

Rapport du projet UML : Co-voiturage

CO-VOITURAGE

BOURICHA MANAL – OUEDI CHAIMAE - ABSSOUS YASSINE – LAANIBI MARWANE



INTRODUCTION :

Nous avons eu l'opportunité de mettre en pratique nos connaissances et compétences acquises lors du cours d'analyse et conception UML. En réalisant une application dont son thème principal est le co-voiturage.

Le covoiturage est une solution de transport alternative à l'autosolisme qui permet d'augmenter les taux d'occupation des véhicules et par conséquent lutter contre la congestion automobile et réduire la pollution atmosphérique.

Mais la spécificité de notre application est qu'elle dédiée plutôt à un usage scolaire ; cela veut dire qu'elle concerne en premier temps les élèves et leurs parents.



1. SPECIFICATIONS DES BESOINS :

1. Besoins fonctionnels :

Un besoin fonctionnel spécifie l'action qu'un système doit être capable d'effectuer, hors contrainte physique : besoin spécifiant un comportement d'entrée/sortie d'un système.

Dans ce contexte notre application de covoiturage scolaire, implémente principalement les fonctionnalités suivantes:

- **Module des inscriptions** : Le système doit permettre aux conducteurs et passagers de pouvoir s'inscrire de façon autonome via le site web.
- **Module de gestion des trajets** : Le système doit permettre l'ajout, modification et suppression d'un trajet par un conducteur ou un passager. Ces derniers auront accès à tous leurs trajets passés et pourront créer de nouveaux trajets à partir d'un trajet déjà effectué dans le passé.
- **Module de gestion des réservations** : Le système doit aussi permettre aux covoitureurs de rechercher des trajets. Des recherches peuvent s'effectuer suivant des critères précis. Suite à l'affichage des résultats le passager peut choisir parmi les plus « intéressants » pour lui. En cas de changement de plan, un système de notification envers le conducteur ou le passager est disponible (messagerie), afin d'aviser que le passager n'embarque plus avec le conducteur ou que le conducteur souhaite annuler son trajet.
 - **Module de gestion des comptes** : Le système permet à chacun des membres de faire la gestion de son compte. Il sera possible de modifier des informations personnelles ou préférences.

2. Besoins non fonctionnels

Un besoin non fonctionnel est besoin spécifiant des propriétés du système, telles que les contraintes liées à l'environnement et l'implémentation, les exigences en matière de performances, de dépendances de plate-forme, de facilité de maintenance, d'extensibilité et de fiabilité.

Dans notre système on distingue les besoins non fonctionnels suivant :

- **Sécurité** : La plateforme doit assurer la sécurité pour les utilisateurs (Authentification).
- **Convivialité** : Ergonomie des interfaces homme machine et facilité d'utilisation.
- **Contrôle** : Saisie contrôlée selon les choix prédéfinis.



II ANALYSE DES BESOINS :

1. Identification des acteurs du système :

Un acteur est une personne, un matériel ou un logiciel qui interagit directement avec le système pour réaliser une tâche. Ainsi, un acteur peut consulter et/ou modifier directement l'état du système en émettant et/ou recevant des messages susceptibles contenir des données. Durant notre analyse nous avons identifiés les acteurs suivants :

- **Administrateur** : L'administrateur est le seul à avoir accès à l'information complète du système. Il peut faire la gestion globale du système, par exemple créer des comptes de covoitureurs. Il lui est aussi possible de suspendre n'importe quel compte.
- **Conducteurs** : Les conducteurs permettent d'alimenter le système en itinéraires disponibles pour la recherche. En effet, c'est la quantité et qualité des conducteurs qui permettront d'avoir une banque de données intéressante dans le système. Suite à la publication d'un trajet, ils sont responsables de vérifier à intervalle régulier s'ils reçoivent des demandes d'embarquement des covoitureurs. Dans un tel cas, ils sont responsables de confirmer l'embarquement du passager à un point donné. De plus, ils peuvent consulter leur historique trajets.
- **Passagers** : Les passagers sont ceux qui bénéficient le plus du système. Ils peuvent lancer diverses recherches pour trouver des trajets qui conviennent à leurs besoins et faire des réservations. Dans le cas où les points d'embarquements conviennent plus ou moins, ils peuvent envoyer une proposition de point d'embarquement au conducteur. De plus, ils peuvent consulter leur historique de réservations.



