**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**XÂY DỰNG WEBSITE CRM**

**QUẢN LÝ HỒ SƠ ỨNG TUYỂN NHÂN VIÊN CHO CÔNG TY**

**PHẠM TIẾN ĐẠT**

dat.pt168655@sis.hust.edu.vn

**Ngành Công nghệ thông tin**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | TS. Trần Quang Đức  Chữ ký của GVHD |
| **Bộ môn:** | **Truyền thông và Mạng máy tính** |
| **Viện:** | Công nghệ thông tin và truyền thông |
| **HÀ NỘI, 3/2021** | |

**ĐỀ TÀI TỐT NGHIỆP**

Hệ thống hỗ trợ quản lý các hồ sơ tuyển dụng của ứng viên theo từng vị trí tuyển dụng của công ty

**Acknowledgements**

Giáo viên hướng dẫn

Ký và ghi rõ họ tên

During the graduation project, I have received a lot of help, suggestions, and enthusiastic advice from teachers, family, and friends.

I want to express my sincere thanks to Mr. Tran Quang Duc, who has guided and taught me during the project development process. I am so grateful for my teachers’ help from my university and from the Institute of Information Technology in particular regarding all the valuable general and advanced knowledge I have acquired. Hence, I have formed a pretty solid theoretical basis and practical experiences through projects, which plays an important role in my problem-solving skills development.

Finally, I am really thankful for my family and friends’ care, your help and emboldening me throughout the learning process and the completion of this graduation project. Much obliged to you all!

**Abstract**

Nowadays, the Internet has become a popular and essential service and has a profound influence on the habits, activities and entertainment of many people. With the rapid growth of the economy, companies have sprung up and boomed, and I researched and learned then realized that not many companies use technology to manage records. Employee applications, most of whom use traditional management methods. With that need, along with the subject requirements, I decided to choose the topic Building a Website to manage the profiles of the candidates for the company to help the company's recruiting staff to monitor and manage. Candidate profiles for easy care and recruitment

In terms of technology, I choose the Java language to build the Back-end for the application, the data will be saved in MySQL and use Google Firebase to save data about images and files. I will apply server authentication according to the OAuth 2.0 standard of google to use the Google Sheets service. On the interface side, I choose the Reactjs framework because of its ease of development and convenience.

After completing the project, I realized I have learned a lot of things like analytical and problem-solving skills with object-oriented thinking.Then, How to use Firebase to create a file-storage and save upload files on it, setting up CI / CD for automatically building and deploying code on the server, using the ReactJs and Spring framework proficiently and manually make a relatively complete product

Sinh viên thực hiện

Ký và ghi rõ họ tên

**TABLE OF CONTENTS**

[CHƯƠNG I. GIỚI THIỆU CHUNG 1](#_Toc66623654)

[1.1 Mục tiêu 1](#_Toc66623655)

[1.2 Mô tả vấn đề 1](#_Toc66623656)

[1.3 Quy trình nghiệp vụ hệ thống 1](#_Toc66623657)

[1.4 Yêu cầu chức năng 2](#_Toc66623658)

[CHƯƠNG II : PHÂN TÍCH HỆ THỐNG 3](#_Toc66623659)

[2.1 Phân tích sơ đồ Use Case 3](#_Toc66623660)

[2.1.1 Phân rã Use Case Quản lý vị trí công việc 4](#_Toc66623661)

[2.1.2 Phân rã Use Case Quản lý thông tin ứng viên 6](#_Toc66623662)

[2.1.3 Phân rã Use Case Quản lý thống kê báo cáo 11](#_Toc66623663)

[2.1.4 Phân rã Use Case Quản lý tài khoản nhân sự 13](#_Toc66623664)

[CHƯƠNG III. THIẾT KẾ HỆ THỐNG 16](#_Toc66623665)

**LIST OF FIGURES**

[Figure 1. Sơ đồ Use Case tổng quan 3](#_Toc69826383)

[Figure 2 Sơ đồ phân rã Use Case quản lý vị trí công việc 4](#_Toc69826384)

[Figure 3 Đặc tả Use Case Tạo vị trí công việc 5](#_Toc69826385)

[Figure 4 Đặc tả Use Case Xóa vị trí công việc 5](#_Toc69826386)

[Figure 5 Đặc tả Use Case Cập nhật vị trí công việc 6](#_Toc69826387)

[Figure 6 Đặc tả Use Case Xem vị trí công việc 6](#_Toc69826388)

[Figure 7 Sơ đồ phân rã Use Case Manage Candidate Contacts 7](#_Toc69826389)

[Figure 8 Đặc tả Use Case Tạo thông tin liên hệ 8](#_Toc69826390)

[Figure 9 Đặc tả Use Case Thêm liên hệ bằng google sheets 9](#_Toc69826391)

[Figure 10 Đặc tả Use Case blacklist liên hệ 9](#_Toc69826392)

[Figure 11 Đặc tả Use Case Cập nhật liên hệ 10](#_Toc69826393)

[Figure 12 Đặc tả Use Case Lọc và tìm kiếm liên hệ 11](#_Toc69826394)

[Figure 13 Đặc tả Use Case Xuất file excel 11](#_Toc69826395)

[Figure 14 Đặc tả Use Case Xóa thông tin liên lạc 12](#_Toc69826396)

[Figure 15 Sơ đồ phân rã Use Case quản lý thống kê báo cáo 12](#_Toc69826397)

[Figure 16 Đặc tả Use Case Xem báo cáo thống kê 13](#_Toc69826398)

[Figure 17 Đặc tả Use Case Tìm kiếm và lọc báo cáo thống kê 13](#_Toc69826399)

[Figure 18 Sơ đồ phân rã Use Case quản lý tài khoản nhân sự 14](#_Toc69826400)

[Figure 19 Đặc tả Use Case Xem thông tin tài khoản 15](#_Toc69826401)

[Figure 20 Đặc tả Use Case Tạo tài khoản 15](#_Toc69826402)

[Figure 21 Đặc tả Use Case Xóa tài khoản 16](#_Toc69826403)

[Figure 22 Đặc tả Use Case Cập nhật tài khoản 16](#_Toc69826404)

[Figure 23 Thiết kế quan hệ Contact – Job 17](#_Toc69826405)

[BFigure 24 Thiết kế quan hệ Contact – Status- Account 18](#_Toc69826406)

[Figure 25 Thiết kế quan hệ Account - Role 19](#_Toc69826407)

[Figure 26 Thiết kế quan hệ Contact Position History 20](#_Toc69826408)

[Figure 27 Sơ đồ hoạt động Login 21](#_Toc69826409)

[Figure 28 Sơ đồ hoạt động Tạo vị trí công việc 21](#_Toc69826410)

[*Figure 29* *Sơ đồ hoạt động Xem chi tiết công việc*  22](#_Toc69826411)

[Figure 30 Sơ đồ hoạt động Xóa logic công việc 22](#_Toc69826412)

[Figure 31 Sơ đồ hoạt động Cập nhật thông tin công việc 23](#_Toc69826413)

# CHƯƠNG I. GIỚI THIỆU CHUNG

## 1.1 Mục tiêu

Xây dựng ứng dụng để quản lý các hồ sơ nhân viên ứng tuyển theo từng vị trí được mở tuyển dụng và giúp công ty có thể nắm bắt được nhân sự để dễ dàng chăm sóc cũng như hẹn gặp

## 1.2 Mô tả vấn đề

Hiện nay,càng ngày càng nhiều các công ty được thành lập và phát triển,dẫn đến việc nhu cầu nguồn lực tăng cao,kèm theo đó là nếu như công ty phát triển mạnh và quy mô lên đến hàng nghìn nhân sự thì sẽ gặp khó khăn trong việc quản lý các hồ sơ ứng tuyển của ứng viên theo từng vị trí được mở tuyển dụng.

Để có thể phát triển sự tự động hóa trong việc quản lý nhân sự, em quyết định xây dựng một phần mềm có thể hỗ trợ một công ty có quy mô lớn về nhân sự trong việc tuyển dụng nhân sự và quản lý các thông tin cá nhân của ứng viên theo từng vị trí

## 1.3 Quy trình nghiệp vụ hệ thống

Đầu tiên,nhân viên nhân sự sẽ tạp các vị trí công việc mà công ty đang có.Khi công ty có nhu cầu tuyển dụng 1 vị trí nào đó,nhân viên nhân sự sẽ mở vị trí ứng tuyển của công việc trên website và tiến hành đăng tin tuyển dụng lên các trang tin như Facebook,Topcv,… sau khi ứng viên đó nộp cv thì nhân sự có thể tạo liên hệ của ứng viên đó trên website bằng tay hoặc có thể tổng hợp vào file excel để upload lên website và ứng dụng sẽ ghi nhận các thông tin của ứng viên để quản lý

Sau đó nhân viên nhân sự sẽ thao tác hoàn toàn trên website để lọc các thông tin liên hệ của ứng viên theo các tiêu chí mà công ty cần (ví dụ : công nghệ , vị trí, năm kinh nghiệm,trạng thái chăm sóc,…) và gọi điện liên lạc với các ứng viên đó,khi liên lạc xong sẽ thay đổi trạng thái (đã liên lạc,đã hẹn gặp..) lên ứng dụng để theo dõi.

