|  |
| --- |
| Pre-Tpi  Ecolopnv |

*Zubieta Pablo*

****

*SI-CA2a*

*06.03.23*

Table des matières

[1 Analyse préliminaire 3](#_Toc131154975)

[1.1 Introduction 3](#_Toc131154976)

[1.2 Organisation 3](#_Toc131154977)

[1.3 Objectifs 3](#_Toc131154978)

[1.4 Planification initiale 3](#_Toc131154979)

[2 Analyse 4](#_Toc131154980)

[2.1 Cahier des charges détaillé 4](#_Toc131154981)

[2.2 Définition de l’audience 7](#_Toc131154982)

[2.3 Analyse concurrentielle 8](#_Toc131154983)

[2.4 Cas d’utilisation 8](#_Toc131154984)

[2.5 Etude de faisabilité 11](#_Toc131154985)

[2.6 MCD 12](#_Toc131154986)

[2.7 Nom du site et du domaine 12](#_Toc131154987)

[2.8 Stratégie de test 12](#_Toc131154988)

[2.9 Budget initial 15](#_Toc131154989)

[2.10 Planification détaillée 15](#_Toc131154990)

[2.11 Historique 16](#_Toc131154991)

[3 Conception 19](#_Toc131154992)

[3.1 Analyse de l’environnement 19](#_Toc131154993)

[3.2 Détermination de l’arborescence du site et des rubriques 19](#_Toc131154994)

[3.3 Définition de la charte graphique 19](#_Toc131154995)

[3.4 Maquette graphique 20](#_Toc131154996)

[3.5 Conception de la Base de données 21](#_Toc131154997)

[3.6 Conception du Code 22](#_Toc131154998)

[3.7 Plugins et librairies 23](#_Toc131154999)

[3.8 Choix de la formule d’hébergement 23](#_Toc131155000)

[4 Réalisation 24](#_Toc131155001)

[4.1 Dossier de réalisation 24](#_Toc131155002)

[4.2 Description des tests effectués 25](#_Toc131155003)

[4.3 Erreurs restantes 26](#_Toc131155004)

[4.4 Dossier d'archivage 26](#_Toc131155005)

[5 Mise en service 27](#_Toc131155006)

[5.1 Rapport de mise en service 27](#_Toc131155007)

[5.2 Liste des documents fournis 30](#_Toc131155008)

[6 Conclusions 31](#_Toc131155009)

[7 Annexes 32](#_Toc131155010)

[7.1 Sources – Bibliographie 32](#_Toc131155011)

[7.2 Manuel d'Installation 32](#_Toc131155012)

[7.3 Manuel d'Utilisation 32](#_Toc131155013)

[7.4 Wireframe 33](#_Toc131155014)

[7.5 Diagramme de flux 38](#_Toc131155015)

[7.6 Archives du projet 42](#_Toc131155016)

# Analyse préliminaire

## Introduction

Le projet ecolopnv est un site web permettant aux enseignants du cpnv de trouver un covoiturage près de chez eux en fonction de leur horaire. Ce site crée avec un Framework PHP : Laravelel.

## Organisation

Elève : Zubieta Pablo [zubieta-rodriguez.pablo-fernando@cpnv.ch](mailto:zubieta-rodriguez.pablo-fernando@cpnv.ch) 0764788611

Responsable de projet : Benzonana Pascal [benzonana.pascal@cpnv.ch](mailto:benzonana.pascal@cpnv.ch)

## Objectifs

* Être capable de mener un projet Web à son terme.
* Comprendre le fonctionnement d’un MVC.
* Faire et savoir gérer une planification.

## Planification initiale

La planification de ce projet sera faite sur Icesrum

Elle se divise en 3 sprints principaux.

Le premier pour la prise en main de Laravel et le login

Le second l’implémentation de l’algorithme de planification des covoiturages

Le dernier pour mise en place du mailing et des API

# Analyse

## Cahier des charges détaillé

### INFORMATIONS GENERALES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Candidat** | Nom : | **ZUBIETA** | **Prénom :** | Pablo |
|  |  | Pablo-Fernando.ZUBIETA- [RODRIGUEZ@cpnv.ch](mailto:RODRIGUEZ@cpnv.ch) |  | |
| **Lieu de travail :** |  | | | |
| **Orientation :** | * 88601 Développement d’applications   ☒ 88602 Informatique d’entreprise   * 88603 Technique des systèmes | | | |
| **Chef de projet** | **BENZONANA** | | Pascal | |
| [Pascal.benzonana@cpnv.ch](mailto:Pascal.benzonana@cpnv.ch) | |  | |
| **Expert 1** |  | |  | |
|  | |  | |
| **Expert 2** |  | |  | |
|  | |  | |
| **Période de réalisation :** | Du 1er février 2023 à 10h35 au 31 mars 2023 à 08h50 | | | |
| **Horaire de travail :** | 16 périodes de 45 min par semaine | | | |
| **Nombre d'heures :** | Environ 74 heures | | | |
| **Planning (en H ou %)** | Analyse : 20% | | | |
| Implémentation : 45% | | | |
| Tests : 15% | | | |
| Documentations : 20% | | | |

1. PROCÉDURE
   * Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le 1er jour.
   * Le cahier des charges est approuvé par les deux experts. Il est en outre présenté, commenté et discuté avec le candidat. Par sa signature, le candidat accepte le travail proposé.
   * Le candidat a connaissance de la feuille d’évaluation avant de débuter le travail.
   * Le candidat est entièrement responsable de la sécurité de ses données.
   * En cas de problèmes graves, le candidat avertit au plus vite les deux experts et son CdP.
   * Le candidat a la possibilité d’obtenir de l’aide, mais doit le mentionner dans son dossier.
   * A la fin du délai imparti pour la réalisation du TPI, le candidat doit transmettre par courrier électronique le dossier de projet aux deux experts et au chef de projet. En parallèle, une copie papier du rapport doit être fournie sans délai en trois exemplaires (L’un des deux experts peut demander à ne recevoir que la version électronique du dossier). Cette dernière doit être en tout point identique à la version électronique.

### TITRE

Plateforme de partage de co-voiturage en lien avec les horaires EDT.

### SUJET

Le projet consiste en la création d’une plateforme permettant du co-voiturage entre collègues et la génération de propositions de co-voiturages qui prennent en compte l’alternance des conducteurs.

### MATÉRIEL ET LOGICIEL À DISPOSITION

1 ordinateur type CPNV

* Windows 10
* PHPStorm ou Visual Studio
* Microsoft Office 2016
* Adobe Photoshop 2017
* Balsamiq Mokup
* Apache et MySQL

### PRÉREQUIS

Le candidat a une bonne connaissance dans les technologie Web et de développement d’applications

### DESCRIPTIF DU PROJET

Chaque trimestre, l’horaire change et plusieurs personnes viennent de la plaine. Afin de faciliter l’organisation du co-voiturage entre collègues, la plateforme va proposer des suggestions de co-voiturages qui prennent en compte l’alternance des conducteurs et le lieu de départ.

1. Les enseignants intéressés se connecteront à l’application et définiront leur ville de domicile. Seuls les collègues possédant un véhicule pourront utiliser l’application.

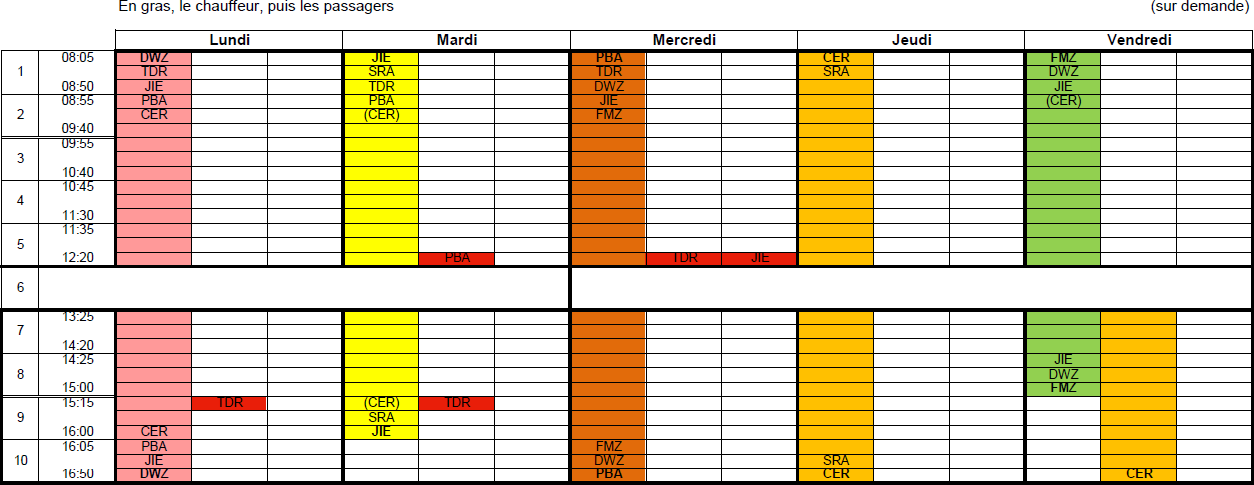
Lors de la première connexion, la personne indiquera le nombre maximum de personnes qu’elle peut transporter.

