



Filière Sciences Mathématiques et Informatique

Projet de Fin d'Etudes

Semestre S_6

Mémoire

Gestion de location de voiture

Présenté par :

Bouhrir Ayat

Zerroudi Jalal

Encadrant : **Pr. Bennani Mohamed Taj**

Pr. Yaden Mohammed Faysal

Soutenu le : 25/05/2024

Jury : Examineur 1 : **Pr. Mohamed El Far**
Examineur 2 : **Pr. Mohamed Lamrini**
Examineur 3: **Pr. Khalid El Fahssi**
Examineur 4: **Pr. Said EL GAROUANI**

Année Universitaire : 2023/2024

Remerciements

Au nom d'Allah le tout miséricordieux,

Nous tenons à adresser nos sincères remerciements à nos encadrants, le professeur **Bennani Mohamed Taj** et le professeur **Yaden Mohammed Faysal**, pour leur accompagnement précieux tout au long de mon projet de fin d'études. Leur expertise, leurs conseils judicieux et leur soutien constant ont été essentiels à la réussite de ce travail. Je suis profondément reconnaissant pour leur patience et leur engagement envers mon développement académique et professionnel

Nous tenons également à remercier **les membres du jury** pour leur évaluation attentive de notre travail.

Un merci spécial à tous ceux qui ont contribué, directement ou indirectement, à la réalisation de ce projet.

Résumé

Ce projet de fin d'études présente le développement d'une application de bureau innovante, destinée à révolutionner la gestion des opérations au sein d'une agence de location de voitures. L'objectif principal est de simplifier et d'automatiser l'organisation des données, améliorant ainsi l'efficacité et la productivité de l'agence.

L'application offre une solution complète pour la gestion des réservations, des clients, des véhicules et des employés. Elle permet aux employés de gérer facilement les flux de travail quotidiens, l'application permet de réduire les erreurs humaines et de maximiser l'utilisation des ressources.

La phase de conception du projet a débuté par une analyse des besoins dans ce secteur, suivie par l'élaboration d'une étude conceptuelle et technique. Cette étape préliminaire a été cruciale pour définir les spécifications et l'architecture logicielle nécessaires.

Le **backend** de l'application est basé sur deux éléments essentiels : le langage de programmation **C#** et **MySQL** pour la gestion de la base de données.

Le **frontend** utilise **Windows Presentation Foundation (WPF)** avec le framework **.NET 8.0** pour offrir une expérience utilisateur riche et interactive.

Sommaire

Remerciements.....	2
Résumé	3
Sommaire	4
Liste des figures.....	6
Liste des Tables.....	6
Liste des Acronymes.....	7
Introduction Générale	8
Chapitre 1 : Analyse et cadre générale de projet	9
I. INTRODUCTION.....	9
II. Étude de cahier de charge	9
2.1 Problématique	9
2.2 L'objectif.....	9
2.3 Solution.....	9
III . Analyse des besoins.....	10
3.1 Les besoins fonctionnels	10
3.2 Les besoins techniques	11
a. Outils et technologies	11
b. Les langages de programmation utilisés.....	13
c. Les cadres applicatifs (Frameworks)	14
1) WPF.....	14
2) NET 8.0	14
d. Architecture d'application	15
IV. Conclusion.....	16
Chapitre 2 : Conception et modélisation.....	17
I. Introduction	17
II. Gestion de projet	17
2.1 Cycle de vie	17
2.2 Les modelés du cycle de vie	17
a. Cycle de vie en cascade.....	17
b. Cycle de vie en v.....	18
2.3 Diagramme de <i>Gantt</i>	19
III. Présentation UML	19
a. Diagramme de cas d'utilisation	20
b. Diagramme de classe	20

c.	Diagramme de séquence	21
IV.	Notre projet de location de voiture	22
a.	Planification.....	22
b.	Cycle de vie en V	23
c.	Diagramme de Gantt.....	23
d.	Modélisation	24
1)	Les acteurs.....	24
2)	Diagramme de cas d'utilisation	25
3)	Diagramme de classe	26
4)	Diagramme de séquence	27
V.	Conclusion.....	38
Chapitre 3 : Réalisation du projet		39
I.	Introduction	39
II.	Application Desktop.....	39
2.1	Les interfaces et les explications	39
1)	Authentification	39
2)	Interface Forget password.....	40
3)	Les interfaces d'administrateur	41
a.	Interface accueil.....	41
b.	Interface voiture.....	42
c.	Interface employée	43
d.	Interface notification	44
4)	Les interfaces d'employée	45
a.	Interface accueil.....	45
b.	Interface client	46
c.	Interface réservation	47
d.	Interface paiements.....	48
e.	Interface notification	49
Conclusion Générale		50
Webographie		51

Liste des figures

Figure 1: Visual Studio logo	11
Figure 2: Laragon logo	11
Figure 3:Entreprise Architect logo	12
Figure 4: MS Project logo	12
Figure 5:C# logo	13
Figure 6: MySQL.....	13
Figure 7: WPF logo	14
Figure 8 : .NET 8.0 logo	14
Figure 9 : MVVM logo	15
Figure 10: MSIX logo.....	15
Figure 11:Cycle de vie en cascade.....	18
Figure 12: Cycle de vie en v.....	19
Figure 13 : Planification.....	22
Figure 14: Cycle de vie en v.....	23
Figure 15 : Diagramme de GANTT partie1	23
Figure 16: Diagramme de GANTT partie2.....	24
Figure 17 : Diagramme de cas d'utilisation	25
Figure 18 :Diagramme de classe	26
Figure 19 : diagramme de séquence(login) Pages : [27-28]	28
Figure 20: diagramme de séquence (employer) Pages : [29-34]	34
Figure 21 : diagramme de séquence (administrateur) Pages : [35-38]	38
Figure 22 : login	39
Figure 23 : forget password.....	40
Figure 24 : interface d'accueil pour l'administrateur	41
Figure 25 : interface des voitures pour l'administrateur	42
Figure 26 : interface des employés pour l'administrateur.....	43
Figure 27 : interface de notification pour l'administrateur.....	44
Figure 28 : interface d'accueil pour l'employé	45
Figure 29 : interface client pour l'employé	46
Figure 30 : interface de réservation pour l'employé	47
Figure 31 : interface de paiements pour l'employé	48
Figure 32 : interface de notification pour l'employé	49

Liste des Tables

Tableau 1:la liste des acronymes dans le projet	7
Tableau 2: table des acteurs	24

Liste des Acronymes

Tableau 1: la liste des acronymes dans le projet

Abréviation	Désignation
UML	Unified Modeling Language
C#	C Sharp
MySQL	My Structured Query Language
MVVM	Model-View-ViewModel
.NET 8.0	Network Enabled Technologies (version 8.0)
WPF	Windows Presentation Foundation
MSIX	Microsoft Installer for XML

Introduction Générale

La location de voitures est devenue un secteur en pleine expansion, dont la compétitivité augmente jour après jour. Ce service permet aux clients, professionnels ou particuliers, de réserver et prendre un véhicule pour une période donnée, allant de quelques jours à plusieurs mois.

À l'approche de la Coupe du Monde en 2030, le Maroc voit affluer de plus en plus de touristes, ce qui augmente la demande pour les agences de voyages et de location de voitures. Pour répondre efficacement aux besoins des visiteurs étrangers, une bonne organisation au niveau des agences de location de voitures est cruciale.

C'est pourquoi nous nous engageons à réaliser une application de bureau pour gérer les voitures, les employés, les clients et l'administration.

Le présent rapport synthétise tout le travail que nous avons effectué. Il est composé de plusieurs chapitres :

Le premier chapitre établit le cadre général du projet en présentant la problématique, les objectifs, la solution envisagée, l'analyse des besoins ainsi que les algorithmes associés.

Le deuxième chapitre détaille l'analyse fonctionnelle et technique du projet, la conception et les différents diagrammes UML utilisés.

Le troisième chapitre expose les différentes interfaces de l'application.

Nous terminons par une conclusion générale.

Chapitre 1 : Analyse et cadre générale de projet

I. INTRODUCTION

Ce chapitre présente le projet de développement d'une application de bureau pour la gestion d'une agence de location de voitures. Nous aborderons les spécifications fonctionnelles et techniques, les défis actuels rencontrés par les agences de location, la solution proposée et un aperçu du planning du projet.

II. Étude de cahier de charge

2.1 Problématique

La plupart des agences de location de voiture utilisent la méthode traditionnelle, qui est basée sur des journaux, ni organisation ni hiérarchie... les documents et les fichiers étant éparpillés, rendant ainsi les tâches plus difficiles et engendrer par la suite une grande perte de temps et d'argent.

2.2 L'objectif

L'objectif principal de ce projet est de fournir un système complet qui contient une application de bureau permettant d'effectuer les tâches suivantes : gérer les voitures, les employés, les clients et l'administration.

2.3 Solution

Après une étude approfondie, qui a abouti à la conception d'une application de bureau offrant les fonctionnalités standards d'une société de location de voitures, celle-ci promet de simplifier le travail au sein de l'agence. L'application comportera deux espaces distincts : un espace réservé aux employés, leur permettant de gérer les clients, les réservations et les paiements, et un autre espace destiné à l'administrateur, lui donnant

accès à la gestion du personnel, des véhicules et de toutes les tâches pouvant être effectuées par ce dernier.

III. Analyse des besoins

Dans la phase d'analyse, on cherche d'abord à bien comprendre et à décrire de façon précise les besoins des utilisateurs de l'application. Que souhaitent-ils faire avec cette application ? Quelles fonctionnalités veulent-ils ? Pour quel usage ? Comment l'action devrait-elle fonctionner ? C'est ce qu'on appelle « l'analyse des besoins ». Il y a deux types de besoins.

3.1 Les besoins fonctionnels

La solution proposée consiste à réaliser une applications desktop qui offerts :

❖ Gestion des clients :

- Voir les voitures disponibles.
- Réserver, signer un contrat et payer.

❖ Gestion des voitures :

- Enregistrement des informations sur les véhicules (numéro de matricule, couleur, équipements, type...) est une tâche effectuée par l'administrateur.

❖ Gestion des employés :

- Renseigner le client
- Voir la disponibilité, établir, vérifier les réservations.
- Confirmer une réservation.
- Valider le paiement.

❖ Gestion d'administration :

- Enregistrement des informations des

employés et des voitures. (Ajouter, modifier, supprimer).

- Maintenance des voitures.

3.2 Les besoins techniques

a. Outils et technologies



Figure 1: Visual Studio logo

Visual Studio est un environnement de développement intégré (IDE) avancé de Microsoft, conçu pour les développeurs de logiciels travaillant principalement sur les plateformes Windows et .NET. Il prend en charge divers langages de programmation, notamment C#, VB.NET, et C++. Visual Studio est reconnu pour ses outils robustes de débogage, de test, et de gestion de versions, qui facilitent le développement d'applications de bureau, web et mobiles. Avec une interface utilisateur intuitive et des options de personnalisation via des extensions, Visual Studio aide les développeurs à optimiser leur flux de travail et à améliorer la qualité de leurs projets.



Figure 2: Laragon logo

Laragon est une plateforme de développement web locale qui simplifie la configuration et la gestion des environnements de développement pour les développeurs. Il intègre un serveur web (Apache ou Nginx), un système de gestion de base de données (**MySQL** ou MariaDB), et un ensemble d'outils pour le développement web (C#, Node.js, Python, etc.) dans une interface conviviale.

Laragon permet aux développeurs de démarrer rapidement leurs projets web sur leur machine locale, en offrant une installation simple, des fonctionnalités de gestion avancées et une compatibilité avec un

large éventail de technologies web.



Figure 3: Enterprise Architect logo

Enterprise Architect est un outil de modélisation UML largement utilisé pour la conception et la modélisation de systèmes d'information et de logiciels. Cet outil offre une large gamme de fonctionnalités pour les architectes d'entreprise, y compris la création de diagrammes UML, la modélisation de processus métier, la gestion des exigences et des tests.



Figure 4: MS Project logo

Microsoft Office Project, aussi appelé **Microsoft Project**, est un logiciel de gestion de projet extrêmement complet et très prisé dans les entreprises et organisations de divers secteurs. Ce programme permet aux utilisateurs de planifier, de suivre et de gérer des projets de différentes envergures et complexités de manière efficace. Microsoft Project offre plusieurs avantages, tels que :

- La capacité de planifier les différentes étapes d'un projet, de fixer des échéances et des calendriers, et de suivre leur progression en temps réel.
- La possibilité d'affecter les ressources nécessaires à chaque phase du projet et de surveiller les coûts associés à chaque activité.
- L'option de visualiser les projets via des diagrammes de Gantt et des diagrammes de réseau pour une meilleure compréhension de l'avancement du projet.
- La personnalisation des tableaux de bord pour qu'ils répondent aux exigences spécifiques de chaque projet

b. Les langages de programmation utilisés



Figure 5: C# logo

C# (C Sharp) est un langage de programmation multiparadigme développé par Microsoft. Il englobe des disciplines telles que la programmation impérative, déclarative, fonctionnelle, générique, orientée objet (basée sur la classe) et orientée vers les composants. C# est largement utilisé pour le développement d'applications logicielles, en particulier dans l'écosystème de développement sur la plateforme .NET de Microsoft. Il offre aux développeurs un ensemble riche de fonctionnalités et une syntaxe intuitive pour la création d'applications robustes et évolutives pour une variété de plates-formes, y compris les applications de bureau, les applications web et les applications mobiles



Figure 6: MySQL

MySQL est un système de gestion de base de données relationnelle open source très populaire. Il est utilisé par de nombreuses entreprises et organisations pour stocker et gérer efficacement de grandes quantités de données.

MySQL est réputé pour sa fiabilité, sa stabilité et sa performance, ainsi que pour sa facilité d'utilisation. Il permet aux utilisateurs de créer, de modifier et de supprimer des bases de données, des tables et des colonnes de manière simple et intuitive.

En outre, **MySQL** prend en charge le langage de requête SQL, qui est utilisé pour interroger et extraire des données de la base de données. C'est une solution de base de données flexible qui est utilisée par des millions d'utilisateurs à travers le monde.

c. Les cadres applicatifs (Frameworks)

1) WPF



Figure 7: WPF logo

Le **Windows Presentation Foundation (WPF)** est un framework de développement logiciel conçu par Microsoft pour créer des applications Windows avec des interfaces utilisateur graphiques modernes et interactives. Il utilise un langage de balisage appelé **XAML (eXtensible Application Markup Language)** pour définir l'apparence et le comportement des éléments de l'interface utilisateur. WPF offre des fonctionnalités telles que la mise en page flexible, les animations, la liaison de données, et facilite la création d'applications riches en fonctionnalités.

