

Projet de fin d'année



Location Voiture

Ons AISSA
Emna KALLEL
Sahar BOUGARES
MI2B 2024/2025

TABLE DES MATIÈRES

1. SPECIFICATION DES BESOINS

- 1.1. Sujet choisi
- 1.2. Répartition des tâches entre les membres du groupe
- 1.3. Cahier des charges fonctionnel

2. ANALYSE

- 2.1. Diagramme des cas d'utilisation

3. CONCEPTION

- 3.1. Modèle Entité-Association (E/A)
- 3.2. Modèle relationnel (tables + types + clés)

4. PROGRAMMATION ET DEVELOPPEMENT

- 4.1. Technologies utilisées
- 4.2. Organisation du code source
- 4.3. Fonctionnalités développées par espace :
 - 4.3.1 Espace client 4.3.2 Espace employé 4.3.3 Espace administrateur
- 4.4. Requêtes SQL principaux

5. TEST ET DEPLOIEMENT

- 5.1. Tests fonctionnels réalisés
- 5.2. Impressions d'écran des interfaces principales avec explication courte :
 - Interfaces client - Interfaces employé - Interfaces admin

6. DIFFICULTES RENCONTREES

1. SPECIFICATION DES BESOINS

1.1. Sujet choisi

Dans le cadre de ce projet, nous avons choisi de concevoir et développer une **application de location de voitures** destinée à couvrir les besoins d'une agence de location moderne. Cette application permet de gérer l'ensemble du processus de location, depuis la création de compte client jusqu'à la restitution du véhicule et le paiement des pénalités éventuelles.

L'application est divisée en **trois espaces fonctionnels principaux** :

- un **espace client** permettant la réservation, le suivi des locations et des paiements,
- un **espace employé** pour la gestion opérationnelle des voitures et des locations,
- un **espace administrateur** pour le contrôle global du système, des utilisateurs et des statistiques.

Scénario fonctionnel global :

L'utilisateur accède à l'application depuis l'interface de login

Une fois connecté, en tant que **client**, il peut effectuer une réservation, visualiser ses anciennes locations, payer ses pénalités de retard, et mettre à jour ses informations personnelles.

Le **personnel employé** de l'agence dispose d'un tableau de bord lui permettant d'ajouter, modifier ou supprimer des voitures, de consulter l'état des locations en cours ou passées, et de consulter également les pénalités appliquées aux clients. Lorsqu'un véhicule est signalé comme **"en panne"**, il devient automatiquement indisponible à la location.

L'**administrateur**, quant à lui, gère les comptes employés (création, suppression, modification), supervise les locations et les pénalités, et accède à des statistiques avancées telles que les voitures les plus louées ou les clients réguliers.

Chaque action déclenchée par un utilisateur est reliée à la base de données via des requêtes SQL ou des procédures PL/SQL, garantissant une bonne séparation entre l'interface, la logique métier et la persistance des données

Ce système vise à faciliter la gestion quotidienne d'une agence de location de voitures tout en assurant une traçabilité complète.

1.2. Répartition des tâches entre les membres du groupe

La réalisation de ce projet a été menée de manière collaborative, avec une répartition équilibrée des tâches entre les trois membres du groupe.

Le **travail de conception** (diagramme des cas d'utilisation, modèle Entité-Association et les scénarios possibles) ainsi que la **modélisation et création de la base de données Oracle** ont été réalisés de manière **collective**, afin d'assurer une compréhension commune de la structure fonctionnelle et des données manipulées.

Pour la partie développement, les **interfaces graphiques** ont été réparties entre les membres du groupe comme suit :

- **Sahar BOUGARES** s'est chargé de :
 - ✓ l'**assemblage du projet**, incluant l'intégration des différentes interfaces et du code source.
 - ✓ le développement de l'**interface de connexion** (login), de l'interface publique.
 - ✓ l'ensemble des fonctionnalités liées à l'**espace client** incluant la réservation, gestion du compte et la consultation des locations et pénalités.
- **Ons AISSA** s'est occupé de :
 - ✓ l'**espace employé**, comprenant la gestion des voitures, la consultation des clients et de leurs pénalités, ainsi que les locations en cours ou passées.
 - ✓ l'**implémentation de la base de données Oracle**, incluant les tables, les requêtes SQL, ainsi que les procédures et fonctions PL/SQL.
 - ✓ l'**assemblage final du projet**, incluant l'intégration des différentes interfaces et du code source en utilisant GitHub.
 - ✓ Rédaction du rapport de projet.
- **Emna KALLEL** a été responsable de :
 - ✓ l'**espace administrateur**, incluant la modification de compte, la gestion des voitures, la gestion des employés, des clients, des locations ainsi que la consultation des pénalités, et des validations de locations,
 - ✓ la création du **diagramme des cas d'utilisation**
 - ✓ la réalisation du **modèle Entité-Association (E/A)** représentant la structure logique des données et du modèle relationnel.
 - ✓ l'**assemblage final du projet**, incluant l'intégration des différentes interfaces et du code source en utilisant GitHub.
 - ✓ Rédaction du rapport de projet.

1.3.Cahier des charges fonctionnel

1. Fonctionnalités côté client

- Création et gestion du compte personnel.
- Recherche et filtrage de voitures par critères (carburant, transmission, disponibilité...).
- Réservation de voitures disponibles avec interface de confirmation.
- Paiement de location et des pénalités en ligne.
- Consultation de l'historique des locations.
- Visualisation des pénalités encourues et possibilité de les régler.

2. Fonctionnalités côté employé

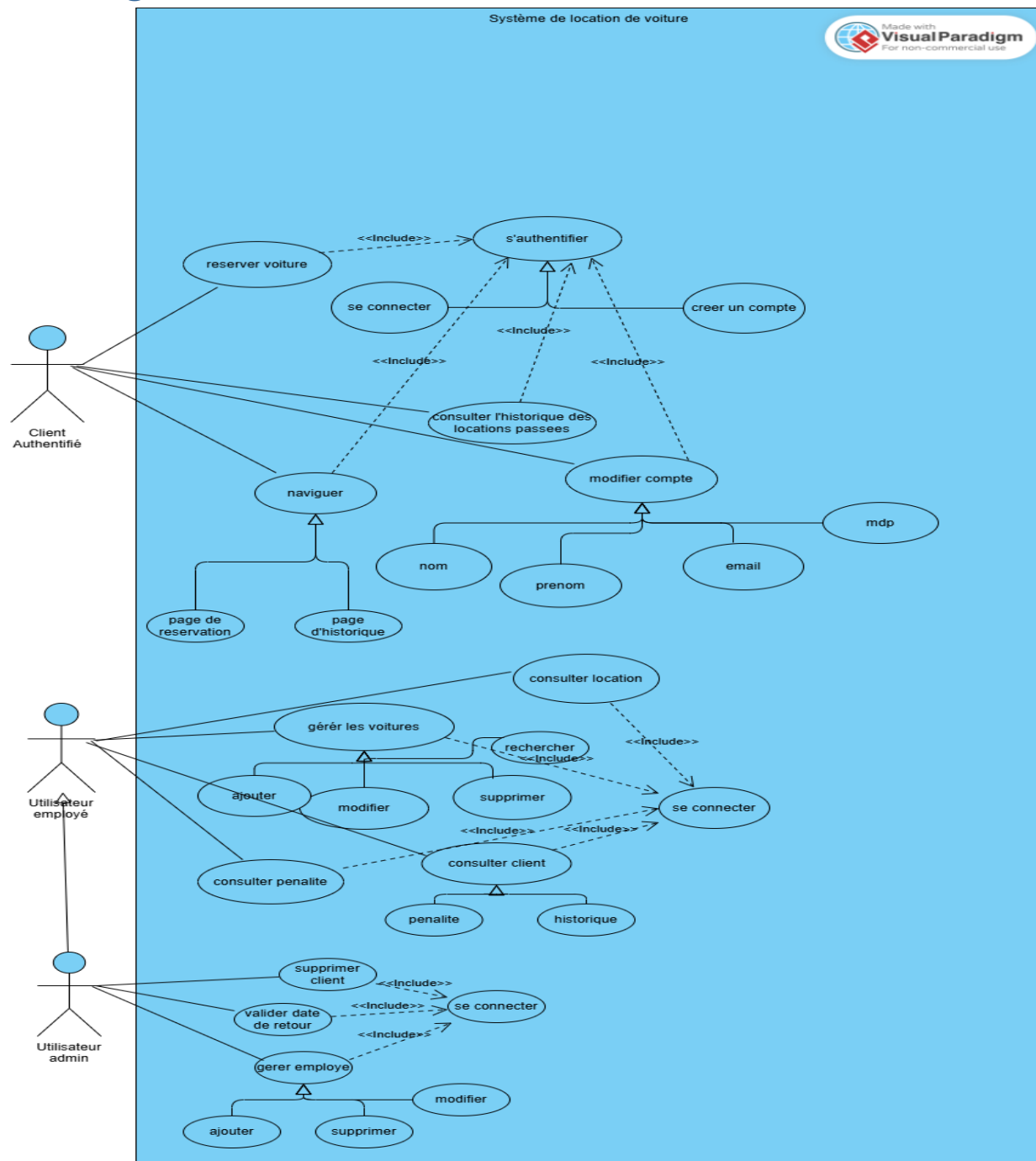
- Ajout, modification, suppression et recherche de voitures.
- Consultation des informations des clients (coordonnées, historique de location et de pénalités).
- Accès aux locations actives et passées.
- Visualisation des pénalités appliquées aux clients.

