

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERRI de Tizi-Ouzou
Faculté de Génie Electrique et d'Informatique
Département d'informatique

MEMOIRE DE FIN D'ETUDE

En vue de l'obtention d'une licence en informatique LMD

Conception et réalisation d'un site web de Covoiturage

Encadré par
Mr M.OUAMRANE

Réalisé par
ARIB Yassine
BOUIZEGARENE Lyna

Promotion 2014/2015



Remerciements

Nous tenons à remercier vivement notre promoteur Mr M. OUAMRANE qui nous a fait confiance et s'est mis à notre entière disposition pour le parfait déroulement de notre travail.

Nous remercions ensuite tous les enseignants de notre cursus, en particulier ceux du département d'Informatique de l'université MOULOUD MAMMERI.

Nous remercions particulièrement nos familles respectives qui nous ont toujours soutenus, qu'ils trouvent ici l'expression de nos sentiments de reconnaissance et de respect. Sans oublier l'équipe Inventeam pour leurs précieux conseils et nos cher(e)s ami(e)s Celia, Dassine, Mohammed, Nacim, Mounir, Ghiles, Juba, Nadir, Ali et Mouloud.

Que les membres du jury trouvent ici nos remerciements les plus vifs pour avoir accepté d'honorer par leur jugement notre travail que nous avons réalisé avec passion.

Sommaire

Introduction générale	12
-----------------------	----

Chapitre I : Généralités sur les technologies Web

1.	Introduction	14
2.	Définition d'un réseau	14
3.	Composants d'un réseau	14
4.	Classification des réseaux	15
5.	Normes et protocoles réseau	16
5.1.	Avantages des modèles de référence :.....	16
5.2.	Modèle de référence OSI :	17
5.3.	Le modèle de protocole TCP/IP :.....	18
6.	Internet	18
6.1.	Définition :	19
6.2.	Intranet :	19
6.3.	Extranet :.....	19
7.	Le Web :	20
7.1.	Définition :	20
7.2.	Les concepts du web	20
8.	Architecture Client-serveur	21
8.1.	Définition :	21
8.2.	Caractéristiques.....	21
8.2.1.	Caractéristiques d'un processus serveur :	21
8.2.2.	Caractéristiques d'un processus client :	22
8.3.	Classification des architectures client/serveur :	22
9.	Objectif des réseaux :.....	23
10.	Conclusion.....	23

Chapitre II: Le Covoiturage

1.	Introduction :.....	25
2.	Définition :	25
3.	Le principe :	25
4.	Historique :	26
5.	A qui s'adresse-t-il ?.....	26

6.	Les types de covoiturages	27
7.	Avantages et Inconvénients du covoiturage :	27
8.	Le covoiturage dynamique	28

Chapitre III : Analyse et Conception

1.	Introduction	30
2.	Problématique et expression initiale des besoins	30
3.	Positionnement et objectif de l'application	30
4.	Analyse et conception:.....	31
4.1.	Analyse :.....	31
4.1.1.	Identification des acteurs	31
4.1.2.	Spécification des tâches	31
4.1.3.	Spécification des scénarios	32
4.1.4.	Diagrammes des cas d'utilisation	34
4.2.	Conception.....	35
4.2.1.	Diagrammes de Séquence	35
4.2.1.1.	Diagramme de séquence détaillée du cas d'utilisation « Créer un compte » : ..	35
4.2.1.2.	Diagramme de séquence détaillée du cas d'utilisation « S'inscrire à un covoiturage » :	36
4.2.1.3.	Diagramme de séquence détaillé du cas d'utilisation «Faire une recherche » :37	
4.2.1.4	Diagramme de séquence système du cas d'utilisation « S'authentifier » :	38
4.2.1.5	Diagramme de séquence système du cas d'utilisation « Publier un covoiturage » :	39
4.2.1.6	Diagramme de séquence système du cas d'utilisation « Editer son profil» :	40
4.2.1.7	Diagramme de séquence système du cas d'utilisation « Commenter un covoiturage » :	41
4.2.1.8	Diagramme de séquence système du cas d'utilisation « Se déconnecter» :	41
4.2.1.9	Diagramme de séquence système du cas d'utilisation « Consulter mes covoiturages » :	42
4.2.1.10	Diagramme de séquence système du cas d'utilisation « Accepter ou refuser des passagers » :	43
4.2.1.11	Diagramme de séquence système du cas d'utilisation « Supprimer un covoiturage » :	44

4.2.1.12	Diagramme de séquence système du cas d'utilisation « Annuler une réservation » :	45
4.2.2	Diagrammes de classes	46
4.2.2.1	Diagramme des classes général	46
4.2.2.2	Diagramme des classes des contrôleurs:.....	47
4.2.2.3	Diagramme des classes des modèles	48
4.2.3	Diagrammes de navigation	49
4.2.3.1	Diagramme global de navigation pour l'internaute	49
4.2.3.2	Diagramme de navigation de la Recherche de Covoiturage	50
4.2.3.3	Diagramme de navigation de la Consultation de Profil	50
4.2.3.4	Diagramme de navigation de la Consultation de Covoiturage	51
4.2.3.5	Diagramme de navigation de la page d'accueil	52
4.2.3.6	Diagramme de navigation de la page Mes Covoiturages	53
4.2.4	Implémentation de bases de données	54
4.2.4.1	Schéma relationnel	54
4.2.4.2	Structure des tables	55
	Structure de la table covoiturages.....	55
	Structure de la table villes	55
	Structure de la table notes	56
	Structure de la table notifications	56
	Structure de la table user_covoiturage_inscrits	56
	Structure de la table user_covoiturage_preinscrits	57
	Structure de la table commentaires	57
	Structure de la table users	57

Chapite IV: Mise en œuvre et réalisation

Introduction

I. Outils et Technologies

1.	PHP : Hypertext Preprocessor	60
1.1	Présentation :	60
1.2	Fonctionnement :	61
2.	HTML5 et CSS3.....	61

2.1	HTML	61
2.2	HTML5	62
2.3	CSS	62
3.	JavaScript	62
4.	SQL	62
4.1	MySQL	63
5.	PhpStorm	63
6.	Git et GitHub	64
7.	Composer	65
II. Outils et Technologies		
1.	Définition d'un framework	65
2.	LARAVEL	65
2.1	Le modèle MVC	67
2.1.1	Présentation :	67
2.1.2	Rôles des composants :	67
2.1.3	Interaction entre les composants :	67
2.1.4	Avantages et inconvénients :	68
2.2	La programmation orientée objet	69
2.3	Le routage	69
2.4	L'ORM Eloquent	69
2.4.1	Les avantages	70
2.4.2	Un simple exemple	70
2.5	Le moteurs de templates Blade	71
2.5.1	Définition :	71
2.5.2	Avantages :	71
2.6	Les middlewares	72
3.	Sécurité	72
3.1	Les injections SQL	72
3.2	Hachage de mots de passe	73
3.3	Filtrage et validation des données	73
3.4	Validation	74
3.5	Rapport d'erreurs	74

4. Bootstrap.....	74
4.1. Quel en est l'intérêt ?	75
4.2. Responsive Web Design.....	76
4.3. Le principe du responsive web design.....	76
5. jQuery	77
6. API Google MAP.....	77
III. Interfaces	78
Conclusion générale	90
Annexe	92
Bibliographie et Webographie	99

Liste des Figures

Figure 1 .1 Composants d'un réseau.....	14
Figure 1. 2 Réseau LAN et réseau WAN	15
Figure 1. 3 Modèles réseau	16
Figure 1. 4 Modèle OSI.....	17
Figure 1. 5 Modèle TCP/IP.....	18
Figure 1. 6 Internet, le réseau des réseaux.....	19
Figure 1. 7 Internet, Intranet et Extranet.....	20
Figure 1. 8 Exemple d'architecture Client/ Serveur.....	21
Figure 1. 9 Quelques objectifs en images.....	23
Figure 2. 1 Panneau d'indication de covoiturage en Europe	25
Figure 2. 2 Nombre de véhicules selon le nombre d'occupant.....	28
Figure 3. 1 Diagramme des cas d'utilisation général.....	34
Figure 3. 2 Diagramme de séquence détaillée du cas d'utilisation « Créer un compte ».....	35
Figure 3. 3 Diagramme de séquence détaillée du cas d'utilisation « S'inscrire à un covoiturage».....	36
Figure 3. 4 Diagramme de séquence détaillé du cas d'utilisation «Faire une recherche »	37
Figure 3. 5 : Diagramme de séquence système du cas d'utilisation «S'authentifier »	38
Figure 3. 6 Diagramme de séquence système du cas d'utilisation « Publier un Covoiturage ».....	39
Figure 3. 7 Diagramme de séquence système du cas d'utilisation « Editer son profil »	40
Figure 3. 8 Diagramme de séquence système du cas d'utilisation « Commenter un covoiturage ».....	41
Figure 3. 9 Diagramme de séquence système du cas d'utilisation «Se déconnecter».....	41
Figure 3. 10 Diagramme de séquence système du cas d'utilisation « Consulter mes covoiturages » ..	42
Figure 3. 11 Diagramme de séquence système du cas d'utilisation « Accepter ou refuser des passagers »	43
Figure 3. 12 Diagramme de séquence système du cas d'utilisation « Supprimer un covoiturage ».....	44
Figure 3. 13 Diagramme de séquence système du cas d'utilisation « Annuler une réservation».....	45
Figure 3. 14 Diagramme des classes général.....	46
Figure 3. 15 Diagramme des classes des contrôleurs	47
Figure 3. 16 Diagramme des classes des modèles.....	48
Figure 3. 17 Diagramme global de navigation pour l'internaute	49
Figure 3. 18 Diagramme de navigation de la Recherche de Covoiturage	50
Figure 3. 19 Diagramme de navigation de la Consultation de Profil	50
Figure 3. 20 Diagramme de navigation de la Consultation de Covoiturage.....	51
Figure 3. 21 Diagramme de navigation de la page d'accueil.....	52
Figure 3. 23 Schéma relationnel	54
Figure 3. 22 Diagramme de navigation de la page Mes covoiturage.....	54
Figure 4. 1 Coopération HTML/CSS.....	61
Figure 4. 2 Interface de l'IDE PhpStorm.....	64
Figure 4. 3 Interactions entre les composants MVC.....	68

Figure 4. 4 Structure d'une route	69
Figure 4. 5 Schématisation d'un middleware.....	72
Figure 4. 6 Exemple de page adaptative.....	76
Figure 4. 7 Interface « Accueil » pour mobile	79
Figure 4. 8 Interface « Accueil » pour un membre authentifié	80
Figure 4. 9 Interface « S'inscrire »	81
Figure 4. 10 Interface « Page détaillée ».....	82
Figure 4. 11 Interface « Comment ça marche »	83
Figure 4. 12 Interface « Publier »	84
Figure 4. 13 Interface «Connexion ».....	84
Figure 4. 14 Interface « Mes Covoiturages ».....	85
Figure 4. 15 Interface « Profil »	85
Figure 4. 16 Interface «Mes covoiturages» en format mobile.....	86
Figure 4. 17 Interface « Profil » en format mobile	86
Figure 4. 18 Interface « Résultats de Recherche»	87
Figure A. 1 Représentation graphique d'une classe.	94
Figure A. 2 Relation d'association.....	95
Figure A. 3 Relation de généralisation	96
Figure A. 4 Exemple du diagramme de séquence.....	97
Figure A. 5 Exemple du diagramme de navigation	98

Liste des Tableaux

Tableau 1: Tableau spécifiant les tâches de chaque acteur.....	31
Tableau 2 : Tableau spécifiant les scénarios de chaque tache	33
Tableau 3: Structure de la table covoiturages	55
Tableau 4 : Structure de la table villes.....	55
Tableau 5: Structure de la table notes.....	56
Tableau 6 : Structure de la table notifications	56
Tableau 7 : Structure de la table user_covoiturage_inscrits.....	56
Tableau 8 : Structure de la table user_covoiturage_preinscrits	57
Tableau 9: Structure de la table commentaires.....	57
Tableau 10 : Structure de la table users	58

Introduction Générale

Aujourd'hui, l'utilisation de la technologie pour développer et renforcer notre capacité de communication arrive à un tournant. La généralisation de l'utilisation d'Internet à l'échelle mondiale s'est opérée plus vite que quiconque aurait pu l'imaginer.

L'évolution rapide de ce réseau mondial induit un bouleversement des interactions sociales, commerciales, politiques et même personnelles. L'étape suivante de notre développement verra les novateurs se servir d'Internet comme d'un tremplin pour créer de nouveaux produits et services spécialement conçus pour exploiter les capacités des réseaux.

Alors que les développeurs repoussent les limites de ce qu'il est possible d'accomplir, les capacités des réseaux interconnectés qui forment Internet sont appelées à jouer un rôle croissant dans le succès de ce projet.

Notre travail consiste à développer un site web pour les personnes souhaitant avoir recours au covoiturage comme moyen de transport en utilisant les technologies récentes du développement Web.

Chapitre I

Généralités sur les technologies Web

1. Introduction

Ce chapitre présente la plateforme de réseaux de données dont nos relations sociales et commerciales sont de plus en plus dépendantes. Nous aborderons notamment les différents services offerts par internet parmi eux le web qui sera l'objet de quatrième chapitre dans laquelle nous développerons quelques points importants tels que les concepts et langages du web.

2. Définition d'un réseau

Un réseau informatique est un ensemble de matériels et de logiciels. Souvent il représente plusieurs équipements (ordinateurs, imprimantes, HUB, routeurs..) interconnectés par des câbles ou sans fils. L'infrastructure réseau constitue la plateforme qui prend en charge le réseau. Elle fournit le canal stable et fiable à travers lequel nos communications peuvent s'établir.

3. Composants d'un réseau

Le chemin emprunté par un message depuis une source jusqu'à une destination peut être aussi simple que la connexion entre deux ordinateurs via un seul câble ou aussi complexe qu'un réseau parcourant le globe terrestre.

L'infrastructure réseau comprend trois catégories de composant réseau :

- Les périphériques
- Les supports
- Les services

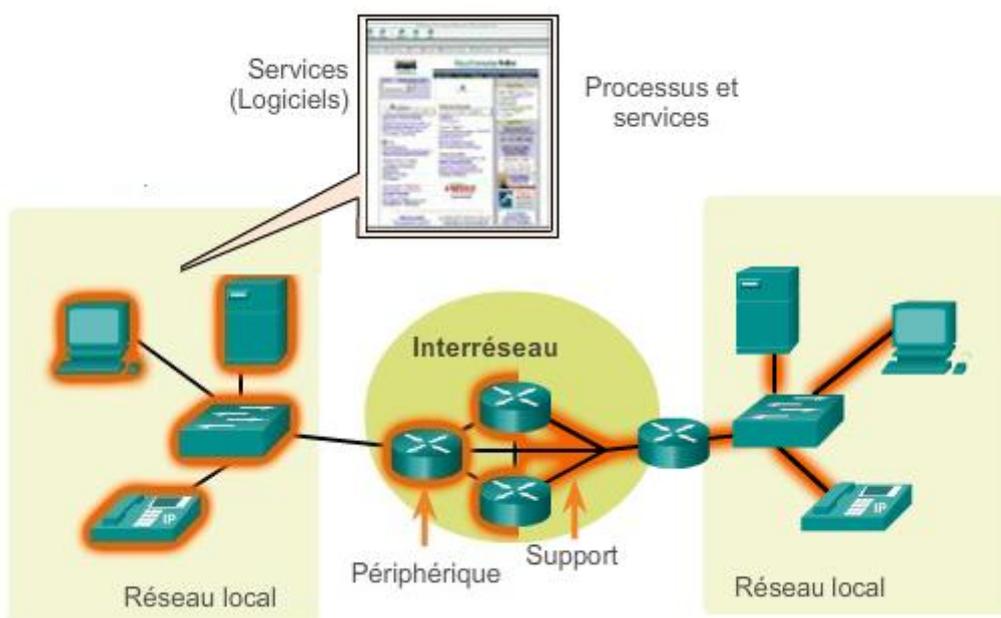


Figure 1 .1 Composants d'un réseau

Le matériel correspond souvent aux composants visibles de la plateforme réseau, par exemple un ordinateur portable, un ordinateur de bureau, un commutateur, un routeur, un point d'accès sans fil ou le câblage qui sert à relier les périphériques.

Parfois, certains composants ne sont pas visibles. Dans le cas d'un support sans fil, les messages sont transmis à travers l'air, à l'aide d'une fréquence radio ou d'ondes infrarouges invisibles.

Les services et les processus sont les programmes de communication, appelés logiciels, qui sont exécutés sur les périphériques réseau. Les processus fournissent les fonctionnalités qui dirigent et déplacent les messages à travers le réseau.

4. Classification des réseaux

Classification selon la taille :

- **Réseau local (LAN)** - infrastructure réseau permettant d'accéder aux périphériques finaux et aux utilisateurs sur une zone peu étendue. par exemple d'une habitation particulière, d'une entreprise, d'une salle informatique, d'un bâtiment. L'infrastructure est privée et est gérée localement.
- **LAN sans fil (WLAN)** - infrastructure similaire à un réseau local, mais sans fil. Elle relie des utilisateurs et des terminaux situés dans une zone peu étendue.
- **Réseau étendu (WAN)** - infrastructure réseau permettant d'accéder à d'autres réseaux sur une vaste zone. **Technologies utilisées :** Câble, fibre optique, satellite, technologie sans fil 3G et ondes hertziennes.
- **Réseau métropolitain (MAN)** - infrastructure réseau qui couvre une zone plus vaste qu'un LAN, mais moins étendue qu'un WAN (par exemple, une ville). Les MAN sont généralement gérés par une seule entité, comme une grande entreprise.
- **PAN (Personal Area Network)** - Désignent des réseaux restreints d'équipements informatiques habituellement utilisés dans le cadre d'une activité personnelle. Les bus les plus utilisés sont l'USB, les technologies sans fil telles que Bluetooth ou IR (infra rouge).

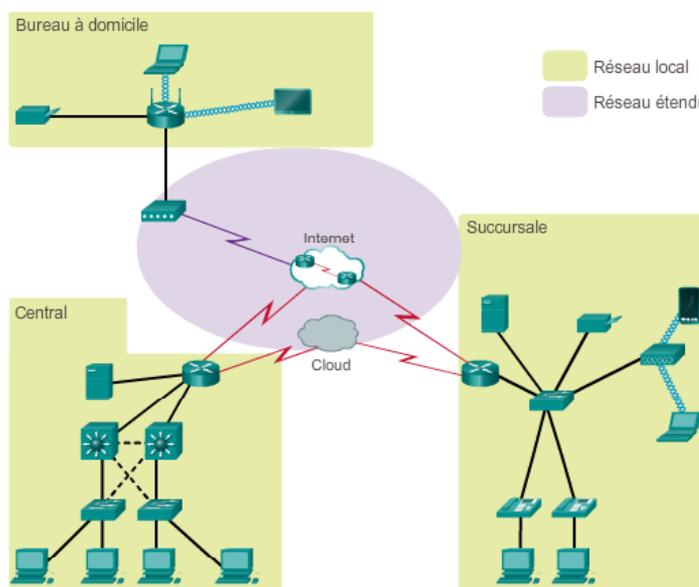


Figure 1. 2 Réseau LAN et réseau WAN

5. Normes et protocoles réseau

5.1. Avantages des modèles de référence :

On utilise souvent un modèle sous forme de couches, tel que le modèle TCP/IP pour aider à visualiser l'interaction entre les différents protocoles. Ce modèle illustre le fonctionnement des protocoles intervenant dans chaque couche, ainsi que leur interaction avec les couches supérieures et inférieures. L'utilisation d'un modèle en couches présente certains avantages pour décrire des protocoles et des opérations sur un réseau :

- Aide à la conception d'un protocole, car des protocoles qui fonctionnent à un niveau de couche spécifique disposent d'informations définies à partir desquelles ils agissent, ainsi que d'une interface définie par rapport aux couches supérieures et inférieures.
- Il permet d'éviter que des changements technologiques ou fonctionnels dans une couche ne se répercutent sur d'autres couches, supérieures et inférieures.
- Il fournit un langage commun pour décrire les fonctions et les fonctionnalités réseau.

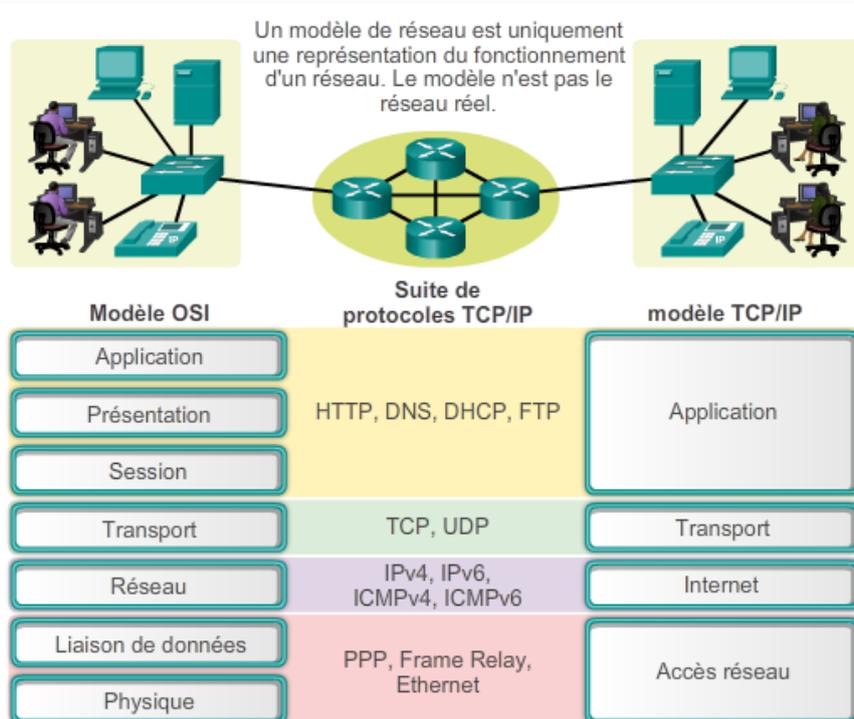


Figure 1. 3 Modèles réseau

5.2. Modèle de référence OSI :

Le modèle OSI (Open System Interconnection) est le modèle de référence interréseau le plus connu. Il est utilisé pour la conception de réseaux de données, des spécifications d'opérations et la résolution des problèmes.

- **La couche Physique** : Les protocoles de la couche physique décrivent les moyens mécaniques, électriques, fonctionnels et méthodologiques permettant d'activer, de gérer et de désactiver des connexions physiques pour la transmission de bits vers et depuis un périphérique réseau.
- **La couche liaison de données** : Les protocoles de couche liaison de données décrivent des méthodes d'échange de trames de données entre des périphériques sur un support commun.
- **La couche réseau** : fournit des services permettant d'échanger des parties de données sur le réseau entre des périphériques finaux identifiés.
- **La couche transport** : définit des services pour segmenter, transférer et réassembler les données de communications individuelles entre les périphériques finaux.
- **La couche session** : fournit des services à la couche présentation pour organiser son dialogue et gérer l'échange de données.
- **La couche présentation** : fournit une représentation commune des données transférées entre des services de couche application.
- **La couche application** : permet d'obtenir une connectivité de bout en bout entre des individus du réseau humain à l'aide de réseaux de données.



Figure 1. 4 Modèle OSI

5.3. Le modèle de protocole TCP/IP :

Le modèle de protocole TCP/IP pour les communications interréseau fut créé au début des années 1970 et est appelé modèle Internet. Il définit quatre catégories de fonctions qui doivent intervenir pour que les communications aboutissent.

- **La couche accès réseau** : Elle « regroupe » les couches physiques et liaisons de données du modèle OSI. Sa seule contrainte est de permettre à un hôte d'envoyer des paquets IP sur le réseau. Elle peut, pour cela utiliser presque toute interface de réseau disponible.
- **La couche internet** : Cette couche n'est rien d'autre que la couche réseau du modèle OSI. Elle se charge de l'acheminement des paquets indépendamment dans n'importe quel réseau. De ce fait, son point critique est le routage, d'où la comparaison avec la couche réseau du modèle OSI.
- **La couche transport** : Cette couche est ressemblante à la couche transport du modèle OSI. Le rôle de la couche transport est d'assurer le transfert des données de bout en bout. Elle a la responsabilité de fournir un échange d'information fiable (protocole TCP et le protocole UDP)
- **La couche application** : Contrairement au modèle OSI, c'est la couche immédiatement supérieure à la couche transport. La couche application offre donc tous les protocoles de haut niveau tel SMTP, http, Telnet, FTP, DNS ...

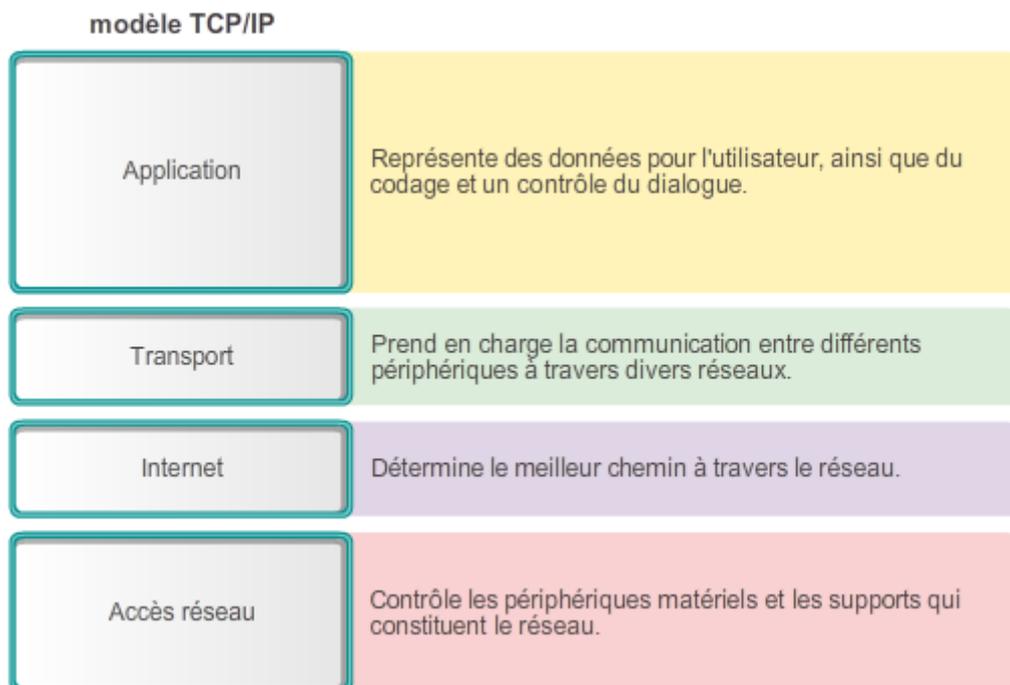


Figure 1. 5 Modèle TCP/IP

6. Internet

Bien qu'il existe des avantages à utiliser un réseau LAN ou WAN, la plupart des utilisateurs doivent communiquer avec des ressources situées sur un autre réseau. Cela est effectué via Internet.

