
Report #2: System Design

PROJECT JOBIT

**Introduction to Software Engineering
(CSC13002)**

GVHD: Cô Nguyễn Thị Minh Tuyền

GVTH: Thầy Nguyễn Đức Huy

Group name: Group 18 - Team 32GB

| STT | Họ và Tên | MSSV |
|-----|-----------------------|---------|
| 1 | Trần Phú Nguyễn (NT) | 1612448 |
| 2 | Trần Ngô Anh Nguyên | 1612440 |
| 3 | Nguyễn Lê Thị Mỹ Linh | 1612334 |
| 4 | Trần Thị Lạng | 1612331 |
| 5 | Trần Thị Lý | 1612367 |

Ho Chi Minh City, 25/11/2018

Revision History

| Date | Version | Description | Author |
|--------------|----------------|--------------------|-------------------------|
| <21/11/2018> | <1.0> | <System design> | <Nguyễn Lê Thị Mỹ Linh> |
| <25/11/2018> | <2.0> | <System design> | <Trần Thị Lý> |
| | | | |
| | | | |

Individual Contributions Breakdown

| MSSV | Họ tên | Vai trò | Phần trăm đóng góp |
|----------------|-----------------------|---|--------------------|
| 1612448 | Trần Phú Nguyễn | Project Manager Developer Designer | 100% |
| 1612440 | Trần Ngô Anh Nguyễn | Developer Designer | 90% |
| 1612334 | Nguyễn Lê Thị Mỹ Linh | Developer Designer | 90% |
| 1612331 | Trần Thị Lạng | Developer Tester Designer | 90% |
| 1612367 | Trần Thị Lý | Developer Designer | 90% |

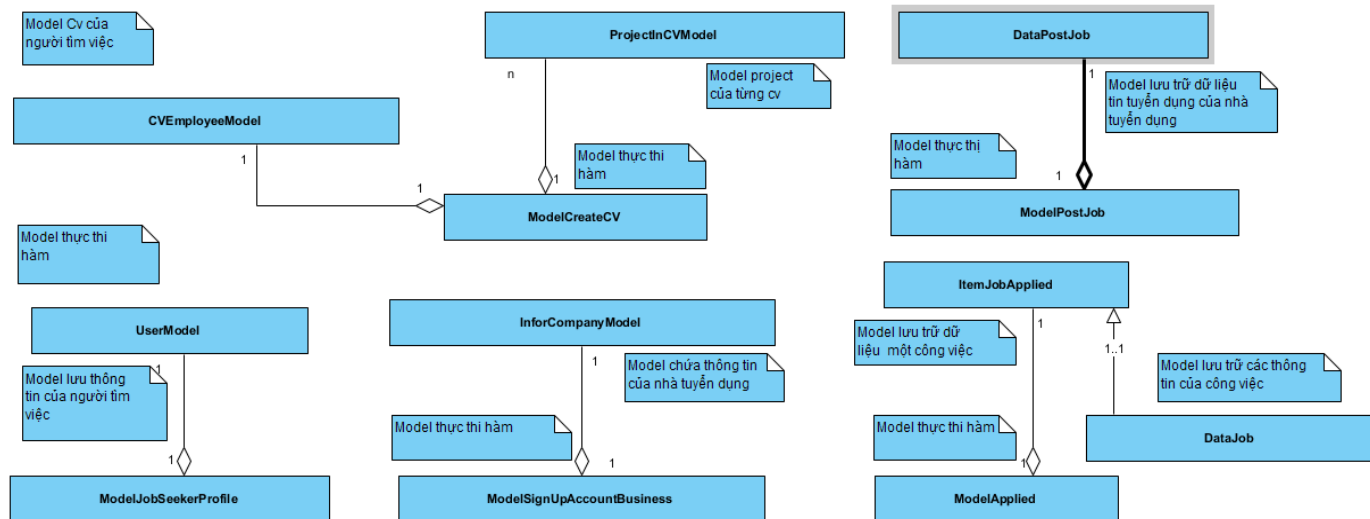
Table of Contents

| | | |
|-----------|---|-----------|
| A. | Class Diagram and Interface Specification..... | 4 |
| 1. | Class Diagram..... | 4 |
| 2. | Data Types and Operation Signatures | 15 |
| B. | System Architecture and System Design | 18 |
| 1. | Architectural Styles..... | 18 |
| 2. | Identifying Subsystems | 18 |
| 3. | Mapping Subsystems to Hardware | 19 |
| 4. | Persistent Data Storage | 19 |
| 5. | Network Protocol..... | 19 |
| 6. | Global Control Flow | 19 |
| 7. | Hardware Requirements..... | 20 |
| C. | Algorithms and Data Structures..... | 21 |
| 1. | Algorithms..... | 21 |
| 2. | Data Structures | 21 |
| 1. | Merging the Contributions from Individual Team Members | 23 |
| 2. | Project Coordination and Progress Report..... | 26 |
| 3. | Plan of Work | 29 |
| 4. | Breakdown of Responsibilities | 32 |
| D. | References..... | 36 |

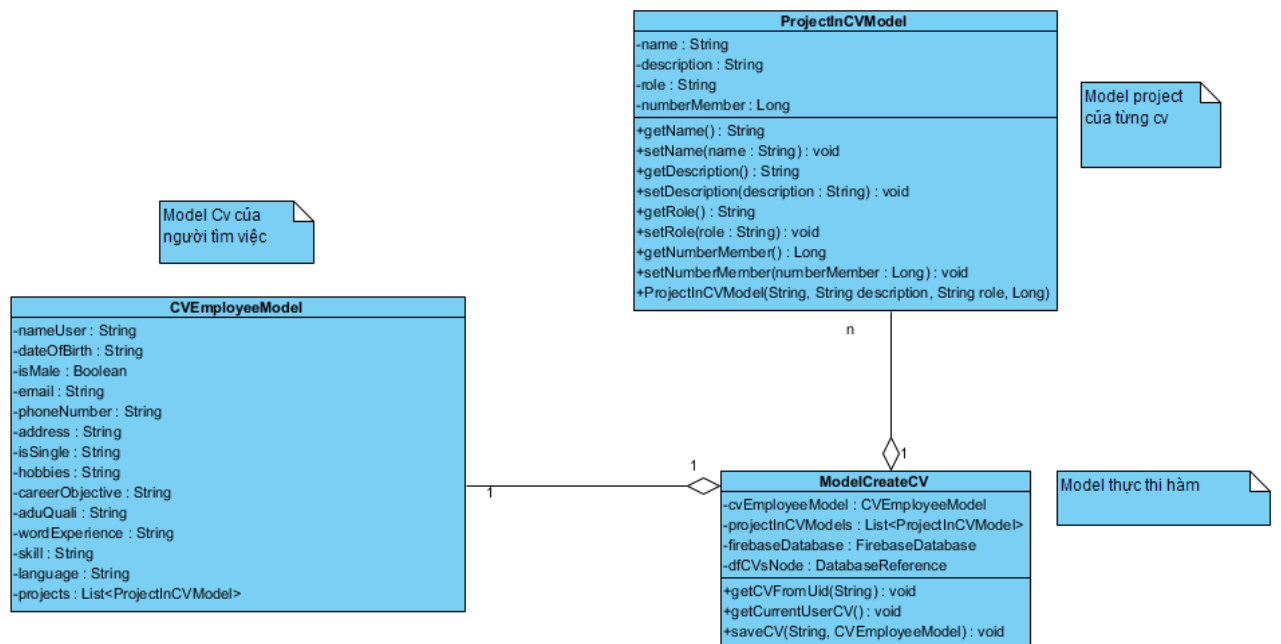
A. Class Diagram and Interface Specification

1. Class Diagram

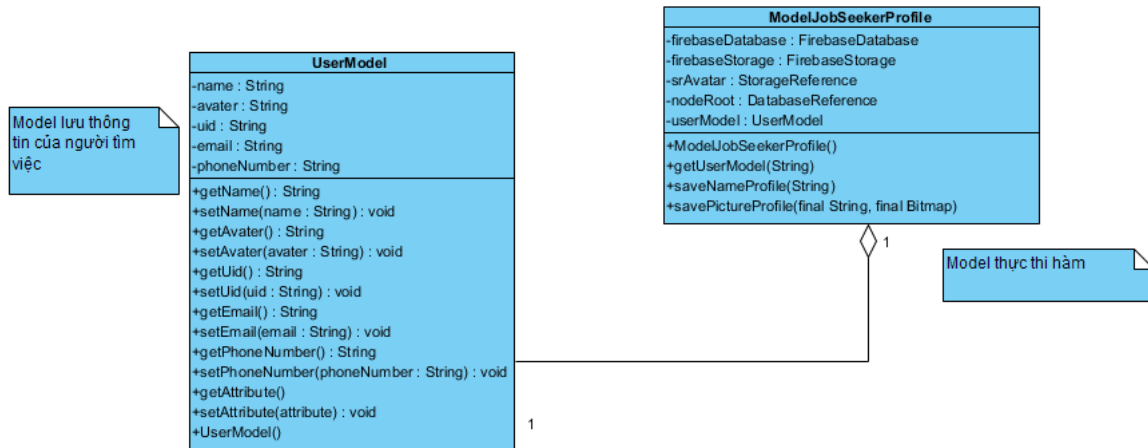
a. Người tìm việc và nhà tuyển dụng



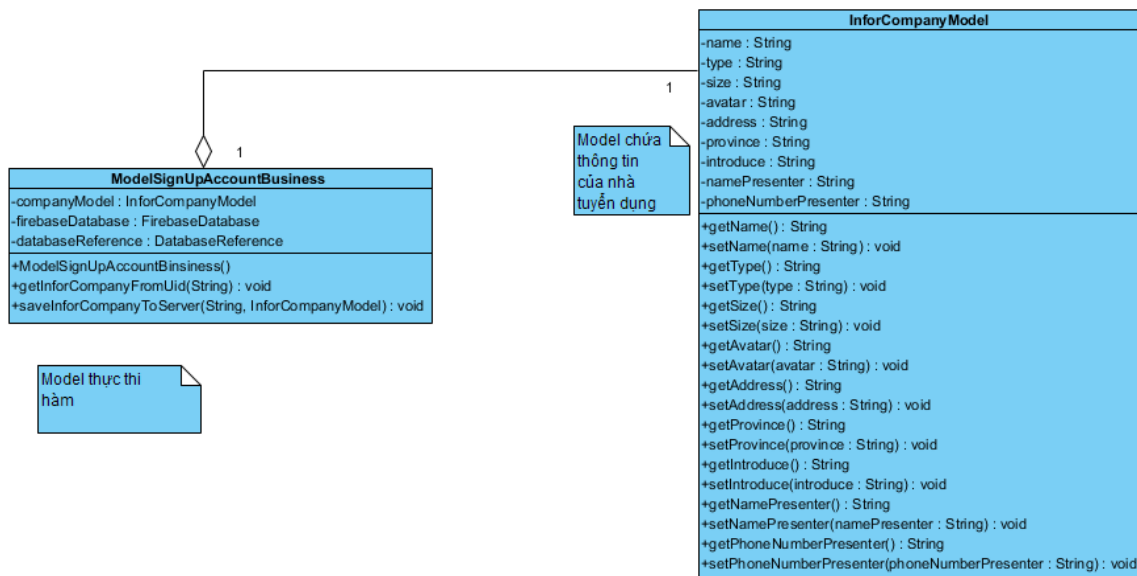
Hình 1. Tổng quan mối quan hệ giữa các class thao tác dữ liệu với Firebase



