

Application Mobile de Covoiturage

Objectif du document

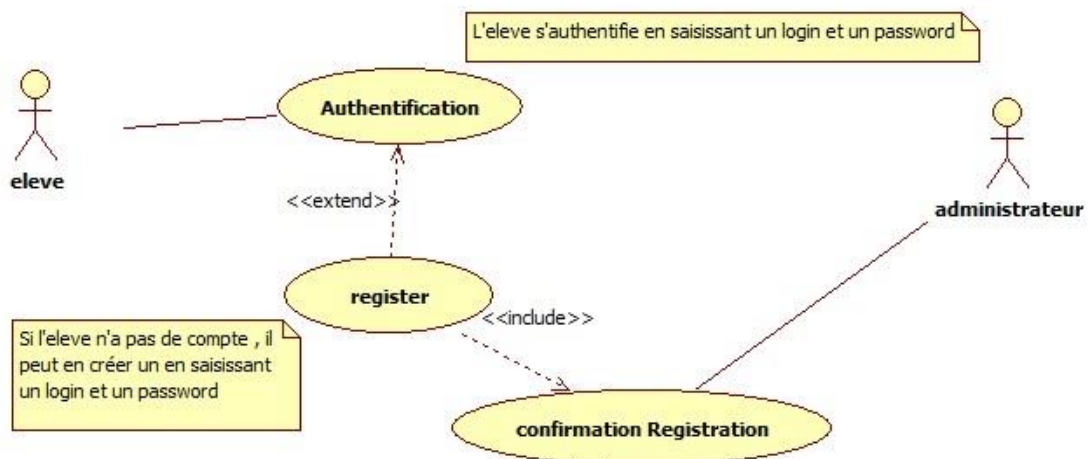
Compte tenu des prix particulièrement élevés des carburants aujourd'hui, Le GRETA souhaite créer sa propre application de covoiturage pour ses élèves.

On vous demande dans un premier temps de créer une API REST.

Dans un second temps une application Front permettra aux élèves d'utiliser ce service.

Cette analyse faite à partir de la méthode UML va permettre d'identifier les objets relatifs aux différents cas d'utilisation, ainsi que le comportement statique et dynamique de chaque objet.

Cas d'utilisation Authentification



Description des interactions entre objets

L'utilisateur saisit son login et son password sur l'application mobile.

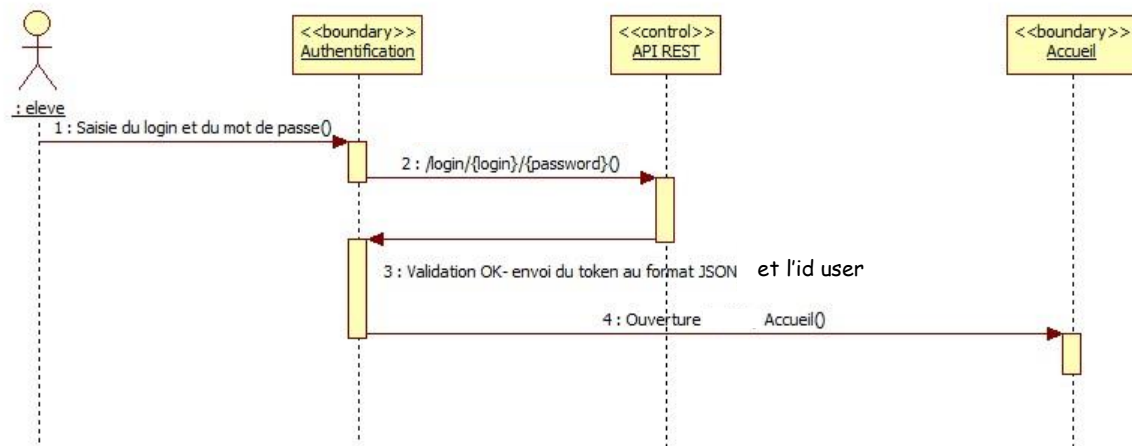
Ses données sont envoyées sur une API REST via la méthode POST.

Si le mot de passe et le login sont corrects, l'API REST renvoie un **token de session** et l'**id user** au format JSON.

Une fois le token réceptionné par l'application, le front lance la page d'accueil.


Dans le cas contraire, l'application renvoie sur le formulaire d'authentification, un **message d'erreur(login ou mot de passe incorrect)**.

Projet Covoiturage - soutenance 4



Vue Authentification

Authentication

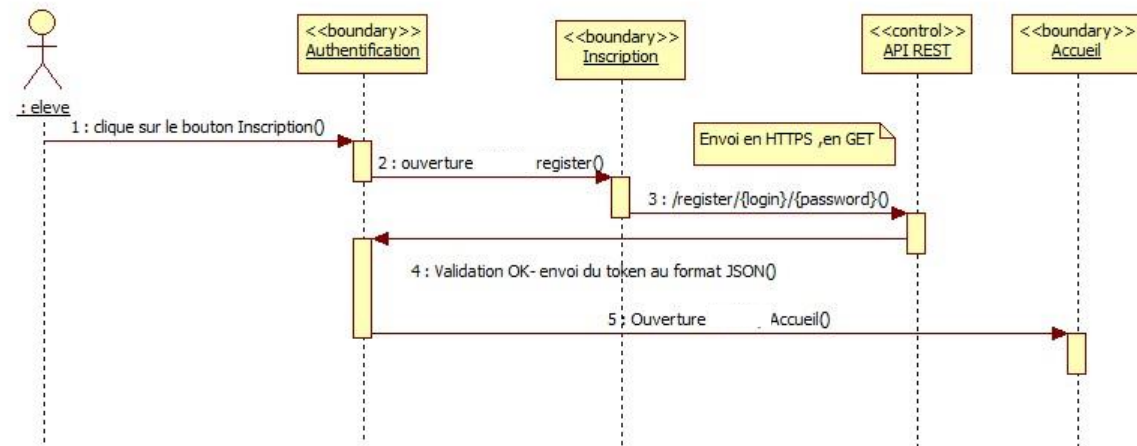


Connexion

[Connexion](#) [Inscription](#) [Effacer](#)

Description des interactions entre objets (Use case Register)

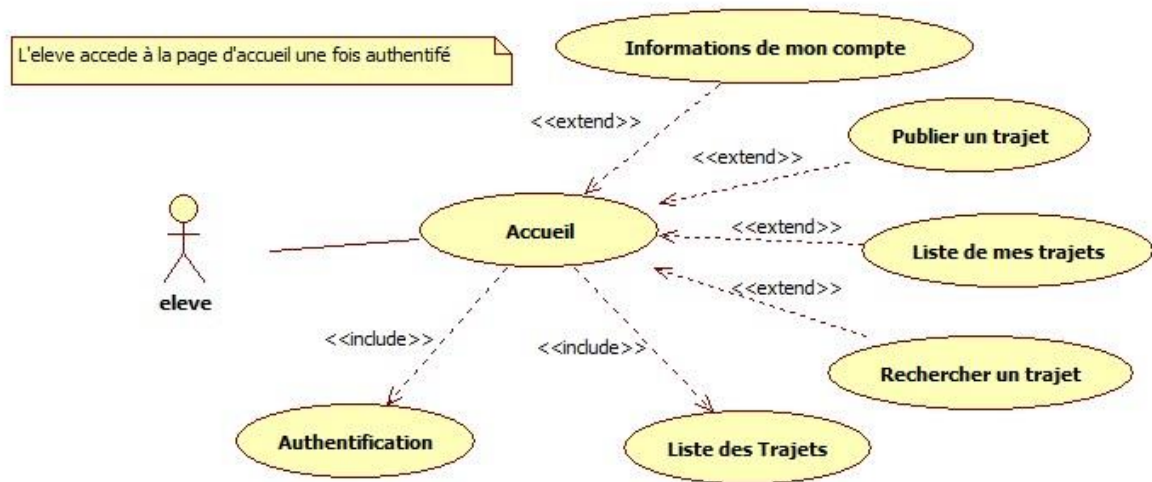
L'utilisateur n'a pas de login et de mot de passe et clique sur le bouton Inscription pour créer son compte.



