# **CURRICULUM VITAE**

**ETAT CIVILE**

LOVAHARITIANA 

Hoby Olivia

Lot 29 SS Sahalava

Fianarantsoa 301

Née le 11 Novembre 1999 à Tambohobe Fianarantsoa

E-mail : [hobyolivia19@gmail.com](mailto:hobyolivia19@gmail.com)

Tél : 0342314175

**FORMATIONS ET DIPLOMES OBTENUS**

2021-2022 : Etudiante en Troisième année de formation en Licence Professionnelle à l’Ecole Nationale d’Informatique, Université de Fianarantsoa.

*Parcours :* Génie Logiciel et Base de Données.

2020-2021 : Etudiante en Deuxième année de formation en Licence Professionnelle à l’Ecole Nationale d’Informatique, Université de Fianarantsoa.

*Parcours :* Génie logiciel et Base de Données

2019-2020 : Etudiante en Première année de formation en Licence Professionnelle à l’Ecole Nationale d’Informatique, Université de Fianarantsoa.

*Parcours :* Génie logiciel et Base de Données

2018-2019 : Etudiante en Première année de formation en Licence Professionnelle au Faculté des sciences, Université de Fianarantsoa.

*Parcours :* MISS

2017-2018 : Classe Terminale C au Lycée Saint Joseph de Cluny Tambohobe Fianarantsoa.

Obtention du diplôme : BACCALAUREAT de mention Assez-Bien.

**STAGES ET EXPERIENCES PROFESSIONNELLES**

**2021-2022 :** Projet d’étude à l’Ecole Nationale d’Informatique

*Thème :* -Conception et réalisation d’une application web de « Gestion des approvisionnements des produits » avec JSP, NodeJs.

- Conception et réalisation d’une application desktop de « Gestion des approvisionnements des produits » avec Java.

- Conception et réalisation d’une application mobile de « Gestion des commerces » avec Java.

**2020-2021 :** -Stagiaire au sein du Ministère de l’Intérieur et de la Décentralisation à Antananarivo.

*Thème :* Gestion de la maintenance des matériels d’un parc informatique avec PHP/Symfony.

*Durée :* 3 mois.

-Projets d’étude à l’Ecole Nationale d’Informatique

*Thème :* Conception et réalisation d’une application web de « Gestion des approvisionnements des produits » avec PHP/MySQL

**2019-2020 : -**Projets d’étude à l’Ecole Nationale d’Informatique

*Thème :* Création d’un logiciel « Gestion des appartements » avec Qt

-Création d’un site web « Ketrika Bacc » avec HTML/CSS

**COMPETENCES EN INFORMATIQUE**

Système d’exploitation : Windows, Linux

Langages de programmation : C/C++, Java, NodeJs, PHP

Framework/Librairies : Qt, Symfony 4, Bootstrap, JQuery

Méthodes de conception : MERISE, 2TUP

Langage de conception : UML

Système de gestion de Base de Données : MySQL, PostegreSQL

Système de gestion de versions : Git

Outils de bureautiques : Collection Microsoft office

**COMPETENCES LIGUISTIQUE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **APTITUDE A :** | | | | |
|  | **Comprendre à l’audition** | **Lire** | **Ecrire et rédiger** | **Parler et communiquer oralement** |
| **FRANCAIS** | TB | TB | B | B |
| **ANGLAIS** | B | B | AB | AB |

Grille d’évaluation :

TB : Très Bonne

B : Bonne

AB : Assez Bonne

**DIVERS :**

Activités sportives pratiquées : Athlétisme, Basket-ball

Loisir et passe-temps : Lecture, musique

# **SOMMAIRE GENERAL**

[CURRICULUM VITAE I](#_Toc123914552)

[SOMMAIRE GENERAL IV](#_Toc123914553)

[REMERCIEMENTS VI](#_Toc123914554)

[LISTE DES FIGURES VII](#_Toc123914555)

[LISTE DES TABLEAUX VIII](#_Toc123914556)

[LISTE DES ABREVIATIONS IX](#_Toc123914557)

[INTRODUCTION GENERALE 1](#_Toc123914558)

[PARTIE I. PRESENTATIONS 2](#_Toc123914559)

[Chapitre 1. Présentation de l’Ecole Nationale d’Informatique 3](#_Toc123914560)

[1. 1. Information d’ordre général 3](#_Toc123914561)

[1. 2. Missions et historique 3](#_Toc123914562)

[1. 3. Organigramme institutionnel de l’ENI 5](#_Toc123914563)

[1. 4. Domaine de spécialisation 7](#_Toc123914564)

[1. 5. Architecture des formations pédagogiques 8](#_Toc123914565)

[1. 6. Relations de l’ENI avec les entreprises et les organismes 10](#_Toc123914566)

[1. 7. Partenariat au niveau international 12](#_Toc123914567)

[1. 8. Débouchés professionnels avec des diplômés 14](#_Toc123914568)

[1. 9. Ressources humaines 16](#_Toc123914569)

[Chapitre 2. Présentation de la société d’accueil 17](#_Toc123914570)

[2.1. Fiche d’identification 17](#_Toc123914571)

[2.2. Brèves historiques 17](#_Toc123914572)

[2.3. Objectifs : Missions et Activités 17](#_Toc123914573)

[2.4. Organigramme de ARATO 18](#_Toc123914574)

[2.5. Ressources humaines 19](#_Toc123914575)

[2.6. Partenaire et Bienfaiteurs 19](#_Toc123914576)

[Chapitre 3. Description du projet 20](#_Toc123914577)

[3. 1. Formulation 20](#_Toc123914578)

[3. 2. Objectif et besoins de l’utilisateur 20](#_Toc123914579)

[3. 3. Moyens nécessaires à la réalisation du projet (Humain, matériel et éventuellement financier) 21](#_Toc123914580)

[3. 4. Résultats attendus 21](#_Toc123914581)

[PARTIE II. ANALYSE ET CONCEPTION 23](#_Toc123914582)

[Chapitre 4. Analyse préalable 24](#_Toc123914583)

[4. 1. Analyse de l’existant 24](#_Toc123914584)

[4. 2. Critique de l’existant 24](#_Toc123914585)

[4. 3. Conception avant-projet 25](#_Toc123914586)

[4. 4. Présentation de la méthode 2TUP 33](#_Toc123914587)

[Chapitre 5. Analyse conceptuelle 36](#_Toc123914588)

[5. 1. Dictionnaire des données 37](#_Toc123914589)

[5. 2. Règles de gestion 39](#_Toc123914590)

[5. 3. Représentation et spécification des besoins 40](#_Toc123914591)

[5. 4. Spécification des besoins techniques 50](#_Toc123914592)

[5. 5. Modélisation du domaine (Modèle de domaine) 50](#_Toc123914593)

[Chapitre 6. Conception détaillée (UML) 52](#_Toc123914594)

[6. 1. Architecture du système 52](#_Toc123914595)

[6. 2. Diagramme de séquence de conception pour chaque cas d’utilisation 53](#_Toc123914596)

[6. 3. Diagramme de classe de conception pour chaque cas d’utilisation 57](#_Toc123914597)

[6. 4. Diagramme de classe de conception global 59](#_Toc123914598)

[6. 5. Diagramme de paquetages 60](#_Toc123914599)

[6. 6. Diagramme de déploiement 60](#_Toc123914600)

[PARTIE III. REALISATION 61](#_Toc123914601)

[Chapitre 7. Mise en place de l’environnement de développement 62](#_Toc123914602)

[7 .1. Installation et configuration des outils 62](#_Toc123914603)

[7 .2. Architecture de l’application 65](#_Toc123914604)

[Chapitre 8. Développement de l’application 67](#_Toc123914605)

[8 .1. Création de la base de données 67](#_Toc123914606)

[8 .2. Codage de l’application 68](#_Toc123914607)

[8 .3. Présentation de l’application (Capture d’écran) 69](#_Toc123914608)

[CONCLUSION X](#_Toc123914609)

[BIBLIOGRAPHIE XI](#_Toc123914610)

[WEBOGRAPHIE XII](#_Toc123914611)

[GLOSSAIRE XIII](#_Toc123914612)

[TABLE DES MATIERES XIV](#_Toc123914613)

[RESUME XV](#_Toc123914614)

[ABSTRACT XV](#_Toc123914615)

# **REMERCIEMENTS**

Premièrement, je tiens à rendre grâce à Dieu pour la bénédiction qu’il m’a sans cesse comblée. Je tiens aussi à adresser mes sincères remerciements à tous ceux qui m’ont accompagné de loin ou de près à la réalisation de ce projet, notamment :

* Monsieur HAJALALAINA Aimé Richard, Professeur, Président de l’Université de Fianarantsoa, qui a bien organisé l’année universitaire.
* Monsieur MAHATODY Thomas, Professeur à l’Université de Fianarantsoa et Directeur de l’Ecole Nationale d’Informatique qui m’a donné l’opportunité de partir en stage afin de consolider mes connaissances et d’en développer de nouvelles.
* Monsieur ENCADREUR PEDAGOGIQUE, en tant qu’encadreur pédagogique pour son dévouement de nous avoir guidés et pour le temps qu’il a consacré jusqu’au cheminement de ce présent mémoire.
* Monsieur Francki Andrianirinarisoa, pour ses précieux conseils et assistance en tant qu’encadreur professionnel.
* Tous les membres du jury de me faire l’honneur d’examiner mon travail.
* Tous les enseignants de l’Ecole Nationale d’Informatique pour les connaissances qu’ils nous ont transmis durant notre formation théorique.

Je remercie également toute ma famille de m’avoir soutenu que ce soit moralement, matériellement et financièrement tout au long de ce stage.

# **LISTE DES FIGURES**

[Figure 1 : Organigramme de l'Ecole Nationale d'Informatique 6](#_Toc123914957)

[Figure 2 : Organigramme de ARATO 18](#_Toc123914958)

[Figure 3: Cycle de développement en Y de 2TUP. 34](#_Toc123914959)

[Figure 4: Relation entre 2TUP et UML. 36](#_Toc123914960)

[Figure 5: Concept d'un acteur. 40](#_Toc123914961)

[Figure 6: Formalisme d'un cas d'utilisation 41](#_Toc123914962)

[Figure 7: Formalisme de la relation d'inclusion 41](#_Toc123914963)

[Figure 8: Formalisme de la relation d'extension 41](#_Toc123914964)

[Figure 9: Formalisme de la relation de généralisation et spécialisation 42](#_Toc123914965)

[Figure 10: Diagramme de cas d'utilisation du système 43](#_Toc123914966)

[Figure 11: Modèle du domaine de l'application. 51](#_Toc123914967)

[Figure 12: Architecture MVC. 52](#_Toc123914968)

[Figure 13: Diagramme de séquence de conception « S'authentifier ». 53](#_Toc123914969)

[Figure 14: Diagramme de séquence de conception « Gérer carte ». 54](#_Toc123914970)

[Figure 15: Diagramme de séquence de conception « Envoyer demande ». 54](#_Toc123914971)

[Figure 16: Diagramme de séquence de conception « Gérer tarification ». 55](#_Toc123914972)

[Figure 17: Diagramme de séquence de conception « Gérer voitures ». 55](#_Toc123914973)

[Figure 18: Diagramme de séquence de conception « Gérer services liés à la voiture ». 56](#_Toc123914974)

[Figure 19: Diagramme de séquence de conception 56](#_Toc123914975)

[Figure 20: Diagramme de séquence de conception "Consulter service". 57](#_Toc123914976)

[Figure 21: Diagramme de classe de conception du cas d'utilisation « Gérer voitures ». 58](#_Toc123914977)

[Figure 22: Diagramme de classe de conception du cas d'utilisation "Gérer chauffeurs". 58](#_Toc123914978)

[Figure 23: Diagramme de classe de la conception globale. 59](#_Toc123914979)

[Figure 24: Diagramme de paquetage. 60](#_Toc123914980)

[Figure 25: Diagramme de déploiement. 60](#_Toc123914981)

[Figure 26: Installation du MongoDB. 62](#_Toc123914982)

[Figure 27: Installation de Django. 63](#_Toc123914983)

[Figure 28: Configuration à faire dans Visual Studio Code. 64](#_Toc123914984)

[Figure 29: Interface de Firefox. 64](#_Toc123914985)

[Figure 30: Interface de Visual Paradigm. 65](#_Toc123914986)

[Figure 31: Architecture de l'application. 65](#_Toc123914987)

[Figure 32: Création de la base de données 67](#_Toc123914988)

[Figure 33: Différentes modèles de notre projet 68](#_Toc123914989)

[Figure 34: Modification d'un chauffeur. 68](#_Toc123914990)

[Figure 35: Page d'authentification. 69](#_Toc123914991)

[Figure 36: Page de réservation. 70](#_Toc123914992)

[Figure 37: Liste des utilisateurs. 70](#_Toc123914993)

[Figure 38: Liste des taxis. 71](#_Toc123914994)

# **LISTE DES TABLEAUX**

[Tableau 1 : Organisation du système de formation pédagogique de l'Ecole 7](#_Toc123915144)

[Tableau 2 : Architecture des études correspondant au système LMD 9](#_Toc123915145)

[Tableau 3 : Listes des formations existantes à l'ENI 9](#_Toc123915146)

[Tableau 4 : Débouchés professionnels éventuels des diplômés 15](#_Toc123915147)

[Tableau 5: Inventaire des moyens matériels 21](#_Toc123915148)

[Tableau 6: Comparaison des solutions proposées 25](#_Toc123915149)

[Tableau 7: Comparaison entre MERISE et 2TUP 26](#_Toc123915150)

[Tableau 8: Comparaison entre outils de modélisation 28](#_Toc123915151)

[Tableau 9: Comparaison entre langage de programmation 28](#_Toc123915152)

[Tableau 10: Comparaison entre framework Python 29](#_Toc123915153)

[Tableau 11: Comparaison entre Vue Js, React Js, Angular Js 30](#_Toc123915154)

[Tableau 12: Comparaison de quelques IDE. 31](#_Toc123915155)

[Tableau 13: Comparaison entre SGGBD. 33](#_Toc123915156)

[Tableau 14: Dictionnaire des données 37](#_Toc123915157)

[Tableau 15: Priorisation des cas d'utilisation 43](#_Toc123915158)

[Tableau 16: Description textuelle du cas d'utilisation "S'authentifier" 44](#_Toc123915159)

[Tableau 17: Cas d'utilisation "Gérer carte". 44](#_Toc123915160)

[Tableau 18: Cas d'utilisation "Envoyer demande" 45](#_Toc123915161)

[Tableau 19: Gestion tarification. 46](#_Toc123915162)

[Tableau 20: Cas d'utilisation "Gérer voitures" 47](#_Toc123915163)

[Tableau 21: Cas d'utilisation "Gérer services liés à la voiture" 47](#_Toc123915164)

[Tableau 22: Cas d'utilisation "Gérer chauffeurs" 48](#_Toc123915165)

[Tableau 23: Cas d'utilisation "Consulter service" 49](#_Toc123915166)

# **LISTE DES ABREVIATIONS**

**API :** Interface de Programmation d’Application.

**AUF :** Agence Universitaire de la Francophonie.

**CARI :** Colloque Africain sur la Recherche en Informatique.

**CCNA :** CISCO Networking Academy.

**CITEF :** Conférence Internationale des Ecoles de formation d’Ingénieurs et Technicien d’Expression Française.

**CNH :** Commission Nationale d’Habilitation.

**CUR :** Centre Universitaire Régional.

**ENI :** Ecole Nationale d’Informatique.

**ESN :** Entreprise de Services du Numérique.

**ESPA :** Ecole Supérieure Polytechnique d’Antananarivo.

**FPPSM :** Forêts, Parcs et Pauvreté dans le Sud de Madagascar.

**GPS :** Global Positionning System.

**IA :** Intelligence Artificielle.

**IDE :** Environnement de Développement Intégré.

**INPG :** Institut National Polytechnique de Grenoble.

**IRD :** Institut de Recherche pour le Développement.

**IREMIA :** Institut de Recherche en Mathématiques et Informatique Appliquées.

**JSON :** JavaScript Object Notation.

**LMD :** Licence-Master-Doctorat.

**MERISE :** Méthode d’Etude et de Réalisation d’un Informatique Système d’Entreprise.

**MVC :** Modèle-Vue-Contrôleur.

**MVT :** Modèle-Vue-Template.

**PRESUP :** Programme de Renforcement en l’Enseignement Supérieur.

**SGBD :** Système de Gestion de Base de Données.

**SI :** Système d’Information.

**2TUP :** 2 Track Unified Process.

**TIC :** Technologies de l’Information et de la Communication.

**UML :** Unified Modeling Language.

**UPST :** Université Paul Sabatier de Toulouse.

# **INTRODUCTION GENERALE**

Actuellement, l’informatique prend une valeur très importante au niveau du développement. Elle est inséparable de la vie d’une entreprise à qui elle apporte une évolution du résultat et l’avancement du travail. C’est une science qui étudie et réalise l’automatisation des systèmes d’information.

De nos jours, il est généralement nécessaire de se déplacer pour chercher un taxi, que ce soit en hélant un taxi dans la rue ou en se rendant à un arrêt de taxi sauf si le client possédait déjà le numéro d’un chauffeur. Cependant, il existe des exceptions parce qu’il peut être difficile pour les utilisateurs de savoir à l’avance combien ils devront payer pour leur trajet vu que les tarifs peuvent varier considérablement d’un lieu à l’autre ou d’un chauffeur à l’autre. Les utilisateurs peuvent être inquiets en ce qui concerne la vérification des antécédents du chauffeur et du véhicule.

De ce fait, la conception et l’élaboration d’un projet feront l’objet de ce mémoire intitulé : « Conception et Réalisation d’une plateforme de réservation de taxi en ligne ». Ce projet consiste à réaliser une application web répondant aux besoins des utilisateurs de taxi.

Pour concevoir et réaliser cette application, nous allons utiliser un langage de conception, un outil de conception, un SGBD, des Framework et un environnement de développement

Nous l’organisons en trois grandes parties. Dans la première partie, la présentation de l’Ecole Nationale d’Informatique, de la société Arato et la description du projet. Dans la seconde partie, nous présentons l’analyse et la conception de l’application. Et dans la dernière partie que nous présentons la réalisation de notre travail.

# **PRESENTATIONS**

## **Présentation de l’Ecole Nationale d’Informatique**

### **Information d’ordre général**

L’Ecole Nationale d’informatique, en abrégé ENI, est un établissement d’enseignement supérieur rattaché académiquement et administrativement à l’Université de Fianarantsoa.

Le siège de l’Ecole se trouve à Tanambao-Antaninarenina à Fianarantsoa.

L’adresse pour la prise de contact avec l’Ecole est la suivante : Ecole Nationale d’Informatique (ENI) Tanambao, Fianarantsoa. Le numéro de sa boîte postale est 1487 avec le code postal 301. Téléphone : 020 75 508 01. Son adresse électronique est la suivante : [eni@univ-fianar.mg](mailto:eni@univ-fianar.mg). Site Web : www.univ-fianar.mg/eni

### **Missions et historique**

L’ENI se positionne sur l’échiquier socio-éducatif malgache comme étant le plus puissant secteur de diffusion et de vulgarisation des connaissances et des technologies informatiques.

Cette Ecole Supérieur peut être considérée aujourd’hui comme la vitrine et la pépinière des élites informaticiennes du pays.

L’Ecole s’est constituée de façon progressive au sein du Centre Universitaire Régional (CUR) de Fianarantsoa.

De façon formelle, l’ENI était constituée et créée au sein du (CUR) par le décret N° 83-185 du 24 Mai 1983, comme étant le seul établissement Universitaire Professionnalisé au niveau national, destiné à former des techniciens et des Ingénieurs de haut niveau, aptes à répondre aux besoins et exigences d’Informatisation des entreprises, des sociétés et des organes implantés à Madagascar.

