**Содержание**

[Введение 2](#_Toc149548283)

[1. Анализ требований к программному средству. 3](#_Toc149548284)

[1.1. Аналитический обзор аналогов. 3](#_Toc149548285)

[1.1.1 Аналог «Spotify». 3](#_Toc149548286)

[1.1.2 Аналог «Яндекс Музыка». 4](#_Toc149548287)

[1.2. Разработка функциональных требований, определение вариантов использования. 5](#_Toc149548288)

[1.3 Вывод по разделу. 5](#_Toc149548289)

[2. Разработка архитектуры проекта. 6](#_Toc149548290)

[2.1. Обобщенная структура управления приложением. 6](#_Toc149548291)

[2.2. Диаграммы UML, взаимосвязь всех компонентов. 6](#_Toc149548292)

[2.3. Описание информационных объектов и ограничений целостности. 8](#_Toc149548293)

[2.3.1. Таблица ROLE\_ID. 8](#_Toc149548294)

[2.3.2. Таблица USERS. 9](#_Toc149548295)

[2.3.3. Таблица AUTORS. 9](#_Toc149548296)

[2.3.4. Таблица GENRES. 9](#_Toc149548297)

[2.3.5. Таблица TRACK\_FILES. 9](#_Toc149548298)

[2.3.6. Таблица TRACKS. 9](#_Toc149548299)

[2.3.7. Таблица MAINTRACK. 10](#_Toc149548300)

[2.3.8. Таблица LIKED. 10](#_Toc149548301)

[2.3.9. Таблица COMPLITATIONS. 10](#_Toc149548302)

[2.3.10. Таблица COMPLITATION. 10](#_Toc149548303)

[2.3.11. Таблица PLAYLISTS. 11](#_Toc149548304)

[2.3.12. Таблица PLAYLIST\_TRACK. 11](#_Toc149548305)

[2.4. Вывод по разделу. 11](#_Toc149548306)

# Введение

Целью данного проекта является разработка настольного приложения под операционную систему Windows с графическим интерфейсом, используя технологию Windows Presentation Foundation (WPF). Это приложение будет предназначено для обеспечения функциональности "Игрового помощника" и взаимодействия с реляционной базой данных, которая будет использоваться в качестве вспомогательного компонента.

Для разработки приложения будет использоваться язык программирования C#. В основе приложения будет лежать идея предоставления администратору и пользователям доступа к информации о видеоиграх, хранящейся в базе данных. Администратор сможет добавлять новые игры в базу данных, управлять уровнем доступа пользователей и модерировать комментарии и оценки. Пользователи смогут регистрироваться, авторизоваться, просматривать и редактировать свои профили, оставлять заметки и комментарии к играм, а также просматривать подробную информацию о выбранных играх.

Для взаимодействия с базой данных будет использован ADO.NET, который предоставляет удобные средства для работы с реляционными базами данных, включая выполнение хранимых процедур.

Безопасность пользователей также будет уделена особое внимание. В проекте будет использована технология шифрования паролей перед их сохранением в базе данных. Кроме того, мультимедийные данные, такие как изображения игр и аудиофайлы, будут храниться в базе данных.

Основной задачей проекта будет создание функционального приложения, демонстрирующего возможности работы с базой данных и удовлетворяющего следующим требованиям:

* Реализация функций администратора, включая добавление новых игр, управление уровнем доступа пользователей, оповещение пользователей о изменениях в расписании и модерацию комментариев и оценок.
* Реализация функций клиента, такие как регистрация, авторизация, редактирование профилей, оставление заметок и комментариев к играм, а также просмотр подробной информации о выбранных играх.

В пояснительной записке будут представлены краткая информация о похожих продуктах, архитектуре приложения, реализации проекта и руководство пользователя.

# Анализ требований к программному средству.

## **Аналитический обзор аналогов**.

Немаловажным этапом в разработке программного продукта является аналитический обзор прототипов и литературных источников.

На сегодняшний день можно встретить достаточно большое количество десктоп программ, разработанных для внутреигровых подсказок, заметок и т.д. Были рассмотрены несколько программ-аналогов.

Desktop-игровые помощники:

* «Omnicoach».
* «Overwolf»

### **Аналог «Overwolf».**

«Spotify» [1] — Это десктопное приложение, предназначенное для геймеров, которое обеспечивает доступ к разнообразным игровым расширениям и позволяет игрокам взаимодействовать с игровыми данными. Overwolf также предоставляет множество инструментов для разработчиков, чтобы они могли создавать собственные плагины и расширения для игр.