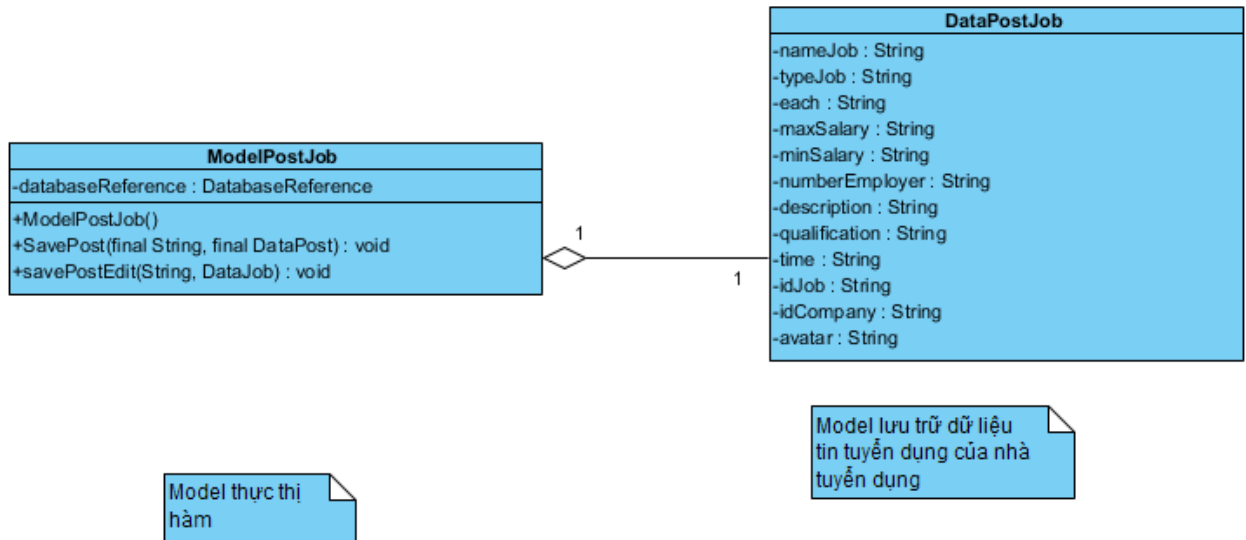
Hình 2. Mối quan hệ giữa class CVEmployeeModel, ProjectInCVModel và ModelCreateCV



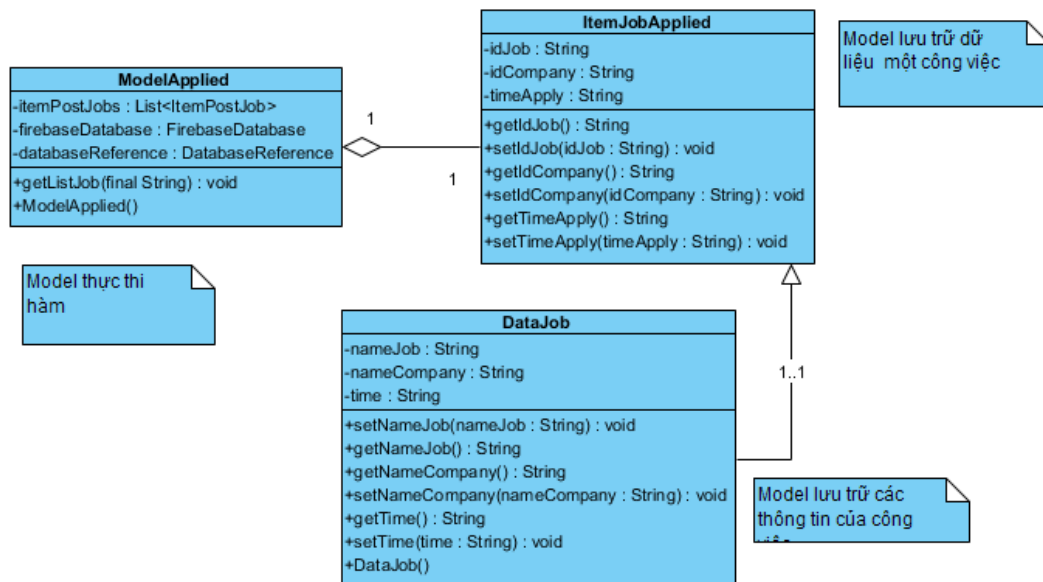
Hình 3. Mối quan hệ giữa class UserModel và ModelJobSeekerProfile



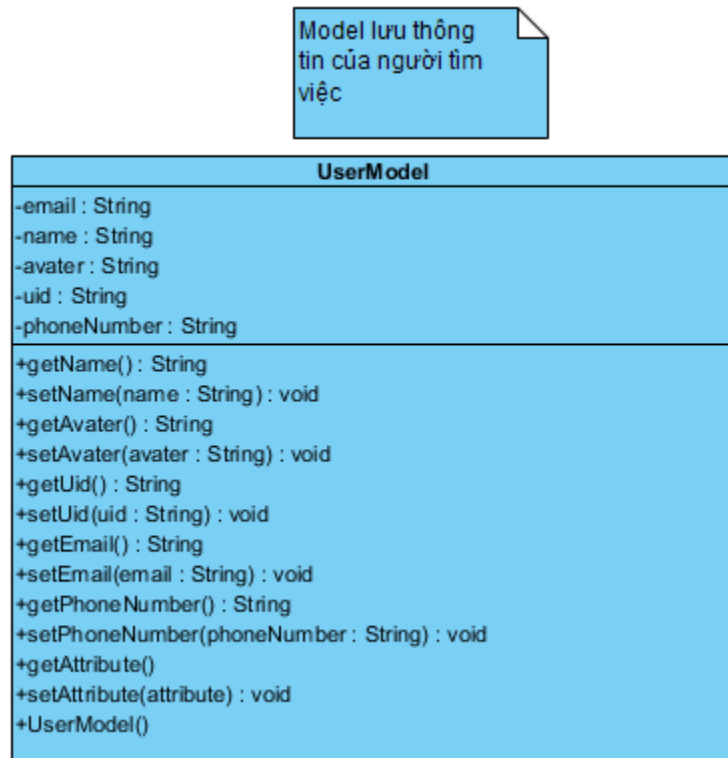
Hình 4. Mối quan hệ giữa ModelSignUpAccountBusiness và InforCompanyModel