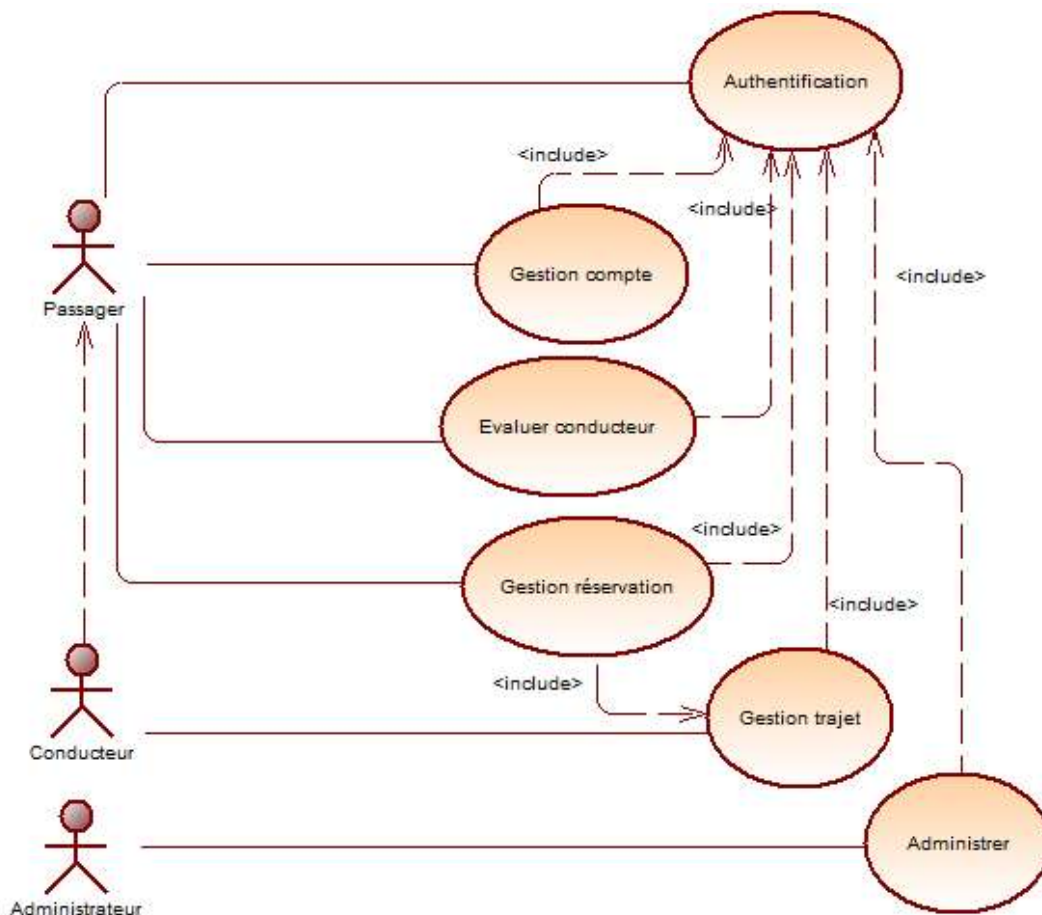
III CONCEPTION :

1. IDENTIFICATION DES CAS D'UTILISATION :

Un cas d'utilisation est une fonctionnalité de système qui produit un résultat observable pour un utilisateur potentiel du système. Le cas d'utilisation regroupe une famille de scénario ou chaque scénario est un traitement particulier du système.

Lors de notre analyse des besoins nous avons pu identifier les actions importantes que nous présenterons ci-dessous et nous les modélisons par la suite avec les diagrammes cas utilisation d'UML.

Diagramme :





Description textuelle des cas d'utilisation :

3. Création Compte _Gestion compte :

Objectif : Ouvrir un compte dans l'application.

Acteurs : Passager, Conducteur.

Pré-condition : l'acteur ne doit pas avoir déjà un compte.

Post-condition : Créer un compte correct qui permet la connexion, et l'accès aux services de l'application.

Scénario nominal :

- 1) Remplir les champs du formulaire (Nom utilisateur, mot de passe ...)
- 2) Soumettre le formulaire
- 3) Vérification des informations par le système, et l'envoi d'un message de confirmation (Email, code par numéro de téléphone ...)
- 4) Confirmation des acteurs concernés.

Scénario alternatif :

- 1) a. Si les champs obligatoires sont vides, ou les données entrées ne vérifient pas la forme demandée : un message d'erreur est affiché, et demande de ressaisie.
- 3) a. Si le compte est déjà existant : envoi d'un message d'erreur.
- 4) a. Demande de connexion ou de sortie à l'acteur.

