

ÉCOLE POLYTECHNIQUE DE LOUVAIN

LSINF1225 2017-2018

GROUPE 12 : RAPPORT FINAL

Conception orientée objet

Auteurs :

Gilles PEIFFER : 2432 1600

Hadrien PLANCQ : 5666 1600

Augustin DELECLUSE : 1923 1600

Baptiste DEMUYSÈRE : 2395 1600

Guillian STEPPE : 2400 1600

Personnes ressources :

Benoît DUHOUX



11 mai 2018

Table des matières

Introduction	2
1 Diagramme de classes	3
2 Diagrammes de séquence	4
2.1 Connexion	4
2.2 Inscription	4
2.3 Création de profil	5
2.4 Amis	6
2.5 Créer un sondage	7
2.6 Répondre à un sondage	8
2.7 Classement des utilisateurs pour un questionnaire	8
3 Récits utilisateurs	9
4 Schéma ORM	12
5 Schéma relationnel	14
6 Manuel d'emploi	15
6.1 LoginActivity	15
6.2 Créer compte	15
6.3 Menu	16
6.4 Menu de création	16
6.5 Menu d'amis	17
6.6 Ajouter un amis	17
6.7 Création questionnaire	18
6.8 Ajout de choix aux questions	18
7 Choix de conception	19
8 Participation respective et organisation du groupe	20
Conclusion	21

Introduction

Dans le cadre du cours *LSINF1225 - Conception orientée objet et gestion de données*, nous avons dû concevoir une application permettant de créer et répondre à divers sondages. Nous l'avons baptisée *MiniPoll*. Mais créer une application passe par différentes étapes nécessaires : des diagrammes de classes aux récits utilisateurs en passant encore par la construction d'une base de données grâce à des faits élémentaires bien réfléchis.

L'élaboration d'un projet ne se fait pas en une seule fois. Il s'agit plutôt de plusieurs cycles, qui nous font avancer peu à peu vers notre objectif. Ainsi, nos différents diagrammes ont été retouchés et nos classes ont changées maintes et maintes fois avant d'avoir leur rendu actuel. Quelques différences sont donc à noter vis-à-vis de nos anciens rapports.

Vous pouvez retrouver ces différents changements ici, en commençant par nos différents diagrammes (de classes et de séquence), des récits utilisateurs associés ainsi que de nos schémas (ORM et relationnel). Un manuel d'emploi est également fourni pour faciliter. Enfin, vous trouverez nos choix de conception et quelques considérations sur notre répartition des tâches.

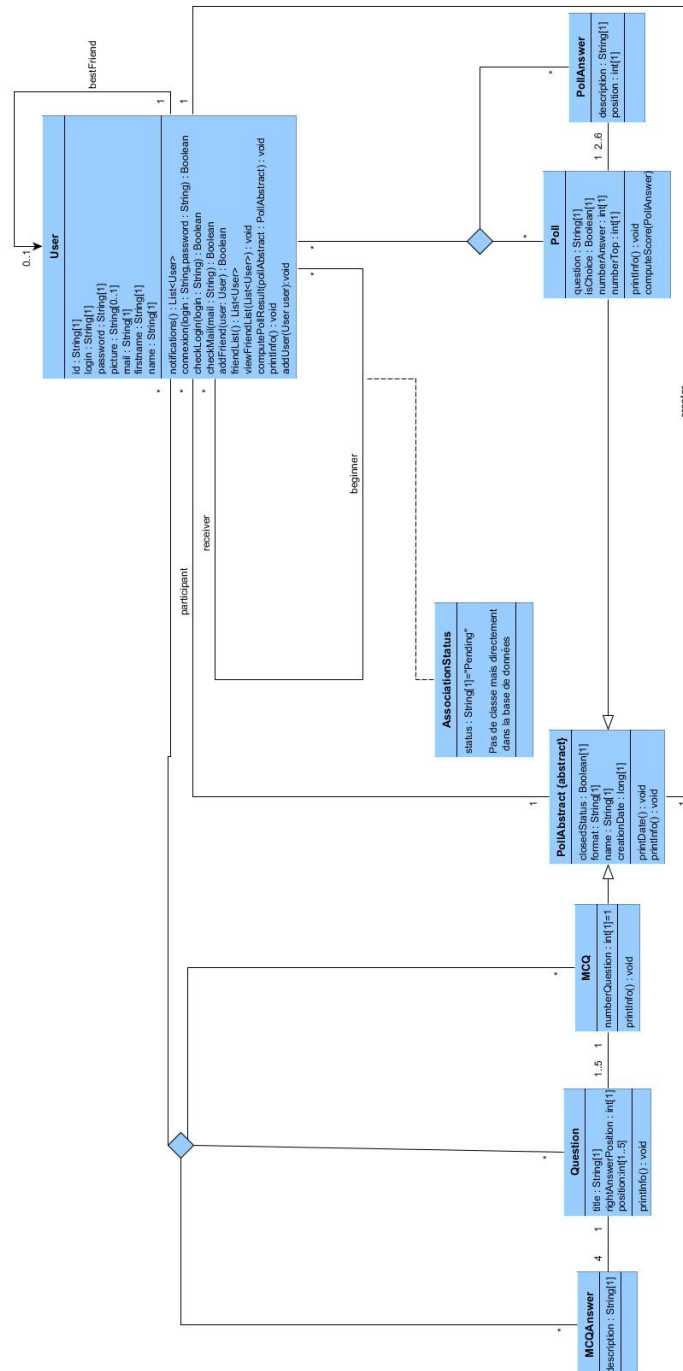
Section 1

Diagramme de classes

Certaines méthodes ont été rajoutées, auxquelles nous n'avions pas pensées lors de l'élaboration de notre diagramme de classes :

- computeScore();
- Register();

Les multiplicités des contraintes ont été modifiées en conséquence et les classes d'association supprimées car elles étaient inutiles.

