Thierry Koetschet

Chemin du perrey 22 | 1670 Ursy

thierry.koetschet.1998@gmail.com

SI-ca2a

Application web de fitness

rapport de tpi

C:\Users\thierry.koetschet\Documents\GitHub\TPI_Thierry_Koetschet\logo_infofit.png

Table des matières

[1 Analyse préliminaire 3](#_Toc134091537)

[1.1 Introduction 3](#_Toc134091538)

[1.2 Organisation 4](#_Toc134091539)

[1.3 Méthode de gestion de projet 5](#_Toc134091540)

[1.4 Objectifs 5](#_Toc134091541)

[1.5 Planification initiale 6](#_Toc134091542)

[1.6 Structure du dossier 7](#_Toc134091543)

[2 Analyse / Conception 8](#_Toc134091544)

[2.1 Concept 8](#_Toc134091545)

[2.1.1 Modèle conceptuel de données 8](#_Toc134091546)

[2.1.2 Modèle logique de données 9](#_Toc134091547)

[2.2 Stratégie de test 16](#_Toc134091548)

[2.3 Risques techniques 16](#_Toc134091549)

[2.4 Planification 17](#_Toc134091550)

[2.5 Dossier de conception 17](#_Toc134091551)

[3 Réalisation 17](#_Toc134091552)

[3.1 Dossier de réalisation 17](#_Toc134091553)

[3.2 Description des tests effectués 18](#_Toc134091554)

[3.3 Erreurs restantes 19](#_Toc134091555)

[4 Conclusions 19](#_Toc134091556)

[5 Annexes 20](#_Toc134091557)

[5.1 Cahier des charges 20](#_Toc134091558)

[5.2 Résumé du travail 20](#_Toc134091559)

[5.3 Sources – Bibliographie 21](#_Toc134091560)

[5.4 Journal de travail 22](#_Toc134091561)

[5.5 Glossaire 22](#_Toc134091562)

[5.6 Manuel d'installation 22](#_Toc134091563)

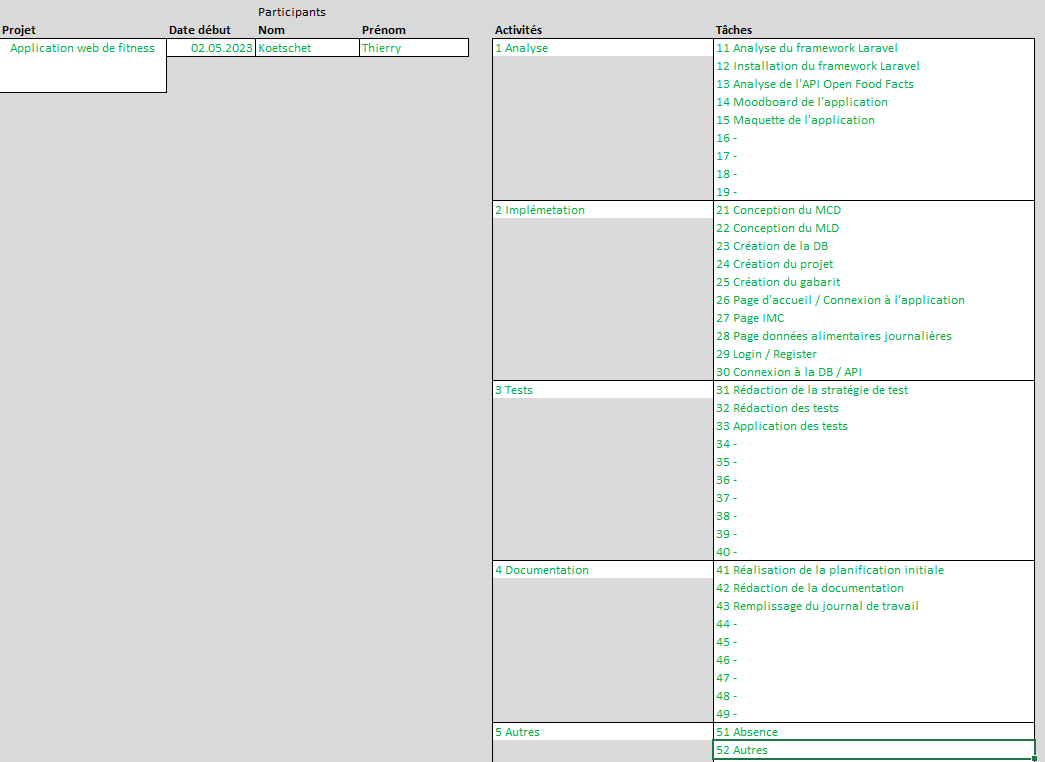
# Analyse préliminaire

## Introduction

Ce rapport va décrire en détail la réalisation de mon projet de TPI sur une application web de fitness. Cette application permettra aux utilisateurs de s’authentifier grâce un compte et d’accéder à ses fonctionnalités. La première fonctionnalité est le calcul de l’indice de masse corporelle de l’utilisateur grâce à son poids, sa taille et son genre. La deuxième fonctionnalité est un calendrier permettant d’enregistrer les aliments consommés quotidiennement par l’utilisateur afin de calculer le total des calories et des macronutriments journaliers. Toutes ces informations sur les différents aliments absorbés seront accessibles grâce à une API publique. L’application doit être développée avec l’aide d’un framework PHP et avoir une structure de type MVC.

La raison de mon choix de partir sur un tel projet s’explique par le fait que le développement est mon domaine de prédilection en informatique, et spécifiquement le développement web, dans lequel j’aimerais idéalement poursuivre ma carrière professionnelle. De plus, lors des débuts de ma formation d’informaticien à Sainte-Croix, j’ai également développé un attrait particulier pour le sport et la musculation. C’est ainsi que m’est venu l’idée de combiner ces deux domaines pour réaliser un projet intéressant et qui pourrait également m’être utile dans ma quête du corps de mes rêves.