L’ENI a pur conséquent pour mission de former des spécialistes informaticiens compétents et opérationnels de différents niveaux notamment :

* En fournissant à des étudiants des connaissances de base en informatique ;
* En leur transmettant le savoir-faire requis, à travers la professionnalisation des formations par rapport aux besoins évolutifs des sociétés et des entreprises.
* En initiant les étudiants aux activités de recherche dans les différents domaines des Technologies de l’information et la communication (TIC).

L’implantation de cette Ecole Supérieure de technologie de pointe dans un pays en développement et dans une Province (ou Faritany) à tissu économique et industriel faiblement développé ne l’a pourtant pas défavorisée, ni empêchée de former des spécialistes informaticiens de bon niveau, qui sont recherchés par les entreprises, les sociétés et les organismes publics et privés sur le marché de l’emploi.

La filière de formation d’Analystes Programmeurs a été mise en place à l’Ecole en 1983, et a été gelée par la suite en 1996, tandis que la filière de formation d’ingénieurs a été ouverte à l’Ecole en 1986.

Dans le cadre du Programme de renforcement en l’Enseignement Supérieur (PRESUP), la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes des informatiques a été mise en place en 1986 grâce à l’appui matériel et financier de la Mission Française de coopération auprès de l’Ambassade de France à Madagascar.

Une formation pour l’obtention de la certification CCNA et / ou NETWORK +.appelée« CISCO Networking Academy » a été créée à l’Ecole en 2002-2003 grâce au partenariat avec CISCO SYSTEM et l’Ecole Supérieure Polytechnique d’Antananarivo (ESPA). Cependant, cette formation n’avait pas duré longtemps.

Une formation de troisième cycle a été ouverte à l’Ecole a été ouverte à l’Ecole depuis l’année 2003 – 2004 grâce à la coopération académique et scientifique entre l’Université de Fianarantsoa pour le compte de l’ENI et l’Université Paul Sabatier de Toulouse (UPST).

Cette filière avait pour objectif de former certains étudiants à la recherche dans les différents domaines de l’Informatique, et notamment pour préparer la relève des Enseignants-Chercheurs qui étaient en poste.

Pendant l’année 2007-2008, la formation en vue de l’obtention du diplôme de Licence Professionnelle en Informatique a été mise en place à l’ENI avec les deux options suivantes de formation :

* Génie Logiciel et base de Données.
* Administration des Systèmes et réseaux.

La mise en place à l’Ecole de ces deux options de formation devait répondre au besoin de basculement vers le système Licence – Master – Doctorat (LMD).

Mais la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes Informatiques a été gelée en 2009.

En vue de surmonter les difficultés de limitation de l’effectif des étudiants accueillis à l’Ecole, notamment à cause du manque d’infrastructures, un système de « Formation Hybride » a été mise en place à partir de l’année 2010. Il s’agit en effet d’un système de formation semi-présentielle et à distance avec l’utilisation de la visioconférence pour la formation à distance.

Le système de formation hybride a été ainsi créé à Fianarantsoa ainsi qu’Université de Toliara.

### **Organigramme institutionnel de l’ENI**

Cet organigramme de l’Ecole est inspiré des dispositions du décret N° 83-185 du 23 Mai 1983.

L’ENI est administrée par un conseil d’Ecole, et dirigée par un directeur nommé par un décret adopté en conseil des Ministres.

Le Collège des enseignants regroupant tous les enseignants-chercheurs de l’Ecole est chargé de résoudre les problèmes liés à l’organisation pédagogique des enseignements ainsi que à l’élaboration des emplois du temps.

Le Conseil Scientifique propose les orientations pédagogiques et scientifiques de l’établissement, en tenant compte notamment de l’évolution du marché de travail et de l’adéquation des formations dispensées par rapport aux besoins des entreprises.

Trois départements de formation caractérisent l’organigramme :

* Le département de formation théorique à l’intérieur de l’Ecole ;
* Le département de formation pratique pour la coordination et la supervision des stages en entreprise et des voyages d’études ;

Le département de formation doctorale pour l’organisation de la formation de 3ème cycle.

La figure 1 présente l’organigramme actuel de l’Ecole.

Conseil d’école

Collège des enseignants

Conseil scientifique

Direction

Service pédagogique

Secrétariat principal

.

Parcours : Génie Logiciel et Base de Données

Service de scolarité

Parcours : Administration des Systèmes et Réseaux

Service de comptabilité

Parcours : Informatique Génerale

Service intendance

*Figure 1 : Organigramme de l'Ecole Nationale d'Informatique*

Sur cet organigramme, l’Ecole placée sous la tutelle académique et administrative de l’Université de Fianarantsoa, et dirigée par un Directeur élu par les Enseignants – Chercheurs permanents de l’Etablissement et nommé par un décret pris en Conseil des ministres pour un mandat de 3 ans.

Le Conseil de l’Ecole est l’organe délibérant de l’Ecole.

Le Collège des Enseignants propose et coordonne les programmes d’activités pédagogiques.

Le Conseil scientifique coordonne les programmes de recherche à mettre en œuvre à l’Ecole.

Le Secrétariat principal coordonne les activités des services administratifs (Scolarité, Comptabilité, et Intendance).

Conformément aux textes en vigueur régissant les Etablissements malgaches d’Enseignement Supérieur, qui sont barrés sur le système LMD, les Départements de Formation pédagogique ont été ainsi remplacés par des Mentions et des parcours. Et les chefs des Départements ont été ainsi remplacés par des responsables des mentions et les responsables des parcours.

Un administrateur des Réseaux et Systèmes gère le système d’information de l’Ecole et celui de l’Université.

### **Domaine de spécialisation**

Les activités de formation et de recherche organisées à l’ENI portent sur les domaines suivants :

* Génie logiciel et Base de Données ;
* Administration des Systèmes et Réseaux ;
* Informatique Générale
* Modélisation informatique et mathématique des Systèmes complexes.

D’une manière plus générale, les programmes des formations sont basés sur l’informatique de gestion et sur l’informatique des Systèmes et Réseaux. Et les modules de formation intègrent aussi bien des éléments d’Informatique fondamentale que des éléments d’Informatique appliquée.

Le tableau 1 décrit l’organisation du système de formation pédagogique de l’Ecole.

*Tableau 1 : Organisation du système de formation pédagogique de l'Ecole*

|  |  |
| --- | --- |
| Formation théorique | Formation pratique |
| * Enseignement théorique * Travaux dirigés * Travaux pratiques | * Etude de cas * Travaux de réalisation * Projets/ Projets tutorés * Voyage d’études * Stages |

### **Architecture des formations pédagogiques**

Le recrutement des étudiants à l’ENI se fait uniquement par voie de concours d’envergure nationale en première année.

Les offres de formation organisées à l’Ecole ont été validées par la Commission Nationale d’Habilitation (CNH) auprès du Ministères de l’Enseignement Supérieur et de la Recherche Scientifique selon les dispositions de l’Arrêté N°31.174/2012-MENS en date du 05 Décembre 2012.

Au sein de l’ENI, il existe une seule mention (INFORMATIQUE) et trois parcours :

* Génie logiciel et Base de Données ;
* Administration des Systèmes et Réseaux
* Informatique Générale

L’architecture des études à trois niveaux conforment au système Licence- Master-Doctorat (LMD) permet les comparaisons et les équivalences académiques des diplômes au niveau international.

* L = Licence (Bac + 3) = L1, L2, L3 = 6 semestres S1 à S6
* M = Master (Bac + 5) = M1, M2 = 4 semestres S7 à S10

Le diplôme de licence est obtenu en 3 années des études après Baccalauréat. Et le diplôme de Master est obtenu en 2 ans après obtenu du diplôme de LICENCE.

Le MASTER PROFESSIONNEL est un diplôme destiné à la recherche emploi au terme des études.

Le MASTER RECHERCHE est un diplôme qui remplace l’ancien Diplôme d’Etudes Approfondies (DEA), et qui permet de s’inscrire directement dans une Ecole Doctorale.au terme des études.

* D = Doctorat (Bac +8)

Le Doctorat est un diplôme qu’on peut obtenir en 3 ans après l’obtention du diplôme de MASTER RECHERCHE.

Le tableau 2 présente l’architecture des études correspondant au système LMD.

*Tableau 2 : Architecture des études correspondant au système LMD*



DTS : Diplôme de Technicien Supérieur

BTS : Brevet de Technicien Supérieur

DUT : Diplôme Universitaire de Technicien

La licence peut avoir une vocation générale ou professionnelle.

Le master peut avoir une vocation professionnelle ou de recherche.

*Tableau 3 : Listes des formations existantes à l'ENI*

|  |  |  |
| --- | --- | --- |
|  | FORMATION EN | |
|  | LICENCE PROFESSIONNELLE ET HYBRIDE | MASTER |
| Condition d’admission | Par voie de concours Formation Professionnelle : 100 candidats  Formation hybride : 150 candidats |  |
| Condition d’accès | Bac de série C, D ou Technique | Etre titulaire de Licence professionnelle |
| Durée de formation | 3 années | 2 années |
| Diplôme à délivrer | Diplôme de Licence Professionnelle en Informatique | Diplôme de Master Professionnel  Diplôme de Master/ Recherche |

L’accès en première année de MASTER se fait automatiquement pour les étudiants de l’Ecole qui ont obtenu le diplôme de Licence Professionnelle.

Le Master Recherche permet à son titulaire de poursuivre directement des études en doctorat et de s’inscrire directement dans une Ecole Doctorale.

Les Ecoles Doctorales jouissent d’une autonomie de gestion par rapport aux Etablissements de formation universitaire.

Il convient de signaler que par arrêté ministériel N° 21.626/2012 – MESupRES publié le 9 Août 2012 par la Commission National d’habilitation (CNH), l’Ecole Doctorale « Modélisation – Informatique » a été habilitée pour l’Université de Fianarantsoa.

Depuis l’année universitaire 2010-2011, l’ENI s’est mise à organiser des formations hybrides en informatique dans les différentes régions (Fianarantsoa, Toliara) en raison de l’insuffisance de la capacité d’accueil des infrastructures logistiques. En effet, le système de formation hybride semi - présentielle utilise la visioconférence pour la formation à distance.

Bien qu’il n’existe pas encore au niveau international de reconnaissance écrite et formelle des diplômes délivrés par l’ENI, les étudiants diplômés de l’Ecole sont plutôt bien accueillis dans les instituts universitaires étrangères (CANADA, Suisse, France…)

### **Relations de l’ENI avec les entreprises et les organismes**

Les stages effectués chaque année par les étudiants mettent l’Ecole en rapport permanent avec plus de 300 entreprises et organismes publics, semi-publics et privés, nationaux et internationaux.

L’Ecole dispose ainsi d’un réseau d’entreprises, de sociétés et d’organismes publics et privés qui sont des partenaires par l’accueil en stage de ses étudiants, et éventuellement pour le recrutement après l’obtention des diplômes par ces derniers.

Les compétences que l’Ecole cherche à développer chez ses étudiants sont l’adaptabilité, le sens de la responsabilité, du travail en équipe, le goût de l’expérimentation et l’innovation.

En effet, la vocation de l’ENI est de former des techniciens supérieurs de niveau LICENCE et des ingénieurs de type généraliste de niveau MASTER avec des qualités scientifiques, techniques et humaines reconnues, capables d’évoluer professionnellement dans des secteurs d’activité variés intégrant l’informatique.

Les stages en milieu professionnel permettent de favoriser une meilleure adéquation entre les formations à l’Ecole et les besoins évolutifs du marché de l’emploi.

Les principaux débouchés professionnels des diplômés de l’Ecole concernent les domaines suivants :

* L’informatique de gestion d’entreprise
* Les technologies de l’information et de la communication (TIC)
* La sécurité informatique des réseaux
* L’administration des réseaux et des systèmes
* Les services bancaires et financiers, notamment le Mobile Banking
* Les télécommunications et la téléphonie mobile
* Les Big Data
* Le commerce, la vente et l’achat, le Marketing
* L’ingénierie informatique appliquée
* L’écologie et le développement durable

Parmi les sociétés, entreprises et organismes partenaires de l’Ecole, on peut citer : ACCENTURE Mauritius, Air Madagascar, Ambre Associates, Airtel, Agence Universitaire de la Francophonie ( AUF) , B2B, Banque Centrale, BFG-SG, BIANCO, BLUELINE, CNaPS, Bureau National de Gestion des Risques et des Catastrophes (BNGRC), CEDII-Fianarantsoa, Data Consulting, Central Test, Centre National Antiacridien, CNRE, CHU, CNRIT, COLAS, Direction Générale des Douanes, DLC, DTS/Moov, FID, FTM, GNOSYS, IBONIA, INGENOSIA, INSTAT, IOGA, JIRAMA, JOUVE, MADADEV, MAEP, MEF, MEN, MESupRES, MFB, MIC, MNINTER, Min des postes/Télécommunications et du Développement Numérique, NEOV MAD, Ny Havana, Madagascar National Parks, OMNITEC, ORANGE, OTME, PRACCESS, QMM Fort-Dauphin, SMMC, SNEDADRS Antsirabe, Sénat, Société d’Exploitation du Port de Toamasina (SEPT), SOFTWELL, Strategy Consulting, TELMA, VIVETEC, Société LAZAN’I BETSILEO, WWF …

L’organisation de stage en entreprise continue non seulement à renforcer la professionnalisation des formations dispensées, mais elle continue surtout à accroître de façon exceptionnelle les opportunités d’embauche pour les diplômés de l’Ecole.

### **Partenariat au niveau international**

Entre 1196 et 1999, l’ENI avait bénéficié de l’assistance technique et financière de la Mission Française de Coopération et d’action culturelle dans le cadre du Programme de Renforcement de l’Enseignement Supérieur (PRESUP) consacré à l’Ecole a notamment porté sur :

* Une dotation en logiciels, micro-ordinateurs, équipements de laboratoire de maintenance et de matériels didactiques
* La réactualisation des programmes de formation assortie du renouvellement du fonds de la bibliothèque
* L’appui à la formation des formateurs
* L’affectation à l’Ecole d’Assistants techniques français

De 2000 à 2004, l’ENI avait fait partie des membres du bureau de la Conférence Internationale des Ecoles de formation d’Ingénieurs et Technicien d’Expression Française (CITEF).

Les Enseignants-Chercheurs de l’Ecole participent régulièrement aux activités organisées dans le cadre du Colloque Africain sur la Recherche en Informatique (CARI).

L’ENI avait également signé un accord de coopération interuniversitaire avec l’Institut de Recherche en Mathématiques et Informatique Appliquées (IREMIA) de l’Université de la Réunion, l’Université de Rennes 1, l’INSA de Rennes, l’Institut National Polytechnique de Grenoble (INPG).

A partir du mois de Juillet 2001, l’ENI avait abrité le Centre de Réseau Opérationnel (Network Operating Center) du point d’accès à Internet de l’Ecole ainsi que de l’Université de Fianarantsoa. Grâce à ce projet américain qui a été financé par l’USAID Madagascar, l’ENI de l’Université de Fianarantsoa avait été dotées d’une ligne spécialisée d’accès permanent au réseau Internet.

L’ENI avait de même noué des relations de coopération avec l’Institut de Recherche pour le Développement (IRD).

L’objet du projet de coopération avait porté sur la modélisation environnementale du Corridor forestier de Fandriana jusqu’à Vondrozo (COFAV). Dans ce cadre, un atelier scientifique international avait été organisé à l’ENI en Septembre 2008. Cet atelier scientifique avait eu pour thème de modélisation des paysages.

Et dans le cadre du programme scientifique PARRUR, l’IRD avait financé depuis 2010 le projet intitulé « Forêts, Parcs et Pauvreté dans le Sud de Madagascar (FPPSM). Des étudiants en DEA et des Doctorants issus de l’ENI avaient participé à ce Programme.

Par ailleurs, depuis toujours la même année 2010, l’ENI de Fianarantsoa avait été sélectionnée pour faire partie des organismes partenaires de l’Université de Savoie dans le cadre du projet TICEVAL relatif à la certification des compétences en TIC ;

Le projet TICEVAL avait été financé par le Fonds Francophone des Inforoutes pour la période allant de 2010 à 2012, et il avait eu pour objectif de généraliser la certification des compétences en Informatique et Internet du type C2i2e et C2imi.

Dans le cadre du projet TICEVAL, une convention de coopération avec l’Université de Savoie avait été signée par les deux parties concernées. La mise en oeuvre de la Convention de Coopération avait permis d’envoyer des étudiants de l’ENI à Chambéry pour poursuivre des études supérieures en Informatique.

Enfin et non des moindres, l’ENI avait signé en Septembre 2009 un protocole de collaboration scientifique avec l’ESIROI – STIM de l’Université de la Réunion.

Comme l’ENI constitue une pépinière incubatrice de technologie de pointe, d’emplois et d’entreprises, elle peut très bien servir d’instrument efficace pour renforcer la croissance économique du pays, et pour lutter contre la Pauvreté.

De même que le statut de l’Ecole devrait permettre de renforcer la position concurrentielle de la Grande Ile sir l’orbite de la modélisation grâce au développement des nouvelles technologies.

### **Débouchés professionnels avec des diplômés**

Le chômage des jeunes diplômés universitaires fait partie des maux qui gangrènent Madagascar. L’environnement socio-politique du pays depuis 2008 jusqu’ à ce jour a fait que le chômage des diplômés est devenu massif par rapport aux établissements de formation supérieure existants.

Cependant, les formations proposées par l’Ecole permettent aux diplômés d’être immédiatement opérationnels sur le marché du travail avec la connaissance d’un métier complet lié à l’informatique aux TIC.

L’Ecole apporte à ses étudiants un savoir-faire et un savoir-être qui les accompagnent tout au long de leur vie professionnelle. Elle a une vocation professionnalisante.

Les diplômés en LICENCE et en MASTER issus de l’ENI peuvent faire carrière dans différents secteurs.

L’Ecole bénéficie aujourd’hui de 34 années d’expériences pédagogiques et de reconnaissance auprès des sociétés, des entreprises et des organismes. C’est une Ecole Supérieure de référence en matière informatique.

Par conséquent, en raison de fait que l’équipe pédagogique de l’Ecole est expérimentée, les enseignants-chercheurs et les autres formateurs de l’Ecole sont dotés d’une grande expérience dans l’enseignement et dans le milieu professionnel.

L’Ecole est fière de collaborer de façon régulière avec un nombre croissant d’entreprises, de sociétés et d’organismes publics et privés à travers les stages des étudiants. Les formations dispensées à l’Ecole sont ainsi orientées vers le besoin et les attentes des entreprises et des sociétés.

L’Ecole fournit à ses étudiants de niveau LICENCE et MASTER des compétences professionnelles et métiers indispensables pour les intégrer sur le marché du travail.

L’Ecole s’efforce de proposer à ses étudiants une double compétence à la fois technologique et managériale combinant l’informatique de gestion ainsi que l’administration des réseaux et systèmes.

D’une manière générale, les diplômés de l’ENI n’éprouvent pas de difficultés particulières à être recrutés au terme de leurs études. Cependant, l’ENI recommande à ses diplômés de promouvoir l’entrepreneuriat en TIC et de créer des cybercafés, des SSII ou des bureaux d’études.

