

# RAPPORT FINAL : WHISLIST



- Groupe 4.3 -

DE SURAY Philippine(11091800), DEMBLON Antoine(15361800),  
DESIDE Guillaume(39731800), GIOT Adrien(23121800),  
SWIERZEWSKI Cezary(11141800)



École Polytechnique de Louvain  
LSINF1225 : Conception orientée objet & Gestion de  
données

Professeur : Kim Mens  
Assistant : Raziel Carvajal-Gomez  
Année académique 2019-2020

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Diagramme de classes</b>	<b>3</b>
<b>3</b>	<b>Modification</b>	<b>4</b>
<b>4</b>	<b>Diagramme de séquences</b>	<b>5</b>
4.1	Rajouter un ami . . . . .	5
4.2	Obtenir le prix total d'une wishlist . . . . .	6
4.3	Rechercher un produit dans une wishlist . . . . .	7
4.4	Récupérer les wishlists publiques d'un utilisateur . . . . .	8
<b>5</b>	<b>Récits utilisateurs</b>	<b>8</b>
5.1	Gestion utilisateur . . . . .	8
5.2	Gestion wishlist . . . . .	9
5.3	Gestion groupe . . . . .	9
5.4	Gestion amitié . . . . .	10
<b>6</b>	<b>Schéma ORM</b>	<b>10</b>
6.1	Modification . . . . .	11
<b>7</b>	<b>Schéma relationnel</b>	<b>12</b>
7.1	Modifications . . . . .	12
<b>8</b>	<b>Manuel d'utilisation</b>	<b>12</b>
8.1	Connexion . . . . .	12
8.2	Ecran de départ : gestion des wishlists . . . . .	13
8.3	Profil de l'utilisateur . . . . .	14
8.4	Liste d'amis . . . . .	15
8.5	Produits . . . . .	16
8.6	Groupes . . . . .	17
<b>9</b>	<b>Choix de conception</b>	<b>17</b>
9.1	Schéma ORM . . . . .	17
9.1.1	Diagramme de séquence . . . . .	17
9.2	Programmation . . . . .	18
9.2.1	Logo . . . . .	18
9.2.2	Diagramme de classe . . . . .	18
9.2.3	Couche intermédiaire . . . . .	18
9.2.4	Base de données . . . . .	18

9.2.5 Récupération d'images . . . . .	19
<b>10 Travail de groupe</b>	<b>19</b>
<b>11 Conclusion</b>	<b>19</b>
<b>12 Bibliographie</b>	<b>20</b>

# 1 Introduction

Après de nombreuses semaines de travail, notre projet touche à sa fin ! Nous avons le plaisir de vous présenter WISH WORLD, notre produit final. Mais avant de profiter de ses nombreuses fonctionnalités, voici un rapport qui vous permettra de comprendre l'envers du décor et le travail fourni pour aboutir à une telle application.

Nous commencerons par vous présenter notre diagramme de classes UML et nos diagrammes de séquences tout en expliquant les modifications apportées afin que ceux-ci correspondent au mieux au produit final. Suivront nos récits utilisateurs et notre schéma ORM final. Un manuel d'utilisation vous détaillera la prise en main de notre application. Enfin, nous terminerons par vous expliquer nos choix de conceptions, ainsi que notre manière de travailler en groupe et une petite conclusion.

# 2 Diagramme de classes

Voici le diagramme de classes UML correspondant à notre application. On y retrouve les objets nécessaires au fonctionnement de l'application avec les attributs et les méthodes utilisées.