Ngoài ra,hệ thống có thêm chức năng báo cáo để theo dõi và đánh giá KPI tuyển dụng của nhân viên nhân sự.

## 1.4 Yêu cầu chức năng

* Quản lý được kho contact của ứng viên : Thêm sửa xóa xem 1 contact của ứng viên(sẽ có validate ở cả front end và backend)
* Quản lý được các job đang cần tuyển dụng: Thêm sửa xóa xem,tìm kiếm job theo tiêu chí
* Lọc tìm kiếm được ứng viên theo tiêu chí (công nghệ , vị trí, năm kinh nghiệm, lịch sử chăm sóc, trạng thái, level....) : tạo form tìm kiếm,xử lý logic tìm,phân trang kết quả
* Quản lý blacklist contact
* Báo cáo thống kê các chỉ số KPI tuyển dụng
* Import contact từ google sheet theo định dạng chuẩn
* Xuất dữ liệu ứng viên thành file excel
* Lưu được lịch sử tham gia tuyển dụng cuả ứng viên

# CHƯƠNG II : PHÂN TÍCH HỆ THỐNG

## 2.1 Phân tích sơ đồ Use Case

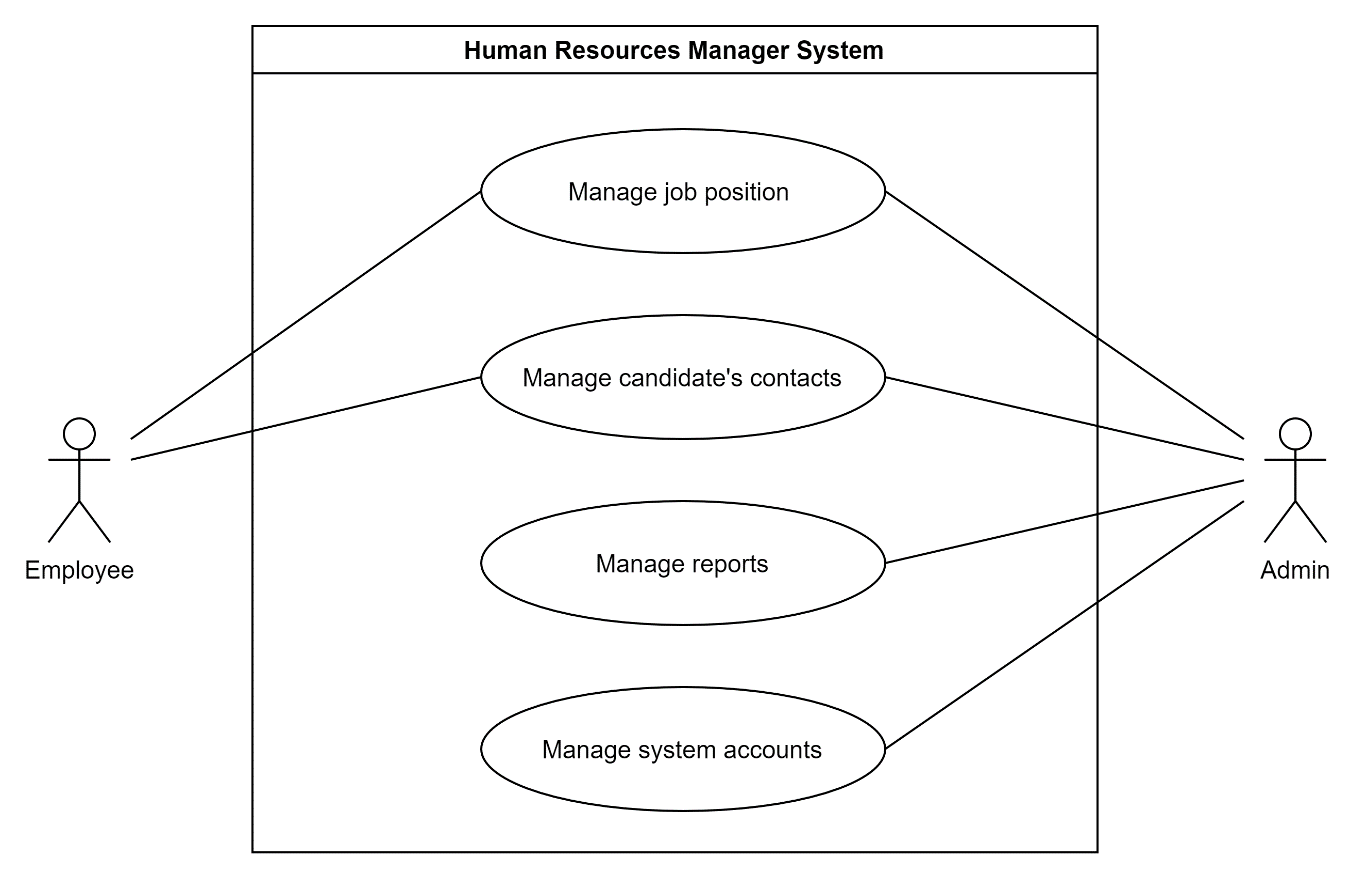
****

Figure . Sơ đồ Use Case tổng quan

Hệ thống sẽ phân quyền thành 2 tác nhân chính là : Trưởng phòng nhân sự và Nhân viên nhân sự.Nhân viên nhân sự sẽ được quản lý về vị trí mở ứng tuyển và thông tin liên hệ của ứng viên.Trưởng phòng nhân sự sẽ quản lý thêm về báo cáo thống kê để đánh giá KPI cho nhân viên nhân sự và quản lý thêm tài khoản của từng nhân viên nhân sự

