**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🙘…………….**

# 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, nhưng đòi hỏi phải cài đặt vào mỗi server, và phải mua license sử dụng cho từng server khác nhau nếu là phần mềm trả phí.

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 nhưng chỉ dùng trên máy tính cá nhân, và còn hạn chế là phải sử dụng tài khoản root để quản trị dịch vụ trên Server.

Các sản phẩm, đề tài trên là những ứng dụng Desktop chạy trên máy tính Windows, không thể truy cập mọi lúc mọi nơi trên nhiều máy và nhiều hệ điều hành khác nhau, và 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 để triển khai các module cấu hình dịch vụ.
* Cho phép nhiều người dùng sử dụng hệ thống.
* Cung cấp cho người dùng quản lý thông tin Server.
* Hổ trợ 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.

### ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Đối tượng nghiên cứu trong luận văn này là công nghệ lập trình web với JSP, Servlet, Hệ quản trị MongoDB, Bootstrap Framework. Xây dựng hệ thống có các chức năng cơ bản sau:

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

### NỘI DUNG NGHIÊN CỨU

* Nghiên cứu cấu hình và lập trình với spring MVC.
* Nghiên cứu và thiết kế cấu trúc Tiles Template với Spring MVC.
* Nghiên cứu cấu hình và thao tác với MongoDB.
* Nghiên cứu ssh connection bằng java thông qua jsch.
* Nghiên cứu kết hợp các hàm điều khiển vào hệ thống.
* Sử dụng GIT vào việc quản lý mã nguồn.

### BỐ CỤC LUẬN VĂN

Bố cục luận văn được trình bày gồm 2 phần:

**Phần 1: 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 2: Kết luận**

* Kết quả đạt được
* Hướng phát triển

Tài liệu tham khảo

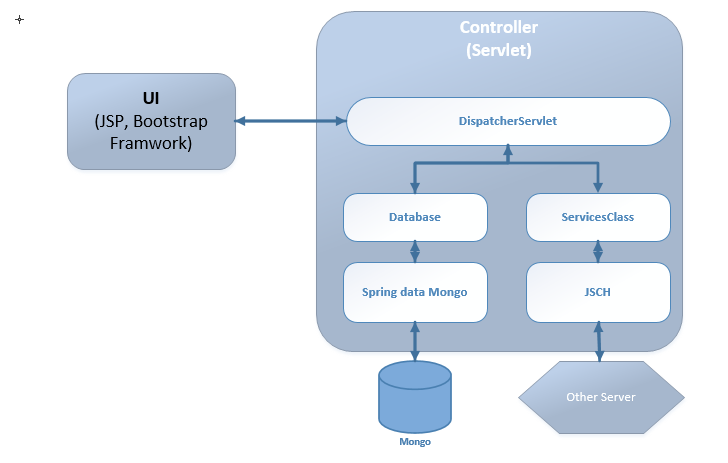
Phụ lục: Hướng dẫn cài đặt, sử dụng chương trình/giải pháp

# 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

# CHƯƠNG 1: MÔ TẢ BÀI TOÁN

# CHƯƠNG 2: THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP



Hình 2.1 - Mô hình cấu trúc tổng quát hệ thống

Hình 2.1 là mô hình tổng quát về hệ thống

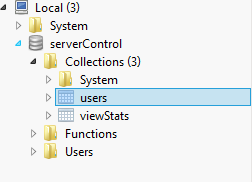
Website viết theo cấu trúc Spring MVC với 3 thành phần Model, View và Controller hoạt động xung quanh một DispatcherServlet xử lý tất cả các yêu cầu Http request và trả về Http response. Các công việc xử lý yêu cầu được mô tả trong mô hình sau.



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.

## Database

Database sử dụng mongodb nên phần dữ liệu lưu trữ có cấu trúc, chỉ lưu những thông tin cần thiết như: user, server… Hình 2.2 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",*

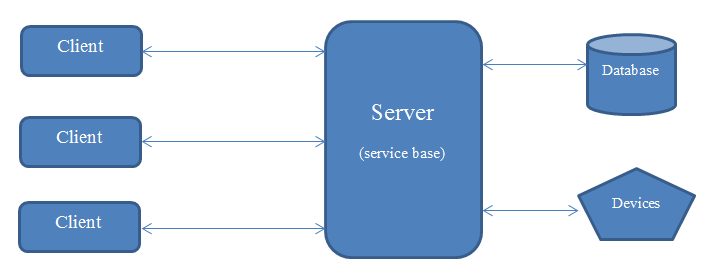
*"serverPassword" : "12345",*

*"status" : []*

*}*

## 2.2 CHỨC NĂNG VÀ CÁCH THỨC HOẠT ĐỘNG CỦA HỆ THỐNG

Hình: mô tả tổng quát cách thức hoạt động của hệ thống.



