|  |
| --- |
| Tpi  Site de génération de labyrinthe réalisables  Get Me Out |

*Zubieta Pablo*

****

*SI-CA2a*

*09.05.23*

Table des matières

[1 Analyse préliminaire 3](#_Toc134448914)

[1.1 Introduction 3](#_Toc134448915)

[1.2 Organisation 3](#_Toc134448916)

[1.3 Objectifs 3](#_Toc134448917)

[1.4 Planification initiale 3](#_Toc134448918)

[2 Analyse 4](#_Toc134448919)

[2.1 Cahier des charges détaillé 4](#_Toc134448920)

[2.2 Définition de l’audience 9](#_Toc134448921)

[2.3 Analyse concurrentielle 9](#_Toc134448922)

[2.4 Cas d’utilisation 9](#_Toc134448923)

[2.5 Etude de faisabilité 10](#_Toc134448924)

[2.6 MCD 11](#_Toc134448925)

[2.7 Nom du site et du domaine 11](#_Toc134448926)

[2.8 Stratégie de test 11](#_Toc134448927)

[2.9 Budget initial 11](#_Toc134448928)

[2.10 Planification détaillée 13](#_Toc134448929)

[2.11 Historique 14](#_Toc134448930)

[3 Conception 14](#_Toc134448931)

[3.1 Analyse de l’environnement 14](#_Toc134448932)

[3.2 Détermination de l’arborescence du site et des rubriques 15](#_Toc134448933)

[3.3 Définition de la charte graphique 15](#_Toc134448934)

[3.4 Maquette graphique 15](#_Toc134448935)

[3.5 Conception de la Base de données 16](#_Toc134448936)

[3.6 Conception du Code 16](#_Toc134448937)

[3.7 Plugins et librairies 16](#_Toc134448938)

[3.8 Choix de la formule d’hébergement 16](#_Toc134448939)

[4 Réalisation 17](#_Toc134448940)

[4.1 Dossier de réalisation 17](#_Toc134448941)

[4.2 Description des tests effectués 17](#_Toc134448942)

[4.3 Erreurs restantes 17](#_Toc134448943)

[4.4 Dossier d'archivage 18](#_Toc134448944)

[5 Mise en service 19](#_Toc134448945)

[5.1 Rapport de mise en service 19](#_Toc134448946)

[5.2 Liste des documents fournis 19](#_Toc134448947)

[6 Conclusions 19](#_Toc134448948)

[7 Annexes 20](#_Toc134448949)

[7.1 Résumé du rapport 20](#_Toc134448950)

[7.2 Sources – Bibliographie 20](#_Toc134448951)

[7.3 Manuel d'Installation 20](#_Toc134448952)

[7.4 Manuel d'Utilisation 20](#_Toc134448953)

[7.5 Wireframe 20](#_Toc134448954)

[7.6 Diagramme de flux 20](#_Toc134448955)

[7.7 Archives du projet 20](#_Toc134448956)

# Analyse préliminaire

## Introduction

Le projet Get Me Out est un site web de création de et de résolution de Labyrinthe. Ce site est créé avec un Framework PHP : Laravel.

Ayant une passion dévorante pour les mathématiques, ce projet est une très bonne manière de faire un peu de topologie. L’algorithmie étant un champ qui m’as toujours intéressé, c’est pourquoi j’ai choisi ce sujet. Je voulais me confronter à des tâches plus complexe que du CRUD.

## Organisation

|  |  |  |
| --- | --- | --- |
| Elève | Zubieta Pablo | [zubieta-rodriguez.pablo-fernando@cpnv.ch](mailto:zubieta-rodriguez.pablo-fernando@cpnv.ch) |
| Chef de projet | Benzonana Pascal | [benzonana.pascal@cpnv.ch](mailto:benzonana.pascal@cpnv.ch) |
| Expert 1 | Bernard Oberson | [oberson.bernard@gmail.com](mailto:oberson.bernard@gmail.com) |
| Expert 2 | Yves Bertino | [yves@bertino.ch](mailto:yves@bertino.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 gestion de ce projet sera Agile et donc la plateforme Icescrum sera utilisé pour porter les différentes story et tâches du projet.