### 2.1.1 Phân rã Use Case Quản lý vị trí công việc

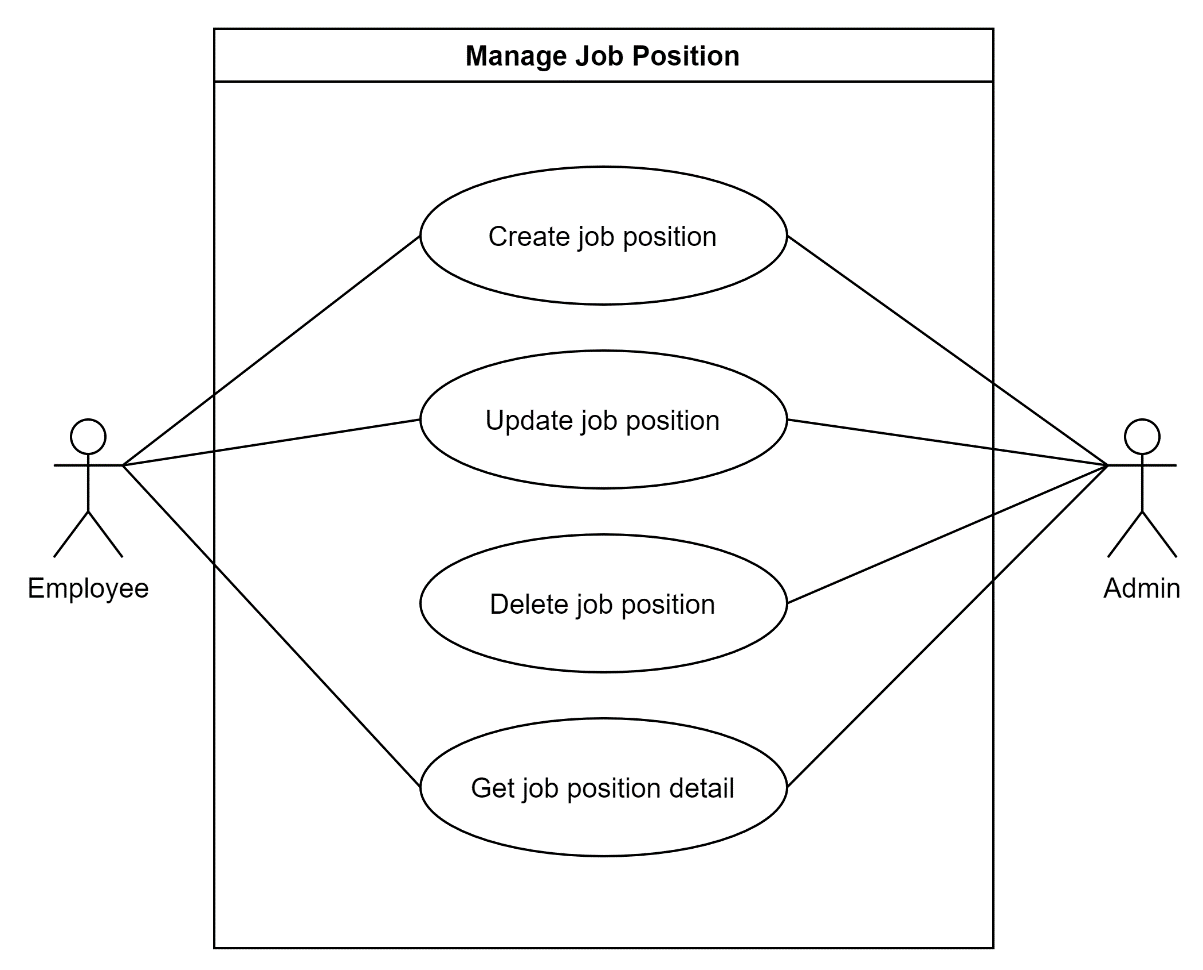


Figure Sơ đồ phân rã Use Case quản lý vị trí công việc

Nhân viên nhân sự và Trường phòng đều có mọi quyền thêm sửa xóa xem các vị trí công việc mà công ty đang có nhu cầu tuyển dụng.

Việc cập nhật vị trí công việc có thể chỉnh sửa thông tin cũng như chỉnh sửa trạng thái công việc có đang mở tuyển dụng hay không

**Đặc tả Use Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC011 | | **UC Name** | Tạo vị trí công việc |
| **Actor** | Nhân viên nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Nhân viên | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Nhân viên nhân sự | Yêu cầu tạo vị trí công việc | |
| 2 | Hệ thống | Hiện form điền thông tin | |
| 3 | Nhân viên nhân sự | Điền thông tin : | |
| Tên công việc | |
| Cấp bậc | |
| Mô tả | |
| Trạng thái mở ứng tuyển | |
| 4 | Hệ thống | Lưu thông tin vị trí công việc và phản hồi | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 3a | Hệ thống | Thông báo lỗi nếu một số thông tin quan trọng bị để trống hoặc không đúng mẫu | |
| **Post condition** | None | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC014 | | **UC Name** | Xóa vị trí công việc |
| **Actor** | Trưởng phòng nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Trưởng phòng | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Trưởng phòng nhân sự | Yêu cầu xem chi tiết nhật vị trí công việc | |
| 2 | Hệ thống | Hiển thị chi tiết thông tin công việc | |
| 3 | Trưởng phòng nhân sự | Yêu cầu xóa vị trí công việc | |
| 4 | Hệ thống | Xóa logic vị trí công việc trong hệ thống | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 4a | Hệ thống | Thông báo lỗi nếu bản ghi đã được xóa | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Tạo vị trí công việc

Figure Đặc tả Use Case Xóa vị trí công việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC013 | | **UC Name** | Cập nhật vị trí công việc |
| **Actor** | Nhân viên nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Nhân viên | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Nhân viên nhân sự | Yêu cầu xem chi tiết nhật vị trí công việc | |
| 2 | Hệ thống | Hiển thị chi tiết thông tin công việc | |
| 3 | Nhân viên nhân sự | Yêu cầu cập nhật thông tin vị trí | |
| 4 | Hệ thống | Hiển thị form thông tin với default values là thông tin trước đó | |
| 3 | Nhân viên nhân sự | Điền form thông tin | |
| 6 | Hệ thống | Cập nhật thông tin mới | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 6a | Hệ thống | Thông báo lỗi nếu một số thông tin quan trọng bị để trống hoặc không đúng mẫu | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Cập nhật vị trí công việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC012 | | **UC Name** | Xem vị trí công việc |
| **Actor** | Nhân viên nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Nhân viên | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Nhân viên nhân sự | Yêu cầu xem chi tiết vị trí công việc: có phân trang và cấu hình phân trang | |
| 2 | Hệ thống | Hiện thông tin chi tiết vị trí công việc | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 2a | Hệ thống | Thông báo không có vị trí công việc | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Xem vị trí công việc

### 2.1.2 Phân rã Use Case Quản lý thông tin ứng viên

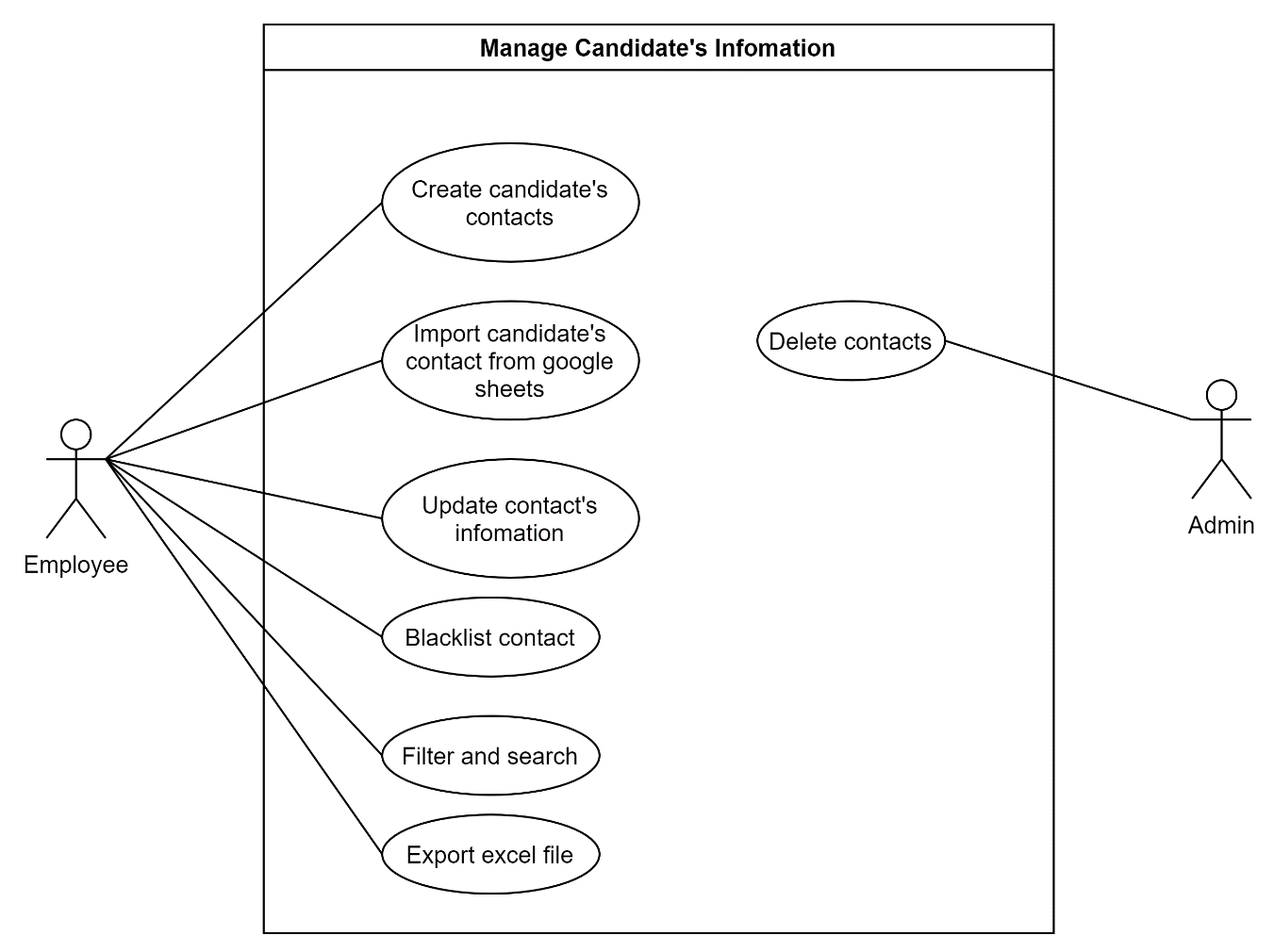


Figure Sơ đồ phân rã Use Case Manage Candidate Contacts

Nhân viên nhân sự có thể tạo mới thông tin ứng viên bằng tay trên giao diện hoặc có thể điền vào thông qua google sheets và hệ thống sẽ đọc và thêm mới dữ liệu đó.Sau khi chăm sóc các ứng viên thì có thể ghi lại phản hồi và hẹn gặp ứng viên đó,có thể cập nhật mới thông tin hoặc thay đổi trạng thái chăm sóc.

