МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Направление специальности 1-40 01 01 10 Программное обеспечение информационных технологий (программирование интернет-изданий)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированные технологии и стандарты проектирования»

Тема «Музыкальный сервис VMusic»

Исполнитель

студент 2 курса группы 6 Валько Сергей Александрович

Руководитель работы ассистент Радиванович Д.А.

Курсовой проект защищён с оценкой

Председатель Пацей Н.В.

Минск 2021

ВВЕДЕНИЕ

Независимо, что люди делают: отдыхают, занимаются спортом, работают – они слушает музыку. Прослушивание музыки прочно вжилось в повседневную жизнь человека. Она помогает: сосредоточится, отдохнуть, получить заряд необходимый эмоций или направить поток мыслей в необходимое русло.

Существует уже немалое количество программных средств и сервисов, обеспечивающих возможность прослушивать треки, в пример можно привести такие сервисы, как: Spotify, Яндекс.Музыка, Youtube Music и др. Также в пример можно привести аудиоплееры: AIMP, KMPlayer и т.д.

Программное средство «Музыкальный сервис VMusic» - это приложение, позволяющее прослушивать предоставляемую коллекцию музыки, создавать плейлисты и прослушивать их, а также оценивать треки и добавлять их в избранные.

В приложении имеется 2 типа пользователей: администратор и пользователь. Администратор формирует и корректирует коллекцию треков, обновляет список популярный треков, а также производит управление пользователями. Пользователь имеет возможность пользоваться основными возможностями приложения, которые были описаны ранее.

Прежде чем преступить к использованию приложения необходимо авторизоваться, а не зарегистрированным пользователем пройти стадию регистрации.

В основы приложения положено применение принципов ООП, архитектурного паттерна MVVM, паттернов проектирования и работы с БД.

Целью данного проекта является получение опыта в разработке настольных приложений под Windows, использовании принципов ООП и паттернов проектирования, а также предоставить удобное средство для прослушивания музыки в повседневном использовании.

