

## РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ СИСТЕМЫ ФОРМИРОВАНИЯ ДОМАШНЕЙ АУДИОТЕКИ

Панов Дмитрий Владимирович<sup>1</sup>, Ушанкова Мария Юрьевна<sup>2</sup>

<sup>1</sup>Студент;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: panov.d@mail.ru.

<sup>2</sup>Старший преподаватель;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: ushankova.m.ju@uni-dubna.ru.

*В работе рассмотрена правовая точка зрения на использование бесплатных аудиофайлов, рассмотрены популярные инструменты, позволяющие формировать домашнюю аудиотеку. Проработаны варианты использования, спроектирована схема данных. Разработана серверная часть приложения с интерфейсом REST API для взаимодействия с клиентскими приложениями, а также проведено тестирование.*

**Ключевые слова:** headless-архитектура, REST API, домашняя аудиотека.

### Для цитирования:

Панов Д. В., Ушанкова М. Ю. Разработка серверной части системы формирования домашней аудиотеки // Системный анализ в науке и образовании: сетевое научное издание. 2024. № 4. С. 14-24. EDN: FLVRQF. URL: <https://sanse.ru/index.php/sanse/article/view/632>.

## DEVELOPMENT OF THE SERVER PART OF THE HOME AUDIO LIBRARY FORMATION SYSTEM

Panov Dmitry V.<sup>1</sup>, Ushankova Maria Yu.<sup>2</sup>

<sup>1</sup>Student;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: panov.d@mail.ru.

<sup>2</sup>Senior teacher;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: ushankova.m.ju@uni-dubna.ru.

*This paper presents the legal point of view on the use of free audio files is considered, popular tools that allow you to create a home audio library are considered. Use cases have been worked out, and a data schema has been designed. The server part of the application with the REST API interface for interaction with client applications has been developed, and testing has also been carried out.*

**Keywords:** headless architecture, REST API, home audio library.

### For citation:

Panov D. V., Ushankova M. Yu. Development of the server part of the home audio library formation system. *System analysis in science and education*, 2024;(4):14-24 (in Russ). EDN: FLVRQF. Available from: <https://sanse.ru/index.php/sanse/article/view/632>.



Статья находится в открытом доступе и распространяется в соответствии с лицензией Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0/deed.ru>

## Введение

Сегодня можно найти любую композицию в сети всего за несколько секунд, а также хранить свою аудиотеку в цифровом формате, что значительно упрощает процесс её формирования и использования. Появляются даже независимые DIY институты, которые стремятся служить хранилищами культуры, собирая и документируя популярную музыку как часть повседневной жизни [1].

Исследователь Дайан Парр Уокер в своей статье отмечает, что музыка станет доступна для всех пользователей, а не только для специалистов. Важно решать вопросы интеллектуальной собственности и находить способы для эффективного управления цифровыми ресурсами [2].

Однако, несмотря на все преимущества, которые предоставляет цифровая эра, все ещё можно столкнуться с различными проблемами при формировании личной аудиотеки.

Процесс формирования домашней аудиотеки можно производить различными способами. Все эти способы можно разделить на две категории:

1. Платные. Используются лицензионные копии композиций;
2. Бесплатные. Используются нелицензионные копии или копии композиций, являющиеся общественным достоянием.

При выборе бесплатных вариантов, возникает вопрос о правомерности использования нелицензионных копий музыкальных композиций. Этот вопрос достаточно дискуссионный в юридических кругах и имеет ряд тонких моментов. Однако, если речь идёт о личном использовании, то правовая точка зрения на этот вопрос вполне ясна.

В тексте 1273 статьи ГК РФ сказано: «Допускается без согласия автора или иного правообладателя и без выплаты вознаграждения воспроизведение гражданином при необходимости и исключительно в личных целях правомерно обнародованного произведения...» [3].

Под воспроизведением имеется ввиду не непосредственное проигрывание композиции, а изготовление аналогичного произведения, т.е. копирование.

В соответствии с упомянутой статьёй снятие копий с чужого произведения разрешено для личного использования и прослушивания в кругу близких людей. Скачивание музыки из музыкальных сервисов или использование программ для скачивания файлов не является правонарушением, если всё вышеописанное делается для личного использования в кругу близких людей.