3. Fonctionnalités côté administrateur

- Accès à toutes les fonctionnalités de l'employé.
- Suppression de comptes clients.
- Consultation des informations des clients (coordonnées, historique de location et de pénalités).
- Gestion des employés (ajout, modification, suppression, recherche).
- Supervision de toutes les locations et validation des retours.
- Consultation de l'ensemble des pénalités.

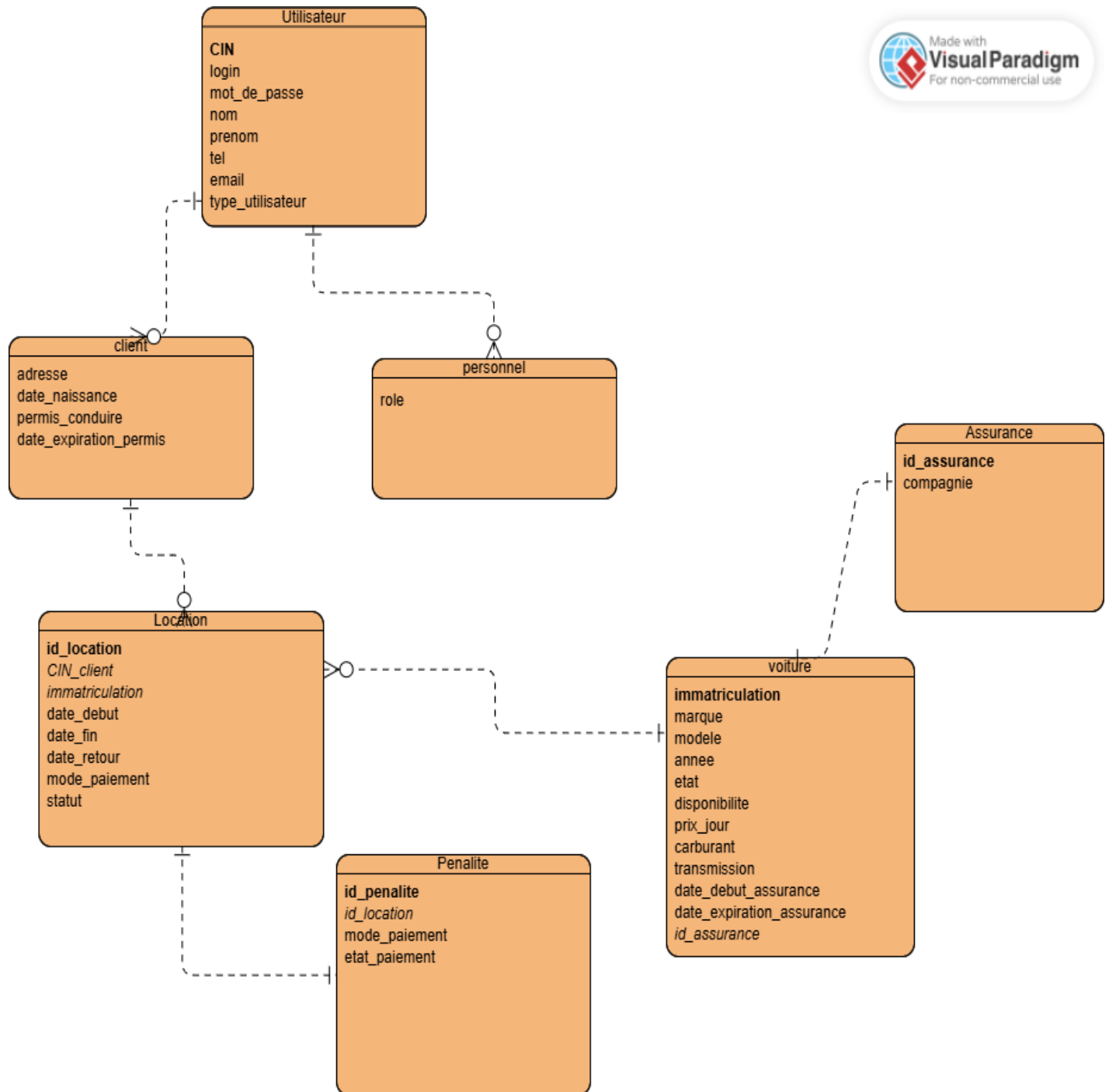
2. ANALYSE

2.1. Diagramme de cas d'utilisation



3. CONCEPTION

3.1. Modèle Entité-Association (E/A)



3.2. Modèle relationnel (tables + types + clés)

- **Utilisateur** (CIN, login, mot_de_passe, nom, prenom, tel, email, type_utilisateur)
- **Client**(#CIN, adresse, date_naissance, permis_conduire, date_expiration_permis)
- **Personnel**(#CIN, role)
- **Voiture**(immatriculation, marque, modele, annee, etat, disponibilite, prix_jour, carburant, transmission, date_debut_assurance, date_expiration_assurance, #id_assurance)
- **Assurance**(id_assurance, compagnie)
- **Location**(id_location, #CIN_client, #immatriculation, date_debut, date_fin, date_retour, mode_paiement, statut)
- **Penalite**(id_penalite, #id_location, mode_paiement, etat_paiement)

4. PROGRAMMATION/DEVELOPPEMENT

4.1. Technologies utilisées

L'étape de réalisation est la dernière de notre projet, elle se présente comme étant l'étape la plus cruciale vu qu'elle traite l'onglet pratique du projet.

Voilà une brève illustration de **l'environnement de travail** ainsi que **l'ensemble des logiciels** qu'on a utilisé dans la réalisation de notre application de gestion de location de voiture et l'implémentation de base de données.

1-Oracle Database

Pour la gestion de la base de données, nous avons opté pour Oracle Database, un système de gestion de base de données relationnelle (SGBDR) robuste, sécurisé et performant. Il permet de stocker, manipuler et interroger efficacement toutes les données relatives à l'activité hôtelière, comme les réservations, les clients, les chambres, etc.

2-Java Development Kit (JDK)

Le Java Development Kit est l'outil fondamental pour le développement en langage Java. Il fournit les bibliothèques, les outils de compilation et d'exécution nécessaires pour coder, compiler et lancer notre application. Java, étant un langage orienté objet, nous a permis d'adopter une architecture modulaire et évolutive.

3-JDBC (Java Database Connectivity)

Afin d'assurer la communication entre notre application Java et la base de données Oracle, nous avons utilisé JDBC, une API Java permettant l'exécution de requêtes SQL directement depuis le code. Cela facilite la connexion, l'insertion, la mise à jour et la suppression de données dans la base.

4-NetBeans IDE

Pour le développement de l'interface graphique et la gestion du code source, nous avons utilisé NetBeans, un environnement de développement intégré (IDE) open source. Il offre une interface conviviale, des fonctionnalités avancées pour la gestion de projets Java, ainsi qu'un excellent support pour l'intégration avec les bases de données.

Les langages de programmation utilisés

1- Le langage SQL

SQL signifie " Structured Query Language " c'est-à-dire " Langage d'interrogation structuré .

En fait SQL est un langage complet de gestion de bases de données relationnelles.

L'accès aux BDD (bases de données) se fait de façon standard à l'aide de requêtes du langage SQL.

Il existe un outil d'administration, PhpMyAdmin, qui nous offre une interface pour manipuler les tables.

La connaissance de quelques requêtes permet de répondre à la majorité des besoins de programmation

2- Définition de Java

Java C'est un langage de programmation orienté objet, développé par Sun Microsystems. Dont le squelette principale et constitué du langage C++.

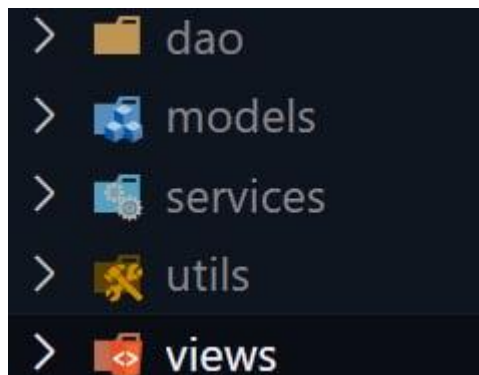
En revanche, ces deux langages sont très différents dans leurs structures (organisation du code et gestion des variables).

Ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp) et la réalisation de jeux fortement basé sur le graphisme. en effet, il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation et la création de graphismes élaborés ainsi que la présentation de texte évolué (hypertexte). L'avantage principal de Java par rapport aux autres langages c'est sa PORTABILITE.

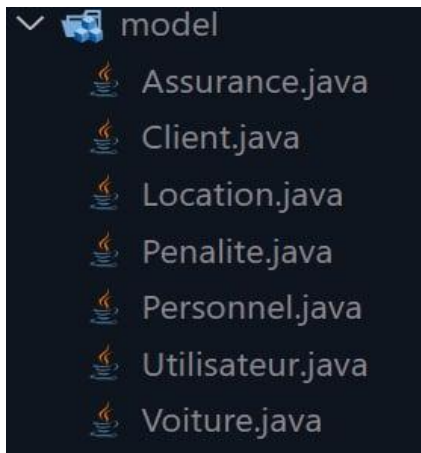
4.2. Organisation du code source

Nous avons organisé notre code source en plusieurs packages afin de garantir une meilleure lisibilité, maintenabilité et modularité de notre application.

- Le package **dao** regroupe les classes responsables de l'accès aux données. Il contient les méthodes nécessaires pour effectuer les opérations de base telles que la recherche, la suppression, la modification et l'ajout d'éléments dans la base de données.
- Le package **models** contient les classes représentant les entités de l'application. Ces classes incluent un constructeur par défaut, un constructeur paramétré, ainsi que les accesseurs (getters) et les mutateurs (setters).
- Le package **views** regroupe l'ensemble des interfaces graphiques nécessaires à l'interaction avec l'utilisateur.
- Le package **utils** contient les classes utilitaires, notamment la classe `OracleConnexion.java`, qui gère la connexion à la base de données Oracle.
- Enfin, un package dédié aux **librairies** rassemble les différentes dépendances externes utilisées dans le projet.



