Министерство образования Кузбасса

Государственное профессиональное образовательное учреждение

«Сибирский политехнический техникум»

|  |
| --- |
|  |

РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ «МУЗЫКАЛЬНАЯ

ПЛАТФОРМА»

Пояснительная записка

КП 09.02.07.00.00.00

Выполнил:

студент группы 2ПР-20

Петелин С.В.

Руководитель КП:

Щербакова К.А.

Защита КП:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка)

Кемерово, 2023

Министерство образования Кузбасса

Государственное профессиональное образовательное учреждение

«Сибирский политехнический техникум»

|  |  |
| --- | --- |
| СОГЛАСОВАНО:  Председатель цикловой  методической комиссии  \_\_\_\_\_\_\_\_\_\_\_К.И. Морозько  «\_\_\_\_»\_\_\_\_\_\_\_\_\_ 20 г. | УТВЕРЖДАЮ:  Заместитель директора  по УПР  \_\_\_\_\_\_\_\_\_\_\_Т.М. Жильцова  «\_\_\_\_»\_\_\_\_\_\_\_\_\_ 20 г. |

ЗАДАНИЕ

для курсового проекта

студента: Петелина Семена Валерьевича

группы 2ПР-20

специальности 09.02.07 «Информационные системы и программирование»

Тема КП: Разработка программного модуля «Музыкальная платформа»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Введение

1 Теоретические аспекты разработки программного модуля

1.1 Общая характеристика приложения

1.2 Обзор инструментов для реализации настольного приложения

1.3 Анализ предметной области

2 Разработка программного модуля

2.1 Функциональные требования

2.2 Диаграмма вариантов использования

2.3 Проектирование интерфейса

2.4 Проектирование базы данных

2.5 Реализация программного модуля

Заключение

Список литературы

Дата выдачи задания « » \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Дата окончания курсового проекта « » \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Щербакова К.А.

« » \_\_\_\_\_\_\_\_\_\_\_\_ 2023

СОДЕРЖАНИЕ

[Введение 4](#_Toc134790711)

[1 Теоретические аспекты разработки программного модуля 5](#_Toc134790712)

[1.1 Общая характеристика приложения 5](#_Toc134790713)

[1.2 Обзор инструментов для реализации настольного приложения 7](#_Toc134790714)

[1.3 Анализ предметной области 10](#_Toc134790715)

[2. Разработка программного модуля 11](#_Toc134790716)

[2.1 Функциональные требования 11](#_Toc134790717)

[2.2 Диаграмма вариантов использования 11](#_Toc134790718)

[2.3 Проектирование интерфейса 13](#_Toc134790719)

[2.4 Проектирование базы данных 15](#_Toc134790720)

[2.5 Реализация программного модуля 16](#_Toc134790721)

[Заключение 18](#_Toc134790722)

[Список литературы 19](#_Toc134790723)

# Введение

С каждым днем популярность музыкальных стриминговых сервисов возрастает, но в основном именно музыкальное приложения имеются только в качестве мобильного приложения, а для пользования на персональных компьютерах (ПК) нужно заходить на специальные веб-приложения этих сервисов. Ввиду не удобства данного подхода, стоит разработать альтернативу в виде настолько приложения.

Цель курсового проекта: разработать программный модуль «Музыкальная платформа».

Для этого необходимо решить следующие задачи:

* ознакомиться с общей характеристикой приложения;
* провести обзор инструментов для реализации настольного приложения;
* провести анализ предметной области;
* разработать функциональные требования к программному модулю;
* выполнить проектирование настольного приложения и базы данных;
* реализовать настольное приложение.

Курсовая состоит из 2х глав.

В первой главе рассказывается о теоретических аспектах разработки программного модуля, а во второй главе рассматривается реализация программного модуля.

# 1 Теоретические аспекты разработки программного модуля

1. Общая характеристика приложения

Настольное приложение - это компьютерная программа, используемая в повседневной работе. Приложение можно запустить прямо с рабочего стола компьютера или ноутбука.

Приложения можно установить, скачав с официального сайта разработчиков или со стороннего сайта установочный EXE-файл или установить приложения с помощью Microsoft Store.

Настольные приложения имеют несколько функций, могут выполнять несколько задач и имеют следующие возможности:

* работа без интернета: приложение работает автономно, сохраняет все данные в память компьютера. Это гарантирует безопасность данных и высокую производительность;
* работа с интернетом: использование облака для сохранности данных, доступ нескольких пользователей к одному проекту, автономное обновление программы без переустановки или длительных настроек. При необходимости, можно работать в локальной сети;
* быстрый запуск: Настольное приложение запускает системные файлы из памяти компьютера и не нуждается в постоянном обновлении данных и загрузки параметров из сети. Поэтому при использовании нормального оборудования, даже мощные приложения запускаются быстро;
* качественный пользовательский интерфейс: Не всем нужны возможности настроить интерфейс «под себя», но программы на ПК обычно позволяют это делать;
* использование дополнительной периферии. Десктоп приложения имеют доступ ко всем устройствам, которые подключаются к компьютеру. Поэтому их без проблем и можно интегрировать с принтерами, сканерами, фискальными аппаратами и другими периферийными устройствами.

Рассмотрим некоторые теоретические основы, которые могут быть применены при разработке программного модуля.

1. Проектирование информационной системы. Первым шагом при разработке компьютерного приложения является проектирование информационной системы. Этот этап включает в себя анализ требований пользователей, проектирование базы данных, определение архитектуры приложения и разработку технического задания.

2. Методы разработки программного обеспечения. Для разработки компьютерного приложения могут быть использованы различные методы разработки программного обеспечения. Каждый метод имеет свои особенности и выбор метода зависит от конкретных требований и условий проекта.

3. Языки программирования. При разработке компьютерного приложения можно использовать различные языки программирования, такие как C#, Java, Python, JavaScript и т.д. Выбор языка программирования зависит от требований проекта, опыта разработчиков и других факторов.

4. Базы данных. База данных является одним из важных компонентов компьютерного приложения. При выборе базы данных необходимо учитывать требования проекта, объемы данных и другие факторы. Для работы с базами данных можно использовать различные технологии, такие как SQL, MySQL и т.д.

5. Интерфейс пользователя. Интерфейс пользователя является важным компонентом компьютерного приложения, который влияет на удобство и эффективность работы с приложением. При разработке интерфейса необходимо учитывать принципы удобства, интуитивности и простоты использования.

6. Тестирование и отладка. Тестирование и отладка являются важными этапами при разработке компьютерного приложения. На этом этапе проверяется работоспособность приложения, выявляются ошибки и недочеты, которые нужно исправить.

В целом, при разработке компьютерного приложения необходимо учитывать множество теоретических основ, которые позволят создать качественное и эффективное приложение.

