|  |  |  |
| --- | --- | --- |
| TopRecettes | TPI 2015 | |
| Documentation Technique | | Cedric Dos Reis |

Table des matières

[1. Introduction 2](#_Toc419452881)

[2. Etude d’opportunité 2](#_Toc419452882)

[3. Analyse fonctionnelle 4](#_Toc419452883)

[3.1. Généralité 4](#_Toc419452884)

[3.2. Fonction globales 5](#_Toc419452885)

[3.3. Interfaces 8](#_Toc419452886)

[3.4. Eléments de sécurité 15](#_Toc419452887)

[4. Analyse Organique 16](#_Toc419452888)

[4.1. Généralité 16](#_Toc419452889)

[4.1.1. Model conceptuel de données 16](#_Toc419452890)

[4.2. Arborescence 18](#_Toc419452891)

[4.3. Description des méthodes (algorithmes) de résolution 19](#_Toc419452892)

[4.3.1. Description des actions 19](#_Toc419452893)

[4.3.2. Description des fonctions 24](#_Toc419452894)

[5. Tests 31](#_Toc419452895)

[5.1. Plans de tests 31](#_Toc419452896)

[5.2. Rapports de tests 33](#_Toc419452897)

[6. Plannings 35](#_Toc419452898)

[7. Conclusion 36](#_Toc419452899)

[7.1. Améliorations 36](#_Toc419452900)

[7.2. Comparaison énoncé avec projet final 36](#_Toc419452901)

[7.3. Comparaison des plannings 36](#_Toc419452902)

[8. Table des illustrations 37](#_Toc419452903)

[9. Références 37](#_Toc419452904)

# Introduction

Ce projet est réalisé dans le cadre du TPI. Le but du projet est de réaliser un site web qui permet de gérer des recettes de cuisine. Les recettes appartiennent à une catégorie de plat. Elles sont composées d’ingrédients avec différentes quantités. Les utilisateurs connectés peuvent ajouter et modifier leurs recettes, ils peuvent également commenter et noter les recettes. Un administrateur peut supprimer toutes les recettes, tous les commentaires, tous les utilisateurs et il peut également modifier les utilisateurs et les recettes.

Le site va être codé avec les langages suivant :

* PHP
* HTML5
* CSS3 (Bootstrap)
* JavaScript

# Etude d’opportunité

Les sites de recettes en ligne existent dans tous les pays et sont très nombreux sur le net.

Mon site étant un gestionnaire de recettes, je vais m’intéresser aux lieders français des recettes de cuisine en ligne

Marmiton propose aux visiteurs plus de 60'000 recettes. On y retrouve tous types de recettes comme par exemple les entrées, les plats principaux, les desserts et autres.

Ses différentes options de recherche permettent aux utilisateurs de trouver rapidement les recettes qu’ils recherchent. Ces options sont : la difficulté de la recette, son cout, le type de plat et d’autres tris.

Un utilisateur inscrit peut proposer une recette qui doit être validée par les administrateurs avant d’être affichée sur le site

Une recette sur marmiton contient un titre, la liste des ingrédients nécessaires et leur quantité, la préparation de la recette, le temps de préparation, le temps de cuisson, le nombre de portions et il y a parfois une vidéo en rapport avec la recette que vous consultez.

Marmiton propose aussi pour ceux qui le souhaitent d’afficher la recette en version imprimable.

750g propose aux visiteurs plus de 77'000 recettes. On y retrouve tous types de recettes comme par exemple les entrées, les plats principaux, les desserts et autres.



Sa recherche est plus simplifiée. Il suffit de taper un mot clé (du style nom d’une recette ou ingrédient) et le résultat est affiché en conséquence

il est impossible pour un visiteur d’ajouter ou de proposer une recette. Le site propose néanmoins de vous abonner au newsletter pour être informé des nouvelles recettes disponible

Une recette sur 750g contient un titre, la liste des ingrédients nécessaires et leur quantité, la préparation de la recette est séparée en plusieurs étapes, le temps de préparation, le temps de cuisson, le nombre de portions, le coût, la difficulté et comme sur marmiton, il y des vidéos en rapport avec la recette que vous consultez.

# Analyse fonctionnelle

## Généralité

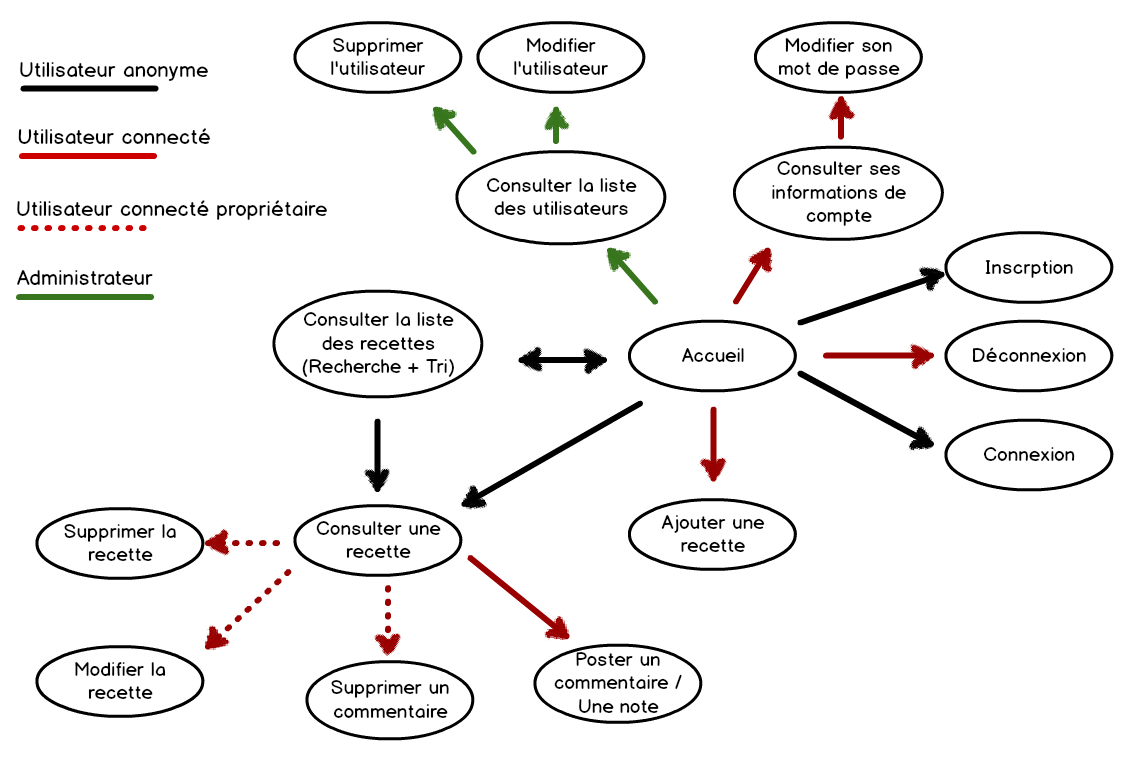


Figure : Schéma de fonctionnalités

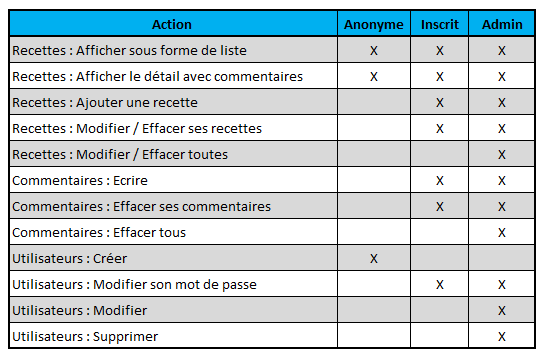


Figure : Droits des utilisateurs

## Fonction globales

**Connexion au site**

Un utilisateur anonyme peut se connecter au site s’il possède un déjà un compte.

Les informations nécessaires à la connexion sont :

* L’email
* Le mot de passe

**Inscription au site**

Un utilisateur anonyme peut s’inscrire au site s’il ne possède pas de compte.

Les informations nécessaires à la connexion sont :

* Un pseudo non utilisé
* Un email non utilisé
* Un mot de passe
* Une confirmation du mot de passe

**Afficher la liste des recettes**

Tout le monde peut afficher la liste des recettes, celles-ci peuvent être triées par date ou par note, il est également possible de rechercher une recette par son nom.

Pour un utilisateur connecté, il est également possible d’afficher uniquement ses recettes.