### Аналитический обзор литературы

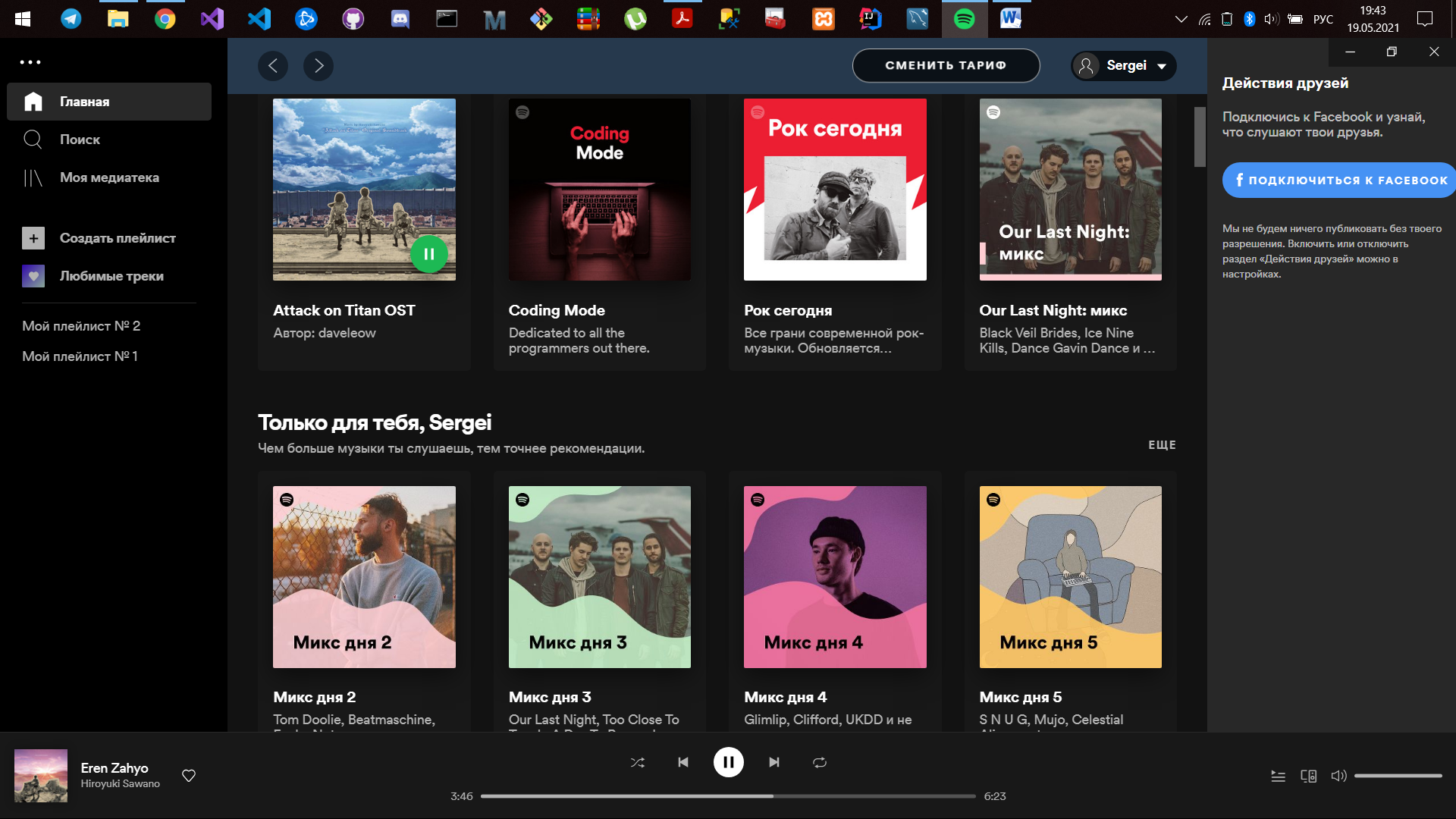
* 1. Обзор литературы

При написании курсового проекта использовалась дополнительная литература, а также разнообразные интернет источники. Источниками основных теоретических и практических сведений является труд Джеффри Рихтера «CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#» и работа Эндрю Троелсона и Филиппа Джепикса «Язык программирования C#7 и платформы .NET и .NET Core». Данные книги дают довольно полное представление о языке программирования C# и платформе .NET.

Не менее используемыми источниками информации были интернет-ресурсы: <https://metanit.com/>, <https://docs.microsoft.com/en-us/dotnet/csharp/> и <https://habr.com/ru/>. Они обладают довольно обширной базой информации по предметной области в довольно доступной и сжатой форме.

