

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



swisscom

Rapport final
PPD – Pre-projet de diplôme

**Localisation de personnes sur une carte
géographique**

HEIG-VD

Yverdon-les-bains

mardi 3 juin 2008

Etudiant : Renaud Touillet

Professeur : M. Stephan Robert

Mandant : M. Daniel Rodellar (Swisscom)

1. Table des matières

1.	Table des matières	2
2.	Préambule.....	3
3.	Introduction	3
4.	Analyse du projet	4
4.1	Définition du projet	4
4.2	Use cases.....	6
4.3	Gestion des groupes	7
4.4	Gestion des contacts.....	8
4.5	Flux de données.....	9
4.6	Techniques de localisation.....	10
5.	Choix techniques	12
5.1	Choix du langage	12
5.2	Restriction pour le travail de diplôme.....	13
5.3	Base de données	13
5.4	Technologies utilisables	14
	API Location.....	14
	J4ME	15
	J2MEMap.....	15
6.	Application de test.....	17
7.	Projets similaires.....	19
7.1	Entreprise de tracking	19
7.2	Traçage de son chemin.....	21
7.3	Mobile social Network	21
8.	Conclusion	23
9.	Annexes	24
9.1	Définition du projet	24
9.2	Liens.....	25
9.3	Planning du PPD.....	26
9.4	Résumé journal de travail	27

2. Préambule

Le pré projet de diplôme (PPD) a comme optique la recherche d'informations, la préparation du projet, l'analyse des besoins et des différentes solutions pouvant être utiles pour le travail de diplôme. Le PPD est se compose de 6 périodes, une demi journée, pendant 13 semaines de fin février à fin mai.

Ce temps nous permet de dégrossir le projet et ainsi poser les bases du projet pour que dès septembre, donc le début du travail de diplôme, il soit beaucoup plus facile de nous plonger directement dans la réalisation de notre projet.

Le projet se réalise en partenariat avec Swisscom, via M. Daniel Rodellar superviseur du travail de diplôme.

En annexe se trouve la 1^{ère} description du projet fournie lors du choix du travail de diplôme.

3. Introduction

Depuis plusieurs années, et l'apparition du GPS, sa propre localisation ou la localisation de proches devient beaucoup plus aisée et nécessaires pour certaines personnes, comme les personnes malades qui doivent être surveillées 24h/24. Se demander où se trouvent nos enfants ou nos amis, qui sont par exemple en retard, est instinctif. Plusieurs applications existent déjà pour localiser des personnes à risques ou pour permettre à des parents inquiets de savoir en tout temps où sont leurs enfants. L'étape suivante est bien naturellement localiser ses amis, savoir en tout temps où ils sont, avec leur accord évidemment.

Le projet à donc pour but **le développement d'une application pour les téléphones mobiles permettant la localisation d'une personne ou de groupes à distance avec l'affichage de la ou leurs position(s) sur une carte géographique**. La localisation doit pouvoir se faire avec ou sans GPS.

L'utilisateur ayant installé le programme pour définir des groupes et y ajouter des contacts. Différents droits peuvent être attribués à chaque groupe. Par exemple lors de la localisation d'un des membres du groupe « Enfants » le téléphone ciblé ne pourra pas refuser d'être localisé.

4. Analyse du projet

4.1 Définition du projet

Suite à notre rencontre du jeudi 28 février 2008 avec M. Rodellar et M. Forchelet nous avons pu énoncer les principales fonctionnalités et buts du projet.

Une fois l'application téléchargée sur le téléphone, normalement sur un serveur WEB, l'utilisateur pourra soit créer un nouveau compte ou se connecter s'il s'est déjà enregistré. Puis une fois loguer, il sera possible pour lui d'insérer des contacts et de les classer par groupes. Chaque fois qu'un contact est ajouté, un SMS sera envoyé à ce contact avec le lien de l'application afin de la télécharger, s'il le souhaite, et s'il n'est pas encore inscrit. Des droits pourront être donnés aux groupes pour par exemple créer un groupe « enfants » dont les membres ne pourront pas refuser d'être localisés.



Figure 1 : Ajout d'un contact

Une fois cette liste d'amis créée il sera possible de localiser une à plusieurs personnes simultanément. L'utilisateur choisira la/les personne(s) qu'il veut localiser, une requête sera envoyée à chaque destinataire. Si la personne à localiser n'a pas le programme qui tourne en fond il ne se passera rien. Si le programme est lancé l'utilisateur recevra une requête de localisation. S'il refuse d'être localisé le dialogue entre les deux intervenants s'arrêtera là. Par contre s'il accepte, l'utilisateur pourra choisir le moyen qu'il souhaite pour être localisé : GPS interne, GPS externe connecté via bluetooth, cellule GSM et/ou simple texte. Une fois les données récupérées une réponse sera envoyée au premier téléphone.

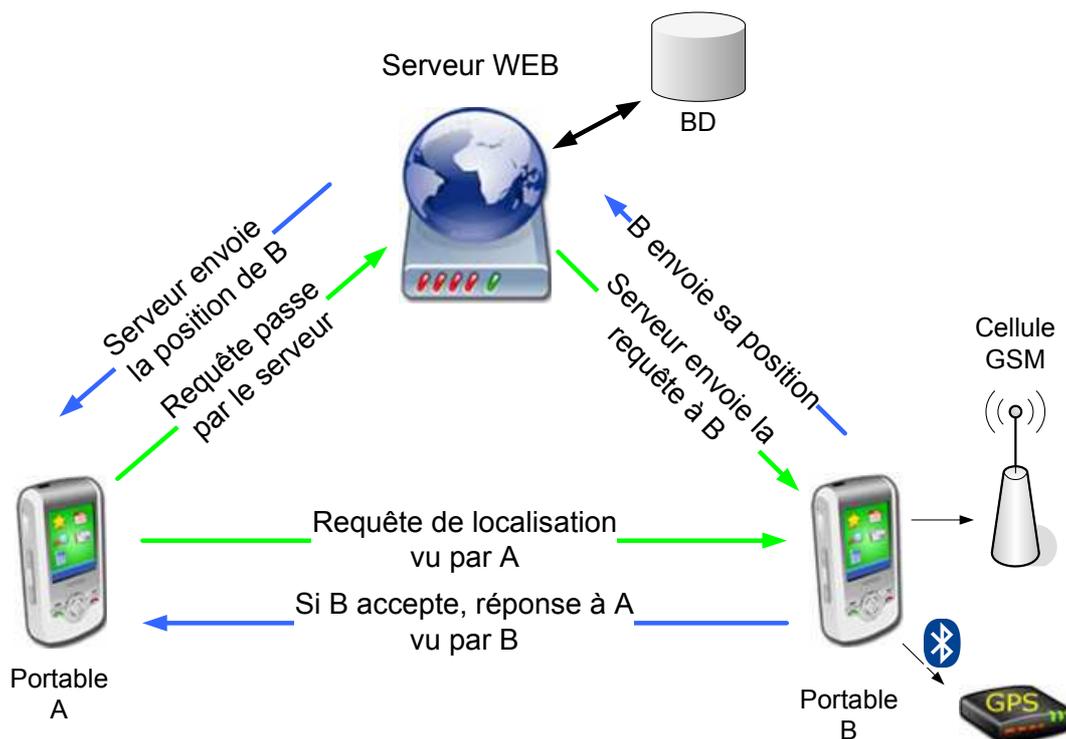


Figure 2 : Localisation d'un portable

Une fois les données reçues elles seront analysées. S'il s'agit de coordonnées GPS (longitude, latitude) le programme pourra afficher les positions sur une carte. Si ce n'est pas possible une estimation de la position pourra être faite grâce aux données de la cellule. Et si ce n'est toujours pas possible le texte, s'il y en a un, sera affiché à l'écran.

