****

****

**BỘ GIAO THÔNG VẬN TẢI**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**LUẬN VĂN TỐT NGHIỆP**

**NGHIÊN CỨU TÍCH HỢP IOT VÀO HỆ THỐNG QUẢN LÝ NHÀ TRỌ**

**Giảng viên: ThS. VŨ ĐÌNH LONG**

**Sinh viên thực hiện:**

**Nguyễn Quang Lĩnh – 1651120032 - CN16A**

Hồ Chí Minh, ngày tháng năm 2020



**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

TP.HCM, ngày … tháng … năm 2020

Giáo viên hướng dẫn

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

TP.HCM, ngày … tháng … năm 2020

Giáo 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 báo. 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 mức 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 em, động viện em, 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 và các bạn thành công trong học tập cũng như là tương lai.

Do sự hạn chế về mặt thời gian, kiến thức cũng như kinh nghiêm nghiên cứu về luận văn tốt nghiệp không trách được những thiếu sót và hạn chế nhất định. Em rất mong được sự đóng góp ý kiến của quý thầy cô giáo cũng như các bạn đọc để báo cáo luận văn tốt nghiệp ngày càng hoàn thiện hơn.

LỜI NÓI ĐẦU

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

Đứng trước vai trò của thông tin hoạt động cạnh tranh và 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 phầm 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 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 khi quản lý nhà trọ, em đã lên ý tưởng “ Nghiên cứu IoT về quản lý nhà trọ”. Với ý tưởng này sẽ giúp cho những chủ trọ quản lý trọ của mình một cách thuận tiện, còn người dùng có thể sử dụng điện một cách tiết kiệm có hiệu quả hơn.

MỤC LỤC

PHẦN I: TÌM HIỂU CHUNG VỀ CÔNG NGHỆ SỬ DỤNG

1. TÌM HIỂU VỀ RUBY ON RAILS TRÊN MÔI TRƯỜNG LINUX

Khái niệm

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 dể 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 mỡ 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 dung.

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.

* + - 1. 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 dể 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 dể tiếp cận, về căn bản code của Rails rất dể dọc - tính năng này rất hữu ích khi chuyển giao dự án cũng như nhiều bên tham gia vào dự án.

* + - 1. 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.

* + - 1. 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
         * /ect/apt/sources.list.d/yarn.list
         * sudo apt-get update
         * sudo apt-get install git-core zlib1g-dev build-essential libss1-dev
         * libreadline-dev libyaml-dev libsqlite3-dev sqlite3 libxml2-dev libxslt1-
         * dev libcurl4-openssl-dev ssoftware-propertie­s-common libffi-dev nodejs
         * yarn
         1. 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
         1. Cài bundler
         * gem install bundler
         1. Cài Rails
         * gem install rails -v 6.0.2.2
         * rbenv rehash
         * rails -v

II. Cấu trúc của Rails

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 snack\_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 ngọn 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/mailer\_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 tắt 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 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 từng 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 chaỵ 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.

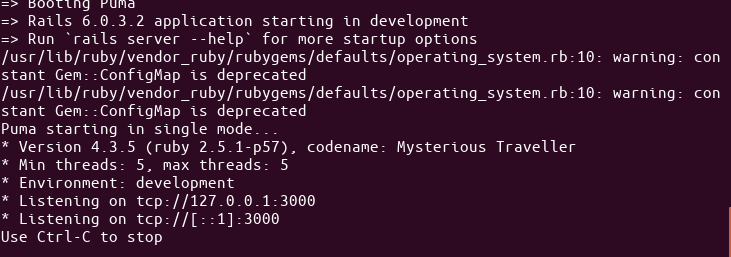
lib/assets: Chứa các library assets có thể hiểu như là các images, scripts, stylesheets mà không có ở ứng dụng nói cách khác là những cái bên ngoài chúng ta đưa vào.

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 dộng theo giờ bằng crontab.

* + - * + Log: Chứa tòan 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 ơ đây sẽ thành một phần của assets pipeline một cách tự động.

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



* + - * + 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

1. TÌM HIỂU VỀ REACT NATIVE

Khái niệm

React native là gì ?

React native là một framework do công ty công nghệ nổi tiếng của facebook phát triển nhầm 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 nối 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 gì ?

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 gì ?

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 vấn đề hiệu năng sẽ bị ảnh hưởng đáng kể cũng như không tương tác được kết hợp với tài nguyên hệ thông phần mềm cũng như phần cứng.

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.

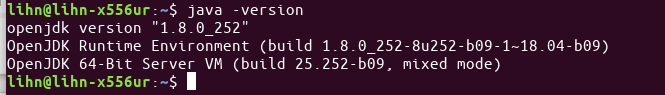
Bỏa mật không cao do dựa trên JS

Quản lý bộ nhớ.

Khả năng tùy biến cũng không thật sự tốt đối với một vài module.

Cài đặt môi trường

* + Reactive native co 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 đặt :curl-o-https://raw.githubusercontent.com/creationix/nvm/v.0.33.8/install.sh | bash
        + reload bash: source ~/.bashrc
        + Kiểm tra: nvm –version
      * Cài Node:
        + Cài đặt version mới nhất: nvm install node
        + Đặt mặt 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



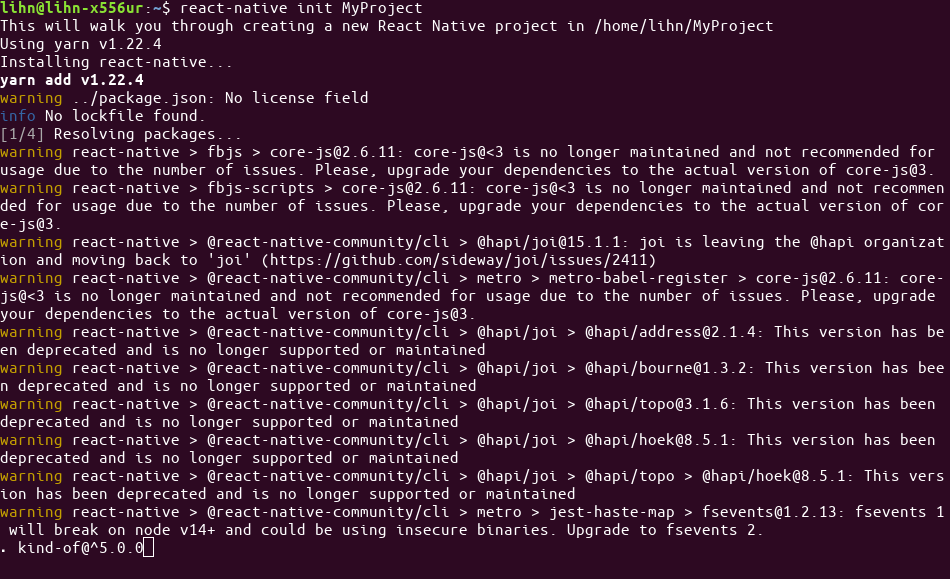
* + - * Cài đặt react-native
        + Mở terminal
        + npm install -g react-native-cli

Cấu trúc của react native

Tạo một ứng dụng với react-native

Mở terminal

react-native init MyProject



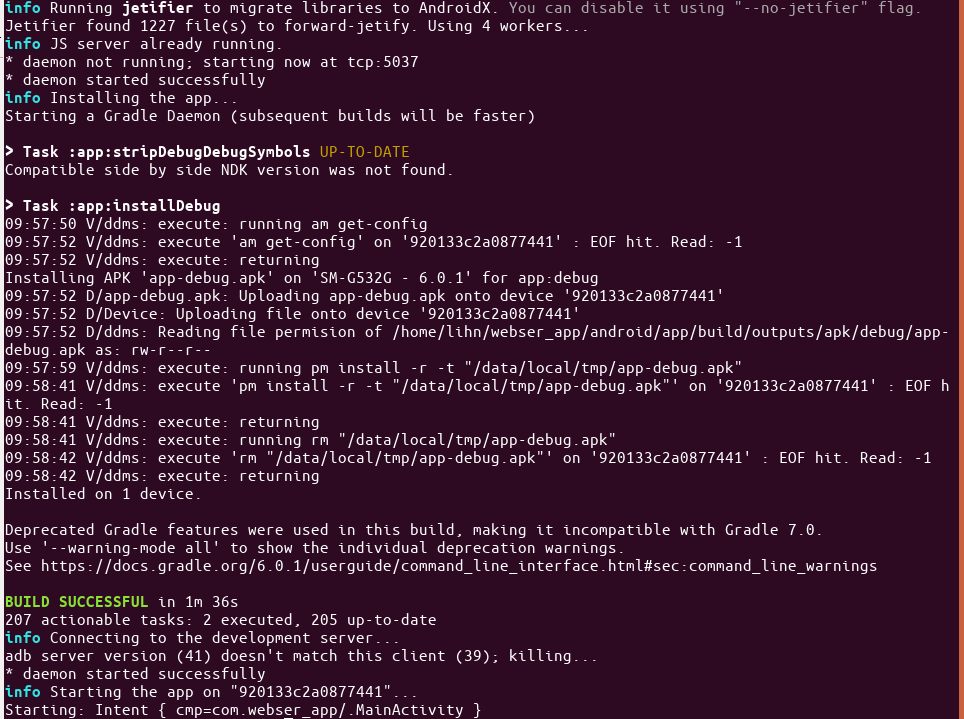
cd MyProject

Vào thư mục android tạo file local.properties với nội dung: sdk.dir = /home/<USER>/Android/Sdk/

Chạy các packager: react-native start

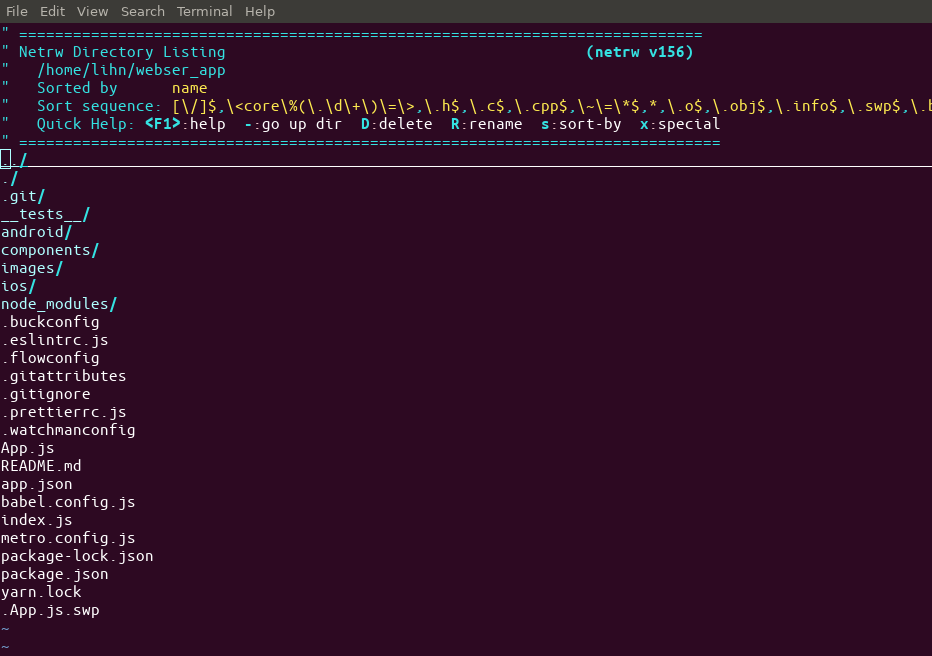


Mở 1 màn hinh terminal khác chạy trên android: react-native run-android



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.

Xây dựng cấu trúc thư mục



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.

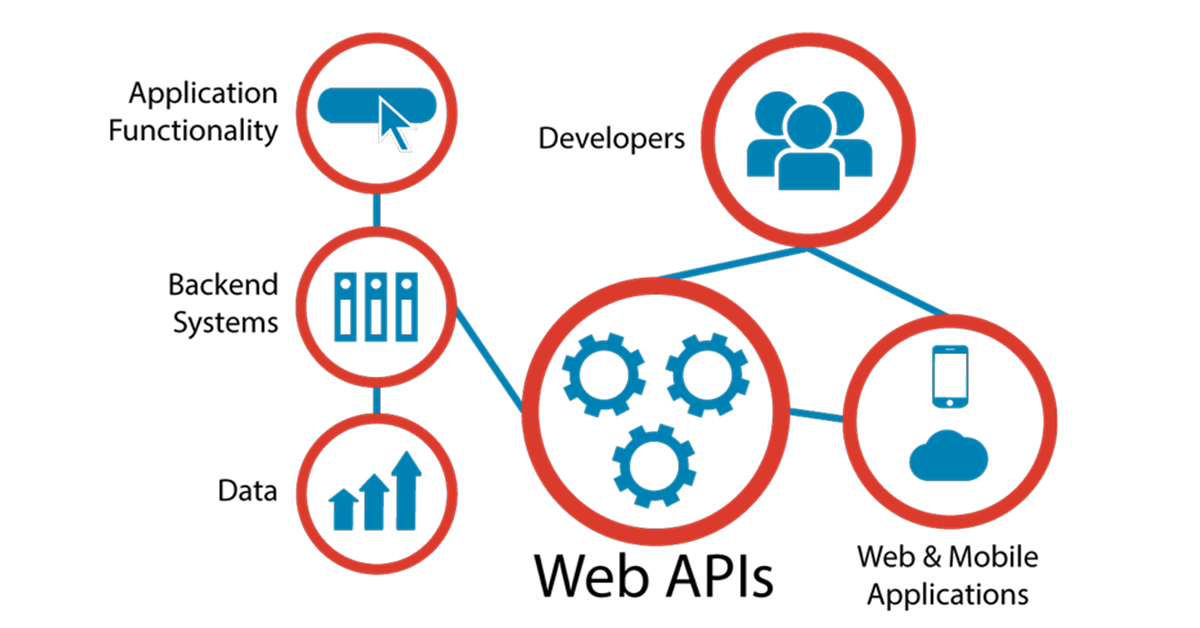
node\_modules: Nơi chứa các thư viện hỗ trợ trong quá trình sử dụng.

App.js: Kết nối các màn hình lại với nhau.

Package: Là nơi lưu trữ các thư viện đã được cài đặt.

1. TÌM HIỂU VỀ API
   * + 1. Khái niệm

API là các phương thức, giao kết nối với các thư viện và ứng dụng khác. Nó viết tắt của Application Programing 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.



* + - 1. 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 pần mềm ứng dụng có tể 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 thử theo tiêu chuẩn REST và HTTP, tạo sự thân thiện dể sử dụng với nhà phát triển. Giúp người dùng dể dàng truy cập. Web API hiện đại dùng các đối tương cụ thể như mobile developer với document, version khác nhau.

* + - 1. 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 xậy 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 dể dàng nếu được cho phép. Ngoài ra việc chia sẽ thông tin được chọn nhưng vẫn tránh được các yêu cầu không mong muốn.

* + - 1. 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 ...

* + - 1. 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.

1. TÌM HIỂU VỀ IOT

Khái niệm

IoT là gì ?

Internet of Things hay IoT là một thiết bị vật lý được kết nối với internet thu nhập và chia sẽ 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 thông 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.

Tầm quan trọng của IoT

Khi bất cứ thiết bị nào kết nối đươc với internet có nghĩa là nó có thể chia sẽ và nhận thông tin hoặc ngược lại. Với việc chia sẽ 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. Ngay bây giờ bạn có thể nghe bất kỳ bài hát nào trên khắp thế giới, không phải vì điện thoại bạn co thể lưu trữ đươc 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.

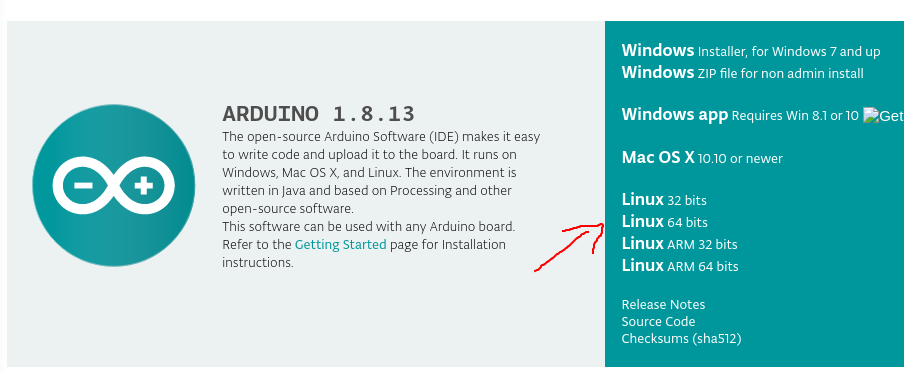
Loại nhận được thông tin và sau đó hành động: Như máy in. Sau khi tiếp nhận thông tin từ thiết bị khác như máy tính thì nó sẽ từ thực hiện hành động in.

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.

Cài đặt

Cài đặt môi trường làm việc

Lên trang chủ arduino để download về cho linux



Sau khi download bạn vào thư mục Dowloads và giải nén.

