Architecture et modèle logique

1. **Interprétation du sujet :**

Le système à développer est un outil de correction automatique dédié à l’enseignement. Néanmoins, le cadre applicatif de cet outil peut être plus large et utilisé dans d’autres domaines. Dans tous les cas d’usage, deux types différents d’utilisateurs interagissent avec l’application : un évaluateur et un évalué. On appellera évaluateur tout utilisateur en charge de mettre en place une évaluation (quizz, QCMs, rédaction, etc…) et éventuellement la corriger, quand la correction automatique n’est pas possible. Et on appellera un évalué tout utilisateur devant répondre à une ou plusieurs évaluations. Les deux utilisateurs doivent pouvoir mettre à disposition leurs travaux sous forme de dépôts dans un espace dédié. Les tâches diffèrent d’un type d’utilisateur à un autre et ont été spécifiées dans le cahier des charges.

L’intérêt d’avoir un tel système est d’automatiser les tâches répétitives dans le cadre de la correction/évaluation d’un travail, et ainsi simplifier le travail à la fois de l’évaluateur et de l’évalué et leur permettre de gagner du temps. Un autre enjeu, aussi important, c’est permettre d’éviter les *erreurs humaines* souvent liées à ce domaine, comme l’oubli de prendre en compte quelques notes, ou la mise en place d’un barème non adapté, et autres problèmes classiques qui peuvent être évités grâce à cet outil. Cet outil permettra aussi de rassembler tous les travaux dans une seul répertoire sous format numérisé, ce qui facilite la conservation des documents et y simplifie l’accès. L’outil pourra aussi être intégré à plusieurs plateforme, ce qui peut être très intéressant notamment dans le cas où des évalués suivent les mêmes cours mais sur des plateformes différentes, de leur permettre d’être évalués sur les mêmes bases et sur les mêmes normes (comme pour les concours nationaux). Cet outil a aussi l’avantage de permettre une correction équitable pour tous les utilisateurs grâce à l’automatisation de la correction (quand c’est possible).

1. **Les problèmes techniques et conceptuels exhibés :**

* **Problèmes conceptuels :**
* Identification de toutes les problématiques liées à ce système.
* Identification des dépendances du système.
* Conception des objets du système subtile.
* Les relations entre les différents objets (comment ils sont liés).
* Prise en compte de l’extensibilité de l’outil.
* Ajout de nouvelles fonctionnalités
* **Problèmes techniques :**
* Gestion des droits d’accès (Evaluateur vs Evalué).
* Gestion de la base de données.

1. **Concepts et invariants  :**

* L’architecture globale du système, à savoir un utilisateur, une interface web, un serveur et une base de données.
* Plusieurs types de dépôt suivant le type de l’utilisateur.
* Au moins deux utilisateurs
* Evaluateur : créer des taches.
* Evalué : soumettre les rendus.

1. **description de l’architecture**

**-architecture globale**

Requête Requête dépôt,…

Réponse Réponse avoir\_note ;.. utlisateur

**Interface Web**

**BDD**

**-diagramme de classes**

**-diagramme de séquentiel**