

RÉPUBLIQUE DU CAMEROUN

UNIVERSITÉ DE DOUALA

FACULTÉ DES SCIENCES

Republic of Cameroon

University of Douala

Faculty of SCIENCES

Département de Maths-Info:

**MIAGE 01 :**

**TP MIA 427(Environnement de Développement Intégré)**

Analyse et Conception de l’application de covoiturage **VROOM CMR**

**Membres du groupe 02 :**

1. DJOMO DJOMO JASON YOANN (chef)
2. ELLA AMOUGOU LANDRY STEPHANE
3. ELOCK LIMALEBA FRANCK ERIC ANTOINE
4. EPANYA NGANGUE ROSE SYLENE TEKINE

**Enseignant :**

**Mr ADAMOU MFOPOU**

**Année académique : 2024-2025**

Table des matières

[**ANALYSE ET CONCEPTION** 3](#_Toc188539290)

[**I.** **CHOIX DES OUTILS ET METHODES DE CONCEPTION** 3](#_Toc188539291)

[**1.** **Méthodes de conception** 3](#_Toc188539292)

[**2.** **Choix de la méthode de conduite de projet** 4](#_Toc188539293)

[**II.** **MODELISATION ET DIAGRAMMES** 5](#_Toc188539294)

[**A.** **PARTIE ANALYSE :** 5](#_Toc188539295)

[**1.** **Organigramme de l’application web** 5](#_Toc188539296)

[**2.** **Diagramme de cas d’utilisation** 5](#_Toc188539297)

[**3)** **Diagrammes de séquences** 7](#_Toc188539298)

[**4)** **Diagramme d’activité :** 10](#_Toc188539299)

[**B.** **PARTIE CONCEPTION :** 11](#_Toc188539300)

[***1.*** **Diagramme de classe** 11](#_Toc188539301)

[**2.** **Diagramme de composants :** 13](#_Toc188539302)

[**3.** **Diagramme de déploiement :** 14](#_Toc188539303)

[**Conclusion :** 14](#_Toc188539304)

**ANALYSE ET CONCEPTION**

*L'analyse nous permettra de comprendre en profondeur les exigences de l'application et de décomposer les fonctionnalités principales en sous-composants logiques. Nous examinerons les interactions entre ces composants, les flux de données et les flux d'exécution pour garantir un fonctionnement cohérent et efficace de l'application.*

*En parallèle, la conception de l'application impliquera la création d'une architecture solide et évolutive. Nous déterminerons les différentes couches logicielles, les modèles de données, les interfaces utilisateur et les mécanismes* *d'interaction. Nous veillerons également à intégrer les meilleures pratiques de conception pour assurer la convivialité, l'esthétique et la facilité d'utilisation de l'application.*

Au cours de cette phase, nous nous appuierons sur les informations recueillies lors de l'étude préliminaire et du cahier des charges

## **CHOIX DES OUTILS ET METHODES DE CONCEPTION**

### **Méthodes de conception**

Le langage UML **(Unified Modeling Language, ou langages de modélisation unifié)** a été pensé pour être un langage de modélisation visuel commun, riche sémantiquement et syntaxiquement. Il est destiné à l’architecture, la conception et la mise en œuvre des systèmes logiciels par leur structures aussi bien que leur comportement. L’UML a des applications qui vont au-delà du développement logiciel, notamment pour le flux du processus dans l’industrie. Il se confond au plan utilisé dans d’autre domaine et dispose de plusieurs diagrammes. En gros, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s’y trouvent. UML a une relation directe avec l’analyse et la conception orienté objet.

Le développement d’un système est axé sur 3 modèles de systèmes à savoir :

**Fonctionnel :** ce sont les diagrammes de cas d’utilisation, qui décrivent la fonctionnalité du système du point de vue de l’utilisateur.

**Objet :** ce sont les diagrammes de classe qui décrivent la structure d’un système en termes d’objet, attributs, association et opérations.

**Dynamique :** ce sont les diagrammes d’interaction, diagrammes état-transition et diagrammes d’activités utilise pour décrire le comportement interne du système.



