WEISHAR Damien

Titre professionnel de Développeur Web / Web Mobile

Par l’ADRAR, années 2021/2022

Dossier projet

IØLR – création d’un site internet pour un restaurateur

Table des matières

[REMERCIEMENTS 3](#_Toc105600613)

[INTRODUCTION 4](#_Toc105600614)

[I) COMPETENCES PROFESSIONNELLES 5](#_Toc105600615)

[II) ANALYSE DU BESOIN 6](#_Toc105600616)

[1) Présentation de l’entreprise 6](#_Toc105600617)

[2) Contexte – Besoin 6](#_Toc105600618)

[a) Contexte : 6](#_Toc105600619)

[b) Besoin : 6](#_Toc105600620)

[3) Cible 7](#_Toc105600621)

[4) Analyse marketing 7](#_Toc105600622)

[5) Contraintes techniques 8](#_Toc105600623)

[6) Graphisme 11](#_Toc105600624)

[a) Charte graphique 11](#_Toc105600625)

[III) SPECIFICATIONS FONCTIONNELLES 12](#_Toc105600626)

[1) Cas d’utilisation 12](#_Toc105600627)

[2) Diagramme d’activité 14](#_Toc105600628)

[a) Ajouter un ingrédient 15](#_Toc105600629)

[b) Supprimer un ingrédient 17](#_Toc105600630)

[3) Diagramme de séquence 20](#_Toc105600631)

[a) Ajouter un ingrédient 22](#_Toc105600632)

[b) Supprimer un ingrédient 23](#_Toc105600633)

[4) Maquettage 24](#_Toc105600634)

[IV) CONCEPTION 30](#_Toc105600635)

[1) Modèle Conceptuel des Données 30](#_Toc105600636)

[2) Modèle Logique des Données 31](#_Toc105600637)

[3) Le SQL 32](#_Toc105600638)

[V) ARBORESCENCE 34](#_Toc105600639)

[VI) OUTILS TECHNIQUES UTILISES 35](#_Toc105600640)

[VII) FONCTIONNALITES 38](#_Toc105600641)

[1) Ajouter un Ingrédient dans la base de données 38](#_Toc105600642)

[a) La page Web 38](#_Toc105600643)

[b) Le code 39](#_Toc105600644)

[2) Supprimer un Ingrédient dans la base de données 55](#_Toc105600645)

[a) La page Web 55](#_Toc105600646)

[b) Le code 56](#_Toc105600647)

[VIII) CONCLUSION 65](#_Toc105600648)

[IX) ANNEXES 67](#_Toc105600649)

# REMERCIEMENTS

Tout d’abord, je voudrais remercier le centre de formation professionnelle de l’ADRAR (Association pour le Développement Régional et Appui au Reclassement) et spécialement le Pôle Numérique de Toulouse, qui m’a permis d’intégrer la formation de Développeur Web/Web Mobile, et ainsi d’entamer ma reconversion professionnelle.

Je remercie Mme CALMETTES Florence (Coordinatrice de la filière), Mr MITHRIDATE Mathieu (formateur référent) ainsi que tous les formateurs qui sont venus nous faire cours, pour le temps qu’ils nous ont consacré et leur disponibilité hors des heures de formation.

Grâce à cette formation, j’ai pu rencontrer des personnes formidables dans ma promotion, appelée CyberDev’ Club. Plusieurs m’ont aidé et soutenu dans des moments compliqués tout au long de l’année, comme Mlle COSTA Laure, Mlle SNEDDER Katherine et Mr DEBONO Damien.

Je tenais à remercier particulièrement Mlle COSTA, sans qui je n’aurai pas trouvé de stage.

Ensuite, je remercie Mr PEYRARD Fabrice, mon maître de stage. Grâce à lui, j’ai pu progresser tout au long de ces deux mois et demi de stage, et renforcer mes capacités.

Enfin, je tenais à remercier tout particulièrement Mr LAFAYE DE MICHEAUX Yoan, Ingénieur en Etudes et Développements (confirmé), pour son aide, son soutien psychologique, et tout le temps qu’il m’a consacré.

# INTRODUCTION

Je m’appelle Damien WEISHAR, j’ai 32 ans. Je suis titulaire d’une Licence LEA (Langues Etrangères Appliquées) en Anglais/Espagnol et d’une Licence STAPS (Sciences et Techniques des Activités Physiques et Sportives). J’ai été animateur maternelle pendant plusieurs années (8 ans). J’ai choisi de me réorienter dans le monde de l’informatique, et plus précisément dans le développement. En effet, ma curiosité à vouloir connaître le fonctionnement des sites, la manière dont ils sont créés, le langage et le code correspondants m’ont poussé vers cette voie-là.

Pour cela, j’ai intégré l’ADRAR dans le but d’obtenir le diplôme de Développeur Web / Web Mobile. Durant cette formation, un projet fil rouge devait être réalisé. Ma passion pour les mangas m’a guidé dans cette direction. C’est tout naturellement que j’ai voulu réaliser un site de partage de mangas d’occasion (Manga4All).

Dans le cadre de la formation, je devais effectuer un stage d’une durée de 2 mois et demie, durant la période allant du 14 mars 2022 au 20 mai 2022. C’est la société FPNET qui m’a accueilli.

La mission en distanciel consistait à créer un site internet pour un client de FPNET, la société de restauration IØLR, située à la Rochelle.

Le premier contact s’est fait par mail. Puis nous avons échangé quelques appels téléphoniques qui ont mené à une offre de stage.

Après avoir signé la convention de stage, nous avons eu des rendez-vous téléphoniques afin de discuter de l’avancée du projet. Le premier point téléphonique m’a permis d’établir un cahier des charges. Le suivi des échanges se faisaient via un fichier un Google Doc.

Au cours de la mission, la priorité a été mise sur la structure de la base de données ainsi que la page administrateur.

Une fois la mission terminée, la page administrateur était fonctionnelle ainsi que la base de données. La version actuelle est prédestinée à évoluer de nouveau.

La mission réalisée étant plus professionnelle et aboutie, je vous présenterai celle-ci au lieu du projet fil rouge (Manga4All).

# COMPETENCES PROFESSIONNELLES

Deux activités-types sont attendues de la part du Développeur Web / Web Mobile :

1. Développer la partie front-end d’une application web ou web mobile en intégrant les recommandations de sécurité
2. Développer la partie back-end d’une application web ou web mobile en intégrant les recommandations de sécurité

De chaque activité-type découlent plusieurs compétences professionnelles qui leur sont associées :

1. Développer la partie front-end d’une application web ou web mobile en intégrant les recommandations de sécurité
2. Maquetter une application
3. Réaliser une interface utilisateur web statique et adaptable
4. Développer une interface utilisateur web dynamique
5. Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce
6. Développer la partie back-end d’une application web ou web mobile en intégrant les recommandations de sécurité
7. Créer une base de données
8. Développer les composants d’accès aux données
9. Développer la partie back-end d’une application web ou web mobile
10. Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce.

Ces compétences sont essentielles à l’obtention du titre de Développeur Web / Web Mobile.

# ANALYSE DU BESOIN

## Présentation de l’entreprise

L'entreprise FPNET est un cabinet d'expertise en Réseaux Informatiques et Cyber Sécurité. Elle a été créée en 2016 par Monsieur PEYRARD (Enseignant-Chercheur en Réseaux Informatique et Cyber Sécurité) et son activité s'adresse aux TPE/PME pour l'analyse et l'audit de leurs réseaux informatiques, la proposition de solutions techniques d'évolution et sécurisation de l'infrastructure. L'entreprise propose aussi des formations en sécurité informatique et CyberSécurité, de l'acculturation et règles d'hygiènes numériques jusqu'aux techniques avancées de traitement et filtrages des flux numériques.

## Contexte – Besoin

### Contexte :

L'entreprise FPNET est un prestataire pour la société de restauration IØLR (co-gérée par Monsieur PEYRARD), en cours de développement à La Rochelle, dans un concept de restauration de proximité avec des produits de saisonnalité, travaillant avec des producteurs d'agriculture et d'élevage raisonnés. IØLR propose une restauration à emporter avec l'usage de consignes spécifiques et un cycle de livraison propre.

Suite à la vente de leur précédent restaurant « Au Jardin », situé à Angoulême, la société IØLR a décidé d’ouvrir un nouveau restaurant. Cependant, aucune date d’ouverture n’est prévue à ce jour.

En partant de la même base, ils ont eu la volonté de faire évoluer le projet.

### Besoin :

L’idée n’était pas un simple restaurant classique.

Basé sur un site existant, la mission consistait à créer une version 2. Cette version, développée à l’aide du framework CSS Bulma, s’articulera autour d’une base de données (produits, clients…). Elle devra intégrer un système de paiement en ligne. Le site devra également intégrer une liste de liens tels que Edgar (menu QR-Code), avis Google et Instagram.

Les fonctionnalités attendues sont :

* une page administrateur permettant de faire toutes les modifications des menus,
* une page client lui permettant de composer sa salade et sa commande,
* gérer la vente à emporter,
* gérer le click and collect,
* gérer les gamelles consignées et les réductions associées,
* gérer la vente de produits dérivés (tire-bouchon, verre, assiette, tablier, t-shirt, veste…),
* créer des menus avec options,
* gérer le paiement en ligne,
* intégrer des liens pour des avis Google et Instagram,
* intégrer un menu QR-Code.

## Cible

IØLR cible tout type de client, de tout âge.

Dans le but de mélanger tradition et modernité, il sera donné la possibilité au client d’accéder au menu de manière physique ou numérique.

## Analyse marketing

L’analyse marketing, ou étude de marché, est la première étape dans le développement marketing d’une société. Elle est cruciale dans la mise en place d’un produit ou d’un service pour mettre en accord notre offre avec la demande.

Pour cela, il faut se renseigner, observer, analyser et comprendre le futur environnement.

Pour se faire, une analyse SWOT a été effectuée :

|  |  |  |  |
| --- | --- | --- | --- |
| Strengths  (Forces) | Weaknesses (Faiblesses) | Opportunities (Opportunités) | Threats  (Menaces) |
| * travailler en local (producteurs) | * pas encore ouvert | * nouveau sur le marché * peu de concurrence spécifique | * les autres restaurants déjà bien intégrés dans le paysage |

## Contraintes techniques

Sur la base du site internet de son ancien restaurant, Monsieur PEYRARD a créé une première version de son futur site sur la même base.

Il a pris la décision d’utiliser certaines technologies. Par conséquent, j’ai utilisé ses choix techniques afin de développer la version 2. La maintenance/évolution en est facilitée après la mission.

Les technologies utilisées sont les suivantes :

* côté front-end :
  + HTML5 (HyperText Markup Language): langage utilisé pour créer et représenter le contenu d’une page web et sa structure.



Figure Logo HTML5

* + CSS3 (Cascading Sheet Style): langage informatique permettant de mettre en forme la page web.



Figure Logo CSS3

* + JavaScript: langage de programmation de scripts pour les pages web les rendant interactives.

Figure Logo JavaScript

* + Bulma: framework CSS développé par Jérémy Thomas, sous licence MIT. Il est semblable à Bootstrap.

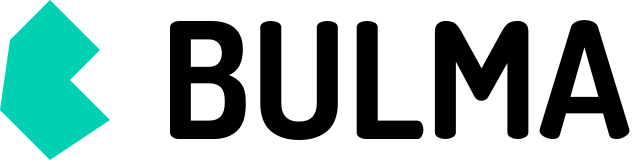


Figure Logo Bulma

* + PHP (Hypertext Preprocessor): langage de programmation libre, utilisé pour produire des pages web dynamiques via un serveur http, mais aussi de façon locale. C’est un langage orienté objet.



Figure Logo PHP

* côté back-end :
  + MySQL : système de gestion de bases de données relationnelles.



Figure Logo MySQL

* + phpMyAdmin : application web de gestion pour les systèmes de gestion de base de données MySQL et MariaDB.

Figure Logo phpMyAdmin

* côté serveur :
  + VPS (Virtual Private Server) sur onetsolutions



Figure Logo OnetSolutions

Le système d’exploitation hébergeant le site est Ubuntu server 20.04 LTS et s’exécute sur une machine virtuelle.

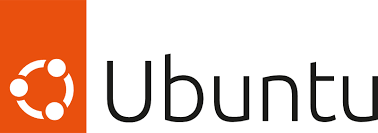


Figure Logo Ubuntu

## Graphisme

### Charte graphique

La société n’a pas encore de logo, ni une identité qui lui est propre pour son restaurant.

Cependant, je devais me baser sur le site préexistant, et surtout sur l’identité visuelle de Bulma.



Figure Charte graphique de Bulma  
Source : bulma.io

Les principales les couleurs utilisées sont :

* Primary
* Link
* Info
* Success
* Warning
* Danger

# SPECIFICATIONS FONCTIONNELLES

## Cas d’utilisation

Les diagrammes de cas d’utilisation sont une représentation modélisée des fonctionnalités d’une application pour le développeur.

Ils décrivent les interactions entre le système et ses acteurs.