Package models



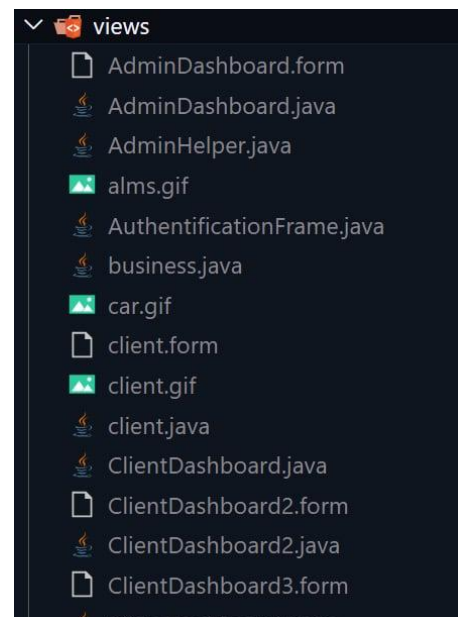
Package dao



Package utils



package views

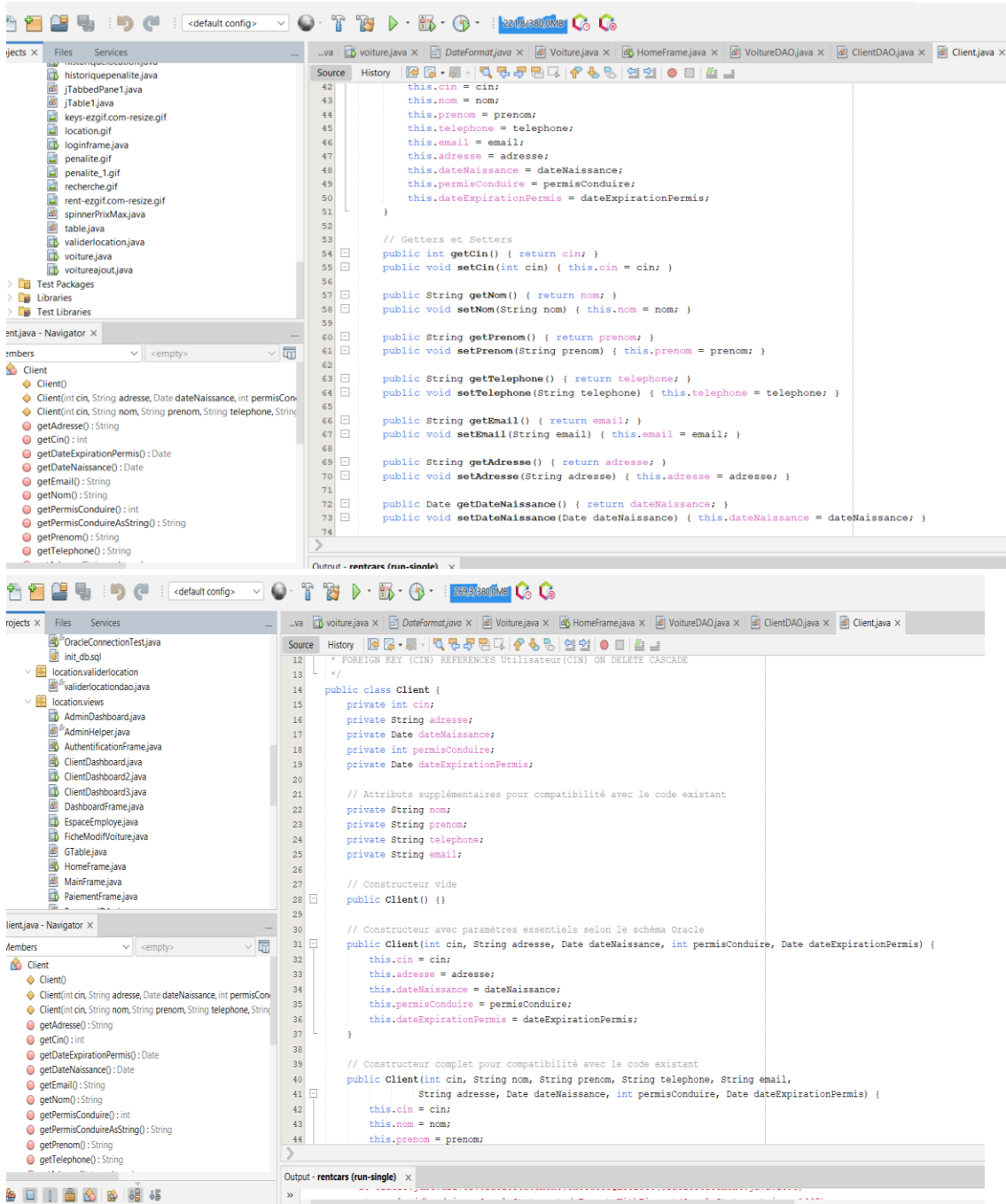


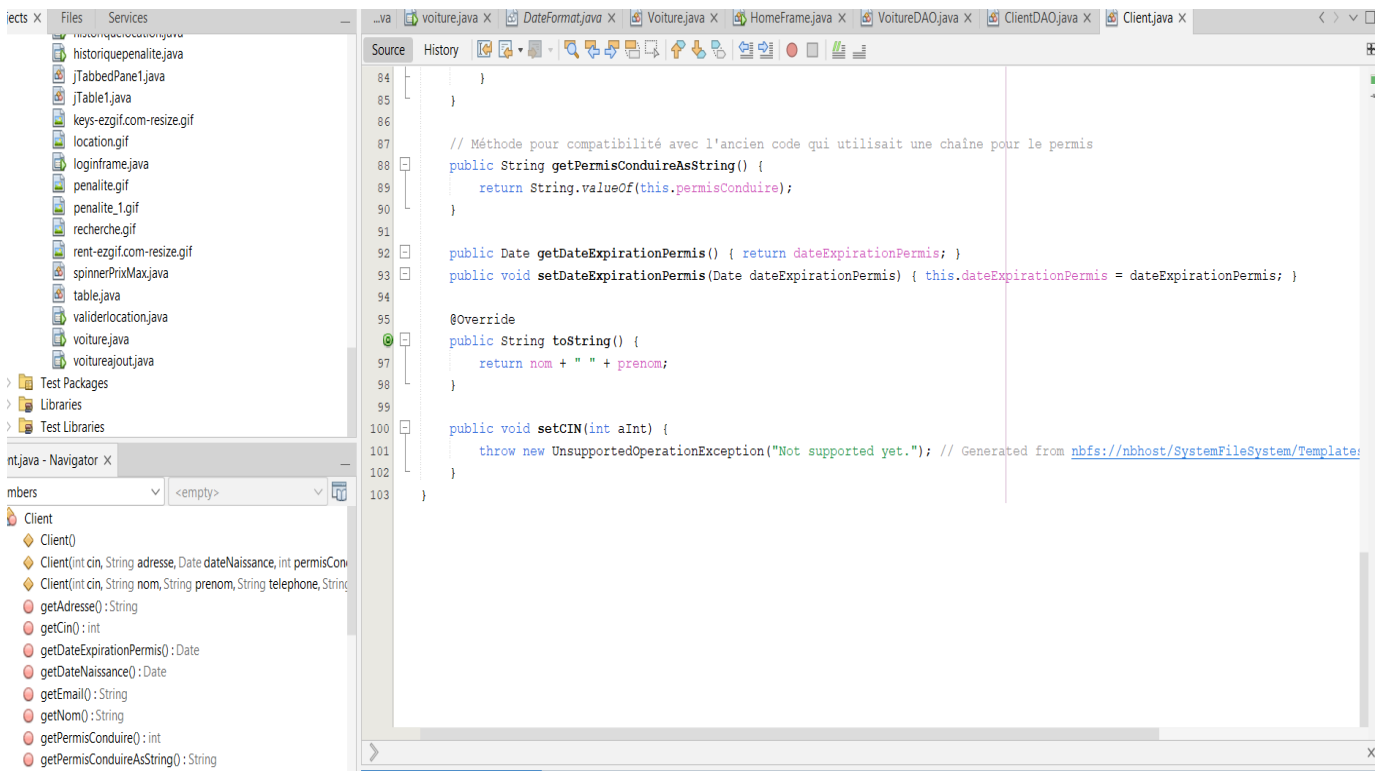
4.1. Fonctionnalités développées par espace :

