

**Ecole 211 : Future is Now**  
**République du Sénégal**  
**Ministère de l'enseignement supérieur**  
**et de la formation professionnelle**  
**Ecole 221**



**Projet de Web Semestre 2**



**École 221**

Nom de l'appli :

**Gestion de location véhicules**

Lien Github : [https://github.com/Libassalaye24/gestion\\_location\\_vehicule](https://github.com/Libassalaye24/gestion_location_vehicule)

Lien de déploiement : <http://libasse24.alwaysdata.net/>

Lien Trello : <https://trello.com/b/LXksr06q/projetlocationvoiture>

Responsable : **Libasse Mbaye**

# **Introduction**

## **I) Présentation de votre projet**

### **i. Recueil d'information**

Le projet que nous allons détailler par la suite est destiné pour la gestion de location de véhicules, et il rentre dans le cadre de réalisation d'une Application web qui assure le bon fonctionnement d'une société de location de véhicules .

## **II) Analyse et Conception**

### **1. Notion Analyse**

Une analyse est un ensemble de procédés de logiques permettant d'aboutir à des résultats logiquement corrects.

### **2. Quelques Méthode Analyse**

- A. RACINES
- B. Merise
- C. NIAM
- D. OMT (Object Modeling Technique)
- E. APTE
- F. Unified Process utilisant la méthode de notation UML

### **3. Choix de la Méthode**

Pour notre projet nous avons utilisé l'approche merise.

Merise est une méthode d'analyse, de conception et de gestion de projet informatique. Elle est basée sur le principe de la séparation des données et des traitements justifiant notre choix.

### **ii. Dictionnaire**

#### **1. Définition**

Le dictionnaire des données est un document qui regroupe toutes les données que vous aurez à conserver dans votre base (et qui figureront donc dans le MCD).

## 2. Formalisme

# Le dictionnaire de données

| Nom             | Format         | Longueur | Type |   | Règle de calcul | Règle de gestion | Document |
|-----------------|----------------|----------|------|---|-----------------|------------------|----------|
|                 |                |          | E    | C |                 |                  |          |
| Numéro          | Numérique      |          | X    |   |                 |                  | Fiche    |
| Nom             | Alphabétique   | 30       | X    |   |                 |                  | //       |
| Prénom          | Alphabétique   | 30       | X    |   |                 |                  | //       |
| Adresse         | Alphabétique   | 50       | X    |   |                 |                  | //       |
| Code Postal     | Alphanumérique | 10       | X    |   |                 |                  | //       |
| Ville           | Alphabétique   | 50       | X    |   |                 |                  | //       |
| Téléphone       | Alphanumérique | 15       | X    |   |                 |                  | //       |
| Mail            | Alphanumérique | 50       | X    |   |                 |                  | //       |
| Date d'adhésion | Date           |          | X    |   |                 |                  | //       |

## 3. Dictionnaire de Données du projet

| Nom Attribut   | Description      | Identifiant | Type | Nature | Taille | Remarque                    |
|----------------|------------------|-------------|------|--------|--------|-----------------------------|
| id_user        | identifiant      | oui         | E    | N      | 15     |                             |
| nom_user       | nom utilisateur  | non         | E    | AN     | 30     |                             |
| prenom_user    | prenom user      | non         | E    | AN     | 30     |                             |
| email_user     | login            | non         | E    | AN     | 50     |                             |
| password_user  | mot de passe     | non         | E    | AN     | 20     |                             |
| telephone_user | numéro téléphone | non         | C    | N      | 10     | indicatif/opérateur/chiffre |

|                 |                     |     |   |    |    |                         |
|-----------------|---------------------|-----|---|----|----|-------------------------|
| fax_user        | le numero de fax    | non | E | AN | 15 | unique                  |
| id_adresse      | identifiant         | oui | E | N  | 15 |                         |
| numero_adresse  | le numéro           | non | E | N  | 15 | unique                  |
| rue_adresse     | la rue              | non | E | AN | 15 |                         |
| code_postal     | le code postal      | non | E | N  | 15 |                         |
| ville           | nom ville           | non | E | N  | 30 |                         |
| pays            | le pays             | non | E | AN | 30 |                         |
| id_vehicule     | identifiant         | oui | E | N  | 15 |                         |
| numero_vehicule | le numéro           | non | E | AN | 15 | Unique                  |
| immatriculation | la plaque           | non | E | AN | 15 | Unique                  |
| kilometrage     | le kilometrage      | non | E | AN | 10 |                         |
| etat_vehicule   | etat de la vehicule | non | E | AN | 50 | disponible/indisponible |
| longueur        | longueur            | non | E | AN | 5  |                         |
| largeur         | longueur            | non | E | AN | 5  |                         |
| hauteur         | longueur            | non | E | AN | 5  |                         |
| volume_m3       | volume              | non | E | AN | 10 |                         |
| charge_max      | charge              | non | E | AN | 10 |                         |

|                             |                  |     |   |    |    |       |
|-----------------------------|------------------|-----|---|----|----|-------|
| <b>id_conducteur</b>        | identifiant      | oui | E | N  | 15 |       |
| <b>nom_conducteur</b>       | nom              | non | E | AN | 30 |       |
| <b>prenom_conducteur</b>    | prenom           | non | E | AN | 30 |       |
| <b>telephone_conducteur</b> | telephone        | non | E | N  | 10 |       |
| <b>numero_conducteur</b>    | le numero unique | non | E | N  | 10 |       |
| <b>id_modele</b>            | identifiant      | oui | E | N  | 15 |       |
| <b>nom_modele</b>           | le nom           | non | E | AN | 50 |       |
| <b>id_marque</b>            | identifiant      | oui | E | N  | 15 |       |
| <b>nom_marque</b>           | le nom           | non | E | AN | 50 |       |
| <b>id_categorie</b>         | identifiant      | oui | E | N  | 15 |       |
| <b>nom_categorie</b>        | le nom           | non | E | AN | 50 |       |
| <b>id_option</b>            | identifiant      | oui | E | N  | 15 |       |
| <b>nom_option</b>           | le nom           | non | E | AN | 50 |       |
| <b>id_reservation</b>       | identifiant      | oui | E | N  | 15 |       |
| <b>date_debut</b>           | date debut       | non | C | N  | 10 | j/m/a |
| <b>date_fin</b>             | date_fin         | non | C | N  | 10 | j/m/a |
| <b>duree_location</b>       | nbr jours        | non | E | N  | 10 |       |

