Dédicaces

Je dédie ce projet de PFE à

Mes très chers parents pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.

Mes frères et sœurs pour leurs encouragements permanents, et leur soutien moral et leur amour.

Toute ma famille qui a supporté mes absences et mes silences répétitifs.

Tous mes amis et collègues.

Tous mes professeurs qui m’ont guidé pendant ces quatre précieuses années.

Tous ceux qui ont contribués à la réalisation de ce projet.

Que ce projet soit l’accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible

Remerciements

En préambule à ce rapport, je souhaite adresser ici tous mes remerciements aux personnes qui m’ont apporté leur aide et qui ont ainsi contribué à l'élaboration de ce rapport.

Tout d’abord, je tiens remercie le bon Dieu, maître des temps et circonstances, pour son amour incommensurable, qui pendant cette période de crise nous a protéger et m’a donné la force nécessaire pour mener ce projet à bon port.

Je tiens à rendre un immense hommage à mon encadrant M. Mohamed Ali Hamdi pour tous ses efforts dans l’élaboration de ce projet. Il a été d’une importance très capitale par son implication, ses conseils, son sérieux, son attention et surtout par sa disponibilité.

Je remercie également l'ensemble du corps enseignant de TIME UNIVERSITE, qui pendant ces quatre années se sont donné corps et âmes de participer à ma formation.

J’exprime ma gratitude aux membres du jury, qui ont accepté d'évaluer ce projet.

J’adresse mes remerciements à tous mes amis qui m’ont toujours encouragé et surtout cru en moi : « MBUMBA N’daya Fortune, MEZNI Rania, BAGNAMA Rebecca, Christ Jordan NYAMA DITOUGOU »

Enfin, je tiens à remercier mes parents et toute ma famille pour leurs prières, leur amour, leur soutien ainsi que leurs encouragements au cours de la réalisation de ce projet et dans ma vie de tous les jours.

**Résumé**

Ce projet de PFE a pour principal objectif la digitalisation des processus de communication dans le domaine du transport, intitulé « Conception et réalisation d’une application web de covoiturage » basée sur les technologies PHP, HTML/CSS et BOOTSTRAP.

Cette application facilitera et réduira le coût de déplacements des usagers au quotidien.

**Mots clés :** PHP, MySQL, Html/CSS, UML

**Abstract**

The main objective of this PFE project is the digitization of communication processes in the transport sector, entitled “Design and implementation of a carpooling web application” based on PHP, HTML/CSS, and BOOTSTRAP technologies.

This application will facilitate and reduce the cost of travel for users on a daily basis

**Keywords:** PHP, MySQL, Html/CSS, UML

Table de matières

[Introduction Générale 10](#_Toc115095708)

[Chapitre 1 : Analyse et spécification des besoins 12](#_Toc115095709)

[**Introduction** 12](#_Toc115095710)

[I.1 Etude et Critique de l’existant 12](#_Toc115095711)

[I.1.1 Etude de l’existant 12](#_Toc115095712)

[I.1.2 Critique de l’existant 13](#_Toc115095716)

[**I.1.3 Solutions proposées** 14](#_Toc115095717)

[**I.2**  **Spécification des bésoins** 15](#_Toc115095718)

[I.2.1 Besoins fonctionnels 16](#_Toc115095719)

[I.2.2 Besoins non fonctionnels 17](#_Toc115095720)

[I.2.3 Méthodologie et Approche adoptée 17](#_Toc115095721)

[**I.2.5 Pourquoi UML** 18](#_Toc115095722)

[**Conclusion** 18](#_Toc115095723)

[Chapitre 2 : Etude conceptuelle 20](#_Toc115095724)

[**Introduction** 20](#_Toc115095725)

[**I.** **Diagramme de cas d’utilisation** 20](#_Toc115095726)

[**I.1** **Diagramme de cas d’utilisation global** 21](#_Toc115095727)

[**I.2**  **Raffinement de cas d’utilisation coté internaute** 22](#_Toc115095728)

[**I.3**  **Raffinement du cas d’utilisation coté passaser** 23](#_Toc115095729)

[**Conclusion** 35](#_Toc115095730)

[Chapitre 3 : Réalisation et implémentation 36](#_Toc115095731)

[**I.** **Introduction** 36](#_Toc115095732)

[**II.** **Environnement et outils de travail** 36](#_Toc115095733)

[Environnement Matériel 36](#_Toc115095734)

[Environnement Logiciel 37](#_Toc115095735)

[Conclusion Générale 45](#_Toc115095741)

[Netographie 46](#_Toc115095742)

Liste des Figures

[Figure 1 :Logo Blablacar 1](#_Toc72919424)3

[Figure 2 :Page D’accueil Blablacar 13](#_Toc72919425)

[Figure 3: Diagramme De Contexte Statique 16](#_Toc72919426)

[Figure 4: Diagramme de cas d’utilisation Global 21](#_Toc72919427)

[Figure 5 :Raffinement Des Cas D’utilisation Cote « Internaute» 22](#_Toc72919428)

[Figure 6: Raffinement Des Cas D’utilisation Cote « Passager» 23](#_Toc72919429)

[Figure 7: Raffinement Des Cas D’utilisation Cote « Conducteur»](#_Toc72919430) 26

[Figure 8: Raffinement Des Cas D’utilisation Cote « Administrateur» 29](#_Toc72919431)

[Figure 9: Diagramme de Sequence « S’inscrire» 31](#_Toc72919432)

[Figure 10 : Diagramme de Sequence « S’authentifier» 32](#_Toc72919433)

[Figure 11: Diagramme De Classe 34](#_Toc72919434)

[Figure 12: Interface « Accueil» 41](#_Toc72919437)

[Figure 13 : Fenêtre « Se connecter» 41](#_Toc72919438)

[Figure 14 : Fenêtre « S’Inscrire» 42](#_Toc72919438)

[Figure 15 : Interface « Admin » 43](#_Toc72919438)

[Figure 16 : Interface « Utilisateur» 44](#_Toc72919438)

Liste des Tableaux

[Tableau 1: Tableau comparatif 14](#_Toc72919444)

[Tableau 2: Description Textuelle du cas d’utilisation « S’inscrire » 23](#_Toc72919445)