*Tableau 4 : Débouchés professionnels éventuels des diplômés*

|  |  |
| --- | --- |
| LICENCE | * Analyste * Programmeur * Administrateur de site web/ de portail web * Assistant Informatique et internet * Chef de projet web ou multimédia * Développeur Informatique ou multimédia * Intégrateur web ou web designer * Hot Liner/ Hébergeur Internet * Agent de référencement * Technicien/ Supérieur de help desk sur Informatique * Responsable de sécurité web * Administrateur de réseau |
| MASTER | * Administrateur de réseau et système * Architecture de système d’information * Développeur d’applications * Ingénieur réseau * Webmaster/ web designer * Concepteur Réalisateur d’applications * Directeur du système de formation * Directeur de projet informatique * Chef de projet informatique * Responsable de sécurité informatique * Consultant fonctionnel ou freelance |

### **Ressources humaines**

* Directeur de l’Ecole : Docteur MAHATODY Thomas, Docteur HDR.
* Responsable de Mention : Monsieur RABETAFIKA Louis Haja, Maître de Conférences.
* Responsable de Parcours « Génie Logiciel et Base de Données » : Monsieur RALAIVAO Jean Christian, Assistant d’Enseignement Supérieur et de Recherche.
* Responsable de Parcours « Administration Système et Réseaux » : Monsieur SIAKA, Assistant d’Enseignement Supérieur et de Recherche.
* Responsable de Parcours « Informatique Générale » : Monsieur Gilante GESAZAFY
* Nombre d’Enseignants permanents : 12 dont deux (01) Professeur Titulaire, six (05) Maîtres de Conférences et cinq (04) Assistants d’Enseignements Supérieur et de Recherche
* Nombre d’Enseignants vacataires : 10
* Personnel Administratif : 23

## **Présentation de la société d’accueil**

### **Fiche d’identification**

La société ARATO est une société spécialisée dans le domaine du développement informatique. Plus précisément une entreprise de service numérique. Son siège se trouve à Fianarantsoa – Madagascar.

Dénomination : ARATO

Adresse : Lot 00 28 SS 3608 à Sahalava Fianarantsoa (Batiment Ex Wisdom School)

Téléphone : +261349881619 - +261341005868

Mail : [contact@arato.mg](mailto:contact@arato.mg)

Site web : [www.arato.mg](http://www.arato.mg)

Activité : ESN

Numéro Statistique : 63122 21 2020 0 00875

Numéro Impôts : 3004115277

Fondateurs : Julia RATOVONDRAHONA et Josué RATOVONDRAHONA

### **Brèves historiques**

Crée en 2018 par Julia RATOVONDRAHONA, diplômée en Master Recherche de l’Université de Fianarantsoa sur la transformation numérique et ses avantages pour les Entreprises. Et Josué RATOVONDRAHONA doctorant en informatique comme support technique pour l’Entreprise. ARATO est basé à Fianarantsoa Madagascar. Les services proposés par ARATO sont : l’urbanisation des SI d’une Entreprise, l’externalisation des services informatiques d’une Entreprise. Création de site web pour la visibilité des produits et services. Ainsi que bons nombres de services liés à la numérisation.

### **Objectifs : Missions et Activités**

Urbanisation des SI : Etudier, Analyser, Concevoir et Réaliser les process pour la transformation numérique d’un client pour augmenter la production, la visibilité, le gain de temps, etc.

Externalisation des services informatiques : offrir les services nécessaires pour le fonctionnement d’une entreprise en matière de logiciels et services informatique.

Création d’application web, mobile : réalisation d’une application spécifique pour le compte d’un client selon ses besoins. Exemple : site web pour la visibilité des services, intégration des APIs, création application mobile pour la traduction, etc.

Développement et implémentation des algorithmes IA pour avoir un système réactif et autonome.

Formation : formation de renforcement de capacité et préparation à des certifications internationales.

### **Organigramme de ARATO**

Conseil d’administration

Responsable administratif

Directeur de projet

Ressources Humaines

Chef de projet

Responsable Finance

Community Manager

Equipe Technique

*Figure 2 : Organigramme de ARATO*

### **Ressources humaines**

|  |  |
| --- | --- |
| **Postes** | **Effectifs** |
| Responsable administratif | 1 |
| Directeur de projet | 1 |
| Chef de projet | 2 |
| Ressources humaines | 1 |
| Responsable Finance | 1 |
| Equipe Technique | 19 |
| Community Manager | 3 |

### **Partenaire et Bienfaiteurs**

ARATO possède des bienfaiteurs pour le développement du projet :

* Le président fondateur de l’association « Andrarangy France ».
* Le président fondateur de l’association sportive « VAS-Y Boaikely France ».
* Le Directeur Général du Zomatel Fianarantsoa.

Comme partenaire :

* L’Association « Ndao Hifanosika Fianara ».
* L’Association « Andrarangy ».
* Le Service Régionale Solde et Pension Fianarantsoa.
* La société « Localease France ».
* Le site « Mon Séjour Cacher France ».
* UFICASM.
* LFBHM
* Société « Nambintsoa Tamatave ».

## **Description du projet**

La description du projet sert à énumérer les besoins du projet à réaliser, les moyens nécessaires afin de pouvoir le réaliser et les résultats attendus de l’utilisateur à son aboutissement.

### **Formulation**

La société Arato souhaite se doter des moyens informatiques pour gérer la réservation de taxi en ligne. En effet, la société Arato souhaite avoir en possession, une application qui offre l’avantage de gérer automatiquement les voitures, les services liés à la voiture, les chauffeurs, les cartes, les tarifications, les utilisateurs ainsi que leurs courses.

### **Objectif et besoins de l’utilisateur**

L’objectif principal de ce projet consiste à concevoir et réaliser une plateforme de réservation de taxi en ligne.

1. **Objectif**

L’application doit être capable de :

* Gérer les voitures.
* Gérer les services liés à la voiture.
* Gérer les chauffeurs.
* Gérer les cartes.
* Gérer les tarifications.

1. **Besoins de l’utilisateur**

Les besoins représentent les fonctionnalités que le système doit posséder.

L’application que nous allons développer devra regrouper toutes les fonctionnalités nécessaires pour :

* L’administrateur afin de :
* Gérer les voitures et les services liés à la voiture.
* Gérer les chauffeurs.
* Gérer les cartes et les tarifications.
* L’utilisateur afin de :
* Visiter les différents services de la plateforme.
* S’authentifier.
* Faire une réservation de taxi en ligne.

### **Moyens nécessaires à la réalisation du projet (Humain, matériel et éventuellement financier)**

Pour réaliser ce projet, on doit disposer des moyens humains, moyens matériels et moyens logiciels afin qu’on puisse répondre aux besoins de l’utilisateur.

* **Moyens humains**

Les personnes suivantes sont requises pour réaliser ce projet :

* Un chef de projet : chargé de mener le projet et de gérer son bon fonctionnement (l’encadreur professionnel).
* Un concepteur : chargé d’analyser et d’étudier le projet (le stagiaire).
* Deux développeurs : ce sont les stagiaires, dont l’une est chargée d’automatiser les affectations des taxis pour le ramassage des clients qui commande un taxi à travers une application . L’autre pour l’application web back office où il y a le suivi des voitures, des services liés à la voiture, des chauffeurs, des cartes et des tarifications.
* **Moyens matériels**

Le tableau 5 montre le matériel nécessaire pour le développement du projet.

*Tableau 5: Inventaire des moyens matériels*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Système d’exploitation** | **Processeur** | **RAM** | **Disque dur** |
| Un laptop | Windows 10 | Core i3 | 4 Go | 500 Go |

* **Moyens logiciels**

Les logiciels utilisés pour la mise en œuvre :

* Firefox comme navigateur.
* Visual Studio Code comme éditeur de texte.
* MongoDB pour un serveur de base de données.
* Visual Paradigm 16.3 pour la conception et modélisation UML.

### **Résultats attendus**

A la fin de ce projet, on attend à obtenir une application opérationnelle, facile à utiliser, plus sécurisée, performante, fiable, qui suit les normes de qualité d’un logiciel pour :

* L’obtention d’une interface graphique simple, conviviale pour manipuler les différentes fonctionnalités (ajout, modification, suppression, liste).
* Une application souple et fiable.
* Avoir une base de données fiable, sécurisée et cohérente.

# **ANALYSE ET CONCEPTION**

## **Analyse préalable**

L’analyse préalable est une étape importante dans la conception et la réalisation d’un projet parce qu’elle consiste à recueillir et à analyser les informations nécessaires pour comprendre les besoins et les attentes des utilisateurs et déterminer comment la plateforme peut répondre à ces besoins de manière efficace et efficiente.

### **Analyse de l’existant**

L’analyse de l’existant consiste à recueillir et à analyser des informations sur l’état actuel, afin de déterminer les forces et les faiblesses du système, d’identifier les opportunités et les menaces auxquels il est confronté et de définir les objectifs et les actions à mettre en place pour améliorer sa performance.

#### **Organisation actuelle**

Actuellement, la prise de taxi consiste à se rendre dans la rue où à un emplacement de prise en charge de taxis et à attendre qu’un taxi libre s’arrête pour nous prendre en charge. Nous pouvons également appeler le numéro de notre chauffeur de taxi habituel si l’on a. Une fois dans le taxi, on indiquera notre destination au chauffeur et on paiera la course à la fin du trajet selon le tarif en vigueur.

#### **Inventaire des moyens matériels et logiciels**

L’inventaire des moyens matériels et logiciels désigne l’ensemble des équipements et outils utilisés pour la réservation de taxi de nos jours. Il peut inclure des éléments tels qu’un téléphone portable ou un poste fixe pour appeler un chauffeur de taxi, un moyen de paiement (espèces, carte bancaire, etc.) pour régler la course, un véhicule de taxi équipé d’un gyrophare et d’un compteur de course ainsi que des dispositifs de sécurité et de confort (ceintures de sécurité et des airbags pour les passagers).

### **Critique de l’existant**

Les différents processus de la prise de taxi mentionnés ci-dessus rencontre actuellement de différents problèmes comme :

* Inconfort : Il peut être difficile de trouver un taxi libre dans la rue et il peut être nécessaire d’attendre longtemps avant qu’un taxi s’arrête pour vous prendre en charge.
* Manque de transparence : Le tarif de la course n’est pas toujours affiché de manière claire et il peut y avoir des surprises au moment de régler la course. De plus, il peut être difficile de savoir à l’avance combien coûtera une course.
* Manque de sécurité : Il peut être difficile de vérifier l’identité et la fiabilité du chauffeur et du véhicule qu’il conduit.
* Manque de personnalisation : Il n’est pas toujours possible de choisir le type de véhicule ou les options de confort qui nous conviennent le mieux.

### **Conception avant-projet**

La conception avant-projet consiste à élaborer un plan détaillé pour la réalisation d’un projet et à évaluer sa faisabilité.

#### **Propositions de solutions**

Face à ces contraintes, on a envisagé la mise en place d’un logiciel facile à déployer. En effet, l’application doit être facile à utiliser et avoir des interfaces simples et avancés.

* Solution 1 : Utilisation d’un logiciel libre et open source.
* Solution 2 : Installer un système de navigation GPS (Global Positioning System) dans les taxis.
* Solutions 3 : Conception et réalisation d’une application pour gérer la réservation de taxi en ligne.

Le tableau 6 représente la comparaison des solutions proposées.

*Tableau 6: Comparaison des solutions proposées*

|  |  |  |
| --- | --- | --- |
| **Solutions** | **Avantages** | **Inconvénients** |
| 1 | * Logiciels gratuits. * Possibilité de modifier le code source pour répondre à vos besoins spécifiques. | * Compatibilité : les logiciels open source peuvent ne pas être compatible avec certaines plateformes ou système d’exploitation. * Coûts cachés : Bien que les logiciels open source soient gratuits, ils peuvent nécessiter l’achat de licences. |
| 2 | * Pour déterminer la position, la vitesse et la direction d’un taxi afin de minimiser les temps d’attente pour les utilisateurs. * Pour optimiser les trajets : ce qui peut augmenter les revenus du conducteur | * Le prix d’un GPS peut être coûteux, en particulier pour les modèles de haute qualité. * Les signaux GPS peuvent être bloqués : ce qui peut entraîner des erreurs de positionnement |
| 3 | * Réserver un taxi depuis n’importe où, à tout moment. * On n’a plus besoin de se déplacer pour prendre un taxi. | * Les zones de réservation sont limitées seulement dans la ville de Fianarantsoa. * Il peut y avoir des erreurs de réservation : en ce qui concerne le type de véhicules ou les tarifs. |

#### **Solution retenue**

En tenant compte les critiques et les besoins de l’utilisateur, on a opté pour la quatrième solution qui est de concevoir et de réaliser une application pour la réservation d’un taxi en ligne.

#### **Méthodes et outils utilisés**

1. Utilisation de la méthode

Il existe plusieurs méthodes pour la modélisation du système d’information. Il est donc difficile de choisir quelle méthode doit-on adopter pour la réalisation de notre projet, c’est pourquoi nous allons comparer quelques méthodes dans le tableau ci-dessous afin de pouvoir trancher sur une qu’on va adopter.

Le tableau 7 montre la comparaison entre MERISE et 2TUP.

*Tableau 7: Comparaison entre MERISE et 2TUP*

|  |  |  |
| --- | --- | --- |
|  | **MERISE** | **2TUP** |
| **DESCRIPTIONS** | Méthode de conception de systèmes d’information qui vise à décrire et à modéliser de manière structurée les besoins et les objectifs d’un système d’information, ainsi que son architecture et son fonctionnement. | Une méthode qui vise à séparer les aspects techniques et fonctionnels d’un projet de développement de logiciels |
| **POINTS FORTS** | * Utilise des diagrammes et des modèles visuels pour représenter les éléments et les relations d’un système. * Elle permet de structurer de manière rigoureuse la conception d’un système d’information. | * Meilleure gestion de projet : elle permet de découper le projet en étapes et en sous-étapes, ce qui peut permettre une meilleure planification et suivi de l’avancement. * Elle peut être utilisée pour des projets de développement de logiciels de différentes tailles et de différentes complexités, ce qui la rend adaptée à une large gamme de projets. |
| **POINTS FAIBLES** | * Moins utilisée. * Elle est moins adaptée aux projets à forte incertitude. | * Elle peut être perçue comme complexe, car elle comprend de nombreuses étapes. * Elle est difficile à mettre en œuvre. |

Le choix s’est porté vers la méthode 2TUP. En effet, le processus 2TUP qui signifie « 2 Track Unified Process » est un processus de développement logiciel qui propose un cycle de développement en Y en dissociant les aspects techniques des aspects fonctionnels. Il propose une étude parallèle des deux branches : fonctionnelle (étude de l’application) et technique (étude de l’implémentation).

1. Utilisation du langage de modélisation UML

UML (Unified Modeling Language) est un langage de modélisation visuel utilisé pour représenter et documenter les aspects fonctionnels et techniques d’un système de logiciel. Il permet de créer des diagrammes qui représentent les différentes parties du système, telles que les classes, les interfaces, les objets et les interactions entre eux.

1. Outils de conception

Pour réaliser les différents diagrammes, nous allons utiliser un outil de modélisation. Nous avons choisi donc de comparer quelques outils de modélisation afin d’identifier celui que nous allons utiliser.

Le tableau 8 montre les comparaisons entre quelques outils de modélisation.

*Tableau 8: Comparaison entre outils de modélisation*

|  |  |  |
| --- | --- | --- |
| **Visual Paradigm** | **ArgoUML** | **StarUML** |
| * Offre de nombreuses fonctionnalités. * Propose différentes licences adaptées aux besoins de différents types d’utilisateurs et de projets. * Facile à utiliser. | * Un outil gratuit. * Dispose d’une interface utilisateur intuitive, ce qui peut être utile pour les utilisateurs qui découvrent l’outil pour la première fois. | * Offre des outils de génération de documentation, tels que la génération de diagrammes et de rapports, ce qui peut être utile pour documenter le système de manière structurée. |

On an opté l’outil de modélisation Visual Paradigm car il permet de dessiner tous les types de diagrammes UML.

Visual Paradigm est une suite logicielle de modélisation et de gestion de projets qui permet de créer et de gérer des modèles de données, de processus et de systèmes. Elle comprend un large éventail d’outils de modélisation, tels que des diagrammes de classes, de séquence, de cas d’utilisation, de déploiement, etc.

1. Choix du langage de programmation

Il est important de connaître qu’on ne peut pas se passer de langage de programmation à la phase de développement de l’application. En effet, elle nous permet d’écrire un code source qui subit une transformation par la machine afin d’obtenir un programme.

Pour nous aider avec le choix de langage de programmation, le tableau 13 suivant montre une comparaison de quelques langages de programmation.

*Tableau 9: Comparaison entre langage de programmation*

|  |  |  |
| --- | --- | --- |
|  | **Java** | **Python** |
| **Avantages** | * Facile à apprendre et à utiliser. * Offre une grande portabilité grâce à la plateforme Java Virtual Machine. | * Offre une grande flexibilité. * Peut être utilisé dans de nombreux domaines, tels que la science des données et le développement Web. * Peut être utilisé sur de nombreuses plateformes et systèmes d’exploitation différents. |
| **Inconvénients** | * Peut-être lent pour les applications nécessitant une grande performance. | * Nécessite l’installation de bibliothèques externes pour certaines fonctionnalités. |

Vis-à-vis de ces comparaisons, nous avons opté notre choix pour le langage Python parce que python est un langage de programmation facile à utiliser dans de nombreux environnements, flexible et puissant.

1. Framework de développement

En programmation informatique, un framework est un ensemble de librairies de code et de conventions de programmation qui fournissent une structure de base pour le développement de logiciels.

Les frameworks ont pour but de simplifier le processus de développement en fournissant un cadre de travail prédéfini et en offrant des fonctionnalités prêtes à l’emploi qui peuvent être utilisées dans les programmes développes.

Le tableau 10 montre la comparaison des frameworks Python.

*Tableau 10: Comparaison entre framework Python*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Django** | **Flask** | **Pyramid** |
| **Avantages** | * Offre une grande flexibilité. * Prend en charge la sécurité et la gestion des utilisateurs. * Permet de créer des applications web rapidement. | * Open source. * Léger et facile à utiliser. * Peut être étendu en utilisant des bibliothèques externes. | * Offre une grande variété de fonctionnalité. * Facile à apprendre. |
| **Inconvénients** | * Peut-être complexe à maîtriser pour les développeurs débutants. * Nécessite une structure de projet spécifique. * Peut-être lent pour les applications à grande échelle. | * Offre moins de fonctionnalité. * Nécessite l’ajout de bibliothèques externes pour certaines fonctionnalités. | * Moins connu. * Nécessite une structure de projet spécifique. |

Pour une application web comme la gestion de réservation de taxi en ligne ; on a choisi d’utiliser le langage Python avec **Django** comme Framework Back-end. Django utilise un modèle-vue-template (MVT) pour structurer les applications web, ce qui signifie qu’il sépare les données de l’application, son comportement et sa présentation visuelle. Cela permet aux développeurs de créer des applications web complexes de manière organisées et efficaces.

Pour styliser notre application, on va ajouter pour un Framework Front-end.

Pour cela, on a le choix entre Vue Js, React Js et Angular Js.

Le tableau 11 montre la comparaison entre ces différents Frameworks.

*Tableau 11: Comparaison entre Vue Js, React Js, Angular Js*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Vue Js** | **React Js** | **Angular Js** |
| **Avantages** | * Open source. * Léger. * Utilisé pour les applications très grands gabarits. | * Il se concentre sur la création d’interfaces utilisateur en utilisant des composant réutilisables. * Utilisé pour le développement d’un projet vaste. | * Angular offre une grande variété de fonctionnalités pour la création d’interfaces utilisateurs. * Prend en charge de nombreux types de données, tels que JSON, XML. |
| **Inconvénients** | * Peut-être complexe. * Moins de support disponible. | * Nécessite l’utilisation des bibliothèques externes. * Incompatibilité avec les navigateurs obsolètes. | * Peut être lent pour les applications qui nécessitent une mise à jour fréquente de l’interface utilisateur. |

1. Utilisation de l’IDE

Un IDE (Environnement de Développement Intégré) est un logiciel qui fournit un ensemble d’outils de développement pour aider les développeurs à créer des applications. Un IDE comprend généralement un éditeur de code, un débogueur et un gestionnaire de versions, ainsi que d’autres outils pour faciliter le développement d’applications.

On trouvera dans le tableau 12 la comparaison de quelques IDE.

