###### 

# LISTE DES ABREVIATIONS

**A**

AEE Agriculture Et Élevage

AGL Atelier de Génie Logiciel

AMI Appel à manifestation d’intérêt

API Application Programming Interface

**C**

CAA Commerce et Administration

CAE Centre d’affaires Eden

**D**

DAF Direction Administrative et Financière

DEP Direction des Études ET Départements

DG Directeur General

DPI Département de Production Informatique

DTJA Droits et Techniques Juridiques des Affaires

**E**

EMII Électro Mécanique et Informatique Industrielle

EMP Économie et Management de Projet

ESIIA Electronics Systèmes Informatiques et intelligence Artificielle

ESSTIM École Supérieure pour la Science et la Technologie de l’Informatique à Madagascar

**F**

FID Fonds d’Intervention pour le Développement

**G**

GCA Génie Civil et Architecture

GRH Gestion des Ressources Humaines

**I**

IAA Industrie Agro-Alimentaire

ICMP Industries Chimiques, Ministères et Pétrolières

IDE Integrated Development Environment

IGGLIA Informatique de Gestion, Génie Logiciel et Intelligence

Artificielle

IMTICIA Informatique, Multimédia, Technologies de l’Informations, de la Communication et Intelligence Artificielle

ISAIA Informatique, Statistiques Appliquées et

Artificielle Intelligence

ISPM Institut Supérieur Polytechnique de Madagascar

**L**

LMD Licence, Masters, Doctorat

**M**

MCD Modèle Conceptuel des Données

MLD Modèle Logique des Données

**N**

NIF Numéro d’Identification Fiscale

**O**

OS Operating System

**P**

PIP Pharmacologie et Industries Pharmaceutiques

PSM Persistent Stored Modules

**R**

RH Ressources Humaines

**S**

SARL Société A responsabilité Limitée

SG Secrétariat General

SGBDR Système de Gestion de Base de Données Relationnelles

SQL Structured Query Langage

STAT Numéro statistique

**T**

TEE Tourisme Et Environnement

TEH Tourisme et Hôtellerie

TTC Tout Taxe compris

TVA Taxe sur la Valeur Ajoutée

**U**

UML

**V**

VB Visual Basic

VS Visual studio

**X**

XML eXtensible Markup Langauage

# LISTE DES FIGURES

[Figure 1. Page d'accueil du site web de l'ISPM 4](#_Toc9399844)

[Figure 2. Logo de l'ISPM 4](#_Toc9399845)

[Figure 3. Cursus d'étude à l'ISPM 8](#_Toc9399846)

[Figure 4. Organigramme à l'ISPM 9](#_Toc9399847)

[Figure 5. Logo EQIMA 12](#_Toc9399848)

[Figure 6. Site web de l’EQIMA 13](#_Toc9399849)

[Figure 7. Organigramme de l’EQIMA 14](#_Toc9399850)

[Figure 8. Représentation UML d'une classe 17](#_Toc9399851)

[Figure 9. Encapsulation 18](#_Toc9399852)

[Figure 10. Association 19](#_Toc9399853)

[Figure 11. Agrégation 19](#_Toc9399854)

[Figure 12. Héritage 20](#_Toc9399855)

[Figure 13. Exemple de relation de dépendance 20](#_Toc9399856)

[Figure 14. Exemple de diagramme mettant en œuvre une interface. 21](#_Toc9399857)

[Figure 15. Exemple de diagramme de classes et de diagramme d'objets associé 22](#_Toc9399858)

[Figure 16. Représentation d'un composant et de ses interfaces requises ou offertes sous la forme d'un classeur structuré stéréotypé « component ». 22](#_Toc9399859)

[Figure 17. Représentation d’un paquetage 23](#_Toc9399860)

[Figure 18. Exemple d'état composite modélisant l'association d'une commande à un client. 23](#_Toc9399861)

[Figure 19. Exemple de représentation d’un acteur 24](#_Toc9399862)

[Figure 20. Exemple de représentation d'un acteur sous la forme d'un classeur 24](#_Toc9399863)

[Figure 21. Exemple de représentation d'un cas d'utilisation 25](#_Toc9399864)

[Figure 22. Exemple de diagramme de cas d'utilisation modélisant une borne d'accès à une banque. 25](#_Toc9399865)

[Figure 23. Exemple de diagramme de cas d'utilisation 26](#_Toc9399866)

[Figure 24. Relations entre cas pour décomposer un cas complexe 27](#_Toc9399867)

[Figure 25. Représentation graphique des nœuds d'activité 28](#_Toc9399868)

[Figure 26. Exemple de diagramme d'activités modélisant le fonctionnement d'une borne bancaire 29](#_Toc9399869)

[Figure 27. Un diagramme d'états-transitions simple 30](#_Toc9399870)

[Figure 28. Exemple d’état simple 30](#_Toc9399871)

[Figure 29. Représentation graphique de l’état initial 31](#_Toc9399872)

[Figure 30. Représentation graphique de l'état final 31](#_Toc9399873)

[Figure 31. Exemple d’utilisation avec point de décision 32](#_Toc9399874)

[Figure 32. Représentation d’un système asynchrone 33](#_Toc9399875)

[Figure 33. Représentation d’un message synchrone 33](#_Toc9399876)

[Figure 34. Représentation d’un message de création et de destruction d’instance 34](#_Toc9399877)

[Figure 35. Syntaxe des messages et des réponses 35](#_Toc9399878)

[Figure 36. Vue d’ensemble de la méthode SCRUM 36](#_Toc9399879)

[Figure 37. Logo IONIC 40](#_Toc9399880)

[Figure 38. Logo HTML 5 46](#_Toc9399881)

[Figure 39. Structure et syntaxe Html 47](#_Toc9399882)

[Figure 40. Logo typescript 48](#_Toc9399883)

[Figure 41. Composants de typescript 49](#_Toc9399884)

[Figure 42. Logo SCSS 50](#_Toc9399885)

[Figure 43. Logo SASS 50](#_Toc9399886)

[Figure 44. Logo Firebase 51](#_Toc9399887)

[Figure 45. Les services de Firebase 52](#_Toc9399888)

[Figure 46. Représentation du service MAPS 54](#_Toc9399889)

[Figure 47. Représentation du service ROUTES 54](#_Toc9399890)

[Figure 48. Représentation du service PLACES 55](#_Toc9399891)

[Figure 49. Logo d’Adobe Photoshop 55](#_Toc9399892)

[Figure 50. Logo Trello 56](#_Toc9399893)

[Figure 51. Logo git 56](#_Toc9399894)

[Figure 52. Interface GIT BLIT 59](#_Toc9399895)

[Figure 53. Page d’accueil de l’application pour les utilisateurs 58](#_Toc9399896)

[Figure 54. Carte et marqueur de position 59](#_Toc9399897)

[Figure 55. Formulaire de recherche 60](#_Toc9399898)

[Figure 56. Traçage de la route sur la carte 61](#_Toc9399899)

[Figure 57. Suivi des bus 62](#_Toc9399900)

[Figure 58. Démarrage de l’application dédiée aux chauffeurs 63](#_Toc9399901)

[Figure 59. Application en marche 64](#_Toc9399902)

[Figure 60. Code source du fichier app.component.ts 65](#_Toc9399903)

[Figure 61. Code source du fichier app.html 65](#_Toc9399904)

[Figure 62. Extrait de code du fichier app.module.ts 66](#_Toc9399905)

[Figure 63. Extrait de code du fichier app.scss 66](#_Toc9399906)

[Figure 64. Extrait de code du fichier map.html 67](#_Toc9399907)

[Figure 65. Extrait de code du fichier map.module.ts 67](#_Toc9399908)

[Figure 66. Extrait de code du map.scss 68](#_Toc9399909)

[Figure 67. Extrait de code du fichier address.class.ts 68](#_Toc9399910)

[Figure 68. Extrait de code du fichier coordinate.class.ts 69](#_Toc9399911)

[Figure 69. Extrait de code du fichier google-geolocation.class.ts 69](#_Toc9399912)

[Figure 70. Extrait de code du fichier config.xml 70](#_Toc9399913)

[Figure 71. Extrait de code du fichier config.json 70](#_Toc9399914)

[Figure 72. Code source du fichier tsconfig.json 71](#_Toc9399915)

[Figure 73. Diagramme de classes 73](#_Toc9399916)

[Figure 74. Diagramme d’activité 74](#_Toc9399917)

[Figure 75. Diagramme de cas d’utilisation 75](#_Toc9399918)

[Figure 76. Diagramme de déploiement 75](#_Toc9399919)

# LISTE DES TABLEAUX

[*Tableau 1.* *Les prix obtenus par l’ISPM* 11](#_Toc9399920)

[*Tableau 2.* Le product Backlog 37](#_Toc9399921)

# SOMMAIRE

[REMERCIEMENTS 2](#_Toc9325930)

[AVANT PROPOS 3](#_Toc9325931)

[SOMMAIRE 4](#_Toc9325932)

[LISTE DES FIGURES 5](#_Toc9325933)

[LISTE DES TABLEAUX 6](#_Toc9325934)

[LISTE DES ABREVIATIONS 7](#_Toc9325935)

[INTRODUCTION 11](#_Toc9325936)

[PARTIE 1 14](#_Toc9325937)

[PRESENTATION GENERALE 14](#_Toc9325938)

[CHAPITRE 1 : PRESENTATION DE L’ISPM 15](#_Toc9325939)

[A. HISTORIQUE 15](#_Toc9325940)

[B. PRESENTATION 16](#_Toc9325941)

# REMERCIEMENTS

C’est avec un grand plaisir que nous réservons ces lignes en signe de gratitude et de reconnaissance à tous ceux qui ont contribué à la réalisation de ce mémoire.

Tout d’abord, nous rendons grâce à DIEU TOUT PUISSANT de nous avoir donné la force, courage et la santé durant nos études et notre stage, et lors de la réalisation de ce mémoire.

Ensuite, nous tenons aussi à remercier vivement :

* Monsieur le Professeur RABOANARY Julien Amédée, Recteur de l’Institut Supérieur Polytechnique de Madagascar de nous avoir accepté comme étudiants au sein de son Institut;
* Monsieur RAKOTOMANIRAKA Eric, le Président Général de l’EQIMA, et tout le personnel de sa société ;
* Monsieur ANDRIAMANANTENASOA Mirana, Lead informatique, pour son encadrement et sa collaboration durant le stage ;
* Monsieur RABOANARY Toky Hajatina, Enseignant-Chercheur à l’ISPM, notre encadreur pédagogique, pour ses remarques constructives durant ce travail de mémoire ;
* Tous les enseignants et tous les personnels de l’Institut Supérieur Polytechnique de Madagascar.

Nous remercions aussi nos chers parents et toute notre famille, qui nous ont soutenus.

# AVANT PROPOS

L’Institut Supérieur Polytechnique de Madagascar (ISPM) est une institut supérieur polytechnique qui suit le système éducatif LMD (Licence Master Doctorat) afin de suivre la norme imposée par le Ministère de l’Enseignement et de la Recherche Scientifique pour renforcer les travaux de recherche des étudiants.

Le présent mémoire rentre dans le cadre de l'obtention du diplôme de fin d’études du second cycle en Informatique et Télécommunications, parcours : Informatique de Gestion, Génie Logiciel et Intelligence Artificielle. Pour cela, nous avons effectués notre stage au sein de la société EQIMA qui a pour thème « CONCEPTION DE L’APPLICATION MOBILE PIKLA ».

Ce stage a apporté un large élargissement à nos connaissances et à nos expériences sur le plan pratique et sur le plan professionnel.

# INTRODUCTION

Depuis l’apparition de l’internet, la façon de penser et de vivre dans le monde a été révolutionnés, elle a permis aux consommateurs de faire des transactions, et d’accomplir leurs tâches sans devoir se déplacer physiquement.

Une dizaine d’années après, cette innovation est suivie par l’apparition de la technologie mobile qui a pris une place importante dans notre société, les assistants personnels, téléphones cellulaires, smartphones, tablettes, etc.

Les moyens de connexion, comme les réseaux sans fil (Wifi, GPRS et d’autres) ont permis de suivre et

D’accéder aux informations dont nous aurons besoin partout où il y a une couverture réseau et cela se fait à l’aide d’applications mobiles.

Comme étant dans le domaine de l’informatique, nous contribuons aussi à cette activité. Nous avons créé au sein de l’EQIMA une application mobile intitulée « PIKLA » qui permet de localiser et de réserver facilement un taxi pour et de pouvoir tracer leurs itinéraires. Pour le moment, notre base se développe sur la région d’Antananarivo.

Pour voir le vif du projet, ce travail se poursuit en trois grandes parties distinctes. Dans la première partie, nous allons parler de la présentation générale. La deuxième partie sera consacrée à la conception et la réalisation du projet. La troisième partie montrera le résultat du travail.

# PARTIE 1

# PRESENTATION GENERALE

## CHAPITRE 1 : PRESENTATION DE L’ISPM

### HISTORIQUE

L’I.S.P.M. (Institut Supérieur Polytechnique de Madagascar) qui célèbre cette année son vingt-sixième anniversaire, a été créé en Janvier 1993 par le Professeur RABOANARY Julien Amédée et qui en est le Recteur.

Au début, l’ISPM était connu sous le nom de l’ESSTIM ou École Supérieure de Sciences et Techniques de l’Information à Madagascar. Il ne comptait qu’un seul département comportant comme filière la filière Informatique de Gestion, Génie Logiciel et Intelligence Artificielle (IGGLIA). Son siège se trouvait à Ankadindramamy.

L’Institut a été homologué par l’État suivant l’Arrêté Ministériel n°3725 du 19 Août 1994. Toutes les formations offertes par l’ISPM sont habilitées par le Ministère de l’Enseignant Supérieur et de la Recherche Scientifique.

|  |  |  |
| --- | --- | --- |
| **DOMAINE/GRADE** | **MENTION** | **REFERENCE ET DATE** |
| Sciences et Technologie/Licence et Master | Biotechnologie | Arrêté n°31172/2012-MESupRES du 5 décembre 2012 |
| Sciences et Technologie/Licence | Génie Industriel | Arrêté n°1949/2013-MESupRES du 31 janvier 2013 |
| Sciences et Technologie/Master | Génie Industriel | Arrêté n°21909/2014-MESupRES du 11 juin 2014 |
| Sciences et Technologie/Licence et Master | Génie Civil et Architecture | Arrêté n°1949/2013-MESupRES du 31 janvier 2013 |
| Sciences et Technologie/Licence et Master | Informatique et Télécommunications | Arrêté n°1949/2013-MESupRES du 31 janvier 2013 |
| Sciences de la société/Licence et Master | Droit et Technique des Affaires | Arrêté n°11566/2013-MESupRES du 23 mai 2013 |
| Arts, lettres et Sciences Humaines/Licence | Technique du Tourisme | Arrêté n°33213/2014-MESupRES du 04 novembre 2014 |
| Arts, lettres et Sciences Humaines/Master | Technique du Tourisme | Arrêté n°33033/2015-MESupRES du 05 novembre 2015 |
| Sciences et Technologies/Licence | Technique de l’environnement et du Tourisme | Arrêté n°33213/2014-MESuoRES du 04 novembre 2014 |
| Sciences et Technologies/Master | Environnement et tourisme | Arrêté n°37440/2014-MESupRES du 26 décembre 2014 |

Habilitation des toutes les formations à l’ISPM par le MESupRES

La détermination du fondateur et de ses collaborateurs d’aller toujours de l’avant a contribué au développement rapide de l’Institut, d’où la formation de nouvelles filières. Son transfert à Ambatomaro Antsobolo était donc nécessaire.

Voici l’ordre chronologique de la mise en place des filières à l’ISPM :

* 1993 : Informatique de Gestion, Génie Logiciel et Intelligence Artificielle (IGGLIA) ;
* 1994 : Électronique, Système Informatique et Intelligence Artificielle (ESIIA) ;
* 1995 : Commerce et Administration des Affaires (CAA) ;
* 1996 : Biotechnologie : la filière Industries Agro-Alimentaire (IAA) et la filière Pharmacologie et Industries Pharmaceutiques (PIP) ;
* 1997 : Tourisme Et Environnement (TEE) ;
* 1998 : Électromécanique et Informatique Industriel (EMII) ;
* 1999 : Génie Civil et Architecture (GCA) ;
* 2004 :
* Informatique Multimédia et Technologie de l’Information et de la Communication et Intelligence Artificielle (IMTICIA) ;
* Finances et Comptabilité (FIC) ;
* Agriculture Et Élevage (AEE) ;
* 2009 :
* Économie et Management de Projet (EMP) ;
* Droit et Techniques Juridiques des Affaires (DTJA) ;
* Tourisme et Hôtellerie (TEH) ;
* Industries Chimiques, Minières et Pétrolières (ICMP) ;
* 2010 : Informatique, Statistiques Appliquées et Intelligence Artificielle (ISAIA).

La diversification des disciplines offre aujourd’hui aux étudiants un large choix d’inscriptions. Actuellement, l’ISPM compte plus de deux mille étudiants issus des quatre coins de l’île et même de l’étranger. Une centaine d’enseignants et d’enseignant-chercheurs assurent la formation au sein de l’Institut.

