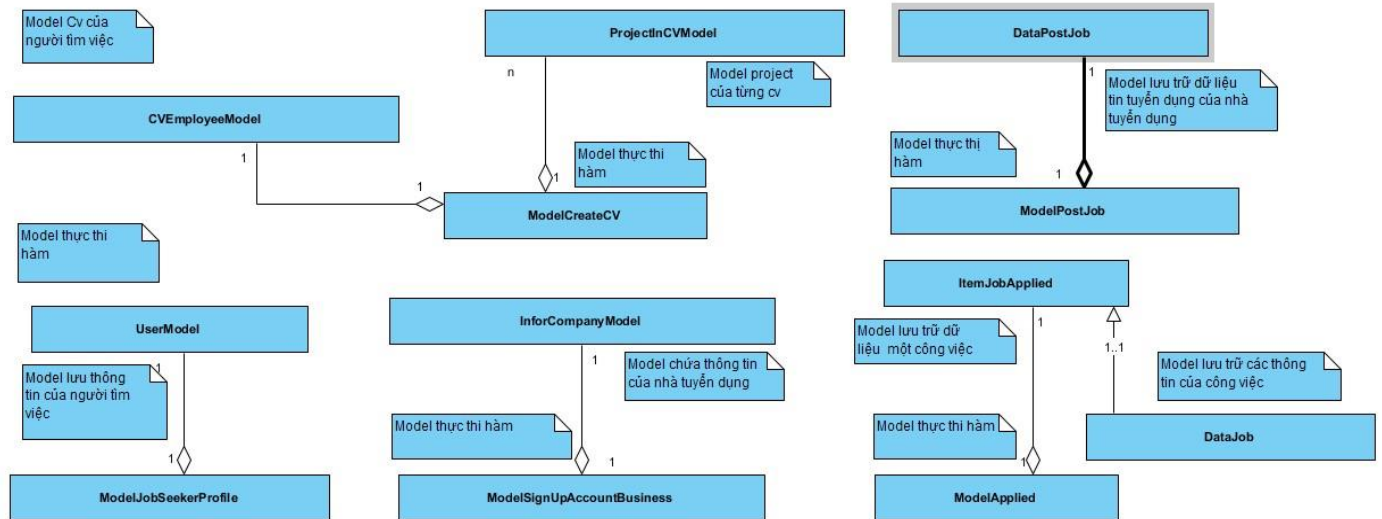


Class Diagram and Interface Specification

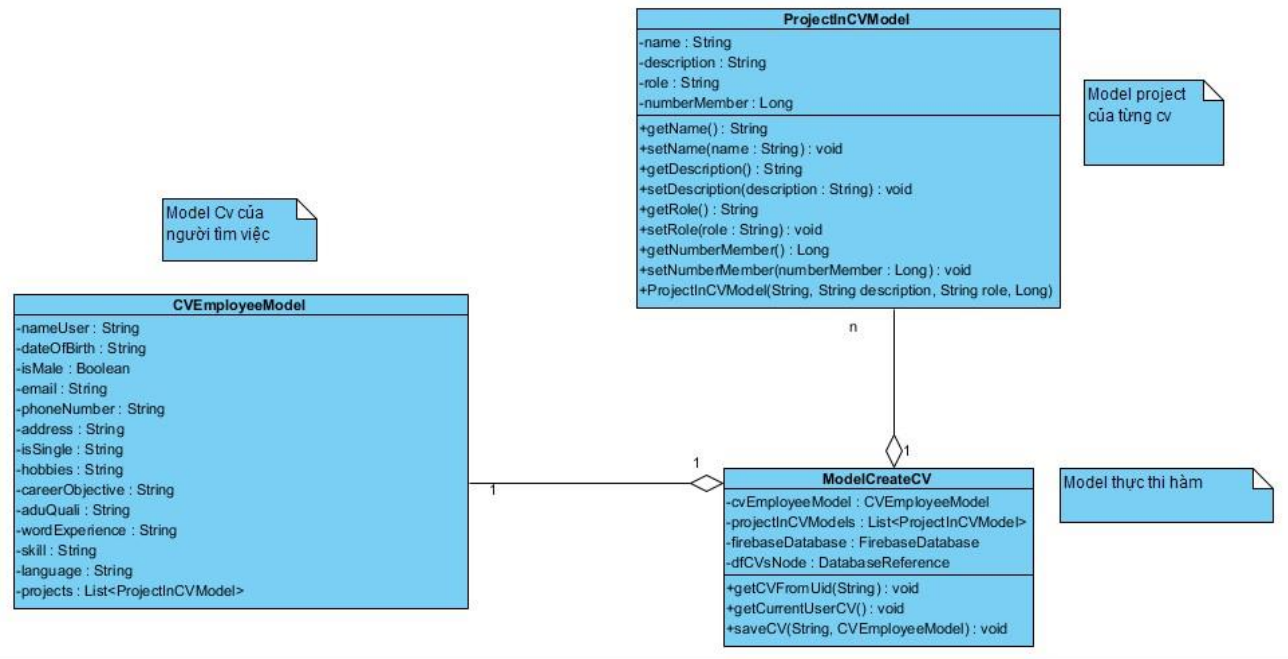
1. Class Diagram

a. Người tìm việc và nhà tuyển dụng

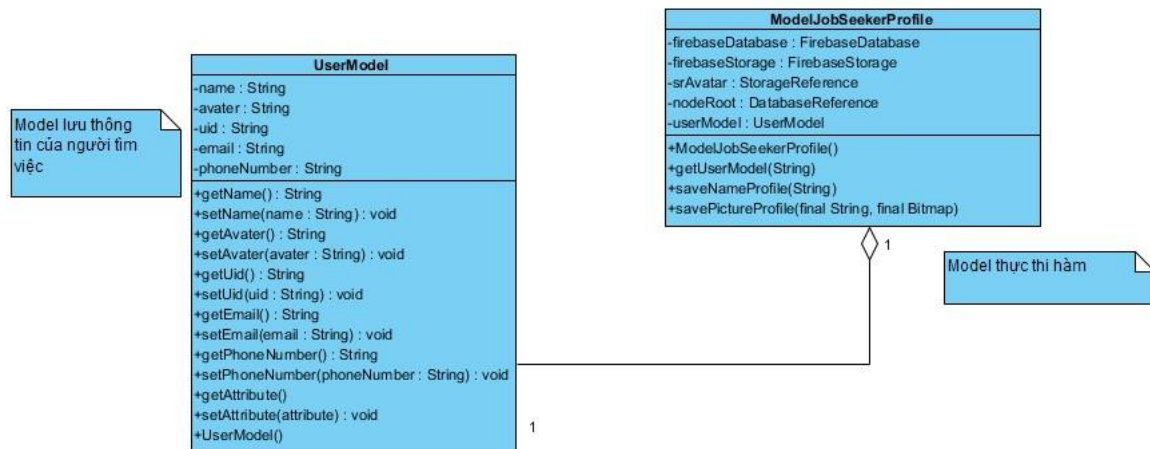


