

Software Requirements Specification

컴퓨터소프트웨어학부

2022085069 손주은

Tables of Contents

Introduction

1. Purpose
2. Scope
3. Overview

General Description

2.1 Program Requirements Analysis

1. Basic Requirements
2. Product Functions
 - User functions
 - Manager functions
3. User Characteristics

2.2 DBMS Requirements Analysis

1. Users
2. Administrator
3. Song
4. Album
5. Artist
6. Playlist
7. Genre
8. Song_Tags

Entity-Attributes Table

1. INTRODUCTION

1. Purpose

이 문서의 목적은 음악 스트리밍 서비스에 적합한 데이터베이스 관리 시스템(DBMS)을 설계하고 그 요구사항을 정의하는 것이다. 이 DBMS는 서비스 사용자에게 신속하고 정확한 음악 검색, 추천, 사용자 데이터 관리 등 다양한 기능을 지원한다.

2. Scope

이 문서는 프로그램의 요구사항과 DBMS의 요구사항을 모두 정의한다. 전체 시스템이 제공하는 기능에서부터 이를 뒷받침하기 위한 데이터베이스의 역할까지 다루며, 사용자가 기대하는 서비스를 안정적으로 지원하기 위한 DBMS의 주요 기능을 상세히 설명한다.

3. Overview

이 문서에서는 먼저 프로그램 요구사항을 정의하고, 이를 지원하기 위한 DBMS 요구사항을 분석한다. 이후, 기능적 비기능적 요구사항 및 시스템 인터페이스, 데이터 요구사항을 제시한다.

2. GENERAL DESCRIPTION

2.1 Program Requirements Analysis

1. Basic Requirements

- 관리자는 서비스에 음악을 등록할 수 있다.
- 사용자는 서비스에 가입하여 음악을 들을 수 있다.
- 사용자는 좋아하는 음악에 표시를 해 둘 수 있다.
- 사용자는 본인의 다양한 플레이리스트를 만들 수 있으며, 공유할 수도 있다.

2. Product Functions

- **User functions**

- ① 음악 재생 기능

- 스트리밍: 원하는 음악을 즉시 재생할 수 있다.
 - 재생목록: 사용자가 플레이리스트(재생목록)을 만들고 관리할 수 있다.
 - 검색기능: 곡, 아티스트, 장르 등으로 곡을 검색할 수 있다.
 - 다양한 재생: 반복재생, 랜덤재생, 앞/뒤 로 넘기기 등의 재생이 가능하다.

- ② 추천 및 개인화

- 추천 시스템: 청취기록을 바탕으로 맞춤형 곡, 아티스트, 앨범 추천을 받을 수 있다.
 - 최신음악 및 트렌드: 최신음악, 인기 곡을 추천 받을 수 있다.

- ③ 음악 라이브러리 관리

- 좋아요, 즐겨 찾기를 통해 좋아하는 곡 표시가 가능하다.
 - 히스토리 기능: 사용자가 들었던 곡을 조회할 수 있다.

- **Manager functions**

- ① 음악 등록 기능

- 음원 파일 업로드: 음원을 파일로 업로드 할 수 있다.
 - 메타 데이터(아티스트, 앨범명, 곡명, 장르, 발매일 등) 입력이 가능하다.
 - 앨범 커버 및 이미지 업로드도 가능하다.
 - 가사 등록을 할 수 있다.

- ② 음원 관리 기능

- 수정 및 삭제가 가능하다.
 - 다중 등록 지원: 한 앨범에 여러 곡 등록이 가능하다.

- ③ 카테고리 및 태그 설정

- 음악을 장르별 혹은 인기곡, 최신곡 등으로 분류할 수 있다.
 - 태그 설정: 태그를 바탕으로 분류한다.

- ④ 통계 및 모니터링

- 음원의 재생횟수, 사용자 반응(좋아요 수) 등을 기록한다.

- 이런 통계는 추천 시스템에서 사용한다.
- ⑤ 권한 관리
 - 모두가 음원을 등록하거나 수정 및 삭제할 수 없다.

3. User Characteristics

- 일반 사용자: 재생, 검색, 추천을 통해 음악을 즐기는 사람들을 지칭한다.
- 관리자: 데이터 등록, 수정, 삭제 등의 권한을 가지는 사람을 지칭한다.

2.2 DBMS Requirements Analysis

1. User(이용자)

- 각 이용자는 분류 요소로 user_id, 이용자 이름, 구독 유형을 가진다.
- 각 이용자는 여러 개의 플레이리스트를 가질 수 있다.
- 각 이용자는 default로 '나의 보관함'이라는 플레이리스트를 가진다.

2. Administrator(관리자)

- 관리자는 admin_id, 관리자 이름을 가진다.

3. Song(곡)

- 각 곡은 분류 요소로 song_id, 곡명, 아티스트, 장르, 수록앨범, 가사를 필수로 가진다.
- 아티스트가 여러 명일 경우 다 기재한다.
- 곡은 장르에 의해서 구분되며 한 곡은 하나의 장르만 가진다.
- 곡은 반드시 앨범에 수록되어 있어야 한다.
- 최근에 재생된 곡은 default playlist에 자동으로 기록된다.
- 가사는 파일로 업로드 될 수 있다.

4. Album(앨범)

- 앨범은 분류 요소로 album_id, 앨범명, 아티스트, 커버 이미지, 발매일자, 장르, 수록된 곡 수를 필수로 가진다.
- 아티스트가 여러 명일 경우 다 기재한다.
- 앨범의 장르는 여러 개를 가질 수 있다.
- 하나의 앨범에는 반드시 하나 이상의 곡이 수록되어 있어야 한다.

5. Artist(아티스트)

- 아티스트는 인적사항으로 artist_id, 이름, 정보, 장르를 가진다.
- 정보에는 솔로, 듀오, 그룹인지를 나타내는 규모 정보와 여성, 남성, 혼성인지를 나타내는 성별 정보를 포함한다.
- 아티스트는 장르를 여러 개 가질 수 있다.

6. Playlist(플레이리스트)

- 플레이리스트는 분류 요소로 playlist_id, 제목, 소유자, 생성 날짜, 수록된 곡 수를 갖는다.
- 한 플레이리스트의 소유자는 한 명이다.
- Default playlist는 최근에 재생한 곡을 300개까지 담을 수 있다. 300개가 넘어갈 경우, 가장 오래된 곡부터 default playlist에서 제거된다.

7. Genre(장르)

- 각 장르는 genre_id, 장르명을 갖는다.
- 장르 이름은 중복되지 않아야 한다.

8. Song_Tags(태그)

- 태그는 tag_id, 태그명을 갖는다.
- 태그 이름은 중복되지 않는다.
- 예를 들면, 최신곡, 인기곡 등이 있다.

3. Entity-Attributes Table

Entity	Attributes
User	<u>User_id</u> , 이름, 구독 유형
Administrator	<u>Admin_id</u> , 이름
Song	<u>Song_id</u> , 곡명, 아티스트, 장르, 수록앨범, 가사
Album	<u>Album_id</u> , 앨범명, 아티스트, 커버 이미지, 발매일자, 장르, 수록된 곡 수
Artist	<u>Artist_id</u> , 이름, 규모 및 성별 정보, 장르
Playlist	<u>Playlist_id</u> , 제목, 소유자, 생성 날짜, 수록된 곡 수
Genre	<u>Genre_id</u> , 장르명
Song_Tags	<u>Tag_id</u> , 태그명