Данный факт позволяет рассматривать бесплатные способы получения доступа к музыке наравне с платными, а также рассмотреть возможность создания бесплатного сервиса для формирования домашней аудиотеки.

### 1. Анализ существующих решений

Существует множество способов для получения доступа к музыке в современном мире. Ниже приведены некоторые из наиболее популярных из них.

- **Стриминговые сервисы.** Это платформы, которые предоставляют пользователям возможность слушать музыку онлайн, не загружая ее на устройство. Они могут работать по подписке или предоставлять бесплатный доступ с ограничениями, например, с рекламой. Примерами стриминговых сервисов могут служить *Apple Music* [4], *Yandex Music* и другие.
- **Torrent-файлы.** Это файлы, которые можно загрузить с помощью протокола BitTorrent. Они могут содержать в себе как отдельные треки, так и целые альбомы или сборники. Загрузка *torrent*-файлов осуществляется с помощью специальных программ-клиентов, например, *uTorrent* или *qBittorrent* [5, 6].
- **Веб-сайты с бесплатной музыкой.** Это сайты, которые предоставляют пользователям возможность скачивать музыку бесплатно. Они могут содержать как легальный контент, предоставленный авторами или правообладателями, так и пиратский контент, размещенный без согласия правообладателей. Примерами таких сайтов могут служить *Hitmo* [7], *Zaicev.net* [8] и другие.

Каждый из этих способов имеет свои преимущества и недостатки, которые более подробно рассмотрены ниже.

## Стриминговые сервисы

Стриминговые сервисы имеют ряд достоинств, благодаря которым, они стали одним из самых популярных способов получения доступа к музыке. Одним из основных преимуществ стриминговых сервисов является доступ к огромному количеству музыкальных композиций. Это позволяет пользователям легко находить и слушать песни различных жанров и исполнителей. Не требуется тратить время на поиск музыки на разрозненных сайтах, так как музыкальные стриминговые сервисы предоставляют централизованный доступ к своему контенту, обеспечивая удобный и быстрый доступ к нужным трекам.

Однако стриминговые сервисы имеют и недостатки.

- **Отсутствие возможности скачать аудиофайлы.** Некоторые пользователи предпочитают хранить коллекции музыки на своих устройствах, а также иметь возможность редактировать файлы композиций. Однако многие стриминговые сервисы ограничивают скачивание файлов.
- **Привязанность к клиентскому приложению.** Многие пользователи имеют любимые плееры и предпочитают использовать их для прослушивания своей аудиотеки. Однако при использовании стриминговых сервисов пользователи ограничены клиентским приложением платформы и встроенным плеером.
- **Региональные ограничения на доступ к контенту.** Пользователи зарубежных стриминговых сервисов могут столкнуться с ограничением к контенту платформы или полной блокировкой сервиса.
- **Отсутствие иностранных исполнителей в библиотеках стриминговых сервисов.** Некоторые популярные иностранные исполнители могут быть представлены только на зарубежных стриминговых платформах, и не заключать контракты с отечественными сервисами.
- **Платное использование.** Необходимо вносить ежемесячную плату для получения полного доступа к аудиотеке сервиса.

## Веб-сайты с бесплатной музыкой

Главным преимуществом данного варианта является бесплатный доступ к музыке. Пользователь может свободно прослушать и скачать понравившуюся композицию.

Однако подобные веб сайты имеют ряд серьёзных недостатков.

- **Периодические проблемы с доступом к сайтам.** Сайты с бесплатной музыкой могут быть заблокированы из-за нарушений правовых норм, связанных с авторским правом.
- **Реклама.** Многие сайты с бесплатной музыкой содержат большое количество навязчивой рекламы. Рекламные банеры могут препятствовать комфортному использованию сайта.
- **Нестандартизированный формат метаданных.** Зачастую на сайтах с бесплатной музыкой не следят за соответствием метаданных единому стандарту для описания композиций, что затрудняет поиск и организацию музыки на таких сайтах. Это приводит к путанице и усложняет поиск нужных треков.
- **Дубликаты.** Могут встречаться дубликаты композиций, что, снова приводит к путанице и вредит удобству пользователей.