Разработка настольных приложений действительно имеет следующие преимущества:

* функциональность: широкие возможности для функционала. Можно реализовать практически любую идею. При этом интерфейс будет удобным, привычным и интуитивно понятным для пользователей.
* безопасность и надежность: браузерные приложения уязвимы. Дело в том, что разработчики не могут повлиять на безопасность браузера, а веб-приложение напрямую зависит от его работы. Десктоп приложение – это гарантия безопасности и сохранности данных.
* мобильность: настольные приложения можно разрабатывать под различные операционные системы и даже использовать с мобильного устройства, сделав нативную версию. Но для синхронизации данных необходимо подключаться к интернету. Тем не менее, мобильность подразумевает использование приложения в любых условиях и дает возможность управлять бизнес-процессами компании, когда угодно.
* быстродействие: программа использует ресурсы компьютера, его память и возможности. С одной стороны, это может стать недостатком, если оборудование недостаточно хорошее. Но с другой, заказывая индивидуальную разработку, учитываются особенности системы. С должной оптимизацией такое приложение будет «летать» и выполнять задачи эффективно.
  1. Обзор инструментов для реализации настольного приложения

Разработка настольных (или десктопных) приложений – это процесс, при котором приложения разрабатываются для настольного компьютера.

Рассмотрим примеры инструментов для реализации настольного приложения под операционные системы Windows. На сегодняшний день существует множество платформ для создания Windows приложений, проанализируем несколько, самых известных и актуальных:

1. Windows Forms;
2. WPF;
3. Qt;
4. WxWidgets;
5. Swing.

Windows Forms

Windows Forms – это интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью платформы Microsoft .NET Framework. Классы, реализующие интерфейс программирования приложения, не зависят от языка разработки.

В Windows Forms форма – это визуальная поверхность, на которой выводится информация для пользователя. Приложение можно создать с помощью перетаскивания и вставки конструктор Windows Forms в Visual Studio. Также Windows Forms включает широкий набор элементов управления, которые можно добавлять на формы: текстовые поля, кнопки, раскрывающиеся списки, переключатели, веб-страницы.

WPF

WPF – система для построения клиентских приложений Windows. В основе WPF лежит векторная система визуализации, которая не зависит от разрешения устройства и которая создана с учётом возможностей современного графического оборудования. Для работы с WPF требуется любой .NET-совместимый язык. WPF предоставляет средства для создания визуального интерфейса, элементыуправления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

Qt

Qt – кроссплатформенный фреймворк для разработки программного обеспечения на языке программирования C++. Существуют также «привязки» ко многим другим языкам программирования:

* Python – PyQt;
* Ruby – QtRuby;
* Java – QtJambi;
* PHP – PHP-Qt.

Отличительная особенность этого фреймворка – использование предварительной обработки исходного кода, что позволяет запускать приложения в большинстве операционных систем. Qt полностью объектно-ориентированный и включает в себя все основные классы, которые требуются при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML.

WxWidgets

WxWidgets (wxWindows) – библиотека инструментов с открытым исходным кодом для разработки кроссплатформенных на уровне исходного кода приложений. Основное применение данной библиотеки является построение графического интерфейса пользователя [10]. WxWidgets позволяет легко реализовать стандартные интерфейсные функции и тем самым сконцентрировать внимание разработчика на функциональных особенностях приложения. Библиотека написана на языке программирования C++, но может подключаться ко множеству других языков, таких, как Ruby, Python и др.

Swing

Swing – библиотека для создания графического интерфейса для программ на языке Java. Данная библиотека содержит ряд графических составляющих, таких как кнопки, поля ввода, таблицы и т.д. Swing относится к библиотеке классов JFC, которая представляет собой набор библиотек для разработки графических оболочек. К этим библиотекам относятся Java 2D, Accessibility-API, Drag & Drop-API и AWT. Архитектура Swing разработана таким образом, что разработчики могут изменять внешний вид компонентов вашего приложения и их поведение.

Вывод

Исходя из анализа существующих решений было принято решение разрабатывать настольное приложение при помощи WPF в среды программирования VisualStudio на языке программирования C#. Данный язык и среда являются универсальными инструментами программирования, поэтому они подходят для решения поставленной задачи по созданию ИС. В качестве СУБД для разработки системы был выбран Microsoft SQL Server 2019.

* 1. Анализ предметной области

На сегодняшний день существует несколько решений прослушивания музыки в виде настольного приложения.

Ниже приведены примеры данных приложений.

1. Яндекс.Музыка

Яндекс.Музыка (Я.Музыка) - это стриминговый сервис для просушивания музыки в онлайн или офлайн режиме, разработанный российской компанией Яндекс. Сервис имеет три вариации использования: мобильное приложение, веб-приложение, настолько приложение. Интерфейс приложения можно увидеть на рисунке 1.1.

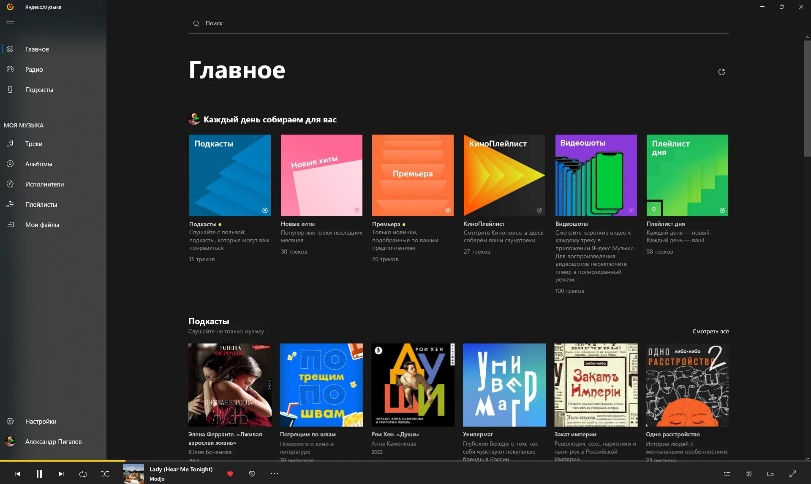


Рисунок 1.1 - Приложение Я.Музыка

Достоинства:

1. Удобный и понятный интерфейс.
2. Удобная подборка песен под пользователя.

Основные недостатки:

1. Наличие только одного вида подписки.
2. Настольное приложение имеется только на операционной системе (ОС) Windows.

2. Spotify

Spotify - стриминговый сервис для просушивания музыки в онлайн или офлайн режиме, разработанный шведской компанией Spotify. Сервис так же имеет три вариации: мобильное приложение, веб-приложение и настольное приложение. Интерфейс приложения можно увидеть на рисунке 1.2.

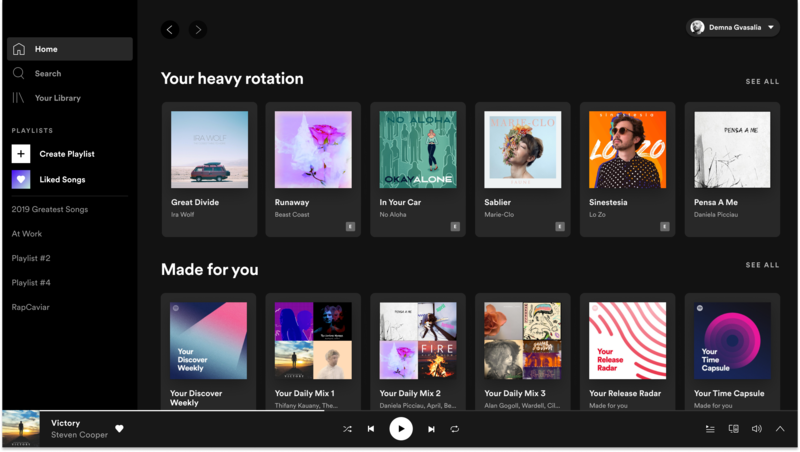


Рисунок 1.2. Приложение Spotify

Достоинства Spotify:

1. Удобный интерфейс.
2. Удобная подборка песен под пользователя.
3. Разнообразие подписок: семейная, обычная, для студентов.
4. Кроссплатформенность (распространена на ОС Windows, macOS, а также ОС, основанных на ядре Linux (к примеру: Ubuntu, Linux Mint)).

