



Programmation avancée : Projet JAVA

Dossier d'analyse

Club de cyclistes

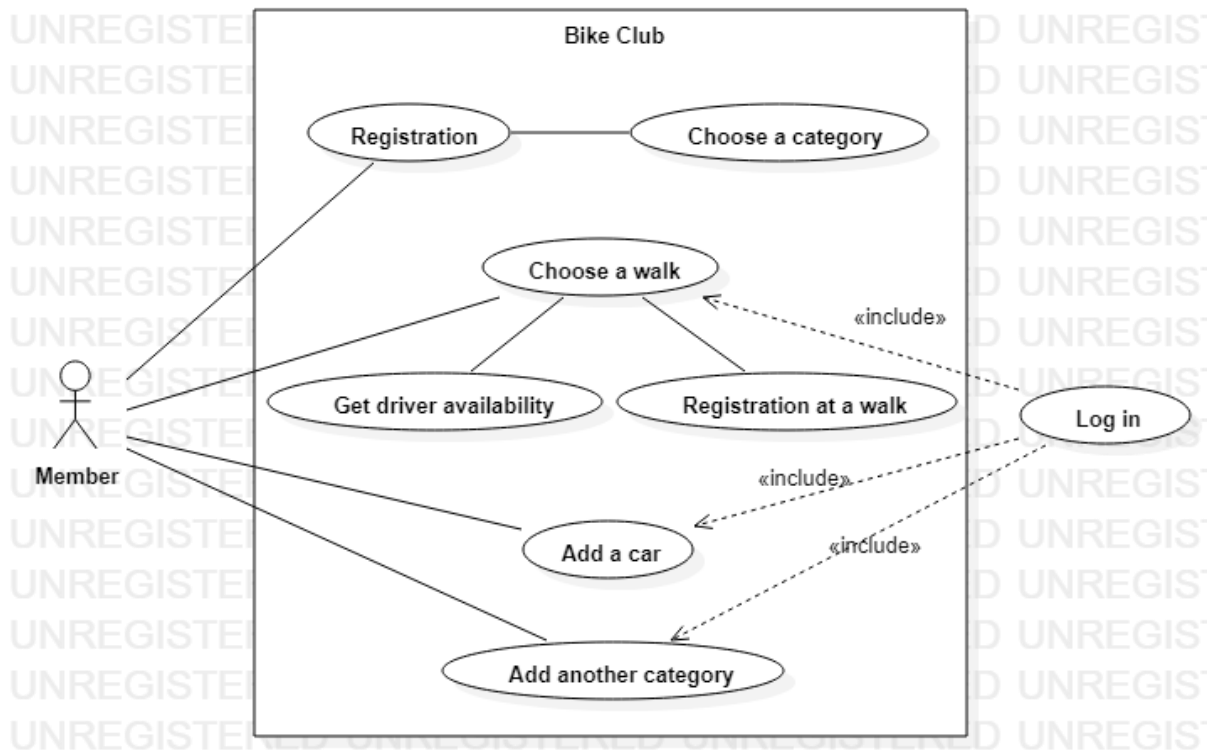
Maître-assistant : L. Masset

Table des matières

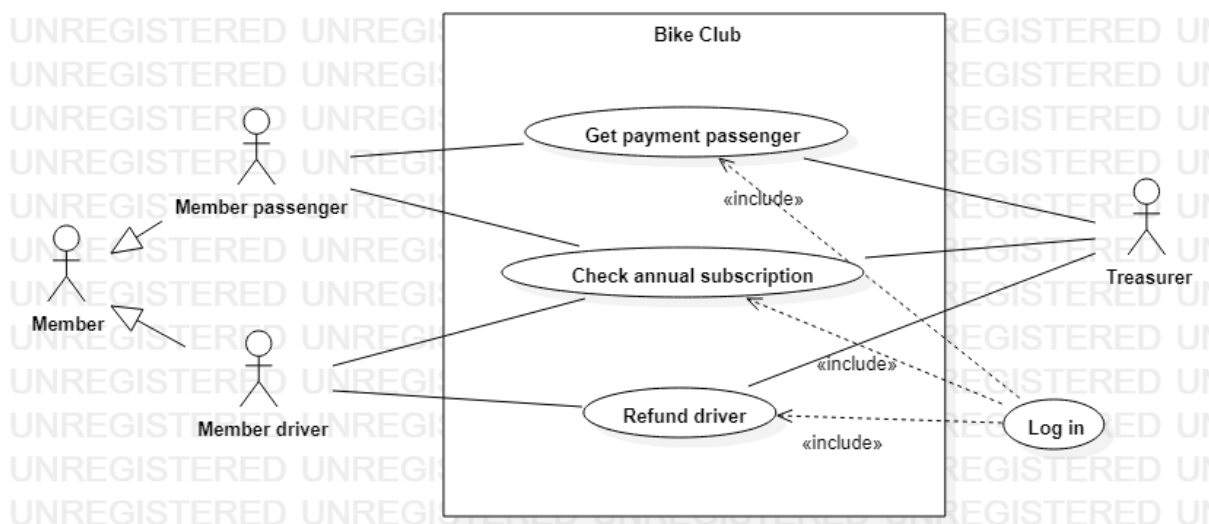
1.	Diagrammes des uses cases	3
1.1	Scénarios	4
2.	Diagramme de classes	14
3.	Diagrammes de séquences.....	15
4.	Explications et accès au programme.....	24

1. Diagrammes des uses cases

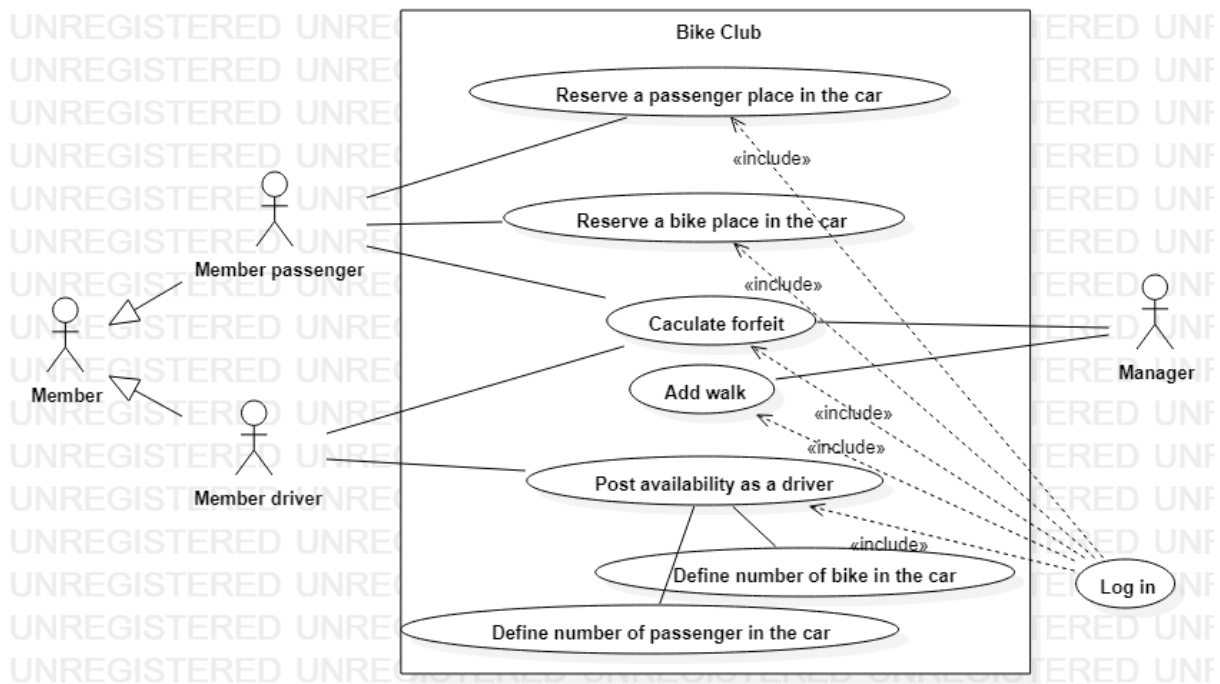
- Voici un diagramme de cas d'utilisation qui représente les actions globales d'un utilisateur :



- Voici un diagramme de cas d'utilisation qui représente les actions que le trésorier peut faire :



- Voici un diagramme de cas d'utilisation qui représente les actions que le manager peut faire :



1.1 Scénarios

▪ « Registration »

Identifiant : UC-01

Acteurs : Membre

Description : S'inscrire à l'application

Préconditions : L'utilisateur ouvre l'application

Scénario nominal :

1. L'utilisateur clique sur le bouton d'inscription.
2. L'utilisateur remplit les champs avec ses données personnelles.
3. L'utilisateur confirme son inscription.

Scénario alternatif :

Cas A : les champs remplis sont erronés ou le compte existe déjà

3. Un message d'erreur par erreur apparaît et demande à l'utilisateur de réécrire ses données.

Postconditions : L'utilisateur accède au choix de sa première catégorie

▪ **« Choose a category »**

Identifiant : UC-02

Acteurs : Membre

Description : Choisir la première catégorie dans laquelle le membre va être inscrit

Préconditions : L'utilisateur a fait son inscription (UC-01).

Scénario nominal :

1. L'utilisateur sélectionne la catégorie dans laquelle il veut s'inscrire en premier
2. L'utilisateur confirme son choix.