6.1. Définition :

Internet est un ensemble mondial de réseaux interconnectés (interréseaux ou « Internet ») qui coopèrent pour échanger des informations en utilisant des normes courantes. Les fils téléphoniques, fibres optiques, transmissions sans fil et liaisons satellites permettent aux utilisateurs d'Internet d'échanger des informations sous diverses formes.

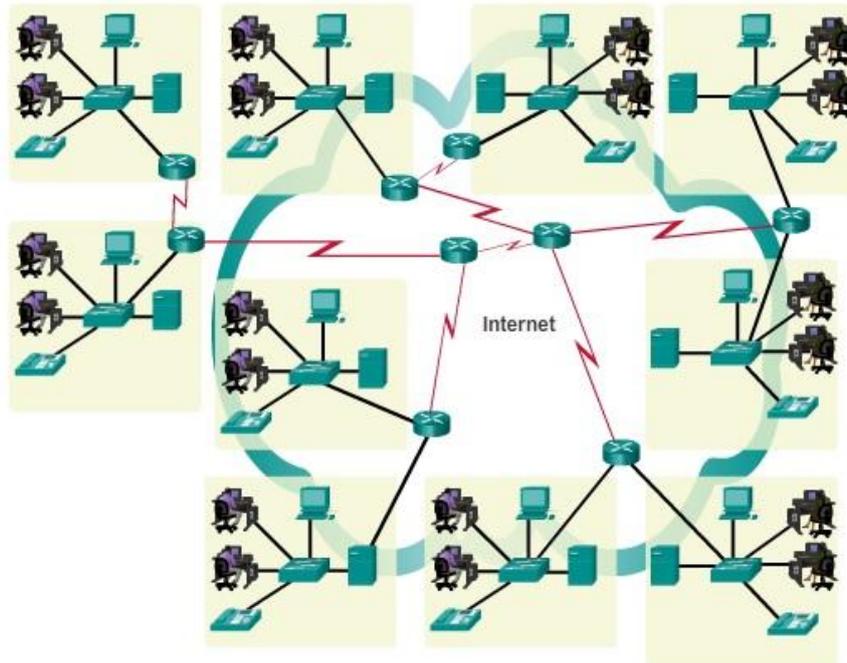


Figure 1. 6 Internet, le réseau des réseaux

Il existe deux autres termes similaires au terme Internet :

6.2. Intranet :

Le terme intranet est souvent utilisé pour faire référence à un réseau LAN privé qui appartient à une entreprise ou une administration et auquel peuvent accéder uniquement ses membres, ses employés ou des tierces personnes autorisées. Un intranet est un interrésseau qui n'est généralement accessible que depuis le site d'une entreprise.

6.3. Extranet :

Une entreprise peut utiliser un extranet pour fournir un accès sécurisé aux personnes qui travaillent pour d'autres entreprises, mais qui ont besoin des données de l'entreprise en question. Voici des exemples d'extranets :

- Une société fournit un accès à des fournisseurs/sous-traitants externes.
- Un hôpital propose un système pour que les médecins puissent planifier les rendez-vous avec leurs patients.
- Un bureau local de formation fournit des informations sur le budget et le personnel aux écoles de la région.

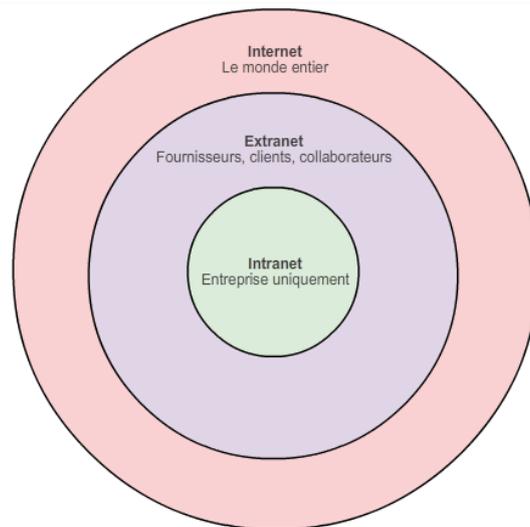


Figure 1. 7 Internet, Intranet et Extranet

7. Le Web :

7.1. Définition :

WWW (littéralement "toile d'araignée mondiale") est le service d'information le plus récent de l'internet. Il repose sur ce qu'on appelle de l'hypertexte. Le concept Web correspond à une base de données universelle où les documents de tous types sont identifiés de manière unique, et pointent les uns vers les autres par des liens. WWW a été développé par le CERN (laboratoire européen de recherche en physique des particules) à Genève en mars 1989.

7.2. Les concepts du web

Le vocabulaire du web s'est beaucoup enrichi au fil des années. Voici quelques termes relatifs au Web :

7.2.1. Page Web : Ensemble de pages web reliées entre elles par des liens hypertextes, accessibles à travers le réseau Internet à l'aide d'une adresse Web (URL).

7.2.2. Navigateur web : Logiciel permettant d'accéder à une page web et d'en afficher le contenu sur l'écran de l'utilisateur en interprétant des fichiers html.

7.2.3. Serveur Web : Désigne à la fois une machine physique et un logiciel. Dans le premier cas, il s'agit d'un ordinateur relié à Internet et hébergeant des ressources. Dans le sens logiciel, un serveur Web est un ensemble de programmes permettant de faire fonctionner et de rendre public un site ou une application web.

7.2.4. Hébergeur Web : Entreprise de services informatiques hébergeant (mettant en ligne) sur ses serveurs Web les ressources constituant les sites Web de ses clients.

7.2.5. HTTP (HyperText Transfert Protocol) : Protocole de Transfert hypertexte. Ce protocole est utilisé par les serveurs Web pour diffuser des documents balisés en HTML sur Internet.

7.2.6. URL (Uniform Ressource Locator): Permet à l'interface de navigation de localiser un document Web ou un fichier sur un serveur FTP, elle indique à l'interface de navigation comment accéder aux informations souhaitées (quel protocole utiliser : HTTP ou FTP), le nom du site internet, le code du domaine et éventuellement, le répertoire ou l'emplacement, sur ce site, du fichier à consulter.

8. Architecture Client-serveur

8.1. Définition :

L'environnement client-serveur désigne un mode de communication à travers un réseau entre plusieurs programmes ou logiciels : l'un, qualifié de client, envoie des requêtes ; l'autre ou les autres, qualifiés de serveurs, attendent les requêtes des clients et y répondent. Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client (client FTP, client de messagerie, etc.).

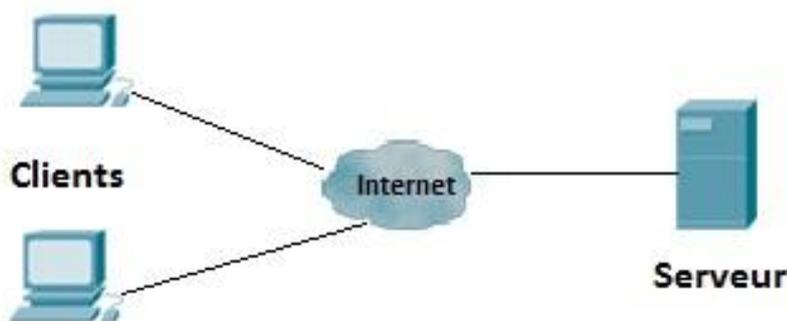


Figure 1. 8 Exemple d'architecture Client/ Serveur

8.2. Caractéristiques

8.2.1. Caractéristiques d'un processus serveur :

- il attend une connexion entrante sur un ou plusieurs ports réseaux ;
- à la connexion d'un client sur le port en écoute, il ouvre un socket local au système d'exploitation;
- suite à la connexion, le processus serveur communique avec le client suivant le protocole prévu par la couche application du modèle OSI.

8.2.2. Caractéristiques d'un processus client :

- il établit la connexion au serveur à destination d'un ou plusieurs ports réseaux ;
- lorsque la connexion est acceptée par le serveur, il communique comme le prévoit la couche applicative du modèle OSI.

8.3. Classification des architectures client/serveur :

8.3.1. Architecture à 2 niveaux : Ce type d'architecture (2-tier en anglais) caractérise les environnements client-serveur où le poste client demande une ressource au serveur qui la fournit à partir de ses propres ressources.

8.3.2. Architecture à 3 niveaux : Dans cette architecture aussi nommée trois tiers en français, un niveau supplémentaire est ajouté :

- Un client (l'ordinateur demandeur de ressources) équipé d'une interface utilisateur (généralement un navigateur web) chargée de la présentation.
- Un serveur d'application qui fournit la ressource, mais en faisant appel à un autre serveur.
- Un serveur de données qui fournit au serveur d'application les données requises pour répondre au client.

8.3.3. Architecture à N niveaux : L'architecture 3 niveaux permet de spécialiser les serveurs dans une tâche précise : avantage de flexibilité, de sécurité et de performance. L'architecture peut être étendue sur un nombre de niveaux plus important : on parle dans ce cas d'architecture à N niveaux

9. Objectif des réseaux :



Figure 9 : Quelques objectifs en images

Un réseau informatique peut servir plusieurs buts distincts :

- Le partage de ressources (fichiers, applications ou matériels, connexion à internet, etc.).
- La communication entre personnes (courrier électronique, discussion en direct, etc.).
- La communication entre processus (entre des ordinateurs industriels par exemple).
- La garantie de l'unicité et de l'universalité de l'accès à l'information (bases de données en réseau).
- Partage d'une connexion Internet.
- Lancement de procédures distantes (client/ serveur).
- Accès à des bases de données centralisées ou réparties.
- Le jeu vidéo multi-joueurs.

10. Conclusion

Ce chapitre s'est porté sur les différentes technologies impliquées dans l'acheminement des données entre machines.

Les différents concepts traités dans ce chapitre nous aideront à mieux comprendre notre mode d'opération et les notions fondamentales pour mener à bien notre application.

Chapitre II

Le Covoiturage

1. Introduction :

Face à la montée du prix de l'essence, les kilomètres d'embouteillages, la perte des liens sociaux, la pollution, le covoiturage constitue une des meilleures alternatives. Pourtant ce système n'a pas la côte : de rares citoyens déclarent partager une voiture, alors que majorité des algériens utilisent leur véhicule personnel pour aller travailler. Ce mode de déplacement permet de diminuer les frais de transport en divisant le cout du trajet par le nombre de passagers et de réduire le trafic et la pollution. Il permet aussi de restaurer une certaine communication qui a disparu dans les transports en commun.



Figure 2. 1 : Panneau d'indication de covoiturage en Europe

2. Définition :

Le covoiturage est l'utilisation conjointe et organisée d'un véhicule, par un conducteur non professionnel et un ou plusieurs tiers passagers, dans le but d'effectuer tout ou une partie un trajet commun.

3. Le principe :

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 décidées. 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.

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 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.

4. Historique :

Le covoiturage est né de dynamiques d'organisation de l'auto-stop dans les années 1950 par des associations allemandes. Cette pratique est encouragée aux États-Unis depuis le début des années 1970 lors de la première crise pétrolière, de nombreuses associations se sont montées pour encourager salariés et étudiants à partager leur véhicule sur les trajets quotidiens « domicile-travail » et « domicile-université ».

Depuis les années 1980, il représente une alternative de transport reconnue, qui s'étend dans le monde occidental. Parallèlement, une politique incitative de construction de voiries réservées aux véhicules à fort taux d'occupations (High Occupancy Vehicle lanes ou HOV lanes) a été mise en place.

En Europe, le covoiturage s'est développé essentiellement à partir des années 1990 ; quelques pays se sont dotés d'outils facilitant et stimulant le covoiturage. Les Pays-Bas ont investi dans des campagnes nationales d'informations, deux grandes associations organisent le covoiturage en Allemagne depuis 1980 et la Belgique fournit une base de données nationale sur laquelle chaque entreprise peut organiser son covoiturage.

Mis à part cela, aucune politique publique n'est venue appuyer les initiatives d'associations créées depuis 1995, qui n'ont touché qu'un public de convaincus. Il faut attendre les années 2000 pour voir apparaître les premiers signes de soutien des collectivités locales aux associations de covoiturage.

L'Internet a beaucoup contribué à l'émergence de cette pratique de transport en facilitant les contacts entre conducteurs et passagers mais il reste peu pratiqué en Algérie.

5. A qui s'adresse-t-il ?

Entreprises

Les entreprises offrent un cadre privilégié pour développer le covoiturage. Dans le cadre de PDE (plan de déplacement entreprise), les entreprises peuvent encourager leurs salariés à le pratiquer. Il représente pour les employeurs comme pour les employés, de nombreux avantages financiers (économie du coût des transports, des infrastructures...) et pratiques, notamment si l'entreprise est située en périphérie et peu accessible car l'organisation du covoiturage est plus facile au sein d'une structure déjà établie.

Particuliers

Pour les particuliers le covoiturage s'adresse à tous. Il s'est répandu et touche maintenant tous les âges et tous les modes de vie. À noter, un système de covoiturage particulièrement bien pensé est mis en place pour les particuliers lors de trajets réguliers mais aussi événementiels : festivals, épreuves sportives, salons...

6. Les types de covoiturages :

On distingue aussi trois types de trajets de covoiturage:

Les trajets réguliers : c'est-à-dire les trajets qui se font de manière hebdomadaire (au moins une fois par semaine). Les trajets quotidiens (ou journaliers) font généralement référence aux trajets entre le lieu de résidence et le lieu de travail (également trajets scolaires). Ils sont assez répandus en Algérie au sein des quartiers populaires.

Les trajets occasionnels : sont, comme leur nom l'indique, occasionnels ; c'est-à-dire qu'ils n'ont pas (ou peu souvent) vocation à se répéter dans le temps.

Ce sont des trajets principalement longue distance très utilisés pour les départs en vacances et les événements à travers un état (soirées, concert, festival...).

Le covoiturage événementiel (covoiturage de crise) : il est nettement moins répandu et beaucoup plus informel. Il s'agit en fait de covoiturer lorsqu'on ne peut pas utiliser son propre moyen de locomotion habituel (panne, grèves des transports collectifs, accident corporel...).

7. Avantages et Inconvénients du covoiturage :

7.1. Avantages :

- Favoriser les échanges entre citoyens car le partage d'un véhicule restaure une communication entre passagers qui n'existe pas dans les transports en commun.
- Réduire l'empreinte écologique car la mutualisation de la voiture permet de diminuer les gaz à effet de serre. En effet, le secteur des transports est le premier consommateur de produits pétroliers et constitue la première source d'émission de GES.
- Fait faire des économies car vous mutualisez l'utilisation de la voiture, le coût des éventuels péages, des places de parking et surtout, de l'essence. Une économie non négligeable dans un contexte d'augmentation du prix du pétrole.
- Dans les zones à faible et très faible densité de population, le covoiturage permet de compenser la carence ou l'absence de transports en commun pour les personnes ne possédant pas de véhicule.



Figure 2. 2 : Nombre de véhicules selon le nombre d'occupant

7.2. Inconvénients :

Bien qu'il présente des avantages non négligeables, le covoiturage possède également ses points faibles dont le premier touche les horaires. En effet, chaque voyage doit se conformer à l'heure de départ soumise par le propriétaire du véhicule ou établie d'un accord commun. D'autres pratiquants pensent que le manque de flexibilité durant le voyage peut aussi constituer un désavantage pour ceux qui, durant de longs voyages, ont l'habitude de s'arrêter pour faire une petite pause ou tout simplement contempler les paysages. Un autre inconvénient présenté par le covoiturage concerne la couverture de l'assurance. Il s'avère effectivement que les compagnies d'assurances ne proposent pas de garanties convenables pour un voyage en covoiturage.

Mais en dépit de ces désagréments, plusieurs particuliers apprécient ce mode de déplacement. Le nombre de covoiturage augmente en effet et il se révèle que près de 97% des pratiquants déclarent être satisfaits et désirent retenter l'expérience.

8. Le covoiturage dynamique :

Le covoiturage dynamique associe plusieurs technologies, la géo-localisation par GPS et les connexions 3G. Il permet de synchroniser en temps réel les demandes et les offres de trajets.

Le covoiturage dynamique prend également en charge la gestion automatique des frais. Avec l'aide de porte-monnaies virtuels, conducteurs et passagers n'ont pas besoin d'échanger physiquement de monnaie, qui est numériquement et instantanément transférée d'un compte à l'autre en fonction de plusieurs paramètres (modèle et année de la voiture, consommation réelle de carburant, etc.).

Chapitre III

Analyse et Conception

1. Introduction

Avant l'implémentation et la réalisation de toute application informatique, il convient de suivre une démarche méthodologique. Pour réaliser notre application, nous commençons par une analyse réfléchie qui permet d'identifier les différents acteurs qui interagissent avec le système ainsi que leurs tâches respectives. La phase d'Analyse sera suivie d'une conception en se basant sur le formalisme UML (Unified Modeling Language), qui offre la possibilité de modéliser un site Web. Ce langage semi formel permet la représentation des aspects, statique et dynamique d'une application web par une série de diagrammes. Enfin, nous spécifierons les structures des différentes tables de la base de données implémentée.

2. Problématique et expression initiale des besoins

La pollution de l'air demeure un problème majeur dans les grandes villes d'Algérie. Cette pollution est principalement causée par l'émission de gaz à effet de serre des véhicules de tourisme et de transport en commun. Le partage de véhicule est donc la meilleure solution.

Cependant, il reste difficile d'étendre le champ de covoitureurs au-delà de l'entourage des conducteurs car le bouche à oreille ne suffit plus.

3. Positionnement et objectif de l'application

L'objectif fondamental du futur site Covoiturage DZ est de mettre en contact des internautes ayant des itinéraires similaires, quotidiens ou occasionnels, pour partager un même véhicule selon le principe du Covoiturage.

L'application servira donc de plateforme où les personnes véhiculées publieront des annonces détaillées, ce dans le but d'attirer des passagers à faire le trajet en leur compagnie contre leur contribution.

4. Analyse et conception :

4.1. Analyse :

4.1.1. Identification des acteurs

Le visiteur: Cette appellation vise toute personne pouvant consulter le site dans le but de rechercher des annonces de covoiturages.

Le membre authentifié : Membre qui visite le site pour publier ou rechercher des annonces de covoiturages et éventuellement s’y inscrire. Il s’agit bien sûr de l’acteur le plus important, celui pour lequel le site existe.

L’administrateur : Appelé aussi Webmaster, est chargé du bon fonctionnement et de la maintenance du site web.

4.1.2. Spécification des tâches :

Pour chaque acteur identifié précédemment, il convient de rechercher les différentes intentions selon lesquelles il utilise le système.

Acteur	Tâches
Visiteur	T 01 : Consulter le site.
	T 02 : Créer un compte
	T 03 : S’authentifier
	T 04 : Faire une recherche
	T 05 : Consulter des profils
	T 06 : Consulter des covoiturages
Membre authentifié	T 01, T 04, T 05, T 06
	T 07 : S’inscrire à un covoiturage
	T 08 : Publier un covoiturage
	T 09 : Editer profil
	T 10 : Commenter un covoiturage
	T 11 : Laisser un avis sur conducteur
	T 12 : Annuler une réservation
	T 13 : Supprimer un covoiturage
	T 14 : Refuser un passager
	T 15 : Accepter un passager
Administrateur	T 16 : Consulter les fichiers Logs
	T 17 : Répondre aux mails des visiteurs
	T 18 : Modifier le contenu

Tableau 1: Tableau spécifiant les tâches de chaque acteur

4.1.3. Spécification des scénarios :

Les scénarios décrivant chacune des tâches définies auparavant sont récapitulés dans le tableau suivant :

Tâches	Scénarios
T 01 : Consulter le site	S1 : taper l'URL du site dans le navigateur.
T 02 : Créer un compte	S2 : cliquer sur le lien « S'inscrire » dans la barre de navigation. S3 : remplir le formulaire. S4 : cliquer sur le bouton « S'inscrire ».
T 03 : S'authentifier	S5 : cliquer sur le lien «Se connecter » dans la barre de navigation. S6 : saisir son login et son mot de passe. S7 : cliquer sur le bouton « se connecter ».
T 04 : Faire une recherche	S8 : saisir les informations relatives à la recherche (ville de départ, ville d'arrivée et date de départ) sur la page d'accueil. S9 : cliquer sur la loupe.
T 05 : Consulter des profils	S10 : cliquer sur le nom ou l'avatar d'un membre.
T 06 : Consulter des covoiturages	S11 : cliquer sur une vignette de covoiturage.
T 07: S'inscrire à un covoiturage	S12: dans la page « détails du covoiturage » cliquer sur le bouton « S'inscrire ».
T 08: Publier un covoiturage	S13 : cliquer sur le lien « publier » dans la barre de navigation. S14 : remplir le formulaire. S15 : clique sur le bouton « Publier ».
T 09: Editer profil	S 16: cliquer sur le menu déroulant dans la barre de navigation. S17 : cliquer sur le lien « Editer profil ». S18 : Modifier le contenu de formulaire. S 19 : Cliquer sur « Enregistrer ».
T 10 : Commenter un covoiturage	S20 : Saisir son commentaire en bas de la page «détails du covoiturage ». S21: cliquer sur «Envoyer ».
T 11 : Laisser un avis sur conducteur	S22 : choisir une appréciation sur le profil d'un conducteur. S23 : saisir un court commentaire résumant la conduite du conducteur. S24 : cliquer sur «Envoyer ».

T 12 : Annuler une réservation	S25 : cliquer sur le lien « Mes Covoiturages ». S26 : dans l'onglet « mes réservations ». S27 : cliquer sur l'annonce de covoiturage à laquelle le membre est inscrit. S28 : cliquer sur le bouton « Annuler ».
T 13 : Supprimer un covoiturage	S29 : cliquer sur le lien « Mes Covoiturages ». S30 : dans l'onglet « mes annonces ». S31 : cliquer sur l'annonce de covoiturage en question. S32 : cliquer sur le bouton « Annuler ce covoiturage ».
T 14 : Refuser un passager	S33 : cliquer sur le lien « Mes Covoiturages ». S34 : dans l'onglet « mes annonces », cliquer sur l'annonce. S35 : cliquer sur le bouton rouge « Refuser » sur la ligne du passager.
T 15 : Accepter un passager	S36 : cliquer sur le lien « Mes Covoiturages ». S37 : dans l'onglet « mes annonces », cliquer sur l'annonce. S38 : cliquer sur le bouton vert « Accepter » sur la ligne du passager.
T 16 : Consulter les fichiers Logs	S39 : ouvrir un fichier de Logs dans www\covoiturageDZ\storage\logs avec un client ftp.
T 17: Répondre aux mails des visiteurs	S40 : consulter la boîte mail et répondre aux doléances des visiteurs
T 18 : Modérer le contenu	S41 : récupérer l'adresse mail de l'utilisateur ayant un comportement abusif. S42 : envoyer un email d'avertissement. S43 : dans le cas échéant, supprimer le compte.

Tableau 2 : Tableau spécifiant les scénarios de chaque tâche

4.1.4. Diagrammes des cas d'utilisation :

Le diagramme de cas d'utilisation général représente les cas d'utilisation des trois acteurs Visiteur, Membre authentifié et Administrateur.