4.3.1 Espace client 4.3.2 Espace employé 4.3.3 Espace administrateur

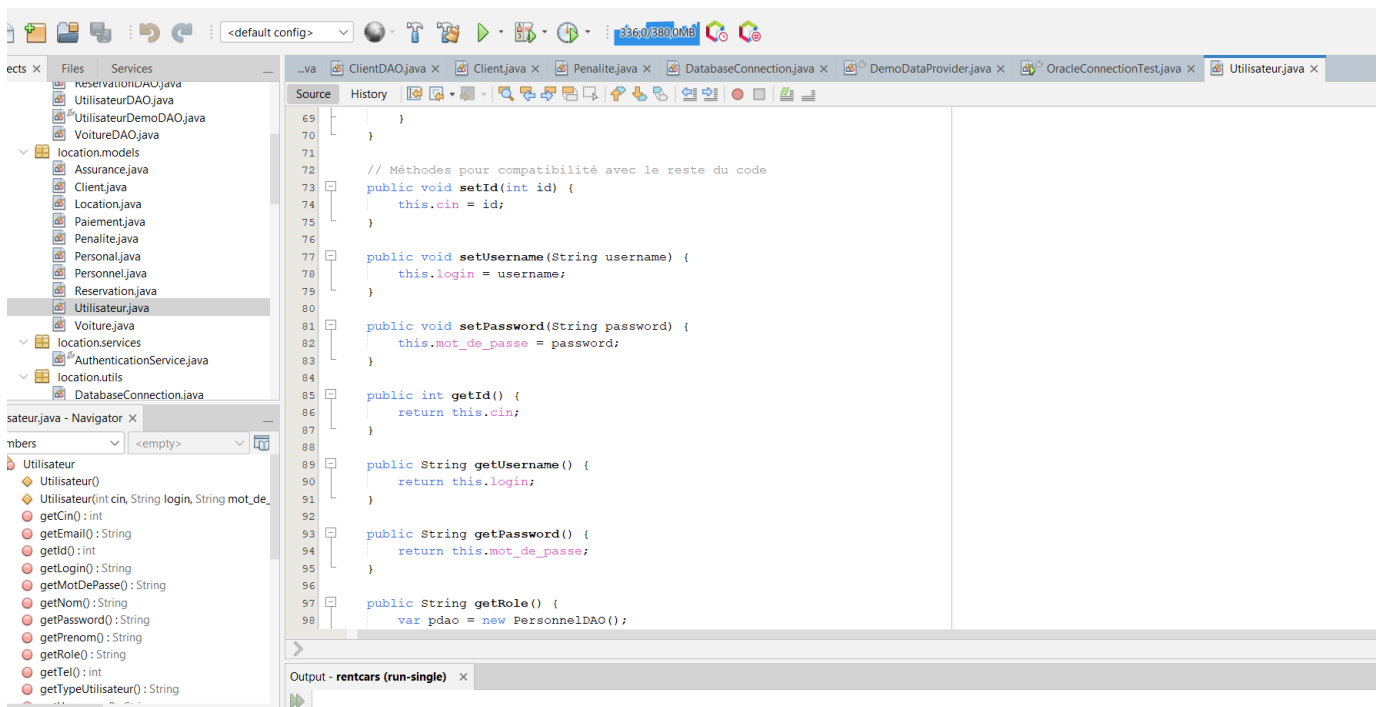
Classe client qui hérite de la classe utilisateur:

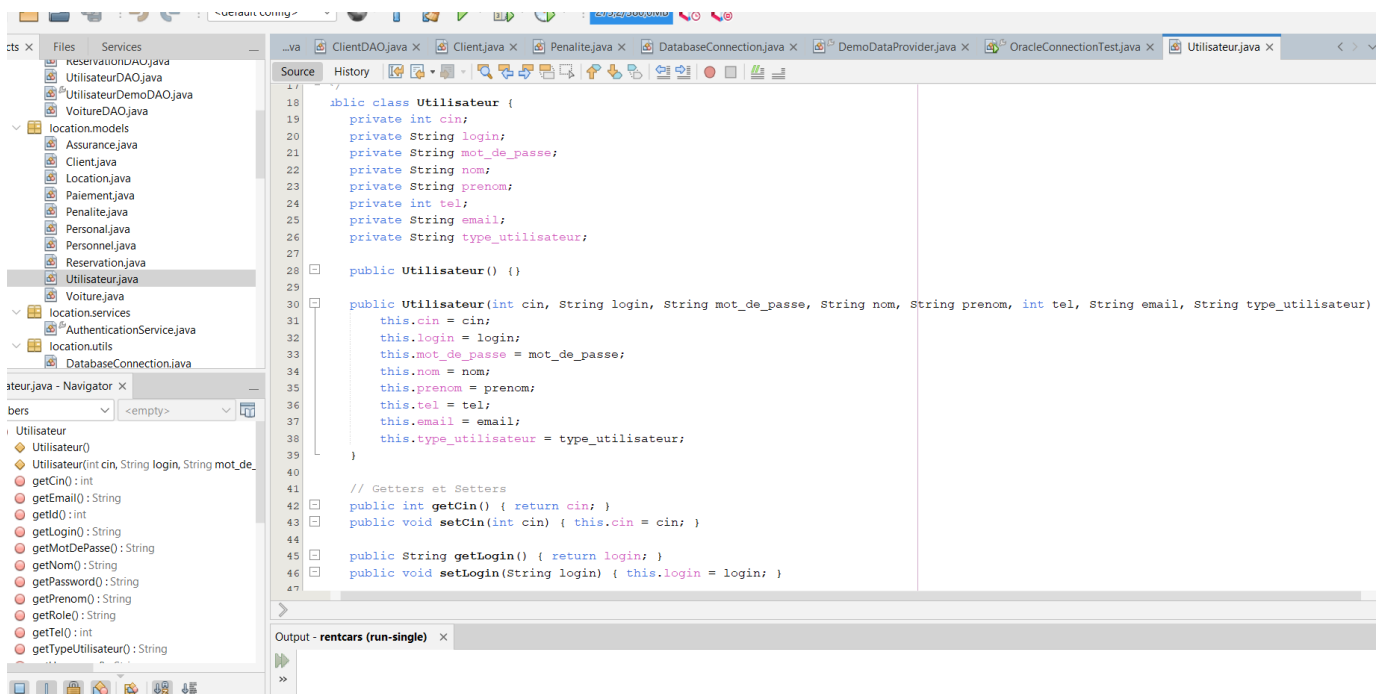
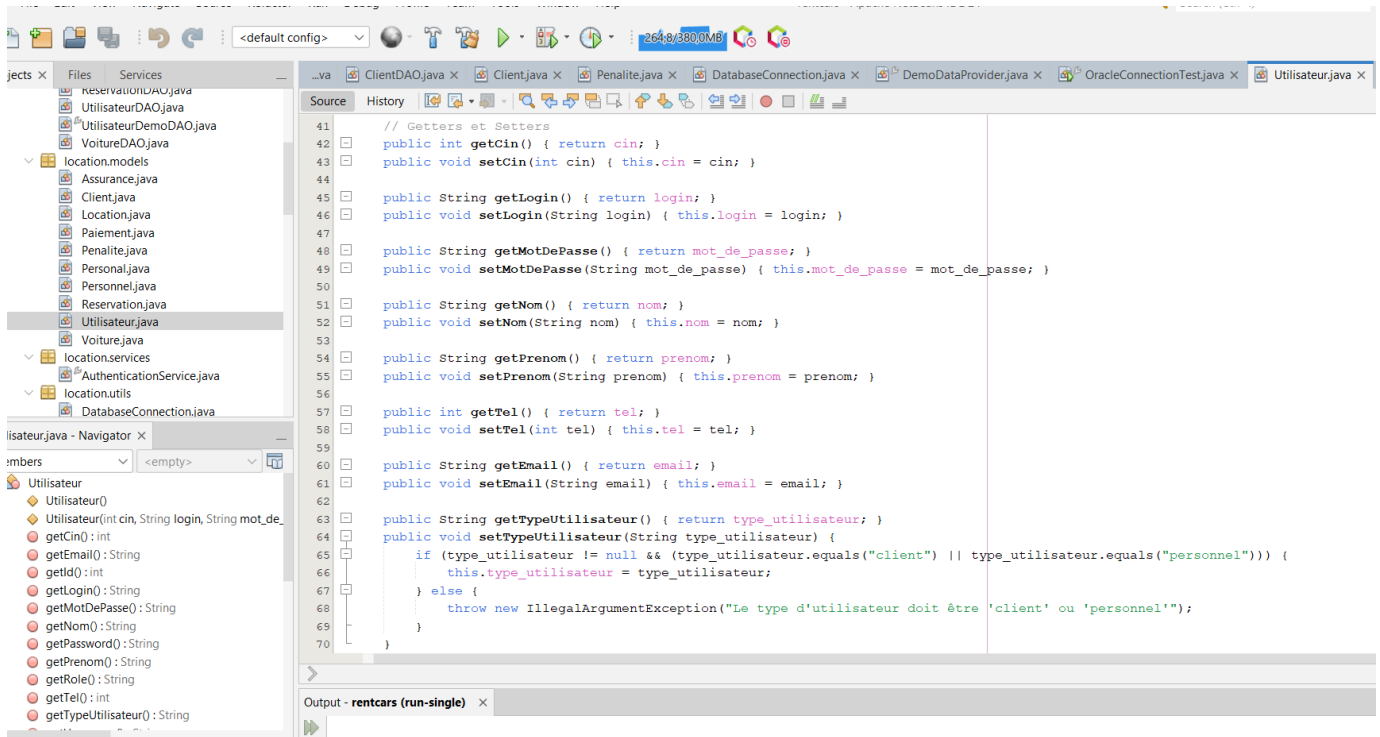
Classe-client:





Classe utilisateur:





Classe-de-connexion :

```
26         System.out.println("Pilote Oracle chargé avec succès");
27     } catch (ClassNotFoundException e) {
28         System.err.println("ERREUR: Pilote Oracle introuvable: " + e.getMessage());
29         e.printStackTrace();
30         return null;
31     }
32
33     // Établir la connexion
34     try {
35         connection = DriverManager.getConnection(URL, USER, PASSWORD);
36         System.out.println("Connexion à la base de données établie avec succès!");
37
38         // Test de la connexion
39         if (connection.isValid(5)) {
40             System.out.println("Connexion testée et validée");
41         } else {
42             System.err.println("ERREUR: La connexion est établie mais semble invalide");
43         }
44     } catch (SQLException e) {
45         System.err.println("ERREUR: Impossible de se connecter à la base de données Oracle: " + e.getMessage());
46         e.printStackTrace();
47         return null;
48     }
49 } else {
50     // Vérifier si la connexion existante est toujours valide
```

```
51     // Vérifier si la connexion existante est toujours valide
52     if (!connection.isValid(2)) {
53         System.out.println("La connexion existante n'est plus valide, tentative de reconnexion...");
54         connection.close();
55         return getConnection(); // Récursion pour rétablir la connexion
56     }
57 } catch (SQLException e) {
58     System.err.println("ERREUR lors de la vérification/fermeture de la connexion: " + e.getMessage());
59     e.printStackTrace();
60     connection = null; // Réinitialiser la connexion en cas d'erreur
61 }
62
63 return connection;
64 }
65
66 // Méthode pour fermer la connexion
67 public static void closeConnection() {
68     try {
69         if (connection != null && !connection.isClosed()) {
70             connection.close();
71             System.out.println("Connexion à la base de données fermée");
72         }
73     } catch (SQLException e) {
74         System.err.println("Erreur lors de la fermeture de la connexion: " + e.getMessage());
```


4.4. Requetes SQL principals

```
1  -- Suppression des tables définitivement si elles existent
2  DROP TABLE Penalite;
3  DROP TABLE Location;
4  DROP TABLE Voiture;
5  DROP TABLE Personnel;
6  DROP TABLE Client;
7  DROP TABLE Utilisateur;
8  DROP TABLE Assurance;
9
10 -- Création des tables
11
12 -- Table Utilisateur
13 CREATE TABLE Utilisateur (
14     CIN NUMBER(8) PRIMARY KEY,
15     login VARCHAR2(50) UNIQUE NOT NULL,
16     mot_de_passe VARCHAR2(255) NOT NULL,
17     nom VARCHAR2(50),
18     prenom VARCHAR2(50),
19     tel NUMBER(8),
20     email VARCHAR2(100),
21     type_utilisateur VARCHAR2(20) NOT NULL CHECK (type_utilisateur IN ('client', 'personnel')),
22     CONSTRAINT email_format CHECK (email LIKE '%@%.%')
23 );
```

```

33
34 -- Table Personnel
35 CREATE TABLE Personnel (
36     CIN NUMBER(8) PRIMARY KEY,
37     role VARCHAR2(20) CHECK (role IN ('admin', 'employe')),
38     FOREIGN KEY (CIN) REFERENCES Utilisateur(CIN) ON DELETE CASCADE
39 );
40
41 -- Table Assurance
42 CREATE TABLE Assurance (
43     id_assurance NUMBER(10) PRIMARY KEY,
44     compagnie VARCHAR2(100)
45 );
46
47 -- Table Voiture
48 CREATE TABLE Voiture (
49     immatriculation VARCHAR2(20) PRIMARY KEY,
50     marque VARCHAR2(50),
51     modele VARCHAR2(50),
52     annee NUMBER(4),
53     etat VARCHAR2(20) CHECK (etat IN ('en_marche', 'en_panne')),
54     disponibilite VARCHAR2(20) CHECK (disponibilite IN ('disponible', 'louee', 'indisponible')),
55     prix_jour DECIMAL(10,3),
56     carburant VARCHAR2(20) CHECK (carburant IN ('essence', 'diesel', 'electrique')),
57     transmission VARCHAR2(20) CHECK (transmission IN ('automatique', 'manuel')),
58     date_debut_assurance DATE,
59     date_expiration_assurance DATE,
60     id_assurance NUMBER(10),
61     FOREIGN KEY (id_assurance) REFERENCES Assurance(id_assurance) ON DELETE CASCADE
62 );
63
64
65

```

```

65
66 -- Table Location
67 CREATE TABLE Location (
68     id_location NUMBER(10) PRIMARY KEY,
69     CIN_client NUMBER(8),
70     immatriculation VARCHAR2(20),
71     date_debut DATE,
72     date_fin DATE,
73     date_retour DATE,
74     mode_paiement VARCHAR2(20) CHECK (mode_paiement IN ('carte bancaire', 'application D17')),
75     statut VARCHAR2(50) CHECK (statut IN ('en_cours', 'terminee')),
76     FOREIGN KEY (CIN_client) REFERENCES Client(CIN) ON DELETE CASCADE,
77     FOREIGN KEY (immatriculation) REFERENCES Voiture(immatriculation) ON DELETE CASCADE
78 );
79
80 -- Table Penalite
81 CREATE TABLE Penalite (
82     id_penalite NUMBER(10) PRIMARY KEY,
83     id_location NUMBER(10),
84     mode_paiement VARCHAR2(20) CHECK (mode_paiement IN ('carte bancaire', 'application D17')),
85     etat_paiement VARCHAR2(50) CHECK (etat_paiement IN ('en_attente', 'effectue')),
86     FOREIGN KEY (id_location) REFERENCES Location(id_location) ON DELETE CASCADE
87 );
88
89 -- Séquence pour les locations
90 CREATE SEQUENCE location_seq START WITH 1 INCREMENT BY 1;
91 -- Séquence pour les pénalités
92 CREATE SEQUENCE penalite_seq START WITH 1 INCREMENT BY 1;
93
94 DROP SEQUENCE location_seq;
95 DROP SEQUENCE penalite_seq;
96
97

```

```

97
98 -- Insertion de données
99
100 -- Insertion dans la table Utilisateur
101 INSERT INTO Utilisateur VALUES (12345678, 'client1', 'password123', 'Doe', 'John', 98765432, 'john.doe@email.com', 'client');
102 INSERT INTO Utilisateur VALUES (87654321, 'admin1', 'adminpass', 'Smith', 'Jane', 12345678, 'jane.smith@email.com', 'personnel');
103 INSERT INTO Utilisateur VALUES (11223344, 'employe1', 'motdepasse123', 'Ali', 'Mejri', 55667788, 'ali.mejri@email.com', 'personnel' );
104
105
106 -- Insertion dans la table Client
107 INSERT INTO Client VALUES (12345678, '12 rue de la Paix, Tunis', TO_DATE('1990-05-15', 'YYYY-MM-DD'), 123456, TO_DATE('2030-05-15', 'YYYY-MM-DD'));
108
109 -- Insertion dans la table Personnel
110 INSERT INTO Personnel VALUES (87654321, 'admin');
111
112 INSERT INTO Personnel VALUES (11223344, 'employe');
113
114 -- Insertion dans la table Assurance
115 INSERT INTO Assurance VALUES (1, 'AXA');
116 INSERT INTO Assurance VALUES (2, 'Allianz');
117
118 -- Insertion dans la table Voiture
119 INSERT INTO Voiture VALUES ('123TUN4567', 'Toyota', 'Corolla', 2020, 'en_marche', 'disponible', 30.500, 'essence', 'automatique', TO_DATE('2021-01-01', 'YYYY-MM-DD'), TO_DATE('2022-02-01', 'YYYY-MM-DD'));
120 INSERT INTO Voiture VALUES ('234TUN4568', 'Honda', 'Civic', 2021, 'en_panne', 'indisponible', 40.250, 'diesel', 'manuel', TO_DATE('2022-02-01', 'YYYY-MM-DD'), TO_DATE('2023-03-01', 'YYYY-MM-DD'));
121 INSERT INTO Voiture VALUES ('123TUN4569', 'BMW', 'X3', 2020, 'en_marche', 'disponible', 30.500, 'essence', 'automatique', TO_DATE('2021-01-01', 'YYYY-MM-DD'), TO_DATE('2022-02-01', 'YYYY-MM-DD'));
122
123 -- Insertion dans la table Location (avec séquence pour id_location)
124 INSERT INTO Location (id_location, CIN_client, immatriculation, date_debut, date_fin, date_retour, mode_paiement, statut) VALUES (location_seq.NEXTVAL, 12345678, '123TUN4567', TO_DATE('2021-01-01', 'YYYY-MM-DD'), TO_DATE('2022-02-01', 'YYYY-MM-DD'), TO_DATE('2022-02-01', 'YYYY-MM-DD'), 'complet', 'en_location');
125 INSERT INTO Location (id_location, CIN_client, immatriculation, date_debut, date_fin, date_retour, mode_paiement, statut) VALUES (location_seq.NEXTVAL, 12345678, '123TUN4568', TO_DATE('2021-01-01', 'YYYY-MM-DD'), TO_DATE('2022-02-01', 'YYYY-MM-DD'), TO_DATE('2022-02-01', 'YYYY-MM-DD'), 'complet', 'en_location');
126
127 -- Insertion dans la table Penalite (id_location = 1 déjà connu)
128 INSERT INTO Penalite (id_penalite, id_location, mode_paiement, etat_paiement) VALUES (penalite_seq.NEXTVAL, 1, 'carte bancaire', 'effectuee');
129 INSERT INTO Penalite (id_penalite, id_location, mode_paiement, etat_paiement) VALUES (penalite_seq.NEXTVAL, 1, 'application D17', 'en_attente');
130
131
132
133
134
135
136
137 select * from assurance;
138 SELECT * FROM voiture;
139 select * from penalite;
140 select * from UTILISATEUR;
141 select * from location;
142 select * from client;
143 select * from personnel;
144
145
146 -- Requetes demandees

```

```

171 | -- Le client régulier
172 | SELECT *
173 | FROM (
174 | SELECT u.CIN, u.nom, u.prenom, COUNT(*) AS total_locations
175 | FROM Client c
176 | JOIN Utilisateur u ON u.CIN = c.CIN
177 | JOIN Location l ON l.CIN_client = c.CIN
178 | GROUP BY u.CIN, u.nom, u.prenom
179 | ORDER BY total_locations DESC
180 | )
181 | WHERE ROWNUM <= 3;
182 |
183 |
184 | CREATE OR REPLACE FUNCTION get_location_summary (
185 |     p_CIN_client IN NUMBER
186 | )
187 | RETURN SYS_REFCURSOR
188 | IS
189 |     v_cursor SYS_REFCURSOR;
190 |     v_total_voitures NUMBER := 0;
191 | BEGIN
192 |     -- Afficher le nombre total de voitures différentes louées par le client
193 |     SELECT COUNT(DISTINCT immatriculation)
194 |     INTO v_total_voitures
195 |     FROM Location
196 |     WHERE CIN_client = p_CIN_client;
197 |
198 |     DBMS_OUTPUT.PUT_LINE('Nombre total de voitures louées par le client ' || p_CIN_client || ' : ' || v_total_voitures);
199 |
200 |     -- Ouvrir le curseur avec les détails des durées de location
201 |     OPEN v_cursor FOR
202 |     SELECT
203 |         v.immatriculation,
204 |         v.marque,
205 |         v.modele,
206 |         SUM(l.date_fin - l.date_debut) AS duree_totale_jours
207 |     FROM Location l
208 |     JOIN Voiture v ON v.immatriculation = l.immatriculation
209 |     WHERE l.CIN_client = p_CIN_client
210 |     GROUP BY v.immatriculation, v.marque, v.modele;
211 |
212 |     RETURN v_cursor;
213 | END;
214 | /
215 |

```

```

241
242 -- Requetes demandées
243
244 -- Liste des voitures existantes et leurs états
245 SELECT immatriculation, marque, modele, etat
246 FROM Voiture;
247
248 -- Liste des voitures disponibles et louées
249 SELECT immatriculation, marque, modele, disponibilite
250 FROM Voiture
251 WHERE disponibilite IN ('disponible', 'louee');
252
253 -- Liste des 3 voitures les plus demandées
254 SELECT *
255 FROM (
256     SELECT immatriculation, COUNT(*) AS nb_locations
257     FROM Location
258     GROUP BY immatriculation
259     ORDER BY nb_locations DESC
260 )
261 WHERE ROWNUM <= 3;
262
264
265 -- Le client régulier
266 SELECT *
267 FROM (
268     SELECT
269         u.CIN,
270         u.nom,
271         u.prenom,
272         COUNT(*) AS total_locations
273     FROM
274         Client c
275         JOIN Utilisateur u ON u.CIN = c.CIN
276         JOIN Location l ON l.CIN_client = c.CIN
277     GROUP BY
278         u.CIN, u.nom, u.prenom
279     ORDER BY
280         total_locations DESC
281 )
282 WHERE ROWNUM = 1;
283
284
285
286
287 select * from Voiture ORDER BY immatriculation;
288
289

```

```

298 CREATE OR REPLACE TRIGGER trg_check_etat_disponibilite
299 BEFORE INSERT OR UPDATE ON Voiture
300 FOR EACH ROW
301 BEGIN
302     IF :NEW.etat = 'en_panne' AND :NEW.disponibilite != 'indisponible' THEN
303         RAISE_APPLICATION_ERROR(-20001, 'Une voiture en panne doit avoir une disponibilité "indisponible.");
304     END IF;
305 END;
306 /
307
308
309 SET SERVEROUTPUT ON
310 BEGIN
311     -- Ce bloc devrait provoquer une erreur
312     INSERT INTO Voiture (
313         immatriculation, marque, modele, annee, etat, disponibilite,
314         prix_jour, carburant, transmission,
315         date_debut_assurance, date_expiration_assurance, id_assurance
316     )
317     VALUES (
318         'TEST1234', 'Renault', 'Clio', 2019, 'en_panne', 'disponible',
319         35.000, 'essence', 'manuel',
320         TO_DATE('2023-01-01', 'YYYY-MM-DD'), TO_DATE('2024-01-01', 'YYYY-MM-DD'), 1
321     );
322 EXCEPTION
323     WHEN OTHERS THEN
324         DBMS_OUTPUT.PUT_LINE('Erreur attendue : ' || SQLERRM);
325 END;
326 /
327
328

```

```

-- Supprimer les enregistrements dépendants d'abord
DELETE FROM Penalite;
DELETE FROM Location;
DELETE FROM Voiture;
DELETE FROM Personnel;
DELETE FROM Client;
DELETE FROM Utilisateur;
DELETE FROM Assurance;

```


6.1. Impressions d'écran des interfaces principales avec explication courte : -Interfaces client - Interfaces employé - Interfaces admin

Interface accueil :

 E.S.O Car Manager - Accueil

E.S.O Car Manager

Bienvenue dans notre service de location de voitures. Veuillez consulter notre parc de véhicules.

[Afficher toutes les voitures disponibles](#)

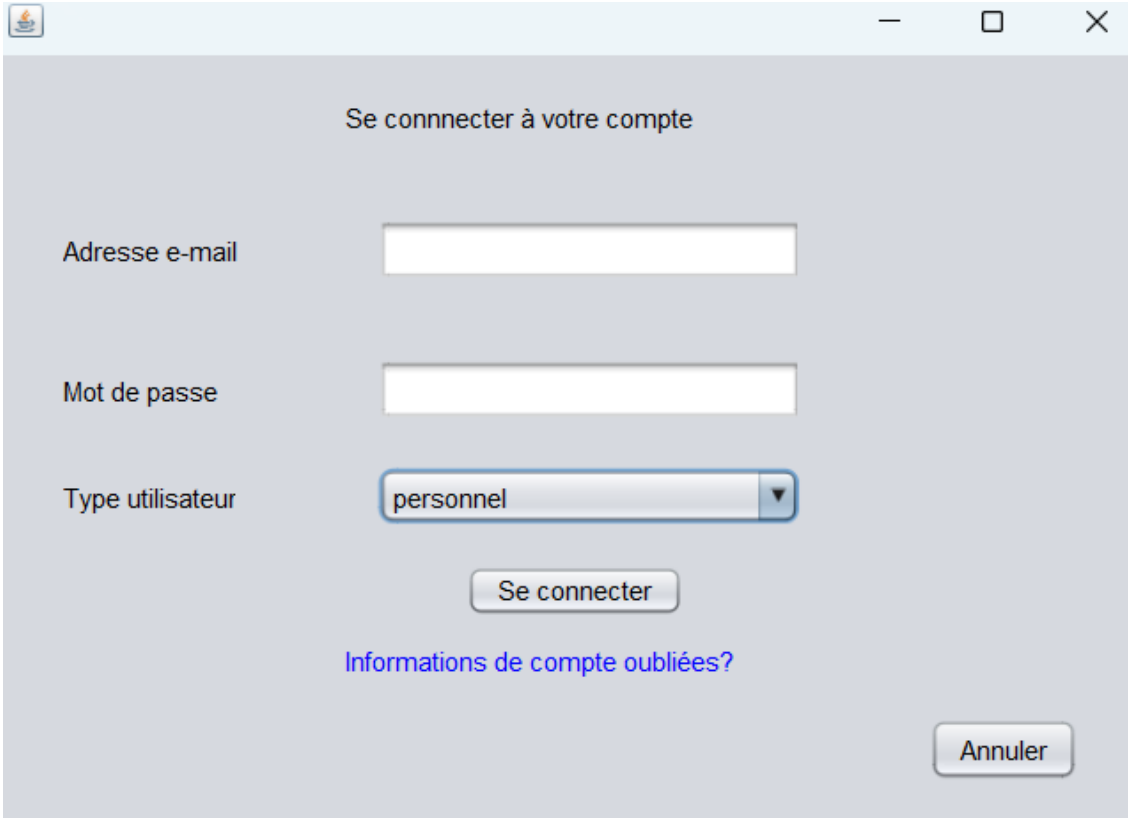
Rechercher une voiture : [Rechercher](#)

Filtrer par marque : Toutes ▼ Prix max/jour : 1 000 ▲▼ [Appliquer les filtres](#) [Réinitialiser](#)

** Utilisez les options ci-dessus pour trouver la voiture qui vous convient*

ID	Marque	modele	annee	Prix/Jour	Couleur	T
111TUN	VOLSWAGEN	POLO	2020	20.0 DH	Non spécifiée	a
123TUN4...	Toyota	Corolla	2020	30.5 DH	Non spécifiée	a
234TUN4...	Honda	Civic	2021	40.25 DH	Non spécifiée	n

Interface pour se connecter :



The image shows a standard Windows-style window with a title bar containing a small icon on the left and minimize, maximize, and close buttons on the right. The window's content area has a light gray background and is titled "Se connecter à votre compte" in a dark gray font. Below the title, there are three input fields arranged vertically. The first is labeled "Adresse e-mail" and is an empty text box. The second is labeled "Mot de passe" and is also an empty text box. The third is labeled "Type utilisateur" and is a dropdown menu with "personnel" selected and a small downward arrow on the right. Below these fields is a button labeled "Se connecter". Underneath the button is a link in blue text that says "Informations de compte oubliées?". In the bottom right corner of the window is another button labeled "Annuler".