Elle se divise en 4 sprint.

Le premier est la mise place de la documentation, et la création de l’interface utilisateur.

Le second est l’implémentation de l’interface de création manuel des labyrinthes.

Le troisième est l’implémentation l’algorithme de solution des labyrinthes.

Le dernier est l’implémentation de l’algorithme de création automatique de labyrinthe.

# 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.benzonan](mailto:pascal.benzonana@cpnv.ch)[a@cpnv.ch](mailto:a@cpnv.ch) | | : 076 230 23 13 | |
| **Expert 1** | Nom: **OBERSON** | | Prénom: Bernard | |
| : [oberson.bernard@gmail.com](mailto:oberson.bernard@gmail.com) | |  | |
| **Expert 2** | Nom: **BERTINO** | | Prénom: Yves | |
| : [yves@bertino.ch](mailto:yves@bertino.ch) | |  | |
| **Période de réalisation :** | Du lundi 2 mai 2023 à 8h00 au mardi 30 mai 2023 à 16h50 | | | |
| **Horaire de travail :** | Lundi 09h50-12h15 13h30-16h55 *Pentecôte 29 mai*  Mardi 08h00-12h15 13h30-16h55  Mercredi 08h00-12h15 -  Jeudi 08h00-12h15 13h30-16h55 *Examen Ascension 18 mai*  *ECG 1er juin matin*  Vendredi - - *Pont de l'Ascension 19 mai Toutes les demi-journées ont une pause obligatoire de 15 minutes, sauf si elles commencent à 9h50 ou si elles se terminent à 15h05.*  Voir en annexe en fin de document | | | |
| **Nombre d'heures :** | 90 heures | | | |
| **Planning** | Analyse : 20% | | | |
| Implémentation : 45% | | | |
| Tests : 10% | | | |
| Documentations : 25% | | | |
| **Présentation** | Dates retenues : 12 ou 13 juin 2023 | | | |

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 la i-CQ VD. 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

Site de génération de labyrinthes réalisables

#### SUJET

Le projet consiste à créer un site Web permettant de générer des labyrinthes solubles.

Les labyrinthes seront générés à partir d’un point central défini soit par l’utilisateur soit par l’application.

### MATÉRIEL ET LOGICIEL À DISPOSITION

1 ordinateur standard du CPNV avec Windows 10 professionnel et la suite Office et des crédits Azure

Les autres logiciels disponibles sont :

* Windows 10
* Suite Office
* Balsamiq
* MySQL Workbench
* PHPStorm
* Outils de gestion de projet : GitHub Project / Notion

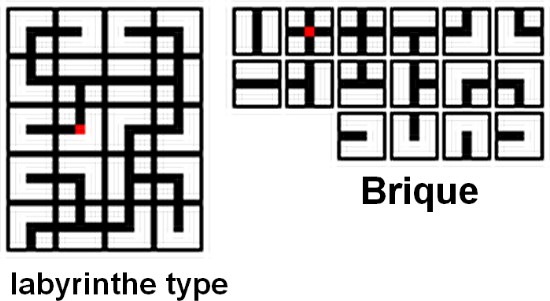
### PRÉREQUIS

Le candidat possède de bonnes connaissances programmation Web et modélisation (I-CT 101, 104, 105, 120, 133, 151) et sait mettre en œuvre les différents concepts nécessaires au développement d’application. La candidate a acquis déjà une expérience en programmation Web et a déjà travaillé avec une base de données MySQL.

### DESCRIPTIF DU PROJET

Le candidat devra identifier les besoins de création de différents modules pour le projet final, il devra pour s’organiser et visualiser la finalité du projet, créer une maquette du site et modéliser sa base de données. Une fois que cela sera fait, il pourra ensuite passer à la programmation et réalisation du projet.