Les informations récupérées sont les suivantes :

* Le titre de la recette
* L’image représentant la recette
* L’auteur de la recette
* La date de publication de la recette
* La moyenne des notes attribuées à la recette

**Afficher les détails d’une recette**

Tout le monde peut consulter les détails d’une recette à partir de la liste.

Les informations récupérées sont les suivantes :

* Le titre de la recette
* L’image représentant la recette
* Les ingrédients qui composent la recette
* Les quantités d’ingrédients nécessaires à la réalisation de la recette
* L’explication pour la réalisation de la recette
* L’auteur de la recette
* La date de publication de la recette
* La moyenne des notes attribuées à la recette
* Les commentaires postés sur cette recette

**Ajouter une nouvelle recette**

Un utilisateur connecté peut ajouter une nouvelle recette

L’ajout d’une nouvelle recette se déroule en 2 étapes.

Les informations nécessaires à l‘ajout sont :

1ère étape :

* Un titre ->1ere étape
* Le type de plat
* Le nombre d’ingrédients nécessaires
* L’origine de la recette (Non obligatoire)

2ème étape

* Les ingrédients et leur quantité
* La préparation de la recette
* Une image (Non obligatoire)

**Modifier une recette**

L’auteur de la recette et les administrateurs ont la possibilité de modifier une recette.

La modification d’une recette se déroule comment l’ajout

**Supprimer une recette**

L’auteur de la recette et les administrateurs ont la possibilité de supprimer une recette.

**Ajouter un commentaire / note à une recette**

Un utilisateur connecté peut commenter une recette à partir de la page qui en affiche les détails.

Les informations nécessaires à l’ajout d’un commentaire sont :

* Un commentaire
* Une note pour la recette (Non obligatoire)

**Supprimer un commentaire / note**

L’auteur du commentaire et les administrateurs ont la possibilité de supprimer un commentaire

Si une note a été attribuée à cette recette

**Modifier son mot de passe**

Un utilisateur connecté à la possibilité de modifier son mot de passe

Les informations nécessaires à la modification du mot de passe sont :

* Le mot de passe actuel
* Le nouveau mot de passe
* Confirmation du nouveau mot de passe

**Modifier un utilisateur**

Un administrateur à la possibilité de modifier tous les utilisateurs/administrateurs

Les informations qu’il peut modifier sont :

* Le pseudo de l’utilisateur
* L’email de l’utilisateur
* Le type de l’utilisateur
* Le mot de passe de l’utilisateur

**Supprimer un utilisateur**

Un administrateur à la possibilité de supprimer tous les utilisateurs/administrateurs

## Interfaces

L’interface du site web est très simple, elle est constituée d’un menu en haut toujours visible par l’utilisateur. Juste en dessous du menu se trouve le titre et le contenu de la page et pour finir tout à la fin se trouve le pied de page.

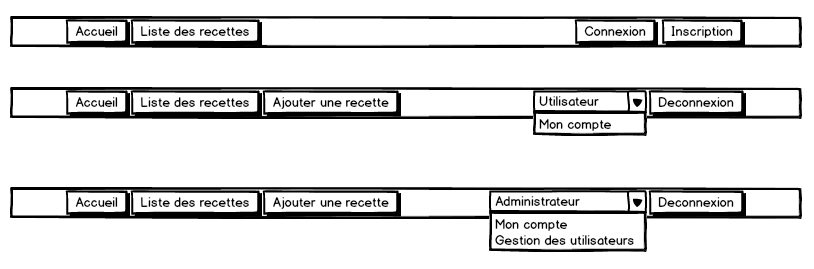
**Le menu change en fonction du type d’utilisateur.**

Un utilisateur anonyme a accès aux liens pour accéder à la fenêtre de connexion, d’inscription et la liste des recettes. Il n’y que lui qui peut accéder à la fenêtre de connexion et d’inscription.

Un utilisateur connecté a accès aux liens pour se déconnecter, pour ajouter une nouvelle recette, pour gérer son compte, pour afficher ses recettes et aussi la liste des recettes.

Un administrateur, lui à accès à toutes les pages disponible pour un utilisateur connecté et plus de celle pour la gestion des utilisateurs.

Figure : Différentes interfaces du menu utilisateur



**La page d’accueil** affiche quelques-unes de meilleures recettes présentes sur le site ainsi que les dernières ajoutées.

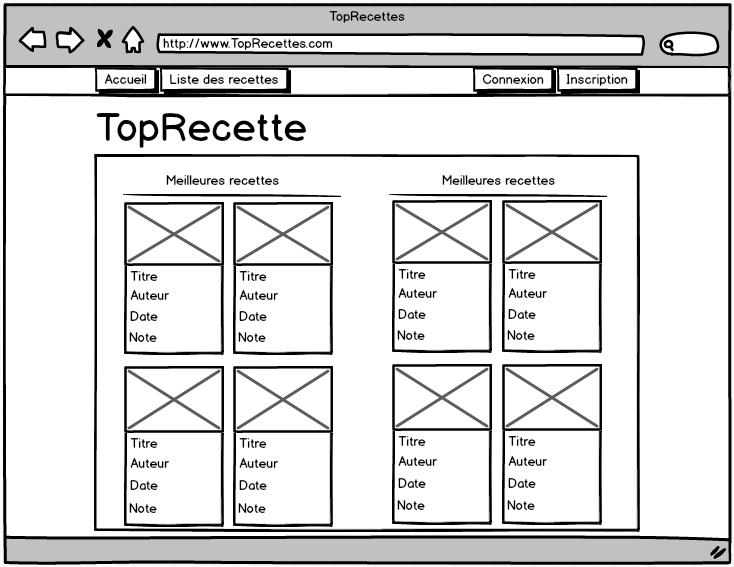


Figure : Interface - Accueil

**La page de connexion** contient un formulaire qui nécessite un email et un mot de passe.

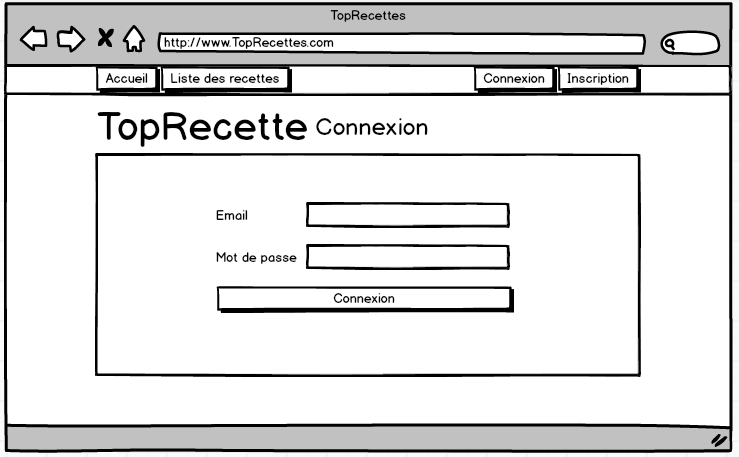


Figure : Interface - Connexion

**La page d’inscription** contient un formulaire qui nécessite un pseudo, un email, un mot de passe et une confirmation du mot de passe.

La page de modification d’un utilisateur ressemble fortement à celle-ci, si ce n’est que le page de modification permet de choisir le type d’utilisateur

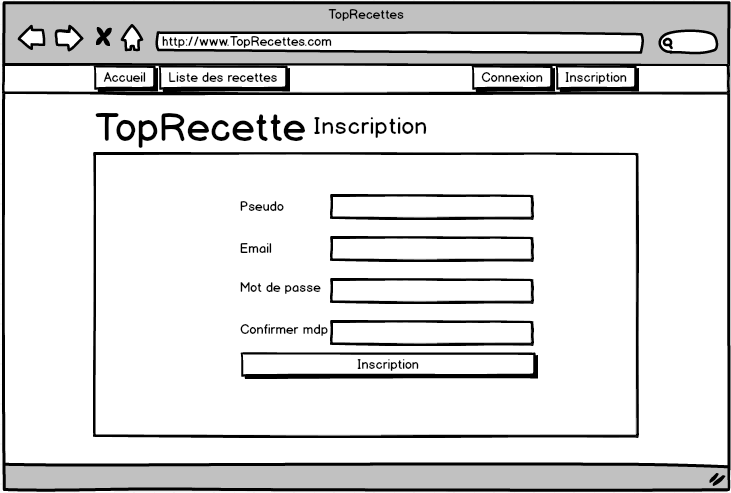


Figure : Interface - Inscription

**La page de consultation de la liste** **des recettes** permet d’afficher un aperçu des recettes disponibles sur le site. Elle permet à l’utilisateur de trier les recettes par date, note ou catégories.

