



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE

DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : **Génie Logiciel**

Par :

DAICH HICHAM

BELAIDABI BEN YUCEF

Sur le thème

Conception et réalisation d'une application web Java EE pour la gestion du covoiturage en Algérie

Soutenu publiquement le ... / 10 / 20 20 à Tiaret devant le jury composé de :

Mr Mokhtari Ahmed	Grade	Université MAA	Président
Mr khaled abid	Grade	Université MAA	Encadreur
Mr Benathmane Lalia	Grade	Université MAA	Examinateur

2019-2020

Remerciements

Je remercie Dieu « ALLAH » de nous avoir donné la force, et la volonté pour bien achever ce modeste travail.

En préambule, on souhaite adresser ici tous nos remerciements aux personnes qui m'ont apportée leur aide et qui ont ainsi contribué à l'élaboration de ce mémoire.

*A Monsieur **Mr. Abid. Khaled** mon encadreur pour sa supervision, ses conseils scientifiques et sa disponibilité.*

*Notre profond respect va aux membres du jury : **Mr. Benathman lalia** et **Mr. Mokhtari ahmed**, pour avoir accepté de juger ce travail.*

*J'exprime ma gratitude à tous les internautes rencontrés (virtuellement) lors de mes recherches effectuées et qui ont accepté de répondre à mes questions, spécialement « **Mr. Recteur Mimouni Abdenbi** »*

Dédicace

*Je dédie ce travail à ...
A ma très chère Méré*

Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles

A mon très cher père

Tu as toujours été à mes cotes pour me, et mes soutenir et m'encourager.

À mes chers frères. Qui n'ont pas cessée de me conseiller, encourager et soutenir tout au long de mes études. Que dieu les protège et leurs offre la chance et le bonheur

*Spéciale dédicace A mes Neveux, **Chiheb, Iheb, Lamis.***

*Sans oublier mon chère binôme **Belaidabi Benyoucef** Pour son soutien moral, sa patience et sa compréhension tout au long de ce projet, A tous mes amis et en particulière **Boucharef Yahia, Houari Zegai** et à tous les étudiants de notre promotion et à tous les gens qui m'ont aidé et soutenu tout au long de ce projet.*

Daiche Hicham

Dédicace

Je dédie ce travail à ...

A ma très chère mère

Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

Votre prière et votre bénédiction m'ont été d'un grand secours pour mener à bien mes études.

A mon très cher père

Tu as toujours été à mes cotes pour me, et mes soutenir et m'encourager.

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

A mes tendres frères et sœurs, pour tant de confiance, d'amour, de patience et d'abnégation.

*A mon, chère Binôme **Daïche Hicham**, ainsi qu'à toute sa famille.
Pour toutes les peines et difficultés partagées ensemble lors de l'élaboration de ce projet. A tous mes amis.*

A tous les étudiants de notre promotion

Et à tous les gens qui m'ont aidé et soutenu tout au long de ce projet.

Belaidabi Benyoucef

Résumé

Le covoiturage est un mode de transport qui consiste à partager le véhicule d'un particulier pour un trajet commun. Cette pratique n'est pas à but lucratif, puisque l'intérêt du propriétaire du véhicule est seulement de partager les frais d'utilisation de la voiture et d'avoir un peu de compagnie pour son voyage. Mais les avantages du covoiturage ont dépassé ce cadre. En effet, le covoiturage joue un rôle important en termes d'impact environnemental puisqu'il contribue à réduire les émissions de gaz à effet de serre et les émissions polluantes. Il pourrait aussi être "la solution" miraculeuse qui aiderait à mettre fin aux conséquences néfastes des congestions routières et contribuer ainsi à améliorer la qualité de transport, surtout dans les grandes villes et dans les périodes estivales. La réussite d'une telle pratique est fortement liée à l'adhésion d'un grand nombre d'adeptes. Pour se faire, il est primordial d'instaurer un système de gestion de covoiturage ralliant efficacité et qualité de service en termes de gestion des réponses aux nombreuses demandes provenant des participants. Notre solution c'est la conception et la réalisation d'une application web Java EE pour la gestion du covoiturage en Algérie.

Les mots clé : Covoiturage, JavaEE, application web.

Abstract

Carpooling is a mode of transportation that involves sharing an individual's vehicle for a common trip. This practice is not for profit, since the interest of the owner of the vehicle is only to share the cost of using the car and to have some company for his trip. But the benefits of carpooling have gone beyond this framework. Indeed, carpooling plays an important role in terms of environmental impact since it contributes to reducing greenhouse gas emissions and polluting emissions. It could also be "the miraculous solution" which would help to put an end to the harmful consequences of road congestion and thus contribute to improving the quality of transport, especially in big cities and in summer periods. The success of such a practice is strongly linked to the adhesion of a large number of followers. To do so, it is essential to set up a carpooling management system combining efficiency and quality of service in terms of managing responses to the numerous requests from participants. Our solution is the design and production of a Java EE web application for the management of carpooling in Algeria.

The key words : Carpooling, JavaEE, web application.

List des abréviations

Java EE :	Java Platform, Enterprise Edition
MVC :	Modèle-Vue-Contrôleur
HTTP :	HyperText Transfer Protocol
URL :	Uniform Resource Locator
JSTL :	Java Standard Tag Library
JSP :	Java Server Pages
EJB :	Enterprises JavaBeans
EIS :	Enterprise Information System
XML :	Extensible Markup Language
XHTML :	Extensible HyperText Markup Language
GUI :	Graphical User Interface
POJO :	Plain Old Java Object
ACC :	Application Client Container
ERP :	Enterprise Resource Planning
SQL :	Structured Query Language
JNDI :	Java Naming and Directory Interface
IIOP :	Internet Inter-Orb Protocol
RMI :	Remote Method Invocation
JSTL :	Java Server Pages Standard Tag Library
JSF :	Java Server Faces
JDBC :	Java Data Base Connectivity
JPA :	Java Persistence API
JTA :	Java Transaction API
JMS :	Java Message Service API
JACC :	Java Authorization Service Provider Contract for Containers
JASPIC :	Java Authentication Service Provider Interface for Containers
JAAS :	Java Authentication and Authorization Service
JAXR :	Java API for XML Registries
JAXP :	Java API for XML Processing
JAXB :	Java Architecture for XML Binding
JAX-WS :	Java API for XML Web Services
JAX-RS :	Java API for RESTful Web Services
JNDI :	Java Naming and Directory Interface
TC :	transport covoiturage
HTML :	HyperText Markup Language
CSS :	Cascading Style Sheets
JDK :	Java Development Kit
IDE:	Integrated Development Environment
XAMPP:	X cross Apache MariaDB Perl PHP

Liste des figures

Figure 1:Architecture MVC [1].....	4
Figure 2:Architecture Java EE standard [2]	7
Figure 3:Communication Java EE entre les différents tiers [2]	9
Figure 4:Les composants de Java EE	10
Figure 5:Architecture évoluée Java EE	11
Figure 6:Serveur Java EE et conteneurs	13
Figure 7:Croissance du nombre de sites de covoiturage Depuis 1990.....	26
Figure 8:Les fonctionnements du site BlaBlaCar [14].....	28
Figure 9:Economies annuelles réalisées sur un trajet quotidien domicile-travail [23]	32
Figure 10:: Economies pour les automobilistes [24].....	32
Figure 11:Pourquoi on choisit le covoiturage ? [29].....	35
Figure 12:Diagramme cas d'utilisation de passager	44
Figure 13:Diagramme cas d'utilisation de conducteur	45
Figure 14:Diagramme cas d'utilisation de l'administrateur	46
Figure 15:Diagramme cas d'utilisation global.....	47
Figure 16:Diagramme de séquence du cas consulté des offres	48
Figure 17:Diagramme des séquences détaillées Rechercher Trajet	49
Figure 18:Diagramme des séquences détaillées Réserver un covoiturage.....	50
Figure 19:Diagramme des séquences détaillées Ajouter Trajet	51
Figure 20:Diagramme des séquences détaillées Consulter Passager	52
Figure 21:Diagramme des séquences détaillées Consulter Notification	53
Figure 22:Diagramme des séquences détaillées Ajouter une réclamation	54
Figure 23:Diagramme des séquences détaillées Ajouter Conducteur	55
Figure 24:Diagramme des séquences détaillées Consulter Conducteur.....	56
Figure 25:Diagramme des séquences détaillées Consulter la liste des affectations.....	57
Figure 26:Diagramme de classe globale	58
Figure 27:Page d'accueil.....	64
Figure 28:Formulaire d'identification.....	65
Figure 29:Inscription sur notre site	65
Figure 30:Chargement les informations de trajet	66
Figure 31:Chargement la Localisation de trajet	67
Figure 32:Modifier le Trajet.....	68
Figure 33:Supprimer le Trajet	69
Figure 34:Le Conducteur Consulter son profile.....	69
Figure 35: Les détaille de conducteur avec son trajet	70
Figure 36:Consulter tous les covoiturages	71
Figure 37:Contacter les Administrateur	71

Liste des tableaux

Tableau 1:les spécifications Java EE [4].....	19
---	----

Table de matière

1	Introduction générale :	1
1.1	Contexte :	1
1.2	Problématique :	1

Chapitre I

Les applications web et javaEE

I.1	Introduction :	3
I.2	Modèle-Vue-Contrôleur (MVC) :	3
I.3	Principe général de MVC :	3
I.4	Vue d'ensemble de MVC :	4
I.4.1	Le modèle :	5
I.4.2	La vue :	5
I.4.3	Contrôleurs et actions :	6
I.5	Composants et architectures Java EE :	6
I.5.1	Les composants clients ou tiers Client :	8
I.5.1.1	Les clients Web :	8
I.5.1.2	Les clients Applets :	8
I.5.1.3	Les applications clientes :	8
I.5.2	Les composants Web ou tiers Web :	10
I.5.3	Les composants métier ou tiers Métier :	10
I.6	Les conteneurs Java EE :	12
I.7	API et services Java EE :	14
I.7.1	Java Servlet (Technologie Java EE - Conteneur Web) :	14
I.7.2	Java Server Pages (JSP) (Technologie Java EE - Conteneur Web) :	14
I.7.3	Java Server Pages Standard Tag Library (JSTL) (Technologie Java EE - Conteneur Web) :	14
I.7.4	Java Server Faces (JSF) (Technologie Java EE - Conteneur Web) :	15
I.7.5	Java Data Base Connectivity (JDBC) (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application) :	15

I.7.6. Java Persistence API (JPA) (Technologie Java SE - Conteneur Web, Conteneur EJB et Conteneur Application) :	15
I.7.7. Java Transaction API (JTA) (Technologie Java EE - Conteneur Web, Conteneur EJB :	16
I.7.8 Enterprise JavaBeans (EJB) (Technologie Java EE - Conteneur EJB) :	16
I.7.9 Java Message Service API (JMS) (Technologie Java EE – Conteneur Web, Conteneur EJB et Conteneur Application) :	16
I.7.10 JavaMail (Technologie Java SE - Conteneur Web, Conteneur EJB) :	17
I.7.11 Java Authorization Service Provider Contract for Containers (JACC) (Technologie Java EE - Conteneur Web, Conteneur EJB) :	17
I.7.12 Java Authentication Service Provider Interface for Containers (JASPIC) (Technologie Java EE - Conteneur Web, Conteneur EJB) :	17
I.7.13 Java Authentication and Authorization Service (JAAS) (Technologie Java EE - Conteneur Web, Conteneur EJB) :	17
I.7.14 Java API for XML Registries (JAXR) (Technologie Java EE - Conteneur Web, Conteneur EJB, Conteneur Application) :	17
I.7.15 Java API for XML Processing (JAXP) (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application) :	17
I.7.16 Java Architecture for XML Binding (JAXB) (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application) :	18
I.7.17 Java API for XML Web Services (JAX-WS) (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application) :	18
I.7.18 Java API for RESTful Web Services (JAX-RS) (Technologie Java EE - Conteneur Web, Conteneur EJB) :	18
I.7.19 Java Naming and Directory Interface (JNDI) (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application) :	19
I.7.20 JavaBeans Activation Framework (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application) :	19
I.8. Conclusion :	20

Chapitre II

Etude Critique des différentes solutions proposées pour le covoiturage

II.1 Introduction :	22
II.2 Définition de covoiturage :	22
II.3 Le Principe de covoiturage :	22

II.4 Les formes du covoiturage :	23
II.4.1 Le covoiturage spontané :	23
II.4.2 Le covoiturage organisé :	24
II.5 Les sites de covoiturage grand-public	25
II.6 Les systèmes opérationnels du covoiturage :	25
II.6.1 Le covoiturage statique :	25
II.6.1.1 Exemple du site statique :	26
i. BlaBlaCar :	26
ii. Uber :	28
iii. Yasir :	29
II.6.2 Le covoiturage dynamique :	30
II.6.3 Exemple du site dynamique :	30
II.6.3.1 Avego:	30
II.6.3.2 Comuto :	31
II.7 Les Avantages de covoiturage :	31
II.7.1 L'aspect économique :	31
II.7.2 Un geste pour l'écologique :	33
II.7.3 Des trajets plus rapides :	34
II.7.4 Une route plus conviviale :	34
II.8 Les problèmes techniques :	35
II.9 Conclusion :	36

Chapitre III

Spécification, Analyse, Conception

III.1 Introduction :	38
III.2 Notion de covoiturage :	38
III.3 Charte du Covoiturage:	38
III.4 Spécifications des besoins:	41
III.4.1 Les besoins fonctionnels :	41
III.4.2 Les besoins non fonctionnels :	41
III.5 Analyse des besoins :	42

III.6 Conception :	43
III.6.1 Identification des cas d'utilisation :	43
III.6.1.1 Diagramme cas d'utilisation de passager :	44
III.6.1.2 Diagramme cas d'utilisation de conducteur :	45
III.6.1.3 Diagramme cas d'utilisation de l'administrateur :	46
III.6.1.4 Diagramme cas d'utilisation Global :	47
III.6.2 Diagrammes des séquences détaillées:	48
II.6.2.1 Diagrammes des séquences détaillées de la partie Passager :	48
i. Diagramme de séquence du cas consulté des offres	48
ii. Diagramme des séquences détaillées Rechercher Trajet	49
iii. Diagramme des séquences détaillées Réserver un covoiturage	50
III.6.2.2 Diagrammes des séquences détaillées de la partie Conducteur :	51
i. Diagramme des séquences détaillées Ajouter Trajet	51
ii. Diagramme des séquences détaillées Consulter Passager	52
iii. Diagramme des séquences détaillées Consulter Notification	53
iv. Diagramme des séquences détaillées Ajouter une réclamation	54
III.6.2.3 Diagrammes des séquences détaillées de la partie Administrateur :	55
i. Diagramme des séquences détaillées Ajouter Conducteur	55
ii. Diagramme des séquences détaillées Consulter Conducteur	56
iii. Diagramme des séquences détaillées Consulter la liste des affectations	57
III.6.3 Diagramme de classe globale :	58
III.7 Conclusion :	59

Chapitre IV

Implémentation

IV.1 Introduction :	61
IV.2 Les choix techniques :	61
IV.2.1 Les outils utilisés :	61
IV.2.2 Les Langages :	63
IV.3 Présentation de notre application web :	64
IV.3.1 Page d'accueil : Elle constitue le point d'entrée de notre application web.	64

IV.3.2 Formulaire d'identification :	65
IV.3.4 Ajouter Trajet :	66
IV.3.4.1 Chargement les informations de trajet :	66
IV.3.4.2 Chargement la Localisation de trajet :	67
IV.3.5 Modifier le Trajet:	68
IV.3.6 Supprimer le Trajet:	69
IV.3.7 L'utilisateur Consulter son profile :	69
IV.3.8 Les détaille de trajet:	70
IV.3.9 Consulter tous les covoiturage :	71
IV.3.10 Contacter les Administrateur :	71
IV.4 Les Avantages de notre application web:	72
IV.5 Conclusion :	72
Conclusion générale :	74
Bibliographie :	75

Introduction générale

1 Introduction générale :

1.1 Contexte :

Les dernières années sont marquées par une augmentation massive des demandes de déplacement. Cette augmentation exige le développement et l'amélioration continue des moyens de transports. Pourtant, et malgré son évolution, les transports en commun restent moins compétitifs par rapport aux exigences des usagers, notamment dans les milieux ruraux. Grâce à sa disponibilité et son efficacité, la voiture individuelle reste le moyen de transport privilégié notamment dans ces zones non urbaines.

1.2 Problématique :

Dans l'optique de réaliser un service de covoiturage répondant aux besoins de déplacement et remédiant aux problèmes écologiques et économiques mentionnés auparavant, il va falloir trouver une solution formée par des affectations optimisées et un ensemble de tournées de véhicules pour servir les requêtes émises dans les meilleures conditions.

Les outils mettant en place de tels services font ainsi l'objet d'intérêt de plusieurs industriels ainsi que des chercheurs académiques.

Usant des technologies nouvelles, la mise en place d'un système plus ou moins évolué prime sur tout autre objectif au niveau des approches existantes.

Il existe des dizaines de sites internet qui proposent des services de covoiturage.

Cependant, la majorité de ces plateformes offrent un service assez limité tels que les publications et les consultations d'offres et de demandes.

En effet, l'utilisateur doit chercher « tout seul » ses partenaires et récupérer ainsi les coordonnées des covoiturer pour une éventuelle prise de contact

Chapitre I

Les applications web et java EE

I.1 Introduction :

L'objectif de ce chapitre, c'est les applications web avec le Modèle-Vue-Contrôleur (MVC), leur principe et architecture. Puis, la définition de JavaEE avec leur Composants, l'architectures, les API et services de Java EE.

I.2 Modèle-Vue-Contrôleur (MVC) :

Avant de nous lancer dans un développement, clarifions ce motif de conception (pattern design) dit Modèle-Vue-Contrôleur (MVC). Il est maintenant très répandu et accepté sans contestation comme un de ceux menant, notamment pour la réalisation de sites Web dynamiques, à une organisation satisfaisant le but recherché d'une organisation rigoureuse et logique du code.

Un des objectifs est la séparation des différentes couches constituant une application interactive, de manière à simplifier la gestion de chacune, par exemple en permettant la remise en cause indépendante de l'un des composants de l'architecture globale. Il devrait par exemple toujours être possible de revoir complètement la présentation d'un site sans toucher au code applicatif (ou métier), et, réciproquement, le code métier devrait être réalisé avec le minimum de présupposés sur la présentation. La question de l'évolutivité est elle aussi essentielle. Un logiciel doit être modifiable facilement et sans dégradation des fonctions existantes (régression). Enfin, dans tous les cas, l'organisation du code doit être suffisamment claire pour qu'il soit possible de retrouver très rapidement la partie de l'application à modifier, sans devoir ouvrir des dizaines de fichiers. C'est notamment très utile sur de gros projets impliquant plusieurs personnes : le partage des mêmes principes de développement et de structuration représente à terme un gain de temps considérable.

I.3 Principe général de MVC :

Le MVC est un motif de conception (design pattern) qui propose une solution générale au problème de la structuration d'une application. Le MVC définit des règles qui déterminent dans quelle couche de l'architecture, et dans quelle classe (orientée-objet) de cette couche, doit être intégrée une fonctionnalité spécifique. Une application conforme à ces règles est plus facile à comprendre, à gérer et à modifier. Ces règles sont issues d'un processus d'expérimentation et de mise au point de bonnes pratiques qui a abouti à une architecture standard.

I.4 Vue d'ensemble de MVC :

L'objectif global du MVC est de séparer les aspects traitement, données et présentation, et de définir les interactions entre ces trois aspects. En simplifiant, les données sont gérées par le modèle, la présentation par la vue, les traitements par des actions et l'ensemble est coordonné par les contrôleurs. **La figure 1** : Architecture MVC donne un aperçu de l'architecture obtenue, en nous plaçant d'emblée dans le cadre spécifique d'une application Web.

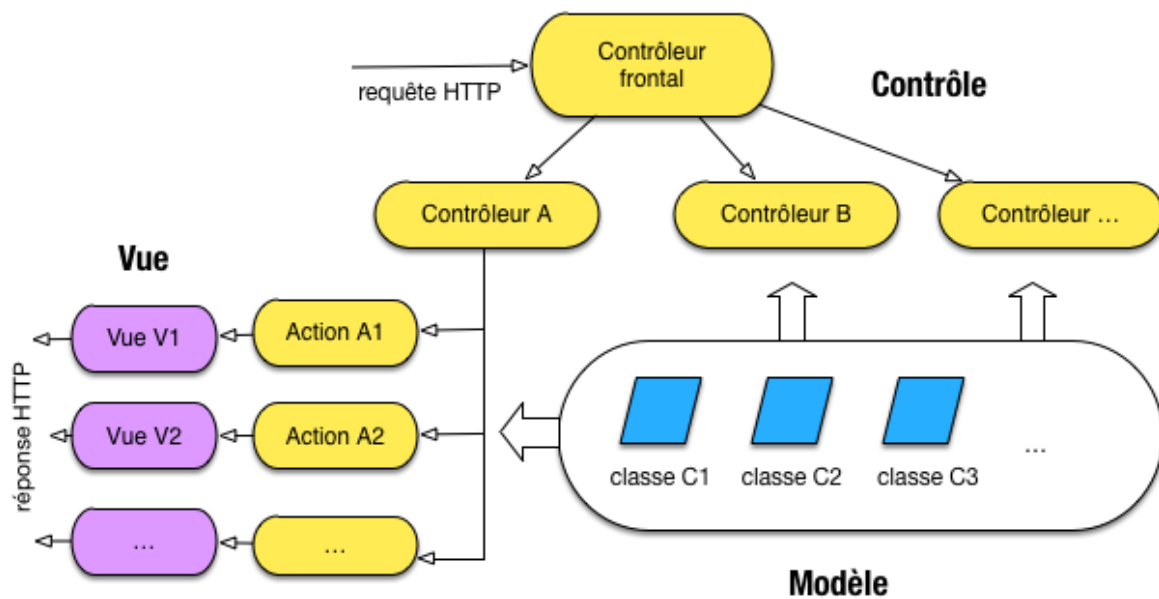


Figure 1:Architecture MVC [1]

La figure 1 : montre une application constituée de plusieurs contrôleurs, chaque contrôleur étant lui-même constitué d'un ensemble d'actions. La première caractéristique de cette organisation est donc de structurer hiérarchiquement une application. Dans les cas simples, un seul contrôleur suffit, contenant l'ensemble des actions qui constituent l'application Web.

Chaque requête HTTP est analysée par le framework qui détermine alors quel sont le contrôleur et l'action concernés. Il existe un contrôleur frontal (intégré au framework et donc transparent pour le programmeur), chargé de recevoir les requêtes HTTP, qui exécute l'action en lui passant les paramètres HTTP. Il se base notamment sur les mappings qui associent des URLs à des servlets,

Au niveau du déroulement d'une action, les deux autres composants, la vue et le modèle, entrent en jeu. Dans le schéma de la figure Architecture MVC, l'action A₁ s'adresse au modèle pour récupérer des données et éventuellement déclencher des traitements spécifiques à ces données.

L'action passe ensuite les informations à afficher à la vue qui se charge de créer la présentation. Concrètement, cette présentation est le plus souvent dans notre cas le document HTML qui constitue la réponse HTTP.

Il s'agit d'un schéma général qui peut se raffiner de plusieurs manières, et donne lieu à plusieurs variantes, notamment sur les rôles respectifs des composants. Sans entrer dans des discussions qui dépassent le cadre de ce document, voici quelques détails sur le modèle, la vue et le contrôleur.

I.4.1 Le modèle :

Le modèle implante les fonctionnalités de l'application, indépendamment des aspects interactifs. Le modèle est également responsable de la préservation de l'état d'une application entre deux requêtes HTTP, ainsi que des fonctionnalités qui s'appliquent à cet état. Toute donnée persistante doit être gérée par la couche modèle, mais des objets métiers non persistant implantant une fonctionnalité particulière (un calcul, un service) sont également dans cette couche. Le modèle gère les données de session (le panier dans un site de commerce électronique par exemple) ou les informations contenues dans la base de données (le catalogue des produits en vente, pour rester dans le même exemple). Cela comprend également les règles, contraintes et traitements qui s'appliquent à ces données, souvent désignées collectivement par l'expression logique métier de l'application.

I.4.2 La vue :

La vue est responsable de l'interface, ce qui recouvre essentiellement dans notre cas les fragments HTML qui sont assemblés pour constituer les pages du site. La vue est également responsable de la mise en forme des données (pour formater une date par exemple) et doit d'ailleurs se limiter à cette tâche. Il faut prendre garde à éviter d'y introduire des traitements complexes qui relève de la logique métier, car ces traitements ne seraient alors pas réutilisables dans un autre contexte. En principe la vue ne devrait pas accéder au modèle et obtenir ses données uniquement de l'action (mais il s'agit d'une variante possible du MVC).