Un mini chat pourra être créé où les membres d'un même groupe pourront communiquer entre eux. Cette option sera développée si le temps le permet à la fin du projet.

Voici donc en résumé les fonctionnalités primaires du projet :

- Ajout de contacts et de groupes,
- Envoi d'un sms lors de l'ajout d'un contact,
- Téléchargement du programme sur le serveur,
- Envoi d'une requête de localisation pour une ou plusieurs personnes,
- Acceptation ou refus de localisations,
- Si acceptation de la localisation, choix de l'équipement pour la localisation :
 - o GPS du téléphone
 - o GPS externe
 - o Cellule GSM
 - o Texte
- Envoi de la position du téléphone,
- Affichage des personnes ayant acceptées d'être localisées sur une carte (par exemple GoogleMap)

- Gestion d'un mini chat entre les membres d'un groupe.

Le programme devra être tout public et donc utilisable facilement par tout le monde.

Un serveur web sera mis en place pour le passage de tout le trafic sur le réseau IP car il est impossible de communiquer directement entre deux téléphones portables via le réseau téléphonique. Le passage par le serveur sera normalement invisible au client, mais tout le trafic sera stocké sur le serveur afin d'éventuellement pouvoir créer une interface web affichant la position des personnes.

4.2 Use cases

Les use cases ont pour but de définir les différents types d'utilisateurs ainsi que leur possibilités d'interactions avec le programme. Pour l'application mobile il y aura un utilisateur qui sera décomposé en deux types :

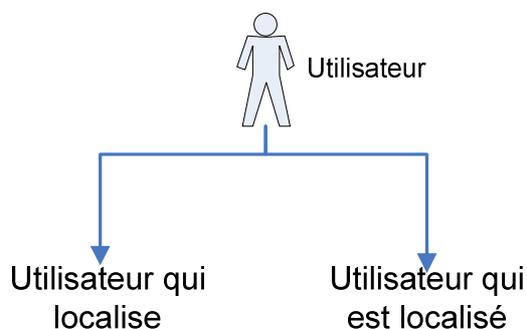


Figure 3 : Deux types d'utilisateurs

Pour l'utilisateur global voici les différents cas d'utilisations et interactions qu'il pourra avoir avec le programme :

- **S'inscrire** : les informations de bases seront collectées auprès du nouvel utilisateur. Son nom, prénom, pseudo, mot de passe, date d'inscription ainsi que différentes informations récoltées directement du téléphone, si possible, tel que le type du téléphone ou s'il dispose d'un GPS interne. Les informations seront envoyées et stockées dans la base de données sur le serveur. Le client, une fois l'inscription validée, sera invité à se connecter.
- **Se connecter** : grâce au mot de passe et au pseudo. Une requête de connexion est lancée sur le serveur. Si l'authentification réussit l'utilisateur passera en mode On-line (voir « Gestion des statuts »), et l'utilisateur arrivera sur le menu principal. Si l'authentification échoue il retombera sur la page de connexion.
- **Gérer ses groupes** : l'utilisateur pourra créer, modifier ou supprimer des groupes. La gestion des groupes est expliquée dans la partie « Gestion des groupes ».
- **Gérer ses contacts** : les contacts pourront être ajoutés, modifiés ou supprimés. Un contact appartient toujours à un groupe. Les contacts seront identifiés par un pseudo ainsi que par le numéro de téléphone. Les différents statuts (on-line, off-line, ..) est définie plus bas (voir « Gestion des contacts »). On pourra faire une recherche des utilisateurs déjà enregistré sur le serveur et l'ajouter le cas échéant, sinon un message sera envoyé à la personne pour lui proposer de télécharger l'application et s'inscrire.

- **Afficher la carte** : l'utilisateur pourra afficher la carte pour éventuellement voir sa position ou encore trouver des endroits qui l'intéresse. Il faudra encore analyser les différentes maps disponibles ainsi que leur fonctionnement selon le langage qui sera utilisé.
- **Modifier les paramètres** : les paramètres personnels (mot de passe par exemple) pourront être modifiés sur un menu spécial. Les modifications seront transmises au serveur pour être modifiées dans la base de données.
- **Tchater** : il sera possible de discuter avec un ou plusieurs utilisateurs. Eventuellement avec un groupe entier. Chaque message envoyé ou reçu nécessitera d'être connecté à un réseau.

Voici maintenant les use cases pour un utilisateur voulant localiser une ou plusieurs personnes :

- **Sélection de contacts** : dans la liste des contacts l'utilisateur pourra sélectionner un ou plusieurs contacts qu'il désire localiser. Une fois cela fait il lancera une requête de localisation au serveur.
- **Attente de réponse** : Une fois les requêtes lancées l'utilisateur attendra les réponses des autres utilisateurs qui transiteront par le serveur. L'utilisateur pourra continuer à utiliser le programme pendant ce temps là.
- **Réception des données** : à chaque fois qu'une information arrive de la part d'un utilisateur, le programme regarde si les données peuvent être affichées sur la carte (données d'un GPS, localisation valide,...), sinon on affichera les données textes qui auront été reçues. Plusieurs points pourront être affichés sur la carte.

Les personnes qui recevront une demande de localisation pourront interagir comme suit :

- **Réception d'une demande de location** : lorsqu'un utilisateur reçoit une demande de location plusieurs choix s'offrent à lui :
 - o Refuser d'être localisé,
 - o Etre localisé grâce à un GPS interne ou Bluetooth,
 - o Etre localisé grâce aux antennes GSM,
 - o Indiquer un texte pour être localisé plutôt qu'une position.

La réponse transitera par le serveur afin que la localisation actuelle soit gardée dans la base de données puis envoyée à la personne qui a émit la requête.

4.3 Gestion des groupes

Les groupes seront répartis en deux catégories principales. Tout d'abord une catégorie qui pourra être localisée n'importe quand, avec un accord préalable unique, par exemple famille ou enfants. Et une catégorie de groupes à qui une demande de localisation devra se faire à chaque fois.

La personne ajoutée comme contact au groupe...	Groupes type 1	Groupes type 2
Doit-elle accepté d'être ajoutée à ce groupe ?	Oui	Non
Peut-elle être localisée à tout moment ? (pour autant que le programme soit lancé)	Oui	Non
Peut-elle refuser d'être localisée ?	Non	Oui

Recevra t-elle une indication comme quoi elle a été localisée ?	Oui	Oui
---	-----	-----

L'utilisateur pourra créer autant de groupes qu'il voudra. Puis de là il pourra ajouter des utilisateurs dans les groupes. Il faudra gérer le fait que l'utilisateur pourra se déplacer et éventuellement changer de téléphone, donc pour ceci les contacts et les groupes seront stockés dans la base de données sur le serveur. Afin d'éviter de mettre à jour cette base de donnée à chaque nouveau groupe, il sera créé, sur la BD, une fois qu'un contact sera ajouté dans dedans.

Pour créer un nouveau groupe il suffira de se rendre dans la page qui affiche tous les groupes, d'ouvrir le menu et de choisir « Créer un nouveau groupe ». Lui donner un nom puis de choisir le type de groupe. De là le il sera ajouté à liste. Mais il ne sera effectif sur le compte de l'utilisateur que lorsqu'il y aura ajouté un utilisateur.

4.4 Gestion des contacts

Les contacts qui apparaitront à l'utilisateur dans le programme seront visuellement différemment selon leur statut.

On peut distinguer 3 statuts dans un premier temps :

- **En ligne** : c'est-à-dire que l'utilisateur a lancé l'application ou qu'il s'est connecté dans les dernières X minutes (par exemple 30 minutes),
- **Hors ligne** : si on a plus de contact avec l'utilisateur depuis plus de X minutes (par exemple 30 minutes, dépend du premier statut).
- **Off ligne** : Utilisateur qui n'a pas encore accepté l'invitation au programme.

