**CURRICULUM VITAE**



|  |  |
| --- | --- |
| ETAT CIVIL | RAKOTOZAFY Nantenaina Jérémiah   * Face bureau Fokontany   Tambohobe-Fianarantsoa  Madagascar  +261 34 78 668 89  [nantenainajeremiah@gmail.com](mailto:nantenainajeremiah@gmail.com)  18 ans, célibataire |

**FORMATION ET DIPLOME**

|  |  |
| --- | --- |
| 2022 – 2023  2021 – 2022  2021 | * Ecole Nationale d’Informatique Fianarantsoa   Première année en formation de Licence professionnelle (Génie logiciel et Base de données).   * Lycée Ambatolampy   Baccalauréat scientifique (série S)   * Alliance Française Andavamamba   Diplôme d’étude de Langue Française (DELF B2) |

**COMPETENCES EN INFORMATIQUE**

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

**COMPETENCES LANGUISTIQUE**

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

**DIVERS**

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

**CURRICULUM VITAE**



|  |  |
| --- | --- |
| ETAT CIVIL | RAMANDIMBIHANISOA Ny Aina Haritiana   * Lot 071 AD/3602   Ankofafa-Fianarantsoa  Madagascar  +261 34 61 716 42  [ainaramandimby@gmail.com](mailto:ainaramandimby@gmail.com)  21 ans, célibataire |

**FORMATION ET DIPLOME**

|  |  |
| --- | --- |
| 2022 – 2023  2020 – 2021 | * Ecole Nationale d’Informatique Fianarantsoa   Première année en formation de Licence professionnelle (Génie logiciel et Base de données).   * IEF Aceem   Baccalauréat scientifique (série C) |

**COMPETENCES EN INFORMATIQUE**

|  |  |
| --- | --- |
| Système d’exploitation  Langage de programmation  Développement Web  Framework  SGBD  Logiciels | Windows, Linux  C, C++, Bash shell  HTML, CSS, JavaScript  Qt, JQuery  SQLite, MySQL  Adobe Photoshop, Illustrator, After Effects, Premiere Pro |

**COMPETENCES LANGUISTIQUE**

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

**DIVERS**

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

**CURRICULUM VITAE**



|  |  |
| --- | --- |
| ETAT CIVIL | RAZANAMARA Haingomanana Tahiry   * Lot 116 A2 3710   Ampitakely-Fianarantsoa  Madagascar  +261 34 55 958 62  [tahiryrazanamara01@gmail.com](mailto:tahiryrazanamara01@gmail.com)  21 ans, célibataire |

**FORMATION ET DIPLOME**

|  |  |
| --- | --- |
| 2022 – 2023  2019 – 2020 | * Ecole Nationale d’Informatique Fianarantsoa   Première année en formation de Licence professionnelle (Génie logiciel et Base de données).   * Lycée Pôle Fort-Dauphin   Baccalauréat scientifique (série D) |

**COMPETENCES EN INFORMATIQUE**

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

**COMPETENCES LANGUISTIQUE**

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

**DIVERS**

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

**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 :

**Professeur MAHATODY Thomas**

Directeur de l’Ecole National d’Informatique de Fianarantsoa

**Docteur RABETAFIKA Louis Haja**

Chef de mention informatique à l’ENI

Enseignant à l’ENI

**Docteur WILLIAM Germain Dimbisoa**

Encadreur et conseillé pour l’innovation du projet

Enseignant à l’ENI

**Monsieur RALAIVAO Jean Christian**

Responsable du parcours GB à l’ENI et de l’initiation au SGBD

Assistant d’Enseignement Supérieur et de Recherche

**Monsieur Gilante GESAZAFY**

Responsable de l’initiation à la programmation sur le Framework Qt

Assistant d’Enseignement Supérieur et de Recherche

Nous profitons de cette occasion pour remercier :

**Les enseignants de l’Ecole Nationale d’Informatique,**

**Les personnels et l’équipe pédagogique,**

**Tous les membres de jury.**

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

Avec toutes nos gratitudes.

**LISTE DES FIGURES**

|  |  |  |
| --- | --- | --- |
| Figure 01 | :Modèle Conceptuel des données.................................................... | 11 |
| Figure 02 | : Premier fenêtre d’installation......................................................... | 12 |
| Figure 03 | : Indiquer le chemin d’installation.................................................... | 12 |
| Figure 04 | : Sélection des composants nécessaires............................................ | 13 |
| Figure 05 | : Installation puis terminer................................................................ | 13 |
| Figure 06 | : Entête de la classe coopérative....................................................... | 14 |
| Figure 07 | : Entête de la classe gestionTrajet..................................................... | 15 |
| Figure 08 | : Entête de la classe trajet.................................................................. | 15 |
| Figure 09 | : Entête de la classe véhicule............................................................ | 16 |
| Figure 10 | : Entête de la classe réservation........................................................ | 17 |
| Figure 11 | : Page d’accueil................................................................................. | 18 |
| Figure 12 | : Interface du côté coopératif............................................................ | 18 |
| Figure 13 | : Interface du côté coopératif............................................................ | 19 |
| Figure 14 | : Interface du côté passager............................................................... | 19 |
| Figure 15 | : Historique des voyages qui étaient déjà fait................................... | 20 |
| Figure 16 | : A propos du logiciel........................................................................ | 20 |

**LISTE DES TABLEAUX**

|  |  |  |
| --- | --- | --- |
| Tableau 01 | : Dictionnaire de donnée de la coopérative...................................... | 7 |
| Tableau 02 | : Dictionnaire de donnée de la gestion du trajet............................... | 7 |
| Tableau 03 | : Dictionnaire de donnée du trajet..................................................... | 8 |
| Tableau 04 | : Dictionnaire de donnée du véhicule............................................... | 8 |
| Tableau 05 | : Dictionnaire de donnée du passager............................................... | 9 |

**LISTE DES ABRÉVIATIONS ET NOMENCLATURE**

**ENI :** Ecole Nationale d’Informatique

**GB :** Génie logiciel et Base de données

**TMS :** Système de gestion de transport

**BD :** Base de données

**OS :** Système d’Exploitation

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

**IDE :** Environnement de Développement Intégré

**UI:** Interface Utilisateur

**N:** Numérique

**AN:** Alpha- Numérique

**D:** Date

**L:** Logique

**MCD:** Modèle Conceptuel de Données

**MLD:** Modèle logique de Données

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

**SQL :** Structured Query Language

**SOMMAIRE**

**CURRICULUM VITAE.....................................................................................................I**

**REMERCIEMENTS..........................................................................................................IV**

**LISTE DES FIGURES.......................................................................................................V**

**LISTE DES TABLEAUX..................................................................................................VI**

