

# Rapport 7 Juin 2024

## Objectifs

- Analyse Cahier de charge (Etude)
- Storage Drive (PyDrive)
- Finaliser Rapport mensuelle

## Etude

### Storage Drive

## Analyse du cahier de charge

### Principales Fonctionnalités de la Plateforme Web

#### 1. Lecture de Musique

- Lecture des chansons au format LRC avec affichage synchronisé des paroles.
- Téléchargement des chansons pour une écoute hors ligne.
- Support des formats audio les plus courants (MP3, WAV).
- Options de personnalisation de l'affichage des paroles.
- Fonctionnalités supplémentaires telles que lecture continue, support des playlists, et commandes de lecture avancées.

#### 2. Quiz Interactifs

- Création de quiz à choix multiples, questions ouvertes, et vrai ou faux.
- Interface utilisateur intuitive avec feedback immédiat.
- Système de scoring automatique avec explication des bonnes réponses.

- Suivi des progrès et tableaux de bord individuels et pour les enseignants.
- Conformité aux standards d'accessibilité et support multilingue.

### 3. **Intégration Cloud et Synchronisation**

- Utilisation de services cloud pour stocker les chansons et les cours (Amazon S3, Google Cloud Storage, Azure Blob Storage).
- Synchronisation en temps réel des progrès et des scores de quiz via WebSockets.
- Base de données centralisée pour stocker et récupérer les données utilisateur.

### 4. **Gestion des Comptes Multi-Utilisateurs**

- Création de comptes administrateurs pour les enseignants et les responsables scolaires.
- Fonctionnalités de gestion de classe pour créer, modifier, et supprimer des groupes d'élèves.
- Tableau de bord administratif pour suivre les performances globales et individuelles des élèves.

## **Modélisation**

### Acteurs

1. **Élève**
2. **Enseignant**
3. **Administrateur**
4. **Système de Stockage Cloud**
5. **Système de Paiement**

### Cas d'Utilisation

1. **Écouter de la musique**
  - Lire des chansons avec affichage synchronisé des paroles.
  - Télécharger des chansons pour une écoute hors ligne.

## 2. Participer à un quiz

- Commencer un quiz.
- Répondre aux questions.
- Recevoir un feedback immédiat.

## 3. Gérer les progrès

- Suivre les progrès individuels.
- Accéder aux tableaux de bord.

## 4. Gérer les contenus éducatifs

- Ajouter/modifier/supprimer des cours et des chansons.
- Synchroniser les contenus avec le cloud.

## 5. S'abonner à un plan

- Choisir un type d'abonnement (mensuel, trimestriel, annuel).
- Effectuer des paiements en ligne.

## 6. Accéder à l'application

- Authentification et gestion des sessions.
- Gestion des rôles et permissions.

[https://lucid.app/lucidchart/51a69a7f-800b-4e49-97bc-5f54667727d7/edit?viewport\\_loc=-276%2C-695%2C3415%2C1782%2C0\\_0&invitationId=inv\\_1fcd7e2c-dee2-4993-8245-8ef15683448f](https://lucid.app/lucidchart/51a69a7f-800b-4e49-97bc-5f54667727d7/edit?viewport_loc=-276%2C-695%2C3415%2C1782%2C0_0&invitationId=inv_1fcd7e2c-dee2-4993-8245-8ef15683448f)

Analyse du modèle

## Diagramme de Classe

### User

Représente un utilisateur de la plateforme, qu'il soit administrateur, enseignant ou étudiant.

- **userId**: Identifiant unique de l'utilisateur.
- **nom**: Nom de l'utilisateur.
- **email**: Adresse e-mail de l'utilisateur.
- **motDePasse**: Mot de passe de l'utilisateur.
- **rôle**: Rôle de l'utilisateur (Admin, Enseignant, Étudiant).

## Admin (Sous-profil de User)

Représente les administrateurs de la plateforme.

- **gestion**: Liste des tâches de gestion assignées à l'administrateur.

## Enseignant (Sous-profil de User)

Représente les enseignants utilisant la plateforme.

- **cours**: Liste des cours sous forme de musique que l'enseignant gère.

## Étudiant (Sous-profil de User)

Représente les étudiants utilisant la plateforme.