|                             |                  |     |    |    |    |                             |
|-----------------------------|------------------|-----|----|----|----|-----------------------------|
| montant_total               | le prix          | non | E  | AN | 30 |                             |
| kilometre_parcours<br>s     | nbr<br>kilometre | non | E  | AN | 30 |                             |
| etat_reservation            | etat             | non | E  | AN | 20 | annuler/valider/en<br>cours |
| montant_reservatio<br>n     | la somme         | non | CL | AN | 30 |                             |
| id_type_vehicule            | identifiant      | oui | E  | N  | 15 |                             |
| nom_type_vehicule           | le nom           | non | E  | AN | 35 |                             |
| id_paiement                 | id               | oui | E  | N  | 15 |                             |
| montant_paiement            | montant          | non | E  | N  | 15 |                             |
| id_mode_paiement            | id               | oui | E  | N  | 15 |                             |
| nom_mode_paieme<br>nt       | le mode          | non | E  | AN | 40 |                             |
| id_permis                   | id               | oui | E  | N  | 15 |                             |
| type_permis                 | le type          | non | E  | An | 15 |                             |
| id_prix_location            | id               | oui | E  | N  | 15 |                             |
| prix_location_jour          | prix             | non | E  | N  | 20 |                             |
| prix_location_kilom<br>etre | prix             | non | E  | N  | 20 |                             |
| caution_location            | prix             | non | E  | N  | 20 |                             |

|                  |                    |            |          |           |           |  |
|------------------|--------------------|------------|----------|-----------|-----------|--|
| <b>id_role</b>   | <b>id</b>          | <b>oui</b> | <b>E</b> | <b>N</b>  | <b>15</b> |  |
| <b>nom_role</b>  | <b>le nom</b>      | <b>non</b> | <b>E</b> | <b>N</b>  | <b>25</b> |  |
| <b>id_image</b>  | <b>identifiant</b> | <b>oui</b> | <b>E</b> | <b>N</b>  | <b>15</b> |  |
| <b>nom_image</b> | <b>le nom</b>      | <b>non</b> | <b>E</b> | <b>AN</b> | <b>15</b> |  |

### iii. Graphe de dépendance

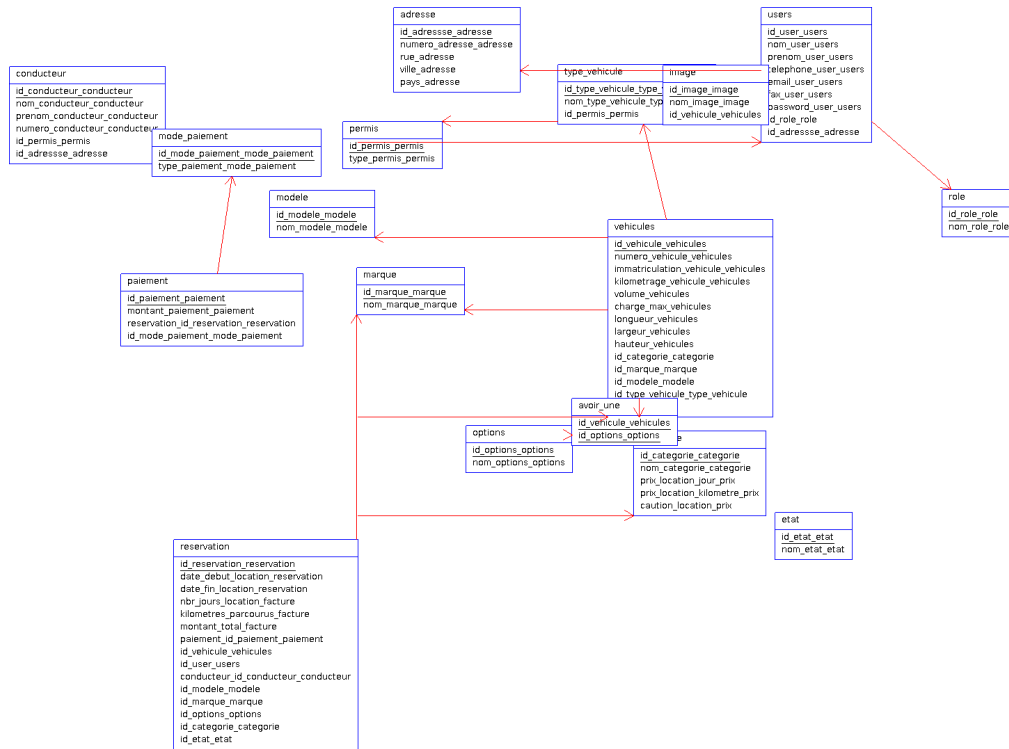
#### 1. Définition

Elle fait apparaître les dépendances fonctionnelles entre les données.

Exemple: la connaissance de la valeur d'un numéro de client nous permet de connaître sans ambiguïté la valeur d'un et d'un seul nom de client.

## 2. Formalisme

### iv. Graphe de dépendance du projet



### v. MCD

#### 1. Définition

Le modèle conceptuel des données (MCD) a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités.