Le site devra générer des labyrinthes solubles à partir d’un point central (point rouge) comme par exemple sur l’image ci-dessous. La sortie se trouve sur la brique qui touche le bord extérieur du carré/rectangle (cercle vert). La représentation des briques utilisables n’est pas exhaustive.



Dans un premier temps, il créera sa base de briques pour pouvoir dans un 2ème temps générer les labyrinthes.

#### **Génération manuelle :**

L’utilisateur après avoir choisi la taille du labyrinthe pourra déposer ses briques en indiquant les points d’entrée et de sortie du labyrinthe. L’application se chargera de vérifier que le labyrinthe proposé soit réalisable. Les tailles de labyrinthe exprimées en nombre de briques sont par exemple :

- 4 x 5

- 6 x 5

- 10x10

#### **Génération automatique :**

Après que l’utilisateur aura choisi sa taille de labyrinthe, l’application se chargera de le créer et de vérifier s’il est soluble. Tant qu’il ne l’est pas, l’application procède à des tentatives

### 7 LIVRABLES

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

* + 02.05.2023 : une planification initiale contenant la liste des sprints ainsi que les dates/heures des sprint reviews (confirmées avec le PO).
  + A chaque fin de sprint :
    - Un commit signifiant clairement le livrable pour le sprint
    - Une documentation mise à jour
    - Un déploiement du site quand la réalisation aura débuté
  + A la fin du TPI, le candidat livrera :
    - Les sources et les données sur le dépôt,
    - La documentation sur le dépôt,
    - Un CD ou clé USB avec les sources, la documentation et le journal à jour
  + Le journal de travail et le rapport devront être transmis par courriel deux fois par semaine.

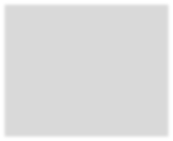
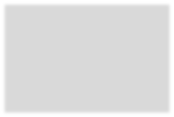
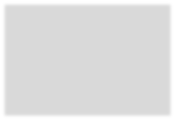
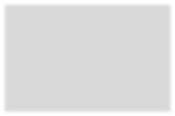
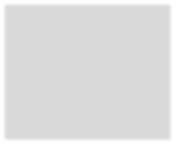
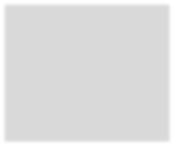
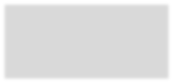
### 8 POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

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 sept points spécifiques suivants :

1. *Le site affiche les différentes briques possibles.*
2. *L’utilisateur peut choisir ses tailles de labyrinthes (min 4x4)*
3. *Ergonomie et facilité d'utilisation du produit (Bastien et Scapin)*
4. *L’application teste si le labyrinthe est réalisable.*
5. *L’utilisateur peut retrouver les labyrinthes qu’il a créés et ceux qu’il a résolus*
6. *L’utilisateur peut générer des labyrinthes manuellement*
7. *L’utilisateur peut générer des labyrinthes automatiquement*

9 HORAIRE DE TRAVAIL



## Définition de l’audience

Toutes les personnes voulant créer ou résoudre des labyrinthes.

## Analyse concurrentielle

Alors oui il existe une très grande quantité de site permettant la création de labyrinthe. Pour en citer quelques-uns :

* [*https://puzzlemaker.discoveryeducation.com/maze*](https://puzzlemaker.discoveryeducation.com/maze)
* [*https://www.mazegenerator.net/*](https://www.mazegenerator.net/)
* [*https://printablecreative.com/maze-generator*](https://printablecreative.com/maze-generator)

Ils sont avant tout orientés sur l’auto-génération permettant à l’utilisateur de choisir forme des bricks., taille, forme globale et même la difficulté. Mais je n’ai pas trop vu mise en avant les fonctionnalité d’interface d’édition ou l’interface de résolution. Ce qui rend ce cahier des charges très pertinents par rapport à la concurrence.

