El Letche



Vitor Silva & Ricardo Delgado Miranda

ETML, Lausanne

Cindy Hardegger

Table des matières

[1 Introduction 3](#_Toc60063991)

[2 Analyse 3](#_Toc60063992)

[2.1 Pages du site 3](#_Toc60063993)

[2.2 Base de données 4](#_Toc60063994)

[2.3 Structure 4](#_Toc60063995)

[3 Réalisation 4](#_Toc60063996)

[3.1 Gestion des images 4](#_Toc60063997)

[3.2 Eco-conception 4](#_Toc60063998)

[3.3 Ajout d’une recette 4](#_Toc60063999)

[4 Tests 4](#_Toc60064000)

[5 Conclusion 4](#_Toc60064001)

[5.1 Conclusion générale 4](#_Toc60064002)

[5.2 Conclusions Personnelles 4](#_Toc60064003)

[5.2.1 Vitor Silva 4](#_Toc60064004)

[5.2.2 Ricardo Delgado Miranda 4](#_Toc60064005)

[5.3 Méthode de projet 4](#_Toc60064006)

[6 Webographie & annexes 4](#_Toc60064007)

# Introduction

Ce projet est exécuté dans le cadre des modules 133, 151 et 105 visant à acquérir les compétences nécessaires à la création d’un petit site internet dynamique récupérant ses données dans une base de données créée pour l’occasion.

Ici, le but est de reproduire un site indexant des recettes de plats. Différentes fonctionnalités sont demandées, comme l’entrée des recettes dans la base de données directement depuis le site, la connexion au site avec ou sans certains droits administrateurs ou encore la possibilité de contacter les administrateurs du site directement par un formulaire de contact.

Le site doit suivre une structure en MVC et les éléments de style sont donnés par un Framework non-imposé (ici, Bootstrap).

La méthode de travail n’est pas imposée non plus et peut être modifiée au cours de la conception du site si nécessaire.

Le projet est rendu le 4 janvier 2021 au plus tard.

# Analyse

## Pages du site

Les pages de ce site suivent une structure MVC. Les pages sont donc divisées en 3 catégories. Les Modèles, les Contrôleurs et les Vues. Chacune de ces catégories de pages ont un rôle bien défini au sein du fonctionnement du site.

1. Les Modèles vont servir à aller récupérer les données à la base de données liée au site. C’est ici que sont faites les requêtes SQL servant à récupérer ou à insérer des données dans une base. C’est également ici que sont gardées d’autres informations comme celles servant à se lier à la bonne base de données avec le bon port, etc. Ces pages sont des pages PHP et ne contiennent normalement pas de code HTML car elles ne sont pas sensées afficher les données qu’elles récupèrent.
2. Les Contrôleurs, eux, servent à faire transiter les données récupérées par le Modèle entre ce dernier et les pages de la Vue. C’est également ici que les données vont être mises en forme si c’est nécessaire, que ce soit pour les afficher à l’écran ou pour les insérer dans la base de données. Ces pages n’affichent pas de contenu à l’écran et ne contiennent donc pas de code HTML mais uniquement du PHP. Le contrôleur principal est appelé Master Controller et est une classe. La plupart des autres pages sont des extensions de cette dernière.
3. Les vues, quant à elles, servent à afficher à l’écran tout ce qui doit être affiché à l’écran de l’utilisateur. Ces pages sont quasi totalement composées de code HTML mais peuvent également contenir du PHP. C’est ici que sont gardées les pages de code du Head, Header ou encore Footer, par exemple.

Le site comporte une page d’accueil qui affiche les 4 dernières recettes ajoutées au site, une page contenant une liste de toutes les recettes disponibles sur le site, une page de contact permettant l’envoi d’un mail, puis une page pour se créer un compte utilisateur. Ceci concerne toutes les personnes arrivant sur le site pour la première fois. Une fois un compte créé, la page création de compte disparaît et est remplacée par un bouton de déconnexion. Si la personne connectée est administrateur, une page supplémentaire d’ajout de recette s’ajoute sur la barre de navigation. Le site dispose aussi d’une page qui affiche les erreurs s’il y en a lors de la connexion, de la création de compte ou lors de l’ajout de recettes. Cependant ces pages ne sont pas accessibles sans avoir passé par la création de compte, la connexion ou l’ajout de recette (admin).