Bạn mở terminal

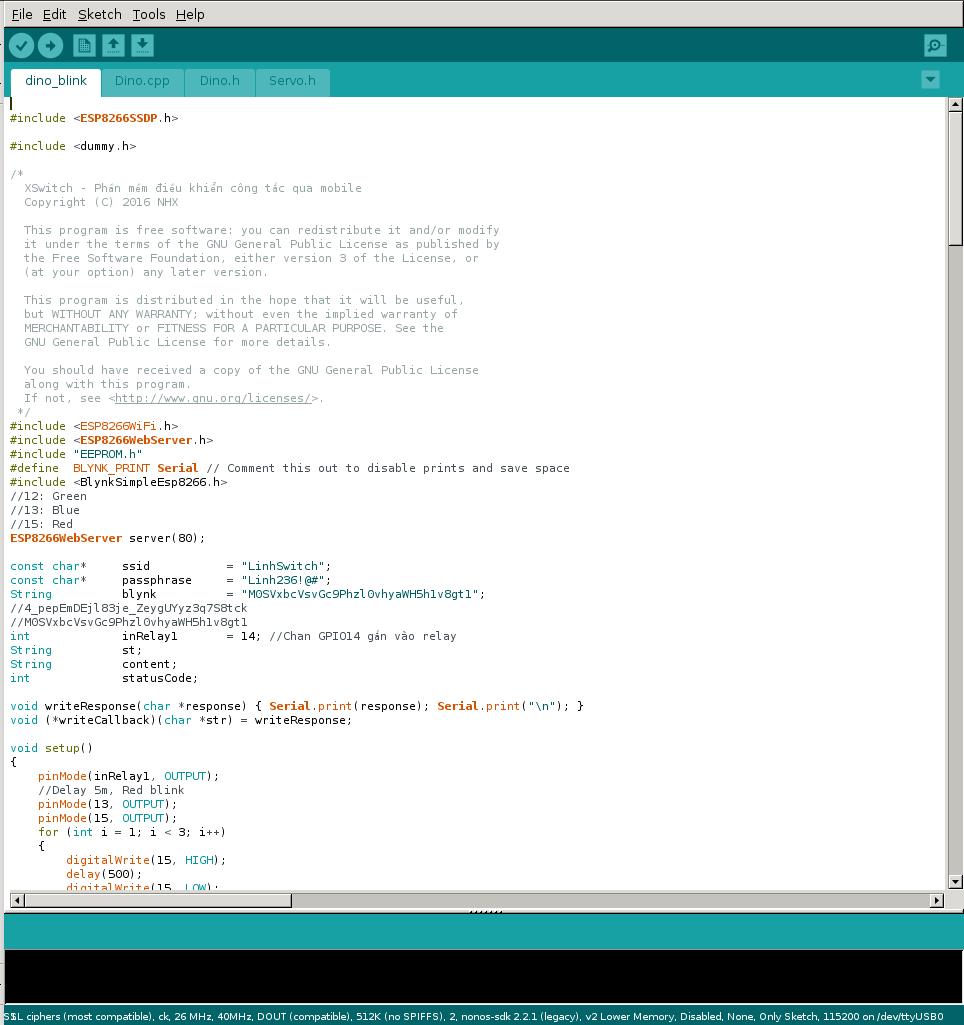
Gõ cd/Downloads/arduino-1.X.X-linuxYY

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: XÂY DỰNG HỆ THỐNG QUẢN LÝ PHÒNG TRỌ

* + 1. ỨNG DỤNG CHẠY TRÊN WEBSITE

Phân tích hệ thống

* 1. Xác định yêu cầu
  + Đặt phòng trọ:
    - Người thuê trọ đặt phòng đến thuê phòng trọ.
    - Người quản lý tiếp nhận từ khách hàng.
  1. Đặc tả yêu cầu
  2. Yêu cầu chức năng
     + Khi khách thuê đánh 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.
  3. 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.
  4. Xây dựng biểu đồ chức năng hệ thống
     1. Tác nhân
        + Có 2 tác nhân tham gia và tương tác với hệ thống: Quản trị viên và quản trị viên 1 khu vực.
     2. Chức năng hệ thống
        + Quản trị viên 1 khu vực:
          - 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 thuê.

Tài khoản người thuê (dùng để đăng nhập điều khiển đèn trên app): mở hoặc khóa tài khoản.

Phản hồi từ người thuê: Trả lời phản hồi từ khách thuê.

Thanh toán: Cập nhật thanh toán phòng hàng tháng.

* + - * + Tạo nhắc nhở: Thêm, sửa, xóa nhắc nhở.
        + Thống kê phòng và người thuê phòng.
        + Gửi mail cho khách thuê: Tại 1 khu vực.
        + Điều hướng web + điều khiển đèn bằng giọng nói.
    1. Quản trị viên:
       - * Đăng nhập
         * Đăng xuất
         * Quản lý thiết bị: Bật, tắt và hẹn giờ thiết bị tất 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.

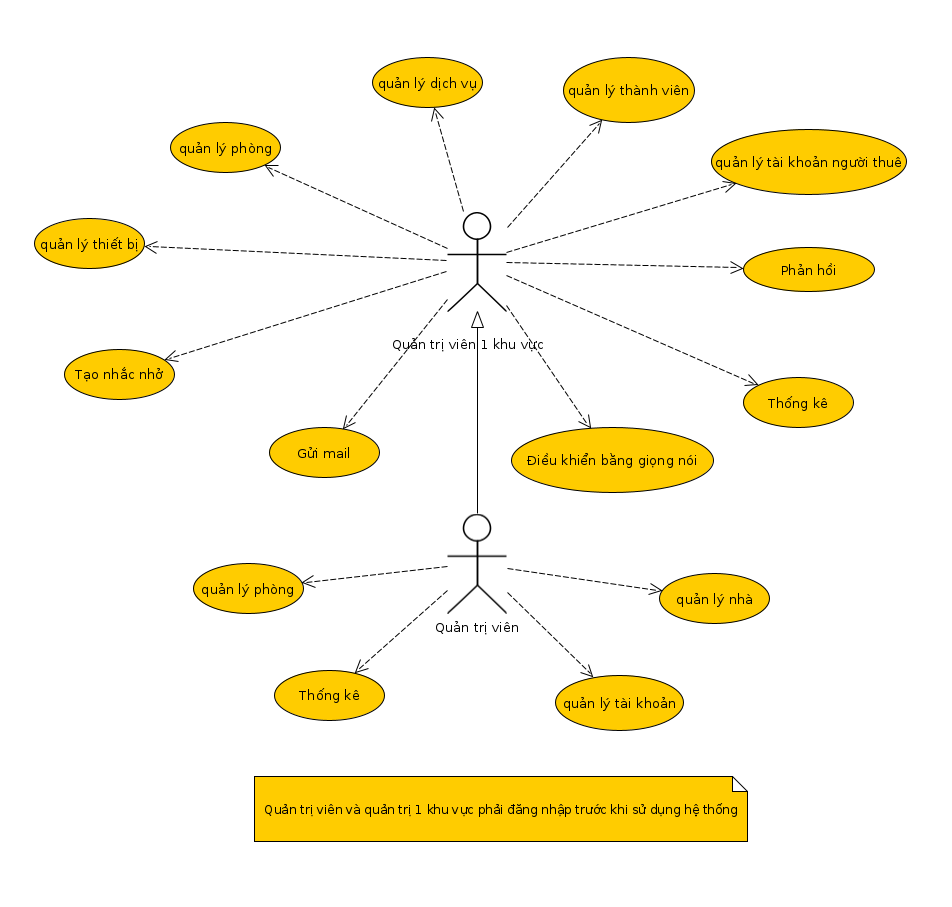
Thanh toán: Tất cả các khu vực.

Gửi mail cho khách thuê: Tất cả các khu vực.

Thống kê: Tiền trọ từng tháng, tất cả người thuê, phòng trống hay không trống.

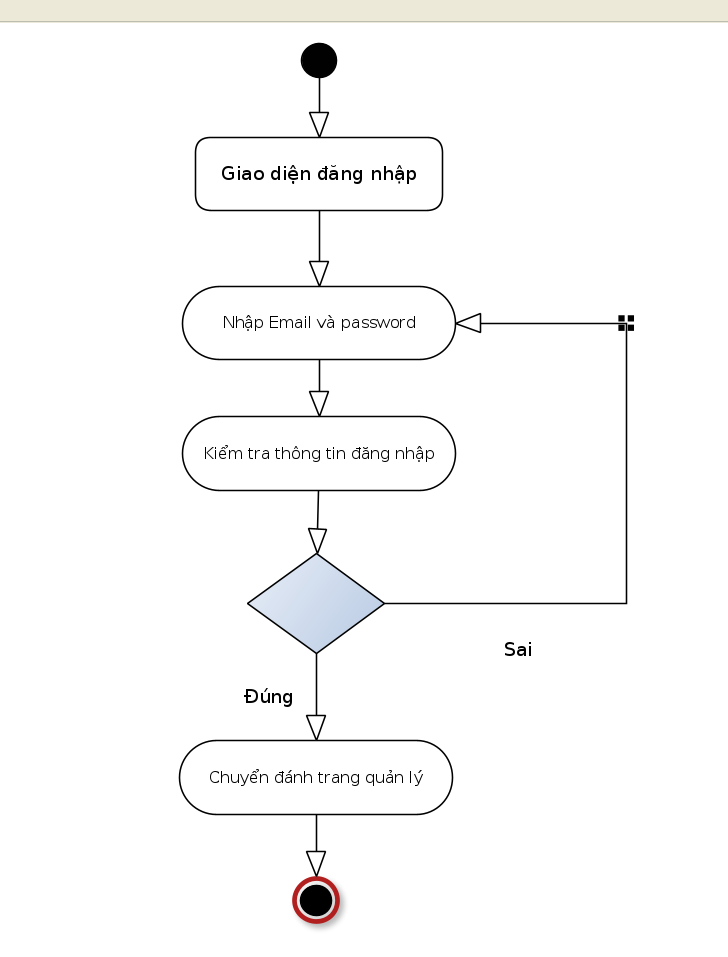
Điều hướng web + điều khiển đèn bằng giọng nói.

→ Từ việc phân tích các đối tượng ở trên ta có mô hình phân ra chức năng như hình 1



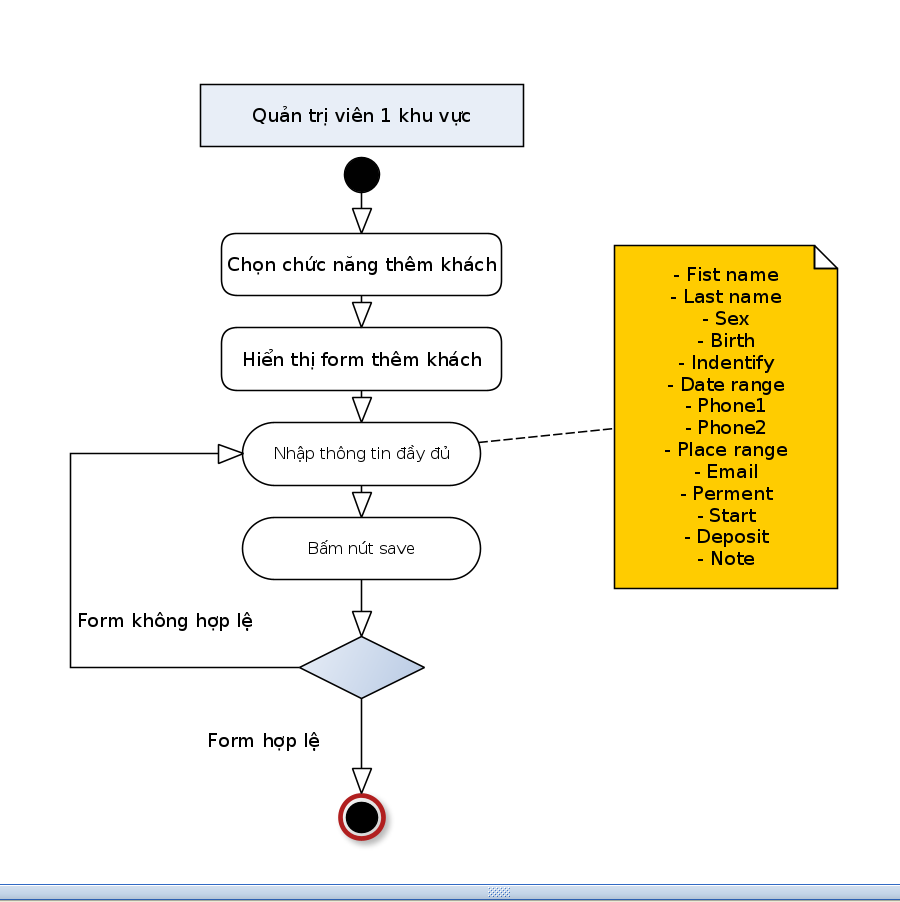
Hình 1: Biều đồ phân rã chức năng hệ thống

* 1. Biểu đồ hoạt động
     1. Biểu đồ hoạt động cho chức năng đăng nhập
        + - Để thực hiện trong việc quản lý nhà trọ thì quản trị viên 1 khu vực cũng như là quản trị viên phải đăng nhập vào hệ thống.

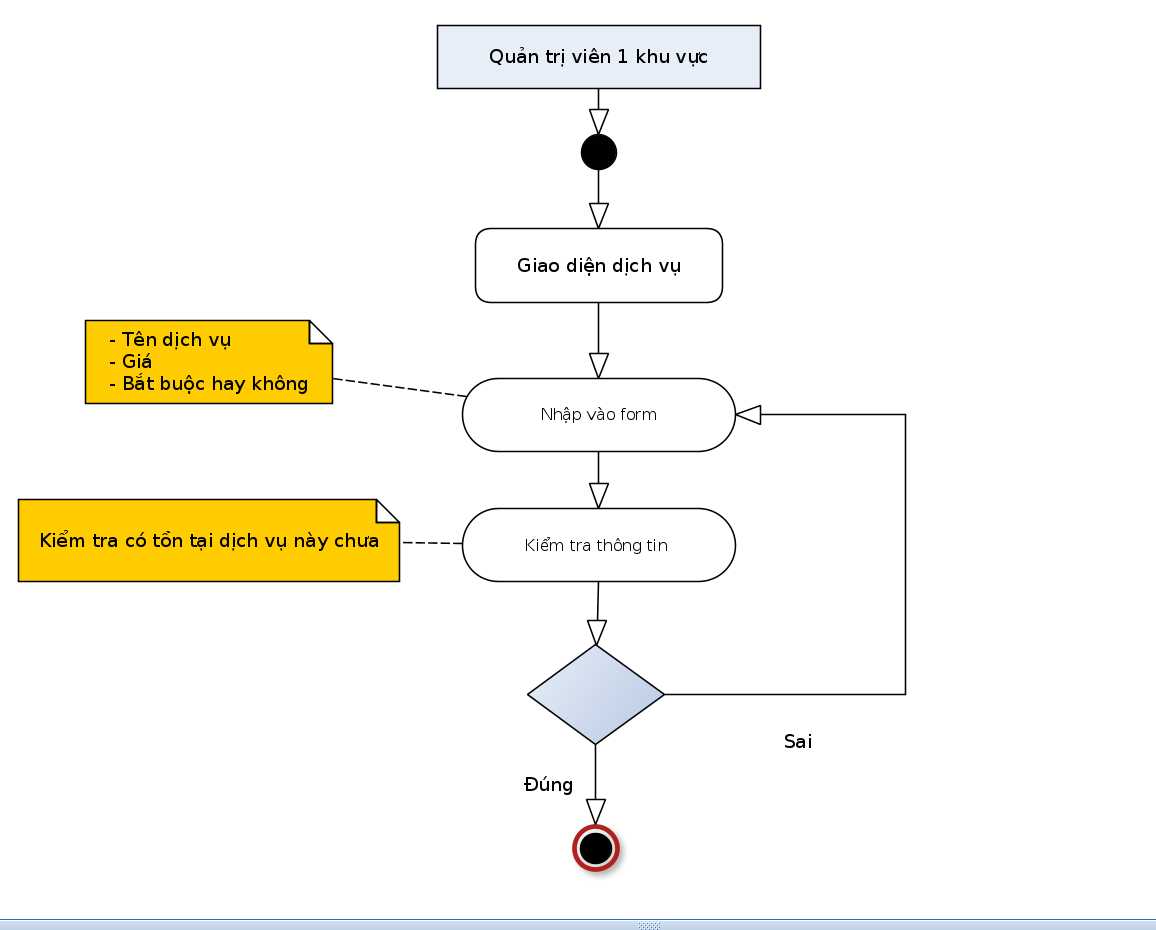


Hình 2: Biểu đồ chức năng đăng nhập của quản trị viên và quản trị viên 1 khu vực

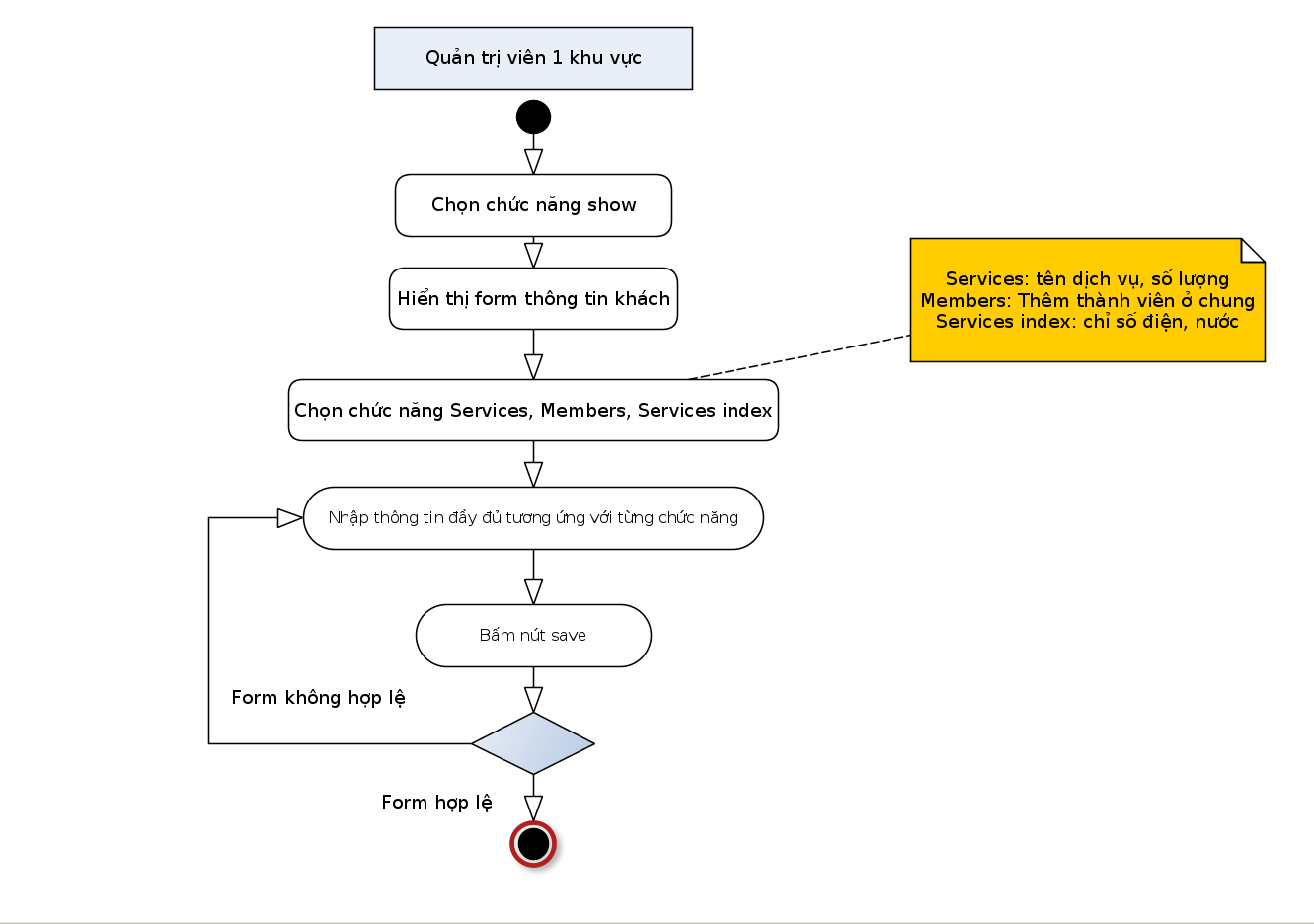
* + 1. Biểu đồ cho chức năng thêm một khách thuê mới



