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

Table des matières

[1. Introduction 2](#_Toc419293471)

[2. Etude d’opportunité 2](#_Toc419293472)

[3. Analyse fonctionnelle 3](#_Toc419293473)

[3.1. Généralité 3](#_Toc419293474)

[3.2. Fonction globales 4](#_Toc419293475)

[3.3. Interfaces 5](#_Toc419293476)

[3.4. Fonctions particulières 12](#_Toc419293477)

[3.5. Eléments de sécurité 12](#_Toc419293478)

[4. Analyse Organique 12](#_Toc419293479)

[4.1. Généralité 12](#_Toc419293480)

[4.1.1. Model conceptuel de données 13](#_Toc419293481)

[4.2. Description des méthodes (algorithmes) de résolution 16](#_Toc419293482)

[4.2.1. Description des actions 16](#_Toc419293483)

[4.2.2. Description des fonctions 18](#_Toc419293484)

[5. Tests 25](#_Toc419293485)

[5.1. Plan de test 25](#_Toc419293486)

[5.2. Rapport de test 26](#_Toc419293487)

[6. Conclusion 27](#_Toc419293488)

[6.1. Améliorations 27](#_Toc419293489)

[6.2. Comparaison énoncé avec projet final 27](#_Toc419293490)

# 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

# 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. Il y a notamment 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.



