

GEOPLAN

Cahier des charges

Et

Spécifications Fonctionnelles Générales

Objet : établir le besoin et les SFG du projet

Contents

I.	Cahier des charges.....	3
a.	Besoin	3
b.	Deadline.....	3
c.	Planning	3
II.	Spécifications Fonctionnelles Générales (SFG)	3
a.	Description du projet.....	3
b.	Fonctionnalités	3
i.	Planning	3
ii.	Géolocalisation	4
iii.	Push position	4
iv.	Radar / Map.....	4
v.	Push message	4
vi.	Deamon	5
c.	Workflow	5
d.	Interface	5
III.	Spécifications Techniques Détaillées (STD)	7
a.	Design	7
b.	Œuvre	7
c.	Architecture, Modules.....	7
i.	Planning	7
ii.	Géolocalisation	7
iii.	Push position	7
iv.	Radar / Map.....	7
v.	Push message	8
vi.	Daemon	8

I. Cahier des charges

a. Besoin

Dans le cadre du cours d'Android du M2 Informatique Logiciel, nous avons la responsabilité de réaliser un projet répondant à certaines conditions :

- Un projet innovant
- Un projet libre

b. Deadline

Mi-Mars.

c. Planning

Environ 280 heures de développement soit avec 6 membres environ 48 heures (2 jours).

Nous disposons d'à peu près 10 semaines avant la mi-Mars, soit environ 5 heures par semaines par membres.

II. Spécifications Fonctionnelles Générales (SFG)

a. Description du projet

Application de planning avec géolocalisation des participants.

b. Fonctionnalités

- 1) Une application de planning (« Google Calendar », « Facebook Event », ou propriétaire)
- 2) Une application de géolocalisation des participants
- 3) Un système de push de position
- 4) Un système de radar ou carte d'affichage de position
- 5) Un système de push de message lié à la position
- 6) Un daemon économique

i. Planning

Utilisation d'un gestionnaire de planning/événement (SEM=Schedule and Event Manager). Le SEM devra permettre de définir des événements/rendez-vous, en prenant en compte les fonctionnalités de bases des solutions déjà existantes, ex :

- **Nom**
- **Date et durée** (date et heure début, date et heure fin)
- **Localisation**

- **Participants** (e-mail, ID user, etc... pour authentifier et joindre le participant)
- **Admin** (voir **Participants**)
- **Description** (détails)
- **Importance**
- **Type** (évènement, Rendez-vous, etc...)
- **Portée** (personnel, groupe)
- **Partage** (Publique, Privée)
- **Rappel** (Temps avant début, répétitions, type (notifications, e-mail, etc...))
- *(Optionnel) Coût*

D'autres options peuvent être déterminées.

ii. Géolocalisation

Lorsque l'application est démarrée et, uniquement, lorsqu'un évènement est commencé (ou sur le point de commencer), le service de géolocalisation se lance. Ce dernier utilisera les fonctionnalités natives d'Android, à savoir, le positionnement « fin » utilisant le GPS, et le positionnement mobile utilisant les réseaux mobiles (3G, 4G, etc...).

iii. Push position

Nous utiliserons un protocole simple et léger pour envoyer les données de position. Il est important de limiter la quantité des données transférée afin d'économiser les données mobiles et d'accélérer la transmission des données. Cette fonctionnalité permettra également de recevoir des positions externes.

Les participants, en utilisant l'application, acceptent de communiquer leur position avec les autres participant de l'évènement. Par défaut, seul l'organisateur de l'évènement reçoit la position des participants, cependant une fonctionnalité permettant de rendre sa position publique à tous les participants est intégrée. Une autre fonctionnalité permet aussi de faire une demande ponctuelle à n'importe quel participant, organisateur ou non. Cette demande doit être validée par le destinataire afin que le destinateur puisse accéder à sa position.

iv. Radar / Map

Il y aura deux moyens d'afficher les coordonnées échangées :

- Le radar : ce dernier affiche le vecteur (distance et direction) de coordonnées entre la source et le receveur. Le receveur se situe toujours au milieu du radar, la source se situe au bout du vecteur. Ainsi nous obtenons de façon précise la distance et la direction entre la source et le receveur (tous les deux des participants). Ce radar est principalement utilisé pour se retrouver dans un lieu avec de l'affluence.
- La carte (map) : cette fonctionnalité utilise la technologie Maps de Google pour afficher la position d'une personne. Elle est utile sur de longue distance.

v. Push message

Cette fonctionnalité permet de communiquer à un ou plusieurs participants de courts messages. Elle propose par défaut des messages types tels que :

- « Je suis en route »
- « Je pars dans X minutes »
- « J'arrive dans X minutes »
- « Je suis coincé dans les bouchons »
- « Il y a des problèmes de transports »
- « Je serais en retard de X minutes »
- Etc...