[Tableau 3: Description Textuelle du cas d’utilisation « S’authentifier » 24](#_Toc72919446)

[Tableau 4: Description Textuelle du cas d’utilisation « Afficher Details Compte » 24](#_Toc72919447)

[Tableau 5: Description Textuelle du cas d’utilisation « Reserver Une Place Dans Un Trajet» 25](#_Toc72919448)

[Tableau 6 : Description Textuelle du cas d’utilisation « Proposer un trajet »](#_Toc72919450) 27

[Tableau 7 : Description Textuelle du cas d’utilisation « Modifier trajet » 28](#_Toc72919450)

[Tableau 8 : Description Textuelle du cas d’utilisation « supprimer un trajet  » 28](#_Toc72919451)

[Tableau 9 : Description Textuelle du cas d’utilisation « Consulter les contacts » 30](#_Toc72919451)

[Tableau 10 : Description Textuelle du cas d’utilisation « Supprimer compte   » 30](#_Toc72919451)

[Tableau 11: Environnement materiel 36](#_Toc72919451)

Introduction Générale

Le secteur du transport en Tunisie ne cesse de faire face à plusieurs difficultés.

En effet, pour certaines personnes telles que les étudiants, les fonctionnaires ou les employés du secteur privé, les moyens de transport en commun dont le bus et le métro léger constituent les seuls moyens pour arriver à sa destination malgré toutes les difficultés auxquelles elles pourraient être confrontées durant le trajet.

Dès les premières heures du matin, les usagers du transport en commun sont confrontés à tout ce qui pourrait changer rapidement leur humeur, d’une humeur gaie à une humeur exécrable.

A l’heure de pointe, les métros se font de plus en plus rares. De ce fait, les gens se collent les uns aux autres, d’autres usagers s’accrochent aux portes et les métros démarrent quand même les portes-ouvertes mettant la vie des passagers en danger.

De plus, au quotidien nous retrouvons le secteur des transports en grèves inattendues, en des encombrements ou en embouteillages dans les heures de pointe.

Afin de réduire ces nombreux problèmes urbains, le concept de covoiturage est né le covoiturage : c’est le fait de mettre en relation des conducteurs et des passagers qui circuleront ensemble dans un même véhicule le temps d'un voyage ou régulièrement. Ce covoiturage s'adresse à tous les automobilistes qui souhaitent rouler plus intelligemment en réalisant des économies et préservant notre environnement.

C’est dans ce cadre que se situe notre projet. Le travail présenté, intitulé "Conception et réalisation d’une application web de covoiturage", consiste à développer une application qui facilite et réduit le coût de déplacements des usager au quotidien. Pour réaliser notre projet, nous allons suivre la démarche suivante :

* Le premier chapitre intitulé « Cadre général du projet » est consacré à la présentation de la concept covoiturage, ainsi qu’une vision détaillée sur le cadre du projet tout en incluant le contexte du projet puis une petite critique sur l’existant pour pouvoir à la suite présenter la solution. Par la suite, nous allons mettre en évidence la méthodologie suivie durant la réalisation de ce projet.
* Le deuxième chapitre est une modélisation conceptuelle où nous mettrons en évidence ses éléments les plus importants, qui sont l’analyse de besoins et conception UML
* Enfin, le dernier chapitre intitulé « Réalisation » expose les outils et les Framework utilisés durant la réalisation de notre projet ainsi que quelques interfaces de cette dernière.

Enfin, nous clôturons ce document par une conclusion générale dans laquelle nous analysons les résultats aboutis et nous dévoilons les éventuelles perspectives de notre projet.

Chapitre 1 : Analyse et spécification des besoins

# **Introduction**

Dans Cette Partie, Nous Présentons Une Etude De L’existant En Décrivant La Démarche Actuelle Adoptée Pour Demander Une Offre De Covoiturage Tel Que L’application « Annuaire Pro Tunipages » Nous Décrivons Ensuite La Solution Proposée Et La Méthodologie Adoptée.

## I.1 Etude et Critique de l’existant

L’étude de l’existant est une phase importante pour comprendre les applications actuelles sur le marché, y déterminer les points forts et poursuivre avec une critique.

Dans cette section, nous présentons une étude basée sur les observations des différentes applications dans le secteur de transport à l’échelle internationale et nationale.

## I.1.1 Etude de l’existant

Nous trouvons plusieurs applications qui répondent aux besoins des réservations à distance des moyens de transport. Parmi ces applications, nous citons l’application étudiée ci-dessous.

Cette partie consiste à introduire la meilleure solution existante dans le marché.

* BlaBlaCar

Anciennement covoiturage.fr, le site **BlaBlaCar** est l’une des références dans le domaine du covoiturage. BlaBlaCar rassemble le plus grand nombre d’utilisateurs.

A vous de renseigner votre itinéraire et de sélectionner le conducteur de votre choix.

Le détail des trajets est très bien réalisé en fonction des nombreux avis. Vous pourrez également publier votre itinéraire pour chercher des personnes qui nous accompagneront, vous permettant ainsi de réduire vos dépenses. La communauté est importante et il est assez simple de trouver des conducteurs. Le site prend une commission de 7% à chaque voyage. Le logo de l’application BlaBlaCar est présenté par la figure1.



## 

## Figure 1 : Logo de BlaBlaCar

## 

## Figure 2 : Page d’accueil de l’application BlaBlaCar

## I.1.2 Critique de l’existant

Le tableau 1 ci-dessous présente les avantages et les inconvénients des solutions qui existent aujourd’hui au niveau de l’interface, de frais des fonctionnalités fournies aux utilisateurs.

|  |  |
| --- | --- |
| Plateforme | BlaBlaCar |
| Langage de programmation | **.** 70% PHP,  **.**20% Python,  **.**10% Java |
| Interface | **.** Claire  **.** Interaction |
| Frais | **.** -0,60 euros de frais fixes + 7,92% du prix du trajet, pour une réservation effectuée plus 72 heures avant le départ.  **.** -1,19 euros de frais fixes + 12,48% pour une réservation effectuée de 24 heures avant le départ. |
| Fonctions fournies | **.** Disponible  **.** Payement  **.** Evaluer le conducteur |

**Tableau 1** : Tableau comparatif

# 

# **I.1.3 Solutions proposées**

Pour faire face aux limites citées dans la section précédente on a pensé à une application de covoiturage qui doit apporter des améliorations des solutions existantes qui vont répondre aux attentes des utilisateurs en offrants de nouvelles fonctionnalités :

* Permet à ses utilisateurs de communiquer entre eux et faciliter la rencontre.
* Un bon design et ergonomie bien étudiée.
* Application sécurisée
* Les utilisateurs peuvent consulter les historiques de trajets.
* Les passagers peuvent évaluer le conducteur.
* Plateforme gratuite.
* Proposer des chemins optimums selon point de départ et point d’arrivée.

