

DelfMeals - Spécifications Techniques

1. Résumé du projet

DelfMeals est un générateur de recettes hebdomadaires personnalisé, développé en Python pour créer des menus équilibrés basés sur des recettes familiales.

Objectifs

- Génération automatique de menus hebdomadaires équilibrés
- Gestion d'une base de données personnalisée de recettes
- Création automatique de listes de courses
- Interface web pour une utilisation familiale

Technologies

- **Prototype** : Python avec Jupyter Notebook
 - **Base de données** : SQLite
 - **Interface web** : Flask
 - **Hébergement** : PythonAnywhere
 - **OCR** : pytesseract pour numériser les recettes papier
-

2. Structure des données

Modèle de recette

python

```
{
  "Recette": "Nom de la recette",
  "Type de plat": "Plats",
  "Origine": "Cuisine française", # Facultatif
  "Catégorie": "Viande rouge",
  "Nombre de personnes": 4,
  "Temps de préparation": 45, # En minutes
  "Difficulté": "Normal",
  "Saison": ["Toutes saisons"], # Liste pour multi-saisons
  "Préparation": [
    "Étape 1 : ...",
    "Étape 2 : ..."
  ],
  "Ingrédients": [
    {"nom": "ingrédient", "quantité": "500", "unité": "g"}
  ],
  "Note": "Commentaires personnels" # Facultatif
}
```

Champs avec valeurs prédéfinies

Type de plat :

- Apéro, Entrées, Plats, Desserts, Boissons, Brunch et petit déj

Origine :

- Cuisine française, Cuisine italienne, Cuisine méditerranéenne, Cuisine indienne
- Cuisine asiatique, Cuisine américaine, Cuisine du Maghreb, Autre

Catégorie :

- Viande rouge, Volaille, Poisson, Végétarien

Difficulté :

- **Rapide** : < 30 min, ustensiles minimaux
- **Normal** : 30-60 min, niveau habituel
- **Élaboré** : > 60 min ou techniques spéciales

Saison :

- Toutes saisons, Hiver, Printemps, Été, Automne
- Support multi-saisons (ex: ratatouille = Été + Automne)

3. Générateur de menus hebdomadaires

Règles de génération (14 plats)

Contraintes temporelles :

- **Midi** : Uniquement recettes "Rapide" (< 30 min)
- **Soirs de semaine** : Rapide ou Normal
- **Weekend** : Toutes difficultés (max 2 élaborés)

Contraintes saisonnières :

- Utiliser uniquement les recettes "Toutes saisons" ou de la saison courante
- Gestion des périodes de transition (mi-mai = Printemps + Été)

Équilibrage nutritionnel :

- **Viande** : Exactement 2 fois/semaine (alternance rouge/volaille)
- **Cuisine américaine** : Maximum 1 fois/semaine
- **Surpondération x2** : Cuisine asiatique et méditerranéenne

Format de sortie

```
python

menu = {
    "Lundi": {"midi": recette_obj, "soir": recette_obj},
    "Mardi": {"midi": recette_obj, "soir": recette_obj},
    # ... pour toute la semaine
}
```

Liste de courses automatique

Agrégation automatique des ingrédients avec gestion intelligente des unités.

4. Fonctionnalités principales

4.1 Recherche de recettes

Critères de recherche :

- Nom de recette (recherche textuelle)
- Type de plat, Origine, Catégorie
- Difficulté, Saison

4.2 Gestion des recettes

- **Ajout** : Formulaire de saisie complet
- **Modification** : Édition de recettes existantes
- **Suppression** : Avec confirmation et sauvegarde

4.3 Administration

- **Import Excel** : Format standardisé (une colonne par champ)
- **Export Excel** : Sauvegarde complète de la base
- **Backup automatique** : Avant toute opération critique
- **Validation** : Contrôles de cohérence lors des imports

4.4 Fonctionnalités avancées

- **Historique** : Éviter la répétition de menus récents
 - **OCR** : Numérisation de recettes papier avec pytesseract
 - **Gestion intelligente des ingrédients** : Affichage contextuel des quantités
-

5. Architecture technique

Phase 1 : Prototype (Actuel)

- Développement Python/Jupyter
- Fichiers locaux pour les données
- Tests de la logique métier

Phase 2 : Base de données

- Migration vers SQLite
- Optimisation des requêtes
- Gestion de la persistance

Phase 3 : Interface web

- Application Flask
- Interface utilisateur intuitive
- Déploiement sur PythonAnywhere

Structure des fichiers

```
delfmeals/  
├─ models/          # Structures de données et ORM  
├─ core/            # Logique métier (génération, équilibrage)  
├─ ui/              # Interfaces (Jupyter puis Flask)  
├─ utils/           # OCR, import/export, helpers  
├─ data/            # Base SQLite  
├─ static/          # CSS, JS (phase web)  
└─ templates/       # Templates HTML (phase web)
```

6. Gestion des saisons

Détection automatique

```
python  
  
def detecter_saison_actuelle():  
    mois = datetime.now().month  
    if mois in [12, 1, 2]: return "Hiver"  
    elif mois in [3, 4, 5]: return "Printemps"  
    elif mois in [6, 7, 8]: return "Été"  
    else: return "Automne"
```

Périodes de transition

- Mi-février à mi-mars : Hiver + Printemps
 - Mi-mai à mi-juin : Printemps + Été
 - Mi-août à mi-septembre : Été + Automne
 - Mi-novembre à mi-décembre : Automne + Hiver
-

7. Logique d'affichage intelligent





Ingrédients

python

```
def afficher_ingredient(ingredient):
    if ingredient['quantité']:
        if ingredient['unité']:
            unites_sans_de = ['pièces', 'pièce', 'tranches', 'tranche']
            if ingredient['unité'] in unites_sans_de:
                return f"{ingredient['quantité']} {ingredient['nom']}"
            else:
                return f"{ingredient['quantité']} {ingredient['unité']} de {ingredient['nom']}"
        else:
            return f"{ingredient['quantité']} {ingredient['nom']}"
    else:
        return ingredient['nom']
```

8. Roadmap de développement

Sprint 1 : Core fonctionnel

-  Structure de données finalisée
-  Prototype générateur de menus
-  Fonction de recherche basique
-  Tests avec jeu de données réduit

Sprint 2 : Optimisations

- Migration SQLite
- Import/export Excel
- Interface utilisateur améliorée
- OCR pour recettes papier

Sprint 3 : Mise en production

- Application Flask complète
 - Interface web responsive
 - Déploiement PythonAnywhere
 - Documentation utilisateur
-

Conclusion

DelfMeals représente un outil personnalisé et pratique pour simplifier la planification des repas familiaux. L'approche progressive (prototype → base de données → web) assure un développement maîtrisé et évolutif.

Le projet combine intelligemment automatisation et personnalisation, offrant un générateur de menus qui respecte les contraintes familiales tout en maintenant l'équilibre nutritionnel et la variété culinaire.