Недостатки:

1. Официальное отсутствие в России (доступ только через VPN).

В данный момент Spotify официально не доступен в России.

Вывод

Рассмотрев аналоги можно сделать вывод, что в данной сфере есть конкурентоспособные решения, однако все из предложенных выполнены в качестве платных веб-сервисов, а полного аналога настольного приложения для администратора не существует.

# Разработка программного модуля

## 2.1 Функциональные требования

В результате анализа предметной области были сформулированы функциональные требования к приложению.

Настольное приложение должно предоставлять пользователю функциональные возможности:

* Регистрации нового пользователя
* Вход в уже созданный аккаунт
* Редактирование данных пользователя
* Прослушивание музыки

## 2.2 Диаграмма вариантов использования

В данной системе существует только один актер, взаимодействующий с приложением, – Администратор, использующий приложение для учета совместных покупок.



Рис. 4. Диаграмма вариантов использования

В разрабатываемом приложении возможны следующие варианты использования:

1. Прослушивание музыки, добавленным администратором

## 2.3 Проектирование интерфейса

Программа должна быть не только функционально полезна, а также иметь логический, приятный и удобный UI, который не будет сильно напрягать глаза пользователя.

Разрабатываемая программа имеет интерфейс в розовых приятных оттенках с сочетанием белого цвета с розовой тенью. Пример представлен на рисунке 2.2.

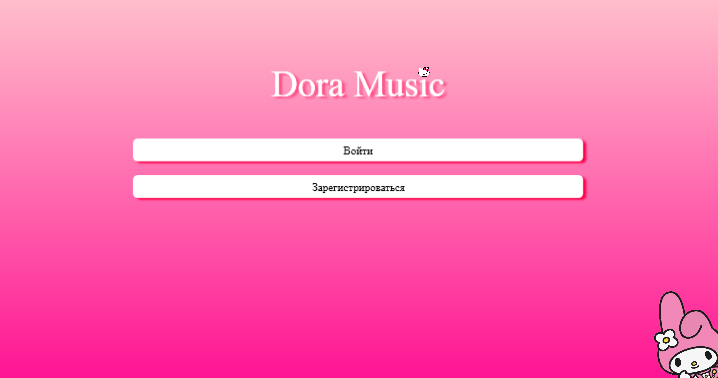


Рисунок 2.2 – Пример страницы

При проектировании интерфейса было принято решение добавить кнопку возвращение на предыдущую страницу, если логически на главную страницу не вернуться, то вместо кнопки «назад» появляется кнопка «выход» при нажатии на которую пользователь выходит из аккаунта.

## 2.4 Проектирование базы данных

Схема базы данных включает в себя описания содержания, структуры и ограничений целостности, используемые для создания и поддержки базы данных. Система управления базами данных (СУБД) использует определения данных в схеме для обеспечения доступа и управления доступом к данным в базе данных.

На схеме (рисунок 2.3) продемонстрированы связи для каждой из таблиц и их элементы.

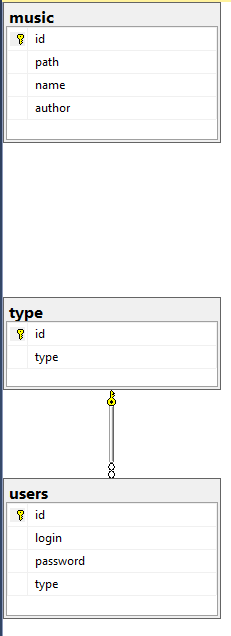


Рисунок 2.3 − ER-диаграмма

База данных представлена следующими таблицами:

1. Таблица «Music» (рисунок 2.4). Содержит информацию об песне:

* ID (Индетификатор)– целочисленный тип, является первичным ключом. Содержит читательский номер ученика;
* Path (Название файла)- символьный тип;
* Name (Название песни) - символьный тип;
* Author (Автор) - целочисленный тип.

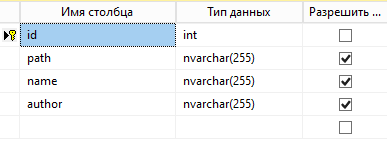


Рисунок 2.4 – Таблица «Students»

1. Таблица «Users» (рисунок 2.5). Содержит информацию о пользователях:

* ID (Идентификатор)– целочисленный тип, является первичным ключом;
* Login (Логин) − символьный тип;
* Password (Пароль) − символьный тип;
* Type (Тип) − целочисленный тип.

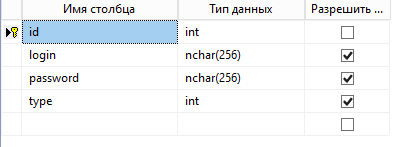


Рисунок 2.5 – Таблица «Users»

1. Таблица «Type» (рисунок 2.6). Содержит информацию о пользователях:

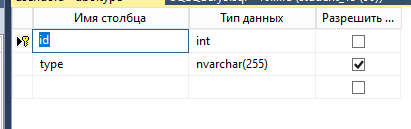
* ID (Идентификатор)– целочисленный тип, является первичным ключом;
* Type (Название типа пользователя) − символьный тип;
* 

Рисунок 2.6 – Таблица «Type»

## 2.5 Реализация программного модуля

При запуске приложения нас встречает окно, с выбором между авторизацией/регистрацией (рисунки 2.7, 2.8, 2.9)

