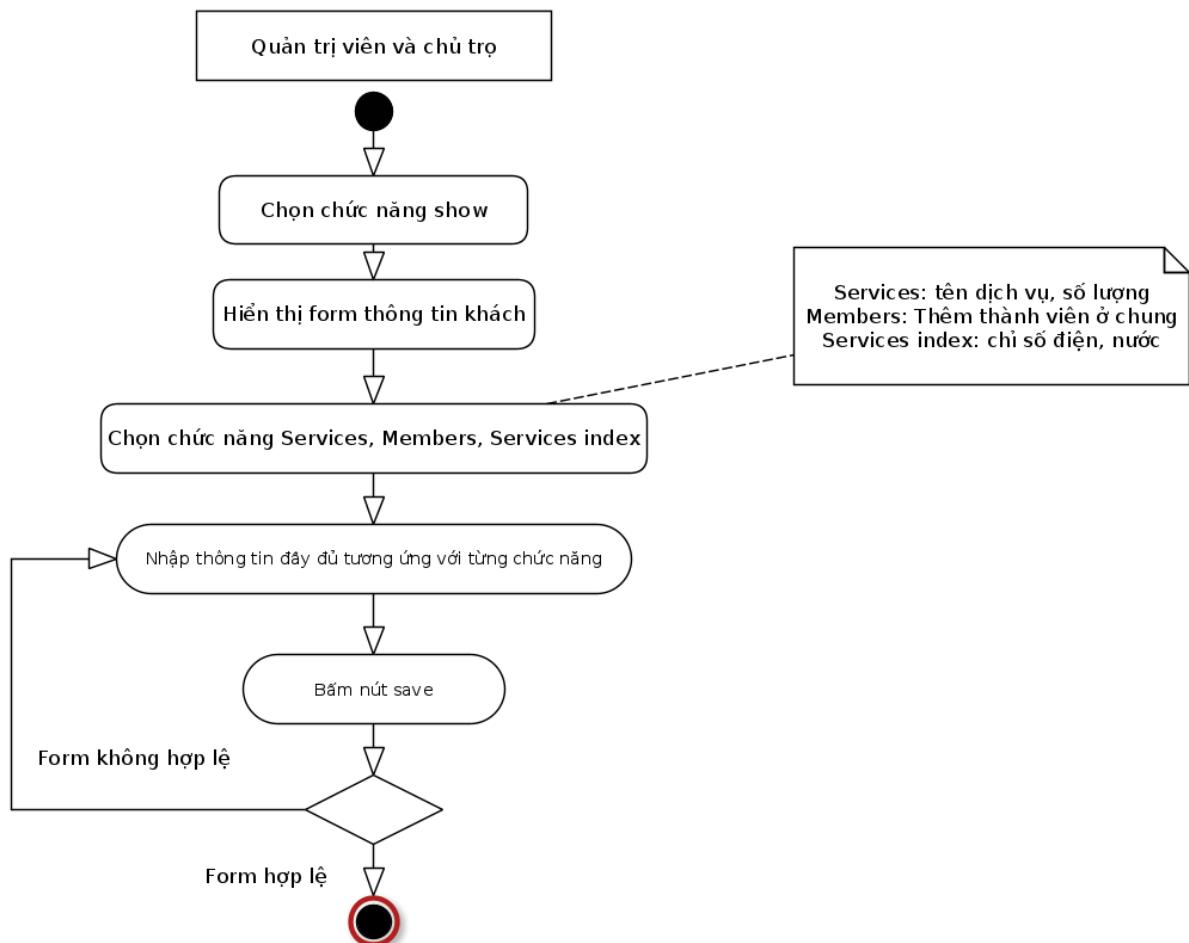
Hình 17: Biểu đồ thêm một khách hàng mới

c. Biểu đồ cho chức năng thêm mới dịch vụ



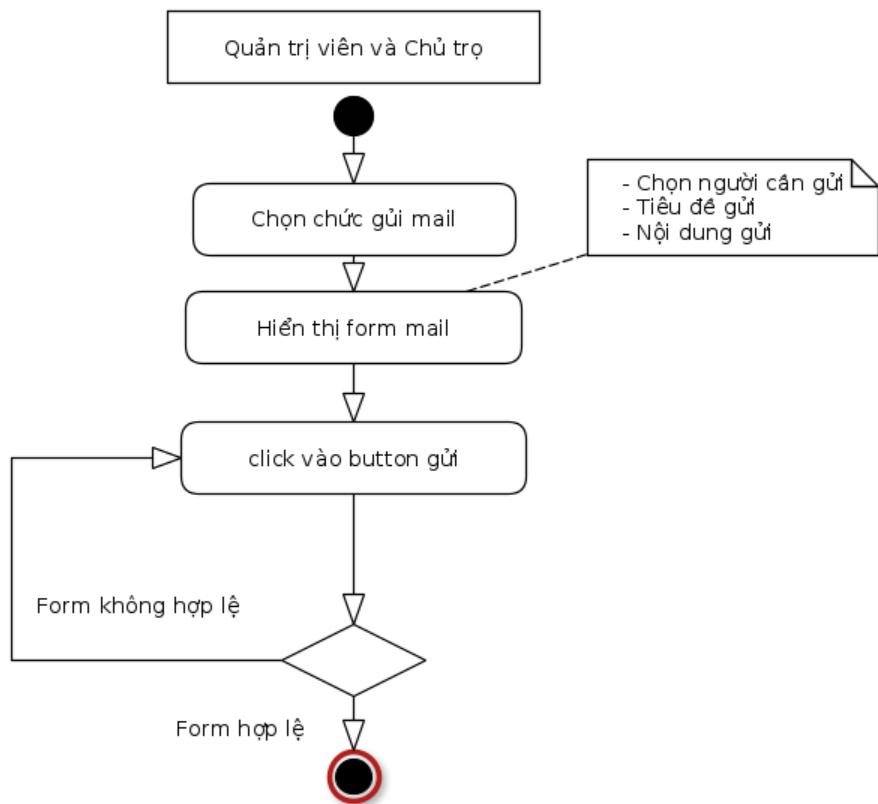
Hình 18: Biểu đồ hoạt động thêm dịch vụ mới

d. Biểu đồ cho chức năng cập nhật dịch vụ, thành viên, chỉ số dịch vụ



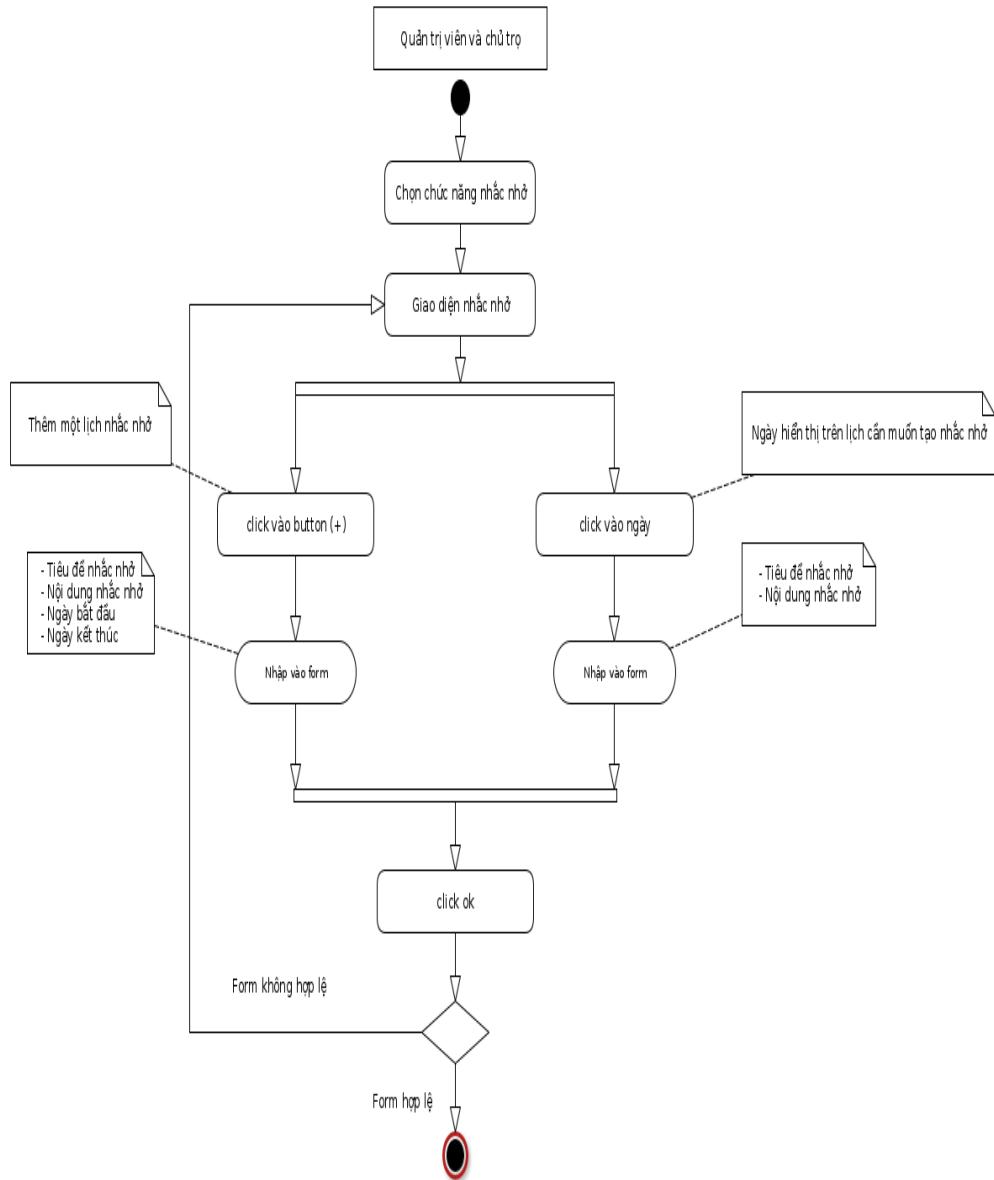
Hình 19: Biểu đồ hoạt động cập nhật dịch vụ, khách hàng chung phòng và chỉ số dịch vụ

e. Biểu đồ cho chức năng gửi mail



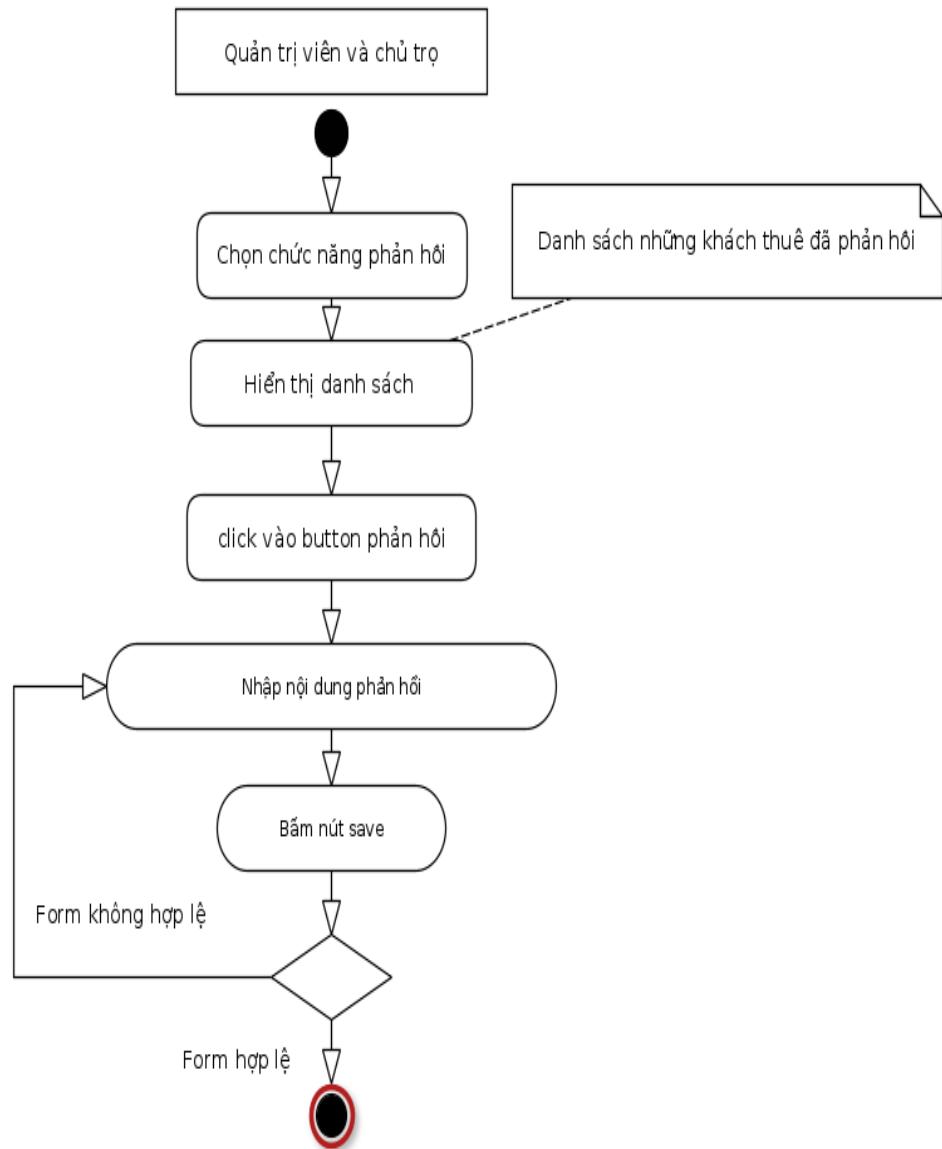
Hình 20: Biểu đồ hoạt động gửi email

f. Biểu đồ cho chức năng thêm mới một nhắc nhở



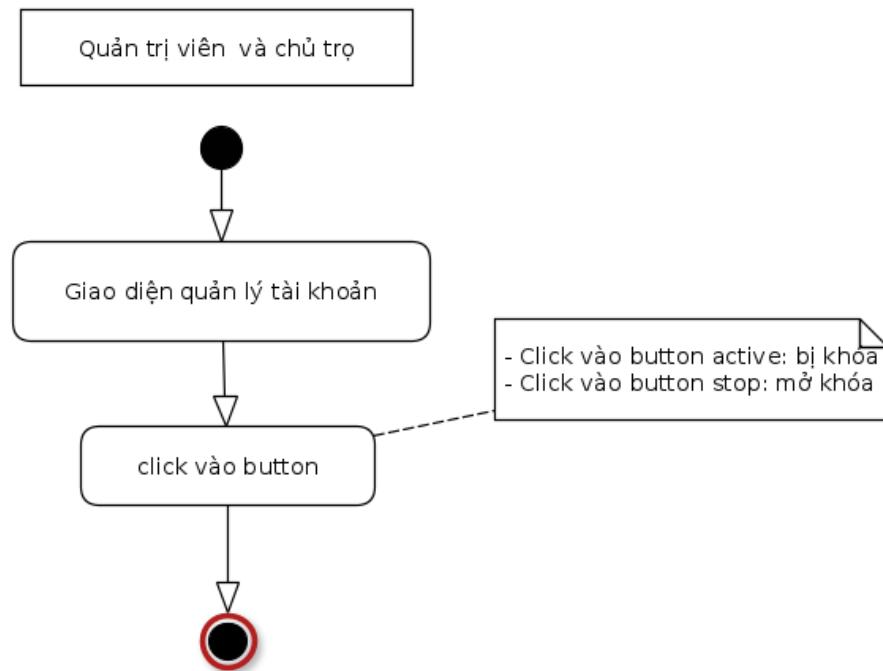
Hình 21: Biểu đồ hoạt động tạo nhắc nhở

g. Biểu đồ cho chức năng phản hồi ý kiến khách hàng qua từ bên app hoặc web



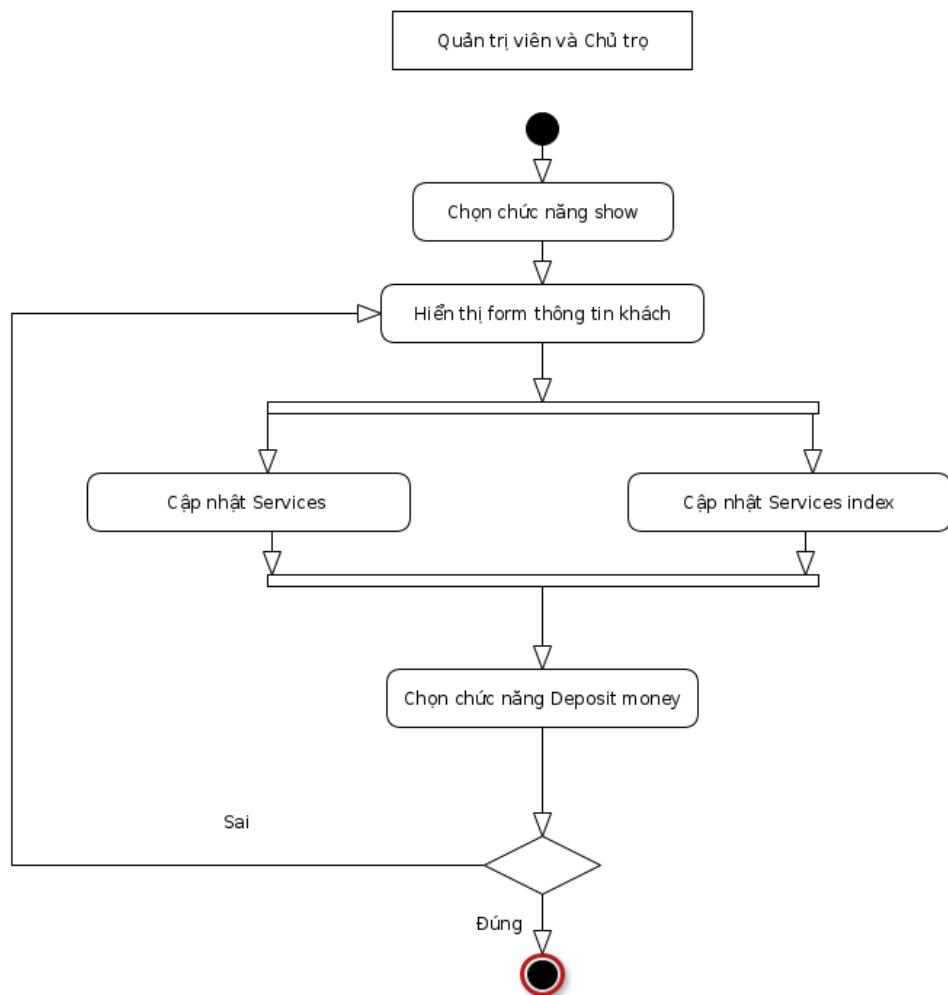
Hình 22: Biểu đồ hoạt động gửi phản hồi

h. Biểu đồ cho chức năng quản lý tài khoản



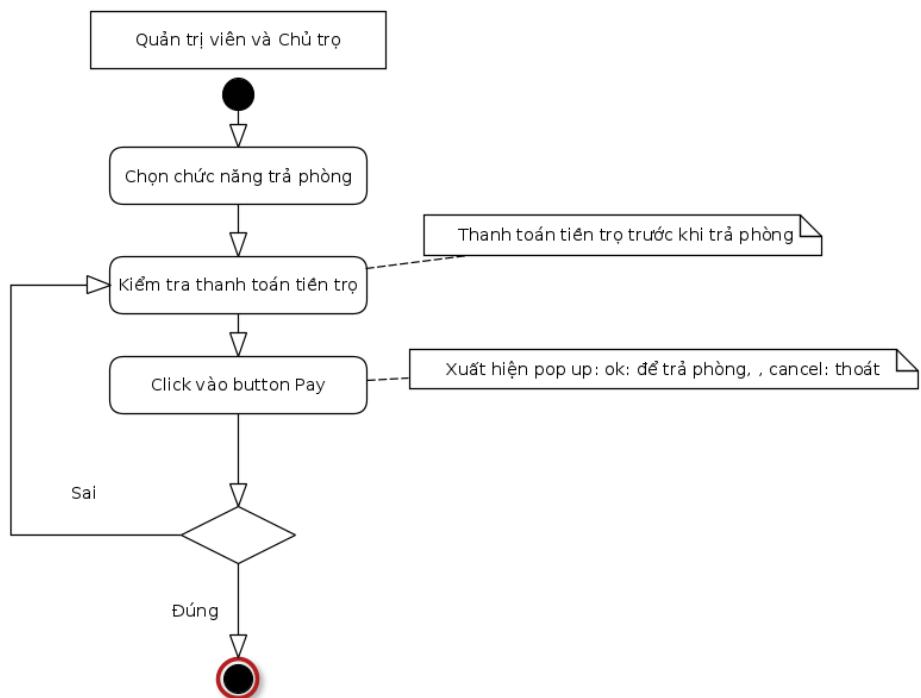
Hình 23: Biểu đồ hoạt động quản lý tài khoản

