

CURRICULUM VITAE

ETAT CIVIL

RAKOTOZAFY Nantenaina Jérémiah

📍 Face bureau Fokontany
Tambohobe-Fianarantsoa
Madagascar

☎ +261 34 78 668 89

@ nantenainajeremiah@gmail.com

18 ans, célibataire



FORMATION ET DIPLOME

2022 – 2023

- **Ecole Nationale d'Informatique Fianarantsoa**
Première année en formation de Licence professionnelle (Génie logiciel et Base de données).

2021 – 2022

- **Lycée Ambatolampy**
Baccalauréat scientifique (série S)

2021

- **Alliance Française Andavamamba**
Diplôme d'étude de Langue Française (DELF B2)

COMPETENCES EN INFORMATIQUE

| | |
|---------------------------------|----------------------|
| Système d'exploitation | Windows, Linux |
| Langage de programmation | C, C++, Bash shell |
| Développement Web | HTML/CSS, JavaScript |
| Framework | Qt |
| SGBD | SQLite |

COMPETENCES LINGUISTIQUE

| | |
|-----------------|-------------------|
| Malagasy | Langue maternelle |
| Français | Avancé |
| Anglais | Intermédiaire |
| Russe | Débutant |

DIVERS

| | |
|-------------------|---------------|
| Musique | Piano, Violon |
| Lecture | |
| Jeux video | |

CURRICULUM VITAE

ETAT CIVIL

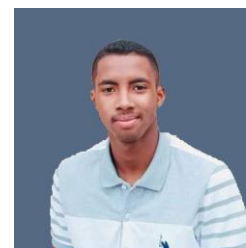
RAMANDIMBIHANISOA Ny Aina Haritiana

📍 Lot 071 AD/3602
Ankofafa-Fianarantsoa
Madagascar

☎ +261 34 61 716 42

@ ainaramandimby@gmail.com

21 ans, célibataire



FORMATION ET DIPLOME

2022 – 2023

- **Ecole Nationale d'Informatique Fianarantsoa**
Première année en formation de Licence professionnelle (Génie logiciel et Base de données).

2020 – 2021

- **IEF Aceem**
Baccalauréat scientifique (série C)

COMPETENCES EN INFORMATIQUE

Système d'exploitation Windows, Linux

Langage de programmation C, C++, Bash shell

Développement Web HTML/CSS, JavaScript

Framework Qt, JQuery

SGBD SQLite, MySQL

Logiciels Adobe Photoshop, Illustrator, After Effects, Premiere Pro

COMPETENCES LINGUISTIQUE

Malagasy Langue maternelle

Français Avancé

Anglais Intermédiaire

DIVERS

Sport Natation, Basket-ball, Foot-ball

Lecture

Jeux video

CURRICULUM VITAE

ETAT CIVIL

RAZANAMARA Haingomanana Tahiry

📍 Lot 116 A2 3710
Ampitakely-Fianarantsoa
Madagascar
☎ +261 34 55 958 62
@ tahiryrazanamara01@gmail.com
21 ans, célibataire



FORMATION ET DIPLOME

2022 – 2023

- **Ecole Nationale d'Informatique Fianarantsoa**
Première année en formation de Licence professionnelle (Génie logiciel et Base de données).

2019 – 2020

- **Lycée Pôle Fort-Dauphin**
Baccalauréat scientifique (série D)

COMPETENCES EN INFORMATIQUE

| | |
|---------------------------------|----------------------------|
| Système d'exploitation | Windows, Linux |
| Langage de programmation | C, C++, Bash shell, Python |
| Développement Web | HTML/CSS, JavaScript |
| Framework | Qt, JQuery, Bootstrap |
| SGBD | SQLite, MySQL |
| Gestion de version | Git |

COMPETENCES LINGUISTIQUE

| | |
|-----------------|-------------------|
| Malagasy | Langue maternelle |
| Français | Avancé |
| Anglais | Intermédiaire |

DIVERS

| | |
|-------------------|------------------------|
| Sport | Basket-ball, Foot-ball |
| Lecture | |
| Jeux video | |

REMERCIEMENTS

Pour la réalisation de ce projet, nous tenons à exprimer notre profonde reconnaissance pour le dévouement, l'expertise et le soutien de tous les responsables pédagogiques tout au long de notre parcours.

Nous tenons particulièrement à remercier :

Monsieur HAJALALAINA Aimé Richard, Docteur HDR, Président de l'Université de Fianarantsoa car son engagement envers les étudiants, les enseignants et les personnels a créé un environnement d'apprentissage dynamique qui enrichi nos vies universitaires.

Monsieur MAHATODY Thomas, Docteur HDR, Directeur de l'Ecole Nationale d'Informatique (ENI), Université de Fianarantsoa pour sa contribution dans la direction et pour nous avoir donné l'opportunité de terminer notre première année de licence professionnelle avec la réalisation de ce projet.

Monsieur RABETAFIKA Louis Haja, Maître de Conférences, Chef de mention informatique à l'ENI, qui supervise notre parcours académique avec son travail acharné au sein de notre mention.

Monsieur RALAIVAO Jean Christian, Assistant d'Enseignement Supérieur et de Recherche, responsable du parcours Génie logiciel et Base de données (GB) à l'ENI, qui à contribuer au succès du parcours GB, son effort a été essentiel pour façonner la voie de nos études vers une excellence académique.

Monsieur DIMBISOA William Germain, Maître de Conférences, Encadreur et conseillé pour l'innovation du projet, qui nous a guidé à la réalisation de notre projet et d'avoir accepté d'en être le rapporteur.

Les enseignants de l'Ecole Nationale d'Informatique, de nous avoir donné les connaissances de base en informatique et de nous préparer à un impact positif sur notre future carrière.

Les personnels et l'équipe pédagogique, qui ont travaillé en harmonie pour assurer le bon fonctionnement de notre programme, en passant par le soutien administratif indispensable.

Tous les membres de jury, de nous apporter une aide précieuse à l'amélioration de notre projet.

Nos parents respectifs, d'avoir investi pour qu'on puisse poursuivre nos études supérieures, de nous avoir permis à se nourrir correctement et de nous soutenir dans la vie pour poursuivre notre rêve.

Nous tenons également à exprimer notre reconnaissance envers notre équipe de projet.

Avec toutes nos gratitude.

LISTE DES FIGURES

| | |
|--|----|
| Figure 1: Modèle Conceptuel des données | 12 |
| Figure 2: Premier fenêtre d'installation | 14 |
| Figure 3: Indiquer le chemin d'installation | 14 |
| Figure 4: Sélection des composants nécessaires | 15 |
| Figure 5: Installation puis terminer | 15 |
| Figure 6: Entête de la classe coopérative | 17 |
| Figure 7: Entête de la classe gestionTrajet..... | 18 |
| Figure 8: Entête de la classe trajet..... | 18 |
| Figure 9: Entête de la classe véhicule | 19 |
| Figure 10: Entête de la classe réservation | 20 |
| Figure 11: Page d'accueil..... | 22 |
| Figure 12: Interface du côté coopératif | 23 |
| Figure 13: Interface du côté coopératif | 23 |
| Figure 14: Interface du côté passager..... | 24 |
| Figure 15: Historique des voyages qui étaient déjà fait | 25 |
| Figure 16: A propos du logiciel | 26 |

LISTE DES TABLEAUX

| | |
|---|---|
| Tableau 1: Récapitulatif des moyens matériels et logiciels existants. | 4 |
| Tableau 2: Dictionnaire de données de la coopérative | 8 |

LISTE DES ABRÉVIATIONS ET NOMENCLATURE

API : Application Programming Interface (Interface de Programmation d'Application).

BD : Base de données.

ENI : Ecole Nationale d'Informatique.

GB : Génie logiciel et Base de données.

IDE : Integrated Development Environment (Environnement de Développement Intégré).

MCD: Modèle Conceptuel de Données.

MLD: Modèle Logique de Données.

OS : Operating System (Système d'Exploitation).

SGBD : Système de Gestion de Base de Données.

SQL : Structured Query Language (Langage de Requête Structurée).

TMS : Transportation Management System (Système de gestion de transport).

UI: User Interface (Interface Utilisateur).