Пример плагина «Professor» для Overwolf представлен на рисунке 1.1

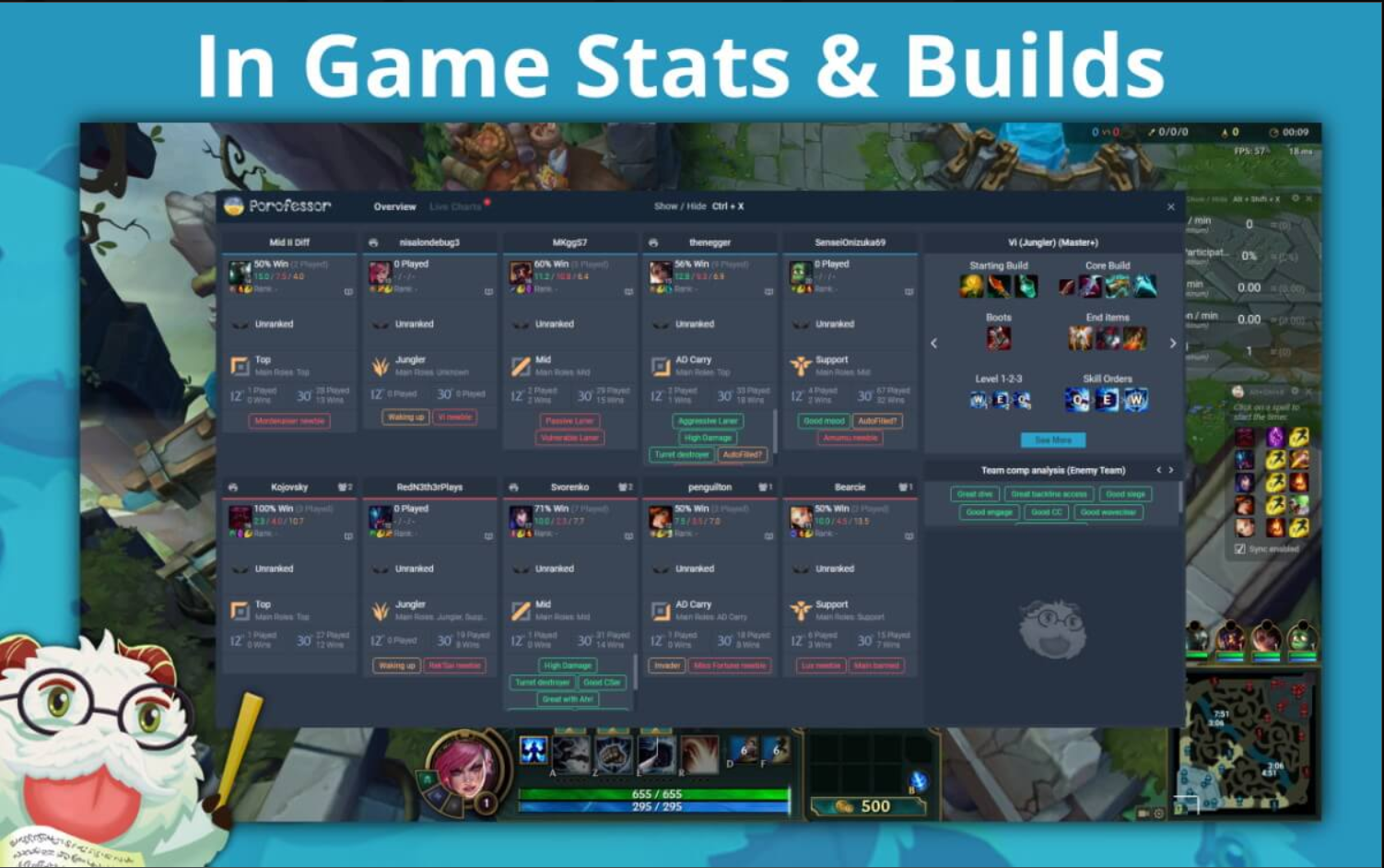


Рисунок 1.1 − Интерфейс «Professor»

Проанализировав «Overwolf», можно выделить основные минусы и плюсы данного программного средства.

Основные минусы:

* Затраты ресурсов.
* Ограничения в некоторых играх.
* Возможные конфликты с антивирусами.
* Приватность и безопасность.

