

INF8405 – Informatique mobile

Travail Pratique N°2 – Application d'organisation d'événements

1. Informations générales

Session	Hiver 2017
Auteur	Fabien BERQUEZ
Chargé de laboratoire	Fabien BERQUEZ (fabien.berquez@polymtl.ca)
Date de début	16 février 2017
Date de remise	18 mars 2017 à 23h 55min au plus tard
Laboratoire	L-4708
Pondération	15%
Directives particulières	<ol style="list-style-type: none"> 1. Tout projet rendu sera pénalisé de 5 points s'il est soumis par une équipe dont la taille est différente de celle définie (3), sans l'approbation préalable du chargé de laboratoire. 2. Rapport à rendre en format PDF ou Word. 3. Le code source + le rapport sont à rendre dans une archive ZIP, avec l'APK de l'application 4. Soumission du rapport et des travaux réalisés par moodle uniquement (http://moodle.polymtl.ca). 5. Tout retard de soumission du rapport et/ou des travaux réalisés sera pénalisé de 5 points par jour de retard.

2. Connaissances préalables

- Langage Java (programmation orientée objet, développement d'interface graphique, programmation événementielle).
- Langage XML.
- Utilisation de l'API Android
- Notions de base sur les réseaux sans fils (WiFi, cellulaire)
- Notions de base sur les capteurs sous Android
- Utilisation des APIs de Google (particulièrement Google Maps)

3. Environnement et outils nécessaires

- Environnement de développement Intégré (IDE) : Android Studio
- Compte Google (Clé Google Maps API)
- Tablette/Cellulaire Android ou Emulateur

4. Objectifs du laboratoire

L'objectif principal de ce laboratoire est approfondir l'apprentissage du développement de services pour des terminaux mobiles utilisant le système d'exploitation Android. Lors du laboratoire précédent, le développement était axé sur la structure de base d'une application Android (activités, communication entre elles, événements, ressources, ...), et sur les possibilités graphiques. Les fonctionnalités réseau, très importantes pour les terminaux mobiles, ainsi que les capteurs n'étaient pas pris en compte. Dans ce laboratoire, il s'agira de se familiariser :

- avec l'utilisation des capteurs sous Android ;
- avec les fonctionnalités réseau et leur gestion sous Android ;
- avec l'utilisation de bases de données en ligne et locales.

5. Présentation de l'application

Dans le cadre de ses activités, un groupe de personnes doit tenir des rencontres. Afin de faciliter l'organisation de ces rencontres, le groupe décide d'utiliser une application mobile. Celle-ci permet à un des membres de groupe, l'organisateur, de proposer des lieux de rendez-vous, de choisir la date, de préciser les modalités. Les membres du groupe ont ainsi la possibilité de noter les lieux proposés par l'organisateur, de consulter la position des autres membres du groupe, de créer dans leur calendrier l'événement correspondant, d'inviter des personnes de leur liste de contact. Une fois les notes de tous les utilisateurs validées, l'organisateur peut voir la note moyenne attribuée à chaque choix. Il peut alors créer l'événement en fixant une date de début et une date de fin, ainsi que le lieu. Les autres membres du groupe peuvent alors voir ces informations et préciser s'ils viennent ou non à l'événement.

6. Requis

6.1. Requis fonctionnels

Les fonctionnalités attendues de l'application sont les suivantes :

- ✓ Au lancement de l'application, l'utilisateur rentre un pseudonyme et prend une photo de profil (vous **devez** utiliser la caméra. La photo de profil **ne** peut **pas** provenir d'un fichier)
- ✓ L'utilisateur rentre le nom du groupe auquel il se joint. Si ce groupe existe déjà, l'utilisateur est ajouté au groupe. Sinon, un nouveau groupe est créé, et l'utilisateur devient l'organisateur.
- ✓ Affichage d'une carte Google Maps indiquant la position des membres du groupe (mise à jour régulièrement selon les dernières informations disponibles)
- ✓ Lorsque l'application s'exécute, la localisation de l'utilisateur est envoyée périodiquement au serveur, pour pouvoir être diffusée aux autres utilisateurs
- ✓ L'organisateur peut, sur la carte Google Maps, appuyer longtemps sur une position. Dans ce cas, un marqueur s'ajoute à cet endroit. L'organisateur peut alors rentrer un nom et éventuellement une photo pour le lieu. Une fois l'opération terminée, l'organisateur doit valider pour que les coordonnées du lieu ainsi que le nom et la photo soient envoyées au serveur.
- ✓ L'organisateur peut indiquer jusqu'à 3 lieux sur la carte. Une fois les 3 lieux choisis, ils sont transmis aux autres membres du groupe.
- ✓ Les autres membres du groupe peuvent consulter sur la carte les lieux proposés par l'organisateur et les informations (nom et photo) relatives à ces lieux. Pensez à bien distinguer les marqueurs désignant des personnes de ceux désignant les lieux.
- ✓ Chaque membre du groupe (y compris l'organisateur) peut noter chacun des lieux sur une échelle de 1 à 5. Une fois que l'utilisateur a choisi sa note pour les 3 lieux, il peut valider l'envoi de la note sur le serveur. Les notes ne peuvent plus être modifiées une fois envoyées.
- ✓ Les membres ayant déjà voté peuvent consulter la moyenne actuelle de chacun des lieux.
- ✓ Une fois que tous les membres ont envoyé leurs notes, l'organisateur est informé des moyennes définitives des 3 lieux. Il peut alors choisir l'un des trois lieux (généralement celui avec la meilleure moyenne) pour créer l'événement.
- ✓ L'organisateur crée alors l'événement, avec un nom, des informations (i.e. un texte libre), une date et heure de début et une date et heure de fin.