Ils se composent :

* du système : représenté par un grand quadrilatère, avec en haut le nom du système
* d’acteurs : représentant un rôle d’un utilisateur qui interagit avec le système. Il est symbolisé par un bonhomme bâton, avec son rôle en-dessous.
* de traits : nommés Association, représentant les droits d’accès à la fonctionnalité
* de traits fléchés : nommés Generalization, représentant la notion d’héritage (transmission des droits).
* de bulles : représentant les cas d’utilisation (un cas d’utilisation équivaut à une fonctionnalité) accessibles par les acteurs auxquels elles sont reliées

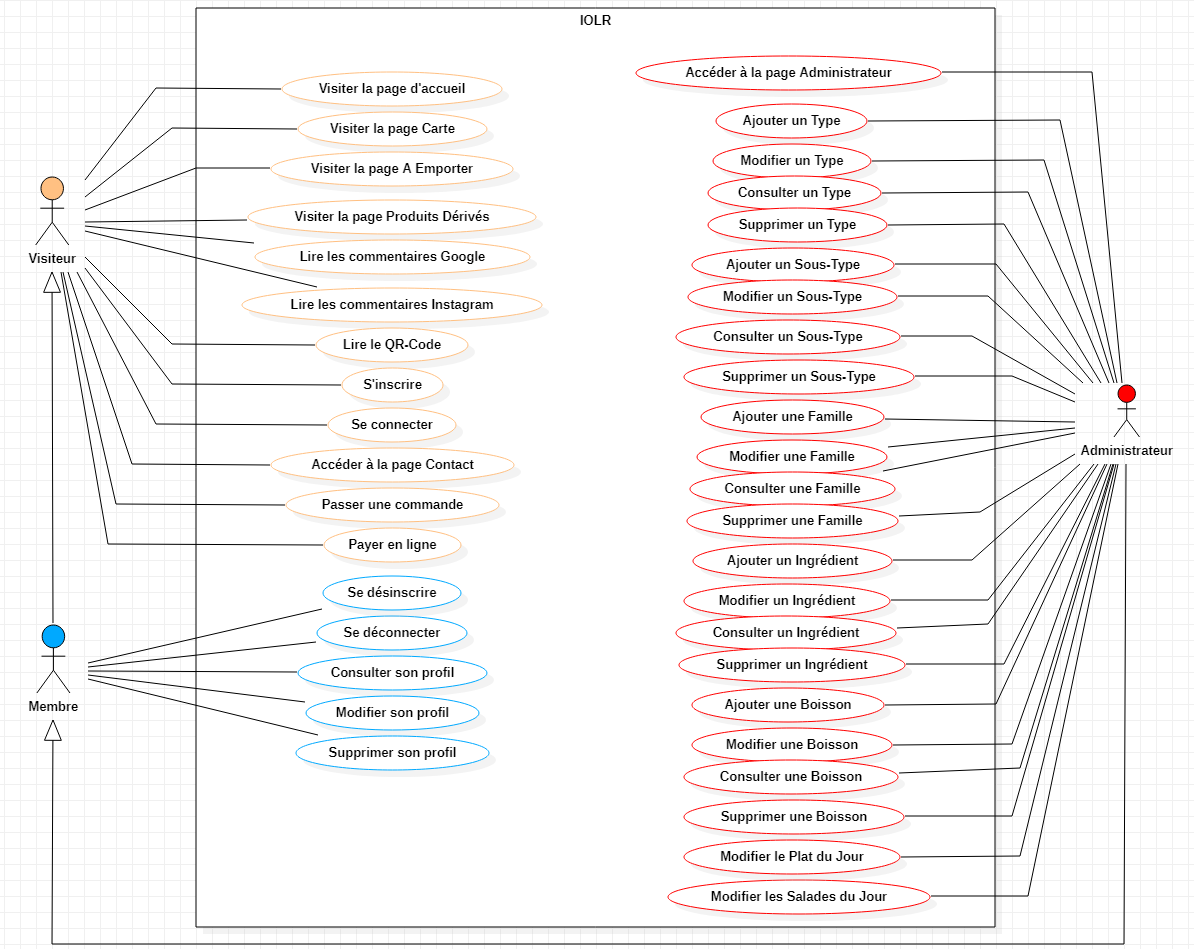


Figure Use Case

Vous pourrez retrouver ce diagramme en annexe, page 60.

Dans mon diagramme d’utilisation pour la société IØLR, il y a trois acteurs : le Visiteur (avec la tête de couleur jaune), le Membre (avec la tête de couleur bleu), et enfin l’Administrateur (avec la tête de couleur rouge).   
Chaque acteur a des droits d’accès à certaines fonctionnalités.

Le Visiteur peut :

* Visiter la page d’Accueil
* Visiter la page Carte
* Visiter la page A Emporter
* Visiter la page Produits dérivés
* Lire les commentaires Google
* Lire les commentaires Instagram
* Lire le QR-Code
* S’inscrire
* Accéder à la page Contact
* Passer une commande
* Payer en ligne

Le Membre peut :

* Se désinscrire
* Se déconnecter
* Consulter son profil
* Modifier son profil
* Supprimer son profil

L’Administrateur peut :

* Accéder à la page Administrateur
* Ajouter un Type
* Modifier un Type
* Consulter un Type
* Supprimer un Type
* Ajouter un Sous-Type
* Modifier un Sous-Type
* Consulter un Sous-Type
* Supprimer un Sous-Type
* Ajouter une Famille
* Modifier une Famille
* Consulter une Famille
* Supprimer une Famille
* Ajouter un Ingrédient
* Modifier un Ingrédient
* Consulter un Ingrédient
* Supprimer un Ingrédient
* Ajouter une Boisson
* Modifier une Boisson
* Consulter une Boisson
* Supprimer une Boisson
* Modifier le Plat du Jour
* Modifier les Salades du Jour

En raison du principe d’héritage, les droits d’accès sont hiérarchisés.  
En effet, le Visiteur a des droits restreints. Le Membre a accès à plus de fonctionnalités (dont celles du Visiteur) et l’Administrateur a tous les droits (dont ceux du Visiteur et du Membre).

## Diagramme d’activité

C’est le processus algorithmique de chaque cas d’utilisation. Il démontre les flux entre les actions.

Il est composé :

* d’acteurs : qui participent au cas d’utilisation,
* de swimlane (ou ligne de vie) : qui représentent le rôle de l’acteur,
* d’un état initial : représenté par un point noir, c’est le départ de l’action,
* d’un accept signal : représenté par un rectangle tronqué derrière,
* d’un send signal : représentés par une flèche,
* d’actions : représentées par des quadrilatères aux bords arrondis,
* de control flow : représentés par des traits fléchés,
* d’un point de décision : représenté par un losange,
* d’un état final : représenté par un point noir dans un cercle, c’est la fin de l’action.

### Ajouter un ingrédient

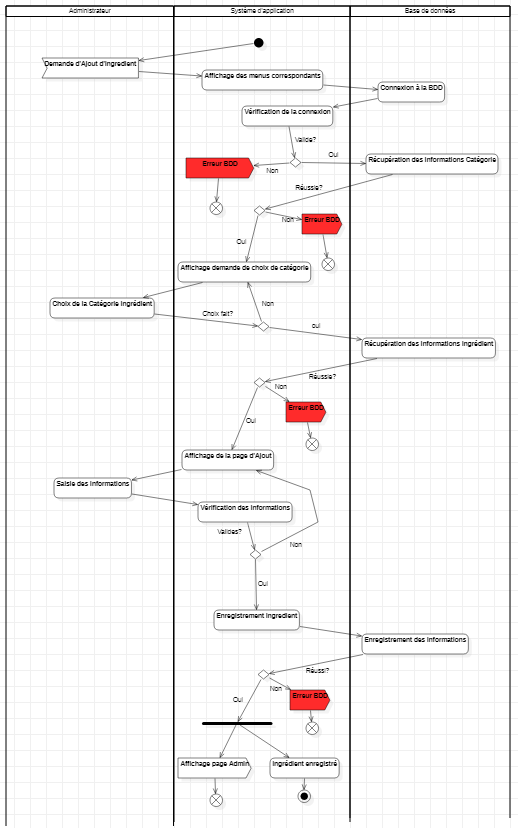


Figure Diagramme d'activité

*Vous pourrez retrouver ce diagramme en annexe, pages 61 – 62.*

Dans ce diagramme d’activité de la fonctionnalité **Ajouter un ingrédient**, il y a trois acteurs :

* l’Administrateur,
* le Système d’application,
* la Base de données.

L’**état initial** se trouve dans la ligne de vie du Système d’application. Celui-ci est en attente d’une action de la part de l’Administrateur (send signal) pour commencer le flux.

L’Administrateur fait une Demande d’Ajout d’ingrédient au Système d’application. Celui-ci fait un affichage des menus correspondants.

Dans le menu Catégorie, j’ai besoin de récupérer des données pour qu’il soit affiché correctement. Pour cela, le Système d’application fait une demande de connexion à la Base de données.

Celle-ci renvoie un message vers le Système d’application.

Celui-ci vérifie le statut de la connexion. A-t-elle réussi ? Ou a-t-elle échoué ? Selon le cas, l’action est différente.

Si la connexion n’est pas valide, le Système d’application envoie un message d’erreur (un send signal) et le flux s’arrête ici (flux final).

Si la connexion est valide, la Base de données récupère les informations concernant le menu Catégorie.

Si la récupération des données échoue, le Système d’application envoie un message d’erreur (un send signal) et le flux s’arrête ici (flux final).

Si la récupération des données est réussie, le Système d’application affiche le menu Catégorie avec les choix (récupérés en base de données).

C’est au tour de l’Administrateur. Il doit faire un choix dans le menu Catégorie.

Si le choix n’est pas fait, le Système d’application sera toujours en attente d’un choix.

Si celui-ci est fait, une demande de récupération de données est effectuée auprès de la Base de données (informations Ingrédient).

Le Système d’application vérifie si la récupération des données est réussie.

Si non, il envoie un message d’erreur (un send signal) et le flux s’arrête ici (flux final).

Si oui, le Système d’application effectue plusieurs actions en même temps : un send signal pour afficher la page Admin et le flux s’arrête ici (flux final), et une action notifiant que l’ingrédient a été enregistré.

C’est l’étape finale. De ce fait, un **point final** est mis en place, fermant le flux initialisé par le point initial au départ de l’action.

### Supprimer un ingrédient

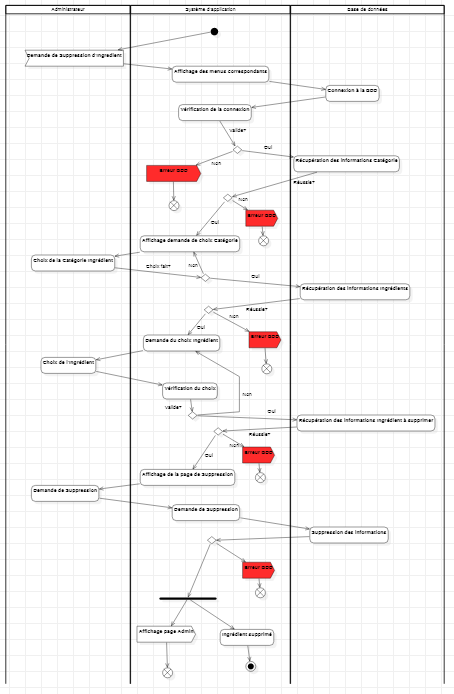


Figure Diagramme d'activité

*Vous pourrez retrouver ce diagramme en annexe, pages 63-64.*

Dans ce diagramme d’activité de la fonctionnalité **Supprimer un ingrédient**, il y a trois acteurs :

* l’Administrateur,
* le Système d’application,
* la Base de données.

Le point initial se trouve dans la ligne de vie du Système d’application. Celui-ci est en attente d’une action de la part de l’Administrateur (send signal) pour commencer le flux.

L’Administrateur fait une Demande de Suppression d’ingrédient au Système d’application. Celui-ci fait un affichage des menus correspondants.

Dans le menu Catégorie, j’ai besoin de récupérer des données pour qu’il soit affiché correctement. Pour cela, le Système d’application fait une demande de connexion à la Base de données.

Celle-ci renvoie un message vers le Système d’application.

Celui-ci vérifie le statut de la connexion. A-t-elle réussi ? Ou a-t-elle échoué ? Selon le cas, l’action est différente.

Si la connexion n’est pas valide, le Système d’application envoie un message d’erreur (un send signal) et le flux s’arrête ici (flux final).

Si la connexion est valide, la Base de données récupère les informations concernant le menu Catégorie.

Si la récupération des données échoue, le Système d’application envoie un message d’erreur (un send signal) et le flux s’arrête ici (flux final).

Si la récupération des données est réussie, le Système d’application affiche le menu Catégorie avec les choix (récupérés en base de données).

C’est au tour de l’Administrateur. Il doit faire un choix dans le menu Catégorie.

Si le choix n’est pas fait, le Système d’application sera toujours en attente d’un choix.

Si celui-ci est fait, une demande de récupération de données est effectuée auprès de la Base de données (informations Ingrédient).

Le Système d’application vérifie si la récupération des données est réussie.

Si non, il envoie un message d’erreur (un send signal) et le flux s’arrête ici (flux final).

Si oui, le Système d’application affiche le menu correspondant et demande à l’Administrateur de faire un choix.

L’administrateur fait son action de choix de l’ingrédient qu’il veut supprimer.

Le Système d’application fait une vérification du choix.

Si l’Administrateur n’a pas choisi d’ingrédient, le Système d’application restera en attente d’action.

Si l’Administrateur a fait son choix et qu’il est validé par le Système d’application, celui-ci fait une demande de récupération des informations de l’ingrédient à supprimer à la Base de données.

Si cette récupération n’est pas réussie, le Système d’application envoie un message d’erreur (un send signal) et le flux s’arrête ici (flux final).

Si cette récupération est réussie, le Système d’application affiche la page de Suppression d’ingrédient.

L’administrateur fait alors une demande de Suppression, et cette demande est transmise à la Base de données par le Système d’application.

Si la suppression n’est pas effective, le Système d’application envoie un message d’erreur (un send signal) et le flux s’arrête ici (flux final).

Si oui, le Système d’application effectue plusieurs actions en même temps : un send signal pour afficher la page Admin et le flux s’arrête ici (flux final), et une action notifiant que l’ingrédient a été supprimé.

C’est l’étape finale. De ce fait, un point final est mis en place, fermant le flux initialisé par le point initial au départ de l’action.

## Diagramme de séquence

C’est la représentation graphique et chronologique de l’action d’un cas d’utilisation. Il expose en détails la façon dont les opérations sont effectuées entre les objets (acteurs et système).

Le diagramme de séquence est toujours lu du haut vers le bas, illustrant l’ordre dans lequel les messages sont envoyés entre les objets.

L’axe vertical représente le temps.

L’axe horizontal représente les objets/acteurs impliqués dans l’interaction.

Il est composé :

* d’objets/acteurs : qui participent à l’action, situés dans la partie supérieure (facilitant son identification),
* de lifeline : ligne verticale représentant sa ligne de vie,
* de messages : représentés par une ligne fléchée directionnelle, symbolisant la communication entre les objets. Au-dessus se trouve un texte informant du message envoyé, et est numéroté,
* de barre d’activation : représenté par un rectangle, illustrant la période pendant laquelle un objet effectue une action,
* d’évènements : qui sont le début et la fin d’un message.