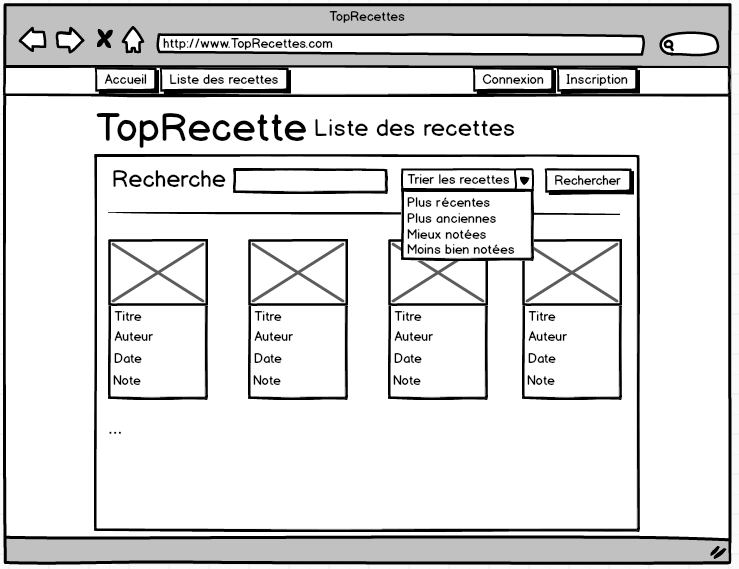
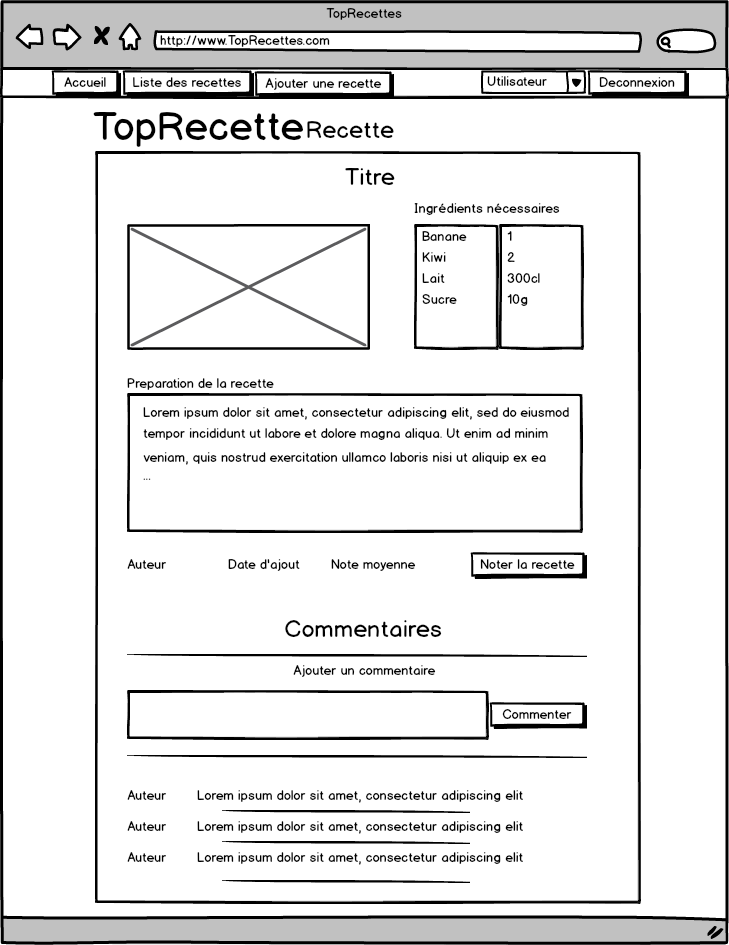


Figure : Interface – Consultation de la liste des recettes

**La page de consultation d’une recette** permet d’afficher toutes les informations nécessaire à la réalisation de la recette, elle affiche également les commentaires poster par les autres utilisateurs. Le propriétaire de la recette et l’administrateur ont la possibilité de supprimer la recette depuis cette page et également d’accéder à la page de modification.

Figure : Interface - Consultation d'une recette



**La page d’édition de recette** permet à un utilisateur connecter de rédiger une recette et l’envoyer. Elle permet également de modifier une recette pour le propriétaire ou l’admin.

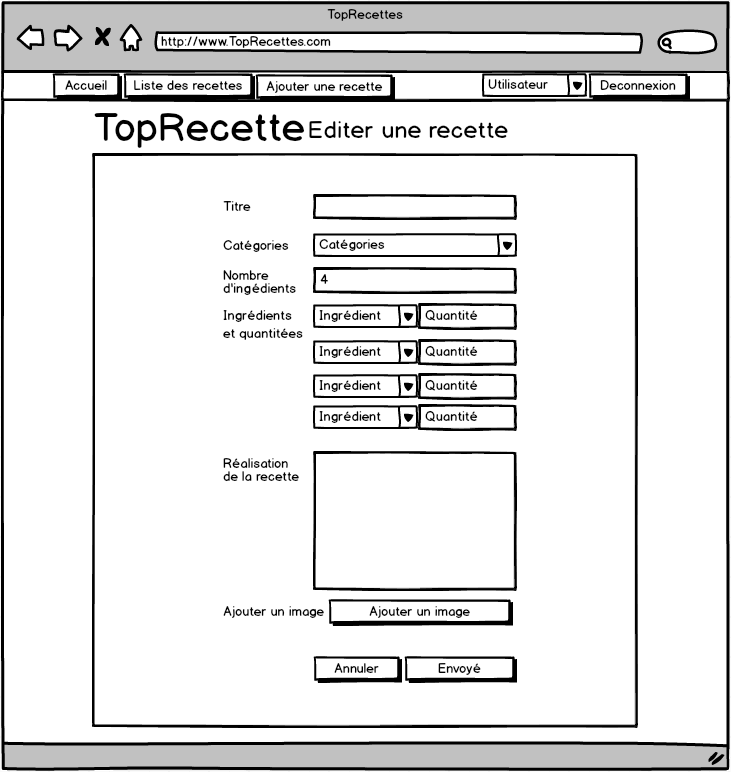
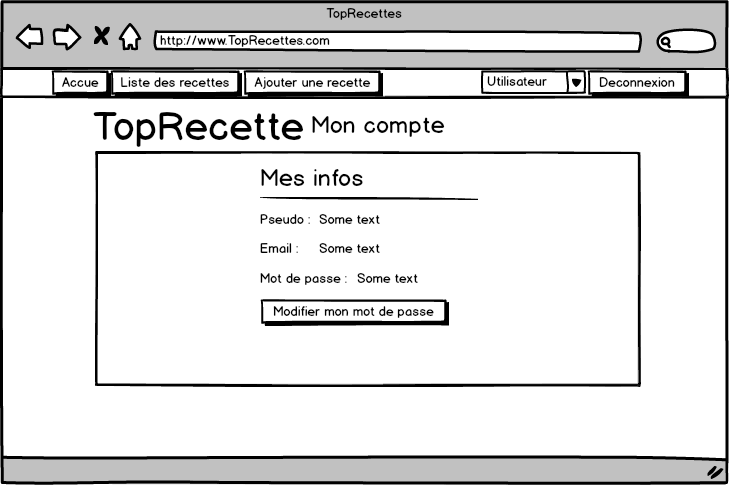


Figure : Interface - édition d’une recette

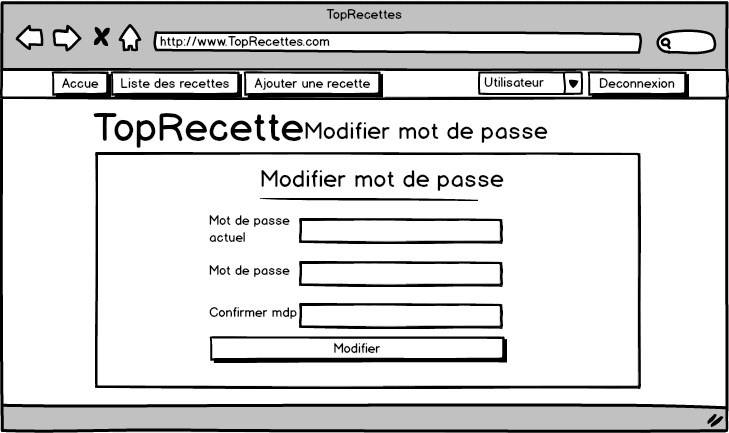
**La page de consultation d’un utilisateur** permet à un utilisateur connecté de consulter ses informations de compte.