Hình 1. Tổng quan mối quan hệ giữa các class thao tác dữ liệu với

Firebase



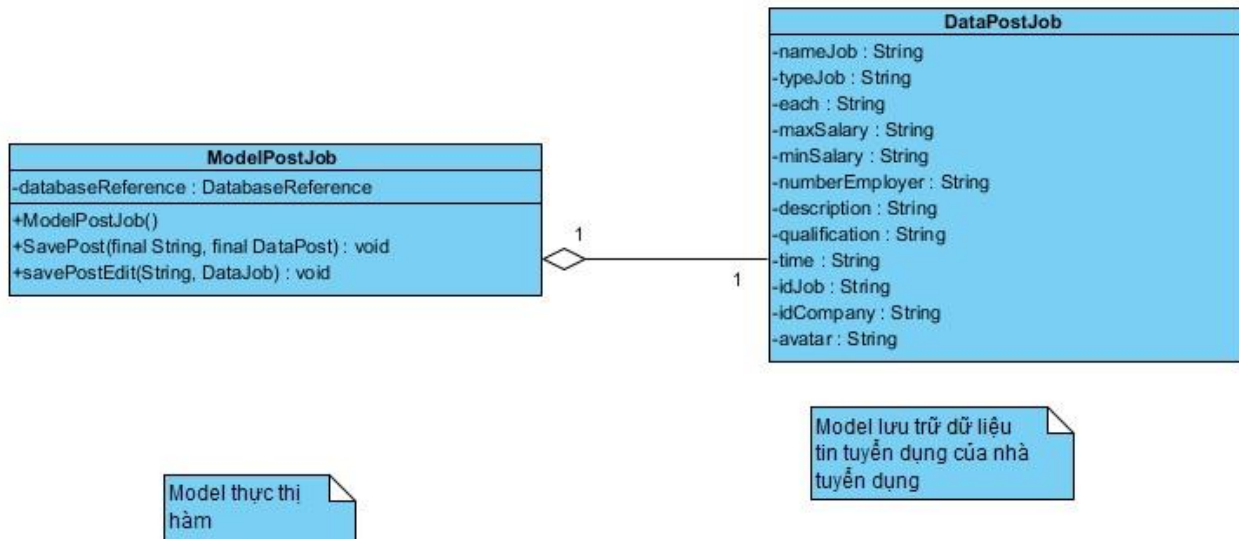
Hình 2. Mối quan hệ giữa class CVEmployeeModel, ProjectInCVModel và ModelCreateCV



