МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Направление специальности 1-40 05 01 Программное обеспечение информационных технологий

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТ:**

по дисциплине «Объектно-ориентированные технологии программирования»

Тема Программное средство «Музыкальная площадка»

Исполнитель

студент (ка) 3 курса группы 2 Коренчук Андрей Васильевич

(Ф.И.О.)

Руководитель работы преп.-стажер Якунович А.В.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Председатель Смелов В.В.

(подпись)

Минск 2023

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc148872082)

[1 Аналитический обзор литературы и формирование требований 4](#_Toc148872083)

[1.1 Анализ прототипов 4](#_Toc148872084)

[1.2 Требования к проекту 6](#_Toc148872085)

[2 Анализ требований к программному средству и разработка функциональных требований 8](#_Toc148872086)

[2.1 Описание средств разработки 8](#_Toc148872087)

[2.1.1 Microsoft Visual Studio 2022 8](#_Toc148872088)

[2.1.2 Программная платформа .NET 7 8](#_Toc148872089)

[2.1.3 Язык программирования C# 8](#_Toc148872090)

[2.1.4 Технология WPF 9](#_Toc148872091)

[2.1.5 Расширяемый язык разметки XAML 9](#_Toc148872092)

[2.1.6 Технология Entity Framework 9](#_Toc148872093)

[2.1.7 PostgreSQL 9](#_Toc148872094)

[2.1.8 Паттерн MVVM 10](#_Toc148872095)

[2.2 Спецификация функциональных требований к программному средству 10](#_Toc148872096)

[2.3 Спецификация функциональных требований 11](#_Toc148872097)

[3 Проектирование программного средства 12](#_Toc148872098)

[3.1 Общая структура 12](#_Toc148872099)

# ВВЕДЕНИЕ

Данный курсовой проект посвящен разработке музыкальной площадки "MusApp", что актуально в современном мире, где музыка играет важную роль в повседневной жизни людей. MusApp представляет собой платформу, предназначенную для прослушивания, поиска и обмена музыкальными композициями. Основные характеристики MusApp включают в себя:

Прослушивание Музыки: Пользователи могут прослушивать музыку онлайн, добавлять музыку в свою библиотеку.

Поиск и Рекомендации: MusApp предлагает мощный поиск музыкальных треков.

Технические детали проекта включают в себя:

Сервер на ASP.NET Core Web API: Для обеспечения серверной части MusApp будет использован ASP.NET Core Web API. Это обеспечит масштабируемость, безопасность и производительность сервера.

База данных PostgreSQL: Для хранения информации о музыкальных треках, пользователях и библиотеках будет использоваться PostgreSQL, мощная реляционная система управления базами данных.

Клиентская часть на WPF: Для создания клиентской части MusApp будет использован Windows Presentation Foundation (WPF). WPF предоставляет мощные средства для разработки интерфейсов и визуализации данных.

Подключение к Серверу: Пользовательский клиент MusApp будет взаимодействовать с сервером через API, обеспечивая обмен данными и потоками музыки.

Проект будет спроектирован с учетом принципов ООП, SOLID.

# 1 Аналитический обзор литературы и формирование требований

## 1.1 Анализ прототипов

Были проанализированы цели и задачи, поставленные в данном курсовом проекте, а также рассмотрены аналогичные примеры их решений. На основании анализа всех достоинств и недостатков данных альтернативных решений были сформулированы требования к данному программному средству.

Первый аналог — Spotify.

Spotify - это музыкальная платформа, которая предоставляет удивительные возможности:

Бесконечная музыкальная библиотека: Spotify предлагает миллионы треков и альбомов в различных жанрах, что позволяет вам наслаждаться музыкой по вашему выбору.

Подписка и бесплатная версия: Можно выбрать между платным и бесплатным планами в зависимости от ваших потребностей.

Персонализированные рекомендации: Spotify использует алгоритмы, чтобы рекомендовать новую музыку и артистов, исходя из пользотельских предпочтений.

Оффлайн прослушивание: Можно загрузить любимые треки и слушать их без интернета.

Подкасты и аудиокниги: Возможность слушать разнообразные подкасты и аудиокниги.