4. Consulter compte _Gestion compte :

Objectif : Pouvoir consulter les informations du compte ainsi que celles personnelles (profil utilisateur).

Acteurs : Passager, Conducteur.

Pré-condition : Etre authentifié.

Scénario nominal :

- 1) Afficher les données et informations désirées en réponse à la demande de l'utilisateur.

5. Modifier compte _Gestion compte :

Objectif : Pouvoir effectuer des modifications sur le compte et ses informations.

Acteurs : Passager, Conducteur.

Pré-condition : Etre authentifié.

Post-condition : Les informations du compte sont modifiées.

Scénario nominal :

- 1) Modifier les informations souhaitées, en entrant les nouvelles valeurs.
- 2) Vérification du système et sauvegarde.

Scénario alternatif :



- 2) a. Si les champs modifiés ne respectent pas la syntaxe imposée ou sont vides alors renvoie d'un message d'erreur et demande de ressaisie.

6. Suppression d'un compte _Gestion d'un compte :

Objectif : Pouvoir supprimer un compte.

Acteurs : Passager, Conducteur.

Pré-condition : Etre authentifié.

Post-condition : Le compte est supprimé, en le gardant dans l'archive.

Scénario nominal :

- 1) Demande de suppression par l'acteur.
- 2) Le système demande la confirmation.
- 3) Affirmation de l'acteur.

Scénario alternatif :

- 3) a. Si l'acteur annule la demande en déclinant la confirmation, la suppression du compte est annulée à son tour.

7. Authentification :

Objectif : Se connecter au système et bénéficier de ces services.

Acteurs : Passager, Conducteur.

Pré-condition : Posséder un compte.

Post-condition : La connexion, et l'accès effectif au système.

Scénario nominal :

- 1) L'entrée du nom d'utilisateur et mot de passe et confirmation.
- 2) Vérification du système.

Scénario alternatif :

- 2) a. Si les entrées sont erronées, demande de ressaisie.

8. Proposer trajet _Gestion trajet :

Objectif : Créer un trajet, et le proposer aux utilisateurs de l'application.

Acteurs : Conducteur.

Pré-condition : Etre authentifié.

Post-condition : Le trajet est créé et ajouté à la liste des trajets disponible.



Scénario nominal :

- 1) Saisie du point de départ, de point d'arrivée et l'ensemble des points intermédiaire traçant le trajet (son parcours).
- 2) Saisie des autres informations : heure, date, nombre de places disponible.
- 3) Le système vérifie les informations saisies.
- 4) Le conducteur confirme la création du trajet.

Scénario alternatif :

- 3) a. Si un champ est vide ou mal saisi : afficher erreur et demande de ressaisie.
- 4) a. Si le conducteur ne confirme pas la demande, le trajet n'est pas créé.

9. Consulter trajet _Gestion trajet :

Objectif : Trouver le trajet recherché, et consulter ses données.

Acteurs : Passager, Conducteur.

Pré-condition : Etre authentifié.

Post-condition : Avoir la liste des trajets répondants aux critères de recherche, et pouvoir consulter un trajet bien spécifique parmi eux.

Scénario nominal :

- 1) Saisie des critères de recherche : Point de départ, point d'arrivée, date, heure ...
- 2) Vérification du système et filtrage de l'ensemble des données, selon les entrées de la recherche.
- 3) Retourner la liste des trajets compatible avec les critères, classés par taux de proximité.
- 4) Sélectionner un des trajets proposés et pouvoir consulter ses informations et son parcours (carte : l'ensemble des points)

Scénario alternatif :

- 2) a. retourner qu'aucun trajet n'est trouvé si c'est le cas.

10. Modifier trajet _Gestion trajet :

Objectif : Pouvoir modifier le trajet proposé.

Acteurs : Conducteur.

Pré-condition : être authentifié, et avoir des trajets suggérés.

Post-condition : Trajet modifié.

Scénario nominal :

- 1) Sélectionner le trajet à modifier.
- 2) Etablir les modifications désirées sur le trajet : date, heure etc...
- 3) Validation des modifications par le conducteur
- 4) Vérification du système et demande une confirmation
- 5) L'utilisateur confirme, et les modifications sont effectuées et enregistrées.
- 6) Notifier les passagers qui ont réservé sur ce trajet.



Scénario alternatif :

- 3) a. Renvoi d'un message d'erreur si c'est le cas.
- b. si l'utilisateur ne confirme pas les modifications sont annulées.