- **progress**: Liste des progrès réalisés par l'étudiant.
- **participations**: Liste des participations de l'étudiant aux quiz.
- **favorites**: Liste des musiques préférées de l'étudiant.

## Musique

Représente une musique utilisée comme support pédagogique.

- **musicId**: Identifiant unique de la musique.
- **titre**: Titre de la musique.
- **interprete**: Interprète de la musique.
- **beatmaker**: Beatmaker de la musique.
- **classe**: Classe pour laquelle la musique est destinée.
- **date\_created**: Date de création de la musique.
- **duree\_enreg**: Durée de l'enregistrement.
- **ecoutes**: Nombre d'écoutes de la musique.

- **lyrics\_enreg**: Paroles de la musique.
- **theme**: Thème abordé par la musique.
- **url\_enreg**: URL de l'enregistrement de la musique.
- **url\_img**: URL de l'image associée à la musique.
- **url\_mp3**: URL du fichier MP3 de la musique.
- **contenus**: Liste des contenus associés à la musique.
- **quizzes**: Liste des quiz associés à la musique.
- **style**: Style de la musique (rap, pop, etc.).
- **matiere**: Matière enseignée par la musique (économie, histoire, etc.).

## Contenu

Représente le contenu multimédia associé à une musique.

- **type**: Type de contenu (texte, image, vidéo).
- **données**: Données du contenu.

## Quiz

Représente un quiz associé à une musique.

- **quizId**: Identifiant unique du quiz.
- **questions**: Liste des questions du quiz.

## Question

Représente une question dans un quiz.

- **questionId**: Identifiant unique de la question.
- **texte**: Texte de la question.
- **options**: Liste des options de réponse.
- **réponse**: Option correcte de réponse.

## Option

Représente une option de réponse pour une question de quiz.

- **optionId**: Identifiant unique de l'option.

- **texte:** Texte de l'option.

## Progrès

Représente le progrès réalisé par un étudiant dans une musique.

- **progrèsId:** Identifiant unique du progrès.
- **étudiantId:** Identifiant de l'étudiant.
- **musiqueId:** Identifiant de la musique.
- **score:** Score obtenu par l'étudiant.

## Participation

Représente une participation d'un étudiant à un quiz.

- **participationId:** Identifiant unique de la participation.
- **étudiantId:** Identifiant de l'étudiant.
- **quizId:** Identifiant du quiz.
- **feedback:** Feedback donné par l'étudiant après le quiz.

## Favori

Représente une musique marquée comme préférée par un étudiant.

- **favoriId:** Identifiant unique du favori.
- **étudiantId:** Identifiant de l'étudiant.
- **musiqueId:** Identifiant de la musique.

## Relations

- **User → Admin**
- **User → Enseignant**
- **User → Étudiant**
- **Enseignant → Musique**
- **Musique → Contenu**
- **Musique → Quiz**
- **Quiz → Question**

- **Question → Option**
- **Étudiant → Progrès**
- **Étudiant → Participation**
- **Participation → Quiz**
- **Étudiant → Favori**
- **Favori → Musique**

## Ajout de la Classe Favori

Avec l'ajout de la classe **Favori**, nous assurons que chaque étudiant puisse marquer ses musiques préférées et y accéder facilement. Cela permet une expérience utilisateur plus personnalisée et engageante.

Je vais tenter de générer à nouveau le diagramme en incluant cette nouvelle classe.

## Diagramme de Classes avec Favori

```
plaintextCopier le code
```

```
User
```

- userId: String
- nom: String
- email: String
- motDePasse: String
- rôle: String

```
Admin
```

- gestion: List<Gestion>

```
Enseignant
```

- cours: List<Musique>

```
Étudiant
```

- progrès: List<Progrès>
- participations: List<Participation>
- favoris: List<Favori>

```
Musique
```

- musiqueId: String
- titre: String
- interprete: String
- beatmaker: String
- classe: String
- date\_created: String
- duree\_enreg: String
- ecoutes: Int
- lyrics\_enreg: String
- theme: String
- url\_enreg: String
- url\_img: String
- url\_mp3: String
- contenus: List<Contenu>
- quizzes: List<Quiz>
- style: String
- matiere: String