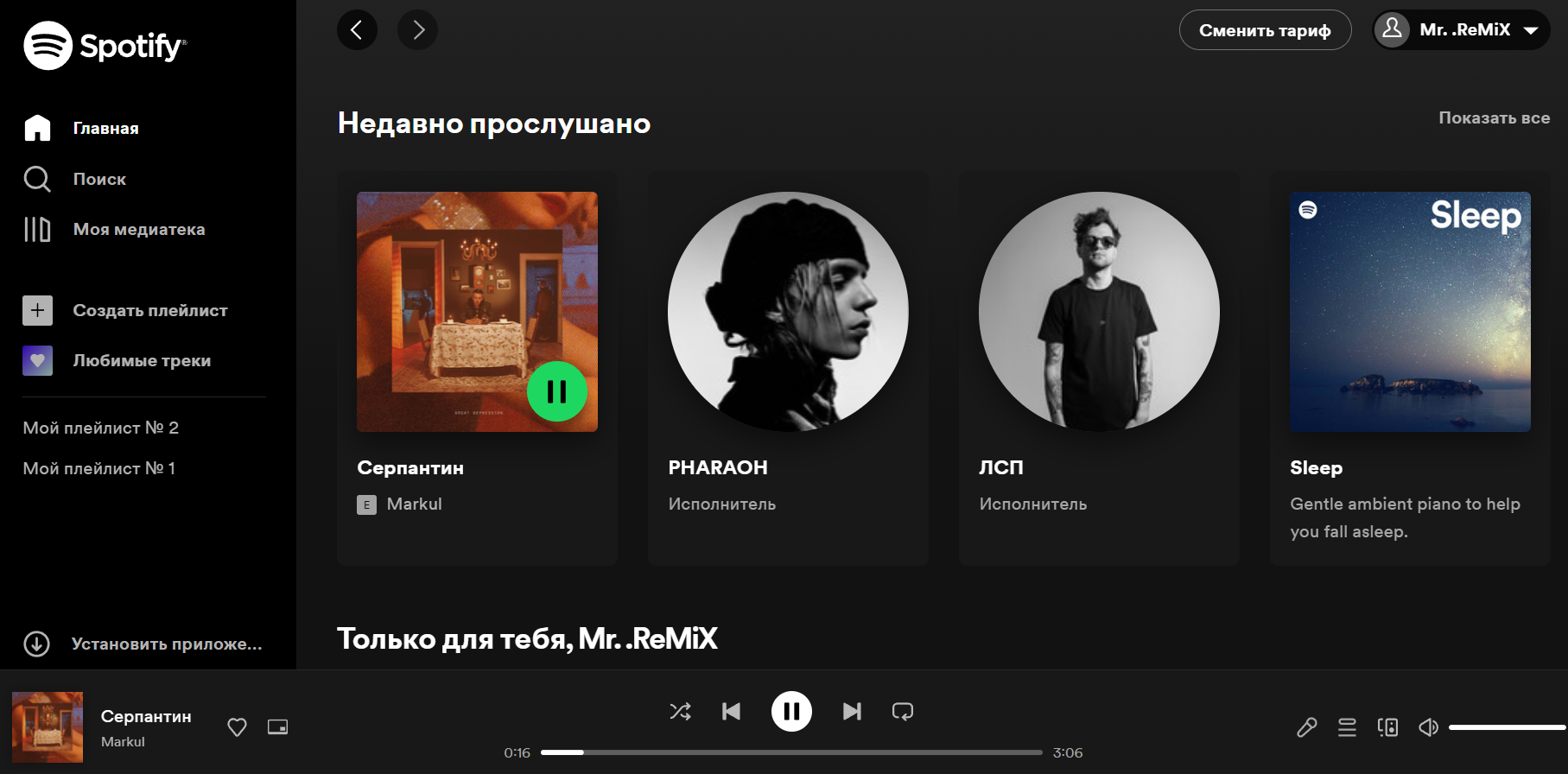
Создание собственных плейлистов: Spotify позволяет создавать собственные плейлисты, чтобы легко организовать музыку.

Интеграция с социальными сетями: Позволяет поделиться музыкальными предпочтениями с друзьями через социальные сети.

Простой и интуитивный дизайн: Интерфейс Spotify очень прост в использовании, с четкой структурой и минималистичным дизайном.

