



Rapport 3 - Développement Android

LSINF1225 Conception orientée objet et gestion de données - 2018-2019

GROUPE J1B

LOUIS BURET

MARTIN DRAGUET

MAXIMILIEN LA BARRE

JOHAN VAN ROY

TIMOTHÉE CLEMENT

RAPHAËL MUGENZI BYIRINGIRO

Professeur : Kim MENS

Assistante : Julie CHATELAIN

Introduction

Ce rapport a pour objectif de rassembler les grands schémas descriptifs de la structure et du fonctionnement final de l'application Android "IQWhizz" réalisée dans le cadre du cours LSINF1225. Le travail a été réparti au sein du groupe en fonction des compétences de chacun :

- Johan, Martin, Louis et surtout Maximilien se sont concentrés sur le codage de l'application en Android Studio.
- Raphaël et Timothée (moins à l'aise en Java) ont participé à la mise à jour des différents diagramme pour le rapport (en cohérence avec les choix de conception faits par l'équipe de codage) et à la rédaction de celui-ci.

Table des matières

| | |
|--|----------|
| 1 Schéma conceptuel ORM | 2 |
| 2 Schéma relationnel de la BDD | 3 |
| 3 Récits utilisateurs | 4 |
| 3.1 Gestion des utilisateurs | 4 |
| 3.2 Gestion des questionnaires | 4 |
| 3.3 Gestion des reviews | 4 |
| 3.4 Gestion des amitiés | 5 |
| 4 Diagramme de classe UML | 6 |
| 5 Diagramme de séquence UML | 7 |
| 5.1 Diagramme de séquence UML pour l'enregistrement (création d'un compte utilisateur) | 7 |
| 5.2 Diagramme de séquence UML pour le login (identification avec un compte utilisateur existant) . . | 7 |
| 5.3 Diagramme de séquence UML pour les "reviews" | 8 |
| 5.4 Diagramme de séquence UML pour l'ajout d'un ami | 8 |
| 6 Choix de conception | 9 |
| 6.1 Lancement de l'application | 9 |
| 6.2 Lancement d'un quiz | 9 |
| 6.3 Quiz | 10 |
| 6.4 Classement | 10 |
| 6.5 Profil | 10 |
| 6.6 Amis | 10 |
| 6.7 Mode d'emploi rapide | 10 |

1 Schéma conceptuel ORM

Le schéma Object-Role-Modelling est présenté ci-dessous. Quelques changements y ont été apportés depuis la remise du premier rapport. Premièrement, l'entité "image" et la valeur "nombre de parties jouées" ont été enlevés, ces fonctionnalités n'ayant pas été ajoutées à l'application pour des raisons de manque d'utilité. Deuxièmement, les contraintes d'unicité pour les relations {mot de passe, utilisateur} et {question, questionnaire} ont également été corrigées. Tel que cela est implémenté dans l'application, une question est associée à un et un seul set de questions (questionnaire). Par conséquent, si l'on désire qu'elle apparaisse dans différents questionnaires, il faudra l'encoder plusieurs fois en l'associant avec des questionnaires différents.