2) NET 8.0



Figure 8 : .NET 8.0 logo

Le **Framework .NET 8.0** est une plateforme de développement logiciel conçue par Microsoft pour créer des applications sur différentes plateformes telles que Windows, macOS et Linux. Il fournit un ensemble d'outils et de bibliothèques pour simplifier le processus de développement et permet aux développeurs de créer des applications robustes et évolutives en utilisant des langages de programmation tels que **C#, F# et VB.NET**.

d. Architecture d'application



Figure 9 : MVVM logo

L'architecture **MVVM** (**Modèle-Vue-VueModèle**) est un modèle de conception utilisé pour développer des applications logicielles, en particulier pour les applications de bureau et mobiles.

Dans **MVVM** :

- Le **Modèle** gère les données et la logique métier.
- La **Vue** représente l'interface utilisateur (**UI**) visible par l'utilisateur.
- Le **VueModèle** agit comme un intermédiaire entre le Modèle et la Vue, gérant la logique de présentation et fournissant les données à afficher.

MVVM facilite la séparation des préoccupations entre l'interface utilisateur et la logique métier, ce qui rend le code plus clair, maintenable et testable. Cela permet également une collaboration plus efficace entre les développeurs d'interfaces utilisateur et les développeurs de logiciels.

e. Format de package d'application



Figure 10: MSIX logo

MSIX est un format de package d'applications Windows qui offre une expérience d'empaquetage moderne pour toutes les applications Windows. Le format

de package **MSIX** conserve les packages d'applications et/ou les fichiers d'installation existants tout en proposant de nouvelles fonctionnalités d'empaquetage et de déploiement modernes pour les applications Win32, WPF et UWP.

MSIX est un format d'empaquetage pour applications Windows conçu pour être sécurisé et fiable.

IV. Conclusion

Ce premier chapitre a fourni un panorama complet du projet, abordant sa problématique, son contexte et les outils techniques utilisés. Le prochain chapitre se concentrera sur l'analyse et la conception, où nous examinerons en détail les concepts essentiels à la gestion de location de voitures, afin d'améliorer et d'optimiser nos processus.

Chapitre 2 : Conception et modélisation

I. Introduction

Dans ce chapitre, nous aborderons les fondamentaux de la gestion de projet et du développement logiciel, en mettant un accent particulier sur l'analyse et la conception de systèmes.

II. Gestion de projet

La Gestion de projet est indispensable aux professionnelles, elle joue un rôle de plus en plus déterminant au quotidien dans toutes les activités professionnelles. Chaque année, les entreprises ont de nombreux challenges à relever :

- Lancement de nouveaux services ou produits innovants (développement d'un nouveau logiciel par exemple).
- Mise à jour de technologies déjà implantées pour rester compétitif.
- Adaptation à des contraintes légales nationales, ou internationales.

2.4 Cycle de vie

Cycle de vie (lifecycle) désigne la période de naissance d'un logiciel à sa mise hors service définitive, en passant par sa construction et son utilisation. La vie d'un logiciel est composée de différentes étapes. La succession de ces étapes forme le cycle de vie du logiciel. Il faut contrôler la succession de ces différentes étapes.

2.2 Les modelés du cycle de vie

a. Cycle de vie en cascade

C'est le modèle le plus simple, il se caractérise par un déroulement de phases successives. Il est adapté pour des projets de petite taille, et dont le domaine est bien maîtrisé.

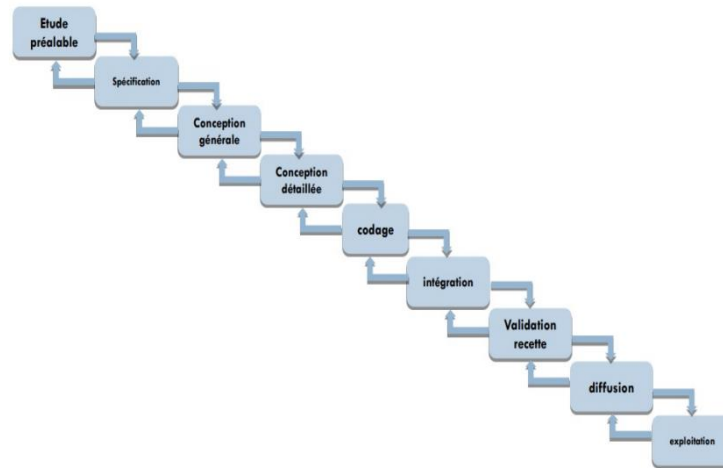


Figure 11: Cycle de vie en cascade

b. Cycle de vie en v

Le modèle du cycle en V est un modèle imaginé suite au problème de réactivité du modèle en cascade.

Il permet, en cas d'anomalie, de limiter un retour aux étapes précédentes. Les phases de la partie montante doivent renvoyer de l'information sur les phases en vis-à-vis lorsque des défauts sont détectés, afin d'améliorer le logiciel. C'est en phase de spécification que l'on se préoccupe des procédures de validation. C'est en phase de conception générale que l'on se préoccupe des procédures d'intégration. C'est en phase de conception détaillée que l'on prépare les tests unitaires.

Il est adapté pour des projets dont le domaine est bien maîtrisé.

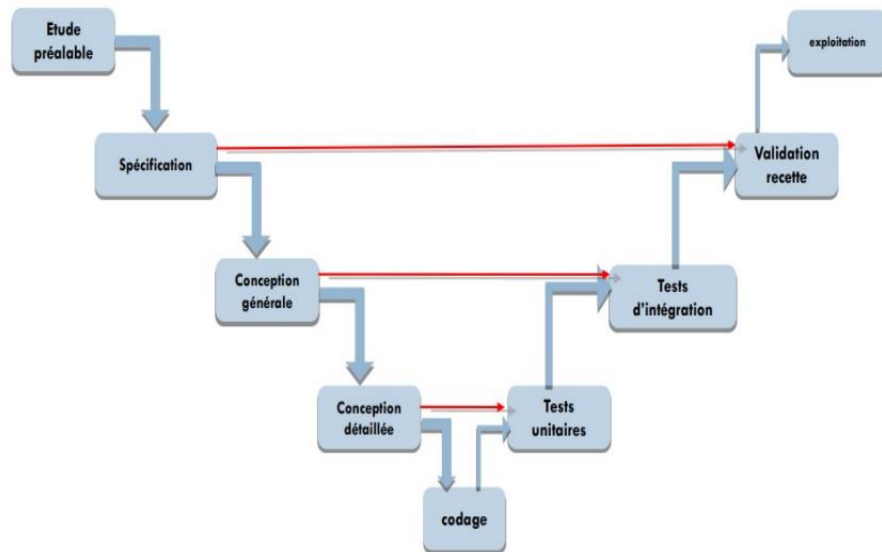


Figure 12: Cycle de vie en v

2.3 Diagramme de Gantt

Le **diagramme de Gantt** est une méthode de représentation graphique qui illustre le calendrier des phases, activités, tâches, et ressources d'un projet. Sur l'axe horizontal, on positionne les jours, semaines ou mois, tandis que l'axe vertical liste les différentes tâches. Chaque tâche est représentée par une barre, dont la longueur correspond à sa durée estimée.

Les tâches peuvent se suivre ou se dérouler simultanément, en totalité ou partiellement. Créé par Henry L. Gantt en 1917, ce diagramme demeure aujourd'hui l'un des outils les plus couramment utilisés pour la gestion de projet.

III. Présentation UML

UML (Unified Modeling Language) est un langage graphique standard de modélisation utilisé pour représenter des systèmes logiciels. Il offre une méthode normalisée pour visualiser la conception d'un système en utilisant différents types de diagrammes.

a. Diagramme de cas d'utilisation

Le **diagramme de cas d'utilisation** est un outil de modélisation essentiel qui illustre les interactions entre les utilisateurs et un système. Il organise les fonctionnalités du système en cas d'utilisation, qui sont des unités logiques décrivant des actions spécifiques que les utilisateurs peuvent réaliser avec le système. Chaque cas d'utilisation capture un besoin utilisateur, offrant une vue centrée sur l'utilisateur plutôt qu'une approche technique.

Composants Principaux :

- **Acteurs** : Les utilisateurs ou autres systèmes qui interagissent avec le système.
- **Cas d'utilisation** : Les fonctionnalités spécifiques du système, pertinentes pour les acteurs.
- **Relations** : Les connexions entre les cas d'utilisation qui montrent comment ils sont liés ou dépendent les uns des autres.

Les diagrammes de cas d'utilisation sont essentiels pour s'assurer que le développement informatique répond précisément aux attentes des utilisateurs. Ils jouent un rôle crucial en garantissant la fonctionnalité et l'efficacité du système final.

b. Diagramme de classe

Un **diagramme de classe** est un type de diagramme structurel utilisé en génie logiciel pour décrire la structure statique d'un système. Il montre les classes du système, leurs propriétés, et les relations entre elles. Cela inclut la visualisation des interactions comme les associations, les héritages, et les dépendances

Composants Principaux :

- ❖ **Classes** : Représentées par des rectangles, les classes

comprennent trois parties :

- **Nom de la Classe** : Le titre de la classe, souvent un nom de substantif.
- **Attributs** : Variables ou propriétés stockées dans la classe.
- **Méthodes** : Fonctions ou procédures que la classe peut exécuter.

❖ **Relations** :

- **Association** : Une connexion générale entre deux classes, représentée par une ligne.
- **Agrégation** : Une forme spéciale d'association qui représente une relation "tout-partie", marquée par un diamant blanc à une extrémité de la ligne.
- **Composition** : Une forme plus forte d'agrégation avec un diamant noir, indiquant une possession exclusive.
- **Héritage** : Représenté par une flèche pointue, indiquant une relation de type "est-un".

- ❖ **Visibilité** : Indique si les attributs ou méthodes sont accessibles depuis d'autres classes, souvent noté par des symboles (+, -, #).

c. Diagramme de séquence

Le **diagramme de séquence** est un outil crucial en génie logiciel, relevant des diagrammes comportementaux et plus spécifiquement des diagrammes d'interactions. Il est utilisé pour représenter les interactions entre les différents objets et acteurs d'un système informatique, montrant comment ils communiquent entre eux au fil du temps.

Composants Principaux :

- ❖ **Lignes de Vie** : Représentées par des lignes verticales qui descendent des entêtes des objets ou acteurs, elles indiquent la présence de l'objet au fil du temps.
- ❖ **Barres d'Activation** : Des rectangles fins sur les lignes de vie qui montrent quand un objet est actif dans le processus.
- ❖ **Messages** : Flèches horizontales qui montrent les interactions entre les objets, avec des annotations pour décrire l'action ou la méthode appelée.

IV. Notre projet de location de voiture

a. Planification






		Nom de la tâche	Durée	Début	Fin	Prédécesseur
1		PFE: App desktop pour la	58 jours?	01/03/2024	21/05/2024	
2		Phase1:vision generale	5 jours?	01/03/2024	07/03/2024	
3		problematique	2 jours	01/03/2024	04/03/2024	
4		Etude de la faisabilite	2 jours	05/03/2024	06/03/2024	3
5		Reunion avec l'encadra	1 jour?	07/03/2024	07/03/2024	4
6		phase2:Analyse	4 jours	08/03/2024	13/03/2024	2
7		Annonce des besoins	3 jours	08/03/2024	12/03/2024	
8		Reunion avec l'encadra	1 jour	13/03/2024	13/03/2024	7
9		phase3:Conception	14 jours?	14/03/2024	02/04/2024	6
10		diagramme de cas d'uti	7 jours	14/03/2024	22/03/2024	
11		diagramme de classe	3 jours	25/03/2024	27/03/2024	10
12		diagramme de sequenc	3 jours	28/03/2024	01/04/2024	11
13		Reunion avec l'encadra	1 jour?	02/04/2024	02/04/2024	12
14		phase4:Realisation	26 jours?	03/04/2024	08/05/2024	9
15		apprendre c#	10 jours	03/04/2024	16/04/2024	
16		apprendre WPF	10 jours	03/04/2024	16/04/2024	
17		apprendre .NET	10 jours	03/04/2024	16/04/2024	
18		apprendre Mysql	4 jours	03/04/2024	08/04/2024	
19		apprendre MVVM	4 jours	03/04/2024	08/04/2024	
20		Realisation de la base c	3 jours	09/04/2024	11/04/2024	18
21		Realisation d'applicatio	15 jours	17/04/2024	07/05/2024	15;16;17;18;1
22		Reunion avec l'encadra	1 jour?	08/05/2024	08/05/2024	21
23		phase5:Finalisation du p	6 jours?	09/05/2024	16/05/2024	14
24		Test d'application	1 jour	09/05/2024	10/05/2024	
25		Reunion avec l'encadra	1 jour?	13/05/2024	13/05/2024	24
26		La mise a jours de l'app	1 jour	13/05/2024	15/05/2024	24
27		Reunion avec l'encadra	1 jour?	16/05/2024	16/05/2024	26
28		Mise en oeuvre	3 jours?	17/05/2024	21/05/2024	23
29		Redaction du rapport	2 jours	17/05/2024	20/05/2024	
30		Redaction du presentat	2 jours	17/05/2024	20/05/2024	
31		Reunion avec l'encadra	1 jour?	21/05/2024	21/05/2024	29;30
32		Le rapport final	1 jour?	21/05/2024	21/05/2024	29
33		Presentation final	1 jour?	21/05/2024	21/05/2024	30
34		Soutenance	0 jour	21/05/2024	21/05/2024	33

Figure 13 : Planification

b. Cycle de vie en V

Nous avons choisi le cycle en V. Car ce modèle est caractérisé par le parallélisme, dans ce modèle verticalement nous trouvons les étapes du développement et horizontalement la vérification.