* 1. Обзор прототипов и аналогов

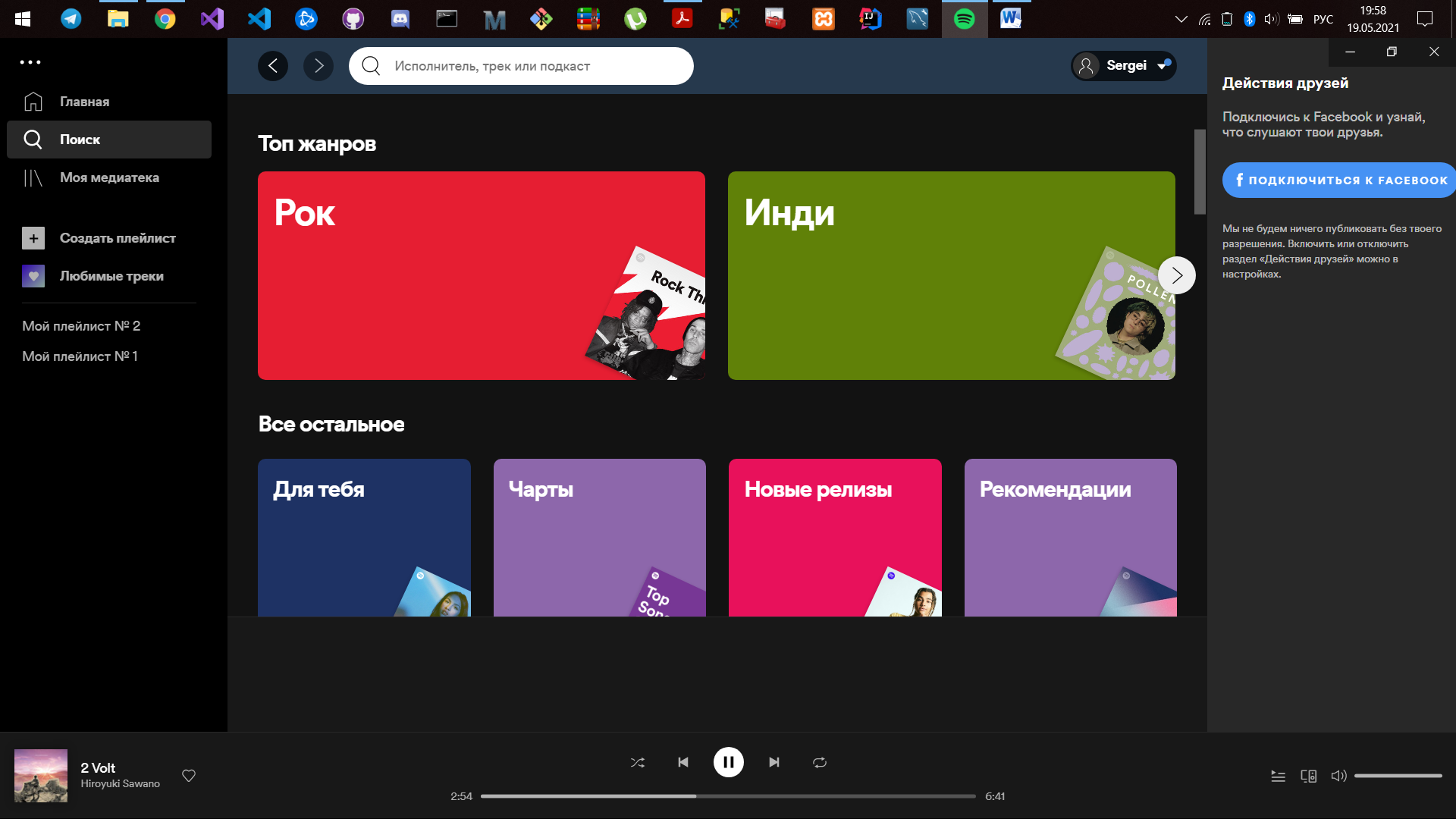
За основу моего приложения был взят музыкальный сервис Spotify. Данное приложение представляет из себя интернет-сервис потокового аудио и предоставляет возможности прослушивания музыки, аудиокниг и подкастов, не скачивая их на устройство. Доступ к нему можно получить через сайт, приложения для различных ОС, а также через приложения для мобильных устройств и медиа-систем автомобилей. Естественно из всего выше предложенного многообразия средств за шаблон было взято настольное приложения под Windows.



