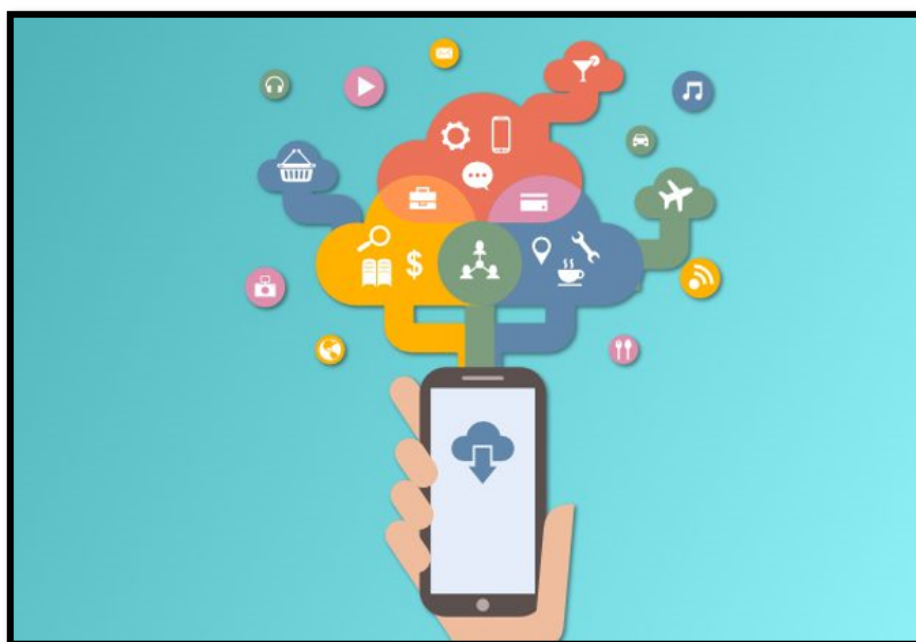


Rapport Final

CRÉATION D'UNE APPLICATION AVEC GESTION DE DONNÉES



Leclercq DELPHINE - 42881800

Dekeyser WILLIAM - 48941700

Fichère HUGO - 68311700

Larue AURÉLIEN - 17611800

Verstraete CHARLOTTE - 39641800

Dermine Loïs - 62901800

Tuteur : Lemaire Valentin

Professeur : Mens Kim

1 Introduction

Pour notre rapport final concernant notre application Wishlist, nous nous sommes axés sur la programmation android de notre application et donc sur la mise en place de tout notre travail préparatoire durant ces douze dernières semaines.

Nous avons fait attention à la cohérence de nos rapports tout au long de ce quadrimestre. Vous pourrez ainsi donc voir, dans ce rapport final, l'évolution de notre travail selon chaque choix de conception.

Nous allons commencer par un lexique de mots spécifiques à notre application pour ensuite vous montrer tous nos schémas relatifs à l'application (ORM, relationnel, diagramme de classe UML diagrammes de séquence UML).

Les choix de conceptions seront précisés et pour finir, nous expliquerons comment utiliser l'application à l'aide d'un mode d'emploi et la façon dont nous avons réparti le travail.

2 Lexique

- **User** : Toute personne identifiée et inscrite sur l'application susceptible de consulter la liste de souhaits d'un autre utilisateur ou de créer sa propre liste de souhaits.
- **WishList** : Liste de produits qu'une personne utilisatrice de l'application ou non, souhaiterait recevoir comme cadeau lors d'un évènement particulier.
- **Product** : Objet ou service proposé par un utilisateur de l'application destiné à une tierce personne ou lui même ou un destinataire non précisé qui le recevra comme cadeau.
- **Login** : Moyen d'identification pour chaque utilisateur.
- **ListNb** : Moyen d'identification pour chaque liste.
- **ProductNb** : Moyen d'identification pour chaque produit.
- **Product - Info (Information)** : Information supplémentaire que l'utilisateur souhaiterait partager avec les personnes qui ont accès au produit en question.
- **Modifier** : Statut de l'utilisateur ayant accès à la liste en tant que soit modificateur, soit lecteur, soit n'y ayant pas accès.
- **Preferences** : L'utilisateur peut partager ses préférences, ses passions... Ces informations sont liées à son profil.
- **Public** : Statut d'une Wishlist, soit tous les amis d'un utilisateur peuvent voir la liste, soit seuls les utilisateurs dont le statut dans la table "Modifier" est lecteur ou administrateur peuvent le voir.

