**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**…………….🙚O🙘…………….**

Cần Thơ, ngày tháng 04 năm 2015

Giảng viên hướng dẫn

ThS. Lâm Chí Nguyện

**NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

**…………….🙚O🙘…………….**

Cần Thơ, ngày tháng 04 năm 2015

Giảng viên phản biện

**NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

**…………….🙚O🙘…………….**

Cần Thơ, ngày tháng năm 2015

Giảng viên phản biện

# LỜI CẢM ƠN

**…………….🙚O🙘…………….**

Trong suốt quá trình thực hiện đề tài luận văn tốt nghiệp “Xây dựng hệ thống quản lý, điều khiển các dịch vụ mạng bằng giao diện web” em luôn nhận được sự giúp đỡ từ giảng viên, bạn bè và các anh chị để em có thể thực hiện tốt đề tài của mình.

Trước tiên, em xin gửi lời cảm ơn chân thành đến thầy Lâm Chí Nguyện đã tận tình hướng dẫn và chỉ ra con đường tốt để em có thể được làm luận văn đồng thời học hỏi và rèn luyện thêm nhiều kinh nghiệm.

Trên hết em xin gửi lời cảm ơn sâu sắc tới quý thầy cô trường Đại học Cần Thơ nói chung và quý thầy cô khoa Công nghệ thông tin và Truyền thông nói riêng, quý thầy cô đã truyền cho em những kiến thức, kinh nghiệm quý báu trong suốt 4 năm qua để em có đủ hành trang và nghị lực bước vào đời và có thể trở thành người có ích cho xã hội.

Kế đến là xin cảm ơn Cộng đồng ZendVN, Study And Share (Các anh chị đã đi làm) đã tận tình hướng dẫn cho em hiểu biết về công nghệ mới, cách áp dụng vào đề tài.

Mặc dù đã cố gắng hoàn thành luận văn trong phạm vi và khả năng nhưng chắc chắn sẽ không tránh khỏi những thiếu sót và cũng như chưa thể hoàn thiện 100% yêu cầu đề tài đặt ra trong khoảng thời gian ngắn. Em kính mong nhận được sự thông cảm và tận tình chỉ bảo của quý thầy cô và các bạn, hi vọng đề tài của em sẽ được phát triển và hoàn thiện để đưa vào sử dụng.

Cần Thơ, ngày tháng năm 2015

Sinh viên thực hiện

# LỜI NÓI ĐẦU

**…………….🙚O🙘…………….**

# MỤC LỤC

**…………….🙚O🙘…………….**

# KÍ HIỆU VÀ VIẾT TẮT

**…………….🙚O🙘…………….**

|  |  |
| --- | --- |
| **Từ/kí hiệu viết tắt** | **Nguyên bản** |
| API | Application Programming Interface |
| JSON | JavaScript Object Noattion |
| SSH | Secure Shell |
| CLI | Command Line Interface |
| POM | Project Object Model |
| JMX | Java Management Extensions |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# DANH MỤC HÌNH VẼ

**…………….🙚O🙘…………….**

# ABSTRACT

**…………….🙚O🙘…………….**

Nowadays, the advent of Cloud Computing with the convenience and reasonable prices makes use of a service like VPS , Cloud VPS or Server become easy (the average cost of $5 a month for Cloud VPS). However, there are many problems in the management server for beginners. To resolve the difficulties that we decided to implement the thesis "Building management systems, network services controlled web based".

The main purpose of this thesis is to develop a set of tools to remotely administer network services of the server, supports system administrator managing server efficiently and quickly./.

# TÓM TẮT

**…………….🙚O🙘…………….**

Hiện nay, với giá thành tương đối rẻ, và sự tiện dụng cùng với sự ra đời của Cloud, thì rất dễ dàng để có thể sử dụng một dịch vụ VPS, Cloud VPS, hoặc Server (sau đây xin gọi chung là “Server”) với giá chỉ từ 100.000đ một tháng với Cloud VPS. Bên cạnh đó vấn đề quản lý tập trung các Server ở nhiều nhà cung cấp và đối với người không chuyên cũng trở nên khó khăn cấp thiết. Để giải quyết phần nào khó khăn đó chúng tôi đã thực hiện đề tài “Xây dựng hệ thống quản lý, điều khiển các dịch vụ mạng bằng giao diện WEB”. Mục đích của đề tài là xây dựng một công cụ quản trị tập trung các dịch vụ mạng trên Server từ xa, hỗ trợ người quản trị có thể quản lý các Server của mình một cách hiệu quả và nhanh chóng.

# PHẦN 1: GIỚI THIỆU

### ĐẶT VẤN ĐỀ

Với một người quản trị, hoặc một webmaster sử dụng nhiều dịch vụ Server khác nhau tại nhiều nhà cung cấp khác nhau thì việc cấu hình, quan sát đều phải thực hiện trên mỗi Server được cài đặt và nếu có nhiều Server cài đặt nhiều dịch vụ thì việc quản lý sẽ trở nên khó khăn. Chắc chắn là sẽ gây ra lãng phí về công sức và thời gian để thực hiện tốt công việc đó nếu như không có một giải pháp hỗ trợ. Hiện tại cũng có không ít phần mềm, dịch vụ có thể điều khiển cài đặt các dịch vụ vào Server, nhưng đa phần phải cài đặt vào trong chính Server đó hoặc là hệ thống quản lý của mỗi nhà cung cấp riêng biệt, hay các ứng dụng chạy trên máy tính, sử dụng bằng dòng lệnh phức tạp, chưa thích hợp cho người dùng không chuyên. Để người quản lý có thể nhanh chóng cấu hình, cài đặt dịch vụ trên Server mọi lúc, mọi nơi và thích hợp cho người dùng không chuyên thì một dịch vụ web quản lý tập trung là cần thiết.

Sẽ thật sự rất khó khăn và tốn một thời gian khá lâu để cài đặt hoặc cấu hình nhiều dịch vụ trên các Server nếu không chuyên hoặc không có một hệ thống trợ giúp. Vì vậy, một hệ hống quản lý tập trung các Server, cho phép cấu hình cơ bản các dịch vụ cần thiết bằng các thao tác đơn giản. Hơn nữa còn có khả năng đáp ứng mọi lúc, mọi nơi là thật sự cần thiết.

Hiện tại cũng có một số sản phẩm thương mại lẫn miến phí có hỗ trợ quản lý Server. Tuy nhiên, đa phần là các ứng dụng riêng lẽ chạy trên máy tính hoặc do chính nhà cung cấp dịch vụ đó hổ trợ, ngoài ra nhằm để hiểu được phương thức hoạt động, các công nghệ hiện hữu, khai thác và tỉm hiểu công nghệ nên cần xây dựng hệ thống dựa trên các công nghệ đó.

### LỊCH SỬ GIẢI QUYẾT

Hiện nay đã có nhiều công cụ hỗ trợ quản trị Server từ xa bằng dòng lệnh, hoặc quản trị chuyên sâu được cài đặt trên chính Server đó, nhưng ứng dụng Cấu hình và cài đặt dịch vụ từ xa trền nền web thì rất ít.

Trên thế giới có dịch vụ ServerPilot.io thành lập bởi Justin Samuel vào năm 2012 cung cấp chức năng tương tự, nhưng giá mỗi tháng sử dụng các dịch vụ cơ bản là 10$ hoặc cấu hình các dịch vụ nâng cao hơn với gia 49$.

Ngoài ra các công cụ nổi tiếng như cPanel (cpanel.com), Directadmin (directadmin.com), Webmin (webmin.com),… là các công cụ quản trị, cấu hình chuyên nghiệp từ trả phí đến miễn phí,ví dụ như webmin đòi hỏi phải cài đặt vào mỗi server, hoặc Cpanel hay directadmin phải mua license sử dụng cho từng server khác nhau.

Ở Khoa Công nghệ thông tin và truyền thông – Trường đại học Cần Thơ, có luận văn “Thiết kế hệ thống tích hợp hiển thị thông tin, quản lý thiết bị mạng và dịch vụ mạng” – Nguyễn Thành Nhân, Nguyễn Hữu Nhật - 2012 đã cho ra ứng dụng chức năng như vậy và đã có sản phẩm trên Desktop và tôi tham khảo nghiên cứu này để triển khai lên web và phát triển cho phép nhiều người sử dụng, khắc phục được việc phải sử dụng root trong cấu hình mà chỉ cần dùng tài khoản sudoer user.

Các sản phẩm, đề tài trên thường là những ứng dụng Desktop, không thể truy cập mọi lúc mọi nơi trên nhiều máy và nhiều hệ điều hành khác nhau, hay các ứng dụng chỉ chạy trên chính Server được cài đặt, không thể cùng quản lý đồng thời nhiều máy chủ. Vì vậy, với đề tài này sẽ cho phép tích hợp các chức năng quản trị từ xa, quản lý tập trung nhiều Server và chạy trên nền web để có thể truy cập mọi lúc, mọi nơi và đa dạng hệ điều hành, thiết bị khác nhau vì chạy trên nền Web.

### MỤC TIÊU ĐỀ TÀI

**Mục đích tổng quát**: Tìm hiểu và vận dụng các công nghệ để xây dựng hệ thống điều khiển, cấu hình và theo dõi Server từ xa qua giao thức SSH.

Việc xây dựng hệ thống được tiến hành theo cách chia Module bao gồm 2 thành phần chính: Module giao diện hệ thống web với nhiệm vụ quản lý người dùng quản lý thông tin Server và kết nối các thành phần, Module cung cấp thư viện để thao tác cài đặt và theo dõi các dịch vụ trên Server.

Dựa trên phân công, module được đảm trách trong báo cáo này là “Hệ thống giao diện quản lý thông tin và thao tác với server”.

**Mục tiêu cụ thể trong Module này**:

* Xây dựng hệ thống triển khai các module thao tác cấu hình dịch vụ.
* Cho phép nhiều người dùng đăng kí và sử dụng hệ thống.
* Hỗ trợ người dùng thêm quan sát thông tin cơ bản của Server đang hoạt động.
* Hỗ trợ người dùng cài đặt và cấu hình dịch vụ trên giao diện.
* Hỗ trợ Admin xem thông tin và quản lý người dùng.

### ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Đối tượng nghiên cứu trong luận văn này là công nghệ lập trình web sử dụng Spring WEB MVC kết hợp với Bootstrap phục vụ cho việc cài đặt và cấu hình các dịch vụ mạng trên máy chủ từ xa thông qua giao thức SSH.

Ứng dụng hoạt động trên nền localhost, máy chủ Apache Tomcat và hệ quản trị cơ sở dữ liệu MongoDB.

Phạm vi chức năng chính giới thiệu trong báo cáo này là:

* Thiết kế hệ thống trên nền Spring WEB MVC.
* Quản lý người dùng.
* Quản lý các chức năng điều khiển Server: Network Interfaces, SSH
* Quản lý cài đặt dịch vụ (module chỉ giới hạn ở 2 dịch vụ) trên Server: DHCP, FTP.

