

N° d'inscription : 0019-80-20

Année universitaire : 2023-2024



UNIVERSITÉ D'ANTANANARIVO

INSTITUT D'ENSEIGNEMENT SUPERIEUR D'ANTSIRABE – VAKINANKARATRA

MEMOIRE DE FIN D'ETUDES

en vue de l'obtention du diplôme de LICENCE

Domaine : Sciences de l'Ingénieur

Mention : AUTOMATISME ET INFORMATIQUE

Parcours : GENIE LOGICIEL

Par : RKOTOMALALA Herimandranto Fandresena

APPLICATION MOBILE DE RESERVATION DE TRANSPORT PUBLIC

Soutenu le _____, devant les Membres du Jury composés de :

Président de Jury : Mr ou Mme Nom et Prénoms

Examineurs :

Mr ou Mme Nom et Prénoms

Mr ou Mme Nom et Prénoms

Encadreur pédagogique : Mr ou Mme Nom et Prénoms

Encadreur professionnel : Mr ou Mme Nom et Prénoms

N° d'inscription : 0019-80-20

Année universitaire : 2023-2024



UNIVERSITÉ D'ANTANANARIVO

INSTITUT D'ENSEIGNEMENT SUPERIEUR D'ANTSIRABE – VAKINANKARATRA

MEMOIRE DE FIN D'ETUDES

en vue de l'obtention du diplôme de LICENCE

Domaine : Sciences de l'Ingénieur

Mention : AUTOMATISME ET INFORMATIQUE

Parcours : GENIE LOGICIEL

Par : RKOTOMALALA Herimandranto Fandresena

APPLICATION MOBILE DE RESERVATION DE TRANSPORT PUBLIC

Soutenu le _____, devant les Membres du Jury composés de :

Président de Jury : Mr ou Mme Nom et Prénoms

Examineurs :

Mr ou Mme Nom et Prénoms

Mr ou Mme Nom et Prénoms

Encadreur pédagogique : Mr ou Mme Nom et Prénoms

Encadreur professionnel : Mr ou Mme Nom et Prénoms

AVANT PROPOS

Après trois années d'études au sein de l'Institut d'Enseignement Supérieur d'Antsirabe Vakinankaratra (IES-AV) dans la mention Automatismes et Informatique du parcours Génie Logiciel, ce mémoire consiste à l'obtention du diplôme de fin d'étude premier cycle.

Ce mémoire porte sur la création d'une application mobile de réservation de transport public. Dans un contexte où les plateformes existantes présentent des limitations significatives, il est crucial de développer une solution qui réponde aux besoins modernes des utilisateurs. Le but de notre recherche est de concevoir une application intuitive et performante, permettant non seulement de réserver des trajets en toute simplicité, mais aussi de suivre la position des véhicules en temps réel et de gérer les réservations de manière flexible et sécurisée.

Nous nous sommes penchés sur les technologies actuelles et les meilleures pratiques en matière de développement d'applications mobiles pour proposer un outil qui améliore l'expérience utilisateur et optimise la gestion des services de transport public. Notre projet vise à combler les lacunes des systèmes existants et à offrir une alternative moderne et efficace, adaptée aux exigences du monde actuel.

FISAORANA

Voalohany indrindra dia isaorana Andriamanitra lehibe nanome fotoana sy hery ary fahasalamana nahafahana nanantanteraka izao dingana izao. Manaraka izany dia isaorana manokana ireto olona voalaza anarana manaraka ireto :

- Andriamatoa RAVOLOMANANA Mamy Raoul, Profesora, filohan'ny Oniverisitean'Antananarivo, amin'ny andraikitra lehibe izay iantsorohany amin'ny fitantanana sy fampandrianana ny raharham-pampianarana manerana ny oniverisitean'Antananarivo, manaraka izany
- Andriamatoa ANTSONAINARIVONY Onomamandimby, dokotera, talehan'ny "Institut d'Enseignement Supérieur d'Antsirabe Vakinankaratra", amin'ny fandriandrany ny eto amin'ny sekoly ka ahafahana manao izao asa fikaroana izao.
- Ramatoa RANAIVOSOA Mamitiana Lalaonirina Olivette, Dokotera, lehiben'ny sampampihofanana "Automatisme et Informatique", izay sady talen'izao fikarohana izao.
- Andriamatoa ANDRIANAIVO Herifidy M, nanoro lalana ahy sy nanampy indrindra tamin'ny fanatanterahana an'ity Fikarohana ity, teo koa izy nanoro hevitra synandrisika ahy hatramin'ny farany. Andrianaivo Herifidy M.

Ao koa anefa ireo vondronan'ny mpitsara izao hatrikasa izao izay tarihin'i :

- ..
- ..
- ..

Nohon'ny faneken'izy ireo mba hitsara izao atrikasa izao.

- Ireo mpampianatra rehetra izay nampita fahalalana sy nanome fiofanana ho ahyhatramin'ny taona voalohany ka hatramin'izao taona fahatelo izao.
- Ireo Ray amandReny nikolokolo sy nanabe ary nanohana nandritry ny dingana izay efa nodiavina hatr'izay ka mbola manohana hatrany amin'izao fotoana lehibe izao.
- Ireo fianakaviana, ny tapaka sy namana rehetra izay nanampy na ara-kevitra na ara-tsaina nahatontosana izao atrik'asa izao.

REMERCIEMENTS

Tout d'abord, nous remercions Dieu de nous avoir donné le temps, la force et la santé pour mener à bien ce processus. Nous souhaitons ensuite remercier tout particulièrement les personnes suivantes :

- Monsieur RAVOLOMANANA Mamy Raoul, Professeur, Président de l'Université d'Antananarivo dans le rôle important qu'il joue dans la gestion et l'enregistrement des affaires éducatives dans toute l'Université d'Antananarivo, suite à cela.
- Monsieur ANTSONAINARIVONY Onomamandimby, Docteur, directeur de l'Institut d'Enseignement Supérieur d'Antsirabe Vakinankaratra, dans l'organisation de l'école pour que ces recherches puissent se faire.
- Madame RANIVOSOA Mamitiana Lalaonirina Olivette, Docteur, responsable du département de formation « Automatisation et Informatique » qui est également le directeur de cette recherche.
- Monsieur ANDRIANAIVO Herifidy M, qui m'a guidé et aidé le plus dans la réalisation de cette Recherche, il m'a également conseillé et encouragé jusqu'au bout.

Mais il y a aussi les groupes de juges de cet atelier qui est animé par :

- ..
- ...
- ...

En raison de leur accord pour juger cet atelier.

- Tous les professeurs qui m'ont transmis des connaissances et m'ont formé de la première à la troisième année.
- Mes parents qui ont nourri, éduqué et soutenu tout au long du processus qui s'est déroulé jusqu'à présent et qui continuent de soutenir en cette période importante.
- Toutes les familles, parents et amis qui ont contribué à rendre cet atelier possible.

TABLE DES MATIERES

Table des matières

AVANT PROPOS.....	
FISAORANA.....	
REMERCIEMENTS.....	
TABLE DES MATIERES.....	
LISTE DES ABREVIATIONS.....	
LISTES DES TABLEAUX ET DES FIGURES.....	
INTRODUCTION ET POSITION DU PROBLEME.....	
CHAPITRE 1 ETUDE DE L’EXISTANT ET FONCTIONNALITER DE L’APPLICATION.....	
CHAPITRE 2 CONCEPTION DU PROJET.....	
CHAPITRE 3 PRESENTATION ET EVALUATION DE L’APPLICATION.....	
3.2.4.2 SQLite.....	
CONCLUSION ET PERSPECTIVES.....	
ANNEXES.....	
BIBLIOGRAPHIE.....	
FICHE DE RENSEIGNEMENTS.....	
FAMINTINANA SY TENY MANADANJA.....	lxii
RESUME ET MOTS CLES.....	lxii
ABSTRACT AND KEYS WORDS.....	lxii

LISTE DES ABREVIATIONS

API	Application Programming Interface
BPMN	Business Process Model and Notation
CPU	Central Processing Unit
CPU	Central Processing Unit
CSRF	Cross-Site Request Forgery
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IoT	Internet of Things (Internet des objets)
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
MySQL	My Structured Query Language
OSM	OpenStreetMap
ORM	Object-Relational Mapping
OWASP	Open Web Application Security Project
REST	Representational State Transfer
SQL	Structured Query Language
SSH	Secure Shell
TLS	Transport Layer Security
UNIX	Uniplexed Information and Computing System
UML	Unified Modeling Language
UV	Ultraviolet
VPN	Virtual Private Network

vscode	Visual Studio Code
XML	Extensible Markup Language
XSS	Cross-Site Scripting

LISTES DES TABLEAUX ET DES FIGURES

1. Liste des tableaux

Table 2.2: Scénario alternatif du cas d'utilisation n°1: « Ajouter une trajet ».....	28
Table 2.3: Scénario nominal du cas d'utilisation n°2: « Réserver une siège ».....	29
Table 2.4: Scénario alternatif du cas d'utilisation n°2: « Réserver une siège ».....	29
Table 2.5: Scénario nominal du cas d'utilisation N°3 : « Mettre à jour un trajet ».....	30
Table 2.6: Scénario alternatif du cas d'utilisation N°3 : « Mettre à jour un trajet ».....	31
Table 2.7: Scénario nominal du cas d'utilisation N°4 : « Rechercher un trajet ».....	31
Table 2.8: Scénario alternatif du cas d'utilisation N°4 : « Rechercher un trajet ».....	32
Table 2.9: Scénario nominal du cas d'utilisation N°5 : « Gérer les réservations ».....	32
Table 2.10: Scénario alternatif du cas d'utilisation N°5 : « Gérer les réservations ».....	33
2.11. Table: La cardinalité.....	44

2. Liste des figures

Figure 1.1: Le site web Cotisse-Transport.....	12
Figure 1.2: Le site taxi-brousse.....	13
Figure 1.3 Site web SoatransPlus.....	15
Figure 1.4 : Interface de l'application mobile TaxiBe.....	17
Figure 2.1: Diagramme de cas d'utilisation.....	26
Figure 2.2 : Diagramme de séquences du cas d'utilisation n°1.....	35
Figure 2.3 : Diagramme de séquences du cas d'utilisation n°2.....	35
Figure 2.4 : Diagramme de séquences du cas d'utilisation n°3.....	36
Figure 2.5 : Diagramme de séquences du cas d'utilisation n°4.....	36
Figure 2.6 : Diagramme de séquences du cas d'utilisation n°5.....	37
Figure 2.7: Diagramme d'activité du cas d'utilisation n°1.....	39

Figure 2.8: Diagramme d'activité du cas d'utilisation n°2.....	39
Figure 2.9: Diagramme d'activité du cas d'utilisation N°3.....	40
Figure 2.10: Diagramme d'activité du cas d'utilisation N°4.....	40
Figure 2.11: Diagramme d'activité du cas d'utilisation N°5 : « Gérer les réservations ».....	41
Figure 2.12 Diagramme de classe du projet.....	45

3. Liste des photos

4. Liste des annexes

INTRODUCTION ET POSITION DU PROBLEME

Depuis l'ère industrielle, le progrès technologique a révolutionné divers secteurs, y compris les transports publics. Les défis de l'optimisation du temps, de la gestion des données et de l'amélioration des services poussent les entreprises à rechercher des solutions informatiques innovantes. Les systèmes traditionnels de réservation de transport public posent souvent des problèmes d'accès aux informations en temps réel, d'efficacité opérationnelle et de gestion des paiements, entraînant des désagréments pour les utilisateurs et les chauffeurs.

L'omniprésence des téléphones mobiles ouvre des opportunités pour améliorer ces systèmes. Les applications mobiles permettent aux utilisateurs de réserver des sièges, de suivre leurs réservations et de gérer leurs trajets de manière interactive. Elles simplifient aussi la création, la mise à jour et la gestion des trajets pour les chauffeurs, tout en offrant une communication en temps réel.

Notre projet de mémoire vise à développer une application mobile pour la gestion des réservations de transport public. Cette application facilitera la réservation de sièges et la gestion des trajets pour les utilisateurs, et simplifiera les opérations pour les chauffeurs. Elle garantira également une gestion sécurisée des paiements et améliorera la communication via des notifications en temps réel.

Pour démontrer la pertinence de ce projet, ce mémoire se divisera en plusieurs chapitres : une analyse de l'existant pour identifier les limites des systèmes actuels, une description de la conception du projet avec les choix techniques et les étapes de développement, et enfin, une présentation et évaluation de l'application développée, mettant en avant ses fonctionnalités et ses performances.

Ce projet vise à répondre aux besoins actuels de gestion des réservations de transport public tout en anticipant les évolutions futures, en proposant des solutions robustes et évolutives. L'intégration de l'application mobile souligne l'importance croissante des technologies mobiles dans la transformation des services de transport public.

CHAPITRE 1 ETUDE DE L'EXISTANT ET FONCTIONNALITER DE L'APPLICATION

1.1 Introduction

L'évolution technologique et la popularité croissante des smartphones ont révolutionné le secteur du transport public, rendant les déplacements plus accessibles et mieux organisés grâce aux applications mobiles. Pour concevoir une application innovante et efficace dédiée à la réservation de transport public à Madagascar, il est essentiel de comprendre l'état actuel du marché et d'identifier les forces et les faiblesses des solutions existantes. Ce chapitre propose une étude approfondie des applications actuellement disponibles, en se concentrant sur leurs fonctionnalités, leurs avantages et leurs limitations. En examinant ces éléments, nous pourrions identifier les opportunités d'amélioration et poser les bases nécessaires au développement d'une application qui répondra aux besoins spécifiques des utilisateurs malagasy

1.2 Etude de l'existant

1.2.1 Site web Cotisse-transport

Cotisse-Transport.mg est un site web dédié à la gestion des réservations de transport public. Le site propose des services visant à faciliter la réservation de billets, à améliorer l'expérience des utilisateurs et à offrir des fonctionnalités supplémentaires pour les voyageurs.

- ◆ **Réservation en ligne** : La fonctionnalité principale de Cotisse-Transport.mg est la possibilité de réserver des billets en ligne. Les utilisateurs peuvent choisir leurs trajets, vérifier la disponibilité des places et effectuer leurs réservations de manière simple et rapide.
 - ◆ **Paiements** : Cotisse-Transport.mg offre plusieurs options de paiement en ligne, incluant les cartes bancaires et les services de paiement mobile. Cette fonctionnalité garantit une transaction sécurisée et offre une commodité accrue pour les utilisateurs.
 - ◆ **Notification** : Les utilisateurs reçoivent des notifications et des rappels concernant leurs réservations. Ces notifications peuvent inclure des informations sur les horaires, des rappels de départ et des alertes en cas de modifications ou de retards.
 - ◆ **Information détaillées sur les trajets** : Le site fournit des informations complètes sur les trajets proposés, incluant les points de départ et d'arrivée, les arrêts intermédiaires, les horaires et les tarifs. Cela aide les voyageurs à planifier leurs trajets de manière efficace.
- [2.1]

Le site permet aux utilisateurs de réserver des billets en ligne à tout moment et depuis n'importe où, ce qui est particulièrement avantageux pour ceux vivant dans des zones éloignées. Cela leur fait gagner du temps en évitant les files d'attente et en optimisant leurs déplacements grâce au suivi en temps réel des véhicules. De plus, les options de paiement en ligne sont sécurisées et pratiques, offrant des reçus électroniques et réduisant la manipulation d'argent liquide. Les notifications et informations détaillées sur les trajets augmentent le confort des utilisateurs, leur permettant de gérer et modifier leurs réservations selon leurs besoins.

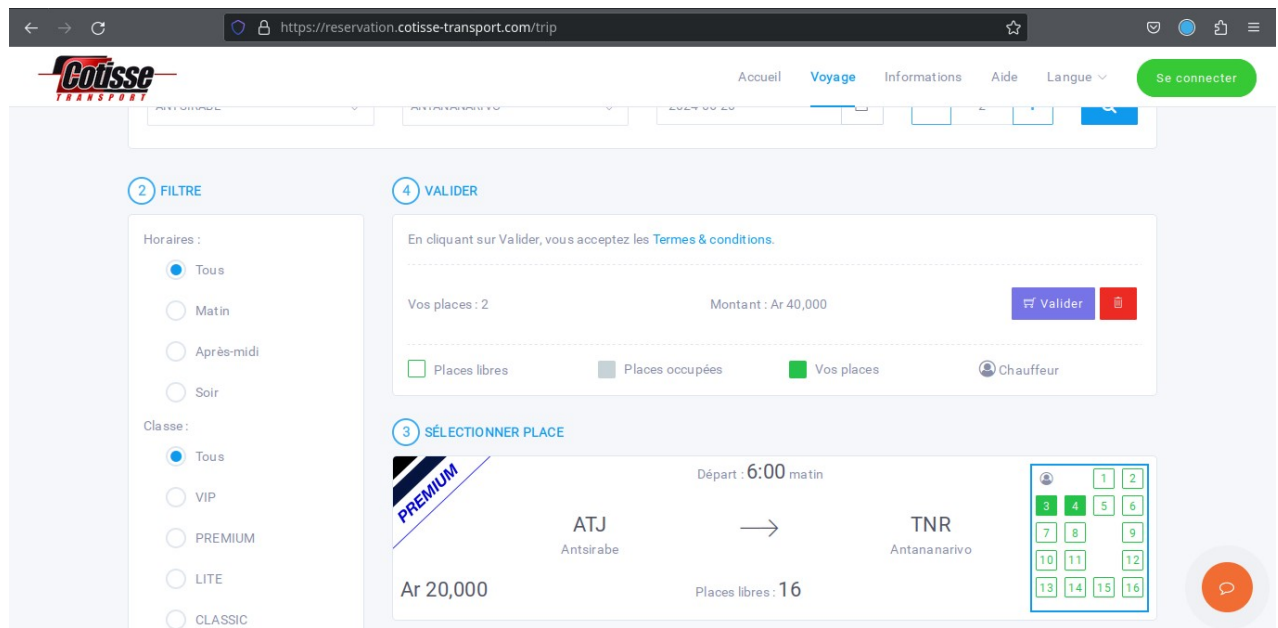


Figure 1.1: Le site web Cotisse-Transport

● Problèmes et limitations

- Dépendance à la connexion internet : Le site Cotisse-Transport.mg nécessite une connexion internet pour accéder à ses services. Dans les régions où l'accès à internet est limité ou instable, cela peut poser un problème pour les utilisateurs.
- Difficulté pour les utilisateurs peu familiers avec la technologie : Les utilisateurs qui ne sont pas à l'aise avec les technologies numériques peuvent trouver le processus de réservation en ligne difficile, ce qui pourrait les décourager d'utiliser ces services.