La structure des pages reste la même pour toutes les pages. Nous avons un header qui ne bouge pas, un menu qui reste tout le temps le même (nous gérons ce qui est affiché avec des conditions php), un contenu qui va varier selon les pages, puis pour finir un footer, qui lui aussi est tout le temps le même. Comme expliquer plus en haut, le site est structuré en MVC, du coup les éléments sont séparés. Afin de gérer ce qui doit être affiché et quel contrôleur doit être appelé, nous passons toujours par la page index qui, elle, sert de chef d’orchestre et qui va renvoyer sur le bon contrôleur, qui lui va faire appel au bon model afin d’obtenir les bonnes informations de la base de données, puis ce même contrôleur va renvoyer quel est le contenu qui doit être affiché.

## Base de données

Pour la base de données, nous avons dû nous y prendre à plusieurs fois. Effectivement, nous n’avions peut-être pas une vision exacte de l’entièreté du projet dès le début. Nous avons assez vite créé notre première base de données pour pouvoir commencer à intégrer certains éléments du site de manière dynamique, mais il nous a par la suite manqué certaines choses ou d’autres pouvaient être modifiées pour faciliter ou clarifier le projet. Nous avions, par exemple, au début, créé une table pour les utilisateurs et une autre pour les administrateurs. Quand il a été temps de mettre en place les connexions, il nous a tout de suite semblé beaucoup plus logique de n’avoir qu’une table pour tous les utilisateurs, à laquelle on a ajouté une colonne qui définit les droits de l’utilisateur en question.

En dehors de cela, la création de notre base de données est assez simple. Pour commencer, nous avons essayé de réfléchir à tout ce dont nous aurions besoin afin d’oublier le moins d’éléments possible. Tous ces éléments ont été notés sur une feuille de brouillon servant de schéma à notre base de données. Nous avons ensuite utilisé le logiciel *dbMain* pour mettre en place un MCD et un MLD ainsi que pour gérer les types des différentes données qui y seraient entrées.

Les données de la base de données sont ensuite entrées dans le fichier config du site afin de faciliter une modification éventuelle dans les identifiants, par exemple. Le fichier database se charge ensuite d’utiliser les données de config pour se connecter à la base de données.

A partir de là, il est possible de récupérer et d’entrer des données directement depuis le site via des fonctions contenant des requêtes SQL.

## Structure

La structure du site est relativement simple. Nous avons une page index qui va afficher dans un premier temps la page principale du site. Dans cette même page nous incluons tous les contrôleurs du site, puis nous mettons une fonction contenant un switch permettant de récupérer le bon contrôleur, puis pour finir nous avons une autre fonction qui dispose de la structure de chaque page. Nous avons un header qui ne bouge pas, un menu qui ne change pas non plus, une partie « contenu » qui va changer pour chaque page du site, puis pour finir le footer qui reste lui aussi inchangé.

Depuis le menu, lorsque nous voulons aller sur la page recettes, en appuyant sur « Recettes », on sera envoyé sur index, qui va récupérer dans le switch le bon contrôleur, qui lui va se charger de prendre les données nécessaires dans le model nécessaire, puis va afficher le contenu désiré. Ce mode de fonctionnement est le même pour tout le site.

Pour ce qui est de la structure des dossiers dans notre projet, nous avons une page index qui se trouve à la racine, puis tout le reste se situe dans des dossiers. Nous avons un dossier « Controller » pour tous les contrôleurs du site, un dossier « Images » pour les images, un dossier « Model » pour tous les modèles, un dossier « Ressources » qui lui contient un dossier « css » qui va contenir le bootstrap, ainsi qu’une page de style personnalisée. Nous avons ensuite le dossier « View » qui contient le plus de fichiers. Tout d’abord nous trouvons trois autres dossiers nommés « home », « recipes » et « user ». Dans le premier nous avons tout ce qui se réfère aux pages se trouvant dans le menu à l’arrivée sur le site. Dans « recipes » tout ce qui concerne les recettes, puis dans « user » ce qui gère de loin ou de près la création ou connexion des utilisateurs.

