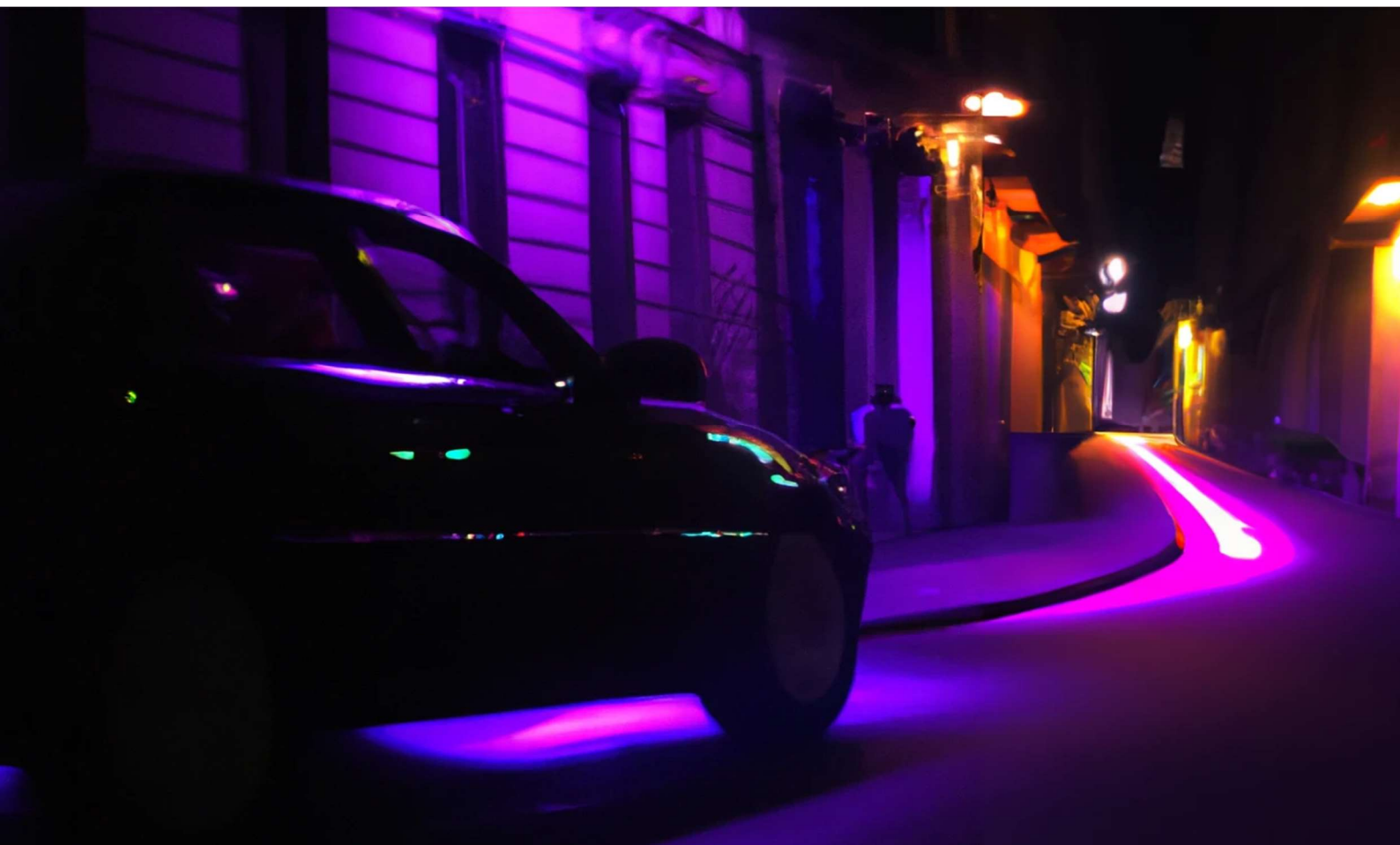


# DriveMyNight project

Colliard Thibaut



# Table des matières

Introduction .....	3
Analyse .....	4
Présentation du projet.....	4
a) Les visiteurs.....	4
b) Les utilisateurs.....	4
c) Les utilisateurs ayant ajouté un véhicule.....	4
Uses Case.....	4
a) Use case de la vue des visiteurs du site :.....	4
b) Use case de la vue des utilisateurs du site :.....	5
Maquettes.....	6
a) Visiteur du site web.....	6
b) Page de connexion.....	6
c) Inscription / création de compte.....	7
d) Page utilisateur.....	7
e) Ajout d'un véhicule.....	8
f) Page créateur de véhicule.....	8
g) Page modification de profil.....	9
Diagramme d'activité.....	9
a) Activité connexion.....	9
b) Activité ajouter une voiture.....	10
Diagramme de séquence systèmes.....	11
c) Séquence système se connecter :.....	11
d) Séquence système rejoindre une voiture.....	12
Schémas ER.....	13
Conception.....	14
Diagrammes de classe.....	14
a) Client.....	14
b) Serveur.....	15
Schéma relationnel.....	16
Diagramme séquence interactions.....	17
a) Login.....	17
b) Rejoindre une voiture.....	18
Conceptions des tests.....	18



# Introduction

---

Ce projet vise à créer une plateforme de covoiturage innovante destinée aux participants de soirées et événements, leur permettant de partager des trajets en toute sécurité.



# Analyse

## Présentation du projet

En connectant les conducteurs sobres ayant des places disponibles avec ceux qui cherchent un moyen sûr de rentrer chez eux, nous offrons une solution pratique et responsable face aux risques de conduite après consommation d'alcool. Cette initiative promeut non seulement la sécurité routière, mais encourage également une culture du partage et de la solidarité au sein de la communauté festive.

Les fêtards qui ne consomment pas d'alcool pourront renseigner leur véhicule comprenant leur trajet sur la plateforme et les fêtards alcoolisés pourront rejoindre un véhicule.

### a) Les visiteurs

La plateforme nécessitera une authentification pour être utilisée, les visiteurs ne pourront que se connecter ou s'inscrire.

### b) Les utilisateurs

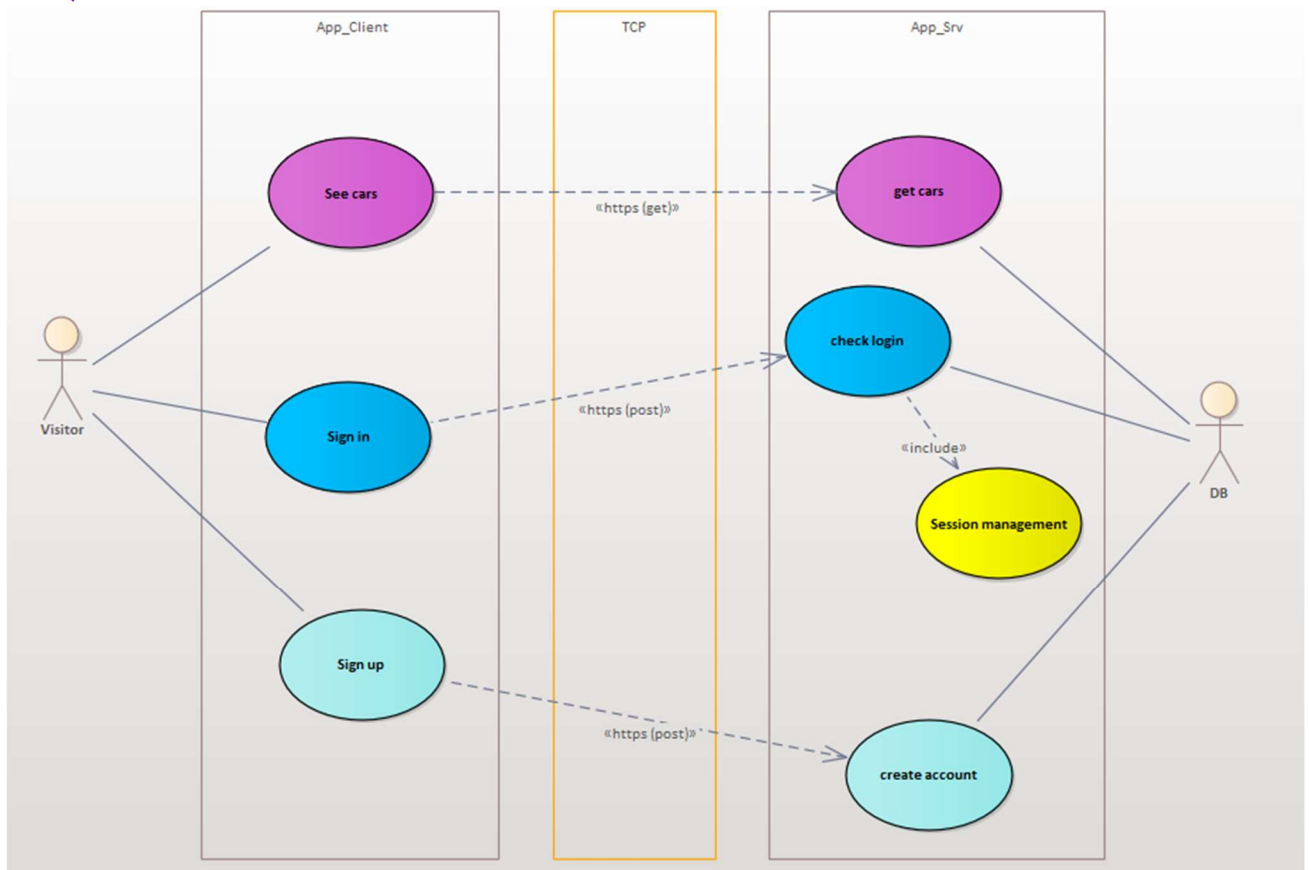
Les utilisateurs pourront rejoindre un véhicule ou en ajouter un. Ils pourront aussi se déconnecter et modifier leur profil.

### c) Les utilisateurs ayant ajouté un véhicule

Les utilisateurs ayant ajouté un véhicule ne pourront plus voir les autres véhicules, ils pourront uniquement modifier le leur.