Le formulaire d'inscription doit lui proposer de confirmer son mot de passe. Si les mots de passe ne sont pas identiques, le formulaire affiche un message d'erreur (mots de passe pas identiques) et réinitialise les champs. Il faut utiliser la méthode POST pour envoyer les données à l'API.

The screenshot shows a mobile application interface for registration. At the top, there is a blue header bar with a back arrow and the title 'Inscription'. Below the header, there is a 'BACK' button. The main area contains three input fields: 'Pseudo', 'Mot de Passe', and another 'Mot de Passe' field for confirmation. At the bottom, there are two buttons: 'Valider l'Inscription' (blue) and 'Annuler' (blue).

Cas d'utilisation Accueil

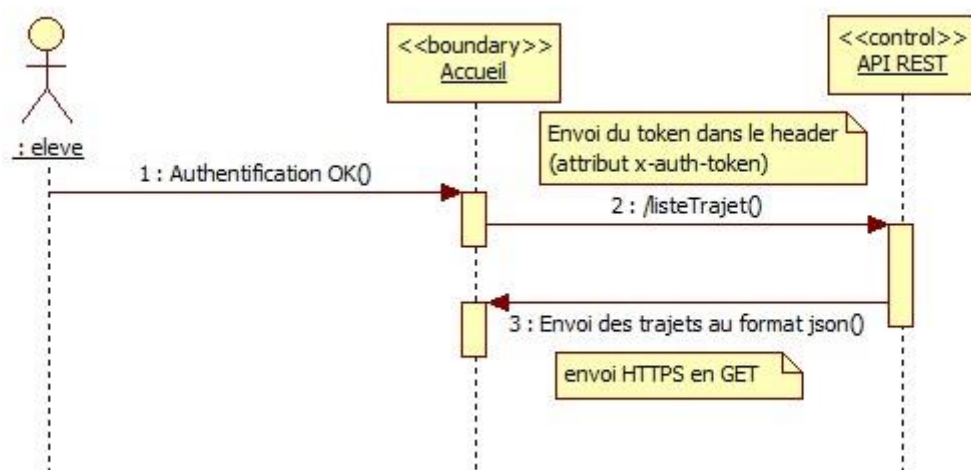


Description des interactions entre objets (Liste des Trajets)

L'utilisateur est authentifié car il a reçu un token lors de la phase d'authentification de l'API REST.

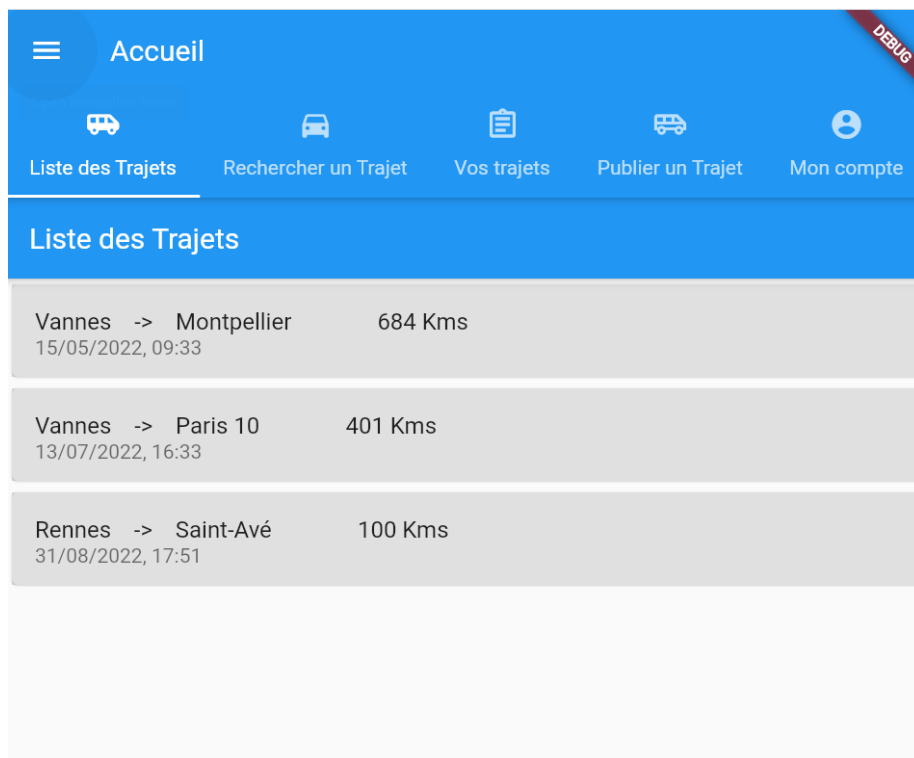
Désormais toutes les phases d'authentications sont faites à partir de ce token, spécifié dans le **header http**, dans l'attribut **x-auth-token**.(cf cours API avec symfony)Ce token est **indispensable** pour récupérer les différentes routes de l'API REST.

Une fois authentifié, il visualise directement la liste de tous les trajets proposés par les conducteurs.



Projet Covoiturage - soutenance 4

Vue Accueil



Cas d'utilisation Détail du trajet

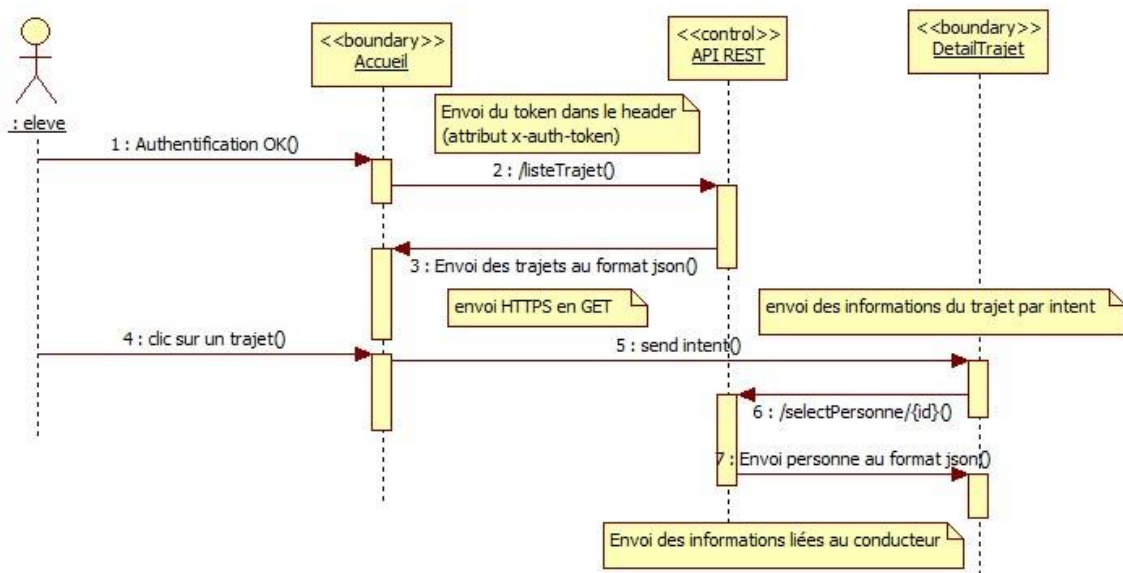


Projet Covoiturage - soutenance 4

Description des interactions entre objets (Use case détail d'un trajet)

L'utilisateur sélectionne un trajet, pour en afficher les détails.

Le détail du trajet permet d'envoyer un email au conducteur, et de réserver une place pour le trajet.



