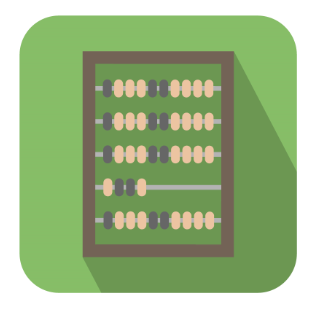


Rapport de projet

1ère année

**Profan**

**Application de gestion de stocks**



Présenté par Flavio RANCHON et Jaël VAVASSEUR

Responsale ISIMA : P.Laurençot

Campus des Cézeaux - 1 rue de la Chébarde - TSA 60125 – 63178 Aubière CEDEX

# Remerciements :

Nous tenons à remercier nos professeurs encadrants, Patrice Laurençot et Loïc Yon qui nous ont beaucoup aidé, notamment sur la mise en ligne de notre application sur le serveur. Nous voulons aussi remercier Muriel Mouzat pour son aide sur la rédaction du cahier des charges et du rapport de projet, ainsi que ses conseils tout au long du projet.

Table des matières

[Remerciements : 2](#_Toc9558531)

[Introduction : 4](#_Toc9558532)

[Problème - Contexte 5](#_Toc9558533)

[Répartition du travail – Méthodes de travail 5](#_Toc9558534)

[Solution – Concept de l’application 6](#_Toc9558535)

[La fonction scanner : 8](#_Toc9558536)

[Choix des langages/technologies : 8](#_Toc9558537)

[Résultats, présentation du site web 9](#_Toc9558538)

[Description de l’application web 9](#_Toc9558539)

[La page d’accueil 9](#_Toc9558540)

[Stock 9](#_Toc9558541)

[Gestion des utilisateurs 10](#_Toc9558542)

[Page tutoriel 10](#_Toc9558543)

[Améliorations – Fonctionnalités non implémentées 11](#_Toc9558544)

[Conclusion 12](#_Toc9558545)

[Annexe 1 : Diagramme de Gantt prévisionnel 13](#_Toc9558546)

[Annexe 2 : Diagramme de Gantt réel 14](#_Toc9558547)

[Annexe 3 : Captures d’écran du site 15](#_Toc9558548)

# Introduction :

Le pôle Plurimédia du lycée Lafayette possède une imprimerie dont la gestion est assurée par des étudiants du lycée accompagnés par leurs professeurs encadrants. Cette imprimerie est une véritable petite entreprise, qui réalise régulièrement des commandes pour des professionnels. Cependant, elle ne possède aucun moyen de gérer ses stocks en temps réel ou d’avoir une idée du contenu de ses salles de stockages de matériel.

Notre projet se déroule donc en collaboration étroite avec le lycée Lafayette et plus particulièrement avec le pôle Plurimédia de ce dernier. Il consiste en la création d’un outil de visualisation et de gestion des stocks de l’imprimerie, qui permettra à l’imprimerie de connaitre en temps réel ce qu’elle possède en stock et leur permettra de répondre à des commandes sans avoir à se déplacer dans les salles de stockage.

Notre solution doit satisfaire plusieurs contraintes, notamment l’accessibilité depuis n’importe où à partir de n’importe quel appareil, c’est pourquoi la solution développée est une application web, qui doit être ergonomique, intuitive et facile d’utilisation.



# Problème - Contexte

L’imprimerie du pôle Plurimédia du lycée Lafayette est une petite entreprise qui a besoin de connaitre l’état de ses stocks pour améliorer sa productivité et son efficacité.

L’imprimerie possède plusieurs salles de stockage, où sont stockés le papier, l’encre ou encore les produits nécessaires à la maintenance des machines :

* Une salle où sont stockés toutes sortes de papiers, de tous formats
* Une autre salle où sont stockés l’encre et les produits d’entretien
* Et enfin, du papier est aussi stocké dans une dernière salle : l’atelier

## Répartition du travail – Méthodes de travail

Notre travail est réparti entre les deux membres de notre groupe, le partage a été effectué au début et ajusté tout au long du projet selon l’avancement et nos besoins. Flavio a plutôt travaillé sur le back-end de l’application, c’est-à-dire tout ce qu’il se passe derrière et que l’utilisateur ne voit pas. Tandis que Jaël a travaillé plutôt sur le front-end, donc tout ce que l’utilisateur voit affiché à l’écran.

