

**Licence 3 Classique**

**Transmission de Données et Sécurité de l’Information**

***La Mise En Place D’Application Web Multi-utilisateurs De Location De Voiture***

PROJET DE FIN D’ANNEE

Présenté Par:

Mr. Khadim Niass

Mlle. Astou Mbengue

**Année académique 2019-2020**

SOMMAIRE

Introduction

1. **Présentation du Projet**

**I.1.Avantage**

**I.2. Différence entre location**

**Traditionnel et électrique**

**II. La Définition des objectifs  
 III. La Cible  
 IV. Méthodes de conception**

**IV.1.Definition d’UML**

**IV.2.Diagramme des cas d’utilisation**

**IV.3.Diagramme de classes  
 V. Identités graphique**

**V.1.Logo**

**V.2.Maquettage des pages**

**V.3.Description environnement de**

**Développement**

Conclusion

**Introduction**

Le monde connait une avance technologique considérable dans tous les secteurs et cela à l’aide de l’informatique, qui joue un rôle important dans le développement de nombreux domaines. Aujourd'hui, l'Internet est exécuté sur une grande échelle pour la plupart des institutions culturelles, scientifiques, économiques et autres

Les locateurs des voitures utilisent une application desktop ou web pour gérer les taches de location, et pour faciliter les relations avec les clients. Dans ce contexte les locateurs de voiture doit avoir un espace de travail sur l'Internet pour publier les annonces et pour en savoir plus sur les nouveaux développements dans le monde de location en ligne pour les voitures pour une période donnée.

Notre projet consiste à réaliser une application web de location de voitures en ligne.

1. **Présentation du Projet**

Le projet que nous allons détailler par la suite est destiné pour la gestion de location de voitures, et il rentre dans le cadre de réalisation d’une Application web fondée sur des technologies et architectures récentes et qui permet l’accès Multi Utilisateurs via un réseau étendu.

**I.1.Avantage**

L’arrivée du numérique dans toutes les sphères de la société entraîne des transformations et des changements qui nécessitent de réinterroger et de repenser nos modes de fonctionnement et de la création. Les nouvelles technologies font naître l’innovation. De ce fait, dans la gestion de location de voiture, on louer ou publier des voiture à distance sans se fatiguer et sans se déplacer. Ainsi

* Un utilisateur peut faire des locations de voiture fournis par un locateur à tout moment, et à partir de n'importe quel endroit.
* Un utilisateur peut faire aussi des publications de voiture à tout moment, et à partir de n'importe quel endroit.
* Informations facilement accessibles. Un client peut voir les informations détaillées  
  pertinentes en quelques secondes des voitures en quelques clics.
* Réduction d’utilisation de papier.
* Facilite la communication entre le client et le locateur

**I.2. Différence entre location traditionnelle et électrique**

Voici les différences qui existent entre la location traditionnelle et électrique.

|  |  |
| --- | --- |
| **Location traditionnelle** | **Location électrique** |
| La forte dépendance à l'échanged'informations d'une personne à uneautre | Le partage de l'information est renduefacile par les canaux de communicationélectronique |
| Uneintervention manuelle est nécessaire pourchaque communication | Lesystème électronique gère les communicationsautomatiquement |
| La location se fait sur place | Offrir la possibilité aux utilisateurs de louer ou publier des voitures en ligne |
| Une perte de temps | Location à distance |

1. **La Définition des objectifs**

Notre travail consiste à créer une application web multiutilisateurs de location de voiture permettant l'échange de bien et de services en ligne, la conception et le développement de notre application vise à atteindre plusieurs objectifs :

* Avoir un accès aux voitures sans frontières.
* La facilité d’échange aux informations
* Offrir la possibilité aux utilisateurs de louer sur l’application.
* Offrir la possibilité aux membres d'avoir un espace de location qui leur est propre, représenter sous forme de parking électronique.
* Réduction de temps de recherche et de diffusion des informations.

**III.La Cible**

* Ouvert au grand public

**IV .Méthodes de conception**

La conception permet la formalisation des étapes préliminaires du développement d’un système, afin de rendre celui-ci plus fidèle aux besoins des utilisateurs. Dans notre cas nous avons utilisé le langage UML.

**IV.1.Definition d’UML**

UML est un langage de modélisation graphique et textuel destiné à comprendre et à décrire les besoins des utilisateurs. UML définit 13 types de diagrammes dont nous avons utilisé que deux dans chaque étape et phase de ce processus de développement. Ces deux diagrammes sont le diagramme des classes et le diagramme des cas d’utilisation.