## Рисунок 1.1 – Окно приложения «Spotify».

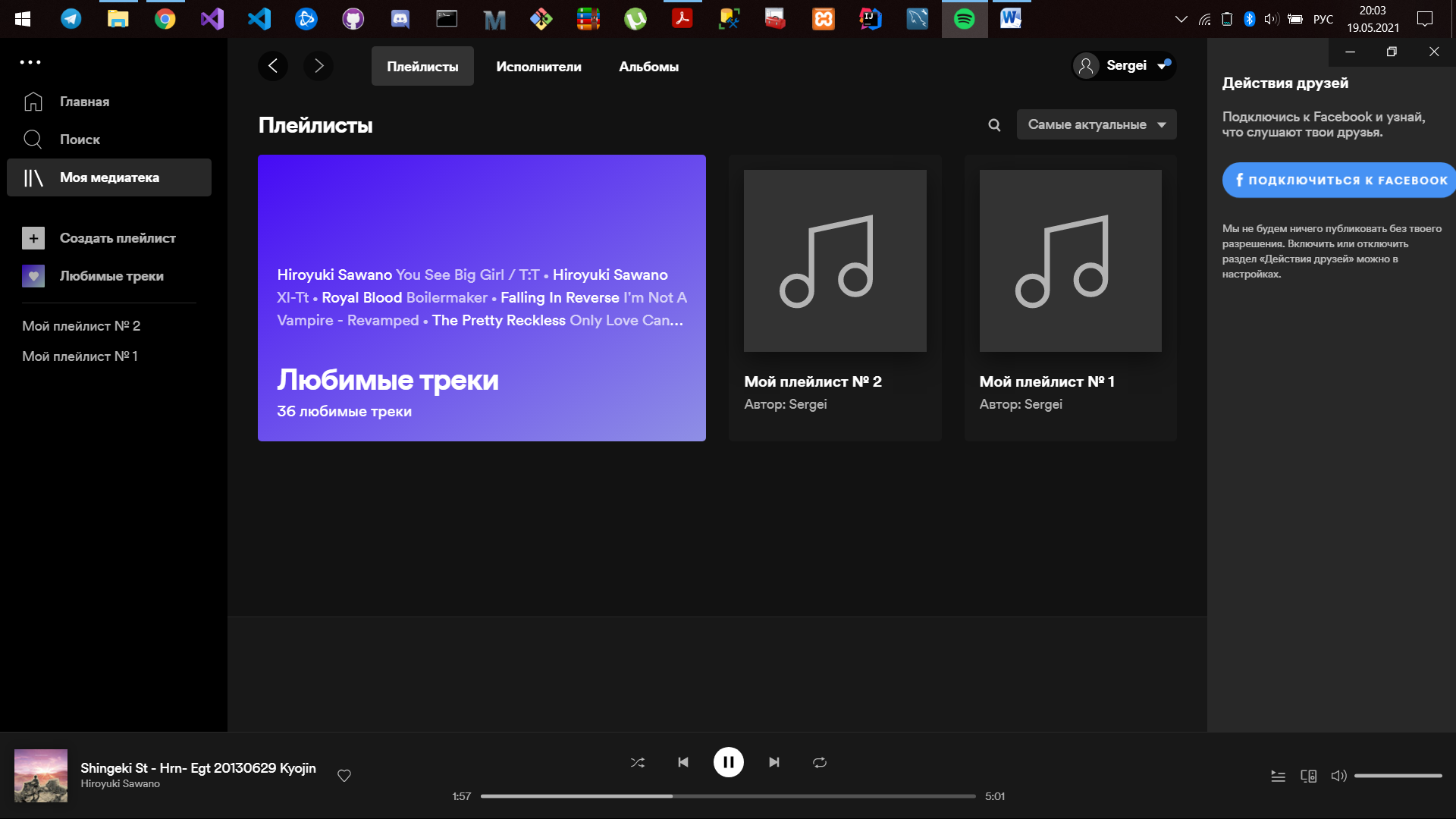
Снизу расположен плеер приложения при помощи которого пользователь имеет возможность управлять прослушиванием трека: изменять громкость, ставить на паузу, перематывать. Слева расположена панель меня, через которое производится навигация по приложению. В центральной части окно происходит отображения основного контента: треков, плэйлистов и т.д.

Имеется возможность как пользовательского поиска так и по жанрам.



## Рисунок 1.2 – Поиск в приложении «Spotify».

По статистике 70% процентов контента воспроизводится из плейлистов. И в приложении предусмотрена пользовательская медиатека, которая предоставляет возможности: управления плэйлистами, создания пользовательских, а также подписываться на понравившихся исполнителей или добавлять любимые альбомы.



## Рисунок 1.3 – Пользовательская медиатека.

Данное приложение обладает платной подпиской которая позволяет избавится от назойливой рекламы, убрать ограничение скипов, получить возможность скачивать треки и слушать их оффлайн.

Я решил взять за основу данное приложение так, как оно по сути наиболее близко к моему: обладает функциями прослушивания музыки и составления/прослушивания плэйлистов. Также оно обладает довольно приятным и в своё время простым дизайном.

### Анализ требований к курсовому проекту и разработка функциональных требований

* 1. Постановка задачи

Задачей курсового проекта является разработка приложения, предназначенного для прослушивания музыки с дополнительными возможностями: создание и прослушивание плейлистов, оценивание треков и добавление их в избранные. Коллекция треков пополняется и редактируется администратором. Администратор также производит управление пользователями и обновляет коллекцию популярных треков.

Для разработанного продукта необходимо предусмотреть следующие задачи:

* Графический интерфейс пользователя;
* Механизм авторизации;
* Отображение коллекции треков;
* Механизм управления контентом для администратора;
* Механизм поиска треков;
* Механизм создания и прослушивания плейлистов для пользователя.

Приложение необходимо реализовать с использованием шаблона проектирования MVVM и с организацией взаимодействия с базой данных. В качестве клиентской части необходимо реализовать оконное приложение с использованием стандартных и иных библиотек.

* 1. Средства разработки

Программный продукт должен быть реализовать на объектно-ориентированном языке программирования C# с использованием технологии WPF. Windows Presentation Foundation (WPF) — система для построения клиентских приложений Windows, с визуально привлекательными возможностями взаимодействия с пользователем; графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования. 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.

Графической технологией, лежащей в основе WPF, является DirectX, в отличие от Window Forms, где используется GDI/GDI+. Производительность WPF выше, чем у GDI+ за счёт использования аппаратного ускорения графики через DirectX.