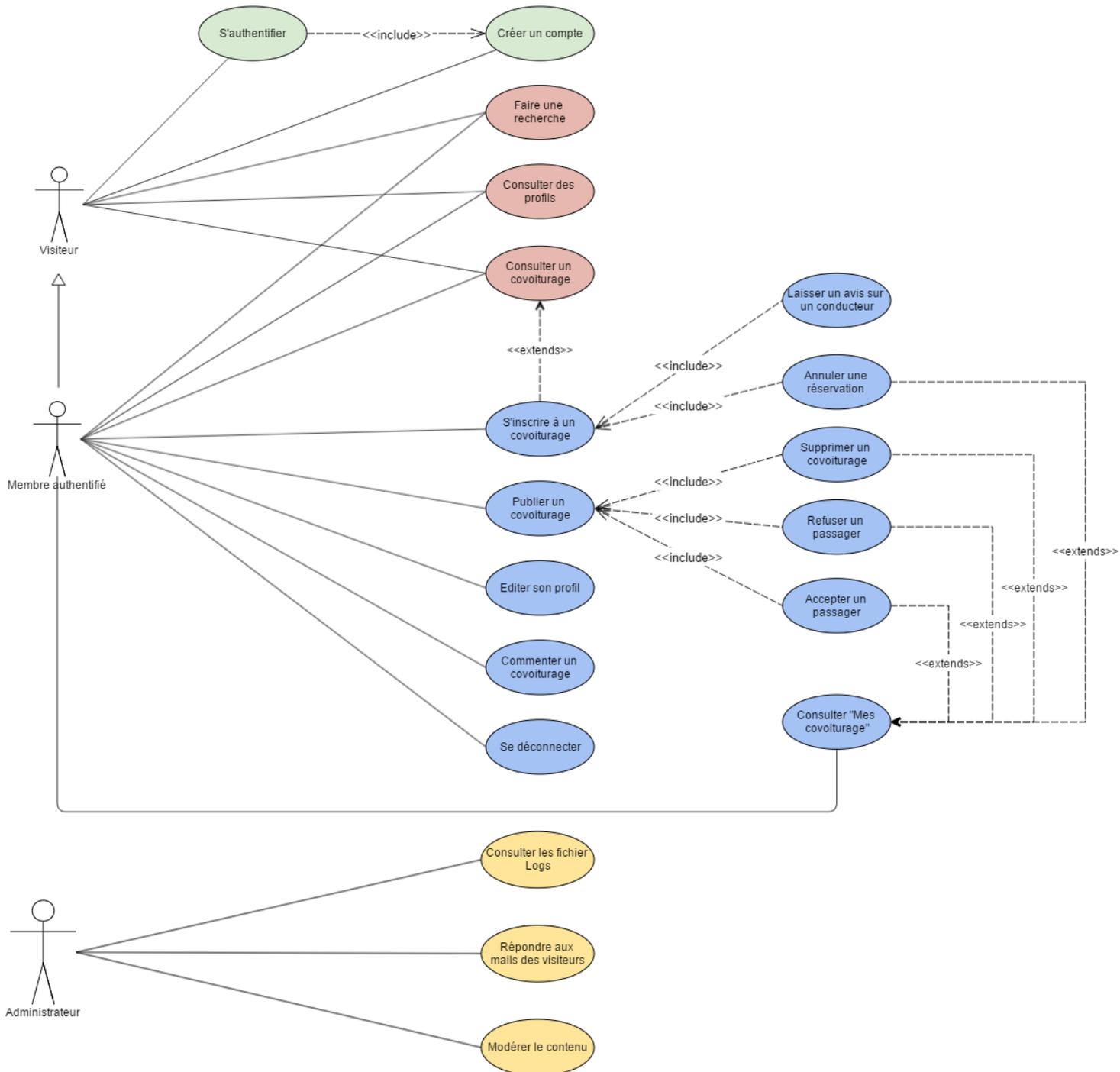


Figure 3. 1 Diagramme des cas d'utilisation général

4.2.1.2. Diagramme de séquence détaillée du cas d'utilisation « S'inscrire à un covoiturage » :

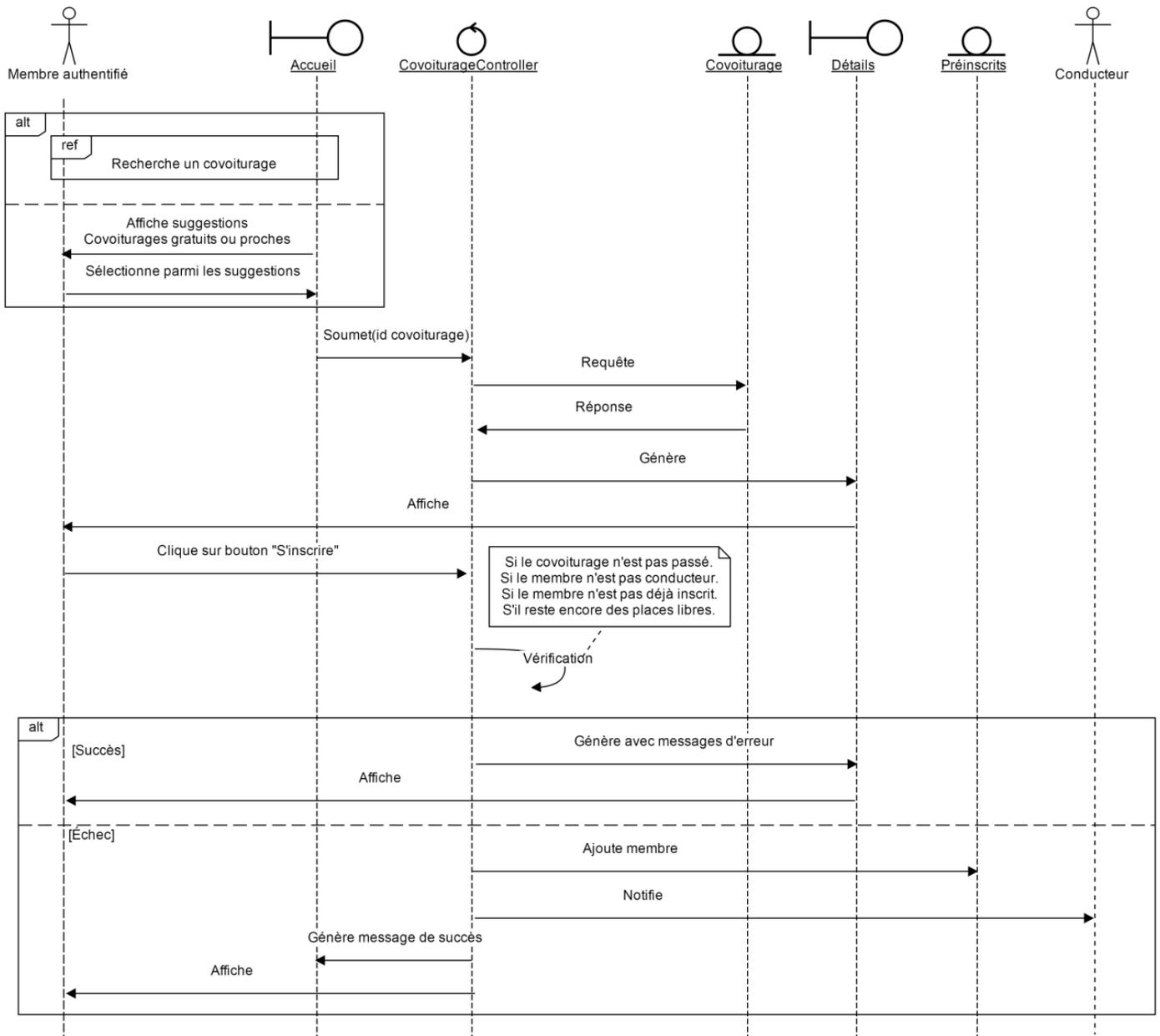


Figure 3. 3 Diagramme de séquence détaillée du cas d'utilisation « S'inscrire à un covoiturage »

4.2.1.3. Diagramme de séquence détaillé du cas d'utilisation «Faire une recherche » :

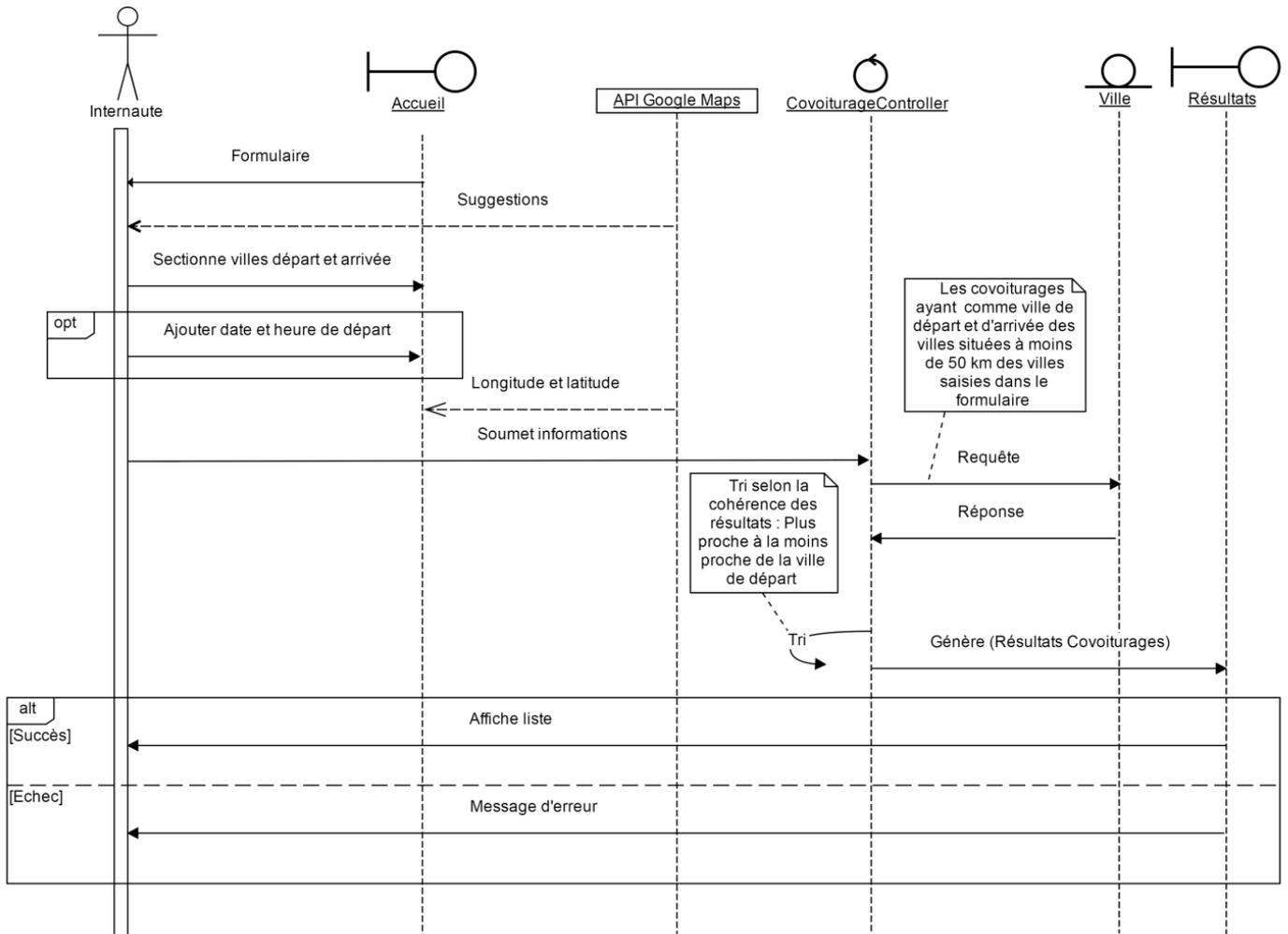


Figure 3. 4 Diagramme de séquence détaillé du cas d'utilisation «Faire une recherche »

4.2.1.4 Diagramme de séquence système du cas d'utilisation « S'authentifier » :

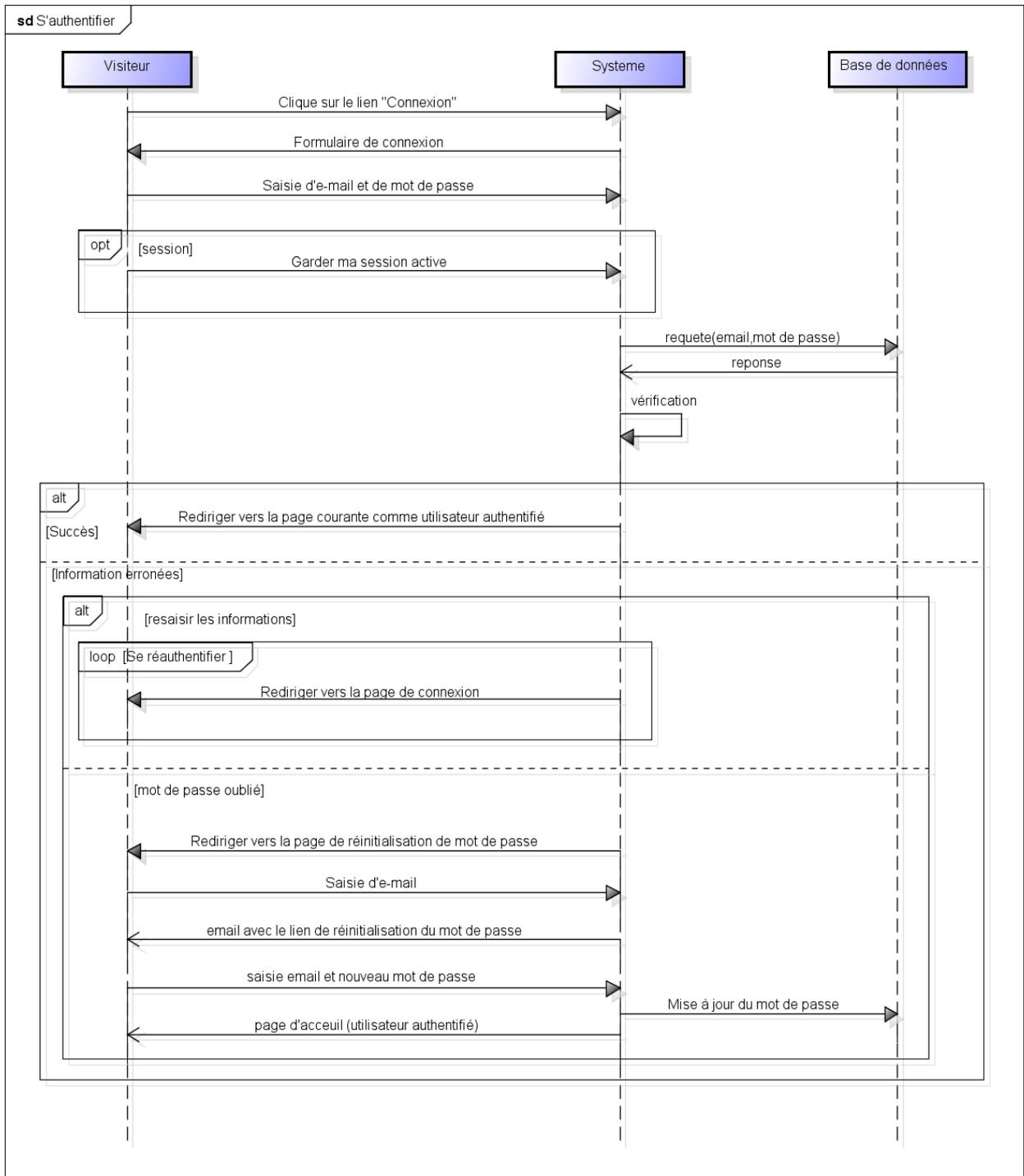


Figure 3. 5 : Diagramme de séquence système du cas d'utilisation « S'authentifier »

4.2.1.5 Diagramme de séquence système du cas d'utilisation « Publier un covoiturage » :

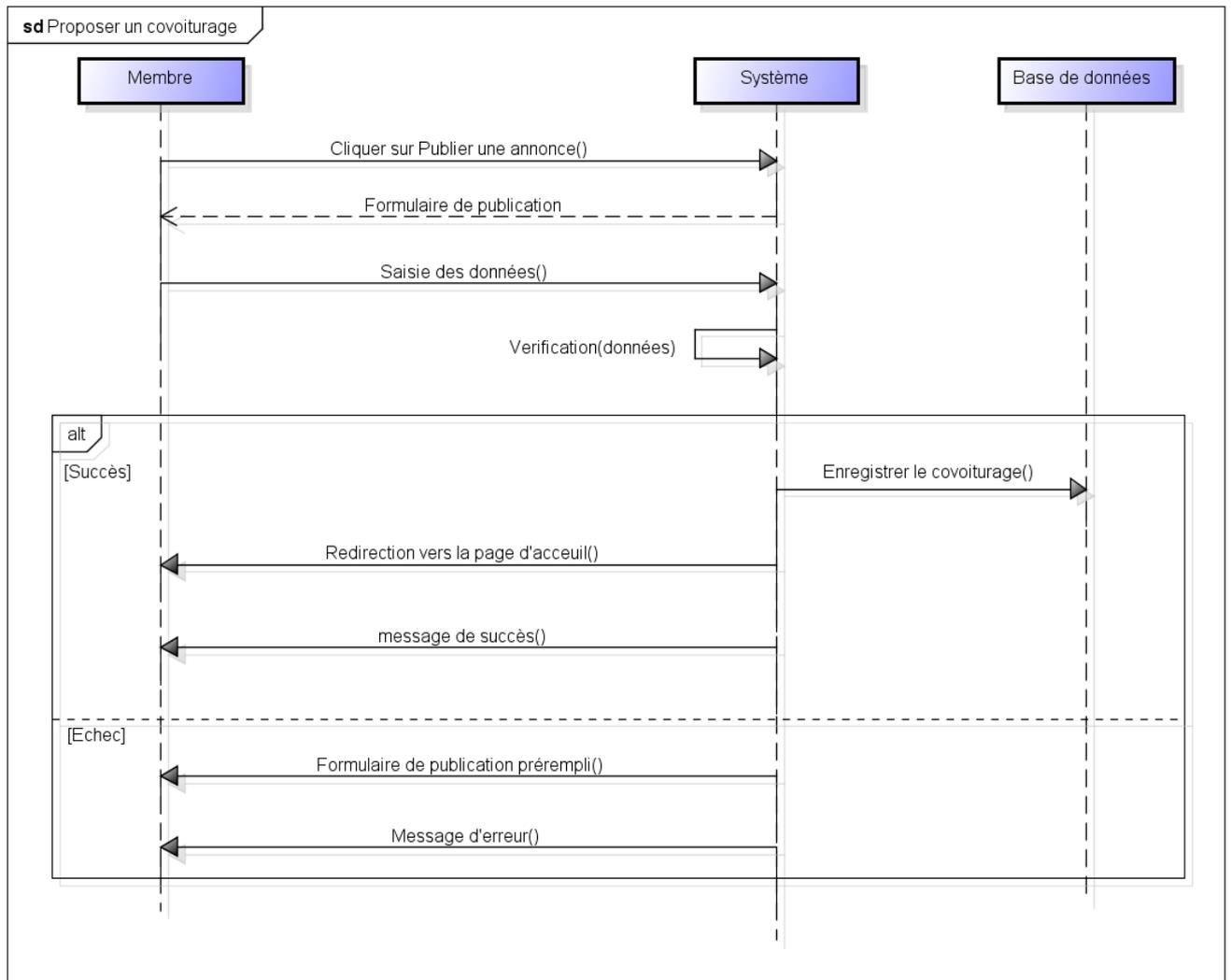


Figure 3. 6 Diagramme de séquence système du cas d'utilisation « Publier un Covoiturage »

4.2.1.6 Diagramme de séquence système du cas d'utilisation « Editer son profil » :

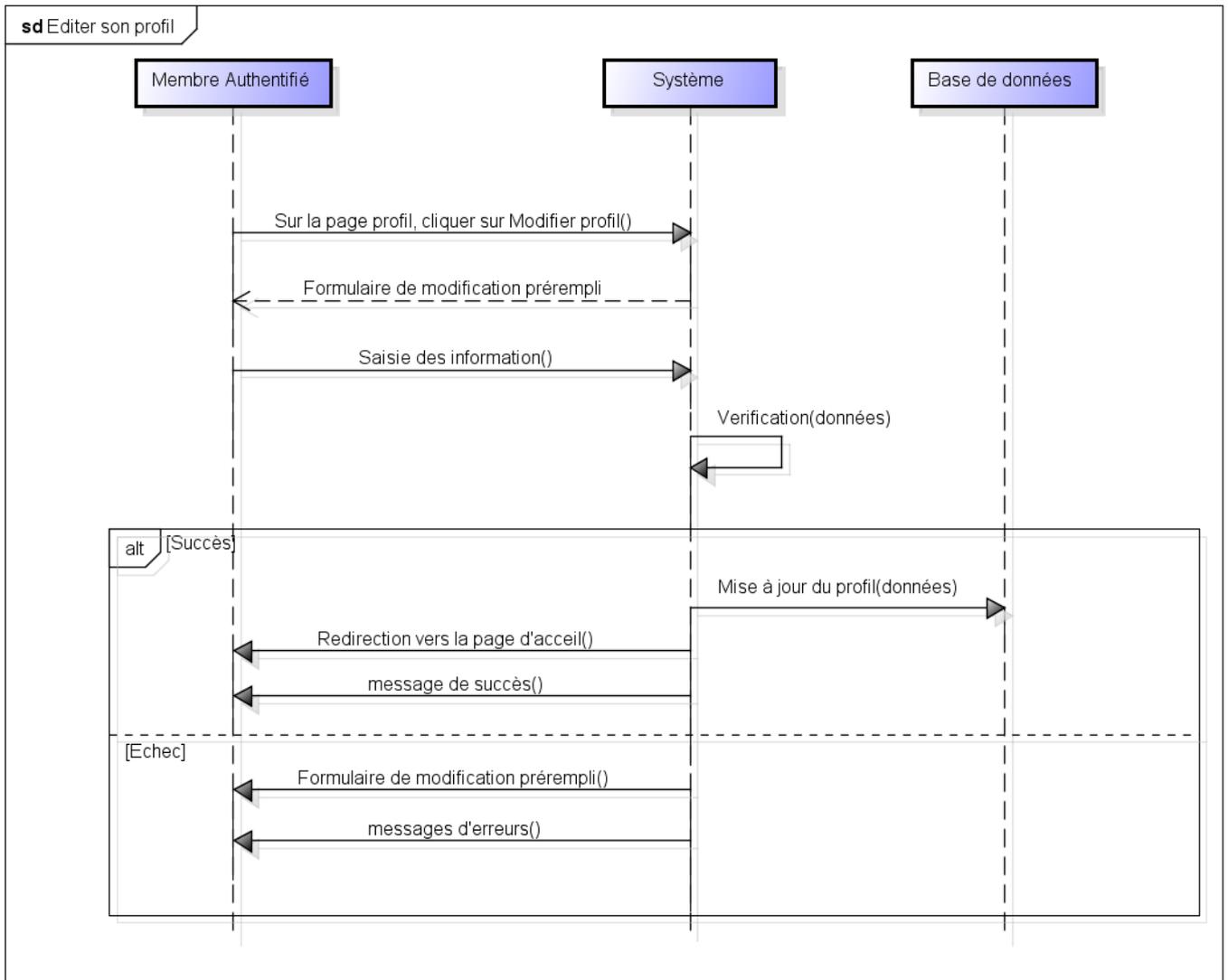
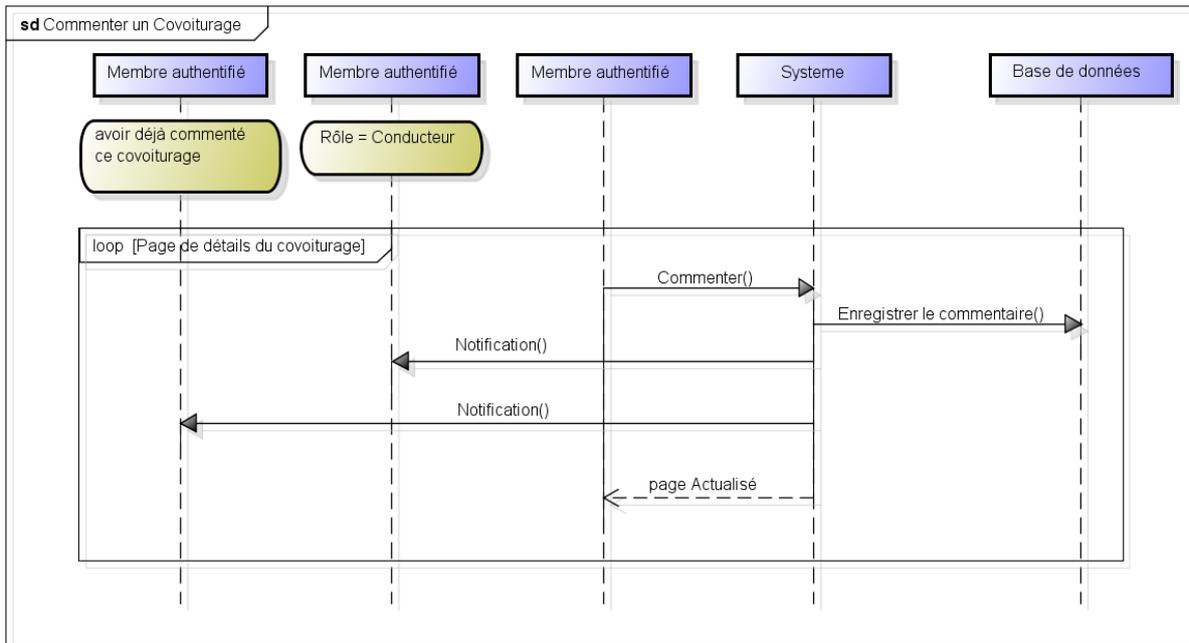


Figure 3. 7 Diagramme de séquence système du cas d'utilisation « Editer son profil »

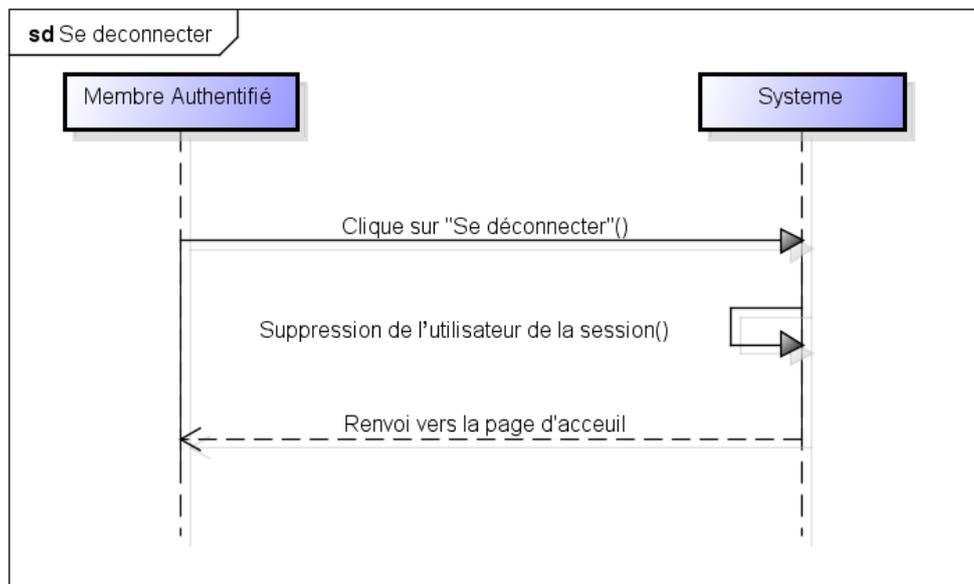
4.2.1.7 Diagramme de séquence système du cas d'utilisation « Commenter un covoiturage » :



powered by Astah

Figure 3. 8 Diagramme de séquence système du cas d'utilisation « Commenter un covoiturage »

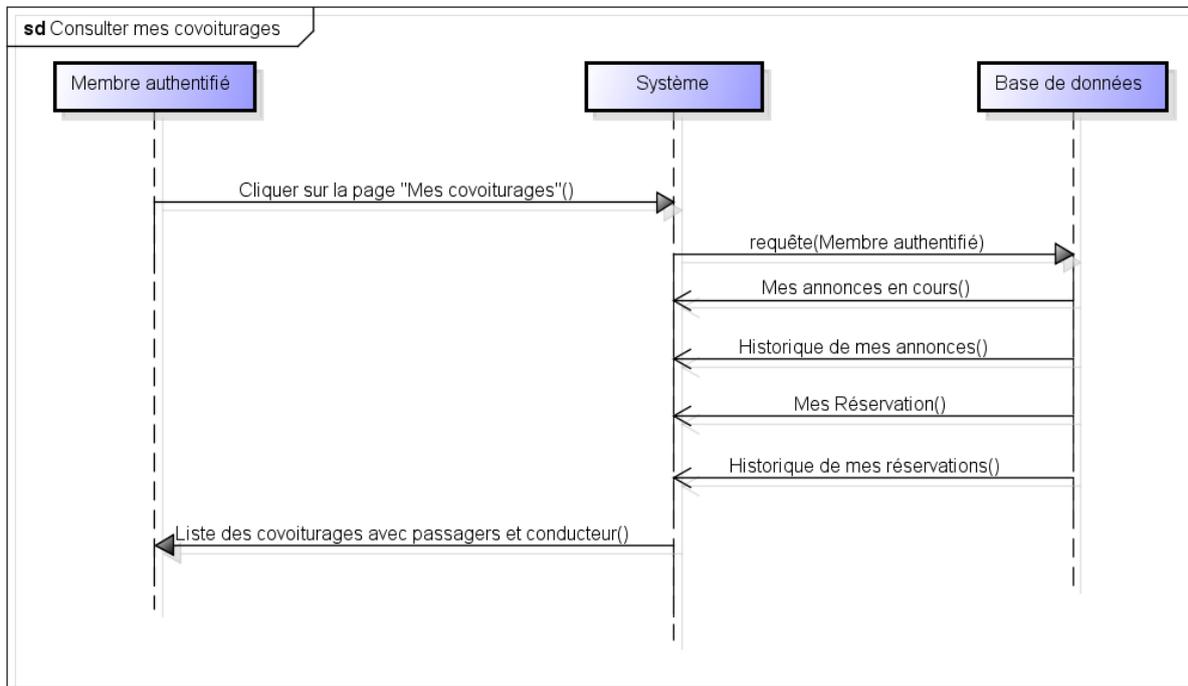
4.2.1.8 Diagramme de séquence système du cas d'utilisation « Se déconnecter » :



powered by Astah

Figure 3. 9 Diagramme de séquence système du cas d'utilisation «Se déconnecter»