**LISTE DES ABRÉVIATIONS ET NOMENCLATURE..............................................VII**

1. **ANALYSE**

[1 Description du projet 2](#_Toc148336206)

[1.1 Formulation 2](#_Toc148336207)

[1.2 Objectif et besoin utilisateur 2](#_Toc148336208)

[1.3 Résultat attendu 2](#_Toc148336209)

[2 Analyse préalable 3](#_Toc148336210)

[2.1 Analyse de l’existant 3](#_Toc148336211)

[2.1.1 Organisation actuelle : 3](#_Toc148336212)

[2.1.2 Inventaire des moyens matériels et logiciels : 3](#_Toc148336213)

[2.2 Critiques de l’existant et proposition de solutions 4](#_Toc148336214)

1. CONCEPTION

[3 Conception avant-projet 5](#_Toc148336215)

[3.1 Système d’exploitation 5](#_Toc148336216)

[3.2 Langage de programmation 5](#_Toc148336217)

[3.3 Environnement de développement 6](#_Toc148336218)

[4 Conception 7](#_Toc148336219)

[4.1 Dictionnaire des données 7](#_Toc148336220)

[4.2 Règles de gestion 10](#_Toc148336221)

[4.3 Modèle Conceptuel des Données (MCD) 11](#_Toc148336222)

[4.4 Modèle Logique des Données (MLD) 11](#_Toc148336223)

1. REALISTION

[5 Installation et configuration des outils 12](#_Toc148336224)

[6 Développement de l’application 13](#_Toc148336225)

[6.1 Description de la base de données 13](#_Toc148336226)

[6.2 Elaboration des algorithmes 14](#_Toc148336227)

[6.3 Quelques captures d’écran 18](#_Toc148336228)

**CONCLUSION......................................................................................................................21**

**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.

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 entrainer des inefficacités, des retards ou une perte de productivité. C’est là que notre 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 notre logiciel en mettant l’accent sur les objectifs, les fonctionnalité clé et les défis rencontrés tout au long du projet. Nous avons suivi une approche méthodique, en commencer 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 notre 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.

1. ANALYSE

# **Description du projet**

## ***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.

## Objectif et besoin 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 BD

- Gérer les réservations des passagers

- Stocker tous les historiques des voyages effectués dans la BD

## 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 connaitre 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.

# ****Analyse préalable****

## ****Analyse de l’existant****

### Organisation actuelle :

D’après notre analyse au niveau des entreprises de transport (coopératives) à Madagascar, hormis les grandes coopératives actuelles tels que SOATRANS, COTISSE 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.

### 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.

## 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. [1]

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 boites 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 évidement 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.

1. CONCEPTION

# Conception avant-projet

## Système d’exploitation

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 facilité le développement de notre application. Ce système d’exploitation est sans doute le plus facile à manipuler grâce à son API simple à utiliser.

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.

## Langage de programmation

Parmi les langages de programmation proposés par notre Framework, nous nous sommes penchés sur le langage 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 appart 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.

## Environnement de développement

Pour mener à bien notre projet, on a utilisé le Framework Qt qui est un Framework multiplateforme développé par « The Qt Company ». 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 évènements, 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. [2]

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. 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.

# Conception

## Dictionnaire des données

***Tableau 1 :*** *Dictionnaire de donnée de la coopérative.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **COOPERATIVE** |  |  |  |
| **Nom** | **Définition** | **Type** | **Longueur** | **Observation** |
| refCoop | Référence de la coopérative | N | 2 | Valeur : 01 |
| nomCoop | Nom de la coopérative | AN | 50 | Majuscule |
| 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/yyyy |

***Tableau 2 :*** *Dictionnaire de donnée de la gestion du trajet.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **GESTION** |  |  |  |
| **Nom** | **Définition** | **Type** | **Longueur** | **Observation** |
| Id | Identification du trajet géré | N | 4 | Autoincrémenté |
| numMAT | Numéro de matriculation du voiture | AN | 8 |  |
| heure | Heure de départ du voyage | AN | 5 |  |
| lieuDepart | Lieu de départ du voyage | AN | 50 |  |
| destination | Lieu d’arrivé du voyage | AN | 50 |  |
| dateVoyage | Date du voyage | D | 10 | aaaa-MM-jj |
| placeDispo | Place disponible dans le voyage | N | 2 |  |

***Tableau 3 :*** *Dictionnaire de donnée du trajet.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **TRAJET** |  |  |  |
| **Nom** | **Définition** | **Type** | **Longueur** | **Observation** |
| refTrajet | Reference du trajet | AN | 6 |  |
| lieuDepart | Lieu de départ du trajet | AN | 50 |  |
| destination | Lieu d’arrivé du trajet | AN | 50 |  |
| heureMatin | Heure de départ matinal | AN | 5 |  |
| heureSoir | Heure de départ en soirée | AN | 5 |  |
| decalageVoyage | Nombre de jours du décalage du trajet | N | 1 |  |

***Tableau 4 :*** *Dictionnaire de donnée du véhicule.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **VEHICULE** |  |  |  |
| **Nom** | **Définition** | **Type** | **Longueur** | **Observation** |
| numMAT | Numéro de matriculation de la voiture | AN | 8 |  |
| 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 |  |
| refTrajet | Reference du trajet effectué par la voiture | AN | 6 |  |

***Tableau 5 :*** *Dictionnaire de donnée du passager.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **PASSAGER** |  |  |  |
| **Nom** | **Définition** | **Type** | **Longueur** | **Observation** |
| Id | Identification du passager | N | 5 | Autoincrémenté |
| refPlace | Reference de la place du passager dans la voiture | AN | 10 |  |
| nomPass | Nom du passager | AN | 50 | Majuscule |
| contactPass | Contact téléphonique du passager | AN | 20 |  |
| cin | Carte d’identité nationale du passager | AN | 12 |  |
| nbPlaceReserve | Nombre de place réserver par le passager | N | 2 |  |
| membreFamille | Membre de la famille du passager | AN | 10 |  |
| contactFamille | Contact téléphonique de la famille du passager | AN | 20 |  |
| isPaye | Payement des frais du passager | L | 1 |  |
| refTrajet | Reference du trajet effectué par le passager | AN | 6 |  |
| heureDepart | Heure de départ du passager | AN | 5 |  |
| dateVoyage | Date du voyage du passager | D | 16 | jjj.MMM jj aaaa |
| voiture |  | AN | 8 |  |
| lieuDepart | Lieu de départ du passager | AN | 50 |  |
| destination | Destination du passager | AN | 50 |  |

## 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 :

* **Enregistrement de la coopérative :**

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.

* **Ajout de trajets :**

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.

