|  |
| --- |
| KuriMediation  (Gestionnaire de cas de médiations)  Expert 1 : M. Nicolas Borboën  Expert 2 : M. Bernard Oberson  Chef de projet : M. Dimitrios Lymberis  Sujet : Développement WEB  Framework : Laravel Breeze - Livewire |

Table des matières

[1 Introduction 2](#_Toc167198318)

[1.1 Titre 2](#_Toc167198319)

[1.2 Introduction 2](#_Toc167198320)

[2 Analyse / Conception 3](#_Toc167198321)

[2.1 Objectifs 3](#_Toc167198322)

[2.2 Planification initiale 4](#_Toc167198323)

[2.3 Méthodologie de travail 5](#_Toc167198324)

[2.4 Environnement 5](#_Toc167198325)

[2.4.1 Laravel 5](#_Toc167198326)

[2.4.2 MVC 5](#_Toc167198327)

[2.4.3 Eloquent (ORM) 6](#_Toc167198328)

[2.4.4 Breeze 6](#_Toc167198329)

[2.4.5 Tailwind CSS 6](#_Toc167198330)

[2.4.6 Livewire 6](#_Toc167198331)

[2.5 Conception 7](#_Toc167198332)

[2.5.1 Concept 7](#_Toc167198333)

[2.5.2 Modélisation de la base de données (Méthode MERISE) 8](#_Toc167198334)

[2.5.3 Justification des types de valeurs dans la modélisation 9](#_Toc167198335)

[2.5.4 Maquettes 10](#_Toc167198336)

[2.6 Stratégie de test 17](#_Toc167198337)

[2.7 Risques techniques 17](#_Toc167198338)

[2.8 Planification 17](#_Toc167198339)

[2.9 Dossier de conception 18](#_Toc167198340)

[3 Réalisation 19](#_Toc167198341)

[3.1 Dossier de réalisation 19](#_Toc167198342)

[3.2 Description des tests effectués 19](#_Toc167198343)

[3.3 Erreurs restantes 19](#_Toc167198344)

[3.4 Liste des documents fournis 19](#_Toc167198345)

[4 Conclusions 20](#_Toc167198346)

[5 Annexes 21](#_Toc167198347)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 21](#_Toc167198348)

[5.2 Sources – Bibliographie 21](#_Toc167198349)

[5.3 Journal de travail 21](#_Toc167198350)

[5.4 Manuel d'Installation 21](#_Toc167198351)

[5.5 Manuel d'Utilisation 21](#_Toc167198352)

[5.6 Archives du projet 21](#_Toc167198353)

# Introduction

## Titre

Réalisation d’une application web qui a pour but de gérer les cas de médiations au sein d’un établissement scolaire.

## Introduction

Ce projet consiste à mettre en place une application Web permettant de gérer les cas de médiation. Cette application est destinée aux médiateurs souhaitant effectuer un bilan en fin d’année du temps consacré aux médiations en répertoriant non seulement le nombre d’heures mais également le nombre de cas de médiation selon leurs types.

Le but de cette application est de simplifier et optimiser le processus de suivi des entretiens au sein de l’établissement scolaire pour les membres de l’équipe santé (conseillers d’orientation, médiateurs, psychologues, infirmières, etc..).

# Analyse / Conception

Dans ce projet, plusieurs fonctionnalités de bases apprises durant la formation seront à mettre en place. Des opérations CRUD devront être mises en place pour gérer les entretiens ainsi que les suivis, créer des utilisateurs qui auront le droit ou non d’accéder à des pages administratives, générer des graphiques exportables au format PDF basés sur les données récupérées d’entretiens et de suivis.