## Cas d’utilisation

Use case : l’utilisateur se connecte sur le site pour faire des labyrinthes

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 username | L’utilisateur possédant cet acronyme doit exister dans la base de données |  |
| L’utilisateur remplit le champ mot de passe. | Le mot passe doit correspondre à celui dans base de donnée |  |
| L’utilisateur clique sur le bouton valider | Tous les champs doivent être remplit | L’utilisateur en question est authentifié et redirigé sur la page d’accueil |

Extension : Les champ sont mal remplis ou le mot de passe ne corresponds pas.

|  |  |  |
| --- | --- | --- |
| 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 s’enregistre sur le site pour faire des labyrinthes

Scénario

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le register de la navbar |  | L’utilisateur est redirigé sur la page register |
| L’utilisateur remplit le champ username | Aucun autre utilisateur doit avoir le même username |  |
| L’utilisateur remplit le champ email | L’entrée doit être un email |  |
| 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 inséré dans la base de donnée, 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 |

## Etude de faisabilité

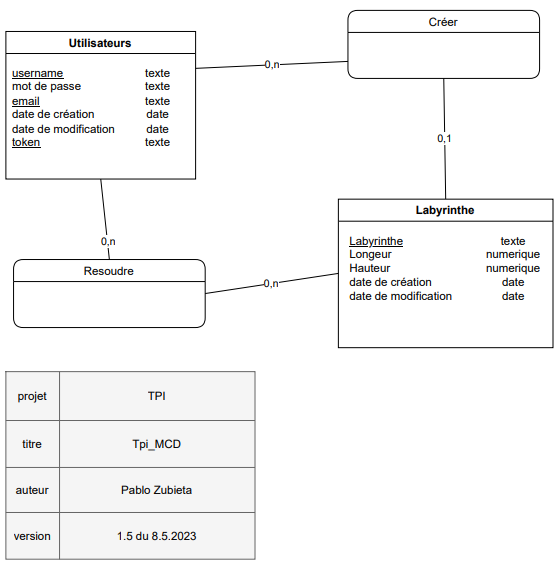
Il existe une infinité de manière de faire une application génératrice de labyrinthe. Utiliser Unity, une diversité de langage allant du JS au Python en passant par le Ruby, Avec des Framework ou en natif, Avec du sql ou du nosql. Ayant beaucoup apprécier le PHP c’est vers ce langage que je me suis orienté.

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.
* 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 que c’est le Framework que j’ai employé pour faire mon Pré-Tpi donc une technologie, je commence à bien comprendre.

## MCD



## Nom du site et du domaine

<http://labyrinthe.mycpnv.ch>   domaine louer par le CdP, même si GetMeOutof.mycpnv.ch aurait été beaucoup plus drôle.

## 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 |
| Temps de travail | la Planification contient l’intégralité des heures du projet, ni plus ni moins. |
| Repository distant | Il y a un Repository Git hébergé sur : https://github.com/PabloZubieta/Tpi |
| Repository local | Il y a un repository local a ce projet |
| Git flow | le projet utilise le framework git flow |
| Laravel | Le projet utilise le framework laravel pour son fonctionnement. |
| Creation de Wireframe | Pour  homepage  login  register  editeur de labyrinthe  resolution de labyrinthe  historique du joueur |
| Connecter | Given  Un utilisateur connecter  When  Il se trouve la page Home  Then  Sa barre de navigation affiche  Home  Résolution  Création  Historique  Logout |
| Deconnecter | Given  Un utilisateur déconnecter  When  Il se trouve la page Home  Then  Sa barre de navigation affiche  Home  Register  Log in |
| creation d'un utlisateur | Given  Un utilisateur quelconque Par exemple :  Tartempion  When  Il rentre son adresse mail et mot de passe 2 fois dans ce cas : Tartempion@cpnv.ch et Pa$$W0rd  Then  il est insérer dans la base de donnée et est connecter au site |
| diffèrent mot de passe | Given  Si deux mot de passe différents sont entrés  When  L’utilisateur valide  Then  une erreur indiquant que les mots de passe sont différents |
| utilisateur existant | Given  si un utilisateur existe déjà dans la base de donnée  When  l’utilisateur valide  Then  une erreur lui est afficher expliquant que cet utilisateur existe déjà |
| connexion réussite | Given  un utilisateur inscrit dans notre base de donnée Par exemple:  Tartempion  When  il utilise son mot de passe correcte dans ce cas: Pa$$W0rd  Then  il est connecter au site |
| mauvais mot de passe | Given  un utilisateur inscrit dans notre base de donnée Par exemple:  Tartempion  When  il utilise son mot de passe incorrecte dans ce cas: M0tdePa$$e  Then  un message d’erreur lui est affiché et il n’es pas connecter |
| champ vide | Given  si un utilisateur ne remplit pas l’un champ du formulaire  When  il essaie de valider sa connexion  Then  un message d’erreur lui est affiché et il n’es pas connecter |
| utilisateur inconnu | Given  un utilisateur non-inscrit dans notre base de donnée Par exemple:  Tartempognon  When  il utilise un mot de passe dans ce cas: Pa$$W0rd  Then  un message d’erreur lui est affiché et il n’es pas connecter |
|  |  |