I. Biểu đồ cho chức năng thanh toán tiền trọ qua mail



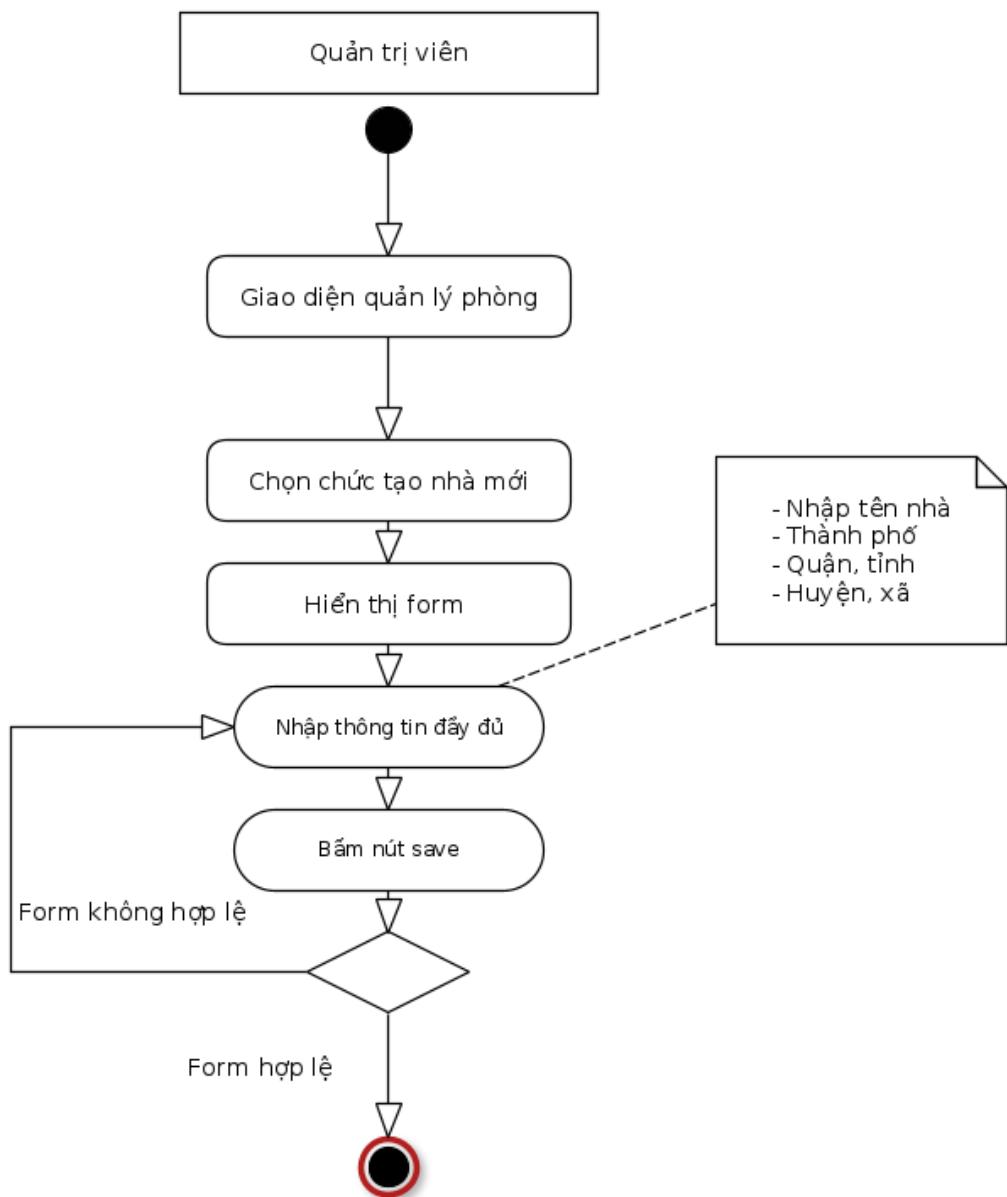
Hình 24: Biểu đồ hoạt động thanh toán tiền trọ qua mail

j. Biểu đồ cho chức năng trả phòng



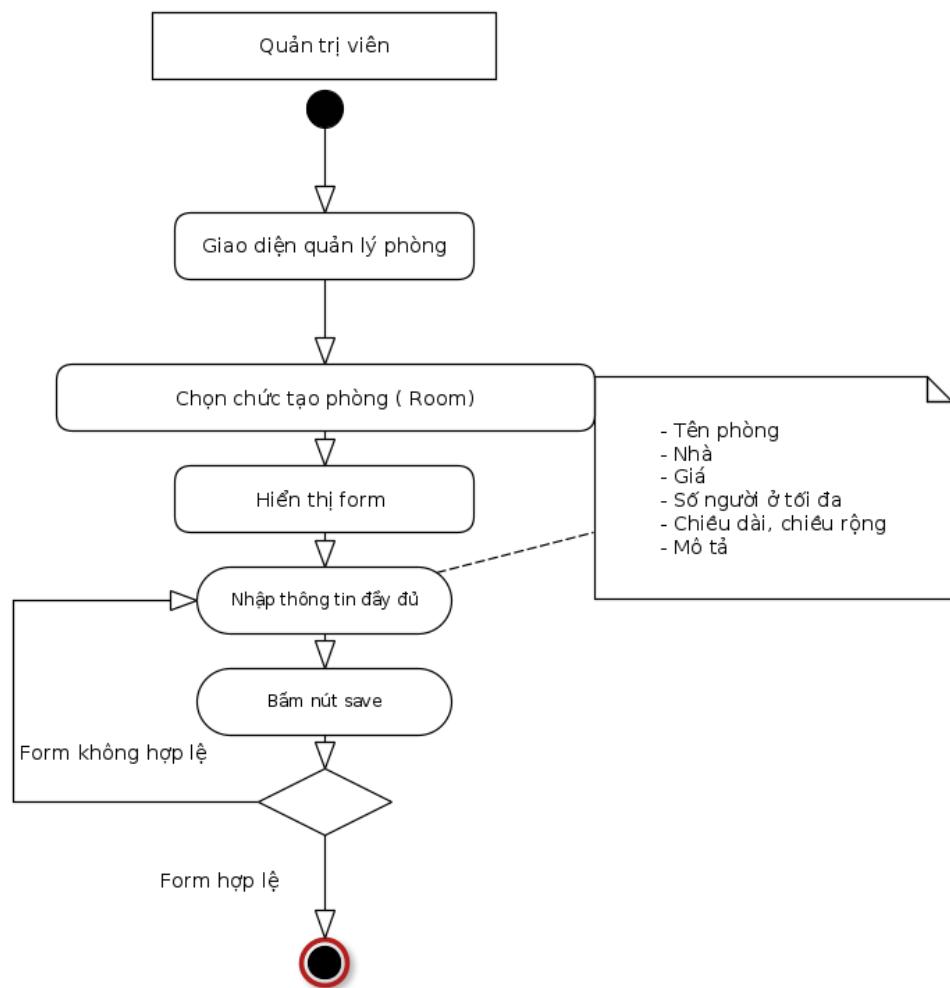
Hình 25: Biểu đồ hoạt động trả phòng

k. Biểu đồ cho chức năng thêm mới một nhà



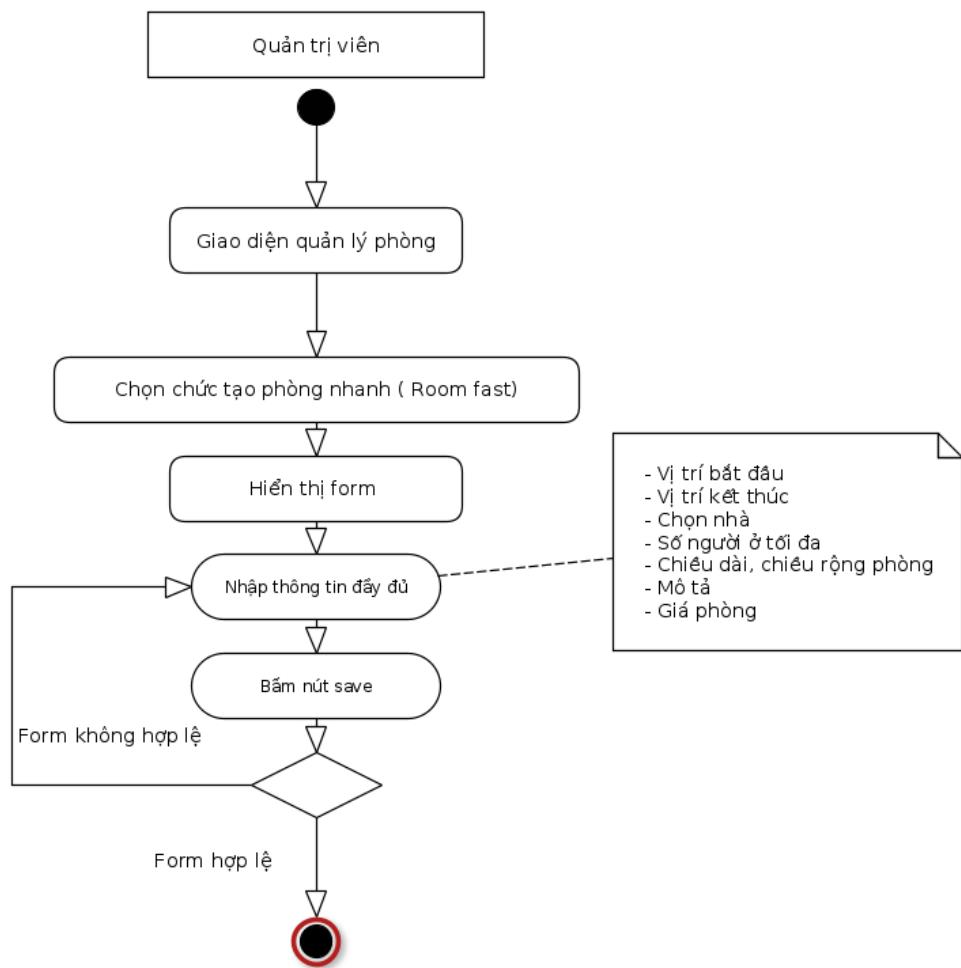
Hình 26: Biểu đồ hoạt động thêm mới một nhà

I. Biểu đồ cho chức năng thêm mới phòng nhanh



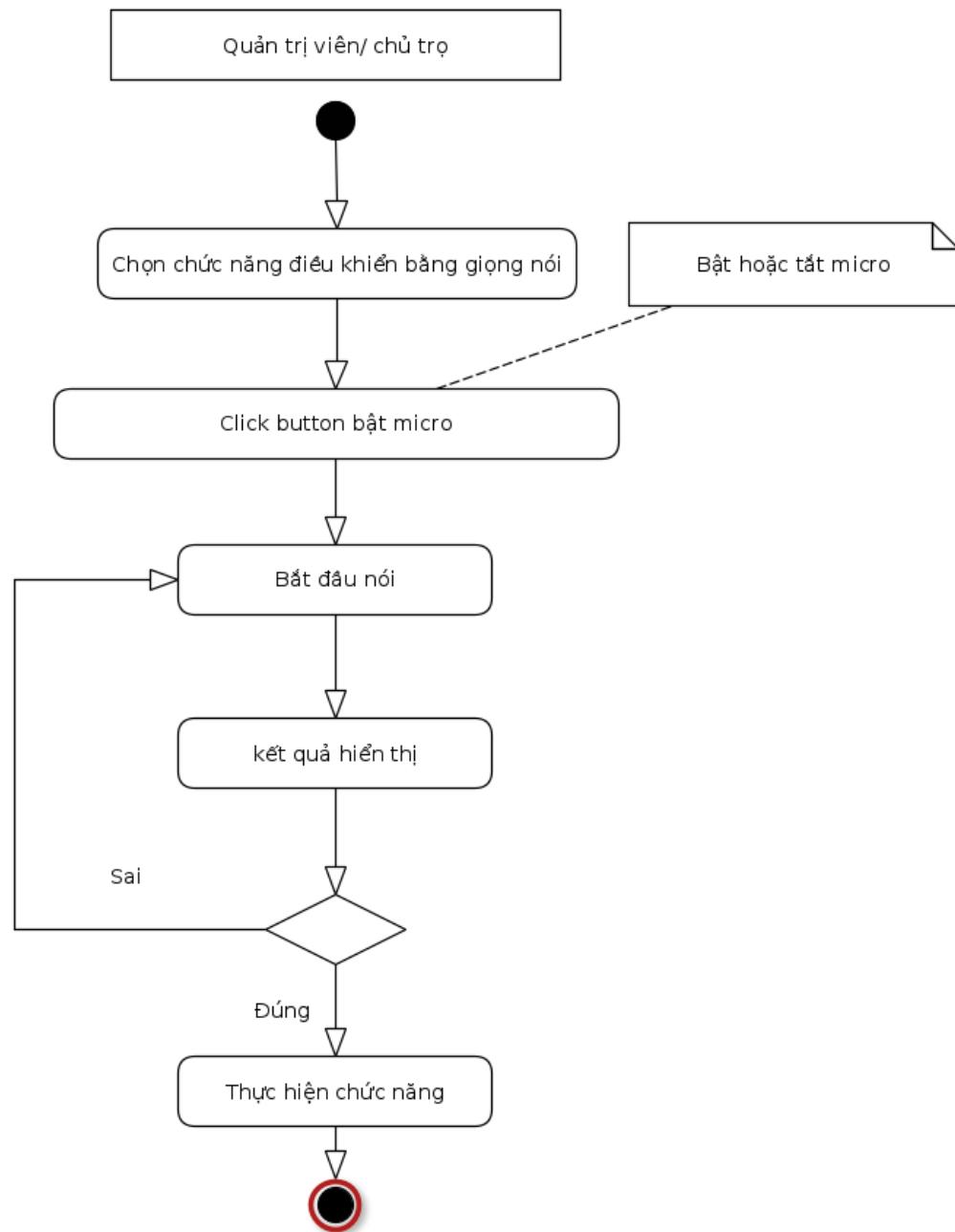
Hình 27: Biểu đồ hoạt động thêm phòng nhanh

m. Biểu đồ cho chức năng thêm mới một phòng



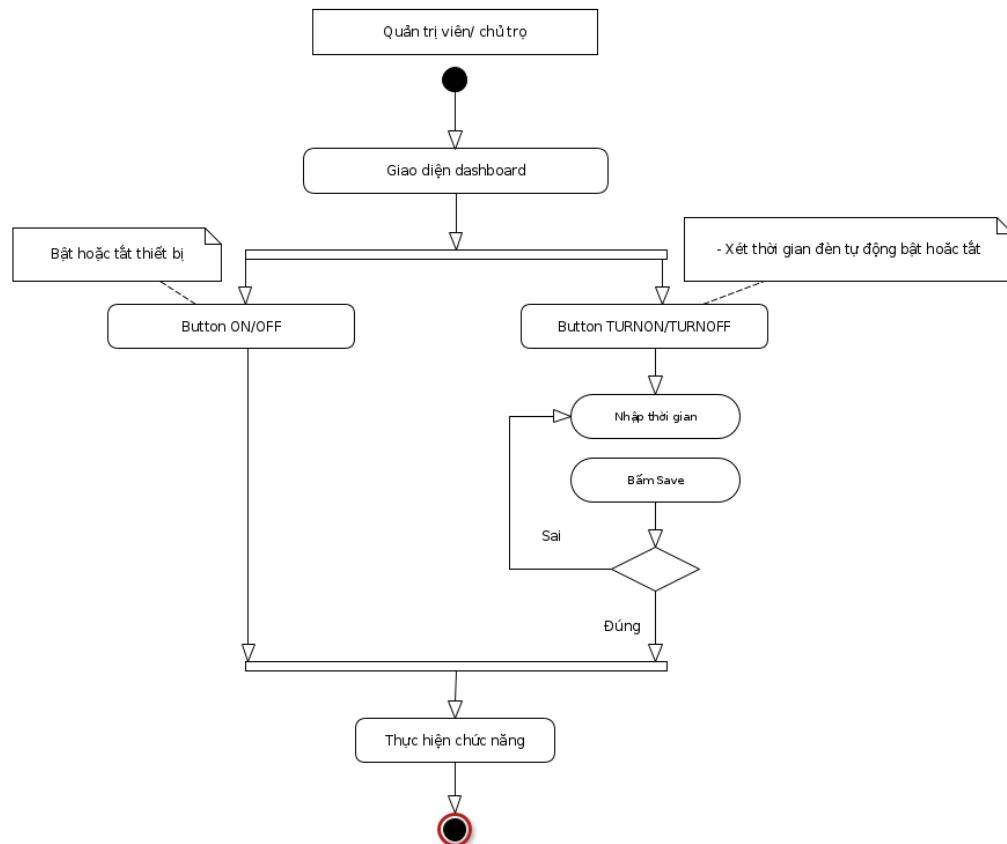
Hình 28: Biểu đồ hoạt động thêm mới phòng

n. Biểu đồ cho chức năng điều khiển bằng giọng nói



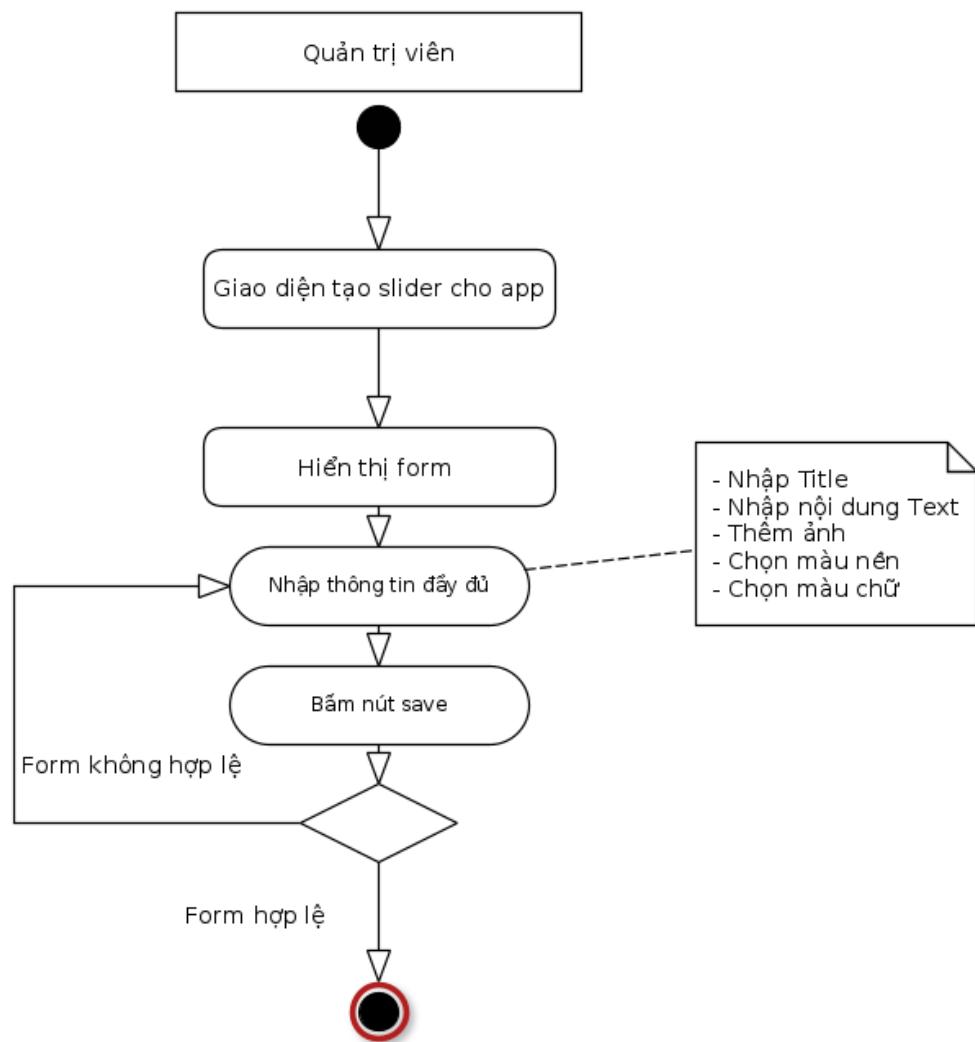
Hình 29: Biểu đồ hoạt động điều khiển bằng giọng nói

o. Biểu đồ chức năng điều khiển thiết bị trong phòng



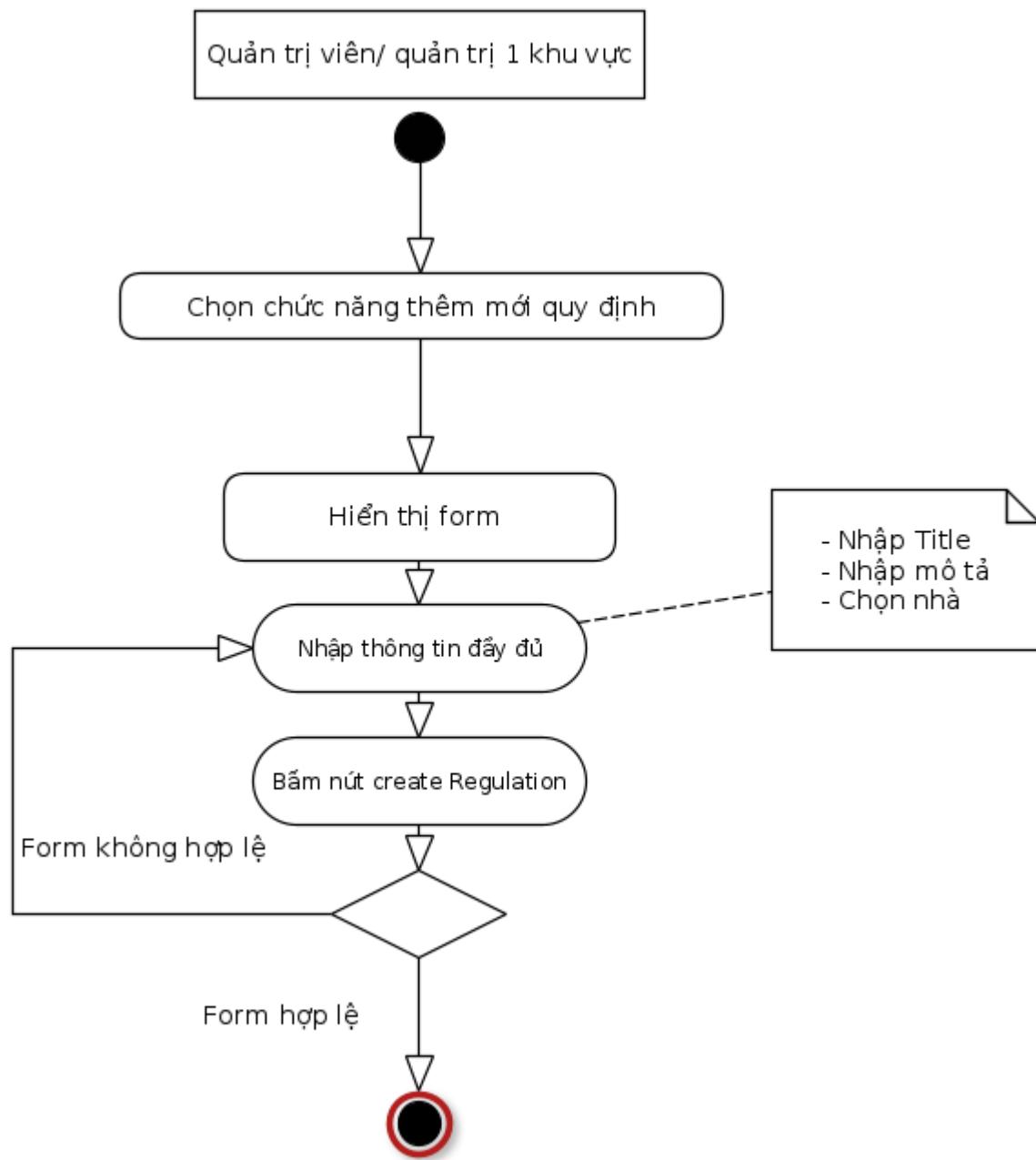
Hình 30: Biểu đồ điều khiển thiết bị trong phòng

p. Biểu đồ chức năng tạo slider cho app và website khách hàng



Hình 31: Biểu đồ hoạt động tạo slider cho app và website khách hàng

q. Biểu đồ chức năng tạo một quy định



Hình 32: Biểu đồ hoạt động tạo một quy định mới

5. Xây dựng biểu đồ thực thể

a. Bảng thực thể

Bảng 1: Users

Thuộc tính	Kiểu dữ liệu
id	INTEGER
email	VAR_CHAR
encrypted_password	VAR_CHAR
reset_passwork_token	VAR_CHAR
reset_password_sent_at	DATETIME
remembder_created_at	DATETIME
created_at	DATETIME
updated_at	DATETIME
disable	INTEGER
admin	INTEGER
house_id	INTEGER

