**Trần Duy Đan**

**Phạm Công Hậu**

**1. Kiến trúc hệ thống tổng quan**

**Sơ đồ kiến trúc tổng thể**

A diagram of a cloud storage

AI-generated content may be incorrect.

**2. Mô tả chi tiết các thành phần**

**Frontend (Flutter App)**

* **Code bằng Flutter (Dart)**
* **Các màn hình:** 
  + **Đăng nhập/Đăng ký**
  + **Trang chủ (danh sách nhạc)**
  + **Player (phát nhạc)**
  + **Tìm kiếm**
  + **Trang cá nhân (thông tin user, lịch sử)**
  + **Giao diện AI (gợi ý nhạc thông minh)**
  + **Quảng cáo (Google AdMob)**

**Backend (API Service)**

* **Ngôn ngữ: Java Spring Boot**
* **Các API chính:** 
  + **Đăng ký / Đăng nhập (JWT)**
  + **Tải nhạc lên Cloud (Firebase Storage)**
  + **Lấy danh sách nhạc (trang chủ)**
  + **Phát nhạc (stream nhạc từ Storage)**
  + **Tìm kiếm bài hát**
  + **Trang cá nhân (thông tin user, playlist, nhạc yêu thích)**
  + **Tích hợp AI**

**Cloud Storage**

* **Firebase Storage (miễn phí giới hạn)**
* **Cho phép app phát nhạc trực tuyến.**

**Database**

* **MySQL**
* **Firestore (nếu dùng Firebase)**

**AI Service**

* **Sử dụng API AI miễn phí hoặc rẻ:** 
  + **Spotify Recommendations API (miễn phí)**
  + **AI Gợi ý nhạc theo hành vi nghe nhạc**
  + **AI tách nhạc vocal/instrumental (tùy chọn)**
  + **AI tạo playlist thông minh**

**Ad Network**

* **Google AdMob để hiển thị quảng cáo banner hoặc interstitial trong app nhằm bật kiếm tiền.**

**3. Luồng dữ liệu (Data Flow)**

**User mở app Flutter, đăng ký/đăng nhập → Gửi request API đến Backend.  
Backend xử lý (xác thực JWT) → lưu thông tin user vào MySQL.  
Khi user upload nhạc, file sẽ được lưu lên Firebase Storage, link lưu trong DB.  
Khi user mở player → Backend trả link nhạc → Flutter phát nhạc.  
Khi user mở trang AI, Backend gọi AI API để gợi ý nhạc → trả về Flutter.  
Flutter hiển thị danh sách gợi ý nhạc.**

**Khi nhạc phát hoặc khi chuyển tab, app hiển thị quảng cáo Google AdMob.**

**4. Kiến trúc tầng**

**Tầng Controller (API Layer)**

* **Chứa các REST API xử lý yêu cầu từ frontend hoặc Postman.**
* **Chỉ gọi service để xử lý logic, không thao tác trực tiếp với database.**

**Tầng Service (Business Logic Layer)**

* **Xử lý nghiệp vụ, gọi repository để thao tác với database.**
* **Giúp Controller không bị quá tải, dễ test và tái sử dụng.**

