

# RAPPORT : WHISLIST



- Groupe 4.3 -

DE SURAYŸ 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>2</b>
<b>2</b>	<b>Mockup</b>	<b>2</b>
<b>3</b>	<b>Diagramme des dépendances</b>	<b>4</b>
<b>4</b>	<b>Faits élémentaires</b>	<b>5</b>
<b>5</b>	<b>Schéma conceptuel</b>	<b>6</b>
<b>6</b>	<b>Schéma relationnel</b>	<b>8</b>
<b>7</b>	<b>Choix de conception</b>	<b>8</b>
7.1	Mockup . . . . .	8
7.2	Faits élémentaires, ORM et schéma relationnel . . . . .	8
7.3	Database . . . . .	9
7.4	Autres points . . . . .	9
<b>8</b>	<b>Conclusion</b>	<b>9</b>
<b>9</b>	<b>Lexique</b>	<b>10</b>
<b>10</b>	<b>Bibliographie</b>	<b>11</b>

# 1 Introduction

Nous avons le plaisir de vous présenter notre projet d'application wishlist réalisé dans le cadre du cours LSINF1225. Ce projet est à l'intersection de nos connaissances théoriques apprises lors des différents cours et de notre créativité. L'objectif principale de ce projet est de répondre aux attentes du cahier des charges fourni par nos nouvelles connaissances et l'apport de nos idées créatives et ingénieuses.

Concernant la structure du rapport, vous trouverez en premier lieu le mockup qui permet de se faire une première idée du design de la future application et des différents lien entre les écrans. Puis s'en suit la partie des faits élémentaires qui décrit les liens utiles pour notre application. Ensuite, on retrouvera le schéma conceptuel (ORM) et le schéma relationnel. Après on exposera les choix de conception faits par le groupe. Enfin, le rapport se terminera par une brève conclusion reprenants les principaux points.