### IDENTITE DE L’ISPM

#### Situation actuelle

Actuellement, l’ISPM et son bureau administratif se trouvent à Ambatomaro Antsobolo, dans un cadre plus étendu, propice aux études.



Photo de l’ISPM

#### Logo

On peut le distinguer et l’identifier par son logo, son Hymne et sa devise « FAHAIZANA - FAMPANDROSOANA – FIHAVANANA ».

Son logo est illustré par trois figures évocatrices de la vision du fondateur, à savoir :

* Une toque,
* Madagascar au sein du monde,
* Deux mains qui se serrent,
* Le tout est érigé sur un socle portant l’abréviation ISPM.
* La toque signifie que l’Établissement procure aux étudiants à la fin de leur parcours à l’ISPM des connaissances dignes des diplômes obtenus.
* La carte de Madagascar au milieu du globe terrestre indique que la formation est de fournir à la nation un rythme de développement au niveau mondial.
* Les deux mains qui se serrent illustrent l’esprit du « Fihavanana » au sein de l’Institut.
* Le logo a été conçu par Falimiamina (1ère promotion CAA) et RAHARINOSY Voajanahary.

Les mots de RAHARINOSY Voajanahary : « Je me souviens très bien du moment où nous (moi et Falimiamina) avons été primés lors de la première journée des portes ouvertes à Antsahavola pour avoir créé ce logo en 1994. La direction avait lancé le concours pour une durée de 1 mois. J’ai conçu la « Graduation Hat », le globe de façon légèrement incliné vers la droite (synonyme d’universalité) et la poignée de main commerciale. Tandis que Falimiamina a conçu le socle ISPM en texture de granite gris et vert, synonyme de quelque chose de solide et durable disait-il avec tant d’enthousiasme. »



Logo de l’ISPM

#### Hira Faneva: « ILAY ISPM TENA MAMINAY »

Ilay ISPM tena maminay  
Tsy mba foinay tokoa rahatrizay  
Toerana nanabeazana ny tenanay  
Mba ho tena olom-banona mahay

**Ho mendrika tokoa   
Ho mafy orina avokoa   
Ny "FAHAIZANA – FAMPANDROSOANA - FIHAVANANA"  
Ho andry sy tokin’ny Tanindrazana**

Ny fahaizana no ampinga ho enti-miady   
Nampitaina, nozaraina sy nomena   
Tsy handrarak’ilo fa ho tena kiady   
Ho enti-mampandroso ity firenenay

Fa ny fihavanana firaisankina   
No asandratray hatrany hatrany   
Manoloana fifaliana fahoriana   
Mandra-pialanay ety an-tany

**Auteur :** RABERANTO Rija / RJA

**Compositeur :** RABERANTO Rija (1ère promotion IGGLIA)

#### Hira Faneva 25ème anniversaire

E, deraina izao Andriamanitra  
Fa na dia nandalo zava-tsarotra   
Tsy mihemotra, tsy mba mikoro   
Ny fianarantsika dia mijoro

**Dimy amby roampolo taona ny Sekoly izao   
Be ireo nandalo ka tafita tao  
Tsy mifidifidy na manavaka  
Fa ISPM dia tena miavaka**

Ry tanora izay mitoetra ao izao  
Maro anie ny soa izay ho azonao  
Fa na lehilahy na vehivavy  
Azo antoka tokoa ny hoavy

Dia mba katsaho ny Fahaizana  
Ka hevero ny Fampandrosoana  
Tanteraho anaty Fihavanana  
Izay no tarigetran’ny fianarana

**A/C :** RAKOTOARIMANANA Johany (CAA 3 2003)

### OBJECTIF ET CURSUS DE FORMATION

L’objectif de l’ISPM est de former des étudiants pour obtenir un diplôme de Master en phase avec les progrès scientifiques et technologiques et conscients des réalités économiques et sociales à Madagascar. Il s’agit d’une formation supérieure à vocation académique et professionnelle de trois cycles.

* L’étudiant passe d’abord une formation de Premier Cycle de trois années (Bac+3) sanctionnée par le Diplôme de Licence.
* Après l’obtention de ce diplôme, l’étudiant peut faire deux choix :
* Soit, entrer dans la vie professionnelle munie du diplôme de Licence.
* Soit, poursuivre ses études du Second Cycle pour la préparation du Diplôme de Master (Bac+5) qui dure deux ans et six mois de stage, après l’obtention du diplôme de Licence.
* Enfin, l’étudiant désirant encore continuer peut effectuer des recherches scientifiques pour obtenir un diplôme de doctorat.

Il est à préciser que les diplômes délivrés par l’ISPM sont reconnus par le Ministère de la Fonction Publique.



Cursus Universitaire à L’ISPM

### FORMATIONS PROPOSEES PAR L’ISPM

Voici la liste des parcours par mention que l’ISPM offre :

* **Mention Informatique et Télécommunications**
* Informatique de Gestion, Génie Logiciel et Intelligence Artificielle (IGGLIA)
* Électronique, Systèmes Informatiques et Intelligence Artificielle (ESIIA)
* Informatique Multimédia, Technologie de l’Information et de la Communication et Intelligence Artificielle (IMTICIA)
* Informatique, Statistiques Appliquées et Intelligence Artificielle (ISAIA)
* **Mention Génie Industriel**
* Électromécanique et Informatique Industrielle (EMII)
* Industries Chimiques, Minières et Pétrolières (ICMP)
* **Mention Génie Civil et Architecture**
* Génie Civil et Architecture (GCA)
* **Mention Droit et Techniques des Affaires**
* Commerce et Administration des Affaires (CAA)
* Économie et Management de Projet (EMP)
* Finances et Comptabilités (FIC)
* Droit et Techniques Juridiques des Affaires (DTJA)
* **Mention Biotechnologie**
* Industries Agro-Alimentaires (IAA)
* Agriculture et Élevage (AEE)
* Pharmacologie et Industries Pharmaceutiques (PIP)
* **Mention Technique du Tourisme**
* Tourisme et Environnement (TEE)
* Tourisme et Hôtellerie (TEH)

Il est à rappeler que toutes les formations de l’ISPM sont habilitées par le Ministère de l’Enseignant Supérieur et de la Recherche Scientifique.

### LA FILIERE ESIIA

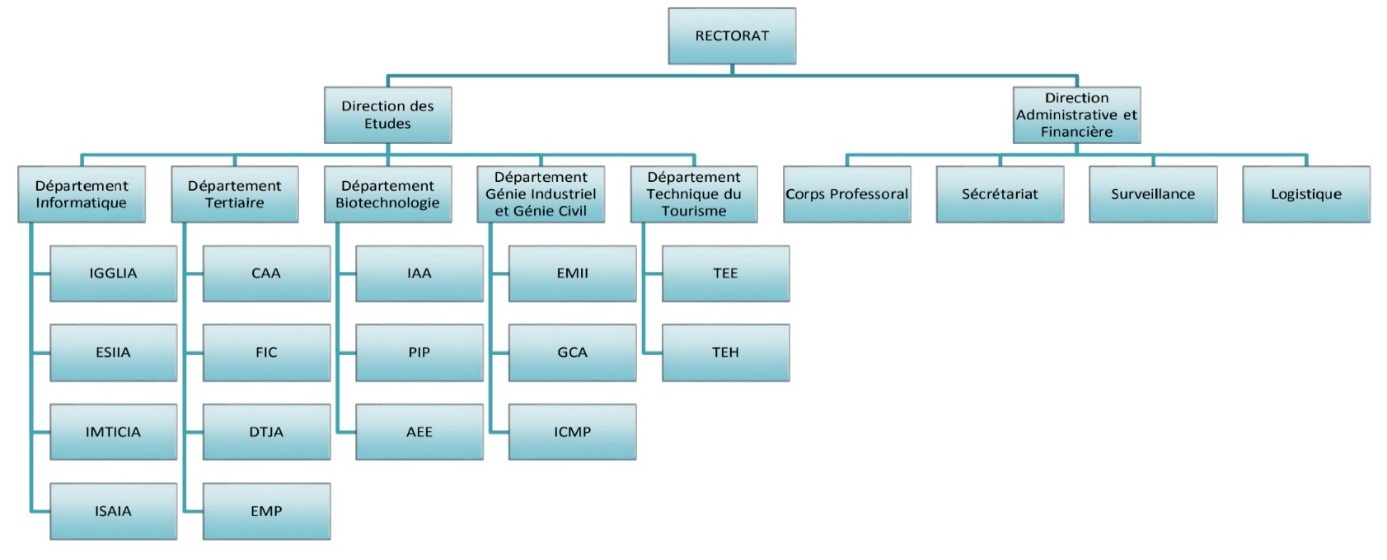
ESIIA (Électronique, Systèmes Informatiques et Intelligence Artificielle) est une filière dont la finalité est la formation des étudiants dans l’informatique et l’électronique et d’une manière générale dans les conceptions des logiciels et applications. Durant le premier cycle, les étudiants de la filière ESIIA sont formés dans les disciplines suivantes :

* Informatique Scientifique, Algèbres Linéaires, Mathématiques Discrètes, Analyse Mathématiques, Logique, Probabilités, Statistiques,
* Algorithmiques et programmation, Structures de Données, Turbo PASCAL, Turbo C, JAVA, Visual Basic, C#, PHP, Technologies web, Programmation Orientée Objet (POO)
* Réseaux informatiques,
* Base de données et MERISE,
* Électricité et Électronique, Structure des ordinateurs, Maintenance des systèmes électroniques (MSE), Maintenance des systèmes informatiques (MSI), Automatique,

En second cycle, la formation initie les étudiants dans l’esprit d’analyse et de conception :

* La Conception Orientée Objet(COO), l’Analyse Orientée Objet (AOO), la Programmation Orientée Objet Avancée (POOA),
* L’Intelligence Artificielle, l’Informatique Décisionnelle, RNA (Réseaux de Neurones Artificiels),
* Méthodologie d’Analyse et le langage de modélisation orientée objet : Unified Modeling Language (UML), Méthode de conduite de projet informatique, Gestion de Projet, Gestion des processus Informatique,
* Théories des Langages, Théories des Automates et Compilateurs,
* Algorithmiques Avancées,
* Cryptographie et Codage,
* SGBDR (Système de Gestion de Base de Données Relationnelles),
* et un cours de management dispensé en cinquième année.

### ORGANIGRAMME DE L’ISPM

Organigramme de l’ISPM

### PARTICULARITES DE L’ISPM

Actuellement, le système LMD est déjà adopté à l’ISPM et les enseignants sont impliqués profondément dans la recherche. Les étudiants doivent également effectuer leurs propres recherches pour approfondir leurs savoirs et leurs acquis.

#### Recherches scientifiques sur les plans national et international

Les étudiants et les enseignant-chercheurs à l’ISPM font des recherches scientifiques d’ordres national et international et les publient dans des journaux internationaux et dans des « proceedings » des conférences internationales prestigieuses. Un extrait de la liste de ces publications est disponible sur http://ispm-edu.com/publications.php.

#### Portes ouvertes et salon de l’ISPM

L’année académique se divise en deux semestres. À chaque semestre ont lieu deux examens dont un « Mid-Term » et un « Final-Exam ». Le deuxième semestre se démarque par l’organisation des « portes ouvertes » et un salon de l’ISPM. Ces évènements sont une occasion pour les étudiants de montrer, au grand public leurs projets effectués dans le cadre des recherches personnelles en équipe.

#### Supports pédagogiques

L’ISPM met à la disposition de ses étudiants :

* Deux laboratoires informatiques pour un total de cent vingt ordinateurs,
* Un laboratoire expérimental pour le département biotechnologie,
* Un laboratoire de travaux pratiques en électricité et électronique,
* Un atelier de travaux pratiques pour les étudiants en mécanique,
* Divers appareils topographiques.

#### L’Examen Clinique

L’Examen « Clinique » est aussi un des plus grandes particularités de l’ISPM. C’est l’épreuve à passer avant la préparation de l’ingéniorat. Il consiste en l’évaluation de l’étudiant sur toutes ses connaissances, depuis la première année jusqu’à le cinquième. La réussite à ce test donne accès au stage ainsi qu’au mémoire de fun du Second Cycle.

1. ***Supports académiques***

L’ISPM offre aussi un bouquet complet de complexe sportif à ses étudiants. À savoir : un terrain de basket-ball, un terrain de football, un terrain de volley-ball, des tables de baby-foot, des tables de tennis de table.

## CHAPITRE 2 : PRESENTATION DE L’ENTREPRISE

### HISTORIQUE

EQIMA est une société incubée par CRP, créée en **2011**, elle s’est donnée comme mission le développement dans l’environnement en tenant compte en permanence des exigences liées à la technologie du Web.

Œuvrant aussi dans le domaine de l’e-Commerce que dans les sites web, dits classiques, et assure les techniques de sécurisation des développements.

Ayant développés pour un certain nombre de clients des extensions pour les **CMS Joomla, WordPress et Prestashop,** et édite **depuis 2013** ses propres extensions et plug-ins. Elles sont disponibles sur son site web  [www.byeqima.com](http://www.byeqima.com) .

#### Logo de l’Entreprise

Le logo de la société se présente comme suit :

* Une main qui clique une souris
* Lettre qui compose le nom EQIMA

C:\Users\ampasampito\Desktop\BOKY ANDRY 5\logo-eqima.png

1. Logo EQIMA

#### Site web de l’EQIMA :

L’adresse du site web est [www.eqima.org](http://www.eqima.org).

Ci-après la représentation graphique de la page d’accueil



1. Site web de l’EQIMA

### ByEQIMA:

ByEQIMA est l’activité d’éditeur de logiciels à destination du marché, des outils de gestion de contenus appartenant à la start-up EQIMA.

La start-up a besoin d’un accélérateur pour développer les ventes en ligne qui est fortement dépendant de la présence dans les annuaires de références mais aussi de l’augmentation conséquente du nombre de visites.

La start-up a une deuxième source de revenus qui est la prestation offshore de développement web (développement spécifique provenant de donneurs d’ordres, développement avancé de boutique e-commerce, etc…)

Voici quelques chiffres :

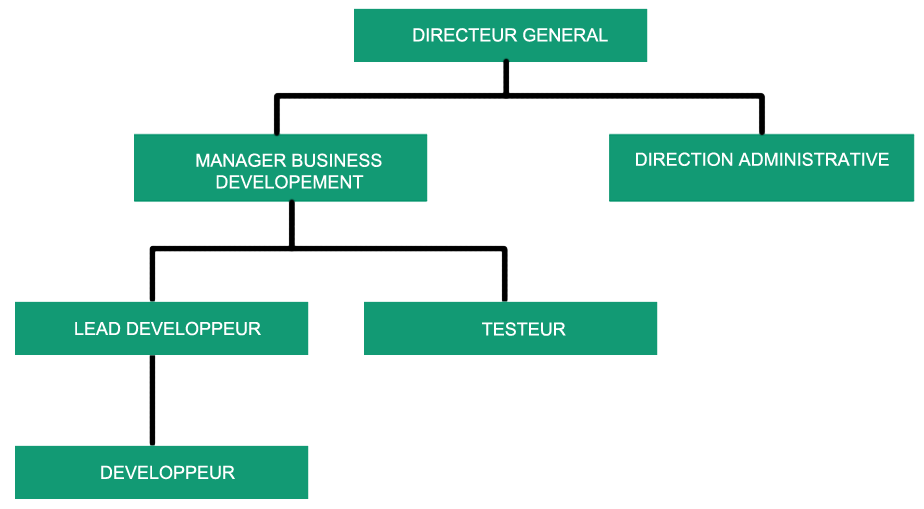
* + 20 000 visiteurs par mois,
  + 35 produits Joomla en rayon,
  + 10 produits WordPress,
  + + de 1400 membres enregistrés.

Les entreprises cibles sont :

* + Les utilisateurs des CMS (gestion de contenu) tel que Joomla, WordPress et Prestashop,
  + Zone géographique internationale (francophone puis anglophone).

### Organigramme de l’EQIMA

On va voir la représentation de l’organigramme de l’Entreprise



1. Organigramme de l’EQIMA

## CHAPITRE 3 : PRESENTATION DU PROJET

### GENERALITE

Ce projet a été effectué au sein de la société EQIMA qui se situe à Andranobevava. Nous avons été assignés à l’équipe des développeurs hybride durant notre stage. Sous la responsabilité d’un Lead développeur, nous avons eu une durée de six mois pour développer le projet.

#### Contexte et Problématique

Depuis quelques années, l’offre de transport n’a cessé de se développer et de s’améliorer, et avec elle les demandes et les exigences des usagés, rendant ces moyens de transport moins compétitifs.

De plus, on peut se perdre facilement en se déplaçant dans les grandes villes. Mis à part, le souci de rater son transport ou d’attendre trop longtemps. Faisant partie du quotidien de la population, ces petits problèmes sur le transport ainsi que le temps qui y est consacré pour trouver un taxi restent toujours un dilemme voire un défi à surélever tous les jours.