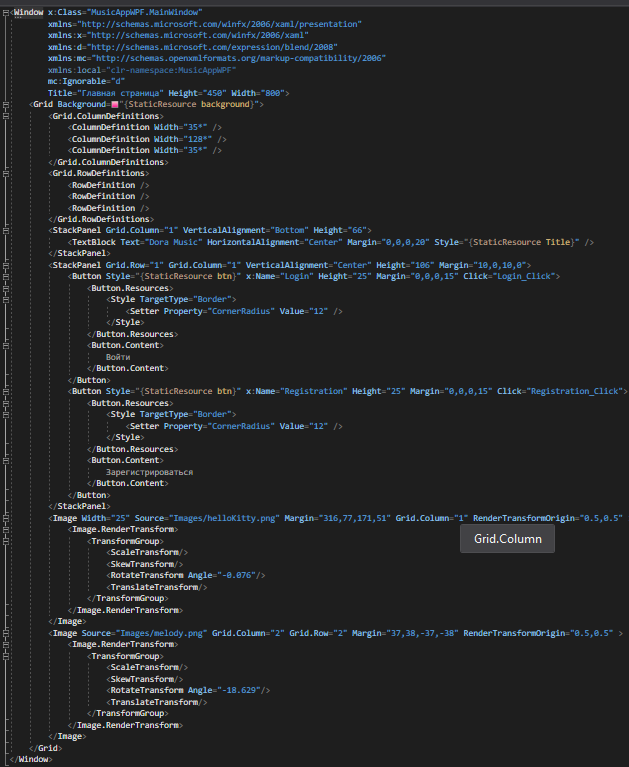


Рисунок 2.7 – XAML разметка окна

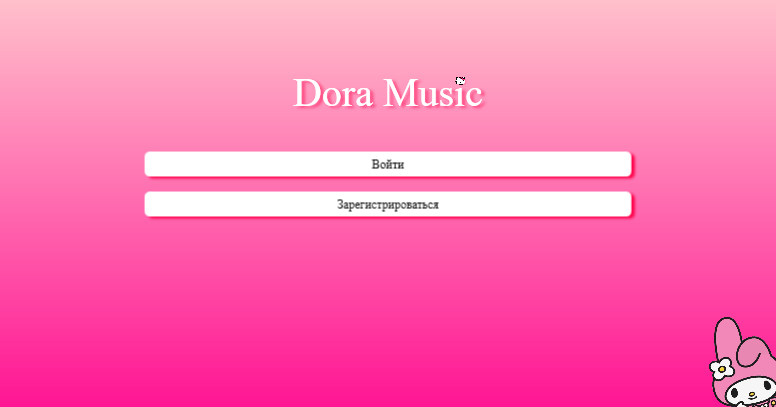


Рисунок 2.8 – Окно страницы



Рисунок 2.9 – CS код окна

При нажатии на кнопку «Регистрация» пользователя встречает окно с формой регистрации (Рисунки 2.10, 2.11, 2.12).

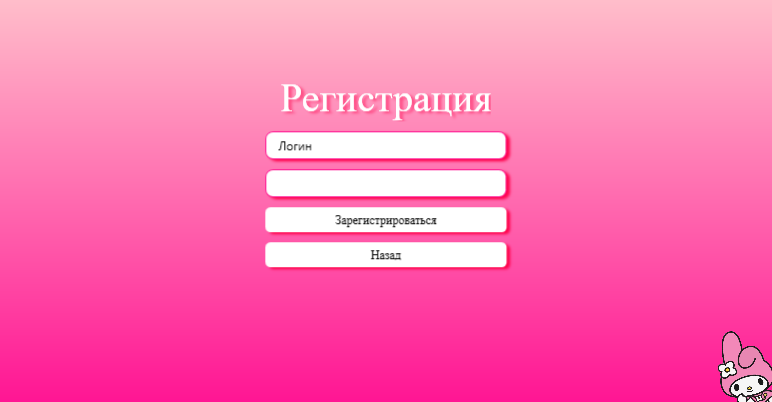


Рисунок 2.10 – Окно страницы

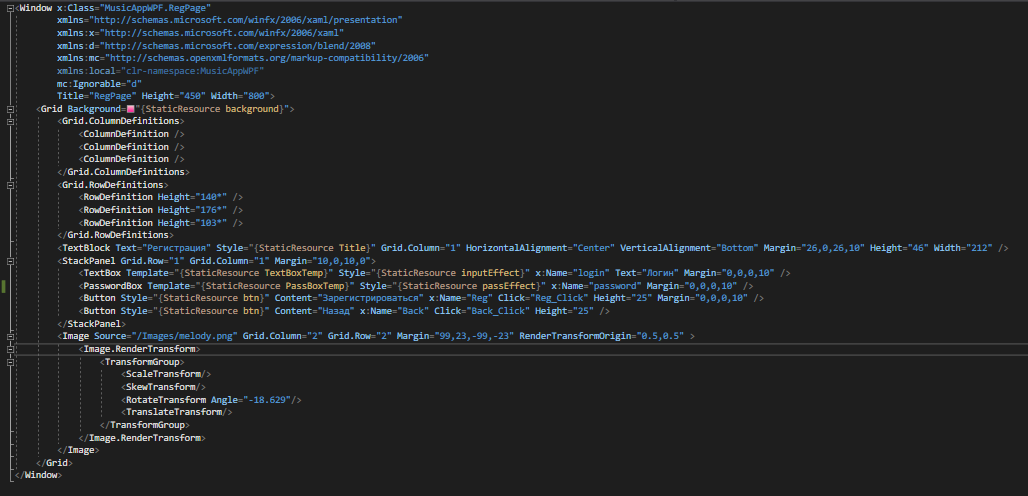


Рисунок 2.11 – XAML разметка окна

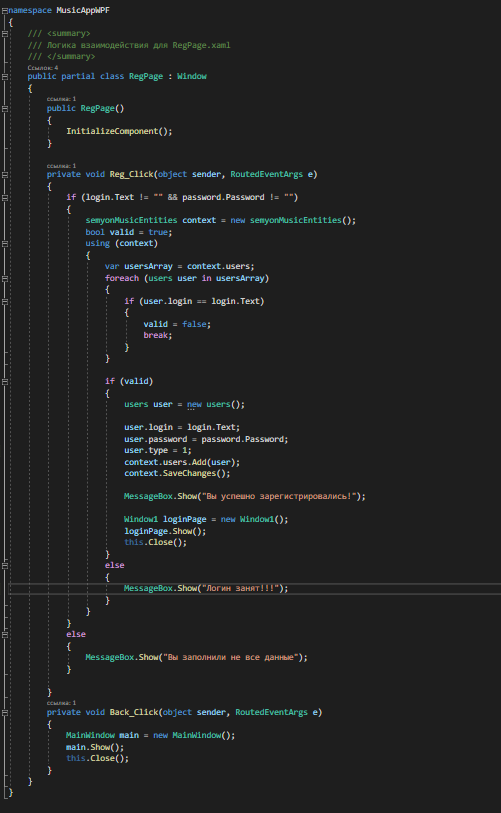


Рисунок 2.12 – CS код окна

После заполнения формы и нажатия кнопки «Зарегистрироваться», нам показывается окно «Вы успешно зарегистрировались», после чего пользователь переходит на страницу входа. При нажатии кнопки «Назад» пользователя перемещает на начальную страницу.

Страница входа показана на рисунках 2.13,2.14,2.15.