База данных была разработана в Microsoft SQL Server.

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире.

SQL Server характеризуется такими особенностями как:

* Производительность. SQL Server работает очень быстро.
* Надежность и безопасность. SQL Server предоставляет шифрование данных.
* Простота. С данной СУБД относительно легко работать и вести администрирование.

Для организации баз данных MS SQL Server использует реляционную модель, которая предполагает хранение данных в виде таблиц, каждая из которых состоит из строк и столбцов. Каждая строка хранит отдельный объект, а в столбцах размещаются атрибуты этого объекта.

Для взаимодействия с базой данных применяется язык SQL (Structured Query Language). Клиент (например, внешняя программа) отправляет запрос на языке SQL должным образом интерпретирует и выполняет запрос, а затем посылает клиенту результат выполнения.

Основной используемый язык запросов —Transact-SQL — реализован на структурированном языке запросов (SQL) с расширениями.

Для осуществления связи между базой данных и приложением на C# необходим посредник, которым будет являться технология Entity Framework. Она основана на платформе .NET Framework и предназначена для работы с данными.

Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Эта технология предоставляет нам набор классов, через которые мы можем отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

Entity Framework предполагает три возможных способа взаимодействия с базой данных:

* Database first: Entity Framework создает набор классов, которые отражают модель конкретной базы данных;
* Model first: сначала разработчик создает модель базы данных, по которой затем Entity Framework создает реальную базу данных на сервере;
* Code first: разработчик создает класс модели данных, которые будут храниться в базе данных, а затем Entity Framework по этой модели генерирует базу данных и ее таблицы.

В курсовом проектировании для разработки был выбран третий подход.

По итогу работы над продуктом, его необходимо протестировать и отладить, проанализировать его возможности и, по необходимости, внести дополнения или улучшения проекта, с возможностью дальнейшего расширения проекта.

* 1. Функциональные возможности приложения

Программное средство предоставляет пользователю следующие функциональные возможности:

* Прослушивание треков;
* Поиск треков;
* Создание плейлистов;
* Прослушивание плейлистов;
* Редактирование плейлистов;
* Редактирование приватной информации пользователя(смена пароля);
* Возможность оценки треков;
* Добавление треков в избранные;

Возможности администратора:

* Добавление и удаление треков;
* Редактирование информации о треках;
* Обновление коллекции популярных треков;
* Удаление и блокировка пользователей;
* Передача прав администратора;

Наглядно это видно на UML-схеме, предоставленной на рисунке 2.1 и в приложении А.

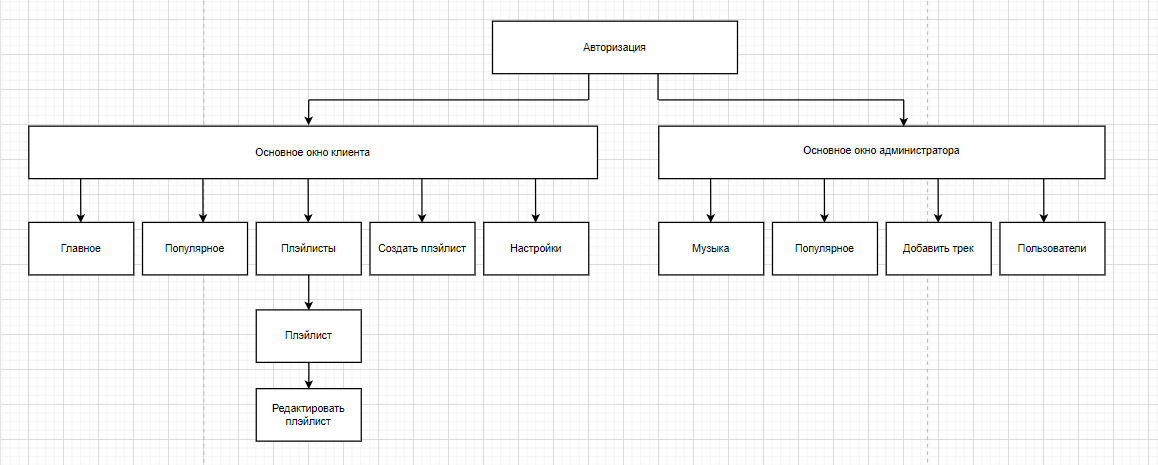


## Рисунок 2.1 – UML-диаграмма возможностей

### Проектирование программного средства

* 1. Схема работы приложения

На рисунке 3.1 представлена общая схема работы приложения.