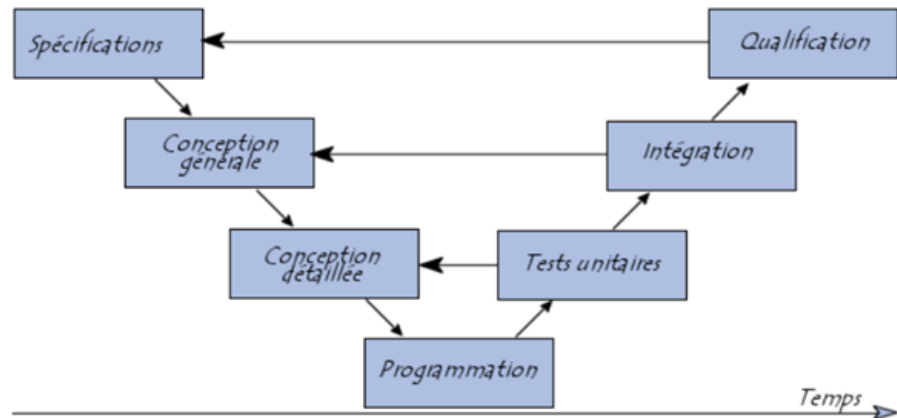


Figure 14: Cycle de vie en v

c. Diagramme de Gantt

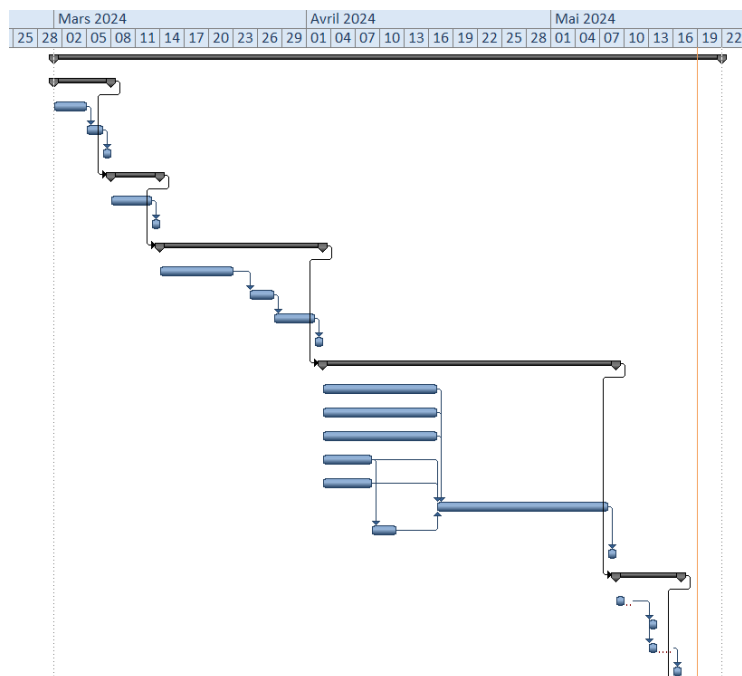


Figure 15 : Diagramme de GANTT partie1

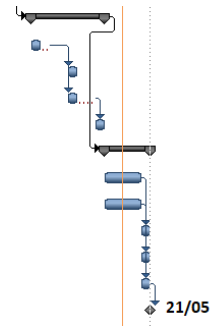


Figure 16: Diagramme de GANTT partie2

d. Modélisation

1) Les acteurs

Identification des acteurs et leur rôle :

Tableau 2: table des acteurs

Acteurs	Rôle
Administrateur	<ul style="list-style-type: none"> Gérer les voitures Gérer les employées
Employé	<ul style="list-style-type: none"> Gérer les clients Gérer les réservations Gérer les paiements
Client	<ul style="list-style-type: none"> Voir les voitures disponibles Faire une réservation

2) Diagramme de cas d'utilisation

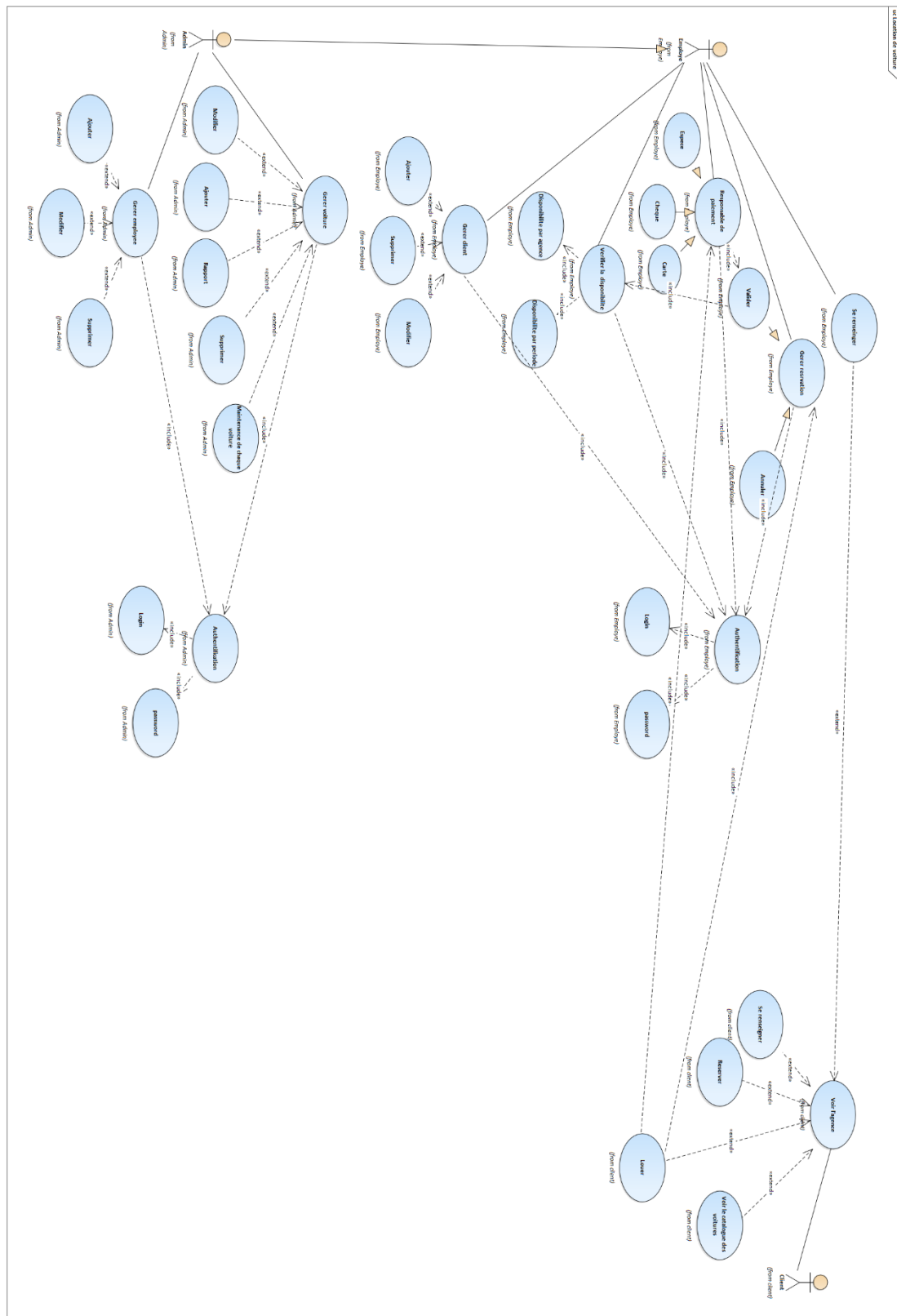


Figure 17 : Diagramme de cas d'utilisation

3) Diagramme de classe

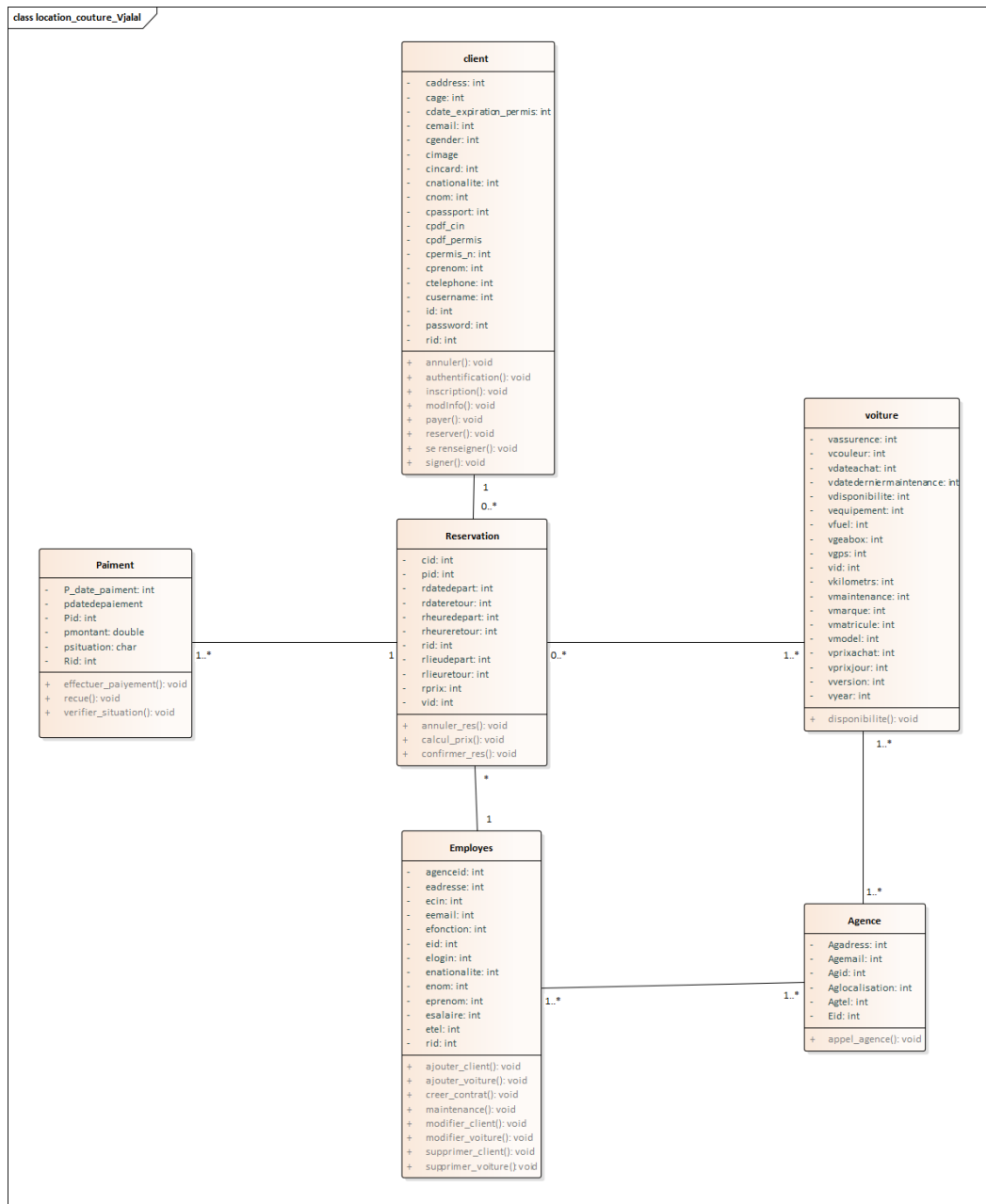
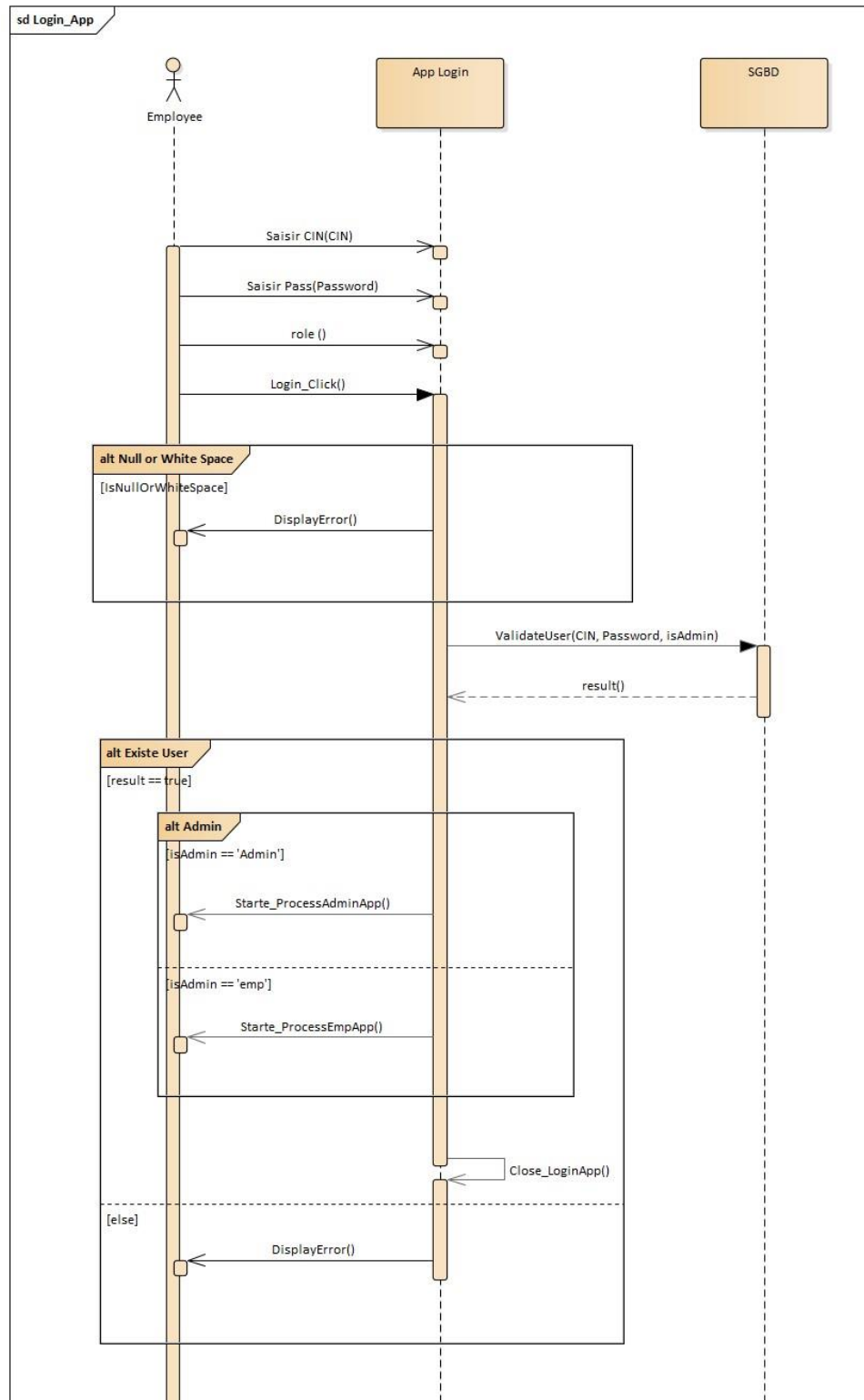


