|  |
| --- |
| CFPT |
| Application web de e-Commerce, partie administration |
| TPI 2021 |

|  |
| --- |
| Morrone Flavio  03/05/2021 |

Table des matières

[1. Introduction 2](#_Toc70951102)

[2. Cahier des charges 2](#_Toc70951103)

[I. Matériels et logiciels à disposition 2](#_Toc70951104)

[II. Descriptif complet du projet 2](#_Toc70951105)

[A. Description de l’application 2](#_Toc70951106)

[B. Planification prévisionnel 3](#_Toc70951107)

[C. Planification effective 4](#_Toc70951108)

[D. Product Backlog 4](#_Toc70951109)

[E. Modèle Logique de données 4](#_Toc70951110)

[III. Livrables 4](#_Toc70951111)

[3. Méthodologie 5](#_Toc70951112)

[I. S’informer 5](#_Toc70951113)

[II. Planifier 5](#_Toc70951114)

[III. Décider 5](#_Toc70951115)

[IV. Réaliser 6](#_Toc70951116)

[V. Contrôler 6](#_Toc70951117)

[VI. Evaluer 6](#_Toc70951118)

[4. Outils utilisés 7](#_Toc70951119)

[I. Laragon 7](#_Toc70951120)

[II. Visual Studio Code 7](#_Toc70951121)

[III. MySQL Workbench 7](#_Toc70951122)

[IV. Github 7](#_Toc70951123)

[5. Architecture 8](#_Toc70951124)

[6. Schéma de l’application 9](#_Toc70951125)

[7. Analyse organique 10](#_Toc70951126)

[I. Fonctionnalités intégrés 10](#_Toc70951127)

[II. Architecture du projet 10](#_Toc70951128)

[III. Diagrammes de classes 10](#_Toc70951129)

[8. Analyse fonctionnelle 10](#_Toc70951130)

[9. Procédure de tests 10](#_Toc70951131)

[10. Conclusion 10](#_Toc70951132)

[I. Difficultés rencontrées 10](#_Toc70951133)

[II. Amélioration possible 10](#_Toc70951134)

[III. Bilan personnel 10](#_Toc70951135)

[11. Annexes 10](#_Toc70951136)

[I. Glossaire 10](#_Toc70951137)

[II. Sources 10](#_Toc70951138)

# Introduction

Ce TPI est un projet réalisé par 3 candidats. Le projet seul était une charge de travail trop importante pour qu’il soit réalisé seul en 3 semaine. Nous avons donc mis en commun une architecture afin de remettre le projet en un afin qu’il soit totalement fonctionnel.

L’application est un site de e-commerce contenant 3 parties :

* Une partie administration et de gestion des utilisateurs
* Une partie de gestion du panier
* Une partie de gestion des stocks

Ma partie était celle de l’administration et de gestion des utilisateurs. Cette partie consistait à gérer les catégories, les utilisateurs ainsi que la connexion et enregistrement tout en étant sécurisant au maximum le site.

# Cahier des charges

## Matériels et logiciels à disposition

* Un PC avec Windows 10
* Deux écrans
* Un IDE au choix (Visual Studio Code, Netbeans, Notepad++)
* Un serveur web au choix (Laragon, EasyPhP, Wamp, Xamp)
* Un navigateur web (Firefox, Chrome, Edge)
* Un outil de versionning (Github)
* Un logiciel de gestion de base de données (PHPMyAdmin, MySQL Workbench)
* Des logiciels de bureautiques (Word, Excel, Figma)