#### 2. Formalisme

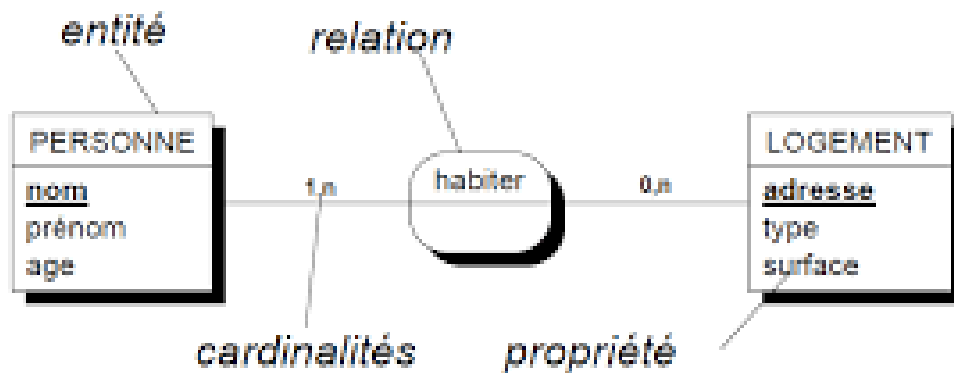
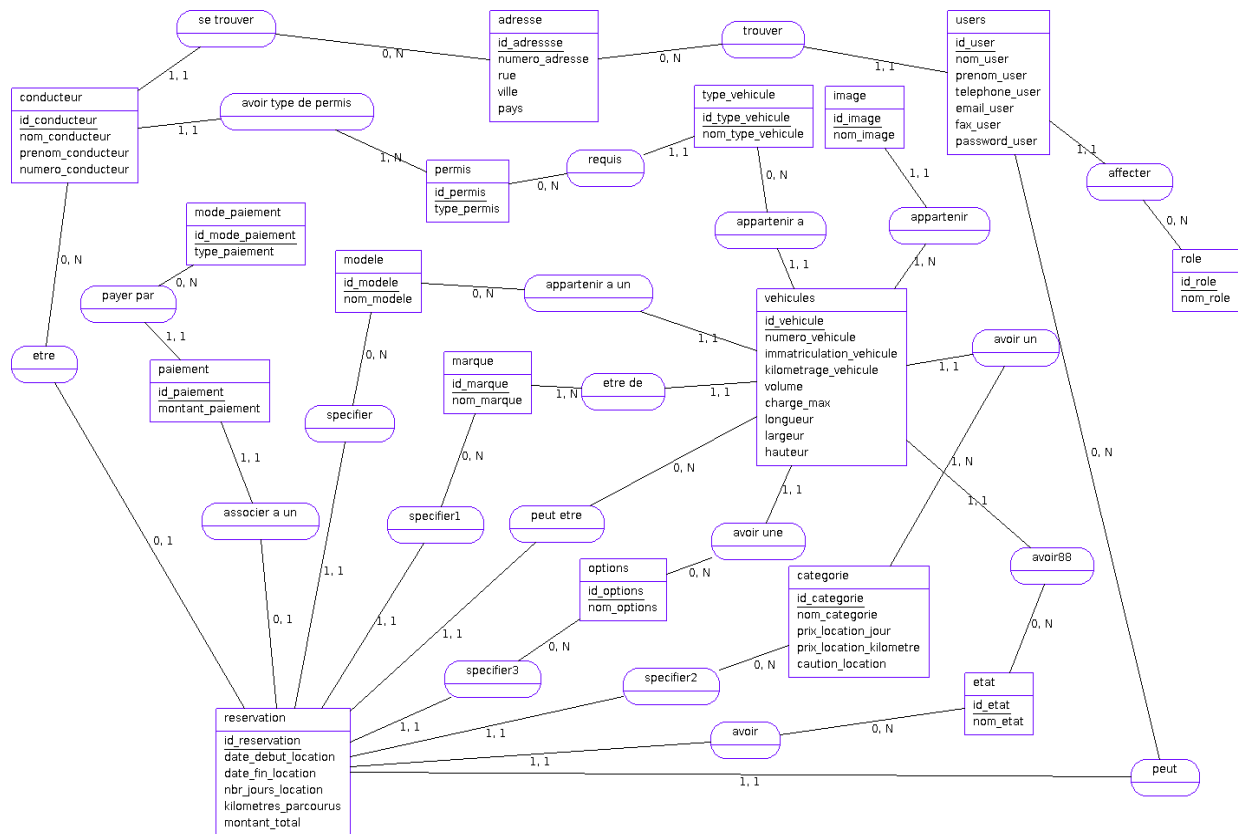


Figure 7.2 : Les concepts du formalisme entité - relation



### 3. MCD du Projet



## vi. MLD

### 1. Définition

Le modèle logique des données consiste à décrire la structure de données utilisée sans faire référence à un langage de programmation. Il s'agit donc de préciser le type de données utilisées lors des traitements.

L'étape logique de données est une étape qui permet de présenter la structure statique du système d'information sans forme d'un modèle de données relationnelles.

Elle a pour but la traduction du modèle conceptuel de données en fonction du SGBD à utiliser lors du développement de l'application.