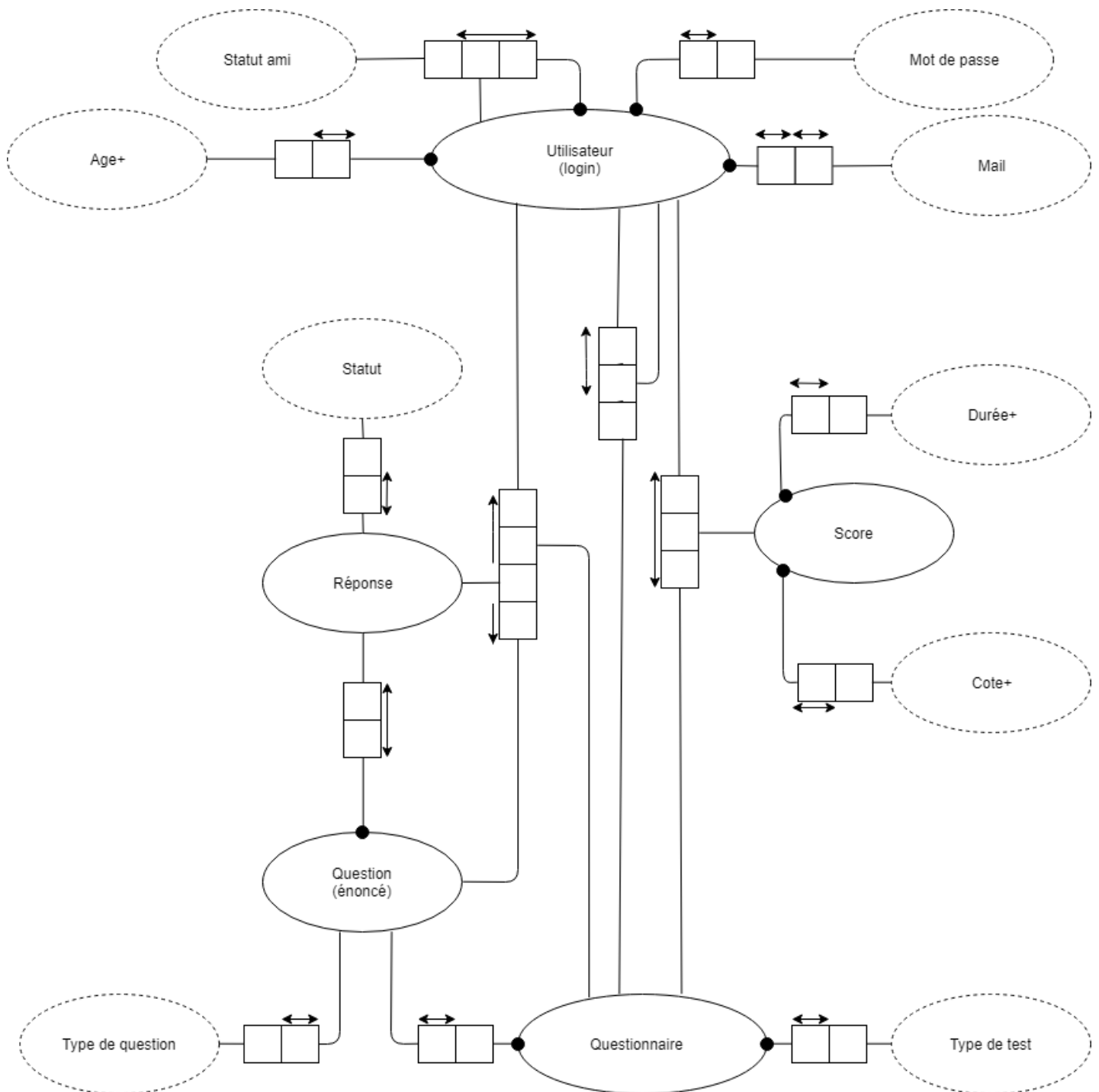


Figure 1: Schéma ORM de l'application IQWhizz

2 Schéma relationnel de la BDD

Le schéma relationnel de la base de données telle qu'elle est réellement implémentée en Android Studio est présenté ci-dessous. Depuis le second rapport, nous avons ajouté les flèches indiquant les références pour les clés étrangères. Une classe "Défi" pour les duels entre deux amis a également été ajoutée. Cependant, la classe "Review" a été supprimée car les variables permettant de voir une révision après un questionnaire ne sont stockées que temporairement pour permettre la visualisation des bonnes et mauvaises réponses juste après le questionnaire, mais ces informations ne sont pas stockées définitivement dans une table de la base de données. On considère en effet que l'utilisateur s'intéresse à la review juste après avoir répondu au quizz mais a peu d'intérêt à retourner voir une review d'un questionnaire réalisé plusieurs jours auparavant. De même, la classe score a été intégrée comme un attribut de la classe "Utilisateur" et la classe "réponse" est intégrée à la table "Question" car à chaque question est associée 4 réponses (3 mauvaises et une bonne).