Hình 5. Mối quan hệ giữa class ModelPostJob, DataPostJob

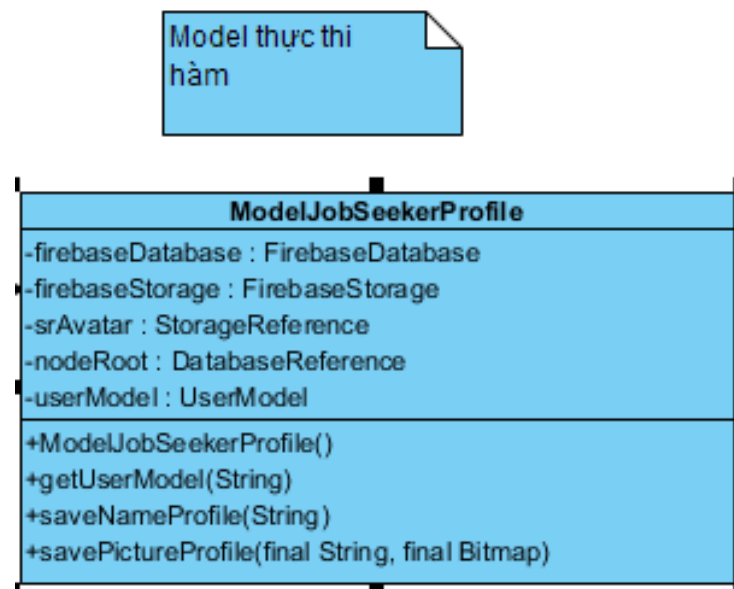


Hình 6. Mối quan hệ giữa class ModelApplied, DataJob và ItemJobApplied



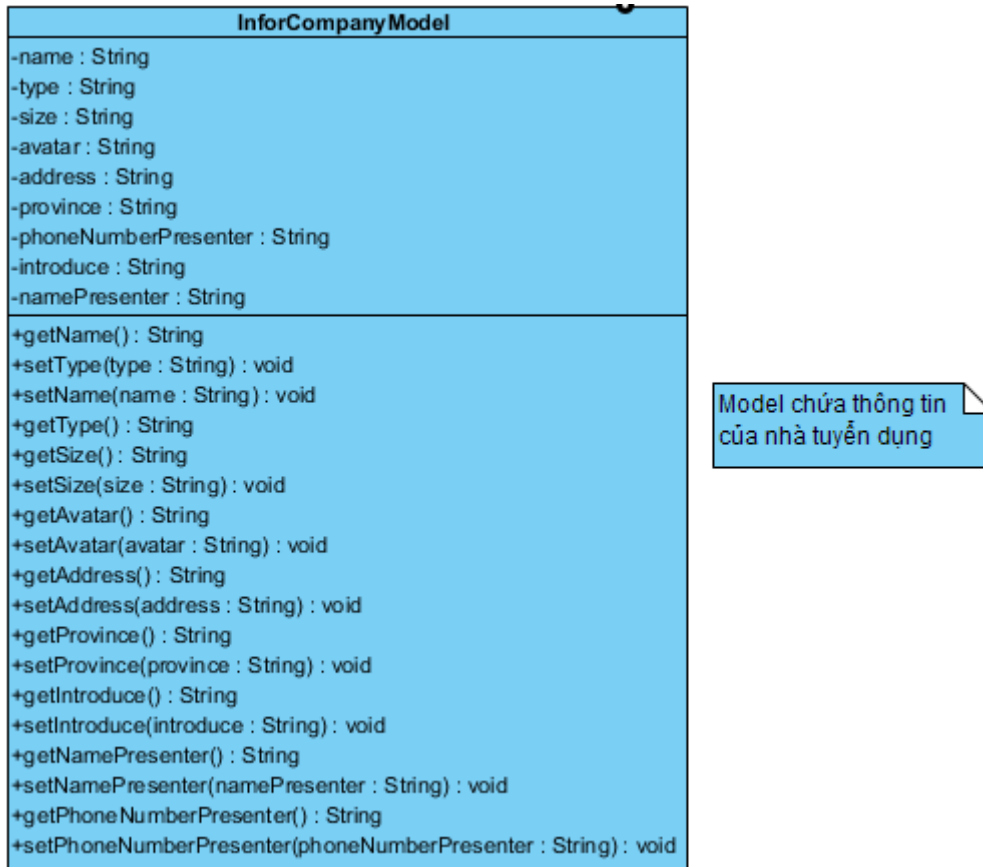
Hình 7. UserModel

- Class UserModel dùng để lưu trữ tất cả các thông tin của người tìm việc



Hình 8. ModelJobSeekerProfile

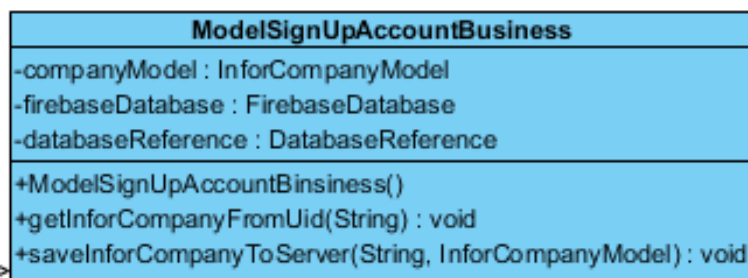
- Class ModelJobSeekerProfile dùng để thực thi các hàm, class này sẽ thực hiện việc lưu trữ thông tin người tìm việc lên Firebase.(hàm saveNameProfile và savePictureProfile)



Hình 9. InfoCompanyModel

- Class Model lưu trữ các thông tin của nhà tuyển dụng, khi nhà tuyển dụng đăng ký một tài khoản thì sẽ cần cung cấp các thông tin trên.

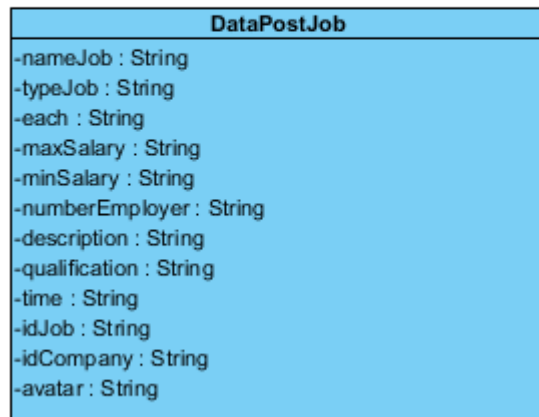
Model thực thi hàm



Hình 10. ModelSignUpAccountBusiness

- Class Model để thực hiện việc lưu dữ liệu của công ty lên Firebase thông qua hàm saveInforCompanyToServer.

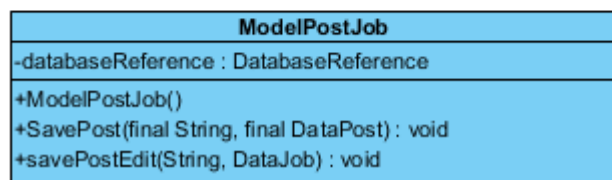
Model lưu trữ dữ liệu
tin tuyển dụng của nhà
tuyển dụng



Hình 11. DataPostJob

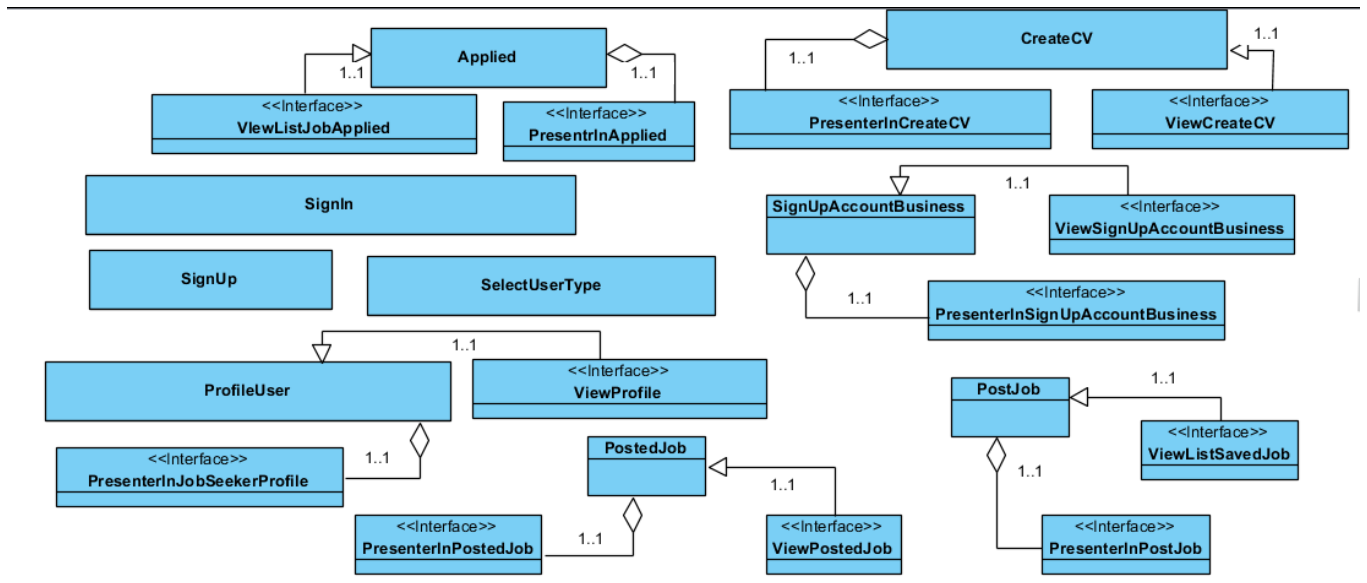
- Class model dùng để lưu trữ các thông tin về công việc mà công ty muốn đăng tuyển

Model thực thi
hàm

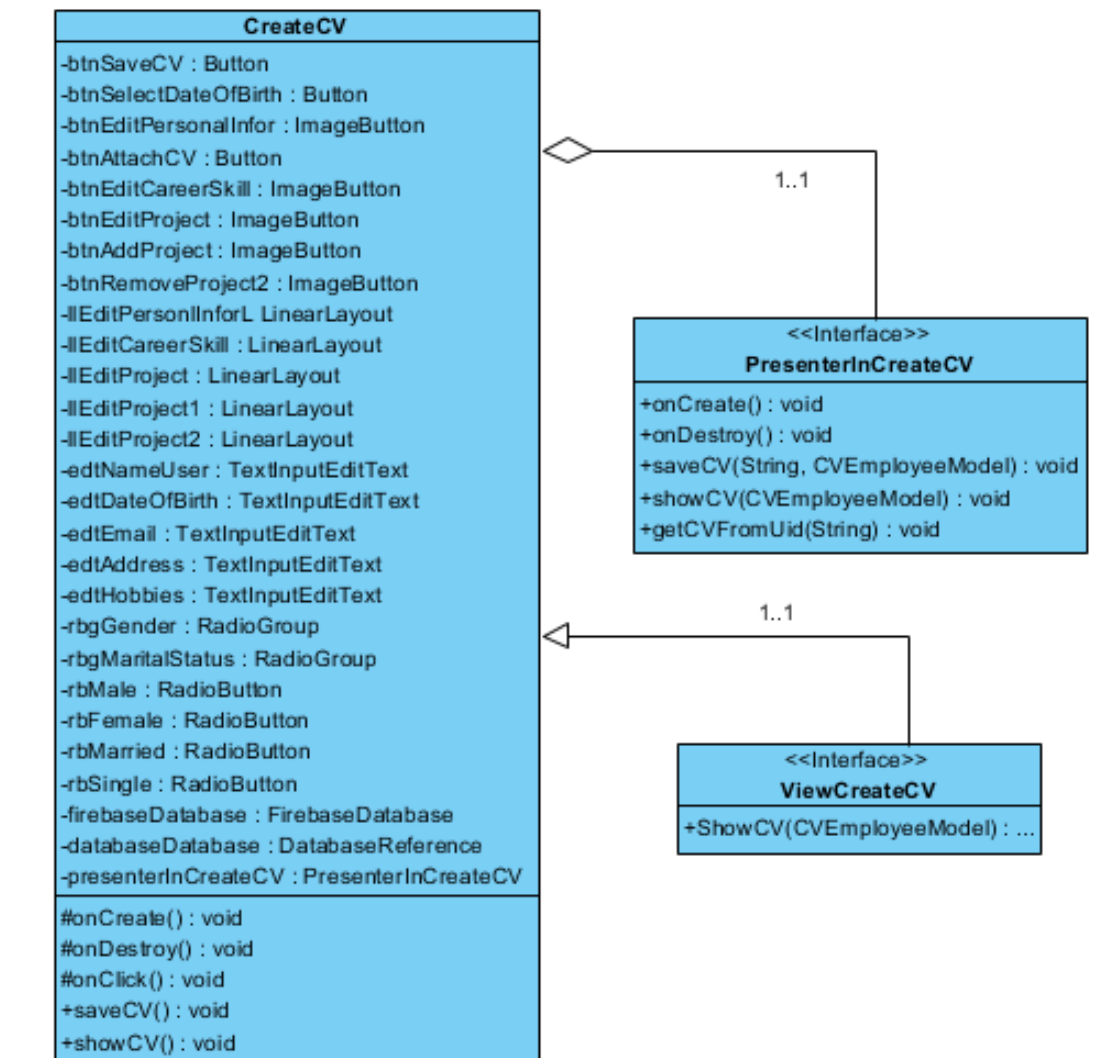


Hình 12. ModelPostJob

- Class ModelPostJob dùng để thực hiện lưu trữ thông tin của công việc lên Firebase thông qua hàm SavePost.