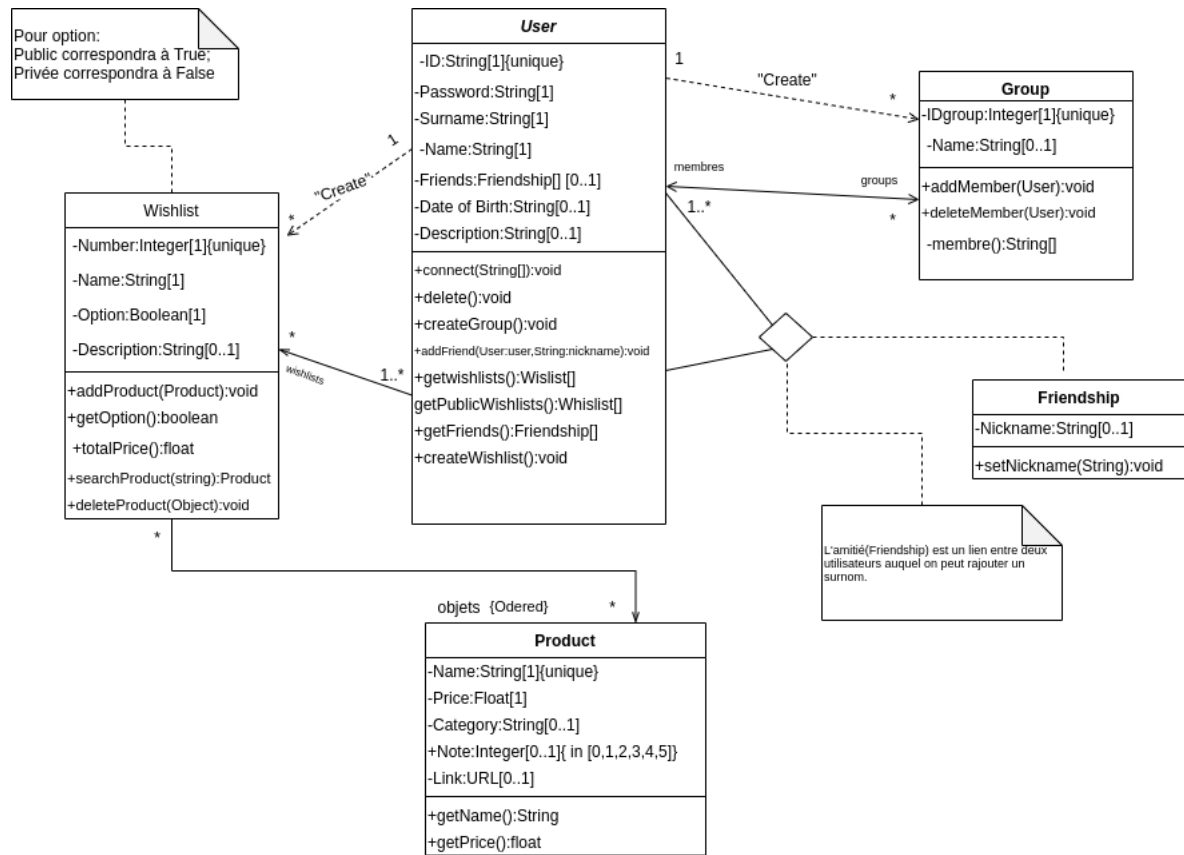


FIGURE 1 – diagramme de classes UML de notre application

### 3 Modification

La première modification effectuée par dépôt, est d'enlever les attributs en lien avec les images. Ce choix a été fait par manque de temps et d'informations pour stocker les images dans les bases de données. Des informations complémentaires sont données dans la partie 9.2.5 de ce rapport.

Une autre modification a été de redéfinir les noms des méthodes qui n'avaient pas de sens. On est revenu à get et set pour certaines.

Les dates d'anniversaire(DOB) des utilisateurs sont stockées sous forme de string à cause du choix de Room comme couche intermédiaire.

Comme souvent avec des logiciels comme android studio, ceux-ci préfèrent quand les attributs sont au maximum mis en "private". Afin d'éviter d'avoir à chaque fois la recommandation qui apparaissent, tous les attributs sont en privé. Cette manière de faire ne change pas grand chose vu l'utilisation de getter et de

setter pour une grande partie des attributs.

## 4 Diagramme de séquences

Dans cette section, vous retrouvez quelques diagrammes de séquence UML en rapport avec les récits utilisateurs qui se trouvent à la section 5 et des explications supplémentaires sont dans la section 9 "choix de conception".