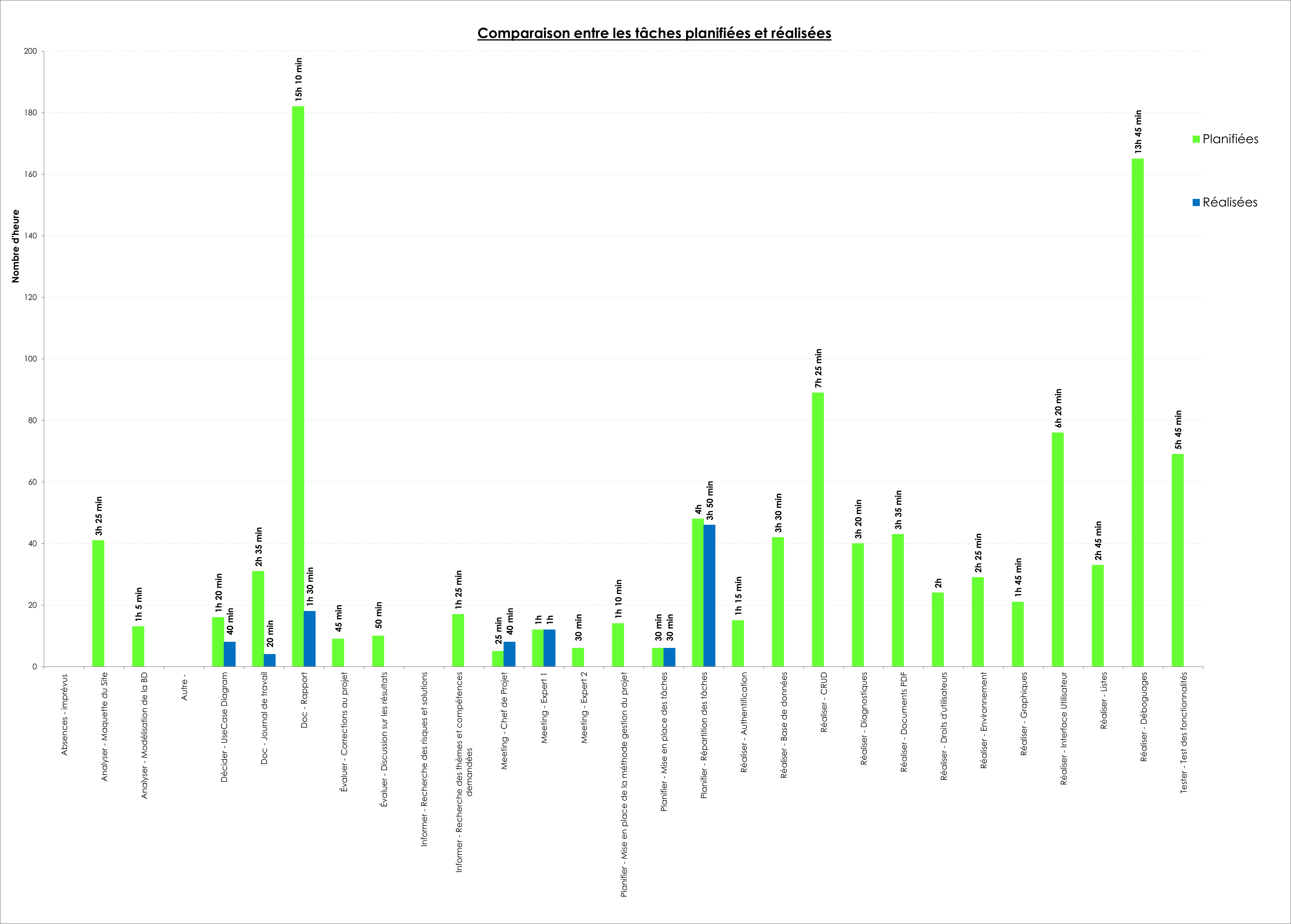
Après avoir validé l’application, elle sera déployée sur le serveur WEB mis en place par Monsieur Lymberis.

## Objectifs

Ce sous-chapitre est consacré aux objectifs demandés. C’est-à-dire, les fonctionnalités vues avec le maître de stage :

* Opérations CRUD pour les entretiens, les suivis et les années.
* Liste d’entretiens et de ses suivis.
* Filtrage d’entretiens par date, ordre alphabétique ou ordre anti-alphabétique.
* Graphiques représentant la somme de cas de médiations par types, du temps consacré aux entretiens par année, de médiations au total. (Camembert, Histogramme, Barres).
* Téléchargement/Téléversement de fichiers en format PDF.
* Exactitude des valeurs entrées dans les formulaires.
* Ajout/Modification/Suppression de documents liés à un entretien.
* Affichage d’erreurs.

