**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**KHOÁ LUẬN TỐT NGHIỆP**

**Tìm hiểu nghiệp vụ quản lý tìm việc online**

**Xây dựng Website Tìm Việc**



**Giáo viên hướng dẫn : Th.S Nguyễn Minh Đạo**

**Sinh viên thực hiện : Lê Quang Sơn - 15110118**

**Nguyễn Xuân Tuấn - 15110145**

*TP.HCM, tháng 07 năm 2019*

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**KHOÁ LUẬN TỐT NGHIỆP**

**Tìm hiểu nghiệp vụ quản lý tìm việc online**

**Xây dựng Website Tìm Việc**



**Giáo viên hướng dẫn : Th.S Nguyễn Minh Đạo**

**Sinh viên thực hiện : Lê Quang Sơn - 15110118**

**Nguyễn Xuân Tuấn - 15110145**

*TP.HCM, tháng 07 năm 2019*

**Nhận xét của giáo viên hướng dẫn**

Giáo viên hướng dẫn

**Nhận xét của giáo viên phản biện**

Giáo viên phản biện

**LỜI CẢM ƠN**

Được sự phân công của quý thầy cô khoa Công nghệ thông tin, Trường Đại Học Sư Phạm Kỹ Thuật Thành phố Hồ Chí Minh, sau gần hết một học kỳ nhóm em đã hoàn thành môn học “Đồ án tốt nghiệp”.

Để hoàn thành nhiệm vụ được giao, ngoài sự tìm hiểu của bản thân mỗi người trong nhóm còn có sự hướng dẫn tận tình của thầy cô.

Em chân thành cảm ơn Thầy giáo – Nguyễn Minh Đạo, người đã hướng dẫn cho nhóm em trong suốt thời gian làm đồ án. Mặc dù thầy rất bận, kẹt thời gian rất nhiều nhưng không ngần ngại chỉ dẫn, định hướng đi cho nhóm em, để nhóm em hoàn thành tốt nhiệm vụ. Một lần nữa em chân thành cảm ơn thầy và chúc thầy dồi dào sức khoẻ.

Tuy nhiên vì kiến thức chuyên môn còn hạn chế và bản thân còn thiếu nhiều kinh nghiệm thực tiễn nên nội dung của báo cáo không tránh khỏi những thiếu xót, em rất mong nhận sự góp ý, chỉ bảo thêm của quý thầy cô hướng dẫn và thầy cô phản biện để báo cáo này được hoàn thiện hơn.

Một lần nữa xin gửi đến thầy cô lời cảm ơn chân thành và tốt đẹp nhất!

Sinh viên thực hiện

Lê Quang Sơn

Nguyễn Xuân Tuấn

**LỜI NÓI ĐẦU**

Công nghệ ngày càng phát triển, đặc biệt là trong kỷ nguyên 4.0 như ngày nay thì công nghệ là thứ không thể thiếu trong đời sống chúng ta. Với công nghệ ta có thể dùng thông tin để xử lý, rút ngắn thời gian các công việc của chứng ta, giúp ta làm được nhiều việc hơn trong thời gian ngắn hơn. Ngoài ra, ứng dụng của công nghệ để giúp chúng ta tập hợp và tìm kiếm thông tin một cách nhanh chóng. Điều đó được thể hiện đặc biệt trong việc tìm kiếm thông tin việc làm, tuyển dụng.

Vì thế, người thực hiện đề tài đã nghiên cứu và phát triển một website tìm việc làm nhằm đáp ứng được nhu cầu tìm kiếm việc một cách nhanh chóng như hiện nay.

MỤC LỤC

[Chương I. Tổng quan 3](#_Toc11676148)

[1. Đặt vấn đề 3](#_Toc11676149)

[2. Ý nghĩa thực tiễn 3](#_Toc11676150)

[3. Mục tiêu và nhiệm vụ của đề tài 3](#_Toc11676151)

[3.1. Mục tiêu 3](#_Toc11676152)

[3.2. Nhiệm vụ 3](#_Toc11676153)

[Chương II. Cơ sở lý thuyết 4](#_Toc11676154)

[1. Kiến thức cơ bản 4](#_Toc11676155)

[2. Mô hình MVC 4](#_Toc11676156)

[2.1. Khái niệm mô hình MVC 4](#_Toc11676157)

[2.2. Ưu điểm và nhược điểm của mô hình MVC 4](#_Toc11676158)

[3. Spring Boot 4](#_Toc11676159)

[3.1. Khái niệm về Spring Boot 4](#_Toc11676160)

[3.2. Ưu điểm và nhược điểm của Spring Boot 4](#_Toc11676161)

[4. ReactJS 4](#_Toc11676162)

[4.1. Sad 4](#_Toc11676163)

[4.2. Fsdf 4](#_Toc11676164)

[4.3. Ds 4](#_Toc11676165)

[5. Restful API 4](#_Toc11676166)

[5.1. Giới thiệu 4](#_Toc11676167)

[5.2. Restful, Restful API là gì? 4](#_Toc11676168)

[5.3. Kiến trúc của Restful 4](#_Toc11676169)

[5.4. Ưu điểm và nhược điểm cảu Restful 4](#_Toc11676170)

[6. Github 4](#_Toc11676171)

[6.1. Github là gì ? 4](#_Toc11676172)

[6.2. Tại sao nên sử dụng Git? 4](#_Toc11676173)

[Chương III. Phân tích thiết kế hệ thống 5](#_Toc11676174)

[1. Xác định yêu cầu, khảo sát nghiệp vụ 5](#_Toc11676175)

[1.1. Xác định yêu cầu 5](#_Toc11676176)

[1.2. Khảo sát nghiệp vụ 5](#_Toc11676177)

[2. Xác định các Actor và Usecase 5](#_Toc11676178)

[2.1. Các tác nhân tham gia vào hệ thống 5](#_Toc11676179)

[2.2. Biểu đồ các Actor và Usecase 5](#_Toc11676180)

[2.3. Đặc tả Usecase 5](#_Toc11676181)

[3. Biểu đồ tuần tự 5](#_Toc11676182)

[3.1. Biểu đồ tuần tự dành cho người tìm việc 5](#_Toc11676183)

[3.2. Biểu đồ tuần tự dành cho nhà tuyển dụng 5](#_Toc11676184)

[3.3. Biểu đồ tuần tự dành cho admin 5](#_Toc11676185)

[4. Biểu đồ lớp 5](#_Toc11676186)

[5. Biểu đồ hoạt động 5](#_Toc11676187)

[6. Biểu đồ thực thể liên kết 5](#_Toc11676188)

[7. Thiết kế cơ sở dữ liệu 5](#_Toc11676189)

[8. Thiết kế giao diện 5](#_Toc11676190)

[Chương IV. Cài đặt và kiểm thử 6](#_Toc11676191)

[1. Cài đặt 6](#_Toc11676192)

[2. Kiểm thử 6](#_Toc11676193)

