TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐOÀN NGỌC NAM - 52000084**

**Xây dựng website đánh giá công ty với GraphQL và Microservices**

**BÁO CÁO DỰ ÁN CÔNG NGHỆ THÔNG TIN**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐOÀN NGỌC NAM - 52000084**

**Xây dựng website đánh giá công ty với GraphQL và Microservices**

**BÁO CÁO DỰ ÁN CÔNG NGHỆ THÔNG TIN**

Người hướng dẫn

**Thạc Sĩ Doãn Xuân Thanh**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**LỜI CẢM ƠN**

Em chân thành cảm ơn!

*TP. Hồ Chí Minh, ngày 15 tháng 7 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Nam*

*Đoàn Ngọc Nam*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của Thạc sĩ Doãn Xuân Thanh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 15 tháng 7 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Nam*

*Đoàn Ngọc Nam*

**TÓM TẮT**

Trong bài cáo cáo này, tôi sẽ trình bày về các chủ đề GraphQL, Microservices và xây dựng hệ thống đánh giá công ty trong lĩnh cực Công nghệ thông tin.

**MỤC LỤC**

[DANH MỤC HÌNH VẼ vii](#_Toc172213756)

[DANH MỤC BẢNG BIỂU viii](#_Toc172213757)

[DANH MỤC CÁC CHỮ VIẾT TẮT ix](#_Toc172213758)

[CHƯƠNG 1. BÀI TOÁN TIME SIRIES 1](#_Toc172213759)

[1.1 Time siries data 1](#_Toc172213760)

[1.2 Thuật toán sử dụng 1](#_Toc172213761)

[CHƯƠNG 2. REINFORCEMENT LEARNING 2](#_Toc172213762)

[2.1 Sơ lược 2](#_Toc172213763)

[2.2 Các thành phần 2](#_Toc172213764)

[2.2.1 Agent and environment 2](#_Toc172213765)

[2.2.2 Policy 3](#_Toc172213766)

[2.2.3 Reward 3](#_Toc172213767)

[2.2.4 Model 4](#_Toc172213768)

[2.3 Khai thác và khám phá 5](#_Toc172213769)

[2.4 Reinforcement Learning Algorithms 5](#_Toc172213770)

[2.4.1 Model-based 5](#_Toc172213771)

[2.4.2 Free-model algorithms 6](#_Toc172213772)

[2.5 Ứng dụng của RL 6](#_Toc172213773)

[2.5.1 Game 6](#_Toc172213774)

[2.5.2 Xe tự hành 7](#_Toc172213775)

[2.5.3 Robotic 8](#_Toc172213776)

[2.5.4 Hệ thống gợi ý 8](#_Toc172213777)

# DANH MỤC HÌNH VẼ

[Hình 2.1: Scaled Dot-Product Attention 3](#_Toc142677545)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1: Thống kê kiểu thực thể trong tập VLSP 2016 4](#_Toc142677565)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| RL | Reinforcement Learning |
| GEC | Grammatical Error Correction |
| MLM | Masked Language Model |
| NLP | Natural Language Processing |
| RC | Recommendation System |

# GIỚI THIỆU ĐỀ TÀI

Sau hàng chục năm phát triển, thế giới Internet ngày nay đã có những bước chuyển mình mạnh vẽ với vô vàn các công nghệ mới. Trong đó, các công nghệ để xây dựng và phát triển các ứng dụng web, tuy không phải là một cơn lốc nhanh và mạnh mẽ như những gì AI và Data Science đã làm được, nhưng lại là mảng có sự mở rộng nhiều nhất, nâng cấp nhiều nhất. Từ những ngày đầu thế kỉ khi chúng ta sử dụng HTML, CSS, JS cho việc xây dựng 1 web site tĩnh, sử dụng PHP để xây dựng hệ thống backend. Rồi đến sự ra đời của NodeJS, môi trường cho phép chạy ngôn ngữ JavaScript từ server thay vì chỉ có thể chạy trên trình duyệt như trước gần như đã mở ra 1 cuộc cách mạng về các framework và thư viện để xây dựng các ứng dụng web. Cùng thời điểm này, với nhu cầu mở rộng không ngừng của xã hội và nền kinh tế, các hệ thống công nghệ thông tin vốn đã trở thành một phần không thể thiếu trong mọi lĩnh vực cuộc sống, cũng đứng trước nhu cầu mở rộng không ngừng và cập nhật liên tục để trở nên phù hợp và đáp ứng các nhu cầu của người dùng. Khi này các hệ thống theo dạng Monolothic cũ đã lộ ra các khuyết điểm khi không thể đáp ứng các nhu cầu cập nhật liên tục và khó để bảo trì, mở rộng. Thời điểm ấy, Microservice được nhắc đến và ứng dụng rộng dãi hơn trong cộng đồng, mặc dù nó đã được các công ty công nghệ lớn như Amazon, Netflix tiên phong sử dụng từ những năm trước đó.