1.2.2 Le site web taxi-brousse

Taxi-Brousse.mg est un site web destiné à la gestion des réservations de taxi-brousse, un moyen de transport public très utilisé à Madagascar. Ce site vise à simplifier la réservation de billets, à fournir des informations utiles aux voyageurs et à améliorer l'expérience globale des utilisateurs.

- ◆ **Réservation en ligne** : La fonctionnalité principale de Taxi-Brousse.mg est la réservation en ligne de billets pour divers trajets en taxi-brousse. Les utilisateurs peuvent choisir leur destination, vérifier la disponibilité des places et effectuer leurs réservations en quelques étapes simples.
- ◆ **Informations sur les trajets** : Le site fournit des informations détaillées sur les trajets disponibles, y compris les points de départ et d'arrivée, les arrêts intermédiaires, les horaires de départ et d'arrivée, et les tarifs. Cela permet aux voyageurs de planifier leurs déplacements de manière efficace.
- ◆ **Options de paiement** : Taxi-Brousse.mg propose plusieurs options de paiement en ligne, telles que les cartes bancaires et les services de paiement mobile, pour faciliter les transactions et offrir une plus grande commodité aux utilisateurs.
- ◆ **Notifications et rappels** : Les utilisateurs reçoivent des notifications et des rappels concernant leurs réservations par SMS ou email. Cela inclut des rappels de départ, des informations sur les retards et des modifications d'itinéraires.
- ◆ **Support client** : Le site offre un support client accessible pour répondre aux questions des utilisateurs et résoudre les problèmes éventuels liés aux réservations. Cela peut se faire via un chat en ligne, une adresse email dédiée ou un numéro de téléphone.[2.2]

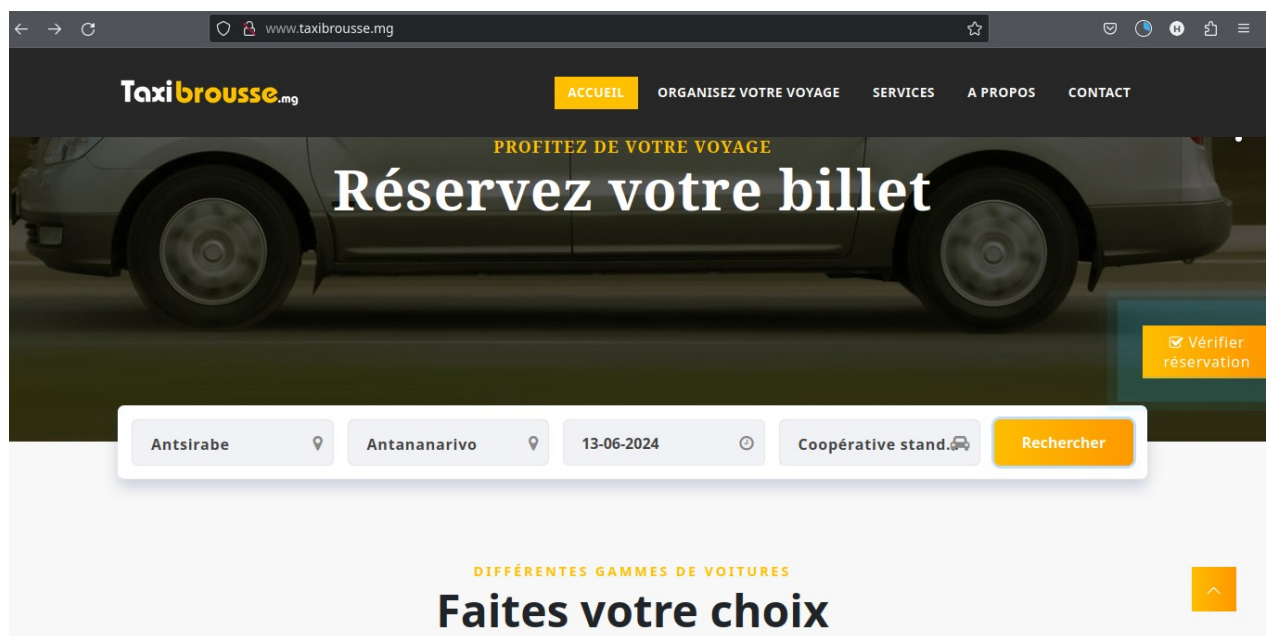


Figure 1.2: Le site taxi-brousse

En plus des avantages du site Cotisse-Transport, le site taxi-brousse offre une interface intuitive facilitant l'utilisation et permettant aux utilisateurs de réserver des billets rapidement. La sécurité des transactions en ligne est renforcée, et les utilisateurs reçoivent des reçus électroniques. Les informations détaillées sur les trajets et les horaires aident également les voyageurs à mieux organiser leurs voyages et à choisir les meilleurs itinéraires.

● **Problèmes et limitations**

-Dépendance à la connexion internet : un site web nécessite une connexion internet pour accéder à ses services. Dans les zones où l'accès à internet est limité ou instable, cela peut poser un problème pour les utilisateurs potentiels.

-Fiabilité des informations : La précision des informations sur les trajets, notamment les horaires de départ et d'arrivée, peut parfois être sujette à des changements imprévus. Cela peut causer des désagréments pour les utilisateurs si les informations ne sont pas mises à jour en temps réel.

1.2.3 SoatransPlus

SoatransPlus.mg est un site web dédié à la gestion des réservations de transport public à Madagascar. Cette plateforme vise à faciliter la réservation de places dans les véhicules de transport en commun, améliorer l'expérience utilisateur et offrir des services supplémentaires aux voyageurs.

- ◆ Gestion des paiements : Soatrans intègre moyens permettant aux utilisateurs de payer leurs réservations via des services de paiement mobile, ou d'autres méthodes locales.
- ◆ Mode de réservation : Les utilisateurs doivent appeler un numéro de téléphone pour effectuer une réservation. Cela implique de parler directement à un représentant pour confirmer les détails de la réservation. De plus les numéros de téléphone sont disponibles pour chaque destination. [2.3]

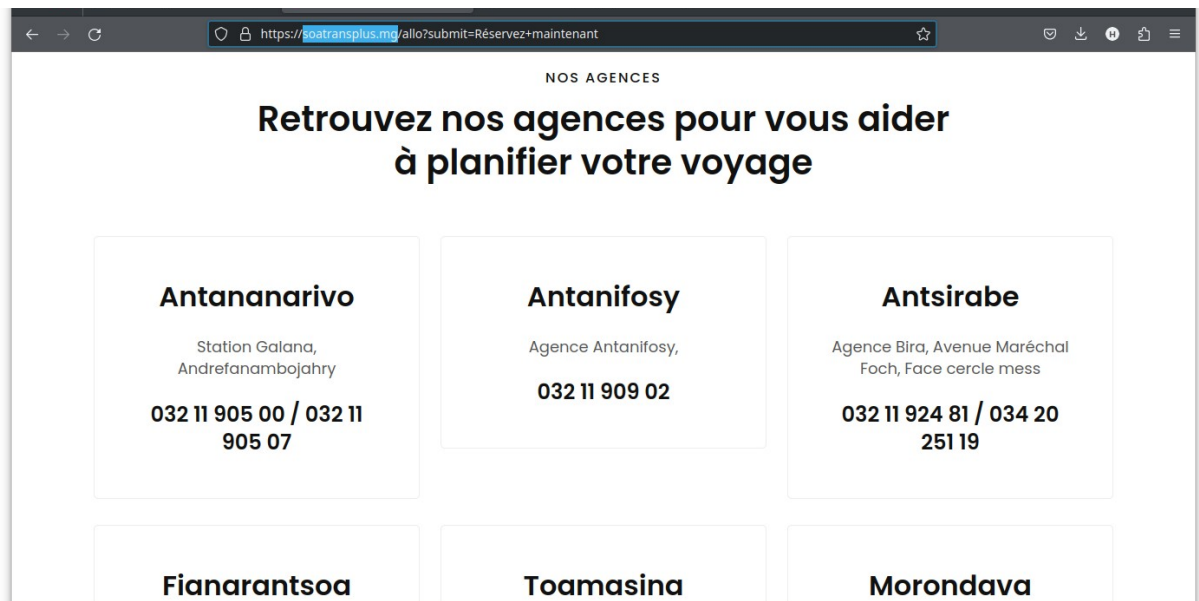


Figure 1.3 Site web SoatransPlus

En plus des avantages du site Cotisse-transport.mg, le site SoatransPlus.mg permet aux utilisateurs d'accéder facilement aux informations sur les trajets, les horaires et les tarifs via une interface simple. La réservation par téléphone offre une assistance personnalisée, permettant aux utilisateurs d'obtenir des réponses immédiates à leurs questions et de confirmer rapidement leurs réservations. Les services de transport spécifiques, comme les navettes et les transferts privés, ajoutent également une valeur supplémentaire pour les voyageurs, leur offrant des options de transport adaptées à leurs besoins.

● Problèmes et limitations

-Manque de réservation en ligne : L'absence d'un système de réservation en ligne limite l'accessibilité pour les utilisateurs qui préfèrent les méthodes numériques. Les coûts élevés des appels téléphoniques pour effectuer une réservation peuvent représenter une dépense supplémentaire pour les utilisateurs par rapport à une réservation en ligne via une connexion internet.

-Dépendance à la disponibilité des représentants : La nécessité de parler à un représentant pour effectuer une réservation peut entraîner des délais si les lignes sont occupées ou si les représentants ne sont pas disponibles.

-Limitation des options de paiement : Le site ne mentionne pas de méthodes de paiement en ligne, ce qui signifie que les utilisateurs doivent probablement payer en personne ou par mobile monnaie sans preuve pour le paiement, ce qui peut être moins pratique.

1.2.4 L'application mobile TaxiBe

TaxiBe est une application mobile multifonctionnelle dédiée à l'utilisation du réseau de transport en commun de la ville d'Antananarivo, ainsi qu'à la réservation de taxis-brousse pour les trajets interurbains à Madagascar. Cette étude se concentrera sur les fonctionnalités, les avantages et les potentielles limitations de TaxiBe en tant que plateforme de réservation de transport.

1.2.4.1 Fonctionnalités

→ Utilisation des bus de la ville d'Antananarivo

-Liste des lignes de bus : TaxiBe permet d'afficher la liste de toutes les lignes de bus de la capitale, regroupant plus de 180 lignes.

-Recherche d'itinéraires : Les utilisateurs peuvent rechercher un itinéraire en entrant un lieu de départ et un lieu d'arrivée. L'application fournit des indications sur le prix du ticket et une estimation du temps de trajet, prenant en compte les conditions de circulation.

-Affichage sur carte : L'itinéraire sélectionné peut être visualisé sur une carte en ligne, permettant aux utilisateurs de choisir la ligne de bus appropriée parmi plusieurs options.

→ Utilisation des taxis-brousse

-Recherche de trajets interurbains : TaxiBe permet de rechercher les meilleurs moyens de transport pour se rendre dans les provinces principales de Madagascar (Tamatave, Diego Suarez, Majunga, ...) en proposant des voyages de jour ou de nuit.

-Affichage sur carte : Le trajet sélectionné peut être visualisé sur une carte, partant de la capitale.

-Liste des coopératives : L'application affiche la liste des coopératives de taxis-brousse desservant ces provinces et permet de les contacter directement via smartphone ou SMS.

-Gare routière : Les utilisateurs peuvent voir la localisation de la gare routière de départ sur une carte.

-Fonctionnalité à venir : La fonctionnalité d'appel de taxis individuels (pikla) n'est pas encore disponible, mais elle sera intégrée dans une version future de l'application.

1.2.4.2 Avantages

TaxiBe offre plusieurs avantages aux utilisateurs, notamment l'accessibilité et la commodité d'utiliser une seule application pour plusieurs modes de transport. Les fonctionnalités comprennent une liste complète et mise à jour des lignes de bus, la recherche d'itinéraires avec estimation des coûts et du temps, et des informations détaillées pour les trajets en taxis-brousse. Ces caractéristiques permettent aux utilisateurs de planifier et de gérer leurs déplacements de manière efficace, que ce soit pour des trajets intra-urbains ou interurbains.

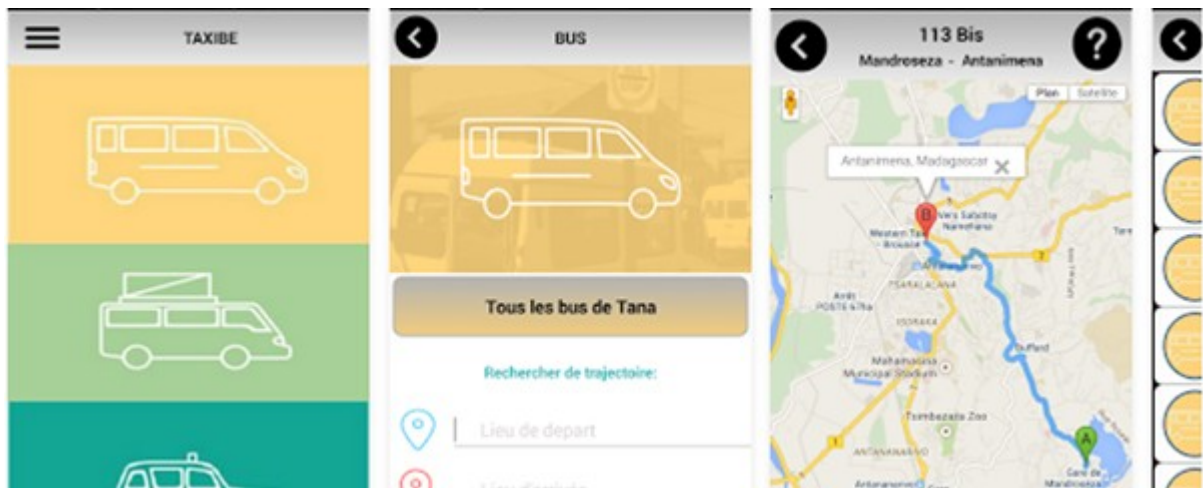


Figure 1.4 : Interface de l'application mobile TaxiBe

1.2.4.3 Problèmes et limitations

- Dépendance à une connexion internet : Comme pour toute application en ligne, TaxiBe nécessite une connexion internet stable pour accéder à ses services. Cela peut être un obstacle pour les utilisateurs dans les zones où la connectivité est limitée ou instable.
- Accessibilité pour certains utilisateurs : Les passagers qui ne sont pas familiers avec les technologies numériques peuvent rencontrer des difficultés lors de l'utilisation de l'application, préférant souvent les méthodes traditionnelles de recherche d'informations et de réservation.
- Fonctionnalité incomplète : La fonctionnalité d'appel de taxis individuels, bien que prévue, n'est pas encore disponible, limitant ainsi l'étendue des services offerts par l'application à ses utilisateurs actuels.
- Coût des appels et SMS : La nécessité de contacter les coopératives de taxis-brousse via appel ou SMS peut entraîner des coûts supplémentaires pour les utilisateurs, particulièrement ceux qui n'ont pas de forfaits d'appels illimités.

En résumé, TaxiBe propose une solution moderne et pratique pour l'utilisation du transport en commun et la réservation de taxis-brousse à Madagascar. Bien que l'application offre une gamme étendue de fonctionnalités, des défis subsistent, notamment en ce qui concerne la dépendance à une connexion internet stable et l'accessibilité pour certains utilisateurs. La future intégration de la fonctionnalité d'appel de taxis individuels augmentera encore l'utilité de l'application.

1.2.5 Conclusion des limites des études de l'existant

L'analyse des plateformes existantes, telles que SoatransPlus.mg, TaxiBe et d'autres, a révélé plusieurs limitations et problèmes communs qui affectent l'expérience utilisateur et l'efficacité opérationnelle des services de réservation de transport public à Madagascar.

