Rapport mini-projet NFA021

Formation

« Retour aux sources »

Session Sept-Déc. 2015

Auteurs : Bruno FROGER et Hélène LE QUEAU

Table des matières

[1. Présentation du mini-projet 3](#_Toc441250442)

[2. Déroulement du mini-projet 4](#_Toc441250443)

[3. Analyse du cahier des charges 5](#_Toc441250444)

[4. Choix d’implémentation 6](#_Toc441250445)

[5. Modélisation 7](#_Toc441250446)

[a. Modélisation de la base SQL avec MySqlWorkbench 7](#_Toc441250447)

[b. Enchainement des pages web 9](#_Toc441250448)

[6. Architecture matérielle et logicielle 9](#_Toc441250449)

[a. Architecture matérielle (infrastructure) 9](#_Toc441250450)

[b. Organisation MVC 9](#_Toc441250451)

[c. Définition des classes et des fonctions de base 10](#_Toc441250452)

[d. Définition des règles d’affichage des pages 10](#_Toc441250453)

[7. Développement 10](#_Toc441250454)

[a. Présentation de la structure logicielle 10](#_Toc441250455)

[b. Construction du modèle de page 11](#_Toc441250456)

[c. Résumé des fonctionnalités du site 13](#_Toc441250457)

[8. Bilan du mini-projet 14](#_Toc441250458)

[a. Problèmes rencontrés 14](#_Toc441250459)

[b. Reste à faire 15](#_Toc441250460)

[9. Conclusion 16](#_Toc441250461)

[10. Remerciements 17](#_Toc441250462)

# Présentation du mini-projet

Ce mini-projet est l’aboutissement d’une formation proposée par SOFT à ORANGE afin de reprendre contact avec le développement logiciel : un « retour aux sources ».

Cette formation est dispensée par Nicolas Baupin, formateur CNAM, et est animée par François Marie, Scrum Master. Camille Oudot est le tuteur référent nommé pour les besoins d’« assistance locale ».

Cette formation fait partie d’un cursus « CNAM » qui intègre 3 Unités d’Enseignement (UE) :

* NFA016 : Création d’une page WEB coté client (HTML, CSS, JavaScript)
* NFA017 : Création de pages WEB générées par un serveur en PHP, avec utilisation de bases de données MySql. Ce module comprend également une ébauche du modèle MVC.
* NFA021 : mini-projet permettant de valider les acquis des deux précédents modules en développant un site WEB complet :
  + Développement des pages HTML
  + Définition des CSS de présentation
  + Utilisation du modèle MVC
  + Modélisation des données
  + Définition de la base de données
  + Définitions des vues
  + Définition des classes PHP
  + Écriture du code PHP des contrôleurs et des modèles
  + Tests

# Déroulement du mini-projet

Le sujet choisi pour le module NFA021 est celui proposé par défaut soit : « le site de recette cuisine ».

Le cahier des charges fourni regroupe les principales consignes (non exhaustives) destinées à servir de base au développement du site.

L’étude et la réalisation du projet s’est déroulée en plusieurs étapes :

1. Analyse du cahier des charges
2. Choix d’implémentation
3. Modélisation

* Des données
* Enchaînement des pages web

1. Architecture matérielle (infrastructure) et logicielle

* Mise en place de la plate-forme commune
* Organisation MVC
* Définition des classes et des fonctions de base
* Définition des règles d’affichage des pages (largeur fixe, définition des blocs et leurs imbrications,…)

1. Développement

* Création de la base de données et des tables
* Provisionner la BDD
* Développement des classes induites de l’implémentation de la BDD
* Construction du modèle de page : en-tête, menu, contenu, commentaire, pied de page.
* Écriture des fonctions générant l’affichage et la gestion des actions et des données saisies.
* Écriture des feuilles de styles.