Nếu muốn xóa ứng viên thì bắt buộc phải đề xuất tới trưởng phòng hoặc có thể chuyển trạng thái ứng viên đó vào danh sách blacklist

**Đặc tả Use Case Quản lý thông tin ứng viên**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC020 | | **UC Name** | Tạo thông tin liên hệ |
| **Actor** | Nhân viên nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Nhân viên | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Nhân viên nhân sự | Yêu cầu tạo thông tin liên hệ | |
| 2 | Hệ thống | Hiện form điền thông tin | |
| 3 | Nhân viên nhân sự | Điền thông tin : | |
| Tên ứng viên | |
| Số điện thoại | |
| Email | |
| Địa chỉ | |
| Vị trí ứng tuyển | |
| Năm kinh nghiệm | |
| Link cv | |
| File cv | |
| 4 | Hệ thống | Lưu thông tin liên hệ và phản hồi | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 3a | Hệ thống | Thông báo lỗi nếu một số thông tin quan trọng bị để trống hoặc không đúng mẫu | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Tạo thông tin liên hệ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC002 | | **UC Name** | Thêm liên hệ bằng google sheets |
| **Actor** | Nhân viên nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Nhân viên | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Nhân viên nhân sự | Yêu cầu Import Sheets | |
| 2 | Hệ thống | Hiện input link google sheets | |
| 3 | Nhân viên nhân sự | Copy link sheets trên thanh url và paste vào input hệ thống | |
| 4 | Hệ thống | Đọc file sheets và lưu vào hệ thống | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 4a | Hệ thống | Thông báo lỗi nếu sheets chưa được public link | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Thêm liên hệ bằng google sheets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC023 | | **UC Name** | Blacklist liên hệ |
| **Actor** | Nhân viên nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Nhân viên | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Nhân viên nhân sự | Yêu cầu xem chi tiết thông tin | |
| 2 | Hệ thống | Hiển thị chi tiết thông tin | |
| 3 | Nhân viên nhân sự | Yêu cầu blacklist liên hệ | |
| 4 | Hệ thống | Thay đổi trạng thái liên hệ thành Blacklisted | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 4a | Hệ thống | Thông báo lỗi nếu liên hệ đã bị blacklist | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case blacklist liên hệ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC022 | | **UC Name** | Cập nhật thông tin |
| **Actor** | Nhân viên nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Nhân viên | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Nhân viên nhân sự | Yêu cầu xem chi tiết thông tin | |
| 2 | Hệ thống | Hiển thị chi tiết thông tin | |
| 3 | Nhân viên nhân sự | Yêu cầu cập nhật thông tin vị trí | |
| 4 | Hệ thống | Hiển thị form thông tin với default values là thông tin trước đó | |
| 5 | Nhân viên nhân sự | Điền form thông tin | |
| 6 | Hệ thống | Cập nhật thông tin mới | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 3a | Nhân viên nhân sự | Thay đổi trạng thái chăm sóc của ứng viên | |
| 5a | Hệ thống | Thông báo lỗi nếu một số thông tin quan trọng bị để trống hoặc không đúng mẫu | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Cập nhật liên hệ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC024 | | **UC Name** | Lọc và tìm kiếm thông tin |
| **Actor** | Nhân viên nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Nhân viên | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Nhân viên nhân sự | Yêu cầu lọc hoặc tìm kiếm thông tin | |
| 2 | Hệ thống | Hiển thị form chọn tiêu chí lọc (Công việc,Năm kinh nghiệm,Trạng thái,…) | |
| 3 | Nhân viên nhân sự | Chọn các tiêu chí mong muốn | |
| 4 | Hệ thống | Hiển thị danh sách liên lạc phù hợp | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 5a | Hệ thống | Thông báo không có thông tin liên lạc phù hợp | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Lọc và tìm kiếm liên hệ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC025 | | **UC Name** | Xuất file excel |
| **Actor** | Nhân viên nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Nhân viên | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Nhân viên nhân sự | Yêu cầu lọc thông tin liên hệ | |
| 2 | Hệ thống | Hiển thị form chọn tiêu chí lọc (Công việc,Năm kinh nghiệm,Trạng thái,…) | |
| 3 | Nhân viên nhân sự | Chọn các tiêu chí mong muốn | |
| 4 | Hệ thống | Hiển thị danh sách liên lạc phù hợp | |
| 5 | Nhân viên nhân sự | Yêu cầu xuất file danh sách đã lọc | |
| 6 | Hệ thống | Tiến hành xuất file excel | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 4a | Hệ thống | Thông báo không có thông tin liên lạc phù hợp | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Xuất file excel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC026 | | **UC Name** | Xóa vị liên lạc |
| **Actor** | Trưởng phòng nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Trưởng phòng | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Trưởng phòng nhân sự | Yêu cầu xem chi tiết thông tin liên lạc | |
| 2 | Hệ thống | Hiển thị chi tiết thông tin liên lạc | |
| 3 | Trưởng phòng nhân sự | Yêu cầu xóa liên lạc | |
| 4 | Hệ thống | Xóa logic thông tin liên lạc trong hệ thống | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 4a | Hệ thống | Thông báo lỗi nếu bản ghi đã được xóa | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Xóa thông tin liên lạc

### 2.1.3 Phân rã Use Case Quản lý thống kê báo cáo

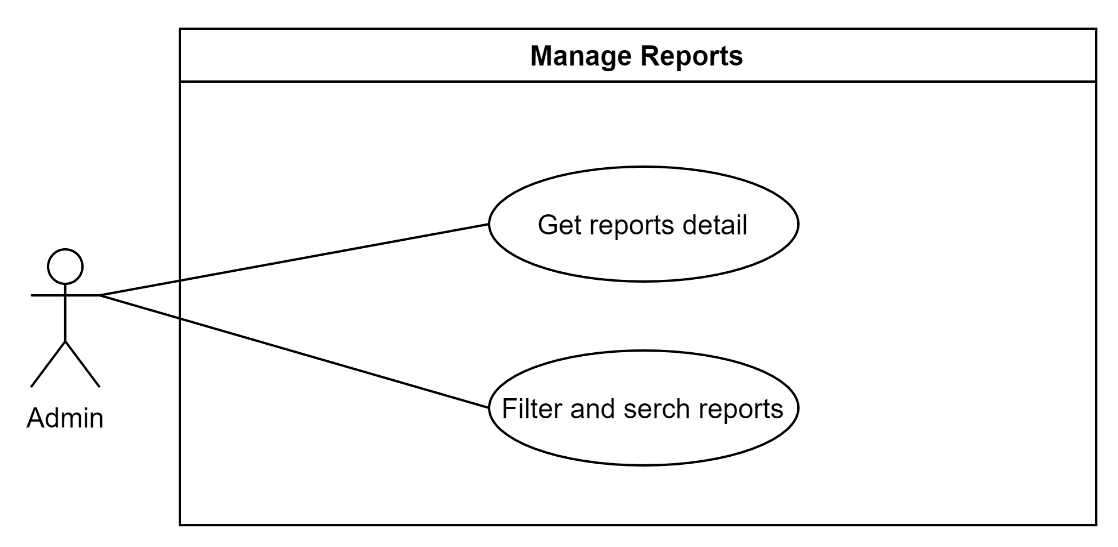


Figure Sơ đồ phân rã Use Case quản lý thống kê báo cáo

Trưởng phòng có thể xem báo cáo,lọc và tìm kiếm báo cáo theo yêu cầu như : báo cáo theo vị trí tuyển dụng,nhân viên tuyển dụng,thời gian theo tháng năm quý,…

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC031 | | **UC Name** | Xem báo cáo thống kê |
| **Actor** | Trưởng phòng nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Trưởng phòng | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Trưởng phòng nhân sự | Yêu cầu xem chi tiết báo cáo | |
| 2 | Hệ thống | Hiển thị chi tiết báo cáo | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 2a | Hệ thống | Thông báo lỗi nếu bản ghi không tồn tại | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Xem báo cáo thống kê

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC032 | | **UC Name** | Tìm kiếm và lọc báo cáo |
| **Actor** | Trưởng phòng nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Trưởng phòng | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Trưởng phòng nhân sự | Yêu cầu tìm kiếm và lọc báo cáo | |
| 2 | Hệ thống | Hiển thị form lọc báo cáo | |
| 3 | Trưởng phòng nhân sự | Chọn tiêu chí mong muốn lọc | |
| 4 | Hệ thống | Hiển thị danh sách báo cáo theo tiêu chí | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 4a | Hệ thống | Thông báo lỗi nếu không có báo cáo theo tiêu chí | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Tìm kiếm và lọc báo cáo thống kê