A partir de cette situation, plusieurs questions se posent : quelle est l’attente de la population ?

* La facilité ? N’a-t-on pas besoin de simplifier notre routine de vie ?
* La fiabilité ? N’a-t-on pas besoin d’un peu d’assurance et de contrôle sur notre propre trajet ?
* Le gain de temps ? N’est-il pas nécessaire d’économiser notre temps pour des raisons plus importantes ?

#### Solutions

La solution que l’on a apportée pour résoudre ce problème consiste à la conception d’une application mobile qui permet aux usagers d’avoir la maitrise sur ces trois problématiques, c'est-à-dire : pouvoir préparer leurs trajets et mieux utiliser les taxis dans la ville d’Antananarivo en prenant le meilleur itinéraire possible. Cette application est destinée, à la fois, aux clients et aux chauffeurs du taxi.

#### Objectifs

L’objectif du projet est de concevoir une application Android qui rend plus facile et plus fiable la recherche et l’utilisation des transports, dans notre cas, pour les taxis.

Le projet consiste alors à garantir une inter-connectivité en temps réel entre les taxis et les usagers.

### CHOIX DES OUTILS ET LANGAGES

Pour ce projet, on a utilisé principalement :

* IONIC FRAMEWORK comme outil de programmation

C’est est un framework pour créer des applications mobiles semi-hybrides. Ces dernières peuvent fonctionner sous IOS, Android et Windows Phone. Ionic peut aussi créer des applications Chrome ou Electron JS. Son principal atout est d’écrire une seule fois le code pour les multiples plateformes.

Ce framework s’utilise en HTML, CSS, JavaScript mais il offre aussi la possibilité d’utiliser le langage TypeScript. Le langage Typescript est converti en JavaScript lors de la compilation de l’application. L’avantage du TypeScript est d’offrir un langage plus propre et plus structuré que le JavaScript, notamment pour la création des classes en orienté objet.

Une application Ionic est en réalité une WebView utilisant les fonctionnalités natives de la plateforme sur laquelle l’application fonctionne. Ainsi, l’apparence est la même sur les différentes plateformes. Mais on verra par la suite que Ionic utilise des fonctionnalités natives de l’appareil. Nous allons voir dans un premier temps les fonctionnalités de Ionic puis les outils de développement mis à disposition.

* FIREBASE comme le server de base de données

Firebase est une plate-forme de développement d'applications mobiles et Web qui fournit aux développeurs une pléthore d'outils et de services pour les aider à développer des applications de haute qualité, à élargir leur base d'utilisateurs et à générer davantage de profits.

* API Google qui contribue à la majorité des fonctionnalités,

Google Maps est un service de cartographie en ligne. Le service a été créé par Google. ... Deux types de vue sont disponibles dans Google Maps : une vue en plan classique, avec nom des rues, quartier, villes et une vue en image satellite, qui couvre aujourd'hui le monde entier.

On a aussi utilisé des outils complémentaires pendant la réalisation du projet :

* TRELLO, qui permet de planifier les tâches,
* GIT était la gestion de versionning,
* Sublime Text nous a servi comme éditeurs de texte
* ADOBE PHOTOSHOP pour la conception des maquettes graphiques.

Ils seront présentés plus en détails dans la partie II.

# PARTIE 2

**CONCEPTION DU PROJET ET OUTILS**

**CHAPITRE 5 : PRESENTATION DES OUTILS DE DEVELOPPEMENTS**

Dans ce chapitre, nous allons voir en détails l’environnement et outils de développement utilisés ainsi que les langages de programmation. Ces derniers, nous ont permis d’arriver au résultat voulu, que nous présentons à travers les interfaces réalisées.

1. ENVIRONNEMENT DE DEVELOPPEMENT

IONIC est un nouvel environnement de développement d’application mobile qui est :

* Basé sur le concept d’un FRAMEWORK ;
* Suit l’architecture MVC ;
* Interagit parfaitement avec FIREBASE qui est une base de données « Real time »

#### Généralisation

##### Définition

Un framework est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un « squelette » de programme, un canevas. Il est souvent fourni sous la forme d'une bibliothèque logicielle et accompagné du plan de l'architecture cible du framework.

Un framework est conçu en vue d'aider les programmeurs dans leur travail. L'organisation du framework vise la productivité maximale du programmeur qui va l'utiliser — gage de baisse des coûts de construction et maintenance du programme. Le contenu exact du framework est dicté par le type de programme et l'architecture cible pour lequel il est conçu.

##### Types de framework

On trouve différents types de frameworks :

###### framework d'infrastructure système :

Le framework d’infrastructure permet de développer des systèmes d'exploitation, des interfaces graphiques, des outils de communication (exemple : Framework .Net, Struts).

###### framework d'intégration intergicielle (middleware) :

Ce framework consiste à fédérer des applications hétérogènes. Pour mettre à disposition différentes technologies sous la forme d'une interface unique.

###### framework d'entreprise :

Le framework d’entreprise est utilisé pour développer des applications spécifiques au secteur d'activité de l'entreprise.

###### framework de gestion de contenu :

Ce sont les fondations d'un système de gestion de contenu pour la création, la collecte, le classement, le stockage et la publication de « biens numérisés ».

Les principaux avantages de ces frameworks sont la réutilisation de leur code, la standardisation du cycle de vie du logiciel (spécification, développement, maintenance, évolution), ils permettent de formaliser une architecture adaptée au besoin de l'entreprise. Ils tirent parti de l'expérience des développements antérieurs.

##### Avantage d’un « Framework »

Autrement dit, un Framework PHP est un ensemble de codes qui fournit une organisation ainsi qu'un grand nombre de fonctionnalités, dont le nombre et la qualité diffèrent selon les Framework. Ainsi, la maîtrise d'un Framework permet de n’occuper que du site et de laisser le reste à d'autres développeurs, c'est-à-dire sa base, son socle, mais aussi tout ce qui s'articule autour : les classes, les fonctions, etc.

Le premier avantage est la liberté : c'est vous qui décidez de la recette pour arriver au gâteau souhaité et personne d'autre, si vous n'aimez pas le café il suffit de ne pas en mettre, c'est tout simple. Ce soir vous êtes 9 ? Une petite conversion et le tour est joué. Aïe, ce macro-ingrédient n'existe pas encore. Aucun problème il suffit de le créer vous-même ;

Le deuxième est la rapidité : faire votre gâteau à partir des ingrédients de base prendrait trop de temps, l'objectif est quand même d'être rassasié rapidement. Les macro-ingrédients sont le compromis idéal ;

Le troisième est la qualité : les macro-ingrédients ont été assemblés par des personnes très compétentes qui ont normalement envisagé toutes les possibilités d'assemblage. De plus, ceux-ci sont distincts donc vous pouvez facilement modifier/améliorer le vôtre pour ensuite éventuellement le partager ou directement récupérer celui d'un ami ;

Le quatrième est la cohérence : pour éviter les fautes de goût, tous les macro-ingrédients ont été assemblés dans un but commun qui est de faire de bons gâteau ;

La cinquième est la pérennité : tout le monde peut monter des blancs en neige, mais il faut toujours retenir ce qu'il faut mettre à la bonne température, combien de temps tourner dans un sens,... les macro-ingrédients « blancs en neige » est toujours fait de la même manière, la meilleure. Et si jamais un jour il en existe un nouvelle encore meilleure, elle sera utilisée sans même que vous le sachiez.

* Normaliser le développement

Un Framework permet de construire toutes les applications avec les mêmes technologies, les mêmes normes et les mêmes méthodes. Ceci est un point crucial pour les développements informatiques d’une agence Web : n'importe lequel de nos collaborateurs peut très facilement intervenir sur une application (développement, débogage, maintenance, ...).

* Accélérer le développement

Un Framework doit fournir un point de départ pour le développement des applications (création des projets à partir d'assistants par exemple) ce qui permet d'avoir à disposition une structure de base de développement.

* Augmenter la robustesse des applications

Un Framework s'appuie sur des composants techniques et sur une méthodologie éprouvée. Les composants techniques utilisés de développements en développements, gagnent en robustesse. Le Framework prend en charge une grosse partie du code et des éléments techniques. Les

Développeurs qui travaillent sur les projets clients ont de moins en moins de code "purement technique" à écrire et peuvent se concentrer sur les règles métiers. Moins il y aura de code à écrire, moins il y a de chance d'avoir des erreurs et plus l'application sera robuste et fiable.

* Supprimer les problèmes techniques

Un Framework prend en charge les parties vraiment techniques et ardues des développements. La majorité des problèmes et du temps passé le sont malheureusement sur des problèmes techniques. L'intérêt du Framework est de pouvoir capitaliser sur ces difficultés et de s'enrichir au fur et à mesure.

Le Framework étant de plus en plus robuste, la plupart des problèmes techniques sont traités directement par le Framework, et les collaborateurs n'ont plus qu'à maintenir la partie métier ou interface des applications. Tous les collaborateurs utilisant le Framework de développement seront à même d’intervenir assez rapidement et assez facilement sur une application qui aura été développée en respectant ce Framework.

#### IONIC Framework

Pour l’implémentation des différents modules de l’application, nous avons utilisé le Framework IONIC.



Logo IONIC

##### Historique :

Ionic est un framework open-source créé en 2013 par Max Lynch, Ben Sperry, et Adam Bradley. Deux versions distinctes sont disponibles, incompatibles entre elles : la première version, 1.3.3, se base sur AngularJS 1.5.3 tandis que la version 3.5.0 se base sur Angular 4.1.3 et TypeScript.

##### Définition :

Ionic Framework est une boîte à outils d'interface utilisateur open source permettant de créer des applications de bureau et mobiles de haute qualité utilisant des technologies Web (HTML, CSS et JavaScript).

Ionic Framework est axé sur l'expérience utilisateur frontale ou sur l'interaction de l'interface utilisateur d'une application (contrôles, interactions, gestes, animations).

Il s’intègre parfaitement à d’autres bibliothèques ou frameworks, tels que Angular, ou peut être utilisé de manière autonome sans framework frontal avec un simple script include.

#### Objectifs

##### Multiplateforme (Cross-platform)

Créez et déployez des applications fonctionnant sur plusieurs plates-formes, telles que iOS, Android, un ordinateur de bureau et le Web en tant qu'application Web progressive - le tout avec une base de code. Écrivez une fois, exécutez partout.

##### Basé sur des standards web (Web standards-based)

Ionic Framework s'appuie sur des technologies Web standardisées et fiables: HTML, CSS et JavaScript, à l'aide d'API Web modernes telles que Custom Elements et Shadow DOM. Pour cette raison, les composants Ionic ont une API stable et ne sont pas à la merci d'un fournisseur de plate-forme unique.

##### Beau design (Beautiful Design)

Propre, simple et fonctionnel. Ionic Framework est conçu pour fonctionner et s'afficher parfaitement sur toutes les plates-formes. Commencez avec des composants prédéfinis, une typographie, des paradigmes interactifs et un thème de base magnifique (mais extensible).

##### Simplicité (Simplicity)

Ionic Framework est construit dans une optique de simplicité, de sorte que la création d'applications Ionic est agréable, facile à apprendre et accessible à quiconque ayant des compétences en développement Web.

##### License

Ionic Framework est un projet libre et open source, publié sous la licence permise MIT. Cela signifie qu'il peut être utilisé gratuitement dans des projets personnels ou commerciaux. MIT est la même licence utilisée par des projets populaires tels que jQuery et Ruby on Rails.

Le contenu de cette documentation (présent dans le référentiel ionic-docs) est sous licence Apache 2.

##### IONIC CLI

La ligne de commande officielle Ionic, ou interface de ligne de commande, est un outil qui échafaude rapidement les applications Ionic et fournit un certain nombre de commandes utiles aux développeurs Ionic. Outre l'installation et la mise à jour d'Ionic, l'interface de ligne de commande est fournie avec un serveur de développement intégré, des outils de génération et de débogage, et bien plus encore. Si vous êtes membre d'Ionic Appflow, l'interface de ligne de commande peut être utilisée pour créer et déployer des déploiements dans le cloud et administrer votre compte.

##### STRUCTURE

Les applications IONIC sont construites avec Cordova. Cordova est un moyen de conditionner html / css / js dans des applications pouvant fonctionner sur des appareils mobiles et de bureau et fournit une architecture de plug-in permettant d'accéder à des fonctionnalités natives au-delà de la portée de JS exécutée à partir d'un navigateur Web. En tant que telles, les applications ioniques ont la structure de fichier Cordova.

Le répertoire des plateformes contient vos projets iOS et Android. En général, vous n’avez pas besoin de travailler dans ces répertoires à moins d’effectuer un piratage natif personnalisé ou d’envoyer éventuellement votre application en production.

Hooks permet d'effectuer des actions personnalisées à mesure que votre application progresse dans le processus de développement de Cordova. Cela peut être utile pour les projets plus importants nécessitant l'exécution de processus automatisés et la modification du code source, mais qui seront normalement inutilisés.

«Fusionne» consiste à remplacer des fichiers pour des plates-formes spécifiques. Supposons que vous ayez merges / ios / someFile.js et www / someFile.js. Lors du déploiement sur iOS, someFile sera remplacé par la version en fusion. Cela peut être un moyen efficace de créer des modifications spécifiques à la plate-forme de votre application, mais d'autres méthodes sont souvent préférées. Voir Développement multiplateforme pour plus d'informations sur ce sujet.

Les plug-ins sont les endroits où Cordova stocke les plug-ins que vous ajoutez à votre projet.

#### ARCHITECTURE MVC

Ionic est composé: Le MVC (Model-View-Controller ou Model-View-Controller). Il faut savoir que c'est un modèle, une vue et un contrôleur:

##### Le modèle :

C'est la couche responsable du chargement des données.

Aussi celui chargé de faire les bases de données pour envoyer ou recevoir des informations.

Nous savons que la base de données peut être locale dans notre application ou à distance.

##### La vue :

Il s’agit du code qui nous permet de présenter les données fournies par le modèle, à titre d’exemple pour indiquer qu’il s’agit d’une application.

Le code HTML qui nous permet d’afficher la sortie des données traitées.

##### Le contrôleur :

Couche qui sert de lien entre la vue et le modèle. Pour envoyer des commandes au modèle qui vous permettent de mettre à jour votre statut afin que la vue puisse changer de présentation.

B- LANGAGES DE PROGRAMMATION

Ionic framework utilise généralement le TYPESCRIPT, HTML et SCSS.

1. HyperText Markup Langage (H.T.M.L) :
2. Définition

Voici le logo du langage HTML :



1. Logo HTML 5

L’HyperText Markup Langage, généralement abrégé HTML, est le format de donnée conçu pour représenter les pages web. C’est un langage de balisage permettant d’écrire de l’hypertexte, d’où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d’inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l’accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (Java script) et des formats de présentation (feuilles de style en cascade). HTML est initialement dérivé du Standard Generalized Markup Langage (SGML).

##### Structure et syntaxe

La structure du dossier HTML est composée comme suit :

1. Structure et syntaxe Html

#### TYPESCRIPT

Représentation du logo de typescript



1. Logo typescript

##### Présentation :

TypeScript est un langage compilé fortement typé, orienté objet. Il a été conçu par Anders Hejlsberg (concepteur de C #) chez Microsoft.

TypeScript est à la fois un langage et un ensemble d'outils. Il est un sur-ensemble typé de JavaScript compilé en JavaScript. En d'autres termes, TypeScript est JavaScript plus quelques fonctionnalités supplémentaires.

###### Caractéristiques :

* TypeScript est juste JavaScript : - vous avez besoin de connaître JavaScript pour utiliser TypeScript. Tout le code de TypeScript est converti en son équivalent JavaScript pour l'exécution.
* TypeScript prend en charge d'autres bibliothèques JS.
* TypeScript est portable : Il est portable sur les navigateurs, les périphériques et les systèmes d'exploitation. Il peut fonctionner sur n'importe quel environnement sur lequel JavaScript est exécuté. Contrairement à ses homologues, TypeScript n'a pas besoin d'une machine virtuelle dédiée ou d'un environnement d'exécution spécifique pour s'exécuter.

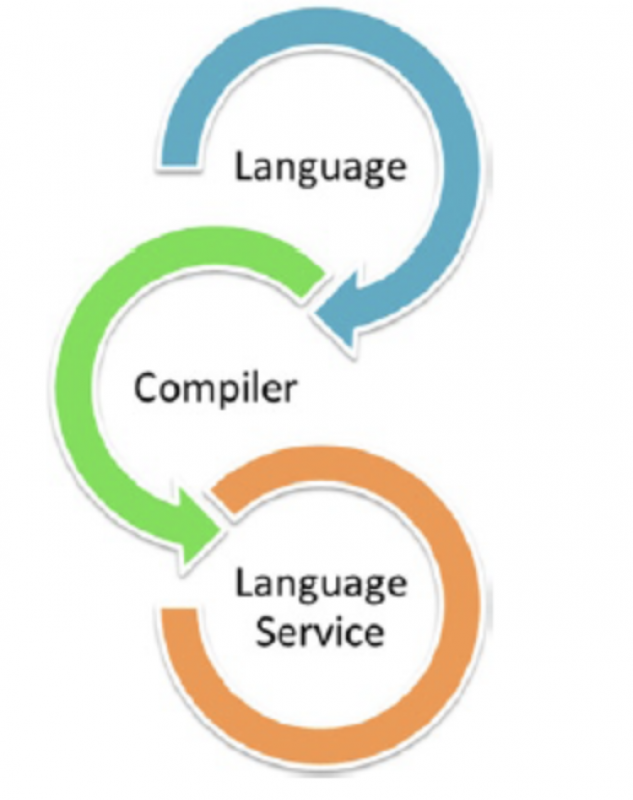
###### Avantages :

* Les erreurs sont détectées plus tôt.
* Moins de bugs.
* Économise beaucoup de temps.
* Fort Typage Statique
* Prend en charge les concepts de programmation orientée objet.
* Langage Compiler.