С Spotify можно наслаждаться музыкой, создавать собственные музыкальные коллекции и находить новых любимых исполнителей, все это в простом и удобном интерфейсе.

Интерфейс приложения представлен на рисунке 1.1.

* 
* Рисунок 1.1 – Приложение «Spotify»

Второй аналог — это приложение Apple Music.

Apple Music - это музыкальное приложение, которое предоставляет доступ к огромной библиотеке музыки и обширным возможностям для прослушивания и организации музыки. Пользователи могут наслаждаться миллионами треков различных жанров, создавать собственные плейлисты и настраивать свой музыкальный опыт. Вот несколько ключевых особенностей приложения Apple Music:

Библиотека с миллионами треков: Apple Music предоставляет доступ к огромной библиотеке музыки, включая новые релизы, классические хиты и многое другое. Можно найти музыку по своему вкусу и наслаждаться ей в любое время.

Подписка и семейное использование: Можно выбрать подписку Apple Music, которая подходит пользователю. Кроме того, семейная подписка позволяет семье разделять музыкальный опыт на разных устройствах.

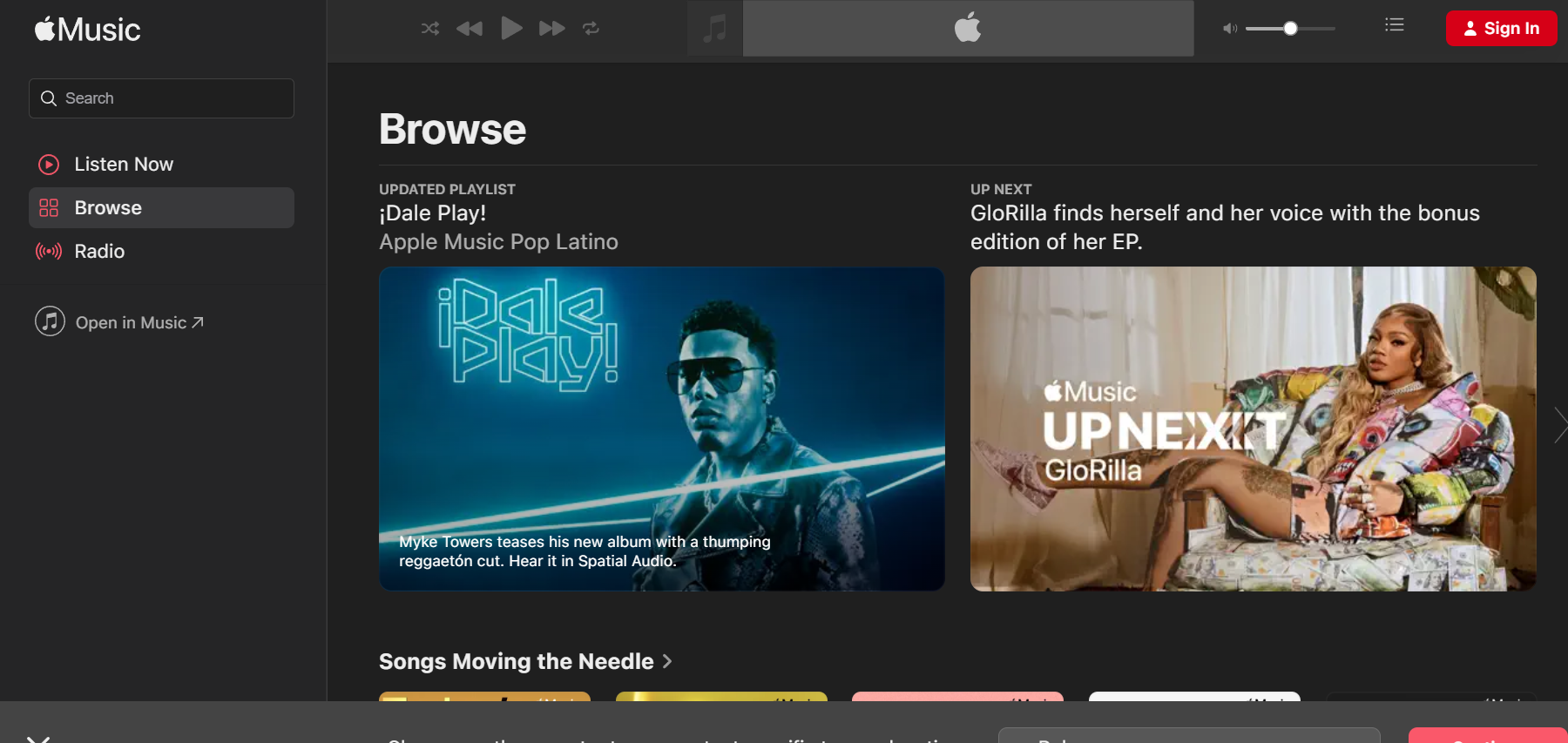
Плейлисты и персональные рекомендации: Apple Music предлагает рекомендации музыки на основе пользовательских предпочтений. Также можно создавать собственные плейлисты и делиться ими с друзьями.

