

Application de Gestion de Parc Automobile – Synthèse du Travail Réalisé

Objectif du projet

L'objectif de ce projet est de concevoir et développer une **application de gestion de parc automobile d'entreprise**, en respectant l'intégralité des exigences du sujet tout en adoptant une approche **modulaire, claire et maintenable**.

L'application permet de :

- Gérer les véhicules, les employés et leurs réservations
 - Suivre les retours, la maintenance et le carburant
 - Mettre en place une authentification avec rôles
 - Centraliser les données dans une base SQLite
 - Fournir une interface graphique simple et fonctionnelle via Tkinter
-

Choix d'architecture

Le projet repose sur une **architecture modulaire en couches**, afin de séparer clairement les responsabilités et faciliter la maintenance.

1. Interface Graphique (GUI)

- Développée avec **Tkinter**
- Utilisation de **ttk.Treeview** pour l'affichage des données tabulaires
- Une fenêtre dédiée par domaine fonctionnel :
 - Véhicules
 - Employés
 - Réservations / Retours
 - Maintenance & carburant
 - Rapports

L'objectif est d'obtenir une interface lisible, cohérente et adaptée à un usage administratif.

2. Couche logique (Business Logic)

La logique métier est découpée en **modules indépendants**, chacun responsable d'un périmètre précis :

- Gestion des véhicules
- Gestion des employés
- Réservations et retours
- Maintenance
- Authentification et rôles
- Statistiques et rapports

Ce découpage permet :

- Une meilleure lisibilité du code
 - Une évolution facilitée des règles métier
 - Une limitation des dépendances entre modules
-

3. Base de données

- Base **SQLite** via le module `sqlite3`
- Schéma relationnel conforme au sujet
- Utilisation de :
 - Clés primaires
 - Clés étrangères
 - Contraintes d'intégrité (`UNIQUE`, `NOT NULL`)

Toutes les requêtes SQL sont centralisées dans un module dédié afin de garantir la cohérence des accès aux données.

Authentification et rôles

Un système de connexion a été mis en place avec :

- Des mots de passe **hashés**
- Trois rôles distincts :
 - **Admin** : accès complet à toutes les fonctionnalités
 - **Gestionnaire** : gestion du parc automobile
 - **Employé** : accès limité à ses propres réservations

L'interface affichée dépend du rôle de l'utilisateur connecté.

Débogage et stabilisation

Au cours du développement, plusieurs points ont été corrigés :

- Alignement strict entre la base de données, les fonctions Python et l'affichage GUI
- Correction des appels aux fonctions SQL (chargement des réservations, véhicules, etc.)
- Séparation claire entre :
 - Chargement des données
 - Affichage dans les composants Tkinter

Ces ajustements ont permis de stabiliser l'application et d'éviter les erreurs liées aux callbacks Tkinter.