3 Schéma conceptuel ORM (Object-Relationnal Mapping)

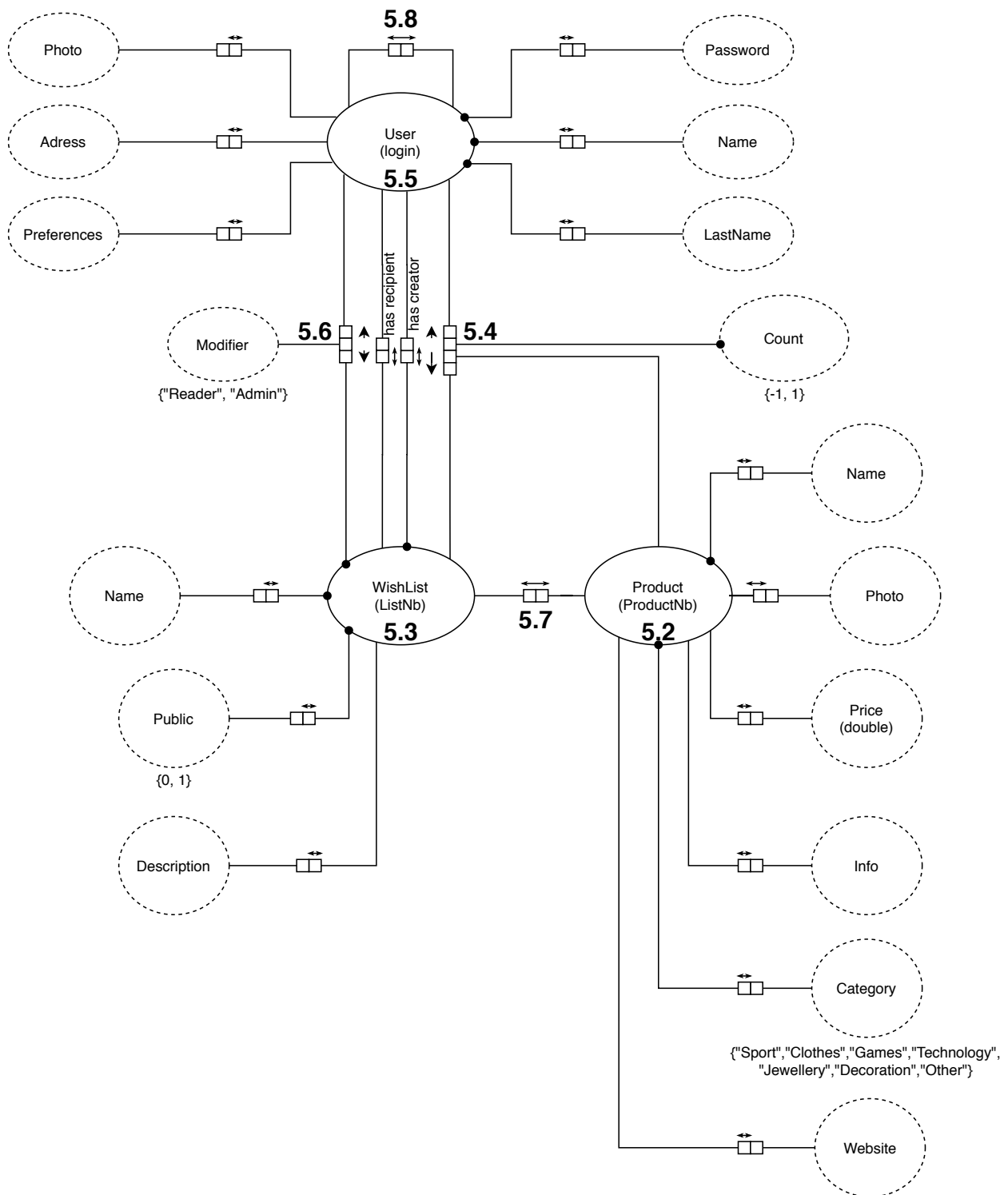


FIGURE 3.1 – ORM

4 Schéma Relationnel

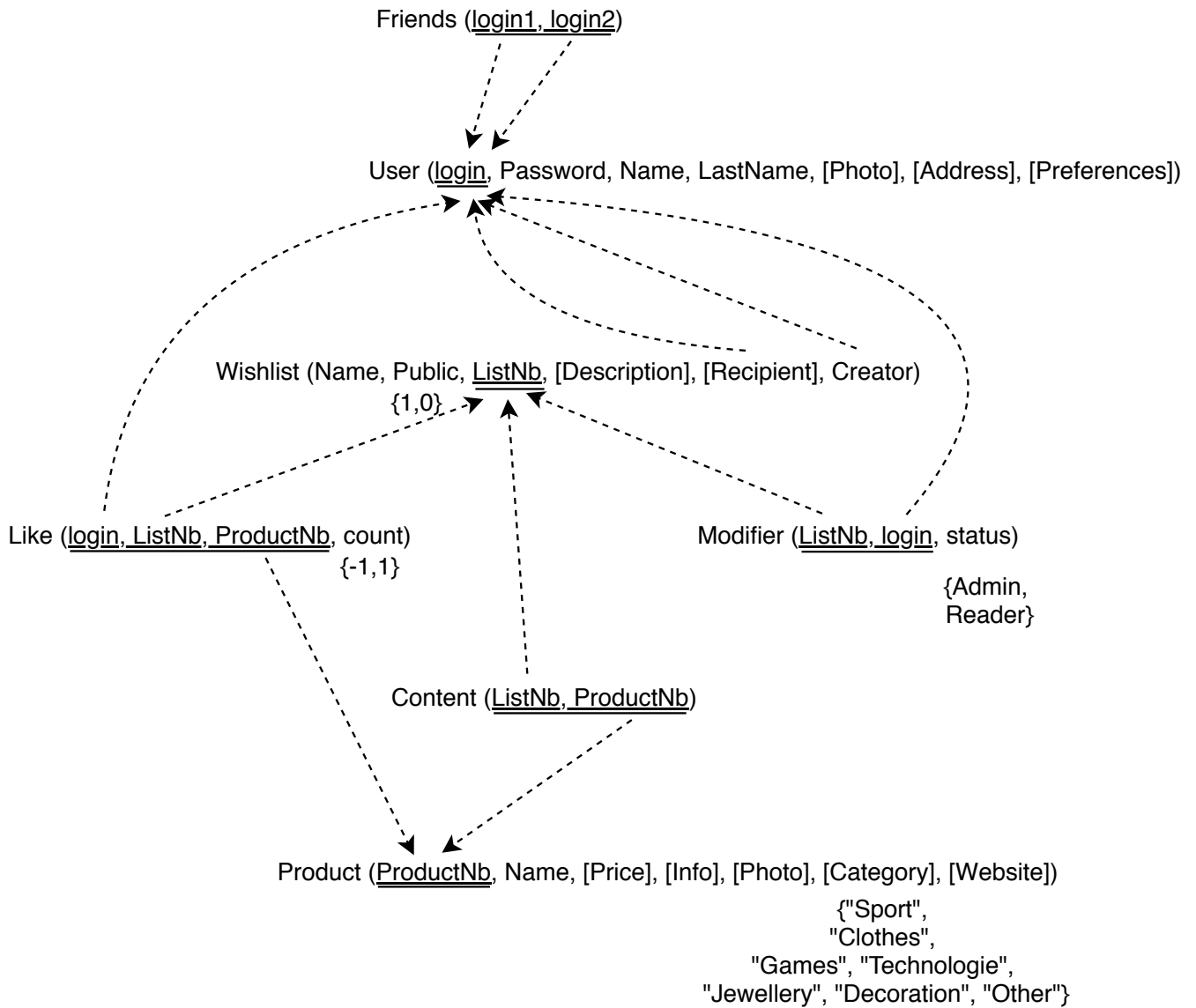


FIGURE 4.1 – Schéma relationnel

5 Choix de conception

Ce chapitre décrit les fonctionnalités obligatoires et optionnelles de l'application. Chaque section contient une brève description des relations et du rôle joué par les données dans l'application. Les tables sont soulignées, les **attributs** sont notés en gras.

5.1 Utilisateurs

User : à la création d'un compte, il est demandé au nouvel utilisateur de trouver un **login** non-attribué. Ce dernier sera demandé à chaque future connexion et utilisé afin d'identifier cet utilisateur de façon unique dans la base de donnée.

De plus, lors de cette inscription, il est nécessaire de fournir un **Password** pour la connexion au profil ainsi qu'un **Name** et un **LastName**, tous trois obligatoires.

Ensuite, il lui est optionnel de renseigner son **Adress**, de fournir une **Photo** pour son profil et finalement d'écrire une **Preference** quant aux cadeaux qu'il désire recevoir.

Friends : un utilisateur peut ajouter d'autres utilisateurs en amis. Dans la base de données, on enregistre les **logins** de la personne qui a fait la demande et de celle qui l'a reçue. Selon la réponse du second utilisateur, la demande d'amis sera acceptée ou non.

Modifier est une table qui correspond à une relation ternaire. Elle permet de créer un lien entre les listes de souhait et les utilisateurs. Pour ce faire, elle fournit, pour chaque utilisateur de chaque liste, un statut qui définit l'accès qu'il a à la liste.