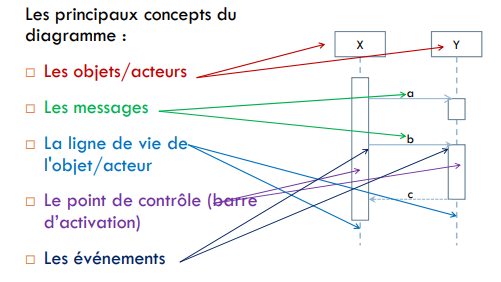


Figure Diagramme de séquence  
Source : internet

Dans un diagramme de séquence, il y a trois cas possibles :

* le **scenario nominal** : c’est le cas où la fonctionnalité se déroule correctement.
* le **cas d’erreur** : c’est le cas où la fonctionnalité ne se déroule pas correctement et reporte un message d’erreur, terminant le flux et mettant fin au scénario nominal.
* le **cas alternatif** : c’est le cas où la fonctionnalité se déroule différemment du scénario nominal, mais ne terminant pas le flux, ni le scénario nominal.

Ici, c’est le **scenario nominal** qui est présenté.

La fonctionnalité est décrite action après action et suit le flux de l’information entre l’Administrateur, le Système d’application et la Base de données.

Sur le côté, sous forme de notes, se trouvent les scenarii alternatifs et d’erreurs, qui décrivent les actions en cas de non application du scénario nominal.

Vous trouverez ci-après les diagrammes de séquence de la fonctionnalité **Ajouter** un ingrédient, et **Supprimer** un ingrédient.

### Ajouter un ingrédient

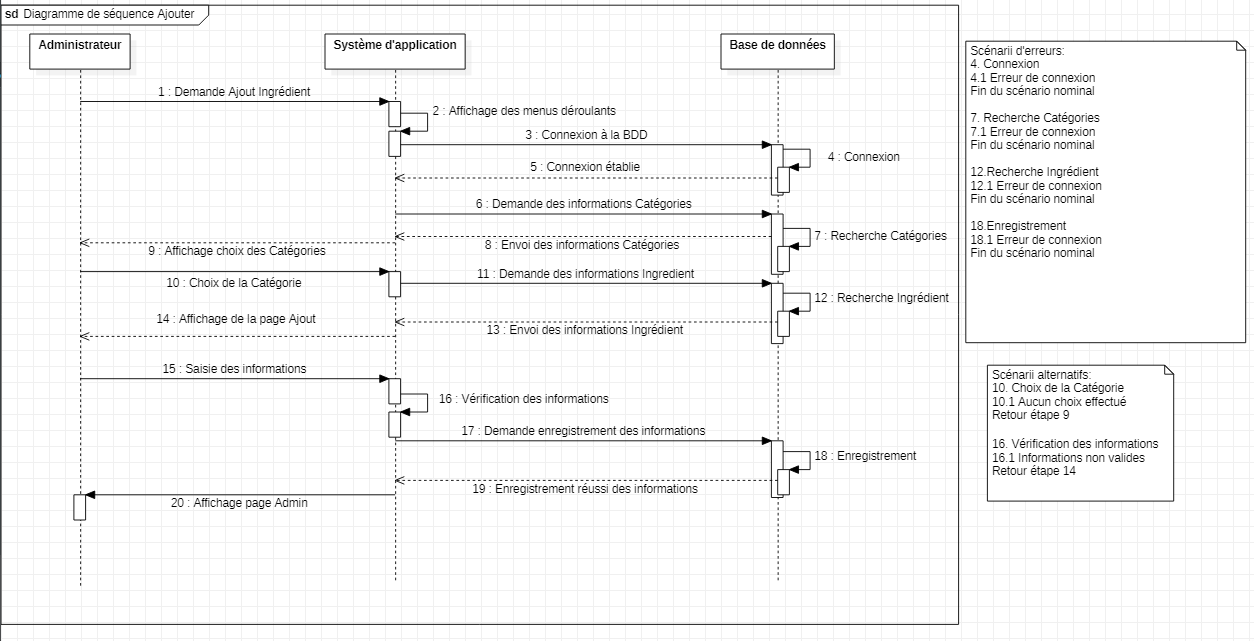


Figure Diagramme de séquence

### Supprimer un ingrédient

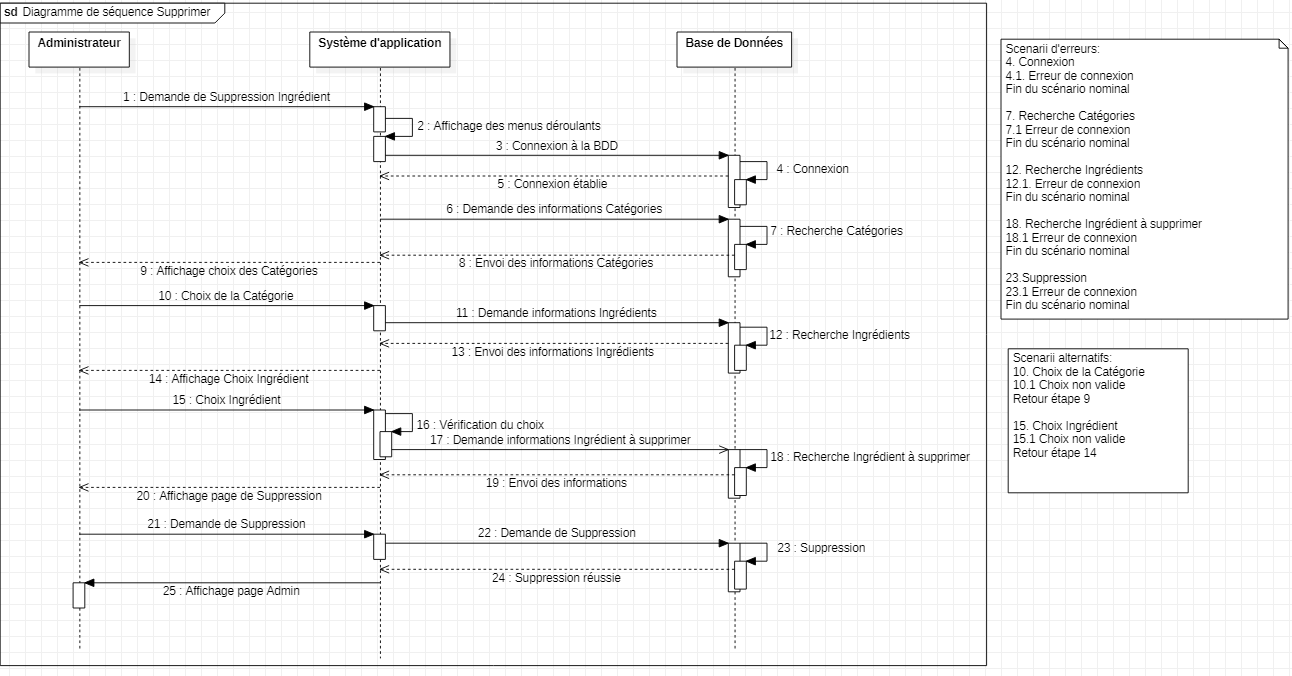


Figure Diagramme de séquence

## Maquettage

Le maquettage est une représentation du concept graphique d’un site web en création ou en refonte.

Il englobe les bases du site : le contenu, l’architecture de l’information et une description simple des interactions prévues

La page Administrateur devait rassembler un maximum de fonctionnalités.

La principale contrainte imposée par Monsieur PEYRARD était d’avoir absolument des menus déroulants.

Il y a eu plusieurs idées/étapes pour le maquettage :

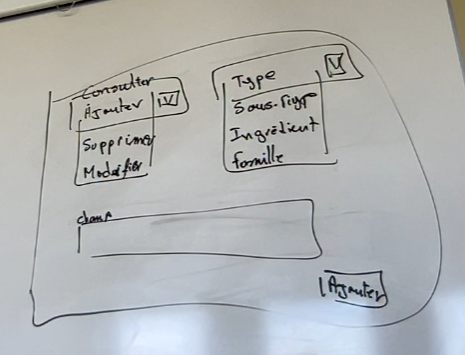


Figure Version 1

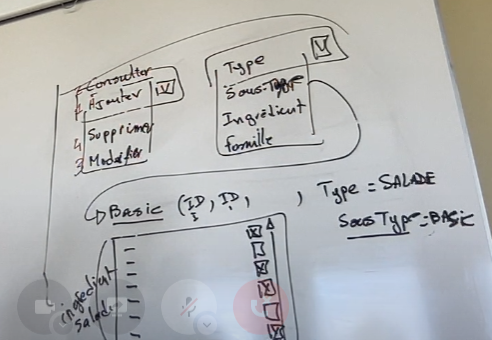


Figure Version 1.1

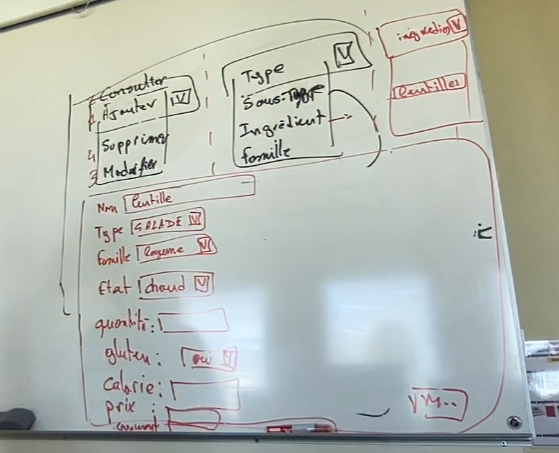


Figure Version 2

Voici le maquettage final sur lequel nous nous sommes mis d’accord :



Figure Version 2.1

Ici, l’administrateur a la possibilité d’effectuer plusieurs actions grâce aux différents menus déroulants. Le 3ème menu sera différent selon la Catégorie qu’il a sélectionnée.

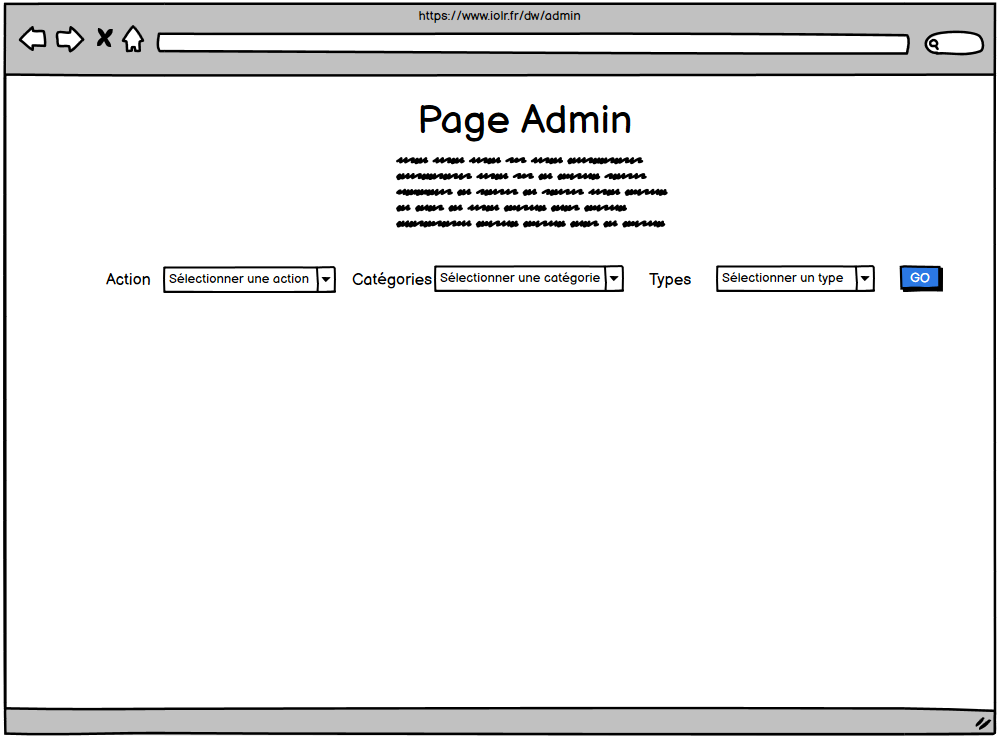


Figure Maquettage de la page Admin

Ici, l’administrateur a sélectionné l’action Ajouter, puis la catégorie Ingrédient et enfin a cliqué sur le bouton Go.

La page d’Ajout apparaît. Il n’aura plus qu’à remplir les champs, choisir dans les menus déroulants (Type, Famille, Etat, Gluten), et cliquer sur **Ajouter** afin d’ajouter l’ingrédient à la base de données.

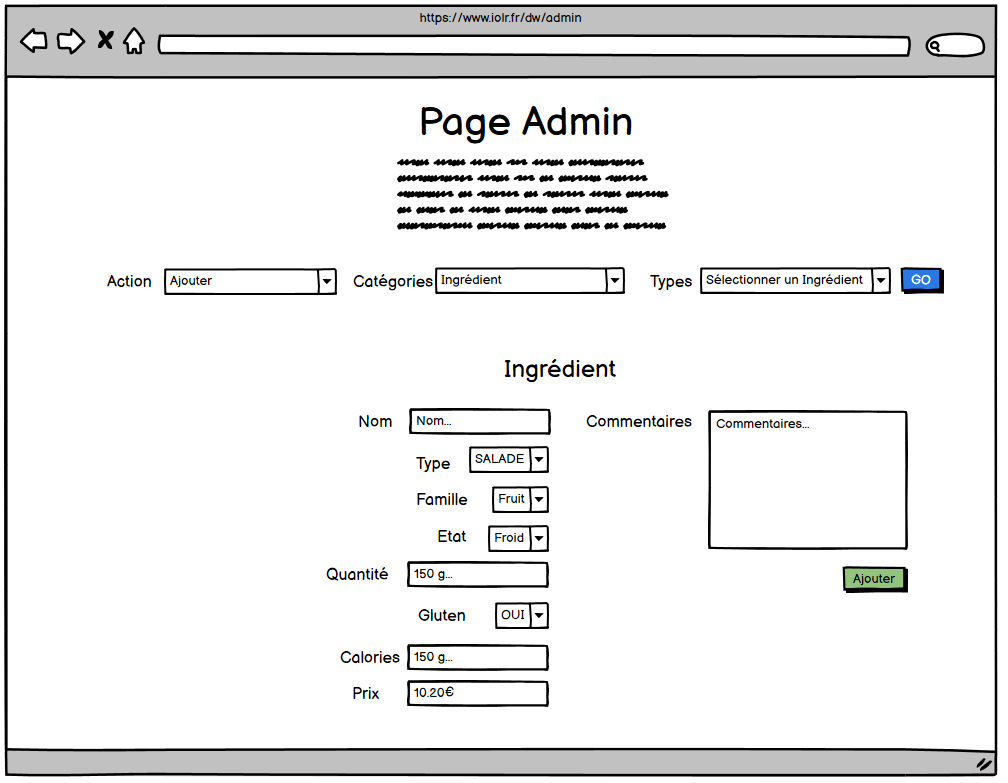


Figure Maquettage de la page Ajout d’ingrédient

Ici, l’administrateur a sélectionné l’action de suppression, puis la catégorie Ingrédient et enfin l’ingrédient qu’il veut supprimer et a cliqué sur Go. La page de Suppression apparaît.