1. Une fois connecté, l’application Web récupérera les informations de l’enseignant connecté et cherchera dans la base de données des collègues habitant la même ville de départ puis cherchera sur un fichier EDT\* (fichier en format texte contenant les horaires de chaque trimestre) des horaires qui peuvent convenir au covoiturage.

L’application tiendra compte aussi de la capacité de la voiture et de la disponibilité des enseignants car certains peuvent être déjà inclus dans un autre co-voiturage.

*Un exemple de proposition se trouve sur la page suivante : libre au candidat de modifier le*

*visuel. Dans l’exemple, il y a une couleur par chauffeur et la possibilité de rentrer en train.*



1. Si l’enseignant connecté est intéressé par la proposition, il pourra la valider et l’application enverra un mail aux autres enseignants inclus dans la proposition.
2. Dès que la proposition est validée par les ¾ des enseignants concernés, le statut des enseignants ayant donné leur accord est mis comme occupé afin d’éviter que l’application ne les intègre dans un autre co-voiturage pour une même période. L’application envoie un mail aux personnes incluses dans le co-voiturage avec le planning similaire à celui montré ci-dessus)

\* Un exemple de l’horaire du 3ème trimestre sera fourni au candidat.

### LIVRABLES

Le candidat est responsable de livrer à son chef de projet et aux deux experts :

* + 02.02.2022: une planification initiale des sprints au format électronique.
  + 17.02.2022 : une version intermédiaire du rapport de projet, ainsi que les stories. L’analyse et la conception du premier sprint sont terminées
  + 02.03.2022 : une version intermédiaire du rapport de projet, ainsi que la planification mise à jour. Un premier livrable du produit sera fourni sur
  + 22.03.2022 : une version du produit avec la liste et le publipostage sera livrée
  + 01.04.2022 : la version finale du rapport de projet ainsi que l'ensemble de son travail pratique.
  + Chaque vendredi en fin de journée, le journal de travail devra être transmis par courriel

### POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

La grille d’évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, …).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (Point A14 à A20) :

1. Les fonctionnalités ont été testées avec l’horaire des derniers trimestres (S3 et S4)
2. La connexion avec l’insertion des données chauffeur dans la base fonctionne
3. *L’algorithme de création de propositions tient compte de l’alternance des chauffeurs*
4. Ergonomie de l’interface : elle respecte les standards vus en module I-CT 120.
5. *La modélisation de la base de données respecte la forme normale de Boyce Codd*
6. Explication pour le déploiement de la web application sur un serveur
7. Description et qualité des tests effectués
8. HORAIRE DE TRAVAIL
9. VALIDATION

|  |  |  |
| --- | --- | --- |
|  | Lu et approuvé le : | Signature : |
| Candidat : |  |  |
| Expert n°1 : |  |  |
| Expert n° 2 : |  |  |
| Chef de projet : |  |  |

## Définition de l’audience

Ce projet est conçu pour les enseignants du cpnv disposant d’une voiture

## Analyse concurrentielle

Il y a, à ma connaissance, aucune application de covoiturage spécifiquement pour les enseignant du cpnv.

Pour ce qui est de blablacar ou E-voiturage.ch, elles ne disposent pas des horaires des enseignants et ne peuvent donc pas auto suggérer des trajets.

Donc ce site n’a pas vraiment de concurrence.

## Cas d’utilisation

Use case : l’administrateur ajoute L’edt

Scénario

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’administrateur  Ajoute le nouvel edt dans la base de données |  | Active un trigger qui lance la drop la table des horaires et lance une procédure stockée |
| La procédure stockée traite d’edt |  | Elle remplit la table horaire avec nouveau les horaire |
| Elle sélectionne tous les utilisateurs uniques de la table horaire |  | Et les insère dans la table utilisateur |

Use case : l’utilisateur s’enregistre sur le site pour participer au covoiturage

Scénario

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le sign in de la navbar |  | L’utilisateur est redirigé sur la page sign in |
| L’utilisateur remplit le champ acronyme | L’utilisateur possédant cet acronyme doit exister dans la base de données |  |
| L’utilisateur remplit le champ email | L’entrée doit être un email |  |
| L’utilisateur sélectionne une des entrées dans le champ lieux d'habitation |  |  |
| L’utilisateur remplit le champ place de voiture | La valeur doit être nombre entier naturel |  |
| L’utilisateur remplit le champ mot de passe et confirmation | Les deux champs doivent être identique |  |
| L’utilisateur clique sur le bouton valider | Tous les champs doivent être remplit | L’utilisateur en question est activé, et il est authentifié et redirigé sur la page d’accueil |

Extension : Les champ sont mal remplis

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le bouton valider |  | Une erreur décrivant le problème apparait sur le champ en question |

Use case : l’utilisateur consulte son horaire

Scénario

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le login de la navbar |  | L’utilisateur est redirigé sur la page login |
| L’utilisateur remplit le champ acronyme du login | L’utilisateur possédant cet acronyme doit être activé |  |
| L’utilisateur remplit le champ mot de passe | Le mot de passe doit être correct |  |
| L’utilisateur clique sur le bouton valider |  | L’utilisateur est authentifié et redirigé sur la page d’accueil |
| L’utilisateur clique sur my schedule |  | L’utilisateur est redirigé sur la page schedule ou son horaire est affiché |

Extension : Les champs sont mal remplis

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le bouton valider |  | Une erreur décrivant le problème apparait sur le champ en question |

Extension : l’utilisateur accède sans se connecter

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur rentre l’url de la page sans être connecter |  | Il est redirigé sur la page login |

Use case : l’utilisateur modifie ses informations

Scénario

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le login de la navbar |  | L’utilisateur est redirigé sur la page login |
| L’utilisateur remplit le champ acronyme du login | L’utilisateur possédant cet acronyme doit être activé |  |
| L’utilisateur remplit le champ mot de passe | Le mot de passe doit être correct |  |
| L’utilisateur clique sur le bouton valider |  | L’utilisateur est authentifié et redirigé sur la page d’accueil |
| L’utilisateur clique sur my profile |  | L’utilisateur est redirigé sur la page profile ou son profile est affiché |
| L’utilisateur clique sur l’un des boutons de modification de |  | Les champs de valeur sont remplacés par des entrée de formulaire et un bouton valider apparait |
| L’utilisateur clique sur le bouton valider |  | La page est rechargée avec les informations mise à jour |

Extension : Les champs sont mal remplis

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le bouton valider |  | Une erreur décrivant le problème apparait sur le champ en question |

Extension : l’utilisateur accède sans se connecter

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur rentre l’url de la page sans être connecter |  | Il est redirigé sur la page login |

Use case : l’utilisateur veut valider ou rejeter son covoiturage

Scénario

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le login de la navbar |  | L’utilisateur est redirigé sur la page login |
| L’utilisateur remplit le champ acronyme du login | L’utilisateur possédant cet acronyme doit être activé |  |
| L’utilisateur remplit le champ mot de passe | L’e mot de passe doit être correct |  |
| L’utilisateur clique sur le bouton valider |  | L’utilisateur est authentifié et redirigé sur la page d’accueil |
| L’utilisateur clique sur my carpooling |  | L’utilisateur est redirigé sur la page carpooling ou ses covoiturages sont affichés |
| L’utilisateur clique sur l’un des boutons de modification de |  | Les champs de valeur sont remplacés par des entrée de formulaire et un bouton valider apparait |
| L’utilisateur clique sur le bouton valider |  | La page est rechargée avec les informations mise à jour |

Extension : Les champs sont mal remplis

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le bouton valider |  | Une erreur décrivant le problème apparait sur le champ en question |

Extension : l’utilisateur accède sans se connecter

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur rentre l’url de la page sans être connecter |  | Il est redirigé sur la page login |

Extension : l’utilisateur refuse le covoiturage

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le bouton reject |  | La page est rechargée avec les informations mise à jour |