11. Supprimer trajet _Gestion trajet :

Objectif : Supprimer un trajet proposé.

Acteurs : Conducteur.

Pré-condition : être authentifié, et avoir des trajets suggérés.

Post-condition : le trajet est supprimé.

Scénario nominal :

- 1) Sélectionner le trajet à supprimer.
- 2) Vérification du système et demande une confirmation
- 3) L'utilisateur confirme, et le trajet est supprimé.
- 4) Notifier les passagers qui ont réservé sur ce trajet.

Scénario alternatif :

- 3) a. Si l'utilisateur ne confirme pas la suppression est annulée.

12. Effectuer une réservation _Gestion réservation :

Objectif : Réserver dans le trajet souhaité.

Acteurs : Passager, Conducteur.

Pré-condition : Etre authentifié, avoir consulté le trajet, et le nombre de place disponible répond au besoin de l'acteur.

Post-condition : La réservation est effectuée et le nombre de places de cette dernière est décrémenté.

Scénario nominal :

- 1) Le passager consulte le trajet et demande une réservation et le trajet est ajouté au "panier" des trajets souhaités.
- 2) La demande est envoyée au conducteur proposant le trajet.
- 3) Le conducteur accepte la demande.
- 4) Le passager confirme la demande
- 5) Le nombre de place est décrémenté selon le nombre demandé par le passager.

Scénario alternatif :

- 2) a. le passager contacte grâce au service de messagerie (ou autre) pour demander s'il y a possibilité de passer par un point du trajet au lieu d'un autre.
- b. passer à l'étape 3 du scénario nominal.
- 4) a. Si le conducteur refuse ; la réservation est annulée.
- 5) a. Si le passager ne confirme pas sa demande, la réservation est annulée.



13. Annuler une réservation _Gestion d'une réservation :

Objectif : Annuler une réservation faite.

Acteurs : Passager.

Pré-condition : être authentifié et avoir déjà fait une réservation sur un trajet.

Post-condition : la réservation est annulée et le conducteur déposant du trajet concerné est notifié.

Scénario nominal :

- 1) Le passager entre dans le panier des trajets réservés et sélectionne la réservation à annuler.
- 2) Le système demande confirmation.
- 3) Le passager confirme et la réservation est annulée et le nombre de place est incrémenté du trajet.
- 4) Le conducteur déposant du trajet concerné est notifié.

Scénario alternatif :

- 3) a. le passager annule la demande.

14. Modifier une réservation _Gestion d'une réservation :

Objectif : Pouvoir modifier la réservation.

Acteurs : passager

Pré-condition : être authentifié, et avoir les réservations effectuées.

Post-condition : Réservation modifiée.

Scénario nominal :

- 1) Sélectionner la réservation à modifier.
- 2) Etablir les modifications désirées sur la réservation : Nombre d'enfants ...
- 3) Validation des modifications par le passager
- 4) Vérification du système et demande une confirmation
- 5) Le conducteur confirme, et les modifications sont effectuées et enregistrées.

Scénario alternatif :

- 4) a. Envoyer un message d'erreur dans le cas des champs invalides.
b. si le conducteur ne confirme pas, les modifications sont annulées

15. Evaluer un conducteur :

Objectif : Pouvoir noter les conducteurs et les classer.

Acteurs : Passager.

Pré-condition : être authentifié.

Post-condition : La note et le classement du conducteur est modifié selon l'évaluation effectuée par le passager.

Scénario nominal :

- 1) Le passager sélectionne le conducteur à évaluer.



- 2) Le passager sélectionne le nombre des étoiles à attribuer au conducteur.
- 3) Entrer une justification : Fumeur, retards, non-respect du trajet, vitesse, ...
- 4) Confirmer et enregistrement de l'évaluation.

Scénario alternatif :

- 4) a. Si aucune justification n'est entrée l'évaluation est rejetée.

16. Administrer :

Objectif : Avoir une vue d'ensemble du système, et pouvoir l'administrer.

Acteurs : l'administrateur.

Pré-condition : Etre authentifié.

Scénario nominal :

- 1) Si un conducteur est signalé ou mal évalué l'administrateur doit avertir ce dernier.
- 2) Si après 3 avertissement l'administrateur bloque le conducteur et supprime son compte.
- 3) Etc...

2. DIAGRAMME DE CLASSE :

Dans cette étape, nous allons présenter les règles de gestion et faire la description des classes en déterminant leurs attributs et leurs méthodes. Ce diagramme regroupe les différentes classes dont ils sont reliés avec des relations et des associations.