SOMMAIRE

| | |
|--|-------------|
| CURRICULUM VITAE..... | I |
| REMERCIEMENTS..... | IV |
| LISTE DES FIGURES..... | VI |
| LISTE DES TABLEAUX..... | VII |
| LISTE DES ABRÉVIATIONS ET NOMENCLATURE..... | VIII |

I. ANALYSE

| | | |
|-------|---|---|
| 1 | Description du projet..... | 2 |
| 1.1 | Formulation..... | 2 |
| 1.2 | Objectif et besoins de l'utilisateur | 2 |
| 1.3 | Résultat attendu..... | 2 |
| 2 | Analyse préalable | 3 |
| 2.1 | Analyse de l'existant..... | 3 |
| 2.1.1 | Organisation actuelle : | 3 |
| 2.1.2 | Inventaire des moyens matériels et logiciels : | 3 |
| 2.2 | Critiques de l'existant et proposition de solutions | 4 |

II. CONCEPTION

| | | |
|-----|---|----|
| 3 | Conception avant-projet | 6 |
| 3.1 | Système d'exploitation..... | 6 |
| 3.2 | Langage de programmation | 6 |
| 3.3 | Environnement de développement..... | 7 |
| 4 | Conception | 8 |
| 4.1 | Dictionnaire des données | 8 |
| 4.2 | Règles de gestion | 10 |
| 4.3 | Modèle Conceptuel des Données (MCD) | 11 |
| 4.4 | Modèle Logique des Données (MLD) | 12 |

III. REALISATION

| | | |
|--------------------------|--|------------|
| 5 | Installation et configuration des outils | 14 |
| 6 | Développement de l'application..... | 15 |
| 6.1 | Description de la base de données | 15 |
| 6.2 | Elaboration des algorithmes..... | 16 |
| 6.3 | Quelques captures d'écran | 21 |
| CONCLUSION..... | | 27 |
| BIBLIOGRAPHE..... | | X |
| WEBOGRAPHIE..... | | XI |
| RESUMÉ..... | | XII |
| ABSTRACT..... | | XII |

INTRODUCTION

Actuellement, la numérisation est devenue une nécessité mondiale ; ce dernier est rendu possible grâce aux diverses avancées technologiques. Cela facilite considérablement la manipulation, le contrôle et la sécurité des données.

Ainsi, le cadre de l'enseignement de l'ENI nous a assigné comme projet, la réalisation d'un logiciel de « gestion de transport », dans le but d'assimiler l'aboutissement de nos efforts, de notre apprentissage et de notre passion pour l'informatique.

Mais pourquoi la numérisation des données au niveau de la gestion de trajets d'une coopérative est-elle nécessaire?

La gestion efficace du transport est un élément crucial pour toute entreprise ou organisation qui dépend du mouvement de personnes ou de ressources d'un point à un autre. Cependant le processus manuel et les systèmes obsolètes peuvent entraîner des inefficacités, des retards ou une perte de productivité. C'est là que ce logiciel de gestion de transport entre en jeu, offrant une solution innovante pour optimiser et rationaliser les opérations de déplacements.

Ce rapport vise à présenter en détail le processus de la conception de ce logiciel en mettant l'accent sur les objectifs, les fonctionnalités clé et les défis rencontrés tout au long du projet. Nous avons suivi une approche méthodique, en commençant par une analyse approfondie des besoins et des exigences, puis en concevant une architecture logicielle robuste et fonctionnalités essentielles. Tout au long de ce rapport, nous explorerons les différentes étapes de conception, les technologies utilisées, les méthodes de développement et les tests effectués pour assurer la qualité et la fiabilité de ce logiciel.

Pour mieux cerner le logiciel, ce rapport sera organisé en trois (3) parties, d'abord l'analyse et la description du projet, puis la conception et ensuite la réalisation du logiciel en question.

I. ANALYSE

1 Description du projet

1.1 *Formulation*

Le projet consiste à la réalisation d'un logiciel capable de gérer les activités d'une entreprise de transport, tel que les coopératives. La conception de ce logiciel est réalisée en utilisant le langage de programmation C++ avec le Framework Qt afin d'assimiler l'aboutissement de nos efforts, de notre apprentissage et de notre passion pour l'informatique. Le TMS (Système de gestion de transport) est une tâche compliquée vue toutes les données qui y circule, c'est la raison de l'utilité de ce logiciel capable de faciliter la gestion.

1.2 *Objectif et besoins de l'utilisateur*

L'objectif de ce projet est de permettre la gestion automatique des flux de déplacement des véhicules d'une coopérative quelconque. Plus précisément, cela consiste à stocker des informations sur les actifs d'une coopérative (Véhicule de transport) et de gérer leurs trajets en fonction du temps.

Cette application permettra de :

- Gérer les tours et l'horaire de voyages des véhicules de transport,
- Ajouter, de modifier ou de supprimer un trajet (itinéraire) ou un véhicule,
- Mettre à jour les informations de la Base de données (BD),
- Gérer les réservations des passagers,
- Stocker tous les historiques des voyages effectués dans la Base de données (BD).

1.3 *Résultat attendu*

Cette application a été donc conçu pour établir et afficher une liste de tour et des horaires de départ des véhicules initialement ajouter dans la BD accompagné de leurs itinéraires et de leurs disponibilités horaires. Elle permettra alors de connaître l'emploi du temps des déplacements de chaque véhicule de la coopérative. Ce logiciel a été conçu pour s'adapter à n'importe quelle coopérative, autrement dit, au début du lancement de l'application, les utilisateurs peuvent entrer le nom de leurs coopératives ainsi que la date de leur ouverture (premier voyage). Puis la coopérative doit ajouter les informations des véhicules en leurs possession dans la BD. Et la gestion se fera automatiquement par un simple clic d'un bouton.

Concernant le côté passager, le logiciel permet la suivie et la classification de chaque réservation en fonction de la date de voyage, du trajet et de l'heure de départ. En plus le nombre de place supporter par chaque véhicule peuvent être personnaliser selon le type et le modèle de chaque véhicule. Bref, le logiciel peut permettre d'afficher les listes de passagers pendant une date précise accompagné par les informations correspondantes tels que l'heure de départ, l'immatriculation du véhicule, l'heure de départ, le nom du chauffeur, et encore plus de détails.

2 Analyse préalable

2.1 *Analyse de l'existant*

2.1.1 Organisation actuelle :

D'après notre analyse au niveau des entreprises de transport (coopératives) à Madagascar, hormis les grandes coopératives actuelles tels que SOATRANS, COTISSE [a] et quelques coopératives, la majorité des petites coopératives malgache n'ont pas encore accès à la numérisation des données en raison de leurs incapacités à s'en procurer. Ces derniers emploient donc la méthode traditionnelle pour la gestion et l'enregistrement des données, c'est-à-dire à l'aide des supports non numériques.

On a remarqué également que certaines coopératives ont déjà accès à la numérisation, mais utilise des logiciels de bureau assez rudimentaire qui sont pratique mais représentent des fonctionnalités limitées et des risques de perte de données.

2.1.2 Inventaire des moyens matériels et logiciels :

Généralement, la majorité des petites coopératives à Madagascar utilisent encore des supports manuels non numériques, c'est-à-dire qu'elles enregistrent les données sur papier avec tous les accessoires qui aille avec, comme les stylos, les tampons, etc.

Mais, ces temps-ci, certaines coopératives ont commencé à utiliser des logiciels de bureau pour éviter les problèmes rencontrés par l'utilisation des papiers. Ce logiciel n'est autre que Microsoft Excel qui est un logiciel fonctionnant sur toute les versions de Windows. Microsoft Excel peut également fonctionner tant au système de 64 bit qu'au 32 bit.

Puis, on a vu aussi que certaines coopératives sont déjà à la pointe de l'évolution numérique et technologique. Elles utilisent des logiciels plus adaptés et avec plus de fonctionnalités obtenues par commande. Certaines ont même des applications disponibles sur Play Store (sur mobile) pour permettre aux clients de faire leurs réservations en ligne.