Puisque le programme sera installer sur le téléphone et devra fonctionner sans forcément être connecté à une borne Wifi ou un avoir un accès internet gratuit, il faudra par conséquent limiter le nombre de connexion au serveur pour limiter les couts. Pour les contacts, leur statut sera mis à jour lors de la connexion de l'utilisateur au programme. Puis l'utilisateur pourra faire une demande de mise à jour des statuts de ses contacts. Il n'est pas imaginable d'avoir une gestion des contacts comme dans « MSN Messenger » par exemple car cela demanderait d'avoir une connexion constante au serveur.

Voici les étapes lorsqu'un utilisateur souhaite ajouter un contact :

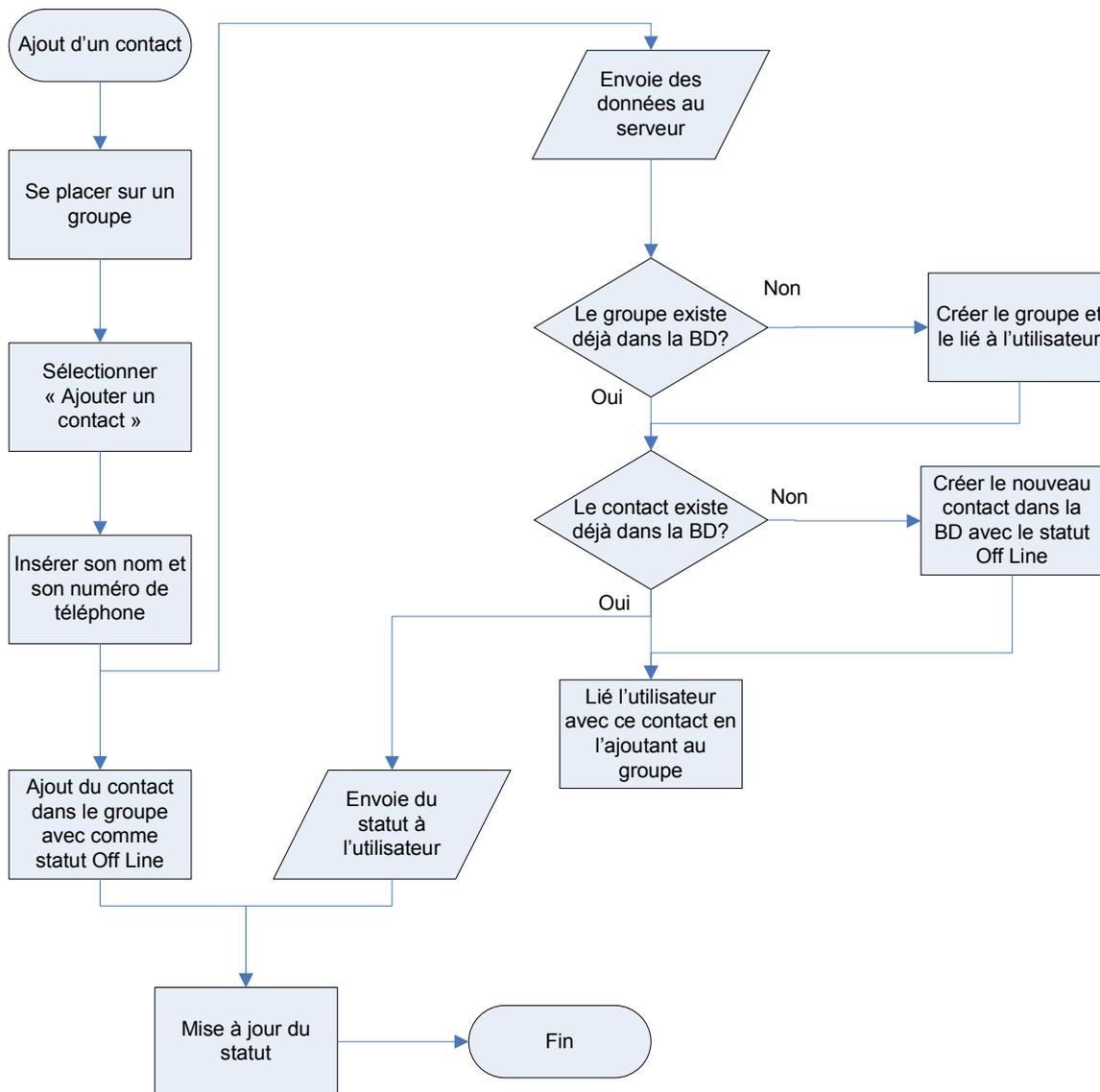


Figure 4 : Ajout d'un nouveau contact

4.5 Flux de données

Suite à cette première analyse, 4 acteurs principaux peuvent être mis en avant :

- Utilisateur
- Serveur Web
- Serveur SMS
- Fournisseur de Map

Voici dans un premier temps le diagramme de flux qui sera généré entre les différents acteurs du projet :

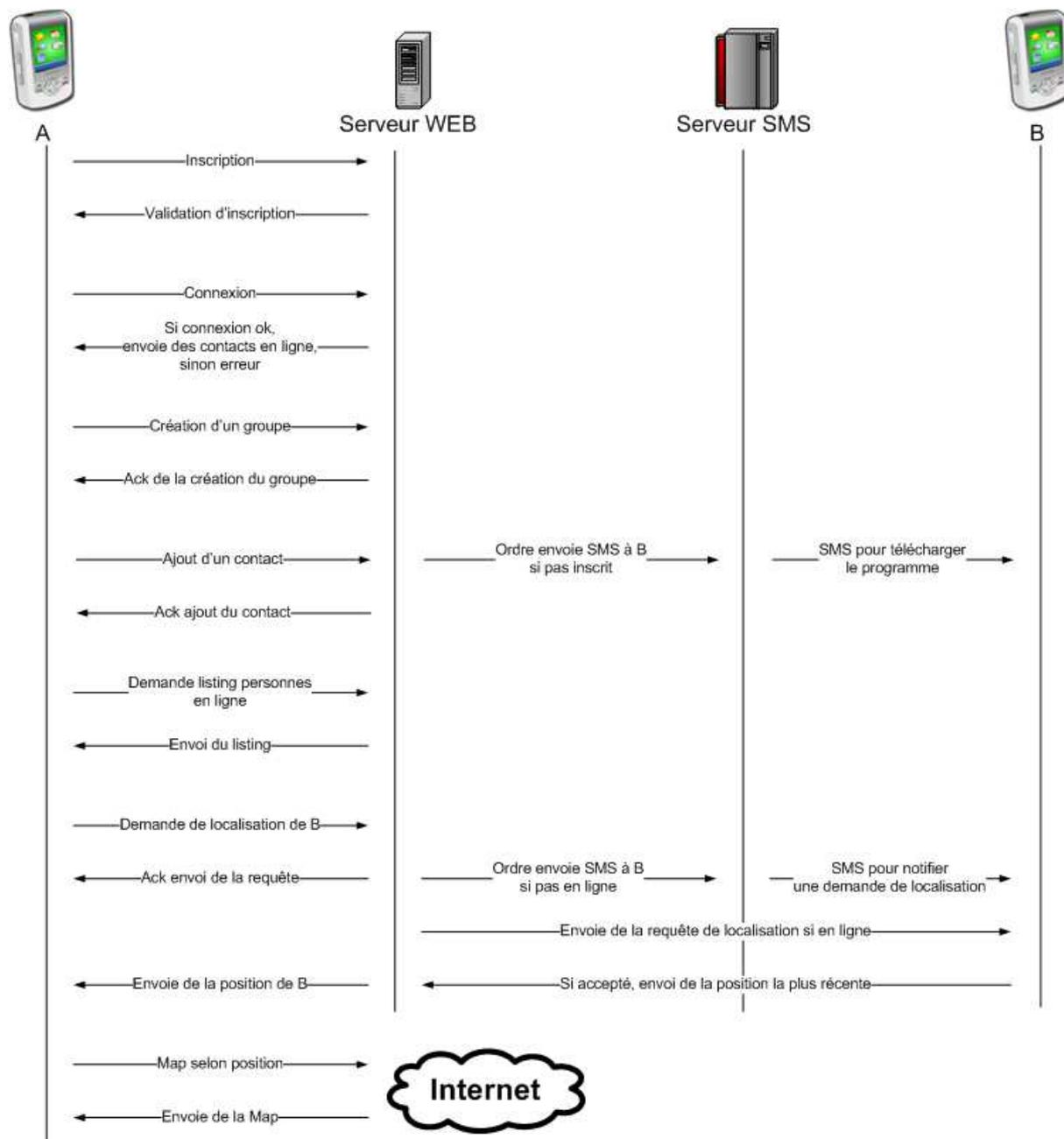


Figure 5 : Ajout d'un nouveau contact

La place du serveur SMS est encore à définir précisément. Du reste il faut encore voir s'il sera possible de contacter un téléphone directement par http si ce dernier n'est pas en « contact » permanent avec le serveur.