- Absence de réservation en ligne : La plupart des plateformes, comme SoatransPlus.mg, ne disposent pas de systèmes de réservation en ligne, obligeant les utilisateurs à effectuer des réservations par téléphone. Cela peut entraîner des coûts supplémentaires en termes de frais d'appels et des délais dus à la disponibilité des représentants.
- Dépendance à la disponibilité des représentants : La nécessité de contacter un représentant pour effectuer une réservation peut être contraignante, surtout en cas de lignes occupées ou de problèmes de réseau.
- Informations dispersées et incomplètes : Certaines plateformes fournissent des informations limitées ou mal organisées sur les trajets, les horaires et les tarifs, rendant difficile la planification et la comparaison des options de voyage pour les utilisateurs.
- La consommation de forfait internet a un impact important : les utilisateurs qui n'ont pas accès à une connexion stable sont désavantagés, la plupart des informations et des fonctionnalités nécessitant une connexion en temps réel.

Ces limitations mettent en évidence la nécessité d'une solution intégrée et moderne qui répondrait aux besoins actuels des utilisateurs de transport public à Madagascar. Une application mobile bien conçue pourrait offrir une réservation en ligne, des notifications et alertes, des options de paiement en ligne sécurisées, et une accessibilité hors ligne pour une expérience utilisateur optimale.

1.3 Fonctionnalités de l'application

1.3.1 Introduction

Dans cette section, nous allons détailler les fonctionnalités de notre application de réservation de transport public, en nous inspirant des limites observées dans les applications existantes. Notre objectif est d'offrir une expérience utilisateur améliorée, en répondant aux besoins non satisfaits par les plateformes actuelles.

1.3.2 Fonctionnalités

1.3.2.1 Réservation en ligne

L'une des principales limitations observées sur les plateformes existantes, comme SoatransPlus.mg, est l'absence de réservation en ligne. Pour répondre à cette lacune, notre application propose une solution simple et intuitive permettant aux utilisateurs d'effectuer des réservations directement via l'application. Voici les fonctionnalités de réservation en ligne que nous offrons : simple et intuitive pour effectuer des réservations directement via l'application. Les utilisateurs pourront :

- **Rechercher des trajets** : Les utilisateurs pourront facilement accéder aux informations sur les trajets disponibles, incluant les lieux de départ et d'arrivée, les horaires, et les tarifs. L'interface de recherche intuitive permet de filtrer les trajets en fonction des préférences de l'utilisateur, facilitant ainsi la planification de leurs déplacements.
- **Réserver des sièges** : Les utilisateurs auront la possibilité de sélectionner et de réserver un siège en temps réel. L'application affichera les sièges disponibles, permettant aux utilisateurs de choisir leur place préférée. Cette fonctionnalité réduit le risque de sursur réservation et assure une répartition optimale des passagers.
- **Confirmation instantanée** : Après avoir effectué le paiement via des solutions de mobile monnaie, même en mode hors connexion, les utilisateurs recevront une confirmation de réservation immédiate. Cette confirmation sera envoyée sous forme de notification et sera également accessible dans l'application. Les utilisateurs pourront ainsi être assurés que leur réservation est bien prise en compte, sans avoir à attendre une confirmation par e-mail ou appel téléphonique.

1.3.2.2 Cartes hors ligne

Pour répondre aux besoins des utilisateurs qui n'ont pas toujours accès à une connexion internet stable, notre application offrira des fonctionnalités de cartes hors ligne. Cette option garantira une navigation fluide et un accès continu aux informations essentielles, même en l'absence de connexion internet. Les fonctionnalités proposées incluront :

- **Cartes** : Les cartes des trajets sont disponibles pour une utilisation sans connexion internet. Cela permettra aux voyageurs de visualiser les itinéraires détaillés et de se repérer facilement tout au long de leur voyage. Les cartes téléchargeables incluront toutes les informations nécessaires pour une navigation aisée, telles que les arrêts principaux, les intersections importantes, et les points de repère clés. De plus, les cartes de Madagascar sont d'ores et déjà incluses dans l'application, offrant ainsi une couverture complète pour les trajets à travers le pays.

- **Accéder aux informations critiques** : En plus des cartes, notre application permettra aux utilisateurs de consulter hors ligne les détails des trajets, tels que les horaires de départ et d'arrivée, les arrêts intermédiaires, et les informations sur les réservations. Cette fonctionnalité garantit que les voyageurs ont toujours accès aux informations essentielles, même lorsqu'ils se trouvent dans des zones à faible couverture réseau ou en itinérance internationale. Les utilisateurs pourront également voir les informations sur leurs réservations passées et futures, leur offrant ainsi une tranquillité d'esprit et une meilleure organisation de leurs voyages.

1.3.2.3 Notifications et alertes

Les notifications et les alertes jouent un rôle crucial dans l'amélioration de l'expérience utilisateur. En fournissant des informations en temps réel et des mises à jour pertinentes, notre application assure que les utilisateurs restent informés et peuvent planifier leurs déplacements en toute confiance. Les fonctionnalités de notifications et d'alertes incluront :

- **Notifications de réservation** : Dès qu'une réservation est confirmée ou modifiée, les utilisateurs recevront une notification instantanée. Cela inclut les détails de la réservation, tels que le numéro du siège, l'heure de départ, et les informations sur le trajet. En cas de modification de la réservation, les utilisateurs seront immédiatement informés des changements, ce qui leur permettra de s'adapter rapidement sans avoir à vérifier manuellement l'état de leur réservation.
- **Alertes de départ** : Avant l'heure de départ prévue, les utilisateurs recevront des alertes pour leur rappeler leur voyage imminent. Ces alertes incluront également des informations sur les retards éventuels, les changements d'itinéraire, ou toute autre mise à jour pertinente. Les utilisateurs pourront ainsi ajuster leur planning en fonction des conditions actuelles et éviter les surprises de dernière minute. De plus, des alertes de pré-départ seront envoyées suffisamment à l'avance pour permettre aux utilisateurs de se préparer et de se rendre à la station à temps.

1.3.2.4 Gestion des réservations

La gestion des réservations sera simplifiée et flexible pour les utilisateurs. Ils pourront :

- **Modifier les réservations** : Changer la date, l'heure ou le siège réservé en fonction des disponibilités, offrant ainsi une flexibilité nécessaire pour les imprévus.

- **Demande d'annulation des réservations** : Annuler une réservation facilement via l'application et recevoir un remboursement selon les politiques en vigueur, simplifiant le processus en cas de changement de plans.
- **Historique des réservations** : Accéder à un historique complet de leurs réservations passées et en cours, permettant aux utilisateurs de suivre et gérer efficacement leurs voyages.

1.3.2.5 Paiement en ligne sécurisé

Pour surmonter les limitations liées aux méthodes de paiement observées dans les applications existantes, notre application proposera un éventail complet de solutions de paiement en ligne. Les utilisateurs bénéficieront de la commodité et de la flexibilité d'effectuer leurs transactions directement depuis l'application, tout en ayant l'assurance que leurs informations financières sont protégées. Les fonctionnalités de paiement en ligne incluront :

- **Options de paiement variées** : Les utilisateurs pourront choisir parmi plusieurs méthodes de paiement populaires, telles qu'Orange Money, MVola et Airtel Money. Cette diversité permet de s'adapter aux besoins et aux habitudes de chaque utilisateur, offrant ainsi une expérience de paiement fluide et sans tracas. De plus, les utilisateurs pourront vérifier les messages de confirmation envoyés par Orange Money, MVola et Airtel Money pour s'assurer de la bonne réception de leur paiement.
- **Reçus électroniques** : Pour chaque transaction effectuée, les utilisateurs recevront automatiquement un reçu électronique détaillé. Ces reçus sont accessibles dans l'application. Les utilisateurs pourront ainsi garder une trace de leurs paiements, ce qui facilite la gestion de leurs dépenses et renforce la transparence des transactions. De plus, les reçus électroniques contribueront à réduire l'usage de papier, soutenant ainsi des pratiques écologiques.

1.4 Conclusion

L'analyse des plateformes de réservation de transport public existantes à Madagascar, telles que SoatransPlus.mg , a révélé plusieurs limitations majeures, notamment l'absence de réservation en ligne, le manque de suivi en temps réel des véhicules, la dépendance à la disponibilité des représentants pour effectuer des réservations, des options de paiement limitées et une accessibilité réduite sans connexion internet. Ces insuffisances mettent en lumière la nécessité d'une solution moderne et intégrée qui pourrait offrir une expérience utilisateur améliorée. Notre

application vise à combler ces lacunes en proposant des fonctionnalités essentielles telles que la réservation en ligne, le suivi GPS des véhicules en temps réel, le téléchargement de cartes hors ligne, des notifications et alertes, une gestion flexible des réservations et des options de paiement en ligne sécurisées. En répondant à ces besoins, notre application se positionne comme une alternative pratique, efficace améliorant considérablement la satisfaction des utilisateurs dans le domaine du transport public. Le second chapitre sera dédié à l'analyse et à la conception de notre application.

CHAPITRE 2 CONCEPTION DU PROJET

2.1 Introduction

Après avoir identifié les limitations des plateformes de réservation de transport public existantes dans le chapitre précédent, nous allons maintenant détailler la conception de notre projet d'application mobile en utilisant UML. Ce chapitre vise à présenter les aspects techniques et fonctionnels de l'application, en mettant en lumière les choix de conception qui répondront aux besoins non satisfaits identifiés. Nous aborderons les différentes phases de conception, incluant l'architecture globale de l'application, la modélisation des données, et l'élaboration des interfaces utilisateur. Chaque section sera accompagnée de diagrammes UML et d'exemples concrets pour illustrer les fonctionnalités clés et garantir une compréhension approfondie du projet. L'objectif est de créer une application innovante, intuitive et performante, capable de révolutionner l'expérience de réservation de transport public à Madagascar.

2.2 Méthode de Conception

Bien qu'il existe plusieurs méthodologies de conception, nous nous concentrerons sur l'utilisation de la notation UML.

2.2.1 Définition UML

L'UML (Unified Modeling Language, ou « langage de modélisation unifié » en français) est un langage de modélisation graphique utilisé principalement dans le génie logiciel et la conception orientée objet. Bien qu'il soit couramment employé dans les projets logiciels, il peut être appliqué à divers types de systèmes, ne se limitant pas au domaine informatique. UML n'est pas un langage de programmation, mais plutôt un langage universel de modélisation objet et un outil visuel de communication. [2.11]

2.2.2 Caractéristique UML

L'une des caractéristiques essentielles de l'UML est sa capacité à structurer l'analyse en permettant la représentation d'un système sous différentes perspectives complémentaires, à travers des diagrammes. Chaque diagramme UML, qui est une représentation graphique focalisée sur un aspect particulier du modèle, offre ainsi une vue spécifique. Ces diagrammes sont structurés selon des types d'éléments de modélisation prédéfinis et véhiculent une sémantique précise, assurant une cohérence dans la représentation du système. En combinant les différents types de diagrammes

UML, on obtient une vue d'ensemble des aspects statiques et dynamiques d'un système. Ils permettent donc d'explorer un modèle sous divers angles et d'orienter l'utilisation des éléments de modélisation grâce à leur structure bien définie.

2.2.3 Avantages

- Visualisation claire et compréhensible : UML permet de représenter visuellement la structure et le comportement d'un système, facilitant ainsi la compréhension pour toutes les parties prenantes, y compris les développeurs, les analystes et les clients.
- Communication améliorée : Les diagrammes UML servent de langage commun entre les membres de l'équipe de projet, réduisant ainsi les malentendus et améliorant la collaboration.
- Documentation standardisée : UML fournit une méthode standardisée pour documenter les systèmes logiciels, ce qui facilite la maintenance et les mises à jour ultérieures.
- Facilitation de l'analyse et de la conception : UML aide à structurer et à organiser les idées pendant la phase d'analyse et de conception, assurant ainsi que toutes les exigences sont prises en compte et correctement implémentées.
- Réutilisation des modèles : Les modèles UML peuvent être réutilisés dans différents projets, ce qui permet de gagner du temps et de réduire les coûts de développement.
- Prise en charge de la modélisation des aspects statiques et dynamiques : UML permet de modéliser à la fois la structure statique (par exemple, les classes et les objets) et le comportement dynamique (par exemple, les interactions et les états), offrant ainsi une vue complète du système.
- Flexibilité et adaptabilité : UML peut être utilisé pour modéliser divers types de systèmes, qu'ils soient logiciels ou non, et s'adapte à différents paradigmes de développement, y compris la méthode agile et le développement en cascade.
- Normalisation et compatibilité : En tant que norme largement acceptée, UML assure la compatibilité entre les outils de modélisation et les environnements de développement, facilitant ainsi l'intégration et l'interopérabilité.

2.3 Conception du projet

L'UML utilise divers types de diagrammes, mais nous allons mentionner les plus couramment utilisés sont diagramme de cas d'utilisation, diagramme de séquence, Diagramme d'activité et diagramme de classe

2.3.1 Diagramme de cas d'utilisation

2.3.1.1 Définition

Les cas d'utilisations permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Ils centrent l'expression des exigences du système sur ces utilisateurs : ils partent du principe que les objectifs du système sont tous motivés. La détermination et la compréhension des besoins sont souvent difficiles car les intervenants sont noyés sous de trop grandes quantités d'information : il faut clarifier et organiser les besoins des clients (les modéliser). Pour cela, les cas d'utilisation identifient les utilisateurs du système (acteurs) et leur interaction avec le système. Ils permettent de classer les acteurs structurer les objectifs du système.

Une fois identifiées et structurées, ces besoins :

- Définissent le contour du système à modéliser (ils précisent le but à atteindre),
- Permettent d'identifier les fonctionnalités principales (critiques) du système.

2.3.1.2 Éléments de modélisation de cas d'utilisation

- **Acteur:** Entité représentant les personnes qui servent réellement des cas d'utilisation. Un acteur est un type stéréotype représentant une abstraction qui réside juste en dehors du système à modéliser. Il représente un rôle joué par une personne ou une chose qui interagit avec le système.

Enfin, un acteur n'est pas nécessairement une personne physique : il peut être un service, une société ou un système informatique.

Il existe 4 catégories d'acteurs :

- Les acteurs principaux : les personnes qui utilisent les fonctions principales du système.
- Les acteurs secondaires : les personnes qui effectuent les tâches administratives ou de maintenance.
- Le matériel externe : les dispositifs matériels incontournables qui font partie du domaine de l'application et qui doivent être utilisés.
- Les autres systèmes : les systèmes avec lesquels le système doit interagir.

- **Cas d'utilisation :** Le cas d'utilisation (ou use case) représente une fonctionnalité du système (visible de l'extérieur du système). Le cas d'utilisation correspond à un objectif du système, motivé par un besoin d'un ou plusieurs acteurs. L'ensemble des use cases décrit les objectifs (le but) du système.

- **La relation :** Elle exprime l'interaction entre un acteur et un cas d'utilisation

Il existe 3 types de relations entre cas d'utilisation :

La relation de généralisation: Dans une relation de généralisation entre 2 cas d'utilisations, le cas d'utilisation enfant est une spécialisation du cas d'utilisation parent.

La relation d'inclusion: Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. L'inclusion a un caractère obligatoire, la source spécifiant à quel endroit le cas d'utilisation source doit être inclus. Cette relation permet ainsi de décomposer des comportements et de définir des comportements partageables entre plusieurs cas d'utilisation.

La relation d'extension: Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise avec une condition. Le comportement ajouté est ajouté au niveau d'un point d'extension défini dans le cas d'utilisation destination. Cette relation permet de modéliser les variantes de comportements d'un cas d'utilisation (selon les interactions des acteurs et l'environnement du système).

