

Cas d'utilisations

Modification des paramètres par le gérant

1. Le superviseur veut modifier les paramètres	
2. Le patron se connecte à la plateforme de gestion.	3. Le système offre le menu employé avec options de gestionnaire.
4. Le superviseur sélectionne de modifier les paramètres.	5. Le système offre une sélection de catégories de paramètres.
6. Le superviseur sélectionne la catégorie à modifier.	7. Le système affiche les valeurs de la catégorie ainsi que les valeurs courantes.
8. Le patron modifie les valeurs courantes.	
9. Le patron sélectionne "sauvegarder"	10. Le système sauvegarde les valeurs et retourne au menu des catégories (Étape 5).
11. Le patron quitte sa session	

* À tout moment, le superviseur peut annuler le processus de modification.

6A. Dans le cas où le superviseur sélectionne la mauvaise catégorie.

- 1- Le patron sélectionne la mauvaise catégorie
- 2- Le système affiche le menu suivant.
- 3- Le patron sélectionne le bouton de retour.
- 4- Le système affiche le menu précédent.

9A. Dans le cas où le superviseur entre une donnée erronée.

- 1- Le superviseur sauvegarde les données avec une erreur.
- 2- Le système avertit qu'une erreur est présente.
- 3- L'utilisateur fait la correction et soumet les modifications.

Ajouter un véhicule

1. Le patron veut ajouter un véhicule à la flotte.	
2. Le patron se connecte à la plateforme.	3. Le système offre le menu de l'utilisateur.
4. Le patron sélectionne l'option d'ajouter un véhicule.	5. Le système affiche le menu d'ajout.
	6. Le système demande les informations du véhicule.
7. Le patron entre les informations.	
8. Le patron sauvegarde l'information	9. Le système présente l'information afin de la confirmer.
10. Le patron confirme l'information.	11. Le système sauvegarde l'information.
	12. Le système retourne le menu principal.
13. Le patron sélectionne de quitter.	

* En tout temps, le patron peut annuler l'ajout d'un véhicule.

4A. Mauvaise sélection de menu

- 1- Le patron sélectionne le mauvais menu.
- 2- Le système affiche le menu sélectionné.
- 3- La patron clique sur le bouton de retour.
- 4- Retour à #4.

8A. Mauvaise information comme entrée.

- 1- La patron a entré des informations non-conformes.
- 2- Le patron sélectionne "sauvegarder".
- 3- Le système détecte une erreur.
- 4- Le patron modifie l'information erronée.
- 5- Le patron sauvegarde.
- 6- Continuation à #9.

Réserver un véhicule

1. Un client veut réserver un véhicule.	
2. Le commis entre ses informations pour se connecter.	3. Le système vérifie le login.
	4. Le système affiche le menu d'options
5. Le commis sélectionne "Connexion au compte client"	6. Le système offre l'outil de recherche pour trouver le client.
7. Le commis sélectionne le compte client	8. Le système offre les options du compte client.
9. Le commis sélectionne "Réserver un véhicule"	10. Le système demande de sélectionner la durée de location.
11. Le commis entre les dates de début et de fin de location.	12. Selon la disponibilité, le système offre une sélection de classes de véhicule.
13. Le commis sélectionne la classe de véhicule.	14. Le système, toujours selon la disponibilité, offre une sélection de véhicule contenu dans la classe
15. Le commis sélectionne le véhicule.	16. Le système confirme la réservation et met la location en suspens. Il met le véhicule indisponible pour la durée sélectionnée.
17. Le commis ferme le compte client.	18. Le système affiche le menu employé.
19. Le commis ferme sa session.	

* En tout temps, l'employé peut annuler la réservation.

3A. Mauvaises informations de connection.

- 1- Le commis entre les mauvaises informations.
- 2- Le système constate une erreur dans les informations.
- 3- Le système affiche une erreur.
- 4- Retour à #2.

5A. Le client ne possède pas de compte.

- 1- L'employé sélectionne création d'un compte client.
- 2- Le système demande les informations du nouveau client.
- 3- L'employé entre les informations du client.
- 4- L'employé clique sauvegarder client.
- 5- Retour au #8.