4.2.1.9 Diagramme de séquence système du cas d'utilisation « Consulter mes covoiturages » :



powered by Astah

Figure 3. 10 Diagramme de séquence système du cas d'utilisation « Consulter mes covoiturages »

4.2.1.10 Diagramme de séquence système du cas d'utilisation « Accepter ou refuser des passagers » :

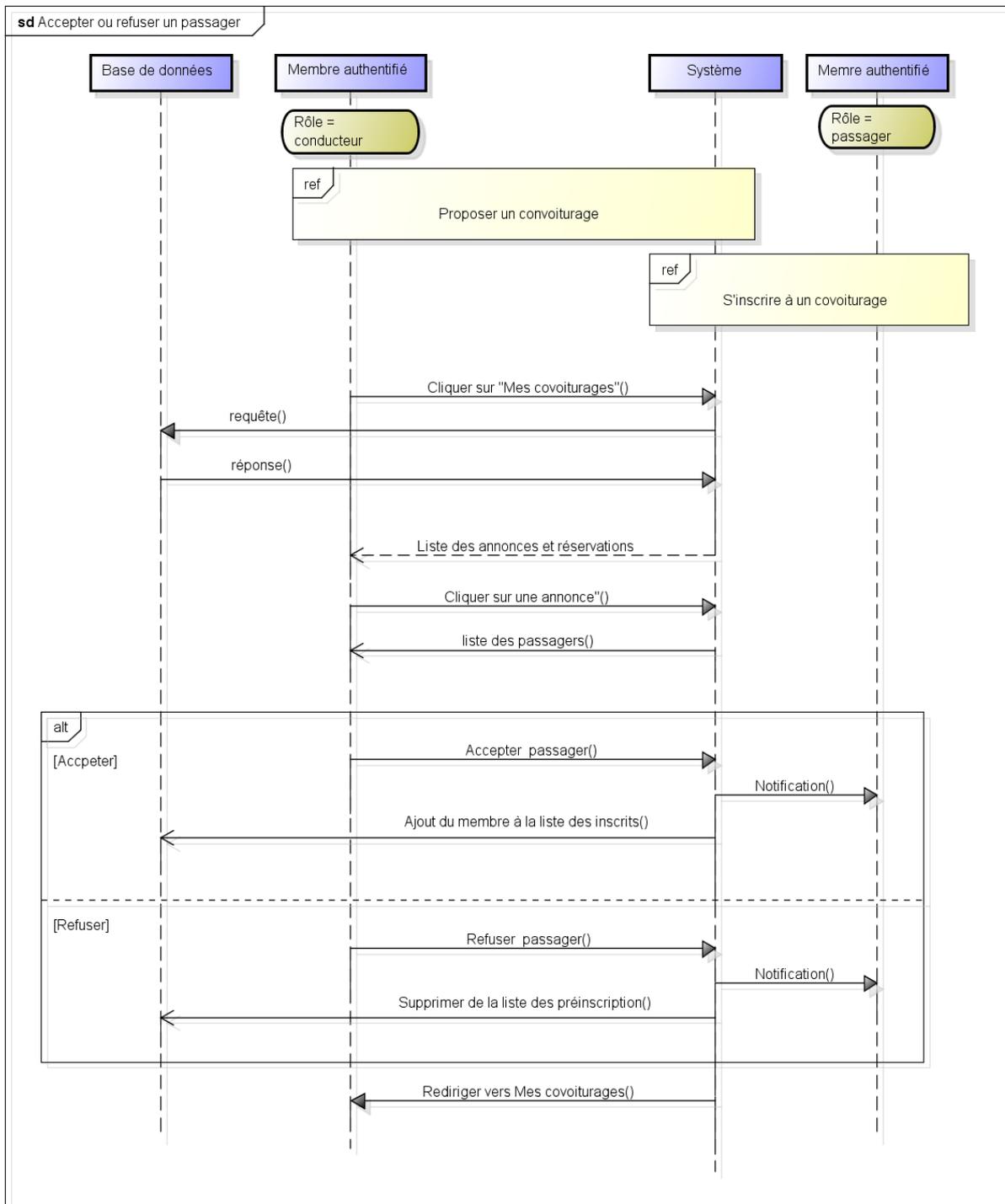
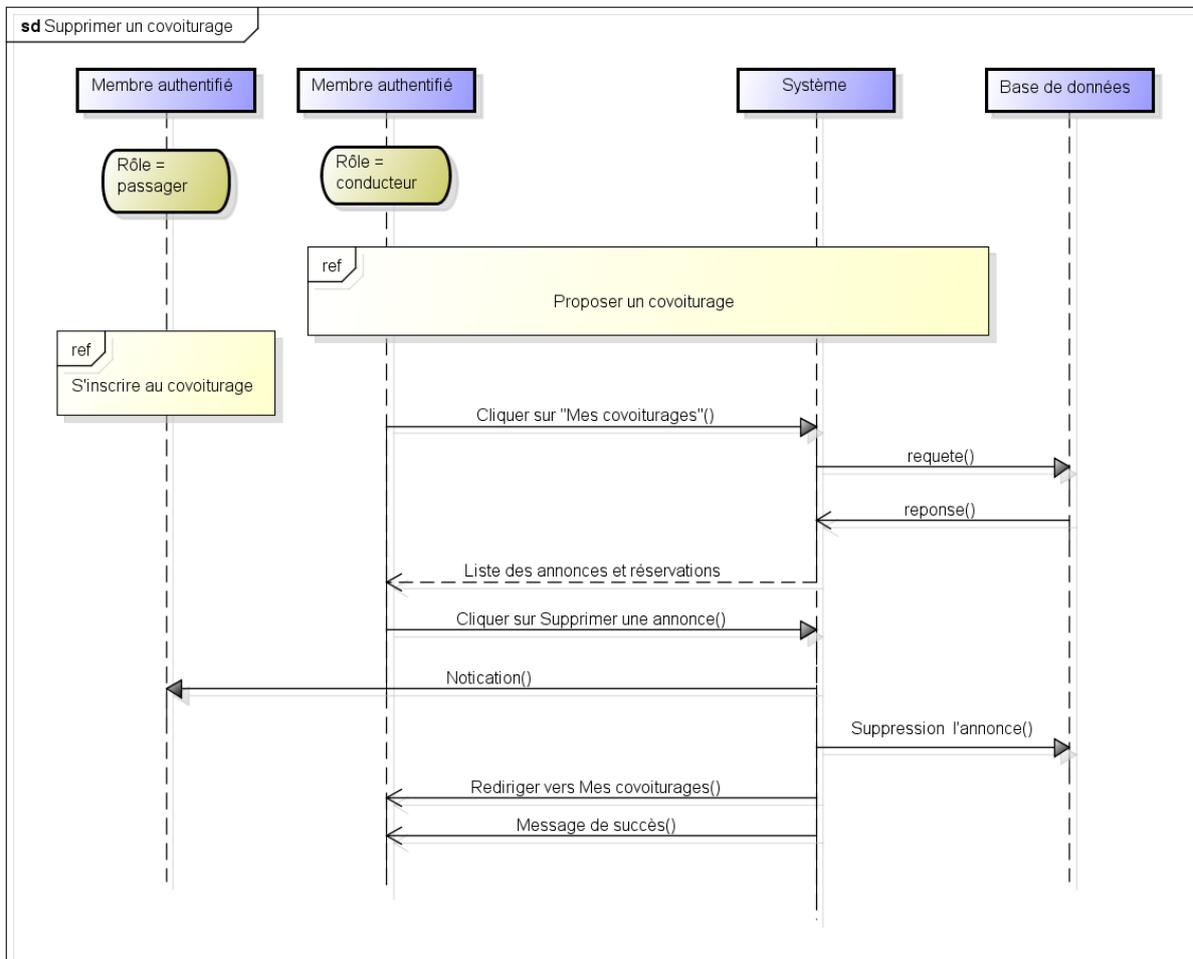


Figure 3. 11 Diagramme de séquence système du cas d'utilisation « Accepter ou refuser des passagers »

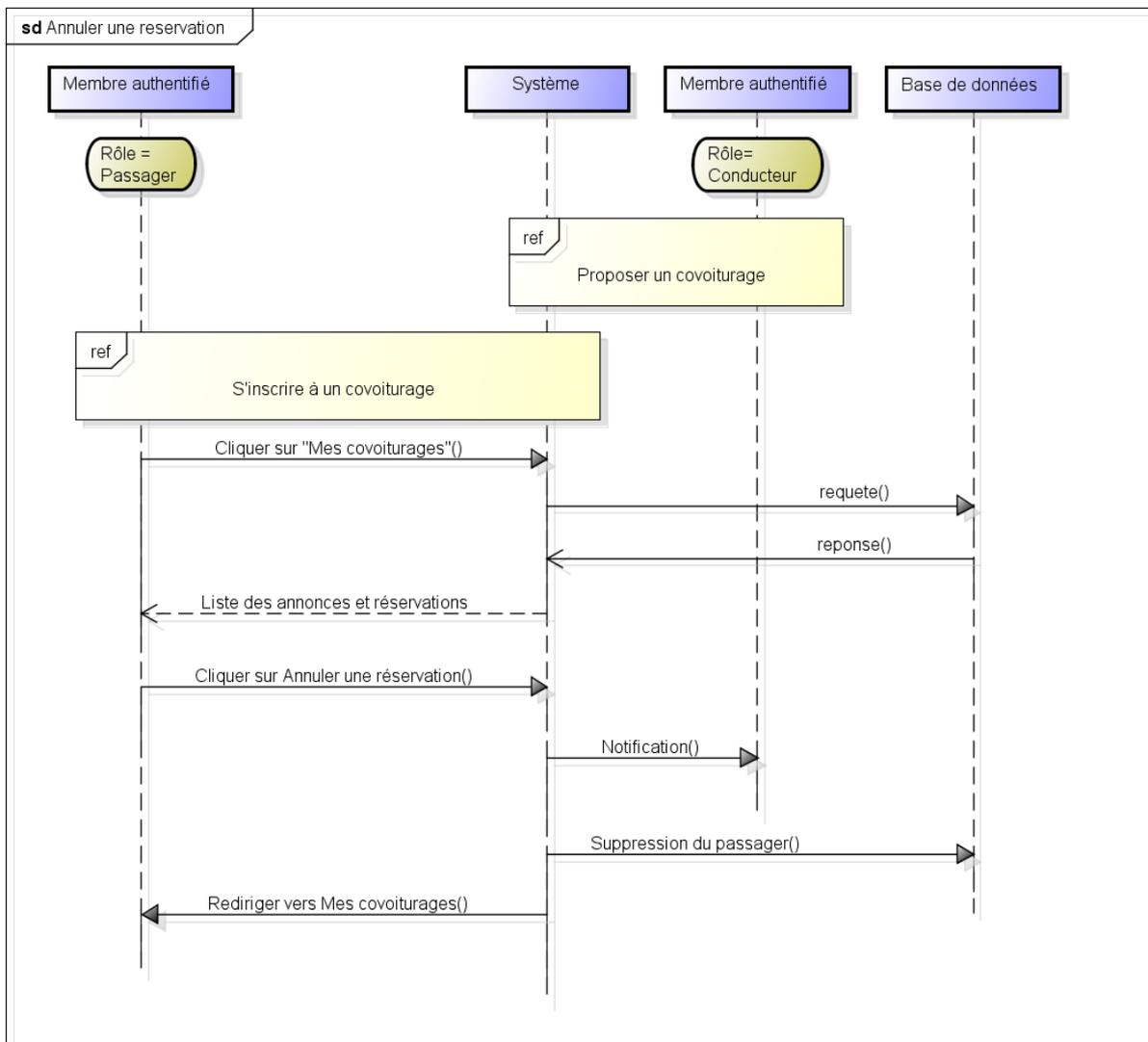
4.2.1.11 Diagramme de séquence système du cas d'utilisation « Supprimer un covoiturage » :



powered by Astah

Figure 3. 12 Diagramme de séquence système du cas d'utilisation « Supprimer un covoiturage »

4.2.1.12 Diagramme de séquence système du cas d'utilisation « Annuler une réservation »



powered by Astah

Figure 3. 13 Diagramme de séquence système du cas d'utilisation « Annuler une réservation »

