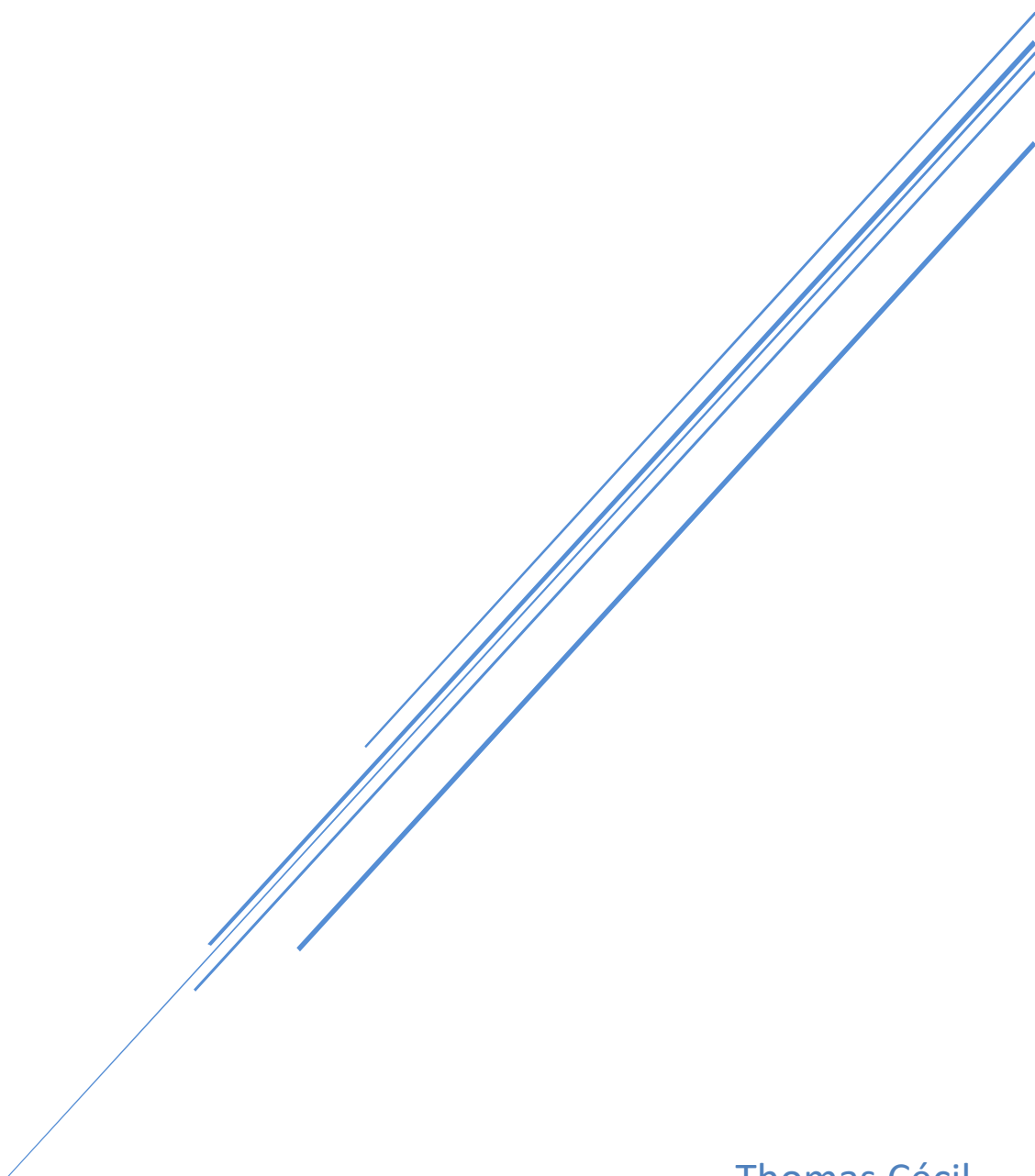


RAPPORT: CRÉATION D'APPLICATION

Not This Corner



Thomas Cécil

« La nuit il fait nuit et on n'y voit rien ! »

