Licence 3 Informatique 2024/2025

**Cahier de Charge**

**Etudiants :**

BOUAROUA Slimane 22402112

Mohamed Idris Benarbia 22303002

**Sous la supervision de :**

Mme. Béatrice Finance

Table des Matières

[1 Introduction 3](#_Toc192234042)

[1.1 Présentation du projet 3](#_Toc192234043)

[1.2 Objectifs du projet 3](#_Toc192234044)

[1.3 Portée du projet 3](#_Toc192234045)

[2 Analyse des besoins 4](#_Toc192234046)

[2.1 Besoins fonctionnels 4](#_Toc192234047)

[2.2 Besoins non fonctionnels 4](#_Toc192234048)

[2.3 Public cible 5](#_Toc192234049)

[3 Description fonctionnelle 5](#_Toc192234050)

[3.1 Parcours utilisateur (scénario d’usage) 5](#_Toc192234051)

[3.2 Fonctionnalités principales 5](#_Toc192234052)

[4 Architecture et technologies 6](#_Toc192234053)

[4.1 Architecture générale 6](#_Toc192234054)

[4.2 Technologies utilisées 6](#_Toc192234055)

[4.4 Modèle de données 7](#_Toc192234057)

[5 Interfaces utilisateur 7](#_Toc192234058)

[5.1 Maquettes et wireframes 7](#_Toc192234059)

[5.2 Navigation et ergonomie 7](#_Toc192234060)

[6 Gestion des données 7](#_Toc192234061)

[6.1 Base de données et stockage 7](#_Toc192234062)

[6.2 Sécurité des données 7](#_Toc192234063)

[6.3 Confidentialité et RGPD 7](#_Toc192234064)

[7 Contraintes et limitations 7](#_Toc192234065)

[7.1 Contraintes techniques 8](#_Toc192234066)

[7.2 Contraintes légales 8](#_Toc192234067)

[8 Développement 8](#_Toc192234068)

[8.1 Plan de développement 8](#_Toc192234069)

[8.2 Tests et maintenance et validation 8](#_Toc192234070)

[9 Annexes 8](#_Toc192234071)

[9.1 Glossaire 8](#_Toc192234072)

[9.2 Références 8](#_Toc192234073)

# Introduction

## Présentation du projet

Notre projet SmartTrip est une application mobile Android conçue pour offrir une expérience enrichie de suivi et de gestion des trajets personnels. Pensée pour les voyageurs, l’application permet de suivre en temps réel un itinéraire GPS, de détecter automatiquement les point d’intérêts et de capturer ou ajouter des photos associées à des emplacement spécifiques.

SmartTrip vise à transformer un simple déplacement en un journal interactif de voyage. L’application permet de visualiser l’itinéraire parcourue avec de marqueurs de début et de fin, les POIs, et les photos, tout en les sauvegardant dans le cloud afin de pouvoir y accéder ultérieurement ou les partager.

Ce projet fait partie de notre semestre universitaire en Licence informatique pour le module 'Projet', portant sur la thématique 'Base de données'. Il vise à mettre en place une application mobile de grande valeur, qui intègre la géolocalisation, le traitement asynchrone, l'interface utilisateur, ainsi que l'utilisation de services tiers comme OpenStreetMap et FireBase.

## Objectifs du projet

L’objectif principal de ce projet est de concevoir et développer une application mobile qu’on a nommé SmartTrip, permettant aux utilisateurs **d’enregistrer automatiquement** leurs trajets via des capture GPS, de **détecter les points d’intérêts significatifs** qu’ils visitent, et de documenter leurs expériences à l’aide **des photos et les noms des lieux**.

Le projet vise à :

* Offrir une expérience immersive aux voyageurs en leur permettant de suivre, revivre et partager leurs itinéraires.
* Faciliter l'organisation et la conservation des souvenirs de voyage, en associant une géolocalisation, photos et descriptions
* Fournir une interface intuitive pour enregistrer automatiquement les points d’intérêts (POI) détectés selon la durée et la position, tout en donnant la possibilité d’ajouter manuellement des contenus multimédias.

L’application est conçue pour répondre aux besoins de divers type de voyageurs, sportifs, aventuriers ou curieux qui souhaitent conserver des souvenirs enrichis de leurs déplacements.