### 4.1 Rajouter un ami

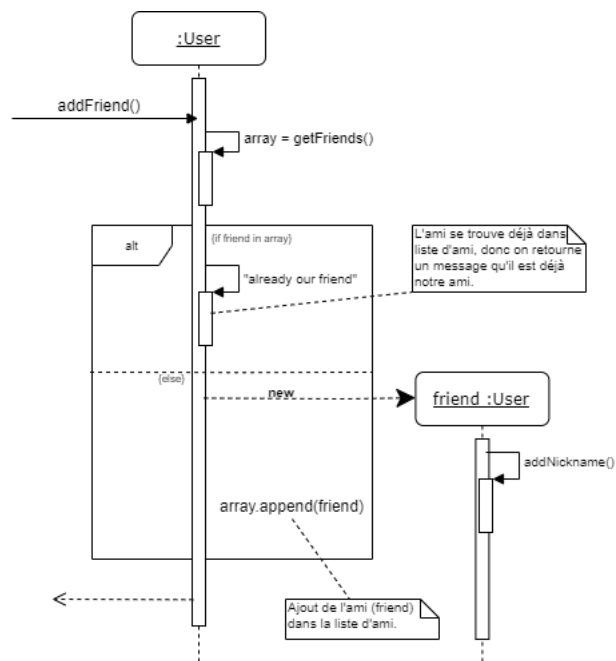


FIGURE 2 – Diagramme de séquence-Rajouter un ami

## 4.2 Obtenir le prix total d'une wishlist

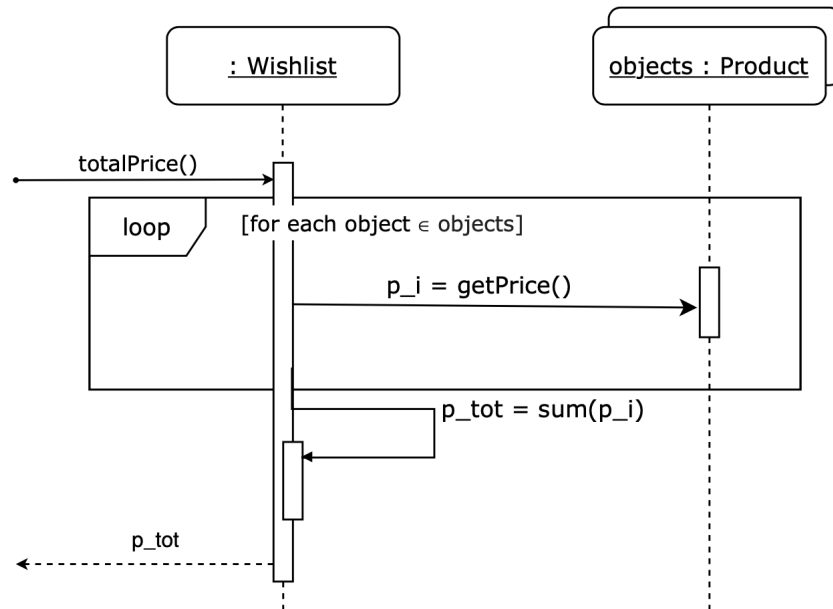


FIGURE 3 – Diagramme de séquence-Obtenir le prix total d'une wishlist

### 4.3 Rechercher un produit dans une wishlist

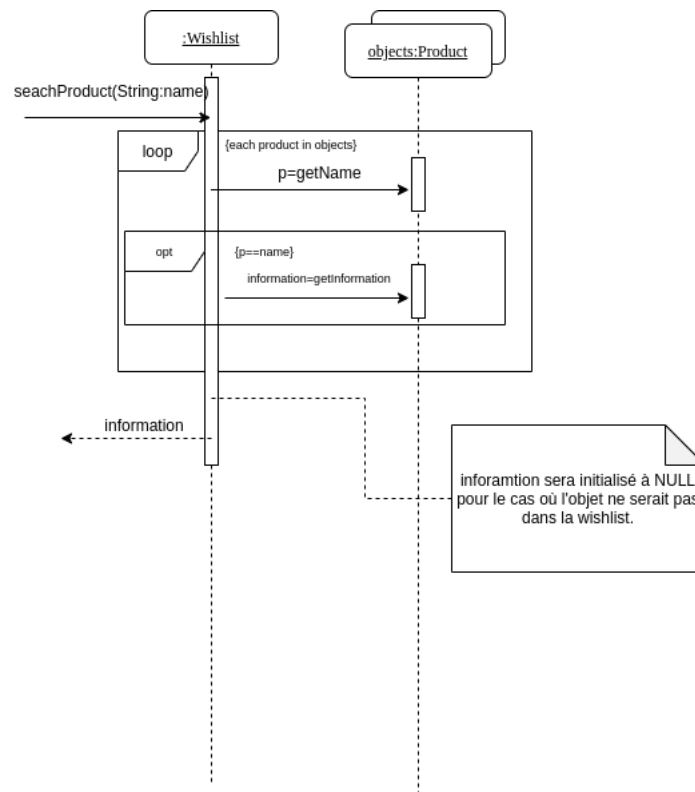


FIGURE 4 – Diagramme de séquence-Rechercher un produit dans une wishlist



## 4.4 Récupérer les wishlists publiques d'un utilisateur

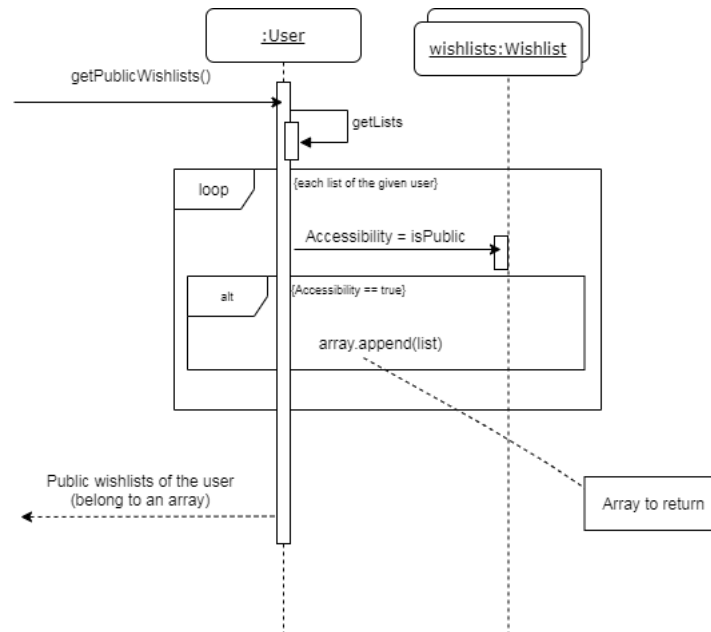


FIGURE 5 – Diagramme de séquence-Récupérer les wishlists publiques d'un utilisateur

## 5 Récits utilisateurs

Dans cette partie du rapport, vous trouverez les récits utilisateurs qui caractérisent notre application. Ils se divisent en 4 parties : gestion utilisateur, gestion wishlist, gestion des groupes et gestion amitié.

### 5.1 Gestion utilisateur

- Afin de pouvoir accéder à l'application, en tant qu'utilisateur, lors du démarrage de l'application, je dois pouvoir m'y connecter ou créer un compte.
- Afin de modifier mes informations personnelles, en tant qu'utilisateur, je dois avoir accès à une interface qui me permet de les modifier.
- Afin de se déconnecter du compte courant, en tant qu'utilisateur, je dois avoir accès à un bouton permettant de le faire.
- Afin de supprimer mon compte, en tant qu'utilisateur, je dois avoir accès à un bouton permettant d'exécuter cette tâche.