# 2 Mockup

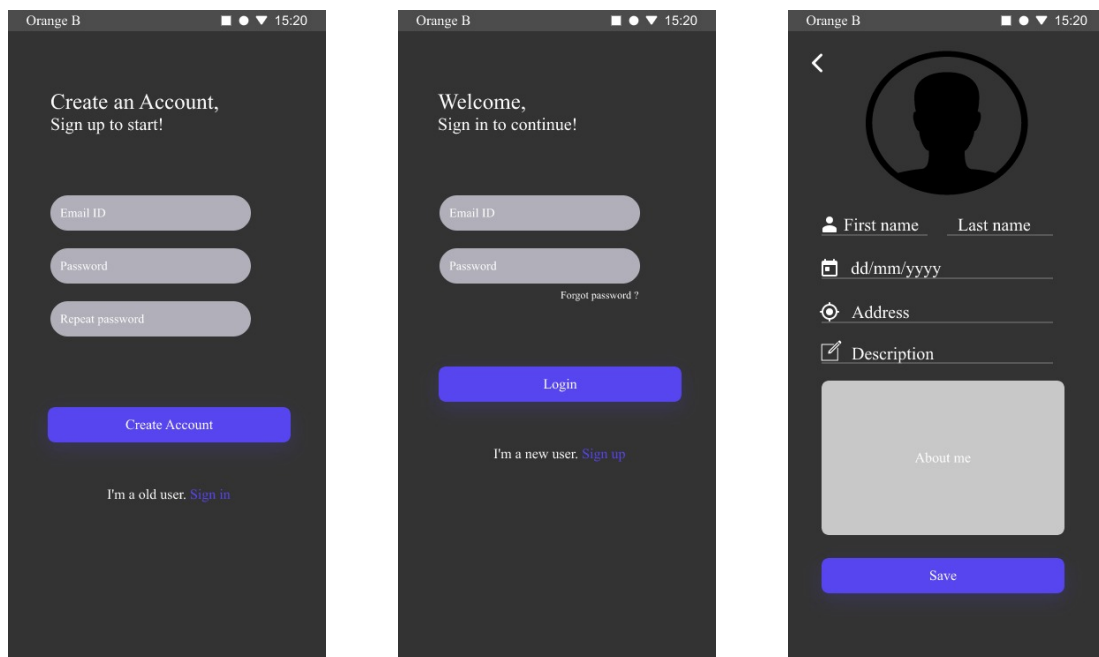


FIGURE 1 – Interfaces d'enregistrement

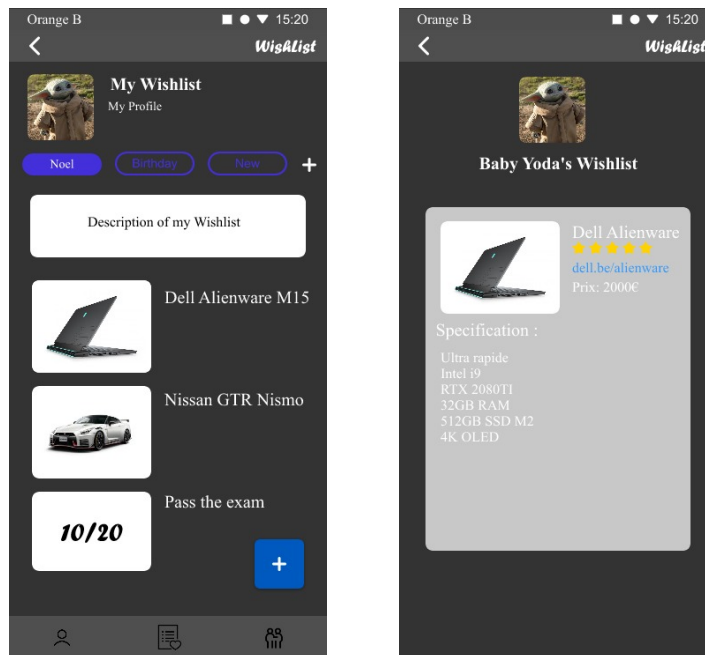


FIGURE 2 – Profile