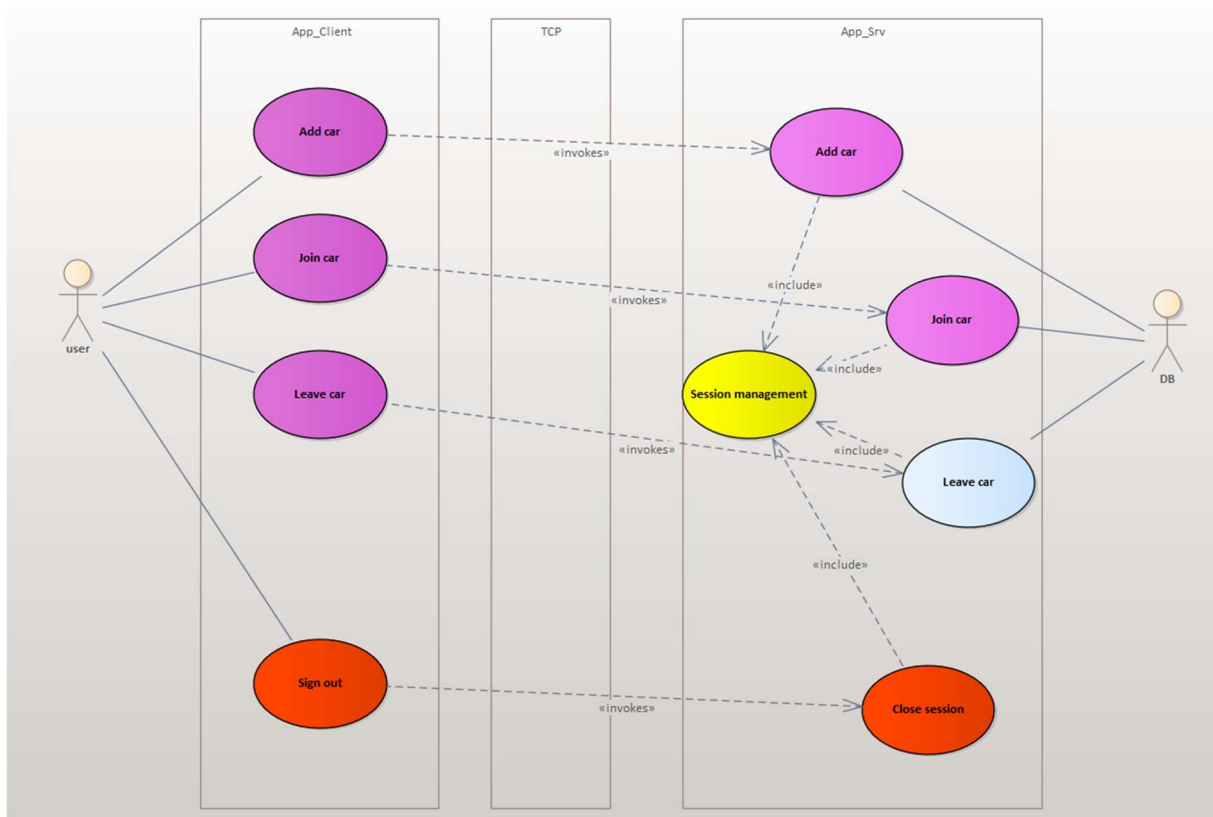
## Uses Case

### a) Use case de la vue des visiteurs du site :



Un visiteur peut voir les voitures de la soirée, il peut créer un compte ou se connecter.

b) Use case de la vue des utilisateurs du site :

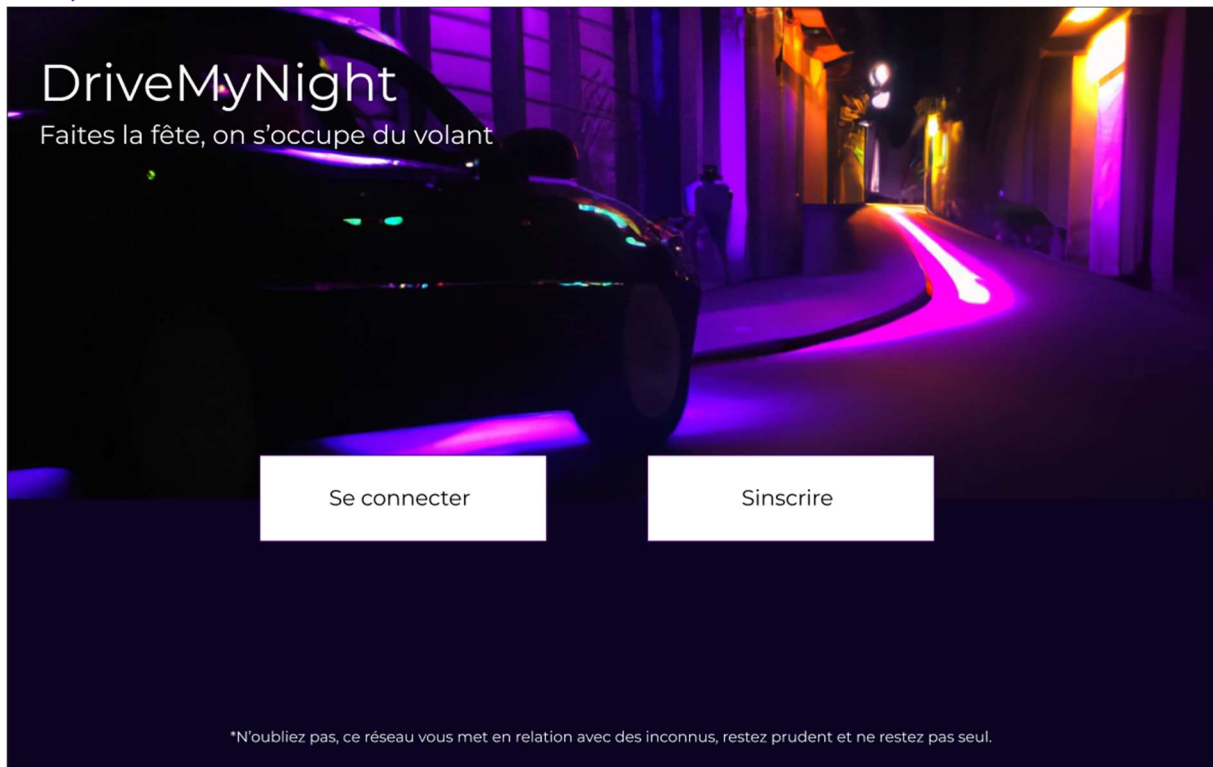


L'utilisateur peut ajouter une voiture, il peut rejoindre une voiture et quitter une voiture, il peut aussi se déconnecter de l'application.



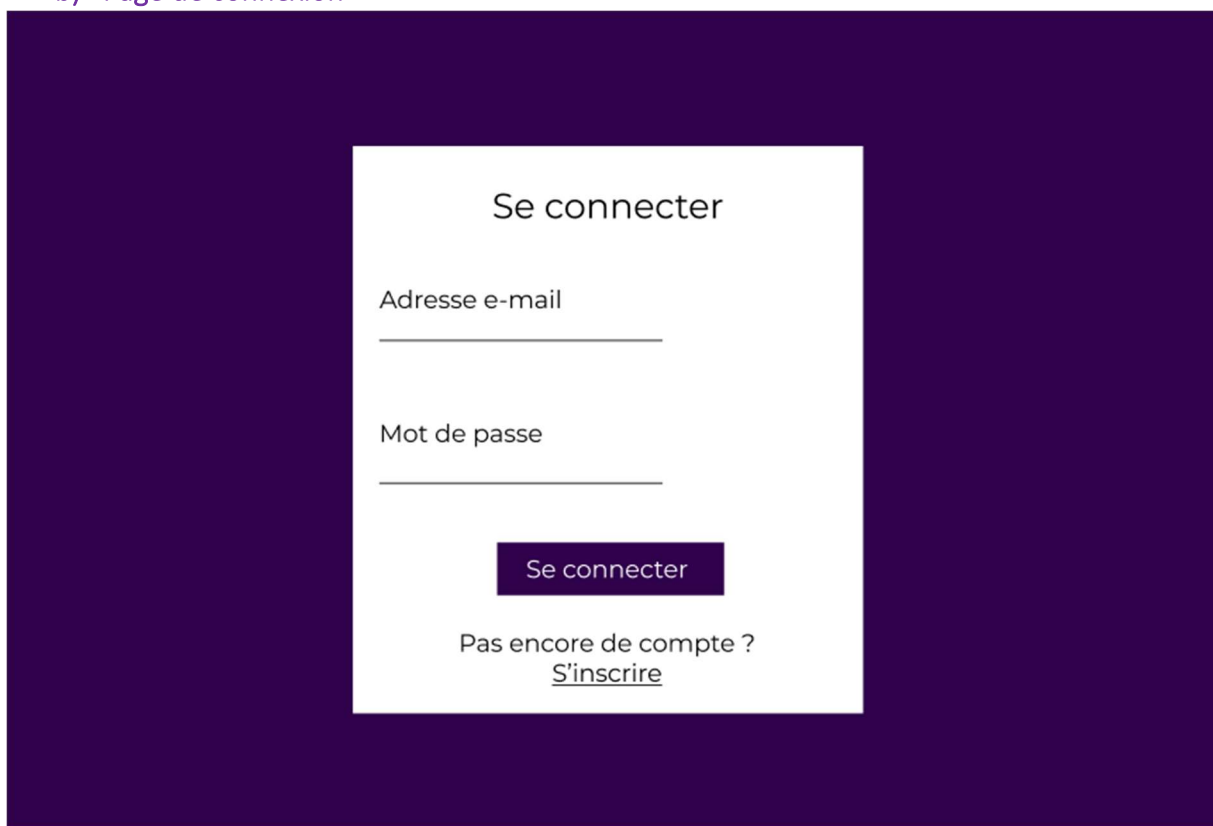
## Maquettes

## a) Visiteur du site web



Lorsqu'un utilisateur arrivera sur ce site web, il y aura une vue attractive avec une ambiance reflétant le thème du site.