1. Validation

# Analyse du cahier des charges

Les points importants retenus, du cahier des charges, sont les suivants :

* + Contraintes de développements : HTML, PHP, CSS, BDD MySQL

Ces choix dans le développement du site web répondent à la mise en application des connaissances apprises dans les modules NFA016 et NFA017 de la formation.

* + 3 types d’utilisateurs : Invité, user, admin
* *En italique gris : les fonctions non précisées explicitement, dans le cahier charges mais jugées intéressantes à insérer dans un site web de recettes de cuisine.*
* **Invité** : accès « Tout Public » au site. Consultation uniquement, mais toutefois avec la *possibilité de voter* pour une recette et de se créer un compte.
* **User** : Le « user » doit avoir un compte, qu’il peut créer via le site. Il dispose des mêmes fonctionnalités que l’invité, mais peut également :
  + - Créer/modifier ses recettes,
    - *Ajouter ses commentaires personnels à toute recette,*
    - *Taguer ses recettes préférées*,
    - Gérer les paramètres de son compte
* **Admin** : c’est un « user » avec des droits complémentaires :
  + - *Validation des recettes créées par les « users »*,
    - *Création / modification des listes de catégories / sous-catégories*,
    - Gestion des membres,
    - Gestion des recettes,

Un compte est défini par :

* + - Login (unique)
    - Mot de passe
    - Adresse mail (*unique*)
    - Nom
    - Prénom

Cette gestion de compte impose 3 niveaux de droits à gérer.

**Nota**: le cahier des charges propose la « suppression » d’éléments de la base de données. Ce point a été pris en compte, mais non implémenté « en l’état » dans la solution proposée. Du fait, du choix de l’utilisation de relations entre tables, qui impose une manipulation importante de données lors de la suppression, c’est l’option des données « masquées » qui a été privilégiée ; c’est-à-dire, lors de la suppression d’un membre, ses recettes ne seront pas supprimés mais réaffectées à un membre fictif.

* + Une recette est définie, au minimum, par :
    - Titre
    - Ingrédients
    - Nombre de personnes
    - Catégorie de prix et catégorie de difficulté
    - Descriptif de la recette
    - Cuisson
    - Conseils présentation
    - Catégorie
    - Sous-Catégorie

Pour les recettes, le cahier des charges proposait un jeu de données minimum, qui a été enrichi en fonction des fonctionnalités ajoutées en cours de développement.

Comme préciser ci-dessus, le cahier des charges n’étant pas limitatif, des fonctionnalités non demandées ont ainsi pu être ajoutées.

* + commentaires personnalisés sur une recette
  + notation des recettes
  + mail,
  + …

# Choix d’implémentation

Le projet est réalisé en binôme par Hélène et Bruno. De ce fait, le travail sera partagé selon les affinités, goûts et compétences de chacun.

La découpe des activités s’organise comme suit :

|  |  |
| --- | --- |
| Hélène | Bruno |
| * Modélisation base SQL * Structure des pages HTML * Définition CSS * Codage des vues de la méthode MVC * Tests de validation | * Mise en place de l’infrastructure technique * Définition des classes en respectant le modèle SQL défini * Développement PHP des contrôleurs et modèles de la méthode MVC * Tests de validation |

Du fait, de ce fonctionnement collaboratif, il a été décidé de mettre en place un environnement dédié et partageable. (Décrit par la suite)

Dans les choix, de développement, pour la réalisation d’un site web en mode client/serveur, c’est le modèle MVC qui a été retenu. Il en découle :

* + Architecture logicielle MVC
  + Code HTML/CSS coté client
  + Code PHP/MySql coté serveur

# Modélisation

Après plusieurs essais manuscrits, c’est l’outil MySqlWorkbench, qui semble le plus adapté au besoin. L’outil est plutôt simple à prendre en main : il suffit de dessiner les structures de données et l’outil génère automatiquement les scripts SQL correspondant.

