

INF8405 – Informatique Mobile

**TP3 –** Outdoor Fishing

Groupe 01

Cloutier, Frédéric 1633375

Lam, Ba Samson 1671028

Vuong, Sylvester 1635535

Soumis à : Berquez, Fabien

28 Avril 2017

Introduction

Dans les années passées, nous avons constaté une croissance importante dans le développement des applications mobiles. Ce progrès fut marqué par la naissance des téléphones intelligents. Il serait donc intéressant d’étudier sur la technologie des téléphones mobiles. Plus précisément, nous allons nous intéresser dans le développement des applications mobiles.

Dans le cadre de ce projet, nous devons concevoir une application mobile à titre éducatif. Le but du travail est d’aider à l’apprentissage avancé pour les étudiants, notamment dans l’utilisation de service et des réseaux sans fils, le partage de données par NFC et l’utilisation de capteurs.

À cette fin, nous concevons une application de partage d’informations de pêches. L’application permet à l’utilisateur de prendre en photo une prise de poissons lors d’une activité de pêche. À ce moment, la localisation de l’appareil ainsi que plusieurs données provenant des capteurs sont enregistrées. La photo et les données sont ensuite stockées dans une base de données sur *Firebase*. L’utilisateur peu visionner les entrées faites par les autres utilisateurs via une carte *Google Maps* sous forme de marqueurs situées à la localisation de l’appareil au moment de la prise de données. ADD STUFF PHOTOS,NFC,SHARE SHIT, SCROLLING VIEW

Pour réaliser ce projet, il faut d’abord conceptualiser les différentes activités de l’application. Donc, ce rapport décrit tous les aspects techniques reliés à la conception, énumère les difficultés rencontrées lors de celle-ci. Enfin, des critiques et suggestions sont proposées.

Présentation technique

**Base de données**

En premier temps, toutes les données et les photos pertinentes à l’application sont stocké dans une base de données *Firebase*. Voici à quoi ressemble la structure des données :

PHOTO FIREBASE

Fig. 1 : Structure des données sur *Firebase*

~~On retrouve à la racine une liste des différents groupes. Sous chacune d’elles, on retrouve le profil de l’organisateur (~~*~~m\_manager~~*~~), les informations de lieu et de temps reliés à l’évènement (~~*~~meeting~~*~~), une liste des différents lieux proposés par l’organisateur (~~*~~placeList~~*~~), et une liste des utilisateurs appartenant au groupe (~~*~~usersList~~*~~). Pour plus de détails, accéder directement à la base de donnée publique via :~~ [~~https://tp2firebaseproject-e31e0.firebaseio.com/~~](https://tp2firebaseproject-e31e0.firebaseio.com/)

Les photos, quant à elles, sont simplement stockées dans un module *Storage* sur *Firebase*.

**Structure du projet**

Il y a 8 classes d’activité, soit *AppCompatPreferenceActivity, BatteryActivity, BeamActivity, FishActivity, GalleryActivity, MainActivity, MapsActivity* et *SettingsActivity*. Chacune est décrite plus en détail dans les sections qui suivent. Ensuite, les POJOs utilisé pour la modélisation de la problématique sont *EnvSensorEntry, FishEntry, FishDTO, Group*. Finalement, la classe *Singleton* se charge principalement de toute interactions avec la base de données *Firebase* et *BatteryReceiver* se charge de recevoir le pourcentage de batterie restante de l’appareil.

Du coté des ressources, les deux fichiers à souligner sont *strings.xml* et *pref\_general.xml*. Le premier contient tous les valeurs constantes de chaînes de caractères utilisés à travers l’application et le deuxième sert à sauvegarder le nom de l’utilisateur.

**MainActivity**

L’activité d’entrée est *MainActivity*. Lors de sa création, elle instancie un objet singleton *Singleton*. Ce dernier se charge de toutes les interactions avec la base de données *Firebase* et est utilisé principalement dans les activités *Maps*Activity, *FishActivity*, *BeamActivity* et *SettingsActivity*. De plus, elle donne accès à tous les autres activités, notamment les suivants : Settings, Map, Gallery, Fish, Battery et NFC Beam.