Основные плюсы:

* Обширная библиотека расширений.
* Совместимость с множеством игр.
* Разработчикам доступны инструменты.
* Социальное взаимодействи.

### **Аналог «Omnicoach».**

Еще одним аналогом выбран сервис «Omnicoach» [2] который показан на рисунке 1.2. Продукт «Omnicoach» — это компьютерное программное обеспечение, разработанное для тренеров, игроков и команд в киберспорте, особенно в дисциплинах, таких как Overwatch и Valorant. Это инновационное решение предоставляет аналитические инструменты и данные, необходимые для улучшения производительности и стратегии в электронных видеоиграх.

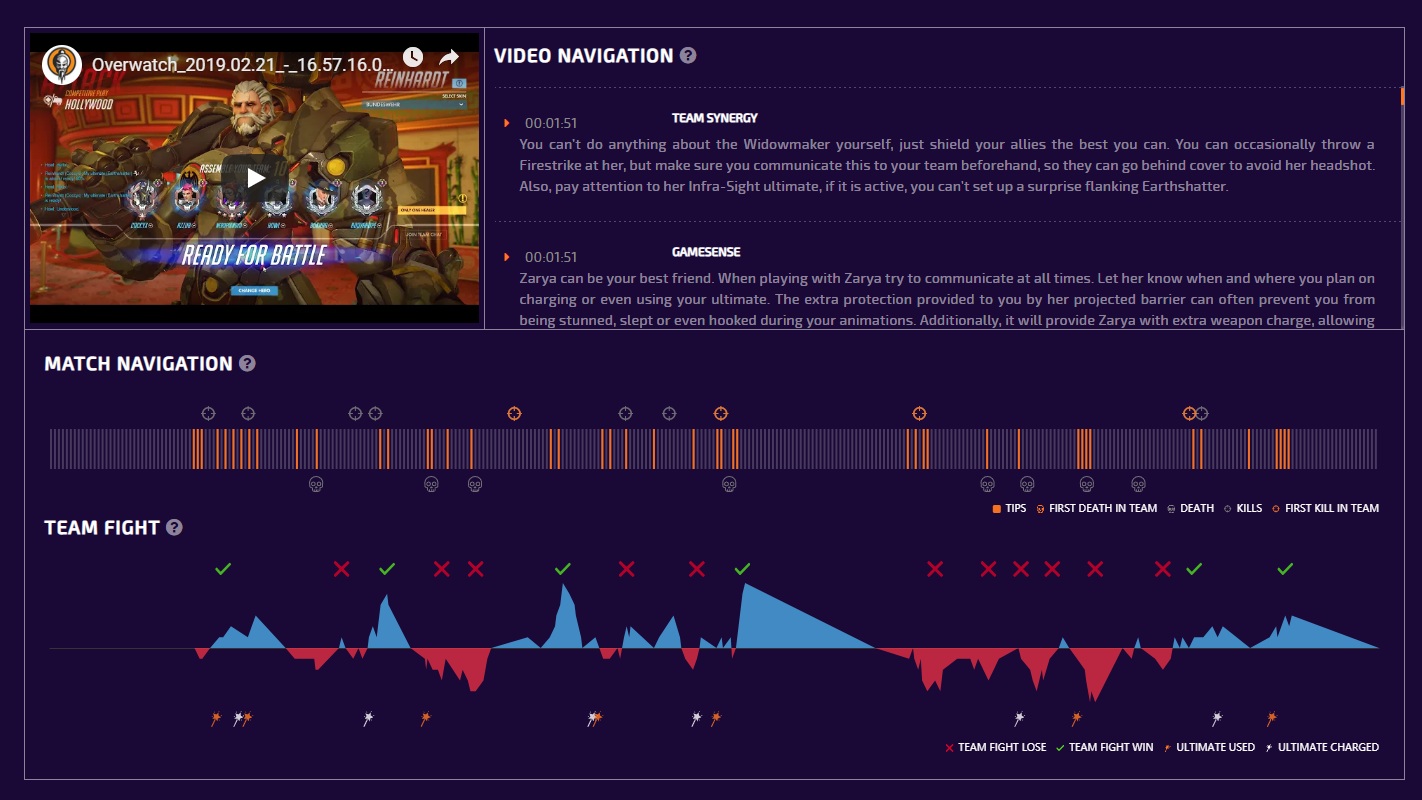


Рисунок 1.2 – интерфейс «Omnicoach»

Проанализировав «Omnicoach», можно выделить его основные минусы и плюсы.