### 2.1.4 Phân rã Use Case Quản lý tài khoản nhân sự

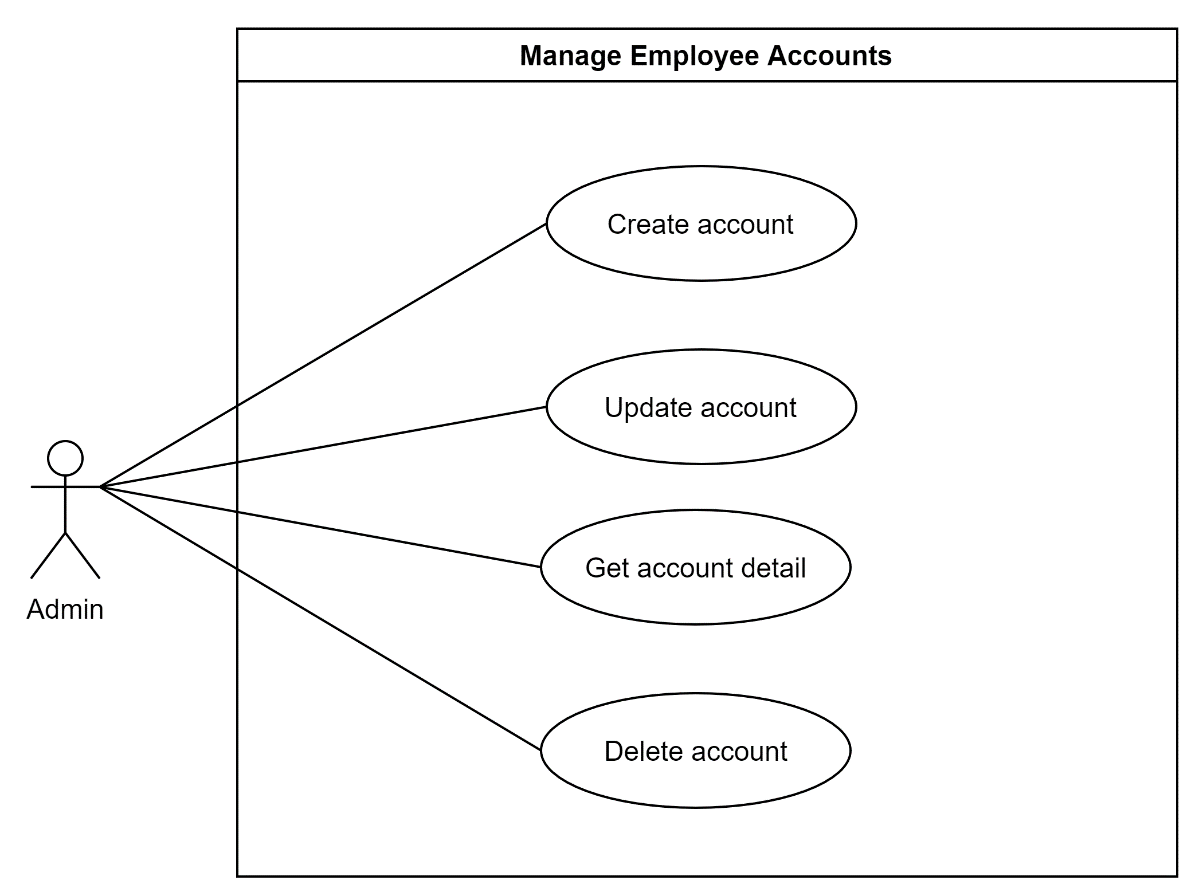


Figure Sơ đồ phân rã Use Case quản lý tài khoản nhân sự

Khi có nhân viên nhân sự mới thì trưởng phòng sẽ tiến hành tạo tài khoản để nhân viên đó có thể thao tác trên hệ thống

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC042 | | **UC Name** | Xem thông tin tài khoản |
| **Actor** | Trưởng phòng nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Trưởng phòng | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Trưởng phòng nhân sự | Yêu cầu xem thông tin tài khoản | |
| 2 | Hệ thống | Hiển thị thông tin chi tiết tài khoản | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 2a | Hệ thống | Thông báo lỗi nếu không tồn tại tài khoản | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Xem thông tin tài khoản

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC041 | | **UC Name** | Tạo tài khoản |
| **Actor** | Trưởng phòng nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Trưởng phòng | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Trưởng phòng nhân sự | Yêu cầu tạo mới tài khoản | |
| 2 | Hệ thống | Hiển thị form thông tin tài khoản | |
| 3 | Trưởng phòng nhân sự | Điền thông tin: Tên chủ tài khoản Số điện thoại Địa chỉ Vị trí công việc Username tài khoản Mật khẩu Loại tài khoản | |
| 4 | Hệ thống | Thông báo thành công | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 4a | Hệ thống | Thông báo lỗi nếu các trường điền vào không hợp lệ | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Tạo tài khoản

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC044 | | **UC Name** | Xóa tài khoản |
| **Actor** | Trưởng phòng nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Trưởng phòng | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Trưởng phòng nhân sự | Yêu cầu xem chi tiết tài khoản | |
| 2 | Hệ thống | Hiển thị chi tiết thông tin tài khoản | |
| 3 | Trưởng phòng nhân sự | Yêu cầu xóa tài khoản | |
| 4 | Hệ thống | Xóa logic tài khoản trong hệ thống | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 4a | Hệ thống | Thông báo lỗi nếu không tồn tại tài khoản | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Xóa tài khoản

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC Code** | UC043 | | **UC Name** | Cập nhật tài khoản |
| **Actor** | Trưởng phòng nhân sự | | | |
| **Precondition** | Đăng nhập tài khoản cấp quyền Trưởng phòng | | | |
| **Main flow of event (success)** | **#** | **Doer** | **Action** | |
| 1 | Trưởng phòng nhân sự | Yêu cầu xem chi tiết tài khoản | |
| 2 | Hệ thống | Hiển thị thông tin chi tiết tài khoản | |
| 3 | Trưởng phòng nhân sự | Yêu cầu cập nhật tài khoản | |
| 4 | Hệ thống | Hiển thị form thông tin với default values là thông tin trước đó | |
| 5 | Trưởng phòng nhân sự | Cập nhật thông tin và xác nhận | |
| 6 | Hệ thống | Thông báo thành công | |
| **Alternative flow of event** | # | **Doer** | **Action** | |
| 6a | Hệ thống | Thông báo lỗi nếu các trường điền vào không hợp lệ | |
| **Post condition** | None | | | |

