

COMPANY RATING WEBSITE

ARCHITECTURE DESIGN

*Report #2*

|  |  |  |
| --- | --- | --- |
| **COMPANY RATING WEBSITE – *KẾ SINH NHAI TEAM*** | | |
| **Group Members** | Trần Quốc Hưng | SE02970 |
| Nguyễn Sơn Lâm | SE03090 |
| Nguyễn Việt Hùng | SE03293 |
| Nguyễn Nhật Quang | SE03122 |
| Trần Đăng | SE03039 |
| **Supervisor** | Bùi Đình Chiến | |
| **Project code** | CRW | |

**- Hanoi, 20/06/2016 -**

SIGNATURE PAGE

**AUTHORS**: Trần Quốc Hưng 20/06/2016

*Project Manager*

Nguyễn Nhật Quang 20/06/2016

*Team Member*

**REVIEWER**: Nguyễn Sơn Lâm 20/06/2016

*Team Member*

Trần Đăng 20/06/2016

*Team Member*

**APPROVAL**: Bùi Đình Chiến 03/06/2016

*Supervisor*

Record of change

\*A – Added; M – Modified; D – Deleted

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Effective Date | Changed Item | A,M,D\* | Change Description | Reason for Change | Rev. Number | |
| 20/06/2016 | Architecture Design | A | First version | Create Database Design | | 1.0 |
|  |  |  |  |  | |  |
|  |  |  |  |  | |  |

# Table of Contents

[SIGNATURE PAGE 2](#_Toc459071590)

[Record of change 3](#_Toc459071591)

[Table of Contents 4](#_Toc459071592)

[1. GIỚI THIỆU 5](#_Toc459071593)

[1.1. Mục đích 5](#_Toc459071594)

[1.2. Thuật ngữ và viết tắt 5](#_Toc459071595)

[1.3. Tham khảo 5](#_Toc459071596)

[2. LỰA CHỌN KHI THIẾT KẾ CẤU TRÚC CHƯƠNG TRÌNH 6](#_Toc459071597)

[2.1. Mô hình MVC 6](#_Toc459071598)

[2.1.1. Giới thiệu mô hình MVC 6](#_Toc459071599)

[2.1.2. Điểm mạnh và điểm yếu 6](#_Toc459071600)

[2.1.3. Lý do lựa chọn 7](#_Toc459071601)

[3. Stack MEAN và framework MEAN.JS 8](#_Toc459071602)

[3.1. Giới thiệu MEAN và MEAN.JS 8](#_Toc459071603)

[3.2. Điểm mạnh và điểm yếu 9](#_Toc459071604)

[3.3. Lý do lựa chọn 9](#_Toc459071605)

[4. BIỂU DIỄN CẤU TRÚC 10](#_Toc459071606)

[5. MỤC TIÊU VÀ GIỚI HẠN 11](#_Toc459071607)

[6. CẤU TRÚC CHI TIẾT 12](#_Toc459071608)

[6.1. Package diagram 12](#_Toc459071609)

[1. Class diagram 14](#_Toc459071610)

[2. Sequence diagram ứng với use case 15](#_Toc459071611)

[7. DEPLOYMENT VIEW 34](#_Toc459071612)

# GIỚI THIỆU

## Mục đích

Tài liệu này cung cấp một cái nhìn tổng quan về cả hệ thống, sử dụng một số góc nhìn kiến trúc hệ thống để thể hiện được các mặt khác nhau của toàn bộ hệ thống. Tài liệu cũng trình bày và giải thích tại sao các công nghệ, kiến trúc được chọn lại phù hợp với dự án.

## Thuật ngữ và viết tắt

1. Các thuật ngữ và từ viết tắt trong văn bản và ý nghĩa của chúng

|  |  |  |
| --- | --- | --- |
| Từ | Ý nghĩa | Chú thích |
| CRW | Company Rating Website | Tên dự án |
| MVC | Model-View-Controller |  |
|  |  |  |

## Tham khảo

1. Các tài liệu tham khảo và mở rộng thêm văn bản

|  |  |  |
| --- | --- | --- |
| Tài liệu | Tên | Chú thích |
| SRS | CRW\_Software Requirements Specifications\_v1.0.docx | Tài liệu thuộc Report #1 của nhóm |
|  |  |  |
|  |  |  |

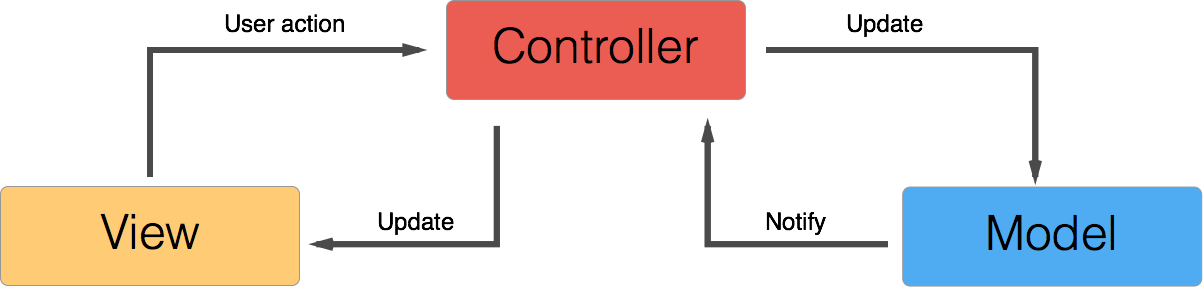
# LỰA CHỌN KHI THIẾT KẾ CẤU TRÚC CHƯƠNG TRÌNH

## Mô hình MVC

### Giới thiệu mô hình MVC

Model-View-Controller, gọi tắt là MVC, là một trong những mô hình cấu trúc thiết kế được sử dụng nhiều nhất trong các dự án cho đến thời điểm hiện tại. Đặc điểm chính của mô hình này là tách biệt vai trò: các đối tượng được chia thành các nhóm dựa vào vai trò của chúng và MVC cũng khuyến khích việc tách biệt các đoạn mã thành các lớp riêng biệt theo vai trò đó.

MVC chia các đối tượng theo 3 vai trò chính: mô hình (model), giao diện (view) và điều khiển (controller).



1. 3 thành phần và liên hệ giữa các thành phần ấy

**Model**

Chứa dữ liệu về đối tượng. Dữ liệu này lấy từ cơ sở dữ liệu hoặc file, và có thể nằm trực tiếp trong ứng dụng hoặc ở trên server khác. Dữ liệu sẽ được truy cập bởi Controller thay vì truy cập trực tiếp bởi View.

**View**

Hiển thị thông tin từ đối tượng tới người dùng. Controller truyền và xử lý dữ liệu trước khi đưa cho view, chỉ đạo việc hiển thị chúng khi nào và ở đâu.

**Controller**

Thành phần chính có trách nhiệm xử lý dữ liệu từ model và truyền vào view. Một controller có thể kết nối nhiều model và nhiều view với nhau. Controller theo dõi tương tác giữa người dùng với view và thông báo thay đổi đó cho model. Ngược lại, các thay đổi từ model cũng được cập nhật trên view. Đây là nơi chứa phần lớn code của hệ thống.

### Điểm mạnh và điểm yếu

**Điểm mạnh**

* Có thể có nhiều view cho cùng một model.
* Tránh lặp code do business logic đã tách biệt khỏi model và có thể chia sẻ.
* Thay đổi một thành phần không gây ảnh hưởng nhiều tới các phần còn lại.
* Khuyến khích phát triển theo hướng module hóa hệ thống, hỗ trợ tái sử dụng tốt.
* Dễ quản lý và thay đổi các phần của hệ thống do cấu trúc phân tầng chi tiết.

**Điểm yếu**

* Không dễ phát triển.
* Quá lớn so với các hệ thống nhỏ và đơn giản.
* Tốn nhiều thời gian thiết kế cấu trúc trước khi đưa vào sử dụng.
* Yêu cầu chú trọng chi tiết do độ phức tạp tăng cao.
* Hiệu năng không cao nếu không có tinh chỉnh

### Lý do lựa chọn

Việc tách biệt nội dung dữ liệu khỏi cách trình bày dữ liệu khiến cho việc lập trình dễ dàng hơn rất nhiều. Các phần có thể được phát triển bởi các nhóm khác nhau và mỗi thay đổi sẽ không gây ảnh hưởng nhiều đến công việc của nhóm còn lại. Việc tách rời cũng có nghĩa là sử dụng lại code sẽ dễ dàng hơn.