Основные минусы:

* Платное использование
* Требует обучения
* Ограничение на платформы
* Интернет-зависимость
* Ограниченная совместимость с играми.

Основные плюсы:

* Аналитика высокого уровня
* Персонализированные рекомендации
* Видеоанализ
* Обучающие материалы
* Совместное обучение и командная работа.
  1. **Описание бизнес функций.**

При анализе ряда аналогичных программных решений, можно выделить общий функционал, необходимый для разработки вашего приложения "Игровой помощник".

Прежде всего, приложение должно поддерживать взаимодействие с игровыми данными и информацией о видеоиграх. Пользователи должны иметь возможность просматривать информацию о различных играх, включая описание, характеристики и пользовательские отзывы.

Вариантами использования базы данных являются:

* Добавление новых игр в базу данных со стороны администратора.
* Оповещение пользователей о изменениях в расписании игровых мероприятий.
* Управление уровнем доступа пользователей, включая администраторские привилегии.
* Модерация комментариев и оценок, предоставленных пользователями.
* Регистрация и аутентификация пользователей.
* Просмотр и редактирование профилей пользователей.
* Добавление заметок и комментариев к играм.
* Просмотр подробной информации о выбранных играх.

## **Вывод по разделу.**

В данном разделе были проведены исследования аналогичных решений, проанализированы их достоинства и недостатки, а также разработаны функциональные требования. Все перечисленные выше пункты являются важными для достижения поставленной цели и создания высококачественного продукта в будущем.

# Проектирование программного средства.

* 1. **Логическая схема базы данных.**

Схема базы данных ограничения целостности, связи и поля представлены на рисунке 2.1.

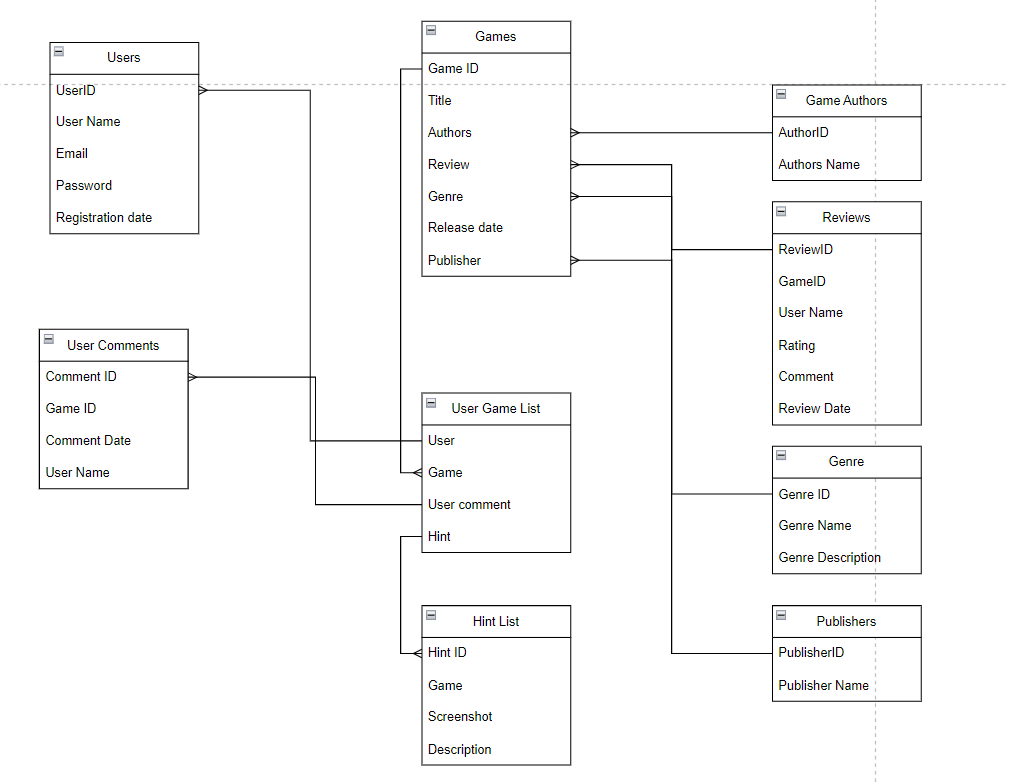
****

Рисунок 2.1 – Схема БД проекта

В проектируемой базе данных располагается 9 таблиц.