2.3.1.3 Diagramme de cas d'utilisation de notre application

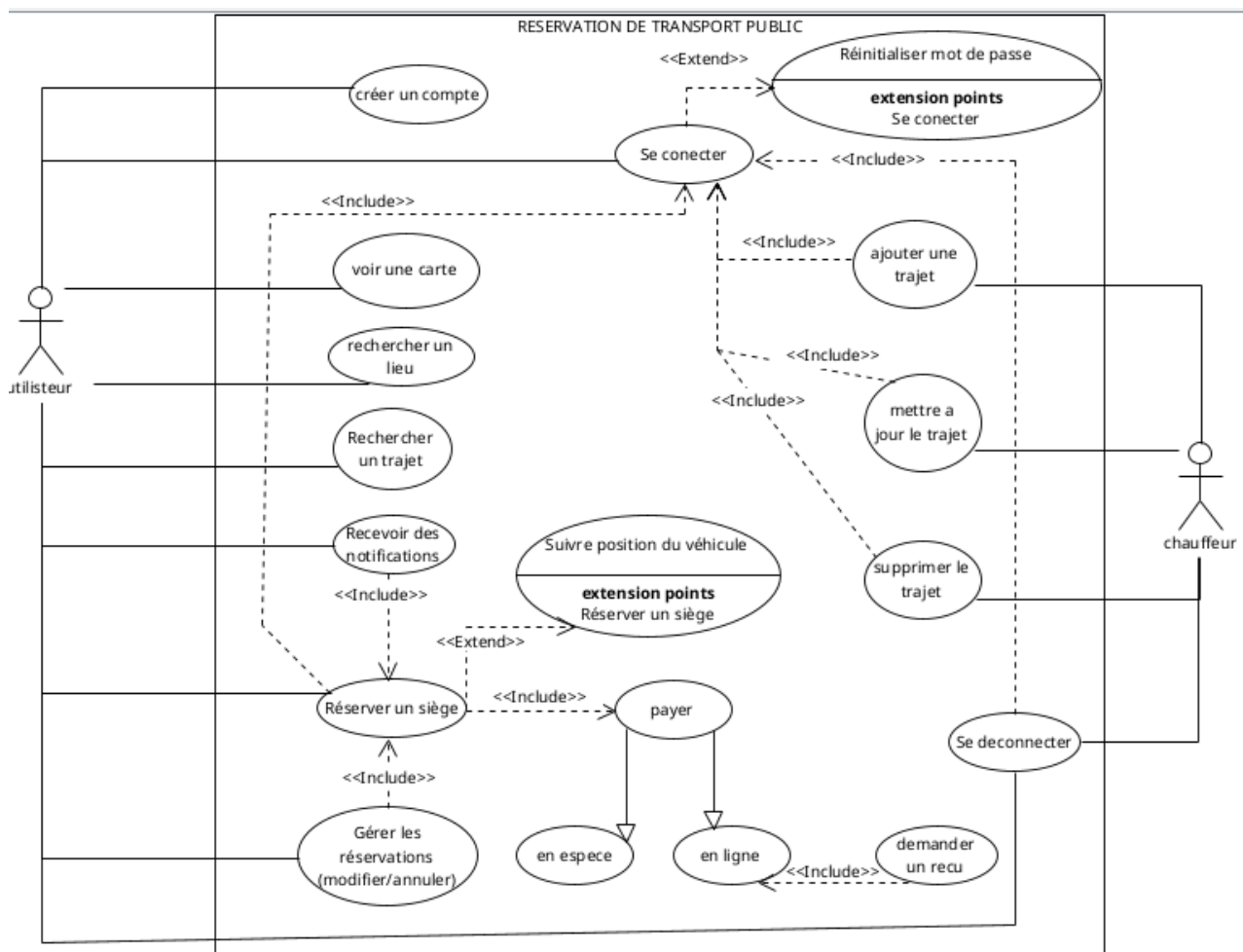


Figure 2.5: Diagramme de cas d'utilisation

2.3.2 Description textuel des cas d'utilisation

2.3.2.1 Définition

-Description textuelle: est une explication détaillée et structurée d'un scénario spécifique qui décrit comment un utilisateur ou un système interagit avec un système pour accomplir un objectif particulier. Cette description doit inclure les acteurs impliqués dans le scénario, les étapes clés de l'interaction utilisateur-système, ainsi que les résultats attendus. Elle doit également fournir des informations sur les préconditions, les exceptions et les variantes possibles du scénario principal. En général, une description textuelle d'un cas d'utilisation vise à fournir une compréhension claire et complète du processus que l'utilisateur ou le système doit suivre pour atteindre son objectif.

En effet, ci-dessous, les différentes sortes de scénario :

-Scénario nominal: également appelé le scénario principal ou le scénario de base, est la séquence d'étapes standard et attendue d'un cas d'utilisation. Il décrit la manière dont le système ou l'application doit réagir dans des conditions normales et attendues.

-Scénario alternatif: c'est un type de scénario de cas d'utilisation qui décrit une séquence d'étapes qui diffère du scénario nominal. Il peut être déclenché par une situation exceptionnelle ou inattendue qui modifie le comportement du système ou de l'utilisateur. Les scénarios alternatifs sont utilisés pour décrire des situations qui ne sont pas couvertes par le scénario nominal. Ils décrivent des comportements de système ou d'utilisateur qui sont différents de ce qui est attendu dans le scénario nominal.

-Scénario d'exception: est un type de scénario de cas d'utilisation qui décrit une situation anormale ou exceptionnelle qui interrompt ou modifie la séquence d'étapes du scénario nominal. Il décrit ce qui se produit lorsque le système ou l'utilisateur rencontre une erreur ou une condition inhabituelle qui n'a pas été prise en compte dans le scénario nominal ou les scénarios alternatifs.

-Pré condition: Réponse de la question « Quand et comment le cas d'utilisation débute ? ».

2.3.2.2 Description textuelle des cas d'utilisation

- **Cas d'utilisation n°1: « Ajouter une trajet »**

Généralité

Cas d'utilisation : Ajouter un trajet

Acteur : Chauffeur

Près condition : Le chauffeur doit être authentifié

Scénario nominal

Figure 2.6: Scénario nominal du cas d'utilisation n°1 : « Ajouter une trajet »

N° d'enchaînement	Action
1	Le chauffeur s'authentifie dans le système.
2	Le chauffeur sélectionne l'option « trajet » dans l'application.
3	Le chauffeur click sur le bouton « + » dans l'application.
4	Le système affiche un formulaire de création de trajet.
5	Le chauffeur remplit les informations requises pour le trajet (lieu de départ, lieu d'arrivée, horaire, voiture).
6	Le chauffeur soumet le formulaire.
7	Le système valide les informations fournies.
8	Le système enregistre le nouveau trajet dans la base de données.
9	Le système confirme au chauffeur que le trajet a été ajouté avec succès.

Scénario alternatif

Table 2.1: Scénario alternatif du cas d'utilisation n°1: « Ajouter une trajet »

N° d'enchaînement	Action
6	Si les informations ne sont pas validées :
6.1	Le système affiche un message d'erreur et demande de corriger les informations.
7	Si le système rencontre une erreur lors de l'enregistrement du trajet :
7.1	Le système affiche un message d'erreur et invite le chauffeur à réessayer.

- **Cas d'utilisation n°2: « Réserver une siège »**

Généralité

Cas d'utilisation : Réserver un siège

Acteur : Utilisateur

Précondition : L'utilisateur doit être authentifié dans le système

Scénario nominal

Table 2.2: Scénario nominal du cas d'utilisation n°2: « Réserver une siège »

N° d'enchaînement	Action
1	L'utilisateur s'authentifie dans le système.
2	L'utilisateur recherche un trajet en utilisant les critères de lieu de départ, lieu d'arrivée et date.
4	L'utilisateur sélectionne un trajet parmi la liste des trajets disponibles
5	Le système affiche les détails du trajet sélectionné, y compris les sièges disponibles.
6	L'utilisateur choisit un siège et clique sur « Réserver un siège ».
7	Le système demande à l'utilisateur de confirmer la réservation et de choisir un mode de paiement.
8	L'utilisateur confirme la réservation et effectue le paiement.
9	Le système enregistre la réservation et met à jour le nombre de sièges disponibles.
10	Le système envoie une notification à l'utilisateur et au chauffeur une confirmant la réservation du siège.

Scénario alternatif

Table 2.3: Scénario alternatif du cas d'utilisation n°2: « Réserver une siège »

N° d'enchaînement	Action
4	Si aucun trajet correspondant n'est trouvé :
4.1	Le système affiche un message indiquant qu'aucun trajet ne correspond aux critères de recherche.

N° d'enchaînement	Action
8	Si le paiement échoue :
8.1	Le système affiche un message d'erreur et demande à l'utilisateur de réessayer
9	Si le système rencontre une erreur lors de l'enregistrement de la réservation :
9.1	Le système affiche un message d'erreur et invite l'utilisateur à réessayer la réservation

- **Cas d'utilisation N°3 : « Mettre à jour un trajet »**

Généralité

Cas d'utilisation : Mettre à jour un trajet

Acteur : Chauffeur

Précondition : Le chauffeur doit être authentifié dans le système et avoir les droits nécessaires pour modifier les trajets.

Scénario nominal

Table 2.4: Scénario nominal du cas d'utilisation N°3 : « Mettre à jour un trajet »

N° d'enchaînement	Action
1	Le chauffeur s'authentifie dans le système.
2	Le chauffeur sélectionne l'option « Mettre à jour un trajet » dans l'application.
3	Le système affiche la liste des trajets existants.
4	Le chauffeur sélectionne le trajet à mettre à jour.
5	Le système affiche les informations actuelles du trajet.
6	Le chauffeur modifie les informations nécessaires (lieu de départ, lieu d'arrivée, horaire, capacité, etc.).
7	Le chauffeur soumet les modifications.
8	Le système valide les nouvelles informations.

N° d'enchaînement	Action
9	Le système enregistre les modifications du trajet dans la base de données.
10	Le système confirme au chauffeur que le trajet a été mis à jour avec succès.

Scénario alternatif

Table 2.5: Scénario alternatif du cas d'utilisation N°3 : « Mettre à jour un trajet »

N° d'enchaînement	Action
6	Si les informations sont incomplètes
6.1	Le système affiche un message d'erreur et demande de compléter les champs manquants.
8	Si les informations sont invalides (ex. : horaire incorrect, capacité non valide) :
8.1	Le système affiche un message d'erreur et demande de corriger les informations.
9	Si le système rencontre une erreur lors de l'enregistrement des modifications :
9.1	Le système affiche un message d'erreur et invite le chauffeur à réessayer

- **Cas d'utilisation N°4 : « Rechercher un trajet »**

Généralité

Cas d'utilisation : Rechercher un trajet

Acteur : Utilisateur

Précondition : L'utilisateur doit être authentifié dans le système pour accéder à la fonctionnalité de recherche de trajets.

Scénario nominal

Table 2.6: Scénario nominal du cas d'utilisation N°4 : « Rechercher un trajet »

N° d'enchaînement	Action
1	L'utilisateur s'authentifie dans le système.
2	L'utilisateur sélectionne l'option « Rechercher un trajet » dans l'application.
3	Le système affiche un formulaire de recherche avec des critères tels que lieu

N° d'enchaînement	Action
	de départ, lieu d'arrivée, date et heure.
4	L'utilisateur remplit les critères de recherche et soumet le formulaire.
5	Le système traite la demande et affiche la liste des trajets correspondants.
6	L'utilisateur parcourt les résultats de la recherche.
7	L'utilisateur sélectionne un trajet pour afficher les détails.
8	Le système affiche les détails du trajet sélectionné.

Scénario alternatif

Table 2.7: Scénario alternatif du cas d'utilisation N°4 : « Rechercher un trajet »

N° d'enchaînement	Action
5	Si aucun trajet correspondant n'est trouvé :
5.1	Le système affiche un message indiquant qu'aucun trajet ne correspond aux critères de recherche.

Cas d'utilisation N°5 : « Gérer les réservations »

Généralité

Cas d'utilisation : Gérer les réservations (modifier/annuler)

Acteur : Utilisateur

Précondition : L'utilisateur doit être authentifié dans le système et avoir des réservations existantes pour pouvoir les gérer.

Scénario nominal

Table 2.8: Scénario nominal du cas d'utilisation N°5 : « Gérer les réservations »

N° d'enchaînement	Action
1	L'utilisateur s'authentifie dans le système.

N° d'enchaînement	Action
2	L'utilisateur sélectionne l'option « Gérer les réservations » dans l'application.
3	Le système affiche la liste des réservations de l'utilisateur.
4	L'utilisateur sélectionne une réservation à modifier ou à annuler.
5	Le système affiche les détails de la réservation sélectionnée.
6	L'utilisateur choisit de modifier ou d'annuler la réservation.
7	Pour une modification, l'utilisateur met à jour les informations nécessaires et soumet les changements.
8	Pour une annulation, l'utilisateur confirme l'annulation de la réservation.
9	Le système enregistre les modifications ou l'annulation dans la base de données.
10	Le système confirme à l'utilisateur que la réservation a été modifiée ou annulée avec succès.

Scénario alternatif

Table 2.9: Scénario alternatif du cas d'utilisation N°5 : « Gérer les réservations »

N° d'enchaînement	Action
7	Si les informations de modification sont incomplètes :
7.1	Le système affiche un message d'erreur et demande de compléter les champs manquants.
9	Si le système rencontre une erreur lors de l'enregistrement des modifications ou de l'annulation :
9.1	Le système affiche un message d'erreur et invite l'utilisateur à réessayer.

2.3.3 Diagramme de séquence

2.3.3.1 Définition

Le diagramme de séquence représente les interactions entre objets selon un point de vue temporel, on y met l'action sur chronologie des envois des messages. Il montre les réactions du système aux actions des utilisateurs, la création et la manipulation des objets.

2.3.3.2 Éléments du diagramme de séquence

- **Ligne de vie** : Représente l'ensemble des opérations exécutées par un objet. Il symbolise une durée qui dépend du scénario et du comportement modélisé. Représentée par une ligne verticale pointillée dirigée vers le bas à partir de chaque objet.

- **Les interactions**: L'interaction se traduit par l'envoi d'un message entre objets. Le diagramme de séquence insiste sur la chronologie des objets en utilisant la ligne de vie des objets.

- **Les activations**: Les diagrammes de séquence permettent de représenter les périodes d'activité des objets. Une période d'activité correspond un temps pendant lequel un objet effectue une action, soit directement, soit par l'intermédiaire d'un autre objet qui lui sert sous-traitant.

- **Le message**: Généralement un appel ou un signal ou une réponse.

Représenté par des flèches horizontales reliant la ligne de vie de l'objet émetteur à la ligne de vie de l'objet récepteur.

Les types de message sont:

- Message synchrone: l'objet émetteur reste bloqué lorsque l'objet récepteur traite le message envoyé.
- Message de création: Un message spécifique qui donne lieu au début de la ligne de vie d'un nouvel objet
- Message de destruction: Un message envoyé à un objet existant et qui donne lieu à la fin de sa ligne de vie
- Message réflexif: Un objet peut envoyer un message à lui-même
- Message récursif: Elle se présente par un dédoublement de la barre d'activation

-**Fragment combiné**: Regroupement logique représenté par un rectangle et contenant les structures conditionnelles qui affecte le flux de message. Il décrit les diagrammes de séquence de manière compacte. Il est défini par un opérateur et des opérandes. Les opérateurs conditionnent la signification du fragment combiné.