#### Contenu

- type: String
- données: Blob

#### Quiz

- quizId: String
- questions: List<Question>

#### Question

- questionId: String
- texte: String
- options: List<Option>
- réponse: Option

#### Option

- optionId: String
- texte: String

#### Progrès

- progrèsId: String



- étudiantId: String
- musiqueId: String
- score: Int

#### Participation

- participationId: String
- étudiantId: String
- quizId: String
- feedback: String

#### Favori

- favoriId: String
- étudiantId: String
- musiqueId: String

## Liste des Endpoints à Coder

### Authentification

1. **POST /auth/register**: Inscription d'un nouvel utilisateur.
2. **POST /auth/login**: Connexion d'un utilisateur.
3. **POST /auth/logout**: Déconnexion de l'utilisateur.
4. **POST /auth/refresh-token**: Rafraîchissement du token d'authentification.

### Gestion des Utilisateurs

1. **GET /users**: Récupérer la liste des utilisateurs (Admin uniquement).
2. **GET /users/{userId}**: Récupérer les détails d'un utilisateur.
3. **PUT /users/{userId}**: Mettre à jour les informations d'un utilisateur.
4. **DELETE /users/{userId}**: Supprimer un utilisateur (Admin uniquement).

### Gestion des Cours (Musique)

1. **GET /musique**: Récupérer la liste des musiques.

2. **GET /musique/{musiqueId}**: Récupérer les détails d'une musique.
3. **POST /musique**: Ajouter une nouvelle musique (Enseignant/Admin uniquement).
4. **PUT /musique/{musiqueId}**: Mettre à jour une musique (Enseignant/Admin uniquement).
5. **DELETE /musique/{musiqueId}**: Supprimer une musique (Enseignant/Admin uniquement).

## Gestion des Contenus

1. **GET /musique/{musiqueId}/contenu**: Récupérer la liste des contenus d'une musique.
2. **POST /musique/{musiqueId}/contenu**: Ajouter un contenu à une musique (Enseignant/Admin uniquement).
3. **DELETE /musique/{musiqueId}/contenu/{contenuId}**: Supprimer un contenu d'une musique (Enseignant/Admin uniquement).

## Gestion des Quiz

1. **GET /musique/{musiqueId}/quiz**: Récupérer la liste des quiz d'une musique.
2. **GET /quiz/{quizId}**: Récupérer les détails d'un quiz.
3. **POST /musique/{musiqueId}/quiz**: Ajouter un quiz à une musique (Enseignant/Admin uniquement).
4. **PUT /quiz/{quizId}**: Mettre à jour un quiz (Enseignant/Admin uniquement).
5. **DELETE /quiz/{quizId}**: Supprimer un quiz (Enseignant/Admin uniquement).

## Gestion des Questions et Options de Quiz

1. **GET /quiz/{quizId}/questions**: Récupérer la liste des questions d'un quiz.
2. **POST /quiz/{quizId}/questions**: Ajouter une question à un quiz (Enseignant/Admin uniquement).

3. **PUT /questions/{questionId}**: Mettre à jour une question (Enseignant/Admin uniquement).
4. **DELETE /questions/{questionId}**: Supprimer une question (Enseignant/Admin uniquement).
5. **GET /questions/{questionId}/options**: Récupérer la liste des options d'une question.
6. **POST /questions/{questionId}/options**: Ajouter une option à une question (Enseignant/Admin uniquement).
7. **PUT /options/{optionId}**: Mettre à jour une option (Enseignant/Admin uniquement).
8. **DELETE /options/{optionId}**: Supprimer une option (Enseignant/Admin uniquement).

## Gestion des Progrès

1. **GET /progres**: Récupérer la liste des progrès des étudiants (Enseignant/Admin uniquement).
2. **GET /progres/{progresId}**: Récupérer les détails d'un progrès.
3. **POST /progres**: Ajouter un progrès pour un étudiant.
4. **PUT /progres/{progresId}**: Mettre à jour un progrès.
5. **DELETE /progres/{progresId}**: Supprimer un progrès.

## Gestion des Participations aux Quiz

1. **GET /participations**: Récupérer la liste des participations aux quiz.
2. **GET /participations/{participationId}**: Récupérer les détails d'une participation.
3. **POST /participations**: Ajouter une participation à un quiz.
4. **PUT /participations/{participationId}**: Mettre à jour une participation.
5. **DELETE /participations/{participationId}**: Supprimer une participation.