-Un anonyme on ne sait où

Sommaire

| | | |
|------|--|----|
| I. | Présentation Générale du projet..... | 4 |
| A. | Pitch..... | 4 |
| B. | Besoin et contexte..... | 4 |
| C. | Description | 4 |
| D. | Cibles | 4 |
| E. | Fonctionnalités principales..... | 5 |
| II. | Exigence fonctionnelles..... | 5 |
| A. | Création du profil | 5 |
| B. | Enregistrement d'itinéraire | 6 |
| C. | La localisation | 6 |
| D. | Report d'agression | 7 |
| E. | ZONE..... | 8 |
| III. | Développement..... | 8 |
| A. | Système d'exploitation choisie..... | 8 |
| B. | Technologie utilisé..... | 8 |
| IV. | Déroulement du Projet..... | 9 |
| A. | Conception | 9 |
| B. | Cycle de vie..... | 9 |
| 1. | Analyses des risques/ juridique..... | 10 |
| 2. | Conception de l'architecture global de l'application | 10 |
| 3. | Conception Détaillé | 11 |
| 4. | Tests Unitaire..... | 12 |
| 5. | Tests d'intégration..... | 12 |
| 6. | Validation | 12 |

I. Présentation Générale du projet

A. Pitch

Un GPS mobile qui affiche des zones dites « peu fréquentable » et un itinéraire plus « sûr » pour rentrer chez soi.

B. Besoin et contexte

On s'est tous déjà retourné d'un mouvement brusque lorsqu'on rentrait de soirée à 3h du matin. On a tous une amie qui nous a raconté qu'un individu louche avait l'air de la suivre le soir en rentrant du travail. Peu importe l'époque la nuit n'a jamais été clémente avec le citoyen lambda. C'est pourquoi j'ai eu l'idée d'une application mobile affichant les zones peu fréquentables où il y'aurait eu des agressions et proposant un itinéraire plus sûr pour rentrer chez soi.

C. Description

Map :

La map affiche des zones de couleur nuancé allant du jaune au rouge vif. L'utilisateur indique son adresse d'arrivée, pour le point de départ on utilisera la géolocalisation et le GPS lui trouve un itinéraire qui évite le plus de zone possible. Il sera bien sur possible de fixer un seuil de «tolérance », permettant à l'utilisateur de traverser ces zones, s'il souhaite éviter de faire un grand détour.

L'affichage des zones sera dépendant de l'heure et du jour, le but est de montrer « à quelle heure fréquenter tel ou tel rue peu devenir « dangereux ».

Report :

L'utilisateur à la possibilité de reporté un évènement lié à une « agression » via un formulaire. Dans ce formulaire il sélectionnera d'abord le lieu sur une carte, l'heure, il indiquera la gravité : le type « d'agressions » (insulte, vol, vol avec violence, agression physique etc.).

En fonction de la fréquence et de la gravité des agressions signalée dans une même ZONE et une même heure, on affichera la ZONE via une couleur sur la MAP.

Connexion :

Dès sa première connexion l'utilisateur aura la possibilité de créer un profil, il va devoir rentrer, la ville dans laquelle il habite, travail, son âge et son sexe, ces informations serviront à rendre pertinent les reports. De plus il aura la possibilité d'enregistrer des itinéraires.

D. Cibles

Tous les détenteurs de smartphones et toutes les personnes disposant d'un abonnement réseau 3 G. Pour la V1, nous ciblerons exclusivement la population parisienne et particulièrement la population intra mural. La cible est technophile, elle utilise son mobile plus d'une heure par jour, se déplace principalement à pied ou en transport en commun, et représente principalement les personnes de âgés de 15 ans au moins.