## Torrent-файлы

Передача файлов через *torrent*-сеть происходит с использованием протокола *BitTorrent*, который позволяет пользователям обмениваться файлами между собой, не завися от централизованного сервера. Благодаря такой организации сети заблокировать доступ к отдельной композиции практически невозможно. Если у пользователя есть *torrent*-файл или *magnet* ссылка, то при наличии активных пиров, пользователь сможет получить файл с нужной композицией.

Однако такой способ получения музыки не лишён серьёзных недостатков:

- **Проблемы с поиском *torrent*-файлов.** Найти подходящий *torrent*-файл для скачивания музыки может быть сложно, особенно если требуется конкретный трек или альбом. Поиск может осуществляться с помощью *torrent* индексов, однако доступ к ним может быть заблокирован по причине нарушения авторских прав. Более того, индексы зачастую охватывают очень широкий диапазон различных файлов и данных, что усложняет поиск именно музыкальных файлов;

- **Необходимо использовать специальные программы.** Для скачивания файлов по протоколу *BitTorrent* требуется установить *torrent*-клиент, например, *uTorrent* или *qBittorrent* [5].

Разрабатываемая система должна объединить достоинства рассмотренных решений. В частности:

1. Система не должна требовать платы за использование;
2. Система предоставляет централизованный доступ к данным;
3. Система предоставляет поиск новых композиций на сторонних ресурсах;
4. Система предоставляет открытый *REST API* для разработчиков клиентских приложений.

## 2. Видение системы

Исходя из результата анализа существующих решений было сформировано видение системы.

Разрабатываемая система представляет собой серверную часть приложения. Для взаимодействия с клиентскими программами предусмотрен интерфейс *REST API*. Система отвечает за хранение данных о композициях, альбомах, авторах, хранит аудио файлы и файлы обложек. Также система обеспечивает доступ к хранящейся в ней информации. Реализована система ролей, для организации доступа к данным для нескольких пользователей. Предусмотрена возможность поиска данных на сторонних ресурсах, которыми являются сайты с бесплатной музыкой, с последующим сохранением найденных данных в базе данных сервиса. Система запускается на домашних серверах и является полностью автономной — не зависит от других процессов системы, запущенных на других серверах или от готового набора данных, которого нужно было бы распространять вместе с копиями самой системы.

Целевой аудиторией системы являются любые пользователи, желающие собрать собственную аудиотеку, с полным доступом к аудио файлам и без необходимости вносить регулярную плату, кроме необходимой для поддержания работы сервера.

## 3. Архитектура системы

В ходе проектирования системы была выбрана *headless*-архитектура. Эта архитектурный шаблон позволяет разделить систему на серверную часть для организации доступа к базе данных и клиентскую часть для обеспечения взаимодействия с пользователем. Данный подход позволяет использовать различные клиентские приложения для разных платформ, а также позволяет преодолеть ограничения, связанные с традиционными монолитными системами, и обеспечить гибкость и масштабируемость приложений.

## 4. Функциональная модель системы

### Диаграммы вариантов использования

Диаграмма прецедентов (*use case*) отражает отношения между акторами и прецедентами и позволяет описать систему на концептуальном уровне.

При взаимодействии с подсистемой поиска композиций на сторонних ресурсах у слушателя есть возможность найти нужную композицию по поисковому запросу (см. рис. 1).



Рис. 1. Use case диаграмма подсистемы поиска композиций на сторонних ресурсах

При взаимодействии с подсистемой добавления новых записей у слушателя есть возможность предложить свои данные о композиции, исполнителе и альбоме и добавить их в очередь на модерацию. Администратор имеет доступ к данным в очереди на модерацию и принимает решение о принятии или отклонении записей, предложенных слушателями (см. рис. 2).

