



INSTITUT NATIONAL DES POSTES ET TÉLÉCOMMUNICATION

RAPPORT DU PROJET JAVAEE

Gestion d'une agence de location de voiture

Réalisé par :

TAMIR taha
SLIMANE Mouad
OUSSIDI Adil
ALALOU Ayoub

Encadré par :

M.EL HAMLAOUI Mahmoud

Filière : DATA ENGINEER

Année universitaire : 2020-2021

Remerciements

Tout d'abord, nous tenons à remercier toutes celles et tous ceux qui nous ont aidé à réaliser ce travail.

Nous voudrions présenter nos remerciements à notre encadreur « M.EL HAMLAOUI Mahmoud ». Nous voudrions également lui témoigner notre gratitude pour sa patience et son soutien qui nous a été précieux afin de mener notre travail à bon port. Merci

Enfin, nos remerciements s'adressent aux professeurs et au corps administratif de l'INPT. Et nous espérons que le présent écrit présentera notre travail décemment et que notre étude sera à la hauteur des espérances.

Résumé

Dans le cadre de notre projet JavaEE, que nous allons détailler par la suite est destiné pour la gestion de location de voitures. On rentre dans le cadre de la réalisation d'une Application web qui assure le bon fonctionnement d'une société de location de voiture.

Il vise à assurer toutes les gestions de la société : gestion des réservations, gestion de location, gestion des voitures et la gestion des clients sans oublier le coté statistiques sur l'ensemble des locations effectuées afin d'assurer la bonne décision.

Ce présent rapport, résumera le déroulement de toutes les étapes du projet de la conception jusqu'à la réalisation.

Abstract

As part of our JavaEE project, which we will detail below is intended for car rental management. This is part of the creation of a Web Application that ensures the proper functioning of a car rental company.

It aims to ensure all the management of the company : management of reservations, rental management, car management and customer management without forgetting the statistical side of all rentals made in order to ensure the right decision.

This report will summarize the progress of all stages of the project from concept to completion.

Table des matières

Remerciements	2
Résumé	3
Abstract	4
Introduction	7
1 Analyse et Conception	8
Analyse	8
Conception	8
A. Présentation de la méthode d'analyse	8
B. Diagramme de cas d'utilisation	8
C. Diagramme de séquence	9
D. Diagramme d'états	11
2 Réalisation	12
A. Environnement du travail	12
B. Les interfaces graphiques	15

Table des figures

1.1	Diagramme de cas d'utilisation	9
1.2	Diagramme de séquence pour ajouter une voiture	10
1.3	Diagramme de séquence pour ajouter une voiture	10
1.4	Diagramme d'états	11
2.1	Logo Java	12
2.2	Logo JavaEE	13
2.3	Logo Eclipse	13
2.4	Logo Apache Tomcat	13
2.5	Modèle-Vue-Contrôleur	14
2.6	Logo MySQL	15
2.7	Authentification d'admin	15
2.8	L'accueil d'administrateur	16
2.9	L'ajout d'une voiture	16
2.10	La liste des réservations	17
2.11	L'accueil d'utilisateur	17
2.12	Formulaire de la réservation	18

Introduction

Dans le cadre de notre formation(Cours+TP) en JavaEE à l'INPT, ce projet nous a été proposé dans le but de la réalisation d'une application web qui permet gérer les voitures (ajout, suppression, modification) et ses prix de location, gérer les locations, ainsi la gestion des Réservation (suppression, validation, confirmation).

Le présent rapport trace les différentes phases qui ont conduit à l'aboutissement de ce projet, et qu'on peut axer sur deux différents chapitres cohérents et complémentaires. Le premier concerne la conception du projet. Tandis que le deuxième chapitre est consacré à la réalisation du projet.

Chapitre 1

Analyse et Conception

Analyse

Le présent projet tend à mettre en place le service 'Gestion de location de voiture. suivant l' exemple que notre professeur encadrant a proposé.