- ✓ Les autres membres du groupe voient alors l'événement et les données entrées par l'organisateur, le lieu retenu est mis en évidence sur la carte. Chaque membre dispose de trois options de réponse : Participe, Ne participe pas, Participe peut-être.
- ✓ Tous les membres du groupe peuvent voir la réponse des autres.
- ✓ Si l'utilisateur choisit Participe ou Participe peut-être, l'application lui propose d'enregistrer l'événement dans son calendrier.
- ✓ Tout au long du processus, l'utilisateur doit pouvoir choisir, dans les préférences de l'application, le délai entre deux envois de la localisation. Il doit également pouvoir quitter le groupe (après demande de confirmation)
- ✓ Lorsque le niveau de batterie devient faible, l'application doit proposer de passer en mode économie d'énergie : l'intervalle de temps entre deux envois de la localisation est alors fortement augmenté.

6.2.Requis non fonctionnels

Les exigences non fonctionnelles attendues sont les suivantes :

- ✓ L'application sera développée pour une plate-forme Android. À des fins de compatibilité, la version de la plate-forme à considérer est **Android 4.3.1 (API 18)** et versions supérieures. Le minSdkVersion est donc **Android 4.3.1 (API 18)**.
- ✓ Le profil de chaque utilisateur doit être stocké dans son propre terminal ainsi que répliqué sur un serveur.
- ✓ L'application doit supporter les interruptions de connexion WiFi, cellulaire ou GPS (i.e. l'application ne doit pas subir de « crash » lors des interruptions)
- ✓ L'application doit être simple d'utilisation. L'interface doit être soignée et relativement intuitive.
- ✓ La production d'un code lisible et suffisamment documenté (commentaires du code) est vivement attendue. Le développement doit se faire en Java avec Android Studio (recommandé) ou Eclipse
- ✓ L'application doit être utilisable sur des terminaux mobiles offrant des services de géolocalisation, ayant accès à Internet et répondant aux exigences indiquées ci-dessus par rapport à la version d'Android. Les tablettes et téléphones intelligents sont donc particulièrement ciblés
- ✓ Les permissions nécessaires pour l'application doivent être gérées sans introduire de bugs.

7. Conseils complémentaires

- L'utilisation des capteurs ainsi que de la géolocalisation nécessite pour l'application d'obtenir des permissions. Pour l'API 22 et en dessous, celles-ci s'obtiennent à l'installation (il faudra parfois les attribuer dans les paramètres des applications lorsque vous testez l'application). Pour l'API 23 et plus, les permissions doivent être contrôlées et demandées dynamiquement en cours d'utilisation.
- Android supporte nativement les bases de données SQLite. Vous pouvez les utiliser pour le stockage local des données (lieux, notes moyennes, profils utilisateurs)
- Pour la base de données en ligne, Firebase est une bonne solution. Cette solution dispose d'une API pour Android et est maintenu par Google, qui est également responsable du développement d'Android. Elle est donc bien intégrée. Firebase permet de plus de stocker également des fichiers (par exemple les photos).
- Il est bien sûr possible d'utiliser d'autres solutions pour la base de données en ligne.
- N'hésitez pas à poser des questions !

8. Livrables

Les livrables à fournir à terme seront regroupés dans une archive (ZIP) dont le nom est formé des numéros de matricules des membres de l'équipe, séparés par le caractère « trait de soulignement » c'est-à-dire ('_'). L'archive contiendra un rapport de TP au format PDF ou Word et le code source de l'application, ainsi que l'APK généré. Le non-respect d'une de ces consignes peut être sujet à

pénalité.

8.1.Rapport de TP

Le rapport doit comporter une introduction, une section consacrée à la présentation technique du travail réalisé, une section consacrée aux difficultés particulières qui ont été rencontrées et une section consacrée aux critiques et suggestions. En outre, une page de couverture est indispensable et doit porter au minimum : le nom ou le logo de l'école ; le numéro et le titre du cours ; la session ; le libellé du TP ; les noms, prénoms et matricules des membres de l'équipe; la mention « Soumis à : **nom et prénom du chargé de laboratoire** » ; la date de soumission.

8.2.Code source de l'application

Afin de faciliter la correction, vous devrez remettre un fichier .zip contenant votre code source, ainsi que l'APK compilé de votre application, et le rapport. Pour être sûrs de ne rien oublier, vous pouvez remettre l'intégralité du dossier de votre projet Android Studio (y compris les fichiers Gradle).

9. Evaluation

La grille d'évaluation est la suivante :

Rubriques	Points
Évaluation de l'exécutable : (fonctionnalité, ergonomie, portabilité, performance)	8
Évaluation du code source (implémentation de l'architecture, logique de programmation, documentation du code)	7
Rapport de TP (fond, forme)	5
Total	20

10. Documentation

- <https://developers.google.com/maps/documentation/android-api/>
- https://developer.android.com/guide/topics/sensors/sensors_overview.html
- <https://developer.android.com/reference/android/hardware/Camera.html>
- <https://firebase.google.com/docs/storage/android/start>
- <https://firebase.google.com/docs/database/android/start/>
- <https://developer.android.com/guide/topics/permissions/index.html>
- <https://developer.android.com/training/location/index.html>