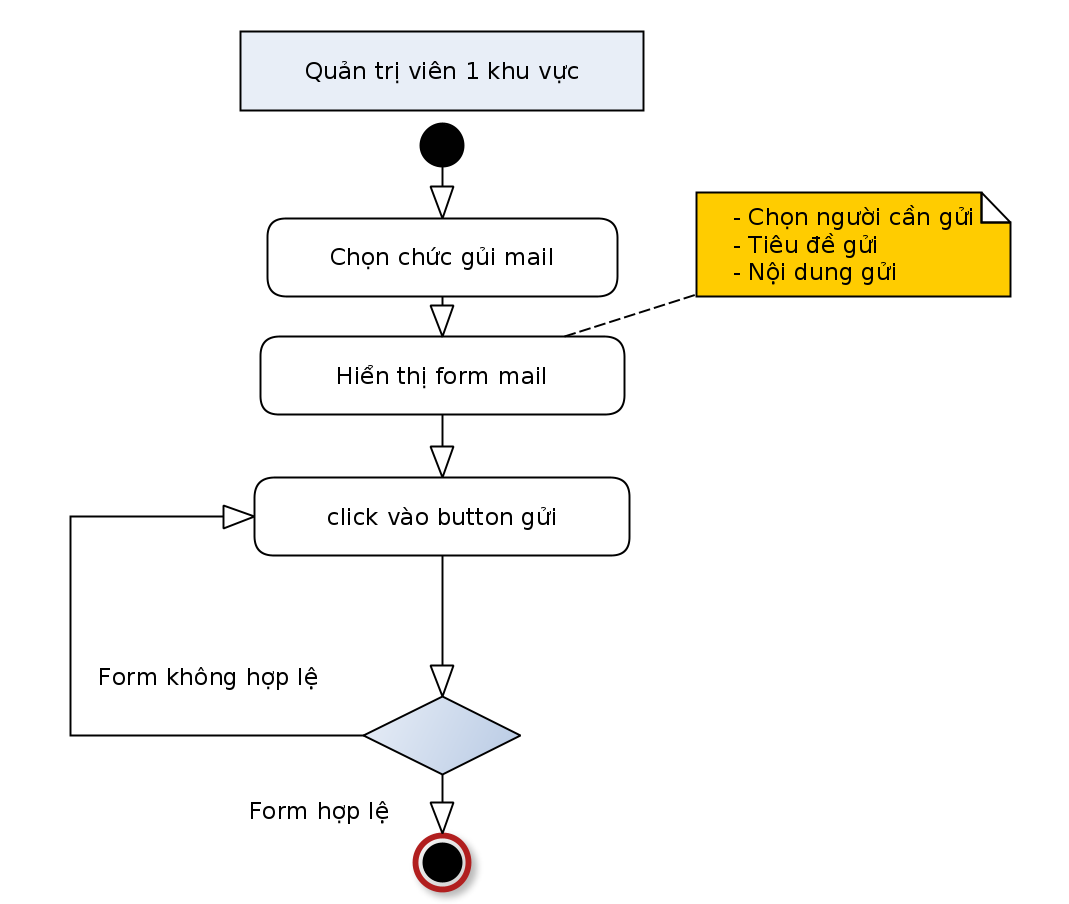
* + 1. Biểu đồ cho chức năng thêm mới dịch vụ



* + 1. Biểu đồ cho chức năng cập nhật dịch vụ, thành viên, chỉ số dịch vụ



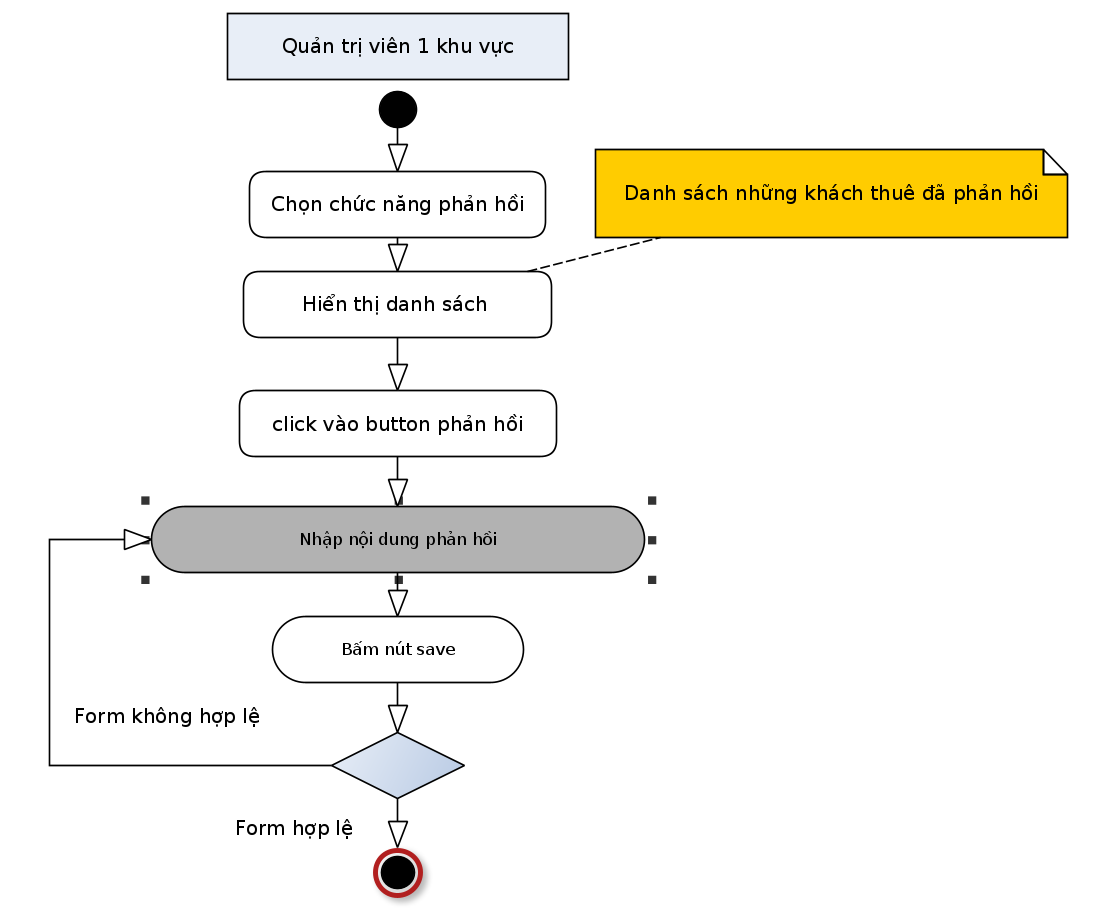
* + 1. Biều đồ cho chức năng gửi mail



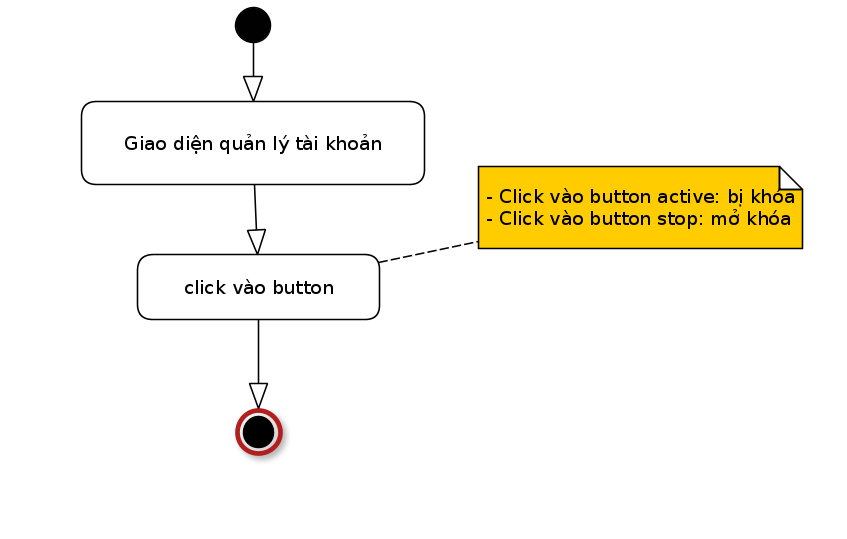
* + 1. Biểu đồ cho chức năng thêm mới một nhắc nhở



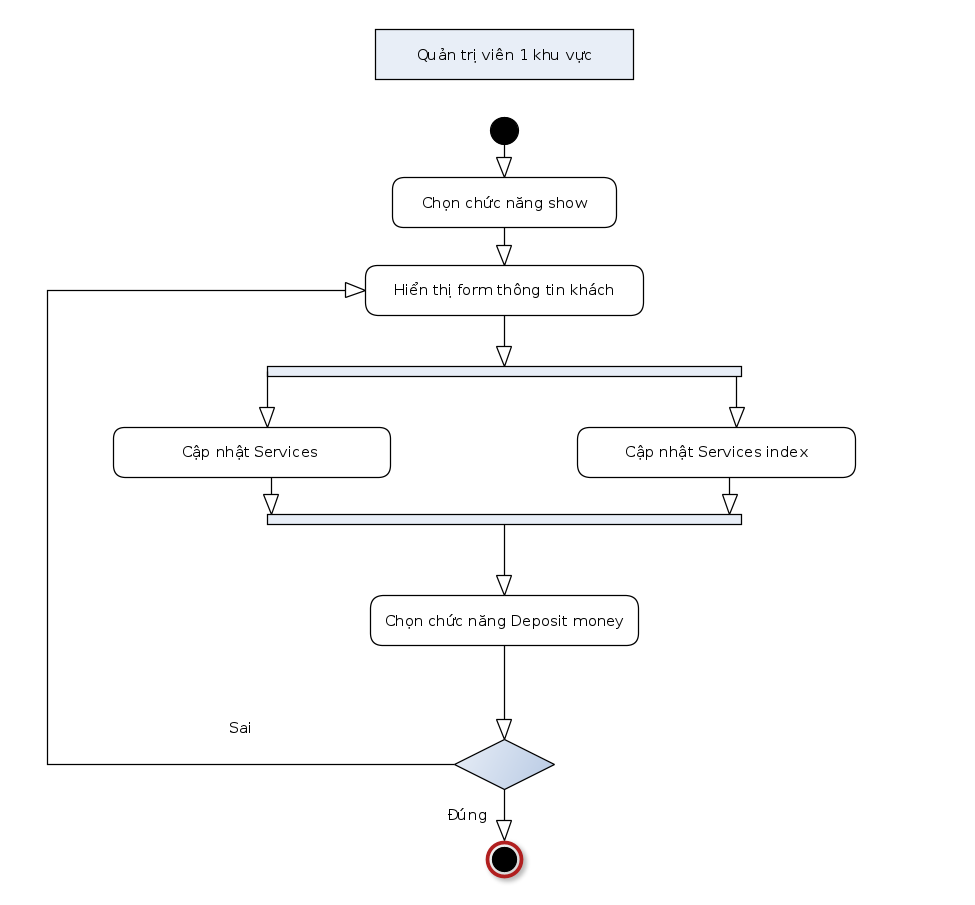
* + 1. Biểu đồ cho chức năng phản hồi ý kiến khách hàng qua từ bên app



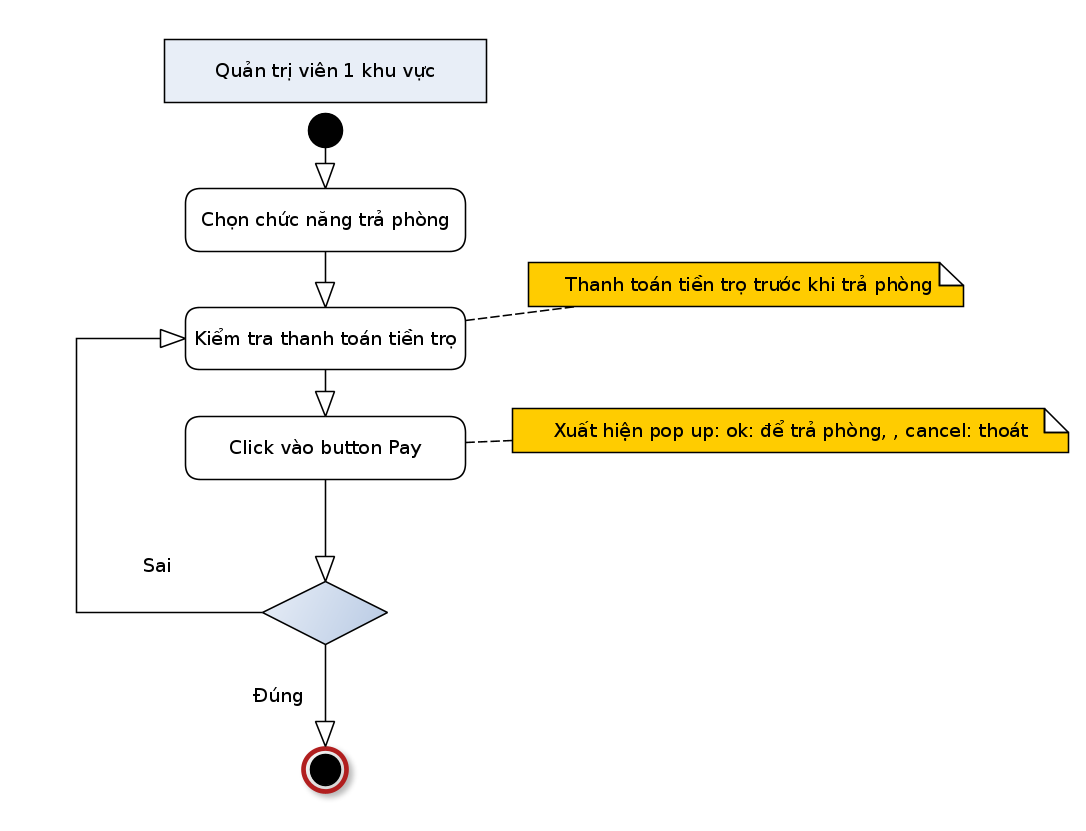
* + 1. Biểu đồ cho chức năng quản lý tài khoản.



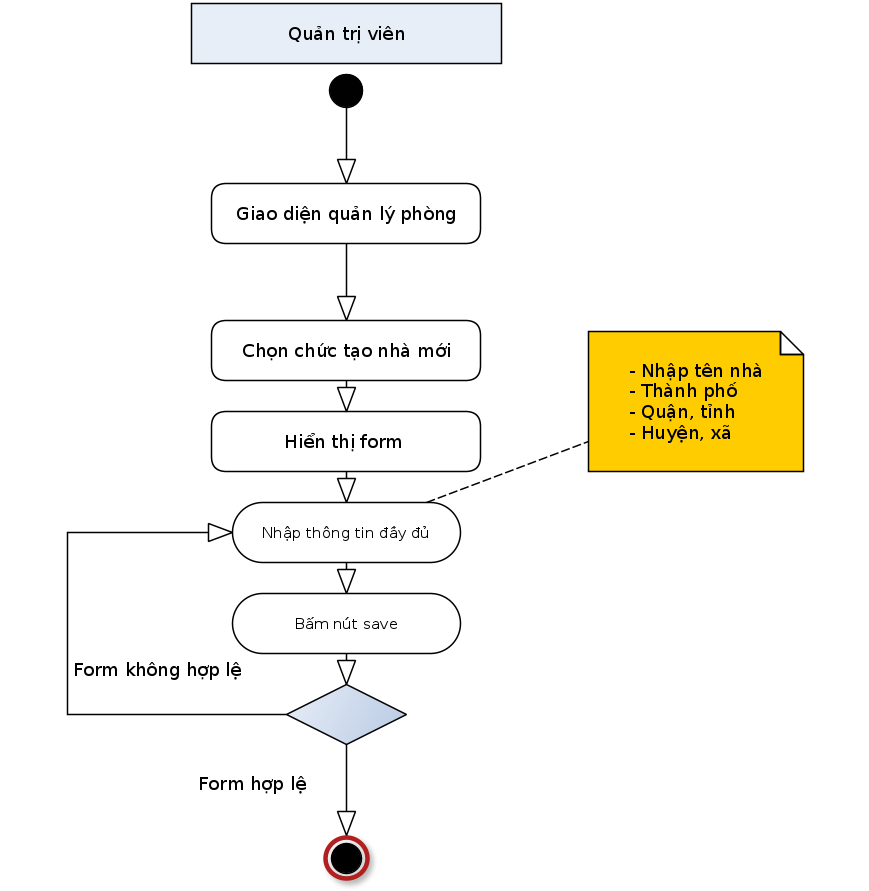
* + 1. Biểu đồ cho chức năng thanh toán tiền trọ qua mail