### NỘI DUNG NGHIÊN CỨU

Được chia công việc theo thời gian với những công việc cụ thể:

* Tuần 1: Phân tích nhu cầu và mục tiêu của đề tài, tìm kiếm và tham khảo các tài liệu liên quan.
* Tuần 2: Tìm hiểu về Spring Tool, MongoDB và cài đặt.
* Tuần 3: Nghiên cứu cấu hình và lập trình thao tác cơ bản với spring MVC.
* Tuần 4: Nghiên cứu sử dụng Tiles Template để thiết kế giao diện cho website với Spring MVC.
* Tuần 5: Tìm hiểu cấu hình và thao tác Spring MVC với MongoDB, Xây dựng chức năng đăng kí và chứng thực người dùng.
* Tuần 6: Xây dựng chức năng thêm và quản lý thông tin Server.
* Tuần 7: Nghiên cứu kết nối và thao tác với server theo giao thức SSH bằng java thông qua thư viện jSch.
* Tuần 8,9,10,11,12,13,14,15: Nghiên cứu kết hợp các hàm điều khiển vào hệ thống
  + Tuần 8,9: Nghiên cứu cải tiến chức năng để có thể cấu hình dịch vụ với quyền Sudoer mà không cần user root. Xây dựng và thử nghiệm chức năng cấu hình Network Card Interfaces với sudoer user.
  + Tuần 10: Xây dựng và thử nghiệm chức năng cấu hình dịch vụ SSH.
  + Tuần 11,12: Xây dựng và thử nghiệm chức năng cài đặt và cấu hình dịch vụ FTP.
  + Tuần 13,14,15: Xây dựng và thử nghiệm chức năng cài đặt và cấu hình dịch vụ DHCP
* Tuần 16: Xây dựng chức năng cập nhật thông tin Profile của người dùng. Xây dựng module quan sát, lấy thông tin Server và vẽ biểu đồ thể hiện.
* Tuần 17: Xây dựng trang quản lý dành cho Admin, chức năng thống kê cơ bản, chức năng quản lý người dùng.
* Tuần 18: Thử nghiệm tổng thể, và sửa chữa các chức năng điều khiển Server. Viết và chỉnh sửa báo cáo.
* Tuần 19: Bảo vệ luận văn.

### BỐ CỤC LUẬN VĂN

Bố cục luận văn được trình bày gồm 3 phần:

**Phần 1: Giới thiệu**

**Phần 2: Nội dung**

* Chương 1: Mô tả bài toán
* Chương 2: Thiết kế và cài đặt giải pháp
* Chương 3: Kiểm thử và đánh giá

**Phần 3: Kết luận**

* Kết quả đạt được
* Hướng phát triển

Tài liệu tham khảo

Phụ lục

# PHẦN 2: NỘI DUNG

# CHƯƠNG 1: ĐẶT TẢ YÊU CẦU

Chương này trình bày yêu cầu của đề tài, phát triển các tính năng dựa trên nền Spring Web MVC, mô tả kiến trúc của hệ thống, phạm vi chức năng và triển khai của hệ thống.

## Yêu cầu đề tài

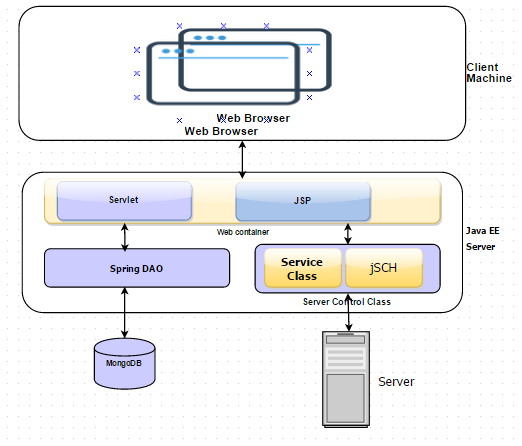
Một Server trước khi sử dụng cần được cài đặt và cấu hình các dịch vụ cần thiết và tùy theo nhu cầu của nhà quản trị, để cấu hình trước tiên cần cài đặt dịch vụ, với mỗi dịch vụ hoặc một chức năng Ubuntu Server sẽ lưu trữ thông tin cấu hình trên một tập tin cấu hình.

Ứng dụng chạy trên nền web, và cho phép người dùng sử dụng, vì vậy tính chất bảo mật, xử lý thông tin lưu trữ của người dùng và server của người dùng rất quan trọng. Vì vậy ứng dụng tạo và sử dụng một user riêng trên server người dùng, nhưng không có quyền hệ thống. Mặc định user này sẽ chỉ nhận các thông tin trạng thái của server người dùng và mật khẩu do người dùng tạo. Khi cấu hình các chức năng nâng cao, mới yêu cầu người dùng nhập thông tin của sudoer user, và chỉ lưu thông tin trong một phiên làm việc với mỗi server. Đề tài hướng đến mục tiêu như sau:

* Đáp ứng yêu cầu nhiều người sử dụng. Có khả năng xử lý trực tuyến.
* Cấu hình dịch vụ không sử dụng root user để tăng bảo mật và chỉ cần sử dụng quyền sudoer.
* Chương trình chạy trên giao diện Web với kiến trúc Client-Server, hệ thống máy chủ ứng dụng phiên bản Ubuntu từ 12.04 hoặc CentOS từ 6.5, các máy trạm (windows xp đến windows 10, các distro Linux), và các thiết bị Smartphone, tablet (Android 4.0 trở lên, IOS 6 trở lên và Windows Phone 8 trở lên).
* Website hỗ trợ bảo mật hạn chế các lỗi dữ liệu chứng thực người dùng để bảo vệ thông tin của người dùng.

## Mô hình kiến trúc hệ thống ứng dụng

Xây dựng ứng dụng nền web với kiến trúc phân tầng và cơ sở dữ liệu tập trung.



*Hình 1.2.1 – Kiến trúc ứng dụng*

Website viết theo cấu trúc Spring MVC với 3 thành phần Model, View và Controller hoạt động xung quanh một DispatcherServlet xử lý tất cả các yêu cầu Http request và trả về Http response. Các công việc xử lý yêu cầu được mô tả trong mô hình sau.



*Hình 2.1.2.2 – Mô hình cấu trúc của Spring Web MVC*

Người dùng sử dụng giao diện thông qua View để sử dụng và thao tác với hệ thống thông qua DispatcherServlet, các controller tương tác với các ServiceClass để thao tác đến server từ xa thông qua JSCH Framwork.

## Phạm vi chức năng

Ứng dụng giải quyết các chức năng sau:

* Cho phép người dùng đăng kí tài khoản và cập nhật thông tin tài khoản.
* Cho phép nhập thông tin nhiều server và quản lý cấu hình nhiều server với mỗi người dùng.
* Cho phép quan sát thông tin tổng quan của server, và cài đặt cấu hình các dịch vụ mà hệ thống cung cấp.
* Cho phép người dùng cấu hình dịch vụ trên giao diện hoặc có thể cấu hình chi tiết dịch vụ bằng dòng lệnh.

## Phạm vi triển khai ứng dụng

Ứng dụng được triển khai trên localhost, hoặc Ubuntu Server 14.04 và cài đặt Jre 7, Apache Tomcat 7, MongoDB3.0

Ứng dụng cho phép người dùng thao tác và cấu hình từ xa các Server chạy hệ điều hành Ubuntu Server 14.04 trở lên, cài đặt và cấu hình Open SSH, enable chứng thực bằng mật khẩu và cho phép kết nối SSH từ xa.

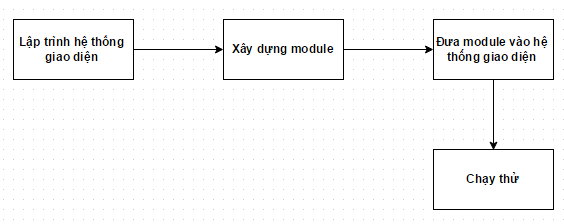
# CHƯƠNG 2: THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP

## THIẾT KẾ GIẢI PHÁP

Chương này trình bày về hướng thiết kế ứng dụng, màn hình giao diện, thiết kế database và sơ đồ cách thức hoạt động của ứng dụng, cách thức cài đăt các thiết kế bằng ngôn ngữ Java và các kĩ thuật liên quan khác.

## Hướng thiết kế hệ thống

Hệ thống điều khiển server được phát triển theo từng giai đoạn với từng module phát triển rộng ra thành một ứng dụng tổng.



*Hình 2.1.1 – Mô hình phát triển ứng dụng*

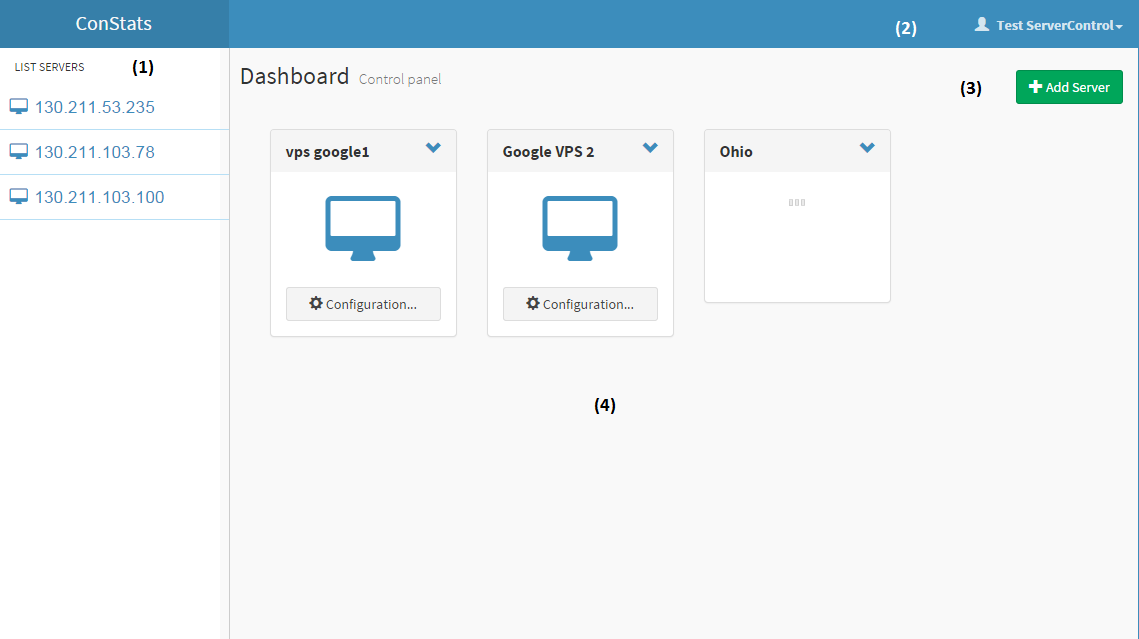
Quy trình được chia làm 4 giai đoạn:

* Giai đoạn 1: Xây dựng hệ thống giao diện web
* Giai đoạn 2: Xây dựng module với các Service Control Class
* Giai đoạn 3: Đưa module vào hệ thống giao diện web
* Giai đoạn 4: Chạy thử nghiệm chức năng
* Quay về giai đoạn 2 để xây dựng module khác.

## Giao diện và sơ đồ hoạt động của các chức năng trong hệ thống quản lý server

* + 1. **Giao diện chính**

Đây là giao diện chính của hệ thống, bao gồm việc hiển thị danh sách các thiết bị và trang thái của từng server.Tại giao diện này có liên kết đến chức năng Add Server.



*Hình 2.2.1 – Giao diện chính của hệ thống.*

Giao diện bao gồm 4 phần:

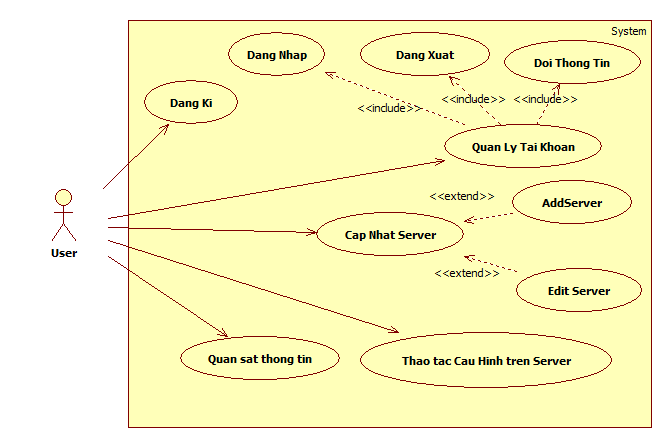
Phần 1: Liệt kê danh sách Server của user đang đăng nhập.

Phần 2: Navbar chứa khung thông tin của user đang đăng nhập.