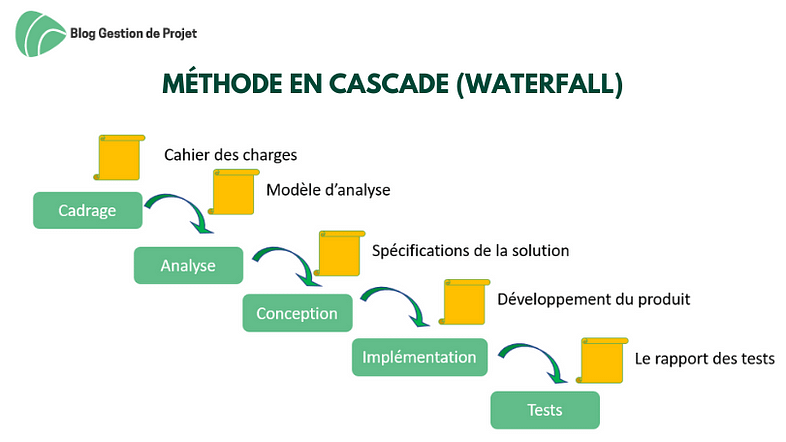
## Organisation



Dans mon organisation personnelle pour la réalisation de mon projet de TPI, j’ai décidé de séparer toutes mes tâches en quatre sprints. Le premier sprint regroupe toutes les tâches concernant la partie analytique de mon travail. Les tâches contenues dans le second sprint concernent la partie pratique du travail d’implémentation de l’application. Le troisième sprint représente la partie tests du travail et enfin le dernier sprint est un peu plus particulier car il contient le travail de rédaction de la documentation et de remplissage du journal de travail qui doit être fait du début à la fin du projet.

## Méthode de gestion de projet

J’ai choisi la méthode en cascade pour la gestion de mon projet de TPI. Je trouve que pour un projet en solo, cette méthode répond parfaitement à mes besoins en terme de gestion de projet.



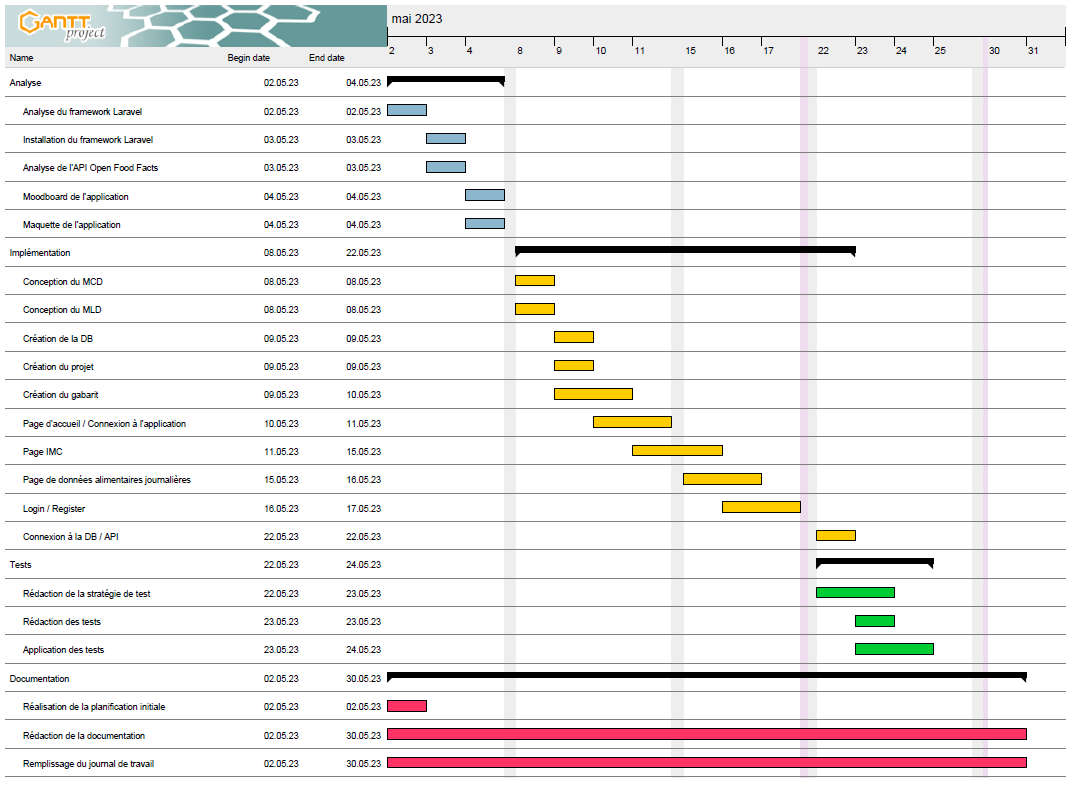
Les cinq étapes de la méthode en cascade correspondent parfaitement à la vision que j’ai pour la réalisation d’un tel projet. En effet, le cadrage en début de projet est primordial afin de structurer notre travail et de ne pas trop s’éparpiller. La planification initiale représente parfaitement cette première étape car elle permet de poser une valeur de temps sur les différentes à réaliser dans la suite du projet. La partie analyse correspond au travail préliminaire à réaliser avant de se lancer dans la conception comme l’analyse du framework à utiliser ou encore l’API recommandée dans le cahier des charges. Après la partie analyse, j’ai réalisé la conception de mon application grâce à un moodboard, une maquette, un modèle conceptuel de données et un modèle logique de données. L’étape suivante est d’implémenter le site web, c’est-à-dire écrire tout le code nécessaire à son bon fonctionnement. Enfin, les tests permettent de vérifier le fonctionnement général de l’application.

