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

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

Table des matières

[1. Introduction 3](#_Toc72160391)

[2. Cahier des charges 3](#_Toc72160392)

[I. Matériels et logiciels à disposition 3](#_Toc72160393)

[II. Descriptif complet du projet 3](#_Toc72160394)

[A. Description de l’application 3](#_Toc72160395)

[B. Planification prévisionnel 4](#_Toc72160396)

[C. Planification effective 5](#_Toc72160397)

[D. Product Backlog 5](#_Toc72160398)

[E. Modèle Logique de données 7](#_Toc72160399)

[III. Points techniques évalués 8](#_Toc72160400)

[IV. Livrables 8](#_Toc72160401)

[3. Méthodologie 9](#_Toc72160402)

[I. S’informer 9](#_Toc72160403)

[II. Planifier 9](#_Toc72160404)

[III. Décider 10](#_Toc72160405)

[IV. Réaliser 10](#_Toc72160406)

[V. Contrôler 10](#_Toc72160407)

[VI. Evaluer 10](#_Toc72160408)

[4. Outils utilisés 11](#_Toc72160409)

[I. Laragon 11](#_Toc72160410)

[II. Visual Studio Code 11](#_Toc72160411)

[III. MySQL Workbench 11](#_Toc72160412)

[IV. Github 11](#_Toc72160413)

[V. Antidote 11](#_Toc72160414)

[5. Architecture 12](#_Toc72160415)

[6. Analyse organique 13](#_Toc72160416)

[I. Fonctionnalités intégrés 13](#_Toc72160417)

[II. Diagrammes de classes 16](#_Toc72160418)

[III. 16](#_Toc72160419)

[IV. Structure 16](#_Toc72160420)

[7. Plan de tests 18](#_Toc72160421)

[8. Rapport de test 20](#_Toc72160422)

[9. Conclusion 21](#_Toc72160423)

[I. Difficultés rencontrées 21](#_Toc72160424)

[II. Amélioration possible 21](#_Toc72160425)

[III. Bilan personnel 21](#_Toc72160426)

[10. Annexes 21](#_Toc72160427)

[I. Glossaire 21](#_Toc72160428)

[II. Sources 21](#_Toc72160429)

# 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

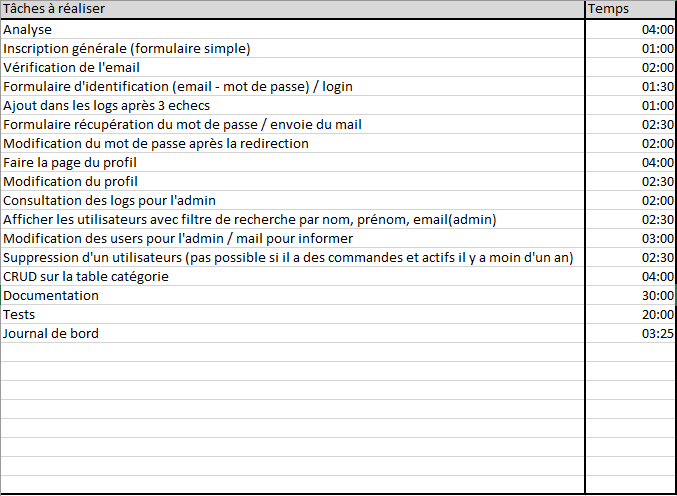
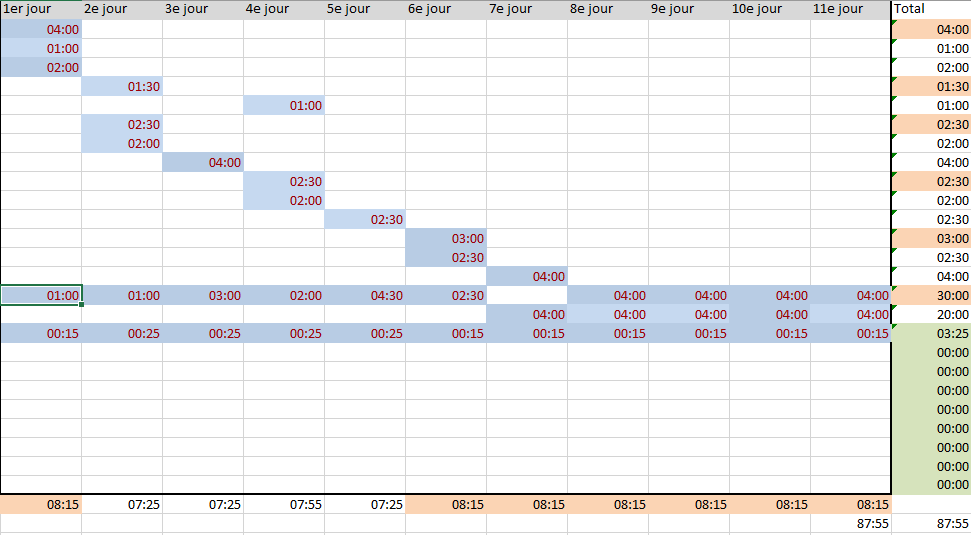
Le principe de ma partie est de gérer les rôles, la connexion et l’enregistrement des utilisateurs.