[Chương V. Tổng kết 7](#_Toc11676194)

[1. Kết quả đạt được 7](#_Toc11676195)

[2. Hạn chế 7](#_Toc11676196)

[3. Hướng phát triển 7](#_Toc11676197)

[TÀI LIỆU THAM KHẢO 8](#_Toc11676198)

**DANH MỤC HÌNH VẼ**

# Chương I. Tổng quan

1. **Đặt vấn đề**

Hiện nay, nền công nghiệp 4.0 đang ngày càng phát triển, thúc đẩy các hoạt động phát triển theo, đặc biệt là các thông tin, tin tức về các ngành nghề, lĩnh vực. Thông tin về nhu cầu công việc ngày càng nhiều, làm thế nào để thông tin dễ dàng đến với mọi người, dễ dàng tìm kiếm, trao đổi và thương lượng các công việc. Nhu cầu tìm kiếm công việc hay các công ty cần tìm đối tác đang tăng cao vì vậy một website đang tuyển việc làm là rất cần thiết để các công ty cũng như đối tác dễ dàng tìm kiếm và trao đổi mang lại hiệu quả công việc cao, tiết kiệm chi phí.

Với nhu cầu tìm kiếm việc làm đang tăng cao làm sao để một cá nhân, công ty muốn tìm những việc làm phù hợp hay lưu lại thông tin công ty mà mình quan tâm, để việc quản lý các thông tin đó trở nên dễ dàng tiện lợi nhất. Ứng tuyển công việc từ xa, chào giá công việc trở nên thuận lợi và tiết kiệm nhất.

Các công ty đăng tin tuyển dụng dễ dàng quản lý được công việc mình đã đăng tuyển, xem các hồ sơ ứng tuyển của ứng viên một cách nhanh chóng và thuận tiện. Việc quản lý lưu trữ hay quan tâm tới các ứng viên tiềm năng, có đủ năng lực phù hợp với yêu cầu của công ty trở nên dễ dàng. Thay vì nhận đơn ứng tuyển bằng các bộ hồ sơ thủ công tốn thời gian, không gian để lưu trữ mà hiệu quả công việc đem lại không cao đôi khi còn xảy ra sai sót, mất mát, thất lạc hồ sơ ứng viên. Hay việc quản lý hạn đăng tin tuyển dụng sẽ được tự động hóa.

Hiện nay có khá nhiều trang website tuyển dụng việc làm nhưng chủ yếu dành cho công ty, doanh nghiệp với cá nhân. Đối với các ngành nghề đặc thù như xây dựng, công trình… việc tìm các đối tác để đảm nhận một gói thầu trong toàn dự án là rất ít. Vì vậy với website tuyển dụng việc làm này sẽ chú trọng vào đối tượng là các công ty, nhà thầu có đấu giá các gói thầu trong toàn bộ công việc hoặc toàn bộ công việc. Website sẽ giúp các công ty, nhà thầu dễ dàng tìm được các đối tác kết hợp với mình để hoàn thành công việc hiệu quả.

1. **Ý nghĩa thực tiễn**

Trong quá trình khảo sát, nhóm chúng em nhận thấy hầu hết các trang web tuyển dụng việc làm chủ yếu tập trung vào các ngành nghề như Công nghệ thông tin, kinh doanh… rất ít trang web làm đến các nhà thầu tìm kiếm các đối tác để chia sẻ các gói thầu trong toàn bộ dự án. Vì vậy các nhà thầu sẽ cần đến một trang web để có thể đăng tải các gói thầu mà mình đang cần tìm đối tác, đồng thời có thể đánh giá được các đối tác qua mỗi lần hợp tác.

Việc phát triển website đăng tin tuyển dụng việc làm sẽ giúp các nhàu thầu, các cá nhân, doanh nghiệp:

* Dễ dàng quản lý công việc, đối tác.
* Dễ dàng đánh giá đối tác qua các lần hợp tác.
* Dễ dàng tìm kiếm, quan tâm đến đối tác, công việc phù hợp.
* Tạo nên một cộng đồng chia sẻ, cùng hợp tác phát triển kinh tế.

Do vậy, nhóm chúng em hy vọng sẽ tạo nên một website hoàn chỉnh nhất có thể để đáp ứng được yêu cầu đặt ra cũng như kỳ vọng,và sự nhiệt tình giúp đỡ từ thầy cô.

1. **Mục tiêu và nhiệm vụ của đề tài**
   1. **Mục tiêu**

Ứng dụng công nghệ mới như ReactJs, Restful, Spring boot vào quy trình phát triển một website tìm việc online toàn diện.

* 1. **Nhiệm vụ**

Tìm hiểu nghiệp vụ quản lý tìm việc online tựa trang web: <https://www.timviecnhanh.com/>

Nghiên cứu công nghệ mới: Spring boot, ReactJs, Restful Webservice.

Phát triển ứng dụng: Website tìm kiếm việc làm.

1. **Phương pháp nghiên cứu**

Đối với client: Hiện thực Front – end bằng ReactJS.

Đối với Server: Xây dựng các service và API thông qua Spring Boot.

Database: Tạo database ảo trên Spring Boot.

# Chương II. Cơ sở lý thuyết

1. **Kiến thức cơ bản**

Java (phiên âm Tiếng Việt: "Gia-va") là một ngôn ngữ lập trình hướng đối tượng (OOP) và dựa trên các lớp (class). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.

Có 6 mục tiêu chính trong việc xây dựng ngôn ngữ Java:

* Đơn giản, hướng đối tượng và quen thuộc.
* Mạnh mẽ và an toàn.
* Kiến trúc trung lập và di động.
* Thực thi với hiệu suất cao.
* Dịch ra bytecode, phân luồng và năng động.
* Dễ sử dụng cho người dùng Java.

1. **Mô hình MVC**
   1. **Khái niệm mô hình MVC**

A close up of a map

Description generated with high confidence

Hình II.2.1 Mô tả về mô hình MVC

Mô hình MVC là chữ viết tắt của Model – View – Controller, đây là một mô hình kiến trúc phần mềm được tạo ra với mục đích quản lý và xây dựng dự án phần mềm có hệ thống hơn. Mô hình này được dùng khá rộng rãi và đặc biệt là trong các ngôn ngữ lập trình web.

Trong mô hình này thì:

Model: có nhiệm vụ thao tác với cơ sở dữ liệu, nghĩa là nó sẽ chứa tất cả các hàm, các phương thức ttruy vấn trực tiếp với dữ liệu và controller sẽ thông qua hàm, phương thức đó để lấy dữ liệu rồi gửi qua view.

View: có nhiệm vụ tiếp nhận dữ liệu từ controller và hiển thị nội dung qua giao diện website.

Controller: đóng vai trò trung gian giữa model và view, có nhiệm vụ tiếp nhận yêu cầu từ client sau đó xử lý request, load model tương ứng và gửi data qua view tương ứng rồi trả kết quả về cho client.