Autrement dit, à chaque paire (**login**, **ListNb** (5.2)) correspond un **Status** "Reader", "Admin" qui sert à définir quels utilisateurs peuvent voir cette liste ou la modifier. De plus, pour les listes publiques (voir **Public** (5.2)), il est aussi possible de rendre cette liste invisible pour certains utilisateurs en les enlevant de la base de données.

5.2 Listes de souhaits

Wishlist : lors de la création d'une liste, l'utilisateur doit obligatoirement fournir un **Name**. Dès la création, la liste sera identifiée par **ListNb** dans la base de donnée. Le login du créateur est automatiquement sauvegardé dans **Creator**. Il est aussi possible d'ajouter une **Description** et un **Recipient** qui seront affichés dans l'application.

Finalement il est obligatoire de décider si la liste est privée ou public. Cela permet de fixer **Public** qui est un booléen, nécessaire pour que l'on puisse décider de l'accès donné à nos amis.

Si la liste est publique ('Public'=1), par défaut tous les amis seront des lecteurs dès le départ. En plus de cela, grâce à cette donnée 'Public', chaque fois que vous vous ferez un nouvel ami, l'application pourra savoir à quelle liste l'ajouter en tant que lecteur. Il sera lecteur pour toutes vos listes publiques. Private ('Public'=0), au contraire, ne donne les accès qu'aux membres que vous ajouterez vous-mêmes. Vos futurs amis, par la suite, devront être ajoutés manuellement si vous le souhaitez. Vous devrez donc préciser, à la création ou plus tard, les amis que vous souhaitez ajouter.

5.3 Produit

Content : Les produits sont liés aux listes grâce à la table détail qui indique une relation binaire (**ListNb,ProductNb**). Pour chaque liste, une ligne est créée par produit qu'elle contient avec donc le numéro de la liste et le numéro du produit.