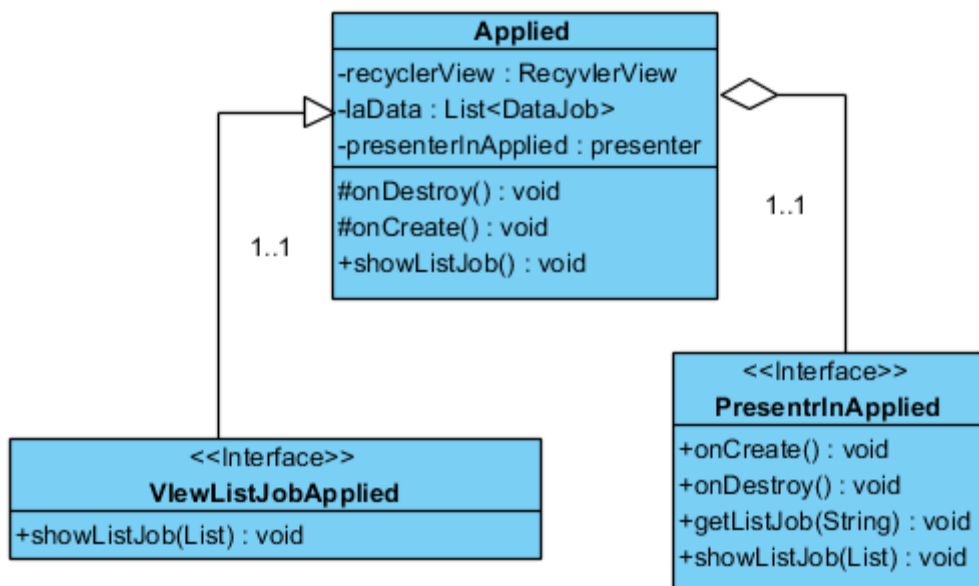


Hình 13. Tổng quan các class xử lý dữ liệu thực hiện với app



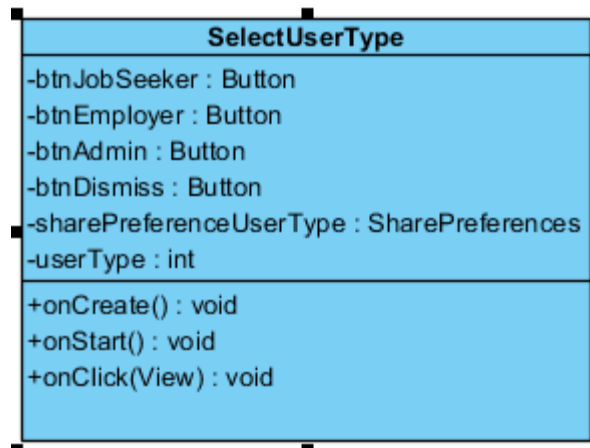
Hình 14. Class CreateCV

- Class tạo cv của người tìm việc, implement từ interface ViewCreateCV, ViewCreateCV có hàm showCV để lấy thông tin của người tìm việc và hiển thị ra thông qua hàm showCV trong class CreateCV.
- Hàm saveCV trong class CreateCV để lưu thông tin của người tìm việc lên Firebase. PresenterInCreateCV dùng làm trung gian để giao tiếp giữa Model và View thông qua hàm saveCV , showCV và getCVFromUid. GetCVFromUid dùng để thao tác với Firebase và lấy thông tin về.



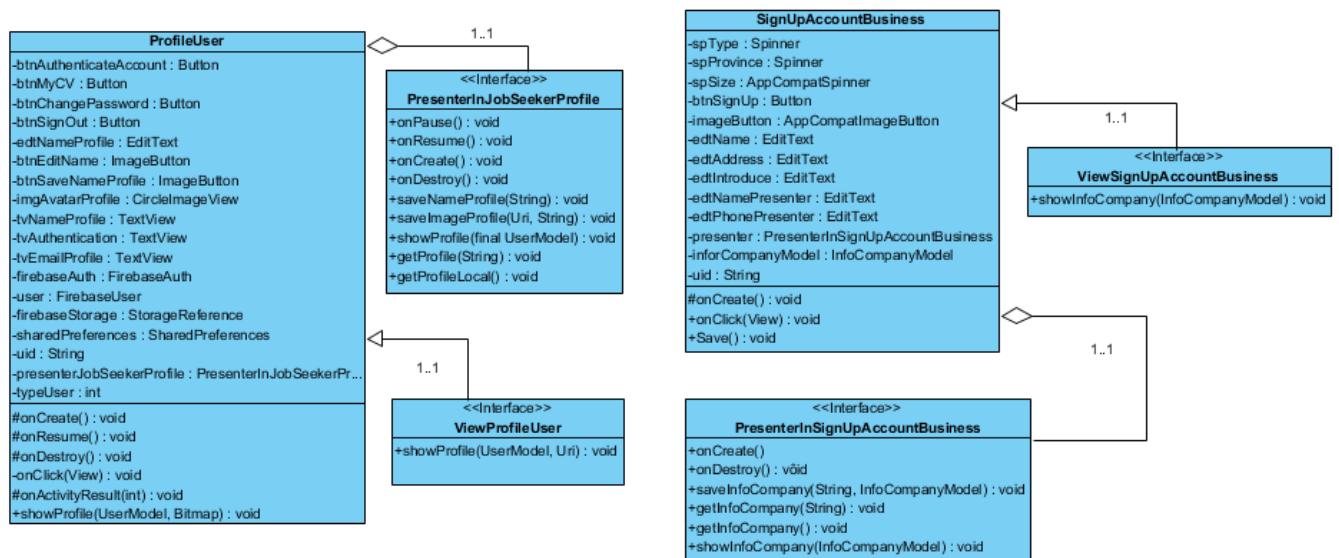
Hình 15. Class Applied

- Class Applied dùng giúp cho người tìm việc có thể apply vào công việc mình mong muốn, implement từ interface ViewListJobApplied, ViewListJobApplied có hàm showListJob để lấy danh sách các công việc mà người tìm việc đã apply và hiển thị ra thông qua hàm showListJob trong class Applied.
- PresenterInApplied dùng làm trung gian để giao tiếp giữa Model và View thông qua hàm showListJob và getListJob. getListJob dùng để thao tác với Firebase và lấy thông tin về. Khi người dùng nhấn Applied thì công việc sẽ được lưu trên Firebase.



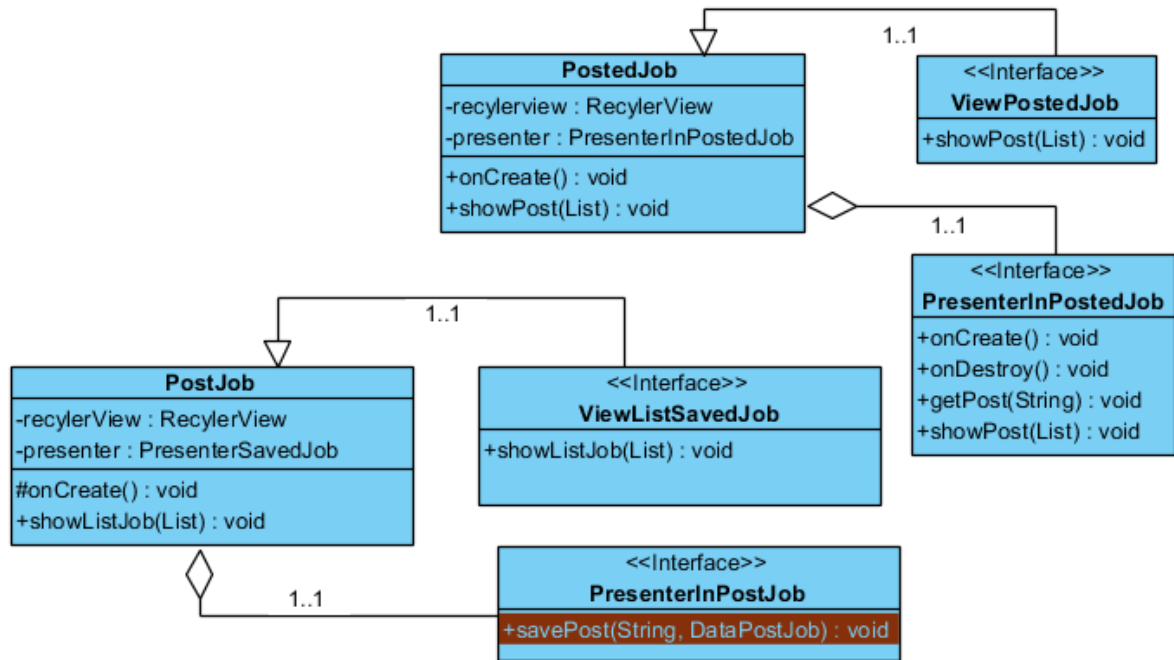
Hình 16. Class UserType

- Class SelectUserType dùng để phân quyền người dùng, là người tìm việc hay nhà tuyển dụng hay là Admin. Mỗi đối tượng có các các năng riêng biệt. Khi một tài khoản chỉ được sử dụng cho một đối tượng.
- Đối với Admin thì chỉ được quyền đăng nhập, không được quyền đăng ký tài khoản



Hình 17. Class ProfileUser và SignUpAccountBusiness

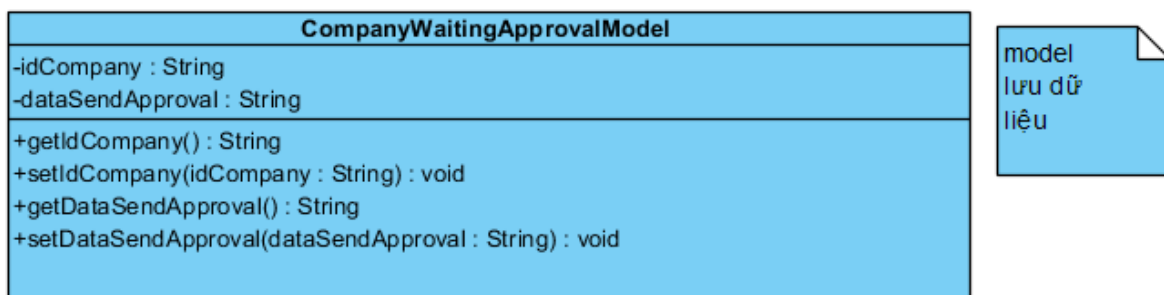
- Đây là 2 class dùng để lưu thông tin người tìm việc và nhà tuyển dụng. Khi nhà tuyển dụng hoặc người tìm việc đăng nhập hoặc đăng ký ứng dụng sẽ yêu cầu người dùng nhập những thông tin cần thiết.
- Các thông tin này được đưa lên Firebase để lưu trữ và quản lý. View và Presenter có chức năng tương tự như những class trên



