

Livrable	1
Sigle du cours	SEG2505
Professeur	Aziz Oukaira
Assistant (e)	Alexia Capo-Chichi
Groupe	G03

Membres du groupe

- Céline Wan Min Kee (celinewmk), 300193369
- Evan Marth (emarth), 300166093
- Samy Touabi (SamyT-code), 300184721
- Tisham Islam (TishamIslam), 300189261
- Othniel Tiendrebeogo (othnielt), 300084968

Livrable 1: Implémentation de la composante de gestion des comptes

1. Diagramme UML pour le livrable 1

Nous avons tout d'abord créé le diagramme UML de notre application avec les classes dont nous avons besoin afin de réaliser le livrable 1 (il y a aussi quelques méthodes incluses pour les livrables prochaines). Ensuite nous avons partagé les tâches entre les différents membres du groupe. Pour le nombre d'activité, nous avons conclu que 3 activités seront nécessaire pour ce livrable:

- MainActivity: la page de connexion qui permet à un utilisateur de se connecter à son compte s'il en possède déjà un. Elle donne aussi l'option de cliquer sur un lien pour créer un compte.
- CreateAccount: page qui permet à un utilisateur de créer un compte avec Service Novigrad.
- WelcomePage: page d'accueil qui affiche un message de bienvenue personnalisé à l'utilisateur.

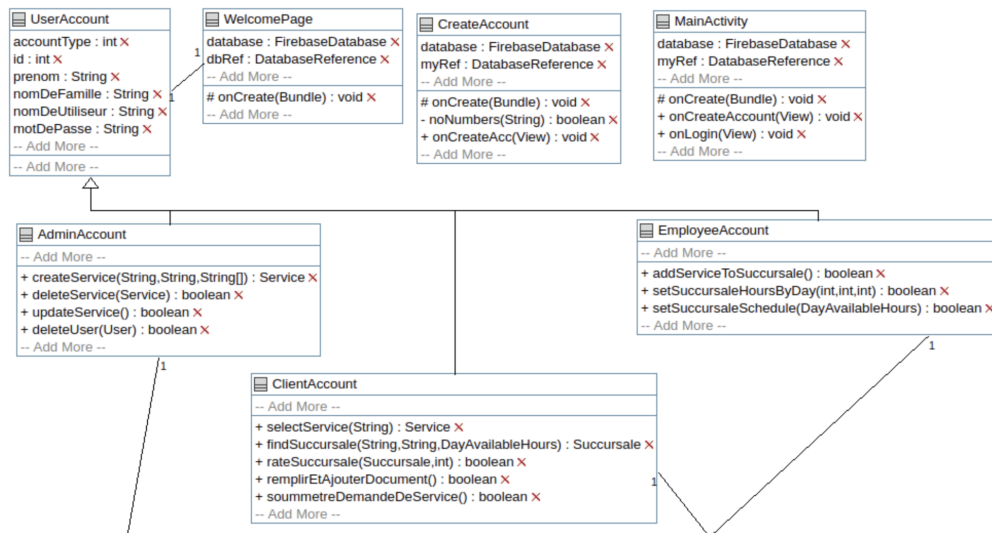


Figure 1: Diagramme UML

Evan a créé les 3 classes représentant les 3 différents types de comptes sur Android Studio qui héritent de la superclasse UserAccount.

```
public class UserAccount
{
    //-----
    // MEMBER VARIABLES
    //-----

    //UserAccount Attributes
    private int accountType;
    private int id;
    private String prenom;
    private String nomDeFamille;
    private String nomDeUtilisateur;
    private String motDePasse;

    //-----
    // CONSTRUCTOR
    //-----

    //blank constructor so that UserAccount can be written to Firebase
    public UserAccount(){}

    public UserAccount(int aAccountType, int aId, String aPrenom, String aNomDeFamille, String aNomDeUtilisateur, String aMotDePasse)
    {
        accountType = aAccountType;
        id = aId;
        prenom = aPrenom;
        nomDeFamille = aNomDeFamille;
        nomDeUtilisateur = aNomDeUtilisateur;
        motDePasse = aMotDePasse;
    }
}
```

```
public class ClientAccount extends UserAccount{

    public ClientAccount(int aId, String aPrenom, String aNomDeFamille, String aNomDeUtilisateur, String aMotDePasse) {
        super( aAccountType: 0, aId, aPrenom, aNomDeFamille, aNomDeUtilisateur, aMotDePasse);
    }

}
```

```
public class EmployeeAccount extends UserAccount{

    public EmployeeAccount(int aId, String aPrenom, String aNomDeFamille, String aNomDeUtilisateur, String aMotDePasse) {
        super( aAccountType: 1, aId, aPrenom, aNomDeFamille, aNomDeUtilisateur, aMotDePasse);
    }

}
```

```
public class AdminAccount extends UserAccount{

    public AdminAccount(int aId, String aPrenom, String aNomDeFamille, String aNomDeUtilisateur, String aMotDePasse) {
        super( aAccountType: 2, aId, aPrenom, aNomDeFamille, aNomDeUtilisateur, aMotDePasse);
    }

}
```