Likes : La relation likes sert à voter pour les meilleurs produits d'une liste. Dans l'application, il est possible de déposer un like ou un dislike sur chaque produit dans chaque liste, tant que vous avez un status de Reader ou Admin.

Lorsqu'un like est déposé, il génère une relation quaternaire qui lie au triplet (**ListNb, login, ProductNb**) une valeur **count** {1,-1} représentant respectivement un vote pour ou contre le produit. Cela permettra dans le futur de trier les produits par ordre de préférence des utilisateurs.

Product : Les détails des produits sont stockés dans la table Product. L'utilisateur doit obligatoirement fournir un **Name** et une **Category** parmi l'ensemble : {"Sport","Clothes","Games","Technology","Jewellery","Decoration","Other"} et peut rajouter, s'il le souhaite un **Price**, une **Photo**, de l'**Info** supplémentaire, et un lien vers un **Website**. Nous n'avons malheureusement pas encore eu le temps d'implémenter la caractéristique "photo" mais cela arrivera dans les versions ultérieures.

6 Récits utilisateurs

Sur tous les récits utilisateurs cités ci-dessous, vous observerez certains marqués de (+) qui sont les récits que nous avons déjà implémentés dans notre application et d'autres de (-) qui nous restent à implémenter.

6.1 Rôles

Voici les différents rôles possibles dans nos récits utilisateurs :

- Utilisateur : Personne utilisant l'application pour créer des listes de souhaits, ou consulter des listes de souhaits créées par d'autres utilisateurs.
- Créateur d'une liste de souhaits : Utilisateur ayant créé cette liste de souhaits.
- Admin d'une liste de souhaits : Utilisateur ayant le pouvoir de par son statut d'Admin de modifier une liste de souhaits.

6.2 Connexion et inscription

Cette section contient tous les récits utilisateur en rapport avec l'inscription et la connexion au compte de l'utilisateur.

- (+) **En tant** qu'utilisateur, lors du démarrage de l'application, **afin d'utiliser** l'application, **je veux** pouvoir me connecter avec mon login et mon mot de passe.
- (+) **En tant** qu'utilisateur, si je n'ai pas encore de compte, **afin de** me connecter, **je veux** pouvoir m'inscrire en remplissant les champs obligatoires (Name, LastName, Login, Password) ainsi que les champs optionnels (Photo, Address, Preference).
- (+) **En tant** qu'utilisateur, **afin de** me déconnecter, **je veux** pouvoir appuyer sur le bouton déconnexion.

- (+) **En tant** qu'utilisateur, **afin de** ne pas perdre mes données, **je veux** qu'en cas de fermeture non désirée mes données d'inscription soient sauvegardées.

6.3 Profil d'utilisateur

Cette section contient tous les récits utilisateur en rapport avec la lecture et modification des informations du profil de l'utilisateur.

- (+) **En tant qu'**utilisateur, **afin de** pouvoir voir mes données personnelles, **je veux** pouvoir consulter mon profil.
- (+) **En tant qu'**utilisateur, **afin de** changer mon profil, **je veux** pouvoir modifier les informations optionnelles de mon profil (Address, Preference, Photo).

6.4 Amis

Cette section contient tous les récits d'utilisateur en rapport avec les amis et requêtes d'amitié d'un certain utilisateur.

- (+) **En tant qu'**utilisateur, **afin de** pouvoir accepter ou refuser un ami, **je veux** avoir accès à la liste d'utilisateurs m'ayant envoyé une requête d'amitié.
- (+) **En tant qu'**utilisateur, **afin d'**afficher mes amis, **je veux** avoir accès à ma la liste d'amis à n'importe quel moment via un menu dédié à celle-ci.
- (+) **En tant qu'**utilisateur, **afin d'**ajouter un ami sur l'application, **je veux** pouvoir écrire son prénom dans la barre de recherche et lui envoyer une invitation.
- (-) **En tant qu'**utilisateur, **afin de** supprimer un ami, **je veux** pouvoir modifier ma liste d'amis.

6.5 Liste de souhaits

Cette section contient tous les récits d'utilisateur en rapport avec la création et modification de liste de souhaits, ainsi que des utilisateurs disposant du droit de modifier ou de lire ces listes de souhaits.

- (+) **En tant qu'**utilisateur, **afin de** pouvoir acheter un cadeau en fonction des goûts de mon ami, **je veux** avoir accès à ses listes de souhaits.
- (+) **En tant qu'**utilisateur, **afin de** consulter mes listes, **je veux** afficher l'ensemble des listes de souhaits auxquels j'ai accès (c'est-à-dire auxquels mon statut est 'Reader' ou 'Admin').
- (+) **En tant qu'**utilisateur et administrateur d'une liste, **afin d'**afficher et de modifier certains paramètres (Name, Public, Recipient), **je veux** avoir accès à ces informations via une interface dédiée.
- (+) **En tant qu'**utilisateur et administrateur d'une liste de souhaits, **afin de** compléter ma liste, **je veux** pouvoir ajouter ou supprimer des produits de cette liste.
- (+) **En tant qu'**utilisateur et créateur d'une liste, **afin de** contrôler quels utilisateurs ont le pouvoir de lire, de modifier ou de ne pas voir la liste, **je veux** pouvoir modifier le statut de mes amis pour cette liste de souhaits à respectivement 'Reader', 'Admin' et en cas de "non-status" il est 'Invisible'.