Conception

A. Présentation de la méthode d'analyse

Le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML), est un langage (semi-formel) graphique pour visualiser, spécifier, construire et documenter un système logiciel. Il fusionne des notations provenant de plusieurs méthodes de développement orienté objet.

B. Diagramme de cas d'utilisation

L'objectif fondamental de cette étape est d'identifier les principaux cas d'utilisation. Dans notre application, nous avons défini deux acteurs :

— **Utilisateur :**

- * Prolonger la durée de location ;
- * Louer une voiture ;
- * Chercher une voiture.

— **Chef du projet :**

- * Se connecter au site Web ;
- * Chercher une voiture ;
- * Gérer la location des voitures ;
- * Voir la liste des réservations ;
- * Annuler une réservation.

Ceci est modélisé par le diagramme de cas d'utilisation suivant :

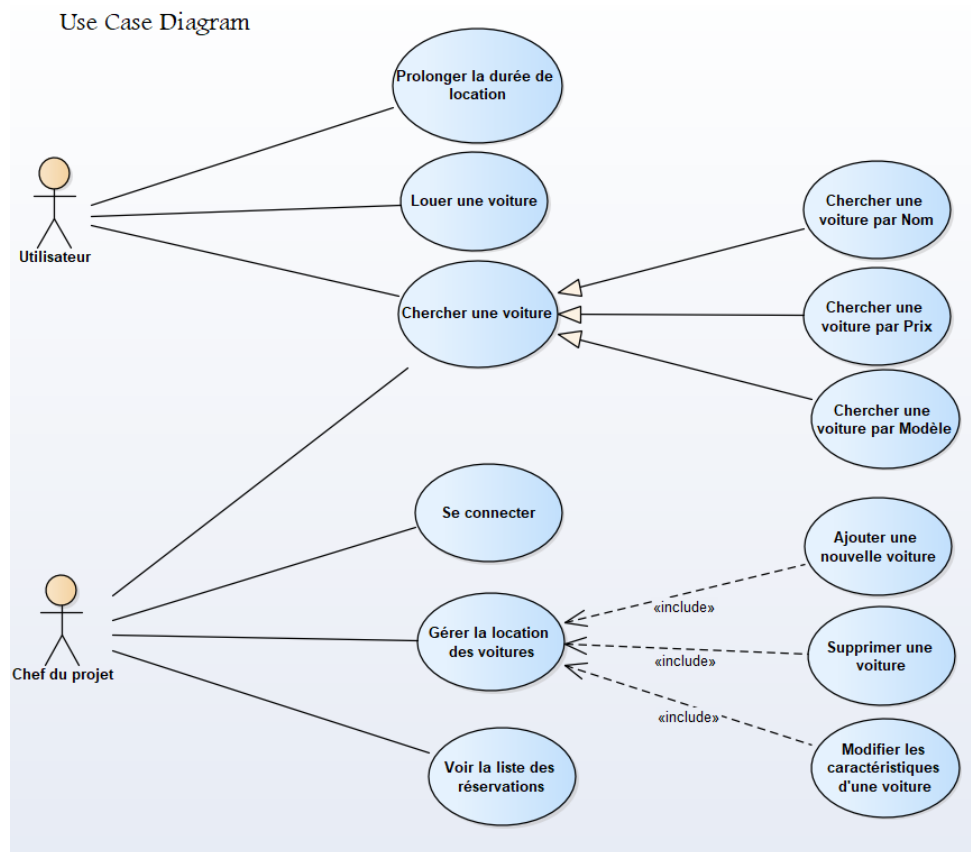


FIGURE 1.1 – Diagramme de cas d'utilisation

C. Diagramme de séquence

Le diagramme de séquence est une représentation intuitive qu'on a utilisée pour concrétiser des interactions entre nos entités.