Figure 18 :Diagramme de classe

4) Diagramme de séquence

Pour login



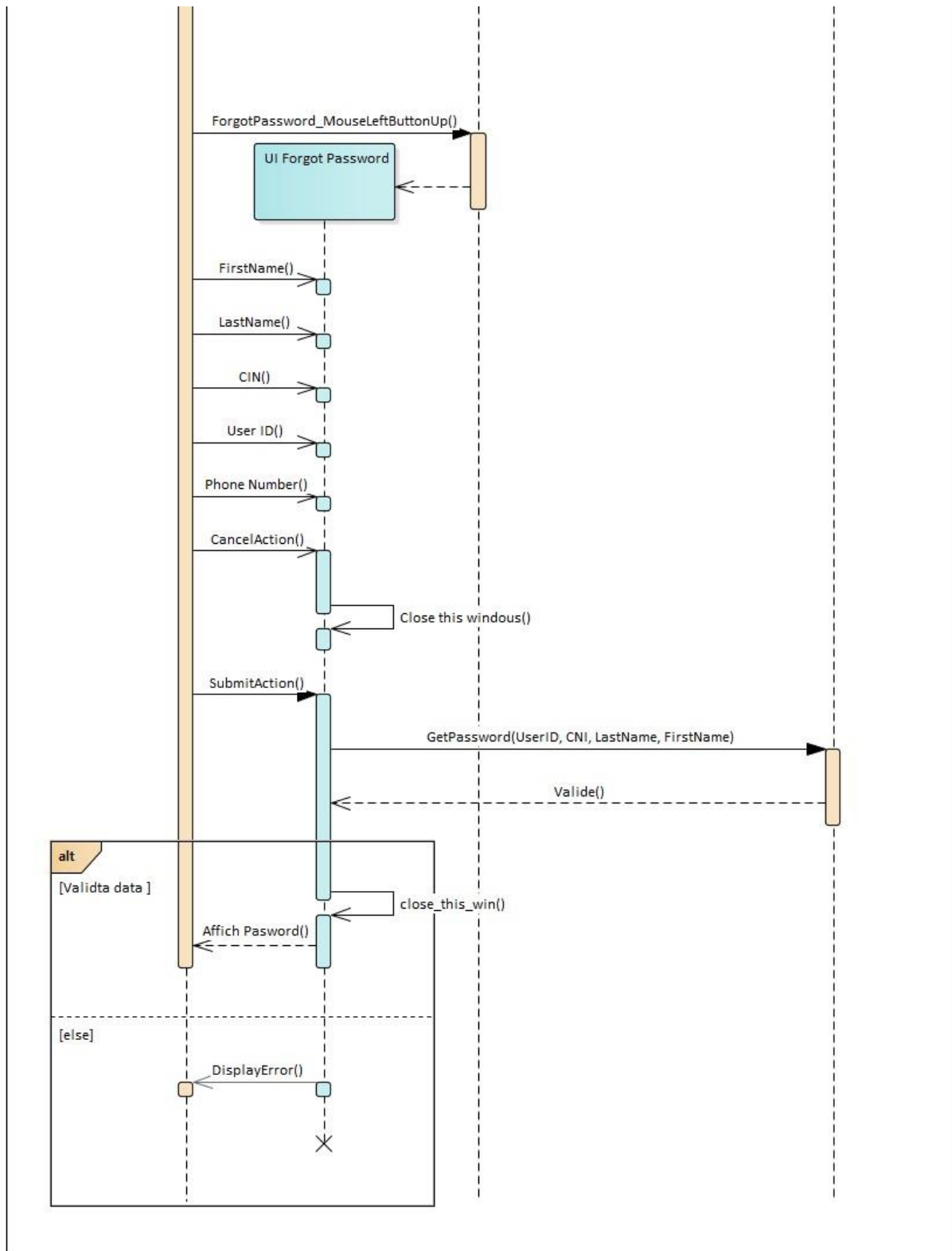
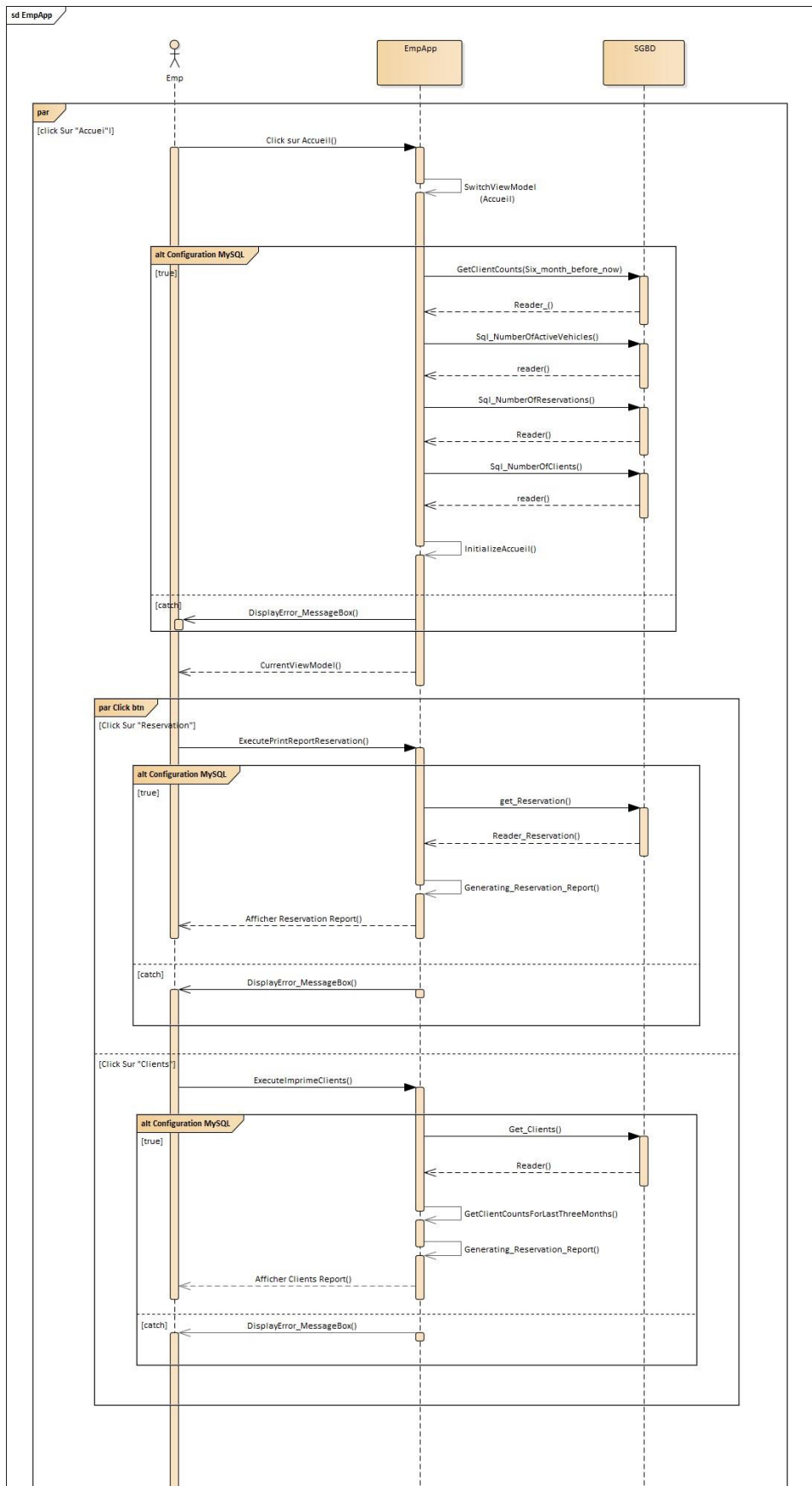
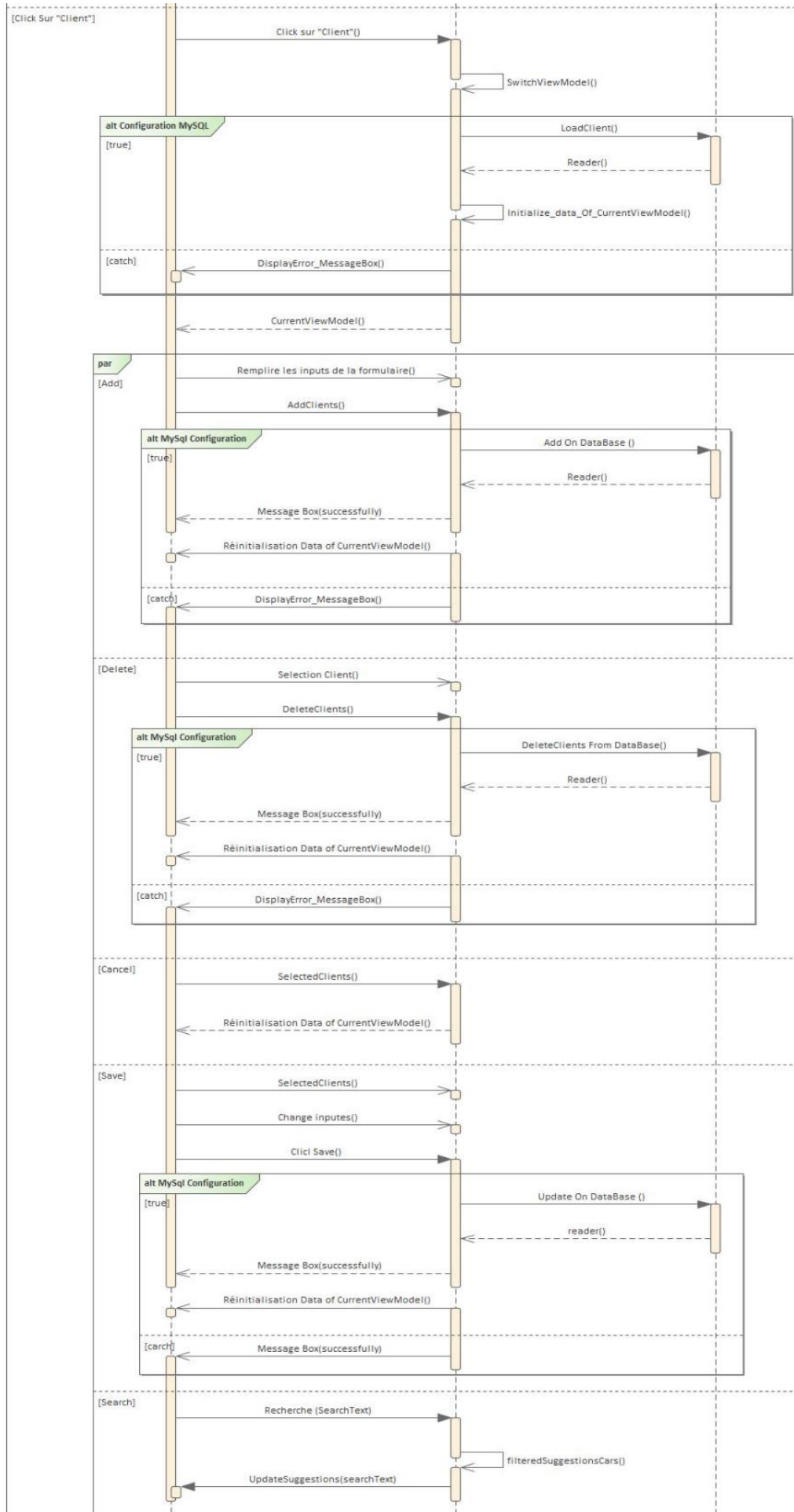
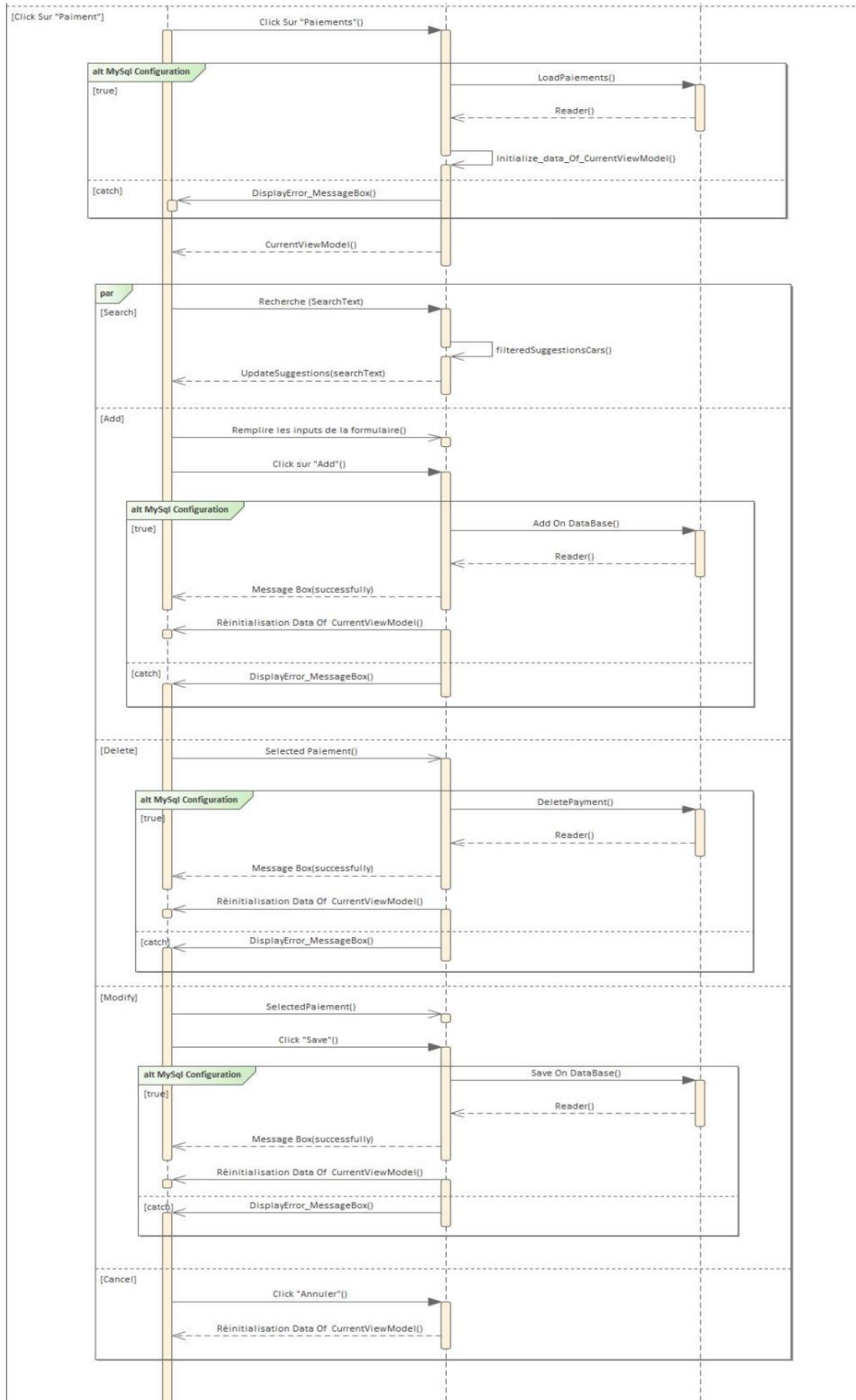