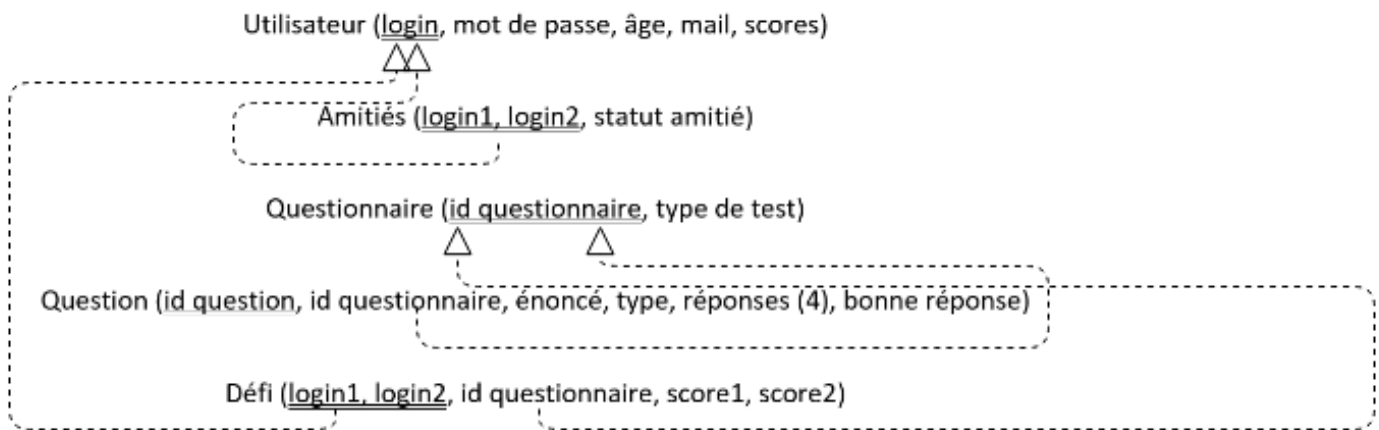


Figure 2: Schéma relationnel de la base de données

3 Récits utilisateurs

Les récits utilisateurs décrivent les actions qui doivent pouvoir être effectuées par les différents rôles de l'application dans un certain but. Chaque récit est accompagné d'un chiffre décrivant le temps estimé pour l'implémentation du récit sur Android Studio. Des changements ont été faits depuis le second rapport rendu. Nous avons classé les différents récits en catégories auxquelles ils s'y rapportent. Certains récits ont également été modifiés. Nous avons par exemple supprimé les récits liés aux avatars de l'utilisateur et à la visualisation du nombre de parties jouées car nous n'avons pas implémenté ces options finalement. Nous avons également regroupé la modification des informations du profil (âge, mot de passe, adresse-mail) en un seul récit car cela est plus cohérent. Les récits d'envoi et de réception de demandes d'amis ont été transformés en un récit d'ajout d'amis, plus fidèle à ce qui a réellement été implémenté dans l'application.

3.1 Gestion des utilisateurs

Tous les récits liés à la création et à la modification des profils utilisateurs.

- ◇ **Connection** : En tant qu'utilisateur, afin d'utiliser l'application IQWhizz, je veux pouvoir rentrer mon login et mon mot de passe pour me connecter. (8)
- ◇ **Enregistrement** : En tant qu'utilisateur, je veux créer un compte, j'ai besoin de pouvoir entrer un login, un mot de passe, mon âge, mon adresse mail. (8)
- ◇ **Modification profil** : En tant qu'utilisateur, je veux actualiser les informations de mon profil, j'ai besoin de pouvoir modifier mon âge, mon mot de passe, mon adresse e-mail. (1)

3.2 Gestion des questionnaires

Tous les récits liés à la réalisation des questionnaires.

- ◇ **Liste de test** : En tant qu'utilisateur, afin de lancer un test au choix, je veux pouvoir accéder à une liste de tests de différents types et d'en sélectionner un. (2)
- ◇ **Questionnaire** : En tant qu'utilisateur, je veux répondre à un questionnaire de QI, j'ai besoin de pouvoir sélectionner un questionnaire et répondre aux questions. Étant donné que la réponse aux questions est minutée, le délai d'apparition des questions ne doit pas durer plus de quelques secondes (besoin non-fonctionnel). (30)

3.3 Gestion des reviews

Tous les récits liés à la (re)visualisation des scores :

- ◇ **Score** : En tant qu'utilisateur, je veux avoir accès à mes scores, j'ai besoin de pouvoir les afficher de manière claire et lisible, pour chaque test réalisé. (3)
- ◇ **Leaderboard** : En tant qu'utilisateur, afin d'avoir une idée de ma situation par rapport à la moyenne, je veux pouvoir comparer mon QI au niveau mondial (comparé à ceux de mes amis). (3 - 20)
- ◇ **Review** : En tant qu'utilisateur, afin de connaître les résultats d'un test que je viens de réaliser, je veux pouvoir accéder à un feedback (review) exhaustif de ce test (reprenant toutes les questions avec mes bonnes et mauvaises réponses). (13)

3.4 Gestion des amitiés

Tous les récits liés aux possibilités d'interaction et d'actions avec ses autres utilisateurs "amis" sur l'application :

- ◇ **Ajout d'ami** : En tant qu'utilisateur, je veux pouvoir ajouter un ami afin de faire des duels avec lui et/ou de comparer nos scores, je dois pouvoir entrer son login, l'ajoutant ainsi à ma liste d'amis. **(5)**
- ◇ **Suppression d'ami** : En tant qu'utilisateur, je veux supprimer un de mes amis car je n'ai plus envie de pouvoir faire des duels et/ou comparer nos scores, j'ai besoin de pouvoir le retirer de ma liste d'amis. **(5)**
- ◇ **Profil d'amis** : En tant qu'utilisateur, je veux pouvoir voir le profil de mes amis afin d'avoir plus d'informations sur eux, je dois avoir accès à leur profil. **(3)**
- ◇ **Défi Ami** : En tant qu'utilisateur, afin de défier mes amis, je veux pouvoir jouer avec eux à des questionnaires de type "duel". **(20)**
- ◇ **Score ami** : En tant qu'utilisateur, je veux pouvoir accéder à mes résultats en tout temps, ainsi qu'à ceux de mes amis. **(5)**
- ◇ **Comparaison Défi ami** : En tant qu'utilisateur, afin de comparer nos résultats après un duel, je veux pouvoir visualiser les résultats de l'ami que j'ai défié. **(3)**

4 Diagramme de classe UML

Le diagramme de classe UML définitif est présenté ci-dessous. Depuis le second rapport, un certain nombre de changements ont été apportés à celui-ci.

Premièrement, la classe "image" ainsi que les associations liées à cette classe ont été supprimées car nous n'avons finalement pas implémenter cette possibilité.