## Etude de faisabilité

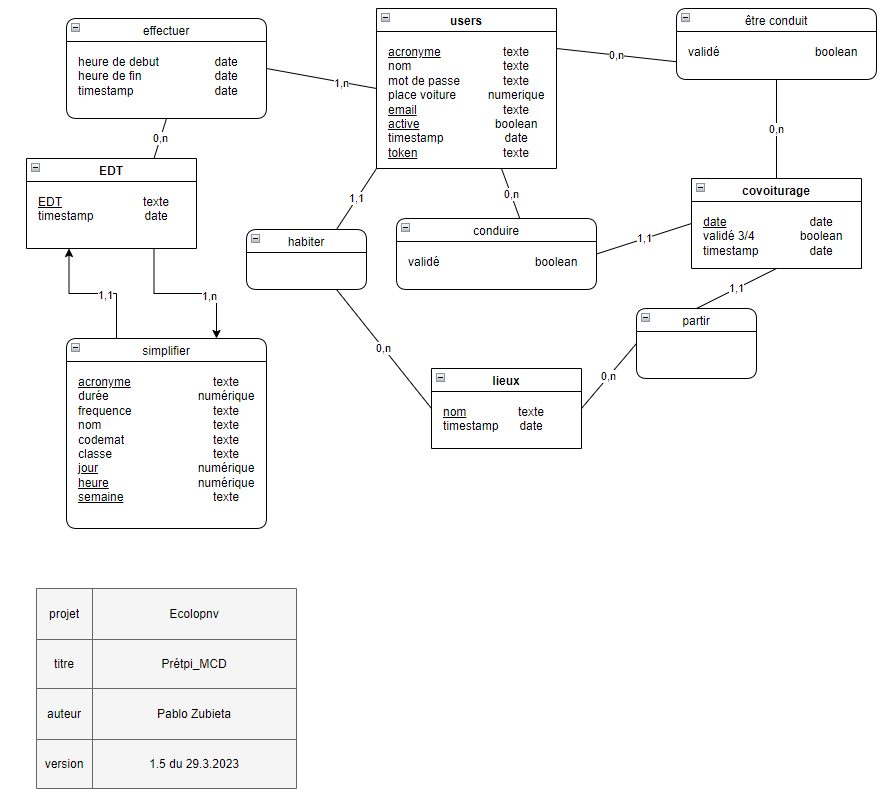
Il existe une infinité de manière de faire un site web. Utiliser un CMS, une diversité de langage allant du JS au Python en passant par le Ruby, Avec des Framework ou en natif, Avec du sql ou du nosql. Ayant beaucoup apprécier le PHP c’est vers ce langage que je me suis orienté.

Il existe j’avais 3 principale piste pour faire le Backend de cette application, qui en est l’aspect principale.

* Le PHP natif : étant la matière enseignée au cpnv c’est de loin l’approche que je maitrise le mieux.
* Laravel : est donc un Framework PHP très commun et populaire pour le bakcend, il est d’ailleurs enseigné au CPNV.
* Symphony : est également un Framework PHP.

Mon choix c’est porté sur Laravel. Pour trois raisons. La première est que Laravel est une technologie employée dans l’industrie. La Deuxième est qu’étant enseigné ici je pense que se sera plus simple de me faire aider en cas de difficulté. Et la dernière est l’envie de défi c’est important de sortir de sa zone de confiance quand on est dans un processus d’apprentissage.

## MCD



## Nom du site et du domaine

[http://covoiturage.mycpnv.ch](http://covoiturage.mycpnv.ch/)   domaine louer par le CdP, meme si j’aimais bien Ecolopnv.

## Stratégie de test

L’utilisation de la méthode agile rend la stratégie de test très évidente.

Les tests sont créés à l’écriture des stories Et sont effectuer lors de la sprint review.

Listes des tests

|  |  |
| --- | --- |
| Nom | Description |
| user login | Given :a user in the database When :enter his username and his right password Then :he’s log in the Website |
| wrong password | Given :a user in the database When :enter his Email and his wrong password Then :he’s not log in the website and informe that he got his password wrong |
| Blank log | Given :login left blank When :try to confirm Then :the user can’t enter |
| user entry | Given :given any user When :the user enter his email address his town, the number of seat in his car and his password twice Then :he is inserted in the databases |
| sign in log | Given :user conforme information When :he sign in Then :he get log |
| different passwords | Given :two different input on the password field When :user validated Then :the not the same password error is display to the user |
| entry existe | Given :the user has enter a username already in the database When :he validated Then :account already exist error is display to the user |
| user phasing | Given :the user is loged in When :he is on the home menu Then :his navbar is : my carpooling my profile my schedule logout |
| unconnected phasing | Given :the user is not log in When :he is on the home menu Then :his navbar is: log in sign in |
| user name | Given :the user is loged in When :he is navigating Then :his name appered on the navbar |
| homepage content | there is the homepage content |
| navbar | there is a nav bar |
| gabarit | there is a gabarit |
| user edt | Given : a edt data When : the user is on his schedule page Then : he can read his schedule |
| create carpooling | Given : a place and a time When : the function is call Then : it create a carpooling entry with a driver and a users\_has carpooling entry with the passenger |
| carpooling general | Given : a time and the entry in the database When : the function is used Then : a carpooling is asigned to all the users in the database |
| one person carpooling | Given : a place or time where only one person need a carpooling When : the function is used Then : no entry will be created in the database (cause there’s ne carpooling himself) |
| schedule in db | Given : a user in the database When : he activate his account Then : a entry is created in the user has edt table with his schedule |
| Place creation | Given : a place not contain in the databases When : a user write it in his form Then : it’s inserted in the place table and the id link as a forein key to the user |
| Place created | Given :a place contain in the databases When : a user write it in his form Then : the id link as a forein key to the user |
| acces | i can acces the web via http://covoiturage.mycpnv.ch |
| no regression | there no regression between the local version and the online one |
| diferent driver | Given : a carpooling record  When :a new carpooling is sugested with the same users  Then :the driver can't be same as the carpooling record |
| new carpooling | Given : a list carpooling  When : some of them fail  Then : a new carpooling instance is generate |
| mail to driver | Given : the driver of a carpooling  When : generate the carpooling entry  Then : send a mail to the driver |
| mail to user | Given : a user of a carpooling  When : generate the carpooling entry  Then : send a mail to the user |
| mail token link | Given : a user  When : the carpooling entry is created and the mail send  Then : the mail contains a link with this user token |
| validate | Given : a user in a carpooling  When : he use the links he recived by mail  Then : the user carpooling entry is validate |
| validation view | Given : a user  When : use his mail link  Then : he his redirected to a validate webpage |

## Budget initial

C’est 0.00 CHF

Pour un petit site comme ça l’hébergement est négligeable selon mes calcule il est des 1,35 CHF/par année

Sauf si on considère le prix des licences dans ce cas

* C’est PHPstorm = 247 CHF
* Balsamiq = 89 CHF
* Total = 336 CHF

## Planification détaillée

Etant donné que le projet se fait Agile la planification est sous forme de Sprint

Il n’y a pas vraiment de planification détaillé appart les dates des Sprint.