## Objectifs

Les objectifs fixés dans le cahier des charges par le chef de projet et les experts sont de réaliser un site web avec l’utilisation d’un framework PHP. L’application web doit également avoir plusieurs fonctionnalités telles qu’un authentificateur sécurisé, un calculateur d’indice de masse corporelle montrant l’évolution du poids des utilisateurs grâce à un graphique et un calendrier permettant de contrôler son alimentation utilisant une API publique.

Personnellement l’objectif principal que j’aimerais atteindre à la fin de ce travail est la maîtrise d’un framework PHP car je pense que cela pourrait peser dans la balance dans la recherche d’un futur emploi dans le développement web.

## Planification initiale

.

## Structure du dossier

Ce dossier se divise en quatre parties principales.

La première partie se compose de l’analyse préliminaire de ce travail, c’est-à-dire d’une introduction, d’une explication de l’organisation du projet, une description des objectifs visés et puis un bref aperçu de ma planification initiale.

La deuxième englobe toute la partie analyse et conception du projet avec l’élaboration du concept, de la stratégie de test, un compte rendu des risques techniques, une révision de la planification initiale du projet et enfin la partie conception du projet.

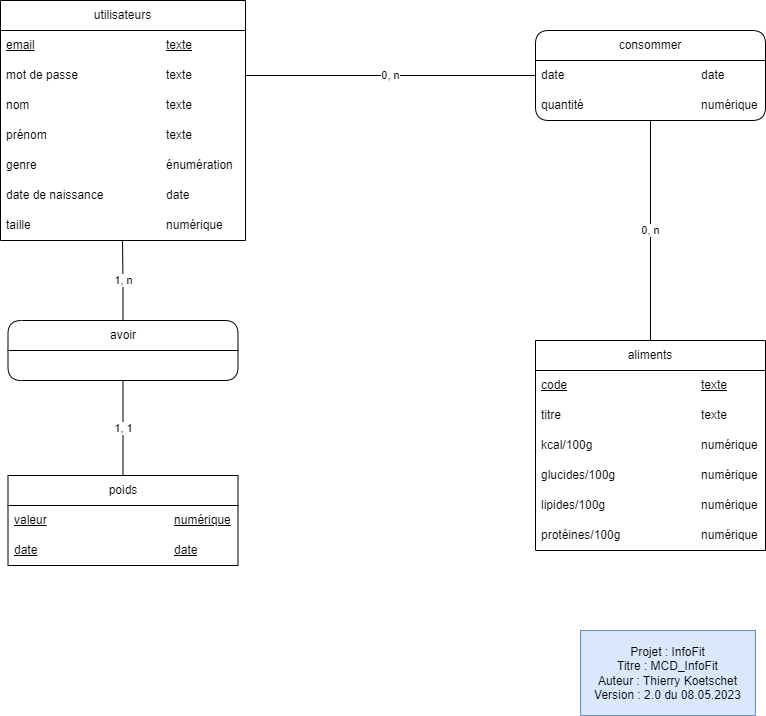
La troisième partie représente toute la réalisation pratique du projet commençant par lister tous les fichiers du dossier de réalisation, puis une description des tests effectués et des erreurs restantes et finalement une énumération des documents fournis à la remise du projet.

La dernière décrit les conclusions auxquelles je suis arrivé à la fin de ce projet telles que les objectifs atteints ou non, mon ressenti au fil du projet, les difficultés rencontrées et les améliorations que je pourrais apporter si je devais refaire un tel projet.

# Analyse / Conception

## Concept

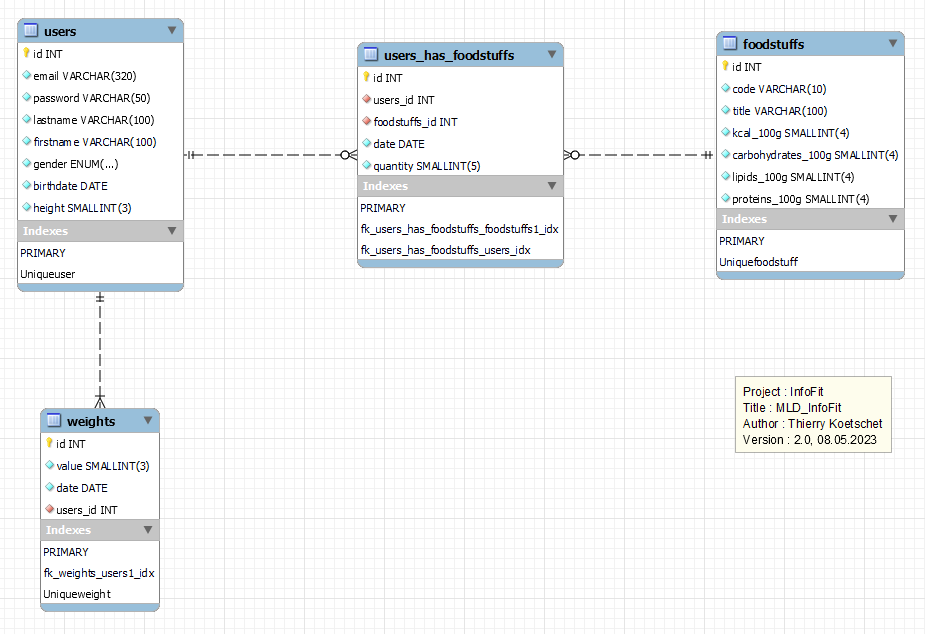
### Modèle conceptuel de données



Le MCD a été créé avec draw.io. Il s’agit d’un logiciel libre d’édition graphique de diagrammes.