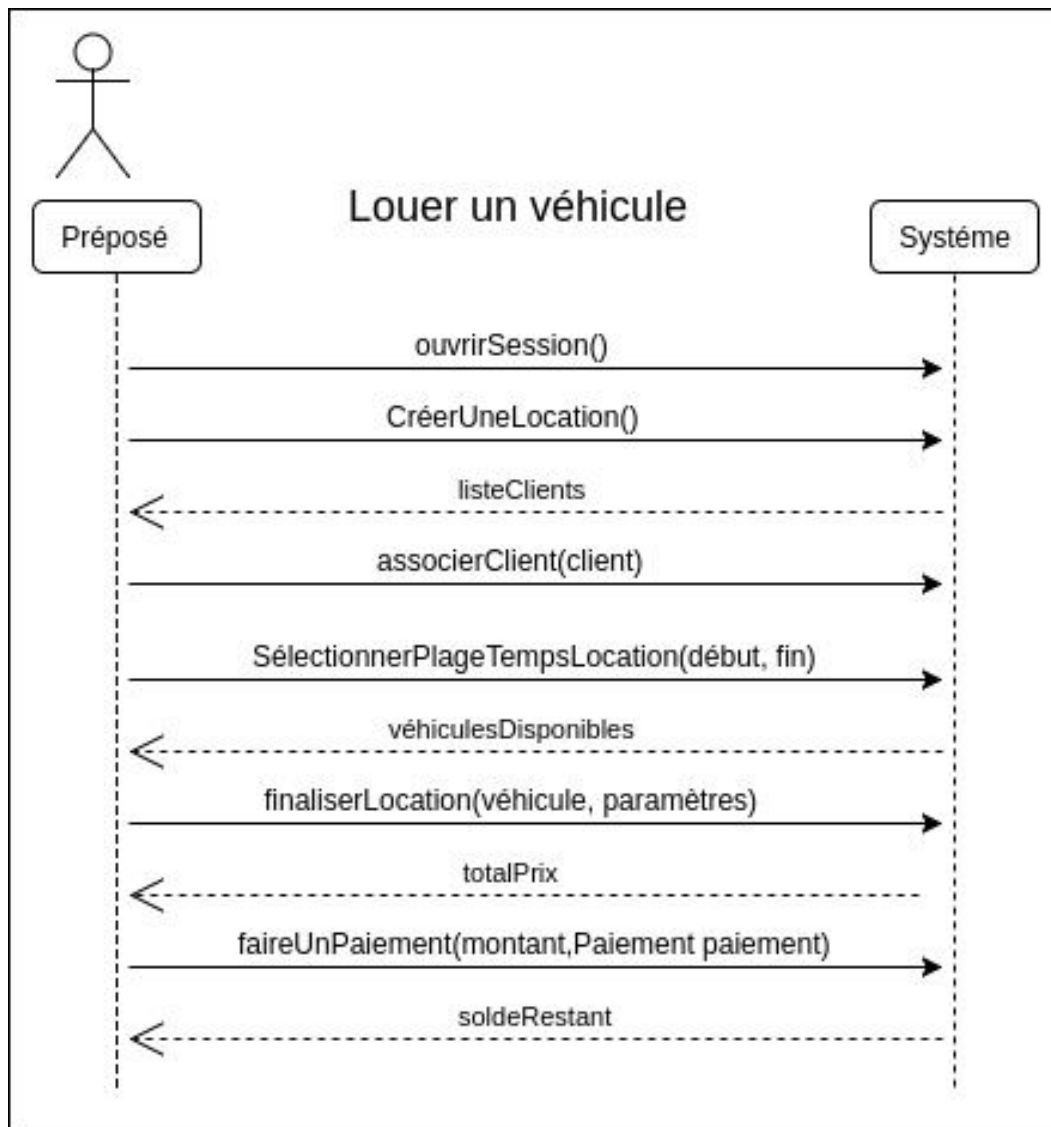
12A. Il ne reste plus aucun véhicule pour ce laps de temps.

- 1- Le système affiche une erreur indiquant la non disponibilité d'un véhicule.
- 2- Le système retourne à la sélection de la date.

