



RAPPORT : APPLICATION DE GESTION DES RECETTES

Mots clés : MongoDB, fonctionnalités recherche et affichage, fonctionnalités CRUD :create, read,update,delete, python tkinter application

Table des matières

RAPPORT : APPLICATION DE GESTION DES RECETTES	1
Chapitre 1	3
1.1 Introduction	3
1.2 Problématique	3
1.3 Objectifs du projet.....	3
Chapitre 2 : Conception de la base de données	4
2.1 Introduction	4
2.2 Analyse des besoins.....	4
2.3 Modélisation de la base de données.....	5
2.3.1 Choix de MongoDB	5
2.3.2 Structure des collections et documents	5
2.4 Justification des choix de conception	6
2.4.1 Organisation des données	6
2.4.2 Choix des types de données	6
2.5 Indexation et optimisation	7
2.6 Contraintes et défis techniques.....	7
2.6.1 Contraintes techniques.....	7
2.7 Conclusion	7
Chapitre 3 : Interfaces utilisateur et fonctionnalités	8
4.1 Introduction	8
Conclusion :	10

Chapitre 1

1.1 Introduction

Dans un monde où les bases de données jouent un rôle central dans le stockage et la gestion de l'information, les solutions NoSQL comme MongoDB sont devenues incontournables pour les applications modernes. Contrairement aux bases de données relationnelles, MongoDB offre une flexibilité et une évolutivité accrues, ce qui la rend idéale pour des projets nécessitant une gestion dynamique des données.

Ce projet vise à concevoir une base de données NoSQL avec MongoDB pour la gestion des recettes de cuisine. L'objectif est de permettre une recherche rapide et efficace des recettes en fonction des ingrédients disponibles, du type de cuisine et du temps de préparation.

1.2 Problématique

Dans un contexte où les applications de gestion culinaire se multiplient, plusieurs défis se posent :

Flexibilité du stockage : Les recettes ne suivent pas une structure rigide, elles contiennent des ingrédients variés, des descriptions détaillées et des instructions de préparation.

Efficacité des recherches : Un utilisateur doit pouvoir trouver une recette en fonction des ingrédients disponibles chez lui.

Performance et évolutivité : La base de données doit supporter un grand nombre de recettes et permettre des recherches rapides.

Les bases de données relationnelles imposent souvent des structures trop rigides pour ce type de données dynamiques. MongoDB, en revanche, offre un modèle de stockage basé sur des documents JSON, permettant une grande souplesse et une indexation efficace pour des recherches rapides.

1.3 Objectifs du projet

Le projet vise à :

- Créer une base de données NoSQL pour stocker les recettes de manière structurée et flexible.

- Développer des fonctionnalités CRUD (Create, Read, Update, Delete) pour gérer les recettes facilement.
- Optimiser la recherche des recettes par nom, type de cuisine, temps de préparation et ingrédients.
- Garantir des performances élevées en utilisant des index sur les champs les plus consultés.
- Assurer une intégration fluide avec une interface graphique développée en python.

Chapitre 2 : Conception de la base de données

2.1 Introduction

La conception de la base de données est une étape essentielle du projet, car elle définit la structure et l'organisation des informations stockées. L'objectif principal est de créer une base de données MongoDB capable de gérer efficacement les recettes de cuisine et d'optimiser la recherche des recettes en fonction de plusieurs critères (ingrédients, type de cuisine, temps de préparation, etc.).

Ce chapitre détaille l'analyse des besoins, la modélisation des données et les choix techniques adoptés pour garantir une structure efficace et évolutive.

2.2 Analyse des besoins

L'application de gestion de recettes s'adresse à des utilisateurs souhaitant stocker et rechercher des recettes de cuisine. Les besoins principaux identifiés sont les suivants :

✅ Stocker et gérer des recettes : Chaque recette doit contenir des informations détaillées (nom, description, ingrédients, instructions, temps de préparation, etc.).

✅ Rechercher des recettes de manière flexible : L'utilisateur doit pouvoir retrouver une recette en fonction de plusieurs critères :

- Par nom
- Par ingrédient

- Par type de cuisine (algérienne, marocaine)

✅ Assurer une gestion efficace des recettes :

Ajouter, modifier et supprimer des recettes

Optimiser l'accès aux données avec un système d'indexation

Utiliser MongoDB pour stocker les recettes sous forme de documents JSON.