Figure 6:logo UML

### **Choix de la méthode de conduite de projet**

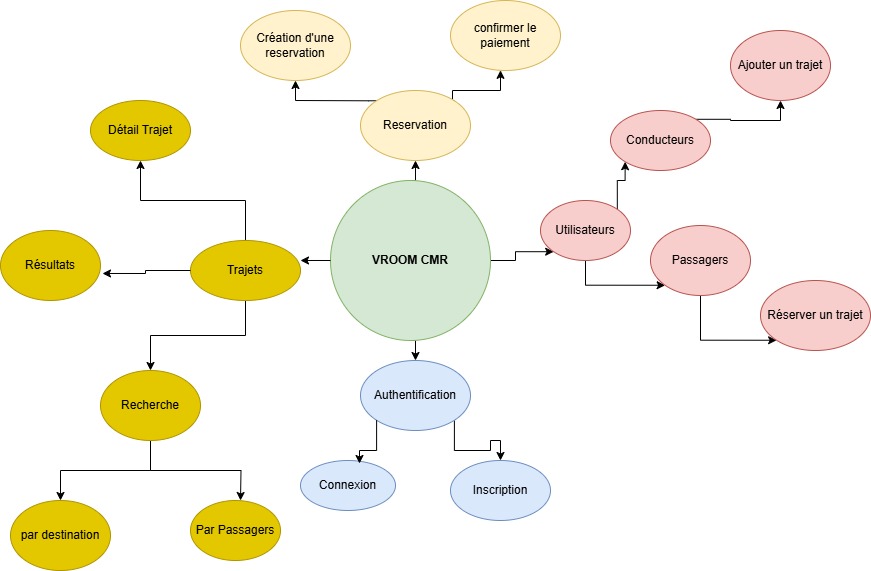
Figure 7:modèle cycle en V

Nous suivons pour ce projet la méthode classique du cycle en V. ceci s’explique d’abord par la petite taille du projet et de l’équipe de développement. De plus, les objectifs sont clairement définis. Cette méthode classique et simple est parfaitement adaptée à notre projet **VROOM CMR.**

## **MODELISATION ET DIAGRAMMES**

1. **PARTIE ANALYSE :**
2. **Organigramme de l’application web**

Ce schéma présente une vue hiérarchique et globale de la structure organisationnelle de l'application **VROOM CMR**. Elle illustre les différentes entités et leurs relations pour offrir une compréhension claire des principaux modules et de leurs interactions.



### **Diagramme de cas d’utilisation**

#### **Concepts de bases**

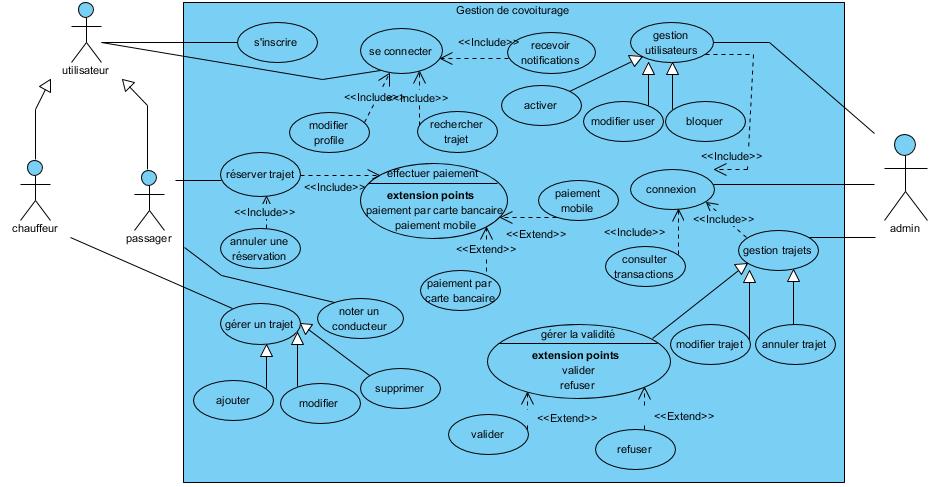
Les diagrammes de cas d’utilisation sont des diagrammes utilisés pour donner une vision globale du comportement fonctionnel d’un système logiciel. Ils sont utilisés pour des présentations auprès de la direction ou des acteurs d’un projet, mais pour le développement, les cas d’utilisations sont plus appropriés. Un cas d’utilisation représente une unité discrète d’interaction entre un utilisateur et un système. Il est une unité de travail significative. Dans un diagramme de cas d’utilisation, les utilisateurs sont appelés acteur (actors) qui interagissent avec les cas d’utilisation (use case).