## b) Page de connexion



La page de connexion sera simple ne demandant que l'adresse e-mail et le mot de passe de l'utilisateur.

c) Inscription / création de compte



**S'inscrire**

Adresse e-mail

Nom

Prénom

Mot de passe

Confirmer le mot de passe

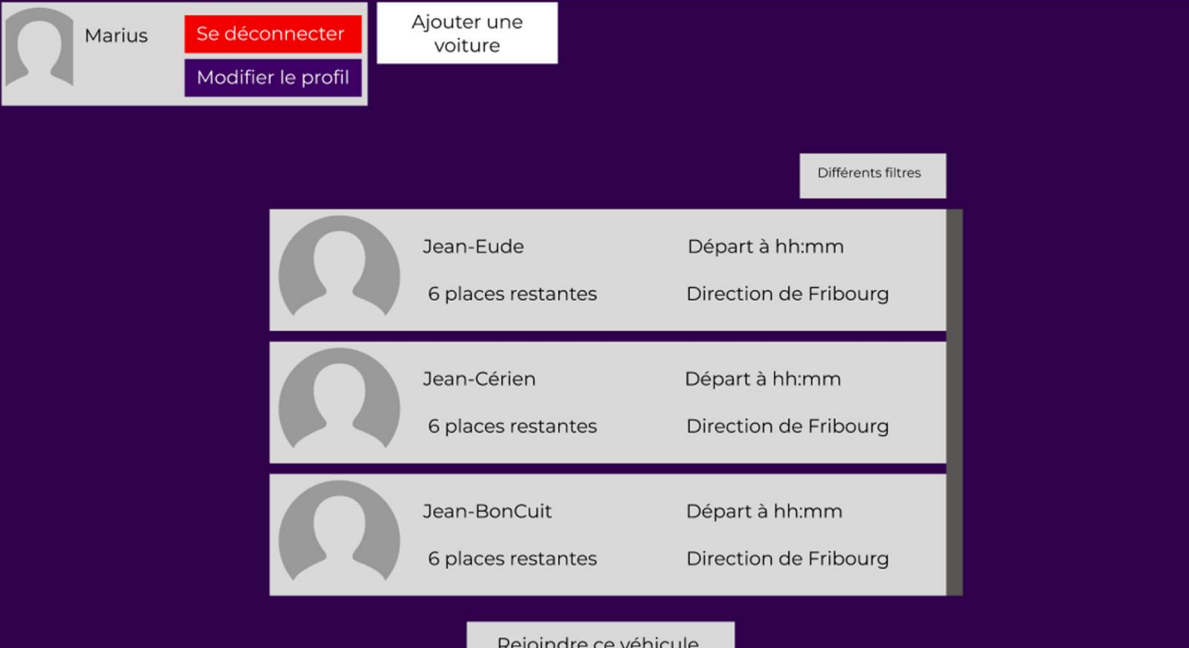
 Ajouter une photo de profil


**S'inscrire**

Déjà un compte ?  
[Se connecter](#)


La page d'inscription demande différentes entrées et la possibilité d'ajouter une photo de profil.

d) Page utilisateur



 Marius **Se déconnecter** Ajouter une voiture  
**Modifier le profil**

Différents filtres

	Jean-Eude 6 places restantes	Départ à hh:mm Direction de Fribourg
	Jean-Cérien 6 places restantes	Départ à hh:mm Direction de Fribourg
	Jean-BonCuit 6 places restantes	Départ à hh:mm Direction de Fribourg

**Rejoindre ce véhicule**



La page d'utilisateur permettra de rejoindre des véhicules, de se déconnecter, de modifier son profil ou encore d'ajouter une voiture.

e) Ajout d'un véhicule



Ajouter une voiture

Départ prévu à

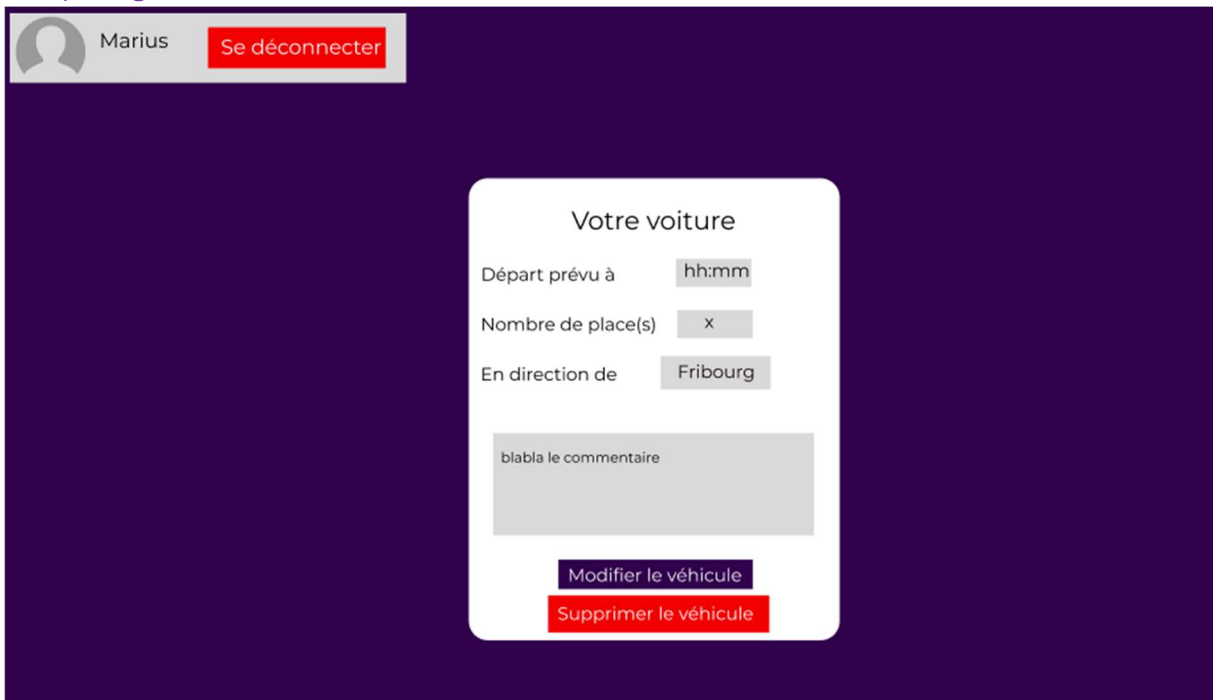
Nombre de place(s)

En direction de

Ajouter des précisions sur le lieu et l'heure de rendez-vous

Pour ajouter un véhicule nous devront sélectionner une heure de départ (à la demi-heure prêt), le nombre de places disponibles (de 1 à 9) et la direction. L'utilisateur aura l'obligance de rentrer un commentaire précisant le lieu de rencontre pour ses passagers afin d'éviter de donner des données personnelles tel que des numéros de téléphone.

f) Page créateur de véhicule



Marius

Votre voiture

Départ prévu à

Nombre de place(s)

En direction de

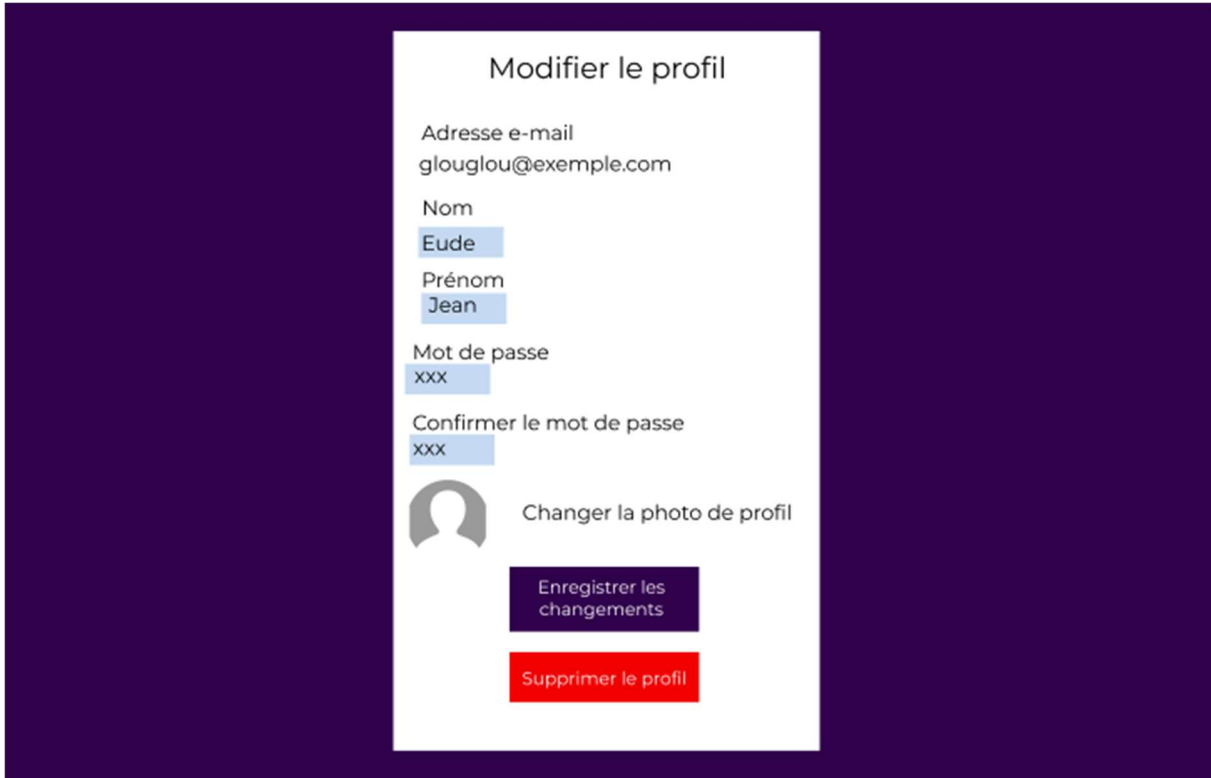
blabla le commentaire

L'utilisateur qui aura ajouté une voiture dans le site, pourra la modifier et la supprimer. Il n'aura plus possibilité de voir les autres véhicules et donc d'en rejoindre.





## g) Page modification de profil



Modifier le profil


Adresse e-mail  
glouglou@exemple.com

Nom  
Eude

Prénom  
Jean

Mot de passe  
xxx

Confirmer le mot de passe  
xxx

 Changer la photo de profil

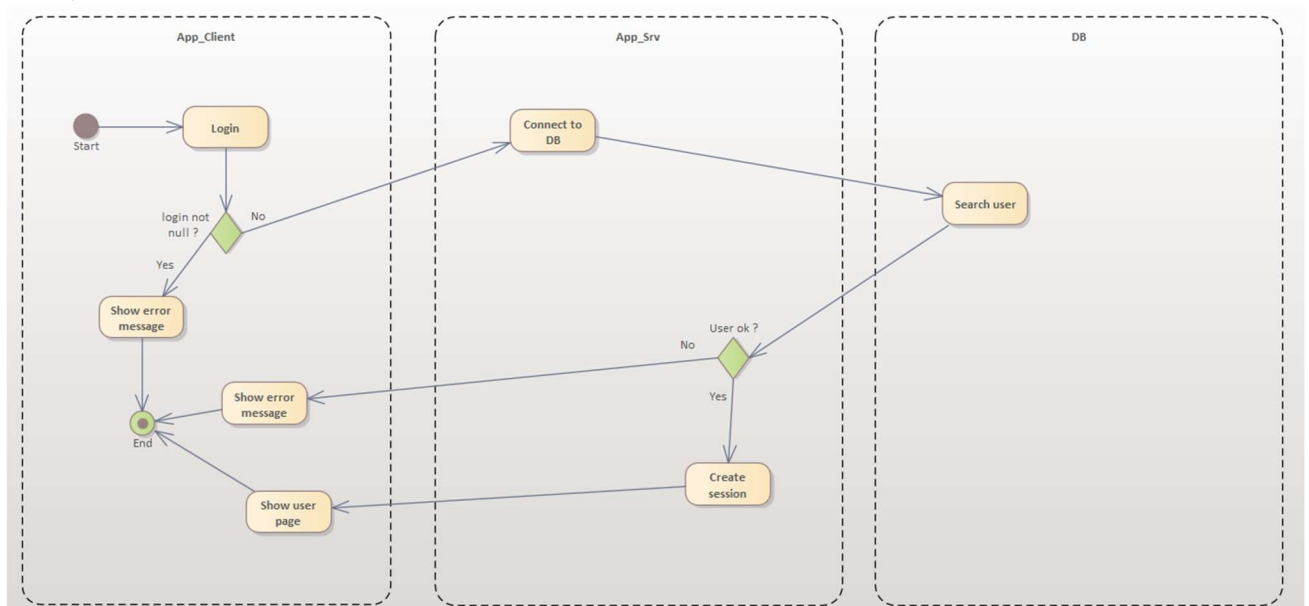
Enregistrer les changements

Supprimer le profil

Les utilisateurs pourront tout modifier sauf l'adresse e-mail, ils auront la possibilité de supprimer leur profil.

