

الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire de Mila

Institut des sciences et de la technologie

Département de Mathématiques et Informatique

Mémoire préparé En vue de l'obtention du diplôme de licence
en: - Filière informatique général

Thème

**Développement d'un site web dynamique
pour une agence de location de voiture**

Préparé par : BOUANANE Zina

SEDRATI Samira

Encadré par: KHALEFI Souheila

Année universitaire : 2013/2014

Dédicaces

Je dédie ce travail a més très chers parents avec tous l'amour que je leurs porte .

A mes frères : Mohamed el hadi, Walid, Hocine, youcef.

A mes sœurs :Wassila, Naima, Farah, Fadila.

A tous mes amis particulièrement , mon binome Samira, Widad, Ayda,houda, Radia, Sonia.

Ainsi qu'a tous ceux qui nous aides de prés ou de loin(surtous Mr :Khaniwi Mohamed el Amine)

ZINA

Je dédie mon travail à ma famille avec tous mes sentiments de respect, d'amour,

De gratitude et de reconnaissance.

A ma chère mère ;

A mon très cher père ;

A mes frères :Adel, Issam ,Rachid.

A mes sœurs :Aziza, Lamia.

A mes enseignants sans exception, pour leurs efforts afin m'assurer une belle base pour bien avancé.

A tous mes amis Zina, Zineb, ayda, farah.

A mes connaissances de proche ou de loin.

SAMIRA

REMERCIEMENTS

*En préambule à ce rapport, Nous tenons à remercier en premier lieu
Notre ALLAH qui nous a donné la force et le courage pour continuer
Et nous avoir aidé et éclairer le chemin pour réaliser de ce modeste travail.
Nous tenons d'abord à exprimer notre profonde gratitude à notre prometteur
M^{elle} KHALFI SOUHEILA de nous avoir confié ce travail, et dirigé dons son
Élaboration jusqu'au bout, et nous tenons à le remercier pour ses conseils, ses
encouragements et sa constante disponibilité.*

*Nous dressons nos remerciements les plus sincère à : Mr directeur de la scolarité du
centre universitaire de Mila.*

Un remerciement infini est adressé à :

*Mr Djezy Mohemed El cherif le gérant de l'agence de location de voiture Milev_Auto
Location De Voiture, qui nous ont apporté aide et soutient.*

*Enfin nous exprimons notre reconnaissance pour tous ceux qui nous ont aidées et
encouragées de prés ou de loin.*

Samira& Zina

Sommaire

Introduction générale.....	1
CHAPITRE I : TECHNOLOGIE WEB	
1.1 Introduction.....	2
1.2 Internet.....	2
1.2.1 Qu'est ce qu'internet.....	2
1.2.2 Histoire d'Internet.....	3
1.3 Le Web.....	3
1.3.1 World Wide Web (WWW).....	3
1.3.2 Site Web.....	4
1.3.2.1 Quelques définitions	4
1.3.2.2 Les types de site web.....	5
1.3.2.2.1 Les sites web statiques.....	5
1.3.2.2.2 Les sites web dynamiques.....	6
1.3.2.2.3 Quel type de site pour quelle utilisation ?.....	7
1.4 Conclusion.....	8
CHAPITRE II : UML	
2.1 Introduction.....	9
2.2 Présentation du langage de modélisation UML.....	9
2.2.1 Historique d'UML.....	9
2.2.2 Définition d'UML.....	9
2.2.3 Pourquoi UML ?.....	10
2.2.4 Présentation des vues et diagrammes d'UML.....	10
2.2.4.1 la vue statique	11
2.2.4.2 la vue dynamique	11
2.2.5 Description de quelques diagrammes UML.....	11
2.2.5.1 Le diagramme de cas d'utilisation.....	11
1 Définition	11
2 Éléments constitutifs des cas d'utilisation.....	12
2.2.5.2 Le diagramme de classe.....	13
1 Définition	13

2 Éléments constitutifs de diagramme de classe.....	14
2.2.5.3 Diagramme de séquence.....	16
1 Définition	16
2 Éléments constitutifs de diagramme de séquence.....	16
2.2.5.4 Diagramme d'activités	17
1 Définition	17
2 les éléments constitutifs.....	18
2.2.6 Avantages et inconvénients d'UML.....	18
2.2.6.1 Les points forts d'UML.....	18
2.2.6.2 Les points faibles d'UML.....	19
2.3 Démarche de modélisation.....	19
2.3.1 Identification des besoins et spécification des fonctionnalités.....	20
2.3.1.1 Identification et représentation des besoins: Diagramme de cas d'utilisation..	20
2.3.1.2 Spécification détaillée des besoins : diagrammes de séquence système....	20
2.3.1.3 Maquette de l'IHM de l'application (non couvert par UML).....	20
2.3.2 Phases d'analyse.....	21
2.3.2.1 Analyse du domaine (modèle du domaine).....	21
2.3.2.2 Diagramme de classes participantes.....	21
2.3.2.3 Diagrammes d'activités de navigation.....	22
2.3.3 Phase de conception.....	22
2.3.3.1 Diagrammes d'interaction.....	22
2.3.3.2 Diagramme de classes de conception.....	22
2.4 Conclusion.....	23
CHAPITRE III :ETUDE DE L'EXISTANT	
3.1 Introduction	24
3.2 Description de l'organisme d'accueil de l'agence	24
3.3 Les étapes suivies pour effectuer une location de véhicule	24
3.4 Etude des documents	25
3.4.1 Définitions.....	25
3.4.2 Notation.....	25
3.4.3 Légende des types	25
3.4.4 Fiche d'étude du document Contrat.....	26
3.5 Conclusion	28
CHAPITRE IV :ETUDE DE CAS	
4.1 Introduction.....	29

4.2 La phase d'identification des besoins.....	29
4.2.1 Diagramme de cas d'utilisation.....	29
4.2.1.1 Identification des acteurs et des besoins fonctionnels.....	29
1 Le client	29
2 L'administrateur	29
4.2.1.2 Identification des diagrammes des cas d'utilisation	30
1 Diagramme de cas d'utilisation pour le client.....	30
2 le diagramme de cas d'utilisation pour l'administrateur.....	31
4.2.1.3 Description textuelle des cas d'utilisation.....	31
1 Fiches descriptives des cas d'utilisation pour le client.....	31
2 Fiches descriptives des cas d'utilisation pour l'administrateur.....	33
4.2.2 Les diagrammes de séquence système.....	36
4.2.2.1 Pour le client.....	36
1 Consulter voitures.....	36
2 réserver voiture.....	37
4.2.2.2 Pour l'administrateur.....	38
1 S'authentifier.....	38
2 Valider-Annuler réservation.....	39
3 Ajouter voiture.....	40
4 Modifier voiture.....	41
5 Supprimer voiture.....	41
4.3 La phase d'analyse	42
4.3.1 Le modèle du domaine.....	42
4.3.2 Diagramme de classe participantes	43
4.3.2.1 Pour le client.....	43
1 Consulter voitures	43
2 réserver voiture.....	44
4.3.2.2 Pour l'administrateur.....	45
1 S'authentifier.....	45
2 Valider-Annuler réservation.....	45
3 Ajouter voiture.....	46
4 Modifier la voiture.....	47
5 Supprimer voiture.....	48
4.3.3 Diagrammes d'activités de navigation.....	49

4.3.3.1 Pour le client.....	49
1 consulter voitures	49
2 réserver.....	50
4.3.3.2 Pour l'administrateur.....	51
1 S'authentifier.....	51
2 Valider-Annuler réservation.....	52
3 Ajouter voiture.....	53
4 Modifier voiture.....	54
5 Supprimer la voiture	55
4.4 Phase de conception	56
4.4.1 Les diagrammes d'interaction.....	56
4.4.1.1 Pour le client	56
1 consulter voitures.....	56
2 réserver.....	57
4.4.1.2 Pour l'administrateur.....	58
1 S'authentifier.....	58
2 Valider-Annuler réservation.....	59
3 Ajouter voiture.....	60
4 Modifier voiture	61
5 Supprimer la voiture	62
4.5 Conclusion	62

CHAPITRE V : IMPLEMENTATION

5.1 Introduction.....	63
5.2 Les langages de programmation et les outils de développement utilisé.....	63
5.2.1 Les langages de programmations.....	63
5.2.1.1 HTML.....	63
5.2.1.2 Java Script.....	63
5.2.1.3 CSS.....	64
5.2.1.4 php.....	64
5.2.1.5 MYSQL	64
5.2.2 Les outils de développement.....	65
5.2.2.1 easy php.....	65

5.2.2.2 PhpMyAdmin	65
5.2.2.3 Macromédia dreamweaver.....	66
5.3 Le passage du modèle du domaine au modèle relationnelle.....	66
5.3.1 Les règles de passage du diagramme de classe au modèle relationnel.....	66
5.3.2 Structure de la base de données.....	66
5.3.3 Implémentation de la base de données.....	67
5.4 Les interfaces du site web.....	68
5.4.1 les interfaces du client.....	68
5.4.2 les interfaces de l'administrateur.....	71
5.5 Conclusion.....	74
CONCLUSION GENERALE.....	75
REFERENCES BIBLIOGRAPHIQUES	76

LISTES DES FIGURES

Figure 1.1 : interconnection entre les ordinateurs.	2
Figure 1.2 : Logo historique du World Wide Web par Robert Cailliau.....	4
Figure 1.3 : site statique.....	6
Figure 1.4 : site dynamique.	7
Figure 2.1 : historique d'UML.....	10
Figure 2.2: diagramme de cas d'utilisation.....	13
Figure 2.3 : représentation graphique d'une classe.....	14
Figure 2.4: diagramme de classe.....	16
Figure 2.5: de diagramme de séquence.....	17
Figure 2.6 : Représentation graphique d'un diagramme d'activité.....	18
Figure 2.7: Quelle méthode pour passer de l'expression des besoins au code de l'application ?.....	19
Figure 2.8 : chaine complète de la démarche de modélisation du besoin jusqu'au code.....	23
Figure 4.1 : le diagramme de cas d'utilisation «client».....	30
Figure 4.2 : Diagramme de cas d'utilisation «administrateur».....	31
Figure 4.3 : diagramme de séquence système «consulter voiture».....	36
Figure 4.4 : diagramme de séquence système «réserver voiture».....	37
Figure 4.5: diagramme de séquence système «s'authentifier».	38
Figure 4.6: diagramme de séquence système «valider-annuler réservation ».....	39
Figure 4.7 : diagramme de séquence système «ajouter voiture».....	40
Figure 4.8 : diagramme de séquence système «modifier voiture».....	41
Figure 4.9 : diagramme de séquence système «supprimer voiture».....	41
Figure 4.10 : le modèle du domaine.....	42
Figure 4.11 : diagramme de classe participante « consulter voitures ».....	43
Figure 4.12 : diagramme de classe participante « réserver voiture ».....	44
Figure 4.13 : diagramme de classe participante « s'authentifier».....	45
Figure 4.14 : diagramme de class participante «valider-Annuler réservation».....	45
Figure 4.1 5: diagramme de class participante « ajouter voiture».....	46
Figure 4.16 : diagramme de class participante « modifier voiture».....	47
Figure 4.1 7: diagramme de class participante « supprimer voiture».....	48
Figure 4.18 : diagramme d'activité « consulter voitures ».....	49

Figure 4.19 : diagramme d'activité « réserver voiture ».....	50
Figure 4.20 : diagramme d'activité « s'authentifier».....	51
Figure 4.21 : diagramme d'activité « valider-Annuler réservation».....	52
Figure 4.22 : diagramme d'activité « ajouter voiture».....	53
Figure 4.23 : diagramme d'activité « modifier voiture».....	54
Figure 4.24: diagramme d'activité «supprimer voiture».....	55
Figure 4.25 : diagramme d'interaction « consulter voitures ».....	56
Figure 4.26 : diagramme d'interaction « réserver voiture ».....	57
Figure 4.27 : diagramme d'interaction « s'authentifier ».....	58
Figure 4.28 : diagramme d'interaction « Valider-Annuler réservation ».....	59
Figure 4.29 : diagramme d'interaction « ajouter voiture ».....	60
Figure 4.30 : diagramme d'interaction «modifier voiture».....	61
Figure 4.31 : diagramme d'interaction « supprimer voiture ».....	62
Figure 5.1 : « l'interface de PHPMYADMIN ».....	65
Figure 5.2 : « table administrateur ».....	67
Figure 5.3 : « table client ».....	67
Figure 5.4 : « table voiture ».....	67
Figure 5.5 : « table réservation ».....	68
Figure 5.6: « interface d'accueil ».....	68
Figure 5.7 : « interface de contact ».....	69
Figure 5.8 : « interface de consulter voitures ».....	69
Figure 5.9 : « interface formulaire de réservation ».....	70
Figure 5.10 : « interface consulter reservations ».....	70
Figure 5.11 : « interface d'authentification ».....	71
Figure 5.12 : « interface gérer voitures ».....	71
Figure 5.13 : « interface formulaire d'ajout d'une voiture».....	72
Figure 5.14 : « interface formulaire de modification d'une voiture».....	73
Figure 5.15 : « interface valider-annuler reservation».....	74

LISTES DES TABLEAUX

Tableau 3.1 : fiche d'étude du document contrat.....	27
Tableau 4.1 : « les cas d'utilisations ».....	30
Tableau 4.2 : «consulter voitures».....	31
Tableau 4.3 : «réserver voiture».....	32
Tableau 4.4 : «s'authentifier».....	33
Tableau 4.5 : «valider-annuler réservation ».....	34
Tableau 4.6 : «ajouter voiture».....	34
Tableau 4.7 : «modifier voiture».....	35
Tableau 4.8 : «supprimer la voiture»	35

LES ACRONYMES

WWW :Wide World Web.

HTML : HyperText markup langage .

CMS : Content Management System.

UML :Unified Modeling Langages.

OMG : Objet Management Group.

UP : unified process.

XP : externe programming.

IHM :Interface Homme-Machine.

JS : java script.

CSS : Cascading Style Sheets.

PHP : Personnel Home Page.

Introduction générale

Avec l'augmentation de la technologie et développement de l'internet en Algérie la meilleure solution pour les sociétés algériennes est la mise en place des sites Web constituent des extensions de leurs société réelles. Grace à des sites en ligne, les sociétés pourront toucher un grand nombre de clients et rendre leurs services et elles s'étendent désormais au monde entier.

En Algérie, pour faire une réservation d'une voiture, les clients doivent se déplacer directement à une société afin de chercher une offre qui satisfait leurs besoins. Leurs déplacements peuvent être inutiles et même peuvent provoquer un gaspillage de temps. D'ailleurs, les services ne peuvent toucher que les clients qui sont dans leurs sphères géographiques, et n'ont aucun moyen pour mettre à disposition leurs annonces de vente et services, à l'exception des supports traditionnels tels que les journaux ou les petites affiches.

Dans cet ordre d'idée se situe notre projet qui consiste à concevoir et à développer un site web pour la location des voitures pour une agence algérienne. Ce site va proposer les offres aux clients, qui peuvent consulter le site, rechercher des offres et faire des réservations en ligne.

Ce mémoire est organisé en cinq chapitres :

- **Chapitre I :** sert à donner un aperçu sur quelques notions jugées essentiels sur Internet, les sites web statiques et dynamiques et ses fonctionnements.
- **Chapitre II :** est dédié à présenter les concepts de base du langage de modélisation UML et d'expliquer la démarche que nous avons adopté pour aboutir à notre site Web.
- **Chapitre III :** s'intéresse à présenter l'étude de l'existant qui décrit de façon détaillée les différentes activités dans une agence de location de voitures ainsi que les conditions concernant la réservation d'une véhicule.
- **Chapitre IV :** constitue l'essentiel de notre travail. En effet, il décrit de façon détaillée la modélisation UML de notre système en appliquant les phases du processus. Il contient tous les diagrammes UML élaborés pour parvenir au logiciel.
- **Chapitre V :** le développement de chaque logiciel a besoin des outils de développement et de langages de programmation. Ce chapitre est consacré donc à présenter ceux que nous avons utilisés et d'exposer également quelques interfaces de notre application.
- Ce mémoire s'achève par une conclusion générale en précisant un récapitulatif de tout ce que nous avons réalisé et les perspectives d'une future extension du système.

1.1 Introduction

S'il est un phénomène qui a marqué le monde de l'informatique, c'est bien celui d'Internet. Ce réseau mondial créé par l'armée américaine, puis utilisé par les chercheurs et autres scientifiques, a connu une croissance phénoménale auprès du grand public avec l'introduction du World Wide Web.

Cependant, il est nécessaire d'avoir un minimum de notion sur la technologie internet et le web, ce qui est l'objectif du présent chapitre.

1.2 Internet [1]

1.2.1 Qu'est ce qu'internet ?

Il existe plusieurs définitions dans la littérature. On cite quelques une :

- **Définition 1** : Internet est un réseau international d'ordinateurs qui communiquent entre eux grâce à des protocoles d'échanges de données standard. Cette communication en réseaux se fait indépendamment des types d'ordinateurs utilisés (Mac, PC, Unix ou autres). Internet est un outil de communication qui utilise les fils téléphoniques, les fibres optiques, les câbles intercontinentaux et les communications par satellite.

- **Définition 2** : Internet est un système d'interconnexion de machines qui constitue un réseau informatique mondial, utilisant un ensemble standardisé de protocoles de transfert de données. C'est un réseau de réseaux, composé de millions de réseaux aussi bien publics que privés, universitaires, commerciaux et gouvernementaux. Internet transporte un large spectre d'information et permet l'élaboration d'applications et de services variés comme le courrier électronique, la messagerie instantanée et le World Wide Web.

Internet ayant été popularisé par l'apparition du World Wide Web, les deux sont parfois confondus par le public non averti. Le World Wide Web n'est pourtant que l'une des applications d'Internet.



Figure 1.1 : interconnection entre les ordinateurs

Il faut savoir qu'Internet est un réseau composé d'ordinateurs. Ceux-ci peuvent être classés en deux catégories.

- **Les clients** : ce sont les ordinateurs des internautes. Chaque client représente un visiteur d'un site web.
- **Les serveurs** : ce sont des ordinateurs puissants qui stockent et délivrent des sites web aux internautes, c'est-à-dire aux clients. La plupart des internautes n'ont jamais vu un serveur de leur vie. Pourtant, les serveurs sont indispensables au bon fonctionnement du Web.

1.2.2 Histoire d'Internet

L'histoire d'Internet remonte au développement des premières télécommunications. L'idée d'un réseau informatique, permettant aux utilisateurs de différents ordinateurs de communiquer, se développa par de nombreuses étapes successives. La somme de tous ces développements conduisit au « réseau des réseaux » (*network of networks*) que nous connaissons aujourd'hui en tant qu'Internet. Il est le fruit à la fois de développements technologiques et du regroupement d'infrastructures réseau existantes et de systèmes de télécommunications.

Les premières versions mettant en place ces idées apparurent à la fin des années 1950. L'application pratique de ces concepts commença à la fin des années 1960. Dès les années 1980, les techniques que nous reconnaissons maintenant comme les fondements d'Internet moderne commencèrent à se répandre autour du globe. Dans les années 1990, sa popularisation passa par l'apparition du World Wide Web

L'infrastructure d'Internet se répandit autour du monde pour créer le large réseau mondial d'ordinateurs que nous connaissons aujourd'hui. Il se répandit au travers des pays occidentaux puis frappa à la porte des pays en voie de développement, créant ainsi un accès mondial à l'information et aux communications sans précédent ainsi qu'une fracture numérique. Internet contribua à modifier fondamentalement l'économie mondiale, y compris avec les retombées de la bulle Internet.

1.3 Le Web

1.3.1 World Wide Web (WWW) [1]

Le World Wide Web (WWW), littéralement la « toile (d'araignée) mondiale », communément appelé le Web, et parfois la Toile, est un système hypertexte public fonctionnant sur Internet.

Le Web permet de consulter, avec un navigateur, des pages accessibles sur des sites. L'image de la toile d'araignée vient des hyperliens qui lient les pages web entre elles.



Figure 1.2 : Logo historique du World Wide Web par Robert Cailliau.

1.3.2 Site Web

1.3.2.1 Quelques définitions

- **Un site Web** est un regroupement de pages web¹ sur un sujet, un thème, un commerce ou une organisation. Un site Web a aussi une page principale. C'est une page Web qui aide les lecteurs à naviguer sur le site pour trouver l'information voulue.

Un site Web doit aussi être structuré. Comment une page Web est-elle reliée à une autre? Y a-t-il un ou plusieurs chemins ou parcours que les lecteurs peuvent utiliser pour naviguer à travers le site? Par exemple. Au début et à la fin de chaque page, il y a plusieurs boutons de navigation pour passer à la page précédente ou suivante.

- Le client Web (navigateur) [1]

On appelle logiciel client un programme qui utilise le service offert par un serveur. Le client envoie une requête et reçoit la réponse. Un navigateur Web est un logiciel client HTTP conçu pour accéder aux ressources du Web.

Sa fonction de base est de permettre la consultation des documents HTML disponibles sur les serveurs HTTP. Le support de d'autres types de ressource et d'autres protocoles de communication dépend du navigateur considéré. Exemples de navigateurs internet : Firefox, Google chrome, Internet Explorer de Microsoft.

- Le serveur web [1]

Un serveur web est un ordinateur connecté à Internet et sur lequel sont hébergés des sites web, composés de pages HTML (le serveur web, également appelé serveur HTTP, peut également

Une **page Web** est un fichier qui contient du texte, des images et des liens à d'autres pages.

être composé d'un groupe d'ordinateurs). Le logiciel fédérateur, sur un serveur web, est le serveur HTTP (Apache, le plus fréquemment), auquel viennent s'adjoindre un interpréteur de langage dynamique (PHP dans la plupart des cas), un gestionnaire de base de données (tel que MySQL) et d'autres programmes, comme un serveur SMTP.

La fonction d'un serveur web est de répondre aux requêtes des navigateurs Internet (Internet Explorer, Firefox, Chrome, Opéra, Safari...)

- **La requête :** C'est un message envoyé par le client au serveur décrivant l'opération à exécuter.
- **La réponse :** C'est le message envoyé par le serveur à un client (l'exécution d'une requête).

1.3.2.2 Les types de site web [2]

Il existe deux types de site web : les sites statiques et les sites dynamiques.

1.3.2.2.1 Les sites web statiques

a. Définition

Ce sont des sites réalisés uniquement à l'aide des langages (X) HTML et CSS. Ils fonctionnent très bien mais leur contenu ne peut pas être mis à jour automatiquement: il faut que le propriétaire du site (le webmaster) modifie le code source pour y ajouter des nouveautés. Ce n'est pas très pratique quand on doit mettre à jour son site plusieurs fois dans la même journée! Les sites statiques sont donc bien adaptés pour réaliser des sites "vitrine", pour présenter par exemple son entreprise, mais sans aller plus loin.

Ce type de site se fait de plus en plus rare aujourd'hui, car dès que l'on rajoute un élément d'interaction (comme un formulaire de contact), on ne parle plus de site statique mais de site dynamique. Lorsque le site est statique, le schéma est très simple. Cela se passe en deux temps :

- Le client demande au serveur une page Web.
- Le serveur lui répond en lui envoyant la page réclamée.

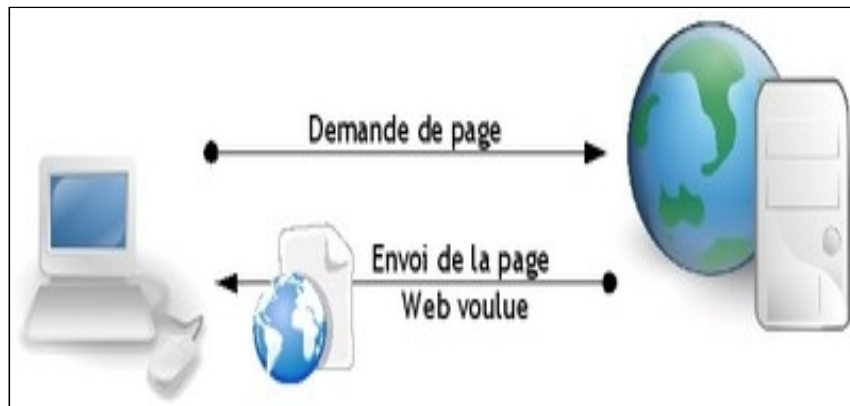


Figure 1.3 : site statique

b. Quels avantages pour un site statique ?