# 

# **I.2 Spécification des bésoins**

La spécification des besoins est une approche scientifique qui raisonne en termes de fonctions devant être assurées par un produit : elle consiste à caractériser et hiérarchiser les fonctions d’un système. C’est pour cela que nous présentons dans ce qui suit tous les besoins fonctionnels et non fonctionnels du système, ainsi que les différents acteurs du système.

**Identification des acteurs**

En UML, un acteur est un élément de modèle qui interagit avec un système. En tant qu'élément modèle, il peut s'agir d'une personne abstraite (par exemple « client ») ou d'un autre système externe. Dans la pratique de la modélisation, il est recommandé de définir le rôle sans ambiguïté avec des stéréotypes – par exemple « Utilisateur », « Intervenant » ou « Système externe ».

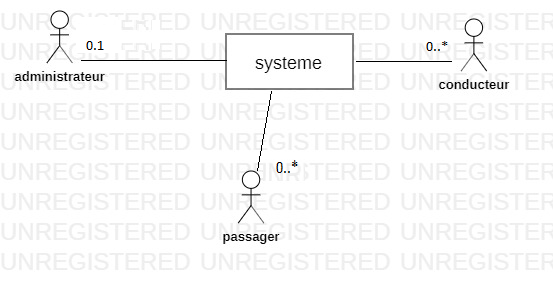
En principe, cependant, un acteur peut non seulement être

* Le **créateur d'une action** (comme l'initiateur d'un dossier de candidature)
* Mais aussi **destinataire d'un résultat attendu**.

Notre système est conçu pour être utilisé par trois profils d’acteurs :

* **Administrateur :** L’administrateur est le seul à avoir accès à l’information complète du système. Il assure la gestion globale du système. En d’autres termes, il a le droit de consulter, vérifier ou supprimer n’importe quel compte.
* **Conducteur :** Chaque conducteur peut publier un ou plusieurs trajets, gérer les demandes de réservation et consulter son historique.
* **Passager :** Les passagers sont ceux qui bénéficient le plus du système. Ils peuvent lancer diverses recherches pour trouver des trajets qui répondent à leurs besoins et faire des réservations. De plus, ils peuvent consulter leur historique de réservations.
* *Diagramme de contexte statique*

Ce diagramme d’UML permet simplement de montrer la relation des différents acteurs avec le système. Il spécifie le nombre d’instances de chaque acteur relié au système à un moment donné.



**Figure3** : Diagramme de contexte statique

Pour expliquer le diagramme ci-dessus, nous pouvons dire qu’à un instant t nous pouvons avoir 0 ou un administrateur qui manipule l’application, 0 ou plusieurs conducteur et passagères qui sont en train d’utiliser l’application.

.

## I.2.1 Besoins fonctionnels

Un besoin fonctionnel spécifie l’action qu’un système doit être capable d’effectuer, hors contrainte physique : besoin spécifiant un comportement d’entrée/sortie d’un système.

Dans ce contexte notre application de covoiturage scolaire, implémente principalement les fonctionnalités suivantes :

* **Module des inscriptions :** le système doit permettre aux conducteurs et passager de pouvoir s’inscrire de façon autonome via le site web.
* **Module de gestion des trajets :** le système doit permettre l’ajout, modification et suppression d’un trajet par un conducteur ou un passager. Ces derniers auront accès à tous leurs trajets passés et pourront créer de nouveaux trajets à partir d’un trajet déjà effectué dans le passé.
* **Module de gestion des réservations :** le système doit aussi permettre aux covoiturés de recherche des trajets. Des recherches peuvent s’effectuer suivant des critères précis. Suite à l’affichage des résultats, le passager peut choisir parmi les plus « intéressants » pour lui. En cas de changement de plan, un système de notification envers le conducteur ou le passager est disponible (messagerie), afin d’aviser que le passager n’embarque plus avec le conducteur ou que le conducteur souhaite annuler le trajet.
* **Module de gestion des comptes :** Le système permet à chacun des membres de faire la gestion de son compte. Il sera possible de modifier des informations personnelles ou préférences.

## I.2.2 Besoins non fonctionnels

Un besoin non fonctionnel est un besoin spécifiant des propriétés du système, telles que les contraintes liées à l’environnement et l’implémentation, les exigences en matière de performances, de dépendances de plate-forme, de facilité de maintenance, d’extensibilité et de fiabilité.

Dans notre système on distingue les besoins non fonctionnels suivant :

* **Sécurité :** La plateforme doit assurer la sécurité pour les utilisateurs (Authentification).
* **Convivialité :** Ergonomie des interfaces homme machine et facilité d’utilisation.
* **Contrôle :** Saisie contrôle selon les choix prédéfinis.

## I.2.3 Méthodologie et Approche adoptée

Avant de programmer l’application et se lancer dans l’écriture du code : il faut tout d’abord organiser les idées, les documenter, puis organiser la réalisation en définissant les modules et les étapes de la réalisation Cette démarche antérieure à l’écriture que l’on appelle modélisation ; son produit et un module. La modélisation consiste à créer une représentation virtuelle d’une réalité de telle façon à faire ressortir les points auxquels on s’intéresse. Dans le cadre de notre projet on a utilisé la méthodologie UML pour la modélisation des différents diagrammes.

**I.2.4 Présentation d’UML :**

Pour spécifier, modifier et construire les documents nécessaires au bon développement d’un logiciel orienté objet, nous avons choisi un langage qui offre un standard de modélisation pour représenter l’architecture logicielle, le Langage de Modélisation Unifié « UML ». UML est un langage de modélisation graphique et textuel, c’est une notation qui permet de décrire une représentation conceptuelle de la réalité et aussi d’unifier plusieurs travaux liés à l’analyse et conception orienté objet. La conception UML peut être traduite en utilisant plusieurs types de diagrammes.