Figure 2: Code des méthodes Client, Employee, et Admin Account ainsi que l'initialisation des variables

2. Design des différentes activités de l'application

Céline et Samy ont fait le design des 3 activités de l'application en utilisant ce qui a été appris dans les laboratoires sur Android Studio.

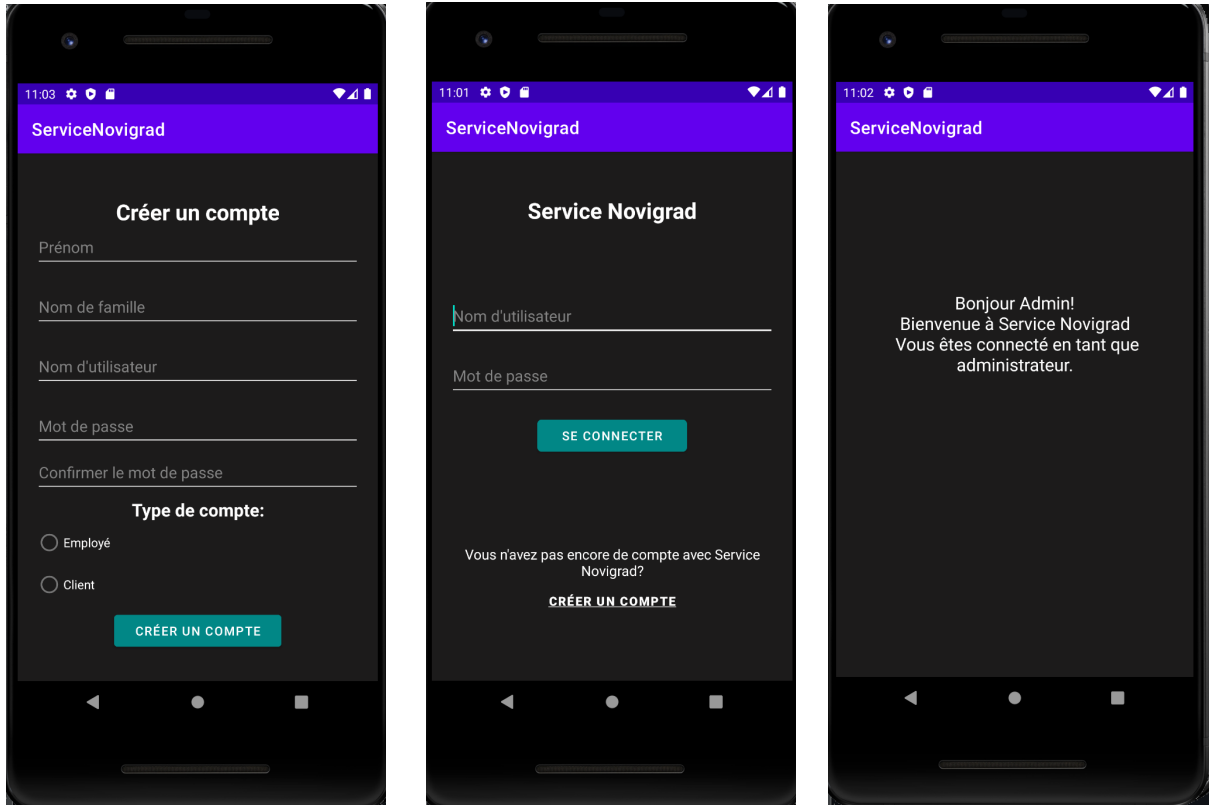


Figure 3: Frontend et design de l'application

3. Backend de l'application

Evan a travaillé sur le backend. Nous avons connecté l'application à un projet Firebase qui nous permet de garder en mémoire les comptes créés dans l'application et coder le MainActivity page et createAccount page.

```
novigrad-1b24a-default-rtdb
├── users
│   └── admin1
│       ├── accountType: 2
│       ├── id: 196884
│       ├── motDePasse: "password123"
│       ├── nomDeFamille: "Min"
│       ├── nomDeUtilisateur: "admin1"
│       └── prenom: "Admin"
```