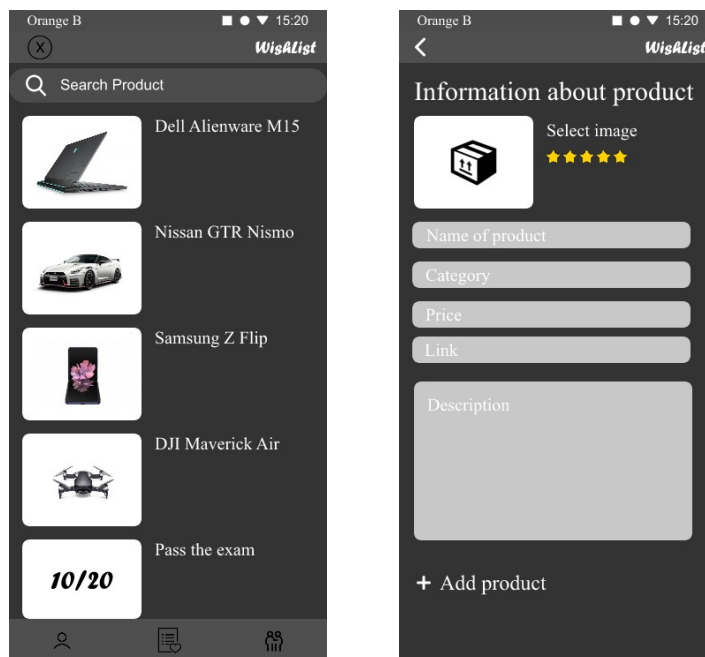


FIGURE 3 – Page d'accueil et ajout de produit

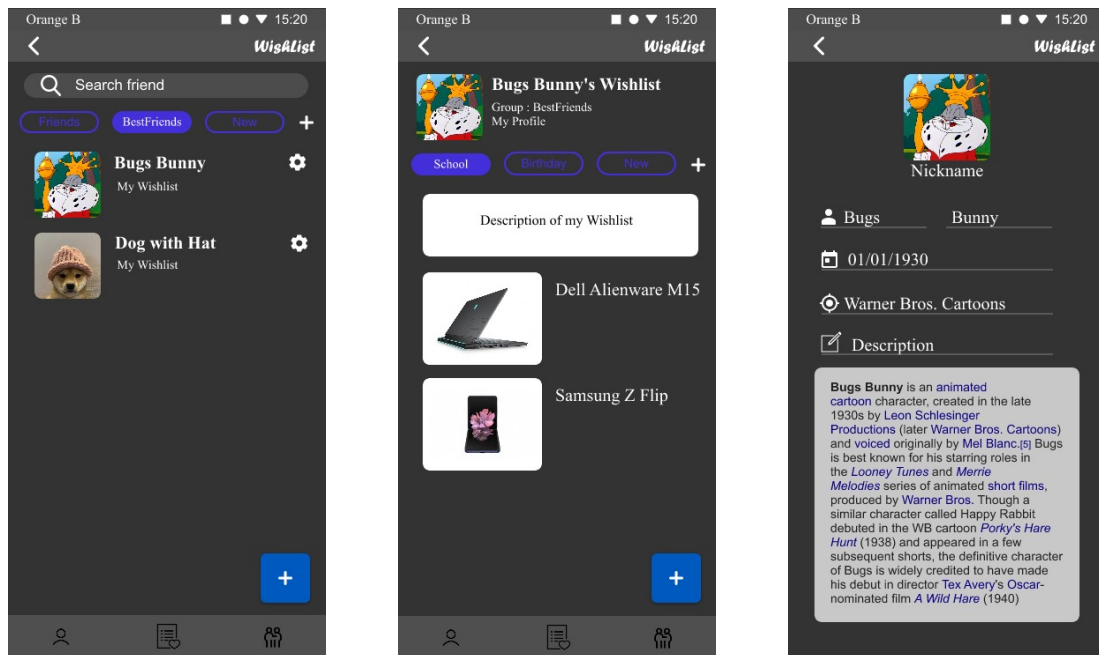
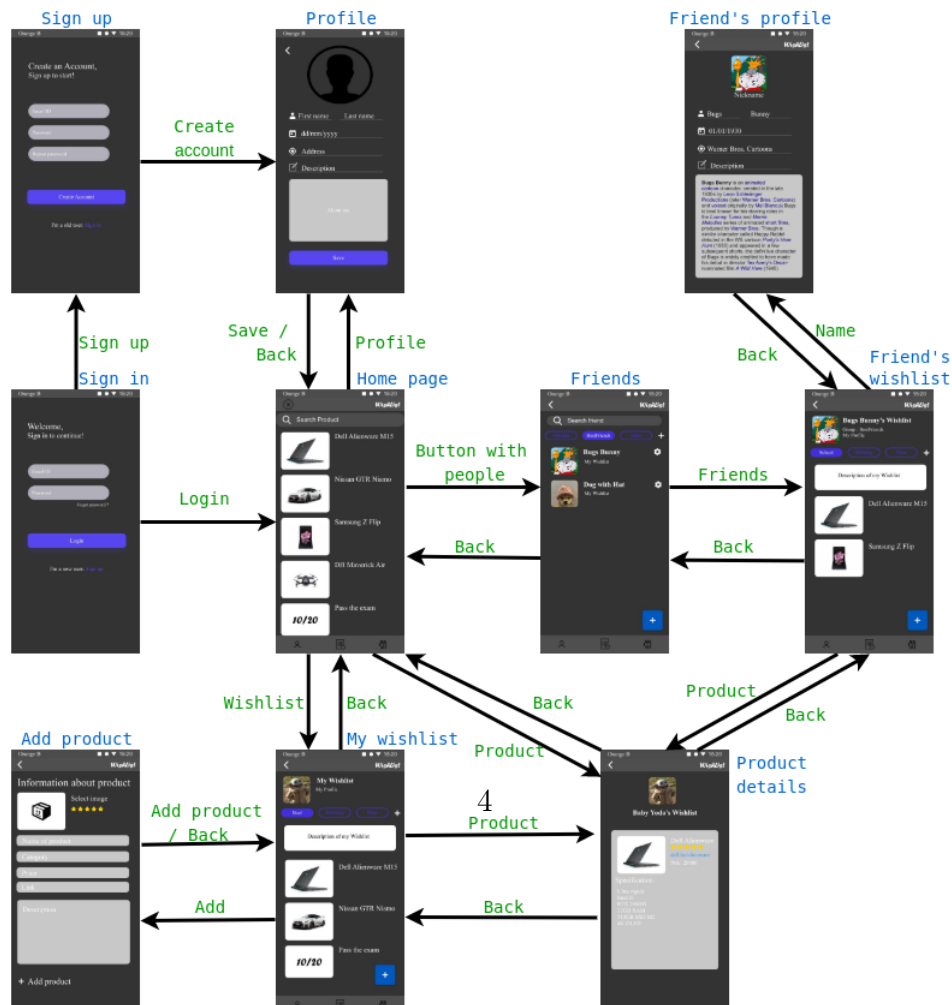