Postconditions : L'utilisateur accède à l'application et son compte est créé

▪ **« Log In »**

Identifiant : UC-03

Acteurs : Membre

Description : Se connecter à l'application

Préconditions : L'utilisateur ouvre l'application

Scénario nominal :

1. L'utilisateur clique sur le bouton de la connexion.
2. L'utilisateur rentre son identifiant et son mot de passe.
3. L'utilisateur confirme ses données.
4. L'utilisateur est connecté et accède à l'application.

Scénario alternatif :

Cas A : les champs remplis sont erronés ou l'utilisateur n'existe pas

3. Un message d'erreur apparaît demandant à l'utilisateur de réécrire ses données.

Postconditions : L'utilisateur accède à l'application

▪ **« Choose a walk »**

Identifiant : UC-04

Acteurs : Membre

Description : Choisir une balade dans la liste

Préconditions : L'utilisateur s'est connecté à l'application avec un compte

Scénario nominal :

1. L'utilisateur accède à la liste des balades en se connectant.

2. L'utilisateur choisit une balade dans la liste.

Scénario alternatif :

Cas A : l'utilisateur ne choisit pas de balade dans la liste

3. Un message d'erreur apparaît car aucune balade n'a été choisie.

Postconditions : L'utilisateur se retrouve sur la page qui comporte les détails sur la balade ainsi que 2 choix

Cas d'inclusion : UC-03 (Log in)

▪ **« Get driver availability »**

Identifiant : UC-05

Acteurs : Membre

Description : Obtenir les disponibilités des conducteurs pour une balade

Préconditions : L'utilisateur a choisi une balade dans la liste (UC-04)

Scénario nominal :

1. L'utilisateur appuie sur le bouton qui concerne les disponibilités des conducteurs pour une balade en particulier.
2. L'utilisateur accède à la liste des conducteur et des passagers pour cette balade.

Postconditions : L'utilisateur accède au récapitulatif des conducteurs et passagers pour une balade en particulier

▪ **« Registration at a walk »**

Identifiant : UC-06

Acteurs : Membre

Description : S'enregistrer pour une balade

Préconditions : L'utilisateur a choisi une balade dans la liste (UC-04)

Scénario nominal :

1. L'utilisateur appuie sur le bouton qui concerne l'enregistrement pour une balade en particulier.
2. L'utilisateur accède à l'enregistrement et choisi s'il veut être conducteur ou passager ou ni l'un ni l'autre.
3. L'utilisateur valide les informations renseignées.

Scénario alternatif :

Cas A : l'utilisateur ne complète pas les informations requises

4. Un message d'erreur apparaît car une ou plusieurs informations sont manquantes.

Postconditions : L'utilisateur s'est enregistré dans une balade et il se retrouve sur la page d'accueil

▪ **« Add a car »**

Identifiant : UC-07

Acteurs : Membre

Description : Ajouter une voiture à son compte

Préconditions : L'utilisateur s'est connecté à l'application avec un compte

Scénario nominal :

1. L'utilisateur clique sur son compte.
2. L'utilisateur clique sur le bouton pour rajouter une voiture.
3. L'utilisateur remplit quelques informations concernant la voiture.
4. L'utilisateur valide les informations.

Scénario alternatif :

Cas A : l'utilisateur ne complète pas les informations requises

4. Un message d'erreur apparaît car une ou plusieurs informations sont manquantes.

Cas B : l'utilisateur possède déjà une voiture

3. Un message apparaît indiquant à l'utilisateur qu'il possède déjà une voiture d'enregistrée à son nom.

Postconditions : L'utilisateur a enregistré une voiture, il retourne à la page d'accueil

Cas d'inclusion : UC-03 (Log in)

▪ **« Add another category »**

Identifiant : UC-08

Acteurs : Membre

Description : Ajouter une autre catégorie à son compte

Préconditions : L'utilisateur s'est connecté à l'application avec un compte

Scénario nominal :

1. L'utilisateur clique sur son compte.
2. L'utilisateur clique sur le bouton pour rajouter une nouvelle catégorie.

3. L'utilisateur choisit sa nouvelle catégorie.
4. L'utilisateur confirme son choix.

Scénario alternatif :

Cas A : l'utilisateur ne complète pas les informations requises

4. Un message d'erreur apparaît car une ou plusieurs informations sont manquantes.

Cas B : l'utilisateur possède déjà toutes les catégories

3. Un message apparaît indiquant à l'utilisateur qu'il a toutes les catégories possibles enregistrées à son nom.

Postconditions : L'utilisateur a enregistré une nouvelle catégorie, il retourne à la page d'accueil

Cas d'inclusion : UC-03 (Log in)

▪ **« Get payment passenger »**

Identifiant : UC-09

Acteurs : Membre passager, Trésorier

Description : Recevoir le paiement du membre qui est passager dans une voiture

Préconditions : L'utilisateur s'est connecté à l'application avec un compte

Scénario nominal :

1. L'utilisateur trésorier clique sur le bouton qui l'amène au récapitulatif des paiements.
2. L'utilisateur trésorier clique sur une balade en particulier.
3. L'utilisateur trésorier choisit un passager.
4. L'utilisateur trésorier envoie un message au passager qui n'as pas encore payé sa place.
5. L'utilisateur passager reçoit le message.
6. L'utilisateur passager paie auprès du trésorier.
7. L'utilisateur trésorier voit le paiement du passager.
8. L'utilisateur trésorier valide son paiement.

Scénario alternatif :

Cas A : l'utilisateur trésorier ne choisit pas de balade en particulier

3. Un message d'erreur apparaît indiquant de choisir une balade.

Cas B : l'utilisateur trésorier ne choisit pas de passager

4. Un message d'erreur apparaît indiquant de choisir un passager.

Cas C : l'utilisateur passager ne reçoit pas le message

5. Après 1 jour, un nouveau message est envoyé à l'utilisateur passager ou conducteur.

Postconditions : L'utilisateur passager est en règle de paiement et l'utilisateur trésorier retourne à l'écran du récapitulatif des paiements

Cas d'inclusion : UC-03 (Log in)

▪ **« Check annual subscription »**

Identifiant : UC-10

Acteurs : Membre passager, Membre conducteur, Trésorier

Description : Vérification du paiement des cotisations annuelle

Préconditions : L'utilisateur s'est connecté à l'application avec un compte

Scénario nominal :

1. L'utilisateur trésorier clique sur le bouton qui l'amène au récapitulatif des paiements.
2. L'utilisateur trésorier clique sur les cotisations annuelles.
3. L'utilisateur trésorier choisit un membre.
4. L'utilisateur trésorier envoie un message au membre (passager ou conducteur) qui n'as pas payé sa cotisation annuelle.
5. L'utilisateur passager ou conducteur reçoit le message de rappel de paiement.
6. L'utilisateur passager ou conducteur paie sa cotisation annuelle.
7. L'utilisateur trésorier reçoit le paiement de l'utilisateur passager ou conducteur.
8. L'utilisateur trésorier valide le paiement de l'utilisateur passager ou conducteur.

Scénario alternatif :

Cas A : l'utilisateur trésorier ne choisit pas de membre en particulier

4. Un message d'erreur apparait indiquant de choisir une balade.

Cas B : l'utilisateur passager ne reçoit pas le message

5. Après 1 jour, un nouveau message est envoyé à l'utilisateur passager.

Postconditions : L'utilisateur passager ou conducteur est en règle de paiement de ses cotisations annuelles et l'utilisateur trésorier retourne à l'écran des récapitulatifs des paiements

Cas d'inclusion : UC-03 (Log in)

▪ **« Refund driver »**

Identifiant : UC-11

Acteurs : Membre conducteur, Trésorier

Description : Rembourser les conducteurs

Préconditions : L'utilisateur s'est connecté à l'application avec un compte

Scénario nominal :

1. L'utilisateur trésorier clique sur le bouton qui l'amène au récapitulatif des paiements.
2. L'utilisateur trésorier clique sur une balade en particulier.
3. L'utilisateur trésorier choisit un conducteur.
4. L'utilisateur trésorier paie le conducteur.
5. L'utilisateur conducteur reçoit le paiement du trésorier.
6. L'utilisateur trésorier valide le paiement au conducteur.

Scénario alternatif :**Cas A : l'utilisateur trésorier ne choisit pas de balade en particulier**

3. Un message d'erreur apparaît indiquant de choisir une balade.

Cas B : l'utilisateur trésorier ne choisit pas de conducteur

4. Un message d'erreur apparaît indiquant de choisir un conducteur.

Postconditions : L'utilisateur conducteur reçoit son paiement et l'utilisateur trésorier retourne à l'écran des récapitulatifs des paiements

Cas d'inclusion : UC-03 (Log in)

▪ **« Reserve a passenger place in the car »**

Identifiant : UC-12

Acteurs : Membre passager

Description : Réserver une place en tant que passager dans une voiture

Préconditions : L'utilisateur s'est connecté à l'application avec un compte et a cliqué sur le bouton d'enregistrement à une balade

Scénario nominal :

1. L'utilisateur passager indique qu'il sera passager pour cette balade.
2. L'utilisateur passager confirme son choix

Postconditions : L'utilisateur devient passager pour une balade en particulier, il retourne à l'écran d'accueil

Cas d'inclusion : UC-03 (Log in)

▪ **« Reserve a bike place in the car »**

Identifiant : UC-13

Acteurs : Membre passager

Description : Réserver une place pour son vélo dans une voiture

Préconditions : L'utilisateur s'est connecté à l'application avec un compte et a cliqué sur le bouton d'enregistrement à une balade

Scénario nominal :

1. L'utilisateur passager indique qu'il a besoin d'une place dans une voiture pour son vélo pour cette balade.
2. L'utilisateur passager confirme son choix

Postconditions : L'utilisateur a réservé une place dans une voiture pour son vélo pour une balade en particulier, il retourne à l'écran d'accueil

Cas d'inclusion : UC-03 (Log in)

▪ **« Calculate forfait »**

Identifiant : UC-14

Acteurs : Membre passager, Membre conducteur, Responsable

Description : Calculer le forfait pour chaque membre

Préconditions : L'utilisateur s'est connecté à l'application avec un compte

Scénario nominal :

1. L'utilisateur responsable clique sur le bouton des forfaits.
2. L'utilisateur responsable calcule le forfait de chaque membre.
3. L'utilisateur responsable renseigne le forfait de chaque membre.
4. L'utilisateur responsable valide les forfaits.