Nous avons établi un diagramme de Gantt prévisionnel *(cf. Annexe 1)* au début du projet en même temps que le cahier des charges, qui a été respecté en grande partie. Cependant, certains points et certaines fonctions de l’application ont évolué ou ont été ajoutées au cours du projet. Notre méthode de travail agile nous a permis une adaptation rapide et efficace à ces changements, et nous a permis d’achever notre application en temps et en heure avec un calendrier légèrement différent de celui prévu dans le Gantt prévisionnel *(cf. Annexe 2)*.

# Solution – Concept de l’application

Le but de notre projet est donc d’élaborer un outil permettant à l’imprimerie de pouvoir avoir accès à une visualisation du contenu de leur stock n’importe quand et n’importe où, ainsi que de gérer ces stocks, c’est-à-dire enlever une certaine quantité d’un produit que l’on va utiliser, ou ajouter le papier qui a été acheté préalablement.

## Etapes préliminaires

Avant de commencer à coder, il a fallu établir plusieurs choses préalables, une structure de données efficace, compréhensible et simple, et un diagramme des cas d’utilisations qui spécifie selon chaque type d’utilisateur, les actions qu’il aura le droit d’exécuter sur l’application. Ici, nous avons trois types d’utilisateurs :

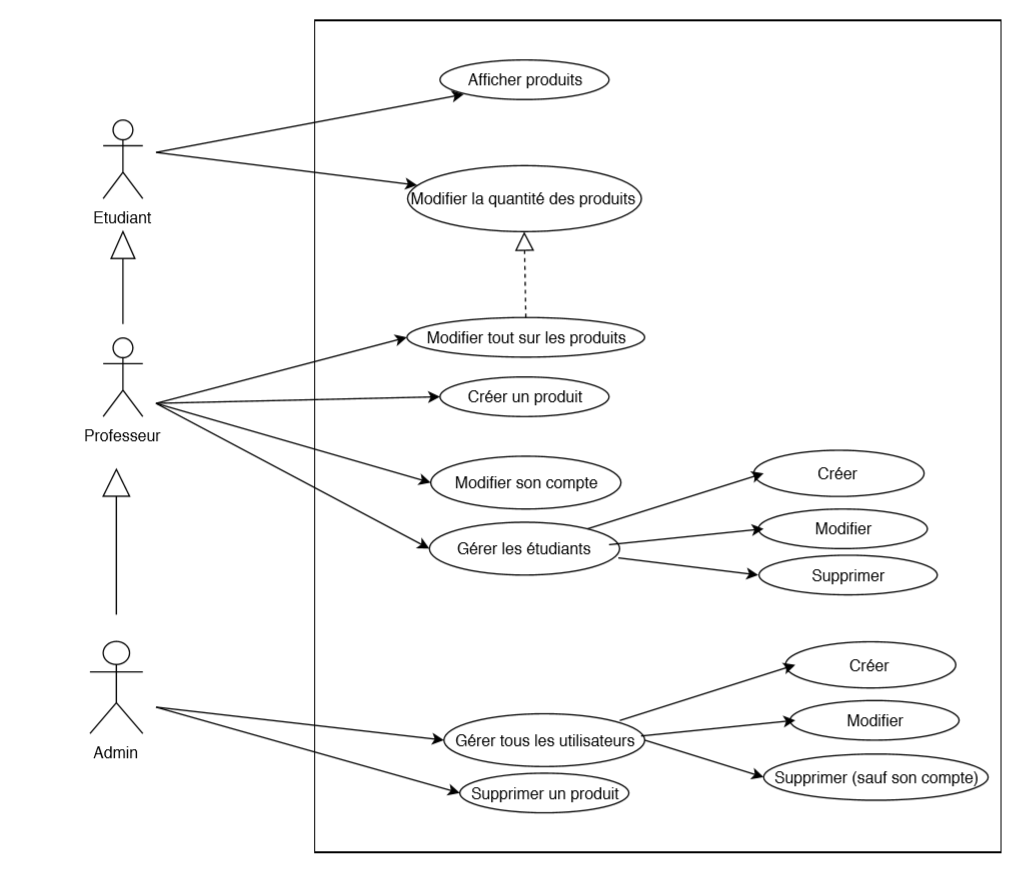
* Administrateur : il aura tous les droits
* Professeur : il aura le droit d’ajouter des nouveaux produits, de créer des comptes étudiants
* Etudiant : il pourra seulement effectuer des recherches et modifier les quantités des produits des stocks

