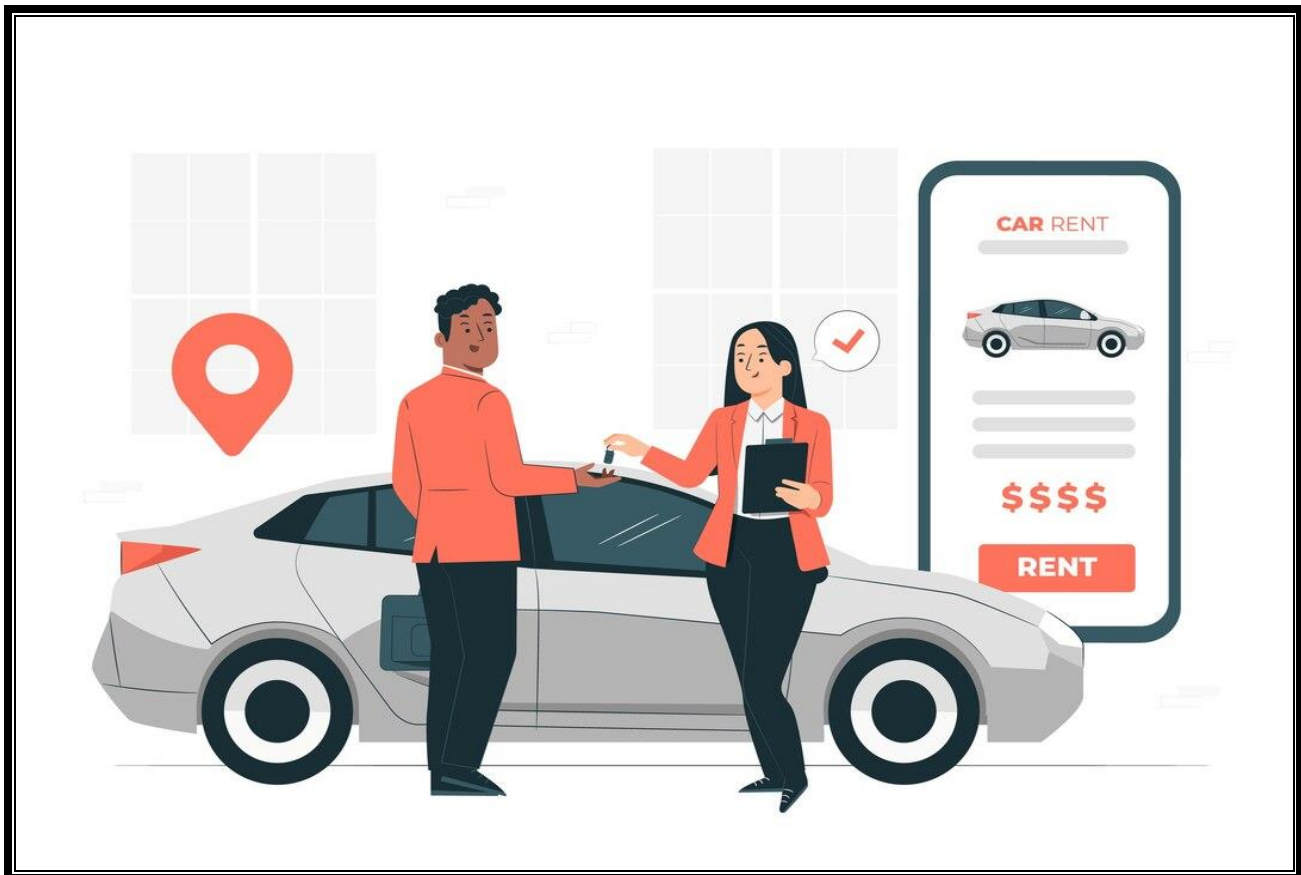




## Rapport de Projet du Gestion de location du voiture



- *Réalisé par :*

Mazin Ali Ahmed

Tariq haoudi

- *Encadré par :*

Pr. Sanae KHALI ISSA

# **Sommaire :**

I / Introduction

II/ Présentation de l'application

III/ Conception

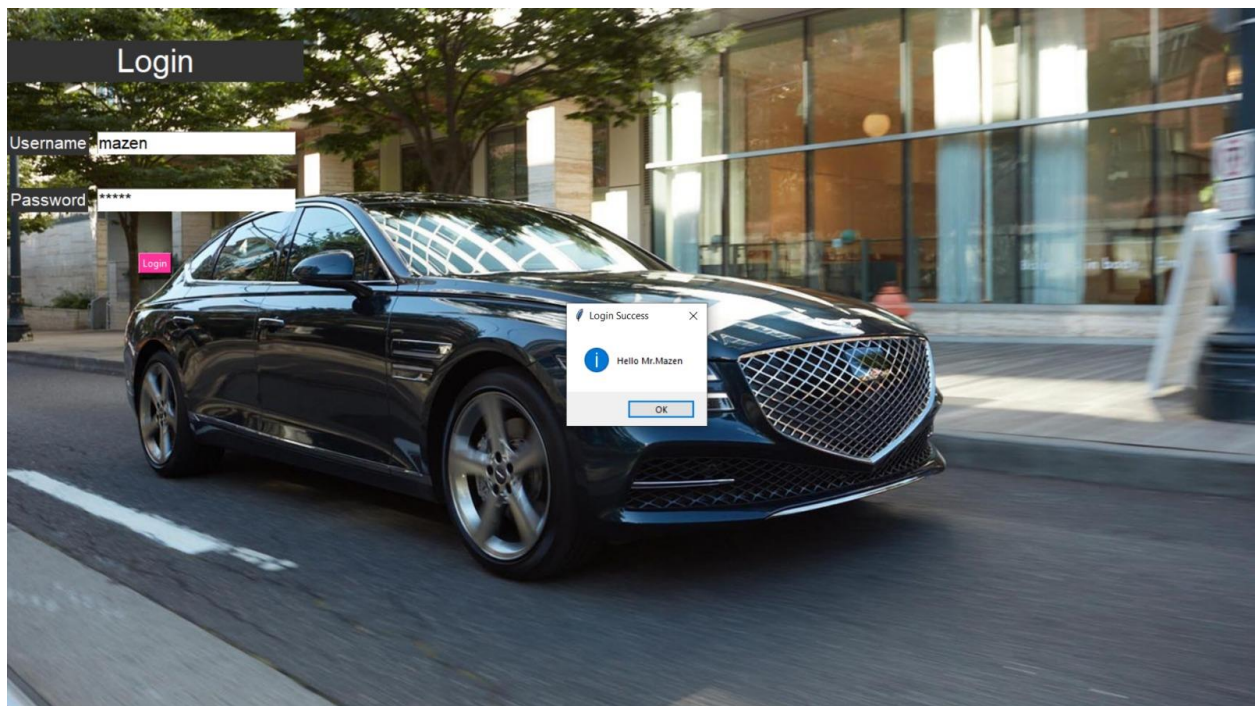
## **I/ Introduction**

L'évolution rapide de l'industrie de la location de voitures, devenue un pilier incontournable de la mobilité contemporaine, a engendré de nouveaux défis en matière de gestion opérationnelle. Afin de répondre à ces impératifs croissants et d'optimiser la gestion des locations, on a élaboré une application novatrice de gestion de localisation de voitures. Cette solution, conçue avec le langage de programmation Python et exploitant la puissante bibliothèque d'interface graphique Tkinter, se positionne comme une réponse agile aux besoins complexes de l'industrie de la location de voitures.

## II/ Présentation de l'application

### ➤ Page de connexion

Dans la page de connexion de l'application de gestion de location de voitures. Cette page permet aux utilisateurs de se connecter à l'application en saisissant leur nom d'utilisateur et leur mot de passe.



## ➤ Page du tableau de bord

Dans la page tableau de bord de l'application de gestion de location de voitures. Ce tableau de bord permet aux utilisateurs de visualiser les informations clés de l'application, telles que :

- Le nombre de réservations en cours
- Le nombre de voitures disponibles
- Le chiffre d'affaires total

Il y a un **bouton Nouvelle Réservation** permet aux utilisateurs de créer une nouvelle réservation. Lorsque l'utilisateur clique sur le bouton, il est redirigé vers la page de création de réservation.