|  |  |  |
| --- | --- | --- |
| **Elément du diagramme** | **Description** | **Représentation graphique** |
| **L’acteur** | Toute entité qui interagit avec le système | Acteur |
| **Le cas d’utilisation** | Fonctionnalité réalisable par le système | Cas\_1  Cas\_1 |
| **L’association** | Lien entre un acteur et les cas d’utilisation auxquels il a accès |  |

Tableau: concept de bas diagramme de cas d'utilisation

1. **Réalisation :**

Dans notre cas nous avons 02 principaux acteurs que sont les utilisateurs (chauffeurs et passagers) et l’administrateur. Ainsi, ci-dessous le diagramme de cas d’utilisation qui en découle :



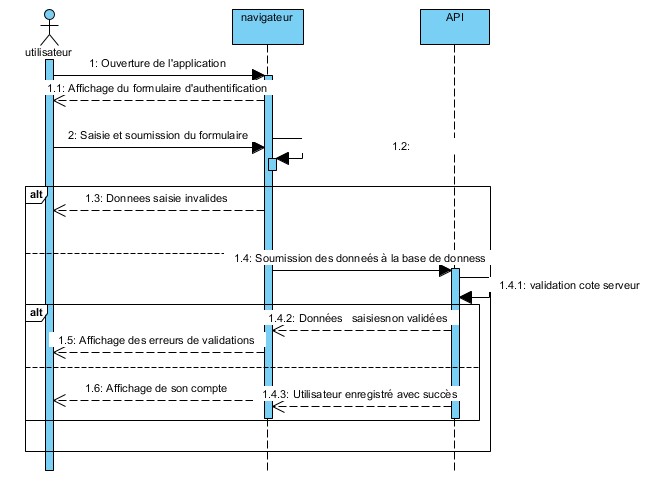
### **Diagrammes de séquences**

#### **Concepts de bases**

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un [scénario](https://fr.wikipedia.org/wiki/Sc%C3%A9nario_(informatique)) d'un [diagramme des cas d'utilisation.](https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation) 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 est de décrire comment se déroulent les interactions entre les acteurs ou objets. La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent à l'aide de messages.

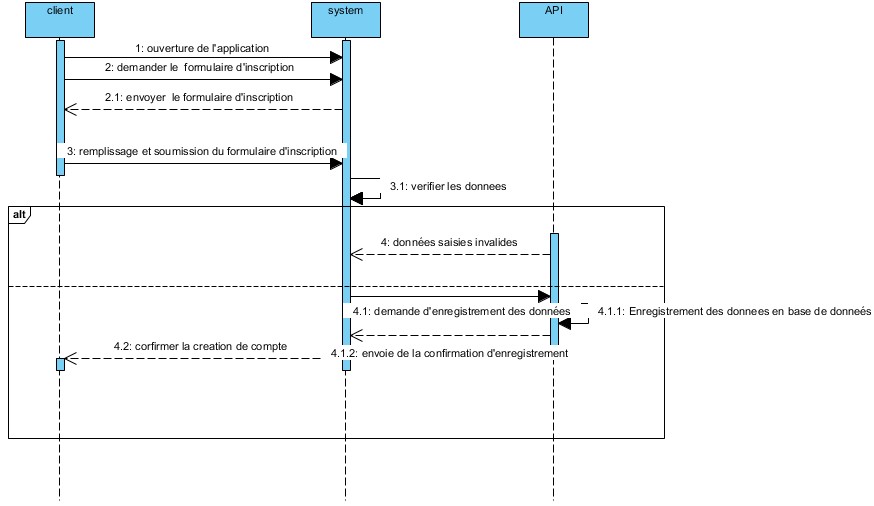
|  |  |  |
| --- | --- | --- |
| **Elément du diagramme** | **Description** | **Représentation graphique** |
| Objet | Ce sont des instances de classe | Objet\_1  Objet\_1 |
| Acteur | Toute entité qui interagit avec le système | Acteur\_1  Acteur\_1 |
| Ligne de vie | Ligne verticale qui identifie l’existence de l’objet par rapport au temps |  |
| Activation | Boite rectangulaire sur une ligne de vie marquant la portée d’une action |  |
| Message | Flèche horizontale portant un texte qui indique les communications objet-acteur et objet-objet | Message\_1 |