## Descriptif complet du projet

### Description de l’application

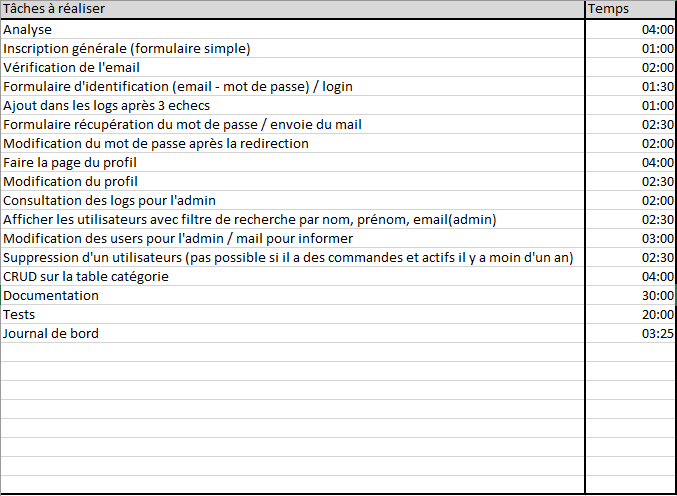
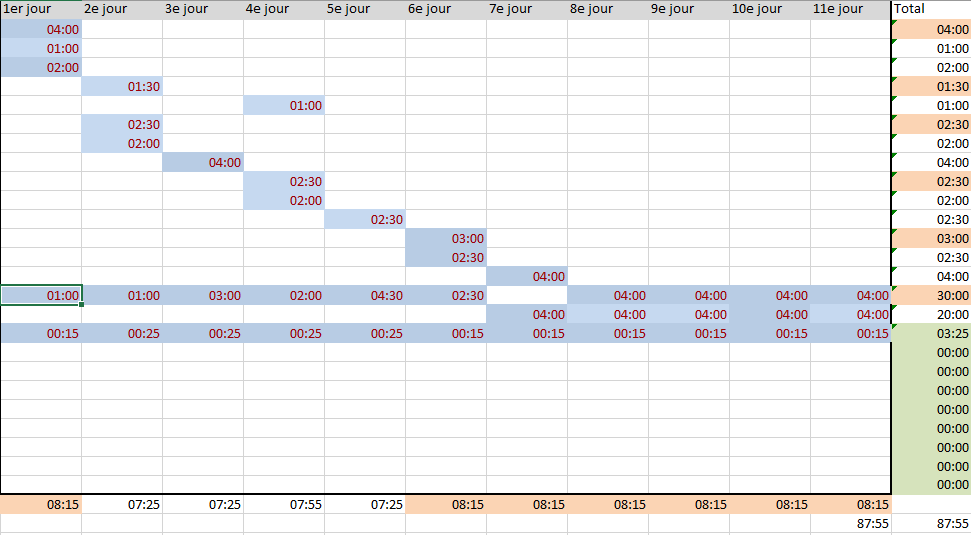
Le principe de ma partie est de gérer les rôles, la connexion et l’enregistrement des utilisateurs.

Lors de l’enregistrement, l’email doit être vérifié afin de ne pas avoir des faux mails.

Chaque utilisateur possède un rôle qui doit lui permettre de faire des choses ou non sur le site.

De plus, je dois gérer les catégories du site avec un CRUD sur leurs tables.

### Planification prévisionnel



### Planification effective

### Product Backlog

|  |  |
| --- | --- |
| **Titre** | Inscription |
| **Description** | Faire en sorte qu’un utilisateur puisse créer son compte avec un email, nom, prénom, adresse et un mot de passe |
| **Test** | 1, 2, 3, 4 |

|  |  |
| --- | --- |
| **Titre** | Vérification de l’email |
| **Description** | Une fois l’inscription réalisé, l’utilisateur peut se connecter mais son rôle est « NotVerified » afin de devenir « Customer » il a reçu un mail au moment de son inscription afin de confirmer son adresse email. Il lui suffit de cliquer sur le lien et de rentrer son email. |
| **Test** | 9, 10, 11 |

|  |  |
| --- | --- |
| **Titre** | Identification |
| **Description** | Après avoir créé son compte, l’utilisateur souhaite se connecter afin de pouvoir accéder au contenu du site. Il lui suffit d’entrer son email et son mot de passe. Si l’utilisateur loupe trois fois la connexion, cela est noté dans les logs. |
| **Test** | 5, 6 |

|  |  |
| --- | --- |
| **Titre** | Récupération du mot de passe |
| **Description** | Si l’utilisateur perd son mot de passe, il a la possibilité de le récupérer.  Il rentre son email dans un formulaire et reçoit un mail de réinitialisation du mot de passe si le compte existe. Il clique sur le lien et doit ensuite mettre son email pour vérifier son identité ainsi que son nouveau mot de passe. |
| **Test** | 12, 13, 14, 15 |

|  |  |
| --- | --- |
| **Titre** | Le profil |
| **Description** | Chaque utilisateur (vérifier ou non) peut apercevoir. Un utilisateur non vérifié ne peut pas le modifier. |
| **Test** |  |

|  |  |
| --- | --- |
| **Titre** | Consultation des logs |
| **Description** | L’administrateur est dans la capacité de consulter les logs de l’application |
| **Test** |  |