**Tầng Repository (Data Access Layer)**

* **Làm việc với database, chứa các câu lệnh SQL hoặc JPA.**

**Tầng Entity (Model Layer)**

* **Chứa các đối tượng ánh xạ (Entity) với database.**

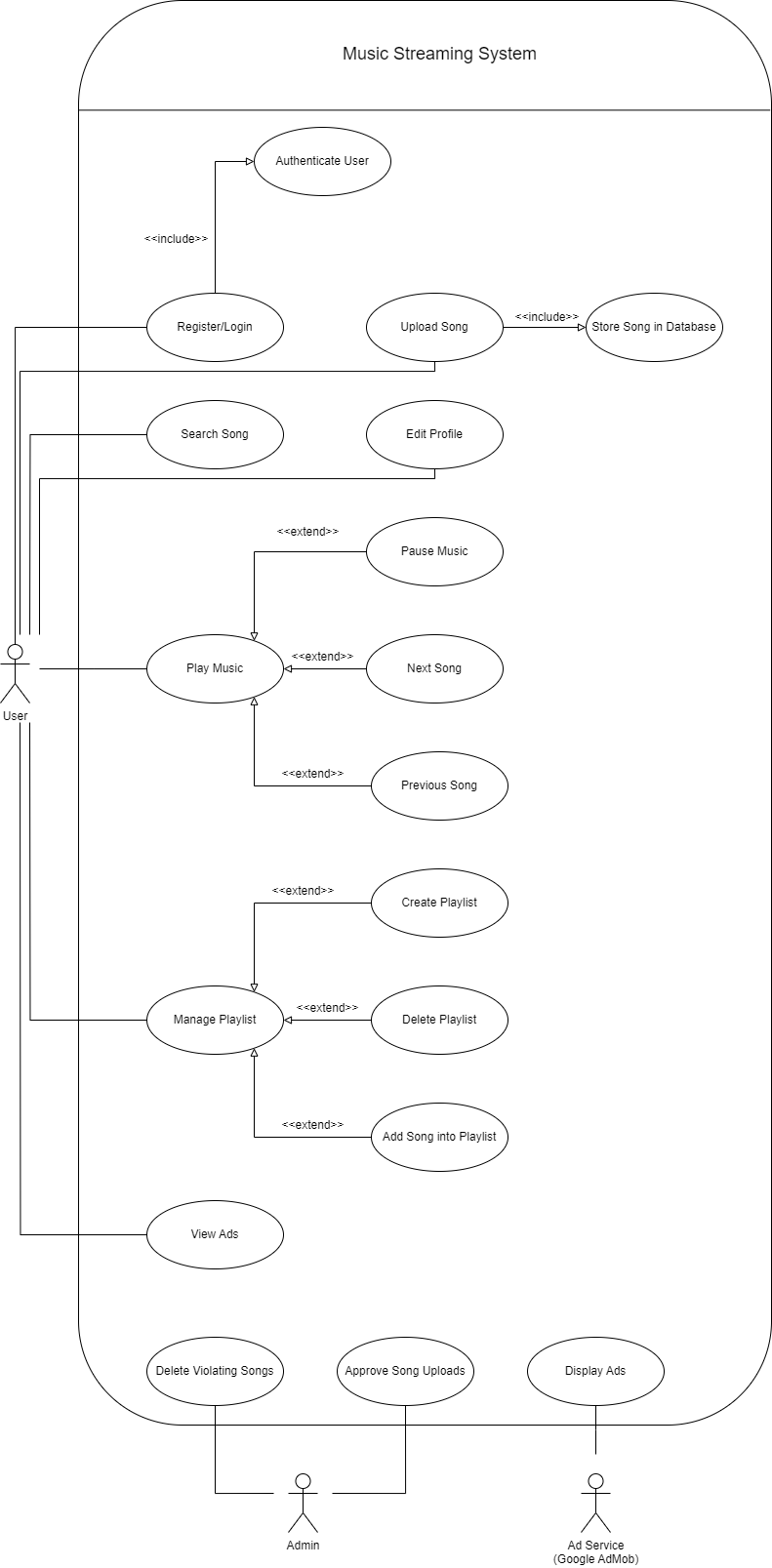
**Tích hợp Firebase (File Storage)**

* **Lưu trữ file nhạc trên Firebase Storage**
* **API backend sẽ upload file nhạc lên Firebase**

**5. Thành phần công nghệ**

|  |  |
| --- | --- |
| **Thành phần** | **Công nghệ** |
| **Client** | **Flutter + Dart** |
| **Backend** | **Java 17 + Spring Boot + REST API** |
| **Database** | **MySQL (PlanetScale/AWS RDS)** |
| **Lưu trữ nhạc** | **Firebase Storage** |
| **AI Recommendation** | **OpenAI API hoặc API nhạc AI khác** |
| **Quảng cáo** | **Google AdMob** |

**6. Biểu đồ UseCase**



**7. Lượt đồ lớp (Class Diagram)**