Règles de gestion :

Règle n°1 : Chaque membre a un identifiant, un nom, un prénom ...

Règle n°2 : Chaque compte a un login et un mot de passe et correspond à un membre.

Règle n°3 : Chaque profil est associé à un compte.

Règle n°4 : Le profil a un identifiant un statut et un grade.

Règle n°5 : Chaque profil peut proposer plusieurs annonces trajets.

Règle n°6 : Le trajet est identifié par un id et a une date/heure un état (disponible, complet...)

Règle n°7 : Chaque profil peut réserver en plusieurs trajets.

Règle n°8 : Chaque trajet est composé de plusieurs point trajet.

Règle n°9 : Chaque voiture a un matricule, nombre de place.

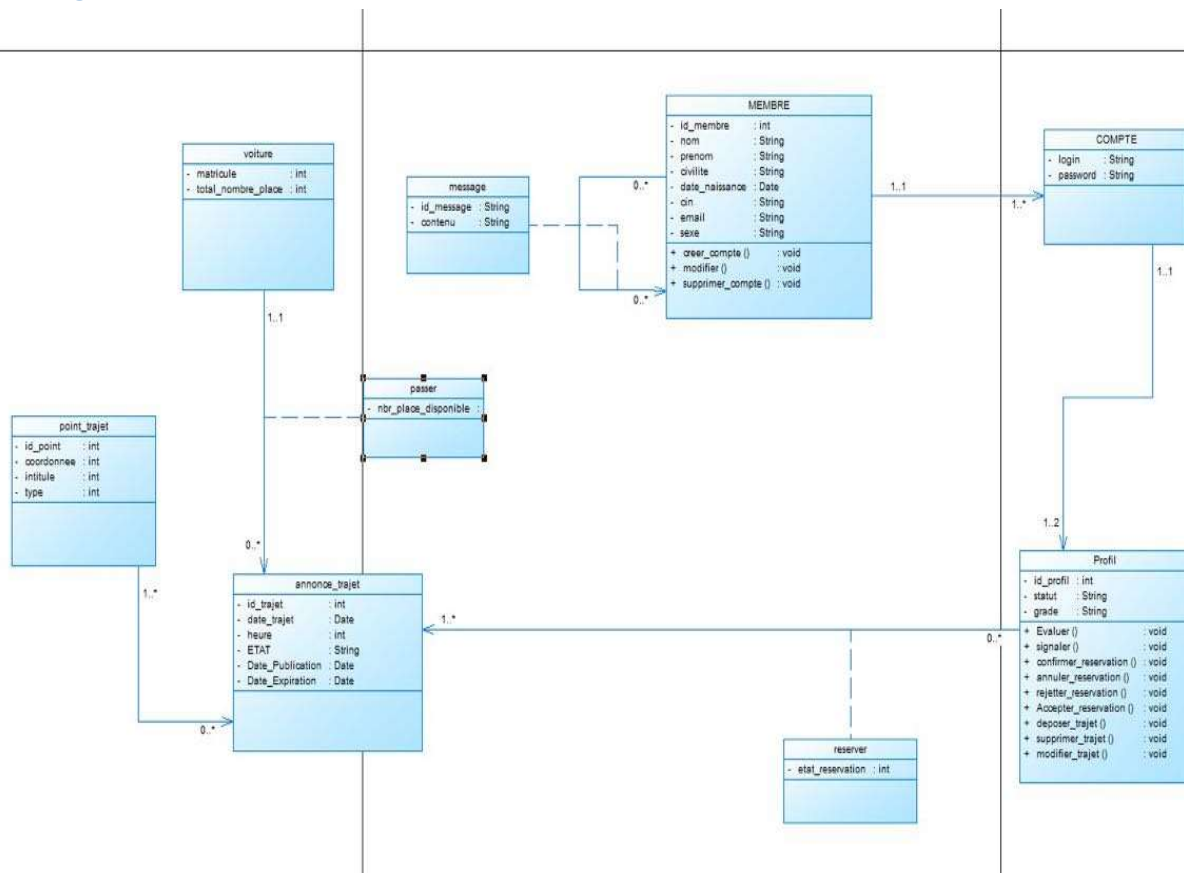
Règle n°10 : Chaque voiture effectue un ou plusieurs trajets.

Règle n°11 : Chaque réservation a un état (confirmé, non confirmé, ...)

Règle n°12 : Un membre peut échanger des messages avec un autres.