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. Les administrateurs ont accès à la liste des utilisateurs qu’ils peuvent modifier ainsi qu’à une liste de log.

### 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** |  |

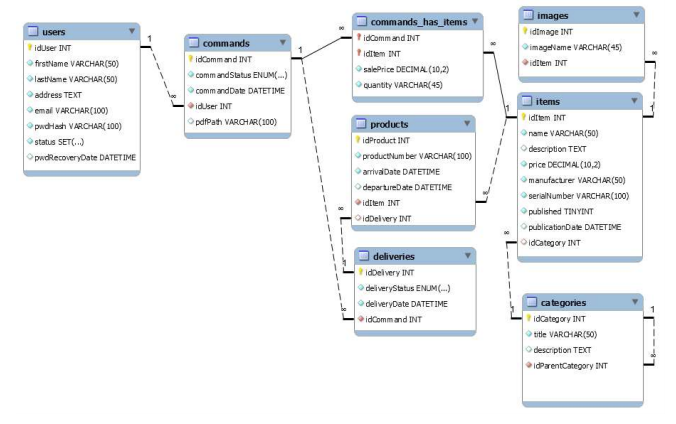
|  |  |
| --- | --- |
| **Titre** | Modifier les utilisateurs |
| **Description** | L’administrateur peut modifier n’importe quels champs de tous les utilisateurs sauf lui |
| **Test** |  |

|  |  |
| --- | --- |
| **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** |  |

|  |  |
| --- | --- |
| **Titre** | Ligne d’Ariane |
| **Description** | Lorsqu’on souhaite voir le contenu de une seule catégorie, un ligne contenant les parents de la catégorie doit s’afficher. |
| **Test** |  |

### Modèle Logique de données



Ce modèle m’a été fourni avec l’énoncé comme nous somme 3 sur le projet afin d’avoir le même modèle logique.