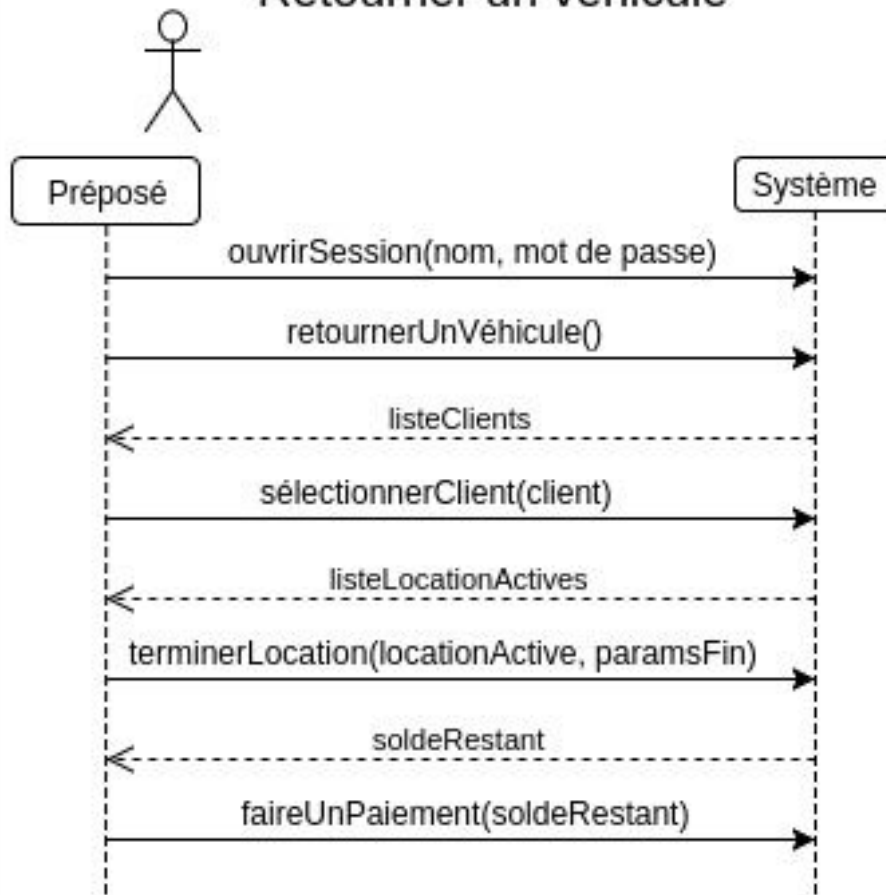
3- Retour # 10.

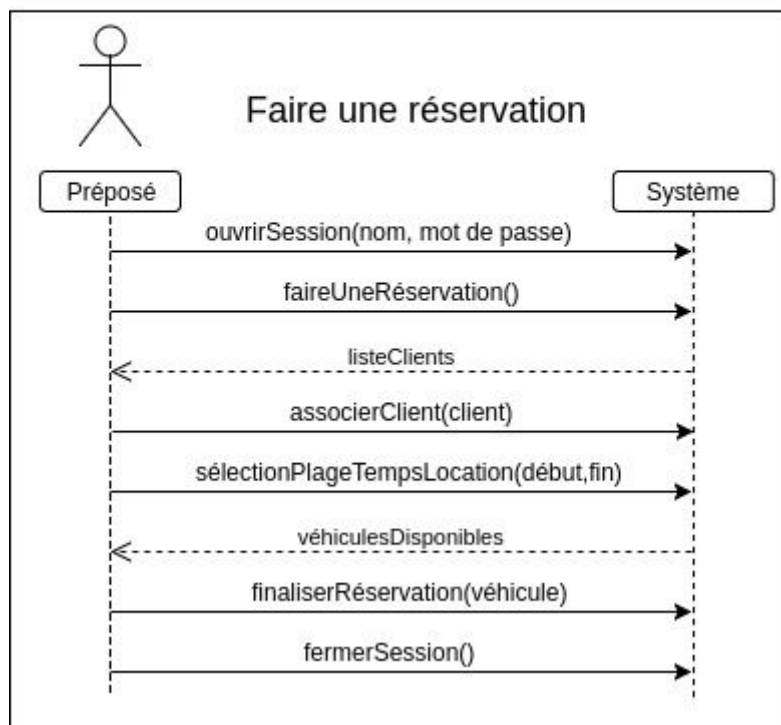
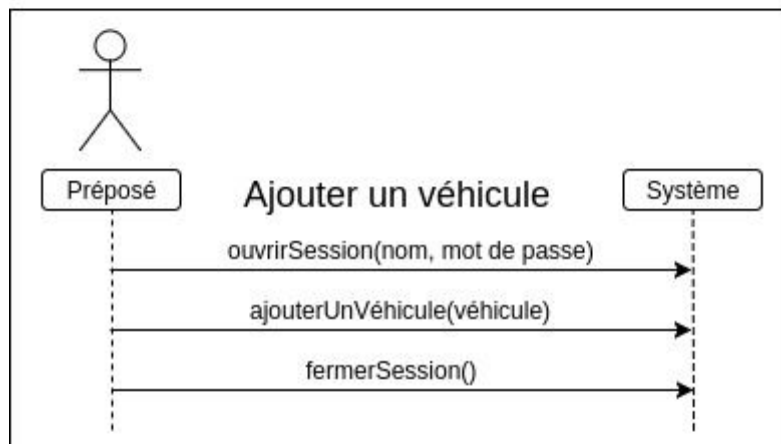
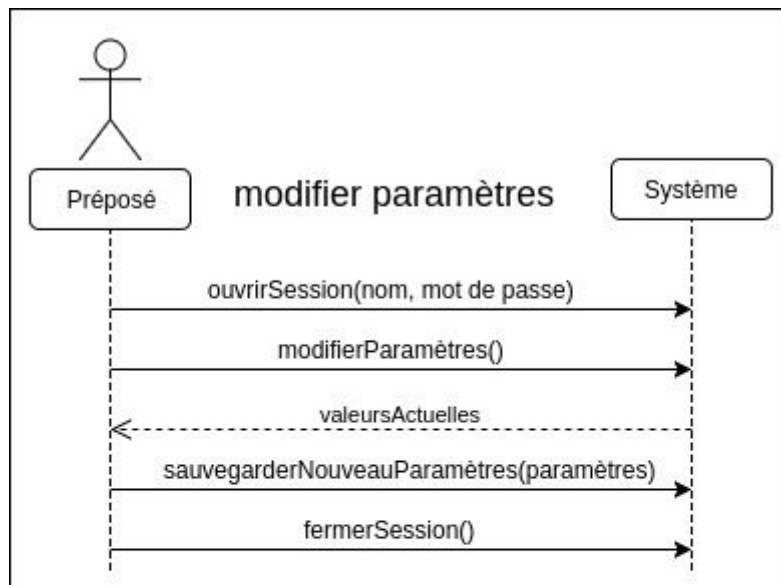
Précision sur la reprise d'une réservation : Une réservation devrait être reprise comme étant une location lorsque le client se présente pour le véhicule et la continuation se fait comme une location.