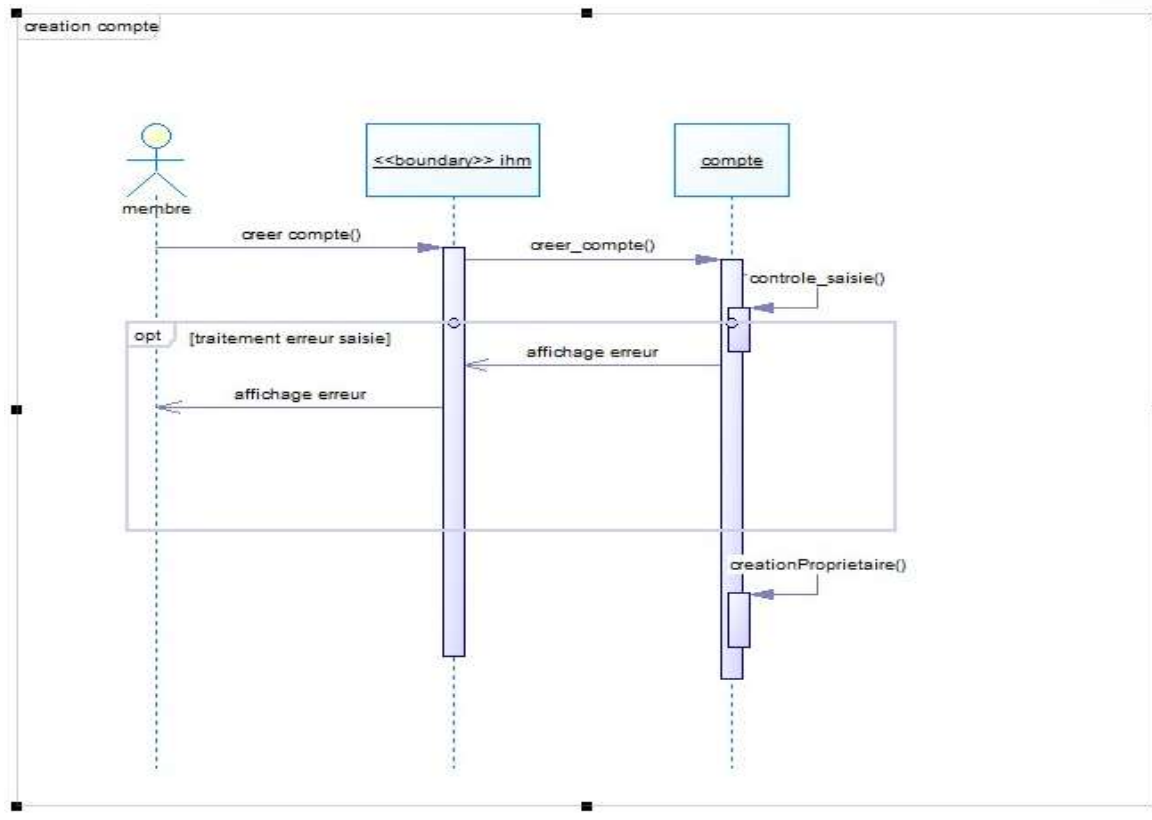


Diagramme :