###### Composants :

TypeScript a trois composants :

* Langage : - Elle comprend la syntaxe, les mots-clés et les annotations de type.
* Le compilateur TypeScript : - Le compilateur TypeScript (tsc) convertit les instructions écrites en TypeScript en son équivalent JavaScript.
* Le service de langage TypeScript : - Le "service de langue" expose une couche supplémentaire autour du pipeline de compilateur de base qui sont des applications de type éditeur. Le service de langage prend en charge l'ensemble commun d'opérations d'éditeur typiques telles que les complétions d'instructions, l'aide à la signature, le formatage et la mise en forme de code, la colorisation, etc.



1. Composants de typescript

#### SCSS  ou SASS :

###### Logo SCSS :

Représentation du logo scss



1. Logo SCSS

###### Logo SASS

Voici le logo du langage SASS



1. Logo SASS

###### Présentation :

Sass (Syntactically Awesome Stylesheets) est un langage dynamique de génération de feuilles de style en cascade initialement développé par Hampton Catlin et Nathalie Weizenbaum. Sass est un préprocesseur CSS. C'est un langage de description compilé en CSS.

C- SERVER DE DONNEES « FIREBASE »

#### 1. Définition :

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel. Lancé en 2011 sous le nom d'Envolve, par Andrew Lee et par James Templin, le service est racheté par Google en octobre 2014. Il appartient aujourd'hui à la maison mère de Google : Alphabet.

Toute l'implémentation et la gestion serveur de Firebase est à la charge exclusive de la société Alphabet. Les applications qui utilisent Firebase intègrent une bibliothèque qui permet les diverses interactions possibles.

Voici le logo de Firebase :



Logo de Firebase

#### NoSQL :

En informatique et en bases de données, NoSQL désigne une famille de systèmes de gestion de base de données (SGBD) qui s'écarte du paradigme classique des bases relationnelles. L'explicitation du terme la plus populaire de l'acronyme est Not only SQL (« pas seulement SQL » en anglais) même si cette interprétation peut être discutée1.

La définition exacte de la famille des SGBD NoSQL reste sujette à débat. Le terme se rattache autant à des caractéristiques techniques qu'à une génération historique de SGBD qui a émergé autour des années 20102. D'après Pramod J. Sadalage et Martin Fowler, la raison principale de l'émergence et de l'adoption des SGBD NoSQL serait le développement des centres de données et la nécessité de posséder un paradigme de bases de données adapté à ce modèle d'infrastructure matérielle3.

L'architecture machine en clusters induit une structure logicielle distribuée fonctionnant avec des agrégats répartis sur différents serveurs permettant des accès et modifications concurrentes mais imposant également de remettre en cause de nombreux fondements de l'architecture SGBD relationnelle traditionnelle, notamment les propriétés ACID.

#### Les services de Firebase :

Les services de Firebase peuvent être divisés en deux groupes :



1. Les services de Firebase

D- LES SERVICES UTILISES

Un Service Web est une fonctionnalité qu’on peut appeler à distance à travers un réseau (comme un intranet d’entreprise ou internet lui-même). Cette fonctionnalité a été réalisée par un tiers langage et qui peut être invoquée à travers des messages XML ou JSON. Nous distinguons deux générations de Services Web : • Service web 1.0 basé sur le protocole SOAP. • Service web 2.0 basé sur le style architectural REST. SOAP est un protocole tandis que REST est un style d’architecture. La différence majeure entre ces 2 éléments est le degré de liaison entre le client et le serveur. Un client développé avec le protocole SOAP ressemble à un logiciel d’ordinateur, car il est étroitement lié au serveur. Si une modification est effectuée d’un côté ou de l’autre, l’ensemble peut ne plus fonctionner. Il faut effectuer des mises à jour du client s’il y a des changements sur le serveur et vice-versa. Un client de type REST sait utiliser un protocole et des méthodes standardisées. Son application doit rentrer dans ce modèle. On ne crée pas de méthodes supplémentaires, on utilise les méthodes standardisées que l’on développe pour le type de media dont on a besoin. Il y a en conséquence beaucoup moins de couplage entre le client et le serveur : un client peut utiliser un service de type REST sans aucune connaissance de l’API. A l’inverse, un client SOAP doit tout savoir des éléments qu’il va utiliser pendant son interaction avec le serveur, sinon cela ne fonctionnera pas. Par ailleurs, REST possède des caractéristiques spécifiques qui le différencient de SOAP. REST est indépendant d’un protocole. On peut utiliser aussi bien le protocole HTTP que FTP, tant qu’il s’agit d’un protocole possédant un schéma standard pour une URI. 55 Expérimentation Vu la simplicité du style REST, nous avons opté pour l’invocation des services web REST existants avec des messages JSON. Dans le cadre de notre application, nous avons utilisé quelques services web existants.

* + 1. Google Maps Android API

Permet de créer des cartes personnalisées et évolutives qui répondent correctement aux besoins des utilisateurs avec des plans statiques et dynamiques, des images Street View et des vues à 360°.

* + 1. Google Service Place AutoComplete

Le service Place Autocomplete (saisie-semi-automatique de lieu) est un service Web qui renvoie des prédictions de lieu en réponse à une requête HTTP. La requête spécifie une chaîne de recherche textuelle et des limites géographiques facultatives. Le service permet de fournir la fonctionnalité de saisie semiautomatique pour les recherches géographiques textuelles et renvoie des lieux tels que des entreprises, des adresses et des points d’intérêt au fur et à mesure de la saisie par l’utilisateur.

* + 1. Google Maps Distance Matrix

Le service Google Maps Distance Matrix API fournit les distances et les durées des trajets pour une matrice de points de départ et de destinations.

* + 1. Direction API

Google Maps Directions API est un service qui calcule des itinéraires entre des points géographiques.

* + 1. Geocoding API

Google Maps Geocoding API est un service qui effectue le géocodage et le géocodage inversé d’adresses.

* Le géocodage : est le processus qui permet de convertir des adresses (comme une adresse postale) en coordonnées géographiques (comme la latitude et la longitude) que vous pouvez ensuite utiliser pour placer des marqueurs sur une carte ou pour positionner la carte.
* Le géocodage inversé : est le processus de conversion de coordonnées géographiques en adresses lisibles. Le service de géocodage inversé de Google Maps Geocoding API vous permet également de retrouver l’adresse correspondant à un identifiant de lieu donné.
  + 1. API FACEBOOK

E- LOGICIELS

#### Sublime Text 3

1. Présentation

Voici le logo de l’éditeur :



Logo de Sublime Text 3

Sublime Text est un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour Vim, riche en fonctionnalités1.

Depuis la version 2.0, sortie le 26 juin 20122, l'éditeur prend en charge 44 langages de programmation majeurs, tandis que des plugins sont souvent disponibles pour les langages plus rares.

1. Fonctionnalités

Sublime Text intègre la plupart des fonctionnalités de base d'un éditeur de texte, dont la coloration syntaxique personnalisable, l’auto complétion, un système de plugins… L'éditeur propose cependant des fonctions plus avancées, dont4 :

* Minimap : prévisualisation de tout le fichier dans une barre latérale ;
* Sélection et édition dans plusieurs sections de code en parallèle ;
* Marque-page au sein même des fichiers ;
* Sauvegarde automatique ;
* Recherche et remplacement par expressions régulières ;
* Support des macros et de plugins en Python ;
* Personnalisation des raccourcis clavier.

Le logiciel propose également d'importer des packages (pour ajouter des langages ou bien des fonctionnalités de Sublime-Text). La version 3 du logiciel propose directement d'installer PackageControl pour ajouter certains thèmes etc.5

#### Google Chrome

Le logo se présente comme suit :



Logo de Google Chrome

C’est un navigateur web propriétaire développé par Google basé sur le projet libre Chromium fonctionnant sous Windows, Mac, Linux, Android et iOS. Chromium est similaire à Chrome, mais ce dernier contient des mouchards14 (envoi d'informations personnelles15 vers un serveur de Google) 16 ou pisteurs17, 18.

Il est annoncé le 1er septembre 2008 dans une bande dessinée de Scott McCloud19, la veille de la sortie de la première version beta. La première version stable est quant à elle dévoilée le 11 décembre 2008. Selon l'institut StatCounter, Google Chrome devient en juin 2012 le navigateur le plus utilisé dans le monde, avec environ un tiers des utilisateurs20.

#### GIT

##### Définition

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version. En 2016, il s’agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes.

Voici le logo de GIT :



Logo git

##### Particularités techniques

Similaire en cela à BitKeeper, Git ne repose pas sur un serveur centralisé, mais il utilise un système de connexion pair à pair. Le code informatique développé est stocké non seulement sur l’ordinateur de chaque contributeur du projet, mais il peut également l'être sur un serveur dédié. C'est un outil de bas niveau, qui se veut simple et performant, dont la principale tâche est de gérer l'évolution du contenu d'une arborescence.

Git indexe les fichiers d'après leur somme de contrôle calculée avec la fonction de hachage SHA-1. Quand un fichier n'est pas modifié, la somme de contrôle ne change pas et le fichier n'est stocké qu'une seule fois. En revanche, si le fichier est modifié, les deux versions sont stockées sur le disque.

Contrastant avec les architectures de logiciel de gestion de versions habituellement utilisées jusqu'alors, Git repose entièrement sur un petit nombre de structures de données élémentaires. Linus Torvalds expliquait ainsi : « Par bien des aspects, vous pouvez considérer git comme un simple système de fichiers. Il est adressé par contenu, et possède la notion de versionnage, mais je l'ai vraiment conçu en prenant le point de vue d'un spécialiste des systèmes de fichiers (après tout, j'ai l'habitude de travailler sur des noyaux) et je n'avais absolument aucune envie de créer un système de gestion de version traditionnel. » Les premières versions de Git offraient une interface rudimentaire pour manipuler ces objets internes avant que les fonctionnalités courantes de gestion de version ne soient ensuite progressivement ajoutées et raffinées.

Git est considéré comme performant, au point que certains autres logiciels de gestion de version (Darcs, Arch), qui n'utilisent pas de base de données, se sont montrés intéressés par le système de stockage des fichiers de Git pour leur propre fonctionnement. Ils continuent toutefois à proposer des fonctionnalités plus évoluées.

Dès le début, Git a été pensé dans le but de fonctionner de façon décentralisée, c'est d'ailleurs l'une des clefs de son succès.

La décentralisation de Git a aussi beaucoup apporté au développement des logiciels libres, puisque le besoin de demander un compte sur un dépôt SVN ou CVS centralisé devient obsolète. Il suffit de forker un projet ou de le cloner pour commencer à travailler dessus (avec tout l'historique du projet en local) et ensuite de proposer sa contribution (pull request) au dépôt principal (mainteneur principal du projet).

Les serveurs Git utilisent par défaut le port 9418 pour le protocole spécifique à Git. Les protocoles HTTP, HTTPS et SSH (et leurs ports standards) peuvent aussi être utilisés.

##### Fonctionnement :

Git possède deux structures de données : une base d'objets et un cache de répertoires. Il existe cinq types d'objets :

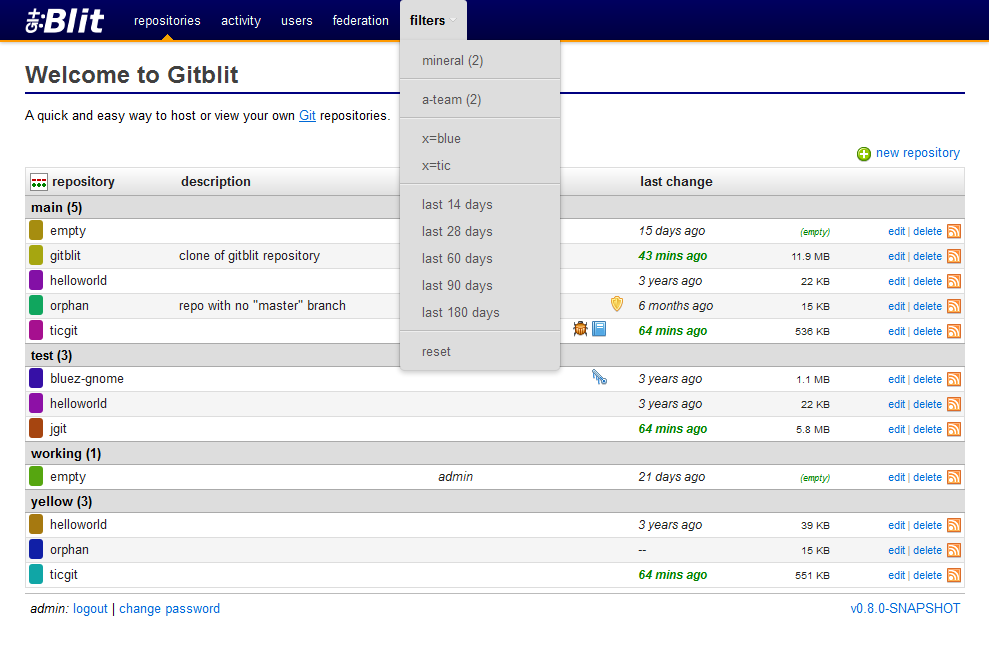
* l'objet blob (pour binary large object désignant un ensemble de données brutes), qui représente le contenu d'un fichier ;
* l'objet tree (mot anglais signifiant « arbre »), qui décrit une arborescence de fichier. Il est constitué d'une liste d'objets de type blobs et des informations qui leur sont associées, tel que le nom du fichier et les permissions. Il peut contenir récursivement d'autres trees pour représenter les sous-répertoires ;
* l'objet commit (résultat de l'opération du même nom signifiant « valider une transaction »), qui correspond à une arborescence de fichiers (tree) enrichie de métadonnées comme un message de description, le nom de l'auteur, etc. Il pointe également vers un ou plusieurs objets commit parents pour former un graphe d'historiques ;
* l'objet tag (étiquette) qui est une manière de nommer arbitrairement un commit spécifique pour l'identifier plus facilement. Il est en général utilisé pour marquer certains commits, par exemple par un numéro ou un nom de version (2.1 ou bien Lucid Lynx).
* l'objet branch (branche) qui contient une partie de l'avancement du projet. Les branches sont souvent utilisées pour avancer dans une partie du projet sans impacter une autre partie.

La base des objets peut contenir n'importe quel type d'objets. Une couche intermédiaire, utilisant des index (les sommes de contrôle), établit un lien entre les objets de la base et l'arborescence des fichiers.

Chaque objet est identifié par une somme de contrôle SHA-1 de son contenu. Git calcule la somme de contrôle et utilise cette valeur pour déterminer le nom de fichier de l'objet. L'objet est placé dans un répertoire dont le nom correspond aux deux premières lettres de la somme de contrôle. Le reste de la somme de contrôle constitue alors le nom du fichier pour cet objet.

Git enregistre chaque révision dans un fichier en tant qu'objet blob unique. Les relations entre les objets blobs sont déterminées en examinant les objets commit. En général, les objets blobs sont stockés dans leur intégralité en utilisant la compression de la zlib. Ce principe peut rapidement consommer une grande quantité de place disque ; de ce fait, les objets peuvent être combinés dans des archives, qui utilisent la compression différentielle (c'est-à-dire que les blobs sont enregistrés sous la forme de différences par rapport aux autres blobs).

##### GIT BLIT



Interface GIT BLIT

Gitblit est une pile Java pur à code source ouvert permettant de gérer, d'afficher et de servir les référentiels Git.

Il est principalement conçu comme un outil pour les petits groupes de travail souhaitant héberger des référentiels centralisés.

#### Android Studio

1. Généralité

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux4.



Logo Android Studio

1. Fonctionnalités

Android Studio permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions variées simultanément6, 7. Il intègre par ailleurs un émulateur permettant de faire tourner un système Android virtuel sur un ordinateur.

#### Adobe Photoshop

Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur, lancé en 1990 sur MacOS puis en 1992 sur Windows. Édité par Adobe, il est principalement utilisé pour le traitement des photographies numériques, mais sert également à la création ex nihilo d’images. Il travaille essentiellement sur images matricielles car les images sont constituées d’une grille de points appelés pixels. L’intérêt de ces images est de reproduire des gradations subtiles de couleurs.