Tableau 1: Récapitulatif des moyens matériels et logiciels existants.

| Logiciel | Version | Utilité |
|-----------------------------------|------------|--|
| Microsoft Excel | 2013, 2016 | Gestion de données |
| Système d'exploitation Windows | 7, 10 | Gestion des ressources matérielles, fichiers, multitâche, etc. |

2.2 Critiques de l'existant et proposition de solutions

D'abord, l'utilisation des supports traditionnels comme les papiers représentent beaucoup de risques tant au niveau de la sécurité et confidentialité des données mais surtout au niveau du stockage des données. De plus, l'automatisation est impossible, du coup, cela nécessite beaucoup d'interventions physiques qui retardent considérablement le traitement des données. A pars cela, le prix des fournitures utilisés ne cessent de croître, alourdissant le budget des coopératives.

Puis, concernant l'utilisation du logiciel Microsoft Excel, certains problèmes persistent encore ; comme la difficulté à automatiser les processus. Excel peut être limité en termes d'automatisation des processus liés à la gestion de transport. Des tâches telles que la génération de rapport ou la mise à jour automatique des données. De plus, il faut faire très attention lors de la manipulation des données pour ne pas les modifier ou supprimer par erreur. Les fonctionnalités sont aussi très limitées et l'interface laisse à désirer (un tableau personnalisable). Le stockage des données est également problématique.

Ensuite, comme nous l'avons dit précédemment, les grandes coopératives (national) utilisent déjà des logiciels sophistiqués spécialisé pour la gestion des transports. Le problème, c'est que la maintenance de ce logiciel nécessite du temps et de l'argent. De plus, ces logiciels sont obtenus par commande auprès des boîtes informatiques avec un coût vraiment très élevé. Voilà pourquoi la majorité des coopératives n'est pas en mesure de s'en procurer. Et l'utilisation de ces logiciels nécessite évidemment des matérielles (ordinateur) de haute qualité avec des performances de configuration élevés.

C'est pourquoi, nous avons essayé de résoudre ces problèmes en créant un logiciel de gestion de transport fonctionnel, pratique, facile à manipuler grâce à la simplicité de l'interface graphique, peu couteux et qui ne requière pas des ordinateurs à haute performance car il s'exécute même sur un système de 32 bit.

Bien que l'analyse nous ait permis de comprendre en profondeur les besoins et les contraintes de l'existant, passons maintenant à la phase de conception, où nous allons élaborer l'environnement de développement, la base de données et les règles de gestion.

II. CONCEPTION

3 Conception avant-projet

3.1 *Système d'exploitation*

Un système d'exploitation est un ensemble de programmes qui dirige l'utilisation des ressources d'un ordinateur par des logiciels applicatifs, ces derniers sont des programmes qui délèguent la gestion de la mémoire et des périphériques aux programmes systèmes [1].

Le choix d'un OS est important pour mieux exploiter une application. Pour la réalisation de ce projet, nous avons pris en compte le système d'exploitation cible la plus utilisé pour satisfaire les besoins de la majorité des utilisateurs. D'après notre sondage, le choix adéquat n'est autre que le système d'exploitation « Microsoft Windows 10, 11 professionnel » qui est actuellement l'OS le plus utilisé. En outre, si l'application (Qt) doit être déployée principalement sur Windows, il peut être plus efficace et pratique de développer sur Windows plutôt que sur Mac ou sur Linux, pour effectuer des tests et des débogages.

Windows est un OS multitâche permettant d'exécuter de façon simultanée plusieurs programmes informatiques (processus) qui nous a permis de faciliter le développement de notre application. Ce système d'exploitation est sans doute le plus facile à manipuler grâce à son API simple à utiliser[b].

A part cela, Windows prend en charge une vaste disponibilité d'environnement de développement et il a également accès à une large communauté et à des ressources, étant donné que Windows est une plateforme populaire. Il existe une vaste communauté d'utilisateurs et de développeurs (de Qt) sur Windows. Il est donc plus facile de trouver des ressources, des tutoriels, des forums de support et des exemples de code spécifiques à Windows, ce qui peut accélérer le processus de développement. En plus, certains outils tiers ou bibliothèques peuvent être mieux pris en charge sur Windows plutôt qu'aux autres plateformes.

3.2 *Langage de programmation*

Parmi les langages de programmation proposés par le Framework, nous nous sommes penchés sur le langage C++ [c]. Ce dernier est un langage de programmation orienté objet et qui est un langage à haut niveau de performance. Il permet un accès direct à la mémoire et offre un contrôle fin sur les ressources système, ce qui nécessite une exécution rapide et utilisation efficace des ressources. C'est également un langage de programmation polyvalent qui peut être

utilisé pour développer différents types de logiciels (application de bureau, système embarqué, jeux vidéo, des bibliothèques, ...). Il offre une grande flexibilité et peut être adapté à plusieurs domaines et besoins.

A part cela, le langage de programmation C++ bénéficie d'une vaste sélection de bibliothèques disponibles ; des bibliothèques qui offrent des fonctionnalités avancées pour la gestion des interfaces utilisateurs, les structures de donnée, les opérations mathématiques, la gestion des fichiers, etc.

Le langage C++ est également un langage multiplateforme, ce qui signifie que le code source écrit en C++ peut être compiler et exécuter sur différentes plateformes, telles que Windows, Mac, Linux, Android, etc. Cela permet de développer des logiciels qui fonctionnent sur différentes plateformes sans avoir à réécrire complètement le code. C++ est dérivé du langage C, ce qui signifie qu'il est compatible avec le code existant en C.

3.3 Environnement de développement

Un environnement de développement est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels [2].

Pour mener à bien notre projet, on a utilisé Qt qui est un puissant Framework de développement qui constitue un ensemble complet d'outil pour la création d'applications multiplateformes, permettant de réduire le temps de développement et d'améliorer la productivité [3]. Qt fournit un ensemble complet de bibliothèques logicielles et d'outils pour le développement d'application. Et les bibliothèques Qt offrent une variété de fonctionnalités, y compris la gestion des événements, la gestion de l'interface utilisateur, l'accès aux bases de données, etc. Le Framework Qt est écrit en C++ et offre des bindings pour d'autres langages de programmation tels que Python.

Concernant l'environnement de développement utilisé, « Qt Creator » est un IDE dédié au développement d'applications avec le Framework Qt. Il offre des fonctionnalités avancées d'édition de code, de débogage, de gestion de projet et de conception d'interface utilisateur. Autrement dit, Qt Creator est un IDE spécialement conçu pour le développement d'application Qt qui offre des fonctionnalités tels que l'achèvement automatique du code, la mise en évidence de la syntaxe qui améliore la productivité. « Qt Designer » est un outil de conception graphique qui permet de créer facilement des UI en utilisant un système de glisser-déposer[d]. Il offre une interface conviviale pour la configuration des widgets et la mise en page des éléments visuels.

Ainsi, pour la réalisation de notre logiciel, nous avons opté pour l'utilisation de la version 5.13.2 de Qt Creator.

4 Conception

4.1 Dictionnaire des données

Un dictionnaire des données ou Référentiel (base de données) est un référentiel centralisé de l'information sur les données, leurs significations, relation avec d'autres données, leurs origine, utilisation et format [4].