Réception du CDC 01.02

Sprint 1 du 09.02 au 24.02

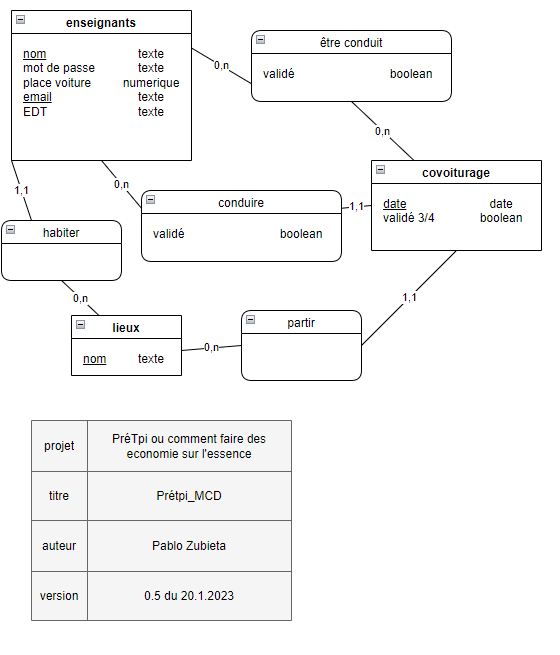
Sprint 2 du 28.03 au 10.03

Sprint 3 du 14.03 au 24.03

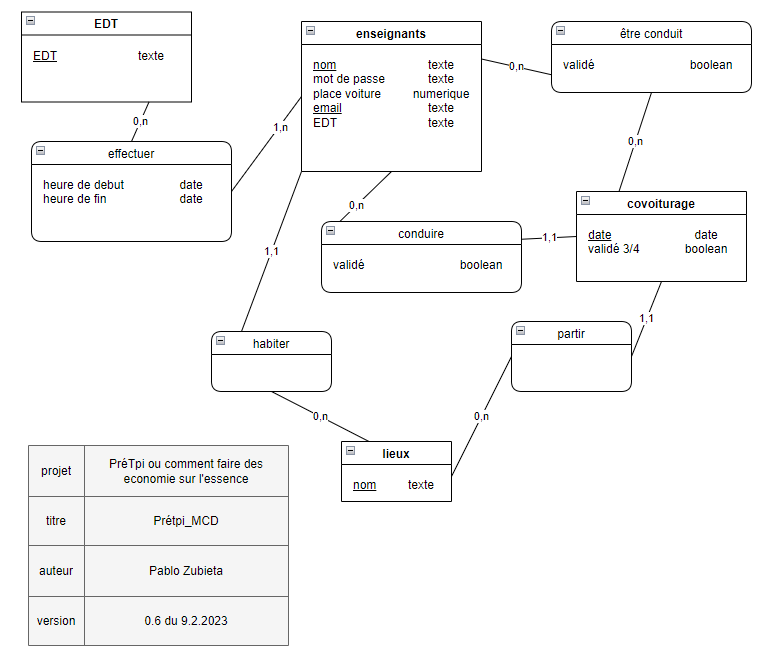
Et release du projet le 29.03

## Historique

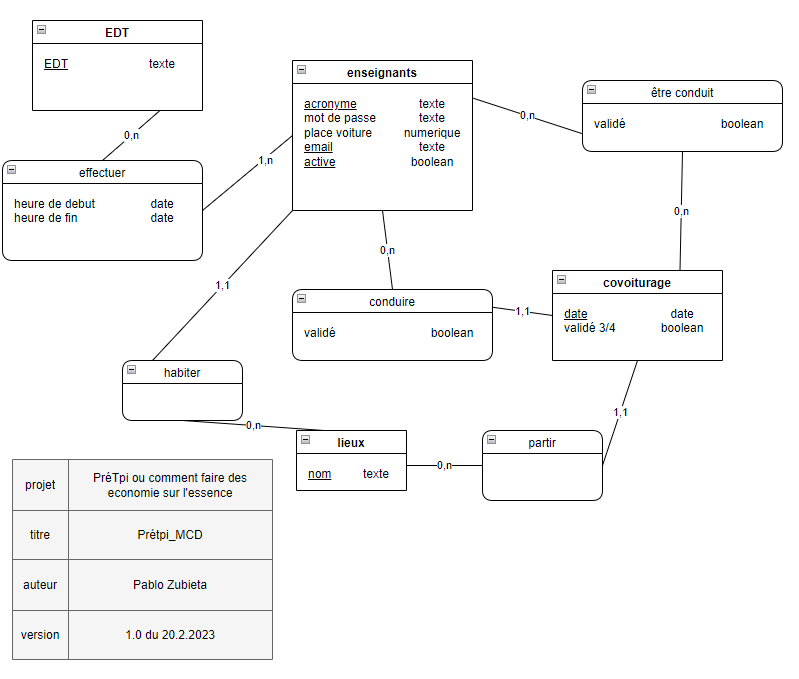
MCD première version du 20.01 obsolète depuis 09.02



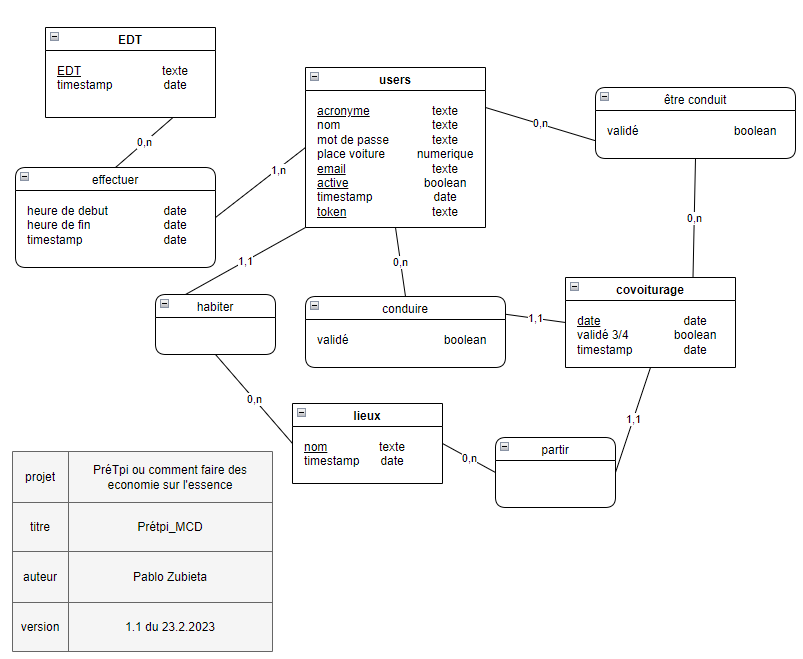
MCD deuxième version du 09.02 obsolète depuis 09.02



MCD troisième version du 20.02 obsolète depuis 23.02



MCD quatrième version du 23.02 obsolète depuis 29.03



MLD première version du 23.02 obsolète depuis 28.02

Une image contenant diagramme

Description générée automatiquement

# Conception

## Analyse de l’environnement

Environement de développement

Windows 10 family

Mariabd 10.10

Phpstorm 2022.2.1

Laravel9.0 ->composer 2.5.2

Php8.1

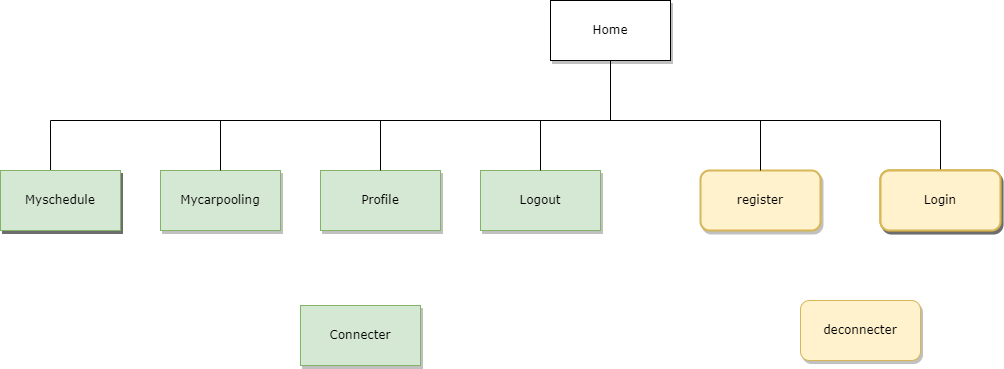
Hébergement

Php8.1

Apache/2.4.37

8.0.32-cll-lve

## Détermination de l’arborescence du site et des rubriques



## Définition de la charte graphique

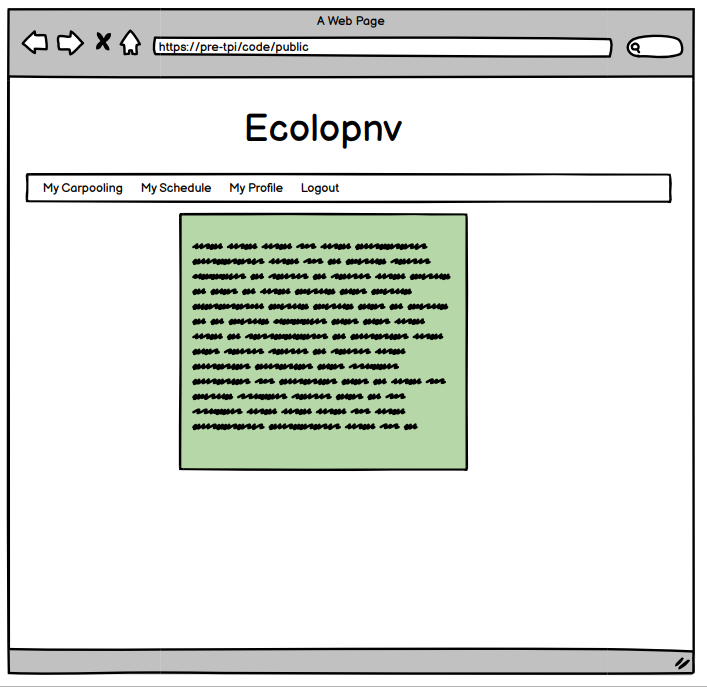
Pour ce qui est de la charte graphique J’ai pris arrangement de couleur autour du Vert, parce que vous savez l’écologie tout ça.