### 2. Formalisme

Schémas des tables :

! Table DEPARTEMENT (Nom\_departement, Effectif) :

! **Matricule**: clé primaire

! Table EMPLOYE (Matricule, Nom\_departement, Nom, Age, Adresse)

! Matricule: clé primaire

! Nom\_departement : clé étrangère vers table DEPARTEMENT

### 3. MLD du Projet

**marque** (id\_marque\_marque, nom\_marque\_marque)  
**categorie** (id\_categorie\_categorie, nom\_categorie\_categorie,  
prix\_location\_jour\_prix, prix\_location\_kilometre\_prix, caution\_location\_prix)  
**modele** (id\_modele\_modele, nom\_modele\_modele)  
**options** (id\_options\_options, nom\_options\_options)  
**reservation** (id\_reservation\_reservation, date\_debut\_location\_reservation,  
date\_fin\_location\_reservation, nbr\_jours\_location, kilometres\_parcourus,  
montant\_total\_facture, #id\_paiement, #id\_vehicule\_vehicules, #id\_user\_users,  
#id\_conducteur, #id\_modele\_modele, #id\_marque, #id\_options, #id\_categorie,  
#id\_etat\_etat)  
**users** (id\_user\_users, nom\_user\_users, prenom\_user\_users,  
telephone\_user\_users, email\_user\_users, fax\_user\_users, password\_user\_users,  
#id\_role\_role, #id\_adresse\_adresse)  
**permis** (id\_permis\_permis, type\_permis\_permis)  
**paiement** (id\_paiement\_paiement, montant\_paiement\_paiement, #reservation\_id\_  
reservation\_reservation, #id\_mode\_paiement\_mode\_paiement)  
**role** (id\_role\_role, nom\_role\_role)  
**vehicules** (id\_vehicule\_vehicules, numero\_vehicule, immatriculation\_vehicule,  
kilometrage\_vehicule, volume\_vehicules, charge\_max\_vehicules, longueur,  
largeur, hauteur, #id\_categorie, #id\_marque, #id\_options, #id\_modele,  
#id\_type\_vehicule, #id\_etat)  
**type\_vehicule** (id\_type\_vehicule, nom\_type\_vehicul\_type\_vehicule, #id\_permis)  
**mode\_paiement** (id\_mode\_paiement, type\_paiement\_mode\_paiement)  
**adresse** (id\_adresse, numero\_adresse, rue\_adresse, ville\_adresse, pays\_adresse)  
**etat** (id\_etat\_etat, nom\_etat\_etat)  
**image** (id\_image\_image, nom\_image\_image, #id\_vehicule\_vehicules).

vii.

### viii. Analyse Fonctionnelle

L'**analyse fonctionnelle** est une démarche qui « consiste à rechercher et à caractériser les fonctions offertes par un produit pour satisfaire les besoins de son utilisateur. » La démarche est généralement conduite en mode projet et peut être utilisée pour créer (concevoir) ou améliorer un produit.

#### 1. Identifier les acteurs

Les acteurs sont :

Les Clients

Le Gestionnaire

Le Responsable Réservation

#### 2. Besoins par Acteur

En tant que :

**Client je peux:**

faire une réservation

filtrer ses réservations par date ou par état

Annuler ma réservation  
S'inscrire  
Se connecter

### **Gestionnaire je peux:**

d'ajouter, de modifier ou d'archiver les voitures ou camions  
D'ajouter, de modifier ou d'archiver des catégories , les marques  
, les modèles, les options et les conducteurs.

### **Responsable Réservation je peux:**

filtrer les réservations par date ou par état ou par client  
Lister les réservations  
Attribuer une véhicule à un client  
Attribuer un conducteur à un client  
Enregistrer les données du retour au retour du véhicule  
Valider ou annuler une réservation  
Générer une facture après enregistrements du donnees de retour

## **v) Méthodologie de gestion de projet**

### **ix. Identification des User Stories**

#### **x. Définition du Backlog**

Pour résumer, le backlog scrum est destiné à recueillir tous les besoins du client que l'équipe projet doit réaliser. Il contient donc la liste des fonctionnalités intervenant dans la constitution d'un produit, ainsi que tous les éléments nécessitant l'intervention de l'équipe projet. Tous les éléments inclus dans le backlog scrum sont classés par priorité indiquant l'ordre de leur réalisation.

#### **xi. Définition des Sprints**

Conceptualisé par la méthode agile Scrum, un sprint informatique désigne le cycle de développement au cours duquel vont s'enchaîner un certain nombre de tâches pour, à terme, s'achever par la conception d'un produit final.

## xii. Planification des Sprint

### 1. US Maquettage

The screenshot shows a Jira board for the 'US Maquettage' project. The board is divided into two main columns: 'A tester' (To Test) and 'Terminer' (To Finish). Each column contains several task cards. Each card includes a sprint name, a user name, a task description, a due date, and a progress indicator.

| Column   | Sprint  | User                                 | Task Description                                                                                      | Due Date            | Progress |
|----------|---------|--------------------------------------|-------------------------------------------------------------------------------------------------------|---------------------|----------|
| A tester | sprint1 | USER                                 | maquettage et prototypage<br>USER2:maquettage et prototypage de la page d'accueil(liste des vehicule) | 16 sept. - 17 sept. | 0/7      |
|          | USER    | maquettage et prototypage<br>sprint0 | USER0:maquettage et prototypage de la page de connexion et d'inscription                              | 16 sept. - 17 sept. |          |
|          | sprint1 | USER                                 | maquettage et prototypage<br>USER4:maquettage et prototypage de la view reservation                   |                     |          |
|          | sprint2 | USER                                 | maquettage et prototypage<br>USER15:maquettage et prototypage page ajout vehicule                     |                     | 3/3      |
| Terminer | sprint2 | USER                                 | maquettage et prototypage<br>USER19:maquettage page liste modeles,options ,categories,marques         |                     | 6/6      |
|          | sprint3 | USER                                 | maquettage et prototypage<br>USER 27: maquettage et prototypage de la views tableau de bord           |                     | 0/3      |
|          | sprint1 | USER                                 | maquettage et prototypage<br>USER3:maquettage et prototypage de la view liste des reservations        |                     | 0/4      |
|          | sprint1 | USER                                 | Integration<br>USER7:integration de la page d'accueil                                                 |                     | 5/5      |