E. Fonctionnalités principales

- Création d'un profil stocker sur le téléphone (sans adresse e-mail)
- Personnalisation du profil : âge, sexe, lieu de travail(ville) , lieu de résidence(ville), possibilité d'enregistrer les itinéraires emprunter : itinéraire de base (sans intervention de l'application) et itinéraire « sûr ».
- Base de données regroupant les lieux publics (bars, clubs, restaurant, cinéma...), ainsi que divers évènement (concert, fête local, évènement sportif ...)
- Une navigation par géolocalisation, ainsi qu'un affichage de la maps délimité en « Zone » et lieux (Google Maps).
- Calcul de l'itinéraire en fonction du lieu, du niveau de « fréquentable » des zones, du mode de transport, de l'heure où la personne se trouve et où elle doit se rendre
- Des fiches complètes avec toutes les informations sur les « ZONES » lieux et évènements : Description des agressions.

Possibilité de reporter une agression et d'écrire un avis détaillé sur le lieux l'évènements : lieu sur une carte, l'heure, il indiquera la gravité : le type d'agressions (insulte, vol, vol avec violence, agression physique etc.).

- Possibilité d'activer une alerte si une zone dépasse un certain seuil de « fréquentabilité » et qu'elle se trouve sur un itinéraire emprunter par l'utilisateur

II. Exigence fonctionnelles

A. Création du profil

Pour la première version de l'application, la création d'un profil n'est pas obligatoire mais fortement recommandé.

Dès le premier lancement de l'application, l'utilisateur va être confronté à sa première interaction avec l'interface. Celle-ci lui proposera de créer un profil ou d'afficher la MAPS.

Au vu des informations très sensible (itinéraire) que l'utilisateur enregistrera, il devient nécessaire de se poser des questions au niveau de la sécurité. Nous partirons dans un premier temps du principe qu'il vaut mieux enregistrer ces données directement sur le téléphone de l'utilisateur, cela permettra d'instaurer un climat de confiance entre l'utilisateur et « l'application ».

Nous mènerons par la suite une étude de faisabilité :

- Est-t-il plus sûr pour un utilisateur de stocker ces données sur un serveur :
 - Avantage : On a une garantie quant à la sécurité qui protège les données de l'utilisateur.
 - Inconvénient : L'intégrité de l'application sera remise en question.
- Ou vaut-il mieux tout stocker sur le téléphone :
 - Avantage : L'utilisateur sera le seul à posséder ses données
 - Inconvénient : L'utilisateur devra s'assurer de la sécurité mise en place sur son téléphone.

Pour le moment nous offrons 1 seul possibilité de création de profil, avec le strict minimum pour laisser le temps à l'utilisateur de se familiarisé avec l'application et de ne pas se lasser :

- Âge
- Sexe
- Lieu de travail (optionnel)
- Lieu de résidence (ville)

B. Enregistrement d'itinéraire

Pour compléter son profil l'utilisateur aura la possibilité d'enregistrer plusieurs itinéraires. Pour ce faire il accèdera à la MAPS Google et tracera son itinéraire. Une fois l'itinéraire tracé il aura la possibilité d'ajouter un seuil de tolérance vis-à-vis des ZONES « peu fréquentable », ce seuil aura pour but de toujours lui proposer un itinéraire selon son besoin, ainsi qu'un type de transport. Il pourra par la suite l'enregistrer.

Il aura besoin dans un premier temps de choisir son mode d'itinéraire :

- géolocalisation ou adresse pour le point de départ
- adresse d'arrivée
- le type de transport
- Niveau de tolérance (optionnel)

Après validation de ces critères de sélection, le tracé est calculé par un algorithme. Ce calcul prendra en compte tous les critères sélectionnés précédemment. Il s'agit donc de créer un algorithme qui calcule le tracé en intégrant particulièrement les préférences des utilisateurs, et surtout **son seuil de « Tolérance »**.

C. La localisation

Tous les smartphone ou presque sont équipés d'une technologie GPS. La géolocalisation est ainsi devenue un élément du quotidien qu'on retrouve dans énormément d'applications. On trouve ainsi plusieurs API liées au concept de localisation :

La localisation par technologie GPS : permettant de localiser la personne. Il s'agit d'une fonction algorithmique (liée à la recherche) à intégrer dans le développement de l'application. L'affichage des cartes : Pour cela, on va passer par **l'API Google Maps**. Elle n'est pas intégrée à Android, mais appartient à une extension appelée « **Google APIs** ». Pour pouvoir utiliser **Google Maps**, il vous faudra demander l'autorisation pour accéder aux services sur internet. On doit demander une clé. (<https://developers.google.com/maps/documentation/android/>).