En parallèle, l’outil HeidiSQL, bien plus simple et souple d’utilisation que PHPmyadmin, a permis de :

* + Visualiser le résultat des scripts de création,
  + Initialiser la base avec quelques données de tests,
  + Exporter et importer les données et structures

Lors de l’import de données, la structure de la base impose de respecter un ordre, lors de la création et du provisionning des tables, du fait des relations entre ces tables.

Pour ces tables, l’ordre à respecter est le suivant :

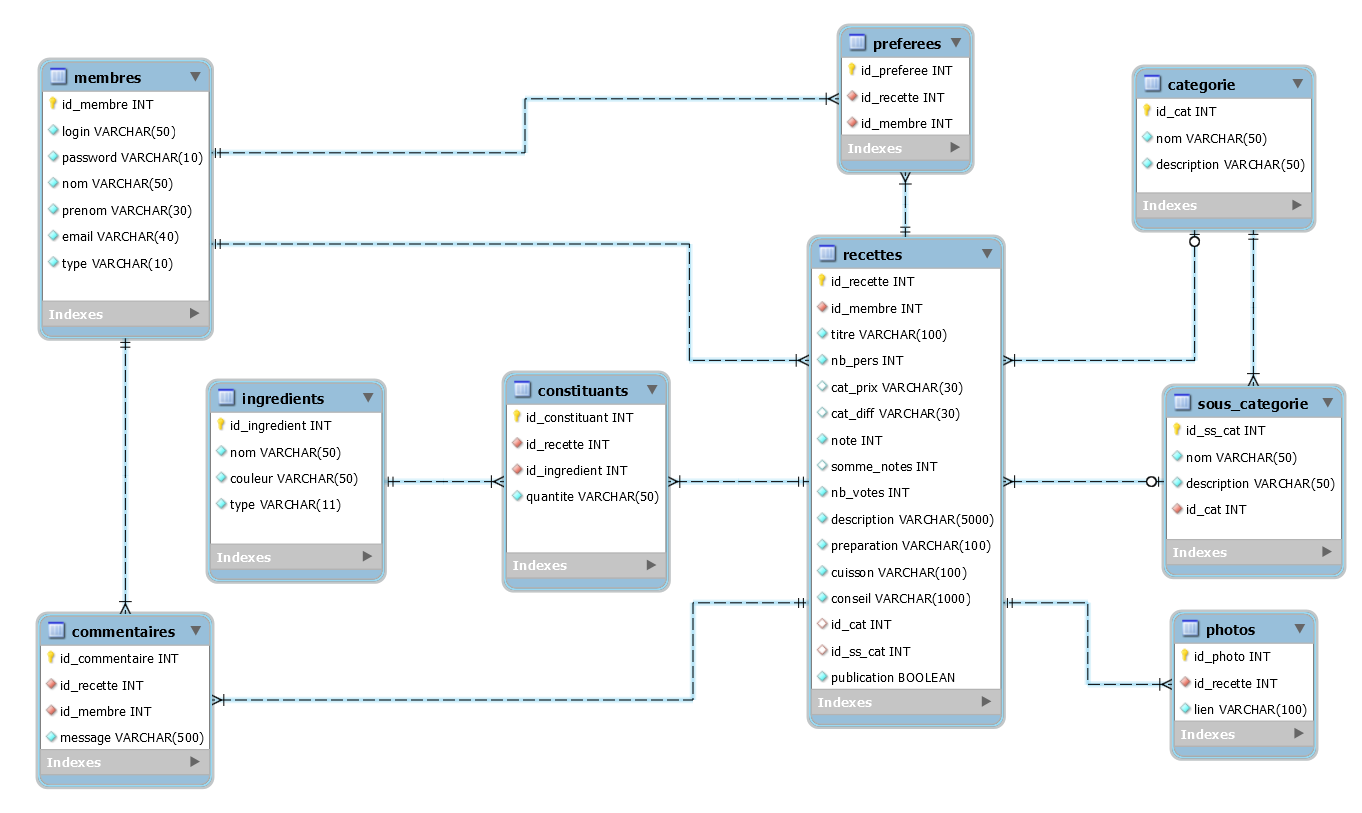
1. « membres », « categorie », « ingredients »
2. « sous-categorie »,
3. « recettes »,
4. « commentaires », « photos », « constituants », « preferees »

L’ordre est listé par nom des tables de la base MySQL.