Au lieu d’avoir un simple champ poids dans l’entité « utilisateurs », j’ai ajouté une entité supplémentaire « poids » avec comme attributs « valeur » et « date » afin de permettre aux utilisateurs de mon application de modifier leur poids au fur et à mesure de leur évolution et d’en avoir un historique.

### Modèle logique de données



Le MLD a été réalisé à l’aide de MySQL Workbench afin de pouvoir générer automatiquement le script de création de la base de données.

### Base de données

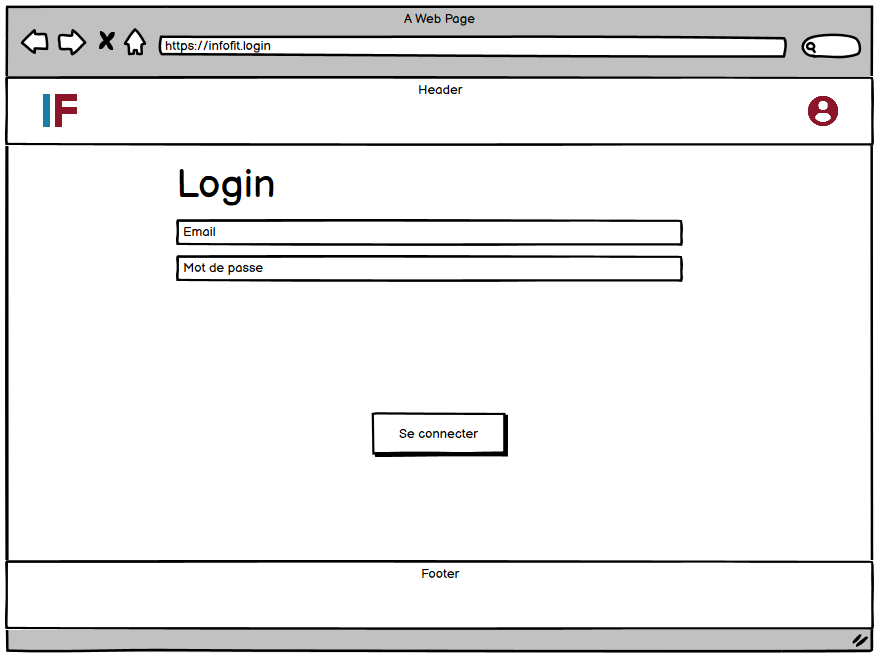
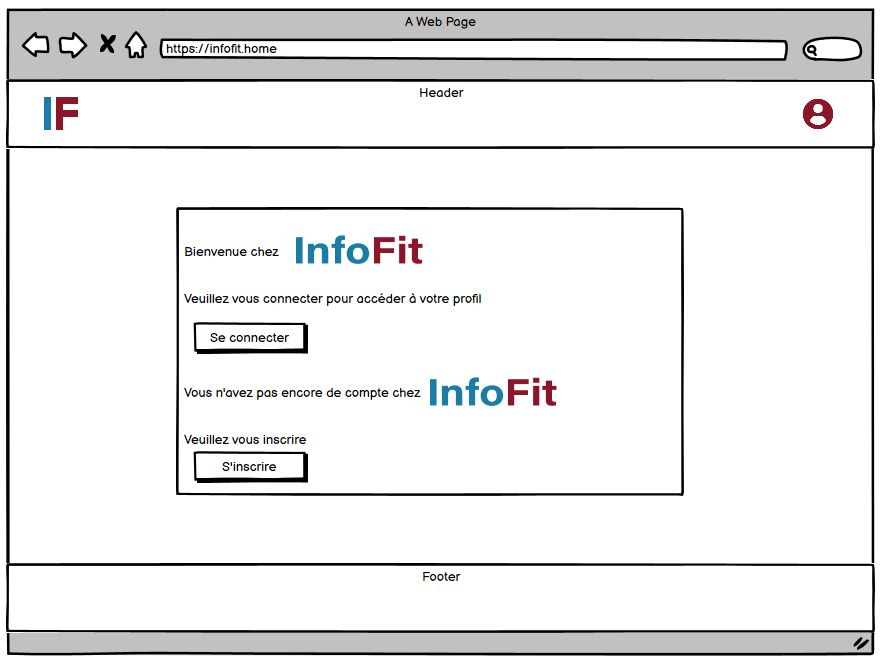
Voici ci-dessus le script de création de la base de données « infofit » également accessible dans mon dépôt github personnel (<https://github.com/ThierryKoetschet/TPI_Thierry_Koetschet>). Il y a quelques différences par rapport au MCD et au MLD. Par exemple, on peut remarquer que j’ai dû ajouter à chaque table un attribut « updated\_at » et un attribut « created\_at » au format « timestamp » afin que je puisse insérer des données dans la DB grâce à mon application car Laravel l’exigeait. J’ai également modifié le type des attributs « kcal\_100g », « carbohydrates\_100g », « lipids\_100g » et « proteins\_100g » de la table « foodstuffs » en « float » car j’ai remarqué que les données retournées par l’API Open Food Facts étaient parfois des nombres à virgules et posaient des problèmes pour l’insertion des données dans la base.