Figure : Interface - Consultation d'un utilisateur



**La page de modification de mot de passe** permet à un utilisateur connecté de modifier son mot de passe

Figure : Interface - Modification du mot de passe



**La page de consultation de la liste** **des utilisateurs** permet à un administrateur d’afficher tous les utilisateurs qui se sont inscrits sur le site.

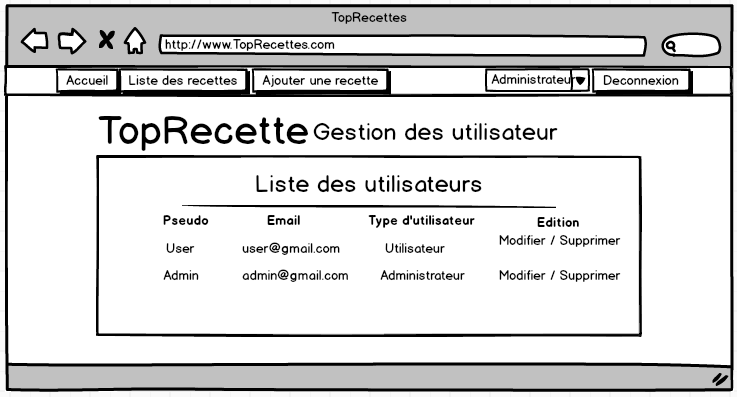


Figure : Interface - Consultation de la liste des utilisateurs

**La page de modification d’un utilisateur** permet à un administrateur de modifier toutes les informations d’un utilisateur

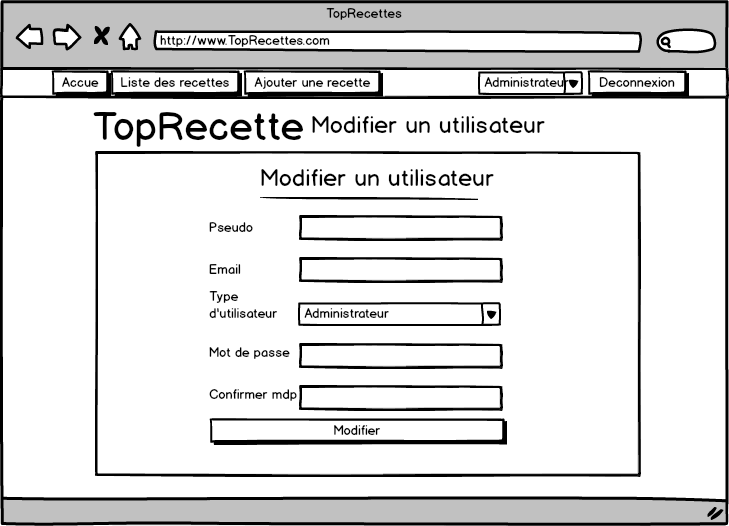


Figure : Interface - Modification d'un utilisateur

## Eléments de sécurité

Pour assurer la sécurité des utilisateurs, les mots de passe son cryptés grâce à la fonction de hachage *Secure Hash Algorithm* « SHA1 ».

# Analyse Organique

## Généralité

### Model conceptuel de données

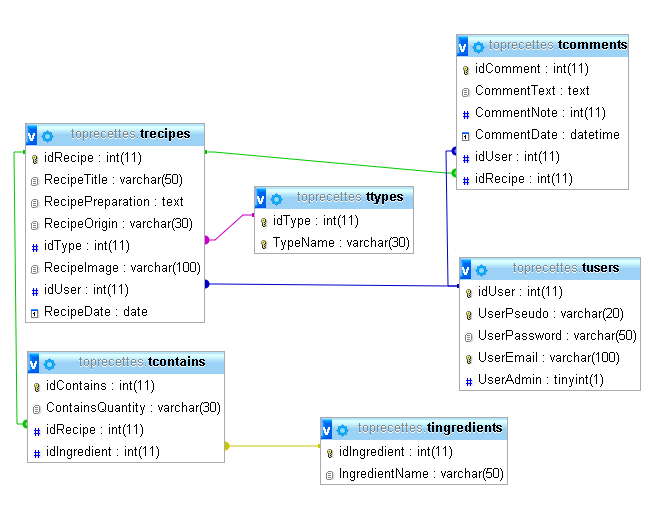


Figure : Model conceptuel de données

Voici mon model conceptuel de données. Il contient 6 tables. La table principale est « trecipes » elle contient toutes les recettes, elle est liée à la table « tingredients » en passant par la table « tcontains » qui contient les contenus des recettes. Ensuite, la table « tusers » elle contient tous les utilisateurs inscrits. Les commentaires que les utilisateurs postent sont dans la table « tcomments »

#### Description des tables

|  |  |  |
| --- | --- | --- |
| **Table trecipes** | | |
| **Nom** | **Type** | **Description** |
| idRecipe | Int(11), AI, Primary | Identifiant de la recette |
| RecipeTitle | varchar(50) | Titre de la recette |
| RecipePreparation | text | Préparation de la recette |
| RecipeOrigine | varchar(30), NULL | Origine de la recette |
| RecipeImage | varchar(100), | Chemin d’accès à l’image  Par défaut: ./imgRecette/toprecette.jpg |
| RecipeDate | date | Date d’ajout de la recette |
| idUser | int(11), Index | Identifiant de l’auteur de la recette |
| idType | int(11), Index | Identifiant du type de recette |

|  |  |  |
| --- | --- | --- |
| **Table tcontains** | | |
| **Nom** | **Type** | **Description** |
| idContains | int(11), AI, Primary | identifiant du contenu |
| ContainsQuantity | varchar(30) | Quantité d’ingrédient |
| idRecipe | int(11), Index | Identifiant de la recette à lquelle appartient ce contenu |
| idIngredrient | int(11), Index | Identifiant de l’ingrédient |

|  |  |  |
| --- | --- | --- |
| **Table tcomments** | | |
| **Nom** | **Type** | **Description** |
| idComment | int(11), AI, Primary | identifiant du commentaire |
| CommentText | text | Commentaire |
| CommentNote | int(11), NULL | Note |
| CommentDate | datetime | Date d’ajout du commentaire |
| idUser | int(11), Index | Identifiant de l’auteur du commentaire |
| idRecipe | int(11), Index | Identifiant de la recette ou est posté le commentaire |

|  |  |  |
| --- | --- | --- |
| **Table tingredients** | | |
| **Nom** | **Type** | **Description** |
| idIngredient | int(11), AI, Primary | identifiant de l’ingrédient |
| IngredientName | varchar(50) | Nom de l’ingrédient |

|  |  |  |
| --- | --- | --- |
| **Table ttypes** | | |
| **Nom** | **Type** | **Description** |
| idIngredient | int(11), AI, Primary | identifiant du type |
| TypeName | varchar(30) | Nom du type |

|  |  |  |
| --- | --- | --- |
| **Table tusers** | | |
| **Nom** | **Type** | **Description** |
| idUser | int(11), AI, Primary | identifiant de l’utilisateur |
| UserPseudo | varchar(20) | Pseudo de l’utilisateur |
| UserPassword | varchar(50) | Mot de passe de l’utilisateur |
| UserEmail | varchar(100) | Adresse email de l’utilisateur |
| UserAdmin | tinyint(1) | Type d’utilisateur (Admin ou Utilisateur) |

## Arborescence

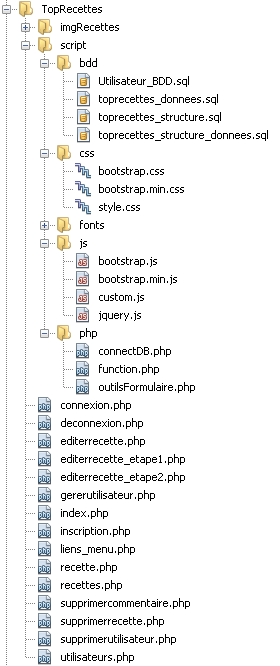


Figure : Arborescence du site

Voici l’arborescence du site. À la racine du dossier **TopRecettes** se trouve 2 dossiers **ImgRecettes** et **script** et page du site**.** Le dossier **imgRecette** contient les images uploadées par les utilisateurs lorsqu’ils ajoutent ou modifient une recette.

Le dossier **script** contient des dossiers qui contiennent les scriptes du site.

Dans le dossier **bdd**, il y a les fichiers de structure et des données de la base.

Le dossier **css** contient les feuilles de style de bootstrap et une feuille de style personnalisée. Ces fichiers sont le thème du site.