On vient de voir qu'un site statique possède beaucoup d'inconvénient : il faut s'y connaître en HTML pour le modifier et l'étape de mise à jour est fastidieuse. (Pour résumer) Mais il faut aussi reconnaître au site statique des avantages dans plusieurs domaines :

- le site internet est mis à jour en local sur la machine de l'administrateur : il n'y a donc pas de surprise une fois que le site est en ligne.
- le site internet ne fait pas appel aux technologies en perpétuelles évolutions qui permettent la mise en place de sites dynamique (PHP, Ruby, Python, Perl, Java, ASP, etc.) : on gagne donc en sécurité et en veille technologique.
- le site internet statique consomme peu de ressource serveur : le site n'utilisant aucune technologie compliquée (au hasard : PHP + MySQL + Apache), les coûts d'entretien et de maintenance en activité sont très inférieurs à ceux d'un site dynamique.
- le site internet statique se sauvegarde plus facilement : ceux qui ont déjà manipulé les bases de données MySQL utilisées pour la création de sites dynamiques savent que c'est une galère à sauvegarder et à restaurer. Le fait de disposer directement des pages HTML du site facilite la sauvegarde (un simple copier / coller sur une clé USB est c'est bon !).

1.3.2.2.2 Les sites web dynamiques

a. Définition

Plus complexes, ils utilisent d'autres langages en plus de HTML et CSS, tels que PHP et MySQL. Le contenu de ces sites web est dit « dynamique » parce qu'il peut changer sans l'intervention du webmaster ! La plupart des sites web que vous visitez aujourd'hui, y compris

le Site du Zéro, sont des sites dynamiques. Le seul pré requis pour apprendre à créer ce type de sites est de déjà savoir réaliser des sites statiques en HTML et CSS.

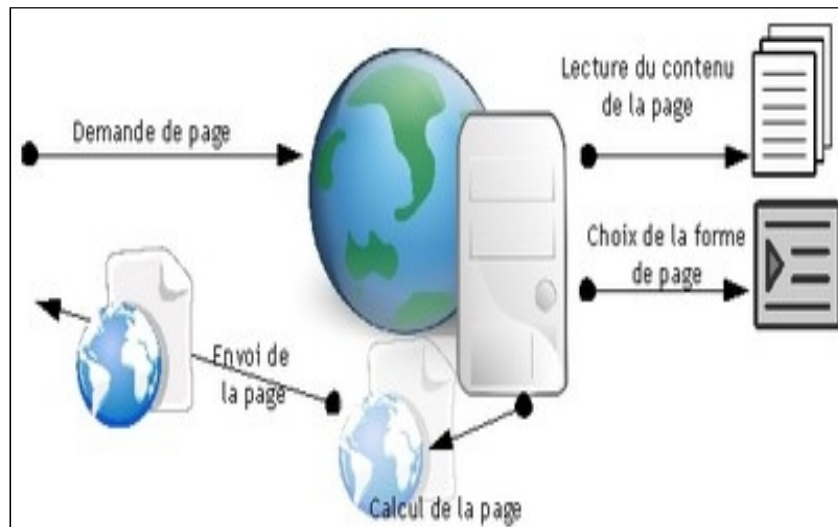


Figure 1.4 : site dynamique

b. Quels avantages pour un site dynamique ?

- De nombreux scripts gratuits existent déjà et permettent de réaliser tous les sites qu'on souhaite. Ainsi en téléchargeant le script (ou CMS : Content Management System) qui va bien, il sera très simple de créer un forum, un blog ou tout autre site.
- La mise à jour est très simple : une fois le script dynamique en place, on met à jour le site en ligne dans la partie « administration » du site. On peut donc mettre à jour le site de n'importe quel ordinateur et même depuis certains téléphones mobiles (avec accès Internet naturellement).
- Avec un site dynamique il est possible de réaliser une grande interaction avec les visiteurs : les visiteurs peuvent donc rester beaucoup plus longtemps sur vos pages si les fonctionnalités sont intéressantes.

1.3.2.2.3 Quel type de site pour quelle utilisation ?

On utilisera un site web statique pour une utilisation bien particulière. On utilisera ce fonctionnement pour un site web nécessitant peu de maintenance, peu de mise à jour et contenant peu de pages. En effet, comme la mise à jour d'un site Internet statique peut être fastidieuse, on utilisera ce type de site uniquement si les mises à jour sont exceptionnelles. Car, à chaque mise à jour il faudra modifier la page HTML du site et la mettre en ligne en effectuant une copie par FTP.

On utilisera plutôt un site dynamique si on souhaite créer une interaction avec ses lecteurs. Le site dynamique permet de se connecter en ligne sur son site pour réaliser sa mise à jour en direct. Une fois la mise à jour du site dynamique effectuée, le résultat apparaît directement aux lecteurs. On privilégiera donc les sites web dynamiques pour les sites permettant aux visiteurs de laisser des commentaires (blogs) ou de converser avec d'autres lecteurs (forums). De même si le site doit être mis à jour très fréquemment (plusieurs fois par semaine) on pourra choisir de créer un site dynamique même si aucune interaction n'est prévue avec les visiteurs.

Par exemple pour réaliser un site vitrine présentant les 10 produits vendus par une entreprise on pourra choisir :

- ❖ **Un site statique** si les produits ne sont modifiés qu'une ou deux fois par mois et qu'un ou deux nouveaux produits sont ajoutés au catalogue chaque année.
- ❖ **Un site dynamique** si on souhaite que les visiteurs ajoutent des commentaires sur les fiches produits et/ou qu'il faut modifier très souvent (ajout, suppression, modification) les fiches produits.

1.4 Conclusion

Nous détaillons dans le prochain chapitre les concepts de base du langage de modélisation UML, avec la démarche de modélisation choisie pour le développement du site web.

2.1 Introduction

Ce chapitre présente en premier lieu les concepts de base du langage de modélisation UML (Unified Modeling Languages). Nous commençons par une description d'UML, en suite, nous citons quelques caractéristiques qui le rendent le langage de modélisation le plus répandu. Ainsi, nous énumérons ses diagrammes avec une description de ceux les plus utilisés dans la modélisation. Cette étude permette une bonne lecture de ce qui va suivre et exprime de manière uniforme l'analyse, la conception et la réalisation d'une application informatique. Finalement, nous allons présenter une démarche de modélisation simplifiée et adaptée au développement de sites web dynamiques.

2.2 Présentation du langage de modélisation UML

2.2.1 Historique d'UML [3]

Au début des années 90, une cinquantaine de méthodes objet 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'association, de partition en sous-systèmes.

En octobre 1994, Grady 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 qu'est née la méthode unifiée (Unified Method) en octobre 1995. Ils sont ensuite rejoints par Evar Jacobson. L'inventeur des cas d'utilisation (Uses cases), et ont publié 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'est acceptée par l'OMG (Objet Management Group). La poursuite du développement d'UML fut ensuite intégralement remise aux mains de l'OMG. En juillet 1998, l'OMG publiait UML 1.2, puis UML 1.3 en juin 1999. En mai 2002 UML 1.4, qui présentait de petites améliorations et quelque extension, voyait le jour UML 1.5 a été publiée en mars 2003 avec également quelques correctifs, Version 2.0 en septembre 2004, Version 2.4.1 en août 2011.

2.2.2 Définition d'UML

Uml (Unified Modeling language) traduit « langage de modélisation objet unifié », il s'agit d'un langage graphique de modélisation objet permettant de spécifier, de construire et de visualiser les détails d'un système logiciel. L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions.

Historique d'UML

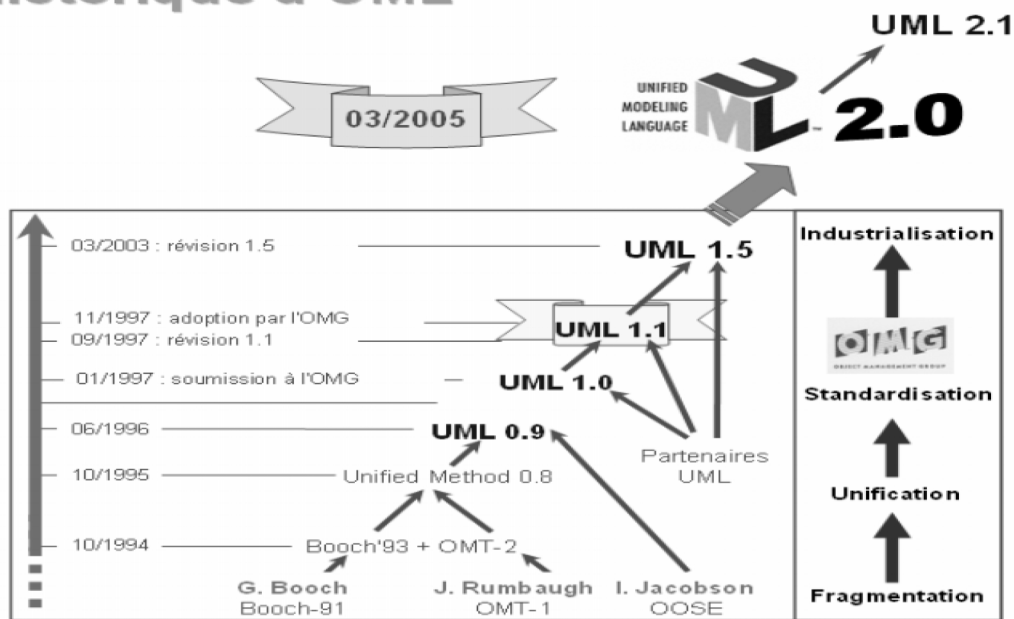


Figure 2.1 : historique d'UML

2.2.3 Pourquoi UML ? [4]

De la même façon qu'il vaut mieux dessiner une maison avant de la construire, il vaut mieux modéliser un système avant de le réaliser.

UML pour :

- Obtenir une modélisation de très haut niveau indépendamment de tout langage de programmation.
- Faire collaborer des participants de tous horizons autour d'un même document de synthèse.
- Faire des simulations avant de construire un système.
- Exprimer dans un seul modèle tous les aspects statiques, dynamiques, juridiques, spécifications, etc.
- Documenter un projet.
- Générer automatiquement la partie logicielle d'un système.

2.2.4 Présentation des vues et diagrammes d'UML [5]

Le méta-modèle UML fournit une panoplie d'outils permettant de représenter l'ensemble des éléments du monde objet (classes, objets, ...) ainsi que les liens qui les relie. Toutefois, Étant donné qu'une seule représentation est trop subjective, UML fournit un moyen astucieux

permettant de représenter diverses projections d'une même représentation grâce aux **vues**. Une vue est constituée d'un ou plusieurs diagrammes.

2.2.4.1 la vue statique

Une vue d'un système qui met l'accent sur la structure des objets, avec leur classificateurs, leurs relations, leurs attributs et leurs opérations. Cette vue comporte les diagrammes suivants :

1. Diagramme de classe
2. Diagramme d'objet
3. Diagramme de composants
4. Diagramme de déploiement
5. Diagramme de paquetage
6. Diagramme de structure composite

2.2.4.2 la vue dynamique

Met l'accent sur le comportement dynamique du système en montrant la collaboration entre les objets et les modifications apportées à l'état interne des objets, cette vue inclut les diagrammes suivants :

1. Diagramme d'activité ;
2. Diagramme des cas d'utilisation ;
3. Diagramme de communication ;
4. Diagramme d'état-transition ;
5. Diagramme globale d'interaction ;
6. Diagramme de séquence ;
7. Diagramme de temps

2.2.5 Description de quelques diagrammes UML

2.2.5.1 Le diagramme de cas d'utilisation

1 Définition : Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. Le diagramme de cas d'utilisation décrit la succession des opérations réalisées par un acteur (personne qui assure l'exécution

d'une activité). C'est le diagramme principal du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.

2 Éléments constitutifs des cas d'utilisation

a- Acteur : entité externe qui agit sur le système. Le terme acteur ne désigne pas seulement les utilisateurs humains mais également les autres systèmes. Les acteurs sont des classificateurs qui représentent des rôles au travers d'une certaine utilisation (cas) et non pas des personnes physiques. Ce sont des acteurs types.

b- Cas d'utilisation : ensemble d'actions réalisées par le système en réponse à une action d'un acteur.

c- Relations dans les diagrammes de cas d'utilisation : Après avoir identifié les acteurs et les cas d'utilisation, il est utile de restructurer l'ensemble des cas d'utilisation que l'on a fait apparaître afin de rechercher les comportements partagés, les cas particuliers et les généralisations.

- **Les relations entre les acteurs :** La seule relation entre les acteurs est la relation de généralisation.
- **Les relations possibles entre cas d'utilisation :** UML définit trois types de relations standardisées entre cas d'utilisation, détaillées ci-après :
 - ❖ **La relation d'inclusion :** Lors de la description des cas d'utilisation, il apparaît qu'il existe des sous-ensembles communs à plusieurs cas d'utilisation, il convient donc de factoriser ces fonctionnalités en créant de nouveaux cas d'utilisation qui seront utilisés par les cas d'utilisation qui les avaient en commun. Un cas d'utilisation A utilise un cas d'utilisation B signifie que une instance de A va engendrer une instance de B et l'exécuter, A dépend de B, B n'existe pas tout seul et A n'existe pas sans B.
 - ❖ **La relation d'extension :** Les extensions (Extend) représentent des prolongements logiques de certaines tâches sous certaines conditions. Autrement dit un cas d'utilisation A étend un cas d'utilisation B lorsque le cas d'utilisation A peut être appelé au cours de l'exécution du cas d'utilisation B. Elle est représentée par une flèche en pointillée avec le terme Extend. Ce type de relation peut être utile pour traiter des cas particuliers ou préciser les objectifs, ou pour tenir compte de nouvelles exigences au cours de la maintenance du système et de son évolution.
 - ❖ **Relation de généralisation entre cas d'utilisation :** La troisième relation est la relation de généralisation ou spécialisation. Le cas d'utilisation A est une

généralisation de B, si B est un cas particulier de A c'est-à-dire lorsque A peut être substitué par B pour un cas précis. Ces relations sont des traits pleins terminés par une flèche en triangle.

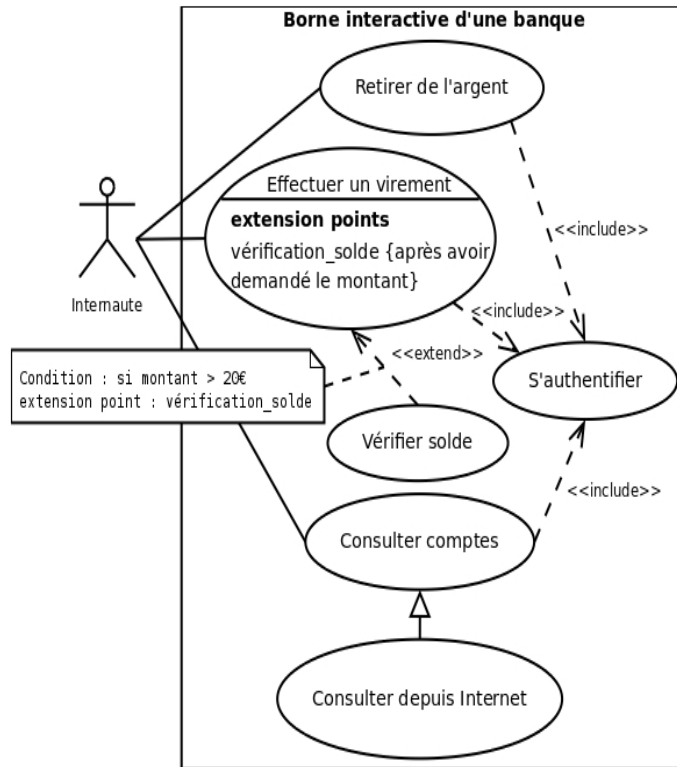


Figure 2.2 : diagramme de cas d'utilisation

d- La description textuelle des cas d'utilisation [5] : Le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation. Bien que de nombreux diagrammes d'UML permettent de décrire un cas, il est recommandé de rédiger une description textuelle car c'est une forme souple qui convient dans bien des situations.

2.2.5.2 Le diagramme de classe

1 définition : Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes. Une classe permet de décrire un ensemble d'objets (attributs et comportement), tandis qu'une relation ou association permet de faire apparaître des liens entre ces objets. On peut donc dire :

- un objet est une instance de classe.

- un lien est une instance de relation.

Le diagramme de classe est un modèle permettant de décrire de manière abstraite et générale les liens entre objets. Il comporte : les classes, les attributs, les identifiants, les relations les opérations ...etc

UML définit trois niveaux de visibilité pour les attributs et les opérations :

- 1-Public qui rend l'élément visible à tous les clients de la classe.
- 2-Protégé qui rend l'élément visible aux sous classes de la classe.
- 3-Privé qui rend l'élément visible à la classe seule

Nom de la classe
- variable privée + variable publique # variable protégée
- opération privée + opération publique # opération protégée

Figure 2.3 : représentation graphique d'une classe

2- Éléments constitutifs de diagramme de classe

a- Nom d'une classe : Le nom de la classe doit évoquer le concept décrit par la classe. Il commence par une majuscule. On peut ajouter des informations subsidiaires comme le nom de l'auteur de la modélisation, la date, etc. Pour indiquer qu'une classe est abstraite, il faut ajouter le mot-clef **abstract**.

b- Attributs de la classe : Les attributs définissent des informations qu'une classe ou un objet doivent connaître. Ils représentent les données encapsulées dans les objets de cette classe. Chacune de ces informations est définie par un nom, un type de données, une visibilité et peut être initialisé. Le nom de l'attribut doit être unique dans la classe.

c- Méthode de la classe : Dans une classe, une opération (même nom et même types de paramètres) doit être unique. Quand le nom d'une opération apparaît plusieurs fois avec des paramètres différents, on dit que l'opération est surchargée. En revanche, il est impossible que deux opérations ne se distinguent que par leur valeur retournée.

d- Relations entre classes : sont les suivantes :

- **Les associations :** Une association représente une relation structurelle entre classes d'objets. La plupart des associations sont binaires, c'est à dire qu'elles connectent deux classes. On représente une association en traçant une ligne entre les classes associées.
- **Les classes associations :** Il peut arriver que l'on ait besoin de garder des informations (attributs ou opérations) Propres à une association. Une classe de ce type est appelée classe association. Une classe association est une classe comme une autre qui peut entretenir des relations avec d'autres classes.
- **Agrégation :** Une agrégation est un type particulier d'association. Elle traduit la volonté de renforcer la dépendance entre les classes. C'est une association non symétrique dans laquelle une des extrémités joue un rôle prédominant par rapport à l'autre extrémité. Les critères suivants impliquent une agrégation :
 - une classe fait partie d'une autre classe.
 - une action sur une classe implique une action sur une autre classe.
 - les objets d'une classe sont subordonnés aux objets d'une autre classe.
- **La composition :** La composition est un cas particulier d'agrégation dans laquelle la vie des composants est liée à celle de l'agrégat. Dans la composition, l'agrégat ne peut être multiple. La composition se représente par un losange noir. Une composition est une association contraignante : la suppression d'un objet agrégat entraîne la suppression des objets agrégés.
- **Généralisation :** UML emploie le terme de généralisation pour désigner la relation de classification entre un élément plus général et un élément plus spécifique. La relation de généralisation signifie «est un » ou « est une sorte de ».

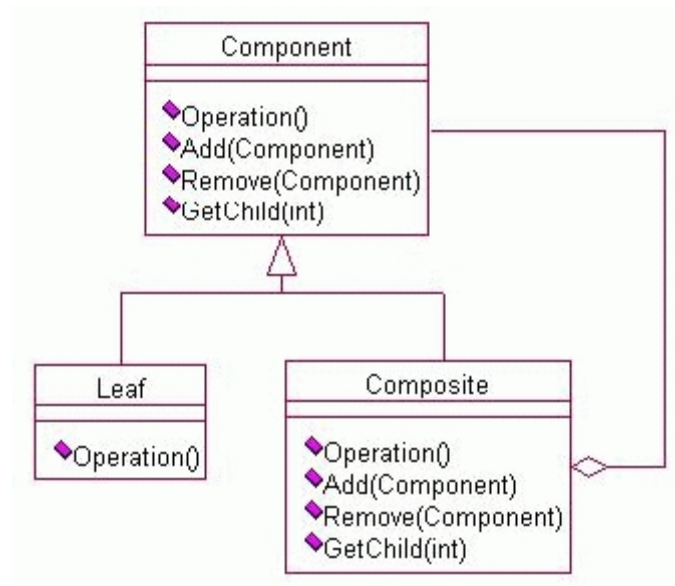


Figure 2.4 : diagramme de classe

2.2.5.3 Diagramme de séquence

1 Définition : Le diagramme de séquences permet de cacher les interactions d'objets dans le cadre d'un - scénario d'un diagramme de cas d'utilisation Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

2 Éléments constitutifs de diagramme de séquence

a- Ligne de vie : Une ligne de vie représente l'ensemble des opérations exécutées par un objet. Un message reçu par un objet déclenche l'exécution d'une opération. Le retour d'information Peut être implicite (cas général) ou explicite à l'aide d'un message de retour.

b- Message synchrone et asynchrone : Dans un diagramme de séquence, deux types de messages peuvent être distingués :

- **Message synchrones :** l'émetteur reste bloqué le temps que le récepteur traite le message envoyé et envoie la réponse. Graphiquement, un message synchrone se représente par une flèche en traits pleins et à l'extrémité pleine, suivi d'une réponse qui se représente par une flèche en discontinue.
- **Message asynchrones :** L'émetteur n'est pas bloqué lorsque le récepteur traite le message envoyé. Graphiquement, un message asynchrone se représente par une flèche en traits pleins et à l'extrémité ouverte.

c- Message de création et destruction d'instance :

- La création d'un objet est matérialisée par une flèche qui pointe sur le sommet d'une ligne de vie.
- La destruction d'un objet est matérialisée par une croix qui marque la fin de la ligne de vie de l'objet. La destruction d'un objet n'est pas nécessairement consécutive à la réception d'un message.

d- Les opérateurs : Les principaux opérateurs de diagramme de séquences sont:

- **Loop (boucle):** Le fragment peut s'exécuter plusieurs fois, et la condition de garde explicite l'itération.
- **Opt (optionnel) :** Le fragment ne s'exécute que si la condition fournie est vraie.
- **Alt (fragments alternatifs):** Seul le fragment possédant la condition vraie s'exécutera.

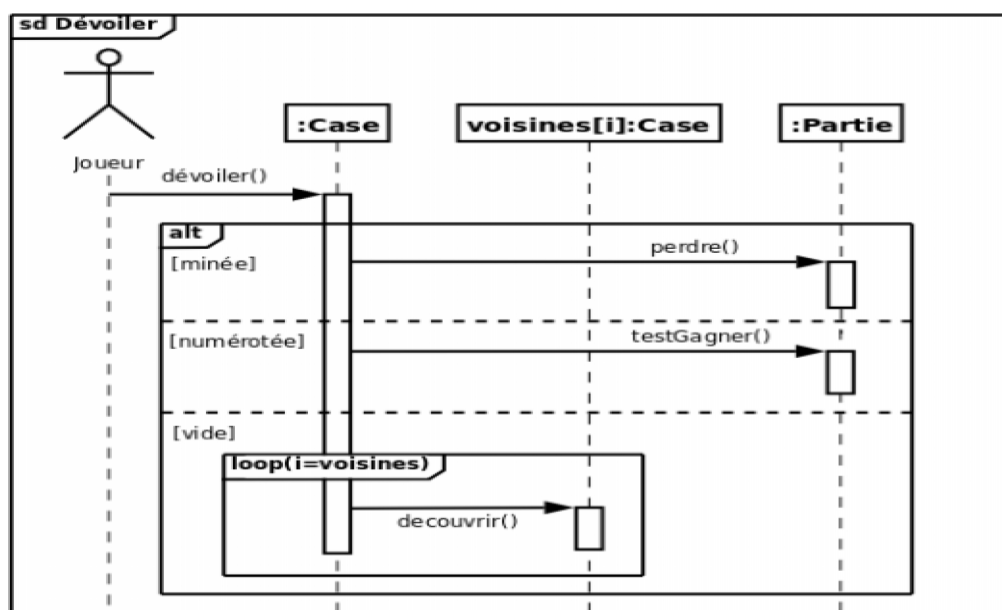


Figure 2.5 : de diagramme de séquence

2.2.5.4 Diagramme d'activités

1 Définition : Le diagramme d'activité n'est autre que la transcription dans UML de la représentation du processus telle qu'elle a été élaborée lors du travail qui a préparé la modélisation. Il montre l'enchaînement des activités qui concourent au processus.

2 les éléments constitutifs

a- Activité : Une activité UML représentée par un rectangle aux coins arrondis et contient la description textuelle des actions de base qu'elle réalise, ou simplement son nom si le niveau de spécification n'est pas encore assez précis pour détailler les actions, aucun syntaxe spécifique n'est proposée dans la norme pour l'expression de ces actions : on utilise le plus souvent une syntaxe proche d'un langage de programmation ou à défaut du pseudo-code.

b- Transition : les activités liées par des transitions représentées par des arcs orientés pouvant porter des gardes qui représentent le cheminement du flot de control de l'application.