# Analyse fonctionnelle

## Généralité

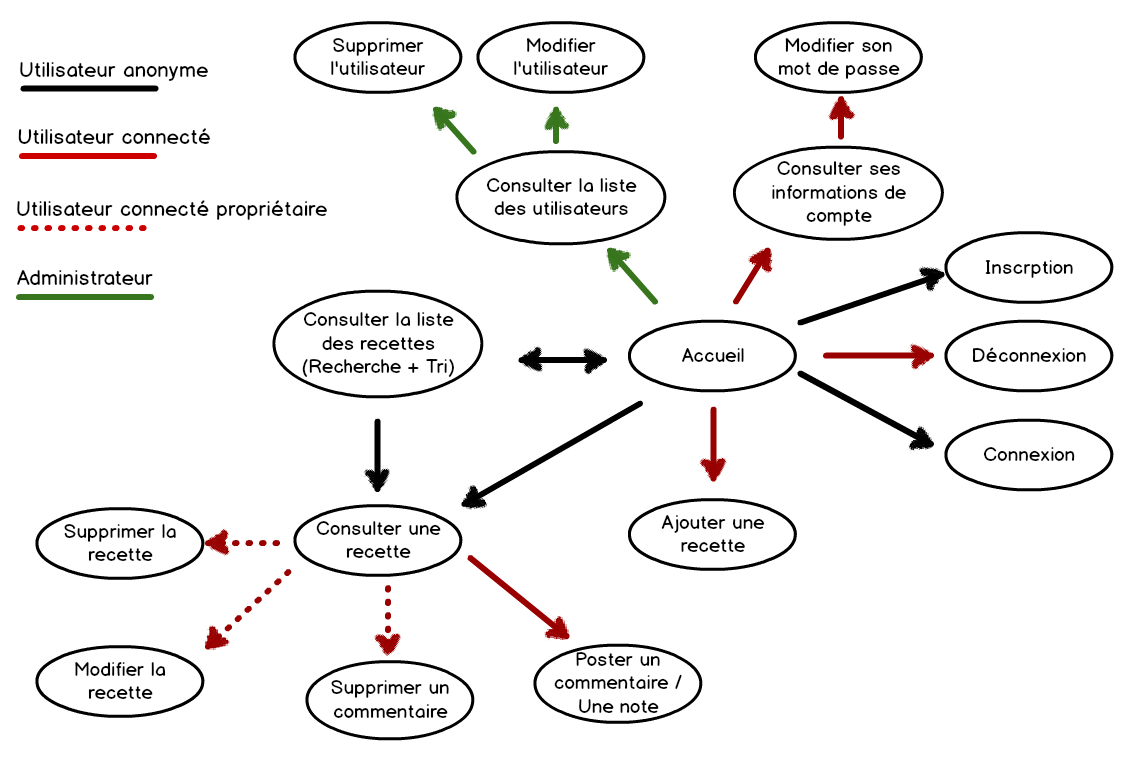


Figure 1 : Schéma de fonctionnalités

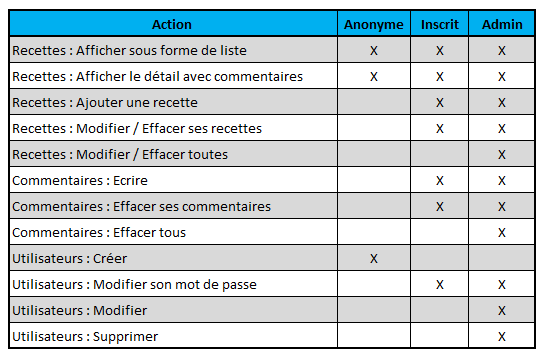


Figure 2 : Droits des utilisateurs

## Fonction globales

Fonctions globales

Connexion au site

Inscription au site

Afficher la liste des recettes

Afficher les détails d’une recette

Ajouter une nouvelle recette

Modifier une recette

Supprimer une recette

Ajouter un commentaire / note à une recette

Supprimer un commentaire / note

Modifier son mot de passe

Modifier un utilisateur

Supprimer un utilisateur

## 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.

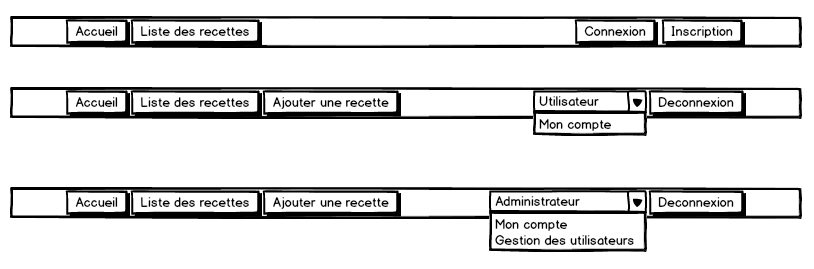
**Le menu change en fonction du type d’utilisateur.**

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 3 : Différentes interfaces du menu utilisateur



