

# Rapport d'analyse

## CONCEPTION D'UNE APPLICATION ANDROID: Itinéraire à Vol d'oiseau



**Réalisé par:**

EL MOUTAOUKKIL Amina

EDOO Omran

BAALACHE Faycal

BELADRAOUI Chaimaa

**Encadré par:**

HANGOUËT Jean-François

DE OLIVEIRA Vincent

<b>Introduction</b>	<b>3</b>
<b>Contexte du projet</b>	<b>3</b>
<b>Objectifs de l'étude</b>	<b>4</b>
<b>Analyse fonctionnelle</b>	<b>5</b>
<b>Etude technique</b>	<b>10</b>
<b>Réalisation et suivi de projet</b>	<b>11</b>
Les risques:	11
<b>Conclusion</b>	<b>13</b>

# I. Introduction

Dans le cadre de notre seconde année du cycle ingénieur à l'Ecole Nationale des Sciences Géographiques, nous réalisons le projet Geodev2, s'étalant sur 3 mois, nous permettant de mettre en pratique nos connaissances et nos compétences acquises au travers la conception et le développement d'une application mobile. Ayant une passion commune pour la Géodésie ainsi que pour le développement informatique, notre groupe composé de Omran EDOO, Amina EL MOUTAOUAKKIL, Chaimaa BELADRAOUI et Fayçal BAALACHE, a saisi l'opportunité d'utiliser cet intérêt commun afin de soumettre un projet abouti à nos commanditaires Jean-François HANGOUËT ainsi que Vincent DE OLIVEIRA.

## II. Contexte du projets

Notre projet Geodev2, intitulé « Itinéraire à vol d'oiseau » a pour but la création d'une application mobile spécifique. Nommée « CityCross », notre application mêle Géodésie et Développement.

Durant ce projet, nous sommes accompagnés par Jean-François HANGOUËT ainsi que Vincent DE OLIVEIRA, qui sont 2 enseignants de l'Ecole Nationale des Sciences Géographiques.

Nous sommes les premiers à prendre en main la réalisation de cette application, nous laisserons donc une trace écrite et de la documentation une fois le projet terminé afin qu'il puisse être continué.

### Pourquoi ?

La géomatique ainsi que certaines notions de sciences géographiques paraissent encore inconnues au grand public à l'heure d'aujourd'hui.

Cette application peut alors apparaître comme un vecteur d'information afin d'en apprendre plus sur les sciences géographiques de façon ludique. En effet, en connaissant la direction ainsi que la distance de l'utilisateur aux villes du monde, ceci permet de mieux comprendre la notion de courbure de la terre, de mieux se situer sur le globe, d'assimiler l'influence du choix d'un ellipsoïde.

Étant conscient que le terme ellipsoïde est assez spécifique, nous aurons pris soin de rédiger une page explicative et vulgarisée à destination du lecteur.

De plus, il sera aussi possible de l'utiliser dans un but totalement ludique ou à titre informatif.

Nos commanditaires n'étant pas les représentants d'une entreprise, les enjeux peuvent apparaître moindre. Cependant, ceci ne signifie pas que nous prendrons ce travail à la légère, il n'y aura juste aucune répercussion financière ou autre sur une entreprise.

### **III. Objectifs de l'étude - Reformulation du besoin**

#### **1. Les objectifs de l'étude**

Développer en langage java une application mobile Android comprenant deux interfaces :

- Une interface de calage permettant d'orienter la direction orthogonale du smartphone parallèlement au plan tangent à un ellipsoïde choisi au point de coordonnées géographiques du lieu où se trouve l'utilisateur à l'aide de l'affichage d'une bulle de centrage. L'interface devra alors afficher les villes présentes jusqu'à une certaine distance maximale choisie sur la ligne géodésique qui part du lieu où on se trouve selon l'azimut de l'orientation stabilisée.
- Une interface de guidage permettant de pointer la direction du smartphone vers une ville choisie par l'utilisateur au sens de la plus petite distance géodésique du lieu où on se trouve à l'aide de l'affichage d'une bulle de centrage. Une fois l'orientation enregistrée ou stabilisée, on peut voir les villes identifiées de la même manière que pour l'interface de calage.

L'interface de guidage consiste en des calculs géodésiques de résolution du problème direct, l'interface de calage consiste en la résolution du problème inverse.

L'application nécessite la création d'une base de données « villes du monde » ou « lieux du monde » construite à partir de bases de données existantes.

Les résultats de ces deux interfaces pourront être exportés sous forme de liste (Localisation du smartphone, villes identifiées et leurs distances, azimuth ...) ou sous forme de carte (dans une projection choisie).

## **2. Les contraintes**

La solution proposée doit obligatoirement être une application Android ou Web, et doit être développée avec le langage Java ou Kotlin.

Nous faisons face à une contrainte de moyens dû à l'absence de smartphones Android à notre disposition pour effectuer les tests de notre application.

L'application fonctionnelle doit être livrée au plus tard le 18 mai 2022.

## **3. Les recueils du besoin - Les acteurs**

L'application est à but ludique et pédagogique, et est donc adressée à un public spécialiste autant qu'au grand public comme on peut le voir sur le diagramme de cas d'utilisations.