Tableau 2: Dictionnaire de données de la coopérative

| COOPERATIVE | | | | |
|--------------------|--|-------------|-----------------|--------------------|
| Rubrique | Description | Type | Longueur | Observation |
| adresseCoop | Adresse de la coopérative | AN | 50 | |
| contactCoop | Contact de la coopérative | AN | 20 | |
| datePremierVoyage | La date de premier transport de la coopérative | D | 10 | jj/mm/aaaa |
| nomCoop | Nom de la coopérative | AN | 50 | Majuscule |
| refCoop | Référence de la coopérative | N | 2 | 1 |
| GESTION | | | | |
| dateVoyage | Date du voyage | D | 10 | aaaa-mm-jj |
| Destination | Lieu d'arrivé du voyage | AN | 50 | |
| Heure | Heure de départ du voyage | AN | 5 | |
| Id | Identification du trajet géré | N | 4 | Autoincrémenté |
| lieuDepart | Lieu de départ du voyage | AN | 50 | |
| numMAT | Numéro de matriculation du voiture | AN | 8 | |
| placeDispo | Place disponible dans le voyage | N | 2 | |
| TRAJET | | | | |

| | | | | |
|------------------|--|----|----|--------------------|
| decalageVoyage | Nombre de jours du décalage du trajet | N | 1 | |
| Destination | Lieu d'arrivé du trajet | AN | 50 | |
| heureMatin | Heure de départ matinal | AN | 5 | |
| heureSoir | Heure de départ en soirée | AN | 5 | |
| lieuDepart | Lieu de départ du trajet | AN | 50 | |
| refTrajet | Reference du trajet | AN | 6 | |
| VEHICULE | | | | |
| Chauffeur | Nom du chauffeur de la voiture | AN | 50 | Majuscule |
| contactChauffeur | Contact téléphonique du chauffeur | AN | 20 | |
| nbPlace | Nombre de place supporté par la voiture | N | 2 | |
| numMAT | Numéro d'immatriculation de la voiture | AN | 8 | |
| refTrajet | Reference du trajet effectué par la voiture | AN | 6 | |
| PASSAGER | | | | |
| Cin | Carte d'identité nationale du passager | AN | 12 | |
| contactFamille | Contact téléphonique de la famille du passager | AN | 20 | |
| contactPass | Contact téléphonique du passager | AN | 20 | |
| dateVoyage | Date du voyage du passager | D | 16 | jjj.mmm jj aaaa |
| Destination | Destination du passager | AN | 50 | |
| heureDepart | Heure de départ du passager | AN | 5 | |

| | | | | |
|----------------|---|----|----|----------------|
| Id | Identification du passager | N | 5 | Autoincrémenté |
| isPaye | Payement des frais du passager | N | 1 | 0 ;1 |
| lieuDepart | Lieu de départ du passager | AN | 50 | |
| membreFamille | Membre de la famille du passager | AN | 10 | |
| nbPlaceReserve | Nombre de place réserver par le passager | N | 2 | |
| nomPass | Nom du passager | AN | 50 | Majuscule |
| refPlace | Reference de la place du passager dans la voiture | AN | 10 | |
| refTrajet | Reference du trajet effectué par le passager | AN | 6 | |
| Voiture | Immatriculation de la voiture du passager en voyage | AN | 8 | |

AN : Alpha Numérique.

D : Date.

N : Numérique.

4.2 Règles de gestion

Le contexte de notre application permet à une coopérative de gérer ses trajets en ajoutant leurs trajets et les véhicules affectés à chaque trajet, afin de faciliter la réservation d'un ou de plusieurs clients pour mieux organiser les voyages. Notons quelques règles de gestion spécifiques existantes :

- **RG1** : la coopérative doit s'inscrire au préalable en fournissant des informations essentielles, telles que son nom, son adresse, ses coordonnées, et la date de son premier voyage.
- **RG2** : les utilisateurs doivent pouvoir ajouter des trajets disponibles en spécifiant sa référence, l'itinéraire, les horaires de départ (matin et soir) et le nombre de jour de décalage des voyages.
- **RG3** : les véhicules doivent être enregistrés avec des détails tels que leurs immatriculations, le nom du chauffeur avec sa coordonnée et la référence de son trajet.

- **RG4** : l'application requiert que chaque véhicule soit attribué à une date de voyage spécifique, une heure de départ déterminée, et un itinéraire précis pour la gestion des trajets.
- **RG5** : les changements ou les suppressions n'altèrent pas les trajets déjà gérés, mais les actualisent automatiquement.
- **RG6** : on effectue une vérification par rapport à la demande du client pour s'assurer que l'itinéraire et la date sont valide et qu'il reste des places disponibles dans les véhicules.
- **RG7** : les passagers peuvent réserver des voyages disponibles en indiquant son nom, son contact, sa carte d'identité national, son membre de famille à contacter en cas d'urgence et le nombre de places dont ils ont besoins.
- **RG8** : un véhicule est assigné à un et un seul trajet et un seul chauffeur titulaire.
- **RG9** : un passager peut réserver plusieurs places.
- **RG10** : le décalage entre deux voyages successive dépend de la distance du trajet.

4.3 Modèle Conceptuel des Données (MCD)

Le modèle conceptuel des données est une représentation statique du système d'information de l'entreprise qui met en évidence sa sémantique [5]. Nous allons explorer une illustration de la structure essentielle des données d'une coopérative, mettant en évidence les entités, les attributs et les associations qui les relient, offrant ainsi une vue holistique de l'organisation des données dans un contexte particulier.

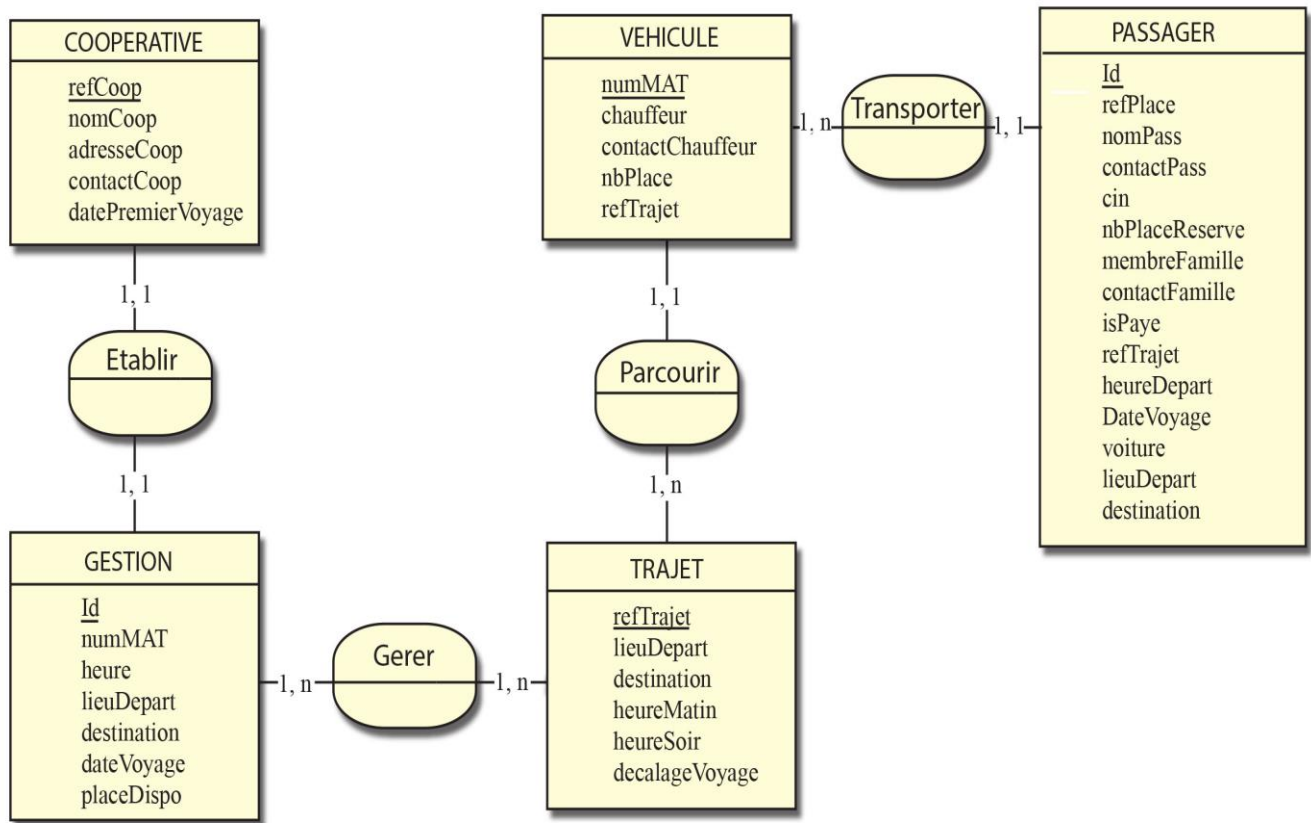


Figure 1: Modèle Conceptuel des données

Cette figure nous montre les associations entre les entités et leurs cardinalités respectives. D'après la figure, voici un briefing de la relation et de l'interdépendance entre ces entités : « La coopérative établit une gestion pour gérer les trajets parcourus par les véhicules permettant le déplacement des passagers ».

