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

*Zubieta Pablo*

****

*SI-CA2a*

*31.05.23*

Table des matières

[1 Analyse préliminaire 3](#_Toc136414349)

[1.1 Introduction 3](#_Toc136414350)

[1.2 Organisation 3](#_Toc136414351)

[1.3 Objectifs 3](#_Toc136414352)

[1.4 Planification initiale 3](#_Toc136414353)

[2 Analyse 4](#_Toc136414354)

[2.1 Cahier des charges détaillé 4](#_Toc136414355)

[2.2 Définition de l’audience 9](#_Toc136414356)

[2.3 Analyse concurrentielle 9](#_Toc136414357)

[2.4 Cas d’utilisation 9](#_Toc136414358)

[2.5 Etude de faisabilité 12](#_Toc136414359)

[2.6 MCD 13](#_Toc136414360)

[2.7 Nom du site et du domaine 13](#_Toc136414361)

[2.8 Stratégie de test 13](#_Toc136414362)

[2.9 Encodage topologique 21](#_Toc136414363)

[2.10 Budget initial 21](#_Toc136414364)

[2.11 Planification détaillée 22](#_Toc136414365)

[2.12 Historique 23](#_Toc136414366)

[3 Conception 24](#_Toc136414367)

[3.1 Analyse de l’environnement 24](#_Toc136414368)

[3.2 Détermination de l’arborescence du site et des rubriques 24](#_Toc136414369)

[3.3 Définition de la charte graphique 24](#_Toc136414370)

[3.4 Maquette graphique 25](#_Toc136414371)

[3.5 Conception de la Base de données 26](#_Toc136414372)

[3.6 Conception du Code 26](#_Toc136414373)

[3.7 Plugins et librairies 27](#_Toc136414374)

[3.8 Choix de la formule d’hébergement 27](#_Toc136414375)

[4 Réalisation 28](#_Toc136414376)

[4.1 Dossier de réalisation 28](#_Toc136414377)

[4.2 Description des tests effectués 29](#_Toc136414378)

[4.3 Erreurs restantes 32](#_Toc136414379)

[5 Mise en service 33](#_Toc136414380)

[5.1 Rapport de mise en service 33](#_Toc136414381)

[5.2 Liste des documents fournis 33](#_Toc136414382)

[6 Conclusions 34](#_Toc136414383)

[7 Annexes 35](#_Toc136414384)

[7.1 Résumé du rapport 35](#_Toc136414385)

[7.2 Glossaire 36](#_Toc136414386)

[7.3 Sources – Bibliographie 36](#_Toc136414387)

[7.4 Manuel d'Installation 37](#_Toc136414388)

[7.5 Manuel d'Utilisation 37](#_Toc136414389)

[7.6 Table des illustrations 37](#_Toc136414390)

[7.7 Wireframes 38](#_Toc136414391)

[7.8 Archives du projet 40](#_Toc136414392)

[7.9 Journal de bord 41](#_Toc136414393)

# 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 complexes 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 sprints.

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

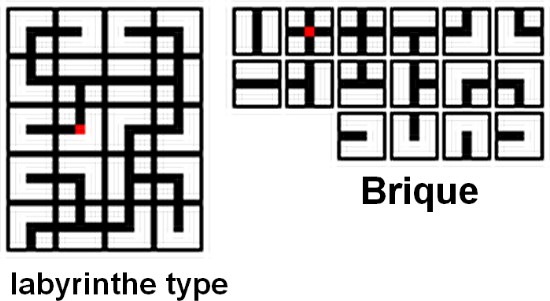
### PRÉREQUIS

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.

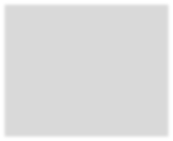
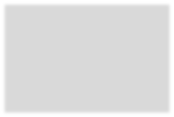
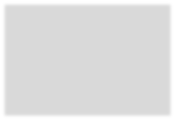
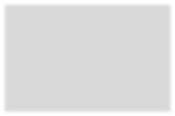
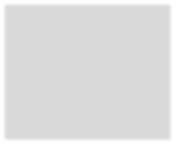
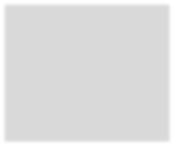
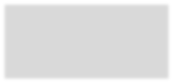
### 8 POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

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 fonctionnalités 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 données |  |
| 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 données, 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 veux créer son propre labyrinthe

Scénario

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le *création* de la navbar |  | L’utilisateur est redirigé sur la page création |
| L’utilisateur remplit le champ de longueur et hauteur | Les deux champs doivent être supérieur à 3 et inférieur à 11 |  |
| L’utilisateur clique sur crée la grille |  | Un labyrinthe vide de taille choisir est généré |
| L’utilisateur choisi son point de départ |  | Le point de départ est encadré en rouge |
| L’utilisateur choisi son point de d’arrivé | Le point d’arrivé doit être en bordure du Labyrinthe et ne peux pas être le point de départ | Le point de départ est encadré en bleu |
| L’utilisateur saisi une tuile et la place dans une des cases de la grille | La tuile du point d’arrivé ne peut pas être un cul-de-sac | La tuile est affichée dans la case en question |
| L’utilisateur valide sa grille | Le labyrinthe doit être soluble | L’utilisateur est redirigé sur la page résolution avec son labyrinthe |

Extension : L’utilisateur n’est pas connecté

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur insère l’url https://labyrinthe.my  cpnv.ch/creation |  | Il est redirigé sur la page login |

Extension : Les champ longueur et hauteur sont mal remplis

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le bouton créer la grille |  | Une erreur est affichée indiquant le problème |

Extension : Remplacement de tuile

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur place une tuile dans une place déjà occuper par une autre | La tuile doit être valide | L’ancienne tuile est remplacée par la nouvelle |

Extension : Reset

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le bouton reset |  | La grille disparait et le champ de configuration de la taille du labyrinthe réapparaisse |