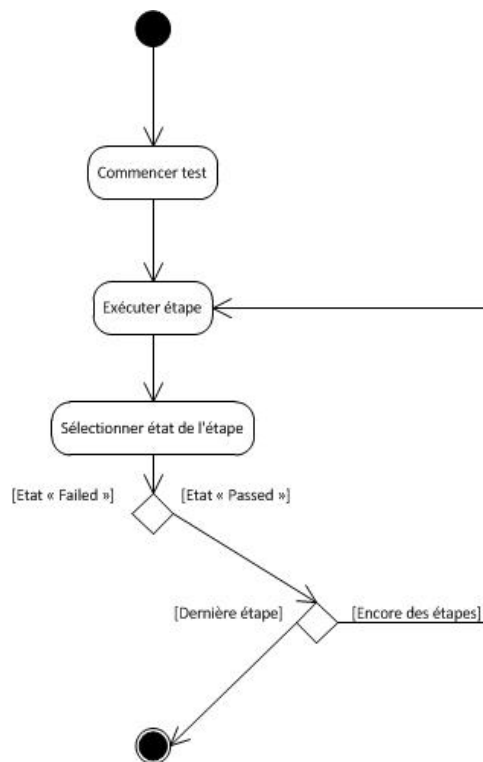


Figure 2.6 : Représentation graphique d'un diagramme d'activité

2.2.6 Avantages et inconvénients d'UML

2.2.6.1 Les points forts d'UML [6]

UML est un langage formel et normalisé :

- ❖ gain de précision.
- ❖ gage de stabilité.
- ❖ encourage l'utilisation d'outils.

UML est un support de communication performant :

- ❖ Il cadre l'analyse.

- ❖ Il facilite la compréhension de représentations abstraites complexes.
- ❖ Son caractère polyvalent et sa souplesse en font un langage universel.

2.2.6.2 Les points faibles d'UML [6]

- ❖ diagrammes UML doivent être synchronisés avec le code du logiciel , qui nécessite du temps pour mettre en place et à entretenir, et ajoute travail à un projet de développement logiciel.
- ❖ La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation., la nécessité de s'accorder sur des modes d'expression communs est vitale en informatique.
- ❖ UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.
- ❖ Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue. les auteurs d'UML sont tout à fait conscients de l'importance du processus, mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse objet performant et standard.

2.3 Démarche de modélisation [7]

UML n'est pas une méthode et ne propose pas une démarche de modélisation explicitant et encadrant toutes les étapes d'un projet. il n'est qu'un langage de modélisation ; ce qui justifie le besoin de faire appel à une méthode qui permet le passage de l'expression des besoins au code de l'application, nous allons donc présenter une des méthode proposée dans la littérature. il s'agit d'une méthode simple et générique proposée par LAURENT AUDIBERT .elle se situe à mi-chemin entre UP (unified process),qui constitue un cadre général très complet de processus de développement et XP qui est une approche minimaliste à la mode centrée sur le code.

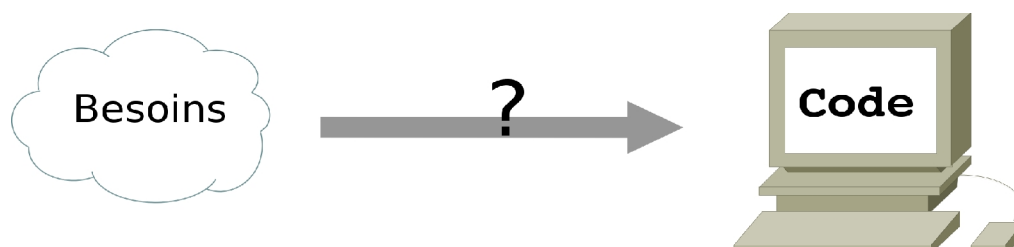


Figure 2.7 : Quelle méthode pour passer de l'expression des besoins au code de l'application ?

2.3.1 Identification des besoins et spécification des fonctionnalités

2.3.1.1 Identification et représentation des besoins : Diagramme de cas d'utilisation

Les cas d'utilisation sont utilisés tout au long du projet. Dans un premier temps, on les crée Pour identifier et modéliser les besoins des utilisateurs .Ces besoins sont déterminés à partir des informations recueillies lors des rencontres entre informaticiens et utilisateurs. Il faut impérativement proscrire toute considération de réalisation lors de cette étape.

Durant cette étape, il faut déterminer les limites du système, identifier les acteurs et recenser les cas d'utilisation. Si l'application est complexe, vous pourrez organiser les cas d'utilisation en paquetages.

2.3.1.2 Spécification détaillée des besoins : diagrammes de séquence système

Dans cette étape, on cherche à détailler la description des besoins par la description textuelle des cas d'utilisation et la production de diagrammes de séquence système illustrant cette description textuelle. Cette étape amène souvent à mettre à jour le diagramme de cas d'utilisation puisque nous sommes toujours dans la spécification des besoins.

Les scénarios de la description textuelle des cas d'utilisation peuvent être vus comme des

Instances de cas d'utilisation et sont illustrés par des diagrammes de séquence système. Il faut au minimum, représenter le scénario nominal de chacun des cas d'utilisation par un diagramme de séquence qui rend compte de l'interaction entre l'acteur, ou les acteurs, et le système.

2.3.1.3 Maquette de l'IHM de l'application (non couvert par UML)

Une maquette d'IHM (Interface Homme-Machine) est un produit jetable permettant aux utilisateurs d'avoir une vue concrète mais non définitive de la future interface de l'application. La maquette peut très bien consister en un ensemble de dessins produits par un logiciel de présentation ou de dessin. Par la suite, la maquette pourra intégrer des fonctionnalités de navigation permettant à l'utilisateur de tester l'enchaînement des écrans ou des menus, même si les fonctionnalités restent fictives. La maquette doit être développée rapidement afin de provoquer des retours de la part des utilisateurs.

2.3.2 Phases d'analyse

2.3.2.1 Analyse du domaine (modèle du domaine)

La modélisation des besoins par des cas d'utilisation s'apparente à une analyse fonctionnelle classique. L'élaboration du modèle des classes du domaine permet d'opérer une transition vers une véritable modélisation objet. L'analyse du domaine est une étape totalement dissociée de l'analyse des besoins. Elle peut être menée avant, en parallèle ou après cette dernière. La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes appelée modèle du domaine.

Ce modèle doit définir les classes qui modélisent les entités ou concepts présents dans le domaine (on utilise aussi le terme de métier) de l'application.

2.3.2.2 Diagramme de classes participantes

Le diagramme de classes participantes est particulièrement important puisqu'il effectue la jonction entre, d'une part, les cas d'utilisation, le modèle du domaine et la maquette, et d'autre part, les diagrammes de conception logicielle que sont les diagrammes d'interaction et le diagramme de classes de conception. Le diagramme de classes participantes effectue la jonction entre les cas d'utilisation.

- **Les classes de dialogues :** Les classes qui permettent les interactions entre l'IHM et les utilisateurs sont qualifiées de dialogues. Ces classes sont directement issues de l'analyse de la maquette. Il y a au moins un dialogue pour chaque association entre un acteur et un cas d'utilisation du diagramme de cas d'utilisation. En général, les dialogues vivent seulement le temps du déroulement du cas d'utilisation concerné.
- **Les classes de contrôles :** Les classes qui modélisent la cinématique de l'application sont appelées contrôles. Elles font la jonction entre les dialogues et les classes métier en permettant aux différentes vues de l'application de manipuler des informations détenues par un ou plusieurs objets métier. Elles contiennent les règles applicatives et les isolent à la fois des dialogues et des entités.
- **Les classes entités :** Les classes métier, qui proviennent directement du modèle du domaine, sont qualifiées d'entités. Ces classes sont généralement persistantes, c'est-à-dire qu'elles survivent à l'exécution d'un cas d'utilisation particulier et qu'elles permettent à des données et des relations d'être stockées dans des fichiers ou des bases de données. Lors de l'implémentation, ces classes peuvent ne pas se

concrétiser par des classes mais par des relations, au sens des bases de données relationnelles.

2.3.2.3 Diagramme d'activités de navigation

Les IHM modernes facilitent la communication entre l'application et l'utilisateur en offrant toute une gamme de moyens d'action et de visualisation comme des menus déroulants ou contextuels, des palettes d'outils, des boîtes de dialogues, des fenêtres de visualisation, etc. Cette combinaison possible d'options d'affichage, d'interaction et de navigation aboutis aujourd'hui à des interfaces de plus en plus riches et puissantes.

UML offre la possibilité de représenter graphiquement cette activité de navigation dans l'interface en produisant des diagrammes dynamiques. On appelle ces diagrammes des diagrammes de navigation. les diagrammes d'activités de navigation sont à relier aux classes de dialogue du diagramme de classes participantes. Les différentes activités du diagramme de navigation peuvent être stéréotypées en fonction de leur nature : « fenêtre », « menu », « menu contextuel », « dialogue », etc. La modélisation de la navigation a intérêt à être structurée par acteur.

2.3.3 Phase de conception

2.3.3.1 Diagrammes d'interaction

Maintenant, il faut attribuer précisément les responsabilités de comportement, dégagée par

Le diagramme de séquence système, aux classes d'analyse du diagramme de classes participantes. Les résultats de cette réflexion sont présentés sous la forme de diagrammes d'interaction UML. Inversement, l'élaboration de ces diagrammes facilite grandement la réflexion. Parallèlement, une première ébauche de la vue statique de conception, c'est-à-dire du diagramme de classes de conception, est construite et complétée.

Durant cette phase, l'ébauche du diagramme de classes de conception reste indépendante des choix technologiques qui seront faits ultérieurement.

Les diagrammes d'interactions sont particulièrement utiles au concepteur pour représenter graphiquement ces décisions d'allocations des responsabilités.

2.3.3.2 Diagramme de classes de conception

L'objectif de cette étape est de produire le diagramme de classes qui servira pour l'implémentation. Une première ébauche du diagramme de classes de conception déjà été élaborée en parallèle du diagramme d'interaction. Il faut maintenant le compléter en précisant

les opérations privées des différentes classes. Il faut prendre en compte les choix techniques, comme le choix du langage de programmation, le choix des différentes bibliothèques utilisées (notamment pour l'implémentation de l'interface graphique) etc.

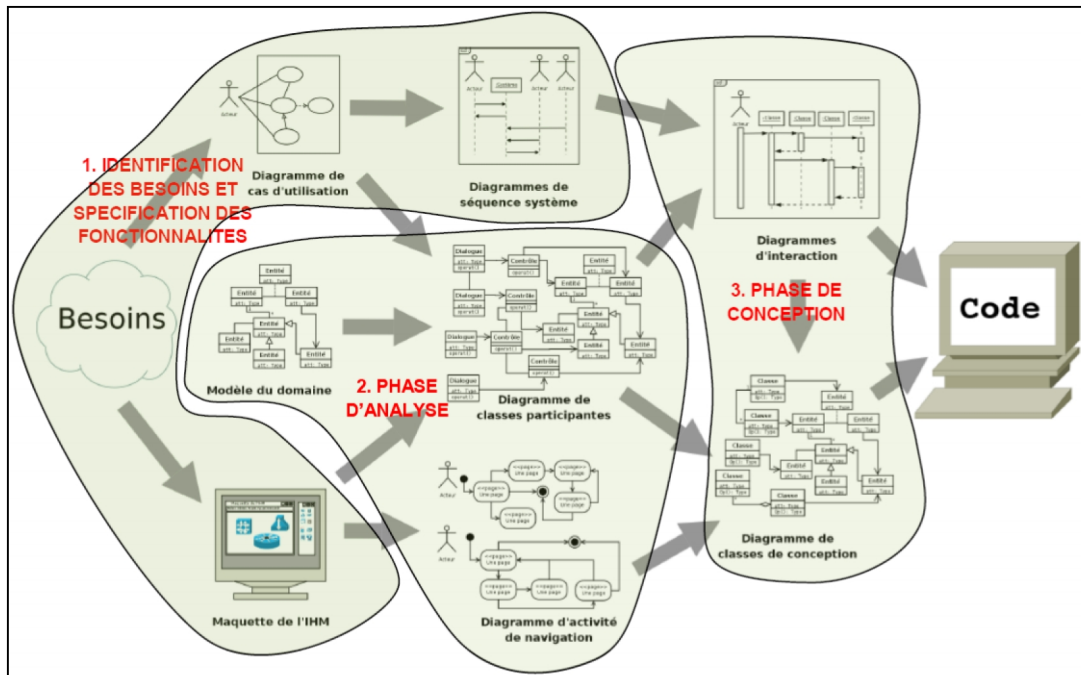


Figure 2.8 : Chaîne complète de la démarche de modélisation du besoin jusqu'au code

2.4 Conclusion

Ce chapitre nous a permis d'étendre nos connaissances sur UML. Nous avons constaté qu'il est un langage riche, il couvre toutes les phases d'un cycle de développement. Il est également ouvert (indépendant du domaine d'application et des Langages d'implémentation), C'est pourquoi nous le choisissons comme un langage de modélisation dans notre travail.

Le prochain chapitre présente les activités effectuées au niveau d'une agence de location de voitures.

3.1 Introduction

Nous souhaitons faire une solution complète d'une location de voiture en ligne qui offre des caractéristiques standards pour le fonctionnement de l'agence. Pour cela nous avons fait une étude d'existant qui est une étape très importante presque dans tous les projets informatiques.

Nous commençons par la présentation de la société (l'agence), le lieu de ce stage. Ensuite, nous citons les documents nécessaires, et les différentes activités d'une location.

3.2 Description de l'organisme d'accueil de l'agence

Miley_auto est une agence de location de voiture avec et sans chauffeur dont le siège est le centre ville de Mila, plus précisément dans la rue Aliouche Nouar. Cette agence a débuté son activité en 2008 et dispose d'une flotte récente de différentes marques et modèles de voitures. Afin de subvenir aux besoins d'une clientèle, elle se propose aussi des tarifs compétitifs.

3.3 Les étapes suivies pour effectuer une location de véhicule

Toutes les activités de la location des voitures résumées comme suit :

Lorsque le client visite l'agence, il se déplace au bureau de la réception pour demander une allocation de voiture. Le responsable lui présente un contrat avec des articles afin de voir les termes et les conditions de la location entre autre :

- le client possède un permis de conduire supérieur à deux ans.
- l'âge minimum requis pour un conducteur (client) est 21 ans.
- le client qui signe le contrat de location est celui qui conduit le véhicule.
- La réception et la livraison de la voiture avec des outils de dépannage est dans le siège de l'agence.
- la durée et le kilométrage limité d'une location doivent être respectés par tous les clients.
- les voitures louées circulent uniquement dans le territoire national.
- le client doit rendre la voiture dans le même état dans la quelle il la reçue (ex : demi plein de carburant).

Par la suite, le responsable de l'agence demande au locataire de fournir les documents nécessaires (si le client est un résident : photo copier de permis. Si le client est un étranger : les mêmes papiers que le résident avec la photo copier de passeport. En cas d'une société : photo copier de registre, les données fiscale). Après avoir examiné le locataire, il choisit la voiture souhaitée et voit son état. Après l'accord du client, ce dernier signe le contrat présenté

en deux exemplaires, l'un pour le client et l'autre pour l'agence puis il paye la facture de location qui est calculé en fonction du tarif stipulé dans le contrat (les prix sont fixes). Durant la période de la location, la voiture est sous la garantie du client.

Une prolongation de la location ou le changement d'une voiture n'est plus possible après la prise du véhicule. Dans ce cas là, le client doit effectuer une nouvelle réservation.

3.4 Etude des documents

3.4.1 Définitions

- Un document est un support qui regroupe un ensemble de rubriques correspondant à des informations manipulées par l'entreprise ou l'établissement (agence).
- Une étude de document consiste à définir la désignation du document, son code (s'il existe), sa source sa destination, et son utilisation.
- L'analyse des documents et des fichiers est nécessaire car elle nous permet de collecter les informations qui serviront à la conception de notre site web pour le suivi d'agence de location de voiture.

3.4.2 Notation

- **Code** : identificateur du document.
- **Désignation** : nom du document.
- **Source** : le poste ou le document est établi.
- **Destination** : destination du document (poste récepteur).
- **Rôle** : a quoi sert ce document.
- **Nombre d'exemplaire**
- **Le contenu du document**
- **Observation sur le document**

3.4.3 Légende des types

- **A** : Alphanumérique.
- **N** : Numérique.
- **AN** : Alpha Numérique.
- **Date** : Date.

3.4.4 Fiche d'étude du document Contrat

Le seul document qui on été identifier durant notre stage est : le contrat

Code et le rôle du document				
Code : contrat Désignation : Contrat Rôle : Comprend des articles complets et contiennent la totalité d'information d'une location				
Caractéristique du document				
Source : responsable de l'agence Destinataire : locataire Nombre d'exemplaires : 2 Nature : sortie				
Elément d'information				
Rurique	Désignation	type	Taille	observation
Num_contrat	Numéro du contrat	N	20	
Nom_resp	Nom du responsable	A	20	
Prénom_resp	Prénom du responsable	A	20	
Adr_agence	adresse de l'agence	AN	50	
Num_tel_agen	Numéro téléphone de l'agence	AN	20	
Num_ccp	Numéro de compte ccp de l'agence	N	20	
Prix_u	Prix unitaire de location d'une voiture	N	20	
Prix_totl	Prix totale d'une location	N	50	
immatriculation	Numéro de matricule d'une	AN	20	

	voiture			
Marque	Marque d'une voiture	AN	20	
Type	Type d'une voiture	N	20	Tourisme, commerciale
Puissance	Puissance d'une voiture	N	20	
Placese_assises	Places assises d'une voiture	N	20	
Adr_société	Adresse de société	AN	50	
Tél_société	Numéro téléphone de société	AN	50	
Num_R	Numéro de registre pour la société (client)	N	20	
Nom_client	Nom de client	A	20	
Prénom_client	Prénom de client	A	20	
Date_nais	Date naissance de client	date	20	Supérieure à 21 ans(age)
Adr_client	Adresse de client	AN	50	
Num_permis	Numéro de permis	N	20	
date_permis	Date d'édition de permis	date	20	Plus que 2 ans
Tél_client	Numéro de téléphone du client	AN	20	
Date_départ	Date départ	date	10	AAAA/MM/JJ
Date_retour	Date retour	date	10	AAAA/MM/JJ
caution	Caution de location	N	20	

Tableau 3.1 : fiche d'étude du document contrat

3.5 Conclusion

Dans ce chapitre, nous avons mené une étude sur l'agence de location de voiture, qui a porté sur plusieurs aspects. Durant cette étude nous avons pu recenser toutes les informations nécessaires à notre travail à savoir la réalisation d'un site web.

Dans le prochain chapitre nous passons à l'étude conceptuelle de notre site.

4.1 Introduction

Dans ce chapitre nous allons utiliser UML pour modéliser notre site de location de voiture. Comme un premier pas, nous allons commencer par la phase d'identification des besoins, puis nous allons passer à la phase d'analyse et nous terminons par la phase de conception.

4.2 La phase d'identification des besoins

Dans cette phase de allons réaliser les diagrammes suivants :

- Diagramme de cas d'utilisation ;
- Diagramme de séquence système.

4.2.1 Diagramme de cas d'utilisation

4.2.1.1 Identification des acteurs et des besoins fonctionnels

Les acteurs essentiels qui interagissent avec le système sont les suivants :

1 Le client : c'est la personne qui accède au site pour effectuer les tâches suivantes :

- Consulter les offres offertes par l'agence: le client effectue des recherches pour trouver l'offre qui satisfait ses besoins.
- réserver une voiture : le client lance une réservation d'une voiture en ligne lorsque il trouve une offre qui correspondant à ses besoins. Cette réservation est effectué lorsque la voiture est disponible (n'est pas réservé ni loué) dans la liste des voitures existe dans l'agence. Le paiement de la facture est effectué après que le client rempli le formulaire de réservation, durant le délai précis entre la date de réservation et la date du début de la livraison. Le paiement se fera par un versement au compte ccp de l'agence ou en espèce au choix du client.
- Consulter les réservations : pour voir si son réservation est acceptée ou pas.

2 L'administrateur : c'est la personne qui gère le site et effectué les mises à jour nécessaires.

Après l'authentification l'administrateur peut faire les tâches suivantes :

- Valider-annuler réservations: l'administrateur visualise et analyse les réservations effectuaient par un nombre des clients au un moment donné soit pour :
 - valider les réservations : pour chaque demande de réservation l'administrateur valide les demandes qui sont déjà payés. (L'administrateur reçoit un bon de ressure confirmant que le paiement a été effectué par le client) ;

- annuler les réservations : L'administrateur annule les réservations qui n'ont pas payées par les clients durant le délai précis.
- Ajouter voiture : l'ajout d'une nouvelle voiture avec ses caractéristiques (immatriculation, type, marque, ...).
- Modifier voiture : la modification des informations d'une voiture existante (le prix, état de la voiture...).
- Supprimer voiture: la suppression d'une voiture depuis la liste des voitures existe.

4.2.1.2 Identification des diagrammes des cas d'utilisation

Pour chacun des acteurs précédents nous avons les cas d'utilisation suivants :

N°	Cas d'utilisation	Acteur
1	Consulter les voitures	Le client
2	Réserver la voiture	Le client
3	S'authentifier	L'administrateur
4	Valider-Annuler les réservations	L'administrateur
5	Ajouter la voiture	L'administrateur
6	Supprimer la voiture	L'administrateur
7	Modifier la voiture	L'administrateur

Tableau 4.1 :« les cas d'utilisations »

1 Diagramme de cas d'utilisation pour le client

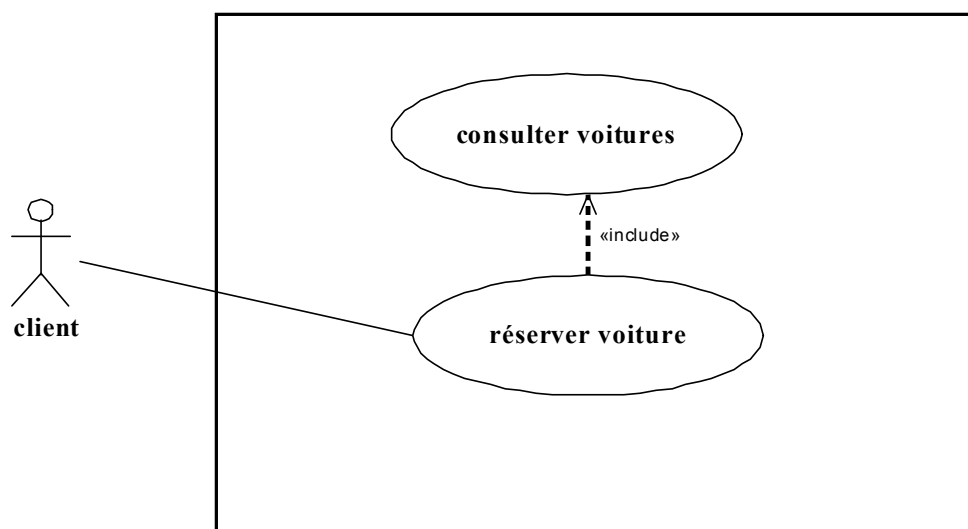


Figure 4.1 : le diagramme de cas d'utilisation «client»

2 le diagramme de cas d'utilisation pour l'administrateur

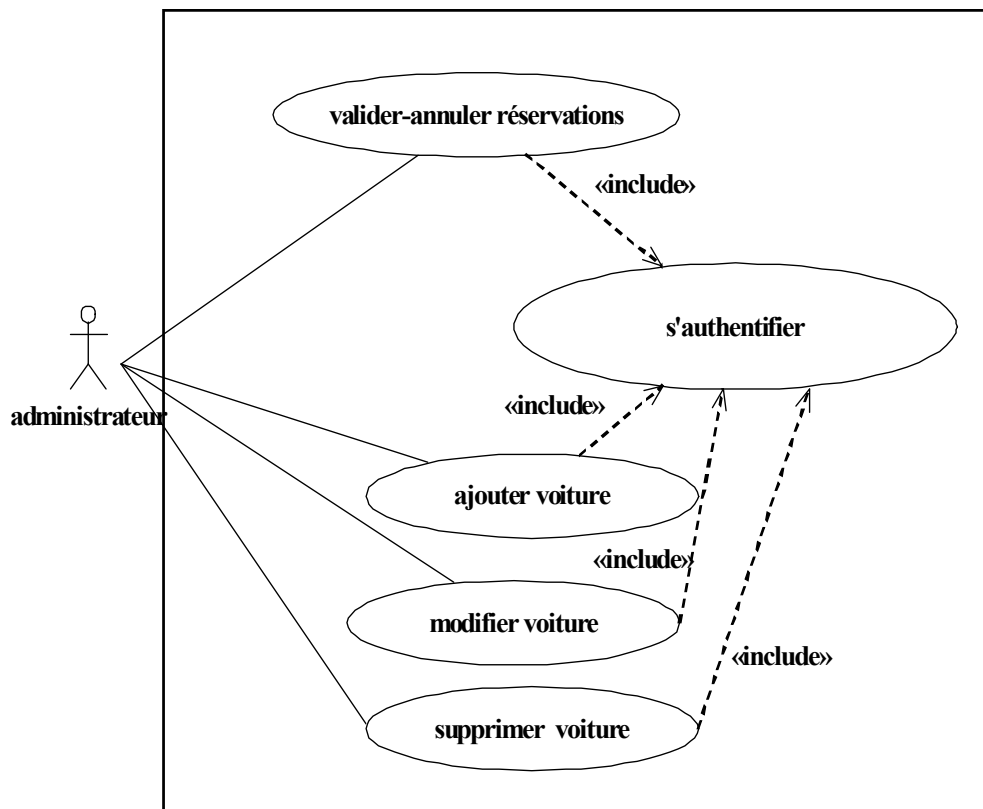


Figure 4.2 : Diagramme de cas d'utilisation «administrateur»

4.2.1.3 Description textuelle des cas d'utilisation

1 Fiches descriptives des cas d'utilisation pour le client

- cas d'utilisation «consulter voitures»

Cas d'utilisation	Consulter les voitures de l'agence
But	Trouver une voiture qui correspond aux besoins du client
Acteur	Client
Pré condition	/
Post condition	Le client a trouvé la voiture recherchée
Scénario nominal	1-le client demande la page de consultation. 2-le système lui affiche la liste des voitures disponibles. 3-le client sélectionne une voiture pour voir ses caractéristiques.