## Budget initial

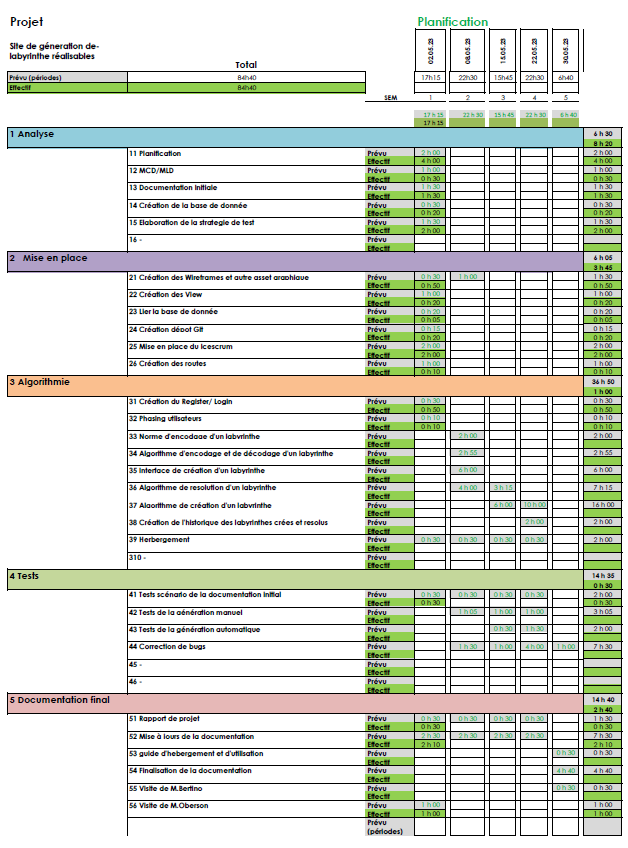
C’est 0.00 CHF

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

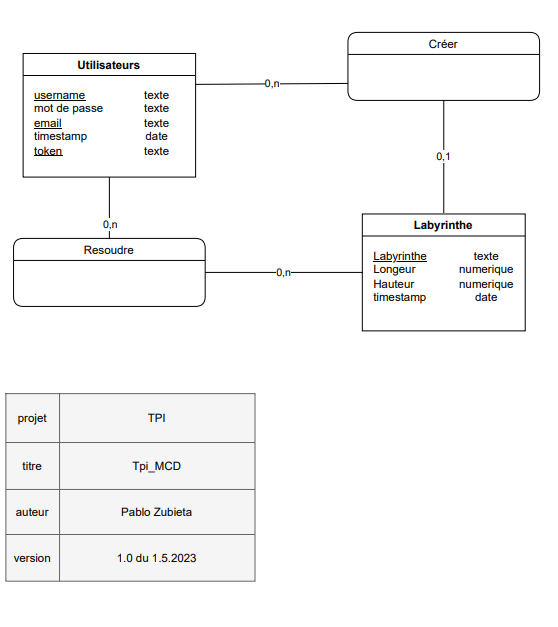
* C’est PHPstorm = 247 CHF
* Balsamiq = 89 CHF
* Total = 336 CHF