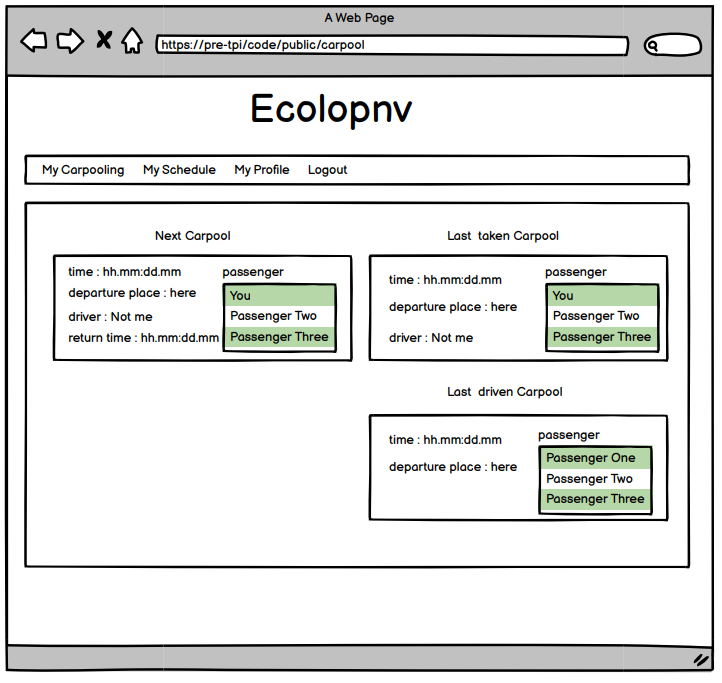
## Maquette graphique

Voici deux des wireframes de projet les autres seront en annexe.

La homepage :

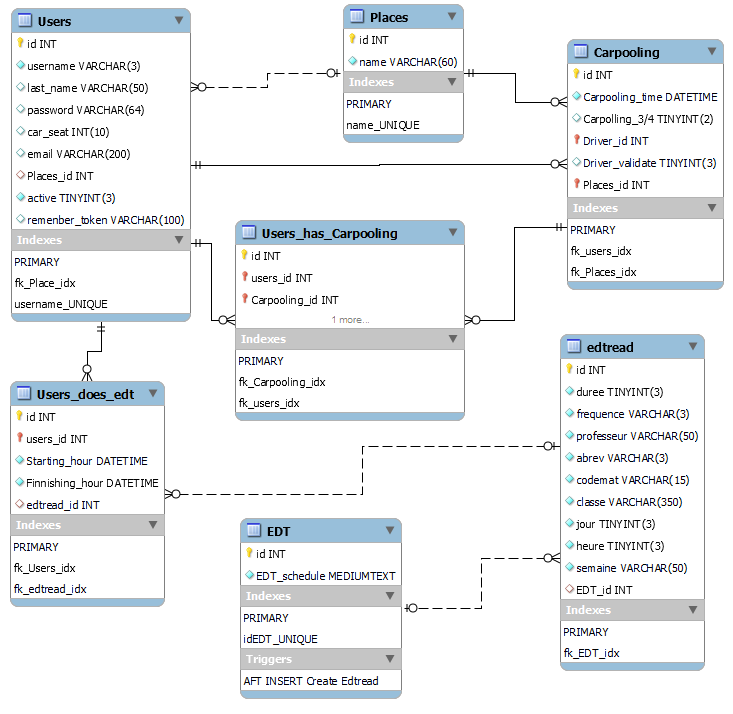


La Page Carpooling :



## Conception de la Base de données

Voici le MLD Actuel :

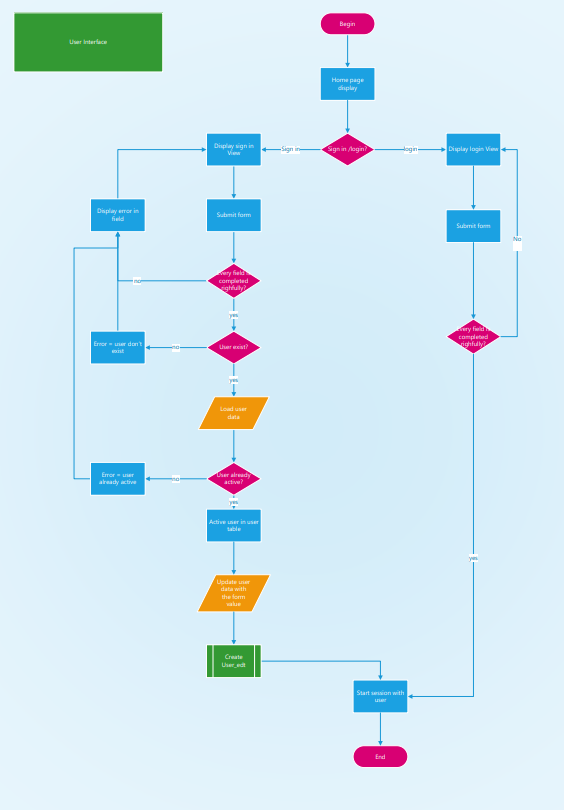


L’implémentation dans la base de données n’est pas exactement celle-ci, car

L’utilisation de procédure stocker en développement rend inutile d’avoir certaines des clés étrangères dans cette base de donnée.

## Conception du Code

Voici un des diagrammes de flux intéressant les autres seront en annexe.



## Plugins et librairies

Etant donné que ce projet c’est fait sous Laravel il y a une tonne de librairie associer tel que Eloquent (la librairie qui gérer l’accès à la base de données).

On peut citer également la Bootstrap qui mon Framework css.

## Choix de la formule d’hébergement

Donc le CdP a choisi l’hébergement sur swisscenter. Je n’ai pas beaucoup plus de précision là-dessus

# Réalisation

## Dossier de réalisation

### Répertoire

1. Répertoire "physique" du projet :
   * Sur la machine SC-C236-PC04
   * C:\xampp\htdocs\Pretpi
2. Repository distant :
   * https://github.com/PabloZubieta/Pre\_Tpi

### Dossier bases de données

|  |  |
| --- | --- |
| **Nom** | **Description** |
| CreateDB.sql | Script de création de la base de données (DB) |
| MCD\_V4 | Fichier contenant le MCD de la DB |
| MLD\_V2 | Fichier contenant le MLD de la DB |
| Edt\_insertion.sql | Script d’insertion de l’EDT dans la base de données |
| EPX\_CRS\_SEMAINE | Ficher texte de l’edt |

### Dossier documentation

|  |  |
| --- | --- |
| **Nom** | **Description** |
| Pré-TPI-22-Zubieta\_Plateforme de partage de co-voiturage | Cahier des charges |
| Timesheet.html | Journal de travail |
| "documentation.pdf" | La documentation que vous êtes en train de lire |
| "Journal de Bord.pdf" | Le journal de bord du projet |

### Dossier code

|  |  |
| --- | --- |
| **Nom** | **Description** |
| App/http/controller : | Tous les fichier de Classe Controllers de ce site |
| CarpolingController.php |
| UserController.php |
| PlaceController.php |
| ScheduleController.php |
| Users\_has\_carpooling  Controller.php |
| App/Model : | Tous les fichier de Classe Model de ce site |
| Carpoling.php |
| User.php |
| Place.php |
| Schedule.php |
| Users\_has\_carpooling.php |
| Route/web | Toutes les Routes de ce site |
| Resources/views : | Toutes les Vues de ce site |
| Carpooling.blade.php |
| home.blade.php |
| layout.blade.php |
| schedule.blade.php |
| Users/signin.blade.php |
| Users/login.blade.php |
| Users/profile.blade.php |
| database/Migrations | Dossier contenant toutes les migrations de ce site |
| Config/database.php | Fichier de configuration de la base de données |
| Vendor | Dossier contenant l’intégralité de la librairie Laravel |

### Autres fichiers

|  |  |
| --- | --- |
| **Nom** | **Description** |
| dailytask.php | Script php d’exécution journalière pour le serveur |

## Description des tests effectués

Listes des tests

|  |  |
| --- | --- |
| Nom | Description |
| user login | Success |
| wrong password | Success |
| Blank log | Success |
| user entry | Success |
| sign in log | Success |
| different passwords | Success |
| entry existe | Success |
| user phasing | Success |
| unconnected phasing | Success |
| user name | Success |
| homepage content | Success |
| navbar | Success |
| gabarit | Success |
| user edt | Success on dev Failed in prod |
| create carpooling | Success on dev Failed in prod |
| carpooling general | Success |
| one person carpooling | Success |
| schedule in db | Success |
| Place creation | Success |
| Place created | Success |
| acces | Success |
| no regression | Success |
| diferent driver | Failed |
| new carpooling | Failed |
| mail to driver | Failed |
| mail to user | Failed |
| mail token link | Failed |
| validate | Failed |
| validation view | Failed |