Scénario alternatif :

Cas A : l'utilisateur responsable ne renseigne pas de forfait

2. Un message d'erreur apparaît indiquant qu'il faut renseigner le forfait du membre.

Postconditions : L'utilisateur responsable retourne sur l'écran d'accueil

Cas d'inclusion : UC-03 (Log in)

▪ **« Add walk »**

Identifiant : UC-15

Acteurs : Responsable

Description : Ajouter une balade à la liste

Préconditions : L'utilisateur s'est connecté à l'application avec un compte

Scénario nominal :

1. L'utilisateur responsable clique sur le bouton d'ajout de balade.

2. L'utilisateur responsable remplit les champs avec les données pour la balade.
3. L'utilisateur responsable valide les données.

Scénario alternatif :

Cas A : les champs remplis sont erronés

4. Un message d'erreur apparaît indiquant que les champs sont vides ou incorrects.

Postconditions : L'utilisateur responsable retourne sur l'écran d'accueil

Cas d'inclusion : UC-03 (Log in)

▪ **« Post availability as driver »**

Identifiant : UC-16

Acteurs : Membre conducteur

Description : Indiquer ses disponibilités en tant que conducteur

Préconditions : L'utilisateur s'est connecté à l'application avec un compte, a ajouté une voiture et s'est enregistré dans une balade

Scénario nominal :

1. L'utilisateur conducteur clique sur le bouton de ses balades.
2. L'utilisateur conducteur choisit une balade en particulier.
3. L'utilisateur conducteur indique qu'il sera conducteur pour cette balade.
4. L'utilisateur conducteur accède à la définition des places disponibles dans sa voiture.

Scénario alternatif :

Cas A : l'utilisateur conducteur ne possède pas encore de balade

2. Un message d'erreur apparaît indiquant qu'il faut qu'il s'enregistre dans au moins une balade.

Cas B : l'utilisateur conducteur ne choisit pas de balade en particulier

3. Un message d'erreur apparaît indiquant de choisir une balade.

Postconditions : L'utilisateur conducteur accède à la définition des places pour vélos disponible dans sa voiture

Cas d'inclusion : UC-03 (Log in)

▪ **« Define number of bikes in the car »**

Identifiant : UC-17

Acteurs : Membre conducteur

Description : Définir le nombre de place de vélo disponible dans une voiture

Préconditions : L'utilisateur a inscrit ses disponibilités comme conducteur (UC-17)

Scénario nominal :

1. L'utilisateur conducteur a indiqué ses disponibilités en tant que conducteur dans une balade en particulier.
2. L'utilisateur conducteur définit le nombre de vélo qu'il peut prendre dans sa voiture.
3. L'utilisateur conducteur valide ce nombre.

Scénario alternatif :

Cas A : aucun nombre n'a été rentré

3. Un message d'erreur apparaît indiquant que l'on doit obligatoirement remplir le champ.

Postconditions : L'utilisateur conducteur accède à la définition des places pour passagers disponibles dans sa voiture

▪ **« Define number of passengers in the car »**

Identifiant : UC-18

Acteurs : Membre conducteur

Description : Définir le nombre de place pour passagers disponible dans une voiture

Préconditions : L'utilisateur a inscrit ses disponibilités comme conducteur (UC-17)

Scénario nominal :

1. L'utilisateur conducteur a indiqué ses disponibilités en tant que conducteur dans une balade en particulier.
2. L'utilisateur conducteur définit le nombre de place pour passagers dans sa voiture.
3. L'utilisateur conducteur valide ce nombre.

Scénario alternatif :

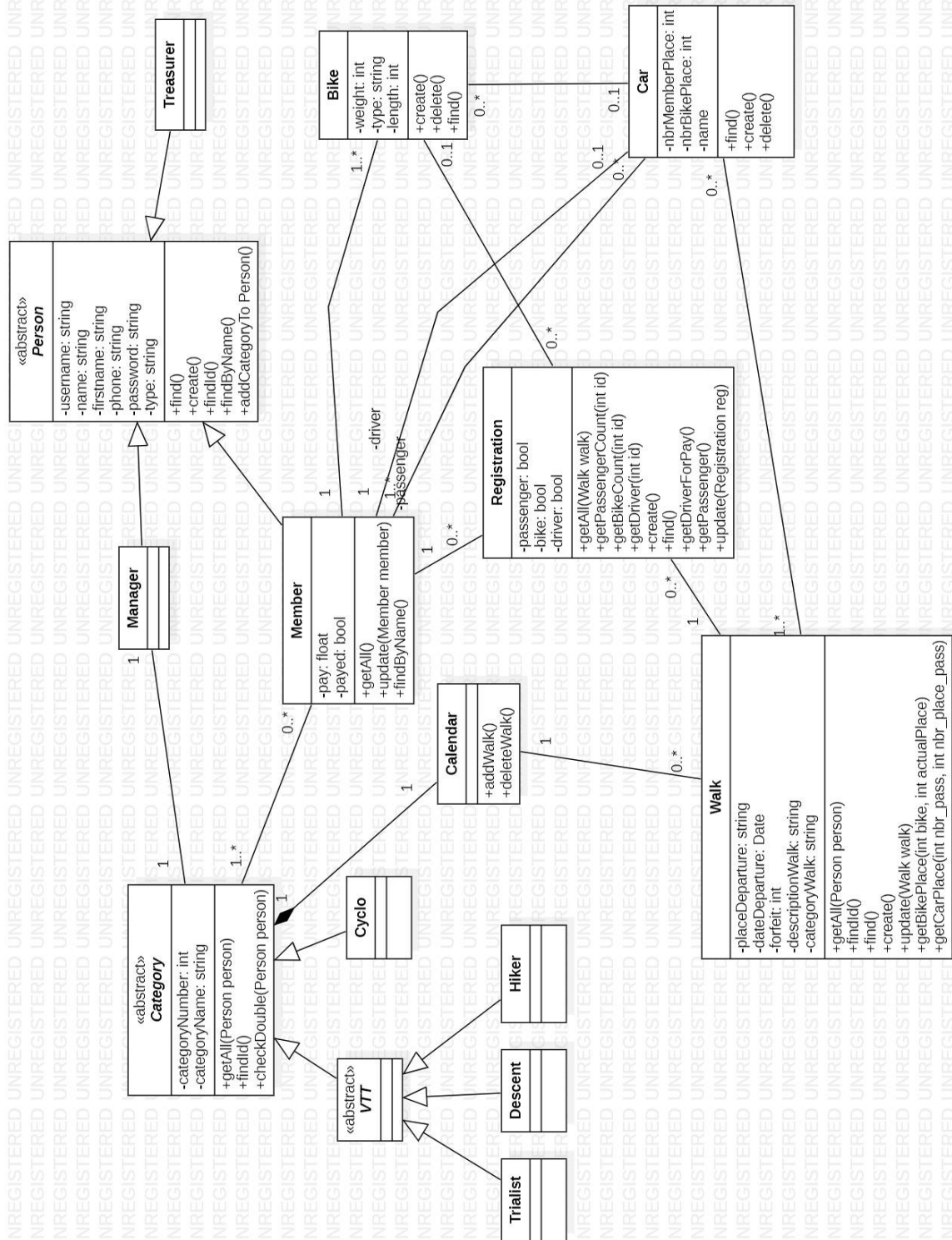
Cas A : aucun nombre n'a été rentré

4. Un message d'erreur apparaît indiquant que l'on doit obligatoirement remplir le champ.

Postconditions : L'utilisateur conducteur retourne à l'écran d'accueil

2. Diagramme de classes

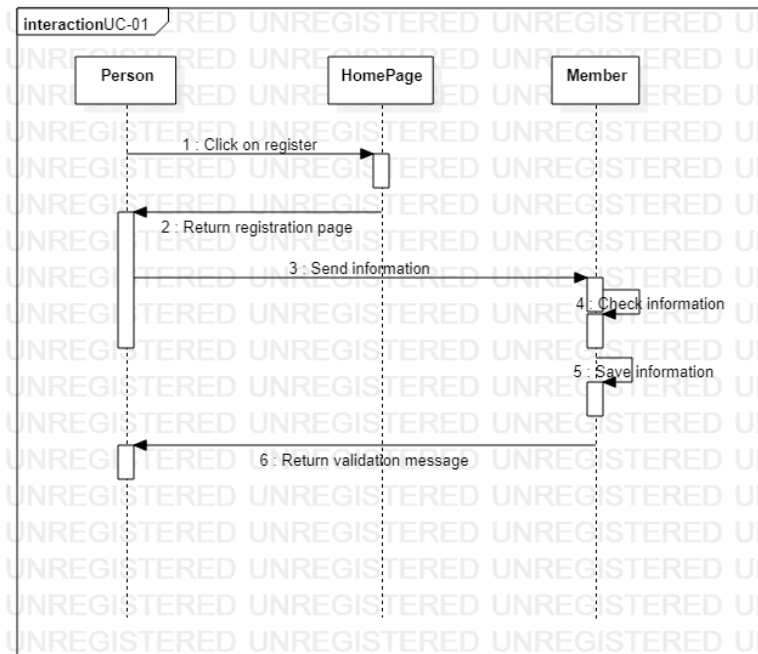
Voici le diagramme de classe revu afin de correspondre à mon code :



