|  |  |
| --- | --- |
|  |  |
| Documentation technique |  |
|  |  |
|  | 02:05:2022  Matos |
|  | Machado David  Francisco Garcia |

1. Table des matières

[2. Tables des versions 3](#_Toc103158799)

[3. Introduction 3](#_Toc103158800)

[4. Rappel de l’énoncé 3](#_Toc103158801)

[5. Méthodologie 4](#_Toc103158802)

[6. Planification 6](#_Toc103158803)

[7. Généralités concernant l’implémentation 7](#_Toc103158804)

### Tables des versions

### Introduction

Voici ma documentation technique du projet *Matos.* Ce projet a été réalisé lors du Travail pratique individuel (TPI) en mai 2022. Il a été réalisé afin de pouvoir valider mes compétences en informatique au bout de quatre ans au Centre de Formation Professionnelle Technique du Petit-Lancy à Genève.

Matos est un site internet qui permet aux enseignants d’emprunter du matériel de l’école. En tant que visiteur il pourra seulement voir le matériel disponible. Connecter enseignant sur le site il pourra sélectionner le matériel pour l’emprunter et sera mis en attente que l’admin accepte. L’admin pourra ajouter ou modifier du matériel et accepter les prêts des enseignants mis en attente.

### Rappel de l’énoncé

#### Organisation

|  |
| --- |
| Élève |
| David Machado  David.mchdb@eduge.ch |

|  |
| --- |
| Responsable TPI |
| Francisco Garcia  Francisco.garcia@edu.ge.ch |

|  |  |
| --- | --- |
| Experts | |
| Yvan Poulin  Yvpoulin@gmail.com | Francesco Foti  Francesco.foti@devinfo.net |

#### Livrable

Pour les experts et le formateur par email :

* Planning détaillé du projet
* Rapport de projet contenant le

code source au format PDF

* Journal de bord
* Résumé du TPI(1 page en A4)

Pour le formateur uniquement :

* L’accès au repository distant du projet avec les droits de « clone »
* Un readme explicitant l’installation du projet en local
* Un dump de la base de données contenant la structure ainsi qu’un set de données de test

#### Matériel et logiciels à disposition

* Un PC standard école, 2 écrans
* Serveur SGBD à choix (EasyPHP, Wamp, Laragon, Adminr, phpMyAdmin, autre)
* Windows 10
* Visual Studio à jour
* Logiciel de création de schémas (Visio, Gliffy, autre)
* Outil de versionnage de code (Git, avec dépôt distant sur GitHub / Bitbucket / GitLab)
* Navigateur web (Mozilla Firefox/Google Chrome/Safari)
* Logiciel de création de maquettes d’interfaces utilisateur (Sketch, Adobe XD, Proto.io, autre)
* Outil bureautique à choix pour les documents (Google Docs, MSOffice, OpenOffice)
* L’étudiant est autorisé à utiliser son matériel personnel au besoin

#### Description de l’application

### Méthodologie

Afin d’être organisé tout au long de mon travail, j’ai décidé d’utiliser la méthode en 6 étapes que nous avons apprises lors de nos 4 années de formation.



Figure 1 Méthode en 6 étapes

#### S’informer

Pour cette première étape de la méthode en 6 étapes, j’ai commencé par m’informer, c’est-à-dire que j’ai dû lire et analyser mon énoncé afin de le décortiquer pour comprendre chaque tâche qui m’était demandée. Pour se faire, j’ai décider d’imprimer l’énoncé pour pouvoir avoir un visu constamment et rapide des points.

#### Planifier

J’ai dû établir un planning de travail afin de me faire une idée du temps que les tâches allaient me prendre. J’ai pris les éléments de l’énoncé.  **A compléter**

J’ai décidé de mettre un ordre de priorité sur chaque tâche afin d’avoir un ordre de réalisation de tâches, voici le code que j’ai instauré pour les priorités :

* P0 : Bloquant
* P1 : Critique
* P2 : Important
* P3 : Secondaire

Une fois mes user stories crées, je les ai intégrées à un product backlog, qui me permet d’avoir un détail de chaque tâche du projet.

J’ai ensuite mis en place un diagramme de Gantt, qui me permet par la suite de voir les différences entre que j’avais prévu et ce que réellement réalisé.

#### Décider

#### Réaliser

La quatrième étape est la réalisation. Elle consiste à exécuter les tâches que nous avons planifiées et décider au préalable. C’est en général l’étape la plus longue du projet, il faut se fier un maximum sur le plan établi.

Lors de la réalisation, j’ai tenu un journal de bord qui m’a servi à expliquer le fond de ma pensée de façon quotidienne. C’est aussi durant la réalisation que j’ai effectué le plus gros du travail. Pour ma part, la réalisation s’est bien passée, entre autres, car j’ai bien planifié le déroulement du projet.