Một lợi ích lớn khác nữa là debug và test. Chúng ta có thể dễ dàng khoanh vùng phần bị lỗi hơn với MVC, từ đó xác định sớm lỗi phát sinh từ phần nào và có những thay đổi tương ứng.

# Stack MEAN và framework MEAN.JS

## Giới thiệu MEAN và MEAN.JS

MEAN là một stack phần mềm chuyên được sử dụng để thiết kế trang web động. MEAN là viết tắt của MongoDB, Express.js, Angular.js và Node.js, 4 thành phần tạo nên MEAN. Tất cả các thành phần này đều hỗ trợ ứng dụng viết bằng Javascript và vận chuyển dữ liệu dưới dạng JSON (MongoDB lưu trữ dữ liệu dưới dạng JSON mở rộng mang tên BSON), nhờ thế mà một ứng dụng MEAN có thể được tạo nên chỉ với một ngôn ngữ duy nhất cho cả client và server.

Sơ lược về các thành phần:

**MongoDB** là một cơ sở dữ liệu dạng văn bản thay vì kiểu SQL thông thường. Sở hữu cấu trúc dữ liệu linh động và dễ thay đổi hơn nhiều so với SQL, cùng với khả năng nhúng dữ liệu vào trong các dữ liệu khác, **MongoDB** có tốc độ truy cập cao, cấu trúc đơn giản và khó gặp vấn đề hiệu năng khi mở rộng quy mô trang web. Hỗ trợ chạy Javascript trên dữ liệu ngay trong cơ sở dữ liệu, cùng với việc lưu trữ dưới dạng tương tự JSON, **MongoDB** tích hợp hoàn hảo với việc truyền, gửi và xử lý dữ liệu bằng Javascript.

**Express.js**là một server framework viết trên nền Node.js và được coi là chuẩn framework cho backend khi lập trình bằng **Node.js**. Nhỏ gọn, cấu trúc route đơn giản và hỗ trợ Javascript, **Express.js** giúp lập trình backend trở nên dễ dàng hơn rất nhiều.

**Angular.js** là một client framework được quản lý bởi Google. Sử dụng HTML làm template và thay đổi nội dung hiển thị thông qua các tag riêng, **Angular.js** đảm bảo việc giao diện có thể thay đổi liên tục mà không làm hỏng cách thể hiện nội dung với trang web. Thay vì khiến controller thay đổi nội dung bằng cách phụ thuộc vào các thẻ HTML, việc **Angular.js** biến đổi dữ liệu thông qua các thẻ riêng gắn với HTML giúp tách rời chức năng của controller và view. Nhờ vậy dù view có thay đổi nhưng các thẻ được gắn lại vào vị trí tương ứng trong view, controller có thể thích ứng ngay lập tức với view mới. Tính năng vượt trội của **Angular.js** là kết nối dữ liệu hai chiều: dữ liệu thay đổi trên giao diện sẽ cập nhật ngay lập tức vào hệ thống và ngược lại một cách tự động.

**Node.js** là một môi trường thực thi trên server. Phát triển để đáp ứng nhu cầu cập nhật thời gian thực của server với một ngôn ngữ dễ sử dụng hơn, backend có thể được viết chỉ với Javascript ngay trên nền **Node.js** với engine V8 của Google một cách dễ dàng. Điểm mạnh của **Node.js** là dễ phát triển thêm module mới bằng Javascript và xử lý dữ liệu đầu vào/đầu ra không đồng bộ, giúp cho thời gian xử lý và phản hồi tới các ứng dụng khác và người dùng giảm đi đáng kể, đáp ứng tốt nhu cầu của thời đại. **Node.js** mang khả năng xử lý dựa trên sự kiện của Javascript tới server, nhờ vậy cải thiện đáng kể hiệu năng nhiều dự án khi số lượng người dùng tăng lên.

MEAN.JS là một trong hai framework phát triển từ MEAN với đầy đủ tính năng và rất nổi tiếng trên thị trường. Được phát triển bởi người hoàn thiện khái niệm stack MEAN và đưa nó vào sử dụng trong thực tế cùng với một nhóm các chuyên gia và có sự hỗ trợ của cộng đồng, MEAN.JS hỗ trợ người dùng bằng cách tự động hóa rất nhiều tác vụ trong quá trình phát triển hệ thống; sắp xếp và quản lý các file và folder một cách hiệu quả; tạo một môi trường đầy đủ công cụ phát triển và kiểm thử tự động ngay khi mới cài đặt xong; bổ sung ví dụ và các template giúp viết ứng dụng nhanh gọn hơn.

## Điểm mạnh và điểm yếu

**Điểm mạnh**

* Sử dụng chung một ngôn ngữ cho cả client và server.
* Dữ liệu di truyển với một dạng duy nhất: JSON giúp tăng tốc độ xử lý và viết RESTful service. JSON có thể đọc hiểu bằng mắt thường, hỗ trợ việc debug và test hệ thống.
* Mã nguồn mở nên nhận được sự hỗ trợ rất lớn từ cộng đồng, đặc biệt là các module có thể sử dụng lại.
* Kích thích phát triển phần mềm theo hướng tách biệt vai trò nên dễ bảo trì và sử dụng lại code.
* Xử lý không đồng bộ cho phép server hỗ trợ lượng truy cập và kết nối rất lớn mà không gặp phải trở ngại đáng kể

**Điểm yếu**

* Mới phát triển nên còn thiếu hướng dẫn và tài liệu.
* Yêu cầu trình duyệt phải sử dụng Javascript và xử lý Javascript đúng chuẩn.
* Sử dụng MongoDB nên dữ liệu nếu không cẩn thận có thể không thống nhất giữa các tài liệu do quá tập trung vào hiệu năng

## Lý do lựa chọn

Việc có một framework chỉ sử dụng một ngôn ngữ giúp cho cả nhóm có thể học tập và phát triển hệ thống trong thời gian ngắn nhất có thể. Phân chia tính năng theo các module giúp cả nhóm có thể tham khảo và học hỏi cấu trúc các thành phần trong stack để tạo và tùy chỉnh module riêng cho mình.

MEAN.JS hoàn toàn mã nguồn mở và được hỗ trợ bởi các tổ chức và hệ thống lớn, lại đi kèm với bộ công cụ template và test riêng của mình khiến cho việc thiết lập môi trường làm việc nhẹ nhàng hơn rất nhiều. Có template và hệ thống test chính thức cũng có nghĩa là hầu hết người sử dụng MEAN.JS sẽ dùng chung các công cụ này, việc trao đổi kiến thức và hỏi đáp trở nên thuận tiện và hiệu quả hơn.

# BIỂU DIỄN CẤU TRÚC



1. Khái quát cấu trúc hệ thống

Được xây dựng dựa trên cấu trúc MVC và nền tảng MEAN.JS, hệ thống bao gồm các thành phần chính:

* **Model – View – Controller:** ba thành phần chính của MVC được nói đến ở phần 2. Đáng chú ý hơn cả là việc Controller được tách ra hai phần cho client và server giúp xử lý chuyên biệt và dễ dàng hơn. Việc tách đôi cũng hỗ trợ rất tốt môi trường phát triển độc lập client với server: chỉnh sửa một phía sẽ không gây ảnh hưởng tới phía kia và ngược lại.
* **Facebook / Google API:** hỗ trợ đăng ký tài khoản và đăng nhập thông qua tài khoản Facebook và Google
* **Service:** kết nối theo REST API (REpresentational State Transfer Application Programming Interface) tới các chức năng của server và nhận dữ liệu trả về.
* **Route:** Gắn mỗi loại kết nối của REST API với một hàm trong một controller để thực hiện xử lý và truyền kết quả.
* **Policy:** Giới hạn một số loại kết nối tùy theo kiểu người dùng nhất định (khách, người dùng thường, quản trị viên).
* **Repository:** Kết nổi và thao tác với cơ sở dữ liệu, là lớp trung gian giúp cho lập trình viên dễ quản lý cơ sở dữ liệu hơn.
* **Database:** Cơ sở dữ liệu của trang web: MongoDB.

# MỤC TIÊU VÀ GIỚI HẠN

Đối chiếu với Software Requirements Specifications, phần 4 “Đặc tả phi chức năng”.

Các giới hạn khác:

* Thống nhất các chức năng server cần cung cấp từ đầu, kết nối các chức năng đó tới REST API từ lúc mới bắt đầu thiết kế dự án
* Hệ thống phải hỗ trợ tạo tài khoản và đăng nhập thông qua Facebook, có thể có với Google
* Client ngoài Angular có thể sử dụng thêm các thư viện ngoài như Bootstrap hay jQuery nhưng không nên sử dụng trừ khi có lý do thực sự cần thiết

# CẤU TRÚC CHI TIẾT

## Package diagram

Giới thiệu các package quan trọng của hệ thống cùng với mối liên hệ giữa chúng.