La vue est souvent implantée par un moteur de templates, dont les caractéristiques, avantages et inconvénients donnent lieu à de nombreux débats. Une des difficultés est qu'il est souvent nécessaire de prendre des décisions concernant l'affichage dans la vue elle-même (par exemple,

si telle donnée a telle valeur, alors on affiche tel fragment). Ces décisions s'implantent avec des instructions impératives comme les tests ou les boucles, ce qui soulève deux problèmes majeurs :

- La vue devient lourde et ressemble à de la programmation, ce qu'on voudrait éviter ;
- Et surtout si on permet de coder dans la vue, qu'est-ce qui empêche un programmeur d'y mettre de la logique métier ? On risque de perdre la clarté du modèle MVC.

Dans le cas de Java, les outils pour gérer les vues ont beaucoup évolué, depuis les Java Server Pages (JSP) jusqu'à la Java Standard Tag Library (JSTL) que nous présenterons plus loin. Les moteurs de Template, dans l'environnement Java, s'appuient sur ces outils.

I.4.3 Contrôleurs et actions :

Le rôle des contrôleurs est de récupérer les données utilisateurs, de les filtrer et de les contrôler, de déclencher le traitement approprié (via le modèle), et finalement de déléguer la production du document de sortie à la vue. Comme nous l'avons indiqué précédemment, l'utilisation de contrôleurs a également pour effet de donner une structure hiérarchique à l'application, ce qui facilite la compréhension du code et l'accès rapide aux parties à modifier. Indirectement, la structuration logique d'une application MVC en contrôleurs et actions induit donc une organisation normalisée.

Les contrôleurs, comme leur nom l'indique, ont pour rôle essentiel de coordonner les séquences d'actions/réactions d'une application web. Ils reçoivent des requêtes, déclenchent une logique à base d'objets métiers, et choisissent la vue qui constituent la réponse à la requête.

I.5 Composants et architectures Java EE :

Les développements Java EE reposent sur un découpage en couches ou tiers, nous parlons alors d'applications multi-tiers. Trois grands tiers sont représentés :

- La couche présentation (tiers Web).
- La couche métier (tiers Métier ou tiers Business).
- La couche stockage des informations (tiers Enterprise Information System).

Les applications sont découpées en plusieurs composants réalisant des fonctionnalités spécifiques et installés sur une machine serveur ou sur plusieurs tiers distribués. Les composants Java EE sont des unités autonomes assemblées dans une application Java EE composée de classes Java et de fichiers, et communiquant avec d'autres composants. De même, le code métier écrit est indépendant de la couche de présentation, ce qui est utile pour changer cette dernière ou pour l'afficher sur divers supports (Internet, Téléphonie...).

D'après le schéma ci-après, la spécification Java EE définit les composants suivants :

- Les composants Clients ou tiers Client sont des applications clientes (logiciel installé en local ou navigateur Web ou Applets) s'exécutant sur la machine des utilisateurs.
- Les composants Web ou tiers Web sont les technologies Servlets, JavaServer Pages et Java- Server Faces.
- Les composants métier ou tiers Métier sont des composants Enterprises Java Beans (EJB) représentant la logique métier, s'exécutant sur le serveur Java EE et dialoguant avec la couche de stockage (EIS : Enterprise Information System).

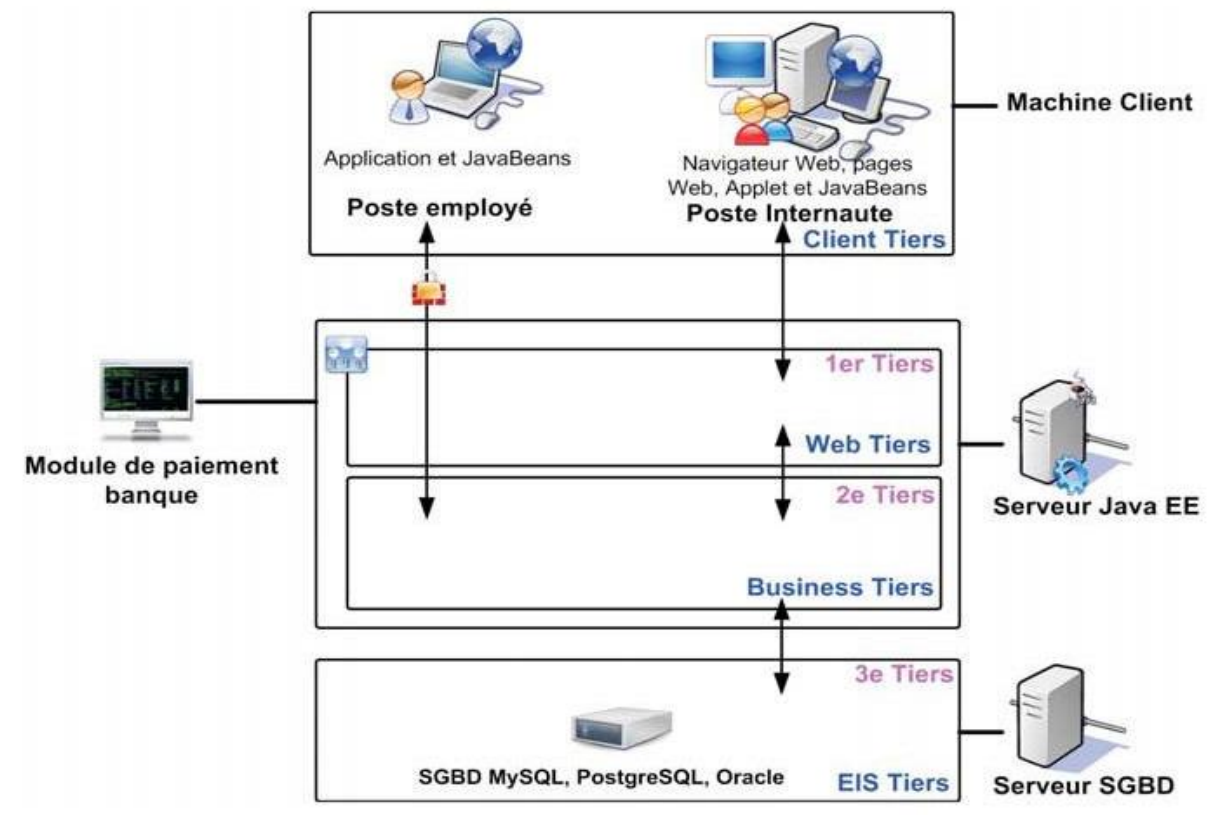


Figure 2: Architecture Java EE standard [2]

Une application Java EE multi-tiers correspond à un ensemble de couches. Souvent considérée comme étant une architecture trois tiers, celle-ci est distribuée de manière physique sur trois machines :

- La machine des clients.
- Le serveur d'applications Java EE.
- Le serveur de base de données ou de stockage.

I.5.1 Les composants clients ou tiers Client :

La plate-forme Java EE propose trois types de clients : les clients Web, les Applets et les applications clientes riches type Java SE.

I.5.1.1 Les clients Web :

Un client Web consiste en des pages Web de différents types (HTML, XHTML, XML, JavaScript ou autres) générées à partir de composants exécutés côté serveur dans un conteneur Web et capables de répondre aux requêtes HTTP en provenance du navigateur Internet. Ces programmes côté serveur sont représentés en Java EE par les Servlets, pages JSP et JSF. Ces programmes serveur ne réalisent en général pas directement les accès aux bases de données mais s'occupent de la logique applicative.

I.5.1.2 Les clients Applets :

Les Applets sont des interfaces graphiques Java SE exécutées dans un navigateur Web. Ces applications utilisent une interface graphique évoluée de type SWING et sont exécutées dans une machine virtuelle Java installée dans le navigateur. Cependant cette technique est plus contraignante à maintenir, requiert des accès et des droits pour la sécurité et nécessite un plug-in pour les navigateurs. Les clients Web sont donc préférables pour la création du tiers Client.

I.5.1.3 Les applications clientes :

Une application de type client est un logiciel riche, qui s'exécute sur la machine du client et fournit un ensemble de services aux utilisateurs par l'intermédiaire d'une interface graphique évoluée encore appelée Graphical User Interface (GUI).

Les clients et le serveur dialoguent ensemble par l'intermédiaire d'un composant standardisé nommé JavaBean ou classe Plain Old Java Object (POJO). Un composant JavaBean peut être vu comme la plus petite unité de communication entre les couches ou tiers Java EE. Les composants JavaBeans sont des objets, instances de classes POJO, composés de propriétés et de leurs accesseurs associés pour accéder aux données. Les JavaBeans s'exécutent sur le serveur Java EE et dialoguent avec les clients ou avec la base de données (tiers EIS).

Ces applications riches sont développées avec les composants SWING en général, exécutées avec Java Web Start ou Application Client Container (ACC)

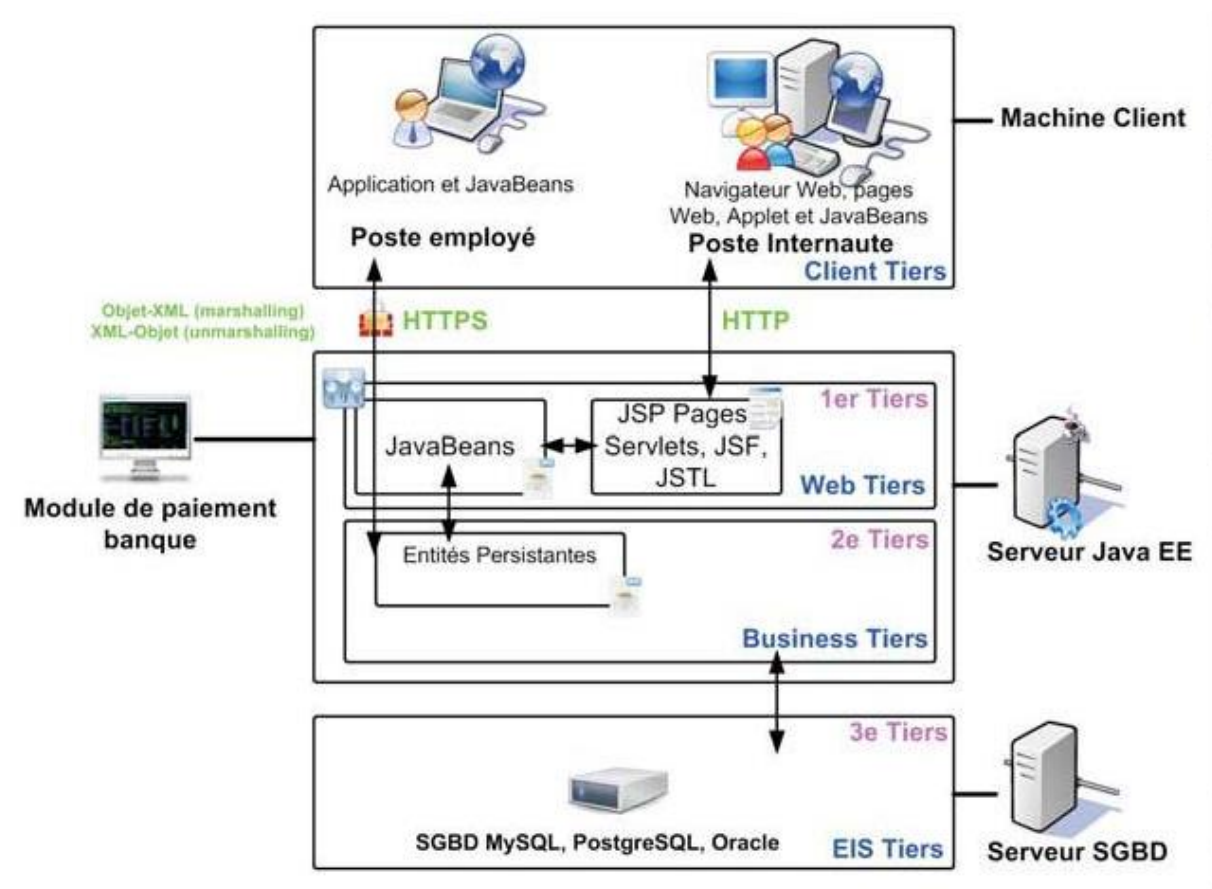


Figure 3: Communication Java EE entre les différents tiers [2]

La figure 2, montre les différents types de clients utilisés dans une plate-forme Java EE. Les clients riches de type application communiquent directement avec le tiers Métier alors que les clients de type navigateur Internet communiquent à travers le tiers Web par l'intermédiaire de pages exécutées sur le serveur et générées par des Servlets, JSP ou JSF.

I.5.2 Les composants Web ou tiers Web :

Les composants Web Java EE sont des Servlets et/ou des pages Java Server Pages et/ou des Java Server Faces. Les Servlets sont des classes Java, capables d'intercepter et de gérer les requêtes du protocole HTTP. Les pages JSP sont des documents textuels exécutés comme des Servlets apportant une solution simplifiée pour la programmation de pages Web. La technologie Java Server Faces est construite à partir de Servlets et fournit un framework de développement pour accélérer la création d'applications Web.

I.5.3 Les composants métier ou tiers Métier :

Les composants métier ou tiers Métier représentent la couche business, avec les données du système, et sont de deux types :

- Les entités beans (entitybean ou beanentity) peuvent être exécutées par un conteneur léger (pas forcément un serveur Java EE) et permettent de réaliser la persistance des données à partir des JavaBeans et de Java Persistence API (JPA) [2].
- Les Enterprise JavaBeans offrent des possibilités plus riches comme la gestion des transactions, les accès directs par clients riches ou encore la gestion automatique des sessions utilisateur, mais sont exécutés sur un conteneur lourd, c'est-à-dire compatible Java EE.

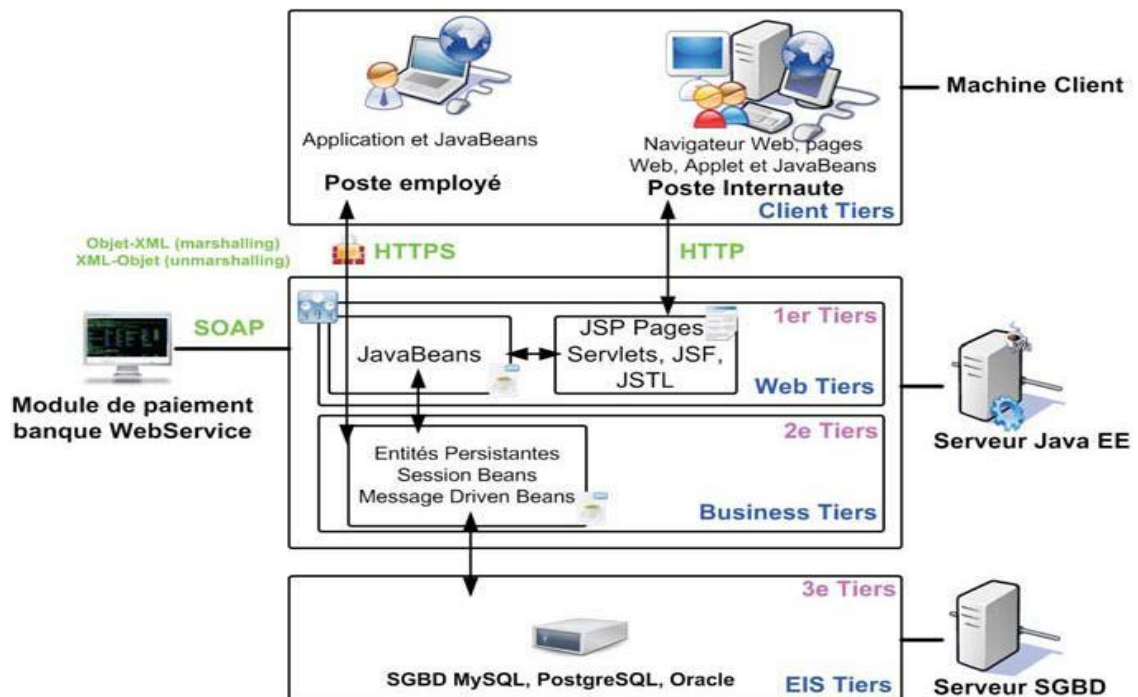


Figure 4: Les composants de Java EE

La figure 3, présente le découpage du tiers Métier avec les entités beans et les Enterprise JavaBeans. Les informations sont reçues par un client léger au travers du tiers Web et des pages Internet, ou directement depuis un client lourd. Ces informations sont ensuite envoyées sous la forme de JavaBeans, images des informations en provenance des clients et à destination du tiers Métier pour stockage. Dans le sens inverse, les Enterprise JavaBeans (EJB) lisent les données en provenance du système de stockage, réalisent des opérations si nécessaires, et sont envoyés au client.

La partie stockage des données, nommée tiers Enterprise Information System (EIS) est directement liée au tiers Métier et correspond dans la majorité des cas, aux systèmes de gestion de bases de données (Derby, MySQL, PostgreSQL, Oracle ou autres) ou à une entreprise ressource planning (ERP), un gros système transactionnel ou n'importe quel système de stockage évolué.

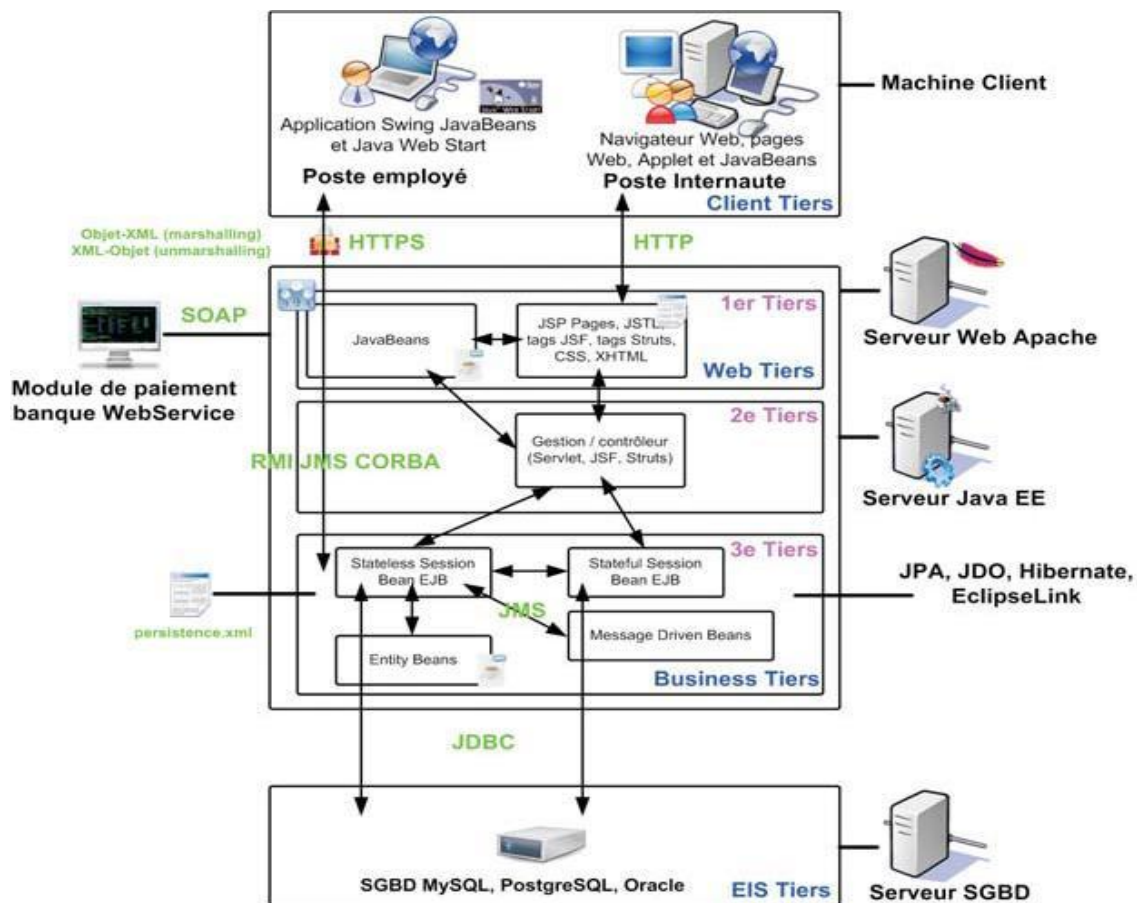


Figure 5: Architecture évoluée Java EE

La figure précédente représente une architecture optimisée d'une application Java EE. Trois types de clients sont représentés (client Web, client Applet et client Application), le serveur Web utilisé en frontal est le serveur Internet de référence : Apache. Ce serveur renvoie indirectement les résultats des pages dynamiques exécutées par le serveur d'applications Java EE, et directement les contenus statiques de type images, CSS, JavaScript ou autres.

La partie contrôleur contient la logique de traitement. Cette couche dialogue avec les différents types d'Enterprise JavaBeans (State less Bean, State ful Bean et Message Driven Beans). La couche métier est gérée par le standard JPA, permettant de paramétrer les classes ainsi que l'outil de persistance Eclipse Link, le tout configuré par le fichier persistence.xml.

I.6 Les conteneurs Java EE :

Les serveurs Java EE proposent plusieurs types de conteneurs (containers) pour chaque type de composant. Chaque conteneur a un rôle bien défini et offre un ensemble de services pour les développeurs :

- L'annuaire de nommage d'accès aux ressources : Java Naming and Directory Interface (JNDI) est une interface unifiée de gestion de nommage pour les services et l'accès à ceux-ci par des applications.

- L'injection dynamique de ressources.
- La gestion des accès aux bases de données.
- Le modèle de gestion de la sécurité.
- Le paramétrage des transactions.

Les exécutions et développements sont donc directement liés au conteneur utilisé et une application Java EE côté tiers Web ne sera pas programmée comme une application côté tiers EJB, par exemple. Les accès aux ressources nommées, les accès aux bases de données ou encore l'injection dynamique de ressources seront différents. Pour résumer, un conteneur permet de gérer le cycle de vie et la gestion des ressources.

Une application Java EE de type Web nécessite un conteneur Web pour son exécution alors qu'une application utilisant les EJB nécessite un conteneur EJB pour son exécution. Chaque conteneur propose un ensemble de services avec ses avantages et ses contraintes.

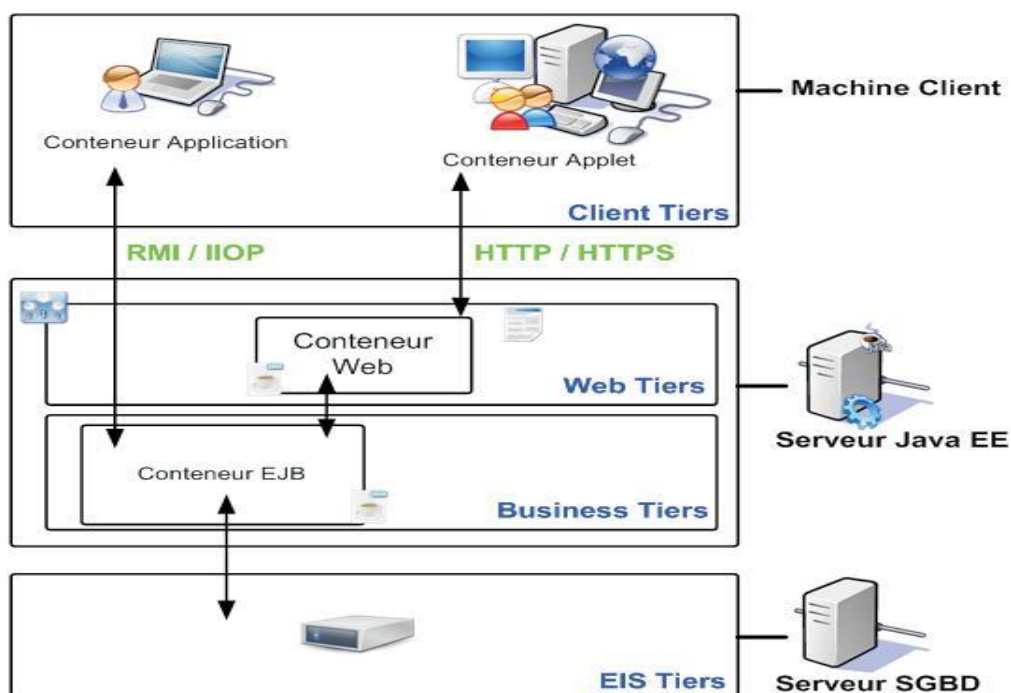


Figure 6: Serveur Java EE et conteneurs

L'architecture Java EE présentée dans la figure 6, est découpée selon plusieurs domaines ou services, appelés des conteneurs ou containers. D'après les précédents schémas, il existe en Java EE cinq types de conteneurs :

- **Java EE serveur**, logiciel fournisseur du conteneur Web et EJB.
- **Web Container** ou **conteneur Web**, capable d'exécuter des pages JSP, Servlets et programmes JSF (implémentation JSF nécessaire). Ces composants s'exécutent avec le conteneur serveur Java EE. Ce conteneur supporte les protocoles HTTP et HTTPS et dialogue avec différents types de clients (navigateur, application).
 - **EJB Container** ou **conteneur EJB**, responsable de la couche métier et de l'exécution des EJB. Ce conteneur fournit le mécanisme de gestion des EJB, la gestion de leur cycle de vie, les transactions, le service de nommage JNDI et la possibilité des déclenchements asynchrones.
 - **Applet Container** ou **conteneur Applet**, responsable de l'environnement d'exécution des Applets Java. Ce type d'application consiste en un navigateur Web et le plug-in Java, lancés en même temps sur le poste client. Ces programmes sont lancés dans un « bac à sable » aussi nommé « sandbox » afin de gérer la sécurité des accès sur le poste client.
 - **Application Client Container (ACC)** ou **conteneur des applications clientes**, permet l'exécution d'applications Java SE. L'application de type client et son conteneur sont lancés sur le client en simultanément. Le conteneur d'applications client (ACC) propose une bibliothèque de classes, un système de gestion de sécurité et le service de nommage JNDI pour les applications

Java SE. La communication directe entre le conteneur client et le conteneur EJB est réalisée avec les technologies RMI-IIOP, alors que la communication avec le conteneur Web est réalisée avec le protocole HTTP pour les services Web.