## Portée du projet

L'application est conçue exclusivement pour les smartphones Android. Après avoir installé l’application sur son téléphone, l’utilisateur doit :

* Créer un compte s’il n’en possède pas un : En resignant son nom, prénom, une adresse électronique et un mot de passe.
* Se connecter via son adresse électronique et son mot de passe dans le cas de possession de compte
* Récupérer son compte dans le cas d’oublie du mot de passe

Après être connecté, des simples appuie sur quelques boutons vont permettre :

* Lancement du suivis GPS, ainsi qu’un enregistrement automatique à la fin du trajet.
* Identification automatique des points d’intérêts selon la durée d’arrêt
* Ajout des photos manuellement, avec géolocalisation
* Visualisation des trajets passé, ou bien favoris avec les marqueurs et les itinéraires sur carte
* Partage d’un trajet afin qu’il soit visible pour les autres utilisateurs.

Le projet ne comprend pas :

* Une version IOS de l’application (pas de Cross-Platform)
* La synchronisation avec des services tiers comme Google Photos ou Google Maps Timeline
* Un mode hors-ligne
* Établissement de relation entre les utilisateurs (ami) ou bien des discussions
* Gestion et récupération avancée des géolocalisations lors de perte d’internet ou l’entrée dans des zones avec des faibles signaux (métros, anciens immeubles…)

Ces choix ont été fait dans le but de se concentrer sur les fonctionnalités essentielles tout en assurant la faisabilité dans les délais impartis du projet semestriel.

# Analyse des besoins

## Besoins fonctionnels

* Authentification des utilisateurs
  + L’utilisateur peut créer un compte avec une adresse électronique et un mot de passe
  + L’utilisateur peut récupérer son compte en changeant de mot de passe en cas d’oublie
  + L’utilisateur peut se connecter et se déconnecter de son compte
* Lancement d’un trajet
  + L’utilisateur peut démarrer un trajet via un bouton dédié
  + L’utilisateur peut nommer le trajet et spécifier la ville de départ
  + Un service de géolocalisation démarre en arrière-plan
* Suivi GPS en temps réel
  + L’application enregistre les positions GPS toutes les ….
  + Un tracé de l’itinéraire est visible sur une carte OpenStreetMap
  + L’utilisateur peut mettre en pause et reprendre le suivi du trajet
* Détection automatique des points d’intérêts
  + Si l’utilisateur reste dans une zone restreinte pendant une durée définie, un point d’intérêt est automatiquement sauvegardé
  + Le nom du lieu et l’adresse sont récupéré via une API (Nominatim/OpenStreetMap)
* Ajout de photo
  + L’utilisateur peut prendre une photo durant le trajet ou bien importer de la galerie.
  + La photo est automatiquement liée au trajet en cours
  + S elle correspond à un point d’intérêt, elle y est associée
* Stockage des données utilisateur (itinéraires, photos, points d'intérêt) sur Firebase.
  + Les trajets sont enregistrés dans une base de données en ligne
  + Chaque trajet contient : itinéraire, des points d’intérêts, des photos, un nom, dates de début et de fin.
* Consultation des trajets passé ou favoris
  + Possibilité de revoir les trajets précédents
  + Possibilité des rajouter des trajets au favoris
  + Chaque trajet peut être consulté individuellement, avec son itinéraire, ses POIs et photos affichés sur une carte OpenStreetMap
* Interface utilisateur intuitive
  + L’application propose une navigation claire entre les fragments :
    - Home : la page d’accueil avec la carte affichée dessus
    - Journey : le fragment où on peut recherche des trajets en se basant sur la ville de départ
    - Favroites : Fragment qui affiche les trajets favoris
    - Profile : Fragment pour consulter son profil, le modifier, se déconnecter ou bien accéder aux trajets passés
  + Les actions principales sont accessibles via une barre de navigation ou des Floating Action Buttons.
* Modification du Profil
  + L’utilisateur peut modifier son nom et son prénom et le mettre à jour dans la base de données
  + L’utilisateur peut aussi modifier sa photo de profil

## Besoins non fonctionnels

* L’application enregistre les données GPS en temps réel avec un intervalle de temps configurable
* Toutes les données utilisateurs (photos, trajets...) sont stockées de manière sécurisée dans FireBase avec des règles d’accès strictes
* Respect des règles de confidentialité et du RGPD.
* Compatibilité avec les versions récentes d'Android.
* Synchronisation rapide des données avec le cloud Firebase.
* Les trajets deviennent publics sauf si l’utilisateur l’accepte
* La structure du Code : bien séparé la logique de la vue
* La possibilité de rajouter des nouvelles fonctionnalités dans le futur