1. Package Diagram

**Model**

1. Danh sách Model trong hệ thống

|  |  |  |
| --- | --- | --- |
| No | Model | Description |
|  | User | Those who have an account on CRW, be it members or administrators |
|  | Company | Basic information on the company for the reviewers to write a review upon |
|  | Review | A review with grading on a company |
|  | OverallReview | A review with grading on a company’s general basis |
|  | SalaryAndBenefitReview | A review with grading on a company’s salary and benefit basis |
|  | Job | Reviewed professions within the CRW |
|  | Comment | User’s commentary on a review |
|  | Notification | A to-be-sent-note triggered by a pre-determined pool upon a required action |
|  | Report | A report upon a review due to its rule breaking nature or more |

**Controller (Server-side)**

1. Danh sách Controller thuộc server trong hệ thống

|  |  |  |
| --- | --- | --- |
| Số | Tên | Vai trò |
|  | admin | Quản lý quản trị viên và người dùng chính |
|  | users | Xử lý CRUD với người dùng thường |
|  | users.profile | Cập nhật thông tin về người dùng như ảnh đại diện, mô tả… |
|  | users.password | Thay đổi và lấy lại mật khẩu |
|  | users.authorization | Lấy thông tin người dùng từ cơ sở dữ liệu dựa theo ID |
|  | users.authentication | Xử lý đăng ký và đăng nhập cho người dùng |
|  | core | Xử lý template trang chủ và thông báo lỗi chung |
|  | errors | Xử lý thông báo lỗi từ máy chủ |
|  | companies | Xử lý CRUD với công ty, đánh giá và bình luận |
|  | jobs | Xử lý CRUD với nghề nghiệp |
|  | chat-config | Gửi thông báo từ server về client khi có sự kiện mới |

**Controller (Client-side)**

1. Danh sách Controller thuộc Client trong hệ thống

|  |  |  |
| --- | --- | --- |
| Số | Tên | Vai trò |
|  | authentication | Nhận thông tin đăng nhập/đăng ký và gửi về server |
|  | password | Gửi yêu cầu và xác nhận khi muốn lấy lại mật khẩu |
|  | user | Xử lý thông tin giữa server và view-user, edit-user |
|  | list-users | Xử lý thông tin giữa server và list-users |
|  | change-password | Xử lý thông tin giữa server và change-password |
|  | change-profile-picture | Xử lý thông tin giữa server và change-profile-picture |
|  | edit-profile | Xử lý thông tin giữa server và edit-profile |
|  | manage-social-accounts | Xử lý thông tin giữa server và manage-social-accounts |
|  | settings | Xử lý thông tin giữa server và settings |
|  | home | Xử lý thông tin giữa server và home |
|  | header | Xử lý thông tin giữa server và header |
|  | company | Xử lý thông tin giữa server và view-company, form-company |
|  | list-companies | Xử lý thông tin giữa server và list-companies |
|  | overall-review | Xử lý thông tin giữa server và view-overall-review, form-overall-review |
|  | list-overall-reviews | Xử lý thông tin giữa server và list-overall-reviews |
|  | salary-review | Xử lý thông tin giữa server và view-salary-review, form-salary-review |
|  | list-salary-reviews | Xử lý thông tin giữa server và list-salary-reviews |
|  | benefit-review | Xử lý thông tin giữa server và view-benefit-review, form-benefit-review |
|  | list-benefit-reviews | Xử lý thông tin giữa server và list-benefit-reviews |
|  | job | Xử lý thông tin giữa server và view-job, form-job |
|  | list-jobs | Xử lý thông tin giữa server và list-jobs |
|  | comment | Xử lý thông tin giữa server và view-comment, form-comment |
|  | list-comments | Xử lý thông tin giữa server và list-comments |
|  | message | Xử lý thông tin giữa server và view-message, form-message |
|  | list-messages | Xử lý thông tin giữa server và view list-messages |
|  | follow | Xử lý thông tin giữa server và view-follow |
|  | report | Xử lý thông tin giữa server và view-report, form-report |
|  | list-reports | Xử lý thông tin giữa server và view list-reports |

**View**

1. Danh sách View trong hệ thống

|  |  |  |
| --- | --- | --- |
| Số | Tên | Vai trò |
|  | layout | Template chứa bố cục cơ bản của trang web |
|  | index | Template kết nối bố cục với Angular phía client |
|  | 500 | Hiển thị màn hình báo lỗi server |
|  | 400 | Hiển thị màn hình báo lỗi HTTP request |
|  | 403 | Hiển thị màn hình báo lỗi không đủ quyền hạn truy cập |
|  | 404 | Hiển thị màn hình báo lỗi không tìm thấy trang cần tìm |
|  | authentication | Đăng nhập hệ thống với tài khoản mạng xã hội |
|  | signin | Đăng nhập hệ thống với email và mật khẩu |
|  | signup | Đăng ký tài khoản với hệ thống |
|  | view-user | Hiển thị thông tin user |
|  | list-users | Hiển thị danh sách user |
|  | edit-user | Thêm hoặc sửa thông tin user |
|  | forgot-password | Hiển thị màn hình lấy lại mật khẩu |
|  | reset-password | Hiển thị màn hình cập nhật mật khẩu sau khi lấy lại |
|  | reset-password-invalid | Thông báo mật khẩu cập nhật không hợp lệ |
|  | reset-password-success | Thông báo mật khẩu cập nhật hợp lệ |
|  | change-password | Cập nhật mật khẩu tài khoản |
|  | change-profile-picture | Thay đổi ảnh đại diện |
|  | edit-profile | Sửa thông tin tài khoản |
|  | manage-social-accounts | Quản lý các tài khoản mạng xã hội liên kết với tài khoản |
|  | settings | Thay đổi các thiết lập của tài khoản |
|  | home | Trang chủ của hệ thống |
|  | header | Thanh điều hướng của hệ thống |
|  | form-company | Thêm hoặc sửa thông tin về một công ty |
|  | list-companies | Hiển thị danh sách các công ty |
|  | view-company | Xem hoặc xóa một công ty |
|  | form-overall-review | Thêm hoặc sửa thông tin về đánh giá tổng thể |
|  | list-overall-reviews | Hiển thị danh sách các đánh giá tổng thể |
|  | view-overall-review | Xem hoặc xóa một bài đánh giá tổng thể |
|  | form-salary-review | Thêm hoặc sửa thông tin về đánh giá lương của một công ty |
|  | list-salary-reviews | Hiển thị danh sách các đánh giá lương |
|  | view-salary-review | Xem hoặc xóa một bài đánh giá lương |
|  | form-benefit-review | Thêm hoặc sửa thông tin về đánh giá chế độ đãi ngộ của một công ty |
|  | list-benefit-reviews | Hiển thị danh sách các đánh giá chế độ đãi ngộ |
|  | view-benefit-review | Xem hoặc xóa một bài đánh giá chế độ đãi ngộ |
|  | form-job | Thêm hoặc sửa thông tin về một nghề |
|  | list-jobs | Hiển thị danh sách các nghề |
|  | view-job | Xem hoặc xóa một nghề |
|  | form-comment | Thêm hoặc sửa thông tin về một bình luận |
|  | list-comments | Hiển thị danh sách các bình luận |
|  | view-comment | Xem hoặc xóa một bình luận |
|  | form-message | Thêm hoặc sửa thông tin về một thông báo |
|  | list-messages | Hiển thị danh sách các thông báo |
|  | view-message | Xem hoặc xóa một thông báo |
|  | view-follow | Xem hoặc xóa công ty trong danh sách theo dõi |
|  | form-report | Thêm hoặc sửa thông tin về báo cáo vi phạm |
|  | list-reports | Hiển thị danh sách các báo cáo vi phạm |
|  | view-report | Xem hoặc xóa một báo cáo vi phạm |

## 

## Class diagram

\* Chú thích: Javascript không có class theo cách hiểu chính thống, dưới đây là diagram thực thể tương tự class trong đồ án. Xem thêm tại <http://raganwald.com/2014/01/19/prototypes-are-not-classes.html>