## Planification initiale

Ce chapitre montre la planification du projet. Celui-ci peut être découpé en tâches qui seront planifiées. Il s'agit de la première planification du projet, celle-ci devra être revue après l'analyse. Cette planification sera présentée sous la forme d'un diagramme.

## Méthodologie de travail

Afin de réaliser ce projet de TPI, la méthode qui sera utilisée est la méthode des six pas qui est une méthode très convenable dans le cadre où une limite de temps est imposée.

## Environnement

### Laravel

Laravel[[1]](#footnote-1) est un framework PHP open-source, destiné au développement d’applications web en suivant l’architecture MVC (Model – View - Controller) crée par Taylor Otwell. Ce framework fait parti d’un des frameworks PHP les plus populaires grâce à sa simplicité d’utilisation, ses diverses fonctionnalités intégrées et une syntaxe très simpliste. De plus, il évolue à constamment et possède une communauté très active, ce qui fait que la documentation et l’aide en cas de soucis est fortement disponible.

### MVC

Le modèle MVC[[2]](#footnote-2) (Model – View – Controller) représente l’architecture centrale utilisé surtout en Laravel. Il vise à simplifier le travail en divisant la charge du travail d’une application web.

Le modèle (Model) représente la partie logique, c’est-à-dire, tout ce qui concerne, la gestion, la récupération, la manipulation ainsi que la sauvegarde de données.

La vue (View) représente la partie visuelle, c’est-à-dire, l’affichage de la page web ainsi que les données. Elle concerne principalement l’interface utilisateur.

Le controlleur (Controller) représente l’intermédiaire entre le modèle et la vue. Par exemple pour la création d’un utilisateur. Il reçoit les entrées à travers des formulaires puis les traites en interagissant avec le modèle puis, renvoie les résultats à la vue pour qu’ils puissent être affichés.

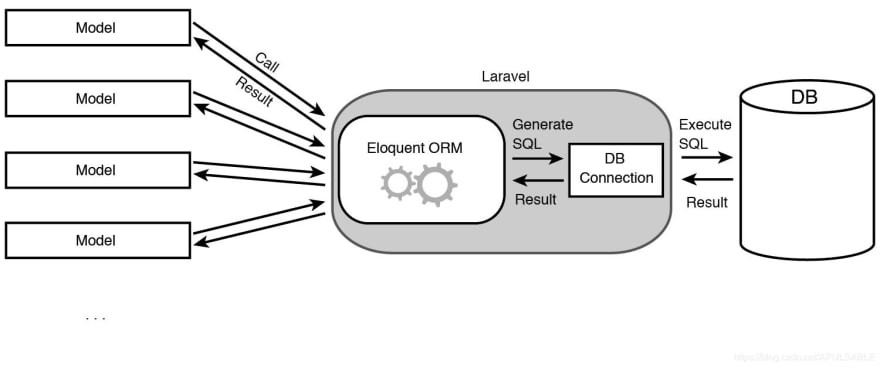
Cette division des tâches facilite énormément le développement, l’évolution et la maintenance des applications.



1 Représentation du fonctionnement du MVC

### Eloquent (ORM)

Eloquent[[3]](#footnote-3) est un ORM (Object-Relatinal Mapping) intégré à Laravel. C’est un programme qui se place entre une application web et une base de données afin de permettre aux développeurs de travailler avec les données sous formes d’objet à la place d’écrire des requêtes SQL.



2 Fonctionnement du Eloquent ORM

### Breeze

Breeze [[4]](#footnote-4)ou Laravel Breeze est un composant supplémentaire de Laravel qui offre une implémentation très simpliste et minimaliste de fonctionnalités d’authentification. Il a pour but de servir de point de départ pour les applications web qui ont besoins de fonctionnalités de bases telles que l’enregistrement, la connexion, la modification de données utilisateurs, la suppression de compte, la vérification d’email etc…

Beeze inclut déjà Tailwind CSS qui est un framework CSS rendant la stylisation de vues plus facile et rapide.

### Tailwind CSS

Tailwind CSS[[5]](#footnote-5) est comme mentionné ci-dessus, un framework CSS qui permet de rendre plus facile la stylisation de vue en évitant de passer par un fichier css.