## Diagramme d'activité

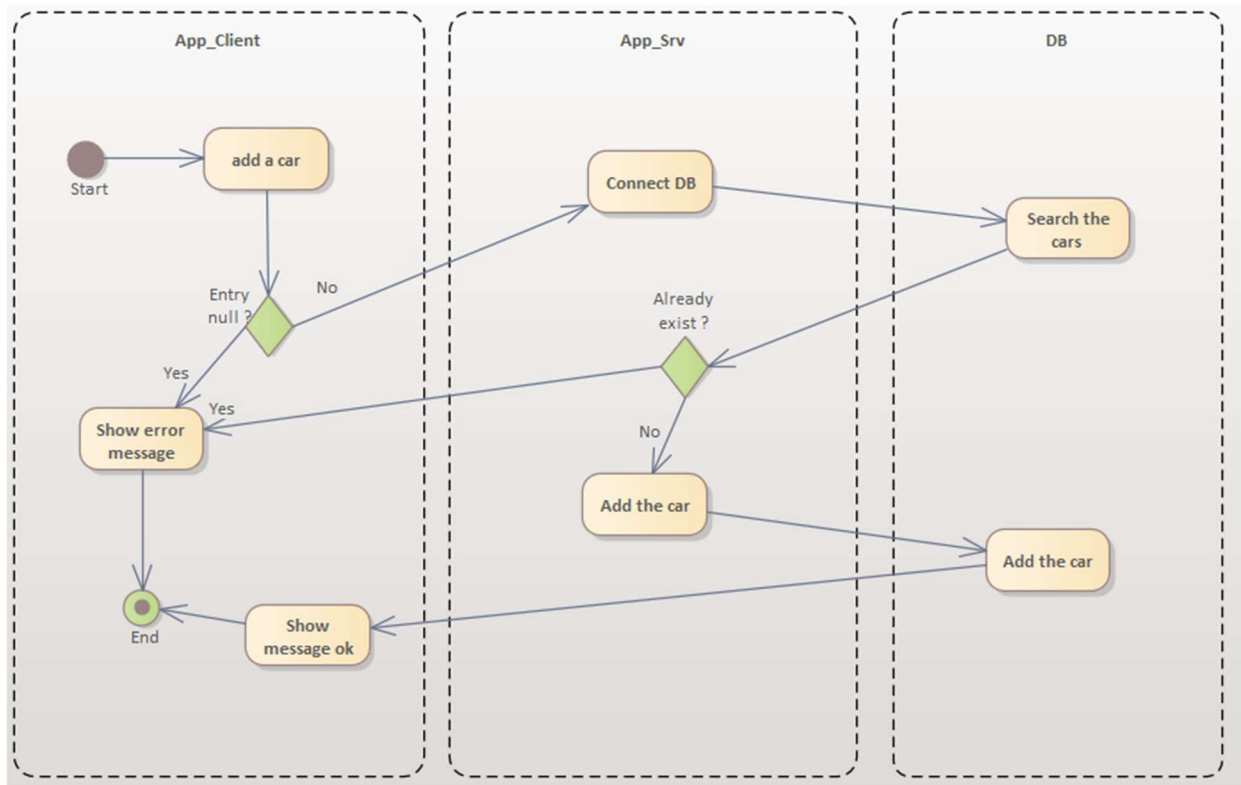
## a) Activité connexion



Ce diagramme permet de visualiser les actions de l'utilisateur pour se connecter à son compte.



## b) Activité ajouter une voiture

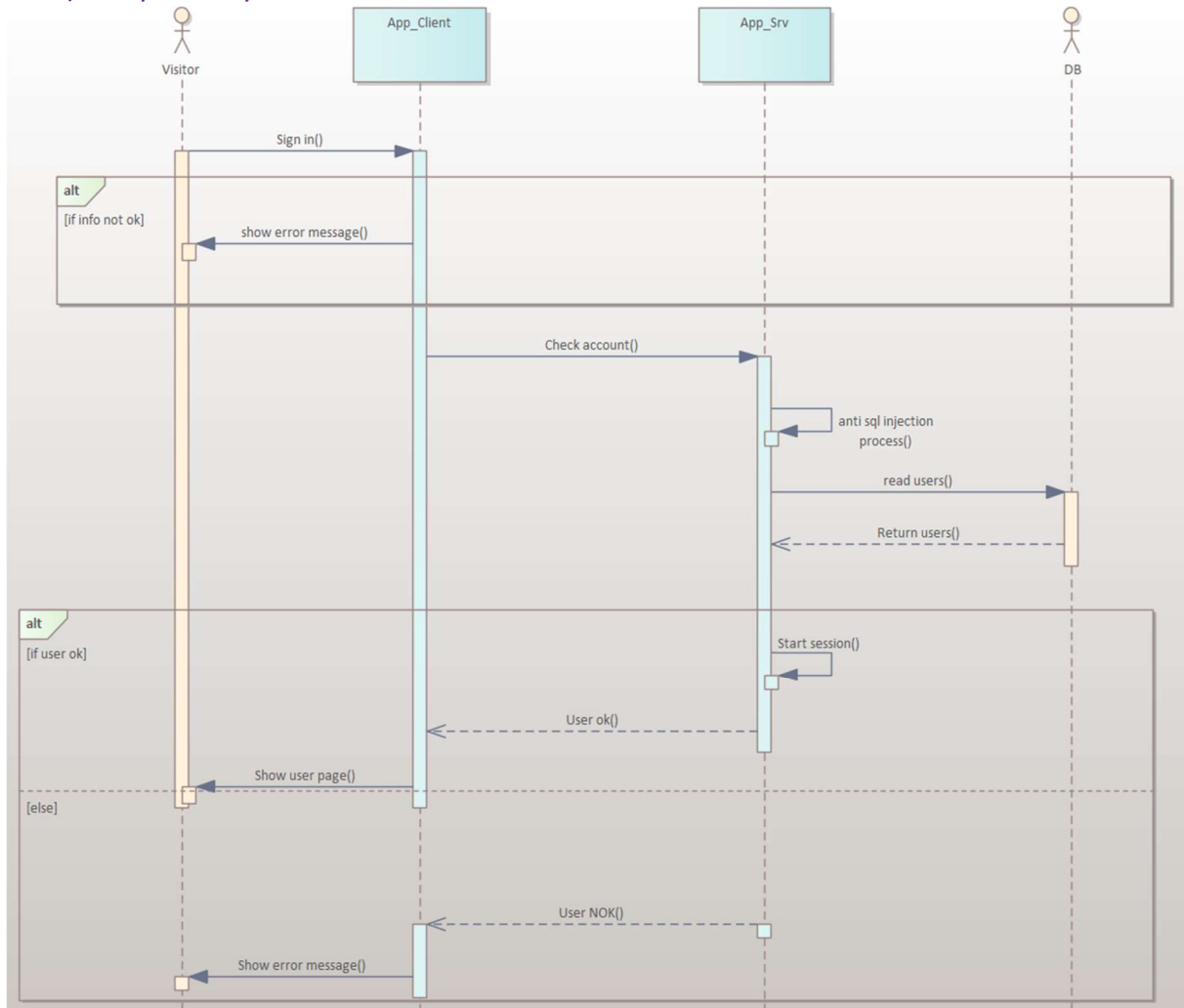


Ce diagramme permet de visualiser les activités de l'ajout d'une voiture dans la base de données.



## Diagramme de séquence systèmes

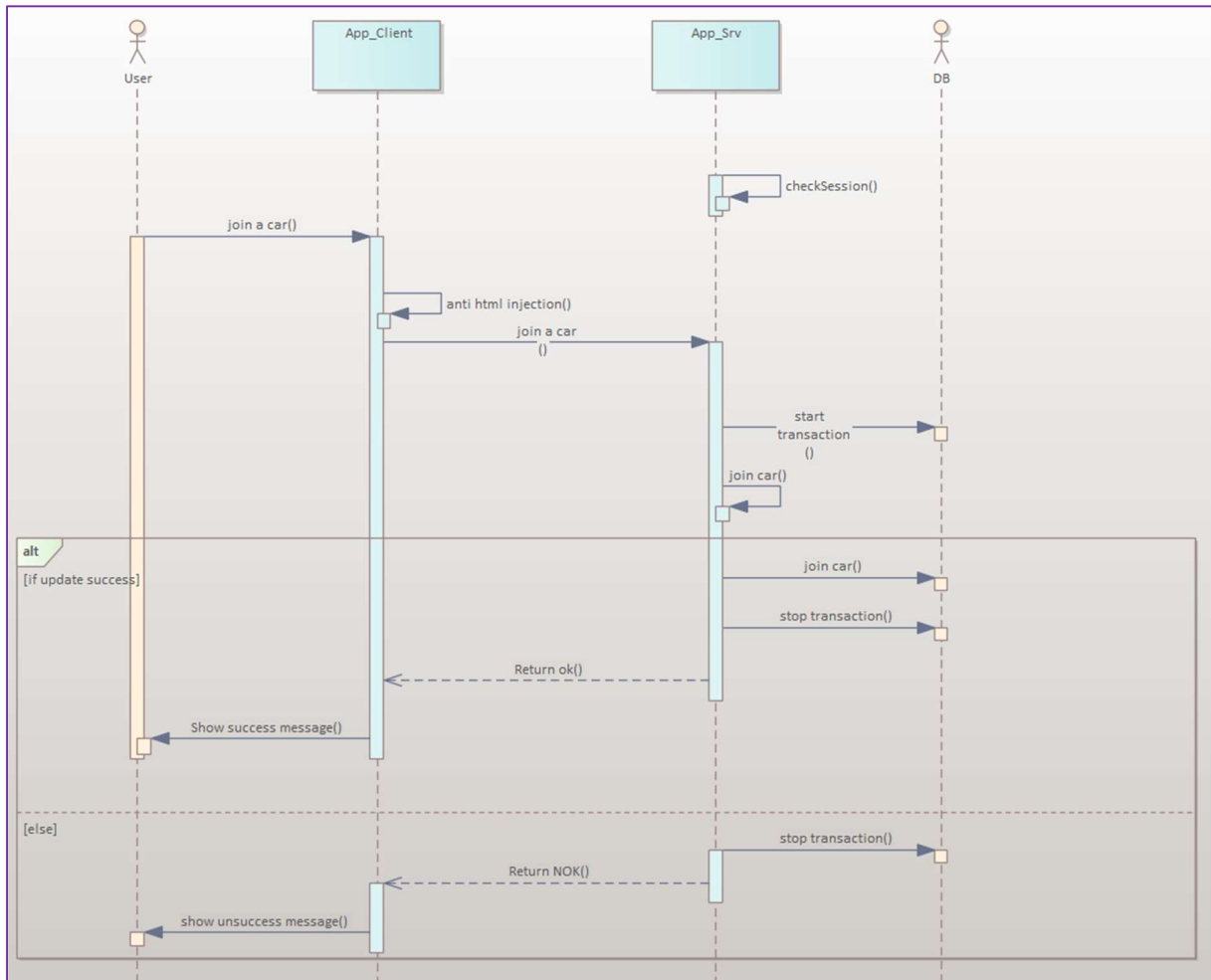
## c) Séquence système se connecter :



Ce diagramme représente la connexion d'un utilisateur avec les étapes précises du déroulement de cette requête avec la gestion de certaines erreurs.