*Tableau 12: Comparaison de quelques IDE.*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SublimeText** | **Visual Studio Code** | **PHP Storm** |
| **Avantages** | * Gratuit. * Prend en charge de nombreux langages de programmation. | * Multiplateforme. * Inclut un débogueur intégré et un gestionnaire de vision. | * Offre une interface intuitive. * Offre une grande flexibilité. |
| **Inconvénients** | * Offre moins de fonctionnalités. * Nécessite l’ajout de plugins pour certaines fonctionnalités avancées. | * Peut-être lent pour les projets de grande envergure. * Nécessite une configuration et une maintenance supplémentaires. | * Payant et nécessite une licence pour être utilisé. |

On a choisi Visual Studio Code comme environnement de développement parce que c’est un éditeur de code source polyvalent qui convient à de nombreux projets de développement grâce à sa prise en charge de différents langages, son débogueur intégré et ses fonctionnalités étendues.

1. Utilisation du SGBD

Un SGBD (Système de Gestion de Bases de Base de Données) est un logiciel conçu pour stocker, gérer et manipuler des données structurées. Les SGBD permettent aux utilisateurs de créer, de lire, de mettre à jour et de supprimer des données de manière organisée et efficace. Ils offrent également des fonctionnalités de sécurité pour protéger les données, ainsi que des outils de gestion pour aider à maintenir et à optimiser les performances de la base de données.

Il existe de nombreux types de SGBD, notamment les SGBD relationnels (tels que MySQL et Oracle) et les bases de données non relationnels (tels que MongoDB et Cassandra). Chacun de ces types de SGBD a ses propres avantages et inconvénients en fonction des besoins en matière de stockage et de gestion de données de l’application.

Le tableau 13 représente la comparaison entre quelques SGBD.

*Tableau 13: Comparaison entre SGGBD.*

|  |  |  |
| --- | --- | --- |
|  | **MySQL** | **MongoDB** |
| **Avantages** | * Les données sont stockées dans des tables relationnelles avec des colonnes et des lignes. * Performances stables. | * Les données sont stockées sous forme de documents JSON (JavaScript Object Notation). * Convient aux applications qui ont besoin de stocker et de gérer des données de manière flexible et à grande échelle. |
| **Inconvénients** | * Utilisé dans des environnements de petite taille. * Modèle de données rigide. | * Consommation de mémoire. * Manques de certaines fonctionnalités. * Formation requise. |

On a choisi d’utiliser MongoDB car elle utilise un modèle de données de documents qui permet de stocker des données de manière flexible et de manière à ce qu’elles soient facilement accessibles.

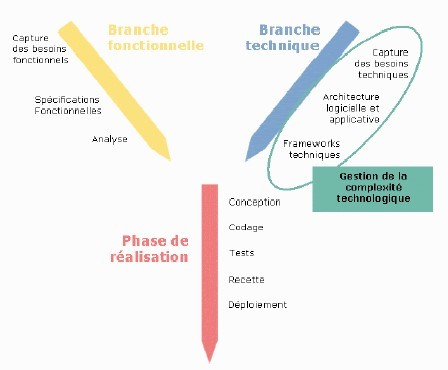
### **Présentation de la méthode 2TUP**

2TUP est un processus de développement logiciel qui implémente le processus unifié (c.à.d itératif, incrémental). Il propose un cycle de développement qui sépare les aspects techniques des aspects fonctionnels en partant du constat que toute évolution peut se traiter parallèlement suivant un axe fonctionnel et un axe technique. Ensuite, en fusionnant les résultats de ces deux axes, on arrive à réaliser le système désiré ; ce qui nous donne un cycle de développement sous forme de Y.

Le processus de développement en Y articule autour de trois branches :

* Branche fonctionnelle : Ceci permet d’étudier d’une manière pointue la spécification fonctionnelle afin d’obtenir une idée de ce que va réaliser le système en termes de métier.
* Branche technique : Elle recense toutes les contraintes à respecter pour réaliser le système. Elle définit ensuite les composants nécessaires à la construction de l’architecture technique.
* Branche de réalisation : Cette phase est la fusion des deux étapes précédentes. Elle concerne les étapes de la conception préliminaire, la conception détaillée, le codage et les tests puis l’étape de recette de déploiement.

La figure 3 montre le cycle de développement en Y de 2TUP.



*Figure 3: Cycle de développement en Y de 2TUP.*

**UML c’est quoi ?**

UML est un langage de modélisation graphique utilisé pour visualiser, spécifier, construire et documenter les éléments d’un système logiciel ou d’un processus métier. Il permet de représenter de manière graphique les différents aspects d’un système, tels que les classes, les objets, les interactions, les processus, les données.

UML propose treize diagrammes qui peuvent être utilisés dans la description d’un système. Ces diagrammes sont regroupés dans deux grands ensembles.

**Les diagrammes structurels** :

Les diagrammes structurels sont des diagrammes utilisés pour représenter la structure et les relations de différents éléments d’un système ou d’un processus. Ils permettent de visualiser comment ces éléments interagissent et sont organisés les uns par rapport aux autres.

* Diagramme de classe : il permet de représenter les classes d’un système et leurs relations (héritage, agrégation, association, etc.).
* Diagramme d’objet : est un diagramme utilisé pour représenter les objets d’un système et leurs relations avec d’autres objets.
* Diagramme de composant : il permet de représenter les composants d’un système et leur relation avec d’autres composants ou avec les classes du système.
* Diagramme de déploiement : il permet de représenter les éléments d’un système et leur emplacement physique ou logique (serveurs, ordinateurs, réseaux, etc.).
* Diagramme de paquetage : est un diagramme utilisé pour représenter la structure des paquetages d’un système et les relations entre ces paquetages.
* Diagramme de structure composite : est un diagramme utilisé pour représenter le structure interne d’un composant et ses relations avec d’autres composants ou avec les classes du système.

**Les diagrammes de comportement**:

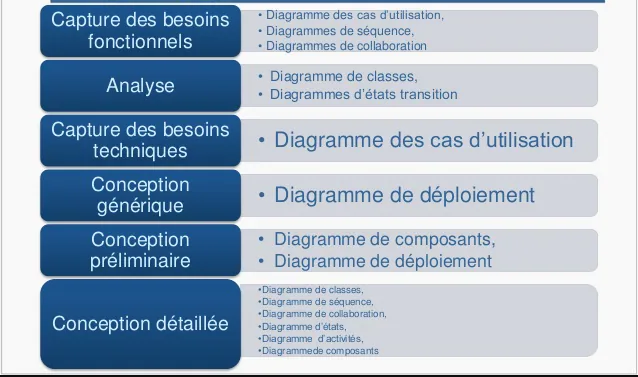
Ce sont des diagrammes utilisés pour représenter les interactions et les échanges d’informations entre les éléments d’un système.

* Diagramme de cas d’utilisation : il permet de représenter les actions que peut réaliser un utilisateur dans un système et les interactions entre l’utilisateur et le système.
* Diagramme d’état-transition : il permet de représenter les différents états d’un objet et les transitions entre ces états.
* Diagramme d’activités : il permet de représenter les activités d’un processus et les relations entre ces activités.
* Diagramme de séquence : il permet de représenter les interactions entre les objets d’un système et leur ordre chronologique.
* Diagramme de communication (ou collaboration) : il permet de représenter les interactions entre les objets d’un système et leur rôle dans la réalisation d’une fonctionnalité spécifique.
* Diagramme global d’interaction : il permet de représenter l’ensemble des interactions et des échanges d’informations entre les éléments d’un système.
* Diagramme de temps : il permet de représenter les évènements d’un système et leur ordre chronologique.

**2TUP et UML** :

La méthode 2TUP utilise UML pour représenter les exigences du logiciel et la conception du logiciel au cours de la phase de planification et de conception. UML est utilisé pour créer des diagrammes de classes, des diagrammes de cas d’utilisation et des diagrammes de séquence qui décrivent la structure et le comportement du logiciel.

La figure 4 représente la relation entre 2TUP et UML.



*Figure 4: Relation entre 2TUP et UML.*

## **Analyse conceptuelle**

L’analyse conceptuelle consiste à définir et à modéliser les concepts clés du domaine d’étude sous forme de diagrammes de concepts. Ces diagrammes permettent de décrire les concepts de manière formelle, précise et de montrer leurs relations avec d’autres concepts.

### **Dictionnaire des données**

Un dictionnaire de données est un document qui décrit les données utilisées dans un système informatique ou une organisation. Il contient des informations sur les différentes données utilisées par le système, telles que leur signification, leur format, leur taille.

Le tableau 14 montre le dictionnaire de donnée de notre application.

*Tableau 14: Dictionnaire des données*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Description | Type | Taille | Format |
| Adresse | Adresse d’un chauffeur | AN | 20 |  |
| carrosserie | Carrosserie du taxi | AN | 10 |  |
| charge\_utile | Charge utile d’un taxi | N |  |  |
| Date | Date de fait du permis | D |  | jj-mm-aaaa |
| dateCrs | Date quand le client a demandé une course | D |  | jj-mm-aaaa |
| date\_certificat | Date de fait d’une capacité d’un taxi | D |  | jj-mm-aaaa |
| date\_fabrication | Date de fait de la carte grise d’un taxi | D |  | jj-mm-aaaa |
| date\_naissance | Date de naissance d’un chauffeur | D |  | jj-mm-aaaa |
| date\_vis | Date de visite d’un taxi | D |  | jj-mm-aaaa |
| Depart | Le lieu de récupération d’un client | AN | 30 |  |
| description | Description du lieu de récupération du client | AN | 255 |  |
| destination | Le lieu de destination d’un client | AN | 30 |  |
| Distance | La distance entre le lieu de depart et la destination d’un client | N |  |  |
| Email | L’email d’un client | AN | 30 |  |
| fin\_ass | Date de fin de la validité de l’assurance d’un taxi | D |  | jj-mm-aaaa |
| fin\_vis | Date de fin de la validité de la visite d’un taxi | D |  | jj-mm-aaaa |
| idCli | Identifiant d’un client | N |  |  |
| idCrs | Identifiant d’une course | N |  |  |
| lieu\_naissance | Lieu de naissance d’un chauffeur | AN | 20 |  |
| marque | Marque d’une voiture | AN | 20 |  |
| montant | Le prix de la course d’un client | N |  |  |
| name | Le nom d’un client | AN | 20 |  |
| nb\_place | Nombre de place d’un taxi | N |  |  |
| nomAg | Nom de l’agence d’une assurance | AN | 10 |  |
| nomChf | Nom du chauffeur | AN | 30 |  |
| numAg | Numéro qui identifie une agence | AN | 10 |  |
| numCap | Numéro qui identifie la capacité d’un taxi | N |  |  |
| numCat | Numéro qui identifie une catégorie de permis | AN | 10 |  |
| numChf | Numéro qui identifie le chauffeur | N |  |  |
| numImm | Numéro d’immatriculation d’un taxi | AN | 10 |  |
| numMoteur | Numéro du moteur d’un taxi | AN | 20 |  |
| numPer | Numéro qui identifie chaque permis | AN | 10 |  |
| numSerie | Numéro qui identifie une carte grise | N |  |  |
| numVis | Numéro qui identifie une visite | N |  |  |
| password | Mot de passe d’un client | AN | 10 |  |
| poids\_total | Poids total d’un taxi | N |  |  |
| poids\_vide | Poids à vide d’un taxi | N |  |  |
| prenomChf | Prénom d’un chauffeur | AN | 20 |  |
| profession | Profession d’un chauffeur | AN | 20 |  |
| Ref | Référence d’une assurance | N |  |  |
| Type | Type d’une catégorie | AN | 5 |  |

AN : Alphanumérique

N : Numérique

D : Date

### **Règles de gestion**

Une règle de gestion est un type de diagramme de règle de séquence qui permet de décrire les règles de gestion d’un système de manière formelle et précise. Elle est décrite en utilisant des symboles et des notations spécifiques qui permettent de représenter les différentes parties du système, les interactions entre ces parties et les règles qui régissent ces interactions.

Le système est soumis à quelques règles de gestions telles que :

RG01 : Un client doit faire au moins une course.

RG02 : Pour effectuer les différentes courses d’un client, on a besoin d’un taxi.

RG03 : Un taxi a besoin d’un chauffeur pour conduire le véhicule afin d’emmener les clients à leur destination.

RG04 : Tout conducteur doit être en mesure de présenter son permis de conduire.

RG05 : Pour conduire différents types de véhicules, chaque permis de conduire doit obtenir de ladite catégories.

RG06 : La capacité de conduire le véhicule est une exigence commune à tous les chauffeurs.

RG07 : Une voiture doit présenter une carte grise qui permet de l’identifier.

RG08 : Tout véhicule doit être assuré au moins avec un certificat d’assurance, afin de couvrir les dégâts matériels et humains qu’il peut provoquer.

RG09 : Un agent d’assurance a pour rôle de s’occuper un à plusieurs assurances de voiture.

RG10 : Le contrôle technique ou visite de voiture est obligatoire pour identifier les défaillances susceptibles de porter atteinte à la sécurité des usagers de la route et à l’environnement.

### **Représentation et spécification des besoins**

#### **Diagrammes des cas d’utilisation**

Le diagramme de cas d’utilisation est le diagramme principal du modèle UML. C’est à partir d’un cas d’utilisation qu’on identifie les acteurs et leurs interactions avec l’application.

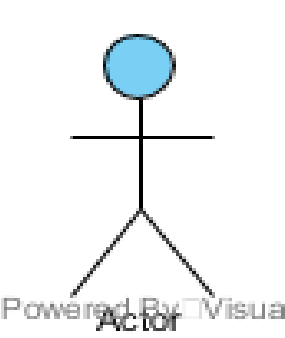
Un cas d’utilisation est donc une représentation d’un ensemble de séquence d’actions qui sont réalisées par le système et produisent un résultat observable intéressant pour un acteur particulier.

Pour représenter un diagramme de cas d’utilisation, on a besoin des éléments de bases suivantes :

* **Acteur :** Entité externe (utilisateur humain, dispositif matériel ou autre) qui interagit directement avec le système étudié.

Formalisme :

La figure 5 montre le concept d’un acteur.



Nom de l’acteur

*Figure 5: Concept d'un acteur.*

* **Cas d’utilisation :** Suite d’interactions entre un acteur et le système. Il doit définir exhaustivement les exigences fonctionnelles du système où chaque cas d’utilisation correspond à une fonction métier du système, selon le point de vue d’un de ses acteurs.

Formalisme :

La figure 6 illustre le formalisme d’un cas d’utilisation.

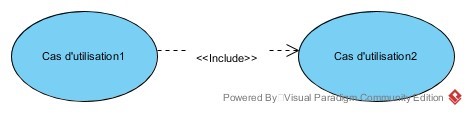
**

*Figure 6: Formalisme d'un cas d'utilisation*

* **La relation d’inclusion :** Une relation représentée par prototype « include ».

Formalisme :

La figure 7 représente la relation d’inclusion.

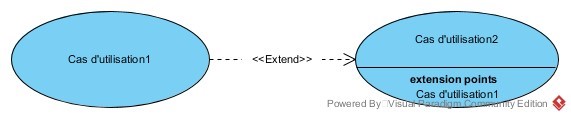
**

*Figure 7: Formalisme de la relation d'inclusion*

* **La relation d’extension :** Une relation représentée par prototype « extends ».

Formalisme :

La figure 8 représente la relation extension.

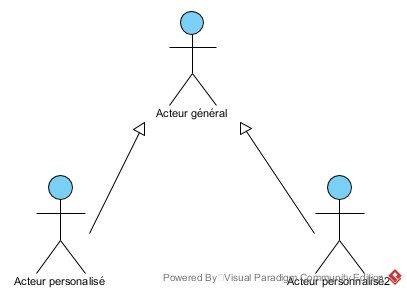


*Figure 8: Formalisme de la relation d'extension*

* **La relation de généralisation/spécification :** Une relation d’héritage. La relation de généralisation entre deux entités exprime le fait que l’entité spécialisée est un cas particulier de l’entité générale. L’entité spécialisée peut réaliser tout ce que l’entité générale peut réaliser.

Formalisme :

La figure 9 montre le formalisme de la relation de généralisation et spécialisation.

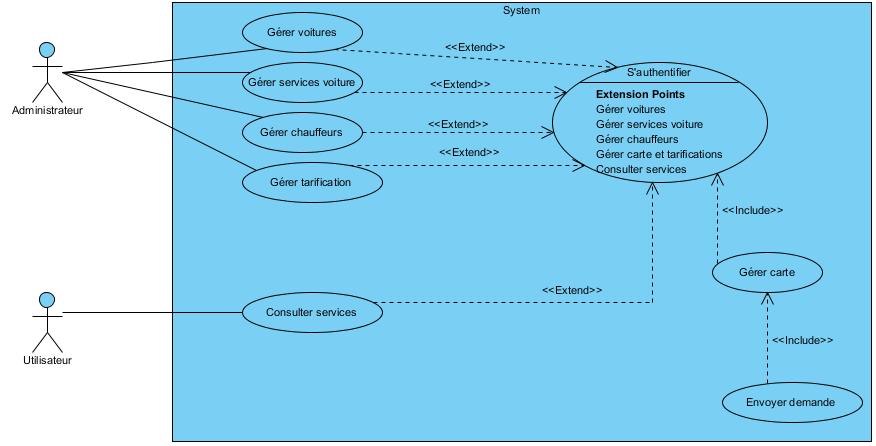
**

*Figure 9: Formalisme de la relation de généralisation et spécialisation*

Pour pouvoir établir le diagramme des cas d’utilisation du système, on a recensé les deux acteurs qui participent au fonctionnement de la gestion de réservation de taxi en ligne :

* L’administrateur : c’est celui qui possède le contrôle total sur les fonctionnalités de l’application.
* Les utilisateurs : Toutes personnes qui visitent le site. Ils peuvent s’inscrire et s’authentifier dans l’application.

La figure 10 montre le diagramme des cas d’utilisation du système.



*Figure 10: Diagramme de cas d'utilisation du système*

#### **Priorisation des cas d’utilisation**

La priorisation des cas d’utilisation permet de décrire chronologiquement les opérations.

Le tableau 15 montre la priorisation des cas d’utilisation.

*Tableau 15: Priorisation des cas d'utilisation*

|  |  |
| --- | --- |
| **Priorité** | **Cas d’utilisation** |
| 1 | S’authentifier |
| 2 | Gérer carte |
| 3 | Envoyer demande |
| 4 | Gérer tarification |
| 5 | Gérer voitures |
| 6 | Gérer services liés à la voiture |
| 7 | Gérer chauffeurs |
| 8 | Consulter services |

Pour détailler la dynamique du cas d’utilisation, la procédure la plus évidente consiste à recenser de façon textuelle toutes les interactions entre les acteurs et le système.

1. Cas d’utilisation « S’authentifier »

A travers le tableau 16, nous présentons la description textuelle du cas d’utilisation « S’authentifier ».

*Tableau 16: Description textuelle du cas d'utilisation "S'authentifier"*

|  |  |
| --- | --- |
| Acteur principal | Utilisateur |
| Acteur secondaire | Néant |
| Précondition | L’utilisateur clique sur le bouton Connexion. |
| Début | L’utilisateur arrive sur la page d’authentification. |
| Postcondition | L’authentification est soumise. |
| Fin | Le système renvoi l’utilisateur à la page de réservation. |
| Scénario nominal | 1-L’utilisateur ouvre l’application.  2-L’acteur clique sur le bouton connexion.  3-Le système affiche la page d’authentification.  4-L’utilisateur saisit son email et son mot de passe.  5-Le système vérifie si les données existent.  6-Le système affiche la page de réservation de taxi. |
| Scénario alternatif | A1-L’acteur n’a pas rempli tous les champs.  A2-Le système affiche un message d’erreur.  A3-Démarrage à l’étape 4 du scénario nominal.  A4-Le système redemande les informations de l’utilisateur.  A5-L’utilisateur réessaye de remplir le formulaire. |
| Scénario d’exception | E1-Si l’email et le mot de passe de l’utilisateur n’est pas valide, l’authentification est refusée. |

1. Cas d’utilisation « Gérer carte »

Le tableau 17 nous montre la description textuelle du cas d’utilisation « Gérer carte ».