Le dossier **fonts** contient les polices d’écritures utilisé par bootstrap.

Le dossier **js** contient les fichiers javascript de bootstrap, jquery et un fichier personnalisé.

Le dossier **php** contient les fichiers exécuté par le serveur. Ces contiennent les fonctions utilisé pour gérer le site.

## Description des méthodes (algorithmes) de résolution

### Description des actions

#### Connexion

La connexion permet à un utilisateur de s’identifier sur le site.

L’utilisateur doit entrer un pseudo et un mot de passe, si l’utilisateur tente de se connecter sans pseudo ou mot de passe, un message d’erreur est affiché et l’action se termine.

Si le pseudo et le mot de passe sont reçus alors le mot de passe est crypté et la fonction « login » est appelée, si elle renvoie « true » alors l’utilisateur est connecté et il est ensuite redirigé à la page d’accueil. Si la fonction renvoie « false » alors un message d’erreur s’affiche et l’action se termine.

#### Inscription

L’inscription permet à un utilisateur de s’inscrire sur le site.

L’utilisateur doit entrer un pseudo, une adresse email, un mot de passe et une confirmation du mot de passe. Si l’utilisateur tente de s’inscrire sans entrer correctement les données demandées alors un message d’erreur est affiché et l’action est annulée.

Si les informations demandées sont reçues alors les mots de passe sont testés. Si les mots de passe ne sont pas pareils alors un message d’erreur s’affiche et l’action est annulée. Si les mots de passe sont pareils alors l’adresse email est vérifiée. Si l’adresse email n’est pas valide alors un message d’erreur s’affiche et l’action est annulée. Si l’adresse email est valide alors le pseudo est vérifié grâce à la fonction « CheckExist\_Pseudo », si la fonction renvoie « true » c’est que le pseudo est déjà utilisé donc un message d’erreur s’affiche et l’action est annulée. Si la fonction renvoie « false » alors le pseudo n’est pas utilisé et l’adresse email est vérifiée grâce à la fonction « CheckExist\_Email », si la fonction renvoie « true » c’est que l’adresse email est déjà utilisée donc un message d’erreur s’affiche et l’action est annulée. Si la fonction renvoie « false » alors l’adresse email pas utilisée et la fonction « register » est appelée, elle insert les données dans la base et l’utilisateur est automatiquement connecté et redirigé à page d’accueil.

#### Afficher les recettes sous forme de liste + Recherche et tri

Il n’y pas besoin d’effectuer un tri ou une recherche pour afficher la liste des recettes

Si l’utilisateur est connecté alors l’option de tri « Mes recettes » est ajoutée aux choix de tri possibles.

Si aucun tri ni recherche n’est effectuée alors la fonction « get\_recipes » est appelée et toutes les recettes sont récupérées et affichées.

Si un tri ou une recherche est effectué alors la fonction « get\_recipes » est appelée et les recettes triées et/ou recherchées sont récupérées et affichées.

#### Afficher les détails d’une recette

Pour afficher les détails d’une recette, il faut recevoir l’identifiant d’une recette dans l’url de la page. Si l’identifiant n’est pas définit dans l’url alors l’utilisateur est redirigé à la liste des recettes. Si l’identifiant est défini alors les informations de la recette sont récupérées grâce à la fonction « get\_recipe ». Si aucune information n’est récupérée c’est que la recette n’existe pas et l’utilisateur est redirigé à la page de liste. Si les informations sont récupérées alors les ingrédients (contenus de la recette) et les commentaires sont récupérés grâce aux fonctions « get\_ingredients\_recipe » et « get\_comments\_recipe ».

Si aucun commentaire n’est récupéré alors le message « Aucuns commentaires » s’affiche.

#### Ajouter une recette

Il faut être connecté pour ajouter une recette, si un utilisateur non connecté tente d’ajouter une recette, il est redirigé à la page d’accueil.

L’ajout d’une recette se déroule en 2 étapes.

Pour la première étape, l’utilisateur doit entrer les champs suivant : le titre de la recette, le type de la recette, le nombre d’ingrédients qui compose la recette et l’origine de la recette. Si une de ses informations est manquante (L’origine n’est pas obligatoirement requise) alors un message d’erreur s’affiche et l’ajout reste à l’étape 1. Si les informations demandées sont reçues alors elles sont récupérées et enregistrées dans la session de l’utilisateur et l’ajout passe à l’étape 2.

Lorsque l’ajout passe à l’étape 2, des champs sont créés dans le formulaire par rapport au nombre d’ingrédients défini dans l’étape 1. Ces champs sont « Nom de l’ingrédient » et « quantité de l’ingrédient ». Les champs « Nom d’ingrédient » proposent à l’utilisateur les ingrédients déjà disponible dans la base. L’utilisateur peut entrer un ingrédient qui ne lui est pas proposé.

Pour la deuxième étape, l’utilisateur doit entrer les champs suivant : les ingrédients et leur quantité, si un ingrédient n’a pas de quantité et inversement alors il ne sera pas ajouté dans le contenu, il faut obligatoirement l’ingrédient et sa quantité pour qu’ils soient ajouté, il faut également la préparation de la recette et une image (l’image n’est pas obligatoire).

Les identifiant des ingrédients entrés par l’utilisateur sont récupérés, si ‘un ingrédient n’existe pas dans la base alors il est ajouté.

La recette est ajoutée avec ou sans image et son contenu est ajouté grâce aux fonctions « add\_recipe » et « add\_contains »

#### Modifier une recette

Pour modifier une recette, il faut recevoir l’identifiant de la recette a modifié dans l’url de la page. Si l’identifiant n’est pas définit dans l’url alors l’utilisateur est redirigé à la liste des recettes. Si l’identifiant est défini alors la modification peut commencer.

Il faut être connecté et être le propriétaire de la recette ou être administrateur pour modifier une recette, si un utilisateur non connecté tente de modifier une recette, il est redirigé à la page d’accueil.

Les informations actuelles de la recette sont affichées dans les champs des formulaires.

La modification d’une recette se déroule en 2 étapes.

Pour la première étape, l’utilisateur peut modifier les champs suivant : le titre de la recette, le type de la recette, le nombre d’ingrédients qui compose la recette et l’origine de la recette. Si une de ses informations est manquante (L’origine n’est pas obligatoirement requise) alors un message d’erreur s’affiche et la modification reste à l’étape 1. Si les informations demandées sont reçues alors elles sont récupérées et enregistrées dans la session de l’utilisateur et la modification passe à l’étape 2.

Lorsque la modification passe à l’étape 2, des champs sont créés dans le formulaire par rapport au nombre d’ingrédients défini dans l’étape 1. Ces champs sont « Nom de l’ingrédient » et « quantité d’ingrédient ». Ils contiennent les ingrédients et quantités que composent actuellement la recette. Les champs « Nom de l’ingrédient » proposent à l’utilisateur les ingrédients déjà disponible dans la base. L’utilisateur peut entrer un ingrédient qui ne lui est pas proposé.

Pour la deuxième étape, l’utilisateur peut modifier les champs suivant : les ingrédients et leur quantité, si un ingrédient n’a pas de quantité et inversement alors il ne sera pas ajouté dans le contenu, il faut obligatoirement l’ingrédient et sa quantité pour que le contenu soit modifié, il faut également la préparation de la recette et une image (l’image n’est pas obligatoire). Si la préparation n’est pas définit alors un message d’erreur s’affiche et la modification n’est pas effectuée.

Les identifiant des ingrédients entrés par l’utilisateur sont récupérés, si ‘un ingrédient n’existe pas dans la base alors il est ajouté.

La recette est modifié avec ou sans image, son contenu actuel est effacé puis son nouveau contenu est ajouté grâce aux fonctions « edit\_recipe », « add\_contains » et « delete\_contains\_recipe ».

#### Supprimer une recette

Pour supprimer une recette, il faut recevoir l’identifiant de la recette a supprimé dans l’url de la page. Si l’identifiant n’est pas définit dans l’url alors l’utilisateur est redirigé à la liste des recettes. Si l’identifiant est défini alors la suppression peut commencer.

Pour supprimer une recette il fait être connecté et être l’auteur de la recette ou un administrateur. Si un utilisateur qui ne fait pas parti de ceux mentionné ci-dessus essaie de supprimer une recette, il sera redirigé sur la page de la liste des recettes.

Si l’utilisateur connecté est le propriétaire de la recette ou un administrateur alors le chemin d’accès à l’image de la recette est récupéré grâce à la fonction « get\_recipe\_image » pour être supprimé, la recette est supprimée, les contenus de la recette sont supprimés, les commentaires postés sur la recette sont supprimé et l’utilisateur est redirigé à la liste.