Cùng trong những năm này, Facebook, một trong những công ty công nghệ lớn nhất thế giới đang gặp phải một vấn đề đó là việc chia sẻ API giữa các nền tảng, hoặc các thành phần trên cùng 1 nền tảng. Ví dụ dễ hình dung nhất chính là việc nếu ở trên website cần 1 API trả về 1 lượng thông tin, thì ở phía mobile, họ chỉ cần 1 phần của lượng thông tin đó. Để giải quyết vấn đề này GraphQL ra đời. Và cũng trong khoảng thời gian này, Facebook đã đem đến 1 cuộc cách mạng cho việc lập trình giao diện người dùng phía client, đó là việc cho ra đời thư viện ReactJS.

Những thuận lợi về công nghệ đã có, vậy còn nhu cầu thực tiễn của đề tài thì sao?

Không thể phủ nhận rằng, hiện nay sự cạnh tranh trong thị trường việc làm ngành Công nghệ thông tin là vô cùng lớn. Sự “thiếu hụt nhân lực IT” vẫn luôn là 1 tiêu đề quen thuộc của các bài báo mạng, tuy nhiên, điều đó chỉ đúng với nhân lực chất lượng cao, với nhiều năm kinh nghiệm. Nhưng dù ở level nào, việc lựa chọn 1 công ty tốt và phù hợp cho việc phát triển bản thân luôn là một vấn đề trăn trở. Trên các trang mạng hiện nay, các trang web cho phép review các công ty thường mang các yếu tố hoặc là quá tiêu cực, hoặc là quá seeding. Điều này vô tình gây ra một vài sự lưỡng lự trong việc tin tưởng và lựa chọn tiếp nhận thông tin.

Với những yếu tố thuận lợi kể trên là điều kiện lý tưởng để em thực hiện đề tài: Tìm hiểu về Microservices và GraphQL ứng dụng vào xây dựng website đánh giá công ty.

# PHÂN TÍCH VÀ THIẾT KẾ

## Phân tích và thiết kế

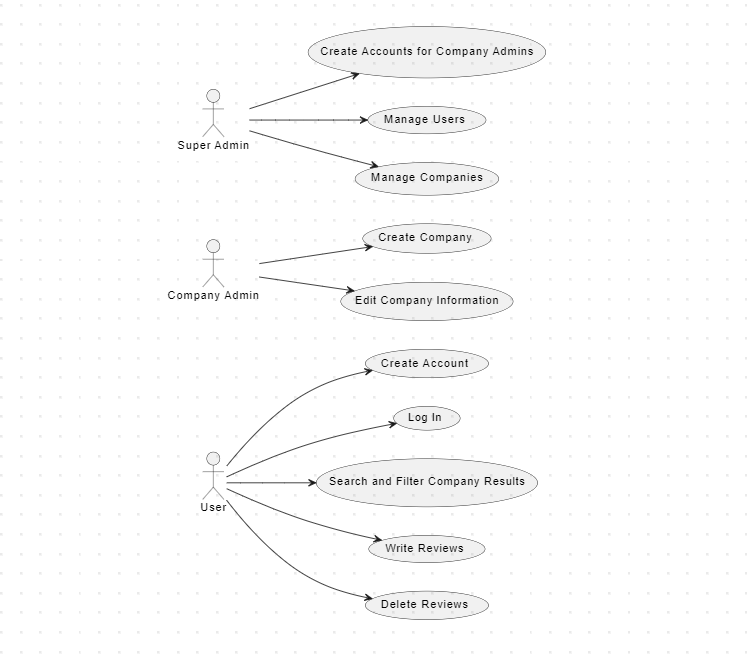
### Sơ đồ usecase

Với một hệ thống như đã được giới thiệu ở trên, ngoài việc phân chia các module sao cho phù hợp và mượt mà, chúng ta cũng cần quan tâm đến việc các bên giao tiếp với nhau để dễ dàng hơn trong việc phân chia services, database sau này.