**I.2.5 Pourquoi UML**

Le langage de modélisation unifié (UML) a été conçu pour répondre à ces besoins.

1. Fournissez aux utilisateurs un langage de modélisation visuel expressif prêt à l'emploi afin qu'ils puissent développer et échanger des modèles significatifs.
2. Fournir des mécanismes d'extensibilité et de spécialisation pour étendre les concepts de base.
3. Être indépendant des langages de programmation et des processus de développement particuliers.
4. Fournir une base formelle pour comprendre le langage de modélisation.
5. Encourager la croissance du marché des outils OO.
6. Prend en charge les concepts de développement de niveau supérieur tels que les collaborations, les Framework, les modèles et les composants.
7. Intégrez les meilleures pratiques.

# **Conclusion**

Dans ce chapitre, nous avons présenté le sujet, ainsi que les grands choix techniques adoptés pour la mise en place du prototype attendu. Dans le chapitre suivant, nous allons examiner la solution proposée, afin de définir les besoins fonctionnels et opérationnels de l’application à développer.

Chapitre 2 : Etude conceptuelle

# **Introduction**

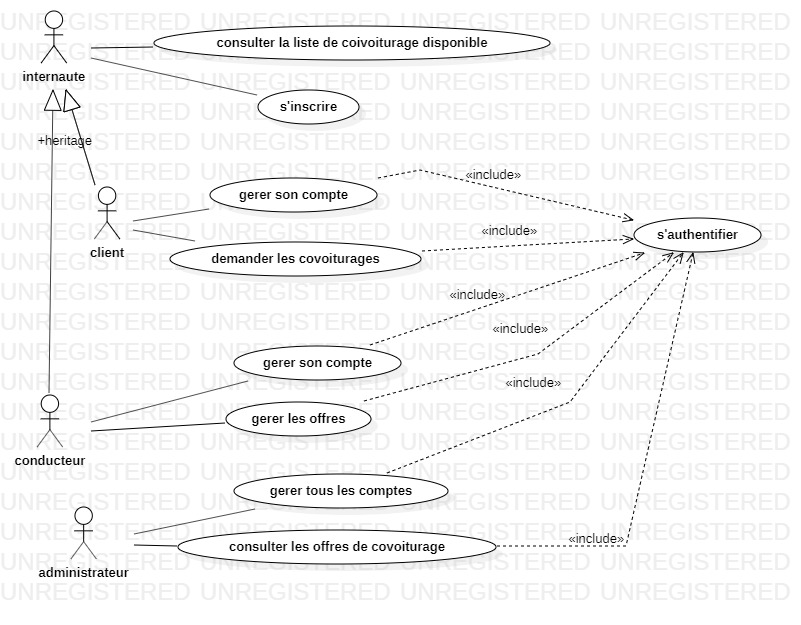
La conception est la phase la plus importante dans le cycle de développement d’une application. Elle permet de décrire d’une manière ambiguïté le fonctionnement d’un système afin de simplifier la concrétisation. C’est dans ce cadre que nous présentons dans ce chapitre les diagrammes de cas d’utilisation et la description textuelle des chaque cas. Par la suite nous détaillons la conception de l’application en établissons les diagrammes de séquences, le diagramme de classe et le diagramme d’activité.

# **Diagramme de cas d’utilisation**

Un diagramme de cas d’utilisation est une fonctionnalité de système qui produit un résultat observable pour un utilisateur potentiel du système. Le cas d’utilisation regroupe une famille de scénario ou caque scénario est un traitement particulier du système.

Pendant notre analyse des besoins nous avons pu identifier les actions importantes que nous présentons ci-dessous et nous les modélisons par la suite avec les diagrammes cas utilisation d’UML.

# **I.1 Diagramme de cas d’utilisation global**



**Figure4** : Diagramme de cas d’utilisation global

# **I.2 Raffinement de cas d’utilisation coté internaute**



**Figure5** : Raffinement des cas d’utilisation côté « Internaute »

Description textuelle de cas d’utilisation « **S’inscrire** »

|  |  |
| --- | --- |
| **Objectif :** | Ouvrir un compte dans l’application. |
| **Acteurs :** | Internaute |
| **Pré-condition :** | L’acteur ne doit pas avoir déjà un compte. |
| **Post-condition :** | Créer un compte correct qui permet la connexion, et l’accès aux services de l’application. |
| **Scénario nominal :** | 1. Demande le formulaire d’inscription 2. Le système envoie le formulaire 3. L’internaute remplit les champs du formulaire (Nom, Prénom…) 4. Soumettre le formulaire 5. Vérification des informations par le système, et l’envoie d’une alerte de confirmation |
| **Exceptions :** | 1. Si les champs obligatoires sont vides, ou les données entrées ne vérifies pas la forme demandée : une alerte d’erreur est affichée, et demande de ressaisie 2. Si le compte est déjà existant : envoie d’un message d’erreur 3. Problème de connexion avec la base de données |

**Tableau2** : description textuelle de cas d’utilisation « S’inscrire »

# **I.3 Raffinement du cas d’utilisation coté passaser**



**Figure6** : Raffinement des cas d’utilisation côté « passager »

Description textuelle des cas d’utilisation « **S’authentifier** »

|  |  |
| --- | --- |
| **Titre** | S’authentifier |
| **Objectif** : | Cette partie doit offrir aux passagers de connecter au système |
| **Acteurs** : | Passager |
| **Précondition** : | Le passager doit avoir un compte |
| **Postcondition** | Passager authentifié |
| **Scénario nominal** : | 1. L’utilisateur demande la page d’authentification 2. Le système envoie le formulaire d’authentification 3. Le passager saisit son login et son mot de passe 4. Il confirme en cliquant sur le bouton « se connecter » 5. Le système affiche l’interface d’accueil propre au passager |
| **Exceptions :** | * Si les paramètres d’identification sont invalides, le système affiche « l’identifiant est invalide vous devez retaper votre identifiant ». |

