

PROGRAMMATION MOBILE

1^{er} Rendu Intermédiaire Développement de la partie EMPLOYEUR

Laurent ANTOINETTE
M1-IASD
Romain CAMPILLO
M1-IASD

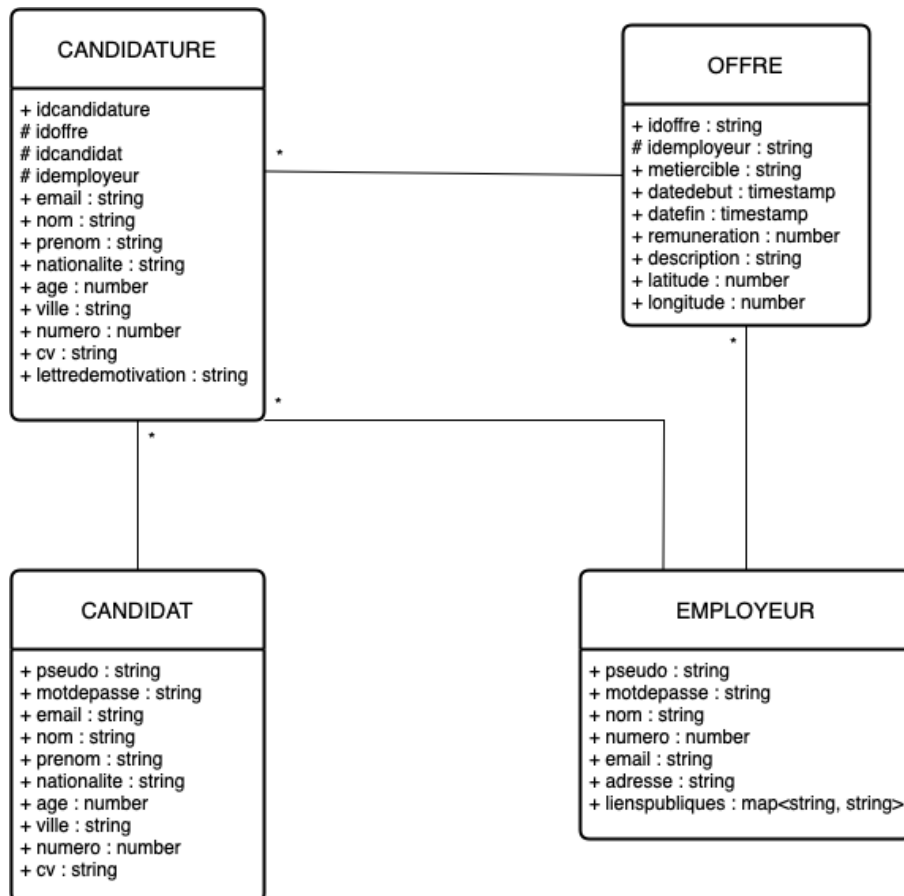


Table des matières :

1. Base de données
 - 1.1 Nouveau schéma UML
 - 1.2 Création d'une base de données Firebase
2. Outils de développements
 - 2.1 Classes d'objets utilitaires
 - 2.2 Classes de layouts utilitaires
3. Implementation de la partie EMPLOYEUR
 - 3.1 Les fonctionnalités implémentées
 - 3.2 L'affichage des cvs, lettres de motivations