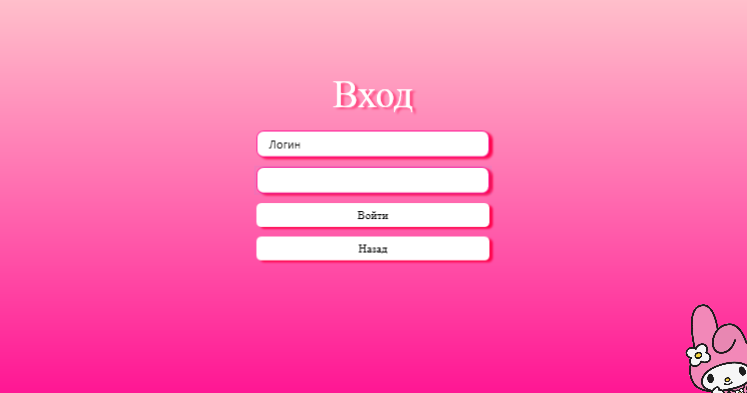


Рисунок 2.13 – Окно входа

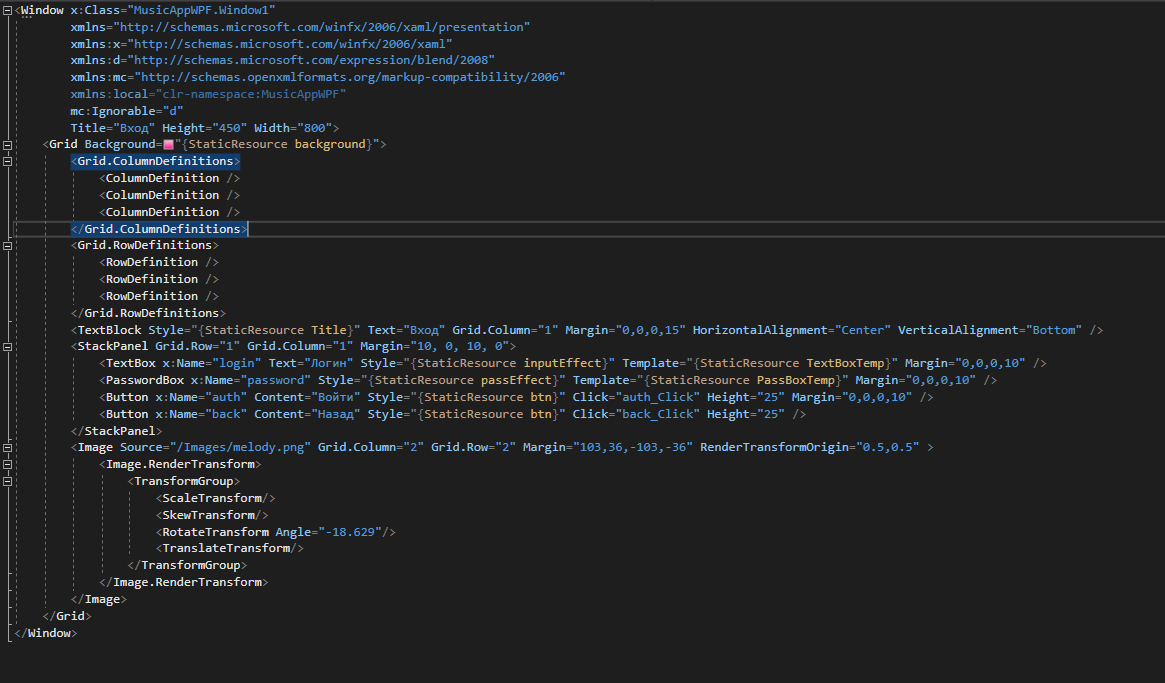


Рисунок 2.14 – XAML разметка окна



Рисунок 2.15 – CS код окна

После входа пользователя перекидывает на окно профиля, где имеется две кнопки «Вся музыка», а также «Настройки» (Рисунки 2.16, 2.17, 2.18)

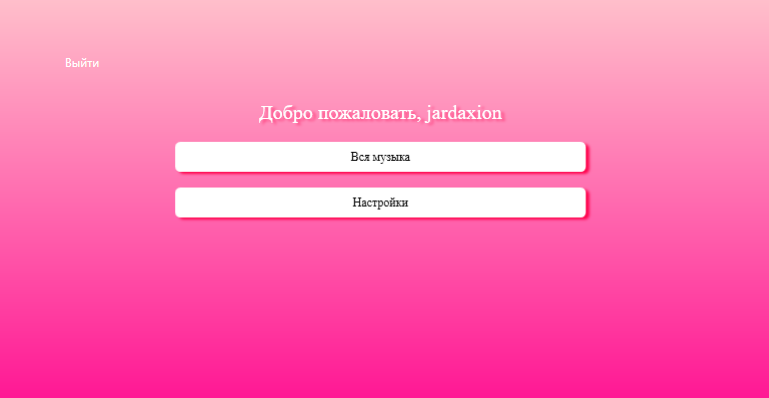


Рисунок 2.16 – Окно профиля

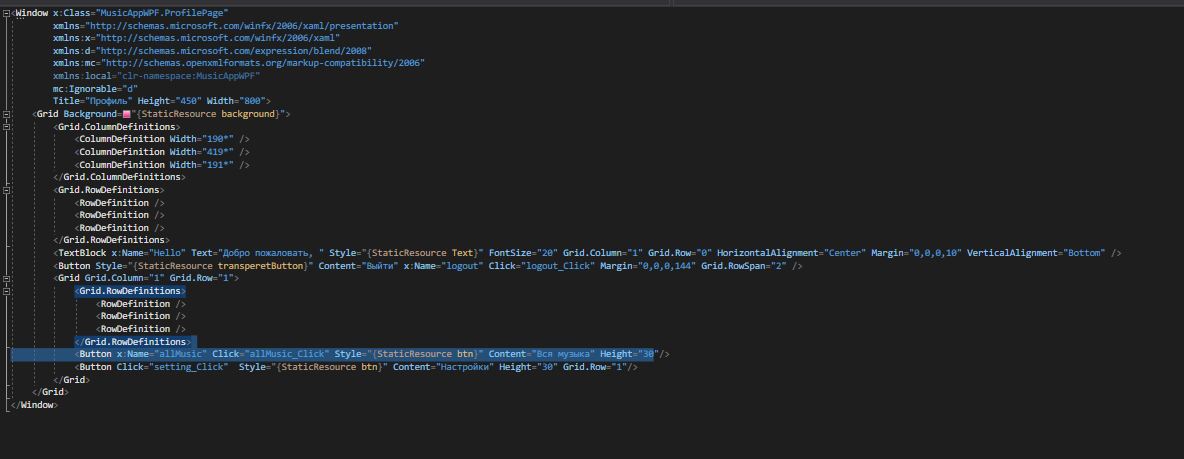


Рисунок 2.17 – XAML разметка окна



Рисунок 2.18 – CS код окна

При нажатии «Выход» пользователь выходит из профиля и перед пользователем вновь открывается окно со входом\регистрацией.  
 При нажатии на кнопку «Вся музыка» перед пользователем открывается окно со всей музыкой, где он может включить или поставить на паузу каталог песен. (Рисунки 2.19, 2.20, 2.21).