I.7 API et services Java EE :

La plate-forme Java EE est la plus importante proposée par Oracle et regroupe les services de la plate-forme Java SE. Les services et API Java EE sont donc associés à une plate-forme et un ou plusieurs conteneurs. Les conteneurs étudiés précédemment proposent plusieurs services pour le développement d'applications. En tant que développeur, nous pouvons utiliser la totalité des services ou bien au contraire sélectionner uniquement les techniques nécessaires à la réalisation du projet. Chaque service s'exécute dans un des cinq conteneurs étudiés précédemment et offre d'autres méthodes en fonction de ce conteneur.

I.7.1 Java Servlet (Technologie Java EE - Conteneur Web) :

Ce service Java EE s'exécute dans le conteneur Web et propose un ensemble de classes de gestion des échanges HTTP de type requête-réponse. Avec la nouvelle version de Java EE, la technologie Servlet 3.0 facilite la configuration et supporte les annotations ou encore les traitements asynchrones.

I.7.2 Java Server Pages (JSP) (Technologie Java EE - Conteneur Web) :

La technologie Java Server Pages (JSP) s'exécute dans le conteneur Web et utilise le mécanisme de génération de Servlets à partir de documents textuels. Les JSP 2.2 contiennent des données statiques de type HTML, XHTML, JavaScript ou autres, et des balises ou tags JSP nommés JSTL correspondant à du code Java.

I.7.3 Java Server Pages Standard Tag Library (JSTL) (Technologie Java EE - Conteneur Web) :

La technologie Java Server Pages Tag Library (JSTL) est encapsulée dans des pages JSP et propose une librairie de balises XHTML standardisées. Ces balises proposent de gérer les conditions et itérations, l'internationalisation, les affichages HTML ou encore l'accès aux bases de données.

I.7.4 Java Server Faces (JSF) (Technologie Java EE - Conteneur Web) :

La technologie JavaServer Faces s'exécute dans le conteneur Web et correspond à la réponse du consortium Java EE en matière de framework de développement côté serveur. JSF 2.0 est un framework de développement de projets Web proposant la construction rapide de code serveur, la validation des entrées, la gestion des modèles objets, la configuration de la navigation, les conversions de données, l'utilisation d'annotations, les balises et technologies d'affichages de données pour la partie Vue, le support de la technologie Ajax en natif et bien d'autres fonctionnalités. JSF est orienté composant et repose sur un cycle de vie proche du développement d'applications Java SE (écouteurs, évènements...).

I.7.5. Java Data Base Connectivity (JDBC) (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application) :

La technologie Java Data Base Connectivity (JDBC) permet de gérer les commandes SQL et les dialogues avec les différents SGBD relationnels. JDBC est utilisée lorsque nous devons accéder à une base de données dans une Servlet, page JSP, page JSF ou autres. L'API JDBC propose un ensemble de classes et interfaces pour se connecter aux différents SGBD du marché mais également les paquetages de manipulation des données. JDBC est une technologie sous-jacente à JPA.

I.7.6. Java Persistence API (JPA) (Technologie Java SE - Conteneur Web, Conteneur EJB et Conteneur Application) :

Java Persistence API (JPA) est un standard Java utilisé pour la persistance des données. Ce mécanisme de persistance utilise le principe de mapping objet/relationnel et relationnel/objet afin de permettre de stocker les objets dans la base de données et inversement de pouvoir lire les données relationnelles et les transformer en objets.

L'API Java Persistence 2.0 propose les services suivants :

- La gestion de la persistance.
- Un langage de requêtage évolué : Java Persistence Query Language (JPQL).
- Un mécanisme de mapping objet/relationnel ORM à partir de métadonnées (fichiers XML ou annotations).

- Une API à typage fort pour le requêtage.

I.7.7. Java Transaction API (JTA) (Technologie Java EE - Conteneur Web, Conteneur EJB :

Java Transaction API (JTA) fournit un standard de gestion des transactions pour une architecture Java EE et propose par défaut le mode autocommit pour la gestion des transactions, mais offre également les mécanismes de commit et rollback pour une optimisation des transactions. Cependant, cette API est rarement utilisée directement par le développeur, mais plutôt en association avec d'autres API.

I.7.8 Enterprise JavaBeans (EJB) (Technologie Java EE - Conteneur EJB) :

Les composants Enterprise JavaBeans s'exécutent dans le conteneur dédié EJB et correspondent à la couche ou tiers Métier de l'application. Ces EJB 3.X sont des classes composées de champs et de méthodes d'accès. Java EE propose deux familles d'EJB : les sessions beans et message-driven beans.

La session beans représentent les conversations entre le client et le serveur, qu'elles soient avec ou sans état. Un Message-Driven Bean (MDB) mélange une session bean et un écouteur de message permettant ainsi la réception asynchrone de messages. Ces services utilisent Java Message Service (JMS) pour leur fonctionnement. La nouvelle version de Java EE offre la possibilité d'empaqueter les EJB directement dans des archives WAR (Web ARchive), l'utilisation du Design Pattern Singleton dans les sessions beans et la possibilité d'utiliser des EJB légers avec un serveur Java ne proposant pas le conteneur EJB.

I.7.9 Java Message Service API (JMS) (Technologie Java EE – Conteneur Web, Conteneur EJB et Conteneur Application) :

Java Message Service API (JMS) est le standard permettant aux applications Java EE de créer, d'envoyer, de recevoir et de traiter les messages asynchrones en mode point à point ou multi destinations.

I.7.10 JavaMail (Technologie Java SE - Conteneur Web, Conteneur EJB) :

Les applications Java EE utilisent l'API JavaMail pour l'envoi d'e-mails. L'API JavaMail fournit pour cela un ensemble de classes de gestion.

I.7.11 Java Authorization Service Provider Contract for Containers (JACC) (Technologie Java EE - Conteneur Web, Conteneur EJB) :

La sécurité des services est regroupée sous le terme Java Authentication and Authorization Service (JAAS) et permet de gérer l'authentification et les accès clients. L'API Java Authorization Service Provider Contract for Containers (JACC) définit un contrat entre une application Java EE et un fournisseur de services. La spécification Java ACC définit les classes du paquetage **java.security.Permission** pour la gestion des autorisations.

I.7.12 Java Authentication Service Provider Interface for Containers (JASPIC) (Technologie Java EE - Conteneur Web, Conteneur EJB) :

L'API Java Authentication Service Provider Interface for Containers (JASPIC) propose une interface de service d'authentification pour les dialogues entre le client et le serveur.

I.7.13 Java Authentication and Authorization Service (JAAS) (Technologie Java EE - Conteneur Web, Conteneur EJB) :

La technologie Java Authentication and Authorization Service (JAAS) fournit les mécanismes d'authentification et d'autorisation pour un groupe ou utilisateur spécifique lançant l'application. JAAS est une version du standard PluggableAuthentication Module (PAM).

I.7.14 Java API for XML Registries (JAXR) (Technologie Java EE - Conteneur Web, Conteneur EJB, Conteneur Application) :

Java API for XML Registries (JAXR) permet un accès au tiers Métier à travers le Web.

I.7.15 Java API for XML Processing (JAXP) (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application) :

La plate-forme Java permet l'utilisation de configuration à base de fichiers au format XML. Les services de la bibliothèque Java API for XML Processing (JAXP) permettent le support des

documents XML en utilisant une bibliothèque pour le parsing XML : Document Object Model (DOM) et Simple API for XML (SAX) ainsi qu'un mécanisme de feuilles de style : Extensible Stylesheet Language Transformations (XSLT). JAXP permet aux différentes applications Java de manipuler des documents XML, pour la description et le déploiement des applications, indépendamment du processeur XML utilisé.

I.7.16 Java Architecture for XML Binding (JAXB) (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application) :

L'API Java Architecture for XML Binding (JAXB) fournit un mécanisme simple de transformation d'objet en document XML et inversement, pour une utilisation en Java. JAXB peut également être utilisée avec JAX-WS ou JAX-RS et permet l'échange de messages avec l'utilisation des services Web.

I.7.17 Java API for XML Web Services (JAX-WS) (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application) :

L'API Java API for XML Web Services (JAX-WS) fournit un support des services Web et utilise pour cela l'API JAXB (pour la transformation Objet/XML et XML/Objet) et l'API SOAP with Attachments API pour Java (SAA) [4]. La bibliothèque JAX-WS remplace Java API for XML-based RPC (JAX-RPC) et propose un support des services Web en utilisant le protocole SOAP. Les messages supportent les accès JNDI des composants et le principe de sérialisation/désérialisation.

I.7.18 Java API for RESTful Web Services (JAX-RS) (Technologie Java EE - Conteneur Web, Conteneur EJB) :

L'API Java for RESTful Web Services (JAX-RS) permet le développement Web de services utilisant l'architecture REpresentational State Transfer (REST). Une application JAX-RS est une application Web composée de classes empaquetées dans une archive WAR fournissant un ensemble de services. JAX-RS permet d'exposer des classes POJO sous forme de ressources REST manipulables via le protocole HTTP.

I.7.19 Java Naming and Directory Interface (JNDI) (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application) :

L'API Java Naming and Directory Interface (JNDI) fournit un mécanisme de nommage de type annuaire pour l'accès aux ressources. Ces ressources peuvent être de différents types mais le but est d'associer les objets à un nom (bind) et de retrouver ces objets (lookup) dans un annuaire de nommage semblable à LDAP, DNS ou NIS. Nous pouvons ainsi nommer une source de données de type base de données, un Enterprise JavaBean, une session Java Mail ou autres. JNDI permet une utilisation et un accès depuis une application de type client, un EJB ou un composant Web.

I.7.20 JavaBeans Activation Framework (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application) :

JavaBeans Activation Framework (JAF) est utilisé par l'API Java Mail pour la gestion des types MIME. JAF permet de déterminer le type d'un ensemble de données encapsulées dans l'e-mail. L'API permet de déterminer le type et de créer un JavaBean adapté en fonction de celui-ci.

La plate-forme Java EE propose également un outil de gestion (management) Java Management Extensions (JMX) pour le contrôle des conteneurs et du serveur. Elle propose enfin un outil de gestion du déploiement afin de standardiser le mécanisme de déploiement entre les produits Java EE et les applications.

Le tableau présenté ci-dessous récapitule les spécifications Java EE 6 définies dans la JSR-316 et contenant 28 spécifications.

Spécification	Version	JSR
Java EE	6	316
Tiers Web		
JSF	2	314
JSP	2,2	245
JSTL	1,2	52
Servlet	3	315
Expression Language (EL)	1,2	245
Tiers Enterprise JavaBean		
EJB	3,1	318

Tableau 1:les spécifications Java EE [4]

I.8. Conclusion :

Dans ce chapitre, nous avons vu le Modèle-Vue-Contrôleur (MVC), leur principe et architecture. Puis, Les Composants et l'architectures de Java EE. En fin de parler sur les API et services Java EE.

Chapitre II

*Etude Critique des différentes solutions
proposées pour le covoiturage*

II.1 Introduction :

Le covoiturage est l'utilisation conjointe et organisée d'une voiture automobile, par un conducteur non professionnel et un ou plusieurs tiers passagers, dans le but d'effectuer un trajet commun. L'objectif de ce chapitre on a donné le principe et les formes de covoiturage avec les deux cas statique et dynamique.

II.2 Définition de covoiturage :

Le covoiturage est l'utilisation conjointe et organisée (à la différence de l'auto-stop) d'une voiture automobile, par un conducteur non professionnel et un ou plusieurs tiers passagers, dans le but d'effectuer un trajet commun.

Il procure des avantages individuels (partager les dépenses de carburant et de maintenance, agrémenter les voyages, développer le lien social) et collectifs (augmenter le taux de remplissage des véhicules, diminuer les embouteillages et la pollution) [5].

En France, le covoiturage est indemnisé dans la limite du barème fiscal kilométrique [6] [7], ce qui permet au plus grand nombre de se déplacer. Le conducteur ne doit donc pas faire de bénéfices (l'argent qu'il reçoit doit correspondre au partage des frais liés au trajet) [8]. Il dépend de la motivation des particuliers et repose en grande partie sur un principe collaboratif d'auto-organisation mais peut être encouragé, coorganisé ou aidé par des collectivités ou entreprises qui y trouvent divers bénéfices. Le mot ne fait son apparition dans la loi qu'à partir de 2015[9].

II.3 Le Principe de covoiturage :

À la différence du taxi où le passager choisit la destination, en covoiturage, c'est le conducteur qui offre de partager son véhicule, éventuellement gratuitement [10] et fixe le trajet.

Un conducteur propose aux passagers de les transporter dans sa voiture pour un trajet (ou une portion de trajet) qu'il doit lui-même effectuer, et donc à la date et à l'heure qu'il a décidée. Généralement, le lieu de départ, déterminé à l'avance, est le même pour tout le monde. À l'arrivée, le conducteur dépose les passagers là où lui-même s'arrête, ou bien à l'endroit que chacun souhaite, en évitant de faire un grand détour. Il peut alors laisser ses passagers par exemple à proximité d'un transport en commun ou bien là où un membre de la famille ou un ami pourront les prendre en charge.

Le conducteur peut demander à ce qu'un des passagers le remplace au volant afin qu'il puisse se reposer un peu.

Le partage des frais est laissé à l'appréciation du conducteur. La formule la plus classique consiste à diviser le coût du carburant et des péages éventuels par le nombre de personnes. Les frais généraux tels que ceux d'entretien ou d'assurance peuvent être inclus dans le calcul du coût du trajet.

Le conducteur est tenu de ne pas faire de bénéfice, en particulier pour ne pas enfreindre la loi [11]. Certains demandent cependant une participation forfaitaire quel que soit le nombre de passagers. Il existe néanmoins aussi des sites web et applis mobiles de covoiturage totalement gratuits, sans contribution financière. Enfin, en 2016, au Luxembourg, est apparue une appli mobile de covoiturage originale, car basée sur l'échange mutuel de service entre un chauffeur, piéton mais apte à conduire, et un possesseur de véhicule qui ne veut ou ne peut pas conduire à l'instant du trajet.

II.4 Les formes du covoiturage :

II.4.1 Le covoiturage spontané :

Le covoiturage spontané fait référence à toutes les personnes qui covoiturent sans passer par une structure de mise en relation. L'auto-stop est la forme de covoiturage spontané la plus simple : elle ne demande aucune organisation d'une structure extérieure et ne nécessite normalement pas la participation aux frais de transport. C'est une forme de transport où l'individu faisant de l'auto-stop essaie de



voyager gratuitement, en prenant place à bords d'un véhicule qui manifeste l'envie d'accueillir un inconnu. Le lever de pouce est internationalement reconnu comme le désir d'être pris en auto-stop, un écriteau (bien souvent un morceau de carton) peut venir compléter le dispositif.

L'autre forme de covoiturage spontané est le covoiturage informel ; il s'agit de covoitureurs indépendants de tout système, s'étant entendu mutuellement sur les modalités de déplacement (horaire, date, frais de transport...). Ce covoiturage informel correspond à des regroupements volontaires de personnes, qu'ils soient amis, collègues ou voisins, qui décident de s'organiser pour réaliser un trajet.

Le covoiturage spontané est à distinguer de l'accompagnement de personnes. En effet, ce dernier nécessite la présence d'un chauffeur, c'est-à-dire d'une personne qui n'a pas de trajet propre à lui et qui est obligée de faire un détour significatif pour déposer le passager.

Cette forme de covoiturage se fait naturellement entre deux ou plusieurs individus qui désirent faire un voyage commun ensemble. Leurs trajets ne sont pas comptabilisés, ils ne sont donc pas quantifiables. Il est difficile de donner une estimation du covoiturage informel.

II.4.2 Le covoiturage organisé :

Le covoiturage peut aussi être organisé ; c'est-à-dire que pour trouver un covoitureur potentiel, il faut passer par une entité intermédiaire de mise en relation. Cette entité peut se décliner sous différentes formes :

- **centrale de mobilité** : structure qui centralise toutes les informations concernant l'offre en transport. Elle apporte une information multimodale et un conseil en mobilité. À l'échelle d'un territoire, elle permet d'optimiser le potentiel de transport alternatif existant en promouvant les réseaux « TC » et « modes doux », et d'autres initiatives publiques ou privées comme le covoiturage ou l'auto-partage. [12]

- **associations** : étant donné son caractère associatif, elle n'a pas de but lucratif.

L'association est donc un cadre privilégié pour encourager le covoiturage ; et en ce sens, elle est souvent appuyée par l'Ademe et les collectivités pour leurs financements.

- **collectivités** : certaines collectivités, notamment les départements, mettent à la disposition de leurs administrés un outil Internet de mise en relation. Cette démarche est tout à fait volontaire et reflète l'intérêt que porte la collectivité aux nouvelles formes de mobilité.

- **entreprises / administrations** : à travers la réalisation d'un Plan de Déplacements Entreprises / Administrations, l'entreprise peut jouer un rôle important. En effet, ce cadre est propice au développement du covoiturage et de plus en plus de structures s'intéressent aux déplacements de leurs salariés dans leurs trajets quotidiens.

- **particuliers** : certains particuliers se sont lancés dans la réalisation d'un site Internet de covoiturage. Souvent, l'initiative de développer un site de covoiturage est privée mais dès lors que l'activité devient plus importante, le particulier devient souvent association.

Ce panel d'acteurs propose une gamme de services plus ou moins complète selon la structure d'organisation, son financement et ses connaissances de l'outil-support (Internet majoritairement et le téléphone). Les services proposés passent de la simple mise en ligne de l'annonce de covoiturage et des coordonnées de la personne à un véritable suivi de l'itinéraire grâce à un outil cartographique.

La présente étude s'est principalement axée sur les sites Internet de covoiturage. En effet, c'est l'outil-support le plus souvent utilisé. Même les structures proposant un service de covoiturage par téléphone ont créé un site Internet.

II.5 Les sites de covoiturage grand-public

Les sites de covoiturage grand-public sont accessibles à tous sous réserve d'avoir accès à un ordinateur et à une connexion Internet. Le fonctionnement est simple : il suffit pour cela de s'inscrire sur un site de covoiturage et de déposer/consulter une offre ou une demande de trajet (selon qu'on soit conducteur ou passager). Lors du dépôt d'une annonce, plusieurs renseignements sont nécessaires comme l'origine et la destination, la date du voyage ou encore la fréquence du trajet. Il ne reste plus, ensuite, qu'à contacter la personne pour fixer les modalités du trajet. [13]

II.6 Les systèmes opérationnels du covoiturage :

II.6.1 Le covoiturage statique :

La majorité des systèmes existants appartiennent à cette catégorie et se présentent souvent sous forme de systèmes non automatisés, accessibles via le web, intégrant tout au plus les fonctionnalités de gestion des offres et des demandes de covoiturage. Le fonctionnement de tels supports en est ainsi simplifié ; elle nécessite une inscription préalable pour pouvoir avoir accès aux différentes fonctionnalités offertes par le biais de ces supports.

La Figure 7 : montre la croissance spectaculaire du nombre de sites internet dédiés au covoiturage statique depuis 1990.

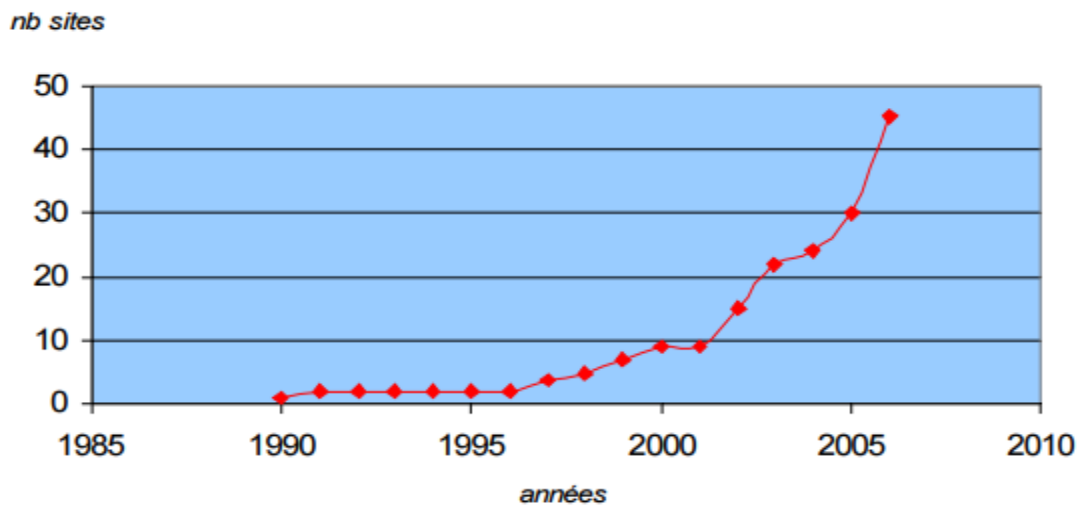


Figure 7: Croissance du nombre de sites de covoiturage Depuis 1990

II.6.1.1 Exemple du site statique :

i. *BlaBlaCar* :

BlaBlaCar, c'est quoi ?



Le covoiturage **BlaBlaCar**, c'est le moyen de transport du futur.

BlaBlaCar rassemble la plus large communauté de covoiturage longue distance au monde. La plateforme met en relation des conducteurs voyageant avec des places libres et des passagers se rendant dans la même direction. Ils partagent un trajet et les frais qui y sont liés. Avec l'intégration de bus et du covoiturage courte-distance à sa plateforme, BlaBlaCar souhaite devenir la place de marché de référence de la mobilité partagée sur la route. Pratique, économique, conviviale et responsable, partout où il y a une route, il y a un BlaBlaCar. [14].

Comment ça marche :

➤ **Recherchez votre trajet :**

Précisez simplement votre destination, votre point de départ et vos horaires. Puis choisissez un covoiturage ou un bus qui vous convient !! Si vous avez besoin de plus de renseignements, vous pouvez écrire aux conducteurs avant de réserver. [14].

➤ Réservez et payez en ligne :

Cliquez sur "Réserver" et payez votre place. Dès que ce sera fait, vous recevrez toutes les informations nécessaires. [14].

➤ Voyagez :

Bon voyage et n'oubliez pas de laisser un avis ! En écrire un multiplier vos chances d'en recevoir un à votre tour [14].