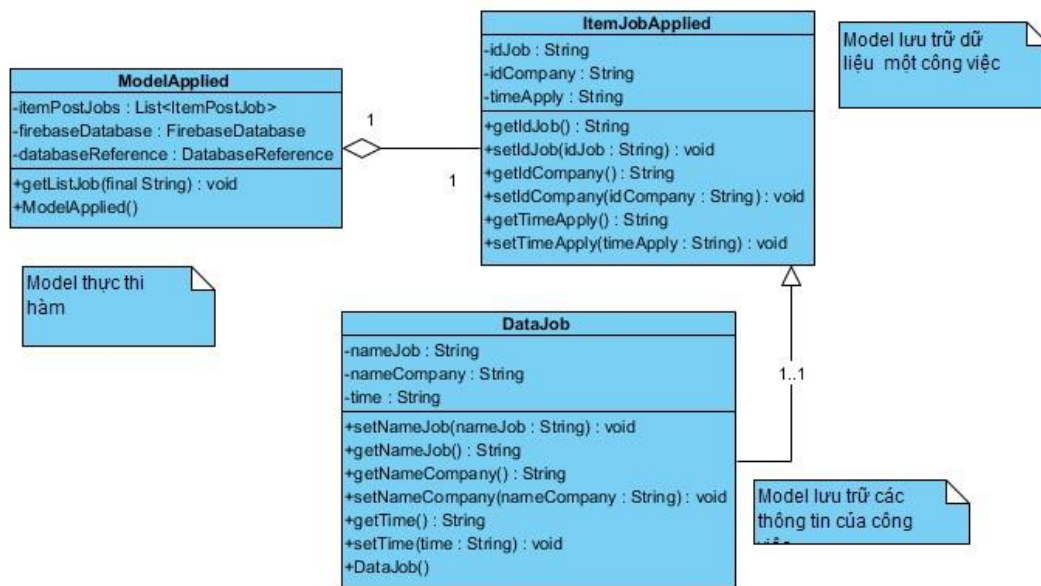
Hình 3. Mối quan hệ giữa class UserModel và ModelJobSeekerProfile