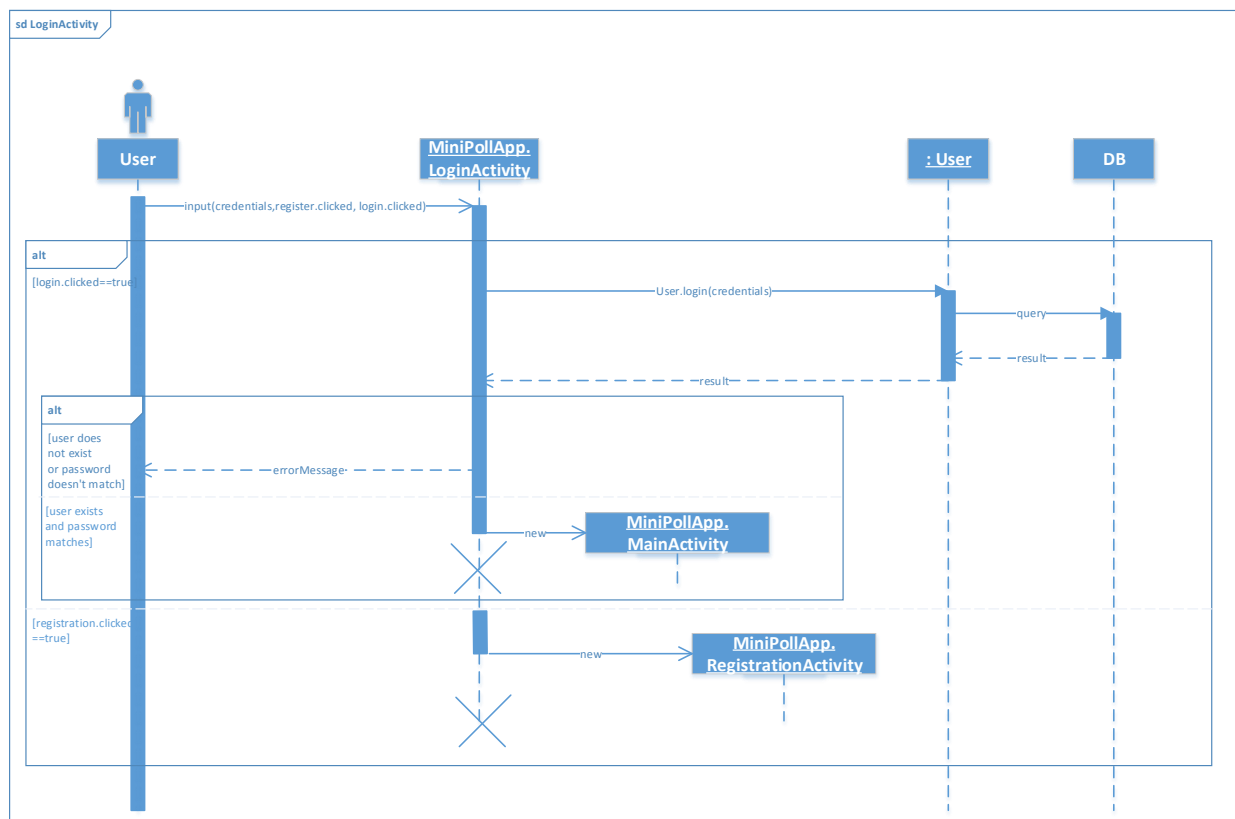


Section 2

Diagrammes de séquence

2.1 Connexion

Il s'agit ici d'un nouveau diagramme de séquence, qui reprend en fait une partie du diagramme d'inscription des missions précédentes.

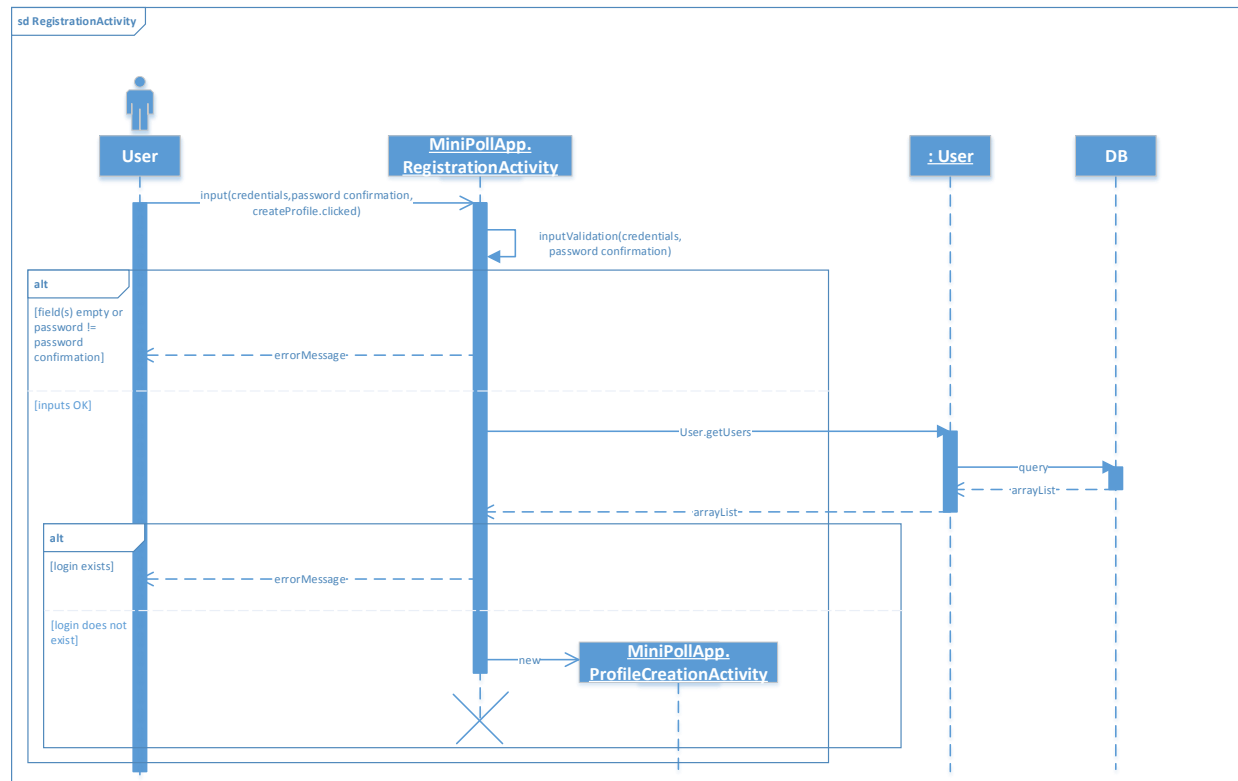


2.2 Inscription

Le diagramme de séquence concernant l'inscription dans les rapports précédents a été divisé en trois diagrammes plus petits car lors de l'implémentation nous nous sommes aperçus que l'ancien diagramme était trop compact pour indiquer toutes les différentes interactions dans ce processus.