* 1. **Ưu điểm và nhược điểm của mô hình MVC**

Ưu điểm:

* Hệ thống phân ra từng phần nên dễ dàng phát triển.
* Chia thành nhiều modun nhỏ nên nhiều người có thể làm chung dự án.
* Vấn đề bảo trì và nâng cấp dễ dàng hơn.
* Dễ dàng debug trong quá trình xây dựng.

Nhược điểm:

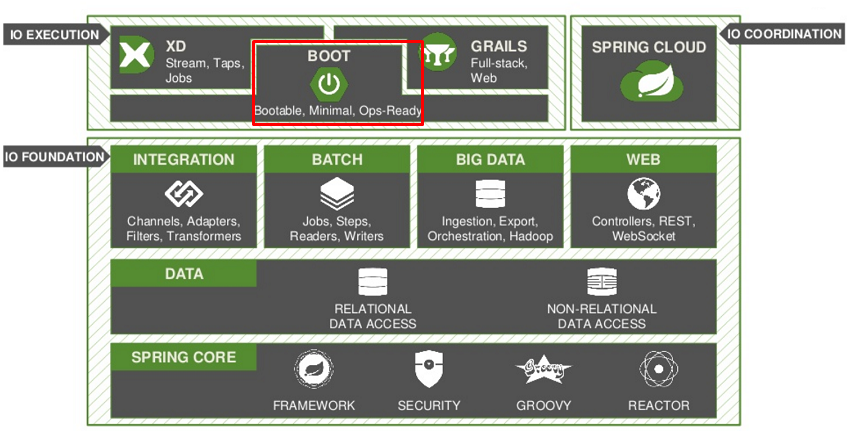
* Xây dựng cầu kỳ và mất thời gian để xây dựng các thư viện.

1. **Spring Boot**
   1. **Khái niệm về Spring Boot**

Spring Boot là một dự án nổi bật trong hệ sinh thái Spring Framework. Nếu như trước đây, công đoạn khởi tạo một dự án Spring khá vất vả từ việc khai báo các dependency trong file *pom.xml* cho đến cấu hình bằng XML hoặc annotation phức tạp, thì giờ đây với Spring Boot, chúng ta có thể tạo các ứng dụng Spring một cách nhanh chóng và cấu hình cũng đơn giản hơn.



Hình II.3.1: Spring Framework Runtime



Hình II.3.2: Spring Features

* 1. **Ưu điểm của Spring Boot**

Dễ dàng để phát triển các ứng dụng dựa trên Spring với Java hoặc Groovy.

Giảm thiểu thời gian phát triển và tăng năng xuất

Tránh việc phải viết nhiều mã nguyên mẫu (boilerplate Code), Annotations và các cấu hình XML.

Dễ dàng để bạn tương tác các ứng dụng Spring Boot với các hệ sinh thái của Spring như Spring JDBC, Spring ORM, Spring Data, Spring Security etc.

Đi theo cách tiếp cận “Nguyên tắc cấu hình mặc định” để giảm thiểu thời gian và nỗ lực cần thiết để phát triển ứng dụng.

Cung cấp các Server nhúng (Embedded HTTP servers) như là Tomcat, Jetty .... để phát triển và test các ứng dụng web nhanh chóng và dễ dàng.

Cung cấp công cụ CLI (Command Line Interface) dể phát triển và test các ứng dụng Spring Boot(Java hoặc Groovy) từ các dòng lệnh (command prompt) rất dễ dàng và nhanh chóng.

Cung cấp rất nhiều các plugin để phát triển và test các ứng dụng Spring Boot nhanh chóng sử dụng các công cụ Build như Maven và Gradle

Cung cấp nhiều plugin để làm việc với các cơ sở dữ liệu nhúng (embedded database) và các cơ sở dữ liệu lưu trữ trên bộ nhớ (in-memory Databases) một cách dễ dàng.

1. **ReactJS**
   1. **Giới thiệu**

Hiện nay các Web Dev đang nổi lên mô hình web single-page.

Web single-page là tất cả các thao tác xử lý của trang web đều được diễn ra trên cùng một trang. Việc này giúp nâng cao trải nghiệm người dùng, các thao tác xử lý của người dùng và animation,... đều được thực hiện một cách nhanh chóng và mượt mà.

So với web single-page thì các web cũ xử lý dữ liệu khá lâu do phải phụ thuộc khá nhiều vào khả năng xử lý của server nên thường xuyên load trang mới hay nhưng trang nhiều logic sẽ khiến một số chức năng không được như ý muốn và dễ sinh ra Bug làm giảm trải nghiệm người dùng.

Từ đó web single-page ra đời và các ngôn ngữ phát triển single-page ngày càng phát triển. Trong số đó ReactJS đang là một thư viện mạnh nhất về phát triển single-page hiện nay.

* 1. **Cơ bản về ReactJS**

ReactJS là một thư viện JavaScript dùng trong việc xây dựng giao diện người dùng và được phát triển bởi đội ngũ Facebook. Tới nay, ReactJS đã thu hút hàng triệu lập trình viên trên thế giới. ReactJS thường được dùng bởi Frontend Developer trong thiết kế Web Application một cách nhanh chóng hơn.

Một trong những điểm mạnh của React JS là virtual DOM - thứ nằm ẩn bên trong mỗi view và là lí do khiến cho React đạt được hiệu năng tốt. Khi một view yêu cầu gọi, tất cả mọi thứ sẽ được đưa vào trong một bản sao ảo của DOM. Sau khi việc gọi hoàn thành, React tiến hành một phép so sánh giữa DOM ảo và DOM thật, và thực hiện những thay đổi được chỉ ra trong phép so sánh trên.

* 1. **Component Life Cycle**

**ComponentWillMount:** đây là method sẽ được thực thi trước khi 1 component được render trên cả server side và client side.

**ComponentDidMount:** method này được thực thi khi 1 component được render trên client side. Đây là nơi các hàm AJAX requests, DOM or update state được thực thi. Method này cũng đucợ sử dụng để kết nối với các JS Framework khác và các function với delayed execution như setTimeout or setInterval.

**ComponentWillReceiveProps:** sẽ được thực thi ngay khi thuộc tính props được update và trước khi component được render lại.

**ShouldComponentUpdate:** sẽ trả về kết quả true or false. Phương thức này sẽ xác định 1 component có được update hay không. Mặc định giá trị này là true. Nếu bạn không muốn component render lại sau khi update state hay props thì return giá trị thành false. Các bạn xem ví dụ dưới để hiểu rõ.