## Gestion des Favoris

1. **GET /favoris**: Récupérer la liste des musiques favorites d'un étudiant.
2. **POST /favoris**: Ajouter une musique aux favoris d'un étudiant.

3. **DELETE /favoris/{favorid}**: Supprimer une musique des favoris d'un étudiant.

## Structure des Endpoints

### Authentication

- **POST /auth/register**

- Request Body: `{ "nom": "string", "email": "string", "motDePasse": "string", "rôle": "string" }`
- Response: `{ "message": "User registered successfully", "userId": "string" }`

- **POST /auth/login**

- Request Body: `{ "email": "string", "motDePasse": "string" }`
- Response: `{ "token": "string", "userId": "string" }`

- **POST /auth/logout**

- Request Body: `{ "token": "string" }`
- Response: `{ "message": "User logged out successfully" }`

- **POST /auth/refresh-token**

- Request Body: `{ "refreshToken": "string" }`
- Response: `{ "token": "string" }`

### Gestion des Utilisateurs

- **GET /users**

- Response: `{ "users": [ { "userId": "string", "nom": "string", "email": "string", "rôle": "string" } ] }`

- **GET /users/{userId}**

- Response: `{ "userId": "string", "nom": "string", "email": "string", "rôle": "string" }`

- **PUT /users/{userId}**

- Request Body: `{ "nom": "string", "email": "string", "rôle": "string" }`
- Response: `{ "message": "User updated successfully" }`

- **DELETE /users/{userId}**

- Response: `{ "message": "User deleted successfully" }`

## Gestion des Cours (Musique)

- **GET /musique**

- Response: `{ "musiques": [ { "musiqueId": "string", "titre": "string", "interprete": "string", "date_created": "string" } ] }`

- **GET /musique/{musiqueId}**

- Response: `{ "musiqueId": "string", "titre": "string", "interprete": "string", "beatmaker": "string", "classe": "string", "date_created": "string", "duree_enreg": "string", "ecoutes": "int", "lyrics_enreg": "string", "theme": "string", "url_enreg": "string", "url_img": "string", "url_mp3": "string", "style": "string", "matiere": "string" }`

- **POST /musique**

- Request Body: `{ "titre": "string", "interprete": "string", "beatmaker": "string", "classe": "string", "date_created": "string", "duree_enreg": "string", "lyrics_enreg": "string", "theme": "string", "url_enreg": "string", "url_img": "string", "url_mp3": "string", "style": "string", "matiere": "string" }`
- Response: `{ "message": "Musique created successfully", "musiqueId": "string" }`

- **PUT /musique/{musiqueId}**

- Request Body: `{ "titre": "string", "interprete": "string", "beatmaker": "string", "classe": "string", "date_created": "string", "duree_enreg": "string", "lyrics_enreg": "string", "theme": "string", "url_enreg": "string", "url_img": "string", "url_mp3": "string", "style": "string", "matiere": "string" }`
- Response: `{ "message": "Musique updated successfully" }`

- **DELETE /musique/{musiqueId}**

- Response: `{ "message": "Musique deleted successfully" }`

## Gestion des Contenus

- **GET /musique/{musiqueId}/contenu**

- Response: `{ "contenus": [ { "contenuId": "string", "type": "string", "donnees": "blob" } ] }`

- **POST /musique/{musiqueId}/contenu**

- Request Body: `{ "type": "string", "donnees": "blob" }`

- Response: `{ "message": "Contenu added successfully", "contenuId": "string" }`
- **DELETE /musique/{musiqueId}/contenu/{contenuId}**
  - Response: `{ "message": "Contenu deleted successfully" }`

