

Miskatonic

Générateur de Quiz

HUGO BABIN | NATHALIE BEDIEE | 10/2025



Table des matières

1.	Introduction.....	1
2.	Architecture du projet	1
3.	Flux principaux.....	2
3.1.	processus ETL.....	2
3.2.	Authentification.....	3
3.2.1.	MCD base de donnée Utilisateurs :	3
3.2.2.	Diagramme d'authentification	3
3.3.	Génération Quizz	4
3.3.1.	Format Json des questions	4
3.3.2.	Diagramme de génération des quizz	4
4.	Conclusion.....	5

I. Introduction

Le projet **Miskatonic** vise à fournir aux enseignants un outil web pour **créer, importer et générer des quiz** à partir d'une base de questions centralisée.

L'application combine deux technologies complémentaires :

- **FastAPI** pour les routes, la logique métier et la communication avec les bases de données.
- **Flask** pour le front-end, chargé du rendu des pages HTML avec **Jinja2**.

Les données sont réparties dans deux systèmes :

- **MongoDB** pour les questions et les quiz.
- **SQLite** pour les utilisateurs, rôles et journaux d'authentification.

Un **pipeline ETL** (Extract – Transform – Load) permet l'importation automatisée de fichiers CSV, la validation des données et leur intégration en base Mongo.

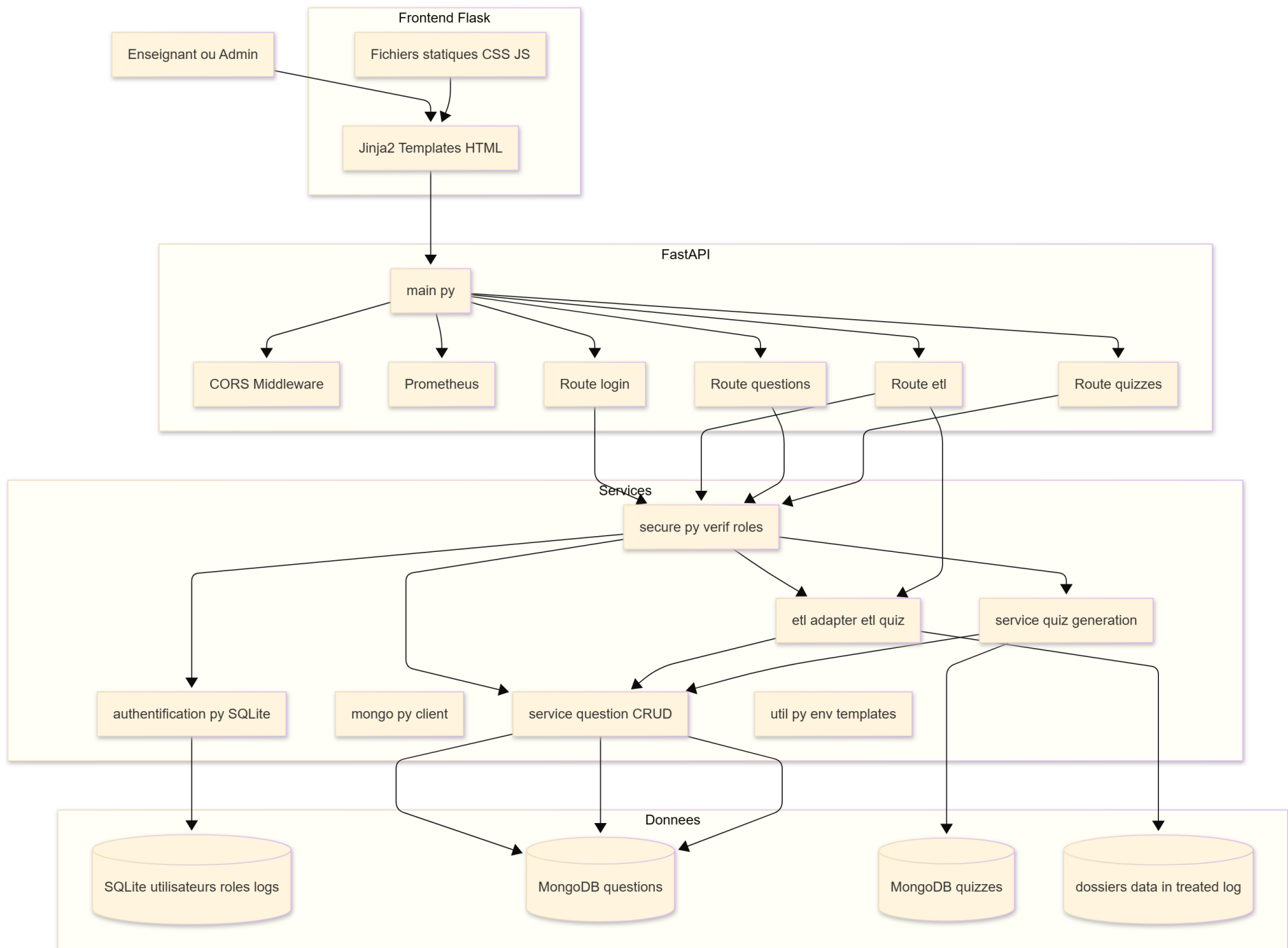
L'ensemble est conteneurisé avec **Docker**, et instrumenté avec **Prometheus** pour le suivi des performances.

