

Gestion de Location de Voitures

Chapitre1 : Dossier d'analyse

Introduction

La location de voitures est un secteur économique en pleine expansion qui répond à des besoins variés tant pour les particuliers que pour les professionnels. Dans un contexte où la mobilité devient un enjeu majeur, ce service offre une alternative flexible à l'achat d'un véhicule et s'adapte parfaitement aux besoins ponctuels ou réguliers des utilisateurs.

Contexte du marché

Le marché de la location de voitures se caractérise par une forte concurrence entre des acteurs internationaux et des entreprises locales. La digitalisation des services de location est désormais essentielle pour optimiser la gestion des parcs automobiles et offrir un service client de qualité.

Enjeux

Les principaux enjeux de la gestion de location de voitures sont :

- L'optimisation de la gestion du parc automobile
- La simplification des processus de location
- La facturation précise et transparente des services
- La traçabilité des opérations de location

Avantages de la location

La location de voitures présente de nombreux avantages pour les clients :

- Flexibilité dans le choix du véhicule selon les besoins spécifiques
- Absence d'investissement initial important

- Solution adaptée aux besoins ponctuels de mobilité

Objectifs du projet

Ce projet vise à développer une application simple mais efficace de gestion de location de voitures. Il se concentre exclusivement sur trois fonctionnalités essentielles :

1. Gestion des voitures (CRUD) :

- a. Ajout de nouveaux véhicules au parc
- b. Consultation des caractéristiques des véhicules
- c. Mise à jour des informations des véhicules
- d. Suppression des véhicules du parc

2. Gestion des locations (CRUD) :

- a. Création de nouvelles locations
- b. Consultation des locations en cours et passées
- c. Modification des détails d'une location
- d. Suppression/annulation d'une location

3. Génération de factures :

- a. Création automatique d'une facture après chaque location
- b. Inclusion des informations essentielles (client, véhicule, durée, coût)

Cette application a pour ambition d'apporter une solution simple et efficace aux agences de location de taille modeste qui souhaitent digitaliser leur gestion quotidienne.

Besoins utilisateurs

Exigences fonctionnelles

L'application "**Gestion de Location de Voitures**" devra regrouper les fonctionnalités nécessaires à la gestion des voitures, des locations et à la génération de factures.

1. Gestion des voitures (CRUD) :

R1 : L'agent de location doit pouvoir ajouter un nouveau véhicule dans le système en spécifiant ses caractéristiques (marque, modèle, année, immatriculation, kilométrage, catégorie, tarif journalier).

R2 : L'agent doit pouvoir consulter la liste de tous les véhicules disponibles dans le parc, avec possibilité de filtrer selon différents critères (disponibilité, marque, catégorie).

R3 : L'agent doit pouvoir consulter les détails d'un véhicule spécifique, notamment son historique de locations.

R4 : L'agent doit pouvoir mettre à jour les informations d'un véhicule (kilométrage, état, tarif journalier).

R5 : L'agent doit pouvoir marquer un véhicule comme indisponible (en maintenance, hors service).

R6 : L'agent doit pouvoir supprimer un véhicule du parc lorsqu'il n'est plus en service.

2. Gestion des locations (CRUD) :

R7 : L'agent doit pouvoir créer une nouvelle location en sélectionnant un client, un véhicule disponible, une date de début et une date de fin.

R8 : L'agent doit pouvoir enregistrer les informations du client pour la location (nom, prénom, coordonnées, numéro de permis).

R9 : L'agent doit pouvoir consulter la liste de toutes les locations en cours et passées, avec possibilité de filtrer par date, client, ou véhicule.

R10 : L'agent doit pouvoir consulter les détails d'une location spécifique.

R11 : L'agent doit pouvoir modifier les détails d'une location existante (dates, véhicule, options).

R12 : L'agent doit pouvoir enregistrer le retour d'un véhicule en précisant la date effective, le kilométrage final et l'état du véhicule.

R13 : L'agent doit pouvoir annuler/supprimer une location qui n'a pas encore débuté.

3. Génération de factures :

R14 : Le système doit générer automatiquement une facture à la fin de chaque location.

R15 : La facture doit inclure les informations essentielles : coordonnées du client, détails du véhicule, dates de location, tarif journalier, nombre de jours, montant total, et éventuels suppléments (dépassement kilométrique, retard, dommages).

Exigences de qualité

L'application sera utilisée principalement par des agents de location qui n'ont pas nécessairement de connaissances approfondies en informatique. Pour garantir une utilisation efficace, l'application doit répondre aux exigences de qualité suivantes :

Interface sobre et efficace :

R18 : L'interface utilisateur doit être intuitive, avec des menus clairs et des formulaires simples à remplir.

R19 : Les actions fréquentes doivent être accessibles rapidement, avec un minimum de clics.

R20 : Les formulaires doivent inclure des validations en temps réel pour éviter les erreurs de saisie.