Vue Détail du trajet

The screenshot shows the 'Vue Détail du trajet' (Trip Detail View) in a mobile application. The interface has a blue header with a back arrow and the title 'Détail du Trajet'. Below the header is a 'Back' button. The main content area displays the trip details in a list of fields:

- Conducteur: Lamy Pascal
- Départ: Rennes
- Arrivée: Saint-Avé
- Date: 31/08/2022
- Heure: 17:51
- Nombre de Kilomètres: 684
- Nombre de Places: 2
- Marque de la voiture: BMW
- Modèle de la voiture: 325 TDS

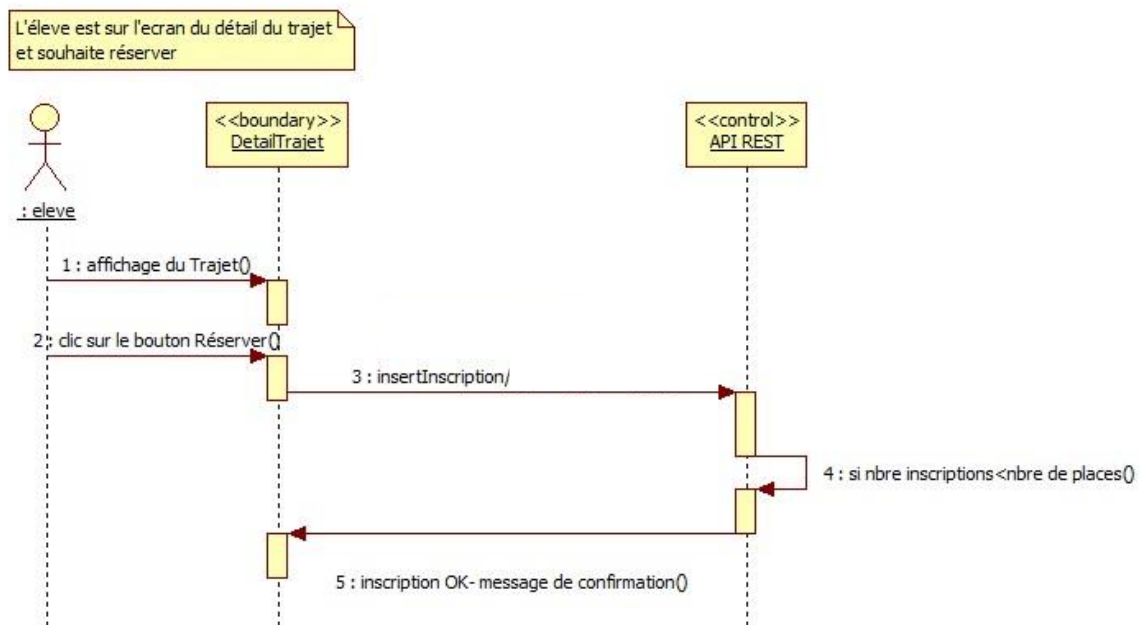
At the bottom of the screen, there are three buttons: 'Envoyer un email', 'Réserver', and 'Annuler'.

Description des interactions entre objets (Use Case Réserver une place)

L'utilisateur clique sur Réserver pour réserver une place pour ce trajet.

Un contrôle est effectué par l'API REST pour vérifier si le nombre de places est suffisantes.

Dans le cas où il n'y a plus de place, l'application affiche un message **réservation impossible (plus de place)**, sinon **réservation effectuée**.



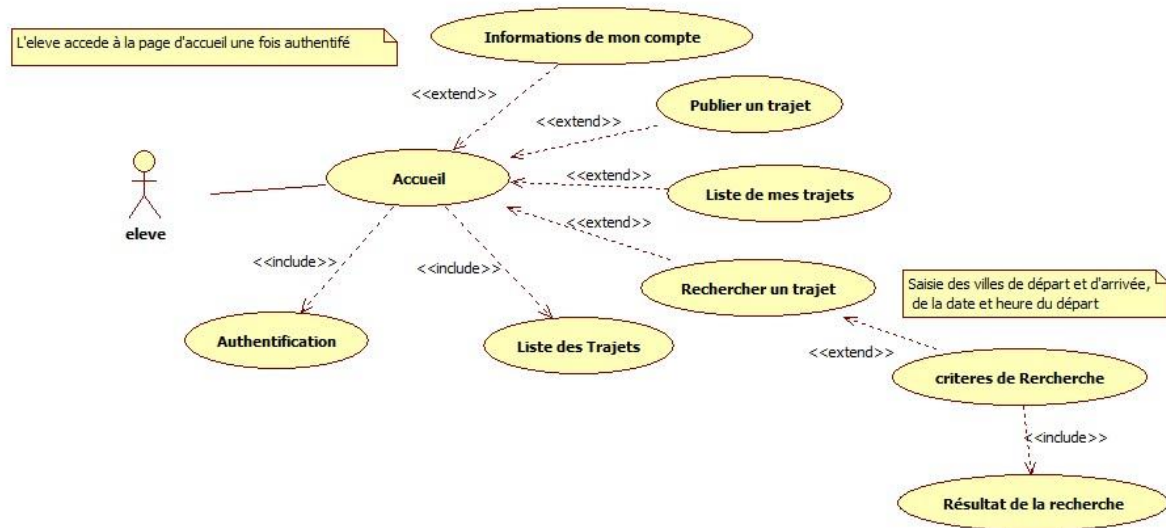
L'API retourne au format JSON le message **OK** indiquant que la réservation est prise en compte.

Dans le cas contraire l'API retourne **NOK**.

Envoi d'un courriel au conducteur

Lorsque l'élève clique sur le bouton **Envoyer un courriel**, l'application l'envoi d'un email au conducteur du trajet, via l'API Brevo.

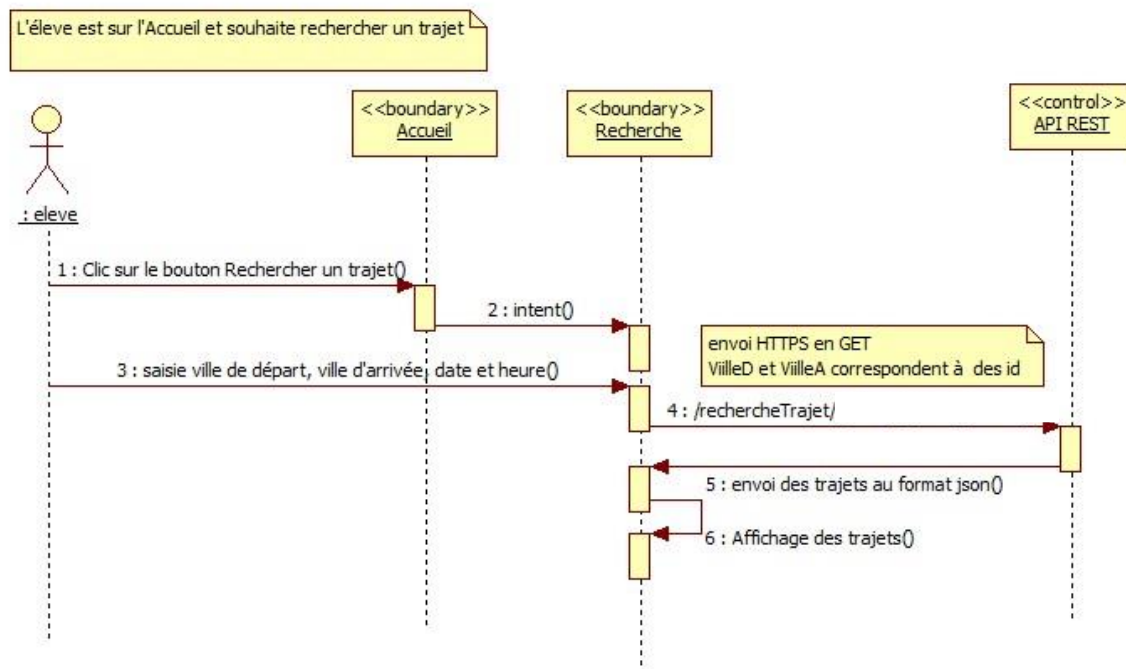
Cas d'utilisation Rechercher un trajet



Description des interactions entre objets (Use case Rechercher un trajet)

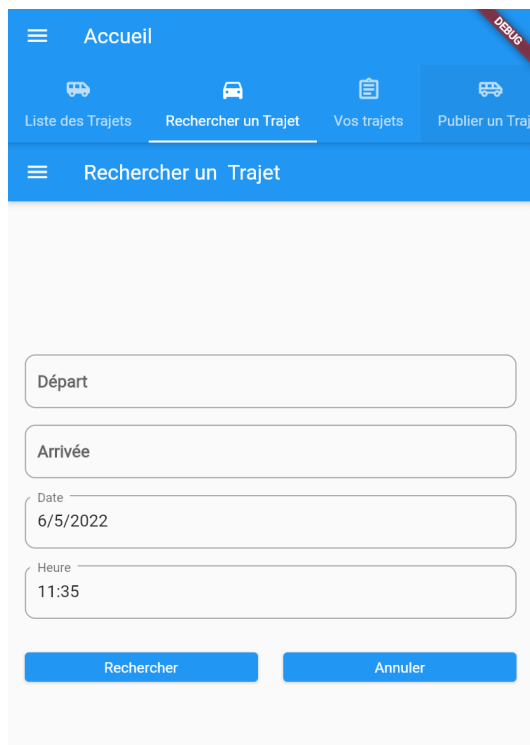
L'utilisateur saisit dans les listes déroulantes la ville de départ ,et la ville d 'arrivée, puis il saisit la date de départ.