Далее будет описана структура каждой таблицы. В таблице USERS хранятся данные для идентификации пользователей, зарегистрированных в приложении, и пользователей, имеющих права модератора, ROLES содержит роли доступные для пользователей, TRACKS содержит данные о загруженных песнях, GENRES для хранения жанров песен, AUTORS для хранения авторов песен, TRACK\_FILES для хранения изображения обложки песни и самой песни, LIKED для хранения избранных песен пользователя, PLAYLISTS для хранения плейлистов созданных пользователем, COMPLITATIONS для хранения подборок созданных модератором, MAINTRACK используется для связи таблицы TRACKS и AUTORS, PLAYLIST\_TRACK используется для связи таблицы PLAYLIST и TRACKS, COMPLITATION используется для связи таблицы COMPLITATIONS и TRACKS.

### **Таблица ROLE\_ID.**

В состав таблицы Comments входят следующие столбцы:

* ID. Хранит уникальный идентификатор роли.
* NAME. Хранит название роли.

Таблица содержит ограничение первичного ключа для столбца ID.

### **Таблица USERS.**

Таблица состоит из следующих столбцов:

* ID. Хранит уникальный идентификатор пользователя.
* ROLE\_ID. Хранит уникальный идентификатор роли пользователя.
* EMAIL. Хранит логин пользователя.
* PASSWORD. Хранит хешированный пароль.

Данная таблица содержит ограничение первичного ключа для столбца ID. Также для столбца ROLE\_ID существует ограничение по внешнему ключу к таблице ROLE\_ID.

### **Таблица AUTORS.**

Таблица состоит из следующих столбцов:

* ID. Хранит уникальный идентификатор исполнителя.
* NAME. Хранит имена исполнителей.

Таблица содержит ограничение первичного ключа для столбца ID.

### **Таблица GENRES.**

В состав таблицы GENRES входят следующие столбцы:

* ID. Хранит уникальный идентификатор жанра.
* NAME. Хранит названия музыкальных жанров.

Таблица содержит ограничение первичного ключа для столбца ID.

### **Таблица TRACK\_FILES.**

В состав таблицы TRACK\_FILES входят следующие столбцы:

* ID. Хранит уникальный идентификатор файлов песни.
* TRACK\_WAY. Хранит ссылку на песню.
* TRACK\_IMG. Хранит ссылку на обложку песни.

Для столбца ID таблица содержит ограничение первичного ключа.

### **Таблица TRACKS.**

В состав таблицы TRACKS входят следующие столбцы:

* ID. Хранит уникальный идентификатор тега.
* NAME. Хранит название песни.
* GENRE\_ID. Хранит уникальный идентификатор жанра.
* TRACK\_FILE\_ID. Хранит уникальный идентификатор файла песни.

Для столбца ID таблица содержит ограничение первичного ключа. Также для столбцов GENRE\_ID, TRACK\_FILE\_ID существуют ограничения по внешнему ключу к таблицам GENRES и TRACK\_FILES.

### **Таблица MAINTRACK.**

В состав таблицы tags входят следующие столбцы:

* ID. Хранит уникальный идентификатор каждой отдельной связи между постом и тегом;
* TRACK\_ID. Хранит уникальный идентификатор поста, к которому относится тег;
* AUTOR\_ID. Хранит уникальный идентификатор тега, которой добавляется к посту.

Для столбца ID таблица содержит ограничение первичного ключа. Также для столбцов TRACK\_ID и AUTOR\_ID содержит ограничение внешнего ключа для таблиц TRACKS и AUTORS соответственно.

### **Таблица LIKED.**

В состав таблицы LIKED входят следующие столбцы:

* ID. Хранит уникальный идентификатор избранной песни;
* USER\_ID. Хранит уникальный идентификатор пользователя, который добавляет песню в избранные;
* TRACK\_ID. Хранит уникальный идентификатор песни, которая добавляется в избранные.

Для столбца ID таблица содержит ограничение первичного ключа. Также для столбцов TRACK\_ID и USER\_ID содержит ограничение внешнего ключа для таблиц TRACKS и USERS соответственно.

### **Таблица COMPLITATIONS.**

В состав таблицы COMPLITATIONS входят следующие столбцы:

* ID. Хранит уникальный идентификатор подборки;
* NAME. Хранит название подборки;
* IMAGE\_WAY. Хранит путь к обложке подборки;
* COMPLITATION\_NUMBER. Хранит количество песен в подборке.