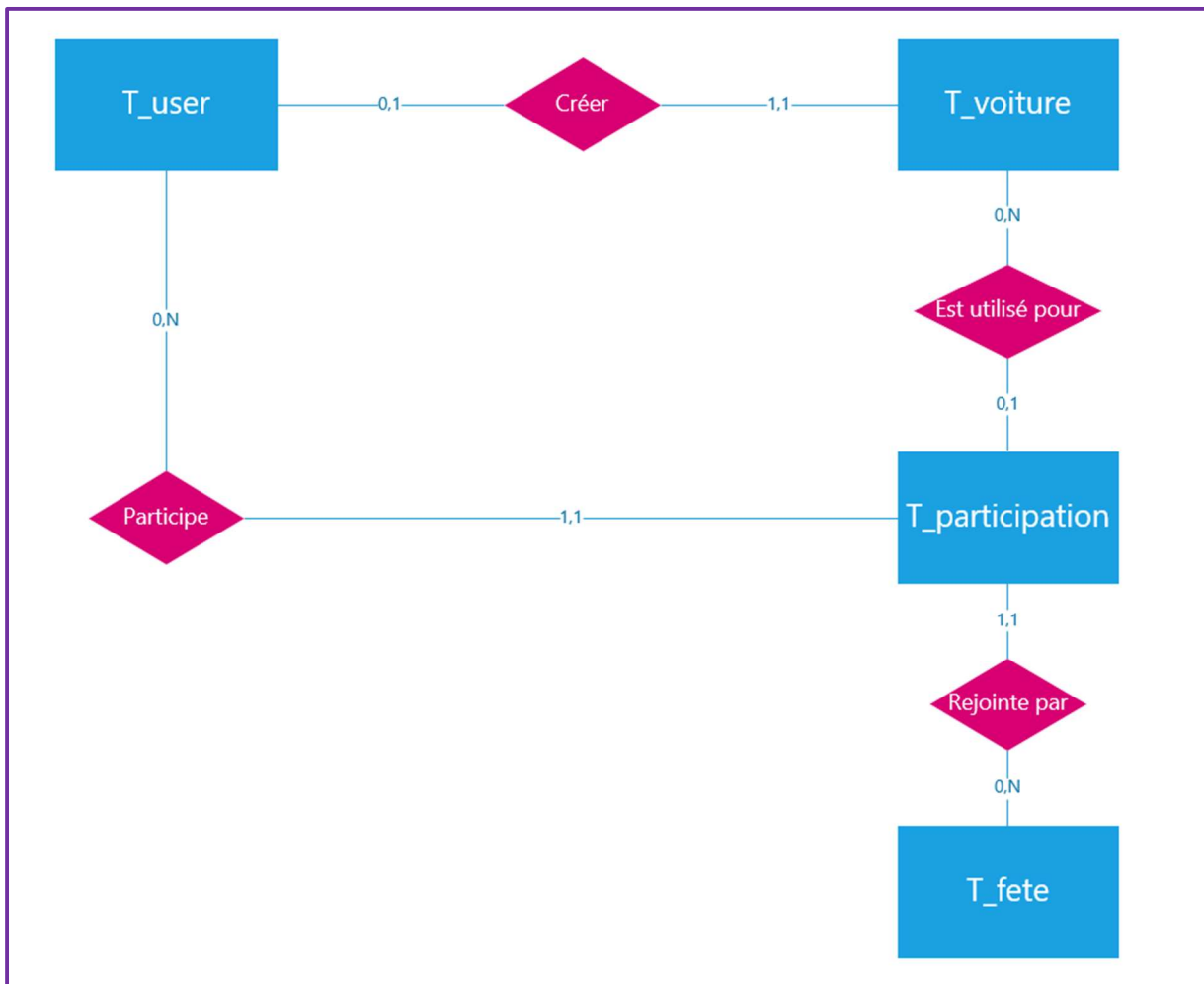
## d) Séquence système rejoindre une voiture



Ce diagramme permet de visualiser le processus pour rejoindre une voiture en gérant différentes erreurs possibles.



## Schémas ER



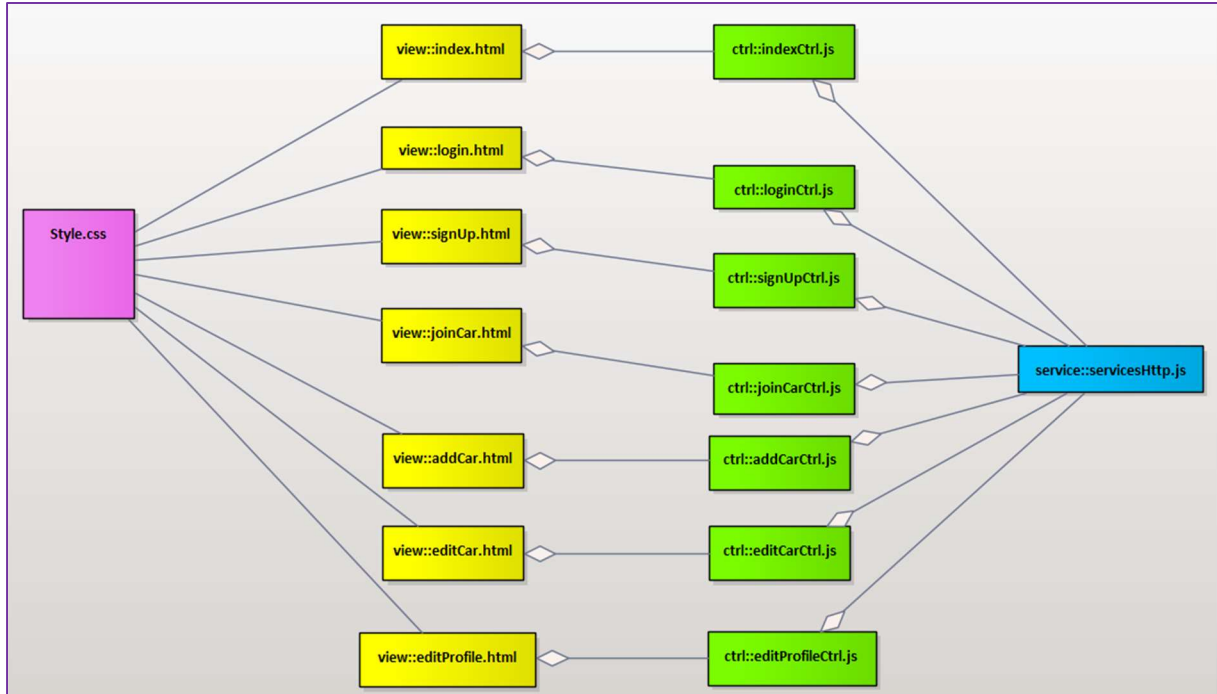
Dans la base de données, les utilisateurs pourront créer une voiture et la réutiliser dans les prochaines soirées, il y'a une table t\_participation qui est prévu pour avoir des traces de tout ce qui se passe dans chaque fête (qui a participé à quelle fête et a ramené ou a été ramené par qui et quel véhicule).



# Conception

## Diagrammes de classe

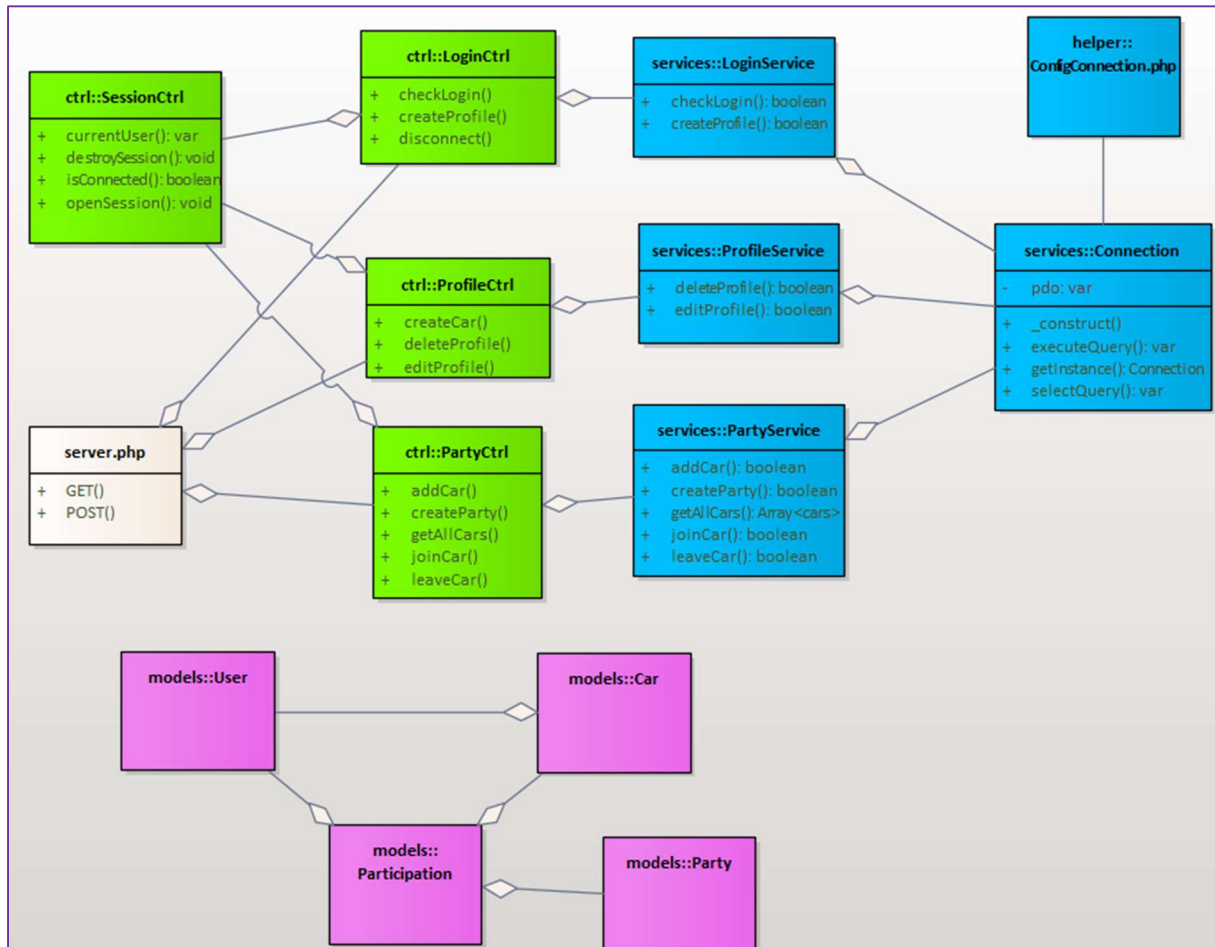
### a) Client



Ce diagramme de classe représente toutes les vues du site ainsi que leur contrôleur et **servicesHttp.js** gère toutes les requêtes vers internet et vers la partie serveur.