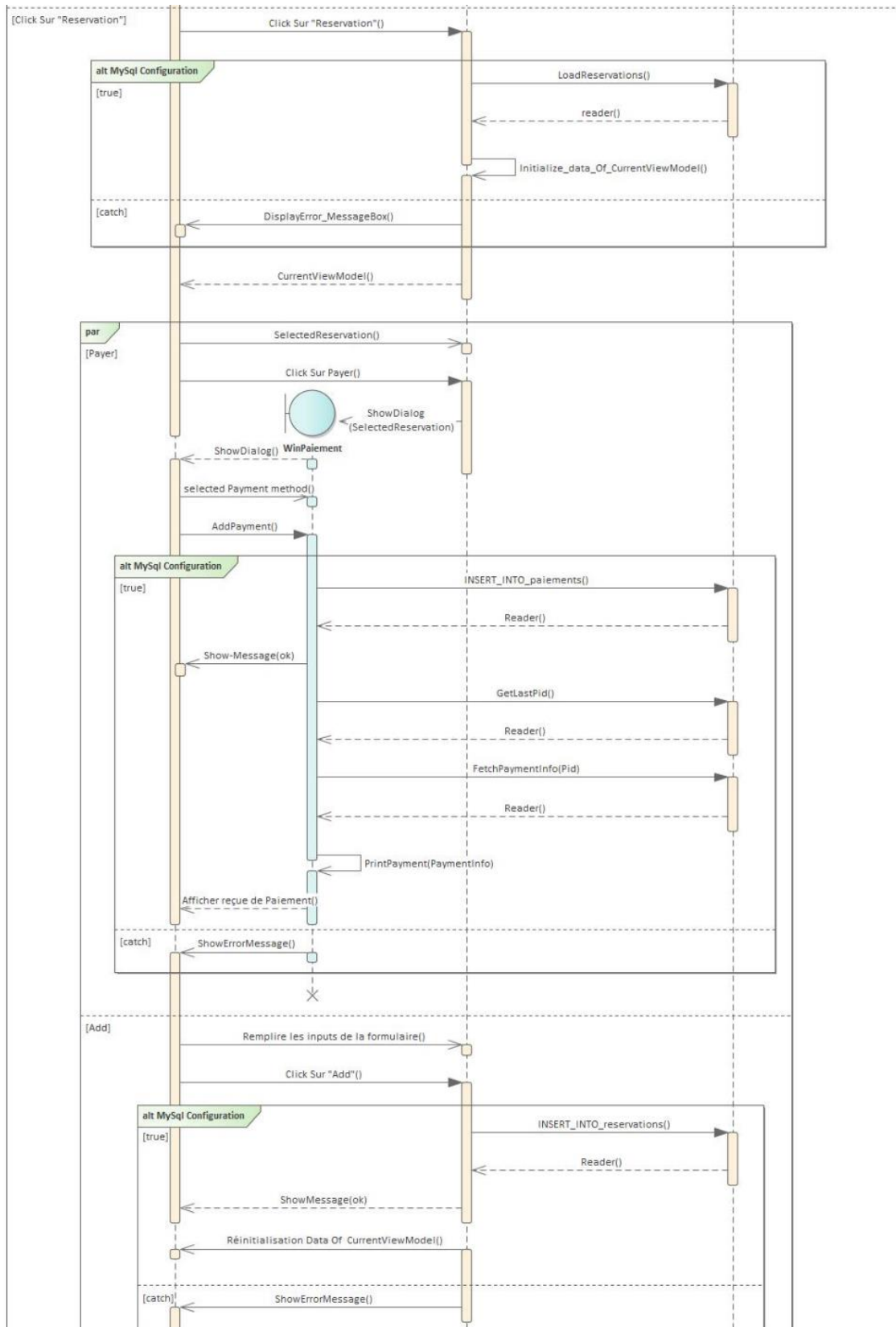
Figure 19 : diagramme de séquence(login) Pages : [27-28]