## 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.

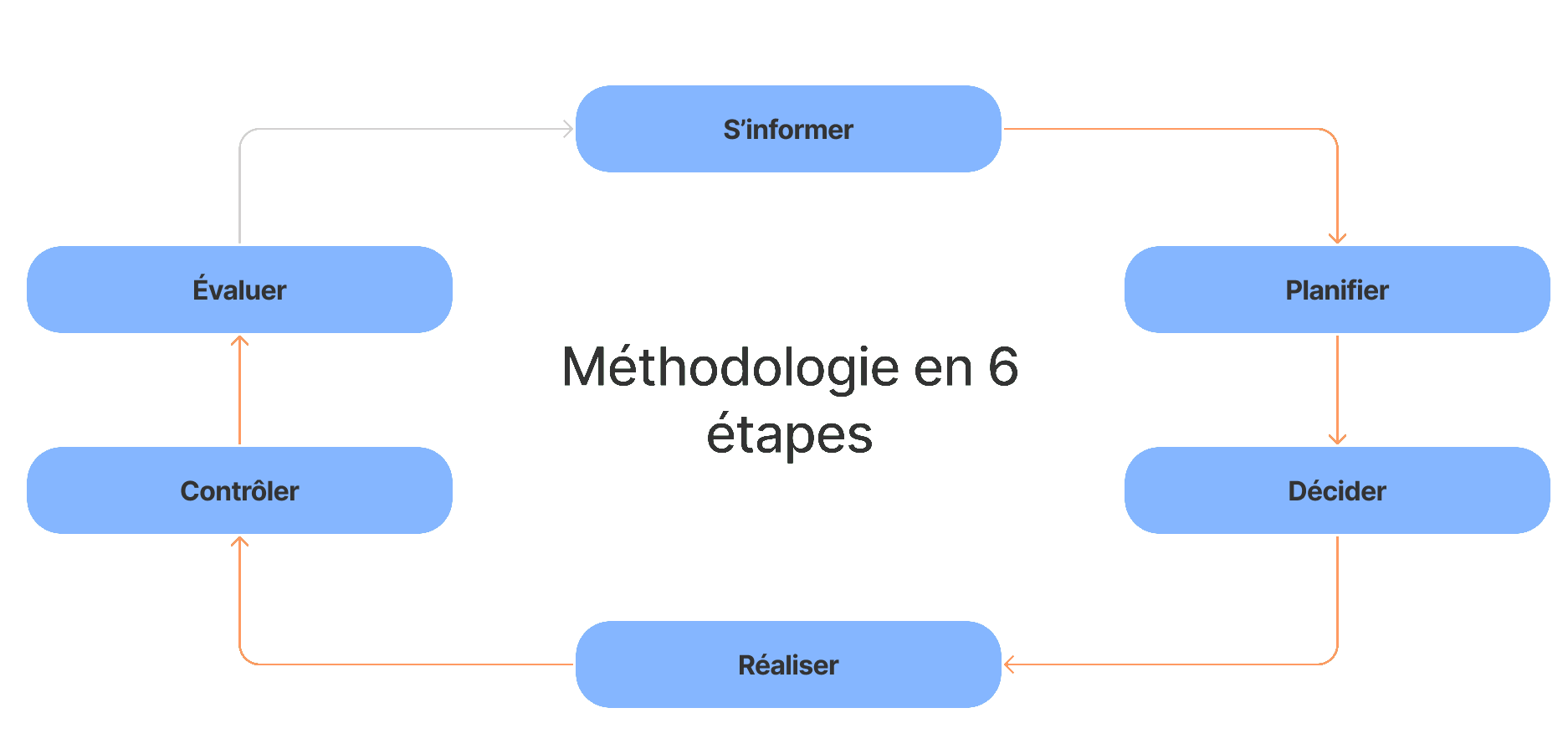
**A19** : L’architecture mise en place avant le TPI est respectée, et permet l’intégration avec les autres modules du projet.

**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 *user-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.

Une fois le **Backlog** terminé, j’ai mis en place mon planning prévisionnel afin de savoir quand est-ce que je pensais faire mes taches au départ du projet.

## Décider

Durant tout mon TPI, il a fallu que je prenne des décisions sur la manière de faire certaines choses. Je pense par exemple au changement d’email ou j’ai pris au du temps à prendre la décision sur quelle procédure je devrais utiliser pour être le plus efficace possible.

## 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 parties et fonctionnalités du site ont été testés et documentés dans ce fichier afin que n’importe quelle personne, interne ou externe au métier, puisse les réaliser sans difficulté.

Chaque test a été effectué sous plusieurs navigateurs afin de vérifier qu’il n’y a pas de problèmes de compatibilités.

## Evaluer

La dernière étape est l’évaluation. Cette étape est incontournable pour analyser comment c’est dérouler le projet dans son ensemble et avoir une pensée critique sur la réalisation.

On peut se poser les questions suivantes :

* Qu’est-ce qui peut être améliorer ?
* Qu’est-ce qui a été positif ?
* Qu’est-ce qui a été négatif ?
* Comment le modifier ?

Le journal de bord est une pièce très importante pour la réalisation de cette étape. Il nous permet de revoir comment nous avons fait chaque chose et nos problèmes.

# 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.

## Antidote

Antidote est un correcteur, de langue, avancé avec filtres intelligents. Il permet de facilement corriger ses fautes de français et même d’autres langues.