*Tableau 17: Cas d'utilisation "Gérer carte".*

|  |  |
| --- | --- |
| Acteur principal | Utilisateur |
| Acteur secondaire | Néant |
| Précondition | L’utilisateur a un compte et est connecté à l’application. |
| Début | L’utilisateur accède à la page de gestion de sa course. |
| Postcondition | L’utilisateur a choisi son lieu de départ et son lieu de destination. |
| Fin | L’utilisateur quitte la page de gestion de sa course. |
| Scénario nominal | 1-L’utilisateur ouvre l’interface de gestion de sa course.  2-L’utilisateur choisit son lieu de départ et son lieu de destination. |
| Scénario alternatif | A1- L’utilisateur n’a pas rempli tous les champs.  A2- Le système affiche un message d’erreur.  A3-Démarrage à l’étape 2 du scénario nominal.  A4- Le système redemande les informations de l’utilisateur.  A5- L’utilisateur réessaye de remplir le formulaire. |
| Scénario d’exception | E1-L’utilisateur décide de ne pas faire de demande de réservation.  E3-L’utilisateur clique sur le bouton Logout.  E2-Le système redirige l’utilisateur vers la page d’authentification. |

1. Cas d’utilisation « Envoyer demande »

Le tableau 18 nous montre la description textuelle du cas d’utilisation « Envoyer demande ».

*Tableau 18: Cas d'utilisation "Envoyer demande"*

|  |  |
| --- | --- |
| Acteur principal | Utilisateur |
| Acteur secondaire | Néant |
| Précondition | La demande est soumise. |
| Début | Le système notifie l’utilisateur que la demande a été bien envoyée. |
| Postcondition | Néant |
| Fin | Néant |
| Scénario nominal | 1- L’utilisateur envoie sa demande de réservation.  2-Le système envoie un message de confirmation que la demande est bien envoyée. |
| Scénario alternatif | Néant |
| Scénario d’exception | E1-L’utilisateur décide de ne pas envoyer sa demande.  E2-L’utilisateur se déconnecte en cliquant sur le bouton Logout.  E3-L’utilisateur est redirigé vers la page d’authentification. |

1. Cas d’utilisation « Gérer tarification »

A travers le tableau 19, nous présentons la description textuelle du cas d’utilisation « Gérer tarification ».

*Tableau 19: Gestion tarification.*

|  |  |
| --- | --- |
| Acteur principal | Administrateur |
| Acteur secondaire | Néant |
| Précondition | Les tarifications doivent être enregistrées dans le système. |
| Début | L’administrateur ouvre l’interface de gestion de tarifications. |
| Postcondition | L’administrateur peut visualiser et modifier les informations sur la tarification. |
| Fin | L’administrateur enregistre les modifications apportées aux informations sur la tarification. |
| Scénario nominal | 1-L’administrateur ouvre l’interface de gestion de tarifications.  2-L’administrateur visualise les informations sur la tarification.  3-L’administrateur modifie les informations sur la tarification selon ses besoins.  4-L’administrateur enregistre les modifications apportées aux informations sur la tarification. |
| Scénario alternatif | A1-L’administrateur décide de ne pas apporter de modifications aux informations sur la tarification.  A2-L’administrateur quitte l’interface de gestion de tarification. |
| Scénario d’exception | E1-La tarification n’est pas enregistrée dans le système. |

1. Cas d’utilisation « Gérer voitures »

A travers le tableau 20, nous présentons la description textuelle du cas d’utilisation « Gérer voitures ».

*Tableau 20: Cas d'utilisation "Gérer voitures"*

|  |  |
| --- | --- |
| Acteur principal | Administrateur |
| Acteur secondaire | Néant |
| Précondition | La voiture doit être enregistrée dans le système. |
| Début | L’administrateur ouvre l’interface de gestion de voitures. |
| Postcondition | L’administrateur peut visualiser et modifier les informations sur la voiture. |
| Fin | L’administrateur enregistre les modifications apportées aux informations sur la voiture. |
| Scénario nominal | 1-L’administrateur ouvre l’interface de gestion de voitures.  2-L’administrateur visualise les informations sur la voiture.  3-L’administrateur modifie les informations sur la voiture selon ses besoins.  4-L’administrateur enregistre les modifications apportées aux informations sur la voiture. |
| Scénario alternatif | A1-L’administrateur décide de ne pas apporter de modifications aux informations sur la voiture.  A2-L’administrateur quitte l’interface de gestion de voitures. |
| Scénario d’exception | E1-La voiture n’est pas enregistrée dans le système. |

1. Cas d’utilisation « Gérer services liés à la voiture »

A travers le tableau 21, nous présentons la description textuelle du cas d’utilisation « Gérer services liés à la voiture ».

*Tableau 21: Cas d'utilisation "Gérer services liés à la voiture"*

|  |  |
| --- | --- |
| Acteur principal | Administrateur |
| Acteur secondaire | Néant |
| Précondition | Les différents services liés à la voiture doivent être enregistré dans le système. |
| Début | L’administrateur ouvre l’interface de gestion de services liés à la voiture. |
| Postcondition | L’administrateur peut visualiser et modifier les informations sur les services liés à la voiture. |
| Fin | L’administrateur enregistre les modifications apportées aux informations sur les services liés à la voiture. |
| Scénario nominal | 1-L’administrateur ouvre l’interface de gestion de services liés à la voiture.  2-L’administrateur visualise les informations sur les services liés à la voiture.  3-L’administrateur modifie les informations sur les services liés à la voiture selon ses besoins.  4-L’administrateur enregistre les modifications apportées aux informations sur les services liés à la voiture. |
| Scénario alternatif | A1-L’administrateur décide de ne pas apporter de modifications aux informations sur les services liés à la voiture.  A2-L’administrateur quitte l’interface de gestion de services liés à la voiture. |
| Scénario d’exception | E1-Les services liés à la voiture ne sont pas enregistrés dans le système. |

1. Cas d’utilisation « Gérer chauffeurs »

A travers le tableau 22, nous présentons la description textuelle du cas d’utilisation « Gérer chauffeurs ».

*Tableau 22: Cas d'utilisation "Gérer chauffeurs"*

|  |  |
| --- | --- |
| Acteur primaire | Administrateur |
| Acteur secondaire | Néant |
| Précondition | Le chauffeur doit être enregistré dans le système. |
| Début | L’administrateur ouvre l’interface de gestion de chauffeurs. |
| Postcondition | L’administrateur peut visualiser et modifier les informations sur le chauffeur. |
| Fin | L’administrateur enregistre les modifications apportées aux informations sur le chauffeur. |
| Scénario nominal | 1-L’administrateur ouvre l’interface de gestion de chauffeurs.  2-L’administrateur visualise les informations sur le chauffeur.  3-L’administrateur modifie les informations sur le chauffeur selon ses besoins.  4-L’administrateur enregistre les modifications apportées aux informations sur le chauffeur. |
| Scénario alternatif | A1-L’administrateur décide de ne pas apporter de modifications aux informations sur le chauffeur.  A2-L’administrateur quitte l’interface de gestion de chauffeurs. |
| Scénario d’exception | E1-Le chauffeur n’est pas enregistré dans le système. |

1. Cas d’utilisation « Consulter service »

A travers le tableau 23, nous présentons la description textuelle du cas d’utilisation « Consulter service ».

*Tableau 23: Cas d'utilisation "Consulter service"*

|  |  |
| --- | --- |
| Acteur primaire | Utilisateur |
| Acteur secondaire | Néant |
| Précondition | Néant |
| Début | L’utilisateur ouvre l’interface de l’application. |
| Postcondition | L’utilisateur a accès à la liste des services disponibles. |
| Fin | L’utilisateur peut naviguer vers d’autres fonctionnalités de l’application. |
| Scénario nominal | 1-L’utilisateur ouvre l’interface de l’application.  2-L’utilisateur a accès à la liste des services disponibles.  3-L’utilisateur peut naviguer vers d’autres fonctionnalités de l’application. |
| Scénario alternatif | Néant |
| Scénario d’exception | E1-Le système n’arrive pas à charger la liste des services en raison d’une erreur de connexion ou d’un autre problème technique. |

#### **Diagramme des séquences système pour chaque cas d’utilisation**

Les diagrammes de séquences système sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique.

Syntaxe des diagrammes de séquences système :

* Les acteurs : Représentés par un bonhomme et une ligne verticale (ligne de vie).
* Les objets : Représentés par un rectangle et une ligne verticale (ligne de vie).
* La ligne de vie : Une ligne de vie se représente par un rectangle, auquel est accroché une ligne verticale pointillée, qui symbolise la participation à une interaction objet ou acteur.
* Les messages : Traduisent les interactions (échange d’informations) entres objets. Ils sont représentés par des flèches orientées de l’émetteur au récepteur. Deux types de messages peuvent être distingués :
* Message synchrone : Un message synchrone se représente par une flèche en traits pleins, partant de la ligne de l’expéditeur et allant vers celle du destinataire. Ce message peut être suivi d’une réponse qui se représente par une flèche en pointillé.
* Message asynchrone : L’émetteur n’attend pas la réponse à son message, il poursuit l’exécution de ses opérations.

### **Spécification des besoins techniques**

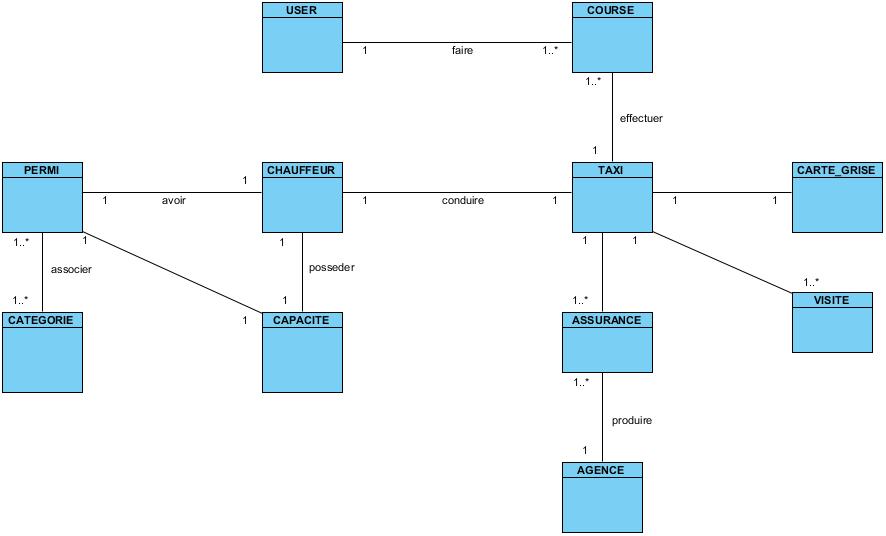
Les besoins techniques décrivent toutes les contraintes techniques auxquelles est soumis le système pour sa réalisation et son bon fonctionnement. En ce qui concerne notre application, nous avons dégagé les besoins suivants :

* La fiabilité : Les données fournies par l’application doivent être fiables.
* L’application doit fournir une interface simple pour tout type d’utilisateur.
* L’application doit être fonctionnelle.

### **Modélisation du domaine (Modèle de domaine)**

Le modèle du domaine est un diagramme de classe très simple qui montre les classes participantes à partir des diagrammes de cas d’utilisation.

La figure 11 montre le modèle du domaine de l’application.



*Figure 11: Modèle du domaine de l'application.*

## **Conception détaillée (UML)**

La conception détaillée c’est la description des fonctions d’un logiciel en vue de sa réalisation. La spécification fonctionnelle décrit dans le détail la façon dont les exigences seront prises en compte et établit l’architecture du système.

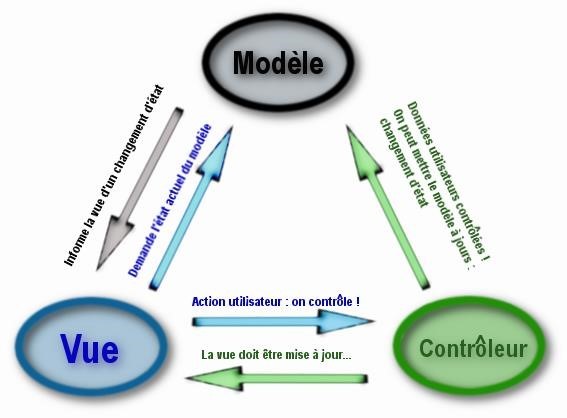
### **Architecture du système**

L’architecture MVC est un pattern architectural qui sépare les données (le modèle), l’interface homme-machine (la vue) et la logique de contrôle (le contrôleur).

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

* Modèle : Il correspond aux données stockées généralement dans une base de données. Dans un langage orienté objet, ces données sont exploitées sous forme de classes.
* Vue : Elle représente l’interface utilisateur. Elle n’effectue aucun traitement, mais elle présente tout simplement les résultats renvoyés par le modèle.
* Contrôleur : Il effectue la synchronisation entre le modèle et les vues.

La figure 12 illustre le schéma de l’architecture MVC.

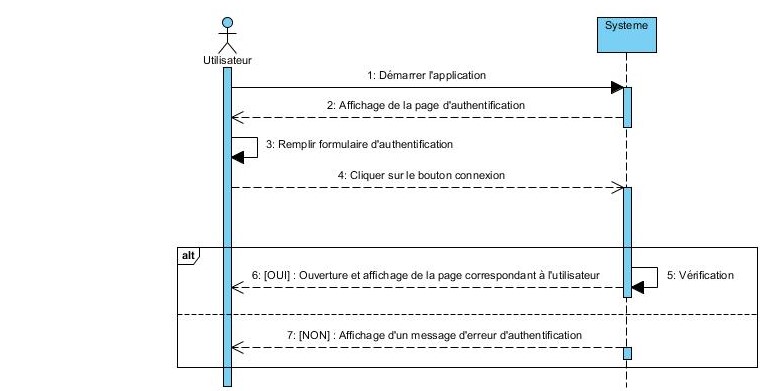


*Figure 12: Architecture MVC.*

### **Diagramme de séquence de conception pour chaque cas d’utilisation**

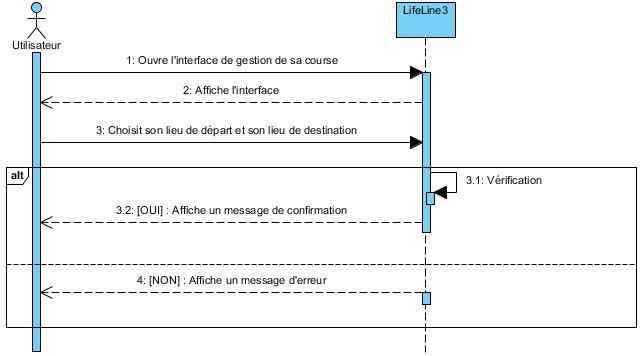
Les diagrammes de séquence de conception sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

La figure 13 illustre le diagramme de séquence de conception du cas d’utilisation « S’authentifier ».

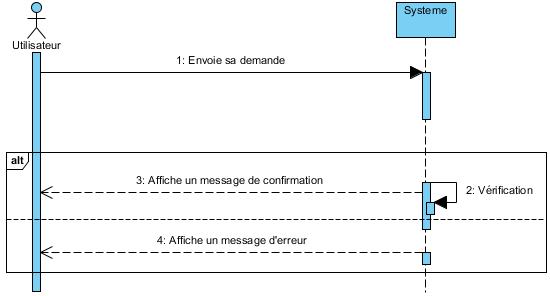


*Figure 13: Diagramme de séquence de conception « S'authentifier ».*

La figure 14 illustre le diagramme de séquence de conception du cas d’utilisation « Gérer carte ».

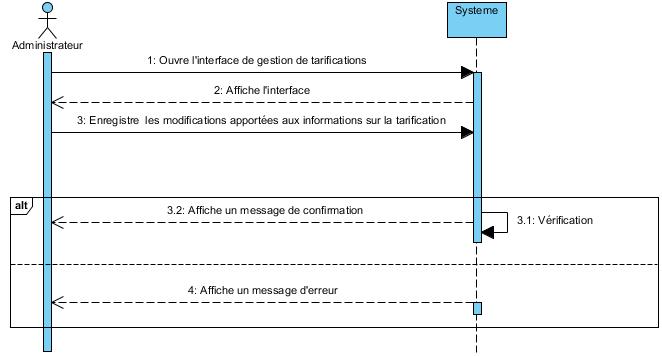


*Figure 14: Diagramme de séquence de conception « Gérer carte ».*

La figure 15 illustre le diagramme de séquence de conception du cas d’utilisation « Envoyer demande ».

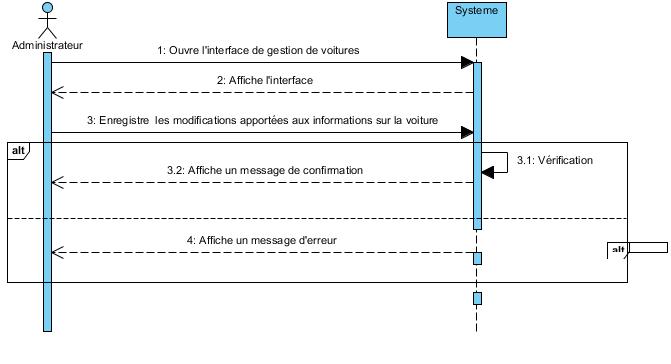
*Figure 15: Diagramme de séquence de conception « Envoyer demande ».*

La figure 16 illustre le diagramme de séquence de conception du cas d’utilisation « Gérer tarification ».



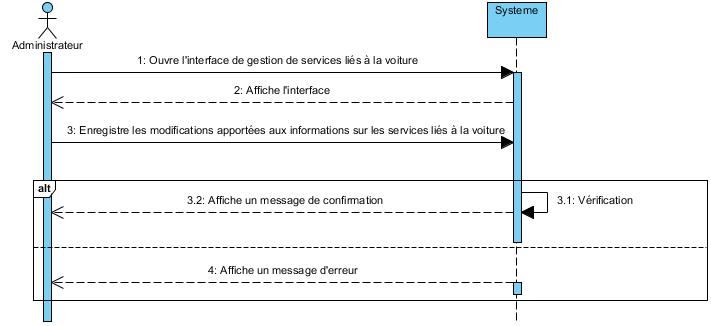
*Figure 16: Diagramme de séquence de conception « Gérer tarification ».*

La figure 17 illustre le diagramme de séquence de conception du cas d’utilisation « Gérer voitures ».



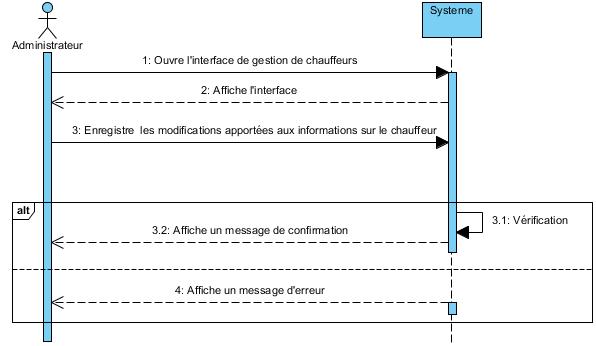
*Figure 17: Diagramme de séquence de conception « Gérer voitures ».*

La figure 18 illustre le diagramme de séquence de conception du cas d’utilisation « Gérer services liés à la voiture ».



*Figure 18: Diagramme de séquence de conception « Gérer services liés à la voiture ».*

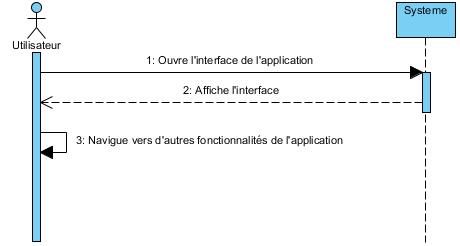
La figure 19 illustre le diagramme de séquence de conception du cas d’utilisation « Gérer chauffeurs ».