FIGURE 4 – Amis

### 3 Diagramme des dépendances



- On va décrire les interactions entre les différents écrans du mockup :
- Si l'utilisateur n'est pas connecté, il arrive dans l'écran "Sign up".
  - > S'il a un compte, il peut se connecter en passant à l'écran "Sign in".
  - > Sinon, il crée son compte, puis est renvoyé vers son profil pour le compléter (il doit indiquer un nom et un prénom).
  - Une fois qu'il s'est connecté, l'utilisateur arrive sur l'écran représentant la page principale(home page).
  - Sur cet écran, il peut rechercher des objets via la barre de recherche. En cliquant sur un objet, la wishlist dont l'objet provient est affichée.
  - On va maintenant voir ce que font les boutons de la page principale :
  - > En cliquant sur le bouton du centre en bas, l'utilisateur accède à ses wishlists. Il peut consulter les objets des wishlists en cliquant sur les icônes qui les représentent.
  - > En cliquant sur le bouton du centre à droite, il peut accéder à sa liste d'amis dans un nouvel écran. En cliquant sur un ami, il accède à sa liste de souhaits et de là s'il clique sur un ami, il accède à son profil. Il peut ajouter un ami avec le bouton du bas à droite
  - > En cliquant sur le bouton du centre à gauche, l'utilisateur accède à son propre profil, qu'il peut mettre à jour.
- Dans les pages qui complètent la page principale, on peut revenir à la page précédente avec le bouton du haut à gauche.
- Dans la partie wishlist :
- > on peut ajouter un objet en utilisant le bouton '+'.
  - > on clique sur un objet, on peut obtenir les informations dessus.

## 4 Faits élémentaires

Voici un ensemble de faits élémentaires pour notre application :