4.4 Modèle Logique des Données (MLD)

Le modèle logique des données est le modèle qui va introduire des notions techniques nécessaires à la résolution des problèmes abordés [6]. Nous allons présenter en détail la structure des données d'une coopérative avec une emphase sur les aspects techniques, tels que les tables, les clés primaires et les contraintes, fournissant ainsi une vue plus contrainte et opérationnelle de la façon dont les données seront stockées et gérées dans le système.

- COOPERATIVE (refCoop, nomCoop, contactCoop, datePremierVoyage)
- GESTION (Id, numMAT, heure, lieuDepart, destination, dateVoyage, placeDispo)
- TRAJET (refTrajet, lieuDepart, destination, heureMatin, heureSoir, decalageVoyage)
- VEHICULE (numMAT, chauffeur, contactChauffeur, nbPlace, refTrajet)

- PASSAGER (Id, refPlace, nomPass, contactPass, cin, nbPlaceReserve, membreFamille, contactFamille, isPaye, refTrajet, heureDepart, dateVoyage, numMAT, lieuDepart, destination).

Après avoir élaboré notre conception détaillée, passant maintenant à la phase de réalisation où nous mettrons en œuvre la configuration des outils et le développement de l'application.

III. REALISATION

5 Installation et configuration des outils

Il existe deux moyen d'installer Qt Creator, qui sont tous deux suggérés dans le site officiel de Qt (une installation online et une installation offline). En se rendant sur le site officiel de Qt www.qt.io [e], il faut entrer dans « Développeur Tools », puis cliquer sur « Download Qt » puis « Download open source ». Il faut ensuite choisir le mode d'installation, dans notre cas nous avons télécharger l'installateur offline. Comme indiqué dans la documentation officiel, Qt est un Framework multiplateforme donc il faut télécharger le pack d'installation de Qt correspondant au système d'exploitation utilisé, ensuite choisir la version de Qt qui est dans notre cas, la version 5.13.2. Une fois téléchargé, on peut procéder à l'installation.

Les quatre (4) figures suivantes représentent les étapes d'installation de Qt Creator version 5.13.2.

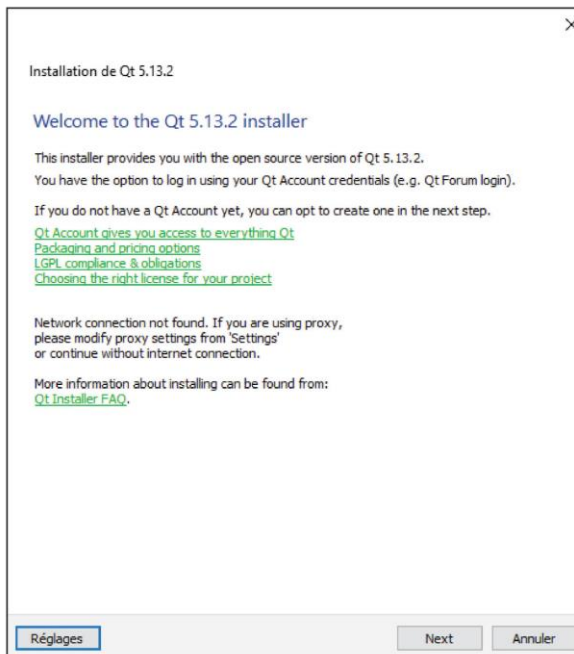


Figure 2: Premier fenêtre d'installation

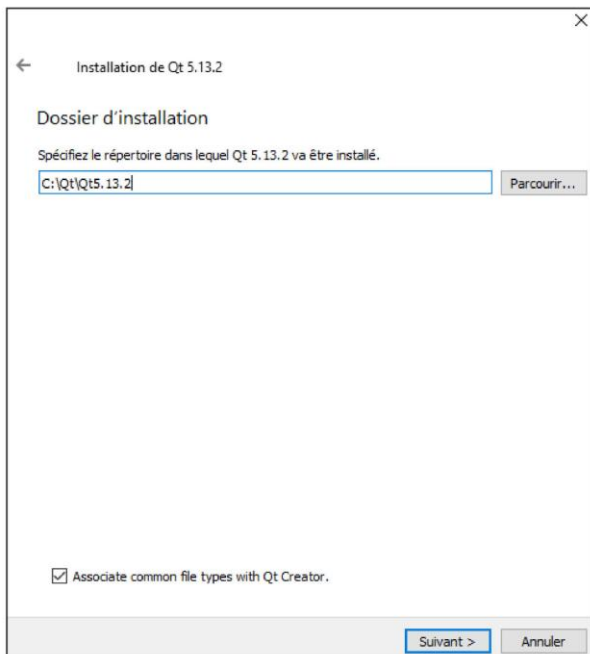


Figure 3: Indiquer le chemin d'installation

La figure 2 représente la fenêtre d'installation hors ligne après avoir exécuté l'installateur pour confirmer la poursuite du processus en cliquant sur « Next ». Il faut ensuite choisir l'emplacement désiré pour dans lequel Qt va être installé (selon la figure 3).

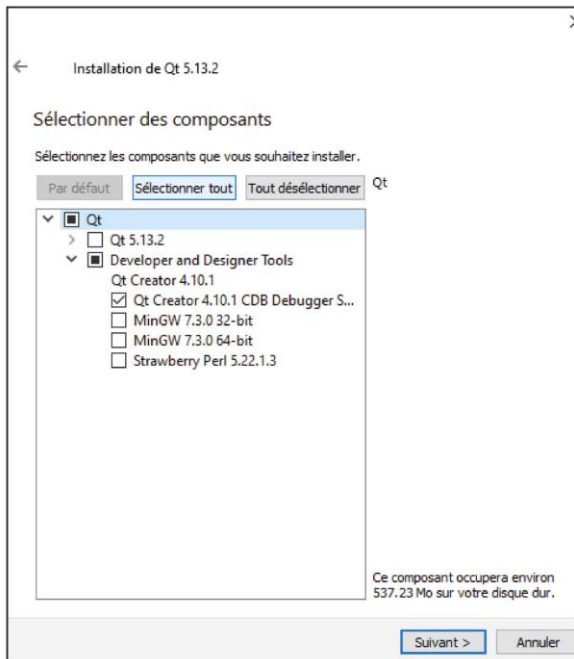


Figure 4: Sélection des composants nécessaires

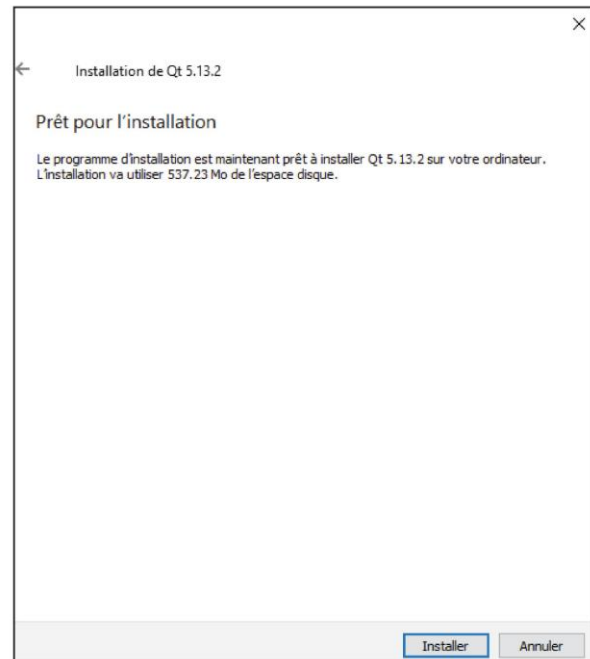


Figure 5: Installation puis terminer

La fenêtre propose une multitude de choix de composants qui peuvent être installés (figure 4). Une fois les composants sélectionnés, l'installation pourra commencer en cliquant sur « Suivant » puis « Installer ». (Figure 5)

6 Développement de l'application

6.1 Description de la base de données

Concernant le SGBD utilisé, nous avons opté pour l'utilisation de SQLite qui est une base de données relationnelle légère et autonome, largement nécessaire lors du développement de notre application. Elle offre une solution efficace pour le stockage et la gestion des données de manière locale. Avec SQLite, on a pu créer des tables pour représenter les entités qui jouent des rôles dans notre application et établir les relations entre ces entités. Cette base de données prend en charge un sous-ensemble du langage SQL, ce qui permet d'effectuer des opérations courantes telles que l'insertion, la mise à jour, la suppression et la recherche de données. De plus, SQLite peut être intégré à de nombreux langages de programmation notamment du langage C++ dans notre cas, offrant ainsi une intégration transparente. Il est aussi très apprécié pour sa simplicité, sa rapidité et sa fiabilité, ce qui en fait un choix populaire pour le stockage local de données.