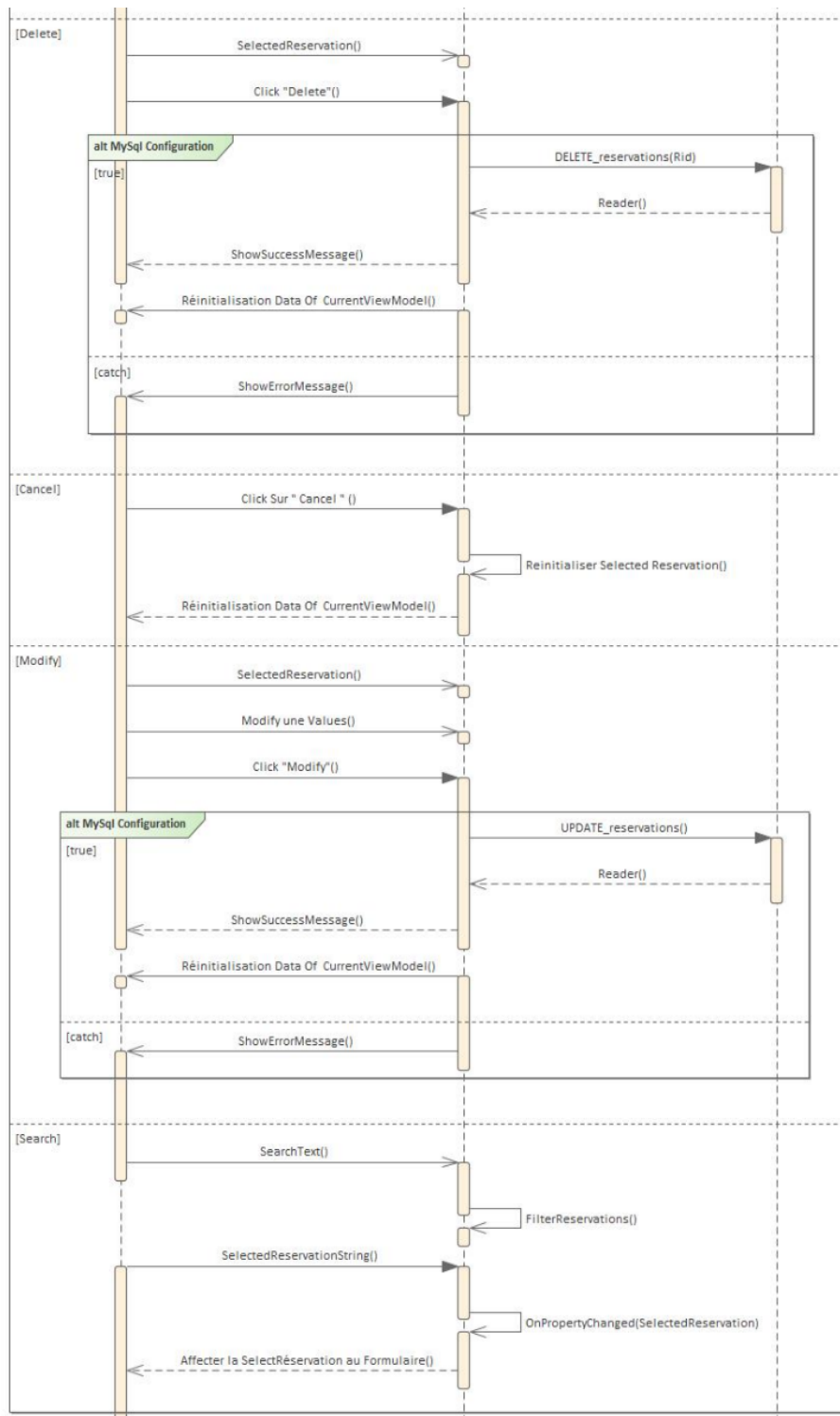
Pour employer











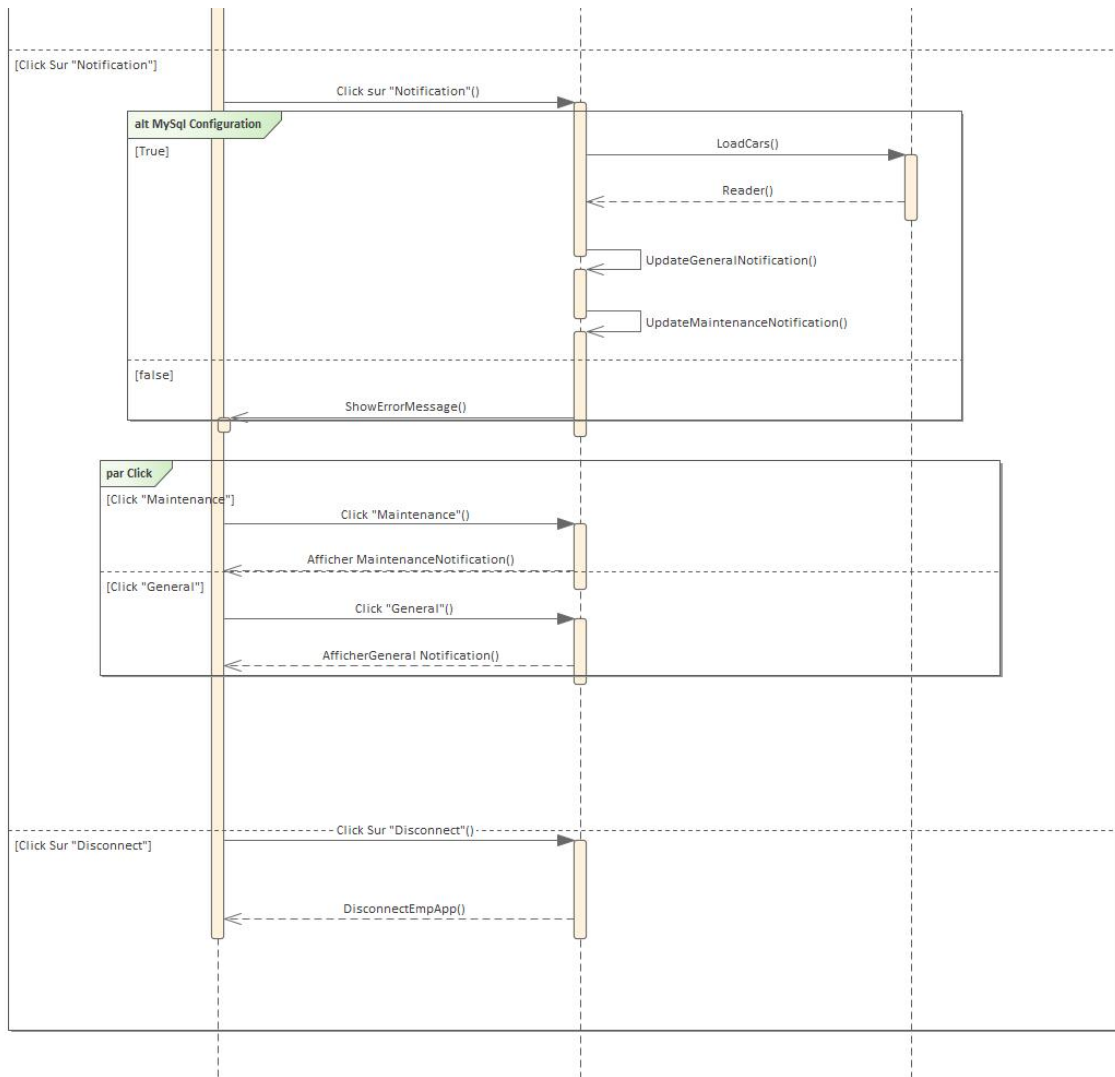
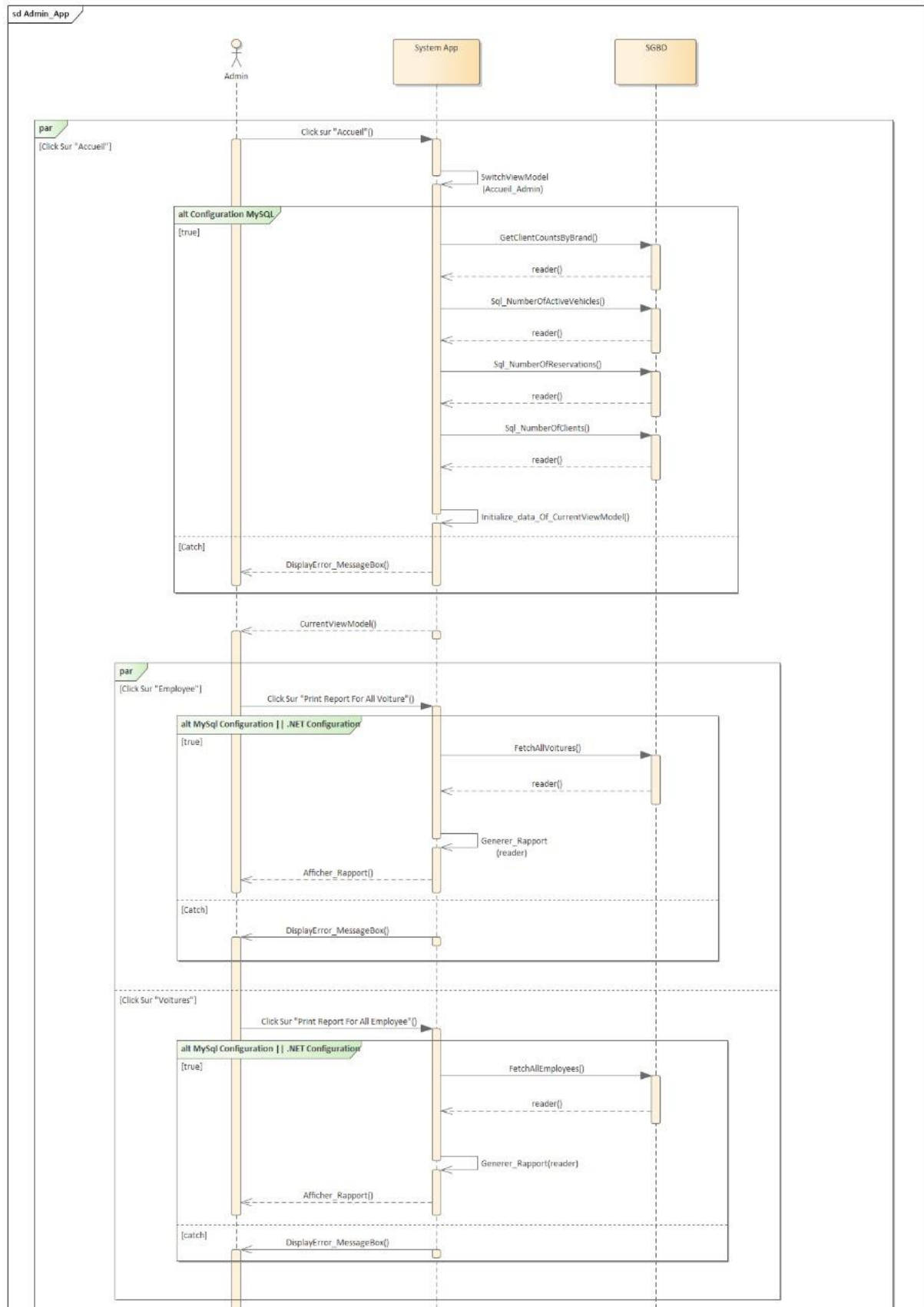
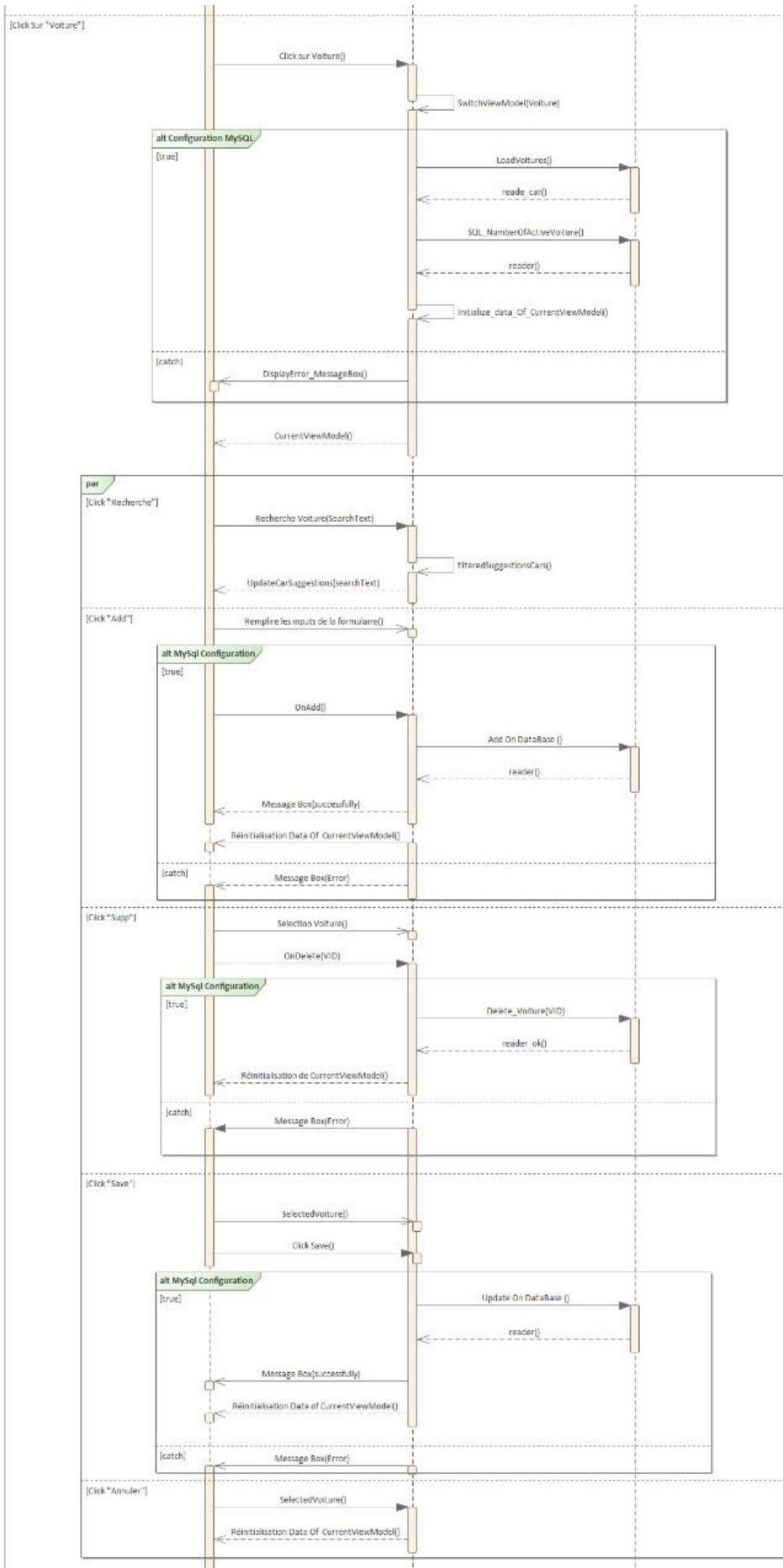
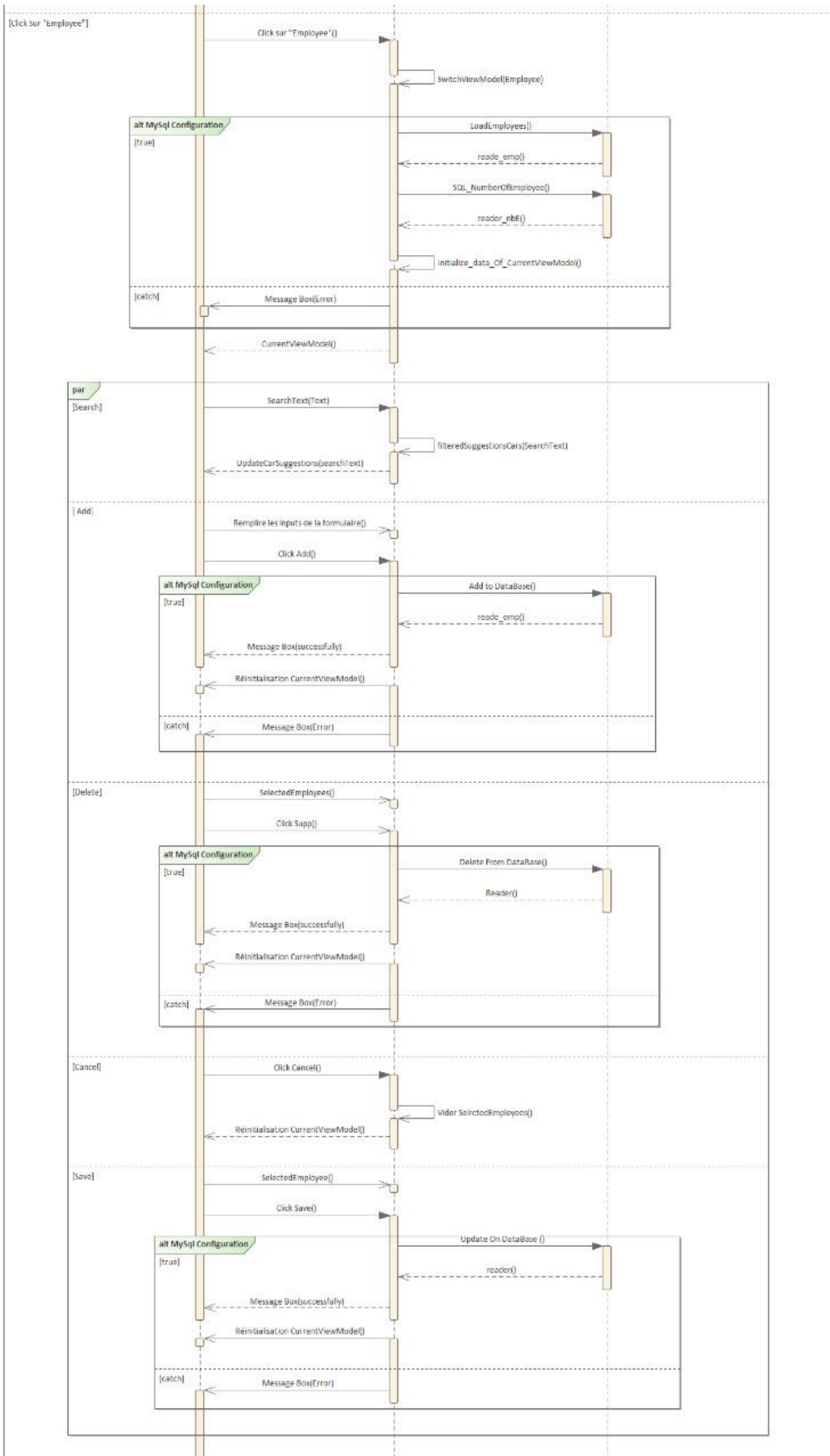


Figure 20: diagramme de séquence (employer) Pages : [29-34]

Pour l'administrateur







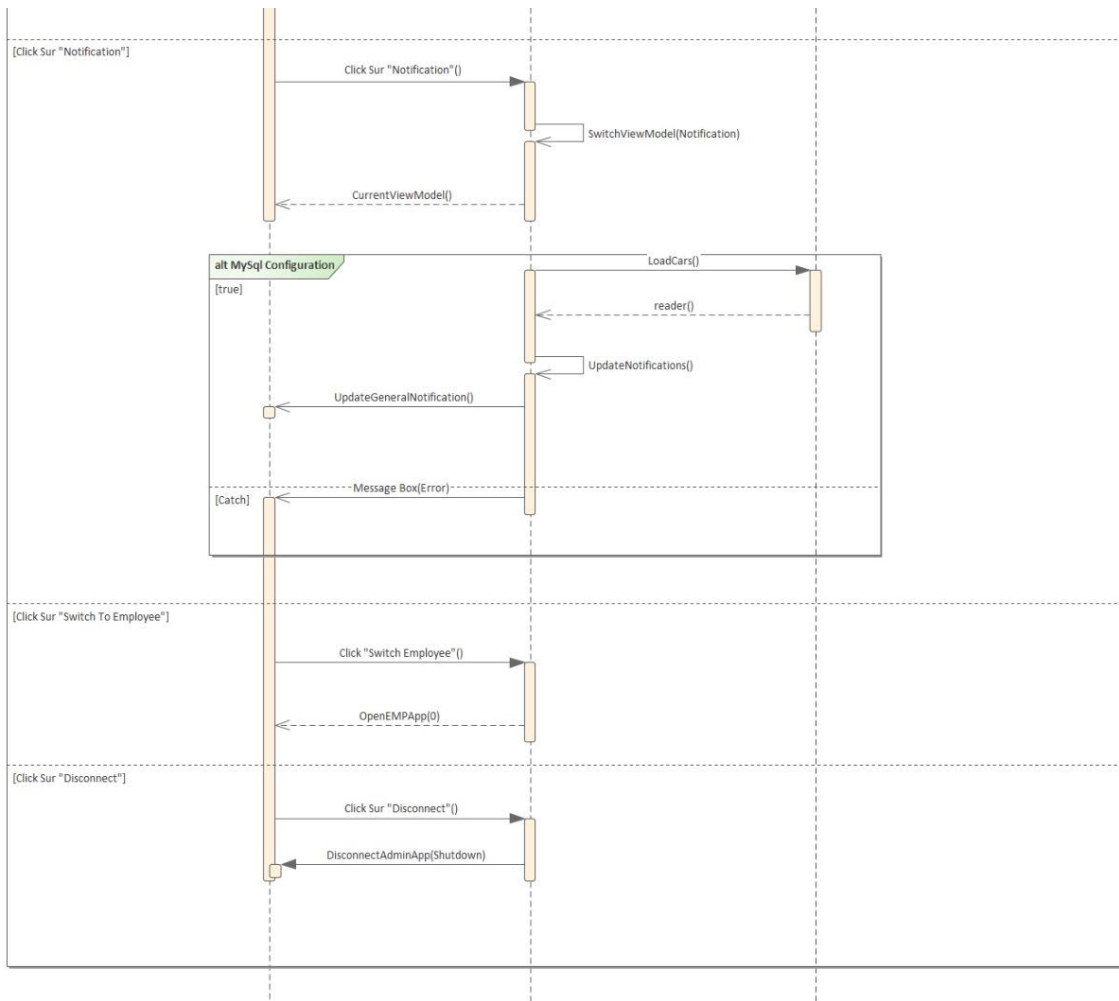


Figure 21 : diagramme de séquence (administrateur) Pages : [35-38]

V. Conclusion

Dans ce chapitre, nous avons développé les divers composants de notre système. Le chapitre suivant sera consacré à la mise en œuvre de notre modèle.

Chapitre 3 : Réalisation du projet

I. Introduction

La phase de réalisation est une étape très importante dans le cycle de vie de nos applications, cette phase permet de concrétiser notre projet par le développement des interfaces et par des réalisations concrètes des fonctionnalités du système. Pour réaliser ces applications nous avons en recours à plusieurs outils de développement. Dans cette dernière partie on va présenter le résultat final de notre application.