### **Pour les entités :**

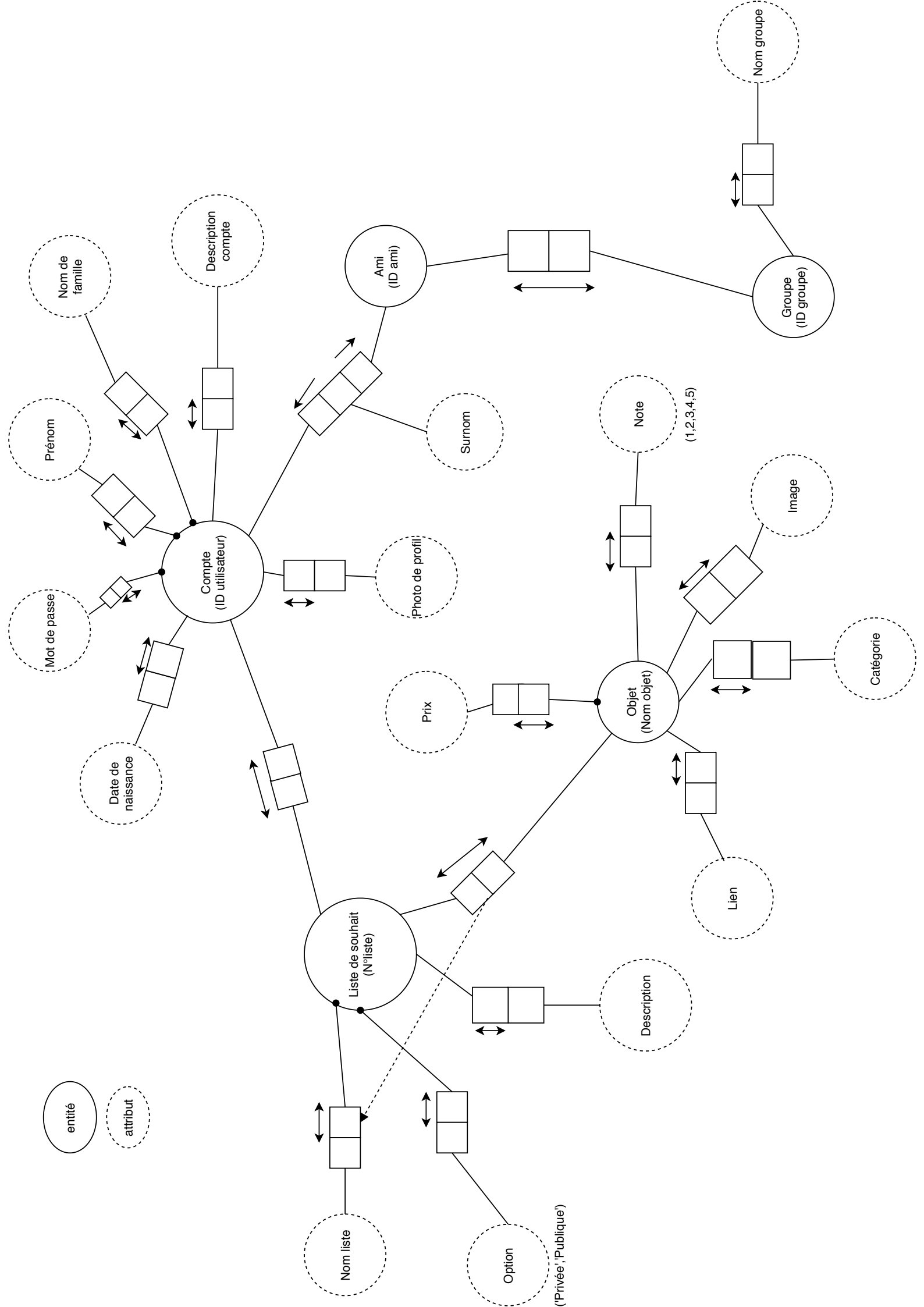
- Un ami peut apparaître une seule fois dans un groupe
- Un compte peut avoir des amis avec un surnom
- Un compte a des listes de souhaits
- Un objet peut apparaître seulement une seule fois dans une liste de souhait
- Seulement les listes stockées peuvent avoir un objet

### **Les attributs des entités :**

- Un compte doit avoir un identifiant unique
- Un compte doit avoir un nom
- Un compte doit avoir un prénom

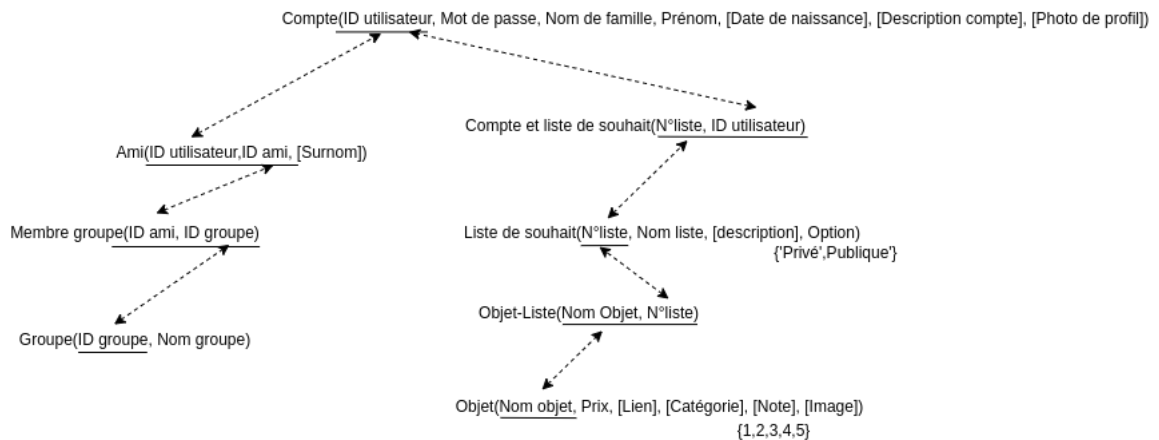
- Un compte peut avoir une image
- Un compte doit avoir un mot de passe
- Une liste doit être publique ou privée
- Une liste doit avoir un numéro unique(numéro liste)
- Un objet a un nom unique(explication dans choix de conception)
- Un groupe d'amis a un ID de groupe
- Un objet doit avoir une estimation de prix
- Un groupe peut avoir un nom

## 5 Schéma conceptuel





## 6 Schéma relationnel



## 7 Choix de conception

### 7.1 Mockup

Pour commencer, nous avons décidé de partir sur un design assez simple et des interactions basiques. En effet, cela nous permettra de ne pas perdre de temps sur Android Studio et d'améliorer au fil du projet et en fonction de nos nouvelles compétences.