* **Ajout des véhicules :**

Les véhicules doivent être enregistrés avec des détails tels que sa matriculations, le nom du chauffeur avec sa coordonnée et la référence de son trajet.

* **Gestion de trajet :**

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.

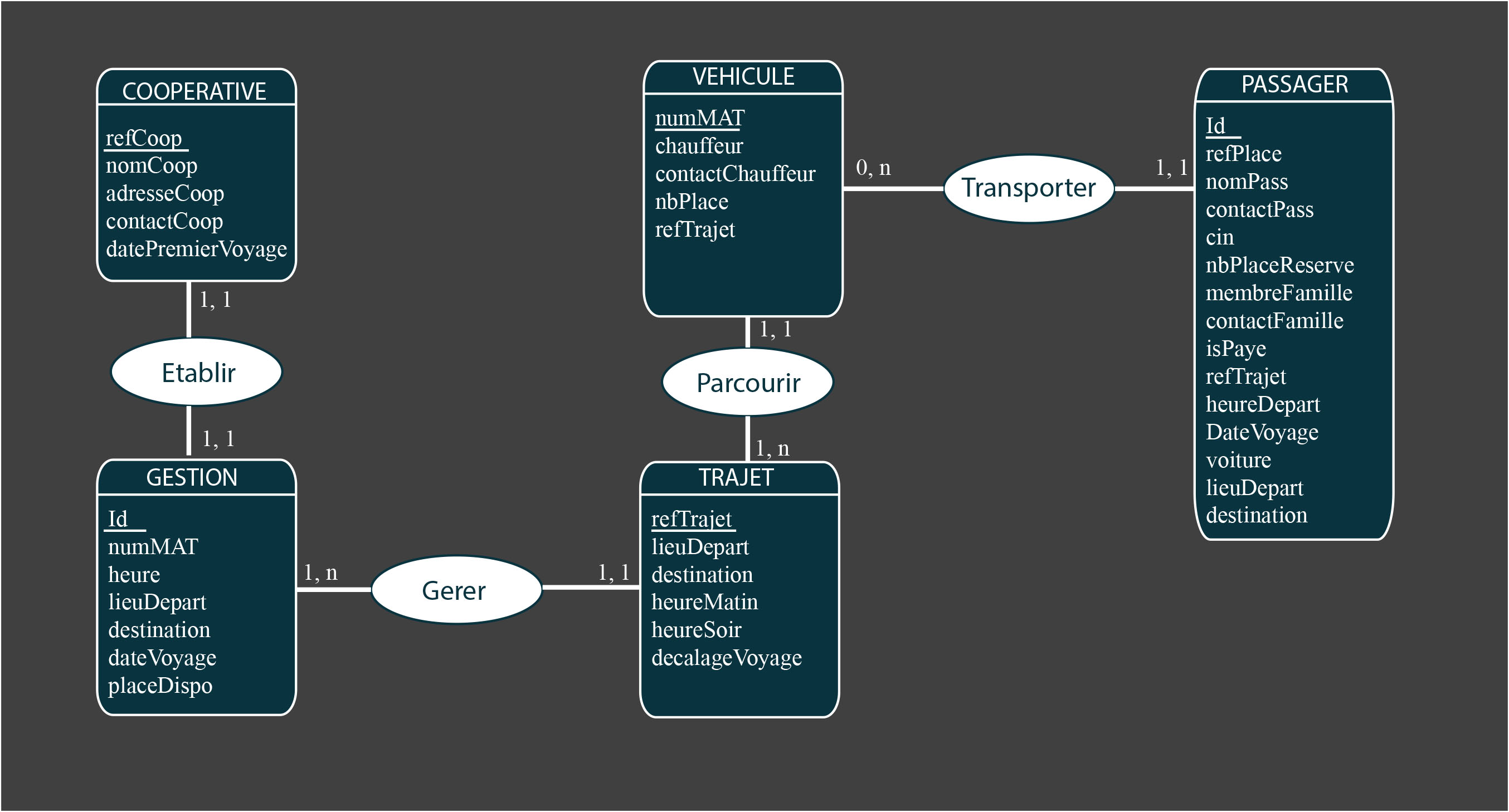
Les changements ou les suppressions n’altèrent pas les trajets déjà gérés, mais les actualisent automatiquement.

* **Gestion des réservations :**

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.

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.

## Modèle Conceptuel des Données (MCD)



***Figure 1 :*** *Modèle Conceptuel des données.*

## Modèle Logique des Données (MLD)

COOPERATIVE (refCoop, nomCoop, contactCoop, datePremierVoyage)

GESTION (Id, numMAT, heure, lieuDepart, destination, dateVoyage, placeDispo)

TRAJET (refTrajet, lieuDepart, destination, heureMatin, heureSoir, decalageVoyage)