### Livewire

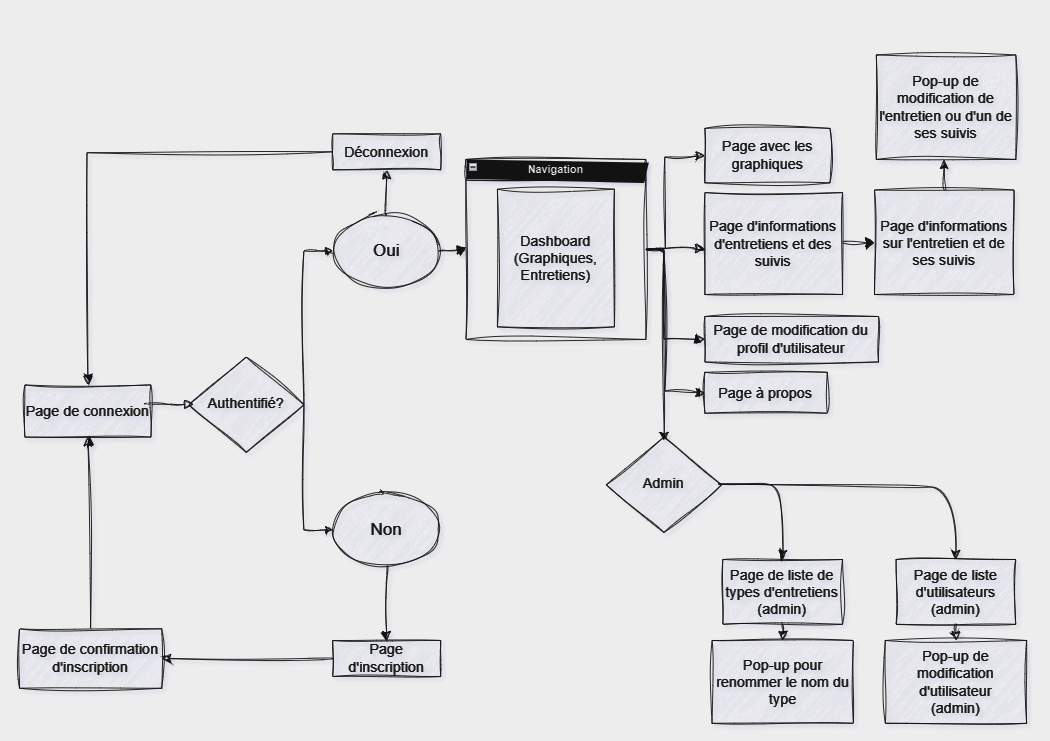
Livewire[[6]](#footnote-6) est un framework full-stack pour Laravel. Cela facilite la création d’interfaces utilisateur dynamiques directement dans Laravel sans avoir nécessairement besoin de connaissances en JavaScript. Par exemple, à la place de rediriger à chaque appui de boutons vers une autre page contenant un formulaire, avec Livewire, c’est possible d’afficher ou masquer le formulaire sur la page.

## Conception

Ce chapitre expliquera comment le site fonctionnera à l’aide d’un schéma de principe, des modèles MCD et MLD et des maquettes du site web.

### Concept

Le schéma ci-dessous représente le fonctionnement du site web.



3 Schéma fonctionnel du site

### Modélisation de la base de données (Méthode MERISE)

Les MCD et le MLD ci-dessous représentent la structure de la base de données utilisée dans le projet. Ces deux modèles ci-dessous respectent les conventions de nommage de l’ETML.

