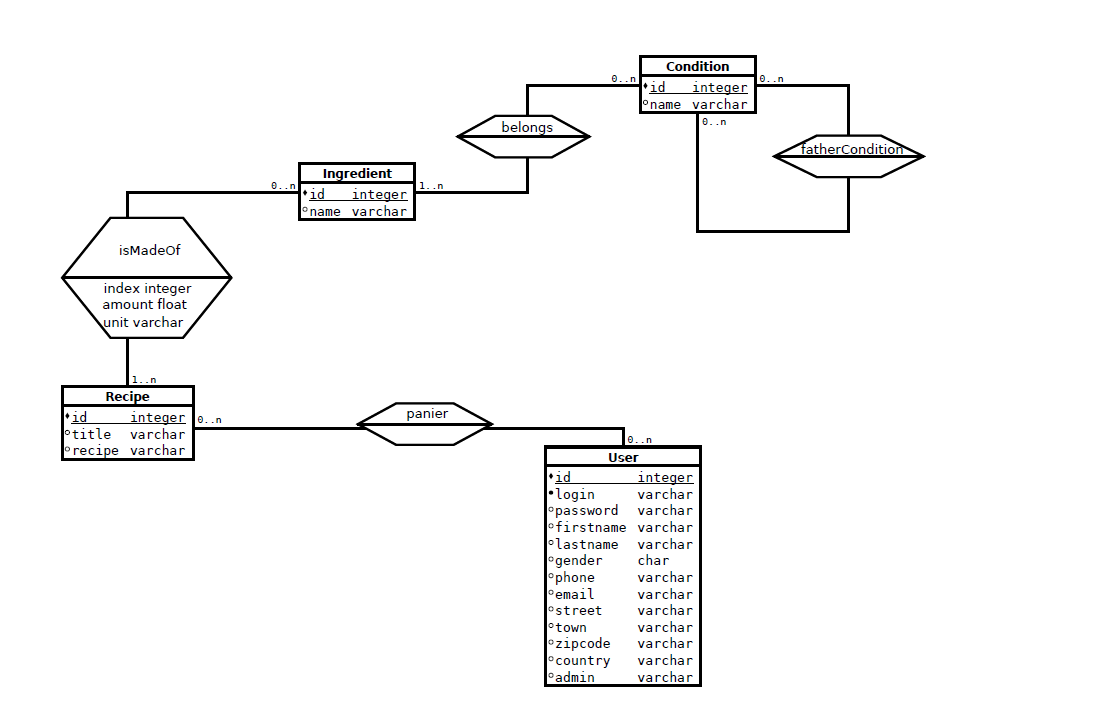
Rapport Projet Web

# Analyse

## Stockage des données et organisation

Pour ce projet, il nous a été fourni un dossier contenant les photos des cocktails ainsi que le fichier contenant toutes les données relatives aux différentes recettes. Pour la gestion des données se présentait donc deux possibilités : les récupérer comme nous l’avions fait en TD et TP, c’est-à-dire par la manipulation simple de fichiers, ou bien réunir nos connaissances et passer par une base de données. Notre choix s’est porté vers la base de données (via EasyPHP et MAMP) car nous possédons des acquis à ce sujet.

Voici le schéma Entité-Association (EA) de notre base de données :



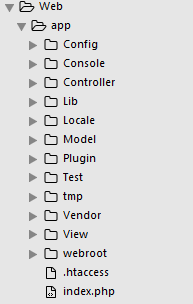
Nous avons donc mis en place les systèmes de gestion d’utilisateurs (connexion et gestion de compte) et du panier de recettes favorites, qui seront décrits dans des parties dédiées.

## Choix de développement

Comme dit précédemment, nous avons opté pour la manipulation de données via une base de données, cette dernière qui est créée à partir du fichier de données fourni.

Concernant le site en lui-même il a été développé via le framework Web CakePHP, framework dont nous connaissions les spécificités car nous l’avions déjà utilisé pour d’antérieurs projets.

Vous trouverez ci-après une image représentant la hiérarchie du projet. Seules les rubriques Controller, Model et View sont le plus souvent sollicitées, car c’est à cet endroit que toute l’application se concentre.



## Gestion des formulaires

Les formulaires ont été implémentés de façon très simpliste et avec les méthodes de mise en page vues en TD telles que des balises <legend> et des balises <label> pour chaque champs. Chaque champ est soit vérifié directement par le navigateur via des attributs « required » ou bien par une expression régulière insérée dans l’attribut « pattern ».

Une vérification après la saisie est tout de même effectuée par CakePHP car ce framework, étant basé sur le modèle Model View Controller (MVC), permet de définir des modèles de validation (caractères minimum requis, ne peut pas être vide, etc…) qui assureront une cohérence des données même si le navigateur ne le fait pas correctement.

Le modèle MVC consiste simplement à passer des données par un Model dit modèle de validation, qui va ensuite aller dans le Controller pour sauvegarder ou autre manipulation et enfin afficher le résultat dans View.