Оффлайн прослушивание: Можно загружать музыку для прослушивания без интернета. Это удобно, когда пользователь находится в месте с ограниченным доступом к сети.

Интерфейс и удобство использования: Приложение Apple Music имеет интуитивно понятный интерфейс с удобным поиском и навигацией. Можно с легкостью найдете музыку, артистов и плейлисты.

В целом, Apple Music - это отличное решение для всех, кто любит музыку и ищет удобный способ ее прослушивания. Это приложение обеспечивает доступ к огромному музыкальному миру и предлагает множество возможностей для настройки вашего музыкального опыта.

Интерфейс приложения представлен на рисунке 1.2.

* 
* Рисунок 1.2 – Приложение «Apple Music»

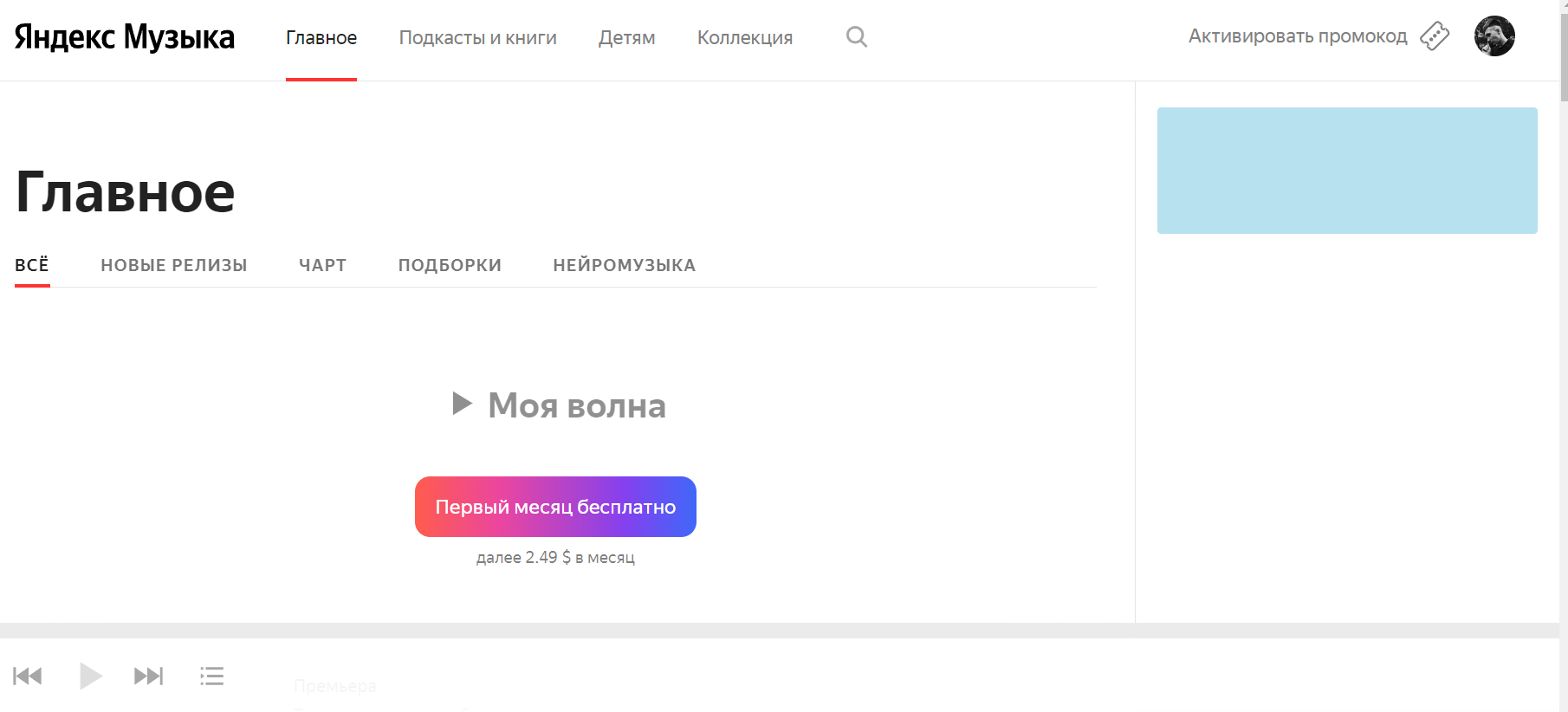
Третий аналог — это Yandex Music.