Il n’aura plus qu’à cliquer sur le bouton **Supprimer** pour déclencher la fonction permettant de supprimer dans la base de données.

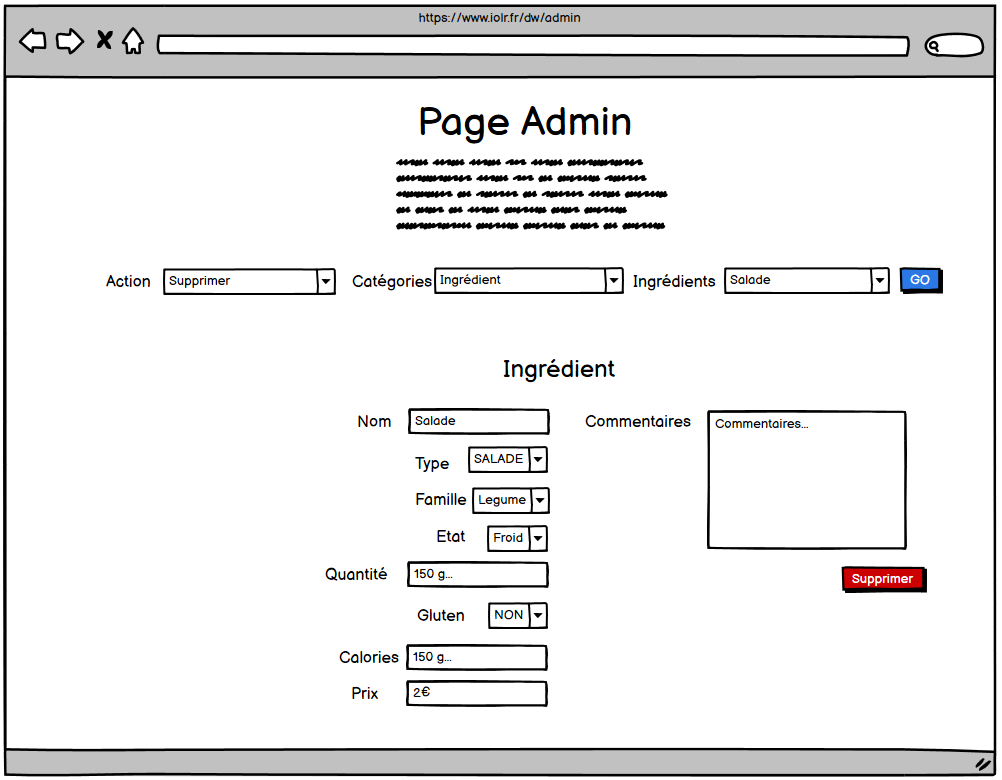


Figure Maquettage de la page Suppression d’ingrédient

# CONCEPTION

## Modèle Conceptuel des Données

Le MCD, ou Modèle Conceptuel des Données), est une représentation graphique permettant de comprendre facilement comment les différents éléments sont liés entre eux. Il structure la donnée en faisant abstraction de la structure physique.

Les informations sont représentées logiquement en utilisant un ensemble de règles et de diagrammes respectant des codes.

Le MCD est composé :

* D’entités : représentées par un rectangle, elles symbolisent un élément matériel ou immatériel du réel,
* Des attributs ou champs : représentées par une liste des données de l’entité,
* De clé primaire : c’est un identifiant unique, obligatoire,
* D’associations : représentées par des ovales avec un verbe, elles expliquent comment les entités sont reliées entre elles, avec des traits,
* Des cardinalités : représentées par des couples de petits chiffres ou lettres (0, 1 et/ou N) symbolisant la borne minimale et la borne maximale. La borne minimale (généralement 0 ou 1) décrit le nombre minimum de fois qu’une entité peut participer à une relation, tandis que la borne maximale (généralement 1 ou N) décrit le nombre maximum de fois qu’une entité peut participer à une relation.
* D’enregistrements ou tuples : ce sont les lignes représentant toutes les informations de chaque champ, associées à un seul enregistrement. On ne les représente pas sur un MCD, mais dans la base de données.

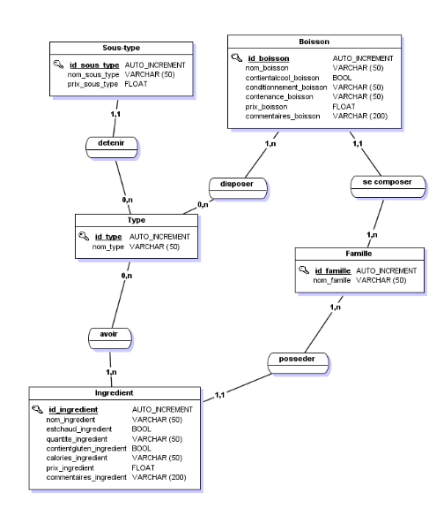


Figure Modèle Conceptuel des Données

*Vous pourrez retrouver ce diagramme en annexe, page 65.*

## Modèle Logique des Données

Le MLD, ou Modèle Logique de Données, est la représentation des données d’un système d’information.

Le MLD utilise un ensemble de règles et de diagrammes codifiés. Il découle du MCD et dispose de caractéristiques similaires.

De ce fait, le MLD est composé :

* De tables : ce sont les « entités »,
* D’attributs/champs,
* De clés primaires,
* De relations et cardinalités : ce sont les « associations »
* De clés étrangères : représentées par les clés primaires qui font référence à une autre table, c’est une contrainte qui garantit l'intégrité référentielle entre deux tables.
* D’enregistrements ou tuples : ce sont les lignes représentant toutes les informations de chaque champ, associées à un seul enregistrement. On ne les représente pas sur un MLD, mais dans la base de données.

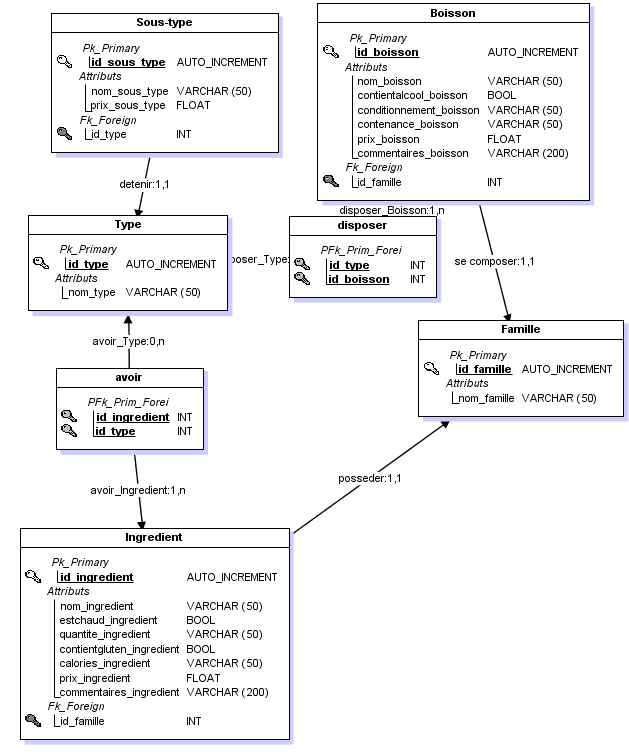


Figure Modèle Conceptuel des Données

*Vous pourrez retrouver ce diagramme en annexe, page 66.*

## Le SQL

Le SQL, ou Structured Query Language, est un langage informatique normalisé permettant de communiquer avec une base de données, à travers des requêtes (simples ou préparées).  
Il permet de lire, insérer, modifier ou supprimer des données dans une base. Mais aussi de modifier la structure de la base de données (ajouter, modifier, supprimer des tables par exemple).

Un fichier .sql a été crée de manière régulière afin de sauvegarder la structure de la base de données.  
Cependant, par souci de clarté et de lisibilité, j’ai préféré utiliser MySQL Workbench pour montrer comment j’ai créé et structuré ma base de données.



Figure Extrait de code de création de base de données

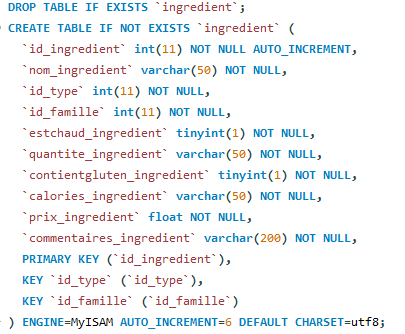


Figure Extrait de code de création d'une table

Ci-dessus se trouvent des extraits de la création de base de données, nommée « iolr » et de la table nommée « ingredient ».

Afin de créer la base de données, j’ai utilisé la commande CREATE DATABASE suivie du nom de celle-ci (le reste est dû à l’import), puis la commande USE suivie du nom de la base pour que le logiciel comprenne qu’il doit la créé puis l’utiliser.

Suite à cela, j’ai créé les tables à l’aide de la commande CREATE TABLE suivie du nom de la table, puis j’ai rajouté les différents champs nécessaires (en marron) ainsi que leurs caractéristiques (en bleu).

J’ai choisi de mettre en clé primaire l’id de la table, en l’occurrence « id\_ingredient ».

Les deux clés étrangères, id\_type (de la table type, faisant référence à sa propre clé primaire id\_type) et id\_famille (de la table famille, faisant référence à sa propre clé primaire id\_famille) n’ont été rajouté qu’après avoir créé les tables correspondantes.

Exemple :



Figure Extrait de code d'ajout de foreign key

*Vous pourrez retrouver tout le code en annexe, pages*

# ARBORESCENCE

L'arborescence est une structure hiérarchisée de données dans des répertoires et des sous-répertoires. L'image de l'arbre est une métaphore pour décrire l'organisation du chemin d'accès par branches et ramifications.

L’objectif est de pouvoir voir la structure d’un site plus facilement et la rendre plus lisible.

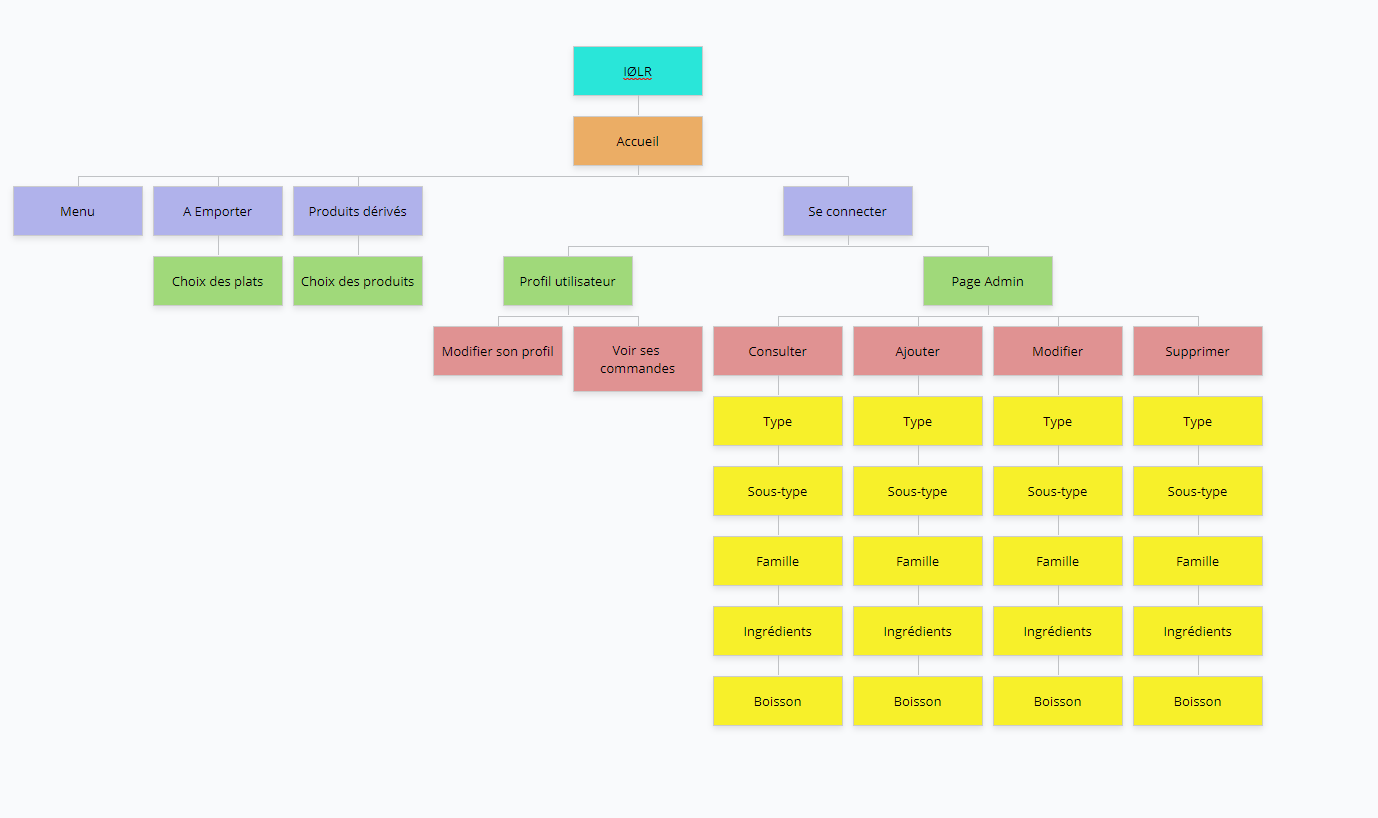


Figure Arborescence du site internet

*Vous pourrez retrouver cette arborescence en annexe, page 67.*

Ici, à partir de l’Accueil, il est possible d’accéder :

* A la page d’Accueil
* A la page des Menus
* A la page A Emporter puis au choix des plats
* A la page des Produits dérivés puis au choix des produits
* A la page Se connecter. Ici deux choix possibles :

1. Accéder au profil en tant qu’utilisateur, puis modifier son profil et voir ses commandes
2. Accéder en tant qu’administrateur, puis pouvoir ainsi accéder à de multiples possibilités comme Consulter, Ajouter, Modifier ou Supprimer. Et selon chaque possibilité, il y a un choix à faire (Type, Sous Type, Famille, Ingrédients, Boissons).