Partie 1 : Base de données

1.1 Nouveau schéma UML

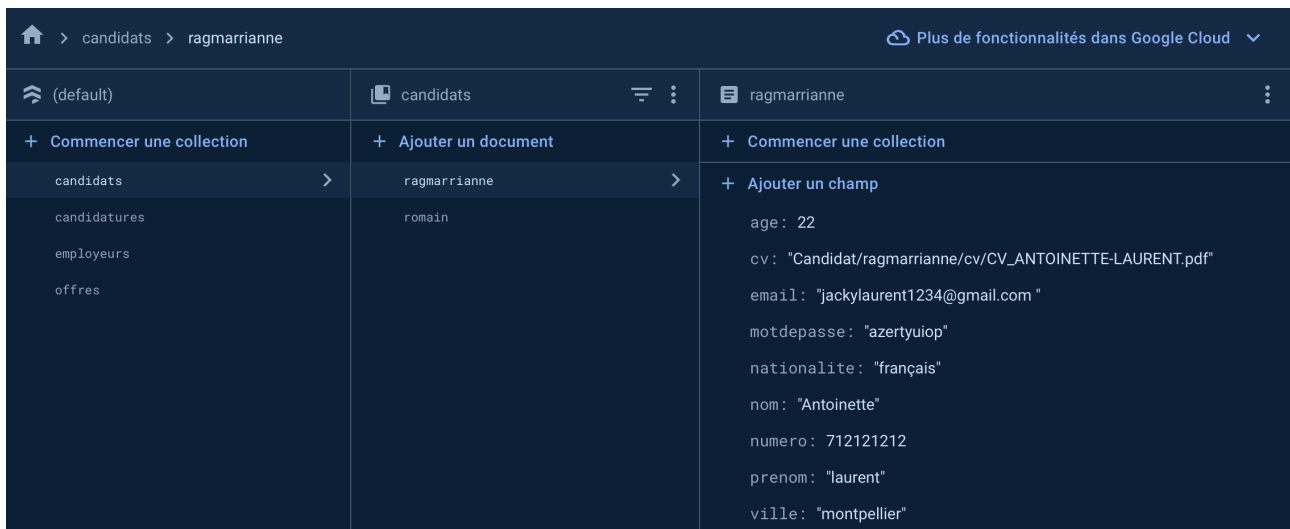


- Firestore Database acceptant les objets de type Map, les liens publiques ont été ajoutés dans l'entité EMPLOYEUR afin de simplifier leurs récupération.
- Dans l'entité CANDIDATURE, on a aussi ajouté l'id de l'employeur à l'origine de l'offre, toujours dans un but de simplifier la récupération des données.

1.2 Création d'une base de données Firebase

1.2.1 Firestore Database

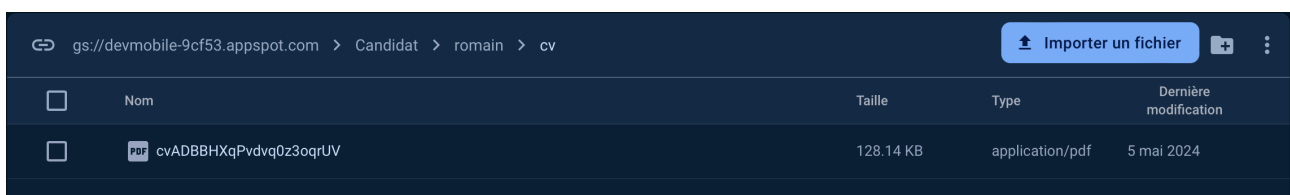
Pour chaque entité (candidat, offre ...), une collection Firestore a été créée et où chaque document appartenant à cette collection représente une instance de la dite entité.



1.2.2 Firebase Storage

Cependant, comme Firestore ne gère pas les attributs de type File et byte[], la création d'une base de données Firestore Storage a dû être faite afin de récupérer les diverses CVs et Lettres de motivations donnés par les candidats.

De ce fait, les attributs cv ou lettrede motivation qu'on peut retrouver dans notre base Firestore ne sont que des chaînes de caractères indiquant le chemin menant au fichier voulu depuis Firebase Storage.



Partie 2 : Outils de développements

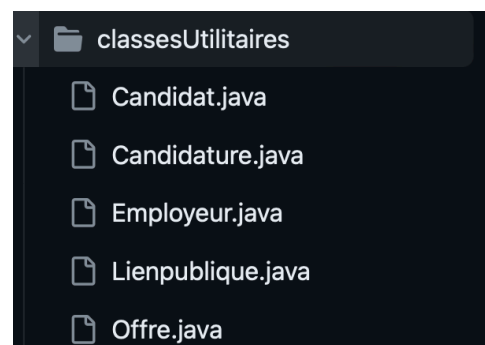
2.1 Classes d'objets utilitaires

Pour récupérer des données (documents) depuis Firebase, il existe deux moyens de recevoir celles-ci.

- Récupérer les données dans un objet de type `Map<String,Object>`, où la clé correspond à un attribut et où la valeur correspond à la valeur de cette attribut.
- Récupérer les données dans un objet personnalisé, dont le nom et type des attributs de la classe correspond à ceux présents dans le document.

Nous avons alors opté pour le second choix.

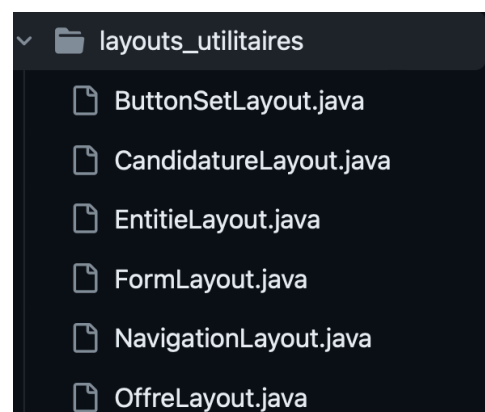
Ainsi pour chaque collection, une classe d'objet a été créée.