Extension : Tuile incorrecte

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur saisi une tuile de type cul-de-sac pour la placer sur la sorti |  | Rien ne se passe |

Use case : l’utilisateur consulte son historique de labyrinthe

Scénario

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le *historique* de la navbar |  | L’utilisateur est redirigé sur la page historique |
| Clique sur un des lien des Labyrinthes de son historique |  | L’utilisateur est redirigé sur la page résolution avec le labyrinthe en question |

Use case : l’utilisateur veux résoudre un labyrinthe

Scénario

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur clique sur le *résolution* de la navbar |  | Un petit formulaire demandant la taille du labyrinthe apparait. |
| L’utilisateur valide le formulaire | La taille minimale du labyrinthe est de 4 et celle maximal de 10 | L’utilisateur est redirigé sur la page résolution avec le labyrinthe en question |

Extension : labyrinthe trop grand ou trop petit

|  |  |  |
| --- | --- | --- |
| Action | Conditions particulières | Réaction |
| L’utilisateur choisi une taille de labyrinthe non conforme |  | Le bouton valider est désactivé. |

## 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

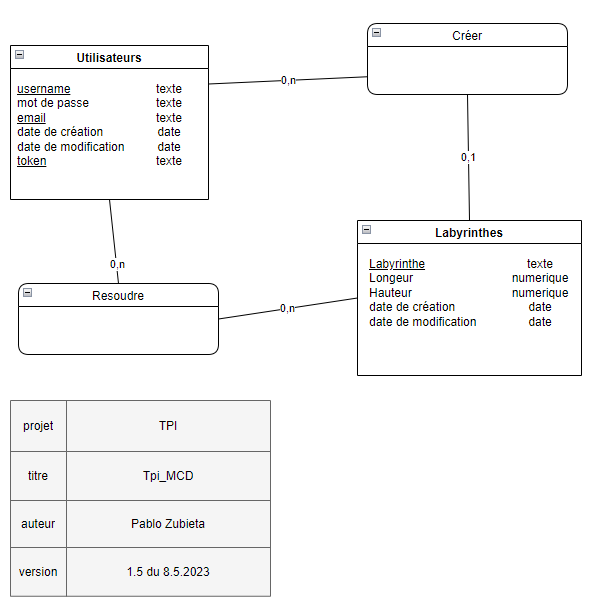


Figure 1 MCD du projet

Voici le MCD du projet :

On peut constater qu’il n’est pas très complexe.

Une table utilisateur et une table labyrinthe lier par les associations que ces deux entités peuvent avoir. Il y a cependant une chose remarquable c’est la traduction du labyrinthe en texte. C’est en effet une convention que j’ai crée pour transcrire un labyrinthe de fait de tuile carrée en une string et ainsi stocker ce qui autrement serais un array bi-dimensionnel.  
La convention se trouve en annexe.

## Nom du site et du domaine

<http://labyrinthe.mycpnv.ch> domaine loué 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

* Mise en place de l’environnement.

1. Planification

|  |  |
| --- | --- |
| Nom | Description |
| Temps de travail | La planification contient l’intégralité des heures du projet, ni plus ni moins. |

1. Base de données

|  |  |
| --- | --- |
| Nom | Description |
| MCD | Le projet dispose d’un MCD |
| MLD | Le projet dispose d’un MLD |
| Création de la base donnée | Le script de création de base de données est conforme au MLD |

1. Repository Git

|  |  |
| --- | --- |
| Nom | Description |
| 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 |

1. Hébergement

|  |  |
| --- | --- |
| Nom | Description |
| Accès | Je peux accéder à mon application via :https://labyrinthe.mycpnv.ch/ |
| Pas de régression | Il n’y a pas de régression entre mon projet git et sa version héberger |

1. Laravel

|  |  |
| --- | --- |
| Nom | Description |
| Laravel | Le projet utilise le Framework Laravel pour son fonctionnement. |

1. Wireframe

|  |  |
| --- | --- |
| Nom | Description |
| Création de Wireframe | Pour  Homepage  Login  Register  Éditeur de labyrinthe  Résolution de labyrinthe  Historique du joueur |

* Interface utilisateur

1. Views

|  |  |
| --- | --- |
| Nom | Description |
| Views | Le dossier views contient  Layout  Home page  Login  Register  Historique  Labyrinthe création  Labyrinthe résolution |

1. Register

|  |  |
| --- | --- |
| Nom | Description |
| Création d'un utilisateur | Un utilisateur quelconque Par exemple :  Tartempion  Quand il rentre son adresse mail et mot de passe 2 fois dans ce cas : Tartempion@cpnv.ch et Pa$$W0rd  Alors il est inséré dans la base de données et est connecté au site |
| Diffèrent mot de passe | Si deux mots de passe différents sont entrés  Quand l’utilisateur valide  Alors une erreur indiquant que les mots de passe sont différents |
| Utilisateur existant | Si un utilisateur existe déjà dans la base de données  Quand l’utilisateur valide  Alors une erreur lui est afficher expliquant que cet utilisateur existe déjà |

1. Login

|  |  |
| --- | --- |
| Nom | Description |
| Connexion réussite | Un utilisateur inscrit dans notre base de données Par exemple : Tartempion  Quand il utilise son mot de passe correcte dans ce cas : Pa$$W0rd  Alors il est connecté au site |
| Mauvais mot de passe | Un utilisateur inscrit dans notre base de données Par exemple : Tartempion  Quand il utilise son mot de passe incorrecte dans ce cas : M0tdePa$$e  Alors un message d’erreur lui est affiché et il n’est pas connecter |
| Champ vide | Si un utilisateur ne remplit pas l’un champ du formulaire  Quand il essaie de valider sa connexion  Alors un message d’erreur lui est affiché et il n’est pas connecté |
| Utilisateur inconnu | Un utilisateur non-inscrit dans notre base de données Par exemple : Tartempognon  Quand il utilise un mot de passe dans ce cas : Pa$$W0rd  Alors un message d’erreur lui est affiché et il n’est pas connecté |