II. Application Desktop

2.1 Les interfaces et les explications

1) Authentification

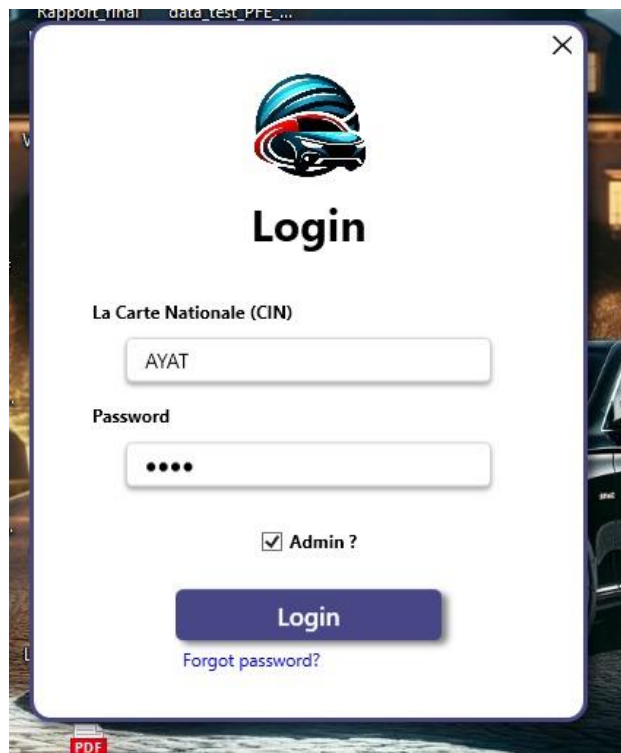
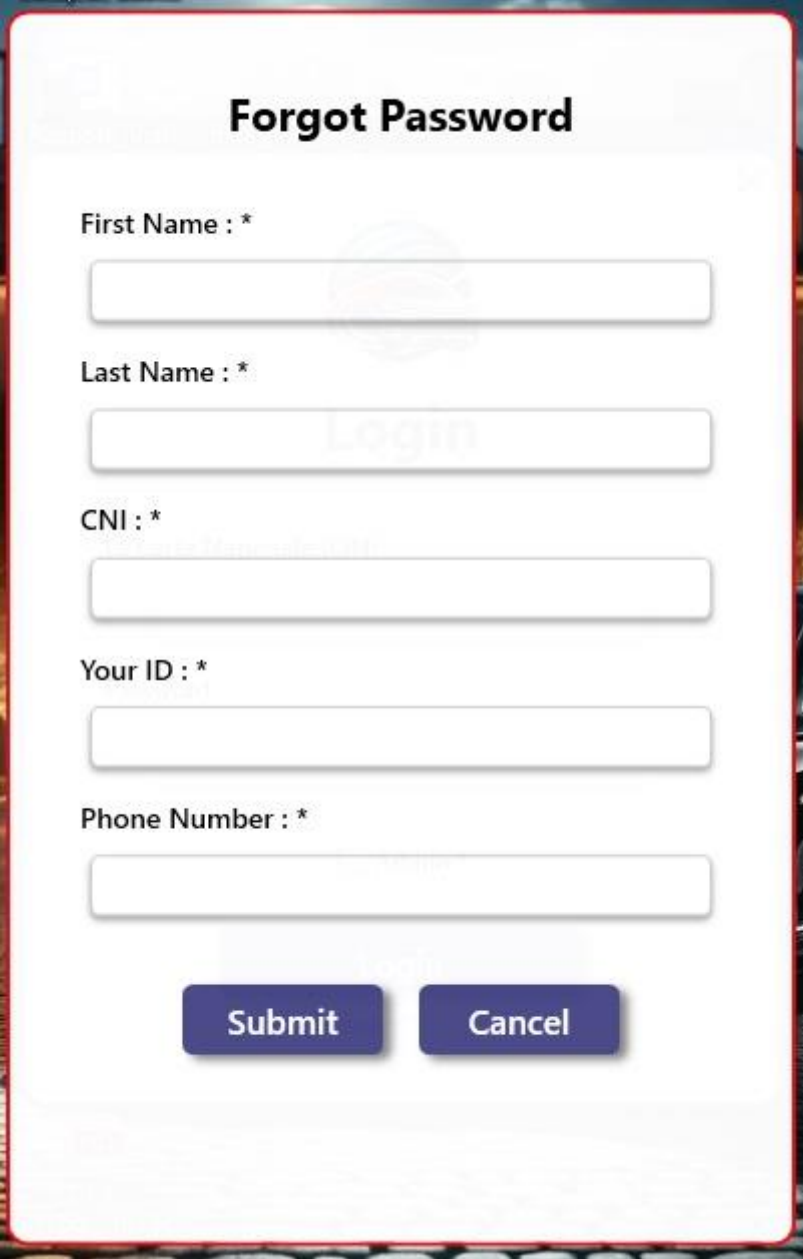


Figure 22 : login

L'interface d'authentification constitue la première étape d'interaction entre l'utilisateur et notre application de bureau.

2) Interface Forget password



Forgot Password

First Name : *

Last Name : *

CNI : *

Your ID : *

Phone Number : *

Submit Cancel

Figure 23 : forget password

L'interface de récupération de mot de passe est une partie essentielle de notre application de bureau. Elle aide les utilisateurs à récupérer leur accès s'ils oublient leurs identifiants.

3) Les interfaces d'administrateur

a. Interface accueil

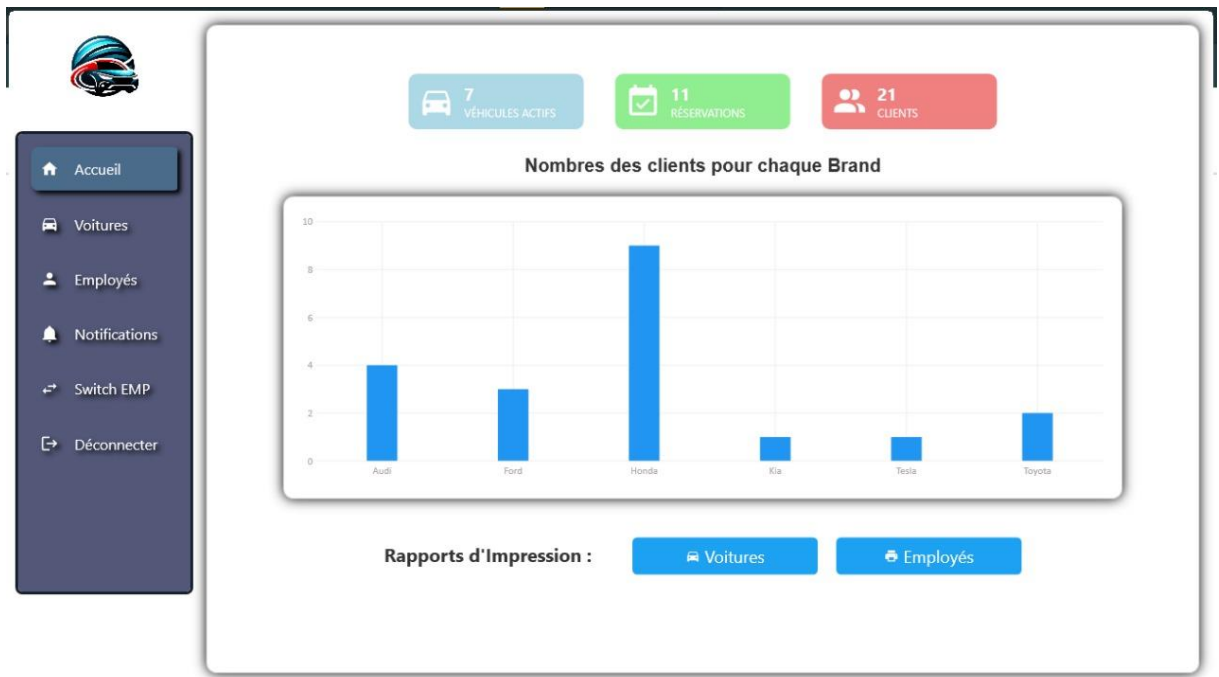


Figure 24 : interface d'accueil pour l'administrateur

L'interface d'accueil pour l'administrateur fournit un aperçu complet des véhicules sortis, des réservations effectuées et du nombre de clients enregistrés dans la base de données. Elle présente des graphiques illustrant le nombre de clients par marque, ainsi que divers indicateurs clés. De plus, elle offre des options pour imprimer des rapports sur les voitures et les employés.

b. Interface voiture

19
Number total des Voiture

ID	Marque	Couleur	Disponibilité	Maintenance	Assuree	Prix Jour (DH)
11	AAA	red	✓ True	5/8/2024 12:00:00 AM	True	30.00
12	Toyota	Red	✗ False	1/1/2023 12:00:00 AM	True	35.00
13	Honda	Blue	✓ True	2/20/2023 12:00:00 AM	True	30.00
14	Ford	Green	✓ True	3/10/2023 12:00:00 AM	False	40.00
15	Chevrolet	Black	✗ False	4/1/2023 12:00:00 AM	True	25.00
22	Toyota	Rouge	✓ True	9/1/2022 12:00:00 AM	True	15000.00
23	Honda	Bleu	✓ True	2/15/2023 12:00:00 AM	False	19000.00
24	Ford	Noir	✗ False	6/1/2021 12:00:00 AM	True	22000.00
25	Tesla	Blanc	✓ True	11/15/2022 12:00:00 AM	True	42000.00
26	Toyota	White	✗ False	7/15/2021 12:00:00 AM	False	50.00
27	Ford	Blue	✗ False	6/10/2022 12:00:00 AM	False	45.00
28	Honda	Red	✗ False	5/20/2022 12:00:00 AM	False	40.00
29	Chevrolet	Black	✗ False	1/30/2023 12:00:00 AM	False	55.00
30	Nissan	Grey	✗ False	11/25/2021 12:00:00 AM	False	50.00
31	Volkswagen	Silver	✗ False	4/15/2022 12:00:00 AM	False	47.00
32	Audi	Pearl	✗ False	3/20/2022 12:00:00 AM	False	52.00
33	BMW	Blue	✗ False	7/7/2023 12:00:00 AM	False	60.00

Form Fields:

- Prix Achat:** 0
- Date Achat:** 12/05/2024
- Prix par Jour:** 0
- Kilométrage par Jour:** 0
- Équipement:** ☐ GPS
- Type de Carburant:**
- Assurance:** ☐ Assurée
- Maintenance:**
- Dernière Maintenance:** 12/05/2024
- Disponibilité:** ☐ Disponible

Buttons: Annuler, Enregistrer, + Ajouter, Supprimer

Figure 25 : interface des voitures pour l'administrateur

L'interface des voitures pour l'administrateur affiche le nombre total de véhicules enregistrés dans la base de données. Elle propose également des fonctionnalités permettant d'ajouter une nouvelle voiture, ainsi que de modifier ou supprimer les entrées existantes, facilitant ainsi la gestion complète des voitures.

c. Interface employée

The interface displays a list of employees with the following data:

ID	CNE	Agency Id	Last Name	Téléphone	Role	Salary
1	ABC23	1	Smith	555-1234	Manager	50000.00
2	XYZ789	1	Doe	555-5678	Developer	70000.00
3	LMN456	2	Brown	555-9012	Sales	45000.00
15	HHHHHH	4	KKK	9876545678	OILUYT	330.00
16	AYAT	10	AYAT	1234	UX/UI	900000.00
17	11111Z	10	ALALAL	123	AZA	234560.00
18	jalal	0	JJJJ	9999999	JJJJ	9876540.00
19	AA	0	jalal	88	JJ	880.00
20	ABC123456	1	Doe	555-1234	Manager	50000.00
49	XYZ789101	2	Smith	555-5678	Engineer	70000.00
50	EEG111213	1	Brown	555-6789	Technician	48000.00
51	HUI141516	2	Johnson	555-2345	Developer	90000.00
52	KLM171819	1	Lee	555-3456	Designer	75000.00
53	NOP20212	2	Garcia	555-4567	Architect	65000.00
54	QRS232425	1	Renault	555-5678	Consultant	55000.00
55	TUV262728	2	Miyazaki	555-6789	Project Manager	80000.00

The right-hand form includes fields for: Prénom, Nom, Nationalité, Adresse, Agence ID (0), CIN, Email, Username, Password, Fonction Admin (Admin checkbox), Salaire (0.00), and buttons for Annuler, Enregistrer, + Ajouter, and Delete.

Figure 26 : interface des employés pour l'administrateur

L'interface des employés pour l'administrateur affiche le nombre total d'employés enregistrés dans la base de données. Elle propose également des fonctionnalités permettant d'ajouter un nouvel employé, ainsi que de modifier ou supprimer les entrées existantes, facilitant ainsi la gestion complète du personnel.

d. Interface notification

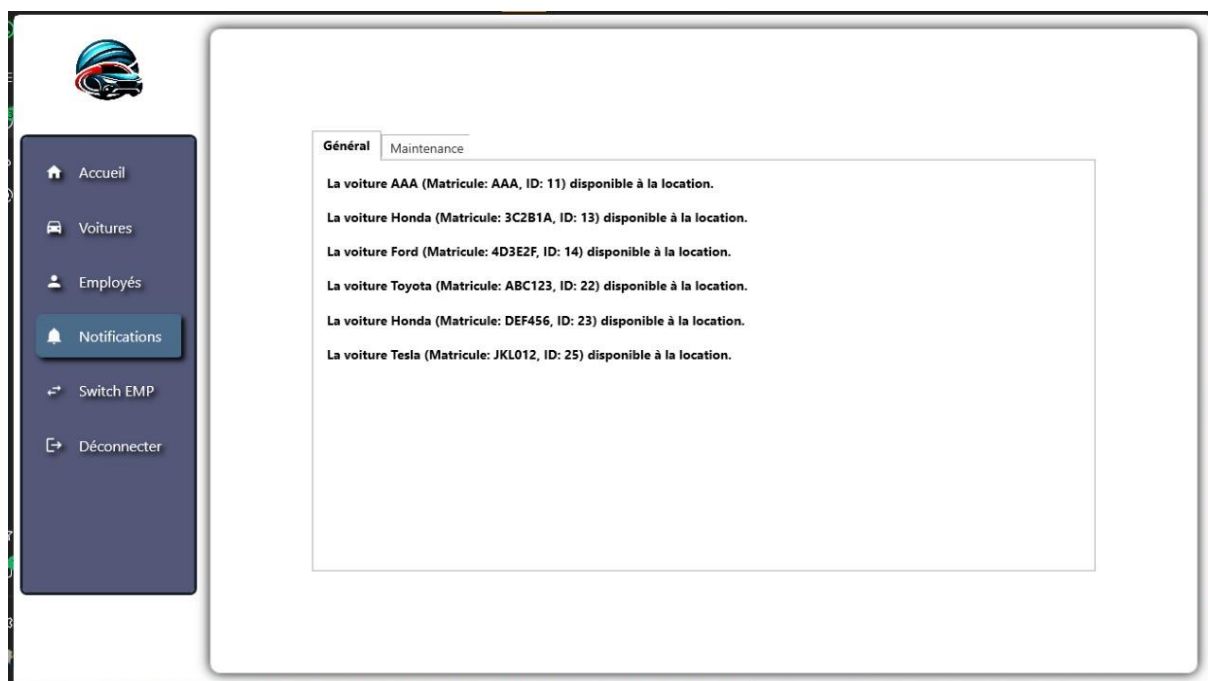


Figure 27 : interface de notification pour l'administrateur

L'interface de notification indique, sous l'onglet "Général", les véhicules disponibles pour la location. Dans l'onglet "Maintenance", elle affiche des alertes pour les véhicules qui requièrent une maintenance. Cette fonctionnalité assure une gestion efficace du parc automobile en offrant des mises à jour en temps réel sur l'état des véhicules, facilitant ainsi l'organisation de la location et de la maintenance des voitures.