Hình - Chức năng tổng quát hệ thống

Hệ thống bao gồm 2 thành phần:

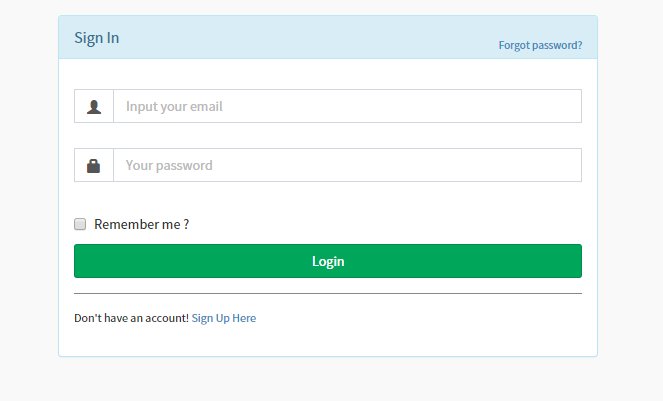
* Một server (dưới dạng Web base) thực thi các hàm do web client gọi, tương tác với database hoặc server của ngườid dùng và trả về kết quả cho client.
* Một hoặc nhiều web client được thiết kế dựa trên các hàm được cung cấp sẵn trên lớp ServiceClass, tạo kết nối đến server và thực thi các hàm tương tác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

### Chức năng login

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

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

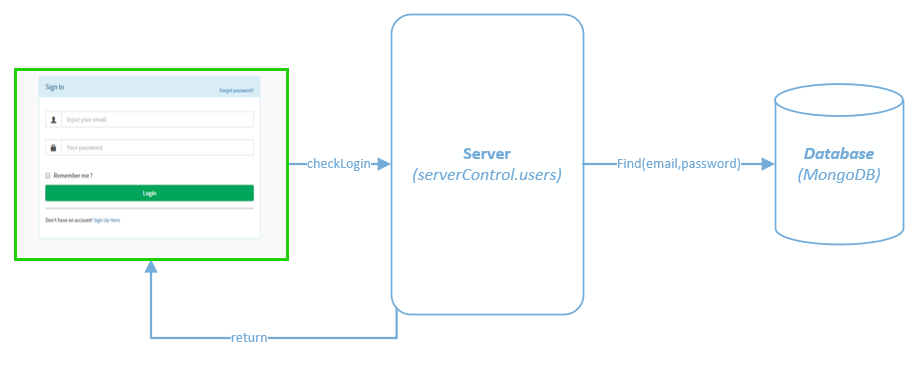
**Giao diện**:



Hình - Giao diện login

**Cách thức hoạt động**: Ở giao diện login có 2 chức năng chính login và exit.

Login:



Hình - Chức năng login

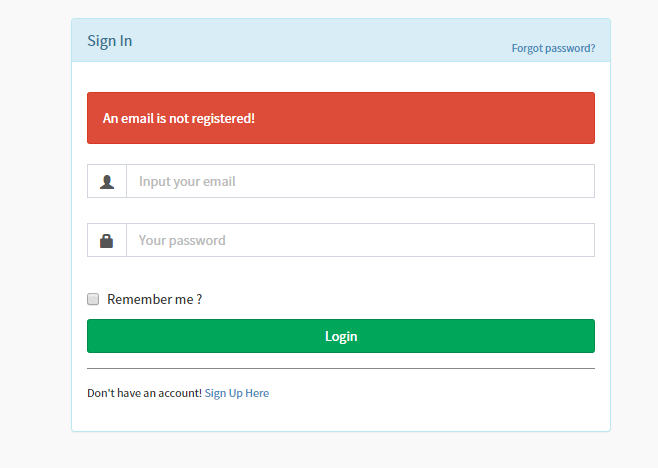
* Sau khi nhập thông tin đăng nhập (email và password), client thực hiện kết nối đến server và thực thi request *checkLogin(email, password).*
* Server khi nhận được yêu cầu xác thực sẽ thực hiện 1 truy vấn xuống database (mongodb) để tìm tài khoản với thông tin được truyền từ client.

User available = mongoTemplate.findOne(new Query(Criteria.where(EMAIL).is(email)), User.class);

* 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.

Kết quả trả về từ server:

* *False*: nếu user không tồn tại, hiển thị thông báo và cho phép nhập lại.