Bảng 2: Information

Thuộc tính	Kiểu dữ liệu
id	INTEGER
name	VAR_CHAR
sex	INTEGER
birth	DATE
indentifycard	VAR_CHAR
daterange	VAR_CHAR
placerange	VAR_CHAR
phone1	VAR_CHAR
phone2	VAR_CHAR
start	DATE
deposit	FLOAT
Email	VAR_CHAR
mark	INTEGER

created_at	DATETIME
updated_at	DATETIME

Bảng 3: Houses

Thuộc tính	Kiểu dữ liệu
id	INTEGER
name	VAR_CHAR
city_id	INTEGER
district_id	INTEGER
ward_id	INTEGER
created_at	DATETIME
updated_at	DATETIME

Bảng 4: Cities

Thuộc tính	Kiểu dữ liệu
id	INTEGER
name	VAR_CHAR
kind	VAR_CHAR
created_at	DATETIME
updated_at	DATETIME

Bảng 5: Districts

Thuộc tính	Kiểu dữ liệu
id	INTEGER
name	VAR_CHAR
kind	VAR_CHAR
created_at	DATETIME
updated_at	DATETIME

Bảng 6: Wards

Thuộc tính	Kiểu dữ liệu
id	INTEGER
name	VAR_CHAR
kind	VAR_CHAR
created_at	DATETIME
updated_at	DATETIME

Bảng 7: Rooms

Thuộc tính	Kiểu dữ liệu
id	INTEGER
name	VAR_CHAR
cost	FLOAT
length	FLOAT
width	FLOAT
amount	INTEGER
allow	INTEGER
description	VAR_CHAR
house_id	INTEGER
picture	VAR_CHAR
information_id	INTEGER
mark	INTEGER
oldelectric	FLOAT
oldwater	FLOAT
newelectric	FLOAT
newwater	FLOAT
created_at	DATETIME
updated_at	DATETIME

Bảng 8: Services

Thuộc tính	Kiểu dữ liệu
id	INTEGER
name	VAR_CHAR
cost	FLOAT
created_at	DATETIME
updated_at	DATETIME
amount	INTEGER

Bảng 9: Use_services

Thuộc tính	Kiểu dữ liệu
id	INTEGER
service_id	JSON
amount	JSON
created_at	DATETIME
updated_at	DATETIME
information_id	INTEGER

Bảng 10: Members

Thuộc tính	Kiểu dữ liệu
id	INTEGER
name	JSON
sex	JSON
indentifycard	JSON
birth	JSON
created_at	DATETIME
updated_at	DATETIME
information_id	INTEGER

Bảng 11: Paytherents

Thuộc tính	Kiểu dữ liệu

<code>id</code>	INTEGER
<code>senddate</code>	VAR_CHAR
<code>receivedate</code>	DATETIME
<code>status</code>	INTEGER
<code>created_at</code>	DATETIME
<code>updated_at</code>	DATETIME
<code>information_id</code>	INTEGER

Bảng 12: Reminders

Thuộc tính	Kiểu dữ liệu
<code>id</code>	INTEGER
<code>title</code>	VAR_CHAR
<code>content</code>	VAR_CHAR
<code>start_time</code>	DATETIME
<code>end_time</code>	DATETIME
<code>created_at</code>	DATETIME
<code>updated_at</code>	DATETIME
<code>mark</code>	INTEGER
<code>user_id</code>	INTEGER

Bảng 13: Reports

Thuộc tính	Kiểu dữ liệu
<code>id</code>	INTEGER
<code>title</code>	VAR_CHAR
<code>content</code>	VAR_CHAR
<code>rep_content</code>	VAR_CHAR
<code>created_at</code>	DATETIME
<code>updated_at</code>	DATETIME
<code>mark</code>	INTEGER
<code>user_id</code>	INTEGER

information_id	INTEGER
----------------	---------

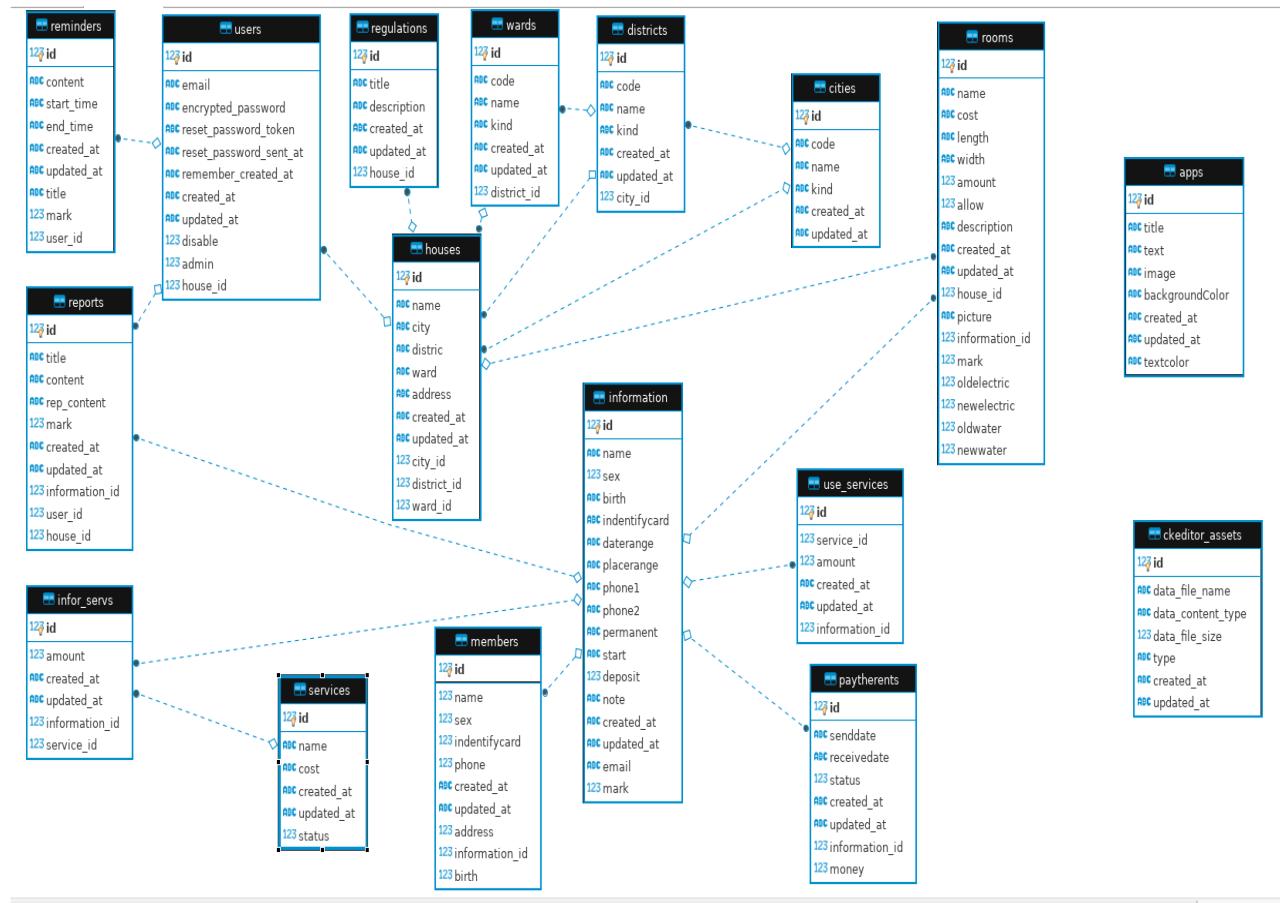
Bảng 14: Regulations

Thuộc tính	Kiểu dữ liệu
id	INTEGER
title	VAR_CHAR
description	VAR_CHAR
created_at	DATETIME
updated_at	DATETIME
house_id	INTEGER

Bảng 15: Apps

Thuộc tính	Kiểu dữ liệu
id	INTEGER
title	VAR_CHAR
text	VAR_CHAR
image	VAR_CHAR
backgroundColor	VAR_CHAR
textcolor	VAR_CHAR
created_at	DATETIME
updated_at	DATETIME

Biểu đồ lớp



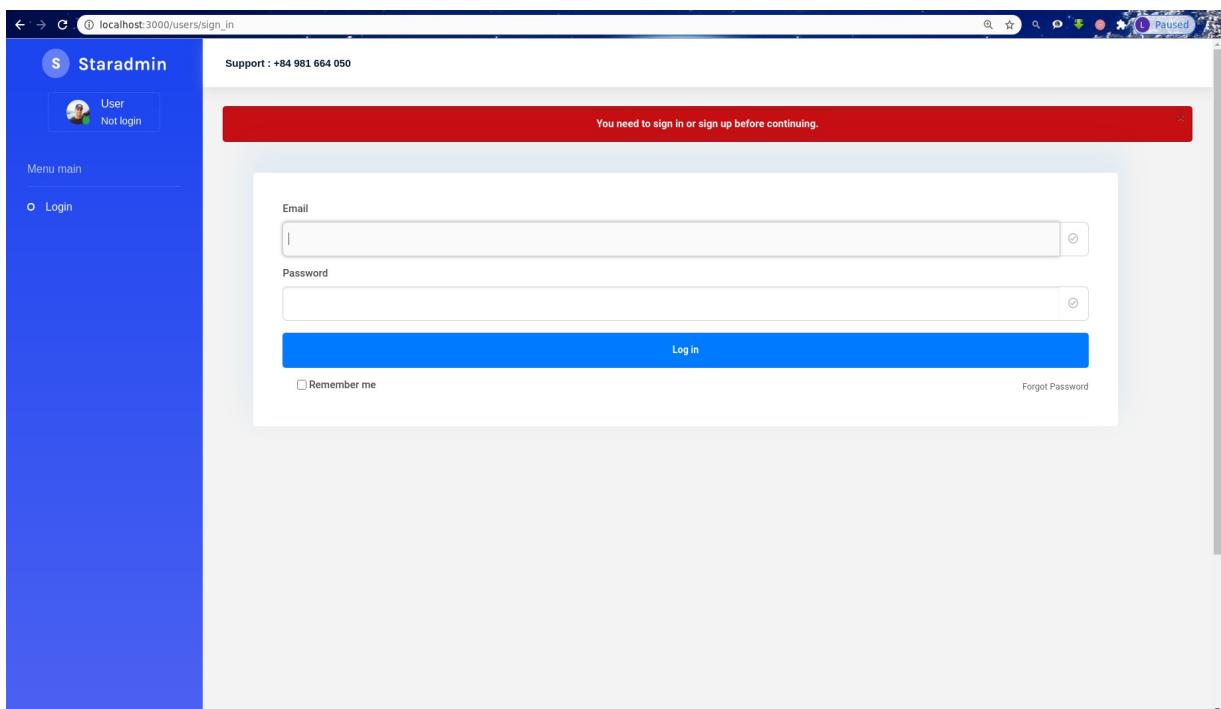
Hình 33: Biểu đồ lớp

II. Thiết kế hệ thống

1. Thiết kế cơ sở dữ liệu

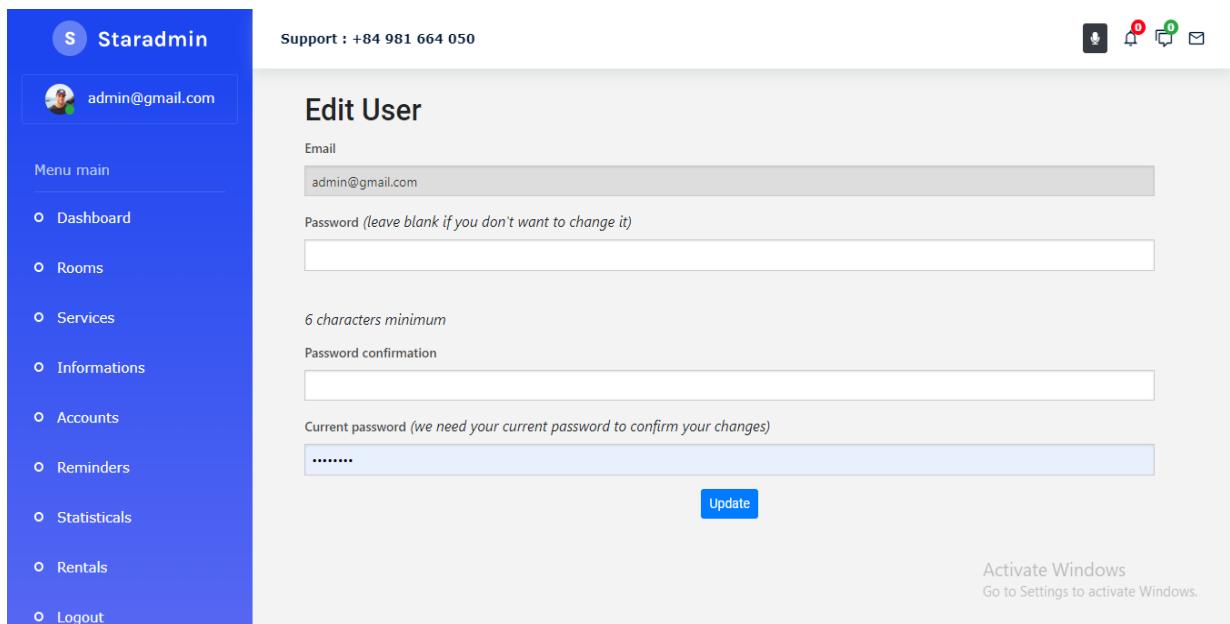
a. Thiết kế giao diện

Trang đăng nhập dành cho quản lý một khu vực và quản lý tất cả khu vực. Muốn sử dụng chức năng của hệ thống thì bắt buộc phải đăng nhập.



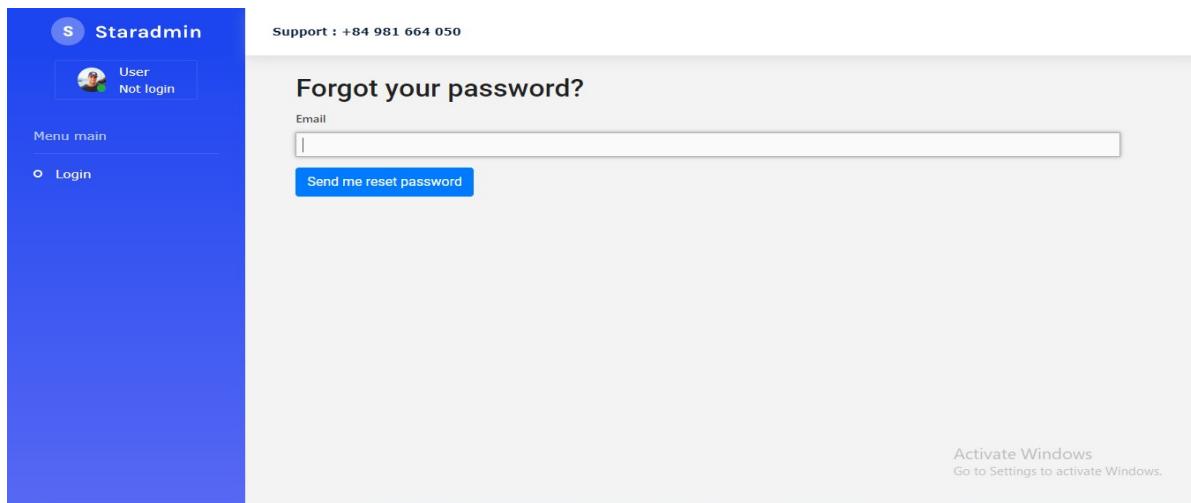
Hình 34: Giao diện đăng nhập

Trang Change Password: Người đăng nhập vào hệ thống có thể thay đổi mật khẩu hiện tại của mình.



Hình 35: Giao diện thay đổi mật khẩu

Trang Fotget Password: Nếu như đăng nhập vào hệ thống báo lỗi về mật khẩu thì tại trang fotget password sẽ cho phép bạn quên mật khẩu và reset lại mật khẩu sẽ được gửi qua mail.

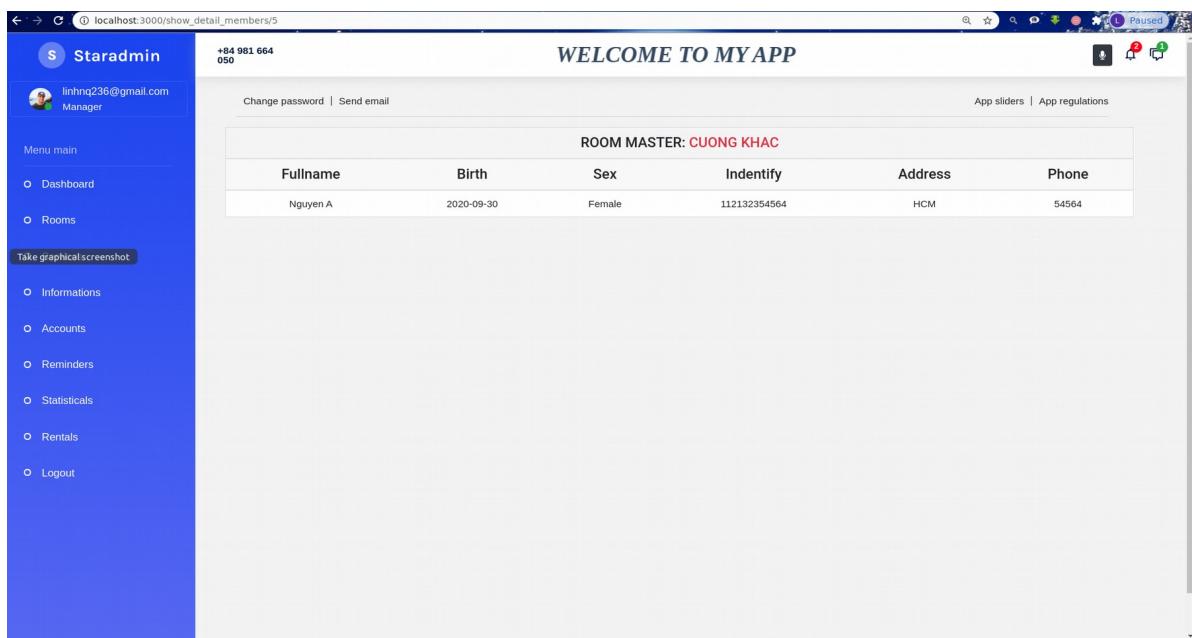


Hình 36: Giao diện quên mật khẩu

Trang Show - thông tin khách hàng đã đặt phòng: Những khách hàng đã đang sử dụng phòng tại mỗi khu vực thì sẽ có được hiển thị tất cả đối với mỗi khu vực đó. Còn người quản lý sẽ nhìn thấy hết tất cả các khu vực mà khách hàng đang thuê.

Hình 37: Giao diện hiển thị thông tin khách hàng đặt phòng

Thông tin khách hàng cùng phòng: Khi thuê phòng có những người ở cùng phòng thì trang này sẽ cho người quản lý biết phòng đang thuê này bao gồm những ai, ai là chủ phòng này.



Hình 38: Giao diện khách hàng cùng phòng

Trang thêm một khách hàng mới: Chủ trọ sẽ lấy thông tin khách hàng để cập nhật vào phòng.

Hình 39: Giao diện thêm một khách hàng mới

Trang New - thêm một khách hàng mới: Người quản lý từng khu vực khi có khách hàng mới đến thuê thì sẽ lập thông tin người đăng ký tại trang này. Cập nhật dịch vụ: Sau khi khách hàng đã đăng ký xong thì yêu cầu khách hàng chọn dịch vụ sử dụng tại phòng mình đang ở.

Order	Service used	Cost (VND)	Quantum
1	water	12000	1
2	electricity	35000	1
3	Wifi	25000	1

Hình 40: Giao diện cập nhật dịch vụ

Thêm thành viên mới: Tại trang này, khách hàng sẽ đăng ký thêm những thành viên ở chung.