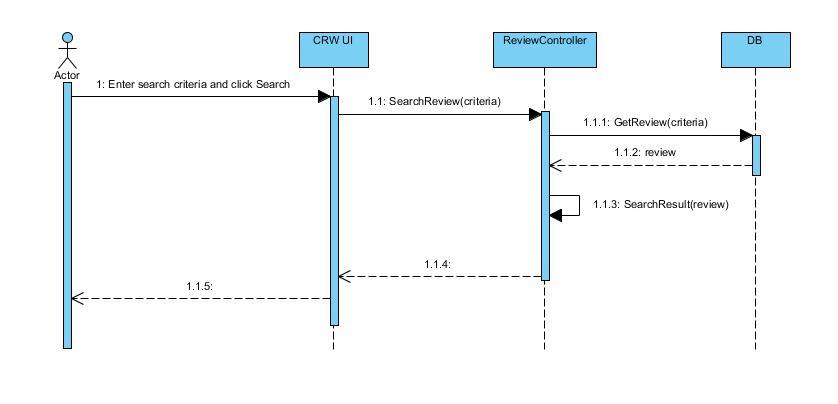
<Picture here>

1. Class Diagram

## Sequence diagram ứng với use case

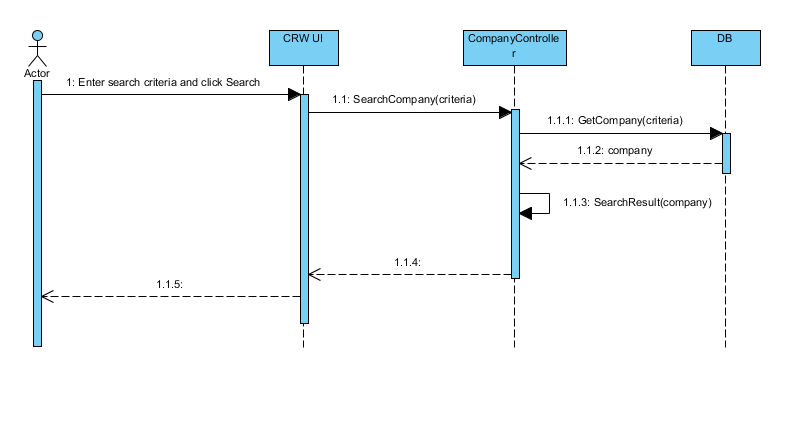
Thể hiện một chuỗi các tương tác và sự kiện diễn ra trong mỗi use case giữa người dùng và các thành phần trong hệ thống. Đối chiếu với Software Requirements Specifications, phần “Tính năng của hệ thống” và “Use case”.

