

# RECIPE FINDER

Université Paris Diderot

Décembre 2015

PIERRE-CHARLES Nicolas  
LI Luting  
COTTIN Kevin



# Table des matières

1.Introduction.....	4
1.1.Description.....	4
1.2.Répartition des tâches.....	4
1.3.Structure du projet.....	4
1.4.Bibliothèques utilisées.....	4
1.5.Contraintes.....	5
1.5.1.Contraintes de l'environnement.....	5
1.5.2.Contraintes de développement.....	5
1.5.3.Contraintes physiques.....	5
1.6.Problèmes rencontrés.....	5
1.6.1.Les données.....	5
1.6.2.Le moteur de recommandation.....	5
1.6.3.Interface.....	6
2.Réalisation.....	7
2.1.Choix techniques.....	7
2.2.Robot d'indexation.....	7
2.2.1.Récupération des données.....	7
2.2.2.Traitement des données.....	8
2.2.3.Stockage des données.....	8
2.3.Le moteur de recommandation.....	8
2.3.1.Description.....	8
2.3.2.Algorithmes.....	8
2.3.3.Analyses.....	9
2.4.L'interface utilisateur.....	9
3.Résultats.....	11
4.Conclusion.....	12
5.Pour aller plus loin.....	13
5.1.Extension.....	13
6.Annexe.....	14
6.1.Diagramme de l'application.....	14

# 1. Introduction

## 1.1. Description

Le but du projet est de faire un moteur de recommandation de recettes de cuisine. La recommandation intervient lorsqu'un utilisateur est authentifié et lance une recherche pour trouver une recette.

L'application se base sur :

- ▶ les ingrédients qu'il veut et ne veut pas dans sa recette
- ▶ le type de recette qu'il cherche (entrée, plat, dessert ou autre)
- ▶ les recettes qu'il a bien noté
- ▶ les recettes favorites de l'utilisateur

L'application permet d'obtenir une liste de recettes.

## 1.2. Répartition des tâches

Tâche	Description	Personne à charge
Robot d'indexation	Récupérer les recettes de cuisine et sur un ou plusieurs site internet automatiquement	LI Luting PIERRE-CHARLES Nicolas
Moteur de recommandation	Moteur de recommandation des recettes de cuisine	COTTIN Kevin
Interface	Serveur HTTP, interface graphique, transfert des données, structure de l'application	PIERRE-CHARLES Nicolas
Structure de l'application	Création de l'architecture de l'application	PIERRE-CHARLES Nicolas
Stockage des données	Création de la base de données	PIERRE-CHARLES Nicolas COTTIN Kevin

## 1.3. Structure du projet

Il est composé de plusieurs dossiers :

- ▶ **doc/** : la documentation technique du projet
- ▶ **src/** : les sources du projet
- ▶ **tst/** : les tests du projet
- ▶ **Makefile** : compilation du programme
- ▶ **start-server.sh** : raccourci pour lancer le serveur tout en restant à la racine

Les détails sur le dossier **src** sont :

- ▶ **server.py** : point d'entrée de l'application qui permet de lancer le serveur HTTP et d'initialiser la base de données
- ▶ **config.txt** : fichier de configuration de l'application
- ▶ **www/** : la partie interface utilisateur
- ▶ **db/** : la base de données et les scripts SQL utilisées
- ▶ **cgi\_bin/** : le robot d'indexation, le moteur de recommandation et les scripts pour interfacier les modules avec l'interface graphique
- ▶ **cgi\_bin/db/** : scripts de gestion de la base de données

## 1.4. Bibliothèques utilisées

Les bibliothèques utilisées pour la majorité du projet :

- bibliothèque standard de python : réaliser le serveur HTTP et la plupart des scripts

Les bibliothèques utilisées pour le serveur :

- **ConfigParser** : parser le fichier de configuration

Les bibliothèques utilisées pour la récupération des données sont :

- **urllib2** : obtenir les pages web
- **BeautifulSoup** : parser facilement les pages HTML
- **sqlite3** : création et utilisation de la base de données

Les bibliothèques utilisées pour l'interface graphique sont :

- **Bootstrap** : amélioration du rendu visuel

## 1.5. Contraintes

### 1.5.1. Contraintes de l'environnement

Aucun système décentralisé a été mis en place. De nombreuses contraintes environnementales découlent de ce choix.

Ensuite, j'ai fait un moteur de recherche et non un moteur de recommandation.

### **1.6.3.Interface**

De même, pour le serveur HTTP, nous avons mis en place un serveur Apache au début. Puis nous avons préféré écrire un script python qui fait juste le nécessaire en utilisant une classe fournie par la bibliothèque standard.

C'est un compromis au niveau de la sécurité que nous avons choisis de faire pour faciliter le développement.

Également, au niveau de l'interface graphique, le fait d'avoir choisi d'utiliser un serveur HTTP a facilité la construction d'un rendu harmonieux mais l'utilisation du javascript pour faire communiquer le client avec le serveur a compliqué la tâche au final.

## 2. Réalisation

### 2.1. Choix techniques

Le projet est sur GitHub à l'adresse <https://github.com/chipp972/RecipeFinder>.