2.3.3.3 Diagramme de séquences du projet

2.3.3.3.1 Cas d'utilisation n°1: « Ajouter une trajet »

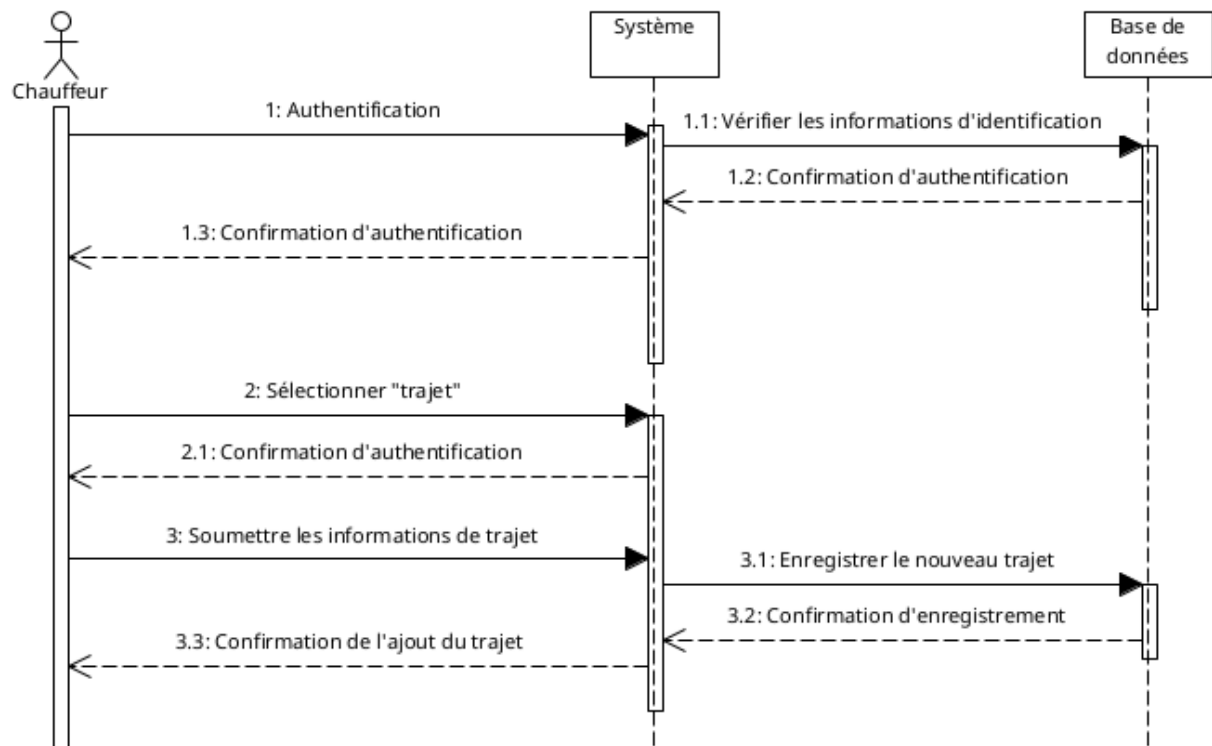


Figure 2.7 : Diagramme de séquences du cas d'utilisation n°1

2.3.3.3.2 Cas d'utilisation n°2: « Réserver une siège »

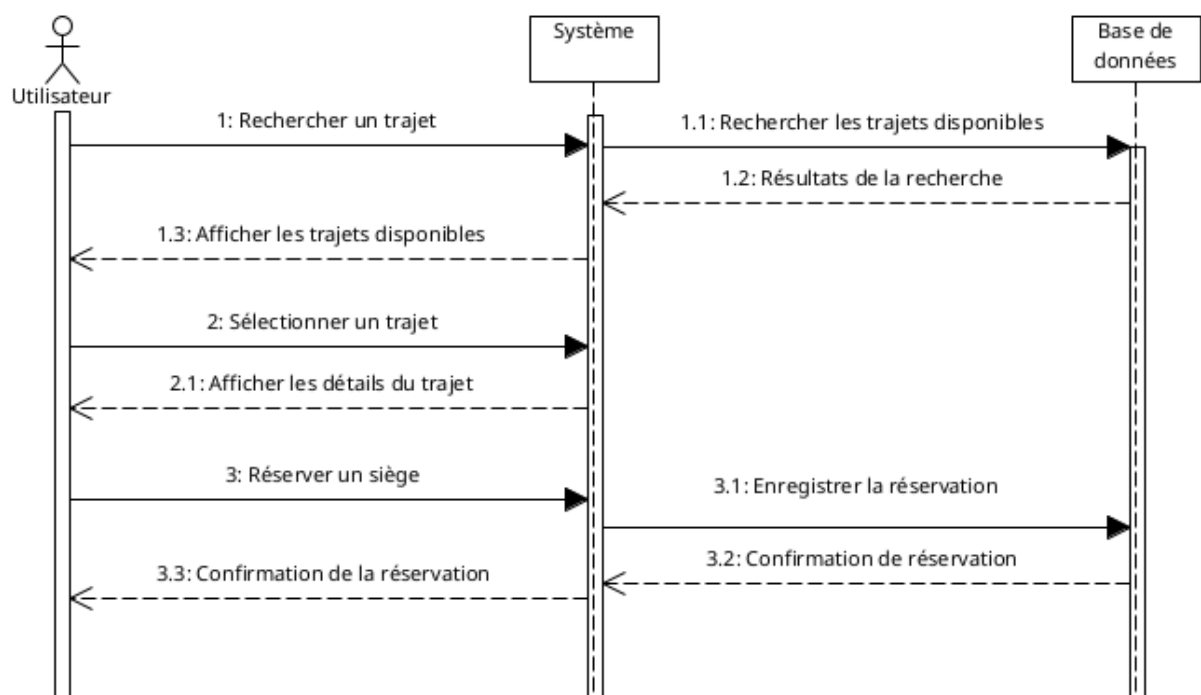


Figure 2.8 : Diagramme de séquences du cas d'utilisation n°2

2.3.3.3.3 Cas d'utilisation N°3 : « Mettre à jour un trajet »

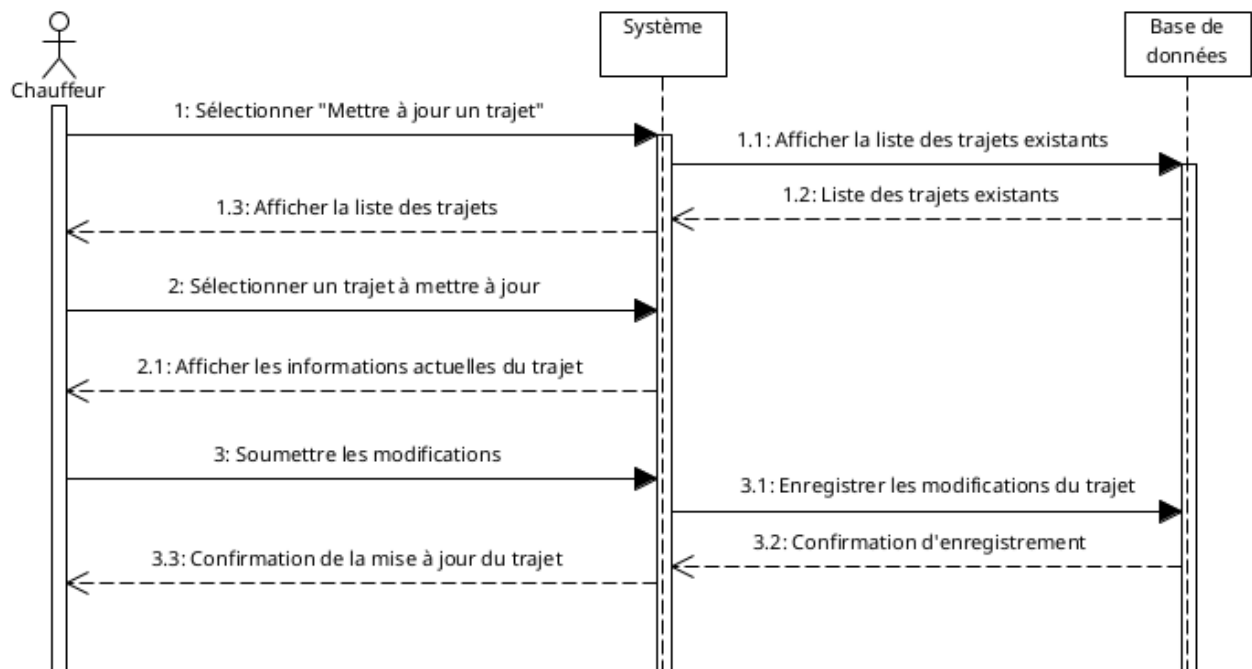


Figure 2.9 : Diagramme de séquences du cas d'utilisation n°3

2.3.3.3.4 Cas d'utilisation N°4 : « Rechercher un trajet »

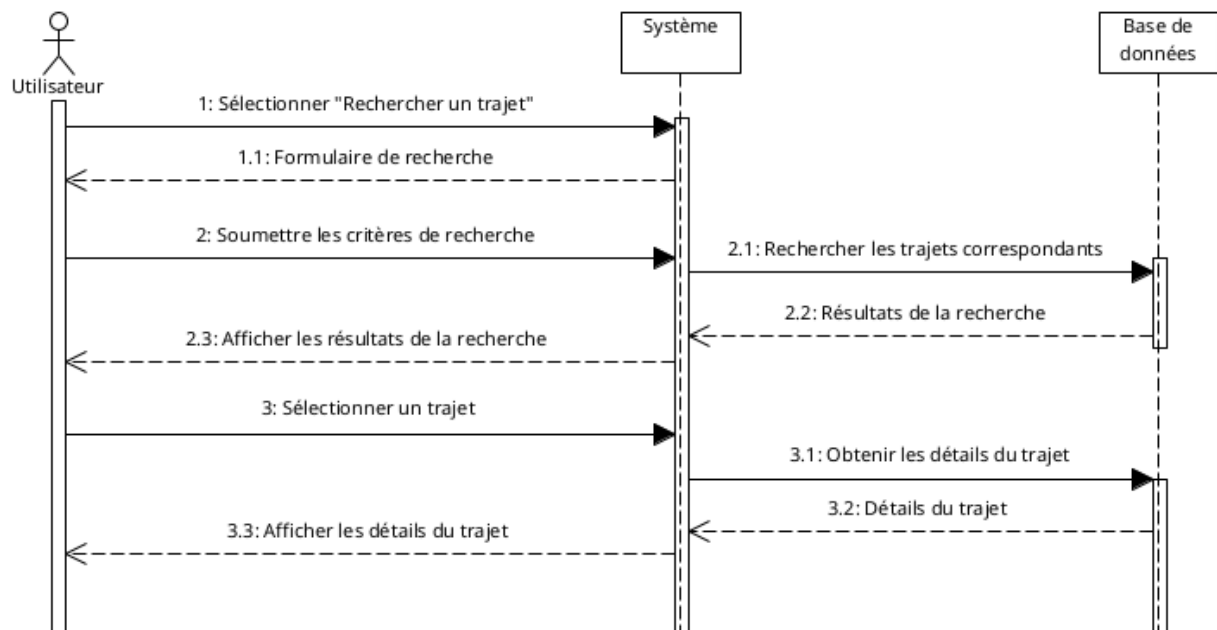


Figure 2.10 : Diagramme de séquences du cas d'utilisation n°4

2.3.3.3.5 Cas d'utilisation N°5 : « Gérer les réservations »

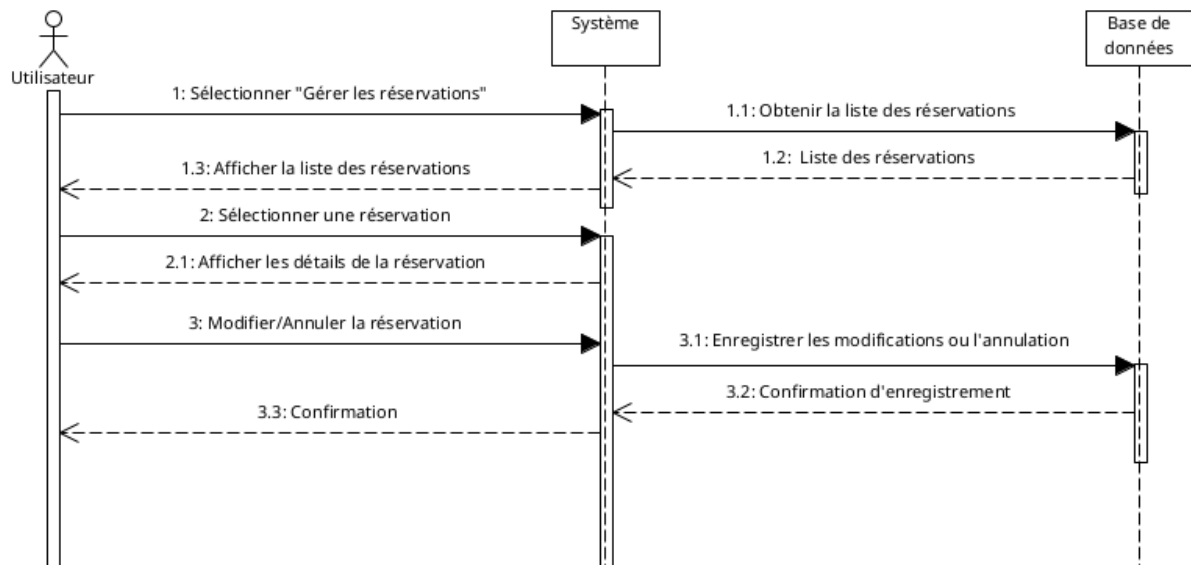


Figure 2.11 : Diagramme de séquences du cas d'utilisation n°5

2.3.4 Diagramme d'activité

2.3.4.1 Définition

Le diagramme d'activité fait partie des diagrammes comportementaux. Il est utilisé pour modéliser les aspects dynamiques d'un système. Il s'agit de représenter les opérations d'un processus et leurs conséquences sur les objets (logiciel ou matériel). La modélisation peut être utilisée pour décrire le déroulement d'un cas d'utilisation ou d'une méthode.

Les diagrammes d'activité affichent le flux de travail d'un point de départ à un point d'arrivée en détaillant les nombreux chemins de décision existant dans la progression des événements contenus dans l'activité.

2.3.4.2 Éléments du diagramme d'activité

- **Les activités:** Une activité définit un comportement décrit par un séquençement organisé d'unité dont les éléments simples sont les actions.

-**Les transitions :** Le passage d'une activité vers une autre est matérialisé par une transition.

Graphiquement les transitions sont représentées par des flèches en traits pleins qui connectent les activités entre elles.

- **Les nœuds de contrôle:** Un nœud de contrôle est un nœud d'activité abstrait utilisé pour coordonner les flots entre les nœuds d'une activité. Il existe 7 types de nœuds de contrôle:

- **Nœud initial:** Un nœud initial est un nœud de contrôle à partir duquel le flot débute lorsque l'activité lorsque l'activité enveloppante est invoquée. Graphiquement, un nœud initial est représenté par un petit cercle plein.

- Nœud de fin d'activité : L'un des arcs d'un nœud de fin d'activité est activé, l'exécution de l'activité enveloppante s'achève et tout nœud ou flot actif au sein de l'activité enveloppante est abandonné. Graphiquement, un nœud de fin d'activité est représenté par un cercle vide contenant un petit cercle plein.
- Nœud de bifurcation (fourche) : également appelé nœud de débranchement ; c'est un nœud de contrôle qui sépare un flot en plusieurs flots concurrents. Graphiquement, on représente un nœud de bifurcation par un trait plein.
- Nœud de décision : permet de faire un choix entre plusieurs flots sortants en fonction des conditions de gardes de chaque flot. N'a qu'un seul flot en entrée. On peut aussi utiliser deux flots de sorties : le premier correspond à la condition vérifiée et l'autre traite le cas sinon.
- Nœud de fin de flot : Lorsque l'un des arcs d'un nœud de fin de flot est activé, le flot spécifique se termine, mais l'activité enveloppante peut continuer. Graphiquement, un nœud de fin de flot est représenté par un cercle vide contenant un "X".
- Nœud de fusion : Un nœud de fusion combine plusieurs flots alternatifs en un seul flot sans évaluer de condition. Graphiquement, un nœud de fusion est représenté par un losange.
- Nœud de jonction : Un nœud de jonction combine plusieurs flots parallèles en un seul flot, synchronisant ainsi les activités parallèles. Graphiquement, un nœud de jonction est représenté par un trait plein avec plusieurs flèches entrantes et une seule flèche sortante.

2.3.4.3 Diagramme d'activité des cas d'utilisation

2.3.4.3.1 Cas d'utilisation n°1: « Ajouter une trajet »

La figure ci-après représente le diagramme de séquence de cas d'utilisation « Ajouter une trajet ».

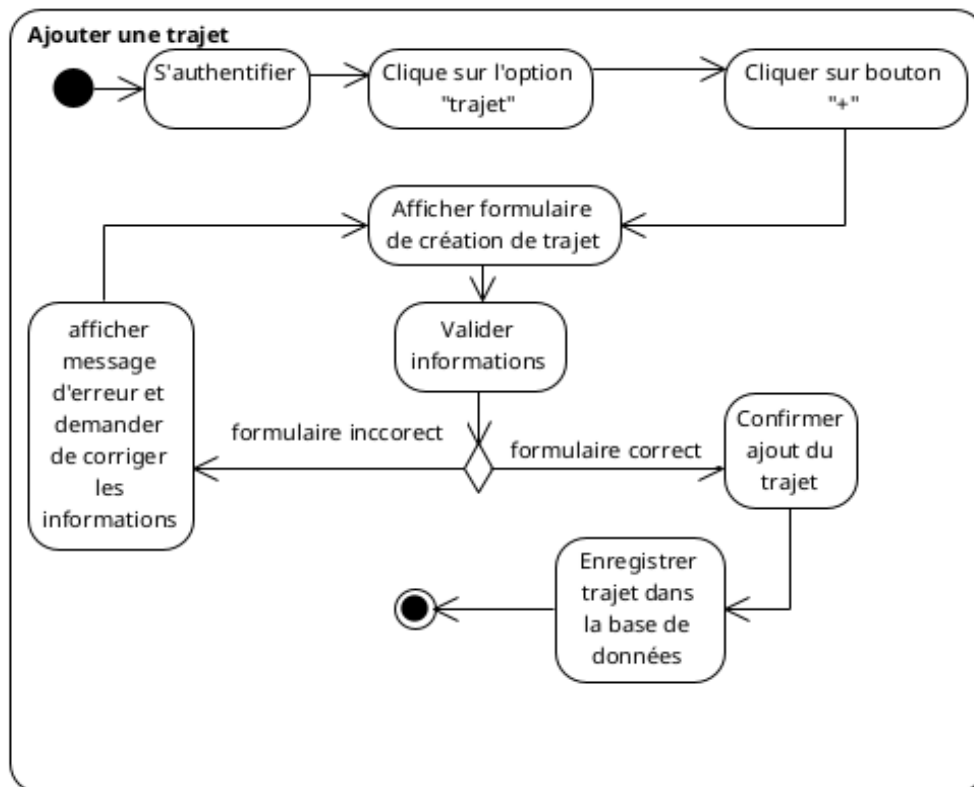


Figure 2.12: Diagramme d'activité du cas d'utilisation n°1

2.3.4.3.2 Cas d'utilisation n°2: « Réserver une siège »

La figure ci-après représente le diagramme de séquence de cas d'utilisation « Réserver une siège».

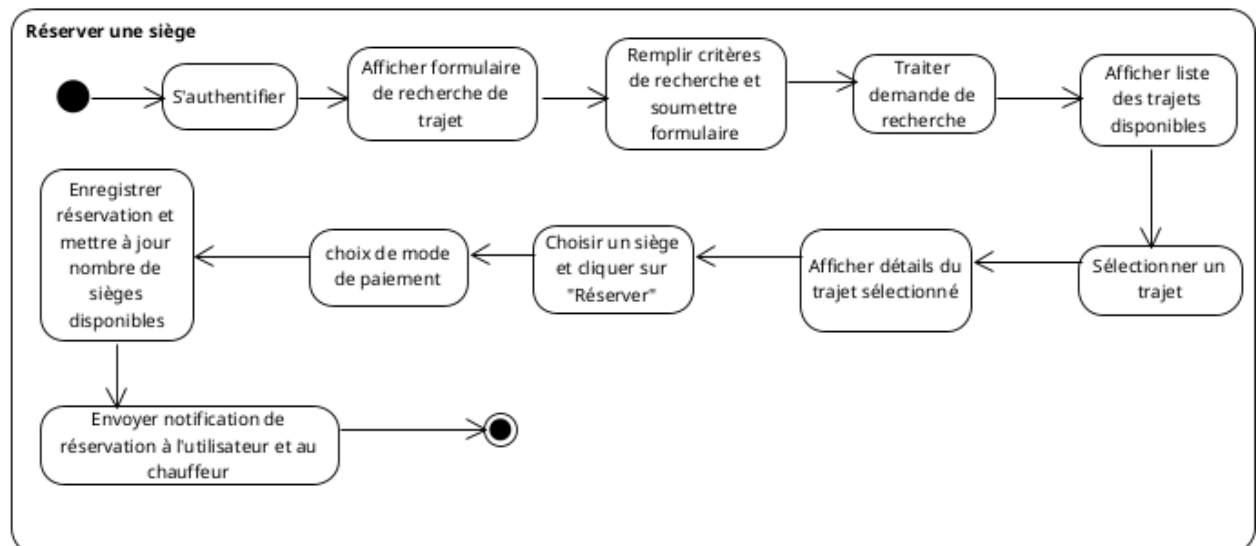


Figure 2.13: Diagramme d'activité du cas d'utilisation n°2

2.3.4.3.3 Cas d'utilisation N°3 : « Mettre à jour un trajet »

La figure ci-après représente le diagramme de séquence de cas d'utilisation « Mettre à jour un trajet ».