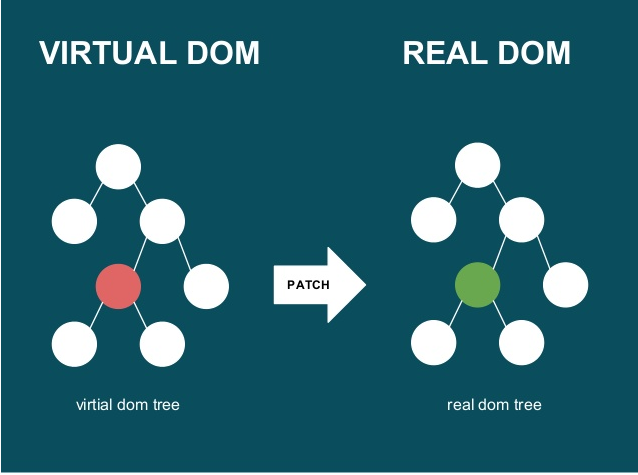
**ComponentWillUpdate:** được gọi khi chúng ta update state của component trước khi nó render lại.

**ComponentDidUpdate:** sau khi componentWillUpdate ở trên được gọi xong thì đến lượt thằng này được goi.

**ComponentWillUnmount:** được gọi khi chúng ta unmout 1 component kiểu như xóa nó khỏi react.

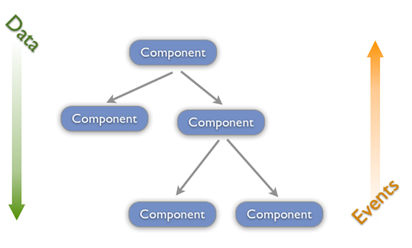
* 1. **Virtual-DOM**

Công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này.Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM Tree thật.



Hình 4: Vitual Dom

React sử dụng cơ chế one-way data binding – luồng dữ liệu 1 chiều. Dữ liệu được truyền từ parent đến child thông qua props. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi.



Hình 5: React Component Tree

Với các đặc điểm ở trên, React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện. Ví dụ như Facebook: trên Newsfeed của bạn cùng lúc sẽ có các status khác nhau và mỗi status lại có số like, share, comment liên tục thay đổi. Khi đó React sẽ rất hữu ích để sử dụng.

* 1. **JSX**

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javacsript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Safer: an toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt. Easier: Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng.

* 1. **Component**

React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Bạn có thể tạo ra một component bằng các gọi phương thức createClass của đối tượng React, điểm bắt đầu khi tiếp cận với thư viện này.

|  |
| --- |
| var Button = React.createClass({  render: function(){  return (  <input type="submit" />  );  }  }); |

Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác.

ReacJS tạo ra thẻ HTML không như chúng ta viết bình thường mà nó dùng Component để gói các thẻ HTML thành các object phân tầng để render

Trong các Component của React thì hàm render là quan trọng nhất. Nó chính là hàm xử lý việc tạo thẻ HTML cũng như là một minh chứng cho khả năng xử lý qua Virtual-DOM. Mọi thay đổi của dữ liệu ở bất cứ thời điểm nào đều cũng sẽ được Virtual-DOM xử lý và update ngay tức thì.

* 1. **Props**

Prop là thì không bị kiểm soát bởi Component

Ngược với State thì Props lại giống phương thức public ta có thể sử dụng và thay đổi thông qua Component khác

|  |
| --- |
| var text = "Click the button";  var Form = React.createClass({  render: function(){  return (  <div>  <h3>{this.props.text}</h3>  <input type="submit" />  </div>  );  }  }); |

* 1. **State**

State là data private bị kiểm soát bởi Component

State chính là dữ liệu chỉ có thể thay đổi trong Component mà ta đã khai báo. Các Component khác không thể dử dụng cũng như thay đổi được(Nó cũng giống như phương thức private mà ta biết đối với Class)

State : với state bạn sẽ phải gọi hàm getInitialState() để sử dụng được và muốn thay đổi state thì ta gọi hàm this.setState().

|  |
| --- |
| getInitialState: function(){  return {  active: true  }  },  handleClick: function(){  this.setState({  active: !this.state.active  });  },  render: function(){  var buttonSwitch = this.state.active ? "On" : "Off";  return (  <input type="submit" onClick={this.handleClick} />  <p>{buttonSwitch}</p>  );  }  }); |

* 1. **Ưu điểm và nhược điểm của ReacJS**

Ưu điểm:

* Cập nhật những thay đổi dữ liệu nhanh
* React không phải là framework nên nó giảm tải các ràng buộc của các thư viện với nhau
* Dễ tiếp cận với những người có kiến thức về JS.

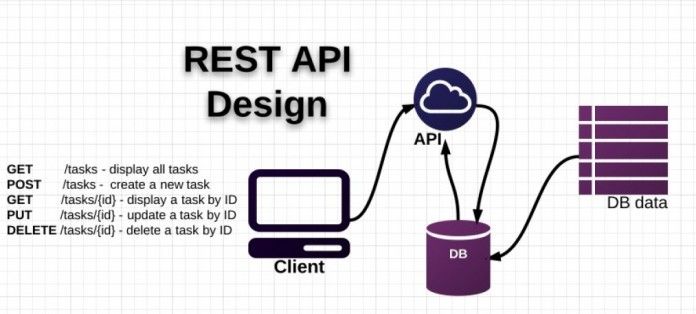
Nhược điểm:

* Reactjs chỉ phục vụ cho tầng View. nhưng dung lượng của thư viện lại ngang ngửa Angular trong khi Angular là một framework hoàn chỉnh.
* Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
* Khó tiếp cận cho người mới học về Web.

1. **Restful API**
   1. **Giới thiệu**

Trong thế giới kết nối như bây giờ, một sản phẩm không thể đứng độc lập, và sản phẩm nào không có APIs, giống như máy tính không được kết nối Internet vậy.

Việc xây dựng một API là một trong những điều quan trọng nhất bạn có thể làm để nâng cao giá trị dịch vụ của bạn. Bởi việc có một API thì dịch vụ hay ứng dụng của bạn mới có khả năng trở thành một nền tảng mà từ đó các dịch vụ khác sẽ phát triển theo. Hãy cùng xem những công ty hiện tại như: Facebook, Twitter, Google, GitHub, Amazon, Netflix…, sẽ không có một công ty nào có thể lớn mạnh như ngày nay nếu như họ không cung cấp mở những dữ liệu của họ thông qua API.



Hình 6: Rest API Design

* 1. **Restful, Restful API là gì?**

Rest là viết tắt của Representational State Transfer. Giải thích đơn giản, Rest là một loạt hướng dẫn và dạng cấu trúc dùng cho việc chuyển đổi dữ liệu. Thông thường, Rest hay được dùng cho ứng dụng web, nhưng cũng có thể làm việc được với dữ liệu phần mềm.

API là viết tắt của Application Programming Interface, phương thức kết nối với các thư viện và ứng dụng khác. Windows có nhiều API, và Twitter cũng có web API, tuy chúng thực hiện các chức năng khác nhau, với mục tiêu khác nhau.