## Рисунок 3.1 – Общая схема работы приложения.

При запуске приложения появляется окно авторизации, которое требует ввода логина и пароля для начала работы.

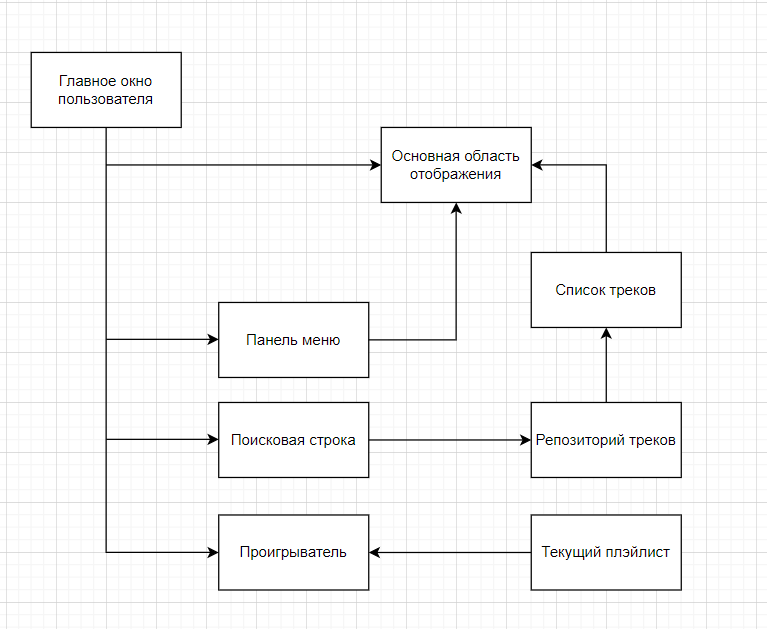
В случае, если вас нету своей учётной записи, её можно создать, нажав на кнопку «Регистрация» в окне авторизации. Вы будете направлены в окно регистрации, где вам нужно будет заполнить все поля. После успешной регистрации, вас перенаправит в окно авторизации для ввода ваших данных.

Также есть кнопка перехода на вход в качестве администратора, где вы сможете, если ваш пользователь обладает должными правами, войти в окно управления приложения и воспользоваться возможностями администратора.

После ввода логина и пароля происходит их проверка и в случае успеха вам откроется основное окно программы, иначе будет выведено сообщение об ошибке.

Главное окно приложения предоставляет возможность навигации по составным компонентам текущего окна (страницам).

Рассмотрим главное окно пользователя в деталях. Оно содержит мини-плеер для управления текущим проигрыванием трека, элемент управления для поиска треков, панель меню для навигации по странице и основную часть, в которой собственно отображаются страницы.



## Рисунок 3.2 – Главное окно пользователя.

При первоначальном переходе в данное окно, отображается главная страница. Далее уже в зависимости от действий пользователя может происходить переход на другие страницы, через панель меню. Пользователь также может искать треки через поисковую строку, результат будет отображаться на главной странице приложения.

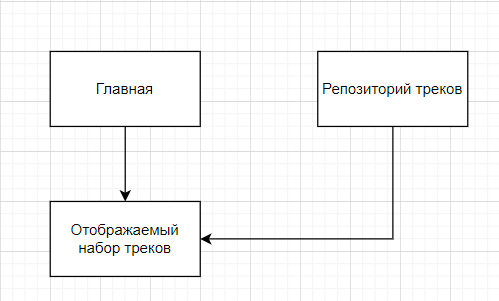
Также пользователь обладает возможностью контролировать процесс воспроизведения треков с помощью проигрывателя. Проигрыватель воспроизводит треки из текущего набора, имеется возможность перехода на следующий или предыдущий трек, регулировки звука. Пользователь может изменить проигрываемый трек и собственно проигрываемый набор, кликнув по треку из набора в главной области программы.

Панель меню содержит кнопки перехода на следующие страницы:

* Главная;
* Популярное;
* Плэйлисты;
* Создать плэйлист;
* Настройки.

Рассмотрим каждую страницу по отдельности.