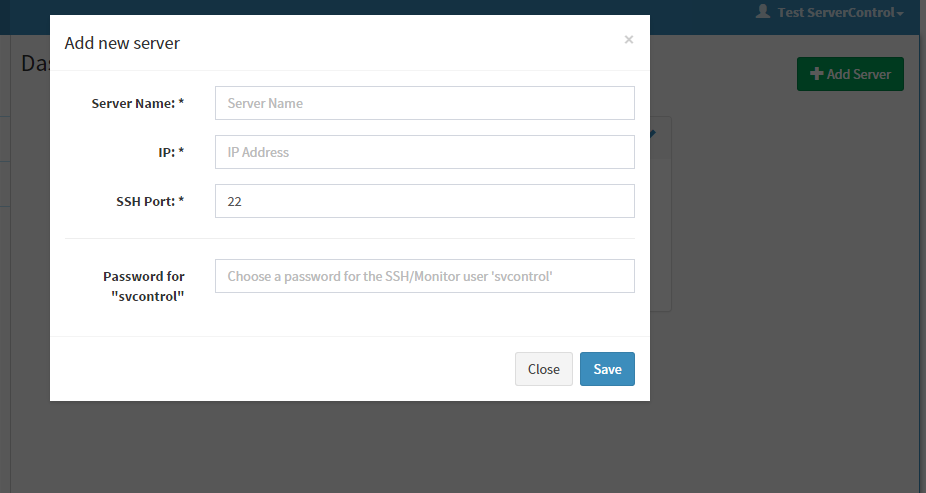
Hình - Sai thông tin đăng nhập

### Chức năng AddServer

Sau khi login thành công vào hệ thống, giao diện quản lý server được hiển thị. Tại giao diện này cho phép người dùng thêm server mới để tiến hành cấu hình.

Sử dụng object name model.Server cho client kết nối và thực thi hàm.

**Giao diện:**



Hình - Giao diện discovery

**Cách thức hoạt động**: Ở giao diện AddServer bao gồm các ô nhập thông tin server và nút xử lý.

Hình 3. 14 - Sơ đồ chức năng AddServer

Save: Thông tin server được lưu vào mongodb, với một document mới trong document users của người dùng.

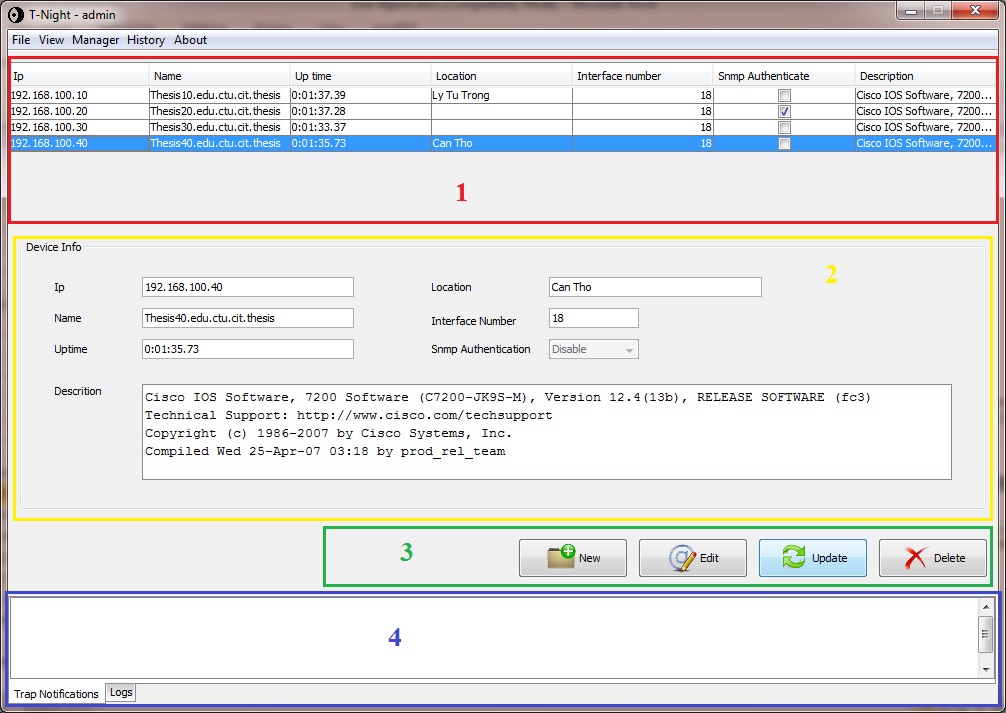
Close: Bỏ qua và đóng cửa sổ.

### Chức năng hiển thị

Đây là giao diện chính của hệ thống, bao gồm việc hiển thị danh sách các thiết bị và thông tin tổng quát của từng thiết bị. Một số chức năng thao tác cơ bản với thiết bị. Tại giao diện này với thanh menu, phím tắt và popup menu cho phép thao tác toàn bộ các chức năng của hệ thống.

Sử dụng 2 object name là *thesis.user* và *thesis.device* phía server cho client kết nối và thực thi hàm.

**Giao diện:**

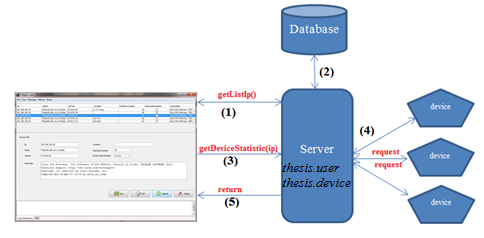


Hình 3. 15 - Giao diện chính

Giao diện gồm 5 thành phần chính:

* (1) Là bảng chứa danh sách các thiết bị có thể được sắp xếp cho tiện dụng.
* (2) Là thông tin cụ thể của từng thiết bị tương ứng với bảng phía trên.
* (3) Là 4 nút chức năng cơ bản.
* (4) Là phần hiển thị thông tin cảnh báo từ thiết bị (trap) và logs từ server.
* Trên cùng là thanh menu bao gồm tất cả các chức năng của hệ thống.