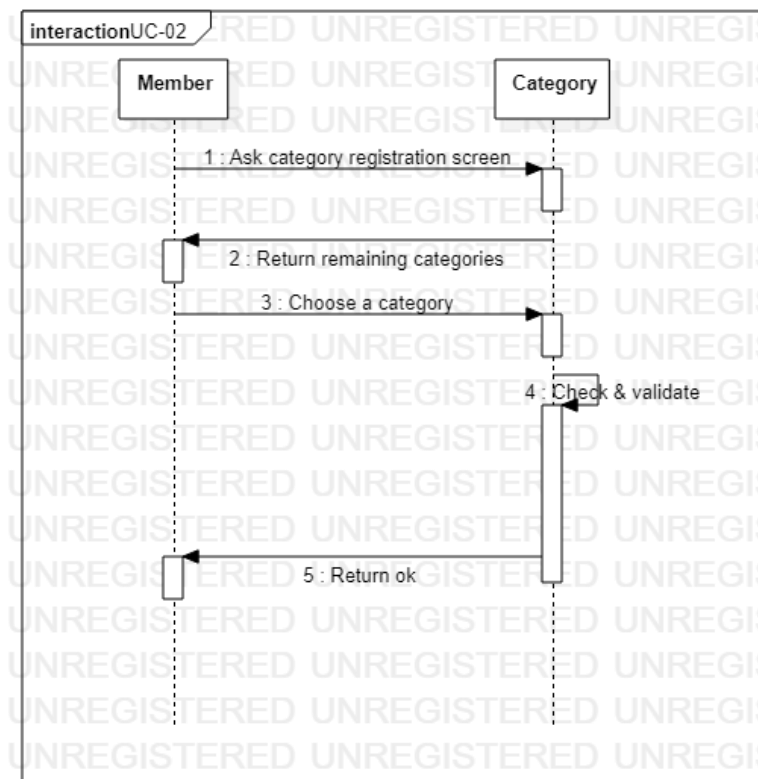
3. Diagrammes de séquences

Voici les diagrammes de séquence. Chaque diagramme correspond à un scénario nominal de chaque scénario.