### 2. US Intégration

The screenshot shows a Jira board for the 'US Intégration' project. The board is divided into two main columns: 'A tester' (To Test) and 'Terminer' (To Finish). Each column contains several task cards. Each card includes a sprint name, a user name, a task description, a due date, and a progress indicator.

| Column   | Sprint  | User        | Task Description                                                                       | Due Date            | Progress |
|----------|---------|-------------|----------------------------------------------------------------------------------------|---------------------|----------|
| A tester | sprint1 | USER        | Integration<br>USRE9:integration de la view reserver                                   |                     |          |
|          | sprint1 | USER        | Integration<br>USER8:integration de la view mes reservations                           |                     |          |
|          | sprint2 | USER        | Integration<br>USER16:integration page ajout vehicule                                  |                     | 2/2      |
|          | sprint3 | USER        | Integration<br>USER 28:integration views tableau de bord                               |                     | 0/3      |
|          | sprint2 | USER        | maquettage et prototypage<br>USER19:maquettage page liste modeles,options              |                     |          |
| Terminer | sprint1 | USER        | Integration<br>USER7:integration de la page d'accueil                                  |                     | 5/5      |
|          | USER    | Integration | sprint0<br>USER5:integration de la page de connexion                                   | 17 sept. - 18 sept. | 0/3      |
|          | USER    | Integration | sprint0<br>USER6:integration de la page d'inscription                                  | 17 sept. - 17 sept. | 0/3      |
|          | sprint2 | USER        | Integration<br>USER20:integration des pages: liste options,categories,marques,model es |                     | 6/6      |
|          |         |             |                                                                                        |                     |          |

### 3. US Fonctionnelle

The screenshot displays a Kanban board for 'US Fonctionnelle' with two columns: 'A tester' and 'Terminer'. Each column contains several cards representing user stories, organized by sprint and user.

**A tester**

- sprint3 USER Fonctionnalitees**  
USER25: en tant que responsable je peux: lister les reservations et les filtrer (2/2)
- sprint3 USER Fonctionnalitees**  
USER:29 en tant que responsable je peux: acceder au tableau de bord (0/7)
- sprint2 USER Fonctionnalitees**  
USER21: en tant que gestionnaire je peux: Ajouter modifier ou archiver vehicule (0/3)
- sprint3 USER Fonctionnalitees**  
USER 30: si un véhicule est reserver son etat change (0/1)
- sprint3 USER Fonctionnalitees**  
USER26: en tant que Responsable je peux: Modifier l'etat d'une reservation (0/2)

**Terminer**

- sprint2 USER Fonctionnalitees**  
USER22: En tant que gestionnaire je peux: Ajouter modifier ,archiver modele,marque,categorie, options (3/3)
- USER Fonctionnalitees sprint0**  
USER10: EN tant que user je peux: M'inscrire et Me connecter et me deconnecter (20 sept. - 21 sept. 0/7)
- sprint1 USER Fonctionnalitees**  
USER11: En tant que client je peux :faire une reservation (1/1)
- sprint1 USER Fonctionnalitees**  
USER12: En tant que client je peux :Lister mes réservations (2/2)
- sprint1 USER Fonctionnalitees**  
USER13: EN tant que client je peux :annuler des reservations (1/1)

### 4. US Déploiement

The screenshot displays a Kanban board for 'US Déploiement' with two columns: 'En cours' and 'A tester'. Each column contains several cards representing user stories, organized by sprint and user.

**En cours**

- sprint3 USER deployment**  
USER27: deployment des fonctionnalitees de responsable reservation
- sprint2 USER deployment**  
USER28: deployment fonctionalites du sprint 2
- USER deployment sprint0**  
USER24: deployment des fonctionnalites de connexion et d'inscription et de la page d'accueil
- sprint0**  
USER: realisation structure MVC

**A tester**

- sprint1 USER integration**  
USRE9: integration de la view reserver
- sprint1 USER integration**  
USER8: integration de la view mes reservations
- sprint2 USER integration**  
USER16: integration page ajout vehicule (2/2)
- sprint3 USER integration**  
USER 28: integration views tableau de bord (0/3)
- sprint2 USER**

## **VI) Outils et technologies utilisées**

### **a. Langages de programmation**

Un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent.

#### **i. PHP**

Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.



#### **ii. HTML**

HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web. Ce langage permet d'écrire de l'hypertexte, d'où son nom, de structurer sémantiquement la page, de mettre en forme le contenu, de créer des formulaires de saisie.



### **iii. CSS**

Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium.



## **b. Serveurs**

### **i. Serveur Web**

#### **1. Définition**

Un serveur web est, soit un logiciel de service de ressources web, soit un serveur informatique qui répond à des requêtes du World Wide Web sur un réseau public ou privé, en utilisant principalement le protocole HTTP.

#### **2. Exemples de Serveurs Web**

Apache

Nginx

IIS

LiteSpeed

Google Servers

### **3. Choix**

Depuis la version 5.4 PHP intègre un serveur web interne qui permet de tester simplement son site. Son avantage est qu'elle permet de définir des variables de configuration à la volée. Si par exemple on souhaite activer l'affichage des erreurs on peut passer l'option directement au démarrage du serveur interne.

#### **c. IDE**

##### **i. Définition**

Un IDE est un ensemble d'outils et d'installations dont les programmeurs ont besoin pour le développement et la conception de logiciels. Il fournit au programmeur un environnement dans lequel il peut écrire et tester ses codes.

##### **ii. Exemples D'IDE**

Visual Studio code.

Sublime text

CODE:: BLOCKS.

**IntelliJ IDEA**

CodeLite.