**Cách thức hoạt động:**



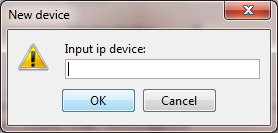
Hình 3. 16 - Mô hình hoạt động khi khởi tạo mainform

Khi khởi tạo form:

* Main form kiểm tra giá trị *skip*, nếu bằng “*true*” client thực hiện kết nối đến server và yêu cầu nhận về danh sách ip (gọi hàm *getListIp())* được lưu trên database.
* Khi server nhận được yêu cầu, thực hiện truy vấn database và trả về danh sách ip (có thể null) cho client.
* Khi có được danh sách các ip, client lặp qua toàn bộ và gởi yêu cầu lấy thông tin chi tiết từng thiết bị (sử dụng hàm *getDeviceStatistic(ip)).*
* Server khi nhận được yêu cầu lấy thông tin thiết bị, thực hiện gởi các gói tin đến ip yêu cầu lấy thông tin (sử dụng snmpget) từ thiết bị và trả về cho client (thông tin được truyền dưới định dạng json).

New:

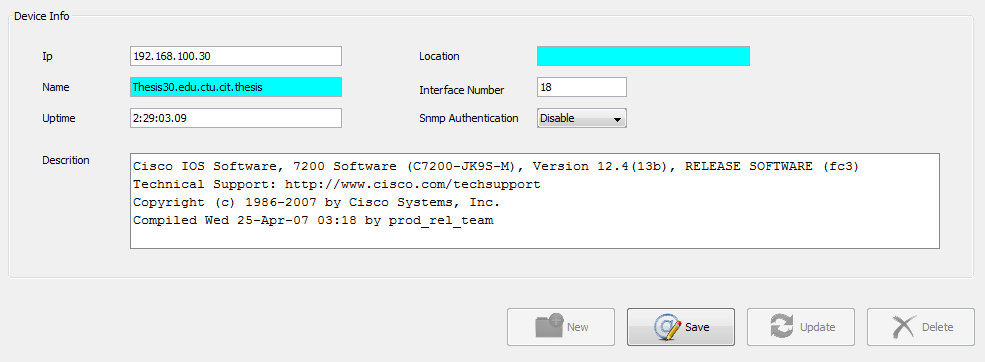
Hiển thị input cho phép nhập vào ip mới.



* Sau khi nhập ip, client thực thi hàm valid(ip) để kiểm tra việc tồn tại thiết bị tại ip đó.
* Nếu kết quả trả về từ server là có (*true*), client tiếp tục yêu cầu lấy thông tin chi tiết và cập nhật vào bảng devices cũng như lưu vào bộ nhớ để quản lý, phía server sẽ thêm ip này vào database cho việc sử dụng lại.

Edit:

Cho phép thay đổi một số thông tin trên thiết bị như name, location.



Hình 3. 17 - Giao diện edit - mainform

* Sau khi bấm nút “Save” client lấy thông tin, chuyển thành định dạng json và gọi hàm *updateDevice(json).*
* Server nhận được yêu cầu, chuyển tham số dạng json dang đối tượng và tạo gói tin snmp (snmp set) gởi đến thiết bị yêu cầu thay đổi giá trị.
* Sau khi thực hiện xong, client tiến hành cập nhật lại thông tin danh sách thiết bị.

Update:

Tiến hành cập nhật lại thông tin toàn bộ thiết bị, giống với việc lấy thông tin lúc khởi tạo form.

Delete:

* Tiến hành xóa device ra khỏi bộ nhớ quản lý của client.
* Gởi yêu cầu xóa ip đã lưu trong database bằng việc gọi hàm *removeDiscoveryIp(ip).*
* Server tiến hành xóa ip ra khỏi database.

mongoTemplate.remove(new Query(Criteria.where("ip").is(ip)), DiscoveryIp.class);

### Chức năng user manager

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.

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:**

**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);*

## 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 thư viện 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.
* Về tính năng: Có thể phát triển thêm ứng dụng kết nối với thiết bị di dộng, theo dõi.

# 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 chạp, và ảnh hướng đến tốc độ mạng, sử dụng ssh thời gian thực thi có phần chậm, trải nghiệm người dùng chưa cao.

# 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. Tổng quan về Spring - http://ducnd94.wordpress.com/2014/07/18/bai-2-tong-quan-ve-spring-framework/
6. Giới thiệu về Spring - http://laptrinh.vn/d/3910-gioi-thieu-ve-spring-framework.html
7. MongoDB - http://bigsonata.com/mongodb/
8. Sử dụng MongoDB kết nối với Spring - http://www.journaldev.com/4144/spring-data-mongodb-example-tutorial
9. http://www.mkyong.com/mongodb/spring-data-mongodb-hello-world-example/
10. Vẽ biểu đồ với HighCharts - http://www.highcharts.com/
11. Jsch - http://www.jcraft.com/jsch/examples/
12. 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**;

}

}

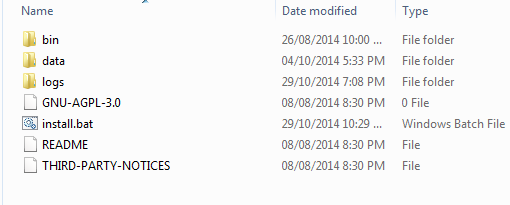
Phương thức này tạo các kết nối đến Server và thực thi lệnh tương tác với server .

