

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DES MATHÉMATIQUES ET
DE L'INFORMATIQUE

DEPARTEMENT DE L'INFORMATIQUE

N° :.....



DOMAINE : MATHÉMATIQUES ET
INFORMATIQUE

FILIERE : INFORMATIQUE

OPTION : RESEAUX ET TECHNOLOGIE
DE L'INFORMATION ET DE
COMMUNICATION

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par : Boudraf Aissa

Intitulé

Développement d'une application mobile pour
randonneurs

Soutenu devant le jury composé de :

Mouhoub Nacer Eddine

Université de M'sila

Président

Akhrouf Samir

Université de M'sila

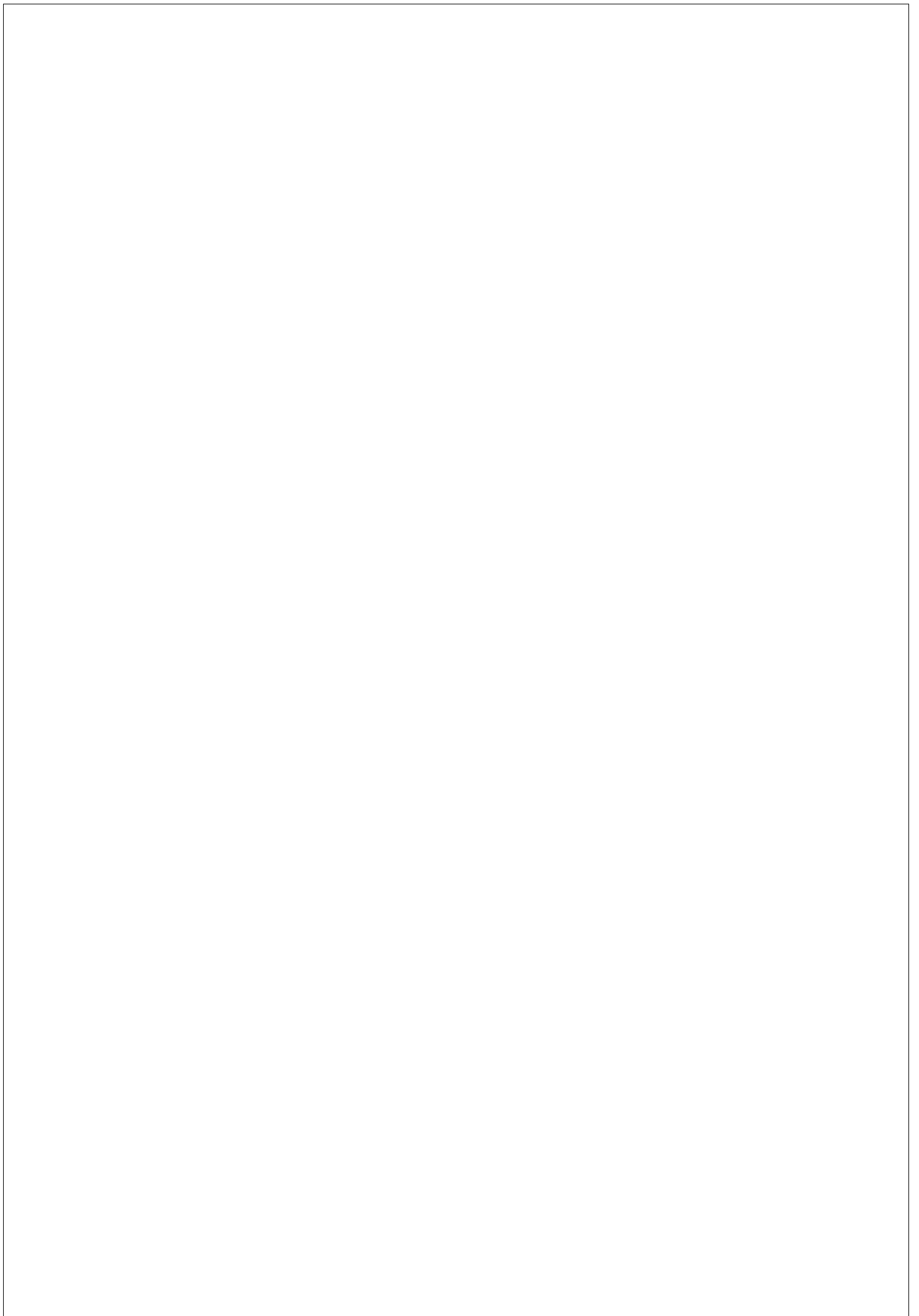
Rapporteur

Boudaa Abdelghani

Université de M'sila

Examineur

Année universitaire : 2018 /2019



DEDICACE

Nous dédions ce modeste travail et ma profonde gratitude :

À mes très chers parents qui nous ont fourni au quotidien un soutien et une confiance sans faille et de ce fait, je ne saurais exprimer ma gratitude seulement par des mots.

Que dieu vous protège et vous garde pour moi.

À tous les membres de ma famille

À tous mes amis avec lesquels j'ai partagé nos moments de joie et de bonheur.

À tous mes enseignants pour votre soutien, votre enseignement et vos conseils tout au long de mon parcours éducatif.

REMERCIEMENTS

Je remercie Dieu tout puissant qui a été toujours avec moi et qui m'a aidé tout au long de mon parcours.

Ensuite, bien sûr, je tiens à remercier mon encadreur Monsieur Samir AKHROUF, enseignant au sein de l'Université de M'sila au département de l'informatique, pour son encadrement, ses recommandations précieuses et sa disponibilité, ainsi que Monsieur Mohamed KAMEL qui m'a aidée par ses recommandations pour réaliser ce travail, Sans oublier Mme SAOUDI Lalia, qui était notre enseignant et notre sœur au même temps ,elle ne nous a rien épargné lors tout au long de son parcours éducatif avec nous.

Enfin je tiens à remercier les membres du jury pour avoir assisté à cette soutenance.

Table des Matières

Dédicace	i
Remerciement	ii
Table des Matières	iii
Table des Figures	iv
Liste des tableaux	viii
Introduction Générale	1
1 Les Données Géographiques	3
1.1 Base de données géographique	3
1.2 Les type de données géographiques	4
1.2.1 Données raster	4
1.2.2 Données vectorielles	5
1.3 Les données du projet	8
1.3.1 OpenStreetMap	8
1.3.2 Mapbox SDK pour Android	9
1.3.2.1 Caractéristiques de Mapbox SDK pour Android	10
1.3.2.2 Les limites des cartes offlines de Mapbox SDKs Android	12
2 L'univers Android	13
2.1 L'informatique mobile	14
2.1.1 Présentation	14
2.1.2 Les terminaux mobiles	14
2.2 Les applications mobiles	16
2.2.1 Définition	16
2.2.2 Les particularités d'une application mobile	17
2.2.3 Domaine des applications mobiles	17
2.2.4 Les différents types d'application mobile	17
2.3 Le système d'exploitation Android	18
2.3.1 Présentation	18

Table des Métiers

2.3.2 Historique	19
2.3.3 Architecture	20
2.4 Les applications Android	22
2.4.1 Présentation	22
2.4.2 État d'une activité (application)	22
2.4.3 Cycle de vie d'une activité Android	23
3 Conception	26
3.1 Diagramme des classes	26
3.2 Identification des cas d'utilisation	27
3.3 Diagramme des cas d'utilisation globale.	28
3.4 Diagrammes de séquences des cas d'utilisation	29
3.4.1 Diagramme de séquence du cas d'utilisation "préparer et télécharger une région"	29
3.4.2 Diagramme de séquence du cas d'utilisation "voir la localisation".	29
3.4.3 Diagramme de séquence du cas d'utilisation "chercher places"	30
3.4.4 Diagramme de séquence du cas d'utilisation "ajouter point d'intérêt"	31
3.4.5 Diagramme de séquence du cas d'utilisation "Gérer les cartes téléchargées".	32
3.4.6 Diagramme de séquence du cas d'utilisation "naviguer vers la destination".	33
3.4.7 Diagramme de séquence du cas d'utilisation "Enregistrer places"	34
3.4.8 Diagramme de séquence du cas d'utilisation "Gérer lieux enregistrés"	35
3.4.9 Diagramme de séquence du cas d'utilisation "voir la boussole Qibla"	35
3.5 Patrons de conception (Design Pattern)	36
3.5.1 Patron client/serveur à 3-tiers	37
3.5.2 Patron Modèle-vue-contrôleur	37
4. Réalisation	39
4.1 Environnement de travail.	39
4.1.1 Environnement matériel.	39
4.1.2 Environnement logiciel.	40
4.1.2.1 Les langages.	40
4.1.2.2 Les outils.	40

Table des Métiers

4.2 Structure de l'application	43
4.3 Captures d'interfaces de l'application.	44
Conclusion Générale et Perspectives	58
Webographie	ix
Bibliographie	xi

Table des Figures

1.1 base de données géographique.....	4
1.2 type de donnée géographique.....	4
1.3 représentation données raster.....	5
1.4 Exemple des données raster.....	6
1.5 Forme géométrique de données vectorielles.....	7
1.6 Table des données vectorielle.....	7
1.7 Exemple des données vectorielles.....	8
2.1 Un assistant personnel.....	15
2.2 Les smartphones	15
2.3 Les tablettes.....	16
2.4 L'architecture d'un système Android.....	20
2.5 Cycle de vie d'une activité Android.....	25
3.1 Diagramme des classes.....	27
3.2 Diagramme de cas d'utilisation global.....	28
3.3 Diagramme de cas d'utilisation " préparer et télécharger une région ".....	29
3.4 Diagramme de cas d'utilisation " voir la localisation ".....	30
3.5 Diagramme de cas d'utilisation " chercher places ".....	31
3.6 Diagramme de cas d'utilisation " ajouter point d'intérêt ".....	32
3.7 Diagramme de cas d'utilisation " gérer les cartes téléchargées ".....	33
3.8 Diagramme de cas d'utilisation " naviguer vers destination ".....	34
3.9 Diagramme de cas d'utilisation " enregistrer places ".....	34
3.10 Diagramme de cas d'utilisation " gérer les lieux enregistrés ".....	35
3.11 Diagramme de cas d'utilisation " direction de Qibla ".....	36
3.12 l'architecture " 3-tiers ".....	37
3.13 Patron MVC (Modèle, Vue, Contrôleur).....	37
4.1 Logo java.....	40
4.2 Logo Android Studio.....	40
4.3 Logo JDK.....	41
4.4 Logo SDK.....	41

Table des Figures

4.5 le jeton d'accès.....	42
4.6 Mapbox studio.....	42
4.7 Plan de l'application.....	43
4.8 logo de l'application.....	44
4.9 Interface d'accueil.....	45
4.10 demande GPS.....	45
4.11 autorisation GPS.....	46
4.12 voir localisation.....	46
4.13 Interface de recherche.....	47
4.14 résultat de recherche.....	47
4.15 le menu de l'application.....	48
4.16 interface de préparer et télécharger.....	49
4.17 saisir nom et confirmer le téléchargement.....	49
4.18 démarrage le téléchargement.....	50
4.19 gérer la liste des régions.....	50
4.20 voir le chemin.....	51
4.21 démarrage la navigation.....	52
4.22 marquer le point.....	53
4.23 remplir les informations.....	53
4.24 enregistrer place.....	54
4.25 gérer places enregistrer.....	55
4.26 direction de Qibla	55
4.27 sélectionner la langue.....	56
4.28 changer la langue.....	56
4.29 changer le style.....	57

Liste des Tableaux

1.1 bibliothèques de mapbox-sdk.....	10
1.2 Caractéristiques de Mapbox-sdk pour Android.....	12
2.1 Historique des versions du système Android.....	19
3.1 Cas d'utilisation associés au système.....	27

Introduction générale

Depuis environ trente ans, l'invention de l'internet a révolutionné la façon de penser et de vivre dans le monde, elle a permis aux consommateurs de faire des transactions, et accomplir leurs tâches sans avoir à se déplacer physiquement. Une dizaine d'années après, cette innovation est suivie par l'apparition de la technologie mobile qui a pris une place importante dans notre société, les assistants personnels (PDA), téléphones cellulaires, smartphones, tablettes, etc. [1]
[2]

Le tourisme en terme général ou la randonnée dans les montagnes et les forêts ou le désert avec les amis ou en famille est considérée bien sûr une chose très amusante surtout s'elle attache à passer leur temps et explorer de nouveaux endroits, camper et passer la nuit dehors.

Malgré l'évolution de la technologie Internet et des téléphones intelligents, nous entendons et parlons toujours de la mort et de la perte des randonneurs pendant leurs randonnées, particulièrement où le réseau téléphonique ou Internet n'est pas disponibles pour faire une demande de l'aide, et le plus grand danger est qu'ils ne connaissent pas les directions et où ils se trouvent à ce moment.

Pour remédier à ça, nous avons pensé à développer une application mobile pour aider les randonneurs à éviter les risques de la perte quand le manque de réseau d'internet en principe et à fournir d'autres services pour faciliter la randonnée. Pour ce faire, notre choix s'est porté sur la plateforme mobile Android qui est open source, gratuite et qui englobe une communauté importante par rapport à d'autres plateformes. **Mais comment on peut réaliser une telle application et qu'elle est la procédure à suivre pour la faire ?**

Pour atteindre cet objectif, nous commençons par comprendre le fonctionnement du système géographique GIS et le système mobile en général. Dans un deuxième lieu, nous choisissons une architecture Client/serveur afin d'implémenter les différents modules logiciels de l'application. Ensuite, nous procédons à la conception. Et enfin, nous développons l'application.

Introduction générale

Dans le premier chapitre, nous allons présenter les données géographiques en faisant un aperçu rapide sur les types de données utilisés dans les cartes géographiques et les données du projet qui nous allons utiliser pour réaliser notre système.

Ensuite, dans le second, nous allons introduire l'univers Android, en décrivant ce qu'est l'informatique mobile, le système Android et les applications Android.

Le troisième chapitre, focalise sur la conception et nous allons identifier les acteurs de notre système et les besoins du client qui deviendront ensuite les fonctionnalités de notre application.

Enfin, le quatrième chapitre, dédié à la réalisation de l'application où nous allons présenter l'environnement de développement, les outils et langages de programmation utilisés dans notre travail, puis quelques interfaces graphiques de notre application réalisée.

Et nous concluons notre travail par une conclusion générale et quelques perspectives.

Chapitre 1. Les Données Géographiques

Introduction

La définition des données géographiques issue des normes ISO (Organisation internationale de normalisation) est (ISO41) : "Données avec référence implicite ou explicite à un emplacement relatif à la Terre". Les données géographiques sont beaucoup plus que des cartes numériques. Ce sont des données de référence spatiale décrivant des objets dans l'espace, notamment des données géométriques et thématiques. Les données géographiques peuvent être utilisées dans un système d'information géographique (SIG) pour diverses analyses scientifiques et la production de cartes.

Dans ce chapitre nous allons définir c'est quoi données géographiques, et quelques notions de base, puis nous allons présenter données du projet qui concerne OpenStreetMap et Mapbox SDK ces deux derniers on va utiliser dans notre application.

1.1 Base de données géographique

La Base de données géographique est l'ensemble constitué par les couches superposables et les données attributaires [3] (Figure 1.1).

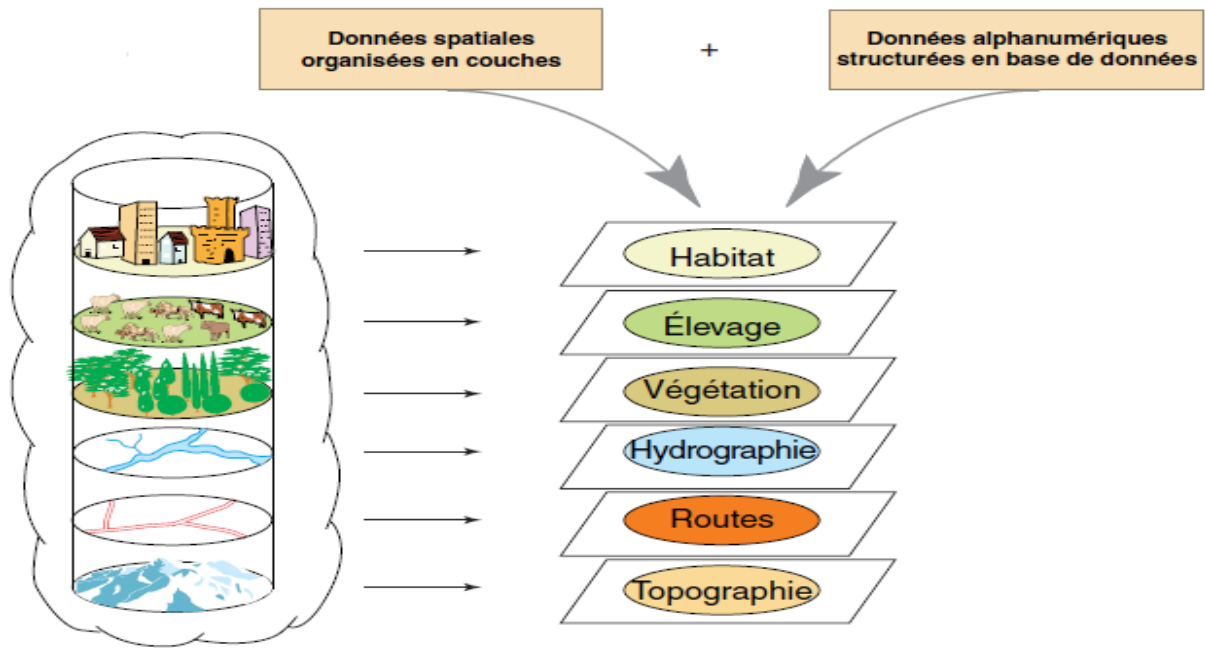


Figure 1.1 : base de données géographique

1.2 Les types de données géographiques

Il existe deux structures de données pour la représentation des données géographiques : la structure vectorielle et la structure raster. Ces deux structures de données ont été développées pour différentes applications (Figure 1.2). [3]

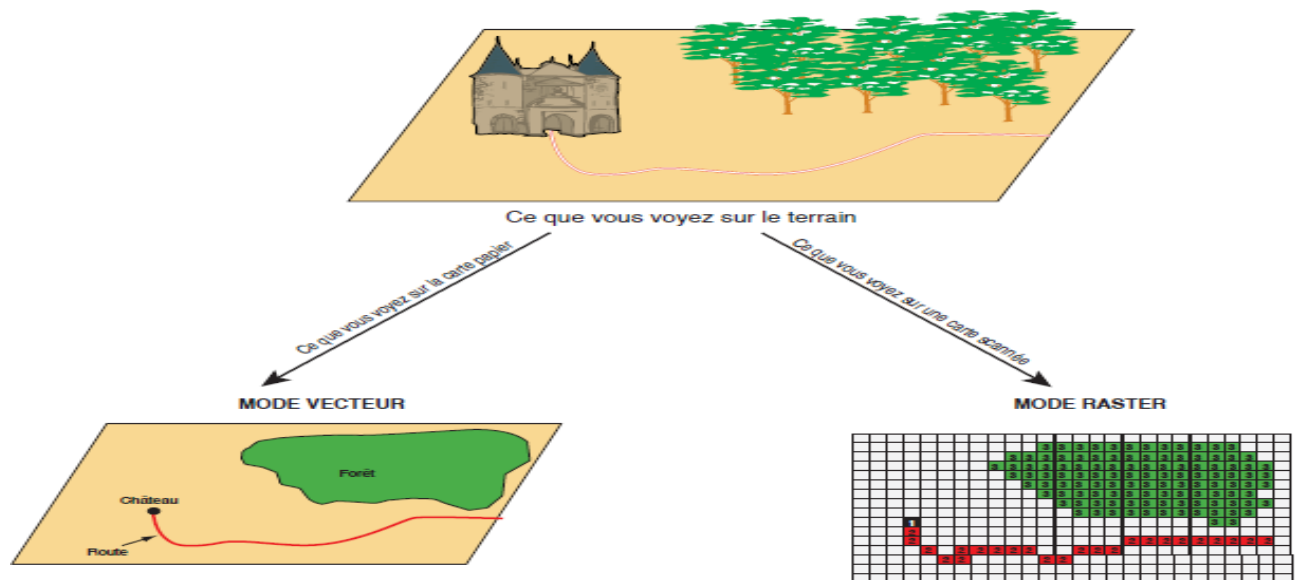


Figure 1.2 : type de donnée géographique.