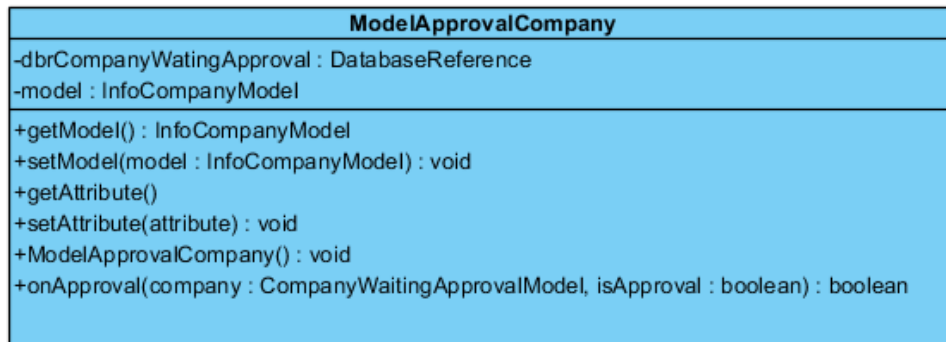
Hình 18. Class PostedJob và PostJob

- Đây là 2 class **PostedJob** và **PostJob**. Các View và Presenter chức năng tương tự các lớp trên dùng để xử lý. 2 class này dùng để lưu trữ dữ liệu tin tuyển dụng của nhà tuyển dụng và lấy dữ liệu trên Firebase về hiện thị lên App.
- Các dữ liệu lấy được từ Firebase dùng để xử lý các chức năng khác như nhà tuyển dụng xem hồ sơ người ứng tuyển nộp vào từng tin đã đăng, sửa thông tin đã đăng...

b. Admin



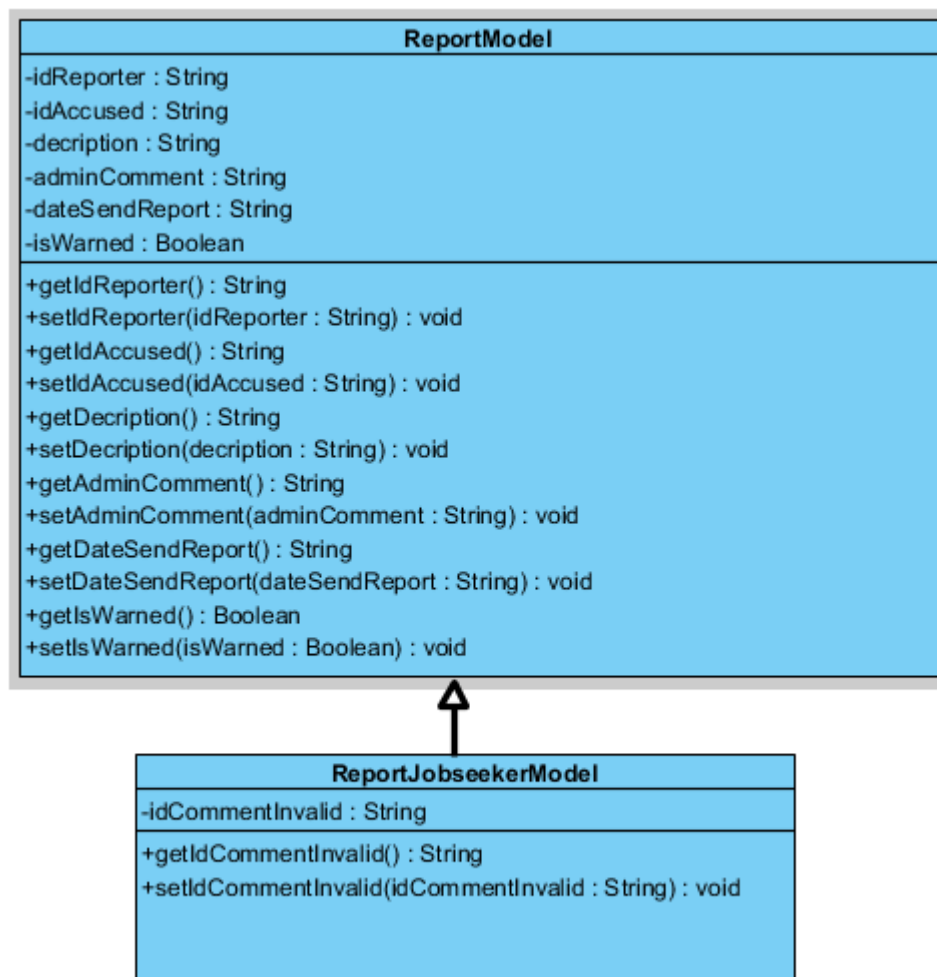
Hình 19. Class model công ty chờ admin duyệt hồ sơ.



Model thực
hiện hàm

Hình 20. Class ModelApprovalCompany

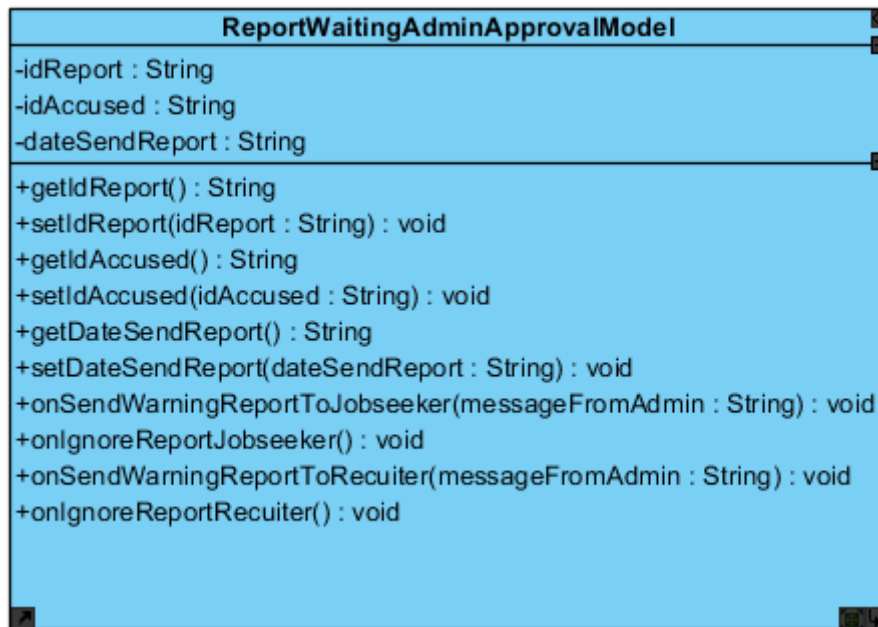
- Class model hồ sơ công ty cần duyệt, class này dùng để thực hiện hàm onApproval (xử lý phê duyệt công ty từ admin).



Hình 21. Class ReportModel và ReportJobSeekerModel

- Class model tổ cáo (bên trên, ReportModel) và model tổ cáo người tìm việc (bên dưới, kế thừa model tổ cáo) . Hai class này là model

lưu dữ liệu.



Hình 22. Class ReportWaitingAdminApprovalModel

- Class model tổ cáo đang chờ duyệt, dùng để lưu thông tin tổ cáo đang chờ admin xử lý, class này có các hàm xử lý dữ liệu.

2. Data Types and Operation Signatures

- User Model:** dùng để lưu trữ thông tin của người tìm việc đã đăng ký tài khoản với hệ thống, được sử dụng như một mẫu cho việc thêm, cập nhật các thông tin trên firebase

Các thuộc tính của lớp:

- Email: chuỗi chứa email của tài khoản (string)
- Name: tên người sử dụng tài khoản (string)
- Avatar: hình ảnh của người tìm việc (string)
- Uid: id của tài khoản (string)
- Phone: số điện thoại người tìm việc (String)

Các phương thức của lớp:

- getName(): lấy tên người sử dụng tài khoản (string)
- setName(string): đổi tên người sử dụng tài khoản (string) – tự động chỉnh sửa theo dạng chuẩn (vd: Trần Văn A)
- getAvatar(): lấy hình ảnh của người sử dụng tài khoản (string)

- `setAvatar(string)`: đổi hình ảnh đại diện của người sử dụng tài khoản (string)
- `getUid()`: lấy id của người sử dụng tài khoản (string)
- `setUid(string)`: đổi id của người sử dụng tài khoản (string) – phát sinh id và tránh bị lặp id của tài khoản khác
- `getEmail()`: lấy email của người sử dụng tài khoản (string)
- `setEmail(string)`: đổi email của người sử dụng tài khoản (void) – xác minh tài khoản
- `getPhoneNumber()`: lấy số điện thoại của người sử dụng tài khoản (string)
- `setPhoneNumber(string)`: đổi số điện thoại của người sử dụng tài khoản (void) – đúng định dạng, gồm 10 số (vd:0909 090909)
- `getAttribute()`: Lấy ra tất cả thông tin của người sử dụng tài khoản (attribute)
- `setAttribute(attribute)`: Thay đổi tất cả thông tin của người sử dụng tài khoản (void)
- `UserModel()`: constructor tạo mới một đối tượng thông tin người dùng

- b. **Model PostJob**: dùng để đưa dữ liệu từ `DataPostJob` (chứa các thông tin trong 1 tin tuyển dụng) lên firebase để lưu trữ

Các thuộc tính trong lớp:

- `databaseReference`: là biến lưu địa chỉ đến vùng nhớ chứa các tin tuyển dụng trên firebase

Các phương thức trong lớp:

- `ModelPostJob()`: khởi tạo giá trị của `databaseReference` để đi đến phần tin tuyển dụng trên firebase
- `SavePost(string Uid, string DataPostJob)`: lưu thông tin của 1 tin tuyển dụng mới (void)
- `savePostEditing(string idJob, string Uid, string DataPostJob)`: chỉnh sửa thông tin của tin tuyển dụng đã có (void)

- c. **ModelApprovalCompany**: dùng để quản lý thông tin của các công ty đã đăng ký với hệ thống và đang chờ duyệt để sử dụng

Các thuộc tính:

- `dbrCompanyWaitingApproval`: đi đến các tin tuyển dụng (`DatabaseReference`)
- `model`: mẫu lưu dữ liệu của các tin tuyển dụng (`InfoCompanyModel`)

Các phương thức:

- `ModelApprovalCompany()`: khởi tạo và đi đến vùng chứa các thông tin của các công ty đang chờ duyệt
- `onApproval()`: xử lý kết quả từ admin (chấp thuận hoặc không chấp thuận) về thông tin đã đăng ký và gửi thông báo về cho tài khoản công ty đó (boolean)

B. System Architecture and System Design

1. Architectural Styles

Describe the software architecture style/pattern used in your design.

a. Kiến trúc MVP (Model – View – Presenter):

MVP là một mô hình kiến trúc hướng giao diện người dùng, được thiết kế để tạo thuận lợi cho việc kiểm thử đơn vị (unit testing) và tăng tính tách biệt giữa tầng dữ liệu và tầng hiển thị dữ liệu trong mô hình MVC. Mô hình MVP cho phép tách tầng trình diễn (Presenter) ra khỏi tầng dữ liệu (Model), vì vậy tương tác với giao diện được tách biệt với cách chúng ta biểu diễn nó trên màn hình (View), hay nói cách khác, tất cả logic khi người dùng tương tác được tách ra và đưa vào tầng trình diễn. Thiết kế lý tưởng nhất là với cùng một logic được áp dụng cho nhiều View khác nhau và hoán đổi được cho nhau.