Tableau 4.2 : «consulter voitures»

• cas d'utilisation «réserver voiture»

Cas d'utilisation	Réserver voiture
But	Réserver la voiture qui correspond aux besoins du client
Acteur	Client
Pré condition	-Le client trouve la voiture voulu
Post condition	- une réservation d'une voiture est enregistrée. - Le client doit payer à travers le versement ccp ou bien en espèce afin de confirmer sa réservation
Scénario nominal	1-le client lance une réservation de la voiture sélectionnée. 2-le système lui affiche le formulaire de réservation. 3-le client saisit ses informations nom, prénom, num_permis..., la date début et la date fin de la location puis il sélectionne le mode de paiement (cache ou par un versement CCP en envoyant les papiers confirmant le paiement). Par la suite, il valide l'opération. 4-le système vérifie et contrôle les informations puis enregistre la nouvelle réservation.
Scénario alternatif	1-le système affiche un message d'erreur lorsque le client manque un champ dans le formulaire. Le client reprend le parcours à partir de l'étape 3

Tableau 4.3 : «réserver voiture»

2 Fiches descriptives des cas d'utilisation pour l'administrateur

- cas d'utilisation «s'authentifier»**

Cas d'utilisation	S'authentifier
But	Permettre à l'administrateur d'accéder au site.
Acteur	L'administrateur
Pré condition	/
Poste condition	-L'administrateur s'authentifier.
Scénario nominal	1. L'administrateur demande l'accès au site 2. le système lui affiche la page d'authentification 3. l'administrateur saisie le login et le mot de passe puis il valide. 4. le système vérifie la validation des champs saisies puis ouvre l'espace de l'administrateur
Scénario alternatif	Si les informations saisies sont incorrectes, le système affiche un message d'erreurs et demande la ressaisie des informations. -l'administrateur reprend l'enchaînement a partir de l'étape 3

Tableau 4.4 : «s'authentifier»

- cas d'utilisation cas d'utilisation «valider-annuler réservation»**

Cas d'utilisation	Valider-Annuler les réservations des clients
But	Permettre d'accepter les réservations payées par les clients pendant un délai précis avant la livraison de la voiture ou d'annuler les réservations non payées.
Acteur	L'administrateur
Pré condition	- L'administrateur doit s'authentifier. - les réservations sont disponibles.
Poste condition	-les réservations sont validées ou annulées.
Scénario nominal	1. L'administrateur demande de visualiser la liste des réservations effectuées 2. le système lui affiche la page demandée. 3. l'administrateur change l'état de la réservation (accepter les réservations payées et refuser celles non payées) puis valide l'opération.

	4. le système enregistre le nouvel état de la réservation puis envois un email au client.
--	---

Tableau 4.5 : «valider-annuler réservation »

• **cas d'utilisation «ajouter voiture»**

Cas d'utilisation	Ajouter une voiture
But	L'ajout d'une nouvelle voiture.
Acteur	L'administrateur
Pré condition	-L'administrateur doit s'authentifier. -le voiture n'existent pas dans la base de donné.
Poste condition	La voiture est ajoutée.
Scénario nominal	1. L'administrateur demande la page d'ajout. 2. le système affiche le formulaire d'ajout d'une voiture. 3. l'administrateur saisit les caractéristiques de la voiture et valide. 4. le système enregistre les informations après les vérifications.
Scénario alternatif	Si les informations sont incorrectes ou incomplètes. -Le système affiche un message d'erreurs et demande la ressaisie des informations. -l'administrateur reprend l'enchaînement à partir de l'étape 3

Tableau 4.6 : «ajouter voiture»

• cas d'utilisation «modifier voiture»

Cas d'utilisation	Modifier une voiture existante
But	Modifier les caractéristiques d'une voiture (le prix, l'état...)
Acteur	L'administrateur
Pré condition	-L'administrateur doit être s'authentifier -la voiture précisée existe
Poste condition	La voiture a été modifiée
Scénario nominal	1. L'administrateur demande la visualisation de la liste des voitures. 2. Le système affiche la liste de tous les voitures existantes. 3. L'administrateur sélectionne une voiture pour modification. 4. le système affiche le formulaire de modification d'une voiture précisée. 5. l'administrateur modifie les informations d'une voiture (le prix ou l'état) puis il valide. 6. le système enregistre les modifications après les vérifications.

Tableau 4.7 : «modifier voiture»

• cas d'utilisation «supprimer voiture»

Cas d'utilisation	Supprimer une voiture
But	La suppression d'une voiture
Acteur	L'administrateur
Pré condition	-L'administrateur doit s'authentifier -La voiture existe
Poste condition	La voiture est supprimée
Scénario nominal	1. L'administrateur demande la visualisation de la liste des voitures. 2. Le système affiche la liste de tous les voitures existantes. 3. l'administrateur choisit la voiture qui veut la supprimée et valide la suppression. 4. le système affiche un message de succès (suppression effectué).

Tableau 4.8 : «supprimer voiture»

4.2.2 Les diagrammes de séquence système

4.2.2.1 Pour le client

1 Consulter voitures

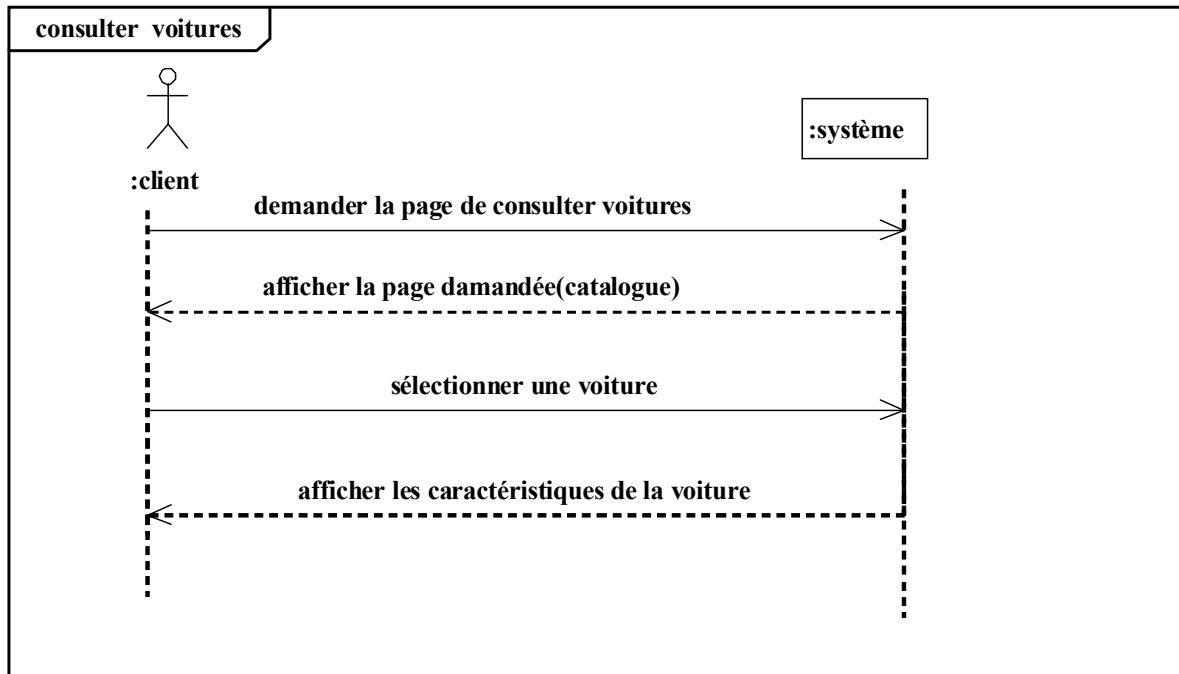


Figure 4.3 : diagramme de séquence système «consulter voitures»

2 réserver voiture

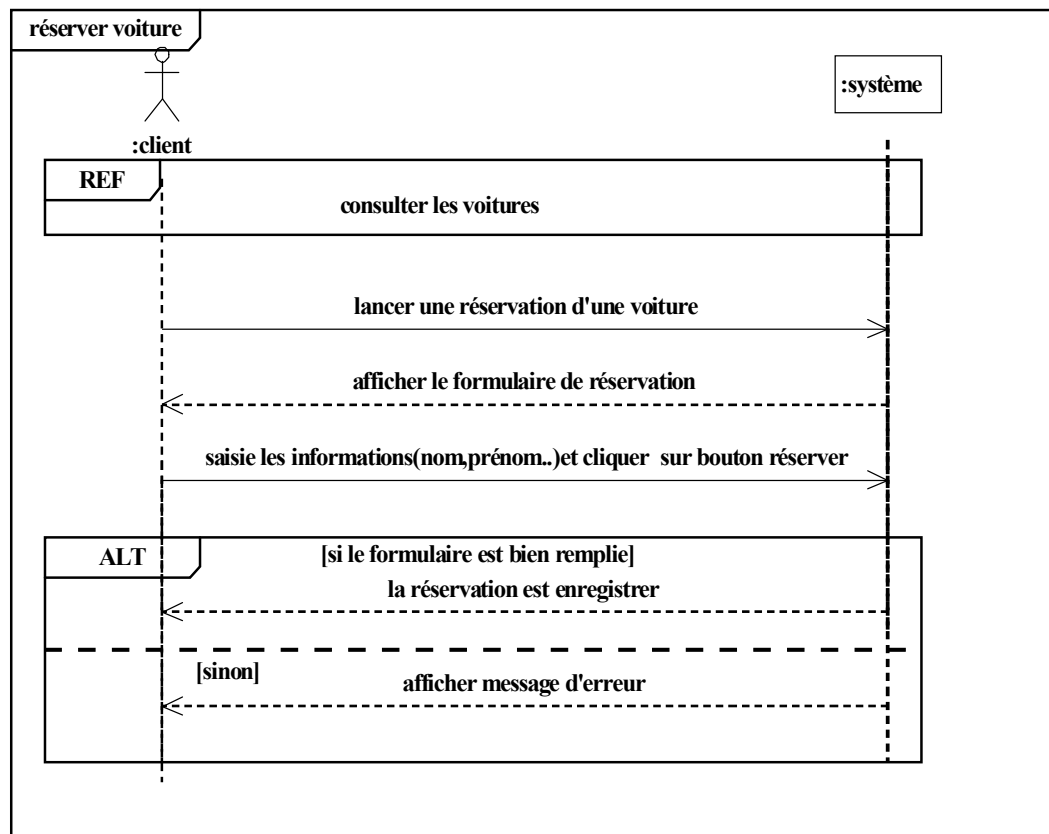


Figure 4.4 : diagramme de séquence système «réserver voiture»

4.2.2.2 Pour l'administrateur

1 S'authentifier

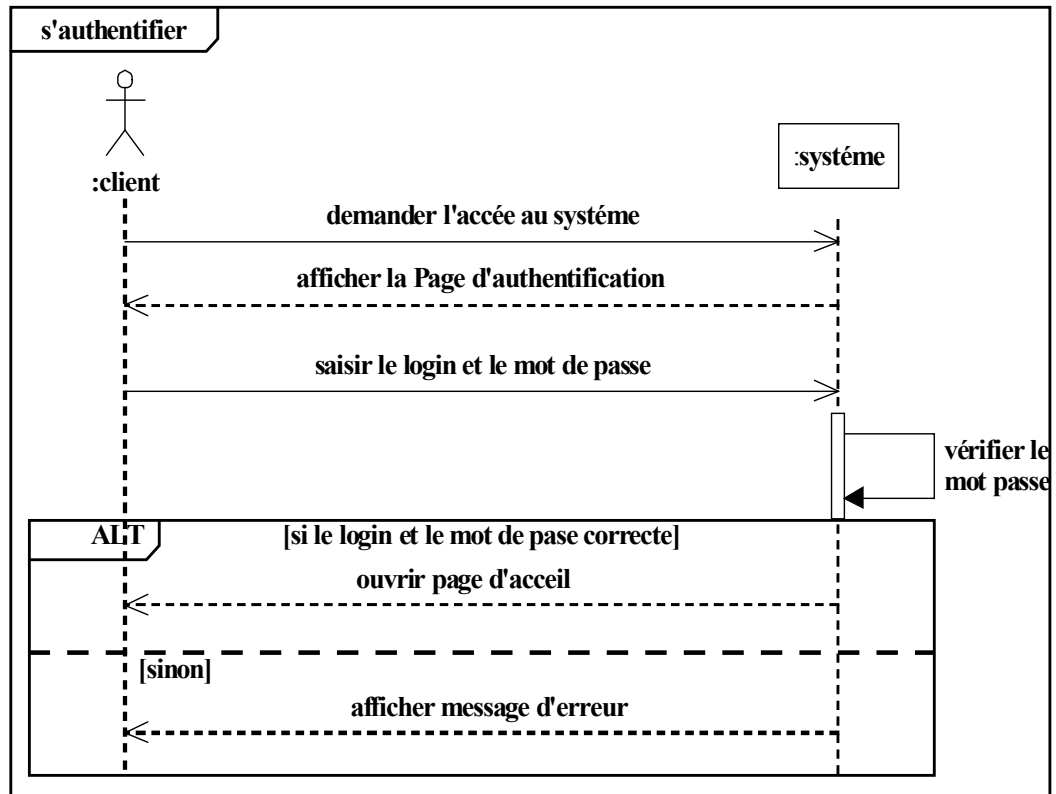


Figure 4.5 : diagramme de séquence système «s'authentifier»

2 Valider-Annuler réservation

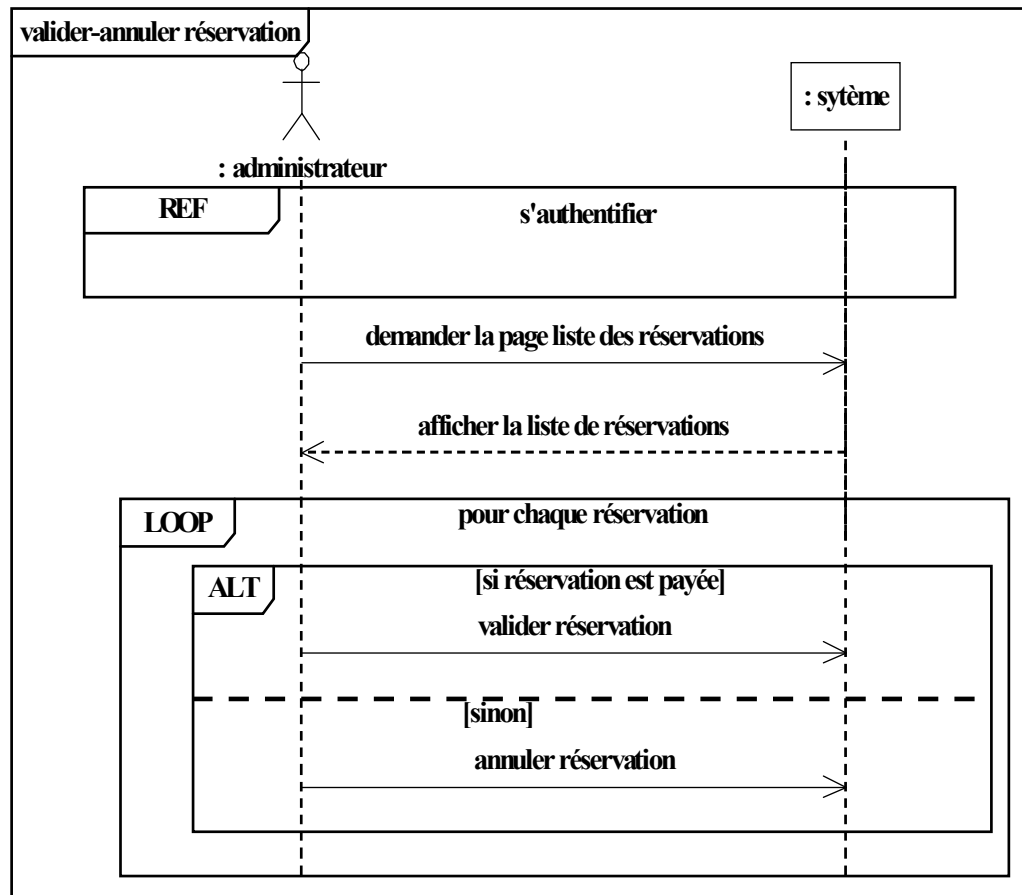


Figure 4.6 : diagramme de séquence système «valider-annuler réservation »

3 Ajouter voiture

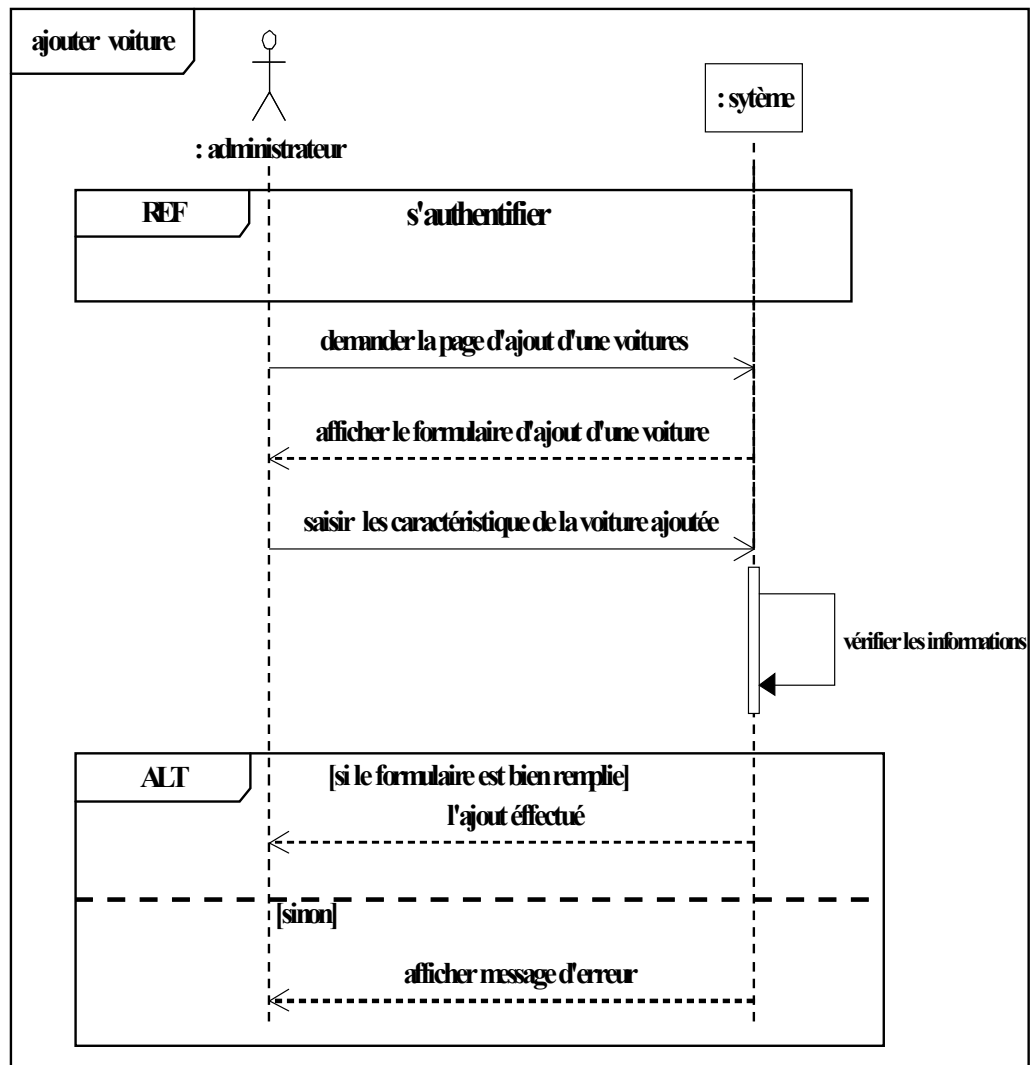


Figure 4.7 : diagramme de séquence système «ajouter voiture»

4 Modifier voiture

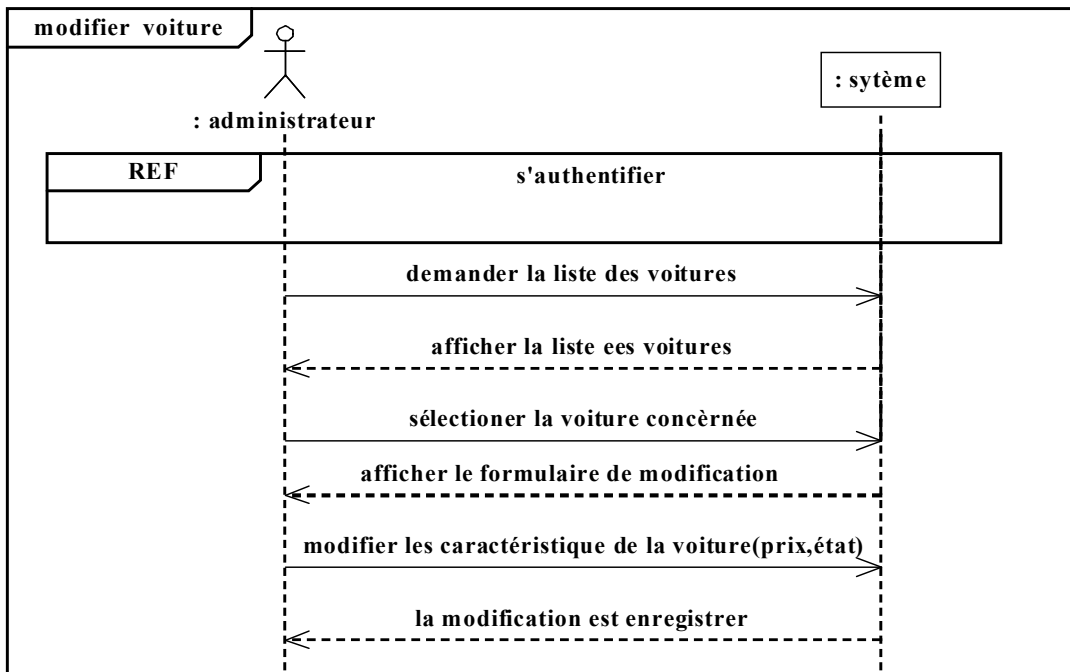


Figure 4.8 : diagramme de séquence système «modifier voiture»

5 Supprimer voiture

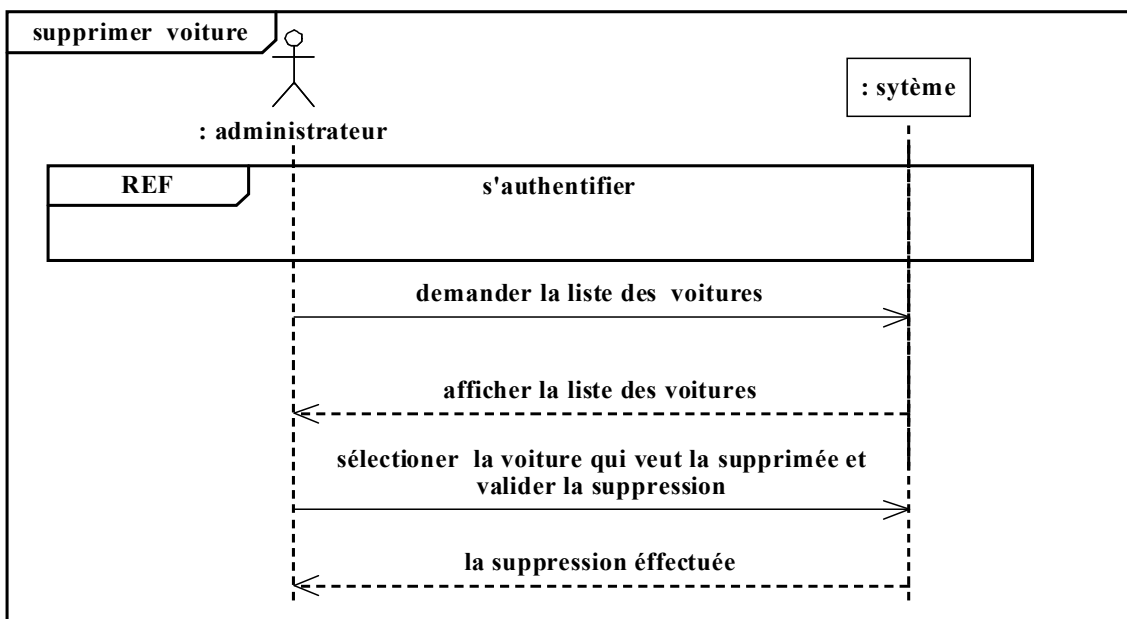


Figure 4.9: diagramme de séquence système «supprimer voiture»