# Architecture

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

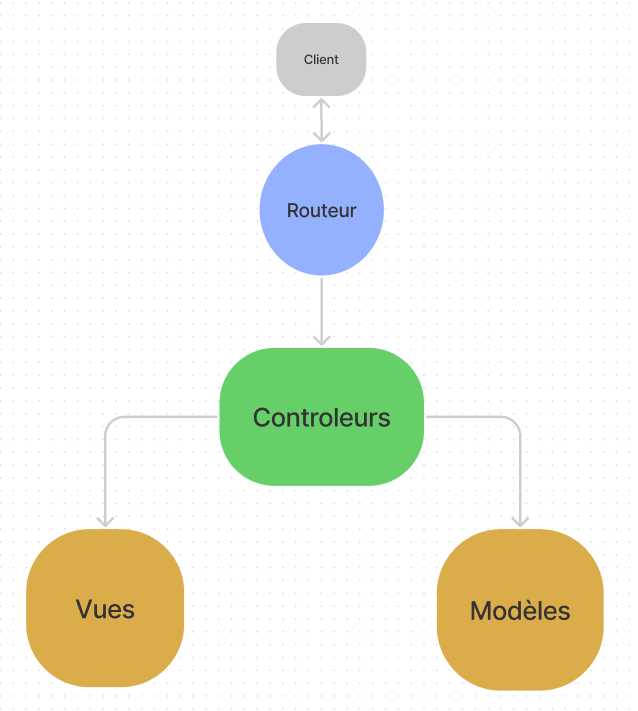
* Le routeur
* Les contrôleurs
* Les modèles
* Les vues

Le routeur est le point central du projet. Il permet la redirection sur les différentes pages selon les actions de l’utilisateur. Cela permet de gérer les accès plus facilement.

Les contrôleurs s’occupent de la liaison entre le modèle et la vue. Il effectue le traitement des données reçu par les vues (formulaires) et les modèles afin de mettre à jour la vue si nécessaire ou la base de donnée.

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

Les modèles contiennent les données que l’on va manipuler. Ils vont assurer leurs gestions. C’est le modèle qui contient les données de la base



# Analyse organique

## Fonctionnalités intégrés

#### Enregistrement d’un utilisateur

La première fonctionnalité de ma partie est l’enregistrement d’un utilisateur. Chaque nouvelle personne arrivant sur le site est considérée comme une personne avec le rôle « Anonymous ». Ses personnes ont donc la possibilité de s’inscrire sur le site en renseignant les informations suivantes :

* Prénom
* Nom
* Email
* Adresse
* Mot de passe

L’utilisateur doit confirmer son mot de passe afin que cela soit confirmé et qu’il ne se soit pas trompé.

#### Vérification de l’email

Après son inscription, l’utilisateur crée peut désormais se connecter. Mais il possède le rôle « NotVerified » ce qui le bride dans l’utilisation du site.

Afin de se débarrasser de ce rôle, il doit vérifier son email. Pour cela, un mail lui a été envoyé lors de son inscription pour la valider.

Il lui suffit de cliquer sur le lien, ce qui le redirigera vers une page du site. Sur cette page se trouve un simple formulaire comprenant un champ pour mettre son email, ainsi qu’un champ en lecture seule avec la clé de vérification.

Il doit remplir le champ avec l’email qu’il a fourni afin de vérifier que cela ne soit pas quelqu’un de malveillant essayer de mettre des clés aléatoires.

Une fois l’email fournis, il clique sur le bouton valider et son compte sera mis à jour. Il pourra désormais apercevoir que son compte possède à présent le rôle « Customer ».

Pour cela, il dispose de 24h au maximum, après son compte pourra être supprimé par les admin et il ne pourra plus le valider.

#### La connexion

Après l’enregistrement, l’utilisateur peut se connecter. Si son email est vérifié, il possède le rôle « Customer ». Sinon, son rôle est « NotVerified ».

Pour se connecter, il doit cliquer sur le bouton « Se connecter » puis d’entrer son email ainsi que son mot de passe.

#### Affichage et modification du profil

Toutes les personnes connectées ont un accès à leur profil.

Pour tout le monde, le profil affiche les informations personnelles de l’utilisateur. C’est-à-dire son nom, son prénom, son adresse, son email. Le mot de passe n’est pas affiché dans un but de sécurité.

Pour les personnes bannies ou non vérifiées, l’accès à la modification de valeurs est impossible.

La modification des informations personnelles est disponible pour tous les autres rôles.

Pour modifier son nom, son prénom, son adresse, il suffit de modifier les champs et de cliquer sur le bouton « Modifier », disponible que si nos rôles le permettent.

Pour le mot de passe, il y a 3 champs dans la page. Un pour vérifier que l’on connaisse bien le mot de passe actuel, deux autre pour mettre le nouveau mot de passe. Si tous les champs sont bien remplis et que le mot de passe actuel est correct, lorsque l’on cliquera sur le bouton « Modifier », le mot de passe sera changé.