QUELQUES CHIFFRES

- 90 millions de membres
- 22 pays : France, Espagne, Royaume-Uni, Italie, Pologne, Allemagne, Portugal, Belgique, Pays-Bas, Luxembourg, Russie, Ukraine, Turquie, Inde, Mexique, Serbie, Roumanie, Croatie, Hongrie, Brésil, Slovaquie, République Tchèque.
- 25 millions de voyageurs par trimestre
- 3,9 personnes par véhicule : c'est le taux d'occupation moyen sur BlaBlaCar, contre 1,9 personne par véhicule sans BlaBlaCar
- 1,6 million de tonnes de CO2 économisées chaque année – soit l'équivalent des émissions générées par les transports à Paris en 1 an ! [14]

Les fonctionnements du site BlaBlaCar :

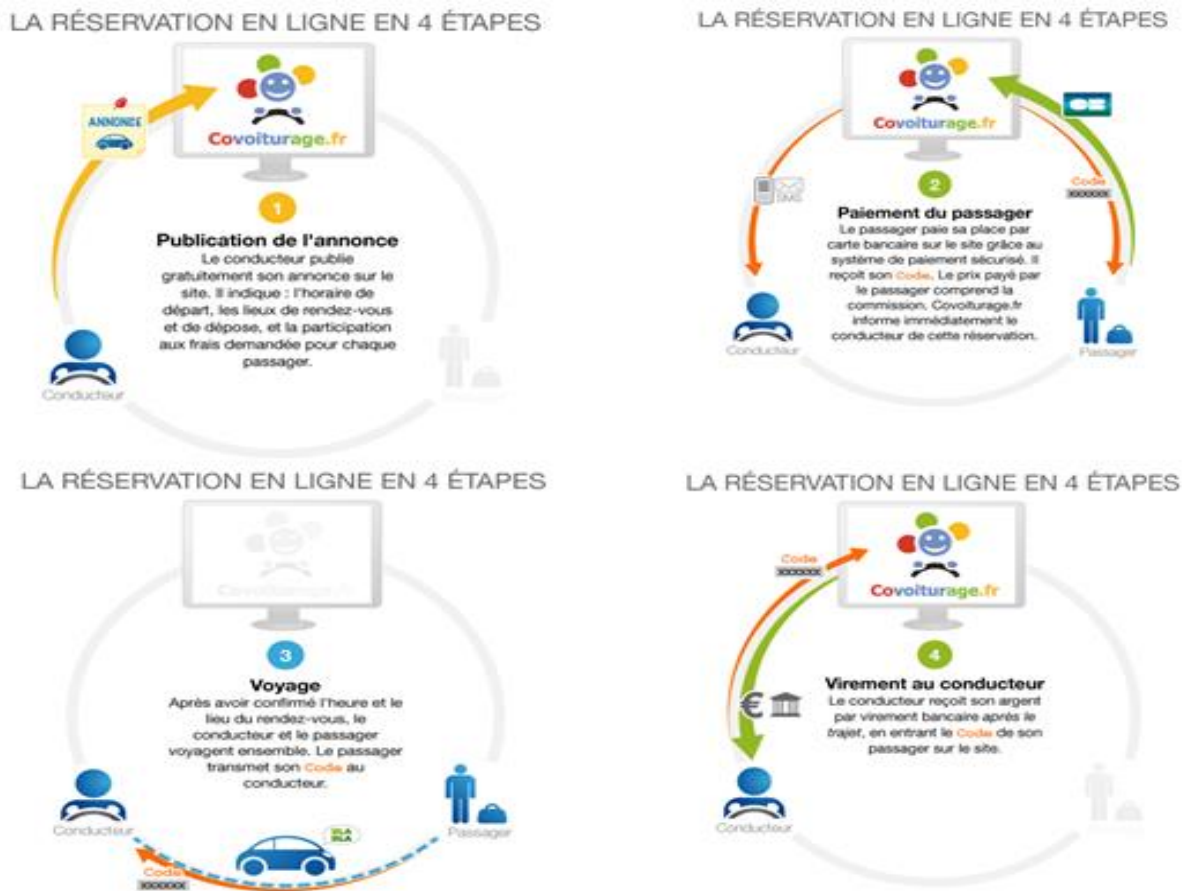


Figure 8:Les fonctionnements du site BlaBlaCar [14].

ii. *Uber :*



U B E R

Uber, anciennement UberCab[15], est une entreprise technologique américaine qui développe et exploite des applications mobiles de mise en contact d'utilisateurs avec des conducteurs réalisant des services de transport.

L'entreprise est basée dans la ville californienne de San Francisco, aux États-Unis. En 2015, elle est valorisée à 50 milliards de dollars [16] et ses applications sont commercialisées dans plus de 310 villes dans le monde [17]. Elle a été fondée par Garrett Camp, Idir Hedjem et Travis Kalanick.

Organisant une activité commerciale régulière de vente de service comme s'il s'agissait d'une activité occasionnelle, la société fait l'objet de nombreuses polémiques, en raison de débats sur ses pratiques, où la libre concurrence qu'elle défend se heurte à des accusations de concurrence

déloyale et de travail dissimulé de la part des pouvoirs et des professionnels qui se réclament du respect de la réglementation sociale, fiscale et administrative en vigueur. Engagée dans un développement rapide, à la fois géographique et via la gamme des services proposés, la société Uber a vu son activité interdite dans plusieurs villes ou pays du monde.

L'extension du concept à plusieurs secteurs économiques est à l'origine du principe dit de l'upérisation.

Pourquoi Uber s'appelle Uber ?

Le nom de la plateforme de mise en relation entre clients et VTC est à l'origine du terme « uberisation » qu'on entend désormais partout. Mais d'où vient le mot « Uber » ?

« **Uberisation** » : ce terme, très voire trop populaire aujourd'hui, décrit tout un nouveau modèle économique. Ce sont des services qui mettent en relation instantanée les clients avec les professionnels, à travers le développement technologique. Ce mot provient de la marque Uber. À son arrivée, cette application mobile a révolutionné les modes de transport. En un glissement du pouce sur un Smartphone, il est possible de faire appel à un chauffeur VTC (véhicule de transport avec chauffeur), où que l'on soit, avec le prix de la course connu à l'avance. [17]

iii. Yasiir :



YASSIR est un service de transport innovant qu'on peut utiliser avec son Smartphone n'importe où et n'importe quand. Il permet à chacun de réserver un chauffeur et de se déplacer en toute sécurité.

Yassir a été créé autour de la conviction que lorsque les gens sont bien traités, ils fournissent un meilleur service. Conducteur heureux, passager heureux. [18]

La simplicité d'usage :

Il vous suffit d'un clic pour obtenir un chauffeur à l'endroit où trouvez ! [18]

La disponibilité :

Partout et à tout moment. Trouvez le chauffeur en quelques secondes avec Yassir. [18]

Au meilleur prix :

Des tarifs ajustés pour garantir un service adapté aux besoins. [18]

II.6.2 Le covoiturage dynamique :

L'idée du covoiturage dynamique est de fournir une offre en quasi temps réel à « l'utilisateur covoitureur ». La personne souhaitant effectuer un itinéraire en covoiturage contacte le service quelques minutes avant son départ. Le service va alors chercher le conducteur adéquat qui est en mesure d'offrir le covoiturage souhaité sur l'itinéraire demandé.

Avec une quantité suffisante de participants, le fonctionnement est souple et permet une bonne qualité de service : la probabilité de trouver le covoitureur adéquat est élevée. Ce système est complémentaire aux transports publics – réguliers et à la demande – et autres solutions alternatives – auto partage, vélo stations, etc.

Les technologies de l'information et de la communication – téléphones portables, Personale Digital Assistant (PDA), internet, Global Positionné System (GPS), etc. – fortement développées à l'heure actuelle, sont des outils pertinents pour exploiter le covoiturage dynamique. Il s'agit de mettre en relation rapidement et de façon appropriée les personnes, via l'utilisation de ces technologies.

II.6.3 Exemple du site dynamique :**II.6.3.1 Avego:**

Avego est une application, mise en place par la société Mapflow, disponible à partir d'un Iphone équipé de GPS. Elle permet de faire du covoiturage dynamique. Avego développe également des solutions pour les services de navettes et de TAD. [19]

Le conducteur rentre sa destination et une fois que cela est connu les sièges vides sont offerts à des potentiels passagers. Si un passager souhaite un trajet à un certain moment, le système sélectionne le conducteur le plus adapté et propose à ce dernier le détour à effectuer. Si le conducteur accepte, une voix permet de guider le conducteur jusqu'à « l'arrêt Avego » approprié où conducteur et passager peuvent se retrouver. Sur l'Iphone, le conducteur peut évaluer entre 1 et 5 son expérience avec le passager en question. [19]

Avego gère automatiquement le partage des frais entre les covoitureurs.

II.6.3.2 Comuto :



Comuto, société d'informatique et de conseil en mobilité durable a créé covoiturage. [20]

Le site de Comuto propose : [20]

- Un service de covoiturage sur téléphone mobile (<http://mobile.covoiturage.fr>) ;
- Des trajets en coordonnées GPS.
- Une interface cartographique via Gmaps .
- Une approche communautaire Web 2.0 du covoiturage.
- Un service de mise en relation confidentielle par téléphone.

Quelques chiffres depuis l'été 2008.

- 65 000 nouveaux trajets par mois.
- 15 000 nouveaux utilisateurs par mois.
- 230 000 euros économisés par mois.
- 782 tonnes de CO2 économisées par mois.

II.7 Les Avantages de covoiturage :

II.7.1 L'aspect économique :

La valeur des sièges vides des quelques 500 millions de voitures privées qui arpentent les routes de la planète chaque année est estimée à 500 milliards de dollars [21]. Dans un contexte économique difficile, la principale raison avancée pour pratiquer le covoiturage est la dimension financière.

L'enquête établie par la MAIF à montre que 94 % des personnes pratiquant le covoiturage, le font pour des raisons financières [22]. En effet, en moyenne, une voiture coûte 6000 euros par an, ce qui représente une somme 60 fois plus importante que celle du vélo et 20 fois plus qu'un

abonnement en transport en commun. Le covoiturage permet le partage de l'ensemble des frais entre les voyageurs : essence, péage, assurance, entretien de la voiture, etc. Du coup, le covoiturage permet, notamment aux salariés, de diviser leur frais de transport travail - domicile par 2, voire par 3. Si on considère que 3 personnes covoiturent sur un trajet de 30km, l'économie réalisée est de 3105 euros par an (Figure 1.8).



Figure 9: Economies annuelles réalisées sur un trajet quotidien domicile-travail [23]

Figure 9 : Economies annuelles réalisées sur un trajet quotidien domicile-travail [23]

Dans le cas des longs trajets, et grâce au partage des frais d'essence et de péage, le Covoiturage permet aux automobilistes d'économiser par exemple 114 euros sur un Paris

- Bordeaux en accueillant 3 passagers à bord (Figure 10).

	Avec 6 centimes de réduction / Litre	Avec 1 passager en covoiturage	Avec 3 passagers en covoiturage
Paris – Lille (18L - 222 km)	1,08 € d'économie	15 € d'économie	45 € d'économie
Paris – Rennes (28L - 350 km)	1,68 € d'économie	22 € d'économie	66 € d'économie
Paris – Bordeaux (47L - 583 km)	2,82 € d'économie	38 € d'économie	114 € d'économie
Nantes – Montpellier (66L - 823 km)	3,96 € d'économie	55 € d'économie	165 € d'économie

Figure 10:: Economies pour les automobilistes [24]

En plus de l'économie pour le covoitureur, le covoiturage permet de réaliser une économie importante pour la collectivité. En effet, les infrastructures routières représentent une part conséquente du budget des collectivités. Développer le covoiturage, c'est réduire la pression sur les installations routières.

II.7.2 Un geste pour l'écologique :

Selon une étude de l'IEA (International Energy Agency), le transport routier est responsable de 18% des émissions de CO₂. Entre 6.500 et 9.500 personnes meurent tous Les ans à cause de la pollution automobile, selon l'Agence Française de Sécurité Sanitaire Environnementale (AF SSE) [22]. En 2011, 50 millions de tonnes équivalent pétrole ont été consommées pour le transport routier, selon le ministère de l'écologie et du développement durable. Cette consommation a provoqué la production de 132,5millions de tonnes équivalent CO₂ de gaz `à effet de serre. Les véhicules particuliers représentent plus de 50% de cette pollution. Et même si les véhicules polluent de moins en moins (134g/km en 2009 contre 124g/km aujourd'hui) de grandes économies de pollution pourraient être faites : les millions de véhicules qui circulent en France contiennent en moyenne 1,1 passager. Plus d'un trajet sur deux est réalisé pour une distance de moins de 10 km. [25]

Selon une enquête réalisée dans la ville de Téhéran, capitale de l'Iran, S. Seyedabrishami a montré dans ses travaux que le choix de covoiturage diminuerait environ 780000 trajets de véhicules par jour et réduirait la consommation annuelle de carburant de 336 530 000 litres [26]. Les résultats montrent aussi que si des stratégies appropriées comme des sites Web de covoiturage sont conçues pour aider les voyageurs à trouver les itinéraires appropriées, le covoiturage augmenterait de 30% et cette augmentation permettra de réduire la consommation annuelle de carburant d'environ 240 millions de litres. Si l'urgence climatique est globale et lourde de conséquences pour notre futur, les émissions de la voiture sont aussi locales et néfastes pour notre sante, d'es maintenant. Selon une étude de l'Inserm, les véhicules particuliers sont responsables de plus du quart des émissions d'oxyde d'azote, gaz irritants qui pénètrent dans les ramifications les plus fines des voies respiratoires. Chez les personnes sensibles, ils peuvent provoquer des difficultés respiratoires, une hyperréactivité bronchique, et chez l'enfant, ils favorisent accroissement de la sensibilité des bronches aux infections [27].

Le covoiturage s'inscrit donc dans une politique générale de développement durable, en limitant la circulation automobile globale. Il présente une pratique qu' permet de diminuer considérable

le nombre de voitures circulant sur les routes. La première séquence est la diminution de la pollution et de l'émission des gaz à effet de serre. Ceci permet également la diminution de consommation d'énergies non renouvelables comme le pétrole.

II.7.3 Des trajets plus rapides :

Grâce à la diminution du nombre de voitures circulant au même moment, les embouteillages et les autres ralentissements aux entrées des villes, aux heures de pointe, seront considérablement réduits. Par conséquent, les routes deviennent plus fluides et les trajets plus rapides. En plus, les problèmes de stationnement seront également résolus car un nombre moins important de véhicules auront besoin de parking. Par la suite, on perd moins de temps en cherchant une place au parking.

En plus, le covoiturage peut résoudre le problème du transport dans les zones rurales.

Donc, il permet de remédier à l'inefficacité du transport en commun, lorsqu'il s'agit de desservir les zones éloignées du centre de la ville. Dans ce contexte, les résultats d'une simulation réalisée par Lease Plan 8 prouvent que pour se déplacer d'une banlieue vers le centre-ville, le temps de trajet en transports en commun est plus de quatre fois plus long qu'en voiture.

En outre, le covoiturage est une solution efficace pour décongestionner les villes. En effet, grâce à la mise en place d'une voie de covoiturage, les volontaires de San Francisco ont réduit de 30 minutes la durée de trajet quotidien. Et à Toulouse, il est prouvé que 5% de trafic en moins permettrait de fluidifier complètement les trajets domicile-travail, notamment, pendant les heures de pointe [28].

II.7.4 Une route plus conviviale :

En plus de ses intérêts écologiques et économiques, le covoiturage permet de développer des liens sociaux entre les voyageurs. En effet, cette pratique permet de couper avec la monotonie des trajets en voiture, lorsque l'on est seul au volant de sa voiture. Ainsi, covoiturer peut-être une occasion de rencontrer et de convivialité. Ces rencontres permettent de révolutionner le modèle classique des voyages, d'étendre sa culture, de découvrir de nouveaux métiers... Tous ceux qui pratiquent le covoiturage régulièrement l'avouent, les conversations en voiture sont

souvent animées et conviviales.

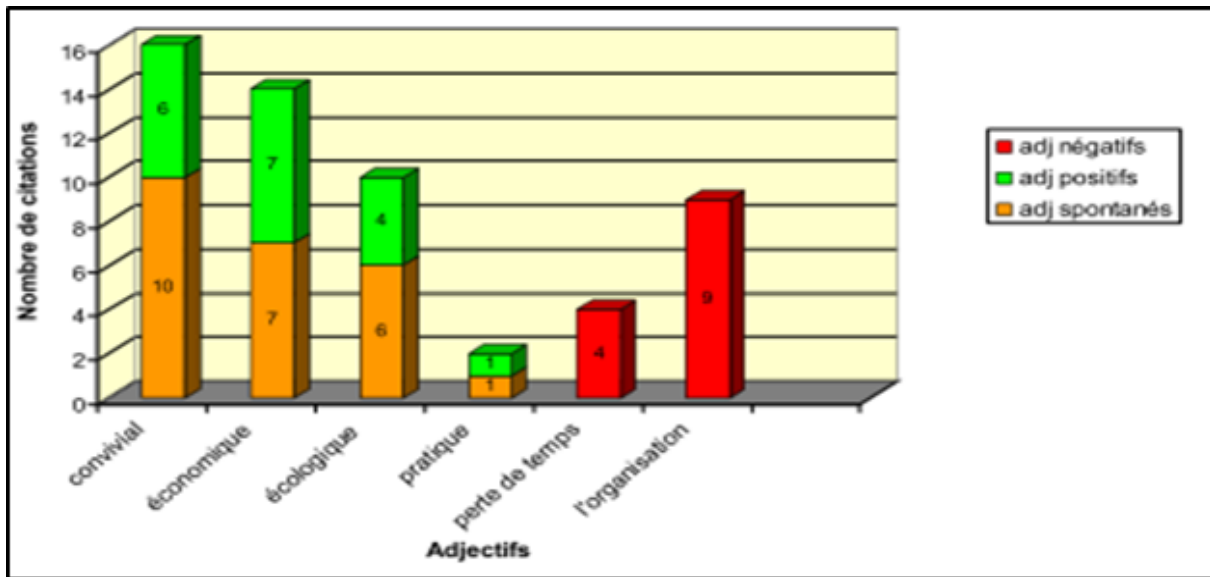


Figure 11: Pourquoi on choisit le covoiturage ? [29]

A la lecture de ce diagramme (Figure 11), le covoiturage s'avère être un mode positif chez la majorité des usagers. Par ailleurs, c'est un mode convivial, ce qui constitue une caractéristique exceptionnelle pour un mode « collectif ». En effet, si les transports communs s'avèrent généralement positifs, vu leurs avantages écologiques, économiques, pratiques, etc. jamais la convivialité n'est vue comme un de leurs atouts [30].

II.8 Les problèmes techniques :

Les contraintes techniques sont perçues par les usagers comme la principale source de critiques et de démotivation. Ils présentent tous les incidents techniques qui peuvent gêner les potentiels covoitureurs, qu'il s'agisse du support du service de covoiturage et son fonctionnement, de la pratique en elle-même ou aussi d'éléments extérieurs indépendants du covoiturage. D'abord, le manque de flexibilité au niveau des horaires de travail est un obstacle majeur à la pratique. Citons l'exemple de deux individus habitant dans le même quartier et travaillant dans la même zone d'activités qui auront tout intérêt à covoiturer pour leur trajet domicile-travail. Toutefois, ces deux personnes ont des horaires distincts, l'un fait 5h – 12h et l'autre 8h – 17h. Dans ce cas, le covoiturage ne peut pas être une solution de déplacement.

De plus, les contraintes s'apparentent à toutes les activités qui exigeraient un détour dans l'itinéraire domicile-travail. Aller chercher les enfants de l'école ou passer aux super marchés

pour faire les courses nécessitent bien souvent un détour dans les trajets, ce qui fait que le covoiturage n'est pas une bonne solution. En effet, généralement, le passager n'est pas forcément enclin à accepter un détour qui peut générer une perte de temps pour lui.

Les différentes situations présentées précédemment confirment le besoin d'une optimisation à cause du caractère plus ou moins aléatoire de l'offre et de la demande. Les horaires et les arrêts ne sont pas connus à l'avance et les itinéraires sont créés à la volée, ce qui nécessite d'introduire des marges temporelles (pour réaliser des détours). Ces marges temporelles se basent essentiellement sur la tolérance des passagers à autoriser plus ou moins de retards causés par les détours occasionnels pour récupérer les passagers potentiels.

D'autre part, la mise en relation entre un conducteur et un passager est un moment clé de la réussite du système. En effet, des échecs répétés de mise en relation peuvent décourager, plus que d'autres problèmes techniques, la motivation des voyageurs, surtout pour les passagers. Autrement dit, le passager ne fera plus de confiance au système de covoiturage. C'est pourquoi la mise en place d'un système d'information pour le covoiturage paraît indispensable pour garantir le succès d'un tel système.

Ce système d'information s'engage à fournir aux participants les renseignements suivants :

- La réception des notifications précises en temps et en heure,
- L'information sur la localisation du passager,
- L'indication réaliste, en cas de congestion, du temps d'attente pour le passager et du temps de détour pour le conducteur. . .

II.9 Conclusion :

Dans ce chapitre, nous avons vu la définition et le principe de covoiturage. Ainsi les formes de covoiturage spontané et organisé. Puis, Les systèmes opérationnels du covoiturage dans le cas statique et dynamique. En fin de parler des avantages et des problèmes techniques de covoiturage.

Chapitre III

Spécification, Analyse, Conception

III.1 Introduction :

Ce chapitre sera divisé en deux parties essentielles, la première sera consacrée pour la notion de covoiturage et son besoin, et la deuxième partie c'est pour présenter quelques diagrammes UML du projet.

III.2 Notion de covoiturage :

À la différence du taxi où le passager choisit la destination, en covoiturage, c'est le conducteur qui offre de partager son véhicule, éventuellement gratuitement ^{et} fixe le trajet.

Un conducteur propose aux passagers de les transporter dans sa voiture pour un trajet (ou une portion de trajet) qu'il doit lui-même effectuer, et donc à la date et à l'heure qu'il a décidée. Généralement, le lieu de départ, déterminé à l'avance, est le même pour tout le monde.

À l'arrivée, le conducteur dépose les passagers là où lui-même s'arrête, ou bien à l'endroit que chacun souhaite, en évitant de faire un grand détour. Il peut alors laisser ses passagers par exemple à proximité d'un transport en commun ou bien là où un membre de la famille ou un ami pourront les prendre en charge.

Le partage des frais est laissé à l'appréciation du conducteur. La formule la plus classique consiste à diviser le coût du carburant et des péages éventuels par le nombre de personnes. Les frais généraux tels que ceux d'entretien ou d'assurance peuvent être inclus dans le calcul du coût du trajet.

III.3 Charte du Covoiturage :

En vous inscrivant sur notre site vous acceptez de respecter les règles de la charte du covoiturage solidaire :

La présente charte a pour objet de formaliser les règles de déontologie et de sécurité relatives à l'utilisation du service de covoiturage.

- **Responsabilité :**

Les utilisateurs du site, acceptent d'agir sous leur seule et entière responsabilité. On ne saurait, en aucun cas, être tenu pour responsable de tout dommage que pourrait subir une personne du fait de l'utilisation du présent site, notamment en cas d'utilisation du site en violation de la présente charte, de dommage survenant à l'occasion d'un voyage, de retard ou d'absence d'un conducteur ou d'un passager.

- **Sincérité**

Toute personne désirant s'inscrire sur le présent site accepte de communiquer des informations exactes et exhaustives et de veiller à leur mise à jour régulière.

- **Utilisation du service**

L'utilisation du présent site Internet et des adresses de courriel de ses utilisateurs doit avoir pour seul but de rechercher des personnes avec qui organiser un covoiturage dans le respect des réglementations en vigueur. Il ne doit pas être utilisé dans le but de transmettre tout autre contenu. L'utilisation du site ne doit pas avoir de finalité commerciale, le service de covoiturage rendu par les conducteurs ne pouvant faire l'objet que d'une indemnisation dans le cas de déplacements réguliers, conformément à l'article « Répartition des frais » et non à une rémunération.

Compte tenu de la spécificité du covoiturage, l'inscription à ce site n'est autorisée qu'aux personnes majeures. Dans le cas d'un covoiturage domicile/école ou lieu de formation pour un mineur c'est le responsable légal du mineur qui s'inscrit sur le site.

Par ailleurs, l'utilisateur nous autorise à accéder aux informations enregistrées dans son profil, notamment pour résoudre certains problèmes techniques.

- **La sécurité**

Le conducteur s'engage à respecter le code de la route. Il garantit également que son permis de conduire est valide et que son véhicule est en parfait état d'usage et d'entretien, que les contrôles de sécurité sont effectués en conformité avec la réglementation, qu'il souscrit un contrat d'assurance pour le véhicule et que ce contrat couvre les passagers transportés. Dans l'hypothèse où le conducteur verrait sa situation modifiée, il s'engage à le déclarer immédiatement.

- **Comportement des utilisateurs**

Les utilisateurs du covoiturage acceptent de prendre toutes les mesures de sécurité qui s'imposent, de propreté et de respect des autres utilisateurs ainsi que de respecter l'ensemble des réglementations en vigueur. De manière générale, les utilisateurs du covoiturage se soumettent à une obligation de ponctualité, s'engagent à ne transporter aucune substance illicite ou dangereuse et à adopter en toute circonstance un comportement respectueux et responsable. Le passager s'engage particulièrement à :

- respecter la propreté du véhicule dans lequel il est transporté ;
- Ne pas gêner le conducteur durant le trajet ;
- être en état de sobriété.

- **Répartition des frais**

Dans le cadre d'un déplacement ponctuel, le conducteur s'engage à ne pas demander de participation au passager conformément aux valeurs de solidarité et d'entraide qui caractérisent le projet.

Dans le cadre de déplacements réguliers, la répartition des frais est librement convenue entre les utilisateurs. Le montant ou la règle de calcul doit être arrêtée dès le départ.

On rappelle toutefois que la participation demandée ne doit pas excéder le montant des frais réellement subis par le conducteur, le transport rétribué étant soumis à des règles particulières et pourrait engendrer des difficultés en cas d'accident.

- **Disponibilité du service**

Dans le cadre d'un déplacement ponctuel, le conducteur s'engage à ne pas demander de participation au passager conformément aux valeurs de solidarité et d'entraide qui caractérisent le projet.

Dans le cadre de déplacements réguliers, la répartition des frais est librement convenue entre les utilisateurs. Le montant ou la règle de calcul doit être arrêtée dès le départ.