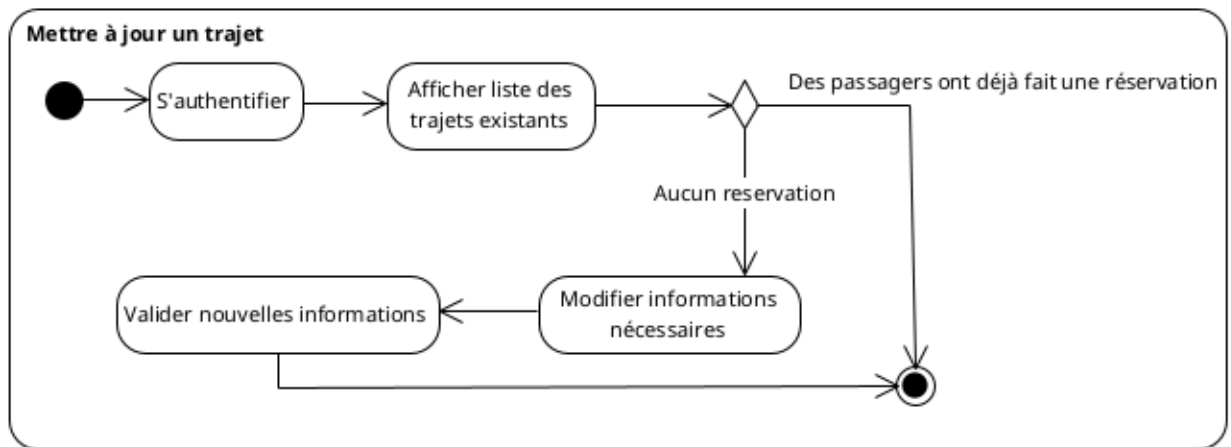


Figure 2.14: Diagramme d'activité du cas d'utilisation N°3

2.3.4.3.4 Cas d'utilisation N°4 : « Rechercher un trajet »

La figure ci-après représente le diagramme de séquence de cas d'utilisation « Rechercher un trajet ».

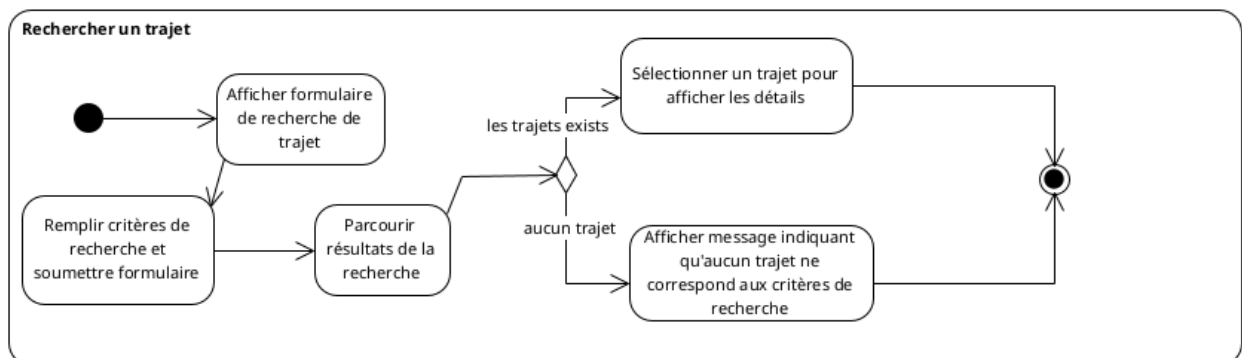


Figure 2.15: Diagramme d'activité du cas d'utilisation N°4

2.3.4.3.5 Cas d'utilisation N°5 : « Gérer les réservations »

La figure ci-après représente le diagramme de séquence de cas d'utilisation « Gérer les réservations ».

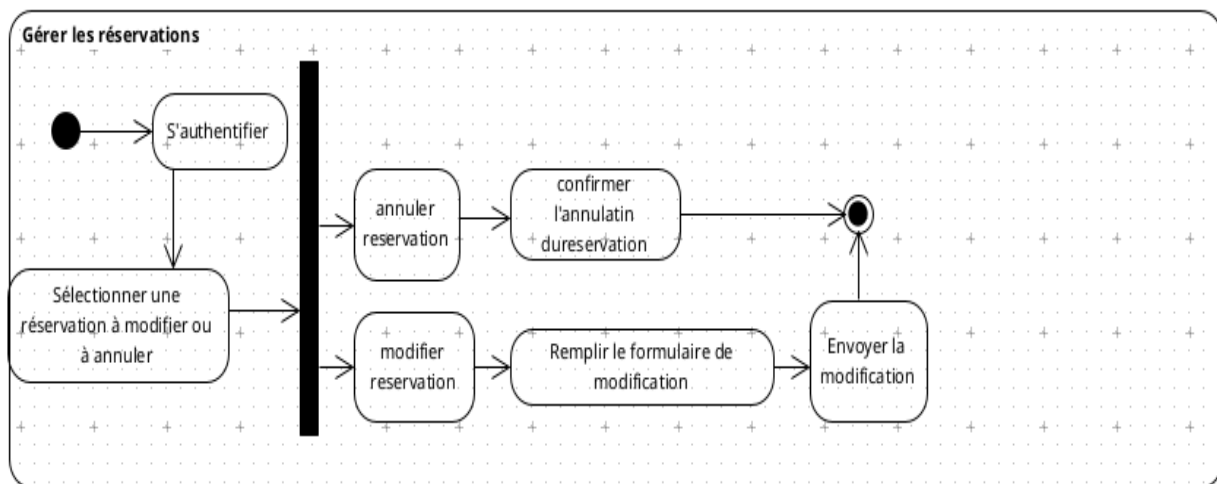


Figure 2.16: Diagramme d'activité du cas d'utilisation N°5 : « Gérer les réservations »

2.3.5 Dictionnaire de données

Un dictionnaire de données est la représentation de l'ensemble des données manipulées par le système. Pour une base de données c'est une collection de données qui sert à la conception d'une base de données relationnelle.

Suite à l'étude des besoins des voyageurs, nous avons un dictionnaire de données représenté dans le tableau :

Classe	Attribut	Code	Type
Utilisateur	Identifiant utilisateur	id_user	Int(11)
	Numéro de téléphone	numero	varchar(10)
	Photo	photo	varchar(100)
	CIN	cin	varchar(20)
	Nom	nom	varchar(100)
	Prénom	prenom	varchar(100)
	Mot de passe	motDePasse	varchar(100)
	Conducteur	est_conducteur	boolean
Notification	Identifiant notification	id_notification	Int(11)
	Utilisateur associé	id_user	int(11)
	Message	message	Text
	Date d'envoi	dateEnvoi	datetime
Trajet	Identifiant trajet	id_trajet	Int(11)

Classe	Attribut	Code	Type
	Lieu de départ	lieuDepart	varchar(100)
	Lieu d'arrivée	lieuArrive	varchar(100)
	Horaire	horaire	datetime
	Prix	prix	int(20)
	Véhicule associé	id_vehicule	int(11)
	Conducteur associé	id_user	int(11)
Vehicule	Identifiant véhicule	id_vehicule	Int(11)
	Propriétaire	id_user	int(11)
	Capacité	capacite	varchar(100)
	Numéro d'immatriculation	numeroVehicule	varchar(100)
	Photo	photo	int(11)
	Nombre de colonnes	nb_colonne	int(11)
	Nombre de rangées	nb_rangee	int(11)
Reservation	Identifiant réservation	id_reservation	int(11)
	Utilisateur associé	id_user	int(11)
	Trajet associé	id_trajet	int(11)
	Numéros de sièges	siegeNumero	Array
Paielement	Identifiant paiement	id_paiement	int(11)
	Référence	ref	varchar(10)
	Preuve	preuve	varchar(100)
	Réservation associée	id_reservation	int(11)
	Montant	montant	int(20)

2.3.6 Règles de gestion

Un règle de gestion est une règle écrite spécifiant ce qu'un système d'information doit faire, ou comment il doit être structuré. Les règles de gestion expriment les règles auxquelles obéit le système à modéliser.

RG1 : Un utilisateur peut recevoir plusieurs notifications.

RG2 : Une notification est liée à un utilisateur unique.

RG3 :Un utilisateur (conducteur) peut posséder plusieurs véhicules.

RG4 :Un véhicule est associé à un utilisateur (conducteur).

RG5 :Un utilisateur peut faire plusieurs réservations.

RG6 :Un utilisateur (conducteur) peut créer ou gérer plusieurs trajets

RG7 :Un véhicule peut lié à plusieurs trajets.

RG8 :Un trajet est associé à un véhicule.

RG9 :Un trajet est associé utilisateur (conducteur).

RG10 :Un trajet peut avoir plusieurs réservations.

RG11 :Une réservation est liée à un utilisateur.

RG12 :Un paiement est associé à une réservation unique.

2.3.7 Diagramme de classe

2.3.7.1 Définition

Le diagramme des classes est un diagramme structurel (statique) essentiel en modélisation UML, permettant de représenter divers aspects d'un système.

Ce type de diagramme met en avant deux éléments principaux :

- **Les classes**, qui incluent à la fois les attributs (les données membres) et les méthodes (les fonctions ou comportements associés à la classe).
- **Les associations**, qui sont les relations ou liens existants entre différentes classes.

Le diagramme de classe est considéré comme le plus crucial parmi tous les types de diagrammes UML. En fait, il est le seul diagramme qui soit absolument indispensable lors de la modélisation orientée objet d'un système.

Une classe, dans ce contexte, est une représentation abstraite d'un ensemble d'objets partageant des caractéristiques communes. Elle contient toutes les informations nécessaires à la construction de ces objets, notamment la définition des attributs (les propriétés ou états des objets) et des méthodes (les comportements ou actions que les objets peuvent accomplir).

En d'autres termes, une classe peut être vue comme un modèle ou un prototype. C'est le moule ou la notice détaillée permettant de créer des instances spécifiques de l'objet. En utilisant les informations fournies par la classe, on peut instancier de multiples objets qui partagent la même structure et les mêmes comportements définis dans cette classe.

Ainsi, le diagramme de classe joue un rôle fondamental dans la phase de conception et d'analyse d'un système, en assurant une représentation claire et structurée des entités et de leurs interactions,

facilitant ainsi la compréhension et la communication entre les membres de l'équipe de développement.

2.3.7.2 Représentation des classes

Une classe dans un diagramme de classes UML est représentée par un rectangle, souvent appelé "classeur", qui est divisé en trois compartiments distincts.

➔ **Le premier compartiment** contient le nom de la classe. Ce nom suit plusieurs conventions:

- Il représente le type de l'objet instancié.
- Il commence par une lettre majuscule.
- Il est centré dans le compartiment supérieur du rectangle.
- Il est écrit en caractères gras.
- Si la classe est abstraite (c'est-à-dire qu'il est impossible d'instancier un objet de cette classe), le nom est écrit en italique.

➔ **Le deuxième compartiment** contient les attributs de la classe. Un attribut est une donnée élémentaire, également appelée "propriété", qui sert à caractériser les entités et les associations au sein du système. Les attributs définissent les états ou les propriétés des objets qui seront instanciés à partir de la classe.

➔ **Le troisième compartiment** contient les méthodes de la classe. Les méthodes décrivent la manière dont une classe interagit avec ses données et effectue des opérations spécifiques. Elles définissent les comportements ou les actions que les objets de la classe peuvent réaliser.

Ainsi, la représentation des classes dans un diagramme UML permet une visualisation claire et structurée des différentes composantes d'une classe, facilitant la compréhension et la communication des concepts de modélisation orientée objet. Les classes, avec leurs attributs et méthodes bien définis, jouent un rôle crucial dans la construction d'un système robuste et bien organisé.

2.3.7.3 Les associations entre classes

Les associations entre classes dans un diagramme UML représentent les relations entre les instances des classes. Une association indique qu'une classe contiendra une référence à l'objet de la classe associée sous la forme d'un attribut. Cela signifie qu'une classe est en relation avec une

autre pendant un certain laps de temps, définissant ainsi une interaction ou une dépendance entre elles.

Nom de l'association : Une association peut être ornée d'un texte pour nommer cette relation, éventuellement avec un sens de lecture, afin de clarifier la nature et l'intérêt de cette relation. Le nom de l'association aide à comprendre le rôle des classes dans la relation et apporte des informations supplémentaires si nécessaire.

La cardinalité (ou multiplicité) : La cardinalité indique le nombre d'instances d'une classe qui peuvent être en relation avec une instance de la classe située à l'autre extrémité de l'association. Elle est essentielle pour comprendre les contraintes de la relation entre les classes. En l'absence de spécification explicite, la cardinalité par défaut est 1.

Voici un tableau récapitulatif des différentes cardinalités et leurs significations :

2.10. Table: La cardinalité

Cardinalités	Significations
1	Un et un seul
0..1	Zéro ou un
N..N	De N à N (entiers naturels)
*	De zéro à plusieurs
0..*	De zéro à plusieurs
1..*	D'un à plusieurs
N	Exactement N (entier naturel)

Les associations, avec leurs noms et cardinalités, permettent de définir clairement comment les objets des différentes classes interagissent et sont liés les uns aux autres. Cela aide à modéliser de manière précise les relations complexes au sein d'un système, assurant ainsi une meilleure conception et compréhension des interactions entre les composants du système.

2.3.7.4 Diagramme de classe du projet

La figure ci-après représente le diagramme de classe pour la réalisation de notre logiciel

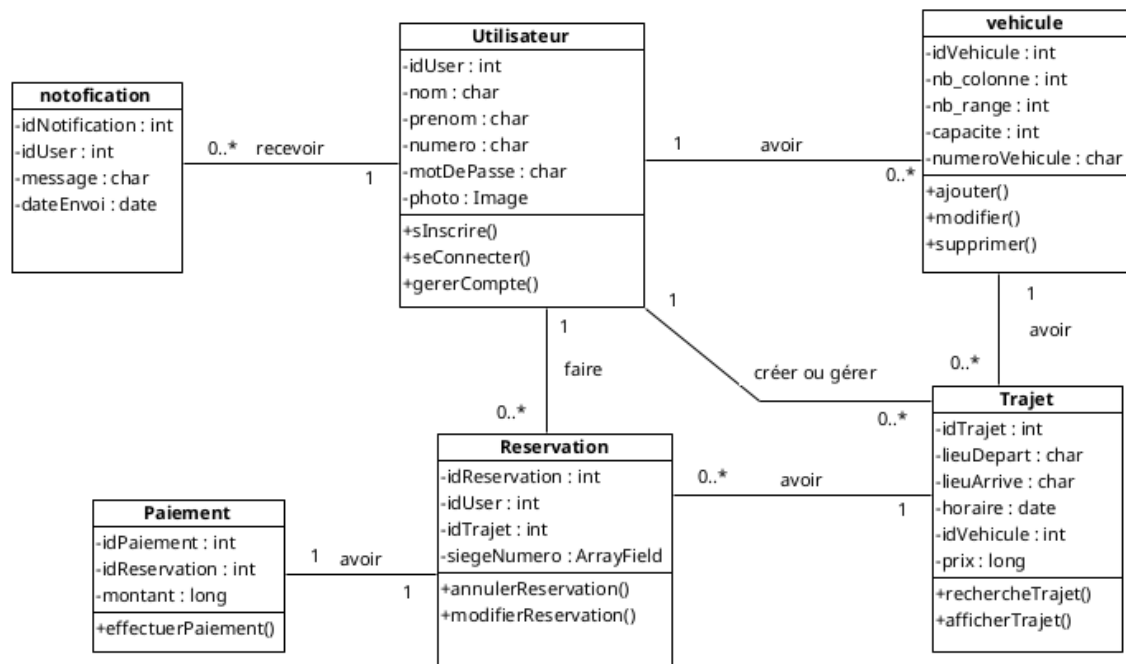


Figure 2.17 Diagramme de classe du projet

2.4 Conclusion

La phase de conception est cruciale dans tout projet informatique, et notre application de réservation de transport public ne fait pas exception. À travers ce chapitre, nous avons détaillé les différentes étapes et approches utilisées pour concevoir notre application, en nous basant sur les besoins identifiés et les limites des applications existantes. Nous avons élaboré des diagrammes UML pour visualiser et clarifier les aspects fonctionnels et techniques de notre solution. Ces diagrammes, incluant les cas d'utilisation, les diagrammes de séquence et d'activité, ont permis de structurer efficacement les fonctionnalités de notre application, de définir les interactions entre les utilisateurs et le système, et d'assurer une compréhension commune parmi les développeurs et les parties prenantes. Grâce à cette conception rigoureuse, nous sommes désormais prêts à entamer la phase de développement avec une vision claire et détaillée de notre projet.