*Figure 19: Diagramme de séquence de conception*

*« Gérer chauffeurs ».*

La figure 20 illustre le diagramme de séquence de conception du cas d’utilisation « Consulter service ».



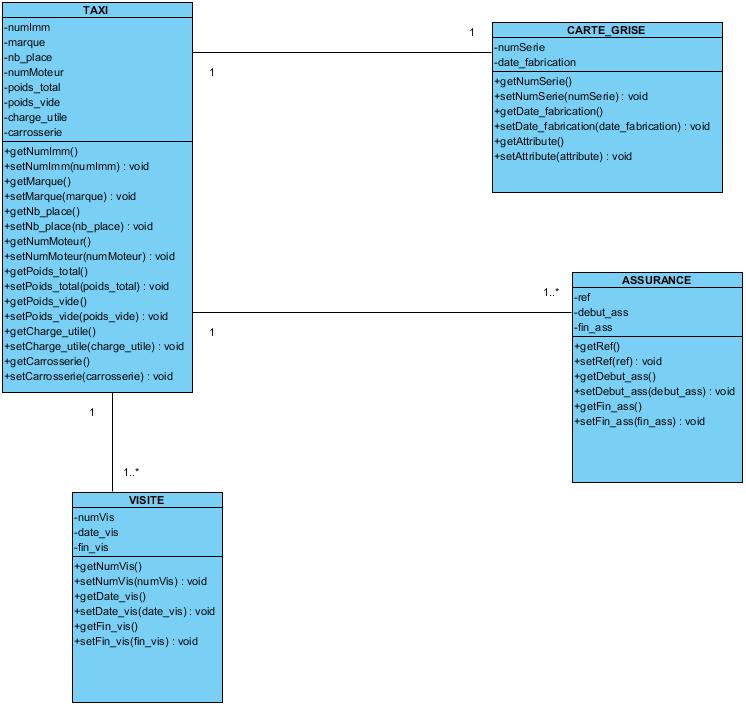
*Figure 20: Diagramme de séquence de conception "Consulter service".*

### **Diagramme de classe de conception pour chaque cas d’utilisation**

Le diagramme de classe est la représentation de la structure interne du système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d’utilisation. Il permet aussi de modéliser les classes du système et leurs relations indépendamment d’un langage de programmation particulier. Les principaux éléments de cette vue statique sont :

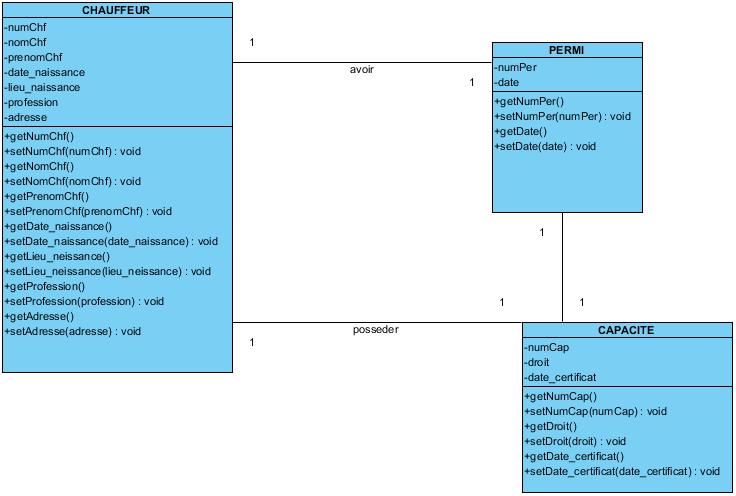
* Les classes : Une classe est représentée par un rectangle divisé en trois sections. La section supérieure est le nom de la classe, la section centrale définit les propriétés de la classe.
* Association : C’est une relation générique entre deux classes.
* Composition : Une relation de composition est indiquée par une ligne avec un « diamant remplit ».
* Dépendance : Une relation de dépendance est représentée par une flèche pointillée.
* Agrégation : Une relation d’agrégation est représentée par une ligne avec un « diamant creux ».
* Généralisation : Une relation de généralisation est indiquée par une flèche creuse se dirigeant vers la classe parent.

Pour une meilleure compréhension, montrons le diagramme de classe de conception du cas d’utilisation « Gérer voitures » dans la figure 21 ci-dessous.



*Figure 21: Diagramme de classe de conception du cas d'utilisation « Gérer voitures ».*

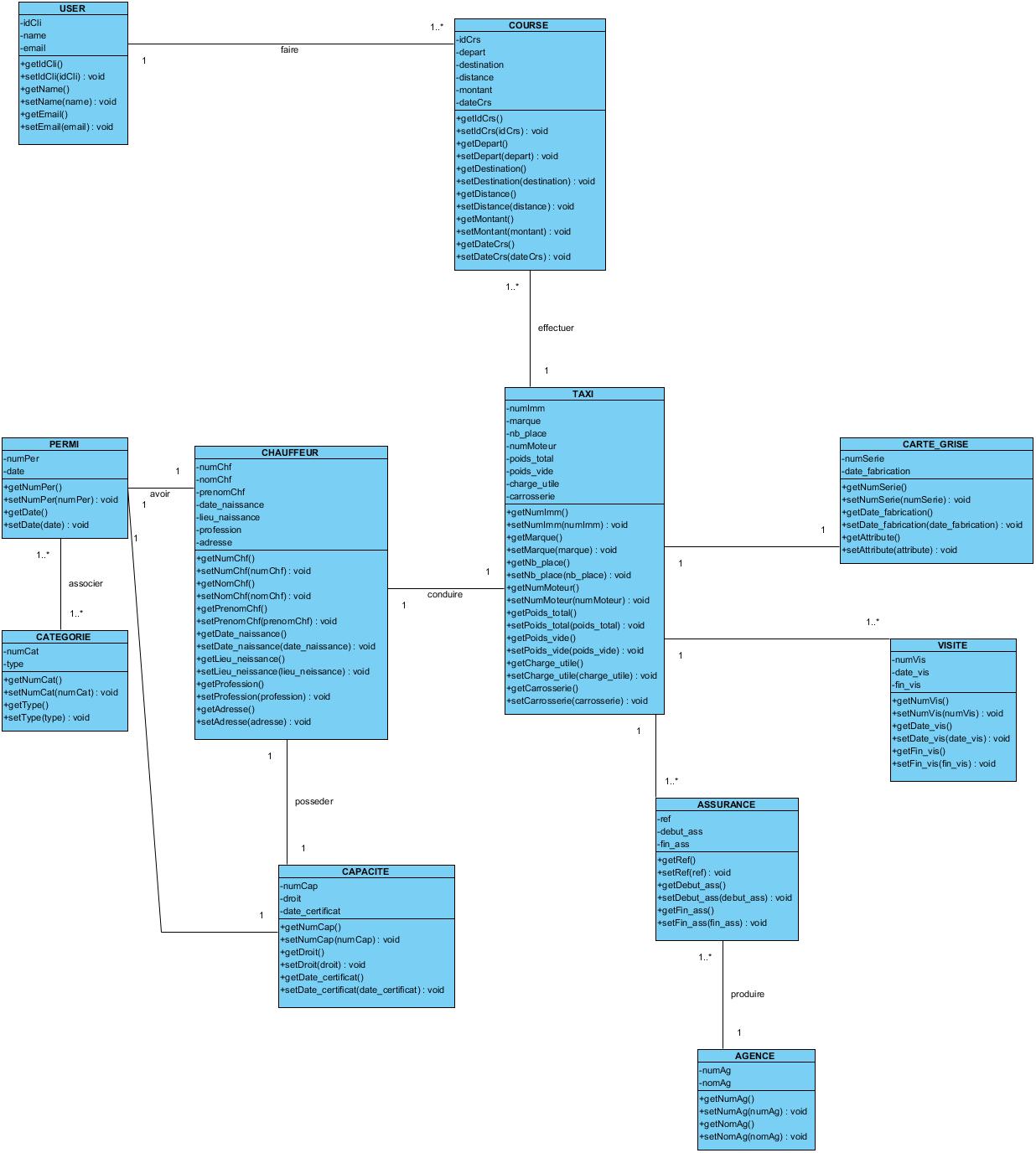
La figure 22 montre le diagramme de classe de conception du cas d’utilisation « Gérer chauffeurs ».



*Figure 22: Diagramme de classe de conception du cas d'utilisation "Gérer chauffeurs".*

### **Diagramme de classe de conception global**

La figure 23 représente le diagramme de classe de conception global.



*Figure 23: Diagramme de classe de la conception globale.*

### **Diagramme de paquetages**

Un diagramme de paquetage est en général un diagramme en boîte qui représente les différents paquetages (ensembles de classes et de composants) et les relations qui existent entre eux. Plus précisément, le diagramme de paquetages permet de visualiser l’organisation des différents éléments du système et de comprendre comment ils interagissent entre eux.

La figure 24 représente le diagramme de paquetage.

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

*Figure 24: Diagramme de paquetage.*

### **Diagramme de déploiement**

Le diagramme de déploiement est un type de diagramme qui montre la structure physique d’un système et comment les différents composants du système interagissent. Il peut être utilisé pour décrire l’architecture d’un système, ainsi que la façon dont le système est déployé sur différents équipements et réseaux.

La figure 25 représente le diagramme de déploiement.

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

*Figure 25: Diagramme de déploiement.*

# **REALISATION**

## **Mise en place de l’environnement de développement**

La mise en place de l’environnement de développement consiste à configurer tous les outils et logiciels nécessaires pour développer l’application.

### **Installation et configuration des outils**

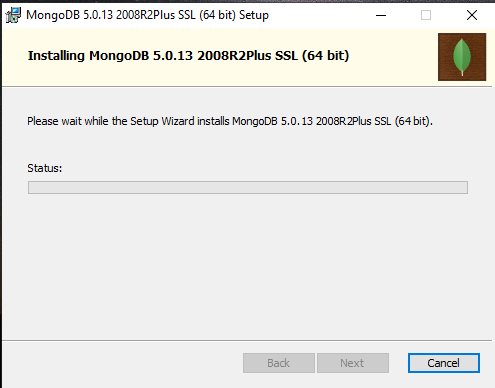
Pour mener à bien notre projet, nous devons installer les outils suivants :

* MongoDB pour un serveur.
* Framework Django.
* Visual Studio Code comme éditeur de texte.
* Firefox comme navigateur.
* Visual Paradigm 8.0 comme outil de conception.

1. **Installation de MongoDB**

MongoDB est un système de gestion de base de données NoSQL open-source qui utilise un modèle de données basé sur des documents. Cela signifie que les données sont stockées sous forme de documents JSON au lieu de tables relationnelles comme dans les bases de données relationnelles.

La figure 26 représente l’installation du MongoDB.



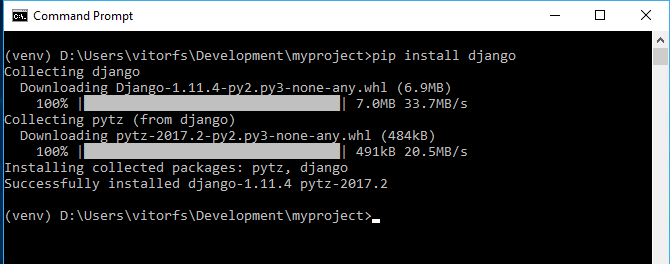
*Figure 26: Installation du MongoDB.*

Après avoir exécuté quelques instructions, l’installation poursuit toute seule et s’achève.

1. **Installation du Framework Django**

Pour installer Django, vous avez besoin d’avoir Python installé sur votre ordinateur. Vous pouvez le télécharger à partir du site web officiel de Python : <https://www.python.org/downloads/> . Une fois que vous avez installé Python, ouvrez une invite de commande ou un terminal et utilisez la commande pip (gestionnaire de paquets Python) pour installer Django.

La figure 27 représente l’installation du Framework Django.

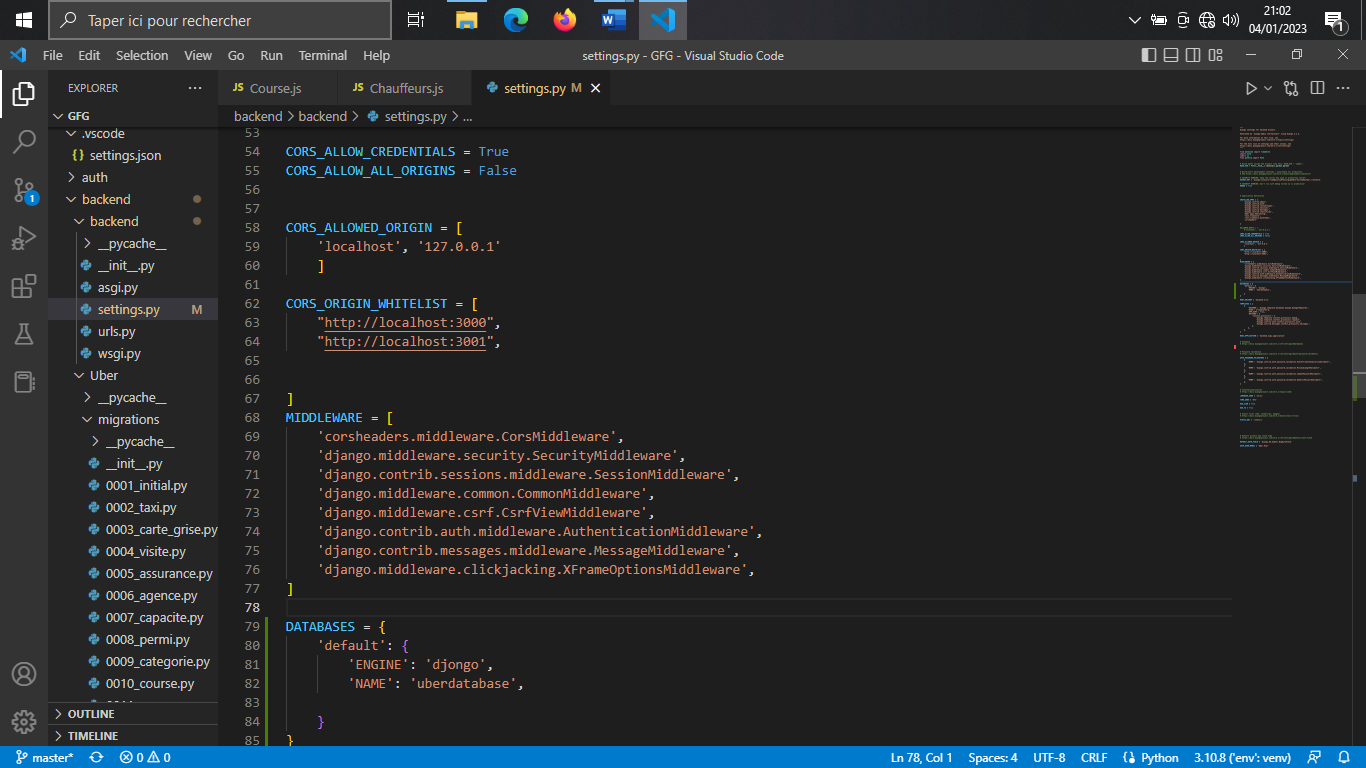


*Figure 27: Installation de Django.*

1. **Installation de Visual Studio Code**

Pour installer l’éditeur Visual Studio Code, il faut aller dans le site officiel <https://code.visualstudio.com/> , ensuite il y a une option donwload et choisir selon le système d’exploitation utilisé. Après l’installation, on a besoin de configurer les éléments suivants dans le fichier « settings.py » de notre projet Django.

La figure 28 montre quelques configurations à faire pour bien mener notre projet.

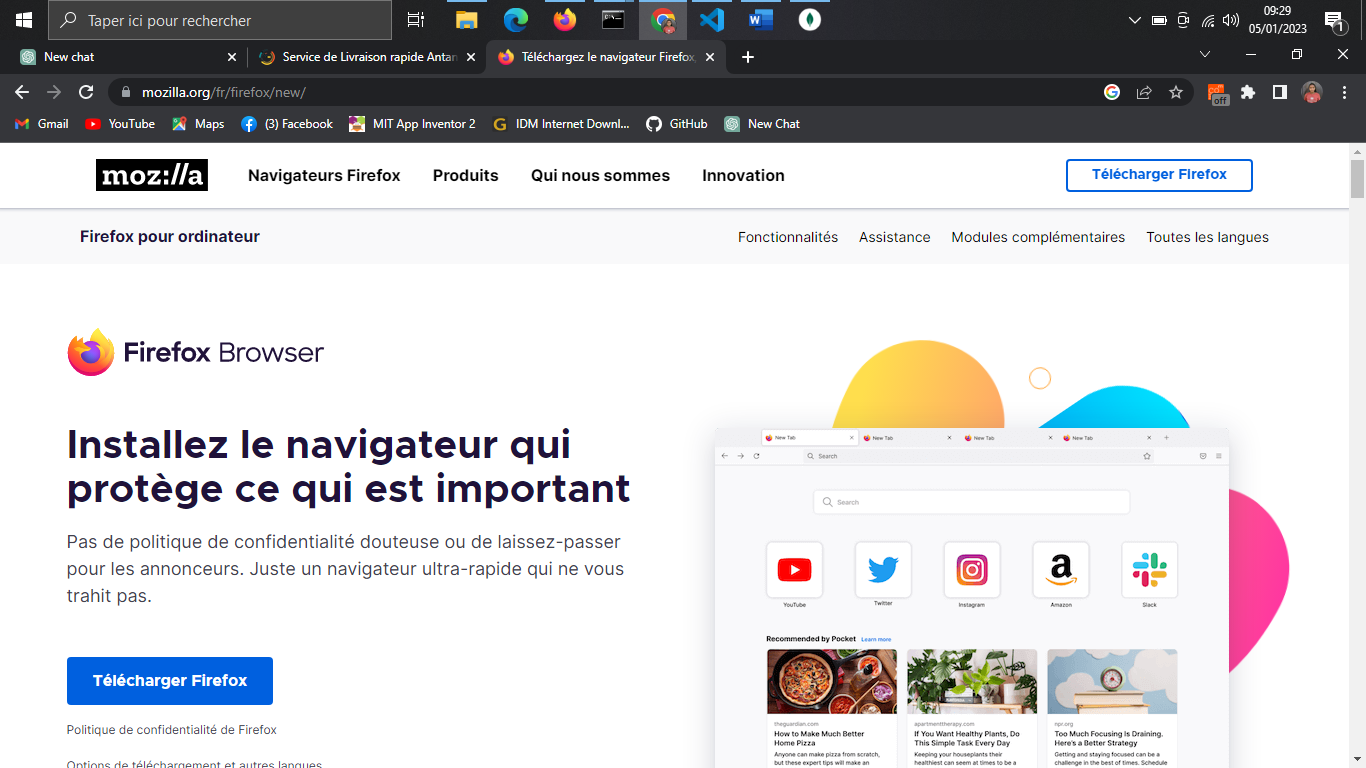


*Figure 28: Configuration à faire dans Visual Studio Code.*

1. **Installation de Firefox**

Il faut télécharger le programme d’installation de Firefox sur son site officiel <https://www.mozilla.org/fr/firefox/new/> . Ensuite, exécuter le fichier téléchargé et l’installation se poursuit en téléchargeant les fichiers nécessaires pour le bon fonctionnement de Firefox.

La figure 29 représente l’interface de Firefox.