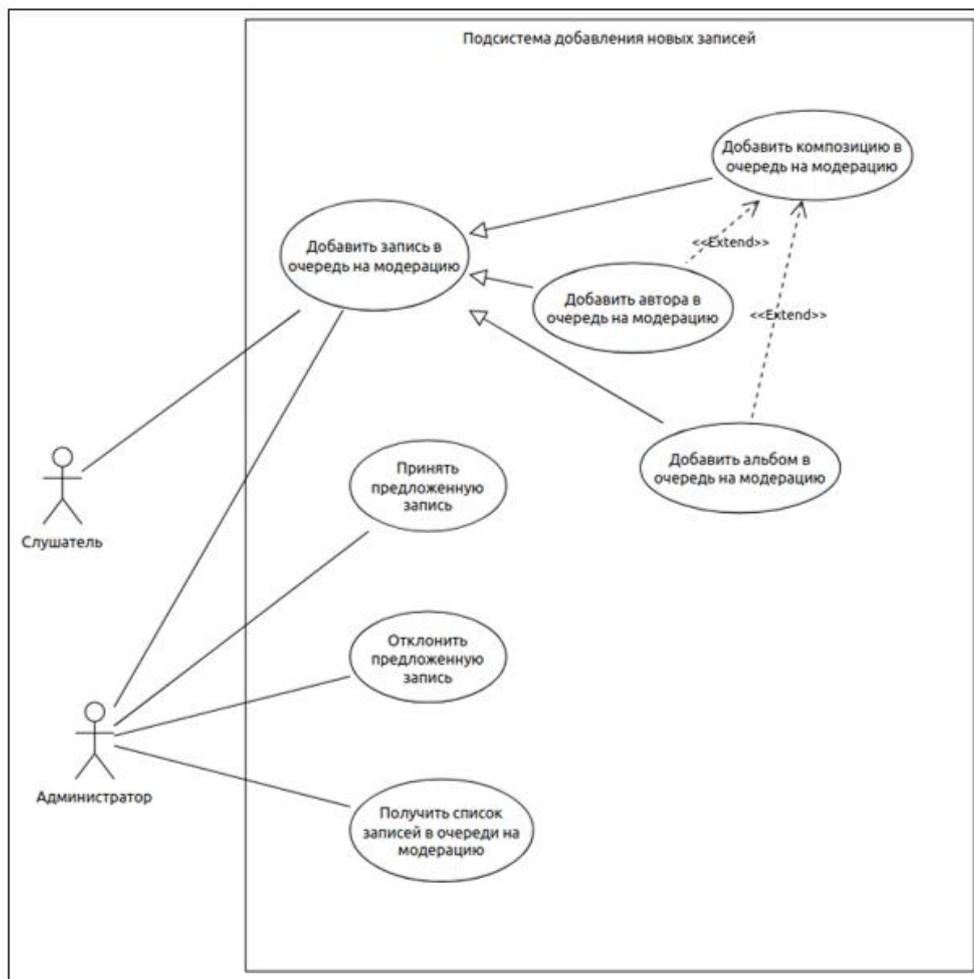


Рис. 2. Use case диаграмма подсистемы добавления новых записей

При взаимодействии с подсистемой управления данными в базе данных у пользователя есть возможность получить список всех записей, включая композиции, исполнителей и альбомы. Также слушатель может получить информацию об истории изменений той или иной записи. Администратор имеет право удалять существующие записи. (см. рис. 3).