Et on voit aussi un **bouton Administrateur** permet aux utilisateurs d'accéder à l'interface d'administration. Cette interface permet aux utilisateurs de gérer les voitures, les clients et les réservations.



## ➤ Page administrateur

Cette interface permet à l'administrateur de faire la gestion de voiture, de client et de réservation. Elle va lui permettre de gérer les voitures disponibles, leurs réservations, et les informations associées telles que les clients, les prix et le carburant et d'autre.

Les fonctionnalités clés de cette interface comprennent :

- Gestion de voiture.

Cette interface permet l'ajout et suppression de voitures avec leurs détails (matricule, marque, modèle, coute par jour et carburant), et aussi de lister toutes les voitures ainsi que de rechercher une voiture par son matricule.

- Gestion de réservation et client

Cette interface permet à l'administrateur de faire la gestion de réservation et de client. Elle va lui permettre de gérer les réservations disponibles comme la suppression et la modification d'une réservation avec leurs détails et aussi de lister toutes les réservations ainsi que de rechercher une réservation ou client par le CIN du client.

Retour

Matricule:

Coût par jour:

Marque:

Modèle:

Carburant:

Ajouter Voiture

Liste de voiture

Supprimer Voiture

Rechercher Voiture

Gestion de voiture

	Numéro	Matricule	Coût par jour	Marque	Modèle	Carburant
1		ABC123	500	Toyota	Camry	Essence
2		XYZ456	400	Honda	Accord	Électrique
3		DEF789	800	Ford	Mustang	Diesel
4		GHI987	700	Chevrolet	Malibu	Électrique
5		JKL321	900	Nissan	Altima	Essence
6		MNO654	600	BMW	3 Series	Électrique

Gestion de réservations

cin	Prénom	Téléphone	Matricule	Marque	Modèle	Date de début	Date de fin	Prix par jour (€)	Nombre de jours	Montant Total
GM123741	Mazen	06125872	ABC123	Toyota	Camry	2/2/24	2/6/24	500	5	2500.0
GK128792	Tariq	07025555	MNO654	BMW	3 Series	2/5/24	2/7/24	600	2	1200.0
KV12368452	Yacin	067821222	JKL321	Nissan	Altima	2/5/24	2/10/24	900	5	4500.0
ML1233587823	Djihad	067855555	GHI987	Chevrolet	Malibu	2/6/24	2/16/24	700	10	7000.0
QZ123582	Omar	074552323	DEF789	Ford	Mustang	2/1/24	2/15/24	800	15	12000.0
HMO215785	Yazen	0677744	XYZ456	Honda	Accord	2/1/24	2/20/24	400	20	8000.0

Liste réservations

Supprimer Réservation

Modifier Réservation

Rechercher Réservation

## ➤ Page nouvelle réservation

L'interface **Nouvelle Réservation** permet aux utilisateurs d'ajouter une nouvelle réservation d'un client.

Elle va permettre à l'utilisateur de saisir toutes les informations de réservation et puis l'enregistrer comme les informations personnelles du client telles que CIN, nom, numéro de téléphone, email et permis de conduire, ainsi que les informations souhaitées sur la voiture, telles que le modèle de voiture, la marque et la couleur. L'utilisateur peut également préciser les dates de début et de fin de location, ainsi que le nombre de jours de location.

Mais aussi cette interface permet aussi de lister toutes les réservations effectuer en cliquant sur le **bouton lister réservation**

## Nouvelle Réservation

[Retour](#)

User Information

CIN	nom	prenom	permis
sex	adresse	telephone	email

Voiture Information

marque	model	matricule	couleur

Location Information

cout par jour	nombre des jour	Date de début	Date de fin
		1/29/24	1/29/24

Confirmer la reservation

Liste des reservation

Affichage des informations:

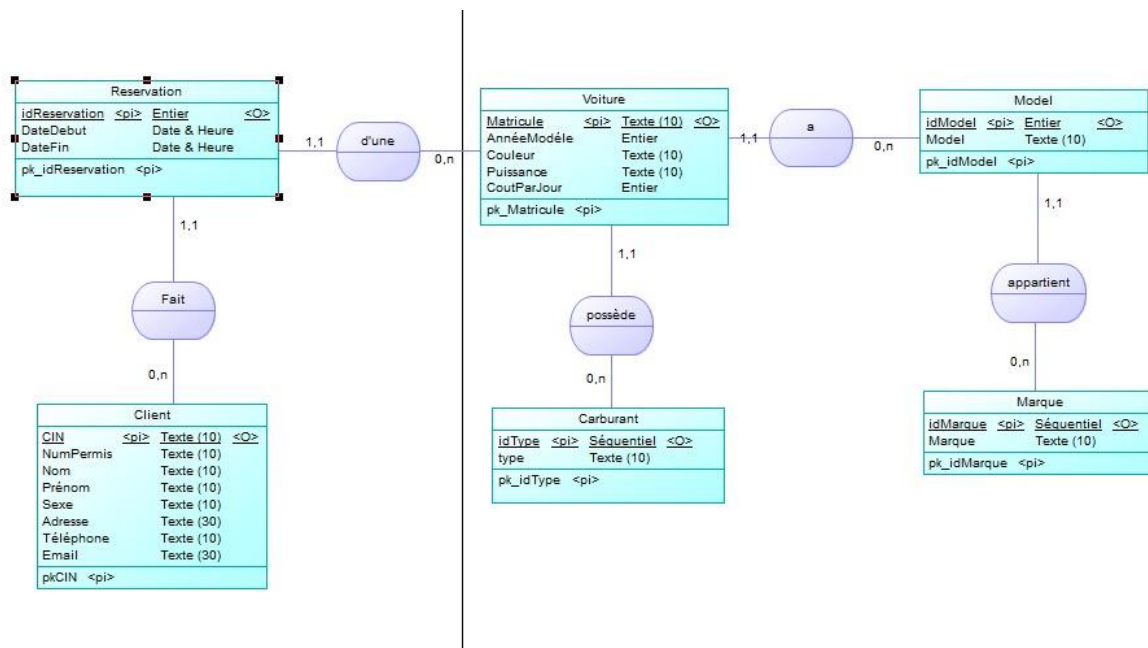
CIN	Prénom	Téléphone	Matricule	Marque	Model	Date de début	Date de fin	Prix par jour (€)	Nombre de jours	Montant Total
GM123741	Mazen	06125872	ABC123	Toyota	Camry	2/2/24	2/6/24	500	5	2500.0
GK128792	Tariq	07025555	MNO654	BMW	3 Series	2/5/24	2/7/24	600	2	1200.0
KY12369452	Yacin	067821222	JKL321	Nissan	Altima	2/5/24	2/10/24	900	5	4500.0
ML1233587823	Djihad	067855555	GH987	Chevrolet	Malibu	2/6/24	2/16/24	700	10	7000.0
QZ123582	Omar	074552323	DEF789	Ford	Mustang	2/1/24	2/15/24	800	15	12000.0
HM0215785	Yazen	0677744	XYZ456	Honda	Accord	2/1/24	2/20/24	400	20	8000.0



## III/ Conception

### ➤ Model conceptuel de donné

Avant de commencer le développement de l'application on a élaboré le model conceptuel de donné (MCD) pour voir la structure de l'application ce que va nous aider a bien déterminer les classe et les fonction principal qu'on utiliser lors du développement.





## ➤ Partie programmation

Pour la conception on utilise 3 classes avec leur fonction appropriée.