- (+) **En tant qu'utilisateur, afin de** modifier l'accès à mes listes de souhaits, **je veux** pouvoir changer le statut de ma liste (Public ou non).
- (-) **En tant qu'utilisateur, afin d'**avoir plus de clarté dans la visualisation d'une liste de souhaits, **je veux** pouvoir trier ma liste en plusieurs catégories.

6.6 Produits

Cette section contient tous les récits d'utilisateur en rapport avec la modification et l'appréciation des produits.

- (-) **En tant qu'utilisateur, afin de** m'informer sur un produit (Price, Website, Photo, Info, Category), **je veux** avoir accès à une interface dédiée à celui-ci.
- (-) **En tant qu'utilisateur, afin de** montrer mon intérêt pour un certain produit, **je veux** pouvoir ajouter un like ou un dislike à un certain produit dans une certaine liste de souhaits.

6.7 Général

(+) **En tant qu'utilisateur, afin d'**avoir une expérience optimale, **je veux** que l'ouverture d'écran et les délais de rafraîchissement de l'application se fassent avec un délai de l'ordre de la seconde.

7 Diagrammes de classe UML

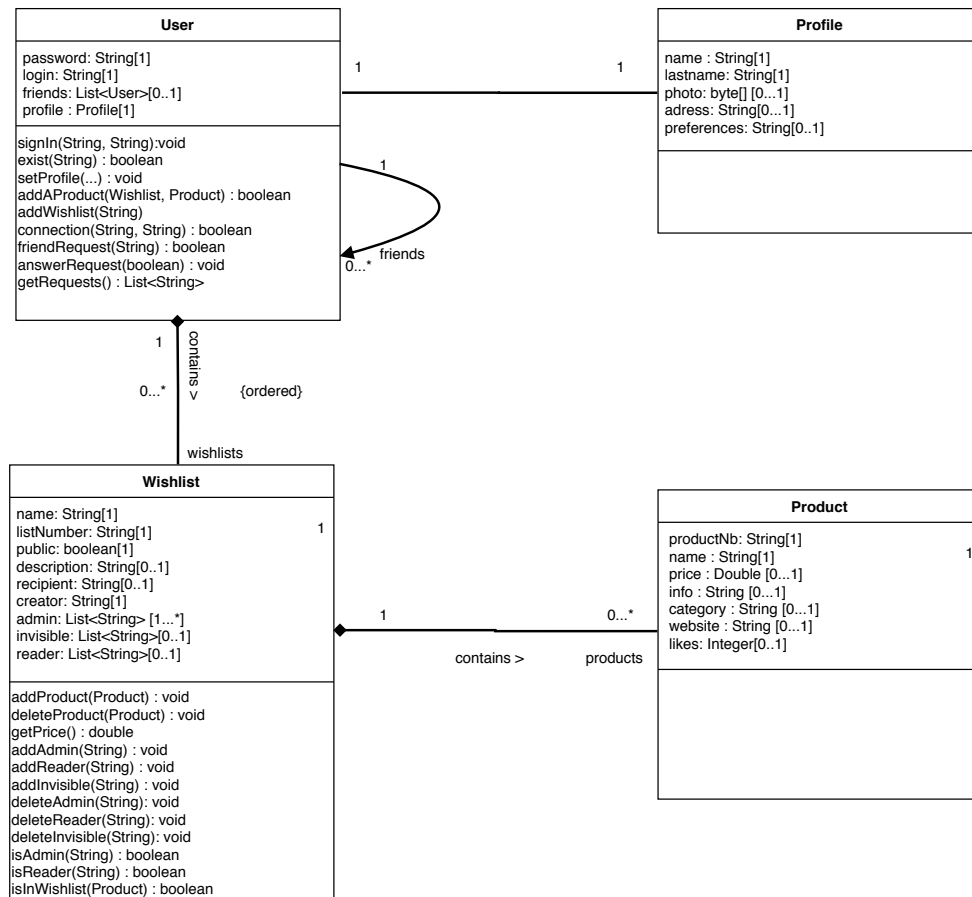


FIGURE 7.1 – Diagramme de classe de l'application

7.1 Diagrammes de classes UML

Comme pour les autres classes ci-dessus, on s'est rendu compte que l'objet `User` devait être implémenté tel une classe dans le code Java. Effectivement, on remarque qu'il y a beaucoup d'interactions entre les objets.

Dans notre application, une de nos fonctionnalités importantes est de créer des amitiés entre les utilisateurs. Nous avons choisi de représenter l'amitié avec l'attribut `"friends"` qui est une liste contenant les objets `User` de toutes les personnes qui sont amis de l'instance.

7.1.1 Profil

La classe de profil a été créée pour avoir accès aux informations personnelles de l'utilisateur, chaque utilisateur doit avoir un profil.