## Erreurs restantes

|  |  |  |  |
| --- | --- | --- | --- |
| **Erreur** | **Description** | **Conséquences** | **Action** |
| Middleware | Depuis le navigateur en insérant l’url l’utilisateur peut accéder à des routes qui ne doivent pas se produire. | Exemple un utilisateur non inscrit peut inscrire /logout ce qui va lui produire une erreur. | Faire des recherches sur le fonctionnement du middleware de laravel, corriger les routes en fonction |
| Nb place négative | Un utilisateur peut rentrer un nombre de place négative | Il est possible que l’algorithme de création d’un covoiturage puisse être altérer | Mettre une contrainte dans le formulaire de création et de modification sur |
| Taches planifiées | La tache planifiée sur le serveur ne s’effectue pas correctement | Aucune Entrée Carpooling est créé (Ah bah dommage ce n’est pas comme si c’était le cœur du projet) | Faire des recherches sur le fonctionnement des taches planifiées et voir pourquoi elle ne s’effectue pas |

## Dossier d'archivage

# Mise en service

## Rapport de mise en service

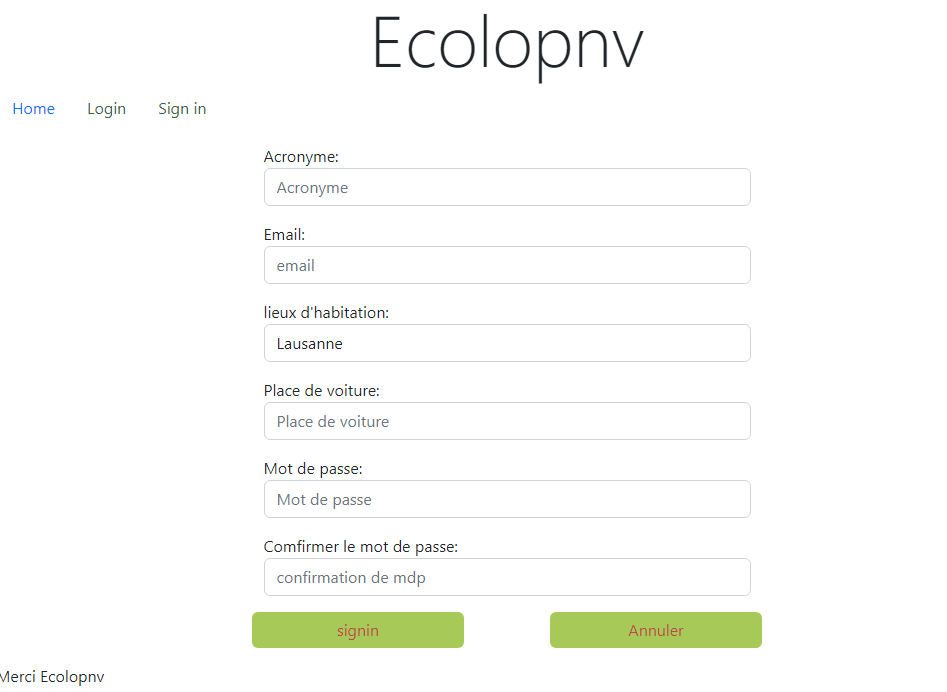
### Publication chez l’hébergeur

Voici une petite visite du site avec home page

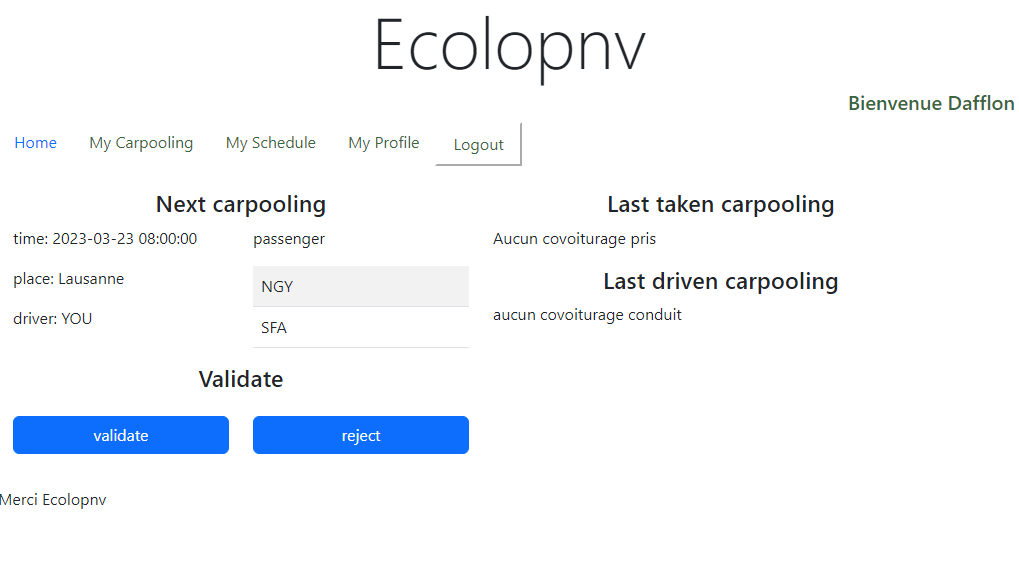
Une image contenant texte

Description générée automatiquement

La page de sign in ou l’utilisateur rentre ses informations



La page ou l’utilisateur voir la liste de ses covoiturages et valider son prochain covoiturage



La page avec son horaire

Une image contenant graphique

Description générée automatiquement

La page contenant ses informations qu’il peut modifier à volonté

Une image contenant texte

Description générée automatiquement

### Guide d’installation

* Pour l’installation il vous fait évidement php8.1 et mariadb
* Il faudra également composer que vous trouverez à cette adresse : <https://getcomposer.org/>
* Une fois tous ses programmes installation réside dans l’obtention de la librairie Vendor
* Si vous récupérer mon projet dans le git vous aurez tout le projet sauf la librairie Vendor qui n’es pas synchroniser
* Vous devrez donc crée un projet Laravel et copier le dossier vendor que composer va vous générer
* Exécuter les scripts sql contenu dans le dossier database de manière à crée la base de données
* Vous localiser dans le dossier Code et lancer la commande : php artisan serve
* Et votre projet devrait tourner

### Erreurs restantes

|  |  |  |  |
| --- | --- | --- | --- |
| ***Erreur*** | ***Description*** | ***Conséquences*** | ***Action*** |
| *Middleware* | *Depuis le navigateur en insérant l’url l’utilisateur peut accéder à des routes qui ne doivent pas se produire.* | *Exemple un utilisateur non inscrit peut inscrire /logout ce qui va lui produire une erreur.* | *Faire des recherches sur le fonctionnement du middleware de laravel, corriger les routes en fonction* |
| *Nb place négative* | *Un utilisateur peut rentrer un nombre de place négative* | *Il est possible que l’algorithme de création d’un covoiturage puisse être altérer* | *Mettre une contrainte dans le formulaire de création et de modification sur* |
| *Taches planifiées* | *La tache planifiée sur le serveur ne s’effectue pas correctement* | *Aucune Entrée Carpooling est créé (Ah bah dommage ce n’est pas comme si c’était le cœur du projet)* | *Faire des recherches sur le fonctionnement des taches planifiées et voir pourquoi elle ne s’effectue pas* |

*.*

## Liste des documents fournis

* Documentation

# Conclusions

Voilà qui conclus mon Pré-Tpi ça a été la croix et la bannière de faire ce projet le résultat final en est d’ailleurs une preuve. Enormément de choses ne sont pas fidèle à ce qui décrit dans le cahier des charges. Mais même si ça m’a causé quelques nuits blanches. J’ai adoré avoir un défi comme ça et je compte bien prendre du temps ces vacances pour réussir à le terminer comme je le voudrais. Je pense cependant que c’est une très bonne préparation pour le métier de Dev backend, qui est une voie que j’envisage de suivre.

Les principales difficultés que j’ai rencontrées étant le traitement de string avec le moteur de mariadb (c’était une horrible idée et on n’y reprendra plus), la conversion de l’edt en temps au format datetime, Les taches planifiées sur le serveur. Je trouve ça frustrant de ne pas avoir réussi à aborder les fonctions de mailing et la gestion des token et api, mais je vais le faire (je suis tenace).

Pour conclure merci pour ce projet intéressant qui a su me défier et l’as donnée un aperçu du métier de développeur back-end. Je tiens à remercier M.Hurni pour l’aide qu’il m’as apporter dans la compréhension du Framework Laravel, Merci également à Lighea Ardia pour avoir relu le texte de ma homepage. Merci à M.Benzonana d’avoir eu suffisamment confiance en moi pour me confier ce projet.