Figure Đặc tả Use Case Cập nhật tài khoản

# CHƯƠNG III. DESIGN SYSTEM

**3.1 Design Database**

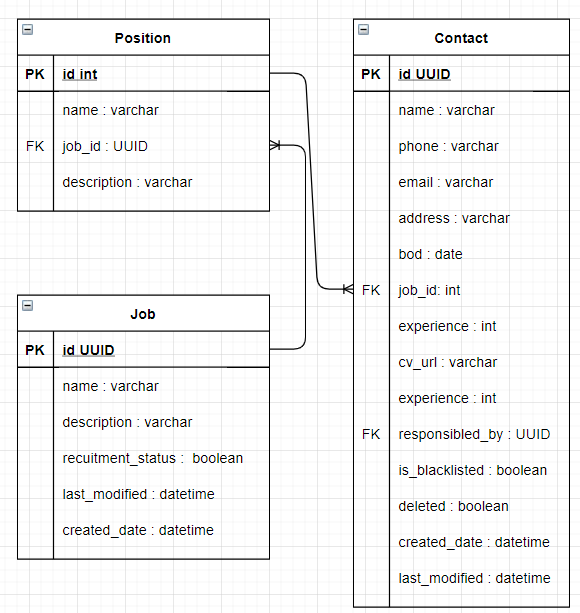
****

Figure Thiết kế quan hệ Contact – Job

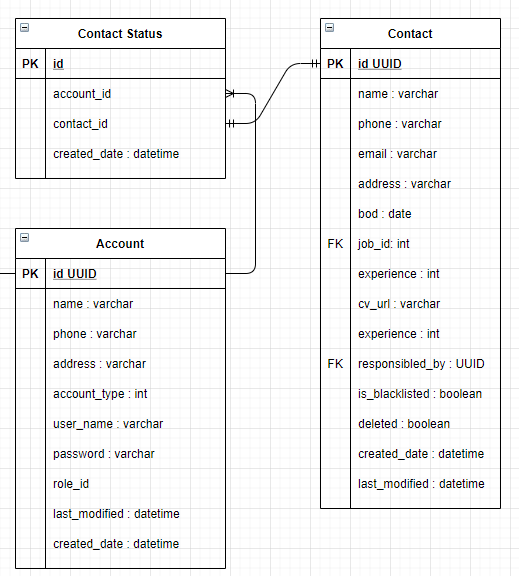
****

Figure Thiết kế quan hệ Contact – Status- Account

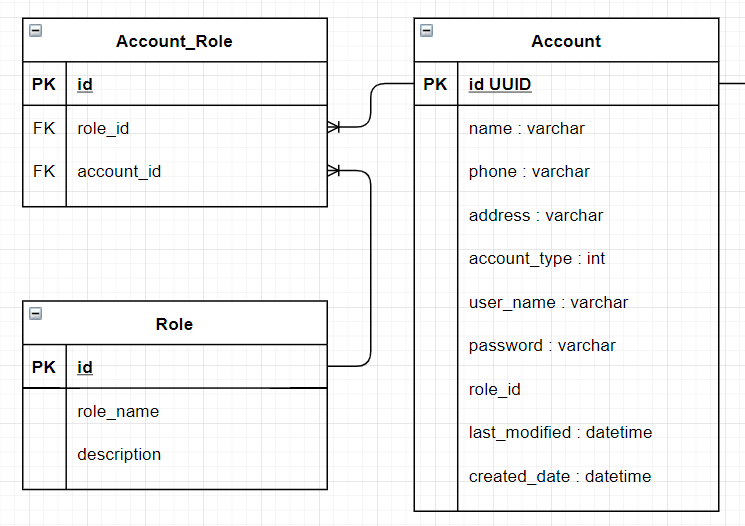
**

Figure Thiết kế quan hệ Account - Role

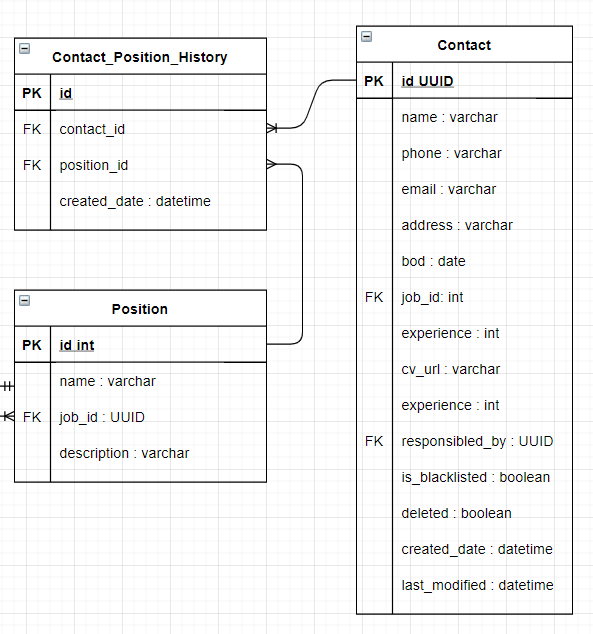
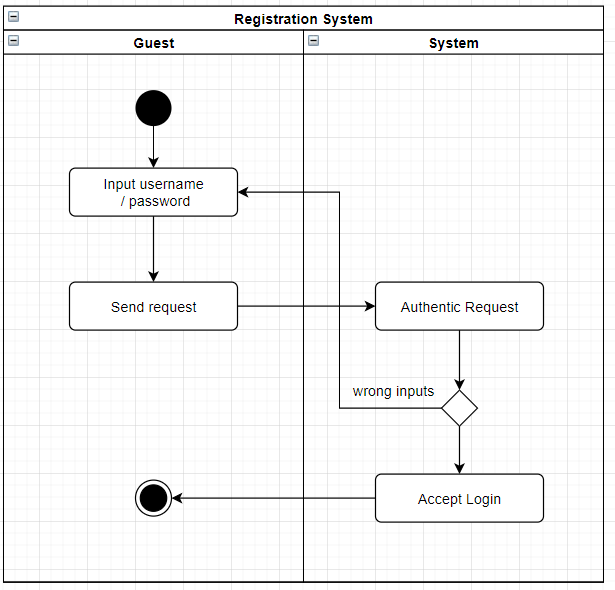
****

Figure Thiết kế quan hệ Contact Position History

** 3.2 Thiết kế hàm thông qua biểu đồ Activity**

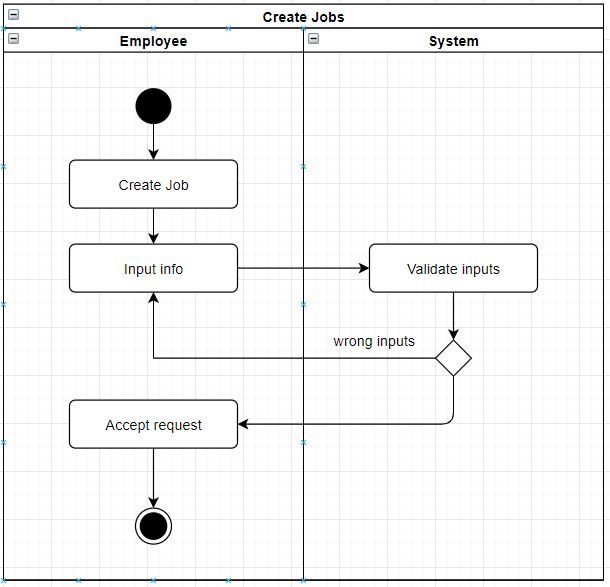
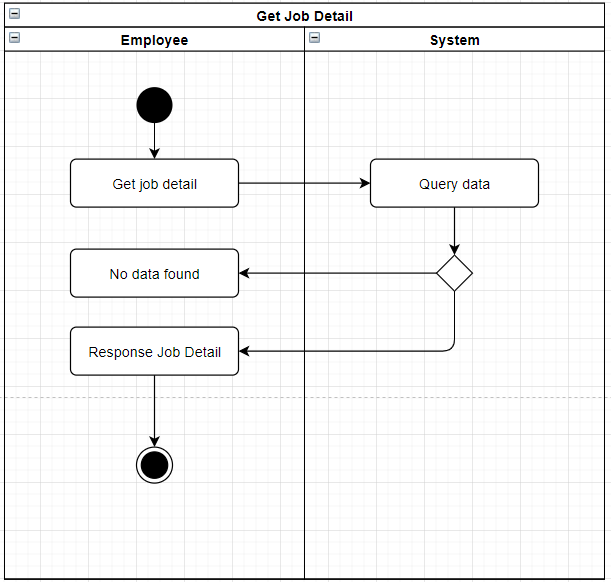
Figure Sơ đồ hoạt động Login

Figure Sơ đồ hoạt động Tạo vị trí công việc

**

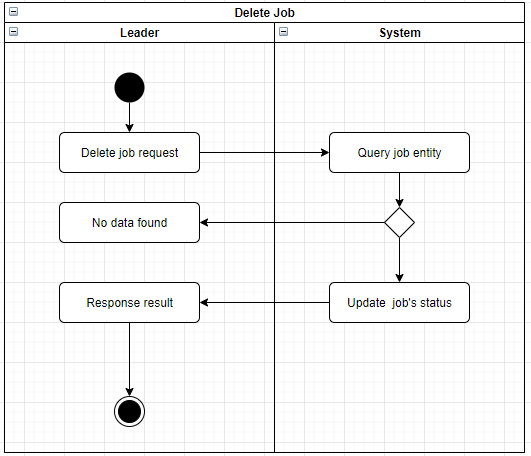
*Figure 29* *Sơ đồ hoạt động Xem chi tiết công việc *