Restful API là những API đi theo cấu trúc Rest.REST viết tắt cho REpresentational State Transfer Khái niệm về REST lần đầu tiên được giới thiệu vào năm 2000 trong luận văn Tiến sĩ của Roy Fielding (đồng sáng lập giao thức HTTP) Là 1 kiểu kiến trúc lập trình, định nghĩa các quy tắc để thiết kế web service chú trọng vào tài nguyên Mọi thứ trong REST đều được coi là tài nguyên và được định danh thông qua URI, và có thể được biểu diễn thông qua dạng văn bản, XML, JSON .... RESTful là những ứng dụng mà có sử dụng kiến trúc REST.

* 1. **Kiến trúc của Rest**
     1. *Client–server architecture*

Hệ thống hoạt động theo mô hình client-server, trong đó server là tập hợp các service nhỏ lắng nghe các request từ client. Với từng request khác nhau thì có thể một hoặc nhiều service xử lý.

* + 1. *Statelessness*

Stateless (phi trạng thái). Đơn giản server và client không lưu trạng thái của nhau, mỗi request lên server thì client phải đóng gói thông tin đầy đủ để thằng server hiểu được. Điều này giúp hệ thống của bạn dễ phát triển,bảo trì, mở rộng vì không cần tốn công CRUD trạng thái của client . Hệ thống phát triển theo hướng này có ưu điểm nhưng cũng có khuyết điểm là gia tăng lượng thông tin cần truyền tải giữa client và server.

* + 1. *Cacheability*

Các response có thể lấy ra từ cache. Bằng cách cache các response , server giảm tải việc xử lý request, còn client cũng nhận được thông tin nhanh hơn. Ở đây ta đặt cache vào giữa : client- cache- server.

* + 1. *Layered system*

Phân lớp hệ thống : trong hệ thống REST chia tách các thành phần hệ thống theo từng lớp, mỗi lớp chỉ sử dụng lớp ở dưới nó và giao tiếp với lớp ở ngay trên nó mà thôi. Điều này giúp giảm độ phức tạp của hệ thống, giúp các thành phần tách biệt nhau từ đó dễ dàng mở rộng từng thành phần.

* + 1. *Uniform interface*

Chuẩn hóa các interface : Đây là một trong những đặc tính quan trọng của hệ thống REST. Bằng cách tạo ra các quy ước chuẩn để giao tiếp giữa các thành phần trong hệ thống, đơn giản hóa việc client có thể tương tác với server. Các quy ước này áp dụng cho toàn bộ các service giúp cho người sử dụng hệ thống của bạn dễ dụng hơn. Dễ hiểu hơn trên hệ thống đặt ra 1 chuẩn API để người dùng dù là mobile, web đều có thể kết nối vào được. Hệ thống REST có yếu điểm ở đây vì khi chuẩn hóa rồi ta không thế tối ưu từng kết nối.

* 1. **Tại sao nên sử dụng REST ?**

Thiết kế web trước đây sử dụng SOAP (Simple Object Access Protocol) và WSDL (Web Service Definition Language), tuy nhiên bây giờ REST tối ưu hơn so với 2 phương pháp này.

Rõ ràng về URL (REST URL đại diện cho resource xác định chứ không phải hành động).

Trả về nhiều định dạng khác nhau như: html, xml, ...

Code ngắn gọn dễ hiểu.

Hiệu suất tốt, tin cậy, dễ phát triển.

* 1. **Resources**

Quản lý resource (tài nguyên) là một phần quan trọng và chiếm phần lớn trong việc phát triển website. Trong đó resource của các website khác nhau có thể sẽ khác nhau. Với các trang mạng xã hội như Facebook thì resource thường là danh sách người dùng (user hoặc account), danh sách các bài viết (post hoặc article), các ảnh được đăng (photo hoặc image), các trang fanpage (fanpage)... Đối với một trang chia sẻ ảnh như Instagram thì resource có thể là các ảnh được đăng (photo), danh sách người dùng (user)... Đối với các trang bán hàng thì resource có thể là danh sách sản phẩm (product), danh sách các người bán (seller), danh sách khách hàng (user hay customer)...

Việc quản lý resource của một website bao gồm 4 tác vụ chính:

* GET - Lấy thông tin một resource (read)
* POST - Tạo mới một resource (create)
* PUT - Cập nhật một resource (update)
* DELETE - Xoá một resource (delete)

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa. Mỗi phương thức trên phải được API call thông qua để gửi chỉ thị cho server phải làm gì. Đại đa số web API chỉ cho phép GET request lấy dữ liệu khỏi một externer server. Authencation không bắt buộc, nhưng nên có khi ta cho phép các lệnh khá “nguy hiểm” như PUT hay DELETE.

* 1. **Mã phản hồi HTTP**

Chúng ta đều biết rằng nếu cố gắng một trang không tồn tại, server sẽ trả lại trang 404 Page not Found. Nếu chúng ta làm rối tung code rails lên, server sẽ trả lại lỗi ứng dụng 500 Application Error. Đây là mã phản hồi mà server gửi cho bạn để biết điều gì đang xảy ra, sau đây là các mã HTTP:

* 200 OK Nhận được từ phương thức GET, POST hoặc DELETE. Có nghĩa là yêu cầu đã được thực hiện thành công
* 201 Created Thông báo phương thức POST đã tạo thành công đối tượng.
* 404 Not found Nhận được từ phương thức GET, POST hoặc DELETE.Không tìm thấy trang.
* 406 Not Acceptable Lỗi này chỉ ra bạn không có quyền truy xuất tài nguyên này.
* 500 Internal Server Error Có một lỗi rất lớn đang xảy ra.

1. **Github**
   1. **Github là gì ?**

Github là một dịch vụ máy chủ repository công cộng, mỗi người có thể tạo tài khoản trên đó để tạo ra các kho chứa của riêng mình để có thể làm việc.

Github hoạt động ddoognj dựa trên hệ thống quản lý phiên bản phân tán (Distributed Vrsion Control System - DVCS) là một trong những hệ thống quả lý phiên bản phân tán phổ biến nhất hiện nay. DVCS nghĩa là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (clone) từ một kho chứa mã nguồn (repository), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (commit) rồi đưa lên máy chủ nơi đặt kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia.

A screenshot of a cell phone

Description generated with very high confidence

Hình 2.3: mô hình hoạt động DVCS

* 1. **Tại sao nên sử dụng Git?**

Những lý do nên sử dụng Git là:

* Git dễ dàng sử dụng, an toàn và nhanh chóng.
* Có thể giúp bạn quy trình làm việc code theo nhóm đơn giản, thuận tiện.
* Có thể làm việc bất kỳ ở đâu vì chỉ cần clone mã nguồn từ kho chứa.
* Dễ dàng trong việc deployment sản phẩm.