Pour finir nous avons les pages qui restent inchangées dans la racine du dossier « View ».

# Réalisation

## Gestion des images

Nous avons un dossier où nous mettons toutes les images nécessaires au site. Nous avons redimensionné les images, puis nous les avons compressés afin qu’elles prennent moins de place. Ensuite nous avons principalement que des images des recettes, ce qui va être géré par la base de données. Nous introduisons le nom de l’image lors de la création de la recette, puis celui-ci sera ajouté automatiquement dans le lien qui va chercher l’image dans le dossier. L’idée de créer des sprites d’images nous est venue, mais cela aurait été compliqué à notre avis de gérer avec la base de données, donc nous avons abandonnée l’idée, surtout que des images sont constamment ajoutée lorsque nous créons une recette.

## Eco-conception

Afin que notre site soit le plus éco-responsable possible, nous avons pris le réflexe de redimensionner les images avant de les ajouter au site, puis nous les avons compressées afin de réduire leur poids. Nous avons aussi fait attention à ne pas laisser du code inutile dans le site. Pour ce qui est des fichiers de style, nous avons pris l’initiative de les compresser eux aussi une fois que nous n’en avions plus besoin. Cela permet de réduire considérablement les lignes de codes et donc d’avoir des fichiers moins volumineux. Bien sûr nous avons fait pareil pour nos pages html.

Nous sommes conscients que le site n’est pas parfait en ce sens-là, mais nous avons essayé de faire tout notre possible pour qu’il soit le plus écologique possible.

## Ajout d’une recette

L’ajout d’une recette se fait via un formulaire simple auquel on peut accéder via le menu en haut de chacune des autres pages du site, mais seulement si une session possédant les droits administrateurs est en cours. Nous pouvons ensuite entrer les données de la recette dans les différents champs du formulaire.

Ces dernières seront ensuite envoyées à la page recipeController, qui se chargera d’organiser les données avant la prochaine étape.

On envoie ensuite ces données dans le bon Modèle et dans la bonne fonction qui va, quant à elle, s’occuper d’entrer correctement toutes ces données dans la base de données.

# Tests

Au départ nous n’avons pas intégré du PHP à notre site, du coup les tests consistaient seulement à avoir un affichage correct des pages et des éléments. Lorsque nous avons commencé à ajouter du PHP et à afficher des éléments de manière dynamique, les tests c’était de voir que tout s’affichait correctement à l’endroit voulu et qu’aucune erreur n’apparaissait.

Le premier élément ajouté au site ce fut un affichage dynamique de toutes les recettes. Nous avons donc ajouté pleins d’éléments dans la base de données pour voir si tout ce qui était ajouté s’affichait correctement dans la page contenant toutes les recettes. Ensuite nous avons ajouté la possibilité d’avoir le détail des recettes. Pour tester cette fonctionnalité, nous avons ajouté un bouton sous chaque recette pour accéder à cela, puis nous avons essayé cette fonctionnalité avec les éléments ajoutés précédemment.

La plus grande partie des tests ce sont fait lorsque nous avons ajouté la possibilité de créer un compte et la possibilité de se connecter au site. Tout d’abord nous avons vérifié que les données étaient bien envoyées dans la base de données, puis une fois ceci fait, nous avons essayer de vérifier que rien n’était envoyé s’il y avait des erreurs. Une fois tout ceci correct, nous avons testé de nous connecter en récupérant les données des utilisateurs et les comparer avec ce qui était mis par la personne essayant de se connecter. Une fois tout ceci en place on a pu procéder au test final.

On a vidé le cache et supprimer les cookies de connexion même si nous ne possédons pas sur notre site. L’objectif était de partir le plus clean possible sur le site, comme si c’était la première fois. Une fois sur le site on vérifiait que l’onglet création de compte apparaissait, mais pas celui de déconnexion, puis que celui de création de recette n’apparaissait pas non plus. Ensuite on a créé un compte utilisateur et nous nous sommes connectés. A ce stade nous avons seulement pu voir si la connexion était toujours possible et que l’onglet déconnexion s’affichait, que la création de compte n’était plus possible et que l’ajout de recette restait toujours invisible puisque nous n’étions pas admin. Pour finir nous nous sommes déconnectés du site, puis avons changé le statut de l’utilisateur en administrateur sur PhpMyAdmin. Nous nous sommes de nouveau connectés, ce qui a permis de voir s’il n’y avait aucun problème avec les sessions et les variables de sessions. Une fois connectés tout devait être comme avant, mais cette fois nous devions pouvoir ajouter une recette. Tout était en ordre donc nous avons jugé que le site était opérationnel.