**Tableau3** : description textuelle de cas d’utilisation « S’authentifier »

Description textuelle des cas d’utilisation « **Afficher détails compte** »

|  |  |
| --- | --- |
| **Objectif** : | Cette partie doit offrir aux passagers des interfaces permettant de consulter les informations de leur compte (Nom, Prénom, adresse, email…) |
| **Acteurs** : | Passager |
| **Pré-condition** : | Être authentifié |
| **Scénario nominal** : | 1- Le passagers clique sur son profile  2- Le système affiche toutes les informations ajoutées dans le compte |
| **Exceptions :** | * Problème de connexion avec la base de données |

**Tableau4** : description textuelle de cas d’utilisation « Afficher détails compte »

Description textuelle des cas d’utilisation « **Réserver une place dans un trajet** »

|  |  |
| --- | --- |
| **Objectif** : | Réservé dans le trajet souhaité |
| **Acteurs** : | Passager |
| **Pré-condition** : | Être authentifié pour pouvoir consulter le trajet et le nombre de place disponible. |
| **Post-condition** : | La réservation est effectuée et le nombre de place de cette dernière est décrémenté |
| **Scénario nominal** : | 1. Le passager recherche un trajet (ville de départ, ville d'arrivée, date) 2. Le système vérifie l'existence réelle des informations saisies. 3. Le système effectue une recherche basée sur les données saisies par le passager et affiche les trajets existants par ordre de pertinence. 4. Le passager choisit le trajet qui lui convient et réserve sa place en appuyant sur le bouton "Réserver". 5. Le système réduit le nombre de places dans le trajet. |
| **Exceptions :** | * Un ou les deux lieux (départ et/ou arrivée) introuvables : retour à (1). * Aucun trajet ne correspond à la recherche : alerte de notification. * Le passager ne trouve pas de trajet à son goût : soit il revient à l'étape (1), soit il revient au menu principal. |

**Tableau5** : description textuelle de cas d’utilisation « Réserver une place dans un trajet »

**I.4 Raffinement de cas d’utilisation côté conducteur**

****

**Figure7** : Raffinement des cas d’utilisation côté « Conducteur »

Description textuelle des cas d’utilisation « **Proposer un trajet** »

|  |  |
| --- | --- |
| **Objectif** : | Créer un trajet, et le proposer aux passagers de l’application |
| **Acteurs** : | Conducteur |
| **Pré-condition** : | Être authentifié |
| **Post-condition** : | Le trajet est créé et ajouté à la liste des trajets disponibles |
| **Scénario nominal** : | 1. Conducteur demande la page ajout trajet 2. Le système renvoie le formulaire d’ajout d’un trajet 3. Le conducteur remplit le formulaire, saisie le point de départ, point d’arrivée et l’ensemble des points traçant le trajet (son parcours) 4. Saisie des autres informations : heures, date, nombre de place disponible 5. Le système vérifie les informations saisies 6. Une notification de succès va être afficher |
| **Scénario alternatif :** | 1. Si un champ est vide ou mal saisi : afficher erreur et demande de ressaisie 2. Si le conducteur ne confirme pas la demande, le trajet n’est pas créé |

**Tableau6** : description textuelle de cas d’utilisation « Proposer un trajet »

Description textuelle des cas d’utilisation « **Modifier trajet** »

|  |  |
| --- | --- |
| **Objectif** : | Pouvoir modifier le trajet proposé |
| **Acteurs** : | Conducteur |
| **Pré-condition** : | Être authentifier, et avoir des trajets suggérés |
| **Post-condition** : | Trajet modifié |
| **Scénario nominal** : | 1. Sélectionner le trajet à modifier 2. Le système affiche les informations du trajet sélectionné 3. Le conducteur établie les modifications désirées sur le trajet : date, heure… 4. Vérification du système 5. L’enregistrement des modifications dans la base de données 6. Notifier le conducteur par l’enregistrement |
| **Exceptions :** | 1. Echec de connexion avec la base de données |

**Tableau7** : description textuelle de cas d’utilisation « Modifier trajet »

Description textuelle des cas d’utilisation « **supprimer un trajet** »

|  |  |
| --- | --- |
| **Objectif** : | Supprimer un trajet |
| **Acteurs** : | Conducteur |
| **Pré-condition** : | Être authentifié |
| **Post-condition** : | La suppression d’un trajet |
| **Scénario nominal** : | 1. Le système affiche la liste des trajets et leurs informations. 2. Le conducteur choisit le trajet à supprimer. 3. Le système affiche le formulaire contenant les données relatives à ce trajet en mode écriture. 4. Le conducteur clique sur le bouton « Supprimer ». 5. Le système supprime le compte et affiche un message de succès |
| **Exceptions :** | Problème de connexion avec la base de données |

**Tableau8** : description textuelle de cas d’utilisation « supprimer un trajet »

**I.4 Raffinement de cas d’utilisation coté administrateur**

****

**Figure8** : Raffinement des cas d’utilisation côté « Administrateur »

Description textuelle des cas d’utilisation « **Consulter les contacts** »

|  |  |
| --- | --- |
| **Objectif** : | Cette partie doit offrir à l’administrateur une interface permettant de consulter les contacts des médecins et ces patients |
| **Acteurs** : | Administrateur |
| **Pré-condition** : | Être authentifié |
| **Scénario nominal** : | 1. L’administrateur demande la consultation des contacts  2. le système lui affiche la liste des contacts |
| **Scénario alternatif :** | Problème de connexion avec la base de données |

**Tableau9** : description textuelle de cas d’utilisation « **Consulter les contacts** »

Description textuelle des cas d’utilisation « **Supprimer compte** »

|  |  |
| --- | --- |
| **Objectif** : | Pouvoir supprimer un compte |
| **Acteurs** : | Administrateur |
| **Pré-condition** : | Être authentifié |
| **Post-condition** : | Le compte est supprimé, en le gardant dans l’archive |
| **Scénario nominal** : | 1. Demande de suppression du compte 2. Le système demande la confirmation 3. Affirmation de l’administrateur |
| **Scénario alternatif :** | Si l’acteur annule la demande en déclinant la confirmation, la suppression du compte est annulée à son tour |