Le langage choisi est python pour sa simplicité d'écriture et les bibliothèques qui permettent d'exploiter les données facilement.

### 2.2. Robot d'indexation

#### 2.2.1. Récupération des données

Pour récupérer les recettes qu'on a besoin pour la base de donnée, l'idée est de partir d'un site web de base (<http://marmiton.org>), et avec l'url de ce site on utilise un parser qui analyse tous les liens sur le page. Pour traiter ces liens on dispose deux listes, un pour les liens déjà traités et l'autre pour les liens à traiter.

ALGO web\_crawler(enter\_url: url, limit: entier):

DEBUT

    base := enter\_url

    url\_to\_treat := [enter\_url]

    url\_treated := liste vide

    recipe\_found := 0

    tant que len(url\_to\_treat) > 0 et recipe\_found < limit alors:

        res = get\_recipe(dernier élément du url\_to\_treat, base)

        #si la clé 'name' appartient dans le dictionnaire « res », ça vaut dire que le page qu'on a trouvé concerne bien une recette

        si la clé 'name' appartient dans res ALORS:

            ajouter des requêtes concernant aux recettes et ingrédients

            recipe\_found++

        fin si

        ajouter res['url'] dans la liste url\_treated

        pour tous les urls i qu'on trouve sur la page courant, si il est bien une recette:

            si i n'est pas encore traité et i n'est pas dans la liste url\_to\_treat:

                ajoute i dans url\_to\_treat

        fin si

    fin pour

fin tant que

FIN

#dans cet algorithme on a utiliser le librairie BeautifulSoup pour parser le page web

ALGO get\_recipe(url: url, base: url):

DEBUT

    parser le page web avec url donnée en argument

    \_urls = []

    pour tous les élément de cet page url

        si c'est une recette alors

            ajouter l'url dans \_urls

        fin si

    fin pour

    ingr\_list = []

    pour tous les i qui est une liste d'ingrédients

        séparer les ingrédients dans une liste

filtrer la liste d'ingrédients  
fin pour  
récupère le titre  
récupère le type  
récupère l'image

retourner tous les information sous forme d'un dictionnaire

FIN

### 2.2.2. Traitement des données

Pour le traitement des données, il faut nettoyer la liste d'ingrédients par un filtre qui enlève les symboles inutiles, les quantités etc afin d'obtenir une liste d'ingrédients la plus uniforme possible.

### 2.2.3. Stockage des données

Il enregistre les recettes et les ingrédients dans la base de données lorsqu'il a terminé l'indexation.

## 2.3. Le moteur de recommandation

### 2.3.1. Description

Le moteur de recommandation recherche les recettes dans la base de données puis les traite pour recommander les meilleurs recettes. Pour traiter les données, il se base sur les précédentes recherches faites par l'utilisateur.

Il prend en entrées :

- l'identifiant de l'utilisateur qui fait la recherche
- la liste des types de recettes recherchés
- la liste des identifiants des ingrédients voulus dans la recette
- la liste des identifiants des ingrédients refusés dans la recette

Il retourne une liste d'identifiant des recettes. Cette liste est trié suivant la meilleur recommandation proposée.

### 2.3.2. Algorithmes

#### Algorithme Recommandation

(entier : identifiant de l'utilisateur,  
liste d'entier : identifiants des type de recettes,  
liste d'entier : identifiants des ingrédients voulus dans la recettes,  
liste d'entier : identifiants des ingrédients refusés dans la recettes)  
recettes = Récupérer les recettes associées à au moins un ingrédient voulu par l'utilisateur  
# On se retrouve avec une liste de tuples avec des doublons sur sur les noms de recettes  
dictionnaireRecettes = Créer le dictionnaire de recettes  
# On se retrouve avec un dictionnaire bien formé de recettes  
ingrédients = Récupérer les ingrédients déjà recherchées par l'utilisateur  
matrice = CalculerMatriceDesPoids  
recettes = Trier les recettes par les poids(matrice)  
Retourner une liste d'identifiants ordonnées des recettes

**Fin algorithme**

#### Algorithme CalculerMatriceDesPoids<sup>3</sup>

entier poidFort = 2, entier poidMoyen = 1, entier poidFaible = -2  
Pour toutes recettes recette dans dictionnaireRecettes faire  
    Pour tout ingrédients voulus faire  
        poids dans la recette = poidFort  
    Pour tout ingrédients voulus dans le passé faire  
        poids dans la recette = poidMoyen  
    Pour tout ingrédients voulus faire

<sup>3</sup> L'algorithme pour calculer la matrice n'est pas séparer dans le code, elle l'est ici pour faciliter la lecture



poids dans la recette = **poidFaible**  
poids total = somme des poids des ingrédients dans la recette

## Fin algorithme

### 2.3.3. Analyses

La complexité en temps de l'algorithme *CalculerMatriceDesPoids* est :

$O(\#recettes * \#ingrédientsVoulus$   
+  $\#recettes * \#ingrédientsVoulusDansLePassé$   
+  $\#recettes * \#ingrédientsRefusés)$

soit  $O(\#recettes * (\#ingrédientsVoulus + \#ingrédientsVoulusDansLePassé + \#ingrédientsVoulusDansLePassé))$

soit  $\#ingrParam = \#ingrédientsVoulus + \#ingrédientsVoulusDansLePassé + \#ingrédientsVoulusDansLePassé$   
soit  **$O(\#recettes * \#ingrParam)$**

Malgré la complexité élevée, l'algorithme est fiable dans la mesure où l'utilisateur a une recette qui correspond au mieux à sa recherche en prenant en compte ses recherches antérieures.