Trong đó:

Model là một interface xác định cách mà dữ liệu được hiển thị trong giao diện người dùng.

View là một giao diện người dùng thụ động hiển thị dữ liệu (Model) và tiếp nhận tương tác người dùng và truyền tới để Presenter xử lý tương tác.

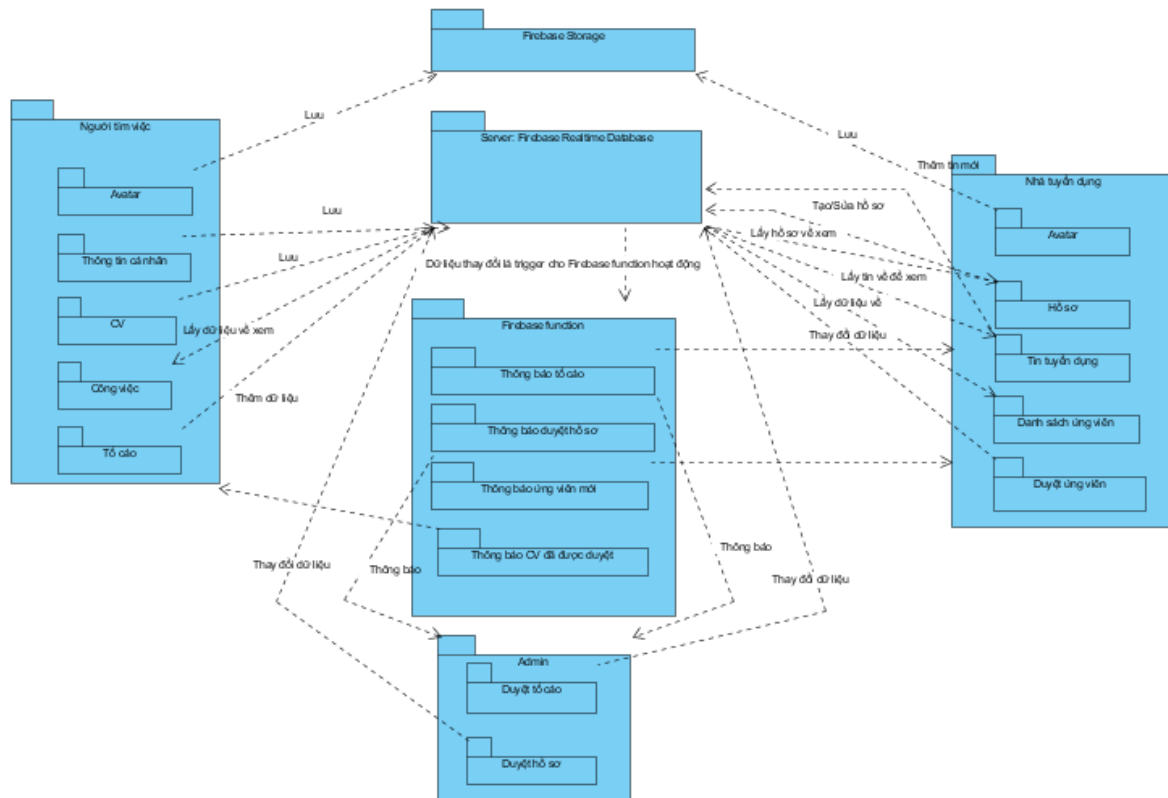
Presenter được ví như middle-man. Khi người dùng tương tác với View, Presenter tiếp nhận tương tác người dùng và update Model. Khi Model được update hay có thay đổi, Presenter lấy dữ liệu từ Model, định dạng và đưa tới View để hiển thị.

b. Kiến trúc Client-Server

- Là mô hình mà trong đó các máy chủ (server) cung cấp và quản lý các nguồn lực và dịch vụ cho máy khách (client)
- Kiểu kiến trúc này có một hoặc nhiều máy khách kết nối với máy chủ trung tâm thông qua mạng hoặc Internet.
- Kiến trúc Client-Server cũng có thể được gọi là mạng mô hình tính toán vì tất cả các yêu cầu và dịch vụ là được phân phối qua mạng.

2. Identifying Subsystems

Không áp dụng.



Hình 23. Sơ đồ subsystem (gồm server- Firebase Realtime Database, Firebase function, Firebase Storage và Admin)

3. Mapping Subsystems to Hardware

Không áp dụng.

4. Persistent Data Storage

Không áp dụng.

5. Network Protocol

Không áp dụng.

6. Global Control Flow

- Quá trình lấy dữ liệu và hiển thị lên view (Activity, Fragment) (Mô hình MVP):

Trong Model chạy 1 tiến trình (gọi là tiến trình A) chạy song song với tiến trình chính, đặt 1 callback trong tiến trình A. Khi dữ liệu được lấy hoàn tất sẽ kích hoạt callback gửi dữ liệu cho Activity hiển thị. Khi đó tiến trình A sẽ kết thúc.

- Quá trình hiển thị và lưu hình ảnh vào bộ nhớ:
Trong hàm onBindViewHolder chạy 1 tiến trình(gọi là tiến trình B) để download ảnh trên server về khi quá trình lấy dữ liệu hoàn tất ta sẽ hiển thị lên view. Sau đó tạo một tiến trình để lưu hình ảnh vào bộ nhớ (gọi là tiến trình C) (ngay khi tạo xong tiến trình C, tiến trình B kết thúc) và khi lưu ảnh xong tiến trình C sẽ kết thúc.
- Quá trình lưu dữ liệu trên server và vào bộ nhớ:
Khi người dùng tiến hành lưu dữ liệu (nhấn button) thì sẽ tạo 2 tiến trình để lưu dữ liệu lên server và vào bộ nhớ. Khi lưu xong tiến trình lưu dữ liệu vào bộ sẽ tự kết thúc, tiến trình lưu dữ liệu trên server sẽ báo thành công hoặc thất bại về và ta sẽ hiển thị lên view cho người dùng biết và sau đó kết thúc.

7. Hardware Requirements

Phiên bản android ≥ 5.0 (Lollipop (API 21)).

Màn hình ≥ 5 inch.

Cần bộ nhớ để chạy và lưu trữ dữ liệu. Sau 1 tuần dùng app bộ nhớ mà app sử dụng trong máy chỉ 8MB rất nhỏ so với bộ nhớ của điện thoại hiện nay. Vì vậy app chỉ cần bộ nhớ máy trống ~ 50 MB là app hoạt động ổn định(Trong điều kiện chỉ có 1 app JobIT hoạt động).

Cần cấp quyền truy cập bộ nhớ cho app để truy cập, lưu trữ dữ liệu trong bộ nhớ trong.

Cần có mạng ≥ 128 Kbps.

C.Algorithms and Data Structures

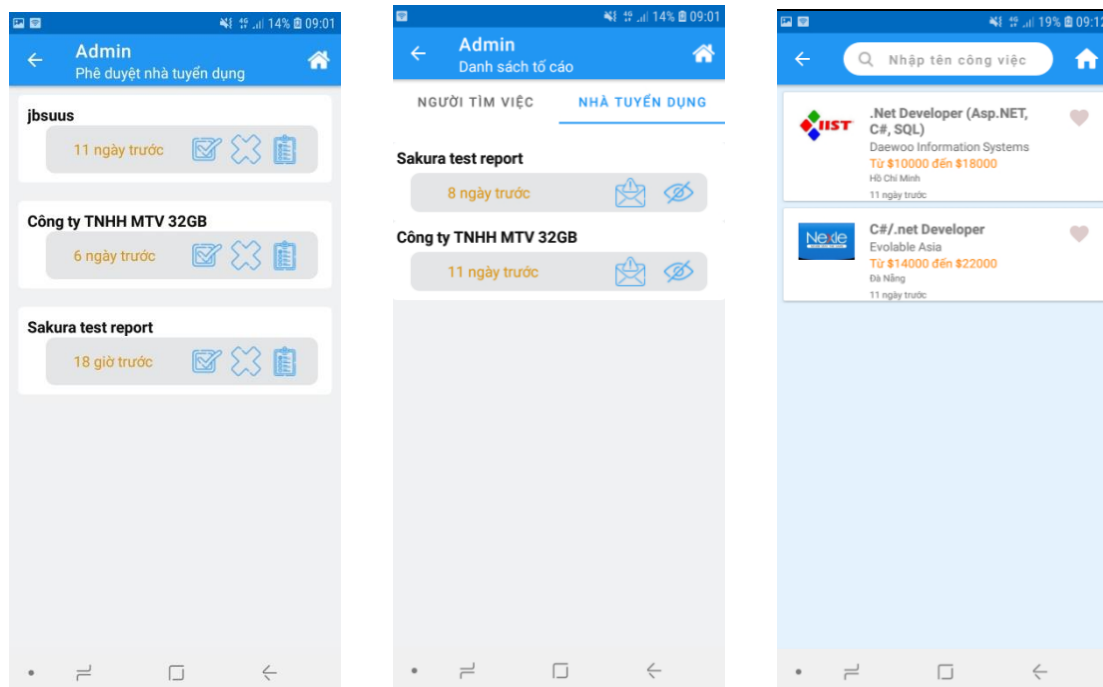
1. Algorithms

(Không áp dụng giải thuật phức tạp trong đề án.)

2. Data Structures

Đề án sử dụng cấu trúc dữ liệu danh sách (lists) và cây (trees).