1.2.1 Données raster :

La réalité est décomposée en une grille régulière et rectangulaire, organisée en lignes et en colonnes, chaque maille de cette grille ayant une intensité de gris ou une couleur. Comme la structure raster ne stocke qu'un seul numéro pour chaque cellule, il est possible de l'étendre pour qu'elle représente les couleurs RVB (rouge-vert-bleu) à l'aide de bandes raster. La juxtaposition des points recrée l'apparence visuelle du plan et de chaque information. Une forêt sera "représentée" par un ensemble de points d'intensité identique [3] (Figure 1.3) (Figure 1.4).

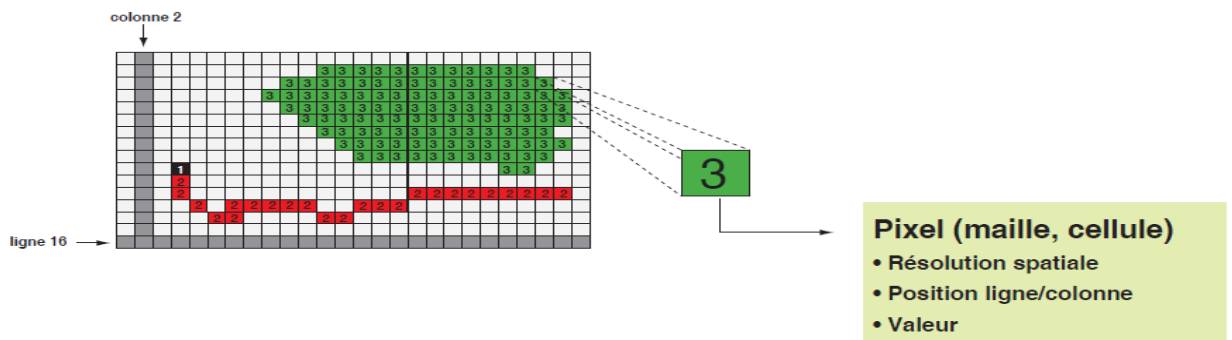


Figure 1.3 : représentation données raster

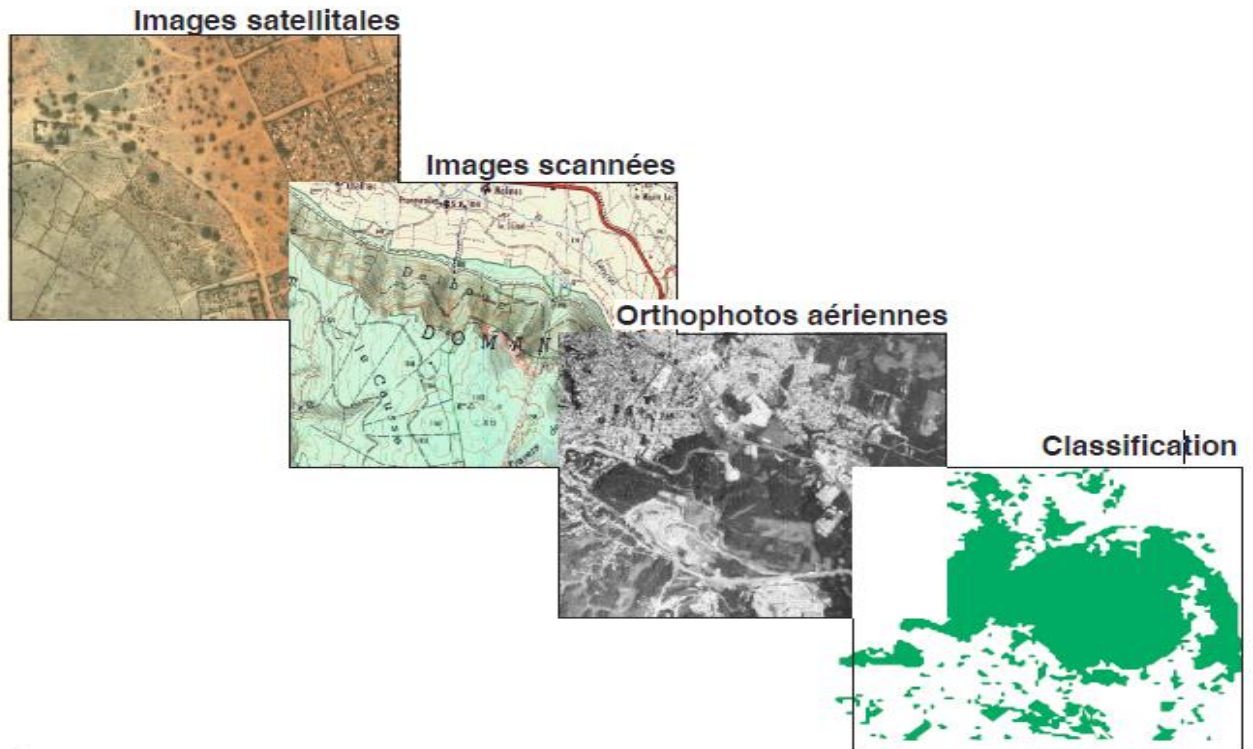


Figure 1.4 : Exemple données raster.

1.2.2 Données vectorielles :

Dans la structure vectorielle, les entités géographiques sont représentées sous forme de formes géométriques. Les formes géométriques de base utilisées dans la structure vectorielle sont le point, la ligne (polyligne) et le polygone (Figure 1.5). Le point est un objet géométrique à 0 dimension qui est fixé par les coordonnées x (ou latitude), y (ou longitude) et z (ou altitude) et représente un certain emplacement dans l'espace géographique. Par exemple, les arrêts de bus peuvent être représentés sous forme de points dans l'échelle de la ville. Chaque objet spatial est doté d'un identifiant qui permet de le relier à une table attributaire [3] (Figure 1.6).

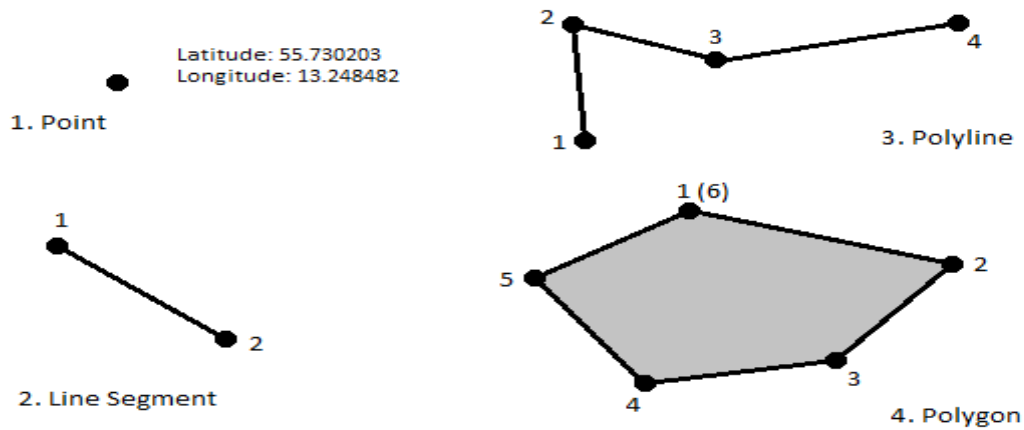


Figure 1.5 : forme géométrique de données vectorielles

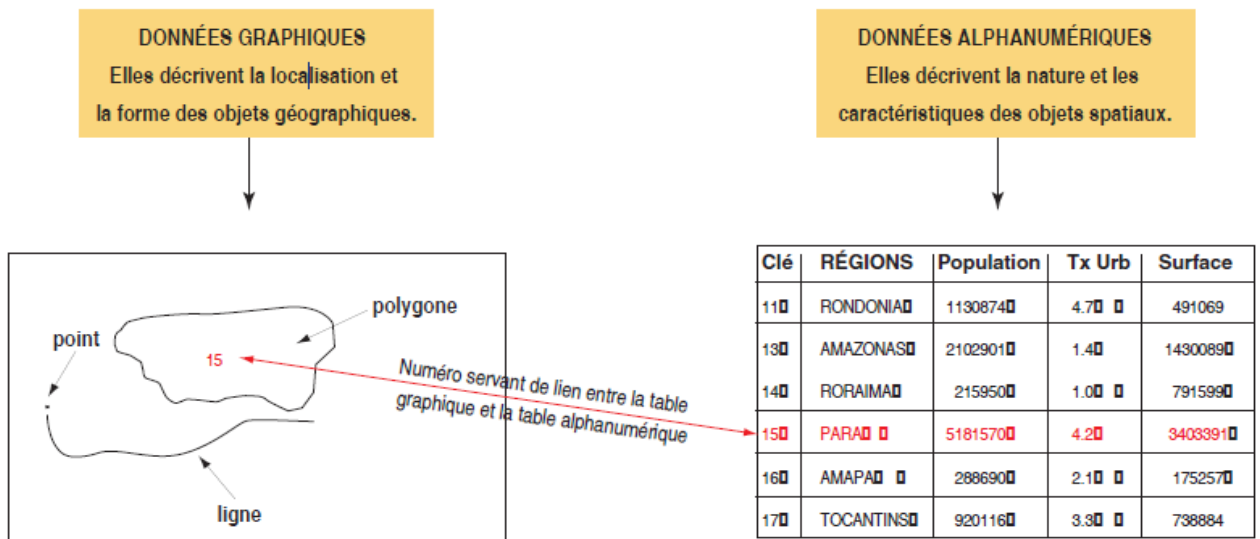


Figure 1.6 : table des données vectorielle.

Les points définissent des localisations d'éléments séparés pour des phénomènes géographiques trop petits pour être représentés par des lignes ou des surfaces qui n'ont pas de surface réelle comme les points cotés.

Les lignes représentent les formes des objets géographiques trop étroits pour être décrits par des surfaces (ex : rue ou rivières) ou des objets linéaires qui ont une longueur mais pas de surface comme les courbes de niveau.

Le polygone est un objet géométrique à 2 dimensions et est représenté sous la forme d'une zone circonscrite par la polyligne, dont les points de départ et d'arrivée se correspondent. Ils

représentent la forme et la localisation d'objets homogènes comme des pays, des parcelles, des types de sols. etc. [3] (Figure 1.7).

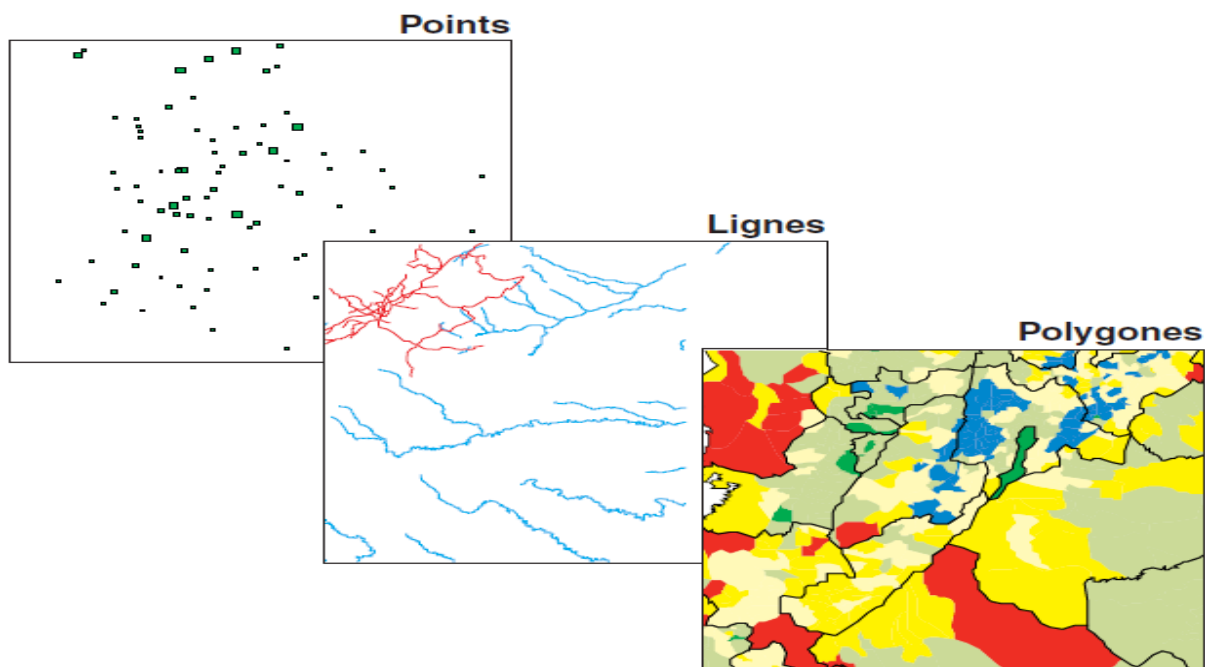


Figure 1.7 : exemple des données vectorielles.

1.3 Les données du projet

Il existe un grand nombre de datasets géographiques. Dans ce travail, nous utilisons deux datasets : OpenStreetMap (OSM) et MapBox

1.3.1 OpenStreetMap

OSM est un service de carte open source qui crée et fournit des données géographiques gratuites. OSM est un exemple d'information géographique volontaire VGI. Les gens collectent des données géographiques à partir de plusieurs sources, telles que des appareils GPS et des images satellites à code source libre, puis téléchargent ces informations dans la base de données de l'OSM ; De cette manière, l'utilisateur peut mettre à jour, ajouter ou corriger les données de la carte dans cette zone.[4]

«L'open source est un système de partage gratuit qui peut être mis à la disposition de tous ceux qui souhaitent y travailler et dont tout le monde peut tirer parti» [5]. L'open source ne concerne pas seulement l'information géographique ou la communauté informatique ; il peut être élargi à divers domaines, par exemple en partageant les recettes de cuisine, etc.

VGI est également basé sur la technologie wiki et constitue un contenu d'information géographique généré par l'utilisateur. Pour ce faire, chaque volontaire est un capteur qui collecte des données géographiques, les télécharge sur le serveur, modifie les données téléchargées ou modifiées par d'autres utilisateurs et télécharge les données. données à usage personnel, sans considérer que l'utilisateur est un expert en GIS ou un amateur. En ce qui concerne Wikipedia, des problèmes se posent concernant le VGI, tels que l'incertitude ou l'exactitude des données, la vie privée, etc. [6]

En ce qui concerne l'exactitude, il y a deux problèmes principaux. Premièrement, les données collectées par les utilisateurs sont directement transmises à la base de données sans contrôle strict de la qualité. Deuxièmement, le contrôle de l'exactitude des données, impossible en raison de la diversité des données, sources et expérience des volontaires en information géographique.

Étant donné que la plupart des cartes ont une utilisation restreinte, OSM offre aux utilisateurs la possibilité d'utiliser les données cartographiques nécessaires de manière plus productive et plus créative. Dans le même but, OSM n'a aucune restriction quant au type de données, qui peuvent être téléchargées dans une base de données pour autant qu'elles soient exactes et ne violent le droit d'auteur de personne. La tâche principale actuelle de l'OSM consiste à étendre la couverture des données cartographiques sans utiliser les cartes existantes. Les données OSM sont soumises à l'accord de la licence open source «Licence Creative Commons Attribution Share-Alike 2.0 (CC-by-SA)». [7]

1.3.2 Mapbox SDK pour Android

Plusieurs bibliothèques permettent aux développeurs Android d'intégrer OpenStreetMap dans leurs propres applications, qu'il s'agisse d'une image de carte statique, d'une carte entièrement interactive ou d'utilisations plus sophistiquées telles que le géocodage et le routage. L'utilisation de ces bibliothèques présente un autre avantage : elles respectent la politique d'utilisation des tuiles d'OpenStreetMap en récupérant des tuiles et d'autres ressources sur des serveurs dédiés au lieu des serveurs donnés par OpenStreetMap. Mapbox est l'une des bibliothèques utilisant les données. [8]

Mapbox SDKs sont suite des bibliothèques à code source libre développée par Mapbox pour incorporer des cartes glissantes personnalisables dans des applications Web, mobiles (Android /IOS) et de bureau. Il existe également une bibliothèque permettant de créer des outils de rendu côté serveur ou en ligne de commande.(figure 1.8) .

Library	Platform	Target languages	Latest release
Mapbox GL JS	Web	JavaScript	Download
Mapbox Maps SDK for Android	Android	Java, Kotlin	Look for "android-"
Mapbox Maps SDK for iOS	iOS	Objective-C, Swift, Interface Builder	Look for "ios-"
Mapbox Maps SDK for macOS	macOS	Objective-C, Swift, Interface Builder, AppleScript	Look for "macos-"
Mapbox Maps SDK for Qt	Qt (including Windows)	C++03, QML	Look for "qt-"
node-mapbox-gl-native	Node.js	JavaScript	Look for "node-"

Table 1.1 : bibliothèques de mapbox-sdk

Le SDK Mapbox Maps est basé sur le rendu Mapbox GL, qui prend des feuilles de style conformes à la spécification de style Mapbox, les applique aux tuiles vectoriels conformes à la spécification de tuile vectorielle Mapbox via les manifestes TileJSON et les rend dynamiquement à l'aide d'OpenGL pour des performances élevées. Par défaut, Mapbox GL affiche les tuiles vectorielles Mapbox Streets basées sur les données OpenStreetMap. Le développeur peut choisir parmi plusieurs styles conçus par Mapbox ou concevoir un style personnalisé dans l'éditeur de style graphique de Mapbox Studio.

Sur les plateformes mobiles, le composant Mapbox Telemetry utilise les données de localisation du périphérique pour faciliter les améliorations des données OpenStreetMap.

Le SDK Mapbox Maps pour Android constitue la base du SDK de navigation Mapbox pour Android [9]

1.3.2.1 Caractéristiques de Mapbox-SDK pour Android

L'ensemble de fonctionnalités du SDK Mapbox Maps est varié selon la plate-forme. En général, les fonctionnalités de rendu sont d'abord ajoutées à Mapbox GL JS, puis portées sur les kits SDKs natifs, tandis que les fonctionnalités centrées sur les smartphones ont tendance à être spécifiques aux SDKs natifs [9]

Ce tableau a été mis à jour pour la dernière fois le 9 novembre 2018 (UTC) à 01h40.

