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

*Zubieta Pablo*

****

*SI-CA2a*

*06.03.23*

Table des matières

[1 Analyse préliminaire 4](#_Toc476150016)

[1.1 Introduction 4](#_Toc476150017)

[1.2 Organisation 4](#_Toc476150018)

[1.3 Objectifs 4](#_Toc476150019)

[1.4 Planification initiale 4](#_Toc476150020)

[2 Analyse 5](#_Toc476150021)

[2.1 Cahier des charges détaillé 5](#_Toc476150022)

[2.2 Définition de l’audience 5](#_Toc476150023)

[2.3 Analyse concurrentielle 5](#_Toc476150024)

[2.4 Cas d’utilisation 5](#_Toc476150025)

[2.5 Etude de faisabilité 5](#_Toc476150026)

[2.6 MCD 6](#_Toc476150027)

[2.7 Nom du site et du domaine 6](#_Toc476150028)

[2.8 Stratégie de test 6](#_Toc476150029)

[2.9 Budget initial 6](#_Toc476150030)

[2.10 Planification détaillée 6](#_Toc476150031)

[2.11 Historique 6](#_Toc476150032)

[3 Conception 7](#_Toc476150033)

[3.1 Analyse de l’environnement 7](#_Toc476150034)

[3.2 Détermination de l’arborescence du site et des rubriques 7](#_Toc476150035)

[3.3 Définition de la charte graphique 7](#_Toc476150036)

[3.4 Maquette graphique 7](#_Toc476150037)

[3.5 Conception de la Base de données 7](#_Toc476150038)

[3.6 Conception du Code 7](#_Toc476150039)

[3.7 Plugins et librairies 7](#_Toc476150040)

[3.8 Choix de la formule d’hébergement 7](#_Toc476150041)

[4 Réalisation 8](#_Toc476150042)

[4.1 Dossier de réalisation 8](#_Toc476150043)

[4.2 Description des tests effectués 8](#_Toc476150044)

[4.3 Erreurs restantes 8](#_Toc476150045)

[4.4 Dossier d'archivage 8](#_Toc476150046)

[5 Mise en service 9](#_Toc476150047)

[5.1 Rapport de mise en service 9](#_Toc476150048)

[5.2 Liste des documents fournis 9](#_Toc476150049)

[6 Conclusions 9](#_Toc476150050)

[7 Annexes 10](#_Toc476150051)

[7.1 Sources – Bibliographie 10](#_Toc476150052)

[7.2 Journal de bord de chaque participant 10](#_Toc476150053)

[7.3 Manuel d'Installation 10](#_Toc476150054)

[7.4 Manuel d'Utilisation 10](#_Toc476150055)

[7.5 Archives du projet 10](#_Toc476150056)

# 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

Voir en annexe

## Définition de l’audience

*Ce projet est conçu pour les enseignant 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

*Cas 1*

* *Wireframes pour Cas 1: Maquettes fonctionnelles (Balsamiq)*
* *Scénarios pour Cas 1 : Sous forme tabulaire : (Action -> Résultat)*

*Cas 2*

* *Wireframes pour Cas 2*
* *Scénarios pour Cas 2*

*Cas 3 …*

*Faire tous les use case va être long*

## 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

*Le MCD est dans sur le Github, lui et toutes ses itérations*

## 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. Voir icescrum

## Budget initial

C’est 0.00 CHF

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.

Reception 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

*Aucune pour l’instant*

# Conception

## Analyse de l’environnement

*Windows 10 family*

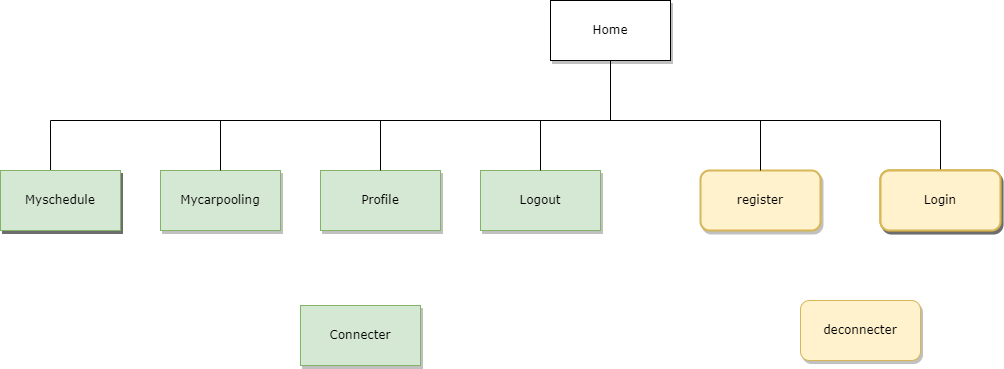
*Mariabd 10.10*

*Phpstorm 2022.2.1*

*Laravel9.0*

*Php8.1*

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



## Définition de la charte graphique

*Définir tout ce qui concerne le design, le choix des couleurs et des polices. (Justifier les choix !)*

*Pas encore defini*

## Maquette graphique

*Réalisation d’une maquette complète par adaptation ou non d’un modèle graphique existant.*

*Pas encore realisé*

## Conception de la Base de données

*Voir MLD sur Github*

## Conception du Code

*Diagrammes UML d’activité, diagramme d’état, diagramme de flux, ou pseudocode pour chaque cas d’utilisation.*

Ça va être très long à faire

## Plugins et librairies

*Etant donnée l’utilisation d’un Framework, il y a une tonne de librairie associer tel que Eloquent*

## 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

*Décrire la réalisation "physique" de votre projet*

* *les répertoires où le logiciel est installé*
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*
* *le numéro de version de votre produit.*
* *programmation et scripts: librairies externes, reconstruction du logiciel cible à partir des sources.*

*NOTE : Inutile d’inclure les listings des sources à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n’incluez que cette partie…*

## Description des tests effectués

*Pour chaque partie testée de votre projet, il faut décrire:*

* *les conditions exactes de chaque test*
* *les preuves de test (papier ou fichier)*
* *tests sans preuve: fournir au moins une description*

## Erreurs restantes

*S'il reste encore des erreurs:*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

## Dossier d'archivage

*Décrire de manière détaillée les archives du projet.*

***Attention: les documents de réalisation doivent permettre à une autre personne de maintenir et modifier votre projet sans votre aide !***

# Mise en service

## Rapport de mise en service

*Fournir une description:*

* *de la publication chez l’hébergeur*
* *de l'installation du projet chez le client*
* *des tests officiels effectués chez le client et/ou par le client.*
* *des erreurs répertoriées   
  - description détaillée   
  - conséquences pour le client  
  - actions envisagées.*

## Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

* *le rapport de projet*
* *le manuel d'Installation (en annexe)*
* *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
* *autres…*

# Conclusions

*Développez en tous cas les points suivants:*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

# Annexes

## 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

## Manuel d'Utilisation

## Archives du projet

<https://cloud.icescrum.com/p/PRETPIVOIT/#/taskBoard/211854/task/1079566>

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