Les principaux changements dans ce diagramme de séquence-ci sont :

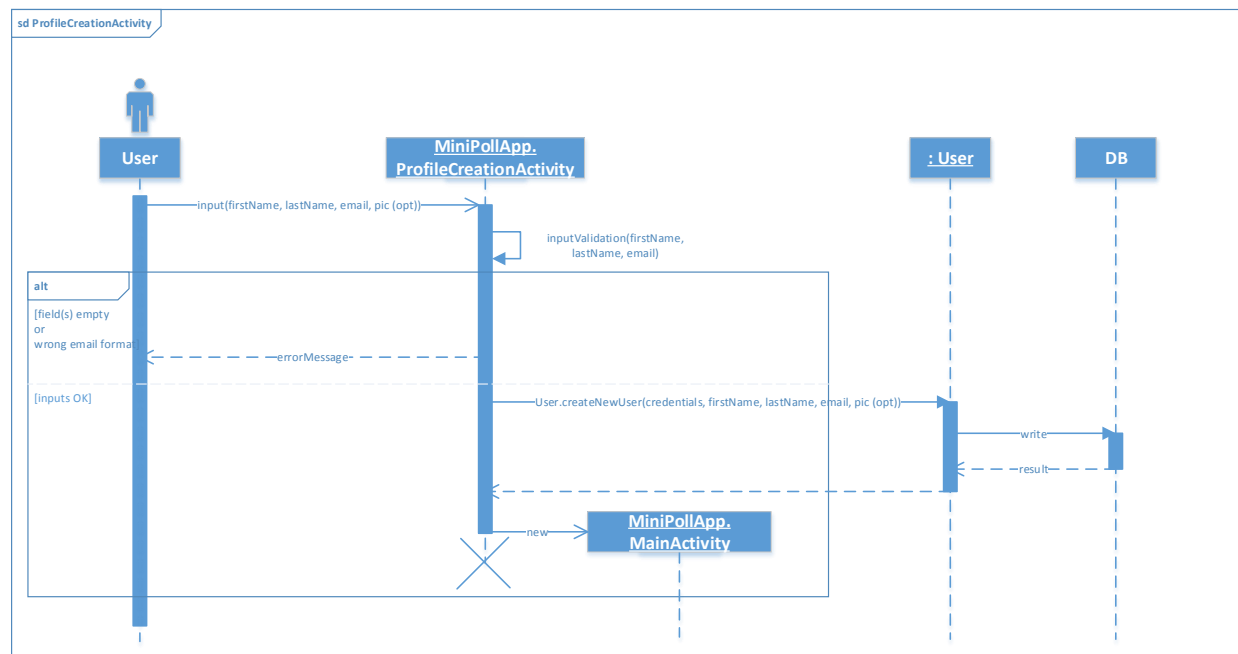
- l'application ne fait pas directement appel à la base de données, mais passe par une des méthodes de la classe **User** qui retourne tous les utilisateurs enregistrés dans la base de données ;
- Après la création des champs **login** et **password**, l'utilisateur n'est pas immédiatement inscrit dans la base de données, mais passe à une activité suivante, la création de profil (section 2.3).



2.3 Création de profil

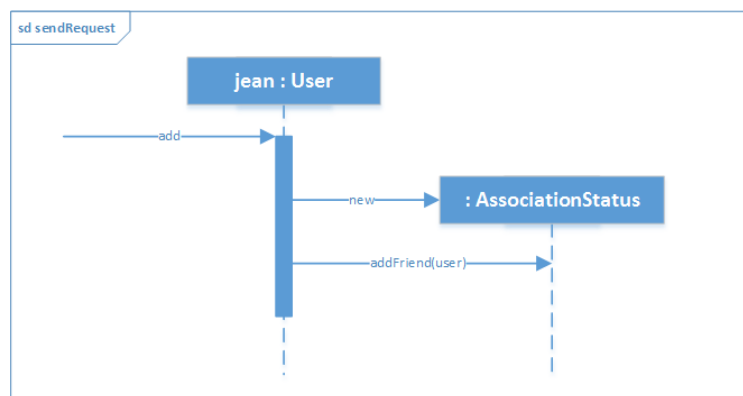
Il s'agit ici d'un nouveau diagramme de séquence. Cependant, on remarque qu'une action qui auparavant ne prenait qu'un processus à se faire (l'insertion dans la base de données d'un nouvel utilisateur) en prend ici trois. Cela est dû à la façon dont on implémente le processus de connexion. Il s'agit donc d'un choix de conception. Les autres choix de conception sont détaillés à la section 7. Ensemble avec la connexion (section 2.1) et l'inscription (section 2.2), ces diagrammes décrivent le même processus (avec plus de détails) que l'ancien diagramme de séquence de l'inscription.

Un autre choix de conception est le fait de séparer la création de profil de la modification (et consultation) de profil.