Рисунок 2.19 – Окно «Вся музыка»

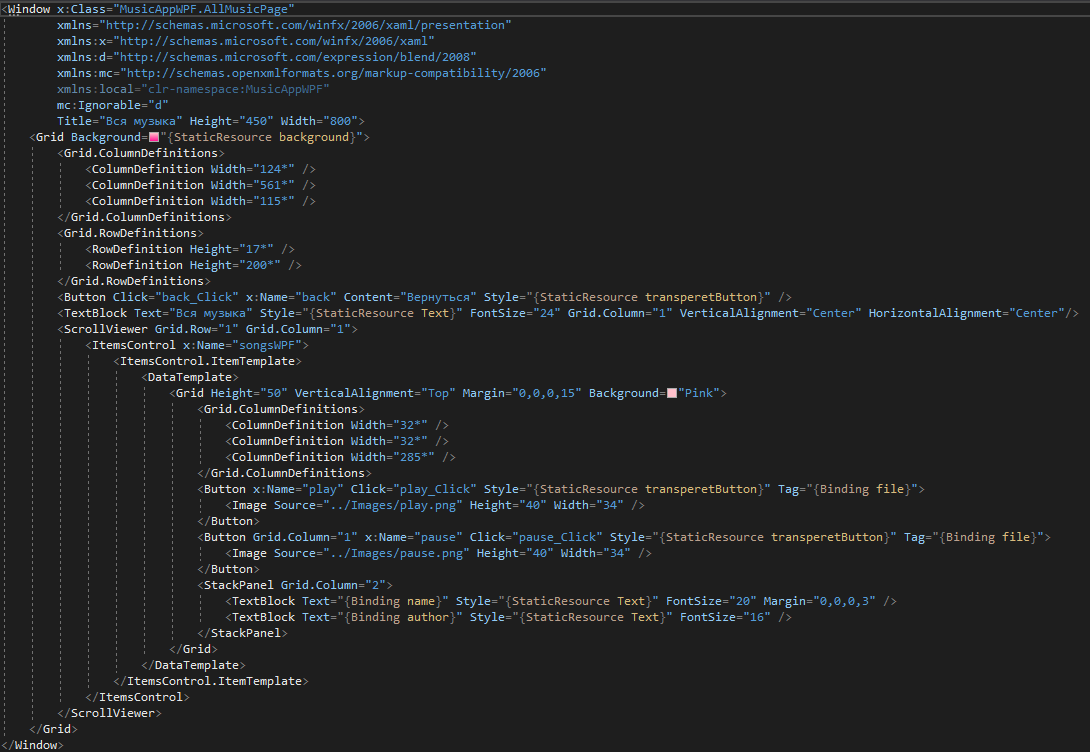


Рисунок 2.20 – XAML разметка



Рисунок 2.21 – CS код окна

При нажатии кнопки пользователь возвращается на окно «Профиль», далее при нажатии на кнопку «Настройки», пользователя перемещает на страницу настройки профиля, где он может редактировать свои данные (Рисунок 2.22, 2.23, 2.24).

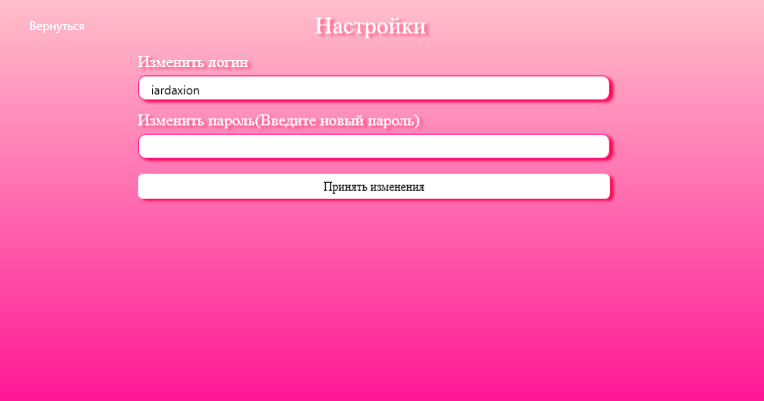


Рисунок 2.22 – Окно «Настройки»

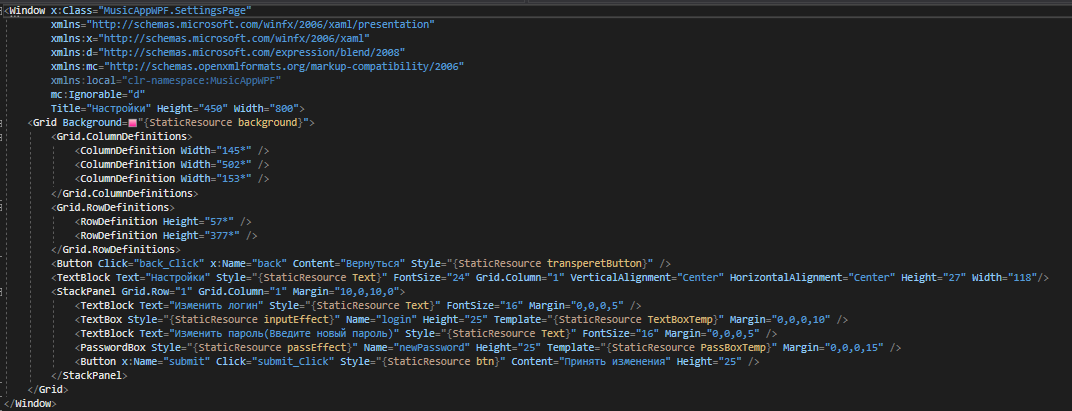


Рисунок 2.23 – XAML разметка окна

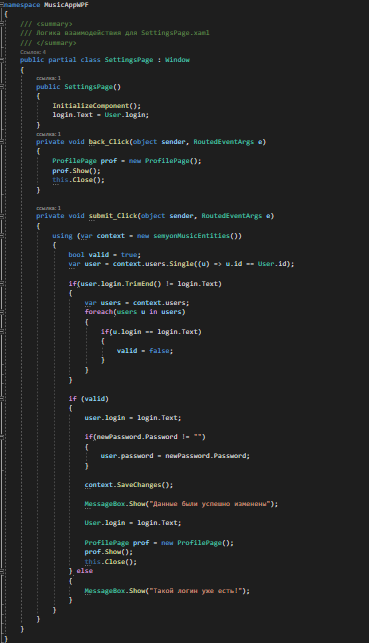


Рисунок 2.24 – CS код окна

При нажатии на кнопку «Принять изменения», пользователь меняет свои данные.

Так же есть окна Администратора. Если данные пользователя совпали с данными администратора из Базы Данных (бд), то он попадает на страницу администратора (Страницы 2.25, 2.26, 2.27)