## 5.2 Gestion wishlist

- En tant qu'utilisateur, afin de rassembler des objets que je veux, je dois pouvoir créer une liste de souhait.
- En tant qu'utilisateur, afin de mettre à jour les informations d'un objet d'une de mes wishlists, je dois pouvoir avoir accès aux informations de l'objet et les modifier.
- En tant qu'utilisateur, afin de compléter/modifier ma wishlist, je dois être capable d'ajouter et de supprimer des objets à une liste.
- En tant qu'utilisateur, afin de mettre à jour une de mes wishlists, je dois pouvoir modifier les spécifications d'une liste quand j'appuie sur le bouton edit.
- En tant qu'utilisateur, afin de retrouver un objet grâce à son nom, je dois pouvoir entrer son nom dans une barre de recherche.
- En tant qu'utilisateur, afin de savoir toutes les wishlists auquel j'ai accès, je dois pouvoir obtenir une liste de l'ensemble des listes de souhaits auquel j'ai accès en allant dans l'interface wishlist.
- En tant qu'utilisateur, afin de savoir si l'ensemble de la wislist rentre dans mon budget, je dois pouvoir obtenir le prix total d'une wishlist.
- En tant qu'utilisateur, afin de savoir ce qu'un ami veut, je dois pouvoir consulter ses wishlists publiques.

## 5.3 Gestion groupe

- En tant qu'utilisateur, afin de regrouper plusieurs amis pour partager des wishlists, je dois pouvoir créer un groupe.
- En tant que membre d'un groupe, afin de savoir ce que désire un autre membre d'un groupe, je dois avoir accès aux wishlists publiques des autres membres.
- En tant qu'utilisateur, afin de savoir dans quels groupes je suis, je dois pouvoir obtenir une liste de tous les groupes dans lesquels je suis membre en allant dans mon onglet groupe.
- En tant que membre d'un groupe, afin de supprimer un membre(dont moi-même), je dois pouvoir avoir accès aux membres et choisir lequel ne fait plus partie du groupe.

## 5.4 Gestion amitié

- En tant qu'utilisateur, afin d'augmenter mon cercle d'amitié ou le diminuer, je dois être capable d'ajouter et supprimer des amis.
- En tant qu'utilisateur, afin d'obtenir des informations sur mon ami, je dois pouvoir consulter son profil avec toutes ses informations.
- En tant qu'utilisateur, afin de savoir avec qui je suis amis, je dois pouvoir obtenir une liste précise de tous mes amis dans l'onglet "amis de l'application".

## 6 Schéma ORM

Dans cette section, nous retrouvons notre schéma ORM qui a été présenté dans le premier rapport mais avec quelques modifications.

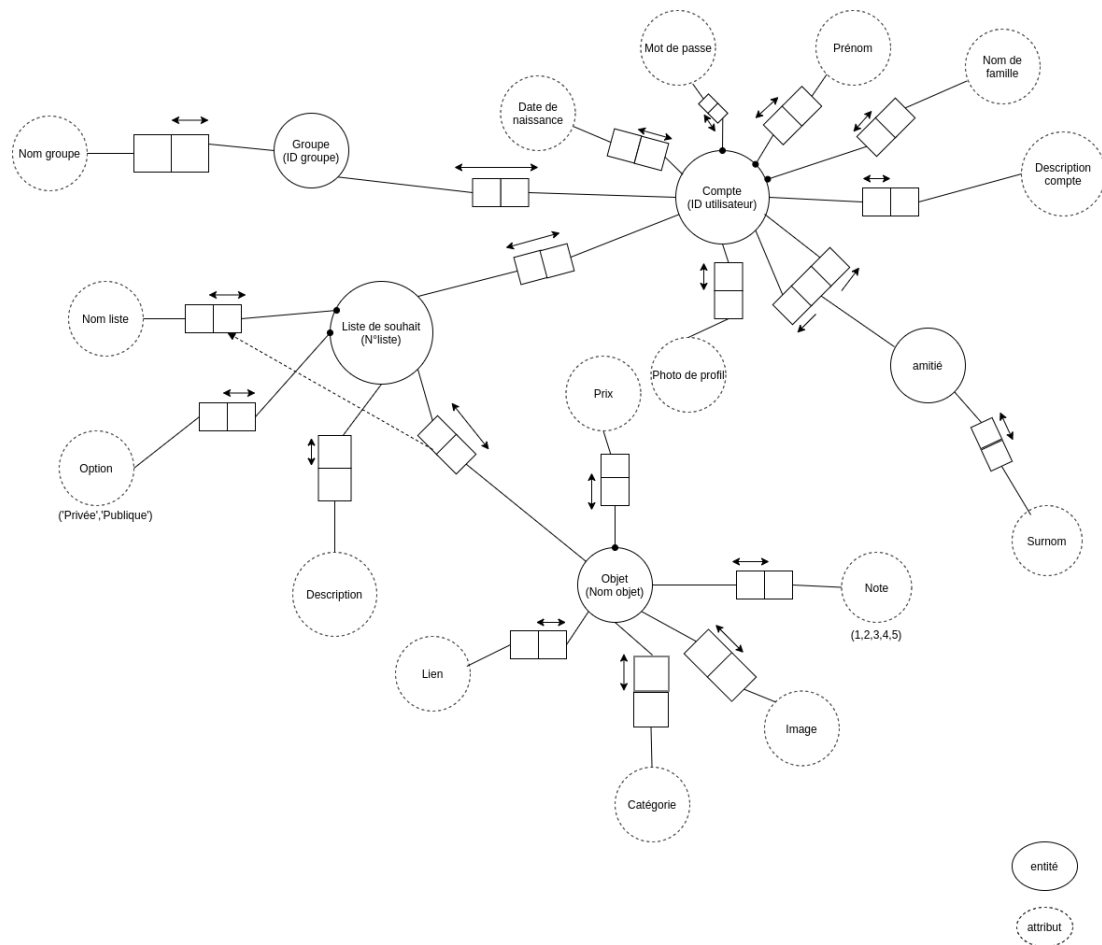


FIGURE 6 – schéma ORM final de notre application

## 6.1 Modification

Nous avons modifié la relation entre compte et amitié. Via ce changement, nous insistons sur le fait que la relation d'amitié est bien une relation entre deux comptes(utilisateurs) avec un surnom pour la caractériser. La relation caractérisant les groupes a aussi été changée car elle n'avait peu de sens dans l'ancien état. C'est bien un utilisateur qui appartient à un groupe et non une amitié.