Figure 1 Diagramme des cas d'utilisations

Afin d’organiser ces stocks, nous avons eu besoin d’organiser les différents produits qui composent les stocks. Nous nous sommes alors dirigés vers une base de données différenciant les différents types de produits, c’est-à-dire que le papier, et plus globalement tous les supports d’impression sont classés ensemble, l’encre possède aussi sa propre table. Enfin tous les produits d’entretien ou tout produit ne correspondant pas aux deux autres classes sont classés dans une troisième et dernière table. Voici un schéma de la première version de notre structure de données :

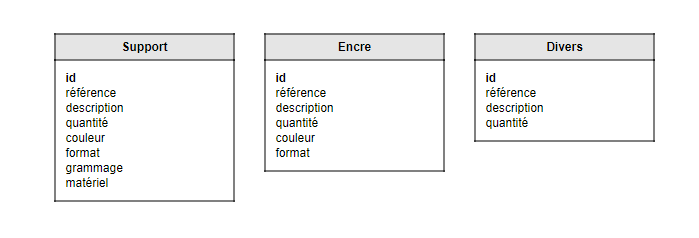


Figure 2 : Schéma base de données Stock

On voit dans cette configuration que les catégories *Support* et *Encre* ont beaucoup de champs. Cette organisation permet une grande précision pour les supports et pour les encres, mais complexifie par la même la recherche et rend fastidieuse l’entrée de nouvelles données due au grand nombre de champs à remplir. De plus, cette organisation ne permet pas une recherche efficace dans la base de données car lors d’une recherche, il faudrait afficher les éléments de ces trois tables dans des tableaux différents du fait de la différence de nombre de champs.

Nous avons donc décidé de changer notre structure de données afin de la rendre plus simple à comprendre, et à gérer. Nous avons donc limité le nombre de champs et harmonisé les classes pour créer une seule classe *Produit* qui contient tous les produits quels qu’ils soient.

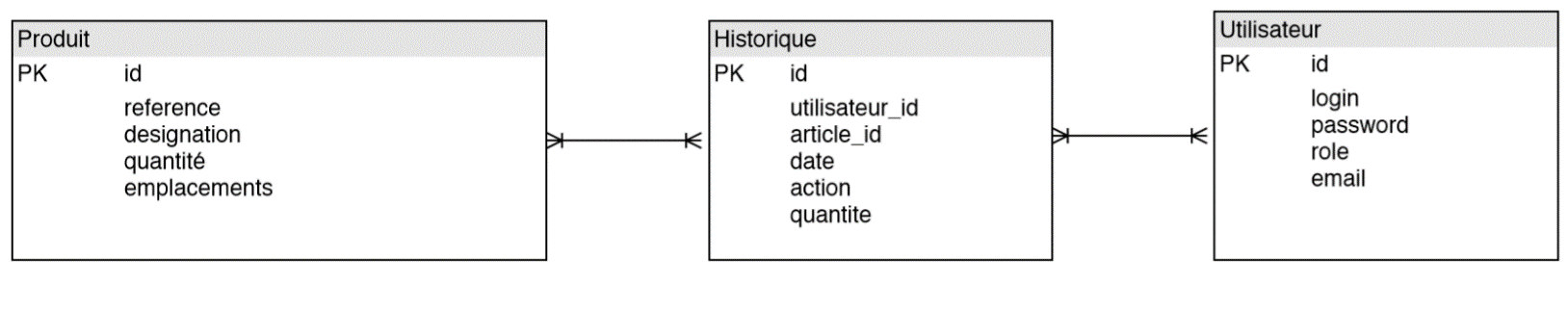


Figure 3 : Schéma base de données finale

Nous avons aussi ajouté une table Utilisateur qui contiendra toutes les informations de connexion des utilisateurs, ainsi qu’une table Historique qui va garder en mémoire chaque changement dans la table des produits. Cette organisation des produits plus pratique permet de regrouper dans la même table tous les produits et donc de rechercher un produit dans la liste globale des produits et plus non seulement dans une des trois tables composant les stocks. Ainsi notre structure de données est très peu complexe, facile à comprendre et donc facile à gérer.