L'API REST retourne les résultats de sa recherche au format JSON.



Projet Covoiturage - soutenance 4

Vue Recherche



Accueil

Liste des Trajets Rechercher un Trajet Vos trajets Publier un Trajet

Rechercher un Trajet

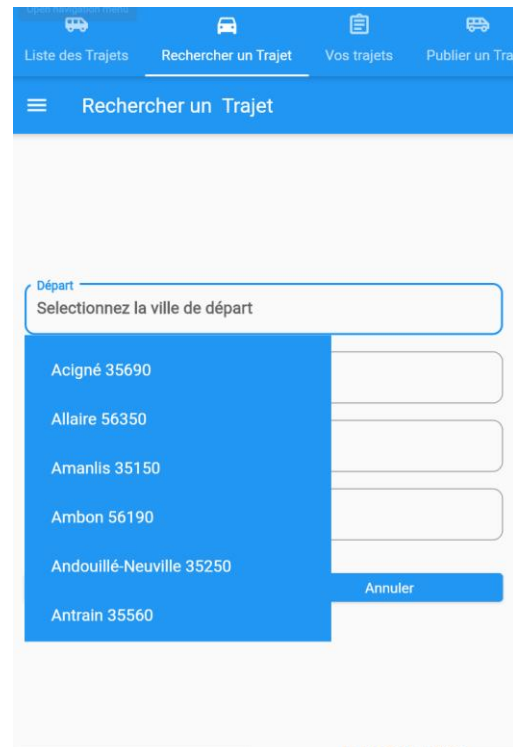
Départ

Arrivée

Date
6/5/2022

Heure
11:35

Rechercher Annuler



Liste des Trajets Rechercher un Trajet Vos trajets Publier un Trajet

Rechercher un Trajet

Départ
Selectionnez la ville de départ

Acigné 35690

Allaire 56350

Amanlis 35150

Ambon 56190

Andouillé-Neuville 35250

Antrain 35560

Annuler

Il est important que l'envoi des données concernant les villes départ et d'arrivée correspondent aux id présents en base de données. Le fichier ville.csv vous sera transmis.

La date sera envoyée au format datetime (AAAA-MM-JJ h :m, ex : 2022-05-16 16 :18)

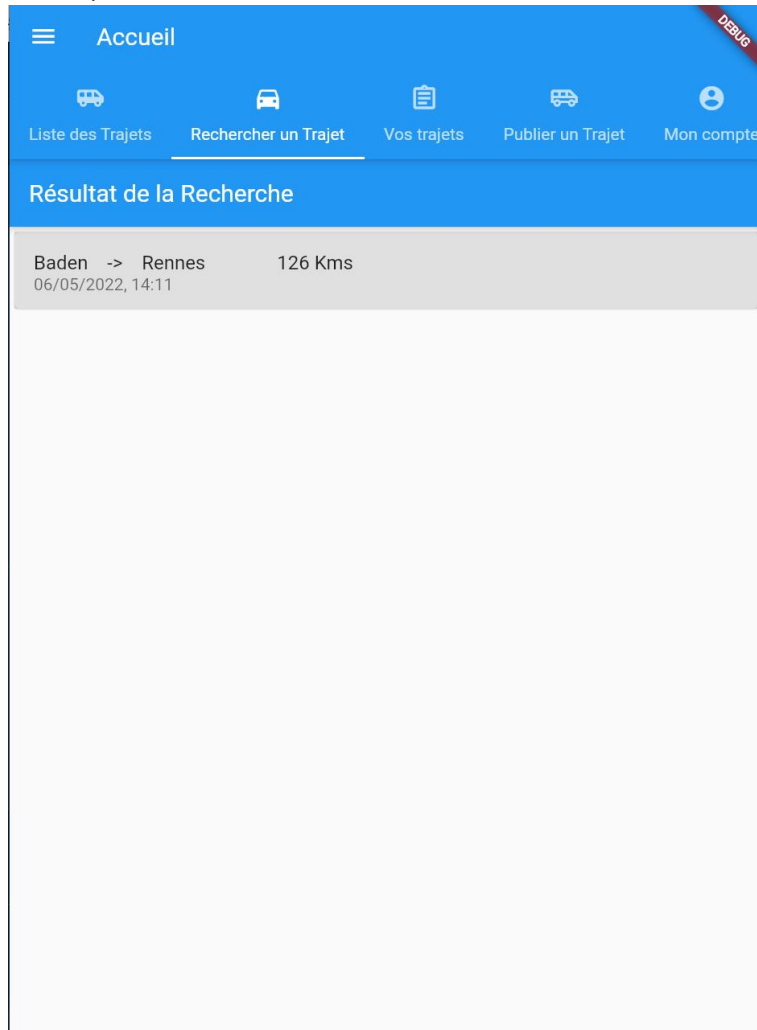
Projet Covoiturage - soutenance 4

The screenshot shows the 'Rechercher un Trajet' screen. The 'Départ' field is set to 'Baden'. The 'Arrivée' field has 're' entered, which has triggered a dropdown menu with the following suggestions: Redon 35600, Renac 35660, Rennes 35000, Rennes 35200, Rennes 35700, and Retiers 35240. An 'Annuler' button is visible at the bottom right of the suggestions list.



The screenshot shows the 'SELECT DATE' modal. The date 'Fri, May 6' is selected and highlighted in blue. The calendar is for May 2022. The days of the week are S, M, T, W, T, F, S. The dates 1 through 31 are displayed in a grid. The 'CANCEL' and 'OK' buttons are at the bottom.

The screenshot shows the 'SELECT TIME' modal. The time '1:11' is displayed, with 'AM' and 'PM' options. A clock face is shown with the hour hand pointing to 1 and the minute hand pointing to 11. The 'CANCEL' and 'OK' buttons are at the bottom.