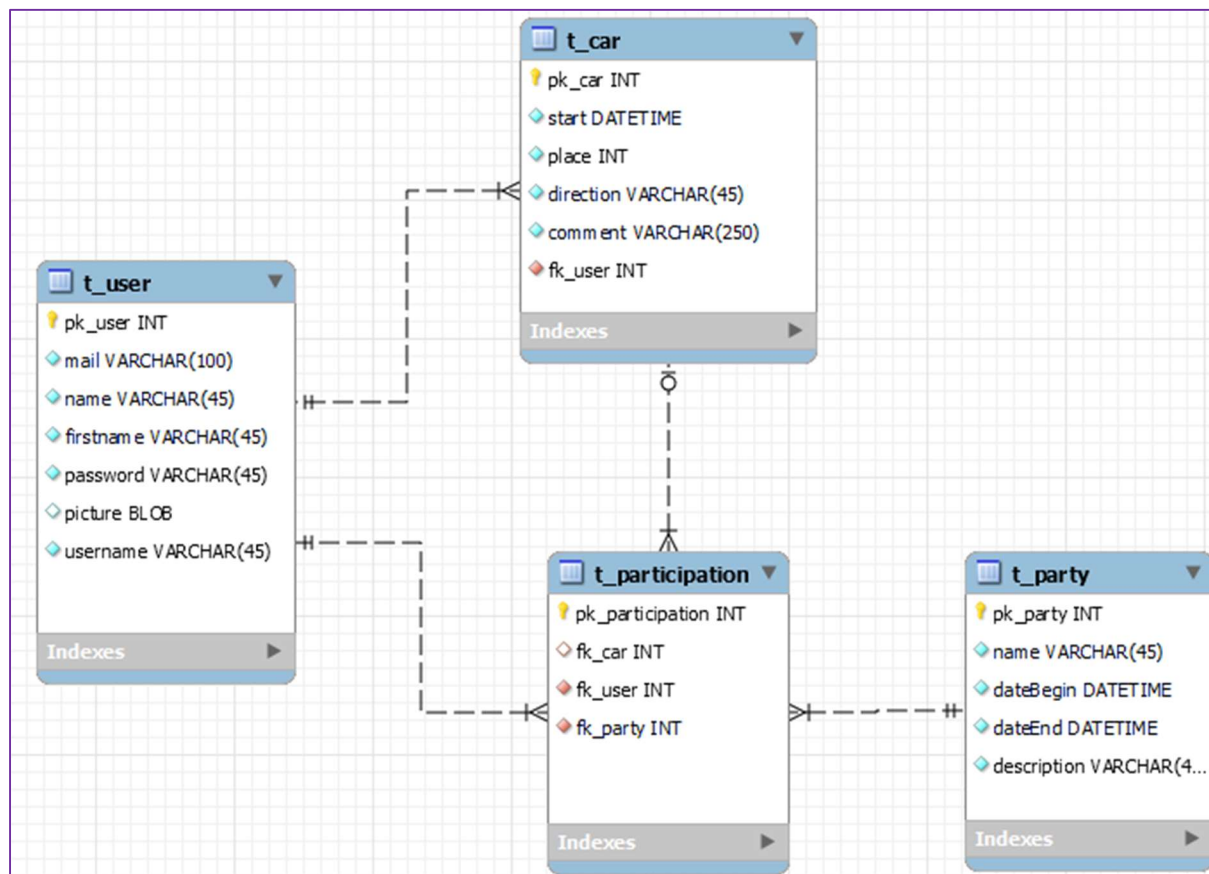
## b) Serveur



Ce diagramme de classe représente les classes côté serveur pour le fonctionnement de l'application.



## Schéma relationnel



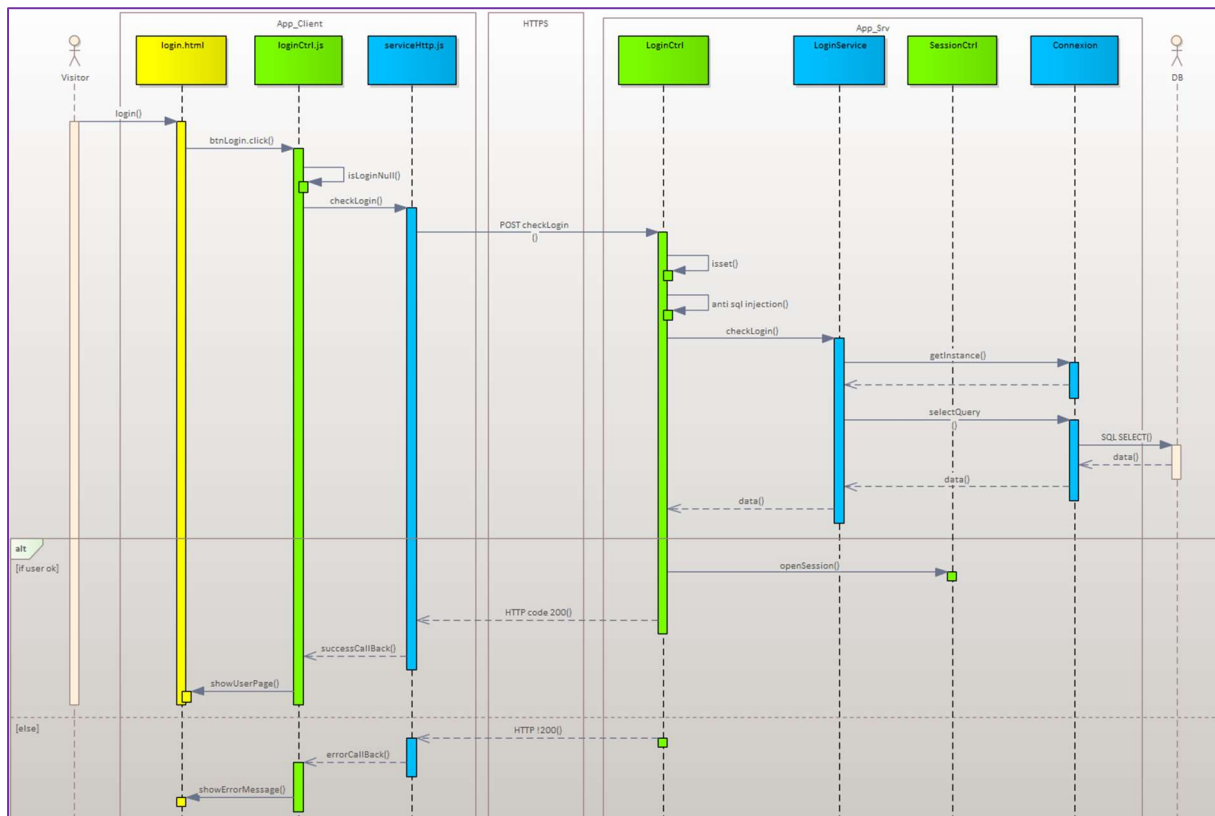
Le schéma respect les attentes du programmes et le schéma ER.





## Diagramme séquence interactions

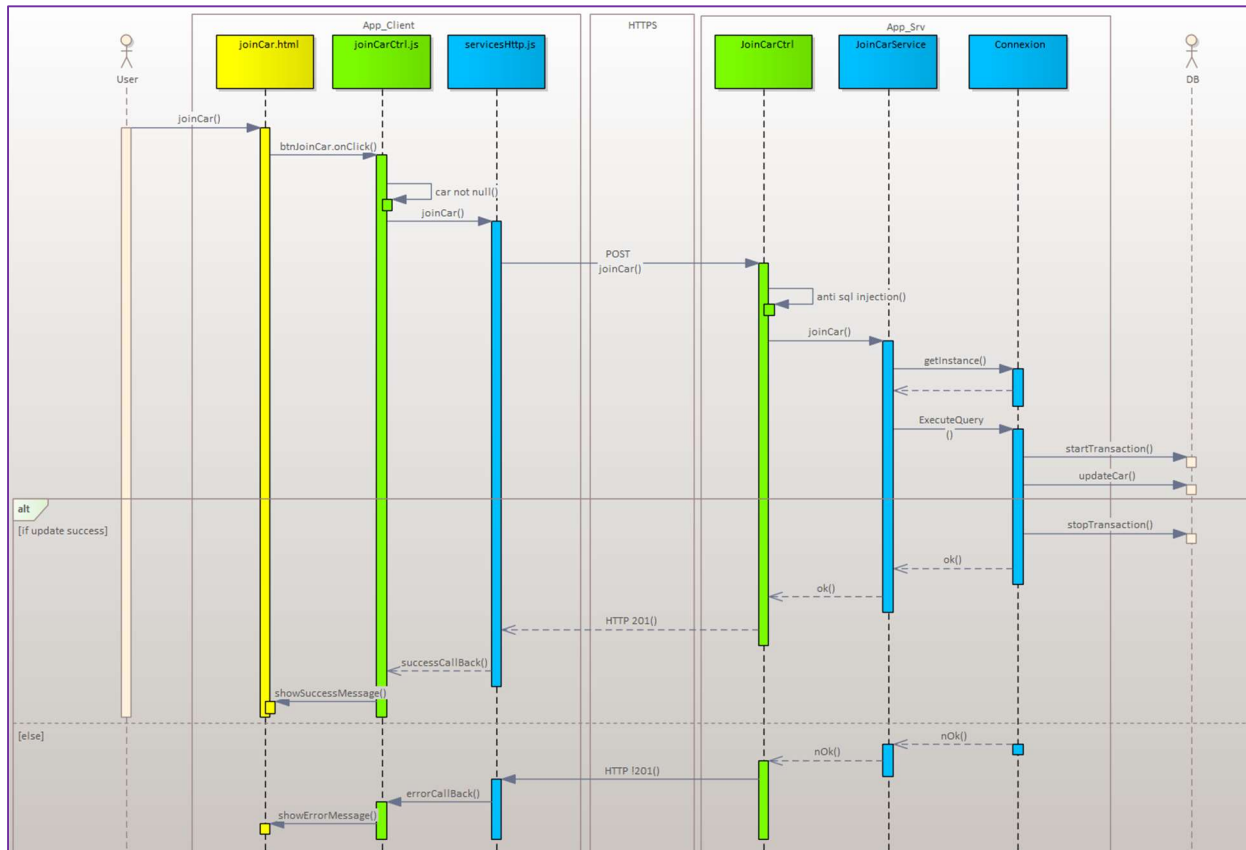
## a) Login



Ce diagramme de séquence représente toutes les actions nécessaires à la tentative de connexion d'une personne en montrant en détail tous les appels de méthodes à travers les différentes classes du côté client et serveur.