Deuxièmement, la classe "réponse" n'existe plus comme une classe à part entière mais a été intégrée comme attribut de la classe "question". En effet, à chaque question est associée 4 réponses (3 mauvaises et une bonne). La même opération est exécutée pour la classe score, car les scores sont encodés dans la base de données comme un attribut de l'utilisateur. Par contre, on ajoute la classe "Classement" qui permet de calculer le leaderboard (top 10 ou 100) des meilleurs scores réalisés par les différents utilisateurs.

Troisièmement, des **directions de navigation** ont été ajoutées aux associations. A titre d'exemple, voici une justification relative à une direction d'association ajoutée : l'association "utilisateur - amis" fonctionne uniquement dans ce sens car on ne part pas du statut d'amitié pour trouver un utilisateur mais on se demande quel est le statut d'amitié entre deux utilisateurs

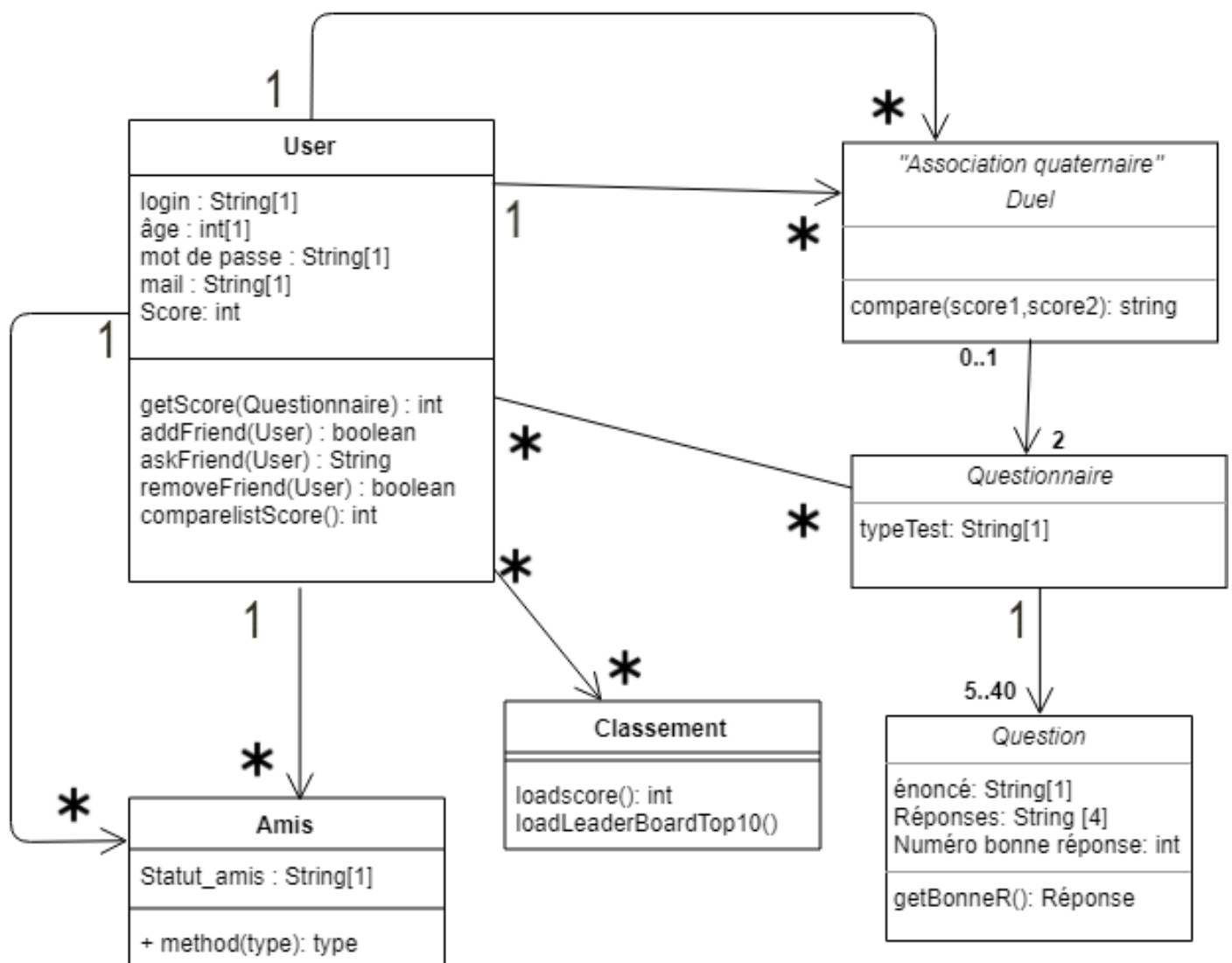


Figure 3: Diagramme de classe UML de l'application IQWhizz

5 Diagramme de séquence UML

Ci-dessous, nous présentons trois diagrammes de séquence UML décrits dans les récits utilisateurs. Ces diagrammes de séquences décrivent la dynamique temporelle des flux d'informations qui ont lieu entre les différentes classes et objets de l'application lors de la réalisation du récit. Depuis le second rapport, ces diagrammes ont été corrigés. Premièrement, nous avons adapté les noms des fonctions afin qu'elles correspondent à celles encodées dans Android studio. Deuxièmement, des flèches doublons ont été supprimées, notamment avant et après les boucles du diagramme d'enregistrement. Troisièmement, des ":" et des soulignements que nous avons oublié pour certaines instances ont été ajoutés. Troisièmement, les débuts et fin des barres d'activation ont également été revus. D'autres petites corrections spécifiques ont également été faites et sont précisées dans les sous-sections des diagrammes correspondants.