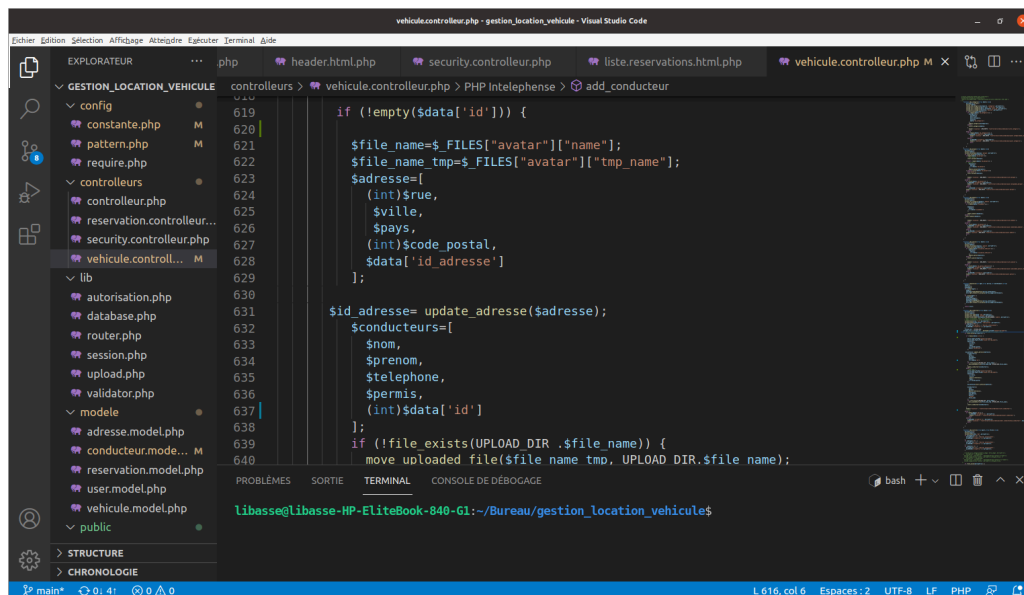
NetBeans.

JetBrains.

##### **iii. Choix IDE**

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code.





## V) Architectures Web

### a. Notion de Client

Dans un réseau informatique, un **client** est le logiciel qui envoie des demandes à un serveur. ... L'ordinateur **client** est généralement un ordinateur personnel ordinaire, équipés de logiciels relatifs aux différents types de demandes qui vont être envoyées, comme un navigateur **web**, un logiciel **client** pour le World wide **web**.

### b. Notion de Serveur

Un **serveur informatique** offre des services accessibles via un réseau. Il peut être matériel ou logiciel, c'est un ordinateur qui exécute des opérations suivant les requêtes effectuées par un autre ordinateur appelé « client ».

### c. Protocoles

Internet protocol est une famille de protocoles de communication de réseaux informatiques conçus pour être utilisés sur Internet.

### d. HTTP ou HTTPS

**HTTPS** n'est pas l'opposé de **HTTP** mais plutôt son petit cousin. Tous deux sont des **protocoles** de transfert hypertexte qui permettent à des données web d'être affichées sur votre écran lorsque vous envoyez une requête. Cependant, **HTTPS** est légèrement différent, plus avancé et bien plus sécurisé.

### e. Notion de requête

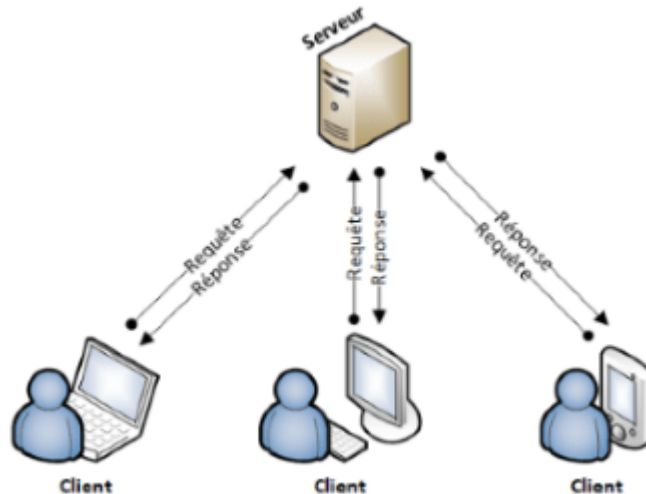
Le terme "*requête*" désigne également l'URL d'une page *web*, saisie dans la barre d'adresse du navigateur *web*, afin d'atteindre cette page.

### f. Notion de réponses

Lorsqu'une ressource demandée est envoyée au navigateur par le serveur **web**, on parle de "réponse".

### g. Communication web

Le protocole ou environnement client–serveur désigne un mode de transaction entre plusieurs programmes ou processus : l'un, qualifié de client, envoie des requêtes ; l'autre, qualifié de serveur, attend les requêtes des clients et y répond. Le serveur offre ici un service au client.



## VI) Architectures MVC

Modèle-vue-contrôleur ou **MVC** est un motif d'**architecture** logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