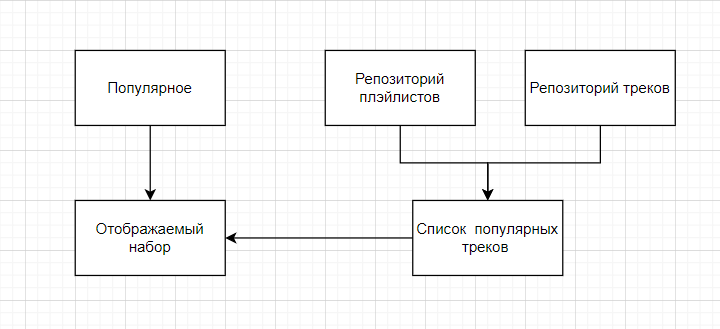
Главная страница показана на рисунке 3.3.



## Рисунок 3.3 – Главная страница.

Главная страница отображает список всех треков из коллекции.

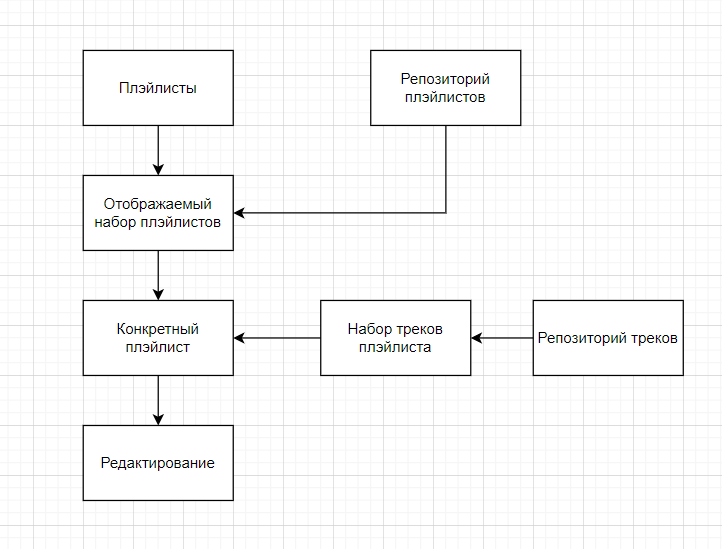
На рисунке 3.4 показана страница «Популярное».



## Рисунок 3.4 – Страница «Популярное».

Данная страница отображает формируемый администратором список популярных треков.

На рисунке 3.5 изображена схема страницы плэйлисты.

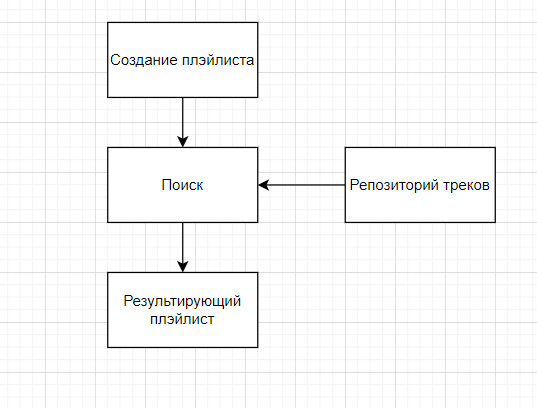


## Рисунок 3.5 – страница «Плэйлисты».

Страница «Плэйлисты» отображает набор плэйлистов соответствующий данному пользователю (в том числе и избранные треки). Отсюда пользователь может перейти к конкретному плэйлисту, запустить его прослушивание, а также перейти на страницу редактирования.

Со страницы редактирования пользователь может изменить информацию о плэйлисте, удалить его или изменить его содержимое.

На рисунке 3.6 изображена схема страницы создания плэйлита.



## Рисунок 3.6 – Создание плэйлиста.

На странице создания плэйлиста пользователь заполняет все необходимые данные, ищет треки и добавляет их в плэйлист. После создания плэйлиста произойдёт переадресация на страницу отображающую все плэйлисты.

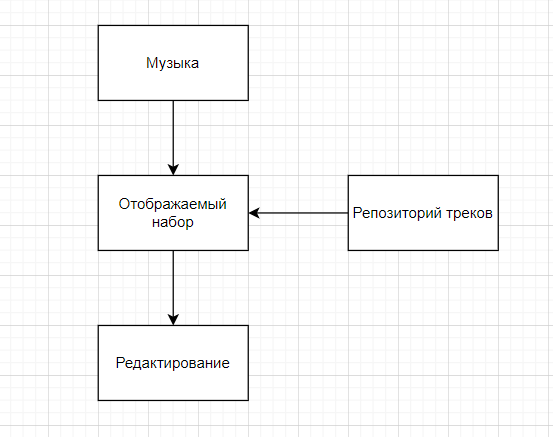
Через страницу настроек пользователь имеет возможность изменить пароль, удалить аккаунт или выйти из текущего аккаунта.