5.1 Diagramme de séquence UML pour l'enregistrement (création d'un compte utilisateur)

La deuxième boucle a été corrigée ; elle check en effet le mot de passe et nom le nom d'utilisateur. De plus, la flèche de cette boucle est interne au "Sign in menu" et ne passe pas par la classe utilisateurs (contrairement à ce qui avait été présenté au second rapport).

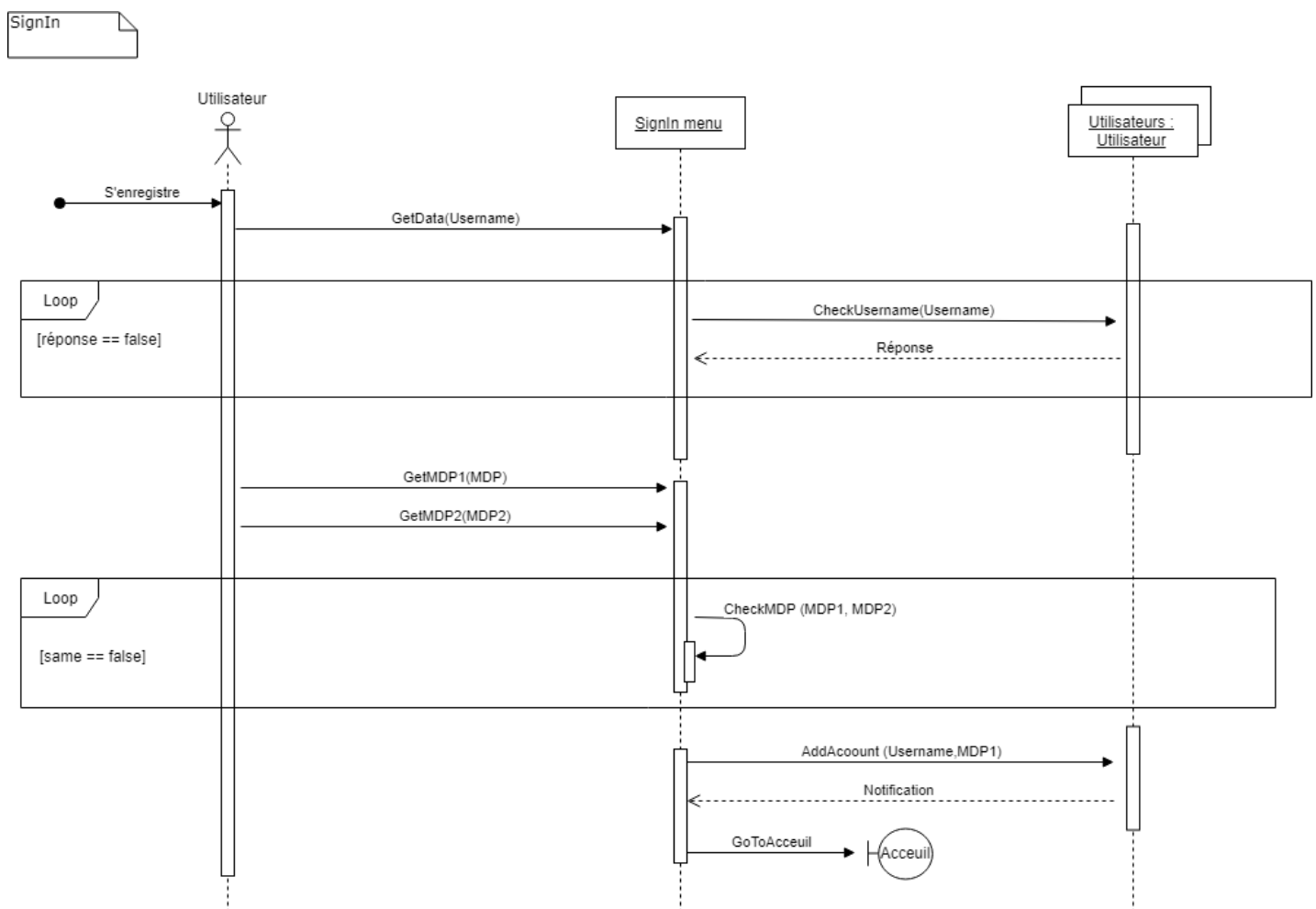


Figure 4: Diagramme de séquence UML pour l'enregistrement d'un nouvel utilisateur

5.2 Diagramme de séquence UML pour le login (identification avec un compte utilisateur existant)

Voici le diagramme décrivant la connexion d'un utilisateur sur l'application. Celui-ci n'a pas subi de changement mis à part la suppression de flèches doublons avant la boucle.