Hệ thống sẽ bao gồm 3 tác nhân (actor) tương ứng với các vai trò và quyền hạn khác nhau bao gồm:

* User: đại diện cho người dùng thông thường, cũng là người dùng cuối cùng của hệ thống, tác nhân này có thể tạo tài khoản (account), đăng nhập (login), tìm kiếm và lọc kết quả các công ty (search) theo các tiêu chí và yêu cầu khác nhau, viết bài đánh giá (review) và xóa bài đánh giá
* Company admin: người quản lý cho 1 công ty, là người chịu trách nhiệm quản lý thông tin và hình ảnh của công ty mình trên hệ thống. Company admin có thể tạo một công ty, sửa thông tin cho công ty, đưa ra các phản hồi và thu thập thông tin đánh giá công ty của mình.
* Super admin: quản trị viên, là người duy trì toàn bộ hoạt động của hệ thống, đảm bảo rằng hệ thống hoạt động trơn chu quản trị viên sẽ là người tạo và cung cấp tài khoản cho Company admin, quản lý tất cả người dùng và các công ty trong hệ thống

Dưới đây là sơ đồ use case tổng quát của hệ thống, cho thấy các chức năng mà mỗi tác nhân có thể thực hiện.



Hình 2.: Sơ đồ Use Case hệ thống

### Thiết kế cơ sở dữ liệu

Sau khi đã có được các service cần thiết, chung ta tiến hành thiết kế cơ sở dữ liệu cho hệ thống. Tuy rằng có các rang buộc về quan hệ cho mỗi bảng và mỗi thực thể, ở phần này, em vẫn sẽ thiết kế CSDL một các cơ bản nhất, phần rang buộc khóa ngoại và mối quan hệ sẽ được triển khai ở tầng Application.

**Bảng User**

Đây là bảng chứa các thông tin cơ bản của một người dùng (tên, email,…), bao gồm cả company admin. Ngoài các thông tin này, bảng sẽ có thuộc tính role/permission để xác định quyền thực hiện các hành động, và xác định người dùng là *thông thường* hay là *company admin.*

**Bảng Company**

Là bảng chứa các thông tin của 1 công ty (tên, email, địa chỉ, lĩnh vực,…). Một công ty sẽ được quản lý bởi một *company admin* duy nhất, và một *company admin* cũng chỉ được quản lý một công ty duy nhất.