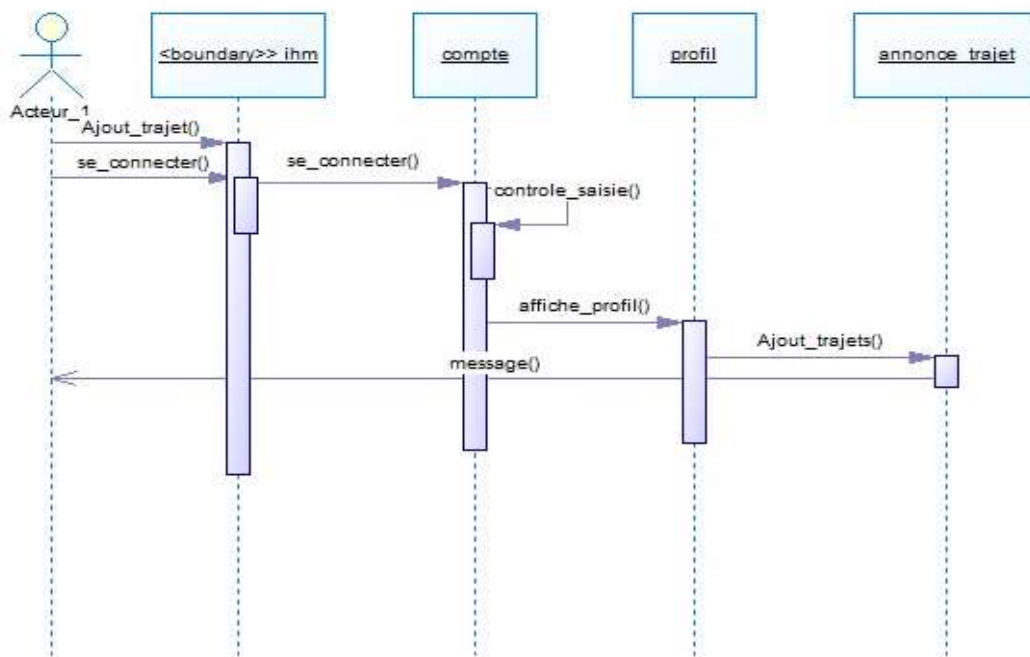


3. DIAGRAMME DE SEQUENCE :

Les diagrammes de séquences suivants montreront une vue dynamique sur l'interaction de l'utilisateur avec le système dans chacun des scénarios possibles à travers notre application.