WELCOME TO MYAPP

Thêm Thành Viên

Back Save

Full name	Birth	Sex	Identify	Address	Phone
	mm/dd/yyyy	Male			
	mm/dd/yyyy	Male			
	mm/dd/yyyy	Male			
	mm/dd/yyyy	Male			

Luân văn tốt nghiệp © 2021

Trên con đường thành công không có dấu chân của kẻ lường biến ❤

Hình 41: Giao diện cập nhật khách hàng cùng phòng

Cập nhật chỉ số dịch vụ điện và nước: Sau khi khách hàng đã ở thì người quản lý sẽ cập nhật chỉ số điện, nước hiện tại và chỉ số điện, nước mới.

MYAPP

Back Save

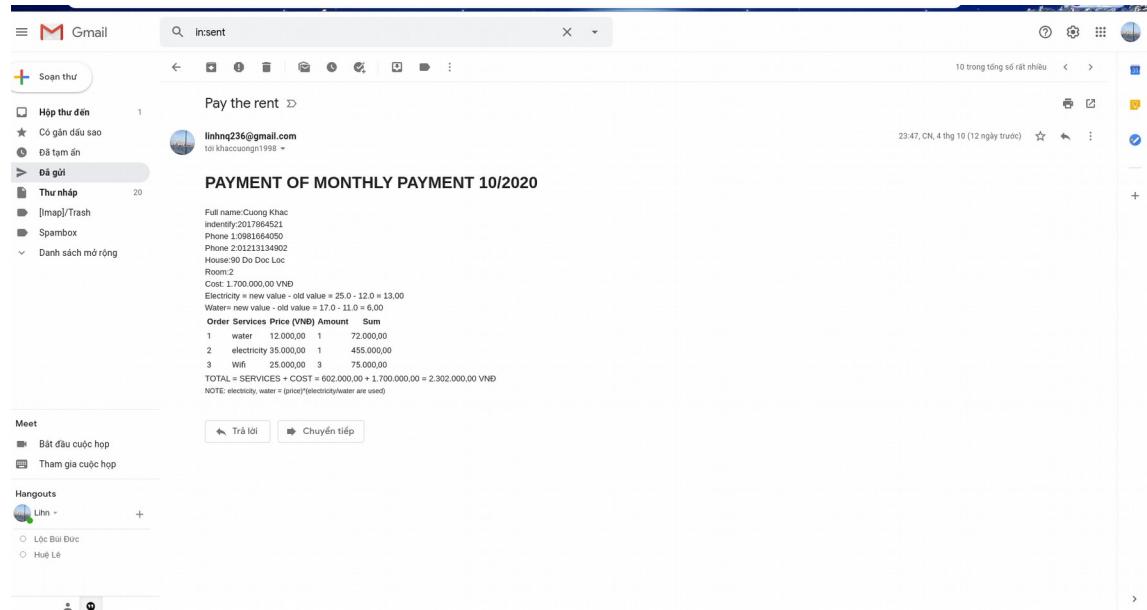
Services index	Old value	New value
Electricity	3	10
Water	4	8

Luân văn tốt nghiệp © 2021

Trên con đường thành công không có dấu chân của kẻ lường biến ❤

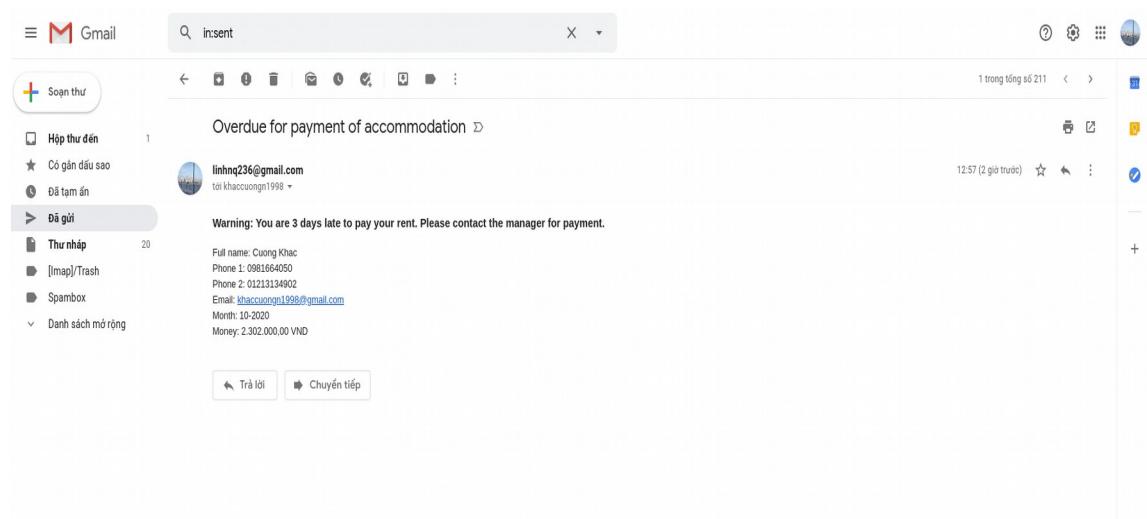
Hình 42: Giao diện cập nhật chỉ số dịch vụ điện, nước

Nội dung gửi mail thanh toán tiền trọ: Sau khi cập nhật xong chỉ số dịch vụ điện, nước thì sẽ có mail thông báo đóng tiền trọ.



Hình 43: Giao diện gửi thanh toán hóa đơn về mail

Nội dung gửi mail khi quá 3 ngày chưa đóng tiền trọ: Nếu sau 3 ngày kể từ ngày gửi thì khách hàng thuê trọ nào chưa thanh toán sẽ có mail thông báo nhắc nhở đóng tiền trọ tháng này.



Hình 44: Giao diện quá hạn trả tiền được gửi về mail

Trang Dashboard: Trang quản lý tất cả thiết bị tại khu vực mà người quản lý đang kiểm soát.

Active	Kind	Status	Time out	Timer
Enable	Door	Off	2020-12-03 18:31	2020-12-03 18:30
Enable	Ligth	Off	0000-00-00 00:00	0000-00-00 00:00
Enable	Ligth	Off	0000-00-00 00:00	0000-00-00 00:00
Enable	Ligth	Off	0000-00-00 00:00	0000-00-00 00:00
Disable	Fan	Off	0000-00-00 00:00	0000-00-00 00:00

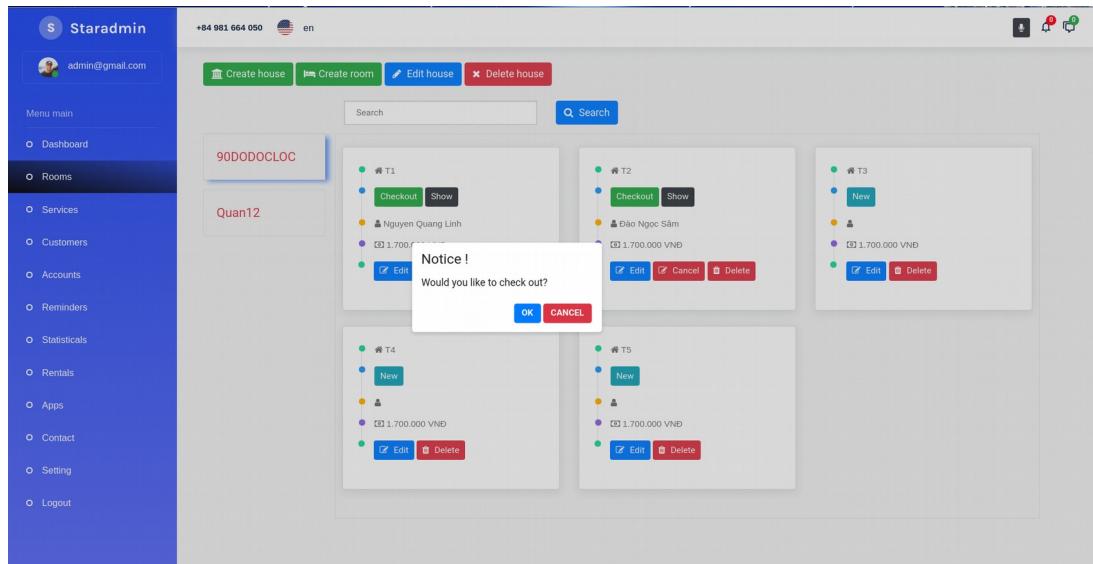
Hình 45: Giao diện quản lý thiết bị của chủ trọ

Trang Rooms: Quản lý tất cả các phòng trống và phòng đã có khách hàng.

T1	New
T2	New
T3	New
T4	New
T5	New

Hình 46: Giao diện quản lý phòng của chủ trọ

Checkout: Xuất hiện confirm xác nhận trả phòng.



Hình 47: Giao diện tiếp nhận trả phòng

Trang Services: Quản lý dịch vụ khách hàng phòng đang sử dụng

The screenshot shows the 'Services' management page. The sidebar menu includes 'Informations', 'Accounts', 'Reminders', 'Statistics', 'Rentals', and 'Logout'. The main content area displays two tables. The first table lists services by order, status, service name, and unit price:

Order	Status	Service name	Unit price
1	Compulsory	electricity	3.500,00
2	Compulsory	water	12.000,00
3	Optional	Wifi	25.000,00
4	Optional	trash	10.000,00

The second table lists services by order, service name, room master, and a delete button:

Order	Services name	Room master	Search service ...
1	electricity, water, Wifi, trash	Nguyen Quang Linh	
2	electricity, water, Wifi, trash	Dao Ngoc Sam	
3	electricity, water, Wifi, trash	Bui Duc Loc	

A watermark at the bottom right corner reads 'Activate Windows Go to Settings to activate Windows.'

Hình 48: Giao diện hiển thị chi tiết các dung vụ khách đang sử dụng của chủ trọ

Trang customers: Trang thông tin cá nhân của khách hàng trong dãy trọ.

The screenshot shows a web application interface for managing guest information. On the left is a sidebar with a blue header 'Staradmin' and a user profile picture. The sidebar menu includes 'Dashboard', 'Rooms', 'Services', 'Customers' (which is currently selected), 'Accounts', 'Reminders', 'Statisticals', 'Rentals', 'Contact', 'Setting', and 'Logout'. At the top right, there are icons for a microphone, a red notification badge, a green notification badge, and an envelope. The main content area has a search bar at the top. A table lists guest information:

Order	House/Room	Full name	Phone 1	Phone 2	Email
1	90DODOCLOC / Phong1	Nguyen Quang Linh	0981664050	01213134902	1651120032@sv.ut.edu.vn
2	90DODOCLOC / Phong2	Đào Ngọc Sâm	0981664050	01213134902	daongocsam98@gmail.com

Hình 49: Giao diện quản lý thông tin khách hàng của chủ trọ

Trang Accounts: Quản lý tài khoản người dùng. Những tài khoản người dùng nào đang hoạt động (active) thì có thể đăng nhập trên app và web

The screenshot shows a web application interface for managing accounts. The sidebar is identical to the one in Figure 49. The main content area has a search bar at the top. A table lists account information:

Order	Email	Kind	Created at	House	Status
1	1651120032@sv.ut.edu.vn	Manager	29-09-2020 15:32	90 Do Doc Loc	Active
2	linhnq236@gmail.com	Customer	29-09-2020 15:25	90 Do Doc Loc	Active
3	daongocsam98@gmail.com	Customer	29-09-2020 15:26	90 Do Doc Loc	Active
4	lockupin@gmail.com	Customer	29-09-2020 15:27	90 Do Doc Loc	Active

Hình 50: Giao diện quản lý tài khoản của chủ trọ

Trang Reminders: Những nhắc nhở mà người quản lý đã tạo nhằm mục đích thuận tiện vào quản lý hơn.

The screenshot shows the 'Reminders' section of the Staradmin application. On the left, a blue sidebar menu lists various options: Dashboard, Rooms, Services, Informations, Accounts, Reminders (which is currently selected), Statistical, Rentals, and Logout. At the top right, there are icons for microphone, notifications (with a red dot), messages (with a green dot), and email. The main content area features a calendar for October 2020. Below the calendar, six reminder cards are displayed, each with a date, a title, and a small 'X' icon to its right. The reminder titles are: '09/29 1 : Test', '10/05 10/06', '10/12 10/13', '10/19 10/20', '10/26 10/27', '09/30 1 : test', '10/01 1 : tết', '10/08 1 : admin1', '10/14', '10/15', '10/21', '10/22', '10/29', '10/30', '10/03', '10/10', '10/16', '10/17', '10/24', '10/31', and '11/01'. At the bottom right of the calendar area, there is a message: 'Activate Windows Go to Settings to activate Windows.'

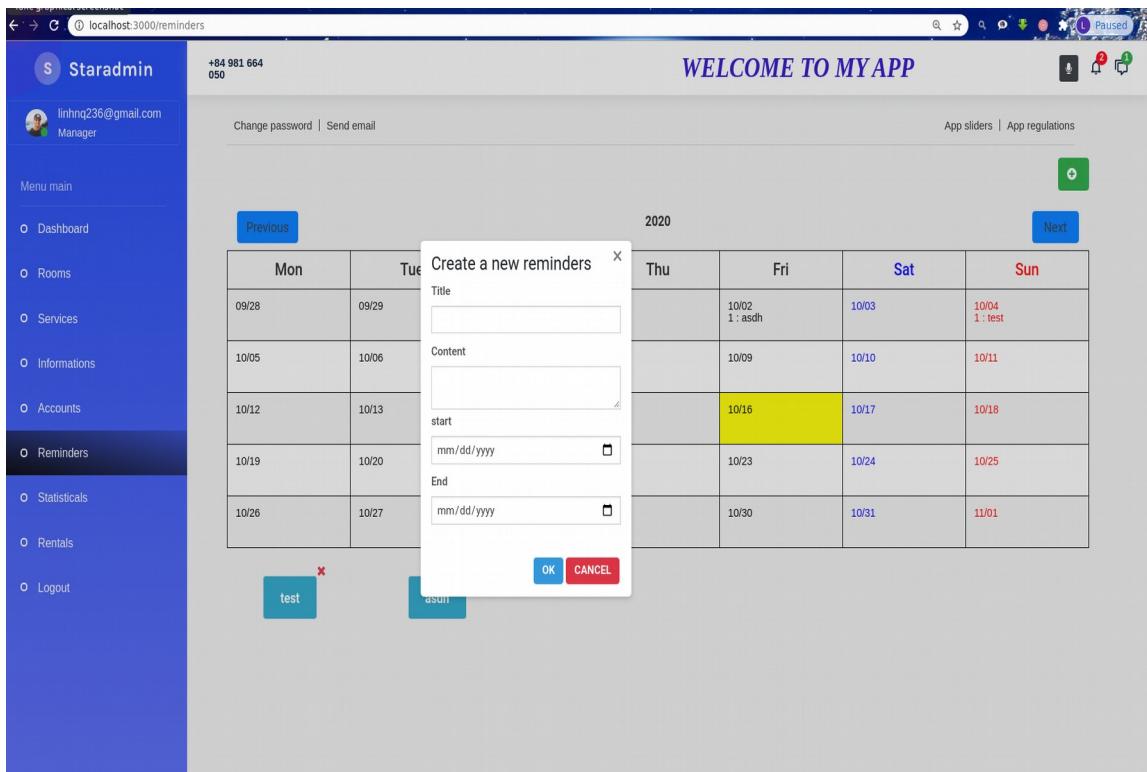
Mon	Tue	Wed	Thu	Fri	Sat	Sun
09/28	09/29 1 : Test	09/30 1 : test	10/01 1 : tết	10/02	10/03	10/04
10/05	10/06	10/07	10/08 1 : admin1	10/09	10/10	10/11
10/12	10/13	10/14	10/15	10/16	10/17	10/18
10/19	10/20	10/21	10/22	10/23	10/24	10/25
10/26	10/27	10/28	10/29	10/30	10/31	11/01

Test X
admin X
admin1 X
test X
tết X
Hlo X

Activate Windows
Go to Settings to activate Windows.

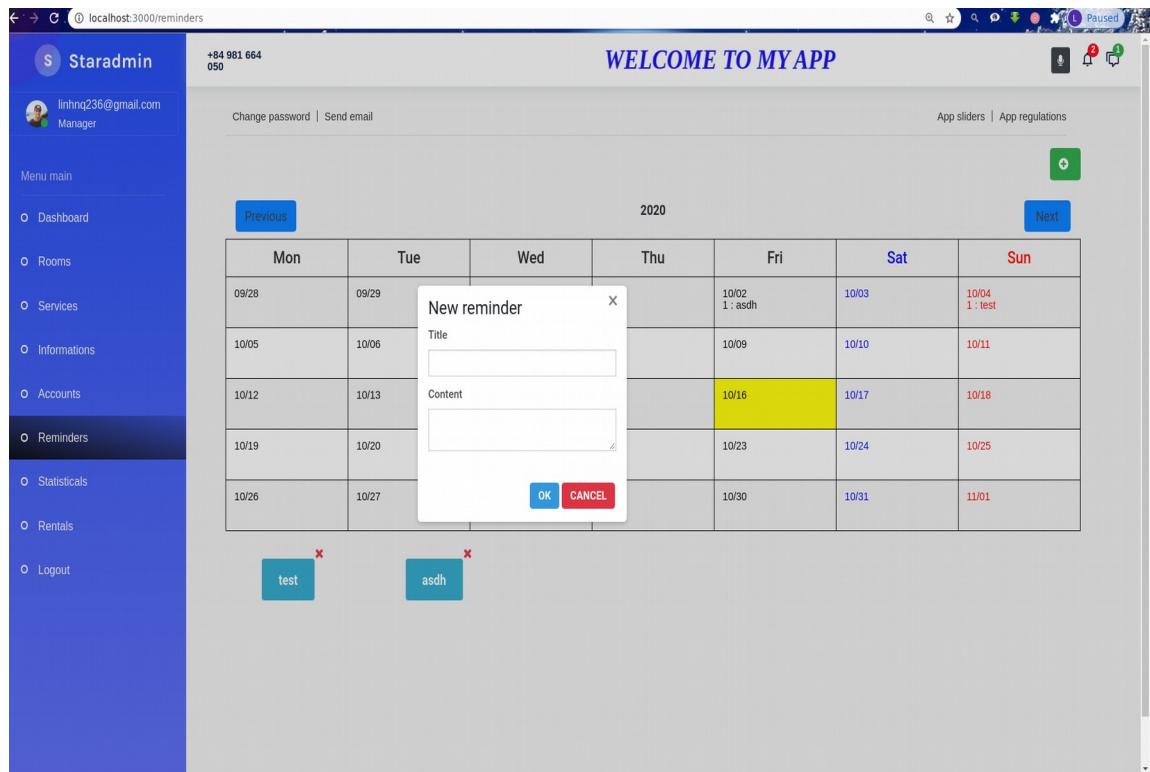
Hình 51: Giao diện tạo nhắc nhở của chủ trọ

Tạo một nhắc nhở mới: Tạo một nhắc nhở mới theo chuỗi ngày bắt đầu và kết thúc.



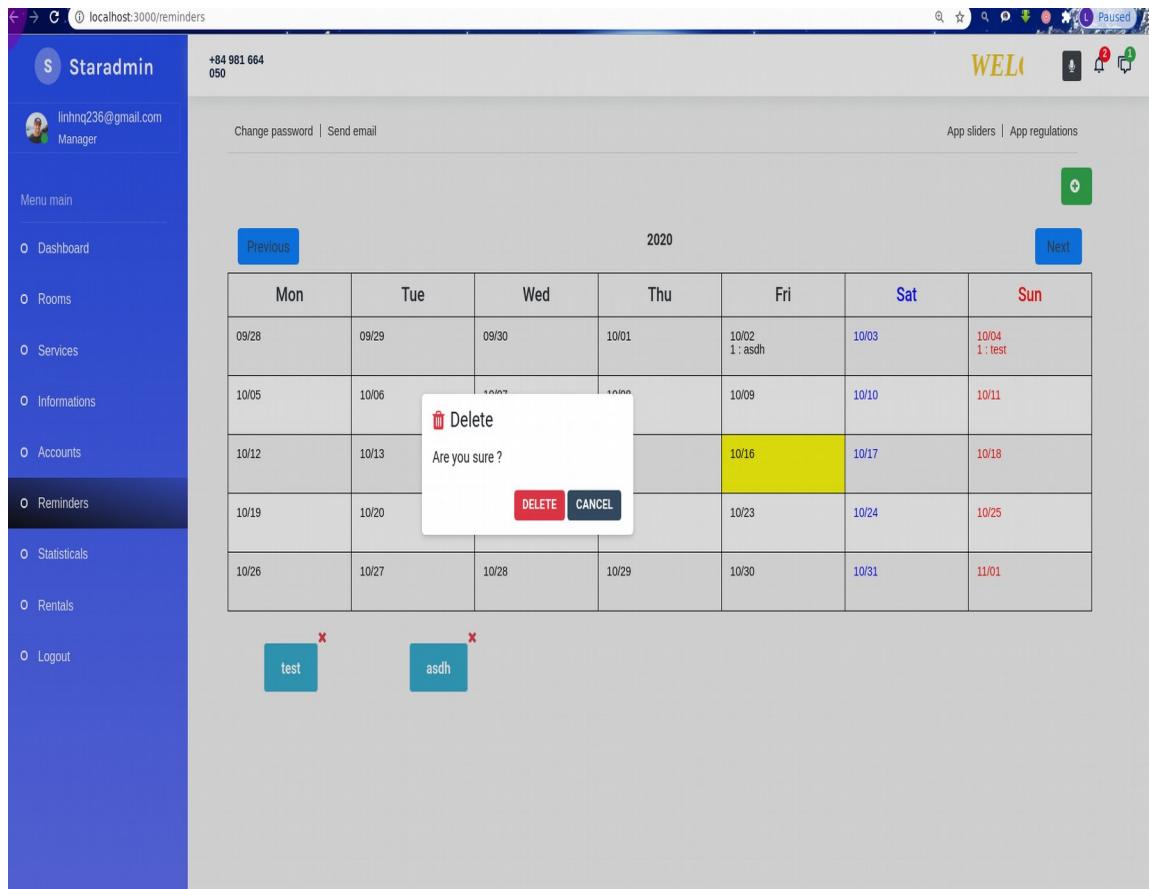
Hình 52: Giao diện tạo mới một nhắc nhở của chủ trọ

Tạo tại ngày: Tạo một nhắc nhở mới tại ngày muôn thông báo nhắc nhở



Hình 53: Giao diện tạo một nhắc nhở ngày đã được chọn của chủ trọ

Xóa 1 nhắc nhở: Xóa một nhắc nhở.



