|  |
| --- |
| 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 un architecture afin de remettre le projet

# 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

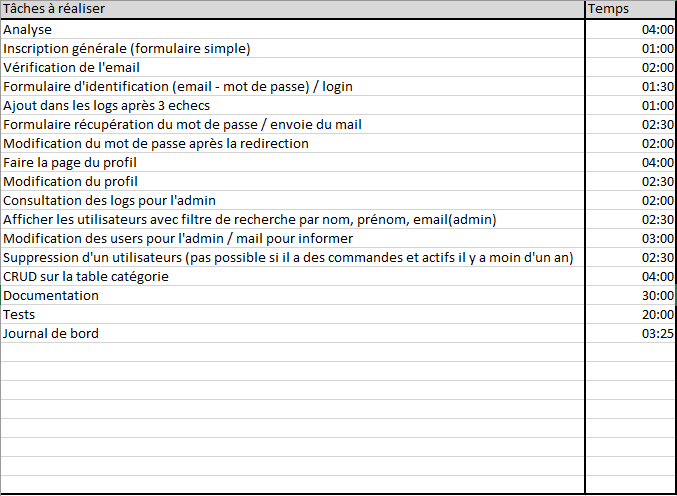
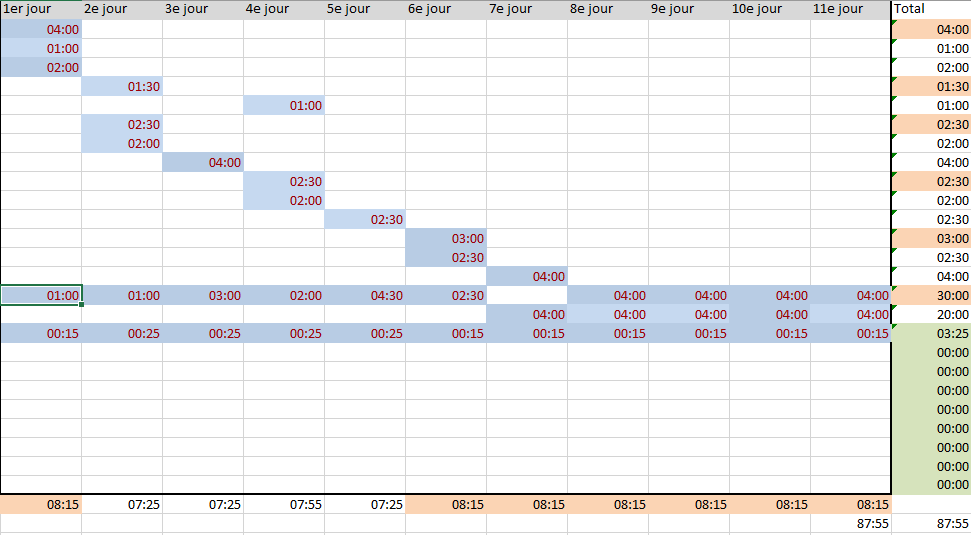
Le principe de ma partie est de gérer les rôles, la connexion et l’enregistrement des utilisateurs.

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.