**Use case:** Tìm kiếm đánh giá



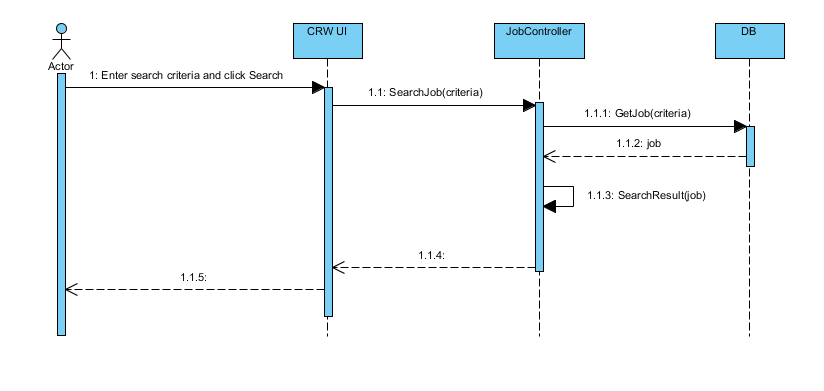
1. Sequence Diagram

**Use case:** Tìm kiếm công ty



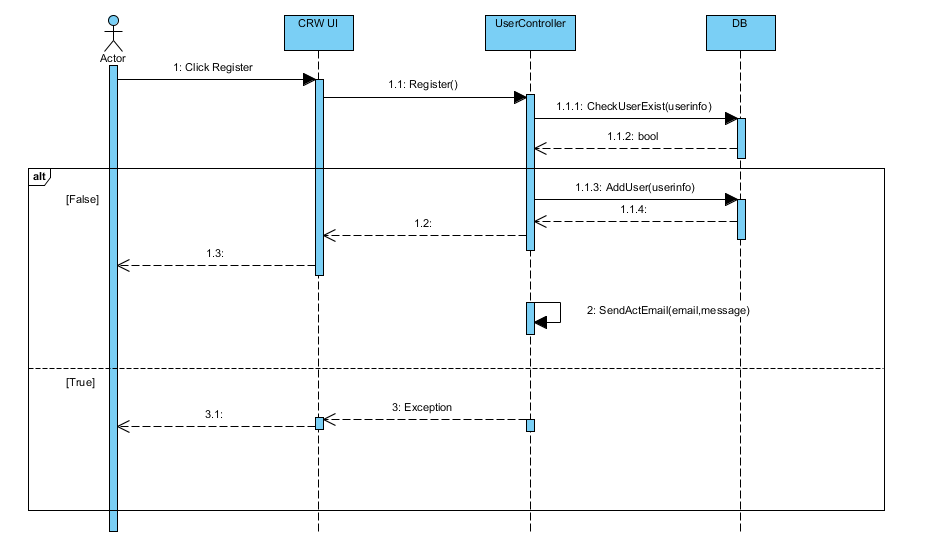
1. Sequence Diagram

**Use case:** Tìm kiếm nghề nghiệp



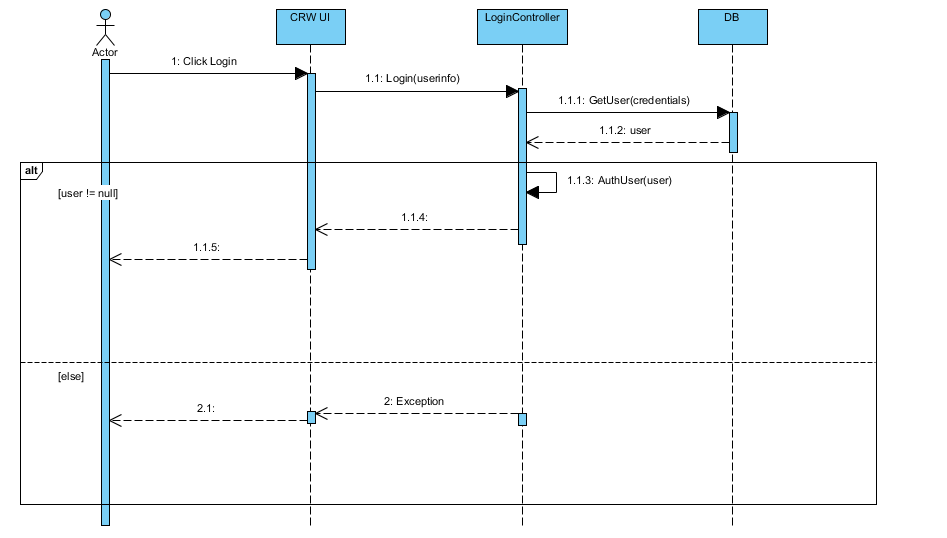
1. Sequence Diagram

**Use case:** Đăng ký tài khoản



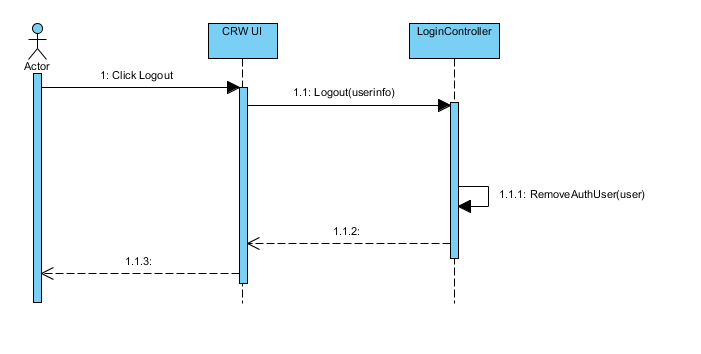
1. Sequence Diagram

**Use case:** Đăng nhập



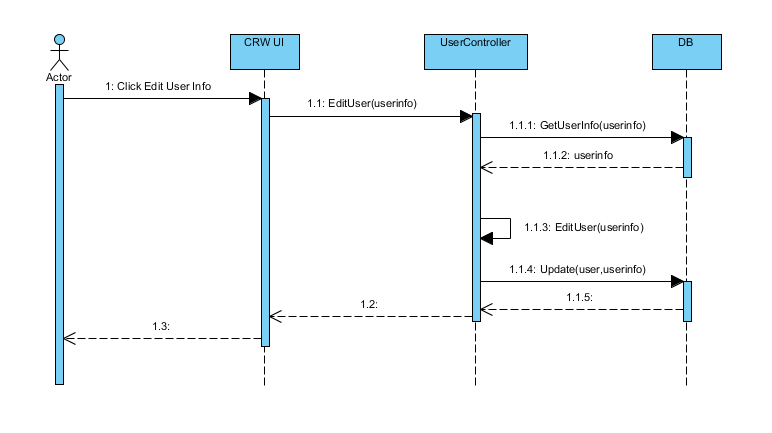
1. Sequence Diagram

**Use case:** Đăng xuất



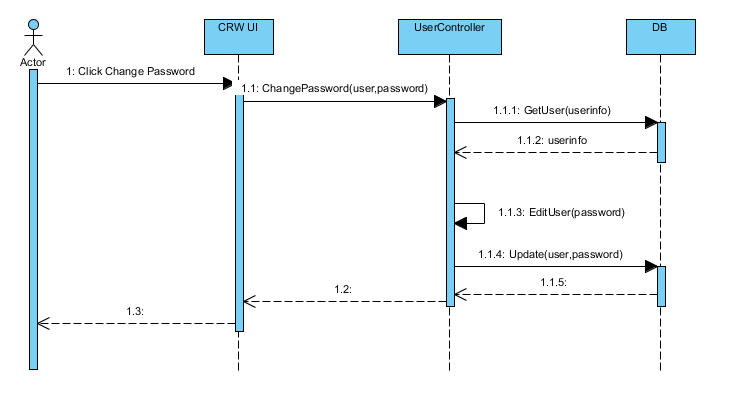
1. Sequence Diagram

**Use case:** Cập nhật thông tin tài khoản



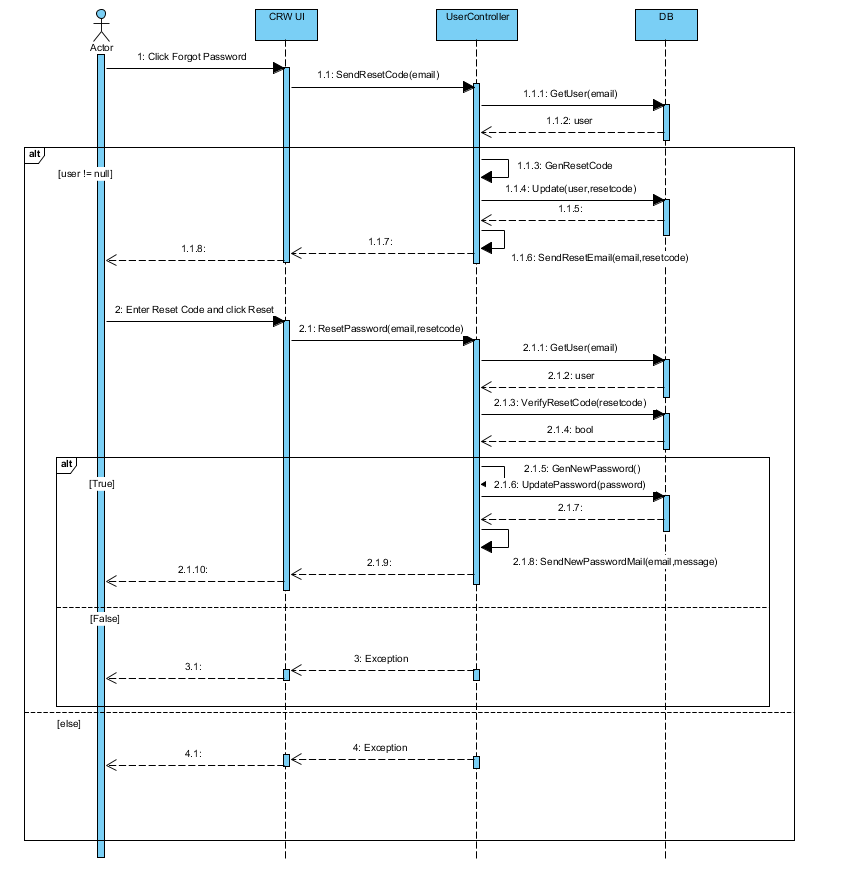
1. Sequence Diagram

**Use case:** Đổi mật khẩu



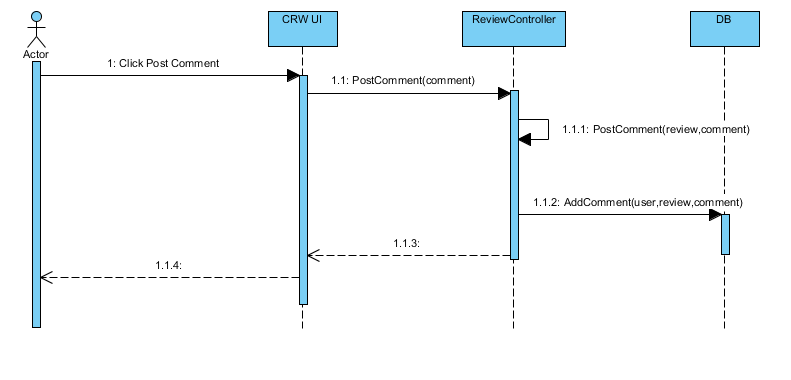
1. Sequence Diagram

**Use case:** Quên mật khẩu



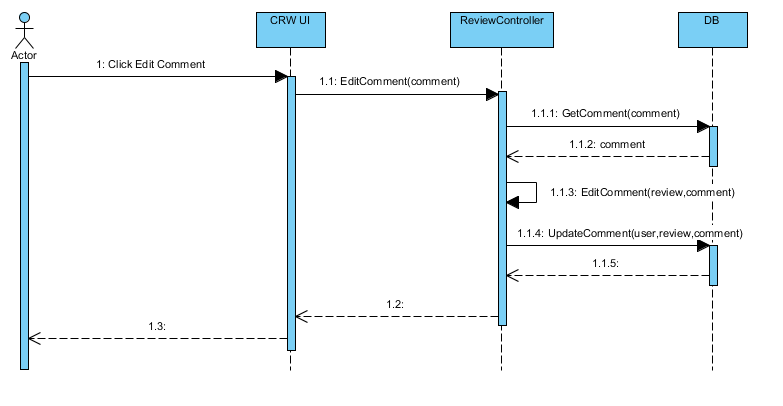