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

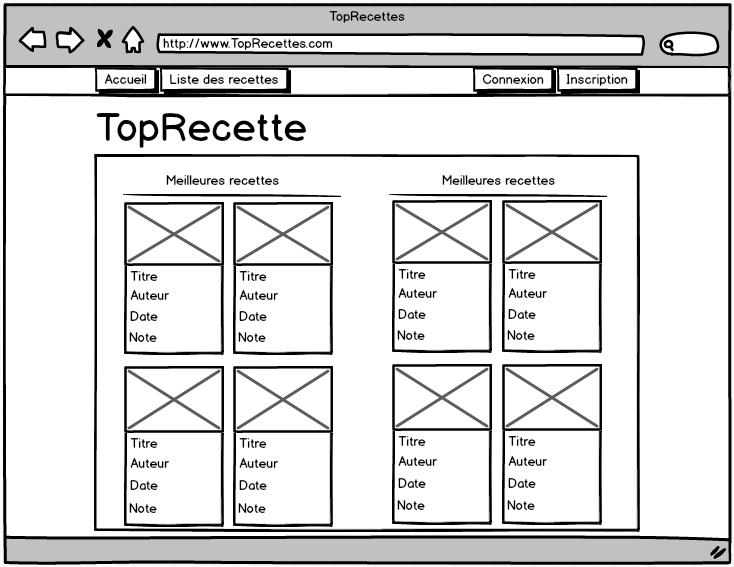


Figure 4 : Interface - Accueil

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

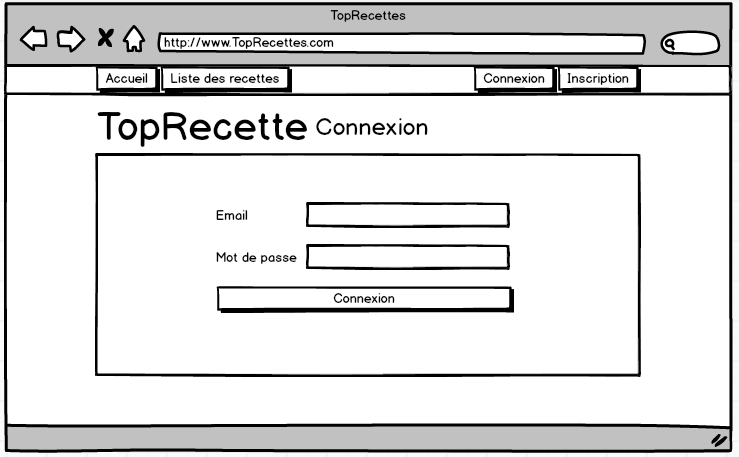


Figure 5 : Interface - Connexion

**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

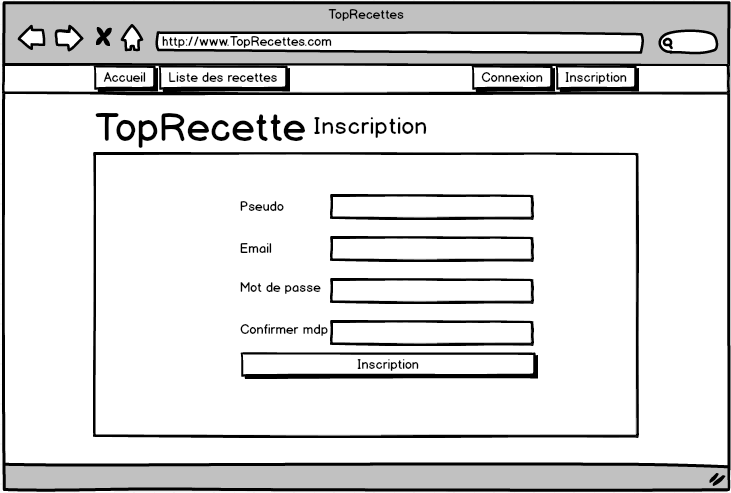


Figure 6 : Interface - Inscription