## Planification détaillée



## Historique

MCD première version du 04.05 obsolète depuis le 08.05



# 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

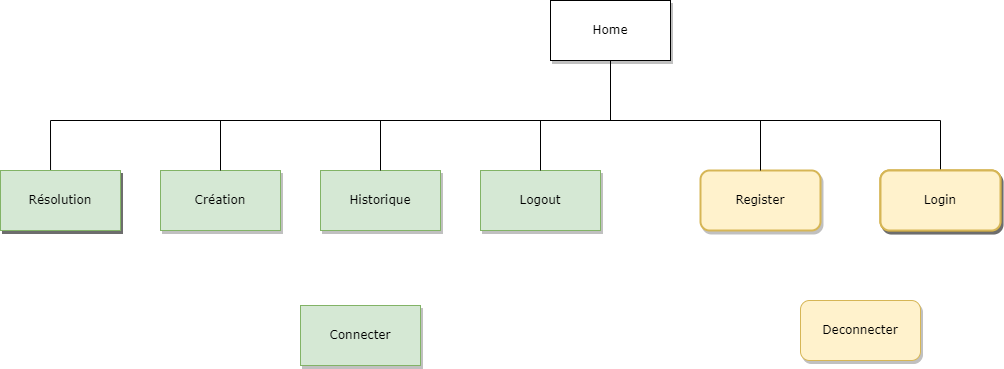
Hébergement

Php8.1

Apache/2.4.37

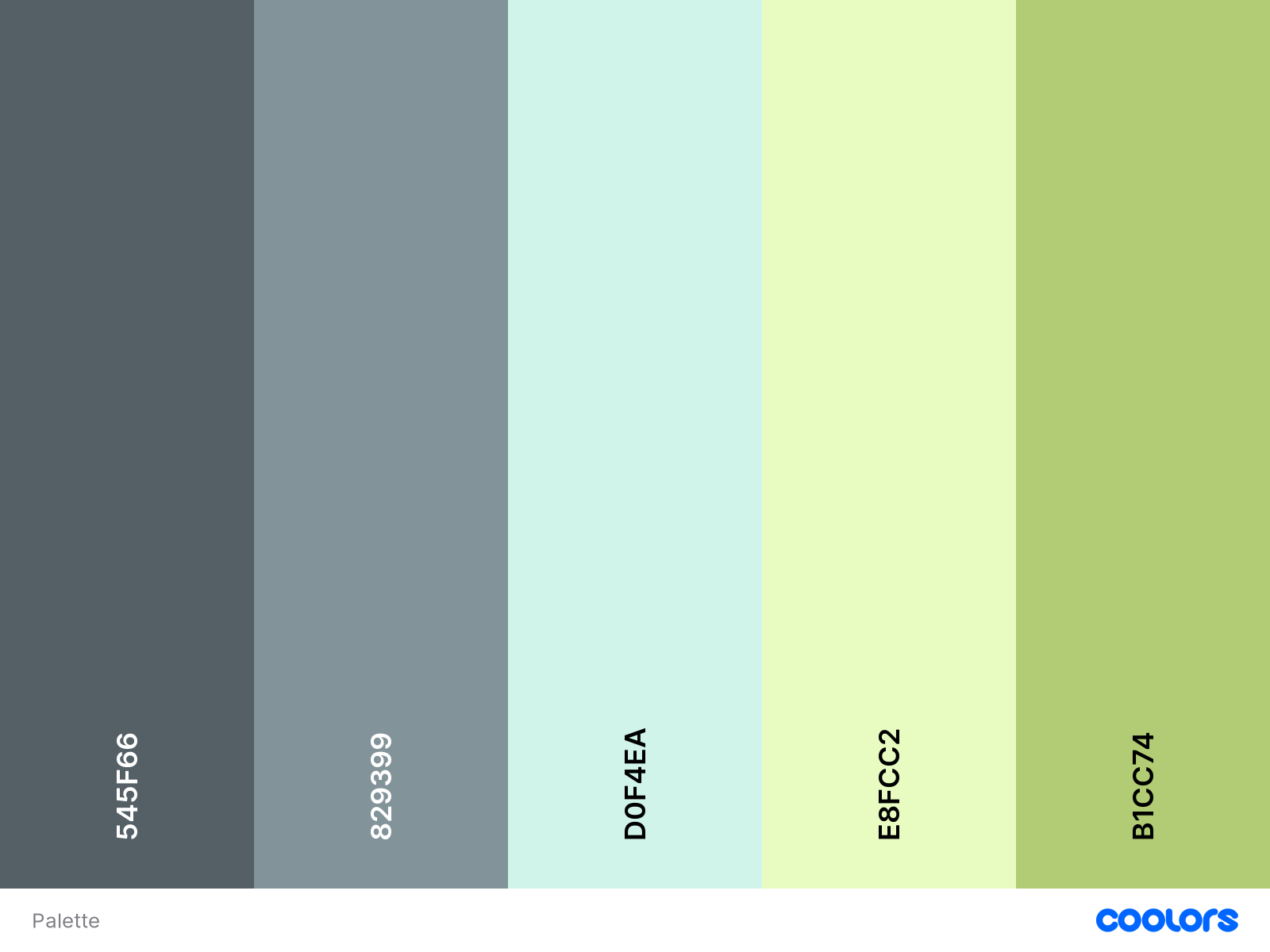
Mysql 8.0.32-cll-lve

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



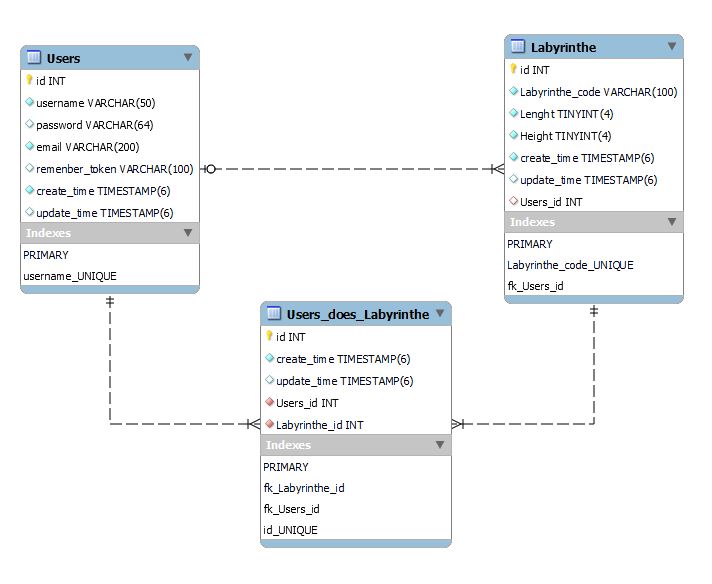
## Définition de la charte graphique

Pour ce qui est de la charte graphique. J’ai pris des couleurs s’axant autour du gris pour rappeler l’environnement informatique et le Vert pour faire écho au logo du Cpnv.



## Maquette graphique

## Conception de la Base de données



## Conception du Code

## 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, ou meme git-flow qui est mon framework Git.

## 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:\Tpi
2. Repository distant :
   * https://github.com/PabloZubieta/Tpi

Il y a une petite subtilité dans ce repository : c’est celle de l’utilisation de gitflow. Dans mon cas il y a 2 branche. La develop et la master la première me servant à faire tous mes commit de développement et la seconde à faire mes releases pour la sprint reviews et le déploiement sur l’hébergeur

### Dossier bases de données

|  |  |
| --- | --- |
| Nom | Description |
| MCD\_V2.pdf | Fichier contenant le MCD de la DB |
| MLD | Fichier contenant le MLD de la DB |
| CreationDB.sql | Script de création de la base de données |

### Dossier documentation

|  |  |
| --- | --- |
| Nom | Description |
| documentation.pdf | La documentation que vous êtes en train de lire |
| Planification.pdf | La planification |
| Timesheet.html | Journal de travail |
| CDC-PBA-Pablo-Zubieta\_Labyrinthe\_v1\_2 | Cahier des charges |