1. Routes

|  |  |
| --- | --- |
| Nom | Description |
| Routes Get | Créer les routes Get suivantes :  Homepage  Login  Register  Historique  Création |
| Routes Post | Créer les routes Post suivantes :  Log  Create  Logout  Check  Labyrinthe  Résolution |
| Phasing des routes | Les routes suivantes sont inaccessibles aux utilisateurs non connectés :  Historique  Création  Résolution  Log  Create  Logout  Check  Labyrinthe |

1. Phasing utilisateur

|  |  |
| --- | --- |
| Nom | Description |
| Connecter | Un utilisateur connecté  Quand il se trouve la page Home  Alors sa barre de navigation affiche  Home  Résolution  Création  Historique  Logout |
| Déconnecter | Un utilisateur déconnecté  Quand il se trouve la page Home  Alors sa barre de navigation affiche  Home  Register  Log in |

1. Historique de labyrinthe

|  |  |
| --- | --- |
| Nom | Description |
| Information personnelle | Étant donné un utilisateur connecté  Quand il est sur la page historique  Alors son nom d'utilisateur et son adresse mail sont affiché |
| Liste des labyrinthes résolus | Étant donné un utilisateur connecté  Quand il est sur la page historique  Alors il y a une liste de tous les labyrinthes qu'il a résolu avec la date en question, la taille et le code du labyrinthe |
| Liste des labyrinthes crées | Étant donné un utilisateur connecté  Quand il est sur la page historique  Alors il y a une liste de tous les labyrinthes qu'il a créé avec la date en question, la taille et le code du labyrinthe |
| Lien labyrinthe | Étant donné la liste du lien sur la page historique d'un utilisateur.  Quand il clique sur un des labyrinthes des dites listes.  Alors il est redirigé sur la page résolution avec le labyrinthe en question. |

1. Interface de création

|  |  |
| --- | --- |
| Nom | Description |
| Tuile de labyrinthe | il y a 48 tuile de labyrinthe |
| Taille du labyrinthe | Etant donné un utilisateur connecté.  Quand il arrive sur la page, la taille du labyrinthe lui est demandé : l’utilisateur entre 5 et 5.  Alors un tableau au proportion 5x5 est généré sur sa droite. |
| Liste des labyrinthes crées | Étant donné un utilisateur connecté.  Quand il est sur la page historique.  Alors il y a une liste de tous les labyrinthes qu'il a créé avec la date en question, la taille et le code du labyrinthe. |
| Donnée inconforme | Etant donné un utilisateur connecté.  Quand il arrive sur la page, la taille du labyrinthe lui est demandé : l’utilisateur entre 2 et 2.  Alors un message d’erreur lui indique que le labyrinthe n’est pas conforme. |
| Tuiles | Etant donné un utilisateur connecté.  Quand il a choisi son point de départ et de fin.  Alors, à sa gauche, l’utilisateur dispose de toutes les tuiles possibles. |
| Début et fin | Etant donné un utilisateur connecté  Après avoir saisi la taille du labyrinthe.  Alors il peut choisir sur le tableau de droite la case de départ et la case de fin du labyrinthe. |
| Reset | Etant donné un utilisateur connecté  Avec une grille contenant des tuiles.  Quand il clique sur le bouton reset.  Alors sa grille est réinitialisée et il doit ressaisir le début et fin de son labyrinthe. |
| Drag and drop | Sur une grille 5x5  Étant donné une tuile de forme SE.  Quand elle est saisie et déplacée sur la case 3x2 vide.  Alors elle est dupliquée dans la case en question. |
| Mauvaise fin | Sur une grille 5x5.  Quand l’utilisateur choisi comme point de fin la case 2x3.  Alors l’application relève une erreur et demande à l’utilisateur d’en choisir une autre. |
| Valider | Etant donné un utilisateur connecté  Avec une grille complète.  Quand il clique sur le bouton valider.  Alors sa grille est contrôlée et est insérer dans la base de données. |

* Labyrinthe

1. Norme d’encodage

|  |  |
| --- | --- |
| Nom | Description |
| Unicité | Toutes les tuiles sont associées à une valeur unique. |
| Encodage | La valeur d’une tuile a un encodage à taille constant. |
| Topologie | La valeur associée à une tuile permet de déduire ses propriétés topologiques |

1. Encoder et décoder un labyrinthe

|  |  |
| --- | --- |
| Nom | Description |
| Encoder | Étant donnée un labyrinthe 2x2  Contenant les tuiles [[SENW, WN], [SE, W]]  Commençant en 2x2 et se terminant en 1x1.  Quand il est inséré dans la base de données.  Alors son code dans la base de données vJDY et taille est respectivement 2 de longueur et 2 de hauteur. |
| Décoder | Un labyrinthe dans la base de données avec le code vJDY : De longueur 2 et hauteur 2.  Quand il est importé dans la page web.  Alors il est représenté par le array [[47,9], [3,24]]. |