## 7 Schéma relationnel

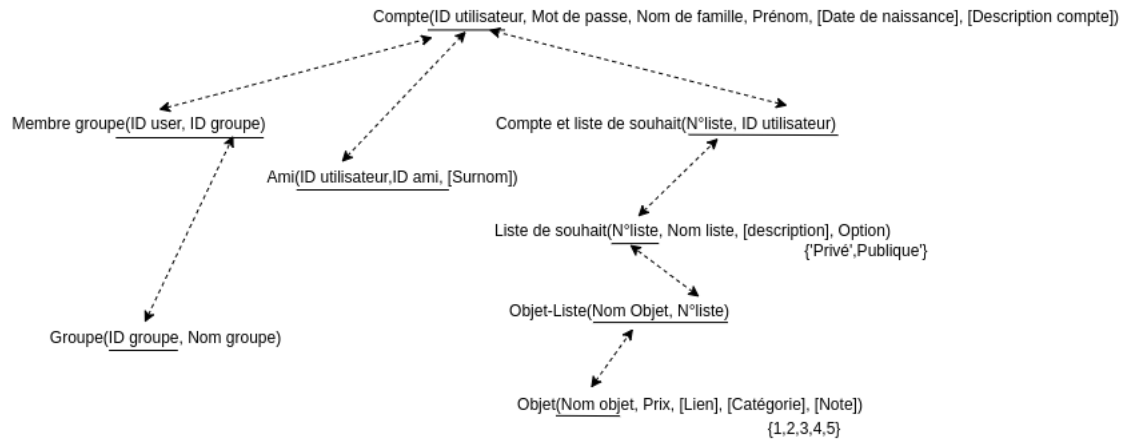


FIGURE 7 – schéma relationnel de notre application

### 7.1 Modifications

Comme pour le schéma ORM afin de mieux correspondre à notre idée d'un groupe, nous avons changé le lien entre groupe et utilisateur. En effet c'est bien un utilisateur qui appartient à un groupe et non un ami qui appartient à un groupe. Les colonnes correspondants à des images ont été supprimées.

## 8 Manuel d'utilisation

Dans cette section, vous trouverez un petit manuel afin d'expliquer le fonctionnement de l'application, comment en profiter un maximum, etc...

Préalable : Pour lancer l'application via android studio, il faut d'abord ouvrir le projet : pour cela, il faut aller dans *file -> open...* puis choisir le dossier "*Wishlist*" compris dans le dossier "git" de notre projet.

On build ensuite le projet et on le lance sur l'émulateur de notre choix préalablement installé.

### 8.1 Connexion

Quand vous ouvrez l'application, l'écran d'accueil permet à l'utilisateur de se connecter (-> LOG IN), en indiquant son identifiant et son mot de passe (-> IDENTIFIANT et PASSWORD respectivement), ou de lancer la création d'un compte (-> CREATE ACCOUNT).

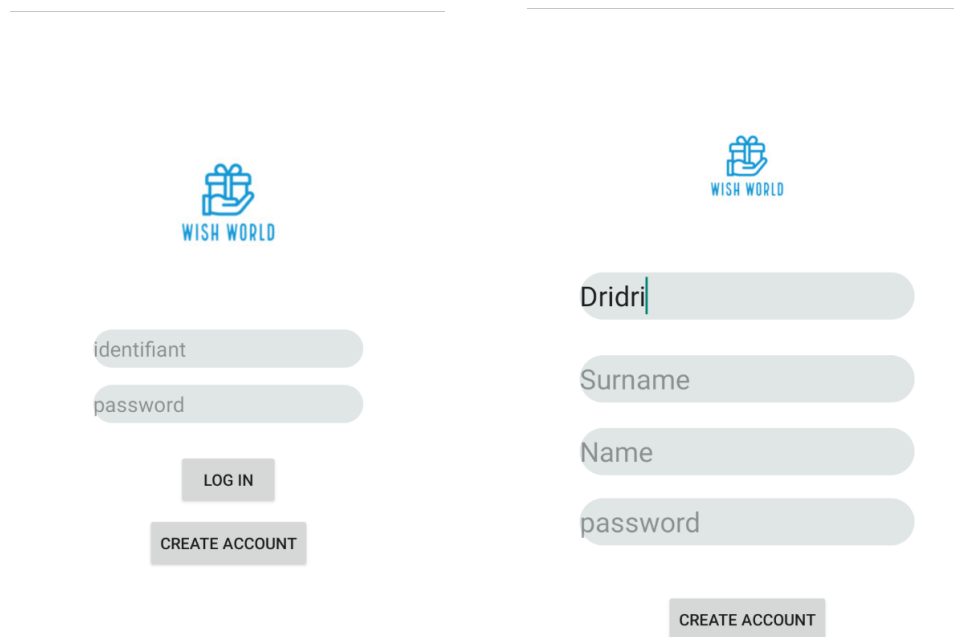


FIGURE 8 – Ecrans de connexion et création de compte

Si vous souhaitez créer un compte, vous arrivez sur un écran avec 4 champs à compléter (→ IDENTIFIANT, PRENOM, NOM et PASSWORD). Tous les champs doivent être remplis. Si l'identifiant renseigné est déjà pris, l'application vous l'indique pour que vous le changiez. Notez que vous pouvez revenir à l'écran d'accueil avec le bouton en bas à gauche. Une fois le compte créé (via CREATE ACCOUNT), vous êtes renvoyé à l'écran d'accueil, où vous pouvez vous connecter avec les informations saisies juste avant.