Le diagramme de séquence décrit un scénario qui modélise une exécution particulière d'un cas d'utilisation du début jusqu'à la fin. Il correspond à une sélection d'enchaînements du cas d'utilisation.

a) Ajouter une voiture :

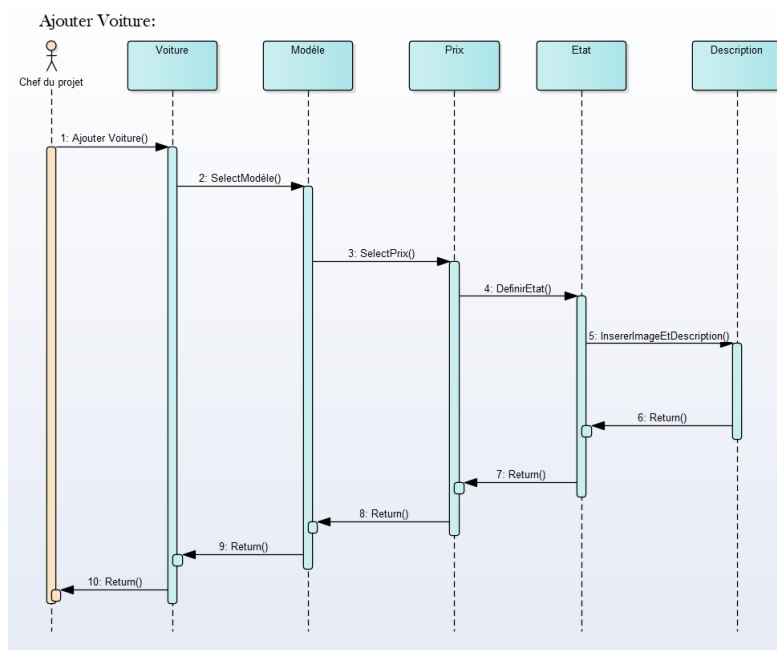


FIGURE 1.2 – Diagramme de séquence pour ajouter une voiture

b) Louer une voiture :