Hình 54: Giao diện xóa một nhắc nhở của chủ trọ

Trang Statisticals: Thống kê số phòng trống, những tài khoản đang hoạt động.



Hình 55: Giao diện thống kê của chủ trọ

Nội dung trong file text: Thông tin trong file text sau khi được export ra và sẽ in ra và cắt đưa cho từng phòng.

```
Fullname: Nguyen Quang Linh
Email   : linhnq236@gmail.com
Phone1  : 0981664050
Phone2  : 01213134902
House    : 90DoDocLoc
Room    : 1
Payment : Yes
Month   : 12-2020
Total   : 1746500 VND

| (Note: Let's check your email)
```

Hình 56: Giao diện xuất ra hóa đơn file txt

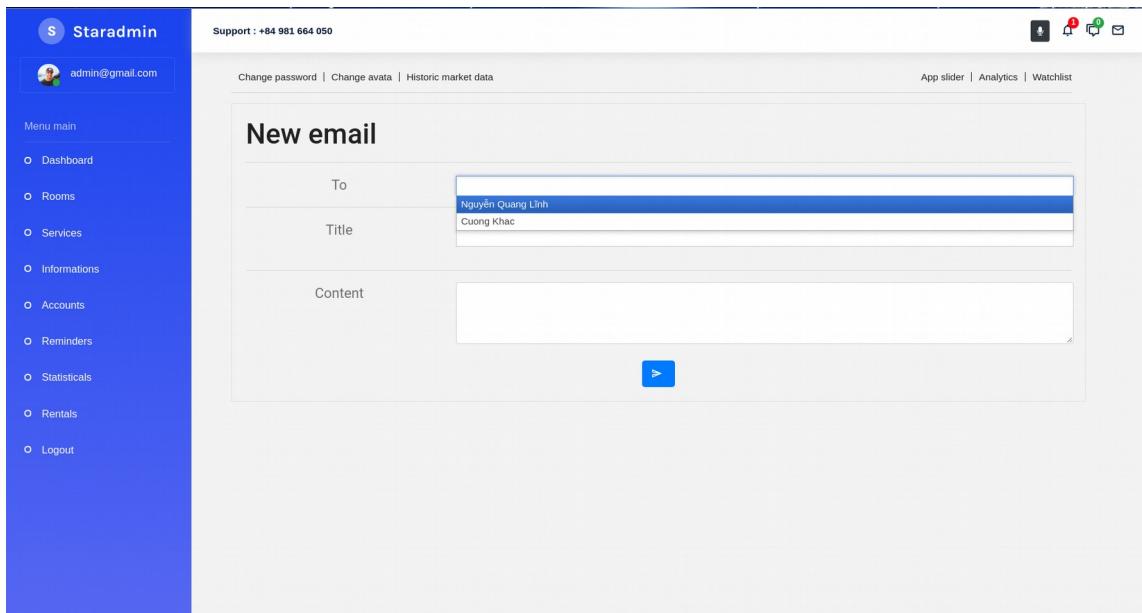
Trang Reports: Quản lý những phản hồi từ khách hàng. Nếu như khách hàng có phản hồi thông tin gì thì người quản lý sẽ tiếp nhận và sẽ phản hồi lại cho khách hàng một cách nhanh nhất.

The screenshot shows the 'Feedback' section of the Staradmin application. On the left, there's a sidebar with a user profile for 'Staradmin' (1651120032@sv.ut.edu.vn, Manager) and a navigation menu with links like Dashboard, Rooms, Services, Informations, Accounts, Reminders, Statistics, Rentals, and Logout. The main content area has a title 'Feedback' and three input fields: 'Title' (filled with 'Den chay'), 'Content' (filled with 'Thay the den chay'), and 'Feedback content' (an empty text area). Below these is a blue 'Send' button. At the bottom, there's a table showing a single feedback entry:

Order	Title	User	Created at	Status
1	Den chay	Nguyen Quang Linh	7 days	Active Windows Chưa phản hồi Go to Settings to activate Windows.

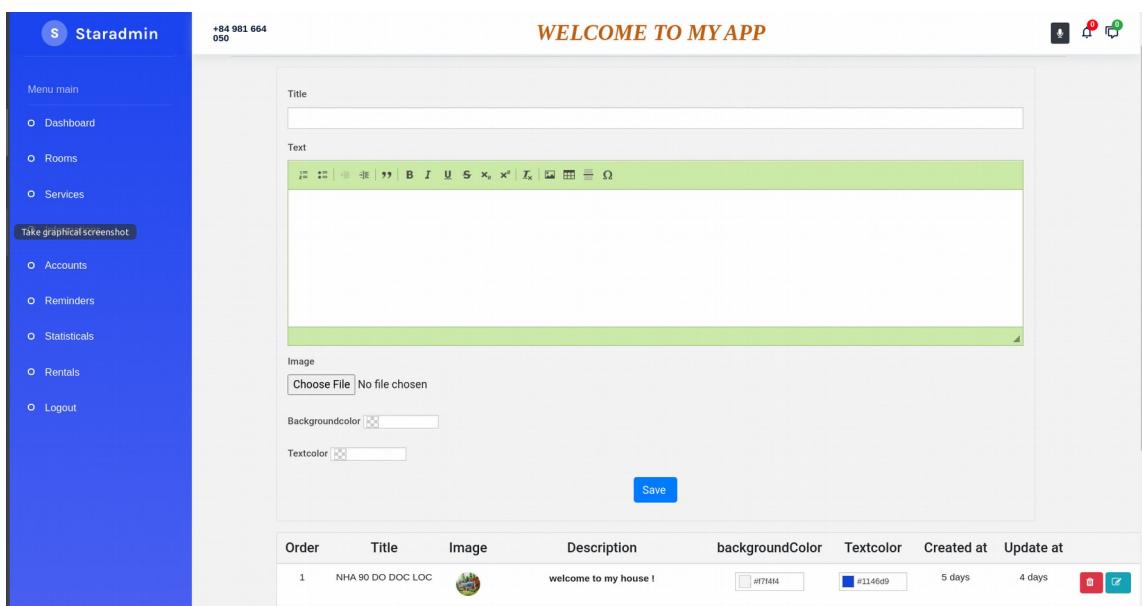
Hình 57: Giao diện quản lý phản hồi từ khách hàng của chủ trọ

Trang Email: Người quản lý có thông báo hay liên lạc có thể thông báo bằng cách gửi mail cho tất cả người trong phòng.



Hình 58: giao diện gửi mail của chủ trọ

Trang App Regulations: Để cho khách hàng thuê có sự an tâm thì nhà trọ sẽ có những quy định được áp dụng ra nhằm mục đích cho tất cả những khách hàng có một sự cảm thấy an tâm khi thuê phòng tại đây.



Hình 59: Giao diện tạo slider cho app và web bên khách hàng của quản trị viên

Trang Dashboard: Giúp cho người quản trị viên nhìn thấy tổng quan về các thiết bị của tất cả khu vực đang sử dụng rồi từ đó sẽ giúp quản lý, theo dõi một cách hiệu quả.

Name	Active	Status	Timer	Time out
DOOR	disable	off	0000-00-00 00:00	0000-00-00 00:00
LGTH 1	enable	off	0000-00-00 00:00	0000-00-00 00:00
POWER SOCKET	enable	off	0000-00-00 00:00	0000-00-00 00:00
LGTH 2	disable	off	0000-00-00 00:00	0000-00-00 00:00
FAN	disable	off	0000-00-00 00:00	0000-00-00 00:00
POWER SOCKET 1	disable	off	0000-00-00 00:00	0000-00-00 00:00
POWER SOCKET 2	disable	off	0000-00-00 00:00	0000-00-00 00:00
POWER SOCKET 3	disable	off	0000-00-00 00:00	0000-00-00 00:00

Hình 60: Giao diện quản lý tất cả các thiết bị ở tất cả khu vực của quản trị viên

Trang Rooms: Giúp cho quản trị viên quản lý và theo dõi các phòng của tất cả khu vực một cách hiệu quả.

Room	Occupants	Actions
90 Do Doc Loc	Nguyen Quang Linh (1.700.000,00 VND)	Edit Delete
Hoa Son	Do Ngoc Sam (1.700.000,00 VND)	Edit Delete
	Bui Duc Loc (1.700.000,00 VND)	Edit Delete
90 Do Doc Loc	Nguyen Quang Linh (1.700.000,00 VND)	Edit Delete
Hoa Son	Do Ngoc Sam (1.700.000,00 VND)	Edit Delete
	Bui Duc Loc (1.700.000,00 VND)	Edit Delete

Hình 61: Giao diện quản lý phòng của tất cả khu vực của quản trị viên

Trang Create House

The screenshot shows the 'Create New House' form. On the left is a blue sidebar menu with the title 'Staradmin' and a user profile for 'admin@gmail.com'. The main content area has a header with phone number '+84 981 664 050', language 'en', and a notification icon showing '0'. The form itself has a title 'Create New House' and two rows of input fields. The first row contains 'House *' and 'Province / City *' with dropdown menus. The second row contains 'District*' and 'Ward*' with dropdown menus, and 'Number *' and 'Road *' with text input fields. At the top right are 'Back' and 'Save' buttons.

Hình 62: Giao diện tạo một nhà mới

Trang Create Rooms

The screenshot shows the 'Create Room' form. It has a similar layout to the house creation form, with a sidebar menu on the left. The main form includes fields for 'Room number*', 'Unit price(vnd)*', 'Width(cm)*', 'Length(cm)*', 'Amount', 'House *', 'The maximum number of people*', 'Description', and a large text area for 'Devices'. Below the devices section, it says 'Total devices: 8' and lists 'Door' and 'Fan' with their respective counts. There are also fields for 'Ligth' and 'Power socket'. At the top right are 'Back' and 'Save' buttons.

Hình 63: Giao diện tạo một phòng mới

Trang Services

CREATE NEW SERVICE

Order	Status	Service name	Unit price	Action
1	Compulsory	electricity	3.500,00	Delete Edit
2	Compulsory	water	12.000,00	Delete Edit
3	Optional	Wifi	25.000,00	Delete Edit
4	Optional	trash	10.000,00	Delete Edit

Order	Services name	Room master	Action
1	electricity, water, Wifi, trash	Nguyen Quang Linh	Delete
2	electricity, water, Wifi, trash	Do Ngoc Sam	Delete
3	electricity, water, Wifi, trash	Bui Duc Loc	Delete

Hình 64: Giao diện tạo một dịch vụ mới của quản trị viên

Trang customers

Order	House/Room	Full name	Phone 1	Phone 2	Email
1	90DODOCLOC / Phong2	Đào Ngọc Sâm	0981664050	01213134902	daongocsam8@gmail.com
2	90DODOCLOC / Phong1	Nguyen Quang Linh	0981664050	01213134902	1651120032@sv.ut.edu.vn

Hình 65: Giao diện quản lý khách hàng

Trang Accounts

The screenshot shows the 'Accounts' section of the Staradmin dashboard. On the left, a sidebar menu includes 'Dashboard', 'Rooms', 'Services', 'Customers', 'Accounts' (which is selected and highlighted in blue), 'Reminders', 'Statisticals', 'Rentals', 'Apps', 'Contact', 'Setting', and 'Logout'. The main content area displays a table of user information:

Order	Email	Kind	Created at	House	Status
1	1651120032@sv.ut.edu.vn	Manager	03-12-2020 18:37	90DODOCLOC	Active
2	linhng236@gmail.com	Customer	03-12-2020 19:01	90DODOCLOC	Active
3	daongocsam98@gmail.com	Customer	05-12-2020 23:21	90DODOCLOC	Active

Hình 66: Giao diện quản lý tài khoản của quản trị viên

Trang Rentals

The screenshot shows the 'Rentals' section of the Staradmin dashboard. The sidebar menu is identical to the one in the previous screenshot. The main content area displays a table of rental payment records:

Order	Room master	Month	Money (VND)	Date of payment	Status
1	Nguyen Quang Linh	12-2020	1.870.500		Open
2	Đào Ngọc Sâm	12-2020	1.917.000		Open
3	Đào Ngọc Sâm	11-2020	1.517.000		Open
4	Nguyen Quang Linh	11-2020	2.317.000		Open
5	Đào Ngọc Sâm	10-2020	2.917.000		Open
6	Nguyen Quang Linh	10-2020	1.117.000		Open
7	Đào Ngọc Sâm	09-2020	1.947.000		Open
8	Nguyen Quang Linh	09-2020	1.437.000		Open
9	Đào Ngọc Sâm	08-2020	1.617.000		Open
10	Nguyen Quang Linh	08-2020	1.934.000	2020-12-06 03:09	Completed
11	Đào Ngọc Sâm	07-2020	1.947.000		Open

Hình 67: Giao diện quản lý chi phí của quản trị viên

III. API

1. Các API được sử dụng trong trang web

Bảng 16: Api AppSlider

Url	http://localhost:3000/api/appSlider
Type	GET
Input	Không
Output	JSON

- Ví dụ

```
url: http://localhost:3000/api/appSlider
type: get
data: none
response:
{
  "slides": [
    {
      "id": 4,
      "title": "Test",
      "text": "tesster",
      "image": {
        "url": "/uploads/avata.jpg"
      },
      "backgroundColor": "#088508",
      "created_at": "2020-10-02T15:10:28.778+07:00",
      "updated_at": "2020-10-02T21:24:06.541+07:00"
    }
  ]
}
```

Bảng 17: Api đăng nhập

Url	http://localhost:3000/api/account
Type	POST
Input	email, password
Output	JSON

- Ví dụ

url: http://localhost:3000/api/account

type: post

data: {

 “email”: “linhnq236@gmail.com”,

 “password”:123456

}

response:

{

 “status”: 200,

 “username”: “Nguyen Quang Linh”,

 “id”: “3”,

 “disable”: 0

}

- Hiển thị thông tin đã đăng ký

url: http://localhost:3000/api/getinfo/4

type: get

data: none

response:

{

 “data”: {

 “id”: 4,

 “name”: “Nguyễn Quang Linh”,

```

    "sex": false,
    "birth": "2020-09-28",
    "indentifycard": "2017864521",
    "daterange": "2016-10-28",
    "placerange": "0145689513",
    "phone1": "",
    "phone2": "132123",
    "permanent": "Hòa Sơn",
    "start": "2021-08-27",
    "deposit": 0,
    "note": " K ",
    "created_at": "2020-09-28T12:53:10.903+07:00",
    "updated_at": "2020-10-03T22:45:11.564+07:00",
    "email": "1651120032@sv.ut.edu.vn",
    "mark": 0
}
}

```

Bảng 18: Api cập nhật thông tin cá nhân

Url	http://localhost:3000/api/updateInfo/id
Type	POST
Input	name, birth, indentifycard, daterange, placerange, phone1, phone2, permant
Output	Trạng thái

- Ví dụ

url: <http://localhost:3000/api/updateInfo/4>

type: post

data: {

“name”: “Nguyen Quang Linh”,

“birth”: “98-23-06”,

```

    "indentifycard": "123456789"
    "daterange": "2017-07-15",
    "placerange": "Da Nang",
    "phone1": "098166405",
    "phone2": "0981664050",
    "permant": "Da Nang"
}
response:
{
  "status": 200
}

```

Bảng 19: Api hiển thị bảng điều khiển

Url	http://localhost:3000/api/led_status/id
Type	GET
Input	Rỗng
Output	JSON

- Ví dụ

```

url: http://localhost:3000/api/led_status/4
type: get
data: none
response:
{
  "leds": {
    "led_status0": {
      "status": "off",
      "turnoff": "2020-10-05 00:05",
      "turnon": "2020-09-29 18:00"
    },
    "led_status1": {

```

```
        "status": "off",
        "turnoff": "2020-09-29 18:09",
        "turnon": "2020-09-30 06:09"
    },
    "led_status2": {
        "status": "off",
        "turnoff": "2020-09-29 05:09",
        "turnon": "2020-09-29 07:09"
    },
    "led_status3": {
        "status": "off",
        "turnoff": "13:00",
        "turnon": "2020-09-29 06:31"
    },
    "led_status5": {
        "status": "off",
        "turnoff": "13:00",
        "turnon": "13:00"
    },
    "led_status6": {
        "status": "off",
        "turnoff": "13:00",
        "turnon": "13:00"
    },
    "led_status7": {
        "status": "off",
        "turnoff": "13:00",
        "turnon": "13:00"
    }
}
```

```

        },
        "led_status8": {
            "status": "off",
            "turnoff": "13:00",
            "turnon": "13:00"
        }
    }
}

```

Bảng 20: Api chế độ điều khiển thiết bị: Bật, tắt và hẹn giờ

Url	http://localhost:3000/api/led_status/id
Type	POST
Input	Name, column, status
Output	Trạng thái

- Ví dụ

url: http://localhost:3000/api/led_status/4

type: post

data:

```
{
    "name": "led_status0",
    "column": "status",
    "status": "on"
}
```

response:

```
{
    "status": 200
}
```

Bảng 21: Api lấy thông tin dịch vụ đã đăng ký

Url	http://localhost:3000/api/getUseServices/id
Type	GET
Input	Rỗng

Output	JSON
<ul style="list-style-type: none"> • Ví dụ <p>url: http://localhost:3000/api/getUseServices/4</p> <p>type: get</p> <p>data: none</p> <p>response:</p> <pre>{ "status": 200, "data": [{ "id": 2, "name": "water", "cost": "12000", "created_at": "2020-09-25T13:23:25.339+07:00", "updated_at": "2020-09-25T13:23:25.339+07:00", "status": 1 }, { "id": 1, "name": "electricity", "cost": "35000", "created_at": "2020-09-25T13:23:25.314+07:00", "updated_at": "2020-09-29T01:25:53.336+07:00", "status": 0 }, { "id": 3, "name": "Wifi", "cost": null, "created_at": null, "updated_at": null, "status": 0 }] }</pre>	

```

        "cost": "25000",
        "created_at": "2020-09-29T01:26:12.322+07:00",
        "updated_at": "2020-09-29T01:26:12.322+07:00",
        "status": 0
    }
],
"service_amount": [
    "1",
    "1",
    "1"
]
}

```

Bảng 22: Api lấy dữ liệu thanh toán hàng tháng

Url	http://localhost:3000/api/getPaytheRent /id
Type	GET
Input	Rỗng
Output	JSON

- Ví dụ

url: <http://localhost:3000/api/getPaytheRent/4>

type: get

data: none

response:

```

{
    "data": [
        {
            "id": 1,
            "senddate": "09-2020",
            "receivedate": "2020-10-04 23:47",
            "status": 1,
        }
    ]
}

```

```

    "created_at": "2020-09-29T15:57:28.413+07:00",
    "updated_at": "2020-10-04T23:47:53.482+07:00",
    "information_id": 4,
    "money": 1913000
  }
]
}

```

Bảng 23: Api gửi phản hồi

Url	http://localhost:3000/api/reports
Type	POST
Input	Title, content, information_id
Output	Status

- Ví dụ

url: <http://localhost:3000/api/reports>

type: post

data: {

“title”: “Test”,

“content”: “Test content”,

“information_id”: 4

}

response:

{

“status”:200

}

Bảng 24: Api Lấy thông tin đã phản hồi

Url	http://localhost:3000/api/showpopup/id
Type	GET
Input	Rỗng

Output	JSON
<ul style="list-style-type: none"> • Ví dụ <p>url: https://localhost:3000/api/showpopup/4</p> <p>type: get</p> <p>data: none</p> <p>response:</p> <pre>{ "data": [{ "id": 1, "title": "Test", "content": "Test content", "rep_content": null, "mark": 0, "created_at": "2020-10-09T15:22:41.119+07:00", "updated_at": "2020-10-09T15:22:41.119+07:00", "information_id": 4, "user_id": 2 }] }</pre>	

Bảng 25: Api xóa một phản hồi

Url	http://localhost:3000/api/reports /id
Type	DELETE
Input	Rỗng
Output	JSON

- Ví dụ

url: <http://localhost:3000/api/reports/4>

```

type: delete
data: none
response:
{
    "data": {
        "id": 1,
        "title": "Test",
        "content": "Test content",
        "rep_content": null,
        "mark": 0,
        "created_at": "2020-10-09T15:22:41.119+07:00",
        "updated_at": "2020-10-09T15:22:41.119+07:00",
        "information_id": 4,
        "user_id": 2
    }
}

```

Bảng 26: Api lấy thông tin nội quy

Url	http://localhost:3000/api/getRegulations /id
Type	GET
Input	Rỗng
Output	JSON

- Ví dụ

url: <http://localhost:3000/api/getRegulations/4>

type: get

data: none

response:

```
{
  "regulations": [
    {
      "id": 5,
      "title": "AN NINH TRẬT TỰ",
      "description": "<ol>\r\n\t<li>Kh&ocirc;ng g&acirc;y tiếng ồn qu&aacute;c mức quy định l&agrave;m ảnh hưởng đến an ninh trật tự v&agrave; y&ecirc;n tỉnh chung.</li>\r\n\t<li>Kh&ocirc;ng g&acirc;y ồn &agrave;o l&uacute;c 21h30 trở đi</li>\r\n</ol>\r\n",
      "created_at": "2020-10-11T03:22:34.354+07:00",
      "updated_at": "2020-10-12T15:07:01.584+07:00",
      "house_id": 2
    },
    {
      "id": 7,
      "title": "NẾP SỐNG VĂN MINH",
      "description": "<ol>\r\n\t<li>Kh&ocirc;ng phơi quần &aacute;o tr&ecirc;n c&aacute;c h&agrave;ng lang chung, cầu thang.</li>\r\n\t<li>Kh&ocirc;ng sử dụng khung chung để trồng c&acirc;y cảnh, hoa...</li>\r\n\t<li>Kh&ocirc;ng xả r&aacute;c thả.. c&aacute;c chất động hại ra m&ocirc;i trường.</li>\r\n</ol>\r\n",
      "created_at": "2020-10-12T15:08:51.143+07:00",
      "updated_at": "2020-10-12T15:08:51.143+07:00",
    }
  ]
}
```

```
    "house_id": 2
  },
],
"house_name": "90 Do Doc Loc"
}
```

CHƯƠNG 2: ỨNG DỤNG NGƯỜI DÙNG CHẠY TRÊN APP

I. Phân tích hệ thống

1. Xác định yêu cầu

Khách hàng quản lý, theo dõi và điều khiển các thiết bị điện trong phòng trọ của mình.