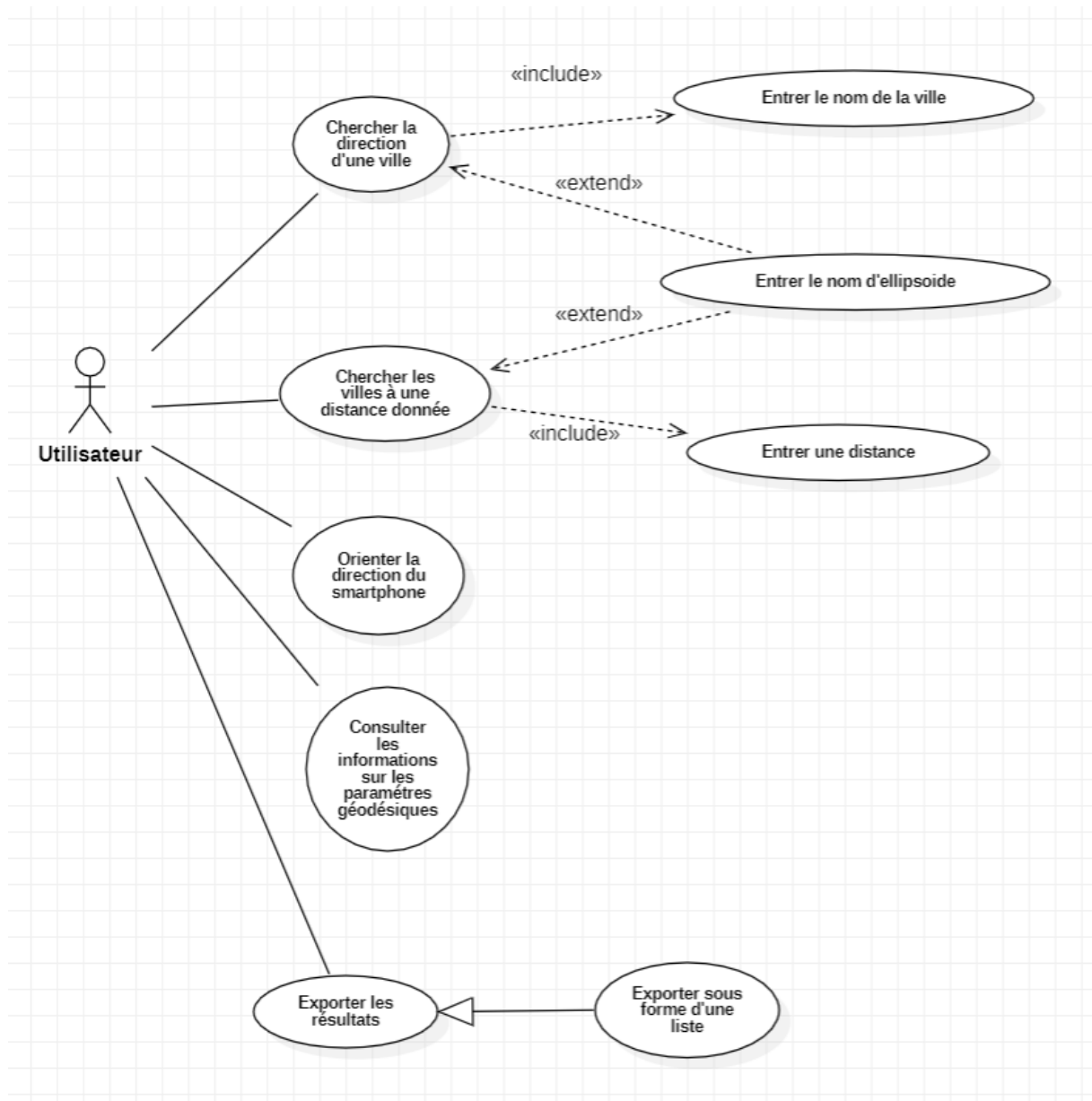


Diagramme de cas d'utilisation

Ainsi, un utilisateur lambda aura accès à des explications sur les différents paramètres géodésiques.

A travers l'interface de guidage, l'utilisateur pourra d'abord caler la direction du smartphone selon une orientation choisie puis visualiser un ensemble de lieux majeurs présents jusqu'à une certaine distance.

A travers l'interface de calage, l'utilisateur pourra choisir un lieu puis caler la direction du smartphone pour visualiser où se situe ce lieu par rapport à sa position (à quelle distance et selon quelle orientation). L'interface de calage hérite de l'interface de guidage pour

permettre également la visualisation de plusieurs lieux importants présents sur la même ligne géodésique.

Ces lieux importants pourront être classés selon la distance à l'utilisateur et livrés sous forme de liste ou de carte.

## IV. Analyse fonctionnelle

Notre application CityCross offre deux grands services à l'utilisateur. Il lui permet de :

1-Se géolocaliser et afficher les villes traversées.

2-Choisir une ville ou un lieu et visualiser les villes traversées

Ces deux grands services nécessitent des fonctionnalités intermédiaires :

- Consulter les informations sur les paramètres géodésiques à choisir.
- Choisir un ellipsoïde
- Se géolocaliser et guider l'orientation du téléphone.
- Visualiser les villes traversées par la ligne géodésique selon un seuil de distance choisi au préalable.
- Afficher la liste des villes classées par pays ou par distance.
- Ajouter des lieux à la liste des villes par défaut.
- Afficher les villes parcourues et leurs distances par rapport à l'emplacement de l'utilisateur.
- Exporter les résultats sous forme d'une liste ou d'une carte.

On peut visualiser les éléments qui composent le système et de leurs relations à partir du diagramme de classe suivant:

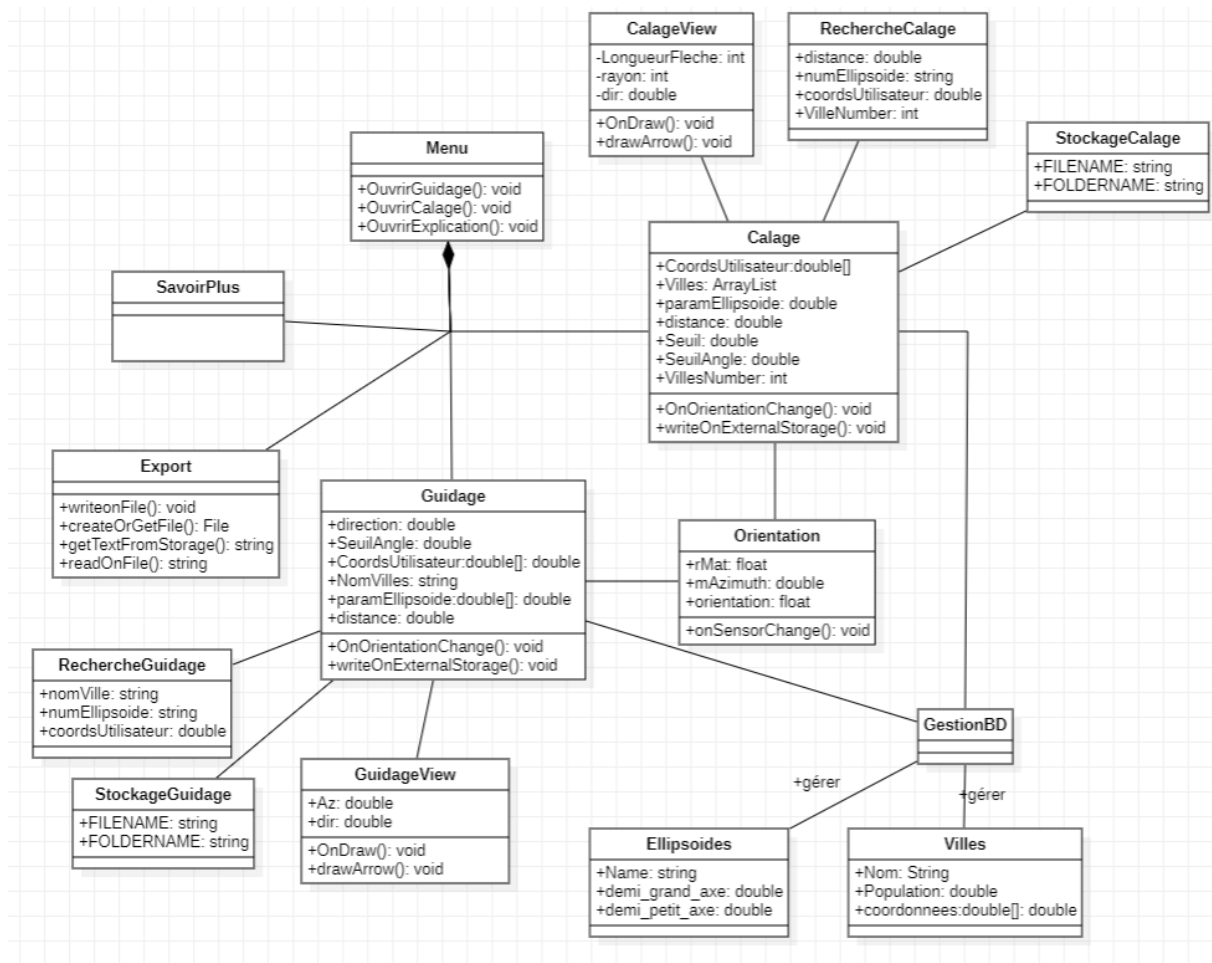
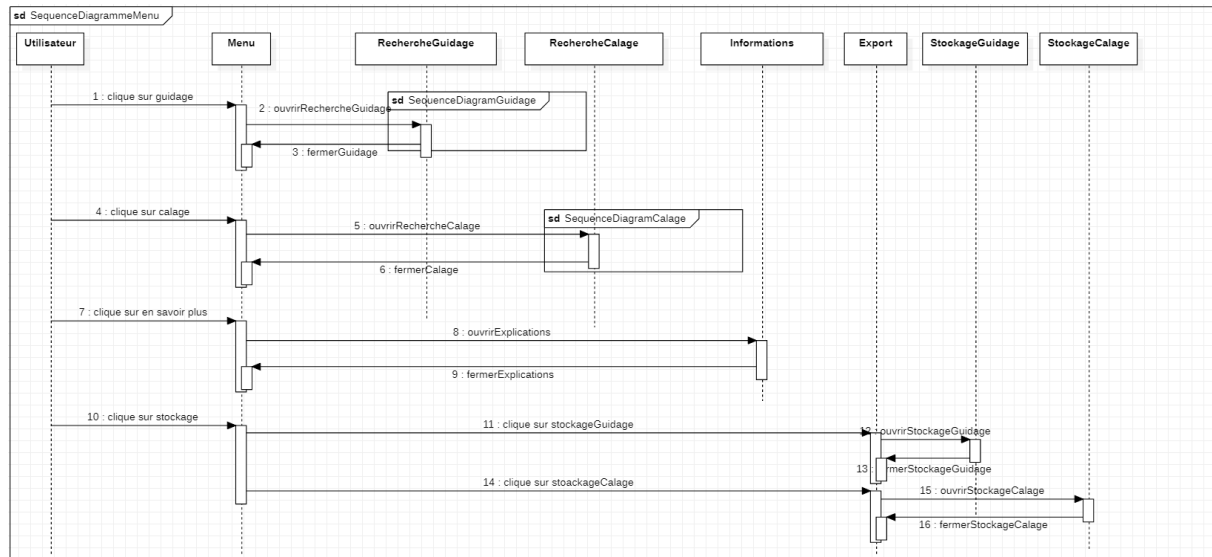


Diagramme de classe

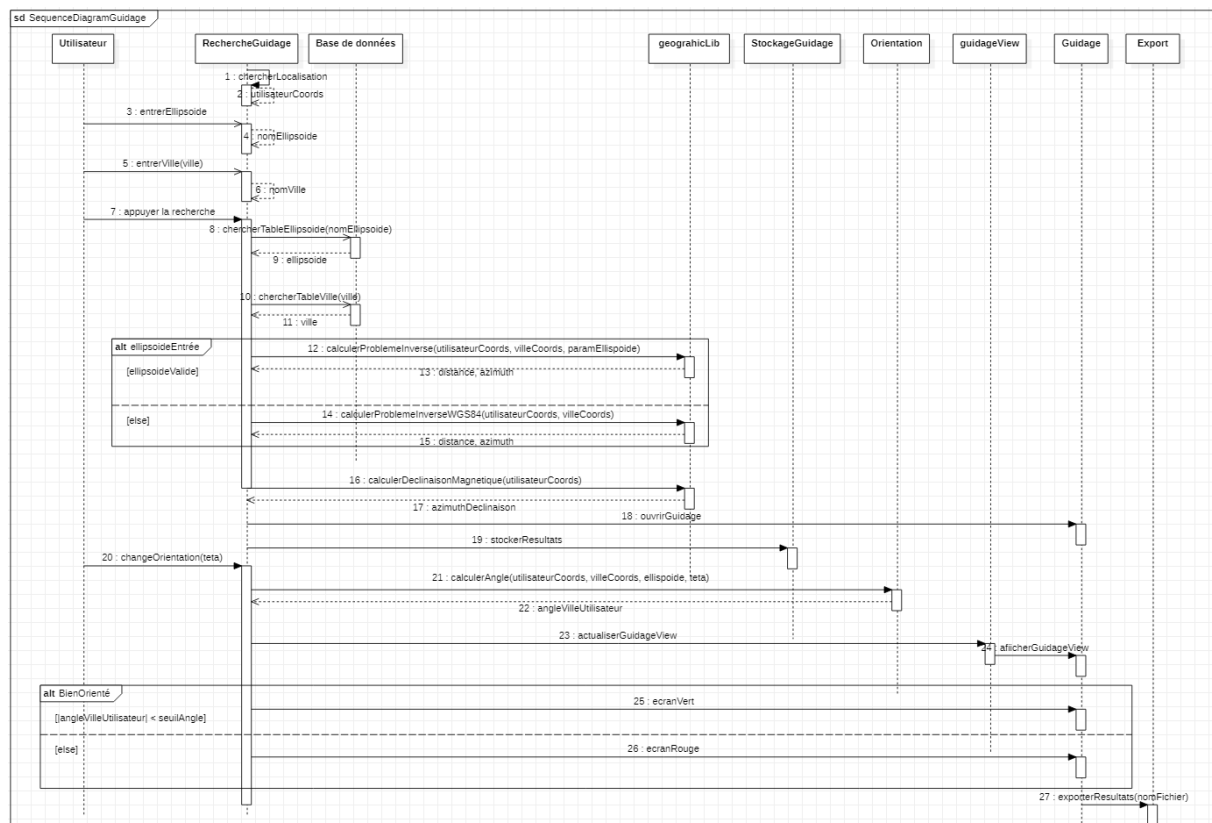
On peut représenter les échanges entre les différents objets et acteurs de ces deux services en fonction de temps grâce au diagramme de séquence.