Pour la modification de l’email, c’est un peu plus sécurisé. Lorsque l’on modifie le champ de l’email, cela va vérifier si l’email n’est pas déjà pris dans la base. S’il n’est pas déjà utilisé, un mail sera alors envoyé, avec un lien, sur notre email pour confirmer que nous voulons vraiment changer d’email. Lorsque l’on clique sur le lien, on est redirigé sur une page nous demandant notre email actuel suivis de notre nouvel email et une clé de validation.

Lorsque vous rentrez votre email actuel puis votre nouvel email, cliquez sur modifier et elle sera alors modifier. Vous recevrez un courriel de confirmation.

#### Récupération du mot de passe

Sur la page de connexion, il y a la possibilité de cliquer sur « Mot de passe oublié » ce qui permet de renseigner son email afin de faire une demande de modification de mot de passe.

Pour cela, lorsque vous rentrez votre email, vous allez recevoir un courriel avec un lien. Si vous cliquez sur le lien, vous aurez la possibilité de rentrer un nouveau de passe. Et comme pour les autres modifications et vérifications, vous devez entrer votre email pour vérifier que c’est bien vous. Si toutes les données sont correctes, le mot de passe sera modifié.

#### Création/Affichage/Modification/Suppression des utilisateurs

Dans ce cite, uniquement pour les administrateurs, il est possible de voir la liste des utilisateurs. Grâce à cette liste, il a la possibilité de modifier leurs données. Ceux-ci seront prévenu par mail.

De plus, il peut aussi supprimer les utilisateurs qui n’ont pas valider leurs emails 24h après leurs demande d’inscription. Il a aussi la permission de retiré ceux qui n’ont pas de commandes en cours et qui sont inactifs depuis plus d’un an.

Pour finir, les admins peuvent créer un nouvel utilisateur. Cet utilisateur sera créé en bypassant les vérifications d’email.

#### Les logs

Il est aussi possible, via un compte admin de voir les logs du site. Les logs sont enregistrés dans la base de données lorsqu’un utilisateur essaie de se connecter 3x de suites ou plus avec le même email et qu’il échoue à chaque fois.

Elles sont aussi enregistrées à chaque demande de récupération d’email.

#### Création/Affichage/Modification/Suppression des catégories

Tous les utilisateurs du site peuvent voir les catégories. Ils ont la possibilité de soit voir la liste des catégories, soit afficher tous les objets contenus dans chacune d’entre-elle à l’aide d’un menu déroulant.

Le menu affiche les catégories selon les parents/enfants.

Une ligne d’Ariane permet de remonter facilement les catégories parentes de celle-ci.

Les WebManager peuvent, contrairement aux autres rôles, modifier, ajouter, supprimer les catégories. Le titre ne doit pas être vide.

## Diagrammes de classes

## Structure

Mon programme est développé avec une architecture MVC. Ce qui signifie que la séparation des fichiers est bien précise et permet une compréhension plus facile.

* annexes/ : ce dossier contient uniquement les documents tel que la documentations les plannings etc. Aucun lien avec le code.
* commons/ : L’use case de base du programme. Ce dossier était fourni avec l’énoncé et uniquement la classe Session a été modifié. Il a les fichiers utiles à la forme de base du site.
  + commons/controller : Ce sont tous les contrôleurs de base et nécessaires pour le projet (accessDenied, home, Routes).
  + commons/model : Possède les modèles FlashMessages, Menu, Session, DbConnection. Ils seront utiles pour tout le projet.
  + commons/views : Contient les vues pouvant être appelées à tout moment afin d’afficher une erreur ou des flashMessages etc…
* css/ : toute la css du projet
* db/ : contient une exportation de la base de donnée
* images/ : possède les images du site
* js/ : tous les scripts JavaScript du site
* uc/ : Tous les cas d’utilisation de la méthode MVC se trouve dans de dossier. A chaque fois qu’une nouvelle table de la base de données est utilisé dans le code, on créer un nouveau cas d’utilisation à l’intérieur. Tous les sous-dossiers de uc/ fonctionne de la même manière. Pour cela, je vais expliquer uniquement le fonctionnement général.
  + uc/\*/ : Pour chaque cas d’utilisation, on retrouve une architecture MVC ainsi qu’un register.php . Le register est indispensable car il permet d’enregistrer l’use case à l’index du site.
    - uc/\*/controllers : Le sous dossier controller contient tous les contrôleurs utiles pour manipuler les données de la base
    - uc/\*/model : Les modèles de chaque cas se trouvent dans le dossier modèle. Ils serviront à stocker les données ainsi qu’à les récupérer de la base et faire des manipulations dessus.
    - uc/\*/views : Toutes les vues de chaque use cases se trouvent dans ce dossier. Elles seront appelées afin d’afficher les données traitées.
* index.php : l’index est le routeur du projet. Il redirige vers chaque contrôleur selon la requête et importe tous les cas d’utilisations.