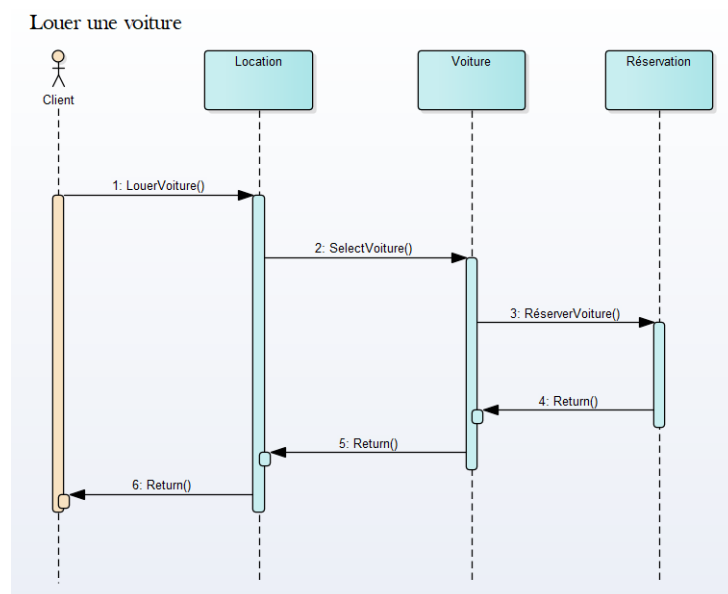


FIGURE 1.3 – Diagramme de séquence pour ajouter une voiture

D. Diagramme d'états

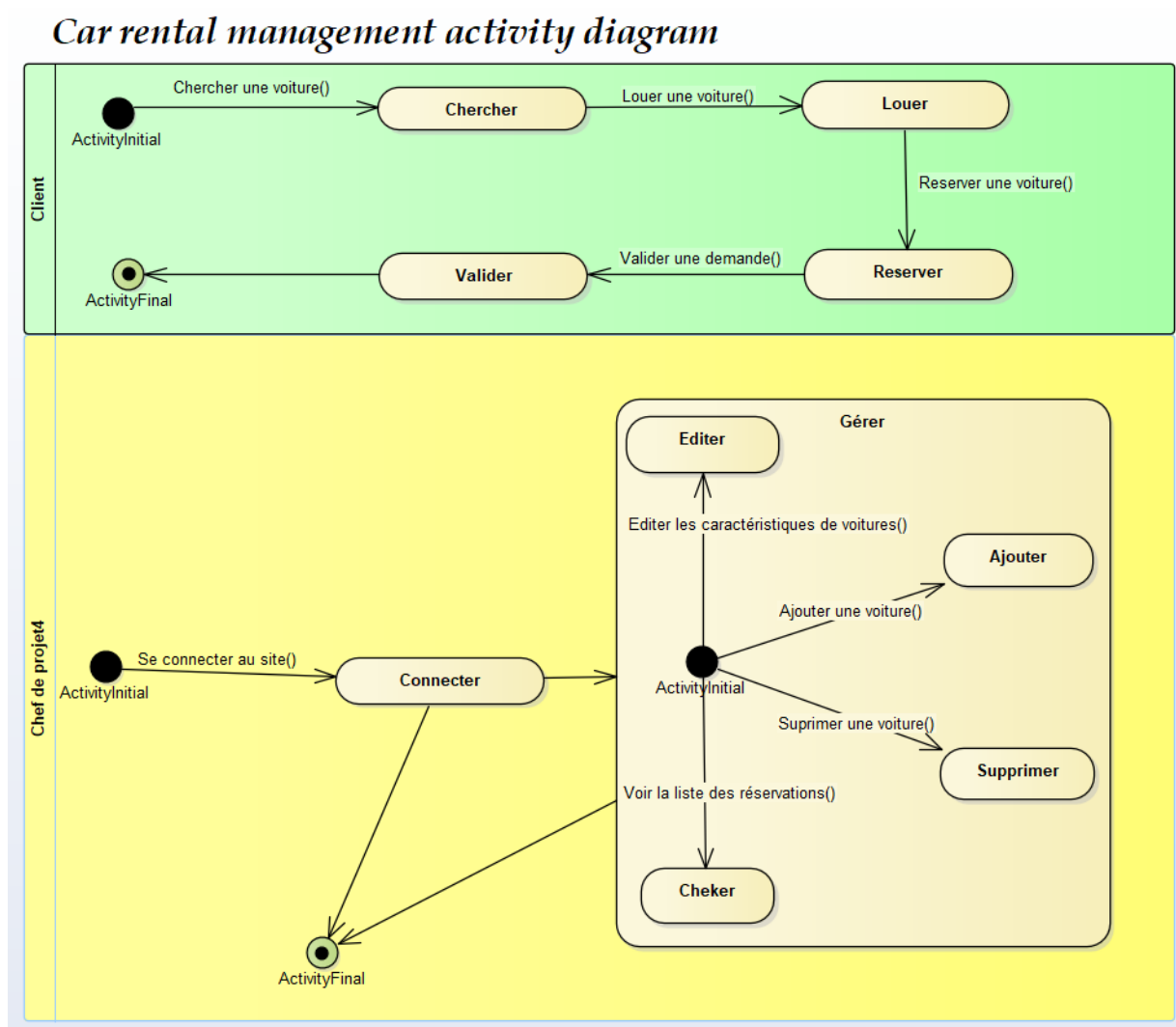


FIGURE 1.4 – Diagramme d'états

Chapitre 2

Réalisation

Après avoir élaboré la conception de notre application, nous abordons tout d'abord une étude technique où nous décrivons les ressources logicielles utilisées dans le développement de notre projet. Nous présentons en premier lieu notre choix de l'environnement de travail, où nous spécifions l'environnement matériel et logiciel qu'on a utilisé pour réaliser notre application puis nous détaillons l'architecture, aussi nous présentons quelques interfaces réalisées pour illustrer le fonctionnement de quelques activités du système.