# Chương III. Phân tích thiết kế hệ thống

1. **Xác định yêu cầu, khảo sát nghiệp vụ**
   1. **Xác định yêu cầu**
   2. *Yêu cầu về chức năng*

Sau khi nắm bắt yêu cầu cũng như tham khảo một số website thì nhóm chúng em thấy có những chức năng quan trọng như sau:

* Chức năng tìm kiếm, bộ lọc kết quả.
* Chức năng đăng ký nhận thông tin qua email.
* Cập nhập chi tiết hồ sơ.
* Đăng ký, đăng nhập.
* Đăng tin tuyển dụng.
* Nộp hồ sơ ứng tuyển.
* Đánh giá đối tác sau khi giao dịch thành công.
  1. *Yêu cầu về giao diện*

Một số yêu cầu về giao diện của trang web như sau:

* Trang chủ giao diện đơn giản, thân thiện
* Trang tìm kiếm, kết quả làm nổi bật lên các trường quan trọng như: tên công ty, thời hạn đăng tuyển…
* Bố cục hài hòa, màu sắc dễ nhìn
  1. *Yêu cầu về hệ thống*

Một số yêu cầu về giao diện của trang web như sau:

* Hệ thống hoạt động nhanh, chính xác
* Có thể nâng cấp, update, bảo trì dễ dàng
  1. **Khảo sát nghiệp vụ**
  2. **Ưu điểm:**

Hiện nay trên internet có rất nhiều website tuyển dụng việc làm như: vietnamworks, itviec, vlance…, trang web nhóm ưu tiên sử dụng để khảo sát nghiệp vụ là vietnamworks và trang web timviecnhanh đã triển khai tốt các nghiệp vụ đăng tin, tìm kiếm việc làm vì vậy đấy cũng là một nguồn tham khảo các nghiệp vụ tốt để em có thể áp dụng vào đề tài kết hợp với yêu cầu bài toán. Song với đặc thù bài toán đặt ra là một trang web tuyển dụng việc làm đặc thù cho các ngành như xây dựng, hạ tầng nên cần có những yêu cầu đặc thù khác. Các trang web tuyển dụng việc làm hiện có trên internet đã triển khai tốt các nội dụng như:

* Đăng tin tuyển dụng
* Gợi ý việc làm phù hợp
* Gửi email thông báo khi có việc làm phù hợp
* Thiết kế giao diện web đẹp.
* Chức năng tìm kiếm việc tiện lợi, tìm theo chuyên ngành, vị trí, …
* Có chức năng quản lý thông tin cá nhân của doanh nghiệp , người dùng.
* Thủ tục nhận việc nhanh chóng.
* Web responsive trên cả mobile và tablet.
* Có chức năng tạo CV để tìm việc
* Đăng nhập được với tài khoản facebook hoặc google cực kỳ tiện lợi.
* Có bộ phận tư vấn cho người dùng.
  1. **Nhược điểm:**
* Không có chức năng tìm kiếm công việc phù hợp dựa trên những kỹ năng người dùng.
* Trang web khá chậm, giảm trải nghiệm người dùng.
  1. **Các đối tượng tham gia vào hệ thống**

Người tìm việc:

* Là người tìm kiếm các công việc trên website

Người đăng việc (Nhà tuyển dụng):

* Là người đăng các thông tin tuyển dụng lên website

Người quản trị hệ thống:

* Là người quản trị hệ thống, thống kê, báo cáo.

Người tìm việc khi truy cập vào trang web sẽ có khả năng tìm việc làm (theo địa điểm, theo ngành nghề, … )

Sau khi đăng ký làm thành viên, người tìm việc hoặc người đăng công việc (nhà tuyển dụng) có thể đăng nhập với tài khoản đã được đăng ký. Sau khi đăng nhập với tư cách người dùng hoặc nhà tuyển dụng, thì có thể quản lý thông tin hồ sơ của mình, một tài khoản người dùng có thể đồng thời là nhà tuyển dụng, hay là người tìm việc.

Khi đăng nhập với tư cách người tìm việc thì có thể xem hồ sơ của mình và đi nhận những công việc được hiển thị trên web.

Khi đăng nhập với tư cách nhà tuyển dụng thì có thể xem hồ sơ của công ty mình và đăng công việc tìm kiếm người làm, xem những người đã apply vào công việc mình đăng

Cả 2 vai trò trên có thể quản lý hợp đồng trong trang hồ sơ của mình.

1. **Xác định các Actor và Usecase**

**2.1. Các tác nhân tham gia vào hệ thống**

Các tác nhân tham gia vào hệ thống gồm:

* Khách: là actor khi vào hệ thống mà chưa đăng ký tài khoản hoặc chưa đăng nhập vào tài khoản.
* Người tìm việc: là actor khi vào hệ thống đã đăng nhập.
* Người đăng việc (nhà tuyển dụng): là actor khi vào hệ thống đã đăng nhập.
* Người quản trị hệ thống: là actor có quyền quản lý tất cả các hoạt động trong hệ thống.

**2.2. Biểu đồ các Actor và Usecase**

A close up of a map

Description generated with high confidence

Hình 3.1 Biểu đồ usecase website

* + 1. **Actor:**
* Khách truy cập: Người dùng trực tiếp sử dụng website nhưng chưa có tài khoản hoặc chưa đăng nhập tài khoản.
* Người tìm việc: Người đi tìm việc đã đăng nhập tài khoản vào website.

Dễ nhận thấy, người tìm việc kế thừa toàn bộ quyền của khách truy cập



Hình 3‑2 Sự kế thừa của actor

* Người đăng việc: Người dùng trực tiếp trên website đã đăng nhập tài khoản.
* Admin: Người quản trị toàn quyền hệ thống.
  + 1. **Usecase:**
* Usecase cho hệ thống của admin:



Hình 1: Hệ thống usecase cho Admin

* Usecase cho hệ thống của người tìm việc:



Hình 2: Hệ thống usecase cho người tìm việc

* Usecase cho hệ thống của người đăng việc (nhà tuyển dụng):



Hình 3: Hệ thống usecase cho người đăng tuyển

**2.3. Đặc tả Usecase**

a. Usecase liên quan đến người tìm việc

* Đăng nhập:

|  |  |  |
| --- | --- | --- |
| Tên | Đăng nhập | |
| Mô tả | Người dùng đăng nhập vào hệ thống | |
| Actor | Người quản trị, khách | |
| Điều kiện | Người dùng đã truy cập website, đã đăng ký tài khoản và chưa đăng nhập. | |
| Luồng sự kiện chính | B1 | Người dùng chọn chức năng đăng nhập |
| B2 | Người dùng nhập tên email và mật khẩu |
| B3 | Người dùng click vào nút đăng nhập |
| B4 | Website thông báo đăng nhập thành công và tiếp tục truy cập website |
| B5 | Kết thúc usecase. |
| Luồng sự kiện phụ | B2.1 | Người dùng không điền đầy đủ thông tin đăng nhập, website đưa ra thông báo yêu cầu điền đầy đủ thông tin và quay lại B2. |
| B2.2 | Người dùng nhập sai thông tin đăng nhập, website đưa ra thông báo yêu cầu nhập lại và quay lại bước B2. |