Tous les .htaccess ont uniquement « deny from all » comme contenu. Cela servira à bloquer tout personne essayant de se balader dans les données à l’aide de l’url ou autre. Uniquement l’index.php peut être appelés et il inclura les fichiers selon les requêtes.

# Plan de tests

J’ai décidé de mettre en place un plan de tests afin de vérifier plus facilement que tout fonctionne. De plus, il est rédigé afin que n’importe qui puisse exécuter les tests sans problèmes de compréhension.

|  |  |  |
| --- | --- | --- |
| **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 | Affichage d’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 | Affichage d’un message d’erreur |
| 5 | **Connexion :**Connexion avec un compte non vérifié | Connexion avec comme rôle « NotVerified » |
| 6 | **Connexion :** Connexion avec des mauvais identifiants | Affichage d’un message d’erreur |
| 7 | **Déconnexion :** Cliquer sur le bouton déconnexion | Déconnecte l’utilisateur et le redirige sur la page d’accueil |
| 8 | **Changement de rôle :** Cliquer sur un rôle non sélectionné et disponible | Change le rôle de l’utilisateur |
| 9 | **Vérification d’email :** Entre le bon email lors de la vérification | Valide le compte + message de succès |
| 10 | **Vérification d’email :** Entre un mauvais email lors de la vérification | Message d’erreur |
| 11 | **Vérification d’email :** Modifie le token lors de la vérification | Message d’erreur |
| 12 | **Changement de mot de passe :** Met son email dans le premier formulaire | Reçois un mail avec un lien |
| 13 | **Changement de mot de passe :** clique sur le lien | Redirigé vers un formulaire |
| 14 | **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 |
| 15 | **Changement de mot de passe :** met le mauvais email | Message d’erreur |
| 16 | **Profil :** affichage du profil | Affiche les données de l’utilisateur |
| 17 | **Profil :** Change le nom et/ou le prénom et/ou l’adresse puis clique sur modifier | Modifie les données et affiche un message de succès |
| 18 | **Profil :** change l’email puis clique sur modifier | Envoie un mail avec un lien. Le lien redirige sur une page |
| 19 | **Profil :** vérifie l’email en entrant sont email actuel et son nouvel email. | Modifie le mail du compte. Envoie un mail au nouvel email pour confirmer. |
| 20 | **Profil :** vérifie l’email en entrant un faux email actuel et son nouvel email. | Message d’erreur |
| 21 | **Profil :** modifie le token de validation | Message d’erreur |
| 22 | **Profil :** met son ancien mot de passe ainsi que le nouveau mot de passe | Modifie le mot de passe et affiche un message de succès. |
| 23 | **Profil :** met le mot de passe actuel mais pas de nouveau mot de passe | Message d’erreur |
| 24 | **Profil :** met le nouveau mot de passe mais pas l’actuel | Message d’erreur |
| 25 | **Profil :** met le mauvais mot de passe actuel | Message d’erreur |
| 26 | **Profil :** tente de modifier le profil avec le rôle NotVerified ou Banned | Impossible |
| 27 | **Catégories :** Voir l’arborescence des catégories dans le menu | Affiche l’arborescence dans le menu catégorie avec les sous menus etc. |
| 28 | **Catégories :** Clique sur une catégorie de l’arborescence | Affiche la page de la catégorie avec la ligne d’Ariane. |
| 29 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 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 |
| 04.05.2021 | 8 | Changement de rôle | OK |
| 04.05.2021 | 9 | Reçois un mail avec un lien+ redirection sur le 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 | Affiche les données de l’utilisateur | OK |
| 17.05.2021 | 17 |  |  |
|  | 18 |  |  |
|  | 19 |  |  |
|  | 20 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Conclusion

## Difficultés rencontrées

## Amélioration possible

## Bilan personnel

# Annexes

## Glossaire

## Sources

[Dossier annexes]

Code source

Manuel utilisateur

Résumé du projet

Planning