En cliquant sur le bouton rechercher, la vue suivante s'affiche :



En cliquant sur le trajet, le détail du trajet (affichant également le nombre de places disponibles) s'affiche et permet à l'utilisateur de réserver une place ou d'envoyer un email au conducteur :

 **Détail du Trajet** 

Conducteur

Lamy Pascal

Départ

Baden

Arrivée

Rennes

Date

06/05/2022

Heure

14:11

Nombre de Kilomètres

126

Nombre de Places

4

Marque de la voiture

BMW

Modèle de la voiture

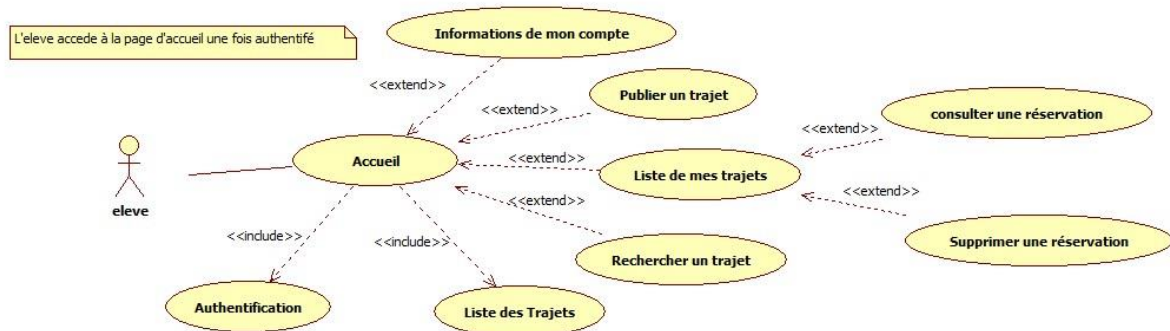
325 TDS

Envoyer un email

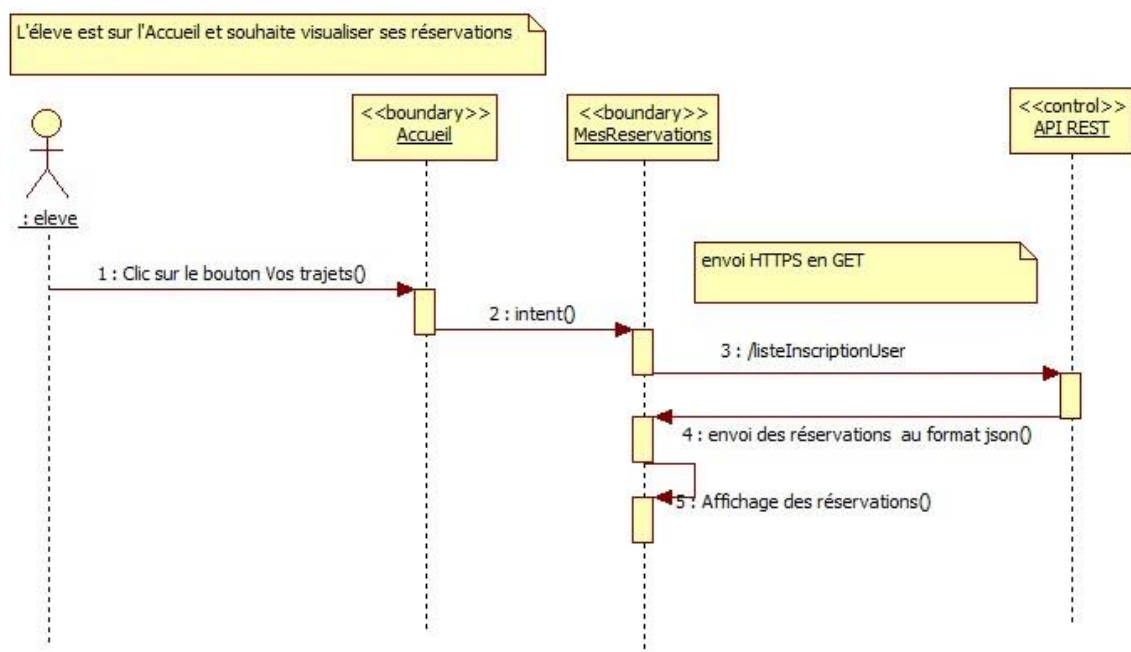
Réserver

Annuler

Cas d'utilisation Liste de mes trajets



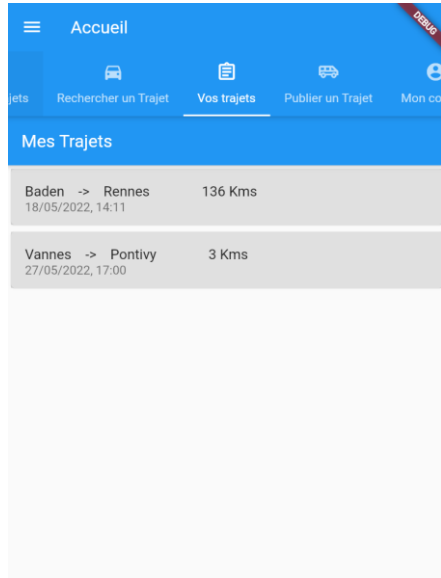
Description des interactions entre objets (Use case Liste de mes trajets)



L'utilisateur clique sur le bouton « Vos Trajets » afin d'accéder à ses réservations. La vue va récupérer la liste de ses réservations au format JSON sur l'Api REST, pour les afficher sous forme de liste.

Projet Covoiturage - soutenance 4

Vue Mes Trajets



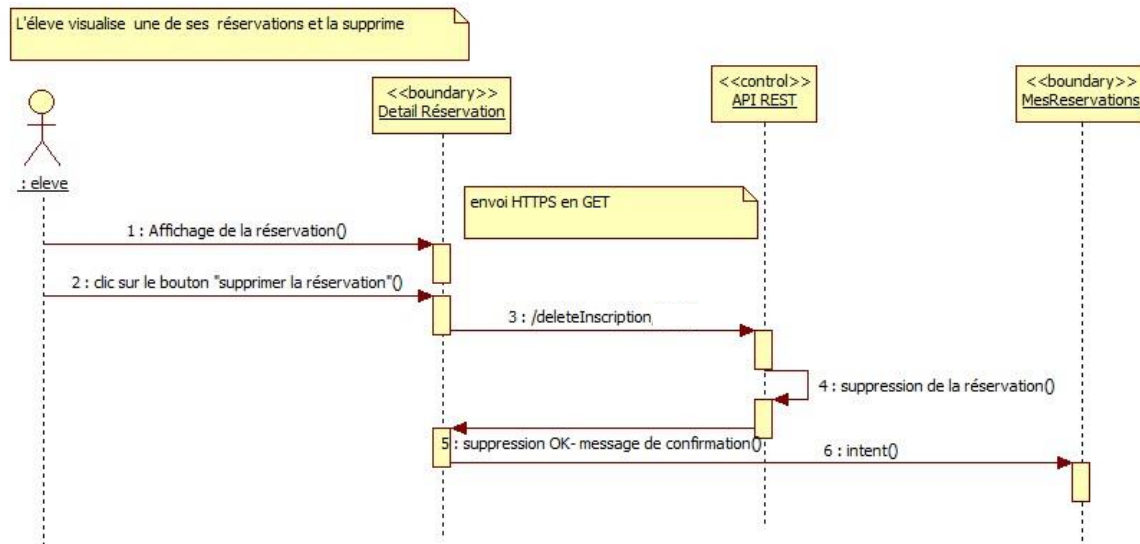
Quand l'utilisateur clique sur un trajet il retrouve le détail de sa réservation.



Projet Covoiturage - soutenance 4

Description des interactions entre objets (Use Case Supprimer une réservation)

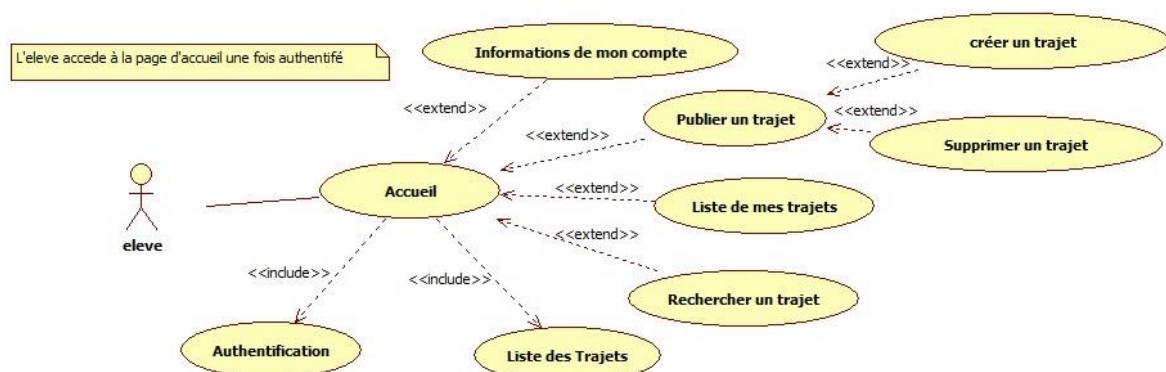
L'utilisateur clique sur « supprimer une réservation » pour annuler sa réservation pour ce trajet.