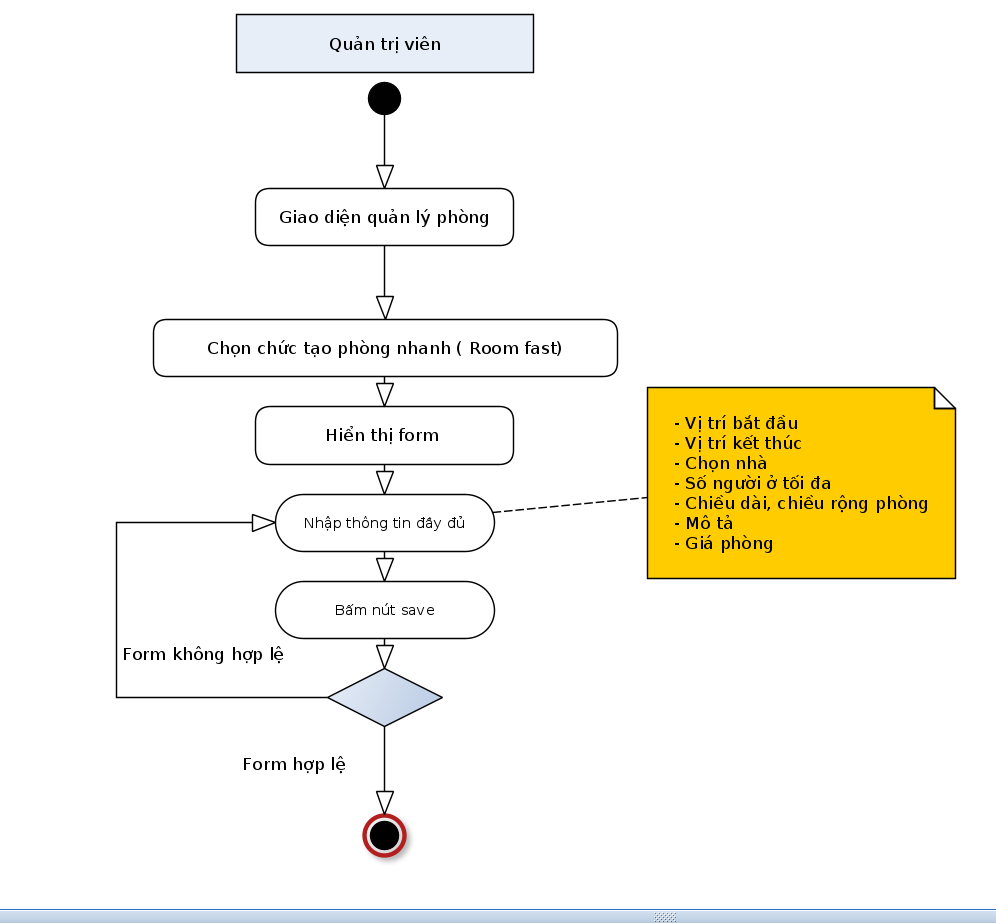
* + 1. Biểu đồ cho chức năng trả phòng



* + 1. Biểu đồ cho chức năng thêm mới một nhà



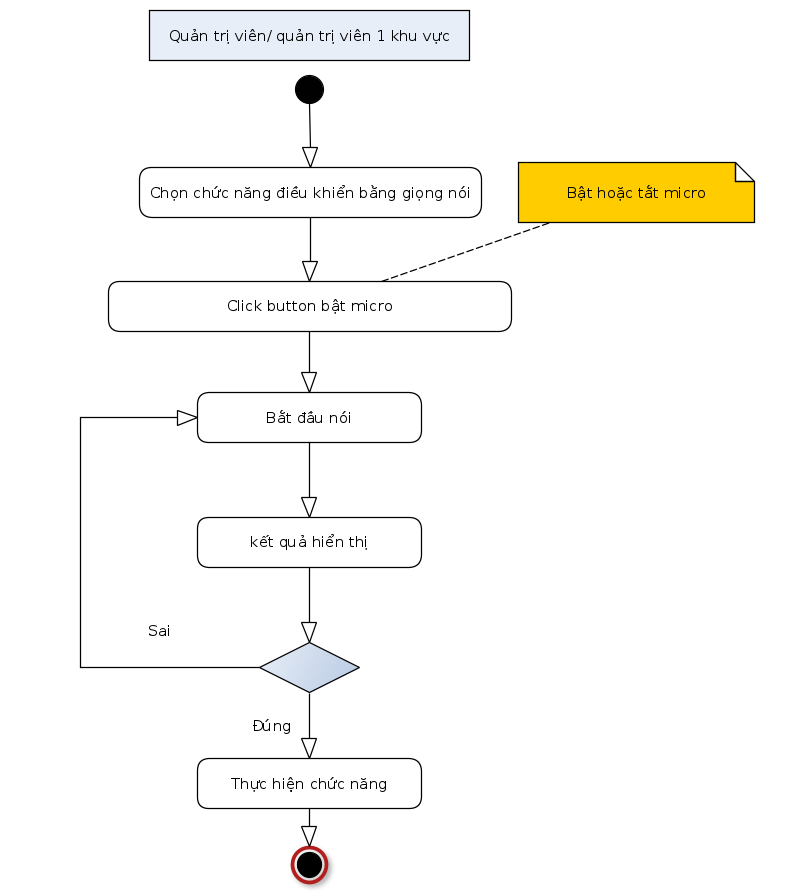
* + 1. Biểu đồ cho chức năng thêm mới phòng nhanh



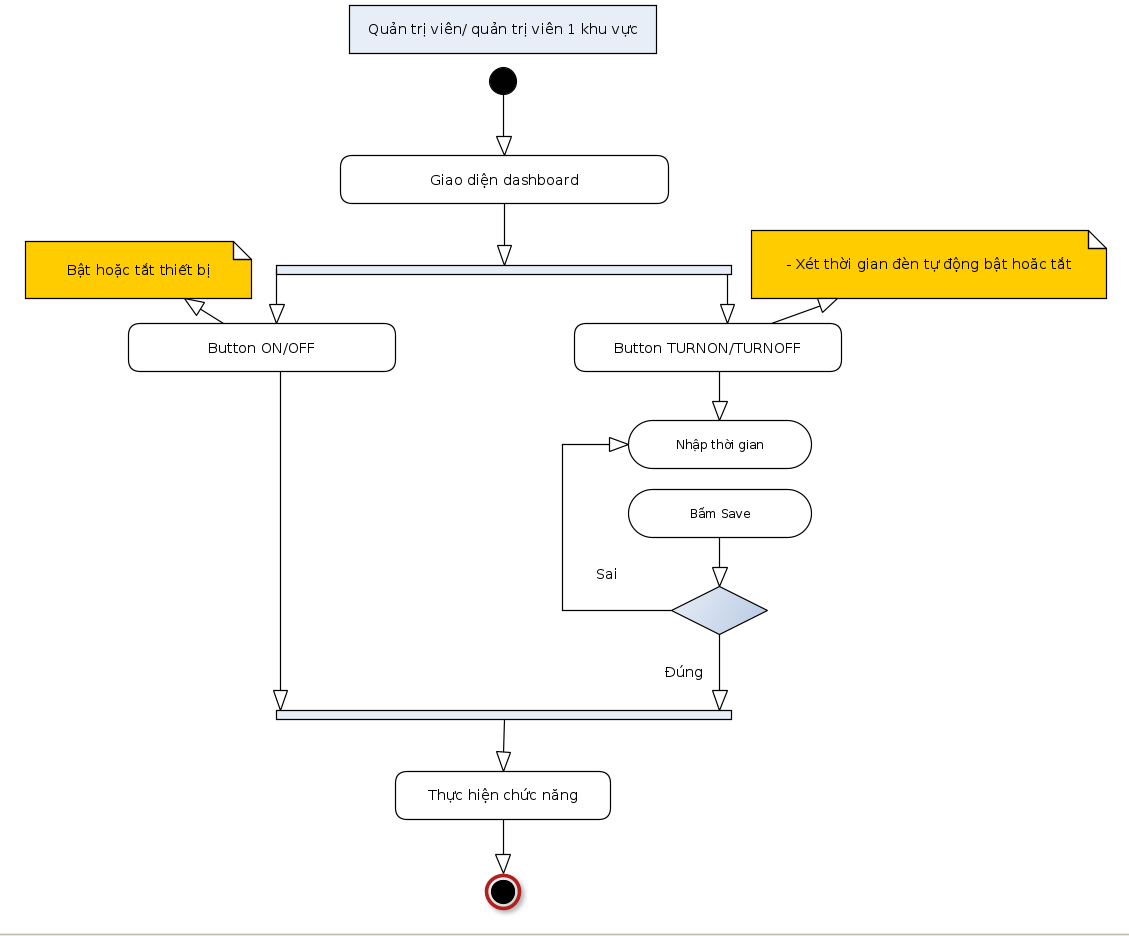
* + 1. Biểu đồ cho chức năng thêm mới một phòng



* + 1. Biểu đồ cho chức năng điều khiển bằng giọng nói



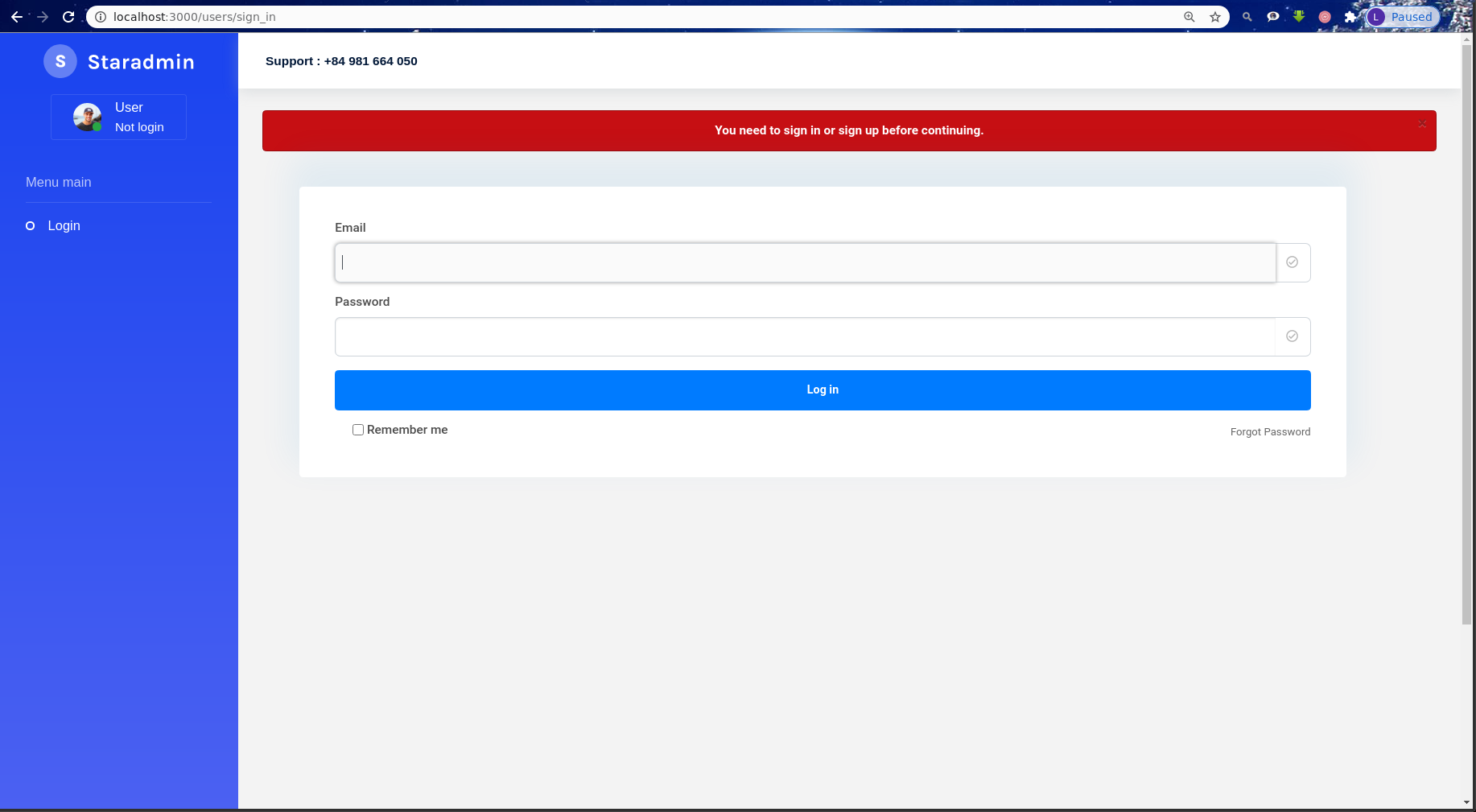
* + 1. Biểu đồ chức năng điều khiển thiết bị trong phòng



* 1. Xây dựng biểu đồ thực thể
  2. Xây dựng biểu đồ tương tác tuần tự

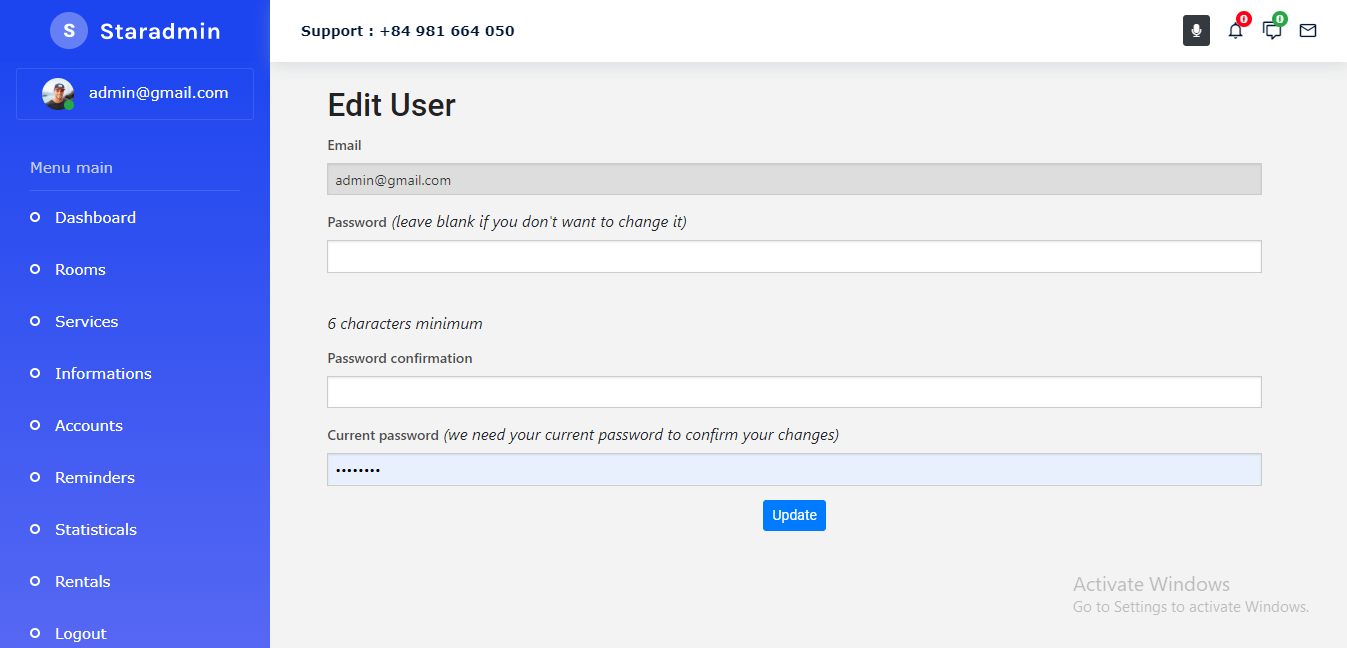
Thiết kế hệ thống

* 1. Thiết kế cơ sở dữ liệu
  2. 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.