- ✓ **Classe voiture**, c'est une classe qui permet de gérer la voiture. Elle contient 4 fonctions, telle que la **fonction ajouter\_voiture** permet de rajouter une nouvelle voiture et le stocker dans un fichier nommé voiture.txt, la **fonction supprimer\_voiture** permet de supprimer une voiture, la **fonction chercher\_voiture** permet de chercher une voiture par son matricule et la **fonction afficher\_données** permet de lister toutes les voitures sur un tableau à côté.

```
class voiture:
    def ajouter_voiture(matricule, cout_par_jour, marque, modele, carburant, entry_matricule, entry_cout_par_jour): ...
    def afficher_donnees(): ...
    def supprimer_voiture(): ...
    def rechercher_voiture(): ...
```

- ✓ **Class gestion réservation**, c'est une classe qui permet de gérer les réservations. Elle contient 4 fonctions, telle que la **fonction modifier\_reservation** permet modifier les informations d'une réservation, la **fonction supprimer\_reservation** permet de supprimer une réservation, la **fonction chercher\_reservation** permet de chercher une réservation et la **fonction afficher\_données** permet de lister toutes les réservations sur un tableau à côté.

```
class gestion_reservation:
    def rechercher_reservation(): ...
    def afficher_reservations(): ...
    def supprimer_reservation(): ...
    def modifier_reservation(): ...
```

- ✓ **Class fair\_reservation**, c'est une classe qui permet de faire une nouvelle réservation. Elle contient 3 fonctions, telle que la **fonction \_\_init\_\_** permet de récupérer tous les information de la nouvelle réservation du l'utilisateur, la fonction **write\_to\_file** permet d stoker les information du réservation dans un fichier nomme client.txt, et la **fonction read\_from\_file** permet de lire les donné du fichier nommé client.txt a fin d'afficher les donné sur un tableau a coté .

```
class fair_reservation:
    def __init__(self, cin, permis, nom, prenom, sex, adresse, telephone, email, marque, model, matricule, couleu
    def write_to_file(self):...
    def open_client(tree_tous_clients):...
```

Et pour la conception de l'interface de l'application en utilisant la bibliothèque tkinter et on a créé 4 pages et chaque a son contenue et style unique :

- **Interface Login**

```
window = tk.Tk()
window.title("Login Form")

# Activer le mode plein écran pour la fenêtre de login
window.attributes("-fullscreen", True)

# Charger l'image d'arrière-plan
background_image = Image.open("background.jpg")
background_photo = ImageTk.PhotoImage(background_image)

# Créer une étiquette pour contenir l'image d'arrière-plan
background_label = tk.Label(window, image=background_photo)
background_label.place(relwidth=1, relheight=1)
```

- **Interface du tableau de bord**

```
def Menu():
    global menu_window # Déclarer la fenêtre comme variable globale
    # Détruire la fenêtre actuelle (si elle existe)
    try:
        window.destroy()
    except tk.TclError:
        pass # Ignorer l'erreur si la fenêtre n'existe pas encore

    # Créer une nouvelle fenêtre
    menu_window = tk.Tk()
    menu_window.title(" le Tableau de bord ") # Titre du tableau de bord
    background_image = Image.open("background2.jpg")
    background_photo = ImageTk.PhotoImage(background_image)
```

- **Interface administrateur**

```
def AdminPage():
    global menu_window, numero_voiture, tree ,tree_reservations # Déclarer la fenêtre, la va
    numero_voiture = 1

    global menu_window, tree # Déclarer la fenêtre et le Treeview comme variables globales
    # Détruire la fenêtre actuelle (si elle existe)
    try:
        menu_window.destroy()
    except tk.TclError:
        pass # Ignorer l'erreur si la fenêtre n'existe pas encore

    # Créer une nouvelle fenêtre pour la page d'administration
    admin_window = tk.Tk()
    admin_window.title("Page d'administration")

    # Activer le mode plein écran
    admin_window.attributes("-fullscreen", True)
```

- **Interface pour la nouvelle réservation**

```
def NouvelleReservationPage():  
    global menu_window # Déclarer la fenêtre comme variable globale  
    # Détruire la fenêtre actuelle (si elle existe)  
    try:  
        menu_window.destroy()  
    except tk.TclError:  
        pass # Ignorer l'erreur si la fenêtre n'existe pas encore  
  
    # Créer une nouvelle fenêtre pour la page de nouvelle réservation  
    reservation_window = tk.Tk()  
    reservation_window.title("Nouvelle Réservation")  
  
    # Appliquer le thème bleu  
    style = ThemedStyle(reservation_window)  
    style.set_theme("aquativo")
```