## 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*

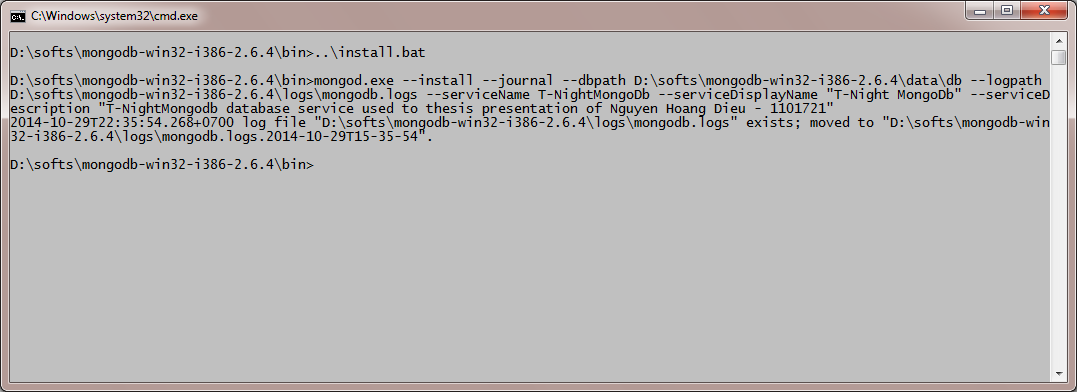
*--serviceName “serverControl”*

*--serviceDescription "ThanhNhan ServerControll"*

Giải thích tham số trong install.bat:

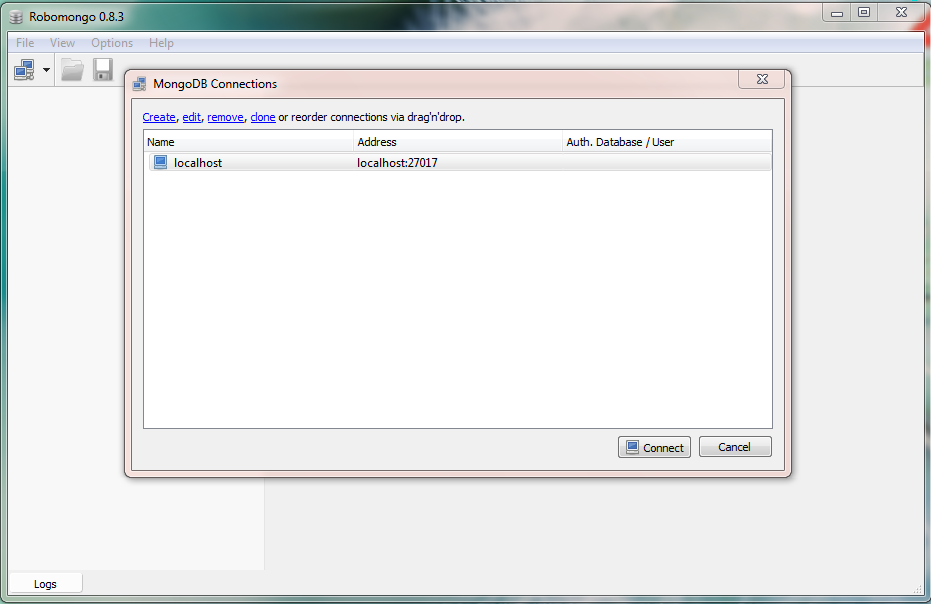
* *dbpath*: là đường dẫn đến thư mục chứa dữ liệu (data), nên chọn đường dẫn tuyệt đối để tránh phát sinh lỗi.
* *logpath*: là đuồng dẫn đến thư mục chứa log.
* *serviceName*: là tên service sẽ được hệ điều hành quản lý khi cài đặt thành công thành service.
* *serviceDescription*: là phần thông tin chi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