Hình 4. Mối quan hệ giữa ModelSignUpAccountBusiness và InforCompanyModel



Hình 5. Mối quan hệ giữa class ModelPostJob, DataPostJob

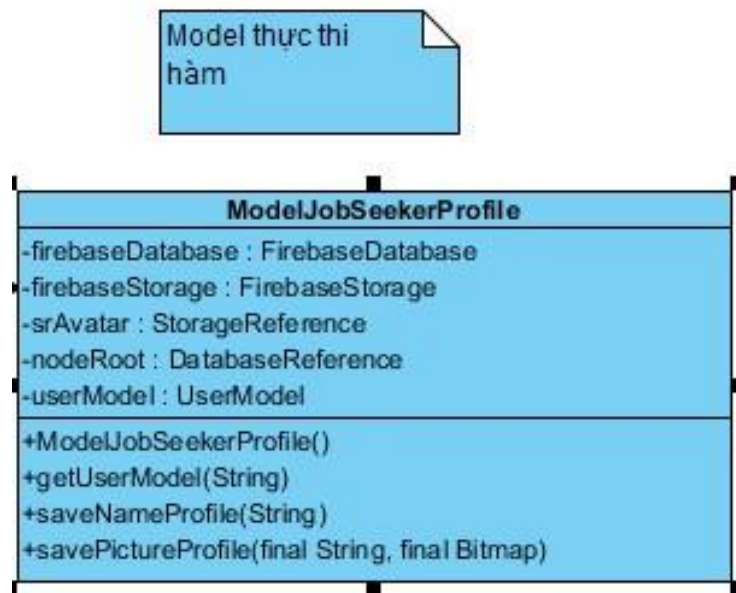


Hình 6. Mối quan hệ giữa class ModelApplied, DataJob và ItemJobApplied



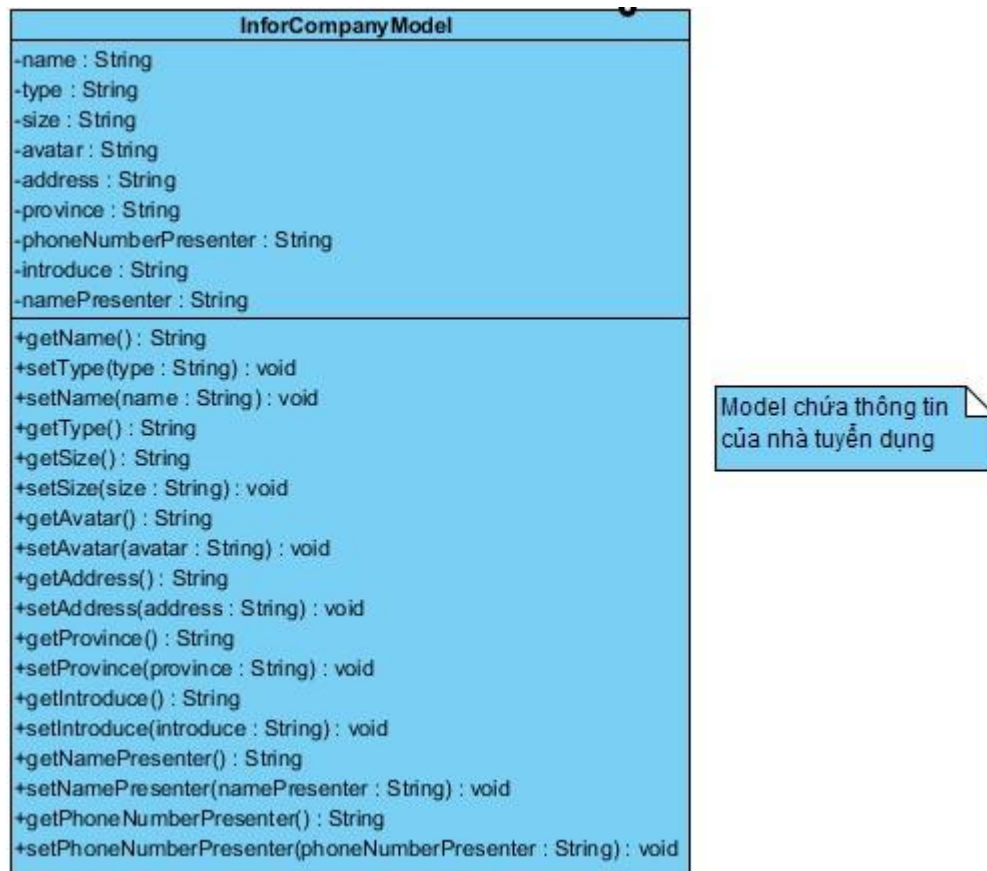
Hình 7. UserModel

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



Hình 8. ModelJobSeekerProfile

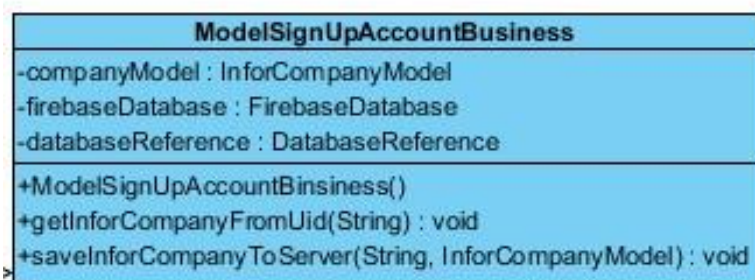
- Class ModelJobSeekerProfile dùng để thực thi các hàm, class này sẽ thực hiện việc lưu trữ thông tin người tìm việc lên Firebase.(hàm saveNameProfile và savePictureProfile)