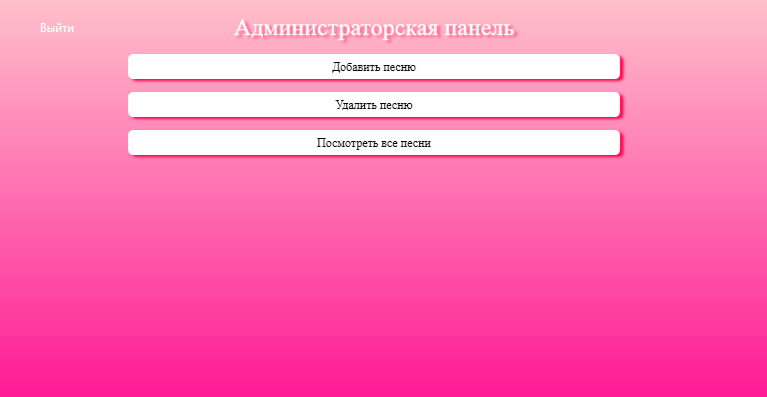


Рисунок 2.25 – Окно Администратора

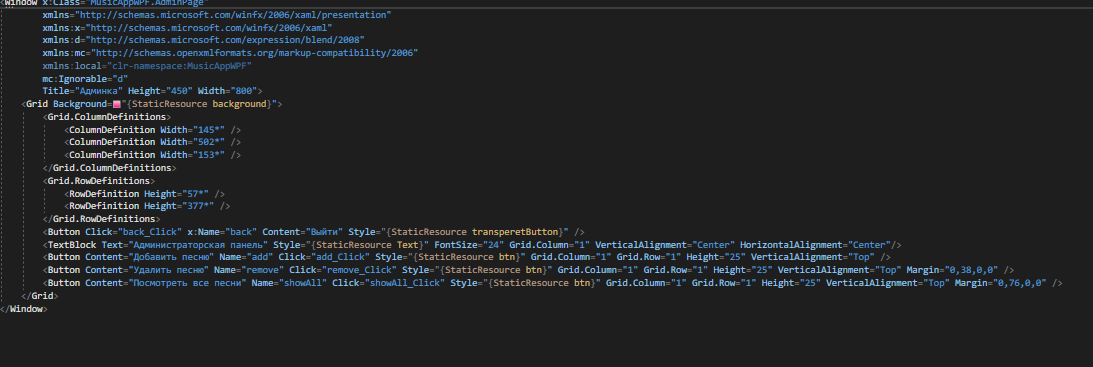


Рисунок 2.26 – XAML разметка окна

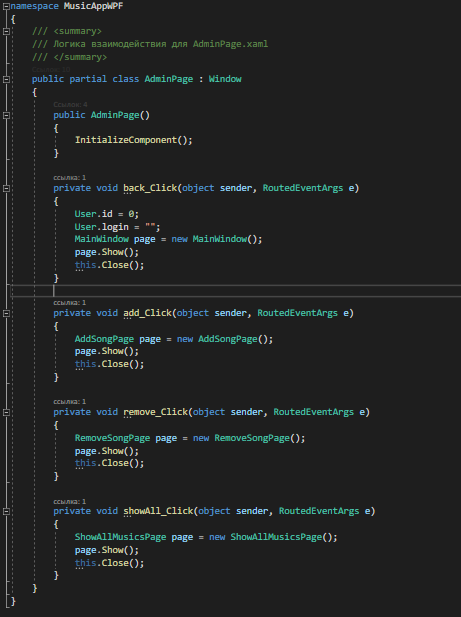


Рисунок 2.27 – CS код окна

При нажатии на кнопку «Добавить песню» пользователя перекидывает на страницу с добавлением песни, где нужно заполнить автора, название, а так же вставить сам файл с песней, после чего нажать на кнопку «Добавить песню», после валидации песня будет добавлена (Рисунок 2.28, 2.29, 2.30).

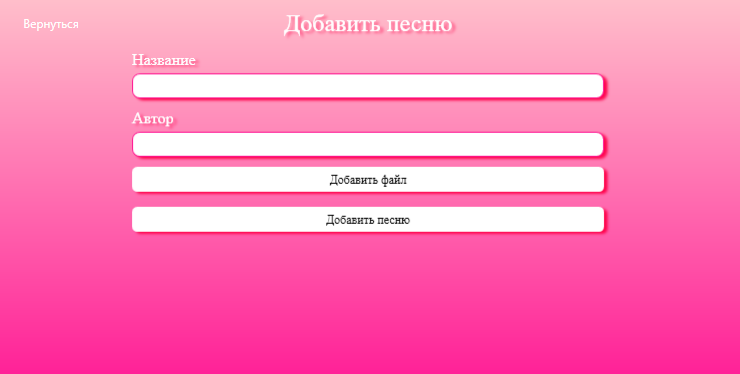


Рисунок 2.28 – Окно «Добавить песню»

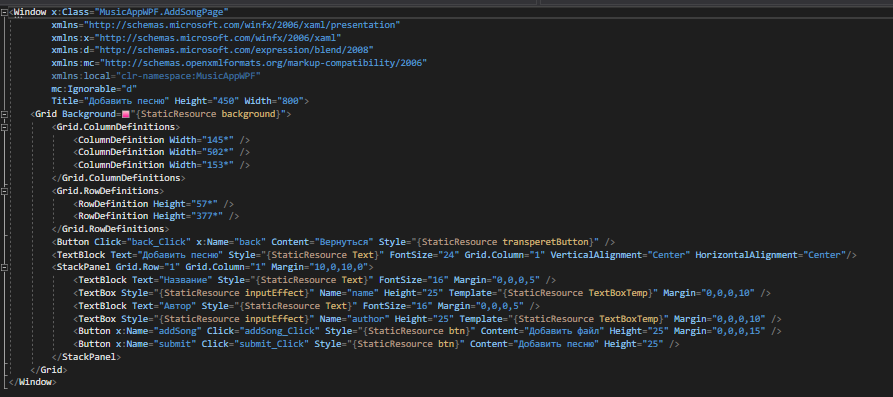


Рисунок 2.29 – XAML разметка окна

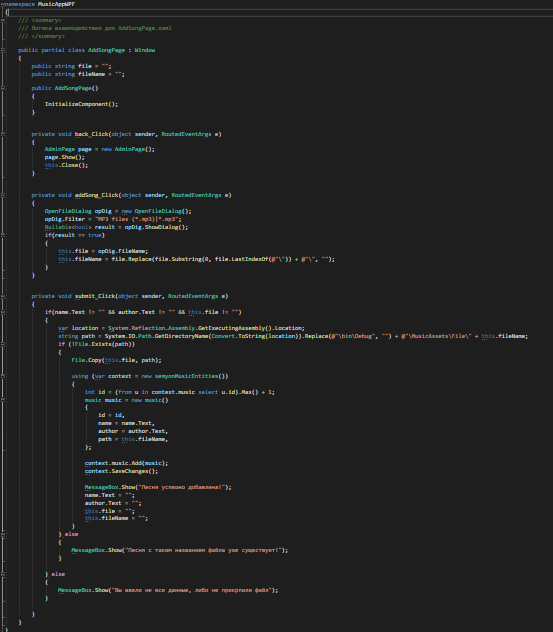


Рисунок 2.30 – CS код окна

При нажатии на кнопку «Удалить страницу» пользователя перекидывает на окно с выбором песни и кнопкой удалении. После выбора песни – она удалится (Рисунок 2.31, 2.32, 2.33).