CHAPITRE 3 PRESENTATION ET EVALUATION DE L'APPLICATION

3.1 Introduction

Notre projet nécessite une réalisation rigoureuse pour garantir son efficacité et sa stabilité. Pour ce faire, nous avons utilisé plusieurs technologies adaptées à nos besoins. Le backend de l'application a été développé avec Django, un framework Python, et nous avons utilisé SQL via PostgreSQL pour gérer notre base de données. L'architecture logicielle repose sur le modèle MVC (Modèle-Vue-Contrôleur) pour assurer une organisation cohérente et modulaire de nos composants. Après avoir effectué des études approfondies et conçu les détails de l'application, il est temps de passer à la phase de réalisation. Ce chapitre est consacré à la présentation des technologies utilisées, telles que les langages de programmation, les logiciels, et les méthodes de travail. Nous détaillerons également les fonctionnalités de l'application, en illustrant son interface utilisateur et en expliquant son fonctionnement. Ensuite, nous procéderons à une évaluation complète de l'application, en mettant en avant ses points forts et les aspects à améliorer. Des tests pratiques et des retours d'utilisateurs seront inclus pour garantir que l'application répond aux attentes et fonctionne de manière optimale. Ce chapitre fournira ainsi une vue d'ensemble complète de l'application, depuis sa conception jusqu'à son utilisation finale, permettant de mesurer son efficacité et son adéquation avec les besoins des utilisateurs.

3.2 Environnement de travail

3.2.1 Environnement matériel

Pour la réalisation de notre travail, nous avons utilisé un seul micro-ordinateur de :

- Modèle : ASUS
- Processeurs : 4 × Intel® Core™ i5-2430M CPU @ 2.40GHz
- Mémoire : 7.7 Gio de mémoire vive
- Processeur graphique : AMD CAICOS
- Système d'exploitation : Linux Kubuntu

Pour les tests, nous avons utilisé un téléphone Android de :

- Modèle : LEX620
- Version : 7.1
- Mémoire (RAM) : 2,73Gio
- Taille de l'écran : 1080 x 1920 pixels

3.2.2 Environnement de développement

L'environnement de développement constitue un élément crucial pour la réussite de notre projet. Il comprend l'ensemble des outils et plateformes utilisés pour coder, tester et déployer notre application. Cet environnement a été soigneusement choisi pour garantir une efficacité maximale tout au long du cycle de développement. Dans cette section, nous allons décrire en détail les différents outils et technologies qui composent notre environnement de développement, et expliquer comment chacun d'entre eux contribue à la création et à la maintenance de notre application de réservation de transport public.

3.2.2.1 Langages :

3.2.2.1.1 Java :

Java est un langage de programmation de haut niveau, orienté objet et largement utilisé dans le développement d'applications entreprise, de mobiles à grande échelle, et même dans l'Internet des objets. Pour notre projet de développement d'application mobile de suivi de transport en temps réel, Java a été choisi pour ses nombreux avantages :

1. **Robustesse et portabilité** : Java est reconnu pour sa robustesse et sa portabilité, grâce à sa capacité à être exécuté sur diverses plateformes sans nécessiter de modifications substantielles.
2. **Richesse des bibliothèques et frameworks** : Java dispose d'une vaste bibliothèque standard et de nombreux frameworks, qui simplifient le développement en offrant des composants prêts à l'emploi pour la gestion des transactions, la sécurité, et bien plus encore.
3. **Support de la communauté et documentation** : Avec une communauté mondiale active, Java bénéficie d'un vaste soutien en ligne, de ressources pédagogiques et de mises à jour régulières, facilitant l'apprentissage continu et la résolution rapide des problèmes.
4. **Flexibilité et performance** : Java offre une grande flexibilité grâce à sa capacité à intégrer facilement d'autres technologies et langages, tout en fournissant des performances élevées adaptées aux applications critiques comme le suivi en temps réel de notre projet.

Dans notre projet, Java joue un rôle central, notamment à travers son utilisation avec Android Studio, qui est largement adopté pour le développement d'applications Android. La combinaison de la fiabilité de Java, et de la puissance d'Android Studio garantit un développement efficace [2.4]

3.2.2.1.2 Python :

Python est un langage de programmation interprété, orienté objet et de haut niveau, apprécié pour sa lisibilité et sa simplicité. Il est largement utilisé dans le développement web, l'analyse de données, l'intelligence artificielle, et bien d'autres domaines [2.5]. Pour notre projet de réservation de transport public, Python a été choisi pour ses nombreux avantages :

1. **Lisibilité et simplicité** : La syntaxe claire de Python permet de réduire le temps de développement et de faciliter la maintenance du code.
2. **Richesse des bibliothèques** : Python dispose d'une vaste collection de bibliothèques et de frameworks, tels que Django, qui accélèrent le développement en offrant des fonctionnalités prêtes à l'emploi.
3. **Support de la communauté** : Une grande communauté de développeurs Python offre un soutien et des ressources abondantes, facilitant la résolution de problèmes et l'apprentissage.
4. **Flexibilité et extensibilité** : Python peut être facilement intégré avec d'autres technologies et langages, permettant une grande flexibilité dans le développement de notre application.

Dans notre projet, Python joue un rôle central, notamment à travers le framework Django, qui est utilisé pour construire le backend de l'application. Django offre une structure robuste et modulaire, facilitant la gestion des données, l'authentification des utilisateurs, et l'interaction avec la base de données. La simplicité et l'efficacité de Python, combinées à la puissance de Django, garantissent un développement rapide et une application performante. [1.2]

3.2.2.2 Framework et bibliothèque :

3.2.2.2.1 Django :

Django est un framework web open-source construit sur Python, choisi pour ses capacités avancées de développement d'API RESTful. Dans notre projet de réservation de transport public, Django a été sélectionné en raison de ses nombreux avantages clés :

1. **Structure modulaire et robuste** : Django propose une architecture basée sur le modèle Modèle-Vue-Template (MVT), simplifiant la séparation des préoccupations et facilitant la maintenance à long terme de l'application.
2. **Gestion des données simplifiée** : Grâce à son ORM intégré, Django permet de définir des modèles de données de manière déclarative, simplifiant ainsi l'interaction avec la base de données et réduisant le temps de développement.
3. **Sécurité intégrée** : Django inclut des fonctionnalités de sécurité avancées telles que la protection contre les injections SQL, les attaques CSRF et la gestion flexible des

permissions utilisateur, garantissant la robustesse de l'application contre les menaces potentielles.

4. **Développement rapide et efficace** : En exploitant les fonctionnalités prêtes à l'emploi de Django, comme son système d'authentification et son administration automatique, notre équipe peut se concentrer sur le logique métier spécifique de la réservation de transport, plutôt que de réinventer la roue.

En résumé, Django combine la simplicité et la puissance de Python avec une structure modulaire et des fonctionnalités intégrées qui accélèrent le développement d'API RESTful robustes et évolutives. Pour notre projet, cette combinaison garantit non seulement une mise en œuvre efficace mais aussi une maintenance aisée et sécurisée à long terme. [1.1][2.6]

3.2.2.2 Retrofit 2 :

Retrofit 2 est une bibliothèque Java qui simplifie la communication réseau dans les applications Android et Java en général. Pour notre projet de réservation de transport public, nous avons choisi Retrofit 2 pour ses nombreux avantages :

1. **Simplicité et facilité d'utilisation** : Retrofit 2 utilise une interface déclarative pour définir les requêtes HTTP et les interactions avec les API RESTful. Cette approche intuitive réduit la complexité du code réseau et facilite la maintenance.
2. **Intégration transparente** : Retrofit 2 s'intègre harmonieusement avec d'autres bibliothèques et frameworks Java, notamment avec Gson pour la désérialisation des réponses JSON, ce qui simplifie la gestion des données entre le serveur Django et l'application cliente.
3. **Gestion flexible des requêtes** : Grâce à son support pour les méthodes HTTP (GET, POST, PUT, DELETE) et à la possibilité d'ajouter des intercepteurs pour la gestion des en-têtes et des requêtes, Retrofit 2 offre une flexibilité totale dans la configuration des appels réseau.
4. **Sécurité et fiabilité** : Retrofit 2 permet de mettre en œuvre des stratégies robustes de gestion des erreurs et de sécurité, essentielles pour assurer la fiabilité des communications réseau dans notre application de réservation de transport public.

En résumé, Retrofit 2 est un choix idéal pour notre projet grâce à sa simplicité, son intégration transparente avec les API RESTful, sa flexibilité dans la gestion des requêtes et sa capacité à garantir la sécurité et la fiabilité des communications réseau. En combinant Retrofit 2 avec la

puissance de Django côté serveur, notre équipe peut construire une application efficace et performante pour la gestion des réservations de transport. [2.7]

3.2.2.2.3 Room :

Room est un Framework SQLite pour une Gestion de Données Locale Simplifiée. Dans le développement d'applications mobiles, la gestion efficace des données locales est cruciale pour offrir une expérience utilisateur fluide et réactive. C'est là que le framework Room SQLite entre en jeu. Développé par Google, Room simplifie considérablement la persistance des données dans les applications Android en s'appuyant sur la puissance de SQLite.[2.12]

Avantages clés de Room :

- **Simplicité et Facilité d'Utilisation** : Room adopte une approche déclarative pour la définition des entités, des relations et des requêtes, ce qui rend la gestion des données locale intuitive et accessible.
- **Intégration transparente avec les autres bibliothèques** : Room s'intègre parfaitement avec d'autres bibliothèques populaires du monde Android, telles que Retrofit . Cela permet une gestion fluide des données entre les couches réseau, de persistance et d'interface utilisateur, garantissant une cohérence et une réactivité optimales.
- **Performances optimisées** : Room tire parti des fonctionnalités natives de SQLite pour offrir des performances de lecture et d'écriture de données rapides et efficaces. De plus, il utilise des techniques de compilation pour optimiser les requêtes et minimiser les accès à la base de données, garantissant ainsi une expérience utilisateur fluide même avec des ensembles de données volumineux.

3.2.2.2.4 Mapsforge

Mapsforge est une bibliothèque open-source permettant d'intégrer des cartes hors ligne dans les applications mobiles. Elle est particulièrement utile pour les applications nécessitant des cartes personnalisées et une fonctionnalité de navigation sans dépendre d'une connexion internet continue.

-Principales fonctionnalités de Mapsforge :

- ◆ **Affichage de cartes vectorielles hors ligne** : Les cartes sont stockées localement sur l'appareil, ce qui permet une consultation rapide et sans interruption.

- ◆ **Personnalisation des cartes** : Grâce aux fichiers de style et aux thèmes, les cartes peuvent être personnalisées pour répondre à des besoins spécifiques, tels que l'affichage de différents types de routes, de points d'intérêt, etc.
- ◆ **Compatibilité Android** : Mapsforge est optimisé pour les applications Android, facilitant ainsi l'intégration dans les projets développés avec Android Studio.
- ◆ **Support des tuiles en ligne** : En plus des cartes hors ligne, Mapsforge peut également afficher des tuiles en ligne, offrant ainsi une flexibilité supplémentaire.
- ◆ **Gestion des données de navigation** : Les fonctionnalités de navigation incluent le calcul de routes et la gestion des traces GPS.

Dans le cadre de ce projet, Mapsforge a été utilisé pour permettre aux utilisateurs de voir les trajets disponibles et de suivre en temps réel la position des véhicules. Cette intégration garantit que les utilisateurs peuvent accéder aux informations de localisation même sans connexion internet, améliorant ainsi l'expérience utilisateur globale.

Avec Mapsforge, le projet bénéficie d'une solution robuste et flexible pour la gestion des cartes, essentielle pour une application de réservation de transport efficace et fiable.[2.8]

3.2.3 Environnement logiciel

3.2.3.1 Visual Studio Code :

Visual Studio Code est un éditeur de code source puissant, moderne et gratuit, développé par Microsoft. Doté d'une interface épurée et d'un moteur de rendu rapide, VSCode offre une expérience de développement fluide et efficace, compatible avec une grande variété de langages, de frameworks et d'environnements. Conçu pour s'intégrer parfaitement avec les outils et les processus de développement les plus courants, VSCode facilite des tâches essentielles comme la gestion de version, le débogage ou l'utilisation du cloud et des conteneurs. Grâce à son écosystème riche en extensions et en paramètres personnalisables, VSCode peut être adapté aux besoins spécifiques de chaque développeur, ce qui, associé à sa grande flexibilité et à sa communauté active, en fait un outil de choix pour les développeurs de tous niveaux.

Principaux avantages de VSCode:

- Riche en fonctionnalités : VSCode propose de nombreuses fonctionnalités avancées telles que la complétion de code intelligent, le débogage intégré, la gestion de version avec Git, et bien plus encore.

- **Personnalisable** : VSCode est très personnalisable grâce à un vaste écosystème d'extensions, de thèmes et de paramètres. Les développeurs peuvent adapter l'éditeur à leurs besoins spécifiques.
- **Rapide et léger** : Comparé à d'autres éditeurs lourds, VSCode reste rapide et consomme peu de ressources système, ce qui le rend très réactif.
- **Intégration avec les outils** : VSCode s'intègre parfaitement avec de nombreux outils populaires tels que les gestionnaires de source (Git), les clouds (Azure, AWS), les environnements de développement (Docker), etc.

3.2.3.2 Android studio :

Android Studio est un environnement de développement intégré (IDE) puissant et complet, développé par Google, spécifiquement pour le développement d'applications Android. Conçu sur la base d'IntelliJ IDEA de JetBrains, Android Studio offre une suite d'outils et de fonctionnalités qui simplifient la création, le test et le déploiement d'applications Android. Son interface utilisateur intuitive et ses capacités de développement avancées permettent aux développeurs de produire des applications de haute qualité de manière efficace et efficiente.

Principaux avantages d'Android studio:

- **Outils de développement intégrés** : Android Studio inclut des outils de développement essentiels comme un éditeur de code intelligent, un émulateur Android rapide, des outils de profilage de performances, et bien plus encore.
- **Interface utilisateur intuitive** : L'IDE propose une interface utilisateur intuitive et des outils de conception visuelle tels que le Layout Editor, qui permet de créer et de modifier les interfaces utilisateur de manière simple et visuelle.
- **Support de Kotlin et Java** : Android Studio prend en charge Kotlin et Java, les deux langages principaux pour le développement Android, facilitant ainsi le travail des développeurs avec leurs langages de prédilection.
- **Débogage et test avancés** : L'IDE propose des fonctionnalités de débogage avancées, ainsi que des outils de test intégrés pour vérifier les performances et la compatibilité des applications sur différentes versions d'Android et types d'appareils.[2.11][1.3]

3.2.3.3 Postman :

Postman est un outil populaire utilisé par les développeurs pour tester, déployer et collaborer sur des API. Cet outil gratuit et multiplateforme offre une interface conviviale pour

faciliter le processus de développement d'API. Postman supporte diverses méthodes HTTP et permet aux développeurs de créer, organiser et automatiser des tests pour leurs API. Parmi ses fonctionnalités essentielles, on trouve la possibilité d'envoyer des requêtes HTTP et d'analyser les réponses, la gestion des environnements pour faciliter le déploiement sur différentes configurations, la documentation automatisée des API, et la collaboration au sein d'équipes grâce à la synchronisation de collections d'API via le cloud. Grâce à ses extensions et intégrations avec d'autres outils de développement, Postman offre une flexibilité accrue pour répondre aux besoins spécifiques des développeurs et des équipes travaillant sur des projets d'intégration et de développement d'API modernes.

3.2.3.4 Visual Paradigm :

Visual Paradigm est un outil de modélisation et de gestion de projet complet, conçu pour faciliter la conception, la documentation et la gestion des systèmes logiciels. Utilisé par des développeurs, des architectes logiciels et des chefs de projet à travers le monde, Visual Paradigm offre une suite d'outils de modélisation UML, de gestion des exigences, de génération de code et de documentation, le tout dans une interface conviviale.

Principaux avantages de Visual Paradigm :

- **Modélisation UML complète :** Visual Paradigm prend en charge l'intégralité des diagrammes UML, facilitant la création et la gestion des diagrammes de classes, de séquences, d'activités, et bien d'autres.
- **Interface utilisateur intuitive :** Visual Paradigm offre une interface utilisateur claire et intuitive, facilitant l'apprentissage et l'utilisation de l'outil pour les développeurs et les chefs de projet de tous niveaux.
- **Documentation et rapports automatiques :** L'outil peut générer automatiquement des documents et des rapports basés sur les modèles et les diagrammes, ce qui est essentiel pour la documentation et la communication des projets.

3.2.4 Base de donne

3.2.4.1 PostgreSQL :

PostgreSQL, également connu sous le nom de Postgres, est un système de gestion de base de données relationnelle open-source et hautement évolutif. Développé depuis les années 1980, PostgreSQL s'est imposé comme l'un des SGBD les plus avancés et les plus populaires au monde.

[1.5]

Principaux avantages de PostgreSQL :

- Conformité aux normes SQL : PostgreSQL implémente de manière complète le standard SQL, offrant ainsi une grande compatibilité avec d'autres systèmes de gestion de bases de données.
- Sécurité avancée : Le système de contrôle d'accès et d'authentification de PostgreSQL est très développé, offrant une sécurité robuste pour les données sensibles.
- Gestion des données complexes : PostgreSQL gère nativement des types de données complexes tels que les JSON, les tableaux, les géométries, etc., le rendant adapté aux applications modernes.
- Performances élevées : Grâce à des optimisations de requêtes et une gestion efficace de la mémoire, PostgreSQL offre des performances exceptionnelles, y compris sur des jeux de données volumineux.

3.2.4.2 SQLite

SQLite est un système de gestion de base de données relationnelle léger, embarqué et open-source. Conçu pour être simple, compact et efficace, SQLite s'est imposé comme l'un des SGBD les plus populaires et largement utilisés, en particulier dans les applications mobiles et embarquées. [1.4]

Principaux avantages de SQLite :

- Légèreté et compacité : SQLite a une empreinte mémoire très faible, permettant son intégration dans des environnements avec des ressources limitées, comme les appareils mobiles.
- Facilité d'utilisation : SQLite est très facile à configurer et à utiliser, ne nécessitant pas de serveur dédié ni de configuration complexe.
- Autonomie : SQLite fonctionne de manière autonome, sans avoir besoin d'un serveur ou d'un processus d'administration séparé.
- Pas de maintenance : SQLite ne nécessite pratiquement aucune maintenance, ce qui en fait un choix idéal pour les applications embarquées et les applications mobiles.

3.2.5 Utilisation des données cartographiques

3.2.5.1 OpenStreetMap

OpenStreetMap (OSM) est une initiative collaborative visant à créer une carte mondiale libre et accessible à tous. Contrairement aux solutions commerciales, OSM permet à des contributeurs du monde entier d'ajouter, modifier et mettre à jour les informations cartographiques, offrant ainsi

une précision et une actualisation constantes adaptées à notre projet de réservation de transport public.

Mapsforge, d'autre part, est une bibliothèque open-source Java pour afficher des cartes basées sur les données d'OpenStreetMap. Pour notre application mobile de suivi de transport en temps réel, nous avons choisi Mapsforge en raison de ses avantages significatifs :

- **Personnalisation et flexibilité** : Mapsforge permet une personnalisation complète des cartes en fonction de nos besoins spécifiques. Nous pouvons ajuster le style des cartes, ajouter des marqueurs personnalisés et intégrer des données contextuelles pour une expérience utilisateur optimale.
- **Performance optimisée** : En utilisant des formats de données optimisés et en exploitant les capacités de rendu matériel, Mapsforge garantit des performances élevées même sur des appareils mobiles avec des ressources limitées, essentiel pour notre application de suivi en temps réel.
- **Intégration transparente avec OSM** : Mapsforge utilise les données d'OpenStreetMap, ce qui nous assure une source de données fiable et constamment mise à jour. Cela nous permet de fournir des informations précises et actuelles sur les itinéraires, les points d'intérêt et les conditions de circulation.
- **Support communautaire et évolutivité** : En tant que projet open-source avec une communauté active, Mapsforge bénéficie d'un support continu et de mises à jour régulières, assurant ainsi une évolutivité et une adaptation aux nouvelles exigences de notre application de suivi de transport. [2.9]

3.2.5.2 Overpass API :

Overpass API est un service web puissant permettant d'interroger et de récupérer des données provenant de la base de données d'OpenStreetMap (OSM). Contrairement aux services commerciaux, Overpass API offre une flexibilité exceptionnelle pour accéder à des informations géospatiales précises et à jour, essentielles pour notre application de réservation de transport public.

Pour notre projet, Overpass API présente plusieurs avantages significatifs :

- **Accès en temps réel aux données OSM** : Overpass API permet de récupérer les données OSM en temps réel, ce qui est crucial pour fournir aux utilisateurs des informations actualisées sur les trajets, les arrêts de bus, et autres points d'intérêt.

- **Flexibilité des requêtes** : Avec Overpass API, nous pouvons formuler des requêtes personnalisées pour extraire des informations spécifiques telles que les routes, les bâtiments, les points d'intérêt, et bien plus encore. Cette flexibilité nous permet d'adapter les données à nos besoins spécifiques sans surcharge inutile.
- **Optimisation des performances** : Les requêtes Overpass API peuvent être optimisées pour récupérer uniquement les données nécessaires, minimisant ainsi le temps de réponse et améliorant les performances de l'application. Cela est particulièrement important pour offrir une expérience utilisateur fluide et rapide.
- **Facilité d'intégration** : L'API est facile à intégrer avec d'autres composants de notre application, notamment Mapsforge, pour afficher les données récupérées directement sur les cartes. Cette intégration transparente améliore la cohérence et l'efficacité de notre solution.

En résumé, l'utilisation de l'Overpass API dans notre projet de réservation de transport public nous offre un accès direct et flexible aux données géospatiales d'OpenStreetMap, optimisant ainsi la précision, la performance et l'expérience utilisateur de notre application. [2.10]

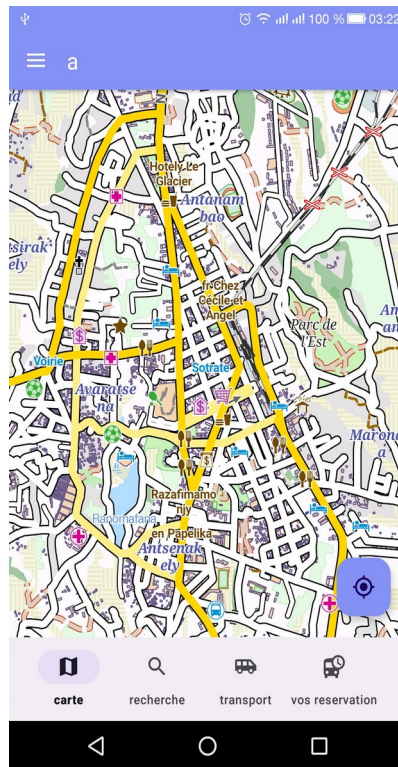
3.3 Présentation des interfaces graphiques :

3.3.1 Introduction

Dans cette section, nous allons explorer les différentes interfaces graphiques de notre application de réservation de transport public. L'interface utilisateur joue un rôle crucial dans la facilitation de l'interaction entre les utilisateurs et le système, offrant une expérience fluide et intuitive.

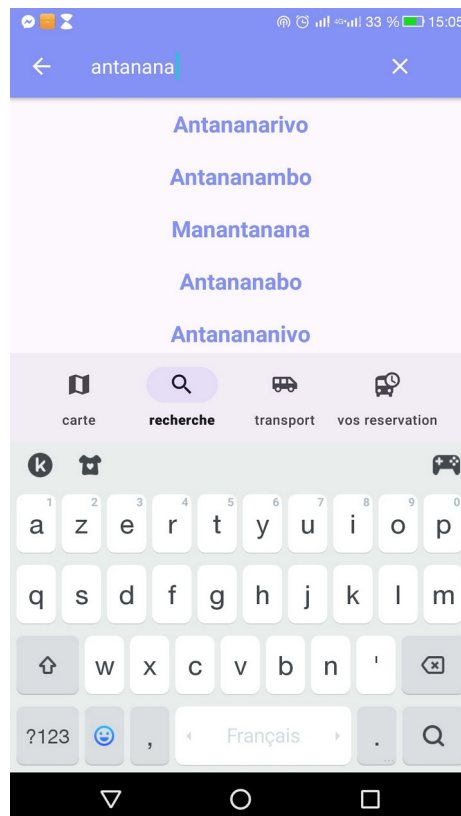
3.3.2 Interface carte

L'interface Carte est une composante centrale de l'application, permettant aux utilisateurs d'accéder à une visualisation interactive et en temps réel de leur position actuelle. Grâce à cette interface, les utilisateurs peuvent suivre leur emplacement exact sur la carte, ce qui est essentiel pour une navigation précise et pratique. En plus de montrer la position de l'utilisateur, cette interface affiche également le chemin des trajets disponibles, facilitant la planification et la visualisation des itinéraires. L'intégration du bouton de recherche permet aux utilisateurs de localiser facilement des lieux spécifiques en saisissant des critères de recherche. Les résultats de cette recherche sont alors directement affichés sur la carte, offrant une expérience cohérente et intuitive.



3.3.3 Interface « Recherche lieu »

L'interface permet de rechercher facilement des lieux et de les afficher sur une carte.



3.3.4 Interface Transport recherche

L'interface Transport Recherche permet de trouver, réserver des trajets de transports en commun. On saisissons les critères de recherche et filtrer les résultats.



3.3.5 Interface réservation Trajet

L'interface de réservation joue un rôle essentiel dans le processus de réservation de trajets pour les utilisateurs. Cette interface propose un affichage détaillé des informations du trajet, y compris le lieu de départ et d'arrivée, l'heure de départ, et le prix du billet. La disposition des sièges est représentée visuellement, où chaque siège est numéroté et coloré en fonction de sa disponibilité : les sièges disponibles sont en blanc, les sièges sélectionnés par l'utilisateur sont en bleu pourpre, et les sièges déjà réservés sont en bleu. Les utilisateurs peuvent facilement sélectionner et désélectionner les sièges en cliquant dessus. Le montant total à payer est automatiquement mis à jour en fonction du nombre de sièges sélectionnés et est affiché de manière claire en bas de l'interface. Une fois les sièges choisis, les utilisateurs peuvent confirmer leur réservation en cliquant sur le bouton "valider la réservation". Cette interface offre une expérience

utilisateur fluide et intuitive, facilitant la réservation de sièges pour un trajet donné. Elle est conçue pour être simple d'utilisation, avec des interactions claires et un retour visuel immédiat, rendant le processus de réservation rapide et efficace.

3.4 Conclusion

Ce chapitre a exploré les aspects techniques et fonctionnels de l'application, en détaillant l'intégration de « Django » pour le backend, « Mapsforge » pour la gestion des cartes, « Retrofit » pour les requêtes réseau, « Room » pour le stockage des données hors connexion et Java pour le développement Android. Chaque technologie a été sélectionnée pour sa capacité à offrir une expérience utilisateur fluide et performante. Nous avons mis en évidence comment ces outils travaillent ensemble pour permettre une gestion efficace des données, une navigation intuitive, et des communications sécurisées entre le client et le serveur. Les interfaces graphiques ont été conçues pour être intuitives et accessibles, facilitant la navigation, la recherche de lieux et la réservation de trajets. En résumé, ce chapitre illustre comment l'intégration harmonieuse de diverses technologies peut conduire à une application mobile conviviale, répondant aux besoins des utilisateurs de manière optimale.

CONCLUSION ET PERSPECTIVES

Nous arrivons maintenant au terme de ce projet, dont l'objectif principal était de développer une solution complète pour la gestion des trajets et des réservations de transport en ligne. Le chapitre 1 a défini le contexte et la problématique, soulignant la nécessité de moderniser les systèmes de réservation de transport tout en détaillant les fonctionnalités essentielles de l'application. Le chapitre 2 s'est concentré sur la conception et l'architecture du système en utilisant UML pour modéliser les relations entre les différentes entités et les fonctionnalités à implémenter. Le chapitre 3 a présenté les choix technologiques cruciaux, tels que Django pour le backend, java pour l'interface, Mapsforge pour la gestion des cartes,... , et a également détaillé la mise en œuvre de plusieurs interfaces utilisateur.

Une des perspectives importantes est la création de coopératives de transport en ligne, permettant à différents opérateurs de gérer leurs flottes et réservations via une application commune. L'intégration du suivi en temps réel des véhicules reste une perspective majeure, permettant aux utilisateurs de suivre l'emplacement de leur véhicule en temps réel. L'optimisation des itinéraires en fonction du trafic et des conditions routières est également envisagée, ainsi que l'amélioration continue de l'expérience utilisateur avec des interfaces plus intuitives et des processus de réservation simplifiés. Enfin, l'extension des services pour inclure des options de transport multimodal pourrait offrir une solution encore plus complète aux utilisateurs. Ces perspectives prometteuses permettront de répondre de manière optimale aux besoins des utilisateurs et des opérateurs de transport.

ANNEXES

Annexe 1 : Extrait code source JAVA

```
12 public class RetrofitClient {
    3 usages
13     private static Retrofit retrofit = null;
    @ Ranto
14     public static Retrofit getClient(String baseUrl, String authToken) {
15         if (retrofit == null) {
16             HttpLoggingInterceptor loggingInterceptor = new HttpLoggingInterceptor();
17             loggingInterceptor.setLevel(HttpLoggingInterceptor.Level.BODY);
18
19             OkHttpClient.Builder clientBuilder = new OkHttpClient.Builder();
20             clientBuilder.addInterceptor(loggingInterceptor);
21
22             if (authToken != null) {
23                 AuthInterceptor authInterceptor = new AuthInterceptor(authToken);
24                 clientBuilder.addInterceptor(authInterceptor);
25             }
26             OkHttpClient client = clientBuilder.build();
27
28             retrofit = new Retrofit.Builder()
29                 .baseUrl(baseUrl)
30                 .client(client)
31                 .addConverterFactory(GsonConverterFactory.create())
32                 .build();
33         }
34         return retrofit;
35     }
36 }
```

Ce code crée et configure un client Retrofit statique pour effectuer des requêtes réseau vers une API. Il commence par créer une instance statique du client Retrofit, puis définit l'URL de base, les intercepteurs et le convertisseur de corps de réponse. Enfin, il fournit une méthode pour obtenir l'instance du client Retrofit, qui peut ensuite être utilisée pour effectuer des requêtes réseau. Cette approche permet une utilisation simple et efficace du client Retrofit pour interagir avec une API.

BIBLIOGRAPHIE

[1.1]	Gérard Rozsavolgyi : « Tuto-Django », 27 juin 2023,90 pages
[1.2]	Patrick Fuchs et Pierre Poulain, « Cours de Python – Université de Paris », page 203 - 218
[1.3]	Neil Smyth : « Android Studio Développement Essentials - Java Edition »,742 pages
[1.4]	Jonathan Gennick : «SQL Pocket Guide» ,240 pages
[1.5]	Regina O. McCormack : «PostgreSQL: Up and Running » ,306 pages

WEBOGRAPHIE

[2.1]	https://reservation.cotisse-transport.com
[2.2]	httpswww.taxibrousse.mg/
[2.3]	https://soatransplus.mg
[2.4]	https://docs.oracle.com/en/java
[2.5]	https://www.python.org/
[2.6]	https://docs.djangoproject.com/
[2.7]	https://square.github.io/retrofit/
[2.8]	https://github.com/mapsforge/mapsforge
[2.9]	https://wiki.openstreetmap.org
[2.10]	https://wiki.openstreetmap.org/wiki/Overpass_API
[2.11]	https://developer.android.com/develop
[2.12]	https://developer.android.com/reference/androidx/room/RoomDatabase

FICHE DE RENSEIGNEMENTS

Nom : RAKOTOMALALA
Prénoms : Herimandranto Fandresena
Adresse de l’auteur : 130C Antanifotsy
Téléphone : 034 67 369 24
E-mail : herimandrantofandresen@gmail.com



Titre du mémoire : APPLICATION MOBILE DE RESERVATION DE
TRANSPORT PUBLIC

Nombre de pages :

Nombre de tableaux :

Nombre de figures :

Encadreur pédagogique :

Téléphone :

Mail :

FAMINTINANA SY TENY MANADANJA

Ny fomba famandriahan-toerana eo amin'ny fitateram-bahoaka dia misedra olana tahakin'ny tsy fisihan'ny famandriahan-toerana "En ligne" sy fahasarotan'ny fandoavana saran-dalana. Izany no namoronana ity "Application" ity mba hamahana ireo olana ireo amin'ny alàlan'ny fampiasana teknolojia avo lenta toy ny "Python" ho an'ny "backend", "PostgreSQL" sy "SQLite" ho an'ny "Base de données" ary "java" hoany famorona ny "Application Mobile", ny "Application" dia noforoina mba hamandrihana toerana ao aminy fiara, fampandrenesana ny vaovao momban'ny fiara, safidy fandoavam-bola azo antoka, sns. Ny tanjon'ity asa ity dia ny hanatsara fitateram-bahoaka.

Teny manadanja : « Python », « Java », « PostgreSQL », « SQLite » famandrihan-toerana eny anivon'ny fitantarem-bahoaka

RESUME ET MOTS CLES

Les méthodes de réservation dans les transports en commun rencontrent des problèmes tels que l'absence de réservations en ligne et la complexité du paiement. C'est pourquoi cette application a été créée pour résoudre ces problèmes en apportant une innovation grâce à l'utilisation de technologies avancées telles que Python pour le backend, PostgreSQL et SQLite pour la bases de données, et Java pour le développement de l'application mobile. L'application a été conçue pour permettre la réservation de places dans les véhicules, fournir des notifications sur les informations concernant les véhicules, offrir des options de paiement sécurisé, etc. L'objectif de ce projet est d'améliorer les transports en commun.

Mots clés : Django, java, PostgreSQL, SQLite, réservation transport public.

ABSTRACT AND KEYS WORDS

The booking methods in public transport face issues such as the lack of online reservations and the complexity of ticket payments. This is why this application was created to solve these problems by bringing a touch of innovation through the use of advanced technologies such as Python for the back-end, PostgreSQL and SQLite for database, and Java for mobile application development. The application is designed to allow seat reservations in vehicles, provide notifications on vehicle information, offer secure payment options. The goal of this project is to improve public transport.

Keys words : Django, PostgreSQL, SQLite, public transport reservation