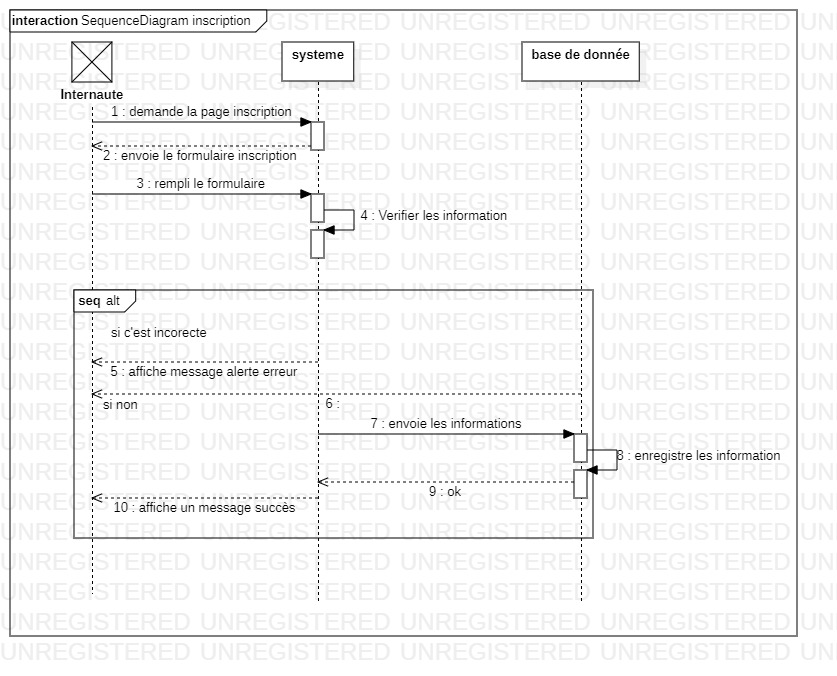
**Tableau10** : description textuelle de cas d’utilisation « Supprimer compte »

**II Diagramme de séquence**

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique. Dans cette partie nous présentons les diagrammes de séquence des cas d’utilisation « S’inscrire » ainsi que « s’authentifier » de plus « Envoyer une question ».

**II.1 Diagramme de séquence du cas d’utilisation « S’inscrire »**

La figure 9 montre le scénario d’exécution de l’inscription à l’application.



**Figure9** : Diagramme de séquence « S’inscrire »

**II.2 Diagramme de séquence du cas d’utilisation « S’authentifier »**

La figure 10 montre le scénario d’exécution de l’authentification à l’application.

****

**Figure10** : Diagramme de séquence « S’authentifier »

**II.3 Diagramme de séquence du cas d’utilisation « Envoyer une question »**

Cette figure montre le scénario d’exécution du processus d’envoi d’une question

**II.4 Diagramme de classe**

Le diagramme de classes [UML](https://en.wikipedia.org/wiki/Unified_Modeling_Language) est une notation graphique utilisée pour construire et visualiser des systèmes orientés objet. Un diagramme de classes dans le langage de modélisation unifié (UML) est un type de diagramme de structure statique qui décrit la structure d'un système en montrant :

* Des classes,
* Leurs attributs,
* Opérations (ou méthodes),
* Et les relations entre les objets.



**Figure11** : Diagramme de classe

La figure 11 met en évidence les classes de notre sprint :

• **Passager :** est la classe qui contient les informations d’un passager aussi son login et son mot de passe

• **Conducteur :** est la classe qui contient les informations des comptes conducteurs (nom, prénom, adresse, login, mot de passe ...)

• **Expériences :** est la classe qui contient les expériences des médecins.

• **Contact :** est la classe qui contient toutes les informations d’un contact entre un médecin et un patient.

# **Conclusion**

Ce chapitre présente les acteurs de l’application et les différentes étapes de la conception de l’application. Le chapitre suivant aura consacré le dernier chapitre qui présente la réalisation de notre application.

Chapitre 3 : Réalisation et implémentation

# **Introduction**

La satisfaction des besoins et des exigences requises se voit dans la réalisation des solutions. C’est dans ce chapitre que nous exposons le fruit de notre étude après le traitement de la partie conceptuelle du projet. Nous allons commencer en premier lieu par la présentation de l’environnement matériel et logiciel utilisé ainsi que le choix technologique pour le développement des solutions demandées ensuite, nous détaillerons techniquement notre solution en présentant par acteur un plan de l’application et un aperçu sur le travail réalisé.

# **Environnement et outils de travail**

## Environnement Matériel

Pour mener à bien ce projet nous avons travaillé sur des ordinateurs portables dont les caractéristiques sont les suivantes :

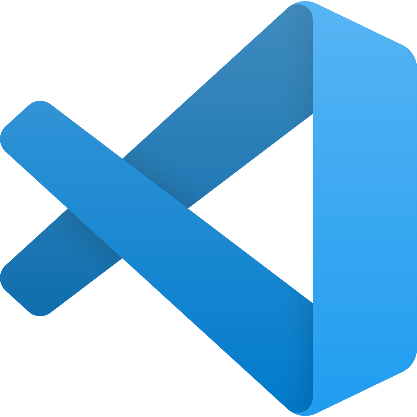
|  |  |
| --- | --- |
|  | **Ordinateur portable** |
| Marque | HP Pavillon Gaming. |
| Modèle | 5cd0015scr |
| Processeur | AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz |
| Mémoire vive | 8,0 Go. |
| Carte graphique | GeForce GTX. |
| Disque dur | 512 SSD. |
| Système d’exploitation | Windows 11 Pro. |

**Tableau11** : Environnement matériel

# 

## 

## Environnement Logiciel

**Environnement de développement**

1. **Visual Studio Code (VS Code)**

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.

1. **XAMPP**

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide.

**Outil de modélisation UML**

1. **StarUML**

Logiciel de modélisation open source utilisé dans le langage de modélisation unifié où la modélisation système et logicielle est prise en charge avec le concept UML pour générer du code pour différents langages en fournissant différents types de diagrammes UML pour développer une plate-forme UML rapide et flexible afin que les exigences et la conception soient donné avant de commencer le projet où il est appelé comme une grande approche de conception initiale pour les utilisateurs qui ont besoin des conceptions logicielles pour améliorer leur projet et donc pour prendre en charge leurs diagrammes de conception UML dans le système.