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.

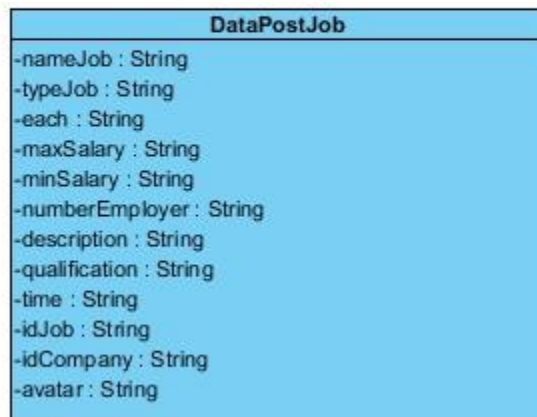
Model thực thi hàm



Hình 10. ModelSignUpAccountBusiness

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

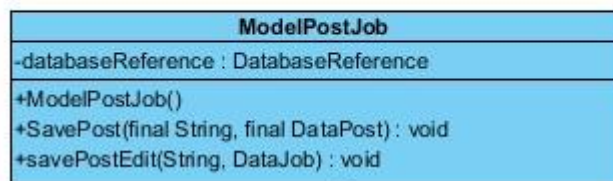
Model lưu trữ dữ liệu tin tuyển dụng của nhà tuyển dụng



Hình 11. DataPostJob

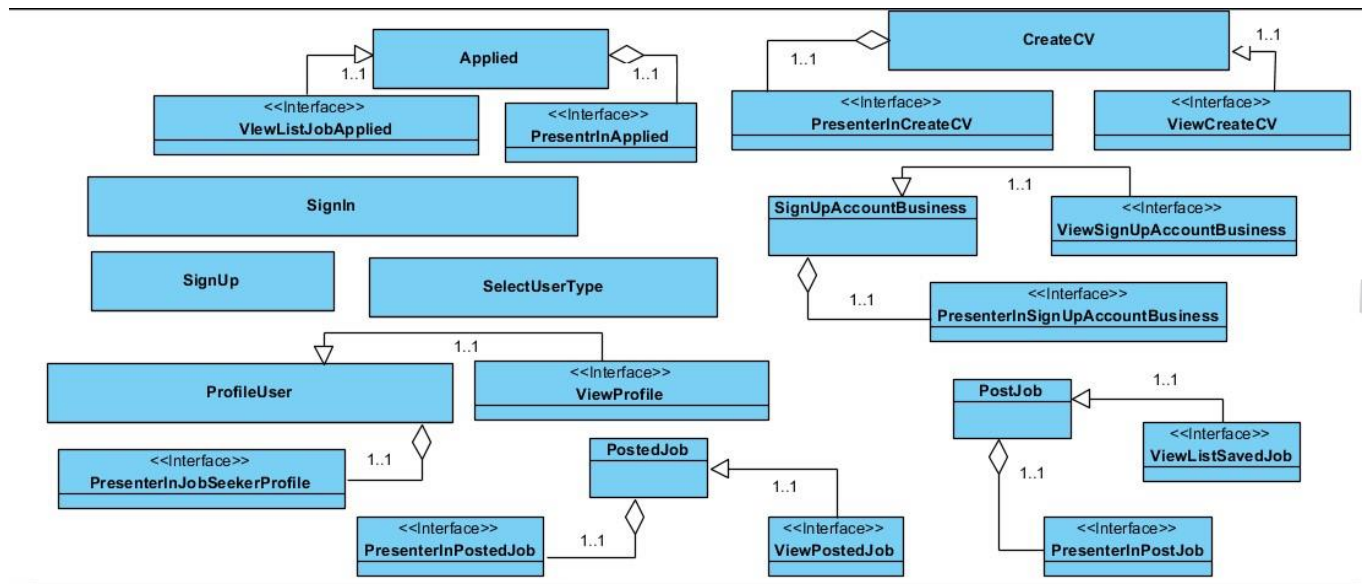
- Class model dùng để lưu trữ các thông tin về công việc mà công ty muốn đăng tuyển