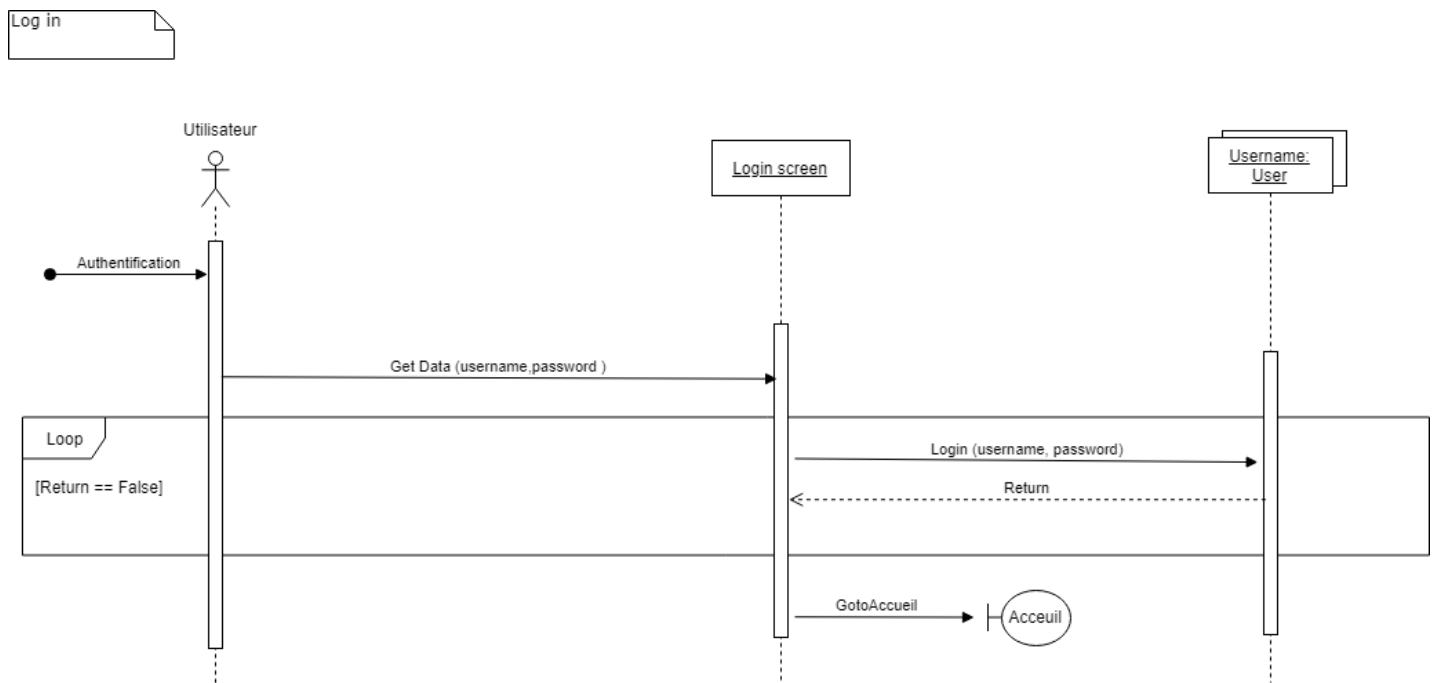


Figure 5: Diagramme de séquence UML pour l'identification d'utilisateur connu par l'application

5.3 Diagramme de séquence UML pour les "reviews"

Cette séquence UML a été remplacée par un simple Layout. A la fin du test, si l'utilisateur veut accéder à son review nous avons incorporé un bouton qui l'emmène directement vers une page avec l'ensemble des questions corrigées.

5.4 Diagramme de séquence UML pour l'ajout d'un ami

Ce diagramme diffère de l'ancien car dans notre application nous n'avons pas réussi à implémenter un refus ou une acceptation de la demande d'ami. A la place, l'utilisateur 1 (Paul dans cet exemple) doit connaître le username de l'utilisateur 2 (Bob) et s'il choisit de l'ajouter ils deviennent amis. Nous avons tout de même implémenté la fonction `removeFriendUser` qui permet à l'utilisateur 2 de supprimer ce nouvel ami s'il le désire.