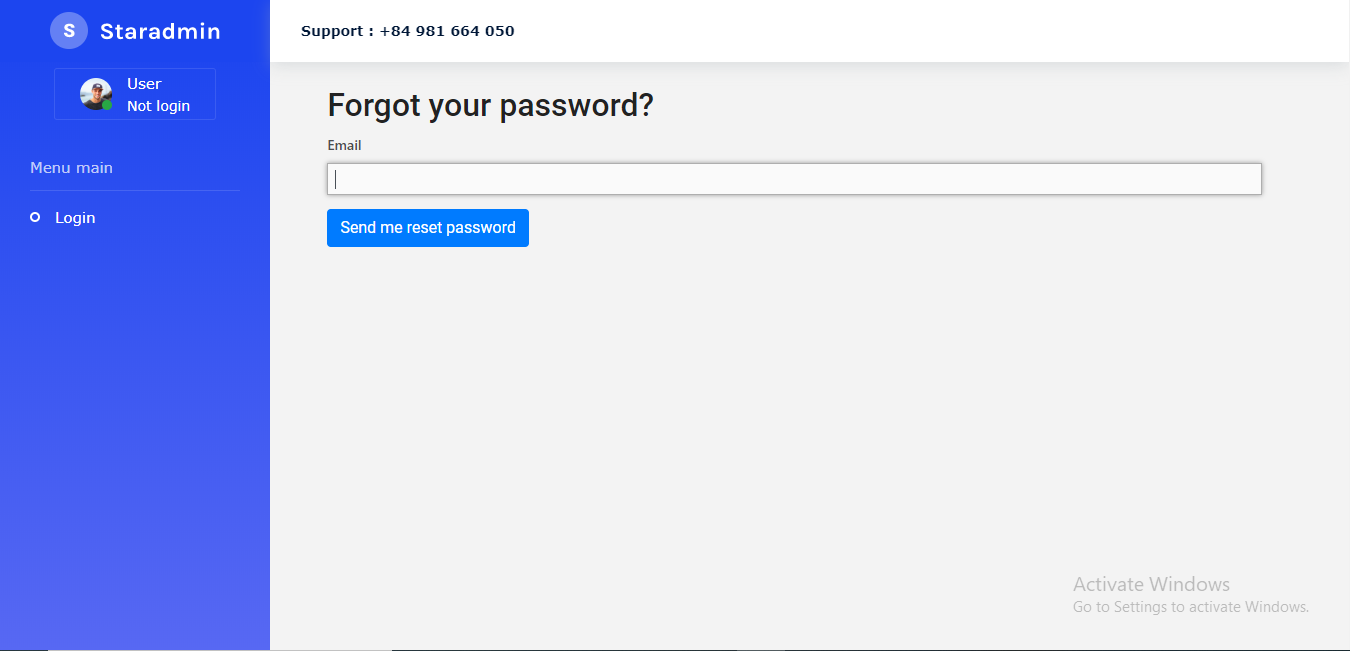


hình 1: Trang đăng nhập

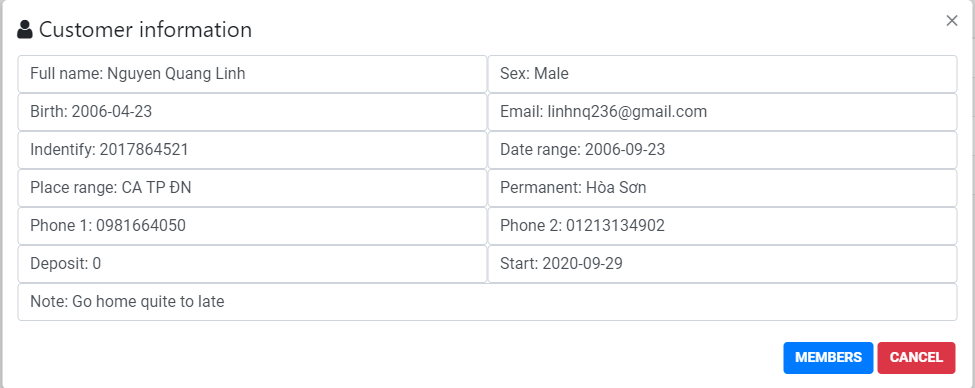
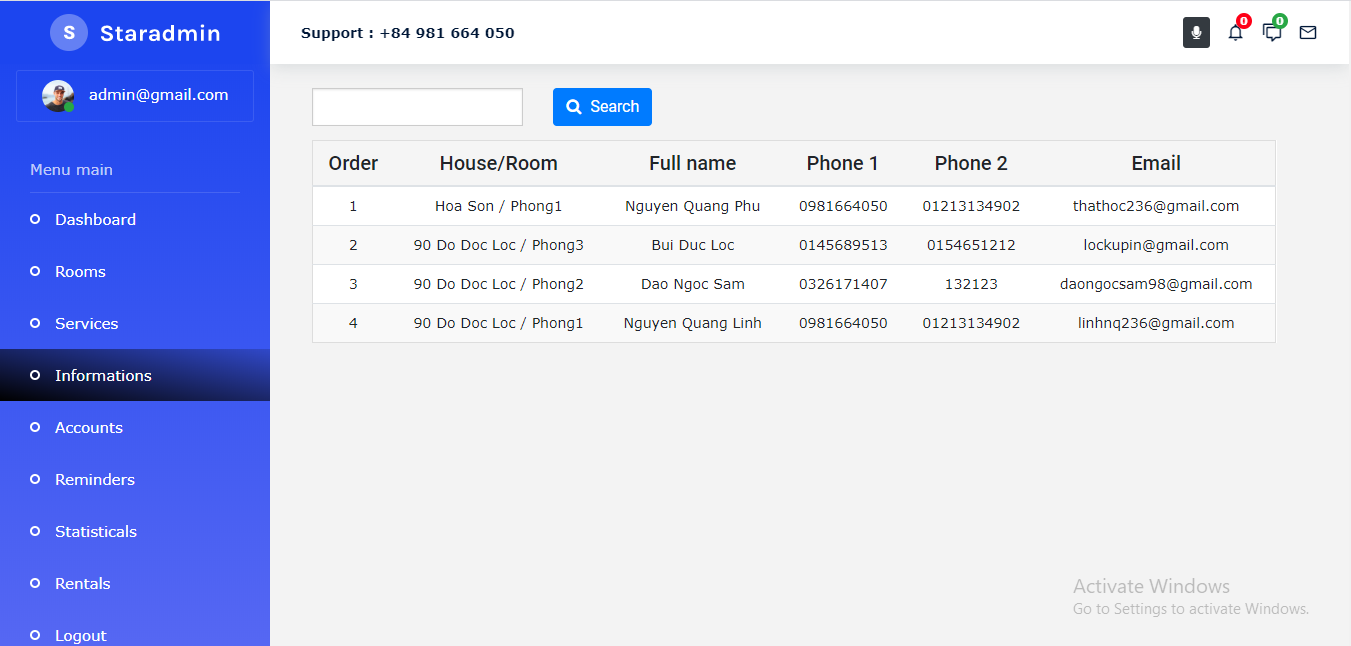
* Trang Change Password



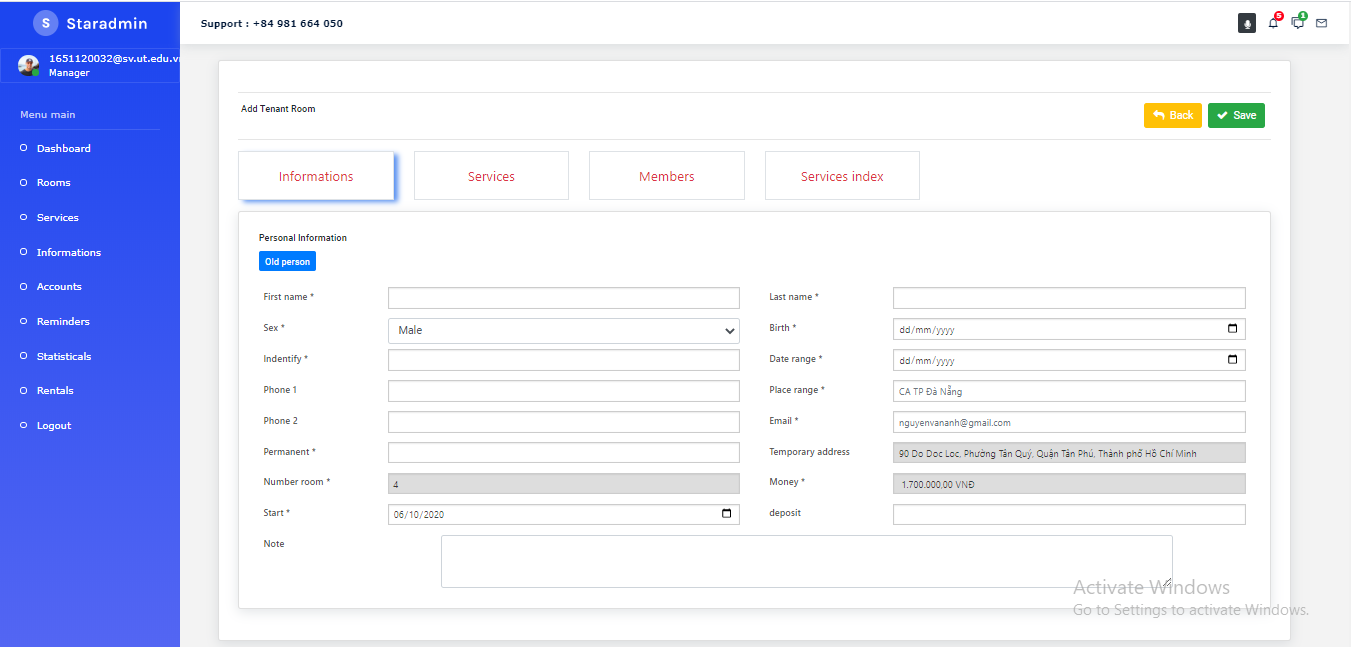
* Trang Fotget Password



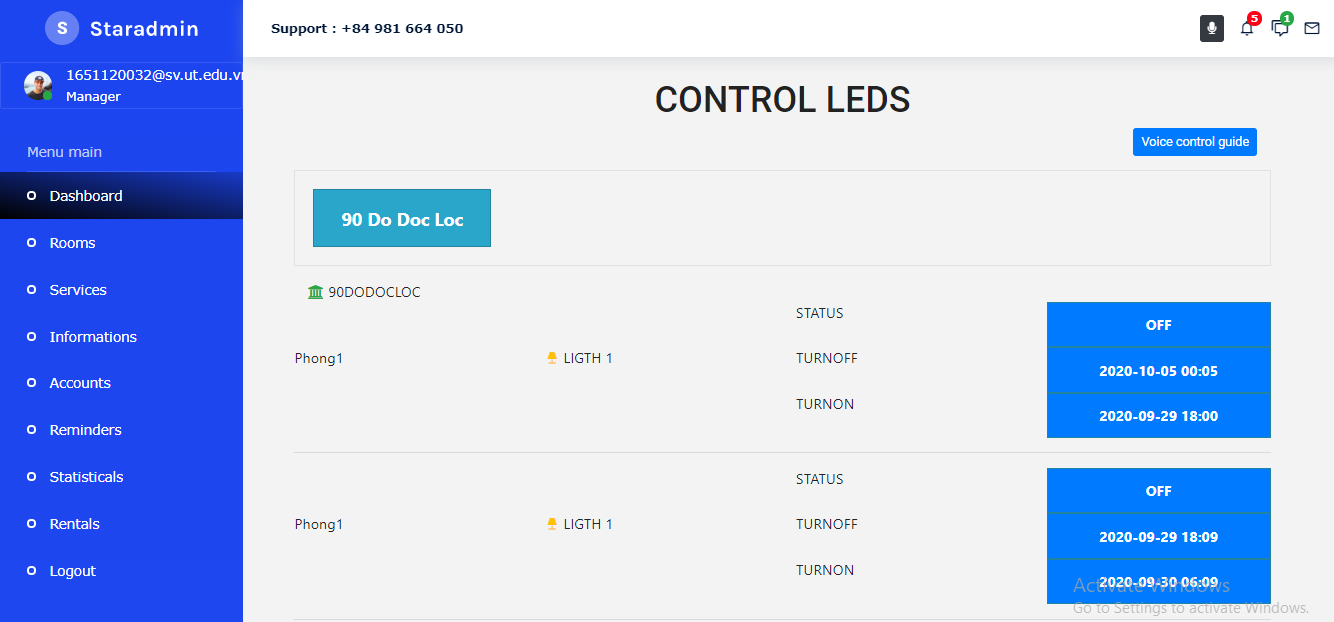
* Trang Change avata
* Trang Show - thông tin khách đã đặt phòng



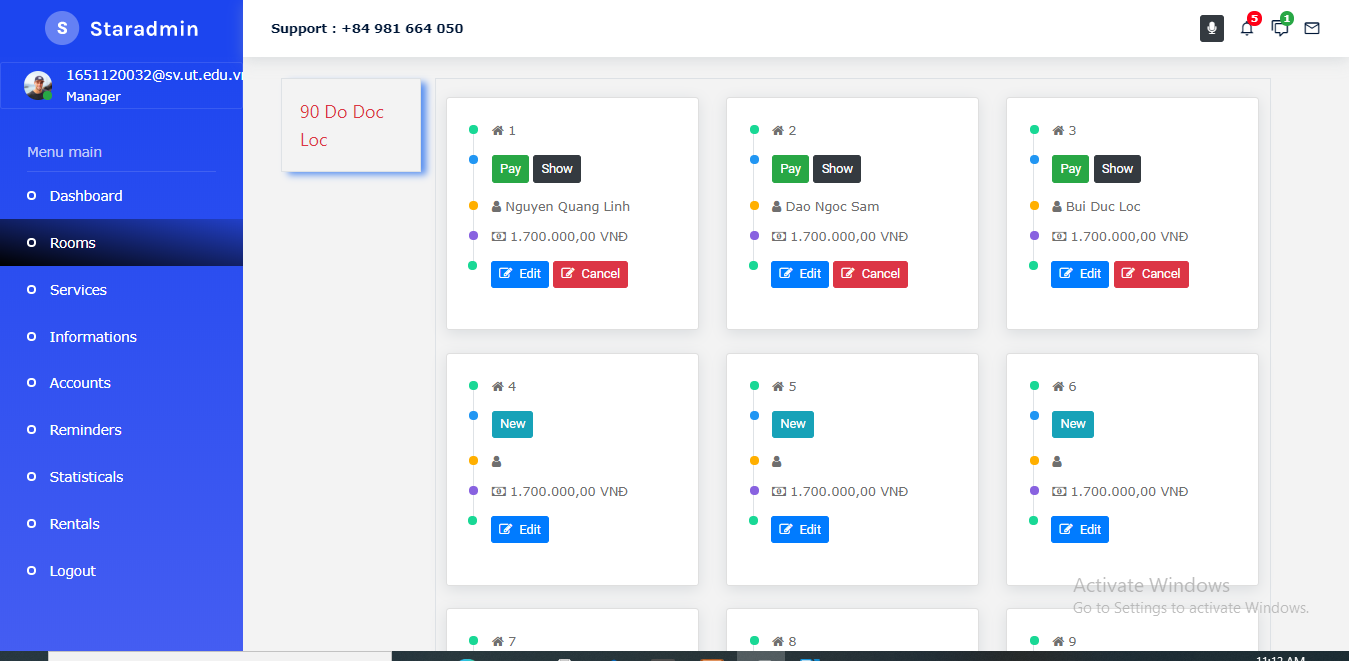
* Trang New - thêm một khách thuê mới



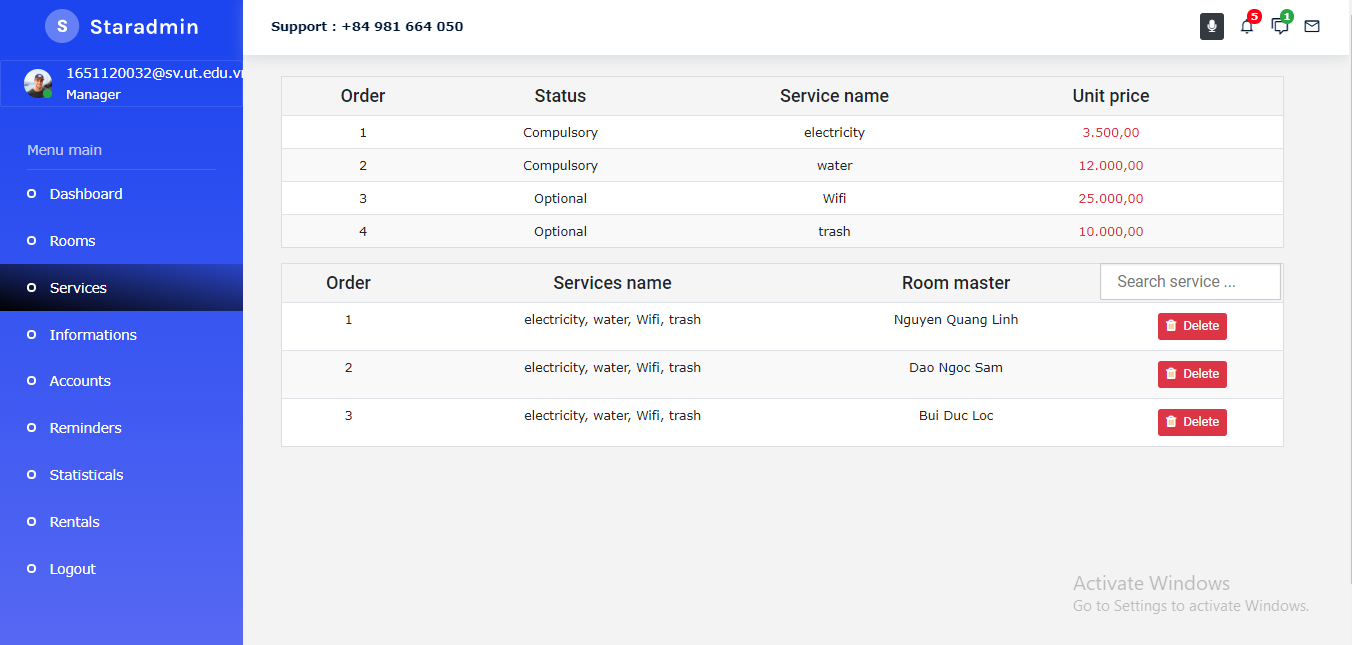
1. Đối với quản lý 1 khu vực
   * Trang Dashboard



