

EPITECH

PFA 2017

---

# WHERE TO MEET

---

*Auteurs:*  
mendie\_j  
vuilla\_l

January 28, 2015

# 1 Présentation du projet

Where to meet sera une application mobile (Android) visant à fournir à ses utilisateurs un moyen de se rencontrer autour d'une activité particulière un moment T. Autrement dit, where to meet permettra de créer des liens sociaux entre des individus qui pratiquent la même activité au même moment et souhaitent partager cette activité avec d'autres. L'idée est de relier des gens qui ont quelque chose en commun, quelles que soient leurs hobbies ou leurs envies. Pour faciliter les rencontres, les recherches se font grâce à des tags<sup>1</sup>.

Voici un cas typique auquel where to meet propose une réponse : Vous êtes guitariste et vous aimeriez jammer, mais vos amis musiciens ne sont pas disponibles pour le moment. Vous lancez alors Where to meet pour rechercher une session de jam en cours. À dix minutes de chez vous, un groupe de percussionnistes s'installe et cherche d'autres musiciens. Eux aussi lancent where to meet et remplissent le formulaire d'ajout d'activité. En un instant, le groupe est enregistré et apparaît sur votre recherche. Vous avez alors plus qu'à vous rendre sur place avec votre instrument pour rencontrer vos nouveaux amis.

## 2 Description Fonctionnelle

L'utilisateur pourra se connecter en utilisant un compte qu'il aura créé au préalable ou bien se connecter via les comptes qu'il utilise habituellement sur les réseaux sociaux. Une fois connecté, l'utilisateur arrivera sur une page présentant le logo de l'application, un menu latéral, une barre de recherche, une carte et un bouton d'ajout d'événement.

### 2.1 Le logo

Préférant nous concentrer sur ce que nous faisons de mieux - la programmation - nous sous-traiterons la création du logo à un designer pour un résultat professionnel, marquant et adapté à l'esprit de where to meet. Nous nous en occuperons peut-être nous-même si besoin est, mais à la fin du projet.

---

<sup>1</sup>Les tags sont des mots-clés permettant de résumer les choses de manière concise.

## 2.2 Le menu latéral

Il présentera plusieurs lien vers:

- le profil de l'utilisateur que ce dernier pourra personnaliser
- l'historique des événements grâce auquel il sera possible d'évaluer les différents participants à chaque événement. La notation ne se fera pas en attribuant une note mais en renseignant un maximum de 5 tags sur la personne en question. Par exemple : "Alcoolique", "égoïste" ou alors "gentil" sont des tags valides.
- Une boîte à idée pour proposer des améliorations aux développeurs.

Il sera possible de masquer le menu latéral afin d'aérer l'application.

## 2.3 La barre de recherche

C'est ici que l'utilisateur entre les tags qui l'intéressent. Il sera possible d'utiliser une liste de filtres afin d'affiner la recherche : le rayon de la recherche, exclusion de certains tags associés aux utilisateurs présents. Cette exclusion se fera par une liste déroulante.

## 2.4 La carte

Dès la connexion de l'utilisateur à l'application, la carte affichera les événements en cours à proximité. Mais après une recherche, elle n'affichera plus que les résultats pertinents, c'est-à-dire ceux qui intéressent l'utilisateur. Il sera alors possible de cliquer sur un événement particulier pour en afficher les détails.

## 2.5 Le bouton d'ajout d'évènement

Il fera apparaître un formulaire comportant quelques champs essentiels la description de l'activité que l'utilisateur souhaite lancer. En plus d'une description textuelle du lieu de rassemblement et de ses objectifs, l'utilisateur devra rajouter une liste de tags afin de fournir une description plus succincte. Par exemple : une partie de foot. Nous recherchons deux joueurs pour rendre le jeu plus intéressant. tags: "zinedine\_zidane" "ballon" "goal"

## 2.6 Le Serveur

Le back-end se présente sous la forme d'un web-service restfull hébergé sur un raspberry-pi le temps du projet. une requête est effectuée par le

biais d'une url. Cette dernière est traitée par le web-service qui récupère les données pertinentes dans la DB et les renvoie serialisées au client

Notre serveur héberge également la base de donnée SQL, qui contiendra les informations nécessaires au bon fonctionnement du projet (mails, passwords, coordonnées géographiques des événements etc...)

## 3 Description Technique

Concernant les technologies utilisées, nous allons développer l'application en java. Nous avons décidé de développer en natif afin d'assurer les meilleures performances nos utilisateurs. Nous aurons un serveur pour héberger la base de données et développerons un web service afin d'abstraire la communication avec cette dernière. Ce web service sera écrit en perl grâce au framework MVC<sup>2</sup> (Model View Controller) Mojolicious. Nous avons choisi d'utiliser perl parce qu'il est simple, malléable et permet de développer très rapidement, ce qui nous permettra d'entrer sur le marché prestement et efficacement. Quant au framework Mojolicious, le modèle MVC nous permettra de faciliter la mise en place du web service.

### 3.1 Contraintes

Nous ferons sans doute face à quelques problèmes, comme la découverte d'un nouveaux langages, le Java. Cependant, nous ne pensons pas rencontrer de problèmes pouvant provoquer un blocage du projet.

## 4 Dérroulement du projet

equipe:

- Vuilla\_l: Application mobile / Site vitrine
- Mendie\_j: Web-Service / Application / Site Vitrine

### 4.1 Planning

Nous attribuerons les tâches en fonction des besoins qui émergeront lors du développement. Ceci dit, nous nous sommes mis d'accord pour suivre le plan d'action suivant : L'application et le web service seront développés en même temps dans le souci d'une avancée concomitante et efficace sur le projet. Dès que le web-service sera terminé, Mr.Mendiela rejoindra Mr.Vuillaume sur le développement de l'application mobile. La présence d'un troisième membre qui servirait d'électron libre est souhaitée.

---

<sup>2</sup>Le patron MVC est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.