A. Environnement du travail

a) Langage de programmation :



FIGURE 2.1 – Logo Java

Java est le nom de marque d'une technique informatique développée initialement par Sun Microsystems puis par Oracle : la «technologie Java™ ». Java est utilisé dans une grande variété de plates-formes depuis les systèmes embarqués et les téléphones mobiles, les ordinateurs individuels, les serveurs, les applications d'entreprise, les superordinateurs, etc. Défini à l'origine comme un langage, « Java » a évolué au cours du temps pour devenir un ensemble cohérent d'éléments techniques et non techniques. Ainsi, la technologie Java regroupe : Des standards (la plate-forme Java) définis sous forme de spécification par le Java Community Process (JCP), en trois éditions : Java SE (standard edition),



FIGURE 2.2 – Logo JavaEE

Java EE (enterprise edition) s'appuyant sur Java SE, est une plateforme normalisée, destinée au développement de logiciels pour les entreprises et les serveurs d'application. La plateforme comporte toutes les interfaces de programmation de Java SE, ainsi que des interfaces de programmation destinées à créer des applications Web, envoyer des courriels, mettre en oeuvre et utiliser des services Web, mettre en oeuvre de la persistance et des transactions.

b) Outils de développement :



FIGURE 2.3 – Logo Eclipse

Eclipse est l'environnement de développement (spécialisé pour le langage Java) qui sera utilisé dans ce projet. Le choix d'Eclipse repose essentiellement sur sa gratuité, sa facilité d'utilisation, sa puissance de développement et surtout ses nombreux plugins (bibliothèques additives).

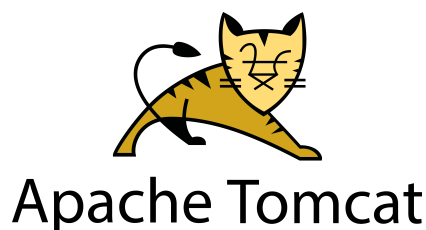


FIGURE 2.4 – Logo Apache Tomcat

L'utilisation d'un serveur Java EE est obligatoire pour le développement de pages Web dynamiques en Java EE. Un serveur HTTP classique reçoit des requêtes HTTP et renvoie des réponses mais il ne connaît pas les Servlets, les JSP... Il est donc essentiel d'utiliser un programme appelé moteur de Servlets qui est contenu dans le serveur Java EE et qui permet de pallier ce manque.

c) **La méthode MVC :**

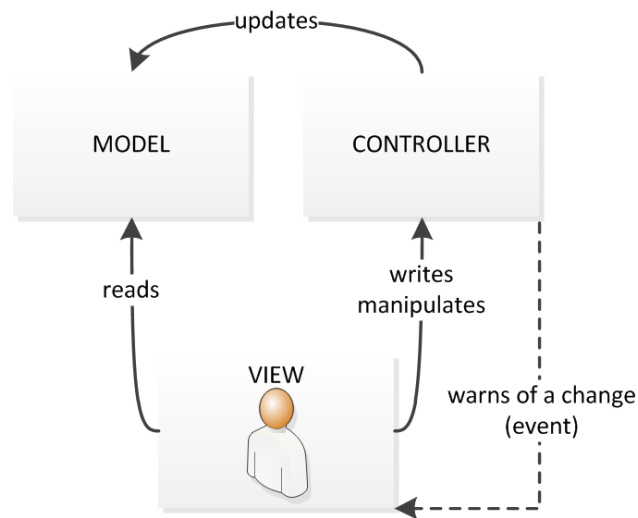


FIGURE 2.5 – Modèle-Vue-Contrôleur

Un des plus célèbres design patterns s'appelle MVC, qui signifie Modèle - Vue-Contrôleur. C'est celui que nous allons découvrir dans cette section du chapitre. Le pattern MVC permet de bien organiser son code source. Il aide à savoir quels fichiers créer, mais surtout à définir leur rôle. Son but est de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts, comme l'explique la description qui suit :

Le modèle représente la partie de l'application qui exécute la logique métier. Cela signifie qu'elle est responsable de récupérer les données, de les convertir selon des concepts chargés de sens pour votre plateforme, tels que le traitement, la validation, l'association et beaucoup d'autres tâches concernant la manipulation des données.