Model thực thi hàm

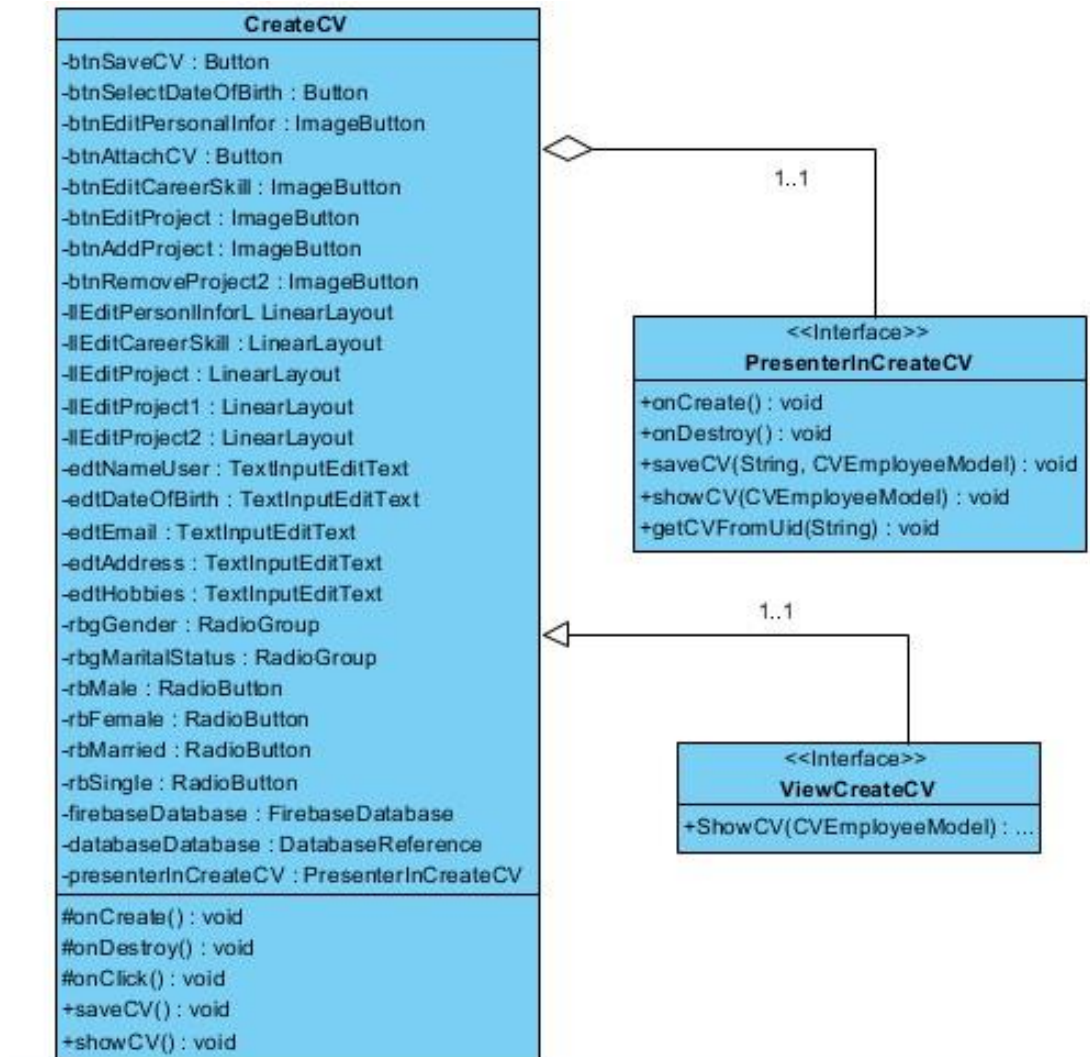


Hình 12. ModelPostJob

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

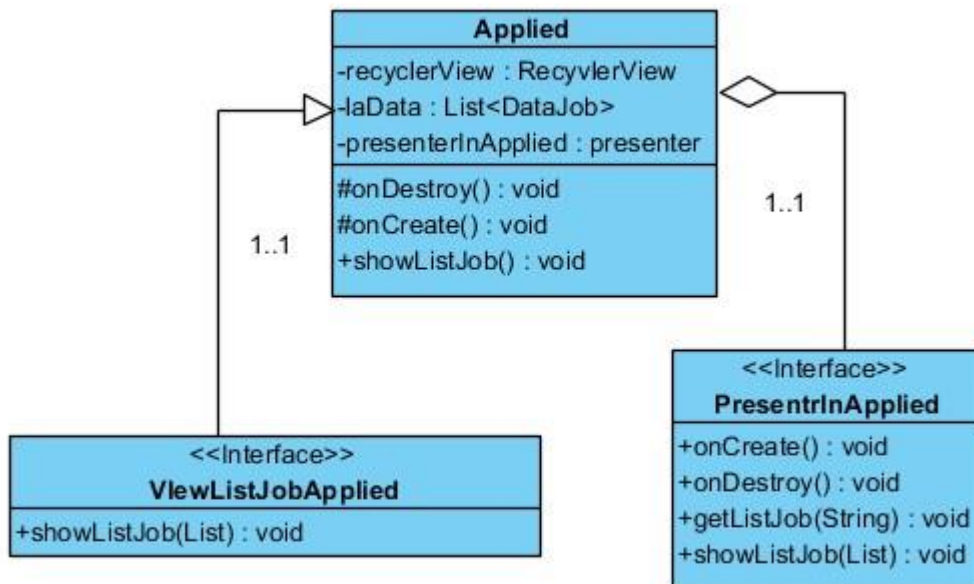


Hình 13. Tổng quan các class xử lý dữ liệu thực hiện với app



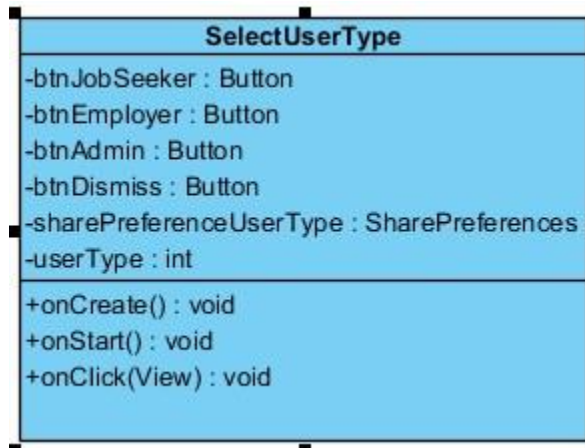
Hình 14. Class CreateCV

- 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.



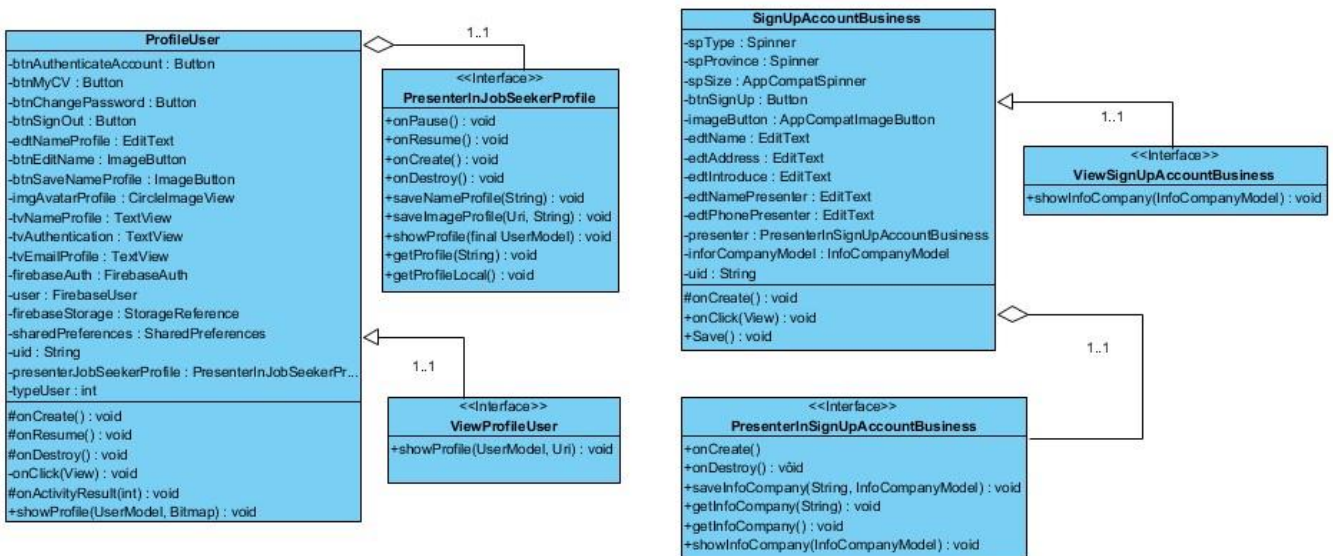
Hình 15. Class Applied

- 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 nhấp **Applied** thì công việc sẽ được lưu trên Firebase.