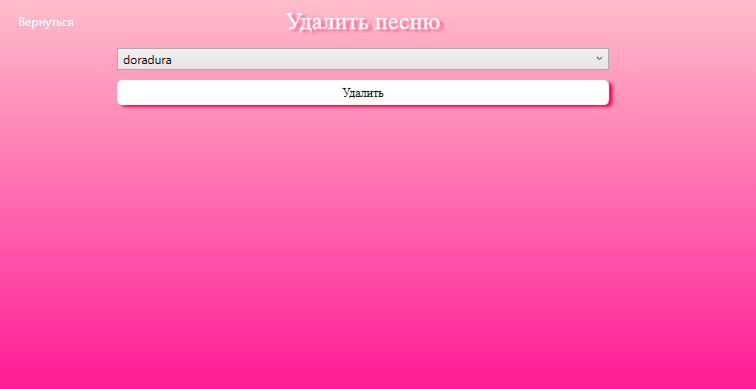


Рисунок 2.31 – Окно «Удалить песню»

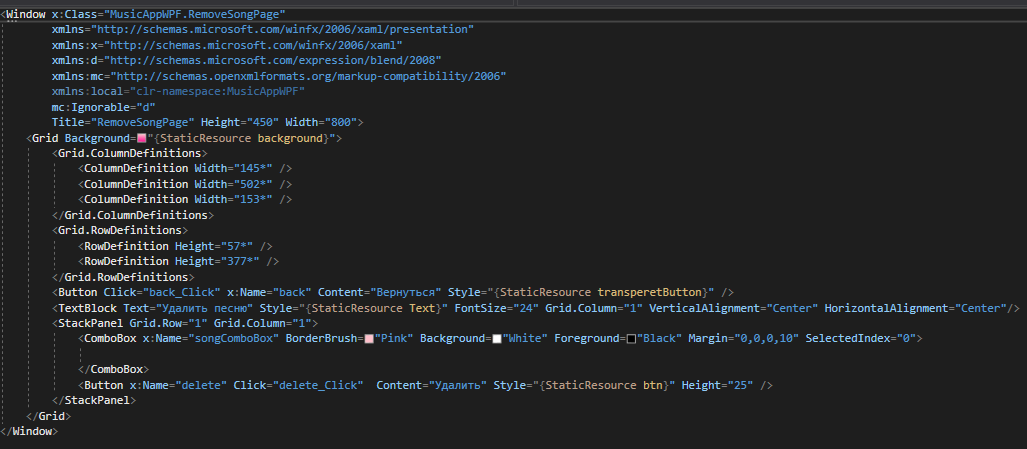


Рисунок 2.32 – XAML разметка окна

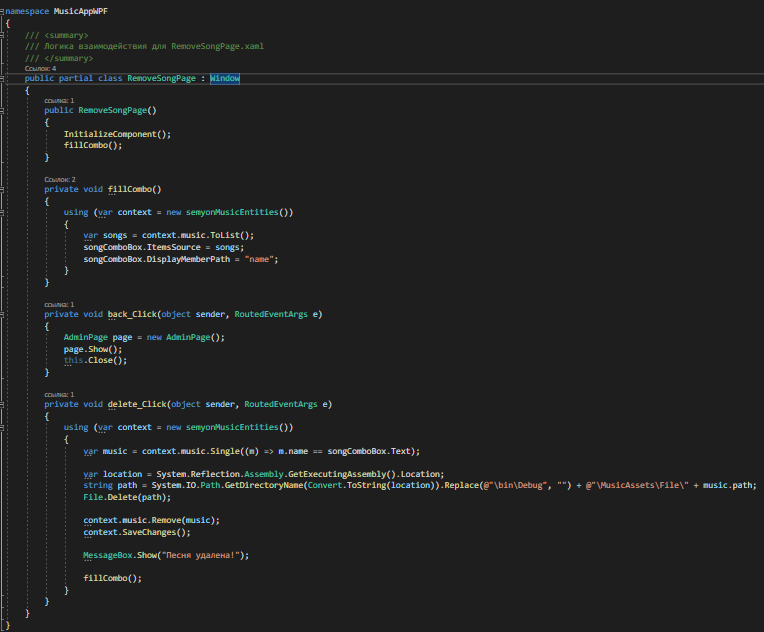


Рисунок 2.33 – CS код окна

При нажатии кнопки «Посмотреть все песни» пользователя перекидывает на окно, где находится таблица со всеми песнями (Рисунок 2.34, 2.35, 2.36).



Рисунок 2.34 – Окна «Вся музыка»

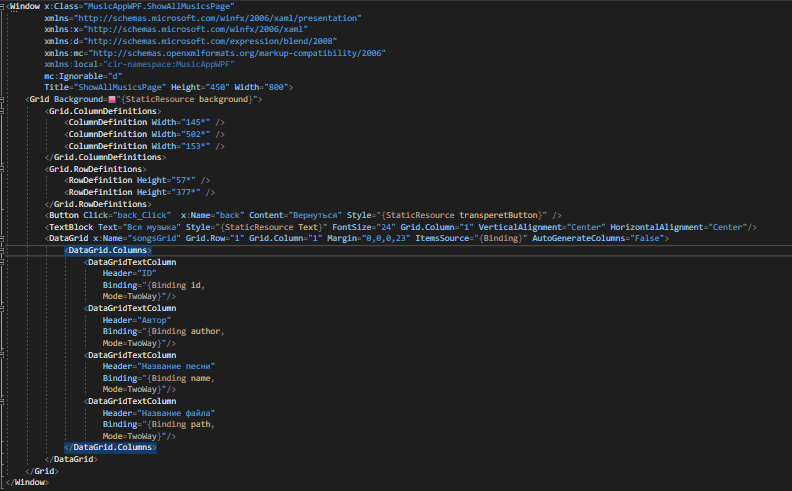


Рисунок 2.35 – XAML разметка окна

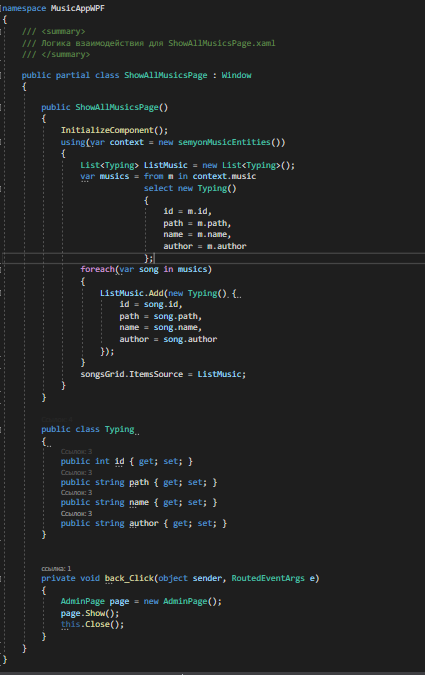


Рисунок 2.36 – CS код окна

# Заключение

В данной курсовой работе было создано настольное приложение «Музыкальная платформа»

Данное приложение должно выполнять добавление песни, воспроизведение песен, показ всех песен, а также удаление песен.

В процессе работы исследована предметная область. Проведен выбор технологии и инструментов реализации программного модуля.

Проект реализован с помощью среды Microsoft Visual Studio, C#, работающая в связке Microsoft SQL Server Management Studio 19. Язык C# способствовал уменьшению объема передаваемой информации по каналам связи, что особенно актуально среди сложившейся конкуренции.

Таким образом, в рамках курсовой работы для достижения цели были решены следующие задачи:

* Было произведено ознакомление с общей характеристикой приложения;
* проведен обзор инструментов для реализации настольного приложения;
* проведен анализ предметной области;
* разработан функциональные требования к программному модулю;
* выполнено проектирование настольного приложения и базы данных;
* реализовано настольное приложение.

# Список литературы

Литературу берем из онлайн библиотеки – znanium.com, e.lanbook.com 15 источников…. Базы данных, С#, WPF, ( 2018 года и позже).