## La fonction scanner :

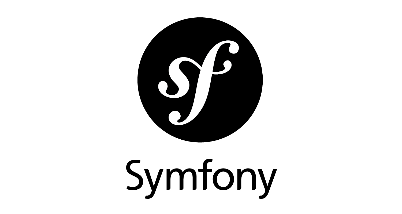
Afin de faciliter la recherche dans la base de données et l’ajout de nouveaux produits, une fonction de scan a été développée, permettant à l’aide de la caméra d’un appareil mobile (smartphone ou tablette) de récupérer le code-barres, qui sert de clé primaire à notre table des produits.

Lors d’un scan, si le code-barres (et donc le produit) est déjà présent dans la base de données, le site redirigera l’utilisateur vers la page produit de l’élément scanné, ce qui donnera des informations sur son emplacement, sa quantité… Si le code-barres n’existe pas dans la base de données, l’application va créer un nouveau produit et va inviter l’administrateur à remplir les champs requis pour créer ce nouveau produit. Si l’utilisateur n’est pas un administrateur, renvoie une erreur.

## Choix des langages/technologies :

Notre solution est une application web, ce qui permet d’y accéder depuis n’importe où depuis n’importe quel appareil. Nous avons donc choisi de développer notre application sous PHP avec Symfony, un Framework très utilisé et très puissant qui a pour avantage d’avoir une grande communauté active, qui peut être d’une grande aide pour résoudre des bugs que d’autres ont déjà connu.

Nous utiliserons en complément de ce Framework PHP, un autre Framework, Bootstrap, qui est codé en html, CSS et javascript, pour gérer le front-end cette fois. Bootstrap est un des Frameworks web les plus utilisés, premièrement car il est libre (comme Symfony) et très facile à gérer. Il permet par exemple une gestion du responsive de notre site, c’est-à-dire une adaptation du site à la taille de l’écran sur lequel il est affiché pour rester toujours lisible.



Pour développer, nous avons choisi d’utiliser **PHPstorm**, un environnement de développement intégré très complet et pratique, il est payant mais une licence d’un an est offerte aux étudiants en passant par l’UCA. Nous avons utilisé **GitHub** en gestionnaire de version et **Wampserver** pour virtualiser un serveur local et pouvoir tester notre code PHP. Enfin, nous avons utilisé divers sites comme *moqups.com* pour réaliser nos maquettes de site et *draw.io* pour certains diagrammes.

# Résultats, présentation du site web

## Description de l’application web

Notre application de gestion de stocks est composée de plusieurs pages, décrites ci-dessous (et dont des captures sont imprimées en annexe) :

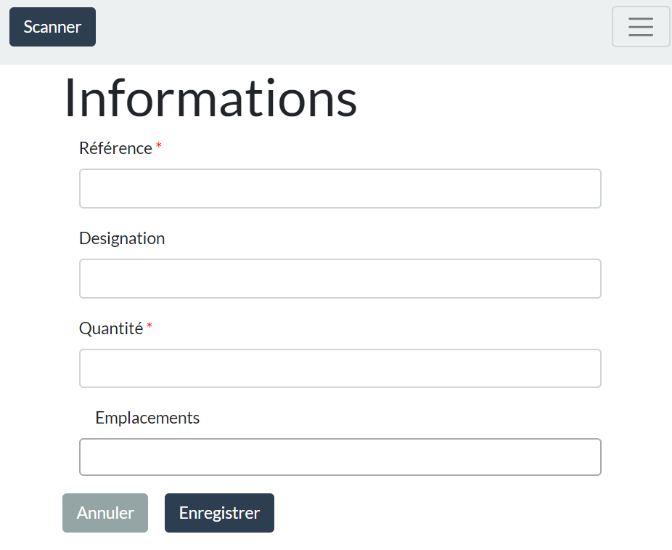
### La page d’accueil

C’est la page principale de notre application, elle est composée tout d’abord d’un historique des transactions effectuées par les utilisateurs sur les produits du stock. Cet historique répertorie la date de la modification, l’utilisateur ayant effectué la modification, le produit impacté et le type d’action effectué. Ce tableau permet d’avoir une idée de ce qu’il s’est passé pendant la journée (ou la semaine). Tous les utilisateurs peuvent le voir, ce qui permet d’éviter ou au moins de remarquer les erreurs ou les mauvaises plaisanteries des étudiants qui pourraient rajouter des millions de produits sans le faire exprès (ou pas).