Hình 16. Class UserType

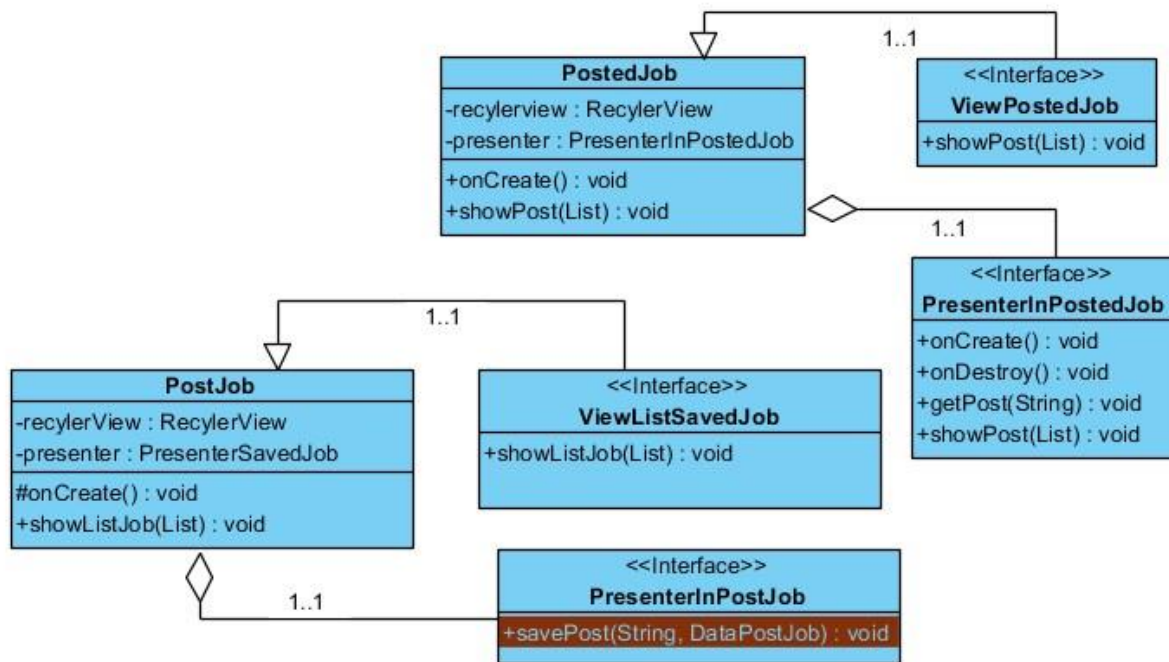
- Class SelectUserType dùng để phân quyền người dùng, là người tìm việc hay nhà tuyển dụng hay là Admin. Mỗi đối tượng có các các năng riêng biệt. Khi một tài khoản chỉ được sử dụng cho một đối tượng.
- Đối với Admin thì chỉ được quyền đăng nhập, không được quyền đăng ký tài khoản



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

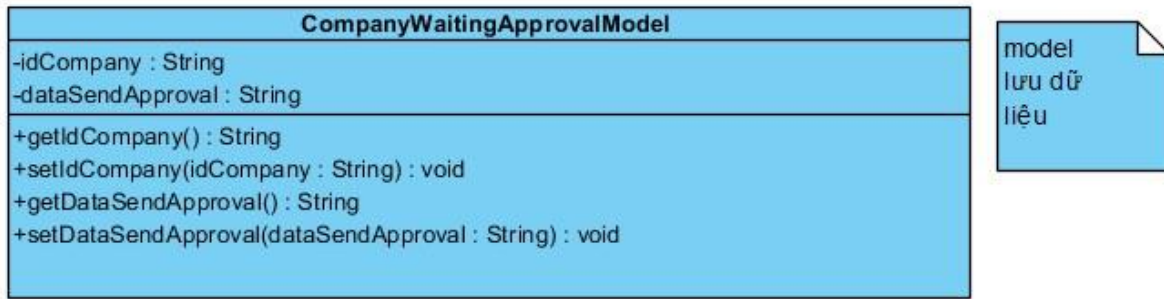
- đăng nhập hoặc đăng ký ứng dụng sẽ yêu cầu người dùng nhập những thông tin cần thiết.
- Các thông tin này được đưa lên Firebase để lưu trữ và quản lý.
View và Presenter có chức năng tương tự như những class trên



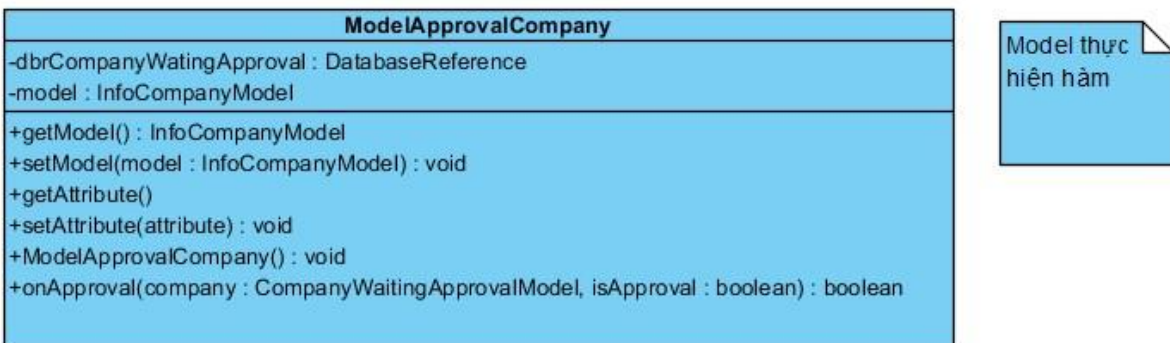
Hình 18. Class PostedJob và PostJob

- 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 hiển 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