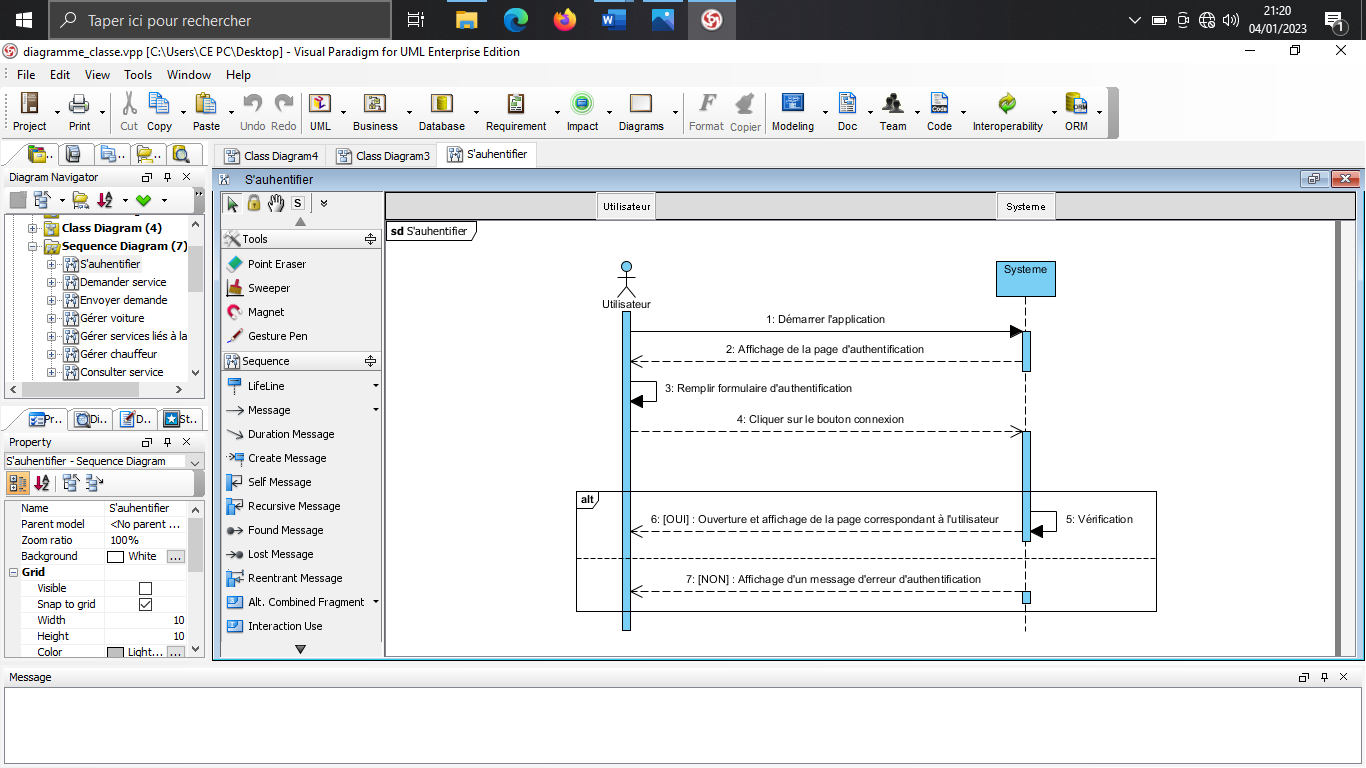


*Figure 29: Interface de Firefox.*

1. **Installation de Visual Paradigm**

Pour l’installation de Visual Paradigm, il faut télécharger l’exécutable et une fois télécharger, on exécute le fichier téléchargé. Une fois l’installation terminée, Visual Paradigm est prêt pour notre conception.

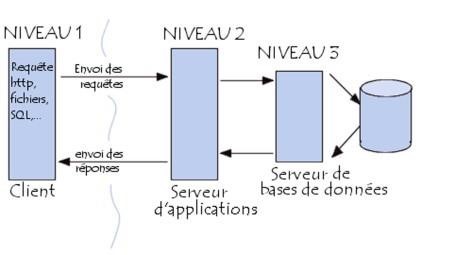
La figure 30 montre un aperçu de l’interface de Visual Paradigm.



*Figure 30: Interface de Visual Paradigm.*

### **Architecture de l’application**

L’application se base sur l’architecture à trois niveaux ou 3-tiers composée de base de données, de serveur d’application et de poste client, illustrée par la figure 31 :

 *Figure 31: Architecture de l'application.*

* Le client : C’est le demandeur de ressources, équipé d’une interface utilisateur.
* Le serveur : Le serveur chargé de fournir la ressource.
* Le serveur de bases de données : Fournissant au serveur d’application les données dont il a besoin.

Les avantages de l’architecture 3-tiers sont les suivants :

* Modularité et réutilisabilité : En séparant l’application en différentes couches, il est plus facile de réutiliser chaque couche indépendamment des autres, ce qui permet de développer des applications plus rapidement et de manière plus efficiente.
* Sécurité : Il est possible de mieux protéger les données de l’application contre les accès non autorisés.
* Facilité de maintenance : Il est plus facile de localiser et de corriger les erreurs et de mettre à jour chaque couche indépendamment des autres.
* Meilleure évolutivité : Il est plus facile de l’évoluer et de l’adapter à de nouvelles exigences sans affecter les autres couches de l’application.

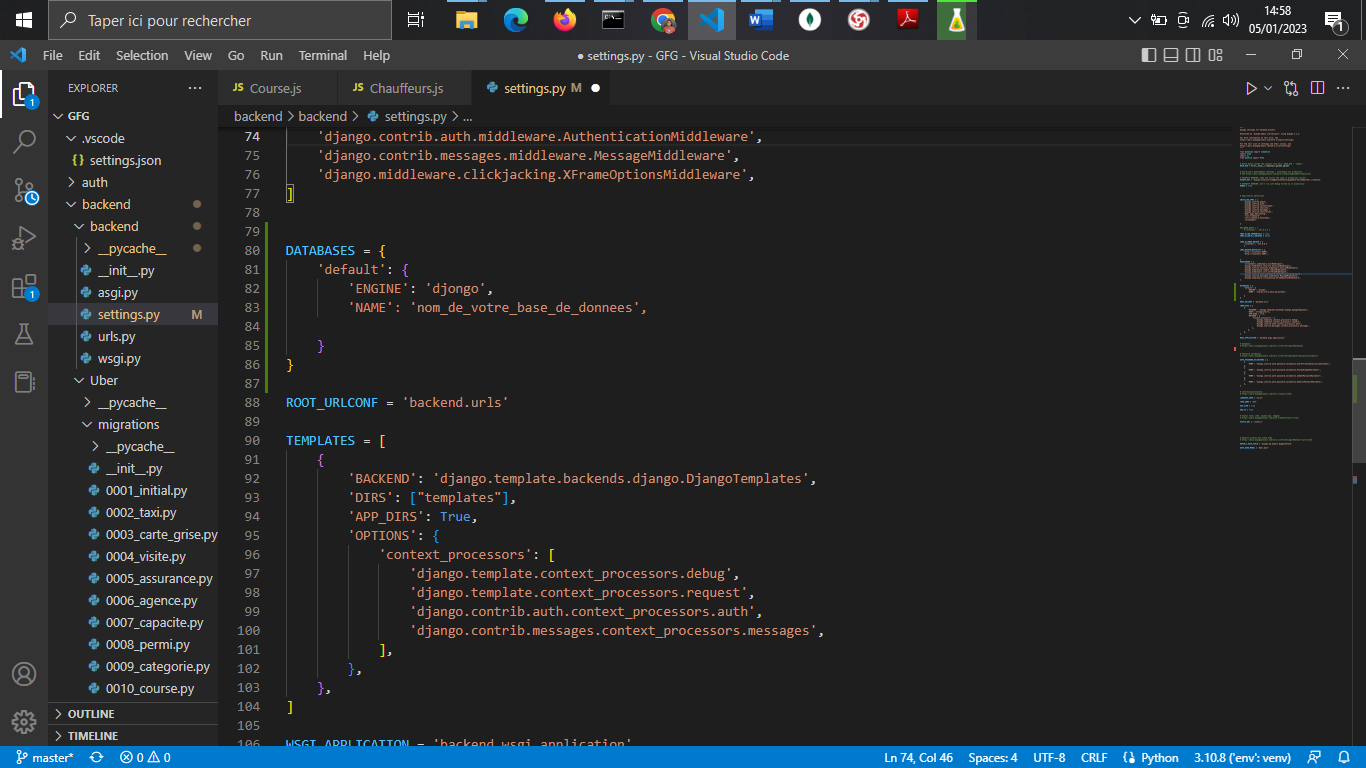
## **Développement de l’application**

### **Création de la base de données**

Pour utiliser MongoDB avec Django, vous pouvez utiliser le package djongo qui traduit les requêtes Django en requêtes MongoDB.

Voici comment configurer Django pour utiliser MongoDB :

1. Installez djongo en exécutant ‘pip install djongo’ dans votre terminal.
2. Dans le fichier ‘settings.py’, ajoutez djongo à la liste ‘INSTALLED\_APPS’.
3. Définissez ‘ DATABASES‘ comme le montre la figure 32 dans le fichier ‘settings.py’.



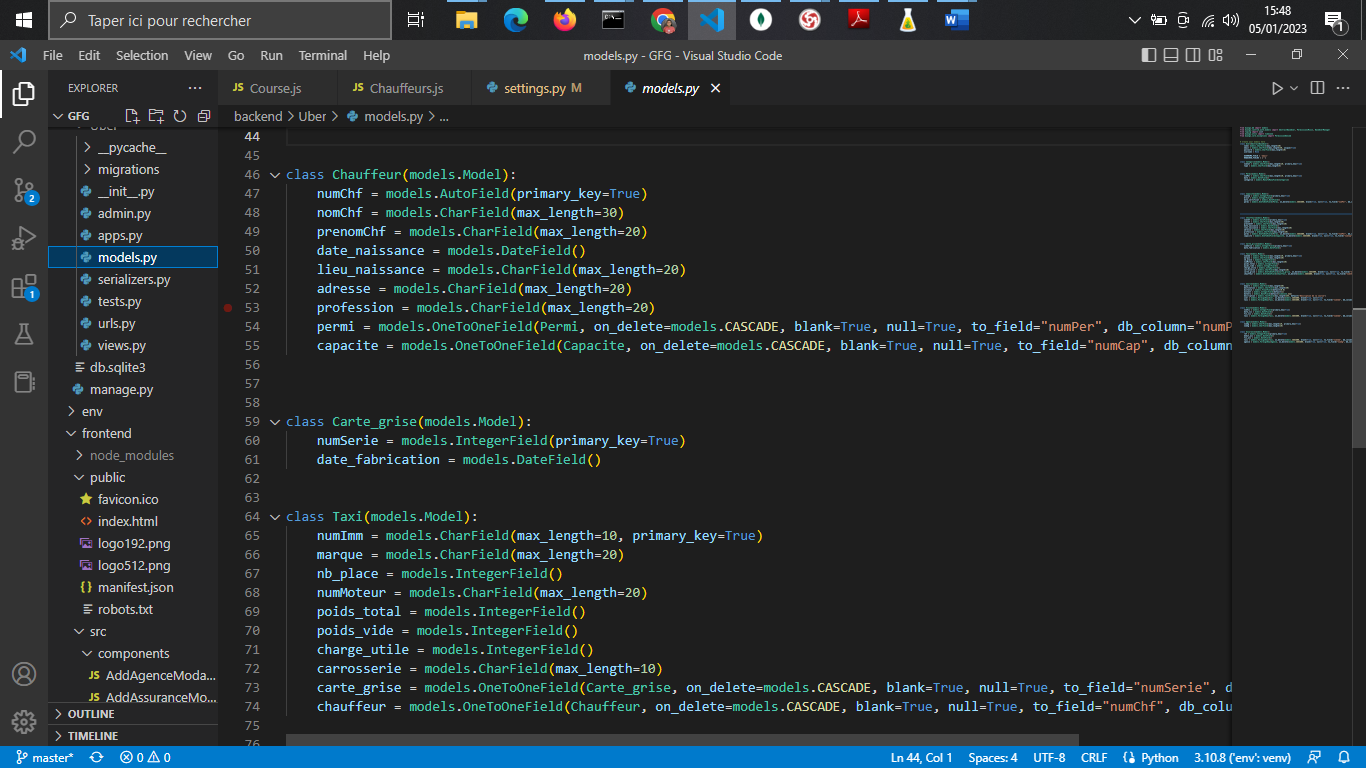
*Figure 32: Création de la base de données*

Une fois vous avez configuré Django, vous pouvez utiliser les commandes :

* « python manage.py makemigrations » : Pour créer les fichiers de migration qui décrivent les changements à apporter à votre base de données.
* «python manage.py migrate » : Pour appliquer les changements à votre base de données.

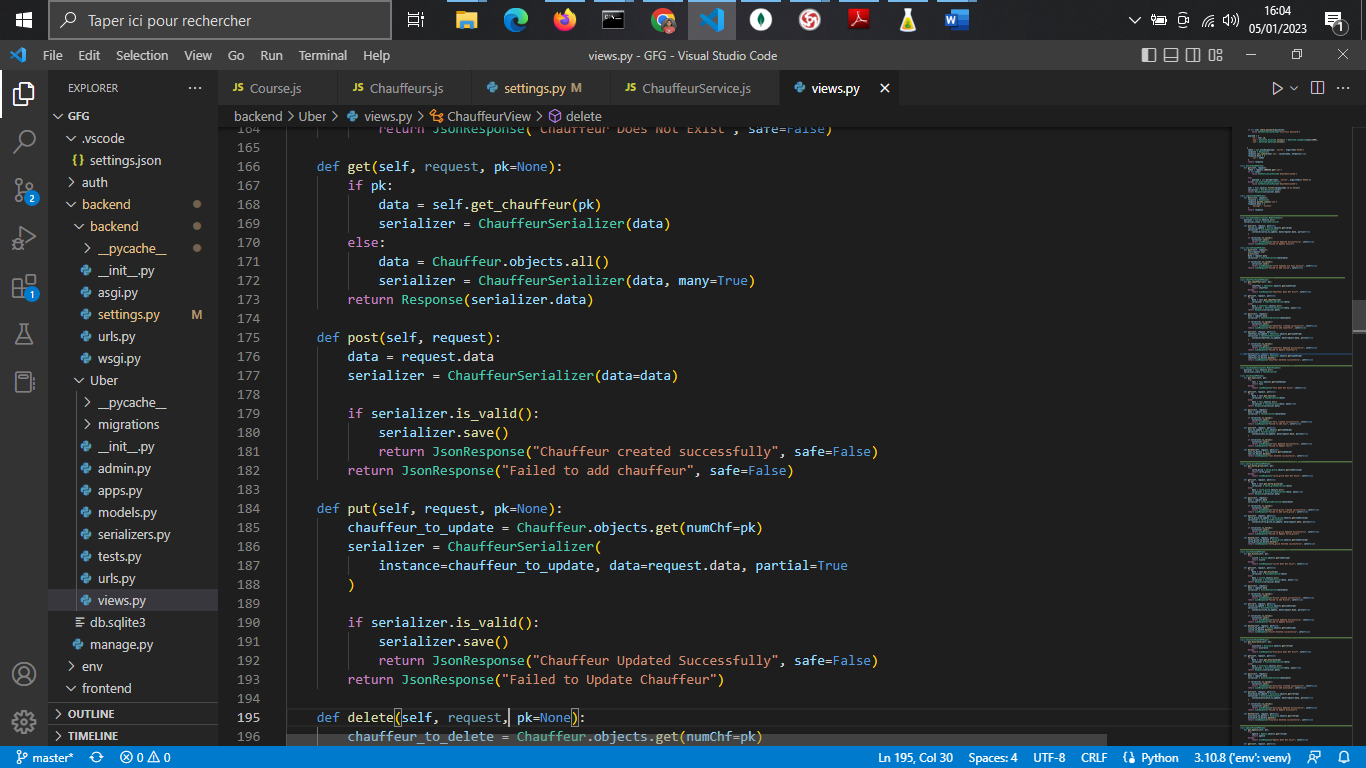
### **Codage de l’application**

La figure 33 montre les codes des différents modèles de notre projet.



*Figure 33: Différentes modèles de notre projet*

La figure 34 affiche un fragment de code concernant la modification d’un chauffeur.



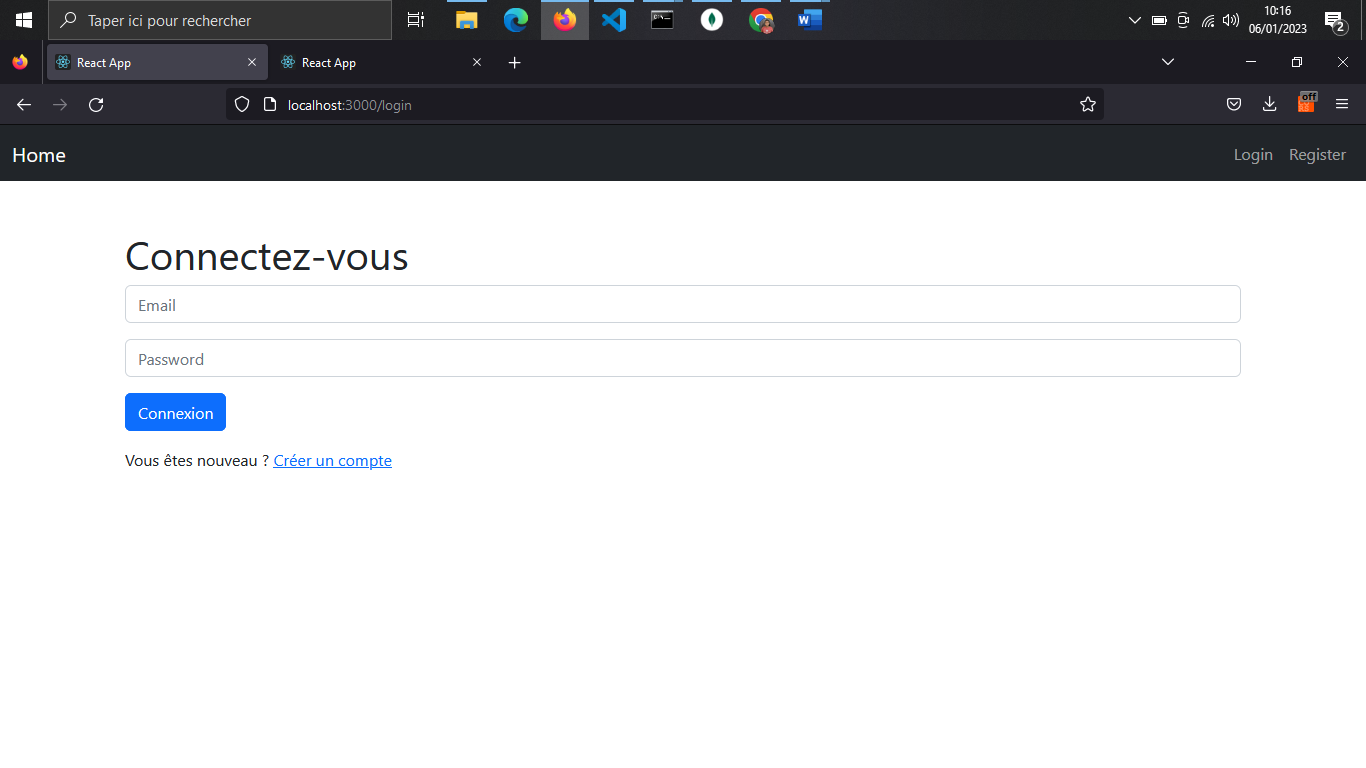
*Figure 34: Modification d'un chauffeur.*

### **Présentation de l’application (Capture d’écran)**

1. **Page d’authentification**

L’authentification est une étape primordiale que l’utilisateur de notre application doit y passer pour faire leur réservation. Cette phase assure, en effet, la sécurité de l’application. L’utilisateur doit s’authentifier en entrant son adresse email et son mot de passe.

La figure 35 décrit la page d’authentification.