Assurer une recherche rapide avec des index sur les champs principaux.

Concevoir une structure évolutive, permettant d'ajouter de nouvelles fonctionnalités (ex : stockage d'images, filtres avancés, etc.).

Assurer une intégration fluide avec l'interface en PyQt5.

2.3 Modélisation de la base de données

2.3.1 Choix de MongoDB

MongoDB est une base de données NoSQL qui offre plusieurs avantages pour ce projet :

Stockage flexible : Pas besoin d'un schéma rigide, chaque document JSON peut évoluer selon les besoins.

Performance élevée : Grâce aux index et à l'absence de jointures complexes, les recherches sont plus rapides.

Scalabilité : Possibilité d'augmenter la capacité de stockage sans restructurer la base.

2.3.2 Structure des collections et documents

Dans ce projet, nous avons identifié une seule collection principale :

Collection : recipes (Recettes)

Stocke toutes les informations sur une recette. Chaque recette est représentée sous forme d'un document JSON :

```
{
  "_id": ObjectId("615c1b3f4f1a2d6a12e3a9c4"),
  "nom": "Couscous algérien",
  "ingredients": [
    { "Semoule",
      "Viande ;
      ....
    },
    "type": "Algérienne",
    "temps_preparation": 120 min,
  }
```

✅ Explication des champs :

nom : Nom de la recette.

ingredients : Liste des ingrédients.

type_cuisine : Permet de classer les recettes (ex : Algérienne, Italienne, Marocaine...).

temps_preparation : Temps nécessaire pour préparer la recette (en minutes).

2.4 Justification des choix de conception

2.4.1 Organisation des données

Les recettes sont stockées dans une seule collection (recipes), ce qui évite la fragmentation des données.

Les ingrédients sont stockés sous forme de tableau imbriqué, permettant d'accéder rapidement aux informations sans jointures complexes.

2.4.2 Choix des types de données

_id : Identifiant unique MongoDB (ObjectId).

Nom, ingredients : Tableau d'objets JSON

temps_preparation : Stocké en minutes pour simplifier les comparaisons et filtres.

2.5 Indexation et optimisation

L'indexation permet d'accélérer les recherches en évitant de scanner toute la base de données.

Index utilisés

```
db.recipes.create_index([("nom", 1)]) # Recherche par nom
```

```
db.recipes.create_index([("type_cuisine", 1)]) # Filtrage par type de cuisine
```

```
db.recipes.create_index([("ingredients.nom", 1)]) # Recherche par ingrédient
```

✅ Avantages :

Recherche plus rapide des recettes par nom.

Filtrage efficace par type de cuisine.

Recherche optimisée par ingrédient, permettant de trouver des recettes basées sur ce que l'utilisateur possède.

2.6 Contraintes et défis techniques

2.6.1 Contraintes techniques

Stockage structuré : Éviter les redondances et organiser efficacement les informations.

Performance des requêtes : Optimiser les index et la structure des documents.

Évolutivité : Permettre l'ajout futur de fonctionnalités

2.7 Conclusion

Ce chapitre a détaillé la conception de la base de données en se basant sur l'analyse des besoins et des contraintes techniques. Nous avons défini une structure optimisée avec MongoDB, permettant un stockage efficace des recettes et une recherche rapide grâce à l'indexation.

Dans le prochain chapitre, nous passerons à l'implémentation de cette base de données et au développement des fonctionnalités CRUD en Python avec PyQt5.

Chapitre 3 : Interfaces utilisateur et fonctionnalités

4.1 Introduction

L'interface utilisateur joue un rôle essentiel dans l'expérience d'utilisation de l'application. Pour ce projet, nous avons utilisé PyQt5 afin de concevoir une interface graphique intuitive et fluide permettant d'interagir avec la base de données MongoDB. Ce chapitre présente les différentes interfaces développées ainsi que les fonctionnalités principales associées.