L'API retourne au format JSON le message **OK** indiquant que la réservation a été supprimée.

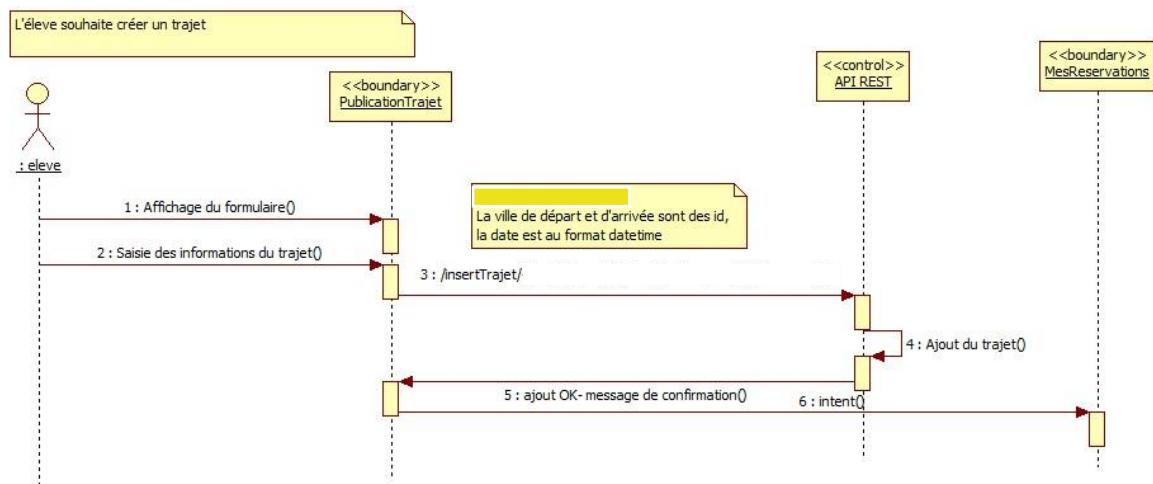
Dans le cas contraire l'API retourne **NOK**. L'application redirige l'utilisateur sur ses réservations.

Cas d'utilisation Publier un Trajet



Projet Covoiturage - soutenance 4

Description des interactions entre objets (Use case Publier un trajet)



*Il est important que l'envoi des données concernant les villes départ et d'arrivée correspondent aux **id** présents en base de données. Le fichier ville.csv vous sera transmis.*

La date sera envoyée au format datetime (AAAA-MM-JJ h :m, ex : 2022-05-16 16 :18)

Vue Publier Trajet

Accueil

des Trajets Rechercher un Trajet Vos trajets **Publier un Trajet** Mon compte

Publier un Trajet

Départ

Arrivée

Date
06/05/2022

Heure
16:34

Distance en Kms

Valider Annuler

Projet Covoiturage - soutenance 4

Accueil

des Trajets Rechercher un Trajet Vos trajets Publier un Trajet Mon compte

Publier un Trajet

Départ

Saisissez la ville de départ

- Acigné 35690
- Allaire 56350
- Amanlis 35150
- Ambon 56190
- Andouillé-Neuville 35250
- Antrain 35560

Annuler

Accueil

des Trajets Rechercher un Trajet Vos trajets Publier un Trajet Mon compte

Publier un Trajet

Départ

Andouillé-Neuville

Arrivée

Saisissez le lieu de destination

- Acigné 35690
- Allaire 56350
- Amanlis 35150
- Ambon 56190
- Andouillé-Neuville 35250
- Antrain 35560

Annuler

Accueil

des Trajets Rechercher un Trajet Vos trajets Publier un Trajet Mon compte

Publier un Trajet

Départ

Andouillé-Neuville

Arrivée

Pontivy

Date

06/05/2022

Heure

16:34

Distance en Kms

VALIDER

Annuler

Accueil

des Trajets Rechercher un Trajet Vos trajets Publier un Trajet Mon compte

Publier un Trajet

Départ

Andouillé-Neuville

Arrivée

Pontivy

Date

11/05/2022

Heure

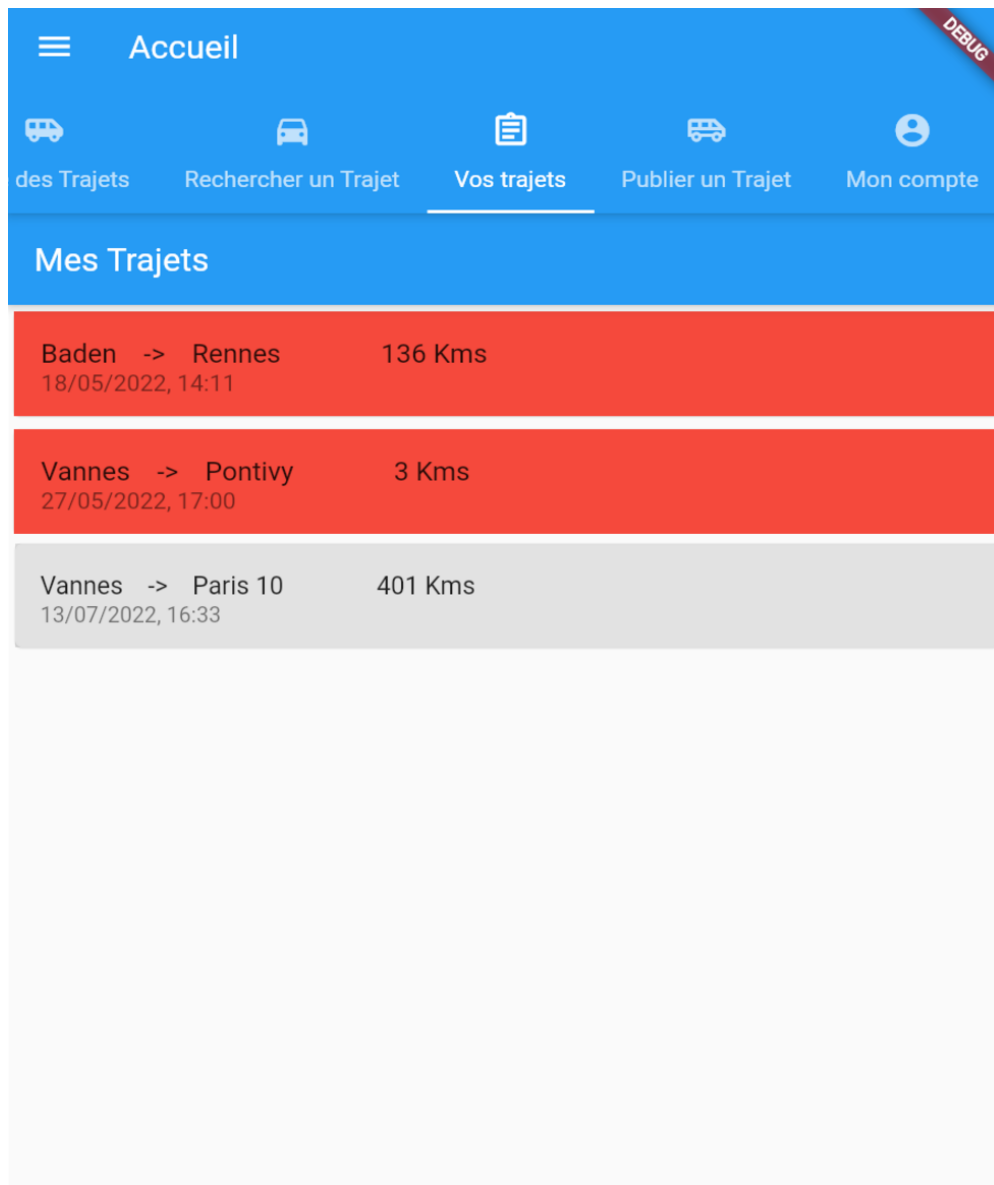
16:34

Distance en Kms

VALIDER

Annuler

Lorsque l'utilisateur clique sur le bouton Valider, l'application renvoie sur la vue **Vos Trajets**.