Nous avons ajouté en dessous un graphique qui répertorie et classe tous les produits du stock selon leur quantité.

### Stock

La page la plus intéressante du projet, elle permet une recherche par mot clé dans la base de données, et pour les utilisateurs ayant les droits, elle permet d’ajouter un nouveau produit, de modifier les produits existants ou de les supprimer.



Pour créer un produit (fonction accessible seulement pour les administrateurs) il suffit de cliquer sur le bouton **Créer un produit**, et un formulaire apparaît, avec les champs à remplir (voir ci-contre).

Le champ référence est en fait le code-barres du produit, il peut être rentré plus facilement en le scannant (directement depuis la fonction scan accessible depuis le bouton **Scanner).**

### Gestion des utilisateurs

Cette page permet aux administrateurs de gérer les comptes, c’est-à-dire créer un compte pour un professeur, modifier le mot de passe d’un étudiant l’ayant oublié ou encore supprimer le compte d’un autre administrateur. Les administrateurs ont droit de tout modifier sur tous les comptes utilisateurs. La seule chose qui leur est impossible est de supprimer leur propre compte (ce qui aurait pu entrainer des problèmes majeurs).

Les comptes ayant l’accès Professeur peuvent modifier les informations des comptes étudiants, ainsi que leurs propres informations.

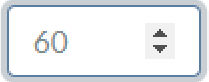
### Page tutoriel

Nous avons créé cette page pour permettre aux utilisateurs de comprendre toutes les fonctionnalités de l’application leur étant accessible. Le seul but de cette page est donc d’apprendre aux utilisateurs à se servir de l’application comme la fonction scan, la gestion des produits, le fonctionnement de la recherche, ou encore le fonctionnement de la gestion utilisateur.

## Améliorations – Fonctionnalités non implémentées

De nombreuses fonctionnalités auraient pu être ajoutées à notre application, afin qu’elle soit plus sécurisée, plus efficace et plus fiable, par exemple :

* Mieux gérer la création de compte utilisateur et seulement demander à l’utilisateur créant le nouveau compte de rentrer une adresse mail qui enverrait à l’utilisateur dont le compte est créé un lien avec un formulaire à remplir avec son nom et son mot de passe.
* Une gestion du multi-accès simultané des produits du stock (ainsi que des utilisateurs) c’est-à-dire bloquer l’accès à la modification d’un produit (respectivement la modification d’un utilisateur) dès qu’un utilisateur le modifie de manière à éviter les doubles modifications entrainant des erreurs.
* La modification de la quantité d’un produit directement dans la page stock sans entrer dans la page du produit, qui faciliterait la modification dans le stock mais pourrait entrainer beaucoup de problèmes de modification intempestives (notamment avec le tactile des tablettes). Cela pourrait être gérable si la modification n’est pas faite instantanément mais requiert une validation.



* Ajouter un « panier » de produits, par exemple pour une préparer une commande qui nécessite beaucoup de produits différents, cette fonction permettrait de réserver tous les produits dont on a besoin pour réaliser la commande.