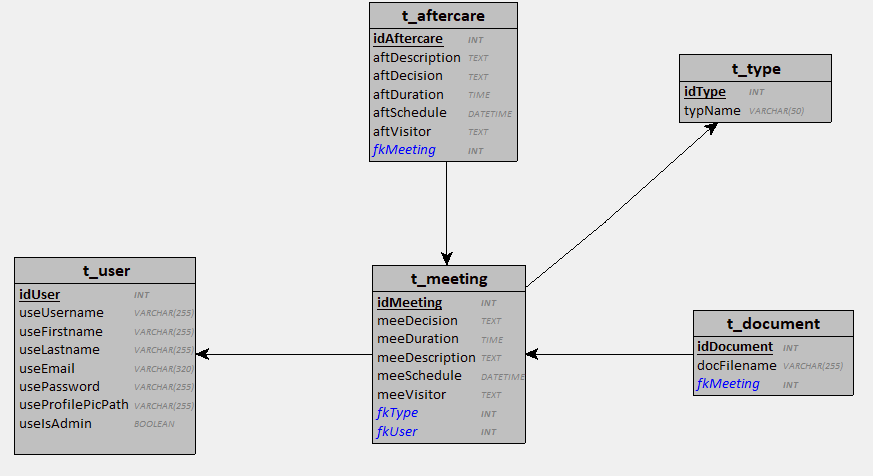
Cependant, pour une raison de compatibilité avec le framework utilisé, c’est-à-dire, Laravel, les noms des tables et des colonnes ont été adaptés à ses conventions de nommage.

Les entités « year » et « visitor » ont été supprimés car l’année pourrait tout simplement être récupérée dans les propriétés « schedule », puis, « visitor » est remplacé par un attribut dans les tables « meeting » et « aftercare »

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

Description générée automatiquement

4 MCD de l'application (Normes ETML)



5 MLD de l’application (Normes ETML)

### Justification des types de valeurs dans la modélisation

### Maquettes

L’application Web devra être responsive, donc des formats mobiles et desktop sont disponibles.

#### Desktop

##### Une image contenant texte, capture d’écran, conception Description générée automatiquementPages de connexion/inscription