Рассмотрим главное окно администратора. Оно состоит из панели меню и области отображения. С помощью панели меню происходит страничная навигация.

Панель меню содержит следующие страницы:

* Музыка;
* Популярное;
* Добавить трек;
* Пользователи;

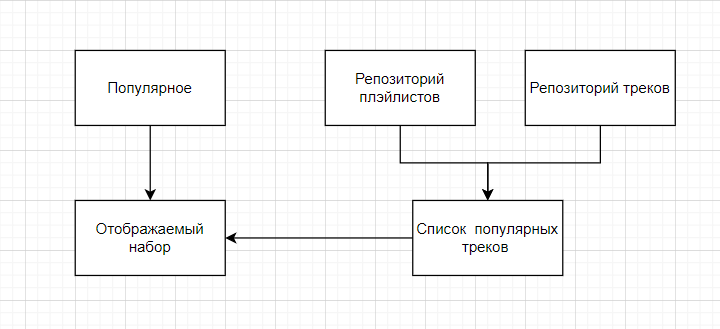
Рассмотрим каждую страницу отдельно. На рисунке 3.7 изображена схема страницы «Музыка».



## Рисунок 3.7 – страница «Музыка»

На данной странице отображается список всех треков. Администратор может удалять треки и списка, а также редактировать, для этого выделена отдельная страница. На странице редактирования администратор может изменить информацию о треке.

На рисунке 3.8 изображена схема страницы популярное.

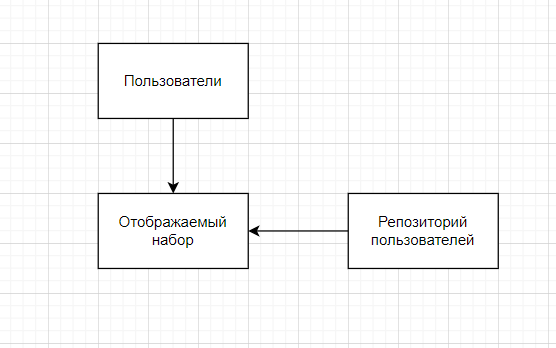


## Рисунок 3.8 – Популярные треки

На данной странице отображается список 10 популярных треков и соответственно администратор может изменять его при устаревании.

Страница добавления трека позволяет заполнить необходимые данные и добавить трек в коллекцию.

На рисунке 3.9 показана страница «Пользователи».



## Рисунок 3.9 – страница «Пользователи»

На странице «Пользователи» предоставляется возможность просмотра списка пользователей, удалять их, блокировать и разблокировывать, передавать права администратора другим пользователям.

* 1. Структура пакетов приложения

В таблице 3.1 описана структура проекта по пакетам классов.

Таблица 3.1 – Описание структурных пакетов проекта

|  |  |
| --- | --- |
| Имя пакета | Описание |
| Models | Здесь описаны модели, с которыми происходит вся работа в приложении:   * Пользователь * Трек * Плэйлист   Также в данной папке содержатся вспомогательные классы для описания модели, класс контекста БД и класс инициализатор БД. |
| Commands | Здесь описана реализация Command используемая в приложении. |
| Converters | Содержит вспомогательные классы конвертеры, используемые при приобразании данных в приложении. |
| Hasher | Содержит класс реализующий алгоритм хеширования md5. |
| Icon | Содержит иконку для окон приложения. |
| Repository | Содержит классы и интерфейсы для работы с данными БД. Они являются связующим элементом между данными в БД и теми данными, с которыми работает приложение |
| Resources | Файлы стилей. |
| Views | Содержит все представления, которые позволяют пользователю работать с приложением. Они описывают графическую составляющую приложения. |
| ViewModels | Содержит логику, которая позволяет получить данные при помощи View, обработать их, использую при этом Model, после чего передать в базу данных. |
| App.config | Файл конфигурации приложения |
| App.xaml | Основная задача данного файла состоит в определении ресурсов, общих для приложения |

В этой таблице приведены основные логические составляющие пакеты, которые используются в работе приложения.

Структуры данных, а также их зависимости друг от друга представлены в приложении Б.

Диаграмма классов показана в приложении Г.