4.2.2 Diagrammes de classes :

4.2.2.1 Diagramme des classes général :

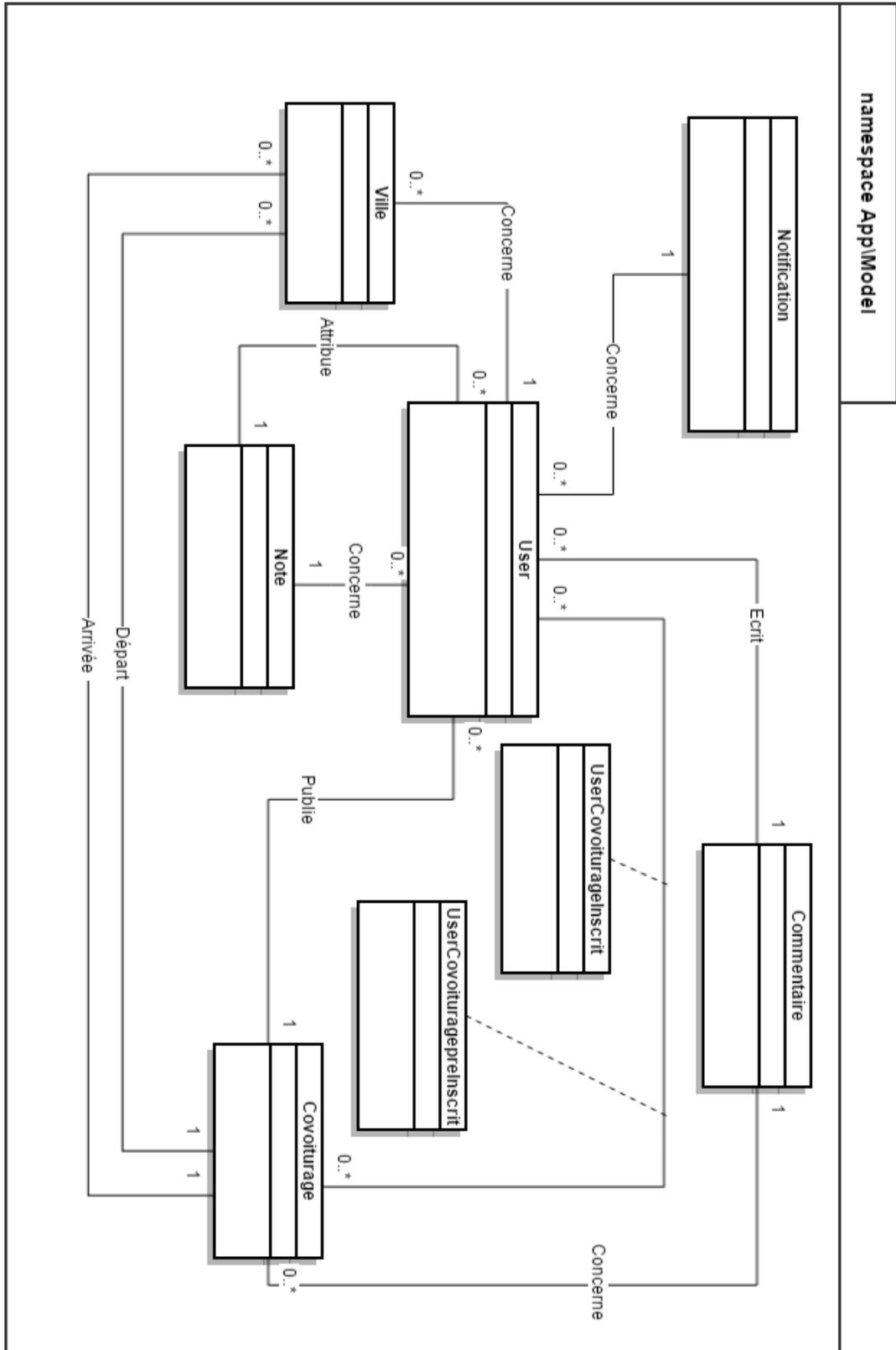


Fig. IV.14 : Diagramme des classes général

4.2.2.2 Diagramme des classes des contrôleurs:

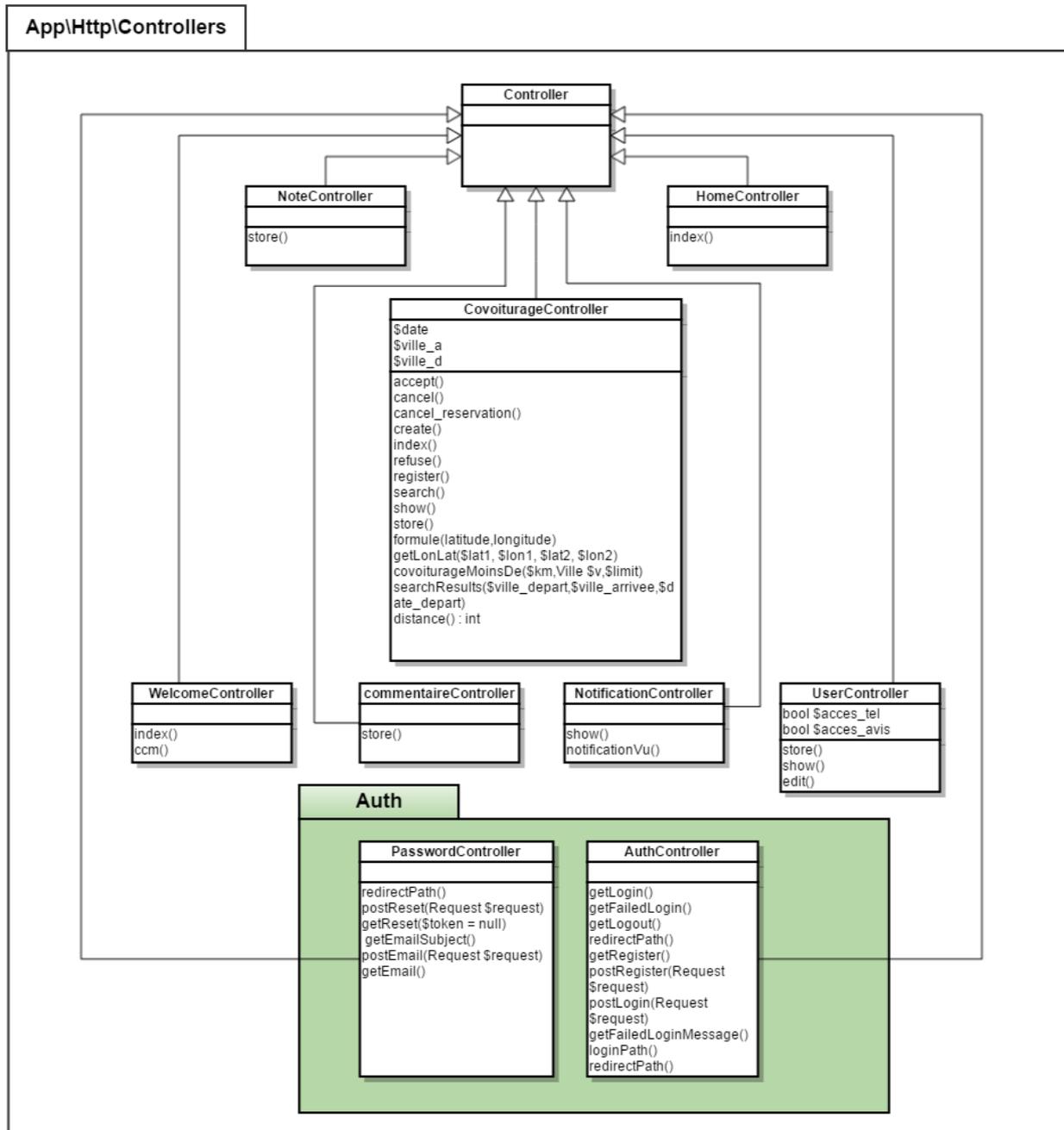


Figure 3. 15 Diagramme des classes des contrôleurs

4.2.2.3 Diagramme des classes des modèles :

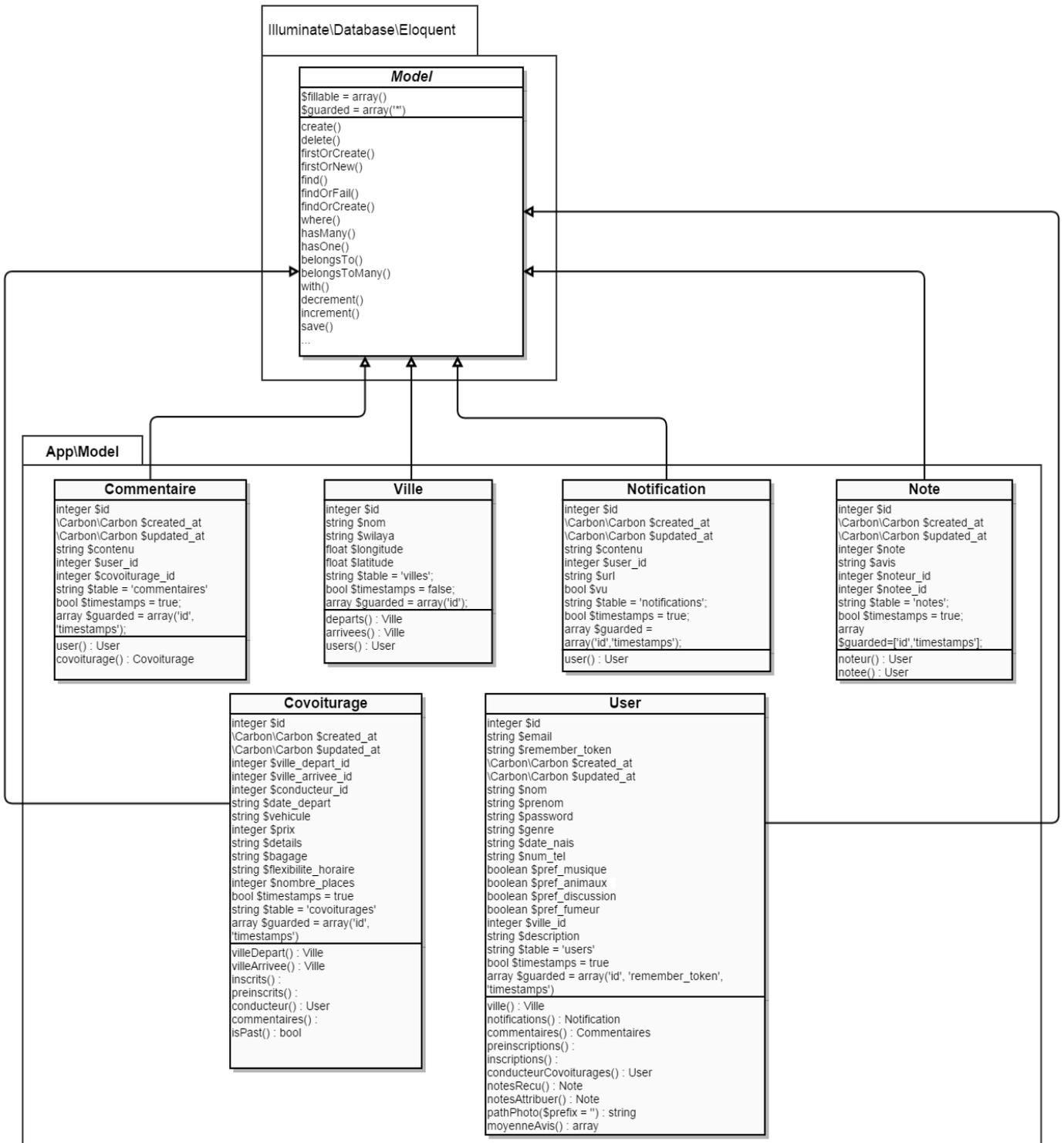


Figure 3. 16 Diagramme des classes des modèles

4.2.3 Diagrammes de navigation :

4.2.3.1 Diagramme global de navigation pour l'internaute :

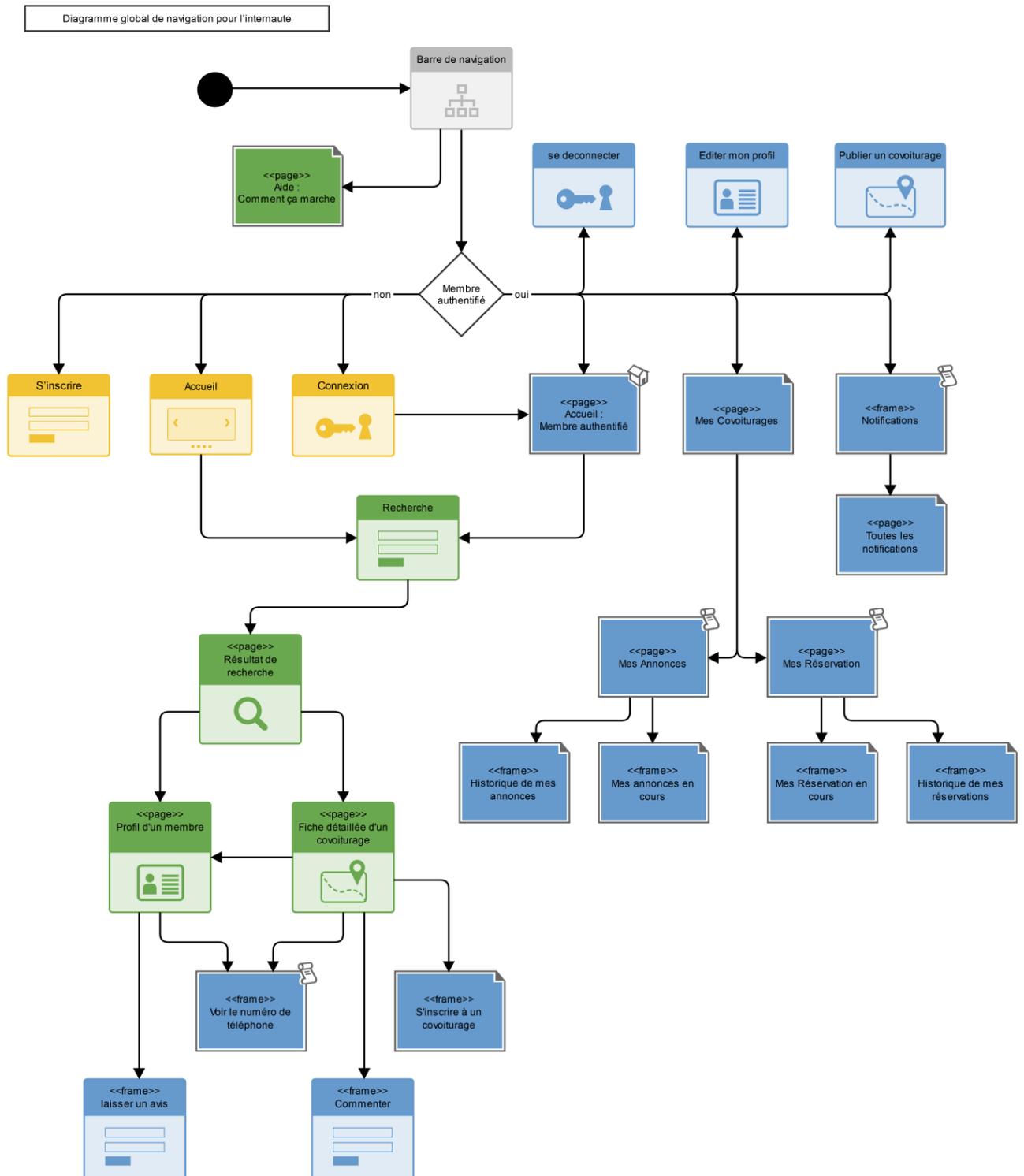


Figure 3. 17 Diagramme global de navigation pour l'internaute

4.2.3.2 Diagramme de navigation de la Recherche de Covoiturage :

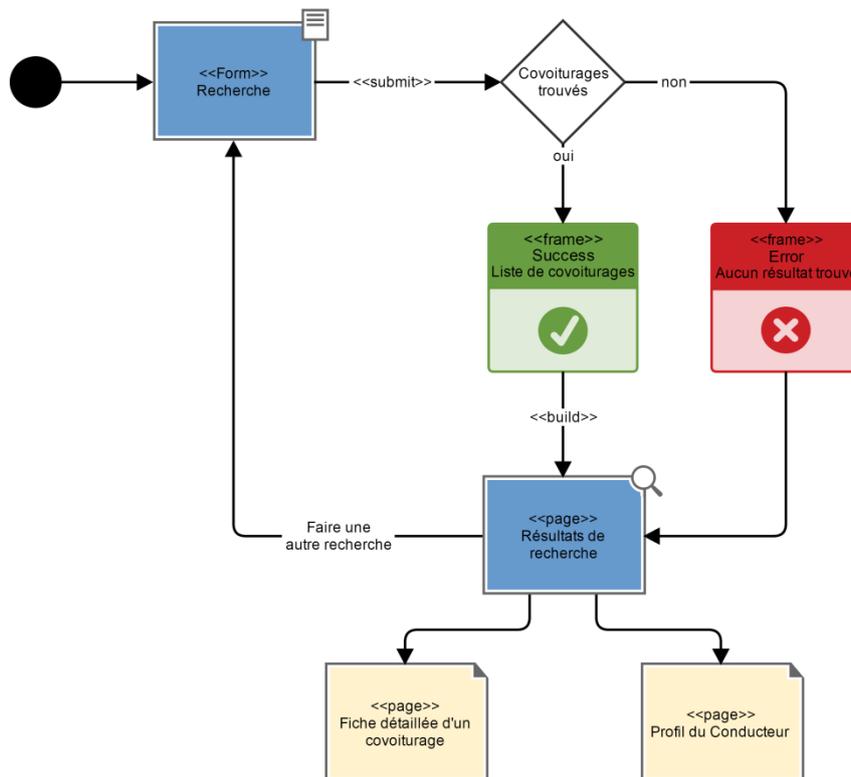


Figure 3. 18 Diagramme de navigation de la Recherche de Covoiturage

4.2.3.3 Diagramme de navigation de la Consultation de Profil :

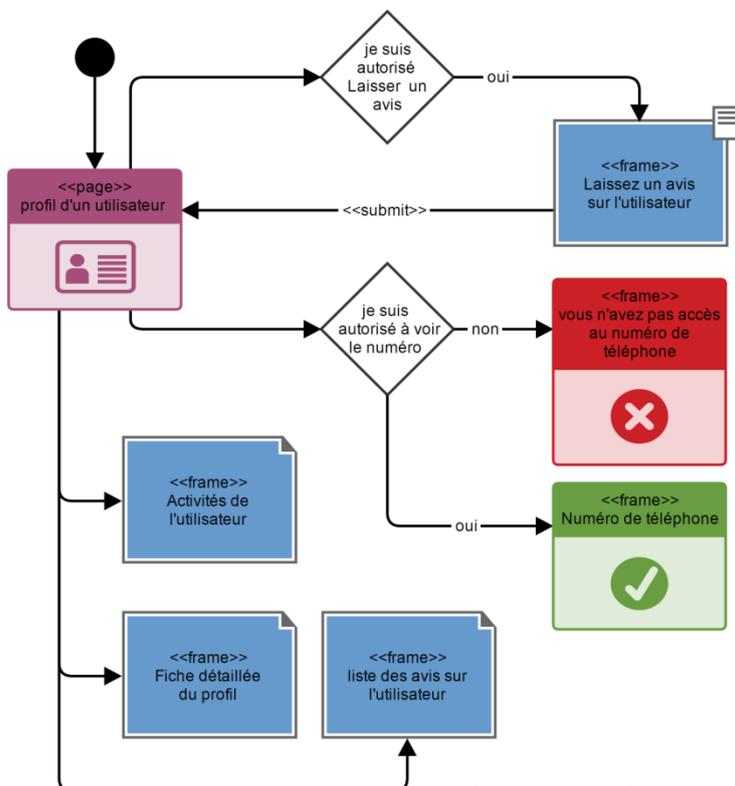


Figure 3. 19 Diagramme de navigation de la Consultation de Profil

4.2.3.4 Diagramme de navigation de la Consultation de Covoiturage :

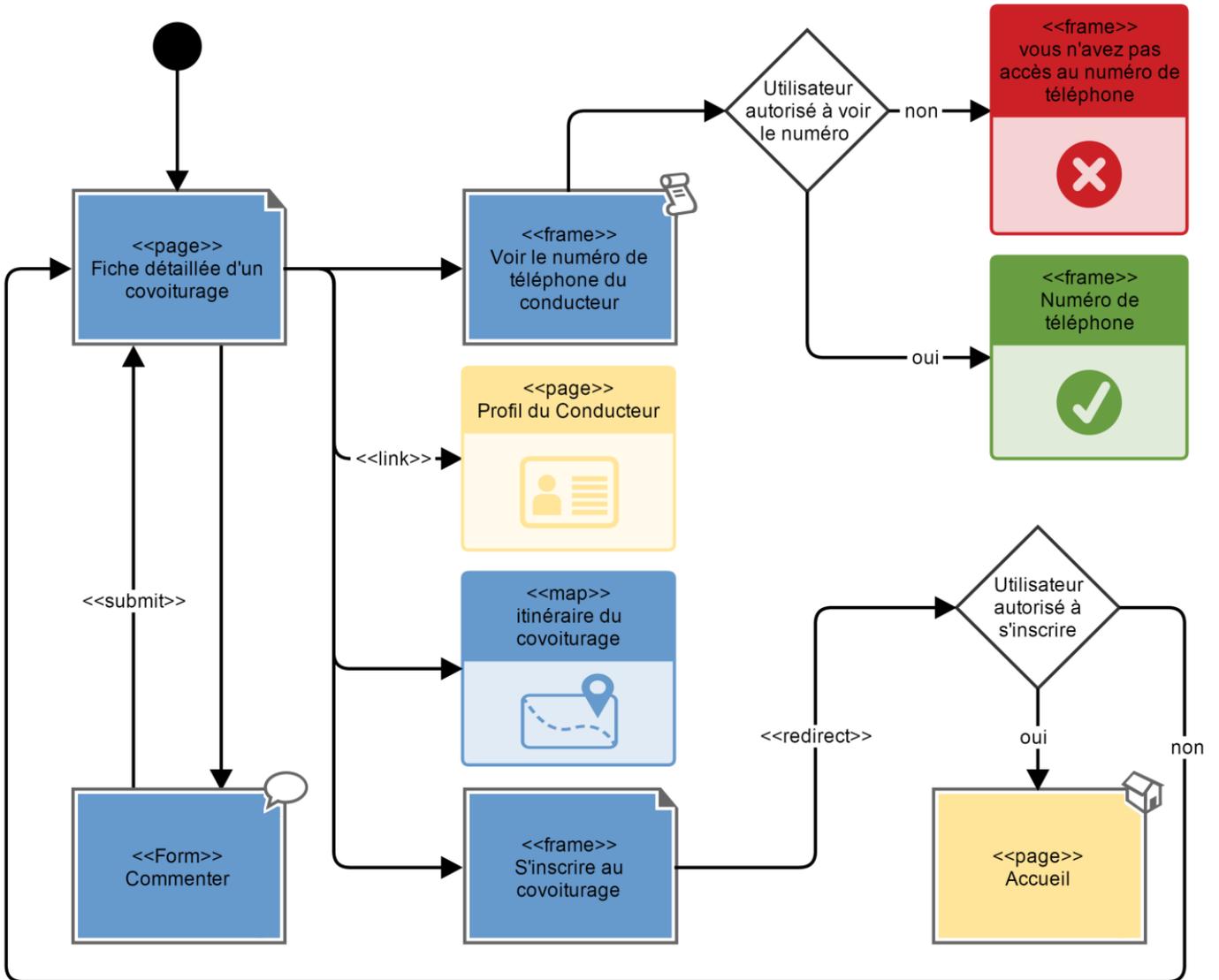


Figure 3. 20 Diagramme de navigation de la Consultation de Covoiturage

4.2.3.5 Diagramme de navigation de la page d'accueil :

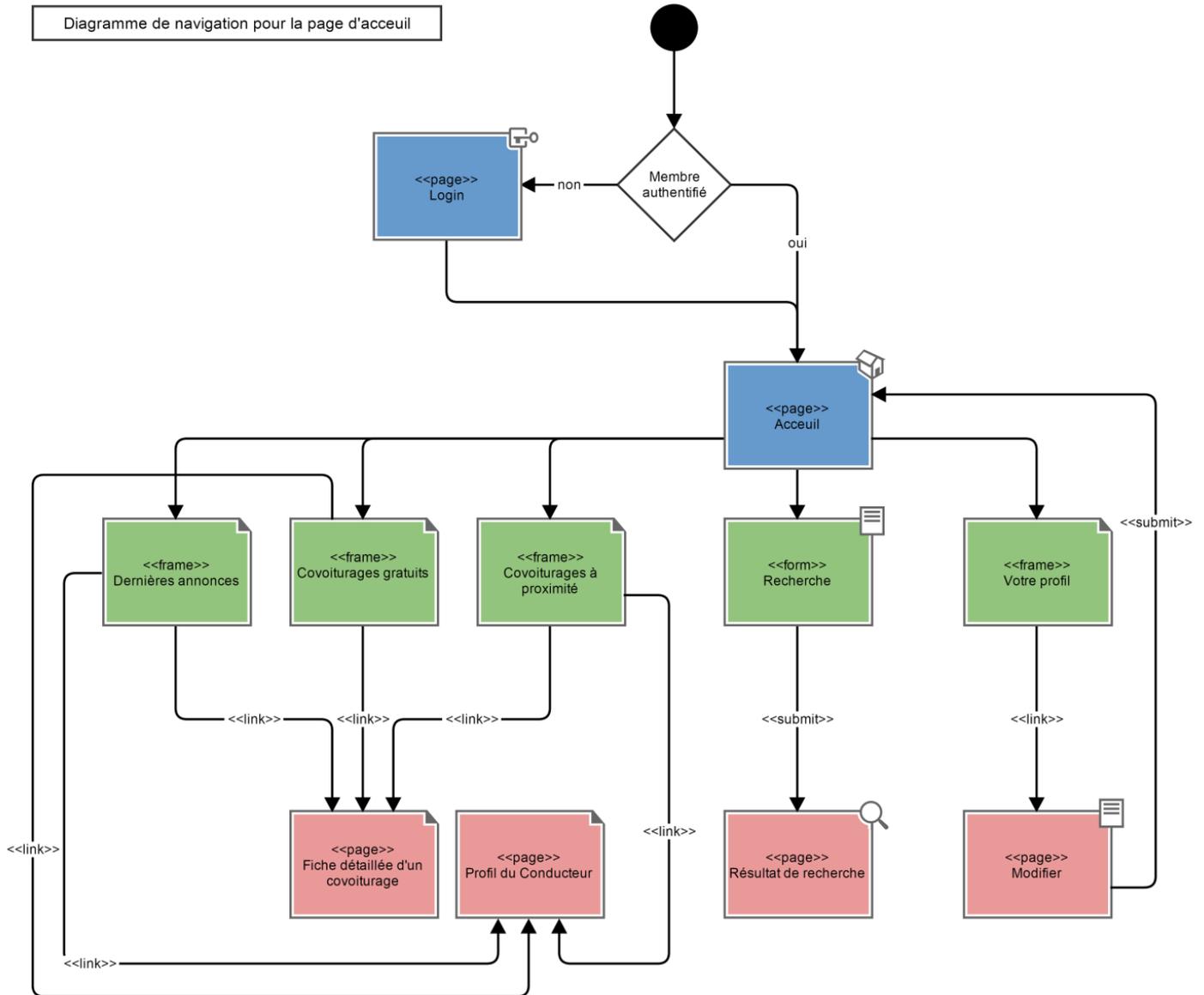


Figure 3. 21 Diagramme de navigation de la page d'accueil

4.2.4 Implémentation de bases de données :

4.2.4.1 Schéma relationnel :

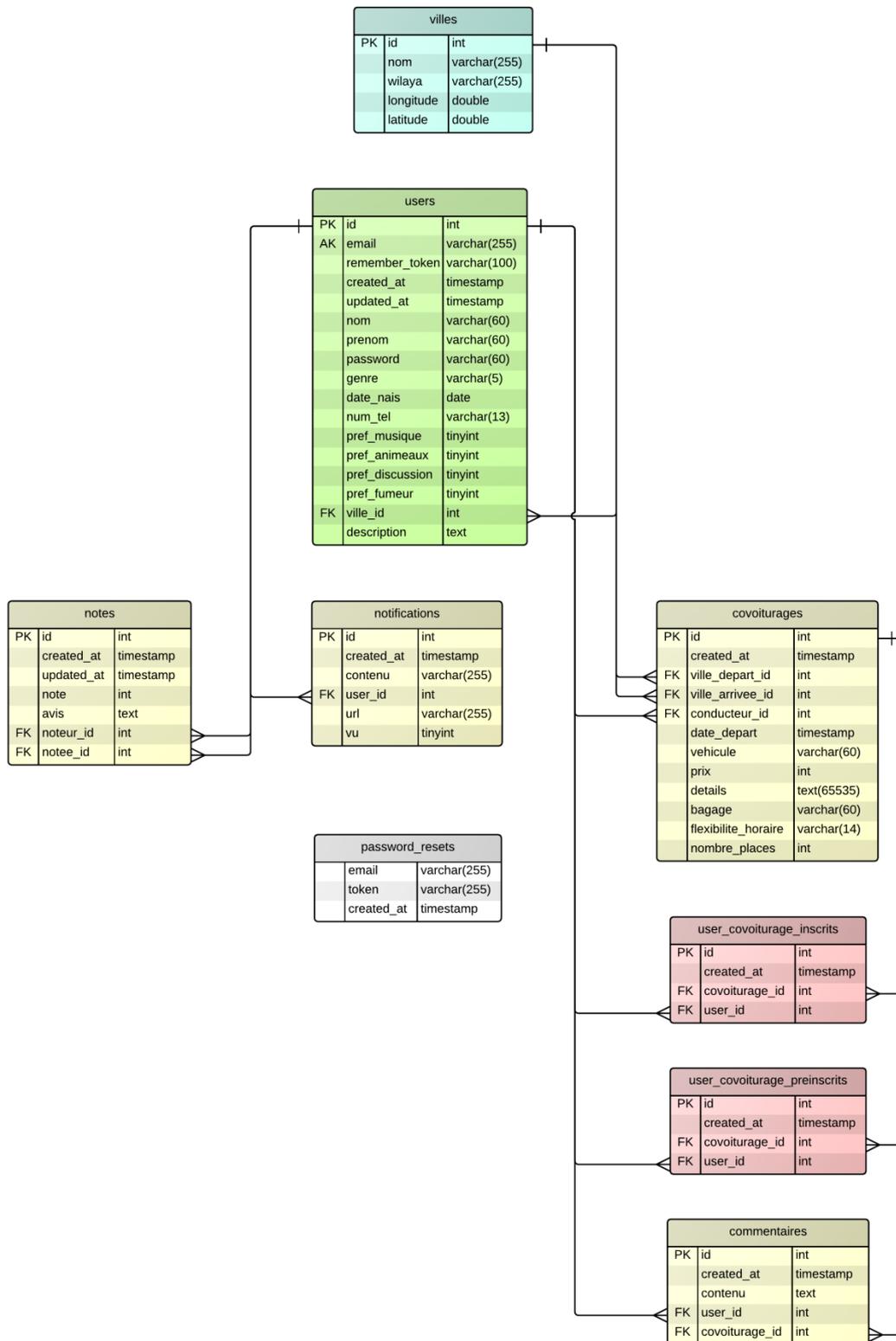


Figure 3. 23 Schéma relationnel

4.2.4.2 Structure des tables :

- Structure de la table **covoiturages**

Colonne	Type	Description	Clé
<i>id</i>	int(10)	Identifiant du covoiturage	Primaire
created_at	timestamp	Date de création du covoiturage	
ville_depart_id	int(10)	Identifiant de la ville de départ	Etrangère
ville_arrivee_id	int(10)	Identifiant de la ville d'arrivée	Etrangère
conducteur_id	int(10)	Identifiant de l'utilisateur qui a publié le covoiturage	Etrangère
date_depart	timestamp	Date de départ du covoiturage	
vehicule	varchar(60)	Modèle du véhicule	
prix	int(11)	Prix de la place	
details	text	Indications concernant le covoiturage	
bagage	varchar(6)	Taille des bagages autorisés	
flexibilite_horaire	varchar(14)	Durée d'attente tolérée par le conducteur	
nombre_places	int(11)	Nombre de places restantes	

Tableau 3: Structure de la table covoiturages

- Structure de la table **villes**

Colonne	Type	Description	Clé
<i>id</i>	int(10)	Identifiant de la ville	Primaire
nom	varchar(255)	Nom de la ville	
wilaya	varchar(255)	Nom de la Wilaya	
longitude	double(23,20)	Longitude de la ville	
latitude	double(23,20)	Latitude de la ville	

Tableau 4 : Structure de la table villes

- Structure de la table notes

Colonne	Type	Description	Clé
<i>id</i>	int(10)	Identifiant	Primaire
created_at	timestamp	Date de création de la note	
updated_at	timestamp	Date de modification de la note	
note	int(11)	Note attribuée	
avis	text	Commentaire sur le conducteur	
noteur_id	int(10)	Identifiant du passager	Etrangère
notee_id	int(10)	Identifiant du conducteur	Etrangère

Tableau 5: Structure de la table notes

- Structure de la table notifications

Colonne	Type	Description	Clé
<i>id</i>	int(10)	Identifiant de la notification	Primaire
created_at	timestamp	Date de création de la notification	
contenu	varchar(255)	Texte résumant un événement	
user_id	int(10)	Identifiant de l'utilisateur qui reçoit la notification	Etrangère
url	varchar(255)	Lien de redirection	
vu	tinyint(1)	Reçoit 1 si le lien a été cliqué 0 sinon	

Tableau 6 : Structure de la table notifications

- Structure de la table user_covoiturage_inscrits

Colonne	Type	Description	Clé
<i>id</i>	int(10)	Identifiant d'une inscription	Primaire
created_at	timestamp	Date d'inscription	
covoiturage_id	int(10)	Identifiant du covoiturage	Etrangère
user_id	int(10)	Identifiant de l'utilisateur inscrit	Etrangère

Tableau 7 : Structure de la table user_covoiturage_inscrits

- Structure de la table `user_covoiturage_preinscrits`

Colonne	Type	Description	Clé
<i>id</i>	int(10)	Identifiant d'une préinscription	Primaire
created_at	timestamp	Date de préinscription	
covoiturage_id	int(10)	Identifiant du covoiturage	Etrangère
user_id	int(10)	Identifiant de l'utilisateur préinscrit	Etrangère

Tableau 8 : Structure de la table `user_covoiturage_preinscrits`

- Structure de la table commentaires

Colonne	Type	Description	Clé
<i>id</i>	int(10)	Identifiant du commentaire	Primaire
created_at	timestamp	Date de création	
contenu	text	Contenu du commentaire	
user_id	int(10)	Identifiant de l'utilisateur qui a commenté	Etrangère
covoiturage_id	int(10)	Identifiant du covoiturage	Etrangère

Tableau 9: Structure de la table commentaires

- Structure de la table users

Colonne	Type	Description	Clé
<i>id</i>	int(10)	Identifiant de l'utilisateur	Primaire
email	varchar(255)	Adresse mail de l'utilisateur	
remember_token	varchar(100)	Token utilisé pour identifier un utilisateur ayant coché « se souvenir de moi »	
created_at	timestamp	Date d'inscription au site	
updated_at	timestamp	Date de dernière connexion	
nom	varchar(255)	Nom de l'utilisateur	
prenom	varchar(255)	Prénom de l'utilisateur	
password	varchar(60)	Mot de passe crypté	
genre	varchar(5)	Homme ou Femme	
date_nais	date	Date de naissance	
num_tel	varchar(13)	Numéro de téléphone	
pref_musique	tinyint(1)	1 si le conducteur accepte d'écouter de la musique, 0 sinon	
pref_animaux	tinyint(1)	1 si le conducteur accepte la	

		présence d'animaux, 0 sinon	
pref_discussion	tinyint(1)	1 si le conducteur tolère les discussions, 0 sinon	
pref_fumeur	tinyint(1)	1 si le conducteur accepte les fumeurs, 0 sinon	
ville_id	int(10)	Identifiant de la ville ou habite l'utilisateur	Etrangère
description	text	Texte de présentation de l'utilisateur	

Tableau 10 : Structure de la table users

Conclusion :

Ce chapitre s'est porté sur l'analyse et la conception de notre projet en utilisant le langage semi formel UML.

Maintenant que les acteurs sont identifiés et que leurs taches nous sont distinctes et détaillées nous pouvons passer à la réalisation.

Le prochain chapitre portera sur les technologies utilisées pour l'implémentation de l'application CovoiturageDZ et révélera quelques interfaces.

Chapitre IV

Mise en œuvre et Réalisation

Introduction

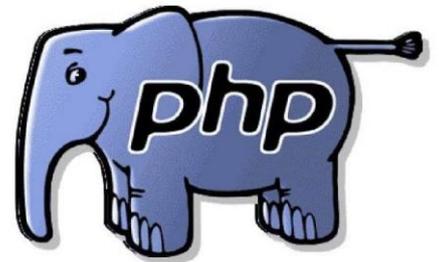
Pour tout développement d'application, il est nécessaire de choisir les technologies et outils adéquats pour faciliter la réalisation. Dans ce chapitre nous allons présenter les technologies et différents outils utilisés afin d'en justifier le choix.

I. Les Outils et technologies

1. PHP : Hypertext Preprocessor

1.1 Présentation :

Plus connu sous son sigle **PHP**, est un langage de programmation libre principalement, utilisé pour produire des pages Web dynamiques via un serveur http, c'est un langage interprété (un langage de script) exécuté du côté serveur, dans cette architecture, le serveur interprète le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS par exemple) et des données (JPEG, GIF, PNG par exemple) pouvant être interprétés et rendus par un navigateur. PHP peut également générer d'autres formats comme le SVG, le PDF ...



Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le Web. PHP est le plus souvent couplé à un serveur Apache bien qu'il puisse être installé sur la plupart des serveurs HTTP. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.

C'est un langage peu typé et souple et donc facile à apprendre par un débutant mais, de ce fait, des failles de sécurité peuvent rapidement apparaître dans les applications. Pragmatique, PHP ne s'encombre pas de théorie et a tendance à choisir le chemin le plus direct. Néanmoins, le nom des fonctions (ainsi que le passage des arguments) ne respecte pas toujours une logique uniforme, ce qui peut être préjudiciable à l'apprentissage.

Son utilisation commence avec le traitement des formulaires puis par l'accès aux bases de données. L'accès aux bases de données est aisé une fois l'installation des modules correspondants effectuée sur le serveur. La force la plus évidente de ce langage est qu'il a permis au fil du temps la résolution aisée de problèmes autrefois compliqués et est devenu par conséquent un composant incontournable des offres d'hébergements.

Il est multiplateforme : autant sur Linux qu'avec Windows il permet aisément de reconduire le même code sur un environnement à peu près semblable (prendre en compte les règles d'arborescences de répertoires qui peuvent changer).

Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite comme tout langage de programmation une bonne compréhension des principales fonctions usuelles ainsi qu'une connaissance aigüe des problèmes de sécurité liés à ce langage.

La version 5.3 a introduit de nombreuses fonctionnalités : les espaces de noms — un élément fondamental de l'élaboration d'extensions, de bibliothèques et de frameworks structurés, les fonctions anonymes, les fermetures, etc.

Aujourd'hui près de 80% des sites internet utilisent le langage PHP sous ses différentes versions. Plusieurs développeurs PHP responsables de ces sites utilisent en majorité la version 5.4 (38%) dans leurs missions quotidiennes.

1.2 Fonctionnement :

PHP appartient à la grande famille des descendants du C, dont la syntaxe est très proche. En particulier, sa syntaxe et sa construction ressemblent à celles des langages Java, à la différence que du code PHP peut facilement être mélangé avec du code HTML au sein d'un fichier PHP.



Figure 4. 1 Coopération HTML/CSS

Dans une utilisation Web, l'exécution du code PHP se déroule ainsi : lorsqu'un visiteur demande à consulter une page Web, son navigateur envoie une requête au serveur HTTP correspondant. Si la page est identifiée comme un script PHP (généralement grâce à l'extension .php), le serveur appelle l'interprète PHP qui va traiter et générer le code final de la page (constitué généralement d'HTML, mais aussi souvent de CSS et de JS). Ce contenu est renvoyé au serveur HTTP, qui l'envoie finalement au client.

2. HTML5 et CSS3

2.1 **HTML** est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade).



2.2 **HTML5** (HyperText Markup Language 5) est la dernière révision majeure d'HTML (format de données conçu pour représenter les pages web). Cette version a été finalisée le 28 octobre 2014, Le langage comprend une couche application avec de nombreuses API, ainsi qu'un algorithme afin de pouvoir traiter les documents à la syntaxe non conforme.

2.3 **CSS** (Cascading Style Sheets, aussi appelées Feuilles de style) : le rôle du CSS est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...). Ce langage est venu compléter le HTML en 1996. L'un des objectifs majeurs des CSS est de permettre la mise en forme hors des documents. Il est par exemple possible de ne décrire que la structure d'un document en HTML, et de décrire toute la présentation dans une feuille de style CSS séparée. Les styles sont appliqués au dernier moment, dans le navigateur web des visiteurs qui consultent le document. Cette séparation fournit un certain nombre de bénéfices, permettant d'améliorer l'accessibilité, de changer plus facilement de présentation, et de réduire la complexité de l'architecture d'un document.