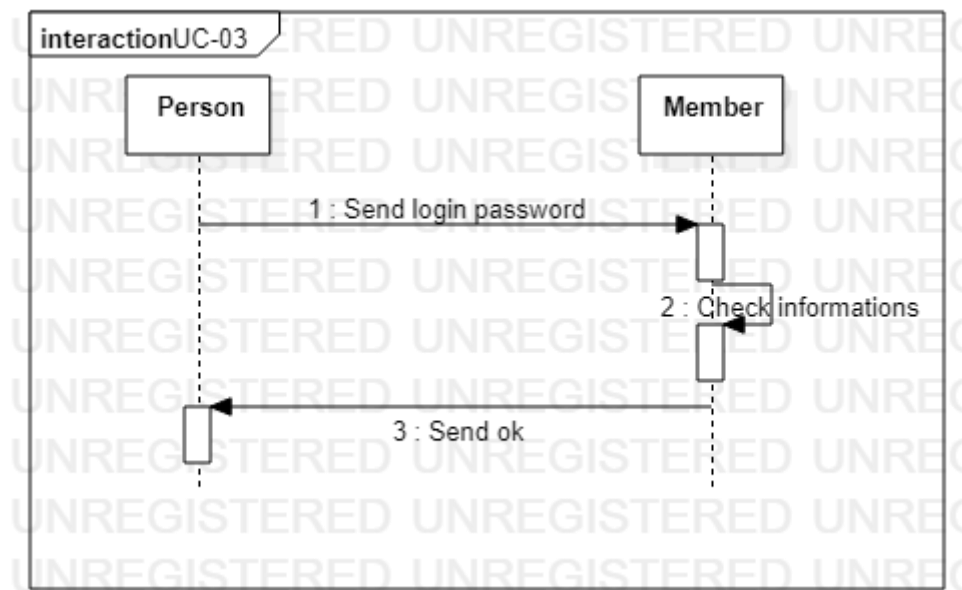
- **UC-01**



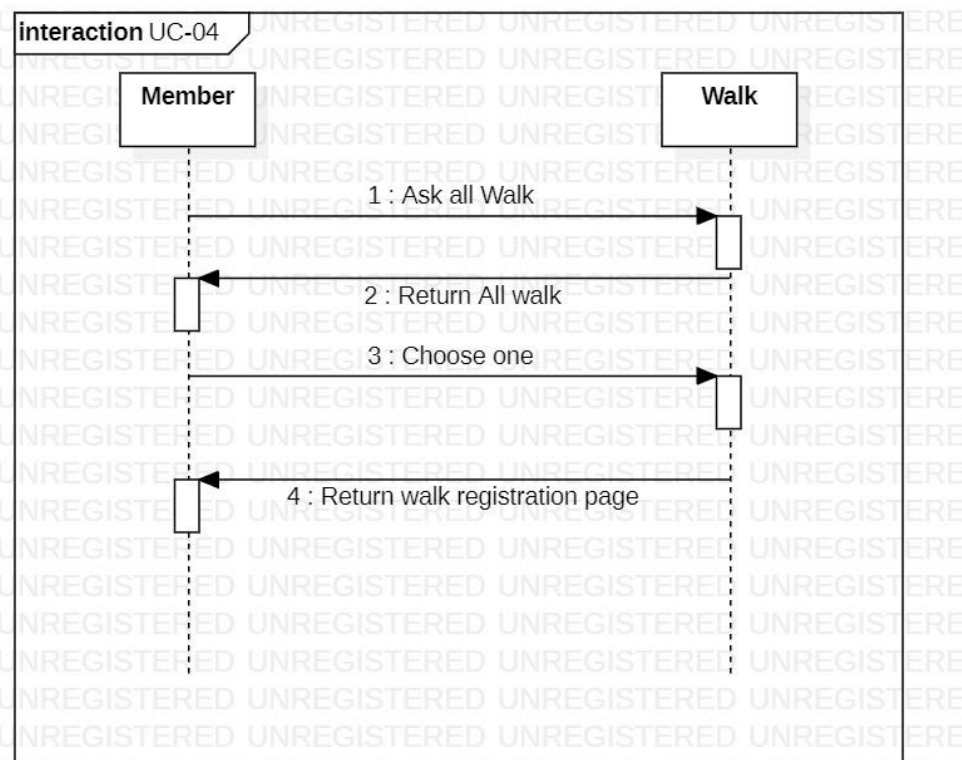
- **UC-02**



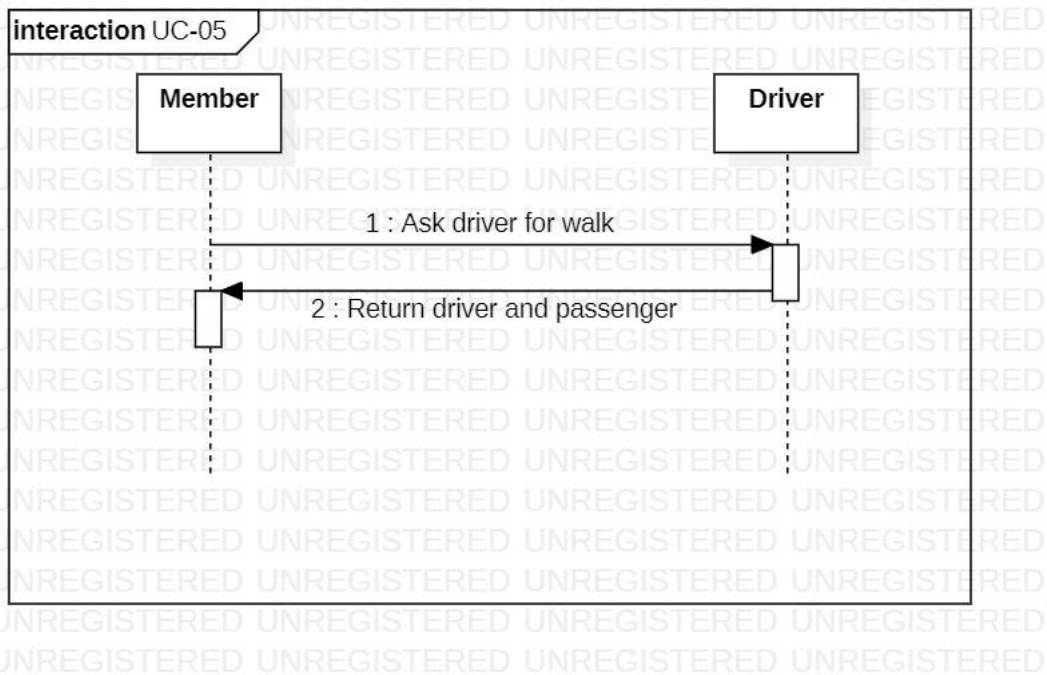
- **UC-03**



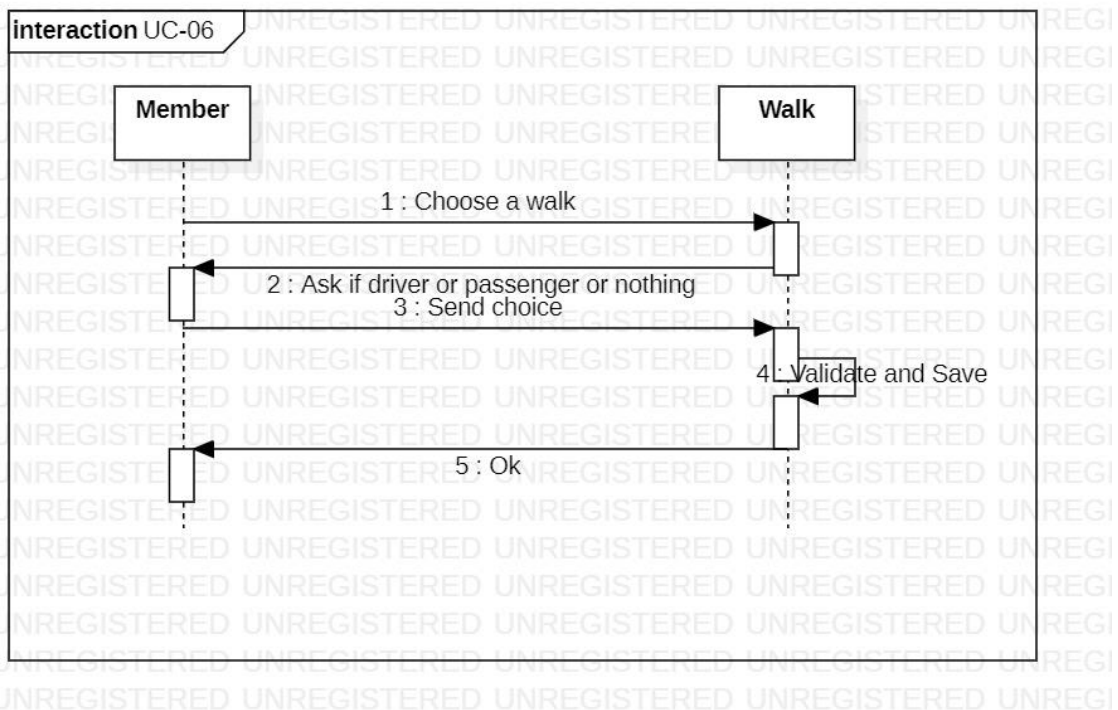
- **UC-04**



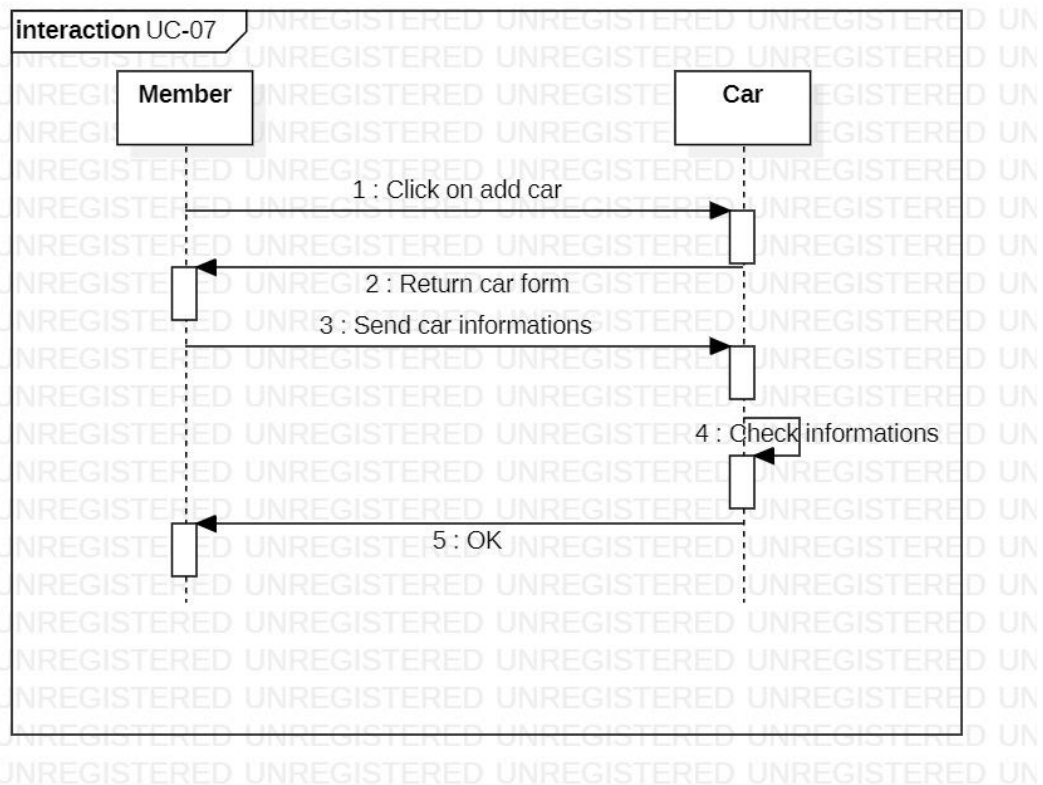
▪ UC-05



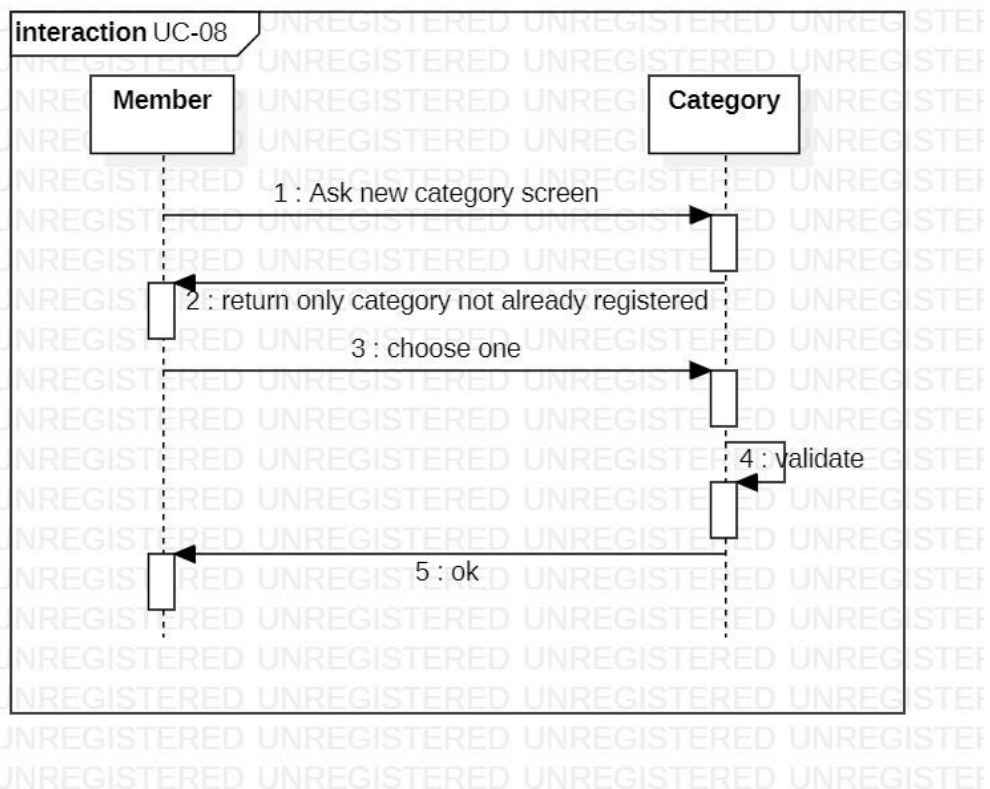
▪ UC-06



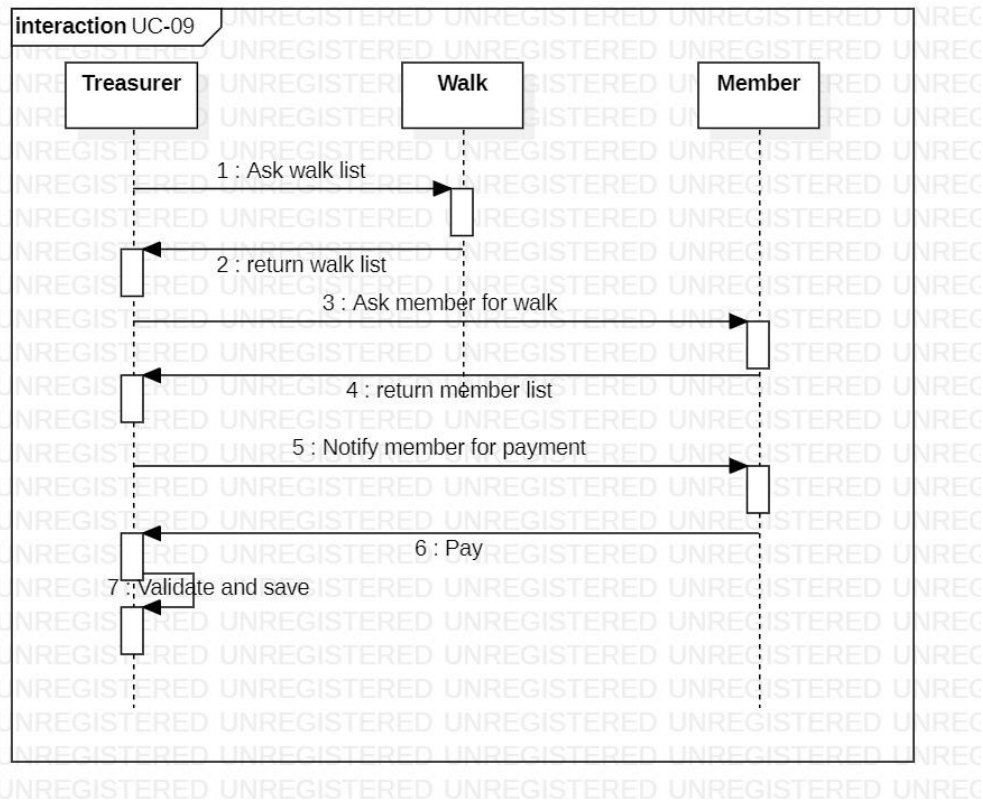
- **UC-07**



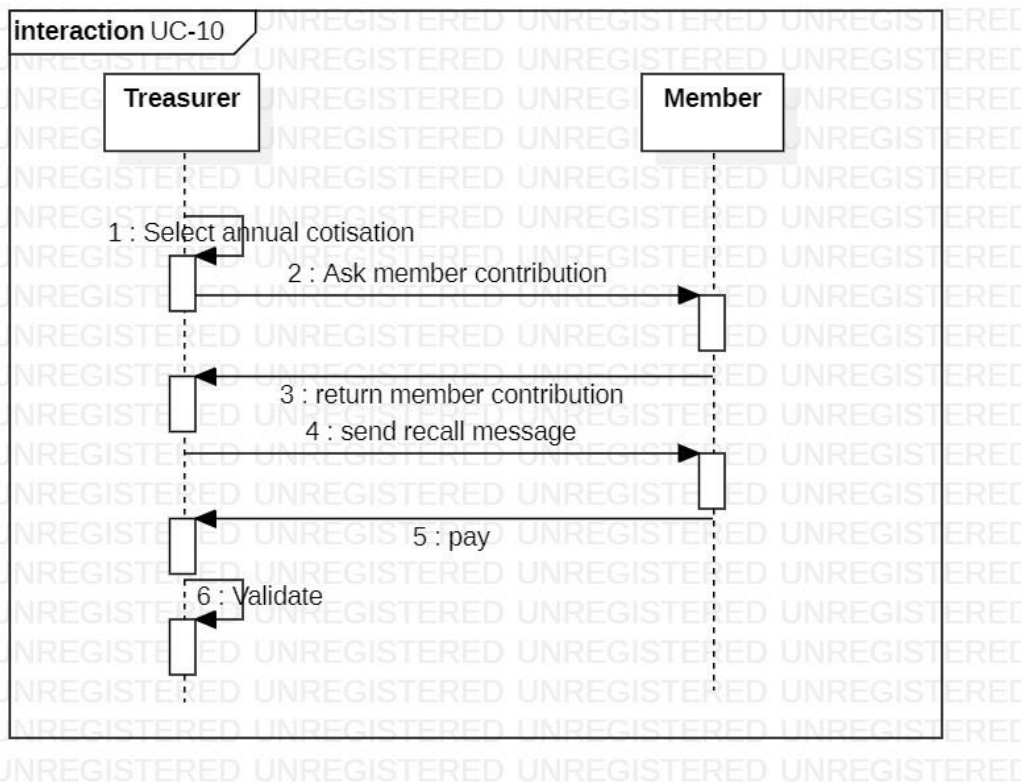
- **UC-08**



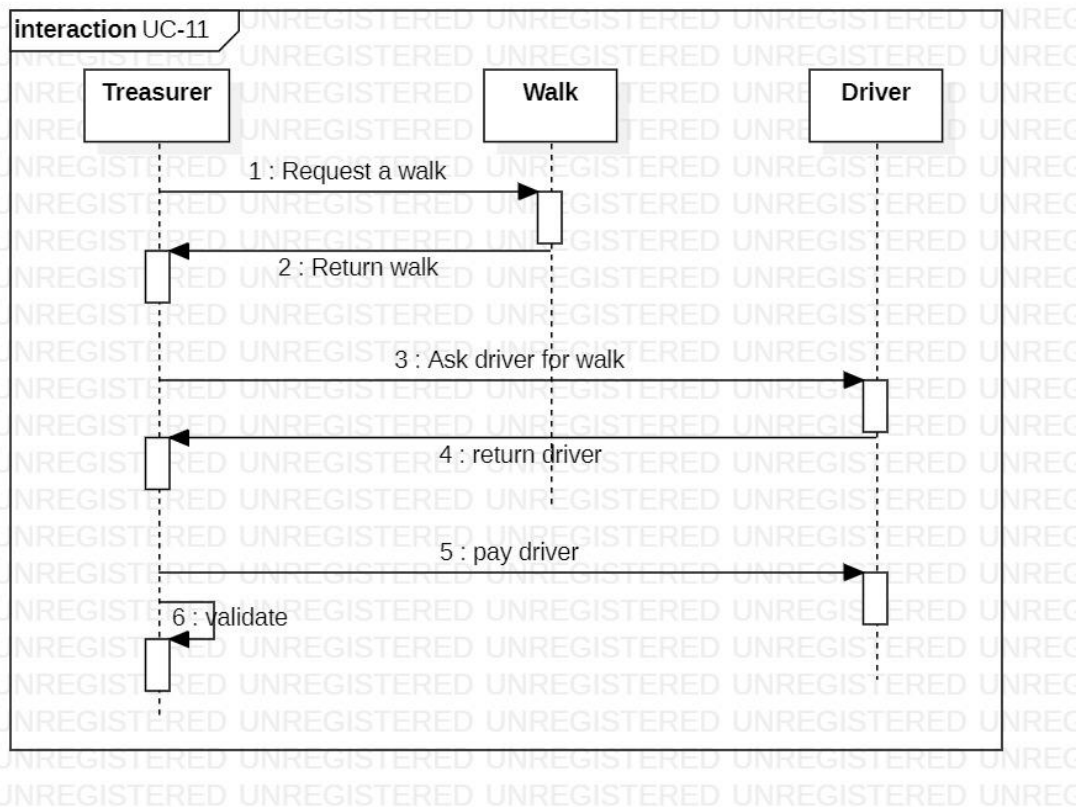
▪ UC-09



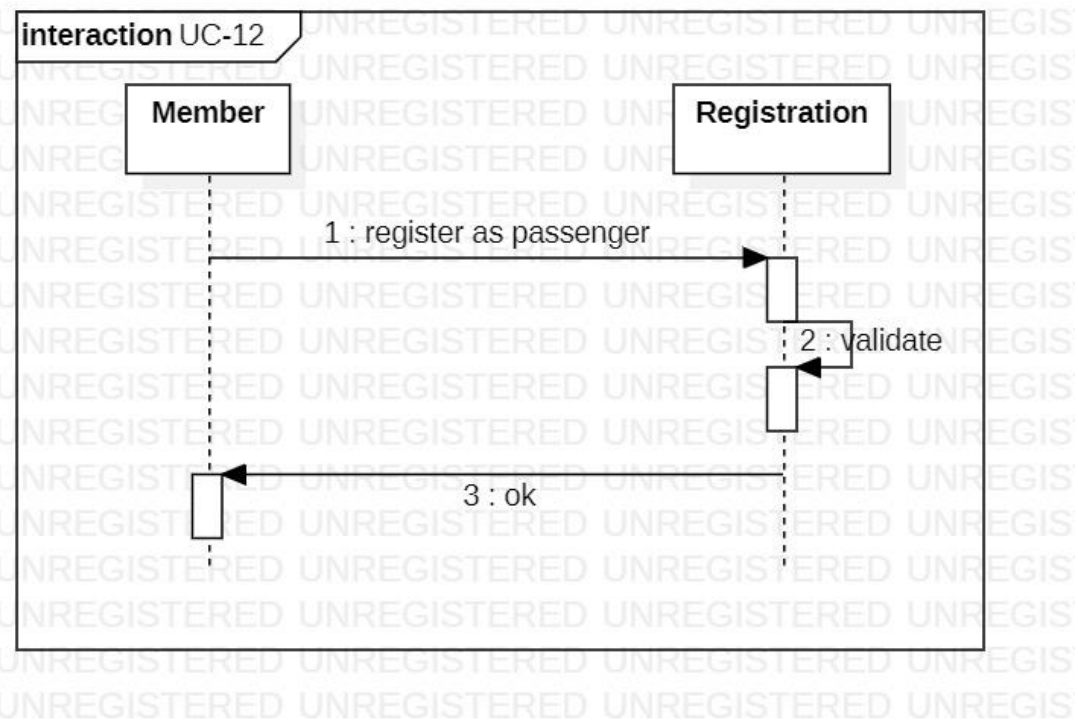
▪ UC-10



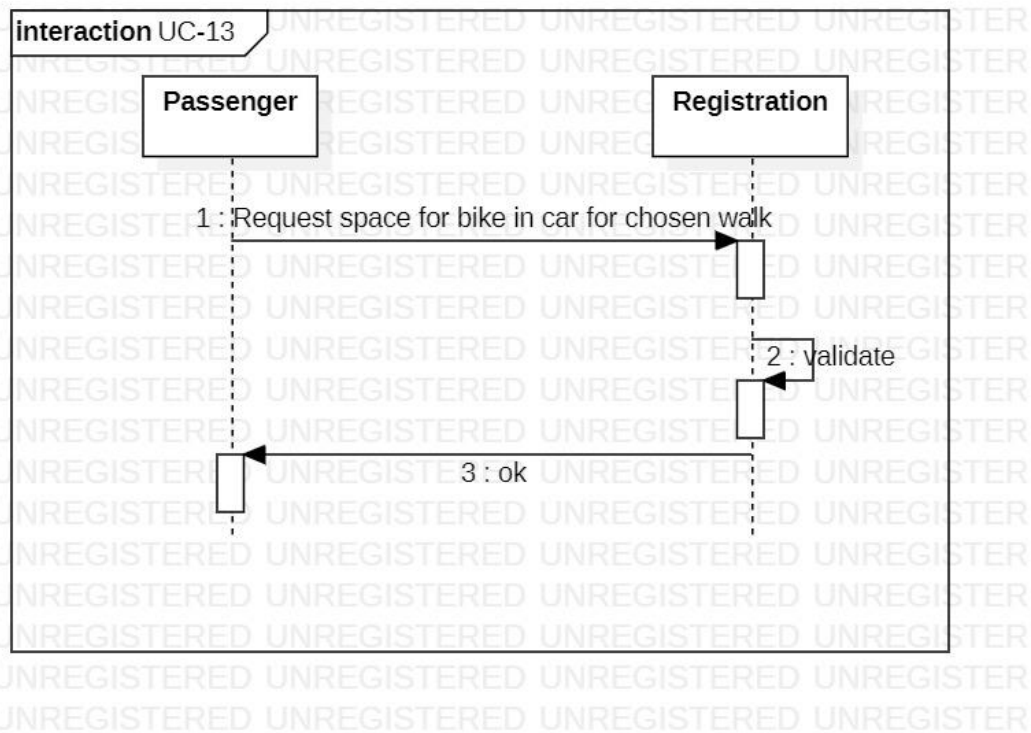
- **UC-11**



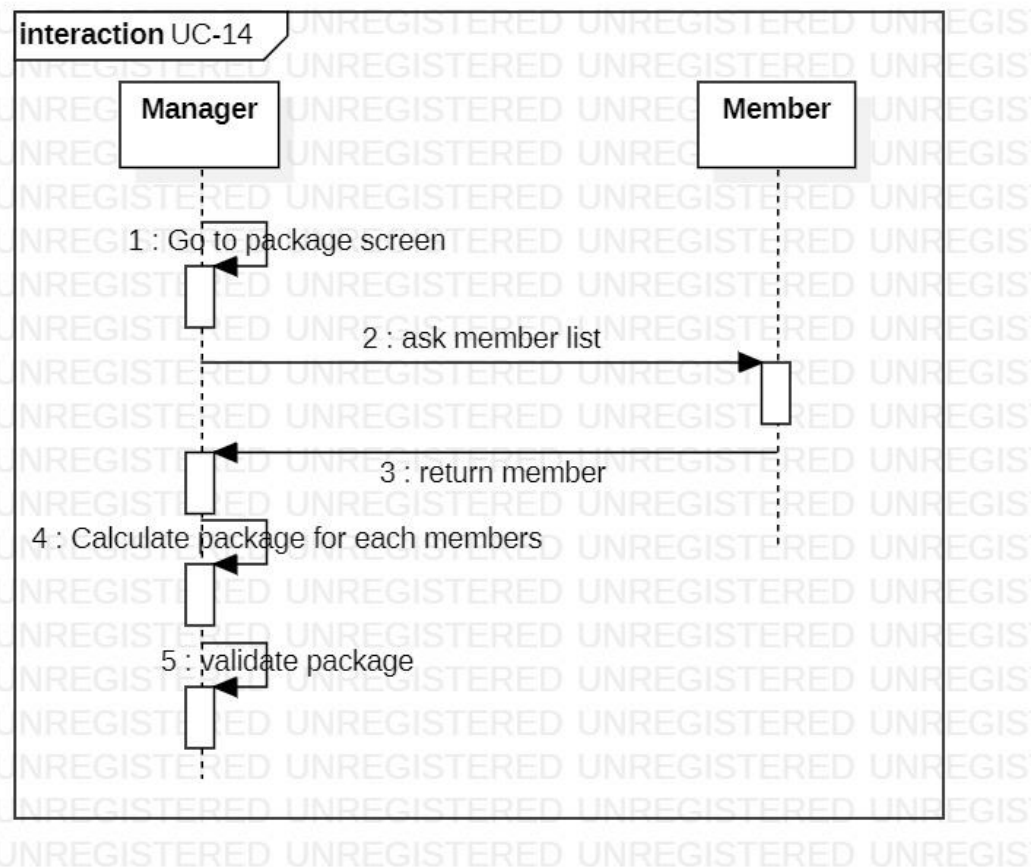
- **UC-12**