2. Architecture du projet

Le schéma suivant illustre l'organisation générale du système :

l'utilisateur interagit avec le front Flask, qui échange avec l'API FastAPI.

Celle-ci gère la logique métier et la persistance des données dans MongoDB et SQLite.

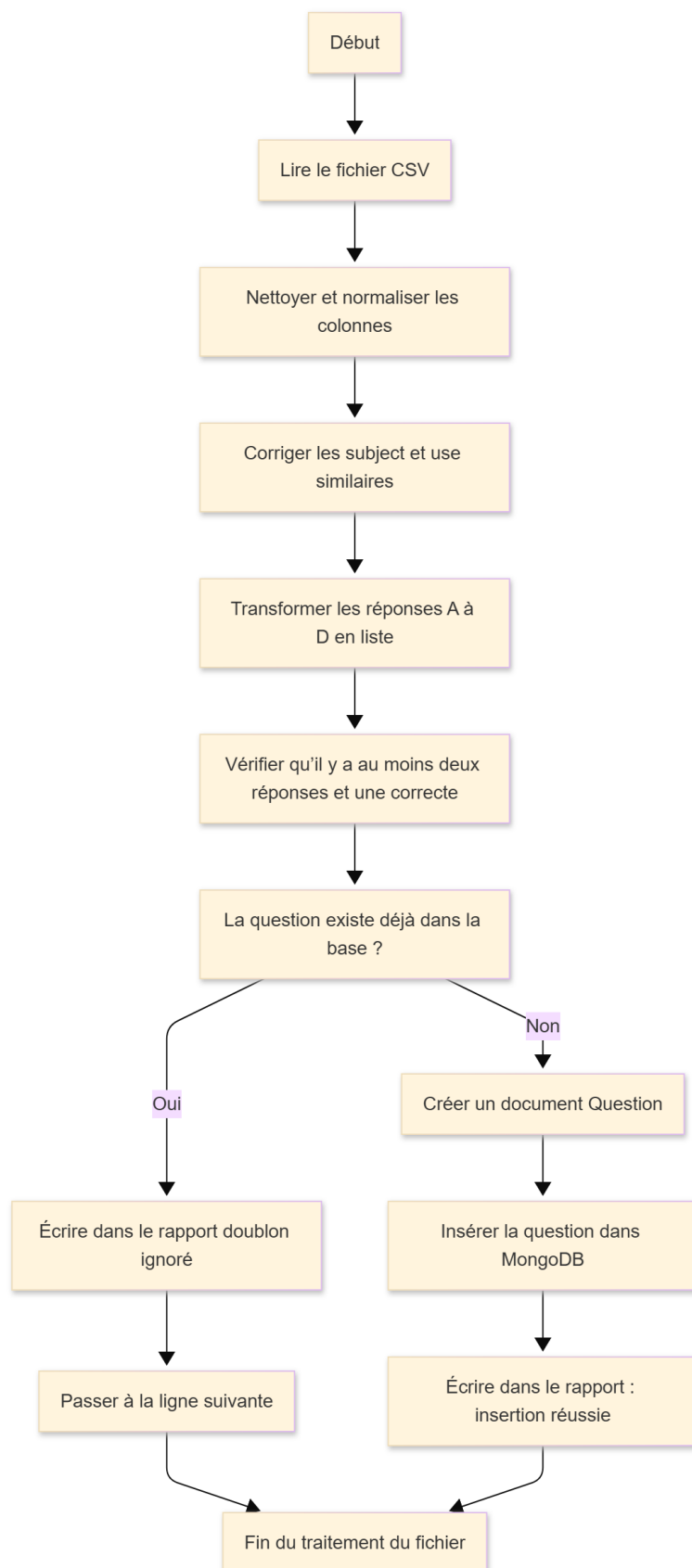