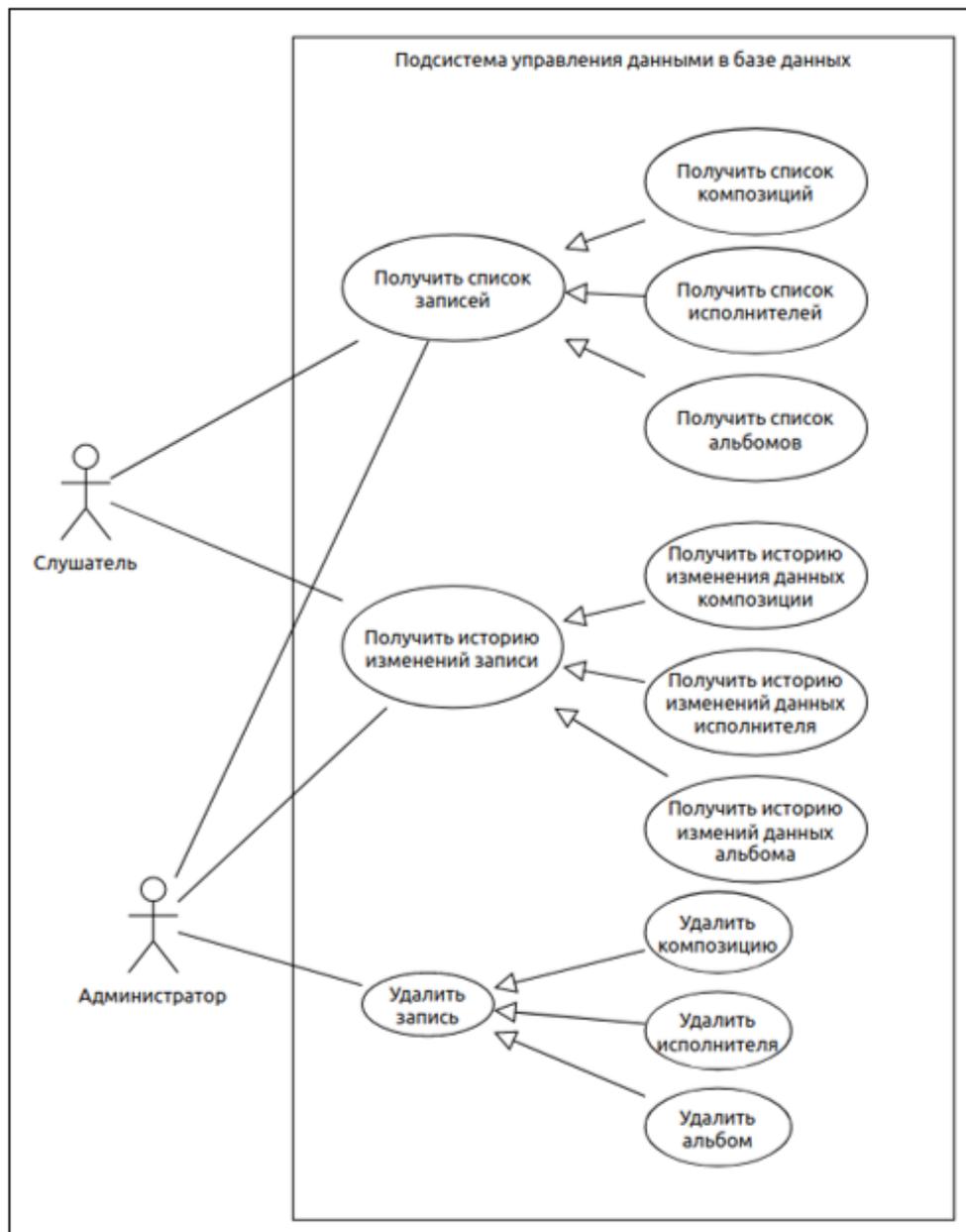


Рис. 3. Use case диаграмма подсистемы управления данными в базе данных

## Диаграммы сущностей

### Логический уровень

Информация, используемая в приложении, хранится в базе данных. Модель данных проектируемой системы состоит из 8 таблиц, которые далее описаны более подробно (см. рис. 4).

В таблице «Композиция» содержится информация о всех композициях, а также об их версиях. В таблице «Дескриптор актуальной композиции» хранятся дескрипторы версий композиции, являющихся последними. Эти значения используются как идентификаторы композиций при *http*-запросах. Таблицы «Альбом» и «Исполнитель», а также «Дескриптор актуального исполнителя» и «Дескриптор актуального альбома» содержат информацию о соответствующих сущностях. Таблица «Пользователь» содержит информацию о всех пользователях системы, включая слушателей и администраторов.