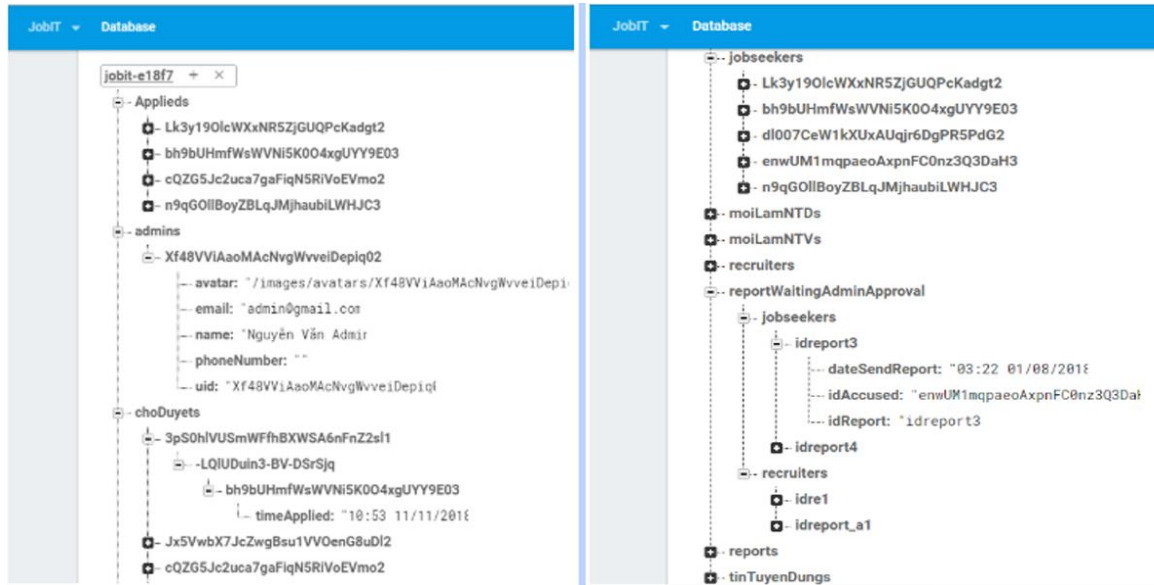
- a. Cấu trúc dữ liệu lists dùng để lưu list các đối tượng (Ví dụ: list các công việc, list ứng viên, list tổ cáo, list hồ sơ công ty cần duyệt...). Vì số lượng phần tử trong mỗi danh sách không biết trước và thay đổi liên tục vì thế cấu trúc dữ liệu dạng list là hợp lý và linh hoạt. Cấu trúc này cho phép ta tạo ra một mảng các phần tử với số lượng phần tử có thể thay đổi được, việc thêm/ xóa/ cập nhật/ truy cập từng phần tử trong danh sách cũng được thực hiện dễ dàng.



Hình 24. Hình ảnh minh họa dữ liệu dạng lists

- b. Cấu trúc dữ liệu dạng cây dùng để lưu data của toàn hệ thống. Hệ thống sử dụng Firebase Realtime Database để lưu database. Trên Firebase Realtime Database, dữ liệu được tổ chức theo dạng tree. Cấu trúc dữ liệu tree phù hợp với app do database của app JobIT

được tổ chức phân tầng, độ sâu không quá 5, có thể truy vấn dữ liệu một cách logic và phù hợp với môi trường Firebase Realtime Database.



Hình 25. Hình ảnh minh họa cấu trúc dữ liệu dạng tree

Project Management and Plan of Work

1. Merging the Contributions from Individual Team Members

Compiling the final copy of the report from everyone's work, ensuring consistency, uniform formatting and appearance.

| STT | Họ tên | Khó khăn | Giải quyết |
|-----|-----------------|---|--|
| 1 | Trần Phú Nguyễn | <ul style="list-style-type: none"> Lần đầu tiếp xúc với firebase (Database, Authentication, Storage, FCM, Function Cloud) Algolia, Mô hình MVP nên gặp nhiều khó khăn trong quá trình code, fix bug. Tài liệu về MVP liên quan đến Firebase, tài liệu về algolia, Function cloud khá hiếm quá trình tìm hiểu, triển khai, fix bug rất tốn thời gian. Khả năng đọc hiểu tiếng anh không tốt mà tài liệu hầu hết là tiếng anh nên khó đọc hiểu. Là nhóm trưởng và là người biết một ít về android trước các bạn trong nhóm nên là người tìm hiểu trước, quyết định các công nghệ (firebase, | <ul style="list-style-type: none"> Tham khảo Medium, StackOverflow, Github,... .Có nhiều bug không tìm thấy trên stackoverflow tự mày mò phân tích để fix. Nhờ các công cụ dịch, từ điển,... để đọc hiểu. Các bạn trong nhóm cũng khá hiểu ý, truyền đạt nhiều lần các bạn cũng hiểu. |

| | | | |
|---|---------------------|--|---|
| | | <p>androidx, algolia), mô hình (MVP) áp dụng trong đề án, giải quyết các vấn đề để hướng dẫn các bạn trong nhóm. Tuy nhiên khả năng truyền đạt còn kém, không truyền đạt được hết ý dẫn đến các bạn khó hiểu ý của em.</p> <ul style="list-style-type: none"> Lần đầu làm nhóm trưởng trong một đề án khá lớn, chọn đề tài và các công nghệ mới, khá khó đối với các bạn và bản thân. Khả năng quản lý nhóm, phân chia công việc còn kém gây ảnh hưởng tới tiến độ của đề án. | <ul style="list-style-type: none"> Nhờ các bạn trong nhóm cùng nhau giải quyết. Các bạn hay nhắc nhở khi em quên phân chia công việc, deadline,... Và sự nhiệt tình hết mình của các bạn trong nhóm. |
| 2 | Trần Ngô Anh Nguyên | <ul style="list-style-type: none"> Lần đầu làm đề án với ngôn ngữ mới: Java, xml Có nhiều kiến thức cần biết để thực hiện đề án: Firebase, mô hình MVP, cách sử dụng Android Studio để debug ứng dụng,... Khả năng hợp tác nhóm còn hạn chế Không có thiết bị Android nên gặp trở ngại khi kiểm thử các chức năng Từ ngữ được sử dụng trong tài liệu tiếng anh đa số là từ chuyên ngành nên tốn khá nhiều thời gian để hiểu | <ul style="list-style-type: none"> Tham khảo các tài liệu có sẵn (Victor Matos...) Tìm hiểu về các chức năng trên trang ứng dụng, tham khảo cách giải quyết các vấn đề trên các trang chia sẻ kiến thức lập trình (stackoverflow...) Học hỏi kiến thức từ các bạn nhiều hơn, cố gắng tương tác với các bạn trong nhóm. Nhờ các bạn trong nhóm kiểm thử một vài chức năng. |

| | | | |
|---|---------------|---|--|
| | | | <ul style="list-style-type: none"> • Tìm hiểu từ mà mình cần hiểu rõ trên google (dịch, wikipedia,...) |
| 3 | Trần Thị Lý | <ul style="list-style-type: none"> • Gặp khó khăn trong quá trình đầu mới học Android. Đây là một môn học mới, cần tìm hiểu và thực hành nhiều mới hiểu và biết cách làm. • Không hiểu hết các kỹ thuật cần để áp dụng trong đồ án. • Một số chức năng trong đồ án được thay đổi so với ban đầu để linh hoạt và phù hợp hơn nên cần thay đổi activity diagram, database, cách xử lý. | <ul style="list-style-type: none"> • Tự đọc nhiều tài liệu, xem video hướng dẫn để làm theo. • Nhờ sự giúp đỡ của các bạn trong nhóm. • Họp nhóm hàng tuần để trao đổi định kỳ và họp nhóm vào những ngày khác nếu cần để bàn bạc về những vấn đề cần thống nhất trong đồ án. |
| 4 | Trần Thị Lạng | <ul style="list-style-type: none"> • Bước đầu tìm hiểu Firebase: Realtime Database, Storage • Bước đầu sử dụng GitKraken, tìm hiểu cách sửa lỗi conflict, đồng bộ code trên Git và máy. • Lần đầu làm quen với ngôn ngữ Java, Android nên bước đầu code khó khăn. • Các thành viên trong nhóm hầu hết lần đầu làm chung nên khi thảo luận đưa ra ý kiến còn có sự bất đồng ý kiến | <ul style="list-style-type: none"> • Tra google, tham khảo trên trang vibo, stackoverflow. • Hỏi các thành viên trong nhóm và sửa giúp • Khi xảy ra mâu thuẫn về vấn đề gì đó thì mọi người đều lắng nghe để đưa ra cách giải quyết và đặc biệt là tôn trọng nhóm trưởng |

| | | | |
|---|-----------------------|---|---|
| 5 | Nguyễn Lê Thị Mỹ Linh | <ul style="list-style-type: none"> Bước đầu tìm hiểu Firebase: Realtime Database, Storage Tìm hiểu các cách thiết kế giao diện đẹp trong android, cách debug trong android Các tài liệu hầu hết đều là tiếng Anh nên đọc khó hiểu Thời gian tìm hiểu một vấn đề tốn thời gian vì chưa biết cách thức hoạt động như thế nào đặc biệt là Firebase lần đầu tiếp xúc Các thành viên trong nhóm hầu hết lần đầu làm chung nên khi thảo luận đưa ra ý kiến còn có sự bất đồng ý kiến | <ul style="list-style-type: none"> Tham khảo trên google, firebase.google.com, vibo, stackoverflow Hỏi các bạn khác trong nhóm và tìm cách giải quyết Nhờ google dịch Khi xảy ra mâu thuẫn về vấn đề gì đó thì mọi người đều lắng nghe để đưa ra cách giải quyết và đặc biệt là tôn trọng nhóm trưởng |
|---|-----------------------|---|---|

2. Project Coordination and Progress Report

Những use cases đã được thực hiện:

| STT | Usecase | Người thực hiện | Ghi chú |
|-----|--|-----------------|---------|
| 1 | Phân quyền người dùng lần đầu mở app | Phú Nguyễn | |
| 2 | Chuyển đổi chức năng (nhà tuyển dụng, người tìm việc, admin) | Phú Nguyễn | |

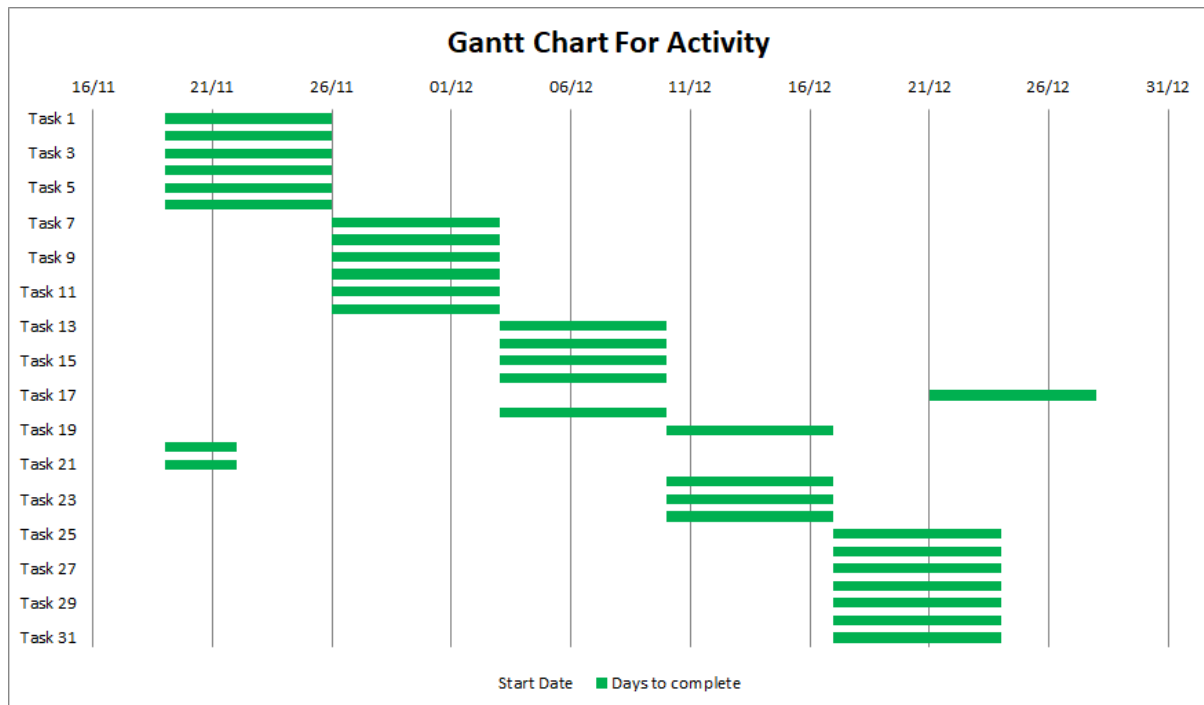
| | | | |
|---|------------------------------------|------------|--|
| 3 | Đăng ký | Phú Nguyễn | |
| 4 | Đăng nhập | Phú Nguyễn | |
| 5 | Đăng tin tuyển dụng | Linh | |
| 6 | Xóa tin tuyển dụng | Linh | |
| 7 | Lưu tin tuyển dụng | Lạng | |
| 8 | Tạo CV, chỉnh sửa CV | Phú Nguyễn | |
| 9 | Chỉnh sửa thông tin nhà tuyển dụng | Linh | |