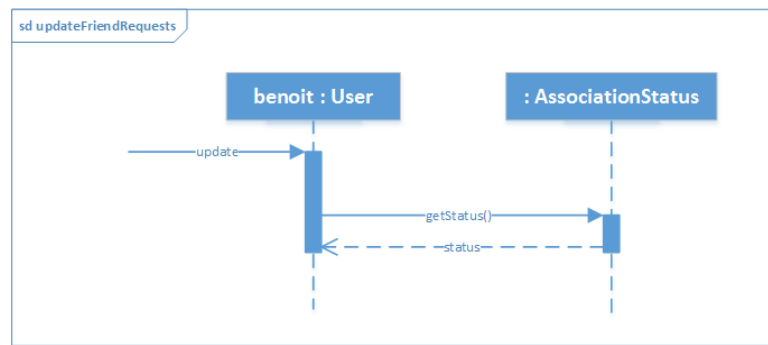


2.4 Amis

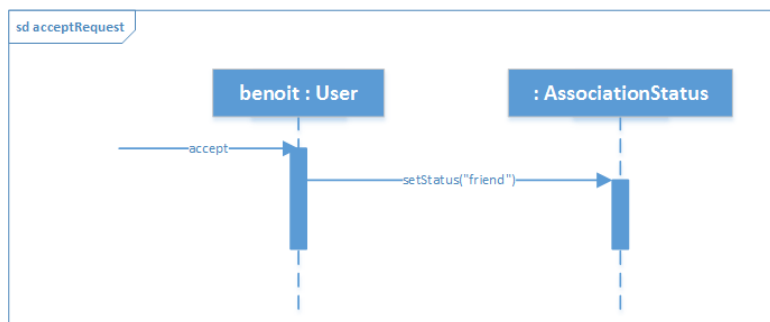
Envoyer une demande d'ami



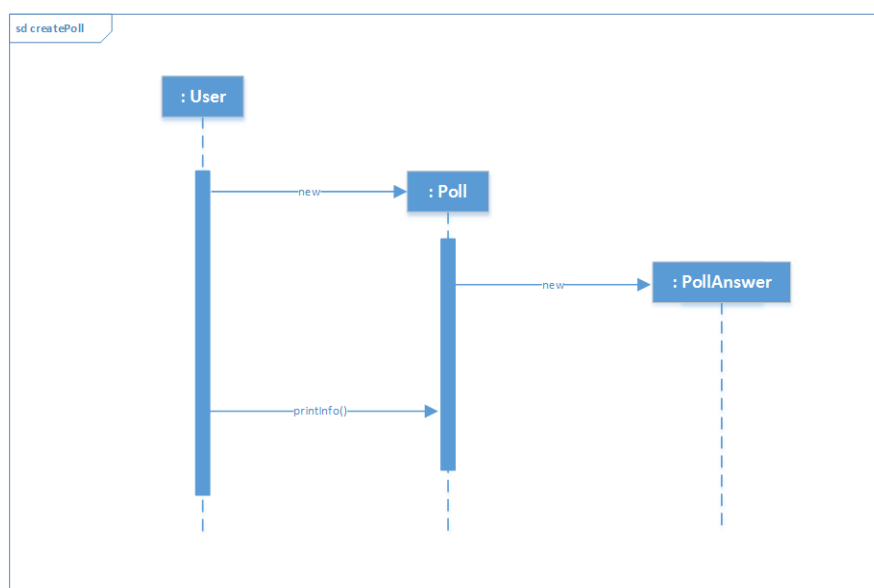
Actualiser ses demandes d'amis



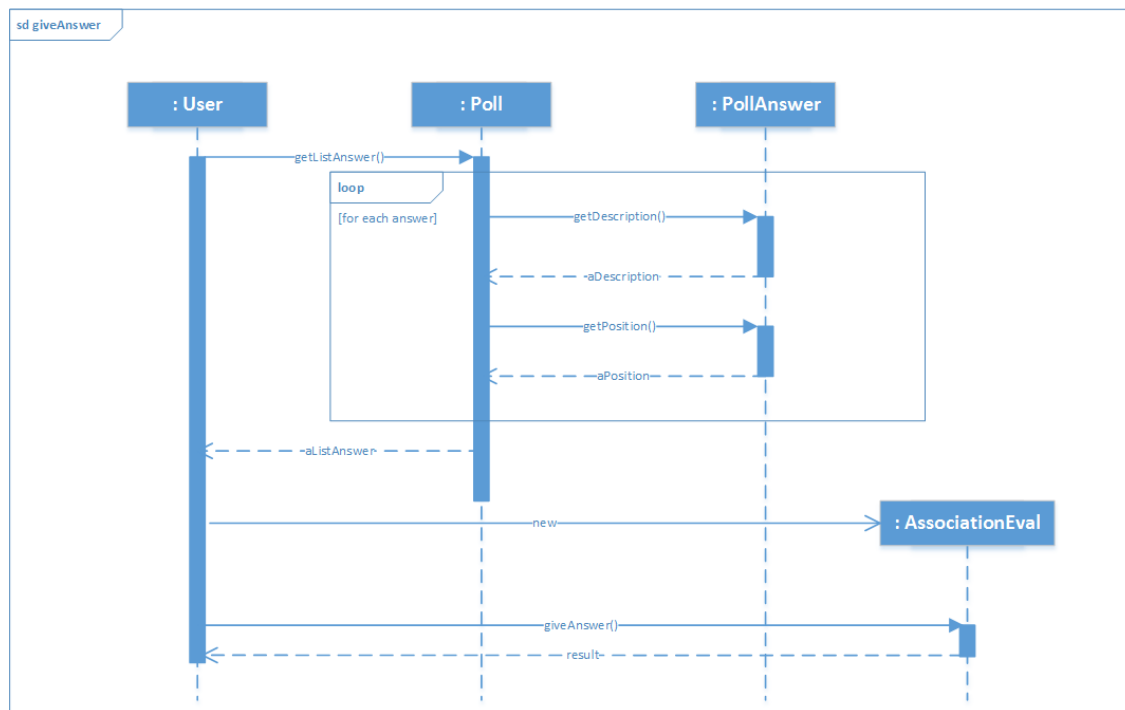
Accepter une demande d'ami



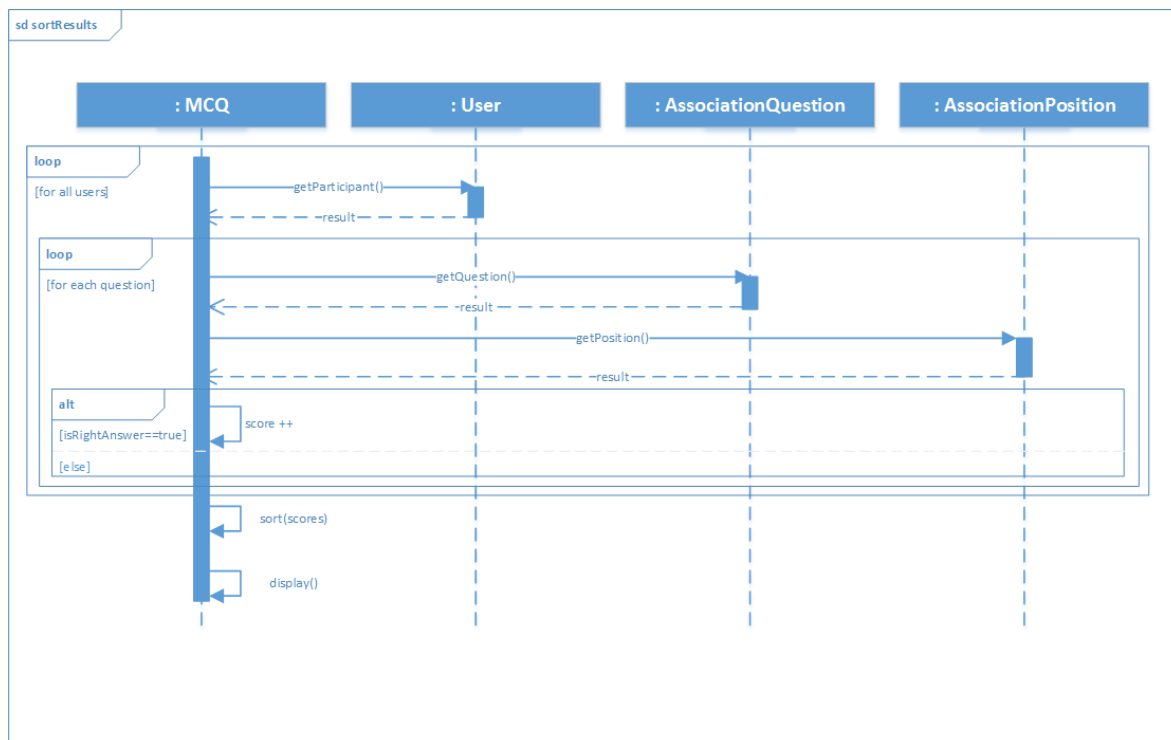
2.5 Créer un sondage



2.6 Répondre à un sondage



2.7 Classement des utilisateurs pour un questionnaire



Section 3

Récits utilisateurs

Dans les récits utilisateurs suivants, "sondage" fait référence aux 3 types de sondages tandis que "questionnaire" ne fait référence qu'aux questionnaires.

Chaque récit utilisateur est pondéré en fonction du temps qu'il prendra à implémenter.

Inscription/modification des informations

- Poids : 5 - En tant que nouvel utilisateur, je veux choisir un(e) < login, mot de passe, photo de profil, adresse mail, prénom, nom > afin de m'inscrire sur l'application.
- Poids : 1 - En tant qu'utilisateur, je veux recevoir un avertissement en cas d'erreur dans les données rentrées lors de ma création/modification de profil afin de réessayer de créer/modifier mon profil.
- Poids : 1 - En tant qu'utilisateur existant, je veux pouvoir modifier mon/ma < login, mot de passe, photo de profil, adresse mail, prénom, nom > afin de mettre à jour mes informations personnelles.
Modification : Poids 3 - Il a fallu une méthode pour modifier dans la base de données.

Connexion

- Poids : 2 - En tant qu'utilisateur, je veux pouvoir rentrer mon login et mon mot de passe afin de me connecter.
- Poids : 1 - En tant qu'utilisateur, je veux recevoir un avertissement en cas d'erreur dans les données rentrées lors de la connexion afin de réessayer de me connecter.