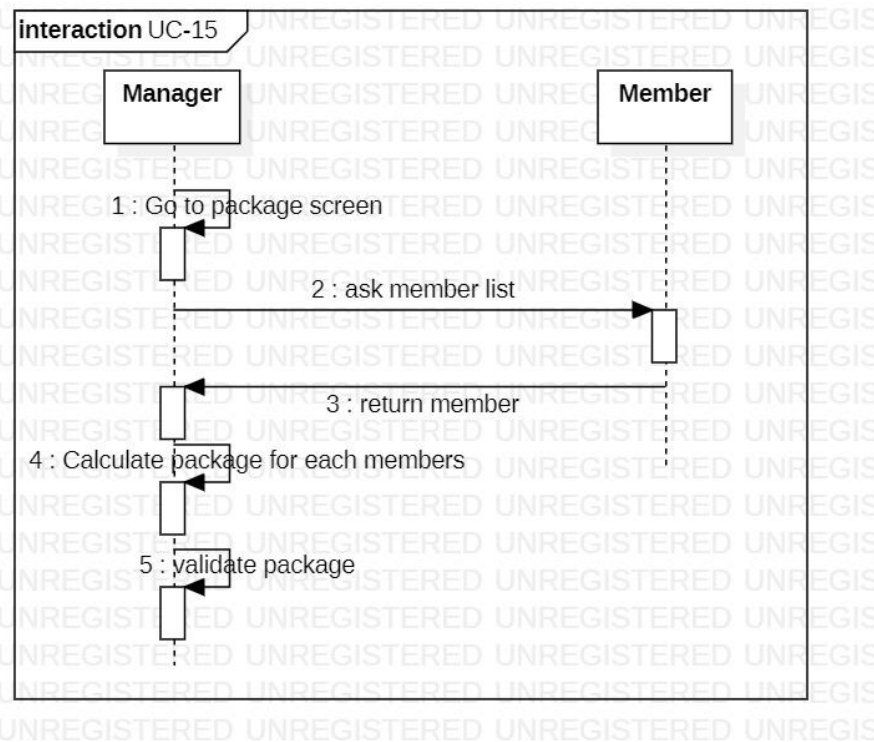
- **UC-13**



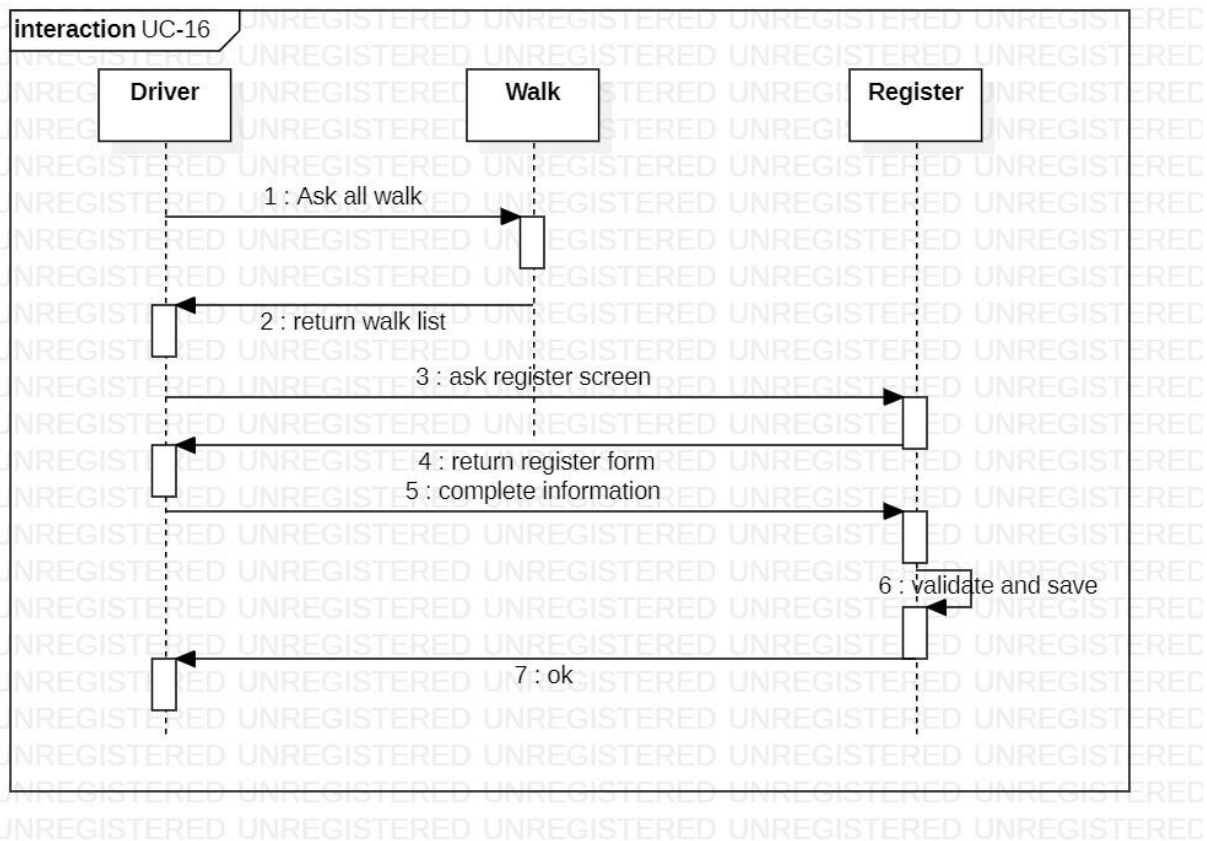
- **UC-14**



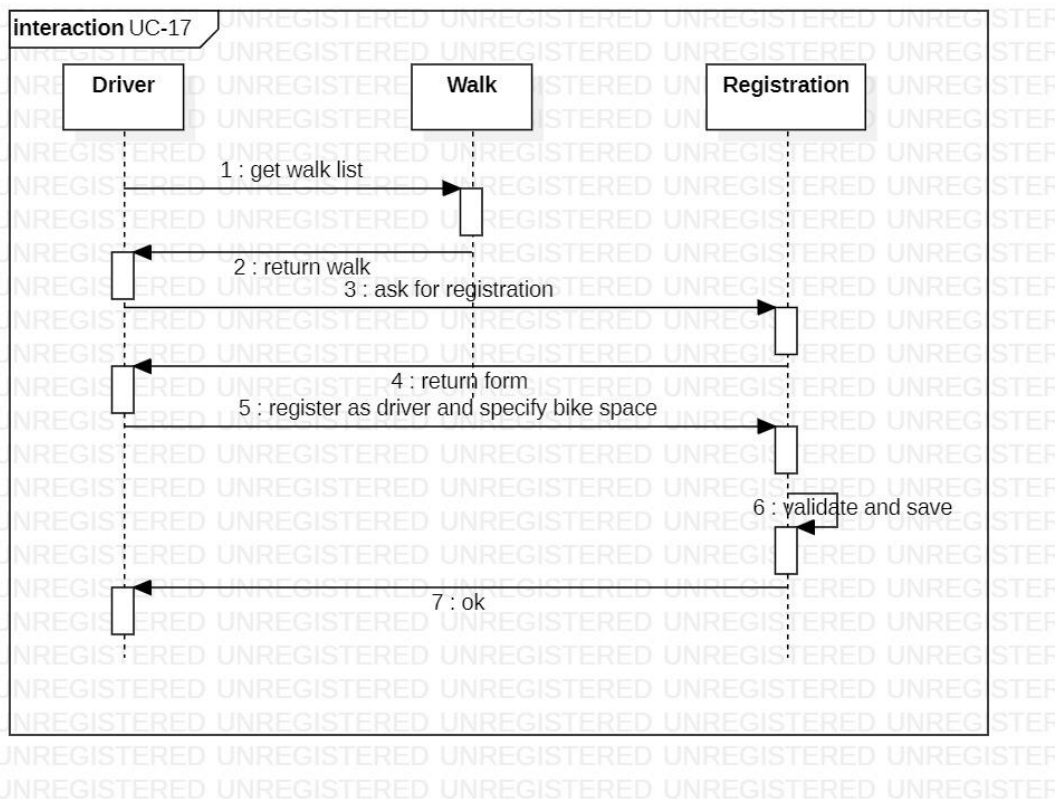
▪ UC-15



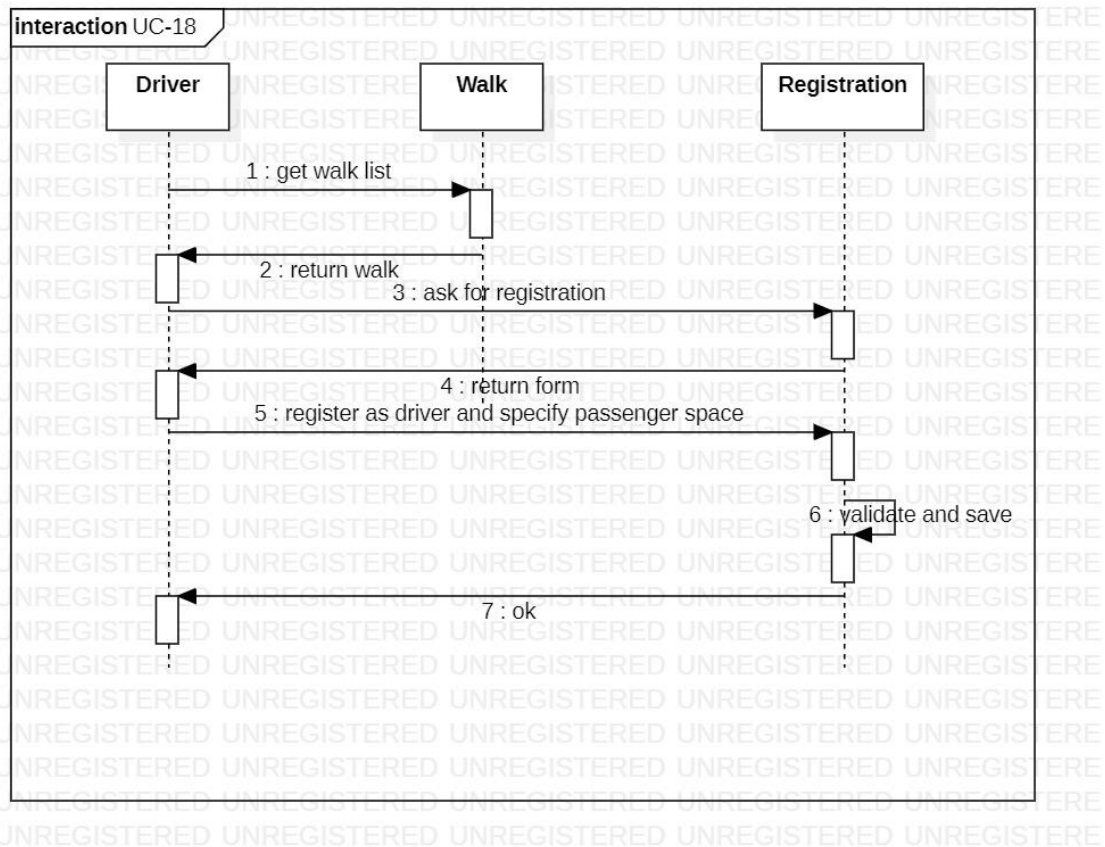
▪ UC-16



■ UC-17



■ UC-18



4. Explications et accès au programme

Explications de quelques fonctions

- Dans le POJO Calendar, la fonction « **setListWalk** » fait appel à la fonction « **getWalkByPersonAndCategory** » du CalendarDAO. Celle-ci fait 3 choses. Premièrement, elle regarde si le paramètre « person » est nul et si le paramètre « category » n'est pas nul. Dans ce cas, on va rechercher toutes les balades correspondantes à la catégorie. Deuxièmement, elle regarde si