**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.

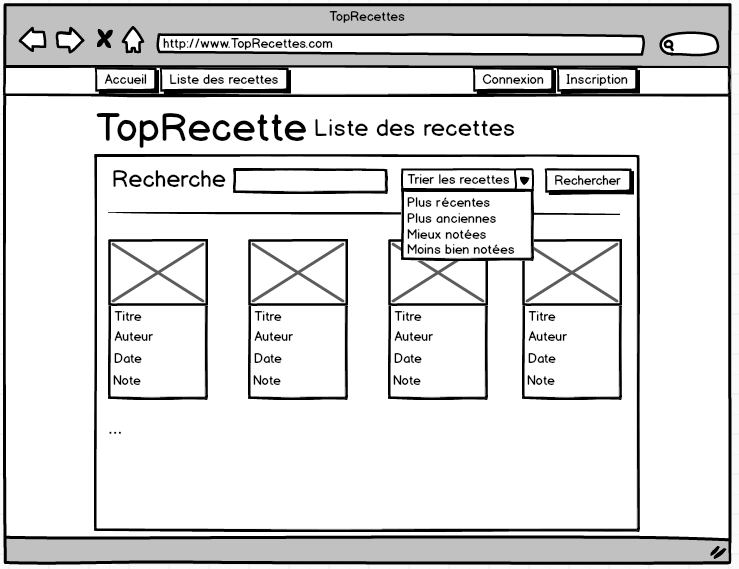
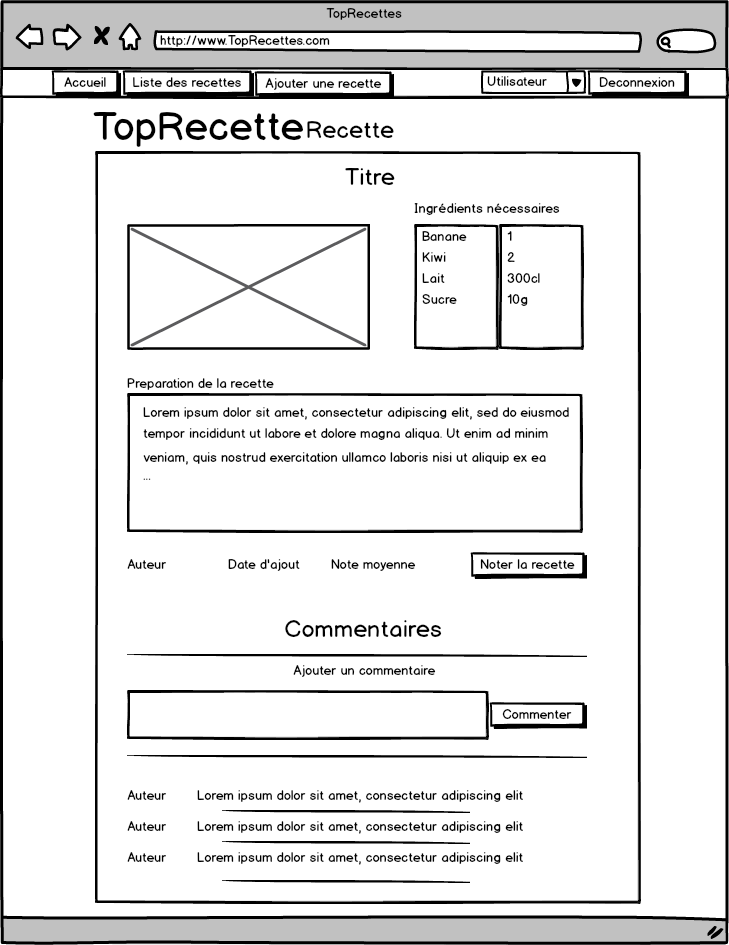


Figure 7 : Interface – Consultation de la liste des recettes

**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 8 : 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.

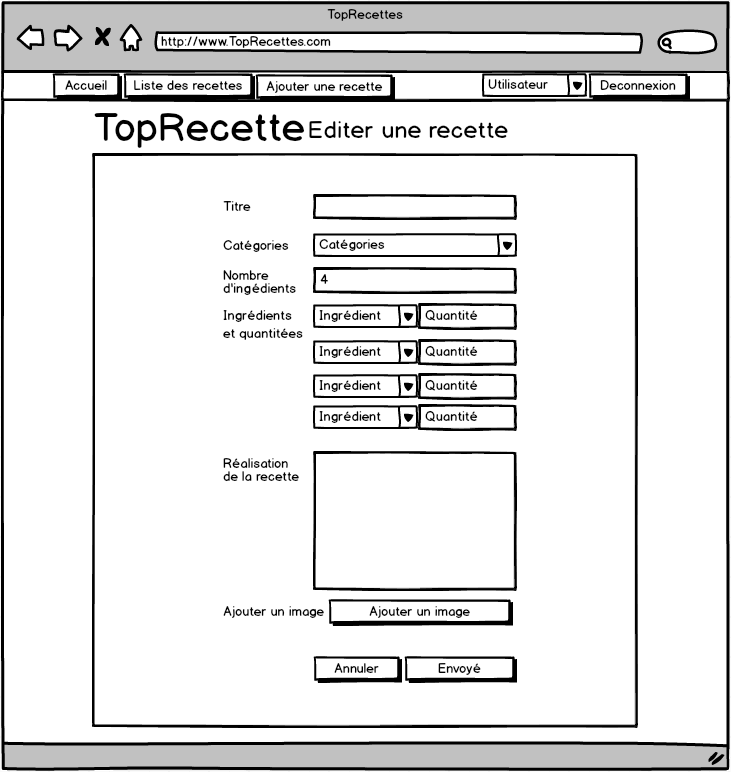
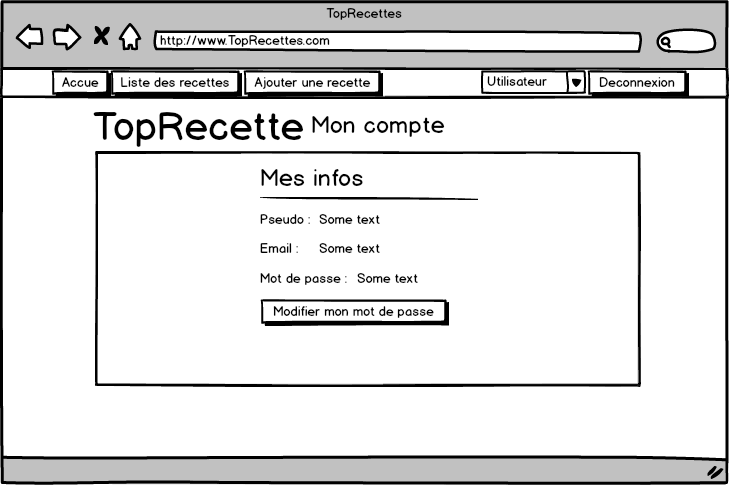