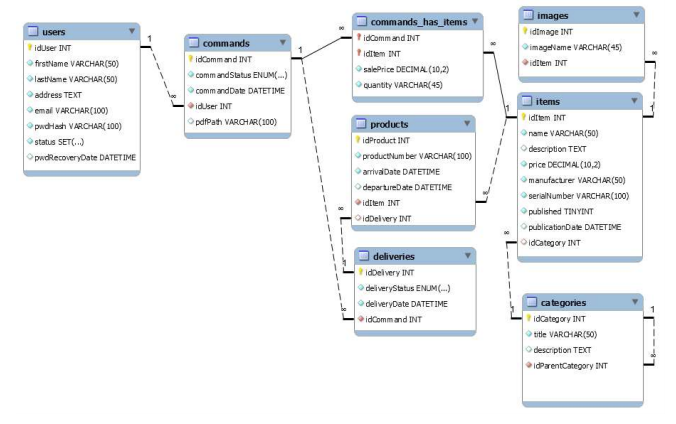
|  |  |
| --- | --- |
| **Titre** | Liste des utilisateurs |
| **Description** | L’administrateur peut voir la liste de tous les utilisateurs. |
| **Test** |  |

|  |  |
| --- | --- |
| **Titre** | Modifier les utilisateurs |
| **Description** | L’administrateur peut modifier n’importe quels champs de tous les utilisateurs sauf lui |
| **Test** |  |

|  |  |
| --- | --- |
| **Titre** | Supprimer les utilisateurs |
| **Description** | L’administrateur peut supprimer les utilisateurs dont la date de validation est expiré |
| **Test** |  |

|  |  |
| --- | --- |
| **Titre** | CRUD Catégories |
| **Description** | L’administrateur à accès à un CRUD sur la table catégories. Il peut créer des catégories et les rattacher à n’importe quelle autre catégorie. Il peut les déplacer ou il veut dans l’arborescence sauf dans ses catégories enfants. Il peut supprimer celles qui n’ont pas d’enfants. |
| **Test** |  |