## Public cible

L’application s’adresse à un public large, mais bien défini, qui partage le besoin de suivre, visualiser et enrichir ses trajets avec des informations pertinentes. Voici les catégories principales :

* **Voyageurs et touristes** : amateurs d’aventures souhaitant conserver une trace de leurs déplacements, marquer les lieux visités et pouvoir les revoir ultérieurement.
* **Blogueurs et créateurs de contenu** : désireux de documenter leurs déplacements pour partager leurs expériences avec leurs communautés, enrichis de photos et de commentaire sur les lieux visités.
* **Organisateur de Road Trips :** utilisateurs souhaitant planifier, enregistrer et visualiser clairement leurs parcours pour une meilleure organisation.
* **Professionnels du tourisme :**  guides, agences ou offices de tourisme qui peuvent proposer ou analyser des itinéraires personnalisés selon les préférences de leurs clients.

Ce groupe de personnes partage une passion commune pour la valorisation des déplacements, la mémorisation des expériences et le partage de leurs parcours.

# Description fonctionnelle

## Parcours utilisateur (scénario)

1. L'utilisateur lance l'application Smart Trip et s'authentifie (inscription ou connexion).
2. Sur la page d’accueil (Home), il doit fournir les permissions de localisation
   1. Dans le cas ou il consente de partager sa localisation : La carte OpenStreetMap s’affiche avec sa position actuelle si le GPS est activé, sinon dans un endroit au hasard.
   2. Dans le cas inverse, la carte reste floue.
3. Il voir une barre de navigation en bas de l’écran avec un bouton ‘+’ au milieu, il appuie dessus et un dialogue s’ouvre pour lui demander de saisir :
   1. Le nom du voyage
   2. La ville de départ
   3. De cocher s’il accepte de rendre le trajet public ou pas
4. L’utilisateur confirme et le voyage commence.
5. Un For-Ground service se lance en arrière-plan pour la récupération des positions GPS chaque minutes
6. S’il reste plus que 30 min dans un même endroit, alors un point d’intérêt et détecté automatiquement.
7. Il peut ajouter manuellement des photos pendant son voyage
8. Il peut mettre pause à l’application pour économiser de la batterie ou pour ne pas montrer une partie de son voyage
9. A la fin, il clique sur le bouton ‘X’ pour finir le trajet.
10. Un dialogue de confirmation s’affiche pour confirmer la fin du voyage
11. Après avoir confirmé, son itinéraire sera dessiné dans la page d’accueil.
12. Il peut voir les itinéraires autres personnes en recherchant une ville de départ dans la page ‘Journey’
13. Il peut voir ses voyages favoris dans la section favoris
14. Il peut voir ses trajets précédents, les rendre publiques s’ils ne le sont pas déjà
15. Il peut modifier son profil (nom, prénom et photo de profil)

## Fonctionnalités principales

* **Authentification et gestion de compte**
  + **Description :** Les utilisateurs peuvent s’inscrire par une adresse électronique ou un compte Google, se connecter et se déconnecter.
  + **Utilité :**  Garantir une expérience personnalisée et sécurisée.
* **Lancement et enregistrement d’un itinéraire**
  + **Description :** L’utilisateur peut lancer un nouveau trajet via un bouton flottant (Floating Action Button). L’application capture et enregistre automatiquement sa position GPS toutes les … minutes jusqu’à l’arrêt manuel.
  + **Utilité :** Permet de suivre précisément le déplacement de l’utilisateur et de tracer l’itinéraire effectué à la fin du trajet.
* **Détection automatique des points d’intérêts (POI)**
  + **Description :** Si l’utilisateur reste plus de ... Minutes dans un rayon de … mètres, un point d’intérêt est automatiquement détecté.
  + **Utilité** : Identifier les arrêts importants du voyage sans action manuelle.
* **Ajout manuel des photos au trajet**
  + **Description :** L’utilisateur peut prendre des photos ou en choisir de sa galerie pour les associer à l’itinéraire ou à un POI proche.
  + **Utilité :** Enrichir le journal de voyage avec des souvenirs visuels
* **Affichage des itinéraires sur la carte**
  + **Description :** Une fois le trajet terminé, l’itinéraire est dessiné sur une carte avec des marqueurs pour le début, la fin ; les POIs et les photos.
  + **Utilité :** Visualiser clairement le parcours et les moments clés
* **Enregistrement des trajets sur Firebase**
  + **Description :** Chaque itinéraire est enregistré dans la base de données en ligne, avec tous les points et photos.
  + **Utilité :** Sauvegarder les voyages pour les consulter ou les partager
* **Consultation des anciens voyages**
  + **Description :** L’utilisateur peut accéder à une liste de tous ses anciens trajets, et afficher les détails d’un voyage spécifique.
  + **Utilité :** Garder un historique complet des voyages passés.