Le participant aura aussi la possibilité d'entrer un message personnalisé avec une limite de caractères. Cette limite est à déterminer dans les STD après avoir étudié les temps de transmissions de messages de différentes longueurs sur différents réseaux mobiles.

Par défaut il n'y a pas de restrictions sur l'échange de messages entre les participants. Potentiellement sur un broadcast réservé à l'organisateur.

vi. Deamon

« Last but not least » feature : le démon. Ce dernier permet de garder l'application en tâche de fond afin qu'elle check régulièrement les événements et qu'elle active automatiquement les fonctionnalités de localisation et de messagerie lorsqu'un événement démarre. Il devra être léger afin d'économiser l'utilisation de la batterie du mobile.

c. Workflow

Pas déterminé.

d. Interface

L'application devrait être compatible avec un maximum de résolutions et dans les deux sens d'orientation : portrait, paysage. Concernant les résolutions, il n'est peut-être pas nécessaire d'intégrer celles des tablettes, l'application étant destinée à une utilisation mobile. Argument discutable. Voici une liste non exhaustive des différentes résolutions sur les mobiles actuels :

<i>Apple iPhone 6</i>	750	1334
<i>Apple iPhone 5</i>	640	1136
<i>Apple iPhone 4</i>	640	960
<i>Apple iPhone 3</i>	320	480
<i>Apple iPod Touch</i>	640	1136
<i>LG G4</i>	1440	2560
<i>LG G3</i>	1440	2560
<i>LG Optimus G</i>	768	1280
<i>Samsung Galaxy Note 2</i>	720	1280
<i>Samsung Galaxy Note</i>	800	1280
<i>Samsung Galaxy S5</i>	1080	1920
<i>Samsung Galaxy S4</i>	1080	1920
<i>Samsung Galaxy S4 mini</i>	540	960
<i>Samsung Galaxy S3</i>	720	1280
<i>Samsung Galaxy S3 mini</i>	480	800

<i>Samsung Galaxy S2</i>	480	800
<i>Samsung Galaxy S</i>	480	800
<i>Samsung Galaxy Nexus</i>	720	1200
<i>LG Nexus 5</i>	1080	1920
<i>LG Nexus 4</i>	768	1280
<i>Microsoft Lumia 1020</i>	768	1280
<i>Microsoft Lumia 925</i>	768	1280
<i>Microsoft Lumia 920</i>	768	1280
<i>Microsoft Lumia 900</i>	480	800
<i>Microsoft Lumia 830</i>	720	1280
<i>Microsoft Lumia 620</i>	480	800
<i>HTC One</i>	1080	1920

Concernant la GUI :

Portrait

Deux activités :

1. Première activité

Un menu latéral (gauche) caché est accessible par glissement depuis le bord de l'écran. Ce dernier permet de changer les paramètres de l'application, se connecter, etc...

Un ou plusieurs fragments agencés en tuiles (interface météo) affichant les différents événements à venir, un calendrier, les événements en cours, etc...

2. Deuxième activité

Le même menu latéral gauche avec des options en plus permettant de revenir à l'activité principale.

Trois fragments verticaux

1. Width : 100% ; Height : 50%
2. Width : 100% ; Height : 30%
3. Width : 100% ; Height : 20%

Le premier fragment contient la carte/radar avec un bouton flottant opaque (top-right) par-dessus afin de switch entre carte et radar.

Le deuxième fragment contient les informations (essentielle ou détaillée) de l'évènement. Si l'utilisateur est l'organisateur, il peut avoir des boutons supplémentaires permettant de modifier l'évènement. Chaque modification est alors notifiée aux participants.

Le troisième fragment contient les participants. Si l'utilisateur est l'organisateur, il aura des boutons supplémentaires permettant de gérer les participants : broadcast, ajout, suppression, droits, etc...

On conserve ici la même organisation graphique, l'orientation des fragments est inversée. Le menu latéral reste latéral si son contenu n'est pas trop grand, sinon il devient un menu horizontal au top de l'écran, avec glissement.

La première activité ne change pas dû à son interface météo qui s'adapte très bien à toutes les orientations.

La deuxième activité verra ses fragments orientés horizontalement et plus verticalement.

Note : Des maquettes sont à venir

III. Spécifications Techniques Détaillées (STD)

a. Design

Développement, Pattern ?

b. Œuvre

Core Java ? Interface Android ? Choix API ?

c. Architecture, Modules

i. Planning

Développement ou implantation du SEM.

ii. Géolocalisation

API Google Maps

iii. Push position

Développement ou implantation d'un protocole léger.

iv. Radar / Map

API Google Maps pour la carte

Développement du radar avec calcul de la distance (=> Pythagore classique avec les coordonnées dans l'espace) et de la direction (=> calcul à déterminer ^^)

v. Push message

Développement ou implantation d'un protocole léger.

vi. Daemon

Aucune idée de comment ça fonctionne sur Android. Il faut que l'on se renseigne.