4.6 Techniques de localisation

La localisation est un des points les plus sensible. En effet on ne peut pas transmettre de données de localisations sans l'accord de la personne pour des raisons évidentes de liberté de la vie privée. Mais

dans ce projet le programme étant basé sur cette localisation il faudra demande à l'utilisateur quel technique il souhaite choisir.

Dans un premier temps la technique la plus évidente est le GPS. Il peut être interne ou externe. Un GPS externe communique généralement via bluetooth avec le téléphone. Le but sera donc de récupérer les données, latitude et longitude principalement, et de les stocker.

La seconde option qui a été choisie est la localisation grâce aux antennes GPRS. C'est-à-dire que grâce au nom de la borne ainsi que la puissance du signal il est théoriquement possible de créer un périmètre approximatif afin de détecter la position de la personne.

Le but du programme en background est d'essayer de mettre en application les deux méthodes et de les rendre complémentaires. C'est-à-dire que le programme une fois lancé, va essayer d'obtenir la position de la personne toutes les X secondes grâce à son GPS et également les informations sur la cellule et sa puissance. Le temps X étant un paramètre qui sera modifiable par l'utilisateur, il pourra également choisir de ne rechercher sa position que lorsqu'il reçoit une requête. Grâce aux informations collectées la localisation de la personne pourra être plus aisée car si au moment de la demande l'utilisateur ne peut pas capter de signal GPS il pourra toujours envoyer sa dernière position. De plus les informations collectées seront envoyées au serveur afin de rendre la localisation d'autres personnes ne possédant pas de GPS plus facile. En effet si une personne ne dispose pas de GPS et envoie uniquement les informations de l'antenne à laquelle elle est reliée et qu'une donnée similaire existe dans la base de données mais cette fois avec une position GPS associée il sera plus aisé de fournir une localisation plus juste. Evidemment il existera toujours une marge d'erreur du fait qu'une antenne émet un signal circulaire autour d'elle. Mais si on arrive à récolter de grandes quantités d'informations du couple GPS/cellule il sera possible grâce à un minimum de triangulation de trouver la position optimale.

Si aucune position GPS n'est trouvable après un nombre d'essais également paramétrable, le programme s'arrêtera de chercher des données de positionnement et reprendra si l'utilisateur le souhaite ou dès qu'une demande de localisation émanera d'un autre utilisateur. Tout ceci dans le but d'économiser la batterie car la recherche de données GPS demande de grandes ressources au niveau énergie.

5. Choix techniques

5.1 Choix du langage

Après quelques recherches 3 langages sont envisageables pour le développement d'une telle application :

- C sharp
- Symbian
- Java

Symbian est un OS propriétaire installé en grande partie sur les Nokia. Une application développée sous Symbian ne pourra donc pas être exportable sur une autre plateforme. De plus je n'ai jamais programmé d'application pour cet OS donc je ne connais pas ces possibilités.

C Sharp, qui est actuellement très utilisé dans le développement d'application sous Windows, est un langage orienté objet proposé par Microsoft. Malheureusement pour le développement d'application mobile il faut disposer d'un environnement d'exécution pour les programmes, la CLR. Cet environnement est très peu répandu sur les plateformes mobiles et requiert de grande capacité mémoires. De plus il faut disposer de Windows Mobile sur le téléphone pour pouvoir exécuter correctement les applications développées avec ce langage.

Le 3^{ème} langage est proposé par Sun, Java, via sa plateforme de développement J2ME pour les téléphones portable. La machine virtuelle KVM est présente sur la plus grande majorité des téléphones portables modernes, qu'ils soient sous Symbian ou sous Windows Mobile. Le plus gros problème vient du fait que j2me ne supporte pas les communications natives donc directement avec le matériel. Après discussion avec M. Jatou, il en est ressorti que si on veut attaquer directement un GPS interne au téléphone il fallait utiliser le framework JNI.

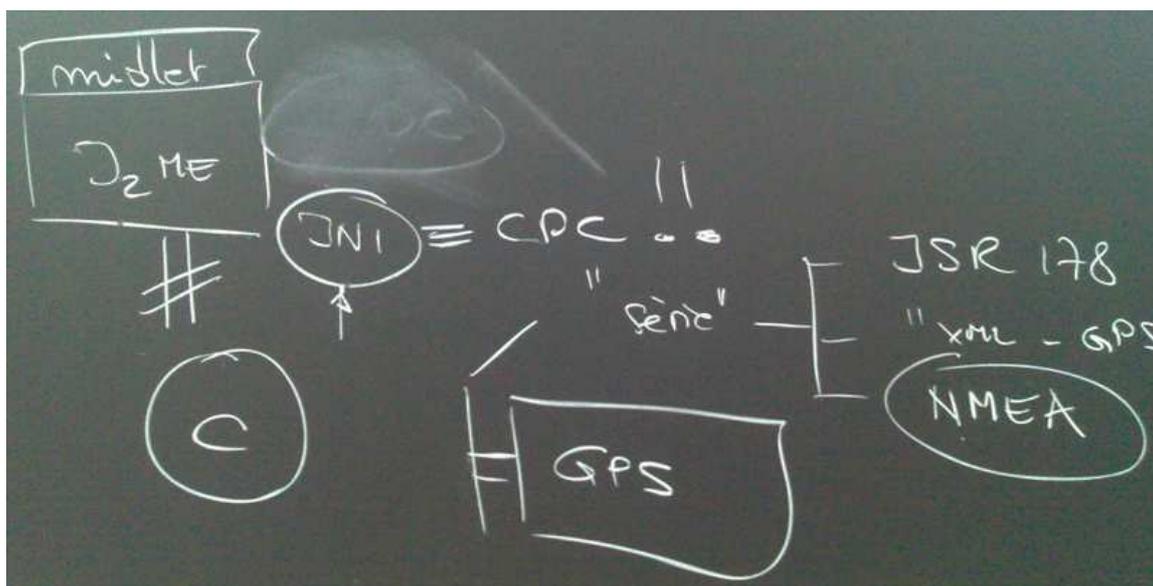


Figure 6 : GPS vers Midlet j2me avec JNI

Le JNI (**Java Native Interface**) est un framework qui permet à du code Java s'exécutant à l'intérieur de la JVM d'appeler et d'être appelée par des applications natives (c'est-à-dire des programmes spécifiques au matériel et au système d'exploitation de la plate-forme concernée).

Source : http://fr.wikipedia.org/wiki/Java_Native_Interface

L'utilisation de JNI théoriquement implique de passer de la configuration CLDC, présent sur tous les téléphones, à CDC plus compliqué et moins présent sur les téléphones. En résumé il faudrait développer en J2EE, Java standard, au lieu de J2ME ce qui rend le nombre de téléphones utilisables quasi nul.