4.3 La phase d'analyse

4.3.1 Le modèle du domaine

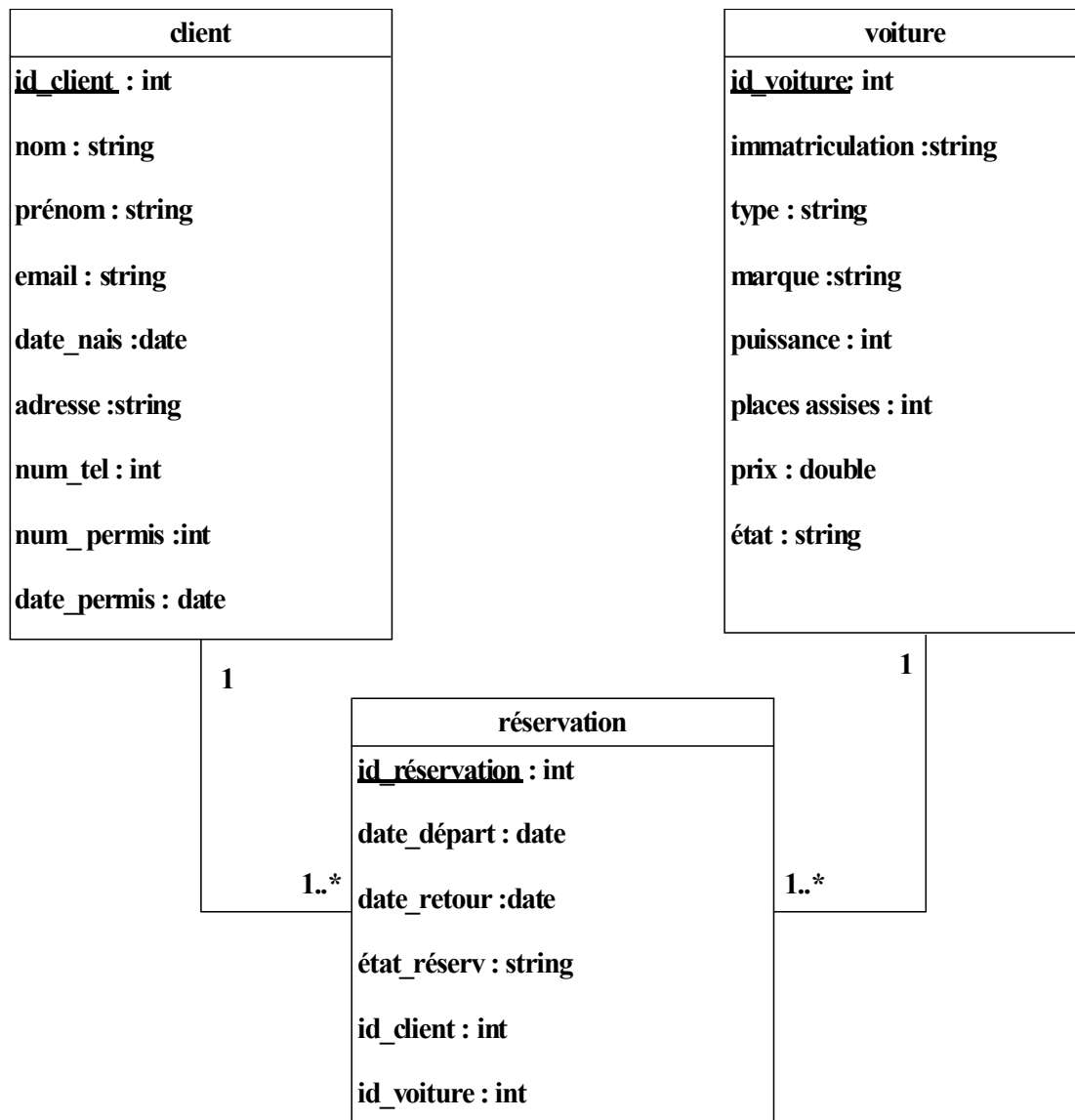


Figure 4.10 : le modèle du domaine

4.3.2 Diagramme de classe participantes

4.3.2.1 Pour le client

1 Consulter voitures

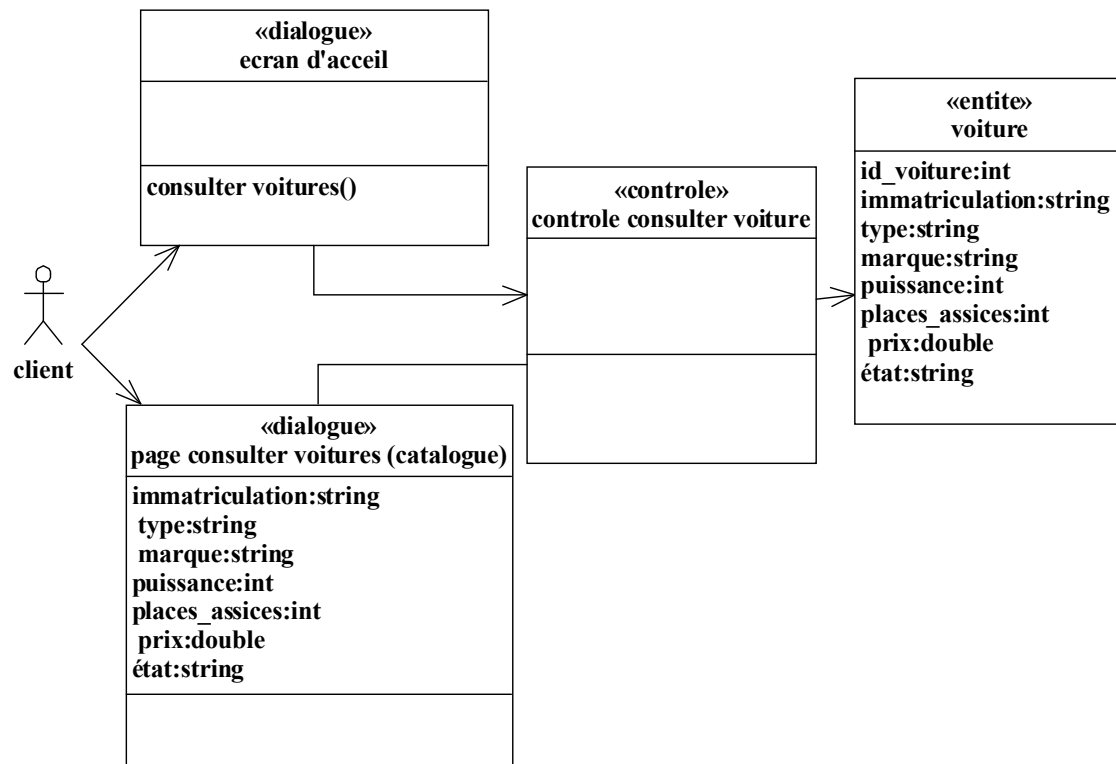


Figure 4.11 : diagramme de classe participante « consulter voitures »

2 réserver voiture

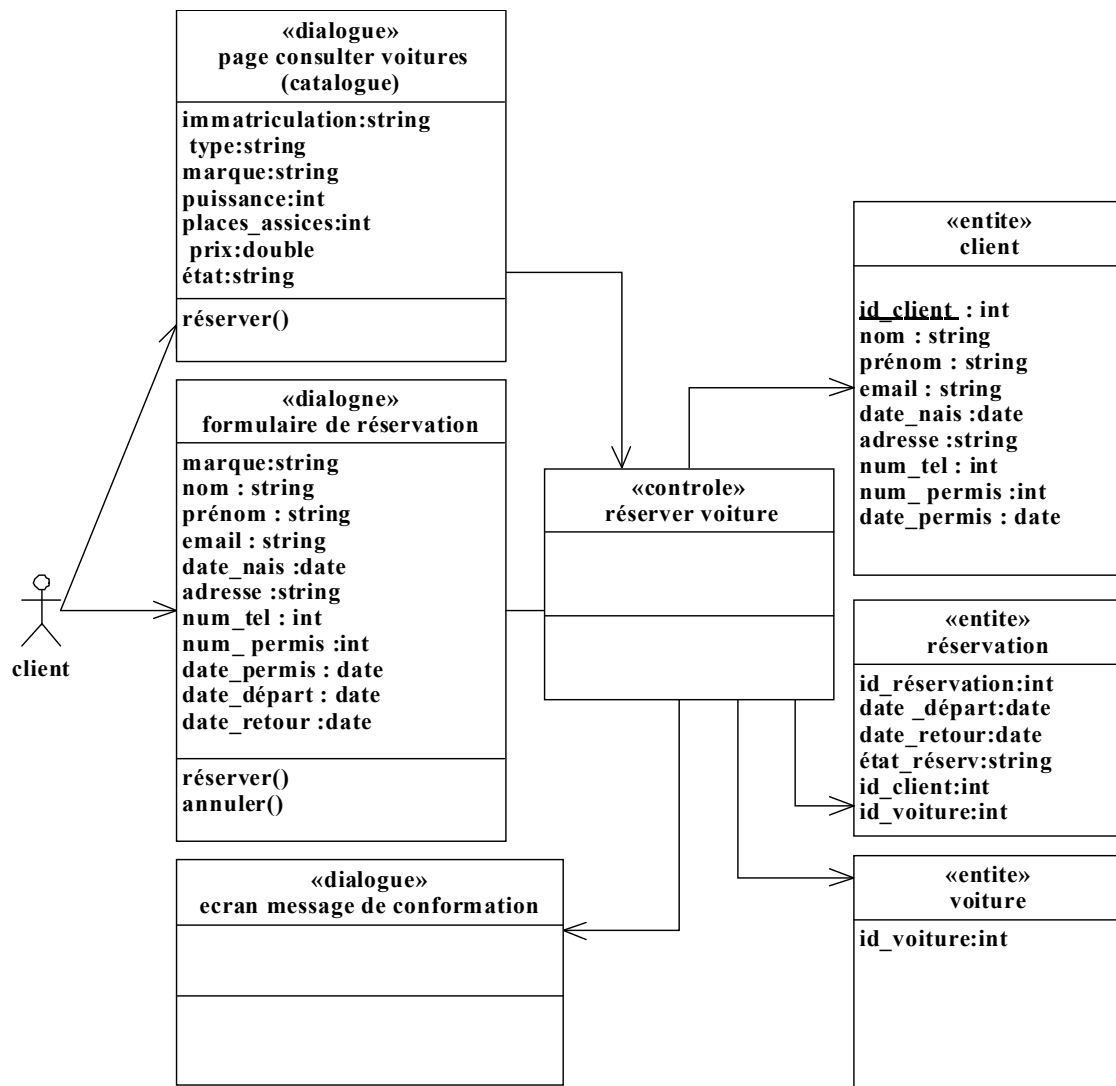


Figure 4.12 : diagramme de classe participante « réserver voiture »

4.3.2.2 Pour l'administrateur

1 S'authentifier

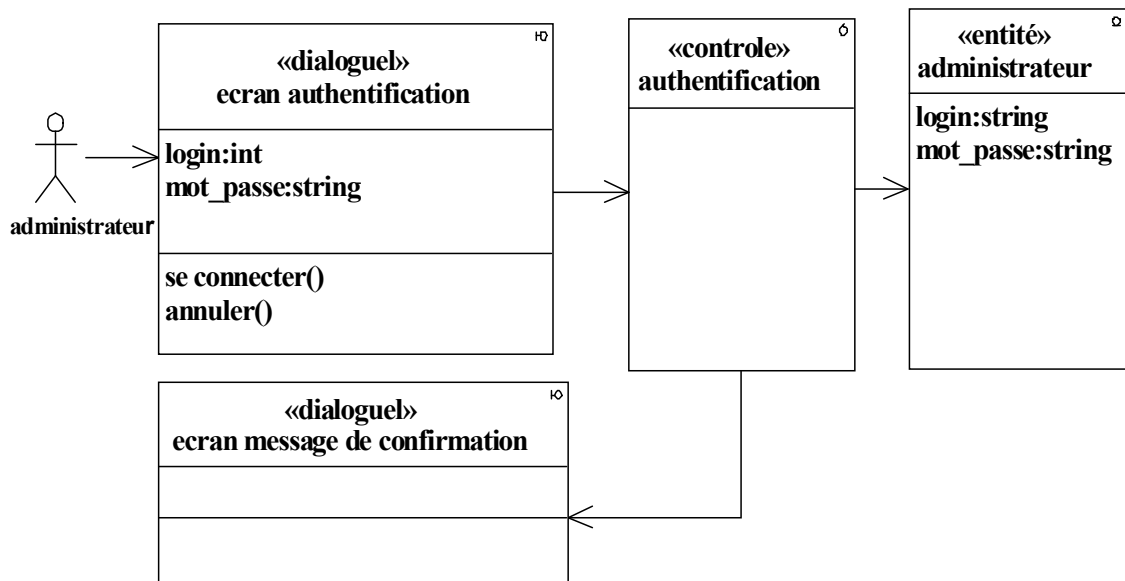


Figure 4.13 : diagramme de classe participante « s'authentifier »

2 Valider-Annuler réservation

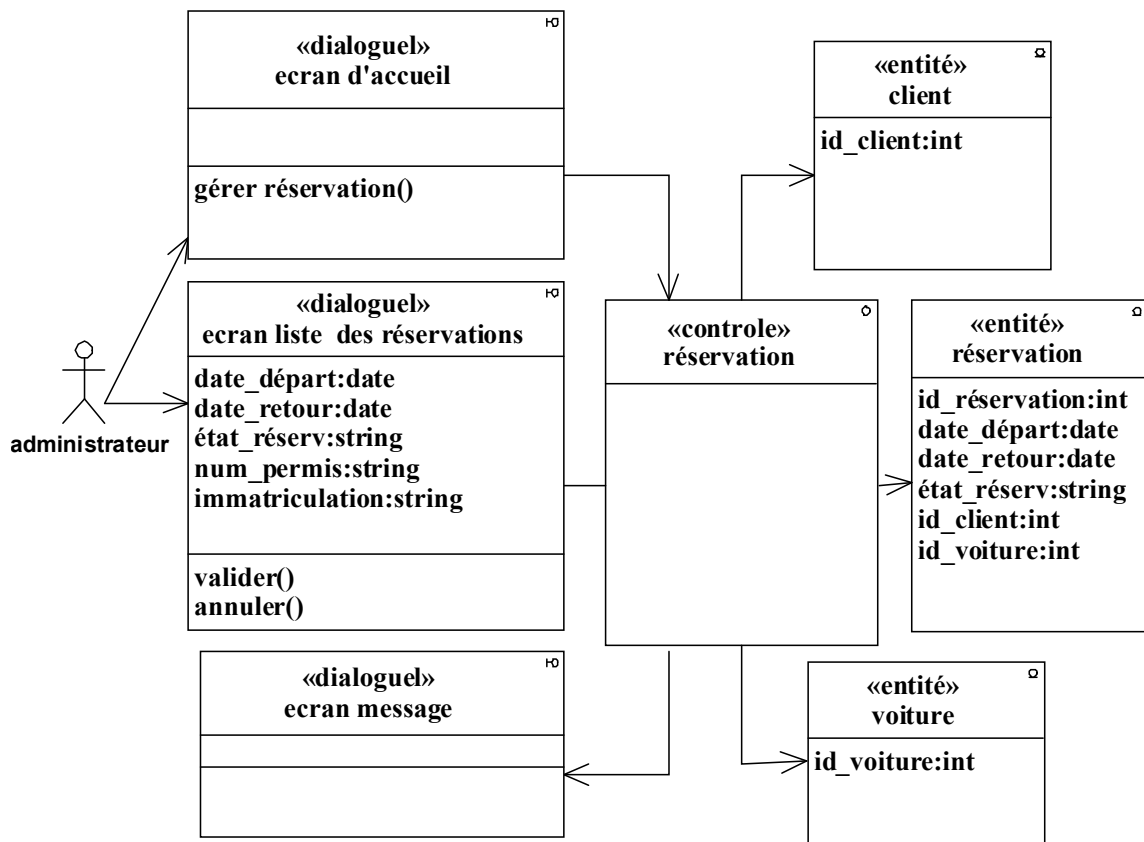


Figure 4.14 : diagramme de class participante «valider-Annuler réservation»

3 Ajouter voiture

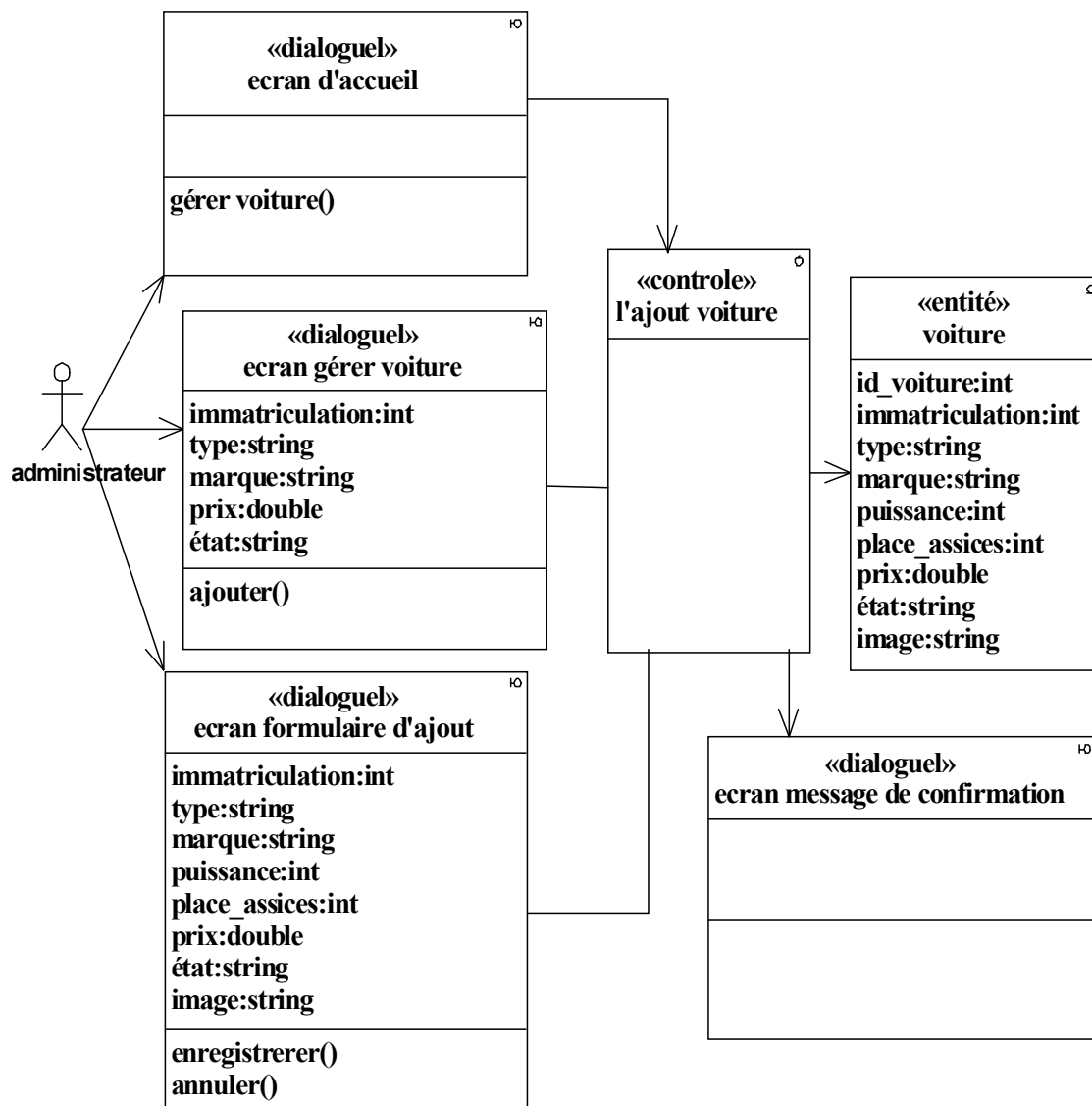


Figure 4.15 : diagramme de class participante « ajouter voiture»

4 Modifier voiture

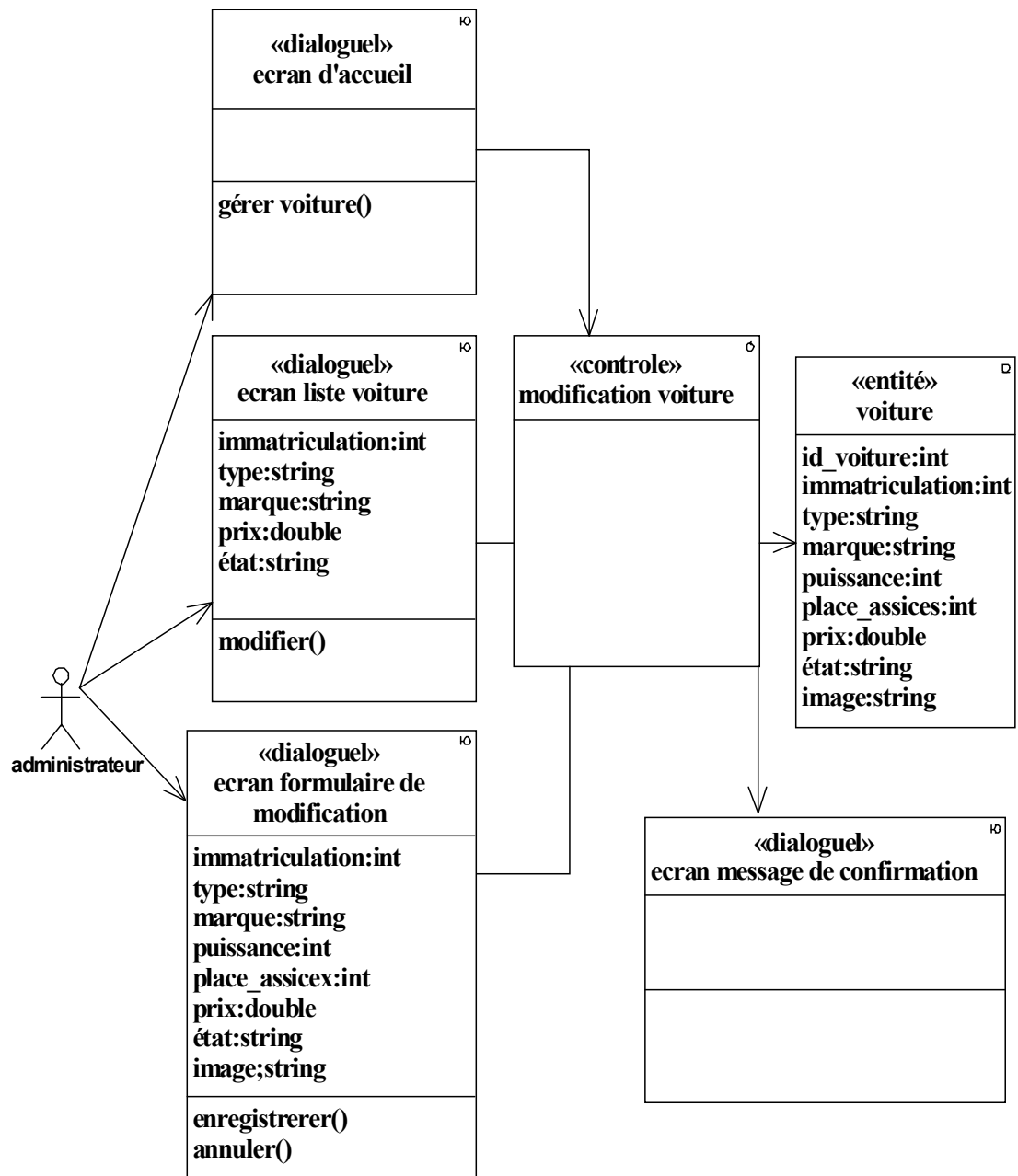


Figure 4.16 : diagramme de class participante « modifier voiture»