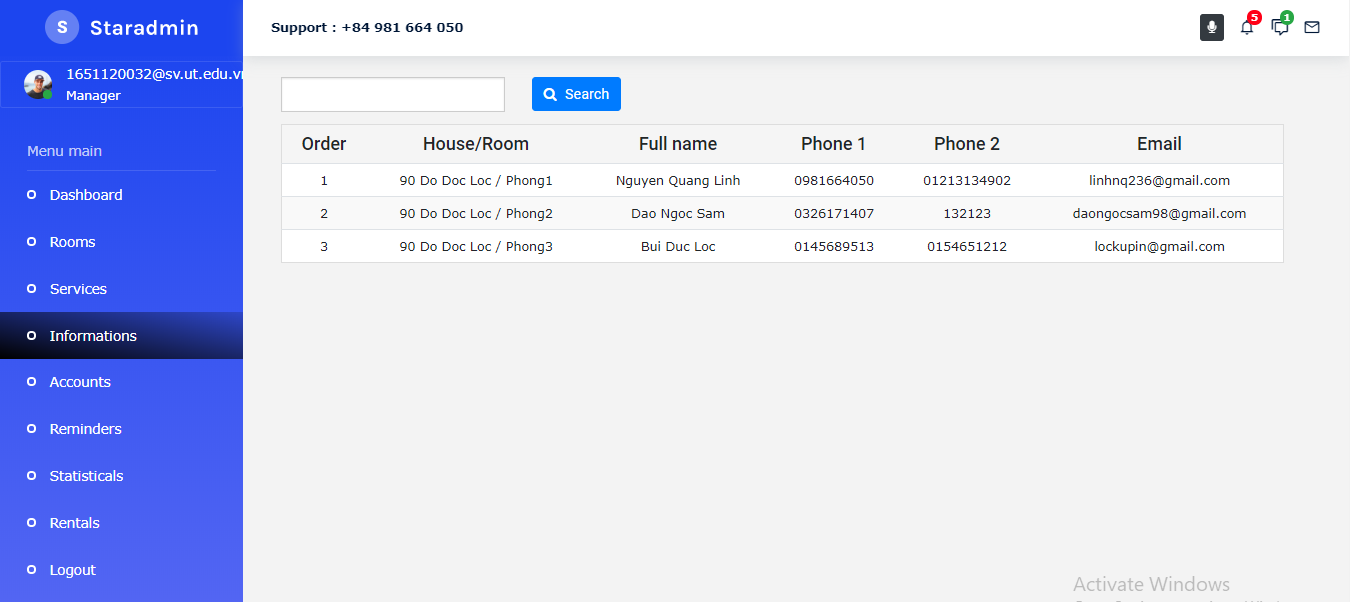
* + Trang Rooms



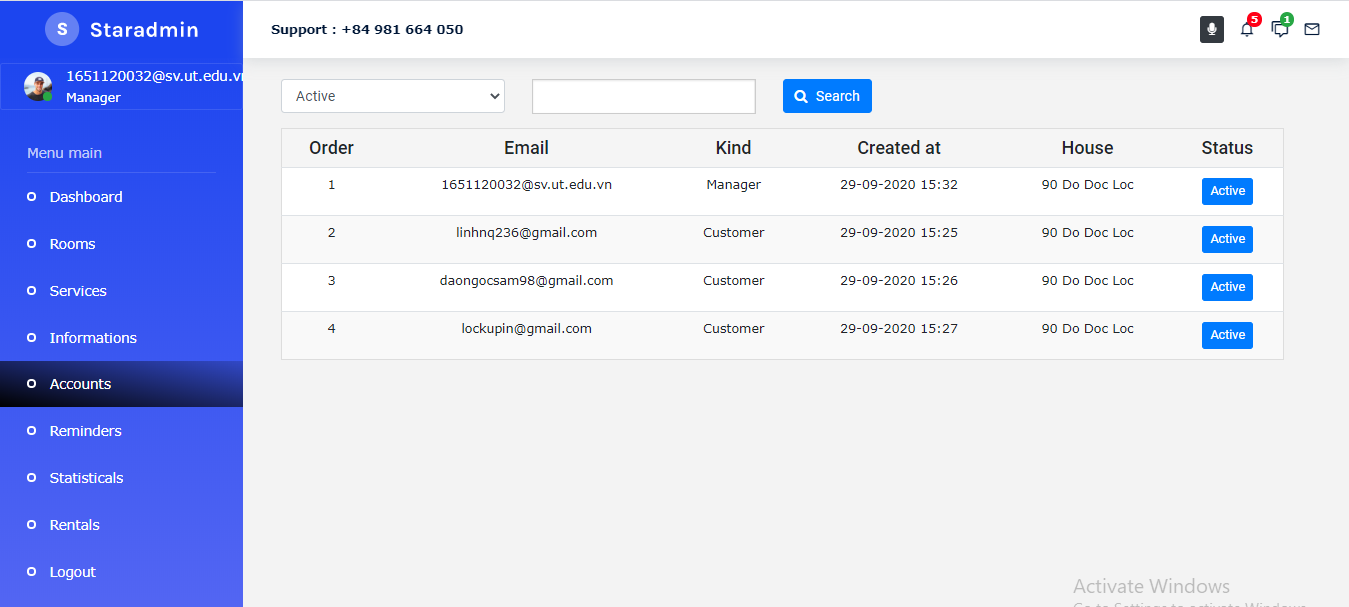
* + Trang Services



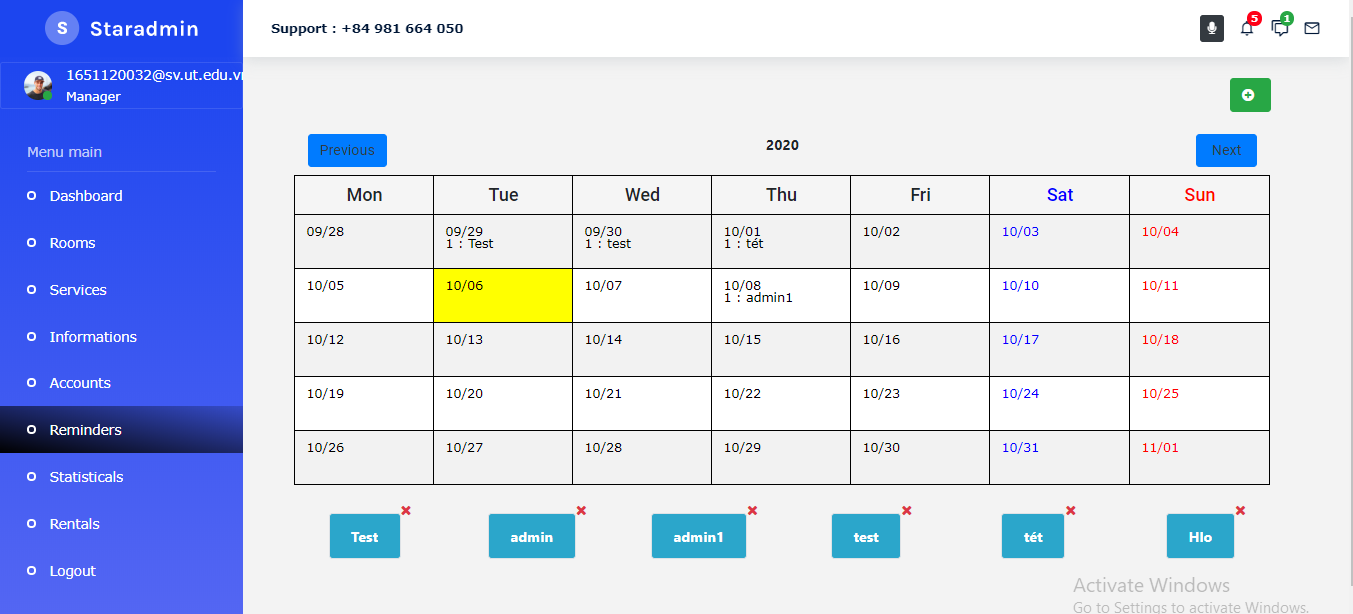
* + Trang Informations



* + Trang Accounts



* + Trang Reminders



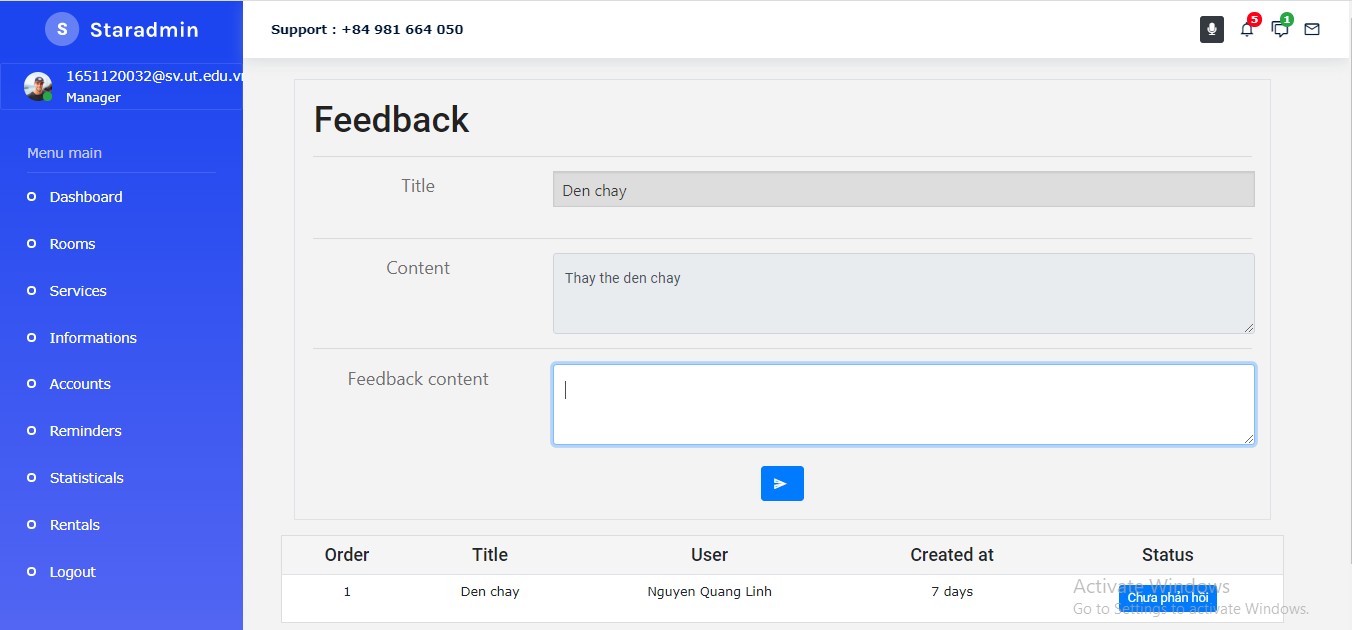
* + Trang Statisticals



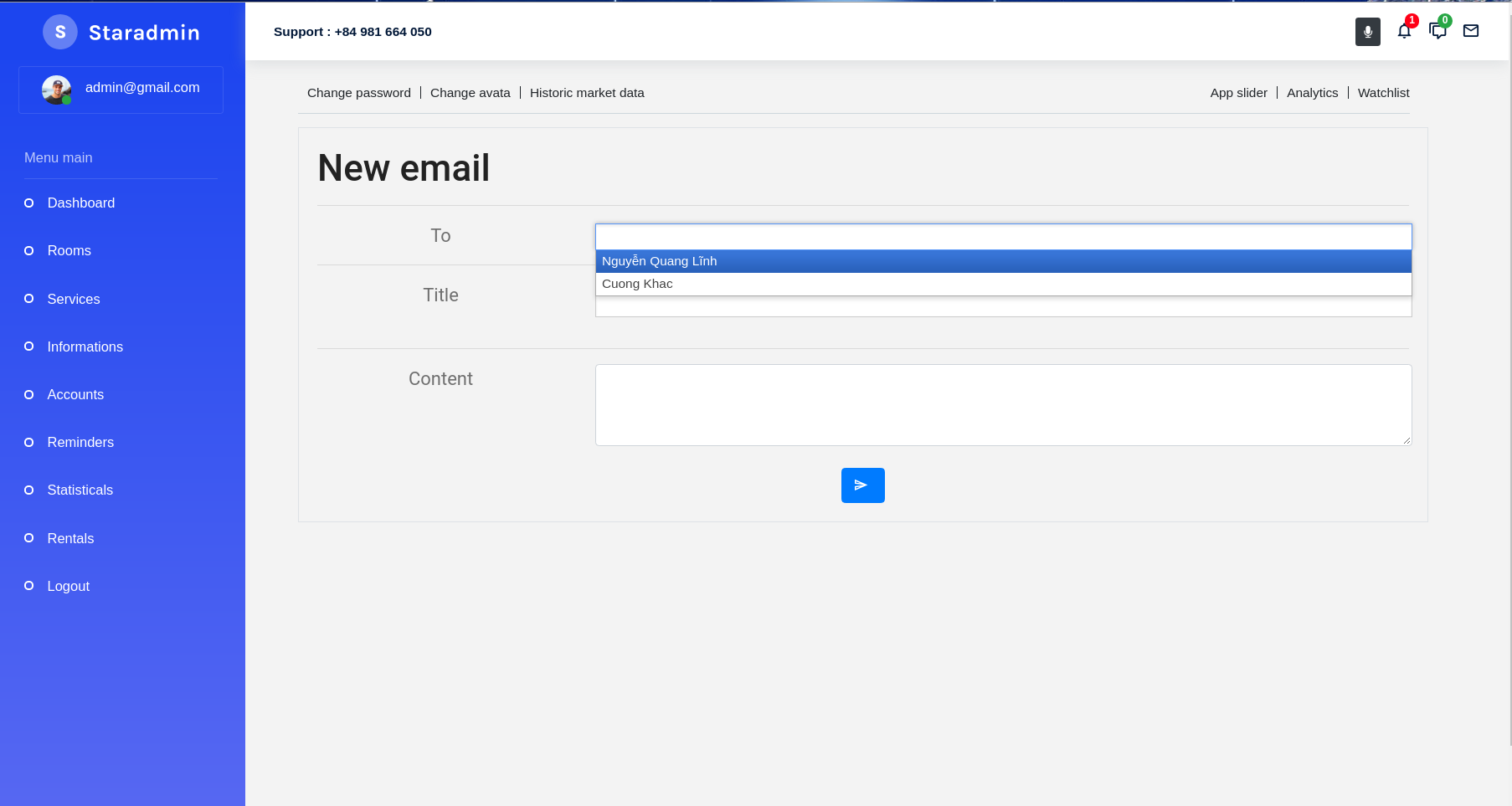
* + Trang Rentals



* + Trang Reports



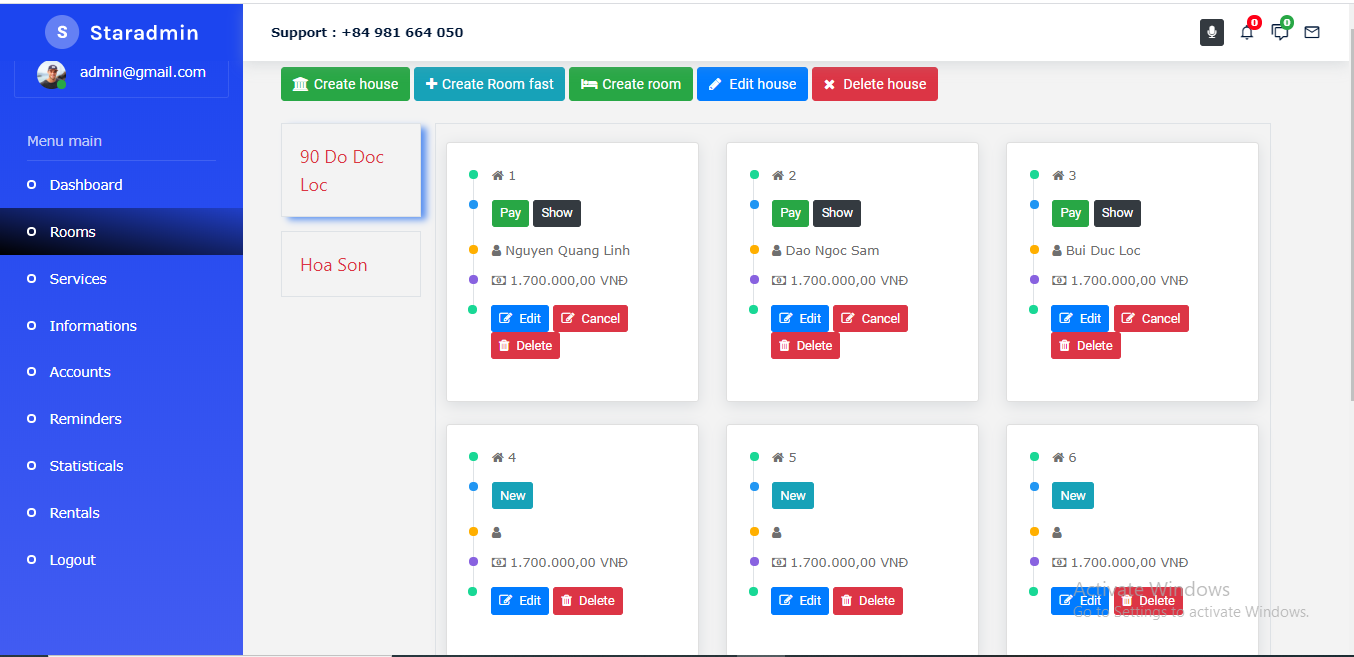
* + Trang Email



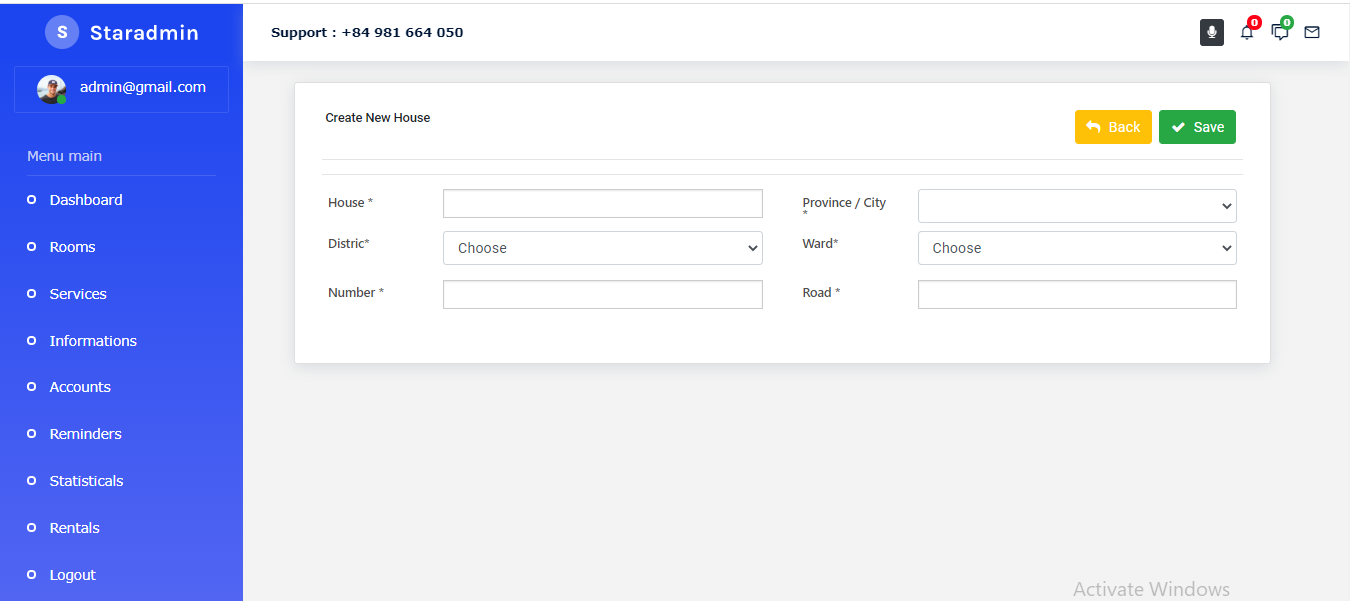
1. Đối với quản lý tất cả khu vực
   * Trang Dashboard