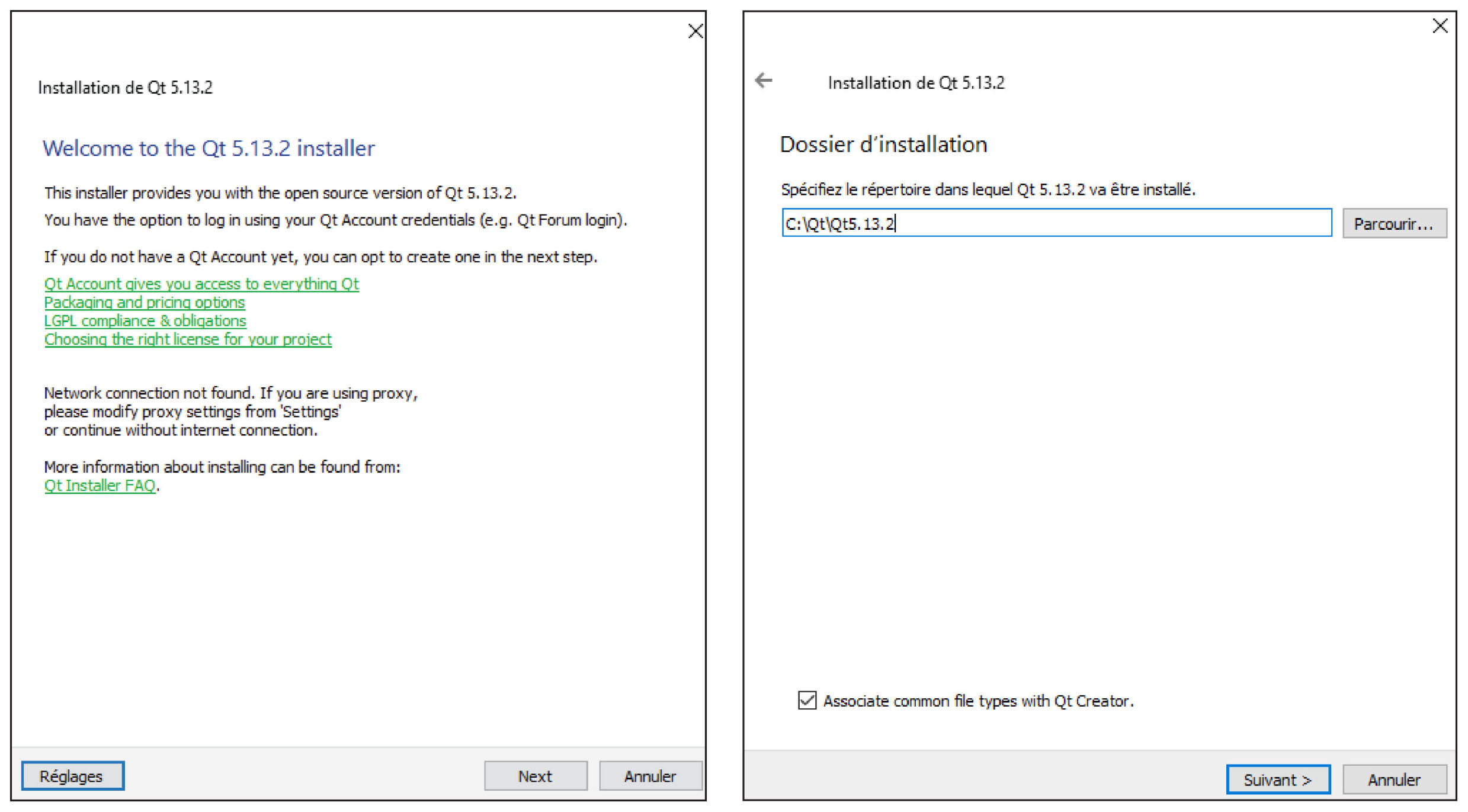
VEHICULE (numMAT, chauffeur, contactChauffeur, nbPlace, refTrajet)

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

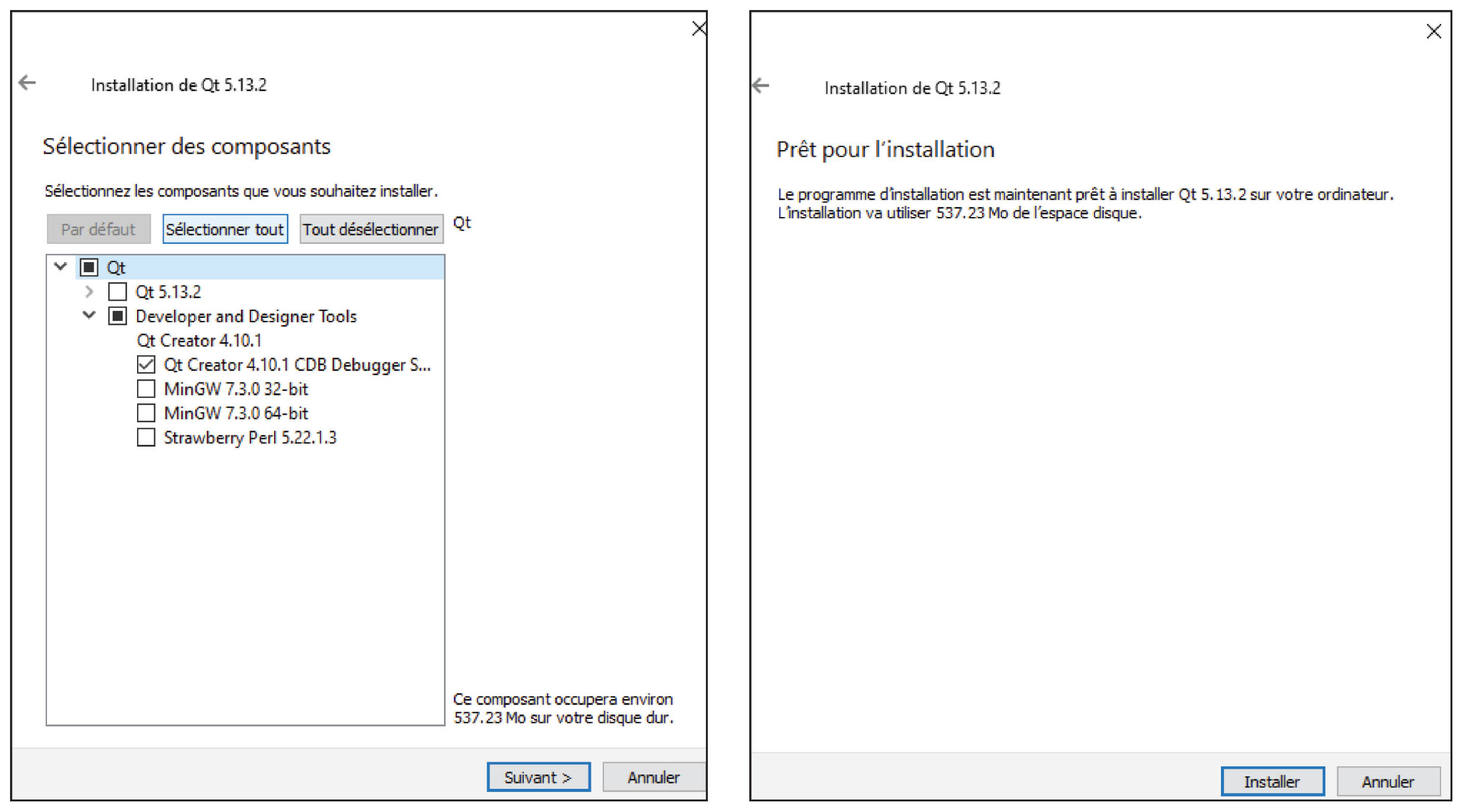
1. REALISATION

# 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**](http://www.qt.io) **, il faut entrer dans « Développent 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’installeur 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.**



***Figure 2 :*** *Premier fenêtre d’installation* ***Figure 3 :*** *Indiquer le chemin d’installation*



***Figure 4 :*** *Sélection des composants nécessaires* ***Figure 5 :*** *Installation puis terminer*

# Développement de l’application

## Description de la base de données

Concernant le SGBD utilisé, nous avons opté pour l’utilisation de SQLite qui est une base de donnée 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 donnée. De plus, SQLite peut être intégrer à de nombreux langage 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.