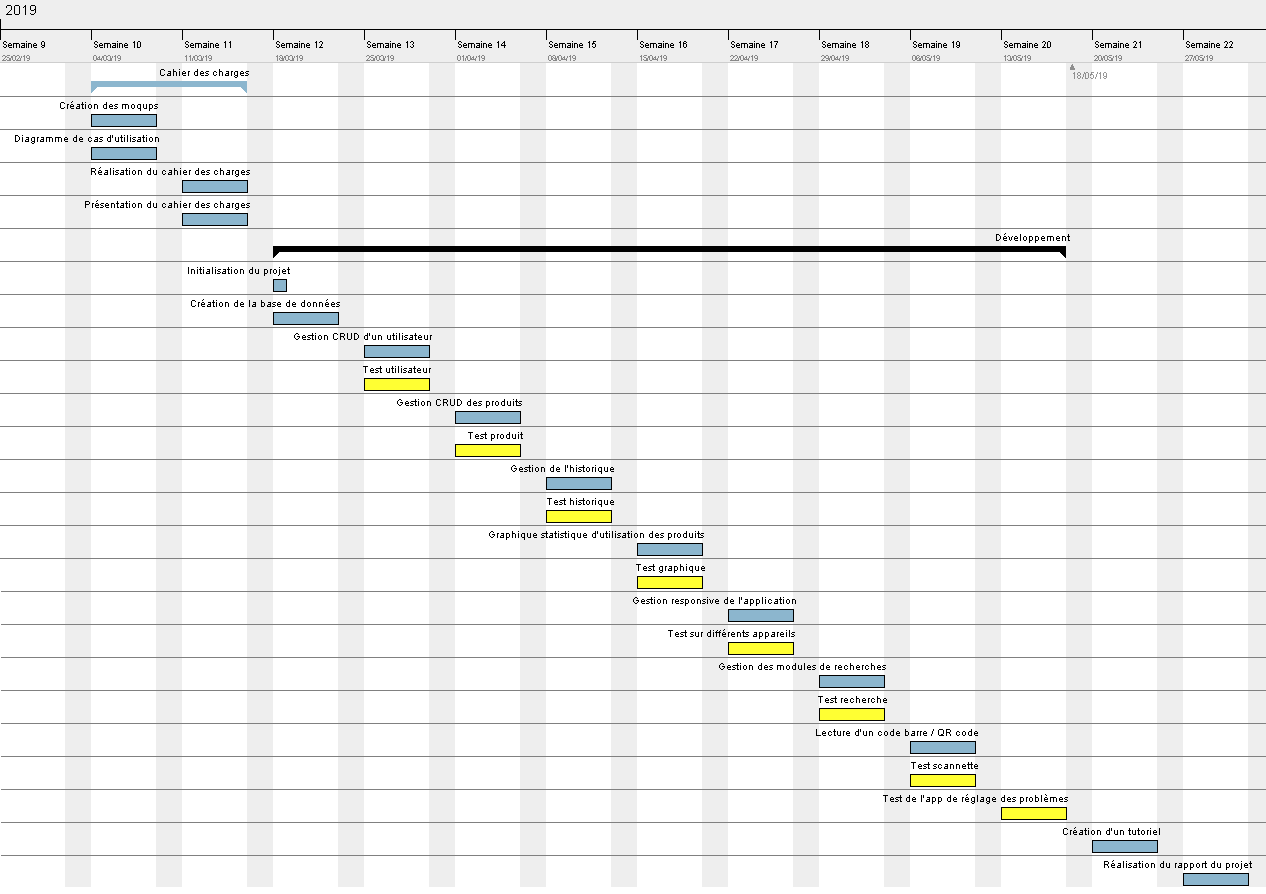
# Conclusion

Notre projet est terminé comme décrit ci-avant, mais il est loin d’être parfait et de nombreuses améliorations ou fonctionnalités supplémentaires auraient pu y être ajoutées. Il est cependant tout à fait fonctionnel et **Gestocks** remplit parfaitement sa mission de gestion de stocks, comme le cahier des charges le demandait. L’application web est déjà en ligne et n’attend qu’à être utilisée par les membres de l’entreprise Plurimédia du lycée La Fayette.