Sécurité :

R23 : Chaque action réalisée dans le système doit être tracée (qui a fait quoi et quand).

Exigences de performance

Une agence de location de taille moyenne nécessite une application réactive et fiable :

R24 : Le système doit supporter au moins 20 connexions simultanées sans dégradation des performances.

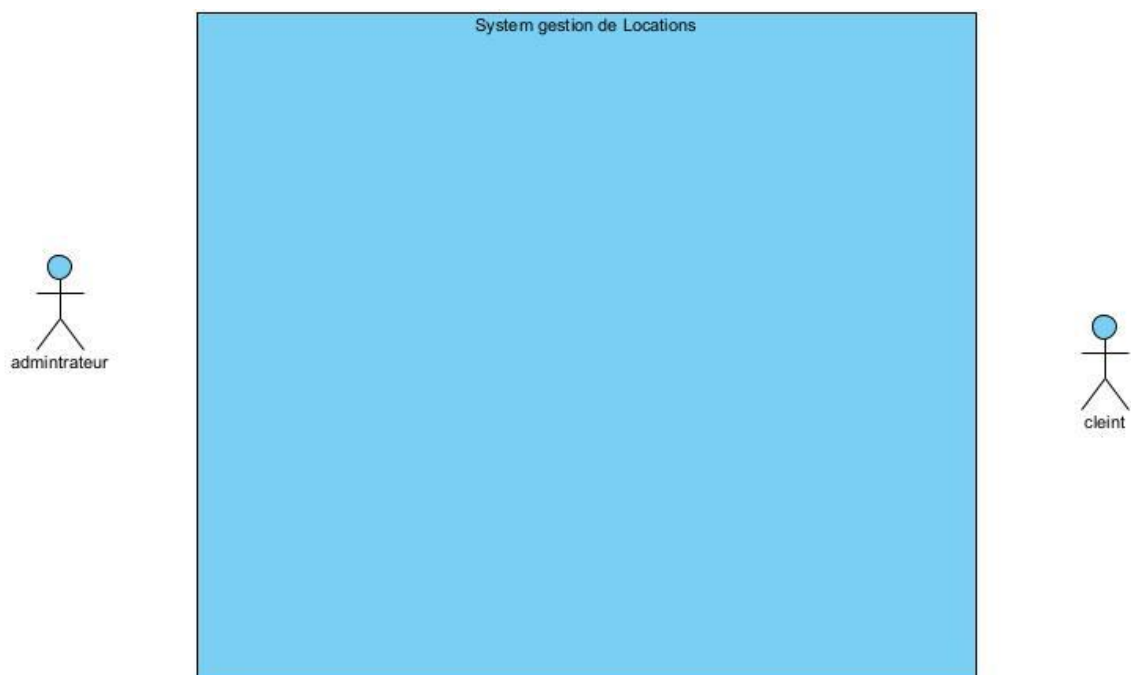
R25 : Le temps de réponse pour les opérations courantes ne doit pas excéder 3 secondes.

R26 : Le système doit pouvoir gérer une base de données contenant jusqu'à 500 véhicules et 5000 locations.

Cas d'utilisation

Identification des acteurs

- Client : Personne qui demander le service de l'agence de Locations
- **Administrateur** : Employé de l'agence qui gère les voitures, enregistre les locations et génère les factures.



Identification des cas d'utilisation

À partir des exigences précédentes, nous pouvons identifier les cas d'utilisation principaux suivants :

UC1 : Gestion des voitures :

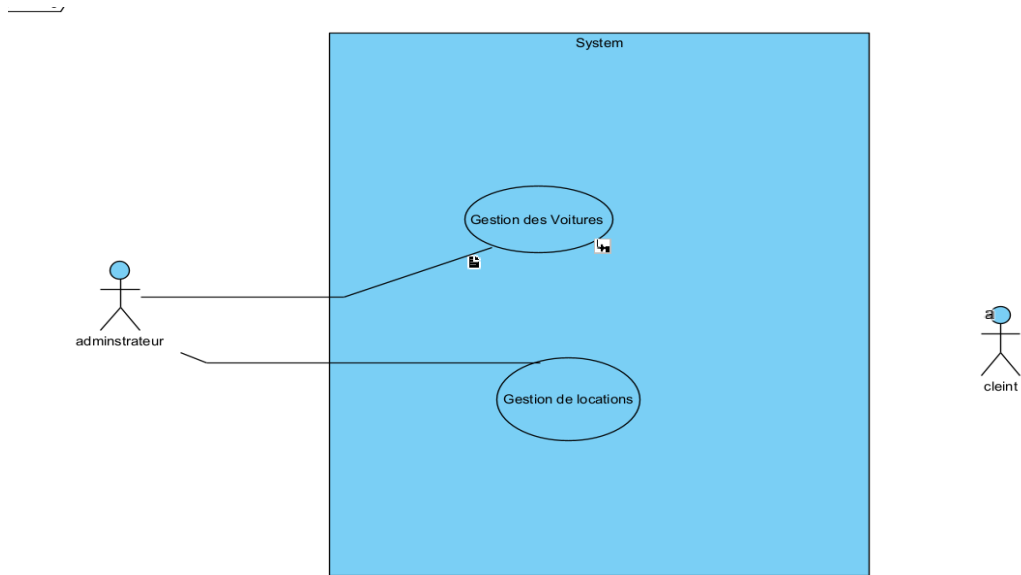
- Ajouter un véhicule
- Consulter les véhicules
- Modifier un véhicule
- Supprimer un véhicule