Ainsi, les avantages des feuilles de style sont multiples :

- La structure du document et la présentation peuvent être gérées dans des fichiers séparés ;
- La conception d'un document se fait dans un premier temps sans se soucier de la présentation, ce qui permet d'être plus efficace ;
- Dans le cas d'un site web, la présentation est uniformisée : les documents (pages HTML) font référence aux mêmes feuilles de styles. Ceci permet de plus une remise en forme rapide de l'aspect visuel ;
- Un même document peut donner le choix entre plusieurs feuilles de style, par exemple une pour l'impression et une pour la lecture à l'écran. Certains navigateurs web permettent au visiteur de choisir un style parmi plusieurs ;
- Le code HTML est considérablement réduit en taille et en complexité, puisqu'il ne contient plus de balises ni d'attributs de présentation.

3. JavaScript

JavaScript est un langage interprété par le navigateur. Le JavaScript est un langage « client », c'est-à-dire exécuté chez l'utilisateur lorsque la page Web est chargée. Il a pour but de dynamiser les sites Internet. JavaScript est un langage objet : chaque objet possède des méthodes (ou fonctions), des propriétés et des objets. Dans une page Web, l'objet le plus élevé dans la hiérarchie est la fenêtre du navigateur : window. Cet objet window contient entre autres l'objet document qui lui même contient tous les objets contenus dans la page Web (paragraphes, formulaires, etc...). En plus de ces objets, il existe des objets créés par l'utilisateur.



4. SQL

SQL (acronyme de **Structured Query Language**, en français langage de requête structurée) est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des



données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

Outre le langage de manipulation des données, la partie langage de définition des données de créer et de modifier l'organisation des données dans la base de données, la partie langage de contrôle de transaction permet de commencer et de terminer des transactions, et la partie langage de contrôle des données permet d'autoriser ou d'interdire l'accès à certaines données à certaines personnes.

Créé en 1974, normalisé depuis 1986, le langage est reconnu par la grande majorité des systèmes de gestion de bases de données relationnelles (abrégé SGBDR) du marché.

4.1 MySQL :

MySQL est un système de base de données relationnelle couronné de succès et commercialisé aussi bien comme projet open source que comme produit Enterprise. Développé à l'origine par la société suédoise MySQL AB, il a été repris plus tard par Sun Microsystems et enfin par Oracle. Le système se base sur un serveur MySQL qui est optimisé pour le stockage rapide, ainsi que l'interrogation d'enregistrements de données contenus dans des bases de données. Ces bases de données contiennent à leur tour plusieurs tableaux se rapportant les uns aux autres.

5. PhpStorm

PhpStorm est un environnement de développement, aussi appelé IDE, développé par la société JetBrains s.r.o. depuis 2009. Écrit en Java, il permet d'éditer du code en PHP, en HTML et en JavaScript. Assurant une compatibilité avec PHP 5.5, ce logiciel permet d'éditer du code source PHP de manière intuitive.



En effet, cet IDE propose la coloration syntaxique et la complétion automatique du code afin d'afficher les classes, les méthodes et les variables en un clin d'œil. Pour une meilleure navigation, il dispose de fonctionnalités de recherche avancée.

PhpStorm propose également des outils de contrôle à distance et plusieurs frameworks PHP. Le logiciel bénéficie aussi du support de SQL, d'un terminal et d'un système de contrôle de versions. Des fonctions de débogage et de test donnent les moyens aux développeurs de configurer au mieux leurs applications.

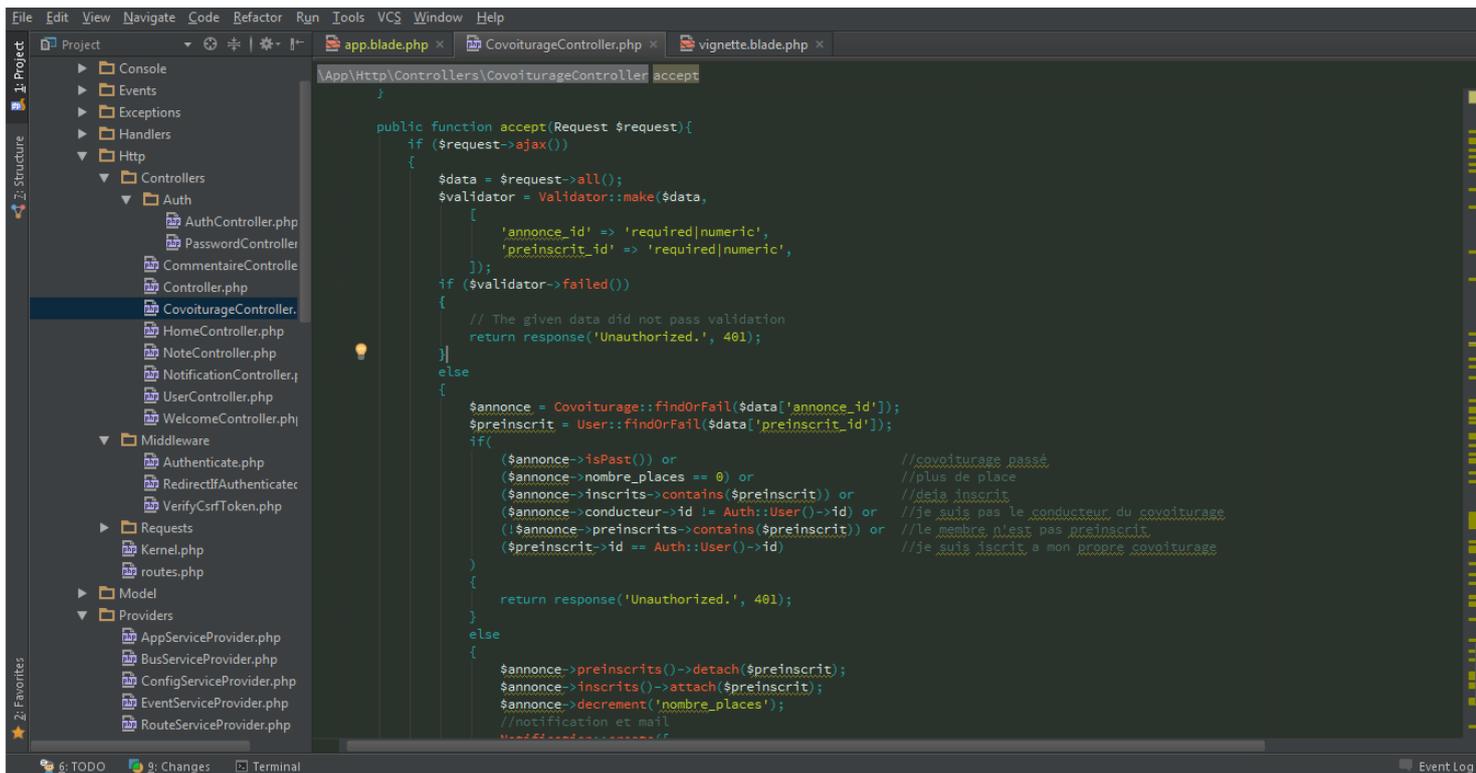


Figure 4. 2 Interface de l'IDE PhpStorm

6. Git et GitHub

C'est un logiciel de gestion de versions (ou VCS en anglais, pour Version Control System) il permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées dessus. Il permet notamment de retrouver les différentes versions.



C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU.

Une des particularités de Git, c'est l'existence de sites web collaboratifs basés sur Git comme GitHub, par exemple, GitHub est très connu et utilisé par de nombreux projets open source : jQuery, Symfony, Ruby on Rails...

Le code source du projet est disponible sur un dépôt GitHub a cette adresse :

<https://github.com/aribyassine/MEMOIRE-DE-FIN-D-ETUDE>

7. Composer

Composer est un gestionnaire de dépendances open source écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin.



L'outil est relativement simple à utiliser. Il suffit de l'installer puis de déclarer un fichier JSON listant les différentes bibliothèques que l'on souhaite implémenter, puis une simple ligne de commande permet de mettre à jour les bibliothèques du projet:

```
> $ composer update
```

Composer est fortement inspiré de npm pour Node.js et de bundler pour Ruby.

II. Frameworks et APIs

1. Définition d'un framework

Un framework ou structure logicielle est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture).

L'utilité d'un framework est d'éviter de passer du temps à développer ce qui a déjà été fait par d'autres souvent plus compétents et qui a en plus été utilisé et validé par de nombreux utilisateurs. On peut imaginer un framework comme un ensemble d'outils à disposition.

2. LARAVEL



Laravel est un framework web open-source écrit en PHP respectant le principe MVC et entièrement développé en POO. Il est devenu à la suite de sa version 4 l'un des frameworks PHP les plus utilisés et les plus reconnus au monde.

Créé par Taylor Otwell, Laravel initie une nouvelle façon de concevoir un framework en utilisant ce qui existe de mieux pour chaque fonctionnalité ainsi que de nouveaux composants exclusifs. Parmi eux :

- un système de routage perfectionné,
- un créateur de requêtes SQL et un ORM performants,
- un moteur de template efficace,
- un système d'authentification pour les connexions,
- un système de validation intégré,
- un système de migration pour les bases de données,
- un système d'envoi d'emails,
- une gestion des sessions...

La version Laravel que nous avons adopté pour la réalisation de notre projet est la 5, elle nécessite la version 5.4 de PHP qui comprend les notions suivantes :

- **Les espaces de noms** : c'est une façon de bien ranger le code pour éviter des conflits de nommage. Laravel utilise cette possibilité de façon intensive. Tous les composants sont rangés dans des espaces de noms distincts.

```

<?php namespace App\Http\Controllers;

use App\Model\Commentaire;
use App\Model\Covoiturage;
use App\Model\Notification;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class CommentaireController extends Controller {

```

- **Les fonctions anonymes** : ce sont des fonctions sans nom (souvent appelées closures) qui permettent d'améliorer le code. Les utilisateurs de Javascript y sont habitués. Laravel les utilise aussi de façon systématique.

```

$covoituages = $covoituages->filter(function($covoituage){
    return $covoituage->date_depart > date("Y-m-d H:i:s");
});

```

- **Les méthodes magiques** : ce sont des méthodes qui n'ont pas été explicitement décrites dans une classe mais qui peuvent être appelées et résolues.
- **Les interfaces** : une interface est un contrat de constitution des classes. En programmation objet c'est le sommet de la hiérarchie. Tous les composants de Laravel sont fondés sur des interfaces.
- **Les traits** : c'est une façon d'ajouter des propriétés et méthodes à une classe sans passer par l'héritage, ce qui permet de passer outre certaines limitations de l'héritage simple proposé par défaut par PHP.

```

\Illuminate\Foundation\Auth\AuthenticatesAndRegistersUsers

<?php namespace Illuminate\Foundation\Auth;

+use ...

trait AuthenticatesAndRegistersUsers {

```

2.1 Le modèle MVC

2.1.1 Présentation

Le modèle MVC décrit une manière d'architecturer une application informatique en la décomposant en trois sous-parties :

- la partie Modèle ;
- la partie Vue ;
- la partie Contrôleur.

Ce modèle de conception (« *design pattern* ») a été imaginé à la fin des années 1970 afin de bien séparer le code de l'interface graphique de la logique applicative. Il est utilisé dans de très nombreux langages : bibliothèques Swing et Model 2 (JSP) de Java, *frameworks* PHP, ASP.NET MVC, etc.

2.1.2 Rôles des composants

- La partie Modèle d'une architecture MVC encapsule la logique métier (« business logic ») ainsi que l'accès aux données. Il peut s'agir d'un ensemble de fonctions (Modèle procédural) ou de classes (Modèle orienté objet).
- La partie Vue s'occupe des interactions avec l'utilisateur : présentation, saisie et validation des données.
- La partie Contrôleur gère la dynamique de l'application. Elle fait le lien entre l'utilisateur et le reste de l'application.

2.1.3 Interaction entre les composants

1. La demande de l'utilisateur (exemple : requête HTTP) est reçue et interprétée par le **Contrôleur**.
2. Celui-ci utilise les services du **Modèle** afin de préparer les données à afficher.
3. Ensuite, le **Contrôleur** fournit ces données à la **Vue**, qui les présente à l'utilisateur (par exemple sous la forme d'une page HTML).
- 4.

Une application construite sur le principe du MVC se compose toujours de trois parties distinctes. Cependant, il est fréquent que chaque partie soit elle-même décomposée en plusieurs éléments. On peut ainsi trouver plusieurs modèles, plusieurs vues ou plusieurs contrôleurs à l'intérieur d'une application MVC.

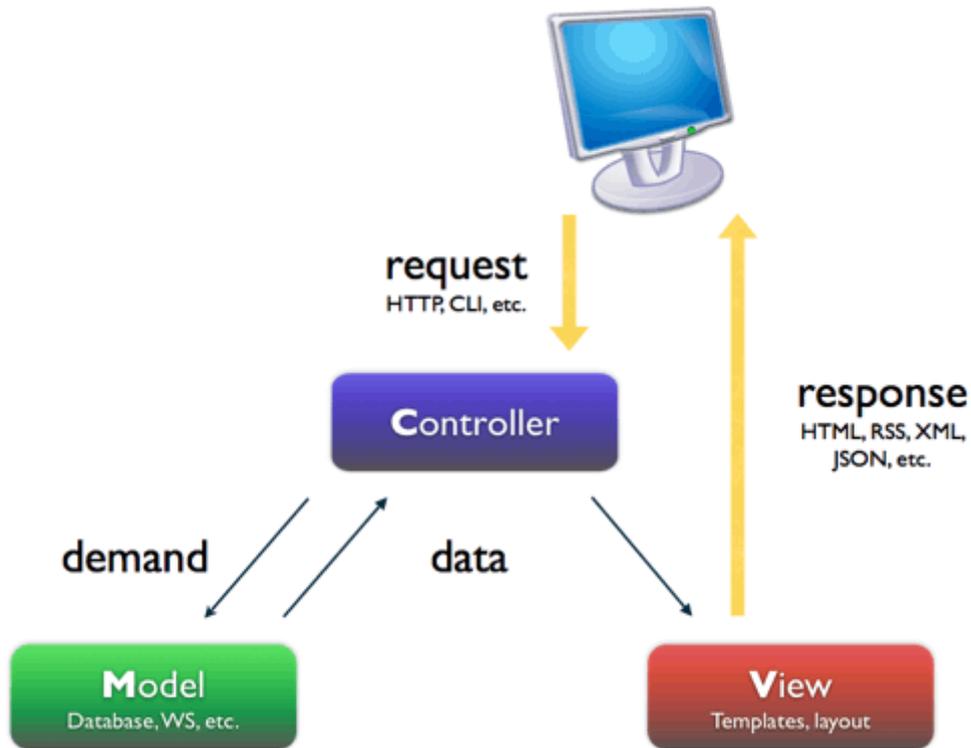


Figure 4. 3 Interactions entre les composants MVC

2.1.4 Avantages et inconvénients

Le modèle MVC offre une séparation claire des responsabilités au sein d'une application. Le prix à payer est une augmentation de la complexité de l'architecture.

Dans le cas d'une application Web, l'application du modèle MVC permet aux pages HTML (qui constituent la partie Vue) de contenir le moins possible de code serveur, étant donné que le scripting est regroupé dans les deux autres parties de l'application.

En général on résume en disant que le modèle gère la base de données, la vue produit les pages HTML et le contrôleur fait tout le reste. Dans Laravel :

- le modèle correspond à une table d'une base de données. C'est une classe qui étend la classe Model qui permet une gestion simple et efficace des manipulations de données et l'établissement automatisé de relations entre tables.
- le contrôleur s'occupe de la logique métier de l'application (traiter les requêtes, validation, traitement des données, passer les données à la vue correspondante...)
- la vue est soit un simple fichier avec du code HTML, soit un fichier utilisant le système de template Blade de Laravel.

2.2 La programmation orientée objet

Laravel est fondamentalement orienté objet. La POO est un design pattern qui s'éloigne radicalement de la programmation procédurale. Avec la POO tout le code est placé dans des classes qui découlent d'interfaces qui établissent des contrats de fonctionnement.

Avec la POO la responsabilité du fonctionnement est répartie dans des classes alors que dans l'approche procédurale tout est mélangé. Le fait de répartir la responsabilité évite la duplication du code qui est le lot presque forcé de la programmation procédurale.

L'utilisation de classes bien identifiées, dont chacune a un rôle précis, crée un code élégant, efficace, lisible, facile à maintenir et à tester. C'est ce que Laravel propose.

2.3 Le routage

Il constitue le point d'entrée unique de l'application. Son rôle est de centraliser la gestion des requêtes entrantes. Il utilise le service d'un autre contrôleur pour réaliser l'action demandée et renvoyer son résultat sous la forme d'une vue.

Le routage d'une requête entrante consiste à analyser cette requête afin d'en déduire le contrôleur à utiliser et l'action (méthode du contrôleur) à appeler.

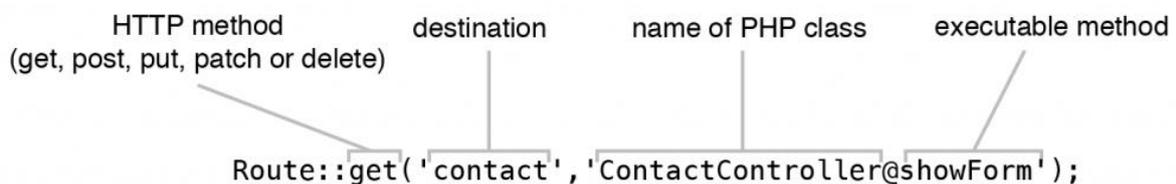


Figure 4. 4 Structure d'une route

2.4 L'ORM Eloquent

L'une des tâches les plus courantes et difficiles pour toute application consiste à lire et à persister des informations dans une base de données. Heureusement, Laravel intègre Eloquent, une bibliothèque dont le seul but est de fournir des outils puissants afin de faciliter la tâche.

Un mapping objet-relationnel (en anglais object-relational mapping ou ORM) est une technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle en définissant des correspondances entre cette base de données et les objets du langage utilisé. On pourrait le désigner par « correspondance entre monde objet et monde relationnel ».

Le mapping objet-relationnel consiste à associer une ou plusieurs classes avec une table, et chaque attribut de la classe avec un champ de la table.

Par exemple, la classe Covoiturage est associée avec la table covoiturages, et les attributs sont associés comme suit :

Covoiturage->id est associée avec covoiturages.id

Covoiturage->prix est associée avec covoiturages.prix

Covoiturage-> nombre_places est associée avec covoiturages. nombre_places

2.4.1 Les avantages

L'ORM permet tout d'abord de simplifier grandement l'accès aux données. Chaque tuple devient une instance d'objet et les méthodes de modification sur les données deviennent uniformisées. Un autre avantage de l'ORM est de rendre l'accès aux données complètement indépendant du SGBD utilisé. Il devient donc très simple de changer de SGBD au cours du développement de l'application.

2.4.2 Un simple exemple

La manière la plus facile de comprendre comment Eloquent fonctionne est de le voir en action.

```
Auth::User()->conducteurCovoiturages()->first()->inscrits
```

Ceci correspond à récupérer les inscrits du premier covoiturage dont l'utilisateur authentifié est le conducteur, la requête SQL serait donc :

```
select * from `users` where `users`.`id` = '2' limit 1
select * from `covoiturages` where `covoiturages`.`conducteur_id` = '2' and
`covoiturages`.`conducteur_id` is not null limit 1
select `users`.*, `user_covoiturage_inscrits`.`covoiturage_id` as `pivot_covoiturage_id`,
`user_covoiturage_inscrits`.`user_id` as `pivot_user_id`,
`user_covoiturage_inscrits`.`created_at` as `pivot_created_at`,
`user_covoiturage_inscrits`.`updated_at` as `pivot_updated_at` from `users` inner join
`user_covoiturage_inscrits` on `users`.`id` = `user_covoiturage_inscrits`.`user_id` where
`user_covoiturage_inscrits`.`covoiturage_id` = '3'
```

2.5 Le moteurs de templates Blade

2.5.1 Définition :

Un gabarit, souvent nommé avec les anglicismes *template* ou *layout*, est un patron de mise en page où l'on place images et textes. Aussi, il est souvent utilisé de manière répétitive pour créer des documents présentant une même structure.

Les moteurs de templates sont les outils utilisant les templates pour en générer les fichiers textes (HTML, code source, etc.). Chacun dispose de son propre langage de programmation limité permettant de décrire le template, à l'aide de structures de contrôle (telles que `if`, `foreach`, etc.) ainsi que de réaliser des opérations simples sur les données (récupérer la sous-chaîne d'une chaîne de caractères, supprimer des caractères indésirables pour éviter des attaques de type *cross-site scripting* sur un site web, etc.). Ces langages étant la plupart du temps interprétés et dynamiques, ils sont considérés comme des langages de script.

Un point fort de Laravel, son moteur de template, Blade. Contrairement à d'autres moteurs de template comme le célèbre Twig, Blade utilise PHP pour son rendu, ce qui veut dire qu'en plus des tags préfourmis il est à tout moment possible d'ouvrir des balises PHP et d'y saisir ce qu'on veut. Il a aussi l'avantage d'être très rapide : Blade recompile les fichiers uniquement si des changements sont apportés dans le fichier de template.

2.5.2 Avantages

- La séparation des deux codes permet une meilleure visibilité dans le code. Idéal pour le travail d'équipe.
- On peut alors toucher ou modifier un des deux fichiers sans que cela ait un impact sur l'autre.
- La mise en cache est proposée et permet ainsi d'économiser les ressources des serveurs.

2.6 Les middlewares

Un middleware effectue un traitement à l'arrivée de la requête ou à son départ. Par exemple la gestion des sessions ou des cookies dans Laravel se fait dans un middleware, ainsi que l'authentification.

Les middlewares HTTP fournissent un mécanisme pratique pour les demandes de filtrage HTTP entrant dans une application. Par exemple, Laravel comprend un middleware qui vérifie l'utilisateur de votre application est authentifié ('auth'). Si l'utilisateur n'a pas authentifié ('guest'), le middleware se redirige l'utilisateur vers l'écran de connexion.

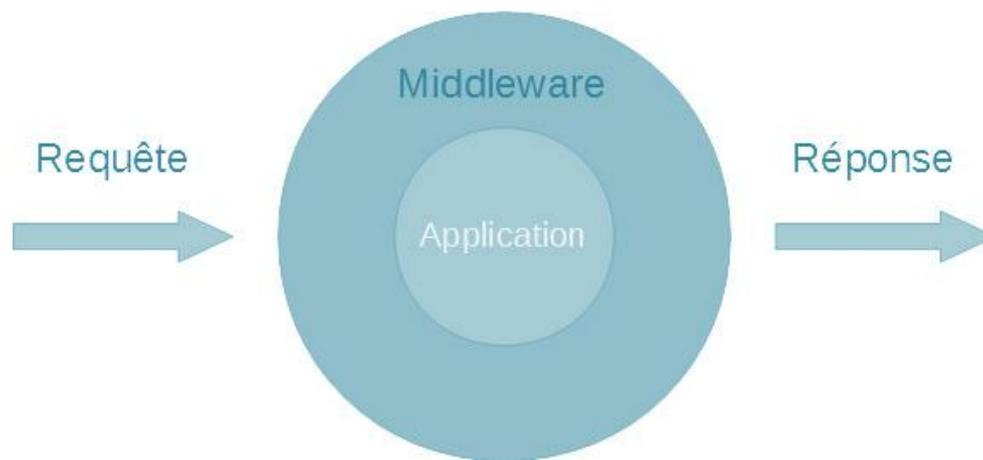


Figure 4. 5 Schématisation d'un middleware

3. Sécurité

Jusqu'à présent, nous n'avons que très peu abordé les problématiques liées à la sécurité du site. Il s'agit d'un vaste sujet. En matière de développement Web, la règle de sécurité essentielle est de ne jamais faire confiance aux données reçues. Le Web est un monde ouvert où il est facile de forger une requête HTTP contenant des paramètres intentionnellement incorrects. Deux exemples fréquents sont l'injection de code sur la page Web finale (cross-site scripting ou XSS) et l'injection SQL (exécution par le SGBD de requêtes SQL non prévues). Afin de se prémunir contre de telles attaques, il est indispensable de « nettoyer » les données reçues ou affichées par le site.



3.1 Les injections SQL

Concernant les injections SQL visant la base de données, la solution consiste à utiliser des requêtes paramétrées plutôt que de construire la requête SQL par concaténation

d'instructions et de paramètres. Dans ce cas, L'ORM Eloquent effectue un nettoyage automatique des paramètres. Notre partie Modèle utilise déjà des requêtes paramétrées : nous n'avons donc pas de précaution supplémentaire à prendre à ce sujet.

3.2 Hachage de mots de passe

Notre application PHP se base sur une authentification de l'utilisateur. Les identifiants et les mots de passe sont stockés dans une base de données et utilisés plus tard pour authentifier les utilisateurs.

Il est important d'utiliser correctement les fonctions de hachage avant de les stocker. Le hachage de mot de passe est une opération irréversible produisant une chaîne de caractères de longueur fixe. Cela signifie qu'on peut comparer le produit d'une fonction de hachage avec le hash stocké en base de données pour déterminer s'il s'agit du même texte. Si les mots de passe stockés en base de données ne sont pas "hachés" alors n'importe qui ayant accès à cette base peut compromettre les comptes utilisateurs. Il arrive souvent que les utilisateurs utilisent le même mot de passe pour d'autres services, c'est pourquoi il faut prendre la sécurité des informations avec sérieux.

3.3 Filtrage et validation des données

Une règle d'or: ne jamais faire confiance aux entrées extérieures dans votre code PHP. L'objet Validator peut nettoyer les entrées textuelles et valider les données.

Les données peuvent être filtrées différemment selon le contexte. Par exemple, quand des données brutes sont envoyées en sortie vers la page HTML, elles peuvent exécuter du Javascript et de l'HTML. Cette technique est connue sous le nom de "Cross-Site Scripting" (XSS) et peut se révéler très dangereux. Une façon d'éviter les attaques XSS est de nettoyer toutes les données générées par l'utilisateur avant de les afficher sur votre page en retirant toutes balises HTML avec la syntaxe du moteur de templates Blade :

```

<div class="panel-body">
  <div class="text-center alert alert-info">
    <button type="button" class="close" data-dismiss="alert" aria-label="Close">
      <span aria-hidden="true">&times;</span>
    </button>
    @if($covoiturages->count() == 0)
      Aucun résultat trouvé
    @elseif($covoiturages->count() == 1)
      Un seul résultat trouvé
    @else
      {{ $covoiturages->count() }} résultats trouvés
    @endif

    @if(!empty($date))
      pour le {{ $date }}
    @endif
  </div>
  @include('covoiturage.vignette', compact('covoiturages','path'))
</div>

```

3.4 Validation

La validation s’assure que les entrées extérieures correspondent à ce que nous nous attendions. Par exemple, nous pourrions vouloir valider une adresse email, un numéro de téléphone ou une donnée de géolocalisation lors du traitement de l’enregistrement d’un compte.