7.1.2 Wishlist

Nous avons décidé d'implémenter une classe Wishlist car il s'agit tout d'abord du concept principal de l'application autour duquel vont s'articuler les différents objets. La deuxième raison de ce choix est de faciliter la création des diagrammes de séquences.

7.1.3 Produit

Nous avons décidé d'implémenter une classe Product parce que chaque produit possède plusieurs attributs (ProductNb, Name, Website, etc) spécifiques qui le différencient des autres produits.

7.1.4 Changements apportés

Dans la section "User", nous avons décidé que l'attribut friends serait représenté par une liste de "User" afin de directement avoir accès aux logins en cas de nécessité.

8 Diagrammes de séquence UML

8.1 Diagrammes de séquences UML

8.1.1 Inscription

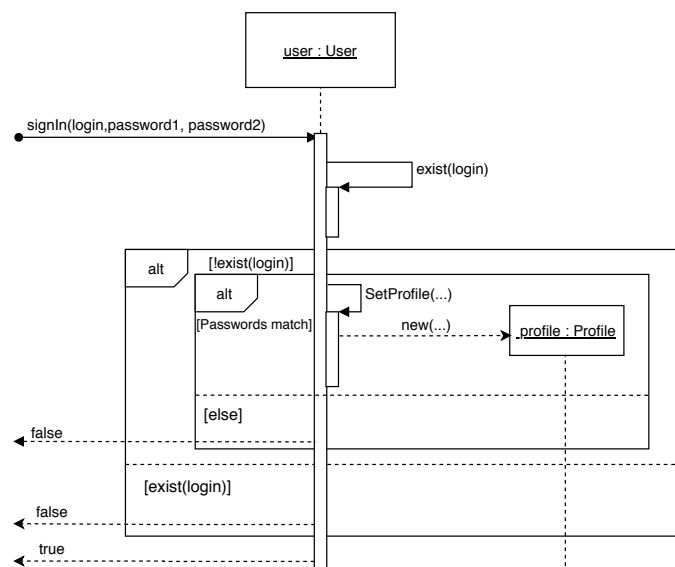


FIGURE 8.1 – Diagramme représentant la création d'un compte utilisateur

Le premier diagramme représente l'inscription d'un utilisateur sur l'application. Celui-ci survient lorsqu'une personne ouvre l'application et s'enregistre en tant que nouvel utilisateur. Il doit donc fournir un identifiant, un mot de passe et confirmer ce mot de passe. Si l'identifiant est libre et que les mots de passe correspondent, il sera amené sur la page de création de profil. Un profil sera alors créé avec ses informations et sera lié à l'utilisateur. Sinon, la procédure échoue et l'utilisateur devra réessayer.

8.1.2 Ajout d'un produit

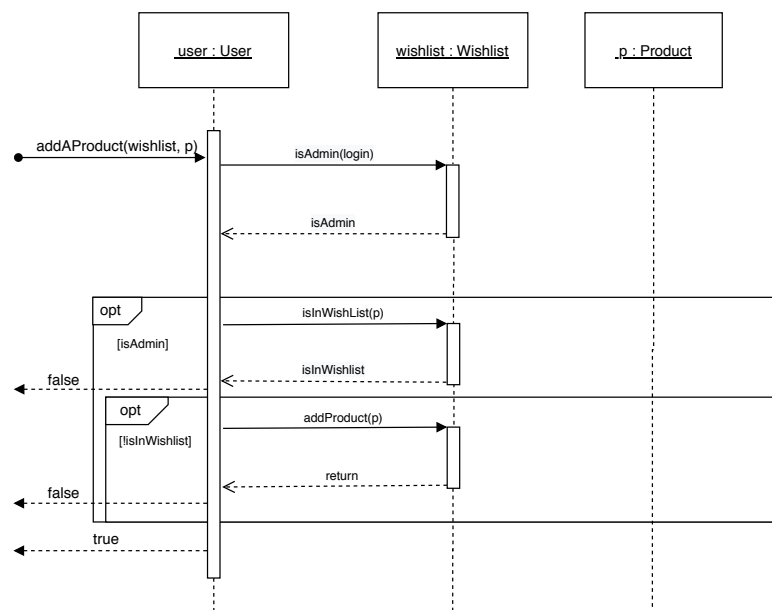


FIGURE 8.2 – Diagramme représentant l'ajout d'un produit dans une wishlist

Le deuxième diagramme a lieu lorsqu'un utilisateur souhaite ajouter un produit existant à l'une de ses wishlists en cliquant sur un bouton prévu à cet effet. Pour ce faire, il faut vérifier si l'utilisateur est un administrateur de la liste. Si c'est le cas, nous vérifions si le produit pris en argument est déjà présent dans celle-ci. S'il n'y est pas, nous appelons la fonction "addProduct(produit)" pour rajouter le produit. S'il n'est pas administrateur ou que le produit était déjà dans la liste, un message d'erreur apparaîtra.

8.1.3 Demande et ajout d'ami

L'ajout d'un ami se fait en deux étapes.

Dans la première, l'utilisateur envoie une demande. Il entre donc un login qui doit exister. S'il n'existe pas, l'utilisateur doit réécrire un login ou quitter la page. Si ce dernier existe, on vérifie qu'il n'essaie pas d'ajouter un utilisateur avec lequel il est déjà ami ou soi-même. Si ce n'est pas le cas, on enregistre la requête dans la base de donnée par "addFriend(login2)".

