

Chapitre 4

Conception

Introduction

Dans le chapitre précédent, nous avons défini les fonctionnalités de notre application, les acteurs qui y auront accès et les interactions possibles avec le système. Ce nouveau chapitre marque une étape clé dans le cycle de développement du logiciel, car nous allons désormais présenter les différentes architectures de l'application à travers les diagrammes de classes, de séquence et d'activité, ainsi que le schéma conceptuel de la base de données. Ces éléments sont essentiels pour la conception de l'application et nous permettront de mieux visualiser son fonctionnement global.

4.1 Conception globale

Cette section est dédiée à la présentation de l'architecture de notre solution, à savoir son architecture physique et logique.

4.1.1 Architecture physique

Architecture 3-tiers :

Couche présentation: l'interface utilisateur de votre application, où l'enseignant et les étudiants peuvent interagir avec l'application. Cela peut inclure des fonctionnalités .

Couche application: cette couche traite les données de l'utilisateur et exécute les processus de l'application. Cela peut inclure l'analyse des données d'émotion, la génération de recommandations et la gestion de la communication en temps réel avec Agora.

Couche données: la couche de stockage de données, où les données de l'utilisateur, les documents et les résultats d'analyse sont stockés. Cela peut inclure une base de données relationnelle ou un système de stockage de données SQLite.

Architecture 4-tiers :

Il est possible d'utiliser une autre architecture physique pour l'application Emorse. Par exemple, on pourrait envisager une architecture physique 4-tiers, qui serait constituée de quatre couches distinctes :

Couche présentation: comme dans l'architecture 3-tiers, cela représente l'interface utilisateur de votre application.

Couche application: cette couche traite les données de l'utilisateur et exécute les processus de l'application. Cependant, cette couche est divisée en deux sous-couches:

- Couche logique: cette sous-couche comprend les fonctions métier de l'application, telles que l'analyse des émotions et la génération de recommandations.
- Couche de service: cette sous-couche fournit des services pour l'application, tels que la communication en temps réel avec Agora.

Couche données: comme dans l'architecture 3-tiers, cela représente la couche de stockage de données.

Architecture client_serveur :

Le client : il s'agit de l'interface utilisateur, c'est-à-dire la partie visible de l'application pour l'enseignant et les apprenants. Cette partie de l'application serait développée en utilisant des technologies web telles que HTML, CSS et JavaScript. Elle permettrait à l'utilisateur de lancer ou rejoindre une session de e-learning, de visualiser les recommandations générées par Emorse, et d'interagir avec les autres participants.

Le serveur : c'est la partie de l'application qui gère la logique métier, c'est-à-dire la détection et l'analyse des émotions des apprenants, la génération de recommandations, et la communication avec Agora pour la diffusion en temps réel de la session de e-learning. Cette partie de l'application serait développée en utilisant Django en tant que framework backend.

choix de l'architecture

Cependant, l'architecture client-serveur est très couramment utilisée pour les applications web, car elle permet une séparation claire entre la logique métier et l'interface utilisateur, et facilite la gestion des données et des connexions à distance. L'architecture 3-tiers est également très populaire, car elle est simple et efficace pour des projets de taille moyenne à grande. Enfin, l'architecture 4-tiers peut être utilisée pour des projets plus complexes, où il est nécessaire de séparer la logique métier en plusieurs sous-couches distinctes pour faciliter la maintenance et l'extensibilité de l'application.

4.1.2 Architecture logique

Couche interface utilisateur : cette couche correspond à l'interface graphique de l'application, qui permet à l'utilisateur d'interagir avec le système. Elle comprend des fonctionnalités telles que la création et la gestion des sessions de e-learning, l'affichage des statistiques d'émotion en temps réel et la consultation des recommandations.

Couche logique : cette couche contient les fonctions métiers de l'application, telles que l'analyse des émotions des apprenants et la génération de recommandations en fonction des résultats de l'analyse. Elle est responsable de la gestion des règles métiers et de l'orchestration des différentes tâches qui doivent être effectuées pour fournir les résultats attendus.

Couche données : cette couche est responsable du stockage et de la gestion des données de l'application, telles que les informations sur les apprenants, les sessions de e-learning et les résultats d'analyse d'émotions. Elle peut être basée sur une base de données relationnelle ou sur un système de stockage de données SQLite, en fonction des besoins de l'application.

Couche communication : cette couche est responsable de la communication en temps réel entre les différents composants de l'application, tels que l'interface utilisateur, la couche logique et la couche de données. Elle peut être mise en œuvre en utilisant des technologies telles que des sockets Web ou des API REST, en fonction des besoins de l'application.

4.2.1 Conception détaillée

Dans cette section, nous allons nous concentrer sur la conception détaillée de deux parties du système Emorse : la partie web (FrontEnd et BackEnd) et la partie Intelligence Artificielle, qui comprend la recommandation et l'analyse des sentiments.

Conclusion

Au cours de ce chapitre, nous avons présenté les aspects conceptuels de notre projet, en commençant par la conception globale avec la représentation des architectures logiques et physiques de notre système. Nous avons ensuite décrit en détail le fonctionnement de notre solution en expliquant la conception détaillée. Le prochain chapitre sera consacré à la phase de mise en œuvre de notre projet.

