Gabriel Huygebaert

Projet JAVA 2022

Gestion d’un club de cyclistes



# Énoncé

Le client est un club de cyclistes qui aimerait s’informatiser.

Voici la façon dont ce club fonctionne :

Le club contient en son sein des amateurs de

* VTT
  + Des descendeurs
  + Des randonneurs
  + Des « trialistes »
* Cyclo (vélo sur route)

Chaque membre du club doit appartenir à au moins une des catégories.

Le club participe régulièrement à des balades. Dans ce cas, il faut se déplacer vers le lieu de départ d’une de ces balades. Afin d’organiser au mieux le covoiturage, une application sera mise au point afin d’optimiser l’offre et la demande : Chaque membre ayant la possibilité de transporter un (ou plusieurs) vélo(s) et / ou membre(s) du club « postera » la (ou les) place(s) disponible(s), tant pour les vélos que pour les membres.

Chaque membre « authentifié » aura donc la possibilité de poster ses disponibilités ou de réserver de la place pour lui-même et / ou son vélo.

Pour chaque catégorie, un « responsable » sera désigné afin d’organiser (et publier) le calendrier des sorties pour cette catégorie.

Lors de chaque balade, le départ du covoiturage sera organisé au départ de l’adresse du club.  
Afin de ne pas léser les personnes mettant à disposition leur véhicule, un « forfait » sera calculé lors de chaque déplacement par le « responsable » de catégorie.  
Le suivi des remboursements vers les « chauffeurs » et le paiement des « passagers » sera assuré par le trésorier du club.

Pour chaque balade, un récapitulatif des disponibilités sera indiqué, afin de savoir s’il manque des chauffeurs ou si au contraire certains chauffeurs sont superflus.

Afin d’être en ordre d’inscription, chaque membre devra s’acquitter d’une cotisation annuelle de 20 EURO. S’il désire s’inscrire dans d’autres catégories, un supplément de 5 EURO par catégorie supplémentaire lui sera demandé.

Le trésorier du club est chargé de s’assurer que les membres ont bien payé leur(s) cotisation(s)

Il vous est demandé de modéliser cette applicaiton à l’aide des notations UML suivantes :

* Cas d’utilisation (use cases).
* Diagramme de classes.
* Diagrammes de séquences

Le programme implémentant la modélisation UML sera écrit en Java. Les données seront sauvegardées dans une base de données Oracle (localisée sur le serveur Oracle de l’école) ou Access (Tout doit être inclus dans le programme pour son exécution). Vous utiliserez le pattern DAO et WindowBuilder pour la partie graphique.

**Le lundi 29 novembre 2021 entre 8h et 12h sur Moodle**, vous devrez remettre dans un dossier compressé à votre nom :

* Le rapport, **en format pdf**, comprenant la modélisation UML (use cases, diagrammes de classes, diagrammes de séquence et explications), l’explication de l’implémentation du pgm Java et la manière d’accéder à l’application en indiquant des usernames si nécessaire. Ajoutez également le lien Github dans le rapport.
* le code Java : donner le workspace de votre application. Tout doit être prévu dans le programme pour son exécution. Testez votre programme sur plusieurs Pcs différents.

Table des matières

[Énoncé 1](#_Toc86056348)

[Façon d’utiliser le programme 4](#_Toc86056349)

[Base de données 5](#_Toc86056350)

[Analyse 6](#_Toc86056351)

[Use case diagram 6](#_Toc86056352)

[Jet 1 6](#_Toc86056353)

[Class diagram 7](#_Toc86056354)

[Mon jet 7](#_Toc86056355)

[Correction commune proposée par l’enseignant 8](#_Toc86056356)

[Sequence diagram 10](#_Toc86056357)

[Conclusion 11](#_Toc86056358)

# Façon d’utiliser le programme

Une image contenant texte

Description générée automatiquement

Librairies -> Class path -> Add JARs -> Sélectionner les 5 fichiers qui sont déjà dans la racine