UC2 : Gestion des locations :

- Créer une location
- Consulter les locations
- Modifier une location
- Enregistrer un retour de véhicule
- Annuler une location

Exigence Fonctionnelle	Intention d'acteur	Acteur
Gestion des voitures	Ajouter un véhicule Consulter les véhicules Modifier un véhicule Supprimer un véhicule	Administrateur
Gestion des locations	Créer une location Consulter les locations Modifier une location Enregistrer un retour de Véhicule Annuler une location	Administrateur

Diagramme de cas d'utilisation :



Classement des cas d'utilisation

Cas d'utilisation	Priorité	Risque
UC1 : Gestion des voitures	Haute	Bas
UC2 : Gestion des locations	Haute	Moyen

Planification du projet

Cas d'utilisation	Priorité	Risque	Itération
UC1 : Gestion des voitures	Haute	Bas	1
UC2 : Gestion des locations	Haut	Moyen	2

Chapitre2 : Dossier de Conception

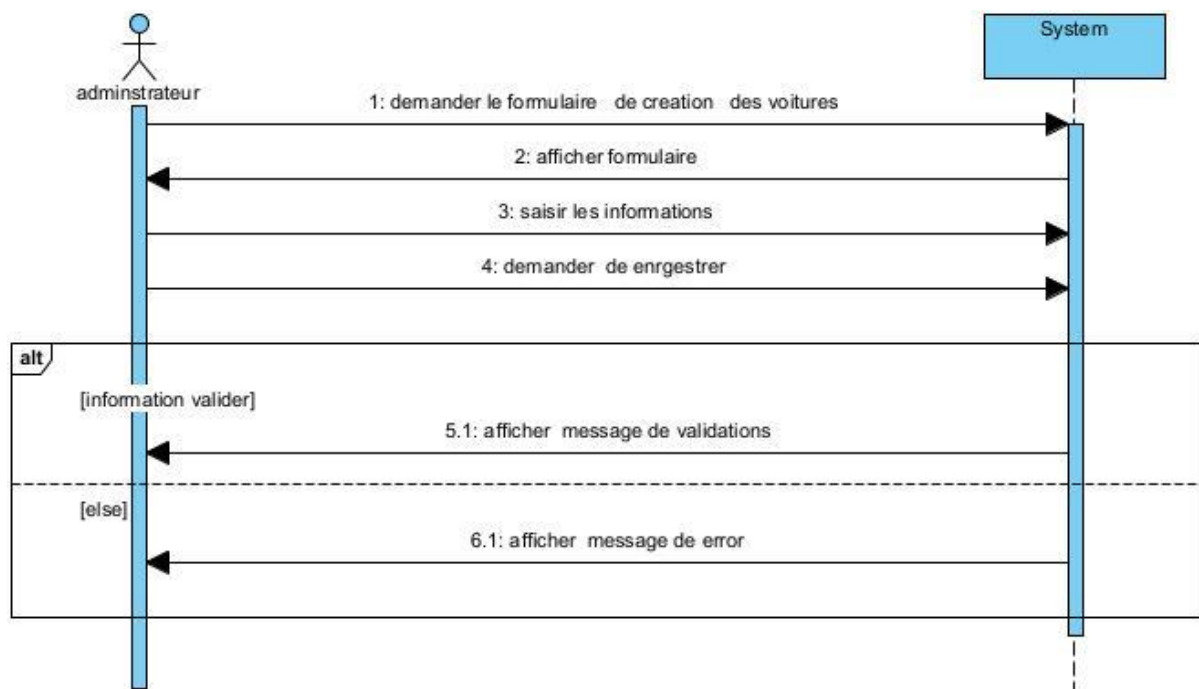
Itération #1 UC1 : Gestion des voitures

Diagramme de séquence système de UC1

1.Flow of Events :

1. demander le formulaire d'ajouter des voitures
2. SYSTEM Afficher le formulaire de la demande
3.Saisir les informations
4. if Informations valides
4.1. SYSTEM Afficher message de validations
5. else
5.1. SYSTEM Message d'erreur
end if

2. Diagramme de séquence :



3.Plan de Test

Tests pour "Ajouter un véhicule"

Cas de test 1 : Ajouter un véhicule avec des données valides → Vérifier que l'enregistrement est réussi.