On a déjà vu précédemment sous forme de tables (objets) les entités qui assurent le flux et le fonctionnement du système de déplacement (transport). La base de données de notre logiciel est conçue pour stocker et gérer efficacement les informations liées aux opérations de transport.

Elle permet de centraliser les données relatives aux véhicules, aux chauffeurs, aux itinéraires (trajets), aux horaires, aux paiements et à d'autres aspects clés du processus de gestion de transport. La structure de la base de données comprend donc des tables interconnectées. Par exemple, une table « VEHICULE » contient des informations telles que le numéro d'immatriculation qui est l'identifiant du véhicule, sa capacité, le nom de son conducteur, la référence du trajet qui lui est assignée, etc. Une table « COOPERATIVE » qui a pour attributs son identifiant (unique), son nom, son adresse, son numéro de contact, etc. Une table « TRAJET » qui pourrait enregistrer les trajets planifiés, avec des informations sur le lieu et l'heure de départ, etc. Pour plus de détails, la représentation du MCD et MLD précédemment en dira davantage (page 11).

La base de donnée permet également de gérer les informations relatives aux passagers (clients) via l'entité ou table « PASSAGER », qui contient leurs coordonnées (adresse), leurs besoins, les historiques de leurs expéditions ainsi que celui des expéditions en cours ou à venir, et d'autres informations correspondantes.

Notre logiciel de gestion de transport utilise nécessairement des requêtes SQL pour interagir avec la base de données, ce qui permet d'effectuer des opérations telles que l'ajout, la modification, la suppression, la mise à jour et la recherche de données. Des fonctionnalités avancées ont aussi été intégrées, telles que la génération de rapport dans l'historiques, l'analyse des nombres de voyages effectués par unité de véhicules, la planification optimisée des itinéraires et l'automatisation de ce processus.

6.2 Elaboration des algorithmes

D'abord, la première étape est la création des classes qui ne sont autres que les entités. C'est-à-dire la détermination des attributs ou propriétés de chaque table, puis la détermination de leurs méthodes respectives [f][g].

```

#ifndef COOPERATIVE_H
#define COOPERATIVE_H

#include "identificateur.h"

class Cooperative : public Identificateur
{
public:
    Cooperative();
    Cooperative(QString ref_coop, QString nom, QString contact, QString contact1,
                QString contact2, QString address, QString datePremierVoyage);

    bool majCoop(QString nom, QString contact, QString contact1, QString contact2,
                QString address, QString datePremierVoyage);
    bool reinitialiser();

private:
    QString _ref_coop;
    QString _address;
    QString _contact1;
    QString _contact2;
    QString _datePremierVoyage;
};

#endif // COOPERATIVE_H

```

Figure 6: Entête de la classe coopérative

La classe Coopérative représente une coopérative avec des méthodes pour mettre à jour les informations de la coopérative et réinitialiser sa base de données. (Figure 6)

Méthodes :

- majCoop () : Cette méthode permet de mettre à jour les informations de la coopérative, ce qui peut inclure la modification de données telle que le nom, l'adresse, les contacts, etc.
- reinitialiser() : Cette méthode permet de réinitialiser la base de données de la coopérative, ce qui signifie qu'elle remettra à zéro toutes les données de la coopérative, revenant ainsi à un état initial.

```

#ifndef GESTIONTRAJET_H
#define GESTIONTRAJET_H

class gestionTrajet
{
    public:
        gestionTrajet();
        bool gerer();
};

#endif // GESTIONTRAJET_H

```

Figure 7: Entête de la classe gestionTrajet

La classe gestionTrajet représente un système de gestion des trajets pour les transports au sein de la coopérative. (Figure 7)

Méthode :

- gerer() : Cette méthode permet de coordonner les trajets en fonctions de la date du premier trajet effectué, les informations sur les trajets et les véhicules disponibles.

```

#ifndef TRAJETS_H
#define TRAJETS_H

#include <QString>

class Trajets
{
    public:
        Trajets();
        Trajets(QString ref_trajet, QString lieu_depart, QString destination, QString heureMatin,
                QString heureSoir, int decalage);

        bool ajout_trajet(QString ref_trajet, QString lieu_depart, QString destination,
                QString heureMatin, QString heureSoir, int decalage);
        bool maj_trajet(QString ref_trajet, QString lieu_depart, QString destination,
                QString heureMatin, QString heureSoir, int decalage);
        bool supprimer_trajet(QString ref_trajet);

    protected:
        QString _ref_trajet;
        QString _lieu_depart;
        QString _destination;
        QString _heureMatin;
        QString _heureSoir;
        int _decalage;
};

#endif // TRAJETS_H

```

Figure 8: Entête de la classe trajet

La classe Trajets est conçue pour la gestion des trajets dans le cadre de la coopérative de transport. Elle permet de modéliser, ajouter, mettre à jour et supprimer des trajets enregistrés, y compris leur point de départ, leur destination, l'heure de départ (matinale ou en soirée), le nombre de jours de décalage, et d'autres informations associées. (Figure 8)

Méthodes :

- ajout_trajet () : Cette méthode permet d'ajouter de nouveaux trajets au système, en spécifiant sa référence, le point de départ, la destination, l'heure de départ le matin, l'heure de départ en soirée, le nombre de jours de décalage.
- maj_trajet () : Cette méthode permet de mettre à jour les informations d'un trajet existant en spécifiant sa référence et les nouvelles données, telles que le nouveau point de départ, la nouvelle destination, les nouvelles heures de départ, etc.
- supprimer_trajet () : Cette méthode permet de supprimer un trajet du système en spécifiant sa référence.

```
#ifndef VEHICULE_H
#define VEHICULE_H

#include "identificateur.h"
#include "trajets.h"

class Vehicule : public Identificateur, Trajets
{
public:
    Vehicule();
    Vehicule(QString num_Matriculation, QString nom, QString contact, QString ref_trajet,
              QString lieu_depart, QString destination, QString heureMatin, QString heureSoir,
              int nb_Place, int decalage);

    bool ajouter_vehicule(QString num_Matriculation, QString nom, QString contact, int nb_Place,
                          QString ref_trajet);
    bool modifier_vehicule(QString num_Matriculation, QString nom, QString contact,
                           int nb_Place, QString ref_trajet);
    bool supprimer_vehicule(QString num_Matriculation);

private:
    QString _num_Matriculation;
    int _nb_Place;
};

#endif // VEHICULE_H
```

Figure 9: Entête de la classe véhicule

La classe Vehicule est destinée à la gestion des véhicules dans le cadre de la coopérative de transport. Elle permet de modéliser, d'ajouter, de mettre à jour et de supprimer des véhicules enregistrés, en spécifiant des informations telles que le numéro d'immatriculation, le chauffeur attitré, la capacité de passagers, le contact du chauffeur, la référence au trajet associé, etc.