#### Contrôler

Les contrôles se font lors de la cinquième étape. À ce moment-là, chaque User Story doit être complet, pour qu’elle soit ensuite vérifiée. Par exemple, nous pouvons vérifier si la description et le résultat de la tâche correspondent bien aux consignes établies. Il s’agit aussi d’effectuer un plan de test sur le projet en lui-même. Il est conseillé de faire effectuer les contrôles et les tests à des personnes tierces, car cela apporte de meilleurs résultats.

Pour ma part, j’ai choisi de faire des tests au milieu du projet. Parce que les fonctionnalités n’étaient pas compliquées au début et je n’ai pas eu d’erreur bloquant . J'avais aussi pensé à le faire uniquement à la fin de la conception de mon projet, mais je n'ai pas opté pour cette façon de faire, parce que j’allais oublier la plupart des tests.

#### Évaluer

La sixième et dernière étape est l’évaluation du travail. Lors de l’évaluation, nous passons en revue le déroulement du projet et nous repérons les aspects qui ont bien fonctionné et ceux qui ont moins bien fonctionné. De plus, pour ne pas avoir à refaire les mêmes erreurs, nous cherchons à connaitre les points qui pourraient être améliorés pour le prochain projet.

### 6.Planification

#### 6.1Product backlog

|  |  |
| --- | --- |
| **Nom** | **T1 système de login** |
| **Description (user story)** | en tant que visiteur permettre de s'inscrire ou de se connecter |
| **Critère d'acceptation** |  |
| **Priorité** | **P0** : Bloquant |
|  |  |
| **Nom** | **T2 affichage du matériel** |
| **Description (user story)** | en tant que visiteur et utilisateur, je dois voir le matériel disponible à emprunts |
| **Critère d'acceptation** |  |
| **Priorité** | **P0** : Bloquant |
|  |  |
| **Nom** | **T3 affichage du matériel par catégorie** |
| **Description (user story)** | en tant que visiteur et utilisateur, je dois voir le matériel par catégorie |
| **Critère d'acceptation** |  |
| **Priorité** | **P1** : Critique |
|  |  |
| **Nom** | **T4 affichage et modification du profil** |
| **Description (user story)** | en tant qu’utilisateur, je dois pouvoir voir mon profil et le modifier |
| **Critère d'acceptation** |  |
| **Priorité** | **P2** : Important |
|  |  |
| **Nom** | **T5 ajout, modification et suppression du matériel** |
| **Description (user story)** | en tant qu’administrateur, je peux ajouter, modifier et supprimer du matériel |
| **Critère d'acceptation** |  |
| **Priorité** | **P1** : Critique |
|  |  |
| **Nom** | **T6 accepter les prêts** |
| **Description (user story)** | en tant qu’administrateur, je dois accepter les prêts qui sont en attente |
| **Critère d'acceptation** |  |
| **Priorité** | **P1** : Critique |
|  |  |
| **Nom** | **T7 calendrier d'emprunt du matériel** |
| **Description (user story)** | en tant qu’utilisateur, lors du choix du matériel un calendrier s'affiche pour choisir la date de l'emprunt |
| **Critère d'acceptation** |  |
| **Priorité** | **P0** : bloquant |
|  |  |
| **Nom** | **T8 affichage des utilisateurs** |
| **Description (user story)** | en tant qu'administrateur on peut voir tous les utilisateurs |
| **Critère d'acceptation** |  |
| **Priorité** | **P3** : Secondaire |
|  |  |
| **Nom** | **T9 pouvoir ajouter un commentaire** |
| **Description (user story)** | en tant qu'utilisateur on peut ajouter un commentaire lors du rendu du matériels |
| **Critère d'acceptation** |  |
| **Priorité** | **P3** : Secondaire |

#### 6.2Diagramme de Gantt

Voici le diagramme utilisé lors de la conception du site. Il m’a permis d’être organisé ainsi que de savoir précisément les tâches à réaliser de chaque jour.