Cas de test 2 : Ajouter un véhicule avec un champ obligatoire manquant → Vérifier que le système affiche un message d'erreur.

Cas de test 3 : Ajouter un véhicule avec un numéro d'immatriculation déjà existant → Vérifier que le système empêche la duplication.

Tests pour "Consulter les véhicules"

Cas de test 4 : Afficher la liste des véhicules → Vérifier que tous les véhicules enregistrés sont affichés.

Cas de test 5 : Rechercher un véhicule par marque → Vérifier que seuls les véhicules correspondants sont affichés.

Tests pour "Modifier un véhicule"

Cas de test 6 : Modifier un véhicule avec des données valides → Vérifier que les modifications sont bien enregistrées.

Cas de test 7 : Modifier un véhicule avec un champ obligatoire vide → Vérifier que le système affiche un message d'erreur.

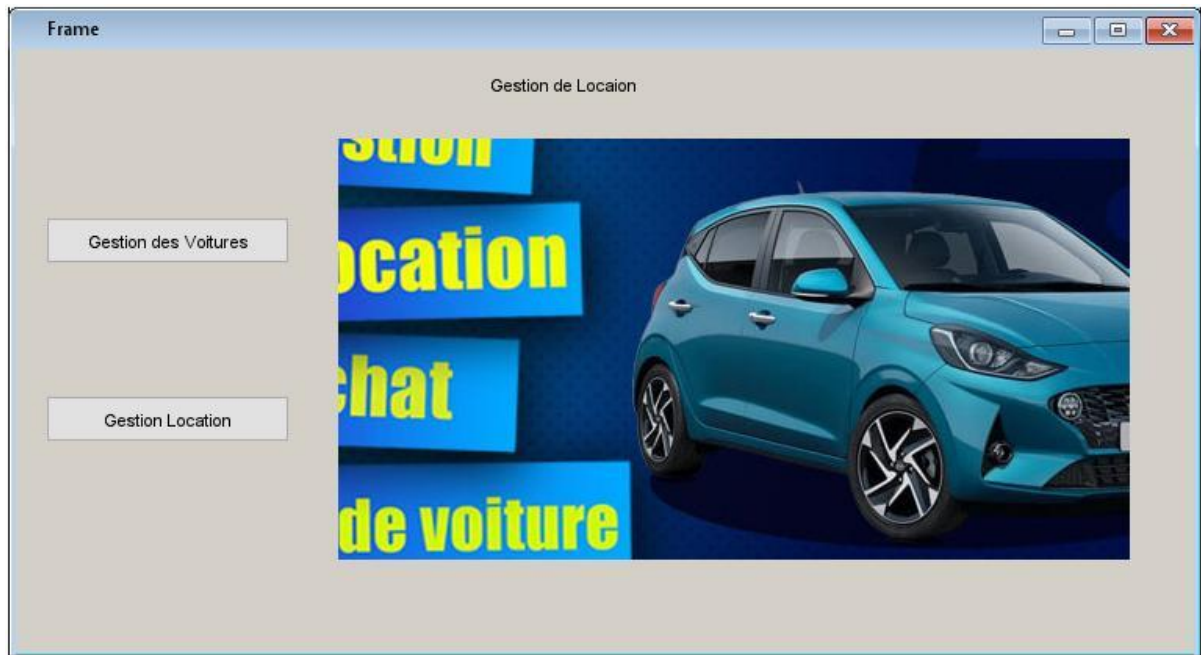
Tests pour "Supprimer un véhicule"

Cas de test 8 : Supprimer un véhicule existant → Vérifier que le véhicule est bien supprimé de la base de données.

Cas de test 9 : Supprimer un véhicule inexistant → Vérifier que le système affiche un message d'erreur.

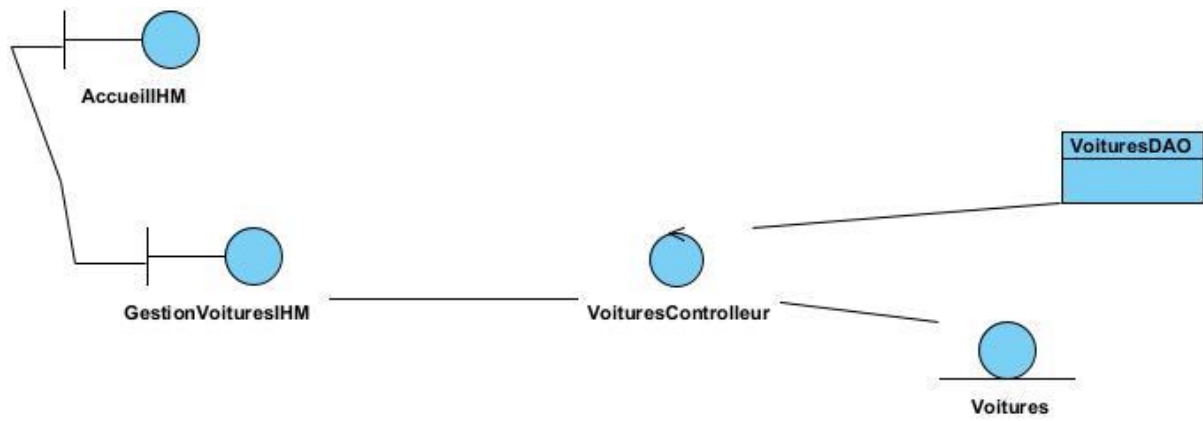
4. User Interface diagramme :