3.5 Rapport d’erreurs

La journalisation des erreurs peut être utile pour repérer les points qui posent problème dans notre application pour pouvoir les corriger. Les erreurs seront toujours enregistrées dans les journaux des erreurs de notre serveur web mais ne seront pas afficher à l’utilisateur.

4. Bootstrap

Twitter Bootstrap est une collection d’outils utiles à la création de sites web et 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.



Cette plate-forme a été conçue par deux développeurs faisant partie de la mouvance de développeurs qui gravitent autour de Twitter, Mark Otto et Jacob Thornton.

Bootstrap est aussi un ensemble de composants structurés qui sert à créer les fondations et à organiser le code source pour faciliter le travail des intégrateurs, que ce soit en termes de productivité ou de simplification de la maintenance. Cela permet une fois que l'on a bien structuré sa page de gagner du temps lors de la création de formulaire par exemple. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

4.1. Quel en est l'intérêt ?

- Les navigateurs sont pleins de fantaisies et ont des comportements très différents malgré leur lente convergence vers les standards. Bootstrap est cross-browser, c'est à dire que la présentation est similaire quel que soit le navigateur utilisé et d'une parfaite compatibilité.
- Bootstrap fait gagner du temps de développement parce qu'ils nous proposent les fondations de la présentation.
- normalise la présentation en proposant un ensemble homogène de styles.
- propose une grille pour faciliter le positionnement des éléments.
- Bootstrap offre des éléments complémentaires : boutons esthétiques, barres de navigation, etc...
- La grande diffusion de nouveaux moyens de visualisation du web (smartphones, tablettes...) impose désormais la prise en compte de tailles d'écran

Contenu du kit, Bootstrap propose :

- Une mise en page basée sur une grille de 12 colonnes bien pratique. Du code fondé sur HTML5 et CSS3 ;
- Une bibliothèque totalement open source sous licence Apache ;
- Du code qui tient compte du format d'affichage des principaux outils de navigation (responsive design) : smartphones, tablettes... ;
- Des plugins jQuery de qualité ;
- Une bonne documentation ;
- La garantie d'une évolution permanente ;
- Une mine de ressources variées sur le web ;

4.2. Responsive Web Design

Plus personne ne doute aujourd'hui que l'usage des smartphones est en train d'exploser. De plus en plus de personnes visitent aujourd'hui des sites web depuis leur navigateur mobile.

Un site web adaptatif (anglais RWD pour responsive web design) est un site web dont la conception vise, grâce à différents principes et technologies, à offrir une expérience de consultation confortable même pour des supports différents. L'utilisateur peut ainsi consulter le même site web à travers une large gamme d'appareils (moniteurs d'ordinateur, smartphones, tablettes, TV, etc.) avec le même confort visuel et sans avoir recours au défilement horizontal ou au zoom avant/arrière sur les appareils tactiles notamment, manipulations qui peuvent parfois dégrader l'expérience utilisateur, tant en lecture qu'en navigation.

4.3. Le principe du responsive web design

C'est une technique de réalisation de site web, qui permet de faire des sites qui s'adaptent automatiquement à l'espace disponible sur l'écran.

En général, on se base sur la largeur de l'écran pour déterminer comment les éléments du site doivent s'afficher :

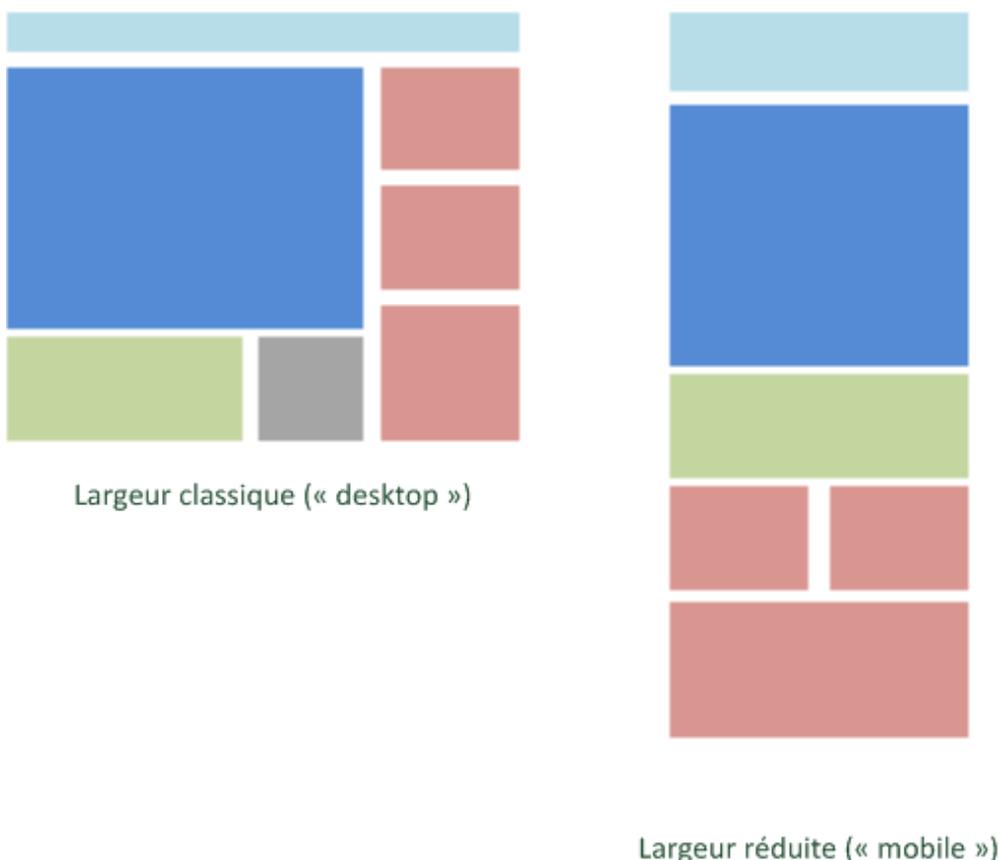


Figure 4. 6 Exemple de page adaptative

Quand on dispose de moins de place en largeur, on peut mettre moins de blocs côte à côte. Dans ce cas, on les positionne plutôt de haut en bas.



5. jQuery

jQuery est une bibliothèque JavaScript libre et multi-plateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. La première version est lancée en janvier 2006 par John Resig.

La bibliothèque contient notamment les fonctionnalités suivantes :

- Parcours et modification du DOM (y compris le support des sélecteurs CSS 3);
- Événements ;
- Effets visuels et animations ;
- Manipulations des feuilles de style en cascade (ajout/suppression des classes, d'attributs...);
- Ajax ;

6. API Google MAP

GoogleMaps est une application Web recouvrant un outil d'annuaire et des données cartographiques mondiales. La base de données de Google Map peut être interrogée comme un service de renseignement classique pour localiser un emplacement, une entreprise, calculer un itinéraire, ou à l'inverse pour rechercher visuellement des lieux préalablement indexés sur une carte.



Google Maps est le service de cartographie en ligne le plus utilisé au monde. Plus de 800,000 sites utilisent l'API et les mobinautes représentent à eux seuls plus de 250 millions d'utilisateur.

La valeur ajoutée de l'outil réside dans la souplesse d'un moteur de recherche à la google et l'intégration de la technologie ajax rend réellement efficace les rafraîchissements de pages.

On utilise dans notre application pour faciliter la saisie des adresses pour l'utilisateur. Tous les champs de texte de nos pages Web peuvent profiter de la puissance de la fonctionnalité de saisie semi-automatique de Google Maps, pour une saisie rapide, précise et simple des adresses. Mais aussi pour récupérer les coordonnées de géolocalisation, afin de les utiliser dans différents algorithmes de calcul de distance entre les villes, notamment pour la recherche. Mais aussi pour avoir une estimation de la durée d'un trajet et avoir un rendu visuel de l'itinéraire sur une carte géographique.

III. Présentation du site

1. **Page d'accueil** : A partir de la page l'internaute peut naviguer sur le site grâce à des liens qui lui permettent d'accéder à de nombreuses pages, il peut entre autre faire une recherche, consulter les dernières annonces de covoiturages et les annonces gratuites.
2. **La page d'accueil membre authentifié** : Idem pour cette page, mais aussi les annonces les plus proches de sa ville et un court résumé de son profil.
3. **La page d'inscription** Contient le formulaire d'inscription au site.
4. **La page détaillée d'un covoiturage** : Cette page contient tous les détails concernant un covoiturage particulier (itinéraire, horaire, nombre de place...) ainsi que les commentaires, une carte et des informations sur le conducteur et les passagers.
5. **La page Comment ça marche** : Interface contenant un tutoriel du fonctionnement du site
6. **La page Mes covoiturages** : Cette page permet à l'utilisateur de gérer ses annonces et réservations.
7. **La page Publier** : Formulaire de publication d'annonces.
8. **La page Profil** : Contient les informations personnelles d'un utilisateur.
9. **La page de résultats de recherche** : Affiche les annonces de covoiturages des plus proches aux plus éloignées des villes saisies.

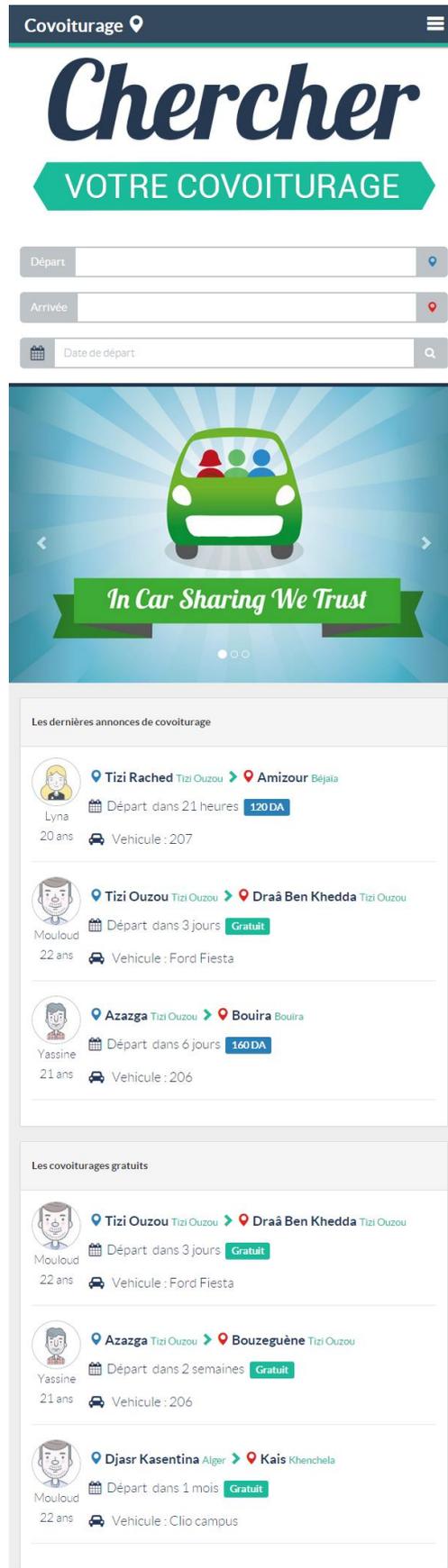


Figure 4. 7 Interface « Accueil » pour mobile

Covoiturage

[Accueil](#)
[Mes Covoiturages](#)
[Publier](#)
[Comment ça marche?](#)
🔔 1
Lyna ▾

Chercher

VOTRE COVOITURAGE

Départ:

Arrivée:

Date de départ:

Votre profil Modifier

Bouizegarene Lyna
Femme
(20 ans)

Ville: **Tizi Ouzou**

Mes préférences:

Ma bio

J'habite à Tizi-Ouzou, je vais à Alger-Centre tous les samedis dès 9h, sans détours

Ca se passe près de chez vous !

Tizi Ouzou Tizi Ouzou → **Draâ Ben Khedda** Tizi Ouzou

Mouloud
22 ans

📅 Départ dans 4 jours Gratuit

🚗 Vehicule : Ford Fiesta

Tizi Rached Tizi Ouzou → **Amizour** Béjaïa

Lyna
20 ans

📅 Départ dans 1 jour 120 DA

🚗 Vehicule : 207

Azazga Tizi Ouzou → **Bouzeguène** Tizi Ouzou

Yassine
21 ans

📅 Départ dans 2 semaines Gratuit

🚗 Vehicule : 206

Les dernières annonces de covoiturage

Tizi Rached Tizi Ouzou → **Amizour** Béjaïa

Lyna
20 ans

📅 Départ dans 1 jour 120 DA

🚗 Vehicule : 207

Tizi Ouzou Tizi Ouzou → **Draâ Ben Khedda** Tizi Ouzou

Mouloud
22 ans

📅 Départ dans 4 jours Gratuit

🚗 Vehicule : Ford Fiesta

Azazga Tizi Ouzou → **Bouira** Bouira

Yassine
21 ans

📅 Départ dans 1 semaine 160 DA

🚗 Vehicule : 206

Les covoiturages gratuits

Tizi Ouzou Tizi Ouzou → **Draâ Ben Khedda** Tizi Ouzou

Mouloud
22 ans

📅 Départ dans 4 jours Gratuit

🚗 Vehicule : Ford Fiesta

Azazga Tizi Ouzou → **Bouzeguène** Tizi Ouzou

Yassine
21 ans

📅 Départ dans 2 semaines Gratuit

🚗 Vehicule : 206

Djasr Kasentina Alger → **Kais** Khenchela

Mouloud
22 ans

📅 Départ dans 1 mois Gratuit

🚗 Vehicule : Clio campus

Figure 4. 8 Interface « Accueil » pour un membre authentifié

Covoiturage

[Publier](#) [Comment ça marche?](#) [Connexion](#) [S'inscrire](#)

Inscription

Whoops! Une erreur est survenue. ✕

- Le champ prenom doit seulement contenir des lettres.
- Le champ photo doit être un fichier de type : jpeg.
- Votre captcha n'est pas valide

Vos coordonnées

Adresse E-Mail:

Mot de passe:

Confirmation:

Nom:

Prénom:

Civilité: Homme Femme

Date de naissance:

Numéro de téléphone:

Ville:

Vos préférences de voyage

Musique: Oui Non

Animaux: Oui Non

Discussion: Oui Non

Fumeur: Oui Non

Présentation:

Votre photo: Aucun fichier choisi

Je ne suis pas un robot

Figure 4. 9 Interface « S’inscrire »

Covoiturage

[Accueil](#)
[Mes Covoitages](#)
[Publier](#)
[Comment ça marche?](#)
🔔 1
Mouloud ▾

Tizi Ouzou Tizi Ouzou ➔ Alger Centre Alger

Départ	Tizi Ouzou
Arrivée	Alger Centre
Date de départ	📅 08/06/2015
Heure	🕒 09:30
Distance	102 km
Durée estimée	1 heure 31 min

Détails du voyage 100 DA

Véhicule	Clio 4
Bagages	Petite taille
Flexibilité Horaire	+/- 15 minutes
Détails	Rendez-vous à la gare multimodale de Tizi Ouzou à 9h15 maximum. ma voiture est grise, année 2013 PAS DE DETOURS

Publier le : 09/06/2015

Questions publiques (1)

Yassine Arrib il y a 21 heures
 Pouvez-vous me déposer à Bab Ezzouar ?

Poser vos questions au conducteur

Il reste 1 places

Profil du Conducteur

Lyna Bouizegarene

Femme
20 ans

Numéro de téléphone

📍 Ville : Tizi Ouzou

📢 Annonces publiées : 5

🕒 Dernière connexion : il y a 12 heures

★ Moyenne des avis : 4.0 sur 3 avis

Préférences de voyage :

Musique
Animaux
Discussion
Fumeur

Les passagers de ce covoiturage

Yassine A
 Homme
21 ans

Chanez L
 Femme
21 ans

Mouloud A
 Homme
22 ans

Avis sur Lyna

Extraordinaire!
 De Chanez L
 Très agréable!

Excellent!
 De Yassine A
 Conduit prudemment

Bien
 De Mouloud A
 Conduit lentement mais le trajet était sympathique

Figure 4. 10 Interface « Page détaillée »

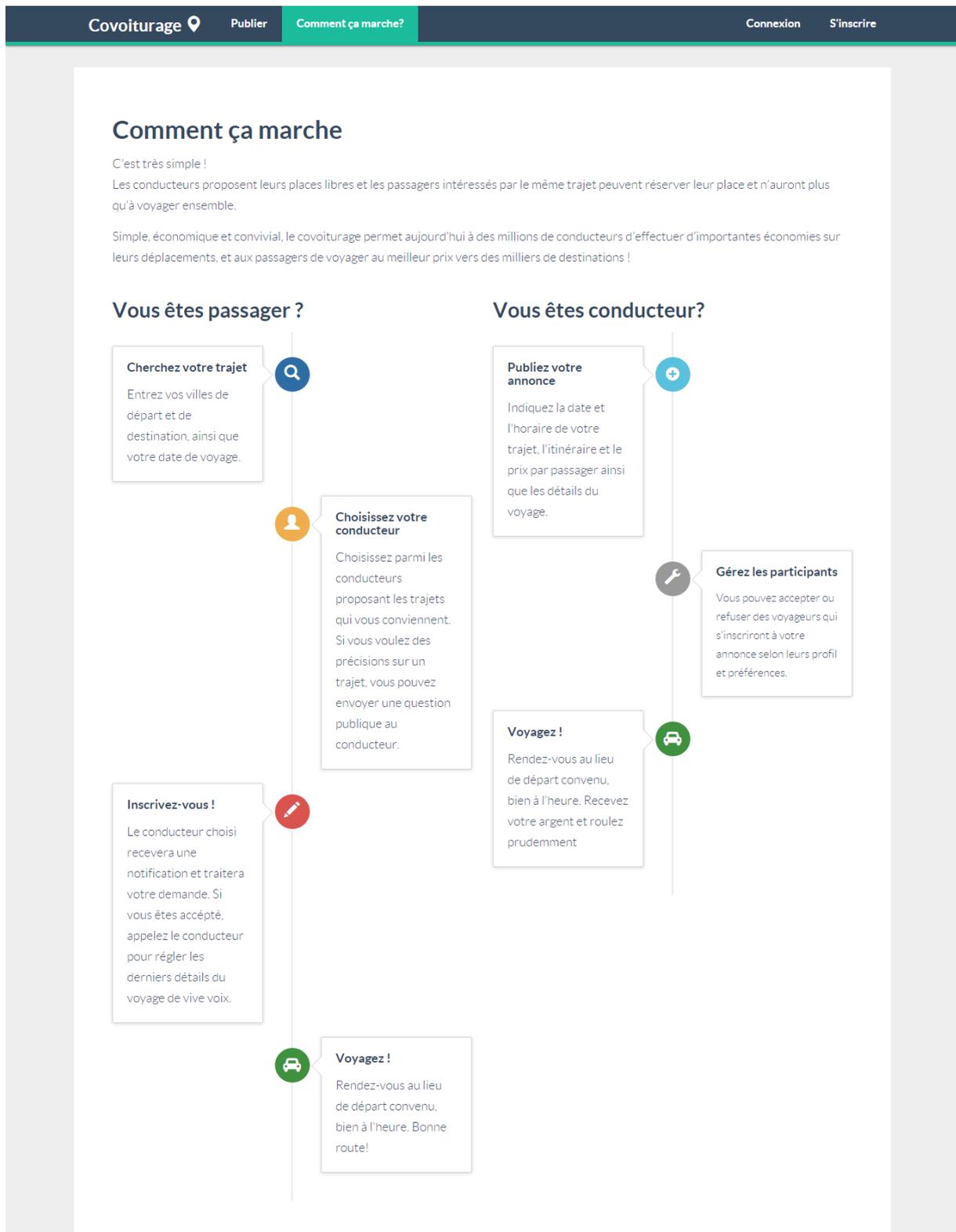


Figure 4. 11 Interface « Comment ça marche »

Figure 4. 12 Interface « Publier »

Figure 4. 13 Interface « Connexion »

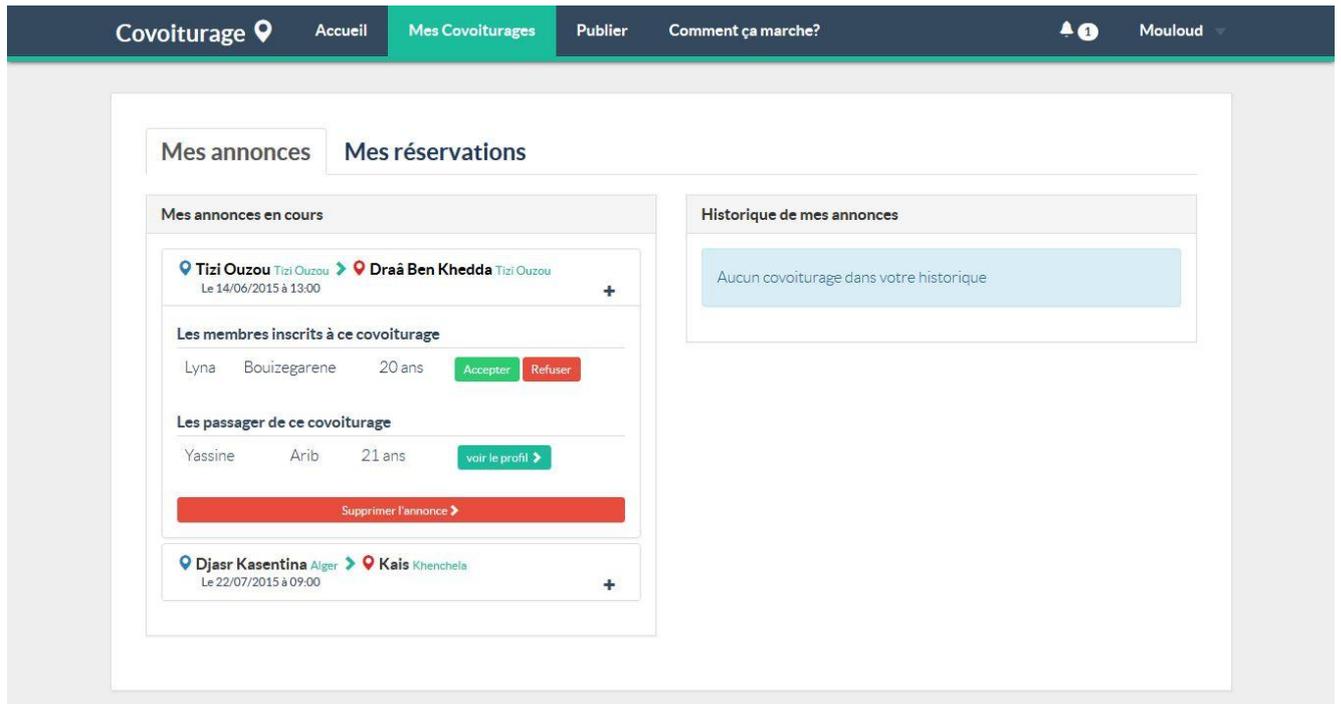


Figure 4. 14 Interface « Mes Covoiturages »

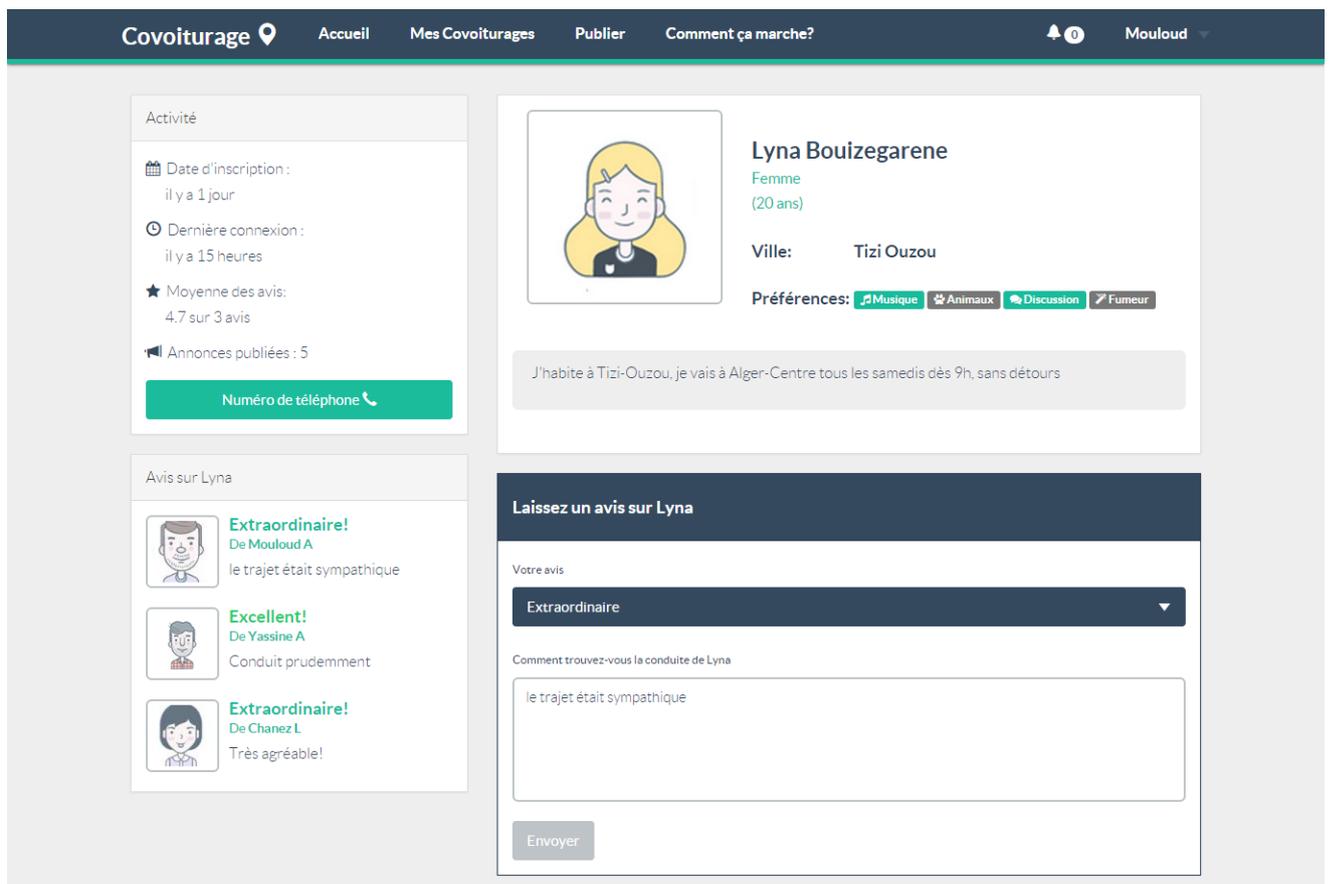


Figure 4. 15 Interface « Profil »