La vue se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples.

Le contrôleur gère les requêtes des utilisateurs. Elle est responsable de retourner une réponse avec l'aide mutuelle des couches Model et Vue. Les Contrôleurs peuvent être imaginés comme des managers qui ont pour mission que toutes les ressources souhaitées pour accomplir une tâche soient déléguées aux travailleurs corrects. Il attend des requêtes des clients, vérifie leur validité selon l'authentification et les règles d'autorisation, délègue les données récupérées et traitées par le Model, et sélectionne les type de présentation correctes que le client accepte, pour finalement déléguer le processus d'affichage à la couche Vue.

d) Gestion de la Base de données



FIGURE 2.6 – Logo MySQL

Le SGBD MySQL est supporté par un large éventail d'outils. MySQL est surtout installé pour les applications Web, ce SGBD est solide et utilisé par de grands groupes spécialisés dans l'Internet. Plusieurs pilotes natifs de type 4 sont disponibles pour MySQL et sont conseillés pour une utilisation en Java.

B. Les interfaces graphiques

Nous présentons dans cette section quelques interfaces principales de notre réalisation qui illustrent les différents cas d'utilisation.

a) Espace chef du projet :

— Accueil :

* *Page d'authentification*

Après la saisie du login et du password correcte une vérification se fait au niveau de la table de chef du projet qui mènera vers sa page d'accueil.

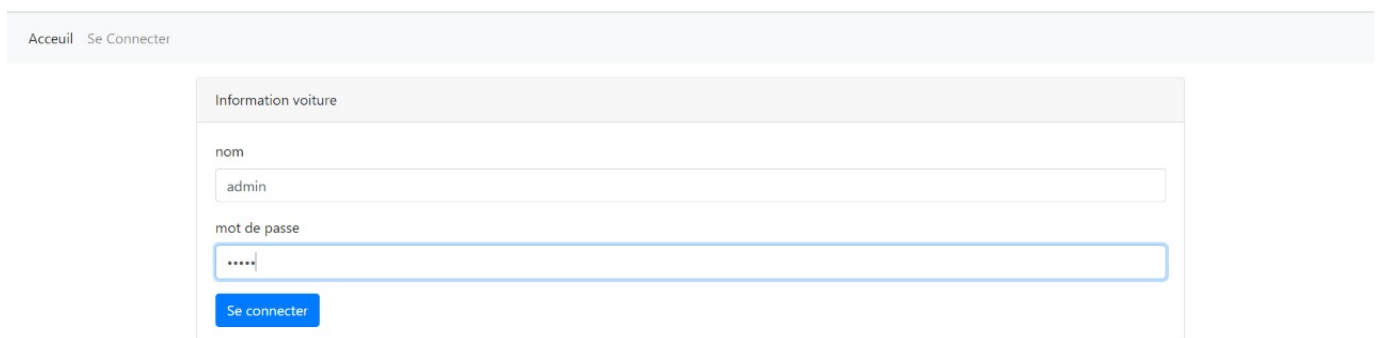
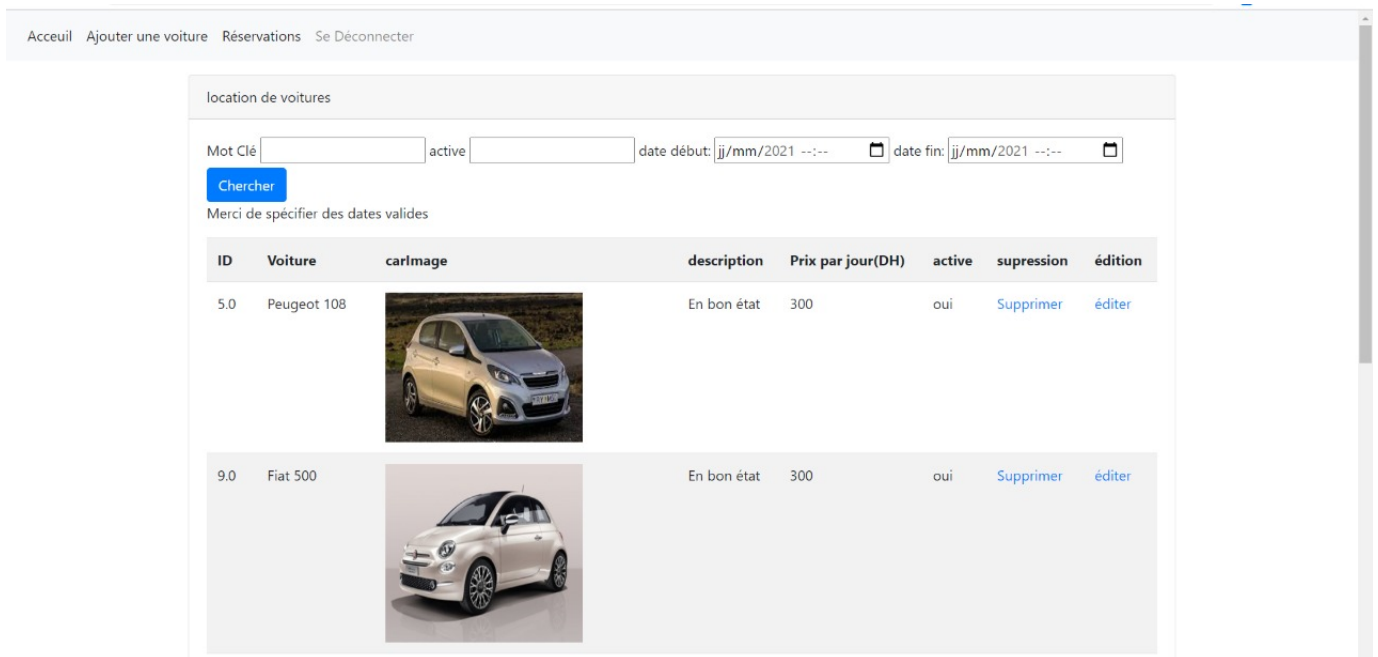