Yandex Music обладает широким спектром функций, включая прослушивание музыки, создание плейлистов, обмен музыкой с друзьями, а также персональные рекомендации для пользователей. Это приложение также предоставляет возможность оформления подписки для прослушивания музыки без рекламы и скачивания треков для прослушивания оффлайн.

Yandex Music имеет простой и интуитивно понятный интерфейс, который делает его удобным в использовании. Основные функции доступны на главном экране, а вы легко найдете свои плейлисты, артистов и альбомы. Приложение также предоставляет возможность настройки внешнего вида с помощью различных тем.

Дизайн Yandex Music отличается своей минималистичностью и легкостью использования. Он предлагает приятный интерфейс с темным фоном, который позволяет выделить контент на экране. Навигация проста, и вы с легкостью найдете желаемую музыку.

Интерфейс приложения представлен на рисунке 1.3.

* 
* Рисунок 1.3 – Приложение «Yandex Music»

Анализируя аналоги Spotify, Apple Music и Yandex Music, можно сделать вывод, что все три приложения предоставляют обширные возможности для прослушивания музыки. Каждое из них имеет свой собственный уникальный каталог треков, позволяет создавать плейлисты, предоставляет персональные рекомендации и позволяет скачивать музыку для прослушивания оффлайн.

Несмотря на индивидуальные особенности и структуру интерфейса, общими чертами всех трех приложений являются простота и удобство использования. Все они предлагают пользовательские настройки и вариативные темы оформления, что позволяет адаптировать приложение под свои предпочтения. Благодаря этим особенностям, каждый пользователь может наслаждаться музыкой и создавать персональный музыкальный опыт, независимо от выбранного сервиса.

## 1.2 Требования к проекту

Обзор аналогов музыкальных платформ позволяет сформулировать ключевые требования к разрабатываемому программному средству. Программное средство, которое будет создано в рамках данного проекта, должно поддерживать следующие функции:

1. Управление базой данных:

- Программное средство должно обеспечивать администратору возможность управления базой данных. Это включает в себя добавление новых музыкальных треков, редактирование информации о треках и удаление треков из базы данных.

2. Регистрация и авторизация:

- Пользователям должна быть предоставлена возможность регистрации на музыкальной платформе с созданием учетной записи.

- Зарегистрированные пользователи должны иметь возможность войти в свои учетные записи с использованием авторизации.

3. Добавление треков в библиотеку:

- Зарегистрированные пользователи должны иметь возможность добавлять музыкальные треки в свою личную библиотеку на платформе. Треки в библиотеке будут доступны для прослушивания и управления.

4. Загрузка треков на платформу:

- Пользователям, включая администратора, должна быть предоставлена возможность загрузки музыкальных треков на платформу. Это позволит расширить каталог доступных треков на платформе.

5. Поиск треков:

- Платформа должна предоставлять пользователям возможность поиска музыкальных треков по различным критериям, включая название трека, исполнителя, жанр и другие характеристики.