Những use case đang được thực hiện

| STT | Usecase | Người thực hiện | Ngày bắt đầu | Thời gian thực hiện |
|-----|---------------------------------|------------------|--------------|---------------------|
| 1 | Tạo hồ sơ nhà tuyển dụng | Linh | 19/11 | 7 |
| 2 | Sửa hồ sơ nhà tuyển dụng | Linh | 3/12 | 7 |
| 3 | Xem hồ sơ nhà tuyển dụng | Phú Nguyễn | 26/11 | 7 |
| 4 | Đánh giá nhà tuyển dụng | Nguyên | 19/11 | 14 |
| 5 | Tổ cáo người tìm việc | Linh | 26/11 | 14 |
| 6 | Tổ cáo nhà tuyển dụng | Linh | 26/11 | 14 |
| 7 | Apply | Phú Nguyễn, Linh | 3/12 | 7 |
| 8 | Duyệt hồ sơ người tìm việc | Linh | 10/12 | 7 |
| 9 | Xem hồ sơ người tìm việc | Linh | 10/12 | 7 |
| 10 | Gửi thông báo có ứng viên apply | Phú Nguyễn | 10/12 | 7 |
| 11 | Thông báo cảnh cáo | Lý | 17/12 | 7 |

| | | | | |
|-----------|--------------------------------------|------------|-------|---|
| 12 | Xóa tổ cáo | Lý | 17/12 | 7 |
| 13 | Cảnh cáo | Lý | 17/12 | 7 |
| 14 | Duyệt hồ sơ thông tin nhà tuyển dụng | Lý | 17/12 | 7 |
| 15 | Tìm kiếm công việc | Phú Nguyễn | 19/11 | 7 |

3. Plan of Work



Hình 26. Biểu đồ Grantt chart thể hiện kế hoạch của nhóm

| Task | Mô tả | Người thực hiện |
|------|--|-----------------|
| 1 | Làm báo cáo System Design | Tất cả |
| 2 | Làm chức năng xác thực người dùng | Nguyễn |
| 3 | Tạo cơ sở dữ liệu | Linh, Lạng |
| 4 | Làm chức năng lấy ảnh/ lưu ảnh avatar | Phú Nguyễn |
| 5 | Làm chức năng tìm kiếm công việc | Phú Nguyễn |
| 6 | Làm chức năng đánh giá | Nguyễn |
| 7 | Làm chức năng cho người tìm việc tố cáo người tìm việc | Linh |
| 8 | Làm chức năng cho người tìm việc tố cáo nhà tuyển dụng | Linh |

| | | |
|-----------|---|------------------|
| 9 | Làm chức năng người tìm việc lưu tin tuyển dụng | Linh |
| 10 | Làm chức năng đăng tin tuyển dụng | Nguyễn |
| 11 | Làm chức năng xóa tin tuyển dụng đã lưu | Linh |
| 12 | Chỉnh sửa giao diện người tìm việc | Linh, Lạng |
| 13 | Chỉnh sửa giao diện nhà tuyển dụng | Phú Nguyễn, Lạng |
| 14 | Chỉnh sửa giao diện login | Phú Nguyễn |
| 15 | Làm chức năng tạo hồ sơ nhà tuyển dụng | Linh |
| 16 | Làm chức năng sửa hồ sơ nhà tuyển dụng | Linh |
| 17 | Test | Lạng |
| 18 | Làm chức năng apply | Phú Nguyễn, Linh |
| 19 | Tìm hiểu các công nghệ, kỹ thuật cần thiết để áp dụng vào đồ án | Phú Nguyễn |
| 20 | Tìm hiểu để đồng bộ nhiều máy cùng làm việc với 1 project trên firebase | Phú Nguyễn |
| 21 | Tìm hiểu và triển khai mô hình MVP cho nhóm | Phú Nguyễn |
| 22 | Làm chức năng duyệt hồ sơ người tìm việc | Linh |
| 23 | Làm chức năng xem hồ sơ người tìm việc | Linh |
| 24 | Làm chức năng gửi thông báo có ứng viên apply | Phú Nguyễn |
| 25 | Làm chức năng gửi thông báo hồ sơ apply đã được duyệt | Phú Nguyễn |
| 26 | Làm chức năng gửi thông báo hồ sơ phê duyệt từ admin | Lý |

| | | |
|-----------|---|----------|
| 27 | Làm chức năng gửi thông báo tố cáo từ admin | Lý |
| 28 | Làm chức năng xóa/gửi tố cáo từ admin | Lý |
| 29 | Làm chức năng phê duyệt hồ sơ của admin | Lý |
| 30 | Làm chức năng thống kê của admin | Lý |
| 31 | Chỉnh sửa giao diện admin | Lý, Lạng |

4. Breakdown of Responsibilities

| | Class | Modules | Người tổng hợp | Người test |
|----------------------------|--|--|----------------|------------------------|
| Trần Phú Nguyễn | <ul style="list-style-type: none"> CreateCV Applied JobSeekerProfile ListJobSearch SignIn SignUp SelectUserType | <ul style="list-style-type: none"> Người tìm việc tạo, chỉnh sửa CV. Apply công việc, khi apply thì sẽ có thông báo đến nhà tuyển dụng có ứng viên đã apply vào công việc tương ứng. Lưu thông tin công việc lên Firebase Hiển thị danh sách công việc tương ứng với từ khóa lúc tìm kiếm Người sử dụng app có thể đăng nhập và đăng ký tài khoản thông qua Facebook, Gmail, Google Phân quyền người dùng, có thể là người tìm việc, nhà tuyển dụng hoặc admin | Nguyễn | Nguyễn Linh Lạng |
| Trần Ngô | <ul style="list-style-type: none"> SignUpAccountB | <ul style="list-style-type: none"> Người tìm việc đánh giá nhà | Linh | Nguyễn, |

| | | | | |
|--------------------|--|--|----|--------------------|
| Anh Nguyễn | usiness <ul style="list-style-type: none"> • SignIn • SignUp | tuyển dụng. <ul style="list-style-type: none"> • Người tìm việc có thể bình luận nhà tuyển dụng • Người tìm việc có thể gửi tố cáo người tìm việc khác hoặc nhà tuyển dụng đến admin • Tìm hiểu thư viện Google và Facebook để hỗ trợ việc đăng nhập và đăng ký tài khoản. | | Nguyễn, Linh, Lạng |
| Trần Thị Lý | <ul style="list-style-type: none"> • AdminApproval • Report | <ul style="list-style-type: none"> • Admin nhận cảnh cáo khi người tìm việc gửi thông báo cảnh cáo đến. • Khi người tìm việc gửi tố cáo đến admin có thể bỏ qua hoặc cảnh cáo. • Thống kê số lượng user đã sử dụng app • Khi có một nhà tuyển dụng đăng ký, thông tin nhà tuyển dụng đó gửi đến admin. Admin có quyền duyệt thông tin nhà tuyển dụng hoặc không. Khi admin duyệt thì nhà tuyển dụng mới có thể đăng tin và | Lý | Lý, Lạng |

| | | | | |
|------------------------------|---|---|------------------|----------------------------|
| | | <p>xem hồ sơ, danh sách ứng viên.</p> <ul style="list-style-type: none"> Admin có thể xóa tài khoản của nhà tuyển dụng hoặc người tìm việc nếu phát hiện có hành vi không đúng hoặc cung cấp dữ liệu rác. | | |
| Trần Thị Lạng | <ul style="list-style-type: none"> SavedJob CreateCV Applied | <ul style="list-style-type: none"> Người tìm việc có thể lưu những công việc mình thích. Lấy dữ liệu trên Firebase về hiển thị lên app Tìm hiểu Storage Firebase để lưu ảnh, upload ảnh từ điện thoại lên ứng dụng Tìm hiểu Function Cloud để thực hiện việc gửi Notification khi có người tìm việc applied vào công việc để gửi thông báo đến cho nhà tuyển dụng | Nguyễn Linh Lạng | Nguyễn, Lạng, Lý |
| Nguyễn Lê Thị Mỹ Linh | <ul style="list-style-type: none"> SignUpAccountBusiness PostedJob PostJob | <ul style="list-style-type: none"> Lưu thông tin công ty khi đăng kí lên Firebase Có thể chỉnh sửa thông tin đã đăng ký lúc đầu. Lưu các tin đã đăng tuyển lên Firebase. | Nguyễn, Linh | Nguyễn, Linh, Lạng, Nguyễn |

| | | | | |
|--|--|--|--|--|
| | | <ul style="list-style-type: none">• Xóa các tin đã đăng trước đó• Lấy danh sách các công việc đã đăng từ Firebase để hiện thị về app• Khi người tìm việc applied vào công việc thì công ty có quyền duyệt hoặc không duyệt hồ sơ của ứng viên đó. Nếu công ty duyệt thì hồ sơ của người tìm việc sẽ được chuyển qua chờ phỏng vấn.• Khi có ứng viên apply vào công việc, nhà tuyển dụng sẽ nhận được thông báo có ứng viên apply vào công việc. | | |
|--|--|--|--|--|

D. References

Mô hình MVP:

<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93presenter>

Firebase:

<https://firebase.google.com/>

Kiến trúc Client-Server

https://en.wikipedia.org/wiki/Client%E2%80%93server_model