1. Algorithme de résolution

|  |  |  |
| --- | --- | --- |
| Nom | Description | Labyrinthe |
| Ligne réussi | Le labyrinthe du fichier Line.  Quand il est validé.  Alors il est redirigé sur la résolution avec ledit labyrinthe. |  |
| Ligne échec | Le labyrinthe du fichier Noline.  Quand il est validé.  Alors il retourne sur la page avec un message insoluble. |  |
| Cul de sac réussi | Le labyrinthe du fichier Deadend  Quand il est validé  Alors il est redirigé sur la résolution avec ledit labyrinthe |  |
| Cul de sac échec | Le labyrinthe du fichier NoDeadend  Quand il est validé  Alors il retourne sur la page avec un message insoluble |  |
| Labyrinthe complexe réussi | Le labyrinthe du fichier Complexe  Quand il est validé  Alors il est redirigé sur la résolution avec ledit labyrinthe |  |
| Labyrinthe complexe échec | Le labyrinthe du fichier NoComplexe  Quand il est validé  Alors il retourne sur la page avec un message insoluble |  |
| Loop réussi | Le labyrinthe du fichier Loop  Quand il est validé  Alors il est redirigé sur la résolution avec ledit labyrinthe |  |
| Loop échec | Le labyrinthe du fichier Noloop  Quand il est validé  Alors il retourne sur la page avec un message insoluble |  |
| Loop extérieur réussi | Le labyrinthe du fichier Outerloop  Quand il est validé  Alors il est redirigé sur la résolution avec ledit labyrinthe |  |
| Loop extérieur échec | Le labyrinthe du fichier Noouterloop  Quand il est validé  Alors il retourne sur la page avec un message insoluble |  |
| Mauvais chemin réussi | Le labyrinthe du fichier Wrongexit  Quand il est validé  Alors il est redirigé sur la résolution avec ledit labyrinthe |  |
| Mauvais chemin échec | Le labyrinthe du fichier Nowrongexit  Quand il est validé  Alors il retourne sur la page avec un message insoluble |  |

1. Algorithme de création

|  |  |
| --- | --- |
| Nom | Description |
| Entrée/sortie dissemblables | Étant donné un utilisateur  Quand il va sur la page résolution  Alors sur un échantillon de 12 labyrinthes générés 10 ont des entrées dissemblables |
| Labyrinthes dissemblables | Étant donné un utilisateur  Quand il va sur la page résolution  Alors sur un échantillon de 12 labyrinthes générés 10 ont des formes et morphologies différentes |

## Encodage topologique

Pour faire correspondre les propriétés topologiques à leur valeur numérique. J’ai choisi de créer un encodage se basant sur du binaire

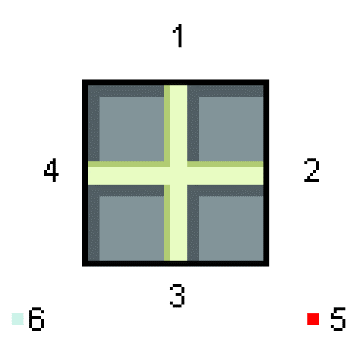


Figure 2 encodage topologique

En effet la valeur d’une tuile est comprise entre 0 et 47. Lorsque ce nombre est transcrit en binaire alors la valeur du premier bit décrit l’embranchement du haut. Sa valeur est de 1 le chemin du haut existe sinon non.

Le 5ème bit décrit si c’est le point de départ et le 6ème le point d’arrivé.

Donc par exemple la tuile 14 est la tuile T car la traduction de 001110.

## 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

Une image contenant texte, capture d’écran, Parallèle, ligne

Description générée automatiquement

Figure 3 Planification détaillée

## Historique

MCD première version du 04.05 obsolète depuis le 08.05.

A cause des champs timestamps qui ne sont pas conforme.

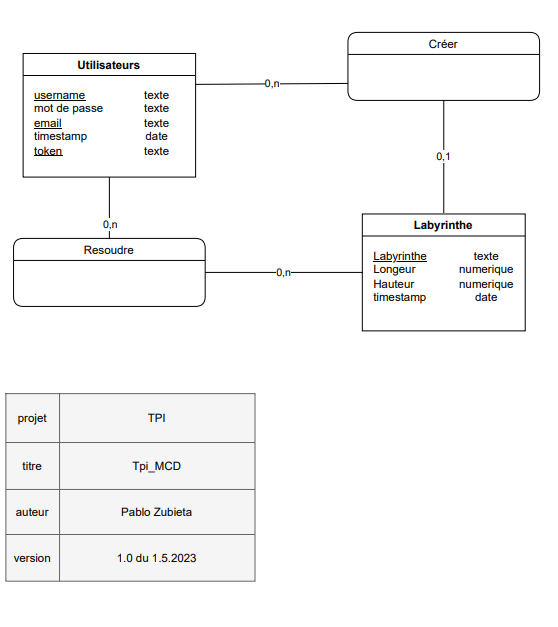


Figure 4 MCD\_v1

MCL première version du 04.05 obsolète depuis le 16.05

Pour des questions d’unicité sur la clé code du labyrinthe. Plus de précision dans la section erreurs restantes.