2. Đặc tả yêu cầu

a. Yêu cầu chức năng

Sau khi khách hàng đã đặt phòng thành công và dọn đến ở thì người chủ trọ sẽ cấp một tài khoản đăng nhập trên app với email người đăng ký và mật khẩu được cấp.

b. Yêu cầu phi chức năng

- App chạy ổn định, tốc độ xử lý nhanh.
- Thông tin phải được bảo mật.
- Giao diện các màn hình đẹp.
- Sử dụng Nodejs 8.17.0, React-native-cli 2.0.1, React-native 0.62.2

3. Xây dựng biểu đồ chức năng

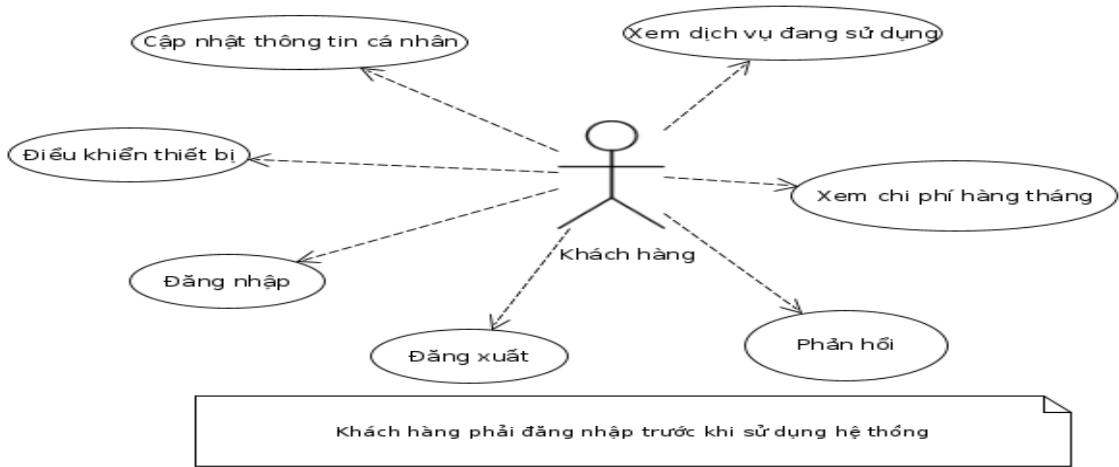
a. Tác nhân

Chỉ có một tác nhân tham gia và tương tác với app: Khách hàng thuê

b. Chức năng hệ thống

- Đăng nhập
- Đăng xuất
- Cập nhật thông tin cá nhân.
- Quản lý thiết bị: Bật, tắt và hẹn thời gian của thiết bị tại chính phòng đang ở.
- Xem dịch vụ đang sử dụng và lịch sử thanh toán tiền trọ
- Phản hồi

c. Biểu đồ phân rã chức năng khách hàng

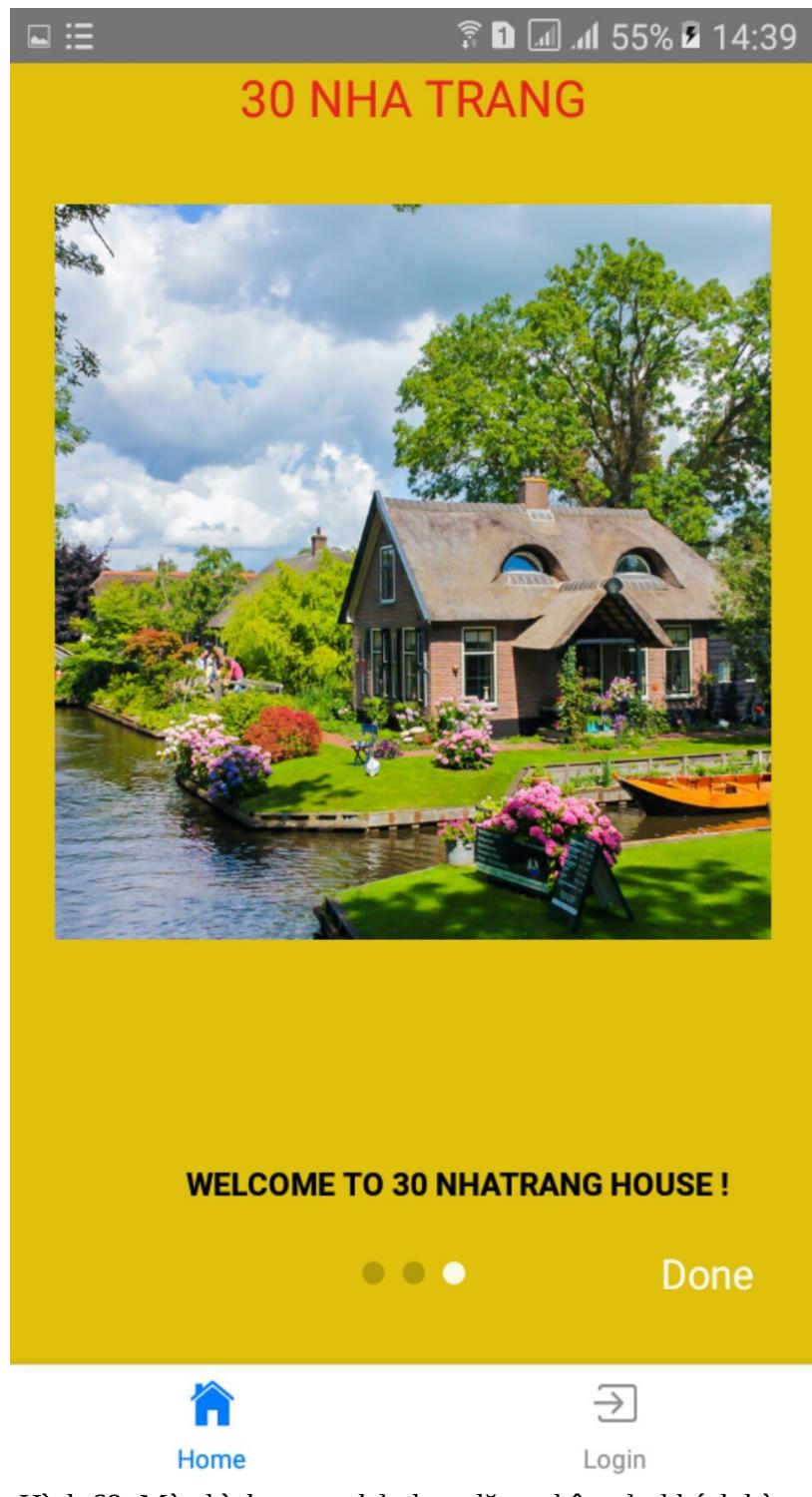


Hình 68: Biểu đồ phân rã chức năng của khách hàng

II. Thiết kế hệ thống

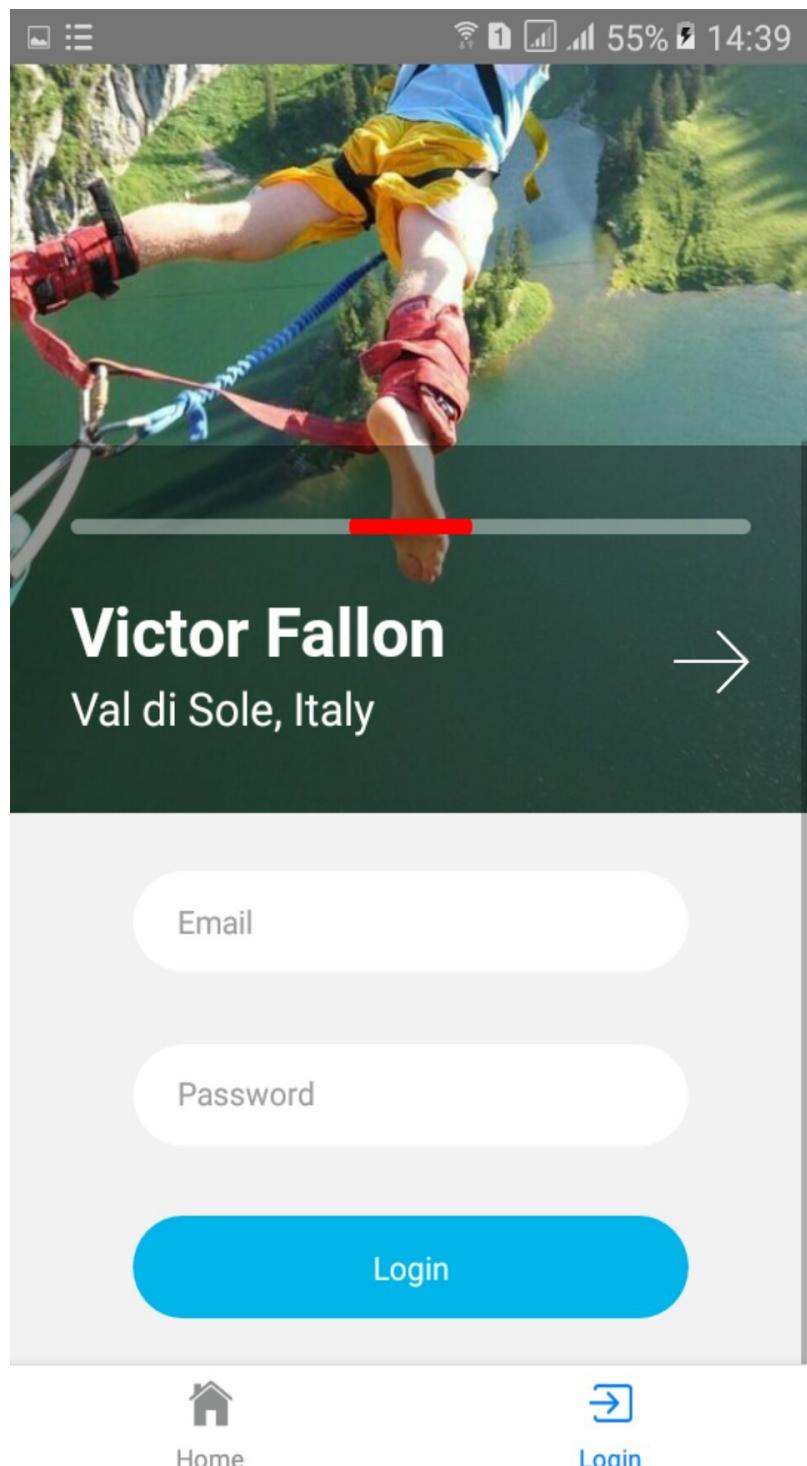
1. Mô tả chi tiết từng màn hình

- Màn hình trang chủ chưa đăng nhập – Home



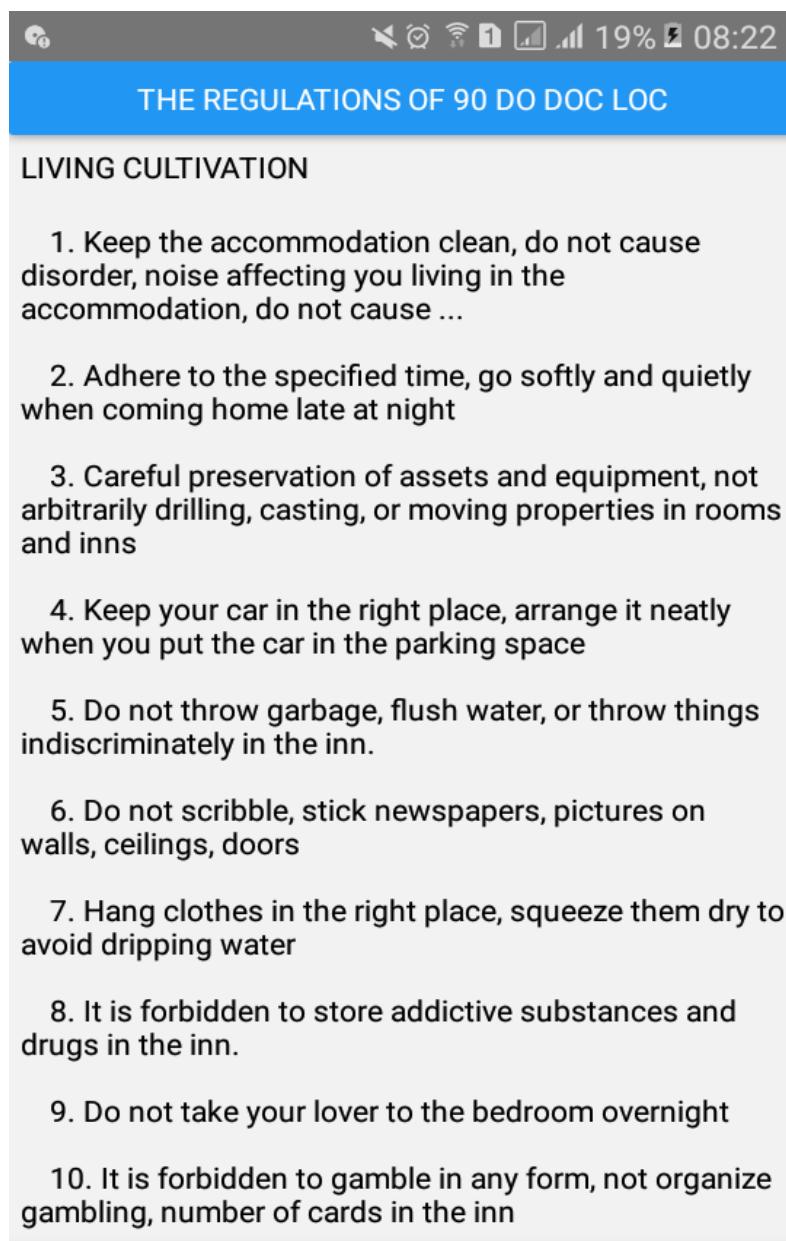
Hình 69: Màn hình trang chủ chưa đăng nhập của khách hàng

b. Màn hình trang đăng nhập – Login



Hình 70: Màn hình đăng nhập của khách hàng

- c. Màn hình trang chủ khi đã đăng nhập – Home

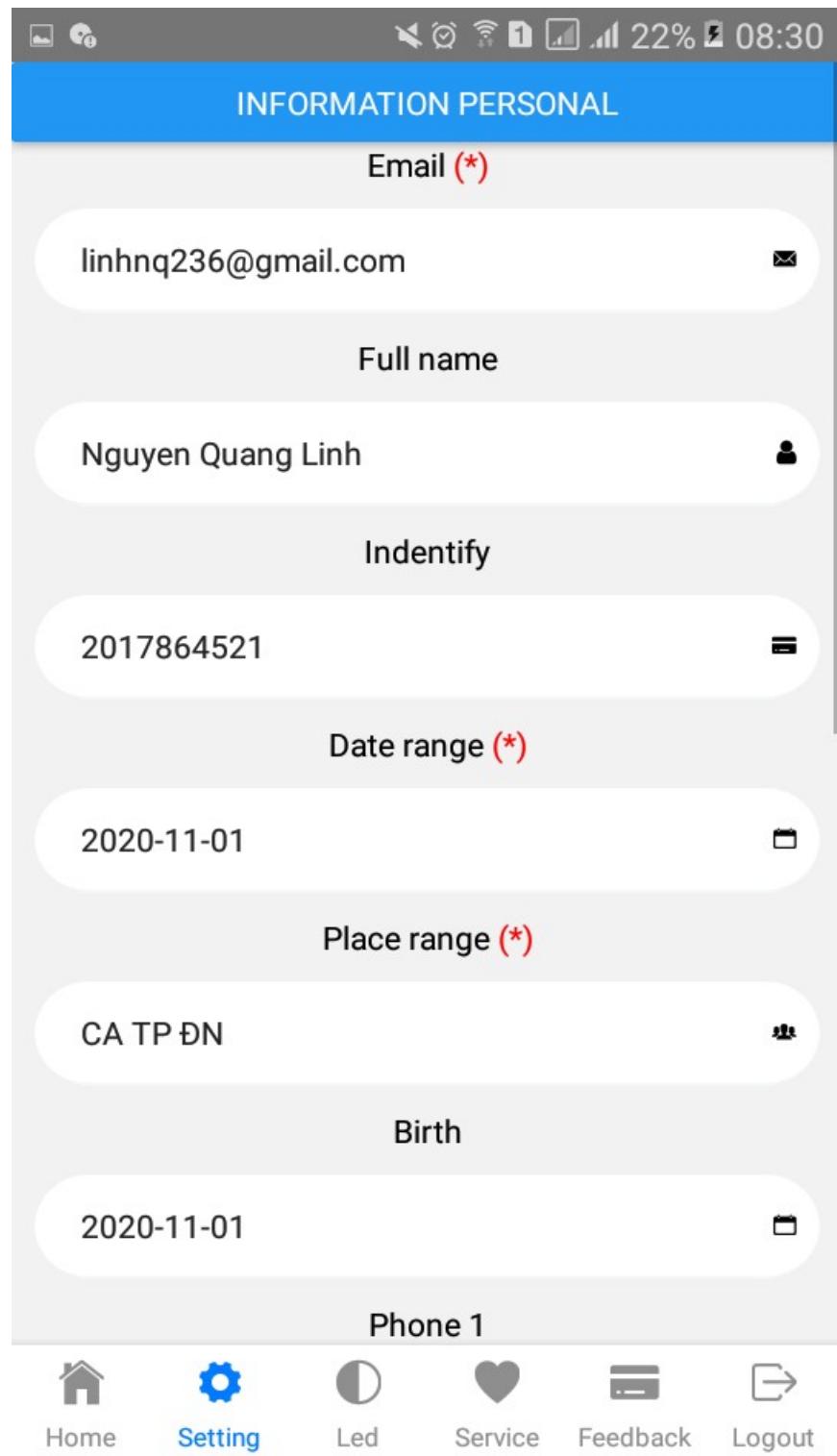


[Home](#) [Setting](#) [Led](#) [Service](#) [Feedback](#) [Logout](#)

Hình 71: Màn hình trang chủ của khách hàng sau khi đăng nhập

d. Màn hình trang thông tin cá nhân – Setting



Hình 72: Màn hình cập nhật thông tin cá nhân khách hàng

e. Màn hình quản lý thiết bị – Leds

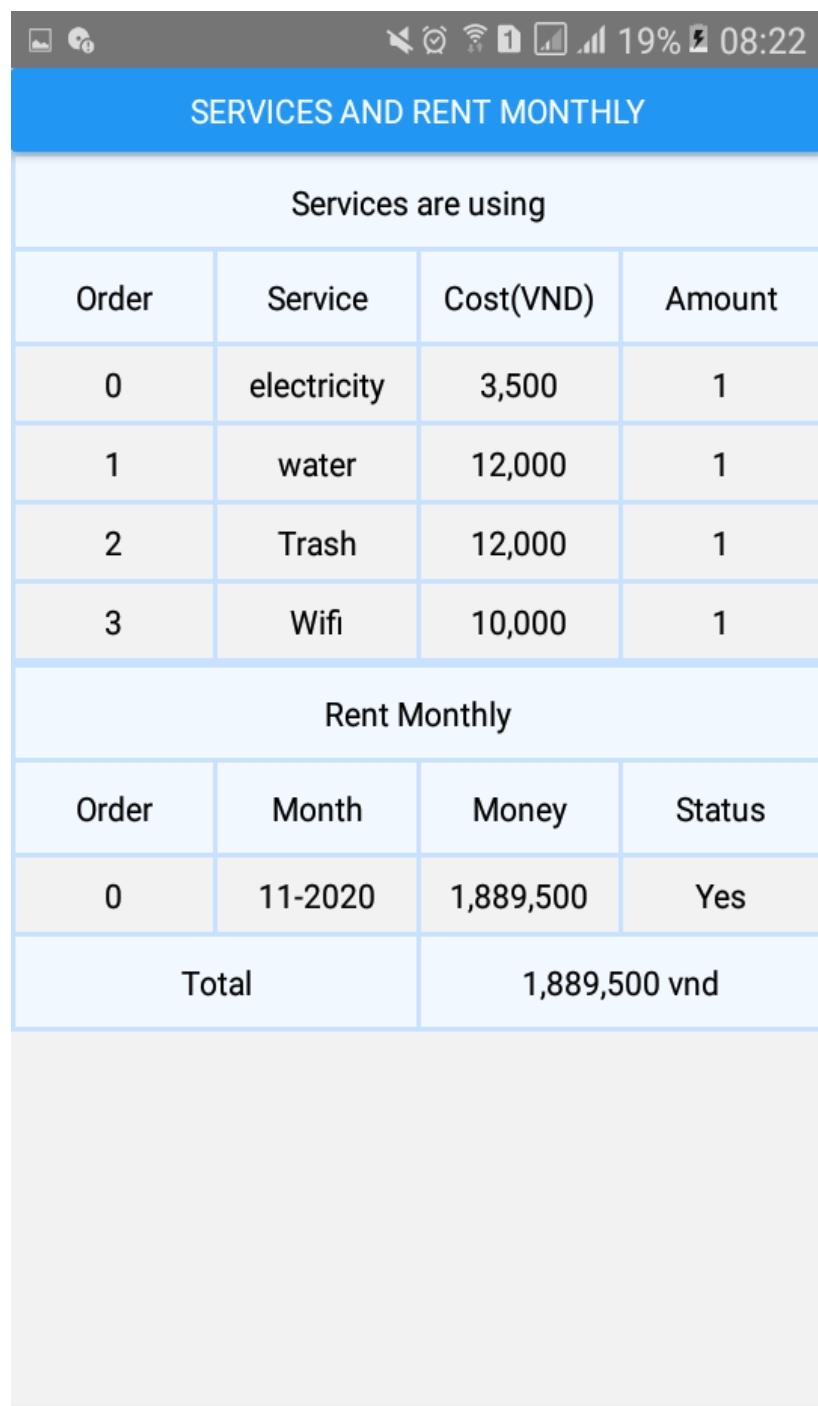
The screenshot shows a mobile application interface titled "DASHBOARD". At the top, there is a header bar with various icons and the time "08:22". Below the header is a yellow header bar with the word "DASHBOARD". The main content is a table with four columns: "Name", "Status", "Timer", and "Timeout". The table lists six devices: Light 1, Light 2, Light 3, Light 4, Fan, and Power socket 1. Each device row contains a lock icon, the device name, its current status ("off"), the timer value ("0000-00-00 00:00"), and the timeout value ("0000-00-00 00:00"). At the bottom of the screen, there is a navigation bar with icons for Home, Setting, Led (highlighted in blue), Service, Feedback, and Logout.