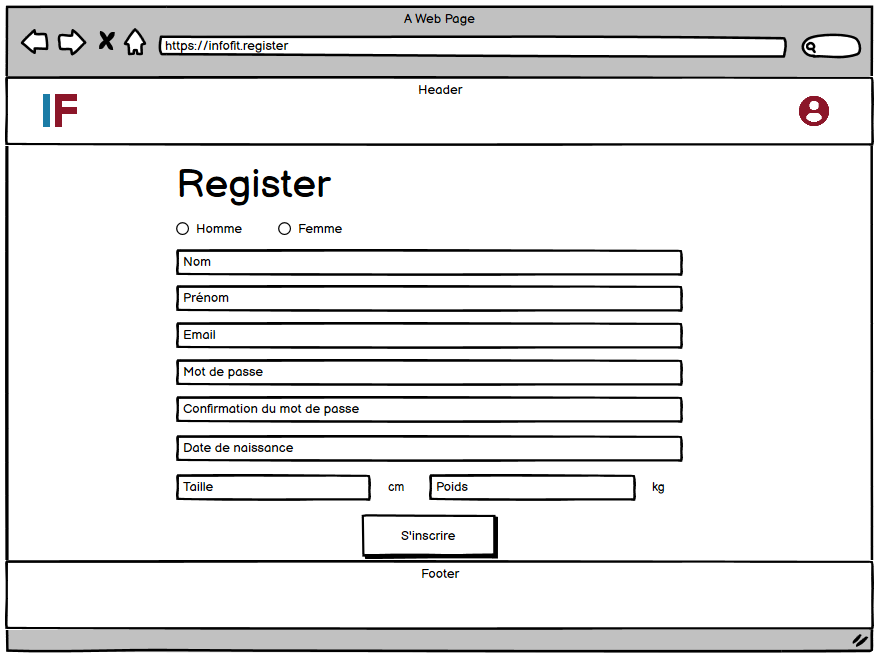
### Moodboard

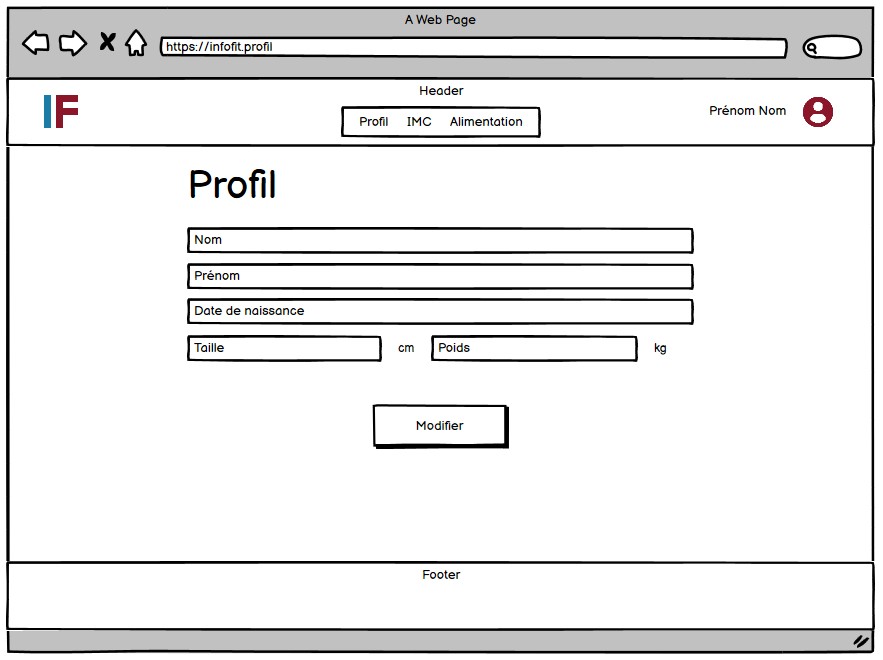


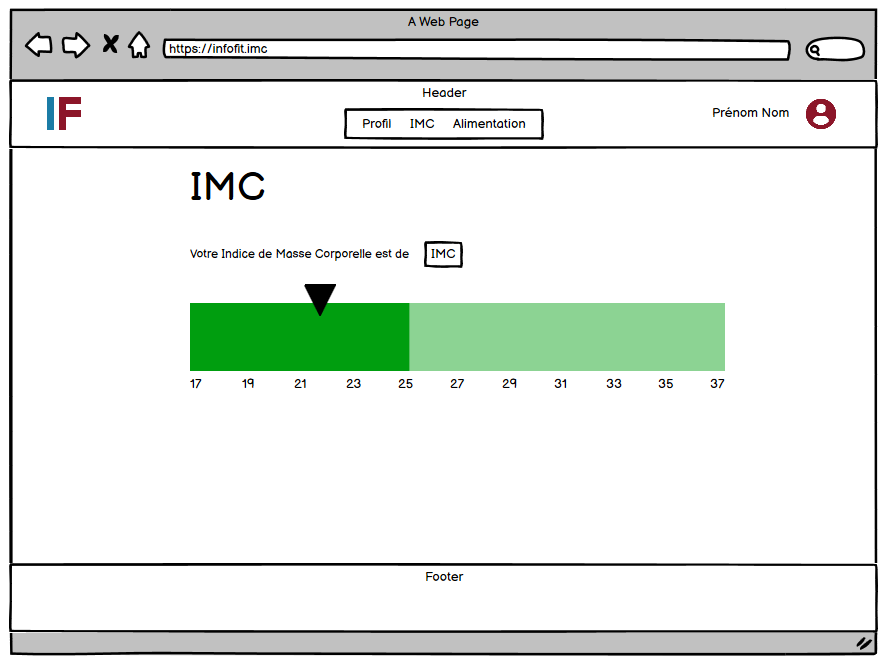
Le moodboard a été réalisé à l’aide de photoshop. Il a principalement servi de croquis pour créer la palette de couleur et les logos utilisé dans le site web.