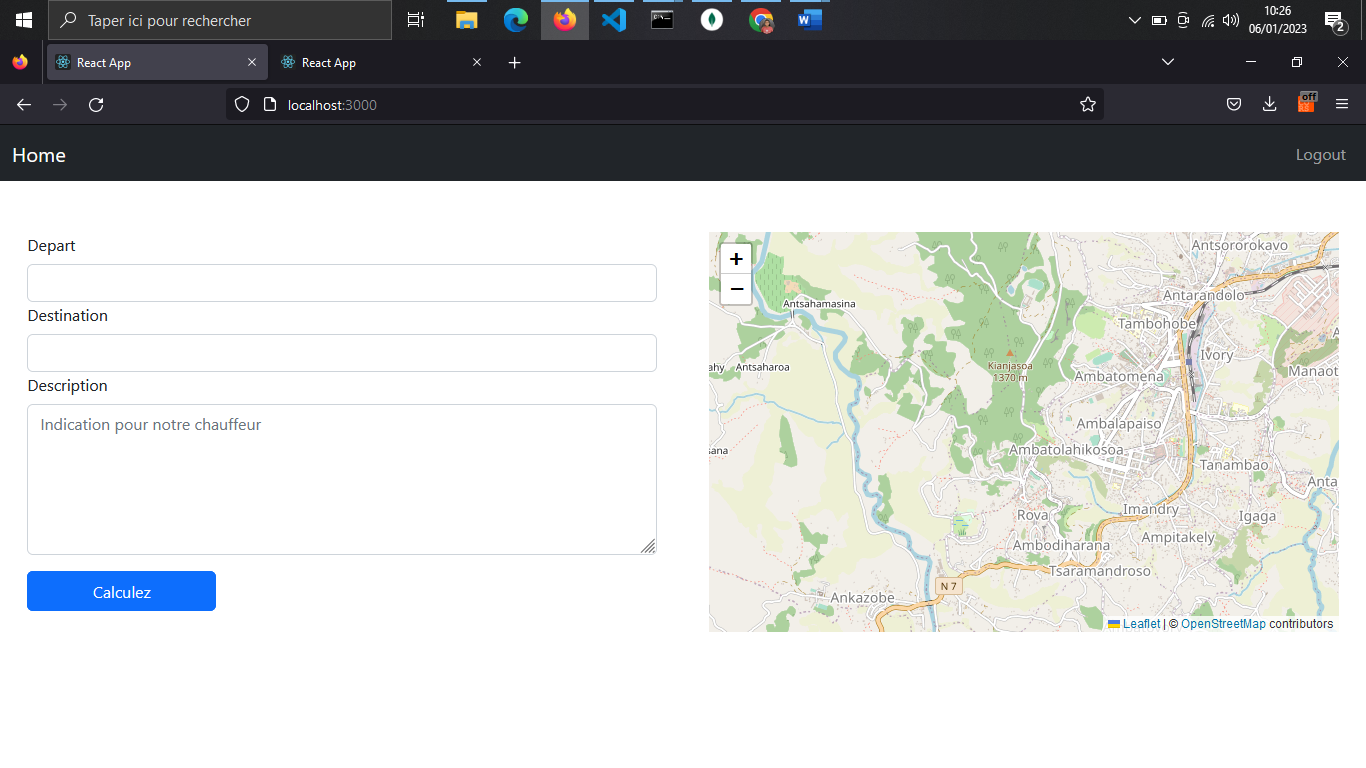
*Figure 35: Page d'authentification.*

1. **Page de réservation**

C’est dans cette page de réservation que l’utilisateur de notre application devra fournir des informations sur son point de départ et sa destination, ainsi que sur le type de véhicule souhaité et le nombre de place requis. Il devra également indiquer ses coordonnées de contact afin que le service de taxi puisse le contacter pour confirmer sa réservation.

Une fois que tous les champs requis soient complétés, l’utilisateur doit cliquer sur le bouton calculer et un autre formulaire s’affiche pour leur informer du tarif de leur course et une estimation de l’arrivé de leur taxi.

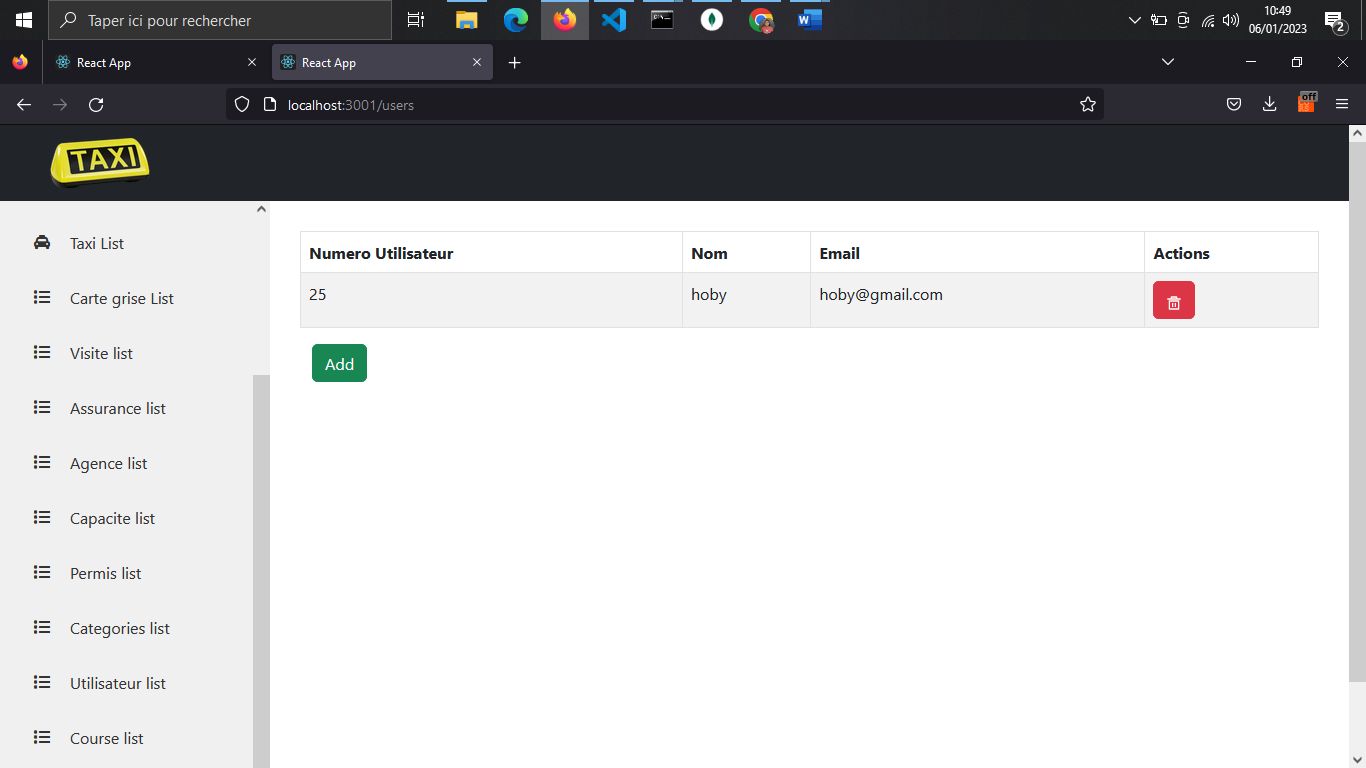
La figure 36 nous permet de montrer la page de réservation.



*Figure 36: Page de réservation.*

1. **BackOffice**

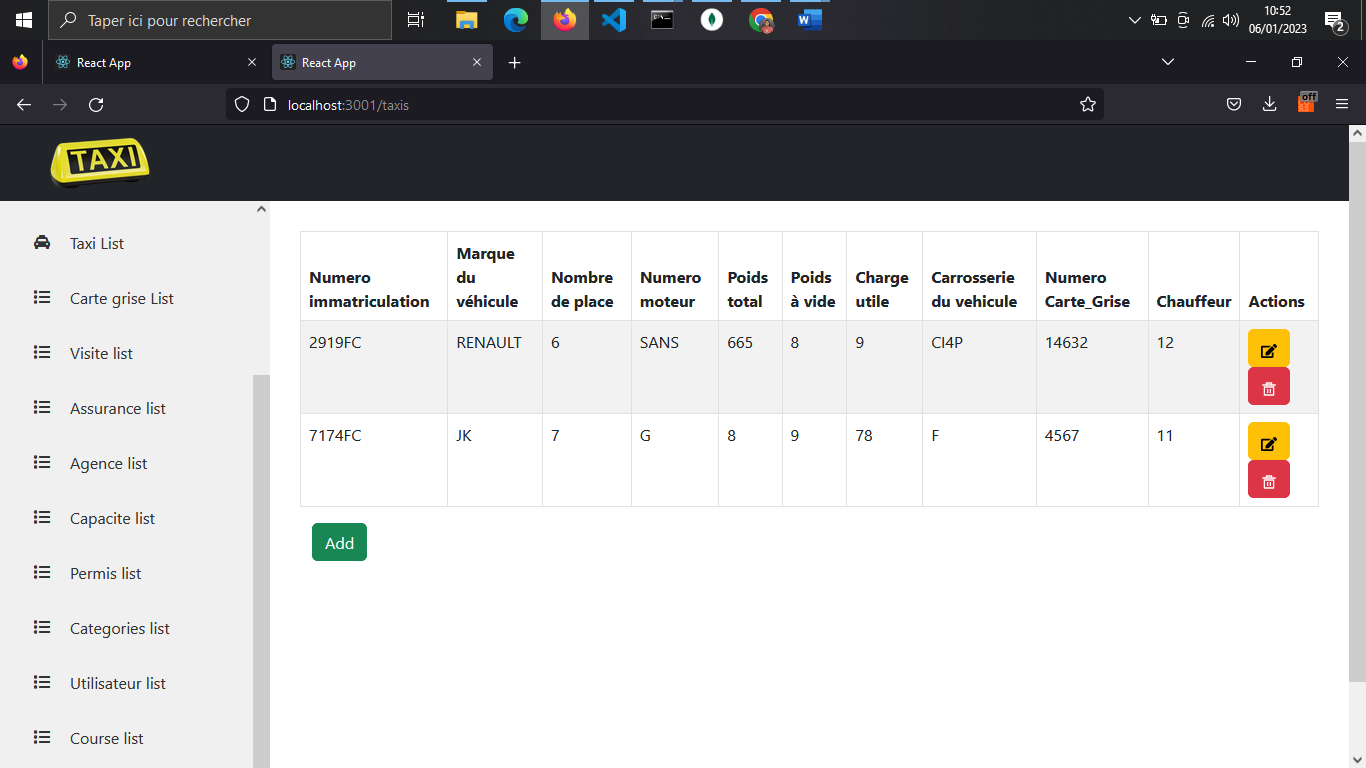
L’administrateur gère les utilisateurs de l’application et les mis à jour. La figure 37 montre la liste des utilisateurs.



*Figure 37: Liste des utilisateurs.*

L’administrateur gère aussi les voitures ainsi que les différents services qui sont en liaison avec elles comme les chauffeurs avec leurs permis respectifs, la carte grise, la validité de chaque assurance et visite, etc.

On a décidé d’illustrer la liste des taxis dans la figure 38 ci-dessous :



*Figure 38: Liste des taxis.*

# **CONCLUSION**

Au bout du stage passé au sein de la société Arato, la réalisation d’une plateforme pour la réservation de taxi en ligne a pu être menée à son terme. Ce projet nous a permis d’avancer dans le monde de programmation grâce à la découverte des nouvelles instructions.

Pour la conception de notre application, on a utilisé le langage de modélisation UML en adoptant la méthode 2TUP. Afin d’assurer la qualité de l’application, MongoDB a été utilisé comme SGBD. Pour le développement de l’application, on a choisi Python couplé avec le framework Django et pour l’interface de l’application on a opté pour React js.

L’application est opérationnelle et exploitable pour les services dédiés à son usage. En perspective, l’amélioration dans le futur est toujours envisageable en ajoutant certaines fonctionnalités comme l’historisation des informations à propos d’une course effectuée par l’utilisateur, l’implémentation d’une gestion de paie du côté client.

Ce stage nous a été d’un très grand bénéfice dans le domaine professionnel dans l’Entreprise parce qu’il nous a permis d’approfondir nos connaissances pratiques en matière de développement d’application web.

# **BIBLIOGRAPHIE**

[1] Gérard Swinnen, Apprendre à programmer avec Python 3, 2000-2012, 473 pages.

[2] Julie Steele, MongoDB : The Definitive Guide, First Edition, September 2010, 193 pages.

[3] Mathieu Xhonneux(MathX) et Maxime Lorant(Ssx’z), Développez votre site web avec le framework Django, 2/04/2013, 204 pages.

# **WEBOGRAPHIE**

[1] <https://www.python.org/> , Documentation sur Python 3, consulté en Octobre 2022.

[2] <https://docs.djangoproject.com/en/4.1/> , Documentation sur Django, consulté en Octobre 2022.

[3] <https://reactjs.org/> , Documentation sur React, consulté en Novembre 2022.

# **GLOSSAIRE**

**API :** Un ensemble de règles et de protocoles qui définissent comment différents systèmes logiciels peuvent communiquer entre eux. Les API sont généralement constitués d’un ensemble de fonctions, de méthodes ou d’autres procédures qui peuvent être appelées par d’autres systèmes logiciels.

**Back office :** Fait référence aux parties d’une entreprise ou d’une organisation qui ne sont pas directement visibles pour les clients ou les utilisateurs finaux.

**Framework :** Une structure de base qui fournit un ensemble de fonctionnalités et de services communs pour faciliter le développement d’applications ou de logiciels.

**Front-end :** Fait référence aux parties d’une application ou d’un site web qui sont directement visibles et accessibles par les utilisateurs.

**Plateforme :** Environnement permettant la gestion de services applicatifs.

**Serveur :** Un ordinateur ou un système de logiciels qui fournit des services ou des fonctionnalités à d’autres ordinateurs ou dispositifs sur un réseau.

# **TABLE DES MATIERES**

[CURRICULUM VITAE I](#_Toc123916206)

[SOMMAIRE GENERAL IV](#_Toc123916207)

[REMERCIEMENTS VI](#_Toc123916208)

[LISTE DES FIGURES VII](#_Toc123916209)

[LISTE DES TABLEAUX VIII](#_Toc123916210)

[LISTE DES ABREVIATIONS IX](#_Toc123916211)

[INTRODUCTION GENERALE 1](#_Toc123916212)

[PARTIE I. PRESENTATIONS 2](#_Toc123916213)

[Chapitre 1. Présentation de l’Ecole Nationale d’Informatique 3](#_Toc123916214)

[1. 1. Information d’ordre général 3](#_Toc123916215)

[1. 2. Missions et historique 3](#_Toc123916216)

[1. 3. Organigramme institutionnel de l’ENI 5](#_Toc123916217)

[1. 4. Domaine de spécialisation 7](#_Toc123916218)

[1. 5. Architecture des formations pédagogiques 7](#_Toc123916219)

[1. 6. Relations de l’ENI avec les entreprises et les organismes 10](#_Toc123916220)

[1. 7. Partenariat au niveau international 12](#_Toc123916221)

[1. 8. Débouchés professionnels avec des diplômés 14](#_Toc123916222)

[1. 9. Ressources humaines 16](#_Toc123916223)

[Chapitre 2. Présentation de la société d’accueil 17](#_Toc123916224)

[2.1. Fiche d’identification 17](#_Toc123916225)

[2.2. Brèves historiques 17](#_Toc123916226)

[2.3. Objectifs : Missions et Activités 17](#_Toc123916227)

[2.4. Organigramme de ARATO 18](#_Toc123916228)

[2.5. Ressources humaines 19](#_Toc123916229)

[2.6. Partenaire et Bienfaiteurs 19](#_Toc123916230)

[Chapitre 3. Description du projet 20](#_Toc123916231)

[3. 1. Formulation 20](#_Toc123916232)

[3. 2. Objectif et besoins de l’utilisateur 20](#_Toc123916233)

[3. 3. Moyens nécessaires à la réalisation du projet (Humain, matériel et éventuellement financier) 21](#_Toc123916234)

[3. 4. Résultats attendus 21](#_Toc123916235)

[PARTIE II. ANALYSE ET CONCEPTION 23](#_Toc123916236)

[Chapitre 4. Analyse préalable 24](#_Toc123916237)

[4. 1. Analyse de l’existant 24](#_Toc123916238)

[4.1. 1. Organisation actuelle 24](#_Toc123916239)

[4.1. 2. Inventaire des moyens matériels et logiciels 24](#_Toc123916240)

[4. 2. Critique de l’existant 24](#_Toc123916241)

[4. 3. Conception avant-projet 25](#_Toc123916242)

[4.3.1. Propositions de solutions 25](#_Toc123916243)

[4.3.2. Solution retenue 26](#_Toc123916244)

[4.3.3. Méthodes et outils utilisés 26](#_Toc123916245)

[4. 4. Présentation de la méthode 2TUP 33](#_Toc123916246)

[Chapitre 5. Analyse conceptuelle 37](#_Toc123916247)

[5. 1. Dictionnaire des données 37](#_Toc123916248)

[5. 2. Règles de gestion 39](#_Toc123916249)

[5. 3. Représentation et spécification des besoins 40](#_Toc123916250)

[5.3. 1. Diagrammes des cas d’utilisation 40](#_Toc123916251)

[5.3. 2. Priorisation des cas d’utilisation 43](#_Toc123916252)

[5.3. 3. Diagramme des séquences système pour chaque cas d’utilisation 50](#_Toc123916253)

[5. 4. Spécification des besoins techniques 50](#_Toc123916254)

[5. 5. Modélisation du domaine (Modèle de domaine) 50](#_Toc123916255)

[Chapitre 6. Conception détaillée (UML) 52](#_Toc123916256)

[6. 1. Architecture du système 52](#_Toc123916257)

[6. 2. Diagramme de séquence de conception pour chaque cas d’utilisation 53](#_Toc123916258)

[6. 3. Diagramme de classe de conception pour chaque cas d’utilisation 57](#_Toc123916259)

[6. 4. Diagramme de classe de conception global 59](#_Toc123916260)

[6. 5. Diagramme de paquetages 60](#_Toc123916261)

[6. 6. Diagramme de déploiement 60](#_Toc123916262)

[PARTIE III. REALISATION 61](#_Toc123916263)

[Chapitre 7. Mise en place de l’environnement de développement 62](#_Toc123916264)

[7 .1. Installation et configuration des outils 62](#_Toc123916265)

[7 .2. Architecture de l’application 65](#_Toc123916266)

[Chapitre 8. Développement de l’application 67](#_Toc123916267)

[8 .1. Création de la base de données 67](#_Toc123916268)

[8 .2. Codage de l’application 68](#_Toc123916269)

[8 .3. Présentation de l’application (Capture d’écran) 69](#_Toc123916270)

[CONCLUSION X](#_Toc123916271)

[BIBLIOGRAPHIE XI](#_Toc123916272)

[WEBOGRAPHIE XII](#_Toc123916273)

[GLOSSAIRE XIII](#_Toc123916274)

[TABLE DES MATIERES XIV](#_Toc123916275)

[RESUME XVII](#_Toc123916276)

[ABSTRACT XVII](#_Toc123916277)

# **RESUME**

Cet ouvrage renferme mes expériences au cours du stage effectué au sein de la société Arato. Ce stage a pour objectif de faire la conception et la réalisation d’une plateforme de réservation de taxi en ligne.

Pour réaliser ce projet, on a utilisé la méthode 2TUP, le langage UML l’outil de conception Visual Paradigm, le langage de script Python, le SGBD MongoDB.

Cette application permettra la gestion des voitures ainsi que ses différents services, la gestion des chauffeurs, des cartes et des tarifications.

Mots-clés : 2TUP, UML, SGBD.

# **ABSTRACT**

This work contains my experiences during the internship at Arato company. The internship aims to design and develop an online taxi booking platform.

To complete this project, we used the 2TUP method, the UML language, the Visual Paradigm design tool, the Python scripting language and the MongoDB database.

This application will manage cars and their various services, as well as drivers, cards and tariffs.

Keywords : 2TUP, UML, DBMS.