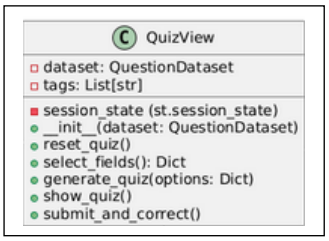
Quiz Application

Classes UML

Class QuizView



C'est la partie "interface utilisateur" (avec Streamlit) qui affiche le quiz, les questions, et les résultats à l'écran.

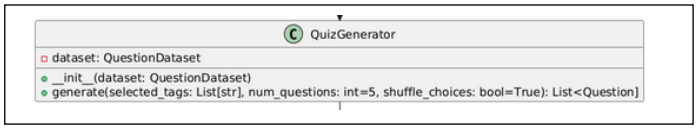
Attributs :

dataset → pour accéder aux questions.
tags → pour afficher les catégories disponibles.
session_state → pour mémoriser les réponses de l'utilisateur.

Méthodes :

reset_quiz() → remet à zéro le quiz.
select_fields() → permet de choisir les options du quiz (tags, nombre de questions…).
generate_quiz() → crée le quiz via QuizGenerator.
show_quiz() → affiche les questions à l'écran.
submit_and_correct() → affiche le score final en utilisant QuizCorrector.

Class QuizGenerator



Génère un quiz à partir des questions du dataset.
Par exemple, elle sélectionne aléatoirement 5 questions sur un thème.

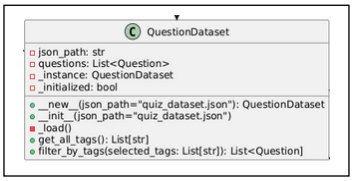
Attributs :

dataset → référence vers un objet QuestionDataset.

Méthodes:

generate(selected_tags, num_questions=5, shuffle_choices=True) => Génère une liste de questions selon les tags choisis.

Class QuestionDataset



C'est le stockage de toutes les questions, elle charge les questions depuis un fichier JSON et les garde en mémoire.

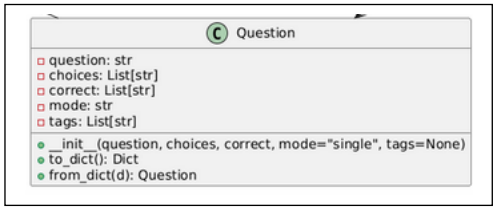
Attributs :

json_path → chemin du fichier JSON contenant les questions.
questions → une liste d'objets Question.
_instance et _initialized → servent à appliquer le design pattern Singleton

Méthodes :

_load() → charge les questions du fichier JSON.
get_all_tags() → récupère tous les mots-clés disponibles.
filter_by_tags() → filtre les questions selon des tags sélectionnés.

Class Question



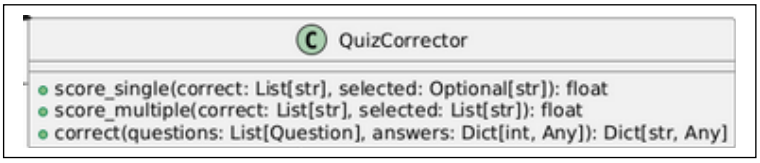
Elle représente une seule question du quiz.

Attributs :
question → le texte de la question.
choices → les choix possibles (liste).
correct → les bonnes réponses (une ou plusieurs).
mode → type de question ("single" ou "multiple").

Méthodes :
to_dict() → transforme la question en dictionnaire (utile pour sauvegarder).
_dict() → fait l'inverse (charge depuis un dictionnaire JSON).

La création de cette classe est pour rôle de structurer les données d'une question au lieu de manipuler des textes ou dictionnaires bruts.