Dans la seconde étape, lorsque l'utilisateur 2 se connectera et accédera à sa page d'amis, il verra les requêtes qui lui sont destinées et pourra choisir sa réponse pour chacune d'entre elles. S'il décide d'accepter, on enregistre "addFriend(login1)" dans la base de donnée et ils sont donc amis.

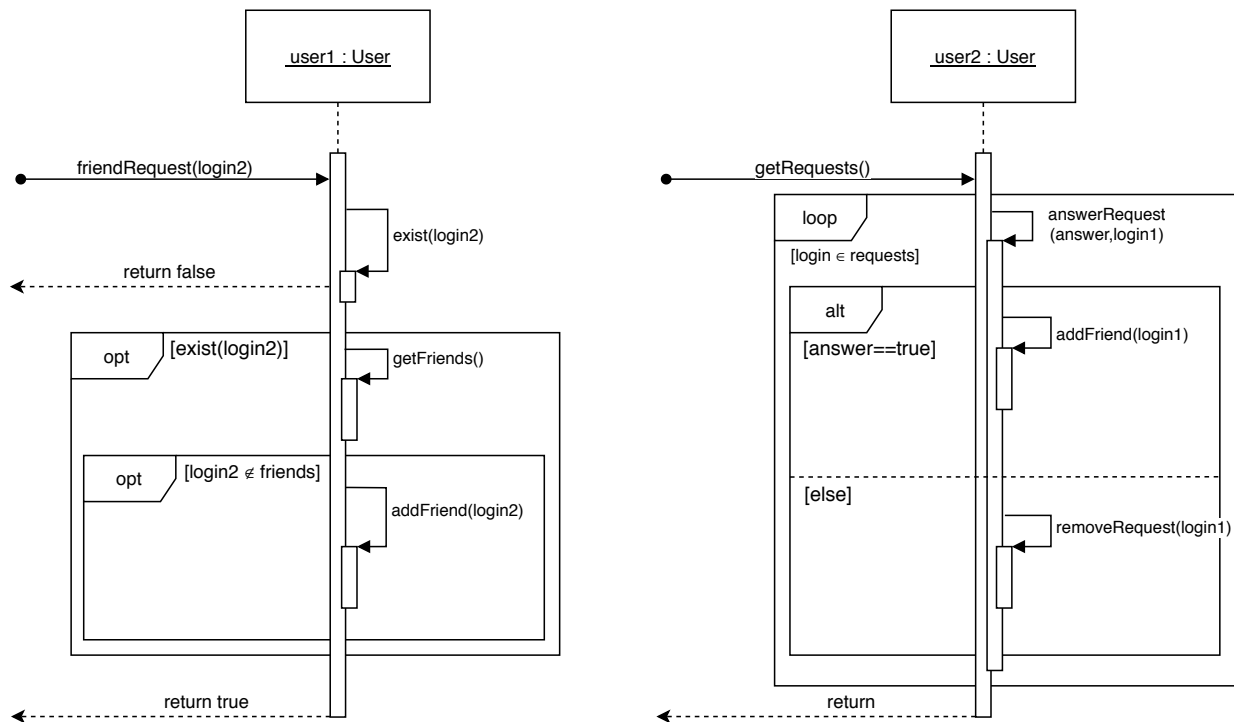


FIGURE 8.3 – Diagrammes représentant l’envoi d’une demande d’ami, et la réponse à celle-ci ¹

Changements apportés

Nous avons décidé de faire renvoyer un booléen à la fonction "friendRequest" car de cette façon, nous avons un retour sur la réponse du second utilisateur.

9 Choix de conceptions et d’implémentations

9.1 Nos points forts

Lorsque nous avons réfléchi à "comment rendre l’application attrayante", nous avons eu deux idées qui sont sorties du lot. La première était de fort nous axer sur le principe de réseau d’ami. En effet, hormis le fait que chaque utilisateur puisse ajouter d’autres utilisateurs en amis, nous avons aussi ajouté la possibilité de pouvoir interagir. De ce fait, un utilisateur peut donner l’autorisation à un ami de pouvoir modifier ses wishlists un maximum. Il peut décider aussi de quels amis pourront voir ses propres wishlists ou non. De plus, il est possible de consulter le profil de ses amis.

Notre deuxième idée a été de rajouter la possibilité de mettre des likes à chaque produits dans une liste. Chaque utilisateurs consultant cette liste peut décider ou non d’y ajouter un like. Un produit pourrait donc avoir 5 likes, tandis qu’un autre pourrait en avoir -3.

10 Manuel d’emploi de l’application

En arrivant sur l’écran du téléphone , il suffit d’appuyer sur le bouton de l’application pour commencer à utiliser DreamList.

10.1 Inscription et connexion

Une fois que l'application est lancée vous serez envoyé vers l'écran de connexion. Si vous n'étiez pas encore inscrit dans l'application vous serez invité à créer un compte en cliquant sur le bouton '*Create an account*'. Vous devez dans un premier temps choisir un login . Deux situations peuvent se présenter : 1) Si celui-ci n'est pas encore utilisé par une tierce personne, vous êtes invité à entrer votre mot de passe, et à le confirmer. Une fois votre compte crée, vous serez automatiquement redirigé vers la page de création de profil, ce qui vous permettra d'entrer toutes les données vous caractérisant (nom, prénom, préférences, adresse).