Для столбца ID таблица содержит ограничение первичного ключа.

### **Таблица COMPLITATION.**

В состав таблицы COMPLITATION входят следующие столбцы:

* ID. Хранит уникальный идентификатор подборки;
* COMPLITATION\_ID. Хранит название подборки;
* TRACK\_ID. Хранит путь к обложке подборки.

Для столбца ID таблица содержит ограничение первичного ключа.

Также для столбцов TRACK\_ID и COMPLITATION\_ID содержит ограничение внешнего ключа для таблиц TRACKS и COMPLITATIONS соответственно.

### **Таблица PLAYLISTS.**

В состав таблицы PLAYLISTS входят следующие столбцы:

* ID. Хранит уникальный идентификатор подборки;
* NAME. Хранит название подборки;
* USER\_ID. Хранит путь к обложке подборки;

Для столбца ID таблица содержит ограничение первичного ключа. Так же для столбца USER\_ID содержит ограничение внешнего ключа для таблицу USERS.

### **Таблица PLAYLIST\_TRACK.**

В состав таблицы PLAYLIST\_TRACK входят следующие столбцы:

* ID. Хранит уникальный идентификатор подборки;
* PLAYLIST\_ID. Хранит название подборки;
* TRACK\_ID. Хранит путь к обложке подборки.

Для столбца ID таблица содержит ограничение первичного ключа. Также для столбцов TRACK\_ID и PLAYLIST\_ID\_ID содержит ограничение внешнего ключа для таблиц TRACKS и PLAYLISTS соответственно.

* 1. **Диаграммы UML, взаимосвязь всех компонентов.**

Диаграмма UML базы данных можно увидеть на рисунке 2.1.

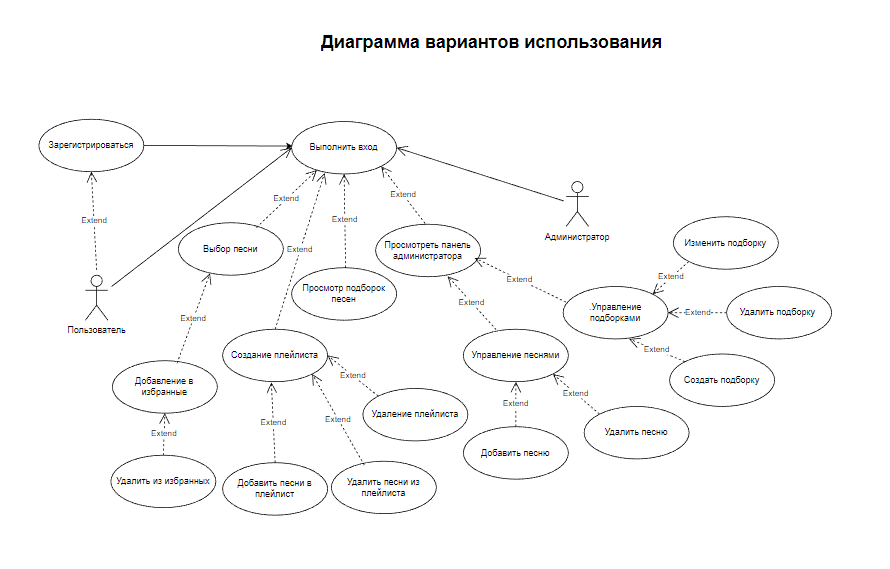


Рисунок 2.2 – Диаграмма использования

В данной схеме представлены варианты использования приложения. В возможности модератора входит:

* Просмотр подборок песен.
* Выбор песни.
* Добавление песен.
* Удаление песен.
* Создание музыкальных подборок.
* Изменение подборок.
* Удаление подборок.

В возможности пользователя входит:

* Просмотр подборок песен.
* Выбор песни.
* Добавление песен в избранные.
* Удаление песен из избранных.
* Создание плейлистов.
* Добавление песен в плейлист.
* Удаление песен из плейлиста.
* Удаление плейлистов.
* Прослушивание песен.

Также модератор может авторизоваться, а пользователи могут еще и регистрироваться.

* 1. **Вывод по разделу.**

В разделе была разработана детальная архитектура проекта, которая включает в себя не только обобщенную структуру управления приложением, но и диаграмму UML, которая поможет разработчикам лучше понимать взаимодействие компонентов приложения. Были также описаны все таблицы базы данных и ограничения целостности, что поможет обеспечить корректное и безопасное хранение данных.