#### Ajouter un commentaire/ une note

Pour commenter une recette, il faut être connecté, dans le cas contraire vous n’aurez pas accès au formulaire.

Il est possible de noter la recette en même temps qu’elle est commentée.

Si le commentaire n’est pas défini ou vide alors un message d’erreur s’affiche et le commentaire n’est pas ajouté.

Si le commentaire reçu est définit alors la note est vérifiée. Si la note n’est pas définie le commentaire est ajouté sans note. Si la note est définie et est un chiffre entre 1 et 5 alors le commentaire sera ajouté avec la note. Si la note n’est pas un chiffre entre 1 et 5 alors le commentaire n’est pas ajouté et un message d’erreur s’affiche.

#### Supprimer un commentaire

Pour supprimer un commentaire, il faut recevoir l’identifiant du commentaire a supprimé dans l’url de la page. Si l’identifiant n’est pas définit dans l’url alors l’utilisateur est redirigé à la liste des recettes. Si l’identifiant est défini alors la suppression peut commencer.

Pour supprimer un commentaire il fait être connecté et être l’auteur du commentaire ou un administrateur. Si un utilisateur qui ne fait pas parti de ceux mentionné ci-dessus essaie de supprimer un commentaire, il sera redirigé sur une autre page.

Si l’utilisateur connecté est le propriétaire du commentaire ou un administrateur alors le commentaire est supprimé de la base grâce à la fonction « delete\_comment» L’utilisateur est ensuite redirigé.

#### Modifier son mot de passe

Un utilisateur connecté peut modifier son mot de passe.

Pour modifier son mot de passe il devra entrer les champs suivant : son mot de passe actuelle, son nouveau mot de passe et une confirmation du nouveau mot de passe.

Si ‘il manque une de ces informations, un message d’erreur apparait et la modification n’est pas effectuée

Si tous les champs sont définis alors le mot de passe actuel de l’utilisateur est récupéré de la base puis il est comparé avec le champ du mot de passe actuel. Si les mots de passe ne correspondent pas alors un message d’erreur s’affiche et la modification est annulée. Si les mots de passe correspondent alors les nouveaux mots de passe sont comparés. S’ils sont différents alors un message d’erreur s’affiche et la modification est annulée. Si les nouveaux mots de passe sont pareils alors le mot de passe de l‘utilisateur est changé dans la base. Le mots de passe sont cryptés grâce à la fonction SHA1

#### Modifier un utilisateur

Pour modifier un utilisateur, il faut être administrateur. Dans le cas contraire l’utilisateur est redirigé.

L’administrateur peut changer le pseudo, l’adresse email, le type d’utilisateur et le mot de passe de l’utilisateur.

Chaque champ est testé pour vérifier s’il a été modifié par rapport aux informations du début. Selon si le champ est modifié et vérifié, il sera modifié dans la base

Si les informations demandées sont reçues alors le pseudo est testé.

Si le pseudo a été modifié et n’est pas déjà utilisé par un autre utilisateur alors il est modifié dans le base. Si le pseudo est déjà utilisé alors un message d’erreur s’affiche. L’email est testé.

Si l’email a été modifié et n’est pas déjà utilisé par un autre utilisateur alors il est modifié dans le base. Si l’email est déjà utilisé alors un message d’erreur s’affiche. Le type d’utilisateur est testé.

Si type d’utilisateur a été modifié alors il est modifié dans le base. Les mots de passe sont testés.

Si un mot de passe et une confirmation de mot de passe sont reçus alors ils sont cryptés et testés. Si les mots de passe ne sont pas pareils alors un message d’erreur s’affiche. Si les mots de passe sont pareils alors il est modifié dans la base.

#### Supprimer un utilisateur

Pour supprimer un utilisateur, il faut être administrateur. Dans le cas contraire l’utilisateur est redirigé.

### Description des fonctions

|  |  |  |
| --- | --- | --- |
| **Fonctions Php** | **Paramètres** | **Description** |
| connectDB |  | Ouvre la connexion à la base de données |
| Select | $name  $liste  $courant  $required | Permet de créer un Select (formulaire) à partir d'un tableau associatif ($liste)  Sélectionne l’option définit dans $courant |
| DataList | $name  $liste | Permet de créer une Datalist (formulaire) à partir d'un tableau associatif ($liste) |
| ShowError | $Message | Affiche un message dans un block rouge (Message d’erreur en paramètre). |
| ShowSuccess | $Message | Affiche un message dans un block vert (Message de succès en paramètre). |
| login | $UserEmail  $UserPassword | Connexion au site.  Récupère l’utilisateur dont les données sont reçues en paramètres correspondent.  Si un utilisateur est récupéré alors elle renvoie vrai et l’utilisateur est connecté.  Si aucun utilisateur n’est récupéré, renvoie faux. |
| register | $UserPassword  $UserEmail  $UserPassword | Inscription au site.  Insert les données en paramètre dans la table tusers. |
| get\_users |  | Récupère tous les utilisateurs présents dans la base.  Renvoie la liste des utilisateurs. |
| get\_user | $idUser | Récupère les données (Email, Pseudo, Type d’utilisateur) de l'utilisateur dont l’identifiant est en paramètre  Renvoie les données de l’utilisateur. |
| get\_user\_pseudo | $IdUser | Récupère le pseudo de l'utilisateur dont l’identifiant est en paramètre.  Renvoie le pseudo. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fonctions Php** | **Paramètres** | **Description** | |
| check\_password | $idUser  $Password | Récupère le mot de passe de l’utilisateur dont l’identifiant est en paramètre.  Vérifie que le mot de passe récupéré correspond au mot de passe en paramètre.  Renvoie vrai si les mots de passe correspondent.  Renvoie faux s’ils ne correspondent pas. | |
| CheckExist\_Pseudo | $UserPseudo | Vérifie si le pseudo en paramètre est déjà utilisé.  Récupère le pseudo qui correspond au pseudo en paramètre.  Si un pseudo est récupéré, renvoie vrai car il existe déjà.  Si aucun pseudo n’est récupéré, renvoie faut car il n’existe pas. | |
| CheckExist\_Email | $UserEmail | Vérifie si l’adresse email reçu en paramètre est déjà utilisés Récupère l’email qui correspond à l’email reçu.  Si un email est récupéré renvoie vrai car il existe déjà.  Si aucun email n’est récupéré, renvoie faut car elle n’existe pas. | |
| CheckAdmin | $idUser | Test le type d’utilisateur.  Récupère le type de l’utilisateur dont l’identifiant est en paramètre.  Si le type est 0, renvoie faux car l’utilisateur n’est pas administrateur.  Si le type est 1, renvoie vrai car l’utilisateur est administrateur. | |
| edit\_user | $UserPseudo  $UserEmail  $idUser | Modifie les données (Pseudo et Email) de l'utilisateur dont l’identifiant est en paramètre.  Si le pseudo en paramètre est défini et l’email paramètre est vide alors modifie uniquement le pseudo de l’utilisateur.  Si le pseudo est vide et l’email est défini alors modifie uniquement l’email de l’utilisateur.  Si le pseudo et l’email sont définis alors modifie le pseudo et l’email de l’utilisateur. |

|  |  |  |
| --- | --- | --- |
| **Fonctions Php** | **Paramètres** | **Description** |
| set\_administrateur | $UserAdmin  $idUser | Modifie le type de l'utilisateur en fonction du type d’utilisateur qui est en paramètre. |
| edit\_user\_password | $idUser  $NewPassword | Modifie le mot de passe de l'utilisateur dont l’identifiant est en paramètre. |
| delete\_user | $idUser | Supprime l'utilisateur dont l’identifiant est en paramètre. |
| get\_recipes | $sort  $search  $idUser  $limit | Récupère le la liste de des recettes en fonction des paramètres reçu.  Si tous les paramètres sont vides alors elle récupère toutes les recettes de la base.  Si le tri ($sort) est défini alors elle récupère toutes les recettes en les triant selon le tri reçu (par date d’ajout, par note).  Si la recherche ($search) est définie alors elle récupère les recettes qui correspondent à la recherche.  Si l’identifiant d’un utilisateur est reçu alors elle récupère les recettes postées par cet utilisateur.  Si une limite est définie alors elle récupère un nombre de recettes défini par la limite  Un tri et une recherche peuvent être effectués en même temps  Renvoie en tableau de données contenant les recettes récupérées |
| get\_recipe | $idRecipe | Récupère les données (Titre, préparation, origine, type, moyenne des notes, image, l’auteur) d’une recette dont l’identifiant est en paramètre.  Renvoie un tableau de données. |

