

**UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE**

**IFT 604
Applications Internet et mobilité**

Projet

**Travail présenté à
M. Armel Ayimdji Tekemetieu**

**Par
Raphael Azou - azor1201
Elorn Kerros - kere0801
Michael Goulet - goum1403
Olivier Tremblay-Turcotte - treo2702**

13 décembre 2022

Table des matières

Introduction 3

Choix de l’application..... 3

Architecture 3

 Serveur 3

 Client 4

Conclusion..... 5

Références 5

Introduction

Pour la réalisation du projet, nous avons choisi de faire une application de type chasse au trésor (*scavenger hunt*). Les explications de ce choix seront suivies par une description de l'architecture du projet du côté du serveur et du côté du client.

Choix de l'application

Le point de départ du choix de l'application est l'obligation d'utiliser la géolocalisation. Le type chasse au trésor répondait bien à cette contrainte. Le fonctionnement de l'application se résume comme suit, des points de géolocalisation sont répartis sur une carte, et chaque point contient une énigme et une réponse associée. Pour pouvoir accéder à la question, il faut être dans un rayon proche du point voulu. Lorsqu'une bonne réponse est donnée, l'utilisateur gagne des points. L'objectif derrière ce concept est de pouvoir répondre à certains besoins en termes d'exploration. C'est-à-dire, l'application pourrait être utilisée pour organiser la visite d'un endroit et impliquer les visiteurs en leur permettant de répondre à des questions en lien ce qu'ils visitent. L'application pourrait aussi servir à explorer des parcours où les réponses sont disponibles sur place, dans un but de compétition ou tout simplement par plaisir.

Architecture

Serveur

NodeJs et ExpressJs sont utilisés pour le serveur. Notre serveur est inspiré de celui utilisé dans les travaux pratiques du cours. Trois modèles d'objets sont utilisés. Le premier est la classe User composée d'un nom d'utilisateur et d'un score. La deuxième classe est Repere qui correspond aux coordonnées géographiques et qui sont composées d'un identifiant unique, d'une latitude, d'une longitude et d'un objet Texte. La troisième classe est la classe Texte qui est composée d'un texte et d'une réponse.

Les données sont conservées dans des fichiers json qui se trouvent le répertoire json. User.json contient les informations des utilisateurs, et reperi.json contient les informations sur les points.

Les routes disponibles sont les suivantes :

- /user
 - Méthode : POST
 - Message JSON :
 - { username : nom de l'utilisateur }

- Description : Permet à l'utilisateur de s'authentifier. Dans le cas où le nom d'utilisateur n'existe pas, le compte est créé avec un score à 0.
- /user/ :user
 - Méthode : GET
 - Description : Retourne le score de l'utilisateur spécifié.
- /user/ :user
 - Méthode : PUT
 - Message JSON : { score : nouveau score }
 - Description : Met à jour le score de l'utilisateur spécifié.
- /repere
 - Méthode : GET
 - Description : Retourne la liste des coordonnées où il y a une énigme.
- /repere
 - Méthode : POST
 - Message JSON : { latitude : point, longitude : point, texte : { texte : text, reponse : reponse } }
 - Description : Permet de créer un nouveau point et y associer une question et une réponse.
- /repere/ :id
 - Méthode : GET
 - Description : Permet de récupérer la question d'un point
- /repere/ :id
 - Méthode : POST
 - Message JSON : { reponse : reponse }
 - Description : Permet d'envoyer la réponse associée au point spécifié.

Client

Le client Android utilise retrofit pour communiquer par requête avec le serveur. Des modèles sont utilisés pour récupérer les informations des coordonnées (CacheList), les informations d'un point (Cache), les informations d'un texte (Texte) et les informations de l'utilisateur (User).

Le client est composé de 4 activités. La première est LoginActivity qui correspond à la page de connexion où il est possible de saisir un nom d'utilisateur. MainActivity correspond à la page principale de l'application où il est possible de voir la carte et les points disponibles via l'API de Google. À partir de cette activité, CachesActivity est accessible. Cette activité permet de récupérer les informations d'un point près de nous et d'y envoyer notre réponse. L'activité UserActivity est

accessible à partir de MainActivity et correspond à la page de profile de l'utilisateur où son score est affiché.

Conclusion

Pour conclure, ce projet a permis de faire la conception et d'implémenter un système client-serveur complet pour intégrer la matière apprise en cours.

Références

Google Maps Platform Documentation | Maps SDK for Android. (2022, February 9). Google Developers. <https://developers.google.com/maps/documentation/android-sdk>