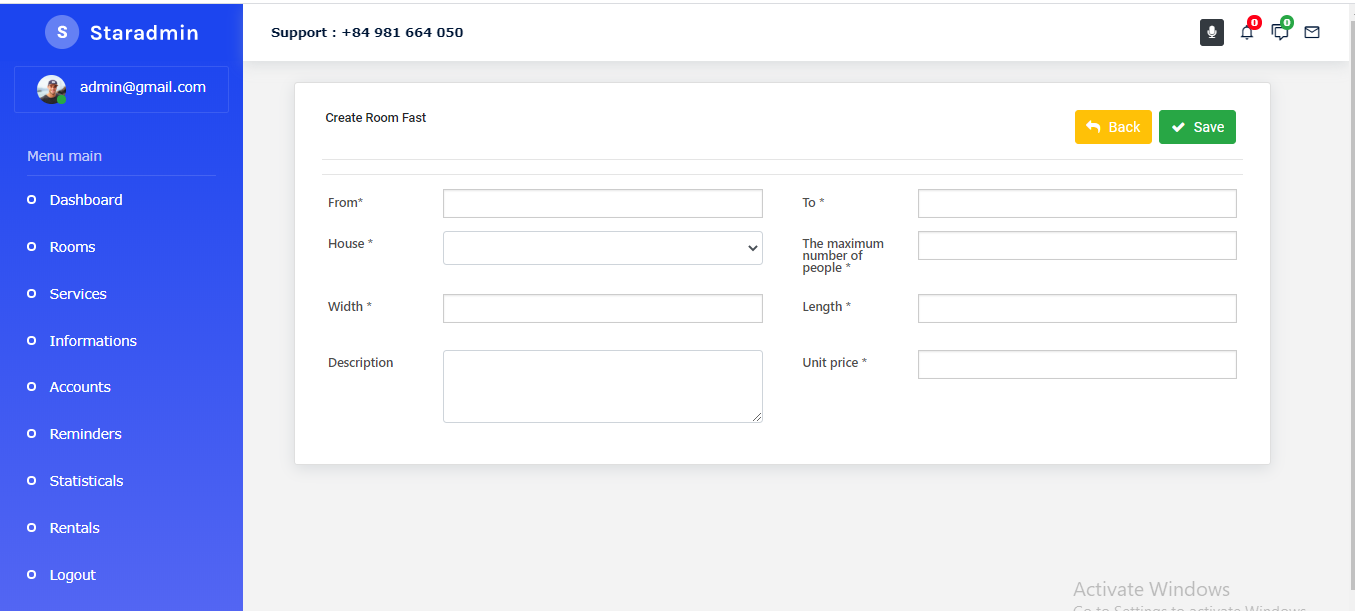
* + Trang Rooms



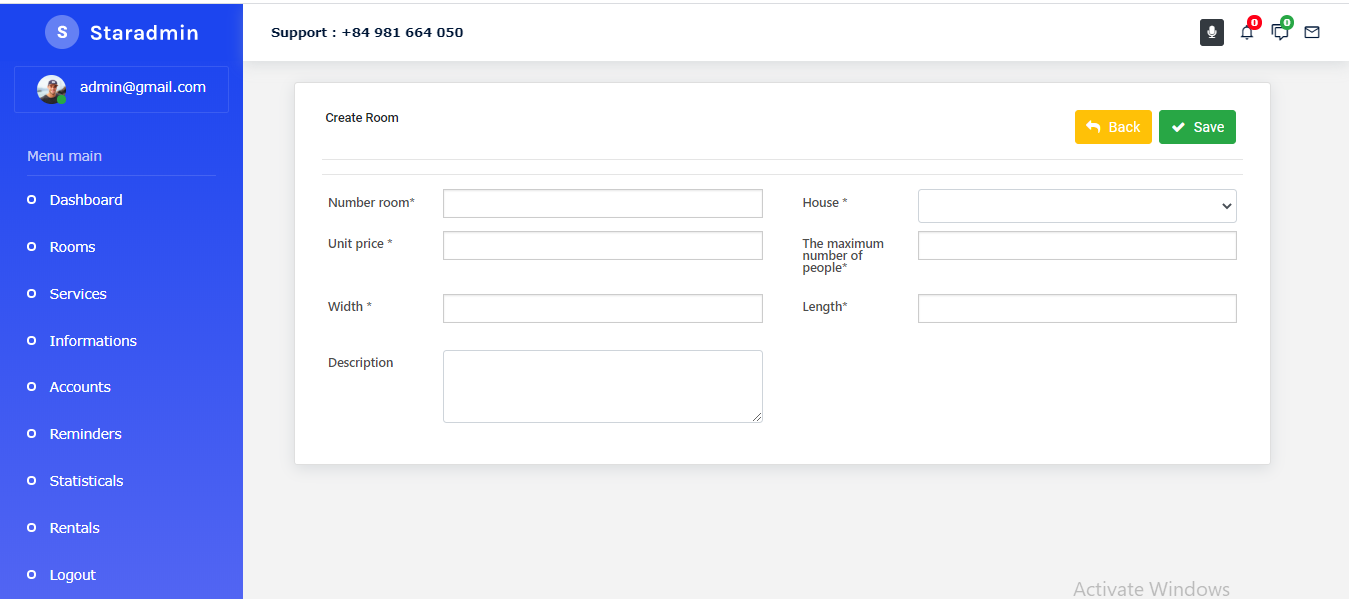
* + Trang Create House



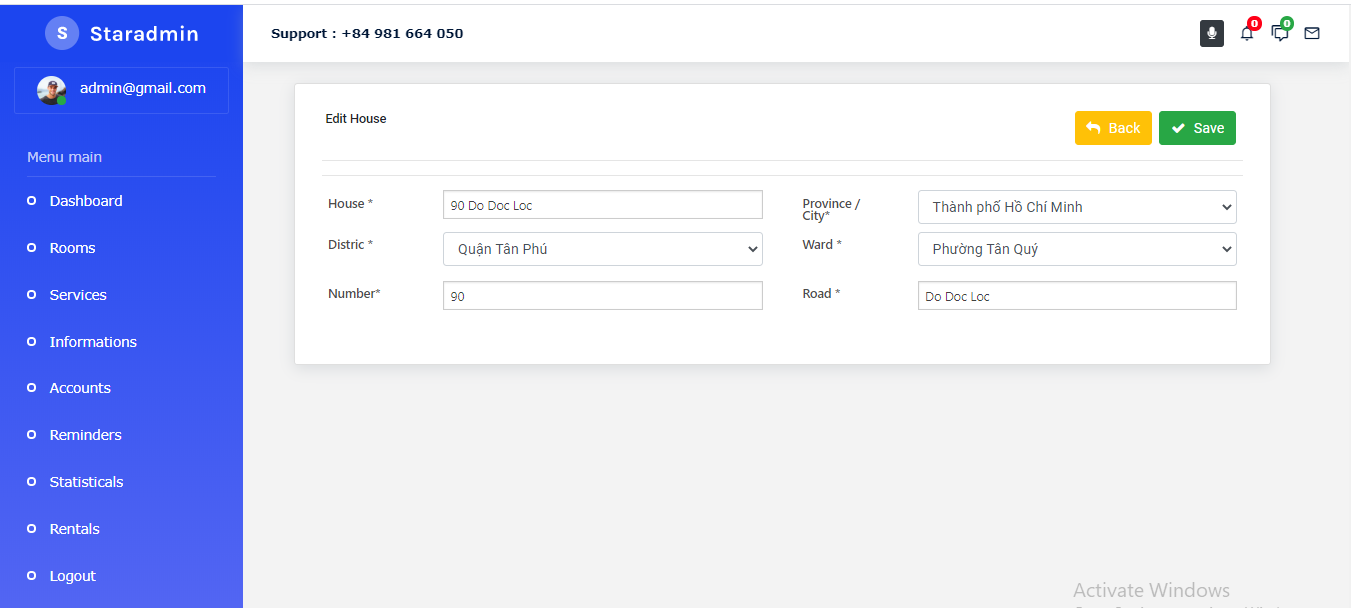
* + Trang Create Room Fast



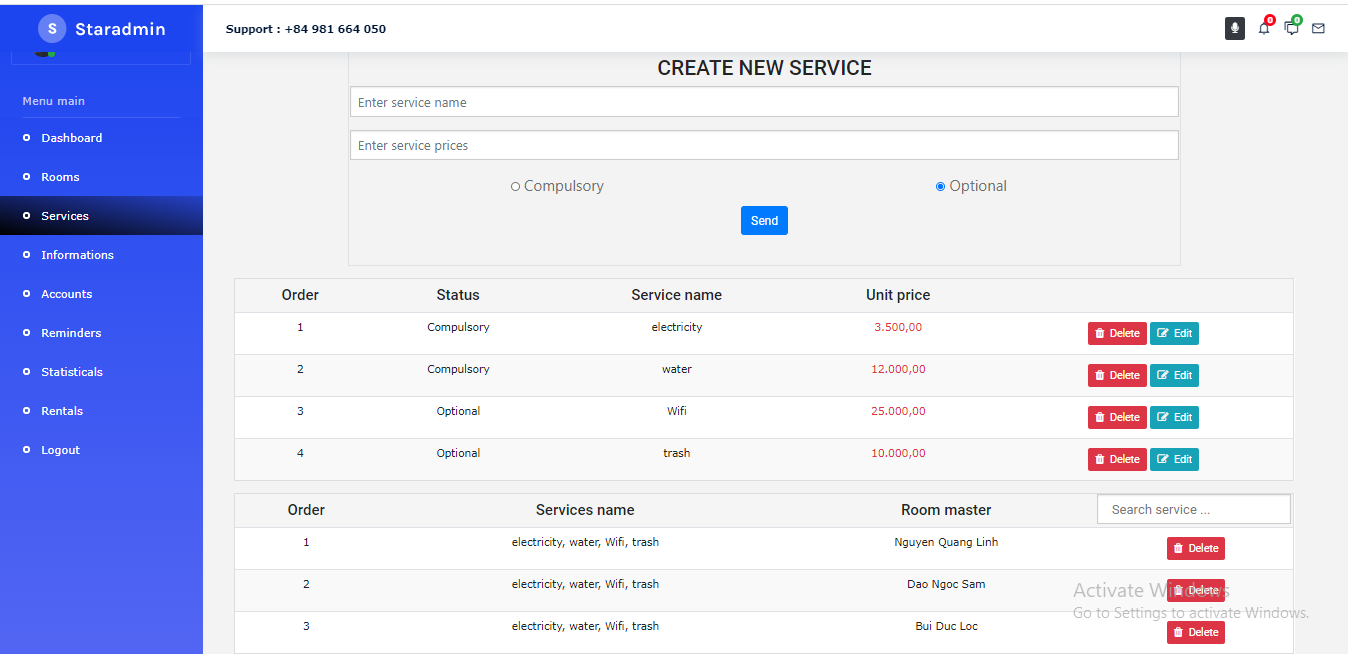
* + Trang Create Room



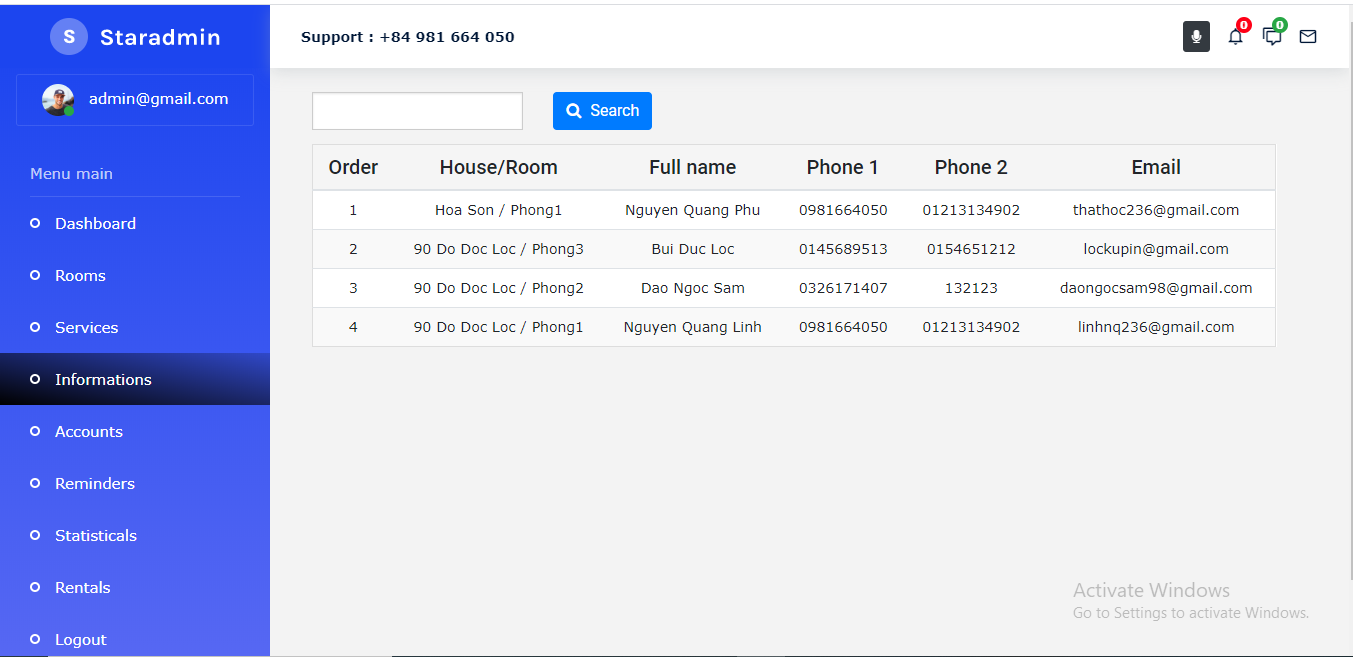
* + Trang Edit House



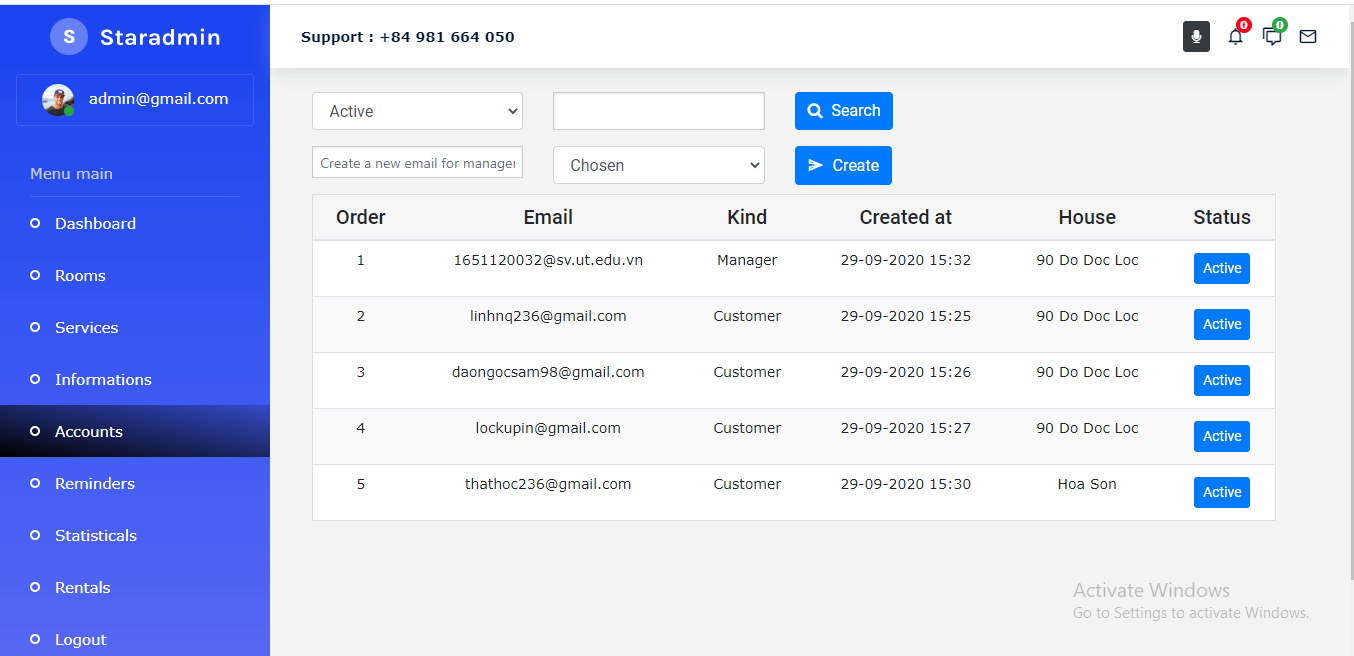
* + Trang Services



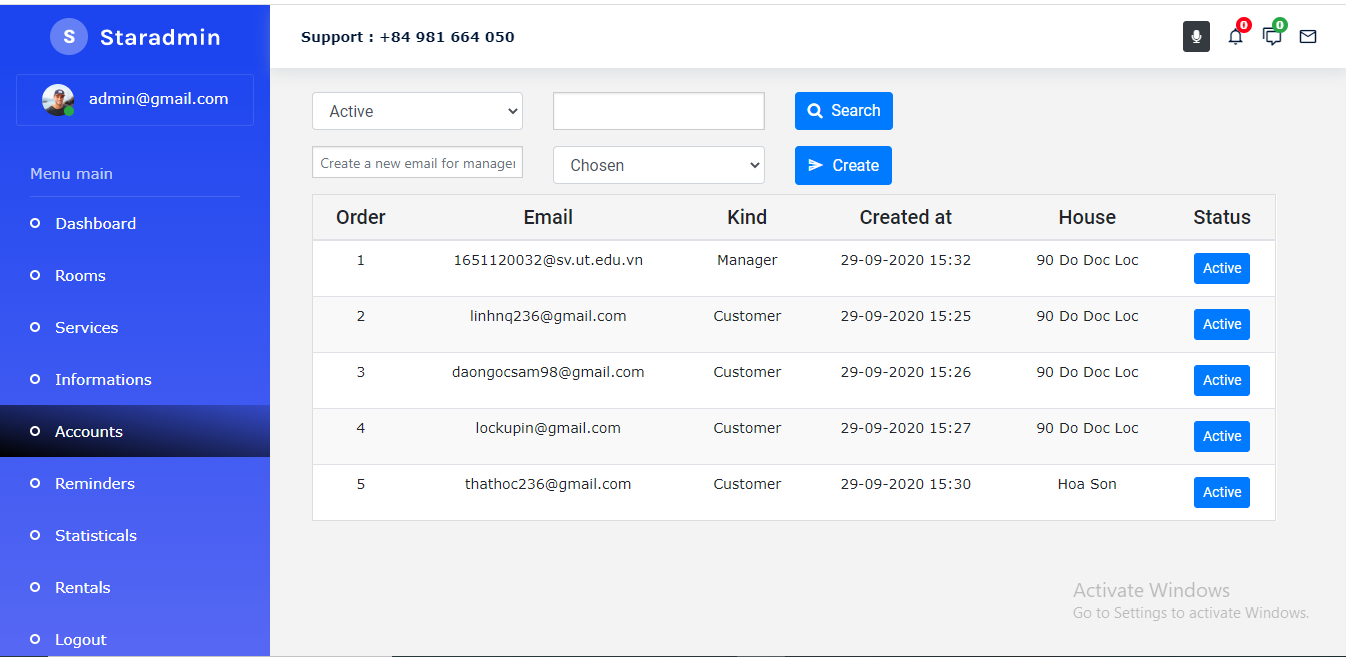
* + Trang Informations



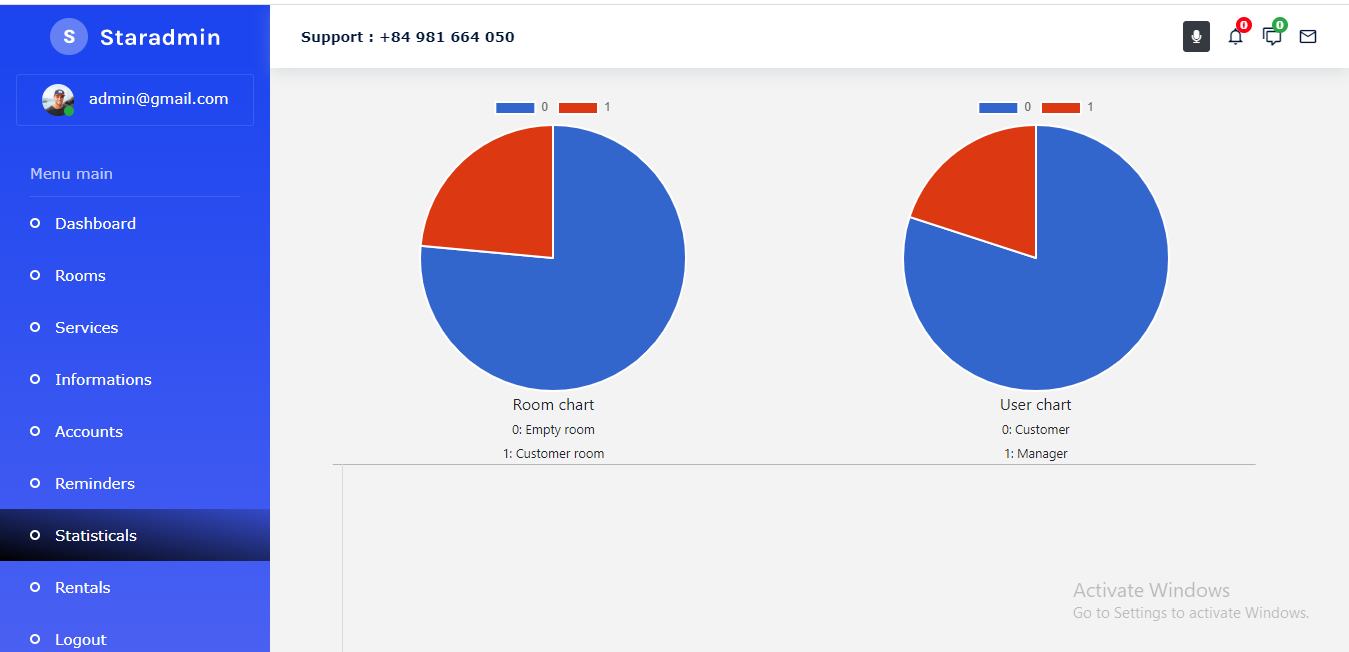
* + Trang Accounts



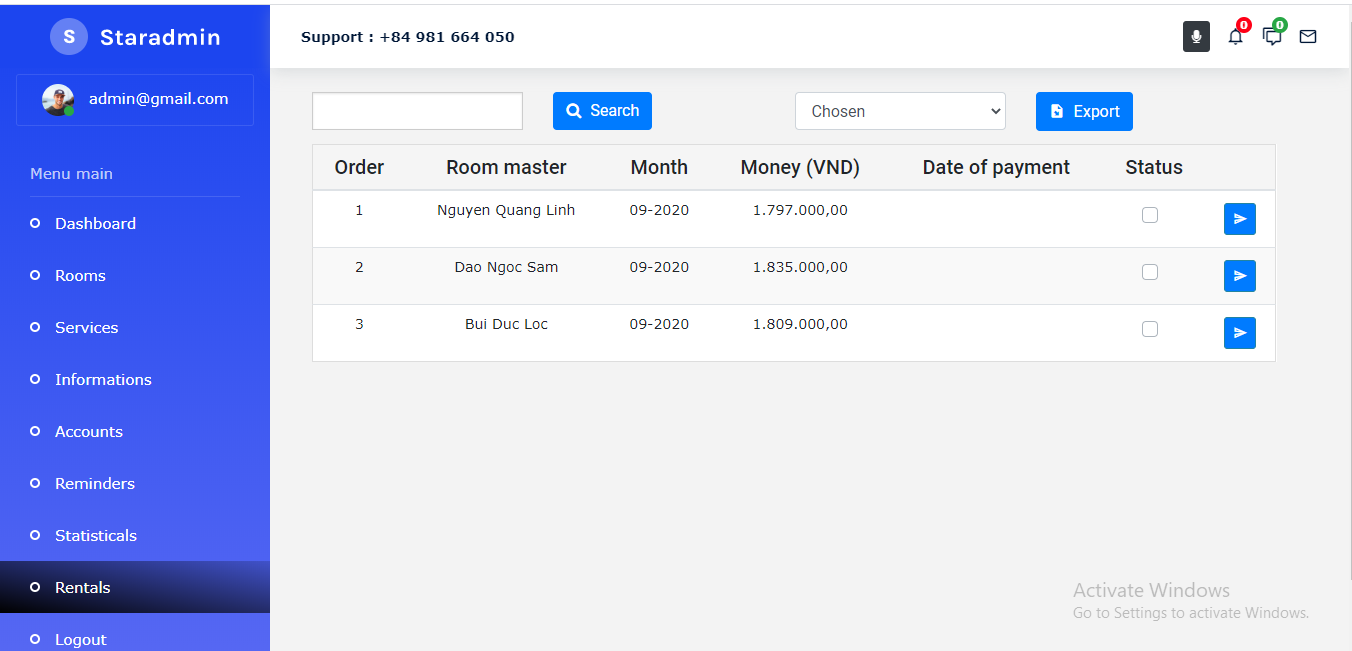
* + Trang Reminders



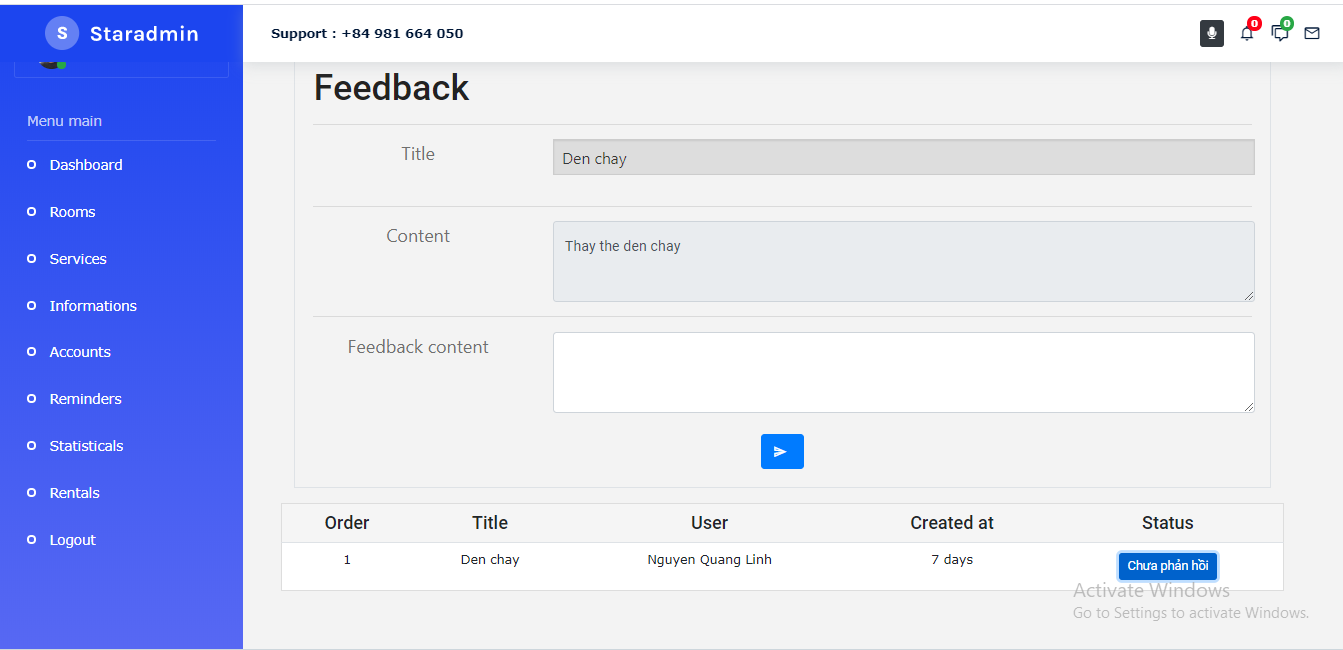
* + Trang Statisticals



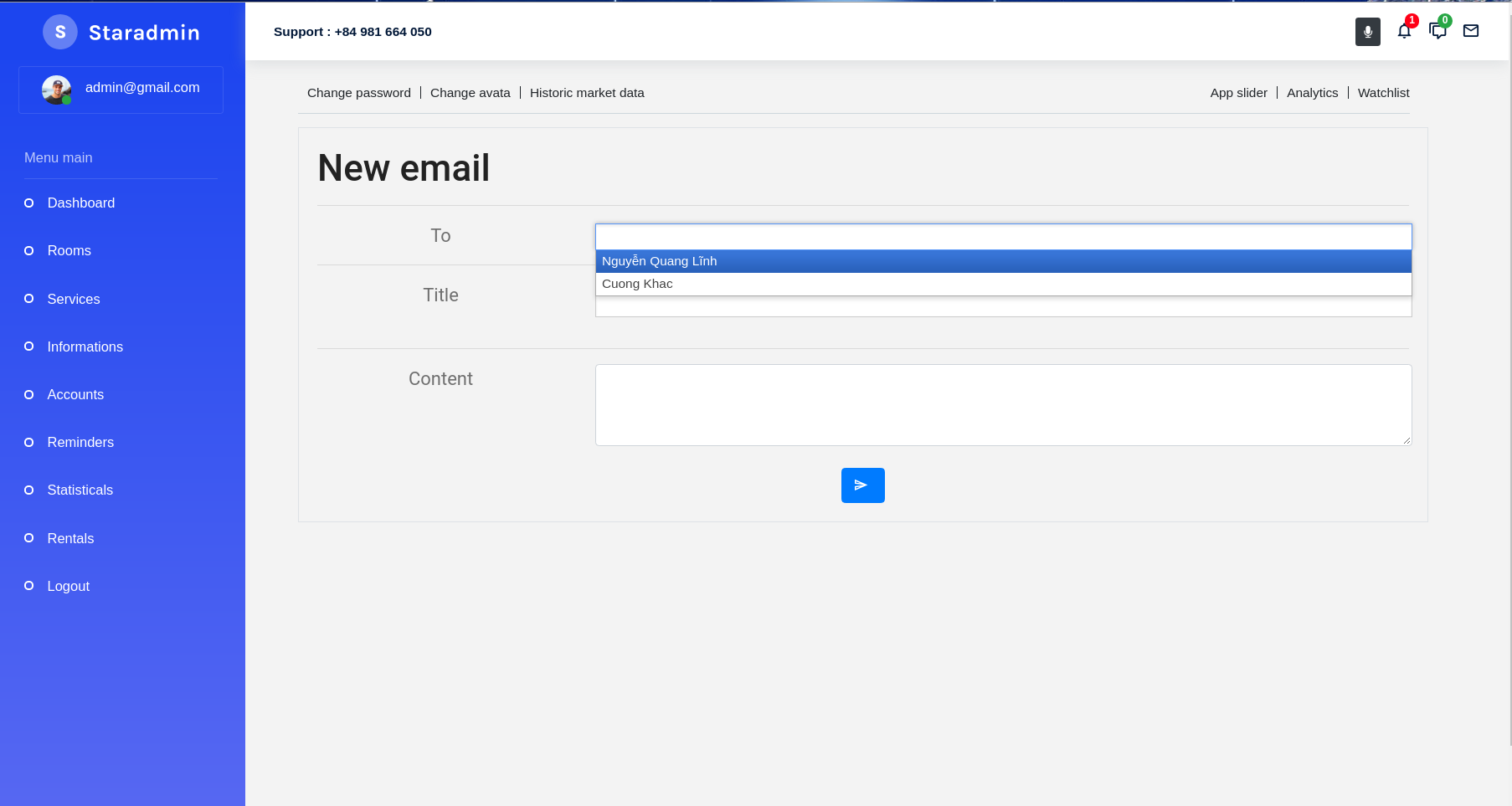
* + Trang Rentals



* + Trang Reports



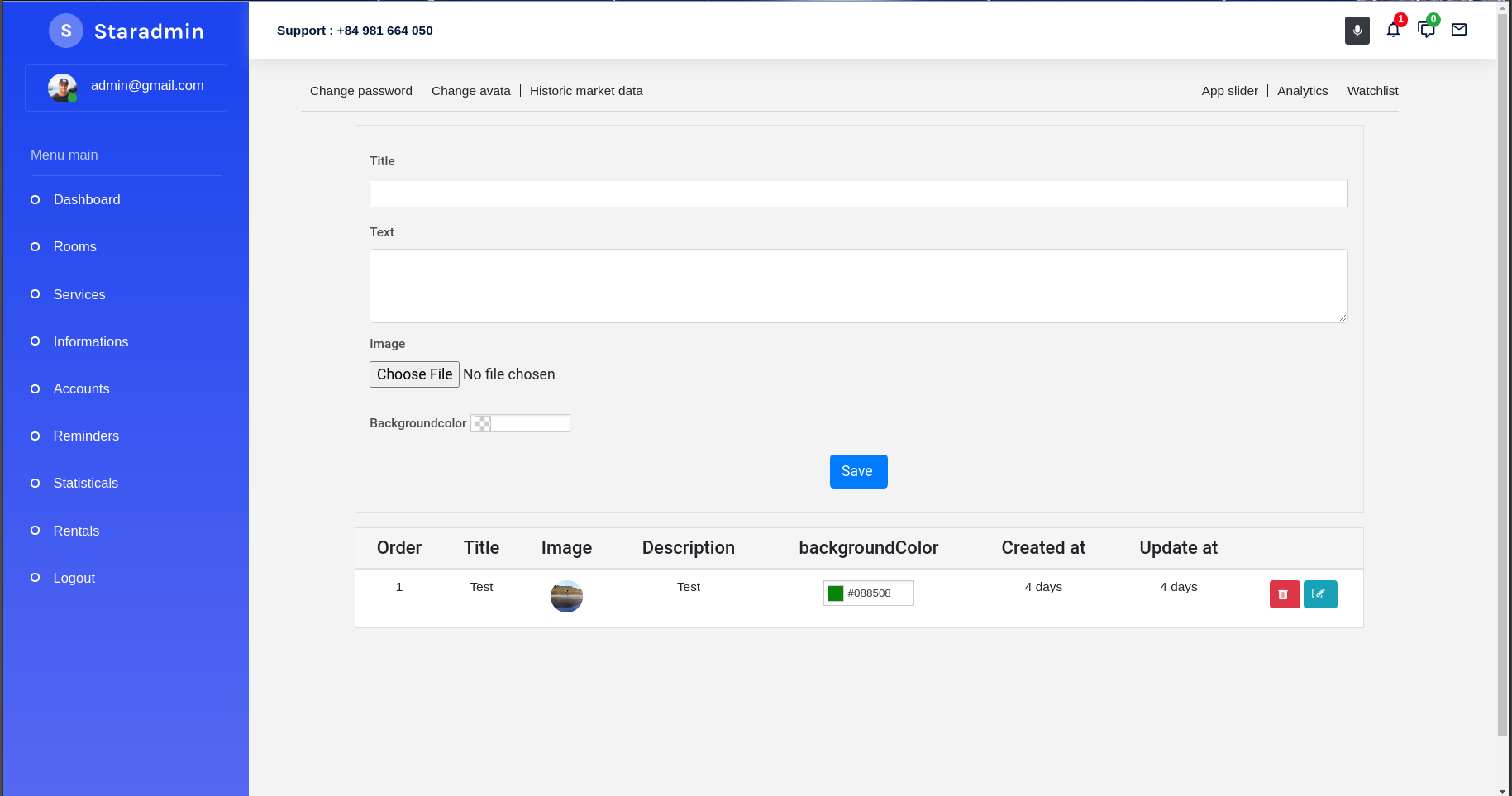
* + Trang Email



* + Trang Change Password



* + Trang App Slider



* 1. API
     + 1. Các API được sử dụng trong trang web
          - appSlider

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"

}

]

}

* + - * + Chức năng đăng nhập

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

type: post

data: {

“email”: “[linhnq236@gmail.com](mailto:linhnq236@gmail.com)”,

“password”:123456

}

response:

{

"status": 200,

"username": "",

"id": "",

"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 Lĩnh",

"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

}

}

* + - * + Cập nhật thông tin cá nhân

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

}

* + - * + Hiển thị bảng điều khiển

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"

}

}

}

* Chế độ điều khiển thiết bị: Bật, tắt và hẹn giờ

url: http://localhost:3000/api/led\_status/4

type: post

data:

{

“name”: “LED\_STATUS0”

“column”: “STATUS”,

“status”: “

}

response:

{

"status": 200

}

* Lấy hông tin dịch vụ đã đăng ký

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

type: get

data: none

response:

{

"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": "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"

]

}

* Lấy dữ liệu thanh toán hàng tháng

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

}

]

}

* Gửi phản hồi