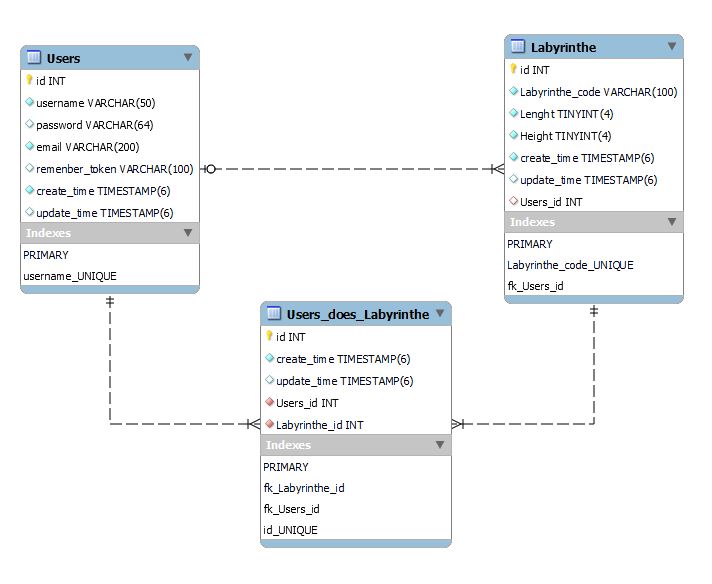


Figure 5 MLD\_v1

# Conception

## Analyse de l’environnement

Environnement 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

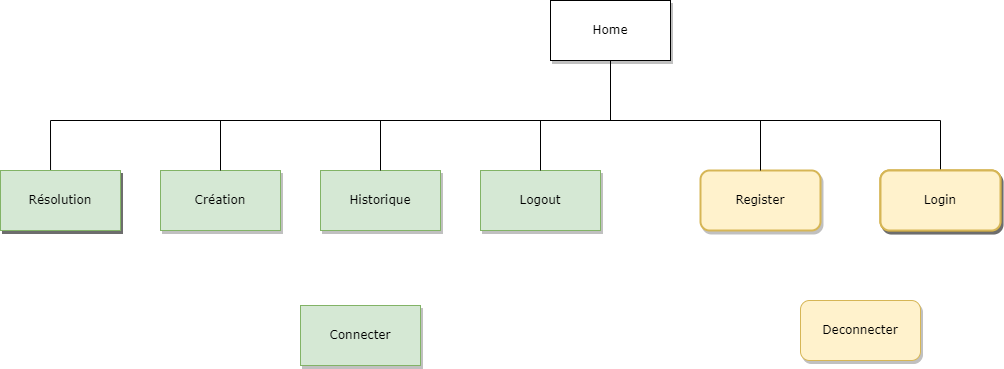


Figure 6 Arborescence du site

## 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.

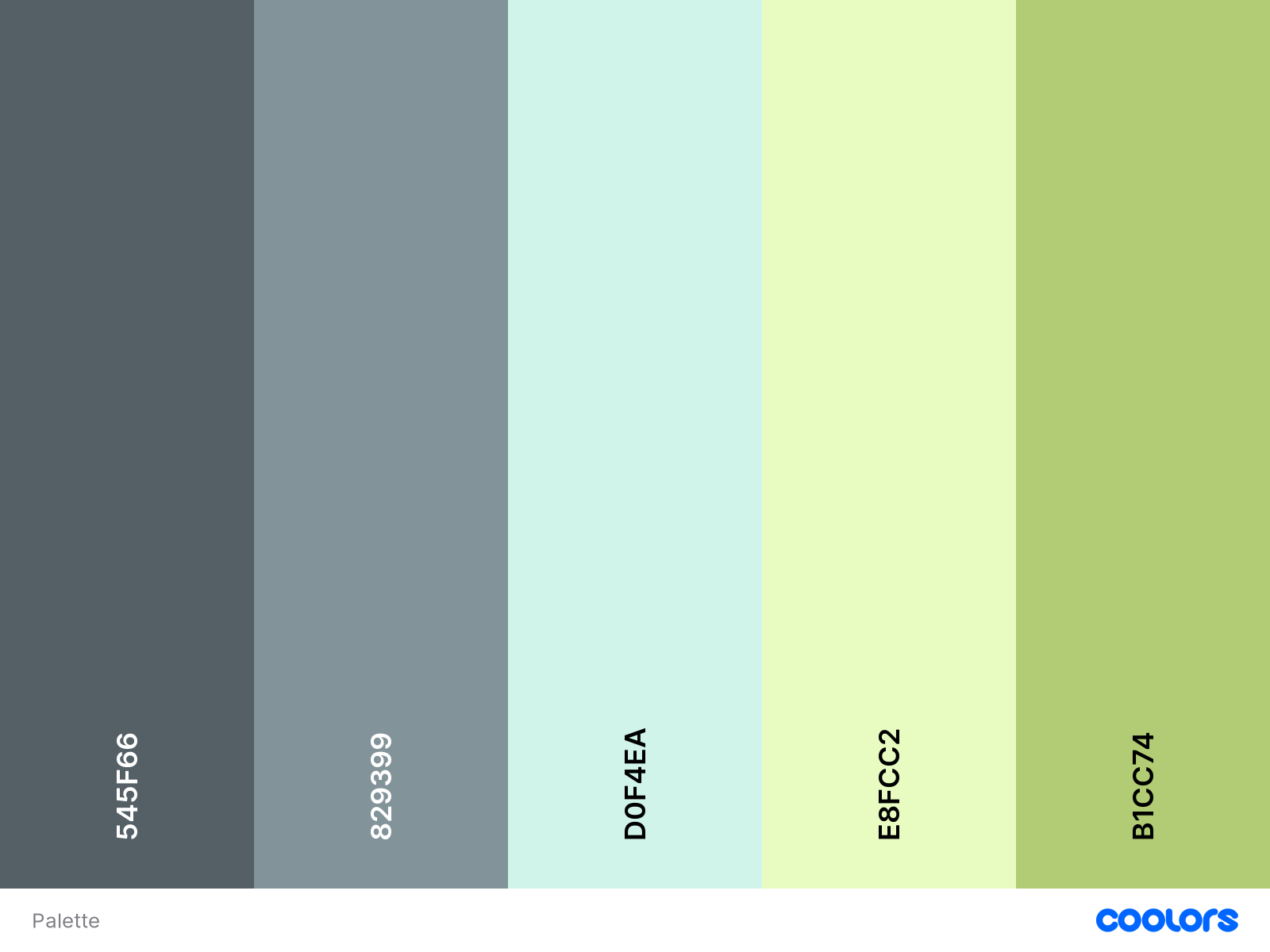


Figure 7 Palette de couleur

## Maquette graphique

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

La homepage :

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Figure 8 Homepage du site

Interface de création :

Une image contenant texte, capture d’écran, Rectangle, Parallèle

Description générée automatiquement

Figure 9 interface de création du site

## Conception de la Base de données

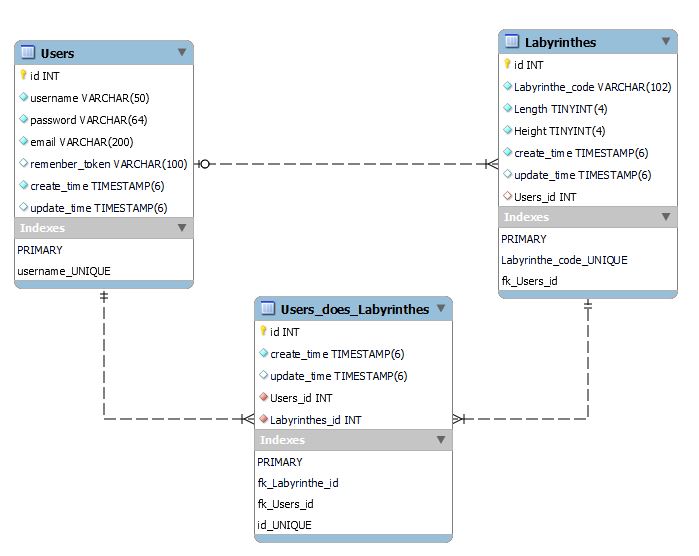


Figure 10 MLD

Le stockage du labyrinthe sous forme de chaine de caractère est possible grâce à un encodage en binaire. Il transforme les propriétés topologies en valeur numérique, elle-même associer à un caractère suivant la norme de la base64.