Logo d’Adobe Photoshop

#### TRELLO

Trello est un outil de gestion de projet en ligne, lancé en septembre 2011, et inspiré par la méthode Kanban de Toyota. Il est basé sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement. La version de base est gratuite, tandis qu'une formule payante permet d'obtenir des services supplémentaires. Le service est disponible en plusieurs langues (23 en juin 2016).

Son logo se présente comme suit :



Logo Trello

## CHAPITRE 6 : ANALYSE ET CONCEPTION

### INTRODUCTION

Ce chapitre aborde une partie très importante de notre mémoire qui est la conception de l’application.

En premier lieu, nous allons présenter le processus de développement utilisé qui est indispensable à n’importe quel type de projet informatique, et pour notre application, nous avons choisis le processus unifié (UP), ce dernier est généralement utilisé pour les grands projets, mais sa simplicité fait qu’il est aussi adapté à n’importe type de projet, il est caractérisé par le fait qu’il est accompagné par les différents diagrammes UML et piloté par les cas d’utilisation.

En second lieu, nous allons présenter les diagrammes UML et s’accentuer sur ceux que nous avons utilisés.

Nous parlerons ensuite de l’analyse de notre système en utilisant les diagrammes associés à cette étape.

En dernier lieu, nous allons parler de la conception de notre système en utilisant les différents diagrammes que l’on va présenter en détails.

### PROCESSUS UNIFIE

* + - 1. Définition

Le processus unifié en anglais Unified Process (UP) est un processus de développement logiciel itératif, centré sur l’architecture, piloté par des cas d’utilisation et orienté vers la diminution des risques. C’est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d’application, à différents types d’entreprises, à différents niveaux de compétences et à différentes tailles de l’entreprise.

#### Les caractéristiques d’UP

Le processus unifié englobe plusieurs caractéristiques parmi elles, on peut en distinguer 3 principales caractéristiques qui sont les suivantes :

* + - * 1. Itératif

L’itération est une répétition d’une séquence d’instructions ou d’une partie de programme un nombre de fois fixé à l’avance ou tant qu’une condition définie n’est pas remplie, dans le but de reprendre un traitement sur des données différentes.

Elle qualifie un traitement ou une procédure qui exécute un groupe d’opérations de façon répétitive jusqu’à ce qu’une condition bien définie soit remplie.

b) Piloté par les cas d’utilisation d’UML

Le but principal d’un système informatique est de satisfaire les besoins du client. Le processus de développement sera donc accès sur l’utilisateur.

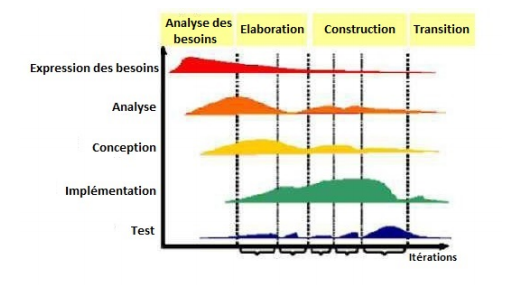
Les cas d’utilisation permettent d’illustrer ces besoins. Ils détectent puis décrivent les besoins fonctionnels (du point de vue de l’utilisateur), et leur ensemble constitue le modèle de cas d’utilisation qui dicte les fonctionnalités complètes du système

c) Centré sur une architecture

Le processus unifié (UP) gère le processus de développement par deux axes.

* L’axe vertical : représente les principaux enchaînements d’activités, qui regroupent les activités selon leur nature. Cette dimension rend compte l’aspect statique du processus qui s’exprime en termes de composants, de processus, d’activités, d’enchaînements, d’artefacts et de travailleurs.
* L’axe horizontal : représente le temps et montre le déroulement du cycle de vie du processus. Cette dimension rend compte de l’aspect dynamique du processus qui s’exprime en termes de cycles, de phases, d’itérations et de jalons.

La figure suivante représente l’architecture du processus unifié.



L’architecture du processus unifié

1. L’axe vertical

Dans l’axe vertical de l’architecture d’UP, on peut distinguer 5 activités qui sont les suivantes :

* Expression des besoins

L’expression des besoins comme son nom l’indique, permet de définir les différents besoins :

* Inventorier les besoins principaux et fournir une liste de leurs fonctions.
* Recenser les besoins fonctionnels (du point de vue de l’utilisateur) qui conduisent à l’élaboration des modèles de cas d’utilisation.
* Appréhender les besoins non fonctionnels (technique) et livrer une liste des exigences.
* Analyse

L’objectif de l’analyse est d’accéder à une compréhension des besoins et des exigences du client. Il s’agit de livrer des spécifications pour permettre de choisir la conception de la solution.

Un modèle d’analyse livre une spécification complète des besoins issus des cas d’utilisation et les structure sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l’architecture), la modification et la maintenance du futur système.

* Conceprion

La conception permet d’acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l’utilisation des composants et au système d’exploitation. Elle Détermine les principales interfaces et les transcrit à l’aide d’une notation commune.

* Implémentation

L’implémentation est le résultat de la conception pour implémenter le système sous forme de composants, c’est-à-dire, de code source, de scripts, de binaires, d’exécutables et d’autres éléments du même type.

* Test

Les tests permettent de vérifier des résultats de l’implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.

1. L’axe horizontal

Dans l’axe horizontal on peut distinguer 4 phases qui sont les suivantes :

* Analyse des besoins

L’analyse des besoins donne une vue du projet sous forme de produit fini. Cette phase porte essentiellement sur les besoins principaux (du point de vue de l’utilisateur), l’architecture générale du système, les risques majeurs, les délais et les coûts.

* Elaboration

L’élaboration reprend les éléments de la phase d’analyse des besoins et les précise pour arriver à une spécification détaillée de la solution à mettre en œuvre.

Cela permet de préciser la plupart des cas d’utilisation, de concevoir l’architecture du système et surtout de déterminer l’architecture de référence. Au terme de cette phase, les chefs de projet doivent être en mesure de prévoir les activités et d’estimer les ressources nécessaires à l’achèvement du projet.

* Construction

La construction est le moment où l’on construit le produit. L’architecture de référence se métamorphose en produit complet. Le produit contient tous les cas d’utilisation que les chefs de projet, en accord avec les utilisateurs ont décidé de mettre au point pour cette version.

* Transition

Le produit est en version bêta. Un groupe d’utilisateurs essaye le produit et détecte les anomalies et défauts. Cette phase suppose des activités comme la formation des utilisateurs clients, la mise en œuvre d’un service d’assistance et la correction des anomalies constatées.

### UML (Unified Modeling Language)

1. Définition

Le langage de modélisation unifié (UML) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

En effet UML est un langage avec une syntaxe et des règles bien définies qui tentent à réaliser des buts d’écrit grâce à une représentation graphique formée de diagrammes et une modélisation textuelles qui vient enrichir la représentation graphique.

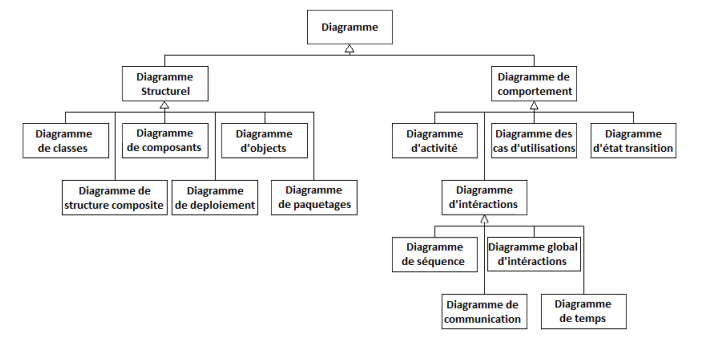
Le langage de modélisation unifié a un certain nombre d’avantages, nous citons quelques-uns :

* Universel ;
* Adopté par les grandes entreprises ;
* Utiliser à l’international UML contrairement à Merise qui est français ;
* Notation unifié ;
* Facile à comprendre ;
* Adopté par plusieurs processus de développement ;
* Limite les risques d’erreurs ;
* N’est pas limité au domaine informatique.

1. Les différents diagrammes UML

Le langage de modélisation unifié (UML) s’articule autour de treize types de diagrammes, chacun d’eux étant dédié à la représentation des concepts particuliers d’un système logiciel. Ces types de diagrammes sont répartis en deux grands groupes représentés.

La figure suivante illustre ces différents diagrammes.



Les diagrammes UML

Les diagrammes que nous allons présenter sont les diagrammes que nous avons jugé nécessaire à utiliser dans notre conception.

* Diagramme de cas d’utilisation

Ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système. Il constitue un des diagrammes les plus structurants dans l’analyse d’un système.

* Diagramme de séquence

C’est la représentation des interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d’utilisation en considérant les différents scénarios associés, en d’autres termes le diagramme de séquence va nous permettre de décrire les scénarios des cas d’utilisation.

* Diagramme d’activité

Le diagramme d’activité est un diagramme comportemental d’UML, permettant de représenter le déclenchement d’événements en fonction des états du système, il présente un certain nombre de points communs avec le diagramme d’état-transition puisqu’il concerne le comportement interne des opérations ou des cas d’utilisation.

* Diagramme de classes

Le diagramme de classe constitue l’un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d’association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes.

Les traitements sont matérialisés par des opérations. Le détail des traitements n’est pas représenté directement dans le diagramme de classe.

* Diagramme de déploiement

Le diagramme de déploiement est une vue statique qui sert à représenter l’utilisation de l’infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que les relations entre eux.

### SPECIFICATION DES BESOINS

1. Besoins fonctionnels

Dans cette partie nous détaillons les fonctionnalités, que le système doit fournir aux différents acteurs, qui se présentent comme suit :

* Localisation de l’utilisateur.
* Réservation d’un taxi en temps réel.
* Traçage de l’itinéraire.
* Estimation du taxi et de la distance.
* Suivre l’approche du bus en temps réel.
* Notifier le client lorsque le taxieur arrive.
* Consultation de l’historique.

1. Besoins non fonctionnels

Il s’agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou de type de conception.

Pour cela notre futur système doit répondre aux caractéristiques suivantes :

* Ergonomie.
* Rapidité de traitement.
* Facilité d’utilisation.
* Sécurité.

### Analyse des besoins

La première étape de la conception consiste à analyser la situation pour tenir compte des contraintes, des risques et de tout autre élément pertinent et assurer un ouvrage ou un processus répondant aux besoins du client.

1. Les acteurs

Un acteur est un utilisateur type qui a toujours le même comportement vis-à-vis d’un cas d’utilisation.

Ainsi les utilisateurs d’un système appartiennent à une ou plusieurs classes d’acteurs selon les rôles qu’ils tiennent par rapport au système. Une même personne physique peut se comporter en autant d’acteurs différents que le nombre de rôles qu’elle joue vis-à-vis du système.

Les acteurs que comporte notre système sont :

* Le client
* Le taxieur

1. Diagramme de cas d’utilisation
2. Identification des acteurs

Un cas d’utilisation représente une fonctionnalité du système. Cette fonctionnalité est définie par une action déclenchant, un ou plusieurs déroulements possibles et éventuellement une fin.

Le tableau suivant représente les différents cas d’utilisation associé à notre système

|  |  |  |
| --- | --- | --- |
| **Fonction** | **Cas d’utilisation** | **Acteurs** |
| Authentification à l’application via Facebook | S’authentifier | Client, Taxieur |
| * Accepter une demande * Refuser une demande | Gérer les demandes de réservation | Taxieur |
| * Sélectionner un taxi * Indiquer la destination * Confirmer * Annuler | Réserver un Taxi | Client |
| Consultation de l’historique | Consulter son historique | Client, Taxieur |
| Disponibilité du taxieur | Disponibilité | Taxieur |

Les différents cas d’utilisation

1. Réalisation du diagramme de cas d’utilisation

La figure suivante représente le diagramme de cas d’utilisation global de notre système.

1. Description textuelles des cas d’utilisations

Cette partie sera consacrée pour la description des cas d’utilisations, chaque description sera représentée sous forme de tableau qui indiquera le nom du cas d’utilisation, l’acteur responsable, l’objectif, ainsi que le scenario.

* Cas d’utilisation « S’authentifier »

Le tableau suivant représente une description textuelle du cas d’utilisation « S’authentifier »

|  |  |
| --- | --- |
| **Cas d’utilisation** | **S’authentifier** |
| **Acteurs** | * Client * Chauffeur de bus |
| **Objectif** | Authentification à l’application |
| **Scénario nominal** | * L’utilisateur lance l’application * Lancement de l’animation de départ * L’interface de connexion avec facebook lui sera affichée |

Description du cas d’utilisation « S’authentifier »

* Cas d’utilisation « Gérer les demandes de réservation »

Le tableau suivant représente une description textuelle du cas d’utilisation « Gérer les demandes de réservation »

|  |  |
| --- | --- |
| **Cas d’utilisation** | **Emettre la position du bus** |
| **Acteur** | Taxieur |
| **Objectifs** | La gestion des demandes de réservation du client |
| **Scénario nominal** | * Le taxieur reçoit une demande de réservation du client * L’itinéraire lui sera affiché sur sa map * Il indique le prix au client et clique sur accepter * Le client confirme la demande et notifie le taxieur * Le taxieur devient automatiquement indisponible et sera supprimé de la map des autres clients |
| **Alternatives** | Si le taxieur n’accepte pas la demande il clique sur refuser |

Description de cas d’utilisation « Gérer les demandes de réservation »

* Cas d’utilisation « Réserver un taxi »

Le tableau suivant représente une description textuelle du cas d’utilisation « Réserver un taxi »

|  |  |
| --- | --- |
| **Cas d’utilisation** | **Rechercher un bus** |
| **Acteur** | Client |
| **Objectif** | La réservation d’un taxi |
| **Scénario nominal** | * Le client sélectionne la destination sur la map * Il sélectionne un taxi * Après qu’il l’ait confirmé la demande de réservation, il l’envoi au taxi sélectionné comprenant sa source, sa destination, le temps ainsi que la distance * Il accepte le prix indiqué par le taxieur après que ce dernier ait accepté la demande de réservation * Tous les autres taxieurs seront supprimés de sa map et ne visualisera que le taxi qu’il a réservé * Le taxieur sera notifié indiquant que la course a été confirmée |
| **Alternatives** | Si le taxieur refuse la demande de réservation, le client recevra une notification qui indique que le taxieur a refusé sa demande |

Description du cas d’utilisation « Réserver un taxi »

### Conception

Cette partie sera consacrée pour la conception de notre système, on utilisera le diagramme d’activité, les diagrammes de séquences pour représenter les scenarios, ainsi que le diagramme de classe pour représenter les entités manipulées par les utilisateurs, et enfin le diagramme de déploiement qui représente l’architecture de notre système.

1. Diagramme d’activités

La figure suivante représente le diagramme global d’activité du système.

1. Patrons de conceptions

Un design pattern ou pattern de conception consiste en un schéma à objets qui forme une solution à un problème connu et fréquent. Le schéma à objets est constitué par un ensemble d’objets décrits par des classes et des relations liant les objets. Les patterns répondent à des problèmes de conception de logiciels dans le cadre de la programmation par objets. Ce sont des solutions connues et éprouvées dont la conception provient de l’expérience de programmeurs [20]. Ainsi, dans le cadre de notre application, nous avons utilisés les designs patterns suivants :

1. Le pattern DAO

Ce pattern permet de faire le lien entre la couche d’accès aux données et la couche métier d’une application. Il permet de mieux maîtriser les changements susceptibles d’être opérés sur le système de stockage des données.

Donc, par extension, de préparer une migration d’un système à un autre

(BDD vers fichiers XML, par exemple. . .). Ceci se fait en séparant accès aux données (BDD) et objets métiers (POJO), ce modèle complète le modèle MVC qui préconise de séparer dans des classes les différentes problématiques.

1. Le pattern MVC

Modèle-vue-contrôleur ou MVC est un motif d’architecture logicielle destiné aux interfaces graphiques, le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

* Un modèle (Model) contient les données à afficher.
* Une vue (View) contient la présentation de l’interface graphique.
* Un contrôleur (Controller) contient la logique concernant les actions effectuées par l’utilisateur

Le schéma suivant représente le modèle vue contrôleur.

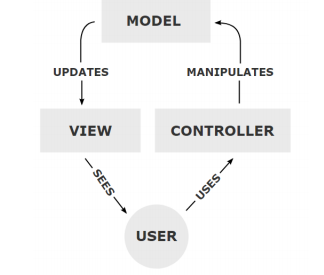


Schéma du MVC

1. Diagrammes de séquences

Nous allons présenter quelques diagrammes de séquences et expliquer le fragment d’interactions que nous avons utilisé dans ces diagrammes.

1. Fragments d’interactions

Un fragment d’interaction correspond à un ensemble d’interactions auquel on applique un opérateur, il se représente globalement comme un diagramme de séquence dans un rectangle avec indication dans le coin à gauche du nom de l’opérateur.

L’opérateur que nous allons utiliser dans les diagrammes de séquences est l’opérateur « alt » qui correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il permet aussi d’utiliser les classes de type sinon et se représente dans un fragment possédant au moins deux parties séparées par des pointillés.