5 Supprimer voiture

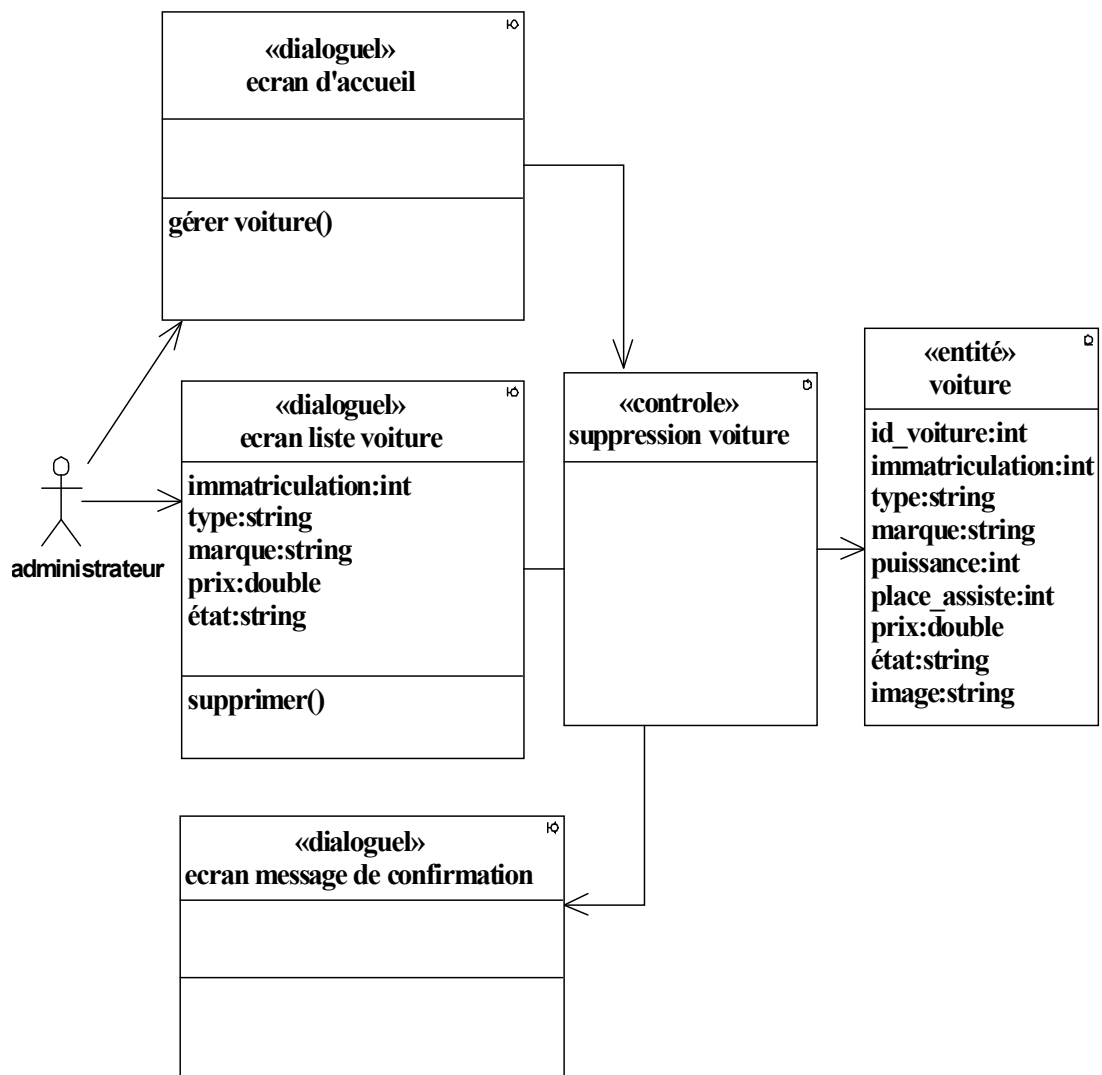


Figure 4.17 : diagramme de class participante « supprimer voiture»

4.3.3 Diagrammes d'activités de navigation

4.3.3.1 Pour le client

1 consulter voitures

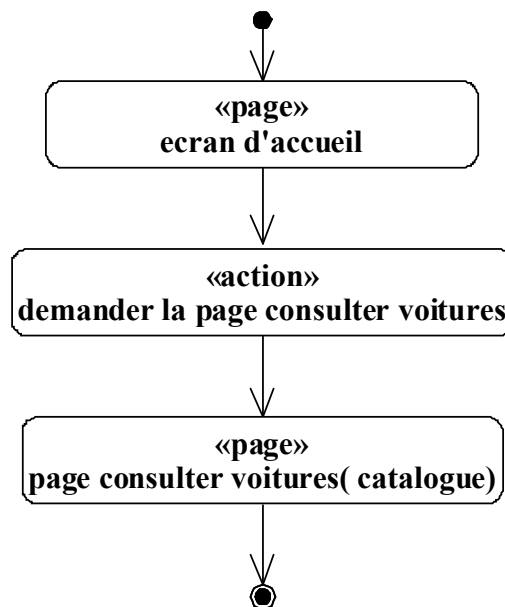


Figure 4.18 : diagramme d'activité « consulter voitures »

2 réserver voiture

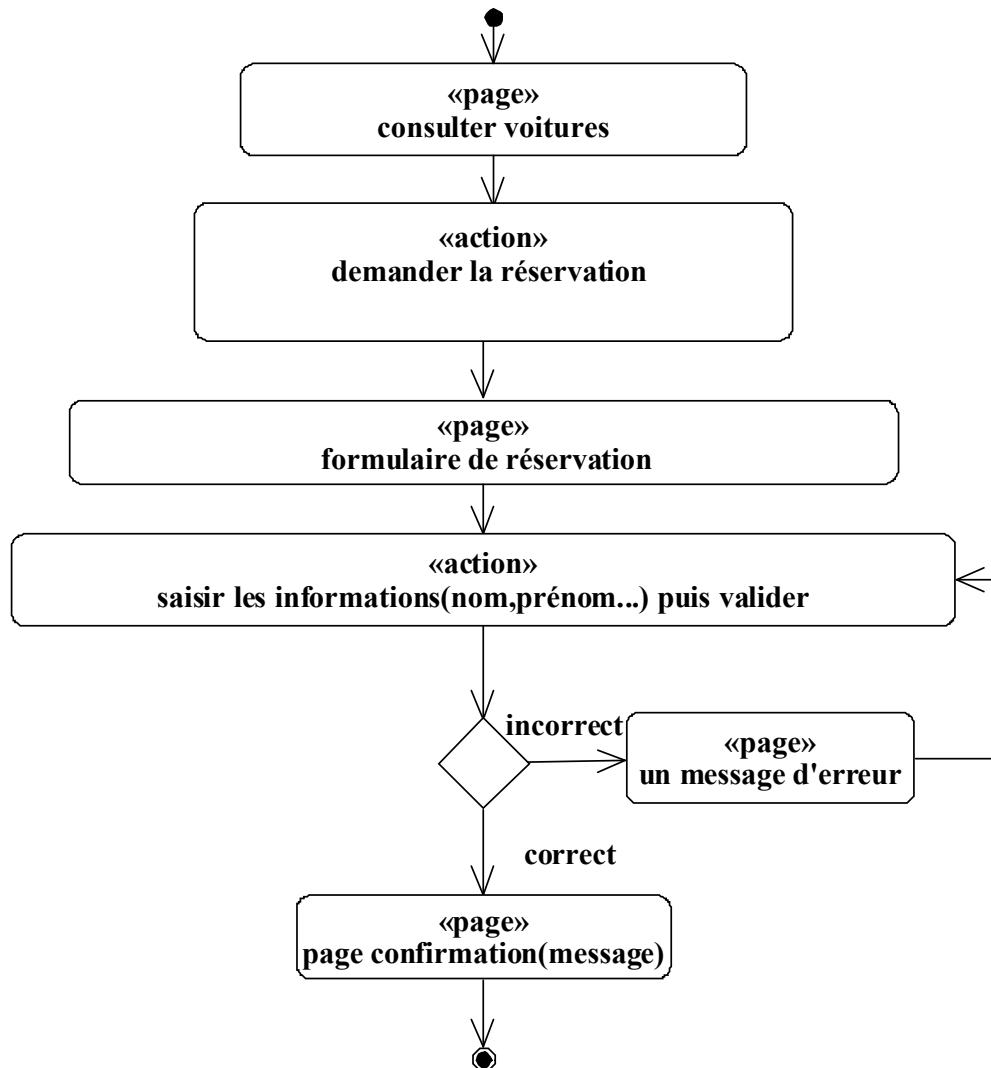


Figure 4.19 : diagramme d'activité « réserver voiture »

4.3.3.2 Pour l'administrateur

1 S'authentifier

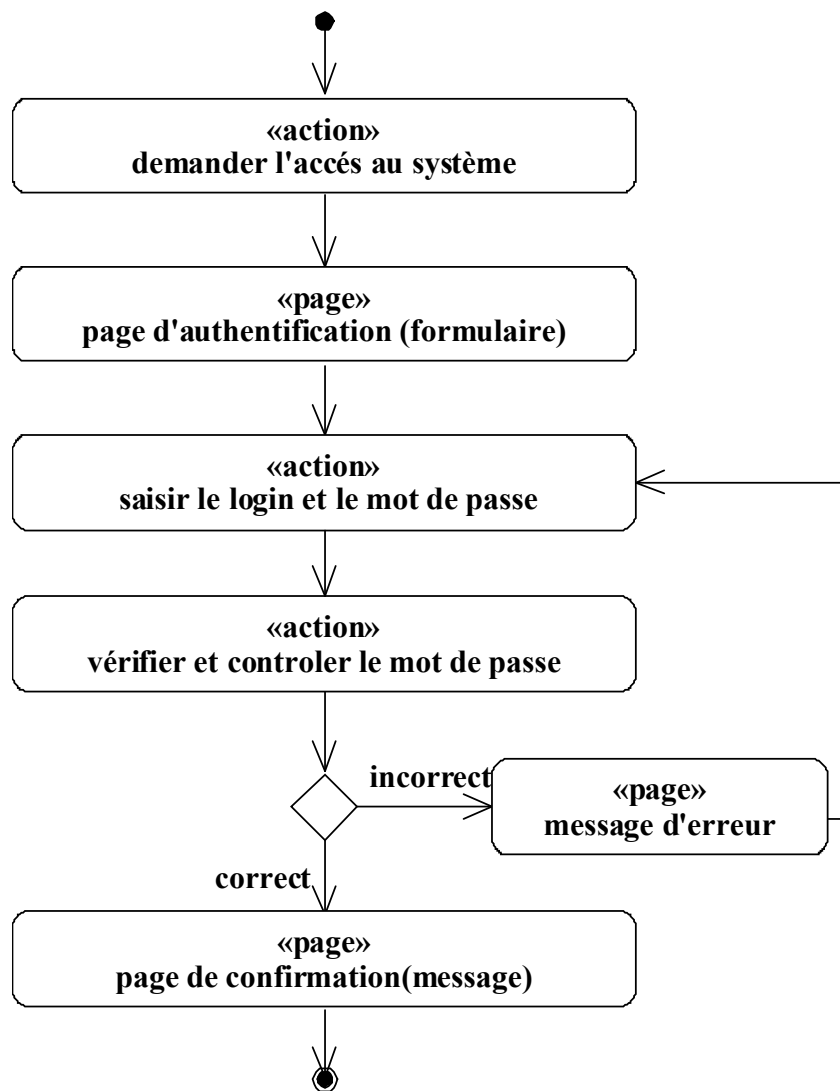
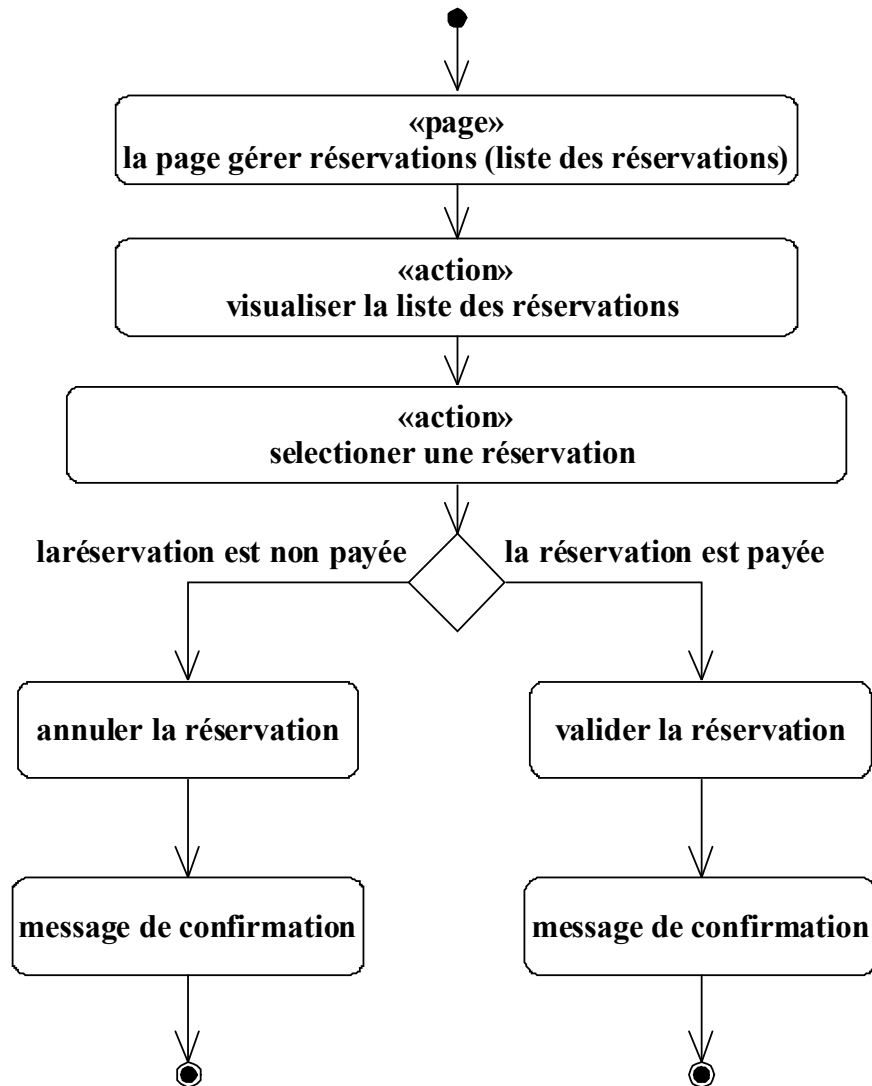


Figure 4.20 : diagramme d'activité « s'authentifier »

2 Valider-Annuler réservation**Figure 4.21 : diagramme d'activité « valider-Annuler réservation»**

3 Ajouter voiture

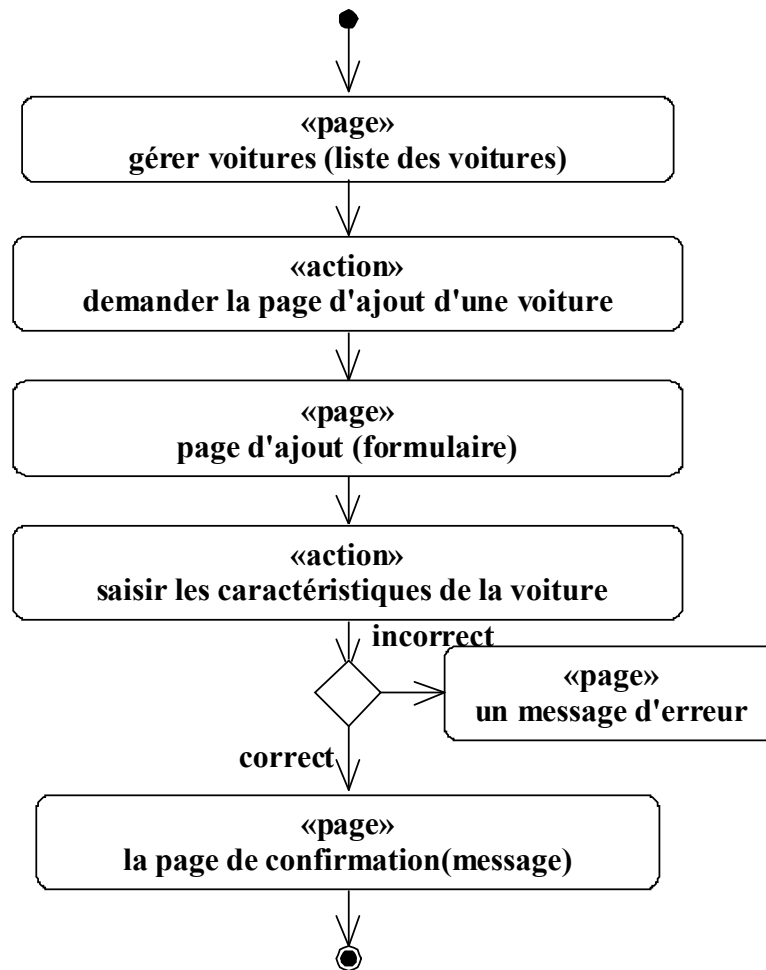
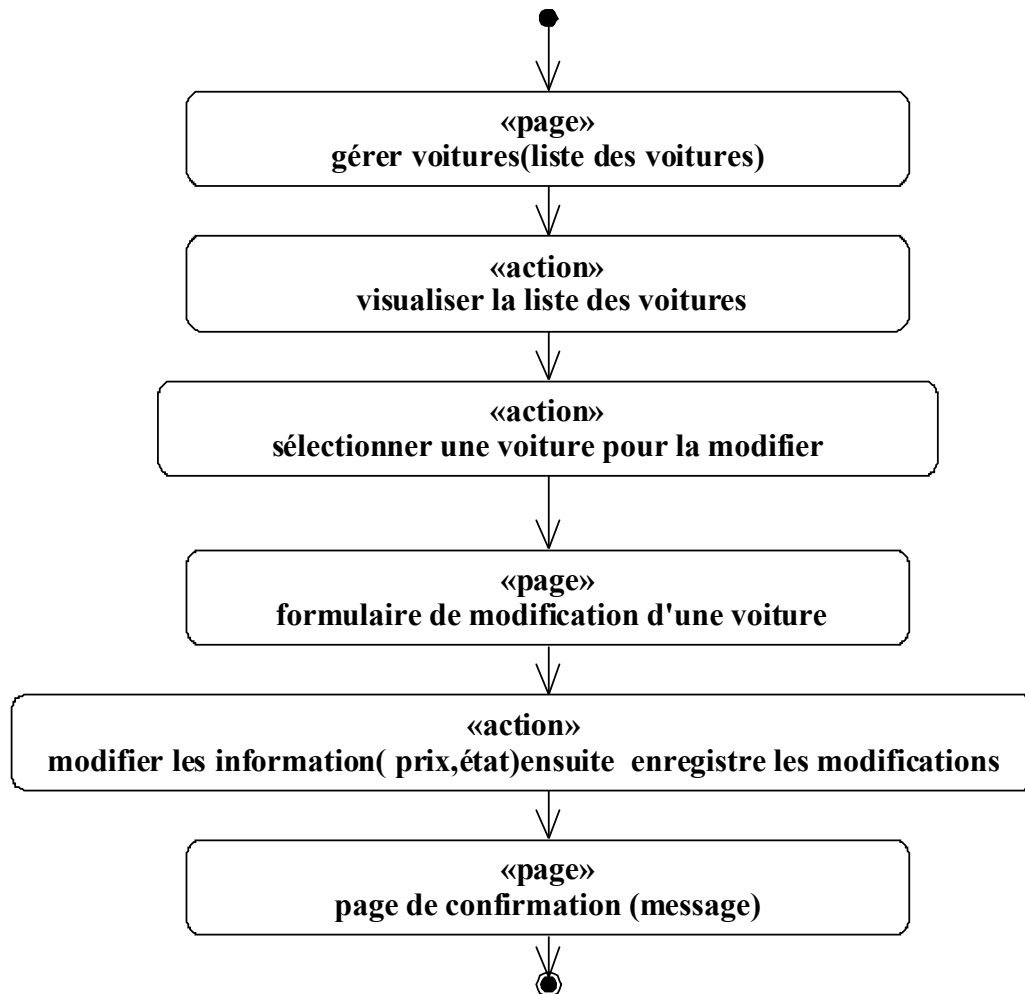


Figure 4.22 : diagramme d'activité « ajouter voiture »

4 Modifier voiture**Figure 4.23 : diagramme d'activité « modifier voiture »**

5 Supprimer la voiture

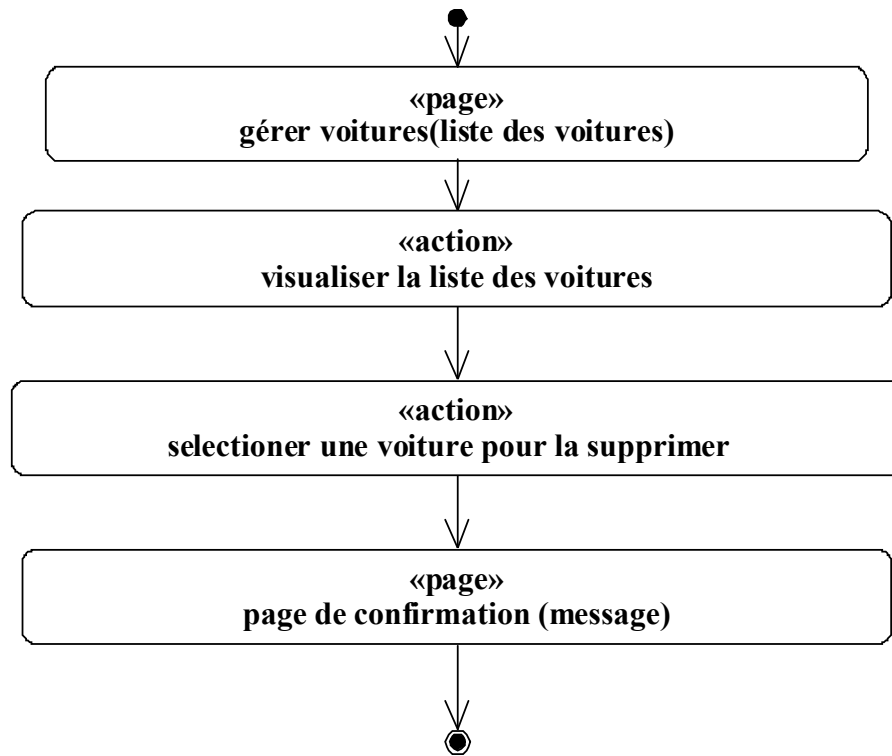


Figure 4.24 : diagramme d'activité «supprimer voiture»

4.4 Phase de conception

4.4.1 Les diagrammes d'interaction

4.4.1.1 Pour le client

1 consulter voitures

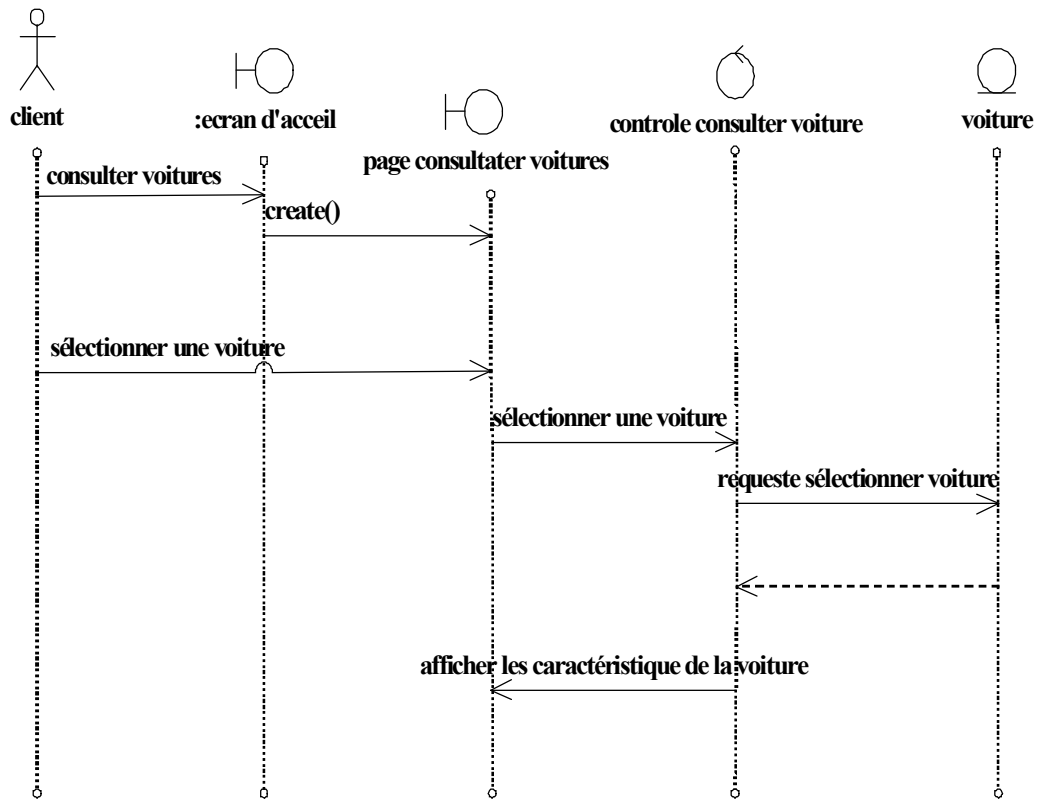


Figure 4.25 : diagramme d'interaction « consulter voitures »