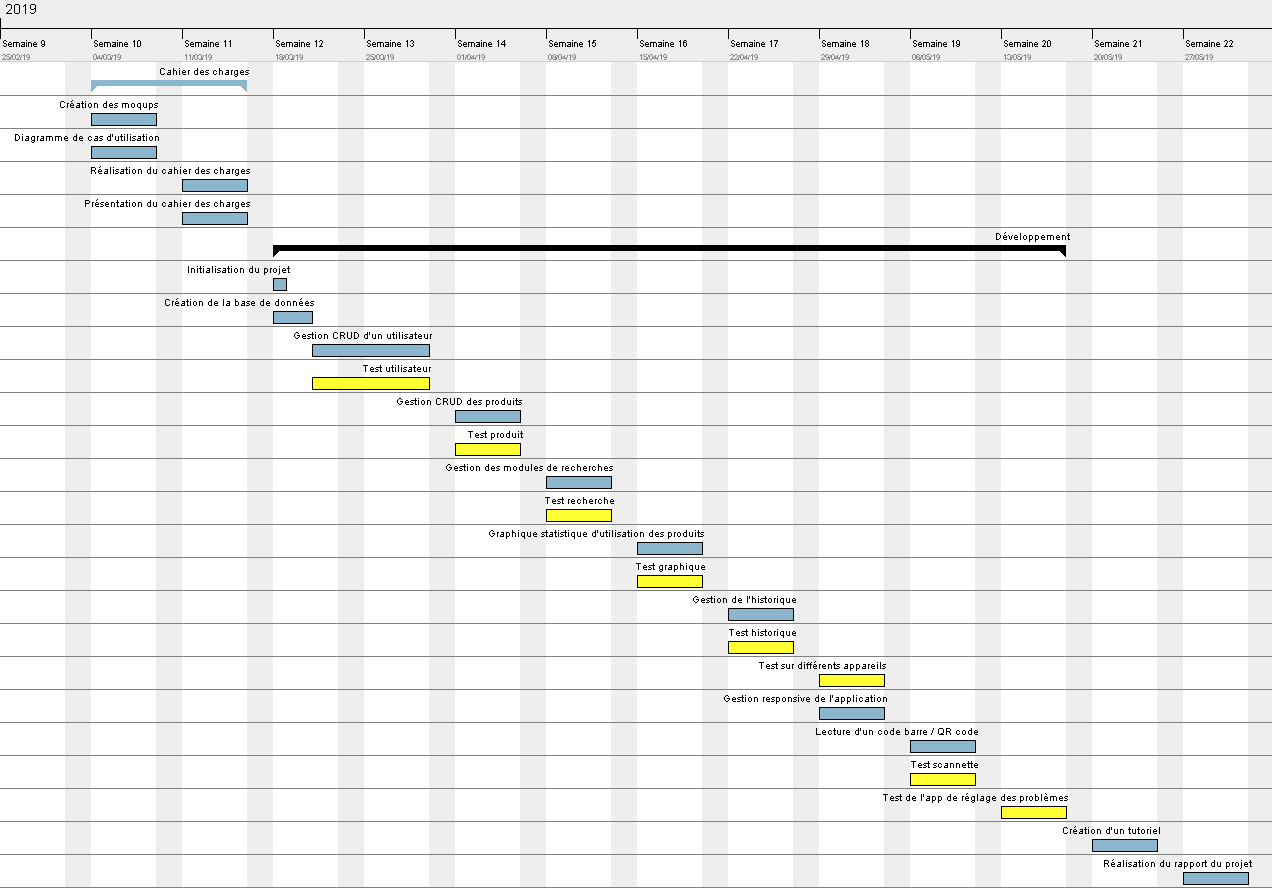
Ce projet nous a permis de nous enrichir personnellement dans plusieurs domaines, notamment en Php et son framework Symfony, un outil simple à utiliser et puissant qui permet un gain de temps et de productivité non négligeables pour la fabrication d’une application web nécessitant une base de données. Nous avons aussi appris à gérer au mieux le responsive avec Bootstrap et ses thèmes Bootswatch, qui eux aussi nous ont fait gagner un temps précieux dans la réalisation d’un front-end responsive et ergonomique.

Finalement, après avoir géré un projet, depuis la commande d’un client, en passant par la réalisation d’un cahier des charges, jusqu’à la livraison du produit, nous avons travaillé de manière agile, c’est-à-dire en s’adaptant au fil des semaines aux attentes changeantes ou originellement mal comprises des clients. Au cœur de notre collaboration en binôme pour la réalisation de ce projet, la communication interne a été un facteur important qui a contribué à l’aboutissement dans les délais du projet.

# Annexe 1 : Diagramme de Gantt prévisionnel



# Annexe 2 : Diagramme de Gantt réel



# Annexe 3 : Captures d’écran du site