Se connecter à votre compte

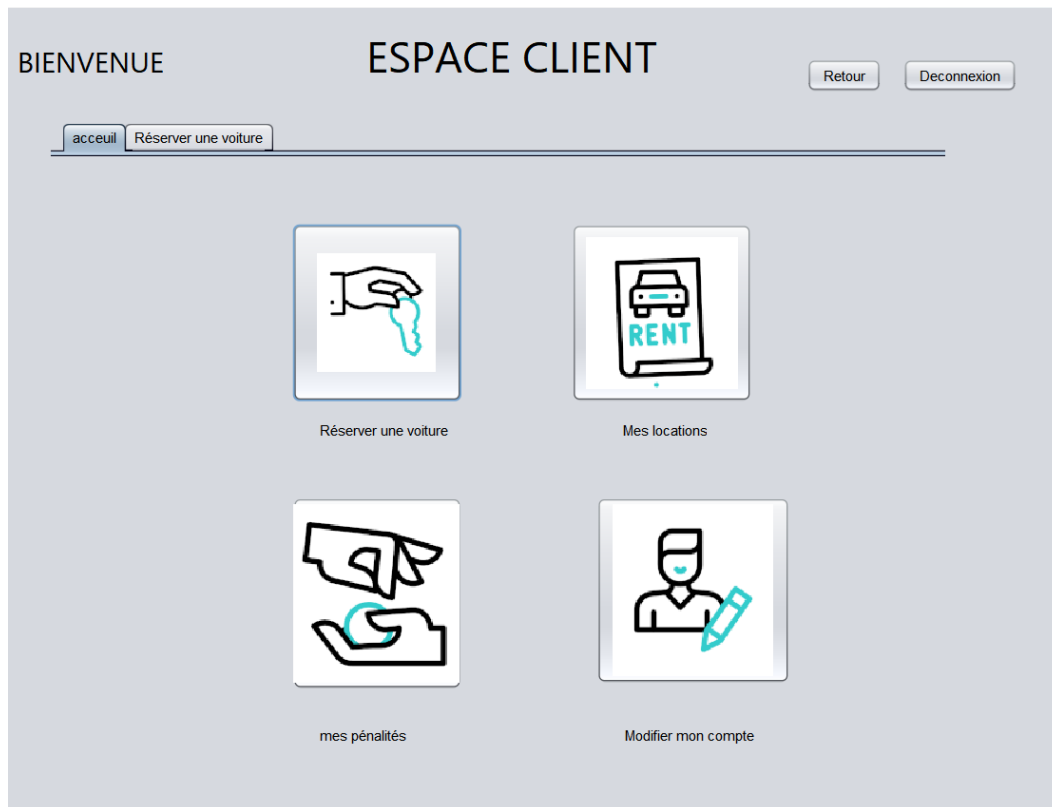
Adresse e-mail

Mot de passe

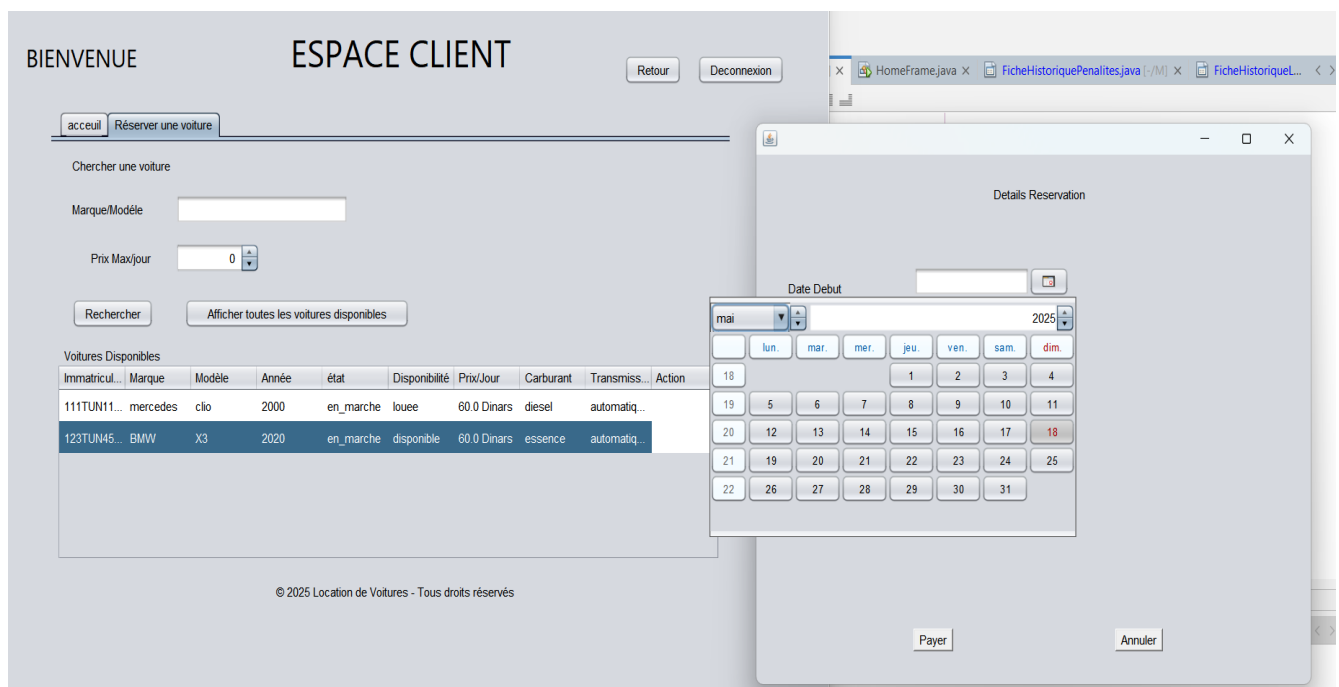
Type utilisateur

[Informations de compte oubliées?](#)

Interface Compte Client



Interface Reserver voiture Client



BIENVENUE

ESPACE CLIENT

Retour

Deconnexion

accueil

Réserver une voiture

Mes Réservations

Mon compte

Nom

Prénom

Email

Téléphone

Adresse

Mettre à jour mes informations

Ancien mot de passe

Nouveau mot de passe

Confirmer mot de passe

Mettre à jour mot de passe

Interface mes penalites Client

The screenshot shows a web application interface for a client's account. The header includes a welcome message 'BIENVENUE', the title 'ESPACE CLIENT', and buttons for 'Retour' and 'Deconnexion'. A navigation bar contains links: 'accueil', 'Réserver une voiture', 'Mes Réservations', 'Mon compte', and 'Mes pénalités'. Below the navigation bar is a 'rafraichir' button. The main content area displays a table of penalties.

ID penalite	ID location	mode paiement	etat paiement	montant calcule	Action
1	1	carte bancaire	effectue	36.6	
2	1	application D17	effectue	36.6	

Interface Compte Admin

Bienvenu Admin ,

Mon Compte | Gestion Voitures | Gestion Employés | Gestion Clients | Gestion Locations | Consultation Pénalités

CIN : 87654321

Nom : Smith

Prénom : Jane

Téléphone : 12345678

Email : jane.smith@email.com

Nouveau mot de passe : adminpass

Mettre à jour mon compte

Interface Gestion Voitures Admin

Bienvenu Admin ,

Mon Compte | **Gestion Voitures** | Gestion Employés | Gestion Clients | Gestion Locations | Consultation Pénalités

Ajouter une voiture | Rafraichir

Immatriculation	Marque	Modèle	Année	Etat	Disponibilité	Prix_jour	Carburant	Transmission	date_debut_a...	date_expirati...	id_assurance	Action
123TUN4567	Toyota	Corolla	2020	en_marche	louee	30,5	essence	automatique	2021-01-01 0...	2022-01-01 0...	1	Modifier
234TUN4568	Honda	Civic	2021	en_panne	indisponible	40,25	diesel	manuel	2022-02-01 0...	2023-02-01 0...	2	Modifier
111TUN	VOLSWAGEN	POLO	2020	en_panne	disponible	20	diesel	automatique	2020-01-01 0...	2022-01-01 0...	1	Modifier

Interface Modifier voiture Admin

Fiche de voiture

Immatriculation

Marque

Modèle

Année

Etat

Type_carburant

Transmission

Prix_jour

Date_debut_assurance

Date_expiration_assurance

id_assurance

Consultation Pénalités

Prix_jour	Carburant	Transmission	date_debut_a...	date_expirati...	id_assurance	Action
30,5	essence	automatique	2021-01-01 0...	2022-01-01 0...	1	<input type="button" value="Modifier"/>
40,25	diesel	manuel	2022-02-01 0...	2023-02-01 0...	2	<input type="button" value="Modifier"/>
20	diesel	automatique	2020-01-01 0...	2022-01-01 0...	1	<input type="button" value="Modifier"/>

Interface Compte Admin

Bienvenu A

Mon Compte Gestion

Ajouter une voiture

Immatriculation	Marque
123TUN4567	Toyota
234TUN4568	Honda
111TUN	VOLVO

Gestion de voiture

Fiche de voiture

Immatriculation

Marque

Modèle

Année

Etat

prix_jour

Carburant

Transmission

Date_debut_assurance

Date_expiration_assuran...

id_assurance

Ajouter

Retour

jour	Carburant	Transmission	date_debut_a...
30,5	essence	automatique	2021-01-01 0...
40,25	diesel	manuel	2022-02-01 0...
20	diesel	automatique	2020-01-01 0...

connexion à la base de données réussie avec succès!