2 réserver voiture

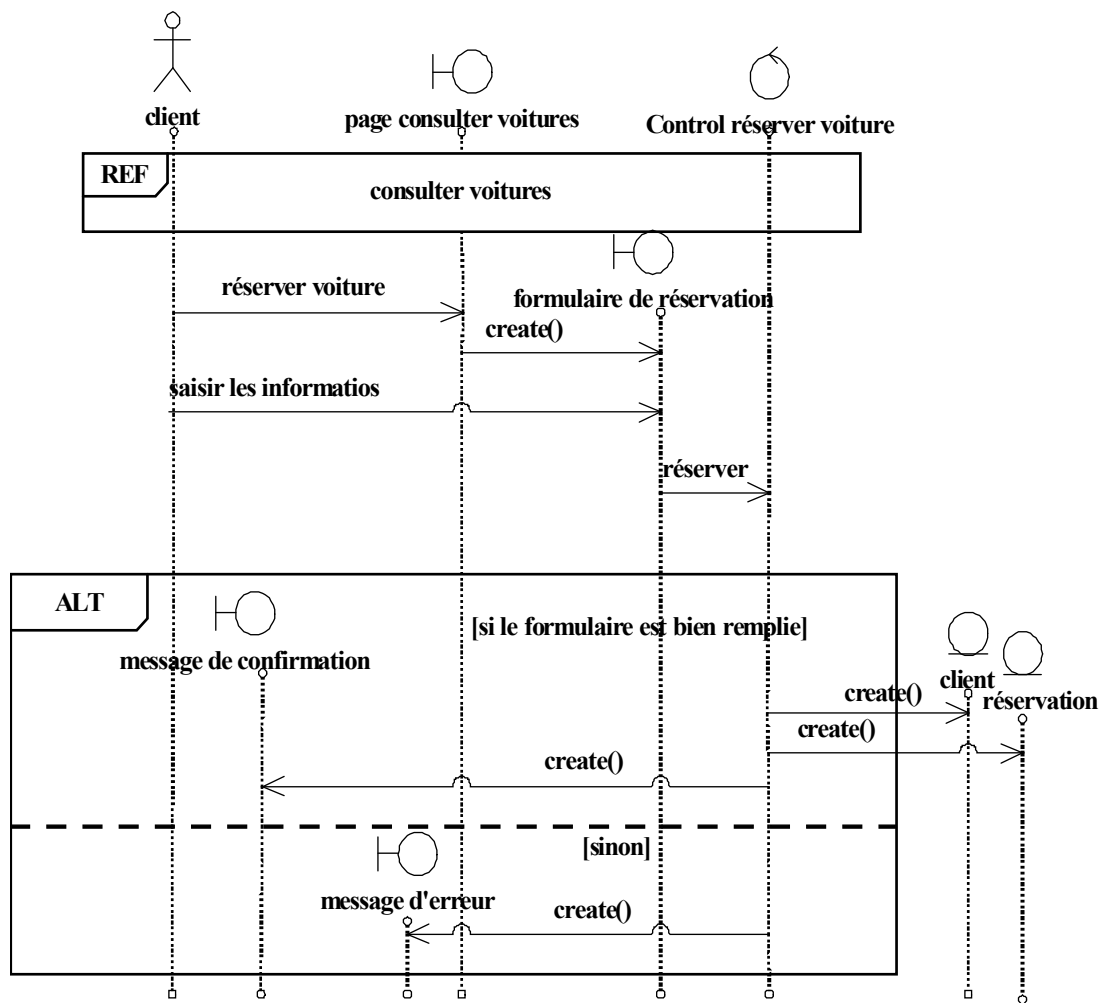


Figure 4.26 : diagramme d'interaction « réserver voiture »

4.4.1.2 Pour l'administrateur

1 s'authentifier

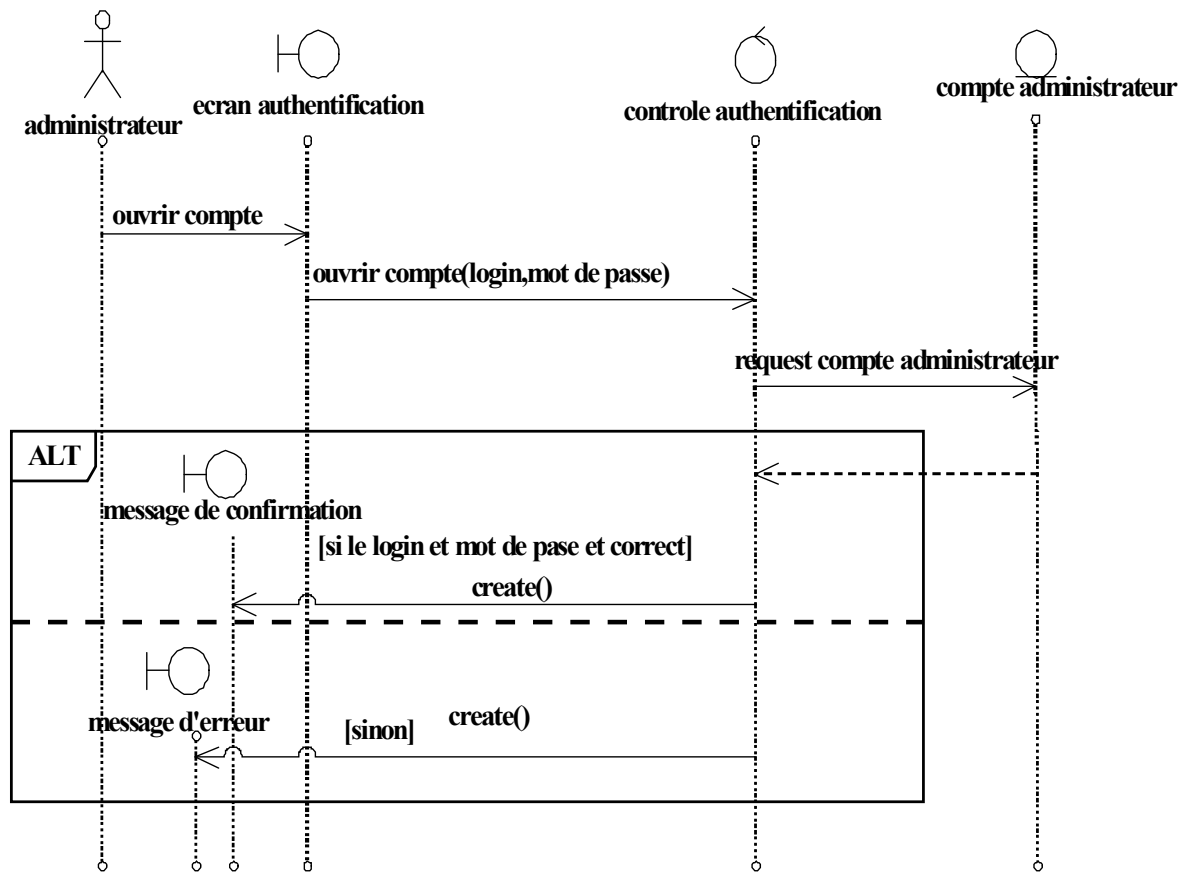


Figure 4.27 : diagramme d'interaction « s'authentifier »

2 Valider-Annuler réservation

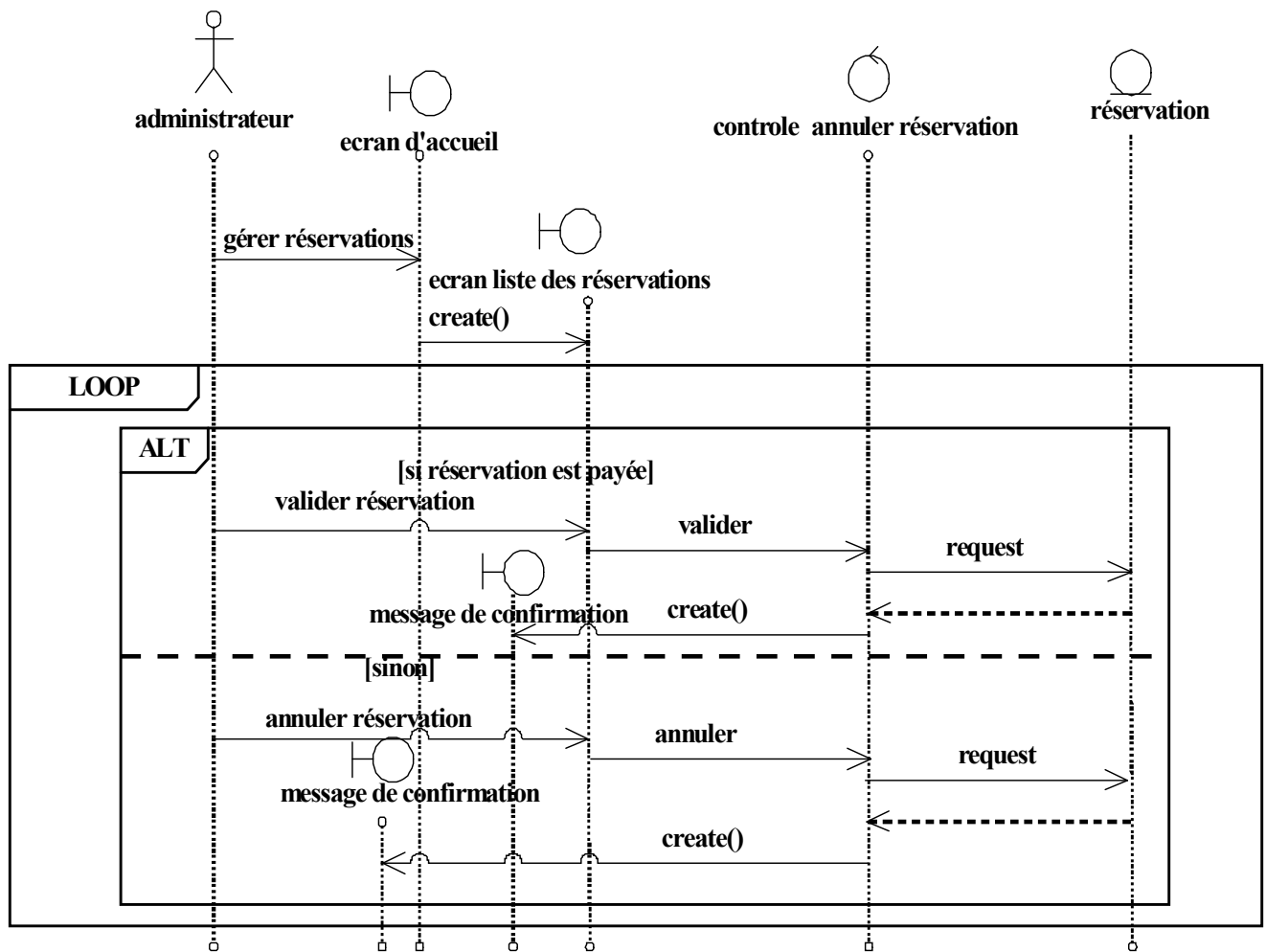


Figure 4.28 : diagramme d'interaction « Valider-Annuler réservation »

3 Ajouter voiture

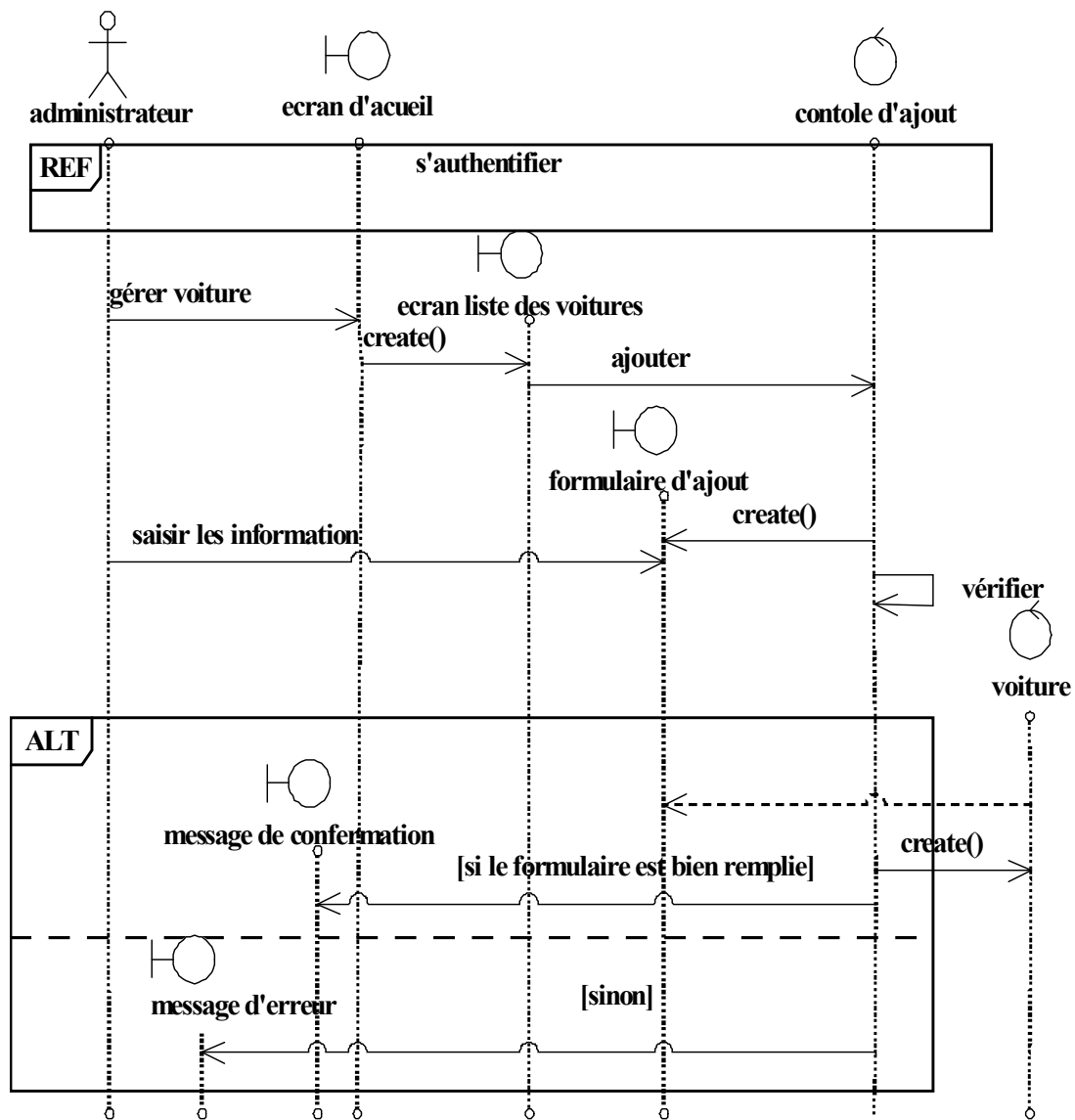


Figure 4.29 : diagramme d'interaction « ajouter voiture »

4 Modifier la voiture

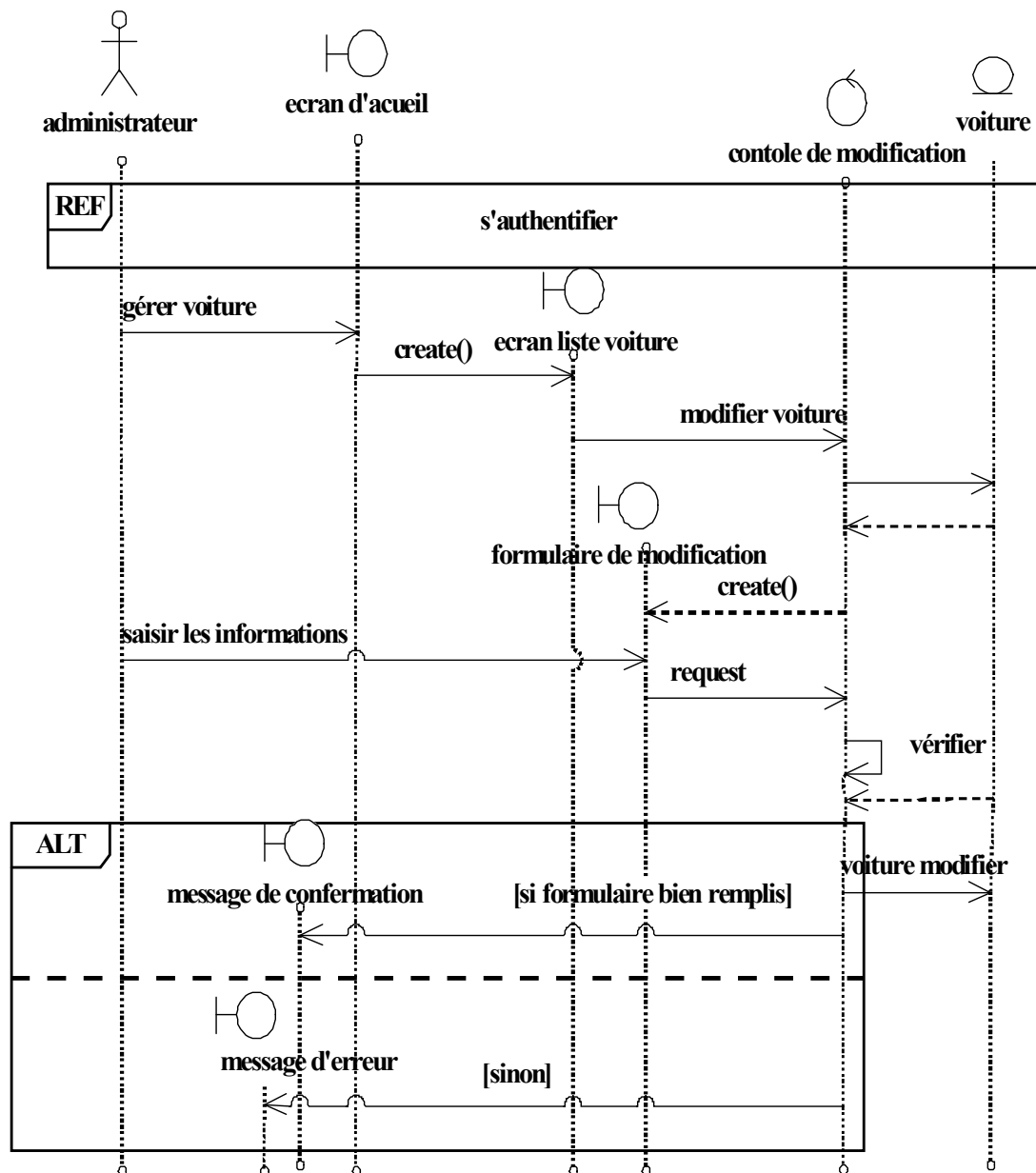


Figure 4.30 : diagramme d'interaction «modifier voiture»

5 Supprimer la voiture

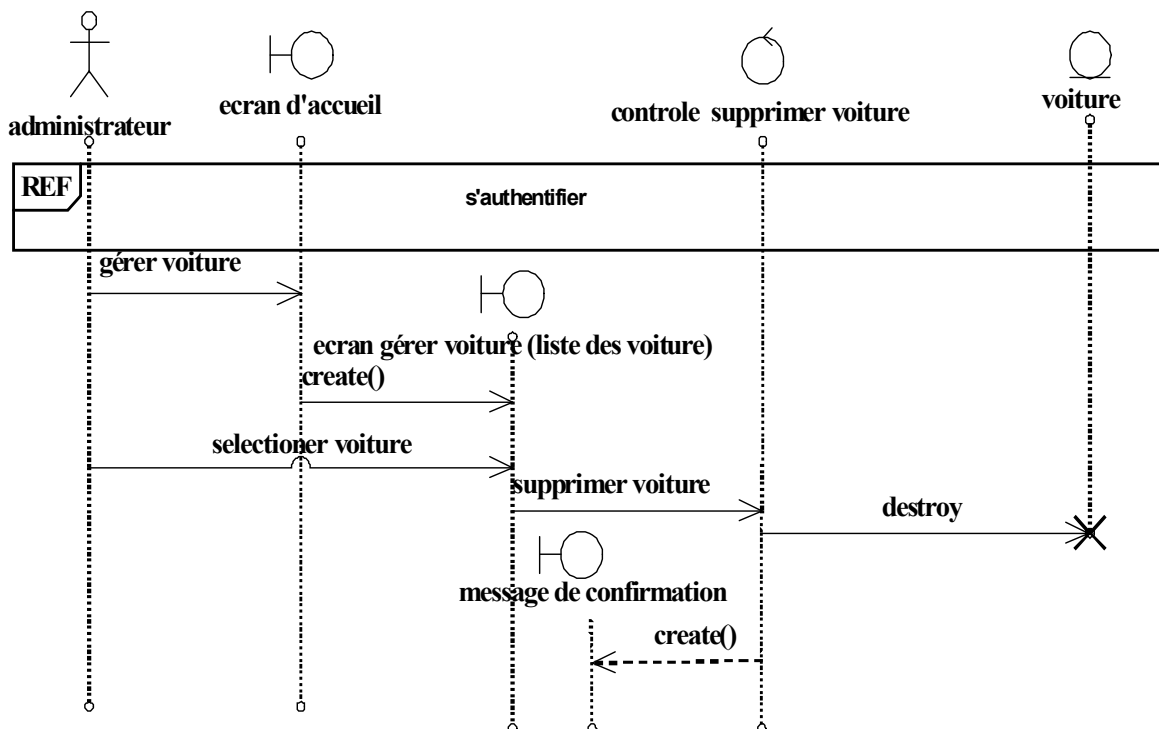


Figure 4.31 : diagramme d'interaction « supprimer voiture »

4.5 Conclusion

A travers les informations collectées, nous remarquons que notre projet demande une étude profonde avant d'entamer le travail, c'est pour ça nous avons fait une étude de cas détaillée pour préciser ce que le système devra faire.

Et dans le chapitre suivant nous allons passer à l'étape d'implémentation de notre agence en ligne. Nous présentons les outils et les langages de programmation utilisés pour l'implémentation et la réalisation de notre site web.

5.1 Introduction

L'objectif de ce chapitre est de montrer comment nous avons implémenté notre site. Nous présenterons d'abord les langages de programmation ainsi que les outils utilisés pour le développement de l'application. Par la suite nous allons présenter quelques pages de notre système.

5.2 Les langages de programmation et les outils de développement utilisé

5.2.1 Les langages de programmations

Dans notre réalisation de site web nous avons utilisé les langages de programmations suivants :

5.2.1.1 HTML

Nous avons utilisé l'HTML pour créer les pages statiques de notre site web. HTML est l'acronyme du terme anglais « HyperText markup language », il n'est pas un langage de programmation à proprement dit mais un ensemble de commandes permettant de formater des documents. Il est interprété comme des balises de la forme suivante :

`<A HREF="URL"> ceci est un lien </A>`

HTML a été développé au sein de la CERN à Genève afin de pouvoir diffuser par Internet des publications diverses et des documents multimédias.

5.2.1.2 Java Script

Est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype. Le langage a été créé en 1995 par Brendan Eich pour le compte de Netscape Corporation. La version 1.8.5 du langage est prévue pour intégrer la 5^e version du standard ECMA.

Le code JavaScript peut être intégré directement au sein des pages web, pour y être exécuté sur le poste client. C'est alors le navigateur Web qui prend en charge l'exécution de ces programmes appelés scripts. Généralement, JavaScript sert à contrôler les données saisies dans des formulaires HTML, ou à interagir avec le document HTML via l'interface Document Object Model,

Exemple :

```
<SCRIPT language="JavaScript">
```

```
.....
```

```
</SCRIPT>
```

5.2.1.3 CSS

Les feuilles de style en cascade de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML.

Exemple :

```
<STYLE type="text/css" >
```

```
a:link {
```

```
Color: blue
```

```
}
```

```
</STYLE>
```

Nous avons utilisé aussi les feuilles de style (CSS) pour :

- Mettre en forme notre site
- Centraliser l'apparence de notre site dans un seul fichier CSS.
- Créer les boutons et leurs propriétés.
- Positionner les éléments du code HTML.

5.2.1.4 PHP

PHP (Personnel Home Page, peut signifier aussi HyperText préprocesseur) est un langage de programmation web (langage de scripts). Il est interprété, par conséquent, il ne nécessite pas d'être compilé pour obtenir un objet, un exécutable avant d'être utilisable (comme en C par exemple).

Nous avons choisi le PHP comme un langage de programmation web dynamique parce qu'il est très riche et très utilisé et supporté par la majorité d'hébergeur. On a utilisé ce langage principalement pour la communication avec la base de données MYSQL. C'est le meilleur support pour le MYSQL.