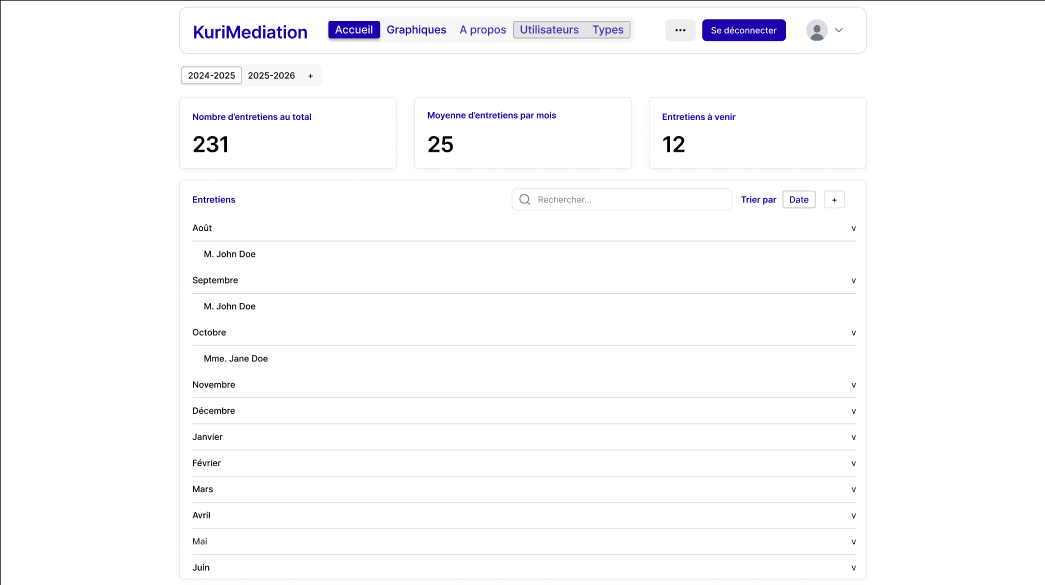
6 Page de connexion

Une image contenant texte, capture d’écran, conception

Description générée automatiquement

7 Page d'inscription

##### Page d’accueil



8 Page d'accueil

##### Une image contenant texte, nombre, logiciel, Police Description générée automatiquementPage de graphiques

9 Page de graphiques

##### Page de modification du profil

Une image contenant texte, capture d’écran, nombre

Description générée automatiquement

10 Page de modification du profil

##### Page d’information sur l’entretien

Une image contenant texte, logiciel, Page web, capture d’écran

Description générée automatiquement

11 Page d'informations sur un entretien

##### Formulaire d’ajout d’un nouvel entretien

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

Description générée automatiquement

12 Formulaire d'ajout d'un nouvel entretien

##### Formulaire d’ajout d’un nouveau suivi

Une image contenant texte, logiciel, capture d’écran, Page web

Description générée automatiquement

13 Formulaire d'ajout d'un suivi

#### Mobile

##### Page de connexion/inscription (Mobile)

*Une image contenant texte, capture d’écran, conception

Description générée automatiquement*

14 Pages d'inscription et de connexion (Mobile)

##### Page d’accueil avec barre de navigation (Mobile)

*Une image contenant texte, capture d’écran, nombre, logiciel

Description générée automatiquement*

15 Pages d'accueil/Barre de navigation (Mobile)

##### Page de graphiques (Mobile)

**Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement**

16 Graphiques (Mobile)

##### Page de modification du profil (Mobile)

##### **Une image contenant texte, capture d’écran, logiciel, conception Description générée automatiquement**

17 Modification du profil (Mobile)

##### Page d’informations sur l’entretien (Mobile)

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

18 Information sur l'entretien (Mobile)

##### Formulaire d’ajout d’un nouvel entretien (Mobile)

Une image contenant texte, capture d’écran, logiciel, conception

Description générée automatiquement

19 Formulaire Entretien (Mobile)

##### Formulaire d’ajout de suivi (Mobile)

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquement

20 Formulaire Suivi (Mobile)

## Stratégie de test

*Décrire la stratégie globale de test:*

* *types de des tests et ordre dans lequel ils seront effectués.*
* *les moyens à mettre en œuvre.*
* *couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).*
* *données de test à prévoir (données réelles ?).*
* *les testeurs extérieurs éventuels.*

## Risques techniques

* *risques techniques (complexité, manque de compétences, …).*

*Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, …).*

## Planification

*Révision de la planification initiale du projet :*

* *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
* *partage des tâches en cas de travail à plusieurs.*

*Il s’agit en principe de la planification* ***définitive du projet****. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l’historique.*

## Dossier de conception

*Fournir tous les document de conception:*

* *le choix du matériel HW*
* *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
* *le choix des outils logiciels pour la réalisation et l'utilisation*
* *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, …*
* *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
* *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme…*

# Réalisation

## Dossier de réalisation

*Décrire la réalisation "physique" de votre projet*

* *les répertoires où le logiciel est installé*
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*
* *le numéro de version de votre produit !*
* *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

*NOTE : Evitez d’inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n’incluez que cette partie…*

## Description des tests effectués

*Pour chaque partie testée de votre projet, il faut décrire:*

* *les conditions exactes de chaque test*
* *les preuves de test (papier ou fichier)*
* *tests sans preuve: fournir au moins une description*

## Erreurs restantes

*S'il reste encore des erreurs:*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

## Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

* *le rapport de projet*
* *le manuel d'Installation (en annexe)*
* *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
* *autres…*

# Conclusions

*Développez en tous cas les points suivants:*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

## Sources – Bibliographie

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)… Et de toutes les aides externes (noms)*

## Journal de travail

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Durée** | **Activité** | **Remarques** |
|  |  |  |  |
|  |  |  |  |

## 

## Manuel d'Installation

## Manuel d'Utilisation

## Archives du projet

*Media, … dans une fourre en plastique*

1. Documentation officielle de Laravel : [https://laravel.com/docs/11.x](https://laravel.com/docs/11.x/installation#why-laravel) [↑](#footnote-ref-1)
2. Documentation du MVC : [https://openclassrooms.com/fr/courses/4670706](https://openclassrooms.com/fr/courses/4670706-adoptez-une-architecture-mvc-en-php/7847928-decouvrez-comment-fonctionne-une-architecture-mvc) [↑](#footnote-ref-2)
3. Documentation sur Eloquent : <https://laravel.com/docs/11.x/eloquent> [↑](#footnote-ref-3)
4. Documentation sur Breeze : <https://laravel.com/docs/11.x/starter-kits#laravel-breeze> [↑](#footnote-ref-4)
5. Documentation sur Tailwind CSS : <https://tailwindui.com> [↑](#footnote-ref-5)
6. Documentation sur Livewire : <https://laravel-livewire.com> [↑](#footnote-ref-6)