Chapitre 1. Les données géographiques

Caractéristiques	Android
OSM base map	oui
Terrain base map	oui
Satellite base map	oui
Custom styles	oui
High-resolution rendering (Retina)	oui
Continuous zooming	oui
Rotation	oui
Perspective (tilt, pitch)	oui
Axonometric (orthographic, skewed)	oui
Localized labels	Optionnel
3D building extrusion	oui
Styleable DEM terrain	oui
Heat maps	oui
Viewport animation	oui
Multitouch gestures	oui
Keyboard shortcuts	oui
Markers (point annotations)	oui
Marker views (view-backed annotations)	Optionnel
Marker keyframe animation	Optionnel
Shape overlays (shape annotations)	oui
Arbitrary GeoJSON	oui
GeoJSON point clustering	oui
Popups (callouts, popovers, info windows)	oui
Tooltips	N/A

Chapitre 1. Les données géographiques

Georeferenced images	oui
Embedded video	Non
User location (user dot)	oui
Customizable user dot	oui
User location tracking (GPS tracking)	oui
Telemetry	optionnel
Offline maps	oui
WMS	oui
TMS	oui
Attribution control	oui
Scale control	Non
Localized UI	oui
Bidirectional (right-to-left) text	oui
Complex text shaping	partial
Vertical CJK text	oui
Data-driven styling (property expressions)	oui
Style introspection (runtime styling)	oui
Data introspection (feature eventing, feature	oui
Screenreader support (accessibility)	Non
Text scaling (Dynamic Type)	Non
Static snapshots	oui
Printing	oui
IDE integration (code completion,	oui

Table 1.2 : Caractéristiques de Mapbox-sdk pour Android

1.3.2.2 Les limites des cartes offlines de Mapbox SDKs Android

La taille totale des ressources hors connexion varie d'une région à l'autre en fonction de :

- L'étendue géographique de la région.
- La plage de niveau de zoom de la région
- Comptabilité et densité.
- Les sources, les polices et les icônes utilisées par le style
- La taille des carreaux individuels requise par la région.
- Si la région chevauche les régions téléchargées existantes

Pour le style Mapbox Streets par défaut, certaines tailles totales typiques et approximatives :

- Ville de Barcelone : 83 MB.
- Le Grand Londres au sein de la M25 aux niveaux de zoom 0-15 : 120 MB.
- États-Unis contigus aux niveaux de zoom 0–9 : 290 MB.

Et pour Mapbox Satellite :

- Ville de Barcelone : 45 MB.
- Grand Londres au sein de la M25 aux niveaux de zoom 0-15 : 400 MB.
- États-Unis contigus aux niveaux de zoom 0–9 : 315 MB. [10]

Après avoir consommé toutes les ressources dans le plan gratuit y a un plan payement qui offre par Mapbox

La méthode de payement est facile et automatique. Si le nombre de demandes d'API incluses dans le forfait payant ou le forfait commercial a été dépassé, des demandes supplémentaires sont automatiquement traitées à 0,50 \$ par 1 000 demandes ou par 500 utilisateurs actifs par mois. [11]

Conclusion

Dans ce chapitre nous avons présenté brièvement les données géographiques et les données de l'OpenStreetMap et MapBox SDKs pour Android et leurs caractéristiques leurs limitations avec les cartes géographiques offlines. Cela nous a aidé à bien comprendre le fonctionnement des données géographiques en terme général.

Le chapitre suivant sera consacré à la phase de l'univers Android.

Chapitre 2. L'univers Android

Introduction

Rares sont les systèmes ayant connu une progression telle que celle que connaît actuellement le système Android.

Dans ce chapitre nous allons définir c'est quoi l'informatique mobile, et quelques notions de base, puis nous allons présenter l'univers Android tout en définissant c'est quoi le système Android, son architecture et le cycle de vie d'une application Android.

2.1 L'informatique mobile

2.1.1 Présentation

Difficile, aujourd'hui, de passer à côté de l'explosion de l'internet mobile. En l'espace de quelques mois, les nouveaux terminaux mobiles que sont les tablettes et les smartphones ont envahi notre quotidien. Cet afflux technologique change radicalement notre rapport à l'informatique en affranchissant de l'utilisation d'un micro-ordinateur classique. [12]

2.1.2 Les terminaux mobiles

Un terminal mobile est un petit appareil informatique ou de communication qu'on peut transporter avec soi dans ses déplacements et utiliser comme terminal donnant accès sans fil à un ou plusieurs réseaux. [13] Parmi les terminaux mobiles, on trouve des assistants numériques personnels (PDA), téléphones intelligents (Smartphone), des tablettes, etc.

- **Les assistants personnels (PDA)**



Figure 2.1 : Un assistant personnel

Un assistant personnel est un périphérique portable qui fonctionne comme un gestionnaire d'informations personnelles. Les PDA sont utilisés pour la navigation sur le Web, les applications bureautique, les vidéos, les photos ou les téléphones mobiles.

Les caractéristiques du modèle PDA varient, mais les fonctionnalités communes courantes Incluent les écrans tactiles, la connectivité Bluetooth et Wifi, etc. Les PDA contiennent des logiciels pour synchroniser les informations. [14]

▪ Les smartphones



Figure 2.2 : Les smartphones

Un smartphone (téléphone intelligent) est un téléphone portable doté de fonctionnalités très avancées. Un smartphone typique dispose d'un écran tactile haute résolution, de la connectivité

WIFI, des capacités de navigation Web et de la capacité d'accepter des applications sophistiquées. La plupart de ces appareils fonctionnent sur l'un de ces systèmes d'exploitation mobiles populaire : Android, Symbian, IOS, BlackBerry OS et Windows. [15]

- **Les tablettes**



Figure 2.3 : Les tablettes

Une tablette PC est un ordinateur portable hybride entre un assistant numérique personnel(PDA) et un ordinateur portable. Équipé d'une interface à écran tactile, possède généralement une application logicielle utilisée pour exécuter un clavier virtuel. Cependant, de nombreuses tablettes PC prennent en charge les claviers externes. Elles ont des fonctions de navigation Web intégrées, des options de connectivité multiples, des écrans tactiles capacitifs et multimédia, y compris un support haut définition (HD). [16]

2.2 Les applications mobiles

2.2.1 Définition

Une application mobile est un type de logiciel ou programme conçu pour s'exécuter sur un appareil mobile, tel qu'un smartphone ou une tablette. Les applications mobiles servent souvent à fournir aux utilisateurs des services similaires à ceux du PC. [17]

2.2.2 Les particularités d'une application mobile

Pour répondre aux besoins et aux attentes du mobinaute, une application doit être pensée pour la mobilité. Même si le service existe déjà, par exemple en application web, l'application mobile ne doit pas être présentée comme une application Desktop (ou être copie).

Tout d'abord car la taille du terminal est différent et aussi car le mobile offre d'autres possibilité que le web. Parmi les particularités d'une application mobile on trouve la performance et l'environnement. [18]

- **La performance**

Une application mobile est développée pour atteindre un niveau de qualité, en termes d'ergonomie mais aussi de performance, inégale par rapport aux autres moyens d'accéder à un service sur le mobile. Une application mobile peut être indépendante ou liée à un service web et utiliser ou non le web. [18]

- **Environnement ou absence d'internet**

Une application mobile n'a donc pas obligatoirement besoin d'internet pour fonctionner sur le téléphone. Si l'utilisateur se retrouve dans une zone privée d'internet, alors il pourra continuer à utiliser son application (même si les informations affichées ne sont pas les dernières en ligne dans le cas où l'application récupère des informations). [18]

2.2.3 Domaine des applications mobiles

Le domaine d'utilisation des applications mobiles est très vaste, on peut citer par exemple la géolocalisation, itinéraire, Paiement mobile, traitement du texte, les jeux, réseaux sociaux et éducation, etc.

2.2.4 Les différents types d'application mobile

Il existe plusieurs types d'applications mobiles qui sont les applications natives, web et hybrides.

- **Les applications natives**

Une application native est une application développée spécifiquement pour une seule plateforme, grâce aux outils conçus pour celle-ci. Elle est développée avec un

langage spécifique à son système d'exploitation et est distribuée uniquement par l'intermédiaire de son store (AppStore IOS, PlayStore pour Android, ...). Pour développer une application pour deux plateformes différentes, il vaut mieux de développer deux applications distincts. [19]

- **Les applications Web (WebApps)**

En opposition à une application native, une WebApps est une application mobile développée avec les outils de développement web actuel : HTML5, CSS3 et JavaScript. C'est une application qui une fois développée est accessible et exécutable sur tous les smartphones via leur navigateur web. L'avantage de ces applications, c'est le gain de temps et d'argent réalisé grâce à leur développement unique et leur déploiement multiplateformes. Dans un cas, vous développez une seule application alors que dans l'autre, vous développez trois applications (pour Android, IOS et Windows Phone). [19]

- **Les applications hybrides**

Les applications hybrides sont des applications qui combinent les éléments d'une WebApps et les éléments d'une application native. Elles reposent essentiellement sur la solution Cordova/PhoneGap, cette solution sert de passerelle entre le langage web et le natif. Cette solution nous permet d'utiliser un seul et même outil pour le développement et les langages issus du développement Web pour tous les mobiles (IOS, Android et Windows Phone). [19]

2.3 Le système d'exploitation Android

2.3.1 Présentation

Android est une plate-forme intégrée pour la première fois dans un smartphone (téléphone intelligent) sorti en France en mars 2009, Android s'est depuis émancipé très rapidement pour conquérir de nombreux appareils, mobile ou non, tels que les netbooks (mini-ordinateurs), les tablettes tactiles et les télévisions connectées, au point de devenir aujourd'hui l'un des systèmes d'exploitation majeurs dans le monde. [20]

2.3.2 Historique

L'histoire d'Android commence en octobre 2003, lorsqu'Andy Rubin, Rich Miner, Nick Sears et Chris White créent la société Android à Palto Alto (Californie).

Google a racheté la société en août 2005. Deux ans plus tard, l'Open Handset Alliance est annoncée et Android devient officiellement open source.

La première version du SDK Android 1.0 sort en 2008 avec le premier téléphone sous Android (HTC Dream). En avril 2009, la version 1.5 (API 3) d'Android sort. Cette version baptisée Cupcake (petit gâteau) inaugure les nouveaux noms des versions d'Android ce qui donnera pour les futures versions comme le montre la figure dessous. [21]

Nom	N version	Date	Niveau API
N/A	1.0, 1.1	Septembre 2008, Février 2009	1, 2
Cupcake	1.5	Avril 2009	3
Donut	1.6	Septembre 2009	4
Eclair	2.0-2.1	Octobre 2009	5-7
Froyo	2.2-2.2.3	Mai 2010	8
Gingerbread	2.3-2.3.7	Décembre 2010	9-10
Honeycomb[a]	3.0-3.2.6	Février 2011	11-13
Ice Cream Sandwich	4.0-4.0.4	Octobre 2011	14-15
Jelly Bean	4.1-4.3.1	Juin 2012	16-18
Kitkat	4.4.x	Octobre 2013	19-20
Lollipop	5.0-5.1.1	Octobre 2014	21-22
Marshmallow	6.0- 6.0.1	Mai 2015	23
Nougat	7.0 - 7.1.1	Septembre 2016	24-25
Oreo	8.0 - 8.1	août-Décembre 2017	26-27
Piel	9.0	août 2018	28

Table 2.1 : Historique des versions du système Android.

2.3.3 Architecture

Cette figure illustre les différentes couches au niveau du système Android.

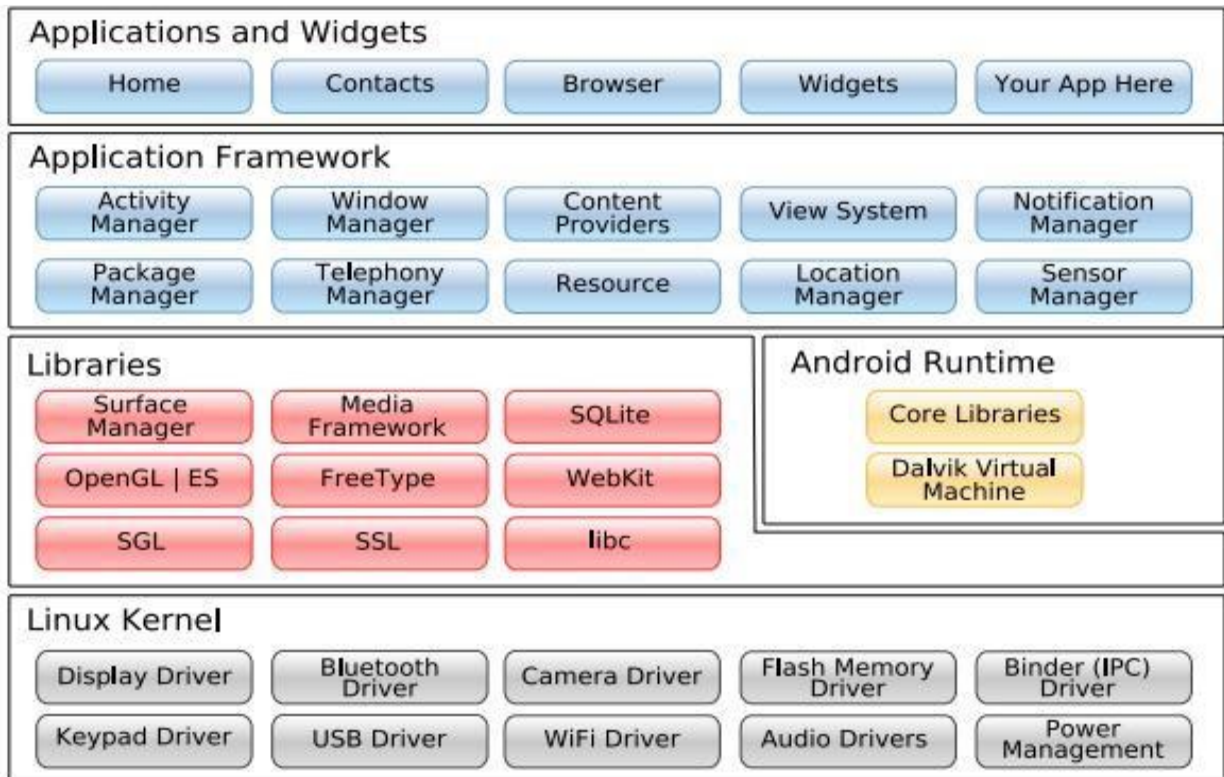


Figure 2.4 : L'architecture d'un système Android

Comme la montre la figure ci-dessus, Le système d'exploitation Android est basé sur une architecture à quatre niveaux.

- **Linux Kernel**

Le système d'exploitation Android s'appuie sur un noyau Linux, régulièrement mis à jour selon les versions du système : si les premières versions utilisaient la version 2.6.x du noyau Linux les actuelles dernières versions (Android 5.x) est basées sur la version 3.4.x de Linux.

Cette première couche prend en charge la gestion des couches basses (gestion des processus, de la mémoire, de la couche matérielle) ainsi que les droits utilisateurs. [22]

Au-dessus de noyau se trouve une couche d'espace utilisateur native, constituée par le binaire init (le premier processus démarré, qui fait tourner tous les autres processus), plusieurs démon et quelque centaines bibliothèque natives sont utilisées dans tout le système. [23]

- **Les librairies**

A ce niveau, figure la couche des bibliothèques principales du système fournies par des tiers. Celles-ci, de bas niveau, sont écrites en C et/ou C++.

Elles fournissent des services essentiels tels que la bibliothèque d'affichage en 2D (SGL), la bibliothèque d'affichage graphique 3D (OpenGL), la bibliothèque de base de données (SQLite), la lecture et l'enregistrement audio et vidéo (Media Framework), un moteur de navigation Web (WebKit) ... [20]

- **Android Runtime (Le moteur d'exécution linux)**

Au même niveau que cet ensemble de librairies se trouve l'environnement fonctionnel d'Android basé sur une technologie Java avec une spécificité unique : sa machine virtuelle optimisée pour les applications mobiles. Cette machine virtuelle, dénommée Dalvik, est initialisée par chaque application lors de son lancement et ouvre, pour chacune, une Dalvik spécifique, ce qui rend l'OS complètement multitâche.

Dalvik est fournie avec son kit de développement spécifique ou Application Programming Interface (API) écrit en Java ; les applications Android sont donc écrites en Java, mais un java spécifique à Android. Le Java Développeur Kit (JDK) Android est, en fait, un sous-ensemble du JDK de SUN et intègre une partie des classes JDK SUN et JDK SWING, plus des classes spécifiques à Android. [24]

- **Framework Android**

Les fonctionnalités offertes par ces bibliothèques sont ensuite reprises et utilisées par cette couche sous forme de bibliothèques Java. Celles-ci fournissent des bibliothèques et composants réutilisables spécifiques à des domaines particuliers. On y retrouve par exemple les bibliothèques de base de données, de téléphone, localisation géographique, de communication en champ proche. [20]

- **Les applications**

Enfin, une dernière couche parachève cette structure : les applications. Ces applications peuvent être, bien sûr, les applications tierces téléchargées sur le magasin d'application officiel, mais également des applications installées par défaut, telles que l'application d'accueil (aussi appelée Launcher), le navigateur web, les

applications de SMS et téléphonie, etc. Toutes ces applications sont communément développées en Java. [22]

2.4 Les applications Android

2.4.1 Présentation

Une application Android est une application mobile spécifiquement développée pour les smartphones et les tablettes utilisant le système Android. Elles sont de nature très variables tel que les applications jeux, mobile commerce, utilitaire, service d'information. [25]

Sous Android, une application est composée d'une ou plusieurs activités.

Une activité est la base d'un composant pour la création d'interfaces utilisateur. Afin de faciliter la cinématique de l'application, il est préconisé de n'avoir qu'une interface visuelle par activité. [26]

2.4.2 État d'une activité (application)

Une application Android peut se trouver dans des états différents qui sont : en cours d'exécution, en pause, stopper et tuer.

- **En cours d'exécution :**

l'activité se trouve au premier plan et reçoit les interactions utilisateurs. Si l'appareil a besoin de ressources, l'activité se trouvant en bas de la back stack.

- **Pause :** l'activité est visible mais l'utilisateur ne peut pas interagir avec (cachée par une boîte dialogue par exemple). La seule différence avec l'état précédent est la non-réception des événements utilisateurs.

- **Stopper :** l'activité n'est plus visible mais toujours en cours d'exécution. Toutes les informations relatives à son exécution sont conservées en mémoire. Quand une activité passe en état stopper, vous devez sauvegarder les données importantes et arrêter tous les traitements en cours d'exécution.

- **Tuer** : l'activité est tuée, elle n'est plus en cours d'exécution et disparaît de la back stack. Toutes les données présentes en cache non sauvegardées sont perdues.