(Figure 9)

Méthodes :

- ajouter_vehicule () : Cette méthode permet d'ajouter de nouveaux véhicules au système en spécifiant le numéro d'immatriculation, le chauffeur, le contact du chauffeur, la capacité de passagers, la référence au trajet auquel le véhicule est associé, etc.
- modifier_vehicule () : Cette méthode permet de mettre à jour les informations d'un véhicule existant en spécifiant son numéro d'immatriculation et en fournissant de nouvelles données.
- supprimer_vehicule () : Cette méthode permet de supprimer un véhicule du système en spécifiant son numéro d'immatriculation.

```
#ifndef RESERVATION_H
#define RESERVATION_H
#include "vehicule.h"
class Reservation : public Vehicule {
public:
    Reservation();
    Reservation(QString ref_place, QString nom, QString contact, QString cin,
        int nb_place, QString membre_famille, QString contact_famille,
        QString num_Matriculation, QString ref_trajet, QString date_voyage,
        QString lieu_depart, QString destination, QString heureMatin,
        QString heureSoir, bool isPaye, int decalage);
    bool reserver(QString ref_place, QString nom, QString contact, QString cin,
        int nb_place_reserve, QString membre_famille, QString contact_famille,
        QString num_Matriculation, QString ref_trajet, QString date_voyage,
        QString heure_depart, bool isPaye, QString lieu_depart, QString destination);
    bool majReservation(int id, QString ref_place, QString nom, QString contact, QString cin,
        int nb_place_reserve, QString membre_famille, QString contact_famille,
        bool isPaye);
    bool supprReservation(int id);
private:
    QString _ref_place;
    QString _cin;
    QString _membre_famille;
    QString _contact_famille;
    QString _date_voyage;
    bool _isPaye; };
#endif // RESERVATION_H
```

Figure 10: Entête de la classe réservation

La classe Reservation est conçue pour gerer les réservations de passager dans le cadre de la coopérative de transport. Elle permet de modéliser, d'ajouter, de mettre à jour et de supprimer des réservations enregistrées. (*Figure 10*)

Méthodes :

- `reserver()` : Cette méthode permet de créer une nouvelle réservation en spécifiant les détails du passager , tels que son nom, son numéro de téléphone, son itinéraire, etc.
- `majReservation()` : Cette méthode permet de mettre à jour une réservation existante en spécifiant son identifiants (Id) et les nouvelles données de la réservation.
- `supprReservation()` : Cette méthode permet de supprimer une réservation du système en spécifiant son identifiant (Id).

Une fois, toutes les classes créées, on passe à l'élaboration de l'algorithmes de gestions proprement dite. Voici les principales étapes de la réalisation de notre algorithme :

- Recueillir la liste des véhicules dans la base de données
- Regrouper les véhicules ayant le même référence trajet (ex : Tana-Fianarantsoa et Fianarantsoa-Tana ont le même référence trajet)
- Récupérer la date du premier voyage
- Déterminer le décalage horaire/journalier d'un voyage par rapport au prochain voyage
- Déterminer l'heure de départ en matinée et en soirée
- Affecter l'heure de départ aux véhicules
- Affecter les lieux de départ et destination dans chaque véhicule
- Automatiser le changement de départ et de destination en tenant compte du décalage.

6.3 Quelques captures d'écran

Les captures d'écrans ci-dessous offrent un aperçu visuel du logiciel, vous permettant de visualiser son interface utilisateur, ces fonctionnalités et son apparence générale.

D'abord, lors du lancement du logiciel, on accède directement à la page d'accueil de notre logiciel de gestion de transport, qui affiche un tableau initialement vide. Une fois que les informations concernant la coopérative et son activité sont insérées dans la base de données, la page d'accueil permettra de visualiser le programme (emploi du temps) des opérations de déplacement de chaque véhicule (*Figure 11*).

e-Coop

Paramètre

Accueil

Coté Coopérative

Coté Passager

Historique

A propos

GUEST_TRANS

Nom Coopérative:

GUEST_TRANS

Adresse Coopérative:

Fianarantsoa

Contact Coopérative:

034 22 222 22

Première voyage:

01/10/2023

Mettre à jour

Annuler

Réinitialiser

Les Trajets

| Référence trajet | Départ/Destination | Destination/Départ |
|------------------|--------------------|--------------------|
| TF | Fianarantsoa | Antananarivo |

Départ/Destination:

Sélectionner

Destination/Départ:

Sélectionner

Heure de départ matin:

00:00

Heure de départ soir:

00:00

Référence Trajet:

Décalage des voyages (jours):

0

Mettre à jour

Ajouter

Supprimer

Annuler

Les Voitures

| N° Matriculation | Chauffeur | Place supportés | Référence trajet |
|------------------|-----------|-----------------|------------------|
| 1324 AC | JOHN | 15 | TF |
| 2637 AB | DOE | 15 | TF |
| 6725 GH | BEN | 20 | TF |

N° de matriculation:

Nombre des places supportés:

0

Chauffeur:

Contact:

03* * * * *

Référence Trajet:

Sélectionner

Mettre à jour

Ajouter

Supprimer

Annuler

Quitter

Gérer les Trajets

Figure 12: Interface du côté coopératif

Paramètre

GUEST_TRANS

Première voyage:

01/10/2023

Mettre à jour

Annuler

Réinitialiser

Les Trajets

| Référence trajet | Départ/Destination | Destination/Départ |
|------------------|--------------------|--------------------|
| TF | Fianarantsoa | Antananarivo |

Départ/Destination:

Sélectionner

Destination/Départ:

Sélectionner

Heure de départ matin:

00:00

Heure de départ soir:

00:00

Référence Trajet:

Décalage des voyages (jours):

0

Mettre à jour

Ajouter

Supprimer

Annuler

Les Voitures

| N° Matriculation | Chauffeur | Place supportés | Référence trajet |
|------------------|-----------|-----------------|------------------|
| 1324 AC | JOHN | 15 | TF |
| 2637 AB | DOE | 15 | TF |
| 6725 GH | BEN | 20 | TF |

N° de matriculation:

Nombre des places supportés:

0

Chauffeur:

Contact:

03* * * * *

Référence Trajet:

Sélectionner

Mettre à jour

Ajouter

Supprimer

Annuler

Gérer les Trajets

Figure 13: Interface du côté coopératif

23

Les figures 12 et 13 montrent l'interface de paramètre pour la coopérative, permettant de saisir des informations, affichant les détails des trajets et des véhicules, ainsi que les options d'ajout et de modification.

Pour effectuer une réservation, on doit accéder dans l'onglet « Côté passager » comme montré dans la figure 14 ci-dessous

e-Coop Trajets et Réservation

Accueil | Côté Coopérative | **Côté Passager** | Historique | A propos

Vérification véhicules et trajets

| Voiture | Lieu de départ | Lieu d'arrivée | Place disponible |
|---------|----------------|----------------|------------------|
| 2637 AB | Antananarivo | Fianarantsoa | 11 |

Référence Trajet: TF
 Date: 09/10/2023
 Heure: 08:00
 [Vérifier] [Annuler]

Réservation

| Id | Référence Place | Nom | Nombre de place | Frais | Contact |
|----|-----------------|----------|-----------------|----------|------------|
| 1 | 12, 13 | JEREMIAH | 2 | Non payé | 0347866889 |
| 2 | 1 | TAHIRY | 1 | Non payé | 0345595862 |
| 3 | 9 | NY AINA | 1 | Payé | 0346171642 |

Nom: [] Contact: 03*****
 CIN: [] Référence place: []
 Un membre de la famille: Sélectionner [] Contact: 03*****
 Référence Trajet: TF Voiture: 2637 AB
 Date du voyage: 09/10/2023 Heure de départ: 08:00
 Nombre de place à réserver: 0 Frais: Non Payé
 [Mettre à jour] [Réserver]
 [Supprimer] [Annuler]

Figure 14: Interface du côté passager

Concernant la fenêtre utilisée pour la réception des commandes (réservations) des clients (passagers), l'interface a été conçue pour faciliter le travail des guichets. Pour effectuer une réservation, il faut d'abord vérifier si un ou plusieurs véhicules seront disponibles à une date donnée pour un trajet spécifique. Ensuite, si le résultat est positif, il faut vérifier si le nombre de passagers pour chaque véhicule n'est pas encore complet avant d'ajouter un passager.

Afin d'apporter un meilleur suivi et contrôle des voyages effectués par chaque véhicule, le logiciel permet d'enregistrer les historiques des voyages comme nous montre la figure 15 suivante.