Amis

- Poids : 5 - En tant qu'utilisateur, je veux pouvoir accéder à ma liste d'amis afin de voir les utilisateurs que j'ai ajoutés.
- Poids : 3 - En tant qu'utilisateur, je veux pouvoir accéder à mon centre de notifications afin de voir les invitations et les demandes que j'ai reçues.
- Poids : 2 - En tant qu'utilisateur, je veux pouvoir ajouter un autre utilisateur dans ma liste d'ami afin de pouvoir l'inviter dans un sondage.
- Poids : 2 - En tant qu'utilisateur, je veux pouvoir accepter les demandes en ami qui m'ont été envoyées afin de pouvoir être invité à un sondage par un ami.
- Poids : 2 - En tant qu'utilisateur, je veux pouvoir *swiper* afin de parcourir ma liste d'amis.
Modification : Poids 4 - Nous avons bien sous-coté le temps pour faire le *swipe*.
- Poids : 2 - En tant qu'utilisateur parcourant ma liste d'amis, je veux pouvoir appuyer sur une icône en forme de croix afin de supprimer un utilisateur avec lequel je suis ami de ma liste d'amis.
- Poids : 1 - En tant qu'utilisateur, je veux pouvoir observer la présence d'une couronne afin de reconnaître la personne que j'ai ajoutée en tant que meilleur ami.
- Poids : 1/2 - En tant qu'utilisateur, je veux pouvoir choisir un autre utilisateur afin de lui donner le statut de meilleur ami.

Créateur de sondage

- Poids : 8 - En tant qu'utilisateur, je veux pouvoir créer un sondage afin de connaître l'avis de mes amis/tester leurs connaissances.¹
- Poids : 5 - En tant que créateur de sondage, je veux pouvoir voir les résultats non anonymisés du sondage afin de voir les réponses de chacun des participants au sondage ayant répondu.