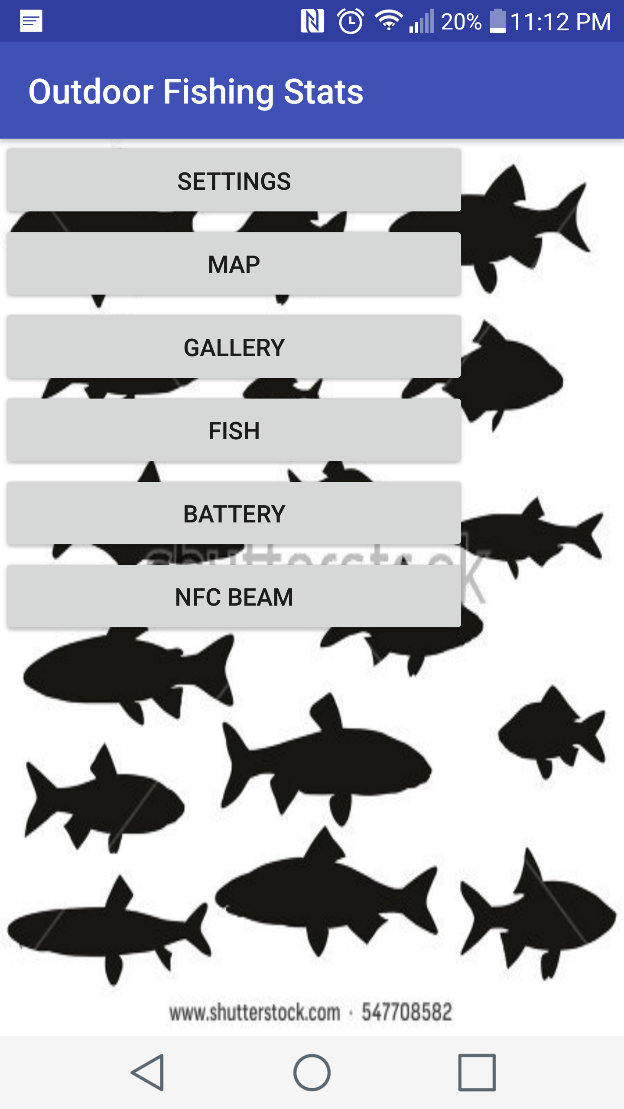


Fig. 3 : Vue de *MainActivity*

**SettingsActivity**

Dans cette activité, l’usager peut modifier son nom d’usager. Ce nom en fait est affiché dans les marqueurs du Google Maps. De plus, ce nom devrait être choisi judicieusement puisque tous les prises de photo de vos captures seront identifiées avec votre nom. Au moment d’une faille de sécurité, vous risquez de perdre accès à ces derniers si vous ne vous rappelez pas de votre nom.

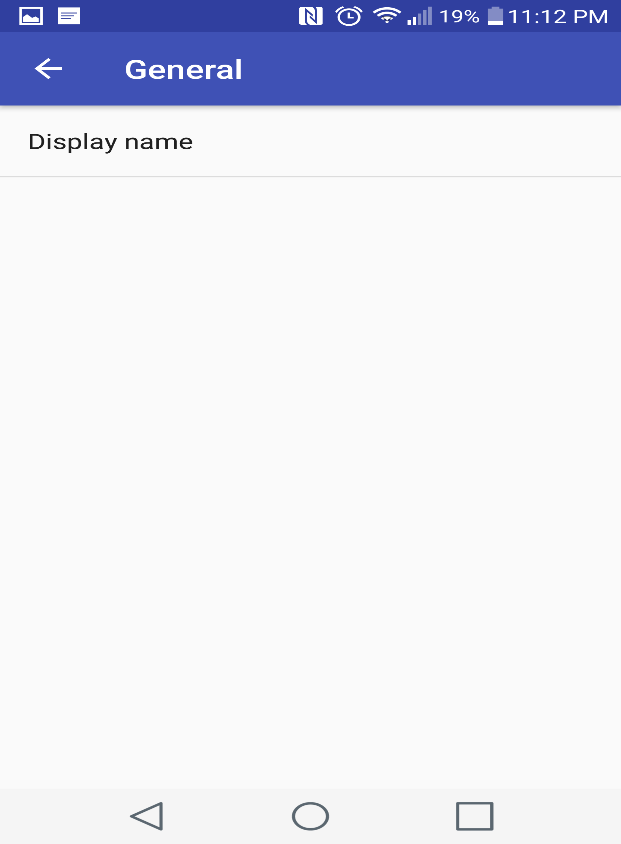


Fig. 4 : Vue de *SettingsActivity*

**MapActivity**

Cette activité permet d’accéder au Google Maps. Elle donne aussi accès à tous les usagers des informations cruciales de l’application, soit celle des emplacements des lieux de pêches. Toutefois, ces marqueurs de lieu révèlent que le nom et le contact d’un usager (celui qui a posé ce marqueur). Pour obtenir plus d’information, l’usager doit contacter ce dernier pour avoir plus d’information telles que la température, les types de poisson, une photo et etc. En fait, pour simplifier l’utilisation de l’application, nous procédons ici par NFC. Lorsqu’un usager contacte le créateur du marqueur et que ces derniers effectuent un échange via NFC, l’usager reçoit une clé d’accès pour débloquer toutes les informations nécessaires en ce qui concerne la pêche. Il est à noter que la permission est requise dans certain des appareils selon sa version.