3.1 Interface de page d'accueil:

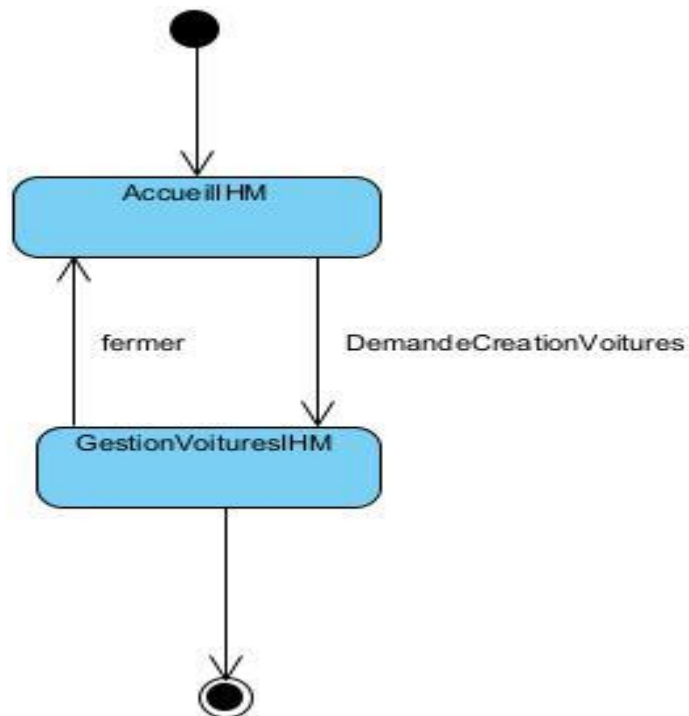


3.2 Interface de gestion des Voitures :

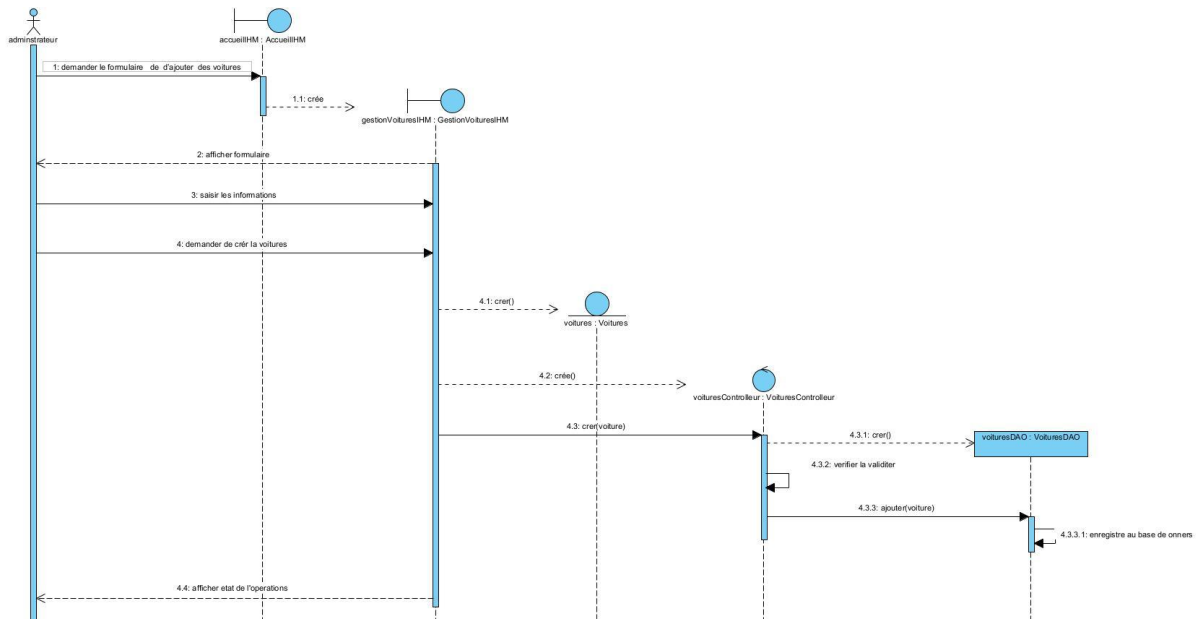
5.Diagramme Classes participante :