# Architecture et technologies

## ****Architecture générale****

L'architecture modulaire de notre projet SmartTrip est basée sur la répartition de plusieurs composants logiques, ce qui rend la séparation des responsabilités, la maintenabilité et l'évolution de l'application plus faciles.

Nous avons adopté une architecture accès proche du MVVM (Model-View-ViewModel) afin de :

* Séparer clairement les responsabilités entre l’interface utilisateur, la logique de présentation et les données
* Assurer une meilleure gestion du cycle de vie avec des composantes tels que LiveData.
* Favoriser la réutilisabilité des composants

Parmi les couches on retrouve

* Vue (View) : Il s’agit généralement des fragments. Nous avons :
  + HomeFragment : La première où l’utilisateur se retrouve après l’authentification. C’est la page ou la carte est affiché.
  + JourneyFragment : La page avec une barre de recherche pour trouver des trajets sur une ville spécifique
  + FavoriteFragment : Ici on trouve tous les trajets favoris de l’utilisateur
  + ProfileFragment : La page où l’utilisateur peut voir son profil, le modifier, accéder à ces trajets précédents ou bien se déconnecter
  + MyTripsFragment : L’utilisateur vois tous ses trajets (publiques ou pas) et peut les visualiser
  + VoyageDetailsFragment : le fragment qui permet la visualisation des trajets favoris ou passés
* Models : Ce sont les fichiers qui implémentent notre présentation de données et on trouve :
  + Itinerary : Une classe itinéraire qui facilite la gestion et la manipulation d’un itiniraire
  + PhotoMeta :
  + PhotoModel : Un model pour représenter une photo afin qu’elle soit associée à un itinéraire et un POI si c’est possible
  + PointOfInterestModel : Comme l’indique leur nom, c’est la représentation des points d’inrérêt
  + Voyage :
* Managers : Ils s’agissent des fichiers avec une ou des tâches spécifiques sur une seule partie du code.
  + JourneyManager : gére le démarrage, l’arrêt et l’enregistrement d’un trajet, ainsi que la récupération des méta données d’une photo choisie de la galerie afin de l’associer au trajet.
  + MapManager : Tous ce qui concerne le management de la map : l’ajout d’un marqueur à une position donnée, déssin d’itinéraire, affichage du dialogue de choix de photo à ajouter et la redimension des icones des marqueurs.
  + PointOfInterestDetector : Une classe qui facilite la détection d’un point d’intérêt avec un intervalle de temps configurable, ainsi que l’attribution d’un nom à un POI si c’est possible et l’ajouter à la liste.
  + VoyageAdapter :
* Helpers : On trouve les classes qui simplifient les tâches liées à une logique de domaine ou technolgoie.
  + GeoHelper : Fichier avec un seule fonction à l’intérieur. C’est l’API qui récupère le nom et l’adresse d’un POI donnée.
  + PermissionHelper : Facilite la gestion des permissions de localisation
  + PhotoHelper : Simplifie la prise de photos au cours d’un trajet ou bien le choix de la galerie, l’association d’une photo à un POI, ainsi que la récupération de la géolocalisation d’une photo choisie.
* Utils : les classes et les objets avec des méthodes génériques et réutilisables partout dans l’application, dans notre cas nous avons qu’un seul :
  + MapConfigurator : Un objet pour initialiser OSMDroid et configure la map afin de pouvoir zoomer…
* Service : le dossier qui contient tous les services sur l’application, Bound Service, en arrière-plan ou bien ForeGround qui est utilisé dans notre projet.
  + GpsTrackingService : Un foreGround service qui tourne même si l’utilisateur ne regarde pas son application. Il s’occupe de la récupération des GeoPoint dans un intervalle de temps configurable. Il possède également des fonctions pour le mettre en pause ou l’arrêter.