* Đăng ký:

|  |  |  |
| --- | --- | --- |
| Tên | Đăng ký | |
| Mô tả | Người dùng đăng ký làm thành viên | |
| Actor | Khách | |
| Điều kiện | Người dùng đã truy cập vào website và chưa có tài khoản hoặc muốn đăng ký tài khoản mới. | |
| Luồng sự kiện chính | B1 | Người dùng chọn mục đăng ký |
| B2 | Nhập đầy đủ thông tin bắt buộc |
| B3 | Người dùng click vào nút đăng ký |
| B4 | Website thông báo đăng ký thành công và đưa tới trang đăng nhập. |
| B5 | Kết thúc usecase |
| Luồng sự kiện phụ | B2.1 | Người dùng không điền đầy đủ thông tin bắt buộc, website đưa ra thông báo yêu cầu điền đầy đủ thông tin và quay lại B2. |
| B2.2 | Người dùng nhập thông tin đăng ký không hợp lệ sẽ trả về thông báo và quay lại B2 |

* Tìm việc

|  |  |  |
| --- | --- | --- |
| Tên | Tìm việc | |
| Mô tả | Người dùng tìm kiếm công việc | |
| Actor | Ngườ dùng, khách, quản trị viên | |
| Điều kiện | Người dùng đã truy cập vào website. | |
| Luồng sự kiện chính | B1 | Người dùng nhập thông tin công việc cần tìm kiếm vào ô tìm kiếm |
| B2 | Nhấn nút tìm kiếm |
| B3 | Website đưa ra kết quả tìm kiếm phù hợp với thông tin người dùng yêu cầu tìm kiếm |
| B4 | Kết thúc usecase |
| Luồng sự kiện phụ | B2.1 | Nếu không có kết quả phù hợp với tiêu chí người dùng đề ra thì sẽ trả về trang trắng |

* Nộp hồ sơ công việc

|  |  |  |
| --- | --- | --- |
| Tên | Tìm việc | |
| Mô tả | Nộp hồ sơ công việc | |
| Actor | Người dùng | |
| Điều kiện | Người dùng đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Chọn công việc cần nộp |
| B2 | Nhấn nút nộp hồ sơ |
| B3 | Hiện thị modal thông tin và chọn file CV cần nộp |
| B4 | Nhấn nút nộp hồ sơ để gửi thông tin và CV cho nhà tuyển dụng |
| B5 | Website đưa ra thông báo thành công |
| B6 | Kết thúc Usecase |
| Luồng sự kiện phụ | B3.1 | Nếu không chọn file CV thì quay lại bước 3 |

* Đăng xuất

|  |  |  |
| --- | --- | --- |
| Tên | Đăng xuất | |
| Mô tả | Đăng xuất tài khoản | |
| Actor | Người dùng | |
| Điều kiện | Người dùng đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Người dùng chọn chức năng đăng xuất |
| B2 | Người dùng chọn chức năng đăng xuất |
| B3 | Website xóa thông tin đăng nhập và trở về trang chủ |
| B4 | Nhấn nút nộp hồ sơ để gửi thông tin và CV cho nhà tuyển dụng |
| B5 | Kết thúc Usecase |

* Cập nhật hồ sơ thông tin

|  |  |  |
| --- | --- | --- |
| Tên | Cập nhật hồ sơ | |
| Mô tả | Cập nhật hồ sơ thông tin | |
| Actor | Người dùng | |
| Điều kiện | Người dùng đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Người dùng chọn chức năng quản lý hồ sơ |
| B2 | Người dùng điền đầy đủ thông tin hồ sơ và click và nút cập nhật hồ sơ |
| B3 | Website đưa ra thông báo cập nhật thành công |
| B4 | Kết thúc Usecase |
| Luồng sự kiện phụ | B2.1 | Nếu người dùng nhập sai thông tin thì quay về bước 2. |

b. Usecase liên quan đến người đăng tin tuyển dụng

* Đăng nhập

|  |  |  |
| --- | --- | --- |
| Tên | Đăng nhập | |
| Mô tả | Người đăng việc đăng nhập | |
| Actor | Người đăng tin tuyển dụng | |
| Điều kiện | Người dùng đã truy cập website, đã đăng ký với tài khoản lf nhà tuyển dụng và chưa đăng nhập. | |
| Luồng sự kiện chính | B1 | Người dùng chọn chức năng đăng nhập |
| B2 | Người dùng nhập tên đăng nhập và mật khẩu,click nút đăng nhập |
| B3 | Website thông báo đăng nhập thành công và tiếp tục truy cập website |
| B4 | Kết thúc Usecase |
| Luồng sự kiện phụ | B2.1 | Người dùng không điền đầy đủ thông tin đăng nhập, website đưa ra thông báo yêu cầu điền đầy đủ thông tin và quay lại B2. |
| B2.2 | Người dùng nhập sai thông tin đăng nhập, website đưa ra thông báo yêu cầu nhập lại và quay lại bước B2. |

* Đăng ký

|  |  |  |
| --- | --- | --- |
| Tên | Đăng ký | |
| Mô tả | Người dùng đăng ký làm nhà tuyển dụng | |
| Actor | Khách, người dùng | |
| Điều kiện | Người dùng đã truy cập vào website và chưa có tài khoản hoặc muốn đăng ký tài khoản mới. | |
| Luồng sự kiện chính | B1 | Người dùng chọn mục đăng ký |
| B2 | Nhập đầy đủ thông tin bắt buộc và click vào nút đăng ký |
| B3 | Website thông báo đăng ký thành công và đưa tới trang đăng nhập. |
| B4 | Kết thúc Usecase |
| Luồng sự kiện phụ | B2.1 | Người dùng không điền đầy đủ thông tin bắt buộc, website đưa ra thông báo yêu cầu điền đầy đủ thông tin và quay lại B2. |
| B2.2 | Người dùng nhập thông tin đăng ký không hợp lệ sẽ trả về thông báo và quay lại B2 |

* Đăng xuất

|  |  |  |
| --- | --- | --- |
| Tên | Đăng xuất | |
| Mô tả | Người đăng việc đăng xuất | |
| Actor | Người đăng việc | |
| Điều kiện | Người đăng việc đã truy cập website và đăng nhập hệ thống | |
| Luồng sự kiện chính | B1 | Người dùng chọn chức năng đăng xuất |
| B2 | Website xác nhận lại yêu cầu đăng xuất |
| B3 | Website xóa thông tin đăng nhập và trở về trang chủ |
| B4 | Kết thúc Usecase |

* Cập nhật hồ sơ thông tin

|  |  |  |
| --- | --- | --- |
| Tên | Cập nhật hồ sơ | |
| Mô tả | Cập nhật hồ sơ thông tin | |
| Actor | Người đăng việc | |
| Điều kiện | Người đăng việc đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Người đăng việc chọn chức năng quản lý hồ sơ |
| B2 | Người đăng việc điền đầy đủ thông tin hồ sơ và click và nút cập nhật hồ sơ |
| B3 | Website đưa ra thông báo cập nhật thành công |
| B4 | Kết thúc Usecase |
| Luồng sự kiện phụ | B2.1 | Nếu người đăng việc nhập sai thông tin thì quay về bước 2. |