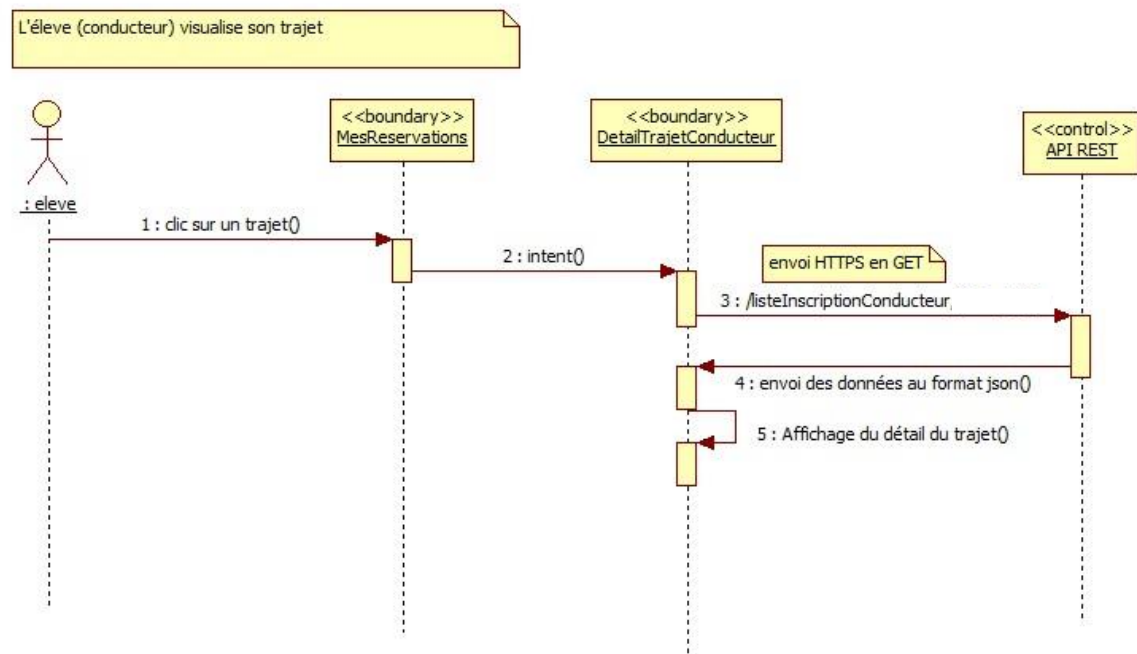


En rouge sont affichés les trajets proposés par l'utilisateur.

En gris sont affichés ses réservations.

Projet Covoiturage - soutenance 4

Description des interactions entre objets (Use Case Détail du trajet conducteur)



Vue détail trajet conducteur

← Détail du Trajet DEBUG

Départ
Vannes

Arrivée
Pontivy

Date
27/05/2022

Heure
17:00

Nombre de Kilomètres
65

Passager
Nicolas Sarkozy

Passager
Emmanuel Macron

Passager
François Hollande

Envoyer un email aux passagers

Supprimer le trajet

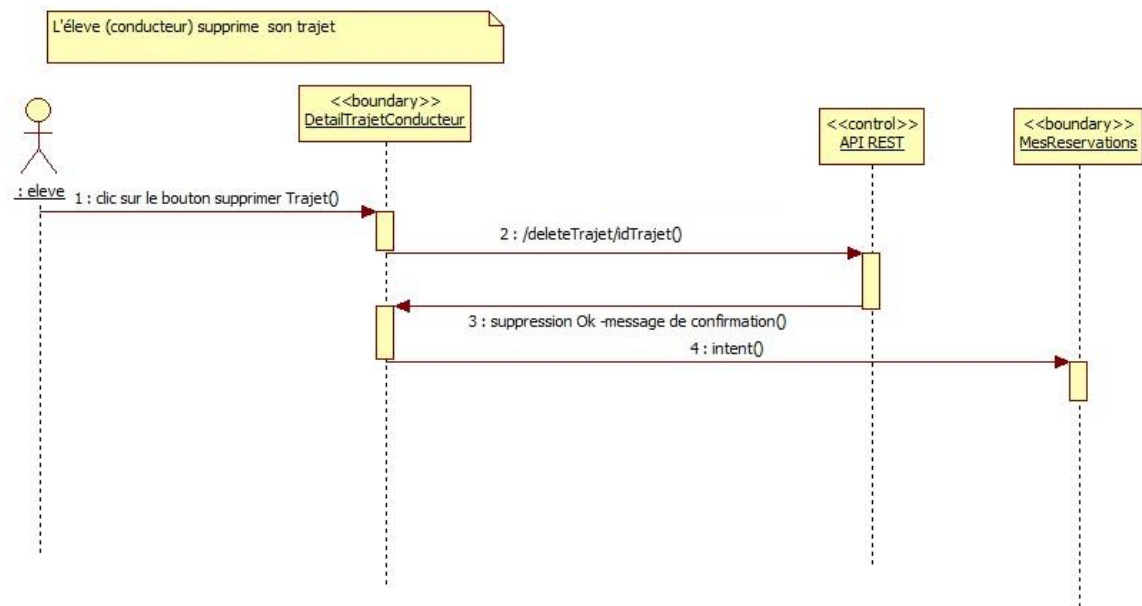
Annuler

Le conducteur peut envoyer un email à tous les passagers.

Le clic sur le bouton « envoyer un email aux passagers » déclenche l'envoi d'un email via l'API Brevo, indiquant la reservation est annulée.

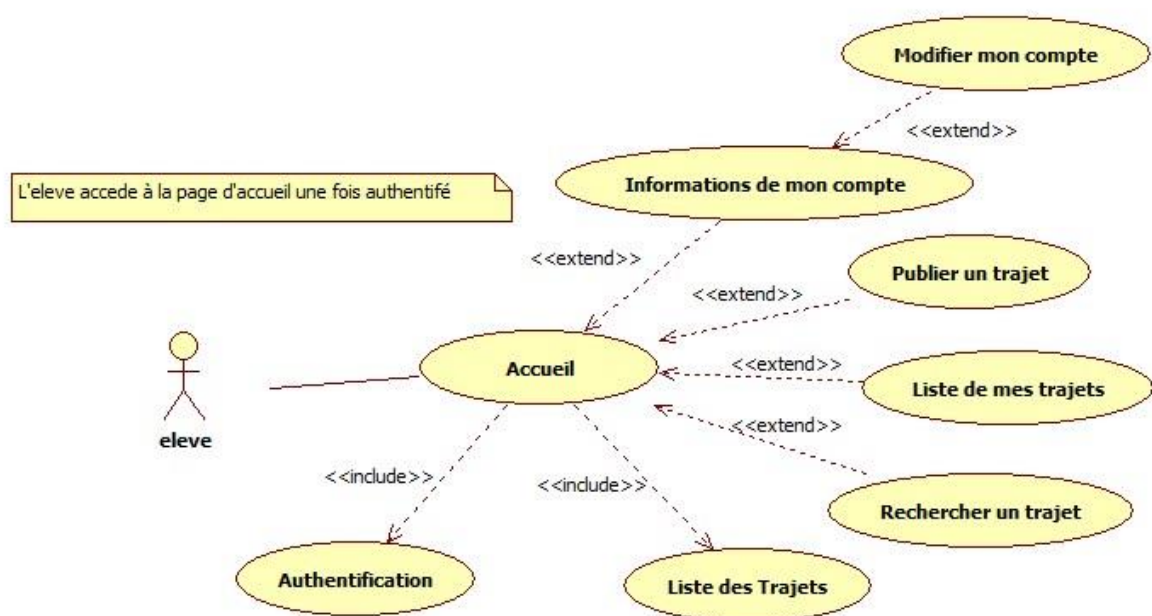
Le clic sur le bouton « supprimer » déclenche la suppression du trajet et renvoie l'utilisateur ses réservations.