5.2.1.5 MYSQL

Pour gérer notre base de données nous avons choisi le système de gestion de base de données MYSQL parce que il est libre (gratuit). Ainsi, MYSQL offre l'interface

PHPMYADMIN pour faire toutes les opérations (créé, modifier, supprimer les tables) en mode graphique. Il représente la meilleure combinaison avec le PHP.

5.2.2 Les outils de développement

5.2.2.1 EasyPHP

C'est une plateforme de développement Web, permettant de faire fonctionner localement des scripts PHP. EasyPHP (5.3.9) contient un serveur web Apache (2.2.21 VC9) et un serveur de bases de données MySQL (5.5.20), un interpréteur de script PHP (5.3.9), ainsi qu'une administration SQL phpMyAdmin.

5.2.2.2 PhpMyAdmin

PhpMyAdmin est une interface d'administration pour le SGBD MySQL. Il est écrit en langage PHP et s'appuie sur le serveur HTTP Apache.

Il permet d'administrer les éléments suivants :

- les bases de données.
- les tables et leurs champs (ajout, suppression, définition du type).
- les index, les clés primaires et étrangères.
- les utilisateurs de la base et leurs permissions.
- exporter les données dans divers formats (CSV, XML, PDF, Open Document, Word, Excel et Latex).

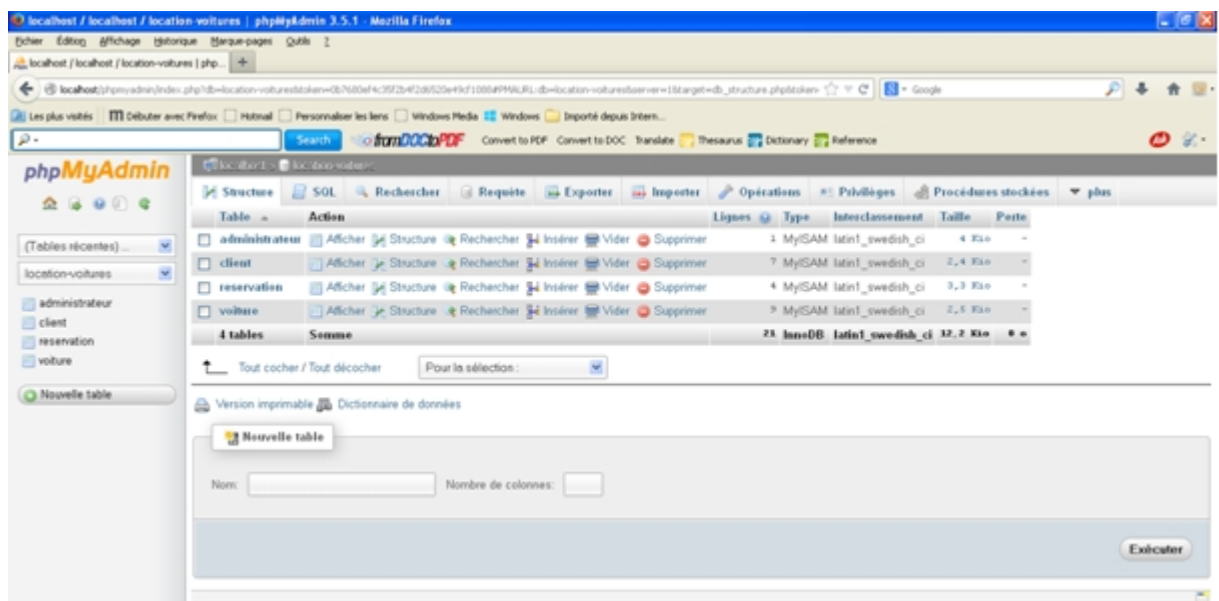


Figure 5.1: « l'interface de PHPMYADMIN »

5.2.2.3 Macromédia dreamweaver

Est un outil de création des pages web, cet éditeur a été créé par macromédia, aussi puissant que Frontpage, passées les premières appréhensions, il s'avérera un bon allié d'édition web grâce à son interface graphique.

5.3 Le passage du modèle du domaine au modèle relationnel

5.3.1 Les règles de passage du diagramme de classe au modèle relationnel

Pour traduire le modèle du domaine en un schéma relationnel équivalent, nous avons utilisé quatre règles (de R1 à R4) qui sont les plus simples et les plus opérationnelles :

- **Transformation des entités/ classes** : La règle est simple :

R1 : Chaque classe devient une relation, Les attributs de la classe deviennent des attributs de la relation. Si la classe possède un identifiant, il devient la clé primaire de la relation, sinon, il faut ajouter une clé primaire arbitraire.

- **Transformation des associations**

Les règles de transformation que nous allons voir dépendent des cardinalités/multiplicités maximales des associations. Nous distinguons trois familles d'association :

Association 1..1 : La règle est la suivante :

R2 : Pour représenter une association 1 vers 1 entre deux relations, la clé primaire de l'une des relations doit figurer comme clé étrangère dans l'autre relation.

Association 1..* : La règle est la suivante :

R3 : Pour représenter une association 1 vers plusieurs, on procède comme pour une association 1 vers 1, excepté que c'est forcément la relation du côté plusieurs qui reçoit comme clé étrangère la clé primaire de la relation du côté 1.

Association *.* : La règle est la suivante :

R4 : Pour représenter une association du type plusieurs vers plusieurs, il faut introduire une nouvelle relation dont les attributs sont les clés primaires des relations en association, et dont la clé primaire est la concaténation de ces deux attributs. Si l'association possède des attributs, ils deviennent des attributs de la relation correspondante.

5.3.2 Structure de la base de données

Après applications des quatre règles précédentes, nous avons obtenus les tables suivantes :

Client (id_client, nom, prénom, email, date_nais, adresse, num_tel, num_peris, date_peris)

Voiture (id_voiture, immatriculation, type, marque, puissance, places_assicess, prix, état)

Réservation (id_réservation, date_départ, date retour, état_réserv, # id_client, # id_voiture)

5.3.3 Implémentation de la base de données

Pour implémenter notre base de données, nous avons utilisé PHPMYADMIN.

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
☐ 1	login	varchar(150)	latin1_swedish_ci		Non	Aucune	
☐ 2	mot_passe	varchar(150)	latin1_swedish_ci		Non	Aucune	

Figure 5.2 : « table administrateur »

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
☐ 1	<u>id_client</u>	int(30)			Non	Aucune	AUTO_INCREMENT
☐ 2	nom	varchar(150)	latin1_swedish_ci		Non	Aucune	
☐ 3	prenom	varchar(150)	latin1_swedish_ci		Non	Aucune	
☐ 4	email	varchar(150)	latin1_swedish_ci		Non	Aucune	
☐ 5	date_nais	varchar(150)	latin1_swedish_ci		Non	Aucune	
☐ 6	adresse	varchar(150)	latin1_swedish_ci		Non	Aucune	
☐ 7	num_tel	varchar(150)	latin1_swedish_ci		Non	Aucune	
☐ 8	num_peris	varchar(30)	latin1_swedish_ci		Non	Aucune	
☐ 9	date_peris	date			Non	Aucune	

Figure 5.3 : « table client »

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
☐ 1	<u>id_voiture</u>	int(30)			Non	Aucune	AUTO_INCREMENT
☐ 2	immatriculation	varchar(150)	latin1_swedish_ci		Non	Aucune	
☐ 3	type	varchar(150)	latin1_swedish_ci		Non	Aucune	
☐ 4	marque	varchar(150)	latin1_swedish_ci		Non	Aucune	
☐ 5	puissance	int(30)			Non	Aucune	
☐ 6	places_assices	int(30)			Non	Aucune	
☐ 7	prix	double			Non	Aucune	
☐ 8	etat	varchar(150)	latin1_swedish_ci		Non	Aucune	
☐ 9	image	varchar(300)	latin1_swedish_ci		Non	Aucune	

Figure 5.4 : « table voiture »

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
☐ 1	id_reservation	int(30)			Non	Aucune	AUTO_INCREMENT
☐ 2	date_depart	date			Non	Aucune	
☐ 3	date_retour	date			Non	Aucune	
☐ 4	etat_reserv	varchar(150) latin1_swedish_ci			Non	Aucune	
☐ 5	id_client	int(11)			Non	Aucune	
☐ 6	id_voiture	int(11)			Non	Aucune	

Figure 5.5 : « table réservation »

5.4 Les interfaces du site web

Dans ce qui suit, nous allons présenter quelques interfaces de notre site de location de voitures.

5.4.1 Les interfaces du client

1. interface d'accueil



Figure 5.6 : « interface d'accueil »

2. interface de contact

Figure 5.7 : «interface de contact »

3. interface consulter voitures

Image	immatriculation:	type:	marque:	puissance:	places_assis:	price:	etat:	button
	15161-111-43	tourisme	peugeot308	8	4	4500	disponible	reserver
	15162-112-43	tourisme	chevrolet_cruze	8	5	4500	disponible	reserver
	15163-113-43	tourisme	clio4	6	4	3500	disponible	reserver
	15164-113-43	tourisme	citroen_c3	8	4	4000	disponible	reserver

Figure 5. 8 : « interface consulter voitures»

4. interface formulaire de réservation

Figure 5. 9 : « interface formulaire de réservation »

5. interface consulter réservations

date_depart	date_retour	etat_res	num_permis	immatriculation
2014-05-14	2014-05-14	refuser	26854279	15161-111-43
2014-05-06	2014-05-04	refuser	01257825	15164-113-43
2014-05-07	2014-05-14	accepter	26854279	15162-112-43

Copyright © 2014 Company Name MILEV AUTO LOCATION DE VOITURE

Figure 5.10: « interface consulter réservations »

5.4.2 les interfaces de l'administrateur

1. interface d'authentification



MILEV_AUTO

formulaire d'authentification

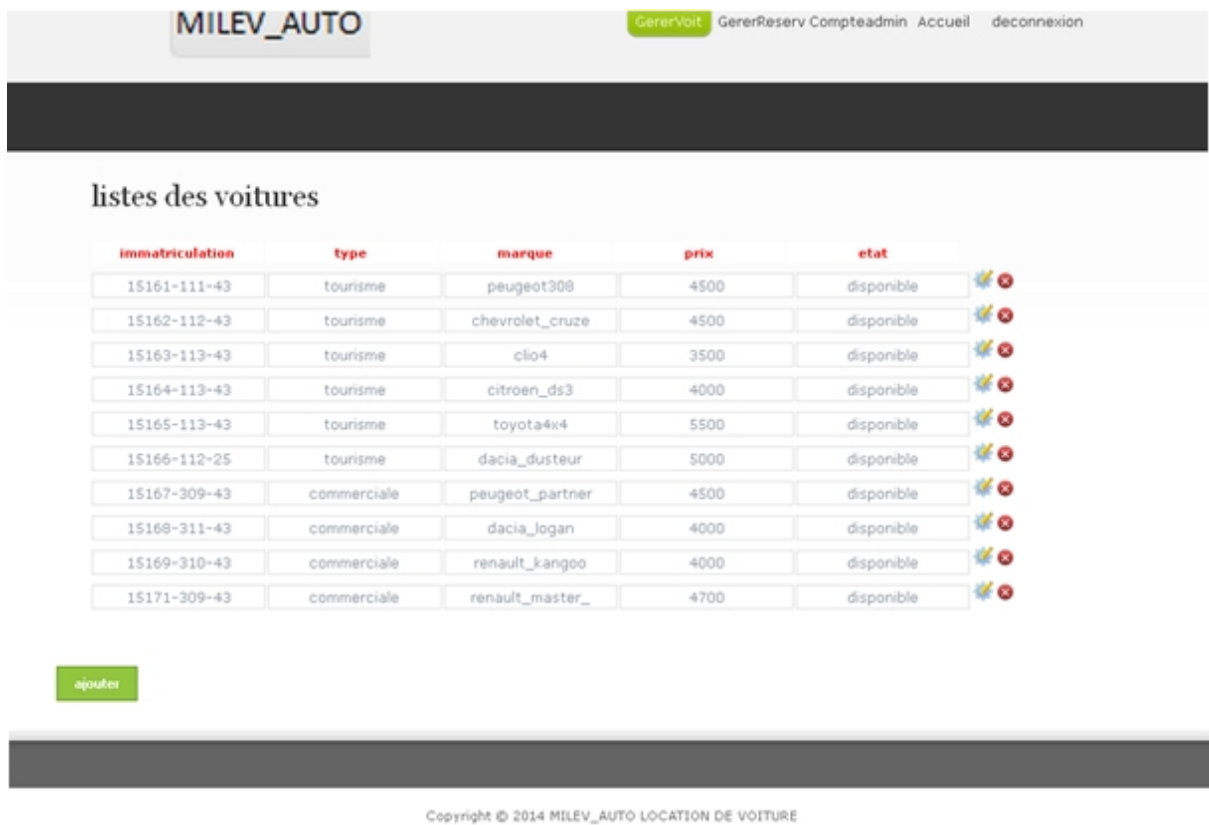
Nom d'administrateur

Mot de passe

Se connect

Figure 5.11 : « interface d'authentification »

2. interface gérer voitures



MILEV_AUTO

GérerVoit GererReserv Comptadmin Accueil deconnexion

listes des voitures

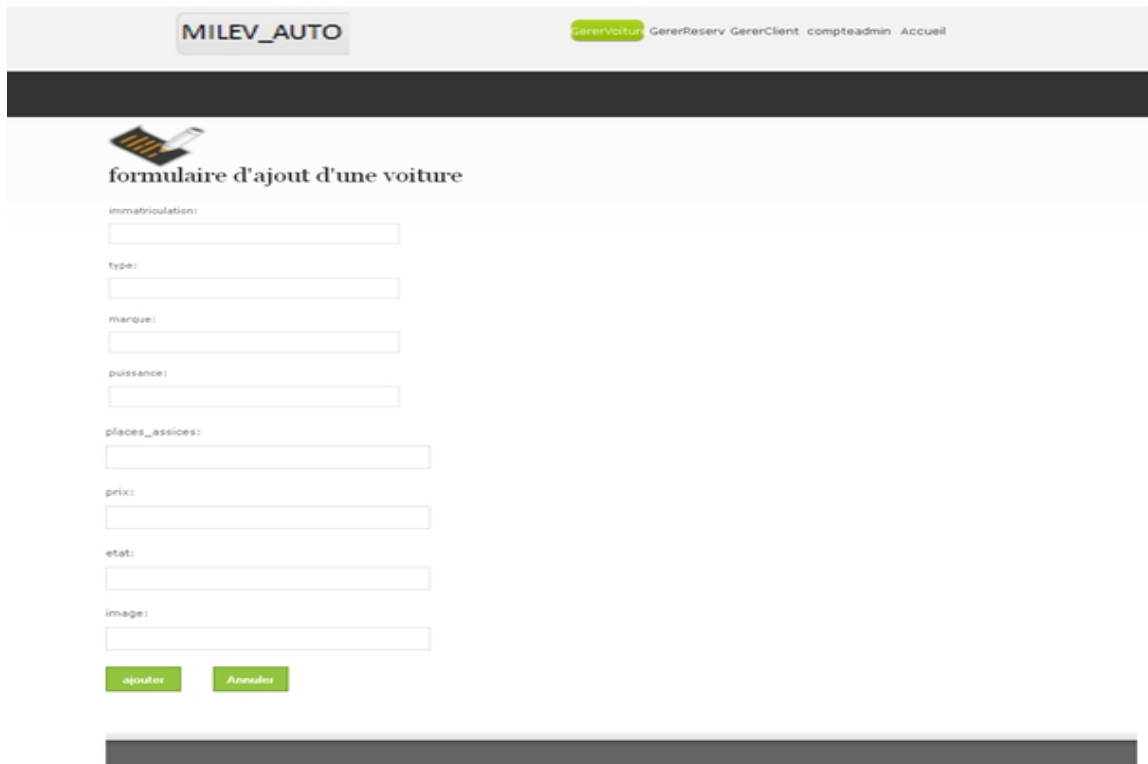
immatriculation	type	marque	prix	etat	
15161-111-43	tourisme	peugeot308	4500	disponible	 
15162-112-43	tourisme	chevrolet_cruze	4500	disponible	 
15163-113-43	tourisme	clio4	3500	disponible	 
15164-113-43	tourisme	citroen_ds3	4000	disponible	 
15165-113-43	tourisme	toyota4x4	5500	disponible	 
15166-112-25	tourisme	dacia_dusteur	5000	disponible	 
15167-309-43	commerciale	peugeot_partner	4500	disponible	 
15168-311-43	commerciale	dacia_logan	4000	disponible	 
15169-310-43	commerciale	renault_kangoo	4000	disponible	 
15171-309-43	commerciale	renault_master_	4700	disponible	 

ajouter

Copyright © 2014 MILEV_AUTO LOCATION DE VOITURE

Figure 5.12 : « interface gérer voitures »

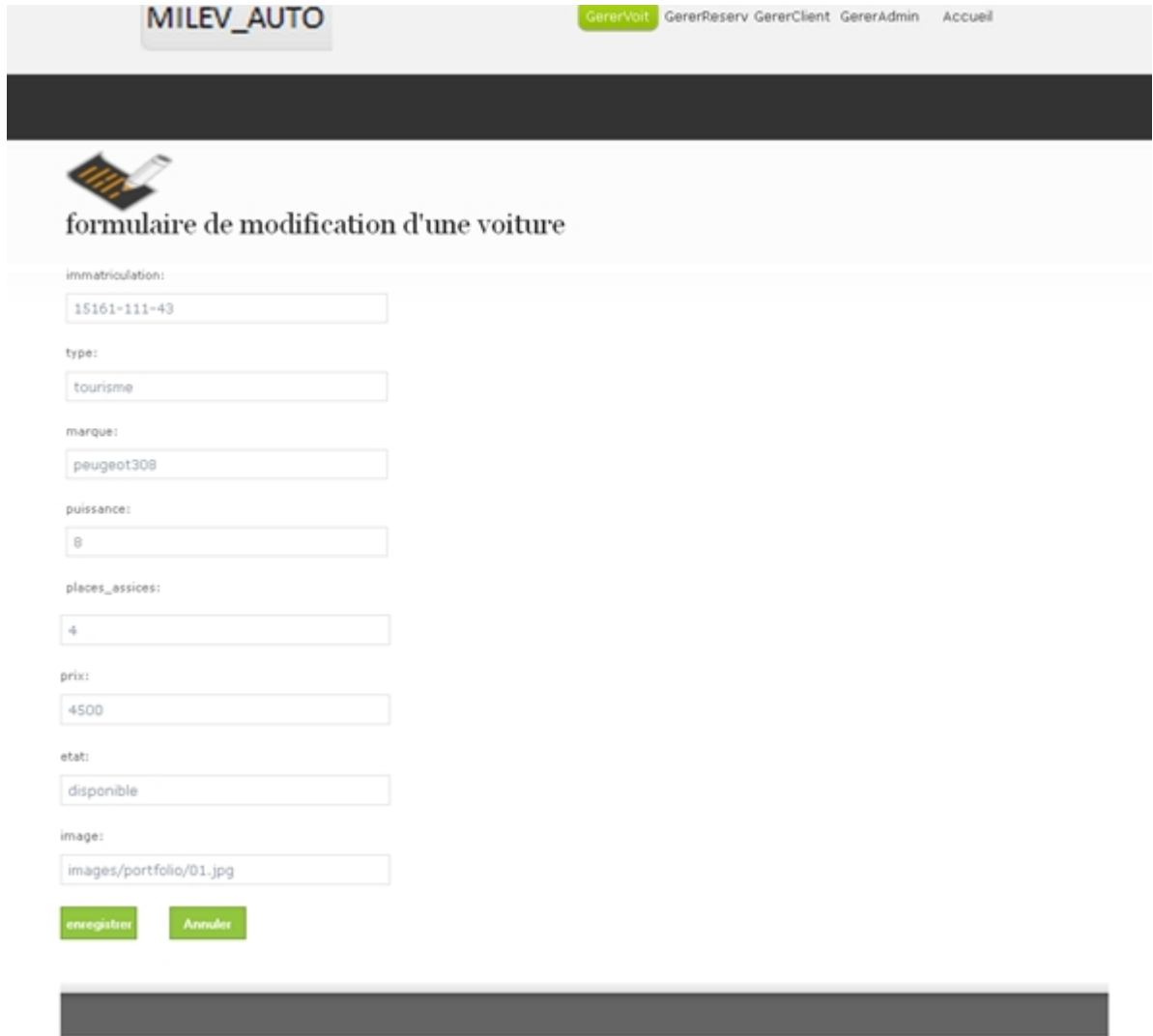
3. interface Formulaire d'ajout d'une voiture



The screenshot displays the 'MILEV_AUTO' web application interface. At the top, there is a navigation bar with the logo 'MILEV_AUTO' and a menu containing 'GérerVoiture', 'GérerReserv', 'GérerClient', 'compteadmin', and 'Accueil'. Below the navigation bar, the main content area features a header with a car icon and the title 'formulaire d'ajout d'une voiture'. The form itself consists of several labeled input fields: 'immatriculation:', 'type:', 'marque:', 'puissance:', 'places_assises:', 'prix:', 'etat:', and 'image:'. At the bottom of the form, there are two buttons: 'ajouter' (in green) and 'Annuler' (in green). The entire interface is set against a light gray background with a dark gray footer bar.

Figure 5. 13 : « interface formulaire d'ajout d'une voiture »

4. interface Formulaire de modification d'une voiture



MILEV_AUTO [GererVoit](#) [GererReserv](#) [GererClient](#) [GererAdmin](#) [Accueil](#)

 **formulaire de modification d'une voiture**

immatriculation:

type:

marque:

puissance:

places_assises:

prix:

etat:

image:

[enregistrer](#) [Annuler](#)

Figure 5. 14 : « interface formulaire de modification d'une voiture »

5. interface valider-annuler réservations

The screenshot shows the MILEV_AUTO web application interface. At the top, there is a header with the logo 'MILEV_AUTO' and navigation links: 'GererVoit', 'GererReserv' (highlighted in green), 'Compteadmin', 'deconnexion', and 'Accueil'. Below the header, the main content area is titled 'liste des réservations'. It contains a table with the following columns: 'date_depart', 'date_retour', 'etat_reserv', 'num_peris', and 'immatriculation'. The table lists three reservations. To the right of each row, there are two circular icons: a green checkmark and a red minus sign. At the bottom of the page, there is a footer with the text 'Copyright © 2014 MILEV_AUTO LOCATION DE VOITURE'.

date_depart	date_retour	etat_reserv	num_peris	immatriculation
2014-05-14	2014-05-14	refuser	26854279	15161-111-43
2014-05-06	2014-05-04	refuser	01257825	15164-113-43
2014-05-07	2014-05-14	accepter	26854279	15162-112-43

Figure 5. 15 : « interface valider-annuler réservations»

5.5 Conclusion

Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement de développement, l'implémentation de la base des données et la démarche suivie pour la réalisation

À la fin de ce travail, nous avons réussi à développer un site web qui gère pratiquement toutes les activités principales d'une agence de location de voitures.

CONCLUSION GENERALE

1 Bilan

A présent, la location des voitures en Algérie est un projet très réalisable. Afin de répondre au besoin des clients et de sociétés algériennes et d'avoir un moyen efficace et accessible sur Internet 24 sur 24 heures, nous avons développé dans le cadre de notre projet un site web dynamique qui offre les principales fonctionnalités nécessaires pour réaliser des réservations en ligne facilement.

Dans notre application nous avons utilisé UML pour modéliser le site, à l'aide des scriptes **PHP** en intégrant des commandes **SQL**. Ce dernier constitue actuellement le langage de requêtes standard pour les bases de données.

Le site web développé est un site Web dynamique pour la location en ligne des voitures. Le client a la capacité de consulter un catalogue des voitures en ligne et d'effectuer une réservation sur l'offre qui satisfait ses besoins et sans déplacement.

Notre site offre toutes les fonctionnalités d'un site Web de location de voitures et reste ouverte à toute évolution, En effet c'est une expérience qui nous a permis des connaissances dans de domaines très variés comme: L'Orienté Objet, UML, le langage HTML...etc. Elle permet aussi de découvrir un domaine en pleine évolution qui est le nouveau cœur de toutes activités dans les sociétés.

2 les perspectives

Malgré les efforts fournis pour la réalisation du ce projet, ce travail nous n'avons pas pu atteindre tous les objectifs fixés.

Pour finir notre projet nous prévoyons comme perspectives :

- Collaborer avec des sociétés nationales et étrangères.
- Se transport à l'endroit demandé.
- La location des voitures pour cortège nuptial.
- La location de court et langue durée.
- La location avec chauffeur (conducteur).

REFERENCES BIBLIOGRAPHIQUES

- [1] www.wikipedia.com
- [2] www.siteduzero.com
- [3] www.fr.wikipedia.org
- [4] www.uml-sysnl.org
- [5] Laurent-peichoki developper.com/totoriel
- [6] Uml.free.com
- [7] Laurent-audibert.developpez.com , cour uml en ligne.