2.2 Classes de layouts utilitaires

Etant donné que dans le desine de notre application, il y avait une bonne similitude entre les vues, que se soit du côté candidat ou employeur. Nous avons alors eu l'idée de créer des layouts « utilitaires » étendues de la classe `LinearLayout`.

Celle-ci avait pour but de nous simplifier l'initialisation des vues, mais aussi de nous faciliter la récolte des données.



Celles-ci nous permettaient par exemple la création de formulaires et de la récupération des données saisies, l'affichage des offres déposées ou bien la création de la barre de navigation.

The image displays three distinct UI components. On the left is a light pink rounded rectangle representing a job offer card, containing text about a 'garde loge' position, dates, salary, and a placeholder for a description. In the center is a light grey rounded rectangle for a login form, with fields for email and password. On the right is a green rounded rectangle representing a user profile bar, featuring a circular icon with a person silhouette.

De plus, dans l'éventualité où une modification du desine doit être faite, il en devient moins fastidieux d'apporter ces modifications à l'ensemble des activités ou fragments utilisant ces layouts utilitaires.

Parite 3 : Implementation de la partie EMPLOYEUR

3.1 Les fonctionnalités implémentées

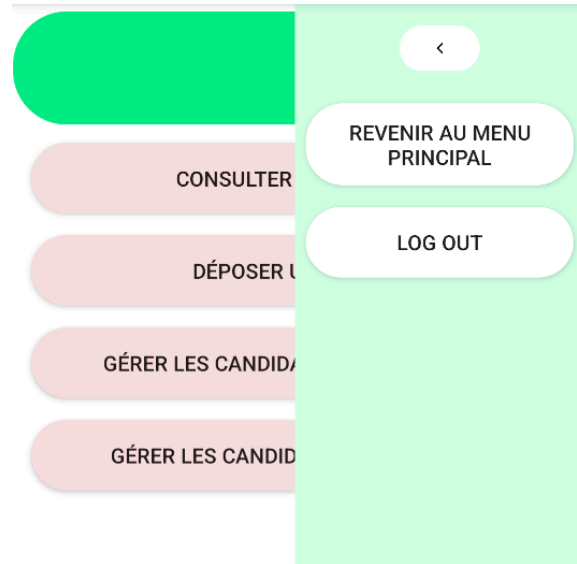
3.1.1 Inscription et authentification

Il est possible de l'utilisateur de s'inscrire et/ou s'authentifier depuis l'activité ReceptionActivity. Ainsi il lui est possible d'accéder à son compte.

Pour ce faire, nous procédons simplement à la vérification de l'existence de l'identifiant et la correspondance du mot de passe dans la base de données Firestore.

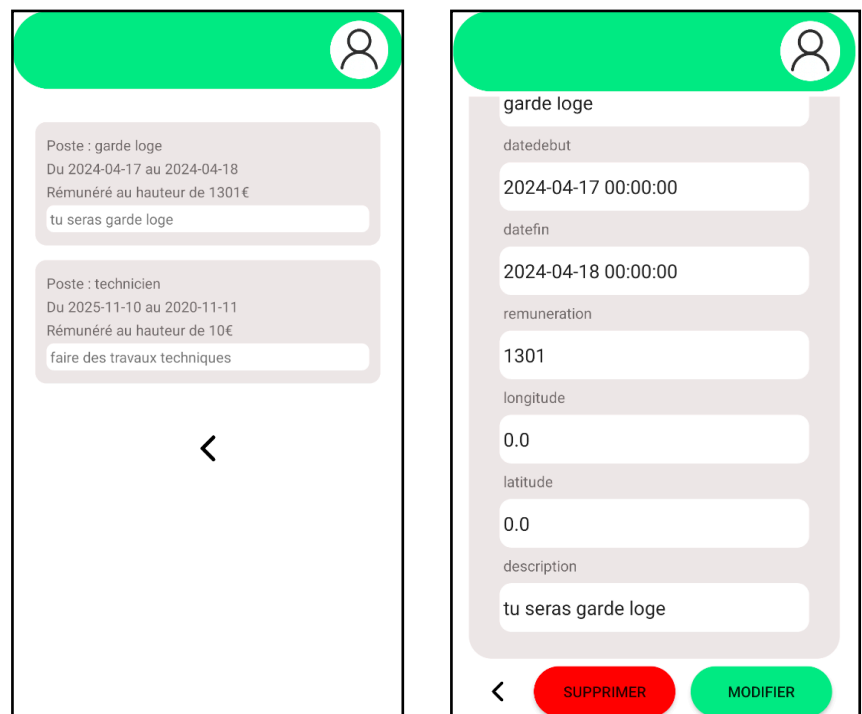
3.1.2 Navigation dans le menu

Après l'authentification, l'utilisateur a accès au menu lui permettant de consulter/modifier et déposer ses offres, mais aussi consulter les candidatures qu'il aurait reçu. Enfin il lui est possible de se déconnecter.