le paramètre « person » n'est pas nul et va rechercher toutes les balades correspondantes à la catégorie de la personne. Troisièmement, elle va rechercher toutes les balades existantes.
- Dans le POJO Category, la fonction « **checkDouble** » va parcourir une liste avec toutes les catégories et va les comparer avec les catégories qu'une personne possède déjà. On va donc obtenir la liste des catégories que la personne n'a pas encore.
- Dans le POJO Registration, on fait appel à deux fonction similaire « **getPassengerCount** » et **getBikeCount** ». Ces deux fonctions du RegistrationDAO vont rechercher pour une balade en particulier toutes les demandes de place comme passager et toutes les demandes de place pour un vélo. Elle va les compter et renvoyer le résultat.

Accès au programme

Pour accéder au programme, des utilisateurs ont été créés :

- Responsable
 - Username : admin Mot de passe : 1234
- Trésorier
 - Username : Quentin Mot de passe : 1234
- Membres
 - Username : Tess Mot de passe : 1234
 - Username : Viny Mot de passe : 1234

Pour démarrer le programme, vous pouvez vous mettre sur n'importe quelle frame et l'application se lancera avec la page de connexion.

Lorsque vous vous inscrivez, vous êtes un « membre » par défaut.

Voici le lien pour accéder au git du projet : https://github.com/TessVeltri/ClubDeVelo_JAVA