Main :

	Name	Type	Time (mins)
1	Algerian Chicken Tagine	algerian	1 hour 30 minutes
2	Couscous Royale	Algerian	1h
3	Moroccan Chicken Tagine	Moroccan	90
4	Chakchouka	algerian	1h
5	Lamb Tagine with Prunes	moroccan	1h20min

Add recipe :

Recipes Application

Recipe Name:

Mhadjeb

Ingredients:

2 cups semolina

1 cup flour

1 teaspoon salt

1 cup warm water

3 tomatoes, grated

1 onion, finely chopped

2 garlic cloves, minced

Type (Algerian, Moroccan):

ALGERIAN

Preparation Time (mins):

45 min

Add Recipe

View All

Delete Recipe

View Details

Search:

Search

	Name	Type	Time (mins)
1	Algerian Chicken Tagine	algerian	1 hour 30 minutes
2	Couscous Royale	Algerian	1h
3	Moroccan Chicken Tagine	Moroccan	90
4	Chakchouka	algerian	1h
5	Lamb Tagine with Prunes	moroccan	1h20min

Delete recipe

Recipes Application

Recipe Name:

Ingredients:

Type (Algerian, Moroccan):

Preparation Time (mins):

Add Recipe

View All

Delete Recipe

View Details

Search:

Search

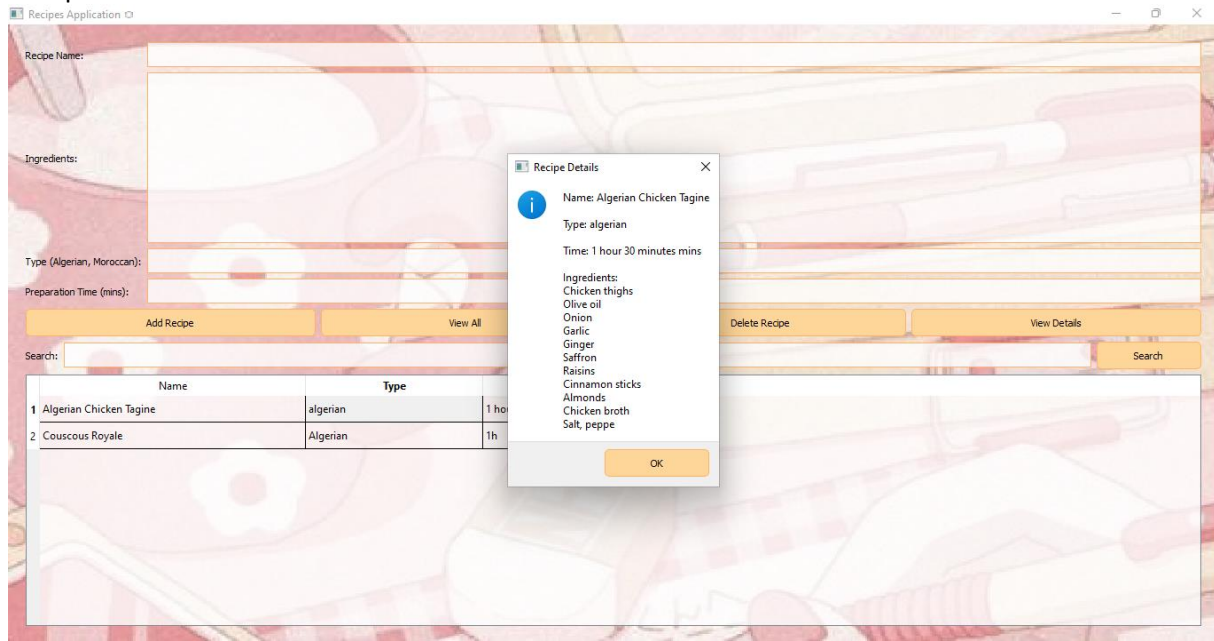
Success

Recipe deleted!

OK

	Name	Type	Time (mins)
1	Algerian Chicken Tagine	algerian	1 hour 30 minutes
2	Couscous Royale	Algerian	1h

Recipe details :



Conclusion :

L'objectif de ce projet était de concevoir et développer une application de gestion de recettes en utilisant une base de données NoSQL (MongoDB) et une interface utilisateur en PyQt5. Grâce à une méthodologie bien définie, nous avons pu structurer et implémenter un système efficace permettant aux utilisateurs de stocker, rechercher, modifier et supprimer des recettes de manière fluide et intuitive.