## Conception du Code

Dans ce projet il y a deux algorithmes fondamentaux et intéressants celui de vérification et celui de création. Et vous les voilà décrit avec du pseudocode.

Vérification de labyrinthe :

Créer 2 tableau de même dimension que le labyrinthe

Lire le code du labyrinthe transmis par la variable post, caractère par caractère

Transformer le caractère en une valeur numérique via la norme d’encodage

Insérer la valeur trouvée dans le premier tableau

Et zéro dans le deuxième tableau

Si valeur de trouvée comprise entre 16 et 31 cette case est le point de départ

Si la valeur est plus grande que 31 c’est le point d’arrivé.

Une fois l’intégralité du code parcouru.

Définissez le tableau position par les coordonnées de votre point de départ

A partir de position valeur se trouvant dans le premier tableau

Sa lecture binaire donnera ses chemins disponibles

Essayer le premier chemin dans le sens horaire

Soustraire à la cellule le chemin emprunter

Jusqu’à trouver une cellule sans nouveau chemin ou la sortie

Si une cellule sans chemin est trouvée

Retourner à la précédente cellule avec chemin non emprunter

Si la sortie est trouvé le labyrinthe à une solution.

Si plus aucun chemin n’est disponible le labyrinthe n’a pas de solution

Les erreurs viennent forcement de cette portion de code qui est dans la fonction create du fichier Labyrinthe Controller. C’est probablement un problème de condition mal établi.

Algorithme de création :

Créer 2 tableau de aux dimensions choisi par l’utilisateur

Mettre une valeur distincte dans chaque cellule du premier tableau

Et 0 dans celles du second

Choisir un nombre aléatoire compris en 0 et le nombre de cellule du tableau

Sélectionner la cellule correspondante

Choisir un nombre aléatoire compris entre 1 et 4

Ouvrir le chemin correspondant

Transférer la valeur de cellule choisi dans la cellule nouvellement connectée

Modifier le second tableau en créant la connexion

Recommencer la sélection aléatoire de cellule jusqu’à ce que toutes les valeurs du premier tableau du soit égale.

Dans votre deuxième tableau.

Choisissez une valeur comprise entre zéro et le périmètre du labyrinthe.

Ouvrer donc vers l’extérieur la cellule correspondante et ajouter lui 32

Cette cellule est votre sortie

Choisissez aléatoirement une cellule de votre tableau différente de votre sortie

Ajouter lui 16

Cette Cellule est votre point de départ

Et lors votre labyrinthe est le deuxième tableau

## 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 même 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 branches. 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\_V2 | 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 |
| Journal de travail.pdf | Journal de travail |
| Journal de bord.pdf | Journal de bord |
| CDC-PBA-Pablo-Zubieta\_Labyrinthe\_v1\_2 | Cahier des charges |

### Dossier wireframe

|  |  |
| --- | --- |
| Nom | Description |
| Tpi\_Zonning.pdf | Le Zoning du projet |
| Tpi\_WF.pdf | Les Wireframes 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 fichiers de Classe Controllers de ce site |
| UserController.php |
| Users\_does\_labyrinthe  Controller.php |
| LabyrintheController.php |
| App/Model | Tous les fichiers 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

* Mise en place de l’environnement.

1. Planification

|  |  |
| --- | --- |
| Nom | Etat |
| Temps de travail | Réussi |

1. Base de données

|  |  |
| --- | --- |
| Nom | Etat |
| MCD | Réussi |
| MLD | Réussi |
| Création de la base donnée | Réussi |

1. Repository Git

|  |  |
| --- | --- |
| Nom | Etat |
| Repository distant | Réussi |
| Repository local | Réussi |
| Git flow | Réussi |

1. Hébergement

|  |  |
| --- | --- |
| Nom | Etat |
| Accès | Réussi |
| Pas de régression | Réussi |

1. Laravel

|  |  |
| --- | --- |
| Nom | Etat |
| Laravel | Réussi |

1. Wireframe

|  |  |
| --- | --- |
| Nom | Etat |
| Création de Wireframe | Réussi |

* Interface utilisateur

1. Views

|  |  |
| --- | --- |
| Nom | Etat |
| Views | Réussi |

1. Register

|  |  |
| --- | --- |
| Nom | Etat |
| Création d'un utilisateur | Réussi |
| Diffèrent mot de passe | Réussi |
| Utilisateur existant | Réussi |

1. Login

|  |  |
| --- | --- |
| Nom | Etat |
| Connexion réussite | Réussi |
| Mauvais mot de passe | Réussi |
| Champ vide | Réussi |
| Utilisateur inconnu | Réussi |

1. Routes

|  |  |
| --- | --- |
| Nom | Etat |
| Routes Get | Réussi |
| Routes Post | Réussi |
| Phasing des routes | Réussi |

1. Phasing utilisateur

|  |  |
| --- | --- |
| Nom | Etat |
| Connecter | Réussi |
| Déconnecter | Réussi |

1. Historique de labyrinthe

|  |  |
| --- | --- |
| Nom | Etat |
| Information personnelle | Réussi |
| Liste des labyrinthes résolus | Réussi |
| Liste des labyrinthes crées | Réussi |
| Lien labyrinthe | Réussi |