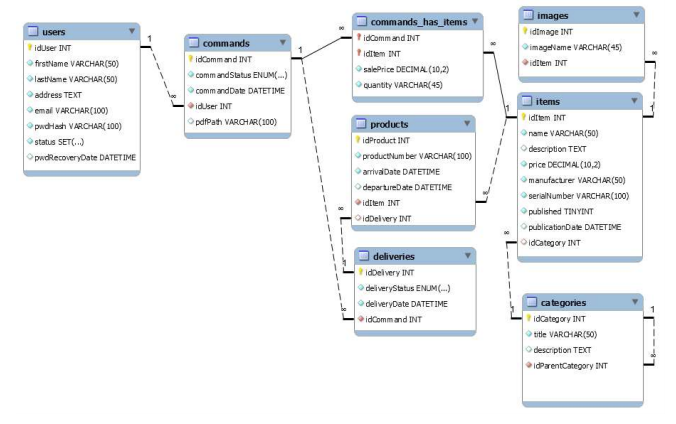
### Planification prévisionnel



### Planification effective

### Product Backlog

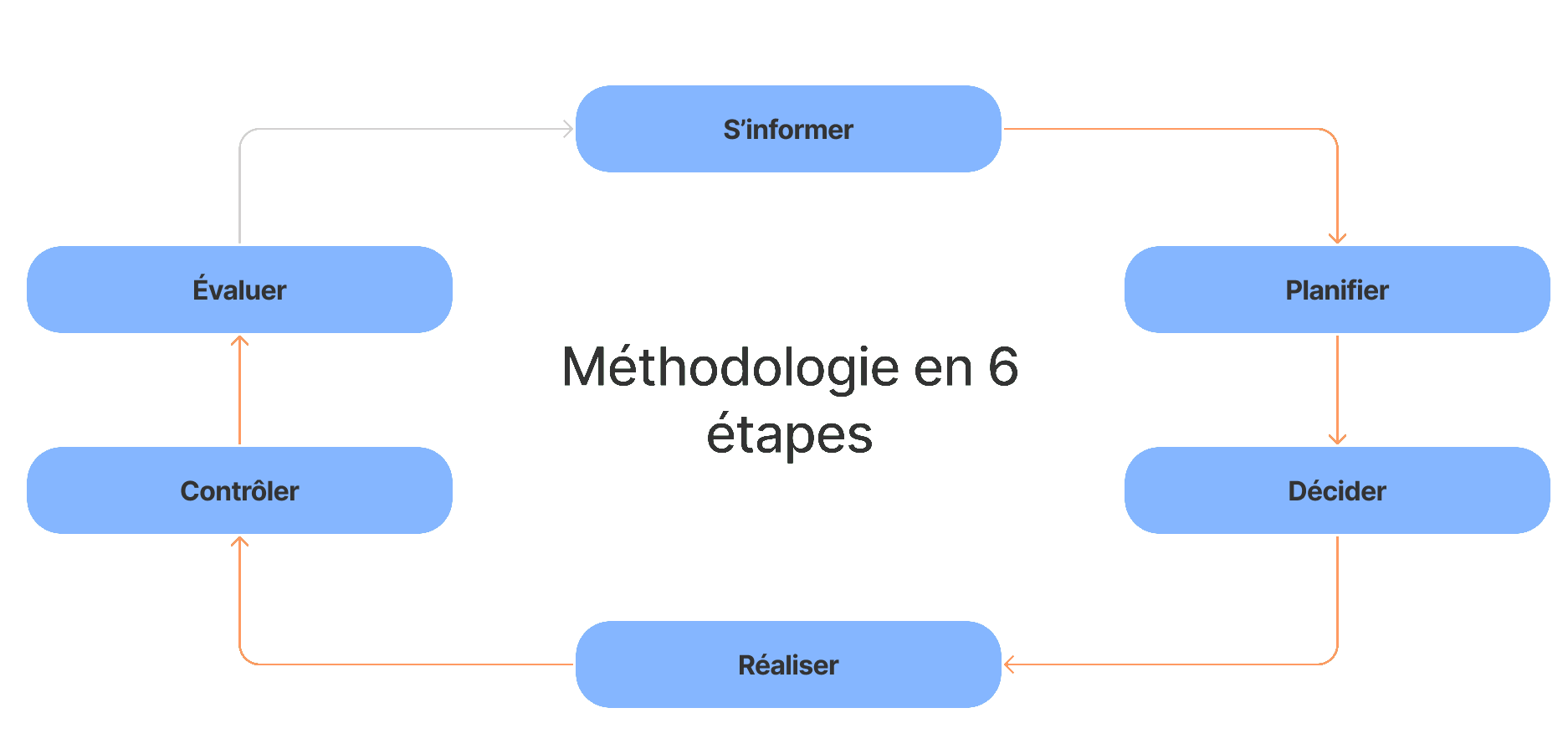
### Modèle Logique de données



## 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

S’informer consiste à comprendre et de se faire l’idée la plus proche possible de la demande du client. C’est une étape incontournable afin de ne pas partir dans le mauvais sens et de faire l’inverse de ce qui est voulu.

On peut se poser les questions suivantes :

* Quel est le but de ce projet ?
* Quel est le résultat attendu ?
* De quelles informations en plus ai-je besoin ?

## Planifier

La planification a pour principe de prévoir l’organisation de son travail. Pour cela il faut de l’imagination afin de pouvoir imaginer l’ordre des taches, de choisir les technologies et la façon de faire son programme, de définir chaque tache et d’estimer leurs temps.

## 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.

# Outils utilisés

## C:\Users\Administrateur\Downloads\laragon.pngLaragon

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

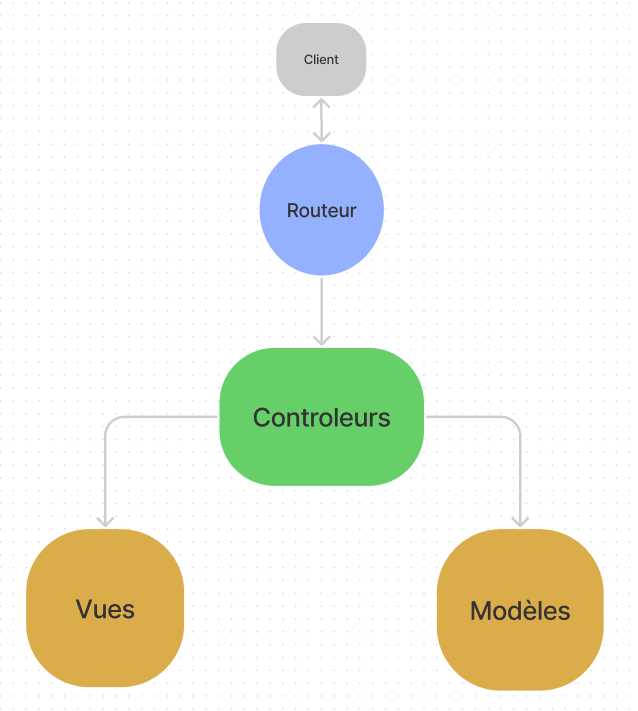
Le routeur est le point central du projet. Il permet la redirection sur les pages demandés selon les requêtes demandées.

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

# Procédure de tests

# Conclusion

## Difficultés rencontrées

## Amélioration possible

## Bilan personnel

# Annexes

## Glossaire

## Sources

[Dossier annexes]

Code source

Manuel utilisateur

Résumé du projet

Planning