Эти требования определяют основной функционал программного средства, которое будет разработано в рамках данного проекта. Реализация этих функций позволит пользователям управлять музыкальной библиотекой, загружать и обмениваться музыкой, а также создавать и входить в учетные записи на платформе.

# 2 Анализ требований к программному средству и разработка функциональных требований

# 2.1 Описание средств разработки

При разработке приложения были использованы:

* интегрированная среда разработки Microsoft Visual Studio 2022;
* программная платформа .NET 7.0;
* язык программирования C#;
* расширяемый язык разметки XAML;
* технология WPF;
* технология Entity Framework;
* PostgreSQL.
* Asp net core.

### 2.1.1 Microsoft Visual Studio 2022

Microsoft Visual Studio 2022 — это интегрированная среда разработки для написания, отладки и сборки кода, а также последующей публикации приложений. Данный продукт позволяет разрабатывать не только консольные, но и десктопные приложения, с использованием таких технологий, как WinForms или WPF.

### 2.1.2 Программная платформа .NET 7

Платформа .NET - это программный фреймворк, разработанный компанией Microsoft, который в основном работает на операционной системе Microsoft Windows. Он предоставляет среду выполнения для запуска приложений и набор библиотек, которые используются для создания программных приложений. Фреймворк .NET состоит из нескольких компонентов, включая общую языковую среду (Common Language Runtime - CLR), библиотеку классов .NET Framework и различные инструменты разработки.

### 2.1.3 Язык программирования C#

В качестве языка программирования используется C# – основной язык разработки в .NET. Язык объектно-ориентированный, имеет строгую статическую типизацию, поддерживает перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения. Используется как основной язык в технологии WPF.

### 2.1.4 Технология WPF

Для предоставления пользовательского интерфейса и разграничения дизайна и бизнес-логики используется технология Microsoft WPF – аналог WinForms, система для построения клиентских приложений Windows с возможностями взаимодействия с пользователем и графическая подсистема в составе .NET, использующая язык разметки XAML.

### 2.1.5 Расширяемый язык разметки XAML

WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (eXtensible Application Markup Language элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление. XAML представляет собой язык декларативного описания интерфейса, основанный на XML.

### 2.1.6 Технология Entity Framework

Entity Framework (EF) - это современный ORM (Object-Relational Mapping) фреймворк, разработанный для .NET Framework. Он предоставляет более высокоуровневый способ работы с базами данных в сравнении с технологией ADO.NET.

Entity Framework (EF) предоставляет следующие возможности: абстракция данных, ленивая загрузка, поддержка различных источников данных, интеграция с LINQ, генерация схемы, отслеживание изменений, миграции, простота использования, уменьшение необходимости вручную писать SQL-запросы, поддержка асинхронных операций, управление транзакциями, облегчение тестирования.

### 2.1.7 PostgreSQL

PostgreSQL - это мощная система управления реляционными базами данных, которая предоставляет следующие возможности: поддержка структурированных данных, множество типов данных (включая числа, строки, даты и многие другие), поддержка индексов для ускорения запросов, многопользовательский доступ, расширяемость с помощью хранимых процедур и функций, поддержка транзакций для обеспечения целостности данных, возможность создания представлений для упрощения запросов, гибкая система безопасности, масштабируемость и производительность при работе с большими объемами данных, множество дополнительных модулей и расширений для различных задач.

Также PostgreSQL поддерживает расширенный язык SQL для выполнения запросов и манипуляций с данными. Это делает его мощным инструментом для работы с данными в различных сценариях, включая веб-приложения, аналитику данных, хранение журналов и многое другое.

### 2.1.8 Паттерн MVVM

MVVM (Model-View-ViewModel) - это паттерн архитектуры, используемый в разработке программного обеспечения для разделения пользовательского интерфейса (View) от бизнес-логики (Model) и связывания их через промежуточный слой ViewModel.

В MVVM модель представляет данные и бизнес-логику приложения, которая может включать в себя операции чтения/записи данных из и в источники данных, такие как база данных или веб-сервисы.

Представление (View) отвечает за отображение данных и взаимодействие с пользователем. Оно не содержит бизнес-логику и должно быть максимально независимым от модели.