1. Sequence Diagram

**Use case:** Thêm bình luận

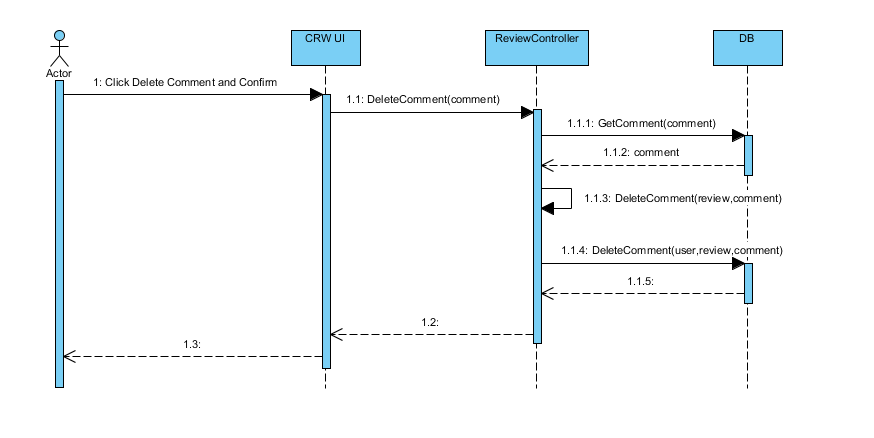


1. Sequence Diagram

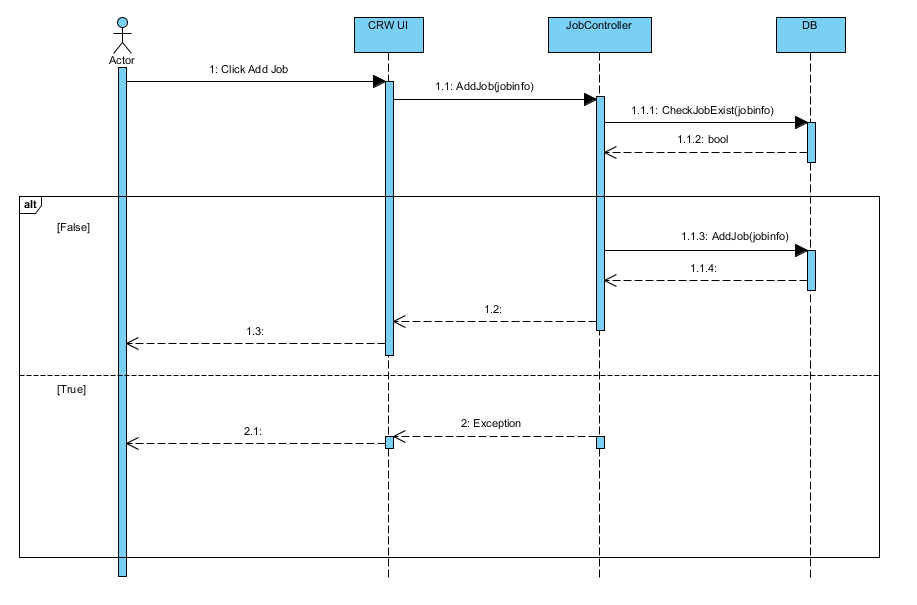
**Use case:** Cập nhật bình luận



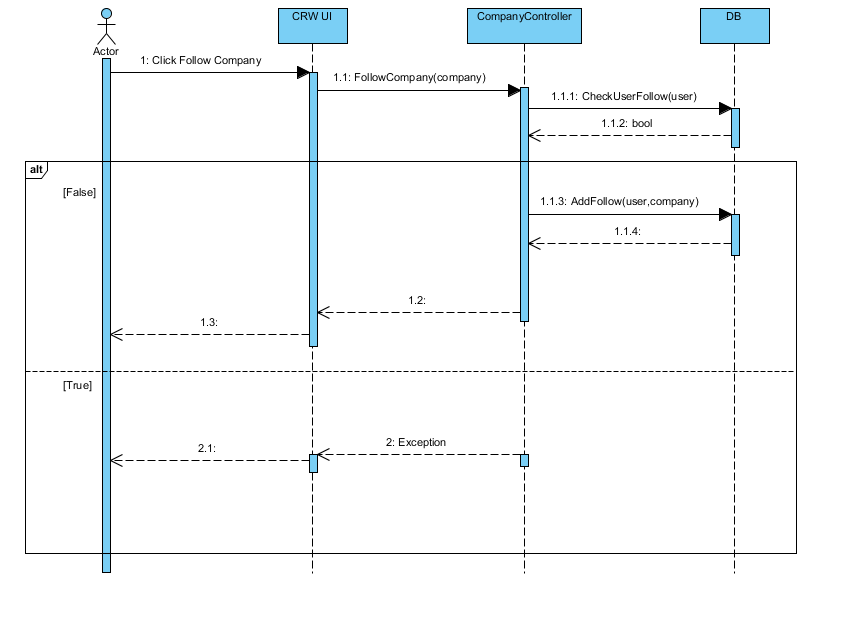
1. Sequence Diagram

**Use case:** Xóa bình luận

1. Sequence Diagram

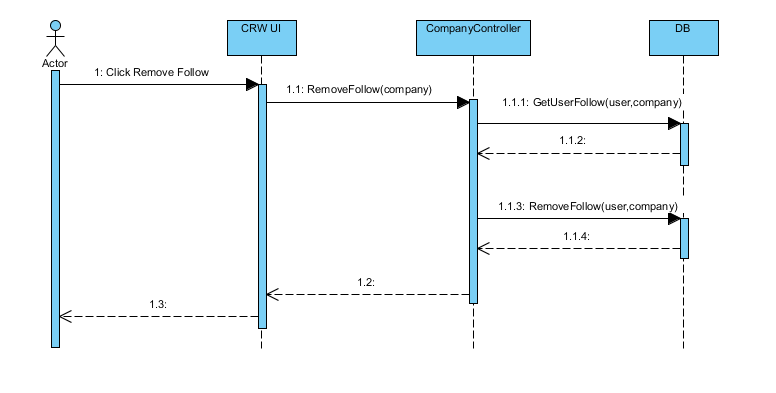
**Use case:** Thêm nghề nghiệp

1. Sequence Diagram

**Use case:** Theo dõi công ty

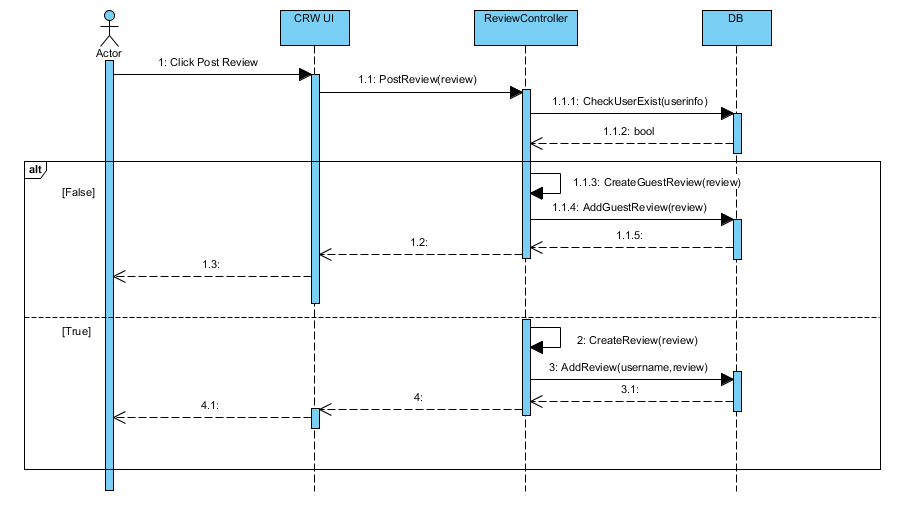
1. Sequence Diagram

**Use case:** Bỏ theo dõi công ty



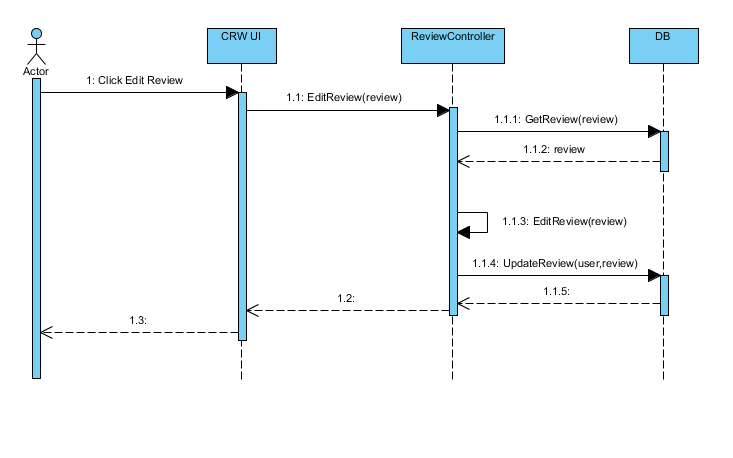
1. Sequence Diagram

**Use case:** Đăng đánh giá



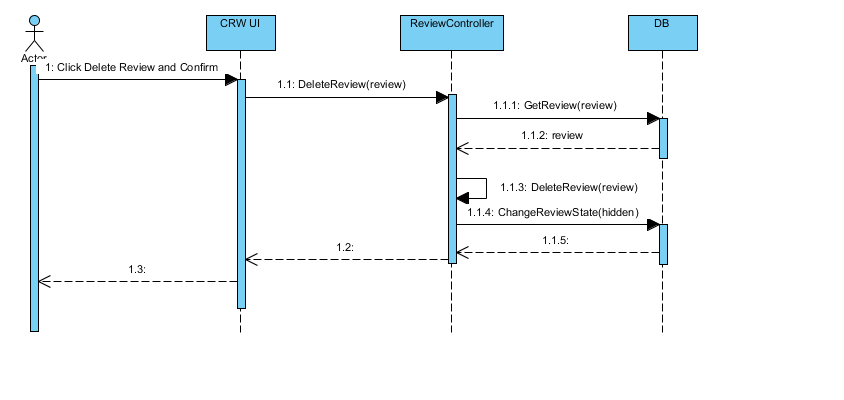
1. Sequence Diagram

**Use case:** Cập nhật đánh giá



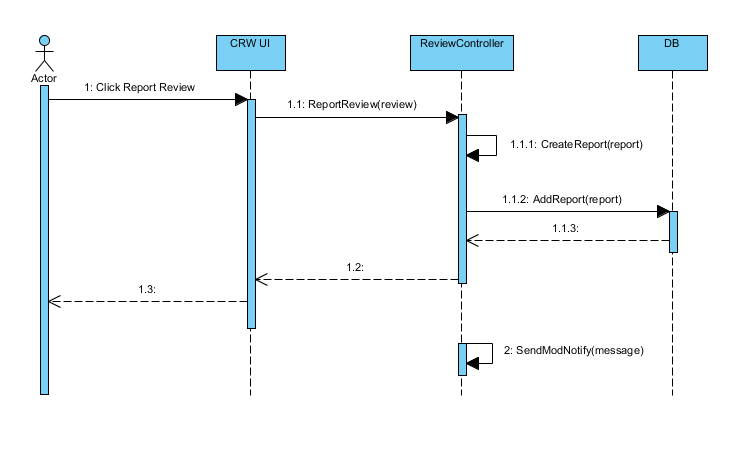
1. Sequence Diagram