### Modèle Logique de données



## Points techniques évalués

**A14** : Les fonctionnalités de l’administrateur du site concernant la gestion des utilisateurs sont réalisées.

**A15** : Les fonctionnalités des autres utilisateurs, en lien avec la table users, sont réalisées.

**A16** : Les mots de passes sont gérés de manière sécurisée.

**A17** : La gestion des catégories du site est réalisée.

**A18** : Les tests sont réalisés et documentés.

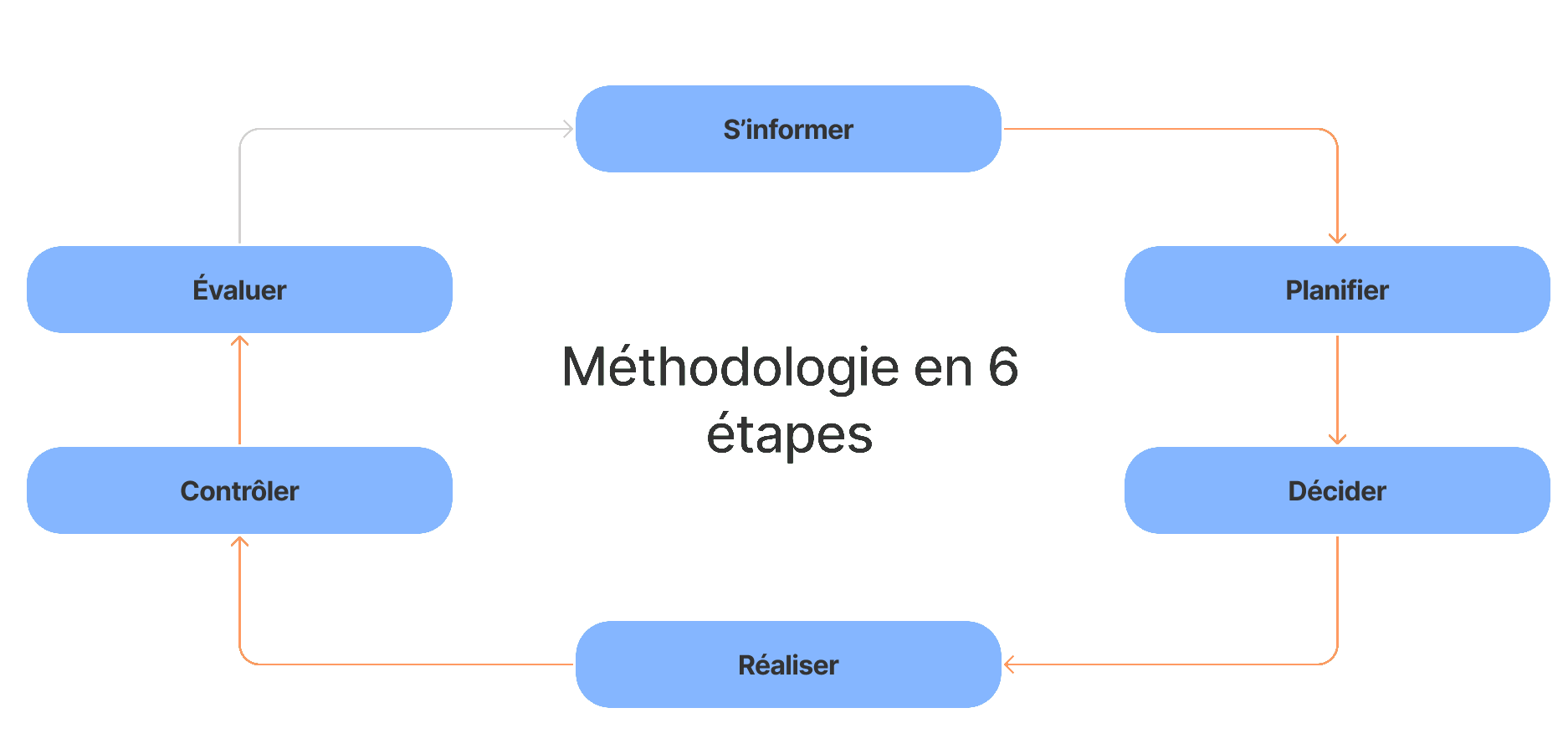
**A19** : L’architecture mise en place avant le TPI est respectée, et permet l’intégration avec les autres modules du projet.

**A20** : L’application est protégée contre les injections SQL et XSS.

## Livrables

* Planning prévisionnel fourni le premier jour du TPI
* Rapport de projet
* Manuel utilisateur
* Résumé du rapport du TPI
* Journal de travail

# Méthodologie

La méthodologie que j’ai utilisée se nomme **la méthodologie en 6 étapes**. Je l’avais expérimenté auparavant lors de différents modules. Elle consiste à séparer la réalisation du programme demandé en 6 étapes différentes. Ces étapes se nomment s’informer, Planifier, Décider, Réaliser, Contrôler et Évaluer.

## S’informer

La première étape de cette méthodologie était de lire en profondeur mon énoncé afin de bien comprendre le sujet. De plus, j’ai dû faire des recherches afin de savoir comment je pouvais faire certaines choses tel que la vérification de l’email ou encore le changement de mot de passe.

Puis j’ai posé mes questions à mon maître de TPI pour être certain d’avoir bien compris certaines choses.

## Planifier

La planification a pour principe de prévoir l’organisation de son travail. Pour cela, je me suis légèrement inspiré de la méthodologie **SCRUM** afin de faire un backlog complet contenant toutes mes *users-stories*. Je me suis inspiré de cette méthodologie uniquement pour le backlog étant donné que je travaille seul, le reste ne m’était pas très utile.

## Décider

Le but de cette étape est de choisir parmi les solutions que nous avons pour réaliser notre projet pour qu’il soit fait de manière optimale.