Diagrammes de séquence du système



Retourner un véhicule





Conclusions concernant les diagrammes de séquence système

Les DSS permettent d'identifier quelques fonctionnalités qui pourront être implémentés sous forme de méthodes.

Le système nécessite un login afin de pouvoir interagir avec celui-ci, pour des raisons de sécurité et de gestion.

La fonction **ouvrirSession(nom, mot de passe)** s'occupe de gérer la connection au système.

Une fois connecté, le système offre des options d'opérations. Chaque opération a sa méthode qui s'occupe de créer l'objet particulier et de sauvegarder les données temporelles ainsi que l'identifiant d'employé responsable de l'opération.

Parmi ces opérations, on retrouve :

- **créerUneLocation()**
- **retournerUnVéhicule()**
- **modifierParamètres()**
- **faireUneRéservation()**
- **ajouterVéhicule(véhicule) **Cette classe nécessite seulement cette étape.**

Lors que le préposé effectue une opération concernant une location, celle-ci doit être associé à un client particulier. La fonction **associerClient()** permet d'effectuer cette tâche.

Pour le cas créerUneLocation(), une fois le client trouvé, le préposé entre les dates et le système détermine ce qui est disponible. Le préposé choisit ce qui est disponible en accord avec les désirs du client et finalise la création avec **finaliserLocation(véhicule, paramètres)**.

Pour le cas retournerUnVéhicule(), le commis sélectionne le client parmi une liste. Lorsque fait, le préposé choisit parmi une liste de location(s) active(s) du client. Pour ce faire, la fonction **terminerLocation(locationActive, paramsFin)** est utilisée, paramsFin représente le kilométrage final ainsi que l'inspection, etc.. . Le système envoie le solde restant et le commis peut collecter le paiement avec **faireUnPaiement(solde)**.

Pour le cas modifierParamètres(), le système envoie les anciens paramètres pour que l'interface les affiche. L'employé change ensuite les valeurs qu'il veut changer et fait appel à la fonction **sauvegarderNouveauParamètres(paramètres)**.

Pour le cas faireUneRéservation(), le procédé est le même que la location sauf qu'on ne se rend pas jusqu'à la fin. On arrête après avoir sélectionné le véhicule.

Pour se déconnecter, l'utilisateur utilise **fermerSession()**.

Identification des patrons de conception

En analysant le modèle du domaine, on peut cerner la présence de quelques patrons de conception.

Le patron expert en information est présent à quelques endroits :

- Location doit être experte en information concernant le paiement car elle connaît le nombre de véhicules loués ainsi que la durée de location et les modalités.
- Les locations sont contenues dans le RegistreLocation, RegistreLocation a accès à toutes les locations actives et les locations inactives, il pourra donc déterminer avec précision les disponibilités des véhicules.

Le patron de création :

- Magasin est créateur de véhicule et destructeur aussi. La liste de véhicule est contenu dans magasin.
- Location contient et créer paiement.
- Magasin contient les clients et doit donc les créer.
- RegistreLocation doit créer nouvelle location

Pour éviter le couplage inutile :

- RegistreLocation contient les locations et location contient paiement, ainsi, si une modification doit être apportée, RegistreLocation demande à la location particulière d'effectuer un changement à son objet paiement. On évite donc des situations de concurrence où un objet modifie quelque chose et qu'un autre objet en ignore les circonstances.

Le patron contrôleur sera appliqué :

- L'interface graphique envoie tous les événements à la classe Magasin et celui-ci décide des opérations à effectuer afin de répondre au besoins de l'utilisateur.