## Modélisation de la base SQL avec MySqlWorkbench

Voir schéma page suivante.

Extrait graphique de la structure définie



## Enchainement des pages web

Après plusieurs essais « papier », un draft « mockup » du site a été élaboré en Powerpoint, faute d’avoir trouvé sur le web, un outil spécifique simple et gratuit pour le réaliser.

cf. fichier joint



# Architecture matérielle et logicielle

## Architecture matérielle (infrastructure)

Une plate-forme dédiée a été montée. Elle est hébergée sur une VM (Virtual Machine) localisée sur un serveur de tests disponible en Lab.

* adresse VM : 10.194.128.254

Pour le projet, il a été nécessaire d’installer les services suivants :

* Serveur Apache : serveur web
* Serveur PHP : exécution des scripts PHP générant les pages HTML du projet,
* Serveur MySQL : gestion de la base de données
* Serveur Samba : partage de fichier en réseau
* Mutt : logiciel d’envoi de mail sous linux utilisant le serveur de rebond d’Orange-Labs

Cette VM dispose d’un accès SSH et du service Samba ce qui permet de développer depuis les postes bureautiques en utilisant cette machine comme un disque distant.

## Organisation MVC

Le modèle MVC ayant été retenu, ci-dessous, la déclinaison qui en a été faite :

|  |  |  |
| --- | --- | --- |
| BDD (Model) | Affichage (Vues) | Navigation (Controller) |
| * Modèle MySQL : définition des tables (gestion de paramètres en mode « atomique ») * Classes : définition des classes en PHP | * Gestion d’une seule page qui se décompose en plusieurs vues (« sous-pages ») * Découpe de la page en 5 blocs (vues) * en-tête * menu * contenu * commentaire * pied de page (non traité) * CSS | * Gestion des différents modules du modèle : * accueil, * admin, * constituants, * membres, * pied de page, * recette * Gestion de la navigation par des variables de session, |

## Définition des classes et des fonctions de base

Une classe a été définie par table de la base de données.

Les attributs de la classe correspondent aux champs de la correspondante.

Les méthodes sont définies au fur et à mesure des besoins de dev.

## Définition des règles d’affichage des pages

Le délai imparti pour réaliser ce projet étant très court (<30hj à 2), l’utilisation d’un Framework CSS aurait été trop coûteuse en temps. Aussi, le choix d’un développement des pages web plus « traditionnel » s’est imposé.

Les principales règles adoptées sont les suivantes :

* Blocs en div avec id,
* Largeur de page fixe,
* Site bicolore,
* Police unique (Verdana)
* …

# Développement

## Présentation de la structure logicielle

La structure logicielle respecte le modèle MVC. Elle est définie comme ci-dessous :