3. Flux principaux

3.1. processus ETL

Ce flux montre comment un fichier CSV est importé, normalisé et inséré dans MongoDB via le service d'importation.

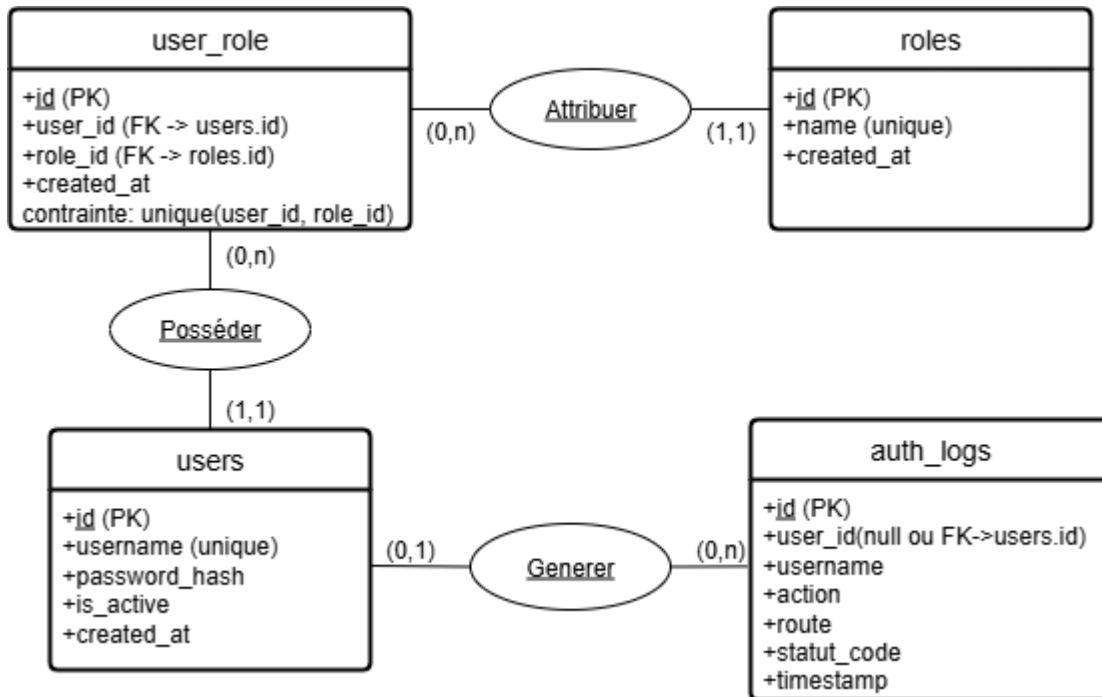
Les éventuelles erreurs ou doublons sont consignés dans un rapport d'exécution.