# Conclusion

## Conclusion générale

Pour conclure, ce projet a été un de ceux qui nous aura procuré le plus de problèmes, mais également un de ceux qui nous aura appris le plus de choses et qui nous sera surement le plus important dans notre carrière. Le manque de connaissances et de temps auront été nos principaux soucis. Manque de connaissances surtout car beaucoup de facettes de ce projet nous étaient encore inconnues au début de celui-ci. Comment faire une bonne base de données qui soit utilisable mais aussi efficace ? Comment intégrer cette base de données à notre site web ? Comment y insérer des données depuis le site ? Comment structurer le projet en MVC ? Comment utiliser efficacement GitKraken afin de faciliter la mise en commun de notre travail ? Nous n’aurions certainement pas su répondre à toutes ces questions avant ce projet. Malgré les difficultés rencontrées, dans le groupe ou individuellement, nous avons aujourd’hui un site fonctionnel et qui nous plait. Ce projet a également servi à mettre en avant les lacunes de chacun de nous deux et nous permettra de connaitre nos points forts et faibles, et de consolider les sujets dans lesquels nous ne sommes pas à l’aise. Nous aurons également appris à créer un site dynamique utilisant une base de données et ce savoir pourra être utilisé lors de projets personnels ou professionnels dans nos futurs respectifs. Ces connaissances sont également des connaissances au niveau organisationnel de notre travail (Gestion du temps, utilisation d’outils comme GitKraken, etc.).

## Conclusions Personnelles

### Vitor Silva

Vitor…

### Ricardo Delgado Miranda

Dès le début, ce projet m’a beaucoup intéressé et motivé. J’ai toujours été plus intéressé par la programmation web que par la programmation c#, par exemple, et dans ce projet j’ai eu l’impression de construire un site qui pourrait réellement être utilisé. J’ai appris énormément de choses qui me seront surement très utiles dans ma vie. J’ai été un peu perdu au moment de passer le site en MVC mais Vitor qui avait bien mieux compris que moi à pu m’aider à comprendre un peu mieux cette structure. J’ai en tout cas appris énormément de choses au long de ce projet et j’ai hâte de pouvoir intégrer tout cela dans des projets personnels ou professionnels.

## Méthode de projet

Notre méthode de travail n’a pas tout de suite été la meilleure. Pour ce qui est de la répartition du travail, nous avons tout de suite réussi à faire les choses assez bien. Le travail était correctement réparti et le projet avançait bien. Le problème est surtout dans le fait que nous avons utilisé initialement Google Drive pour faire les échanges de fichiers et ça rendait les mises en commun assez compliquées à certains moments. Plus tard, nous avons appris à utiliser GitKraken et l’avons incorporé au développement du site et ça a beaucoup facilité les choses. Cependant, notre faible maitrise de l’application nous a posé quelques problèmes au moment des *push* sur le *main*. Nous avons dû gérer quelques conflits dans certains fichiers qui nous auront fait perdre un peu de temps mais ce n’est rien par rapport au gain de temps que nous a apporté GitKraken finalement. Nous avons également pas mal manqué de temps pour ce projet. En effet une partie des éléments n’étaient pas finis avant les vacances et nous avons dû nous organiser pour travailler à distance et finir le projet dans les délais impartis. Pour ce qui est de la communication au sein du groupe, tout s’est toujours plutôt bien passé. Nous avons déjà travaillé ensemble dans pas mal de projets ce qui fait qu’on connait la méthode de travail de l’autre, ce qui fluidifie les échanges et le travail.

# Webographie & annexes

w3schools: <https://www.w3schools.com/>

Regex101: <https://regex101.com/>

Bootstrap: <https://getbootstrap.com/>

Font Awesome: <https://fontawesome.com/>

Section INF : <https://enseignement.section-inf.ch/>