Les fonctionnalités de la carte et la géolocalisation pour l'application :

- Connaître la dernière position connue de l'utilisateur.
- Tracer l'itinéraire calculé
- Permet d'être informé quand on s'approche d'un endroit ou qu'on s'en éloigne.
- Fonction Zoom
- Se déplacer dans la MAP
- Utiliser les différents modes d'affichage : satellite / plan
- Afficher la position actuelle
- Intégrer les lieux et flux d'évènement et les fiches informative

La MAP sera composée de :

- D'une carte
- De l'itinéraire

- Des ZONES indiquant un niveau de « fréquentabilité »

D. Report d'agression

Ici on entre dans le vif du sujet, le but initial de cette application sera de pouvoir indiquer en temps réel où et quand une agression a-t-elle eu lieu. Pour ce faire l'utilisateur devra passer par un formulaire puis devra indiquer sur la MAP où l'agression a-t-elle eu lieu. Seuls les utilisateurs **possédant un profil** pourront reporter une agression, le but étant d'apporter de la **crédibilité** aux témoignages via les **informations contenue dans le profil**. Le lieu sera indiqué sur la MAP, via un point.

Le formulaire sera composé de :

- Date
- Evènement particulier (optionnel)
- Type (vol, agression physique, agression verbale etc...) → on définira un classement de « gravité » de l'incident 1 à 5
- Description sur le lieu de l'agression
- L'utilisateur cliquera sur un boutons « LIEU » lui permettant d'indiquer sur la MAP où se situe le lieu de l'agression, on enregistrera les coordonnées GPS du lieu dans une base de donnée. Un lieu peut être un **point indiqué par l'utilisateur** ou un lieu figurant dans **la base de données** Google (restaurant, bars etc...).

Chaque report sera voué à disparaître au bout de 3 mois, et cela pour éviter une surcharge de report ainsi qu'une obsolescence des informations.

E. ZONE

On affichera la latitude et la longitude sur la MAP. Une ZONE sera délimitée par 500 mètre de longueur et de largeur. Elle sera caractérisé par une couleur nuancé allant du jaune au rouge tout en gardé la possibilité qu'elle reste « neutre » (affichage par défaut, sans filtre de couleur). Pour fournir des informations pertinentes on devra tenir compte du **nombre d'habitant** de la ZONE.

En fonction du nombre de Report d'agression et de la gravité de ces report, on sera en mesure de lui attribuer une couleur. A chaque nouveau report un algorithme calculera le ratio entre le nombre d'agression, leurs gravité ainsi que le nombre d'habitant de la ZONE et sera en mesure de fournir **un score** qui servira de base pour afficher une couleur.

Une ZONE est composé de :

- Lieux
- D'habitant
- Report

III. Développement

A. Système d'exploitation choisie

Nous choisissons de développer l'application sous Android. Android utilise des bibliothèques open source puissantes.

Android est gratuit ou presque, autant pour vous que pour les constructeurs. Poster vos applications sur le Play Store, il vous en coûtera la modique somme d'environ 20 euros. Ces 20 euros permettent de publier autant d'applications que vous le souhaitez et éternellement. Facile à développer : toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès.

Il existe quelque contrainte cependant :

- L'application ne doit pas demandé au système beaucoup de ressource en terme de mémoire vive.
- La multitude d'appareil sous Android complexifie la présence d'interface graphique différente pour chaque téléphone (résolution de l'écran). **L'interface de l'application doit s'adapter a toute les résolutions d'écran.**

B. Technologie utilisé

Les applications Android sont presque essentiellement codées en Java. De plus, il faut connaître le langage SQL pour répondre aux requêtes.