# OUTILS TECHNIQUES UTILISES

En plus des contraintes technologiques utilisées (cf. page 8), j’ai utilisé :

* **Visual Studio Code** : éditeur de code développé par Microsoft pour Windos, Linux et macOS. Il m’a servi pour écrire du code.



Figure Logo Visual Studio Code

* **Balsamiq Wireframes** : logiciel de conception qui permet de créer des maquettes, basé sur du glisser-déposer, facile d’utilisation. Je l’ai utilisé pour faire mon maquettage (cf. pages 21, 22, 23).



Figure Logo Balsamiq Wireframe

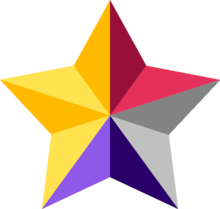
* **Star UML** : logiciel de modélisation UML. Je l’ai utilisé pour faire mon diagramme de cas d’utilisation (cf. page 12), mes diagrammes d’activité (cf. pages 14, 15 et **annexe**), et mes diagrammes de séquence (cf. pages 17, 18).

Figure Logo StarUML

* **JMerise :** logiciel de modélisation des MCD de Merise (méthode d’analyse, conception et gestion de projet informatique). Il permet de générer aussi le MLD. Je l’ai utilisé pour faire mon MCD (cf. page 24 et **annexe**), et mon MLD (cf. page 25 et **annexe**).



Figure Logo JMerise

* [**https://www.gloomaps.com**](https://www.gloomaps.com) **:** site internet spécialisé dans la création d’organigramme. Je l’ai utilisé pour faire l’arborescence (cf. page 31 et annexe page 67).



Figure Logo GlooMaps

* **Google Docs :** traitement de texte en ligne inclus dans la suite Web gratuite Google Docs Editors proposée par Google. Je l’ai utilisé en collaboration avec mon maître de stage (Mr PEYRARD), lors de nos rendez-vous hebdomadaires, pour reporter toutes les informations importantes et les modifications nécessaires.



Figure Logo Google Docs

* [**https://www.github.com**](https://www.github.com)**:** site internet d’hébergement et de gestion de logiciels, utilisant le logiciel de version Git. Je l’ai utilisé pour mettre mes dossiers et mon code. Ainsi, j’ai pu les sauvegarder en ligne. Mon maître de stage pouvait y avoir accès aussi, lui permettant de voir ma progression tout au long du stage.



Figure Logo GitHub

* **GitHub Desktop :** logiciel qui permet d’interagir avec GitHub en utilisant une interface graphique plutôt que de la ligne de commande ou un navigateur web. Je l’ai utilisé pour sauvegarder mon projet et toutes les modifications



Figure Logo GitHub Desktop

* **XAMPP et WAMP :** deuxensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Selon l’ordinateur utilisé, j’ai utilisé XAMPP ou WAMP.

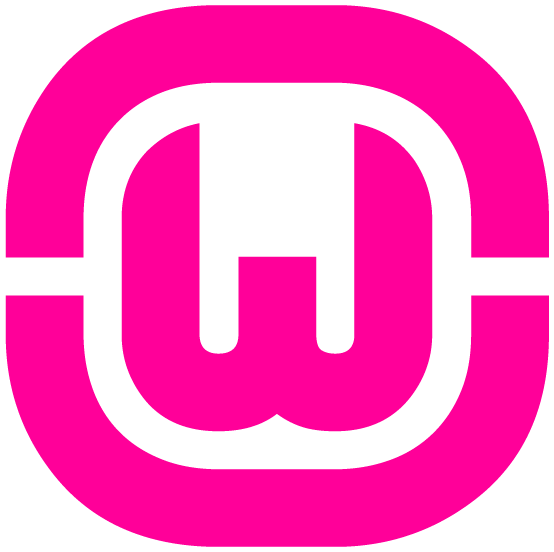


Figure 9 LogoWAMP



Figure 8 Logo XAMPP

# FONCTIONNALITES

## Ajouter un Ingrédient dans la base de données

### La page Web

Cette fonctionnalité a pour objectif l’**Ajout** d’un ingrédient dans la base de données.

En amont, l’administrateur doit avoir créé tous les types, les sous-types et les familles nécessaires pour l’ajout d’un ingrédient.  
En effet, un ingrédient doit avoir un type, un sous-type, une famille pour pouvoir être ajouté à la base de données.

Voici les étapes :

1. Choix de l’action dans le menu **Action**
2. Choix de la catégorie dans le menu **Catégorie**



Figure Page Admin

1. Cliquer sur le bouton **Go**
2. Saisir les informations
3. Cliquer sur le bouton **Ajouter**

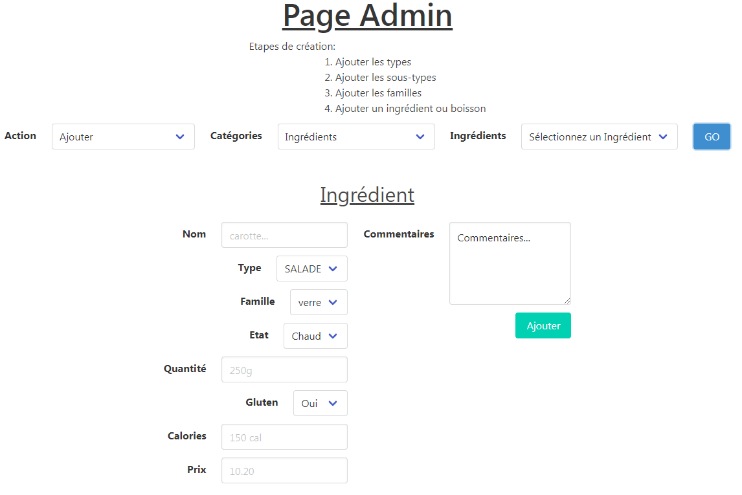


Figure Page Admin Ajout d'ingrédient

*Vous pourrez retrouver ce diagramme en annexe, page 68.*

### Le code

**ATTENTION !**

**Dans cette partie, il n’y aura pas de sécurité. Mr PEYRARD a précisé qu’il voulait s’en occuper personnellement.**

J’ai utilisé la **méthode MVC (Modèle-Vue-Controller).** C’est un style d’architecture logicielle composé de trois modules, ayant chacun sa responsabilité :

* Un modèle (Model) contient les données à afficher.
* Une vue (View) contient la présentation de l'interface graphique.
* Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.

Fonctionnement de la méthode MVC :

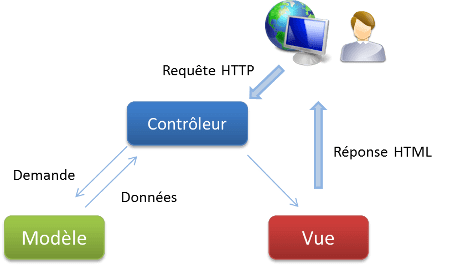


Figure Illustration du fonctionnement du modèle MVC  
Source : internet

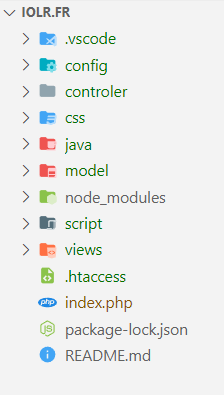


Figure Structure du dossier du code source

Voici la structure du dossier contenant tout mon code.

En utilisant la méthode MVC, je retrouve plusieurs dossiers, chacun ayant son rôle.

Le fichier index.php et ht.access doivent être en dehors d’un dossier spécifique pour fonctionner.

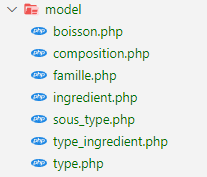


Figure 44 Dossier model et ses fichiers

Dans le dossier model, je retrouve tous les fichiers nécessaires au bon affichage des données.

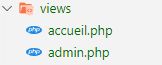


Figure Dossier views et ses fichiers

Dans le dossier views, je retrouve tous les fichiers nécessaires pour l’interface graphique.



Figure Dossier controler et son fichier

Dans le dossier controler, je retrouve mon fichier admin.controler.php contenant la logique.



Figure Dossier config et ses fichiers

Dans le dossier config, je retrouve mon fichier bdd.sql (extraction du code SQL de création de ma BDD) et le fichier connectBdd.php qui contient le code de connexion à la BDD.



Figure Dossier css et son fichier

Dans le dossier css, je retrouve mon fichier style.css, qui me sert uniquement pour une barre de scroll (visible uniquement dans un autre choix dans les menus (ajout d’ingrédient dans un sous-type spécifique).

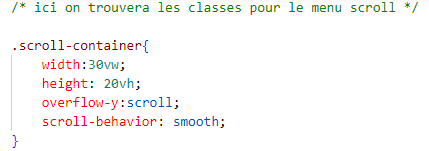


Figure Extrait de code pour la barre de scroll



Figure Dossier script et son fichier

Dans le dossier script, je retrouve le fichier admin.js contenant tout le JavaScript utilisé pour le site.

Dans le fichier admin.php (dossier views), j’ai créé 3 menus déroulants.

Pour les mettre en forme, j’ai utilisé Bulma et celui-ci gère la responsivité naturellement.

Pour cela, j’ai intégré un lien dans le head de mon HTML.



Figure Extrait du code HTML de ma page admin

(j’ai utilisé un peu de Json pour vérifier que mes variables se remplissaient correctement).

Le lien me permet d’importer le fichier CSS directement depuis jsDeliver grâce à un CDN.

* JSDeliver est un CDN public gratuit pour les projets open source. Il peut servir des fichiers Web directement à partir du registre npm et des référentiels GitHub sans aucune configuration.
* Un CDN (content delivery network), ou réseau de diffusion de contenu en français, est constitué d’ordinateurs reliés en réseau à travers Internet et qui coopèrent afin de mettre à disposition du contenu ou des données à des utilisateurs.

J’ai modifié le style des menus (par rapport aux menus classiques en HTML) grâce aux différentes classes inhérentes de Bulma, comme les classes control, et select. Cette dernière est spécifique pour les menus déroulants.

Chaque menu a sa propre mission :

* Le premier permet de choisir l’**Action**.
* Le deuxième permet de choisir la **Catégorie**.
* Et le troisième dépend totalement de la Catégorie utilisée. Par exemple, si la catégorie sélectionnée est Ingrédient, ce sera le menu **Ingrédients** qui sera affiché. Mais si la catégorie sélectionnée est Boisson, ce sera le menu **Boissons** qui sera affiché.



Figure Extrait de code pour le menu Ingrédient

*Vous pourrez retrouver ce bout de code en annexe, page 70, ainsi que les autres menus en annexe, pages 69.*

Dans le fichier index.php (situé dans dossier du code source), j’ai choisi de faire un routing pour gérer mes pages et mes chemins.

Le **routing** est le mécanisme par lequel des chemins sont sélectionnés dans un réseau pour acheminer les données d’un expéditeur jusqu’à un ou plusieurs destinataires.  
En amont, j’ai utilisé un fichier de configuration **.htaccess** me permettant les redirections d’URL.

Pour gérer mes chemins, j’ai choisi d’utiliser un **switch case** (l'instruction switch évalue une expression et, selon le résultat obtenu et le cas associé, exécute les instructions correspondantes) dans un **try …catch** (l'instruction try...catch regroupe des instructions à exécuter et définit une réponse si l'une de ces instructions provoque une exception).  
En parallèle, j’ai défini mon URL grâce à la fonction native **define()** que l’on peut voir tout en haut du code ci-dessous.



Figure Code du fichier index.php

*Vous pourrez retrouver ce bout de code en annexe, page 71-72.*

En tant que préparation, dans un fichier JavaScript (dossier script), j’ai effectué plusieurs pointeurs (chacun ayant sa propre variable) menant vers différents éléments de ma page HTML.  
Chaque pointeur sera réutilisé plus tard dans d’autres fonctions.

Ceux qui nous intéressent sont les suivants :



Figure Extrait de code du fichier admin.js

* La variable divMenuIngredient correspond à la <div> du 3ème menu, dépendant de la Catégorie choisie. En l’occurrence la catégorie choisie est Ingrédient, et par conséquent le menu correspondant est celui des Ingrédients (cf. annexe page 73).
* La variable divFormIngredient correspond à la <div> principale de la section Ingrédient qui apparaît après le clic (cf. annexe pages 74-75-76).
* La variable selectIngredient correspond au 3ème menu de sélection Ingrédients (cf. annexe page 77).
* La variable formIngredient correspond au formulaire de la section Ingrédient qui apparaît après le clic (cf. annexe pages 74-75-76).
* La variable buttonIngredient correspond au bouton **Ajouter** dans la section Ingrédient (cf. annexe page 77).
* La variable buttonRefresh correspond au bouton **GO** après les 3 menus déroulants

(cf. annexe page 78).

#### Action de l’administrateur et les conséquences engendrées

Après avoir sélectionné ses choix dans les menus déroulants, l’administrateur clique sur le bouton **GO**.

Avec cette action sur le bouton, l’attribut **onclick** s’active et fait appel à la fonction **bouttonRefreshOnClick()** (qui se trouve dans mon dossier Script, dans le fichier admin.js).

Figure Code du bouton

Dans cette situation, l’administrateur a choisi dans le menu **Action** l’option **Ajouter :**

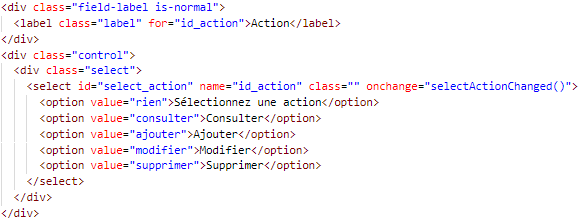


Figure Code du menu Action

Ici, seule la valeur de la sélection dans le menu **Action** est importante. Elle permet de rentrer dans la condition selon laquelle la valeur est égale à ajouter.