3.2. Authentification

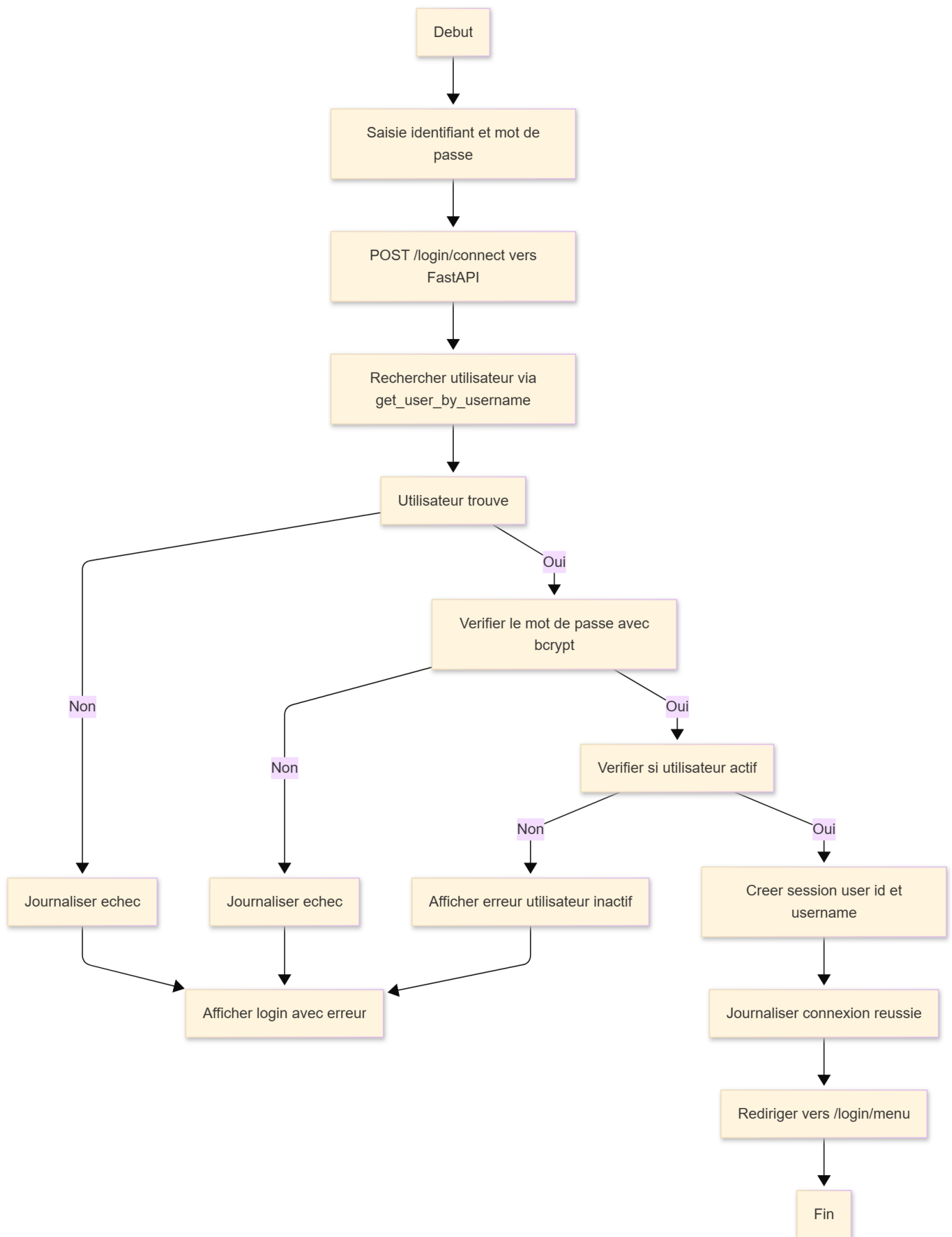
3.2.1. MCD base de donnée Utilisateurs :

Les utilisateurs sont gérés dans une base SQLITE, ci-dessous le MCD :



3.2.2. Diagramme d'authentification

Les utilisateurs se connectent via Flask, les sessions sont vérifiées par FastAPI.