Class QuizGenerator

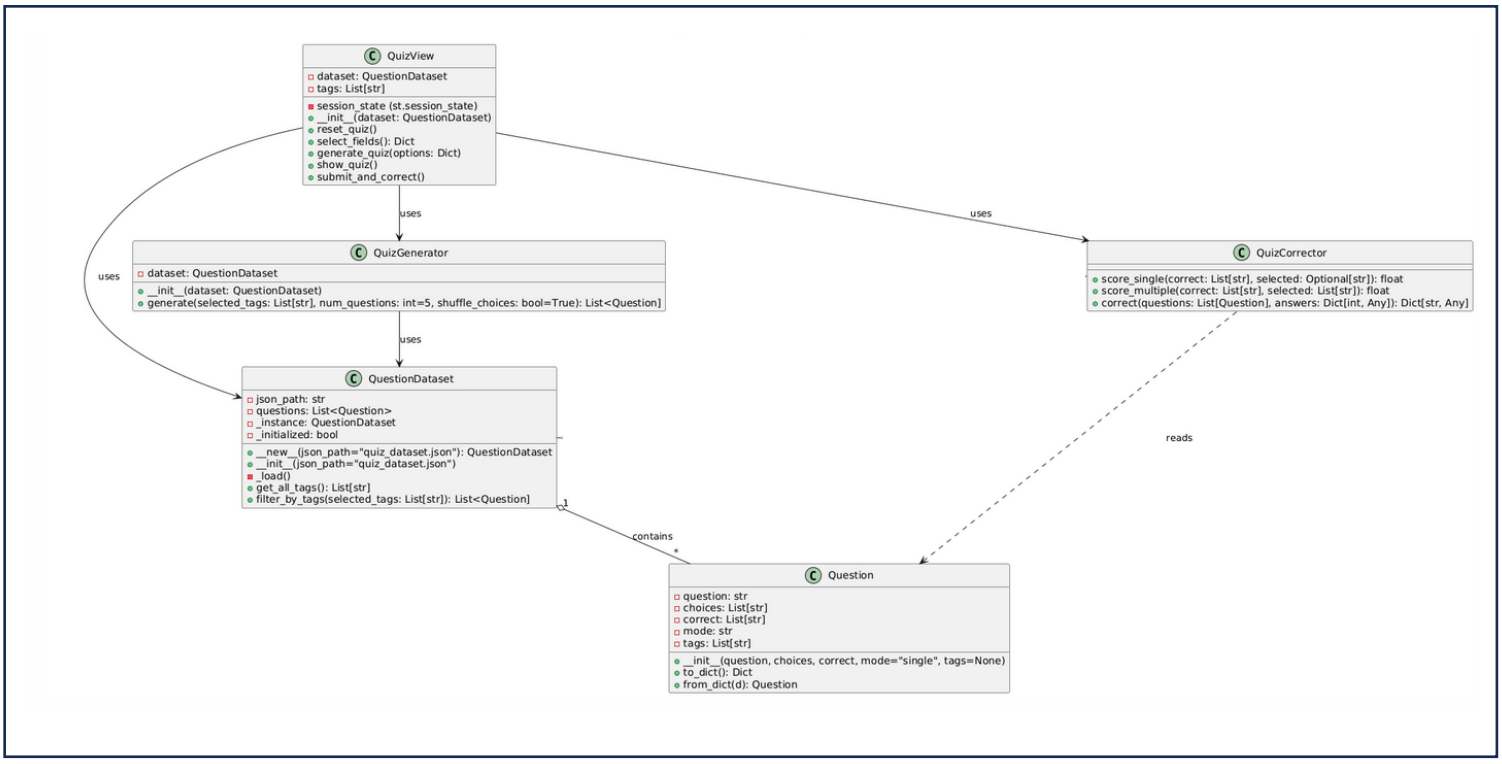


Corrige les réponses données par l'utilisateur.

Méthodes :

score_single() → calcule le score pour une question à choix unique.
score_multiple() → calcule le score partiel pour une question à choix multiple.
correct() → compare toutes les réponses de l'utilisateur avec les bonnes réponses.

DIAGRAMME DE CLASSE



Dans notre diagramme UML, plusieurs relations existent entre les classes. La classe **QuestionDataset** contient plusieurs objets **Question** c'est pourquoi on a une relation **1--> n** (un dataset peut regrouper plusieurs questions).

La classe **QuizGenerator** utilise **QuestionDataset** pour sélectionner un ensemble de questions selon les tags choisis par l'utilisateur. Ensuite, la classe **QuizCorrector** lit les objets **Question** pour comparer les réponses données avec les bonnes réponses et calculer le score. Enfin, la classe **QuizView** joue un rôle important, elle utilise **QuizGenerator** pour créer le quiz, **QuizCorrector** pour corriger les réponses et **QuestionDataset** pour afficher les catégories et les questions disponibles.

Légende

 Rouge : Attribut privé

 Vert : Méthode publique