## Gestion des Quiz

- **GET /musique/{musiqueId}/quiz**
  - Response: `{ "quizzes": [ { "quizId": "string", "questions": [ { "questionId": "string", "texte": "string" } ] } ] }`
- **GET /quiz/{quizId}**
  - Response: `{ "quizId": "string", "questions": [ { "questionId": "string", "texte": "string" } ] }`
- **POST /musique/{musiqueId}/quiz**
  - Request Body: `{ "questions": [ { "texte": "string", "options": [ { "texte": "string" } ], "réponse": { "texte": "string" } } ] }`
  - Response: `{ "message": "Quiz created successfully", "quizId": "string" }`
- **PUT /quiz/{quizId}**
  - Request Body: `{ "questions": [ { "texte": "string", "options": [ { "texte": "string" } ], "réponse": { "texte": "string" } } ] }`
  - Response: `{ "message": "Quiz updated successfully" }`
- **DELETE /quiz/{quizId}**
  - Response: `{ "message": "Quiz deleted successfully" }`

## Gestion des Questions et Options de Quiz

- **GET /quiz/{quizId}/questions**
  - Response: `{ "questions": [ { "questionId": "string", "texte": "string", "options": [ { "optionId": "string", "texte": "string" } ] } ] }`
- **POST /quiz/{quizId}/questions**
  - Request Body: `{ "texte": "string", "options": [ { "texte": "string" } ], "réponse": { "texte": "string" } }`
  - Response: `{ "message": "Question added successfully", "questionId": "string" }`

- **PUT /questions/{questionId}**

- Request Body: `{ "texte": "string", "options": [ { "texte": "string" } ], "réponse": { "texte": "string" } }`
- Response: `{ "message": "Question updated successfully" }`

- **DELETE /questions/{questionId}**

- Response: `{ "message": "Question deleted successfully" }`

- **GET /questions/{questionId}/options**

- Response: `{ "options": [ { "optionId": "string", "texte": "string" } ] }`

- **POST /questions/{questionId}/options**

- Request Body: `{ "texte": "string" }`
- Response: `{ "message": "Option added successfully", "optionId": "string" }`

- **PUT /options/{optionId}**

- Request Body: `{ "texte": "string" }`
- Response: `{ "message": "Option updated successfully" }`

- **DELETE /options/{optionId}**

- Response: `{ "message": "Option deleted successfully" }`

## Gestion des Progrès

- **GET /progres**

- Response: `{ "progrès": [ { "progrèsId": "string", "étudiantId": "string", "musiqueId": "string", "score": "int" } ] }`

- **GET /progres/{progresId}**

- Response: `{ "progrèsId": "string", "étudiantId": "string", "musiqueId": "string", "score": "int" }`

- **POST /progres**

- Request Body: `{ "étudiantId": "string", "musiqueId": "string", "score": "int" }`
- Response: `{ "message": "Progrès created successfully", "progrèsId": "string" }`

- **PUT /progres/{progresId}**

- Request Body: `{ "étudiantId": "string", "musiqueId": "string", "score": "int" }`
- Response: `{ "message": "Progrès updated successfully" }`
- **DELETE /progres/{progresId}**
  - Response: `{ "message": "Progrès deleted successfully" }`

## Gestion des Participations aux Quiz

- **GET /participations**
  - Response: `{ "participations": [ { "participationId": "string", "étudiantId": "string", "quizId": "string", "feedback": "string" } ] }`
- **GET /participations/{participationId}**
  - Response: `{ "participationId": "string", "étudiantId": "string", "quizId": "string", "feedback": "string" }`
- **POST /participations**
  - Request Body: `{ "étudiantId": "string", "quizId": "string", "feedback": "string" }`
  - Response: `{ "message": "Participation created successfully", "participationId": "string" }`
- **PUT /participations/{participationId}**
  - Request Body: `{ "étudiantId": "string", "quizId": "string", "feedback": "string" }`
  - Response: `{ "message": "Participation updated successfully" }`
- **DELETE /participations/{participationId}**
  - Response: `{ "message": "Participation deleted successfully" }`

## Gestion des Favoris

- **GET /favoris**
  - Response: `{ "favoris": [ { "favoriId": "string", "musiqueId": "string" } ] }`
- **POST /favoris**
  - Request Body: `{ "musiqueId": "string" }`
  - Response: `{ "message": "Favori added successfully", "favoriId": "string" }`



- **DELETE /favoris/{favorid}**


- Response: `{ "message": "Favori deleted successfully" }`

---

## Recherche de template

pulse - Music, Audio and Radio web application

Music, Musician, Bootstrap

 <https://flatfull.com/themes/pulse/browse.html>

## Etude 3

Pipeline soumission des musique