**IV.2.Diagrammes des cas d’utilisation**

**IV.2.1.Definition**

Le diagramme des cas d’utilisations identifie les fonctionnalités fournies par le système, les utilisateurs qui interagissent avec le système (acteurs), et les interactions  
entre ces derniers.

**IV.2.2. L’identification des acteurs**

Un **acteur** est une entité qui interagit avec le système et qui utilise le système afin d’accomplir un besoin. Les différents acteurs qu’on va trouver dans notre diagramme de cas d’utilisation sont:

**Locateur :** il a le droit

* De s’authentifier
* De s’inscrire
* Publier voiture
* Gérer voiture
* De gérer modèle
* De gérer marque
* Détails les clients
* De gérer les voitures
* Valider location

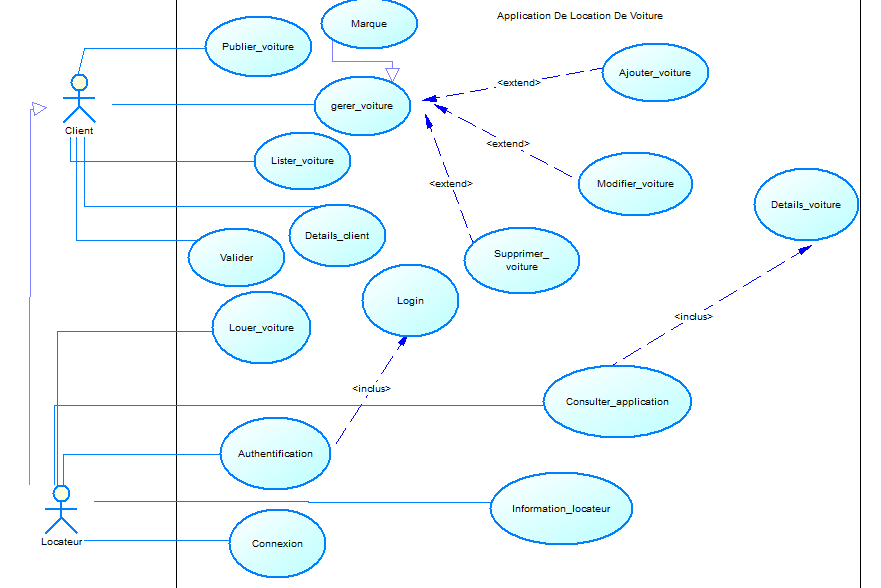
**Client :** il a le droit

* De s’authentifier
* De s’inscrire
* De consulter voiture
* De louer voiture

**IV.2.2. L’identification des cas d’utilisation**

Un cas d’utilisation correspond à un certain nombre d’actions que le système devra exécuter en réponse à un besoin d’un acteur.

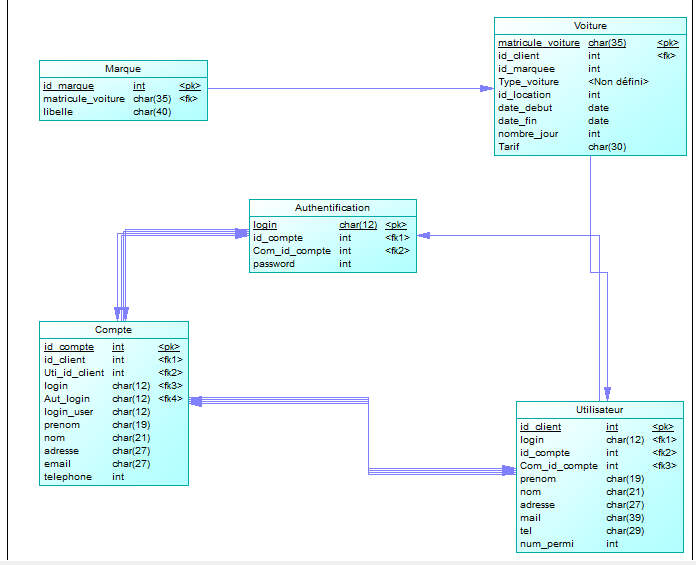
* Gestion des comptes
* Gestion des voitures
* Gestion des clients
* Gestion des locations
* Gestion des publications



**IV.3.** **Diagramme des classes**

**IV.3.1.Definition**

Le diagramme des classes décrit la structure interne du système. Il est considéré comme le plus important de la modélisation orientée objet.  
Une **Classe** est regroupement d’objet de même nature.  
Un **objet** est une entité qui représente l’instance d’une classe. Il est décrit par son identité, son attribut ou état et son comportement ou opération.  
 **IV.2.3.2. Description des classes**Dans chaque classe on y trouve les attributs, les descriptions et les types. On a aussi des relations entre les entités c’est-à-dire les associations et les cardinalités entre les classes.