Phần 3: Nút thêm server.

Phần 4: Vùng hiển thị Server và trạng thái hoạt động của server.

* + 1. **Sơ đồ Use case của hệ thống chính**



*Hình 2.2.2 – Sơ đồ Use Case của hệ thống chính*

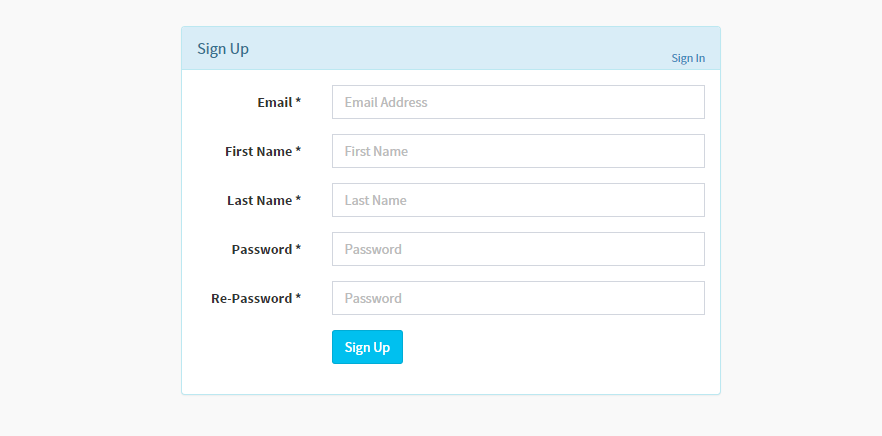
Chi tiết các tác vụ trong sơ đồ:

* + - 1. **UC1 - Đăng kí**

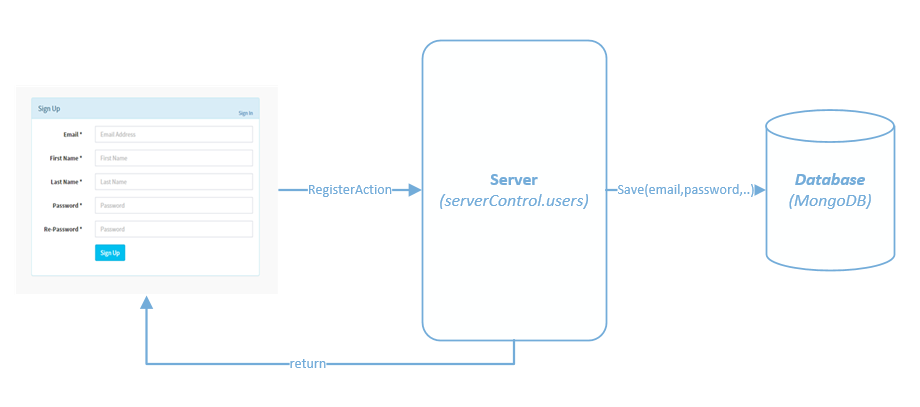
Đây là chức năng đầu tiên của hệ thống, để sử dụng hệ thống, người dùng phải đăng kí thành viên.

Sử dụng Object name Users và document *users* trên mongodb để kết nối dữ liệu.

**Giao diện:**



*Hình – Giao diện đăng kí tài khoản*

**Cách thức hoạt động:**  Màn hình đăng kí có chức năng chính là signup và đường dẫn đến trang đăng nhập (signin).

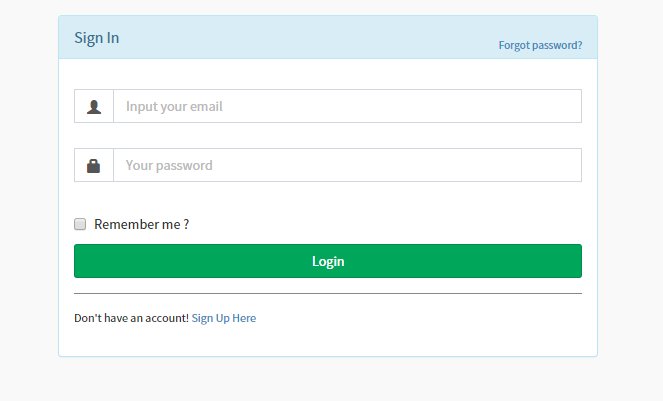
Hình – Sơ đồ chức năng đăng kí (signup)

* Sau khi nhập thông tin đăng kí, client thực hiện kết nối đến server và thực thi request *registerAction().*
* Server khi nhận được yêu cầu sẽ thực hiện 1 truy vấn xuống database (mongodb) để thêm tài khoản với thông tin được truyền từ client.
* Nếu tìm thấy user có cùng email trong database, server trả về kết quả *false* và thông báo email này đã tồn tại, nếu không có trùng thông tin email, hệ thống sẽ tạo một người dùng mới và trả về *true*, đồng thời thông báo đã đăng kí thành công.
  + - 1. **UC2 – Đăng nhập**

Đây là chức năng quan trọng của hệ thống và là cổng đầu tiên để vào sử dụng hệ thống, để sử dụng được các chức năng khác của hệ thống yêu cầu phải login thành công.

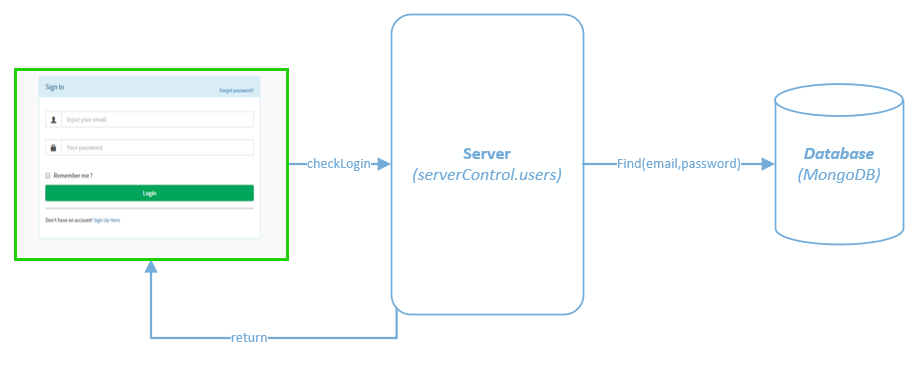
Sử dụng object name users và doccuments trên mongodb là *users* cho client kết nối.

**Giao diện**:



*Hình - Giao diện login*

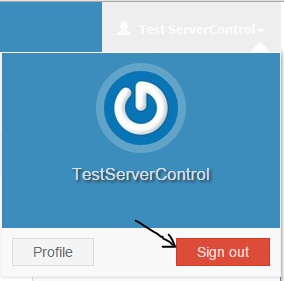
**Cách thức hoạt động**: Ở giao diện login có 2 chức năng chính đăng nhập, đường dẫn đến chức năng đăng kí.

**

*Hình – Mô hình chức năng đăng nhập (login)*

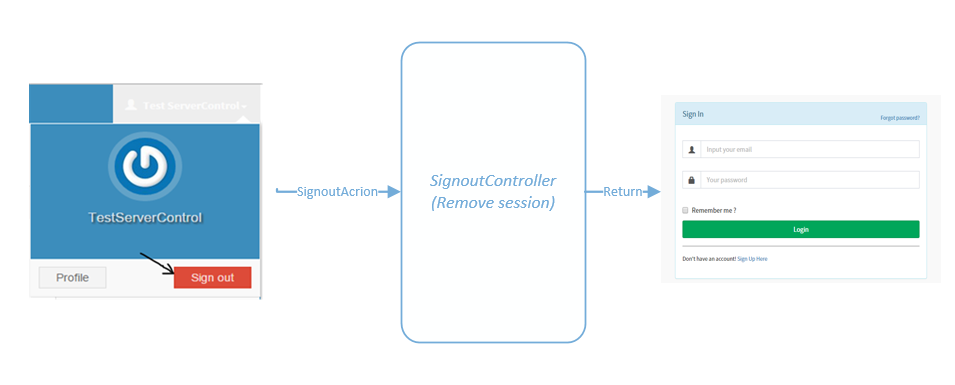
* Sau khi nhập thông tin đăng nhập (email và password), client thực hiện kết nối đến server và thực thi request *checkLogin(email, password).*
* Server khi nhận được yêu cầu xác thực sẽ thực hiện 1 truy vấn xuống database (mongodb) để tìm tài khoản với thông tin được truyền từ client.
* Nếu tìm thấy user trong database, server trả về kết quả *true* và chuyển vào trang Dashboard, nếu không tìm thấy server trả về *false và thông báo lỗi*.
* Nếu client nhận về kết quả từ server là “*true*”, chuyển vào trang quản lý, nếu nhận kết quả trả về “*false*” hiện thông báo và tiếp tục cho người dùng nhập lại thông tin.
  + - 1. **UC3 – Đăng xuất**

**Giao diện:**



*Hình – Đường dẫn gọi chức năng đăng xuất*

**Cách thức hoạt động**: Ở giao diện chính của hệ thống, vào đường dẫn Sign out trong phần thông tin người dùng.

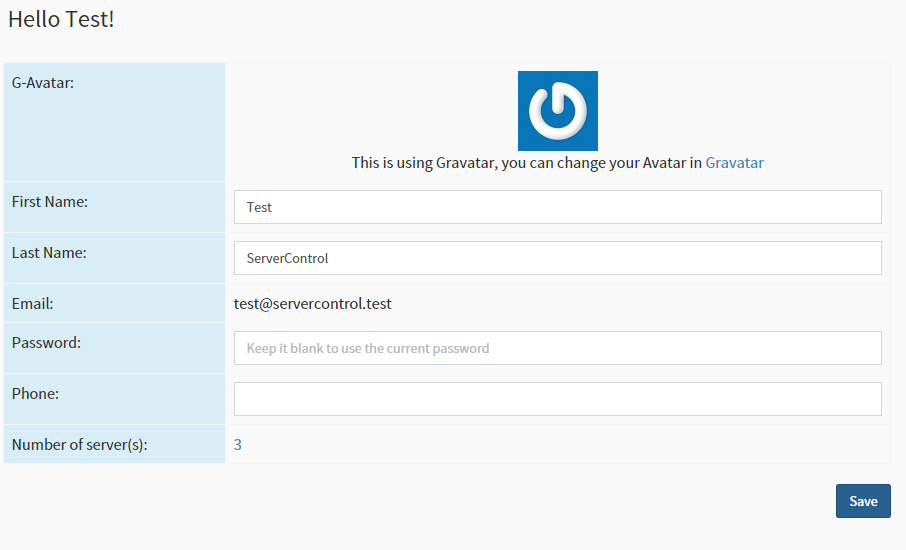


*Hình – Mô hình hoạt động của chức năng đăng xuất*

* Khi client gọi lệnh đăng xuất (SignOut), server tiến hành kiểm tra và thực thi hàm *SignoutAction().*
* Tiến hành xóa các session của user.
* Trở về màn hình đăng nhập.
  + - 1. **UC4 – Đổi thông tin**

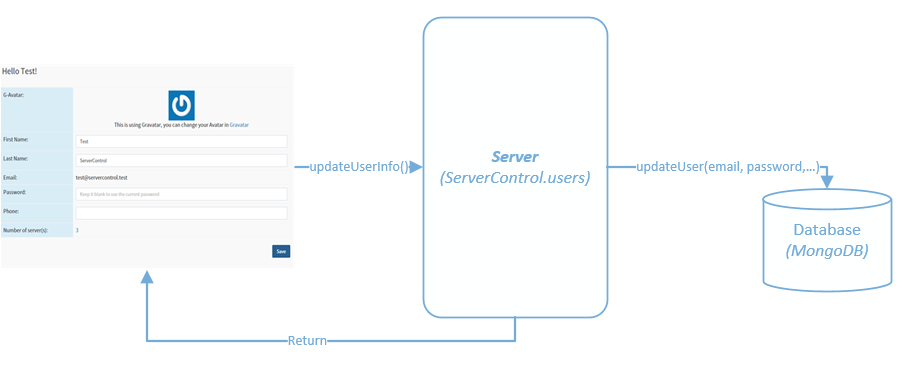
Đây là chức năng cho phép người dùng thay đổi thông tin của profile, mật khẩu, tên, avatar, email, sđt,...