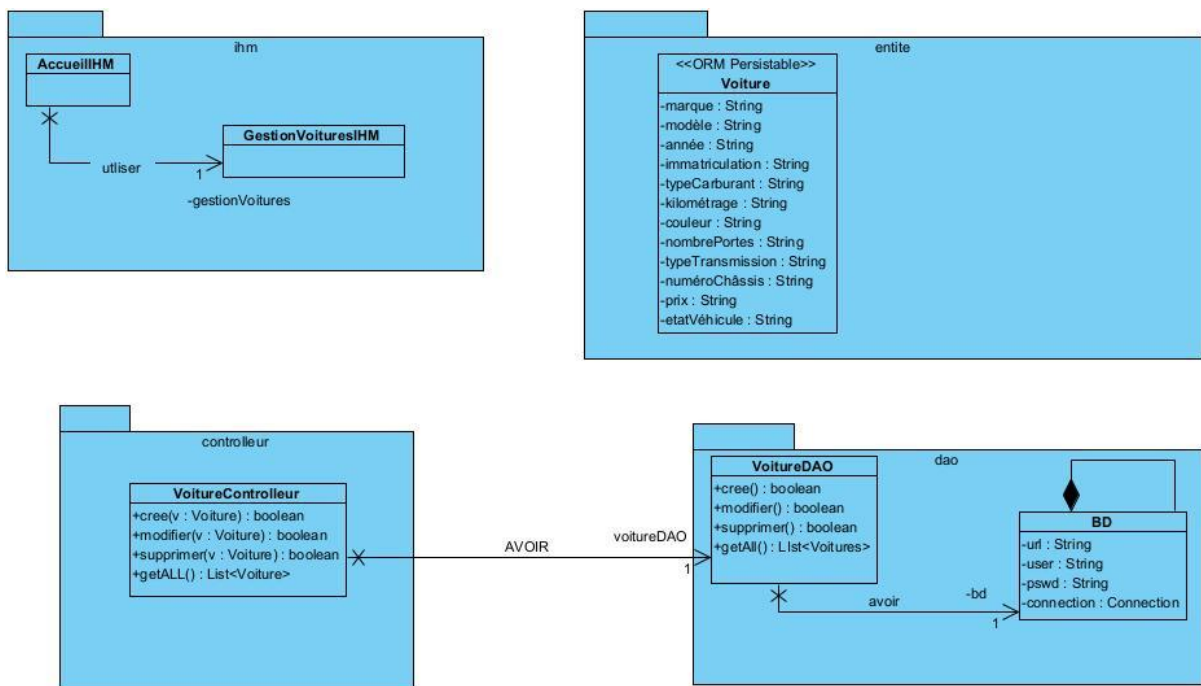
4. Diagramme Navigations :



6. Diagramme d'interactions :



7.Diagramme de classe :



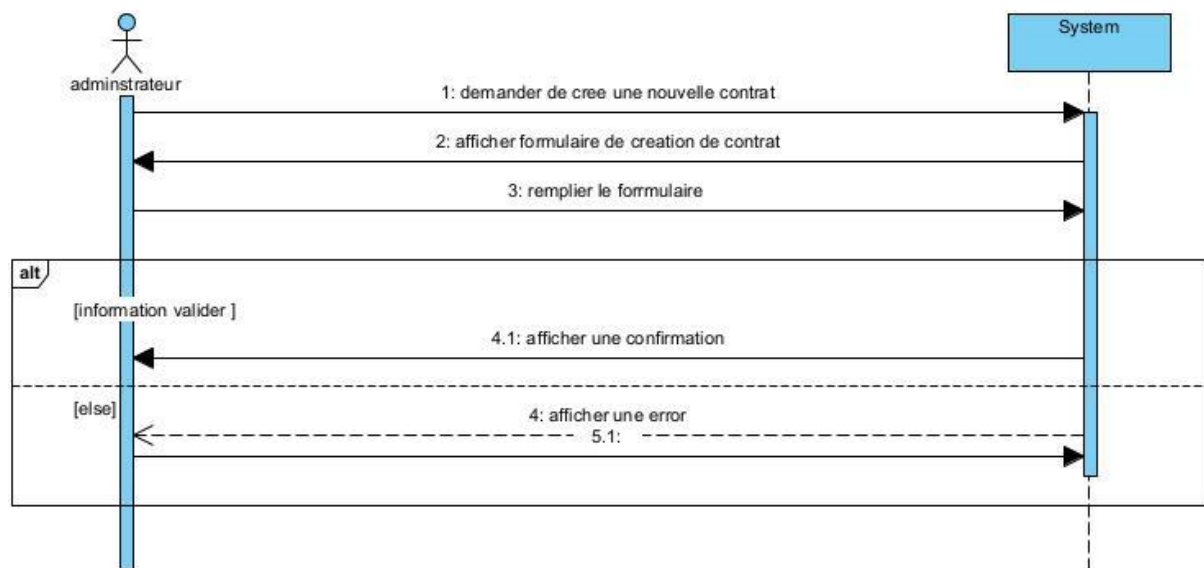
Itération #2 UC2 : Gestion des Contrat

1.Flow of Events :

