

ÉCOLE POLYTECHNIQUE DE MONTRÉAL



Hiver 2017

INF8405 - TP2

BLANCHE PAIEMENT (1682019)

ALAIN TRAN-DANG (1687869)

ALEXIS ST-ONGE (1688297)

Travail présenté à Fabien Berquez,
dans le cadre du cours de INF8405

23 mars 2017

Introduction

L'apprentissage de la programmation Android à l'aide de l'IDE Android Studio ainsi que de l'utilisation des capteurs de la tablette s'est poursuivi durant ce deuxième travail pratique du cours d'INF8405. Le travail en soi a été de concevoir une application qui permet de créer et gérer des événements (ou activités) partagés avec plusieurs utilisateurs de l'application. Les utilisateurs doivent prendre une photo à la création de leur profil et il leur est demandé de créer ou rejoindre un groupe. Par la suite le créateur du groupe peut suggérer des lieux pour des activités et lorsque les autres membres ont votés pour l'activité qu'ils veulent, l'organisateur peut créer l'événement et tout le monde peut voir la position géographique des autres membres du groupe ainsi que leur statut de participation à l'événement. Dans les lignes qui suivent une présentation technique du travail sera faite ainsi qu'une description des problèmes rencontrés lors de son accomplissement et des suggestions d'améliorations qui pourraient y être apportées.

Présentation technique

Connexion d'un utilisateur

Au démarrage de l'application, l'utilisateur peut s'identifier, c'est-à-dire donner son nom d'utilisateur ainsi que son mot de passe. Pour notre application, nous avons décidé d'utiliser Firebase afin de stocker les informations sur nos utilisateurs. Donc pour vérifier les informations de l'utilisateur, nous allons simplement voir dans Firebase s'il existe un utilisateur qui répond à ces informations, sinon, un message d'erreur est retourné. Si c'est la première fois que l'utilisateur utilise l'application, il a l'option de se créer un compte. Ainsi, il doit créer un nom d'utilisateur et un mot de passe et il ne peut pas s'enregistrer tant qu'il ne prend pas une photo de profil. Pour prendre une photo, le concept est assez simple, il suffit de cliquer sur la zone réservé à la photo et l'application change pour la caméra de l'appareil. L'utilisateur a toujours l'option de modifier sa photo avant de s'enregistrer. Nous vérifions aussi qu'il n'existe pas déjà un utilisateur avec le même nom d'utilisateur pour éviter d'éventuels problèmes.

Choix du groupe

Une fois que l'utilisateur s'est identifié, il doit choisir le groupe auquel il désire se joindre. Pour ce faire, nous avons choisi un "AutoCompleteTextView" qui fait en sorte qu'il suffit de taper quelques lettres pour se voir proposer des noms de groupe existants. Nous avons fait ce choix tout d'abord pour des fins de simplification de nos tests pendant le développement de notre application, mais nous considérons qu'il s'agit également d'un élément qui ajoute de la valeur à notre application étant donné que nous accompagnons l'utilisateur dans son choix de groupe en l'aidant. Pour la gestion des choix de groupe, il y a plusieurs possibilités: un utilisateur qui est le "propriétaire" du groupe n'a d'autre choix que de choisir son groupe

sinon une erreur va s'afficher. Nous avons posé le fait qu'un utilisateur ne peut faire partie que d'un seul groupe à la fois, donc quand un membre régulier veut choisir un groupe il peut choisir n'importe lequel des groupes existants. Dans un tel cas, s'il faisait partie d'un groupe auparavant, nous le retirons de ce groupe pour l'ajouter au nouveau groupe. Il peut également créer un nouveau groupe, il devient alors le propriétaire de celui-ci.

Préférences

Il était précisé dans l'énoncé qu'à tout moment, l'utilisateur peut quitter le groupe dans lequel il est présentement et qu'il peut modifier la fréquence d'envoi de sa position. Pour ce faire, nous avons ajouté un menu au haut de l'application. Lorsque qu'on est au niveau du "Login", du "Register" ou encore du choix du groupe, la seule option est de modifier les préférences de l'application. Par la suite, pour les autres activités de notre application, il est possible de quitter le groupe, si l'utilisateur n'en est pas le propriétaire et être redirigé vers le choix d'un nouveau groupe ou encore se "Logout" et être redirigé vers la page de démarrage de l'application.

Utilisation de *Google Maps*

Toutes les fonctionnalités en lien avec Google Maps ont été implémentées dans un fragment, car nous avons déterminé que la carte de Google allait représenter qu'une partie de l'interface utilisateur.

Lorsqu'un nouveau lieu et/ou utilisateur est créé dans un groupe donné, il est stocké dans des hashmap statiques qui contiennent comme clé le nom du lieu ou utilisateur ainsi que leur marqueur respectif.

Lorsque la fonction `OnLocationChanged` est appelée périodiquement selon un intervalle définie, l'utilisateur connectée envoie à Firebase sa position actuelle et recueille la position de tous les autres membres du groupe. Considérant que les requêtes à Firebase sont asynchrones, le traitement de l'envoi ainsi que du recueil des positions sont faites dans une la classe `DatabaseManager`. Pour la requêtes des positions des utilisateurs du groupe, essentiellement, nous balayons les utilisateurs du groupe actuel, recueillons leurs coordonnées et mettons à jour la hashmap statique des utilisateurs du groupe. Après ce traitement, tous les marqueurs des utilisateurs sont mis à jour sur la Google Map. Les membres d'un groupe sont représentés par un marqueur bleu.