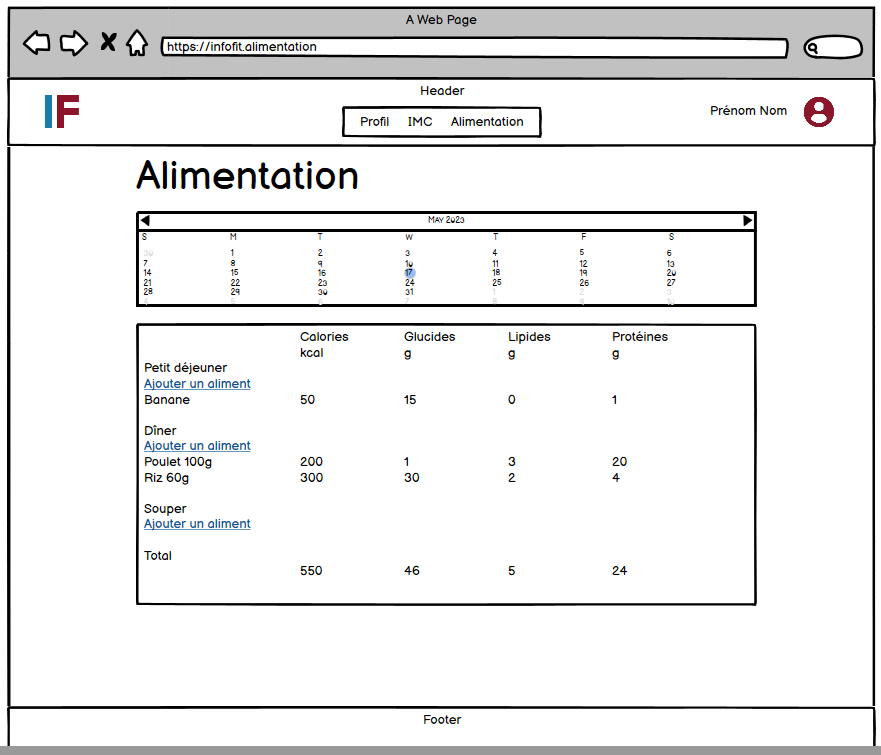
### Maquettes

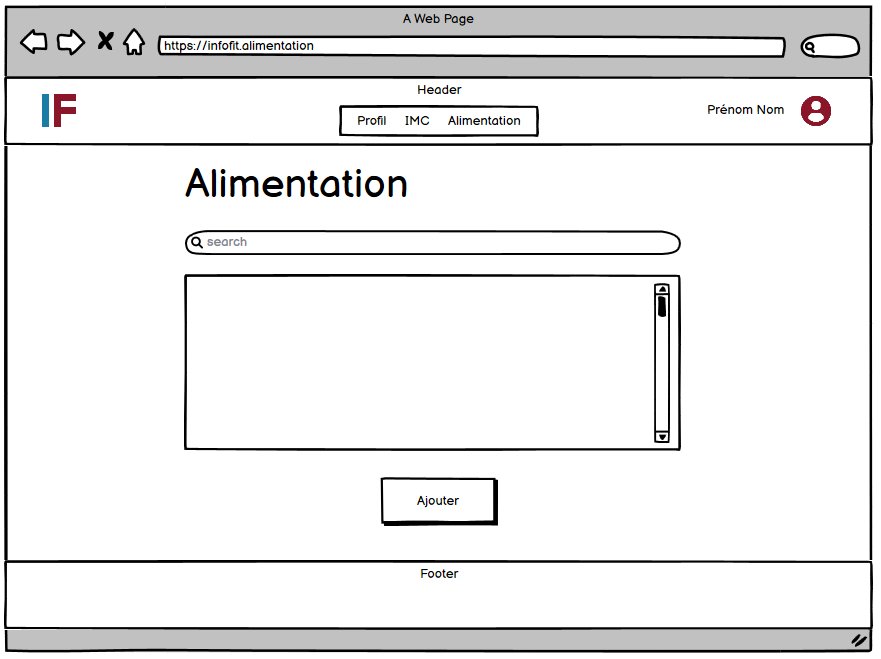












Ces maquettes m’ont servi de base tout au long de l’implémentation de mon application, cependant dans la réalisation de certaines fonctionnalités, je me suis rendu compte que je ne pouvais pas copier la maquette complètement. Par exemple, la page sur l’indice de masse corporelle ne ressemble pas à la maquette car j’avais tout d’abord oublié d’y ajouter le graphique d’évolution du poids, puis j’ai également réalisé que mon envie d’implémenter un slider, qui n’était pas mentionner dans mon cahier des charges, allait me demander trop de temps.

## Stratégie de test

Dans le cadre de ce projet, je pense qu’une stratégie relativement simple est un choix judicieux. En effet, l’application est composée de trois principales fonctionnalités : un moyen d’authentification en tant qu’utilisateur et de saisie de données personnelles, un calculateur d’indice de masse corporelle et un tableau permettant d’enregistrer la consommation alimentaire journalière de l’utilisateur. Afin de vérifier le fonctionnement de l’application et de la base de données, il suffit d’effectuer des tests fonctionnels sur ces trois fonctionnalités en commençant par l’authentification car les deux autres fonctionnalités ont besoin de certaines informations enregistrées par l’utilisateur.

Je vais également tester les routes de mon application, c’est-à-dire que tous les liens fonctionnent correctement et que toutes les pages du site sont accessibles.

J’ai créé un utilisateur dont je me suis servi au cours de la réalisation de mon application contenant déjà une certaine quantité d’informations. Celui-ci sera utilisé au dans la réalisation de mes tests. Voici ces identifiants :

* Email : [thierry.koetschet@cpnv.ch](mailto:thierry.koetschet@cpnv.ch)
* Mot de passe : Pa$$w0rd

## Risques techniques

Le risque technique principal est l’apprentissage d’un nouveau framework. En effet, j’ai choisi de réalisé mon site web avec le framework PHP Laravel avec lequel je n’avais encore jamais travaillé. L’apprentissage de l’utilisation d’un tel outil est assez fastidieux et demande beaucoup de temps. Sachant que le temps est une denrée précieuse dans le TPI, il va s’en dire que je prends un paris osé en partant sur cette voie. L’une des raisons qui m’a orienté sur cette décision est que l’un de mes camarades de classe, Pablo Zubieta, avait déjà réalisé son projet de pré-TPI avec Laravel et me l’a vivement recommandé. Il m’a également fait une présentation de l’outil au début du TPI ce qui n’a fait que renforcer mon choix.