**Choix Technologiques**

**Technologies de programmation**

1. **HTML**

HTML signifie **Langage de balisage hypertexte**. C'est un langage de balisage standard pour la création de pages Web. Il permet la création et la structure de sections, de paragraphes et de liens à l'aide d'éléments HTML (les blocs de construction d'une page Web) tels que des balises et des attributs.

1. **CSS**

CSS signifie **Cascading Style Sheets language** et est utilisé pour styliser les éléments écrits dans un langage de balisage tel que HTML. Il sépare le contenu de la représentation visuelle du site. La relation entre HTML et CSS est fortement liée car HTML est le fondement même d'un site et CSS est l'ensemble de l'esthétique d'un site Web entier.

1. **PHP**

PHP est un langage de script côté serveur qui est utilisé pour développer des sites Web statiques ou des sites Web dynamiques ou des applications Web. PHP signifie HyperText Pre-processor, qui signifiait auparavant Personal Home Pages. Les scripts PHP ne peuvent être interprétés que sur un serveur sur lequel PHP est installé.

Les ordinateurs clients accédant aux scripts PHP nécessitent

1. **JAVASCRIPT**

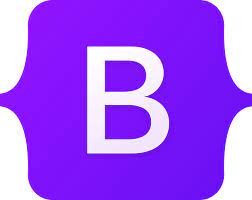
JavaScript est un langage de programmation de scripts, principalement utilisé dans les pages web interactives. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont équipés de constructeurs permettant de générer leurs propriétés. Le langage a été créé en 1995 par Brendan Eich pour le compte de Netscape Communications Corporation.

1. **SQL**

Structured Query Language (SQL) est un langage de programmation standardisé utilisé pour gérer des bases de [**données relationnelles**](https://www.techtarget.com/searchdatamanagement/definition/relational-database)et effectuer diverses opérations sur les données qu'elles contiennent. Initialement créé dans les années 1970, SQL est régulièrement utilisé non seulement par les administrateurs de bases de données, mais également par les développeurs écrivant des scripts d'intégration de données et les analystes de données cherchant à configurer et exécuter des requêtes analytiques.

**Framework, outils et technologies utilisés**

1. **Bootstrap**

[Bootstrap](https://getbootstrap.com/) est un Framework de développement Web gratuit et open-source. Il est conçu pour faciliter le processus de développement Web de sites Web réactifs et mobiles en fournissant un ensemble de syntaxes pour les conceptions de modèles.

En d'autres termes, Bootstrap aide les développeurs Web à [créer des sites Web](https://www.hostinger.com/tutorials/how-to-make-a-website) plus rapidement car ils n'ont pas à se soucier des commandes et des fonctions de base. Il se compose de scripts [HTML](https://www.hostinger.com/tutorials/what-is-html) , [CSS](https://www.hostinger.com/tutorials/what-is-css) et JS pour diverses fonctions et composants liés à la conception Web.

1. **Git**

Git est un système de contrôle de version distribué [gratuit et open source conçue pour tout gérer, des petits aux très grands projets, avec rapidité et efficacité.](https://git-scm.com/about/free-and-open-source)

Git est [facile à apprendre](https://git-scm.com/doc) et a une [petite empreinte avec des performances ultra-rapides](https://git-scm.com/about/small-and-fast) . Il surclasse les outils SCM tels que Subversion, CVS, Perforce et ClearCase avec des fonctionnalités telles que [la création de branches locales bon marché, des](https://git-scm.com/about/branching-and-merging)[zones de préparation](https://git-scm.com/about/staging-area) pratiques et [plusieurs flux de travail](https://git-scm.com/about/distributed) .