1. Réalisation des diagrammes de séquences
   * 1. Diagramme de séquences du cas d’utilisation « S’authentifier »

L’utilisateur qui représente le client ou bien le chauffeur du bus doit s’authentifier via facebook.

Le diagramme ci-dessous représente le diagramme de séquence du cas ou l’utilisateur s’authentifie.

* + 1. Diagramme de séquence du cas d’utilisation « Rechercher un taxi »

Pour que le client puisse rechercher un bus, il devra passer par le scenario indiquer dans la figure suivante qui représente le diagramme de séquence du cas d’utilisation « Rechercher un bus ».

* + 1. Diagramme de séquence du cas d’utilisation « Emettre la position du bus »

1. Diagramme de classes
   1. Dictionnaire de données

Le tableau ci-dessous représente le dictionnaire de données du diagramme de classe. Ces informations sont le fruit d’une collecte informations pertinentes jugées, il est à signaler que ce dictionnaire a été épuré.

1. Création de la base de données

Nous assistons ces dernières années à un changement des habitudes des utilisateurs notamment sur l’Internet mondial avec la démocratisation des offres haut débits.

Dans ce contexte certaines personnes se sont rendu compte qu’un modèle relationnel des données atteignait ses limites : le NoSQL allait faire son entrée dans le monde de la représentation des données.

Le NoSQL, pour "not only SQL" désigne les bases de données qui ne sont pas fondées sur l’architecture classique des bases de données relationnelles. Développé à l’origine pour gérer du Big Data, l’utilisation des bases de données NoSQL a explosé depuis quelques années, elles ont été conçues pour résoudre les problèmes de traitements de données en volume, multi-sources et multi-formats.

Aujourd’hui le NoSQL apporte une nouvelle façon d’appréhender la modélisation des données. Le NoSQL n’est pas là pour remplacer les bases de données relationnelles, il répond à des besoins différents mais les deux approches peuvent cohabiter. Le choix de l’une ou de l’autre sera donc fortement dépendant du contexte et du besoin.

L’intérêt des systèmes de stockage NoSQL réside surtout dans les choix d’architecture logicielle qui ont été pris lors de leurs conceptions. Parmi les raisons principales qui ont mené à la création de ces systèmes, on retrouve surtout deux points principaux :

* La possibilité d’utiliser autre chose qu’un schéma fixe sous forme de tableaux dont toutes les propriétés sont fixées à l’avance.
* La possibilité d’avoir un système facilement distribué sur plusieurs serveurs et avec lequel un besoin supplémentaire en stockage ou en montée en charge se traduit simplement par l’ajout de nouveaux serveurs

Même si initialement le NoSQL est une réponse à la croissance toujours plus importante sur Internet, il trouve également sa place dans le monde de l’entreprise sur des échelles moindres.

Notre application relève des systèmes distribués et gère des données hébergées sur des serveurs et qui changent en temps réel nous avons donc besoin de stocker nos données en temps réel et y accéder aussi rapidement, et les bases de données traditionnelles à savoir relationnelles n’arrivent pas à fournir un temps de réponse assez satisfaisant, c’est la raison pour laquelle, nous avons opté donc pour l’utilisation des bases de données NoSQL.

Les familles de bases de données NoSQL

La mise en œuvre de la base de données NoSQL est possible grâce à la dénormalisation, dans une relation entre deux tables A et B, la dénormalisation qui consiste à dupliquer certaines données de la table B dans la table A afin d’optimiser les requêtes qui pourront se contenter d’interroger la table A sans avoir à faire de jointures entre A et B.

Nous allons représenter dans ce qui suit, les principales familles du concept NoSQL.

Clé/valeur

Le but de la famille clé-valeur est l’efficacité et la simplicité. Un système clé-valeur agit comme une énorme table de hachage distribuée sur le réseau. Tout repose sur le couple Clé/Valeur.

La clé identifie la donnée de manière unique et permet de la gérer. La valeur contient n’importe quel type de données.

En colonnes

Traditionnellement, les données sont représentées en ligne, représentant l’ensemble des attributs.

Le stockage orienté colonne change ce paradigme en se focalisant sur chaque attribut et en les distribuant. Il est alors possible de focaliser les requêtes sur une ou plusieurs colonnes, sans avoir à traiter les informations inutiles (les autres colonnes).

Orientées graphes

Dans la base orientée graphe, les données stockées sont :

* Les nœuds
* Les liens
* Les propriétés sur les nœuds et les liens

Les requêtes que l’on peut exprimer sont basées sur la gestion de chemins.

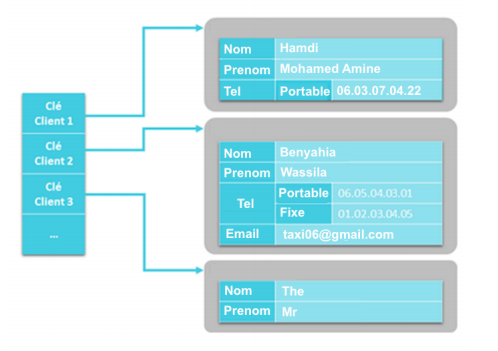
Orientées document :

Les bases orientées documents ressemblent sans doute le plus à ce que l’on peut faire dans une base de données classique pour des requêtes complexes.

Le but de ce stockage est de manipuler des documents contenant des informations avec une structure complexe (types, listes, imbrications).

Il repose sur le principe de la clé/valeur, mais avec une extension sur les champs qui composent ce document].

La figure suivante illustre un exemple de cette famille.



Exemple d’une BDD NoSQL orienté document.

Dans le cadre de notre application, notre choix est porté sur la conception orientée document et ceci pour les raisons suivantes :

* L’avantage de cette solution est d’avoir une approche structurée de chaque valeur, formant ainsi un document.
* Les documents sont sous le format du très populaire format d’échange de données Json (JavaScript Object Notation), ce qui nous permet donc une grande interopérabilité.
* Elle nous permet une sérialisation de nos objets sans mapping : on stock directement nos objets java et on les récupère en tant qu’objet.
* Nous pouvons requêter et manipuler ces documents, et notamment récupérer, via une seule clé, un ensemble d’informations structurées de manière hiérarchique.

Schéma de notre base de données

La figure illustré ci-dessous représente notre schéma de base de données qui est en NoSQL de type orienté document.

1. Diagramme de déploiement

Le diagramme de déploiement illustré dans la figure ci-dessous démontre le matériel nécessaire qu’on a utilisé dans notre projet et qui représente un terminal mobile, en plus des differentes API utilisés et des services de firebase auxquels on a eu recours, sans oublier l’utilisation de l’internet qui a été indispensable, de plus les échanges de données se sont fait en json.

# PARTIE 3

# RESULTATS

## CHAPITRE 5 : PRESENTATION DE L’APPLICATION

### GENERALITE

Cette application mobile intitulée « PIKLA », propose une facilité d’utilisation et beaucoup de fiabilité des résultats. Le projet présente 2 applications l’une dédiée pour les chauffeurs et l’autre pour les clients.

En général, les applications interagissent entre eux et permettent de réserver un taxi qui se trouvent aux alentours de la position actuelle du client, de voir les distances et le temps d’attente.

### UTILISATEURS

Deux types d’utilisateurs pourront utilisés cette application

* Chauffeur de taxi ou taxieur,
* Client.

### CONFIGURATION REQUISE

Afin d’utiliser l’application, il faut avoir :

* Un smartphone avec système Android 4 ou plus,
* Une connexion internet,
* Une permission pour déterminer les positions.

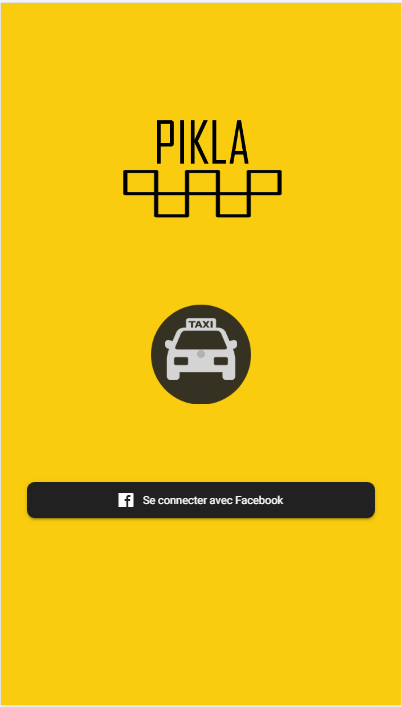
### INTERFACES ET FONCTIONNEMENT

Nous allons voir la représentation graphique de chaque page pour les 2 applications :

* + - 1. Interface d’authentification

Avant toute action, un nouvel utilisateur devra s’authentifier par Facebook avant de bénéficier de l’utilisation de l’application.

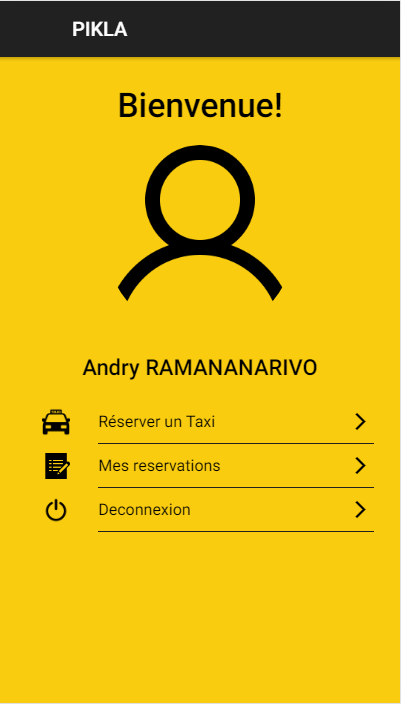
Cette étape est illustrée dans la figure suivante :



Interface d’authentification

* + - 1. Interface principal du Client

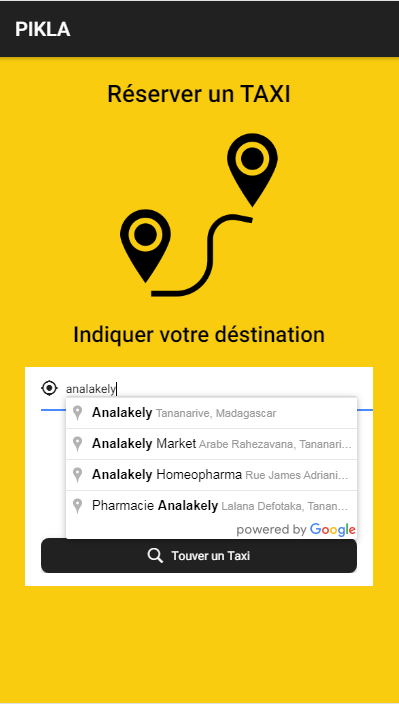
Lorsque le client accède à l’application, il pourra consulter son menu comme le montre la figure suivante :



Interface principal du client

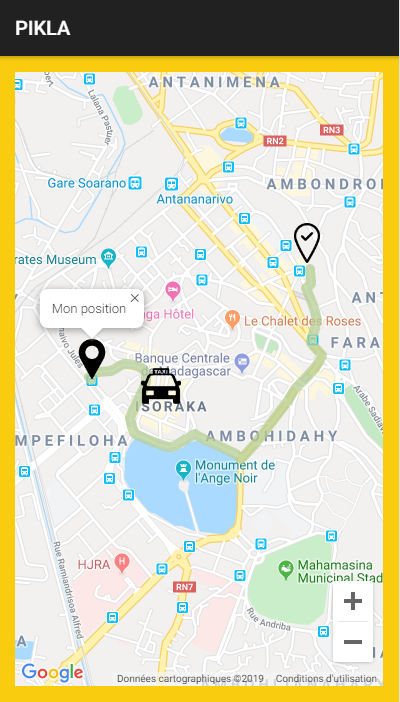
* + - 1. Etapes de réservation

La première étape sera d’indiquer la destination où il veut se rendre.



Interface de recherche de destination

Ensuite, il aura un aperçu de tous les taxieurs qui se trouvent à un rayon de 5km, ainsi que le trajet à faire qui part de la position actuelle du client.

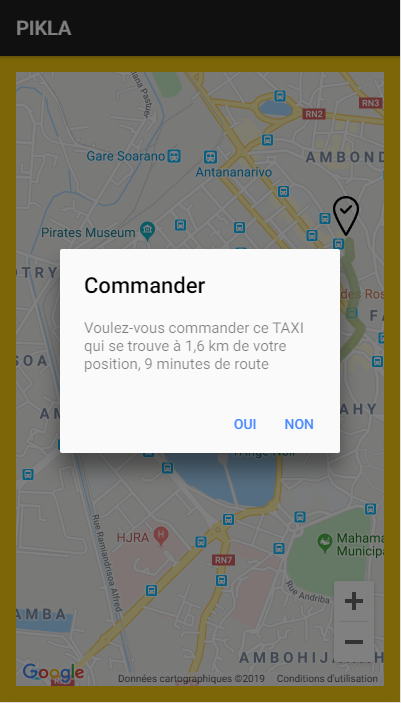


Aperçu des taxis et du projet

Le Client Choisi ensuite un taxi et un message de confirmation serait affiché avec la distance et la durée entre le client et le taxi.



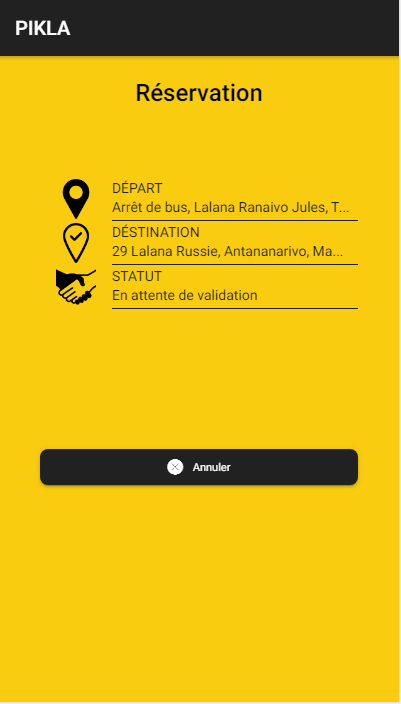
Position du Taxi



Confirmation pour commander le Taxi

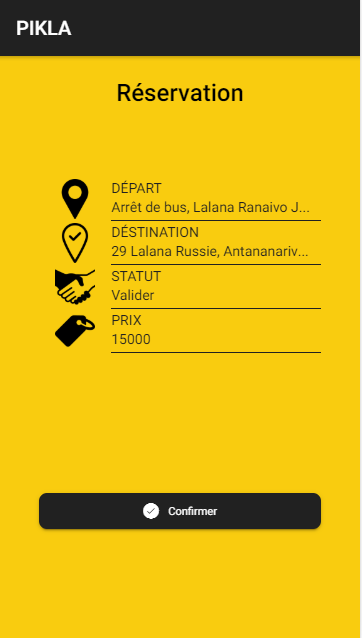
Si le client clique sur « NON », il pourrait choisir à nouveau un autre taxi disponible. Mais sinon, il serait redirigé vers une page « Réservation » qui affiche le détail de son trajet.

Une demande de réservation serait envoyée au taxieur et le statut de réservation est « En attente de validation ». C’est-à-dire que le taxieur devrait valider ou refuser la demande selon son choix. Dans ce cas, le client pourrait aussi « Annuler »la demande de réservation avant que le taxieur confirme.



Réservation en attente de validation du taxieur

Quand le taxieur valide la demande, il devrait mentionner le prix de la course et attend à son tour la confirmation du client.



Réservation en attente de Confirmation du client

Lorsque le Client confirme la réservation, il pourrait visualise la distance et la durée d’approche du Taxi en temps réel, et pourrait aussi voir l’aperçu dans la carte.