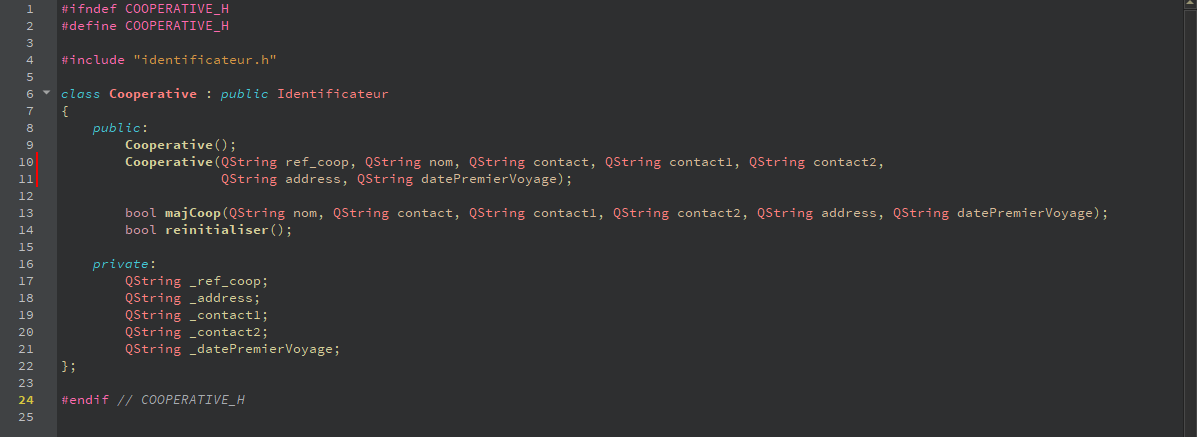
On a déjà vu précédemment sous forme de tables (objets) les entités qui assurent le flux et le fonctionnement du système de déplacement (transport). La base donnée de notre logiciel est conçue pour stocker et gérer efficacement les informations liées aux opération de transport. Elle permet de centraliser 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 coordonné (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 effectuer par unité de véhicules, la planification optimisée des itinéraires et l’automatisation de ce processus.

## 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.



***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 reinitialiser 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.

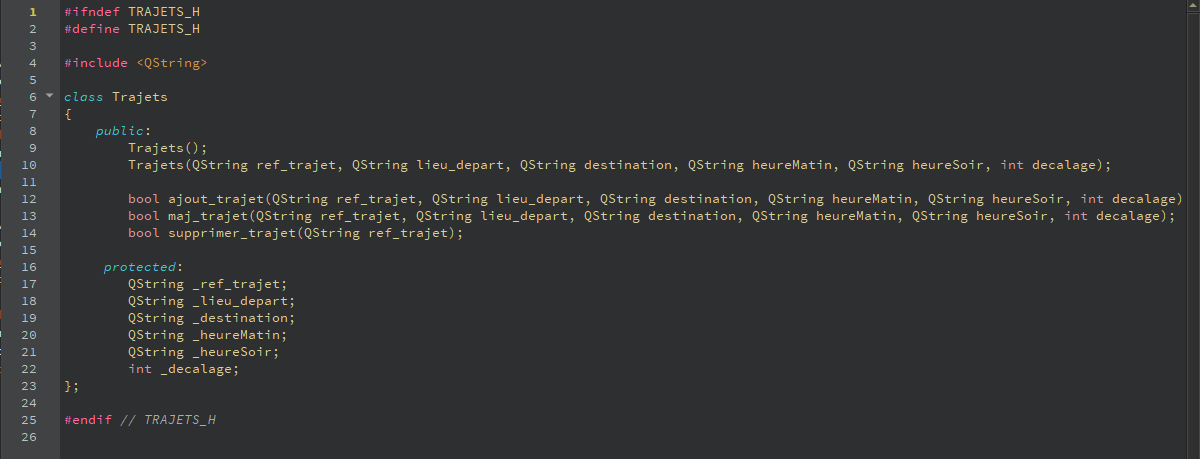


***Figure 7 :*** *Entête de la classe gestionTrajet*

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.



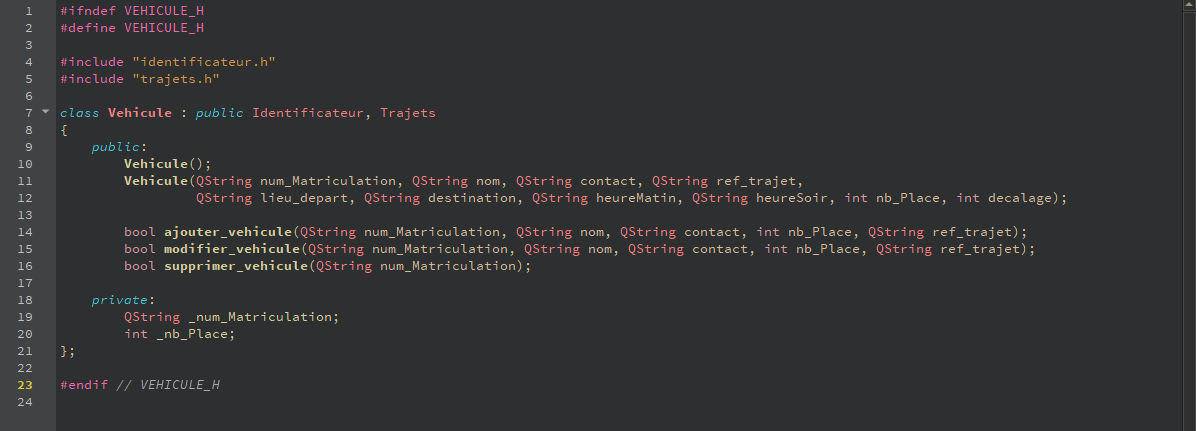
***Figure 8:*** *Entête de la classe trajet*

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 pont 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.