Une image contenant texte, capture d’écran, moniteur, intérieur

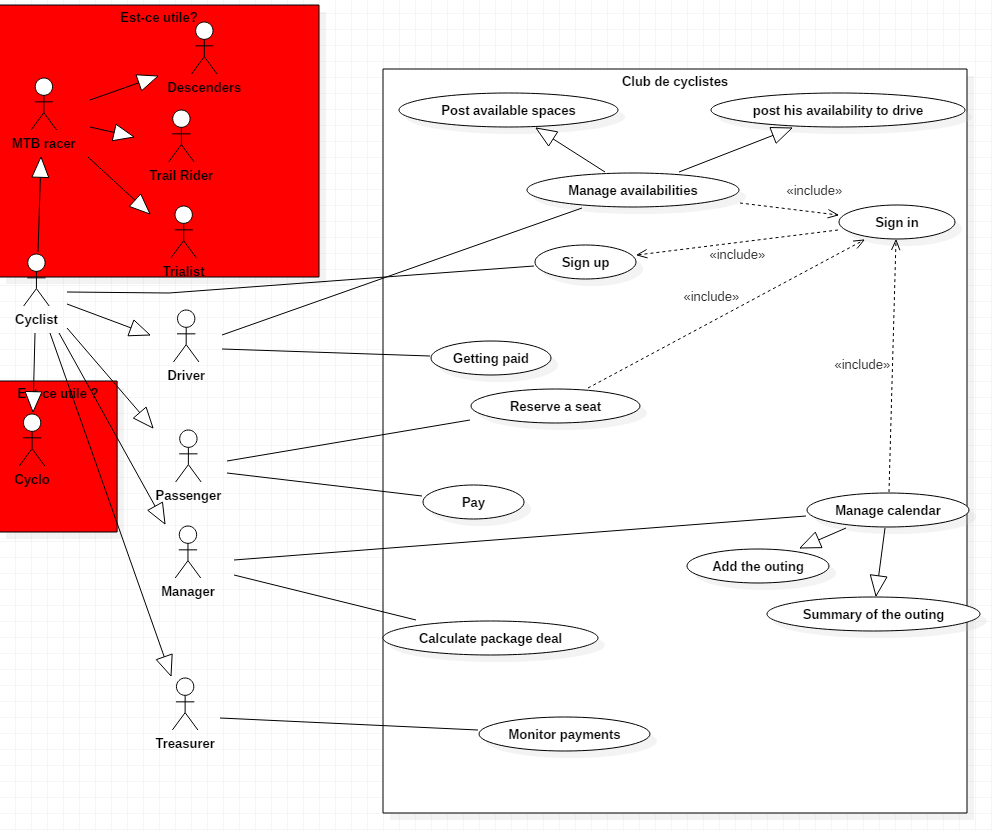
Description générée automatiquement

# Base de données

# Analyse

## Use case diagram

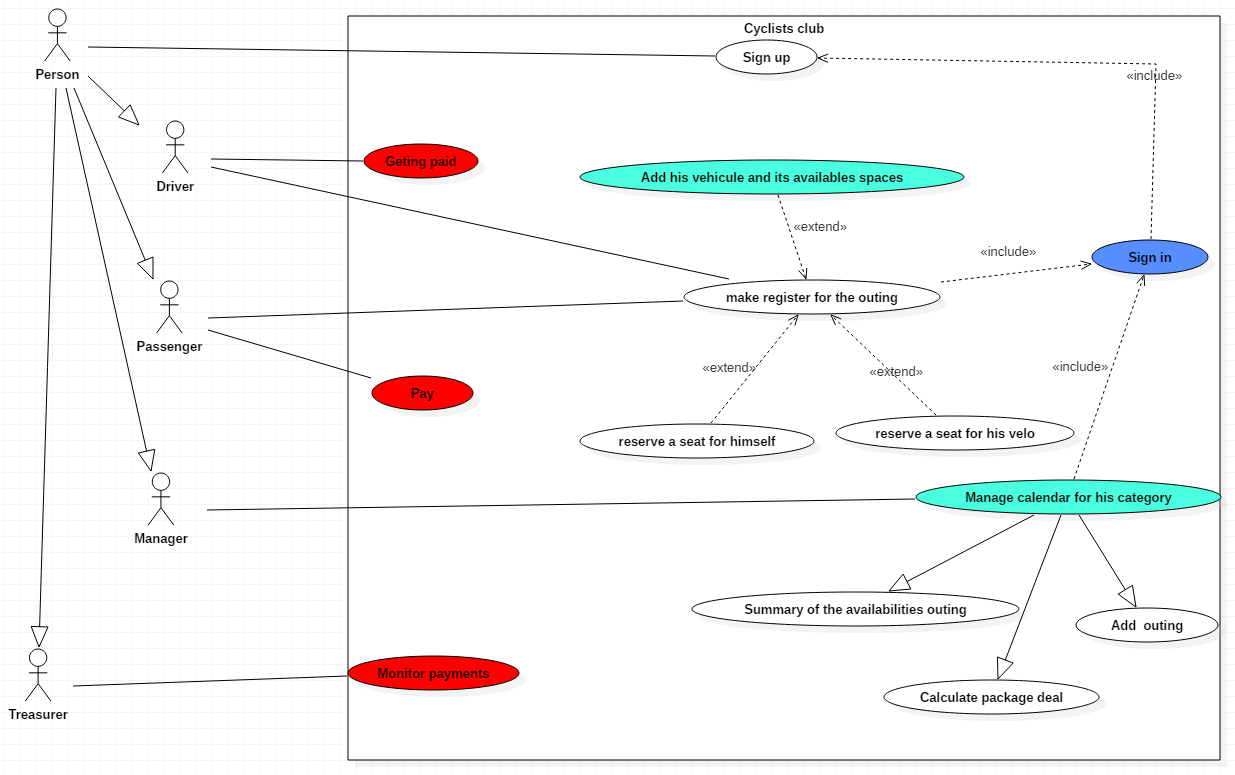
### Jet 1



#### Commentaire

/

## Jet 2

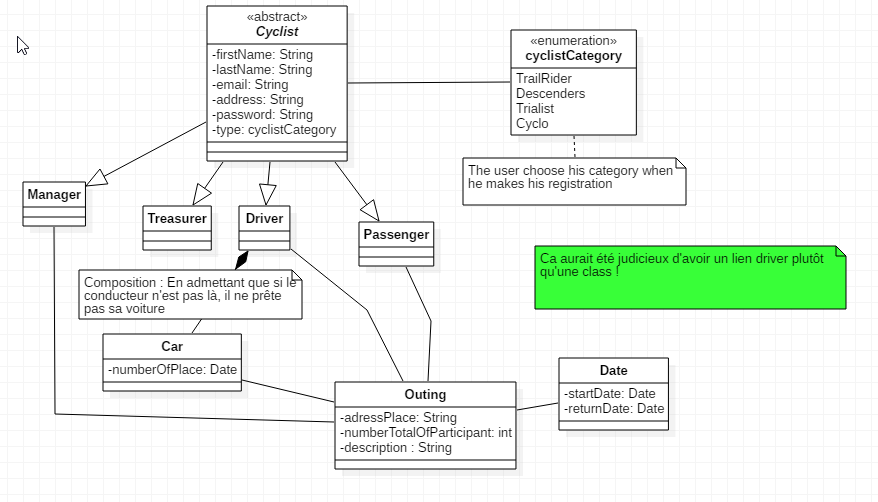


## Commentaire

Bien que le conducteur et le passager soient deux acteurs dans le use case, cela ne veut pas dire qu’ils seront caractérisés par deux classes distinctes dans le diagramme de classe. Ces derniers sont des membres et ce sera leur relation avec le véhicule qui les différenciera.