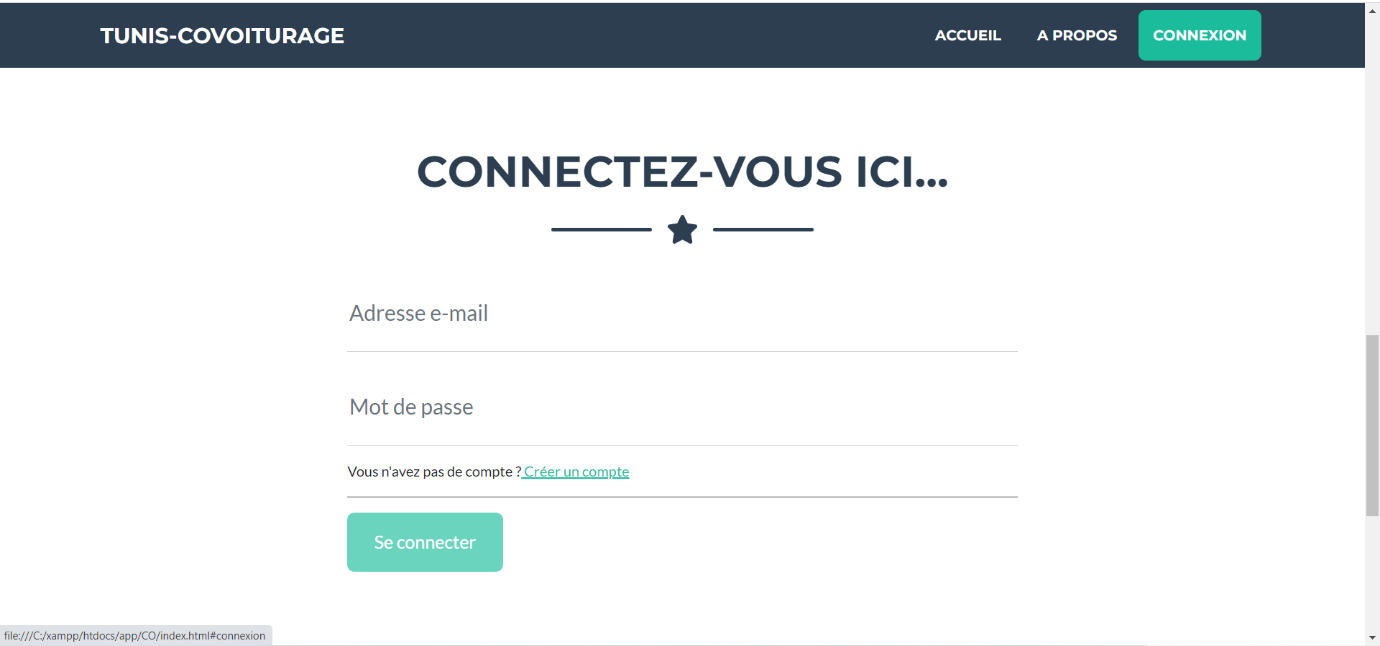
**Interfaces de l’application**

## Interface « Accueil »

C’est le point de départ de notre plateforme

**Figure12** : Interface « Accueil »

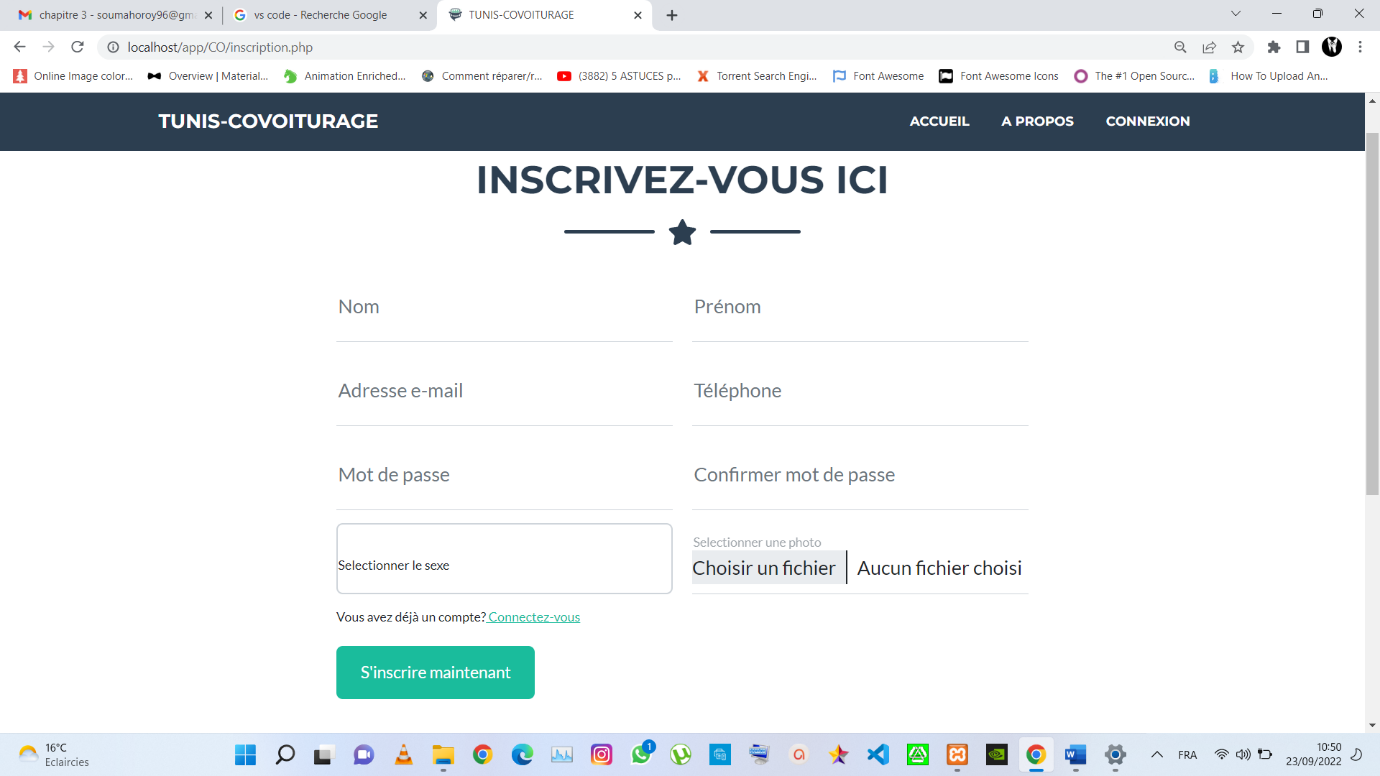
## Fenêtre « Se connecter »

Ici, l’utilisateur est invité à se connecter afin d’accéder à sa session, s’il est déjà inscrit sur la plateforme

**Figure13** : Fenêtre « Se connecter »

## Fenêtre « S’inscrire »

Ici, l’utilisateur est invité à s’inscrire sur la plateforme afin de bénéficier des services offerts



**Figure14** : Fenêtre « S’inscrire »

## Interface « Administrateur »

Cette interface permet à l’administrateur d’avoir un contrôle absolu sur les interactions de notre plateforme

**Figure15** : Interface « Administrateur »

## Interface « Administrateur »

****Ici, l’utilisateur accède à un certain nombre de fonctionnalités bien définies sur notre plateforme

**Figure16** : Interface « Utilisateur »

Conclusion Générale

Dans le cadre de notre Projet de Fin d’Année (PFA), nous avons conçu et développé une application de covoiturage en ligne qui permet de mettre en contact un passager avec un conducteur d’une manière flexible et sécurisé.

Le présent rapport décrit toutes les étapes par lesquelles nous sommes passées afin d’atteindre le résultat attendu. Nous avons essayé de construire notre application étape par étape en utilisant la méthodologie UML et le modèle en cascade durant ces trois derniers mois.

En premier lieu, dans le rapport présent, nous avons introduit le contexte général. Puis nous avons identifié les différentes exigences et besoins grâce à la réalisation de l’étude de l’existant, et introduit le cadre méthodologique adopté. Par la suite, nous avons procédé à la conception de notre travail, en citant les trois diagrammes statique et dynamique qui sont le diagramme de cas d’utilisation, le diagramme de séquence et le diagramme de classe toutes les tâches à réaliser après avoir attribuer les degrés d’importance. À la fin, nous avons pu fournir les interfaces de notre application Web.

Malgré toutes les difficultés rencontrées, nous avons eu la chance d’améliorer nos connaissance théoriques et pratiques ; ce qui nous a aidé à réaliser la totalité de notre application en respectant l’aspect sécuritaire et en préparant la documentation nécessaire.

Enfin, le bilan de ces trois derniers mois de travail est dans l’ensemble positif, plusieurs perspectives sont envisageables pour l’extension de ce projet tel que ; l’envoie d’une notification à l’administrateur de l’application lors de chaque action importante faite par les autres utilisateurs, et de créer une version mobile.

.

Netographie