Pour pouvoir développer l'application et la tester, il faudra se munir de différent logiciel:

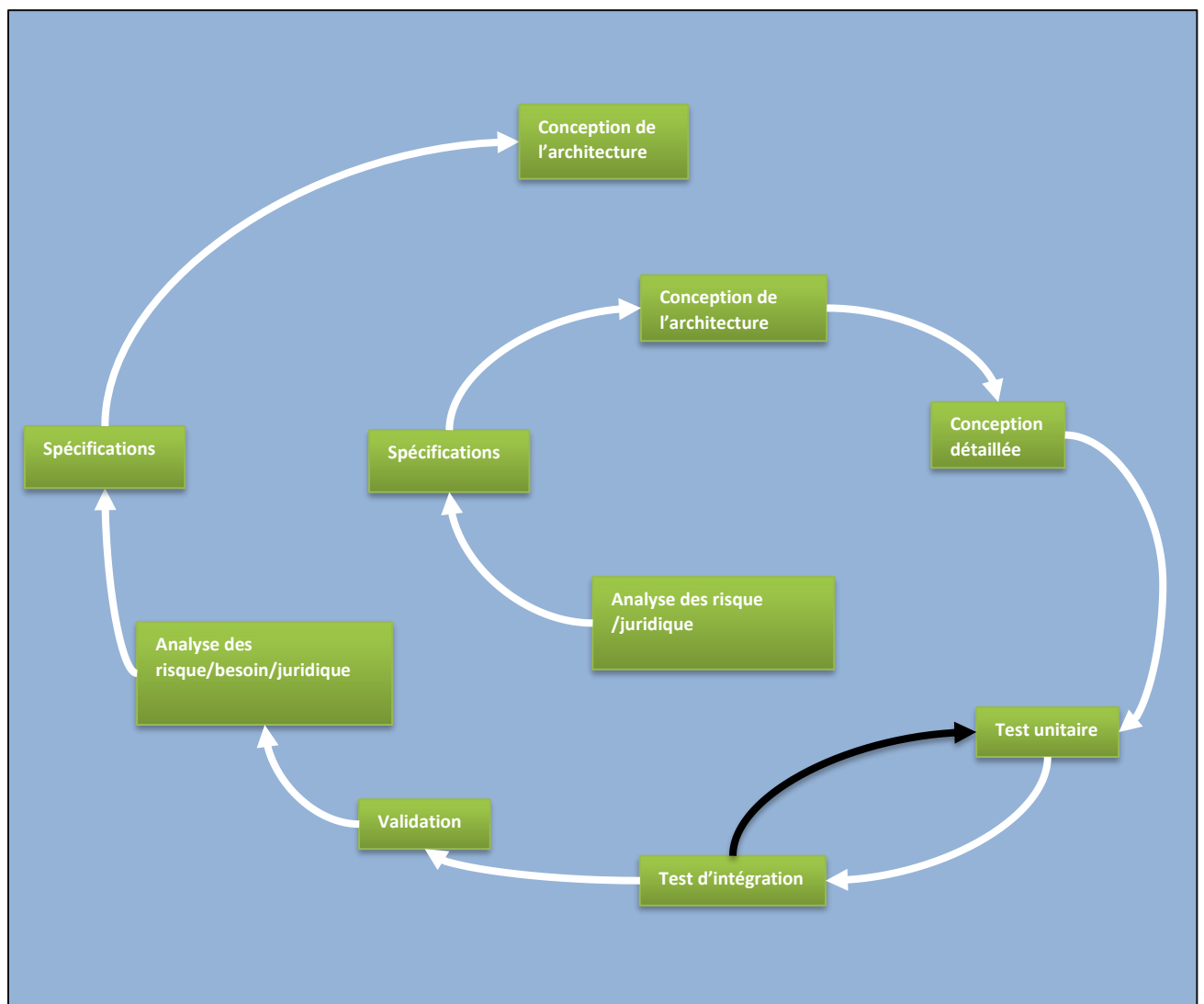
- Le Java développement Kit : permet de développer en Java
- L'IDE Eclipse/Netbeans : permet de facilité le développement
- Un émulateur pour simuler des tests sur le téléphone

IV. Déroulement du Projet

A. Conception

Cette phase se compose de séance de travail. On abordera toutes les étapes du cycle de vie de l'application, où on analysera les détails du cahier des charges, le besoin exprimer etc... Cette phase marque la validation de l'architecture, de l'ergonomie, de l'algorithme et des fonctionnalités.