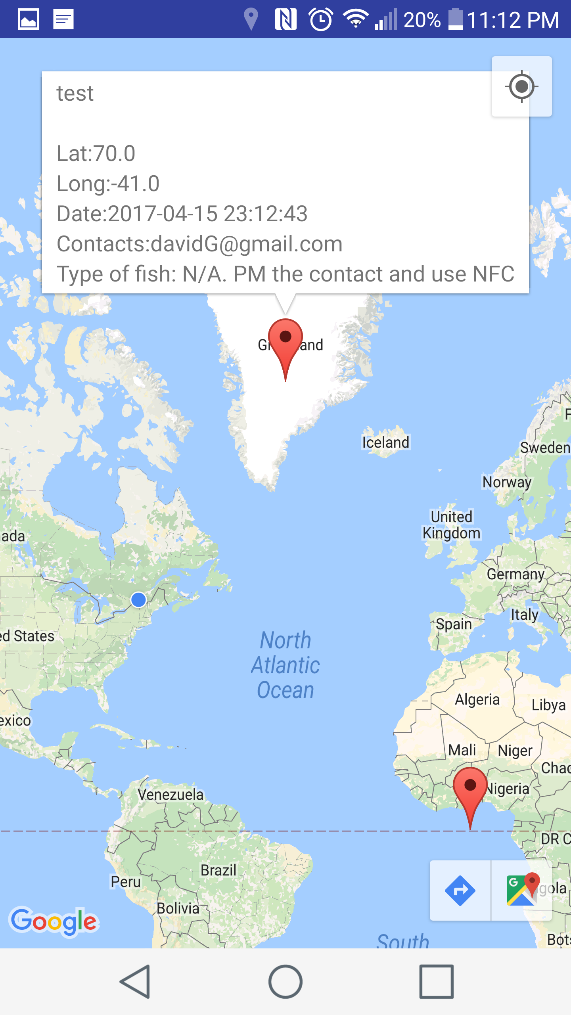


Fig. 5 : Google Maps

**GalleryActivity**

**//TODO FRED ou SYLV**

**FishActivity**

**//TODO SYLV**

**BatteryActivity**

Cette activité est accessible à partir du menu principal. Elle affiche le pourcentage de batteries restant de l’appareil et donne l’option à l’utilisateur de passer en mode économie d’énergie. Cela permet de réduire le nombre de lecture des senseurs. De plus, cette activité renvoie une information pertinente pour ceux et celles qui sont *énergiephile* : l’utilisation de la batterie depuis le lancement.

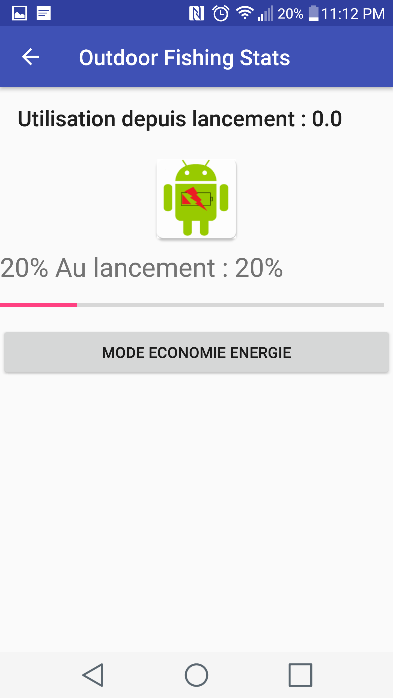


Fig. 9 : BatteryActivity

**BeamActivity**

Cette activité permet l’échange d’information. Elle est très simpliste; elle nécessite que deux usagers rapprochent leur appareil en mode beam (NFC activé bien sûre) et que les deux échangent leur information. Cette information permet donc de débloquer plus d’informations pour accéder à plus de données dans la base de données locale et distante.



Fig. 10 : BeamActivity

Difficultés rencontrées

L’une des difficultés rencontrées était la modélisation de la problématique. Nous devons revoir à répétition toutes nos solutions et notre objectif du projet. Nous avons changé plusieurs fois nos SRS en raison de manque de spécification et de clarté et ce, à l’interne.

Une autre difficulté réside dans le matériel disponible. Tous nos appareils ne supportent pas certains des senseurs souhaitables (température, etc.). Nous devons nous tenter d’implémenter ces lacunes presqu’aveuglement, mais avec beaucoup de soins.

Une dernière difficulté est la gestion de temps avec nos travaux scolaires. Nous avons donc beaucoup conflit d’intérêt en ce qui concerne ce projet et la qualité est donc affecté gravement. Le temps alloué nous semble un peu restreint et il rentre en conflit avec nos temps d’étude pour les examens finaux.

Critiques et suggestions

Il serait intéressant d’allouer beaucoup plus de temps pour se travail. Selon nous (et c’est notre humble avis personnel), nous trouvons que le temps alloué entre dans le temps d’étude pour les examens finaux. Cela, nous force à prioriser nos objectifs. Il sera donc intéressant de diminuer la charge du travail ou bien d’allouer plus de temps. Dans la seconde solution, nous suggérons de diminuer la proportion de la valeur du projet.

Sinon, ce projet nous a permis d’assembler toutes nos notions acquises dans ce cours. Dans un premier temps, nous avons appliqué les notions de base en programmation d’Androïde. Dans un deuxième temps, nous avons usé les APIs populaires telles que Google Maps. Dans un dernier temps, nous avons utilisé des capteurs de nos appareils et nous avons fait bon usage de ces derniers