Figure 9 : Interface - édition d’une recette

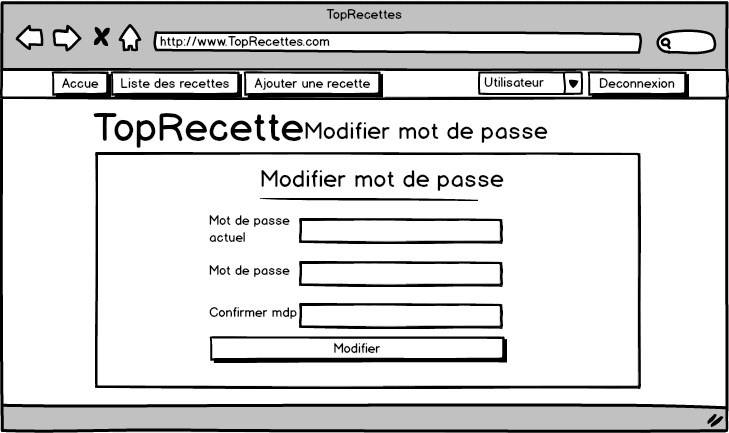
**La page de consultation d’un utilisateur** permet à un utilisateur connecté de consulter ses informations de compte.

Figure 10 : 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 11 : 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.