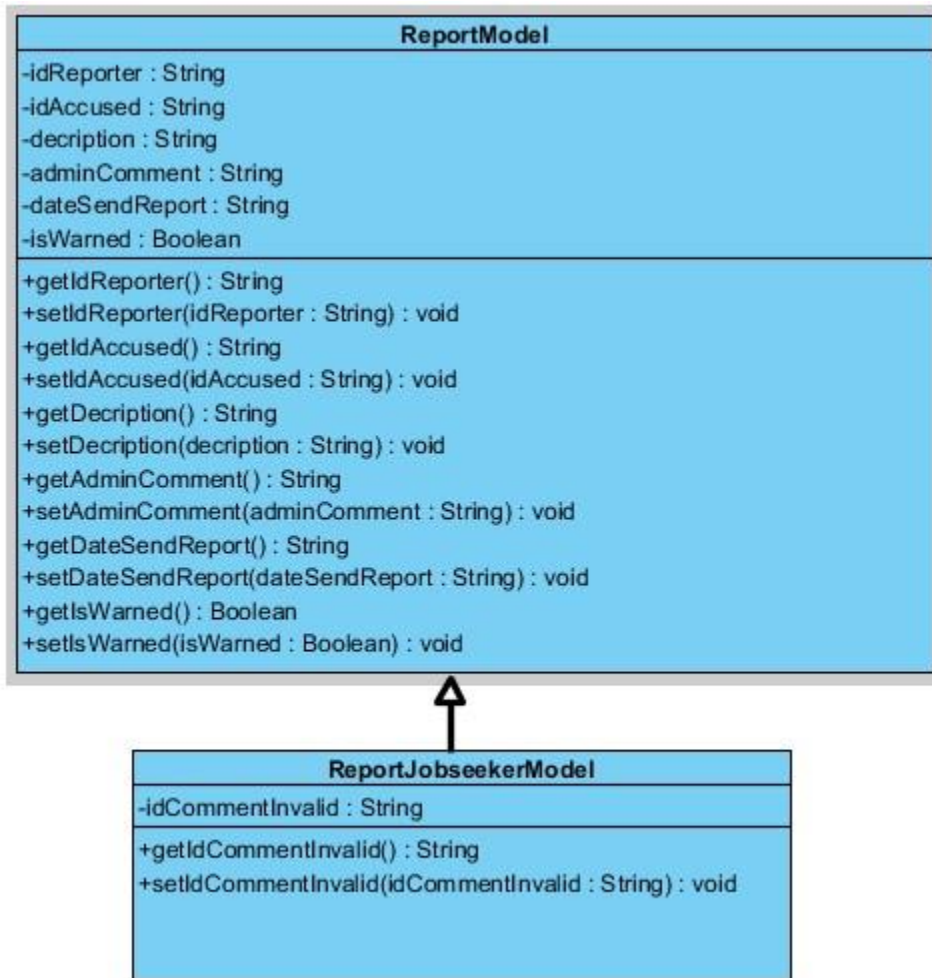


Hình 19. Class model công ty chờ admin duyệt hồ sơ.



Hình 20. Class ModelApprovalCompany

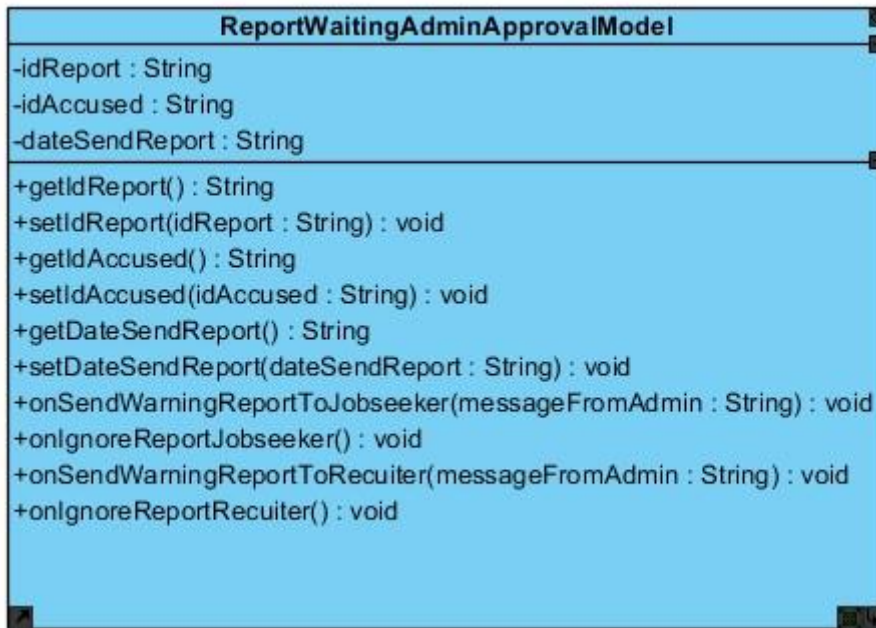
- Class model hồ sơ công ty cần duyệt, class này dùng để thực hiện hàm onApproval (xử lý phê duyệt công ty từ admin).



Hình 21. Class ReportModel và ReportJobSeekerModel

- Class model tố cáo (bên trên, ReportModel) và model tố cáo người tìm việc (bên dưới, kế thừa model tố 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

a. **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)
- getUserId(): 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 (InfoCompanyMo

del) **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)