Name	Status	Timer	Timeout
Light 1	off	0000-00-00 00:00	0000-00-00 00:00
Light 2	off	0000-00-00 00:00	0000-00-00 00:00
Light 3	off	0000-00-00 00:00	0000-00-00 00:00
Light 4	off	0000-00-00 00:00	0000-00-00 00:00
Fan	off	0000-00-00 00:00	0000-00-00 00:00
Power socket 1	off	0000-00-00 00:00	0000-00-00 00:00

Home
 Setting
 Led
 Service
 Feedback
 Logout

Hình 73: Màn hình điều khiển thiết bị của khách hàng

f. Màn hình dịch vụ – Services



The screenshot shows a mobile application interface titled "SERVICES AND RENT MONTHLY". The top status bar displays icons for signal strength, battery level (19%), and time (08:22). The main content area is divided into two sections: "Services are using" and "Rent Monthly".

Services are using:

Order	Service	Cost(VND)	Amount
0	electricity	3,500	1
1	water	12,000	1
2	Trash	12,000	1
3	Wifi	10,000	1

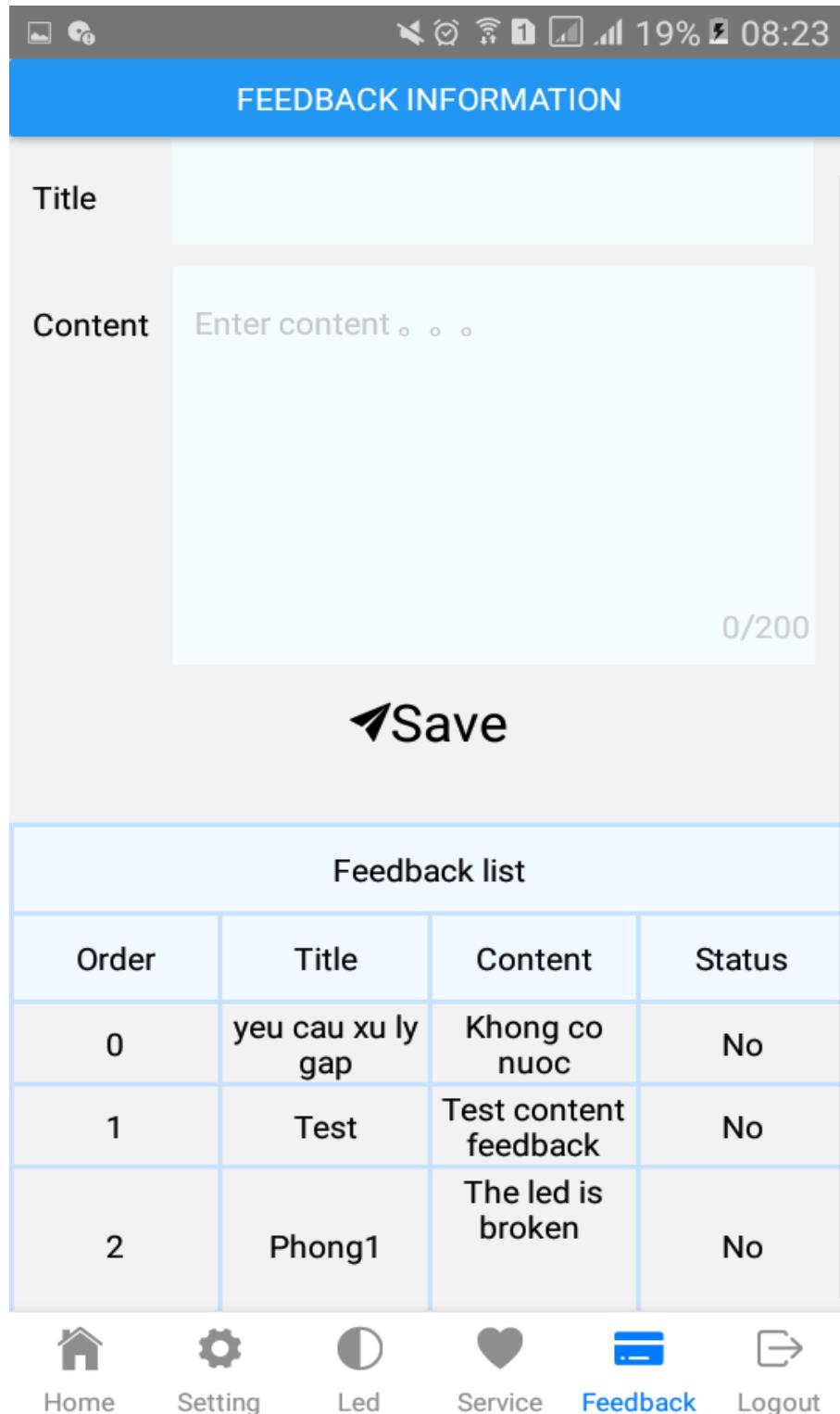
Rent Monthly:

Order	Month	Money	Status
0	11-2020	1,889,500	Yes
Total		1,889,500 vnd	

At the bottom, there is a navigation bar with icons and labels: Home, Setting, Led, Service (highlighted in blue), Feedback, and Logout.

Hình 74: Màn hình hiển thị dịch vụ và chi phí hàng tháng của khách hàng

f. Màn hình phản hồi – Feedback



Hình 75: Màn hình gửi phản hồi của khách hàng

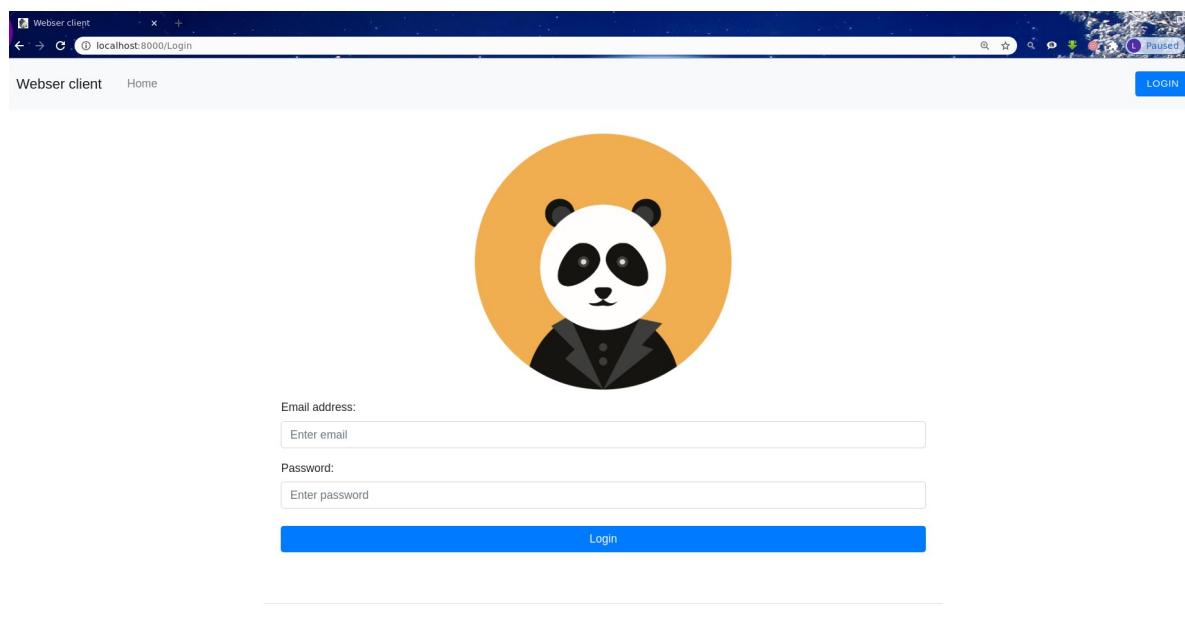
CHƯƠNG 3: ỨNG DỤNG CỦA NGƯỜI DÙNG CHẠY TRÊN WEB

I. Phân tích yêu cầu

Tương tự như ứng dụng người dùng chạy trên app thì ứng dụng người dùng chạy trên web được dựa trên các API được trả về. Từ đó ứng dụng chạy trên web được phát triển.

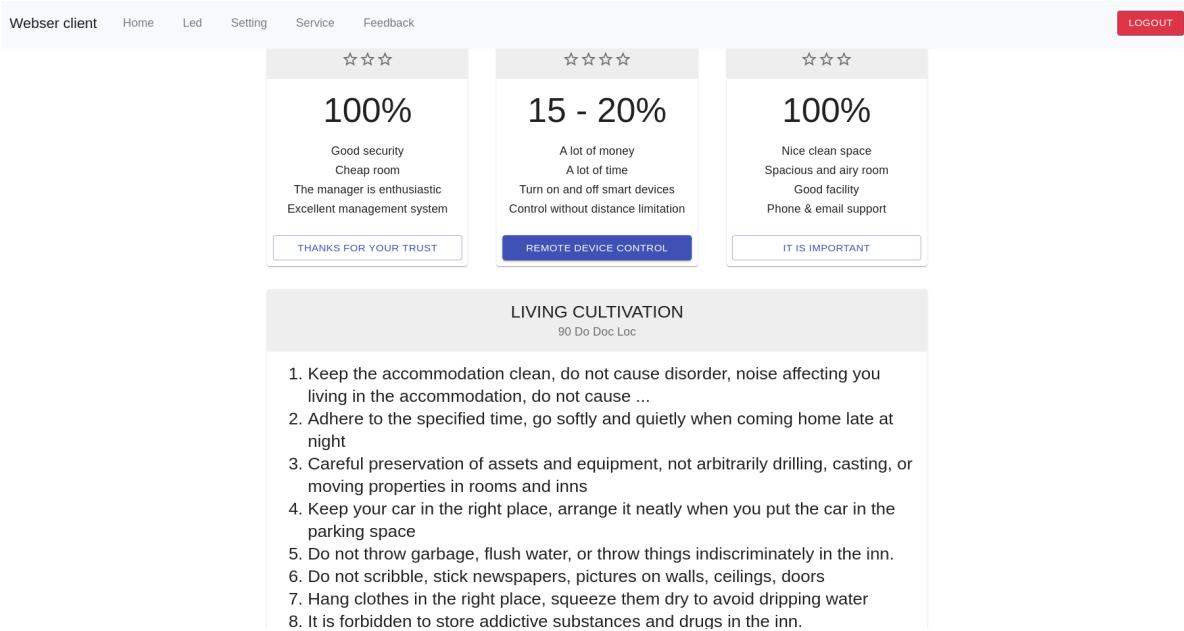
II. Các màn hình trên website của khách hàng

Màn hình login



Hình 76: Giao diện đăng nhập của khách hàng

Màn hình Home khi đã đăng nhập



Hình 77: Giao diện trang chủ của khách hàng

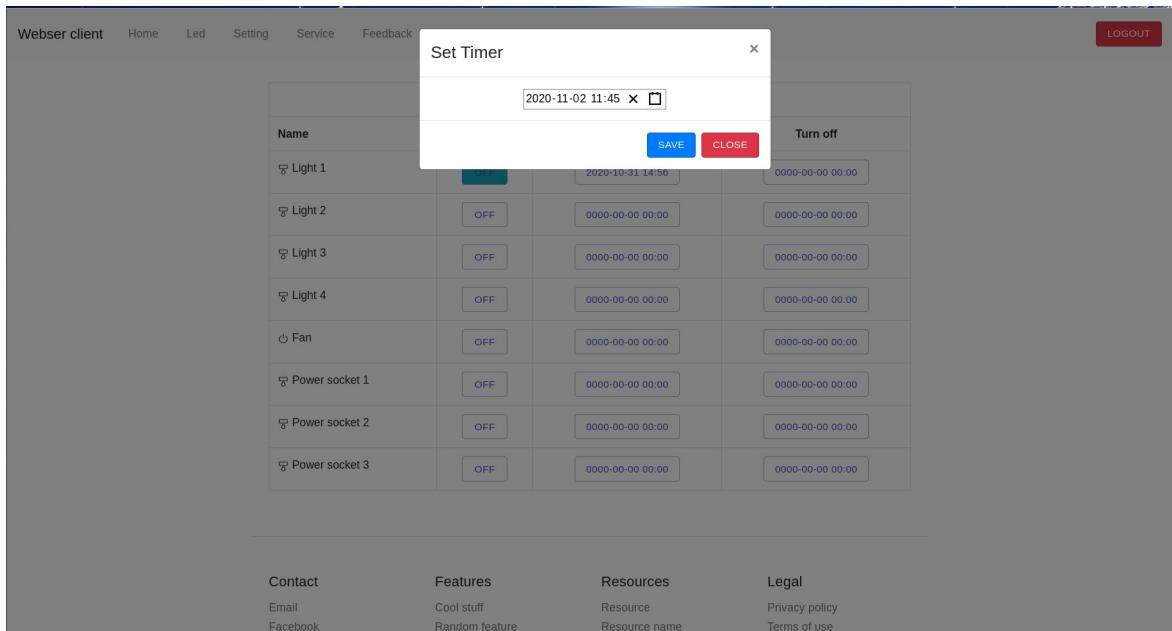
Màn hình Led - điều khiển thiết bị

The screenshot shows the 'Webser client' interface with a navigation bar at the top: Home, Led, Setting, Service, Feedback, and a red 'LOGOUT' button. Below the navigation bar is a table titled 'DEVICES IN MY ROOM' with columns: Name, Status, Turn on, and Turn off.

Name	Status	Turn on	Turn off
Light 1	OFF	2020-10-31 14:56	0000-00-00 00:00
Light 2	OFF	0000-00-00 00:00	0000-00-00 00:00
Light 3	OFF	0000-00-00 00:00	0000-00-00 00:00
Light 4	OFF	0000-00-00 00:00	0000-00-00 00:00
Fan	OFF	0000-00-00 00:00	0000-00-00 00:00
Power socket 1	OFF	0000-00-00 00:00	0000-00-00 00:00
Power socket 2	OFF	0000-00-00 00:00	0000-00-00 00:00
Power socket 3	OFF	0000-00-00 00:00	0000-00-00 00:00

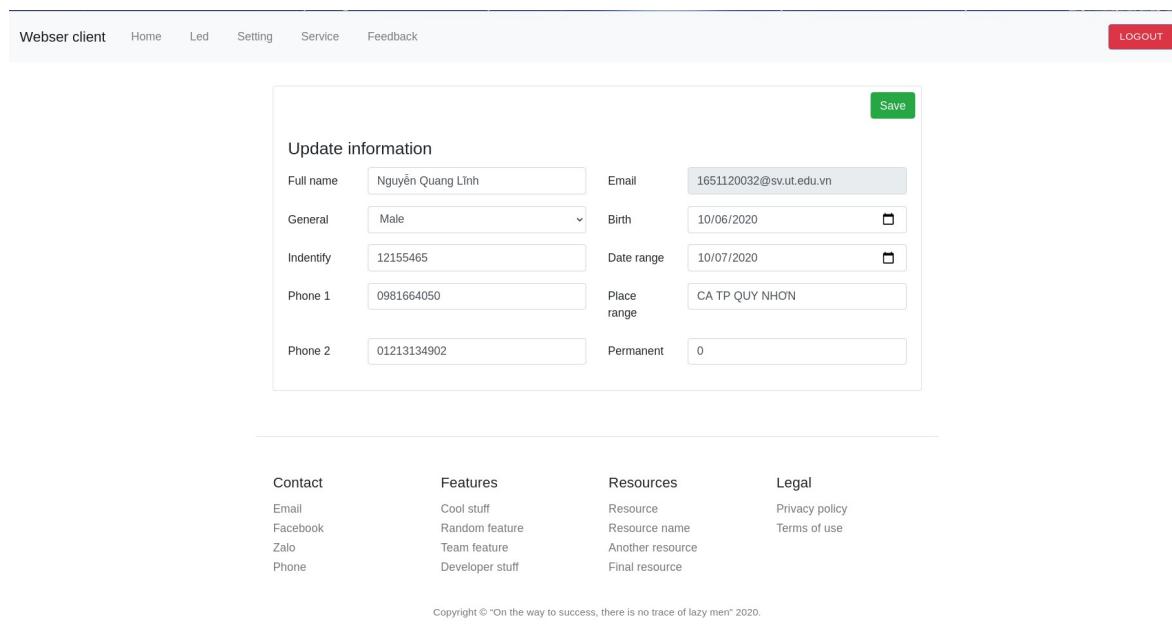
Hình 78: Giao diện điều khiển thiết bị của khách hàng

Màn hình xét thời gian



Hình 79: Giao diện xét thời gian cho thiết bị của khách hàng

Màn hình Setting



Hình 80: Giao diện cập nhật thông tin cá nhân của khách hàng

Màn hình Feedback - phản hồi thông tin

The screenshot shows a web application interface for 'Webser client'. At the top, there is a navigation bar with links: Home, Led, Setting, Service, and Feedback. On the far right of the header is a red 'LOGOUT' button. Below the header, the main content area has a title 'FEEDBACK' centered above a form.

The form consists of two input fields: 'Title' and 'Content', both with placeholder text ('Type here...'). Below the form is a blue 'Submit' button. Underneath the submit button is a section titled 'THE LIST HAS RESPONDED' containing a table.

Order	Title	Content	Status	Created at	
0	Test	test	Yes	2020-10-30 02:03	Detail Delete

At the bottom of the page, there is a footer section with four categories: Contact, Features, Resources, and Legal. Each category lists some items. The footer also includes a copyright notice: 'Copyright © "On the way to success, there is no trace of lazy men" 2020.'

Contact:
Email
Facebook
Zalo
Phone

Features:
Cool stuff
Random feature
Team feature
Developer stuff

Resources:
Resource
Resource name
Another resource
Final resource

Legal:
Privacy policy
Terms of use

Hình 81: Giao diện phản hồi của khách hàng

CHƯƠNG 4: INTERNET OF THINGS (IOT)

I. Cách nạp code vào module ESP8266

1. Module ESP8266 là gì ?

ESP8266 là một vi mạch điều khiển có thể giúp chúng ta điều khiển thiết bị điện tử thông qua internet bằng việc tích hợp sẵn module Wifi bên trong vi mạch. ESP8266 rất được ưa chuộng trong ngành tự động hóa.

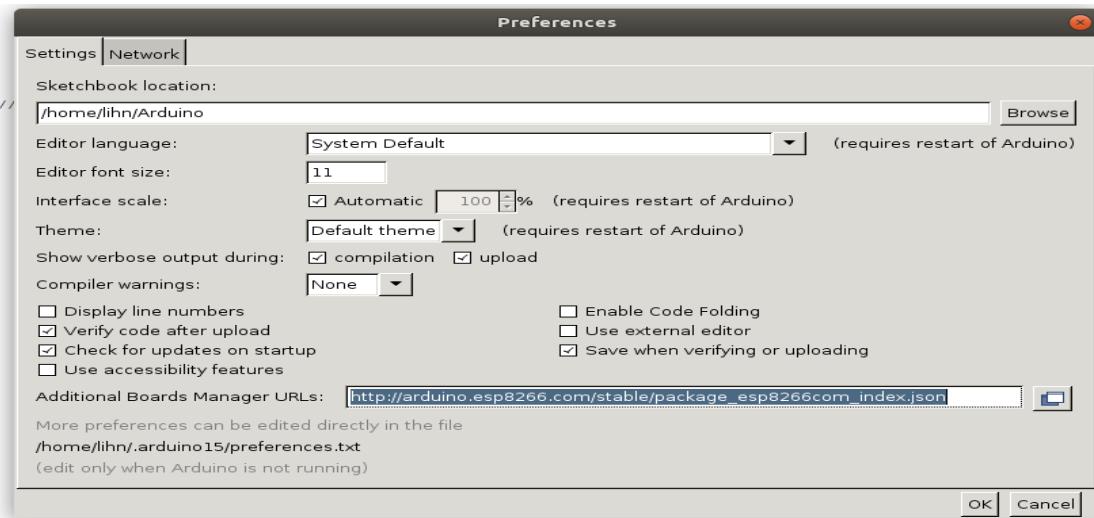
ESP8266 có nhiều phiên bản và mỗi phiên bản có một chút sự khác biệt riêng như ESP8266v1, ESP8266v7 và ESP8266v12. Tuy nhiên lại khá giống nhau về chức năng và khả năng lập trình.