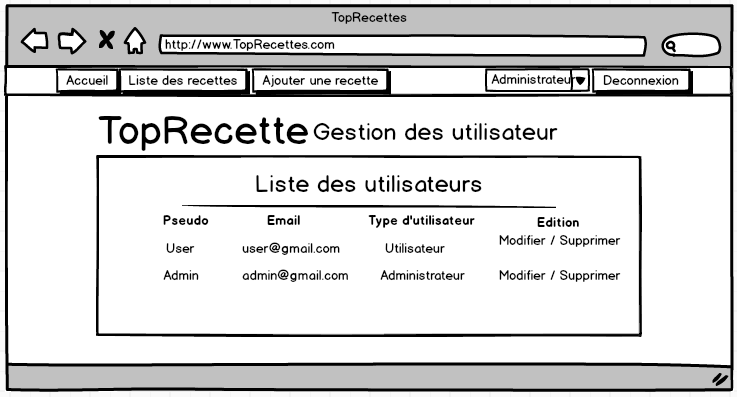


Figure 12 : Interface - Consultation de la liste des utilisateurs

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

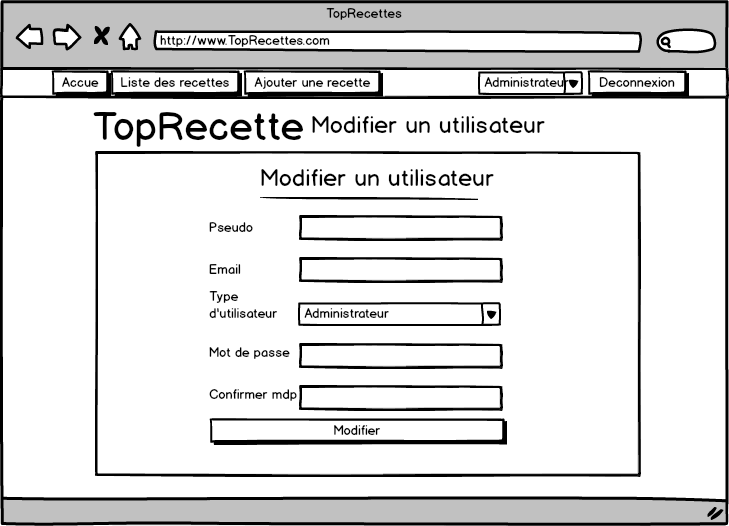


Figure 13 : Interface - Modification d'un utilisateur

## Fonctions particulières

## 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

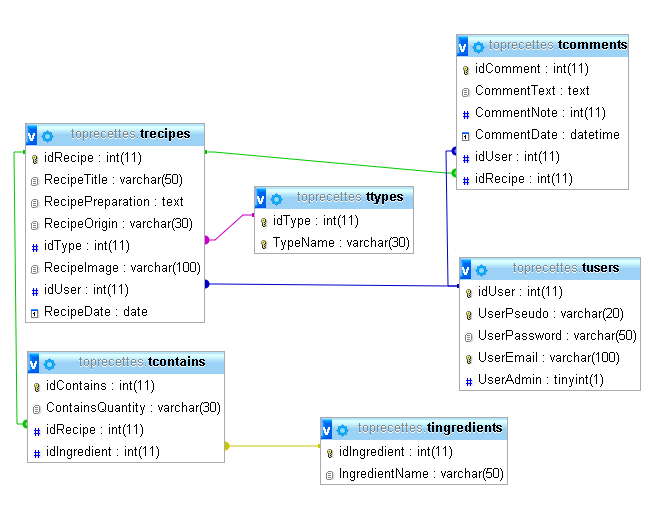


Figure 14 : Model conceptuel de données

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) |

## 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 se termine.

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 se termine. Si les mots de passe sont pareils alors l’adresse email est vérifiée. Si l’adresse email n’est valide alors un message d’erreur s’affiche et l’action se termine. 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 se termine. 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 se termine. 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 au choix possible.

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.

Le titre de la page change en fonction du tri effectué

#### 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 (contenu 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.

La première étape nécessite 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 information 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 enregistrée dans la session de l’utilisateur et l’ajout passe à l’étape 2.

La deuxième étape les ingrédients et leur quantité, la préparation de la recette et une image (l’image n’est pas obligatoire)

#### Supprimer une recette

#### Ajouter un commentaire

#### Supprimer un commentaire

#### Modifier son mot de passe

#### Modifier un utilisateur

#### Supprimer un utilisateur

### Description des fonctions

|  |  |  |
| --- | --- | --- |
| **Fonction** | **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 | Construit une série de cases à cocher (formulaire) basée sur le tableau associatif $liste,  Coche l'ensemble de éléments cochés signalé comme tel dans le tableaus $checked  et ajoute les éventuels attributs |
| 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 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. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fonction** | **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. |

|  |  |  |
| --- | --- | --- |
| **Fonction** | **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. |

|  |  |  |
| --- | --- | --- |
| **Fonction** | **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 |

|  |  |  |
| --- | --- | --- |
| **Fonction** | **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 |

|  |  |  |
| --- | --- | --- |
| **Fonction** | **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. |
| **Fonction** | **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

## Plan de test

* 1. Rapport de test

# Conclusion

## Améliorations

## Comparaison énoncé avec projet final