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

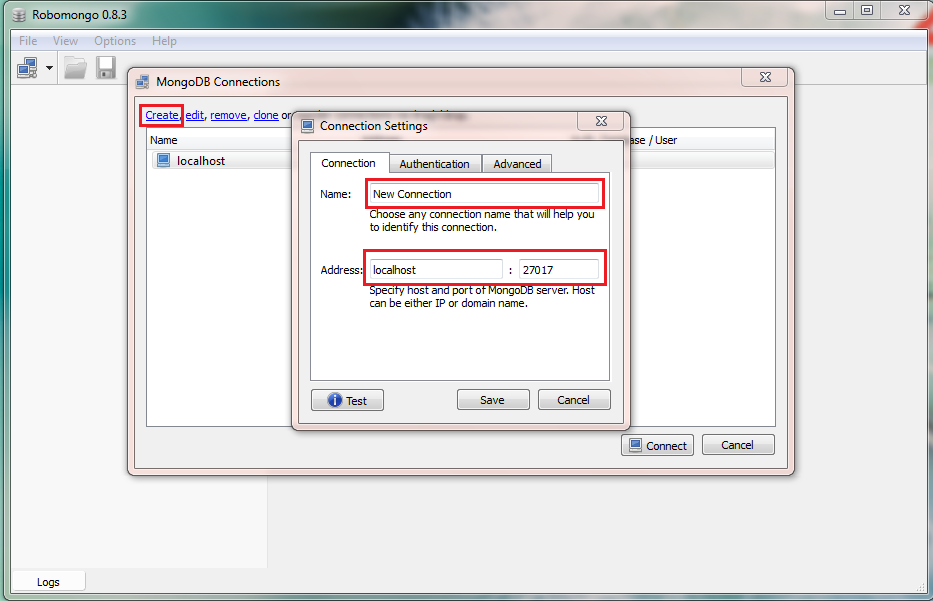


### TRUY VẤN

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.



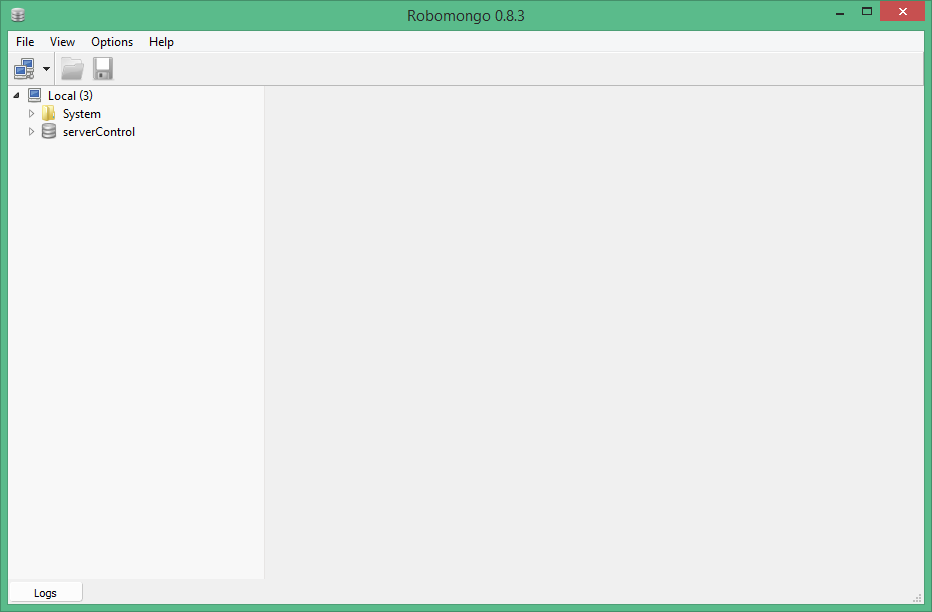
Giao diện tổng quát Robomongo



Tạo mới connection

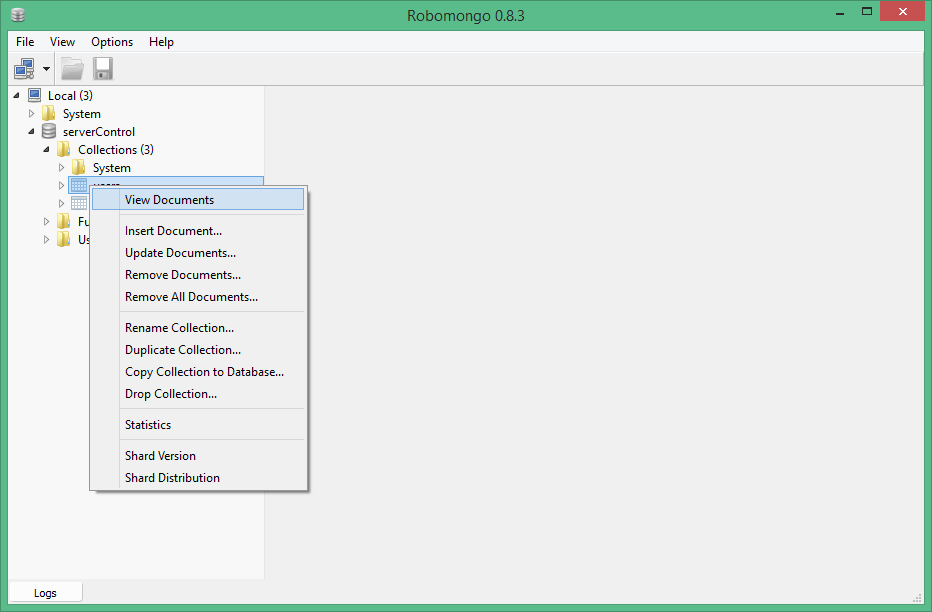
Để tạo kết nối đến MongoDB cần tạo mới 1 connection với 1 số thông tin như sau:

* 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.