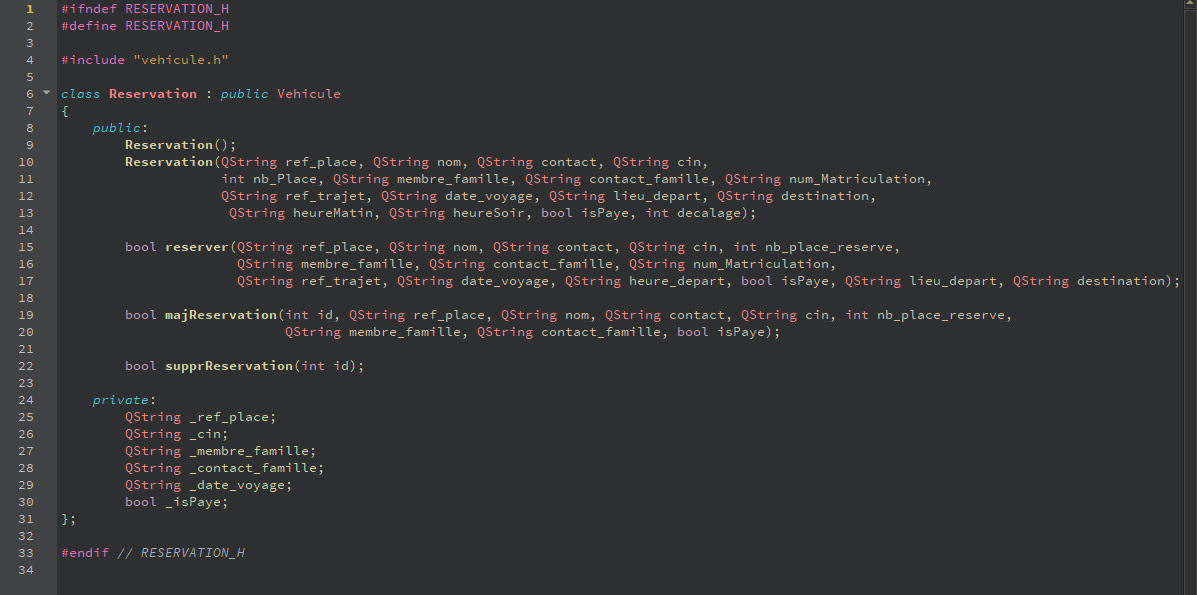


***Figure 9:*** *Entête de la classe véhicule*

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.



***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*)

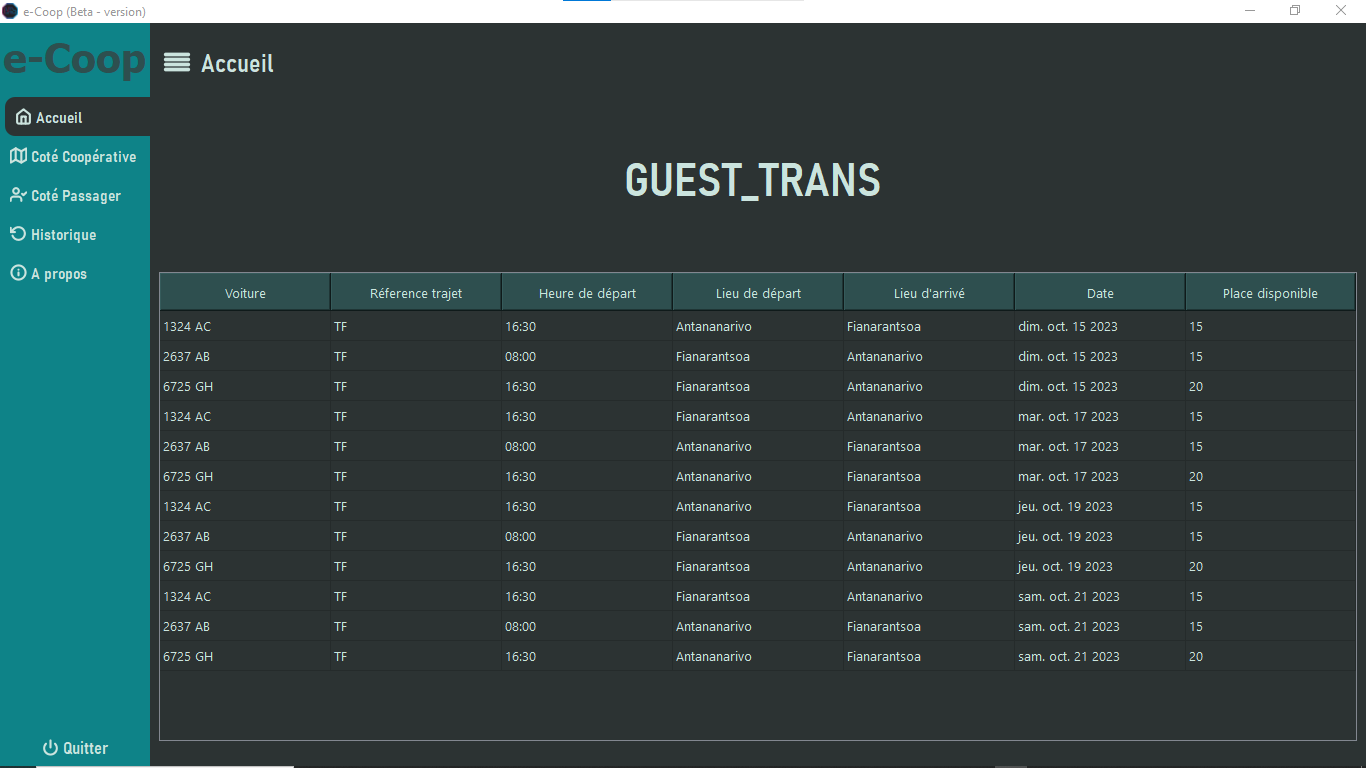
Méthodes :

* 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).

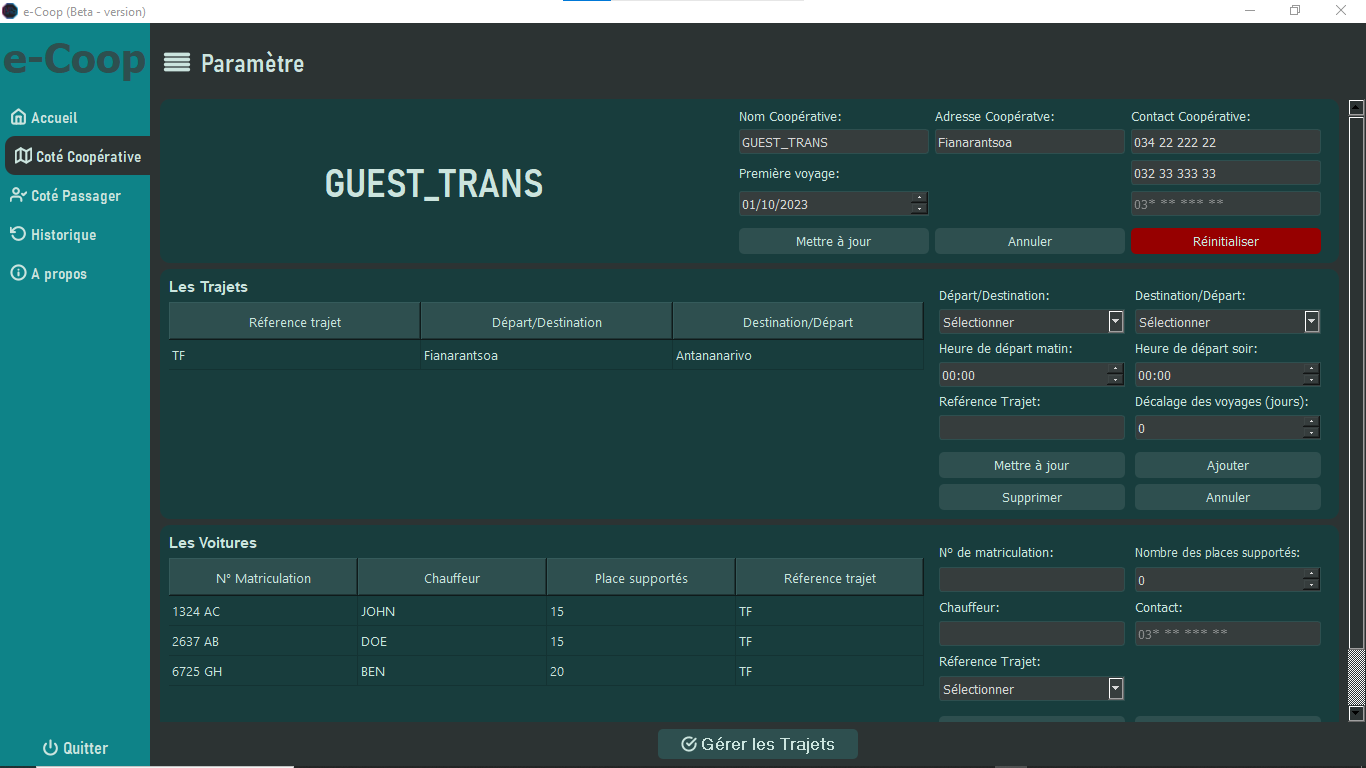
Une fois, toutes les classes créées, on passe à l’élaboration de l’algorithmes de gestions proprement dite. Voici les principales étapes de la réalisation de notre algorithme :