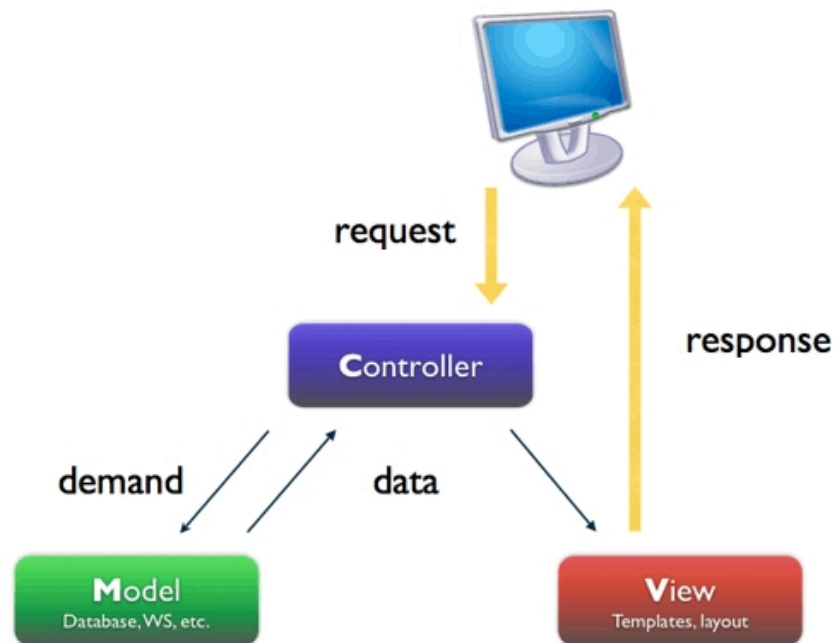
### a. Notion de design pattern

En informatique, et plus particulièrement en développement logiciel, un patron de conception (souvent appelé **design pattern**) est un arrangement caractéristique de modules, reconnu comme bonne pratique en réponse à un problème de conception d'un logiciel.

### b. Design pattern MVC

#### i. Définition

Le modèle de conception MVC (Model View Controller) spécifie qu'une application se compose d'un modèle de données, d'informations de présentation et d'informations de contrôle. Le modèle exige que chacun d'eux soit séparé en différents objets. ... MVC concerne principalement l'interface utilisateur / la couche d'interaction d'une application.



## ii. Modèle

Cette partie gère les *données* de votre site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc entre autres les requêtes SQL.

## iii. Les vues

Cette partie 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, pour afficher par exemple une liste de messages.

## iv. Contrôleurs

Cette partie gère la logique du code qui prend des *décisions*. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du PHP. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès).

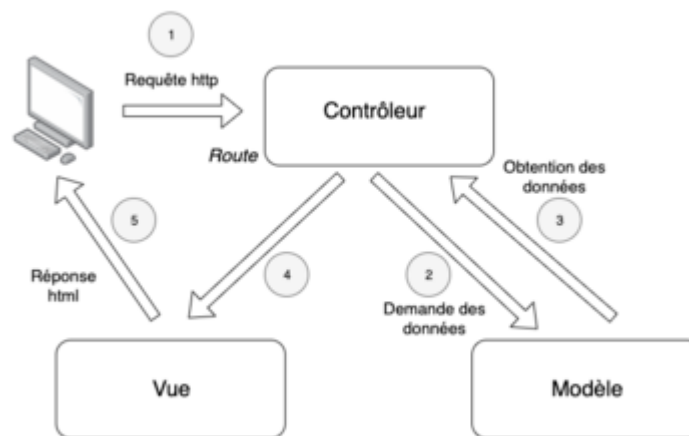
## v. Communication

Il faut tout d'abord retenir que le contrôleur est le chef d'orchestre : c'est lui qui reçoit la requête du visiteur et qui contacte d'autres fichiers (le modèle et la vue) pour échanger des informations avec eux.

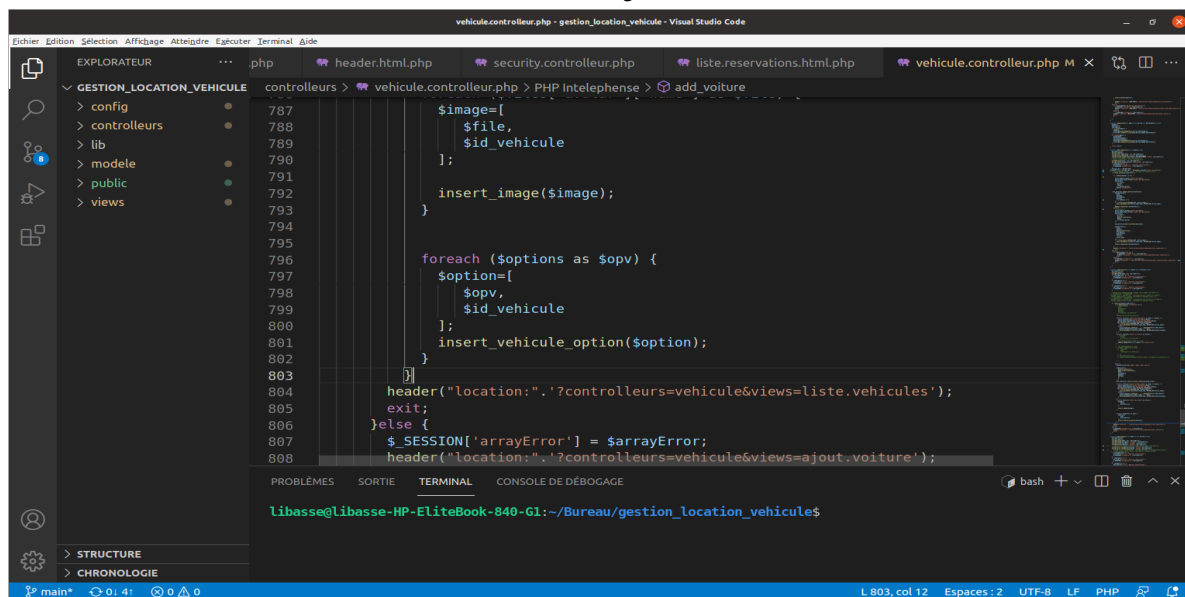
Le fichier du contrôleur demande les données au modèle sans se soucier de la façon dont celui-ci va les récupérer. Par exemple : « Donne-moi la liste des 30

derniers messages du forum numéro 5 ». Le modèle traduit cette demande en une requête SQL, récupère les informations et les renvoie au contrôleur.