Figure Sơ đồ hoạt động Xóa logic công việc

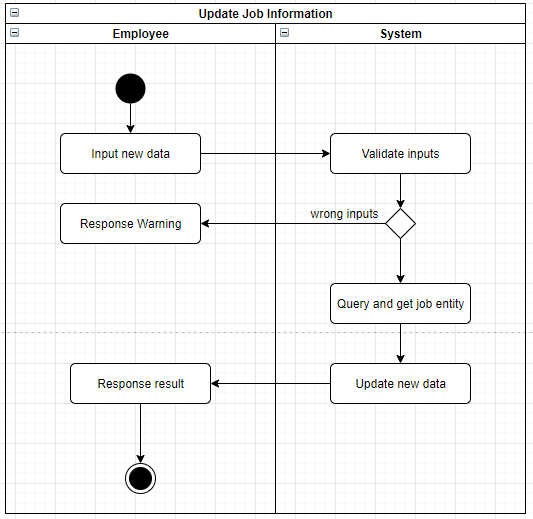
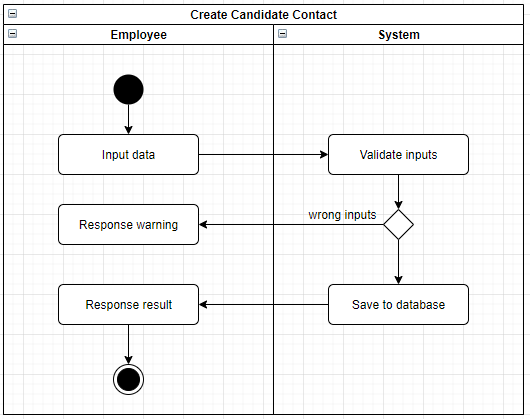
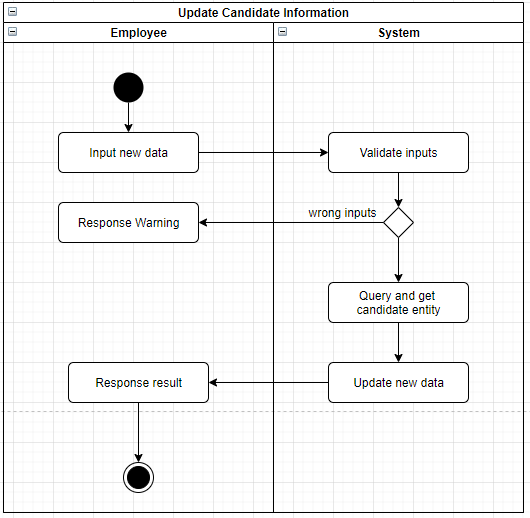
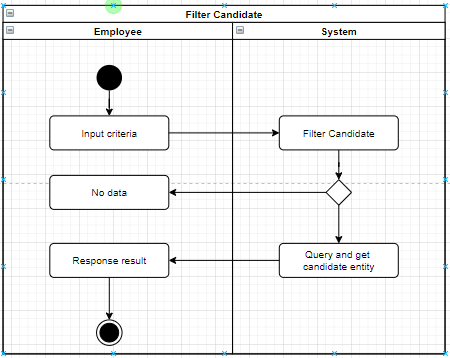
**

Figure Sơ đồ hoạt động Cập nhật thông tin công việc

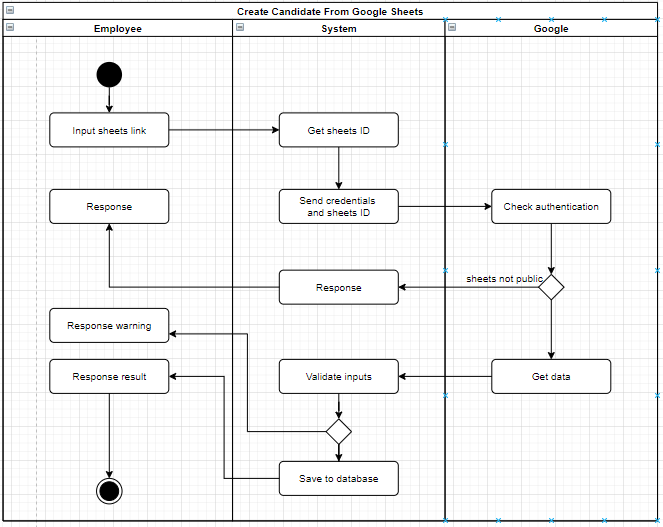
**

*Bảng 6.6 Sơ đồ hoạt động Tạo mới thông tin ứng viên*

**

*Bảng 6.7 Sơ đồ hoạt động Cập nhật thông tin ứng viên*

*Bảng 6.8 Sơ đồ hoạt động Lọc và tìm kiếm ứng viên*

**

*Bảng 6.9 Sơ đồ hoạt động Import ứng viên thông qua google sheets*

# CHƯƠNG IV . IMPLEMENTATION

## 4.1 [Selecting technologies to build back-end server](#_heading=h.1hmsyys)

Relied on system with large user data and high load capacity, the system needs a low-cost backend that automatically expands as the number of users grows and shrinks as the number of users decreases. The system used AWS (Amazon Web Services) technology with the above features and used NodeJS language

Amazon web services is a comprehensive development cloud platform provided by [Amazon.com](http://amazon.com/). Web services are sometimes called cloud services or remote computing services.



Amazon web services operate on cloud computing (cloud computing), the distribution of on-demand computing resources - everything from applications to data centers to the internet pay for use.

AWS services used in the system include:

### 4.1.1 Amazon EC2

Provide the AWS cloud computing platform. Using AWS EC2 requires no hardware investment, so it helps develop and deploy applications faster. You can use AWS EC2 to create as many or as few servers as you need, security and network configuration, memory management. Amazon EC2 allows scaling up or down to handle changes in requests or spikes, reducing the need for traffic forecasts

### 4.1.2 Unlimited resource storage with Amazon S3

****

As object storage secure scalability, security, and high scalability in the cloud. It is used to store and retrieve any amount of data anytime and anywhere. Amazon S3 makes it easy for users to store data as much as possible and access it whenever needed. Using a tree-like storage structure allows users to organize folders and access them easily. Besides, S3 users will be limited to buying hardware and paying for unused memory. Users only pay for the bandwidth they use. The main purpose of S3 for backup and storage, there are different storage packages suitable for different types of data, long-term storage, or short-term storage. It also allows storing the version, can record changes, and roll back to the previous version when needed. Amazon S3 provides Rest API and SOAP services for storing, retrieving, and managing data. Besides, users can also manage resources via the AWS console

Amazon S3 stores data in the form of objects, and objects are stored in folders called buckets. For each bucket, you can control access to each bucket, such as creating, deleting, and displaying bucket objects. To store objects in Amazon S3, you need to upload the file to the specified bucket.

### 4.1.3 Lambda function

Lambda function is a great service of recently launched AWS. It allows users to select the environment and run their code directly in that environment without installing a computer like EC2. The Lambda function needs to be called via the API. It acts as the business logic of the system, receiving information, and retrieving information from other modules, reading and writing data to the MongoDB or S3 database.

Cost: Lambda function only charges based on the time it is called, so it is more economical

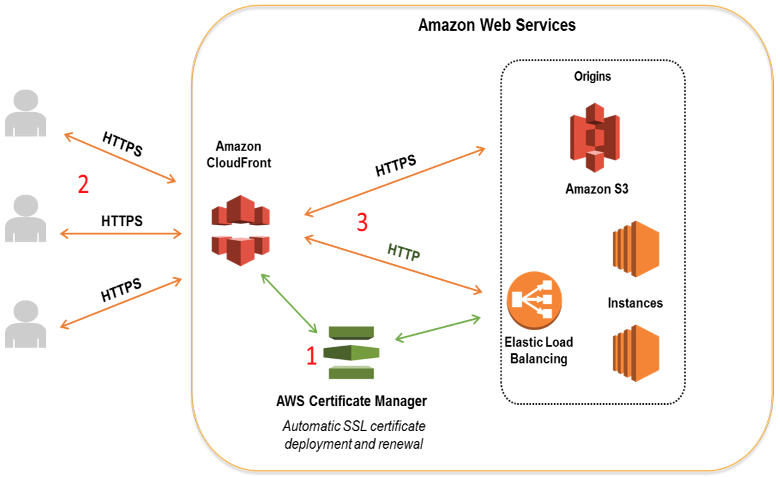
### 4.1.4 API Gateway

API Gateway is a service that provides external methods (routers) that can access system resources such as API, S3. With flexible and customizable settings, this service can be restricted by permission, token key, or IP Address.

### 4.1.5 SAM ( Serverless Application Model )

SAM is an AWS service that helps deploy, install, configure system services and modules automatically on the AWS environment via the configuration file template.yaml. It is very convenient in developing, testing, and deploying with different parameter settings for each environment.

### 4.1.6. Cloudfront

******

**Figure 4.1 Cloudfront Architecture**

Amazon CloudFront is a global content distribution service (CDN) that helps distribute resources from S3 at a cheaper cost, increasing video data security and increasing transfer speeds. CloudFront also provides regional boundary buffer locations within the standard incentive framework, ensuring a consistently high rate of buffers across the globe.

Cost: The price of Amazon CloudFront is very simple and easy to calculate. With CloudFront, there is no need to pay in advance or pay a fixed platform fee, no long-term commitment, no separate fee for flexible content, and no professional service is required to get started. Using AWS sources like Amazon S3 or Elastic Load Balancing, only have to pay for storage costs, not data transfer between these services and CloudFront.