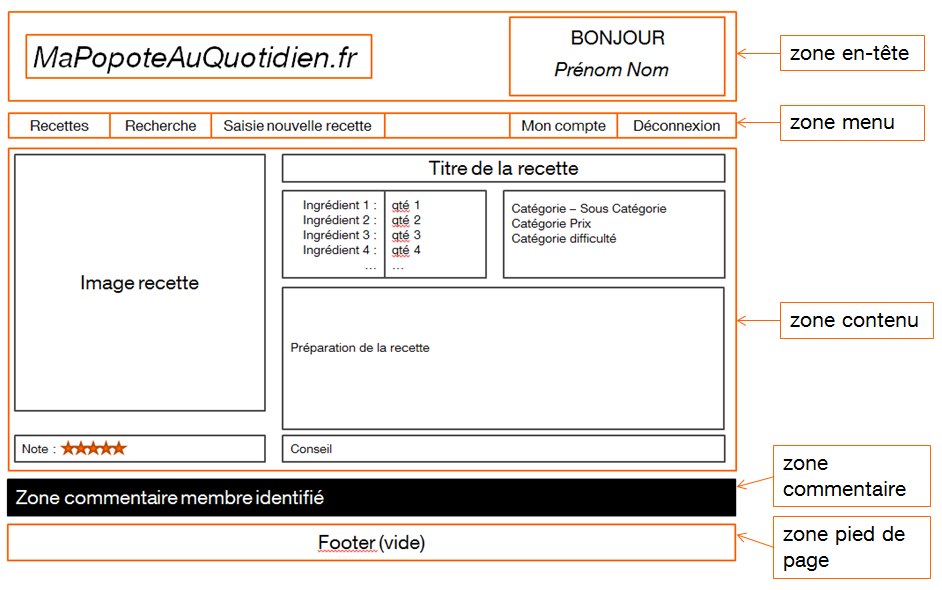
|  |  |
| --- | --- |
|  | **Popote**  **controleurs** : contient les fonctions de traitement et de navigation, ainsi que la génération des blocs HTML affichés dans la section « vues ».  **librairies** : contient les fichiers de configurations et les photos utilisées par les recettes (upload)  **modeles** : contient les fichiers de classes qui permettent de manipuler les données en relation avec la BDD.  **vues** : contient le fichier décrivant la structure de la page et le fichier de la feuille de style CSS |
|  |  |

Les enchainements de pages et la description des fichiers de code sont décrits dans le document, ci-dessous :



## Construction du modèle de page

La page affichée « vue » sur ce site est toujours la même. Elle est constituée de 5 zones, qui sont construites dynamiquement par les fonctions des « contrôleurs » selon les actions des utilisateurs et des données récupérées grâce aux fonctions des « modèles ».



*Description des zones d’affichage du site*

La mise en page est gérée par une feuille de style dans le fichier CSS du site.

Dans chaque zone, des blocs (div) déterminent les caractéristiques d’affichage.

Les éléments contenus dans toutes ces zones sont gérées dynamiquement en fonction de variables de sessions.

Description des différentes zones d’affichage

* **Zone en-tête** : contient le lien de refresh pour afficher une recette de façon aléatoire, ainsi que la bannière d’accueil qui mentionne l’identité du membre connecté.
* **Zone menu** : contient les différents items de menus selon le niveau d’authentification de l’utilisateur connecté (Invité, user, admin)
* **Zone contenu** : c’est la zone la plus importante du site, c’est la zone « dynamique » du site. Elle permet d’afficher les informations en fonction des actions de l’utilisateur (affichage recette, liste, création, admin, gestion de compte….).
* **Zone commentaire** : contient les commentaires d’un membre connecté relatifs à la recette affichée.
* **Zone pied de page**: zone destinée à recevoir des d’informations (copyright, mail vers admin, plan du site …) – non traitée à ce jour

**Restrictions et contraintes d’usage du site :**

L’emploi de balises HTML5 telles que <datalist> impose l’utilisation des browsers Firefox ou Chrome dans un environnement e-buro.

La balise <datalist> est un élément de formulaire permettant de lier une liste de choix à un élément <input>. Elle offre les mêmes fonctionnalités qu’un <select> avec en plus la possibilté de saisir une valeur non listée.

**Nota** : IE est compatible HTML5 à compter de sa version 10, mais e-buro n’est qu’en version 8.

Autre point à noter, via les CSS, les boutons et liens ont pu être personnalisés de manière simple et homogène, sauf pour le bouton <input type =file> car il n’existe pas de CSS permettant de le manipuler. Ce type d’élément n’est modifiable que via javascript (ou jquery).