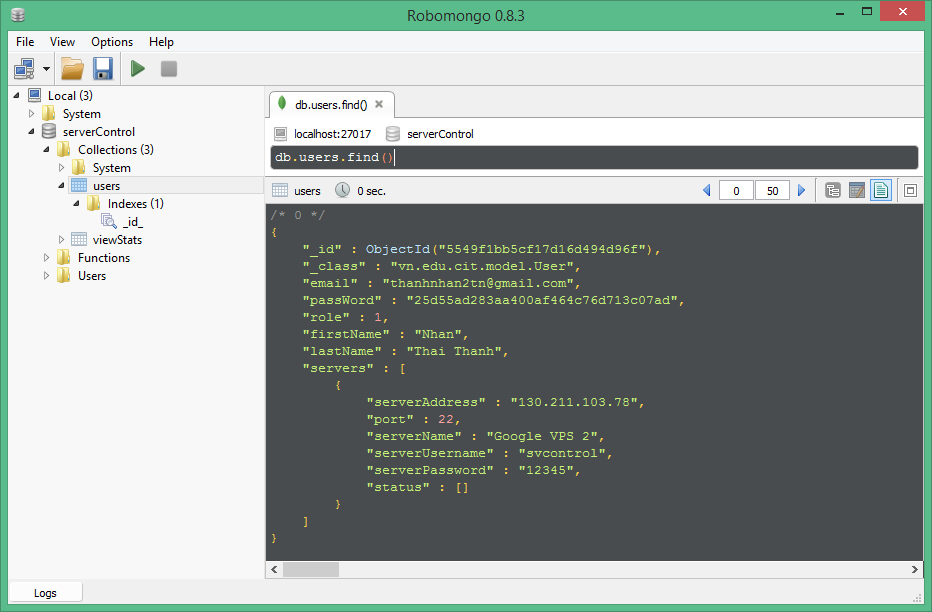
Giao diện kết nối thành công MongoDB

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



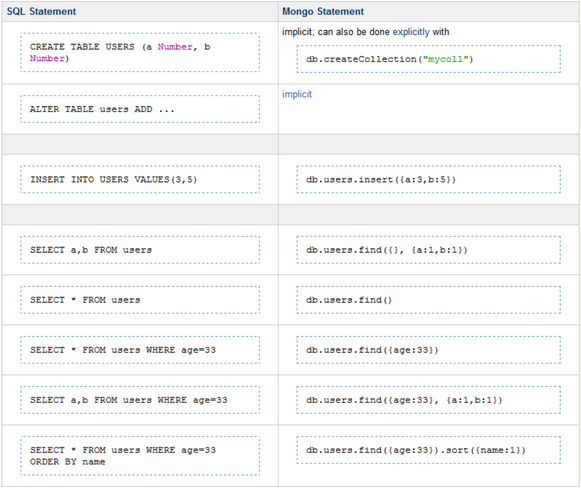
Hiển thị document trong Robomongo

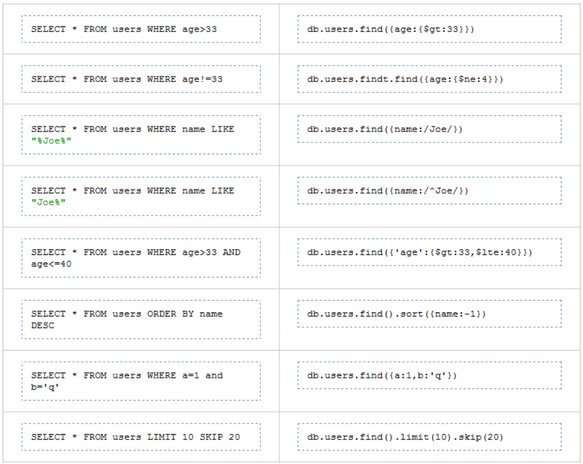
Sau khi hiển thị dữ liệu của 1 document nào đó, ta có thể thực hiện các câu truy vấn như CSDL quan hệ. Các câu truy vấn cần nhập vào khung truy vấn và nhấn f5 để hiển thị kết quả.



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

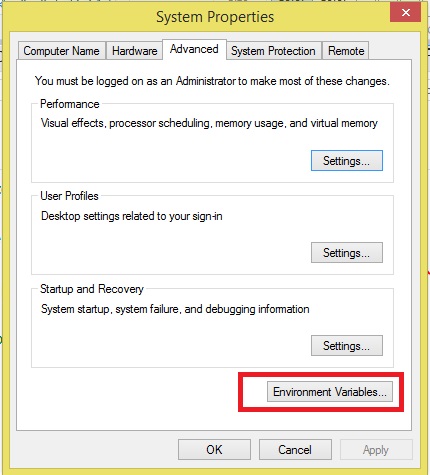
Vào trang:

[*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.