On rappelle toutefois que la participation demandée ne doit pas excéder le montant des frais réellement subis par le conducteur, le transport rétribué étant soumis à des règles particulières et pourrait engendrer des difficultés en cas d'accident.

III.4 Spécifications des besoins:

III.4.1 Les 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, implémente principalement les fonctionnalités suivantes :

Module des inscriptions : Le système doit permettre aux conducteurs et passagers 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 covoitureurs de rechercher 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 son 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.

III.4.2 Les besoins non fonctionnels :

Les besoins non fonctionnels caractérisent les propriétés de notre site, les contraintes d'environnement et d'implémentation, les dépendances de la plate-forme, les capacités de maintenance, l'extensibilité et la fiabilité.

Une première analyse des conditions d'exploitation souhaitées, nous a permis d'identifier les besoins non-fonctionnels décrits ci-après :

- La fiabilité : Le système doit être disponible à tout moment pour l'utilisateur, avec un accès sécurisé par la définition d'un login et d'un mot de passe.

- La simplicité : Le système doit présenter une interface compréhensible, facile à manipuler permettant d'accroître la rentabilité et l'efficacité du système.
- L'ergonomie : L'interface de notre application doit être optimiser pour réduire les erreurs tout en facilitant l'utilisation.
- Le temps de réponse : Le système doit répondre aux requêtes des utilisateurs en un minimum de temps possible.

III.5 Analyse des besoins :

Identification des acteurs du système :

Un acteur est une personne, un matériel ou un logiciel qui interagit directement avec le système pour réaliser une tâche. Ainsi, un acteur peut consulter et/ou modifier directement l'état du système en émettant et/ou recevant des messages susceptibles contenir des données. Durant notre analyse nous avons identifiés les acteurs suivants :



Conducteur

- ✓ Inscription
- ✓ Authentification
- ✓ Publier une annonce du covoiturage
- ✓ Recevoir des notifications sur la réservation des passagers.

Conducteurs : Les conducteurs permettent d'alimenter le système en itinéraires disponibles pour la recherche. En effet, c'est la quantité et qualité des conducteurs qui permettront d'avoir une banque de données intéressante dans le système. Suite à la publication d'un trajet, ils sont responsables de vérifier à intervalle régulier s'ils reçoivent des demandes d'embarquement des covoitureurs. Dans un tel cas, ils sont responsables de confirmer l'embarquement du passager à un point donné. De plus, ils peuvent consulter leurs historiques trajets.

**Passager**

- ✓ Inscription
- ✓ Authentification
- ✓ Consultation des covoiturages
- ✓ Consultation des conducteurs
- ✓ Effectuer des réservations

Passagers : Les passagers sont ceux qui bénéficient le plus du système. Ils peuvent lancer diverses recherches pour trouver des trajets qui conviennent à leurs besoins et faire des réservations. Dans le cas où les points d'embarquements conviennent plus ou moins, ils peuvent envoyer une proposition de point d'embarquement au conducteur. De plus, ils peuvent consulter leur historique de réservations.

**Administrateur**

- ✓ Gestion des administrateurs
- ✓ Gestion des covoiturages
- ✓ Gestion des conducteurs
- ✓ Suivi des réservations

Administrateur : L'administrateur est le seul à avoir accès à l'information complète du système. Il peut faire la gestion globale du système, par exemple créer des comptes de covoitureurs. Il lui est aussi possible de suspendre n'importe quel compte.

III.6 Conception :

III.6.1 Identification des cas d'utilisation :

Un 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 chaque scénario est un traitement particulier du système.

III.6.1.1 Diagramme cas d'utilisation de passager :

• **Le Passager** : c'est celui qui possède un compte afin de consulter la liste des offres de covoiturage et d'effectuer des réservations, rechercher un trajet les prix et consulter les informations sur un conducteur sélectionné.

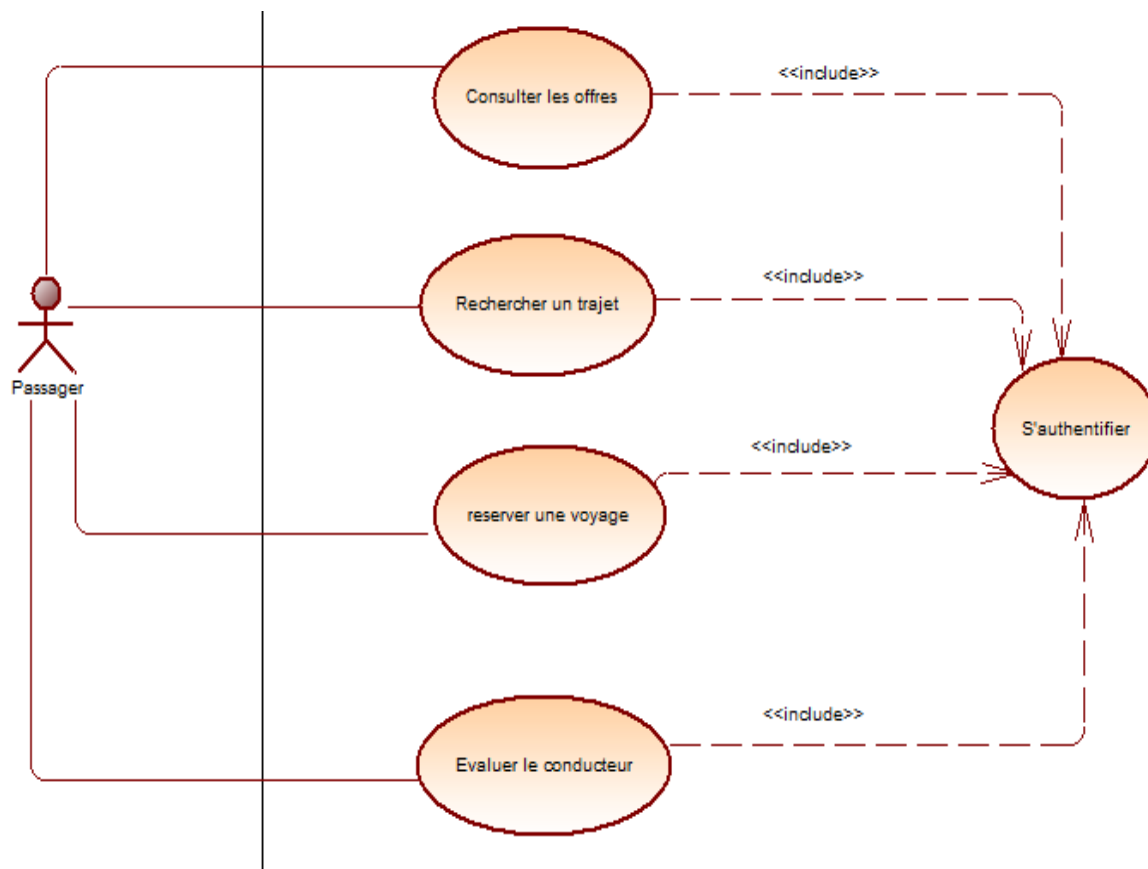


Figure 12:Diagramme cas d'utilisation de passager

III.6.1.2 Diagramme cas d'utilisation de conducteur :

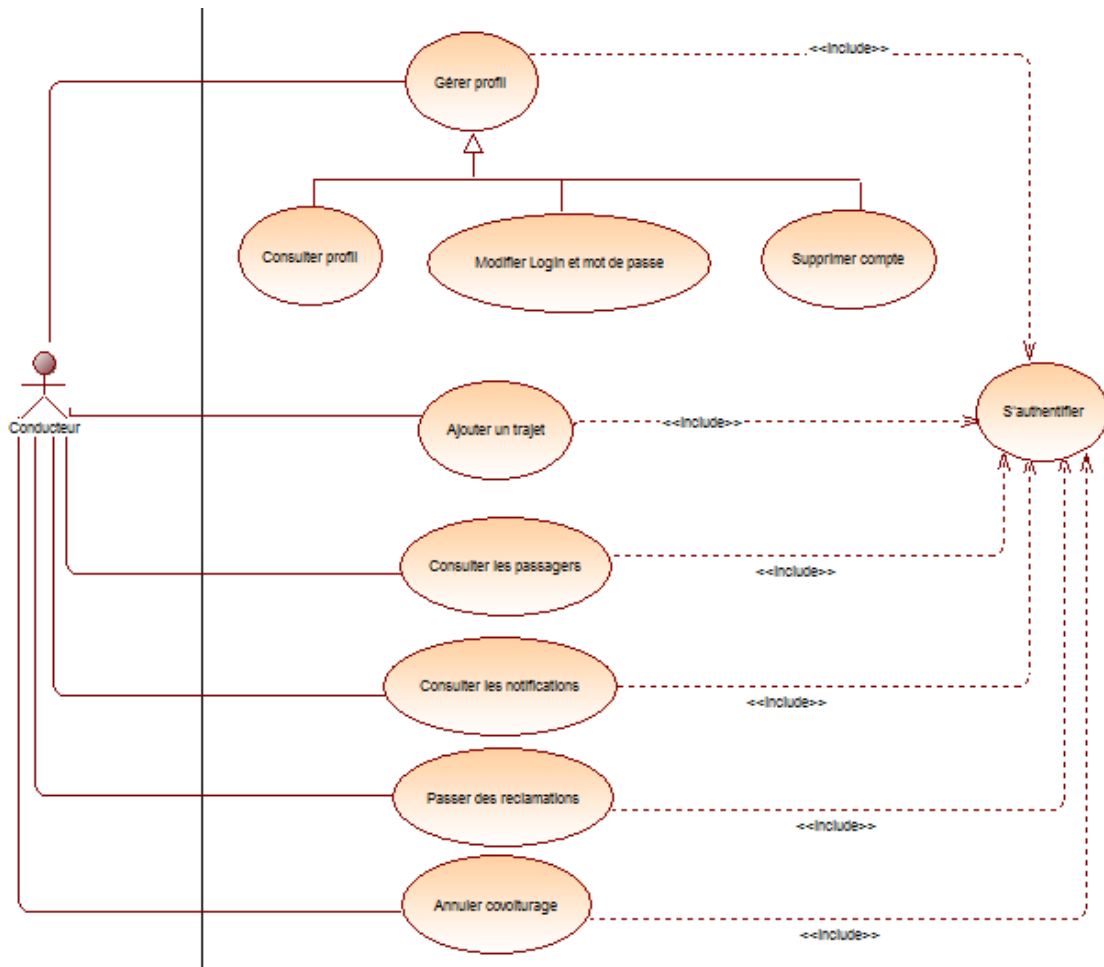


Figure 13:Diagramme cas d'utilisation de conducteur

• **Le Conducteur** : c'est celui qui possède un compte afin d'un covoiturage ainsi ajouter son trajet, il consulte les passagers et il reçoit des notifications sur son covoiturage, aussi il peut gérer son profil par consulter son profil, modifier son login et mot de passe ou supprimer compte.

III.6.1.3 Diagramme cas d'utilisation de l'administrateur :

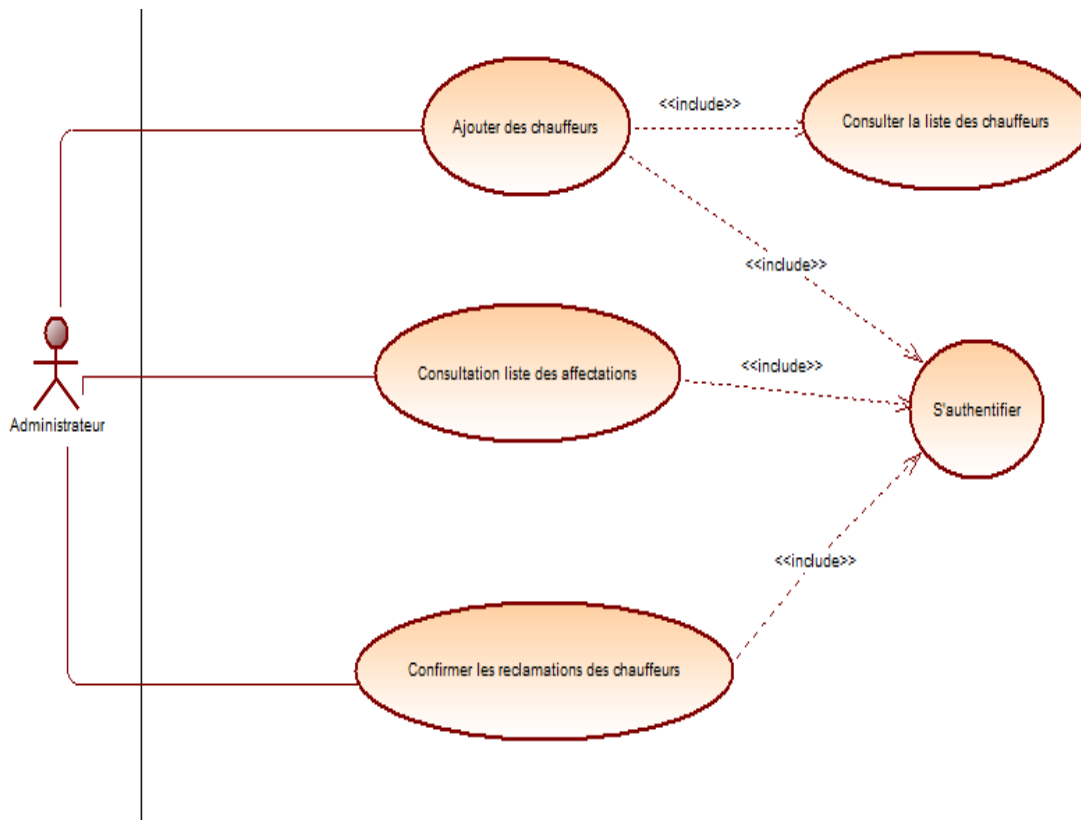


Figure 14:Diagramme cas d'utilisation de l'administrateur

• **L'Administrateur** : c'est celui qui le responsable de la gestion de la partie "Administration" (back office), il a comme rôle de gérer les covoiturages, également les conducteurs et confirme les réclamations des conducteurs. Toutes ces activités ont une relation d'inclusion vers l'authentification.

III.6.1.4 Diagramme cas d'utilisation Global :

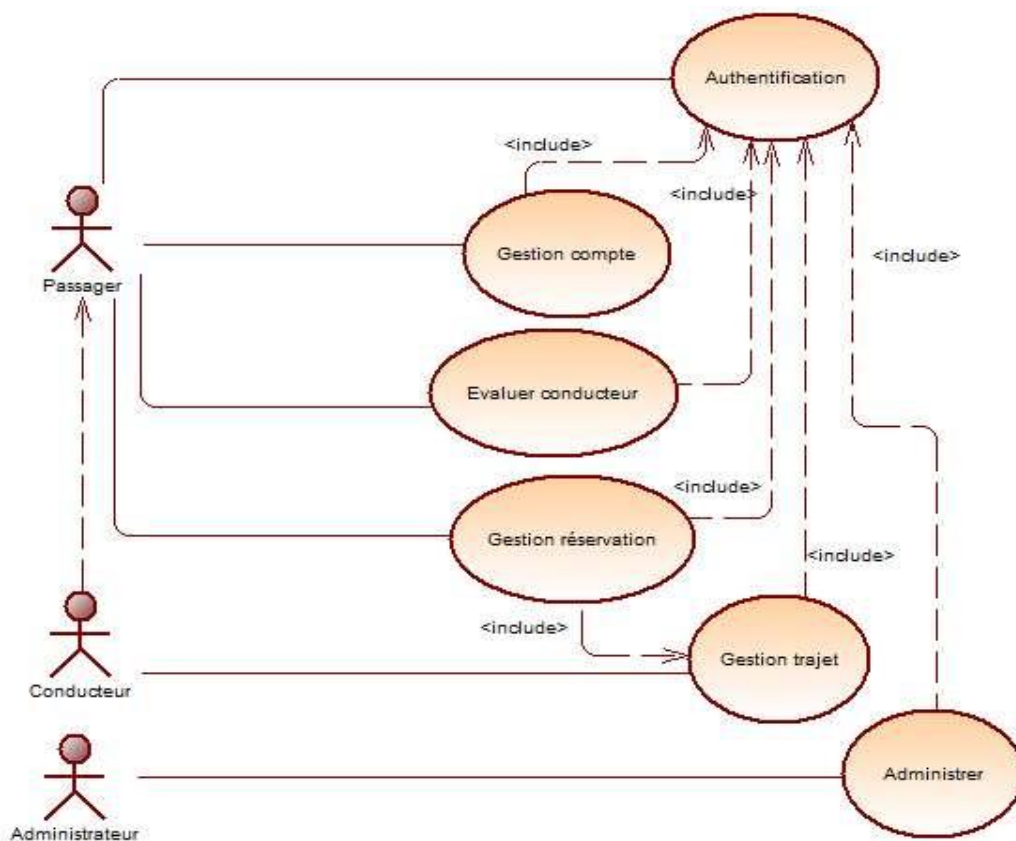


Figure 15:Diagramme cas d'utilisation global

III.6.2 Diagrammes des séquences détaillées:

II.6.2.1 Diagrammes des séquences détaillées de la partie Passager :

i. Diagramme de séquence du cas consulté des offres

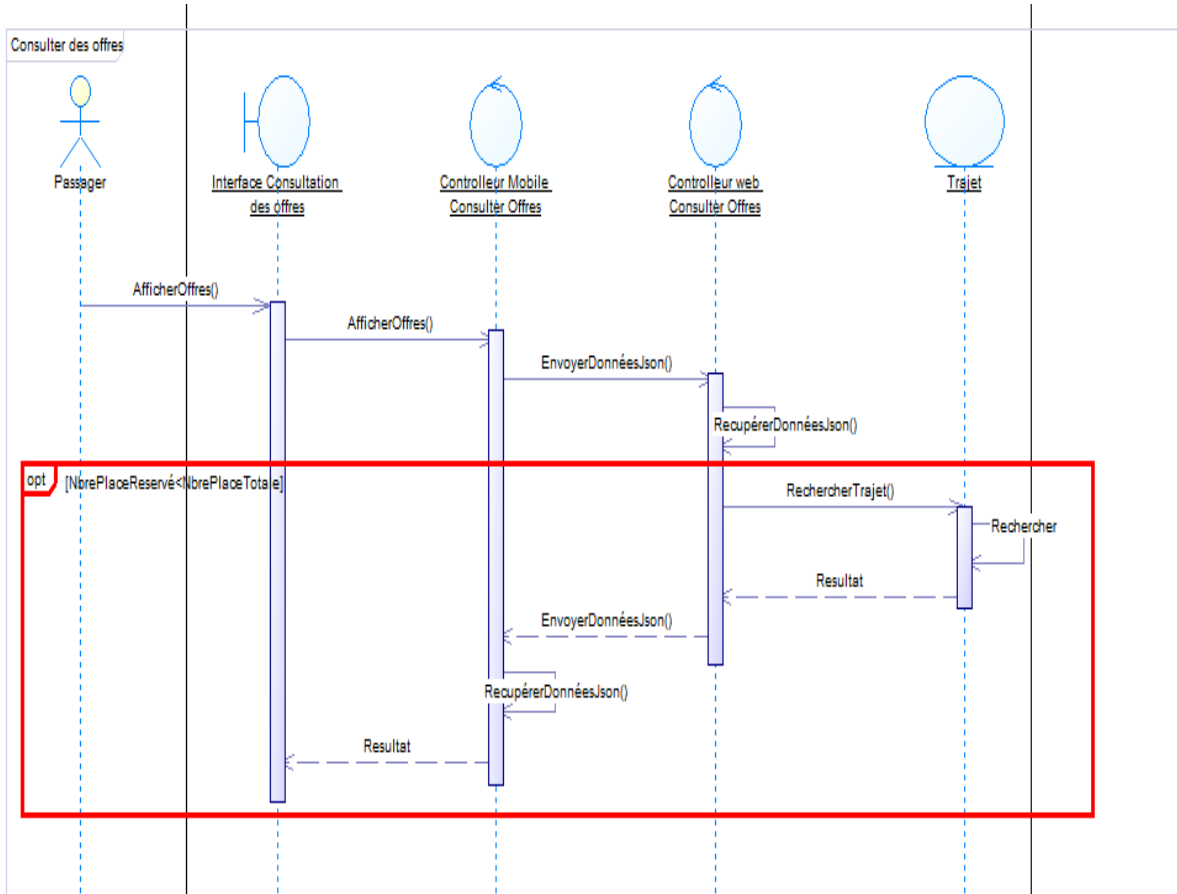


Figure 16:Diagramme de séquence du cas consulté des offres

- Ce diagramme permet au passager de consulter toutes les offres.

ii. Diagramme des séquences détaillées Rechercher Trajet

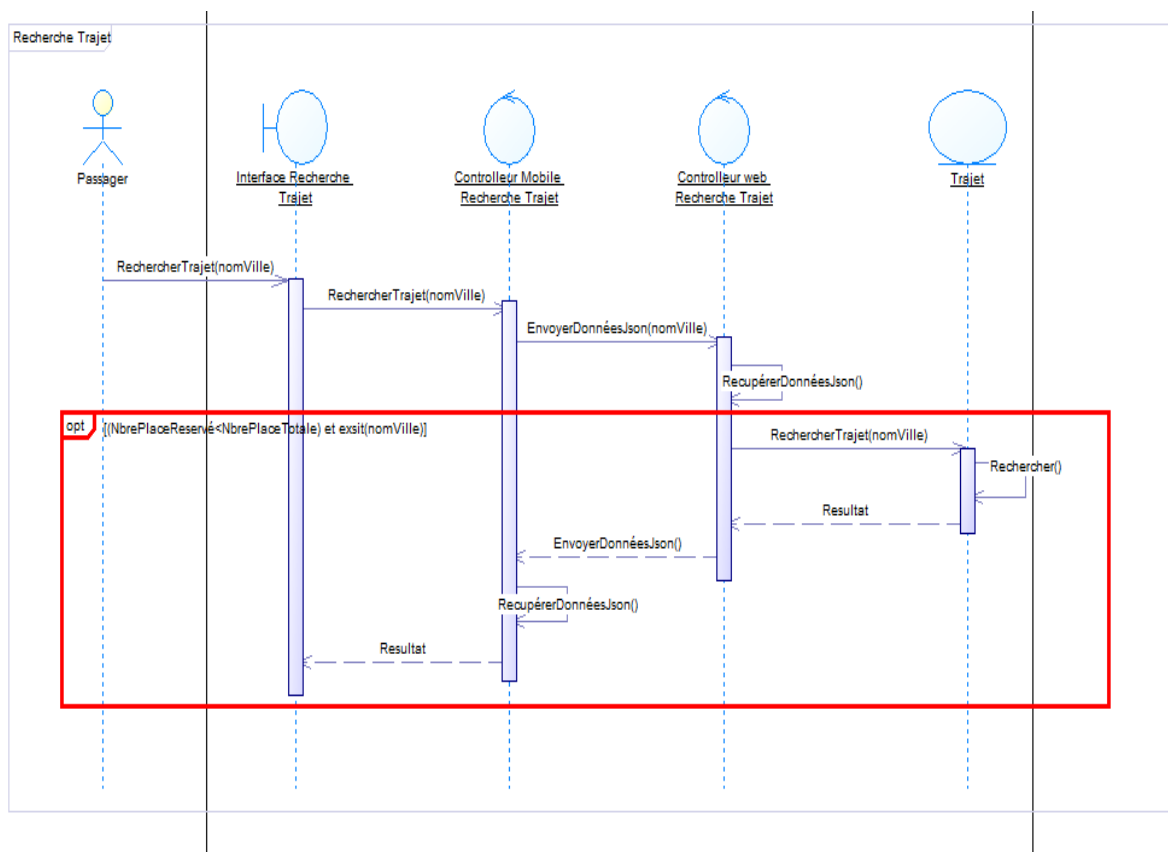


Figure 17: Diagramme des séquences détaillées Rechercher Trajet

- Ce diagramme décrit les différentes les étapes nécessaires qu'un passager doit rechercher un trajet par des critères spécifiques que ce soit ville départ, ville arrivée et date.

iii. Diagramme des séquences détaillées Réserver un covoiturage

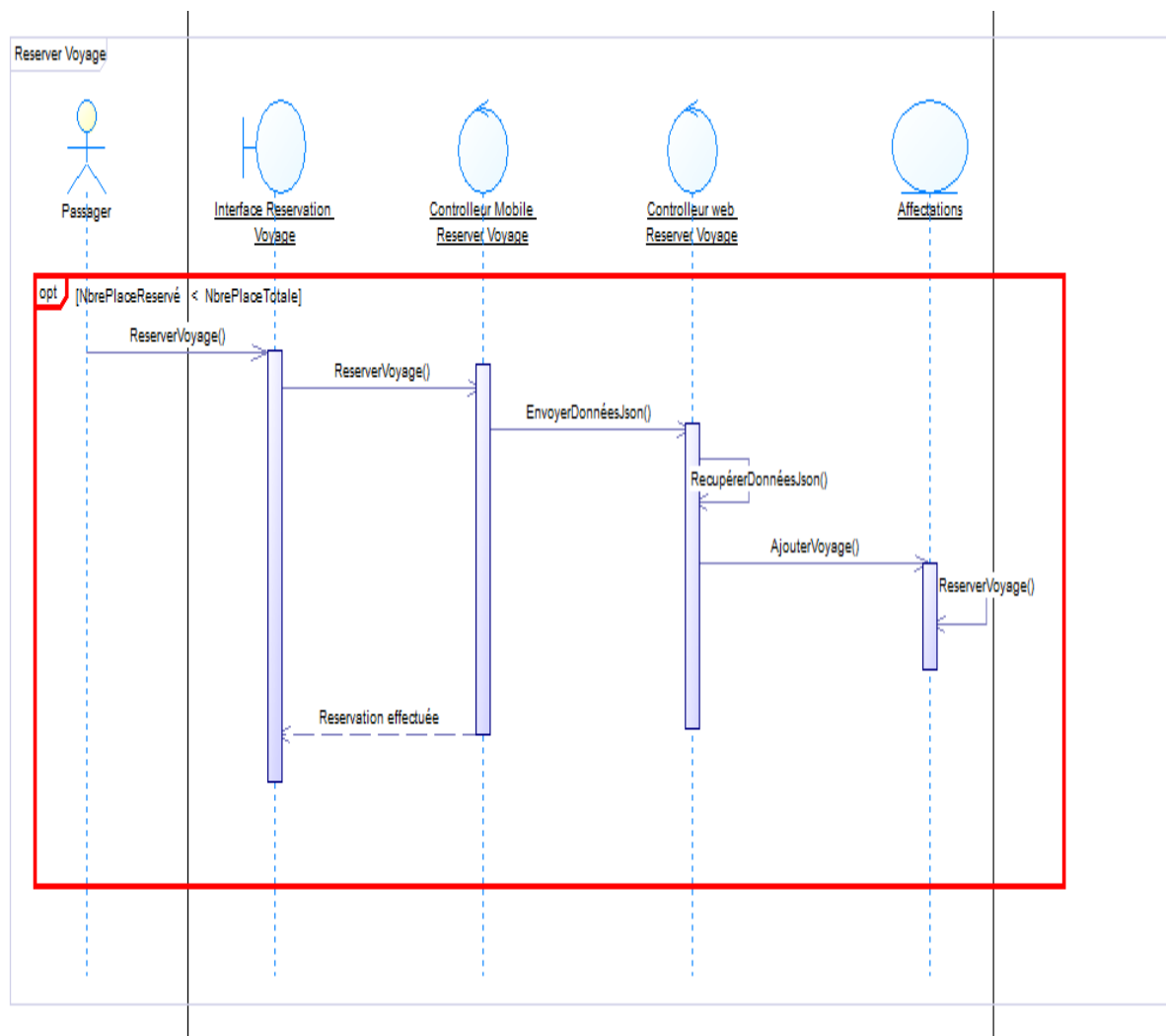


Figure 18:Diagramme des séquences détaillées Réserver un covoiturage

- Ce diagramme décrit les différentes les étapes nécessaires qu'un passager doit suivre afin de réserver un covoiturage.