### Dossier wireframe

|  |  |
| --- | --- |
| Nom | Description |
| Tpi\_Zonning.pdf | Le zonning du projet |
| Tpi\_WF.pdf | Les Wireframe du projet |
| Arboweb.png | Arborescence web du projet |
| Palette.png | Palette graphique du projet |
| Asset graphique | Dossier contenant toutes les tuiles au format png |

### Dossier code

|  |  |
| --- | --- |
| Nom | Description |
| App/http/controller | Tous les fichier de Classe Controllers de ce site |
| UserController.php |
| Users\_does\_labyrinthe  Controller.php |
| LabyrintheController.php |
| App/Model | Tous les fichier de Classe Model de ce site |
| User.php |
| Labyrinthe.php |
| Users\_does\_labyrinthe |
| Route/web | Toutes les Routes de ce site |
| 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 |
| Resources/views | Toutes les Vues de ce site |
| layout.blade.php |
| home.blade.php |
| users/login.blade.php |
| users/register.blade.php |
| users/historique.blade.php |
| Maze/creation.blade.php |
| Maze/resolution.blade.php |

## Description des tests effectués

Listes des tests

|  |  |
| --- | --- |
| Nom | Etat |
| Temps de travail | Réussi |
| Repository distant | Réussi |
| Repository local | Réussi |
| Git flow | Réussi |
| Laravel | Réussi |
| Creation de Wireframe | Réussi |
| Connecter | Réussi |
| Deconnecter | Réussi |
| creation d'un utlisateur | Réussi |
| diffèrent mot de passe | Réussi |
| utilisateur existant | Réussi |
| connexion réussite | Réussi |
| mauvais mot de passe | Réussi |
| champ vide | Réussi |
| utilisateur inconnu | Réussi |
|  |  |

## Erreurs restantes

|  |  |  |  |
| --- | --- | --- | --- |
| **Erreur** | **Description** | **Conséquences** | **Action** |

## Dossier d'archivage

# Mise en service

## Rapport de mise en service

### Publication chez l’hébergeur

### 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*** |

*.*

## Liste des documents fournis

* Documentation

# Conclusions

# Annexes

## Résumé du rapport

## Sources – Bibliographie

|  |  |
| --- | --- |
| Sources | Consulté le |
| <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow> | 03.05.2023 |
| <http://tecfa.unige.ch/tecfa/teaching/LMRI41/CriteresB%26S.html> | 03.05.2023 |
| <https://laravel.com/docs/9.x/migrations> | 04.05.2023 |
| <https://www.youtube.com/watch?v=MYyJ4PuL4pY> | 04.05.2023 |
| <https://stackoverflow.com/questions/25970458/how-to-create-new-user-in-laravel> | 04.05.2023 |
| <https://www.php.net/manual/en/datetime.format.php> | 04.05.2023 |
| [*https://puzzlemaker.discoveryeducation.com/maze*](https://puzzlemaker.discoveryeducation.com/maze) | 08.05.2023 |
| [*https://www.mazegenerator.net/*](https://www.mazegenerator.net/) | 08.05.2023 |
| [*https://printablecreative.com/maze-generator*](https://printablecreative.com/maze-generator) | 08.05.2023 |
| [*https://coolors.co/545f66-829399-d0f4ea-e8fcc2-b1cc74*](https://coolors.co/545f66-829399-d0f4ea-e8fcc2-b1cc74) | 08.05.2023 |
| <https://fr.wikipedia.org/wiki/Base64> | 09.05.2023 |
| <https://devconnected.com/how-to-switch-branch-on-git/> | 09.05.2023 |
|  |  |
|  |  |
|  |  |
|  |  |

## Manuel d'Installation

Voir guide d’installation

## Manuel d'Utilisation

Voir publication chez l’hébergeur

## Wireframe

## Diagramme de flux

## Archives du projet

[*https://github.com/PabloZubieta/Tpi*](https://github.com/PabloZubieta/Tpi)