**Giao diện**:



*Hình – Giao diện chức năng đổi thông tin người dùng*

**Cách thức hoạt động**: Ở giao diện chính, người dùng nhấn vào liên kết *“Profile”.*

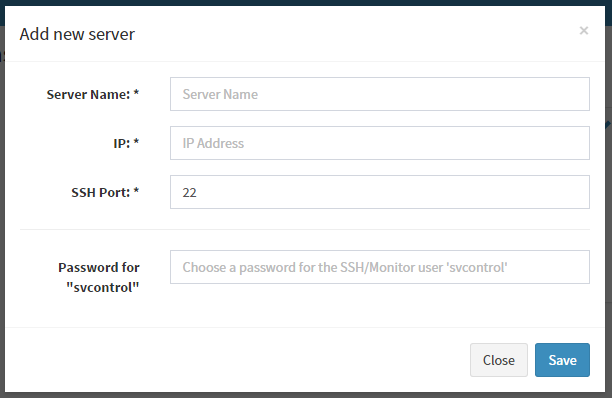


*Hình – Sơ đồ chức năng đổi thông tin người dùng*

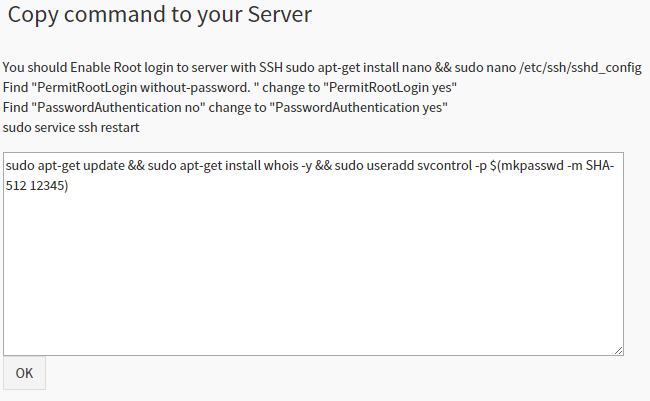
* Người dùng nhập các thông tin yêu cầu, nhấn Save để lưu thay đổi.
* Nếu không muốn đổi mật khẩu, người dùng có thể để trống.
* Người dùng có thể thay đổi avatar bằng các vào trang dịch vụ Gravatar để thay đổi.
* Người dùng không được phép thay đổi email.
  + - 1. **UC5 – Thêm Server (AddServer)**

Đây là chức năng cho phép người dùng thêm thông tin server vào hệ thống để quản lý.

**Giao diện**:

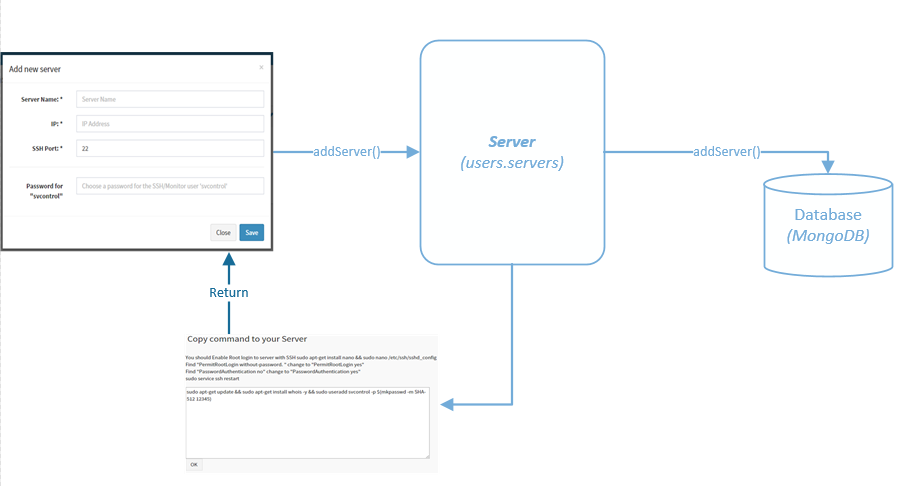


*Hình – Màn hình thêm server*



*Hình – Màn hình yêu cầu cấu hình sau khi thêm Server*

**Cách thức hoạt động**: từ giao diện chính, người dùng nhấn vào nút “Add Server”, màn hình Add new server hiện ra cho phép nhập thông tin server.

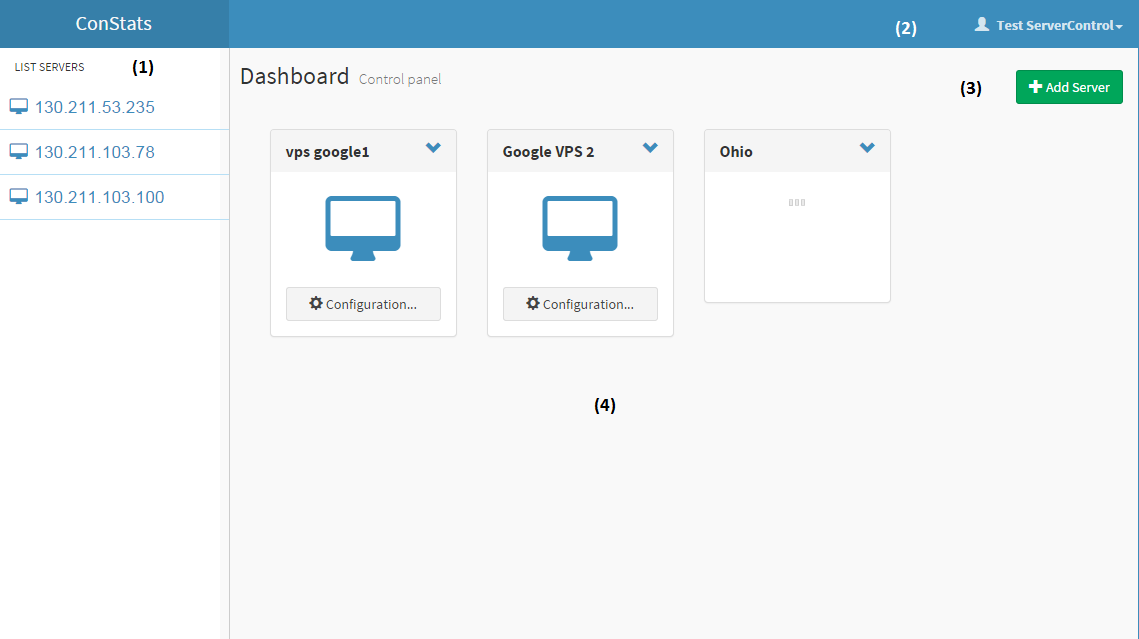


*Hình - Sơ đồ chức năng thêm server*

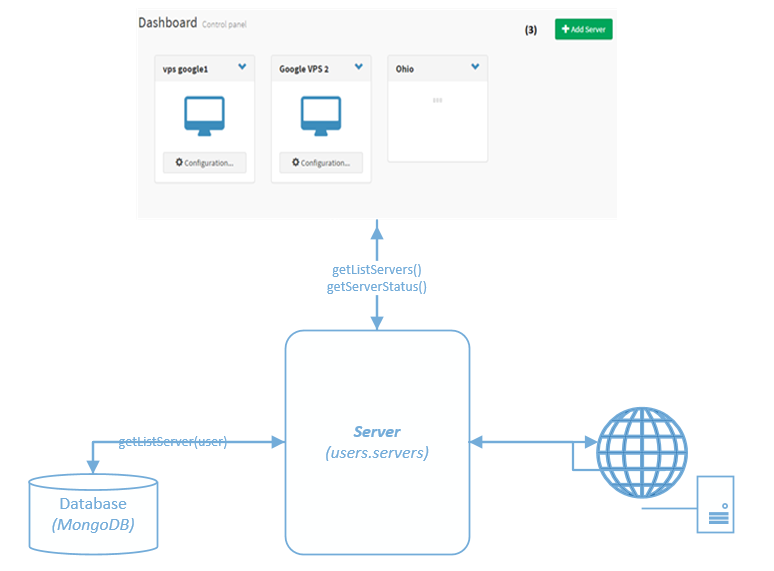
* Server Name: cho phép nhập tên gọi, tên của server để dễ nhận dạng.
* IP: cho phép nhập địa chỉ của server, địa chỉ này là địa chỉ public, hoặc nếu chỉ dùng trong mạng LAN thì có thể là địa chỉ ip mạng Lan của server cần quản lý.
* SSH Port: là cổng kết nối giao thức SSH, mặc định ở đây là 22, người dùng có thể thay đổi.
* Password for “svcontrol”: đặt mật khẩu cho user svcontrol, để sử dụng hệ thống, người dùng phải cấu hình một số lệnh, và lệnh tạo một user bình thường để nhận biết trạng thái của server. Ở đây cho phép người dùng định nghĩa password cho user này.
* Bấm save để lưu thông tin và qua màn hình tiếp theo, bấm close để tắt cửa sổ và quay về màn hình chính.
  + - 1. **UC6 – Hiển thị danh sách Server**

Đây là chức năng ở màn hình chính sau khi đăng nhập, và thêm các server. Và là đường chính để người dùng có thể thao tác với các Servers.

**Giao diện:**



**Cách thức hoạt động:**



*Hình – Sơ đồ chức năng hiển thị danh sách Server và kiểm tra trạng thái server*

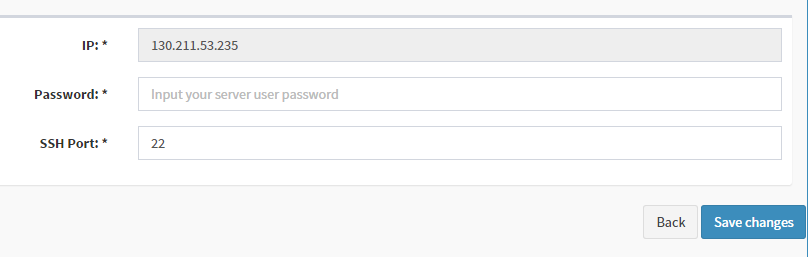
Sau khi đăng nhập vào thệ thống, và đã thêm các Servers, thì hệ thống sẽ lấy thông tin từ cơ sở dữ liệu mongoDB và hiển thị lên giao diện này.

Sau khi hiển thị danh sách server, thư viện Ajax sẽ lấy thông tin trạng thái của Server để hiển thị trạng thái của server và cho phép cấu hình.

* + - 1. **UC7 – Sửa, xóa server**

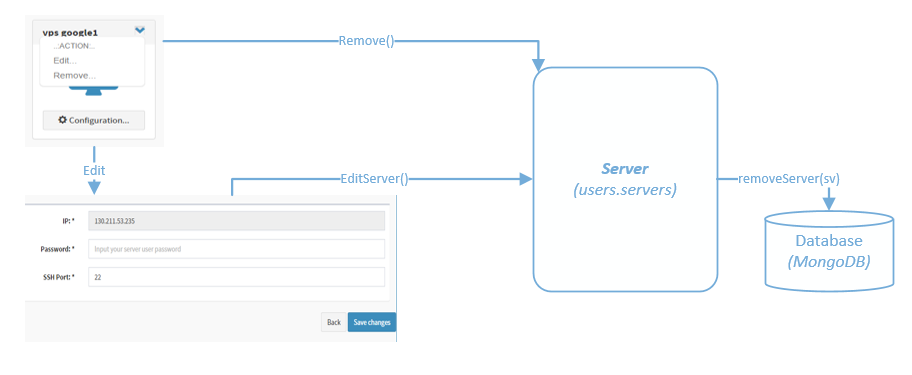
Đây là chức năng Sửa thông tin và xóa server khi đã thêm vào quản lý

**Giao diện**:



*Hình – Giao diện chức năng sửa server*

**Cách thức hoạt động**:



Hình – Sơ đồ chức năng Sửa và xóa Server

Tại màn hình chính, người dùng trỏ và mũi tên ở server cần sửa/xóa chọn Edit.. nếu muốn sửa, chọn Remove nếu muốn xóa.