## Class diagram

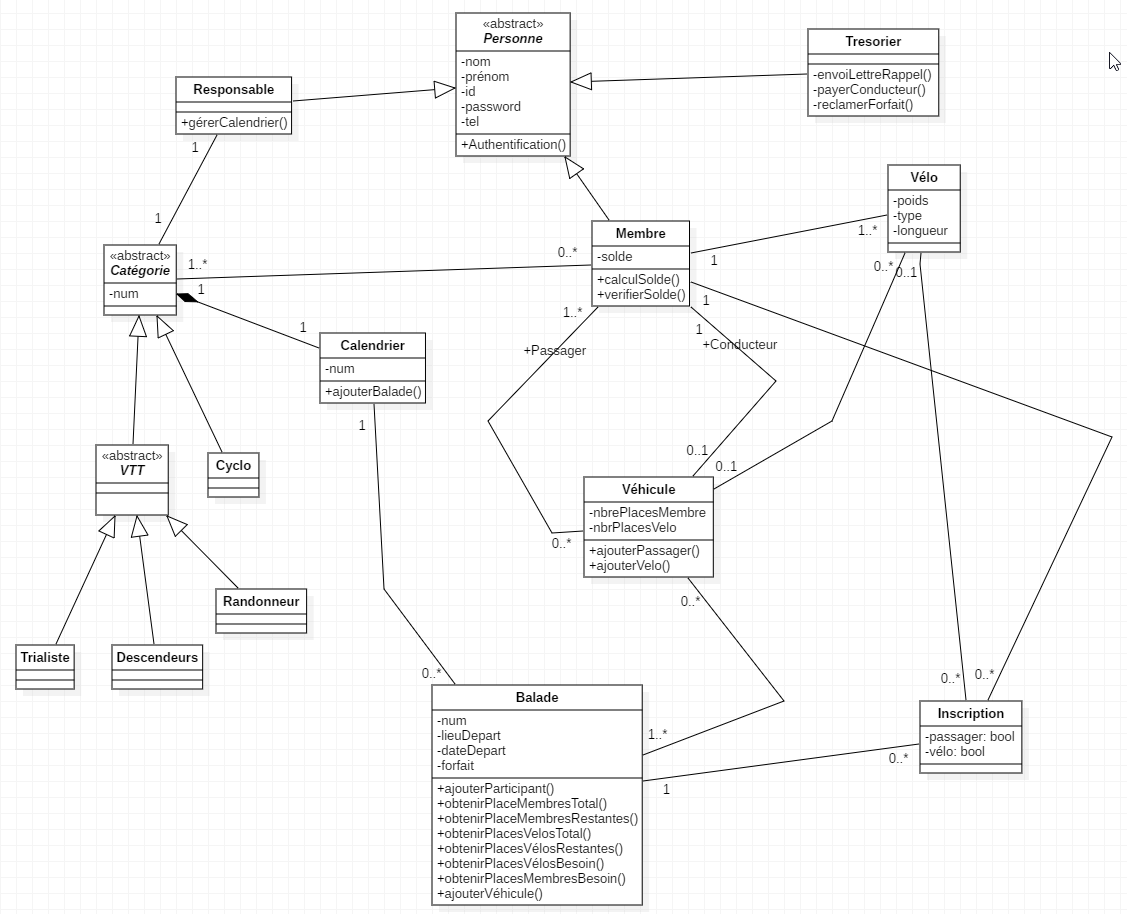
### Mon jet



#### Commentaire

Mauvaise perception de ce qui était demandé. Il aurait été plus judicieux d’avoir des liens nommés plutôt que de créer deux class (Passager/conducteur).

### Correction commune proposée par l’enseignant



#### Commentaire

Pas de sortie en tandem. Un vélo = une personne. Le conducteur n’a qu’un seul ou aucun véhicule.   
Le calendrier n’existe pas si une catégorie n’existe pas ( prévoir que le calendrier peut être supprimé ).  
Le forfait pourrait être un total pour tout le monde à diviser et payer ou un forfait individuel.   
Le participant n’est pas obligé de prendre une place dans le véhicule.

* Quand il s’inscrit en tant que participant, il doit alors préciser s’il a besoin ou non d’une place pour le vélo ou lui-même.

Dans l’héritage, s’il n’y a pas de méthodes ou d’attributs en particulier dans les class filles, il est possible de remplacer par un énumérateur. ( Cf. les catégories )

## Sequence diagram

# Conclusion