1. demander de crée une nouvelle contrat
2. **SYSTEM** Afficher le formulaire de contrat
- 3.Saisir les informations

4. if Informations valides
4.1. SYSTEM Afficher message de validations
5. else
5.1. SYSTEM Message d'erreur
end if

2. Diagramme de séquence :



3. Plan de Test

Scénarios de test

3.1. Création d'une location

Cas de test 1 : Création avec des données valides

→ **Préconditions** : Un client et un véhicule disponibles dans la base de données

→ **Actions** :

1. Ouvrir l'interface de création d'une location
2. Sélectionner un client
3. Sélectionner un véhicule
4. Définir les dates de location

5. Valider la création

→ **Résultat attendu** : La location est enregistrée et un message de confirmation s'affiche.

Cas de test 2 : Création avec des champs obligatoires manquants

→ **Résultat attendu** : Affichage d'un message d'erreur

3.2. Consultation des locations

Cas de test 3 : Affichage de la liste des locations

→ **Actions** :

1. Aller sur la page de gestion des locations
2. Vérifier que toutes les locations enregistrées s'affichent correctement

→ **Résultat attendu** : Toutes les locations existantes apparaissent avec leurs informations

3.3. Modification d'une location

Cas de test 4 : Modification d'une location existante

→ **Préconditions** : Une location existe déjà

→ **Actions** :

1. Sélectionner une location
2. Modifier une ou plusieurs informations (ex. : dates)
3. Valider la modification

→ **Résultat attendu** : Les nouvelles informations sont enregistrées et mises à jour dans la base de données

3.4. Enregistrement du retour d'un véhicule

Cas de test 5 : Enregistrer le retour d'un véhicule

→ **Préconditions** : Un véhicule est actuellement loué

→ **Actions** :

1. Sélectionner une location en cours
2. Cliquer sur "Retourner le véhicule"

3. Vérifier que le statut de la location passe en “Terminée”
→ **Résultat attendu** : Le véhicule est marqué comme retourné

3.5. Annulation d’une location

Cas de test 6 : Annulation d’une location en cours

→ **Préconditions** : Une location est en cours mais pas encore terminée

→ **Actions** :

1. Sélectionner la location
2. Cliquer sur “Annuler”
→ **Résultat attendu** : La location est annulée et le véhicule redevient disponible

Critères de réussite

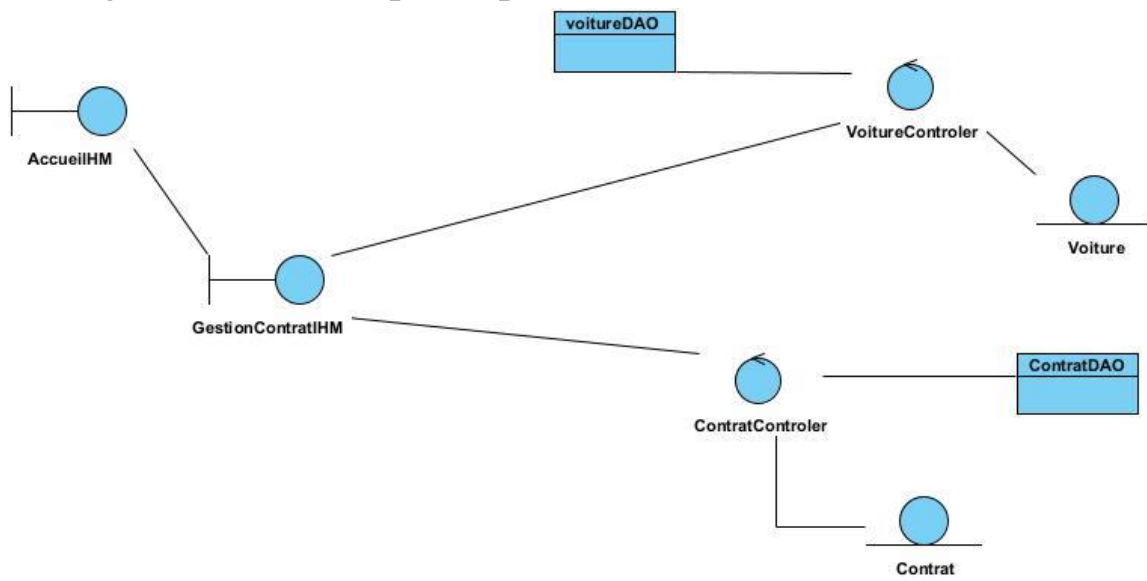
- 100% des tests fonctionnels doivent être validés
- Les messages d’erreur doivent être affichés en cas d’erreur utilisateur
- Les modifications doivent être correctement sauvegardées