Accueil

Coté Coopérative

Coté Passager

Historique

A propos

Rechercher

Historique

Filtre

Voiture: Tous

Date de Voyage: 14/10/2023

Heure de départ: Tous

Par défaut

| Voiture | Référence trajet | Heure de départ | Lieu de départ | Lieu d'arrivé | Date | Places occupés |
|---------|------------------|-----------------|----------------|---------------|-------------------|----------------|
| 6725 GH | TF | 16:30 | Antananarivo | Fianarantsoa | ven. oct. 13 2023 | 0 |
| 2637 AB | TF | 08:00 | Antananarivo | Fianarantsoa | ven. oct. 13 2023 | 0 |
| 1324 AC | TF | 16:30 | Fianarantsoa | Antananarivo | ven. oct. 13 2023 | 0 |
| 6725 GH | TF | 16:30 | Fianarantsoa | Antananarivo | mer. oct. 11 2023 | 0 |
| 2637 AB | TF | 08:00 | Fianarantsoa | Antananarivo | mer. oct. 11 2023 | 0 |
| 1324 AC | TF | 16:30 | Antananarivo | Fianarantsoa | mer. oct. 11 2023 | 0 |
| 6725 GH | TF | 16:30 | Antananarivo | Fianarantsoa | lun. oct. 9 2023 | 0 |
| 2637 AB | TF | 08:00 | Antananarivo | Fianarantsoa | lun. oct. 9 2023 | 0 |
| 1324 AC | TF | 16:30 | Fianarantsoa | Antananarivo | lun. oct. 9 2023 | 0 |
| 6725 GH | TF | 16:30 | Fianarantsoa | Antananarivo | sam. oct. 7 2023 | 0 |
| 2637 AB | TF | 08:00 | Fianarantsoa | Antananarivo | sam. oct. 7 2023 | 0 |
| 1324 AC | TF | 16:30 | Antananarivo | Fianarantsoa | sam. oct. 7 2023 | 0 |
| 6725 GH | TF | 16:30 | Antananarivo | Fianarantsoa | jeu. oct. 5 2023 | 0 |
| 2637 AB | TF | 08:00 | Antananarivo | Fianarantsoa | jeu. oct. 5 2023 | 0 |
| 1324 AC | TF | 16:30 | Fianarantsoa | Antananarivo | jeu. oct. 5 2023 | 0 |

Nombre de voyages effectués par les voitures :

Voiture (1324 AC): 7 voyages effectués (Tous les jours)

Voiture (2637 AB): 7 voyages effectués (Tous les jours)

Voiture (6725 GH): 7 voyages effectués (Tous les jours)

Nombre de voyages effectués par les voitures :

Voiture (1324 AC): 7 voyages effectués (TF)

Voiture (2637 AB): 7 voyages effectués (TF)

Voiture (6725 GH): 7 voyages effectués (TF)

Figure 15: Historique des voyages

Il existe également une section pour les historiques de chaque voyage effectué, permettant de contrôler et de suivre chaque véhicule. Cela pourrait être bénéfique pour déterminer si un véhicule nécessite un entretien, tel que le remplacement des pneus ou une vidange, par exemple.

Enfin, pour obtenir davantage plus de détails concernant le logiciel ou pour signaler un problème, nous avons laissé nos coordonnées dans la section "À propos".



Figure 16: A propos du logiciel

Cette section a été conçu pour rassembler les informations essentielles, notamment la définition de l'application, les détails des développeurs, la version de l'application, la date de création et le système d'exploitation compatible au fonctionnement de ce logiciel.

CONCLUSION

En conclusion, le projet de conception du logiciel de gestion des trajets d'une coopérative a été un succès. Le logiciel répond aux besoins spécifiques des entreprises en matière de gestion efficace des opérations de transport. Il offre des fonctionnalités plus ou moins avancées telles que la planification des itinéraires, la gestion des véhicules, la gestion des horaires, la gestion des réservations, ainsi que génération de rapport détaillés.

Nous avons développé notre application à l'aide du Framework Qt, en utilisant l'IDE Qt Creator et Qt designer en raison de leurs interfaces d'utilisation simple et de leurs fonctionnalités. Le code source de notre application est écrit en C++ et la base de donnée utilisée est SQLite.

Notre équipe a travaillé en étroite collaboration pour garantir le fonctionnement de l'application, ainsi que pour garantir une interface conviviale et intuitive. Grâce à ce logiciel, les entreprises seront en mesure d'optimiser ses processus de transport, d'améliorer la satisfaction clientèle, mais surtout de réduire les coûts ou dépenses.

Ce projet a été une étape importante dans la transformation numérique et ouvre la voie à de nouvelles opportunités de croissance et d'efficacité. Ce projet nous a permis de nous initier au monde du travail mais aussi de nous apprendre à coopérer et travailler en équipe. Cela nous a donc permis d'acquérir des expériences sur la programmation spécifiquement de la programmation orientée objets qui nous sera peut-être utile dans notre future carrière en tant que programmeur.

Cette application est opérationnelle et prête à être utilisée. Néanmoins, elle possède encore quelque problématique comme la limite des informations qu'on peut gérer sur l'application. On envisage une étude plus approfondie pour effectuer des améliorations en utilisant d'autres SGBD comme MySQL, MongoDB, MariaDB. A part cela, nous pouvons encore optimiser cette application en ajoutant plus de fonctionnalité tel que l'insertion manuelle dans le tableau de gestion généré automatiquement qui remettra à jour et modifiera la suite du programme des opérations à venir. La génération et la modélisation des tickets ne sont pas encore intégrées dans notre logiciel, encore une fonctionnalité de plus à ajouter dans notre application, ainsi qu'un bouton pour imprimer les tickets.

BIBLIOGRAPHIE

- [1] : Zeste de savoir, Les systèmes d'exploitation, 12 Août 2019, 43p.
- [2] : Donald G. Firemith et B. Hench – Seller, The OPEN Process Framework: An Introduction Pearsen Education – 2002.
- [3] : Lee Zhi Eng, Ruy Rishpater, Application Developpment with Qt Creator, Build Cross-platform application and GUIs using Qt 5 and C++, 3rd ed., 2020, 426p.
- [4] : ACM, Dictionnaire informatique IBM, 10ème Edition, 1993, 758p.
- [5] : Mohamed NEMICHE, Analyse et conception du système d'information (Merise), 2012-2013, 60p.
- [6] : Stride world, Modèle Logique Des donnée, le 11 Octobre 2009, 15p.

WEBOGRAPHIE

- [a] : Cotisse Transport, <https://www.cotisse-transport.com>, consulté le mois d'Octobre 2023
- [b] : Wikipedia, <https://www.wikipedia.org>, consulté le mois d'Octobre 2023.
- [c] : FormationVideo, formation-video.org, consulté le mois de Septembre 2023.
- [d] : Cabro Group, <http://www.youtube.com/@cabrogroup>, consulté le mois de Septembre 2023.
- [e] : Qt Group, www.qt.io, consulté le mois de Septembre 2023.
- [f] : ChatGPT, chat.openai.com, consulté le mois de Septembre 2023.
- [g] : Bard, bard.google.com, consulté le mois d'Octobre 2023.

RESUMÉ

Le cadre de l'enseignement de l'ENI nous a assigné comme projet, la réalisation d'un logiciel de « gestion de transport », dans le but d'assimiler l'aboutissement de nos efforts, de notre apprentissage et de notre passion pour l'informatique.

L'objet de cet ouvrage porte sur la conception d'un logiciel de gestion de transport plus précisément concernant la gestion de trajet ; mais nous y avons ajouté une fonctionnalité de gestion de réservation, c'est-à-dire la gestion des clients. Nous avons développé notre application à l'aide du Framework Qt, en utilisant l'IDE Qt Creator et Qt designer en raison de leurs interfaces d'utilisation simple et de leurs fonctionnalités. Le code source de notre application est écrit en C++ et la base de donnée utilisée est SQLite.

L'objectif principale était de développer une application efficace pour organiser les opérations effectuées au niveau d'une coopérative. Cette application est opérationnelle et prête à être utilisée. Cependant, certaines optimisations peuvent être ajoutées pour innover notre logiciel.

Mots-clés: logiciel, gestion de transport, Framework Qt, gestion de trajet, C++.

ABSTRACT

As part of our studies at ENI, we were given the project of developing a transportation management software. This was an opportunity for us to showcase our hard work, learning, and passion for computer science.

This book describes the design of a transportation management software, with a focus on journey management. We also added a reservation management functionality, which allows users to manage their customers. We developed our application using the Qt framework, which provides a simple and powerful user interface. The source code of our application is written in C++, and the database used is SQLite.

The primary goal was to develop an efficient system for organizing operations at the cooperative level. This application is operational and ready to use. However, some potential optimizations could be made to improve the software.

Keywords: Software, transportation management, Framework Qt, journey management, C++.