### 4.1.7. Java – Spring framework back end construction language

**Java** is a [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [programming language](https://en.wikipedia.org/wiki/Programming_language) that is designed to have as few implementation [dependencies](https://en.wikipedia.org/wiki/Dependency_(computer_science)) as possible. It is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_language) programming language intended to let [application developers](https://en.wikipedia.org/wiki/Application_developer) *write once, run anywhere* (WORA),[[16]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-16) meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need for recompilation.[[17]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-design_goals-17) Java applications are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of the underlying [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture)



The **Spring Framework** is an [application framework](https://en.wikipedia.org/wiki/Application_framework) and [inversion of control](https://en.wikipedia.org/wiki/Inversion_of_control) [container](https://en.wikipedia.org/wiki/Servlet_container) for the [Java platform](https://en.wikipedia.org/wiki/Java_platform). The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the [Java EE](https://en.wikipedia.org/wiki/Java_EE) (Enterprise Edition) platform. Although the framework does not impose any specific [programming model](https://en.wikipedia.org/wiki/Programming_model), it has become popular in the Java community as an addition to the [Enterprise JavaBeans](https://en.wikipedia.org/wiki/Enterprise_JavaBeans) (EJB) model. The Spring Framework is [open source](https://en.wikipedia.org/wiki/Open-source_software).

Spring makes building web applications fast and hassle-free. By removing much of the boilerplate code and configuration associated with web development, you get a modern web programming model that streamlines the development of server-side HTML applications, REST APIs, and bidirectional, event-based systems.

**Advantages:**

* Reduces the time spent on development and increases the overall efficiency of the development team.
* Helps to autoconfigure all components for a production-grade Spring app.
* Facilitates the creation and testing of Java-based applications by providing a default setup for unit and integration tests.
* Helps to avoid all the manual work of writing boilerplate code, annotations, and complex XML configurations.
* Comes with embedded HTTP servers like Jetty and Tomcat to test web applications.
* The integration of Spring Boot with the Spring ecosystem which includes Spring Data, Spring Security, Spring ORM, and Spring JDBC is easy.
* Provides many plugins that developers can use to work with embedded and in-memory databases smoothly and readily.
* Allows for easily connecting with database and queue services like Oracle, PostgreSQL, MySQL, MongoDB, Redis, Solr, ElasticSearch, Rabbit MQ, ActiveMQ, and many more.
* Provides admin support – you can manage via remote access to the application.
* Eases the dependency and comes with Embedded Servlet Container.
* Offers flexibility in configuring XML configurations, Java Beans, and Database Transaction.
* Offers easy access to Command Line Interface which makes the development and testing of Spring Boot apps built with Java or Groovy agile.

## 4.2 [Options technology for building front-end interfaces](#_heading=h.41mghml)

**React** (also known as **React.js** or **ReactJS**) is an [open-source](https://en.wikipedia.org/wiki/Open-source), [front end](https://en.wikipedia.org/wiki/Front_end_and_back_end), [JavaScript library](https://en.wikipedia.org/wiki/JavaScript_library)[[3]](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-react-3) for building [user interfaces](https://en.wikipedia.org/wiki/User_interfaces) or UI components. It is maintained by [Facebook](https://en.wikipedia.org/wiki/Facebook) and a community of individual developers and companies.React can be used as a base in the development of [single-page](https://en.wikipedia.org/wiki/Single-page_application) or mobile applications. However, React is only concerned with state management and rendering that state to the [DOM](https://en.wikipedia.org/wiki/Document_Object_Model), so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality

**ADVANTAGES OF REACTJS**

* **Intuitive**  
  ReactJS is extremely intuitive to work with and provides interactivity to the layout of any UI. Plus, it enables fast and quality assured application development that in turn saves tome for both - clients and developers.
* **Declarative**  
  ReactJS enables significant data changes that result in automatic alteration in the selected parts of user interfaces. Owing to this progressive functionality, there is no additional function that you need to perform to update your user interface.
* **Provides Reusable Components**  
  ReactJS provides reusable components that developers have the authority to reuse and create a new application . Reusability is exactly like a remedy for developers. This platform gives the developers the authority to reuse the components build for some other application having the same functionality. Thereby, reducing the development effort and ensuring a flawless performance.
* **JavaScript library**

A strong blend of JavaScript and HTML syntax is always used, which automatically simplifies the entire process of writing code for the planned project. The JS library consists several functions including one that converts the HTML components into required functions and transforms the entire project so that it is easy to understand.

* **Components Support**

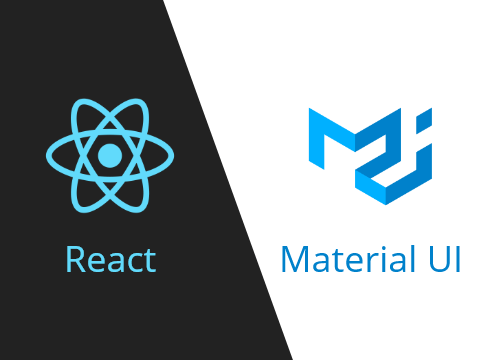
ReactJS is a perfect combination of JavaScript and HTML tags. The usage of the HTML tags and JS codes, make it easy to deal with a vast set of data containing the document object model. During this time, ReactJS works as a mediator which represents the DOM and assists to decide which component needs changes to get the exact results.

* **SEO-friendly**

React JS was introduced after immense research and improvements by Facebook. Naturally, it stands out from the crowd and allows developers to build amazing, SEO-friendly user interfaces across browsers and engines.

* **Proficient Data Binding**

ReactJS trails one-way data binding. This means that absolutely anyone can track all the changes made to any particular segment of the data. This is a symbol of its simplicity.



**Material-UI**

Material-UI is simply a library that allows us to import and use different components to create a user interface in our React applications. This saves a significant amount of time since the developers do not need to write everything from scratch. Material-UI widgets are heavily inspired by Google’s principles on building user interfaces. It is, therefore, easy for developers to build visually-appealing applications

For programmers, the name boostrap is probably too familiar to beautify your website. In reactjs, everything is attributed to a component so everything supporting it is also attributable to the component. Boostrap is no exception. Boostrap also has component systems to support reactjs projects. But today I would like to introduce another library. Looks like it was born to accompany the react to the present time is like that. Right from the first introduction lines, it is advised to know react before learning about it. That is Material-UI. So this article I will guide you to use it in your projects using reactjs. Material-UI also gives you quite a few components to create a web page more quickly, without having to sit around css bit by bit. Very convenient, just install this library and use it. Depending on our requirements that custom according to intelligence.

## 4.3 Database selection and database installation

**MySQL** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). Its name is a combination of "My", the name of co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, and "[SQL](https://en.wikipedia.org/wiki/SQL)", the abbreviation for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). A [relational database](https://en.wikipedia.org/wiki/Relational_database) organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an [operating system](https://en.wikipedia.org/wiki/Operating_system) to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

**Google Cloud SQL** is a MySQL database that lives in Google Cloud and doesn’t require any product establishment and support since it is provided as help by Google Cloud. So, it’s maintained, oversaw, and managed by Google Cloud itself. Like an ordinary MySQL database, it allows you to create, change, arrange, and use a public database.

The storage space was expanded from 10 gigabytes to 100s of gigabytes. It was changed with the goal that it presently permits the cases up to 16 gigabytes of RAM. With more RAM, clients would now be able to enjoy better performance. No concurrent replication was included as a component that consolidates the advantages of repeated and non-recreated databases. There are different highlights that make Google Cloud an ideal answer for little to medium-sized database remaining burdens

**Advantages :**

* **Great costs** – Around 0.020 USD per GB/month with the Regional class and 0.007 USD per GB/month with the Cold line class
* **Diverse stockpiling classes for every need**: Regional (visit use), near line (inconsistent use) and Cold line (long haul stockpiling)
* **High availability:** Google Cloud Storage offers 99.999999999% (11 9’s) of the strength of articles over a given year. This implies information endures even in case of the concurrent loss of two circles
* **Numerous areas accessible to store your information:** North America, South America, Europe, Asia, and Australia. Simple to coordinate with other Google Cloud Services like Kubernetes Engine, App Engine, or Compute Engine
* The “Reassure” tab in the documentation enables you to choose simple SDKs. It’s inconceivably helpful for designers.
* Extraordinary compared to other free layers in the business: $300 free acknowledge beginning for any GCP item during the main year. Subsequently, 5 GB of Storage allowed utilizing for eternity

## 4.4 External services

**The Google Sheets API** lets you read and modify any aspect of a spreadsheet. Spreadsheets have many settings that let you craft beautiful and functional sheets, so the API has many settings too. The API offers two main ways to interact with the spreadsheet:

* Reading/writing cell values only
* Reading/writing any aspect of the spreadsheet

Both of these collections are straightforward to use, although the spreadsheets collection provides many more options. For more about these collections and their use, see the reference documentation links above or the following developer guides

**Firebase** is a Backend-as-a-Service — BaaS

Firebase Storage provides a simple way to save binary files — most often images, but it could be anything — to Google Cloud Storage **directly from the client**

Firebase Storage has it’s own system of security rules to protect your GCloud bucket from the masses, while granting detailed write privileges to your authenticated clients.