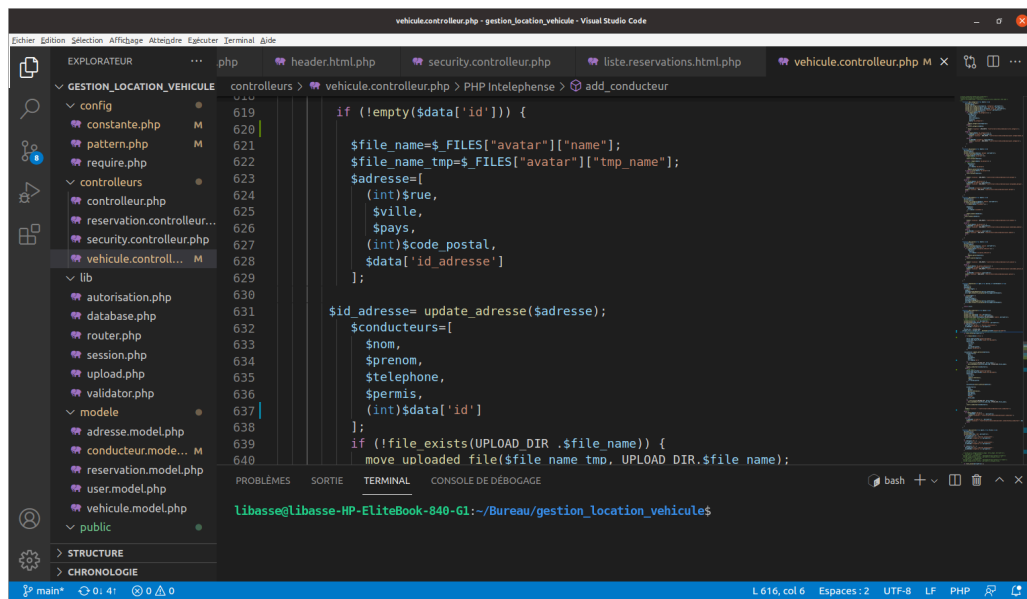
Une fois les données récupérées, le contrôleur les transmet à la vue qui se chargera d'afficher la liste des messages.



## vi. Structure de votre Projet



## VII) Capture d'écrans du projet



## Conclusion :

L'objectif de ce projet de fin d'année était de concevoir et développer une application web pour agence de location de véhicules.

Ce projet nous a donné la possibilité de découvrir et connaître de nouvelles approches de développement dédiée aux applications web et nous a aussi permis de monter en connaissances et en compétences.

## Table des Matières:

|                                      |          |
|--------------------------------------|----------|
| <b>Projet de Web Semestre 2</b>      | <b>1</b> |
| Introduction                         | 2        |
| I) Présentation de votre projet      | 2        |
| i. Recueil d'information             | 2        |
| II) Analyse et Conception            | 2        |
| 1. Notion Analyse                    | 2        |
| 2. Quelques Méthode Analyse          | 2        |
| 3. Choix de la Méthode               | 2        |
| ii. Dictionnaire                     | 2        |
| 1. Définition                        | 2        |
| 2. Formalisme                        | 3        |
| 3. Dictionnaire de Données du projet | 3        |
| iii. Graphe de dépendance            | 7        |
| 1. Définition                        | 7        |
| 2. Formalisme                        | 7        |
| iv. Graphe de dépendance du projet   | 7        |
| v. MCD                               | 7        |
| 1. Définition                        | 7        |
| 2. Formalisme                        | 8        |
| 3. MCD du Projet                     | 8        |
| vi. MLD                              | 9        |
| 1. Définition                        | 9        |
| 2. Formalisme                        | 9        |
| 3. MLD du Projet                     | 9        |
| viii. Analyse Fonctionnelle          | 10       |
| 1. Identifier les acteurs            | 10       |
| 2. Besoins par Acteur                | 10       |
| V) Méthodologie de gestion de projet | 10       |
| ix. Identification des User Stories  | 10       |
| x. Définition du Backlog             | 10       |
| xi. Définition des Sprints           | 11       |
| xii. Planification des Sprint        | 11       |
| 1. US Maquettage                     | 11       |
| 2. US Intégration                    | 12       |
| 3. US Fonctionnelle                  | 12       |
| 4. US Déploiement                    | 13       |
| VI) Outils et technologies utilisées | 13       |
| a. Langages de programmation         | 13       |

|                                 |    |
|---------------------------------|----|
| i. PHP                          | 13 |
| ii. HTML                        | 14 |
| iii. CSS                        | 14 |
| b. Serveurs                     | 15 |
| i. Serveur Web                  | 15 |
| 1. Définition                   | 15 |
| 2. Exemples de Serveurs Web     | 15 |
| 3. Choix                        | 15 |
| c. IDE                          | 15 |
| i. Définition                   | 15 |
| ii. Exemples D'IDE              | 15 |
| iii. Choix IDE                  | 16 |
| V) Architectures Web            | 16 |
| a. Notion de Client             | 16 |
| b. Notion de Serveur            | 16 |
| c. Protocoles                   | 17 |
| d. HTTP ou HTTPS                | 17 |
| e. Notion de requête            | 17 |
| f. Notion de réponses           | 17 |
| g. Communication web            | 17 |
| VI) Architectures MVC           | 17 |
| a. Notion de design pattern     | 17 |
| b. Design pattern MVC           | 18 |
| i. Définition                   | 18 |
| ii. Modèle                      | 18 |
| iii. Les vues                   | 18 |
| iv. Contrôleurs                 | 18 |
| v. Communication                | 19 |
| VII) Capture d'écrans du projet | 20 |
| Conclusion :                    | 20 |
| Table des Matières:             | 21 |