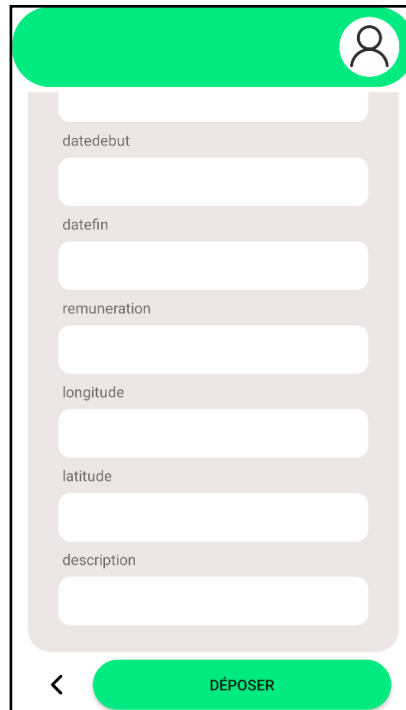
3.1.3 Consulter/modifier les offres

Les offres qui ont été déposées pourront être consultées et modifiées ou supprimées. (Note : il faudrait ajouter une confirmation lors de la modification ou la suppression des offres)



3.1.4 Déposer les offres

(Note : pour l'instant la latitude et la longitude sont à renseigner au lieu de l'adresse. L'objectif sera de mettre plus tard l'adresse et d'utiliser un framework proposé par google pour afficher l'itinéraire et calculer la distance)



A mobile application form for depositing offers. The form is displayed on a screen with a green header bar containing a user profile icon. The form fields are labeled as follows: 'datedebut', 'datefin', 'remuneration', 'longitude', 'latitude', and 'description'. Each label is followed by a white input field. At the bottom of the form, there is a green button labeled 'DÉPOSER' and a back arrow icon.

3.1.4 Gérer les candidatures

Dans les deux cas (en cours et accepté), les candidatures sont listées avec quelques informations utiles. Nom du métier cible, nom prénom du candidat, etc...

3.1.4.1 Cas d'une candidature en cours

Lorsque la candidature est encore en cours, l'employeur pourra afficher les informations sur le candidat. Et décider de refuser, accepter ou bien de laisser en cours la candidature. De plus Il lui est possible d'appeler le candidat.

The interface shows a candidate's details in a light gray box. At the top is a green header with a white user icon. The candidate information includes: Nom : Campillo, Prénom : Romain, Âge : 21, Nationalité : francais, Ville : Mtp, Email : romain@email.fr, and CV :. Below this are two buttons: 'AFFICHER CV' and 'AFFICHER LETTRE DE MOTIVATION'. At the bottom, there is a back arrow, a red 'REFUSER' button, a green 'ACCEPTER' button, and a gray 'APPELER LE CANDIDAT' button.

Nom : Campillo
Prénom : Romain
Âge : 21
Nationalité : francais
Ville : Mtp
Email : romain@email.fr
CV :
AFFICHER CV
Lettre de motivation :
AFFICHER LETTRE DE MOTIVATION
< REFUSER ACCEPTER
APPELER LE CANDIDAT

3.1.4.2 Cas d'une candidature acceptée

Lorsque la candidature a été acceptée, l'employeur pourra alors envoyer un mail ou bien appeler le candidat.

The interface shows the candidate's profile in a light gray box. At the top is a green header with a white user icon. The profile information includes: Offre : technicien, Profil du candidat, Nom : Campillo Romain, Age : 21, and Ville : Mtp. Below this are two buttons: 'APPELER' and 'ENVOYER UN EMAIL'. At the bottom, there is a back arrow.

Offre : technicien
Profil du candidat
Nom : Campillo Romain
Age : 21
Ville : Mtp
APPELER ENVOYER UN EMAIL
<

3.2 L'affichage des cv, lettre de motivations

Afin d'afficher les cvs et lettres de motivations, nous avons dû ajouter une nouvelle dépendance à notre projet car depuis AndroidX, il n'existait pas encore de vue spécifique pour l'affichage de documents pdf.

Nous avons alors ajouté la dépendance [PDFViewer](#) depuis MavenCentral.

Lien Github du projet

https://github.com/RagMarrianne/projet_devmobile.git