Approche du taxi

CHAPITRE 6 : EXTRAITS DE CODE

* + 1. GESTION DES PACKAGES ET FICHIER DE CONFIGURATION
    2. Gestion des packages

Voici le contenu du fichier package.json

{

"name": "pikla",

"version": "1.0",

"author": "Eqima",

"homepage": "http://www.eqima.org",

"private": true,

"scripts": {

"start": "ionic-app-scripts serve",

"clean": "ionic-app-scripts clean",

"build": "ionic-app-scripts build",

"lint": "ionic-app-scripts lint"

},

"dependencies": {

"@agm/core": "^1.0.0-beta.5",

"@angular/animations": "5.2.11",

"@angular/common": "5.2.11",

"@angular/compiler": "5.2.11",

"@angular/compiler-cli": "5.2.11",

"@angular/core": "5.2.11",

"@angular/fire": "^5.2.3",

"@angular/forms": "5.2.11",

"@angular/http": "5.2.11",

"@angular/platform-browser": "5.2.11",

"@angular/platform-browser-dynamic": "5.2.11",

"@ionic-native/core": "^4.16.0",

"@ionic-native/facebook": "^5.17.1",

"@ionic-native/geolocation": "^5.5.1",

"@ionic-native/google-maps": "^4.14.0",

"@ionic-native/splash-screen": "~4.20.0",

"@ionic-native/status-bar": "~4.20.0",

"@ionic/storage": "2.2.0",

"cordova-android": "7.0.0",

"cordova-browser": "6.0.0",

"cordova-plugin-compat": "1.2.0",

"cordova-plugin-facebook4": "^6.2.0",

"cordova-plugin-geolocation": "4.0.1",

"cordova-plugin-googlemaps": "2.6.2",

"firebase": "^7.5.0",

"install": "^0.13.0",

"ionic-angular": "3.9.5",

"ionicons": "3.0.0",

"npm": "^6.13.1",

"promise-polyfill": "^8.1.3",

"react-native-image-picker": "git+https://github.com/abartolo/react-native-image-picker#UpgradeToRNv56",

"rxjs": "5.5.11",

"sw-toolbox": "3.6.0",

"tslib": "^1.10.0",

"typescript": "^3.6.2",

"zone.js": "0.8.29"

},

"devDependencies": {

"@ionic/app-scripts": "3.2.2",

"@types/google-maps": "^3.2.1",

"cordova-plugin-device": "2.0.2",

"cordova-plugin-ionic-keyboard": "^2.0.5",

"cordova-plugin-ionic-webview": "^4.0.0",

"cordova-plugin-splashscreen": "^5.0.2",

"cordova-plugin-statusbar": "2.4.2",

"cordova-plugin-whitelist": "^1.3.3"

},

"description": "An Ionic project",

"cordova": {

"plugins": {

"cordova-plugin-geolocation": {

"GEOLOCATION\_USAGE\_DESCRIPTION": "To locate you"

},

"cordova-plugin-whitelist": {},

"cordova-plugin-statusbar": {},

"cordova-plugin-device": {},

"cordova-plugin-splashscreen": {},

"cordova-plugin-ionic-webview": {

"ANDROID\_SUPPORT\_ANNOTATIONS\_VERSION": "27.+"

},

"cordova-plugin-ionic-keyboard": {},

"cordova-plugin-facebook4": {

"APP\_ID": "945275605845214",

"APP\_NAME": "PIKLA",

"FACEBOOK\_HYBRID\_APP\_EVENTS": "false",

"FACEBOOK\_ANDROID\_SDK\_VERSION": "5.5.2"

},

"cordova-plugin-googlemaps": {

"PLAY\_SERVICES\_VERSION": "15.0.1",

"ANDROID\_SUPPORT\_V4\_VERSION": "25.+"

},

"cordova-plugin-compat": {}

},

"platforms": [

"browser",

"android"

]

}

}

* + 1. Fichiers de configuration

Le contenu du fichier de **config.xml** est la suivante :

<?xml version='1.0' encoding='utf-8'?>

<widget id="com.pikla.tracker" version="0.0.1" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">

<name>Pikla</name>

<description>Réservation d'un Taxi</description>

<author email="support@eqima.org" href="http://www.eqima.org/">Eqima</author>

<content src="index.html" />

<access origin="\*" />

<allow-intent href="http://\*/\*" />

<allow-intent href="https://\*/\*" />

<allow-intent href="tel:\*" />

<allow-intent href="sms:\*" />

<allow-intent href="mailto:\*" />

<allow-intent href="geo:\*" />

<preference name="ScrollEnabled" value="false" />

<preference name="android-minSdkVersion" value="19" />

<preference name="BackupWebStorage" value="none" />

<preference name="SplashMaintainAspectRatio" value="true" />

<preference name="FadeSplashScreenDuration" value="300" />

<preference name="SplashShowOnlyFirstTime" value="false" />

<preference name="SplashScreen" value="screen" />

<preference name="SplashScreenDelay" value="3000" />

<preference name="android-targetSdkVersion" value="25" />

<preference name="GOOGLE\_MAPS\_ANDROID\_API\_KEY" value="AIzaSyDYAs15UwMEkhXAhBqRzoQET2b3aVy4JgM" />

<platform name="android">

<allow-intent href="market:\*" />

<icon density="ldpi" src="resources/android/icon/drawable-ldpi-icon.png" />

<icon density="mdpi" src="resources/android/icon/drawable-mdpi-icon.png" />

<icon density="hdpi" src="resources/android/icon/drawable-hdpi-icon.png" />

<icon density="xhdpi" src="resources/android/icon/drawable-xhdpi-icon.png" />

<icon density="xxhdpi" src="resources/android/icon/drawable-xxhdpi-icon.png" />

<icon density="xxxhdpi" src="resources/android/icon/drawable-xxxhdpi-icon.png" />

<splash density="land-ldpi" src="resources/android/splash/drawable-land-ldpi-screen.png" />

<splash density="land-mdpi" src="resources/android/splash/drawable-land-mdpi-screen.png" />

<splash density="land-hdpi" src="resources/android/splash/drawable-land-hdpi-screen.png" />

<splash density="land-xhdpi" src="resources/android/splash/drawable-land-xhdpi-screen.png" />

<splash density="land-xxhdpi" src="resources/android/splash/drawable-land-xxhdpi-screen.png" />

<splash density="land-xxxhdpi" src="resources/android/splash/drawable-land-xxxhdpi-screen.png" />

<splash density="port-ldpi" src="resources/android/splash/drawable-port-ldpi-screen.png" />

<splash density="port-mdpi" src="resources/android/splash/drawable-port-mdpi-screen.png" />

<splash density="port-hdpi" src="resources/android/splash/drawable-port-hdpi-screen.png" />

<splash density="port-xhdpi" src="resources/android/splash/drawable-port-xhdpi-screen.png" />

<splash density="port-xxhdpi" src="resources/android/splash/drawable-port-xxhdpi-screen.png" />

<splash density="port-xxxhdpi" src="resources/android/splash/drawable-port-xxxhdpi-screen.png" />

</platform>

<platform name="ios">

<allow-intent href="itms:\*" />

<allow-intent href="itms-apps:\*" />

<icon height="57" src="resources/ios/icon/icon.png" width="57" />

<icon height="114" src="resources/ios/icon/icon@2x.png" width="114" />

<icon height="40" src="resources/ios/icon/icon-40.png" width="40" />

<icon height="80" src="resources/ios/icon/icon-40@2x.png" width="80" />

<icon height="120" src="resources/ios/icon/icon-40@3x.png" width="120" />

<icon height="50" src="resources/ios/icon/icon-50.png" width="50" />

<icon height="100" src="resources/ios/icon/icon-50@2x.png" width="100" />

<icon height="60" src="resources/ios/icon/icon-60.png" width="60" />

<icon height="120" src="resources/ios/icon/icon-60@2x.png" width="120" />

<icon height="180" src="resources/ios/icon/icon-60@3x.png" width="180" />

<icon height="72" src="resources/ios/icon/icon-72.png" width="72" />

<icon height="144" src="resources/ios/icon/icon-72@2x.png" width="144" />

<icon height="76" src="resources/ios/icon/icon-76.png" width="76" />

<icon height="152" src="resources/ios/icon/icon-76@2x.png" width="152" />

<icon height="167" src="resources/ios/icon/icon-83.5@2x.png" width="167" />

<icon height="29" src="resources/ios/icon/icon-small.png" width="29" />

<icon height="58" src="resources/ios/icon/icon-small@2x.png" width="58" />

<icon height="87" src="resources/ios/icon/icon-small@3x.png" width="87" />

<icon height="1024" src="resources/ios/icon/icon-1024.png" width="1024" />

<splash height="1136" src="resources/ios/splash/Default-568h@2x~iphone.png" width="640" />

<splash height="1334" src="resources/ios/splash/Default-667h.png" width="750" />

<splash height="2208" src="resources/ios/splash/Default-736h.png" width="1242" />

<splash height="1242" src="resources/ios/splash/Default-Landscape-736h.png" width="2208" />

<splash height="1536" src="resources/ios/splash/Default-Landscape@2x~ipad.png" width="2048" />

<splash height="2048" src="resources/ios/splash/Default-Landscape@~ipadpro.png" width="2732" />

<splash height="768" src="resources/ios/splash/Default-Landscape~ipad.png" width="1024" />

<splash height="2048" src="resources/ios/splash/Default-Portrait@2x~ipad.png" width="1536" />

<splash height="2732" src="resources/ios/splash/Default-Portrait@~ipadpro.png" width="2048" />

<splash height="1024" src="resources/ios/splash/Default-Portrait~ipad.png" width="768" />

<splash height="960" src="resources/ios/splash/Default@2x~iphone.png" width="640" />

<splash height="480" src="resources/ios/splash/Default~iphone.png" width="320" />

<splash height="2732" src="resources/ios/splash/Default@2x~universal~anyany.png" width="2732" />

</platform>

<plugin name="cordova-plugin-whitelist" spec="1.3.3" />

<plugin name="cordova-plugin-statusbar" spec="2.4.2" />

<plugin name="cordova-plugin-device" spec="2.0.2" />

<plugin name="cordova-plugin-splashscreen" spec="5.0.2" />

<plugin name="cordova-plugin-ionic-webview" spec="^4.0.0" />

<plugin name="cordova-plugin-ionic-keyboard" spec="^2.0.5" />

</widget>

Ci apres le contenu du fichier **tsconfig.json**

{

"compilerOptions": {

"allowSyntheticDefaultImports": true,

"declaration": false,

"emitDecoratorMetadata": true,

"experimentalDecorators": true,

"lib": [

"dom",

"es2015"

],

"module": "es2015",

"moduleResolution": "node",

"sourceMap": true,

"target": "es5"

},

"include": [

"src/\*\*/\*.ts"

],

"exclude": [

"node\_modules",

"src/\*\*/\*.spec.ts",

"src/\*\*/\_\_tests\_\_/\*.ts"

],

"compileOnSave": false,

"atom": {

"rewriteTsconfig": false

}

}

* + 1. COMPONENT PRINCIPAL

1. app.component.ts

import { Component, ViewChild } from '@angular/core';

import { Nav, NavController, Platform } from 'ionic-angular';

import { StatusBar } from '@ionic-native/status-bar';

import { SplashScreen } from '@ionic-native/splash-screen';

import { Environment } from '@ionic-native/google-maps';

import { LoginPage } from '../pages/login/login';

@Component({

templateUrl: 'app.html'

})

export class MyApp {

rootPage:any = LoginPage;

@ViewChild(Nav) nav: Nav;

pages: Array<{title: string, component: any}>;

constructor(platform: Platform, statusBar: StatusBar, splashScreen: SplashScreen) {

this.pages = [];

platform.ready().then(() => {

// Okay, so the platform is ready and our plugins are available.

// Here you can do any higher level native things you might need.

statusBar.styleDefault();

splashScreen.hide();

});

}

goTo(page){

this.nav.setRoot(page.component);

}

}

1. app.module.ts

import { BrowserModule } from '@angular/platform-browser';

import { ErrorHandler, NgModule } from '@angular/core';

import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';

import { SplashScreen } from '@ionic-native/splash-screen';

import { StatusBar } from '@ionic-native/status-bar';

import { MyApp } from './app.component';

import { HomePage } from '../pages/home/home';

import { LoginPage } from '../pages/login/login';

import { MapPage } from '../pages/map/map';

import { SearchPage } from '../pages/search/search';

import { ConnectedPage } from '../pages/connected/connected';

import { Facebook } from '@ionic-native/facebook/ngx';

import { FIREBASE\_CONFIG } from './app.firebase.config';

import { GoogleMaps } from '@ionic-native/google-maps';

import { Geolocation } from '@ionic-native/geolocation/ngx';

import { AngularFireModule } from '@angular/fire';

import { AngularFirestoreModule } from '@angular/fire/firestore';

import { PiklaPage } from '../pages/pikla/pikla';

import { BookingPage } from '../pages/booking/booking';

import { AngularFireDatabaseModule } from '@angular/fire/database';

import { FirebaseProvider } from '../providers/firebase/firebase';

@NgModule({

declarations: [

MyApp,

HomePage,

LoginPage,

MapPage,

SearchPage,

ConnectedPage,

PiklaPage,

BookingPage

],

imports: [

BrowserModule,

AngularFireDatabaseModule,

AngularFireModule.initializeApp(FIREBASE\_CONFIG),

AngularFirestoreModule.enablePersistence(),

IonicModule.forRoot(MyApp)

],

bootstrap: [IonicApp],

entryComponents: [

MyApp,

HomePage,

LoginPage,

MapPage,

SearchPage,

ConnectedPage,

PiklaPage,

BookingPage

],

providers: [

StatusBar,

SplashScreen,

Facebook,

GoogleMaps,

Geolocation,

{provide: ErrorHandler, useClass: IonicErrorHandler},

FirebaseProvider

]

})

export class AppModule {}

1. app.scss

// http://ionicframework.com/docs/theming/

// App Global Sass

// --------------------------------------------------

// Put style rules here that you want to apply globally. These

// styles are for the entire app and not just one component.

// Additionally, this file can be also used as an entry point

// to import other Sass files to be included in the output CSS.

//

// Shared Sass variables, which can be used to adjust Ionic's

// default Sass variables, belong in "theme/variables.scss".

//

// To declare rules for a specific mode, create a child rule

// for the .md, .ios, or .wp mode classes. The mode class is

// automatically applied to the <body> element in the app.

.text-input-md {

font-size: 12px;

}

.toolbar-background-ios, .toolbar-background-md

{

background-color: #212121 !important ;

}

.content-md {

color: #000;

background-color: #f9cc0f;

}

.bar-button-default-md, .bar-button-clear-md-default, .bar-button-md-default {

color: #fff !important;

}

ion-app.\_gmaps\_cdv\_ .nav-decor {

background-color: transparent !important;

}

#map\_canvas {

height: 100%;

position: relative;

width: 100%;

}

.toolbar-title-md {

padding: 0 12px;

font-size: 2rem;

font-weight: 500;

color: #ffffff;

}

.item-input .label-md, .item-select .label-md, .item-datetime .label-md {

color: #212121;

}

* + 1. PROVIDER

Voici le contenu du fichier **firebase.ts** :

import { Injectable } from '@angular/core';

import { AngularFireDatabase } from '@angular/fire/database';

@Injectable()

export class FirebaseProvider {

constructor(public firebase: AngularFireDatabase) {

}

fetchById(path:string, id:string){

return new Promise((resolve)=>{

this.firebase.object(`${path}/${id}`).valueChanges().subscribe((object)=>{

resolve(object);

})

})

}

fetchAll(path){

return new Promise((resolve)=>{

this.firebase.object(path).valueChanges().subscribe((list)=>{

resolve(list);

})

})

}

save(path:string,object:any, key?:string){

if (key) {

return this.firebase.list(path).update(key,object);

} else {

return this.firebase.list(path).push(object).key

}

}

getById(path:string, id:string){

return this.firebase.object(`${path}/${id}`).valueChanges();

}

delete(path:string, key:string){

this.firebase.list(path).remove(key);

}

}

* + 1. FONCTIONNALITES

1. Récupération de la position actuelle

getCurentPosition(){

return new Promise((resolve)=>{

let watch = this.geolocation.watchPosition();

watch.subscribe((data) => {

let position = {

lat: data.coords.latitude,

lng: data.coords.longitude

};

resolve(position);

});

})

}

1. Affichage de la carte

initMap(position){

this.loadMap(position)

.then((loaded)=>{

if (loaded) {

this.addMarker(position,"source");

this.addMarker(this.target,"destination");

this.traceRoute(position,this.target).then((response:any) => {

this.addPolylines(response).then((target) => {

this.moveCamera(target);

})

})

}

})

.then(() => {

this.firebasePrvd.fetchAll('taxi').then((data:any)=>{

for(let key in data){

let pikla = data[key];

let coords = new LatLng(pikla.coords.lat,pikla.coords.lng);

if (pikla.status == 0) {

this.addMarker(coords,"pikla",key);

}

}

})

})

}

loadMap(position){

return new Promise((resolve) => {

let mapOptions: GoogleMapOptions = {

camera: {

target: position,

zoom: 18,

tilt: 30

},

controls: {

compass: false,

mapToolbar: false

}

};

this.map = GoogleMaps.create('map\_canvas', mapOptions);

this.map.clear();

this.map.one(GoogleMapsEvent.MAP\_READY).then(() => {

resolve(true);

})

})

}

addMarker(pos, type?:string,key?:string){

let url = "";

let title = "";

if (type == "source") {

url = './assets/icon/pin-source.png';

title = "Mon position"

} else {

if (type == "destination") {

url = './assets/icon/pin-destination.png';

title = "Déstination"

} else{

if (type == "pikla") {

url = './assets/icon/taxi.png';

title = "TAXI";

}

}

}

let icon : MarkerIcon = {

url: url ,

size: {

width: 40,

height: 40

}

}

let position = new LatLng(pos.lat,pos.lng);

let marker: MarkerOptions = {

position: {

lat: position.lat,

lng: position.lng

},

icon:icon,

title: title

}

this.map.addMarker(marker).then((markerObject)=>{

if (type == "pikla") {

markerObject.on(GoogleMapsEvent.MARKER\_CLICK).subscribe((marker) => {

this.selectedPikla = key;

console.log(marker);

this.clickPiklaMarker(marker);

});

}

})

}

1. Détermination de la durée et distance d’un trajet

getTravelDetails(origin,destination){

return new Promise((resolve)=>{

var service = new google.maps.DistanceMatrixService();

service.getDistanceMatrix(

{

origins: [origin],

destinations: [destination],

travelMode: google.maps.TravelMode.DRIVING

},function(response,status) {

let travelDetailsObject;

if (status == 'OK') {

var origins = response.originAddresses;

var destinations = response.destinationAddresses;

for (var i = 0; i < origins.length; i++) {

var results = response.rows[i].elements;

for (var j = 0; j < results.length; j++) {

var element = results[j];

var distance = element.distance.text;

var duration = element.duration.text;

var from = origins[i];

var to = destinations[j];

travelDetailsObject = {

distance: distance,

duration: duration

}

}

}

resolve(travelDetailsObject);

}

});

})

}

## CHAPITRE 7 : APPROCHE QUALITATIVE DU PROJET

### LA GESTION QUALITES

#### Définition de la qualité

La qualité est l’ensemble des caractères, des propriétés qui font que quelque chose, matérielle ou immatérielle, corresponde d’une bonne ou d’une mauvaise manière aux attentes qui la concernent.

