



Rapport

LPSIL IDSE - 2017/2018 - IUT Nice - Sophia-Antipolis

Marvin RAMEIX - Yanice TRECH - Mohamed EL KARMOURDI - Romain LEMBO

Sommaire

Objectifs	Page 3
Déroulements des 3 Sprints	Page 4
Problèmes rencontrés	Page 5
Accès aux outils d'intégration continue et qualité de code	Page 7
MCD initial et final	Page 8
UML	Page 9

Objectifs

Le principal objectif de l'application CovoGreen est de fournir aux étudiants de l'université de Nice - Sophia-Antipolis la possibilité de créer ou de participer à des offres de covoiturage entre eux.

Dans un premier temps, ce projet va permettre d'une manière ou d'une autre de diminuer la pollution et ainsi participer à la sauvegarde de notre planète. De plus, l'utilisation d'une application comme CovoGreen par les étudiants ou toute autres personnes va permettre de diminuer le trafic routier.

Puis dans un second temps, cette application va permettre de faire des rencontres et de consolider les liens entre étudiants.

Donc en quelques points cela donne :

- Entraider les étudiants de l'université de Nice – Sophia-Antipolis de voyager ensemble dans le même véhicule.
- Participer à l'écologie.
- Participer à la réduction du trafic routier.
- Renforcer les liens entre étudiants.

Déroulements des 3 Sprints

Chaque fin de Sprint donné lieu à des mises au point entre coéquipier pour connaître l'état d'avances des tâches attribuées à chacun, et via Git, une fusion de nos travaux tout en corrigeant les conflits de codes éventuels.

Sprint 1:

Date début : 09/01/2018

Date fin : 10/02/2018

Récapitulatif :

Essentiellement basé sur l'architecture de la base de données et sa mise en place via Sequelize, sur les User Stories concernant l'inscription d'un utilisateur, la connexion à l'application, la déconnexion à l'application, la recherche d'un trajet, puis la création de trajet.

Sprint 2:

Date début : 10/02/2018

Date fin : 06/03/2018

Récapitulatif :

Essentiellement basé sur les User Stories concernant le contact en utilisateurs (Chat), l'administration des comptes utilisateurs (révocation, privilège), et la gestion d'un compte personnel (modification et suppression).

Sprint 3 :

Date début : 06/03/2018

Date fin : 27/03/2018

Récapitulatif :

Essentiellement basé sur les User Stories concernant l'affichage et suppression des offres de covoiturage, l'évaluation de la prestation d'un conducteur (calcul moyenne d'un conducteur, ajout d'une note par un passager, ajout de commentaire), et l'ajout d'une fonctionnalité permettant à un utilisateur connecté de voir l'historique de ses trajets (possibilité depuis cet historique de voir pour chaque trajet d'accéder au chat, et au détail de l'offre), une page utilisateur afin de voir la note moyenne d'un conducteur ainsi que les commentaires des personnes l'ayant évalué.

On a également mis en place l'outil d'intégration continue (CircleCI) et l'outil de qualité de code (Codacy).

Problèmes rencontrés

Mohamed :

- **Mise en place de CircleCI**

J'ai rencontré un problème au niveau de la mise en place de CircleCI (intégration continue).

Le fichier de configuration fourni par CircleCI a dû être adapté. L'image docker utilisée par défaut ne nous permettait pas d'avoir un conteneur MySQL pour lancer nos tests avec des informations de test à l'intérieur.

- **Solution** : Création d'une image docker qui se base sur l'image MySql et NodeJS. La nouvelle image a été hébergée sur le hub de docker à l'adresse suivante : <https://hub.docker.com/r/mohelkar/test>

- **Résultat** : Lancement des tests fonctionnel.

Le fichier de configuration CircleCI se trouve dans le dossier ".circleci/" à la racine du projet.

Marvin :

- **Intégration API Google Maps :**

Ayant eu un module sur l'API de Google Maps à l'IUT, je pensais intégrer l'API Google Maps assez aisément. Cependant, Angular étant assez spécifique, il m'a fallu trouver un module spécifique pour la bonne version d'Angular, la version 1 et les autres versions étant codés de façon totalement différente.