## b) Rejoindre une voiture



Ce diagramme de séquence représente toutes les actions nécessaires dès qu'un utilisateur tente de rejoindre une voiture en montrant en détail tous les appels de méthodes à travers les différentes classes du côté client et serveur.

## Conceptions des tests

Test No	Description	Condition(s) préalable(s)	Action	Résultat attendu
1	Connexion d'un utilisateur	Avoir cliqué sur le bouton « Se connecter »	Entrer une adresse mail et un mot de passe et se connecter	Si l'utilisateur est correct alors charger la page suivante, si il est incorrecte afficher un message d'erreur
2	Inscription d'un visiteur	Avoir cliqué sur le bouton « S'inscrire »	Remplir le formulaire puis s'inscrire	Si toutes les données sont correctes affiche un message de succès, sinon d'erreur
3	Rejoindre une voiture	S'être connecté	Sélectionner une voiture et cliquer « rejoindre »	Si l'utilisateur rejoint la voiture alors message de succès, sinon d'erreur
4	Se déconnecter de la plateforme	Être connecté	Cliquer sur le bouton « se déconnecter »	L'utilisateur est déconnecté et il est ramené à la première page
5	Filtrer les véhicules	Être connecté	Utiliser les filtres de véhicules et les appliquer	La voitures sont filtrées en fonction du filtre
6	Ajouter une voiture	Être connecté et	Remplir le formulaire et ajouter la voiture	Si le formulaire est correctement rempli et que



		avoir cliqué « Ajouter une voiture »		le véhicule a été ajouté message de succès, sinon d'erreur
7	Modifier une voiture	Être connecté et avoir déjà créé une voiture	Remplir le formulaire et modifier la voiture	Si le formulaire est correctement rempli et que le véhicule a été modifié message de succès, sinon d'erreur
8	Suppression d'une voiture	Être connecté et avoir déjà créé une voiture	Cliquer « supprimer le véhicule »	Si le véhicule a été supprimer alors message de succès et retour sur la page de l'utilisateur, sinon message d'erreur
9	Modification d'un utilisateur	Être connecté	Cliquer « modifier le profil », remplir le formulaire et enregistrer les changements	Si le formulaire est correctement rempli et que le profil a été modifié alors message de succès, sinon d'erreur
10	Suppression d'un profil	Être connecté	Cliquer « Supprimer le profil » dans la page de modification de profil	Si le profil a été supprimé correctement alors message de succès, sinon d'erreur



# Implémentation

## Descente de code

### a) Login

LoginCtrl.js

D'abord les infos sont récupérées du formulaire puis envoyé à la méthode du serviceHttp

```
checkLogin() {
    const email = this.form.elements["mail"].value;
    const password = this.form.elements["password"].value;
    const http = new ServiceHttp();
    http.checkLogin(email,password, this.successLogin, this.errorLogin);
}
```

Ensuite le serviceHttp fait une requête ajax vers le serveur pour soumettre les infos et demandés la connexion.

```
checkLogin(mail, password, successCallback, errorCallback) {
    this.disconnect();
    $.ajax({
        type: "POST",
        contentType: "application/x-www-form-urlencoded",
        url: this.URL,
        data: "action=checkLogin&mail=" + mail + "&password=" + password,
        xhrFields: {
            withCredentials: true
        },
        success: successCallback,
        error: errorCallback
    });
}
```

Ensuite le serveur appel la méthode du loginCtrl

```
if ($_POST['action'] == "checkLogin") {
    $loginCtrl->checkLogin($_POST['mail'], $_POST['password']);
}
```

Ensuite le loginCtrl hash le mot de passe et donne toutes les infos sous la forme d'un model user au servic afin de retourner les bons codes http en fonction de la réponse du service.

```
public function checkLogin($mail, $password)
{
    if (!empty($mail) && !empty($password)) {

        $user = new User(null);
        $user->setMail($mail);
        $user->setPassword($password);
        $result = $this->loginService->checkLogin($user);
        if ($result) {
            http_response_code(200);
        }
    }
}
```



```

        //Ajoute le mail dans la session
        $this->session->set('mail', $user->getMail());
        $this->session->set('pkUser', $result);
    } else {
        http_response_code(401);
    }
} else {
    http_response_code(400);
}
}

```

Le service, lui, test avec la base de données si les infos sont correctes et il retourne la pk de l'utilisateur afin de la stocker en session ou alors un false.

### b) Rejoindre une voiture

UserPageCtrl.js

Cette fonctionnalité commence sur le contrôleur javascript de la page d'utilisateur. Sur cette page, les voitures à rejoindre sont des div dynamique et sur chacun d'eux il y a un écouteur qui écoute le click. Si une de ces div est cliquée il y aura alors une pop up de confirmation.

```

bloc.addEventListener('click', function () {
    document.getElementById('joinPopup').style.display = 'block';
// Montre la popup
    document.getElementById("comment").innerText = comment;

document.getElementById('confirmJoin').dataset.username=username;

});

```

Dans le cas où nous appuyons sur confirmer, la méthode joinCar() du serviceHttp sera appelée avec en paramètre l'username de la voiture à rejoindre et le success et error callback.

```

document.getElementById('confirmJoin').addEventListener('click', function () {
    // L'utilisateur confirme vouloir rejoindre le véhicule
    const usernameToJoin = this.dataset.username;
    self.http.joinCar(usernameToJoin, self.joinCarSuccess
, self.error);
    document.getElementById('joinPopup').style.display = 'none'; //
Cache la popup
});

    document.getElementById('cancelJoin').addEventListener('click',
function () {
    document.getElementById('joinPopup').style.display = 'none'; //
Cache la popup sans autres actions
});

```

ServiceHttp.js



Cette méthode établit une requête ajax post vers le serveur lui donnant comme paramètre le nom d'utilisateur de la voiture a rejoindre.

```
joinCar(usernameToJoin, successCallback, errorCallback) {
    $.ajax({
        type: "POST",
        contentType: "application/x-www-form-urlencoded",
        url: this.URL,
        data: 'action=joinCar&username=' + usernameToJoin,
        xhrFields: {
            withCredentials: true
        },
        success: successCallback,
        error: errorCallback
    });
}
```

Voici les success et errorCallback

```
//Si l'utilisateur a réussi a rejoindre une voiture
joinCarSuccess(){
    alert('Voiture rejoint !');
    window.location.reload();
}
error(xhrFields) {
    switch (xhrFields.status) {
        case 401: alert("Votre session a pris fin, veuillez vous
reconnecter"); window.location.href = 'login.html';
            break;
        case 500: alert('Problème serveur');
            break;
        case 404: alert("Vous n'êtes dans aucune party");
            break;
        case 409 : alert("Conflit, il est possible que vous soyez déjà
dans une voiture ou que la voiture dont vous essayez de rejoindre est
pleine");
            break;
    }
}
```

Server.php

L'écouteur de la requête

```
if ($_POST['action'] == 'joinCar') {

    $usernameToJoin = $_POST['username'];
    $partyCtrl->joinCar($usernameToJoin);
}
```

Le contrôleur appelle le service et d'après son retour il retourne au client les bons codes d'erreurs ou de succès

```
public function joinCar($usernameToJoin)
```



```

{
    if (!empty($usernameToJoin)) {
        if ($this->session->has('mail')) {
            //Si il n'est pas déjà dans une voiture
            $pkUser = $this->session->get('pkUser');
            $result = $this->partyService->joinCar($usernameToJoin,
$pkUser);

            if ($result == 'ok') {
                //Ajout réussi
                http_response_code(200);
            } elseif ($result == 'notSameParty') {
                //Pas dans la meme soirée
                http_response_code(403);
            } elseif ($result == 'carNotExist') {
                //La voiture a rejoindre n'existe pas
                http_response_code(404);
            } elseif ($result == 'alreadyInCar') {
                //L'utilisateur est déjà dans une voiture
                http_response_code(409);
            } elseif ($result == 'notInParty') {
                //Pas dans une soirée
                http_response_code(403);
            } else {
                //sinon si il n a pas pu etre bien réalisé
                http_response_code(500);
            }
        } else {
            //L'utilisateur n'est pas connecté
            http_response_code(401);
        }
    } else {
        //Le conducteur est vide
        http_response_code(400);
    }
}
}

```

Le service test tous les cas pour éviter les erreurs puis il crée une nouvelle participation avec la voiture, l'utilisateur et la fête.

```

public function joinCar($usernameToJoin, $pkJoiner)
{
    $return = false;
    $participationsOfJoiner = $this->connection->selectSingleQuery('SELECT
* FROM t_participation WHERE fk_user=?', [$pkJoiner]);
    $pkUserToJoin = $this->connection->selectSingleQuery('SELECT pk_user
FROM t_user WHERE username=?', [$usernameToJoin]);

    if ($participationsOfJoiner) {
        //Si le joiner est dans une soirée
    }
}

```



```

        if (!$this->connection->selectQuery('SELECT * FROM t_participation
WHERE fk_user=? AND fk_car IS NOT NULL', [$pkJoiner])) {
            //Le joiner n'est pas déjà dans une voiture

            if ($participationUserToJoin = $this->connection-
>selectSingleQuery('SELECT * FROM t_participation WHERE fk_user=? AND fk_car
IS NOT NULL', [$pkUserToJoin[0]])) {
                //La voiture que l'on veut rejoindre existe

                if ($participationsOfJoiner['fk_party'] ==
$participationUserToJoin['fk_party']) {
                    //Les deux utilisateurs sont dans la meme party
                    $nbUsersInCar = $this->connection-
>selectSingleQuery('SELECT COUNT(*) - 1 AS nbUsersInCar FROM t_participation
WHERE fk_car = ?', [$participationUserToJoin['fk_car']]));

                    $nbPlacesInCar = $this->connection-
>selectSingleQuery('SELECT place FROM t_car WHERE pk_car=?',
[$participationUserToJoin['fk_car']]));

                    $availableSeats = $nbPlacesInCar[0] -
$nbUsersInCar[0];

                    if ($availableSeats > 0) {

                        //Si il reste de la place alors créer une entrée
dans t_participation

                        if ($this->connection->executeQuery('INSERT INTO
t_participation (fk_car,fk_user,fk_party) VALUES (?,?,?)',
[$participationUserToJoin['fk_car'], $pkJoiner,
$participationUserToJoin['fk_party']]))) {

                            //Si l'ajout a bien pu être fait
                            $return = 'ok';
                        } else {
                            //Si il n a pas été fait correctement
                            $return = false;
                        }
                    }
                } else {
                    //Ils ne sont pas dans la meme party
                    $return = 'notSameParty';
                }
            } else {
                //La voiture que l'on veut rejoindre n'existe pas
                $return = 'carNotExist';
            }
        } else {
            //Il est déjà dans une voiture
            $return = 'alreadyInCar';
        }
    }
}

```





```

    }
  } else {
    //Il est pas dans une soirée
    $return = 'notInParty';
  }
  return $return;
}

```

### Problèmes rencontrés

Les problèmes les plus récurrent étaient des problèmes simples, jamais eu de problème très intense et qui dure sur le long terme lors de ce projet. La plupart ont vite été réglé et ont été causé par manque d'attention ou alors par manque de prévision.