* 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.

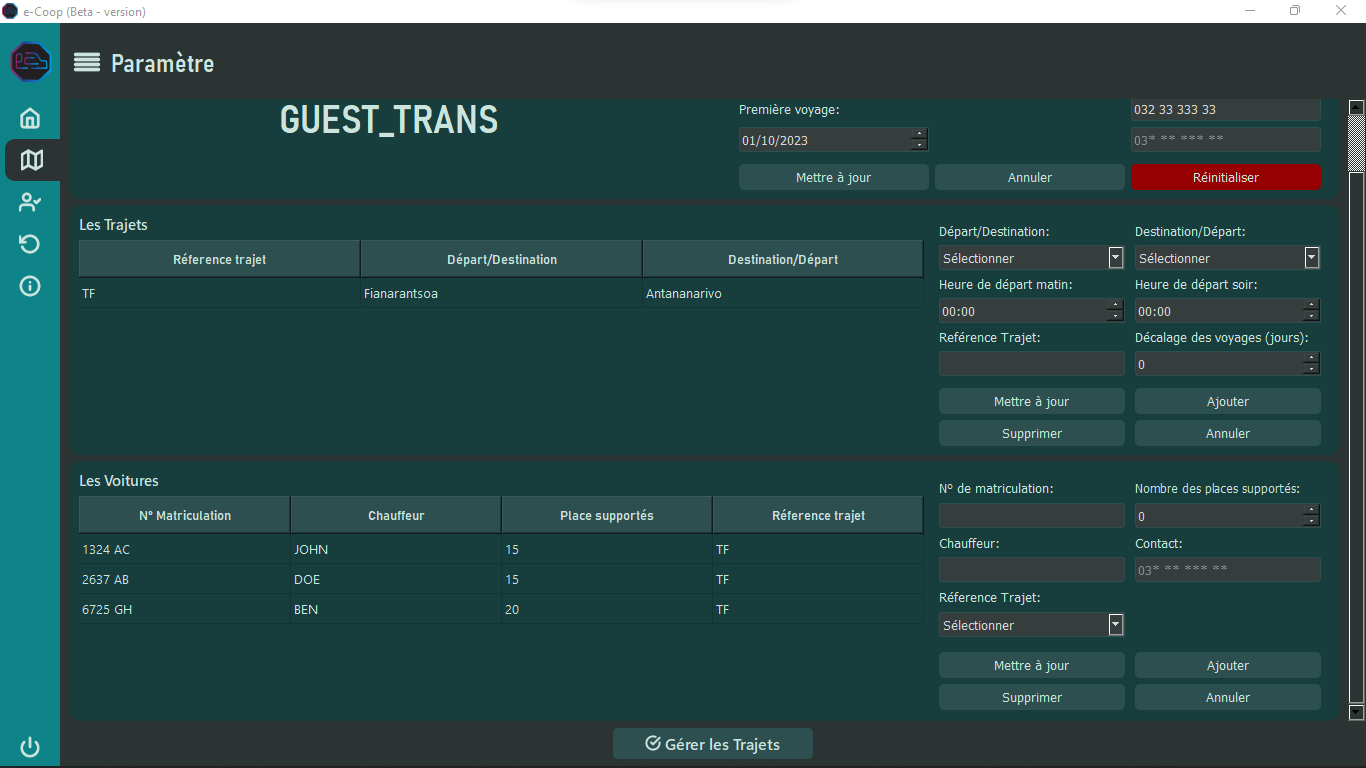
## Quelques captures d’écran



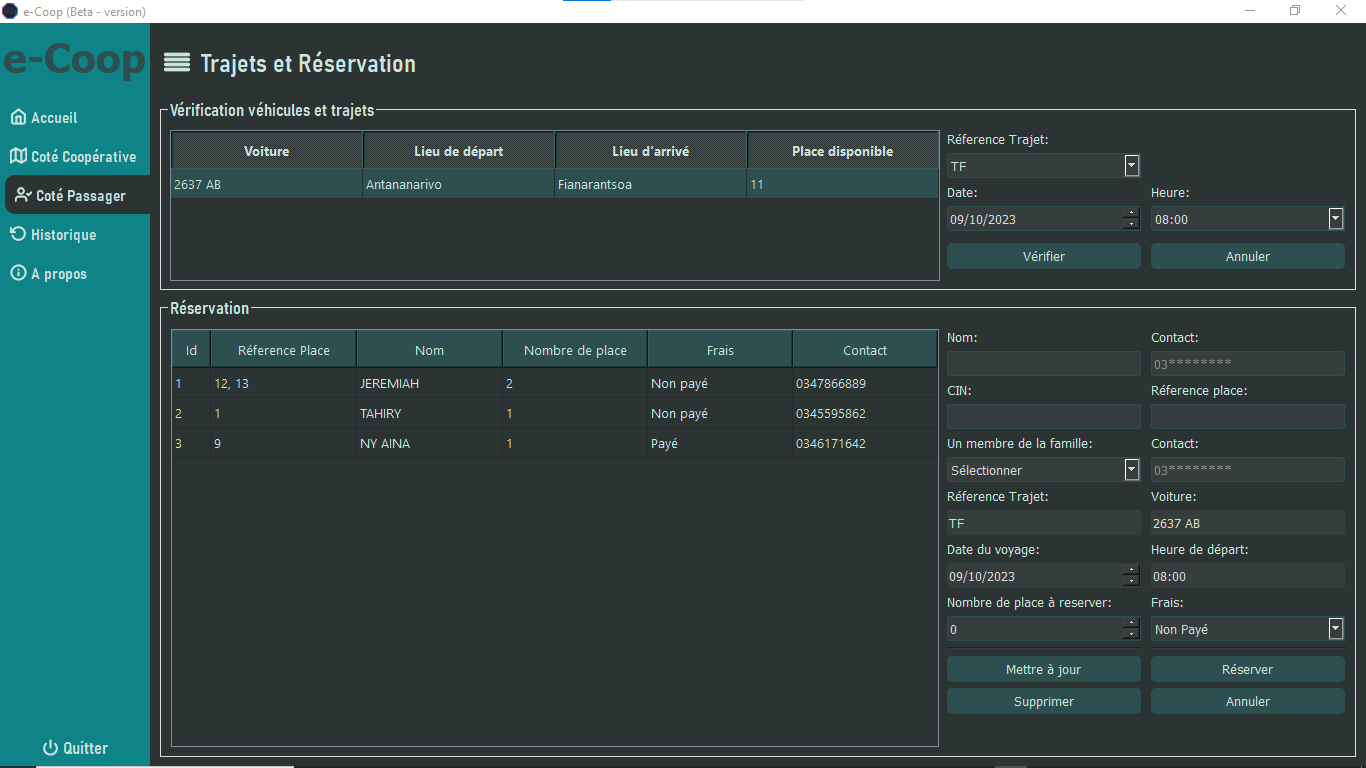
***Figure 11:*** *Page d’accueil*



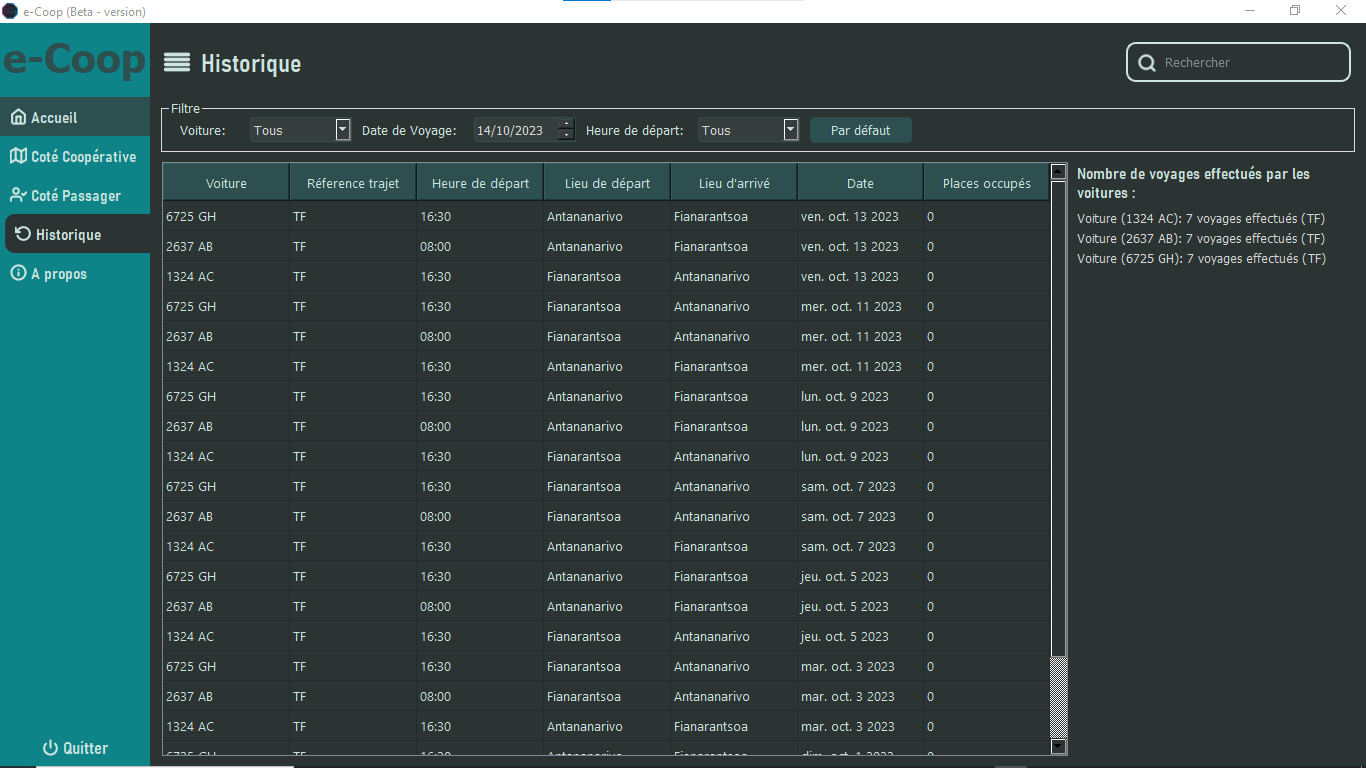
***Figure 12:*** *Interface du côté coopératif*

******

***Figure 13 :*** *Interface du côté coopératif*



***Figure 14:*** *Interface du côté passager*



***Figure 15:*** *Historique des voyages qui étaient déjà fait*



***Figure 16 :*** *A propos du 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 avancés telles que la planification des itinéraires, la gestion des véhicules, la gestion des horaire, 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 utiliser. 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. RALAIVAO Jean Christian, édition 2023, Initiation BD.pdf, ENI Fianarantsoa, 20p.
2. Gilante GESAZAFY, édition 2023, POO-Dev-GUI-Qt5\_L1.pdf, 82p.

**WEBOGRAPHIE**

[www.qt.io](http://www.qt.io)

chat.openai.com

bard.google.com

formation-video.org

<http://www.youtube.com/@codequestions>

http://www.youtube.com/@cabrogroup

<https://stackoverflow.com>

**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 un système efficace pour organiser les opérations effectuées au niveau d’une coopérative. Cette application est opérationnelle et prête à être utiliser. Cependant, certaines optimisations peuvent être ajouter pour innover notre logiciel.

**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, there are some potential optimizations that could be made to improve the software.