Le choix du langage s'est tout de même porté pour J2me car il représente une plus grande portabilité que les deux autres. Malgré sa faible puissance au niveau du développement et du graphisme il permet grâce à diverses API de toucher le GPS interne ou Bluetooth. De plus il gère assez bien la connexion http.

5.2 Restriction pour le travail de diplôme

Vu le nombre possible de téléphones et d'OS qui existent il est impensable de pouvoir faire, du premier coup, une application qui fonctionne sur toutes les plateformes. Du reste toutes les applications que j'ai pu tester ou rencontrer avaient des restrictions quant au type de plateforme ou de matériel nécessaire. Donc pour mon travail de diplôme je pense me restreindre à un type de plateforme quitte à avoir un programme qui ne tourne que sur un certain type de téléphone mais pouvant quand même être déployé sur les autres plateformes mais avec des restrictions. Le but de cette restriction est d'avoir une application qui, à la fin de mon travail de diplôme, soit fonctionnelle. Voici les caractéristiques techniques qui permettront d'utiliser l'application :

- OS Symbian Série 60
- Plateforme Java
- CLDC 1.1 pour J2me
- Possédant la JSR 179

Tout sera mis en œuvre pour essayer de faire tourner l'application même si toute ces contraintes ne sont pas respectées.

Dans le cadre du travail de diplôme le développement et les tests se feront grâce au Nokia N95.

5.3 Base de données

Suite à la première analyse nous pouvons tirer que la base de données devra stocker les éléments suivants :

- Les utilisateurs,
- Les groupes, chaque groupe appartenant à un utilisateur,
- Les utilisateurs qui appartiennent aux groupes,

- Les statuts des utilisateurs,
- Les positions des utilisateurs, coordonnées et/ou texte.

La base de données devrait se trouver sur le serveur web. Il s'agira vraisemblablement d'une base MySQL contenant les informations précitées. Voici un premier jet de du schéma relationnel des données qui seront stockées dans la base de données. Evidemment elle pourra évoluer selon les besoins futurs.

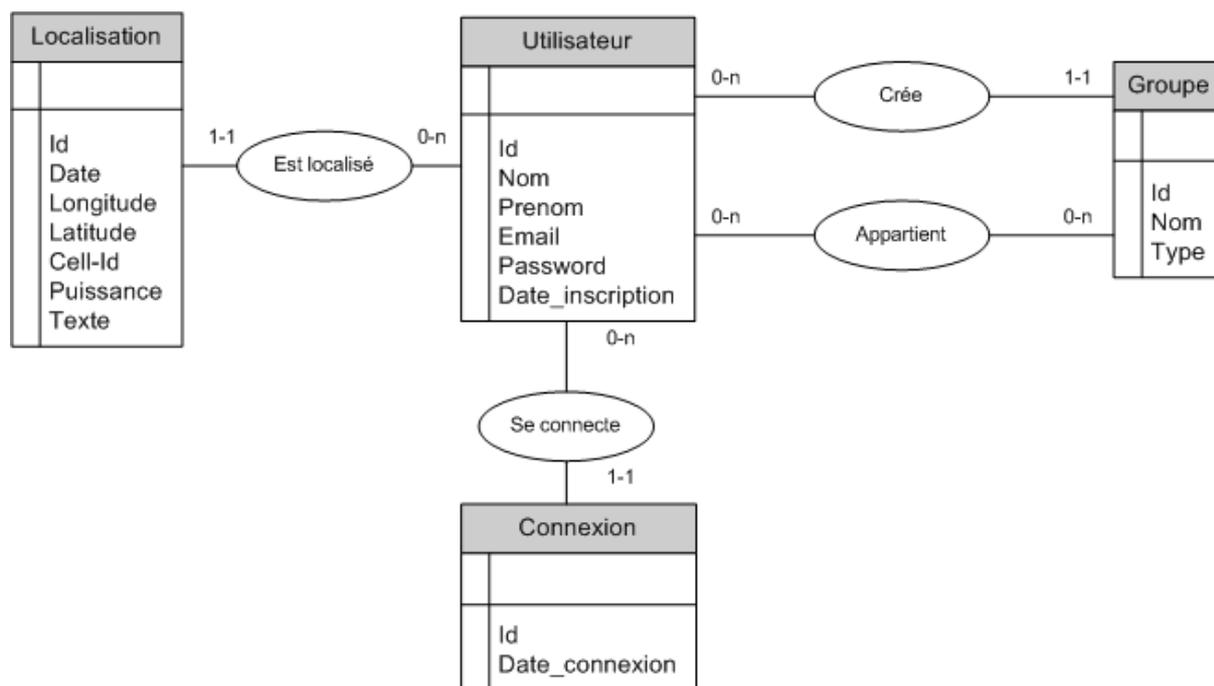


Figure 7 : Schéma provisoire de la base de données

Comme expliqué plus haut les utilisateurs définissent des groupes, donc chaque groupe appartient à un utilisateur, avec des droits. Puis ils peuvent ajouter d'autres utilisateurs aux groupes créés. Pour gérer l'état du contact (connecté ou non) une table Connexion permet de stocker toutes les connexions d'un utilisateur. De là il est aisé, grâce à l'heure de la connexion, de savoir de quand date sa dernière connexion au serveur.

5.4 Technologies utilisables

Lors de mes recherches d'exemples et d'applications pouvant être proche de mon projet, j'ai trouvé plusieurs classes ou technologies pouvant être utilisées dans le développement de mon projet. En voici une liste non exhaustive :

API Location

L'API Location, où JSR 179, est un service proposé à la base par Nokia. Le but est de pouvoir définir la position physique d'un mobile dans un mode standardisé. Cette API est présente dans la plupart

des nouveaux téléphones disposants de l'OS Symbian avec la version CLDC 1.1 (Connected Limited Device Configuration). Malheureusement elle ne marche pas avec Windows Mobile.

La détermination d'une position se passe grâce à des critères fournis à la classe. Il faut tout d'abord définir un objet du type « criteria » :

```
Criteria cr= new Criteria();
```

Lui donner des critères de recherche, comme la précision de la position par exemple, puis l'objet fourni des fonctions qui nous retournent les valeurs désirées, comme la latitude ou la longitude par exemple :

```
double lat = c.getLatitude();
```

```
double lon = c.getLongitude();
```

De là on peut utiliser ces valeurs. Ces API est très facile à utiliser mais malheureusement il y a un temps non négligeable avant de pouvoir avoir accès aux données du GPS. Il faut compter environ une 30aine de secondes si le téléphone se trouve en pleine air dans un endroit dégagé. Et bien évidemment si nous nous trouvons dans un bâtiment son utilisation est impossible du fait que le GPS ne fonctionne pas.

Explication : <http://developers.sun.com/mobility/apis/articles/location/>

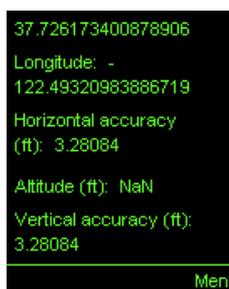
Final release : <http://jcp.org/aboutJava/communityprocess/final/jsr179/index.html>

J4ME



librairie au projet (dans Eclipse par exemple).

J4me est une collection de classes fournies par Google afin d'aider au développement en J2me. On y trouve notamment une implémentation simplifiée pour le graphisme qui permet de garder un design cohérent et agréable tout au long de l'utilisation du programme. De plus elle fournit facilités pour accéder au GPS. Pour l'utiliser il suffit de l'ajouter comme



Lien : <http://code.google.com/p/j4me/>

Installation : <http://code.google.com/p/j4me/wiki/Setup>

J2MEMap

J2MEMap est une interface qui permet d'accéder à l'entière base de données de GoogleMap, d'afficher les cartes et de naviguer dessus. En résumé ce SDK permet une aide concrète à l'utilisation de GoogleMap. Basé sur une licence GPL, cette librairie est vraiment très puissante et en quelques lignes elle offre l'accès simplifié aux cartes.