4. User Interface diagramme :

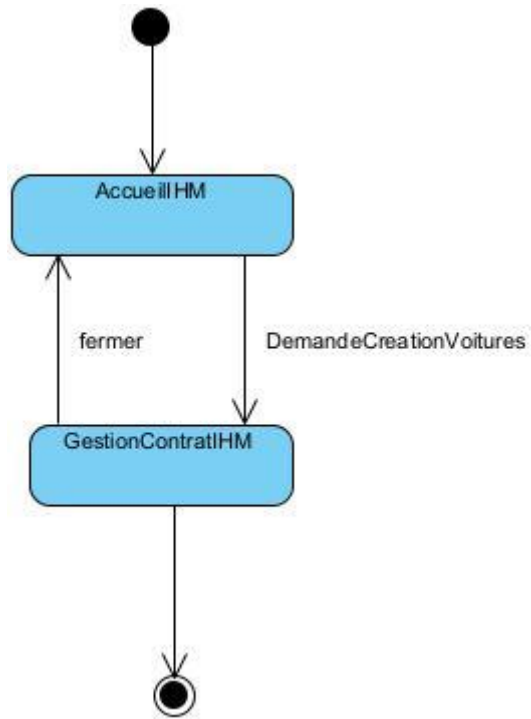
3.1 Interface de gestion des Contrat :

The screenshot displays a web application window titled "Gestion des Contrat". The interface includes several input fields for user details: "nom", "prenom", "CIN", "Contrat", "licence Conduit", "Etat d'locations" (with radio buttons for "au cours" and "Terminer"), "Date de debut", "Date de fin", and "PRIX". On the right side, there is an "Actions" section with buttons for "Ajouter", "Modifier", "Modofier", and "Details". At the bottom, there are two search sections: "Rechercher contrat" and "Rechrcer Voitures", each with a search input field and a table with columns "Name", "Date modified", "Type", and "Size". The tables show "File 1", "File 2", "File 3", and "...".

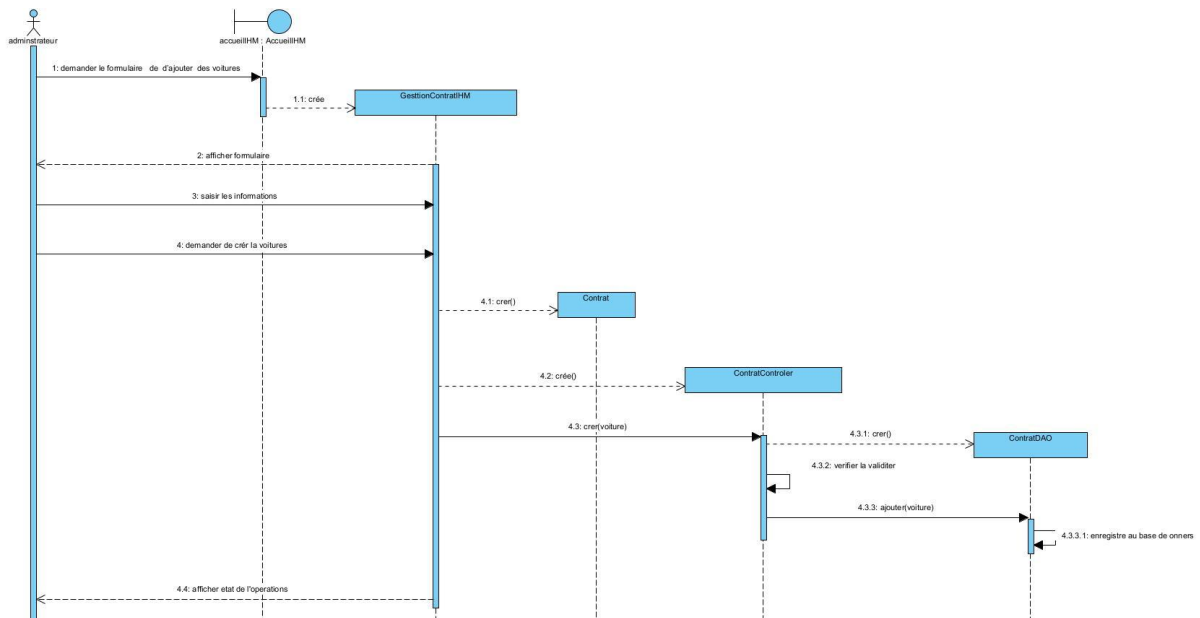
5.Diagramme Classes participante :



5.Diagramme Navigations :



6. Diagramme d'interactions :



7. Diagramme de classe :