Une fois connecté, vous arrivez sur l'écran de départ. La barre du bas permet d'accéder à d'autres écrans. On va les décrire de gauche à droite.

## 8.2 Ecran de départ : gestion des wishlists

Le bouton d'extrême gauche donne accès à l'écran de départ lui-même. Ce dernier affiche les wishlists de l'utilisateur. En cliquant sur l'icône liée à une wishlist, on voit s'afficher les objets de la liste, et le coût total de la liste. Un objet peut être ajouté à la liste au départ de ce dernier écran.

Une wishlist peut être créée via un écran dédié, auquel on accède par le bouton vert. Le nom doit être unique (pour les wishlists de l'utilisateur).

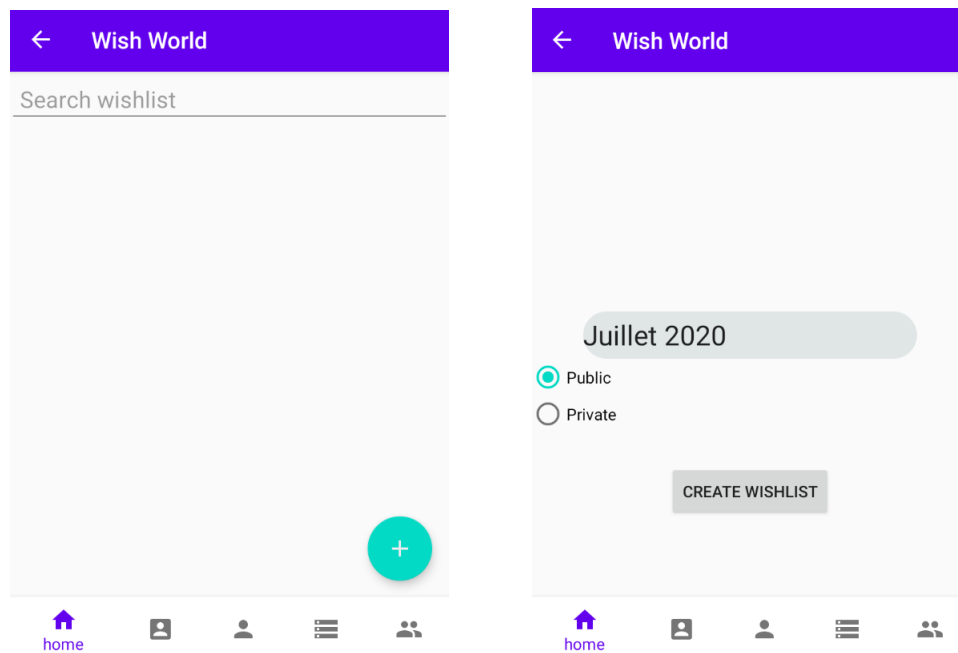


FIGURE 9 – Ecrans de gestion de Wishlists

### 8.3 Profil de l'utilisateur

Le bouton suivant débouche sur un écran qui affiche les informations relatives au profil de l'utilisateur.

Elles peuvent être modifiées par un écran dédié, auquel on accède par le bouton outil. Par ailleurs, le profil peut être supprimé en cliquant sur un bouton poubelle au milieu à droite