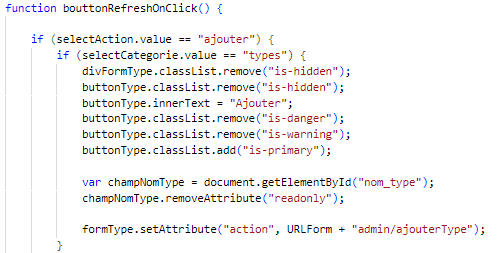


Figure Code de la fonction buttonRefreshOnClick()

Puis, l’administrateur a choisi la Catégorie des **Ingrédients :**

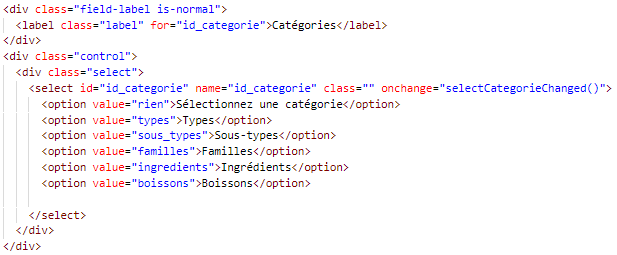


Figure Code du menu Catégorie

Ici, la valeur de la sélection dans le menu **Catégorie** est importante. Elle permet de rentrer dans la condition selon laquelle la valeur est égale à ingrédients. 

Figure Suite du code de la fonction buttonRefreshOnClick

Parmi les instructions dans la fonction **bouttonRefreshOnClick(),** à la fin, il y a une fonction qui modifie la variable formIngredient : c’est le setAttribute. C’est une fonction native qui permet d’ajouter un nouvel attribut ou de changer la valeur d'un attribut existant pour l'élément spécifié. Si l'attribut existe déjà, la valeur est mise à jour ; sinon, un nouvel attribut est ajouté avec le nom et la valeur spécifiés.  
Ici, c’est l’URL qui est modifiée.

Ainsi, l’administrateur arrive sur une autre page (admin/ajouterIngredient), grâce au routing de l’index. Comme l’URL est modifiée, le routing l’emmène dans un autre cas :

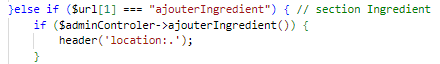


Figure Extrait du code du fichier index.php

Ici, comme l’URL est bien égale à « ajouterIngredient », je rentre dans la condition.

Ensuite, on applique la fonction ajouterIngredient() se trouvant dans le fichier admin.controler.php (dossier controler) à la variable $adminControler :



Figure Code de la fonction ajouterIngredient

Dans cette fonction **ajouterIngredient,** il y a plusieurs étapes dans les instructions :

1. Etape 1 :

Le **if** (structure de condition basique) me permet de vérifier si la variable globale **$\_POST** contient la moindre information.



Figure Extrait de code

* Si la condition est KO – je vais directement à la fin des instructions, où se trouve la fonction native **unset()** qui nettoiera le **$\_POST** (sans effet car déjà vide) puis je quitte la fonction avec le résultat "false" (return false).

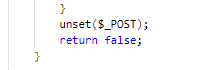


Figure Extrait de code

1. Etape 2 :

* Si la condition est OK – je croise un nouveau "**if**" dont le but est de vérifier si toutes les informations nécessaires sont fournies, grâce à **isset** (qui détermine si une variable est définie, et différente de null).

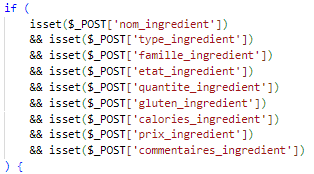


Figure Extrait de code

* Si la condition est KO – je me rends à la fin des instructions, où se trouve le **unset()** qui nettoie le **$\_POST** (non vide cette fois si) puis je quitte la fonction avec le résultat false.

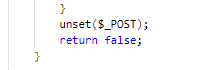


Figure Extrait de code

1. Etape 3 :

* Si la condition OK – je récupère dans des variables temporaires les informations contenues dans la variable globale **$\_POST**.

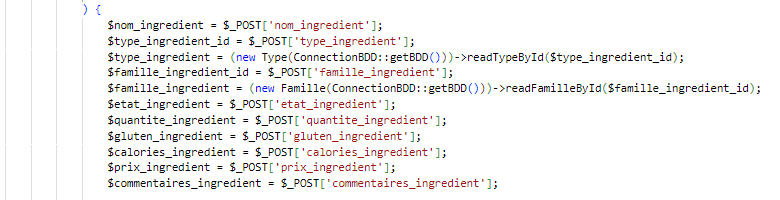


Figure Extrait de code

* Après avoir récupéré les données intéressantes, je nettoie le **$\_POST** car il ne m’est plus d'aucune utilité.



Figure Extrait de code

1. Etape 4 :

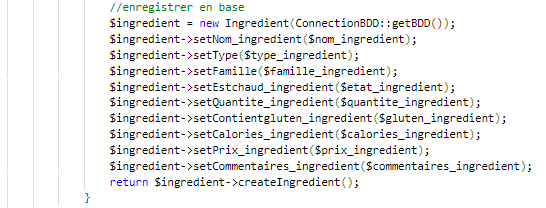
* Je crée en mémoire un objet de type Ingrédient à l'aide des infos précédemment stockées, pour préparer à l’enregistrement en base de données. 

Figure Extrait de code

* J’exécute la fonction de création en base de données **createIngredient()** afin que l'ingrédient soit créé en base de donnée et rangé dans la table correspondante.
* ConnexionBDD :: getBDD() : ici j’effectue la connexion à la BDD. Pour cela, je fais appel à la classe ConnexionBDD (qui se trouve dans le fichier connectBdd.php, dans le fichier config).

Puis je fais appel à la fonction statique getBDD. Elle est en **static,** car je veux qu’elle soit appelable sans devoir instancier la classe ConnectionBDD. Je peux donc y accéder statiquement seulement si je fais une instance d’objet, d’où le **ConnexionBDD ::**

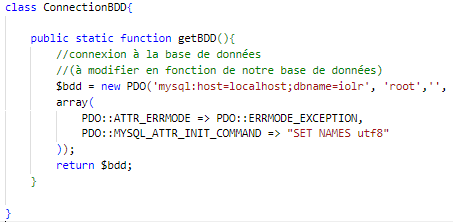


Figure Code de la classe ConnectionBDD

Dans cette fonction **createIngredient()** (dans le dossier model, dans le fichier ingredient.php), il y a plusieurs étapes dans les instructions :



Figure Code de la fonction createIngredient

* 1. Etape 1
* J’utilise un **try … catch**.
  1. Etape 2
* Dans le **try**, j’effectue ma **requête préparée SQL**, en faisant un **INSERT INTO**, puis j’exécute la requête SQL.
  1. Etape 3 :
* Je vérifie si la requête s’est bien déroulée avec un **if,** qui me retournera true si c’est le cas, et par conséquent si l’ajout a bien été effectué.
  1. Etape 4
* Dans le **catch**, s’il y a une erreur, il m’affiche un message d’erreur.

De retour au routing dans index.php, comme la condition a été réalisée, la fonction native **header()** est appliquée. Elle permet de ramener l’administrateur à la page actuelle de départ.

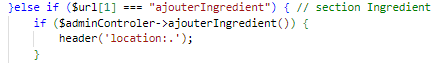


Figure Extrait du code du fichier index.php

## Supprimer un Ingrédient dans la base de données

Cette fonctionnalité a une structure similaire à la fonctionnalité **Ajouter.**

### La page Web

Cette fonctionnalité a pour objectif de supprimer un ingrédient dans la base de données.

L’ingrédient à supprimer doit avoir été enregistré en base de données au préalable, pour pouvoir être supprimé.

De ce fait, l’administrateur doit avoir utilisé la fonctionnalité **Ajouter** pour pouvoir utiliser la fonctionnalité **Supprimer**.

Voici les étapes :

1. Choix de l’action dans le menu **Action**
2. Choix de la catégorie dans le menu **Catégorie**
3. Choix de l’ingrédient dans le menu **Ingrédients**



Figure Page Admin

1. Cliquer sur le bouton **Go**
2. Vérifier que c’est le bon ingrédient à supprimer
3. Cliquer sur le bouton **Supprimer**

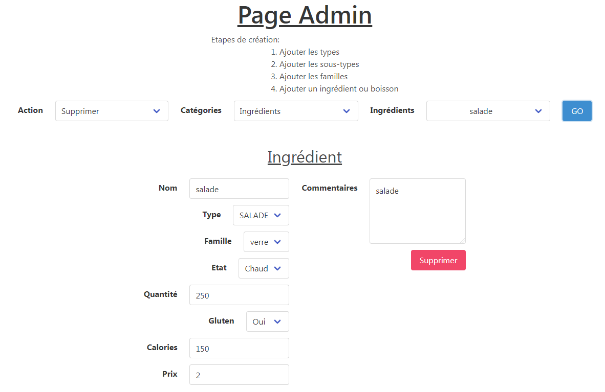


Figure Page Admin Suppression d'ingrédient

*Vous pourrez retrouver ce diagramme en annexe, page 79.*

### Le code

**ATTENTION !**

**Dans cette partie, il n’y aura pas de sécurité. Mr PEYRARD a précisé qu’il voulait s’en occuper personnellement.**

J’ai utilisé la **méthode MVC (Modèle-Vue-Controler).**

La préparation en amont de cette fonctionnalité est la même que celle pour la fonctionnalité d’**Ajout.**

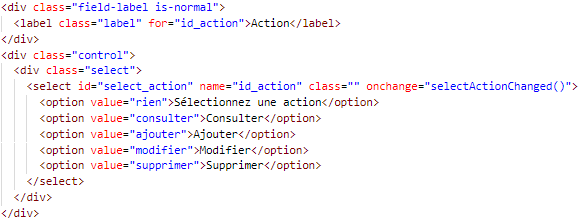
En effet, toute la préparation (lien CDN, menus, routing et pointeurs JS) est identique.

Seules les actions de l’administrateur changent et engendrent des conséquences différentes.

#### Action de l’administrateur et les conséquences engendrées :

Après avoir sélectionné ses choix dans les menus déroulants, l’administrateur clique sur le bouton **GO**.

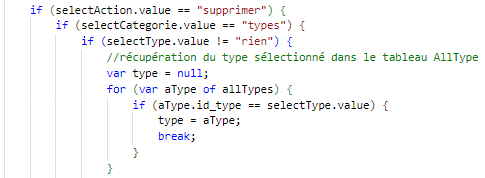
Avec cette action sur le bouton, l’attribut **onclick** s’active et fait appel à la fonction **bouttonRefreshOnClick()** (qui se trouve dans mon dossier Script, dans le fichier admin.js).

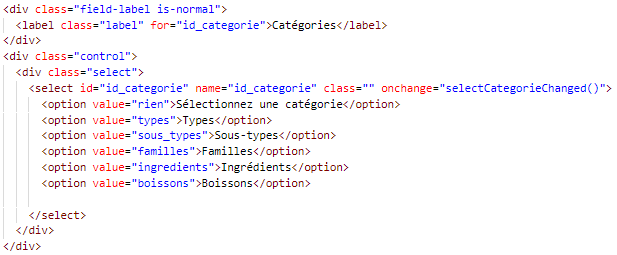
Dans cette situation, l’administrateur a choisi dans le menu **Action** l’option **Supprimer :**

Ici, seule la valeur de la sélection dans le menu **Action** est importante. Elle permet de rentrer dans la condition selon laquelle la valeur est égale à supprimer.



// code non nécessaire à la compréhension



Puis, l’administrateur a choisi la Catégorie des **Ingrédients :**

Ici, la valeur de la sélection dans le menu **Catégorie** est importante. Elle permet de rentrer dans la condition selon laquelle la valeur est égale à ingrédients.



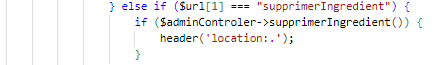
Parmi les instructions dans la fonction **bouttonRefreshOnClick(),** à la fin, il y a une fonction qui modifie la variable formIngredient : c’est le setAttribute.



Ici, c’est l’URL qui est modifiée.

Ainsi, l’administrateur arrive sur une autre page (**admin/supprimerIngredient**), grâce au routing de l’index.

Comme l’URL est modifiée, le routing nous emmène dans un autre cas :



Ici, comme l’URL est bien égale à « supprimerIngredient », on rentre dans la condition.

Ensuite on applique la fonction supprimerIngredient() (dans le dossier controler, dans le fichier admin.controler.php) à la variable $adminControler :



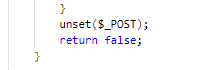
Dans cette fonction **supprimerIngredient,** il y a plusieurs étapes dans les instructions :

1. Etape 1 :

Le **if** (structure de condition basique) me permet de vérifier si la variable globale **$\_POST** contient la moindre information.



* Si la condition est KO – je vais directement à la fin des instructions, où se trouve la fonction native **unset()** qui nettoiera le **$\_POST** (sans effet car déjà vide) puis je quitte la fonction avec le résultat "false" (return false).

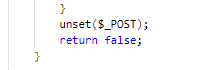


1. Etape 2 :

* Si la condition est OK – je fais un nouveau "**if**" dont le but est de vérifier si toutes les informations nécessaires sont fournies, grâce à **isset.**



* Si la condition est KO – je me rends à la fin des instructions, où se trouve le **unset()** qui nettoie le **$\_POST** (non vide cette fois si) puis je quitte la fonction avec le résultat false.



1. Etape 3 :

* Si la condition OK – je récupère dans des variables temporaires les informations contenues dans la variable globale **$\_POST**.



* Après avoir récupéré les données intéressantes, je nettoie le **$\_POST** car il ne m’est plus d'aucune utilité.



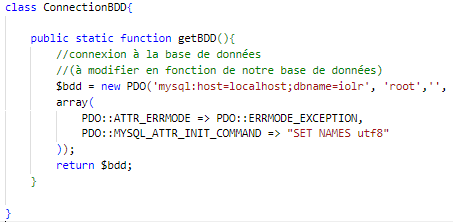
1. Etape 4 :

* Je crée en mémoire RAM un objet de type Ingrédient à l'aide des infos précédemment stockées, pour préparer à l’enregistrement en base de données.
* J’exécute la fonction de suppression en base de données **deleteIngredient()** afin que l'ingrédient soit créé en base de donnée et rangé dans la table correspondante.