Il manque aussi l'interface d'allumage de l'application car nous sommes en discussion du logo (première idée voir page de garde)[1] et du nom de l'application donc c'est une chose qui reste à faire.

### 7.2 Faits élémentaires, ORM et schéma relationnel

Nous avons 5 entités qui sont : Compte, Liste de souhait, Objet, Ami et groupe. Un point important est le lien entre compte et liste de souhait car une liste de souhait peut-être liée à plusieurs comptes comme par exemple à tous les comptes d'un groupe d'amis. C'est là que l'attribut 'option' est importante car c'est ce qui dira si plusieurs comptes peuvent avoir accès à cette liste. Cette contrainte nous avons pas réussi à l'intégrer dans les différents schémas mais nous devons y tenir compte dans la programmation de l'application.

En ce qui concerne le compte et le profil, nous avons pris la décision de ne créer qu'une seule entité pour faciliter les relations et cela nous semble plus efficace.

Nous avons aussi choisi que le nom et le prénom serait obligatoire pour répondre aux attentes du cahier de charges. Pour un objet, nous avons décidé de ne pas utiliser d'ID. SQLite visant des bases de données de taille moyenne, on tente de limiter la redondance entre les différents objets stockés dans la base de donnée, pour en limiter la taille.

Donc pour éviter d'avoir une donnée supplémentaire avec l'ID nous allons utiliser comme clé primaire le nom qui permet d'éviter certains doublons. Nous avons donc une "tentative" d'unicité. L'utilisation d'un ID sur un choix d'objet nous semblait peu productif.

Nous laissons actuellement aucune contrainte sur le type de fichier que doit avoir l'image mais cela pourra évoluer en fonction des possibilités lors de la programmation.

### 7.3 Database

Celle-ci est fourni dans le dossier.

Pour Option, nous avons choisi qu'on utiliserait la colonne '*Public*' du tableau serait **1** si elle est publique et **0** si privée.

### 7.4 Autres points

Dans le cahier des charges, il est demandé de créer deux listes qui sont celle avec les objets reçus et celle avec les objets achetés, nous partons avec l'idée que ces deux listes seront créées par défaut pour chaque compte.

## 8 Conclusion

En conclusion, les différentes étapes déjà réalisées nous permettent d'avoir une idée bien précise du type d'application que nous voulons et comment nous allons y parvenir.

Nous avons une idée du design de l'application même si nous avons décidé de commencer par quelque chose de simple ne sachant pas comment nous allons maîtriser Android Studio. Donc des améliorations sont encore possible au niveau design. Les bases de données sont prêtes et nous avons déjà regarder pour les commandes sql qui seront utiles pour notre application.

Une des principales étapes qui nous reste à faire et qui sera le point culminant de ce projet est la programmation de l'application sur Android Studio. Nous pourons ainsi tester notre application et trouver d'autres améliorations.

## 9 Lexique

- **Entité** :Objet décrit par un type, une valeur et un mode de référence.
- **Mock up** : prototype d'interface utilisateur
- **Faits élémentaires** :Ce sont des assertion simples sur le domaine de l'application
- **Schéma conceptuel** : Un schéma conceptuel est une représentation structurée d'un ensemble d'entités et d'attributs liés entre eux.
- **schéma relationnel** : Manière de modéliser les relations existantes entre plusieurs informations, et de les ordonner entre elles.
- **Diagramme des dépendances** :Permet de définir les liens entre les différents interfaces.
- **Database** :C'est un ensemble de relations n-aires qui permet de stocker les données en lien avec l'application.

## 10 Bibliographie

### Références

- [1] Auteur INCONNU. *Site de création de logo*. URL : <https://fr.freelogodesign.org/>.