Un autre risque auquel je fais généralement face dans de tels projets est que j’ai tendance à trop vouloir en faire, c’est-à-dire à me rajouter du travail supplémentaire, pas forcément nécessaire, simplement car j’ai envie de rendre un produit fini le plus complet possible. Généralement, ce travail supplémentaire impact le temps total à disposition et fait que la fin du projet est un peu tendue.

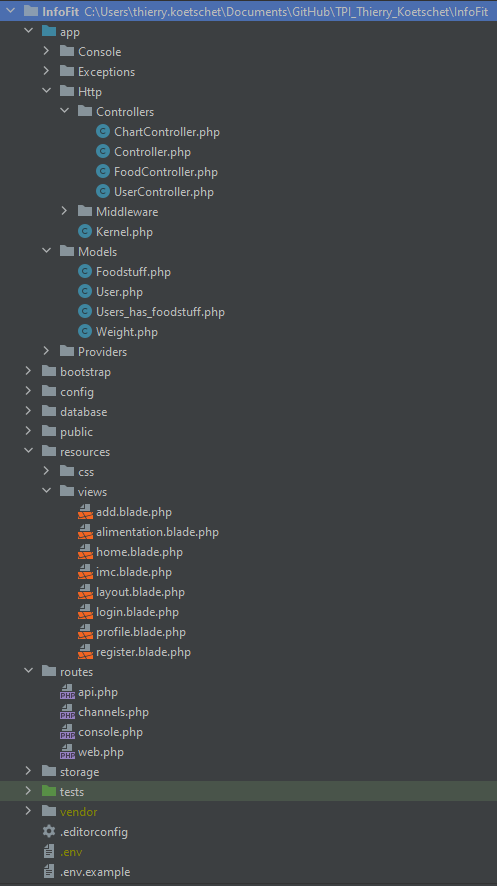
Le fait de transcrire ces risques techniques de manière écrite me permet d’en prendre conscience et d’y faire particulièrement attention pendant la durée du travail de TPI.

## Planification

## Dossier de conception

# Réalisation

## Dossier de réalisation



Vous trouverez ci-dessus une capture d’écran du dossier de mon application « InfoFit » avec les principaux dossiers et fichiers visibles.

Tout d’abord, le dossier InfoFit\resources\**views** contient tous les fichiers impactant le front end du site web. Il contient par exemple le gabarit (layout.blade.php) qui affiche le header et le footer de chaque page du site. Le reste des fichiers de ce dossier sont les différentes pages du site comme la page d’accueil (home.blade.php) ou encore la page d’enregistrement d’un utilisateur (register.blade.php). Ces fichiers sont principalement composés de code en HTML et CSS avec un peu de JavaScript et de PHP. Cependant, la majorité du CSS du site est contenu dans les fichiers **styles.css** et **charts.css** du dossier InfoFit\public\css. Les images, donc le logo et les différentes icônes, sont quant à elles contenues dans le dossier InfoFit\public\assets.

Un autre fichier important est le fichier InfoFit\routes\**web.php** car il contient toutes les routes de l’application, c’est-à-dire qu’il permet de faire le lien entre toutes ces vues et permet la transmission de données d’une page à l’autre.

## Description des tests effectués

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Test | Résultat | Validation |
| Tests unitaires | | | |
| 28.03.2023 |  |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 28.03.2023 |  |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 28.03.2023 |  |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 28.03.2023 |  |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 28.03.2023 |  |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 28.03.2023 |  |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| Tests d’intégration | | | |
| 28.03.2023 |  |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| Tests fonctionnels | | | |
| 28.03.2023 |  |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 28.03.2023 |  |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 28.03.2023 |  |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 28.03.2023 |  |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 28.03.2023 |  |  | Croix-Rouge Marque PNG Transparent - PNG All |
| 28.03.2023 |  |  | Free symbole de coche ronde avec fond transparent 17177933 PNG with Transparent  Background |
| 28.03.2023 |  |  | Croix-Rouge Marque PNG Transparent - PNG All |
| 28.03.2023 |  |  | Croix-Rouge Marque PNG Transparent - PNG All |

## Erreurs restantes

## Liste des documents fournis

# Conclusions

Pour conclure, je pense avoir atteint les objectifs fixés autant dans mon cahier des charges comme les objectifs personnels que je m’étais fixés au début du projet. En effet, toutes les fonctionnalités décrites dans le cahier des charges semblent être respectées. Le site contient un système d’authentification sécurisé, la page de calcul d’indice de masse corporelle fonctionne correctement et enfin la page permettant aux utilisateurs de l’application d’enregistrer leurs données alimentaires journalières est opérationnelle avec des requêtes API allant chercher des informations sur l’API Open Food Facts. Je pense également avoir acquis de nombreuses connaissances sur l’utilisation du framework Laravel qui était mon objectif personnel à atteindre. J’espère que ces connaissances auront une utilité dans ma future carrière professionnelle.

Cependant, je me rends compte qu’il y a certains points de la réalisation de projet que je pourrais améliorer comme la gestion de mon temps qui pourrait être meilleure. La fin de ce projet de TPI fut quelque peu intensive car j’ai consacré un peu trop de temps à l’implémentation de l’application ce qui fait que j’ai manqué de temps pour la rédaction de la documentation et ai dû faire un peu de travail à domicile.

En somme, je suis satisfait du produit que je remets car il s’agit d’un outil je vais utiliser quotidiennement dans mon activité sportive afin de suivre mes progrès et mes informations personnelles. Et comme mon ambition future est de potentiellement lancer mon entreprise d’informatique, je pourrai m’inspirer et utiliser tous ces outils appris lors de ce projet de TPI.