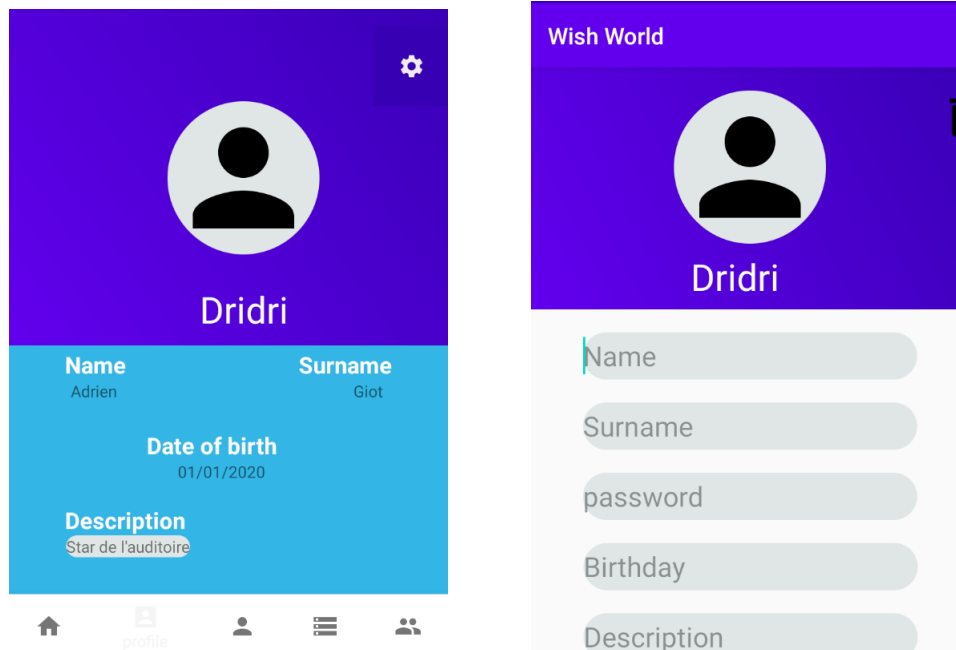


FIGURE 10 – Ecrans de gestion de Wishlists

## 8.4 Liste d'amis

On accède ensuite à l'écran relatif aux amis par le bouton du centre. Cet écran affiche une liste des amis. En cliquant sur l'icône relatif à un ami puis sur l'icône wishlist affiché, on peut voir ses listes publiques. En cliquant sur le bouton paramètre, on peut donner un nouveau surnom à l'ami.

On accède à l'écran où ajouter un ami en cliquant sur le bouton vert. Un surnom peut être donné. CREATE FRIENDSHIP permet de créer l'amitié. Si l'username est invalide, c'est signalé à l'utilisateur et il n'y a donc pas de création.



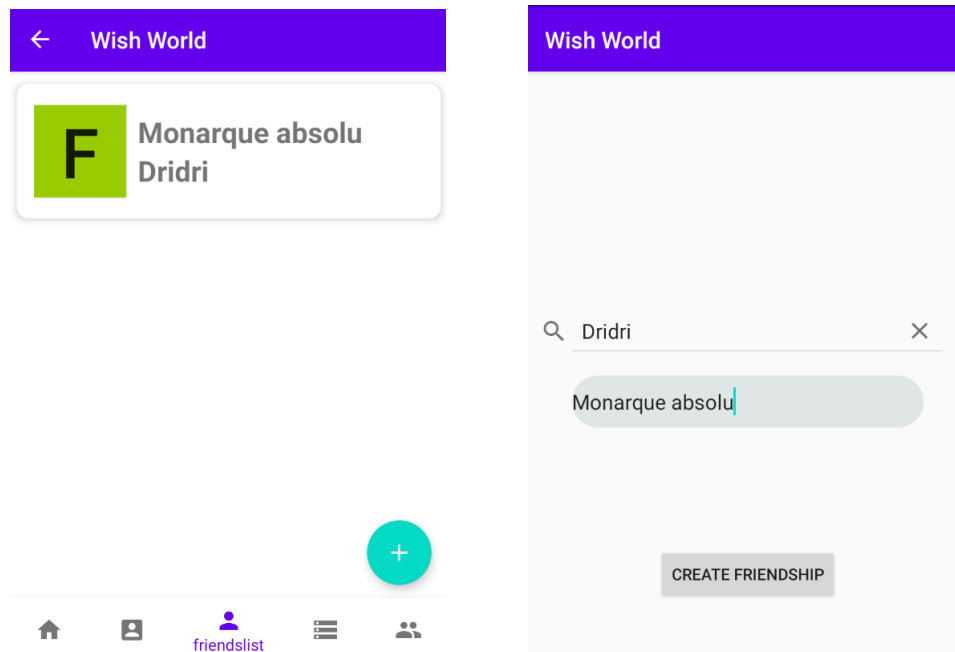


FIGURE 11 – Ecrans de gestion de la liste d'amis

## 8.5 Produits

L'avant-dernier bouton de la barre du bas donne sur un écran qui affiche les produits. On peut faire défiler la liste.

De là, on peut créer d'autres produits dans un écran accessible en cliquant sur le bouton vert. Il suffit de préciser le prix et le nom du produit. Notez que le nom doit être unique à tous les produits, sinon la création est refusée. On peut enfin modifier un produit et lui attribuer une note.

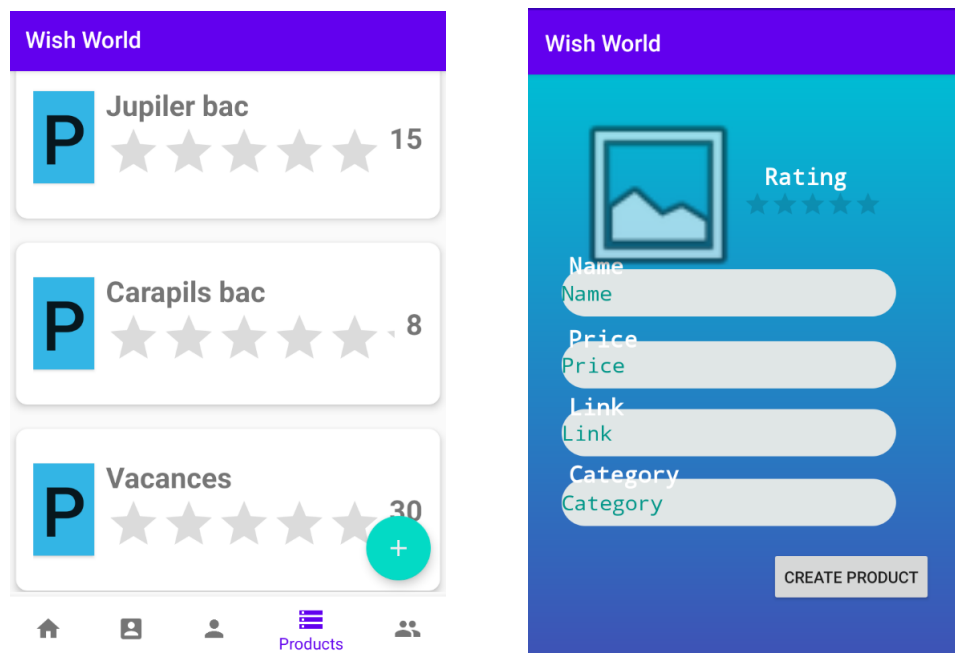


FIGURE 12 – Ecrans de gestion des produits

## 8.6 Groupes

Un dernier bouton a été placé pour l'implémentation de groupes. Il n'est pas fonctionnel sur le prototype.