III.6.2.2 Diagrammes des séquences détaillées de la partie Conducteur :

i. Diagramme des séquences détaillées Ajouter Trajet

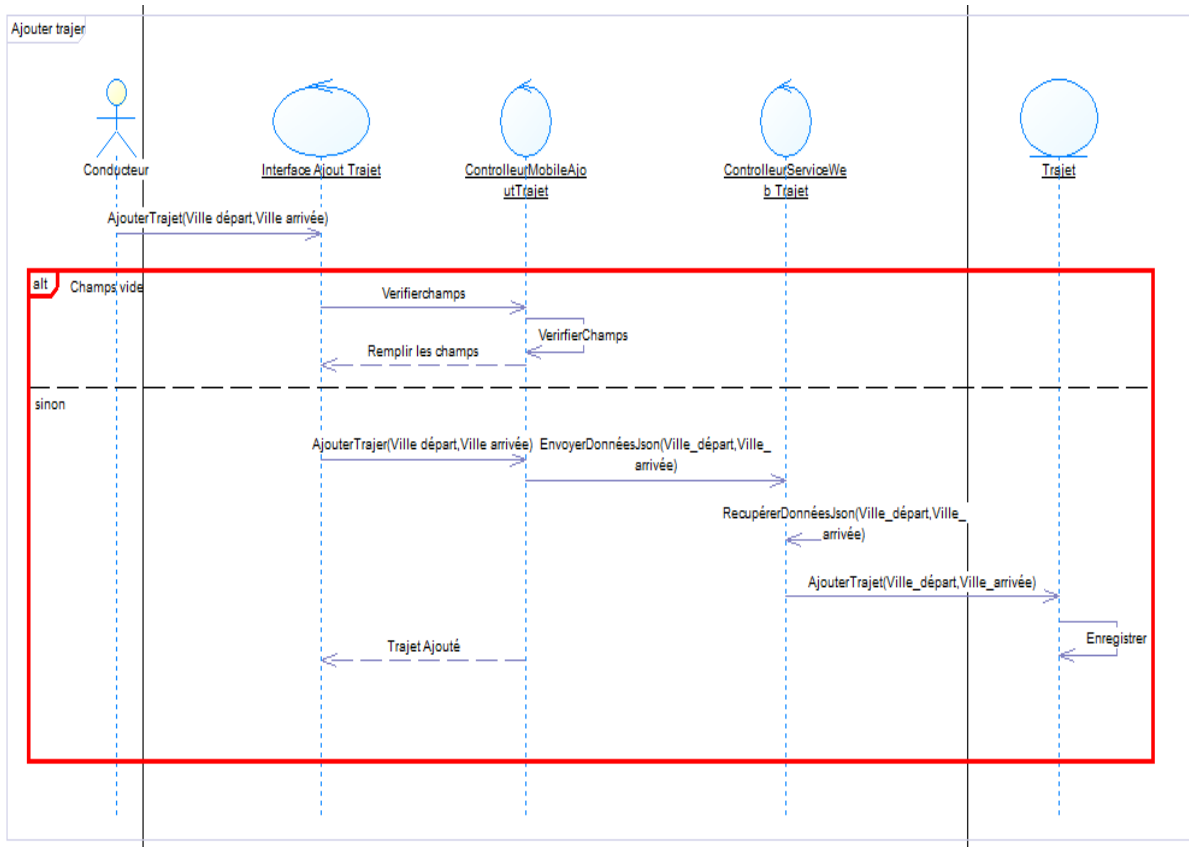


Figure 19:Diagramme des séquences détaillées Ajouter Trajet

- Ce diagramme décrit les différentes les étapes nécessaires qu’un conducteur doit suivre afin d’ajouter des trajets avec spécifications des informations nécessaires tel que la ville départ, ville d’arrivée.

ii. Diagramme des séquences détaillées Consulter Passager

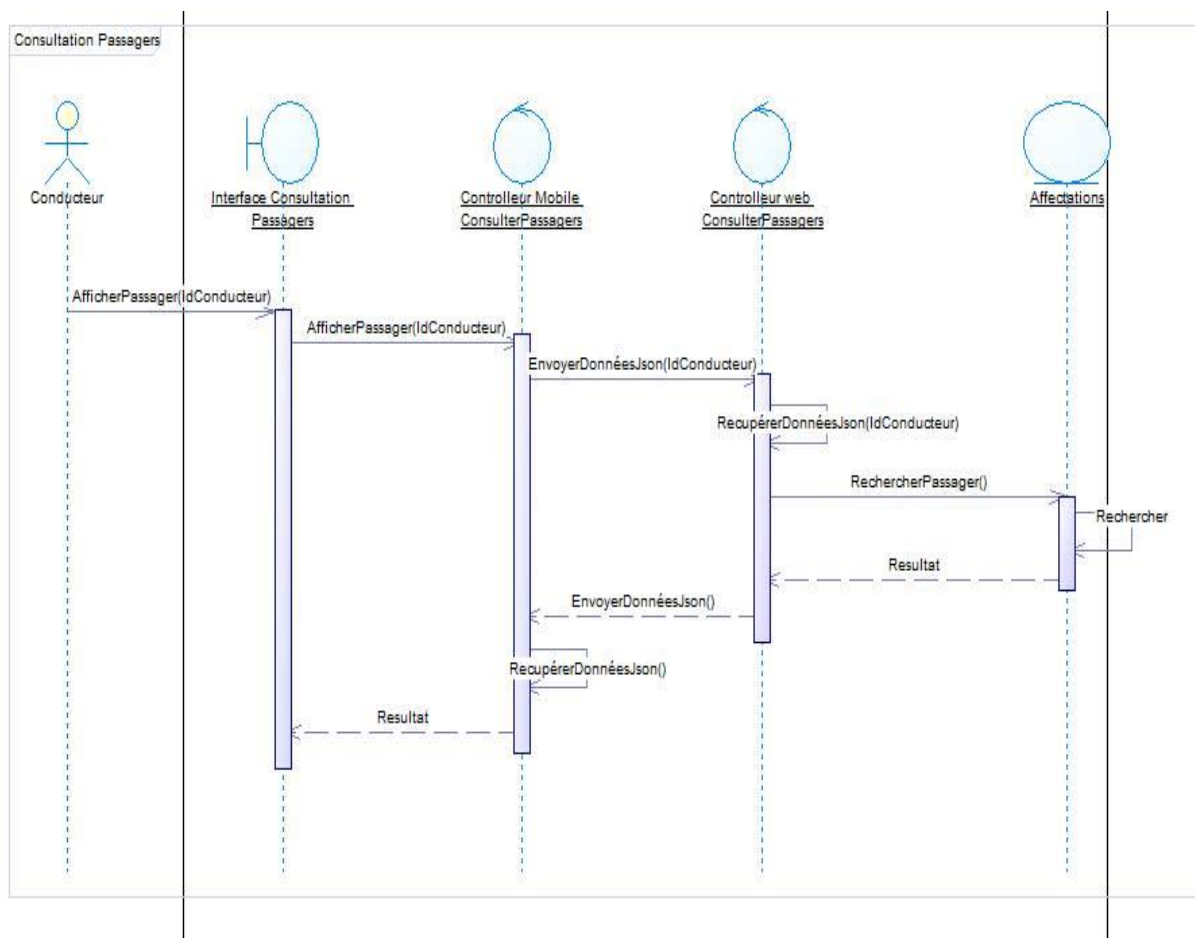


Figure 20:Diagramme des séquences détaillées Consulter Passager

- Ce diagramme décrit les différentes les étapes nécessaires qu'un conducteur doit suivre pour consulter un passager et voir son profil.

iii. Diagramme des séquences détaillées Consulter Notification

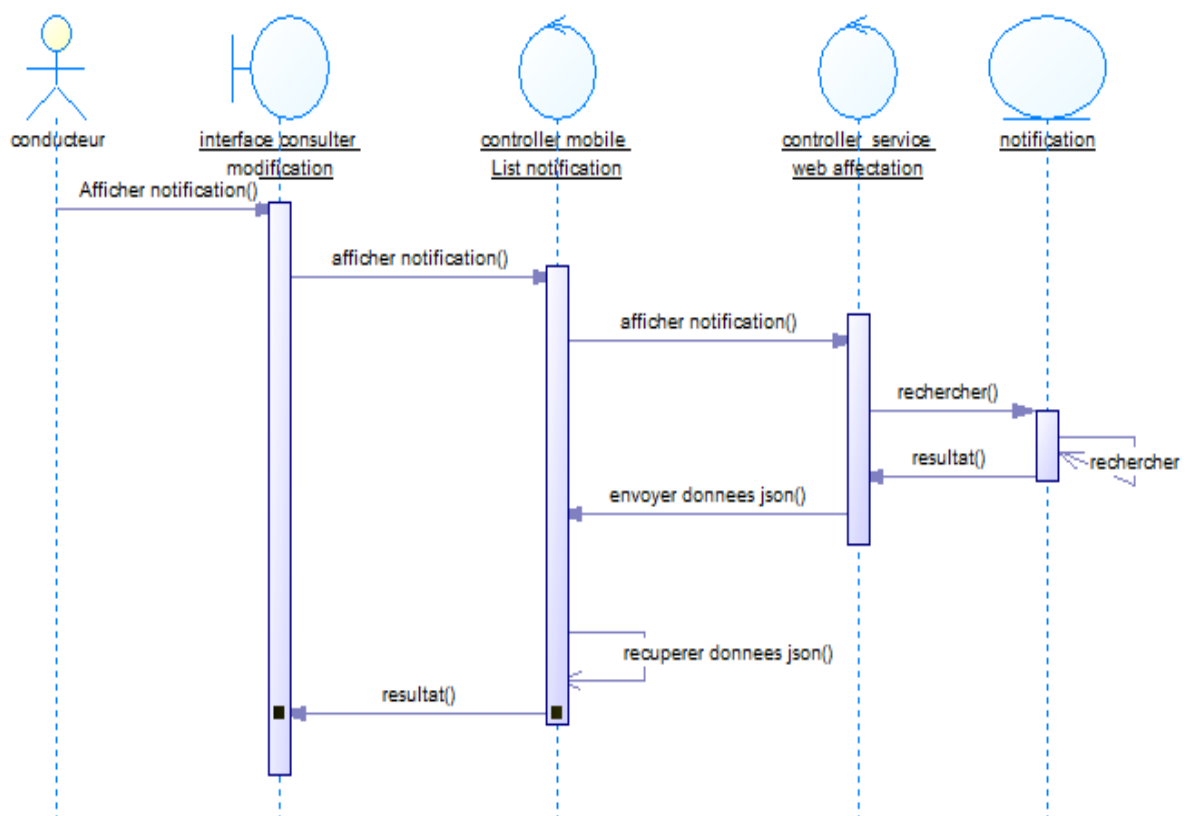


Figure 21:Diagramme des séquences détaillées Consulter Notification

- Ce diagramme décrit les différentes les étapes nécessaires qu'un conducteur doit suivre pour consulter les notifications sur ces covoiturages ajoutés.

iv. Diagramme des séquences détaillées Ajouter une réclamation

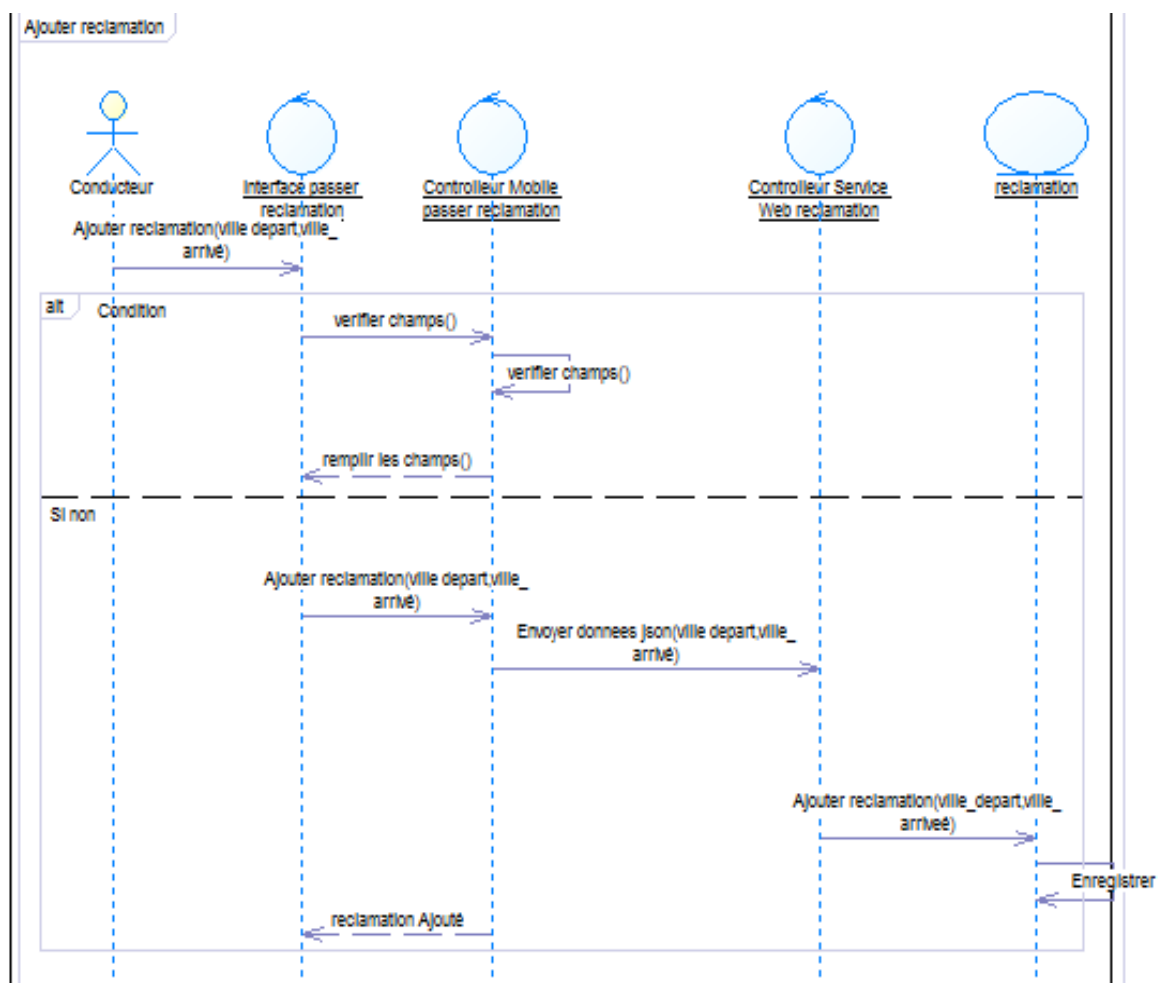


Figure 22:Diagramme des séquences détaillées Ajouter une réclamation

- Ce diagramme décrit les différentes les étapes nécessaires qu'un conducteur doit suivre pour ajouter une réclamation d'un nouveau covoiturage.

III.6.2.3 Diagrammes des séquences détaillées de la partie Administrateur :

i. Diagramme des séquences détaillées Ajouter Conducteur

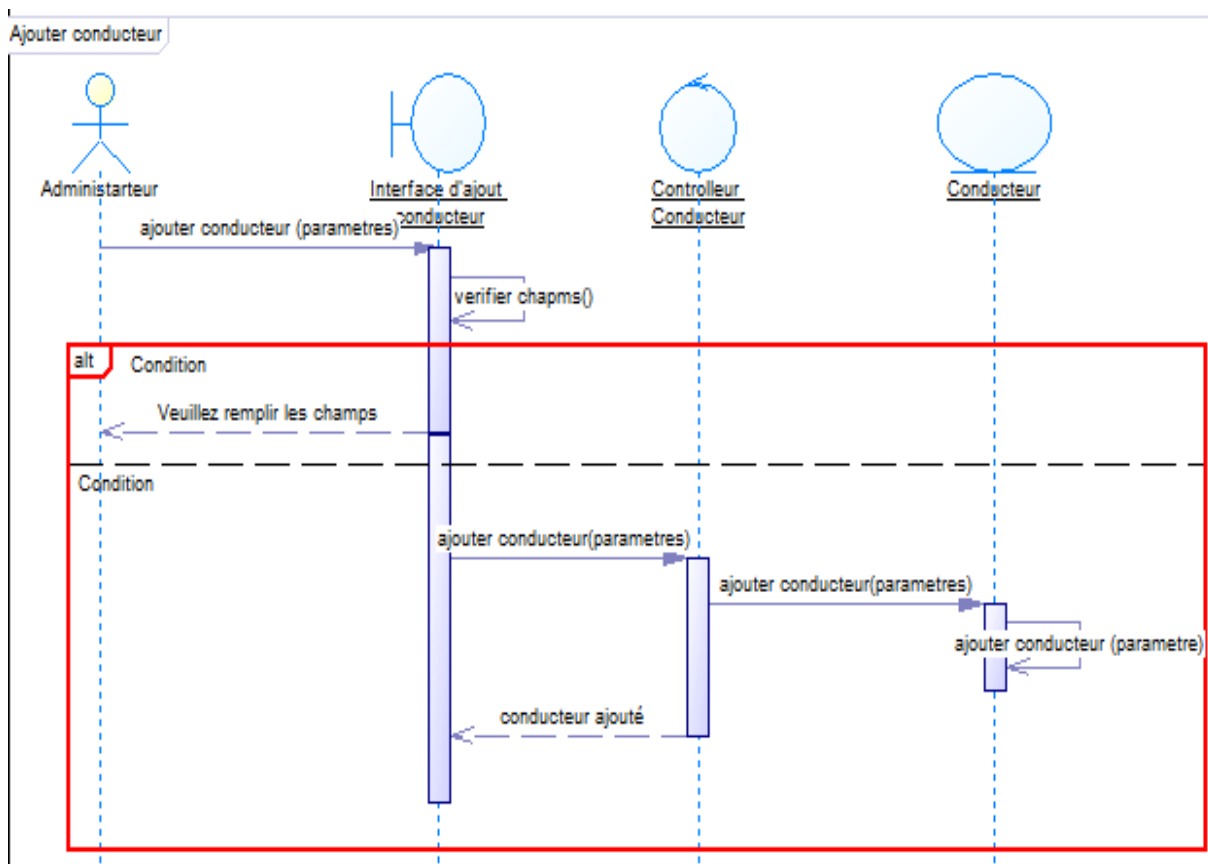


Figure 23:Diagramme des séquences détaillées Ajouter Conducteur

- Ce diagramme décrit les différentes les étapes nécessaires qu'un administrateur doit suivre afin d'ajouter un nouveau conducteur et cela après validation de l'authenticité des données saisies.

ii. Diagramme des séquences détaillées Consulter Conducteur

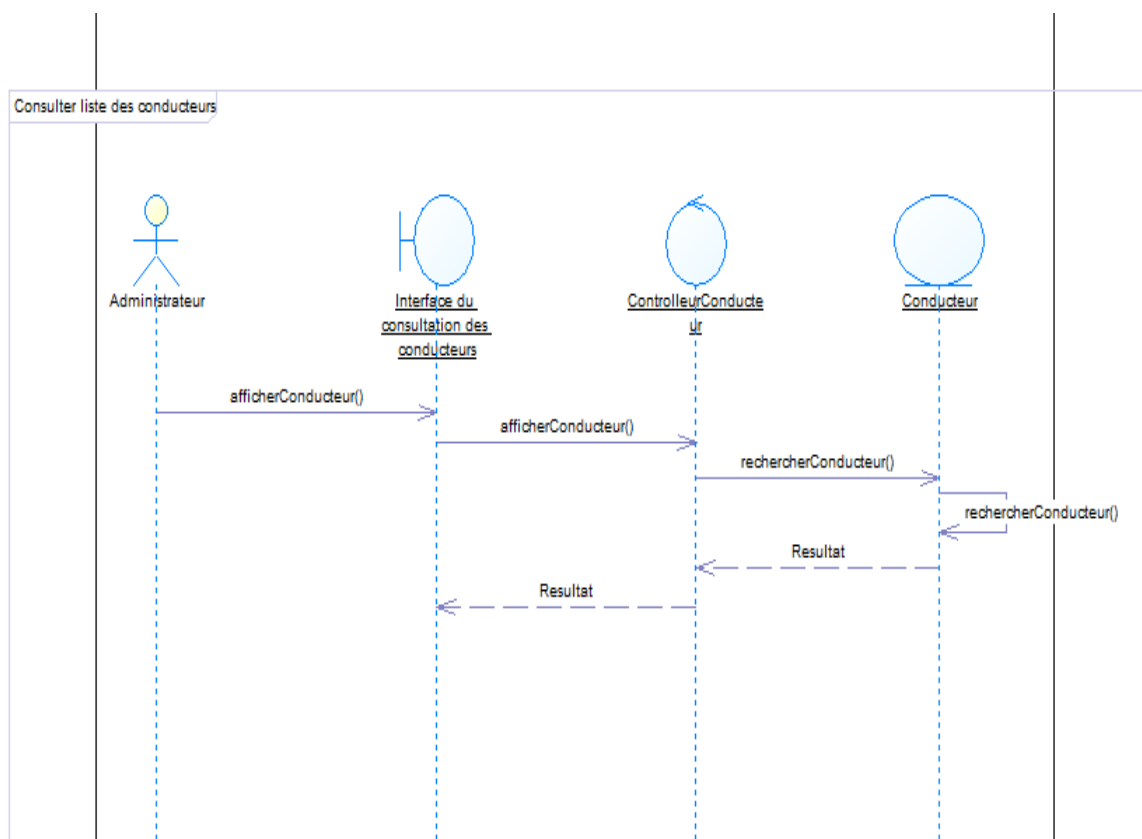


Figure 24:Diagramme des séquences détaillées Consulter Conducteur

- Ce diagramme décrit les différentes les étapes nécessaires qu'un administrateur doit suivre afin de consulter la liste des conducteurs et ses informations.

