# Décision technique

## Plan de la base de données

{

"groups" : {

"Groupe" : {

"events" : {

"description" : {

"coordinate" : {

"latitude" : 0,

"longitude" : 0

},

"dateEnd" : "string",

"dateStart" : "string",

"eventName" : "string",

"going" : {"id" : "string",...},

"maybe" : {"id" : "string",...},

"notgoing" : {"id" : "string",...},

"picture" : "string"

}

},

**"lieux" : {**

**"string":{**

"coordinate" : {

"latitude" : 0,

"longitude" : 0

}

"pictureURI" : "string",

"name" : "string"

}

},

"users" : {

"string" : {

"coordinate" : {

"latitude" : 0,

"longitude" : 0

},

"organisateur" : "false",

"pictureURI" : "string",

"username" : "string"

}

}

}

},

"groupsNames" : {

"id" : "string"

}

}

Pour notre base de données entre les cellulaires, nous avons utilisé le système noSQL de Firebase. Nous avons fait ce choix car nos données étaient relativement petites et limité en nombre. De plus, ce système offre un système de listener qui permet de recevoir les update immédiatement, plutôt que notre application les demande explicitement à chaque fois. Voici notre structure de base de données.

À la racine, nous avons groups qui contient les groups et un liste des nom des groups pour pouvoir les identifier sans downloader toute la base de données.

Goups est diviser en plusieurs groupes qui utilisent leurs nom comme id.

Ce groupe contient la liste des lieux, l’évènement et les utilisateurs qui font partis du groupe. Encore une fois, leurs noms est leur ID.

## Activités

Chaque interface a sa propre activité. (est-ce nécéssaire?)

## Services

TODO

-listeners

## Markers

Un aspect intéressant de notre design est comment nous traitons nos markers. Nous avons utilisé l’option de personnaliser les markers de google maps que l’API offre. Par contre, cet API ne permet qu’un design par carte. Alors nous avons mis une option dans le snippet du marker. Selon cette option, nous avons identifié le type de marker (user, location, event) et nous avons utilisé un layout différent dans chaque cas. De plus, les markers de google maps ne permettent pas de mettre de boutons dedans car les markers ne sont que des images. Alors nous avons mis le listener pour les clicks sur tout le marker plutôt que simplement sur le bouon.

# Difficultés

Une des grandes difficultés que nous avons rencontrées est apprendre à utiliser tous les librairies et modules externes différents qui constituent notre application. Ils étaient nombreux et nous n’avions pas utilisé un seul d’eu au paravent.

Une autre difficulté que nous avons rencontrée est connecter et structurer notre base de donnée. Au début de la construction de notre application, nous avons eu plusieurs problèmes de connexion avec firebase. En effet, seulement un de nous réussissions à nous identifier correctement à notre projet.

Enfin la dernière difficulté que nous avons eue était la longueur de l’Application. Nous avons dû mettre un grand nombre de journées de travail pour faire fonctionner correctement toutes les nombres fonctionnalité de l’application donc, nous avions de la difficulté à rentrer cela dans notre horaire, même avec la semaine de relâche.

# Critique et suggestions

Comme il a été mentionné dans la section difficultés, ce TP était très long. Pour diminuer la charge de travail tout en gardant la partie apprentissage du TP, on pourrait simplement couper la partie de lieu et de passer immédiatement à la section évènement.

Un autre solution serait de répartir la charge. De mettre un peu plus de sections du TP2 dans le TP1 (par exemple, un leaderboard pour introduire firebase dans le TP1 pour qu’une partie du temps à passer a l’apprendre soit mieux rependu à travers la session.