Description des interactions entre objets (Use Case Supprimer un trajet)

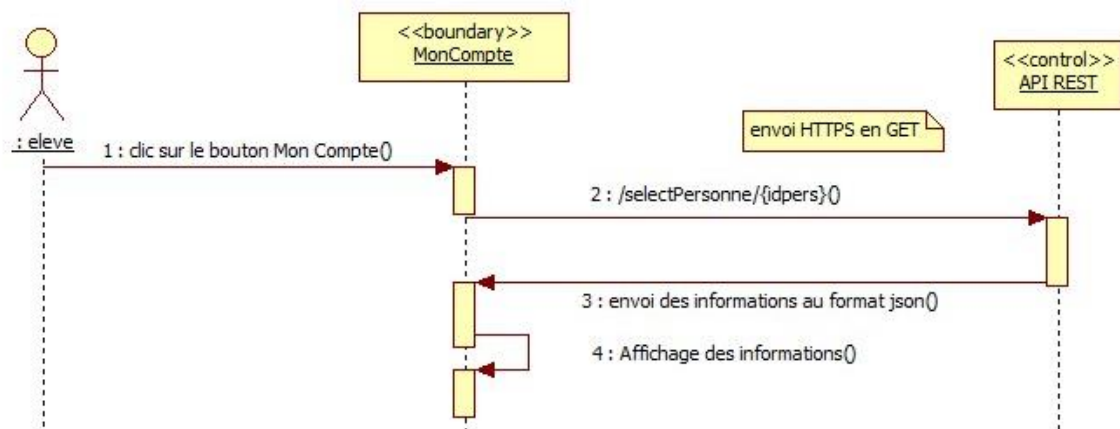


En supprimant son trajet, l'API REST supprime également toutes les réservations.

Cas d'utilisation Information sur mon compte



Description des interactions entre objets (Information de mon compte)



Vue Mon Compte

The screenshot shows the 'Mon Compte' screen of a mobile application. At the top, there is a blue navigation bar with a hamburger menu icon, the title 'Accueil', and a 'Debug' button. Below the navigation bar, there are five icons representing different features: 'Liste des Trajets', 'Rechercher un Trajet', 'Vos trajets', 'Publier un Trajet', and 'Mon compte' (which is currently selected). The main content area has a blue header 'Mon Compte'. Below this, there are several input fields for user information: 'Pseudo' (Mr LAMY), 'Prénom' (Pascal), 'Nom' (Lamy), 'Téléphone' (0645124578), 'Email' (pascal@lamy.mobi), 'Marque Voiture' (BMW), 'Modèle voiture' (325 TDS), and 'Nombre de places' (4). At the bottom, there are three buttons: 'Modifier' (blue), 'Annuler' (blue), and 'Déconnexion' (red).

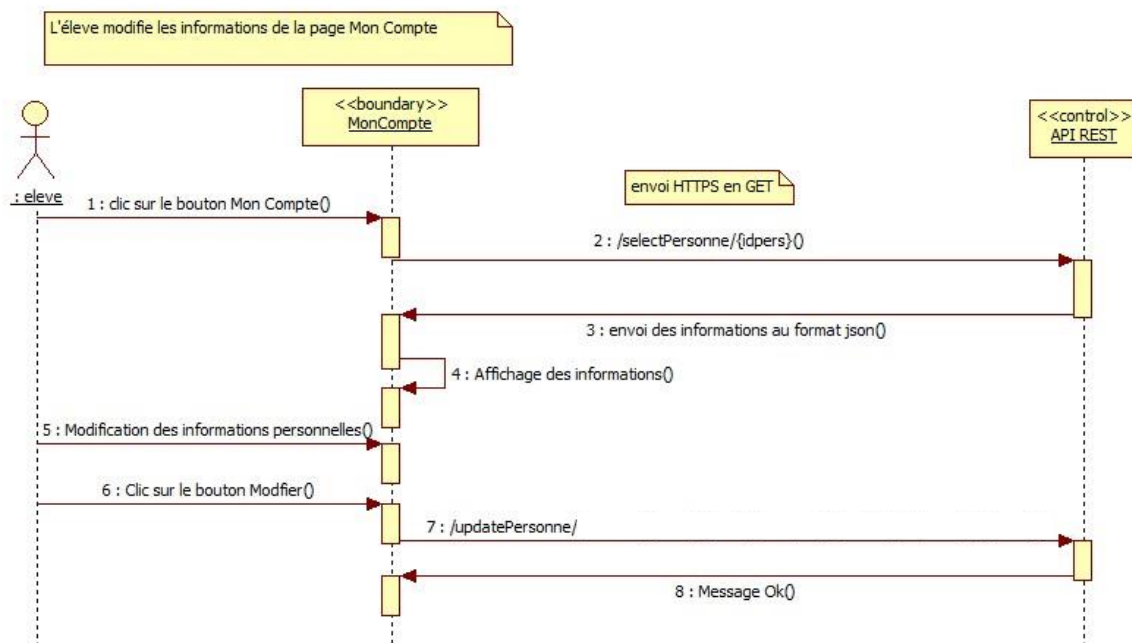
Description des interactions entre objets (Modifier mon compte)

L'utilisateur a la possibilité de changer les informations de son compte.

Si l'utilisateur possède une voiture, une liste déroulante propose la liste des marques et deux champs textes permettent de saisir le modèle et le nombre de places.

En outre, l'utilisateur peut changer son adresse email, son téléphone, son nom et son prénom.

En aucun cas, il peut changer son login.



Un message s'affichera sur l'écran indiquant que la modification a bien été effectuée.

Le bouton Annuler renvoie l'utilisateur sur la liste des trajets.

Le bouton Déconnexion ferme l'application.

Tests

Vous devez mettre en place des tests sous Symfony pour tester les routes marque, personne, trajet, register, login, ville et voiture.

Pour réaliser des tests sous Symfony, utilisez la commande suivante :

```
symfony console make:test
```

Il est important que les tests soient automatisés, dans Gitlab.

Travail à réaliser

Vous devez concevoir l'API REST permettant de répondre à l'ensemble des cas d'utilisation, à partir du Framework Symfony.

Vous devrez concevoir la base de données sous Mysql.

Il est important de définir les contraintes d'intégrités référentielles au niveau des relations des tables.

Vous devez planifier votre charge de travail ainsi qu'un rétro-planning sous Jira, pour gérer au mieux la réalisation de ce projet.

Vous devez également concevoir un dossier d'architecture technique expliquant le fonctionnement technique de votre API (MCD, MLD, diagrammes de classe ...).

Vous devez mettre en place des tests d'intégration. Une fois la réussite des tests effectués via Gitlab, le déploiement de l'application doit s'effectuer automatiquement sur votre espace web O2Switch.

Un rapport sur la qualité de votre code devra être généré par SonarQube.

Lors de la soutenance, les formateurs testeront l'application via Insomnia.



Vous devez solliciter les formateurs si vous avez des questions sur les spécifications techniques et fonctionnelles de l'application.

Annexe

Liste des routes de l'API REST

Nom de la route	Champs	verbe
listeInscriptions		GET
listePassagersConducteur	{idtrajet}	GET
listeInscriptionsUser	{idpers}	GET
inscription	{idpers},{idtrajet}	POST
inscription	{id}	DELETE
marque	{nom}	POST
marque	{id}	DELETE
marque		GET
listePersonnes		GET
personne	{prenom},{nom},{tel},{email},{ville},{voiture}	POST
personne	{id}	DELETE
personne	{id}	GET
personne	{prenom},{nom},{tel},{email},{marque},{modele},{nbplaces},{idpers}	PUT
listeTrajets		GET
findTrajet	{villeD},{villeA},{dateT}	GET
trajet	{kms},{idpers},{dateT},{villeD},{villeA}	POST
trajet	{id}	DELETE
register	{login},{password}	POST
login	{login},{password}	POST
ville	{ville},{cp}	POST
ville	{id}	DELETE
listeVilles		GET
listeCodePostals		GET
voiture	{modele},{place},{marqueId},{immatriculation}	POST
voiture	{id}	DELETE
listeVoitures		GET