B. Cycle de vie



On choisit un cycle de vie en spirale car une application mobile est vouée à évoluer sans cesse, il est primordial de pouvoir compléter le produit au fur et à mesure et proposer un produit de plus en plus complet.

1. Analyses des risques/ juridique

Risque :

Avant de faire de se lancer dans le développement d'une application il est nécessaire au préalable de faire une analyse des risques.

Cela nous permettra de définir un plan d'actions correctrices et préventives visant à protéger le développement des différent problème pouvant être rencontré au cours du projet :

| Risques | Probabilité(P) | Gravité(G) | Criticité (P*G) | Action préventive/corrective |
|---|----------------|------------|-----------------|--|
| Cahier des charges incomplet | 3 | 5 | 15 | Finaliser le cahier des charges. |
| Difficultés matérielle | 1 | 3 | 3 | Etablir une date limite pour chaque étape du cycle de vie. |
| Dépassement du délai | 3 | 4 | 12 | Etablir une date limite pour chaque étape du cycle de vie. |
| Sous-estimation de l'intégration | 3 | 4 | 12 | |

On estime qu'un cahier des charges incomplet est ce qui a le plus de chance de freiner l'avancement du projet. On y tournera dans un premier temps toute notre attention afin de contrer ce risque ;

Juridique :

Au vu des informations sensible qu'utilise l'application il est obligatoire de consulter un expert juridique. En effet les itinéraires des utilisateurs sont un des aspects les plus sensible qu'il faudra traiter et il sera donc primordiale de prendre des précautions et bien s'informer sur le sujet.

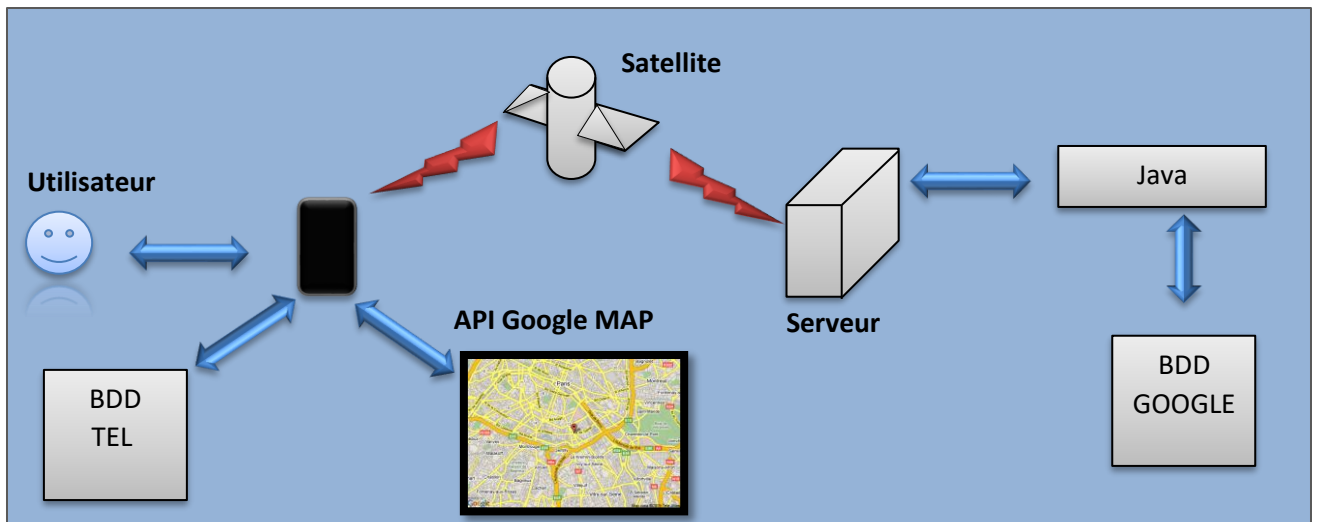
De plus on peut imaginer dans un futurs pas si lointain que l'application soit utiliser à des fin politiques ce qui n'est bien sûr pas souhaité.