2.4.3 Cycle de vie d'une activité Android

Par défaut, chaque application Android s'exécute dans un processus séparé. Android gère les ressources disponibles dans un appareil et peut, si besoin, fermer des applications pour libérer des ressources (hors application en cours d'exécution).

Le choix de l'application à fermer dépend fortement de l'état du processus dans lequel elle se trouve. Si Android doit choisir entre deux applications se trouvant dans le même état, il choisira celle qui se trouve dans cet état depuis plus longtemps.

Le cycle de vie d'une activité est assez complexe et sa compréhension est indispensable dans le développement Android. Le schéma ci-dessous résume ce cycle de vie.

Lors du lancement d'une activité, la méthode **onCreate** est appelée. Dans cette méthode, vous devez initialiser votre vue et lier les données à une liste. Cette méthode prend en paramètre un Bundle (pile) contenant l'état précédent de l'activité.

Cet appel est suivi par la méthode **onStart** afin de signifier le lancement effectif de l'application (elle est maintenant visible). Cette méthode peut aussi être appelée par la méthode **onRestart**.

Puis la méthode **onResume** est appelée afin d'exécuter tous les traitements nécessaires au fonctionnement de l'activité (thread, processus, traitement), initialiser des variables et les listeners. Ces traitements devront être arrêtés lors de l'appel à la méthode **onPause** et relancés si besoin lors d'un futur appel à la méthode **onRésumé**.

Après ces trois appels, l'activité est utilisable et peut recevoir les interactions utilisateurs. Si une autre activité passe au premier plan, l'activité en cours d'exécution passera en pause. Juste avant l'appel à la méthode **onPause**, la méthode **onSaveInstanceState** est appelée afin de vous permettre de sauvegarder les informations importantes portées par l'activité.

Ces informations pourront être appliquées aux futurs lancements de l'activité lors de l'appel à la méthode **onRestoreInstanceState** ou de l'appel à **onCreate**.

La méthode **onPause** permet de stopper tous les traitements effectués (traitement non nécessaire si l'activité n'est pas visible) par l'activité (traitement, thread, processus). Si votre activité est de nouveau visible, cela correspondra à un appel à la méthode **onResume**. Le passage de l'activité à l'état "stopper" correspond à un appel à la méthode **onStop**. Dans cette méthode, il faut arrêter tous les traitements restants. Une fois stoppée, votre activité peut :

- **Soit être relancée** : cela s'effectue par un appel à la méthode **onRestart** suivi du cycle de vie normal de l'activité.
- **Soit être tuée** : cela s'effectue par un appel à la méthode **onDestroy**, dans laquelle vous devez arrêter tous les traitements restants, fermer toutes les connexions à la base de données, tous les threads, tous les fichiers ouverts, etc. Vous pouvez provoquer l'appel à la méthode **onDestroy** en utilisant la méthode **finish**. [27]

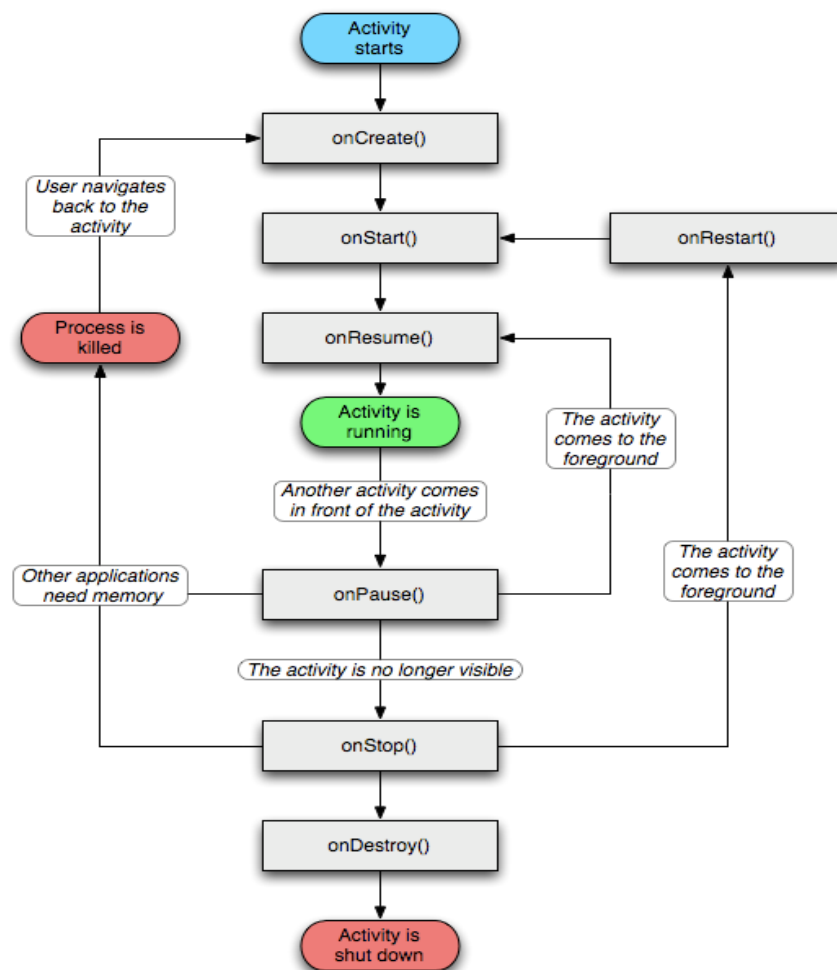


Figure 2.5 : Cycle de vie d'une activité Android

Conclusion

Dans ce chapitre nous avons présenté brièvement l'informatique mobile et le monde Android tout en décrivant le système Android, son architecture et l'évolution de ses versions à travers le temps depuis son apparition jusqu'à la version actuel. Cela nous a aidés à bien comprendre le fonctionnement de ce système.

Le chapitre suivant sera consacré à la phase de la conception.

Chapitre 3. Conception

Introduction

La démarche de conception est une étape fondamentale dans le processus de développement puisqu'elle fait correspondre la vision applicative (le modèle d'analyse) à la vision technique (l'environnement de développement et d'exécution).

Ce chapitre vise à illustrer la phase de conception et les modèles UML associés.

Nous allons établir le diagramme des cas d'utilisation global puis on va décrire ses cas d'utilisation, ensuite les diagrammes de séquence des cas d'utilisation et après nous présentons les patterns de la conception, et nous terminons avec une petite conclusion.

3.1 Diagramme des Classes

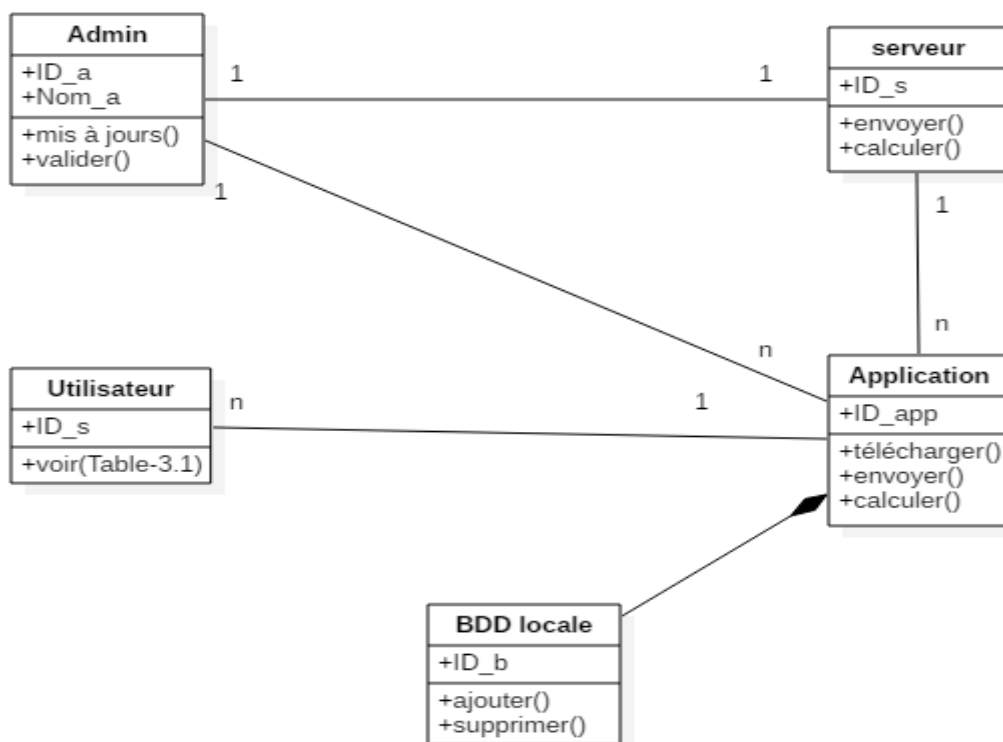


Figure 3.1 : Diagramme des classes

3.2 Identification des cas d'utilisation

Un cas d'utilisation (use case) représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier.

Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée "notable" à l'acteur concerné.

- Le tableau suivant liste les différents cas d'utilisation associés à notre application.

N°	Cas d'utilisation	L'acteur
1	Voir sa localisation	Randonneur
2	Voir le plus court chemin entre	Randonneur
3	Naviguer vers point de destination	Randonneur
4	Chercher des places	Randonneur
5	Préparer et télécharger une région	Randonneur
6	Gérer la liste des régions téléchargées	Randonneur
7	Ajouter des points d'intérêt	Randonneur
8	Enregistre des places visitées	Randonneur
9	Partage sa localisation	Randonneur
10	Partager les places visitées	Randonneur
11	Voir la direction de Qibla	Randonneur

Table 3.1 : Cas d'utilisation associés au système

3.3 Diagrammes des cas d'utilisation global

Le diagramme suivant résume tous les cas d'utilisation associés au 'randonneur'.



Figure 3.2 : Diagramme de cas d'utilisation global

3.4 Diagrammes de séquence des cas d'utilisation

3.4.1 Diagramme de séquence du cas d'utilisation "préparer et télécharger une région"

Pour que l'utilisateur ou le randonneur utilise la carte géographique pendant sa randonnée, il doit avoir une connexion internet, puis il faut appuyer le bouton mettre l'écran à la région qui sera téléchargée, puis appuyer sur le bouton de téléchargement et lorsque le téléchargement est terminé il peut naviguer sur cette région.

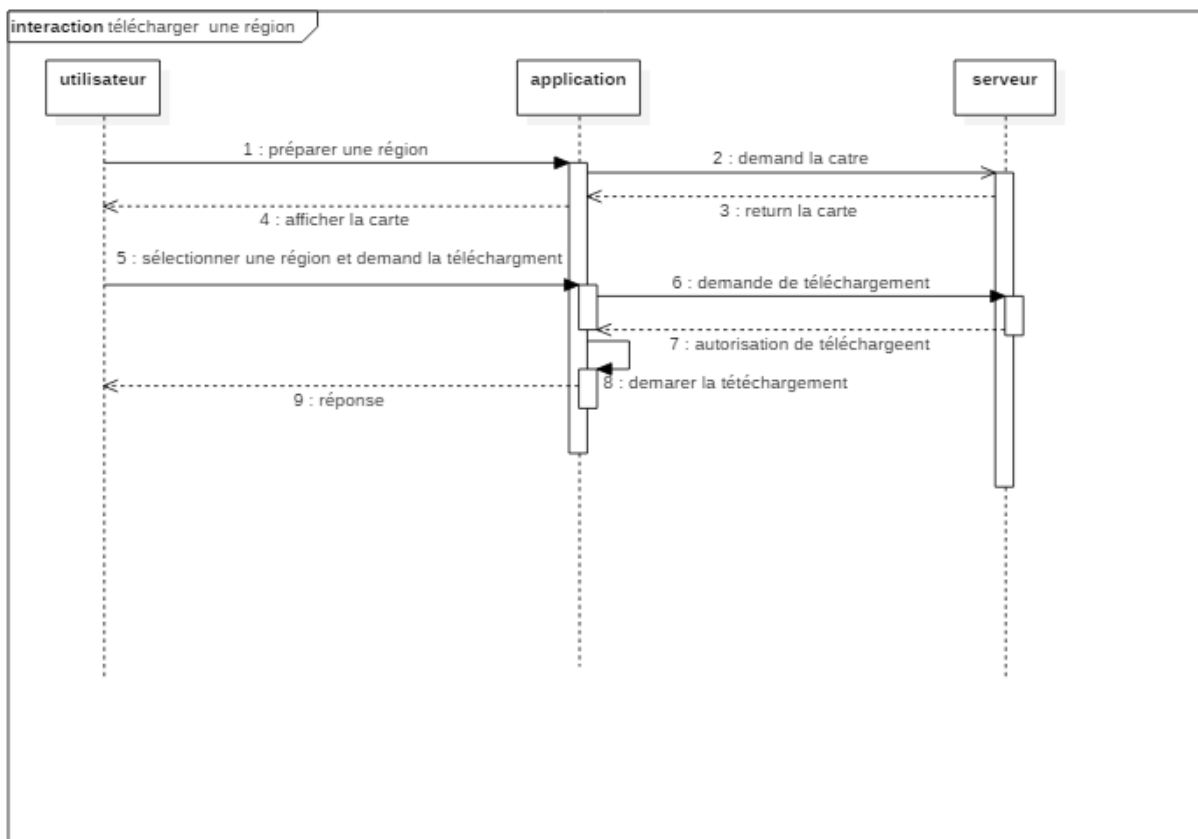


Figure 3.3 : Diagramme de cas d'utilisation "préparer et télécharger une région".

3.4.2 Diagramme de séquence du cas d'utilisation "voir la localisation"

Pour que l'utilisateur voir sa localisation dans la carte géographique avec ou sans accès à l'internet, il doit autoriser le service de localisation dans sa propre smartphone, puis si a accès internet il peut voir sa localisation sur la carte après appuyer le bouton de localisation , via internet, ou s'il y a pas accès internet il faut ouvrir la carte qu'il déjà télécharger et il peut voir sa localisation sur cette carte .

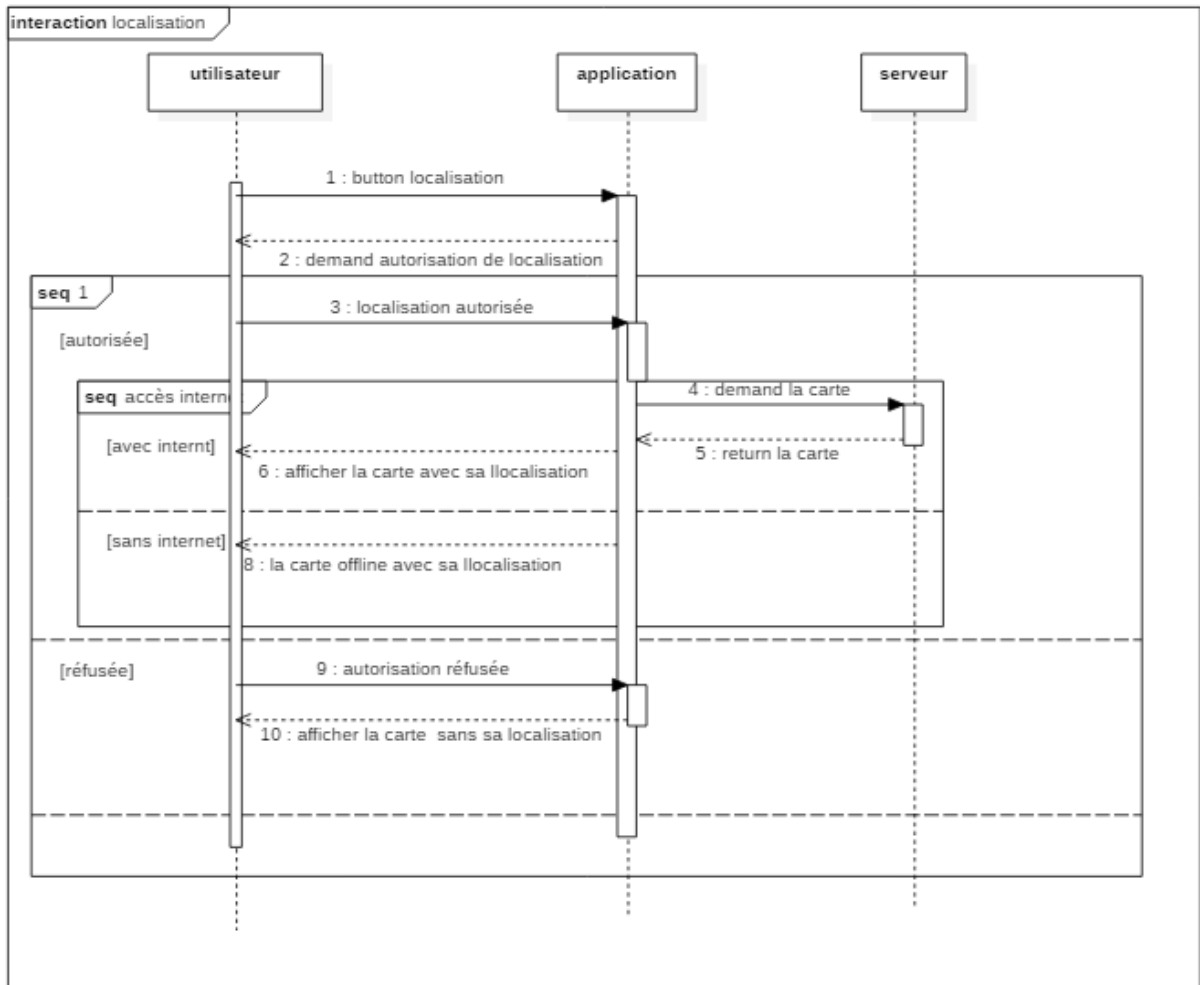


Figure 3.4 : Diagramme de cas d'utilisation "voir la localisation".

3.4.3 Diagramme de séquence du cas d'utilisation "chercher places"

Pour que l'utilisateur puisse effectuer des recherches sur la carte, il doit appuyer le bouton de recherche, puis saisir l'adresse ou le nom du lieu, et le résultat va être comme liste qui contient cette adresse, puis lorsque choisir le système affiche cette place avec un marqueur, si l'utilisateur n'a pas une connexion d'internet, le système affiche liste des places comme historique de recherche peut utiliser comme recherche offline.

