

Département de génie informatique et de génie logiciel

**INF8405 - Informatique Mobile**

**Laboratoire #2 : Application réseau pour Android**

**Sacha Licatese-Roussel, 1635849**

**Yann Dago, 1635682**

**Golnoush Rahimzadeh, 1759612**

**Travail soumis à Aurel Josias Randolph**

**Remis en date du**

**23 mars 2016**

# **Sommaire**

[Sommaire 2](#_Toc443136494)

[Introduction 3](#_Toc443136495)

[Présentation technique du travail 3](#_Toc443136496)

[Aspect de présentation de l’application 4](#_Toc443136497)

[Logique d’avancement au sein du jeu 5](#_Toc443136498)

[Logique & construction de la grille de jeu 5](#_Toc443136499)

[Difficultés rencontrées 7](#_Toc443136500)

[Critiques et suggestions 7](#_Toc443136501)

[Conclusion 7](#_Toc443136502)

# Introduction

**Dans ce premier travail pratique, il a fallu effectuer le développement d’une application de prise de rendez-vous par l'entremise du logiciel Android Studio. Ainsi, nous avons dû nous familiariser avec différents modules à intégrer afin de réaliser cette application. Une base de données, l’utilisation de “Google Place”, “Google Maps” et “Google Calendar” a permis de générer des rendez-vous et les partager entre les personnes concernées.**

# Présentation technique du travail

Dans cette première partie, nous présenterons les différents aspects relatifs à la conception des différentes parties de l’application et la manière selon laquelle nous avons abordé chacun de ces aspects techniques.

Les différents aspects techniques requis pour ce travail sont les suivants :

* Utilisation d’une base de données pour gérer les groupes de personnes ainsi que les profils des utilisateurs voulant organiser des rendez-vous.
* Utilisation de **“Google Places” afin de proposer différents lieux de rendez-vous.**
* Utilisation de **“Google Maps” afin de montrer la dernière position connue de chacun des membres d’un groupe**
* **Utilisation de “Google Calendar” afin de trouver des journées de disponibilités communes et générer un événement dans le calendrier une fois le rendez-vous officialisé.**

## Aspect de présentation de l’application

L’application présente trois pages principales qui sont les suivantes :

|  |  |
| --- | --- |
| Figure 1 : Page principale | Figure 2 : Page de création d’événements (événement en cours de vote) |
| C:\Users\yannd\AppData\Local\Microsoft\Windows\INetCache\Content.Word\12873388_10208865269455157_1155565763_o.jpg  Figure 3 : Page de création d’événements (événement en cours de vote) | C:\Users\yannd\AppData\Local\Microsoft\Windows\INetCache\Content.Word\12874254_10208865270295178_377843044_o.jpg  Figure 4 : Page de modification du profil utilisateur |

* La page principale :

C’est la page d’entrée à l’application, elle permet de se connecter (**“Login”), avec un compte existant ou de s’enregistrer (“Signin”).**

Des rétroactions sont faites à l’utilisateur pour l’informer que son profil n’existe pas, ou que le groupe auquel il veut s’enregistrer possède déjà un organisateur, ou s’il lui manque des données afin de se connecter ou s’enregistrer.

* La page de création d’événements :

Cette page présente à l’utilisateur propose à l’utilisateur de créer des événements et de donner une description et une photo à des événements finalisés.

Tous les utilisateurs peuvent voir les différents rendez-vous, avec trois boutons de lieux s’il faut voter, ou un seul endroit si l’événement est finalisé.

Il est aussi possible de voir la position de tous les membres du groupe sur la carte, un marqueur rouge sur lequel le nom du membre apparait lorsqu’on clique dessus.

* La page de modification d’un profil utilisateur :

Cette page permet à l’utilisateur de modifier sa photo, et ses préférences.

## Gestion de la base de données d’utilisateurs

**“Firebase” a été utilisé afin de de gérer la banque de groupes et d’utilisateurs, ainsi la synchronisation entre le téléphone et la base de données est gérée par ce cadriciel. Nous y avons sauvegardé une classe *Group*** contenant des *UserProfiles,* ainsi, pour se connecter, l’utilisateur doit fournir la bonne combinaison de groupe et nom d’utilisateur. Pour créer un nouveau groupe, il doit être organisateur. Les *MeetingEvents* sont aussi sauvegardés dans la classe groupe.

## Proposition de lieux et dates

En ce qui a trait à la proposition de lieux, la position qui est au centre de tous les membres du groupe est jugée idéale pour un rendez-vous. Ainsi, grâce à Google Place API, une requête http est faite afin de trouver trois lieux les plus proches de ce points qui respectent le plus possible les préférences de lieu des utilisateurs. Toutes les préférences sont considérées également.

Pour le choix de la date, les disponibilités sont en fonction des jours durant lesquels il n’y a aucun événement dans le calendrier d’une personne. Ensuite, pendant les jours durant lesquels il n’y a aucun événement, le rendez-vous est créé à une certaine heure pour tous. Il n’y a pas de vote pour le temps, seulement pour le lieu.

## Affichage des positions des membres sur une carte

Afin d’afficher la position des membres sur une carte, Google Maps a été utilisé. En utilisant les services de Localisation du téléphone, la position du membre est sauvegardée dans son profile sur la base de données. Tous les membres d’un groupe ont accès à cette position qui est affichée sur la carte comme un marqueur. I suffit de cliquer sur ce marqueur pour avoir le nom du membre, et le nom du groupe.

# Difficultés rencontrées

Les principales difficultés furent l’utilisation des bonnes clés pour les API de Googles, puisqu’il en fallait plus qu’une. Comprendre comment fonctionnaient les API de Google en général, car ils sont assez complexes, mais aussi la synchronisation pour toutes nos tâches asynchrones. L’architecture a aussi été une difficulté puisqu’il y a beaucoup d’éléments qui doivent être connectés.

# Critiques et suggestions

# Conclusion