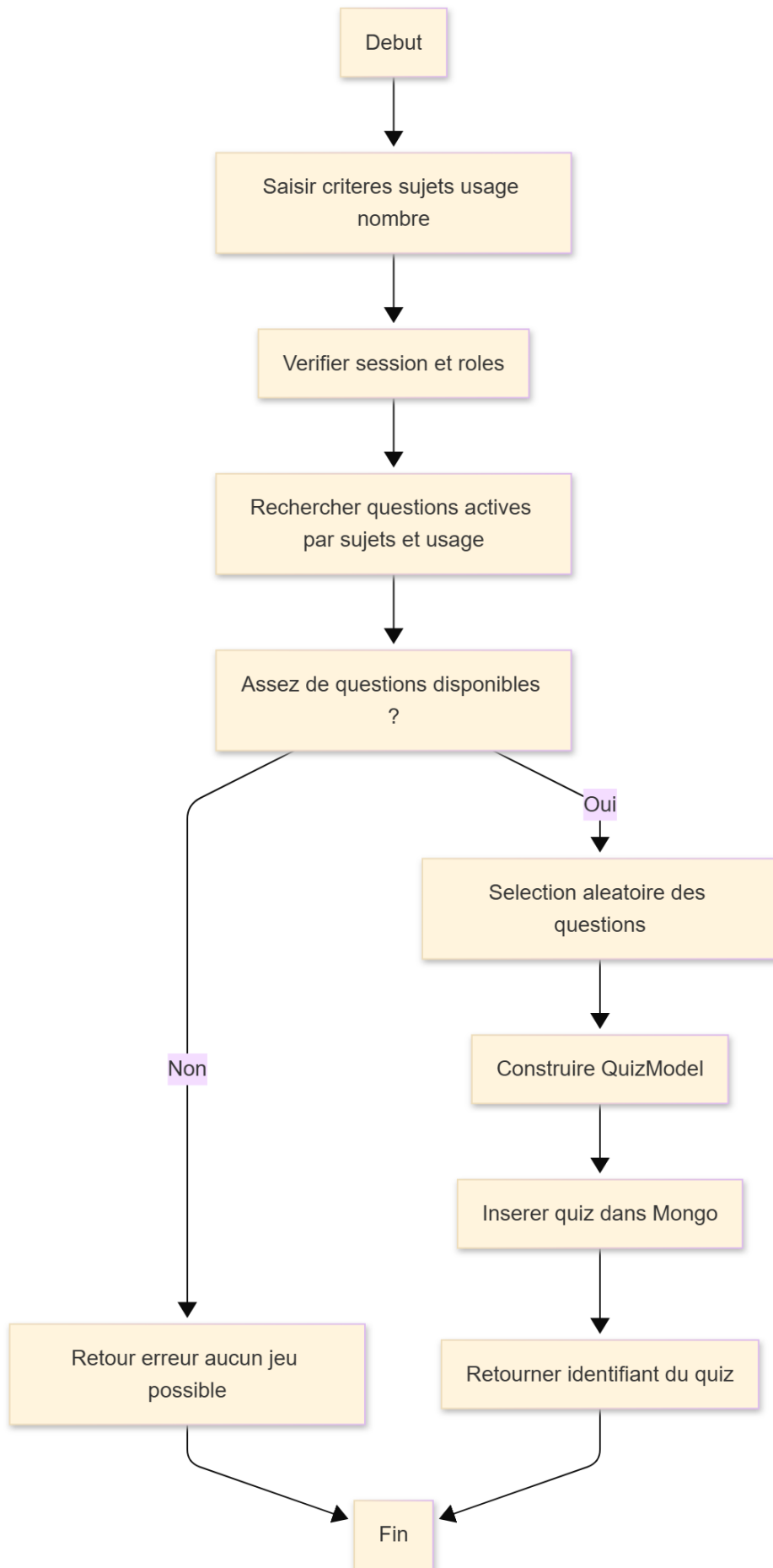
3.3. Génération Quizz

3.3.1. Format Json des questions

```
{
  "question": "Que signifie le sigle No-SQL ?",
  "subject": "BDD",
  "use": "Test de positionnement",
  "responses": [
    {"answer": "Pas seulement SQL", "isCorrect": true},
    {"answer": "Pas de SQL"},
    {"answer": "Pas tout SQL"}
  ],
  "remark": null,
  "metadata": { "author": "enseignant_X" },
  "date_creation": { "$date": "2025-09-17" },
  "date_modification": { "$date": "2025-09-17" }
}
```

3.3.2. Diagramme de génération des quizz

Les quiz sont ensuite générés dynamiquement à partir des questions disponibles selon les critères choisis.



4. Conclusion

Le projet Miskatonic met en œuvre une architecture web combinant FastAPI et Flask, appuyée sur MongoDB et SQLite. Cette solution répond aux besoins de création et de gestion de quiz en ligne, tout en garantissant la fiabilité, la traçabilité et la réutilisation des données.

L'approche retenue permet d'envisager facilement de futures évolutions, telles que l'ajout d'une interface plus interactive, l'intégration d'exports vers d'autres plateformes d'apprentissage,