#### **Réalisation**



BD

Figure: Diagramme de séquences Connexion



BD

Figure: diagramme de séquences Inscription

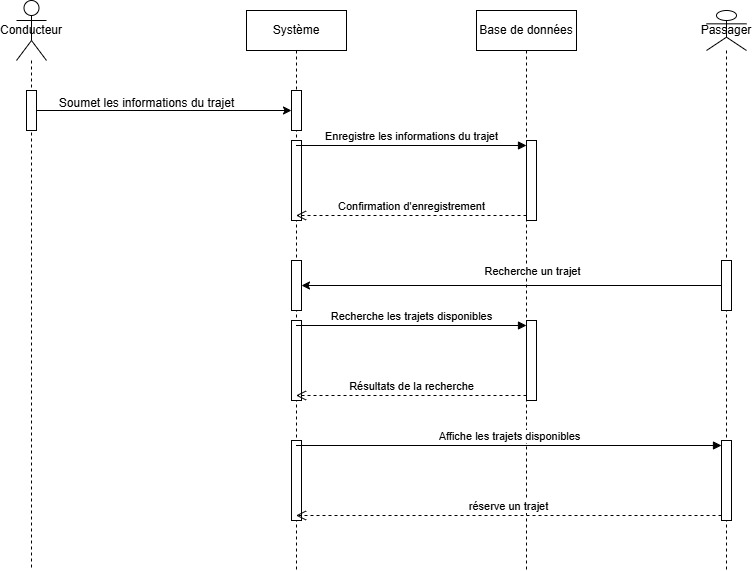


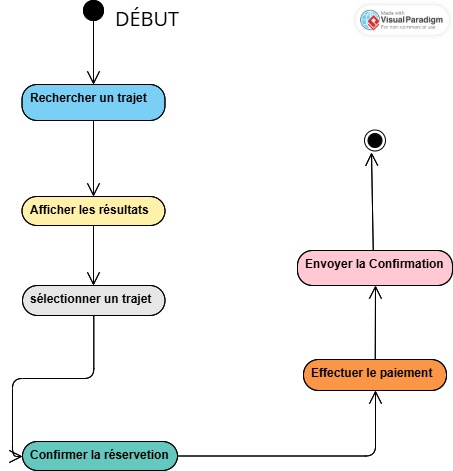
Figure: diagramme de séquences pour ajout d’un conducteur et pour un passager

1. **Diagramme d’activité :**

#### **Concepts de bases**

Ce diagramme décrit le flux des activités et des décisions liées à un processus spécifique de l'application. Elle est utile pour identifier les chemins alternatifs et les points de décision dans le système.