2) Si par contre le login entré est déjà utilisé ou que votre mot de passe ne correspond pas à celui que vous confirmez, un message vous indiquant de devoir réitérer l'opération s'affichera.

Si vous avez déjà créé un compte sur l'application, vous devez vous connecter et vous serez redirigé vers le menu principal.

10.2 Menu principal

Le menu principal donne accès aux fonctionnalités suivantes :

- Wishlist
- Friends
- Profile

En cliquant sur chaque fonctionnalité, un nouvel écran spécifique apparaît :

10.3 Profile

L'écran 'profil' permet de voir ses informations personnelles (Nom, Prénom, préférences et adresse) Si vous souhaitez changer votre profil, vous êtes invité à cliquer sur le bouton en haut à droite pour modifier vos données personnelles dans les espaces prévus à cet effet.

10.4 Friends

Notre application permet d'ajouter des amis. En effet, depuis le menu principal vous pouvez accéder à l'écran Friends qui permet de voir votre liste d'amis ou d'ajouter un nouvel ami. Si vous souhaitez ajouter un ami, vous devez appuyer sur le bouton '+' ; un nouvel écran apparaîtra où vous pourrez écrire son login (si il existe) dans l'espace prévu à cet effet, ensuite il suffit de confirmer l'ajout sur le bouton '*Confirm*' et la demande sera envoyée automatiquement au nouvel ami.

10.5 Wishlist

Quand vous cliquez sur '*Wishlist*', vous aurez la possibilité de rechercher parmi toutes les wishlists auxquelles vous avez accès, et de naviguer parmi celles-ci. Vous pouvez aussi créer une nouvelle liste, en allant sur le bouton '+'. Une fois que vous avez appuyé sur ce bouton, un nouvel écran apparaît et il est alors possible de créer la nouvelle wishlist sur un espace prévu à cet effet. Une fois l'opération terminée, il suffit de confirmer la création de la liste.

10.6 Produits

Quand vous cliquez sur une de vos wishlist, vous aurez la possibilité de voir tous les produits qui font parties de celle-ci. Vous pouvez aussi créer un nouveau produit, en allant sur le bouton '+'. Une fois que vous avez appuyé sur ce bouton, un nouvel écran apparaît et il est alors possible de créer le nouveau produit. Une fois l'opération terminée, il suffit de confirmer la création du produit.

11 Répartition et organisation du travail

Lors des travaux pratiques, nous avons travaillé ensemble à chaque fois sans se répartir le travail mais bien en avançant tous ensemble. Pour la programmation de l'application, nous avons beaucoup travaillé en groupe pour initier le projet. Une personne partageait son écran lors de nos meetings sur Microsoft Teams et tout le monde proposait ses idées. De cette façon, nous avons six personnes pour se rappeler de tous les liens et choix de conception que nous avons choisi lors du début du quadrimestre. Néanmoins, pour avancer plus vite, nous nous sommes séparés les tâches par groupes de deux. Nous nous expliquions le tout dans une réunion avec le groupe au complet. Certes nous avons sûrement perdu plus de temps que si nous nous étions vus en physique, mais nous avons malgré tout fait preuve d'une grande capacité à communiquer entre nous via tous les supports informatiques mis à notre disposition (Teams, Messenger, Overleaf, Draw.io,...).

Lors de la rédaction des rapports, cela s'est avéré relativement facile car nous corrigions nos schémas à chaque fois que nos nouvelles idées le demandaient. Cela nous a permis de simplement placer les schémas dans le rapport en expliquant nos choix de conception. Pour la finalisation du rapport, il ne restait plus grand chose, mise à part poser nos dernières questions à Valentin, notre tuteur.

Au niveau de la répartition des différents éléments du travail, nous pouvons dire :

- Charlotte s'est axée sur login, le profil, les paramètres d'une wishlist et les récits utilisateurs.
- Loïs s'est concentré sur la liste de wishlist, la liste d'ami et le diagramme de séquence User et Product.
- Aurélien s'est chargé de la classe wishlist et du diagramme de séquence User et Product.
- Hugo s'est occupé du profil, du diagramme de séquence Friends et la rédaction du rapport
- William s'est axé sur la classe wishlist, le lexique, les récits d'utilisateurs et la rédaction du rapport
- Delphine s'est concentrée sur la page wishlist, le diagramme de séquence Friends et la rédaction du rapport.

Pour toutes les autres tâches, nous considérons que nous les avons faites ensemble.

Finalement, les seules difficultés qu'on nous avons rencontrées étaient la période d'adaptation sur Teams et de l'outil Git. Nous avons du mal à ne pas empiéter sur le travail des autres lors du début de la rédaction du code. Trouver une plage horaire qui allait à tous pour se voir et travailler le code de l'application n'était pas aussi facile que prévu. Mais comme vous le verrez, nous nous en sortons bien et nous avons su trouver une harmonie !