FIGURE 2.7 – Authentification d'admin

** Page d'accueil*

Le chef de projet peut consulter la liste des réservations, ajouter/ supprimer/ editer une voiture :



The screenshot shows the administrator dashboard. At the top, there is a navigation bar with links: Accueil, Ajouter une voiture, Réservations, and Se Déconnecter. Below this, there is a section titled "location de voitures". It contains a search bar with fields for "Mot Clé", "active", "date début" (with a calendar icon), and "date fin" (with a calendar icon). A blue "Chercher" button is next to the search fields. Below the search bar, there is a message: "Merci de spécifier des dates valides". Below this, there is a table with the following columns: ID, Voiture, carImage, description, Prix par jour(DH), active, suppression, and édition. The table contains two rows of data:



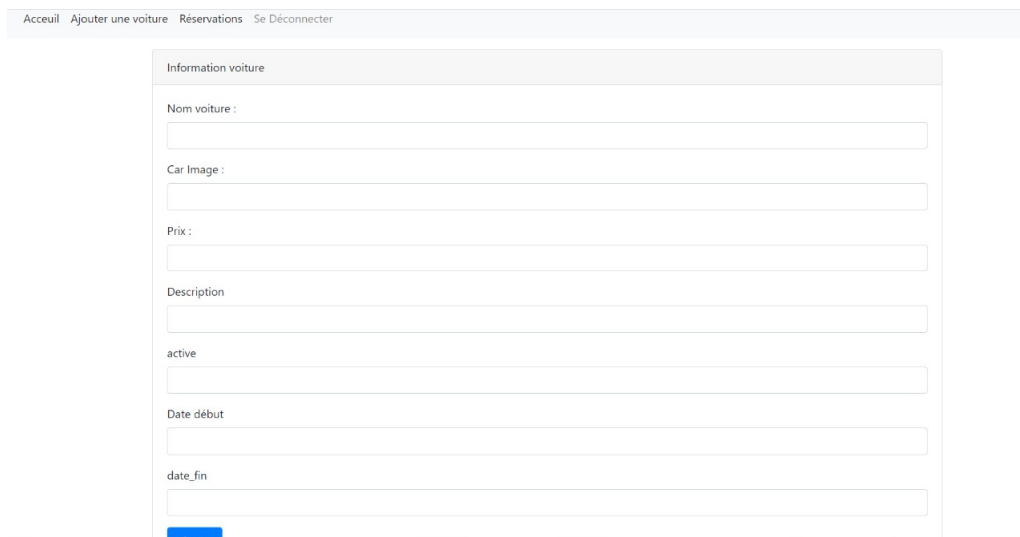
ID	Voiture	carImage	description	Prix par jour(DH)	active	suppression	édition
5.0	Peugeot 108		En bon état	300	oui	Supprimer	éditer
9.0	Fiat 500		En bon état	300	oui	Supprimer	éditer

FIGURE 2.8 – L'accueil d'administrateur

— L'ajout d'une voiture :

L'ajout d'une voiture se fait à partir de la définition de toutes les caractéristiques montrées si-dessous :



The screenshot shows the "Information voiture" form. It has a header "Information voiture" and a blue "Ajouter" button at the bottom. The form contains the following fields:

- Nom voiture :
- Car Image :
- Prix :
- Description :
- active :
- Date début :
- date_fin :

FIGURE 2.9 – L'ajout d'une voiture

— Liste des réservations :

L'admin peut consulter la liste des réservations et les informations liées au clients :

Accueil Ajouter une voiture Réservations Se Déconnecter						
Mes réservations						
ID	Voiture	Prix par jour(DH)	Nom	Prenom	Tel	supression
5	Peugeot 108	300	Test	Taha	0600112233	Annuler

FIGURE 2.10 – La liste des réservations

b) Espace utilisateur :

— Accueil Client :

Le client peut chercher une voiture, afficher la liste de toutes les voitures et réserver une voiture mais il n'a pas le droit de se connecter.

Accueil	Se Connecter
---------	--------------

location de voitures

Voiture date début: date fin:

Merci de spécifier des dates valides




Voiture	carImage	description	Prix par jour(DH)	active	réserver
Peugeot 108		En bon état	300	oui	Réserver
Fiat 500		En bon état	300	oui	Réserver
Dacia Lodgy		En bon état	400	oui	Réserver

FIGURE 2.11 – L'accueil d'utilisateur

— La formulaire de la réservation :

Lorsque le client a choisi la voiture à réserver, il sera être obligé de remplir la formulaire de réservation :

The screenshot shows a web interface for car reservations. At the top, there is a navigation bar with links 'Accueil' and 'Se Connecter'. Below this is a form titled 'Information voiture'. The form contains several input fields, each with a label to its left:

- Nom voiture :** The input field contains the text 'Peugeot 108'.
- Prix :** The input field contains the text '300'.
- Description**: The input field contains the text 'En bon état'.
- Votre prénom**: The input field contains the text 'Taha'.
- Votre nom**: The input field contains the text 'Test'.
- Votre numéro de téléphone**: The input field contains the text '0600112233'.

At the bottom of the form, there is a blue button labeled 'Réserver'.

FIGURE 2.12 – Formulaire de la réservation

Conclusion

Pour conclure, ce projet a été une opportunité pour nous initier à JavaEE. Nous vous avons présenté tout au long de ce rapport la démarche que nous avons suivie pour mettre au point cette plateforme. Toutefois pour améliorer notre site web on peut utiliser Q-learning pour pouvoir s'en passer du chef du projet et de proposer automatiquement des recommandations par apprentissage.

Bibliographie

* [1] <https://openclassrooms.com/>

* [2] <https://develloppers.com/>

* [3] <https://stackoverflow.com/>

* [4] <https://github.com/>