Khi chọn remove, hệ thống sẽ thực thi hàm RemoveServer() để xóa thông tin server của Ip tương ứng của người dùng tương ứng trong mongoDB.

Khi chọn Edit, màn hình sửa thông tin server hiện ra, người dùng nhập lại thông tin và nhấn Save Changes, nhấn Back để quay lại.

IP không được phép sửa, để đổi IP người dùng phải thêm Server mới.

* + - 1. **UC8 – Thao tác cấu hình Server**

**Cách thức hoạt động:**

Sau khi thêm server, Người dùng nhấn vào Button Configuration... hệ thống sẽ chuyển sang giao diện Quan sát thông tin và Cấu hình server.

Khi hệ thống không kết nối được với server, thì Button Configuration không hiện lên mà sẽ hiện thông tin “Server is off”, khi đó người dùng không thể Quan sát thông tin và Cấu hình server.

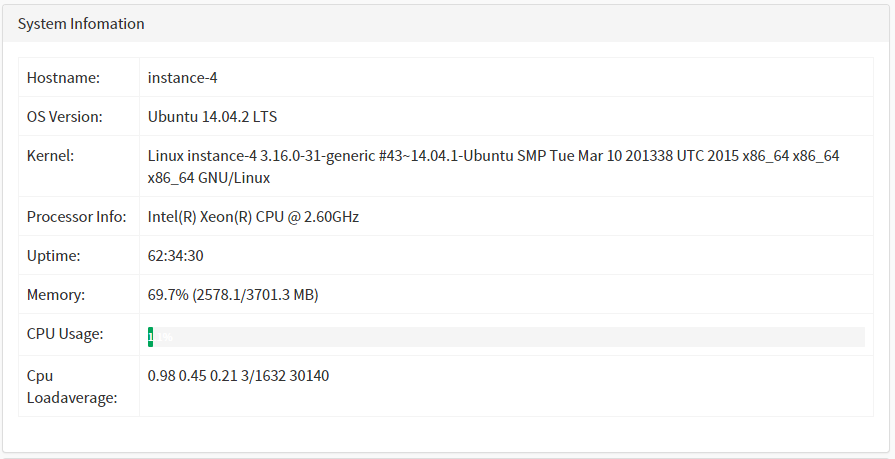
Sau khi vào giao diện Configuation, sẽ có hai tùy chọn: Quan sát thông tin server và Cấu hình service.

Để sử dụng được phần Cấu hình service, hệ thống đòi hỏi phải nhập user và password của SUDOer user.

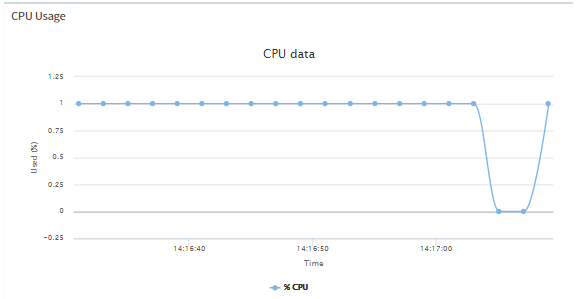
* + - 1. **UC9 – Quan sát thông tin Server**

Đây là chức năng cho phép người dùng xem các thông tin hiện trạng của Server: Tên, hệ điều hành phiển bản hệ điều hành đang chạy, Bộ nhớ, CPU,..

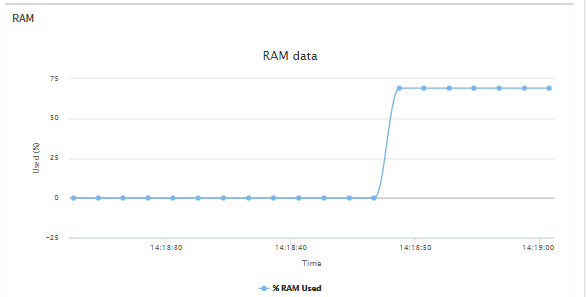
**Giao diện**:



*Hình – Giao diện hiển thị thông tin server*



*Hình – Giao diện quan sát thông tin CPU*



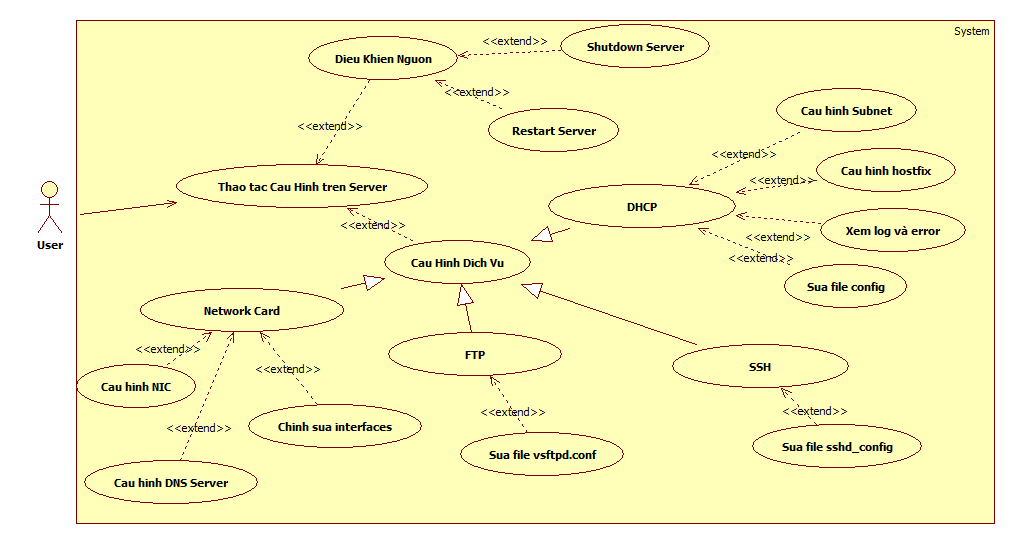
*Hình – Giao diện quan sát thông tin RAM*

**Cách thức hoạt động:**

Ở chức năng này, khi người dùng vào trang hiển thị thông tin, hệ thống sẽ tiến hành lấy thông tin từ server và hiển thị lên giao diện, đồng thời vẽ biễu đồ hoạt động của server dự trên thông tin trạng thái đó.

Thông tin được lấy ngay khi người dùng xem và vẽ biểu đồ hoạt động, ứng dụng chưa lưu được hoạt động của server trong thời gian dài và trong khi tắt hệ thống.

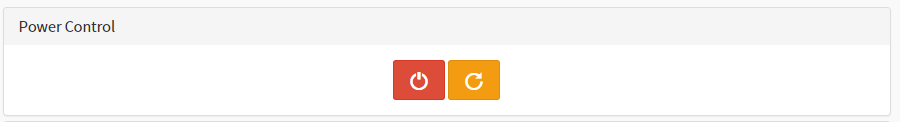
* + 1. **Sơ đồ Use case của Module cấu hình server**



Chi tiết các tác vụ trong sơ đồ

* + - 1. **UC8 – Điều khiển nguồn**

**Giao diện:**



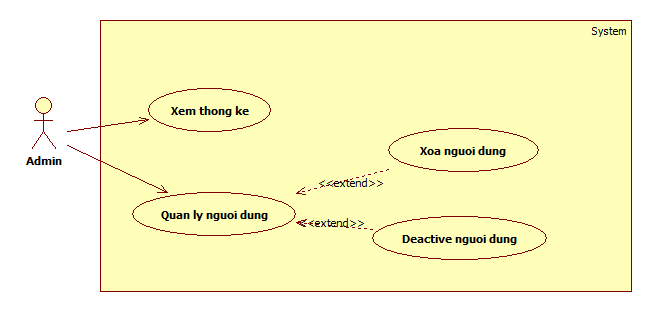
**Cách thức hoạt động:**

Khi người dùng truy cập giao diện Service Config, nhập user/password của sudoer, giao diện điều khiển nguồn sẽ hiển thị như trên.

Khi người dùng chọn vào nút Shutdown (màu đỏ) thì hệ thống hỏi xác nhận có thực hiện thao tác hay không, nếu người dùng bấm Yes, hệ thống sẽ thực thi hàm Shudown(sv) để gửi lệnh yêu cầu shutdown server.

Khi người dùng chọn vào nút Reboot (màu cam) thì hệ thống hỏi xác nhận có thực hiện thao tác hay không, nếu người dùng bấm Yes, hệ thống sẽ thực thi hàm Reboot (sv) để gửi lệnh yêu cầu Reboot server.

* + - 1. **UC9 – Cấu hình network card**
      2. **UC9 – Cấu hình ssh service**
      3. **UC10 – Cài đặt & cấu hình FTP Service**
      4. **UC11 – Cài đặt và cấu hình DHCP Service**
    1. **Sơ đồ Use case của trang quản lý Admin**



* + - 1. **UC11: Chức năng hiển thị thông tin tổng quan**

Chức năng này dành cho admin quản trị hệ thống, giúp admin có thể biết được số lượng User đang sử dụng hệ thống, số lượng Server đang được quản lý.

**Giao diện:**



**Cách thức hoạt động:**

Sau khi admin vào trang này, hệ thống sẽ gọi hàm getStats() và sẽ trả về các thông số cho biết số user đang sử dụng hệ thống (bao gồm cả admin) và các server đang quản lý.

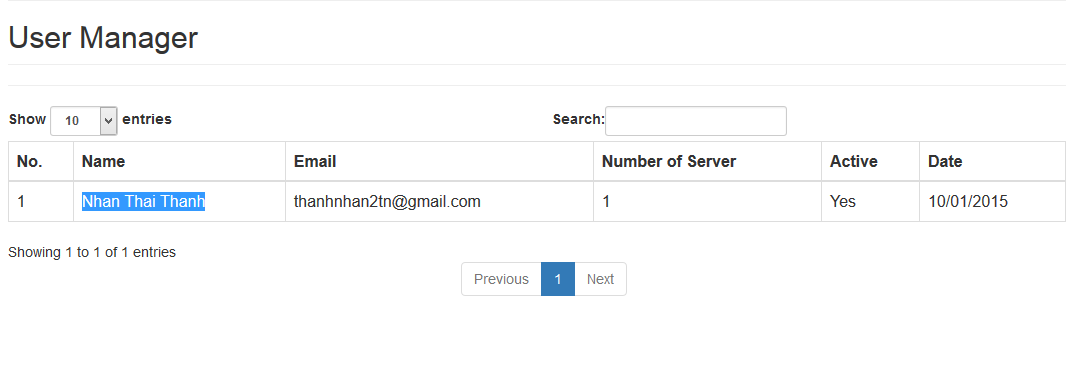
* + - 1. **UC12: Chức năng Quản lý người dùng**

Là chức năng dùng để quản lý user, bao gồm các thao tác cơ bản như: xem thông tin, chuyển user trở thành admin (với user chưa phải admin). Để có thể sử dụng được chức năng này cần phải login với quyền admin.

Sử dụng object name là *user* thực thi hàm, và document “*users*” trong mongoDB để lưu dữ liệu.

Để sử dụng chức năng, từ giao diện login, phải login với quyền admin, sau đó từ giao diện chính chọn vào tên user 🡪Admin hình là giao diện User manager.