1. Dans un questionnaire on ne demande pas vraiment l'avis des participants, on pose plutôt une question.

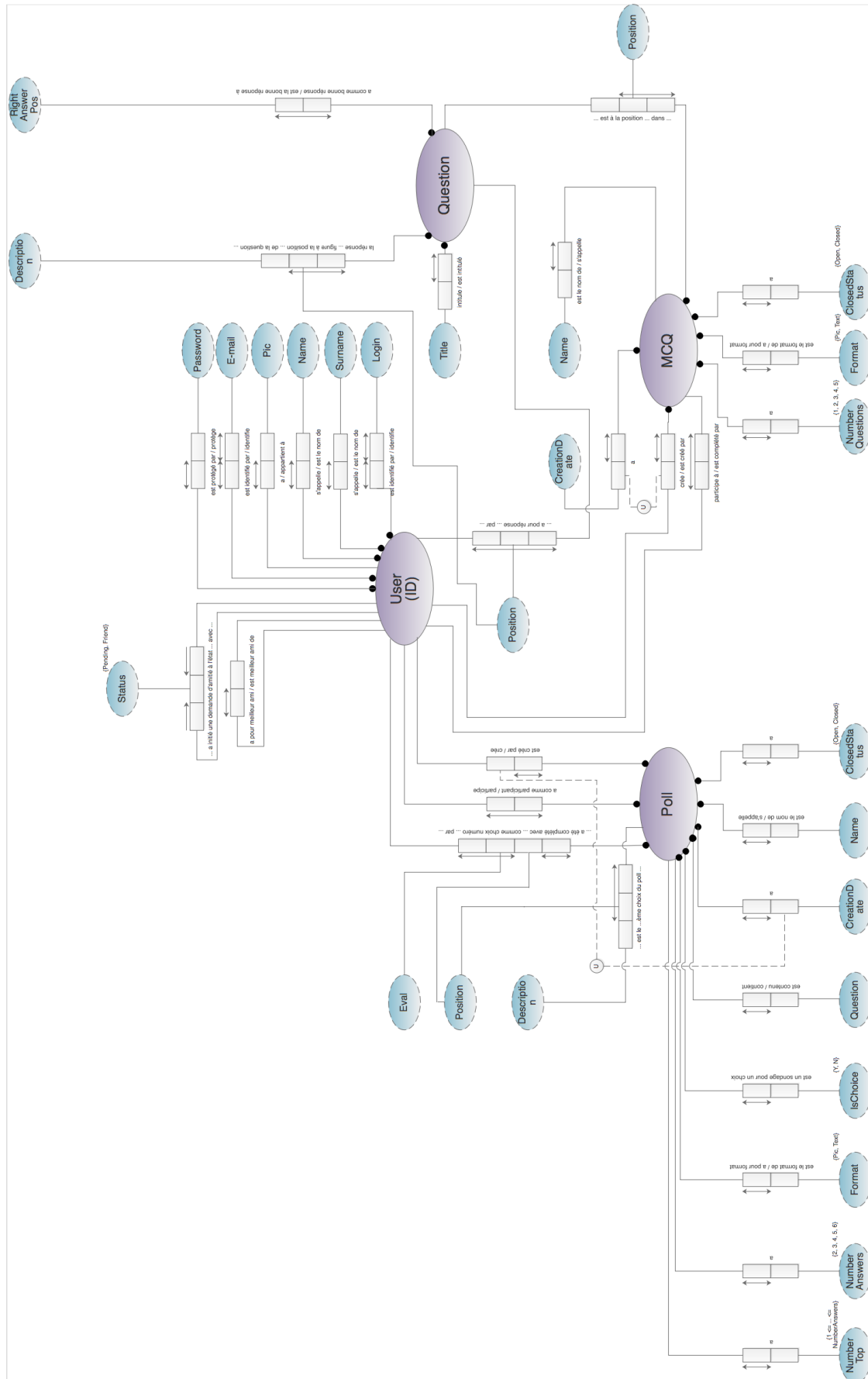
- Poids : 3 - En tant que créateur de sondage, je veux pouvoir ajouter une question au sondage afin de demander l'avis/tester la connaissance de mes amis.¹
- Poids : 3 - En tant que créateur de sondage, je veux pouvoir ajouter un choix à une question afin de proposer une possibilité de réponse pour les participants.
- Poids : 3 - En tant que créateur de sondage, je veux pouvoir spécifier le/la < nom, format, nombre de questions, longueur de la liste de préférence > afin de personnaliser mon sondage.
- Poids : 3 - En tant que participant/créateur d'un sondage, je veux pouvoir accéder à la fiche d'identité d'un sondage afin de voir rapidement les informations relatives à ce sondage.
- Poids : 3 - En tant que créateur de sondage, je veux pouvoir sélectionner des utilisateurs afin de les faire participer à ce sondage.
- Poids : 3 - En tant que créateur de sondage, je veux pouvoir prévisualiser mon sondage afin de voir si il correspond au résultat désiré.
- Poids : 1 - En tant que créateur de questionnaire, je veux pouvoir désigner une réponse comme correcte afin de permettre la comptabilisation des scores des participants.
- Poids : 1/2 - En tant que créateur de sondage, je veux pouvoir clôturer un sondage prématurément afin d'archiver ce dernier.²

Participant à un sondage

- **Modification : Poids 10 - En tant que participant à un sondage, je veux pouvoir accéder à la liste des sondages auxquels je participe.**
- Poids : 5 - En tant que participant à un sondage, je veux pouvoir donner mes choix afin de répondre au sondage.
- Poids : 5 - En tant que participant à un sondage, je veux pouvoir voir les résultats anonymisés de ce sondage afin de voir la décision qui a été prise/voir mon score à un questionnaire.¹
- Poids : 3 - En tant que participant à un sondage, je veux pouvoir modifier mon choix afin d'actualiser ma réponse.
- Poids : 1 - En tant que créateur de questionnaire, je veux pouvoir définir l'ordre des questions afin de personnaliser mon questionnaire.