Figure 4: Aperçu de notre base de données sur Firebase

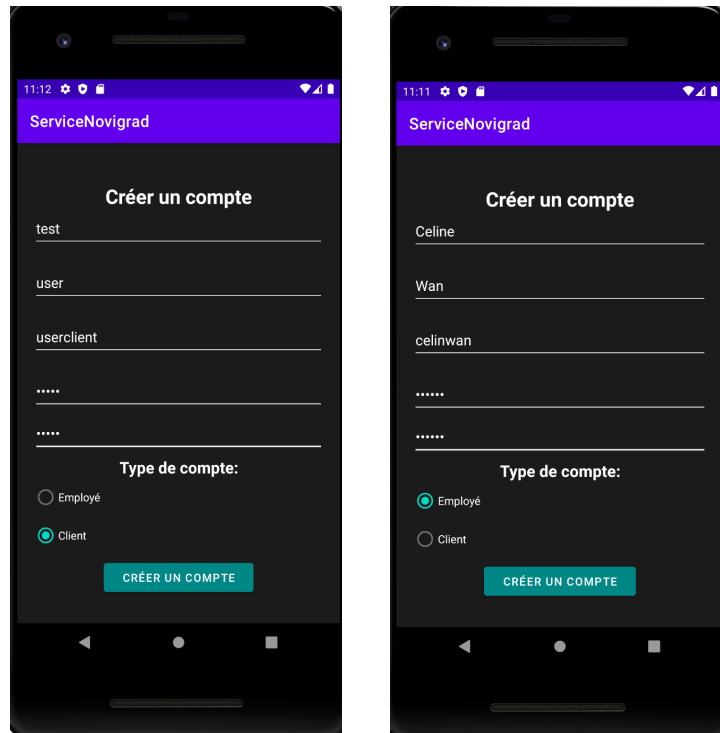


Figure 5: Ajout de 2 comptes: un client et un employé

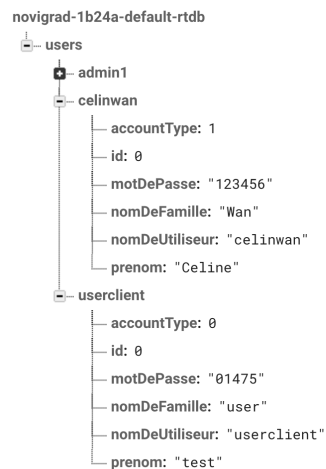


Figure 6: Aperçu de notre base de données sur Firebase après l'ajout des 2 comptes

4. Ajout de message d'erreur lors d'une champ de texte invalide

Othniel, Samy, Tisham et Céline ont ajouté des Toast messages qui s'affichent lorsqu'un mot de passe est incorrect, le nom d'utilisateur n'existe pas, il y a des champs de textes

manquants, etc... Céline a débuggé le WelcomePage car le type correct d'utilisateur n'était pas tout le temps affiché correctement, Tisham a débuggé le CreateAccount Page car les Toast messages n'apparaissaient pas tout le temps, ils apparaissaient aussi quand ce n'était pas nécessaire.