**Use case:** Xóa đánh giá



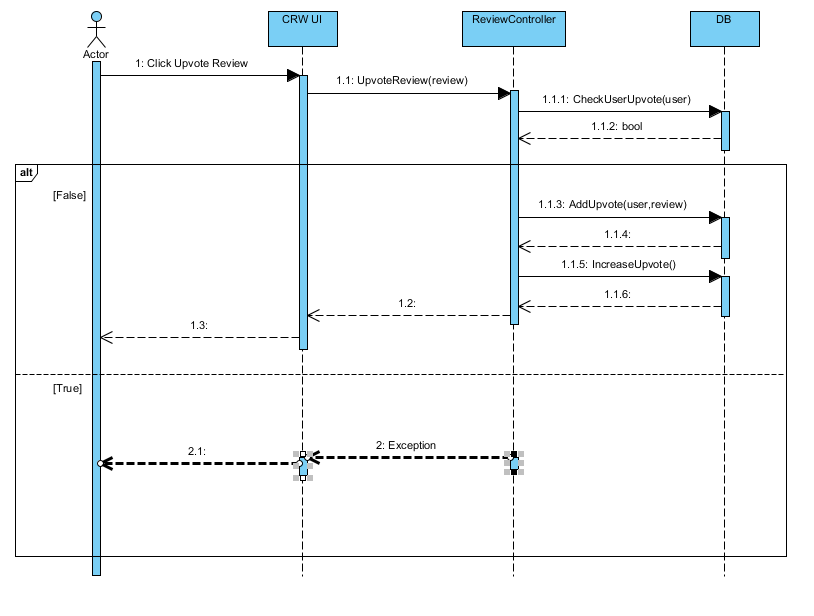
1. Sequence Diagram

**Use case:** Báo cáo đánh giá vi phạm



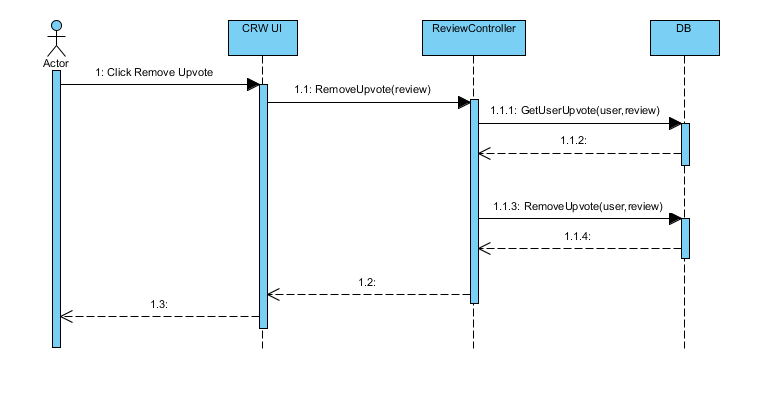
1. Sequence Diagram

**Use case:** Ủng hộ đánh giá



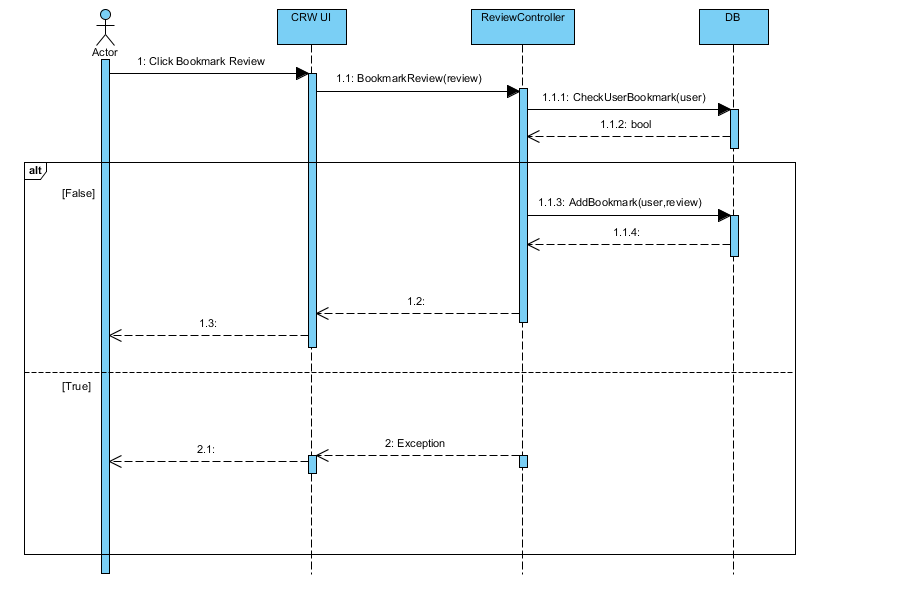
1. Sequence Diagram

**Use case:** Bỏ ủng hộ đánh giá



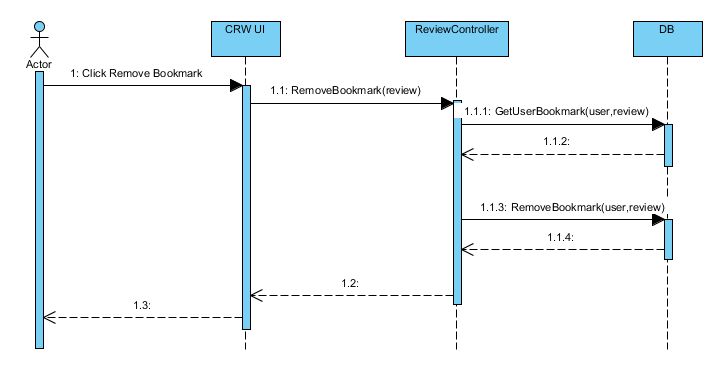
1. Sequence Diagram

**Use case:** Quản lý dấu trang



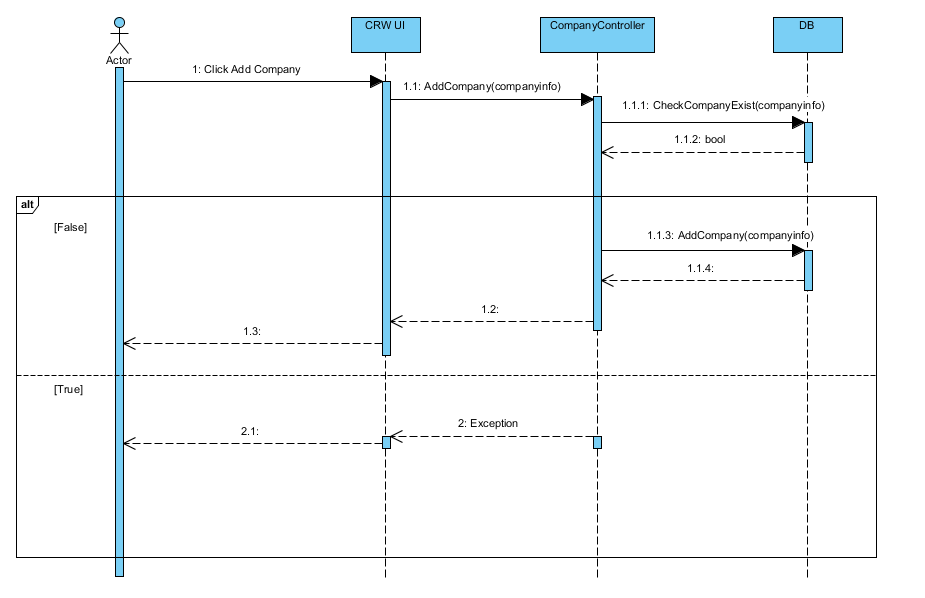
1. Sequence Diagram

**Use case:** Xóa dấu trang



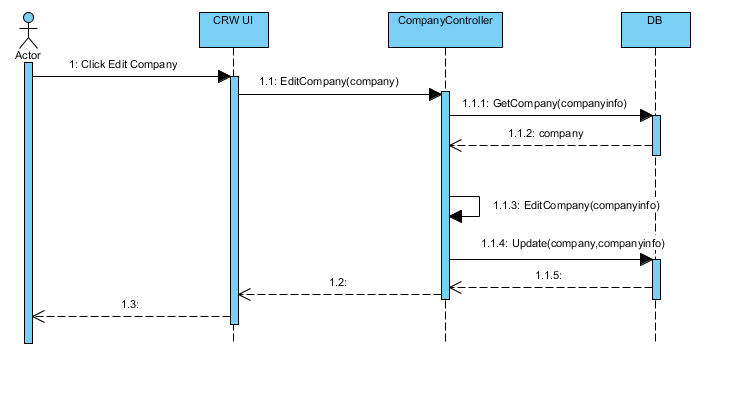
1. Sequence Diagram

**Use case:** Thêm công ty



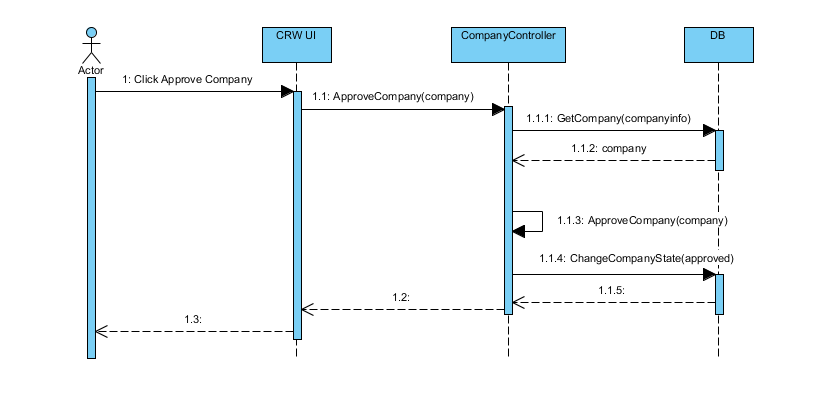
1. Sequence Diagram

**Use case:** Cập nhật công ty



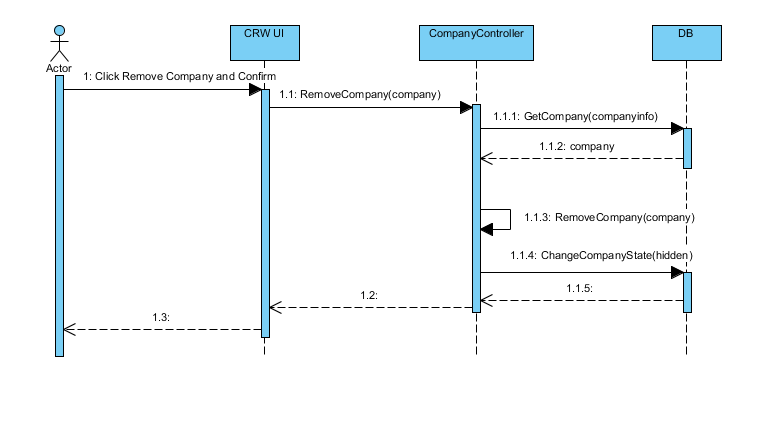
1. Sequence Diagram

**Use case:** Duyệt công ty



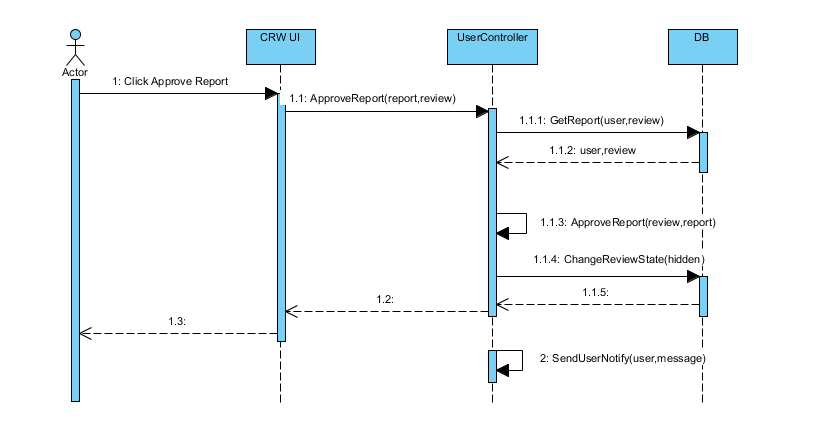
