## Itération 2 : Description des technologies

**Projet Jormun**

**Tables des Matières :**

1. **Introduction**
2. **Présentation des technologies côté clients**
3. **Présentation des technologies côté serveur**

**Introduction.**

Notre projet requiert la connaissance en temps réel de la position géographique de chaque utilisateur ainsi que d’offrir un stockage en lignes des informations relatives aux objets partagés entre les joueurs tel que les boss, pour certaines quêtes ainsi que pour les villes qui peuvent êtres visitées par les autres joueurs.

En raison de ces caractéristiques, le projet possède une partie serveur et une partie client.

La partie client concerne l’application mobile en elle-même, et elle possède des rôles précis comme la récupération de la position du joueur en Latitude et Longitude mais aussi le stockage temporaire des informations dans une base de données locale afin de limiter le transfert de données entre le serveur et le client lors des interactions avec l’application. En effet, il n’ait pu être pas requis d’envoyer des informations sur les héros du joueur, à la base de données du serveur de manière fréquentes contrairement à améliorations d’une ville ou bâtiments d’un village.

La partie serveur, est divisé en deux partie, la première offre une communication en continue de la positions de chaque joueur dans le monde ( avec une vérifications afin d ‘éviter d’envoyer des informations inutiles au joueurs). La deuxième partie concerne plus le stockage des informations de manière permanente et ainsi offrir une centralisation des informations pour les utilisateurs.

**Présentation des technologies côté clients**

Cette partie utilise l’API de google maps qui nous permet d’avoir l’environnement principale du jeu, sur lequel nous pourront ajouter des éléments avec lesquels l’utilisateur pourra interagir afin de faire progresser son jeu.

La base de données locale permet de garder en mémoire des informations relatives aux éléments autour du joueur ou du joueur lui-même.

La base de données locale est géré par le système relationnel SQLite qui à l’avantage d’offrir la possibilité de lié des fichiers JSON à des éléments du jeux et de pouvoir rapidement les retrouver, modifier ou les supprimer en fonction des besoins de l’application.

**Présentation des technologies côté serveur**

Du coté serveur, nous avons donc deux serveurs avec des rôles spécifiques.

Le premier serveur est un serveur java qui permet de communiquer en temps réel les positions entre les joueurs. Lors du lancement de l’application et après l’enregistrement de l’utilisateur, le système se connectera à ce serveur et communiquera la position du joueur mais aussi les limites de la vue du joueur, ce qui nous permet de savoir ce qu’il voit sur l’écran et donc de lui envoyer la position des utilisateurs autour de lui. Il offre l’avantage de ne pas stocker des informations superflu, en effet, nous sommes intéresser par les positions du joueurs seulement qu’en temps réel et donc jugeons que stocker ces informations dans une base de données, serait inefficace et encombrant. A l’inverse, l’inconvénient est que c’est un serveur en plus à gérer et qu’il faut gérer la charge des utilisateurs sur ce serveur. Il implique aussi une bonne gestions des erreurs notamment si un utilisateur rencontre une panne, que sa déconnexion soit gérer de manière souple et sans engendrer d’erreur pour les autres utilisateurs.

Nous sommes en train de remettre en forme la structure du serveur Jormun via l’implémentation de ‘design patterns’ et la réduction de dépendance, notamment afin d’amélioré la qualité du code mais aussi de facilité la modification du code.

Le second serveur se charge de la gestion des données du jeu, soit la sauvegarde des données du joueur sur une base centralisé mais aussi la gestion des éléments du jeu qui ne sont pas contrôler par les joueurs comme les boss et les zones à ressources. Ceci permet de garder une trace des informations du joueur même après la suppression de l’application. Il repose sur les même avantages que le stockage en locale sur la machine du client, en permettant de rapidement récupérer tous les fichiers d’un même utilisateur/élément du jeu.

En raison de sécurité, nous avons décidé de limiter les ports ouverts et de transformer le serveur PHP en un serveur ASP.NET.

**Reférences :**

Pour connaître les technologies