Les services externes :

* Firebase :
* OpenStreetMap : affichage de la carte, des itinéraires et des marqueurs.
* Nominatim (OSM API) : Pour obtenir les noms des lieux à partir de coordonnées géographiques.

Pour la communication entre les composantes, Le MainActivity agit comme un point d’ancrage principal entre les fragments. Nous avons également utilisé des événements comme :

* Bundles : Ce sont des centenaires de données passés comme un seul argument
* LiveData : les données liées au cycle de vie, permettant de réagir automatiquement aux modifications
* Companion Objects : Fonctions et variables accessibles par tout dans l’application
* Arguments : les paramètres passés lors d’un appel d’une fonction

## Technologies utilisées

Pour développer l'application, nous avons utilisé une variété cohérente de technologies modernes, qui ont été choisies pour leur compatibilité avec les besoins fonctionnels, leur robustesse, leur documentation abondante et leur intégration rapide dans un environnement Android.

Voici les principales technologies utilisées :

* **Kotlin** : Choisi pour sa compatibilité optimale avec Android, Kotlin est concis, expressif et améliore la sécurité en réduisant les erreurs courantes comme les NullPointerExceptions. Il permet également le développement multiplateforme (Android, iOS) et peut être utilisé côté backend avec Ktor.
* **Android Studio** : IDE officiel pour le développement Android, il offre des outils performants pour le débogage, la conception UI et les tests.
* **Firebase** : Utilisé pour l'authentification des utilisateurs, la base de données en temps réel et le stockage des images, Firebase simplifie la gestion des données cloud.
* **OpenStreetMap** : Bibliothèque utilisée pour intégrer des cartes libres dans l’application, permettant d’afficher des itinéraires GPS, des marqueurs personnalisés, et d’interagir avec la map sans dépendre de Google Maps.
* **Google Play Services (GPS)** : Permet un suivi GPS précis et optimisé en termes de consommation d'énergie.
* **Firebase Authentification**: Service utilisé pour l’authentification des utilisateurs. Il permet la gestion des connexions via email/mot de passe.
* **Appareil photo et galerie**: Gestion de la caméra native et des images depuis la galerie via les APIs Android moderne (ActivityResultIContracts), avec une prise en charges des métadonnées EXIF pour obtenir la localisation des photos.

## Modèle de données

Le modèle de données inclut plusieurs collections dans Firebase :

* **Utilisateurs** : Ce sont les personnes qui vont utiliser l’application et sont identifié par :
  + **IdUser**: identifiant unique pour l’utilisateur
  + **Nom**: chaine de caractère
  + **Prénom**: chaine de caractère
  + **Adresse électronique**: afin de pouvoir s’authentifier
  + **Mot de passe**: chaine de caractère
  + **Photo de profil**: photo
* **Voyages** :
  + ID : identifiant unique pour chaque voyage
  + Nom : chaine de caractère
  + Utilisateur : l’identifiant de l’utilisateur qui a réalisé le voyage
  + DateDebut : date de début do voyage
  + Date\_de\_fin : date de fin du voyage
  + villeDepart : chaine de caractère pour représenter la ville de départ
  + Points : les coordonnées GPS capturés pour tracer l’itinéraire
  + Photos : listes des photos associées au trajet
  + Public : Booléen pour distinguer si le voyage est visible pour tous le monde ou pas
* **Points d'intérêt** :
  + Name : chaine de caractère
  + Description : chaine de caractère
  + Location : une coordonnées GPS
* **Photos** :
  + **URI**: uri locale de la photo
  + **Position**: coordonnées GPS
  + **Date**: la date où la photo était prise
  + **attachedPoiName**: le nom du POI auquel la photo est rattachée
  + **urlFirebase**: url de la photo après l’upload dans firebase

# Interfaces utilisateur

## Maquettes et wireframes

## Navigation et ergonomie

# Gestion des données

## Base de données et stockage

## Sécurité des données

## Confidentialité et RGPD

# Contraintes et limitations

## Contraintes techniques

## Contraintes légales

# Développement

## Plan de développement

## Tests et maintenance et validation

# Annexes

## Glossaire

## Références