**Giao diện:**



*Hình – Giao diện quản lý thông tin người dùng*

**Cách thức hoạt động:**

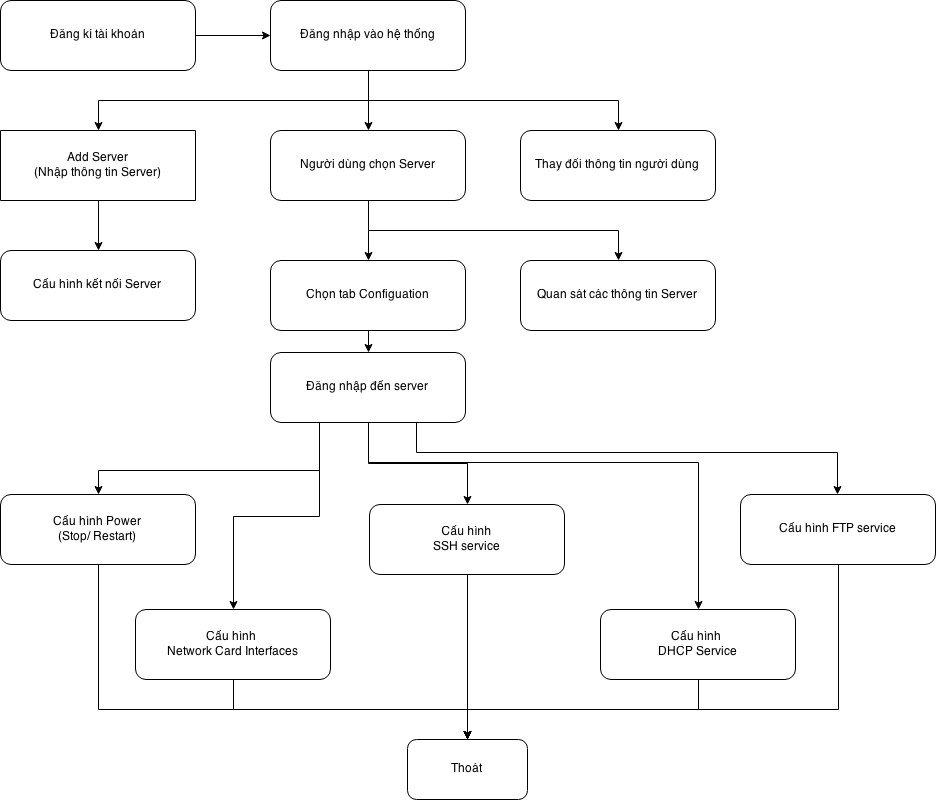
Khi khởi tạo, client thực hiện kết nối đến server thực thi hàm *getAll()* để lấy về danh sách toàn bộ user.

Server khi nhận được yêu cầu, thực hiên truy vấn xuống database lấy về danh sách user và trả về cho client.

Server nhận được yêu cầu tiến hành lưu thông tin vào database.

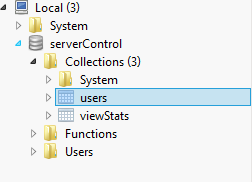
*mongoTemplate.save(user);*

## Sơ đồ Activity của hệ thống



## Thiết kế Database

Hệ thống điều khiển cấu hình server sử dụng mongodb để lưu trữ dữ liệu người dùng, dữ liệu quan sát và thông tin server, nên phần dữ liệu lưu trữ không có cấu trúc, chỉ lưu những thông tin cần thiết như: users, servers… Hình dưới đây là documents (table) của hệ thống dưới sự hiển thị của Robomongo.



Hình 2.2 Dữ liệu lưu trữ user và server

Chức năng mỗi document trong hệ thống:

**Users**: lưu trữ thông tin cơ bản của một người dùng, bao gồm:

* Email, password (được mã hóa dưới dạng md5) .
* Role với các role: -1 – người dùng bị khóa, 1-người dùng đang sử dụng, 2 - admin.
* firstName, lastName: để lưu trữ thông tin người dùng.
* Servers: lưu trữ danh sách các server của người dùng.

Sau đây là dữ liệu mẫu:

*{*

*"\_id" : ObjectId("5549f1bb5cf17d16d494d96f"),*

*"\_class" : "vn.edu.cit.model.User",*

*"email" : "thanhnhan2tn@gmail.com",*

*"passWord" : "25d55ad283aa400af464c76d713c07ad",*

*"role" : 1,*

*"firstName" : "Nhan",*

*"lastName" : "Thai Thanh",*

*"servers" : [ {}]*

*}*

**Servers**: lưu thông tin về server của mỗi người dùng sử dụng hệ thống.

* ServerAddress: địa chỉ kết nối
* Port: cổng SSH để kết nối đến server.
* ServerName: tên người dùng đặt để nhận dạng.
* ServerUsername: tên người dùng, mặc định là svcontrol để nhận biết được các tình trạng cơ bản của server.
* ServerPassword: mật khẩu của người dùng svcontrol
* Status: thể hiện thông tin của servers

*{*

*"serverAddress" : "130.211.103.78",*

*"port" : 22,*

*"serverName" : "Google VPS 2",*

*"serverUsername" : "svcontrol",*

*"serverPassword" : "12345",*

*"status" : []*

*}*

## CÀI ĐẶT GIẢI PHÁP

1. Hệ thống bao gồm các thành phần:

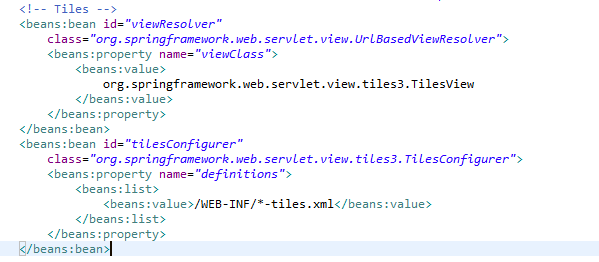
* Một server (dưới dạng Web base) thực thi các hàm do web client gọi, tương tác với database hoặc server của ngườid dùng và trả về kết quả cho client.
* Một hoặc nhiều web client được thiết kế dựa trên các hàm được cung cấp sẵn trên lớp ServiceClass, tạo kết nối đến server và thực thi các hàm tương tác đến server cũng như trên database.

1. Cài đặt cấu trúc template website với Tiles.

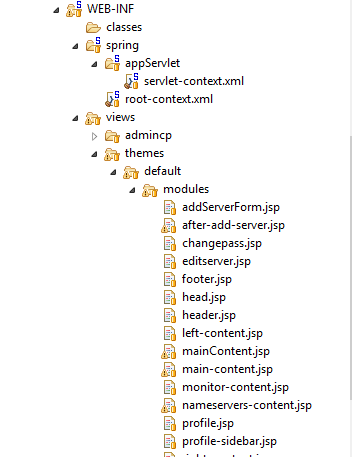


*Hình – Mô tả một giao diện với Tiles*

Cấu hình View trả về phải được khai báo trong servlet-context.xml



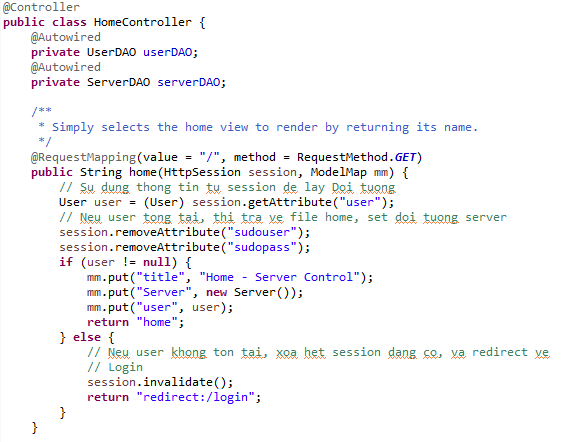
*Hình – Khai báo cấu hình Tiles trong servlet-context*



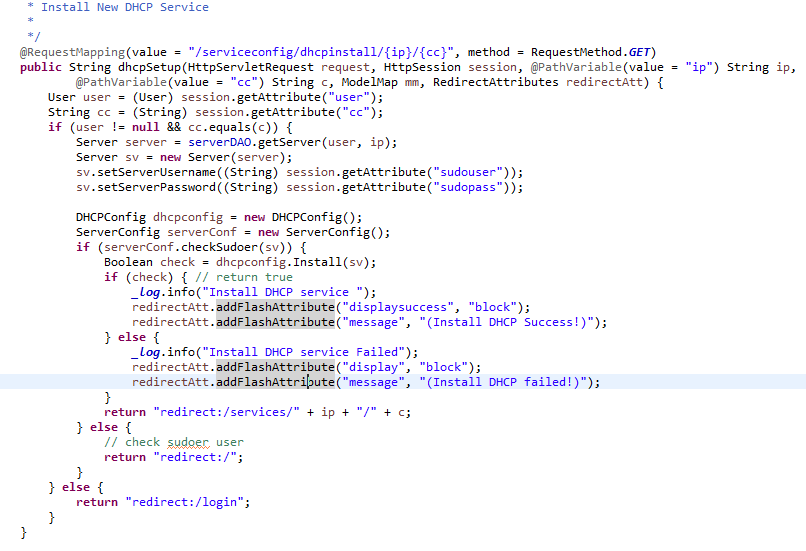
*Hình – Cấu hình cây thư mục các tập tin cho Templates*

1. Xây dựng Controller Mapping để nhận các yêu cầu từ View.

Với một view, được ấu hình như sau



*Hình – Mô tả cài đặt cho một RequestMaping (/)*



*Hinh - Mô tả cài đặt cho một RequestMaping (/dhcpinstall)*

# CHƯƠNG 3: KIỂM THỬ VÀ ĐÁNH GIÁ

# PHẦN 3: KẾT LUẬN

# KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN

**Kết quả đạt được:**

Hệ thống mạng ngày càng rộng lớn, dịch vụ Server càng trở nên đa dạng và số lượng tăng nhanh. Việc quản lý, cấu hình server ngày càng trở nên là vấn đề phức tạp đối với các nhà quản trị. Việc xây dựng được hệ thống hỗ trợ quản lý tiện dụng là vấn đề hết sức cần thiết.

Luận văn này, nghiên cứu tìm hiểu các công nghệ mới và nghiên cứu từ dự án thực tế để rút kết lại những phần cốt lỗi mở ra hướng phát triển ứng dụng một cách nhanh chóng và dể dàng trên các công nghệ mới đó.

Hệ thống nếu được đầu tư và phát triển đến mức toàn diện có thể trở thành hệ thống thực sự cần thiết với các nhà quản trị và tiện dụng đến mức có thể quản trị, cấu hình hay nhận thông báo xử lý lỗi của một hệ thống Server chỉ với thiết bị smartphone.

Những vấn đề đạt được:

* Nắm bắt được phần cốt lõi và triển khai được khung hệ thống.
* Hiểu và áp dụng được giao thức SSH cho cấu hình cũng như lấy thông tin từ server
* Nắm bắt được công cụ git và áp dụng một cách uyển chuyển để quản lý project một cách thuận tiện.

**Hướng phát triển:**

Hệ thống nếu phát triển một các toàn diện:

* Sẽ tiếp tục phát triển hoàn thiện các dịch vụ để có thể cấu hình nâng cao trên Server.
* Phát triển hệ thống giám sát server thời gian thực và có thể cảnh báo người dùng nếu như server quá tải, xảy ra downtime..
* Hỗ trợ tốt đa người dùng.
* Có thể phát triển thêm ứng dụng kết nối với thiết bị di dộng, theo dõi, cảnh báo.

# HẠN CHẾ

Do phạm vi luận văn có giới hạn, nên kết quả chỉ dừng lại ở mức cơ bản với 2 dịch vụ và 2 chức năng hệ thống và chưa tương thích hoàn toàn với các thiết bị cầm tay.

Chỉ sử dụng các thiết bị trên mô hình với sự hỗ trợ của VirtualBox và một vài dịch vụ Server cho phép dùng thử, chưa thể nắm hết được tất cả các dịch vụ Server trên thị trường.