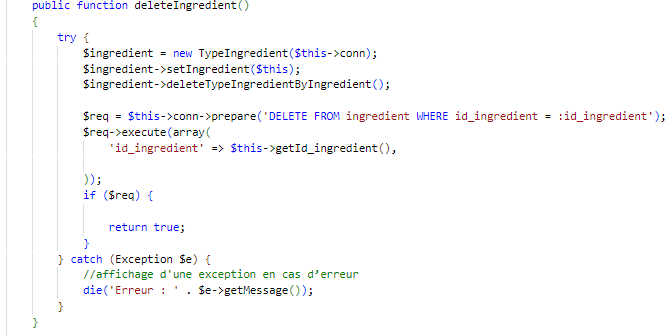


* Puis, nous créons en mémoire RAM un objet de type Ingrédient à l'aide des infos précédemment stockées, pour préparer à l’enregistrement en base de données.
* ConnexionBDD :: getBDD() : ici j’effectue la connexion à la BDD. Pour cela, je fais appel à la classe ConnexionBDD (qui se trouve dans le fichier connectBdd.php, dans le fichier config).

Puis je fais appel à la fonction statique getBDD. Elle est en **static,** car je veux qu’elle soit appelable sans devoir instancier la classe ConnectionBDD. Je peux donc y accéder statiquement seulement si je fais une instance d’objet, d’où le **ConnexionBDD ::**



Dans cette fonction **deleteIngredient()** (dans le dossier model, dans le fichier ingredient.php), il y a plusieurs étapes :



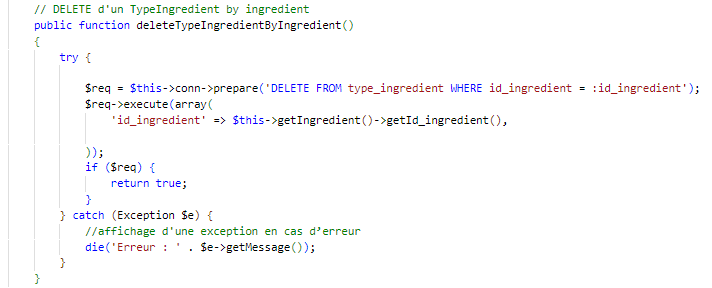
1. Etape 1 :

* J’utilise un **try … catch**.

1. Etape 2 :

* Dans le **try**, je crée un nouvel objet **TypeIngredient()** (classe située dans le fichier type\_ingredient, dans le dossier model) que je mets dans une variable, j’effectue les modifications de cette variable avec le **setter** et je lui applique la fonction **deleteTypeIngredientByIngredient()**.

*Précision : cette étape est présente car elle devait servir pour une version ultérieure déjà prévue. En effet, certains ingrédients pouvaient faire partie d’une liste prévue pour être reliée à un type selon le sous-type. Cette classe, type\_ingredient, a été réalisée mais utilisée partiellement. Dans le cas actuel, la notion a été intégrée pour éviter de modifier les fonctions plus tard.*



Dans cette fonction **deleteTypeIngredientByIngredient()** (dans le dossier model, dans le fichier type\_ingredient.php), il y a plusieurs étapes :

* J’utilise un **try…catch**.
* Dans le **try**, j’effectue une **requête préparée** de type **DELETE FROM**, et j’exécute cette requête.
* Je vérifie qu’elle a été effectuée avec le **if.** Si la condition est réalisée, elle retourne true.
* Dans le **catch**, s’il y a une erreur, il m’affiche un message d’erreur.

1. Etape 3 :

* J’effectue une **requête préparée** de type **DELETE FROM** et je l’exécute.

1. Etape4 :

* Avec un **if**, je vérifie si la requête a bien été effectuée. Si la condition est réalisée, elle retourne true.

1. Etape 5 :

* Dans le **catch**, s’il y a une erreur, il m’affiche un message d’erreur.

De retour au routing dans index.php, comme la condition a été réalisée, la fonction native **header()** est appliquée. Elle permet de ramener l’administrateur à la page actuelle de départ.



SECTION ENGLISH

# CONCLUSION

Depuis la fin du stage, après avoir discuté avec le client, il a été convenu d’une évolution du site.

De ce fait, le MCD et MLD ne correspondaient plus. Vous trouverez les nouveaux en annexe, pages………………

Ce stage a été un défi sur plusieurs aspects : sur les conditions de travail, sur la recherche d’informations, sur le fait de se former soi-même et sur mes envies pour le futur.

Pour les conditions de travail, j’ai été confronté au monde du travail pour la première fois en tant que développeur. J’ai eu affaire à un maître de stage qui faisait office de supérieur hiérarchique, à qui je devais des comptes. Il m’a poussé dans ma réflexion et à aller plus loin dans ma façon de construire mon code.

Le télétravail a été compliqué à vivre au quotidien car mon domicile faisait office bureau. Il n’y avait pas de distinction claire entre mon lieu de vie et mon lieu de travail. J’ai eu l’impression d’avoir été enfermé pendant toute la durée du stage.

J’ai appris que travailler seul n’était pas une situation qui me convenait. Même si j’apprécie de travailler dans le calme, dans mon coin, j’aime bien avoir du monde autour de moi pour partager des points de vue et des solutions que chacun peut apporter. Cela permet d’ouvrir ses horizons, de ne pas rester étriqué dans une seule direction de recherche par exemple.

D’un point de vue de la recherche d’informations, j’ai appris qu’il fallait diversifier ses sources. Aller voir ailleurs permet d’obtenir d’autres options, et d’autres informations qui peuvent être plus pertinentes, voire plus parlant pour nous.

En ce qui concerne le fait de se former soi-même, j’ai beaucoup apprécié. En effet, j’ai découvert des technologies qui m’étaient inconnues, comme Bulma. Sa syntaxe particulière (proche de Bootstrap dans son approche) et son utilisation pour l’esthétique d’un site internet facilitent grandement, quand on sait bien l’utiliser. En pleine expansion, Bulma pourrait être un futur grand, aux côtés de Bootstrap. De plus, ça m’a donné envie de me tourner vers d’autres technologies, d’en découvrir de nouvelles.

Enfin, pour mon avenir, il est clair que le travail en équipe est primordial pour moi, que ce soit d’un point de vue social, et même psychologique. Mes expériences professionnelles précédentes, ainsi que celle-ci ont confirmé mon envie de travailler dans une entreprise à taille humaine, au contact de mes collègues. Cependant, si je travaille pour une grande entreprise, j’aimerai intégrer une équipe de développeur afin de garder cette proximité sociale.

J’ai envie d’aller plus loin dans mes compétences, mes connaissances et mes études. Pour cela, à la fin de la formation actuelle, j’ai l’objectif d’intégrer le CESI (Ecole Supérieure de l’Alternance), pour faire un Bachelor Concepteur Développeur d’Applications en logiciel. J’ai choisi de me spécialiser dans le logiciel, plutôt que de continuer dans le Web. La formation débutera en octobre 2022.

# ANNEXES

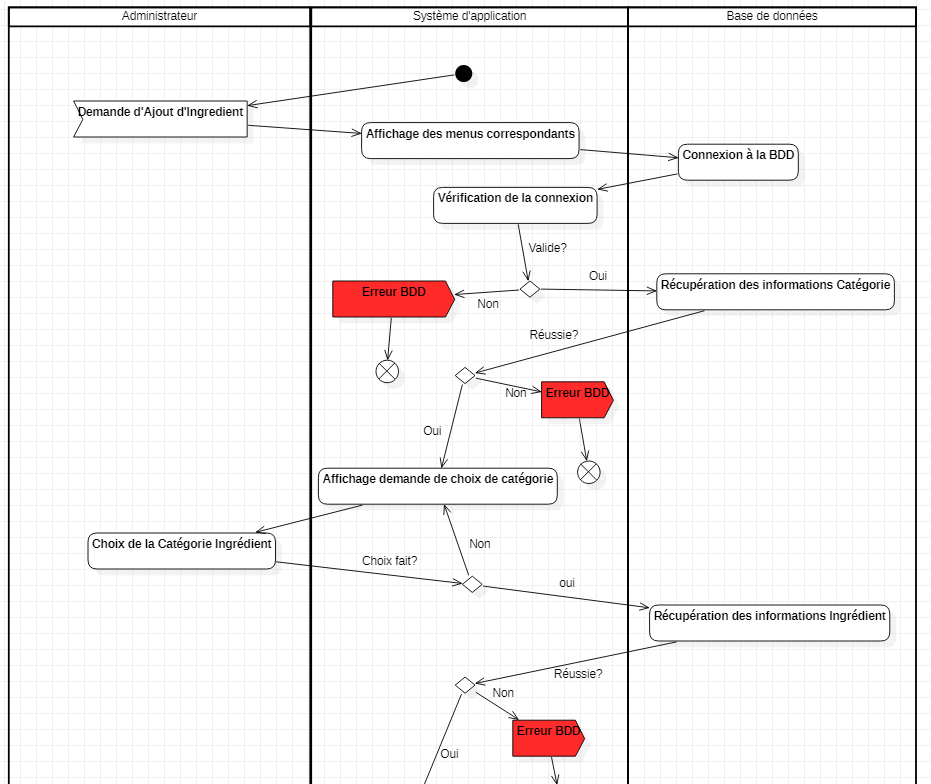
TABLE DES MATIERES

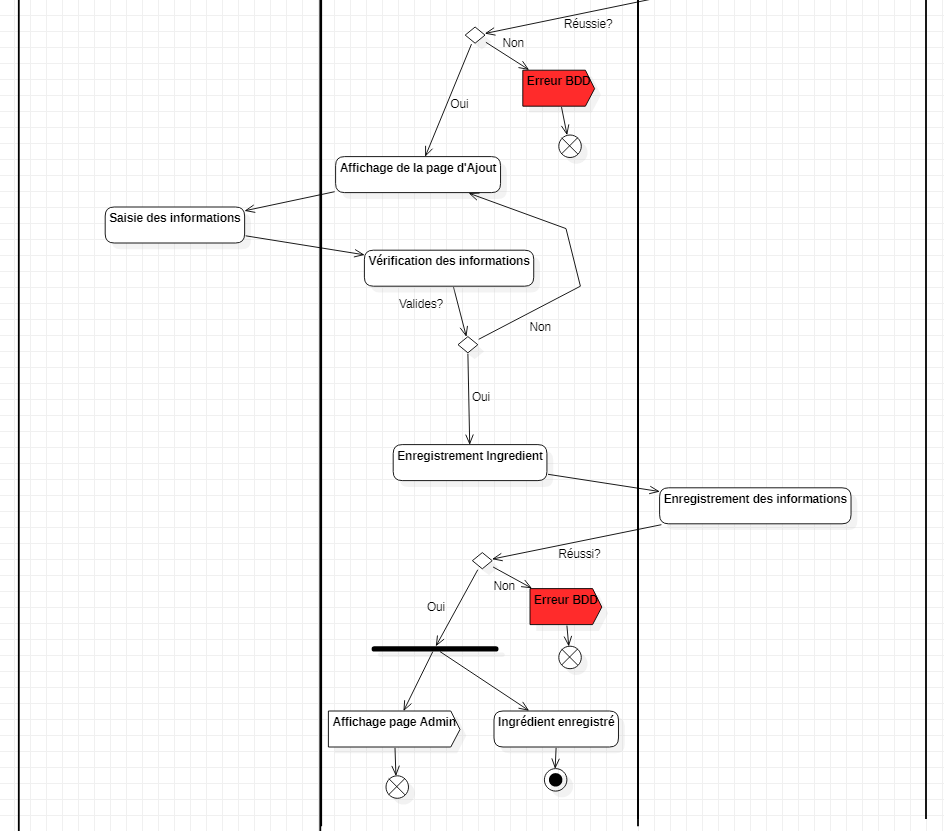
##### Compétences attendues :

*Source : REAC*

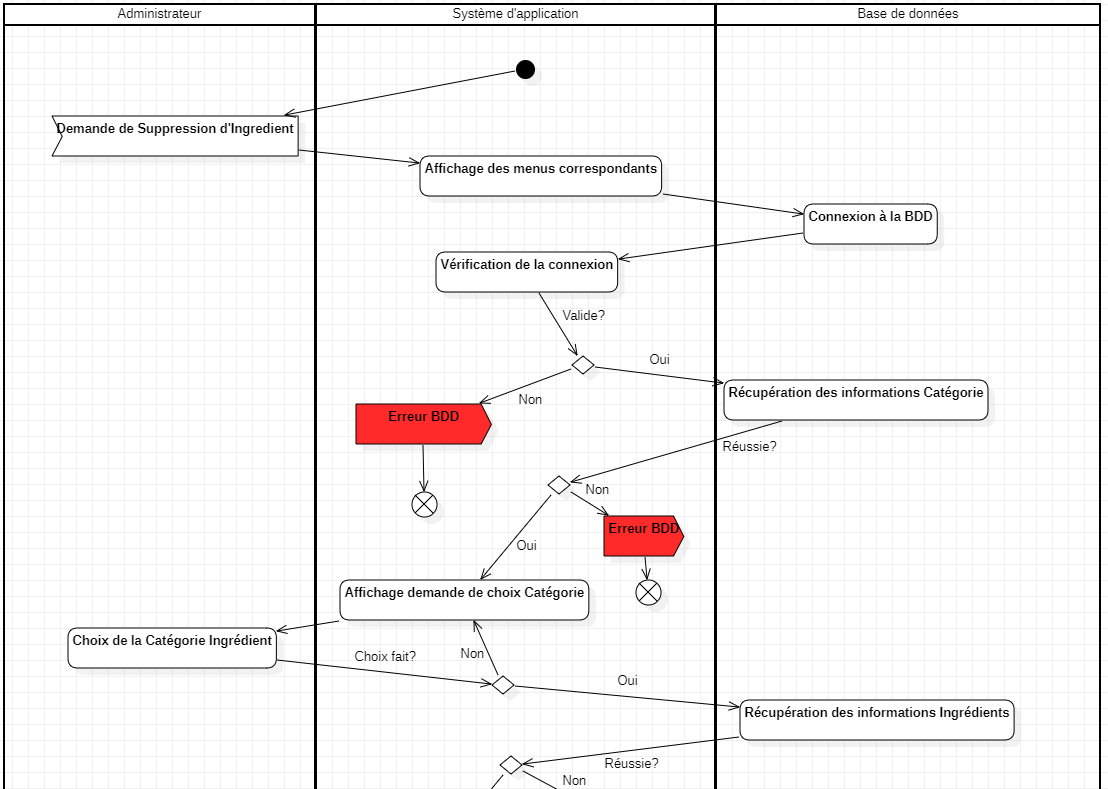
##### Use Case :