## Gestion du panier de recettes

Concernant le panier de recettes, et pour respecter au mieux les spécifications déclarées dans le sujet, il en découle deux gestions :

* Il est initialement vide lorsqu’un nouvel utilisateur non inscrit ou qu’un utilisateur inscrit mais non connecté arrive sur le site, puis selon les choix du visiteur, le panier se remplit ou se voit retiré des recettes. Les recettes ne sont pas stockées durablement, lorsque la session va expirer, le panier sera détruit avec la session, le visiteur devra alors refaire son panier lors d’une prochaine visite. Nous avons donc décidés de passer par ces variables de session comme nous l’avons vu en TD et en TP afin que les données ne soient stockées que pour la session active de l’utilisateur.
* Lorsqu’un utilisateur, qui est inscrit, arrive sur le site et se connecte, son panier est stocké en base de données, il dispose des mêmes opérations que citées précédemment (ajout et suppression de recettes). Cependant, comme ce panier est stocké en base de données, lorsqu’il se connectera ultérieurement, il disposera à nouveau de son panier contenant toutes les recettes déjà sélectionnées auparavant.

## Gestion des utilisateurs

Cette gestion est assez simpliste, un utilisateur lambda arrive sur le site, peut choisir de s’inscrire ou non, s’il opte pour l’inscription, il devra alors remplir un formulaire contenant des champs de données utiles telles qu’un login et un mot de passe qui sont obligatoires. Plus tard, lorsqu’il sera connecté, il pourra bien entendu compléter son profil avec des données plus personnelles (nom, prénom, …).

Pour une meilleure gestion et sécurité de l’application, nous avons implanté un compte administrateur dans la base de données lors de sa création qui va permettre de gérer les utilisateurs en cas de soucis et aussi de visualiser toutes les opérations effectuées sur le site. Cette administration sera décrite plus tard dans ce rapport.

Concernant les autres utilisateurs, ceux dits « visiteurs », ils pourront accéder à leur profil et en modifier les informations.

# Administration de l’application - Mode d’emploi développeur

Concernant la création et l’insertion des données dans notre base de données, elle se fait lors de votre première arrivée sur le site, vous n’avez qu’à cliquer sur « Création BDD » ce qui va exécuter notre script PHP (contenant toutes les requêtes SQL), que vous pouvez visualiser via le chemin « Cocktails\app\View\Database\index.ctp ». Nous avons mis en place un système qui va détecter si oui ou non la base de données existe, si c’est le cas l’action de création vous sera autorisée, sinon il n’y a plus lieu de vous y donner accès.

Afin de modifier la base de données, si vous voulez tester avec un autre fichier de données, il vous suffit de vous connecter sur le site avec les identifiants suivants :

* Login : « root »
* Mot de passe : « root »

Vous avez maintenant accès à toute la gestion du site mais tout d’abord, cliquez sur « Modification BDD », cela va exécuter notre script PHP qui va recréer et insérer les nouvelles données du fichier « Donnees.inc.php » qui est situé dans la vue de « Database » (Cocktails\app\View\Database\Donnees.inc.php) et que vous pouvez modifier à tout moment.

# Mode d’emploi utilisateur

Nous allons à présent vous décrire en détails les fonctionnalités disponibles pour l’utilisateur dit  « visiteur » :

* Profil : cette page permet à l’utilisateur de gérer lui-même son profil, il peut modifier toutes ses informations personnelles comme lors de son inscription, par un simple formulaire avec les données existantes pré-saisies qu’il peut modifier à sa volonté.
* Recettes : cette page consiste à présenter à l’utilisateur toutes les recettes enregistrées, il peut choisir de l’ajouter au panier ou bien de la voir en détails.
* Panier : comme le titre l’indique, c’est la page de visualisation du panier ou l’utilisateur peut supprimer des recettes ou encore vider totalement son panier.

Maintenant voici une présentation des fonctionnalités disponibles pour l’administrateur du site (voir identifiants de connexion au mode d’emploi développeur) :

* Profil : cette page est similaire en tout point du profil visiteur.
* Utilisateurs : cette page affiche la liste des utilisateurs inscrit et dont vous pouvez modifier les informations, ou bien les supprimer.
* Recettes : cette page affiche toutes les recettes comme si vous étiez visiteur, sauf que vous pouvez modifier ou supprimer une recette en plus de pouvoir la voir en détails.
* Catégories : vous trouverez ici toutes les catégories extraites du fichier de données ainsi que leurs catégorie(s) père(s). Vous pouvez ajouter, modifier ou supprimer une catégorie.
* Ingrédients : cette page affiche la liste des ingrédients ainsi que leur(s) catégorie(s). Vous avez également la possibilité d’ajouter, modifier ou supprimer un ingrédient.
* Modification BDD : cette option est présente pour vous permettre de réinitialiser la base de données lorsque vous avez effectué vos tests (par exemple si vous avez supprimé trop d’ingrédients).