4) Les interfaces d'employée

a. Interface accueil

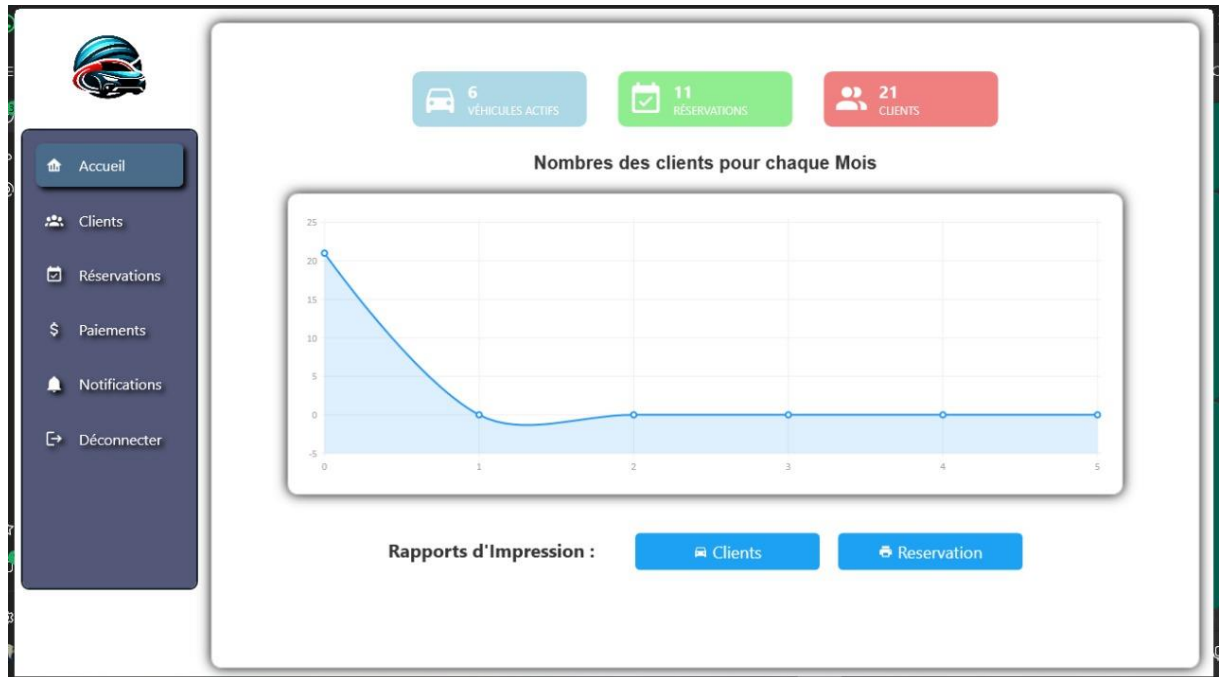


Figure 28 : interface d'accueil pour l'employé

L'interface d'accueil pour l'employé présente un résumé détaillé des réservations effectuées et du total de clients enregistrés dans la base de données. Un graphique y illustre le nombre de clients ayant loué une ou plusieurs voitures chaque mois, accompagné d'autres indicateurs clés. De plus, cette interface offre la possibilité d'imprimer des rapports approfondis sur les clients et les réservations.

b. Interface client

Navigation Sidebar:

- Accueil
- Clients
- Réservations
- Paiements
- Notifications
- Déconnecter

Client Data Table:

ID	CNE	UserName	Téléphone	Reservation ID	Permis
40	111	11111	11	10	111
82	D123A456	john doe	555-1234	2	P9875
83	DA654321	jane smith	555-5678	2	P1345
84	AD234567	alice johnson	555-6789	3	P545321
85	D345678	bob brown	555-7890	5	P43210
86	SD456789	charlie davis	555-8901	6	P321109
87	DD567890	diana evans	555-9012	1	P210198
88	D678901D	ethan frank	555-0123	11	PF10987
89	D789012	grace hall	555-1234	4	PG9876
90	D8901D23	harrying ram	555-2345	7	GP98765
91	D90D1234	isla jennings	555-3456	7	PG87654
92	D012D345	jack knight	555-4567	8	P7654G3
93	D12DS3456	lily lawson	555-5678	10	P6543G2
94	D234SS567	mason murray	555-6789	1	P5G4321
95	D34567Q8	nora nash	555-7890	2	P4321G0
96	D4567Q89	oscar owen	555-8901	2	P3G2109
97	D56Q7890	piper perry	555-9012	4	GP21098
98	D6F78901	quinn quincy	555-0123	6	P1G0987

Client Details Form:

Left Column:

- Change Photo
- PDF CIN / Pasport
- afficher CIN / Passeport
- PDF Permis
- afficher Permis
- CIN: []
- ID: [0]
- Username: []
- Password: []
- Reservation ID: [0]

Right Column:

- Prenom: []
- Nom: []
- Adresse: []
- Téléphone: []
- Email: []
- Passport: []
- Age: [0]
- Nationale: []
- Permis: []
- DatePermis: [1/1/0001 12:00:00 AM]

Actions:

- Annuler
- Enregistrer
- + Ajouter
- Delete

Footer Note: NB : pour trier avec un champ . clics sur le champ , (exemple . Id clics sur ID)

Figure 29 : interface client pour l'employé

L'interface client affiche le nombre total de clients enregistrés dans la base de données. Elle offre également des options pour ajouter, modifier ou supprimer des clients, facilitant ainsi la gestion efficace de la clientèle. Cette interface centralise les informations client, améliorant l'organisation et la communication au sein de l'entreprise.

c. Interface réservation

Détails de la Réservation

Réservation ID	Durée (jours)	Prix (MAD)	Date de départ	Lieu de départ
1	5	150.00	6/1/2024 12:00:00 AM	Paris
2	3	100.00	6/3/2024 12:00:00 AM	Lyon
3	7	230.00	6/10/2024 12:00:00 AM	Marseille
4	2	80.00	7/1/2024 12:00:00 AM	Toulouse
5	10	300.00	7/15/2024 12:00:00 AM	Nice
6	6	180.00	8/1/2024 12:00:00 AM	Nantes
7	14	420.00	8/10/2024 12:00:00 AM	Strasbourg
8	4	120.00	9/1/2024 12:00:00 AM	Bordeaux
9	1	50.00	10/1/2024 12:00:00 AM	Lille
10	21	630.00	10/15/2024 12:00:00 AM	Rennes
11	10	120.00	1/1/0001 12:00:00 AM	fes

Détails de la Réservation

Client ID:

Voiture ID:

Réservation ID:

Durée (jours):

Prix (MAD):

Date de départ:

Date de retour:

Heure de départ:

Heure de retour:

Lieu de départ:

Lieu de retour:

NB : pour trier avec un champ , clics sur le champ , (exemple Id clics sur ID)

Figure 30 : interface de réservation pour l'employé

L'interface de réservation permet d'ajouter, de modifier ou de supprimer des réservations, simplifiant ainsi leur gestion. Elle centralise toutes les informations relatives aux réservations, ce qui contribue à améliorer l'organisation et la communication dans l'entreprise.

d. Interface paiements

The interface consists of a sidebar on the left with the following menu items: Accueil, Clients, Réservations, **Palements** (highlighted), Notifications, and Déconnecter. The main content area displays a table of payments with the following data:

Payment ID	Réservation ID	Amount	Méthode	Date	Situation
46	11	120.00	Credit Card	5/12/2024 12:00:00 AM	Payé

Below the table, a note states: **NB : pour trier avec un champ , clics sur le champs , (exemple Id clics sur ID)**. To the right of the table is a form for adding or editing payments, with fields for: ID Paiement (0), ID réservation (0), Méthode de paiement (dropdown), Montant (0), and Date de paiement (01/01/0001). At the bottom of the form are four buttons: Annuler, Enregistrer, + Ajouter, and Delete.

Figure 31 : interface de paiements pour l'employé

L'interface de paiements permet d'ajouter, d'enregistrer, ou de supprimer des paiements.

e. Interface notification

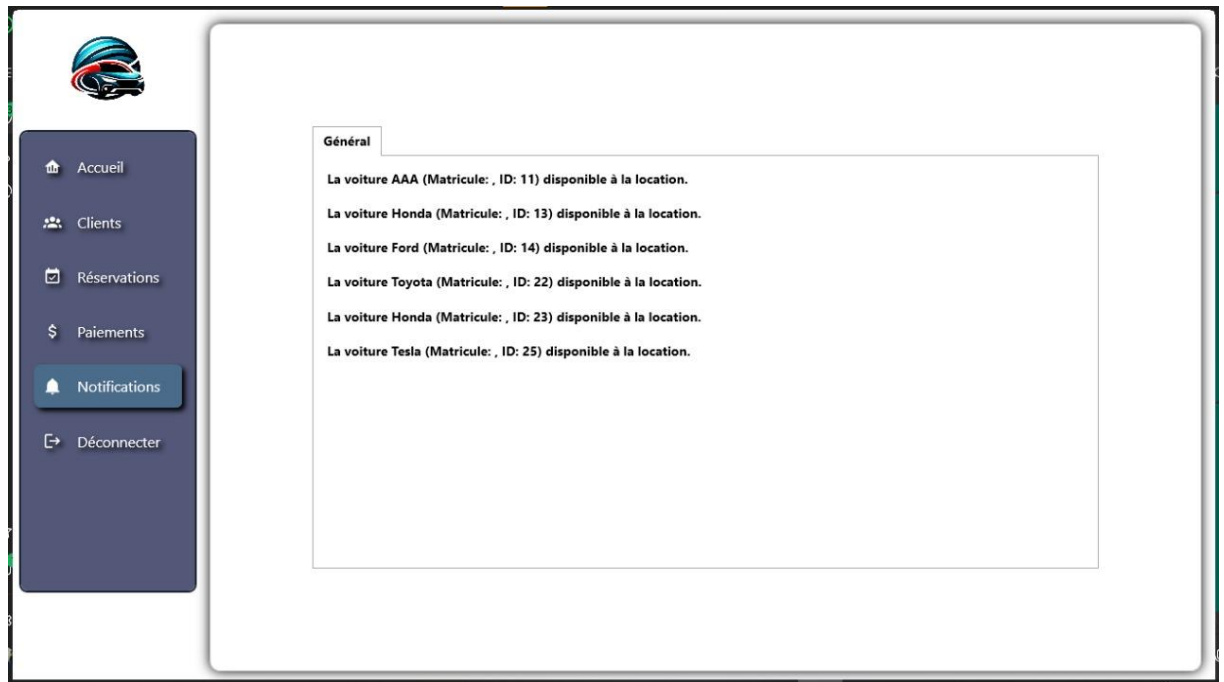


Figure 32 : interface de notification pour l'employé

L'interface de notification indique, sous l'onglet "Général", les véhicules disponibles pour la location.

Conclusion Générale

Ce projet constitue une étape cruciale de notre parcours de formation, offrant une excellente opportunité pour mettre en pratique des connaissances théoriques préalablement acquises tout en nous permettant de développer de nouvelles compétences techniques. Pour mener à bien ce projet, nous avons élaboré un plan détaillé, facilitant l'organisation de notre temps limité et optimisant notre efficacité.

Parallèlement, nous avons découvert l'importance cruciale de la recherche et de la communication dans l'accès à des informations pertinentes, ainsi que la gestion du temps et la planification des tâches pour une exécution fluide des travaux. Grâce à un environnement de travail propice et à une coordination efficace, nous avons réussi à achever le projet conformément au cahier des charges, tout en y ajoutant des fonctionnalités supplémentaires pour en augmenter l'efficacité et l'attrait.

Bien que notre projet réponde déjà à tous les besoins énoncés dans le cahier des charges, nous envisageons d'apporter des améliorations futures pour améliorer la performance de l'application.

Les résultats obtenus jusqu'à présent sont prometteurs et nous motivent à poursuivre le développement de ce projet.

Webographie

MySQL : <https://www.w3schools.com/>

MSIX : <https://youtu.be/4t2TI8ImwMY>

C# WPF :

<https://www.youtube.com/watch?v=t9ivUosw-il&list=PLih2KERbY1HHOOJ2C6FOrVXIwg4AZ-hk1>

WPF MVVM :

<https://www.youtube.com/watch?v=fZxZswmC-BY&list=PLA8ZIAM2Io3hS41Fy4vFpRw8AdYNBXmNm>

<https://learn.microsoft.com/en-us/dotnet/csharp/>

<https://learn.microsoft.com/fr-fr/dotnet/framework/get-started/overview>

https://fr.wikipedia.org/wiki/Microsoft_.NET

<https://learn.microsoft.com/fr-fr/dotnet/architecture/maui/mvvm>