- **Solution** : installation du module ng2-ui/map et intégration sur l'interface des trajets
- **Résultat** : Succès

- **Communication au sein du groupe :**

Malgré les outils de communications instaurés au sein de notre groupe, nous avons eu quelques problèmes de communication, engendrant quelques tensions et du travail effectué par deux personnes. Cela n'a pas eu d'impact majeur sur le projet et a permis de régler quelques soucis que nous avons eu plus tard dans le projet.

- **Solution** : Faire le point sur le travail réalisé, ré-ouvrir la discussion et savoir avancer de façon homogène
- **Résultat** : Succès

Romain :

- **Dépendances corrompues :**

Lors de fusions de branches via Git entre coéquipiers pour réunir nos travaux, certaines nouvelles dépendances ne s'installent pas correctement.

- **Solution :** Forcer l'installation de ces dépendances une par une par une commande (exemple : "npm install -f [nomDeLaDependance] --save").
- **Résultat :** Succès.

- **Communication asymétrique :**

La communication n'a pas toujours été claire et précise, ce qui a engendré par exemple, de la duplication de tâches (notamment concernant la tâche sur le détail d'une offre selon l'utilisateur et le trajet), et aussi certaines tensions.

- **Solution :** Renouer le dialogue est la répartition des tâches, tout en optimisant le travail de l'un et l'autre.
- **Résultat :** Succès.

- **Style de programmation différentes entre coéquipier :**

La méthode de travail est différente entre coéquipier, ce qui a engendré une compréhension difficile du code de certains pour résoudre quelques bugs.

- **Solution :** Communiquer avec l'auteur du code concerné pour essayer de résoudre des bugs (Exemple : affiches des offres lentes, ce qui bloque d'autres requêtes SQL en attente, comme le Login).
- **Résultat :** Succès partiel.

Alex :

- **Rétablissement de la communication entre Marvin et Romain.**

- **Solution :** Pour le résoudre, j'ai pris l'initiative de bien analyser le problème, discuter avec chacun d'entre eux et puis les mettre d'accord pour éviter de mettre le projet en péril.
- **Résultat :** Succès.

- **Récupération de l'identifiant d'utilisateur correctement via le token (dans l'entête des requêtes).**

- **Solution :** Après plusieurs réunions, Romain nous a proposé une méthode qui nous récupère le token, le décode et nous passe un objet qui contient l'identifiant de l'utilisateur.
- **Résultat :** Succès.

- **Difficultés pour la mise en place des tests unitaires selon les Tokens.**

- **Solution :** Après une explication de la part de mon équipe, j'ai pu enfin réaliser mes tests unitaires en plus des tests que j'ai réalisés avec PostMan.
- **Résultat :** Succès.

Yanice :

- **Mise en place de Codacy :**

Le principal problème rencontré fut lors de la mise en place du testeur et analyseur de code “Codacy”, il y avait en fait plusieurs types d’erreurs détectés et connu par ce dernier (sécurité/boucle/etc...).

Cependant il y avait des erreurs dites de “type code”, qui sont des erreurs d’indentations et qui diffèrent d’un IDE à l’autre, c’est donc des erreurs facultatives et qui n’ont pas lieu d’être pour le bon déroulement du projet.

Il a donc fallu trouver une solution afin d’éviter à Codacy d’analyser ce type d’erreurs.

Pour ce faire j’ai dû exploiter certains Patterns proposés par Codacy, certains ont été modifié, d’autres supprimés.

- **Solution** : Modification et suppression de certains Patterns spécifiques
- **Résultat** : Succès

Accès aux outils d’intégration continue et de qualité de code

Intégration continue (CircleCI):

Pour accéder aux builds du projet CovoGreen vous devez vous rendre sur la page suivante :

<https://circleci.com/gh/MarvinRmx/covogreen>

Aucun identifiant nécessaire.