### 7. Généralités concernant l’implémentation

#### 7.1 librairies et outils externes utilisés

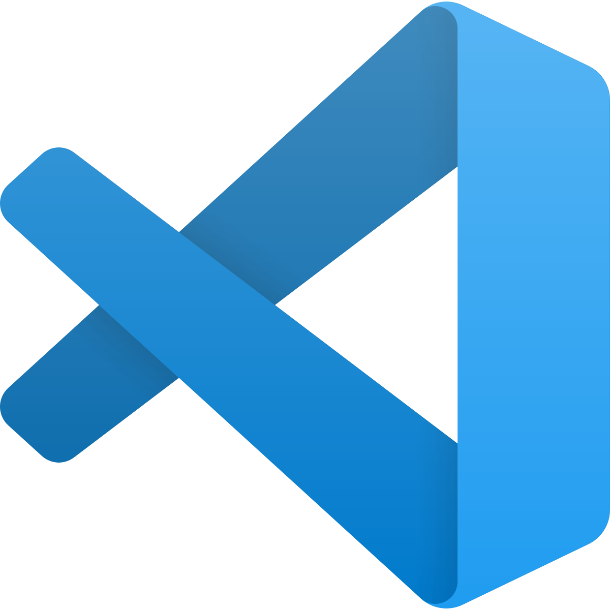


Figure 2 - icône de l'IDE Visual Studio Code

**Visual Studio Code** ou **VsCode** est un environnement de développement (IDE) développé par *Microsoft.* C’est selon moi le meilleur outil de développement on peut quasiment tout faire. Avec le nombre d’extensions qui existe.

Pour ma part, j’ai utilisé VsCode pour quelque des ces extensions. PHP getter and setter qui permet de crée automatiquement les get et set des propriétés de classe. Cela permet de gagner du temps. VsCode nous propose aussi des extensions pour l’utilisation d’un débogueur en PHP, ou bien de formater votre code se qui permet de bien structurer votre code, que j’ai évidemment utilisé.



Figure 3 - icône windows subsystem for linux

L’application Ubuntu on Windows ma permit de faire fonctionner localement des scripts PHP. J’ai installé tous les packages nécessaires pour crée un serveur LAMP comprenant les trois serveurs *apache2, mariaDB et PHP.* Il est simple d’utilisation si on a un minimum de connaissant de l’environnement linux.

Pour ma part, j’ai utilisé ce logiciel pour la simplicité et la rapidité en cas de problème on rapidement nos service *apache ou mariaDB.*

La version de PHP utilisé : PHP 8.1.5 .

La version de mariaDB utilisé : mysqld Ver 10.3.34-MariaDB-0ubuntu0.20.04.1 for debian-linux-gnu on x86\_64 (Ubuntu 20.04).

Figure 4 - icône du logiciel d'administration de base de données DBeaver

DBeaver est un logiciel client SQL qui offre un outil d'administration de bases de données. Il fournit un éditeur qui prend en charge la complétion de code et la coloration syntaxique. Son interface graphique permet de facilement effectuer des tâches qui peuvent sembler compliquées. Il offre également la possibilité d’afficher sous forme de diagramme une base de données.

Pour ma part, ce logiciel m’a permis de modifier la base de données avec facilité. Mais pour pouvoir créer et vérifier facilement mes requêtes avant de les implémenter dans mon code. J’ai directement fait avec Ubuntu je me suis connecter en ligne de commande à MySQL. l’éditeur script SQL je n’arrive pas à comprendre le fonctionnement.



Figure 5 - icône de l'outil de gestion de version GitHub

**Git** est un outil de gestion de version décentralisé, gratuit et open source. Il est simple à apprendre et permet de gérer du plus petit projet au plus grand. Pour exploiter cet outil, j’ai utilisé l’intermédiaire **GitHub** qui est service web d’hébergement de projet. De plus entant étudiant, je profite de l’intégralités qu’il propose.

Pour ma part, j’ai utilisé ces outils dans le but de pouvoir sauvegarder mes projets à un autre endroit que sur mon ssd.

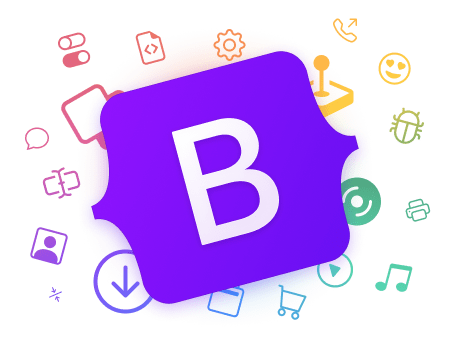


Figure 6 - icône du Framework CSS Bootstrap 5

Bootstrap 5 est un Framework CSS qui permet de concevoir et personnalisez très rapidement des sites web, en prenant compte de la taille de l’écran utilisé. C’est l’un des Framework CSS les plus utilisés au monde. Il est très complet et permet l’utilisation de variable, de système de grille responsive, ainsi que de nombreux composants préconstruits. Par conséquent, certaines de ces fonctionnalités utilisent de puissants plug-ins JavaScript.

Pour ma part, j’ai utilisé ce Framework CSS, car il m’a permis d’avoir un site web visuellement bon et rapide sans perdre beaucoup de temps. J’ai choisi ce Framework CSS, parce que j’ai déjà utilisé plusieurs fois dans plusieurs projets, ayant la main dessus, je tenais à le réutiliser. De plus il y a beaucoup d’exemple se qui permet de trouver la bonne class.