* Tìm kiếm, quan tâm ứng viên và mời ứng tuyển

|  |  |  |
| --- | --- | --- |
| Tên | Tìm kiếm và ứng tuyển | |
| Mô tả | Tìm kiếm, quan tâm ứng viên và mời ứng tuyển | |
| Actor | Người đăng việc | |
| Điều kiện | Người đăng việc đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Tìm kiếm, quan tâm ứng viên và mời ứng tuyển |
| B2 | Website đưa ra danh sách ứng viên phù hợp với lĩnh vực, ngành nghề, kỹ năng tìm kiếm |
| B3 | Người đăng việc đưa click vào quan tâm ứng viên hoặc mời ứng tuyển |
| B4 | Website đưa ra thông báo thành công |
| B5 | Kết thúc usecase |

* Xem danh sách ứng tuyển

|  |  |  |
| --- | --- | --- |
| Tên | Xem danh sách ứng tuyển | |
| Mô tả | Xem danh sách ứng tuyển | |
| Actor | Người đăng việc | |
| Điều kiện | Người đăng việc đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Người đăng việc chọn chức năng xem danh sách ứng tuyển |
| B2 | Chọn công việc cần xem danh sách |
| B3 | Website đưa ra danh sách |
| B4 | Kết thúc usecase |

* Xem danh sách ứng tuyển

|  |  |  |
| --- | --- | --- |
| Tên | Xem danh sách ứng tuyển | |
| Mô tả | Xem danh sách ứng tuyển | |
| Actor | Người đăng việc | |
| Điều kiện | Người đăng việc đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Người đăng việc chọn chức năng xem danh sách ứng tuyển |
| B2 | Chọn công việc cần xem danh sách |
| B3 | Website đưa ra danh sách |
| B4 | Kết thúc usecase |

* Quản lý ứng viên

|  |  |  |
| --- | --- | --- |
| Tên | Quản lý ứng viên | |
| Mô tả | Quản lý ứng viên | |
| Actor | Người đăng việc | |
| Điều kiện | Người đăng việc đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Người đăng việc chọn chức năng xem danh sách ứng viên |
| B2 | Website đưa ra danh sách |
| B3 | Kết thúc usecase |

* Đăng tin tuyển dụng

|  |  |  |
| --- | --- | --- |
| Tên | Đăng tin tuyển dụng | |
| Mô tả | Đăng tin tuyển dụng | |
| Actor | Người đăng việc | |
| Điều kiện | Người đăng việc đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Người đăng việc chọn chức năng đăng tin tuyển dụng |
| B2 | Người đăng việc nhập đầy đủ thông tin yêu cầu và nhấn vào nút đăng tin |
| B3 | Website đưa ra thông báo thành công |
| B4 | Kết thúc usecase |
| Luồng sự kiện phụ | B2.1 | Nếu người đăng việc không nhập thông tin thì quay lại bước 2 |
| B2.2 | Nếu người đăng việc nhập sai thông tin thì quay lại bước 2. |

* + 1. Usecase liên quan đến người quản trị hệ thống
* Quản lý tin đăng tuyển

|  |  |  |
| --- | --- | --- |
| Tên | Quản lý tin đăng tuyển | |
| Mô tả | Đăng tin tuyển dụng | |
| Actor | admin | |
| Điều kiện | Người dùng đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Chọn chức năng quản lý tin đăng tuyển |
| B2 | Chọn chức năng quản lý tin đăng tuyển |
| B3 | Kết thúc usecase |

* Quản lý người dùng

|  |  |  |
| --- | --- | --- |
| Tên | Quản lý người dùng | |
| Mô tả | Quản lý người dùng | |
| Actor | admin | |
| Điều kiện | Người dùng đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Chọn chức năng quản lý người dùng |
| B2 | Website đưa ra danh sách người dùng |
| B3 | Kết thúc usecase |

* Quản lý kỹ năng

|  |  |  |
| --- | --- | --- |
| Tên | Quản lý kỹ năng | |
| Mô tả | Quản lý kỹ năng | |
| Actor | admin | |
| Điều kiện | Người dùng đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Chọn chức năng quản lý người dùng |
| B2 | Website đưa ra danh sách người dùng |
| B3 | Kết thúc usecase |

* Quản lý ngành nghề

|  |  |  |
| --- | --- | --- |
| Tên | Quản lý ngành nghề | |
| Mô tả | Quản lý ngành nghề | |
| Actor | admin | |
| Điều kiện | Người dùng đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Chọn chức năng quản lý người dùng |
| B2 | Website đưa ra danh sách người dùng |
| B3 | Kết thúc usecase |

* Gửi email phù hợp

|  |  |  |
| --- | --- | --- |
| Tên | Gửi email phù hợp | |
| Mô tả | Gửi email phù hợp | |
| Actor | admin | |
| Điều kiện | Người dùng đã truy cập website và đăng nhập vào hệ thống | |
| Luồng sự kiện chính | B1 | Chọn chức năng gửi mail thông báo |
| B2 | Gửi mail phù hợp với danh sách đã đăng ký |
| B3 | Website đưa ra thông báo gửi mail thành công |
| B4 | Kết thúc usecase |

1. **Biểu đồ tuần tự**

**3.1. Biểu đồ tuần tự dành cho người tìm việc**

A screenshot of a map

Description generated with very high confidence

Hình 3.3: Biểu đồ tuần tự đăng nhập người tìm việc

A close up of a map

Description generated with very high confidence

Hình 3.3: Biểu đồ tuần tự đăng ký người tìm việc

A close up of a map

Description generated with high confidence

Hình 3.5: Biểu đồ tuần tự đăng xuất người tìm việc

A screenshot of a cell phone

Description generated with very high confidence

Hình 3.6: Biểu đồ tuần tự người dùng tìm kiếm công việc

A close up of a map

Description generated with very high confidence

Hình 3.7: Biểu đồ tuần tự người dùng ứng tuyển công việc

**3.2. Biểu đồ tuần tự dành cho nhà tuyển dụng**

**3.3. Biểu đồ tuần tự dành cho admin**

1. **Biểu đồ lớp**
2. **Biểu đồ hoạt động**
3. **Thiết kế cơ sở dữ liệu**
4. **Thiết kế giao diện**

# Chương IV. Cài đặt và kiểm thử

1. **Cài đặt**
2. **Kiểm thử**

# Chương V. Tổng kết

1. **Kết quả đạt được**
2. **Hạn chế**
3. **Hướng phát triển**

# TÀI LIỆU THAM KHẢO

**[1] :** [**Blogtienao.com**](https://blogtienao.com/) **-** Ethereum là gì? Tất tần tật kiến thức về đồng tiền ảo Ethereum (ETH)

https://blogtienao.com/ethereum/

**[2] :** Phan Long - Ethereum là gì?

<https://phanlonghi.com/ethereum-la-gi/>