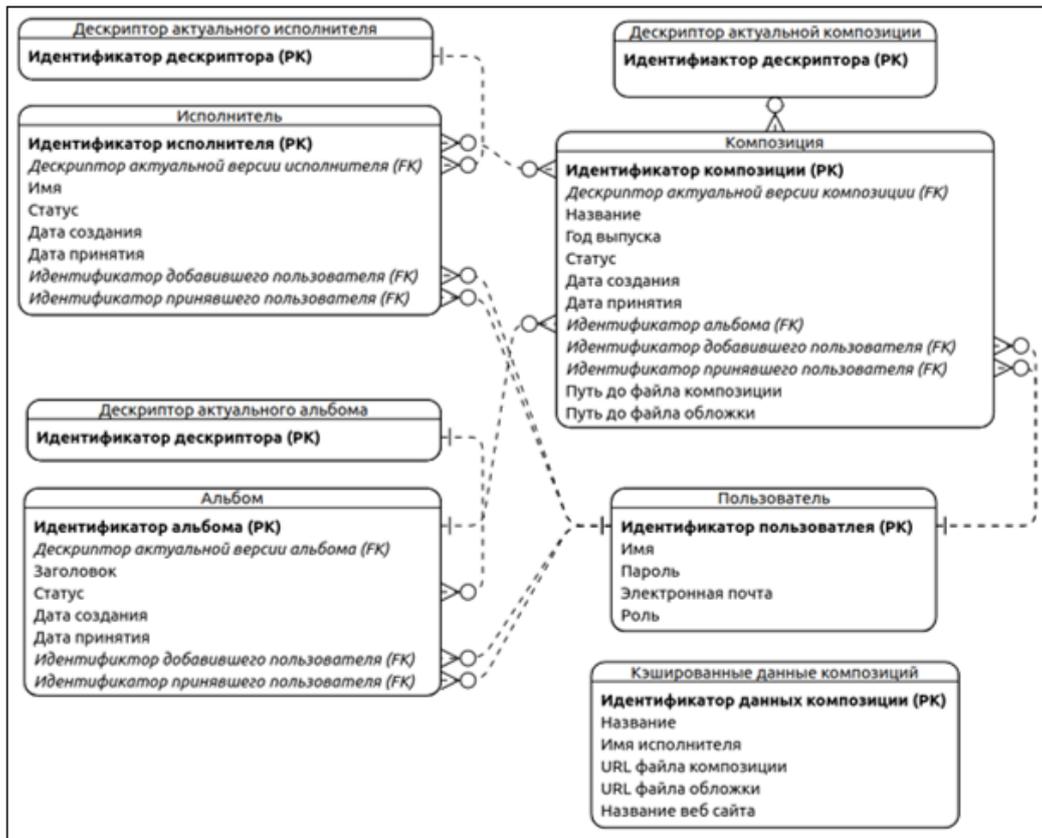


Рис. 4. Логическая модель данных

### Физический уровень

Данная модель представлена для СУБД PostgreSQL. Все атрибуты модели указаны с соответствующими типами данных. Данная модель представляет собой реляционную модель и построена с помощью программы Visual Paradigm (см. рис. 5).

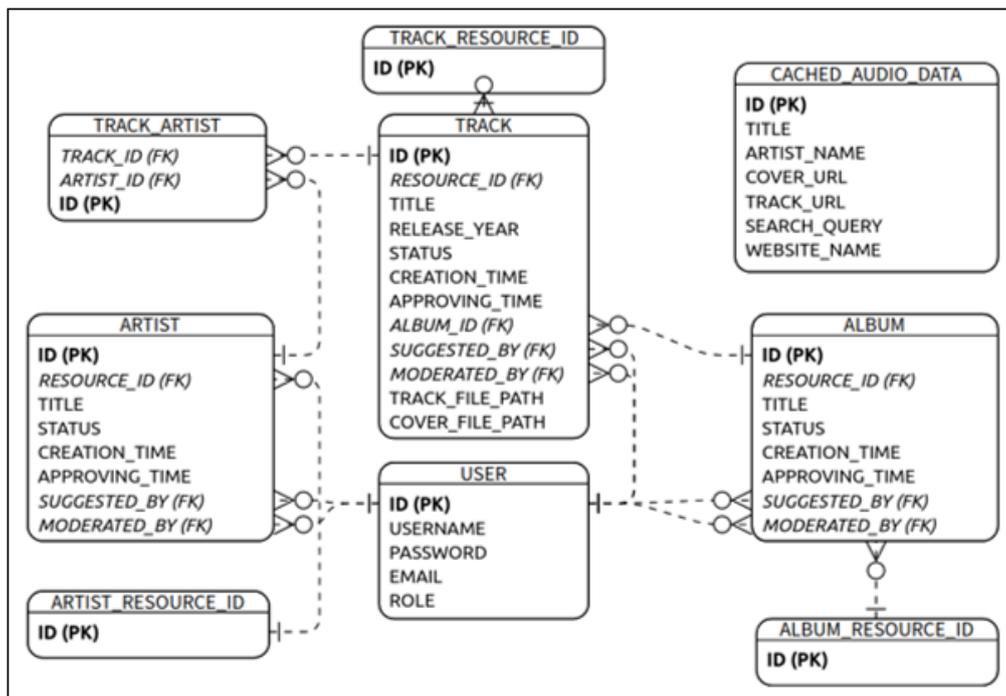


Рис. 5. Физическая модель данных

## 5. REST API

### Получение всех актуальных записей

GET <http://localhost:8080/{entity}>

Табл. 1. Параметры запроса для получения всех актуальных записей

Наименование	Тип	Пример	Описание
<i>entity</i>	текст	<i>album</i>	Сущность ( <i>album, artist, track</i> )