|  |  |  |
| --- | --- | --- |
| **Fonctions Php** | **Paramètres** | **Description** |
| add\_recipe | $RecipeInfos  $idUser | Insert une nouvelle recette dans la table trecipe (Titre, préparation, origine, type, moyenne des notes, image, l’auteur).  Si une image est reçue alors elle upload et ajoute la recette avec l’image reçu en paramètre.  Su aucune image n’est reçue alors elle ajoute la recette sans image.  Renvoie l’identifiant de la recette qui vient d’être ajouté. |
| edit\_recipe | $idRecipe  $RecipeInfos | Modifie la recette dans la table trecipes dont l’identifiant est reçu en paramètre.  Si une nouvelle image est reçue alors elle vérifie l’image actuelle de la recette.  Si l’image actuelle de la recette est celle par défaut alors elle upload la nouvelle image et modifie la recette avec la nouvelle image.  Si l’image actuelle est différente de celle par défaut alors l’image actuelle est supprimée du dossier, la nouvelle image est uploadée et la recette est modifiée avec la nouvelle image.  Si aucune image n’est reçue alors elle modifie la recette sans changer l’image actuelle. |
| delete\_recipe | $idRecipe | Supprime la recette dont l’identifiant est en paramètre de la table trecipes |
| get\_recipe\_image | $idRecipe | Récupère le chemin d'accès à l'image de la recette dont l'identifiant est en paramètre.  Si le chemin récupéré correspond au chemin de l'image par défaut alors elle renvoie false.  Si le chemin récupéré ne correspond pas au chemin de l'image par défaut alors elle renvoie le chemin récupéré. |
| get\_ingredients |  | Récupère tous les ingrédients de la base.  Renvoie un tableau de données avec tous les ingrédients de la base |

|  |  |  |
| --- | --- | --- |
| **Fonctions Php** | **Paramètres** | **Description** |
| count\_ingredient\_recipe | $idRecipe | Récupère le nombre d'ingrédients qui composent la recette dont l'identifiant est en paramètre  Renvoie le nombre d’ingrédients |
| checkExist\_ingredient | $IngredientName | Vérifie si l’ingrédient en paramètre existe dans la base.  Récupère l’identifiant de l’ingrédient en paramètre.  Si aucun identifiant n’est récupéré alors elle renvois faux car l’ingrédient n’existe pas.  Si un identifiant est récupéré alors elle le renvoie car l’ingrédient existe ; |
| add\_ingredient | $IngredientName | Insert un nouvel ingrédient dans la table ingrédients. |
| ingredients\_associate |  | Transforme un tableau de données en un tableau associatif.  Récupère un tableau de données qui contient la liste des ingrédients.  Transforme les valeurs du tableau dans un nouveau tableau avec comme valeur le nom de l’ingrédient et en index l’identifiant de l’ingrédient.  Renvoie le tableau associatif |
| get\_recipe\_types |  | Récupère tous les types de recettes.  Renvoie un tableau de données |
| recipe\_types\_associate |  | Transforme un tableau de données en un tableau associatif.  Récupère un tableau de données qui contient la liste des types de recettes.  Transforme les valeurs du tableau dans un nouveau tableau avec comme valeur le nom de l’ingrédient et en index l’identifiant de l’ingrédient.  Renvoie le tableau associatif |
| add\_contains | $idIngredient  $IngredientQuantity  $idRecipe | Insert un contenu (un ingrédient et une quantité) à la recette en paramètre dans la table tcontains |

|  |  |  |
| --- | --- | --- |
| **Fonctions Php** | **Paramètres** | **Description** |
| get\_ingredients\_recipe | idRecipe | Récupère les contenus de la recette dont l’identifiant est en paramètre.  Renvoie un tableau de données. |
| get\_comments\_recipe | idRecipe | Récupère les commentaires postés sur la recette dont l’identifiant est en paramètre. |
| add\_comment | idUser  idRecipe  comment  note | Insert un nouveau commentaire dans la table tcomments. |
| delete\_user\_comments | idUser | Supprime les commentaires postés par l’utilisateur dont l’identifiant est en paramètre. |
| delete\_link\_user\_recipes | idUser | Supprime les liens entre l'utilisateur dont l’identifiant est en paramètre et les commentaires qu'il a posté  Remplace l’identifiant de l’utilisateur par un 0. |
| check\_owner\_recipe | idUser  idRecipe | Vérifie si l'utilisateur est le propriétaire de la recette.  Récupère l’identifiant de l’utilisateur qui représente le propriétaire de la recette dont l’identifiant est en paramètre.  Si l’identifiant récupéré correspond à l’identifiant en paramètre alors elle renvois vrai car l’utilisateur est le propriétaire de la recette.  Si l’identifiant récupéré ne correspond à l’identifiant en paramètre alors elle renvois faux car l’utilisateur n’est pas le propriétaire de la recette. |
| check\_owner\_comment | idUser  idComment | Vérifie si l'utilisateur est le propriétaire du commentaire.  Récupère l’identifiant de l’utilisateur qui représente le propriétaire du commentaire dont l’identifiant est en paramètre.  Si l’identifiant récupéré correspond à l’identifiant en paramètre alors elle renvois vrai car l’utilisateur est le propriétaire du commentaire.  Si l’identifiant récupéré ne correspond à l’identifiant en paramètre alors elle renvois faux car l’utilisateur n’est pas le propriétaire du commentaire. |
| **Fonctions Php** | **Paramètres** | **Description** |
| delete\_comment | idComment | Supprime le commentaire dont l’identifiant est en paramètre |
| delete\_comment\_recipe | Recipe | Supprime les commentaires, notes qui ont été postés sur la recette dont l’identifiant est en paramètre |
| delete\_contains\_recipe | idRecipe | Supprime les contenus de la recette dont l’identifiant est en paramètre |
| show\_avg\_note\_recipe | AvgNote | Affiche 5 étoiles vides et/ou remplies (Affichage de la note)  L’étoile est remplie si la note reçue en paramètre est plus grande |
| upload | file | upload l’image en paramètre dans le dossier imgRecettes.  L'image est traitée afin de changer son nom.  Son nouveau nom est définit par la fonction uniqid(). |