1. Interface de création

|  |  |
| --- | --- |
| Nom | Etat |
| Tuile de labyrinthe | Réussi |
| Taille du labyrinthe | Réussi |
| Liste des labyrinthes crées | Réussi |
| Donnée inconforme | Réussi |
| Tuiles | Réussi |
| Début et fin | Réussi |
| Reset | Réussi |
| Drag and drop | Réussi |
| Mauvaise fin | Réussi |
| Valider | Réussi |

* Labyrinthe

1. Norme d’encodage

|  |  |
| --- | --- |
| Nom | Etat |
| Unicité | Réussi |
| Encodage | Réussi |
| Topologie | Réussi |

1. Encoder et décoder un labyrinthe

|  |  |
| --- | --- |
| Nom | Etat |
| Encoder | Réussi |
| Décoder | Réussi |

1. Algorithme de résolution

|  |  |
| --- | --- |
| Nom | Etat |
| Ligne réussi | Réussi |
| Ligne échec | Réussi |
| Cul de sac réussi | Réussi |
| Cul de sac échec | Réussi |
| Labyrinthe complexe réussi | Réussi |
| Labyrinthe complexe échec | Réussi |
| Loop réussi | Réussi |
| Loop échec | Réussi |
| Loop extérieur réussi | Réussi |
| Loop extérieur échec | Réussi |
| Mauvais chemin réussi | Réussi |
| Mauvais chemin échec | Réussi |

1. Algorithme de création

|  |  |
| --- | --- |
| Nom | Etat |
| Entrée/sortie dissemblables | Réussi |
| Labyrinthes dissemblables | Réussi |

## 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 |
| Unicité du code du labyrinthe. | Le code du labyrinthe étant une string ne représentant que les valeurs des cellules. | Deux labyrinthes différents peuvent avoir un code égal : exemple un labyrinthe 4X5 et un labyrinthe 5x4. Leur longueur et leur hauteur n’est encoder dans la string | Il faut également en coder la longueur et hauteur dans le code labyrinthe. L’action a déjà été entreprise la base de données ayant été modifier en conséquence. Cependant cette modification n’a pas encore été faites dans l’algorithme d’encodage et de décodage |
| Les chemins vides | Figure 11 exemple d'erreur | Crée un message d’erreur et parvient pas définir que ce labyrinthe est insoluble | Repenser soit l’interface de création en empêchant l’utilisateur de soumettre des chemins vides(solution probablement la plus pertinentes) ou consolider l’algorithme de solution sur les chemin sans issue |
| Route Post sur Résolution | Enter des valeurs non conformes sur le formulaire résolution depuis la page résolution | Crée une erreur de qui rend la page inaccessible | Cette erreur est probablement dû à un problème d’accès. Si le formulaire de résolution est incorrectement rempli il renvoie l’utilisateur de retour sur la page précédente cependant dans le cas de page résolution il le renvoie sur lui-même mais sans variable post d’où l’erreur en question |

# Mise en service

## Rapport de mise en service

### Publication chez l’hébergeur

* Pour la publication chez l’hébergeur il faut un client Ftp pour transférer tous les ficher du dossier code
* Il vous faudra créer la base de donner sur l’hébergeur grâce au script de création de base de donnée
* Définir le dossier public ou se trouve Htacces comme racine du service Apache

### Guide d’installation

* Pour l’installation il vous faut évidemment 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

## Liste des documents fournis

* Documentation
* Journal de travail
* Journal de Bord

# Conclusions

Et voilà qui conclus mon Tpi. Ce projet c’est déroulé très différemment de que ce j’avais anticipé. Il y a 6 mois lorsque j’ai déposé ma fiche signalétique ce que j’attendais de ce travail, était de vraiment pouvoir travailler sur de l’algorithmie, créer différents algorithmes de construction de labyrinthe, comparer leur complexité, comparer leurs propriétés. Je ne pensais pas vraiment aux questions d’interface utilisateur, persistance des données ou du MVC. Il y a de ça un mois lorsque j’ai fini mon pré-tpi (un site de covoiturage), un projet qui m’as vraiment fait plonger dans la compréhension des bases de données, procédure stocké et problème d’optimisation de requêtes SQL, je voyais donc mon Tpi comme la continuation de ce travail, (un projet Back-end). D’où la surprise avérée, lorsque j’ai découvert mon Tpi. Il était en effet très accès sur de l’interface utilisateur (comme l’interface de génération manuelle, le respect de critère de Bastien et Scapin, responsivité du site), il correspondait donc plus au cahier des charges d’un projet Front-end. Je n’avais une grande expérience en JavaScript (je n’avais que très peu vu les notion d’évènement ou de modification du dom en Js), langage indispensable pour ce type de projet. Malgré ça je suis assez content du résultat, il est évidement bien en dessous des attentes d’une personne aussi consciencieuse que moi. Mais il m’a permis d’étoffer mes connaissances en Js. Et vraiment prendre le temps de réfléchir au interface utilisateur.

Les principales difficultés que j’ai rencontrées étant l’interface de création manuelle et l’algorithme testant la solution.

Pour ce qui est de l’interface de création, la difficulté est venue de la compréhension du dom et de la gestion d’évènement. J’ai choisi de faire cette interface via la fonctionnalité drag and drop d’HTML, bien que cela me semblait sur le moment la meilleure idée, je pense aujourd’hui que ce n’était pas la bonne approche en effet cette fonctionnalité n’est pas convenante sur une interface mobile. Je pense qu’il m’aurait créé une interface plus dynamique basé sur la fonctionnalité *onclick* aurait été plus pertinente.

Pour ce qui est l’algorithme testeur de solution, sa difficulté a été dans le fait de passer de la solution topologique, a la solution informatique et numérique. En effet j’ai pris un temps non-négligeable pour corriger le test de chemin topologiquement équivalent mais informatiquement distinct. Et des erreurs à ce niveau-là persiste.