1. Sequence Diagram

**Use case:** Xóa công ty



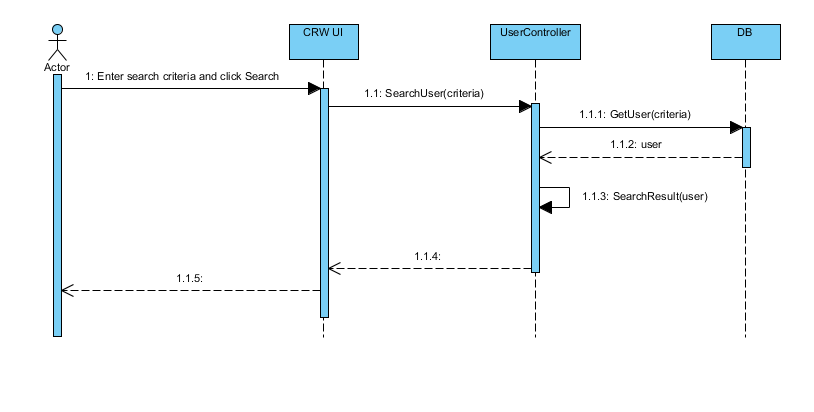
1. Sequence Diagram

**Use case:** Duyệt vi phạm



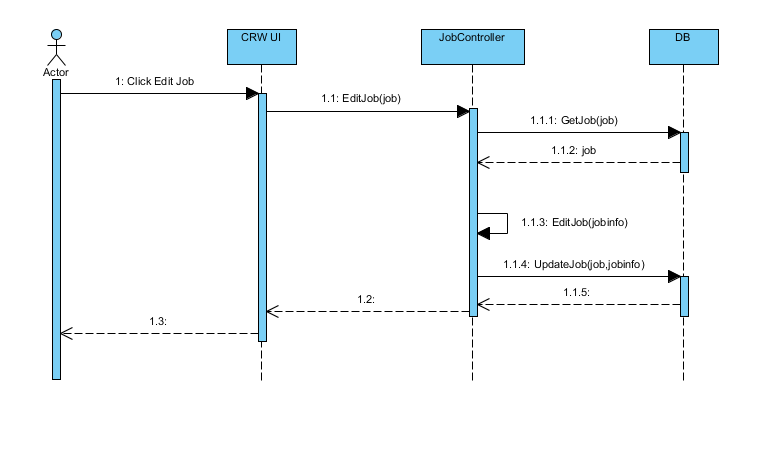
1. Sequence Diagram

**Use case:** Tìm kiếm tài khoản



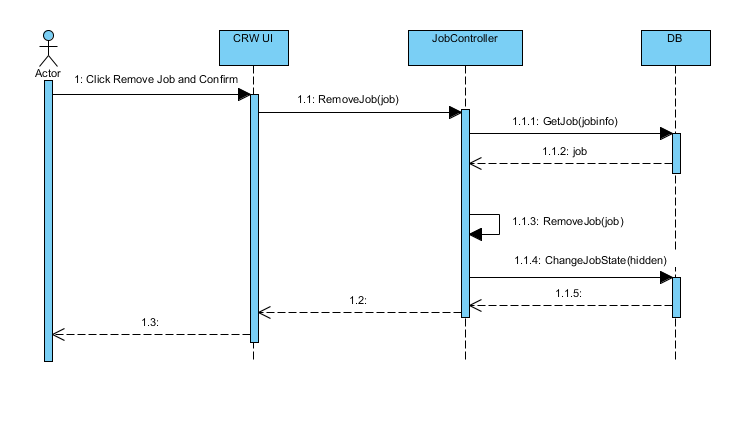
1. Sequence Diagram

**Use case:** Cập nhật tên nghề



1. Sequence Diagram

**Use case:** Xóa nghề nghiệp



1. Sequence Diagram

# DEPLOYMENT VIEW



1. Mô hình triển khai hệ thống
2. Các thành phần của hệ thống theo Deployment view

|  |  |  |
| --- | --- | --- |
| Số | Tên | Mô tả |
|  | Web Server | Server chứa các thành phần chính của trang web |
|  | Client | Máy khách truy cập và sử dụng dịch vụ từ hệ thống |
|  | Facebook | Facebook API Server |
|  | Google | Google API Server |
|  | Database | Chứa dữ liệu hệ thống. Sử dụng MongoDB. |