2. Dans le cas où un sondage n'est plus d'actualité et que l'un des participants n'a jamais répondu.

Schéma ORM

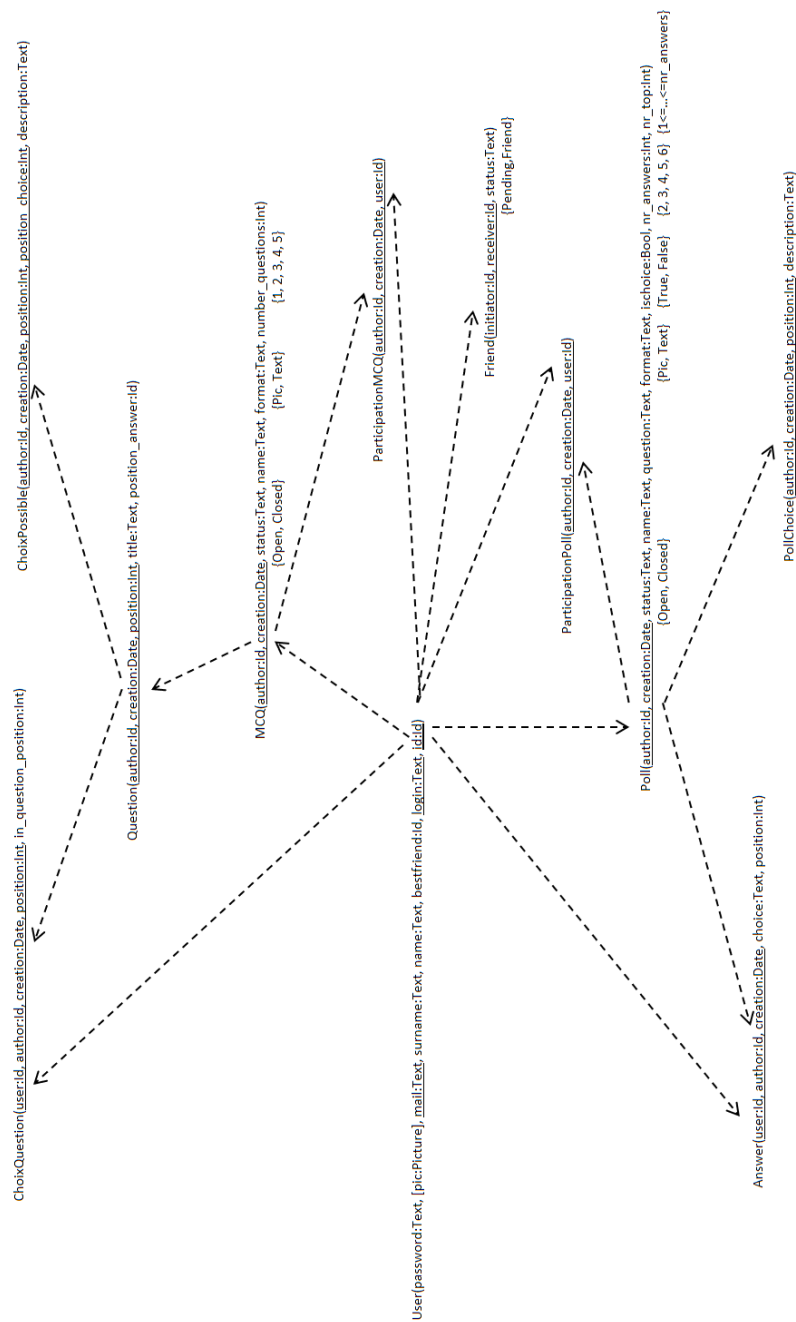


Un seul changement a été apporté à notre schéma conceptuel ORM. Il s'agit des réponses faites par chaque *User* à un *Poll*, qui sont dorénavant identifiées par la combinaison de *User*, *Poll*, *Position* et *Eval* (*Description* remplaçait *Position*).

Section 5

Schéma relationnel

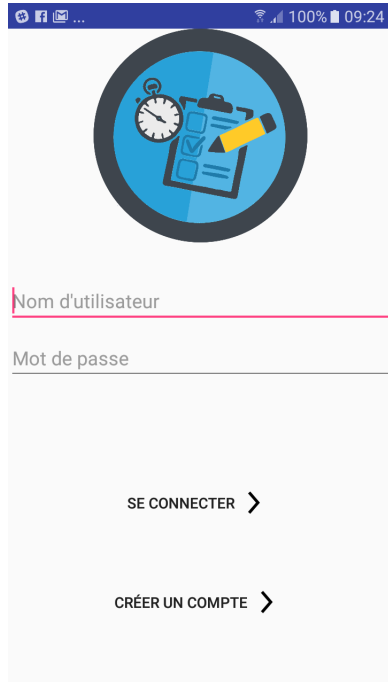
Aucune modification n'a été apportée depuis la première version.



Section 6

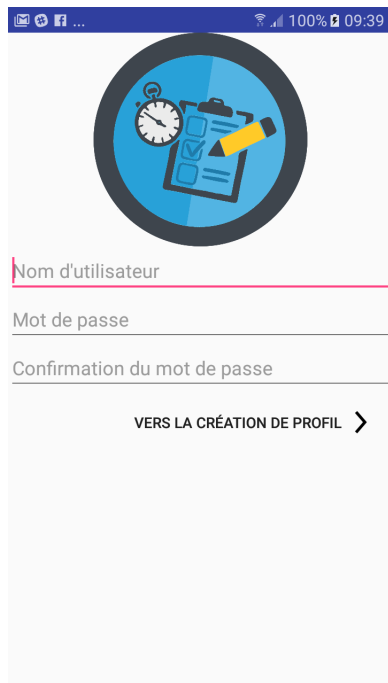
Manuel d'emploi

6.1 LoginActivity



Tout d'abord, une fois l'application MiniPoll lancée. Nous arrivons sur le Login où nous avons le choix de soit se connecter, soit créer un compte.

6.2 Créer compte



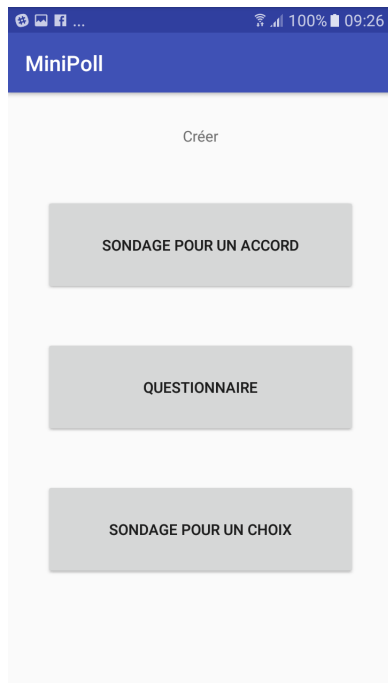
Après avoir cliqué pour créer un compte, on arrive sur cette frame où l'on peut encoder un nom d'utilisateur (unique) ainsi qu'un mot de passe à mettre deux fois. Et envoie un message d'erreur si les mots de passe de son pas similaire. Puis on arrive sur une autre frame où l'on peut rentrer son adresse mail (unique, format email valable) , prénom, nom de famille.

6.3 Menu



Menu principal de l'application qui permet d'accéder aux 6 sous-menus de l'application.