Pour conclure même si ce projet m’as fait quelques cheveux blancs, ça m’as vraiment permis de voir les interactions entre un Front-end (en Js) et un aspect Back-end (en PHP), ainsi qu’un léger contact avec la création d’algorithme. Merci à M.Benzonana d’avoir eu confiance en moi pour me suivre dans ce projet, malgré des horaires différents. J’espère que ce projet suffira à sortir du Cpnv.

# Annexes

## Résumé du rapport

Mon projet était la création d’un site de labyrinthe. Les fonctionnalités principales étant l’interface de création manuel et un algorithme de génération automatique. Pour ce se faire je disposais 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, et pour assurer persistance des données j’ai utilisé une base de données MYSQL.

Pour la mise en œuvre, elle s’est séparée en quatre parties(sprints). La première semaine, j’ai avant tout mis en place mon environnement de développement, crée repository, MLD, MCD, ainsi que le Login et le register. La deuxième semaine, j’ai mise en place mon interface de création de labyrinthe, pour être fonctionnel cette interface à demander beaucoup de JavaScript. La troisième semaine, j’ai créé un algorithme vérifiant si les labyrinthes créés par l’utilisateur avaient une solution. La dernière semaine Les gros soucis de la mise en œuvre ayant été l’interface de création et l’algorithme de solution.

Donc la réalisation est relativement conforme au cahier des charges, il contient des points intéressant tel que l’encodage en base64 ou l’algorithme de création automatique. Je pense que l’on peut aussi souligner les défauts : une interface de création avec une faible responsivité, et peu ergonomique, ainsi trop permissive pour l’utilisateur (ce qui a pour conséquence de mal se combiner avec l’algorithme de résolution) et mon inconsistance dans ma documentation.

## Glossaire

|  |  |
| --- | --- |
| Mot | Définition |
| AGILE | Méthode des gestion de projet propice à une plateforme comme Icescrum |
| Apache | Service d’hébergement sur un serveur web |
| Base64 | Norme d’encodage transformant du binaire en caractère imprimable |
| Composer | Logiciel d’installation de librairie notamment Laravel ou Symphony |
| Drag and drop | Fonctionnalité HTML permettant de saisir et déposé les éléments d’une page web |
| Framework | Ensemble de librairie encadrant une technologie préexistante |
| GitFlow | Framework de Git générant un archivage sur plusieurs branche |
| Htacces | Fichier de configuration web |
| Laravel | Framework PHP |
| MCD | Modèle conceptuel de données |
| MLD | Modèle logique de données |
| MVC | Méthode structurelle d’organisation d’une application (Model, Views, Controller) |
| Nosql | Not Only SQL sont des bases de donnée qui ne sont pas seulement relationnelles |
| Repository | Dossier d’archivage versionné local ou distant |
| Topologie | Champ d’étude Mathématique de la géométrie sans mesure. C’est elle qui définir si un labyrinthe a une solution ou non s’il est parfait ou non. |
| Vendor | Dossier contenant l’intégralité de la libraire de Laravel |

## 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 |
| <https://www.youtube.com/watch?v=sXwPfnsKGiE> | 10.05.2023 |
| <https://www.w3schools.com/JSREF/prop_html_id.asp> | 10.05.2023 |
| <https://www.youtube.com/watch?v=N77_0lq9JZI> | 11.05.2023 |
| <https://www.w3schools.com/JSREF/prop_node_childnodes.asp> | 11.05.2023 |
| <https://developer.mozilla.org/en-US/docs/Web/API/Node/removeChild> | 11.05.2023 |
| <https://www.codexworld.com/how-to/convert-php-array-to-javascript-array/> | 15.05.2023 |
| <https://en.wikipedia.org/wiki/Maze-solving_algorithm> | 16.05.2023 |
| <http://www.mit.edu/~6.005/sp11/psets/ps3/assignment.html> | 16.05.2023 |
| <https://www.php.net/manual/en/language.operators.bitwise.php> | 16.05.2023 |
| <https://www.youtube.com/watch?v=HCQxjQoeWpg> | 23.05.2023 |
| [https://fr.wikipedia.org/wiki/Mod%C3%A9lisation \_math%C3%A9matique\_d%27un\_labyrinthe](https://fr.wikipedia.org/wiki/Mod%C3%A9lisation_math%C3%A9matique_d%27un_labyrinthe) | 23.05.2023 |
| <https://www.ilay.org/yann/articles/maze/> | 25.05.2023 |
| <https://getbootstrap.com/docs/4.0/components/dropdowns/> | 25.05.2023 |

## 

## Manuel d'Installation

Voir guide d’installation

## Manuel d'Utilisation

Voir publication chez l’hébergeur

## Table des illustrations

[Figure 1 MCD du projet 13](#_Toc136355807)

[Figure 2 encodage topologique 21](file:///C:\Tpi\Documentation\Documentation_Tpi.docx#_Toc136355808)

[Figure 3 Planification détaillée 22](#_Toc136355809)

[Figure 4 MCD\_v1 23](#_Toc136355810)

[Figure 5 MLD\_v1 23](#_Toc136355811)

[Figure 6 Arborescence du site 24](#_Toc136355812)

[Figure 7 Palette de couleur 24](#_Toc136355813)

[Figure 8 Homepage du site 25](#_Toc136355814)

[Figure 9 interface de création du site 25](#_Toc136355815)

[Figure 10 MLD 26](#_Toc136355816)

[Figure 11 exemple d'erreur 32](#_Toc136355817)

## Wireframes

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Une image contenant texte, capture d’écran, nombre, Rectangle

Description générée automatiquement

Une image contenant texte, capture d’écran, affichage, nombre

Description générée automatiquementUne image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement Une image contenant texte, capture d’écran, Rectangle, Parallèle

Description générée automatiquement Une image contenant texte, capture d’écran, diagramme, Rectangle

Description générée automatiquement

## Archives du projet

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

## Journal de bord

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement