

L'entreprise **"AutoMaroc Location"** est une agence de location de voitures disposant de plusieurs succursales à travers le Maroc.

Elle souhaite digitaliser la gestion de ses activités afin de faciliter :

- La réservation des voitures par les clients,
- La gestion du parc automobile,
- La planification et le suivi des locations,
- Ainsi que la facturation et le paiement.

L'objectif du TP est de concevoir puis développer une application web moderne pour automatiser ce processus, en utilisant UML pour la modélisation et React.js pour la réalisation.

Partie 1 : Analyse et modélisation UML

A. Diagramme de cas d'utilisation

Les acteurs principaux :

1. **Administrateur**
 - Gère les voitures (ajout, modification, suppression)
 - Gère les clients et les contrats de location
 - Consulte les statistiques (revenus, taux de location)
2. **Client**
 - Consulte la liste des voitures disponibles
 - Effectue une réservation en ligne
 - Consulte son historique de locations
3. **Agent de location**
 - Valide les réservations
 - Enregistre les paiements
 - Met à jour l'état des voitures

Cas d'utilisation à modéliser

1. Authentification et gestion des rôles
2. Consulter la liste des voitures disponibles
3. Créer un contrat de location
4. Enregistrer un paiement
5. Modifier ou annuler une réservation
6. Gérer les voitures (ajouter, modifier, supprimer)
7. Consulter les statistiques de l'agence (par voiture, par client, par période)

B. Diagramme de classes

Le système doit représenter les entités suivantes avec leurs relations et cardinalités :

- **Voiture** (id, matricule, marque, modèle, catégorie, prixJour, état)
- **Client** (id, nom, prénom, cin, téléphone, email)
- **Location** (id, dateDébut, dateFin, montantTotal, statut)
- **Paiement** (id, datePaiement, montant, modePaiement)
- **Utilisateur** (id, login, motDePasse, rôle)
- **Agence** (id, nom, adresse, ville)

Relations :

- Une **agence** possède plusieurs **voitures**.
- Un **client** peut effectuer plusieurs **locations**.
- Une **location** concerne une seule **voiture**, mais peut avoir un ou plusieurs **paiements**.
- Chaque **utilisateur** appartient à un rôle : *admin*, *agent* ou *client*.

Travail demandé :

- Dessiner le **diagramme de classes UML** complet.
- Indiquer les **cardinalités**, les **associations**, et les **héritages** éventuels (par exemple : Utilisateur → Client / Agent).

C. Diagramme de séquence

Version boîte noire

Scénario : un **client** effectue une **réservation**.

1. Le client se connecte au système.
2. Il consulte la liste des voitures disponibles.
3. Il choisit une voiture et saisit les dates souhaitées.
4. Le système vérifie la disponibilité.
5. Le système crée une location et confirme la réservation.

Version boîte blanche

Détail des interactions entre :

- ClientUI (interface utilisateur)
- ReservationController
- LocationService
- VoitureDAO
- LocationDAO

Travail demandé :

- Réaliser **les deux versions** (boîte noire et boîte blanche) du diagramme de séquence correspondant au scénario "Réserver une voiture".

D. Diagramme d'état-transition

Exemple : Cycle de vie d'une voiture

États possibles :

1. Disponible
2. Réservée
3. Louée
4. En entretien
5. Indisponible

Transitions déclenchées par les événements :

- Réservation validée → "Réservée"
- Contrat signé → "Louée"

- Retour véhicule → “Disponible”
- Panne détectée → “En entretien”

Travail demandé :

- Réaliser le **diagramme d'état-transition** pour l'objet **Voiture**.