Comme le montrent les figures suivantes, le diagramme de séquence décrit comment et dans quel ordre plusieurs objets fonctionnent ensemble.

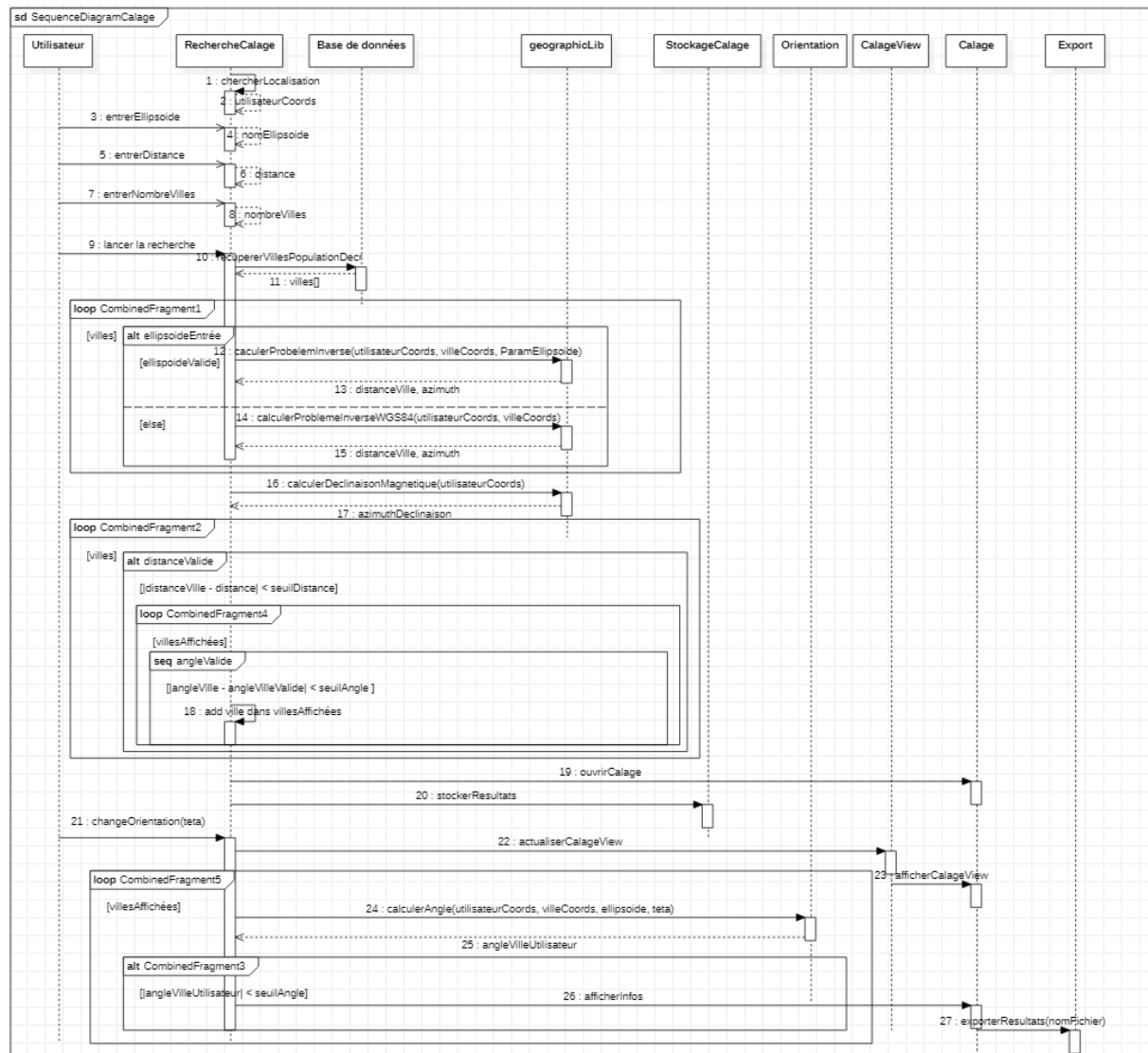




## Service guidage :

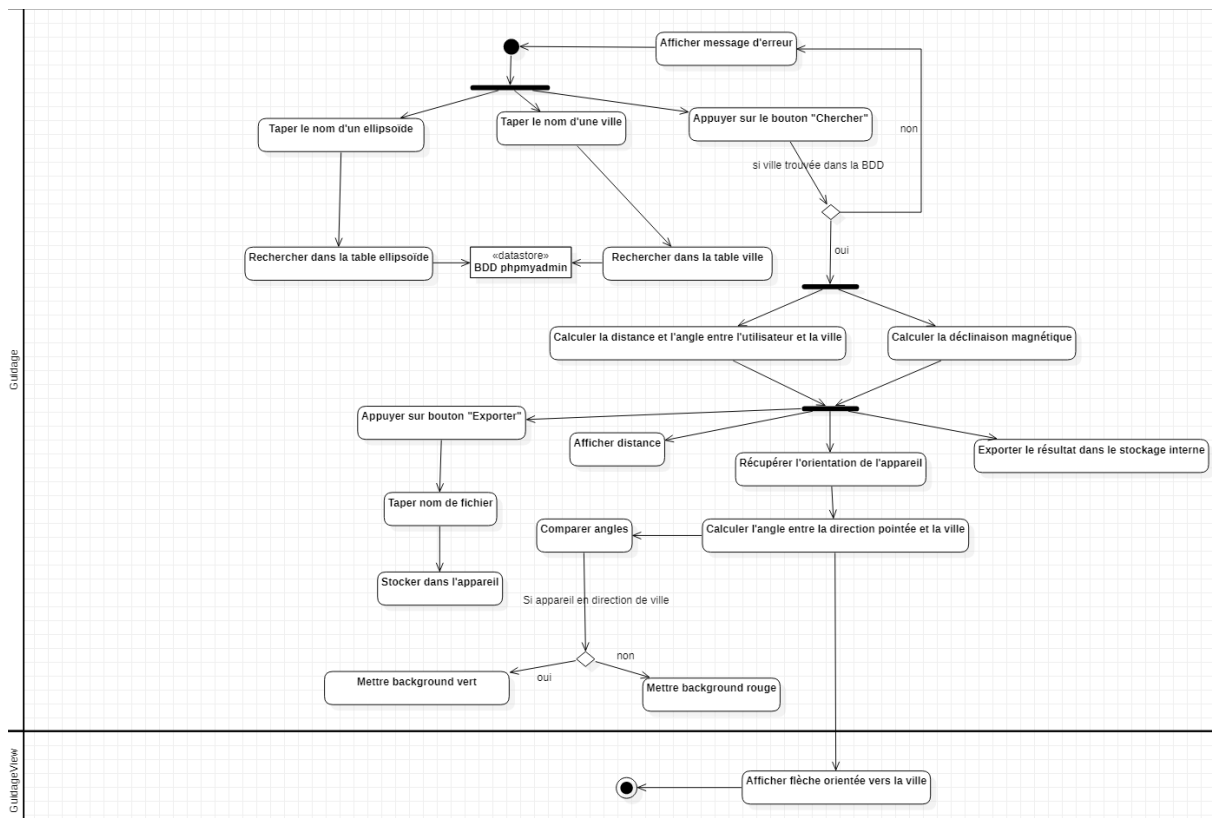
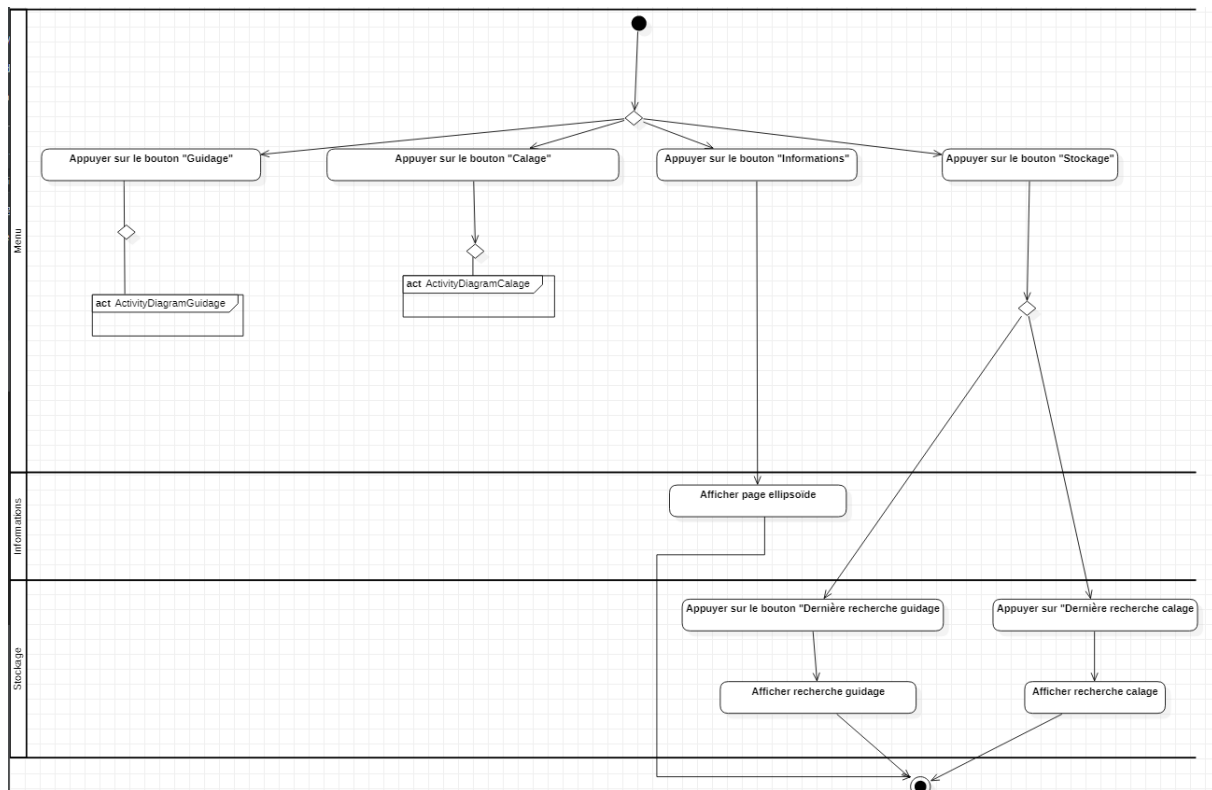