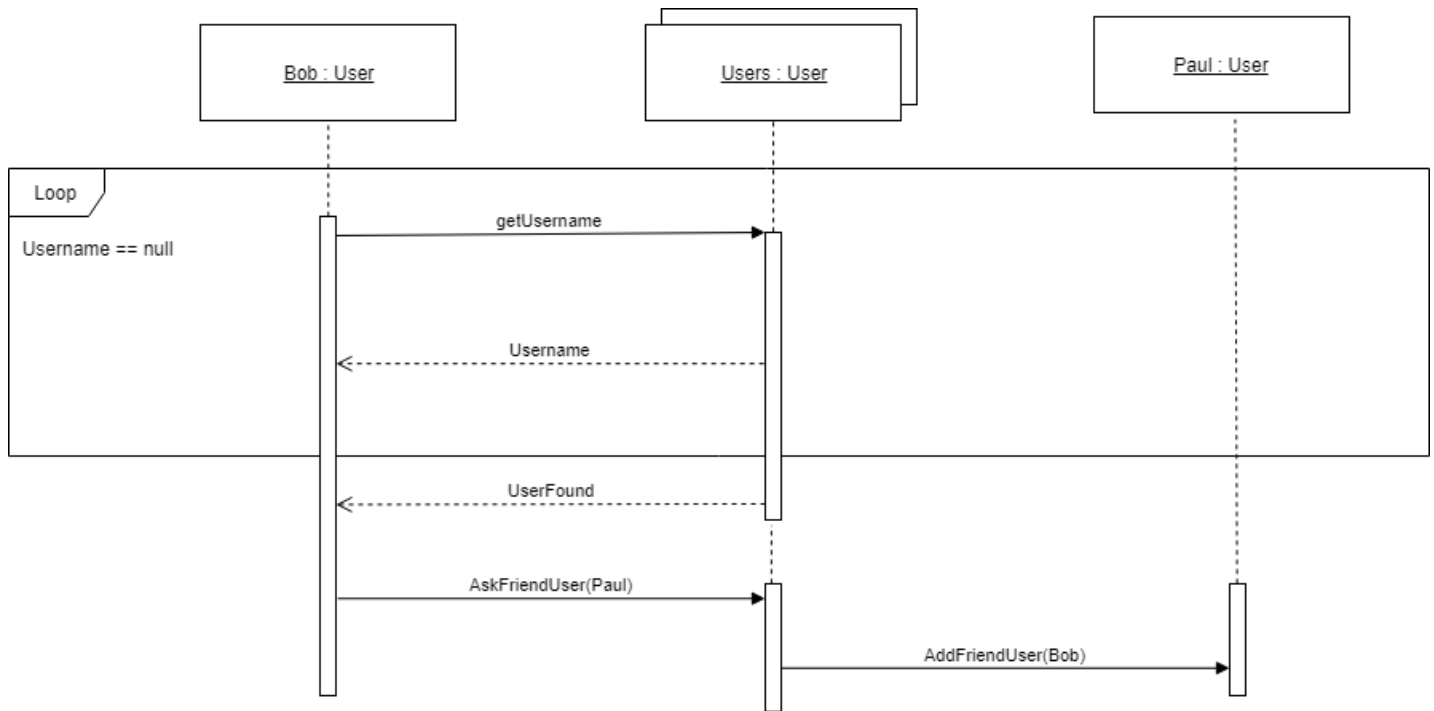


Figure 6: Diagramme de séquence UML pour l'ajout d'un ami

6 Choix de conception

Dans cette section de notre rapport final, nous allons reprendre en long et en large les choix de conception de notre application "IQWhizz".

6.1 Lancement de l'application

Pour commencer, nous avons décidé que lorsque l'utilisateur lance l'application (si elle n'est pas lancée en arrière plan), la première chose qu'il devra faire est de se connecter sur l'application. Si l'utilisateur n'a pas encore de compte, il aura alors la possibilité de s'enregistrer, ce qui l'ajoutera dans la base de données (dans la table des utilisateurs). Cette table contiendra alors toutes les informations nécessaires pour lui permettre de se connecter mais aussi modifier son profil, ce point étant davantage expliqué dans la suite de cette section du rapport.

6.2 Lancement d'un quiz

Une fois connecté sur l'application, l'utilisateur est envoyé sur l'écran d'accueil, qui est l'endroit où il pourra lancer les quiz, accéder au classement de l'application ou encore son profil. Il est très facile de lancer un quiz pour un utilisateur : ce dernier n'a qu'à sélectionner, directement à partir de l'écran d'accueil, les différents sets de question (40 pour un test normal et 5 pour un test rapide) ainsi que la catégorie de question dans laquelle l'utilisateur veut jouer (Exemples de catégories : programming, math, géographie, etc). Ensuite, il a le choix entre deux boutons : le "quiz normal" pour 40 questions et le "quiz rapide" pour 5. Mais comment cela fonctionne-t-il ? nous avons créé deux tables dans notre base de données , une pour les questions (énoncé, option 1 à 4, le numéro de la bonne réponse ainsi que la catégorie et le set de question) et une pour les catégories. En réalité ces deux tables sont liées puisque dans la table de questions se trouve la catégorie de cette question. La catégorie est donc une "foreign key" à la table de questions. Tout cela permet qu'on puisse "choisir" directement à la création de la question dans quel set et quelle catégorie la question se trouve, mais pas seulement. En effet, c'est aussi grâce à cela que l'utilisateur peut choisir la sorte (catégorie et set) de quiz qu'il veut lancer.