## Réaliser

La réalisation représente la partie la plus importante de cette méthodologie. Elle consiste à réaliser les taches mises en place lors de la planification et de la manière qui a été choisis lors de l’étape Décider.

## Contrôler

Chaque partie de la réalisation doivent être contrôlées et testées avant d’être remises aux clients ou à des tiers. Contrôler signifie relire, recalculer, vérifier le cahier des charges ainsi que tester minutieusement chaque partie du programme afin d’éviter d’éventuels bugs.

## Evaluer

Le but de la dernière étape est de passer en revue tout le déroulement des 6 étapes en prenant du recul et de réaliser un bilan afin d’éviter de reproduire nos erreurs dans le futur.

# C:\Users\Administrateur\Downloads\laragon.pngOutils utilisés

## Laragon

Laragon est un environnement de développement web uniquement disponible sous Windows. Il intègre de différentes technologies tel qu’Apache, MySQL et PHP. Il est très pratique pour installer des packages et des librairies. De plus, il facilite l’envoie de mails afin de ne pas être mis en spam.

## C:\Users\Administrateur\Downloads\visual studio code.pngVisual Studio Code

Visual studio code est un IDE puissant disponible sous Windows, Linux et Ubuntu. Il possède de base un support intégrant JavaScript, TypeScript et Node.js. Mais il dispose d’un très riches nombres d’extensions permettant de prendre en compte d’autres langages tel que C++, C#, Java, Python PHP et bien d’autres.

Les extensions que j’ai choisies ne sont autres que « PHP Intelephense » et « PHP Debug ».



## MySQL Workbench

MySQL Workbench est un outil visuel pour les architectures de bases de données MySQL. Il fournit des outils de modélisations de données, de développement SQL et d’administration pour la configuration d’un serveur. Il est disponible sous Windows, Linux et Mac.



## Github

Github est un service d’hébergement et outil de versionnage de logiciel. Il utilise le programme git afin de gérer les versions.

# Architecture

L’architecture de l’application se nomme Modèle Vue Contrôleur (MVC). Elle consiste à séparer le programme en 4 parties.

* Le routeur
* Les contrôleurs
* Les modèles
* Les vues

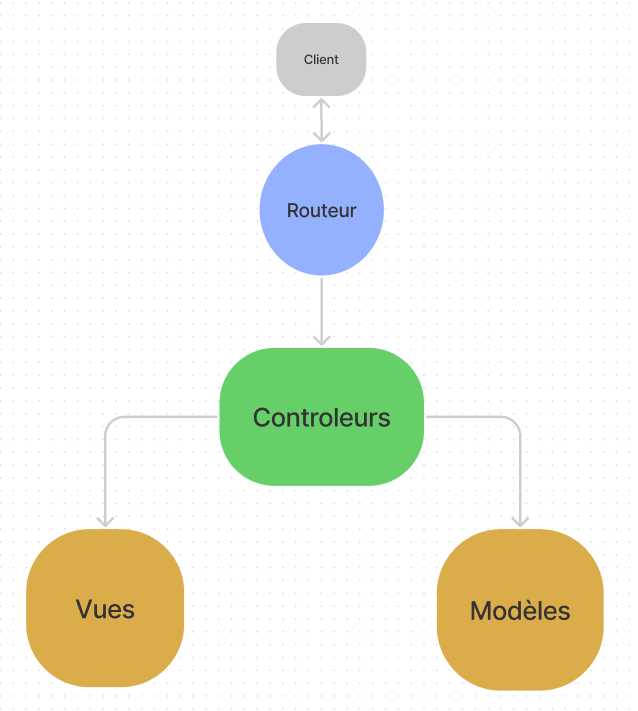
Le routeur est le point central du projet. Il permet la redirection sur les pages demandés selon les requêtes demandées.

Les contrôleurs sont les liaisons entre la base de données et les vues. Ils permettent de récupérer et de filtrer toutes les données de la base de données avant de les envoyer aux vues.

Les vues permettent l’affichage des données envoyées par les contrôleurs. Elles n’ont aucune liaison directe avec la base de données.