Dans le contexte d’une application mobile, la qualité est la capacité de l’application à effectuer de manière fiable et sécurisée sa fonction principale ainsi que l’ensemble des critères visuels ou non qui rendent son utilisation plus agréable.

Généralement, la qualité d’une application correspond à sa capacité à satisfaire les clients et à être conforme aux attentes des clients. Mais la qualité d’un logiciel ou d’une application peut être mesurée plus précisément par les six caractéristiques définit par la norme ISO 9126 :

- La capacité fonctionnelle, c’est-à-dire la capacité qu’ont les fonctionnalités d’un logiciel à répondre aux exigences et besoins explicites ou implicites des usagers ;

- La facilité d’utilisation, qui porte sur l’effort nécessaire pour apprendre à manipuler le logiciel ;

- La fiabilité, c’est-à-dire la capacité d’un logiciel de rendre des résultats corrects quelles que soient les conditions d’exploitation. En fait partie la tolérance aux pannes ;

- La performance, c’est-à-dire le rapport entre la quantité de ressources utilisées (moyens matériels, temps, personnel), et la quantité de résultats délivrés ;

- La maintenabilité, qui mesure l’effort nécessaire à corriger ou transformer le logiciel pour y ajouter de nouvelles fonctions ;

- La portabilité, c’est-à-dire l’aptitude d’un logiciel de fonctionner dans un environnement matériel ou logiciel différent de son environnement initial.

1. L’importance du projet

Notre application est destinée, à la fois, aux clients et aux chauffeurs de taxis possédant un smartphone sous Android. En résumé, l’application côté client permet la réservation de taxis en temps réel après une géolocalisation sur la carte.

L’application côté chauffeur permet de recevoir les demandes de réservation et d’afficher la position du client ainsi que sa destination, le chauffeur de taxi indique le prix au client dans le cas où il accepte la demande, et précisera s’il est libre dans l’immédiat ou non en indiquant quand il le sera, et dans le cas où il décide de ne pas accepter la demande il l’annule. A son tour le client aura la possibilité d’accepter ou de refuser le prix indiqué par le taxieur si ce dernier ne lui convient pas.

Avec notre application les clients et les taxieurs gagnent du temps, de faciliter avec une solution rapide et fiable.

1. Qualité technique du projet
   1. Qualité externe

La qualité externe est la qualité visuelle de l’interface ainsi que sa facilité d’utilisation. Cela inclut le design général de l’application, le choix des couleurs, la disposition des éléments. Le design doit rendre l’application intuitive pour que l’utilisateur sache ce qu’il doit faire et ce qu’il peut faire lorsque l’application lui est présentée.

Tout d’abord, l’utilisateur ne va pas perdre son temps à remplir des formulaires pour l’inscription. Notre système d’authentification est simplement basé sur l’utilisation d’un compte facebook, qui est classé le numéro un mondial en terme de réseau sociaux et qui possède une API très puissante qui nous permet de récupérer les informations nécessaires du client et d’assurer la sécurité du système.

De plus, l’utilisateur ne risque pas de se perdre dans l’application ni de s’ennuyer pendant l’utilisation, parce que c’est simple et facile à utiliser. Seulement deux Boutons seront proposés dans le menu principal après avoir été authentifié.

Mais aussi, l’utilisation de IONIC à donner à notre application un style d’interface simple et intuitive.

* 1. Qualité interne

La qualité interne est la capacité du produit à assumer sa fonction principale et son niveau de performance. Cette qualité peut être indiquée par la capacité fonctionnelle, la fiabilité, la performance, la maintenabilité, la portabilité.

La fonction principale de l’application, qui est la réservation d’un taxi, est opérationnelle. La capacité fonctionnelle est donc satisfaite.

Concernant la performance, Firebase assure un temps de réponse rapide et en temps réel. Ce qui permet à l’application d’afficher rapidement les résultats voulus.

1. Futures améliorations

Plusieurs améliorations de l’application sont encore envisageables :

* L’intégration de module de chat, appel et message pourrait être très utile si par exemple le client veut négocier avec le taxieur en terme de prix ou d’autre question indispensable.
* Détection des embouteillages et meilleur proposition d’itinéraire
* Délimiter l’usage aux taxieurs, c’est-à-dire de mettre en place la possibilité de trouver des livreurs, des voitures de location, des taxi-moto et même l’utilisation des transports en commun et les taxis brousse.
* On peut aussi envisager un système de paiement simple et rapide en utilisant la solution du money banking ou même de concevoir notre propre système de transaction en ligne.

# CONCLUSION

Le but de projet était de concevoir une application mobile qui permet de faciliter la recherche des bus ainsi que la navigation dans une ville.

Ce projet peut apporter un gain de temps et de sécurité aux utilisateurs qui souhaitent trouver des bus dans les lieux moins connus ou simplement s’ils veulent connaitre les bus qui vont passer sur le lieu.

En effet, l’application offre des positions fiables et en temps réelles. Mais cette fiabilité dépend aussi de la qualité du module GPS du smartphone.

En somme, la réalisation de ce mémoire et de ce stage au sein de la société EQIMA, nous a permis d’acquérir beaucoup d’expériences au niveau théorique mais surtout au niveau professionnel car cela enrichi nos compétences et de nos connaissances.

Pour conclure, nous espérons que la collaboration entre l’ISPM et la société EQIMA se développera toujours de plus en plus pour un avenir meilleur.

# ANNEXES

# BIBLIOGRAPHIE

### Cours dispensés à l’ISPM

1. Professeur RABOANARY Julien Amédée, Gestion de qualité, Management, 2014-2015
2. Professeur RABOANARY Herinaina Andry, Structures de données, 2013-2014
3. Professeur RABOANARY Toky Hajatiana, C#, 2015-2017
4. Professeur RAKOTONDRAHAJA Clarel : SGBDR, COO, 2015-2017
5. ProfesseurRAKOTOARIMANANAAndriambao Johnson : UML, 2015-2017
6. Professeur RASANDIMANANA Tanjona Tsihoarana, SQL, 2012-2013

### Revues et documentations

1. : David Autissier & Valérie Delaye, Mesurer la performance du système d’information, Éditions d’organisation groupe Eyrolles, 2008.
2. : Introduction à Visual C# et Visual Studio 2012
3. : SQL Server 2012 - Administration d'une base de données transactionnelle, Éditions ENI

### Webographies

1. : [http://www.ispm-edu.com](http://www.ispm-edu.com/)
2. : <http://www.eqima.mg>
3. : <http://ecoaustral.com/mbolatiana-andriamanantsoa-le-secret-dune-reussite>
4. : http://www.fid.mg/presentation/
5. : <https://fr.wikipedia.org/wiki/Gestion_des_ressources_humaines>
6. : <https://fr.wikipedia.org/wiki/Microsoft_SQL_Server>
7. : <https://msdn.microsoft.com/fr-fr/>
8. : <https://www.sage.com/>
9. : https://openclassrooms.com/fr/courses/218202-apprenez-a-programmer-en-c-sur-net
10. : https://sqlserver.developpez.com/cours/
11. : <https://jmbruel.github.io/MobileModeling2016-1/>

# TABLE DES MATIERES

[REMERCIEMENTS I](#_Toc9399996)

[AVANT PROPOS II](#_Toc9399997)

[SOMMAIRE III](#_Toc9399998)

[LISTE DES FIGURES IV](#_Toc9399999)

[LISTE DES TABLEAUX VII](#_Toc9400000)

[LISTE DES ABREVIATIONS VIII](#_Toc9400001)

[INTRODUCTION 1](#_Toc9400002)

[PARTIE 1 1](#_Toc9400003)

[DOMAINE D’ETUDE ET DE TRAVAIL 1](#_Toc9400004)

[CHAPITRE 1 : PRESENTATION DE L’ISPM 2](#_Toc9400005)

[A. HISTORIQUE 2](#_Toc9400006)

[1. Le département Informatique et Télécommunications : 2](#_Toc9400007)

[2. Le département Techniques des Affaires : 2](#_Toc9400008)

[3. Le département Biotechnologique et Agronomie : 3](#_Toc9400009)

[4. Le département Génie Industriel et Génie Civil : 3](#_Toc9400010)

[5. Le département Techniques du Tourisme : 3](#_Toc9400011)

[B. PRESENTATION 3](#_Toc9400012)

[1. Objectif 3](#_Toc9400013)

[2. Identification 3](#_Toc9400014)

[a. Siege 3](#_Toc9400015)

[b. Site web 4](#_Toc9400016)

[c. Devise 4](#_Toc9400017)

[d. Logo 4](#_Toc9400018)

[e. Hymne 5](#_Toc9400019)

[f. Hymne du 25ème anniversaire de l’ISPM 6](#_Toc9400020)

[3. Etudes et formations 7](#_Toc9400021)

[a. Premier cycle 7](#_Toc9400022)

[b. Second cycle 7](#_Toc9400023)

[c. Formation du troisième cycle 7](#_Toc9400024)

[d. Cursus de l’ISPM 8](#_Toc9400025)

[4. Organigramme de l’ISPM 9](#_Toc9400026)

[5. Prix obtenue par l’ISPM 10](#_Toc9400027)

[CHAPITRE 2 : PRESENTATION DE L’ENTREPRISE 12](#_Toc9400028)

[A. HISTORIQUE 12](#_Toc9400029)

[1. Logo de l’Entreprise 12](#_Toc9400030)

[2. Site web de l’EQIMA : 13](#_Toc9400031)

[B. ByEQIMA: 13](#_Toc9400032)

[C. Organigramme de l’EQIMA 14](#_Toc9400033)

[PARTIE 2 15](#_Toc9400034)

[METHODES ET OUTILS 15](#_Toc9400035)

[CHAPITRE 3 : METHODES 15](#_Toc9400036)

[A. UML (Unified Modeling Language) 15](#_Toc9400037)

[1. Introduction 15](#_Toc9400038)

[2. Diagrammes 15](#_Toc9400039)

[a. Diagrammes structurels ou diagrammes statiques (UML Structure): 16](#_Toc9400040)

[3. Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior): 24](#_Toc9400041)

[B. La méthode SCRUM 35](#_Toc9400042)

[CHAPITRE 4 : OUTILS DE CONCEPTION 40](#_Toc9400043)

[A. IONIC FRAMEWORK 40](#_Toc9400044)

[1. FRAMEWORK 40](#_Toc9400045)

[a. Définition 40](#_Toc9400046)

[b. Types de framework 41](#_Toc9400047)

[c. Avantage d’un « Framework » 41](#_Toc9400048)

[2. GENERALISATION 43](#_Toc9400049)

[1. Historique : 43](#_Toc9400050)

[2. Définition : 43](#_Toc9400051)

[3. Objectifs 43](#_Toc9400052)

[a. Multiplateforme (Cross-platform) 43](#_Toc9400053)

[b. Basé sur des standards web (Web standards-based) 44](#_Toc9400054)

[c. Beau design (Beautiful Design) 44](#_Toc9400055)

[d. Simplicité (Simplicity) 44](#_Toc9400056)

[e. License 44](#_Toc9400057)

[f. IONIC CLI 44](#_Toc9400058)

[g. STRUCTURE 45](#_Toc9400059)

[4. ARCHITECTURE MVC 45](#_Toc9400060)

[a. Le modèle: 45](#_Toc9400061)

[b. La vue: 45](#_Toc9400062)

[c. Le contrôleur: 46](#_Toc9400063)

[5. LANGAGES DE PROGRAMMATIONS 46](#_Toc9400064)

[a. HyperText Markup Langage (H.T.M.L) 46](#_Toc9400065)

[b. Définition 47](#_Toc9400066)

[c. Structure et syntaxe 47](#_Toc9400067)

[6. TYPESCRIPT 48](#_Toc9400068)

[a. Présentation 48](#_Toc9400069)

[b. SCSS  ou SASS: 50](#_Toc9400070)

[B. FIREBASE 51](#_Toc9400071)

[1. Logo 51](#_Toc9400072)

[2. Définition 51](#_Toc9400073)

[3. NoSQL 52](#_Toc9400074)

[4. Les services de Firebase 52](#_Toc9400075)

[C. API GOOGLE 53](#_Toc9400076)

[1. Description d’une API 53](#_Toc9400077)

[2. GOOGLE MAPS PLATFORME 53](#_Toc9400078)

[a. MAPS 53](#_Toc9400079)

[b. ROUTES 54](#_Toc9400080)

[c. PLACES 54](#_Toc9400081)

[D. ADOBE PHOTOSHOP 55](#_Toc9400082)

[E. TRELLO 56](#_Toc9400083)

[F. GIT 56](#_Toc9400084)

[1. Définition 57](#_Toc9400085)

[2. Particularités techniques 57](#_Toc9400086)

[3. Fonctionnement 58](#_Toc9400087)

[4. GIT BLIT 59](#_Toc9400088)

[PARTIE 3 15](#_Toc9400089)

[RESULTATS 15](#_Toc9400090)

[CHAPITRE 5 : PRESENTATION DE L’APPLICATION 57](#_Toc9400091)

[A. GENERALITE 57](#_Toc9400092)

[B. UTILISATEURS 57](#_Toc9400093)

[C. CONFIGURATION REQUISE 57](#_Toc9400094)

[D. CAPTURE D’ECRAN 57](#_Toc9400095)

[1. Côté utilisateur 57](#_Toc9400096)

[a. Page d’accueil 58](#_Toc9400097)

[b. Détermination de la position actuelle 58](#_Toc9400098)

[c. Formulaire de destination 59](#_Toc9400099)

[d. Traçage de la route 60](#_Toc9400100)

[e. Suivi des bus 61](#_Toc9400101)

[2. Côté chauffeur 62](#_Toc9400102)

[a. Page d’accueil 62](#_Toc9400103)

[b. Application est en marche 63](#_Toc9400104)

[E. EXTRAIT DE CODE 64](#_Toc9400105)

[1. src/app 65](#_Toc9400106)

[a. app.component.ts 65](#_Toc9400107)

[b. app.html 65](#_Toc9400108)

[c. app.module.ts 66](#_Toc9400109)

[d. app.scss 66](#_Toc9400110)

[2. src/pages/map 67](#_Toc9400111)

[a. map.html 67](#_Toc9400112)

[b. map.module.ts 67](#_Toc9400113)

[c. map.scss 68](#_Toc9400114)

[3. src/classes 68](#_Toc9400115)

[a. address.class.ts 68](#_Toc9400116)

[b. coordinate.class.ts 69](#_Toc9400117)

[c. google-geolocation.class.ts 69](#_Toc9400118)

[4. config.xml 70](#_Toc9400119)

[5. package.json 70](#_Toc9400120)

[6. tsconfig.json 71](#_Toc9400121)

[PARTIE 4 57](#_Toc9400122)

[PRESENTATION GENERALE 57](#_Toc9400123)

[CHAPITRE 6 : DISCUSSIONS 72](#_Toc9400124)

[A. OUTILS DE MODELISATION 72](#_Toc9400125)

[1. Les diagrammes UML 72](#_Toc9400126)

[2. Modélisation du système 72](#_Toc9400127)

[a. Diagramme de classe 73](#_Toc9400128)

[b. Diagramme d’activité 73](#_Toc9400129)

[c. Diagramme de cas d’utilisation 74](#_Toc9400130)

[d. Diagramme de déploiement 75](#_Toc9400131)

[B. ETAT DU PROJET 76](#_Toc9400132)

[1. MISE EN PLACE 76](#_Toc9400133)

[2. BILAN PROJET 76](#_Toc9400134)

[C. LA GESTION QUALITES 77](#_Toc9400135)

[1. Définition de la qualité 77](#_Toc9400136)

[CHAPITRE 7 : AMELIORATION FUTURE 78](#_Toc9400137)

[CONCLUSION 79](#_Toc9400138)

[ANNEXES 80](#_Toc9400139)

[BIBLIOGRAPHIE 81](#_Toc9400140)

[A. Cours dispensés à l’ISPM 81](#_Toc9400141)

[B. Revues et documentations 81](#_Toc9400142)

[C. Webographies 81](#_Toc9400143)

[TABLE DES MATIERES 82](#_Toc9400144)