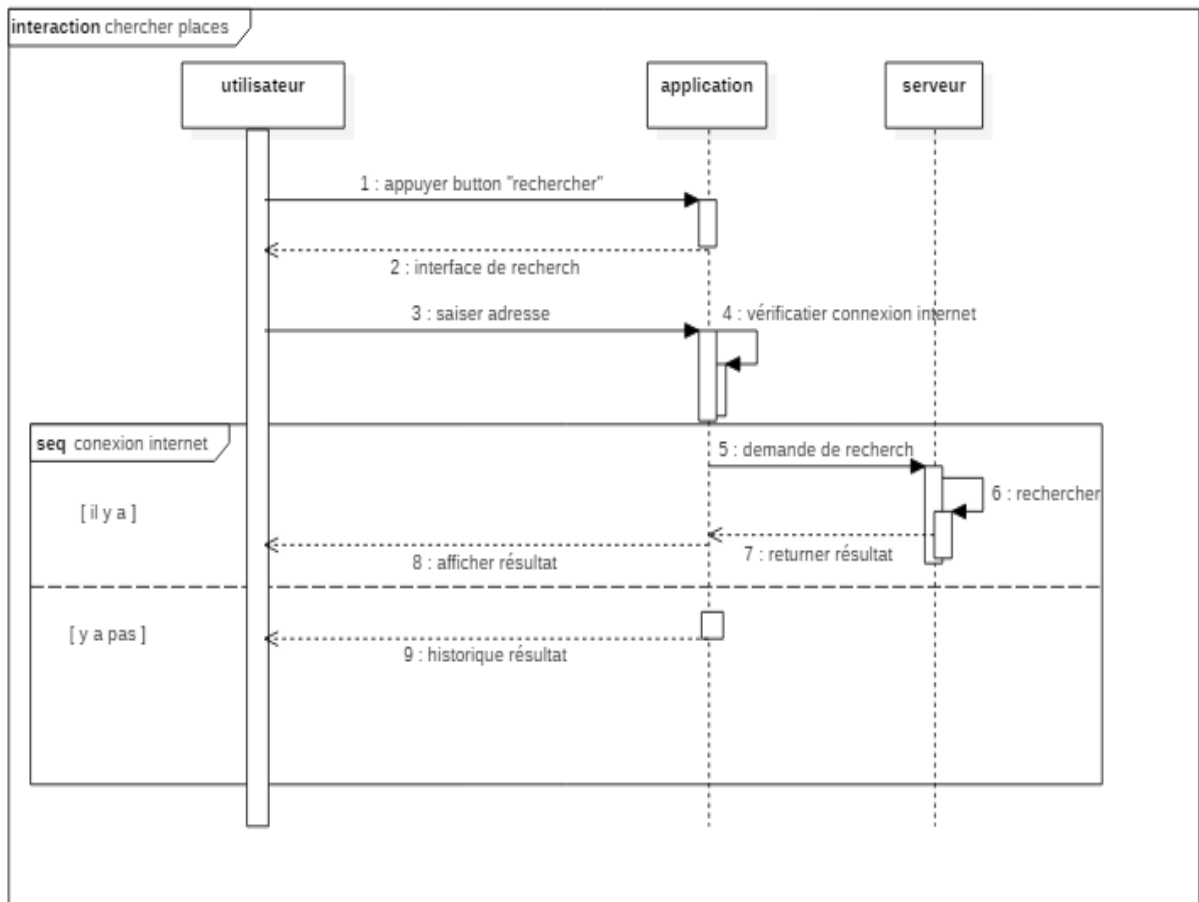


Figure 3.5 : Diagramme de cas d'utilisation "chercher places".

3.4.4 Diagramme de séquence du cas d'utilisation "ajouter point d'intérêt"

Pour que l'utilisateur lorsque découvrir un nouveau point d'intérêt pendant sa randonnée il peut ajouter ce point à la carte pour une autre randonnée, il doit d'abord appuyer le bouton de l'ajout, puis appuyer un peu long sur la carte où le point qu'il veut ajouter, puis remplir un formulaire retourné par l'application et appuyer sur envoyer, après la vérification du formulaire, si le point est valide il sera ajouté et sera visible pour tous les utilisateurs

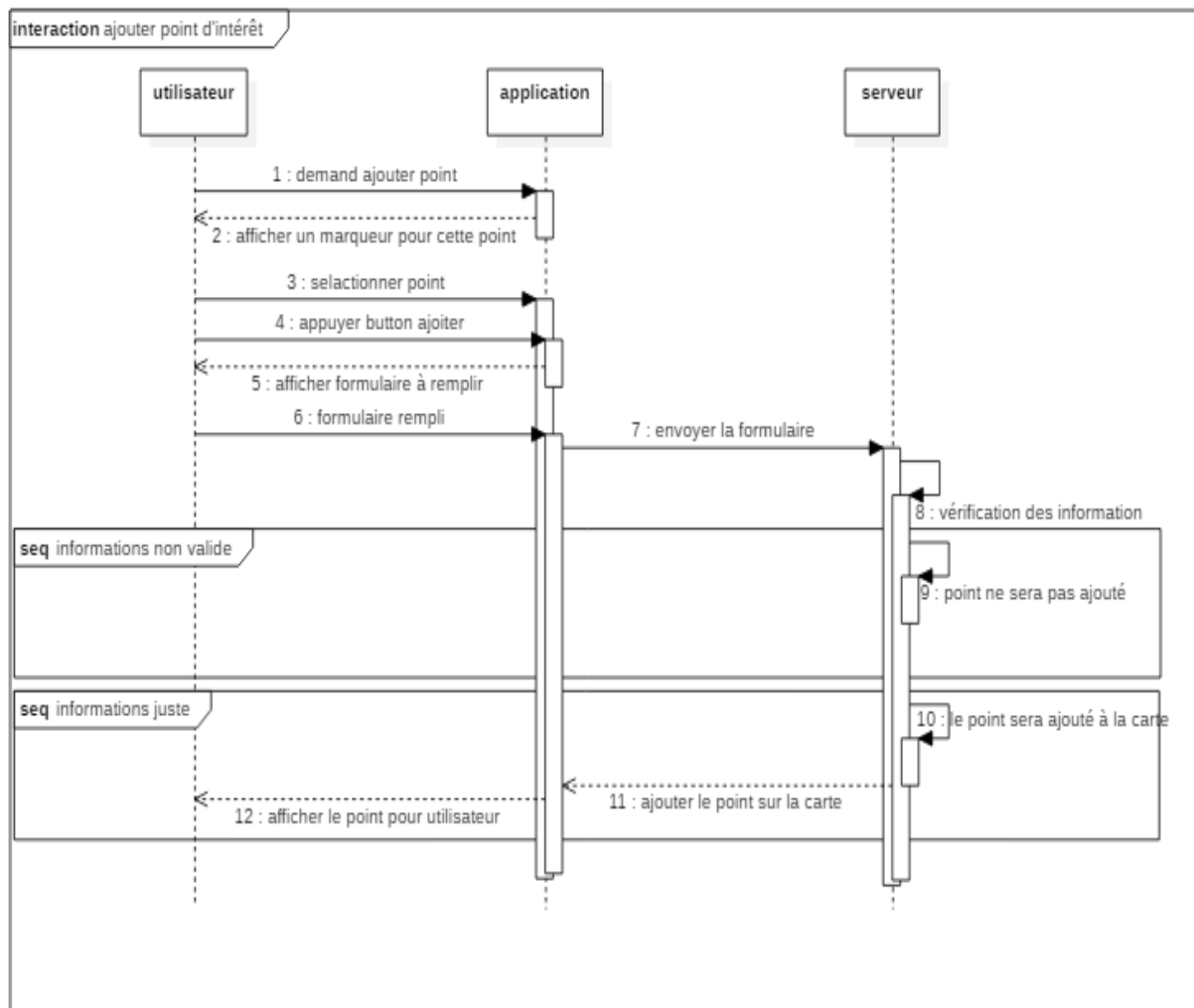


Figure 3.6 : Diagramme de cas d'utilisation "ajouter point d'intérêt".

3.4.5 Diagramme de séquence du cas d'utilisation "gérer les cartes téléchargées"

Pour que l'utilisateur peut gérer ses cartes qui sont déjà téléchargé, il peut supprimer ou naviguer vers, il doit d'abord accède à la liste des cartes à travers le menu, puis faire ces utilisations.

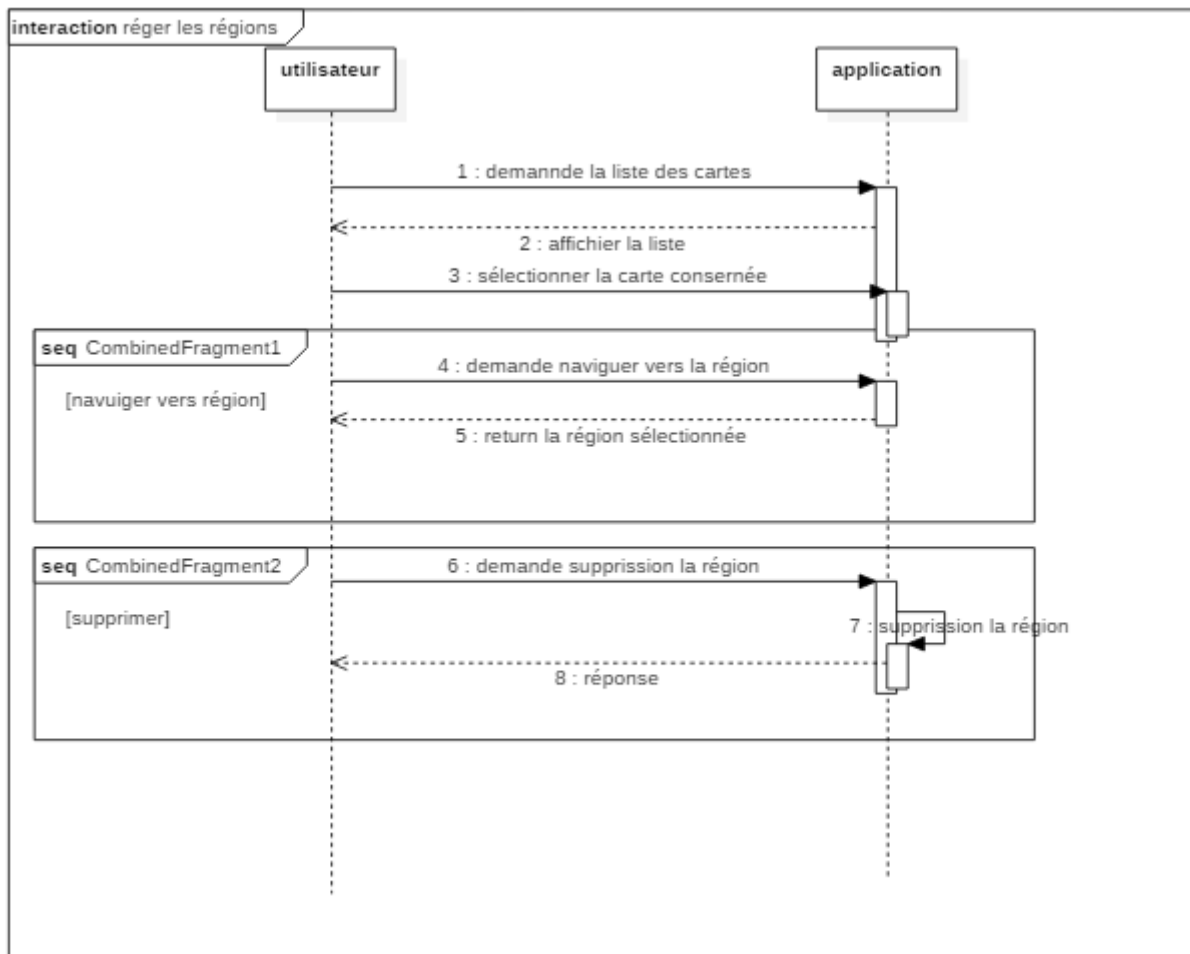


Figure 3.7 : Diagramme de cas d'utilisation "gérer les cartes téléchargées".

3.4.6 Diagramme de séquence du cas d'utilisation "naviguer vers la destination "

Pour que l'utilisateur voir le chemin de sa localisation vers la destination avec le plus court chemin par vélo ou à pied ou bien par véhicule et après avoir le chemin il peut démarrer la navigation similaire avec les virages et les avertissements et l'état de la circulation etc...

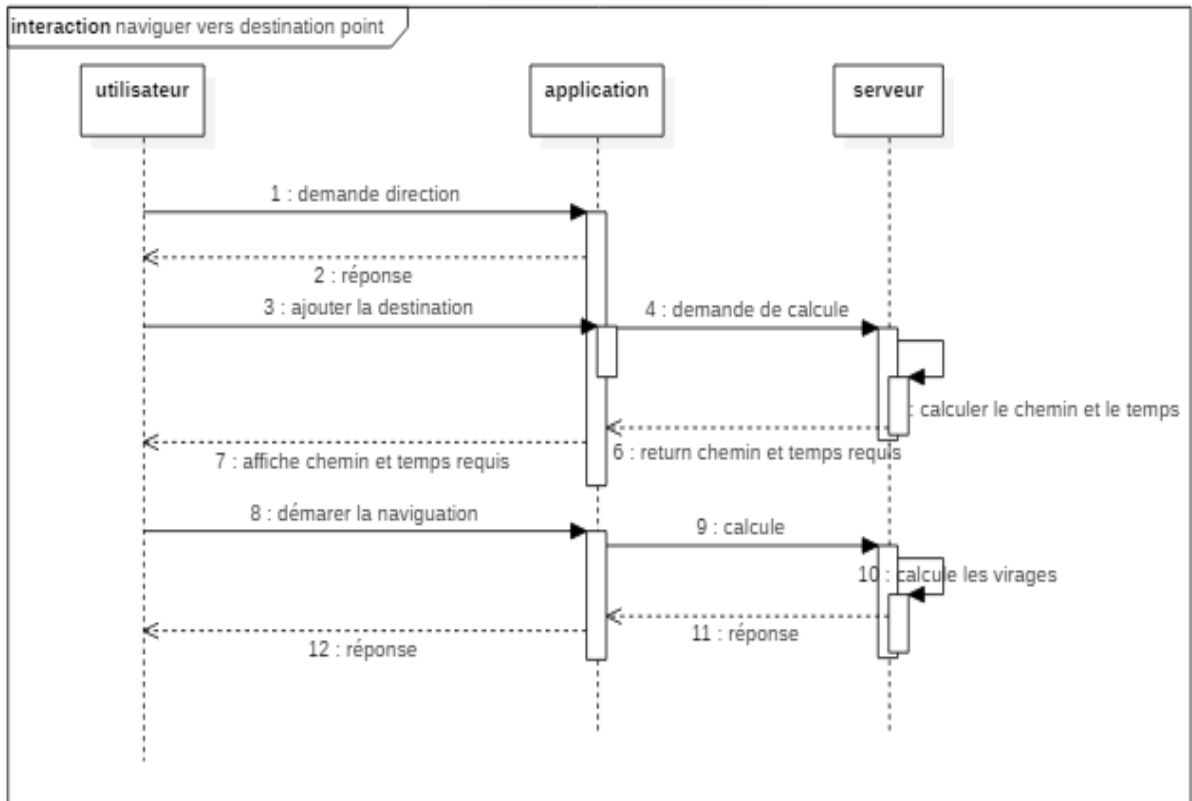


Figure 3.8 : Diagramme de cas d'utilisation "naviguer vers destination".

3.4.7 Diagramme de séquence du cas d'utilisation "enregistrer places "

Pour que l'utilisateur peut garder les places visitées ou favorites pour aider à un 'autre visite etc...

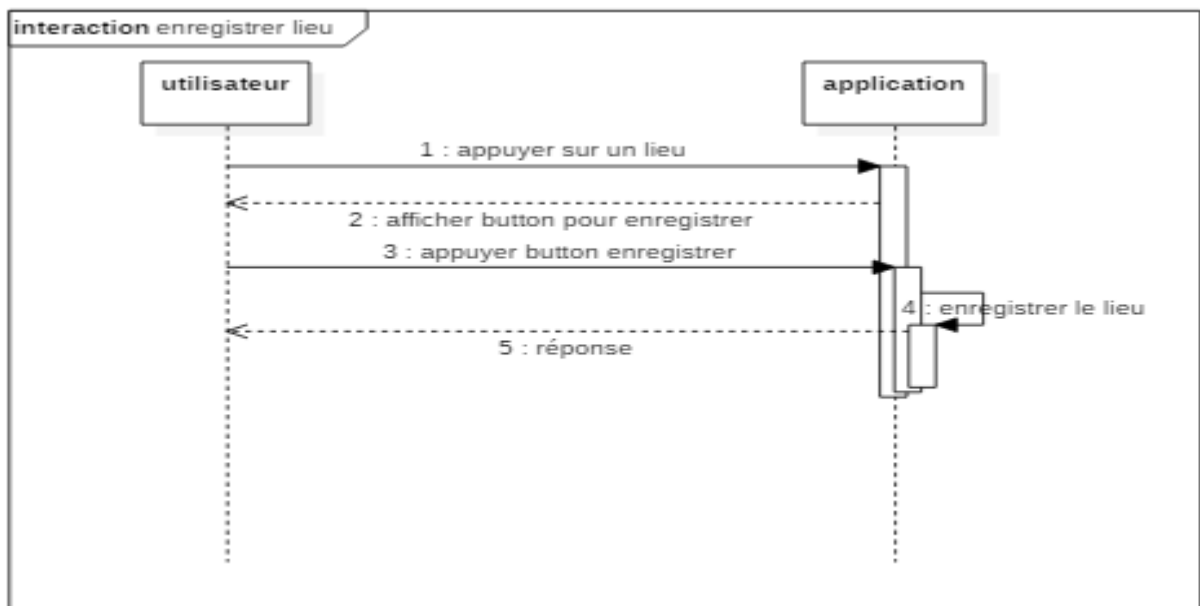


Figure 3.9 : Diagramme de cas d'utilisation " enregistrer places ".

3.4.8 Diagramme de séquence du cas d'utilisation "gérer les lieux enregistrés"

Pour que l'utilisateur peut gérer ses lieux qui sont déjà enregistrés, il peut supprimer ou naviguer vers, il doit d'abord accède à la liste des cartes à travers le menu, puis faire ces utilisations.

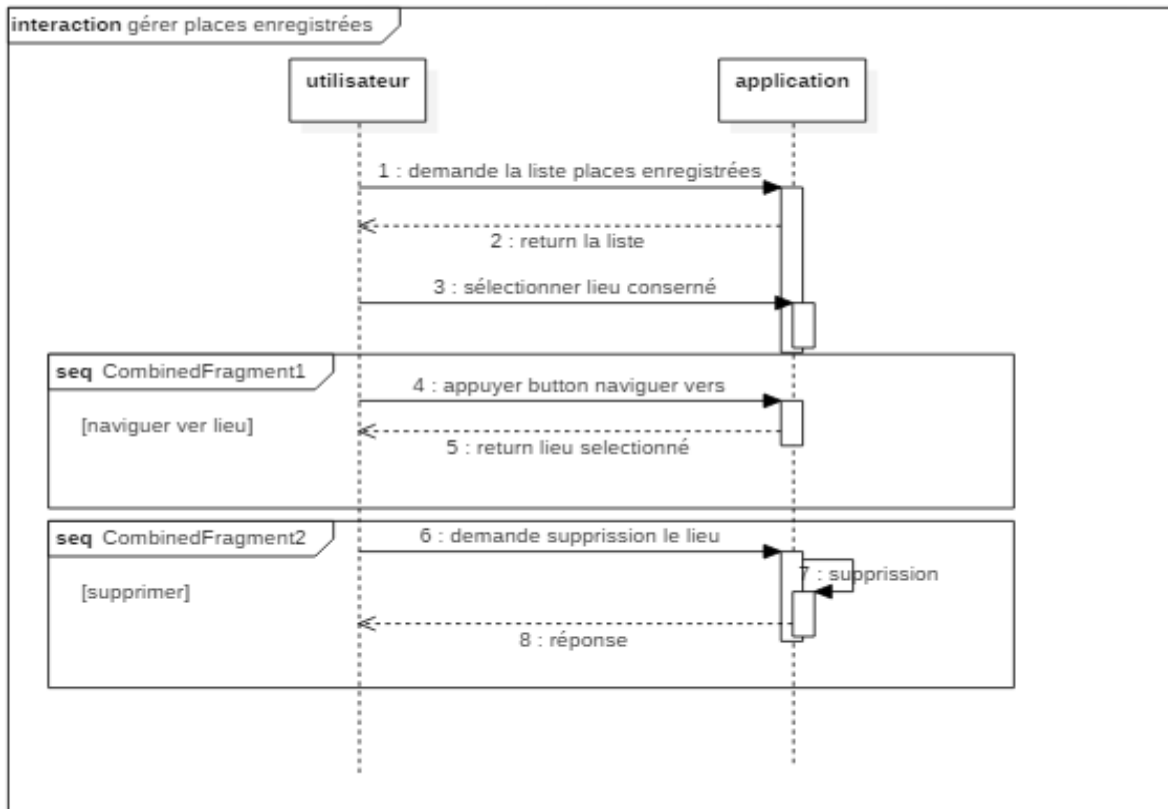


Figure 3.10 : Diagramme de cas d'utilisation " gérer les lieux enregistrés ".