Interface Gestion Employe Admin

envenu Admin ,

Mon Compte | Gestion Voitures | **Gestion Employés** | Gestion Clients | Gestion Locations | Consultation Pénalités

[Ajouter un employé](#) [Rafraichir](#)

CIN	Nom	Prénom	Tel	Email	Action
11223344	Ali	Mejri		55667788 ali.mejri@email.com	Modifier

Interface Ajouter Employe Admin

Gestion d'employé

Fiche d'employé

CIN

Login

Mot de passe

Nom

Prénom

Téléphone

E-mail

Type utilisateur

Rôle

[Ajouter](#) [Retour](#)

	Email	Action
55667788	ali.mejri@email.com	Modifier

Interface Compte Admin

ienvenu Admin ,

Mon Compte | Gestion Voitures | Gestion Employés | **Gestion Clients** | Gestion Locations | Consultation Pénalités

Rafraichir

IN	Nom	Prénom	Tel	Email	Adresse	Date de ...	Permis d...	Date expi...	Action
12345678	Doe	John	98765432	john.doe...	12 rue de...	1990-05-...	123456	2030-05-...	<button>Historique pénalité</button> <button>Historique location</button> <button>Supprimer</button>

Interfaces pour consultation historique pénalité client

envenu Admin ,

on Compte | Gestion Voitures | Gestion Employés | **Gestion Clients** | Gestion Locations | Consultation Pénalités

Rafraichir

	Nom	Prénom	Tel	Email	Adresse	Date de ...	Permis d...	Date expi...	Action
345678	Doe	John	98765432	john.doe...	12 rue de...	1990-05-...	123456	2030-05-...	<button>Historiq...</button> <button>Historiq...</button> <button>Supprim...</button>

Historique Pénalités

id_penalite	id_location	mode_paiement	etat_paiement
2	1	application D17	effectue
1	1	carte bancaire	effectue

Rafraichir Retour

Interface Consultation Penalite Admin

Bienvenu Admin ,

Mon Compte Gestion Voitures Gestion Employés Gestion Clients Gestion Locations **Consultation Pénalités**

Rafraichir

Id pénalité	Id location	mode_paiement	etat_paiement
1	1	carte bancaire	effectue
2	1	application D17	effectue

Interface Gestion Location Admin :

Bienvenu Admin ,

Mon Compte Gestion Voitures Gestion Employés Gestion Clients **Gestion Locations** Consultation Pénalités

Rafraichir

Id_Location	CIN_Client	Immatriculation	Date Début	Date fin	date_retour	mode_paiement	statut	Action
1	12345678	123TUN4567	2023-04-01 00:00:00...	2023-04-05 00:00:00...	2023-04-06 00:00:00...	carte bancaire	terminee	Valider
3	12345678	123TUN4567	2025-05-03 00:00:00...	2025-05-04 00:09:27...		application D17	en_cours	Valider

Location

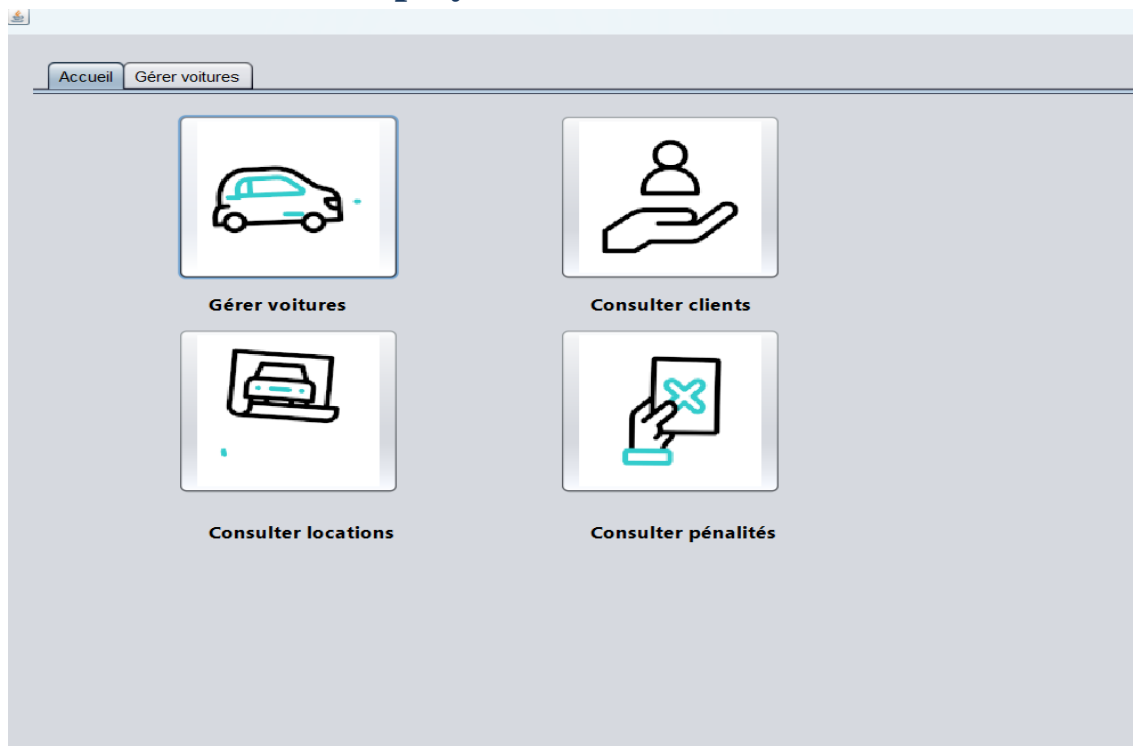
Date de retour :

mai 2025

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
18				1	2	3	4
19	5	6	7	8	9	10	11
20	12	13	14	15	16	17	18
21	19	20	21	22	23	24	25
22	26	27	28	29	30	31	

Connexion à la base de données réussie avec succès!

Interface Accueil Employé



Interface Gestion voiture Employé

The screenshot shows a web application interface for managing vehicles. At the top, there are two tabs: 'Accueil' and 'Gérer voitures' (selected). Below the tabs, there are four dropdown menus for filtering: 'Marque', 'Année', 'Carburant', and 'Transmission'. Below these, there are two price range sliders: 'Prix Min : 20 TND' and 'Prix Max : 41 TND'. There are also four buttons for filtering by status: 'Voitures en marche', 'Voitures en panne', 'Voitures louées', and 'Voitures disponibles'. A 'Rechercher' button is located below the price sliders. An 'Ajouter' button is located at the bottom left. Below the filters, there is a table with the following data:

Immatriculation	Marque	Modèle	Année	État	Disponibilité	Prix	Carburant	Transmission	Date début assu.	Date exp. assu.	ID Assurance	Action
111TUN	VOLSWAGEN	POLO	2020	en_panne	louee	20.0	diesel	automatique	2020-01-01	2022-01-01	0	Modifier
123TUN4567	Toyota	Corolla	2020	en_marche	louee	30.5	essence	automatique	2021-01-01	2022-01-01	0	Modifier
234TUN4568	Honda	Civic	2021	en_panne	indisponible	40.25	diesel	manuel	2022-02-01	2023-02-01	0	Modifier

Accueil

Consulter clients

Consulter

Marque

Prix Min : Prix Min : 60 TND

Ajouter

Immatriculation	Marque
111TUN1111	mercedes
123TUN4569	BMW

Fiche Voiture

Immatriculation :

Marque :

Modèle :

Année :

Etat :

en_marche

Disponibilite :

disponible

Prix jour :

Carburant :

essence

Transmission :

automatique

Date début assurance :

Date expiration assurance :

ID assurance :

1

Enregistrer

Fermer

Interface Consultation locations Employe

Accueil

Consulter clients

Consulter locations

Gérer voitures

Rafraichir

id_location	CIN_client	immatricul...	date_debut	date_fin	date_retour	mode_paie...	statut
2	12345678	123TUN45...	2023-05-01	2023-05-04	2023-05-05	application ...	terminee
3	12345678	111TUN11...	2025-05-03	2025-05-17		application ...	en_cours
4	12345678	111TUN11...	2025-05-03	2025-05-17		application ...	en_cours

Interface Consultation pénalités Employe

Accueil

Consulter clients

Consulter locations

Gérer voitures

Consulter pénalités

Rafraichir

id_penalite	id_location	mode_paiement	etat_paiement
3	2	application D17	effectue
4	2	carte bancaire	effectue

Interface Consultation clients Employe

Accueil

Consulter clients

Rafraichir

CIN	Nom	Prénom	Téléphone	Email	Adresse	Date de naissance	Permis	Date exp. permis	Action
12345678	Doep	Johnn	98765432	john.doe@email.com	12 rue de la Paix, Tunis	1990-05-15	123456	2030-05-15	<div>Pénalités</div> <div>Locations</div>

Rafraichir

CIN	Nom	Prénom	Téléphone	Email	Adresse	Date de naissance	Permis	Date exp. permis	Action
12345678	Doep	Johnn	98765432	john.doe@email.com	12 rue de la Paix, Tunis	1990-05-15	123456	2030-05-15	<div>Pénalités</div> <div>Locations</div>

Historique Locations

Fermer

ID Location	Immatriculation	Date début	Date fin	Date retour	Mode paiement	Statut
2	123TUN4569	2023-05-01	2023-05-04	2023-05-05	application D17	terminee
3	111TUN1111	2025-05-03	2025-05-17		application D17	en_cours
4	111TUN1111	2025-05-03	2025-05-17		application D17	en_cours

6. DIFFICULTES RENCONTREES

Au cours de la réalisation de ce projet, plusieurs difficultés ont été rencontrées :

***Contraintes temporelles :**

Le temps alloué s'est avéré insuffisant au regard de l'ampleur des tâches à accomplir, notamment en période de surcharge académique.

***Fatigue et baisse d'efficacité :**

L'intensité du travail, combinée à un manque de repos, a parfois affecté la productivité de l'équipe.

***Volume de travail important :**

Le projet demandait une implémentation fonctionnelle complète, accompagnée d'une documentation structurée, ce qui a nécessité une gestion rigoureuse des priorités.

***Manque de guidance :**

L'absence d'accompagnement régulier ou de directives précises a engendré certaines incertitudes, notamment dans les choix techniques.

***Problèmes de conception initiale :**

Les premières étapes de modélisation n'étaient pas suffisamment abouties, ce qui a entraîné des ajustements en cours de développement et une perte de temps.

MERCI