Kết nối đến Server xử lý còn chậm vì qua nhiều bước xử lý kết nối đến server, kiểm tra bảo mật, và ảnh hưởng bởi tốc độ mạng, trải nghiệm người dùng chưa cao.

Chưa thể tiến hành phân tích và theo dõi các dịch vụ, và server một cách định thời mà chỉ thể hiện khi người dùng vào xem hệ thống.

# TÀI LIỆU THAM KHẢO

1. Nguyễn Thành Nhân, Module Cài đặt và theo dõi các dịch vụ mạng, Luận văn: Thiết kế hệ thống tích hợp hiển thị thông tin quản lý thiết bị mạng và dịch vụ mạng 2012.
2. Spring MVC -http://www.tutorialspoint.com/spring/spring\_web\_mvc\_framework.htm
3. http://docs.spring.io/spring-framework/docs/2.0.8/reference/mvc.html
4. JSCH - http://nikunjp.wordpress.com/2011/07/30/remote-ssh-using-jsch-with-expect4j/
5. Giới thiệu về Spring - http://laptrinh.vn/d/3910-gioi-thieu-ve-spring-framework.html
6. MongoDB - http://bigsonata.com/mongodb/
7. Sử dụng MongoDB kết nối với Spring - http://www.journaldev.com/4144/spring-data-mongodb-example-tutorial
8. http://www.mkyong.com/mongodb/spring-data-mongodb-hello-world-example/
9. Vẽ biểu đồ với HighCharts - http://www.highcharts.com/
10. Jsch - http://www.jcraft.com/jsch/examples/
11. Tiles Framework -http://docs.spring.io/spring/docs/3.2.x/spring-framework-reference/html/view.html#view-tiles

# PHỤ LỤC

## PHỤ LỤC: JSCH FRAMEWORK

#### Giới thiệu:

JSCH(Java Secure Channel) là một framework hỗ trợ thực hiện các kết nối thông qua SSH đến thiết bị bằng ngôn ngữ Java.

#### Sử dụng:

Để sử dụng jsch framework, cần thêm dependency trong pom.xml, tùy phiên bản hoặc vào trang chủ maven repository để lấy phiên bản mới nhất:

<!-- Java Secure Chanel -->

<dependency>

<groupId>com.jcraft</groupId>

<artifactId>jsch</artifactId>

<version>0.1.520</version>

</dependency>

Sau khi thêm repository, tiến hành build maven để clipse tải về và cài đặt thư viện vào project, ta sử dụng 1 số hàm cơ bản của jsch để tạo kết nối và giữ session cũng như thực thi một số lệnh cơ bản trên thiết bị có hỗ trợ. Sau đây là 1 method chủ đạo trong việc thực thi Command Line Interface (CLI) trong hệ thống.

**public** **boolean** sendCMDToServer(Server sv, String cmd) {

Session ss = sv.getSession(sv);

**try** {

Channel channel = ss.openChannel("exec");

((ChannelExec) channel).setCommand(cmd);

((ChannelExec) channel).setErrStream(System.***err***);

InputStream in = channel.getInputStream();

channel.connect();

**byte**[] tmp = **new** **byte**[1024];

**while** (**true**) {

**while** (in.available() > 0) {

**int** i = in.read(tmp, 0, 1024);

**if** (i < 0)

**break**;

//System.out.print(new String(tmp, 0, i));

}

**if** (channel.isClosed()) {

**if**(in.available()>0) **continue**;

System.***out***.println("exit-status: " + channel.getExitStatus());

**break**;

}

**try**{Thread.*sleep*(1000);}**catch**(Exception ee){}

}

channel.disconnect();

**if** (channel.getExitStatus() == 0) {

**return** **true**;

} **else** {

**return** **false**;

}

} **catch** (Exception e) {

**return** **false**;

}

}

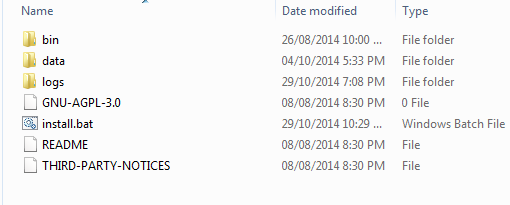
Phương thức này tạo các kết nối đến Server và thực thi lệnh tương tác với server .

## PHỤ LỤC: CÀI ĐẶT VÀ CẤU HÌNH MONGO DATABASE

### CÀI ĐẶT

Tải về phiên bản mongodb tương thích ở địa chỉ: <http://www.mongodb.org/downloads>

Sau khi tải về, gải nén ra thư mục. vào thư mục mongodb tạo 2 thư mục data và logs để lưu dữ liệu và log. Cấu trúc thư mục mongodb:



Tạo file install.bat trong thư mục mongodb với nội dung cấu hình như sau:

*mongod.exe --install --journal*

*--dbpath D:\softs\mongodb-win32-i386-2.6.4\data\db*

*--logpath D:\softs\mongodb-win32-i386-2.6.4\logs\mongodb.logs*

*--serviceName “serverControl”*

*--serviceDescription "ThanhNhan ServerControll"*

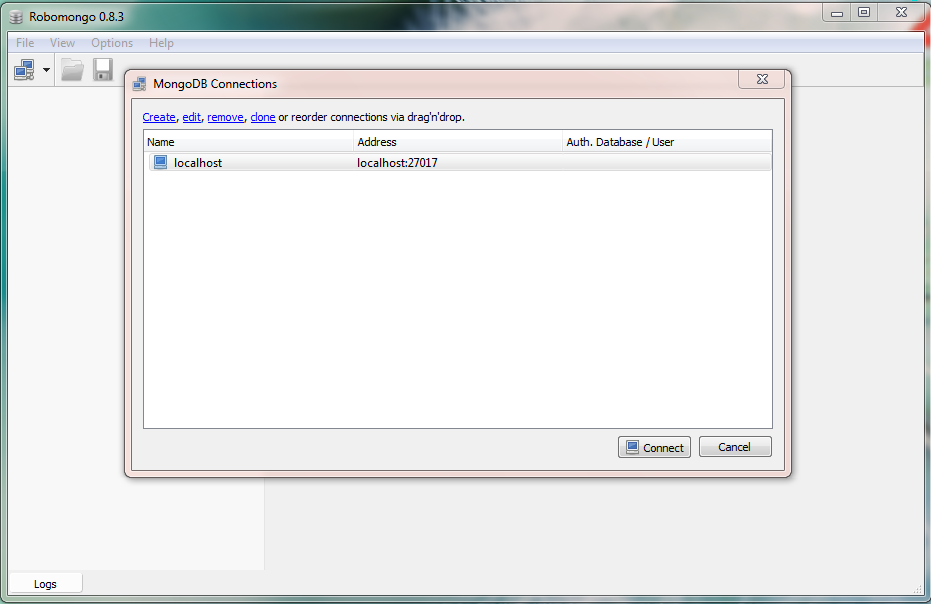
Giải thích tham số trong install.bat:

* *dbpath*: là đường dẫn đến thư mục chứa dữ liệu (data), nên chọn đường dẫn tuyệt đối để tránh phát sinh lỗi.
* *logpath*: là đuồng dẫn đến thư mục chứa log.
* *serviceName*: là tên service sẽ được hệ điều hành quản lý khi cài đặt thành công thành service.
* *serviceDescription*: là phần thông tin chi tiết của service.

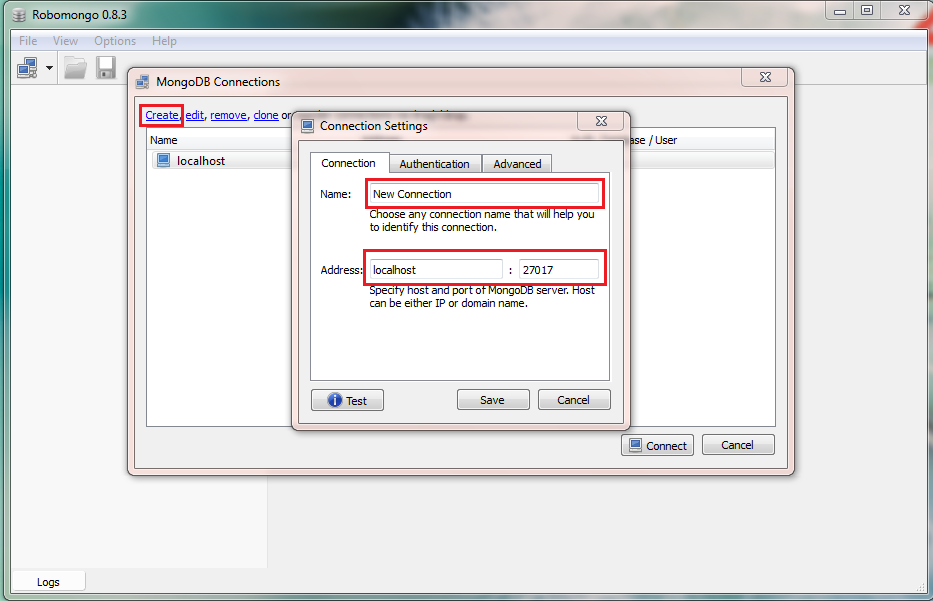
Sau khi tạo file install.bat, mở cmd tại thư mục bin của mongodb và thực thi install.bat.

### TRUY VẤN

Giống như CSDL quan hệ, MongoDB cũng hỗ trợ truy vấn với các câu điều kiện phức tạp. Robomongo là một công cụ cho phép thiết lập kết nối và thực hiện các truy vấn cũng như hiển thị kết quả với các câu truy vấn MongoDB, hình dưới là giao diện tổng quá của Robomongo.



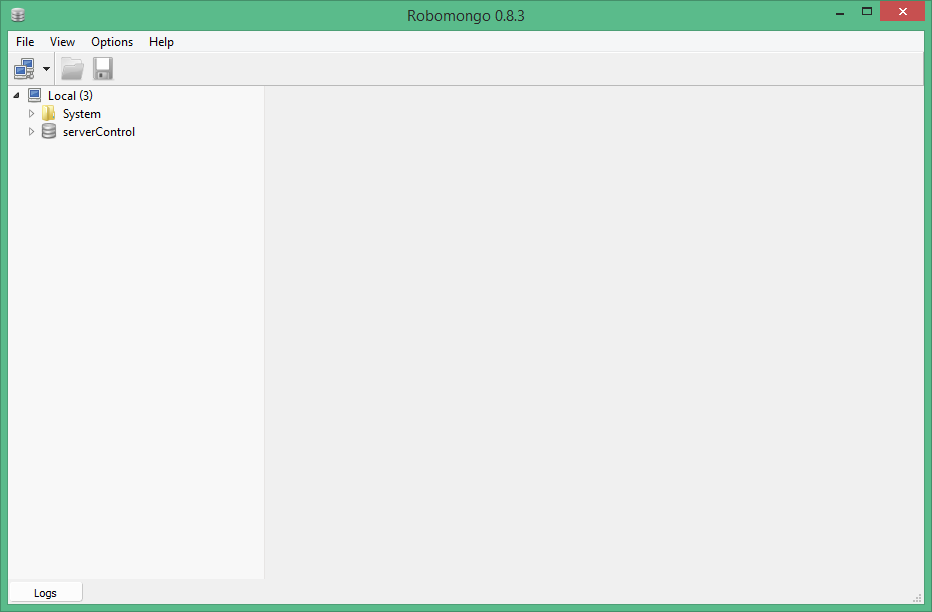
Giao diện tổng quát Robomongo



Tạo mới connection

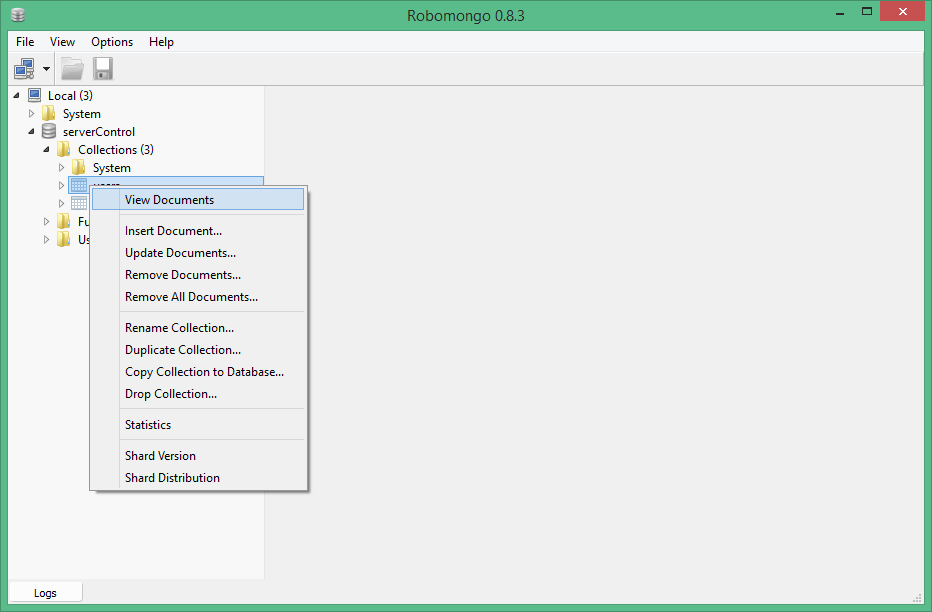
Để tạo kết nối đến MongoDB cần tạo mới 1 connection với 1 số thông tin như sau:

* Chọn Create, 1 popup hiện lên và điền các thông tin cần thiết
* Address: địa chỉ ip server MongoDB
* Port: mặc định là 27017
* Nếu database có cần xác thực thì nhập thông tin xác thực và chọn database, sau đó thực hiện kết nối. Hình 2.14 là giao diện Robomongo khi kết nối thành công đến MongoDB.



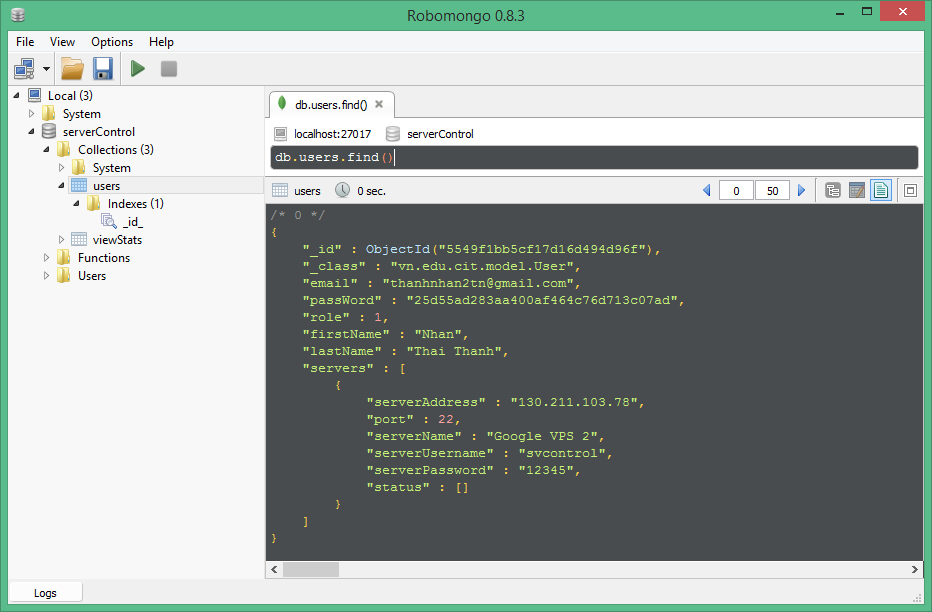
Giao diện kết nối thành công MongoDB

Để hiển thị toàn bộ dữ liệu của một document (table) của một database, chọn vào document và chọn “View Documents”.



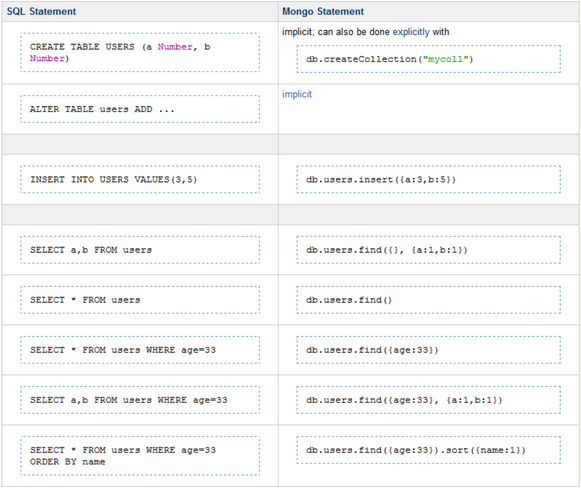
Hiển thị document trong Robomongo

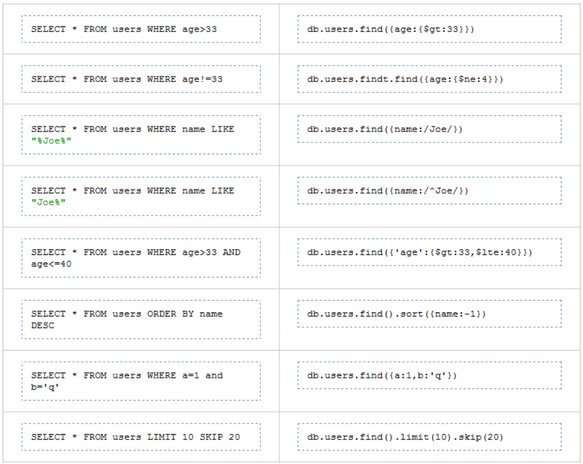
Sau khi hiển thị dữ liệu của 1 document nào đó, ta có thể thực hiện các câu truy vấn như CSDL quan hệ. Các câu truy vấn cần nhập vào khung truy vấn và nhấn f5 để hiển thị kết quả.



Truy vấn trong Robomongo

Sau đây là phần so sánh giữa các câu lệnh truy vấn cơ bản của CSDL quan hệ (mysql) và MongoDB.







## PHỤ LỤC: CÀI ĐẶT VÀ THIẾT LẬP MÔI TRƯỜNG JAVA

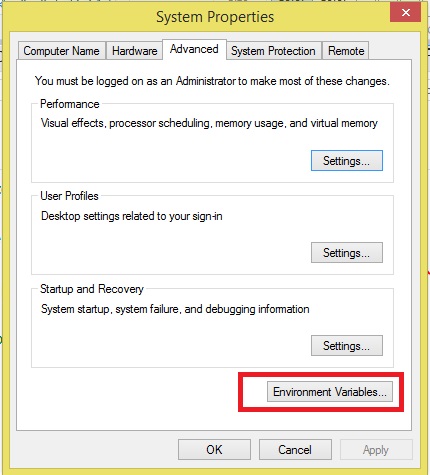
Vào trang:

[*http://www.oracle.com/technetwork/java/javase/downloads/index.html*](http://www.oracle.com/technetwork/java/javase/downloads/index.html) để tải về phiên bản Java phù hợp với hệ điều hành. Sau khi download về cài đặt vào máy tính như bình thường.

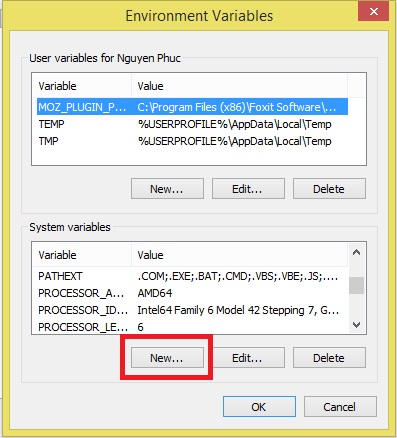
Bây giờ bạn tiếp tục thiết lập môi trường cho JAVA trên máy bạn. Bạn R-Click vào Mycomputer trên Desktop chọn Properties.

Kế đến chọn**Advanced System Setings**:

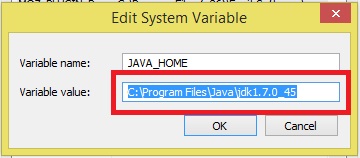
Trong Cửa Số System Properties (sau khi bạn chọn Advanced system settings) chọn Tab Advanced và chọn**Environment Variables….**

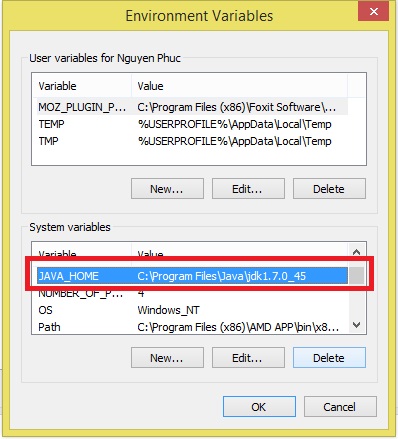
[](http://clbtinhoc.dntu.edu.vn/wp-content/uploads/2013/12/Untitled2.jpg)

Trong cửa sổ**Environment Variables** chọn **New** như trong hình dưới:

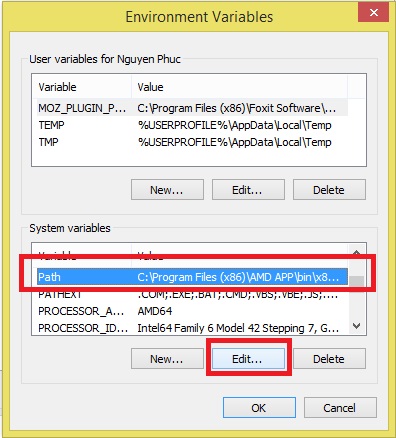
[](http://clbtinhoc.dntu.edu.vn/wp-content/uploads/2013/12/Untitled3.jpg)

Thiết lập như hình bên dưới: Lưu ý phần trong khung đỏ để chính xác tuyệt đối bạn vào (thư mục cài đặt JAVA bạn thiết lập – ở đây tôi cài mặc định) C:\Program Files\JAVA\….. để xem

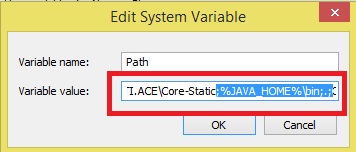
[](http://clbtinhoc.dntu.edu.vn/wp-content/uploads/2013/12/Untitled4.jpg)

[](http://clbtinhoc.dntu.edu.vn/wp-content/uploads/2013/12/Untitled6.jpg)

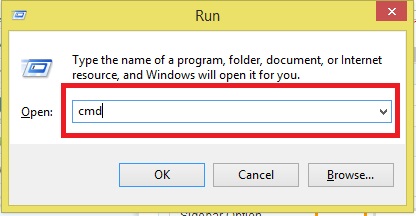
Bây giờ bạn cần tìm đến **Path** trong **System Variables** và chọn edit:

[](http://clbtinhoc.dntu.edu.vn/wp-content/uploads/2013/12/Untitled7.jpg)

Trong cửa sổ **PATH** bạn **THÊM**vào sau những dòng sau: “ **%JAVA\_HOME%\bin”**

[](http://clbtinhoc.dntu.edu.vn/wp-content/uploads/2013/12/Untitled8.jpg)

Mở của sổ cmd để kiểm tra hoạt động của Java.

[](http://clbtinhoc.dntu.edu.vn/wp-content/uploads/2013/12/Untitled9.jpg)

Của sổ CMD xuất hiện. Nhập lệnh “ **Java -version**” để kiểm tra