3.4.9 Diagramme de séquence du cas d'utilisation "voir la boussole Qibla "

Pour que l'utilisateur peut voir la direction du Qibla à tout moment avec ou sans accès à l'internet quand il veut prier, il faut d'abord autorise le service de localisation de son appareil.

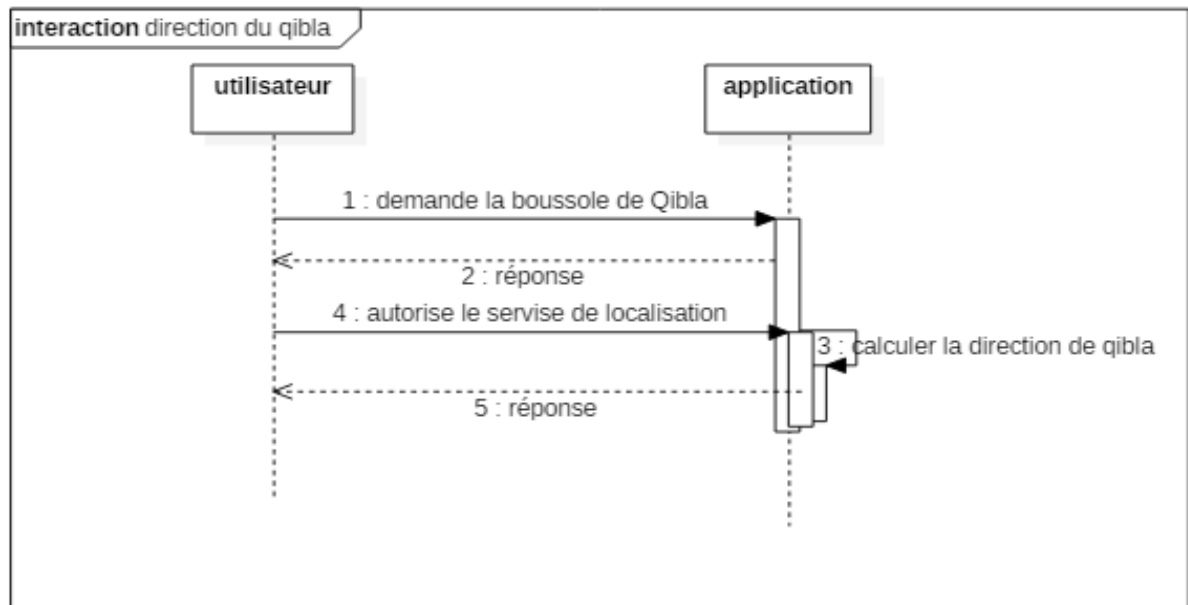


Figure 3.11 : Diagramme de cas d'utilisation "direction de Qibla".

3.5 Patrons de conception (Design Pattern)

En informatique, et plus particulièrement en développement logiciel, un patron de conception (design pattern) est un arrangement caractéristique de modules, reconnu comme bonne pratique en réponse à un problème de conception d'un logiciel. Il décrit une solution standard, utilisable dans la conception de différents logiciels.

Les patrons utilisés dans notre projet sont les suivants :

3.5.1 Patron client/serveur à 3-tiers

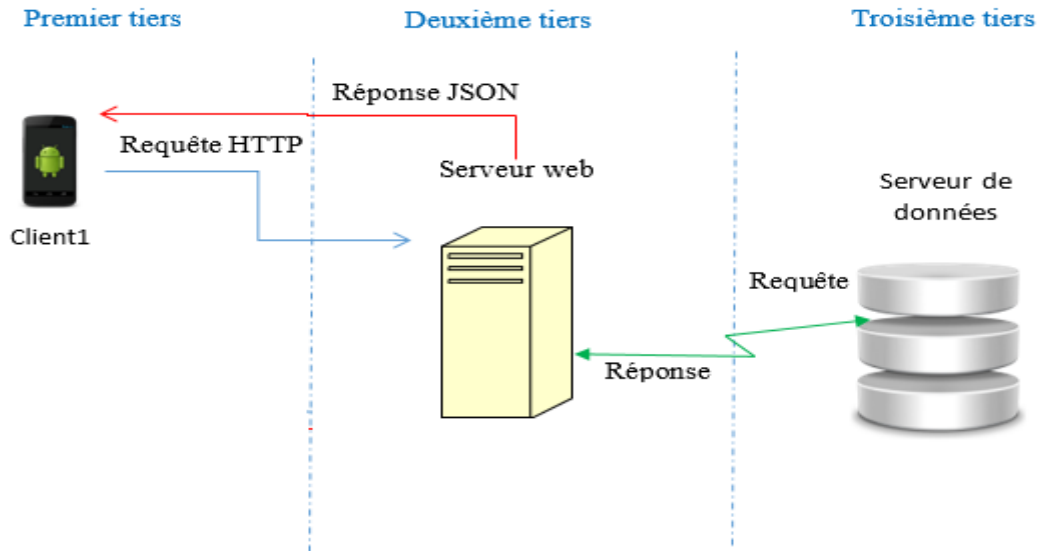


Figure 3.12 : l'architecture "3-tiers".

3.5.2 Patron Modèle-vue-contrôleur

Le patron d'architecture logicielle MVC, est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leurs architectures respectives.

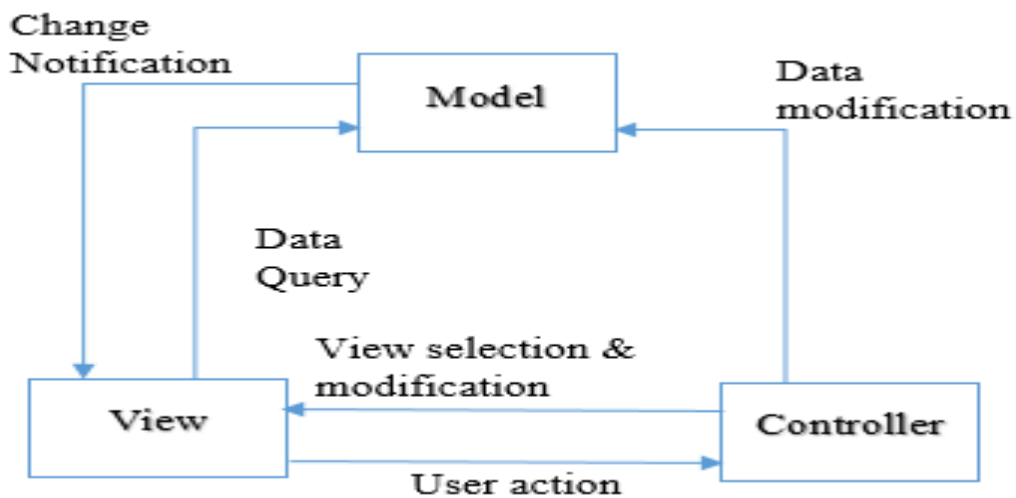


Figure 3.13 : Patron MVC (Modèle, Vue, Contrôleur)

Ce modèle de conception impose donc une séparation en 3 couches :

- **Le modèle** : Il représente les données de l'application .Il définit aussi l'interaction avec la base de données et le traitement de ces données.
- **La vue** : Elle représente l'interface utilisateur, ce avec quoi il interagit. Elle n'effectue aucun traitement, elle se contente simplement d'afficher les données que lui fournit le modèle. Il peut tout à fait y avoir plusieurs vues qui présentent les données d'un même modèle.
- **Le contrôleur** : Il gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante.

Conclusion

Nous avons présenté dans ce chapitre la phase de conception de notre projet qui contient les diagrammes de séquence des cas d'utilisations, qui nous ont aidés à décrire d'une façon détaillé, le fonctionnement de système dans le but de faciliter la réalisation.

Dans le chapitre suivant nous entamons l'étape finale de notre projet qui est l'étape de réalisation et de développement.

Chapitre 4. Réalisation

Introduction

La réalisation représente l'étape qui suit immédiatement la phase de conception. C'est l'aboutissement final de notre projet et c'est la raison d'être du projet lui-même.

Dans ce chapitre nous allons présenter dans un premier lieu l'environnement matériel et logiciel de développement, par la suite, nous décrirons la phase d'implémentation en nous basant sur quelques interfaces.

4.1 Environnement de travail

4.1.1 Environnement matériel

Pour la réalisation de notre projet, nous avons utilisé un ordinateur HP caractérisé par :

- Système d'exploitation : Windows 10.
- Processeur : Intel R Core(TM) i5 4300U 2.40 GHz.
- Mémoire vive : 8 Go.
- Disque Dur : 750 Go.

Pour les différentes étapes de test, d'installation et le déploiement de l'application nous avons eu besoin d'une terminale mobile supportant le système d'exploitation Android dont les caractéristiques sont les suivantes :

- Nom de l'appareil : HUAWEI P 10 lite.
- Système d'exploitation : Android 8.0 (Oreo)
- Connexion: 4G, EDGE, GPRS.
- Mémoire vive : 4 Go.
- Disque Dur : 64 Go.

4.1.2 Environnement logiciel

4.1.2.1 Le langage

JAVA : « Java est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables. » [28]



Figure 4.1 : Logo java

4.1.2.2 Les outils

Android Studio : Android Studio est un environnement de développement intégré (IDE) pour le développement sur la plateforme Android. Il a été annoncé en mai 2013. Android est disponible librement sous la licence Apache 2.0. Basé sur le logiciel IDEA de JetBrains 'IntelliJ', Android Studio est conçu spécifiquement pour le développement Android. Il est disponible en téléchargement sur les systèmes d'exploitation ; Windows, Mac OS et Linux. [29]

Android Studio permet principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android. Il propose aussi des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des différents types et tailles d'écrans avec des résolutions variées simultanément [30].



Figure 4.2 : Logo Android Studio

Java Development Kit (JDK) : «Le Java Development Kit (JDK) désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en bytecode destiné à la machine virtuelle Java.»[31]



Figure 4.3 : Logo JDK

Android Software Development Kit (SDK) : Le SDK est un ensemble d'outils que met à disposition Google afin de nous permettre de développer des applications pour Android. Il est disponible pour Windows, MacOs X et linux et inclut des outils ainsi qu'un émulateur Android pour exécuter des applications.[32]



Figure 4.4 : Logo SDK

API de géolocalisation de Mapbox (Acces token) : L'API « Acces token» est un kit de développement logiciel disponible de façon libre fournie par l'entreprise Mapbox sur le web, qui utilisé pour configurer et afficher le contenu de la base de donnée et pour d'autre utilisation de développement des applications web et Android et IOS et même pour Unity.

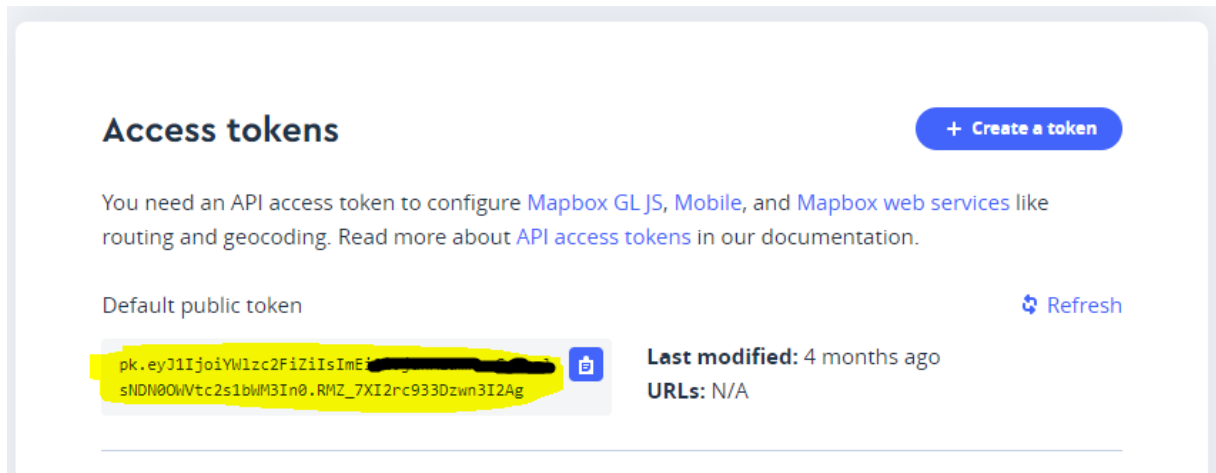


Figure 4.5 : le jeton d'accès.

Mapbox Studio : Mapbox Studio est l'application de Mapbox qui permettant de gérer les données géographiques et de concevoir des styles de carte personnalisés. Mapbox Studio utilisé pour créer et concevoir une carte conforme à propre spécifications en téléchargeant et en modifiant les propre données du développeur , en utilisant les tuiles fournies par Mapbox, en ajoutant des polices et des icônes personnalisées ou en affinant les styles de carte de modèle intégrés. Avec Mapbox Studio, la gestion des données et le contrôle de la conception sont plus facile.[33]

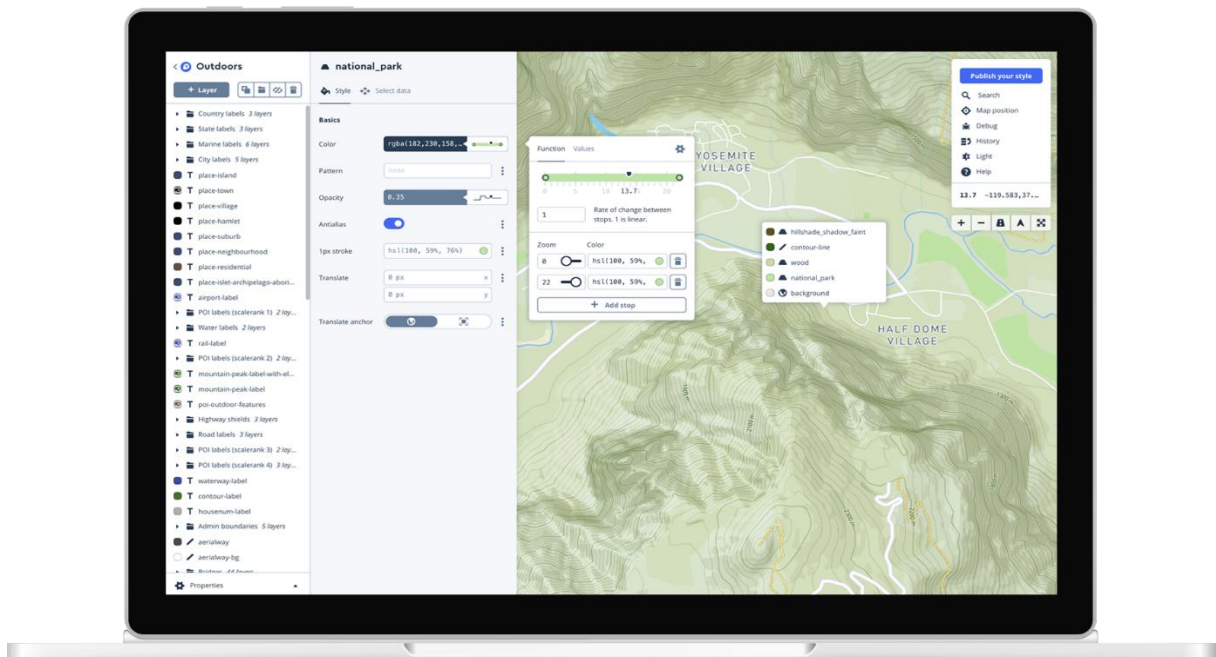


Figure 4.6 : Mapbox studio.

4.2 Structure de l'application

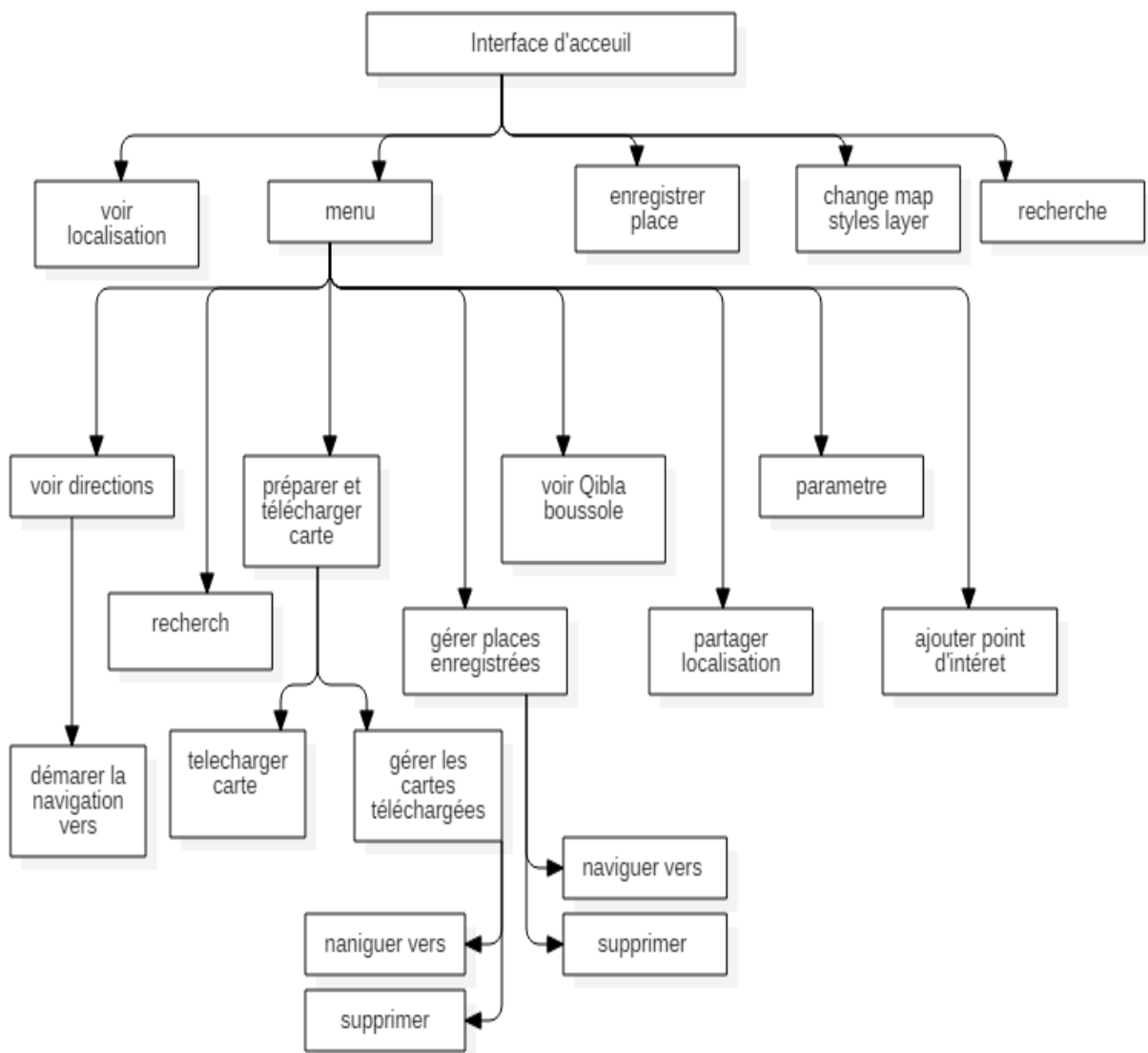


Figure 4.7 : Plan de l'application

4.3 Captures d'interfaces de l'application

Dans cette partie nous allons présenter les interfaces de notre application avec des explications importantes.