Introduction : <http://i2memap.landspurg.net/> (public)
<http://i2memap.8motions.com/> (développeurs)

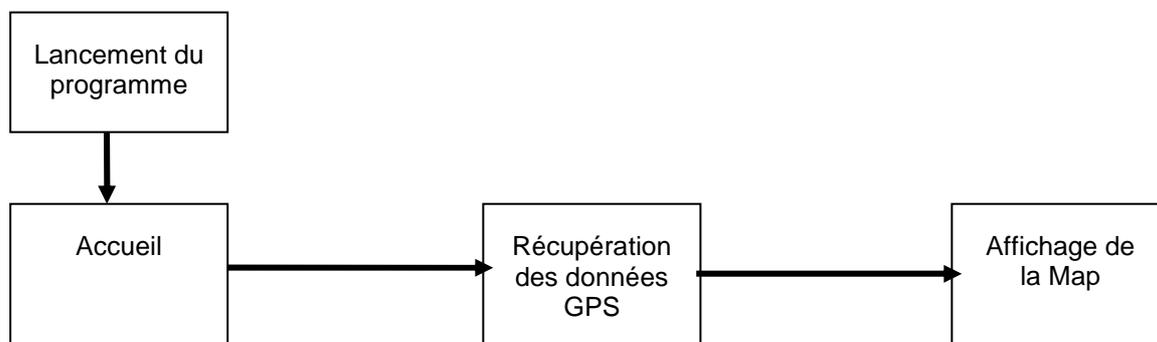
Téléchargement du SDK : <http://i2memap.8motions.com/user/sdk> (Il faut être enregistré)

6. Application de test

Afin de reprendre la main sur la programmation en J2me, qui peut être un peu rébarbative au début, une petite application de test a été créée. Elle n'est évidemment pas très évoluée vu le peu de temps, mais elle permet de mettre en avant 4 technologies précitées :

- J2me pour la programmation,
- J4me pour le graphisme,
- API Location pour accéder aux données GPS,
- Google Map pour récupérer la carte selon la position, grâce à J2meMap.

Le programme se compose de 3 écrans :



Tout d'abord il faut télécharger le fichier .jar (via bluetooth) sur le téléphone. Puis il faut installer le programme. Sur les Nokia les programmes Java installés se retrouvent dans la partie « Applications ». Une nouvelle icône apparaît avec le nom du programme.

Une fois l'application lancée elle vous propose de lancer la localisation (bouton bas à droite). Une fois cliquez sur le bouton la récupération des données GPS prend un peu de temps. Elles s'affichent automatiquement à l'écran. Malheureusement si le



téléphone n'arrive pas à accéder aux données de localisation il y a une erreur qui est générée. Pour l'instant aucun traitement des exceptions n'est implémenté.

Mais si les données s'affichent il suffit de nouveau de cliquez sur le bouton en bas à droite pour afficher votre position sur une carte image générée avec Google Map.



La carte ainsi générée est une vue de Google Map avec un point (petite image au centre) où se trouve précisément les coordonnées trouvées. Le zoom est fixé d'avance dans le code, mais vous pouvez vous promenez sur la carte (gauche, droite,...). De plus la vue est en mode « Plan » mais il est envisageable de passer en mode « Satellite », car il ne s'agit que d'un seul paramètre à changer lors de la requête à Google Map. Ce dernier permet également de positionner plusieurs « points » donc de localiser plusieurs personnes sur une seule carte.

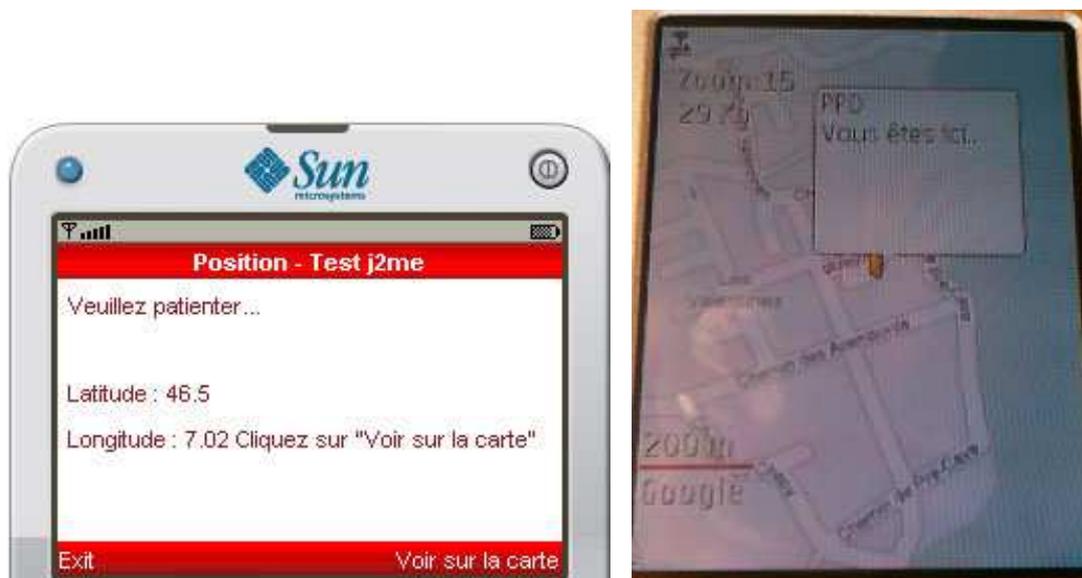


Figure 8 : Position GPS et affichage dans Google Map sur le N95

Cette application n'est évidemment pas grandiose mais permet une petite idée de ce qui est possible de faire avec quelques lignes de codes.

Le fichier fournit pour la démo est demoPPD.jar, testable sur le téléphone, de préférence un N95. Il est possible d'utiliser le programme avec le compilateur WTK de Sun et son simulateur de téléphone grâce au fichier demoPPD.jad également fournit (il faut les deux fichiers dans ce cas).

Le code du programme se trouve dans le fichier demoPPD.rar, et une fois décompressé dans le dossier src/PPD.

7. Projets similaires

Voici un listing des applications similaires ou se rapprochant du projet.

7.1 Entreprise de tracking

« GPS Tracking

GPS Tracking est un terme utilisé pour définir l'emplacement d'une personne, un véhicule ou tout autre actif en utilisant le système de positionnement global. Habituellement, la localisation de l'objet est enregistrée à intervalles réguliers. L'enregistrement d'un objet peut avoir lieu dans le récepteur GPS unité, ou peuvent être transmis à un emplacement central, tels que la base de données, où il peut être consulté via la technologie de téléphonie mobile ou sur Internet avec l'utilisation d'un ordinateur. Suivi GPS des unités cellulaires peuvent utiliser le GPRS, une technologie ou un modem satellite ondes radio pour transmettre l'information à l'endroit central où les données sont enregistrées et consignées. Habituellement, la plupart des systèmes de repérage par GPS peuvent être visualisées en temps réel et ont une carte à l'arrière-plan pour facilement afficher la position actuelle d'un objet. »



<http://www.tech-faq.com/lang/fr/gps-tracking.shtml>

La plupart de ses entreprises fournissent un service payant au départ (achat d'un « tracker ») puis l'utilisation est libre selon les besoins de l'entreprise (ou du particulier) ayant acheté le produit. Certaines entreprises fournissent un logiciel de visualisation directement sur leur site sinon ils proposent de visualiser les données sur Google Earth.