Afin de faciliter les choses, nous donnons la possibilité à l'administrateur d'un groupe de créer des lieux directement sur le fragment de Google Map qui se trouve sur son tableau de bord. Tout ce qu'il a à faire, est de cliquer longuement sur l'endroit sur la Google Map où il veut créer son lieu. Par la suite, une fenêtre apparaîtra pour qu'il puisse nommer ce lieu et, finalement, une fenêtre lui demande s'il confirme ce lieu et, s'il accepte, ce lieu sera ajouté à la base de données Firebase et les membres du groupe verra un marqueur rouge apparaître sur leur Google Map pointant à ce nouveau lieu. Le `OnMapLongClick` est gérée dans l'activité qui contient la Google Map, mais le traitement de la création d'un nouveau lieu est

fait dans DatabaseManager. Une fois qu'il y a 3 lieux sur la Google Map, il n'est plus possible pour l'administrateur d'ajouter un nouveau lieu.

Finalement, Il est à noter qu'avant de pouvoir voir la moyenne des notes de chaque lieu du groupe, il faut d'abord qu'un utilisateur donne un vote.

Utilisation de *Firebase* comme base de données

Nous avons utilisé Firebase comme notre base de données principales. Nous y stockons les profils des utilisateurs, les groupes ainsi que les événements, bref tout ce qui a besoin d'être partagé entre les utilisateurs. Firebase est une base de données qui utilise des requêtes asynchrones pour lire et écrire ce qui nous a donné beaucoup de fil à retordre étant donné que ce n'est pas toujours évident. En effet, nous ne pouvons pas faire des requêtes les unes après les autres étant donné qu'il n'y a aucune garantie de l'ordre dans lequel elles seront exécutées, donc cela alourdit beaucoup de code étant donné qu'on veut instaurer un sentiment d'ordre dans les étapes alors qu'il n'y en a pas vraiment. Pour la gestion de toutes nos requêtes vers Firebase, nous avons créé une classe "DatabaseManager" afin de regrouper celles-ci ensemble et de rendre plus agréable la lecture du code des activités.

Utilisation de SQLite comme base de données locale

Étant donné qu'il était mentionné dans l'énoncé qu'il fallait un stockage local, nous avons implanté un SQLiteOpenHelper qui permet de gérer les interactions avec une base de données SQLite sur l'appareil. Pour ce faire, nous avons créé une base de données et nous avons implanté des fonctions pour stocker les informations sur l'utilisateur courant soit son nom d'utilisateur, son mot de passe ainsi que son groupe actuel. Toutefois, lorsque nous avons implanté cette fonctionnalité, nous avons déjà créé des variables statiques dans nos activités qui faisaient en sorte que nous avions accès à ces informations, donc nous l'avons mis en place, mais nous ne l'avons pas intégré à notre application concrètement, c'est-à-dire que la classe est là, mais nous ne l'avons pas instanciée. Il s'agit toutefois d'une solution assez simple que nous allons utiliser pour le projet final.

Gestion du mode économie d'énergie

Pour cette fonctionnalité, nous avons eu beaucoup de difficultés à la tester étant donné qu'il faut que notre appareil ait un niveau de pile assez faible pour que l'exception soit lancée. Nous avons toutefois implanté la solution en ajoutant un "Receiver" au fichier manifest de l'application qui est appelé lorsque la pile devient à un niveau faible. Lorsque celui-ci est appelé, on modifie les préférences pour faire en sorte que la position actuelle de l'utilisateur ne soit envoyée qu'une seule fois par minute.

Gestion des événements

Pour pouvoir créer des événements il fallait que l'organisateur du groupe sélectionne trois lieux, et que tous les utilisateurs les aient noté. Pour ce faire une vérification sur l'élément "hasVoted" de chacun des utilisateurs du groupe est faite lorsque le bouton de création d'événement est activé. Si tous les utilisateurs ont voté, on se rend sur l'activité qui nous permet de créer et de sauvegarder un événement sur la base de données, sinon un simple message est affiché à l'écran pour nous dire qu'il manque des votes et on reste sur l'activité principale. Pour créer un événement il faut lui spécifier un nom, un lieu, une description, une date de début et une date de fin. Une fois que l'événement est créé le fragment qui contenait le bouton "vote" entre autre est changé pour un fragment qui permet de choisir si on participe à l'événement, si on ne participe pas à l'événement ou si on participe peut-être à l'événement. On peut aussi voir les utilisateurs ainsi que leur statut de participation à l'événement. Si un utilisateur choisi de participer ou de peut-être participer à l'événement, celui-ci est ajouté dans son calendrier.

Difficultées rencontrées

Configurer Google Maps était extrêmement long et laborieux, d'une part la documentation n'était pas toujours très claire et ne semblait pas très complète pour pouvoir être en mesure de l'utiliser. La courbe d'apprentissage pour utiliser ses fonctions était très élevée aussi.

L'utilisation de Firebase s'est avéré plutôt ardue. En effet puisque les requêtes à cette base de donnée sont faites de manière asynchrone il fallait que les traitements sur les données de la base de données soient faits dans les requête parce que sinon il se pourrait que l'affichage se fasse avant que la requête ait les données nécessaire.

Suggestions

Pour un travail pratique la charge de travail était plutôt élevée, il aurait peut-être fallu diminuer le nombre de requis ou en faire un projet de session au lieu d'un travail pratique.

Conclusion

En conclusion, ce travail pratique nous a permis d'enrichir énormément nos connaissances en développement Android. Il nous a fallu apprendre à utiliser la caméra ainsi que la puce gps de la tablette. Nous avons aussi eu à apprendre à travailler avec une base de données distante avec une tablette et de faire des requêtes qui nous permette de sauvegarder ou d'aller chercher les informations nécessaires. Le tout nous a permis de faire une application dynamique qui communique d'une tablette au moyen d'une base de données.