1. **Réalisation**



1. **PARTIE CONCEPTION :**

### **Diagramme de classe**

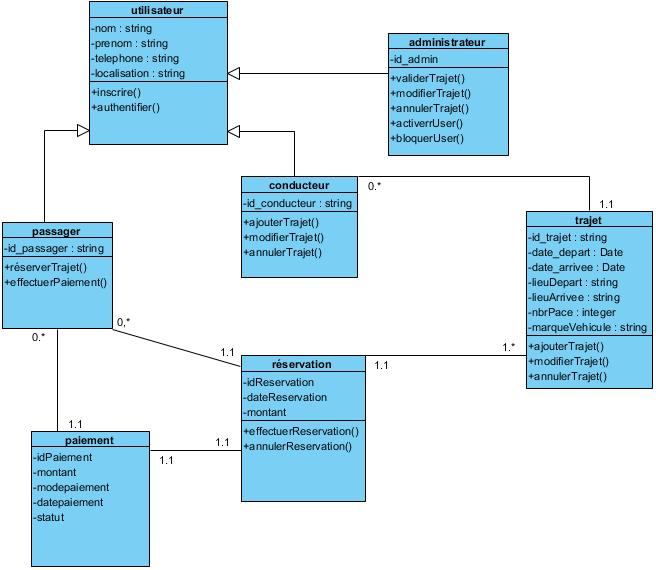
#### **Concepts de base**

Le diagramme de classe est un schéma utilisé en génie logiciel pour présenter les classe et les interfaces des systèmes ainsi que les liens entre celles-ci. Ce diagramme fait partie du côté statique d’UML car il reflète les aspects temporels et dynamiques. Une classe décrit les responsabilités, le comportement et le type d’un ensemble d’objet. Les éléments de ces ensembles sont les instances de la classe. Une classe est un ensemble de fonctions et de données(attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles ont pour rôle de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simple. Les classes peuvent être liées entre elles par la notion d’héritage qui permet de mettre en évidence des relations de parenté.

|  |  |
| --- | --- |
| Note | Une note est un commentaire placé sur un diagramme. |
| **Classes** | Modelé de représentation des objets qui possède des données et des comportements |
| **Attribut** | Propriétés d’une classe qui décrit un domaine de valeur possibles partagé par tous les objets de classes. |
| **Opération** | Traitement qui peut être demandé à n’importe quel objet de la classe et partagée par tous les objets de la classe. |
| Héritage  Composition  Agrégation | Relations qui pourraient exister entre les classes. |

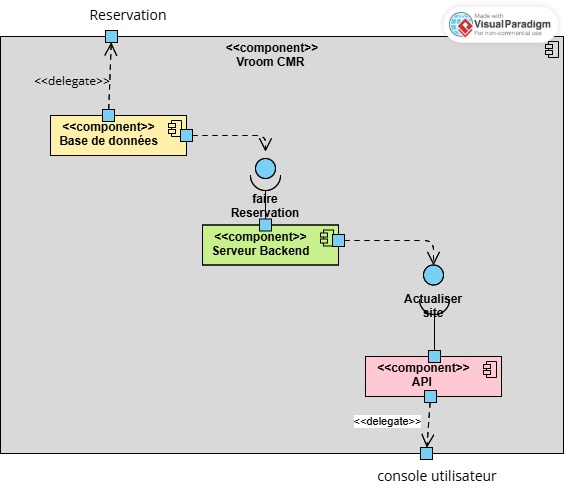
Tableau: concepts de base diagramme de classe

#### **Réalisation :**



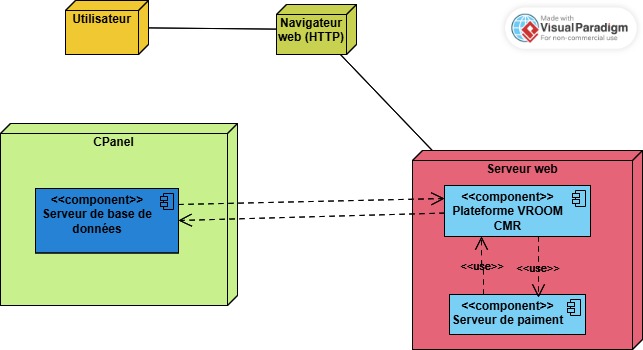
1. **Diagramme de composants :**

Cette figure illustre l'architecture logicielle de l'application en présentant les composants logiciels et leurs relations. Elle sert à définir la manière dont les fonctionnalités de l'application sont réparties entre les différentes parties du système.



1. **Diagramme de déploiement :**

Ce diagramme montre l'architecture physique de l'application **VROOM CMR**. Il présente les différents nœuds (serveurs, bases de données, etc.) et leur communication, permettant de comprendre la manière dont le système sera déployé.



# **Conclusion :**

L'analyse et la conception de l'application web **VROOM CMR** ont permis de poser les bases solides pour la mise en œuvre de ce projet innovant. À travers les différents diagrammes UML, nous avons pu modéliser les besoins fonctionnels et techniques, tout en identifiant les interactions clés entre les utilisateurs et le système. L'organigramme a offert une vision hiérarchique de l'application, les diagrammes de cas d'utilisation ont clarifié les fonctionnalités principales, et les diagrammes de séquence et d'activité ont détaillé les processus critiques.