# Gestion de cocktails

Organisatorisches :

* Projet.zip
  + Fichier de données Donnes.inc.php
    - $Recette : un tableau de recettes de cocktails
    - $Hierarchie : un tableau décrivant une hiérarchie d’aliments
  + Dossier Photos -> photos de cocktails
* Développer en PHP sans base de données
* Rapport au format PDF (4 pages maximum)
* Bien justifier dans ces différentes parties von choix de représentation, de manipulation, d’implémentation

Recettes (tableau PHP) :

* Titre
* Ingrédients séparés par des « | », quantité, unité l’aliment concerné
* Préparation
* Index -> correspond à un tableau d’aliments extrait des ingrédients : chaque aliment -> correspondance dans la hiérarchie des aliments

Hiérarchique des aliments (catégorie / sous-catégorie, tableau PHP) :

* Chaque cellule est indicée par le nom de l’aliment, auquel est associé un tableau avec les liens
  + Super-catégories
  + Sous-catégories (feuille ne possèdent pas de sous-catégorie

# Application à développer

1. **Accès hiérarchique aux recettes à partir de la hiérarchie des aliments**

Naviguer dans la hiérarchie dans la hiérarchie des aliments et de visualiser les resettes utilisant un aliment en tant qu’ingrédient.

Navigation -> sélectionner de plus en plus précis (fruit  agrume  orange).

Afficher le chemin menant à l’aliment courant

Affichage de la recette avec la photo du cocktail

1. **Panier de recettes ou « Mes recettes préférées »**

Initialement vide quand l’utilisateur n’est pas connecté.

Augmente quand l’utilisateur sélectionne des recettes préférées (ajouter cette recette à mes recettes préférées)

Diminuer lorsque l’utilisateur supprime une recette de ses recettes

Est complété par les recettes préférées (« déjà connues » de l’utilisateur quand celui-ci se connect

L’ensemble des recettes préféréesdoit être stocké de façon durable

Lien « Mes recettes préférés » dans l’interface

1. **Identification et données utilisateur**

Utilisateur doit pouvoir se connecter à n’importe quel moment

Connexion n’est pas obligatoire

Si l’utilisateur n’est pas connecté, recettes ne seront pas stockées durablement

Login : login, mot de passe, nom, prénom, sexe, adresse électronique, date de naissance, adresse postale (adresse, code postal, et ville), numéro de téléphone (login et mot de passe obligatoire)

1 fois tout saisies, l’utilisateur pourra s’identifier par login & mot de passe

Pouvoir modifier ses données personnelles à tout moment

Login obligatoirement unique

1. **Interface de recherche de recettes**

Rechercher des recettes à partir d’un ensemble d’aliments qu’il souhaite utiliser et d’un ensemble qu’il ne souhaite pas d’utiliser

Mettre en place une interface la plus conviviale et la plus efficace possible. L’utilisation de JavaScript/AJAX souhaitée. « Ju » -> liste de propositions

Une recherche sur « Jus de fruit » devra retourner des recettes contenant du « Jus de pommes », du « Jus d’orange », etc.

Traiter une recherche approximative (recettes qui ne satisfont pas toutes les contraintes de l’utilisateur. Classer les recettes avec un score de satisfaction

Présenter les résultats de la façon la plus claire possible à l’utilisateur

# Questions

Stocker les logins de façon durable sans base de données

Définition durable

Première connexion affichage de tous les cocktails et ensuite de moins en moins en fonction des filtres ?

# Rapport – Partie

## L’utilisation de la variable SESSION

Elle est utilisée pour la **gestion du panier**. Pour stocker les cocktails préférés de l’utilisateur, l’ID du cocktail concerné est ajoutée dans $\_SESSION[”MesRecettes”].

En outre, elle est utilisée pour le **login de l’utilisateur**. Si l’internaute s’est connecté, la variable $\_SESSION[”utilisateur”] est affectée au nom de login de l’utilisateur. Cette variable est *unset* si l’utilisateur se déconnecte.

## Le panier

La page Panier affiche tous les cocktails dont l’ID se trouve soit dans **$\_SESSION[”MesRecettes”]** ou dans le tableau **json** de l’utilisateur. Le nom de chaque cocktail est affiché et il est cliquable, menant à une page qui détaille sur ce cocktail.

## L’utilisateur

Ses données sont stockées dans le fichier **DonneesUtilisateurs.json**. Ce fichier sert à stocker tous les utilisateurs. Si l’utilisateur n’a pas fourni une valeur (comme son email par exemple), la valeur du champ est une chaine vide (ex. ”email” : ””).

En cliquant sur l’icone qui montre une personne en haut du site, l’utilisateur a la **possibilité** de se connecter, de créer un nouveau compte ou, s’il est déjà connecté, la possibilité de se déconnecter, de modifier ses données ou de supprimer son compte. Pour chaque possibilité, il y a un bouton qui est géré par la fonction compte JavaScript qui elle-même fait une requête POST à la page gestionUtilisateur.php. Cette page soit exécute directement l’option choisie et renvoie sur index, soit elle renvoie l’utilisateur sur une autre page.

**La création d’un compte et la modification des données** utilise le même formulaire qui se trouve dans compte-detail.php. Il est complètement géré par JavaScript. Dans le cas d’une modification, JavaScript remplit le formulaire par les données qui sont stockées dans le tableau json de l’utilisateur. Ceci est réalisé grâce à Ajax (première fonction dans le fichier javascript.js) qui fait une requête POST à la page gestionUtilisateur.php. Une requête avec Ajax est aussi faite pour savoir si le login est encore disponible.

Le **login** est réalisé par le formulaire de la page identification\_form.php. Il est aussi entièrement géré par Javascript. Une requête au serveur assure que la combinaison de login et mot de passe (mdp) est valide.

La page qui **détaille** sur un cocktail offre la fonctionnalité d’ajouter ce cocktail au panier ou de l’enlever. Cette fonctionnalité est utilisable en cliquant sur le cœur. Quand l’utilisateur a ajouté un cocktail à ses cocktail préférés (mis au panier) le cœur devient rouge. La page qui modifie la préférence s’appelle modifierPreference.php. Elle reçoit une requête POST de la fonction JavaScript modifierPreference qui se trouve dans le fichier javascript.js.