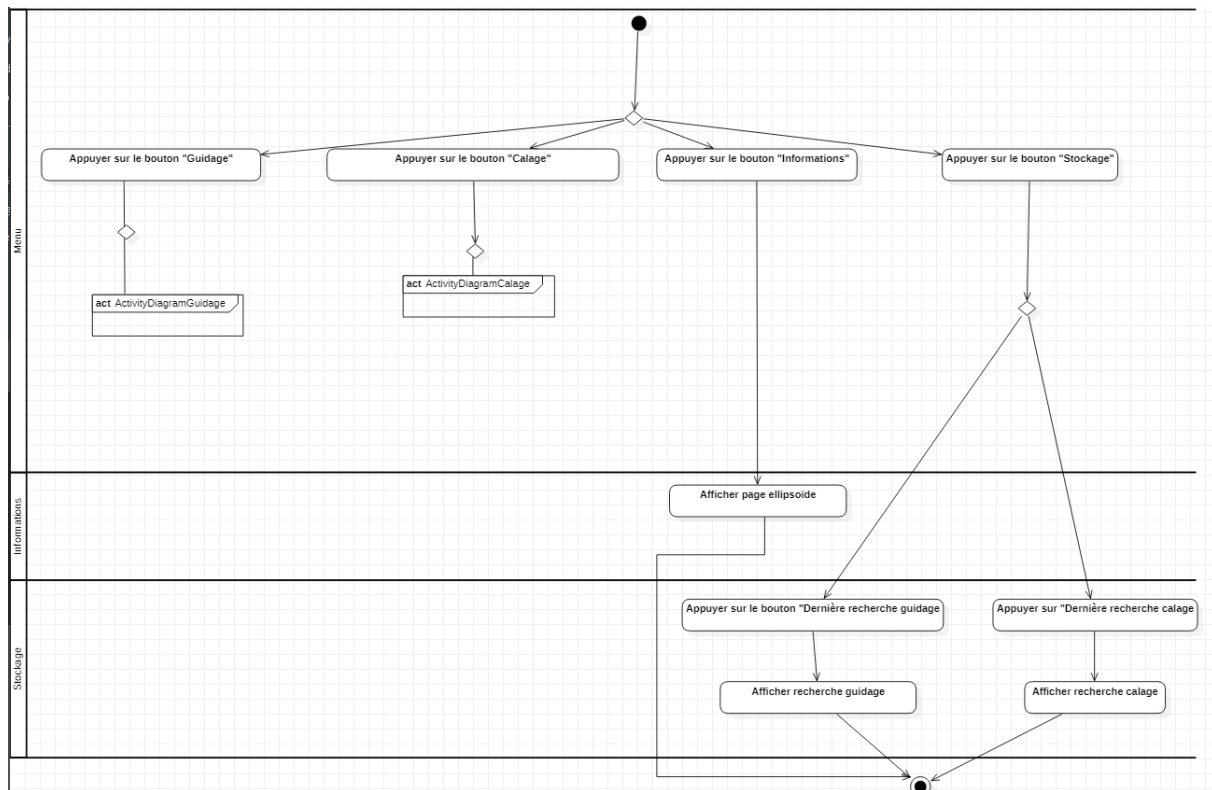
## Service calage:



Pour mettre davantage l'accent sur les traitements, on va opter pour la modélisation du cheminement de flots de contrôle et de flots de données. On va détailler particulièrement le comportement des deux grands services en réalisant le diagramme d'activité.