Qualité de code (Codacy):

Pour accéder à l’interface de Codacy il faut se rendre sur la page suivante

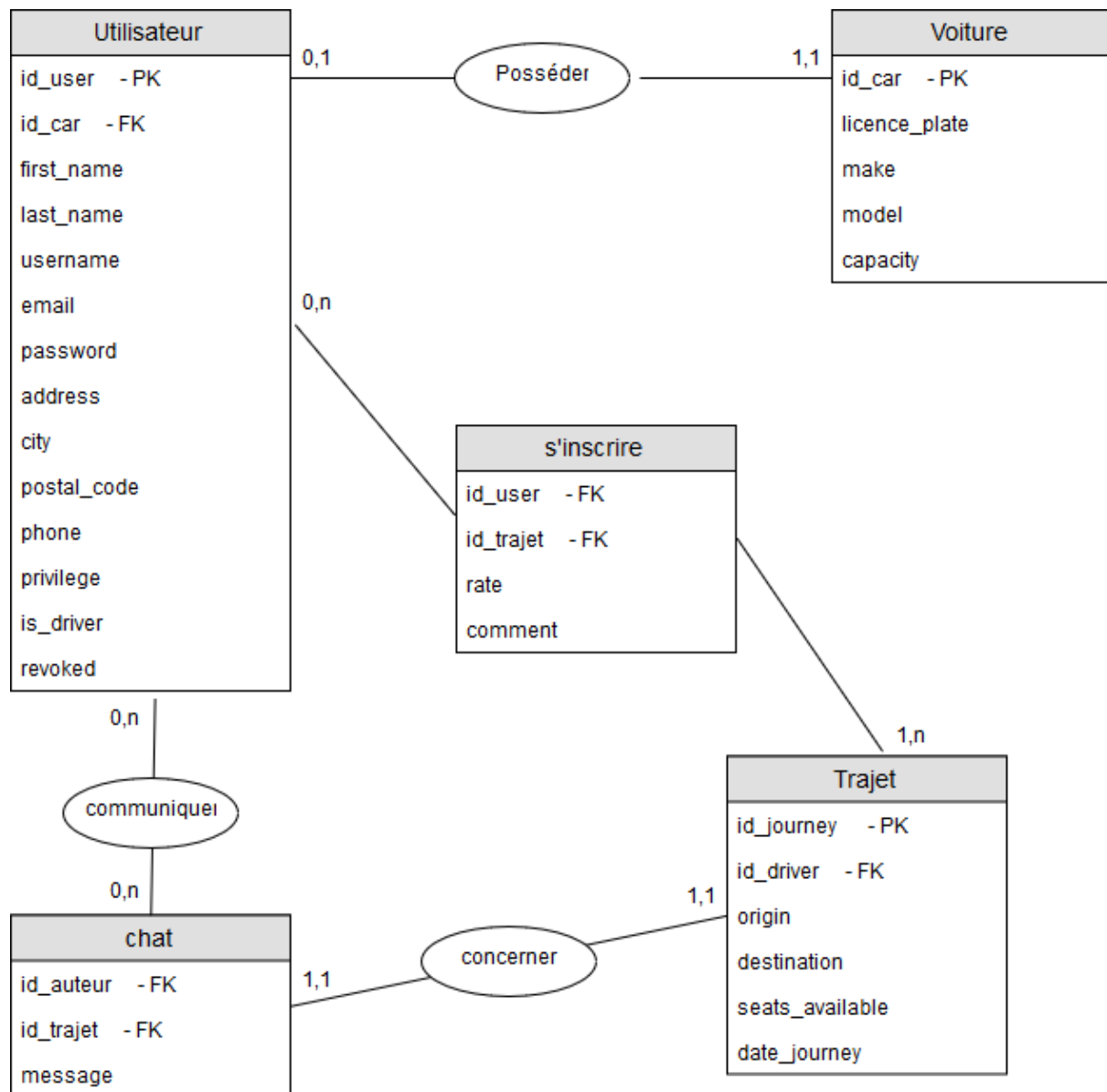
[:https://app.codacy.com/login](https://app.codacy.com/login)

Puis vous devez cliquer le sur bouton “**Login with GitHub**” et utiliser les identifiants suivant :

- **Email** : covogreenproject@gmail.com

- **Password** : Azerty123

MCD Initial et final



UML