## Résumé des fonctionnalités du site

Liste des fonctionnalités implémentées pour ce site (en gris, fonctionalités partiellement traitées)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fonctionnalités | | Description | Droits d’usage | | |
| domaine | fonction | invité | user | admin |
| Recettes | afficher une recette |  | x | x | x |
| afficher liste | liste de toutes les recettes validées | x | x | x |
| liste des recettes d’un membre |  | x | x |
| liste de recettes à valider |  |  | x |
| création | création d’une nouvelle recette |  | x | x |
| modification | modification des recettes par leur créateur |  | x | x |
| recherche | recherche multi critères : titre, catégorie, sous-catégorie | x | x | x |
| vote | notation de la recette | x | x | x |
| recette préférée | tag perso sur une recette |  | x | x |
| suppression | gestion par changement de référence du propriètaire de la recette (partiellement traité) |  |  | x |
| validation | autoriser la publication d’une recette |  |  | x |
| Compte | création | création d’un compte pour un nouveau membre | x |  |  |
| modification | modification de certains paramètres du compte |  | x | x |
| connexion | accessible uniquement en mode invité, permet à un membre de se connecter | x |  |  |
| Gestion admin | modification du rôle | changement du rôle d’un membre (admin/user) |  |  | x |
| suppression membre | (non traité) |  |  | x |
| catégorie | création | création d’une nouvelle catégorie dans la liste proposée pour les recettes |  |  | x |
| modification | modification d’une nouvelle catégorie dans la liste proposée pour les recettes |  |  | x |
| sous-catégorie | création | création d’une nouvelle sous-catégorie dans la liste proposée pour les recettes |  |  | x |
| modification | modification d’une nouvelle sous-catégorie dans la liste proposée pour les recettes |  |  | x |
| ingrédient | création | création d’un ingrédient non proposé par la base |  | x | x |
| commentaire | création | ajout un commentaire personnaliser pour chaque recette du site |  | x | x |
| modification | modifie le commentaire personnel préalablement saisi |  | x | x |
| envoi mail | recette | envoyer un mail au créateur de la recette | x | x | x |
| Mdp perdu | envoyer un mail pour retrouver ses paramètres de connexion | x |  |  |
| contact | envoyer un mail vers les « admin » du site [non traité - fonction pied de page en RAF] | x | x |  |
| photo | affectation à une recette | utilisation de la fonction upload |  | x | x |
|  |  |  |  |  |  |

# Bilan du mini-projet

L’objectif principal de ce module NFA021 est de mettre en pratique les éléments vus en cours lors des 2 modules NFA016 & NFA017. Dans cette optique, les choix d’implémentations qui ont été faits, correspondent à ces critères, tout en tenant compte de la contrainte de temps.

Le délai global d’étude et de réalisation pour ce mini projet a été de moins de 30 H.J. à 2 personnes.

Il en résulte que certains aspects ont volontairement été dépriorisés, afin de pouvoir découvrir et développer un plus grand nombre de fonctionnalités.

## Problèmes rencontrés

Bien entendu, beaucoup de problèmes sont apparus lors de la création de ce premier site web, conçu de bout-en-bout, dont en voici les principaux :

* + Prise en main des « nouveaux » outils : MySQLWorkBench, HeidiSQL, Netbeans, Debug sous Firefox…

Tous ces outils étaient nouveaux et ont nécessité un temps d’apprentissage.

* + Définition du modèle SQL & Classes PHP :

Pour démarrer ce projet, le plus difficile a été de formaliser un modèle de données répondant aux critères fixés. Cette étude préliminaire a été un peu longue mais à permis de gagner du temps, par la suite, lors de la phase de développement. L’aide du tuteur « local » a été d’un grand secours et a permis de mener à bien cette étape préparatoire.

* + Requête SQL sur tables croisées :

Ces aspects n’ont pas été abordés lors de la formation, aussi, des recherches complémentaires ont été nécessaires.