- Page de connexion

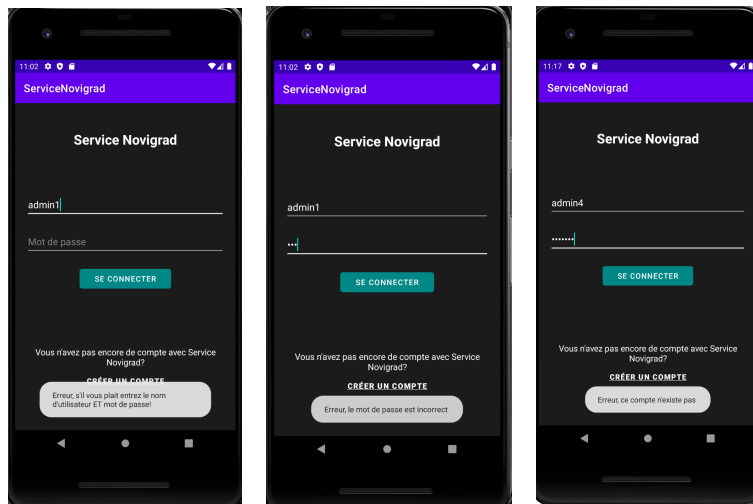


Figure 7: Démonstration des messages d'erreurs dans la page de connexion

- Page pour créer un compte

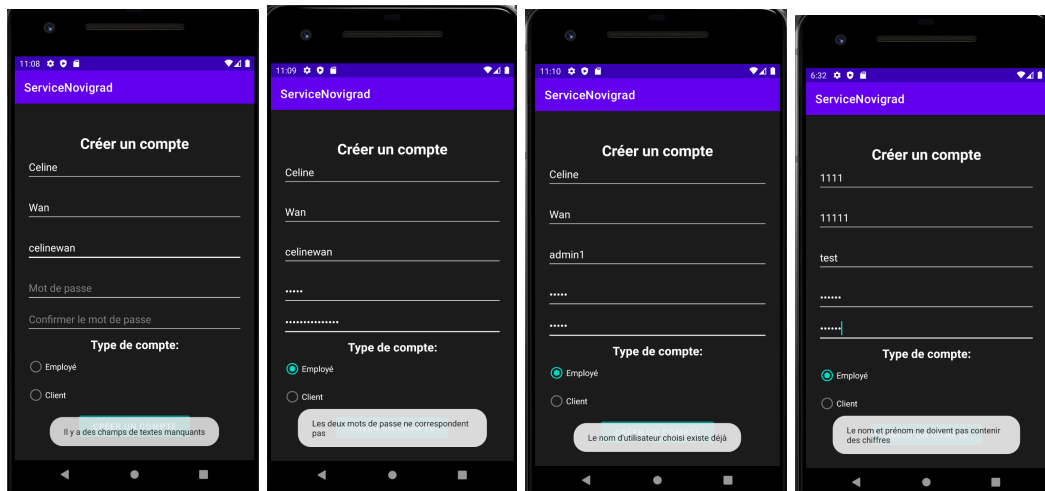


Figure 8: Démonstration des messages d'erreurs dans la page de création de compte

5. Ligne de temps de développement

- 23/10/2021: Évaluation des besoins du livrable 1
- 24/10/2021: Développement du diagramme UML initial et création du projet Android
 - Tisham, Céline, Samy, Evan ont contribué au diagramme UML
 - Céline a commencé le développement sur Android studio (Login)
- 25/10/2021: Développement front-end pour le design
 - Samy a fait le design de la page pour pouvoir créer un compte
 - Céline et Samy ont travaillé sur le design et front-end
- 26/10/2021: Développement backend avec Firebase + débogage + frontend
 - Evan s'est occupé du backend avec Firebase et a créé les classes représentées dans l'UML
 - Tisham, Samy, Othniel et Céline ont travaillé sur les messages d'erreur qui s'affiche lorsqu'un mot de passe est incorrect, le nom d'utilisateur n'existe pas, il y a des champs de textes manquants, les nom/prénom sont invalides, etc...
 - Tisham a débogué l'activité CreateAccount et Céline a débogué le WelcomePage
- 27/10/2021: Création du "release" sur Github + retouches finales au projet:
 - Tisham a ajouté, dans l'UML, les classes des activités utilisées dans Android Studio
 - Tisham a créé le "release" et ajouté l'APK finale.

6. Conclusion

Dans ce livrable nous avons commencé à développer notre application pour le Service Novigrad qui offre des services à ses clients. Nous avons implémenté la composante de gestion des comptes utilisateurs en utilisant l'outil Firebase. Grâce à Firebase, notre application garde en mémoire tous les comptes qui sont créés dans l'application même si on la quitte. Nous avons des difficultés au début à bien enregistrer les informations sur Firebase, car quand on a créé un compte, le compte ne s'affichait pas tout le temps dans la base de données. La page de connexion aussi ne marchait pas tout le temps au début car l'application ne lisait pas le nom d'utilisateur en tant que String.