# Schéma de l’application

L’application respectant l’architecture MVC se présente donc comme ceci :



# Analyse organique

## Fonctionnalités intégrés

## Architecture du projet

## Diagrammes de classes

# Diagramme de navigation

# Analyse fonctionnelle

# Plan de tests

J’ai décidé de mettre en place un plan de test afin de vérifier plus facilement que tout fonctionne.

|  |  |  |
| --- | --- | --- |
| **N° Test** | **Descriptif** | **Résultat attendu** |
| 1 | **Inscription :**Faire une inscription avec des données valables (nom, prénom, adresse, email, mot de passe) | Redirection sur la page login avec un message de confirmation + envoi d’un mail |
| 2 | **Inscription :** Faire une inscription sans remplir toutes le données | Redirection sur la même page avec un message d’erreur |
| 3 | **Inscription :** Inscription avec un email déjà pris | Redirection sur la page login avec un message d’erreur |
| 4 | **Inscription :** Inscription avec le mot de passe et confirmation de mot de passe différente | Redirection sur la même page avec un message d’erreur |
| 5 | **Connexion :**Connexion avec un compte non vérifié | Connexion avec comme rôle « NotVerified » |
| 6 | **Connexion :** Connexion avec des mauvais identifiants | Message d’erreur |
| 7 | **Déconnexion :** Cliquer sur le bouton déconnexion | Vide la session et déconnecte l’utilisateur |
| 8 | **Changement de rôle :** Cliquer sur un rôle non sélectionné et disponible | Change le rôle de l’utilisateur |
| 9 | **Vérification d’email :** Entre son email dans le formulaire sur la page de vérification | Redirige sur la page login avec un message de succès et valide l’utilisateur |
| 10 | **Vérification d’email :** Entre le bon email lors de la vérification | Valide le compte + message de succès |
| 11 | **Vérification d’email :** Entre un mauvais email lors de la vérification | Message d’erreur |
| 12 | **Vérification d’email :** Modifie le token lors de la vérification | Message d’erreur |
| 13 | **Changement de mot de passe :** Met son email dans le premier formulaire | Reçois un mail avec un lien |
| 14 | **Changement de mot de passe :** clique sur le lien | Redirigé vers un formulaire |
| 15 | **Changement de mot de passe :** rentre le bon email et le nouveau mot de passe dans les deux champs | Redirigé vers la page login avec message de succès |
| 16 | **Changement de mot de passe :** met le mauvais email | Message d’erreur |
| 17 |  |  |

# Rapport de test

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **N° Test** | **Résultat obtenu** | **OK / KO** |
| 04.05.2021 | 1 | Redirection sur la page login avec un message de confirmation + envoi d’un mail | OK |
| 04.05.2021 | 2 | Rafraichissement + message d’erreur | OK |
| 04.05.2021 | 3 | Redirection sur la page login + message d’erreur | OK |
| 04.05.2021 | 4 | Rafraichissement + message d’erreur | OK |
| 04.05.2021 | 5 |  | OK |
| 04.05.2021 | 6 | Message d’erreur | OK |
| 04.05.2021 | 7 | Redirection à la page d’accueil déconnecté | OK |
|  | 8 |  |  |
| 04.05.2021 | 9 | Reçois un mail avec un lien+ redirection sur le login | OK |
| 04.05.2021 | 10 | Message d’erreur | KO |
| 04.05.2021 | 10 | Message de validation + redirection sur la page login | OK |
| 04.05.2021 | 11 | Message d’erreur | OK |
| 04.05.2021 | 12 | Message d’erreur | OK |
| 04.05.2021 | 13 | Reçois un mail | OK |
| 04.05.2021 | 14 | Redirection sur une page avec un formulaire | OK |
| 04.05.2021 | 15 | Redirection + message succes | OK |
| 04.05.2021 | 16 | Message d’erreur | OK |
|  | 17 |  |  |
|  | 18 |  |  |
|  | 19 |  |  |
|  | 20 |  |  |
|  |  |  |  |

# Conclusion

## Difficultés rencontrées

## Amélioration possible

## Bilan personnel

# Annexes

## Glossaire

## Sources

[Dossier annexes]

Code source

Manuel utilisateur

Résumé du projet

Planning