

RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88
http://www.ifri-uac.net Courriel : contact@ifri.uac.bj



MÉMOIRE

pour l'obtention du

Diplôme de Licence en Informatique

Option : Génie Logiciel

Présenté par :
Freddy Jonas **GOUDOU**

Conception d'une application mobile de co-trajets domicile-travail basée sur un regroupement par quartier

Sous la supervision de :
Ing. Jean-Pierre **KOUKPAKI**

Année Académique : **2018 - 2019**

Dédicace

A

Tous ceux qui œuvrent activement pour le développement de solutions informatiques appliquées au secteur du transport en Afrique et particulièrement au Bénin.

Remerciements

Nous tenons particulièrement à remercier :

- Notre famille, pour tout le soutien et l'encouragement dont elle a fait preuve à notre égard durant ces trois années de formation à l'IFRI ;
- Monsieur Jean-Pierre KOUKPAKI, notre maître de mémoire pour sa disponibilité, son accompagnement et son soutien dans la réalisation de ce mémoire ;
- L'administration de l'IFRI ;
- Tous nos enseignants, camarades et amis de l'IFRI pour avoir créé un environnement propice à un partage de connaissances tout au long de notre formation.

À tous ceux qui de près ou de loin ont contribué à la réalisation de notre projet de fin d'étude trouve à travers ce travail le signe de notre profonde reconnaissance.

Table des matières

Sigles et Abréviations iv

Glossaire v

Résumé/Abstract vi

Introduction 1

1 Revue de littérature 3
- 1.1 Solutions existantes . . . 3
  - 1.1.1 Sites Web . . . 3
  - 1.1.2 Applications mobiles . . . 5
- 1.2 Limites . . . 9
  - 1.2.1 Vroomiste . . . 9
  - 1.2.2 Afri Ride . . . 9
  - 1.2.3 BlaBlacar . . . 9
  - 1.2.4 Andando . . . 10
- 1.3 Intérêt du sujet . . . 10
- 1.4 Généralité sur Android . . . 11
- 1.5 Description du processus metier . . . 11

2 Solution et choix techniques 13
- 2.1 Architecture du système : Firebase . . . 14
  - 2.1.1 Les services de Firebase . . . 14
  - 2.1.2 L'application mobile android . . . 16
- 2.2 Modélisation . . . 16
  - 2.2.1 Diagramme de cas d'utilisation . . . 16
  - 2.2.2 Diagramme de classes . . . 18
- 2.3 Processus de correspondance des trajets à la demande . . . 19
- 2.4 Motivation du choix techniques . . . 20
  - 2.4.1 Méthodologie de développement . . . 20
  - 2.4.2 Outils et technologies utilisées . . . 20

3 Résultats et discussion 24
- 3.1 Interfaces des applications . . . 24
  - 3.1.1 Interface d'acceuil de l'application . . . 24
  - 3.1.2 Interface de demande de co-trajet . . . 26
  - 3.1.3 Interface de visualisation des résultats de demande de co-trajets . . . 27

3.1.4 Interface de gestion des trajets 28
3.1.5 Interface de visualisation des propositions de co-trajets 30
3.1.6 Interface de visualisation des demandes de co-trajets l'utilisateur 31
3.2 Limites 32
3.3 Inconvenients liés à la solution 32

Conclusion générale et perspectives **34**

Bibliographie **35**

Webographie **36**

Liste des figures

1.1 Logo de Vroomiste . . . 3
1.2 Site web `vroomiste.com` . . . 4
1.3 Logo d'Afri-Ride . . . 4
1.4 Site web `afri-ride.com` . . . 5
1.5 Logo de Andando . . . 5
1.6 Andando | Liste des trajets disponibles . . . 6
1.7 Vroomiste | Accueil et trajets disponibles . . . 7
1.8 Afri Ride | Accueil et trajets en cours . . . 8
1.9 Logo de BlaBlacar . . . 9

2.1 Logo de Wego . . . 13
2.2 Comparaison entre l'architecture traditionelle et celle de Firebase . . . 14
2.3 Logo d'UML . . . 16
2.4 Diagramme de cas d'utilisation de Wego . . . 17
2.5 Diagramme de classes de Wego . . . 18
2.6 Méthode Scrum . . . 20

3.1 Accueil de l'application . . . 25
3.2 Demande de co-trajet . . . 26
3.3 Résultats de recherche de trajet . . . 27
3.4 Liste des trajets confirmés et /ou commencés . . . 28
3.5 Historique des trajets . . . 29
3.6 Propositions faites à l'utilisateur . . . 30
3.7 Toutes les recherches de co-trajet de l'utilisateur . . . 31

Sigles et Abréviations

JSON : JavaScript Object Notation 16

MVVM : Model View Viewmodel 32

TIC : Technologies de l'Information et de la Communication 1

UML : Unified Modeling Language 16

WORA : Write Once, Run Anywhere 21

XML : eXtensible Markup Language 22

Glossaire

covoitureurs :

Il s'agit des acteurs du covoiturage. En effet le covoiturage est l'utilisation conjointe et organisée (à la différence de l'auto-stop) d'une voiture automobile, par un https://www.overleaf.com/project/5d5e4l non professionnel et un ou plusieurs tiers passagers, dans le but d'effectuer un trajet commun.
vi

Résumé

Le secteur de l'automobile s'est considérablement développé ces dernières années en Afrique. Pour se deplacer rapidement les populations qui ont les moyens financiers s'offrent des motos et des voitures. Cependant il existe encore certaines personnes qui n'ont pas les moyens de s'acheter une voiture ou une moto. Ces derniers adoptent le transport en commun pour vaquer à leurs occupations. Malgré ce soulagement que les transports en commun viennent apporter à ces populations, il y en a qui réfléchissent toujours à comment réduire leurs dépenses concernant le transport de tous les jours. D'autres ne supportent pas le temps perdu chaque jour dans les embouteillages avant d'aller au travail ou faire un quelconque trajet. Sachant que les trajets domicile-travail représentent la majorité des trajets de tous les jours et que plusieurs se déplacent avec des sièges non occupés, il est temps de concevoir une solution adaptée à ces types de trajets. Ce travail va contribuer à créer un réseau de covoitureurs résidant et travaillant à peu prêt dans la même zone,à diminuer les dépenses faites chaque jour pour le deplacement et à contribuer à la protection de l'environnement en réduisant la pollution. Il va également dans le sens de réduire les embouteillages. Nous avons réalisé une application mobile dont l'objectif est de permettre à un maximum de personnes d'avoir accès au service.

Mots clés : covoiturage, covoitureur, co-trajet, application mobile, domicile-travail.

Abstract

The automotive sector has grown considerably in recent years in Africa. To move quickly, people who have the financial means buy motorcycles and cars. However, there are still some people who cannot afford a car or a motorcycle. The latter adopt public transport to go about their business. Despite the relief that public transportation brings to these populations, there are some who are still thinking about how to reduce their expenses for everyday transportation. Others cannot bear the time lost every day in traffic jams before going to work or making any commute. Knowing that home-work trips represent the majority of daily trips and that many travel with unoccupied seats, it is time to design a solution adapted to these types of trips. This work will contribute to creating a network of gls carpoolers residents and workers with little ready in the same area, to decrease the expenses made every day for the displacement and to contribute to the protection of the environment by reducing pollution. It also goes in the direction of reducing congestion. We have created a mobile application whose objective is to allow as many people as possible to have access to the service.

Key words: carpooling, carpoolers, co-path, mobile application , home-office.

Introduction

L'utilisation grandissante des Technologies de l'Information et de la Communication (TIC) dans les différents domaines socio-économiques a contribué à la digitalisation de plusieurs secteurs. Dans le secteur du transport en afrique, ces nouvelles technologies ne sont pas encore vraiment utlisées pour améliorer les conditions d'activité du secteur. La présente étude s'articulera autour de l'application des TIC dans le domaine du transport pour contribuer à une utilisation plus intelligente de la route et faire en même temps permettre aux usagers de faire des déplacement de façon économique.

Problématique

Pour garantir la sécurité des personnes et des biens et reduire les temps de trajet sur nos routes en milieu urbain, il est important de trouver de nouvelles méthodes et techniques pour rendre fluide la circulation. Malheureusement, plusieurs conditions rendent difficile cette fluidité ; et notamment sur les anciennes routes que nous pratiquons encore sont d'une largeur obligeant à rester tout le temps sur le qui-vive creant ainsi le stress nuisible à la santé. Ce qui affecte la santé de la ressource humaine dont le pays a besoin pour se développer. Une autre conséquence aussi est le temps perdu en circulation qui occasionne le retard à l'origine de la diminution du rendement au travail. Notre étude s'intègre donc à une initiative de réduction des accidents de la route et des frais de transport de chaque jours à travers une plateforme mobile de mise en relation ce qui réduirait le nombre de véhicule en circulation et donc rendrait un peu plus fluide le traffic urbain.

Contexte

La route du développement passe par le développement de la route a dit Modibo Koné. Mais à quoi sert une route si elle devient de part sa grande solicitation un danger pour ceux qui la pratiquent. Il est vrai que le transport est une pièce maîtresse pour tout types d'échange mais ceux qui font le transport à savoir les populations restent cependant plus important car ils constituent la ressource humaine qui propulse le pays vers un développement de plus en plus grand. Le présent projet vise donc à trouver une solution numérique pour rendre la route moins solicité et de ce fait de diminuer les risques d'accidents. Aussi cette solution participera à rendre les trajets plus économiques pour les covoitureurs. Nous nous sommes pour le moment concentré sur la vile de Cotonou et plus précisement sur les arrondissements 1,7 et 9 car cette ville est sans doute celle qui enrégistre le plus grand

traffic.

Objectifs

La solution vise à mettre en relation des personnes selon les quartiers ou se situent les lieux de travail et de domicile. En particulier, elle permettra :

- de créer un compte gratuitement pour gérer ces demandes de covoiturage.
- d'effectuer des demandes de co-trajet ;
- de faire des propositions de trajets en commun à des personnes ;
- de discuter via une méssagerie interne sur les conditions de trajet en commun avec le conducteur

Organisation du mémoire

Le présent travail est structuré en trois chapitres. Le chapitre 1 est une revue de littérature, qui présente les existants, leurs forces et leurs faiblesses. Le chapitre 2 expose notre solution à travers sa modélisation, son fonctionnement et les choix techniques liés à sa réalisation. Les résultats obtenus par rapport à la réalisation de la solution et les insuffisances qui lui sont liées seront quant à elles présentées dans le chapitre 3.

Chapitre 1

Revue de littérature

Introduction

Pour mener à bien la présente étude, il est nécessaire d'effectuer un état de l'art afin de mieux la situer dans le contexte des solutions existantes s'intégrant dans le même sens. Plusieurs solutions ont déjà été mises sur pied autant sous version web que sous version mobile. Après une analyse profonde de celles-ci, nous en dégagerons l'intérêt de notre sujet.Ainsi, ce chapitre fait un résumé de l'existant en ce qui concerne le covoiturage.

1.1 Solutions existantes

1.1.1 Sites Web

1.1.1.1 vroomiste.com

Vroomiste.com est la première application de transport collaboratif au Bénin.Elle a pour objectif de connecter des personnes souhaitant voyager au niveau intra et interurbain à celles ayant un moyen de transport personnel. Ce site permet de faire une recherche de trajet grâce un formulaire. Il dispose donc d'un système de matching pour faire correspondre ceux qui proposent des trajets et les passagers.



FIGURE 1.1 – Logo de Vroomiste

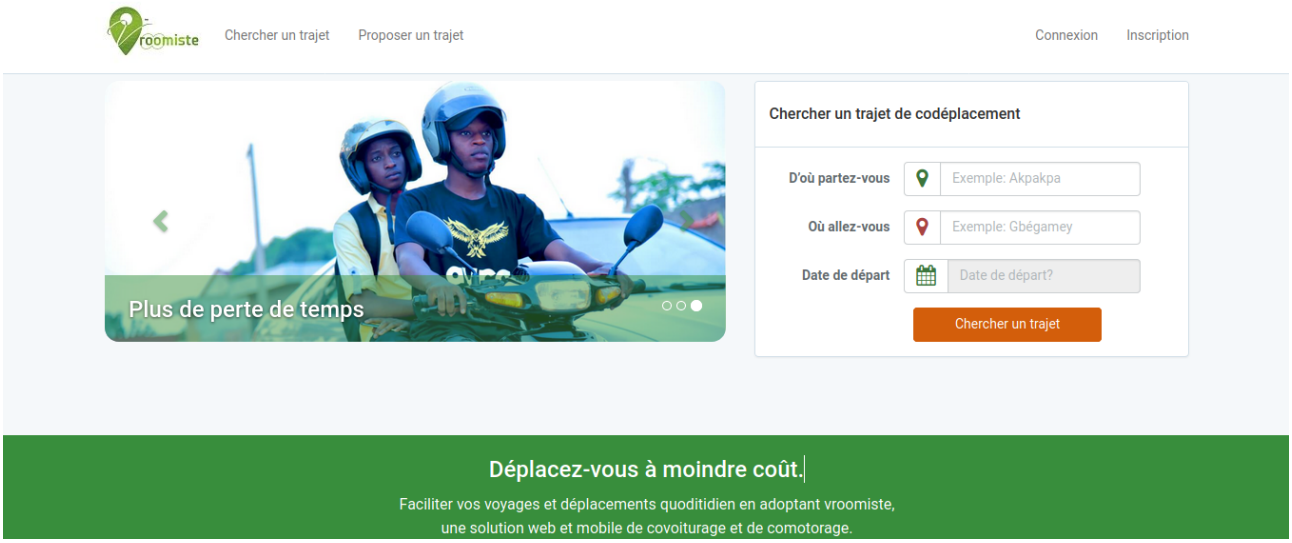


FIGURE 1.2 – Site web vroomiste.com

1.1.1.2 afri-ride.com

Afri Ride est une application sud-africaine qui permet à ses utilisateurs de renseigner leurs détails de voyage tels que le lieu de début et de fin, la date de voyage et le nombre de sièges vides disponibles, cela permet alors aux voyageurs se dirigeant dans la même direction de réserver leur siège. Afri Ride a intégré un service de chat (Mbira Chat) qui permet aux utilisateurs de communiquer (localisation en direct, SMS, appel vocal ou appel vidéo) avant même de réserver le siège ou la voiture. De plus, Afri Ride protège l'argent de ses utilisateurs en facilitant tous les paiements via une plateforme sécurisée 3D pour éviter toute utilisation frauduleuse par des tiers.



FIGURE 1.3 – Logo d'Afri-Ride

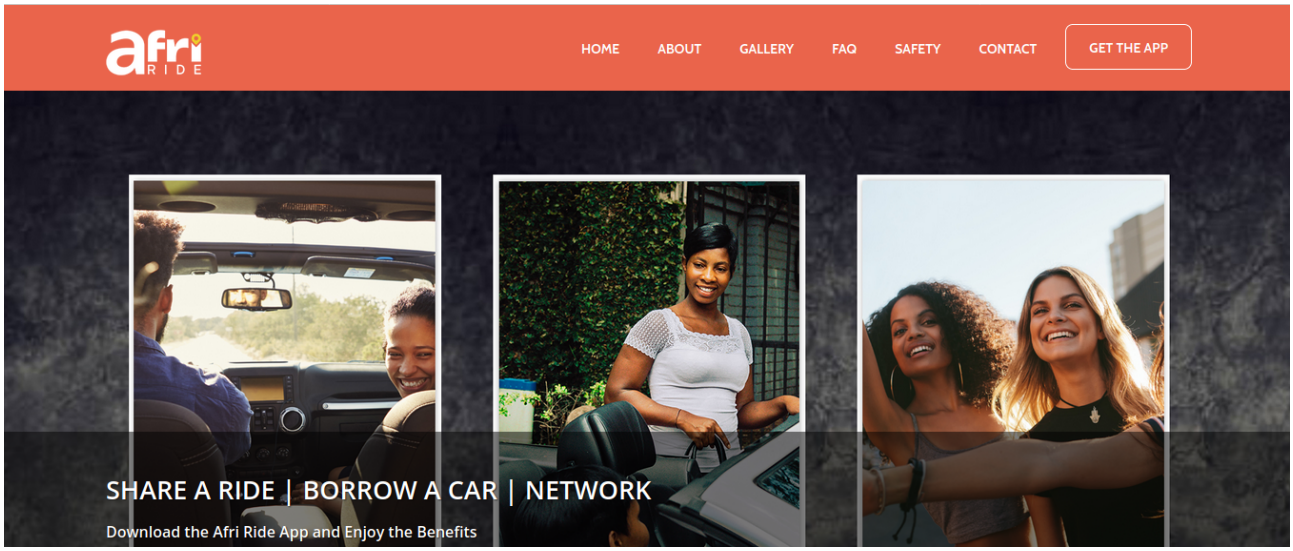


FIGURE 1.4 – Site web afri-ride.com

1.1.2 Applications mobiles

1.1.2.1 Andando

Andando permet de trouver une personne avec qui covoiturer, quel que soit votre trajet : domicile-lieu de travail, domicile- marché, quartier- quartier, ville- ville, L'application permet non seulement aux passagers de faire des économies mais aussi permet aux conducteurs de trouver plus facilement des clients. Elle est disponible sur playstore à **sa version3.2.3**[1]



FIGURE 1.5 – Logo de Andando

[1]Source :https://https://play.google.com/store/apps/details?id=dev.cityway.andando&hl=en_US

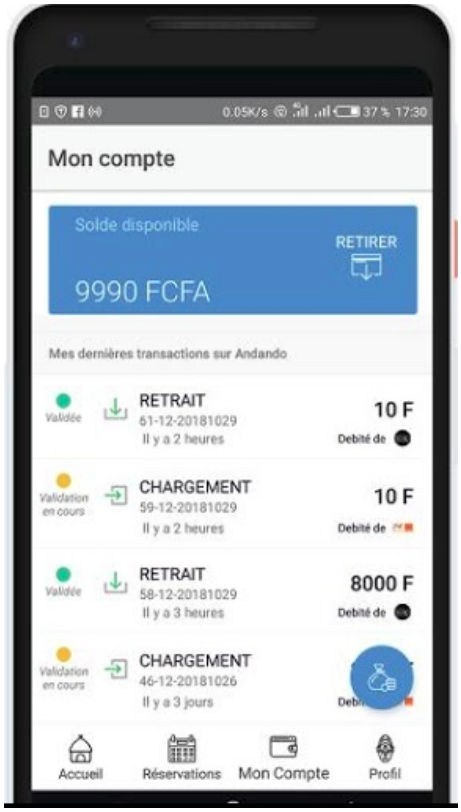


FIGURE 1.6 – Andando | Liste des trajets disponibles

1.1.2.2 Vroomiste

Déjà mis sur playstore et actuellement à **sa version 1.1** [2]. **Vroomiste** est la version mobile de l'application web vroomiste.com et proposant les mêmes services.

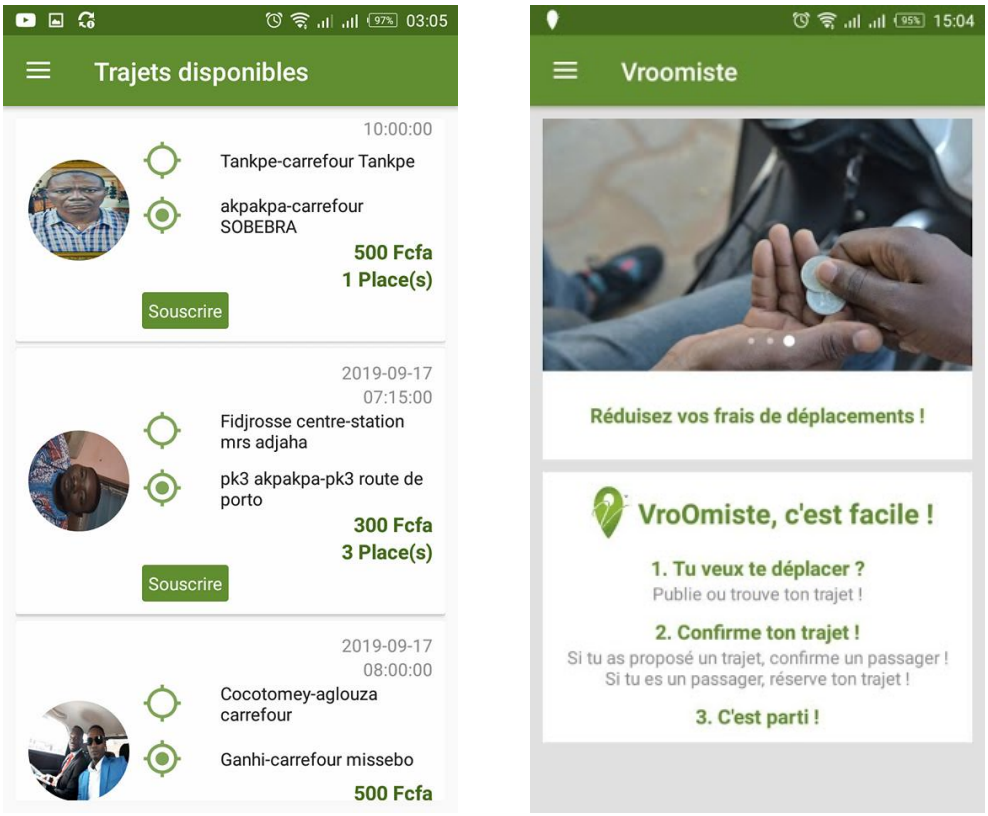


FIGURE 1.7 – Vroomiste | Accueil et trajets disponibles

[2]Source :https://play.google.com/store/apps/details?id=com.vroomiste.pathinov.vroomise&hl=en_US

1.1.2.3 Afri Ride

Afri Ride est la version mobile de l'application web afri-ride.com et proposant les mêmes services. Elle est disponible sur playstore et actuellement à **sa version 1.1** [3]

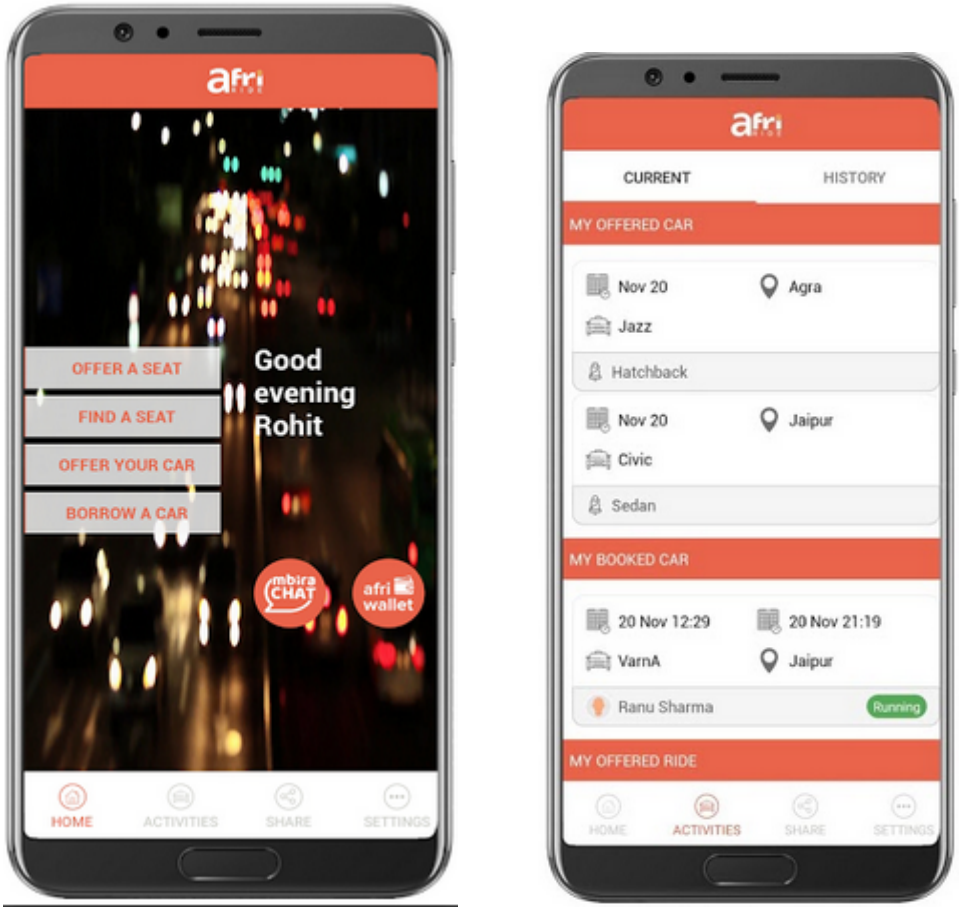


FIGURE 1.8 – Afri Ride | Accueil et trajets en cours

1.1.2.4 BlaBlacar

BlaBlaCar est une plateforme communautaire payante de covoiturage développée par la société Comuto. Avec 70 millions d'utilisateurs en 2019, BlaBlaCar est leader mondial du covoiturage.

Créée en 2004 sous le nom de covoiturage.fr, il s'agit d'une plateforme de covoiturage pour mettre en relation des conducteurs et des passagers souhaitant partager un trajet en voiture et les frais associés. Les conducteurs publient leurs places disponibles et les passagers les achètent en ligne, sur des trajets dont la distance moyenne est de 330 kilomètres.

La mise en relation entre usagers, gratuite pendant de nombreuses années, est devenue payante en 2011. Depuis, la société prélève en France une commission d'environ dix-huit à vingt et un pourcent, qui s'ajoute à la participation aux frais demandés par les conducteurs. Lorsqu'il crée son annonce, le conducteur fixe un prix par passager en fonction d'une recommandation établie par la plateforme de l'ordre de 0,065 par kilomètre. Le prix ne doit pas excéder le coût du trajet afin que le conducteur ne puisse pas en tirer un bénéfice.

[3] Source :https://play.google.com/store/apps/details?id=com.afriride&hl=en_US



FIGURE 1.9 – Logo de BlaBlacar

1.2 Limites

1.2.1 Vroomiste

Le 3 septembre 2019 un utilisateur de Vroomiste nom de **YEBE RYCH**, a laissé un commentaire sur playstore donnant son avis sur l'application. Nous citons : «Nullllllllllllllllll sa ne marche pas»[4]. Pour l'avoir tester nous avions constaté que l'authentification ne marche pas. L'application n'est disponible que sur Android

1.2.2 Afri Ride

Cette application a été lancé en décembre 2019. Il s'agit donc d'une application naissante. Pour le moment nous n'avions notifié aucun disfonctionnement technique. Cependant elle est aussi une application de tous types de trajets qui ne se concentre pas sur les trajet domicile-travail.

1.2.3 BlaBlacar

Cette application comparativement aux autres a eu le temps de s'améliorer. Elle dispose donc de plusieurs fonctionnalités.Elle se concentre essentiellement sur les trajets de longue et courte distance. Cependant elle n'est pas orientée de façon spécifique sur les trajets domicile-travail.

[4]Source :https://play.google.com/store/apps/details?id=com.vroomiste.pathinov.vroomiste&hl=fr&showAllReviews=true

1.2.4 Andando

Les utilisateurs de cette application laissent des commentaires sur sa page PlayStore[5]. Au nombre de ces commentaires nous avons :

- Le 11 octobre 2019 un utilisateur de Andando nom de **YEBE RYCH**, a laissé un commentaire sur playstore donnant son avis sur l'application. Nous citons : «Belle initiative. La plupart du temps les destinations ne se chargent pas allrs que la connexion marche.»
- Le 8 avril 2018 Un autre utilisateur du nom de **Captain Dams** a laissé un commentaire toujours sur playstore donnant son avis.Nous citons : «J'ai tenter de m'inscrire sans succès, toujours un message d'erreur qui me dit de vérifier mes informations. Que ce soit avec l'inscription via le compte Google, Facebook ou manuel. Aucun ne marche alors que toutes les informations sont correctes ! Edit 8/04 : toujours aucune nouvelle et toujours dans l'impossibilité de s'inscrire. Je déconseille.»

Aussi cette application n'aborde pas précisement les trajets domicile-travail.

1.3 Intérêt du sujet

Tenir compte des différentes limites des applications mobiles et sites web présentés ci dessus dans la réalisation de notre projet, est essentiel pour concevoir une solution répondant au besoin qu'évoque notre thème et qui est adaptée à la réalité de la population béninoise. Ainsi, l'application mobile qui sera conçue disposera aussi de sa version mobile disponible sur iOS prochainement.
Notre application permettra donc de proposer chaque semaine un trajet domicile-travail avec les jours de la semaine où ils vont au travail.En particulier elle permettra de regrouper par quartier les utilisateurs par quartier ; ce qui leur permet d'avoir une vue globale sur les covoitureurs proches d'eux. Grâce à un alogorithme nous trions parmis les trajets ceux qui correspondent à l'utilisateur et nous lui proposons. Pour toutes confirmation, suppression, refus et modification d'une demande de trajet une notification sera envoyée à tous les utilisateurs ayant un lien quelconques avec le trajet en question. Cette application dispose d'un service de messagerie qui permet aux utilisateurs de discuter entre eux des conditions de trajets.
Ayant aussi pour objectif de rendre l'expérience utilisateur meilleure, nous envisageons dans un futur proche **mettre en place un système de matching qui n'est rien d'autre qu'une intelligence artificielle** qui sera chargée de proposer de façon automatique des trajets en se basant sur les coordonnées géographiques dans le but d'apprendre des déplacements quotidiens. .

[5]Source :https://play.google.com/store/apps/details?id=dev.cityway.andando&hl=fr&showAllReviews=true

1.4 Généralité sur Android

Android est le système d'exploitation mobile créé par Google. Il équipe la majorité des téléphones portables du moment (smartphones). C'est un système à source ouverte et personnalisable, il permet de télécharger des applications (navigateur Internet, GPS, Facebook...). Android équipe également les tablettes tactiles.

1.5 Description du processus metier

Le covoiturage est une pratique où plusieurs personnes partagent un même véhicule pour se rendre à une destination commune, souvent dans le but de réduire les coûts de transport et l'impact environnemental. Il permet aux conducteurs de partager les frais de carburant et de péage, et aux passagers de bénéficier d'un trajet confortable à moindre coût. Cette pratique favorise également la réduction du trafic routier, de la pollution atmosphérique et des émissions de gaz à effet de serre. Le covoiturage peut être organisé de manière informelle entre amis, collègues ou voisins, ou bien de manière plus formelle à travers des plateformes en ligne dédiées. Les avantages du covoiturage incluent la convivialité, la possibilité de rencontrer de nouvelles personnes et de partager des expériences de voyage. Il offre également une solution de transport flexible et économique pour les personnes n'ayant pas accès à un véhicule personnel. Les utilisateurs du covoiturage peuvent planifier leurs trajets à l'avance ou trouver des opportunités de covoiturage spontanées via des applications mobiles ou des sites web. Cette pratique contribue à la diminution de la congestion routière en encourageant le partage des voitures et la réduction du nombre de véhicules sur les routes. Le covoiturage peut être utilisé pour des trajets quotidiens domicile-travail, des déplacements ponctuels ou des voyages longue distance. Il offre une alternative durable à la conduite individuelle en encourageant l'utilisation plus efficace des ressources disponibles. Les passagers peuvent contribuer aux frais de voyage en fonction de la distance parcourue ou convenir d'un partage équitable des coûts avec le conducteur. Le covoiturage peut également offrir des avantages sociaux en favorisant les interactions sociales et en renforçant les liens communautaires. Certaines entreprises encouragent le covoiturage en offrant des incitations financières ou des avantages aux employés qui partagent leur trajet. Les gouvernements et les municipalités soutiennent souvent le covoiturage en fournissant des voies réservées aux véhicules partagés et des incitations fiscales. Le covoiturage contribue à la réduction de la dépendance à l'égard des combustibles fossiles en favorisant l'utilisation plus efficace des ressources énergétiques. Il peut également améliorer l'accès aux transports dans les régions mal desservies par les transports en commun. Enfin, le covoiturage est une solution flexible et adaptable aux besoins de déplacement des individus, offrant une alternative viable à la possession individuelle de véhicules.

Cependant, cette façon de s'organiser n'existe pas au Bénin malgré tous ces avantages. C'est pour toutes ces raisons que nous proposons cette solution afin de faciliter le transport, particulièrement en zone urbaine où la circulation est suffisamment dense.

Conclusion

Dans ce chapitre nous avions présenté les solutions existantes ayant rapport à notre étude et il en ressort que ces dernières sont bien fonctionnelles. Néammoins nous tirons l'intèrêt de notre thème des limites de ceux-ci. Le prochain chapitre présente notre système à travers sa conception et les outils utilisés pour sa réalisation.

Chapitre 2

Solution et choix techniques

Introduction

La réalisation d'une application nécessite un regroupement et une hiérarchisation des idées liées à sa conception et à son développement. Sa réussite passe par le respect des contraintes de délai et de budget, ainsi que la conformité de la solution finale à la demande. Il est donc obligatoire de bien analyser le projet avant tout développement. Le présent chapitre présente la solution proposée ainsi que les différentes phases de sa conception et de sa modélisation. Il présente également un aperçu des différents outils et technologies utilisés pour la réalisation de la solution, que nous avons nommée **Wego**.



FIGURE 2.1 – Logo de Wego

2.1 Architecture du système : Firebase

Le système que nous utilisons est basé sur une architecture Firebase. Firebase est le nom d'une plateforme mobile de Google qui facilite la création de Backend à la fois scalable et performant. En d'autres termes, il s'agit d'une plateforme qui permet de développer rapidement des applications pour mobile et pour le web. Dans Firebase, vous trouverez des API intuitives regroupées dans un SDK unique. Ces API, en plus de vous faire gagner du temps, vous permettent de réduire le nombre d'intégrations que vous devez gérer par le biais de votre application. .

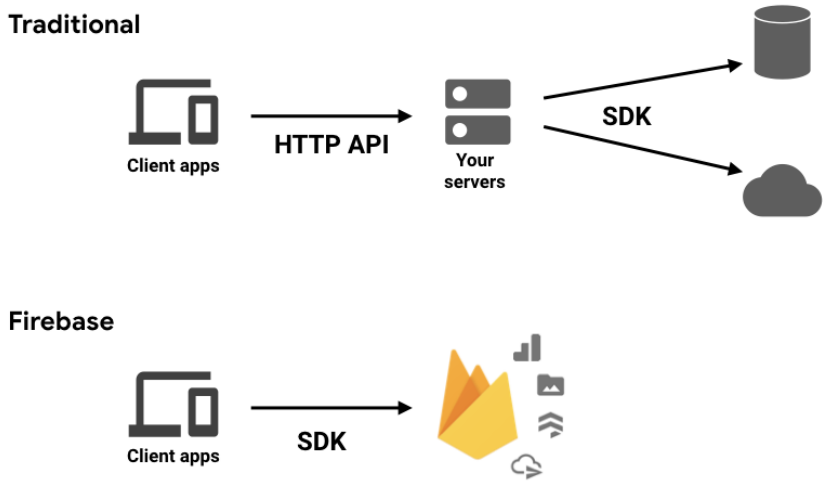


FIGURE 2.2 – Comparaison entre l'architecture traditionelle et celle de Firebase

2.1.1 Les services de Firebase

Firebase propose plusieurs services **backend** aux développeurs pour la conception d'application mobile et web. Au nombre de ces services nous avons entre autre :

2.1.1.1 Firebase database

Firebase Realtime database n'est autre qu'une base de données NoSQL bénéficiant d'un hébergement « in the Cloud » et permettant le stockage et la synchronisation de données de vos utilisateurs. Les développeurs peuvent gérer cette base de données en temps réel. Firebase Realtime database est capable de fournir à votre application la valeur des données et les mises à jour appliquées sur ces derniers avec à une simple API. Grâce à la synchronisation en temps réel, les utilisateurs de votre application peuvent consulter leurs données depuis n'importe quel terminal (sur le web ou depuis leur mobile). Notez que cette base de données est livrée avec des SDK mobiles et web et permet à cet effet la création d'applications sans utiliser des serveurs. Quand vos utilisateurs passent en mode hors ligne, les SDK de base de données en temps réel utilisent le cache pour enregistrer les modifications. Quand l'appareil est en ligne, les données locales connaissent une synchronisation automatique.

2.1.1.2 Firebase authentification

Cet outil fournit des SDK faciles à exploiter, des services backend ou encore des bibliothèques d'interface utilisateur. Ces bibliothèques vous permettent d'authentifier vos utilisateurs. En général, la configuration manuelle d'un système d'authentification prend plusieurs mois. Par la suite, il faut engager une équipe pour la maintenance. Avec Firebase, les choses se déroulent autrement. La configuration du système ne prend que quelques heures même s'il faut prendre en charge des opérations délicates comme la fusion de comptes. Plusieurs méthodes s'offrent à vous pour authentifier vos utilisateurs notamment l'exploitation de :

- Leur email et mot de passe
- Google Cloud
- Twitter
- Facebook
- Numéro de téléphone
- Etc.

2.1.1.3 Firebase Cloud Messaging (FCM)

Cette fonctionnalité vous permet de connecter vos périphériques à votre serveur dans les meilleures des conditions (fiabilité et économie de batterie). Vous pourrez alors recevoir ou envoyer gratuitement des notifications ou encore des messages sur le web ou sur vos terminaux sous iOS ou sous Android. Les messages de notifications seront limités à 2 ko tandis que les messages de données seront plafonnés à 4 ko.L'envoi de messages instantanés ou à un moment ultérieur dans le fuseau horaire de vos utilisateurs est également possible avec Firebase Cloud Messaging. Vous pouvez également procéder à l'envoi de données d'application personnalisées notamment la définition des sons, des dates d'expiration ou en encore des priorités. La fonctionnalité vous permet également de faire un suivi des évènements de conversion.

2.1.1.4 ML Kit

Cette nouvelle fonctionnalité est dédiée à l'apprentissage automatique et s'intègre facilement à votre application mobile. Vous n'aurez aucun mal à l'utiliser que vous soyez débutant ou expérimenté.

2.1.1.5 Cloud Storage

Firebase storage vous permet de partager ou encore de stocker du contenu produit par vos utilisateurs à ne citer que les images, les vidéos ou encore les fichiers audio. C'est une solution de stockage d'objets puissant qui se démarque par sa simplicité et son caractère économique.

Firebase possède également des fonctionnalités pour booster votre business à savoir :**Firebase Analytics, Remonte config, Dynamics Links** et d'autres pour optimiser la qualité de vos apps tels que : **Crashlytics, Performance monitoring, Test lab**

2.1.2 L'application mobile android

L'application mobile est accesible à tout type d'acteurs du système, il représente également un client pour notre serveur. À travers ses diverses fonctionnalités des requêtes sont lancéés depuis les téléphones android. Le back-end que nous offre firebase traite les requêtes à son niveau et renvoie les résultats issus de cette exécution. Les résultats sont transformés au format JavaScript Object Notation (JSON) et sont retournés vers l'application mobile.

2.2 Modélisation

Afin de modéliser les fonctionnalités de **Wego**, le choix a été porté sur **Unified Modeling Language (UML)**, qui est aujourd'hui un outil de communication incontournable, utilisé sur des centaines de projets de par le monde [1]. UML est un langage de modélisation graphique conçu pour fournir un ensemble de représentations standardisées afin de réaliser la conception architecturale et fonctionnelle d'un logiciel. Il est également indépendant du langage de programmation utilisé pour le développement dudit logiciel. Ainsi, les diagrammes de cas d'utilisation et de classes exposeront la modélisation de la présente application.



FIGURE 2.3 – Logo d'UML

2.2.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation regroupe tous les cas d'utilisation possibles d'un logiciel. Il est la représentation complète des fonctionnalités du système modélisé. Dans notre système il existe un seul acteur. Pour n'importe quelles actions dans le système, il est obligatoire de s'authentifier. C'est une authentification au choix. L'utilisateur à le choix de faire une authentification par email et password ou avec son compte **Google**. Ensuite il pourra avoir accès aux fonctionnalités de l'application. Ainsi l'utilisateur peut faire une demande de trajet, modifier ou supprimer un trajet qu'il a proposé, confirmer ou refuser un trajet qu'on lui a proposé. Il peut modifer sa photo de profil ou sa méthode de trajet [1]. Enfin il peut aussi envoyé un message à un autre utilisateur. La figure 2.4 illustre le diagramme de cas d'utilisation de **Wego**.

[1] Il s'agit ce ce qu'il donne en retour pour un co-trajet par exemple : 200 FCFA/Trajet ou Chacun à son tour ; ce qui signifie une alternance de voiture ou de moto

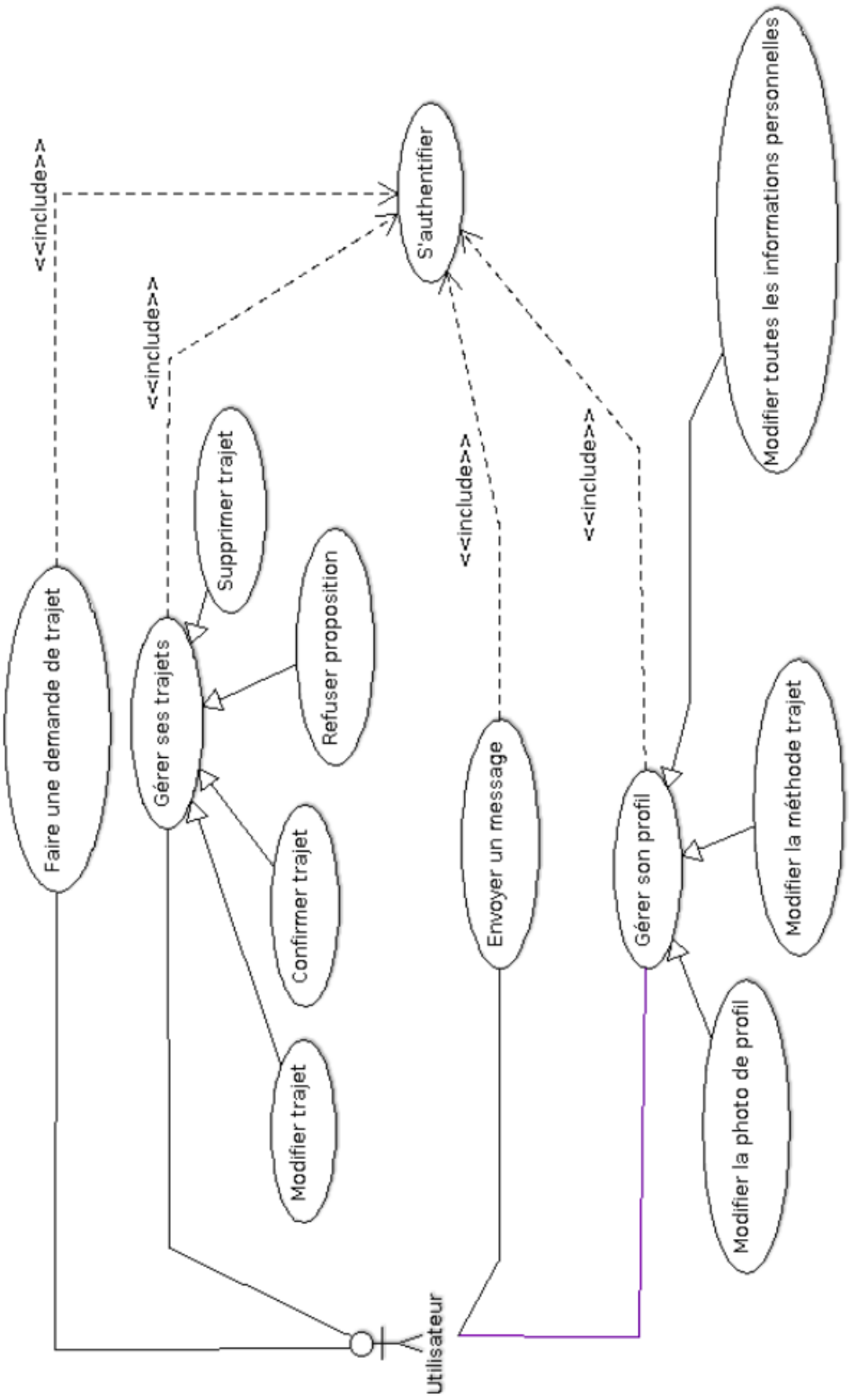


FIGURE 2.4 – Diagramme de cas d'utilisation de Wego

2.2.2 Diagramme de classes

Afin de spécifier les classes qui interviennent dans le système et quels liens elles entretiennent entre elles, nous utiliserons un diagramme de classes.La figure 2.5 présente le diagramme de classes de Wego.

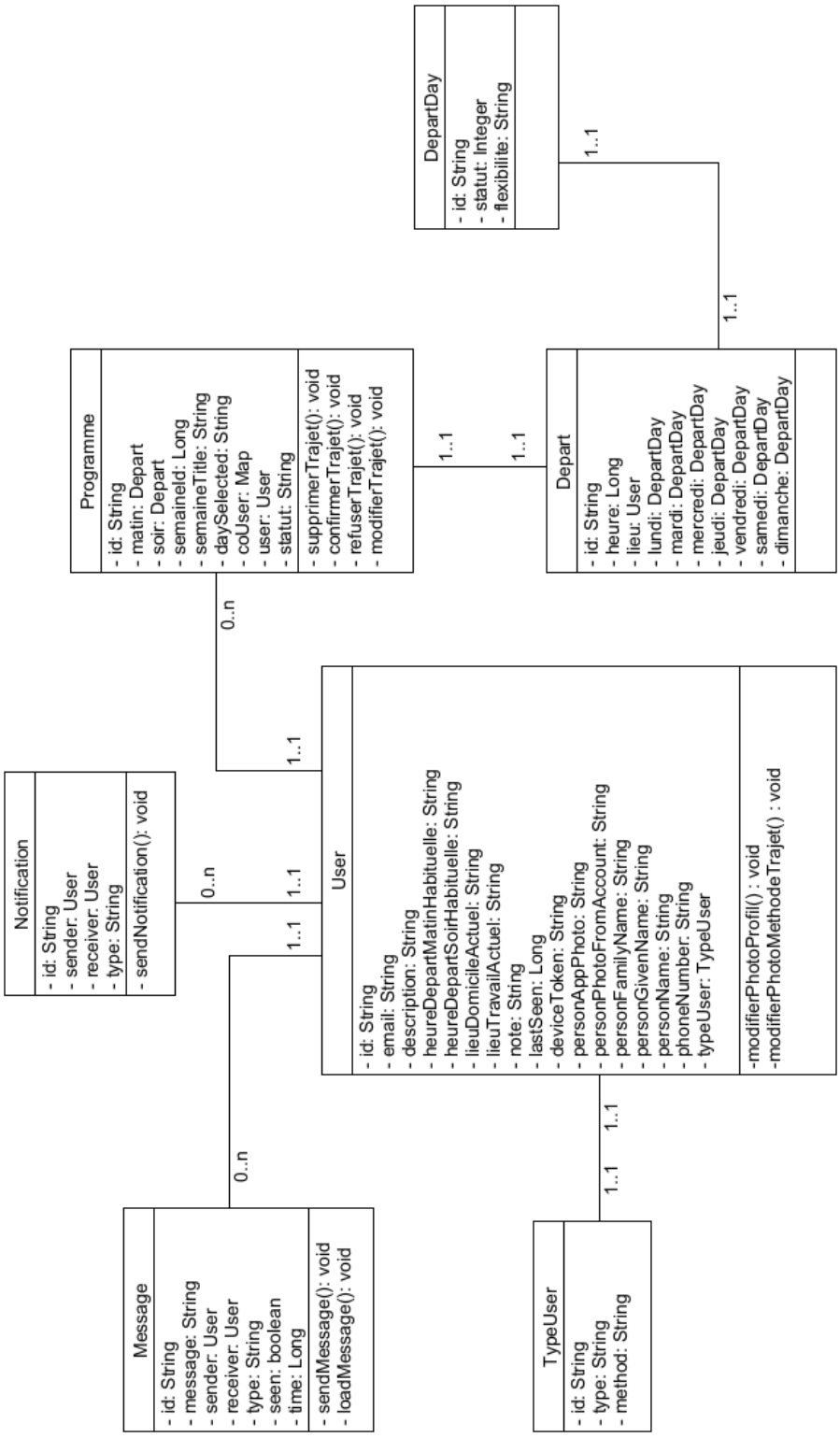


FIGURE 2.5 – Diagramme de classes de Wego

2.3 Processus de correspondance des trajets à la demande

Pour la phase correspondance des trajets à la demande nous procédons avec quatre listes de la classe **Programme** à savoir :

- **mDataForSameDaysTrajets**
  Ils s'agit des trajets de ceux qui désirent faire un trajets les mêmes jours que ceux de la demande de covoiturage. Cette liste est donc issue de la liste des trajets proposés pour le compte de la semaine à venir.

- **mDataForCurrentUserIn20HoursIntervalle**
  Cette liste est issue de la précédente. Ce sont des trajets de ceux dont les heures de départ le matin et le soir sont dans un intervalle de +/- 20 minutes respectivement des heures de départ du matin et du soir conformément à la demande de l'utilisateur. C'est donc la flexibilité du trajet qu'on désire gérer ainsi.

- **mDataForSameDomicileTravail**
  Cette liste est issue de la précédente. Ce sont des trajets de ceux qui vivent dans le même quartier et qui ont leurs lieux de travail dans le même quartier que l'utilisateur qui fait la demande.

- **mDataForDomicileTravailInSameArrondissement**
  Cette liste est issue de la liste **mDataForCurrentUserIn20HoursIntervalle**. Si par rapport à la demande de l'utilisateur, un trajet ne satisfait pas la condition d'appartenance à la liste **mDataForSameDomicileTravail** mais que les lieux de domicile et de travail se retrouvent respectivement dans les mêmes arrondisments alors on ajoute le trajet correspondant dans la liste **mDataForDomicileTravailInSameArrondissement** dont il s'agit ici.

Enfin on cummule tous les trajets des quatre listes ci-dessus et on élimine tous les doublons possibles. Si l'utilisateur est un passager on lui affiche que des conducteurs qui proposent un tarif de trajet mais s'il est conducteurs on lui affiche et la liste des conducteur mais aussi celle des passagers. C'est ainsi que nous obtenons une liste de trajets qui correspondent à la demande de l'utilisateur.

2.4 Motivation du choix techniques

2.4.1 Méthodologie de développement

La conduite du projet s'est faite en se conformant à la **méthodologie agile**, une approche de développement itératif qui permet d'observer concrètement l'évolution du projet. La notion même de "gestion de projet" est remise en question au profit de "gestion de produit", de façon à raisonner davantage "produit" que "projet" [3]. Le choix a été porté sur **Scrum** qui est une méthode agile où le développement est découpé en plusieurs phases courtes appelées itérations ou **sprint**. L'ensemble des fonctionnalités du sprint (habituellement 2-4 semaines) viennent du "backlog" du projet, qui est un ensemble priorisé d'exigences importantes à achever [4]. Chaque itération inclut des travaux de conception, de spécification fonctionnelle et technique quand c'est nécessaire, de développement et de test [3]. Cette méthode permet de mieux s'adapter aux imprévus pouvant subvenir pendant le projet et facilite la livraison d'un produit de qualité dans un temps planifié.

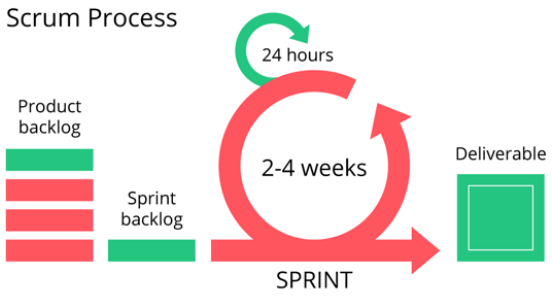


FIGURE 2.6 – Méthode Scrum [**?**]

2.4.2 Outils et technologies utilisées

La réalisation de notre solution a nécessité l'utilisation de plusieurs technologies. Elles seront regroupées en deux groupes : le Back-end et le Front-end.

2.4.2.1 Motivation ud choix de technologie Back-end

Le Back-end représente tout ce que l'utilisateur ne voit pas relativement à l'application. Il s'agit des traitements logiques et actions effectués pour pouvoir conserver, fournir, ou modifier les données au sein de l'application. Les technologies que nous avons utilisées à cet effet sont uniquement celles que Firebase nous offre pour la gestions d'un back-end solide. Plus haut nous avons défini la plupart de ces services.
Celles que dont nous sommes servi dans ce projet sont **Firebase database, Firebase authentification et Firebase Cloud Messaging** Dans le tableau suivant nous faisons une étude comparative justifiant le choix ci dessus.

	Firebase (et ses outils)	PostgreSQL
Avantages		
Facilité de configuration	Facile à configurer avec des outils de gestion prêts à l'emploi	Configuration flexible avec de nombreuses options de personnalisation
Temps de développement	Développement rapide grâce à une configuration simple et à des SDK bien documentés	Peut nécessiter plus de temps de configuration initiale et de développement
Mises à jour	Gestion automatique des mises à jour et de l'évolutivité par Firebase	Gestion manuelle des mises à jour et de l'évolutivité
Scalabilité	Évolutivité automatique en fonction de la charge avec des frais supplémentaires si nécessaire	Évolutivité contrôlée par l'utilisateur avec des options de mise à l'échelle flexibles
Intégration des services Google	Intégration transparente avec d'autres services Google tels que Google Analytics, Google Cloud Messaging, etc.	Peut nécessiter des efforts supplémentaires pour intégrer d'autres services avec PostgreSQL
Inconvénients		
Dépendance aux services Google	La dépendance à une seule plateforme peut être considérée comme un risque	Indépendance vis-à-vis de toute plateforme tierce
Coût	Peut être coûteux à grande échelle en raison des frais basés sur l'utilisation	Les coûts initiaux et de maintenance peuvent être plus élevés pour une configuration personnalisée
Contrôle sur l'infrastructure	Moins de contrôle sur l'infrastructure sous-jacente	Contrôle total sur l'infrastructure et la gestion des données
Sécurité	Les données sont stockées sur les serveurs de Google avec des mesures de sécurité intégrées	La sécurité dépend de la configuration et des pratiques de sécurité mises en œuvre par l'utilisateur

TABLE 2.1 – Comparaison entre Firebase (et ses outils) et PostgreSQL

2.4.2.2 Motivation ud choix de technologie Front-end

Le Front-end, par opposition au Back-end fait référence aux éléments que l'on voit à l'écran, avec lesquels on peut interagir ; il désigne la partie visuelle de l'application. Il s'agit d'éléments comme les images, les fenêtres, les boutons, les menus déroulants, etc. Les technologies que nous avons utilisées à cette fin sont :

- **Java**
  C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il est conçu pour permettre aux développeurs d'applications d'écrire une fois, d'exécuter n'importe où Write Once, Run Anywhere (WORA). Ce qui signifie que le code Java compilé peut être exécuté sur toutes les platesformes prenant en charge Java sans nécessiter de recompilation. WORA est réalisé en compilant un programme Java dans un langage intermédiaire appelé bytecode. Le format du bytecode est indépendant de la Une machine virtuelle, appelée Java Virtual Machine, est utilisée pour exécuter le sur chaque [8]. Ce langage nous a été utile pour le développement

de l'application mobile.

- **XML**
  eXtensible Markup Language (XML) (qui signifie : langage de balisage extensible) est un langage de balisage de description de données. XML peut être considéré comme un metalangage permettant de définir d'autres langages, c'est-à-dire définir de nouvelles balises permettant de décrire la présentation d'un texte [9]. Les interfaces de notre application mobile on été rendues possibles grâce à ce langage.

Dans le tableau suivant nous faisons une étude comparative justifiant le choix ci dessus.

	Android natif (Java)	**Flutter**
Avantages		
Performance	Performances élevées grâce à l'exécution native sur Android	Bonnes performances grâce à la compilation en code natif
Intégration	Intégration étroite avec les fonctionnalités natives d'Android	Accès facile aux fonctionnalités natives via des plugins
Communauté	Grande communauté de développeurs et de ressources	Communauté en croissance rapide avec un support actif
Interface utilisateur	Contrôle total sur l'apparence et le comportement de l'interface utilisateur	Interface utilisateur fluide grâce au framework Material Design de Google
Inconvénients		
Développement multiplateforme	Nécessite de développer des applications distinctes pour iOS	Permet de créer des applications pour iOS et Android à partir d'un seul codebase
Courbe d'apprentissage	Peut être plus longue en raison de la complexité de l'écosystème Android	Courbe d'apprentissage plus courte grâce à sa simplicité et à sa cohérence
Maintenance	Les mises à jour peuvent être plus fastidieuses en raison de la fragmentation d'Android	Les mises à jour sont plus faciles à gérer grâce à un seul codebase
Performances sur iOS	Nécessite des compétences supplémentaires pour le développement sur iOS	Peut présenter des performances légèrement inférieures sur iOS en raison de l'interprétation du code Dart

TABLE 2.2 – Comparaison entre Android natif (Java) et Flutter

Conclusion

Ce chapitre a présenté notre solution à travers sa modélisation, son fonctionnement et les choix techniques liés à sa réalisation. Le prochain chapitre fournira les résultats obtenus et quelques critiques constructives.

Chapitre 3

Résultats et discussion

Introduction

Nous présentons tout au long de ce chapitre, les résultats de notre travail à travers les différentes interfaces de notre application mobile ainsi que ses limites. A la fin de ce chapitre, les objectifs fixés doivent avoir été atteints et le système proposé doit être prêt pour être exploité par les utilisateurs finaux.

3.1 Interfaces des applications

3.1.1 Interface d'acceuil de l'application

La première interface qui s'affiche à l'utilisateur après sa connexion est cruciale pour orienter son expérience. Dotée d'une barre de navigation intuitive, elle offre à l'utilisateur la possibilité de naviguer facilement entre les différentes fonctionnalités de l'application. Des menus courants sont également disponibles pour simplifier la navigation et rendre l'accès aux fonctionnalités clés plus rapide et plus pratique. De plus, un bouton bien en évidence permet à l'utilisateur de créer facilement un nouveau trajet, offrant ainsi une expérience fluide et efficace dès le départ. Cette interface constitue la porte d'entrée vers toutes les fonctionnalités de l'application, et son design réfléchi vise à garantir une expérience utilisateur agréable et intuitive.



FIGURE 3.1 – Accueil de l'application

3.1.2 Interface de demande de co-trajet

Au niveau de cette interface l'utilisateur recherche un co-trajet en renseignant le quartier de son domicile, celui de son lieu de travail, son heure de départ le matin et son heure de retour le soir. Il renseigne aussi les jours de la semaine au il désire voyager. Il faut préciser que la recherche d'un trajet pour une semaine donnée se fait pendant la semaine précédente. Par exemple sur cet exemple il s'agit d'un trajet du quartier Zogbo au quartier Missite. Le depart pour le travail le matin est prévu pour 07h10 et le depart pour rentrer au domicile le soir est pour 18h30. Ensuite il faut selectionner enfin les jours de la semaine ou on veut planifier un co-trajet puis enfin valider cette demande avec le boutton "VALIDER".

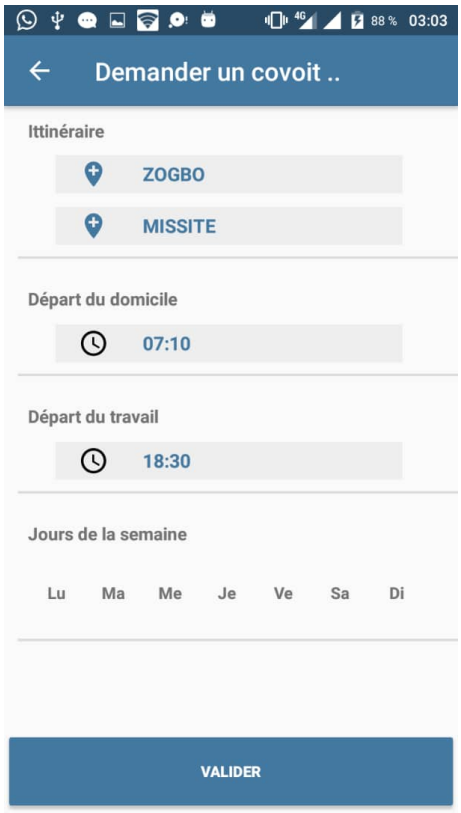


FIGURE 3.2 – Demande de co-trajet

3.1.3 Interface de visualisation des résultats de demande de co-trajets

Les résultats de la demande de trajet sont affichés sur cette interface, basés sur notre processus de correspondance de trajets. Tout d'abord, la demande de covoiturage est présentée en haut de la page, suivie de la liste des résultats trouvés, c'est-à-dire des utilisateurs de l'application correspondant à la demande de l'utilisateur. En appuyant sur le bouton en fond bleu "Proposer un trajet à...", le co-voiturage est validé, permettant ainsi à ces deux utilisateurs de se déplacer ensemble aux jours et aux heures convenus. Cette présentation des résultats de recherche de trajet offre une visualisation claire et structurée des options disponibles, facilitant ainsi la prise de décision pour les utilisateurs. La visualisation de la demande de co-trajet en tête de liste met en évidence l'importance de la correspondance des besoins des utilisateurs et des conducteurs potentiels. L'accès rapide au bouton de proposition de trajet simplifie le processus de validation du covoiturage, favorisant ainsi une expérience utilisateur fluide et efficace.

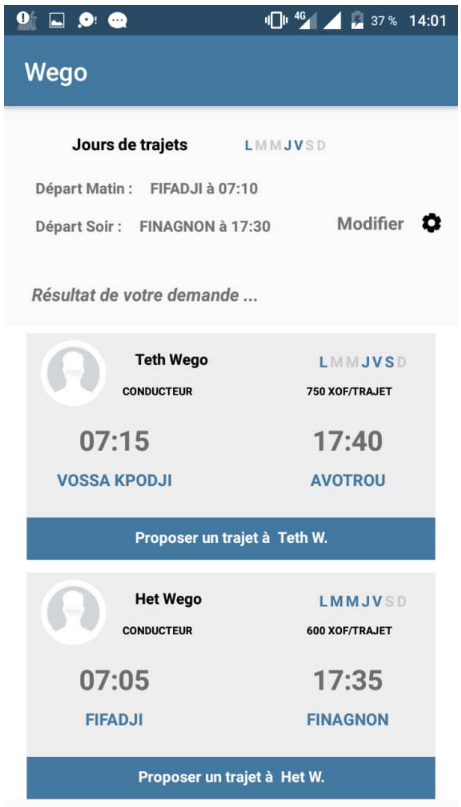


FIGURE 3.3 – Résultats de recherche de trajet

3.1.4 Interface de gestion des trajets

3.1.4.1 Interface des trajets en cours

Cette interface présente les trajets confirmés par l'utilisateur, indiquant ainsi les trajets planifiés avec les utilisateurs correspondants à la date convenue. L'utilisateur a la possibilité de gérer ces trajets directement depuis cette interface. Il peut mettre fin à un trajet, communiquer avec le covoitureur ou consulter les détails relatifs à ce dernier. Les trajets en cours sont facilement identifiables grâce à un fond vert distinctif, offrant ainsi une vue claire et rapide des trajets actifs. La figure suivante correspondante à cette interface illustre de manière visuelle la liste des trajets confirmés et/ou en cours, fournissant ainsi une référence visuelle pour l'utilisateur. Cette interface centralisée permet à l'utilisateur de suivre et de gérer efficacement ses trajets planifiés, garantissant ainsi une expérience de covoiturage fluide et organisée.

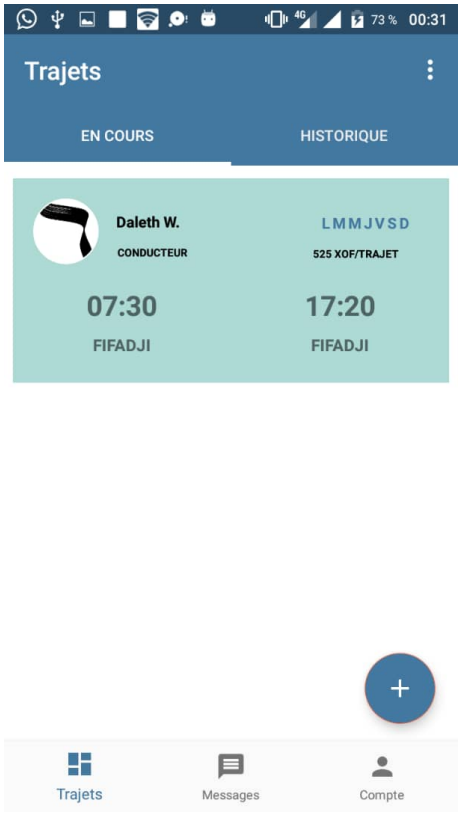


FIGURE 3.4 – Liste des trajets confirmés et /ou commencés

3.1.4.2 Interface de l'historique des trajets

Cette interface présente une liste des trajets précédemment effectués par l'utilisateur, offrant ainsi un historique de ses voyages passés. L'utilisateur a la possibilité d'interagir avec les covoitureurs associés à ces trajets, que ce soit pour communiquer avec eux ou pour consulter des détails spécifiques. Les trajets terminés sont clairement indiqués par un fond rouge distinctif, ce qui permet à l'utilisateur de repérer facilement les voyages passés. Cette interface est conçue de manière ergonomique, avec une disposition simple et intuitive. Les trajets sont organisés par semaine, facilitant ainsi la recherche et la gestion des voyages antérieurs. Cette approche permet à l'utilisateur de consulter rapidement son historique de trajets et de retracer ses expériences passées de covoiturage.

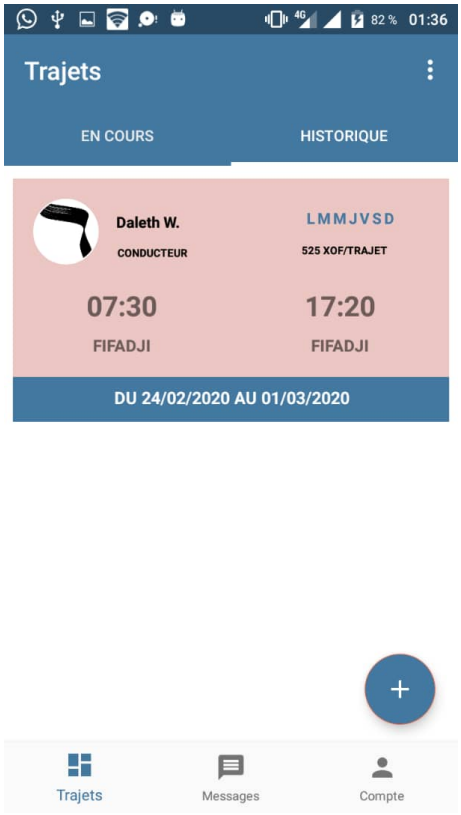


FIGURE 3.5 – Historique des trajets

3.1.5 Interface de visualisation des propositions de co-trajets

Cette interface présente à l'utilisateur la liste des trajets proposés pour la semaine à venir, offrant ainsi une vue d'ensemble des possibilités de covoiturage disponibles. L'utilisateur est invité à accepter un seul trajet parmi ceux proposés. De plus, il a la possibilité de refuser un trajet, d'examiner les détails de la proposition ou de communiquer directement avec la personne qui fait la proposition. Cette interface est conçue de manière ergonomique, avec une disposition simple et intuitive. Les trajets sont organisés et classés par semaine, ce qui facilite la gestion des propositions et permet à l'utilisateur de prendre des décisions éclairées concernant ses déplacements à venir.



FIGURE 3.6 – Propositions faites à l'utilisateur

3.1.6 Interface de visualisation des demandes de co-trajets l'utilisateur

Cette interface permet à l'utilisateur d'accéder aux trajets qu'il a proposés, avec la possibilité de les modifier ou de les supprimer selon ses besoins. De plus, l'utilisateur peut consulter les résultats de ses demandes de covoiturage au cas où aucune correspondance n'aurait encore été trouvée. La figure 3.7 correspondante à cette interface offre une vue d'ensemble de toutes les recherches de co-trajet effectuées par l'utilisateur, facilitant ainsi le suivi et la gestion de ses demandes de covoiturage. Cette interface est conçue de manière ergonomique, offrant une navigation simple et intuitive pour une expérience utilisateur fluide et efficace.

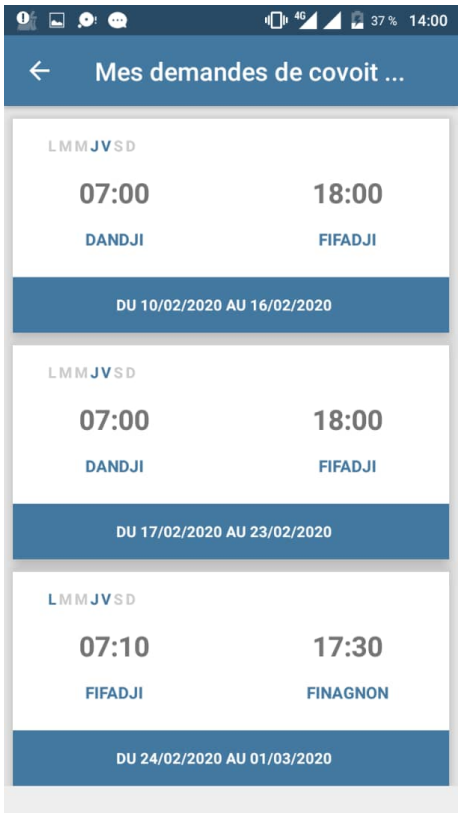


FIGURE 3.7 – Toutes les recherches de co-trajet de l'utilisateur

3.2 Limites

- La nécéssité pour les utilisateurs de pouvoir préciser le nombre de place disponible à son niveau. Pour le moment l'application fixe ce nombre à 1.
- Le processus de correspondance de trajets prend en compte les facteurs individuels tels que (le quartier de résidence, celui du lieu de travail puis les heures de départ du matin et du soir), Renseigner ces informations chaque fois peut être trop demander pour certains utilisateurs. **C'est pourquoi nous prévoyons dans la suite d'écrire une intelligence artificielle pour apprendre des déplacements des utilisateurs et ainsi leur proposer de façon automatique les trajets utilisateurs qui font les mêmes chemins.**
- Disponible uniquement sur android,
- Permettre à l'utilisateur de préciser lui même la flexibilité en ce qui concerne ses heures de départ. Pour le moment l'application fixe celle flexibilté à +/- 20 minutes.
- Améliorer la messagerire car elle est en temps réelle mais le chargement des messages est pas encore parfait. Une optimisation du code s'impose alors.
- Améliorer la gestion des écouteurs d'évènements notament la suppression des "listener" en ce qui concerne la méthode **void removeEventListener(ValueEventListener listener)**. Le problème ne se posant pas seulement à notre niveau car d'autres personnes en parlent et pensent qu'elle sera mise à jour par Google.
- Reécrire le code suivant l'architecture Model View Viewmodel (MVVM) pour améliorer l'expérience utilisateur et rendre le code plus lisible.

3.3 Inconvenients liés à la solution

Le co-voiturage reste un sujet complexe malgré la solution que nous y apportons. Au nombre des problème qu'il soulève nous avons :

- Un impact direct sur l'économie car plus il y a de covoiturage, moins il y a d'achat de carburant et c'est l'état qui voit la Taxe sur la Valeur Ajoutée (TVA) diminuer,
- Le retard dans les administrations publiques et privées qui à cause du possible retard d'un passager ou même du conducteur,
- La sécurité des covoitureurs eux même car un incident peut survenir

Conclusion

Ce chapitre a fait l'objet d'un exposé des résultats obtenus à travers la présentation des interfaces des différentes applications. Les insuffisances liées à la solution ont été soulevées et seront utilisées comme base pour formuler les perspectives.

Conclusion générale et perspectives

Le nombre de smarthphones en Afrique est en pleine expansion. La mise en relation des personnes et la création de communautés sont des choses que cette croissance facilite.

Nous avons proposé une application mobile capable de mettre en relation des personnes ayant un objectif partagé qu'est le transport de façon économique. Nous la voyons donc non pas comme la solution au transport urbain mais comme une approche de solution. La dite application a été developpée en Android natif Java.

Des perspectives intéressantes sont envisagées pour ce travail. Par exemple, dans la vision d'un développement plus abouti, il serait interressant d'écrire une intelligence artificielle pour apprendre des déplacements des utilisateurs et ainsi leur proposer de façon automatique les trajets utilisateurs qui font les mêmes chemins. Cette façon de procéder améliorera l'expérience utilisateur

Bibliographie

[1] Pascal Roques, *UML 2 par la pratique*, Eyrolles, 2011.

[2] Cyrille Herby, *Apprenez à programmer en Java*, OpenClassrooms - ex-Site du Zéro, 2015.

Webographie

[3] L'Agiliste, *Introduction aux méthodes agiles et Scrum* , https://agiliste.fr/introduction-methodes-agile14, consulté le 17 Mars 2019.

[4] Supinfo, *Méthodologies de développement de logiciel*, https://www.supinfo.com/articles/single/6765-methodologies-developpement-logiciel, consulté le 17 Mars 2019.

[5] Firebase, *Documentation | Firebase*, https://firebase.google.com/docs, consulté le 1er Juin 2019.

[6] Youdi Raymondo, *Applications et covoiturage en afrique de l'ouest*, http://cafe.tg/applications-et-covoiturage-en-afrique/, consulté le 21 Septembre 2019.

[7] Wikipedia, *BlaBlaCar*, https://fr.wikipedia.org/wiki/BlaBlaCar, consulté le 21 Septembre 2019.

[8] HowToDoInJAVA, *What is Java programming language ?*, https://howtodoinjava.com/java/basics/what-is-java-programming-language/, consulté le 26 septembre 2019.

[9] Comment C aMarche, *XML - Introduction à XML*, https://www.commentcamarche.net/contents/1332-xml-introduction-a-xml, consulté le 26 septembre 2019.

[10] TVAC Studio, *Firebase Chat App - Android Studio | One to One Chat Application*, https://www.youtube.com/results?search_query=lapit+chat, consulté le 12 Décembre 2019.

[11] INSAE, *CAHIER DES VILLAGES ET QUARTIERS DE VILLE DU DEPARTEMENT DU LITTORAL (RGPH-4, 2013)*, https://www.insae-bj.org/images/docs/insae-statistiques/enquetes-recensements/RGPH/1.RGPH_4/resultats20finaux/Cahiers20villages/Cahier20des20villages20et20quartiers20de20ville20Littoral.pdf, consulté le 31 Décembre 2019.