ViewModel представляет промежуточный слой между View и Model. Он содержит логику, необходимую для обработки пользовательских действий, обновления данных в модели и уведомления View об изменениях. ViewModel предоставляет свойства и команды, которые привязываются к элементам пользовательского интерфейса во View, позволяя им отображать данные и реагировать на действия пользователя.

MVVM позволяет достичь разделения ответственности между компонентами приложения, облегчает тестирование и повышает переиспользуемость кода. Он также способствует лучшей поддержке параллельной разработки пользовательского интерфейса и бизнес-логики, так как разработчики могут работать независимо над своими частями приложения.

## 2.2 Спецификация функциональных требований к программному средству

Программное средство должно предоставлять следующие функциональные возможности:

Для пользователя:

* регистрация;
* авторизация;
* редактирование профиля;
* персонализация приложения;
* загружать треки;
* редактирование и удаление треков;
* прослушивание треков.

Для администратора:

* авторизация;
* рассылка сообщений;
* просмотр статистики приложения;
* просмотр всех аккаунтов;
* удаление аккаунтов;
* просмотр всех треков;
* удаление треков.

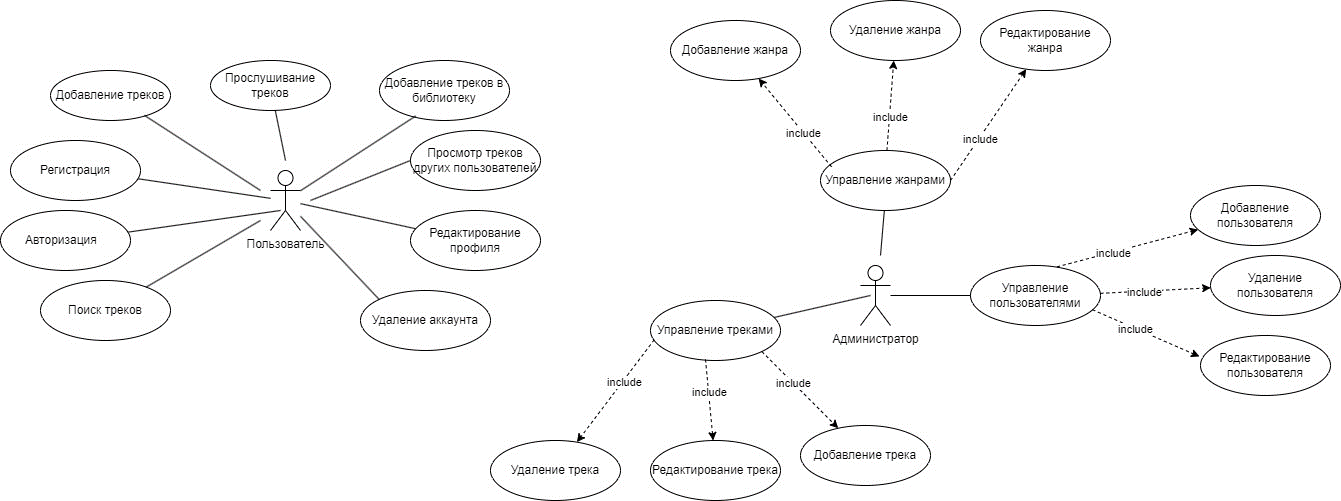
## 2.3 Спецификация функциональных требований

Для функциональности ПС необходимо создание базы данных для хранения информации приложения. Подробно база данных описано в следующем разделе.

В программном средстве необходимо реализовать регистрацию и авторизацию пользователей для доступа ко всем возможностям приложения. Для авторизации входными параметрами являются логин и пароль пользователя, которые содержатся в базе данных. Для регистрации входными данными являются имя, фото, пароль, e-mail. Введенные данные, успешно прошедшие валидацию, заносятся в базу данных.

Администратор обладает расширенными правами включая возможность удаления треков, редактирования пользователей, а также добавления и удаления жанров.

Описание функциональности программного средства «MusApp» представлено на UML-схеме, изображенной на рисунке 2.1.

* 
* Рисунок 2.1 — UML-схема

Разграничение прав пользователей, включая администратора, позволяет эффективно управлять доступом и ограничивать возможности использования приложения в соответствии с требованиями и политиками.

# 3 Проектирование программного средства

## 3.1 Общая структура