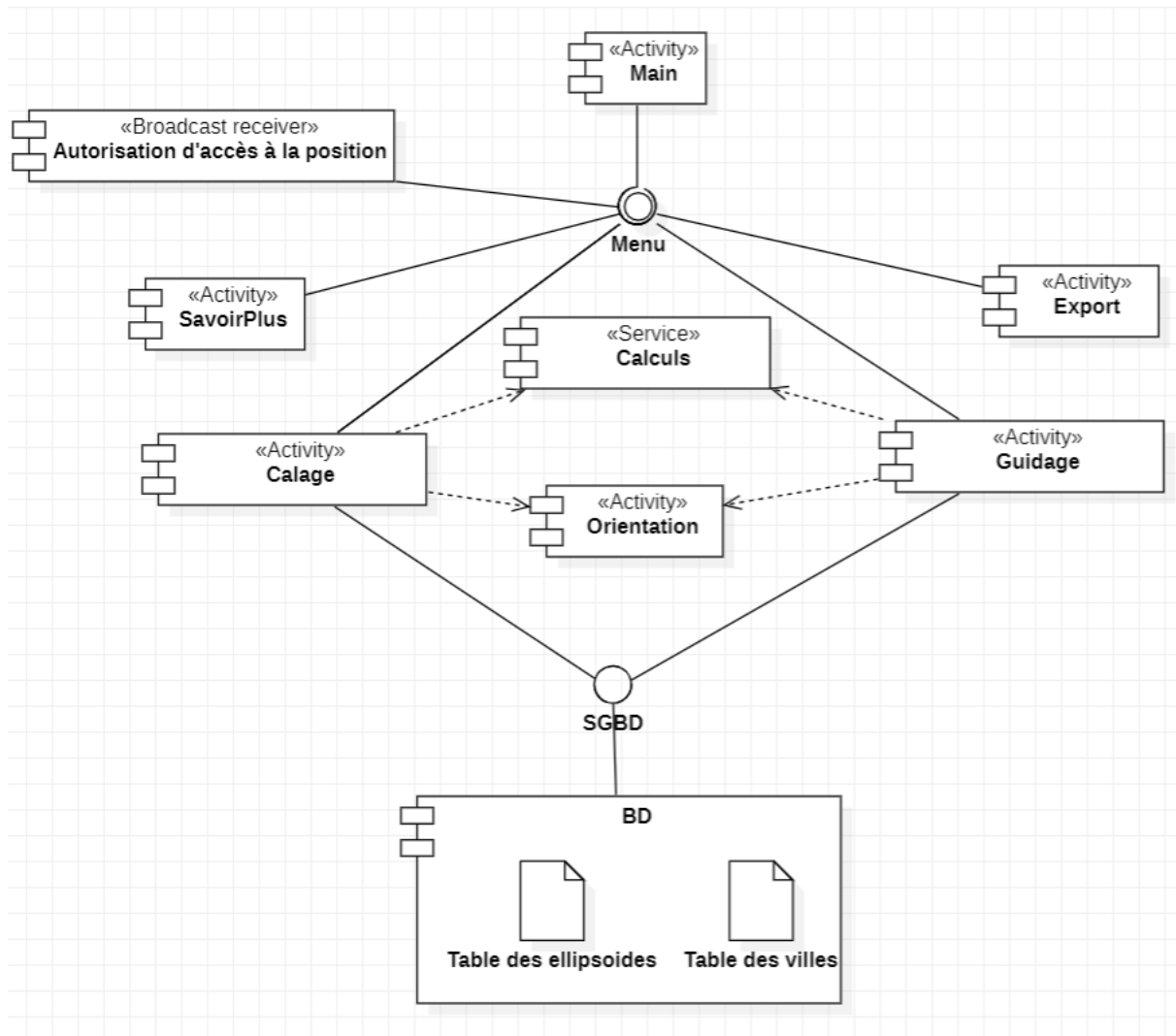
### Diagrammes d'activités:



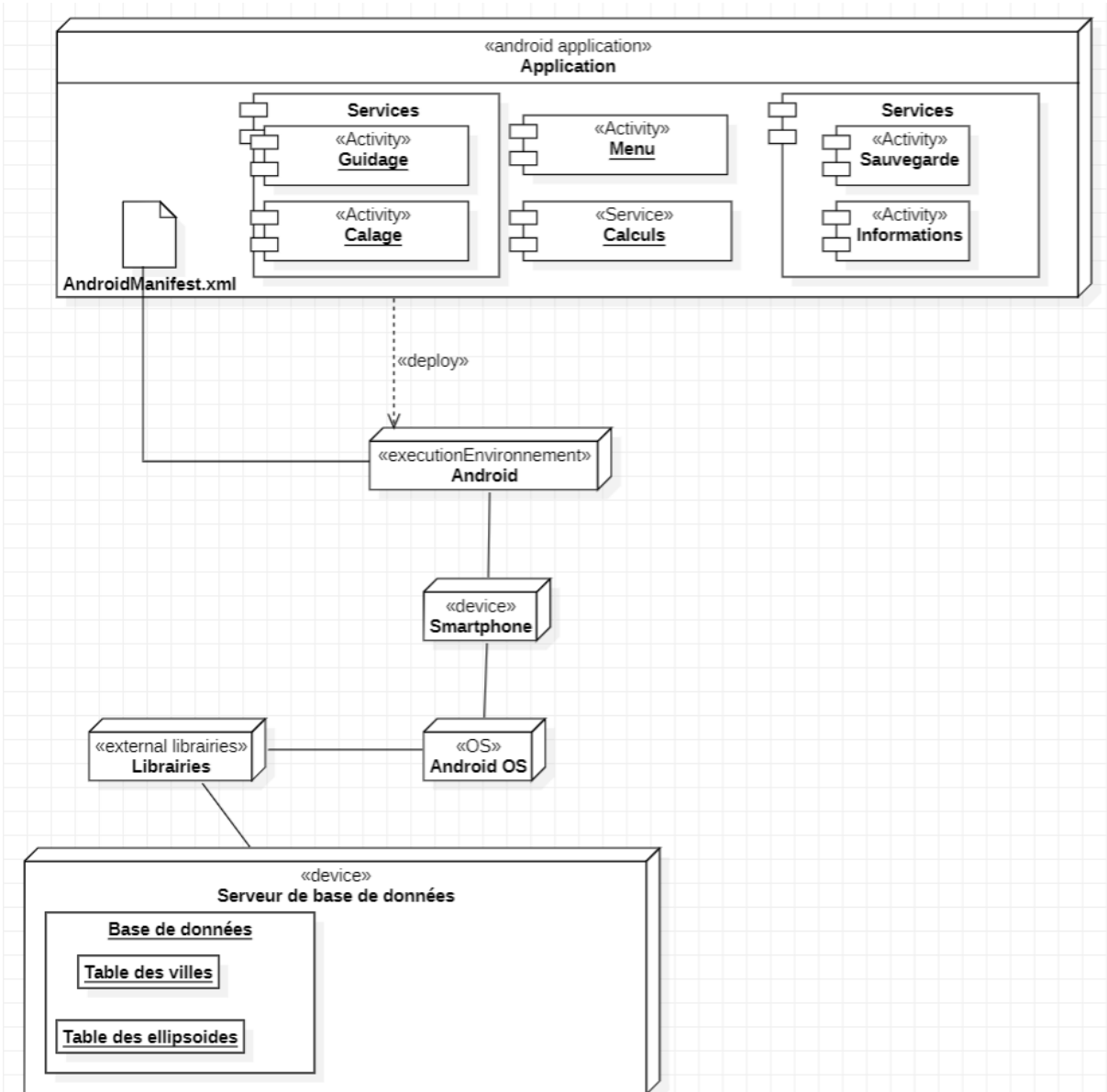


## V. Etude technique

Diagramme de composant:



**Diagramme de déploiement:**



Choix du langage:

	A	B	C
1		<b>Java</b>	<b>Kotlin</b>
2	<b>performances</b>	Assez similaire	
3	<b>syntaxe</b>	syntaxe assez lourde, compliqué	syntaxe plus allégée, courte
4	<b>sécurité null</b>	L'erreur NullPointerExceptions est une erreur fréquente car le langage ne possède pas de sécurité nulle	Kotlin limite cette exception grâce à un système de sécurité
5	<b>pérennité</b>	langage mature donc moins de changements	langage jeune donc fonctions susceptibles d'évoluer
6	<b>classes de données</b>	Plus compliqué à instancier	Plus simple à instancier
7	<b>inférence de type</b>	il faut spécifier les types des variables	pas besoin de spécifier les types des variables
8	<b>communauté, apprentissage</b>	grande communauté, facile de trouver ce que l'on cherche	langage récent donc moins grande communauté, moins de forums
9			
10		Dépend	
11		Avantage	
12		Désavantage	

Il n'y a pas de différences majeures entre les deux langages. Kotlin permet de coder plus rapidement mais java est un peu plus robuste. Il serait intéressant d'utiliser Kotlin afin d'apprendre un nouveau langage de programmation en pleine évolution qui pourrait devenir le langage principale sous android. Cependant, au vu des multiples risques qui ont pour conséquences le manque de temps, nous avons jugé préférable de choisir de coder en Java que nous connaissons déjà.

### Comment accéder à la position sur Android Studio:

Il existe deux moyens permettant d'obtenir la position du smartphone de l'utilisateur :

- le fournisseur de localisation GPS **(1)**
- le fournisseur de localisation réseau **(2)**

Comparaison des deux :

- **(1)** est plus précis que **(2)**
- **(1)** est très lent en intérieur
- **(2)** retourne la position de la tour de téléphonie cellulaire la plus proche, il est donc moins précis que **(1)**

Il semble donc judicieux d'utiliser les deux moyens selon le cas optimal.

Tout d'abord, il faut demander dans le « manifest » la permission d'accès à la position avec l'une des lignes de code suivantes

a) `<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />`

`<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />`.

On peut utiliser uniquement l'une des permission, la précision de COARSE\_LOCATION est faible et celle de FINE\_LOCATION est forte. Cependant, si l'utilisateur n'autorise que l'accès à sa position approximative (ce qui est possible) alors FINE\_LOCATION ne marchera pas. Il est donc conseillé d'utiliser les deux et c'est même obligatoire sur Android 12 ou plus sous peine d'erreur.

b) `<uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />`

Dans le cas où l'on utilise le fournisseur de localisation réseau, il faut également demander la permission d'accès à internet avec

`<uses-permission android:name="android.permission.INTERNET" />`.

Cela n'autorise pas à la place de l'utilisateur l'accès de la position, il devra le paramétrer lui-même. Cela permet juste d'ajouter l'option qui lui permet d'accepter de partager sa position.

Il existe deux types de permission :

- celle du premier plan **(a)** qui permet d'envoyer sa position un nombre limité de fois
- celle de l'arrière plan **(b)** qui permet d'envoyer sa position en continue

On peut utiliser l'API des services google play pour avoir accès à la position. Cela entre dans la

catégorie (1). Pour cela, il faut tout d'abord installer les services google play via le SDK manager. Ensuite il faut ajouter la dépendance à l'API dans le fichier build.gradle avec la ligne :

implementation 'com.google.android.gms:play-services-location:19.0.1'.

On peut alors utiliser le « fused location provider » et la méthode « getLastLocation() » ou « getCurrentLocation() » afin d'obtenir la position. Il ne faut cependant pas oublier de vérifier l'accès les permissions avec un « try catch » ou une syntaxe dédiée.

Pour afficher une fenêtre de demande d'autorisation d'accès à la localisation, on utilise la fonction requestPermissions de ActivityCompat.

### **Comment obtenir les angles de rotation d'un smartphone ?**



Le vecteur de rotation est un capteur virtuel qui dépend de l'accéléromètre et du magnétomètre. Il permet d'obtenir directement l'orientation de l'appareil selon toutes les directions. Cependant, si la grande majorité des appareils possèdent un accéléromètre, beaucoup n'ont pas de magnétomètre et n'ont donc pas accès au vecteur de rotation.

L'accéléromètre permet de mesurer l'accélération de l'appareil dans toutes les directions mais aussi d'évaluer l'orientation de l'appareil dans un repère terrestre.

Le magnétomètre permet de déterminer la direction et la norme du champ magnétique terrestre et peut servir de boussole couplé à l'accéléromètre.

Ainsi un accéléromètre et un magnétomètre suffisent pour obtenir l'orientation de l'appareil dans le repère des points cardinaux.