6.4 Menu de création



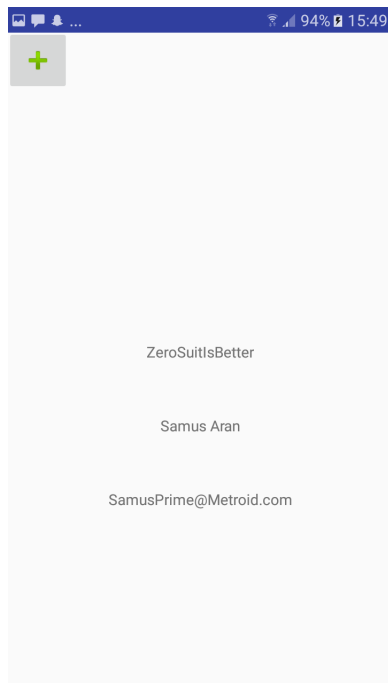
Menu de création qui permet d'accéder aux créations du Sondage pour accord, sondage pour choix, questionnaire.

6.5 Menu d'amis



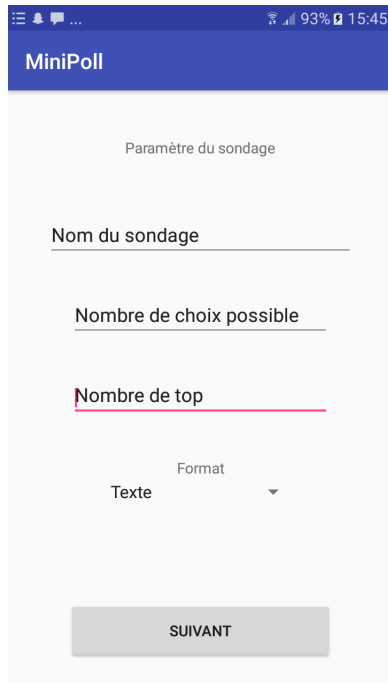
Menu d'amis qui permet d'accéder à tout nos amis, nos demandes d'amitié et aussi nous permet d'ajouter un ami.

6.6 Ajouter un amis




Frame qui permet de visualiser un User de l'application et de pouvoir l'ajouter.

6.7 Création questionnaire



Frame qui crée un questionnaire, en lui donnant un nom, un format, un nombre de top, et le nombre de choix possibles.

6.8 Ajout de choix aux questions



Frame de création de questionnaire qui permet de rajouter les choix.

Voici une liste d'utilisateurs déjà existants, afin de vous connecter plus facilement lors des tests :

Utilisateur	Mot de passe
Radiant	Aether
ItsMe	Peach
ZeroSuitIsBetter	Ridley
Poltergeist	Daisy

Section 7

Choix de conception

Certains choix ont dû être faits pour pouvoir implémenter correctement l'application à nos yeux. En voici les principaux :

- la commande `rawQuery` de la classe `SQLiteDatabase` a été employée plusieurs fois, au lieu de la commande `query`. Plus archaïque, elle nous semblait plus intéressante pour être plus près de nos requêtes SQL. Elle est également plus simple d'emploi à nos yeux pour pouvoir par exemple demander le `MAX(ID)` dans notre table de `User` ou encore pour des requêtes ;
- nous avons premièrement pensé mettre les ID des utilisateurs en *String*. Toutefois, il devient dès lors plus coûteux de comparer tous ces ID pour en trouver un non utilisé. Nous les avons donc changés en `int` afin d'avoir des méthodes plus rapides et plus simples d'emploi ;
- les dates sont enregistrées au format `int` plutôt qu'en *datetime* dans notre base de données SQL. L'utilisateur ne les employant jamais, le choix du type importe peu pour lui. Il est dès lors plus simple de les enregistrer grâce à la commande `java.lang.System.currentTimeMillis()` qui nous renvoie un `long` sauvegardé en `int` dans la base de données ;
- conscients du problème de sécurité que pose l'enregistrement de mots de passe en simple `String`, nous aurions bien voulu pouvoir les passer par un *hash*. Faute de temps, cette idée est simplement restée en suspens. Elle aurait été faite avec plus de temps disponible ;
- Comme dit brièvement lors de la discussion du diagramme de séquence de la création de profil (2.3), nous avons décidé de séparer la création de profil et la modification de profil. Il s'agit là plus d'un choix de facilité d'implémentation qu'un choix dû à un réel avantage de la conception choisie.

Section 8

Participation respective et organisation du groupe

Nous devons coder l'application MiniPoll et faire un rapport. Une partie du travail a été assignée à chaque membre du groupe :

- Baptiste s'est occupé des classes modèles (Poll, PollAnswer, MCQ, MCQAnswer, Question) ainsi que les fonctions y figurant et une partie du rapport final ;
- Augustin s'est chargé de la base de données (SQLiteHelper), d'ajouter des amis, de la création de questionnaire et de la liste d'amis avec les checkbox ;
- Gilles a géré la connexion/enregistrement/création/modification de compte/profil, des problèmes sur *git*/Github, d'une partie du rapport et de la réponse à un questionnaire (Activity) ;
- Hadrien s'est occupé de la coordination du groupe, de toute la classe User ainsi que les méthodes y figurant, de la résolution des *bugs* majeures ;
- Guillian s'est occupé du SQLite, Gestions des amis, Menu principal.

Plusieurs problèmes se sont posés lors de l'implémentation en *Java Android*. Le plus important étant l'impossibilité de passer facilement (à nos yeux du moins) des objets d'activité en activité. L'étape de prévisualisation d'un questionnaire a ainsi été abandonnée, faute de temps pour implémenter une solution efficace pour passer les objets *Question* générés au fur et à mesure.

Nous avons également eu des problèmes avec notre synchronisation par *Github*. Travaillant de temps en temps sur les mêmes classes chacun de notre côté, nous avons dû faire plusieurs *merges* afin d'harmoniser le rendu. Gilles a beaucoup structuré notre répertoire pour éviter de tels problèmes. Dans cette même optique d'éviter le plus possible les conflits, un fichier `.gitignore` personnalisé a également été inestimable.

Conclusion

Nous sommes désormais à même de proposer une application quasi-fonctionnelle, permettant de créer et de participer à des sondages comme à des questionnaires et de gérer une liste d'amis. Un rendu visuel plus agréable aurait été appréciable, mais la fonctionnalité l'emporte sur le rendu. Avec un retour positif de notre application ainsi que plus de temps permis, nous pourrions l'étoffer. Une envie de demander à vos amis quels films ils veulent aller voir ce soir ? Finies les discussions avec des centaines de messages, *MiniPoll* fera le travail pour vous !