* + Différence de comportement d’affichage selon les configurations matérielles et/ou logicielles :

Selon les types de navigateurs (IE, Chrome, Firefox), on sait qu’il peut y avoir des différences de comportement dans l’affichage d’une page web, mais on pourrait s’attendre à ce que, pour une même version de browser, l’affichage soit identique, ce qui n’est pas le cas. Par exemple, pour la taille de la zone de saisie de champs <input> entre 2 PC e-buro avec Firefox en version 38.2.1 avec un affichage à normal 100%, une zone <textarea> peut déborder ?! …sans raison apparente.

* + Gestion des accents / police de caractères (BDD/ Browser/ PHP / HTML) & droits d’usages

En cours HTML, cette problématique d’encodage des polices de caractères avait été abordée, mais elle est globale de bout-en-bout de la chaîne de génération des pages web.

Une autre question s’est posée sur les droits d’usages des polices de caractères. C’est donc une police standard du browser qui a été utilisée, ainsi, le problème ne se pose pas.

* + Définition de la page de création d’une recette :

Lors de la conception de la page de « création d’une recette », le fait d’avoir des tables croisées compliquait énormément la création d’une nouvelle entrée dans la base. Aussi, la création se décompose en 2 étapes :

1. Création d’une recette vide pour récupérer un nouveau ID de recette
2. Création des entrées dans les tables associées, et complétion de la table « recettes »
   * Contraintes externes au projet :

Le planning de la période de réalisation du mini-projet NFA021 a légèrement glissé, afin de terminer les 2 modules précédents (examens et cours), et n’a pu commencer que début décembre. C’est ajouté à cela des contraintes de formation planifiées de longue date ainsi que les congés de fin d’année. Finalement, à la date de la soutenance, moins de 30HJ ont pu être consacré à la réalisation de ce mini-projet.

## Reste à faire

Le projet n’est pas totalement terminé, mais les points suivants ont été identifiés comme les plus pertinents dans le « reste à faire » :

* + Corrections des bugs connus
* Calcul de la note d’une recette,
* affichage des recettes non validées en page d’accueil
* …
  + Validation E2E complète du site dans les 3 modes d’accès :
* Invité,
* Membre authentifié,
* Administrateur
  + Ajout de l’affichage des recettes préférées via un critère de recherche (ou sur un item de menu spécifique), la gestion étant déjà opérationnelle.
  + Compléter le pied de page :
* possibilité d’envoyer un mail aux administrateurs du site,
* plan du site,
* mentions légales,
* copyright
* …
  + Gestion de la suppression d’éléments dans la BDD (long à traiter car problématique de gestion des données croisées à prendre en compte comme déjà précisé dans les paragraphes ci-dessus),
  + …et bien autres choses pour rendre ce site publiable.

# Conclusion

Le mini-projet nous a permis de mettre en application les connaissances étudiées dans les UE NFA016 et NFA017.

Ce site n’est pas finalisé mais respecte assez bien le cahier des charges proposé avec une couverture assez complète du programme étudié. Ce site est certainement perfectible, mais globalement, fonctionne correctement.

Cette réalisation nous a fait monter en compétences en passant de la théorie à la pratique. Cette première expérience de développement de site web nous a amené à faire des choix (techniques, implémentation, architecture…) qui nous ont parus pertinents, au moment de la décision, mais il s’avère, avec le recul, que certains de ces choix n’étaient peut-être pas tous très judicieux (manque d’expérience et de temps).

Ce travail collaboratif, réalisé en binôme, nous a imposé rigueur et organisation, afin gérer au mieux les contraintes et problèmes rencontrés tout au long du projet. La découpe des taches s’est faite naturellement en exploitant au mieux les prédispositions de chacun.

Après 2 mois de cours « intensifs et éprouvants », ce 3ème module, plus ludique, nous a récompensé des efforts du début de la formation.

# Remerciements

Nous tenons à remercier toutes les personnes qui nous ont aidé et ainsi contribué au succès de ce mini-projet. Grand merci à :

* + Nicolas Baupin, notre formateur,
  + François Marie, notre Scrum master,
  + Camille Oudot, notre tuteur,
  + L’ensemble des équipes SOFT Lannion pour leur accueil,