Hình 82: Module Esp8266

2. Cách nạp code

Vào File → Preferences, sau đó thêm vào URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json



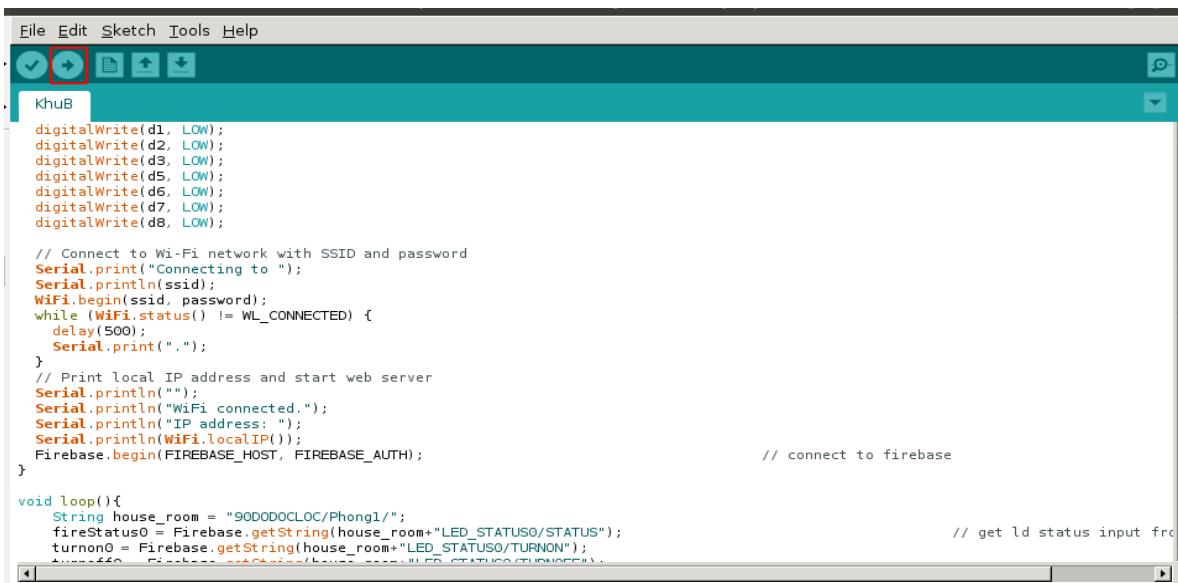
Hình 83: Cài đặt package cho arduino

Tiếp theo sẽ cài một số thư viện hỗ trợ như ESP8266WiFi.h, FirebaseArduino.h, ...
Tools → Boards Manager : Tìm kiếm thư viện ESP8266, Firebase ... và cài đặt



Hình 84: Cài đặt thư viện trong arduino

Sau khi cài đặt xong thì chúng ta sẽ kết nối cổng USB của máy tính với module ESP8266 và tiến hành nạp code bằng cách bấm nút “Upload” trong ô vuông đỏ hình chữ nhật.



```
File Edit Sketch Tools Help
KhuB
digitalWrite(d1, LOW);
digitalWrite(d2, LOW);
digitalWrite(d3, LOW);
digitalWrite(d5, LOW);
digitalWrite(d6, LOW);
digitalWrite(d7, LOW);
digitalWrite(d8, LOW);

// Connect to Wi-Fi network with SSID and password
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
// Print local IP address and start web server
Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // connect to firebase

void loop(){
    String house_room = "90DODOCLOC/Phong1/";
    fireStatus0 = Firebase.getString(house_room+"LED_STATUS0/STATUS");
    turnon0 = Firebase.getString(house_room+"LED_STATUS0/TURNON"); // get led status input from
    turnoff0 = Firebase.getString(house_room+"LED_STATUS0/TURNOFF");
}
```

Hình 85: Cách nạp code cho module Esp8266

Sau khi xong thì chúng ta sẽ nhìn thấy “Hard resetting via RTS pin ...” đã hoàn tất nạp code cho module ESP8266

```

KNUB | Arduino 1.8.13 Hourly Build 2020/03/23 01:25
File Edit Sketch Tools Help
KhuB
digitalWrite(d1, LOW);
digitalWrite(d2, LOW);
digitalWrite(d3, LOW);
digitalWrite(d5, LOW);
digitalWrite(d6, LOW);
digitalWrite(d7, LOW);
digitalWrite(d8, LOW);

// Connect to Wi-Fi network with SSID and password
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

// Print local IP address and start web server
Serial.println("");
Serial.println("Wi-Fi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP()); // connect to firebase
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

void loop(){
    String house_room = "QODODOCLOC/Phong1/";
    fireStatus0 = Firebase.getString(house_room+"LED_STATUS0/STATUS");
    turnOn0 = Firebase.getString(house_room+"LED_STATUS0/TURNON");
    // get led status input from
}

Done uploading.

TRACE +0.000 Read 1 bytes: 00
TRACE +0.000 Read 3 bytes: 136532
TRACE +0.001 Read 1 bytes: 3b
TRACE +0.000 Read 11 bytes: 4cf25299a15436d9d84b00
TRACE +0.001 Read 1 bytes: 00
TRACE +0.000 Read 1 bytes: c0
TRACE +0.000 Received full packet:
0113120000000000 97061365923b4cf2 | .....e2;L.
S299a15436d9d84b 0000 | R..T6..K..
Hash of data verified.

Leaving...
TRACE +0.001 command op=0x02 data len=16 wait_response=1 timeout=3.000 data=00000000000000004000000000000000
TRACE +0.000 Write 26 bytes:
c0000210000000000 00000000000000000000 | .....@.....
0000400000000000 00c0 | .....@.....
TRACE +0.003 Read 1 bytes: c0
TRACE +0.002 Read 3 bytes: 0020202
TRACE +0.001 Read 1 bytes: 00
TRACE +0.000 Read 7 bytes: 000000000c0
TRACE +0.000 Received full packet: 01020200000000000000
TRACE +0.000 command op=0x12 data len=4 wait_response=1 timeout=3.000 data=01000000
TRACE +0.000 Write 14 bytes: c000120400000000000100000c0
TRACE +0.002 Read 1 bytes: c0
TRACE +0.001 Read 1 bytes: 01
TRACE +0.000 Read 5 bytes: 1202000000
TRACE +0.001 Read 1 bytes: 00
TRACE +0.000 Read 4 bytes: 000000c0
TRACE +0.000 Received full packet: 01120200000000000000
Hard resetting via RTS pin...

```

Hình 86: Sau khi nạp code thành công

3. Module ESP8266 kết nối wifi

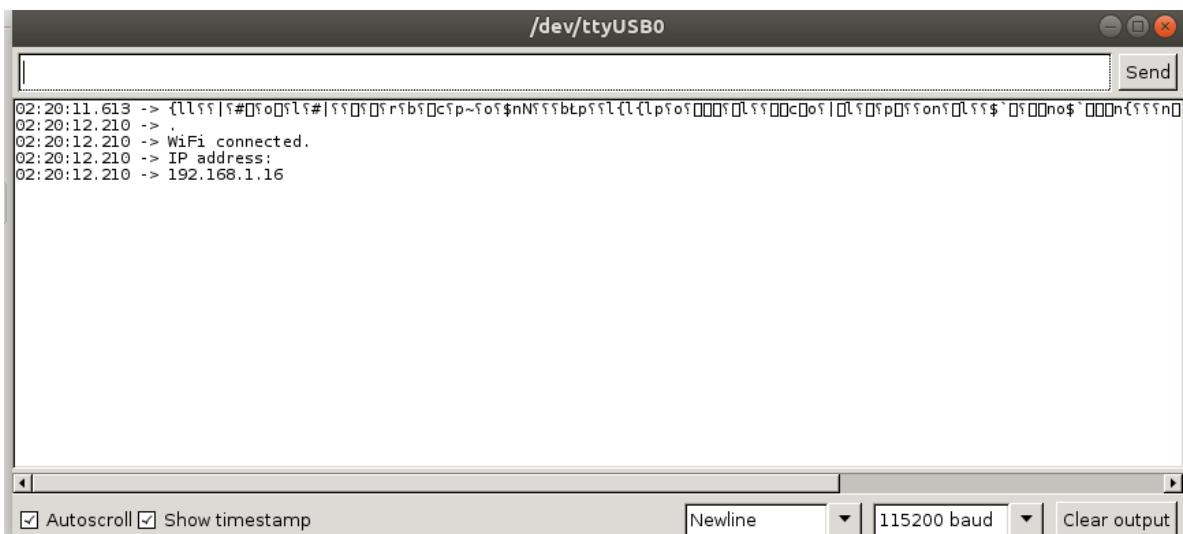
- Module ESP8266 kết nối wifi như thế nào ?
 - const char* ssid = “TenWifi”;
 - const char* password= “PasswordWifi”;
 - Trong phần void setup();
Serial.begin(115200);
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_ConNECTED) {
delay(500);

```

    Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

```



Hình 87: Hiển thị trạng thái kết nối với wifi thành công

b. Các thiết bị kết nối với module ESP8266 như thế nào ?

Chúng ta sẽ sử dụng các chân của module d0, d1, d2, d3, d5, d6, d7 và d8. Mỗi chân này tương ứng cho một giá trị FPIO pins: 16, 5, 4, 0, 14, 12, 13 và 15.

```

const int d0 = 16;
const int d1 = 5;
const int d2 = 4;
const int d3 = 0;
const int d5 = 14;
const int d6 = 12;
const int d7 = 13;
const int d8 = 15;

```

- Trong phần void setup()
 - Khởi tạo các biến đầu ra dưới dạng kết quả đầu ra


```
pinMode(d0, OUTPUT);
pinMode(d1, OUTPUT);
pinMode(d2, OUTPUT);
pinMode(d3, OUTPUT);
pinMode(d5, OUTPUT);
pinMode(d6, OUTPUT);
pinMode(d7, OUTPUT);
pinMode(d8, OUTPUT);
```
 - Xét tín hiệu đầu ra là tín hiệu thấp (LOW)


```
digitalWrite(d0, LOW);
digitalWrite(d1, LOW);
digitalWrite(d2, LOW);
digitalWrite(d3, LOW);
digitalWrite(d5, LOW);
digitalWrite(d6, LOW);
digitalWrite(d7, LOW);
digitalWrite(d8, LOW);
```
 - Trong vòng lặp loop()

Nếu chúng ta muốn thiết bị chân d0, d1, d2, d3, d5, d6, d7 và d8 sáng thì
`digitalWrite(d0, HIGH).` Ngược lại `digitalWrite(d0, LOW).`

4. Lưu trữ dữ liệu các thiết bị ở đâu ?

- Các dữ liệu, thuộc tính hay trạng thái của thiết bị sẽ lưu trữ trên cơ sở dữ liệu: realtime firebase.
- Kết nối cơ sở dữ liệu với module ESP8266
 - `#include <FirebaseArduino.h>`
 - `#define FIREBASE_HOST "iotpro-58c44.firebaseio.com"`

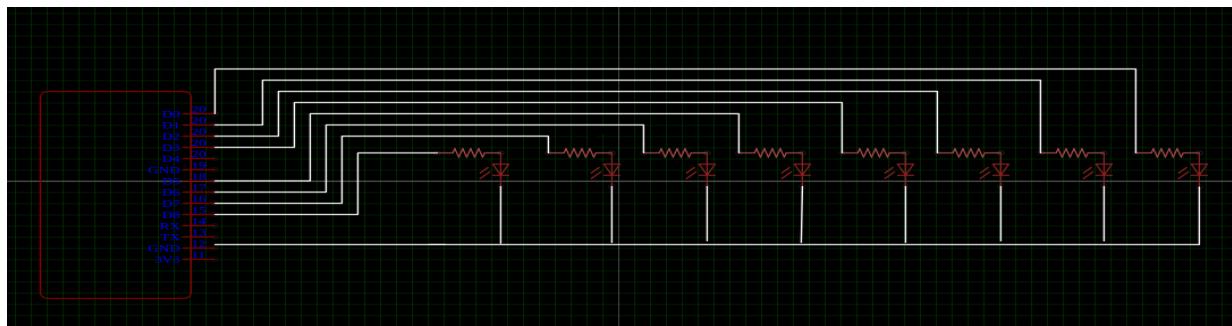
- #define FIREBASE_AUTH
"F4mMmNXp1CPYvJYX5KwtrLifqw6UvVO4fyCUKhoj"
- Trong phần setup()
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
- Trong phần loop()

```
String house_room = "90DODOCLOC/Phong1/";
firestatus0 = Firebase.getString(house_room+"led_status0/status");
turnon0 = Firebase.getString(house_room+"led_status0/turnon");
turnoff0 = Firebase.getString(house_room+"led_status0/turnoff");
if (firestatus0 == "on") {
    digitalWrite(LED_BUILTIN, LOW);
    digitalWrite(d0, HIGH);
}
else if (firestatus0 == "off") {
    digitalWrite(d0, LOW);
}
else {Serial.println("Wrong Credential!"); }
```

II. Mô hình lắp đặt

1. *Cách lắp đặt*

- a. Chuẩn bị dụng cụ
 - 1 Module ESP8266
 - 1 Dasdboard
 - 8 Điện trở
 - 8 Đèn led
 - 20 Dây cắm vào dashboard
 - 1 Dây cáp kết nối nguồn điện cho ESP8266
- b. Lắp đặt mô hình



Hình 88: Mô hình kết nối với led

Giải thích: Từ trái sang phải

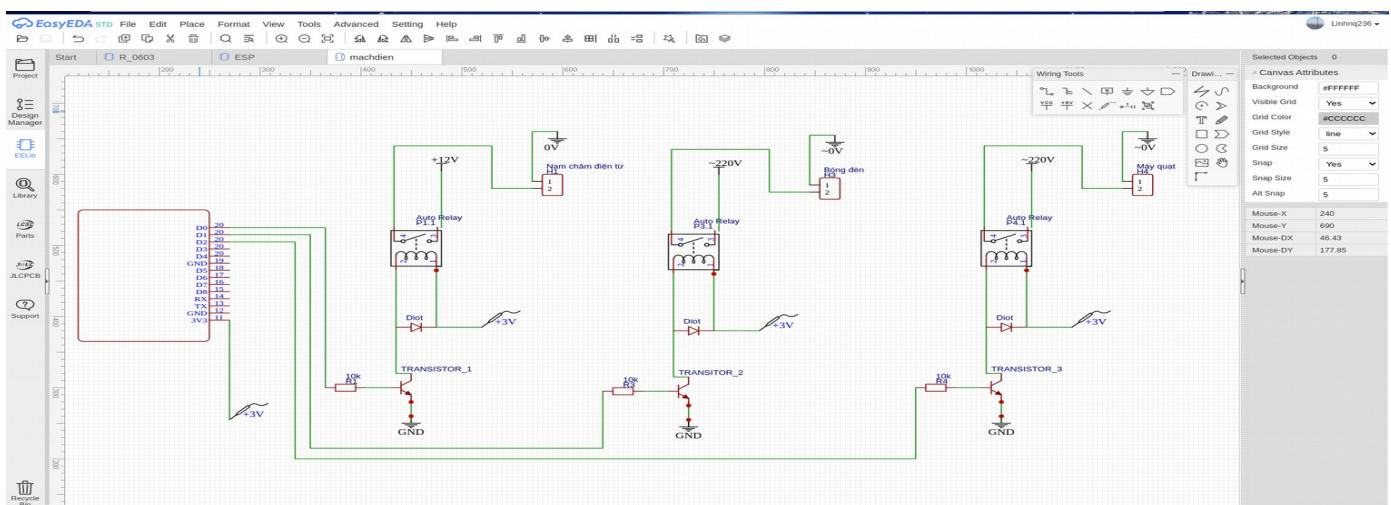
- Module ESP8266
- Dây cáp kết nối
- Điện trở
- Đèn led

Quy trình lắp:

Tất cả cực âm của đèn led đều kết nối với đầu GND của mạch.

Các cực dương của đèn led đều kết nối qua 1 đầu của điện trở. Sau đó, đầu còn lại của điện trở sẽ kết nối với các Pin trong mạch tương ứng đèn từ trái sang phải ứng với mã Pin D0, D1, D2, D3, D5, D6, D7 và D8.

c. Chuyển đổi mô hình ra với nguồn điện 220v



Hình 89: Sơ đồ mạch điện mô hình thực tế

PHẦN 3: ĐÁNH GIÁ VỀ HƯỚNG PHÁT TRIỂN

I. Đánh giá và kết luận

Đánh giá

- Thuận lợi
 - Với sự tận tình, chu đáo của thầy hướng dẫn trong quá trình phát triển dự án cũng như trong quá trình viết tài liệu cho dự án.
 - Cùng với sự hỗ trợ từ các anh, chị, em, bạn bè trong quá trình bắt đầu dự án và kết thúc.
 - Quản lý source code trên github một cách hiệu quả.
- Khó khăn
 - Thiếu kinh nghiệm trong việc phát triển một dự án về tính chuyên nghiệp cũng như là tài liệu chuẩn bị trước, trong và sau dự án.
 - Thời gian phát triển dự án còn hạn chế so với kế hoạch đã vạch ra.
 - Tích hợp với IoT nên cần phải có tìm hiểu thêm kiến thức bên điện tử.

Kết luận

Dự án bắt đầu từ tháng 08-2020 và kết thúc nay đã kết thúc một cách thuận lợi.

Mặc dù có nhiều khó khăn, thử thách trong việc phát triển nhưng cuối cùng cũng đi đến hoàn thiện sản phẩm.

II. Hướng phát triển

Sắp tới:

- Web: Tập trung vào nâng cấp, phát triển các vấn đề sau.
 - Nhiều chức năng vẫn còn chưa được thực hiện hoặc đã phát triển chưa xong.
 - Giao diện người quản lý còn đơn giản.
- App: Tiếp tục nâng cấp, bảo trì
 - Các màn hình còn đơn giản.
 - Các chức năng còn chưa hoàn thiện
- IoT: Tiếp tục nâng cấp.
 - Thiết các mạch điện còn khá đơn giản, các thiết bị được kết nối với module Esp8266 còn khá phức tạp chưa rõ ràng.

Trong tương lai:

- Web:
 - Phát triển thêm chức năng kiểm soát người vào nhà trọ,..
 - Kết hợp với các hệ thống khác.
- App:
 - Phát triển chức năng điều khiển bằng giọng nói.
 - Tích hợp thanh toán online.
- IoT:
 - Cảm biến ánh sáng, cảm biến nhiệt độ, nhận diện giọng nói ... để cho các thiết bị hoạt động một cách tự động.

TÀI LIỆU KHAM KHẢO

[Tìm hiểu về Ruby on Rails, tác giả: Hồng Anh, link: <https://itviec.com/blog/ruby-on-rails-la-gi>]

[Tìm hiểu về Ruby on Rails, tác giả:Lý Tất Thành , link: <https://ngoisoso.vn/blog/6-Ly-Do-Tai-Sao-Ban-Nen-Hoc-Ruby-on-Rails-2104.html>]

[StarAdmin-Free-Bootstrap-Admin-Template, tác giả: Bootstrap4, link: <https://www.bootstrapdash.com>]

[Tìm hiểu về Ruby on Rails, tác giả: Lý Đức Long , link: <https://viblo.asia/p/uu-nhuoc-diem-cua-ruby-on-rails-eXaRlwDoGmx>]

[Cài đặt Rails, tác giả: GoRails, link: <https://gorails.com/setup/ubuntu/20.04>]

[Cấu trúc thư mục Rails, tác giả: Ho Doan Ba Thien, link: <https://viblo.asia/p/cau-truc-folder-mot-du-an-rails-ByEZk0GylQ0>]

[Tìm hiểu về React Native, tác giả: NORDIC CODER , link: <https://nordiccoder.com/blog/tong-quan-ve-react-native>]

[Tìm hiểu về Api, tác giả: TopDev , link: <https://topdev.vn/blog/api-la-gi>]

[Tìm hiểu về React, tác giả: Hải G.,link: <https://www.hostinger.vn/huong-dan/react-la-gi-va-no-hoat-dong-nhu-the-nao>]

[-Ưu điểm và nhược điểm React, tác giả: Hoang Trong Hieu , link: <https://viblo.asia/p/reactjs uu-diem-va-nhuoc-diem-V3m5WzexlO7>]

[Tìm hiểu về IoT, tác giả: Dat Nguyễn, link: <https://gocinfo.com/cai-dat-arduino-ide-tren-ubuntu.html>]

[Ứng dụng IoT, tác giả: Long Vân System Solution , link: <https://longvan.net/iot-la-gi-ung-dung-cua-iot-trong-thuc-tien.html>]

[Cài đặt Arduino và lập trình Arduino Esp8266, tác giả: cộng đồng Arduino Việt Nam, link: <http://arduino.vn/bai-viet/1496-esp8266-ket-noi-internet-phan-1-cai-dat-esp8266-lam-mot-socket-client-ket-noi-toi#:~:text=ESP8266%20l%C3%A0%20m%E1%BB%99t%20m%E1%BA%A1ch%20vi,con%20vi%20%C4%91i%E1%BB%81u%20khi%E1%BB%83n%20ch%C3%ADnh.>]

[Control 12VDC Solenoid Door Lock Using A Relay On Arduino, tác giả: CytronTech, link: <https://www.youtube.com/watch?v=0wYp7rd4IIo&list=LL&index=1>]

[ESP8266 y FIREBASE Conexión Leer y Escribir Datos [RealTime Database - IoT], tác giả: Alex7 Tutoriales, link: <https://www.youtube.com/watch?v=5ezyekFaVJI>]