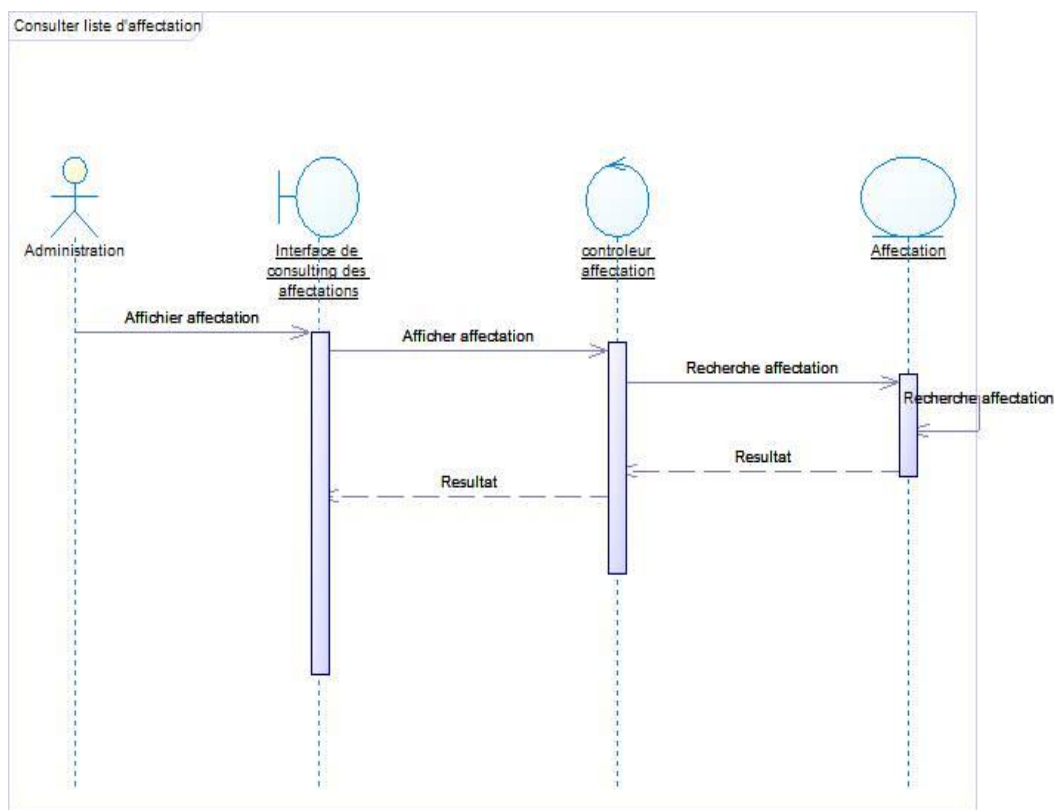
iii. Diagramme des séquences détaillées Consulter la liste des affectations

Figure 25:Diagramme des séquences détaillées Consulter la liste des affectations

- Ce diagramme décrit les différentes les étapes nécessaires qu'un administrateur doit suivre afin de consulter la liste des affectations et ses effectuées par les passagers sur les covoiturages.

III.6.3 Diagramme de classe globale :

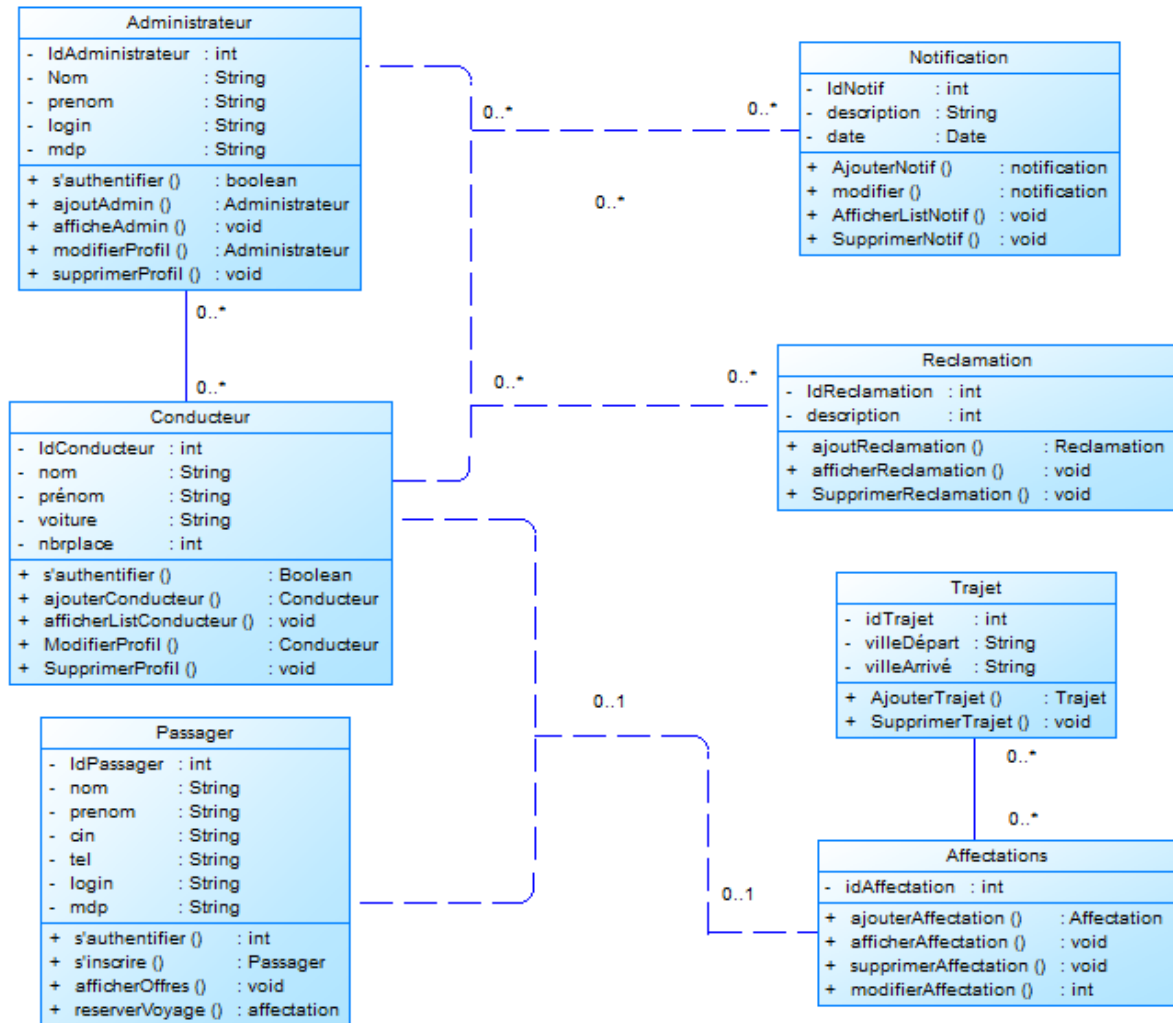


Figure 26:Diagramme de classe globale

Ce diagramme de classe illustre toutes les moles d’objets lors de l’exécution du programme. On note 6 classes et 4classes associatives :

- Classe :
 - Classe Administrateur : on présente un modèle de tous les informations nécessaires pour l’administrateur, comme son identifiants unique (Clé primaire), son nom, son prénom, son login et mot de passe.
 - Classe Passager : on présente un modèle de tous les informations nécessaires pour les passagers, comme son identifiants unique (Clé primaire), son nom, son prénom, son login...

- Classe Conducteur : on présente un modèle de tous les informations nécessaires pour les conducteurs, comme son identifiants unique (Clé primaire), son nom, son prénom, son login...
- Classe Trajet : On présente uniquement les informations sur le trajet comme vile départ, ville arrivée
- Classe Notification : Classe qui sert à gérer les informations des notifications.
- Classe Réclamation on présente un modèle de tous les informations nécessaires pour l'objet Réclamation.
- Classe Affectations : Classe qui sert à gérer les informations des affectations.

III.7 Conclusion :

Ce chapitre nous mène à dégager et à définir la notion de covoiturage, Charte du covoiturage, Ensuite nous identifions la spécification des exigences, l'analyse de besoins et enfin la conception de notre projet.

Chapitre IV
Implémentation

IV.1 Introduction :

Notre travail a pour but de faire une application web qui présenter notre simulation du covoiturage.

Avant de présenter notre implémentation, nous allons présenter les outils nécessaires utilisés.

IV.2 Les choix techniques :

Afin de mener à bien notre projet, nous avons utilisé comme outils de développement :

(Maven, Java Développement kit, IntelliJ IDEA, Xamp, bootstrap, springboot) et les langages (Java EE, JavaScript, HTML, CSS, SQL) pour développer notre application.

IV.2.1 Les outils utilisent :



Maven : Apache Maven (couramment appelé Maven) est un outil de gestion et d'automatisation de production des projets logiciels Java en général et Java EE en particulier. Il est utilisé pour automatiser l'intégration continue lors d'un développement de logiciel. Maven est géré par l'organisation Apache Software Foundation. L'outil était précédemment une branche de l'organisation Jakarta Project. [31]



Le Java Development Kit (JDK) : désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en bytecode destiné à la machine virtuelle Java. [32]

Il existe plusieurs éditions de JDK, selon la plate-forme Java (« Consolidated JDK 15 Release Notes ») considérée (et bien évidemment la version de Java ciblée) :

- JSE pour la Java 2 Standard Edition également désignée J2SE.
- JEE, sigle de Java Enterprise Edition également désignée J2EE.
- JME 'Micro Edition', destinée au marché mobile.



IntelliJ IDEA : également appelé « IntelliJ », « IDEA » ou « IDJ » est un environnement de développement intégré (en anglais Integrated Development Environment - IDE) de technologie Java destiné au développement de logiciels informatiques. Il est développé par JetBrains (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, open source, sous licence Apache 2 et l'autre propriétaire, protégée par une licence commerciale. Tous deux supportent les langages de programmation Java, Kotlin, Groovy et Scala. [33]



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 (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus. [34]



Bootstrap : est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub. [35]



Spring Boot : assure la surveillance des projets de disponibilité basés sur http, ssh et telnet. Spring Boot met en œuvre la configuration automatique, réduit la complexité de la construction et comprend un ensemble complet de kits de développement rapide. [36]

IV.2.2 Les Langages :



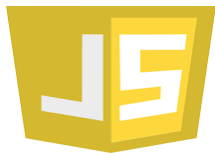
J2EE : est l'acronyme de Java 2 Entreprise Edition. Cette édition est dédiée à la réalisation d'applications pour entreprises. J2EE est basé sur J2SE (Java 2 Standard Edition) qui contient les API de base de Java. Depuis sa version 5, J2EE est renommée Java EE (Enterprise Edition). [37]



L'**HTML** est un langage informatique utilisé sur l'internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie (*HyperText MarkupLanguage*), ce qui signifie en français "*langage de balisage d'hypertexte*". Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage. [38]



Le terme **CSS** est l'acronyme anglais de (*Cascading Style Sheets*) qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers [HTML](#) ou [XML](#). Ainsi, les feuilles de style, aussi appelées les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en [HTML](#). [39]



JavaScript : est un **langage de programmation** (dit "de script") interprété par le navigateur Internet. Le code javascript est inséré dans le code HTML des pages web afin de permettre des traitements (calculs, animations, ...) qui ne peuvent être effectués par le langage HTML, ces est un langage séquentiel réservé uniquement à la mise en forme de documents. [40]



MySQL : est un système de gestion de base de données (SGBD). Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde. MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multithread et multiutilisateur. [41]

IV.3 Présentation de notre application web :

Nous allons presentie quelques pages de l’application :

IV.3.1 Page d'accueil : Elle constitue le point d’entrée de notre application web.

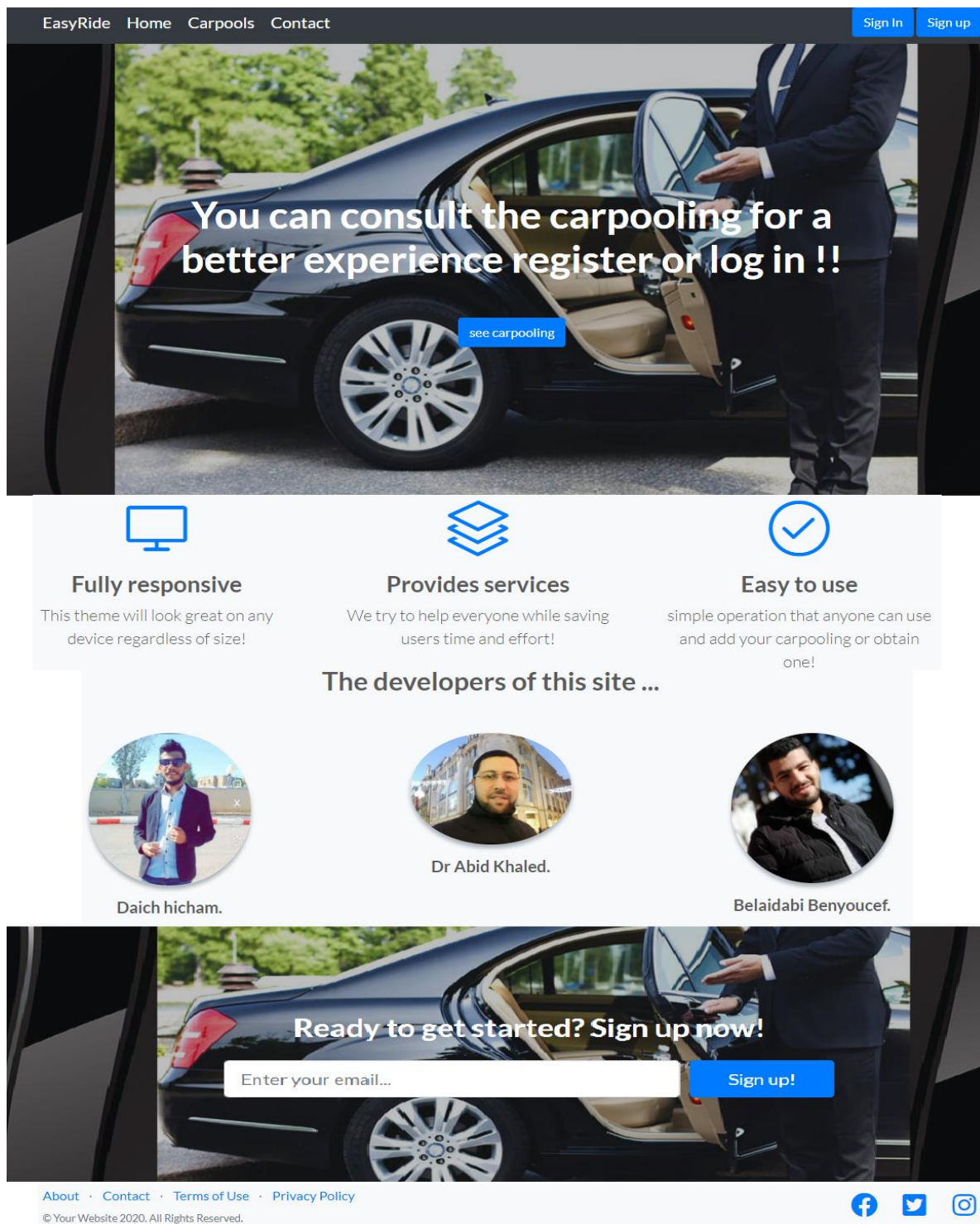
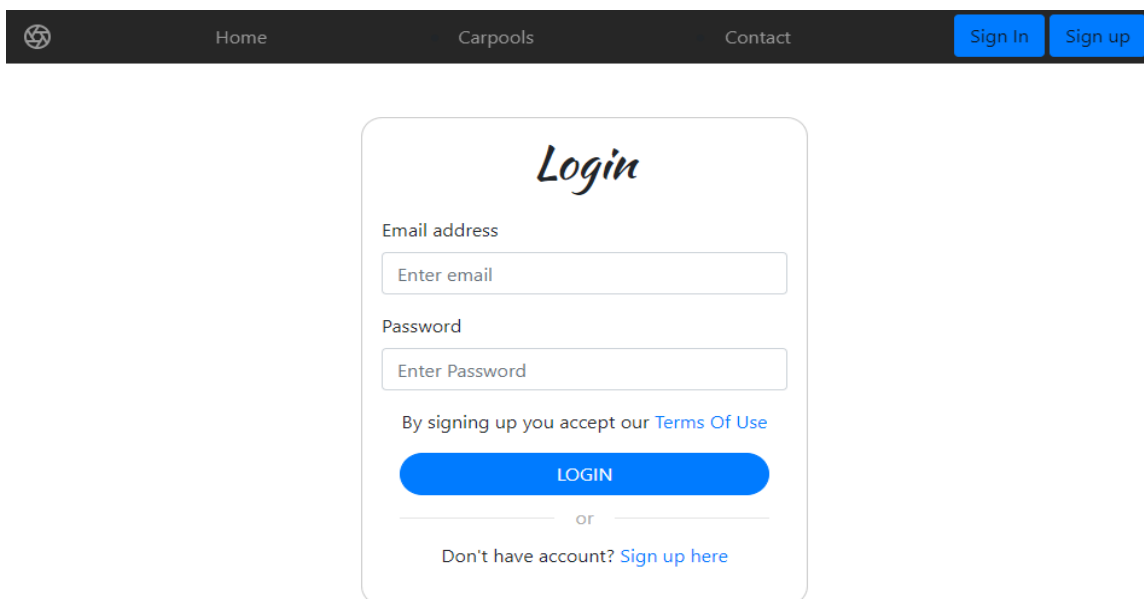


Figure 27:Page d'accueil

Cette fenêtre c'est notre accueil elle se compose de plusieurs Buttons chaque une d'elle nous achemine vers une interface par exemple le formulaire d'identification qui va être représenté dans le point suivant.

IV.3.2 Formulaire d'identification :

L'utilisateur peut connecter en introduisant son Email adress et son Password.

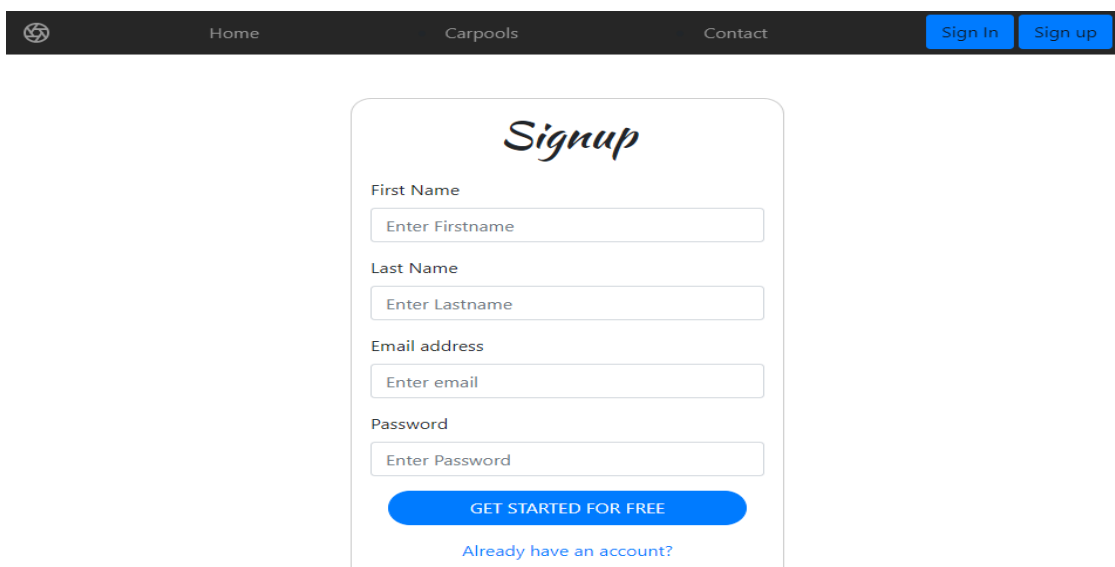


The screenshot shows a dark navigation bar at the top with a logo on the left and links for 'Home', 'Carpools', and 'Contact'. On the right side of the bar are two blue buttons labeled 'Sign In' and 'Sign up'. Below the navigation bar is a white rounded rectangle containing the 'Login' form. The form has the title 'Login' in a cursive font. It includes an 'Email address' field with the placeholder 'Enter email', a 'Password' field with the placeholder 'Enter Password', and a link for 'Terms Of Use'. A prominent blue button labeled 'LOGIN' is centered below the fields. Underneath the button is the text 'or' and a link 'Don't have account? Sign up here'.

Figure 28:Formulaire d'identification

En cas ou l'utilisateur ne possède pas un compte à notre niveau il doit s'inscrire pour avoir la connexion l'interface suivant représente cet aspect.

Inscription sur notre site :



The screenshot shows the same dark navigation bar as in Figure 28. Below it is a white rounded rectangle containing the 'Signup' form. The form has the title 'Signup' in a cursive font. It includes four input fields: 'First Name' (placeholder 'Enter Firstname'), 'Last Name' (placeholder 'Enter Lastname'), 'Email address' (placeholder 'Enter email'), and 'Password' (placeholder 'Enter Password'). A prominent blue button labeled 'GET STARTED FOR FREE' is centered below the fields. Below the button is a link 'Already have an account?'.

Figure 29:Inscription sur notre site

IV.3.4 Ajouter Trajet :

IV.3.4.1 Chargement les informations de trajet :

Quand l'utilisateur se connecte il a tous les droits. Cette partie représente l'interface pour rajouter un trajet.

+ Add Trip
Fill the below information to add a trip:

My way

Point depart..
Search Depart from google map

Point Arrive..
Search Destination from google map

Do you want to play Music

Yes
 No

There is any gosra in your trip

Yes
 No

Can people smoke in your trip

Yes
 No

Your price for the trip

Price

Number of places for the trip

place

Bugs allowed for the trip

Bugs

Your phone number

Phone number

Ride Date

Description

Save Cancel

© 2019-2020

Features	Resources	Resources	About
Cool stuff	Resource	Business	Team
Random feature	Resource name	Education	Locations
Team feature	Another resource	Government	Privacy
Stuff for developers	Final resource	Gaming	Terms
Another one			

Figure 30:Chargement les informations de trajet

IV.3.4.2 Chargement la Localisation de trajet :

Dans l’interface de rajoutèrent l’utilisateur peut rechercher un point de départ ou un point d’arrive en utilisant le service de google map, sinon il tape un.

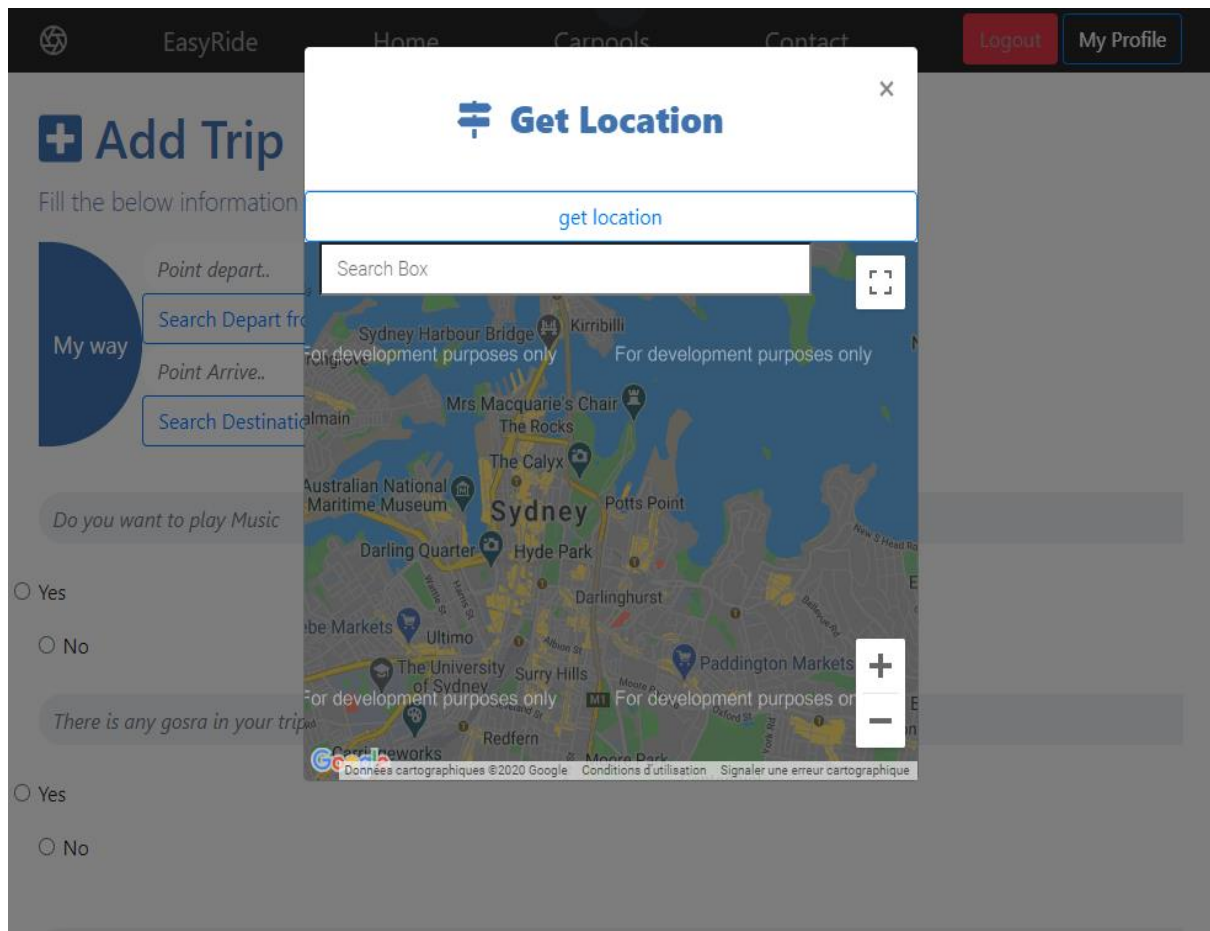


Figure 31:Chargement la Localisation de trajet

IV.3.5 Modifier le Trajet:

Après le rajoutement l'utilisateur peut changer les informations de trajet si il y a un changement il peut modifier.