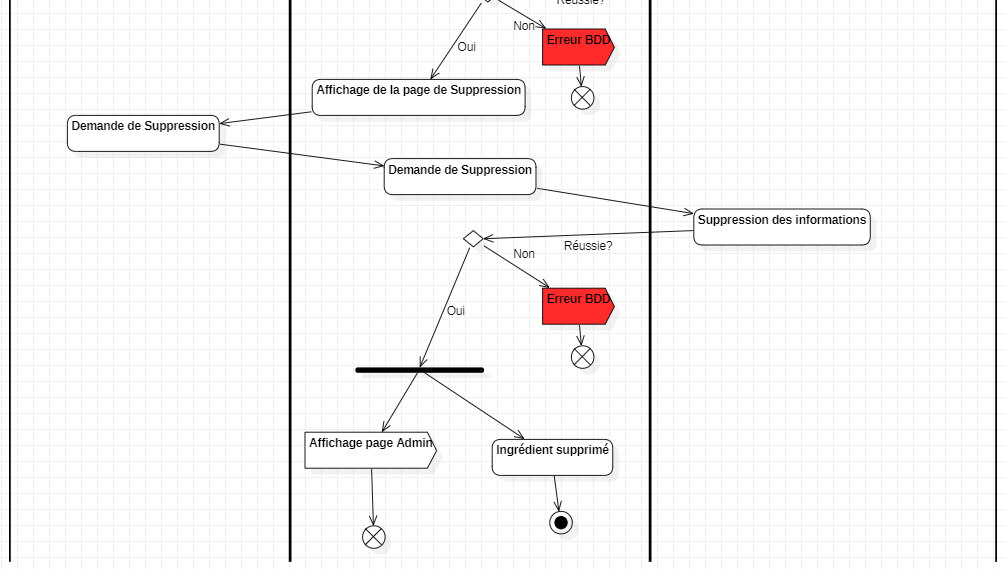
##### Diagramme d’activité Ajout d’ingrédient :



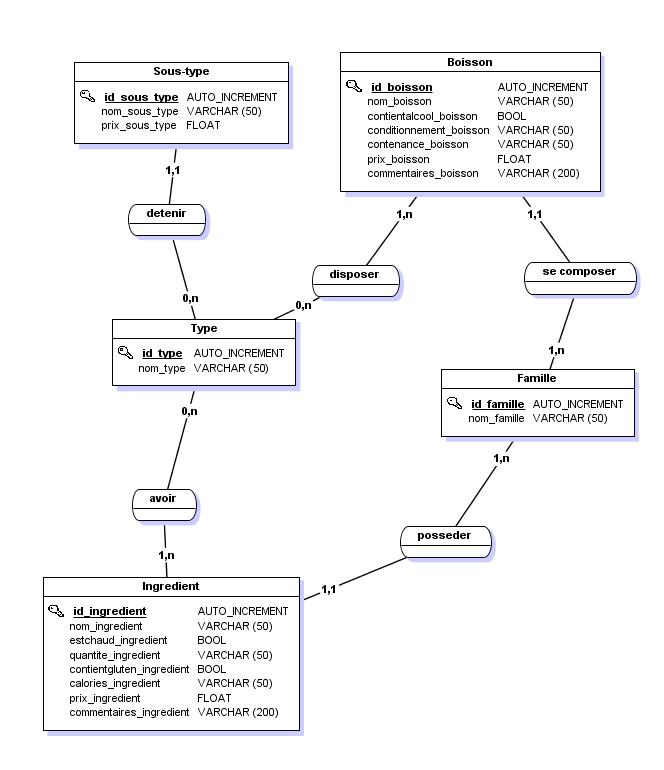


##### Diagramme d’activité Suppression d’ingrédient :

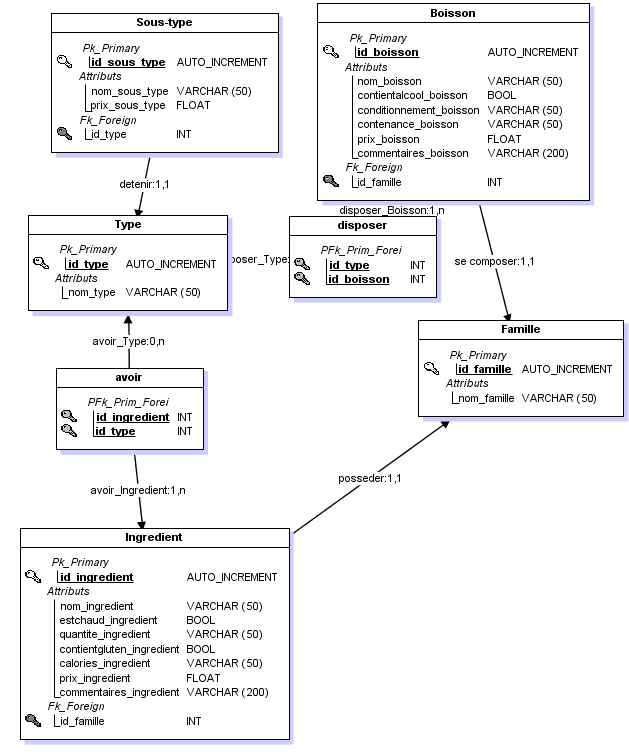




##### Modèle Conceptuel des Données :

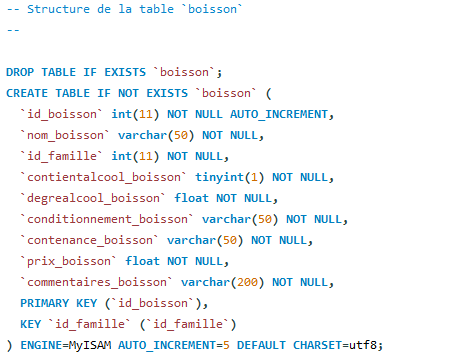


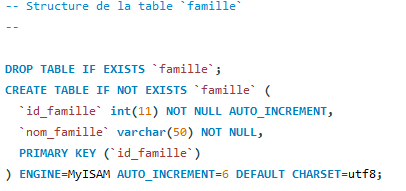
##### Modèle Logique des Données :

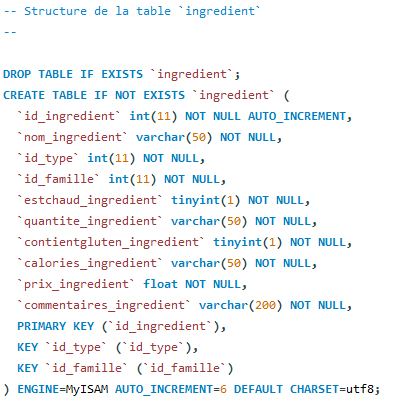


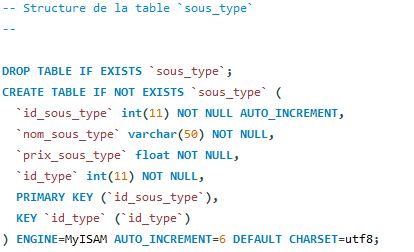
##### Code pour la création de la base de données :

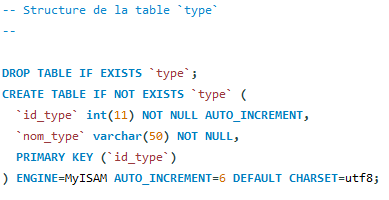


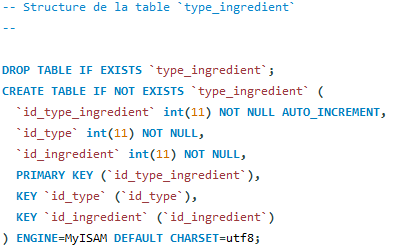


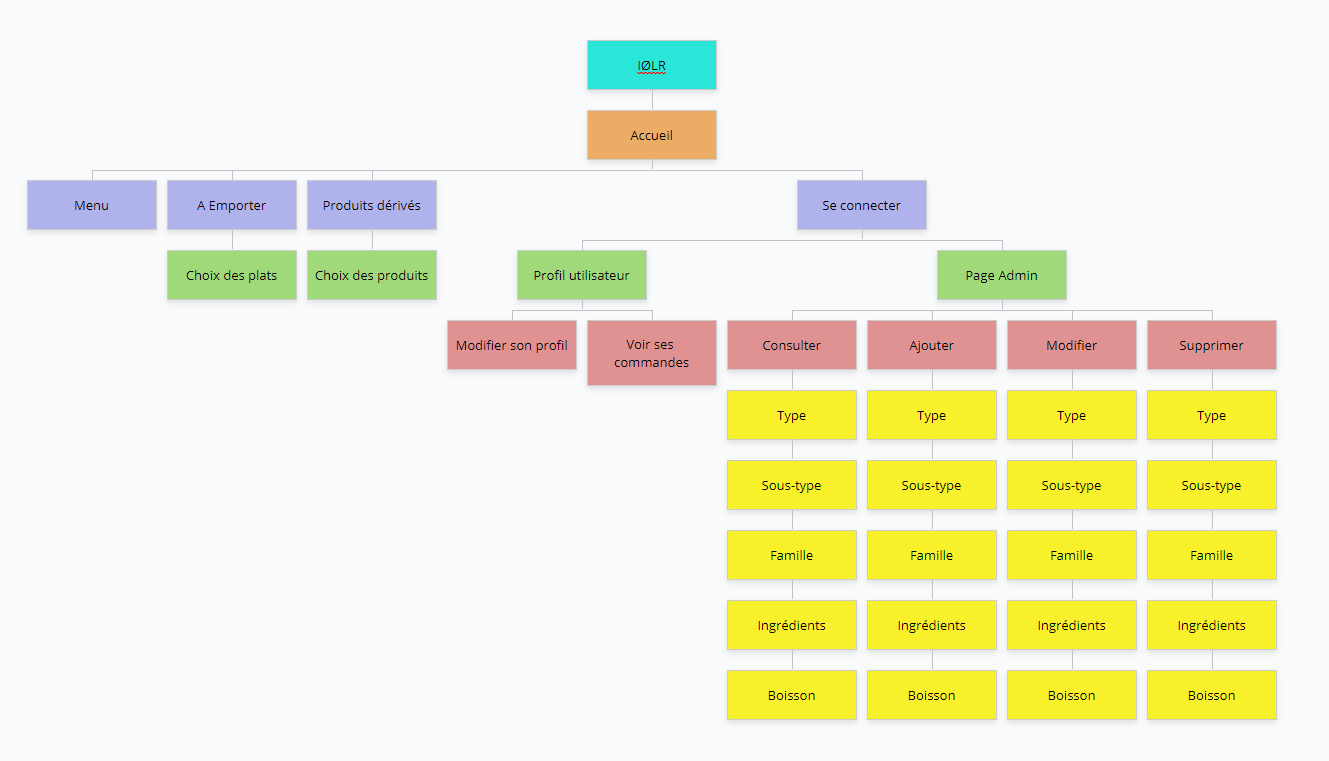




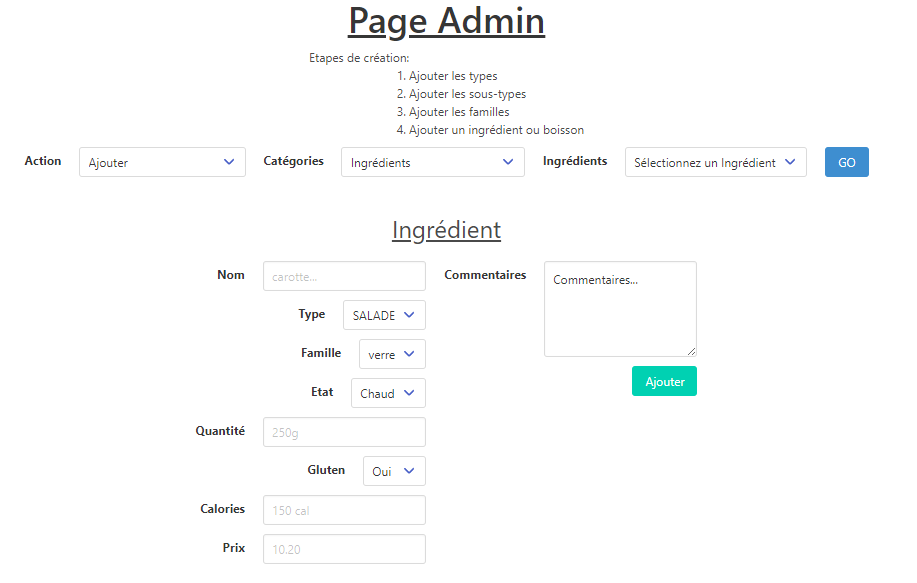






Arborescence :

Côté client, Fonctionnalité **Ajouter** un ingrédient, page admin :



Code pour le menu **Action** :



Code pour le menu **Catégories** :

Code pour le menu **Ingrédients** :



Code de la page index.php





Code pour la divMenuIngedient :



 Code pour la divFormIngredient et pour la div formIngredient:

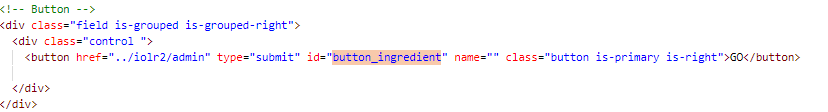




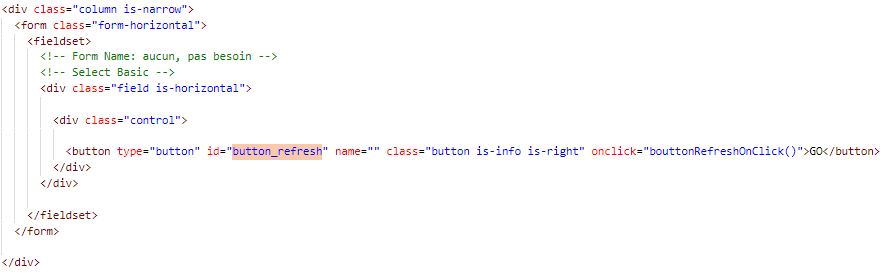
Code pour la div selectIngredient :



Code pour la div buttonIngredient :



Code pour la div buttonRefresh :



Côté client, Fonctionnalité **Supprimer** un ingrédient, page admin :