### Добавление новых записей альбомов и исполнителей

POST <http://localhost:8080/{entity}>

Табл. 2. Параметры запроса для добавления новых записей альбомов и исполнителей

Наименование	Тип	Пример	Описание
<i>entity</i>	текст	<i>album</i>	Сущность ( <i>album, artist</i> )

### Добавление новых записей композиций

POST <http://localhost:8080/track>

Пример тела запроса:

```
{
  "title": "Ohotа na volkov",
  "releaseYear": "1968",
  "artists": [
    {
      "resourceId": 1,
      "name": "Vladimir Visockii"
    }
  ],
  "album": {
    "resourceId": 1,
    "title": "Unnamed"
  },
  "audio": "file content",
  "cover": "file content"
}
```

### Изменение существующих записей альбомов и исполнителей

PUT <http://localhost:8080/{entity}>

Табл. 3. Параметры запроса для изменения актуальных записей

Наименование	Тип	Пример	Описание
<i>entity</i>	текст	<i>album</i>	Сущность ( <i>album, artist</i> )

**Изменение существующих записей композиций**PUT <http://localhost:8080/track>

Пример тела запроса:

```
{
  "resourceId": 1,
  "title": "Ohota na volkov",
  "releaseYear": "1968",
  "artists": [
    {
      "resourceId": 1,
      "name": "Vladimir Visockii"
    }
  ],
  "album": {
    "resourceId": 1,
    "title": "Unnamed"
  },
  "audio": "file content",
  "cover": "file content"
}
```

**Получение списка записей, находящихся в очереди на модерацию**GET <http://localhost:8080/{entity}/pending>

Табл. 4. Параметры запроса для получения списка записей, находящихся в очереди на модерацию

Наименование	Тип	Пример	Описание
<i>entity</i>	текст	<i>album</i>	Сущность ( <i>album, artist, track</i> )

**Принятие или отказ записи, находящейся в списке на модерацию**PATCH <http://localhost:8080/{entity}/pending/{action}/{id}>

Табл. 5. Параметры запроса для принятия или отказа записи, находящихся в списке на модерацию

Наименование	Тип	Пример	Описание
<i>entity</i>	текст	<i>album</i>	Сущность ( <i>album, artist, track</i> )
<i>action</i>	текст	<i>approve</i>	Принятие ( <i>approve</i> ) или отказ ( <i>reject</i> ) предложенной записи
<i>id</i>	число	1	Идентификатор записи, находящейся в списке на модерацию

**Получение файлов актуальной композиции**GET <http://localhost:8080/track/{file}/{id}>

Табл. 6. Параметры запроса для получения файлов актуальной композиции

Наименование	Тип	Пример	Описание
<i>file</i>	текст	<i>trackFile</i>	Получить аудио ( <i>trackFile</i> ) или обложку ( <i>coverFile</i> )
<i>id</i>	число	1	Идентификатор композиции

### Поиск данных на сторонних ресурсах

GET <http://localhost:8080/scraper?q={query}&src={resource}&page={num}&r={reload}>

Табл. 7. Параметры запроса для получения файлов актуальной композиции

Наименование	Тип	Пример	Описание
<i>query</i>	текст	<i>torero</i>	Поисковой запрос
<i>resource</i>	текст	<i>hitmo</i>	Сторонний ресурс
<i>num</i>	число	0	Номер страницы выдачи
<i>reload</i>	логический	<i>true</i>	Повторить запрос к сторонним ресурсам

### Технологии реализации

Для разработки описанного решения были использованы следующие технологии:

- *Java* – высокоуровневый объектно-ориентированный язык программирования, один из наиболее распространенных в разработке серверных приложений.
- *Spring Boot* – фреймворк с открытым исходным кодом, разработанный для разработки приложений на базе *Java*. Он построен на основе фреймворка *Spring* и предоставляет набор инструментов, библиотек и подходов к конфигурации, которые позволяют разработчикам создавать приложения, готовые к запуску.
- *Maven* – система автоматизации сборки проектов на языке программирования *Java*.
- *PostgreSQL* – это реляционная система управления базами данных с открытым исходным кодом.
- *Neovim* – усовершенствованный текстовый редактор, основанный на *Vim*, с улучшенной производительностью, надежностью и расширяемостью.
- *Git* – распределенная система управления версиями, предназначенная для отслеживания изменений в файлах и каталогах.
- *GitHub* – веб-сервис для хостинга репозитория *Git*, который предоставляет инструменты для работы над проектами, управления версиями и публикации кода.

### Заключение

В ходе проведения работ по созданию системы формирования домашней аудиотеки «*Ourmusic*» был проведён анализ предметной области, была выполнена постановка задачи и проведено проектирование системы. Была разработана схема данных, а также серверная часть приложения с интерфейсом *REST API* для взаимодействия с клиентскими приложениями.

Система позволяет пользователям находить нужную музыку, учитывая параметры, такие как исполнитель, альбом и год выпуска. Благодаря *REST API* пользователи смогут легко управлять своей аудиотекой, добавлять новые треки и удалять старые.

Архитектура системы оставляет возможность для гибкого добавления новых возможностей в систему. В частности, в будущем планируется добавление системы оповещения пользователей о результате модерации их записей для улучшения пользовательского опыта. Также возможно

добавление системы рейтинга пользователей и внедрить систему управления доступом к приложения для конкретных пользователей с целью автоматизировать процесс администрирования пользователей.

### Список источников

1. Abbazio J. M., Boddie A., Ogihara E. Music libraries and an expanding repertory: Suggested strategies for building diverse music library collections // Notes. – 2022. – Т. 78. – №. 3. – С. 353-379.
2. Walker D. P. Music in the academic library of tomorrow // Notes. – 2003. – Т. 59. – №. 4. – С. 817-827.
3. О свободном воспроизведении произведения в личных целях : Федеральный закон № 230ФЗ : принят Государственной Думой 24 ноября 2006 года : одобрен Советом Федерации 8 декабря 2006 года // Российская газета. Федеральный выпуск. – 2006. – 22 декабря (№ 289(7633)). – С. 15.
4. Schrepel T. The Apple Music Streaming Case: The Good, The Bad, and The Ugly // Network Law Review. Spring. – Дата публикации: 27.03.2024. URL: <https://www.networklawreview.org/apple-music-streaming/> (дата обращения 01.06.2024).
5. Understanding peer exchange in bittorrent systems/ D. Wu, D. X. Hei [и др.] //2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P). – IEEE, 2010. – С. 1-8. – DOI: 10.1109/P2P.2010.5569967.
6. qBittorrent : [торрент клиент] // GitHub : [web platform]. GitHub, Inc, 2025. – URL: <https://github.com/qbittorrent/qBittorrent> (дата обращения 01.06.2024).
7. Hitmo : [веб-сервис]. – hitmo, 2018-2024. – URL: <https://rus.hitmotop.com/> (дата обращения 01.06.2024).
8. Музыкальный портал ZAYCEV.NET : [веб-сервис]. – Москва: ЗАЙЦЕВ.НЕТ/ZAYCEV.NET, 2004-2023. – URL: <https://zaycev.net/?ysclid=lww5nwfyо6117514220> (дата обращения 01.06.2024).
9. Spring framework // Spring. – Broadcom, 2005-2024. – URL: <https://spring.io/projects/spring-framework> дата обращения 01.06.2024).
10. Worsley J., Drake J. D. Practical PostgreSQL. – O'Reilly Media, Inc, 2002. – 640 p.
11. Git : [распределенная система управления версиями с открытым исходным кодом]. – URL: <https://git-scm.com> (дата обращения: 09.04.2024).
12. GitHub: Let's build from here. – GitHub, Inc., 2024. – URL: <https://github.com/> (дата обращения: 09.04.2024).