## 9 Choix de conception

Beaucoup de choix de conception ont été expliqués dans les rapports précédents, mais nous avons décidé d'en rappeler certains et de les compléter.

### 9.1 Schéma ORM

Nous rappelons qu'une wishlist peut être modifiée par différents utilisateurs dans le cas d'un groupe. Cette idée nous semble intéressante à implémenter.

#### 9.1.1 Diagramme de séquence

Ceux-ci représentent certains faits élémentaires que nous avons cité dans la section 5. Dans l'implémentation de l'application avec Room, il devenait dur de

représenter ce qu'il se passait réellement lors de l'exécution de certaines fonctions. Donc nous avons décidé n'appliquer aucune modification à nos diagrammes de séquence.

## 9.2 Programmation

Pour cette partie, nous avons utiliser plusieurs vidéos que nous avons regroupé dans une playlist[2].

### 9.2.1 Logo

Nous avons de décider de modifier notre logo de base. Ce changement s'est fait sur un site[3] permettant de créer des logos gratuitement.

### 9.2.2 Diagramme de classe

Dans le cadre de l'utilisation de Room, nos liens/dépendances entre les différentes classes ont été remplacés lors de la programmation par des classes intermédiaires car nous n'aurions pas su rajouter les liens dans la base de données. Donc dans l'implémentation de notre application, nous avons beaucoup plus de classes. En effet, aux différentes classes présentent dans le diagramme de classes, nous avons dû rajouter les repository, les classes Database et les DAO. Les fonctions autres que set et get ont dû être implementées dans les classes liées à la base de données.

### 9.2.3 Couche intermédiaire

Dans le cadre de ce projet, il nous était conseillé de programmer avec une couche intermédiaire pour la base de données. Nous avons regardé les différentes options qui se proposaient à nous. Nous nous sommes arrêté sur Room qui est conseillé sur le site de developpement android[1]. Grace à ce choix, il nous était plus facile de se concentrer sur la programmation Java et non sur les commandes SQL.

### 9.2.4 Base de données

En utilisant Room et n'ayant pas toutes les connaissances sur ce qu'on pouvait faire, nous avons fait le choix pour nous simplifier la tâche, de ne pas utiliser une seule base de données mais d'en utiliser plusieurs. Chaque base de données correspond à un tableau de notre base de données initiale.

### 9.2.5 Récupération d'images

C'est la partie qui nous a causé des problèmes. Nous n'avons pas trouvé de tutorial nous permettant de stocker les images via Room. Les seuls tutorials à notre disposition nous permettaient seulement de récupérer des images du GSM mais pas de les stocker dans une base de données. Donc nous avons décidé de passer cette partie pour se concentrer sur les autres parties.

## 10 Travail de groupe

Nous pouvons que nous réjouir de l'implication de chacun dans le projet au fil des semaines, chaque membre du groupe a trouvé facilement une place qui lui correspondait. Au final, tout le monde aura apporté sa pierre à l'édifice, permettant la réalisation du travail dans une bonne entente et cohésion d'équipe.

Ce travail nous aura aussi demandé de jongler entre rigueur, organisation et cohésion. Ce défi fut d'autant plus important cette année au vu de la situation actuelle. Nous avons pu développer des compétences en télé-travail et avons su nous en accoutumer et parvenir à trouver une bonne communication à travers l'outil "Teams".

## 11 Conclusion

En conclusion, ce projet nous a permis d'apprendre à gérer un projet informatique de A à Z. Nous avons découvert les étapes de développement qui précèdent la programmation. Nous avons commencé par la conception de la base de données et du schéma ORM. Pour ensuite, se concentrer sur la partie conception orientée objet du projet avec les diagrammes de classes UML et de séquences. La dernière étapes fut la programmation Android.

Dans le groupe, tout le monde s'est impliqué à sa manière au fil des étapes. Nous avons réussi à bien nous organiser et à communiquer. Nous sommes fiers du travail fourni tout au long du quadrimestre et du résultat final accompli.

Nous vous laissons maintenant découvrir et expérimenter notre application et ses nombreuses fonctionnalités. Nous espérons qu'elle répondra à vos attentes et se montrera utile.

## 12 Bibliographie

### Références

- [1] Android DEVELOPERS. *Site de developpement android*. URL : <https://developer.android.com/training/data-storage/room>.
- [2] Deside GUILLAUME. *Playlist des vidéos utilisées pour la programmation*. URL : <https://www.youtube.com/playlist?list=PLie58dW7C4tA1K1G7bLBNbb1XKjzKyTt1&jct=QjtUEn98IxlhNY3DRz9LpIO5SBeh6g>.
- [3] Auteur INCONNU. *Site de création de logo*. URL : <https://studio.tailorbrands.com/brands/free>.