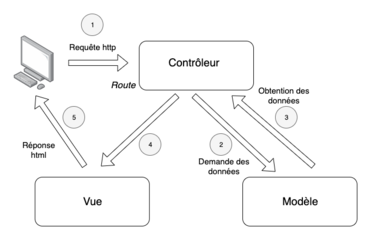
#### 7.2 Architecture du projet

7.2.1 modèle Vue Contrôleur

L’architecture de mon projet se base sur le motif Modèle-vue-contrôleur ou MVC. Comme on peut s’y attendre, ce type d’architecture sépare le projet en trois parties distinctes, la première étant le modèle, la seconde la vue et la dernière le contrôleur. Chaque de ces parties à sa propre fonction.

* Le modèle contient ou récupère les données à afficher.
* La vue contient la partie graphique du projet, autrement dit l’affichage.
* Le contrôleur contient la logique concernant les actions, c’est lui qui va filtrer les données et les analyser.

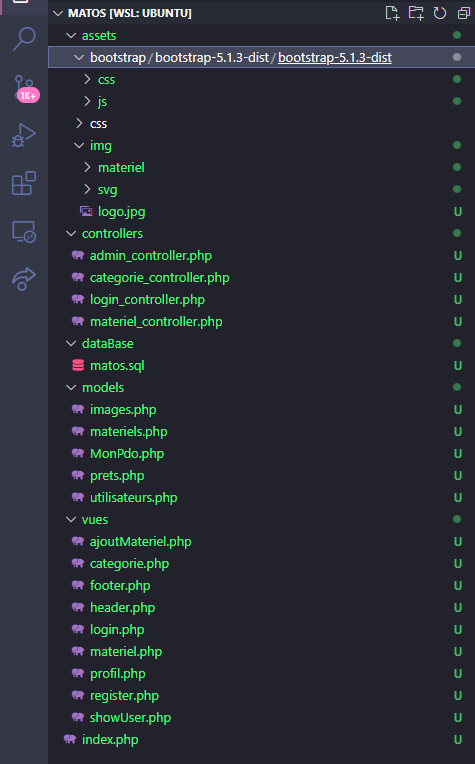
Voici une image qui schématise bien l’architecture MVC :



Si j’ai choisi cette façon de procéder, c’est parce que l’architecture MVC permet de séparer les couches, ce qui améliore considérablement la maintenabilité de l’application.

7.2.2 Arborescence

L’arborescence de mon projet s’organise autour de plusieurs répertoires importants :

* **asset**: contiens toutes les ressources du projet.
  + **Css**: comporte les fichiers de style du siteweb.
  + **Img**: inclus l’entièreté des images de l’application web.
* **Controllers** : englobe tous les contrôleurs du projet qui détiennent la logique du projet.
* **Database**: contiens la structure de la base de données et avec quelques données.
* **Models**: renferme les différents modèles du projet, ils disposent des données du projet.
* **Vues**: contiens les pages du projet, c’est lui qui fait le lien entre l’utilisateur et le site.

#### 7.3 bases de données

Le diagramme de base de données m’a été fourni par mon maitre d’apprentissage. J’ai alors modifié quelques champs. Elle est encodée en utf8 avec comme collation par défaut utf8\_general\_ci et utilise un moteur InnoDB.

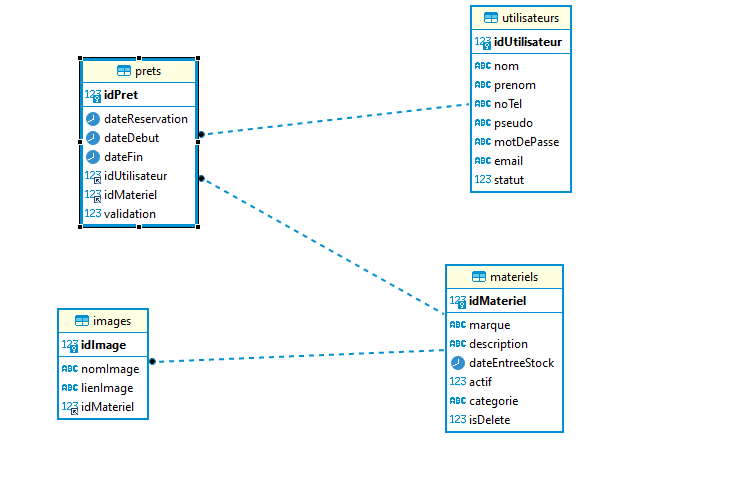


Figure 7 - modèle de la base de données de matos

#### 7.4 Fonctionnalités intégrées

Les fonctionnalités que j’ai implémentées dans mon projet sont décrites ci-dessous. Je les ai séparés par rapport aux pages sur lesquels elle agisse.