## 2.4. L'interface utilisateur

L'interface utilisateur du Recipe Finder est accessible depuis un navigateur web grâce au serveur HTTP qui permet d'accéder à l'application.

On y rentre son e-mail pour qu'on puisse associer la recommandation au bon utilisateur, le type de la recette (entrée, plat principal, dessert ou autre), les ingrédients que l'on cherche à avoir dans la recette et ceux que l'on ne veut pas trouver dans la recette.

On peut également indiquer le type de recommandation souhaité pour choisir l'algorithme de recommandation utilisé (non implémenté pour le moment).

The screenshot shows a web browser window with the address bar set to localhost:8080. The page title is "Recipe Finder". The main content area has a heading "Welcome in the Recipe Finder !!!". Below the heading is a form with the following fields:

- Email:** A text input field with the placeholder "Enter email".
- Type:** A dropdown menu currently showing "dessert".
- Ingredients Liked:** A text area containing the text "'vache qui rit' ou equivalent", "ail", "amandes en poudre", and "betteraves rouges".
- Ingredients Disliked:** A text area containing the same text as the "Liked" field.

Below the form fields are three radio buttons for recommendation type:

- ☐ Recommendation by Favorites
- ☒ Recommendation by Mark
- ☐ Recommendation by Click

A large green "Submit" button is located below the radio buttons. At the bottom of the page, there is a footer that reads: "Data Mining Project 2015 - COTTIN Kevin - LI Luting - PIERRE-CHARLES Nicolas".

Après l'envoi du formulaire on obtient une liste de recettes dans l'ordre calculé par le moteur de recommandation. On y trouve les ingrédients qu'elles contiennent, une image si il y en avait une sur le site d'origine et si on clique dessus, on est redirigé vers le site de la recette. On peut également y trouver une liste des


opinions des utilisateurs. Cependant le formulaire pour les ajouter n'est pas encore implémenté.

### 3. Résultats

Recipe Finder

localhost:8080/cgi\_bin/form.py

Tarte au chocolat noir meringuee : Recette de Tarte au chocolat noir meringuee - Marmiton



Ingredients

oeufs

pate sablee

creme fraiche

chocolat noir corse


beurre

gros oeuf

peu parfume

Opinions

Tarte aux fraises : Recette de Tarte aux fraises - Marmiton



Ingredients

citron

fraises

pate sablee

## 4. Conclusion

L'application pour le moteur de recommandation est fonctionnelle.

Le robot d'indexation permet de récupérer des recettes sur Marmiton et de les utiliser dans des recommandations de recettes. Il ne permet pas de récupérer des recettes sur plusieurs sites différents, correcteur<sup>4</sup> des ingrédients.

Le moteur de recommandation prend en compte les précédentes recherches d'un utilisateur. Il ne prend pas en compte l'avis de tous les utilisateurs, les précédentes ingrédients recherchés sur des recherches similaires, les favoris des utilisateurs.

L'interface graphique affiche la liste des recettes recommandées, leurs ingrédients et les avis. Il affiche les favoris. Il ne permet pas d'ajouter des avis, d'ajouter en favori une recette.

---

<sup>4</sup> Les ingrédients qui sont les mêmes mais avec une syntaxe différente

## 5. Pour aller plus loin

### 5.1. Extension

Les extensions possibles pour l'application sont :

- ▶ l'application n'étant qu'un prototype, on peut déjà penser à refaire la structure avec un framework python adapté comme Django, par exemple
- ▶ une meilleure gestion des utilisateurs
- ▶ la gestion des mots de passe
- ▶ une base de donnée plus adaptée à un trafic d'application web
- ▶ ajouter une persistance de l'authentification des utilisateurs, au moyen de cookies par exemple
- ▶ ajouter les tests automatiques

Les extensions possibles pour le robot d'indexation sont :

- ▶ récupérer plus d'informations relatives aux utilisateurs (ustensiles disponibles, ingrédients favoris...) afin d'affiner leurs profils et donc les recommandations
- ▶ ajouter les quantités pour chaque ingrédient
- ▶ ajouter le support de plus de sites de recettes (seul marmiton est intégré pour le moment)
- ▶ support de plusieurs langues

Les extensions possibles pour le moteur de recommandation sont :

- ▶ on pourrait faire une enquête lorsque l'utilisateur crée un compte afin de connaître ses goûts tout de suite et d'avoir une recommandation moins hasardeuse lorsque l'utilisateur est nouveau
- ▶ ajouter les images des recettes dans le calcul des statistiques
- ▶ déterminer les recettes jamais choisi pour retourner d'autres recettes à la place

## 6. Annexe

### 6.1. Diagramme de l'application