La (**Figure 4.8**) ci-dessous illustre l'Interface du logo de l'application. Cette page dure trois secondes au maximum.



Figure 4.8: logo de l'application

Lorsque l'application se lance, l'interface d'accueil comme présenté sur la (**Figure 4.9**) ci-dessous se lance et affiche la carte géographique avec le bouton de localisation et le bouton de recherche dans le couté bas à droite, et dans le haut on peut voir le nom de l'application et deux boutons à droite et à gauche du nom ces bouton on va expliquer ses roules dans les interfaces suivants.

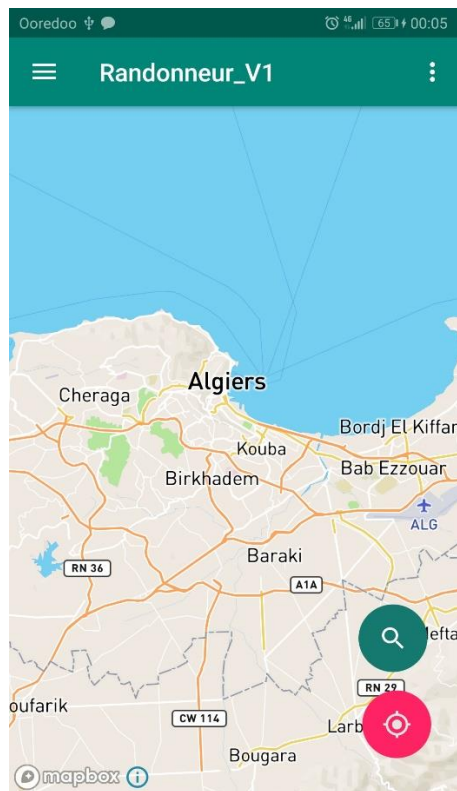


Figure 4.9: Interface d'accueil

Lorsque l'utilisateur clique sur le bouton de "voir localisation", une interface de dialogue s'affiche pour demander l'autorisation d'activer le service de GPS si n'est activé. Comme la (Figure 4.10) ci-dessous.

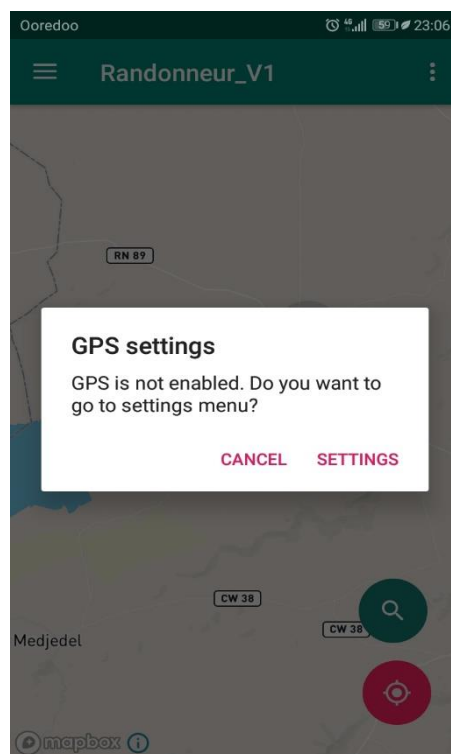


Figure 4.10 : demande GPS

Puis ouvrir les paramètres de téléphone et choisir le service de localisation comme la (**Figure 4.11**) ci-dessous

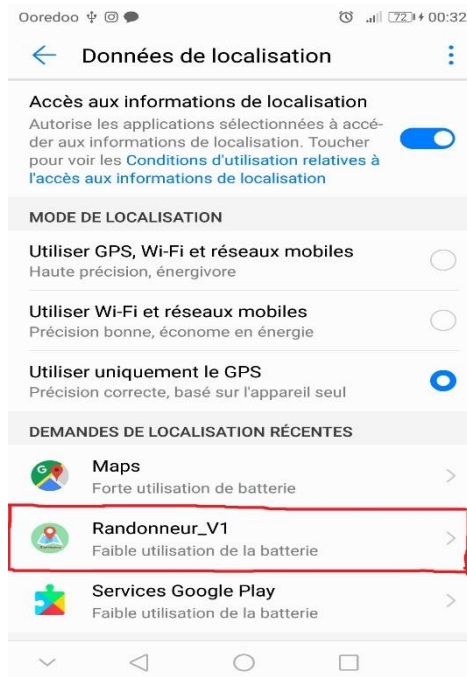


Figure 4.11 : autorisation GPS

Et si le service de localisation est activé l'application va afficher la localisation sur la carte avec la possibilité de se localiser à tous moments, comme (**Figure 4.12**).

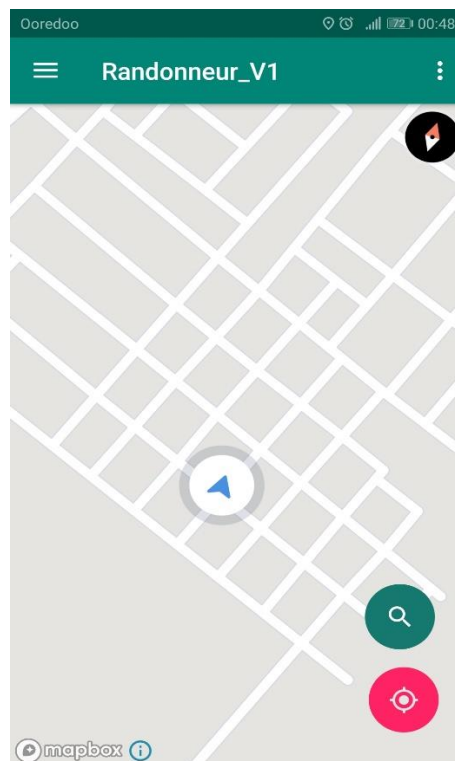


Figure 4.12 : voir localisation

Chapitre 4. Réalisation

Lorsque l'utilisateur appuie sur le bouton de recherche une interface de recherche s'affiche sur la (Figure 4.13) ci-dessous



Figure 4.13 : Interface de recherche

Et quand choisir l'adresse demandée l'application affiche cette adresse sur la carte comme la (Figure 4.14) ci-dessous.

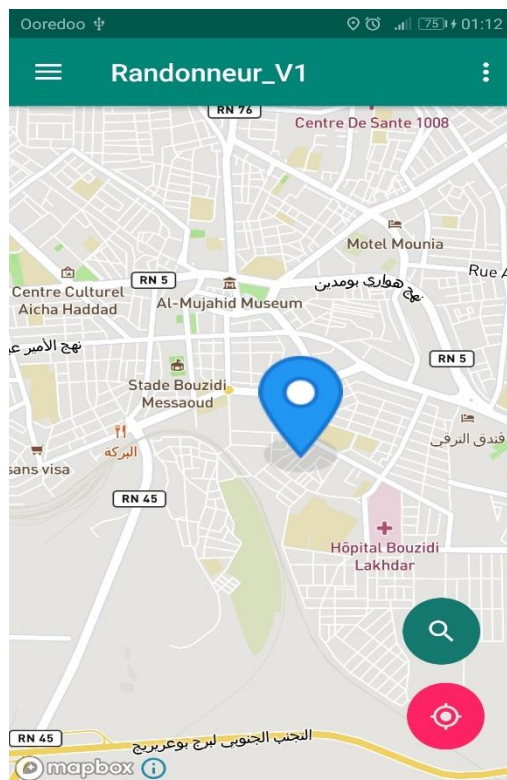


Figure 4.14 : résultat de recherche

Lorsque l'utilisateur veut accéder aux autres fonctionnalités de notre application il suffit de glisser son doigt de la gauche vers la droite u appuyer sur le button de menu qu'en haut gauche comme, la (**Figure 4.15**) ci-dessous.

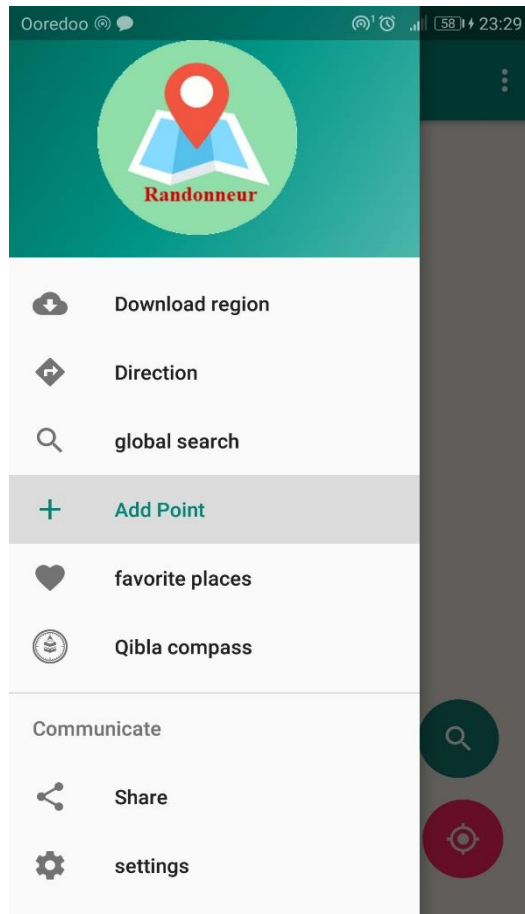


Figure 4.15 : le menu de l'application

Quand l'utilisateur a besoin de télécharger une région pour faire une randonnée ou pour gérer ses cartes qui sont déjà téléchargées il doit appuyer sur le button de «Download region» et quand il appuye, une interface s'affiche comme, la (**Figure 4.16**) ci-dessous.

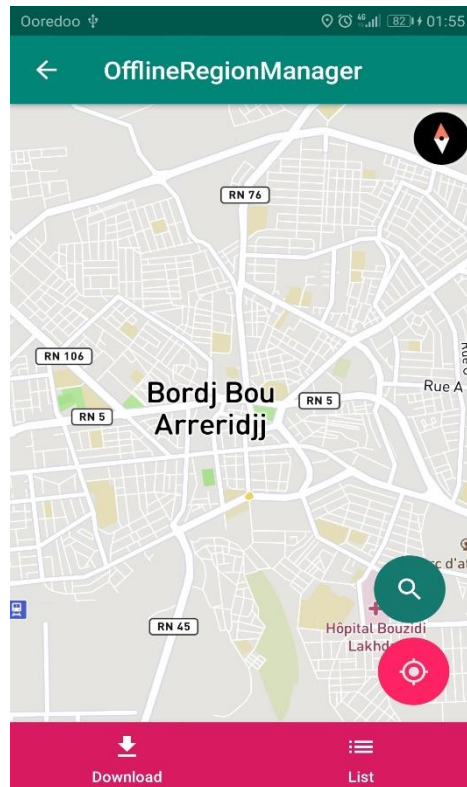


Figure 4.16 : interface de préparer et télécharger.

Si par exemple l'utilisateur veut télécharger la région de « M'sila », après avoir positionné l'écran à la région qui veut télécharger, appuyer sur le bouton « Download » et l'application rend une interface de dialogue pour saisir le nom de la région et pour confirmer le téléchargement comme la (**Figure 4.17**) ci-dessous.

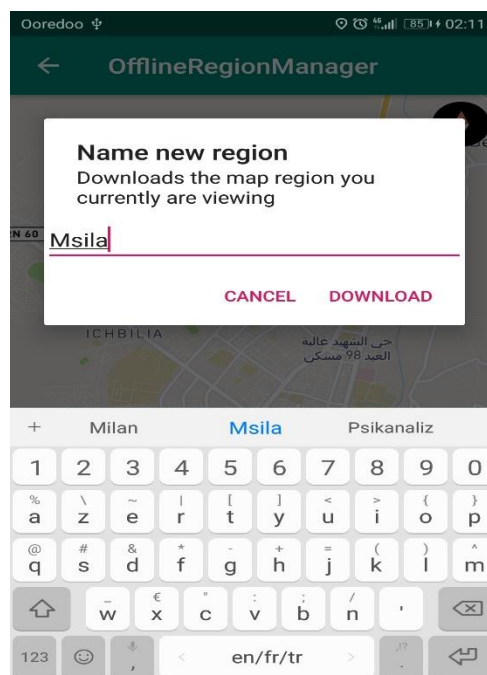


Figure 4.17 : saisir nom et confirmer le téléchargement.

Après avoir appuyé sur le bouton « download » de la (Figure 4.17), l'application démarre le téléchargement de la région avec une (Progress bar) comme la (Figure 4.18) ci-dessous.

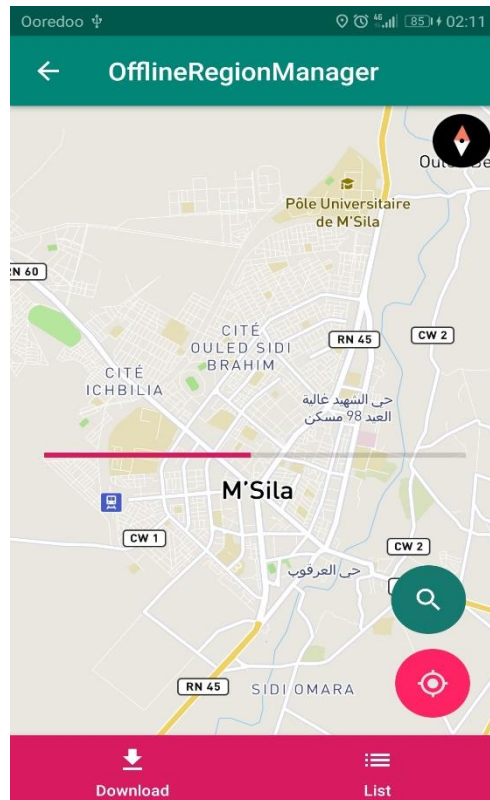


Figure 4.18 : démarrage le téléchargement

Lorsque le téléchargement est terminé la région va être ajoutée dans la liste des régions téléchargées, et quand l'utilisateur veut gérer ses régions comme naviguer vers une région ou comme supprimer une région qu'il n'a pas besoin encore il suffit d'appuyer sur le bouton « List » et peut gérer ses régions comme (Figure 4.19) ci-dessous.

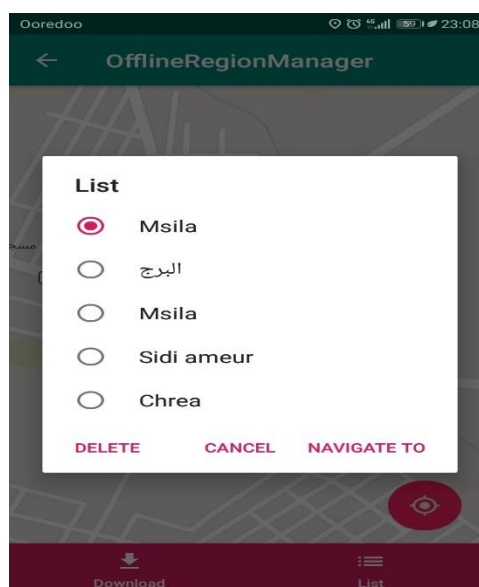


Figure 4.19 : gérer la liste des régions

Chapitre 4. Réalisation

Lorsque l'utilisateur veut voir le chemin de sa localisation vers une autre il suffit de naviguer au menu puis sélectionner « Directions » puis une interface comme la (**Figure 4.20**) ci-dessous.

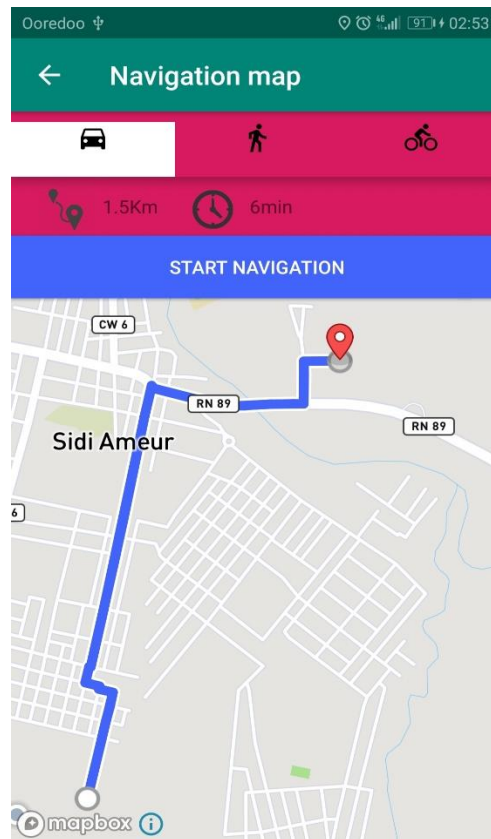


Figure 4.20 : voir le chemin

Lorsque le chemin est visible une button « START NAVIGATION » est devenu active pour une fois l'utilisateur veut démarrer la navigation similaire il peut naviguer et ça sera comme la (**Figure 4.21**) ci-dessous.