<p>Trackstick : http://www.trackstickpro.fr/</p>	<p>Entreprise française qui vend des traqueurs GPS pour les entreprises et les particuliers. Comme la plupart des autres sites l'application la plus intéressante et mise en avant est la localisation de flottes d'entreprises.</p>
<p>Spotnik Mobile par Inov Mobile : http://pagesperso-orange.fr/spotnik/presentation.html</p>	<p>Développé par un indépendant. Solution de géolocalisation pour la plate forme windows mobile. La figure suivante montre le fonctionnement de Spotnik.</p>

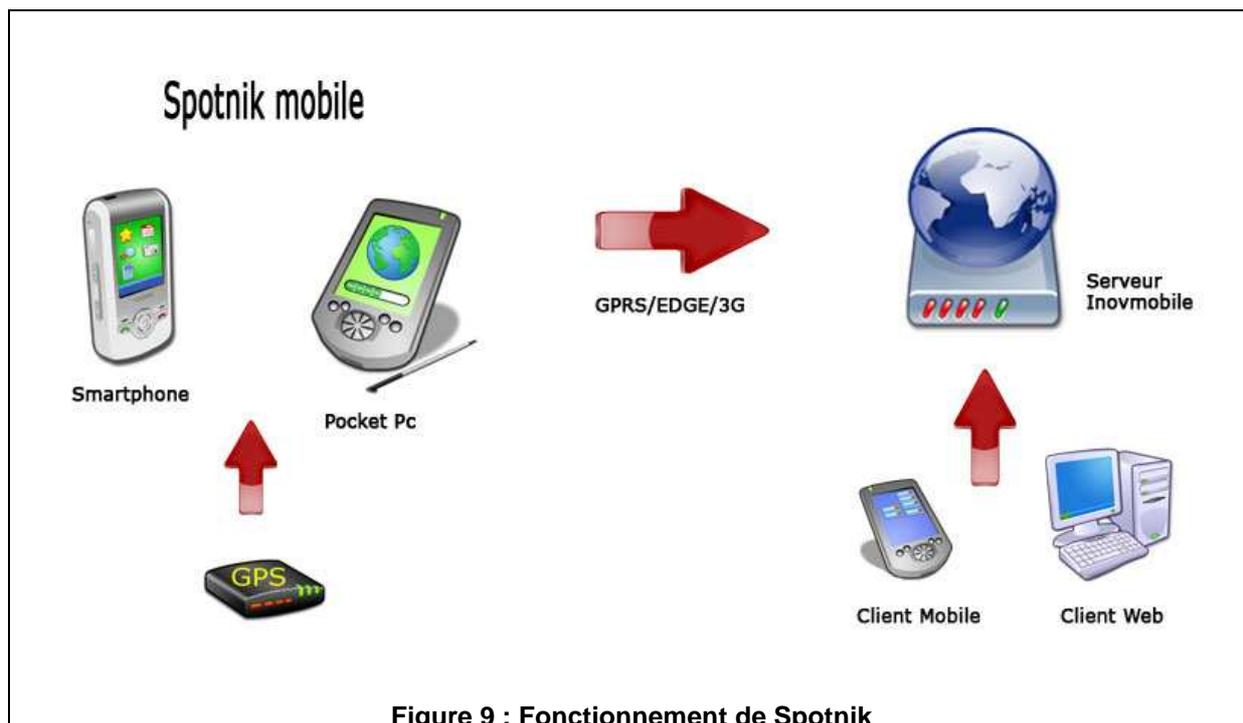


Figure 9 : Fonctionnement de Spotnik

<p>Locate Angel : http://www.espace-mobilite.com/geolocalisation_locate_angel.asp</p>	<p>Cette entreprise propose un traqueur pour les personnes souhaitant être localisée telle que les personnes âgées ou malades.</p>
<p>Ootay : http://www.ootay.fr</p>	<p>Entreprise française qui permet la localisation de téléphone portable directement via les relais GSM des opérateurs. Il faut s'inscrire, son numéro de téléphone notamment, mais surtout il faut enregistrer les personnes que l'on veut localiser sur le site.</p>

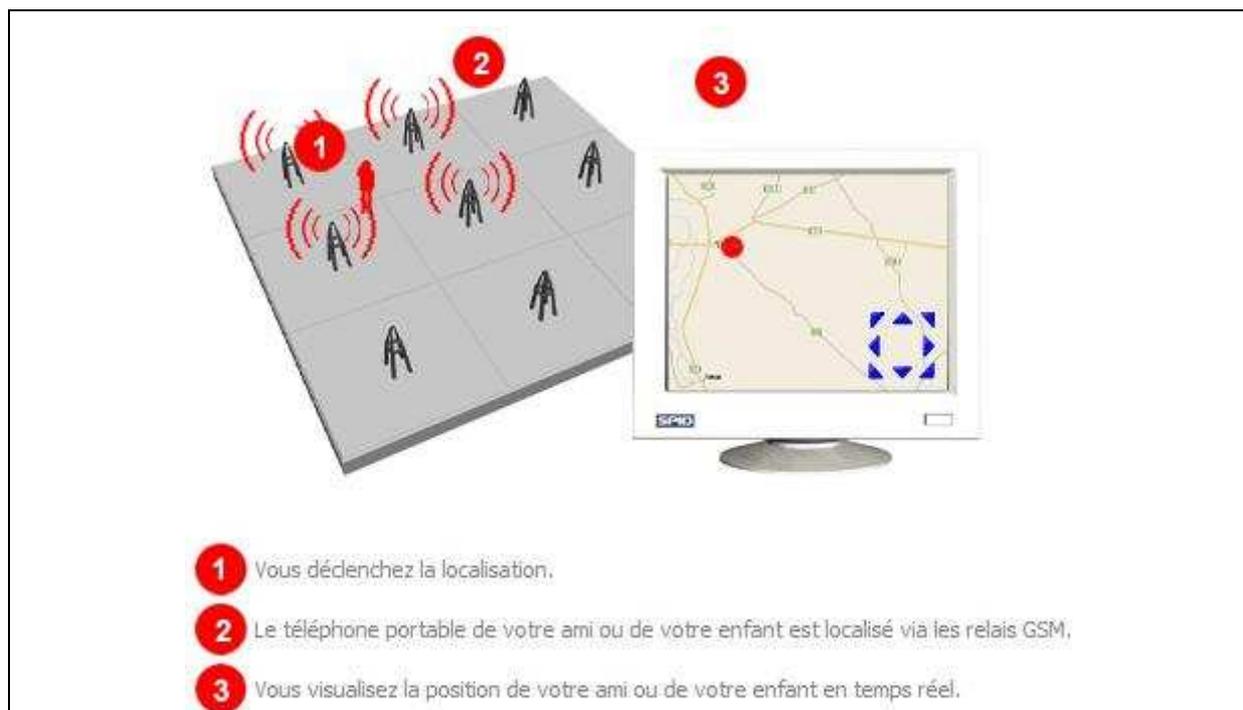


Figure 10 : Fonctionnement de Ootay

Vidéo, de 2006, sur le concept de Ootay ainsi que sur le développement d'une application en Suisse sur le site Nouvo.ch : <http://www.nouvo.ch/103-2>

GeoGeny :

<http://www.geogeny.ch>

Entreprise Suisse qui permet la localisation de flottes de voitures et de personnes grâce à des émetteurs GPS. Présent au Montreux Jazz pour localiser les agents de sécurité par exemple. Dispose également d'un module pour la localisation d'enfants.

7.2 Traçage de son chemin

<http://www.substanceofcode.com/software/mobile-trail-explorer/>

Application Open Source permettant d'enregistrer son chemin lors d'un déplacement. Les informations tel que la distance parcourue ou le dénivelé sont accessibles. Il faut avoir un gps soit dans le téléphone ou accessible via Bluetooth. Le programme permet d'exporter les points au format kml pour que le chemin parcouru puisse être visualisé sur google earth

7.3 Mobile social Network

Ce point est le plus proche à première vue du projet souhaité. Les projets de social network permettent en général d'avoir sa liste d'amis sur son portable et de pouvoir les localiser, discuter ou encore partager des documents avec eux. Il est en général fréquent de pouvoir localiser des

personnes ayant les mêmes centres d'intérêt que soi-même. On peut comparer cela à des MySpace ou des Facebook Mobile.