Pour les machines ne possédant pas de magnétomètre, il ne semble pas exister d'alternatives.

Afin d'utiliser les capteurs on utilise la classe `SensorManager`. On doit implémenter l'interface `SensorEventListener` afin d'écouter les changements des capteurs.

Dans le « manifest » on peut identifier un capteur comme requis pour utiliser l'application, ainsi les utilisateurs n'ayant pas le capteur ne verront pas l'application sur google play.

## VI. Réalisation et suivi de projet

### 4. Les risques:

L'analyse des risques est une étape indispensable de la planification du projet qui nous permet d'identifier les dangers, leurs niveaux de gravité, la prévention de chaque risque ainsi que les actions préventives et correctives à prendre. Cela nous permet d'éviter les pires scénarios en préparant des plans d'atténuation.

On commence par identifier les risques tels que chaque risque présente deux dimensions : sa gravité et sa fréquence.

On peut estimer la fréquence comme suit :






Fréquence	Cotation
Possible	1

<b>Certain</b>	<b>2</b>
<b>Fréquent</b>	<b>3</b>
<b>Très fréquent</b>	<b>4</b>

Ensuite, on peut estimer la gravité d'un risque comme suit :

<b>Gravité</b>	<b>Cotation</b>
<b>Mineur</b>	<b>1</b>
<b>Moyen</b>	<b>2</b>
<b>Majeur</b>	<b>3</b>

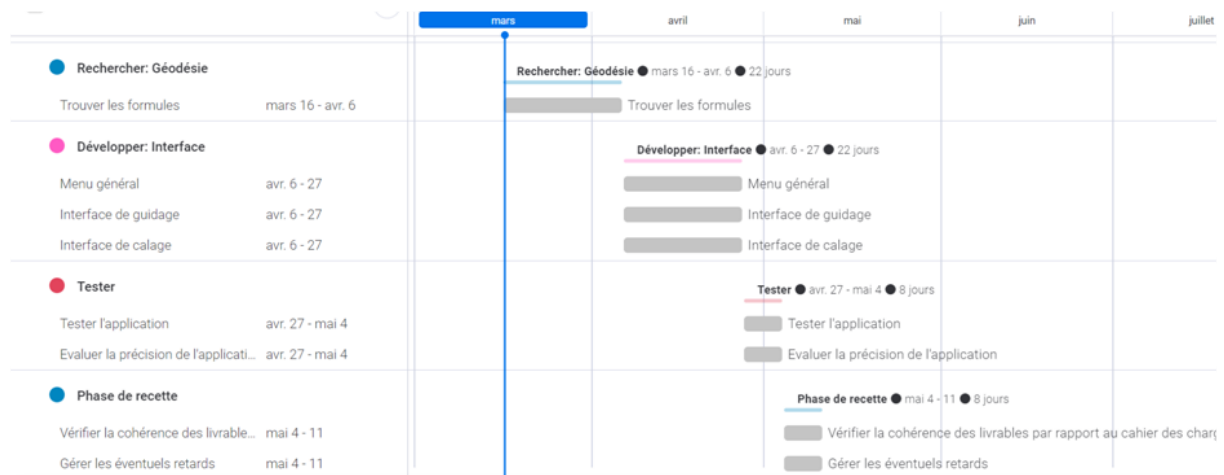
Par la suite, on peut calculer la criticité : **Criticité=gravité\*fréquence**

<b>Intitulé de risque</b>	<b>Fréquence</b>	<b>Cotation</b>	<b>Gravité</b>	<b>Cotation</b>	<b>Criticité</b>	<b>Conséquence</b>	<b>Solution Corrective</b>	<b>Evolution</b>
Application bug pendant la soutenance	Possible	1	Majeur	3	3	Présentation ratée	Faire une vidéo de démonstration	
Manque de disponibilité des membres	Possible	1	Moyen	2	2	Difficulté de Gestion de temps	Bonne gestion des rôles	
Manque du matériel	Possible	1	Moyen	2	2	Difficulté de faire des Tests	Achat/Emprunt	
Mauvaise estimation des délais des livrables	Fréquent	3	Majeur	3	9	Retard du projet	Revoir le diagramme de Gantt régulièrement	
Pas de compromis entre les membres sur le choix d'une fonctionnalité	Possible	1	Moyen	2	2	Gestion du projet difficile	Arbitrage extérieur/Concertation avec commanditaire	

## 5. Le Planning prévisionnel et explications

Nous avons planifié la réalisation des tâches dans le temps à l'aide d'un diagramme de Gantt:





Nous nous sommes répartis les postes suivants:

Chef de Projet : Omran Edoo

Responsable de communication: Faycal Baalache

Gestionnaire de base de données: Amina El Moutaouakkil

Responsables pôle programmation: Chaimaa Beladraoui, Omran Edoo

Responsables pôle Géodésie: Amina El Moutaouakkil, Faycal Baalache

Afin de communiquer notre travail entre membres du groupe et avec les commanditaires, nous utilisons un dossier partagé Google Drive dans lequel nous stockons notamment le cahier de laboratoire et un git pour le code.

Nous avons choisi le nom de CityCross pour notre application car il est appélatif pour le grand public et il indique explicitement l'esprit de découverte des villes et lieux qui entourent l'utilisateur.

Nous avons créé le logo suivant pour représenter notre application:



Ce logo permet d'évoquer la traversée de lieux du monde que propose notre application à l'aide d'orientations représentées par des flèches.

## VII. Conclusion

Pour conclure, lors de cette phase d'analyse, nous nous sommes beaucoup questionnés, notamment sur les méthodes techniques à utiliser (choix langage, fond de carte à utiliser, ...) où chacun a pu proposer sa vision des choses. De notre réflexion est née cette analyse enrichissante et formatrice de notre application que nous avons élaborée. En imaginant ce

projet, nous avons beaucoup d'idées en tête qu'il a fallu structurer concernant l'élaboration de cette application, mais nous ne connaissions pas les processus à utiliser.

En tentant de maîtriser le projet dans sa globalité, nous avons eu un aperçu complet du travail restant à faire ainsi que des différentes contraintes que nous pourrions rencontrer.

D'autre part, notre projet rassemble diverses compétences : programmation, informatique, géodésie, cartographie et d'autres. Cette pluridisciplinarité peut apparaître comme un obstacle que nous pourrions surmonter à l'aide de la polyvalence de notre équipe. Ceci nous a permis de nous répartir les différentes tâches au sein de l'équipe du projet.

Enfin, nous avons essayé d'estimer au mieux le temps de travail restant ainsi que la remise des livrables. Pour finir, nous avons hâte de commencer la partie technique de notre projet afin d'y profiter pleinement.