2. Conception de l'architecture global de l'application

On définira le « schéma » de communication entre l'utilisateur, l'application, et le serveur.

On va définir le nombre de classes nécessaire pour créer un premier prototype de l'application.

Ceci nous permettra d'avoir une vision globale de l'architecture de l'application.

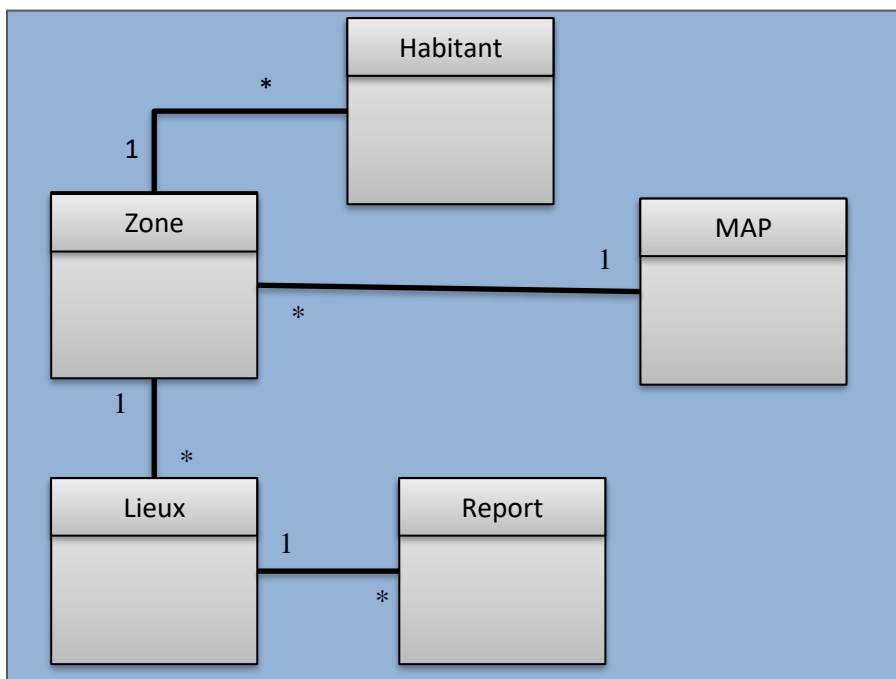


3. Conception Détaillé

Ici on va s'intéresser à chaque sous partie de l'architecture :

- Comment les données stockées sur le téléphone de l'utilisateur pourront elle communiquer avec l'application ;
- Quelle sont les classes java qui composent l'application
- Comment interagir avec la base de données Google (requête SQL)
- Le but étant de pouvoir créer une architecture ne permettant pas la suite une implémentation simple de nouveau module.

Voici un début de diagramme de classe pour définir une ZONE



4. Tests Unitaire

On testera ici chaque nouveau module séparément. Le but est de pouvoir ajouter du contenu en minimisant l'impact sur l'ensemble de l'application et ainsi pouvoir intervenir en local en cas de défaillance technique.

Ainsi on saura si un module est défaillant.

5. Tests d'intégration

Comme dit précédemment on ajoutera le ou les nouveaux modules à l'ensemble des modules déjà existant.

On testera que l'ensemble des modules fonctionnent bien et n'ont aucun problème à communiquer entre eux.

Si l'intégration du nouveau module entraîne des erreurs, on procédera à un retour aux tests unitaire.

6. Validation

Une fois tous le test effectué et compléter on sera en mesure d'effectuer une mise à jour en prod, et fournir ainsi à l'utilisateur un contenu plus complet.