# Annexes

## Cahier des charges

## Horaire de travail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Lundi | Mardi | Mercredi | Jeudi | Vendredi |
| 08:00 – 09:35 |  |  |  |  |  |
| 09:50 – 12:15 |  |  |  |  |  |
| Pause midi | | | | | |
| 13:30 – 15:05 |  |  |  |  |  |
| 15:20 – 16:55 |  |  |  |  |  |

## 

## Résumé du travail

Ce projet de TPI est l’achèvement d’une formation de deux ans en tant qu’informaticien d’entreprise et consiste en la réalisation d’un travail pratique d’environ 90 heures. Mon cahier des charges me fixait comme objectifs de réaliser un site web avec l’utilisation d’un framework PHP et devant contenir trois principales fonctionnalités : La première fonctionnalité demande à l’utilisateur de s’authentifier afin de pouvoir accéder aux deux autres fonctionnalités. La seconde permet aux utilisateurs de calculer leur indice de masse corporelle en fonction du poids saisi dans leur profil et d’avoir une représentation graphique de leur évolution. La troisième un genre de calendrier alimentaire permettant aux utilisateurs de récupérer de nombreuses informations sur les différents aliments qu’ils ont consommé dans leur journée avec l’aide de requêtes API.

## Sources – Bibliographie

Site Yii Framework :

<https://www.yiiframework.com/> consulté le 02.05.2023

Site Laravel :

<https://laravel.com/> consulté le 02.05.2023

Site Open Food Facts :

<https://ch-fr.openfoodfacts.org/> consulté le 02.05.2023

Site MyFitnessPal :

<https://www.myfitnesspal.com/fr> consulté le 02.05.2023

Site Composer :

<https://getcomposer.org/download/> consulté le 02.05.2023

API denrées alimentaires :

<https://developer.edamam.com/food-database-api-docs> consulté le 02.05.2023

<https://world.openfoodfacts.org/data> consulté le 04.05.2023

Source d’information pour la stratégie de tests :

https://www.atlassian.com/fr/continuous-delivery/software-testing/types-of-software-testing consulté le 09.05.2023

Liens pour templates HTML/CSS :

<https://startbootstrap.com/themes> consulté le 09.05.2023

Liens pour graphiques JS :

[https://developers.google.com/chart?hl=fr](https://developers.google.com/chart/interactive/docs/gallery?hl=fr) consulté le 16.05.2023

## Journal de travail

## Glossaire

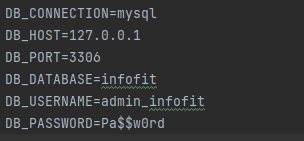
|  |  |
| --- | --- |
| API | Application Programming Interface est un ensemble de composants logiciels facilitant le développement d’application. |
| CRUD | Acronyme pour « Create Read Update Delete » représentant la relation entre la base de données et l’application. |
| CSS | Cascading Style Sheets est language de programmation utile à la mise en page d’un site web. |
| DB | Database ou base de données |
| Framework | Ensemble de composants logiciels servant à la programmation d’application. |
| Front end | Eléments visibles d’une site web |
| HTML | Hypertext Markup Language est un langage de programmation très utilisé dans le développement web. |
| IDE | Environnement de développement |
|  |  |
|  |  |
|  |  |
| Moodboard | Planche graphique d’ambiance ou d’inspiration. |
| MVC | Model View Controller est une manière d’organiser la structure d’un projet de développement. |
| OS | Système d’exploitation |
| PHP | Langage de programmation principalement utilisé en web. |
| SGBD(R) | Système de gestion de bases de données (relationnelles) |
| Use case | Description écrite de la réalisation de tâches utilisée dans le développement logiciel. |
|  |  |

## Manuel d'installation

Tout d’abord, ce manuel d’installation explique les étapes nécessaires au lancement du site InfoFit de manière locale et dans un système d’exploitation Windows 10.

Afin de pouvoir ouvrir l’application InfoFit en local, plusieurs installations sont nécessaires. La première installation à faire est la librairie php disponible sur le site <https://www.php.net/>. La deuxième installation à effectuer est Composer (<https://getcomposer.org/>), pour ce faire, ouvrez le fichier php.ini de la librairie php que vous venez d’installer et décommentez la ligne « **extension=php\_fileinfo** », puis enregistrez le fichier. Ensuite, saisissez la commande **composer** **install** dans une invite de commande.

Une étape importante pour que l’application fonctionne correctement est de créer la base de données localement. Vous pouvez utiliser le SGBDR de votre choix ; j’ai personnellement utilisé Mariadb. Veuillez créer une nouvelle base de données nommée « infofit » et un utilisateur propre à cette base et y possédant tout droits possibles. Exécutez ensuite le scripte de création de ma base de données disponible sur mon dépôt GitHub (<https://github.com/ThierryKoetschet/TPI_Thierry_Koetschet>) sous le nom de **script\_creation\_db.sql**. Enfin, modifiez le fichier **.env.example** du projet InfoFit pour qu’il corresponde au informations que vous avez saisi lors de la création de la base de données comme l’exemple ci-dessous et renommez-le en **.env**.



Pour faciliter le lancement de l’application, je conseille l’installation d’un environnement de développement supportant le langage php. Par exemple, Visual Studio Code ou encore PhpStorm fonctionnent parfaitement et j’ai personnellement utilisé dans la réalisation de mon projet l’IDE PhpStorm. Ouvrez donc le projet InfoFit de mon dépôt GitHub dans l’IDE de votre choix ouvrez un terminal. Placez vous à la racine du projet avec un commande cd et saisissez ensuite la commande **php artisan serve** afin de lancer l’application de manière locale.