Plus d'informations : http://en.wikipedia.org/wiki/Mobile_social_network

<p>Projet en développement par Inov Mobile : http://pagesperso-orange.fr/spotnik/autres_projets.html</p>	<p>Description du programme :</p> <p>Garder le contact avec votre communauté! Soyez prévenu si un ami est proche de vous, Localisez vous, Dialoguer en direct, Partager vos photos avec votre "tribu", Rechercher le distributeur de billet le plus proche, Et bien plus encore....</p>
<p>ZYB : http://zyb.com/</p>	<p>Communauté de personne connectée sur internet mais les contacts sont transférées depuis le téléphone portable. Description du site :</p> <p>« ZYB stocke de votre téléphone vos contacts, votre agenda et vos messages en ligne et vous met en rapport avec des personnes que vous connaissez. »</p>

8. Conclusion

Savoir où se trouve une autre personne que soit même n'est pas une mince affaire. De nombreux points s'opposent à une simplicité de développement d'une telle application. Même si plusieurs sites proposent déjà des produits similaires il est de toute évidence très difficile d'imposer un produit qui demande une constante connexion et une autorisation complète de la personne localisée. Ce projet est donc l'opportunité de développer un programme simple d'utilisation qui permet à tout en chacun de l'utiliser, avec quelques restrictions matérielles évidemment.

Même s'il faut savoir imposer des restrictions à tout projet, j'espère pouvoir atteindre mes objectifs pour ce travail. C'est-à-dire une application fonctionnelle, même si elle n'intègre pas toutes les fonctionnalités que propose d'autres concurrents, agréable à utiliser, mais surtout mettre au service de ce projet les connaissances acquises durant ces 3 années d'études.

Ce pré-projet de diplôme fut donc une expérience très intéressante dans le domaine de la localisation en générale. Même si je n'ai pas pu analyser à 100% tout les besoins de l'application, j'ai pu dégrossir le travail et cela me permettra de partir confiant lors du travail de diplôme en septembre. De plus j'ai pu y découvrir quelques programmes ou librairies très intéressantes à utiliser.

Bex, le mardi 3 juin 2008

Touillet Renaud

9. Annexes

9.1 Définition du projet

Titre Localisation de personnes sur une carte géographique (Réservé)

Responsable Robert Stephan

Résumé

Le sujet du travail consiste à développer une application pour téléphones mobiles qui permette de créer plusieurs groupes (comme famille, amis, collègues, etc) et qui permette de les localiser sur une carte géographique. Si l'application se trouve sur un téléphone qui contient un GPS alors la position sera renvoyée à la personne après avoir reçu la permission (« Stephan demande : où es-tu ? veux-tu lui répondre ? OUI-NON»). On pourrait même envisager que certains groupes aient la permission par défaut (des parents, par exemple, et qui pourraient à tout moment localiser leurs enfants). Si le téléphone n'a pas de GPS alors c'est la position de la cellule à laquelle il est rattaché qui donne la localisation. La représentation de la localisation peut se faire avec des points sur la carte d'un rayon plus ou moins grand en fonction de l'incertitude. En référence Google Maps donne un exemple de leur service « My Location ». Le sujet permet de travailler sur plusieurs aspects : - développement sur des téléphones mobiles - utilisation de la technologie GPS - réalisation d'une application pratique d'intérêt au grand public - possibilité pour l'étudiant de proposer d'autres alternatives ou de compléter le système proposé - implications au niveau des serveurs au back-end et des aspects de sécurité des informations transmises Le langage de programmation n'est pas imposé. Java pourrait être plus portable que de faire le développement en Symbian, mais c'est à l'étudiant de faire son choix. De même pour les protocoles d'échange d'information et des serveurs au back-end. Référence : <http://www.google.com/gmm/mylocation.html?hl=en>

9.2 Liens

Tracking:

<http://www.avltrack.com/>

www.chaeron.com

Geolocalisation:

<http://www.google.com/gmm/mylocation.html?hl=en>

<http://www.windows-mobile.ch/forum/showthread.php?t=2357>

http://www.espace-mobilite.com/explication_suivi_gps_sur_telephone_gsm.asp

J2ME:

<http://www.cs.kent.ac.uk/teaching/07/modules/CO/8/31/wtkdocs/docs/UserGuide-html/index.html>

<http://www-adele.imag.fr/users/Didier.Donsez/cours/tpj2me.htm>

J2me et gps :

<http://www.developpez.net/forums/archive/index.php/t-392541.html>

http://www.forum.nokia.com/info/sw.nokia.com/id/175bf8e6-a1f5-4d3d-a591-6fc936506a6b/MIDP_Location_API_Developers_Guide_v2_0_en.pdf.html

<http://www.devx.com/wireless/Article/30380/0/page/3>

http://www.iprofs.nl/pdf/Location_Based_Services_J-Spring.pdf

<http://www.jcp.org/en/jsr/detail?id=179>

Google map j2me:

<http://j2memap.landspurg.net/index.html>

J2me et bluetooth :

<http://www.geekzone.fr/ipb/index.php?showtopic=33725>

Java et gps :

<http://www.superwaba.com.br/etc/wi-gps-a4.pdf>

<http://mobilepit.com/05/javame-location-api-example-application-with-source-code.html>

CDC :

http://www.developer.com/java/j2me/article.php/10934_3639871_1

9.3 Planning du PPD

Le pré projet de diplôme ne durant que 14 semaines avec 6 périodes par semaine, voici en résumé comment elles seront utilisées :

Quoi	Temps
Analyse	2 semaines
Choix du langage	1 semaine
Test d'applications	3 semaines
Analyse de l'architecture globale du projet.	2 semaines
Formation J2ME	2 semaines
Choix de l'environnement de développement	1 semaine
Développement d'une première application avec géo localisation	2 semaines
Rapport	1 semaine

9.4 Journal de travail

Le pré-projet de diplôme se passe du 21 février au 29 mai à raison d'une après-midi par semaine le jeudi. Voici en résumé ce qui a été fait durant ces après-midi.

Date	Quoi
21 février	<ul style="list-style-type: none">- Début du pré projet de diplôme- Rencontre avec M. Robert- Recherche d'exemple dans le domaine du projet, plus ou moins proche
28 février	<ul style="list-style-type: none">- Rencontre avec M. Rodellar et M. Forchellet- Définition du projet
6 mars	<ul style="list-style-type: none">- Première analyse du projet- Recherche de documentation globale
13 mars	<ul style="list-style-type: none">- Analyse des types de langages- Recherche d'applications testables sur les plateformes j2me et C Sharp
20 mars	<ul style="list-style-type: none">- Recherche d'applications à tester- Test de l'API Location
3 avril	<ul style="list-style-type: none">- Use cases
10 avril	<ul style="list-style-type: none">- Gestion des groupes et des utilisateurs
17 avril	<ul style="list-style-type: none">- BD- Restrictions
24 avril	<ul style="list-style-type: none">- Réunion chez Swisscom de Daniel Rodellar et discussion des points suivants :- Restriction sur la plateforme de déploiement- Localisation : GPS et Antennes GPS- Acquisition, envoi et des données de localisation- Gestion des users online/offline
8 mai	<ul style="list-style-type: none">- Mise à jour du rapport- Création du fichier d'avancement demandé par Daniel Rodellar
15 mai	VISITE DDM + Projet PEP donc peu avancé
22 mai	<ul style="list-style-type: none">- Flux de données- Technique de localisation
29 mai	<ul style="list-style-type: none">- Développement de l'application démo- Ecriture rapport final
3 juin	<ul style="list-style-type: none">- Rendu du pré-projet de diplôme