ajout_trajet

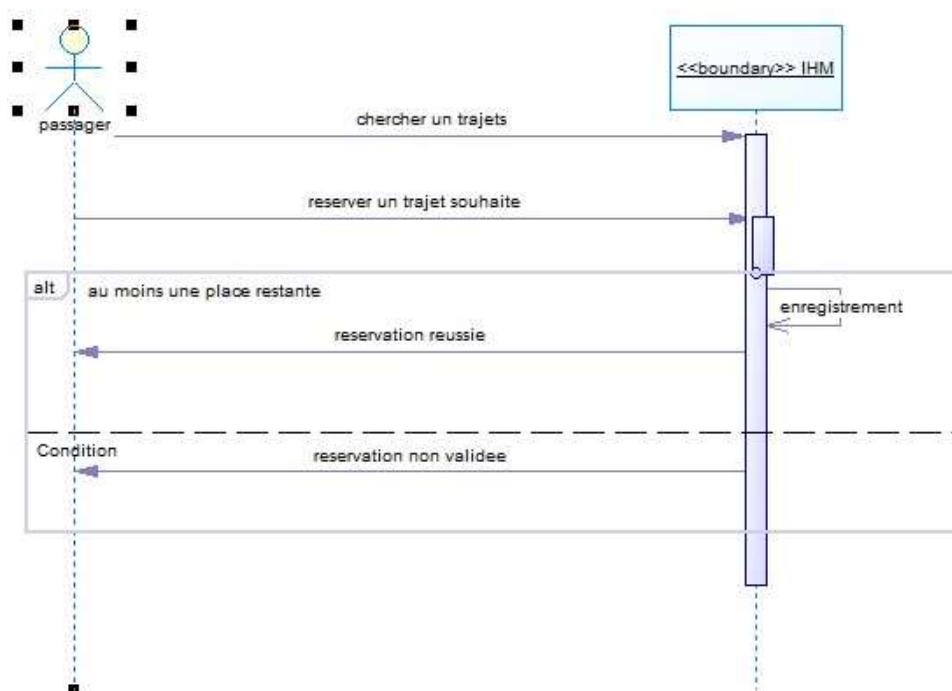




affichage de tous les trajets



reservation

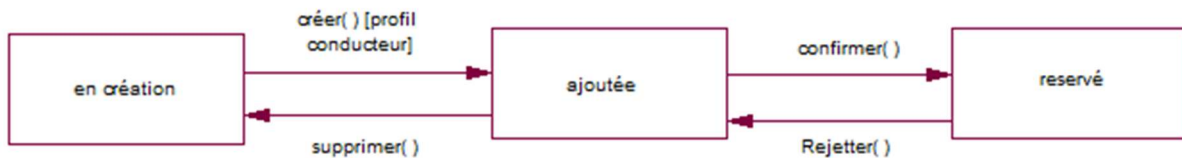




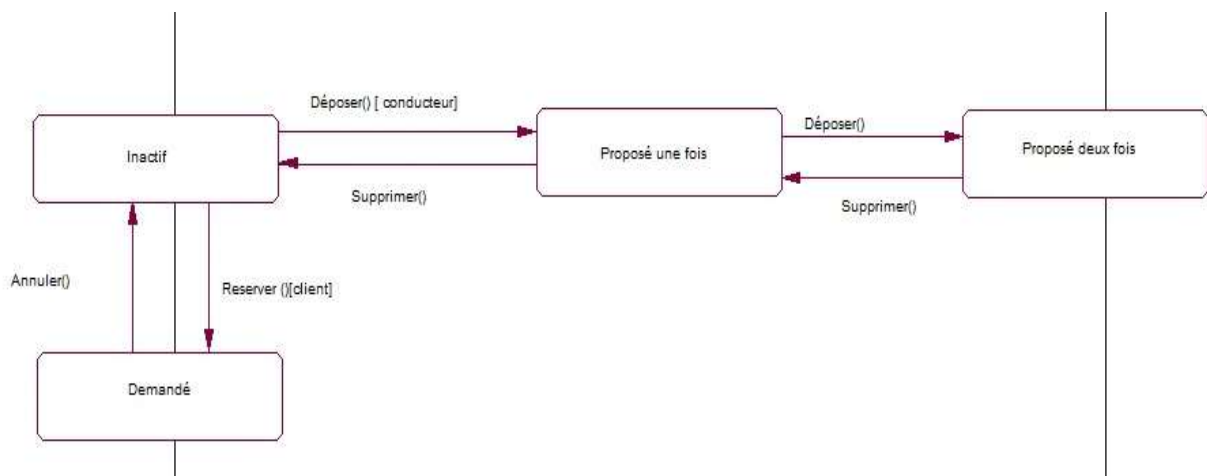
4. DIAGRAMME D'ETAT TRANSITION :

Nous présenterons ci-dessous les diagrammes d'états-transitions d'UML qui décrivent le comportement interne d'un objet à l'aide d'un automate à états finis. Ils présentent les séquences possibles d'états et d'actions qu'une instance de classe peut traiter au cours de son cycle de vie en réaction à des événements discrets.

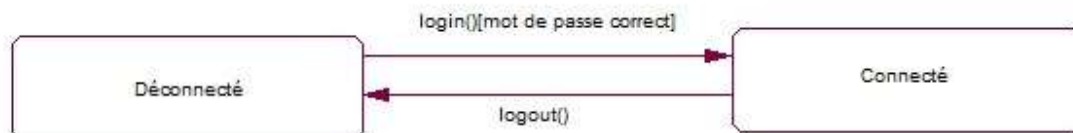
Annonce d'un trajet :



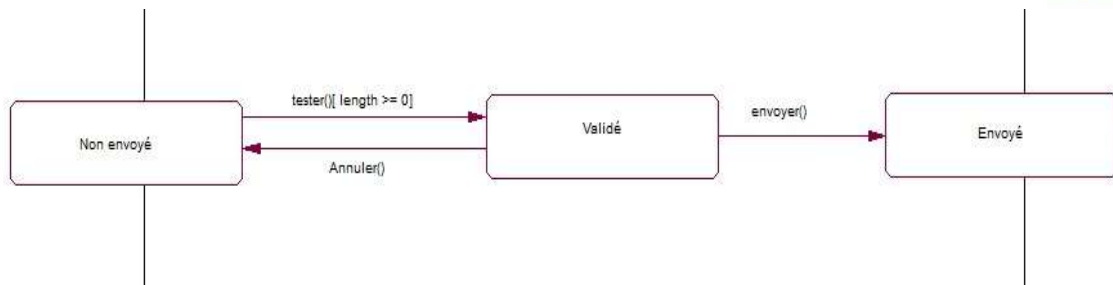
Profil :



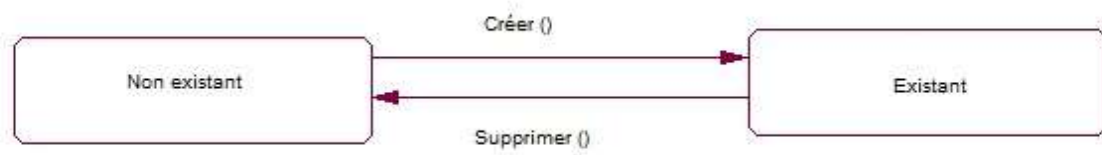
Compte :



Message :



Membre :





CONCLUSION :

Tout au long de ce projet, nous avons été amenés à concevoir une application dédiée au covoiturage scolaire. Ce travail a été l'occasion d'appliquer les connaissances acquises durant notre formation de quatrième année. Il nous a permis de mettre en œuvre les acquis du cours de conception et modélisation UML ainsi de savoir les démarches à suivre dans les futurs projets.