# Annexes

## Résumé du rapport

Mon projet était la création d’un site de covoiturage. Pour ce se faire je disposais d’un EDT (un fichier texte contenant la totalité des horaires et des collaborateurs du cpnv). Je disposais également d’un hébergeur. J’avais choisi de faire cette application/site en PHP, et en utilisant le Framework Laravel. J’ai choisi de travailler en méthode AGILE.

Pour la mise en œuvre, elle s’est séparé en quatre partie. Les 2 premières semaines, j’ai avant tout pris en main mon Framework et crée repository, MLD, MCD. Les deux semaines suivantes, J’ai traité mon EDT pour en extraite les horaires et enseignants, et mettre ainsi en place mon login. Les semaines 5 et 6 J’ai travaillé les horaires de mon EDT de manière à les transformer en heure de départ et heure de fin. Les deux dernières semaines ont été consacrée à la création des covoiturages et l’hébergement. Les gros soucis de la mise en œuvre ayant été le suivi de la documentation et l’hébergement.

Donc par rapport au Cdc, la réalisation mon projet est clairement incomplet. Mais il a des réussites significatives, comme la création d’une procédure stockée, l’hébergement, ou même la prise en main d’un Framework Php. Je pense que l’on peut aussi souligner les défauts : tel que l’utilisation de icecrum résultant dans un journal et une planification pas conforme à ce qui était attendu. Ma difficulté à choisir mes priorités, comme le fait omettre des points techniques simple a réalisé, juste par désintéressement. Mon inconsistance dans ma documentation.

## Sources – Bibliographie

<https://www.youtube.com/watch?v=FC7JQItVqzM>

<https://www.youtube.com/watch?v=MYyJ4PuL4pY>

<https://laravel.com/docs/9.x>

<https://www.w3schools.com>

<https://roadmap.sh/backend>

<https://www.educba.com/data-science/data-science-tutorials/sql-tutorial/>

<https://mariadb.org/wp-content/uploads/2023/02/MariaDBServerKnowledgeBase.pdf>

<https://www.youtube.com/watch?v=NrBJmtD0kEw>

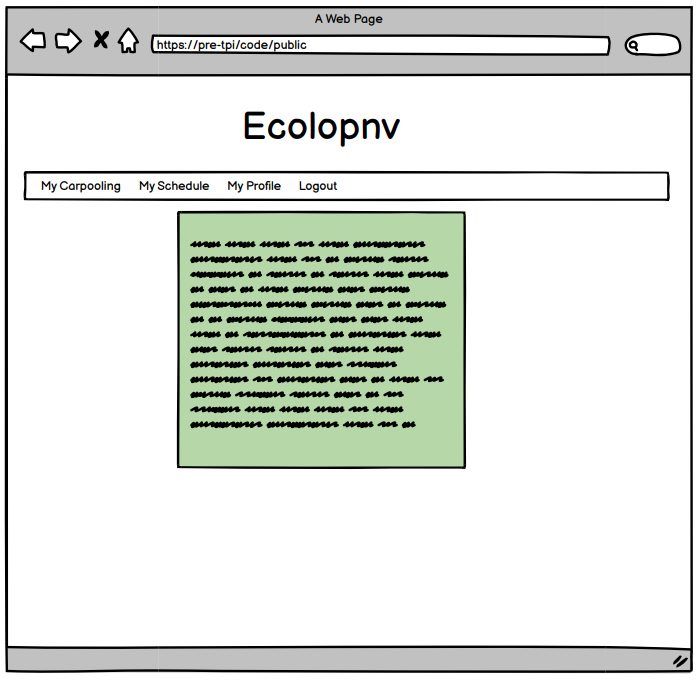
## Manuel d'Installation

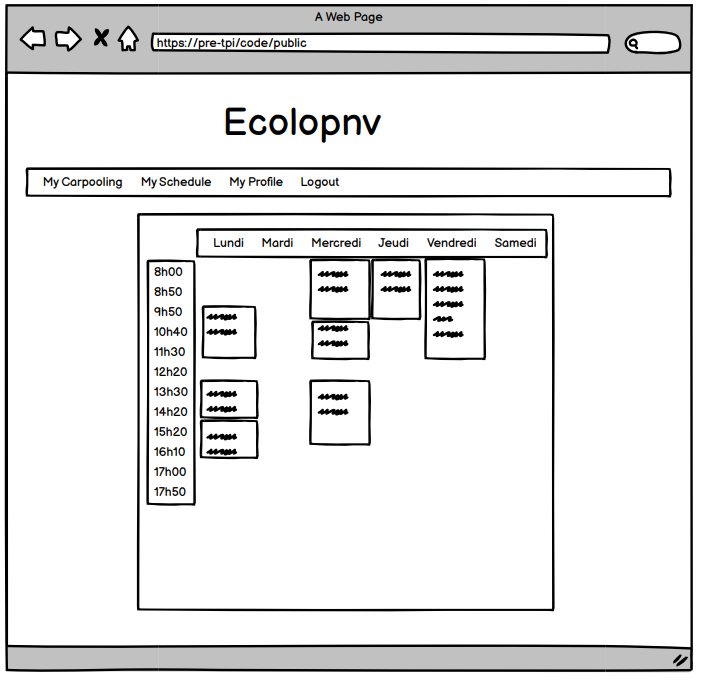
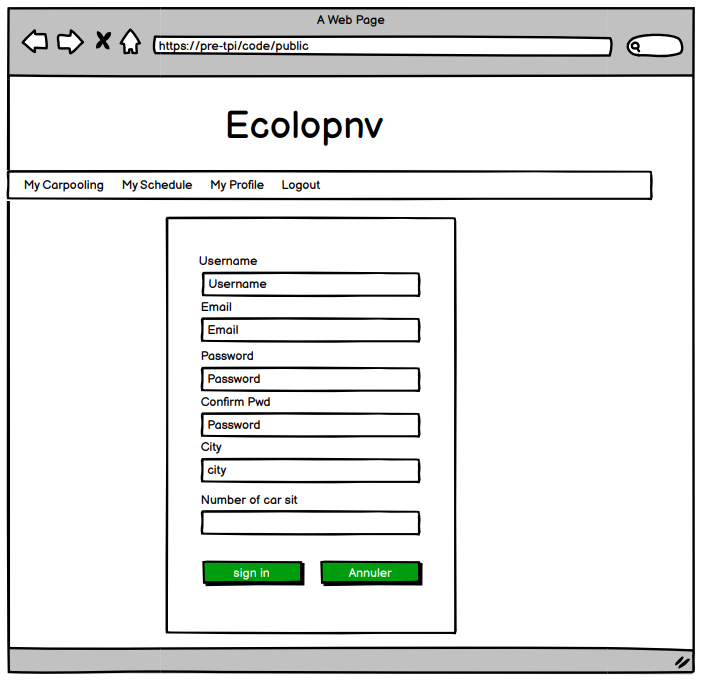
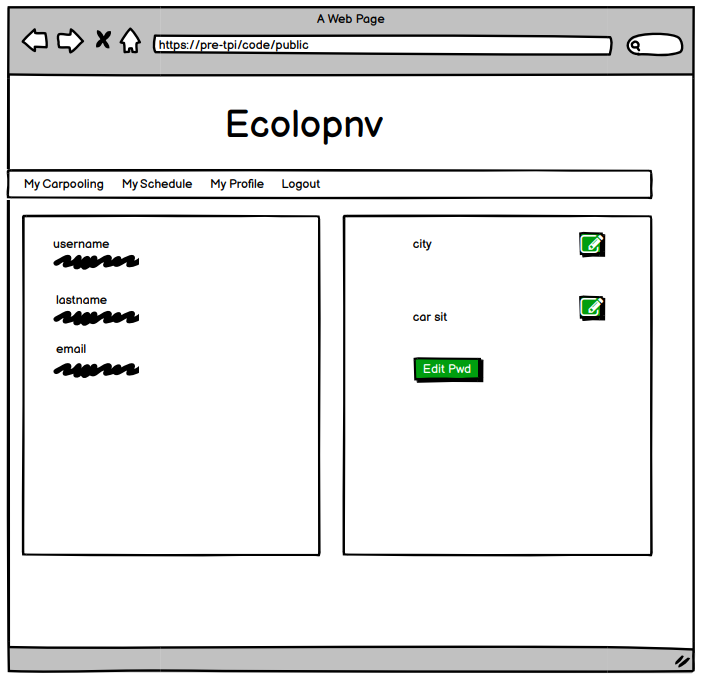
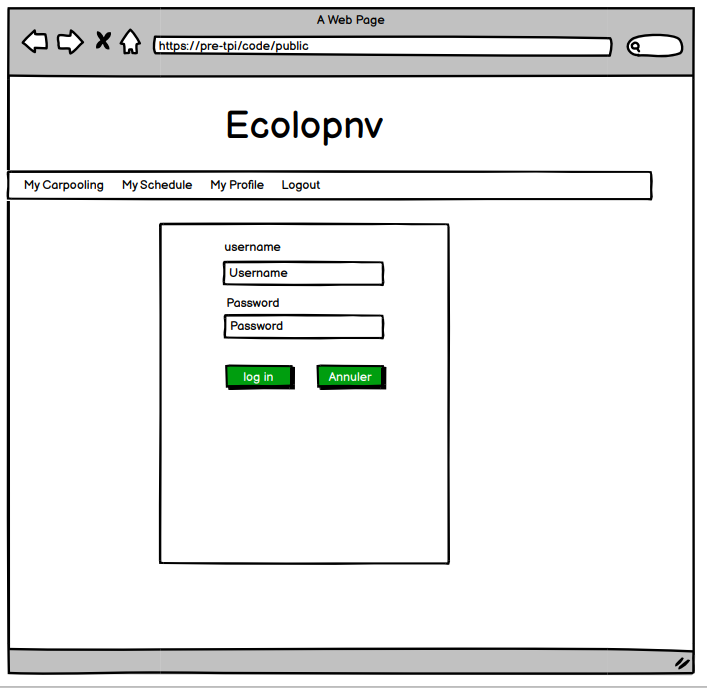
Voir guide d’installation