### Tests fonctionnels

Test No	Description	Condition(s) préalable(s)	Action	Résultat attendu	Résultat obtenu
1	Connexion d'un utilisateur	Avoir cliqué sur le bouton « Se connecter »	Entrer une adresse mail et un mot de passe et se connecter	Si l'utilisateur est correct alors charger la page suivante, si il est incorrecte afficher un message d'erreur	L'utilisateur accède à la page d'utilisateur avec une popup si c'est bon et obtient un message d'erreur si non.
2	Inscription d'un visiteur	Avoir cliqué sur le bouton « S'inscrire »	Remplir le formulaire puis s'inscrire	Si toutes les données sont correctes affiche un message de succès, sinon d'erreur	L'utilisateur est redirigé vers la page de connexion avec un message de succès, sinon un message d'erreur.
3	Rejoindre une voiture	S'être connecté	Sélectionner une voiture et cliquer « rejoindre »	Si l'utilisateur rejoint la voiture alors message de succès, sinon d'erreur	Une popup avec un message de succès ou d'erreur apparaît.
4	Se déconnecter de la plateforme	Être connecté	Cliquer sur le bouton « se déconnecter »	L'utilisateur est déconnecté et il est ramené à la première page	L'utilisateur est redirigé vers la page de connexion.
5	Filtrer les véhicules	Être connecté	Utiliser les filtres de véhicules et les appliquer	La voitures sont filtrées en fonction du filtre	Pas en place.
6	Ajouter une voiture	Être connecté et avoir cliqué « Ajouter une voiture »	Remplir le formulaire et ajouter la voiture	Si le formulaire est correctement rempli et que le véhicule a été ajouté message de succès, sinon d'erreur	Si la voiture a été ajouté il y a un message de succès et sinon d'erreur
7	Modifier une voiture	Être connecté et avoir déjà créé une voiture	Remplir le formulaire et modifier la voiture	Si le formulaire est correctement rempli et que le véhicule a été modifié message de succès, sinon d'erreur	Si la voiture a été modifié alors la page se reload et un message de succès apparaît, sinon d'erreur.
8	Suppression d'une voiture	Être connecté et avoir déjà créé une voiture	Cliquer « supprimer le véhicule »	Si le véhicule a été supprimé alors message de succès et retour sur la page de l'utilisateur, sinon message d'erreur	Message de succès ou d'erreur.



9	Modification d'un utilisateur	Être connecté	Cliquer « modifier le profil », remplir le formulaire et enregistrer les changements	Si le formulaire est correctement rempli et que le profil a été modifié alors message de succès, sinon d'erreur	Message de succès et reload de la page sinon d'erreur
10	Suppression d'un profil	Être connecté	Cliquer « Supprimer le profil » dans la page de modification de profil	Si le profil a été supprimé correctement alors message de succès, sinon d'erreur	Redirection à la première page et message de succès sinon d'erreur.

### Hébergement

Le site est hébergé sous <https://colliardt.emf-informatique.ch/151/client/views/index.html> mais le serveur ne réagit pas comme avec docker.

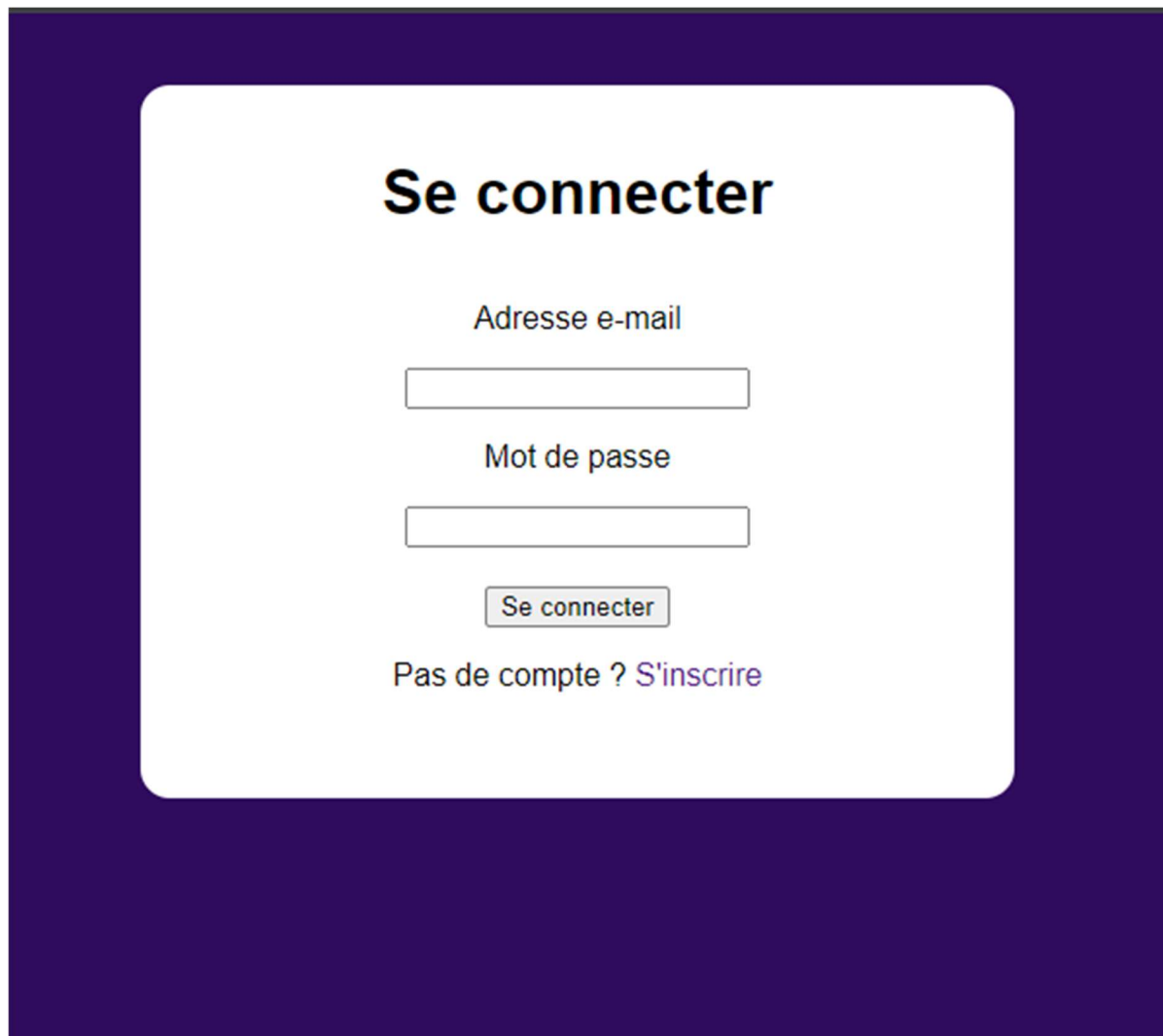


# Synthèse

---

Présentation réalisation

Page de connexion



**Se connecter**

Adresse e-mail

Mot de passe

Pas de compte ? [S'inscrire](#)

Page d'inscription



## S'inscrire

Adresse e-mail


Pseudo / nom d'utilisateur

Nom

Prénom


Mot de passe

Confirmer le mot de passe

 [Sélectionner une photo](#)

Déjà inscrit ? [se connecter](#)

Page d'utilisateur


 Thibaut


Modifier son profil


Se déconnecter

Ajouter ma voiture à la soirée

Quitter la voiture

**chloepetit**2 places restantes  
Direction: WestDépart: 21:00

**mariedurand**3 places restantes  
Direction: SouthDépart: 19:00

**leolemaire**7 places restantes  
Direction: veveyDépart: 19:10



Modifier le profil

## Modifier mon profil

Pseudo / nom d'utilisateur

Nom

Prénom

Changer le mot de passe

Confirmer le nouveau mot de passe


[Ajouter une photo](#)

Modifier / Ajouter sa voiture



## Modifier ma voiture

Départ

28.03.2024 17:13 

Nombre de place(s) sans compter le conducteur

1 

En direction de

ggdfgdf

Infos supplémentaires

gdfgfdgdfg

Modifier ma voiture

Retirer ma voiture de la soirée

Supprimer ma voiture

### Les utilisateurs présents dans votre voiture :

Il n'y a personne par ici !



## Ajouter une voiture

Départ

Nombre de place(s) sans compter le conducteur

En direction de

Infos supplémentaires

Ajoutez des informations supplémentaires pour l'heure de rendez-vous et l'emplacement précis de votre voiture

### Différences entre planning et réalisation

Les quelques différences sont qu'il était prévu d'appuyer sur un bouton pour rejoindre une voiture mais que finalement il suffit de cliquer dessus, les filtres n'ont pas été ajoutés comme prévu, l'utilisateur qu'il ait ajouté ou rejoint une voiture aura toujours accès à la page d'utilisateur alors que ce n'était pas prévu comme cela et les conducteurs peuvent voir qui est dans leur voiture ce qui n'était pas le cas.

### Conclusion

Pour conclure ce projet m'a pris énormément de temps, j'ai sûrement été un peu ambitieux mais cela me fait une bonne base pour continuer ce projet qui est un projet personnel hors scolaire.

J'en ai énormément appris sur la programmation web dans ce module et j'en garde une bonne expérience malgré l'énergie considérable qui a été fournie.