**V. Identités graphique**

Après avoir abordé les différents diagrammes, on a pu déterminer bien précisément le fonctionnement de notre application et mettre en évidence l’interaction entre nos acteurs et l’application. Maintenant ça nous reste la phase d’implémentation qu’on va aborder.

**V.1.Logo et nom de l’application**

**V.2.** **Maquette des pages**

Pour réussir toutes nos pages on va utiliser quelques langages de programmation comme Java EE, le serveur Apache-Tomcat, HTML et CSS, javascript, Bootstrap et MySQL

.

**V.3.Description environnement de Développement**

* **Java entreprise édition**

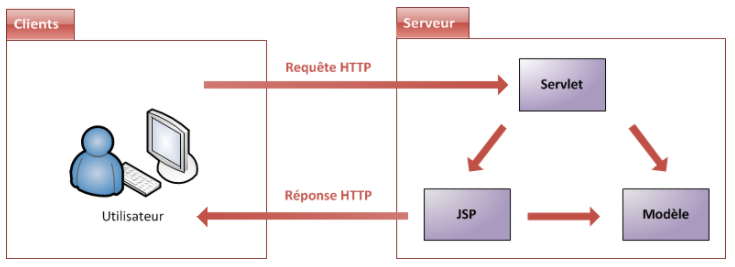
Le terme **Java EE** anciennement raccourci en **J2EE** signifie Java Enterprise Edition, il fait référence à une extension de la plate-forme standard. Autrement dit, la plate-forme Java EE est construite sur le langage Java. L'objectif majeur de Java EE est de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications.  
Java EE regroupe un ensemble d'API chacun est destiné à faire un traitement spécifique. Dans le cadre de notre application, nous avons choisi d'utiliser les API suivants :

**--Servlet** : Une servlet est une classe Java, qui a la particularité de permettre le traitement de requêtes et la personnalisation de réponses.  
**-- JSP** (Java Server Pages) : Les pages jsp sont les pages que les utilisateurs vont voir. Elles comportent des balises html avancées plus du code java.  
**-- JDBC** (Java DataBase Connectivity) Il s'agit en fait de classes Java permettant de se  
connecter et d'interagir avec des bases de données qui nous permet de sauvegarder les données.

**--JavaBean** : Le JavaBean souvent raccourci en "bean", désigne tout simplement un  
composant réutilisable. En réalité, un bean est un simple objet Java.  
**Le modèle MVC** (Model, Vue, Contrôle) permet de séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données. Comme son nom l'indique, le design pattern MVC est composé de trois parties :

* Tout ce qui concerne le traitement, le stockage et la mise à jour des données de l'application doit être contenu dans la couche nommée "Modèle" (le M de MVC) ;
* Tout ce qui concerne l'interaction avec l'utilisateur et la présentation des données doit être contenu dans la couche nommée "Vue";
* Tout ce qui concerne le contrôle des actions de l'utilisateur et des données doit être  
  contenu dans la couche nommée "Contrôle" (le C de MVC).

En pratique, le modèle correspond aux objets java, la vue correspond aux pages JSP, et  
le contrôleur correspond aux Servlet.



* **Le serveur Apache-Tomcat**

Apache-Tomcat est le serveur d'application Java. Ce serveur libre, sous licence Apache permet d'exécuter des applications web développées avec les technologies java.

* **MySQL (My Structured Query Language)**

MySQL est un système de gestion de bases de données relationnelles.

* **HTML et CSS**

HTML signifie ***HyperText Mark up Langage*** ou en français **Langage Hypertexte de Balisages,** son rôle est de gérer et organiser le contenue des pages.

CSS :est l’abréviation de **Cascading Style Sheets** ou en français **feuilles de styles en cascade**, c’est le langage qui vient compléter le HTML. Il gère la mise en forme d’un site web.

* **JavaScript**

Le JavaScript est un langage informatique utilisé sur les pages web. La particularité du JavaScript consiste à créer des petits scripts sur une page HTML dans le but d'ajouter une petite animation la page. Cela permet en général d'améliorer l'ergonomie ou l'interface utilisateur.

* **Bootstrap**

Il permet de construire rapidement et facilement des sites web esthétiques et responsives. Bootstrap offre aussi des plugins jQuery de qualité pour enrichir les pages du site.

* **L'IDE Eclipse**

Eclipse est un IDE, *Integrated Development Environment* (EDI environnement de développement intégré en français), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation.

**Conclusion**