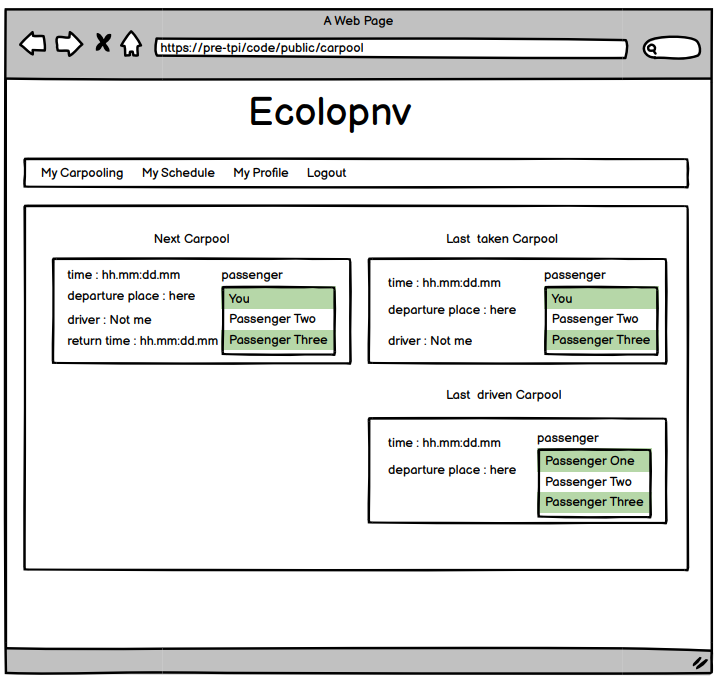
## Manuel d'Utilisation

Voir publication chez l’hébergeur

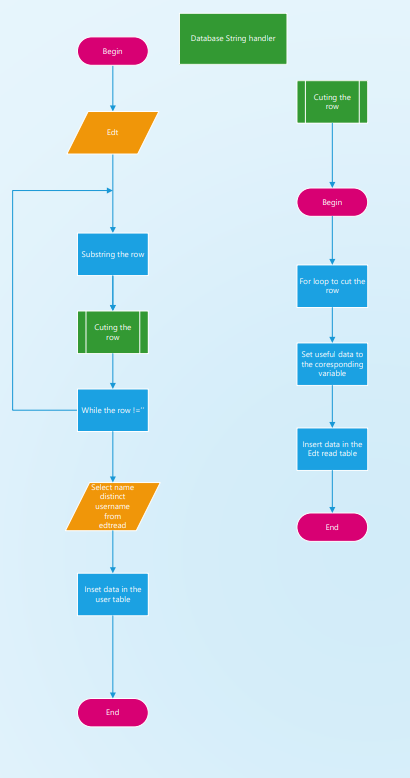
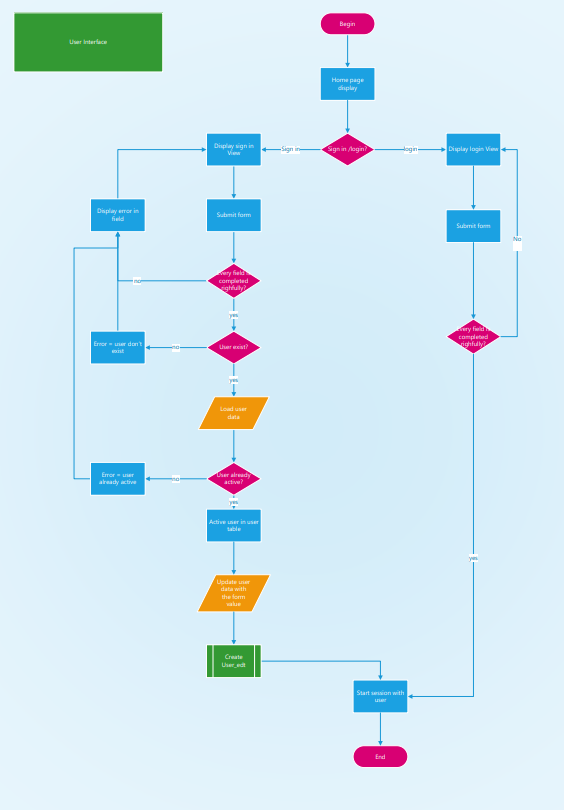
## Wireframe

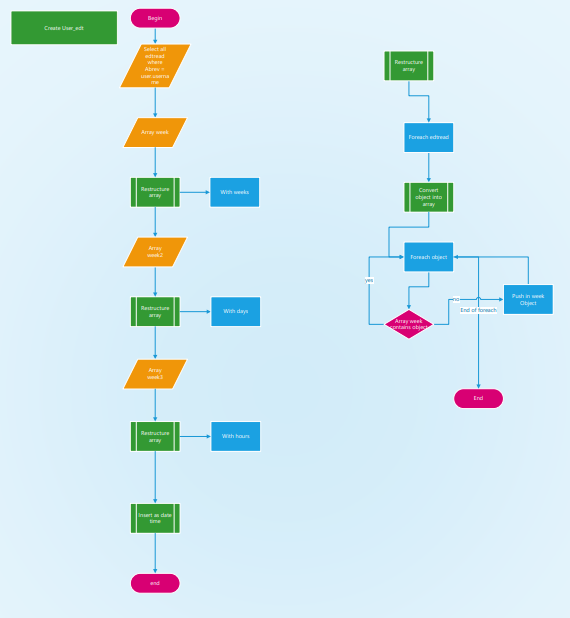


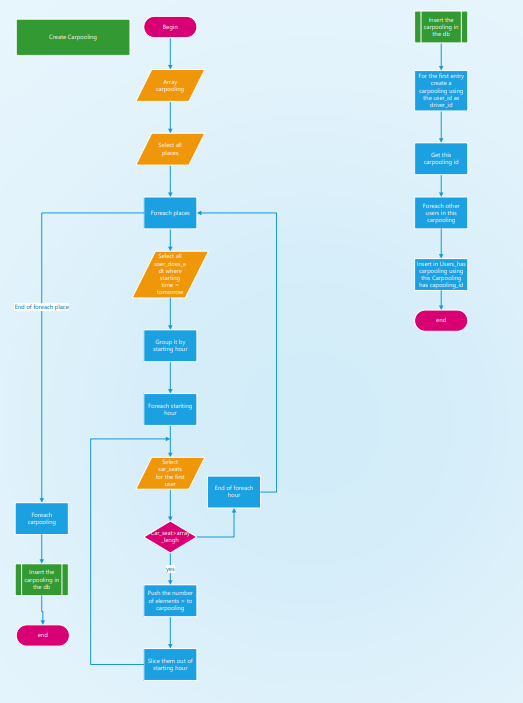




## Diagramme de flux







## Archives du projet

[https://cloud.icescrum.com/p/PRETPIVOIT/#/project](https://cloud.icescrum.com/p/PRETPIVOIT/%23/project)

[*https://github.com/PabloZubieta/Pre\_Tpi*](https://github.com/PabloZubieta/Pre_Tpi)