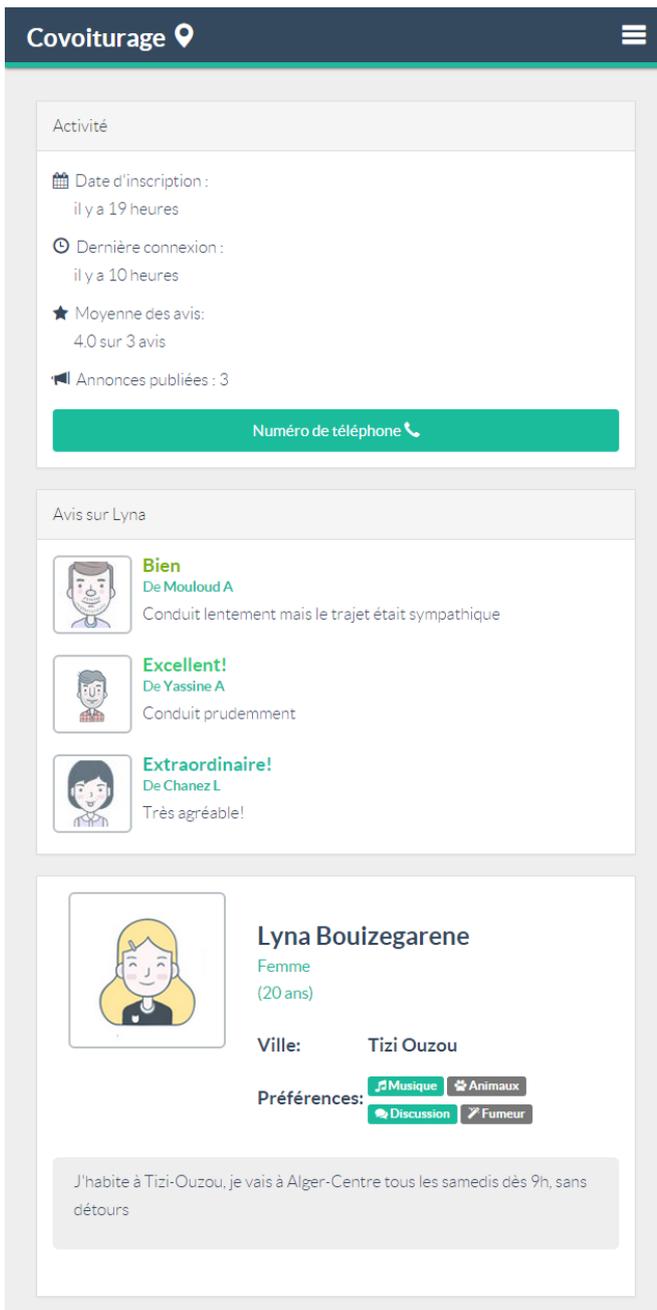


Figure 4. 17 Interface « Profil » en format mobile

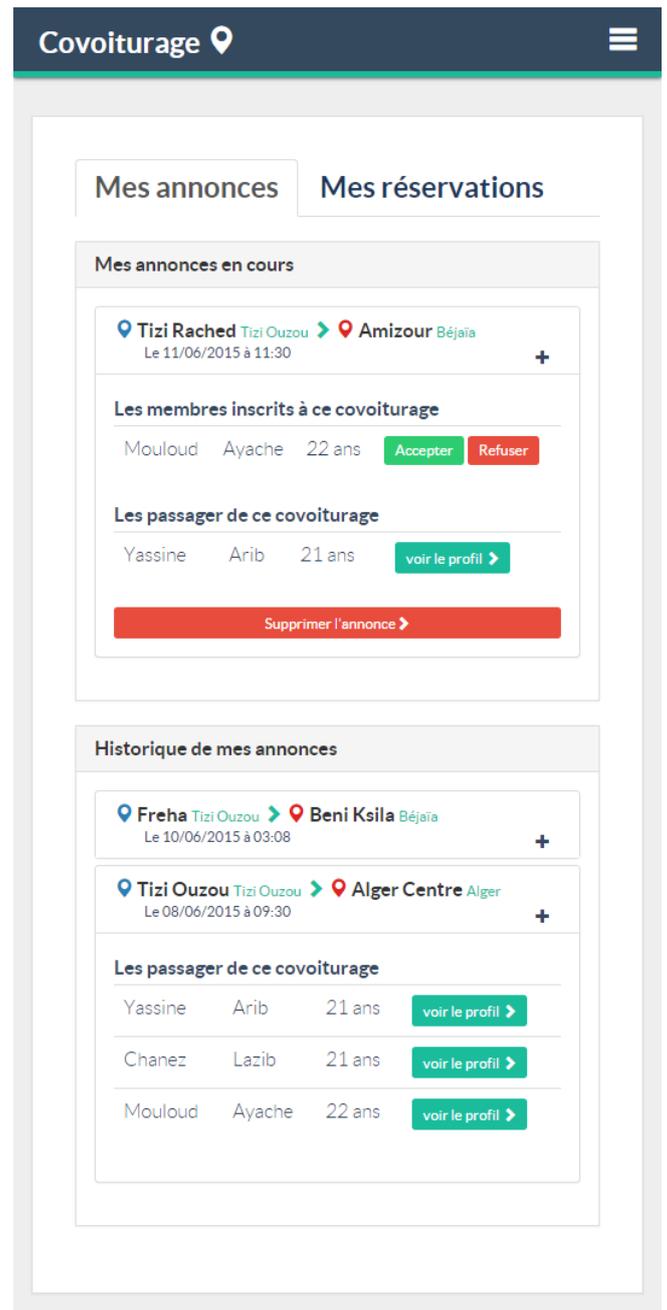


Figure 4. 16 Interface «Mes covoiturages» en format mobile

The screenshot shows the 'Chercher VOTRE COVOITURAGE' interface. At the top, there is a navigation bar with 'Covoiturage' and a location icon, 'Accueil', 'Mes Covoiturages', 'Publier', and 'Comment ça marche?'. A user profile 'Lyna' is visible in the top right. The main heading is 'Chercher VOTRE COVOITURAGE'. Below this is a search form with fields for 'Départ' (Indiquez un lieu), 'Arrivée' (Indiquez un lieu), and 'Date de départ'. The search results are displayed in a white box with a light blue header '4 résultats trouvés'. The results are as follows:

Driver	Age	Vehicle	Departure	Price
Lyna	20 ans	Clio 4	Départ dans 2 semaines	30 DA
Mouloud	22 ans	Clio	Départ dans 1 semaine	40 DA
Lyna	20 ans	206	Départ dans 1 semaine	50 DA
Mouloud	22 ans	206	Départ dans 1 semaine	60 DA

Figure 4. 18 Interface « Résultats de Recherche»

Conclusion générale

L'objectif de notre projet de fin d'étude était de concevoir et réaliser un site web qui permet de mettre en contact des personnes souhaitant avoir recours au covoiturage comme moyen de transport afin de réduire les frais du voyage et leurs empreintes écologique. Nous pensons avoir atteint les objectifs fixés au départ.

Cette étude est une expérience très enrichissante, elle nous a permis d'acquérir des connaissances multiples dans le domaine du développement web. Notamment :

- Des notions avancé en PHP et en développement orienté objet
- Séparation des classes en packages conformément au design pattern MVC
- L'utilisation de Framework PHP tel que Laravel
- L'utilisation de l'API JavaScript Google Maps
- La modélisation avec UML
- L'utilisation d'outils professionnel pour la gestion de projet comme Composer et Git
- La réalisation de sites adaptatifs pour le format mobile

Annexe

1. Définition de l'UML :

UML (Unified Modeling Language), que l'on peut traduire par « langage de modélisation unifié », est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, il est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

UML n'est pas une méthode, dans la mesure où elle ne représente aucune démarche. A ce titre, UML est un formalisme de modélisation objet. Le mot méthode parfois utilisé par abus de langage ne doit pas donc être entendu comme une démarche.

2. Un peu d'histoire :

Au début des années 90, une cinquantaine de méthodes objets ont vu le jour. Ce qui est un signe de l'intérêt du sujet, et également de confusion. Toutes ces méthodes utilisaient à peu près les mêmes concepts de classes, d'associations, de partition en sous-systèmes...

En octobre 1994, Grandy Booch et Jim Rumbaugh fondèrent la Rational Software Corporation pour unir leurs efforts en vue de créer une norme industrielle unique à partir de leurs méthodes. C'est ainsi que née la méthode unifiée (Unified Method) en octobre 1995. Ils sont ensuite rejoints par Ivar Jacobson, l'inventeur des cas d'utilisation (Uses Cases), et ont publiés UML 0.9 en juin 1996, puis UML 1.0 en janvier 1997.

La version 1.1, mise au point avec d'autres partenaires, devient un standard en novembre 1997 lorsqu'elle est acceptée par l'OMG (Object Management Group). La poursuite du développement d'UML fut ensuite intégralement remise aux mains de l'OMG. En juillet 1998, l'OMG publiait l'UML 1.2, puis UML 1.3 en juin 1999. En mai 2002, UML 1.4, qui présentait de petites améliorations et quelques extensions, voyait le jour. UML 1.5 a été publié en mars 2003 avec également quelques correctifs.

UML 2.0, publié en 2005, présente quant à lui de profondes modifications et de véritables extensions par rapport aux versions 1.x, comme par exemple le diagramme d'activité et le diagramme de séquence.

3. Présentation générale des diagrammes :

UML dans sa version 2 propose treize diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles :

3.1. Les diagrammes structurels ou statiques :

Ces diagrammes, au nombre de six, ont vocation à représenter l'aspect statique d'un système (classes, objets, composants...).

- Diagramme de classe.
- Diagramme d'objet.
- Diagramme de composant (modifié dans l'UML 2).
- Diagramme de déploiement (modifié dans l'UML 2).
- Diagramme de paquetage (nouveau dans l'UML 2).
- Diagramme de structure composite (modifié dans l'UML 2).

3.2. Les diagrammes de comportement :

Ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Sept diagrammes sont proposés par UML :

Diagramme des cas d'utilisation.

- Diagramme d'état-transition.
- Diagramme d'activités (modifié dans l'UML 2).
- Diagramme de séquence (modifié dans l'UML 2).
- Diagramme de communication (anciennement appelé collaboration).
- Diagramme global d'interaction (nouveau dans l'UML 2).

3.3. Les diagrammes utilisés pour la conception du site :

3.3.1. Diagramme de cas d'utilisation :

3.3.1.1 Définition :

Formalisés par Ivar Jacobson, un cas d'utilisation (en anglais use case) permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. Le

format de représentation d'un cas d'utilisation est complètement libre mais UML propose un formalisme et des concepts issus de bonnes pratiques.

Le diagramme de cas d'utilisation permet de représenter visuellement une séquence d'actions réalisées par un système, représenté par une boîte rectangulaire, produisant sur un acteur, appelé acteur principal, et ceci indépendamment de son fonctionnement interne

3.3.1.2 Eléments de modélisation des cas d'utilisation :

L'acteur : est un ensemble cohérent de rôles, joués par les utilisateurs en interaction avec le cas d'utilisation .Il peut être un humain, une machine ou tout autre système. Il est représenté par un bonhomme en fil de fer (en anglais stick man).

Acteur principal: regroupe les acteurs qui utilisent les fonctions principales du système. On le représente généralement à gauche (idéalement il y en a un seul).

Acteur secondaire: regroupe les acteurs qui effectuent les tâches administratives et de maintenances. On le représente généralement à droite.

Le cas d'utilisation : ou un use case est un ensemble de séquences d'actions qu'un système exécute pour produire un résultat tangible par un acteur. Chaque cas d'utilisation doit comporter un nom qui le distingue des autres cas d'utilisation.

3.3.1.3 Relations entre cas d'utilisation :

- « **includ** » : (**Inclusion**) indique que le cas d'utilisation source incorpore explicitement le comportement d'un autre cas d'utilisation à un emplacement précisé par la source.
- « **extend** » : (**Extension**) indique que le cas d'utilisation cible étend le comportement du cas d'utilisation source.
- **Généralisation**: les cas d'utilisation descendants héritent des propriétés de leur parent/ généralisation dans laquelle les objets de l'élément spécialisé (l'enfant) peuvent remplacer les objets de l'élément général (le parent).

3.3.2. Diagramme de classe :

Les diagrammes des classes représentent un ensemble de classes, d'interfaces, de collaboration, et des associations qui relient ses différentes classes. Ce sont les diagrammes les plus utilisés dans la modélisation des systèmes orientés objet. Ils présentent la vue de conception statique des systèmes. Les diagrammes de classes permettent de représenter l'ensemble des informations formalisées, qui sont regroupées dans des classes.

Le diagramme de classes comporte 3 concepts :

- Classe
- Attribut
- Opération
- Relation

Les notions utilisées par le diagramme de classes :

La notation de classe : une classe est une description abstraite (condensée) d'un ensemble d'objets du domaine de l'application : elle définit leur structure, leur comportement et leurs relations.

Représentation : les classes sont représentées par des rectangles compartimentés :

- ✓ le 1er compartiment représente le nom de la classe
- ✓ le 2ème compartiment représente les attributs de la classe
- ✓ le 3ème compartiment représente les opérations de la classe

Formalisme :

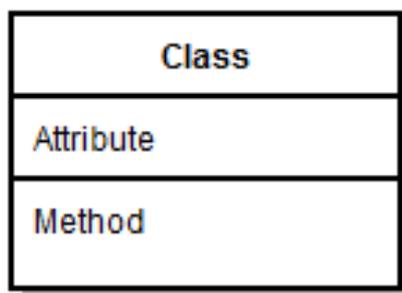


Figure A. 1 Représentation graphique d'une classe.

1. **la notation d'attribut** : Une classe correspond à un concept global d'information et se compose d'un ensemble d'informations élémentaires, appelées attributs de classe. Un attribut représente la modélisation d'une information élémentaire représentée par son nom et son format.
2. **la notation d'opération** : l'opération représente un élément de comportement des objets, défini de manière globale dans la classe. Une opération est une fonctionnalité assurée par une classe. La description des opérations peut préciser les paramètres d'entrée et de sortie ainsi que les actions élémentaires à exécuter.
3. **notation de relation** : s'il existe des liens entre objets, cela se traduit nécessairement par des relations qui existent entre leurs classes respectives. Les liens entre les objets doivent être considérés comme des instances de relations entre classes.

Il existe plusieurs types de relations entre classes :

- l'association
- la généralisation/spécialisation

Les relations entre les classes :

L'association : L'association est la relation la plus courante et la plus riche, elle relie plusieurs classes entre elles. L'association existe entre les classes et non entre les instances elle est introduite pour montrer une structure et non pour montrer des échanges de données.

la multiplicité : elle définit le nombre d'instances de l'association pour une instance de la classe. La multiplicité est définie par un nombre entier ou un intervalle de valeurs. La multiplicité est notée sur le rôle (elle est notée à l'envers de la notation MERISE).

1	Un et un seul
0,1	Zéro ou un
N,*	N (entier naturel)
M, N	De M à N (entiers naturels)
0,*	De zéros à plusieurs
1,*	De 1 à plusieurs

Tableau : Les différentes multiplicités d'une association.

Formalisme et exemple en employant le nom de l'association et la multiplicité

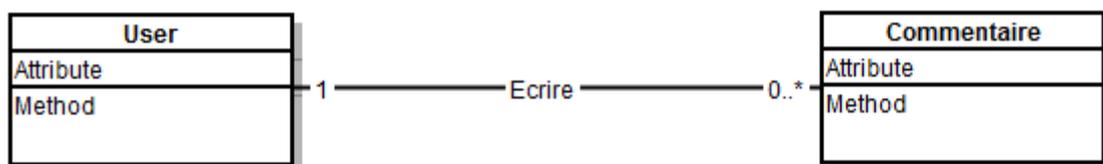


Figure A. 2 Relation d'association

La généralisation/spécialisation : Le principe de généralisation/spécialisation permet d'identifier parmi les objets d'une classe (générique) des sous-ensembles d'objets (des classes spécialisées) ayant des définitions spécifiques. La classe la plus spécifique (appelée aussi classe fille, classe dérivée, classe spécialisée, classe descendante ...) est cohérente avec la classe la plus générale (appelée aussi classe mère, classe générale ...), c'est-à-dire qu'elle contient par **héritage** tous les attributs, les membres, les relations de la classe générale, et peut contenir d'autres.

Les relations de généralisation peuvent être découvertes de 2 manières :

- la généralisation : il s'agit de prendre des classes existantes déjà mises en évidence et de créer de nouvelles classes qui regroupent leurs parties communes ; il faut aller du plus spécifique au plus général.

- La spécialisation : il s'agit de sélectionner des classes existantes (déjà identifiées) et d'en dériver des nouvelles classes plus spécialisées, en spécifiant simplement les différences.

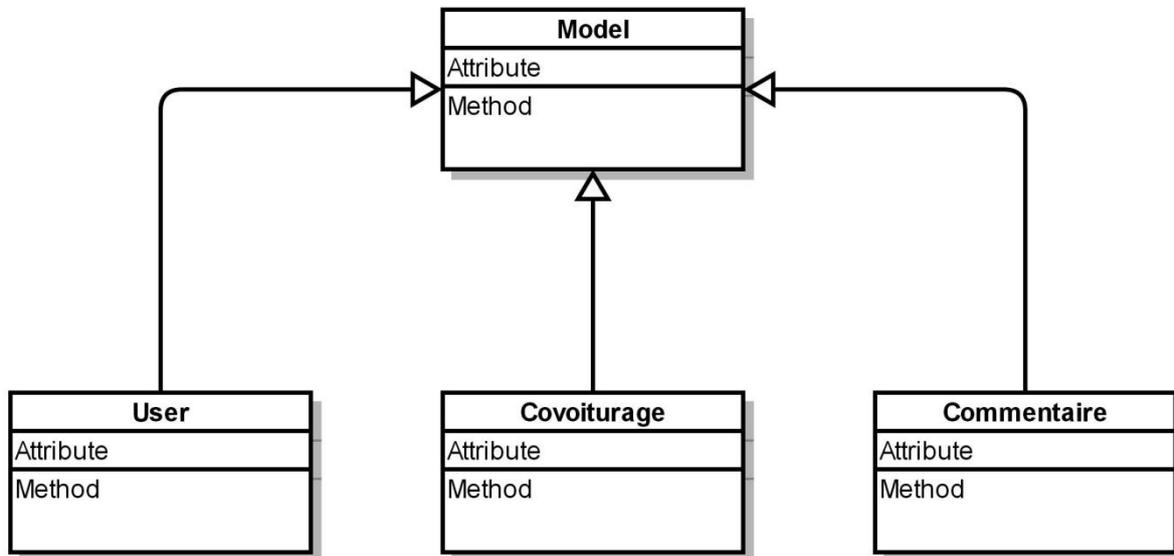


Figure A. 3 Relation de généralisation

3.3.3. Diagramme de séquence :

Le diagramme de séquence représente des échanges de messages entre objets. Il permet de représenter un processus de façon simplifiée, en se centrant sur les échanges entre acteurs ou avec le système d'information.

Le diagramme de séquence permet de visualiser les messages par une lecture de haut en bas. L'axe vertical représente le temps, l'axe horizontal représente les objets qui collaborent. Une ligne verticale en pointillé est attachée à chaque objet et représente sa durée de vie.

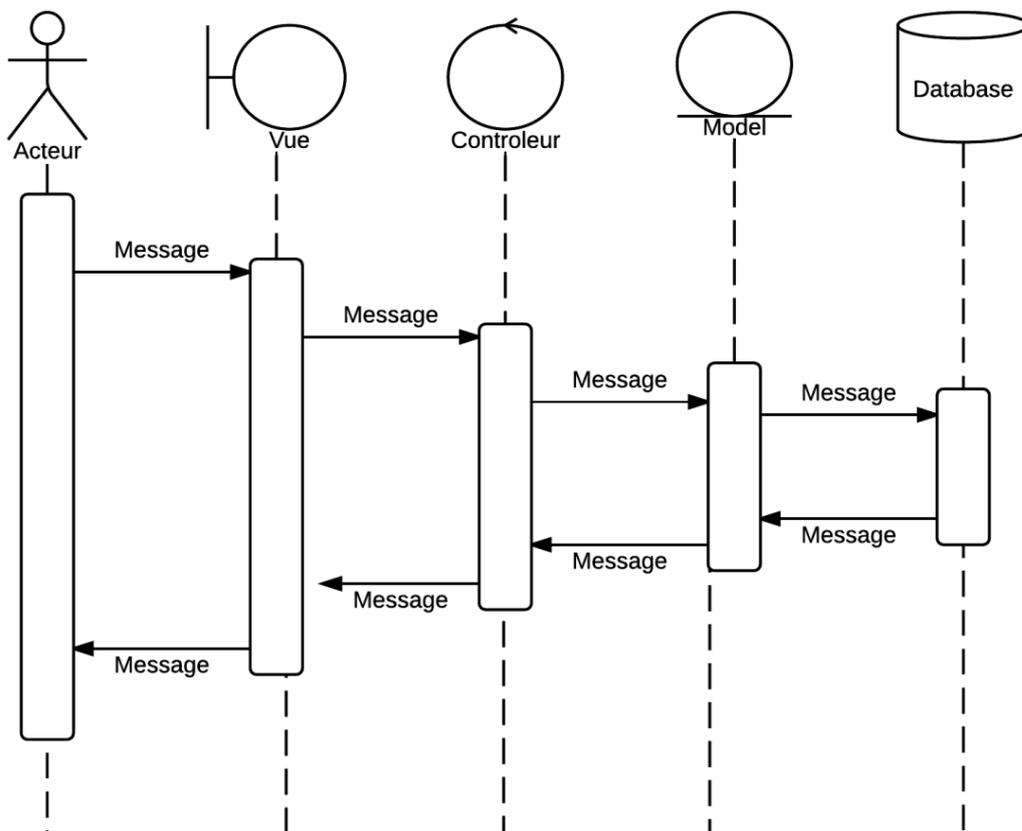


Figure A. 4 Exemple du diagramme de séquence

3.3.4. Diagramme de navigation :

Diagrammes dynamiques représentant de manière formelle l'ensemble des chemins possibles entre les principaux écrans proposés à l'utilisateur. Ces diagrammes s'appellent des diagrammes de navigation.

La navigation est particulièrement importante pour les sites web. Elle peut prendre la forme de menus, liens, hyperliens, boutons directionnels, etc. Une navigation fastidieuse n'incitera pas l'internaute à revenir sur le site web incriminé...

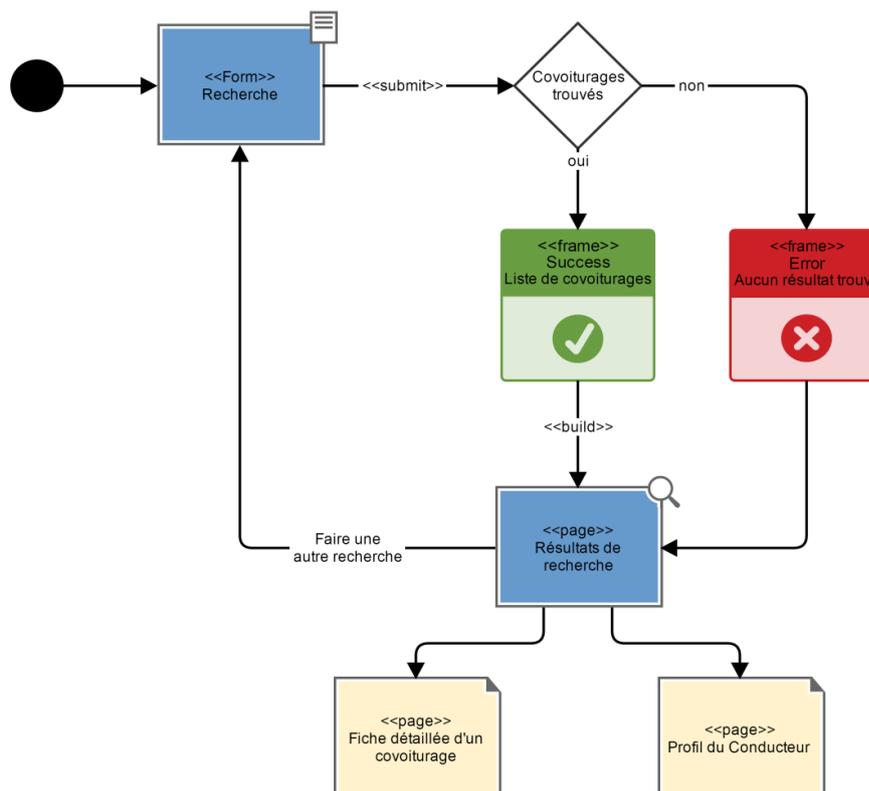


Figure A. 5 Exemple du diagramme de navigation

4. Extension d'UML pour le Web :

- **Définition** : L'extension d'UML pour le Web définit un ensemble de stéréotypes, d'étiquettes et de contraintes, qui rend possible la modélisation d'application Web. Le principal élément spécifique des applications Web étant la page Web, plusieurs stéréotypes, qui lui sont destinés
- **Un stéréotype** : est une extension du vocabulaire d'UML, il permet d'associer une nouvelle signification à un élément du modèle. On représente un stéréotype par une chaîne de caractères entre guillemets.
- **Stéréotypes utilisés** :
 - Une page complète du site « page »,
 - Un frame particulier à l'intérieur d'une page « frame »,
 - Un formulaire « Form »,
 - Un lien HyperText vers une page « Link »,
 - Une soumission de formulaire « Submit » ,
 - Une redirection vers une page « Redirect »

Bibliographie & Webographie

PHP et MySQL Maitrisez le développement d'un site web dynamique et interactif par Olivier HEURTEL

Les Cahiers du Programmeur UML : Modéliser une application Web

<http://fr.wikipedia.org>

<http://laravel.com/docs/5.1>

<http://www.phptherightway.com/>

<http://openclassrooms.com/courses/prenez-en-main-bootstrap>

<http://bpesquet.developpez.com/tutoriels/php/evoluer-architecture-mvc/>

<http://openclassrooms.com/courses/decouvrez-le-framework-php-laravel>

<http://openclassrooms.com/courses/evoluez-vers-une-architecture-php-professionnelle>

Résumé

Notre travail consiste à développer un site web pour les personnes souhaitant avoir recours au covoiturage comme moyen de transport en utilisant les technologies récentes du développement Web.

Le covoiturage est l'utilisation conjointe et organisée d'un véhicule, par un conducteur non professionnel et un ou plusieurs tiers passagers, dans le but d'effectuer tout ou une partie un trajet commun.

L'objectif de notre projet de fin d'étude était de concevoir et réaliser un site web qui permet de mettre en contact des personnes souhaitant avoir recours au covoiturage comme moyen de transport afin de réduire les frais du voyage et leurs empreintes écologique. Nous pensons avoir atteint les objectifs fixés au départ.

Cette étude est une expérience très enrichissante, elle nous a permis d'acquérir des connaissances multiples dans le domaine du développement web. Notamment :

- Des notions avancé en PHP et en développement orienté objet
- Séparation des classes en packages conformément au design pattern MVC
- L'utilisation de Framework PHP tel que Laravel
- L'utilisation de l'API JavaScript Google Maps
- La modélisation avec UML
- L'utilisation d'outils professionnel pour la gestion de projet comme Composer et Git
- La réalisation de sites adaptatifs pour le format mobile