# Tests

## Plans de tests

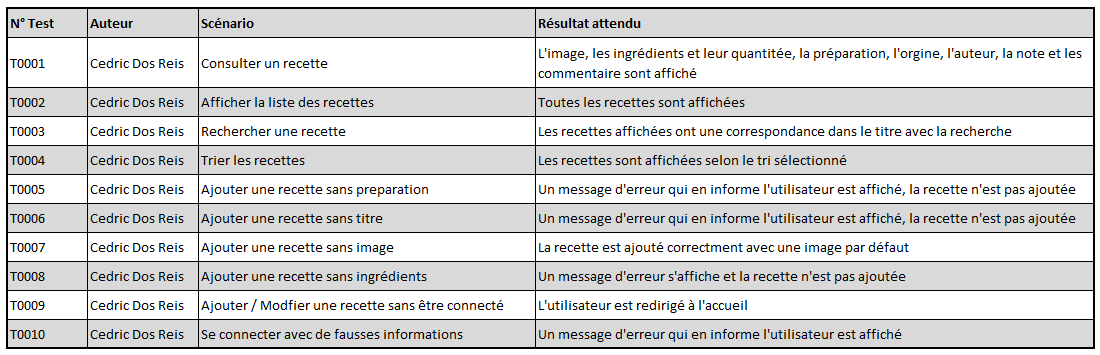


Figure : Plan de tests n°1

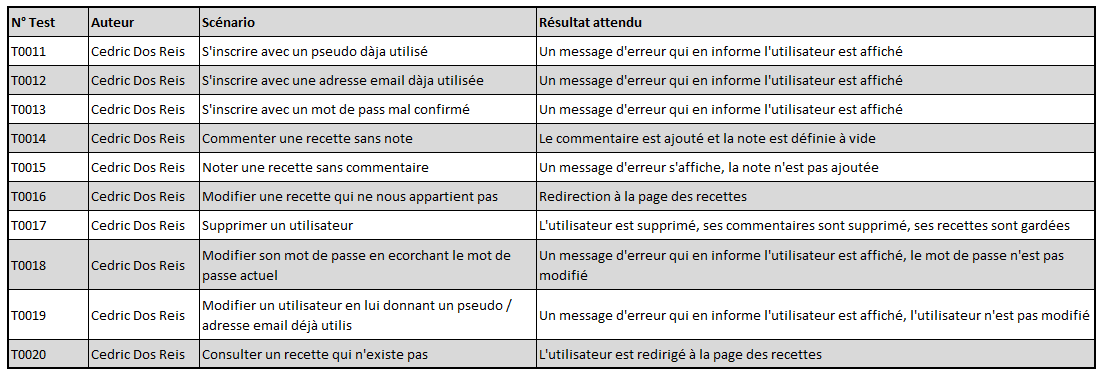


Figure : Plan de tests n°2

* 1. Rapports de tests

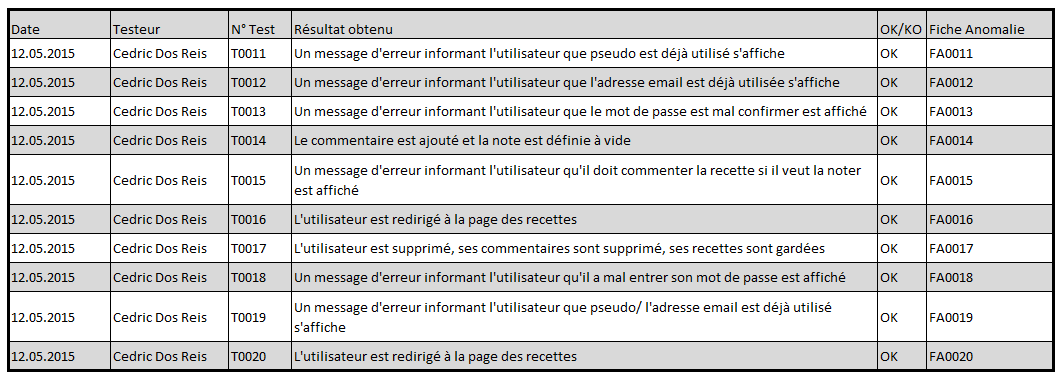
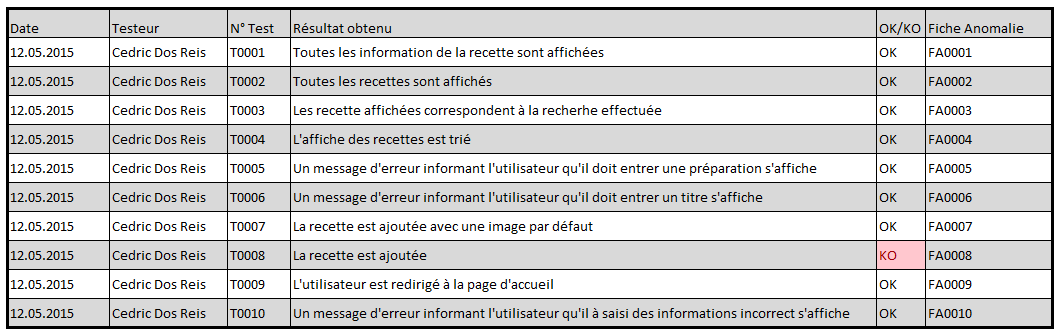
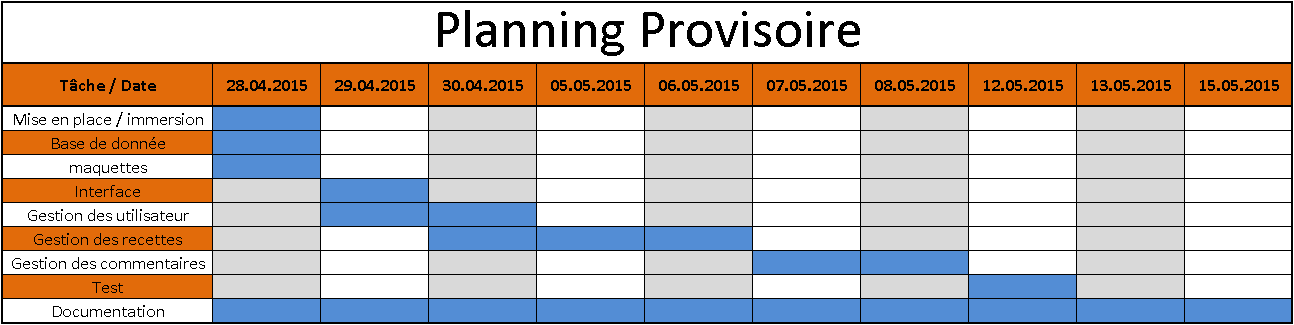


Figure : Rapport de tests n°1

Figure : Rapport de tests n°2

# Plannings



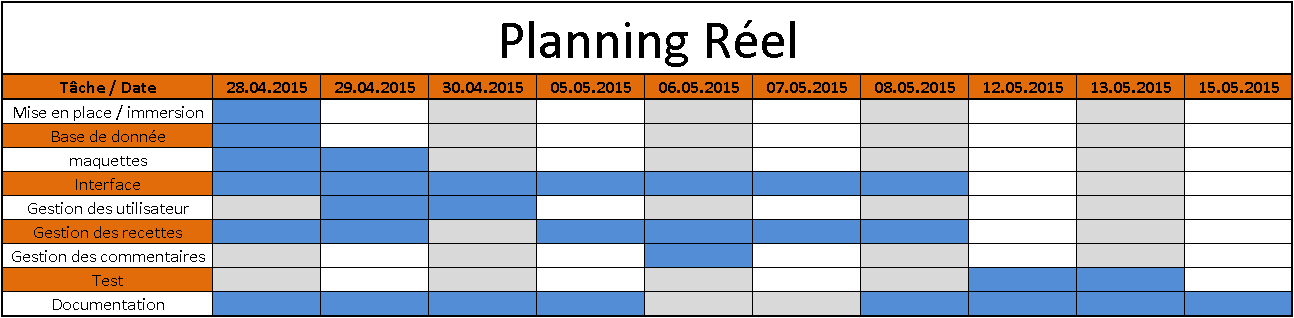
Figure : Planning provisoire

Figure  : Planning réel

# Conclusion

## Améliorations

Voici quelques améliorations possibles pour le site :

* Les utilisateurs connectés pourraient gérer leurs recettes favorites
* Ajouté les champs suivant à une recette : Le temps de cuisson, le temps de préparation et les nombre de portions

## Comparaison énoncé avec projet final

Le projet final correspond au cahier des charges. Les droits des utilisateurs sont respectés. Les actions possibles sont respectées. Il y a eu quelques modifications nécessaires dans la base de données par rapport au model conceptuel du début.

## Comparaison des plannings

Le planning a été partiellement respecté. J’ai pu finir quelques fonctionnalités en avances et j’ai pris du retard sur d’autres. Au final j’ai pu terminer toutes les fonctionnalités.

# Table des illustrations

[Figure 1 : Schéma de fonctionnalités 4](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449188)

[Figure 2 : Droits des utilisateurs 4](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449189)

[Figure 3 : Différentes interfaces du menu utilisateur 8](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449190)

[Figure 4 : Interface - Accueil 9](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449191)

[Figure 5 : Interface - Connexion 9](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449192)

[Figure 6 : Interface - Inscription 10](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449193)

[Figure 7 : Interface – Consultation de la liste des recettes 10](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449194)

[Figure 8 : Interface - Consultation d'une recette 11](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449195)

[Figure 9 : Interface - édition d’une recette 12](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449196)

[Figure 10 : Interface - Consultation d'un utilisateur 13](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449197)

[Figure 11 : Interface - Modification du mot de passe 13](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449198)

[Figure 12 : Interface - Consultation de la liste des utilisateurs 14](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449199)

[Figure 13 : Interface - Modification d'un utilisateur 14](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449200)

[Figure 14 : Model conceptuel de données 16](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449201)

[Figure 15 : Arborescence du site 18](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449202)

[Figure 16 : Plan de tests n°1 31](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449203)

[Figure 17 : Plan de tests n°2 32](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449204)

[Figure 18 : Rapport de tests n°1 33](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449205)

[Figure 19 : Rapport de tests n°2 34](file:///C:\Users\dricc\Desktop\EasyPHP\data\localweb\myportablefiles\TopRecettes\docs\DocTechnique_CedricDosReis.docx#_Toc419449206)

[Figure 20 : Planning provisoire 35](#_Toc419449207)

[Figure 21 : Planning réel 35](#_Toc419449208)

# Références

<http://php.net>

<http://bootsnipp.com/forms?version=3>

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

<http://www.750g.com/>

<http://www.marmiton.org/>

<http://getbootstrap.com/>

<https://bootswatch.com/>