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

type: post

data: {

“title”: “Test”,

“content”: “Test content”,

“information\_id”: 4

}

response:

{

“status”:200

}

* Lấy thông tin đã phản hồi

url: <https://localhost:3000/api/showpopup/4>

type: get

data: none

response:

{

"data": [

{

"id": 1,

"title": "Test",

"content": "Test contnet",

"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

}

]

}

* Xóa một phản hồi

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

type: delete

data: none

response:

{

"data": {

"id": 1,

"title": "Test",

"content": "Test contnet",

"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

}

}

* Lấy thông tin nội quy

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; 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"

}

* + 1. ỨNG DỤNG CHẠY TRÊN APP

1. Phân tích hệ thống
   * 1. Xác định yêu cầu
        + Quản lý tại phòng đang ở.
     2. Đặc tả yêu cầu
        1. Yêu cầu chức năng
           + Sau khi khách hàng đã đặt phòng thành công và dọn đến ở thì người quản lý 1 khu vực sẽ cấp một tài khoản đăng nhập trên app với email người đăng ký và mật khẩu được cấp.
        2. 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.
           + S-ử-dụng Nodejs 8.17.0, React-native-cli 2.0.1, React-native 0.62.2
     3. Xây dựng biểu đồ chức năng
        1. Tác nhân
           + Chỉ có một tác nhân tham gia và tương tác với app: Khách hàng thuê
        2. 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
        3. Biểu đồ hoạt động
           + Biều đồ cho chức năng đăng nhập
2. Thiết kế hệ thống
   * 1. Mô tả chi tiết từng màn hình
        1. Màn hinh trang chủ chưa đăng nhập - Home
        2. Màn trang đăng nhập - Login
        3. Màn hình trang chủ khi đã đăng nhập - Home
        4. Màn hình trang thông tin cá nhân – Setting
        5. Màn hình quản lý thiết bị – Leds
        6. Màn hình dịch vụ – Services
        7. Màn hình phản hồi - Feedback
     2. INTERNET OF THINGS ( IOT)
        1. Cách kết nối chip ESP8266 với wifi
        2. H

PHẦN 3: ĐÁNH GIÁ VỀ HƯỚNG PHÁT TRIỂN

1. Đánh giá và kết luận
2. Hướng phát triển

TÀI LIỆU KHAM KHẢO