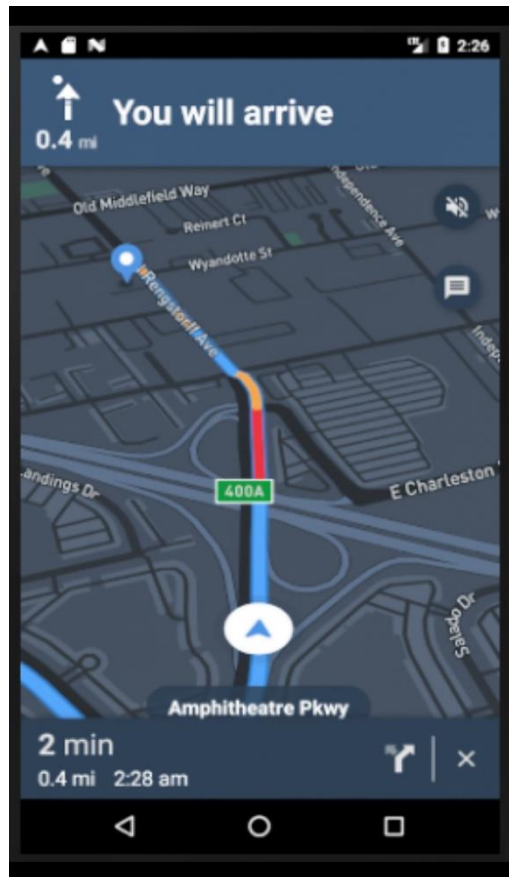


Figure 4.21 : démarrage la navigation

Quand l'utilisateur découvrir un nouveau point d'intérêt l'application donne la possibilité pour ajouter ce point sur la carte pour aider les nouveaux utilisateurs de télécharger les régions avec des nouvelles données, il suffit de naviguer vers «Add Point » à travers le menu, une interface s'affiche pour laisser l'utilisateur marquer le point sur la carte qui active un bouton pour compléter l'opération comme la (**Figure 4.22**) ci-dessous.

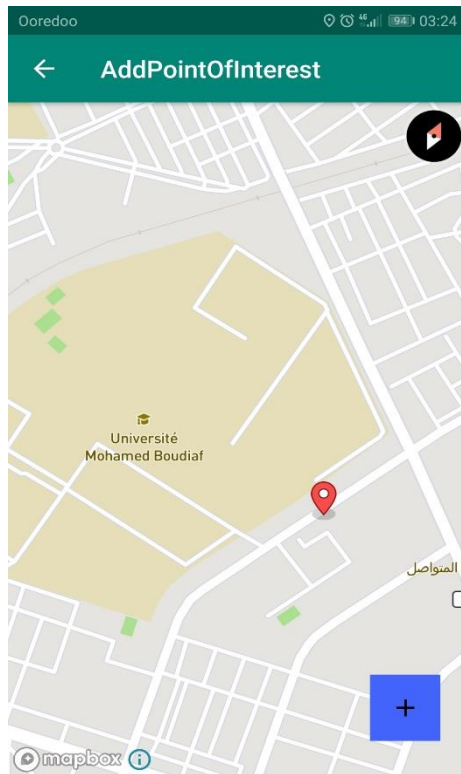


Figure 4.22 : marquer le point

Et lorsque appuyer sur le button «plus» un formulaire à remplir s’affiche et après remplir les informations appuyer le button « send », comme la **(Figure 4.23)** ci-dessous

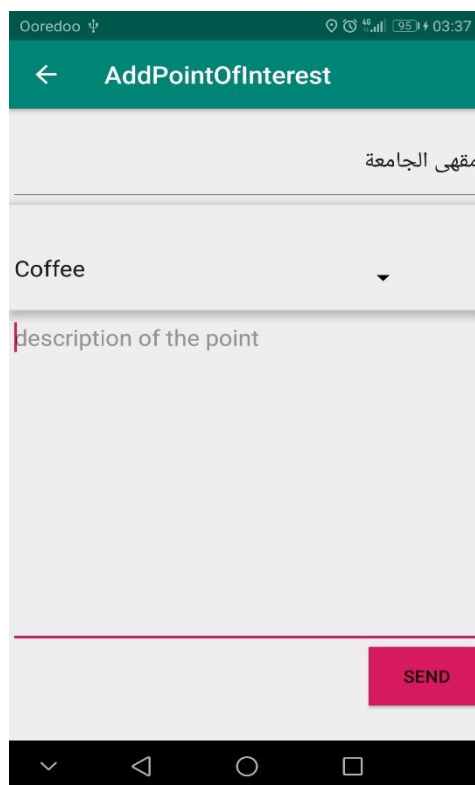


Figure 4.23 : remplir les informations

Et aussi quand l'utilisateur visite place et veut l'enregistrer il suffit d'appuyer sur cette place puis l'application affiche une marqueur sur lui-même et affiche un cœur rouge vide et après appuyer sur ce cœur, la place est sera enregistrée et le cœur sera complètement rouge et une fois appuyer sur la carte le marqueur et le cœur deviennent invisibles, comme la (Figure 4.24) ci-dessous

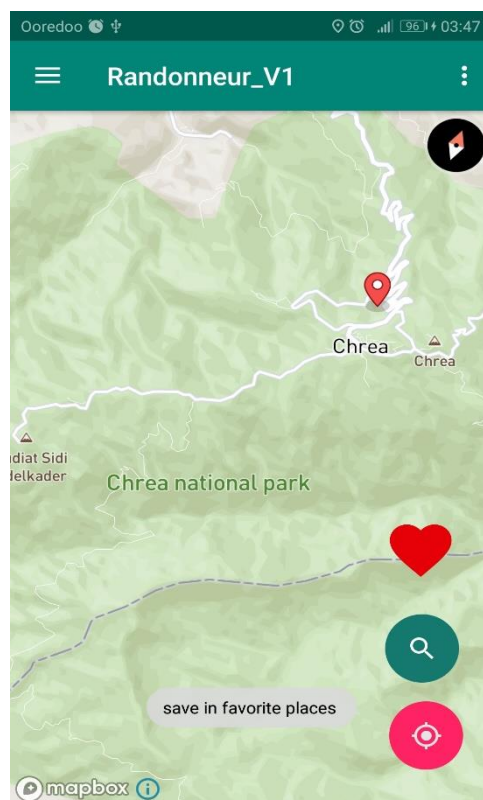


Figure 4.24: enregistrer place

Et une fois il veut gérer ses places comme naviguer vers ou pour supprimer une places, il suffit de naviguer vers « favorite places » à travers le menu puis une liste de ses places s'affiche, il peut naviguer vers une place comme il peut la supprimer comme la (Figure 4.25) ci-dessous

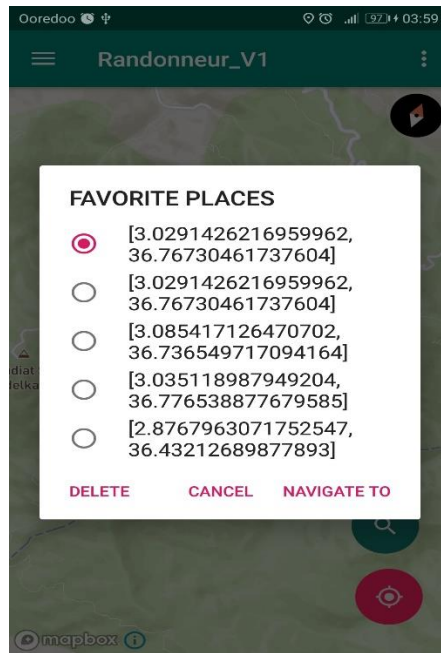


Figure 4.25 : gérer places enregistrer

Quand l'utilisateur ou le randonneur ne connaît pas la direction de la Qibla, il suffit de naviguer vers « Qibla compass » à travers le menu pour afficher la boussole, mais d'abord la boussole doit obtenir sa localisation à travers GPS puis calculer le degré à travers le nord et la localisation de Kaaba comme la (**Figure 4.26**) ci-dessous

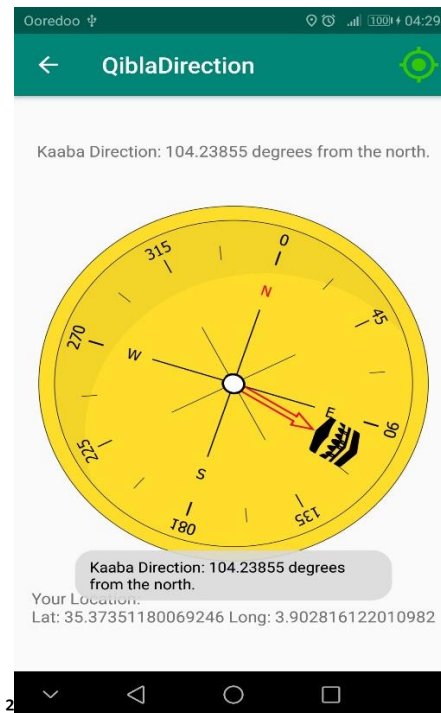


Figure 4.26 : direction de Qibla

Chapitre 4. Réalisation

L'utilisateur peut changer la langue de la carte à travers le menu puis appuyer sur « settings », un dialogue va afficher pour choisir une des trois langues arabe ou English ou français comme la (Figure 4.27) et la (Figure 4.28) ci-dessous

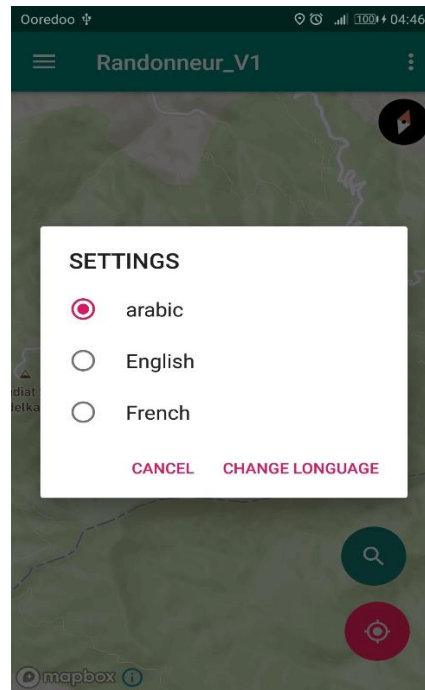


Figure 4.27 : sélectionner la langue

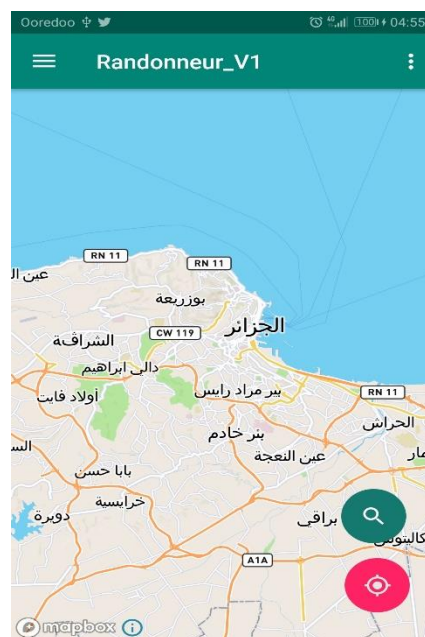


Figure 4.28 : changer la langue

La (Figure 4.29) ci-dessous explique comment l'utilisateur changer les styles de la carte.

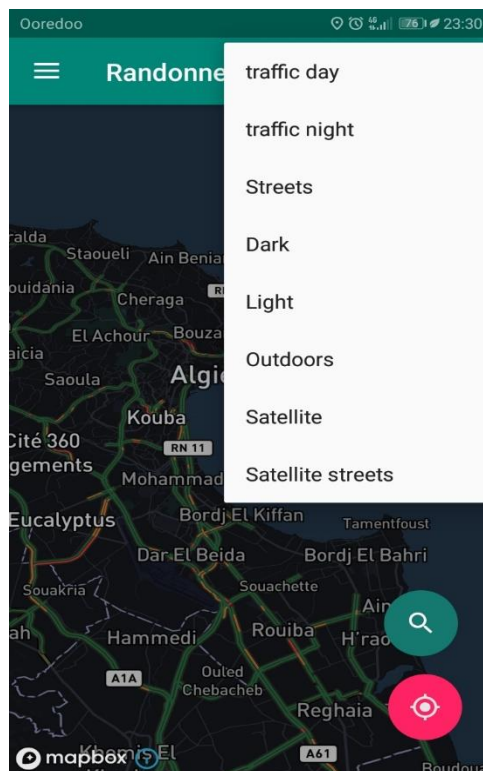


Figure 4.29 : changer le style

Conclusion

La phase de réalisation est l'étape la plus importante dans le cycle de vie d'une application. Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement, les outils et les langages de développement associés à notre système. En effet, nous avons achevé l'implémentation tout en respectant la conception élaborée.

Conclusion générale et Perspectives

Face à l'importance d'aider les randonneurs à la randonnée, nous avons étudié, conçu et réalisé à travers de ce travail un système (application) mobile qui permet d'aider à la randonnée à travers de préparer et télécharger la carte pour la utilisée au long de leur trajet sans accès internet, il permet aussi à l'utilisateur de se localiser à tous moment sans accès, localiser et enregistrer les points importants qu'ils vont explorer lors de la randonnée, puis ajouter ces points qu'ils ont découverts, détecter la direction précisément de Qibla ça aussi sans accès internet, et quelques autres options requis l'accès internet talque le partage de localisation et de voir direction et navigation similaire, etc.

Nous avons dans un premier temps présenté les données géographiques en faisant un aperçu rapide sur les types de données utilisées dans les cartes géographique et les données du projet qui nous avons utilisés pour réaliser notre système. Puis nous avons présenté dans la deuxième partie l'univers Android, en parlant d'abord sur l'informatique et les applications mobiles, puis nous avons expliqué c'est quoi le système Android et ses applications. Ensuite nous avons défini la phase de la conception qu'est importante pour mieux comprendre les fonctionnalités de notre application, et en fin nous avons fait la réalisation qu'est le beurre de notre travail, afin de mettre en œuvre notre solution à la problématique.

Bien que notre application n'est pas encore fini, nous avons l'intention de la compléter et l'améliorer en terme design (ergonomie), et augmenter les fonctionnalités hors connexion tel que la recherche à proximité, les directions doit complètement offline, et améliorer la méthode d'ajouter les points d'intérêt, ajouter d'autres langues à l'application afin d'attirer un grand nombre d'utilisateurs, et déploiement de l'application sur Play store.

Webographie

- [1] “<https://www.herodote.net/almanach-ID-3291.php>.” consulté le 26/05 /2019
- [2] “ <https://appligo.biz/lhistoire-du-smartphone/>.” consulté le 26/05/2019
- [3] “ https://docs.qgis.org/2.8/fr/docs/gentle_gis_introduction/ consulté le 8/02 /2019
- [4] “ <http://www.bel-horizon.eu/la-cartographie/cartes-raster-tuilees/osm-presentation.html>.” consulté le 8/02 /2019
- [5] “<https://www.diva-portal.org/smash/get/diva2:374640/FULLTEXT01.pdf>.” (Jia, 2010, p. 5). consulté le 12/02 /2019
- [6] “<https://www.diva-portal.org/smash/get/diva2:374640/FULLTEXT01.pdf>.” (Jia, 2010). consulté le 12/02 /2019
- [7] “https://wiki.openstreetmap.org/wiki/Category:CC_BY-SA.” consulté le 12/01 /2019
- [8] “<https://wiki.openstreetmap.org/wiki/Android>.” consulté le 12/01 /2019
- [9] “https://wiki.openstreetmap.org/wiki/Mapbox_GL.” consulté le 12/01 /2019
- [10] “ <https://docs.mapbox.com/help/troubleshooting/mobile-offline/>.” consulté le 5/01 /2019
- [11] “<https://www.mapbox.com/pricing/>.” consulté le 15/01 /2019
- [12] “<https://www.grizzlydev.com/developpement/mobilite-informatique/>.” consulté le 8/03 /2019
- [13] “https://gdt.oqlf.gouv.qc.ca/ficheoqlf.aspx?id_fiche=8360495.” consulté le 8/03 /2019
- [14] “<https://www.techopedia.com/definition/4619/personal-digital-assistant-pda>.” consulté le 10/03 /2019
- [15] “<https://www.techopedia.com/definition/2977/smartphone>.” consulté le 10/03 /2019
- [16] “<https://www.techopedia.com/definition/2662/tablet-pc>.” consulté le 10/03 /2019
- [17] “<https://www.techopedia.com/definition/2953/mobile-application>.” consulté le 10/03 /2019
- [18] “https://www.synbioz.com/blog/le_guide_ultime_du_developpement_mobile.” consulté le 12/03 /2019
- [19] “https://www.supinfo.com/articles/single/145-application-mobile-native_webhybrideconsulté.” consulté le 2 /04 /2019
- [21]“<https://www.phonandroid.com/toute-l-histoire-et-la-chronologie-d-android-dossier.html>.” consulté le 2/04 /2019
- [25] “<http://www.definitions-marketing.com/definition/application-android>.” consulté le 2/04 /2019

- [28]-“<http://www.futura-sciences.com/tech/definitions/internet-java-485/>” consulté le 20/05/2019
- [29]-“<http://www.zdnet.fr/actualites/android-studio-une-version-10-pour-l-ide-de-google-39811025.html>” consulté le 19/05/2019
- [30]-“<http://tvaira.free.fr/dev/android/android-installation.html>”. consulté le 21/05 /2019
- [31]“<http://www.commentcamarche.net/contents/559-java-introduction>” consulté le 21/05 /2019
- [32] <https://www.techopedia.com/definition/4220/android-sdk>.” consulté le 21/05 /2019
- [33] <https://docs.mapbox.com/studio-manual/overview/>.” consulté le 21/02 /2019

Bibliographie

- [20] S. P. Hébuterne Sylvain, *Android Guide de développement d'applications Java pour Smartphones et Tablettes*. 2ème édition, 2014.
- [22] H. Sylvain, *Développez une application Android programmation en Java sous Android Studio*. Septembre 2015.
- [23] N. Elenkov, *Android Security Internals An in-Depth Guide to Android's Security Architecture*. 2015.
- [24] U. Serge, *Développez une application Android Programmation en Java sous Eclipse*. 2ème édition, juin 2013.
- [26] R. Meier, *Professional Android Application Development*. 2012. 7/04 /2019
- [27] B. Nazim, *Android 5 les fondamentaux du développement d'applications Java*. Avril2015.

ملخص

هذا العمل هو نهاية مشروع الدراسة، وهدفه هو تطوير تطبيق للهاتف المحمول لمساعدة المتجولون في رحلتهم. لقد بدأنا بجزء نظري يتكون من الخرائط ونظام الاندرويد، متبوعًا بمفهوم تخطيطي للحل المقترح. للتنفيذ، استخدمنا بيئة التطوير المتكاملة ستوديو الاندرويد وحزمة ماب بوكس للانرويد وخدمة الويب رمز الوصول لماب بوكس.

الكلمات الرئيسية: تطبيق أندرويد، رحلة، متجول، خارج التغطية

Résumé

Ce travail est le projet de fin d'étude, et son but est de développer une application mobile pour aider les randonneurs à leur randonnée. Nous avons commencé par une partie théorique qui consiste les cartes géographiques et le system Android, suivie d'une conception architecturale de la solution envisagée. Pour l'implémentation, nous avons utilisé l'environnement de développement intégré Android Studio, Mapbox-Sdk Android et L'API jeton d'accès Mapbox, le Mapbox studio.

Mots clés : Application Android, randonnée, randonneur, hors connexion.

Abstract

This work is end of study project, and this aim is to develop a mobile application to help hikers to their hike. We started with a theoretical part consisting of the maps and the Android system, followed by an architectural conception of the proposed solution. For the implementation, we used the integrated development environment Android Studio, Mapbox-Sdk and the Mapbox access token API, the Mapbox studio.

Keywords: Android application, hike, hiker, offline.