6.3 Quiz

Maintenant que l'utilisateur a choisi son quiz, il a 60 secondes pour répondre à la question. S'il ne répond pas, la question suivante sera affichée et le score ne sera pas incrémenté, de même s'il s'est trompé à la question. Sur cet écran, nous avons donc décidé d'afficher en haut à gauche le numéro de la question à laquelle l'utilisateur est en train de répondre et le nombre total de questions du quiz, la catégorie choisie ainsi que le set de question. En haut à droite se trouve le chronomètre qui devient rouge en dessous de 10 secondes pour que l'utilisateur remarque qu'il lui reste peu de temps pour répondre à la question. Après avoir choisi une réponse, l'utilisateur peut confirmer sa réponse en cliquant sur le bouton, ce qui affichera la question suivante, incrémentera le nombre de questions auquel il a répondu et remettra le compteur à 60 secondes. Le quiz se termine quand l'utilisateur a répondu à toutes les questions. Le score est alors sauvegardé dans la table utilisateur et ce dernier aura la possibilité d'avoir une review des réponses. Dans cette review il verra tout d'abord son QI final et, en cliquant sur le bouton adéquat, aura la possibilité de voir toutes les questions auxquelles il a répondu dans le quiz, ainsi que la réponse choisie et la bonne réponse. Ceci n'étant pas stocké dans la base de données, tout est conservé et enregistré dans le code.

6.4 Classement

Ensuite, l'utilisateur peut retourner dans le menu d'accueil et aller voir son classement dans l'application. Le classement est un endroit où les dix plus gros scores enregistrés dans l'application sont affichés, ce qui lui permet de se comparer aux autres utilisateurs et d'instaurer un sentiment de compétition pour être le premier de ce classement. Cela a été implémenté de la sorte que nous récupérons les 10 meilleurs résultats en fonction de la colonne score de notre table, ainsi que le nom de l'utilisateur, ce qui, avec l'aide d'une "ListView", permet d'afficher un classement assez propre. En dessous, on peut observer un autre bouton qui va permettre d'afficher de nouveau sous forme de "ListView", l'entièreté des utilisateurs enregistrés sur l'application avec leurs scores respectifs.

Par la suite, nous appliquons la formule sur le score pour afficher le QI final sur les deux classements.

6.5 Profil

Une autre fonctionnalité se trouve à l'écran d'accueil : l'utilisateur a accès à son profil d'où il aura la possibilité de modifier toutes les informations qu'il a entrées lors de son enregistrement, c'est à dire son pseudo, son adresse mail, son âge et son mot de passe. Après avoir précisé les modifications qu'il souhaite apporter, la base de données est mise à jour avec les nouvelles informations. Il peut alors se connecter et jouer avec son compte modifié.

6.6 Amis

De retour à l'écran d'accueil, l'utilisateur aura la possibilité d'ajouter des amis en cliquant sur le bouton "Amis". De là, il pourra ajouter des amis en entrant leur nom. Tout cela est stocké dans la base de données, ce qui permet de ne pas ajouter plusieurs fois la même personne, et accessoirement ne pas s'ajouter soi-même. Puisque notre système n'envoie pas de requête, cela veut dire que l'on peut ajouter n'importe qui, bien que la liste d'amis reste propre à chaque utilisateur. Nous comptons sur le fait que quand un utilisateur veut défier un ami, il soit à côté de cet ami et voit au final qui a la plus de QI sur le même test. De cet écran, il a aussi la possibilité de supprimer un ami s'il le désire.

Pour lancer un quizz avec un ami, il faut que les deux amis soient l'un à côté de l'autre. En allant dans sa liste d'ami, l'utilisateur peut cliquer sur le pseudo de son ami et ainsi jouer avec lui. Pour cela il faut en réalité lancer deux fois le quiz pour avoir le comparatif final des QI et voir qui est le plus intelligent des deux.

6.7 Mode d'emploi rapide

Pour conclure cette section, nous allons finir par un bref mode d'emploi de l'application. Lors du lancement de celle-ci, si l'utilisateur ne s'est jamais connecté il aura la possibilité de s'enregistrer dans l'application, ensuite il est redirigé vers le menu de connexion (menu qu'il verra chaque fois qu'il quittera l'application si elle n'est plus en tâche de fond). Après avoir entré ses identifiants (si ceux-ci sont corrects), l'utilisateur sera envoyé à l'écran d'accueil.

De là, plusieurs choix s'offrent à lui : Il peut choisir de se déconnecter. Ou alors il peut choisir de lancer différents quizzs ; un test rapide (5 questions) et un test normal (40 questions). Pour cela, il n'a qu'à choisir entre les 3 catégories disponibles dans l'application et un des 3 sets de questions. Après avoir complété le quizz, l'utilisateur aura deux choix : soit revenir à l'écran d'accueil, soit voir la review des questions du quizz fraîchement réalisé, pour voir où il s'est trompé.

Après avoir vu ses erreurs en regardant la review, l'utilisateur pourra revenir au menu d'accueil. De là il pourra aller voir son classement pour se comparer aux autres utilisateurs, avec la possibilité d'avoir le classement total et un top 10.

Ensuite, en revenant à l'écran de départ, l'utilisateur pourra consulter les informations de son profil (age, mot de passe, mail) où il a la possibilité de les modifier.

Enfin, le bouton "amis" de l'écran d'accueil permettra à l'utilisateur connecté d'ajouter des amis et de voir sa liste d'amis, mais aussi de lancer des quizz contre eux afin de comparer leur QI.