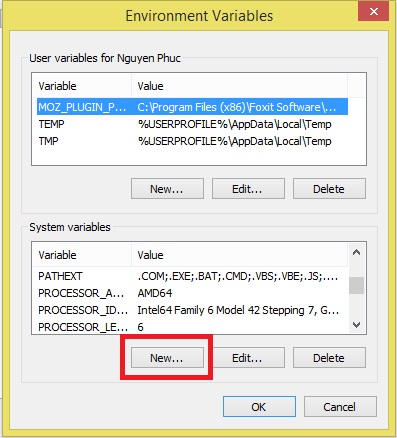
Bây giờ bạn tiếp tục thiết lập môi trường cho JAVA trên máy bạn. Bạn R-Click vào Mycomputer trên Desktop chọn Properties.

Kế đến chọn**Advanced System Setings**:

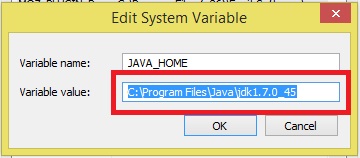
Trong Cửa Số System Properties (sau khi bạn chọn Advanced system settings) chọn Tab Advanced và chọn**Environment Variables….**

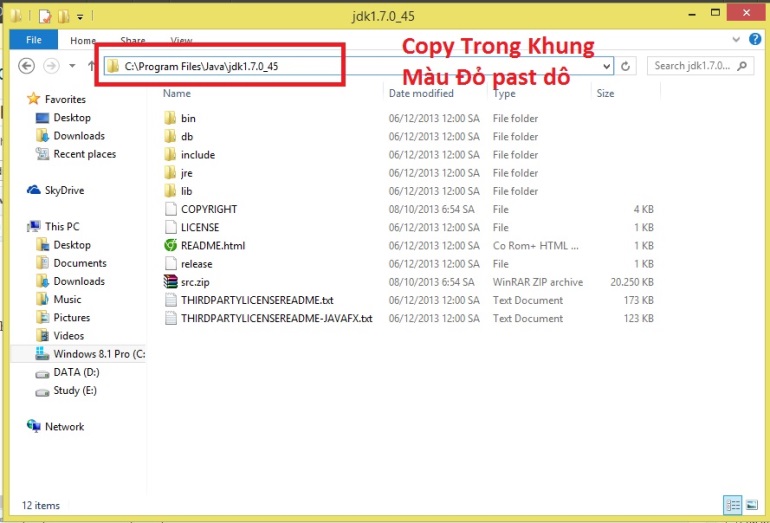
[](http://clbtinhoc.dntu.edu.vn/wp-content/uploads/2013/12/Untitled2.jpg)

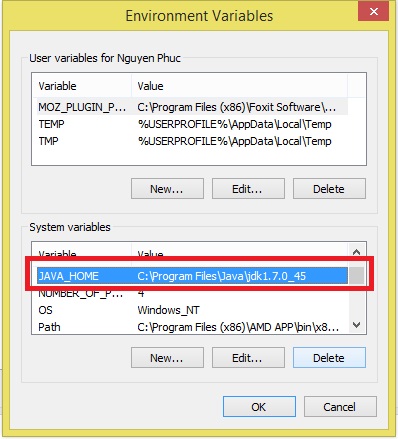
Trong cửa sổ**Environment Variables** chọn **New** như trong hình dưới:

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

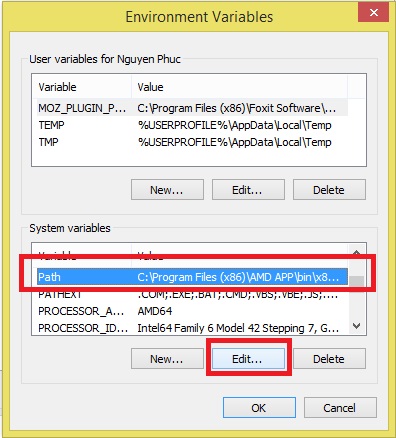
Thiết lập như hình bên dưới: Lưu ý phần trong khung đỏ để chính xác tuyệt đối bạn vào (thư mục cài đặt JAVA bạn thiết lập – ở đây tôi cài mặc định) C:\Program Files\JAVA\….. để xem

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

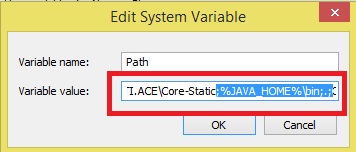
[](http://clbtinhoc.dntu.edu.vn/wp-content/uploads/2013/12/Untitled5.jpg)

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

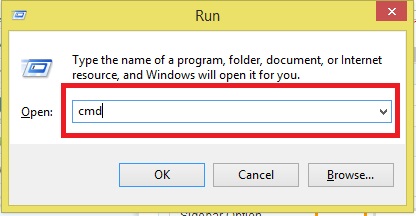
Bây giờ bạn cần tìm đến **Path** trong **System Variables** và chọn edit:

[](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