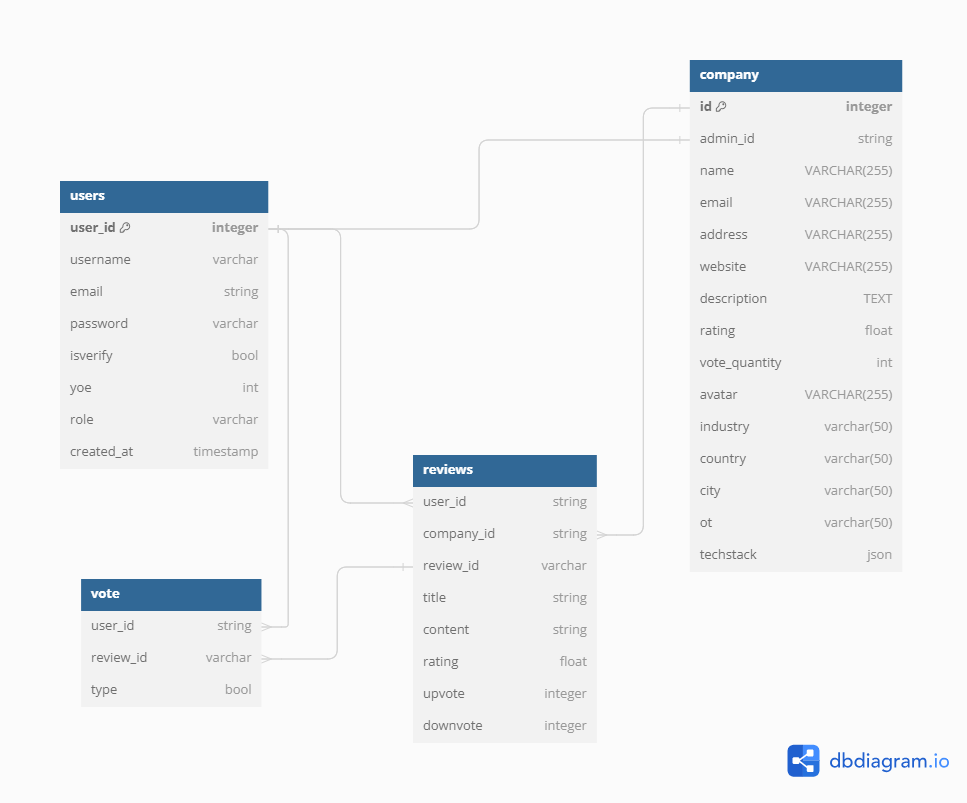
**Bảng Review**

Bảng chứa thông tin về mỗi nhận xét của người dùng cho một công ty, một người dùng có thể nhận xét một lần duy nhất về một công ty, và một công ty có thể nhận được nhiều lượt đánh giá từ nhiều nhân sự khác nhau. Mỗi đánh giá có thể nhận về lượt upvote/downvote từ người dùng khác.

**Bảng Vote**

Bảng xác định lượt upvote/downvote cho một đánh giá, đảm bảo rằng một người dùng chỉ có thể up/down vote 1 lần cho 1 đánh giá, và chỉ có thể là một trong hai giá trị này.

Dựa trên các thông tin đã có được từ những gì phân tích ở trên, chúng ta tiến hành khởi tạo cơ sở dữ liệu và vẽ lược đồ cơ sở dữ liệu:



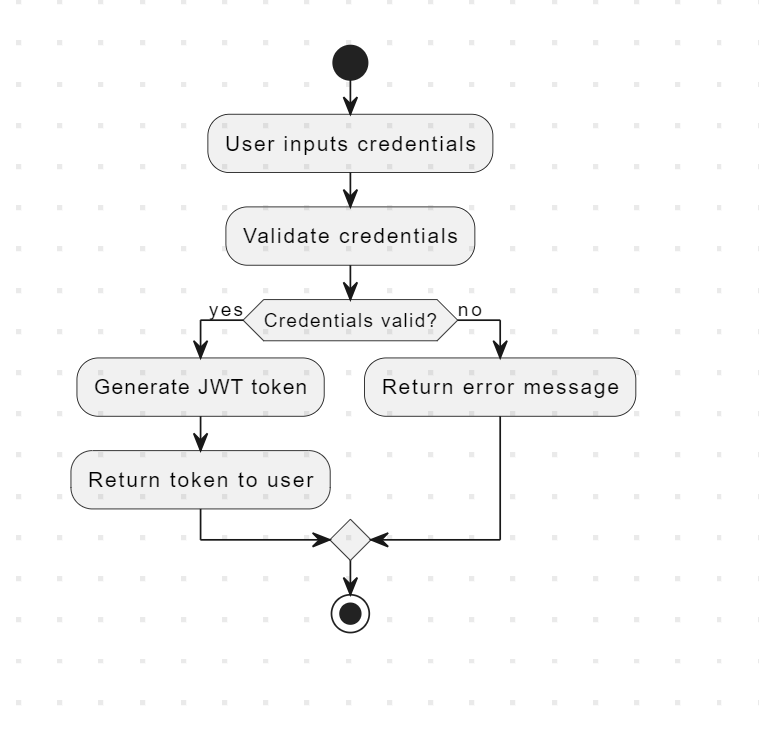
Hình 2.: Lược đồ cơ sở dữ liệu

### Thiết kế các module

#### Login

Ở phần này, chúng ta sử dụng chiến lược xác thực cơ bản là Token Base Authentication với thư viện rất phổ biến JWT (Json Web Token). Quy trình xác thực diễn ra như sau:

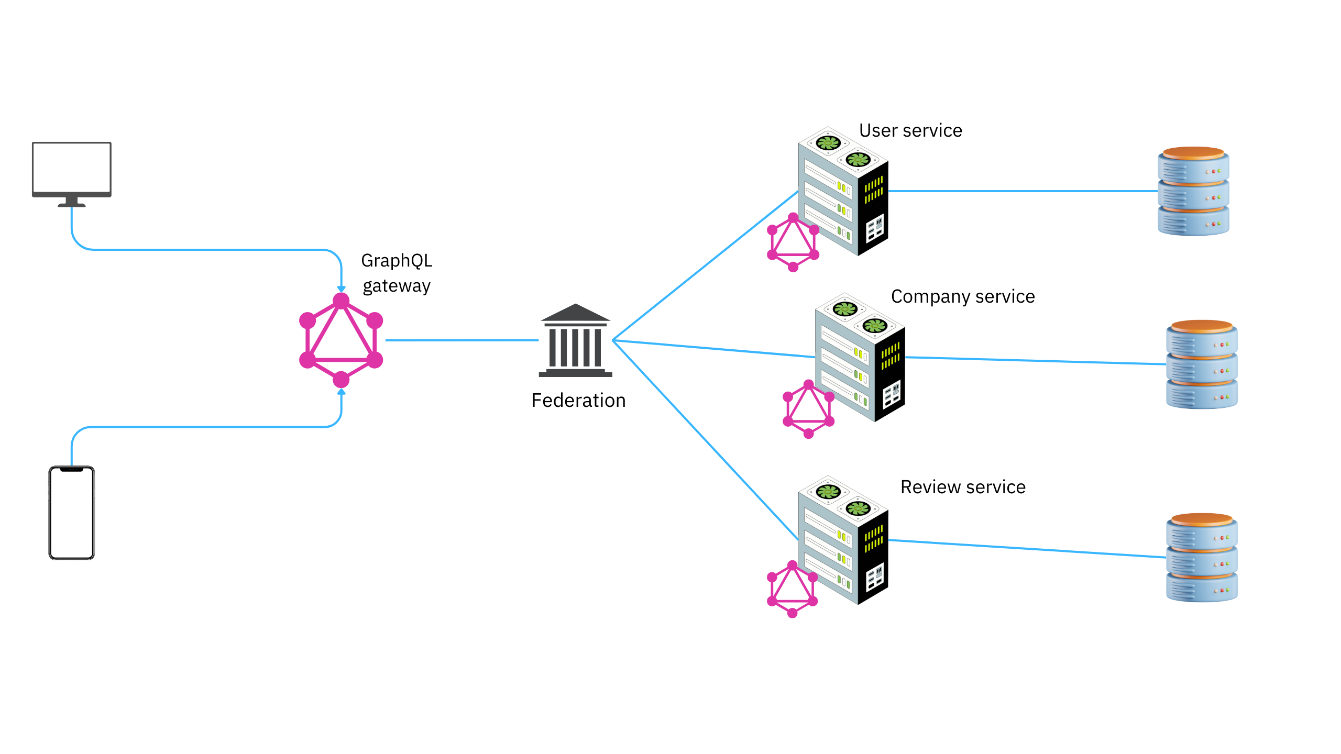
* **Người dùng gửi yêu cầu đăng nhập:** Người dùng nhập thông tin đăng nhập (username và password) vào form đăng nhập và gửi yêu cầu tới server.
* **Xác thực thông tin đăng nhập**: Server kiểm tra thông tin đăng nhập của người dùng bằng cách đối chiếu với dữ liệu đã lưu trong cơ sở dữ liệu. Nếu thông tin đúng, server sẽ tạo ra một mã token (JWT - JSON Web Token).
* **Cung cấp Token cho người dùng:** Token được gửi lại cho người dùng cùng với phản hồi đăng nhập thành công. Khi người dùng thực hiện các yêu cầu đến server (ví dụ: truy cập trang dashboard), token sẽ được gửi kèm trong phần header của HTTP request



Hình 2.: Activity Diagram chức năng Login

# KIẾN TRÚC HỆ THỐNG

Dựa trên những thiết kế đã phân tích ở trên, chúng ta đi đến xây dựng kiến trúc tổng quát hơn với việc triển khai các module thành các service nhỏ lẻ, tách cơ sở dữ liệu ban đầu thành các cơ sở dư liệu nhỏ hơn và triển khai microservice cùng với GraphQL.



Hình .: System Diagram

## Microservices