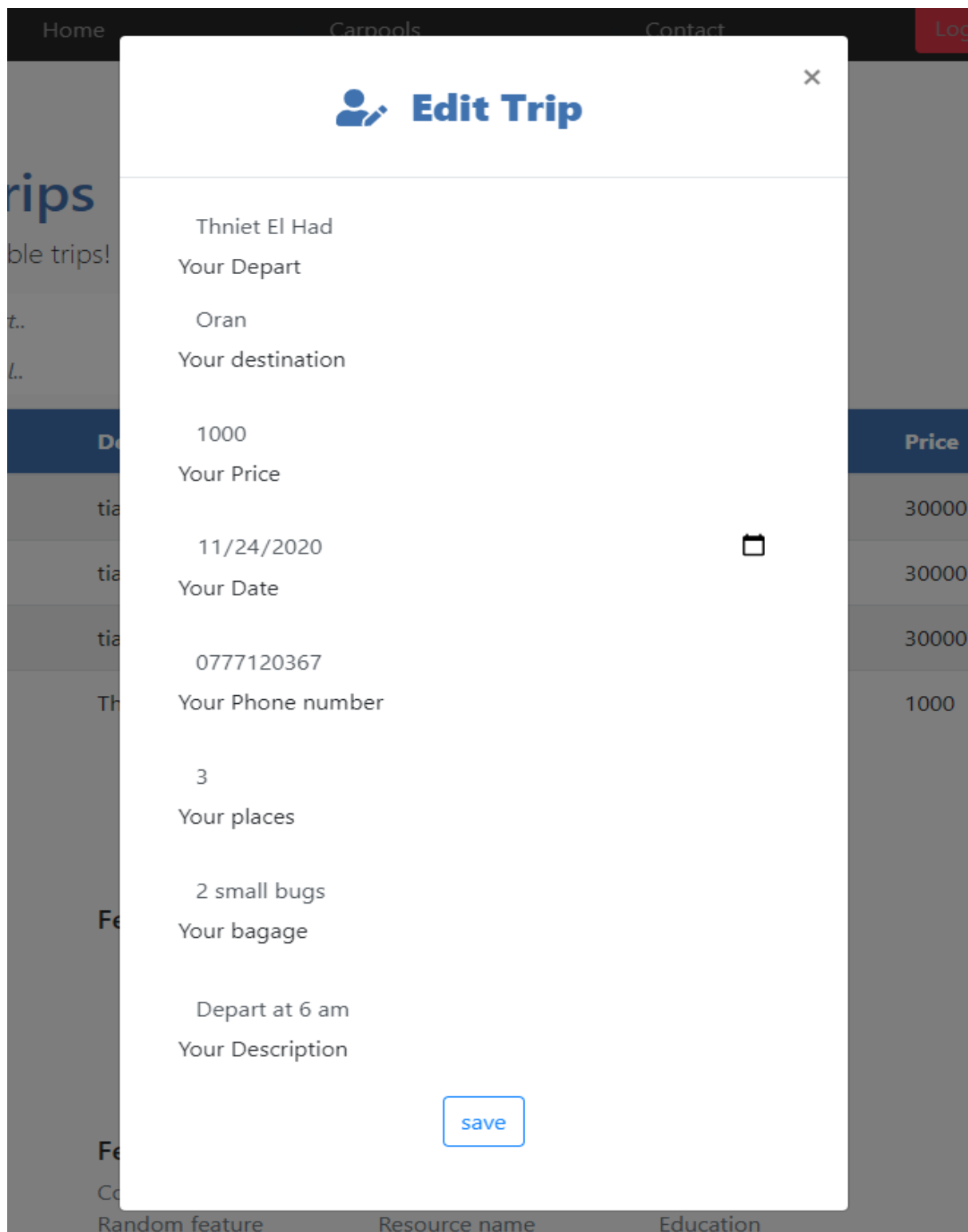


Figure 32:Modifier le Trajet

IV.3.6 Supprimer le Trajet:

L'utilisateur peut aussi supprimer son trajet.

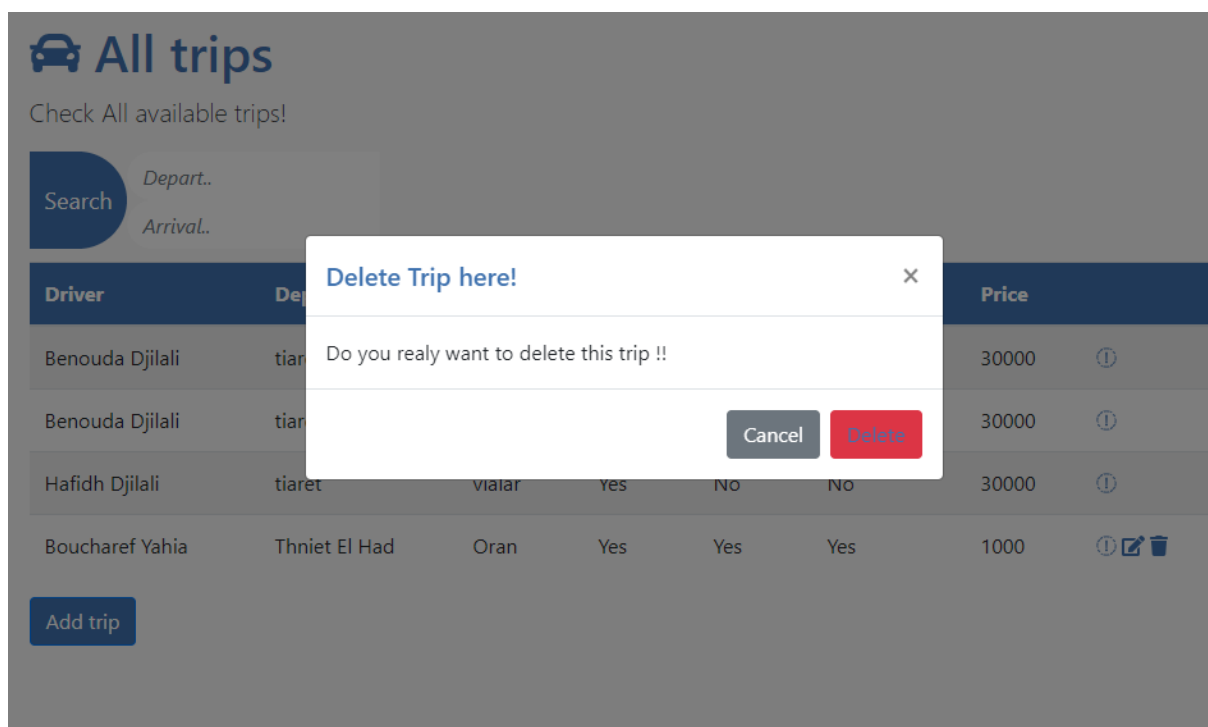


Figure 33:Supprimer le Trajet

IV.3.7 L'utilisateur Consulter son profile :

L'utilisateur peut aussi consulter son profile.

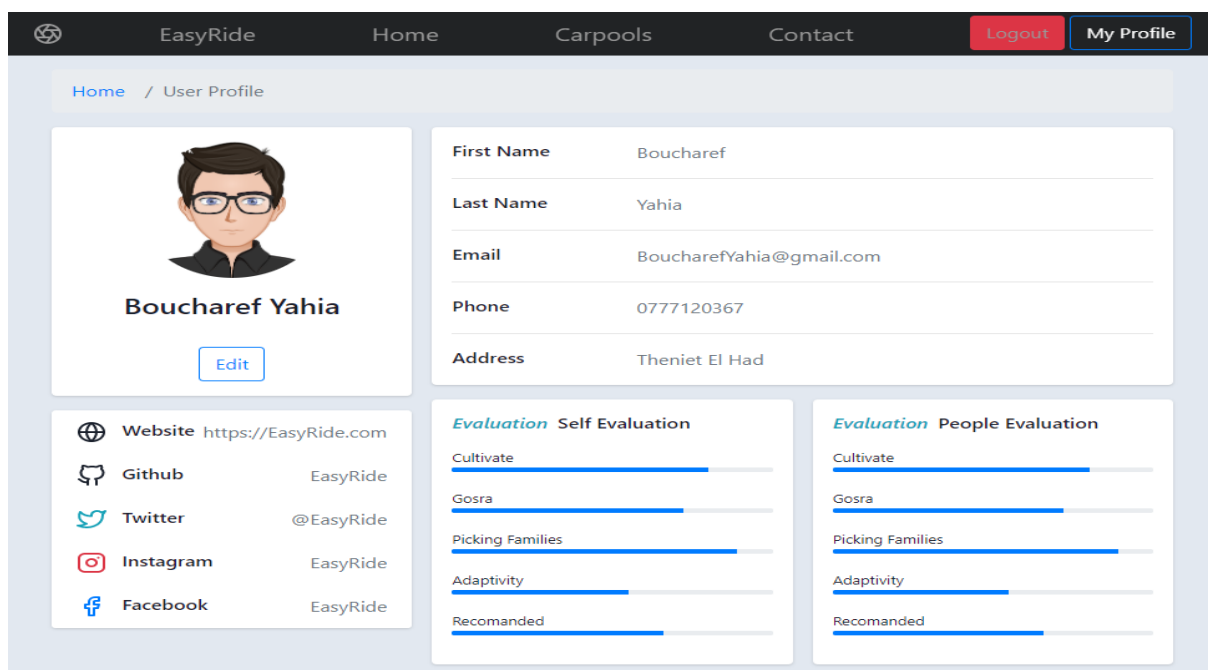


Figure 34:Le Conducteur Consulter son profile

IV.3.8 Les détaille de trajet:

Cette interface représente tous les détails d'un trajet.

The screenshot displays the 'Who's the driver' section of the EasyRide app. At the top, a navigation bar contains 'EasyRide', 'Home', 'Carpools', 'Contact', 'Logout', and 'My Profile'. Below this, the driver's profile is shown with a circular image of a car interior and the text 'Driver name: Boucharef Yahia'. A light gray box contains two date entries: 'Adding Date: 2020-11-18' and 'Trip Date: 2020-11-24'. The 'General information' section lists: 'Depart: Thniet El Had', 'Arrival: Oran', 'Music: Yes', 'Gosra: Yes', 'Smoking: Yes', 'Price: 1000 DA', 'Luggage: 2 small bugs', and 'Number of seats: 3'. The 'Detailed information' section shows 'Phone number: 0777120367'. Below this is a description box with the text 'Description' and 'Depart at 6 am'. At the bottom of the main content area are two buttons: 'GetRide' and 'Back'. The footer of the page contains a copyright notice '© 2019-2020' and five columns of links: 'Features' (Cool stuff, Random feature, Team feature, Stuff for developers, Another one, Last time), 'Resources' (Resource, Resource name, Another resource, Final resource), 'Resources' (Business, Education, Government, Gaming), and 'About' (Team, Locations, Privacy, Terms).

Figure 35: Les détaille de conducteur avec son trajet

IV.3.9 Consulter tous les covoiturage :

Cette interface represente la liste des trajets.

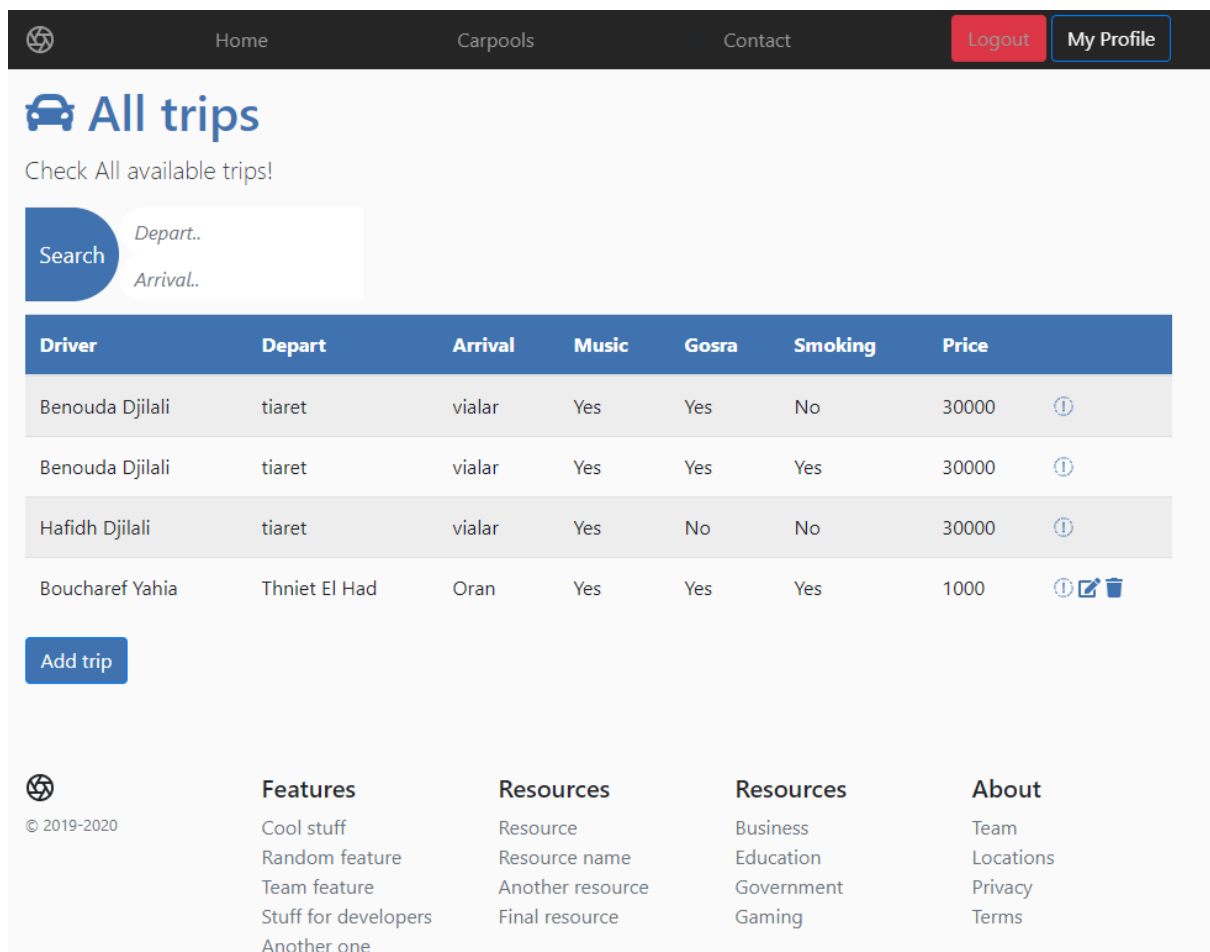


Figure 36: Consulter tous les covoiturages

IV.3.10 Contacter les Administrateur :

Si l'utilisateur a une envie de contacter les administrateurs il peut formuler pour nous avoir

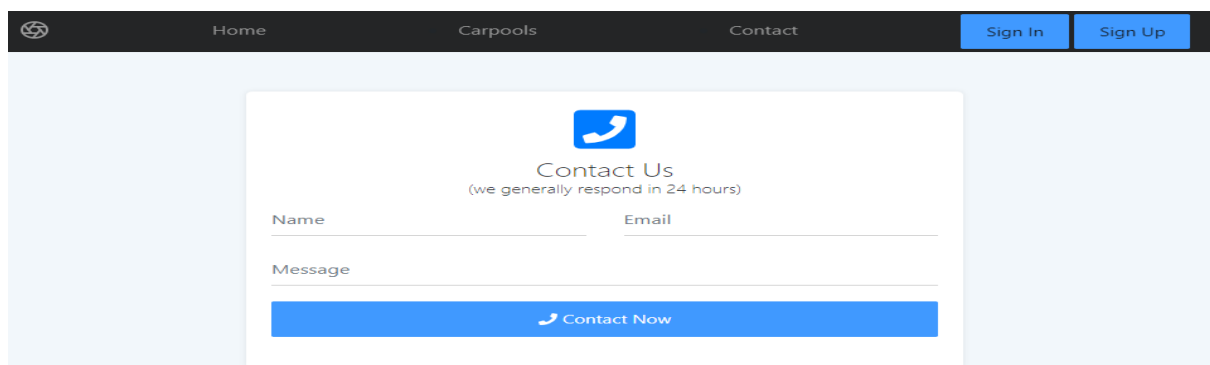


Figure 37: Contacter les Administrateur

IV.4 Les Avantages de notre application web:

- **Entièrement réactif**

Le thème de notre application web aura fière allure sur n'importe quel appareil, quelle que soit sa taille.

- **Fournit des services**

Nous essayons d'aider tout le monde tout en économisant du temps et des efforts aux les conducteurs et les passagers.

- **Facile à utiliser**

Opération simple que n'importe qui peut utiliser et ajouter votre covoiturage.

- Gestion en temps réel d'un ensemble d'offres de covoiturage envoyées par les conducteurs et un ensemble de demande émises directement par les passagers.

IV.5 Conclusion :

Dans ce chapitre, nous allons aborder la partie pratique de notre projet qui consiste à exploiter le coter technique de notre application web propose. Nous allons définir les différents outils et les langages utilise, en puis présenter les interfaces de notre application web, en fin, les avantages de notre application web.

Conclusion générale

Conclusion générale :

Ce projet fait l'objet de la conception et la réalisation d'une application web Java EE pour la gestion du covoiturage en Algérie. Il nous a permis de développer une application web de simulation pour la gestion du covoiturage en Algérie.

En premier lieu, nous avons commencé par une étude théorique du problème à aborder.

Dans le premier chapitre, nous avons vu le Modèle-Vue-Contrôleur (MVC), leur principe et architecture, Les Composants et l'architectures de Java EE et les API et services Java EE.

Dans le deuxième chapitre, nous avons vu la définition et le principe de covoiturage, les formes de covoiturage spontané et organisée, Les systèmes opérationnels du covoiturage dans le cas statique et dynamique, les avantages et les problèmes techniques de covoiturage.

Dans le troisième chapitre, nous définir la notion de covoiturage, Charte du covoiturage, la spécification des exigences, l'analyse de besoins et la conception de notre projet.

Dans l'étape de réalisation, nous avons sélectionné les technologies les plus adaptées à notre choix technique, pour finir par une illustration dès notre application web de de la conception et la réalisation d'une application web Java EE pour la gestion du covoiturage, nous avons eu l'opportunité de nous familiariser avec les outils récents telles que, Maven, Java Développement kit, IntelliJ IDEA, Xamp, Bootstrap, Springboot et les langages Java EE, JavaScript, HTML, CSS, SQL pour développer notre application.

Notre application web est Offre une Entièrement réactif pour fière allure sur n'importe quel appareil, quelle que soit sa taille, Fournit des services d'aider tout le monde, tout en économisant du temps et des efforts ou les conducteurs et les passagers doivent les utiliser facilement.

Comme perspectives de ce travail, nous proposons pour permettre de disposer le système de gestion de covoiturage avec des solutions de recommandation basée sur les algorithmes génétiques.

Bibliographie :

- [1] - Raphaël Fournier-S'niehotta et Philippe Rigaux, Applications orientées données (NSY135), Modèle-Vue-Contrôleur (MVC), Département d'informatique, Conservatoire National des Arts & Métiers, Paris, France.
- [2] - Taha Zerrouki, Approche MVC Model-View-Controller, Université de Bouira Département Informatique – Matser Génie des systèmes Informatiques Méthodes et Technologies de l'implémentation, 2002
- [3] - Stéphane HUOT, Applications interactives, Modèle Vue-Contrôleur (MVC), 2009
- [4] - <https://mediaserver.cnam.fr/permalink/v125f3585f56d5y921cu/>
- [5] - Jean-Marc Lorach, « *Le covoiturage, une alternative solidaire aux transports en commun* » [archive], sur [Le Figaro](#), 29 janvier 2009.
- [6] - Barème kilométrique, calcul des frais et indemnités, 2015.
- [7] - Arrêt de la Cour de Cassation, 2013.
- [08] - Avis de la Cour de cassation sur service-public.fr.
- [09] - *Code des transports - Article L3132-1*.
- [10] - Mauss Marcel, « Essai sur le don. Forme et raison de l'échange dans les sociétés primitives » in L'Année Sociologique, seconde série, Paris, 1923-1924.
- [11] Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publiques (France), « *Étude sur les obstacles juridiques au développement des nouveaux services de transport P45* ».
- [12] – Jean-Christophe Ballet (Université Lyon II) Robert Clavel (Certu), Le covoiturage en France – État des lieux et perspectives, octobre 2007.
- [13] - ICARO : Recommandations techniques pour la mise en oeuvre d'une politique de covoiturage, projet européen, 1999.
- [14] – <https://www.blablacar.fr/>
- [15] - Connie Loizos, « *Handcuffed to Uber* », TechCrunch, 29 avril 2016 ([lire en ligne](#), consulté le 8 mai 2016).

- [16] - *Uber valorisé 40 milliards de dollars* », sur *journaldunet*, 5 décembre 2014.
- [17] - Valerie De Senneville, « *Uber : dans les coulisses d'une machine de guerre juridique* », sur *Les Échos*, 12 octobre 2015 (consulté le 26 octobre 2015).
- [18] - <https://yassir.com/>
- [19] - <http://www.avego.com/ui/index.action>
- [20] - <http://comuto.com.webs.co.com/>
- [21] - Sidawy, E. Le covoiturage dynamique sort de l'impasse. available : <http://www.greencode.fr/2011/11/le-covoiturage-dynamique-sort-de-l>.
- [22] – Usages et attitudes des utilisateurs du site internet covoiturage.fr. [online]. available : <https://www.maif.fr/content/pdf/particuliers/automoto/covoiturage/maif-etude-covoiturage-12-2009.pdf>
- [23] - ademe.fr eco-d'éplacements calculette
- [24] - BlaBlaCar éditeurs de covoiturage.fr
- [25] – Sondes Ben Cheikh. Optimisation avancée au service du covoiturage dynamique. Automatique. Ecole Centrale de Lille, Français, 2016.
- [26] - S.Seyedabrishami, A.Mamdoohi, A.Barzegar, and S.Hasanpour. Impact of carpooling on fuel saving in urban transportation case study of tehran. In *Procedia Social and Behavioral Sciences*. 2012.
- [27] – <http://www.ecolutis.com/le-covoiturage/25-covoiturage/45-economique>
- [28] - Covoiturage : dur de changer ses habitudes. [online]. available : <http://nouvelles-mobilites.net/2010/04/08/covoiturage-dur-de-changer-seshabitudes/>.2010.
- [29] - Développer une infrastructure de covoiturage, Juin 2010.
- [30] - Développer une infrastructure de covoiturage, 2010. available : <https://www.google.fr/url?sa=trct=jq=esrc=ssource=webcd=1ved=0cb0qfjaaahukewj239g8qy3jah>.
- [31] - [https://en.wikipedia.org/wiki/Maven_\(disambiguation\)](https://en.wikipedia.org/wiki/Maven_(disambiguation))

[32] - [Oracle Technology Network Java Overview.](#)

[33]- <http://blog.jetbrains.com/idea/2017/02/intellij-idea-2017-1-eap-extends-debugger-with-async-stacktraces/>

[34] – [«https://www.apachefriends.org/blog/new_xampp_20190814.html»](https://www.apachefriends.org/blog/new_xampp_20190814.html)

[35] - [https://fr.wikipedia.org/wiki/Bootstrap_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework))

[36] - <https://arabicprogrammer.com/article/6662208139/>

[37] - <https://www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.htm>

[38] - <http://glossaire.infowebmaster.fr/html/>

[39] - <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203277-css-cascading-style-sheets-definition-traduction/>

[40] - <https://fr.scribd.com/document/3749571/penser-java>

[41] - https://nanopdf.com/download/sgbd-mysql_pdf