



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК5 «Системы обработки информации»

РАСЧЕТНО - ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе на тему:

Веб-приложение для прослушивания музыкальных композиций.

по дисциплине Архитектура автоматизированных систем обработки
информации и управления

Студент гр. ИУК5-52Б _____ (Панченко А.П.)
(подпись) (Ф.И.О.)

Руководитель _____ (Смирнов М.Е.)
(подпись) (Ф.И.О.)

Оценка руководителя _____ баллов _____
30-50 (дата)

Оценка защиты _____ баллов _____
30-50 (дата)

Оценка проекта _____ баллов _____
(оценка по пятибалльной шкале)

Комиссия: _____ (_____)
(подпись) (Ф.И.О.)

_____ (_____)
(подпись) (Ф.И.О.)

_____ (_____)
(подпись) (Ф.И.О.)

Калуга, 2021

Калужский филиал
федерального государственного бюджетного образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУК5

_____ (Е.В. Вершинин)

« ____ » _____ 20__ г.

ЗАДАНИЕ
на выполнение курсовой работы

по дисциплине Архитектура автоматизированных систем обработки информации и управления

Студент

Панченко А.П. ИУК5-52Б

_____ (фамилия, инициалы, индекс группы)

Руководитель

Смирнов М.Е.

_____ (фамилия, инициалы)

График выполнения проекта: 25% к 4 нед., 50% к 7 нед., 75% к 10 нед., 100% к 14 нед.

1. Тема курсового проекта

Веб-приложение для прослушивания музыкальных композиций

2. Техническое задание

Разработать веб-приложение для прослушивания музыкальных композиций использованием базы данных.

3. Оформление курсового проекта

3.1. Расчетно-пояснительная записка на _____ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.) _____

Дата выдачи задания « ____ » _____ 2021г.

Руководитель курсового проекта _____ / _____ /

(подпись)

(Ф.И.О.)

Задание получил _____ / _____ / « ____ » _____ 2021г.

(подпись)

(Ф.И.О.)

Примечание:

Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ.....	5
1.1. Общие сведения	5
1.1.1. Полное наименование системы.....	5
1.1.2. Плановые сроки начала и окончания работы по созданию системы.....	5
1.2. Назначение и цели создания (развития) системы.....	5
1.2.1. Назначение системы	5
1.2.2. Цели создания системы	5
1.3. Характеристики объекта автоматизации	5
1.4. Требования к системе	6
1.4.1. Требования к системе в целом	6
1.4.2. Требования к структуре и функционированию системы	6
1.4.3. Требования к надежности	6
1.5. Состав и содержание работ по созданию системы	7
1.6. Порядок контроля и приемки системы	8
1.6.1. Состав, объем и методы испытаний системы и ее составных частей.....	8
1.6.2. Общие требования к приемке работ	8
1.7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие	9
1.8. Требования к документированию	9
1.9. Источники разработки	9
2. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ	10
2.1. Постановка задачи проектирования.....	10
2.2. Описание предметной области.....	11
2.3. Анализ аналогов	13
2.4. Перечень задач, подлежащих решению в процессе разработки.	15
2.5. Обоснование выбора инструментов и платформы для разработки.....	16
3. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ	18
3.1. Разработка алгоритмов обработки информации.....	18
3.2. Разработка алгоритмов обработки информации.....	20
3.3. Разработка интерфейса взаимодействия пользователя с системой	21
3.4. Разработка архитектуры приложения	24
4. ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ	25
4.1. Проектирование начального и тестового наполнения базы данных.	25
4.2. Технологические решения, поддерживающие эксплуатационный цикл программы.....	25
4.3. Порядок развертывания системы.	26
4.4. Разработка руководства пользователя и руководства администратора.	26
ЗАКЛЮЧЕНИЕ	33
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	34

ВВЕДЕНИЕ

Главными преимуществами стриминговых музыкальных сервисов считаются небольшая стоимость использования и универсальность. Кроме того, пользователи прислушиваются к музыкальным рекомендациям по стилям и жанрам и слушают плейлисты, созданные при помощи анализа их предпочтений. Размещение музыки на стриминговых сервисах выгодно и музыкантам. Кроме лицензионного вознаграждения(роялти) за прослушивания, они получают новых слушателей.

Стриминговые сервисы работают по принципу передачи контента от провайдера к пользователю. Весь контент уже загружен на стороннем сервере, конечному пользователю не требуется ничего скачивать для просмотра или воспроизведения. Контент транслируется в режиме реального времени, скорость загрузки напрямую зависит от скорости интернета пользователя. С нынешним даже самым простейшим интернетом можно без проблем прослушать музыку. Онлайн контент стал отличной заменой скачиванию файлов.

Все показатели указывают на то, что рост музыкальной индустрии в мире продолжится. И ключевую роль в этом будет играть стриминг – пока самый выгодный легальный способ слушать музыку.

1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1.1. Общие сведения

1.1.1. Полное наименование системы

Веб-приложение для поиска и прослушивания музыкальных композиций.

1.1.2. Плановые сроки начала и окончания работы по созданию системы

Начало работы: 01.09.2021

Окончание работы: 01.12.2021

1.2. Назначение и цели создания (развития) системы

1.2.1. Назначение системы

Назначением системы является автоматизация поиска и прослушивания музыкальных композиций с помощью веб-приложения с использованием базы данных.

1.2.2. Цели создания системы

Цель создания системы – получить прибыль за счет предоставления платных подписок на музыкальные композиции.

1.3. Характеристики объекта автоматизации

Объектом автоматизации является музыкальный плеер. Автоматизированию подлежит демонстрация, поиск и прослушивание имеющихся музыкальных альбомов и композиций.

1.4. Требования к системе

1.4.1. Требования к системе в целом

Программный продукт должен представлять собой веб-приложение.

1.4.2. Требования к структуре и функционированию системы

Приложение должно быть разделено на 3 слоя:

- Слой клиентского приложения в виде веб-страницы;
- Слой бизнес-логики приложения (сервер);
- Слой хранения данных (база данных).

Функционал приложения должен содержать:

- Авторизацию пользователей;
- Поиск музыкальных композиций и альбомов;
- Хранение пользовательских плейлистов и списков альбомов;
- Воспроизведение музыкальных композиций;
- Создание недельных чартов, основанных на количестве прослушиваний музыкальных композиций за текущую неделю;
- Хранение недельных чартов предыдущих недель.

1.4.3. Требования к надежности

Система должна обеспечивать корректную обработку исключительных ситуаций, вызванных, например, вводом неверных пользовательских данных или неоднозначной информации в базу данных.

1.5. Состав и содержание работ по созданию системы

Стадии	Этапы работ	Сроки исполнения
1. Формирование требований к АС	1.1. Исследование объекта и подтверждение необходимости создания АС. 1.2. Формирование требований пользователя к АС. 1.3. Оформление отчёта о выполненной работе и заявки на разработку АС.	
2. Разработка концепции АС.	2.1. Изучение выбранного объекта. 2.2. Проведение научно-исследовательских работ. 2.3. Проектирования концепции АС, удовлетворяющей потребности пользователя. 2.4. Оформление отчёта о выполненной работе.	
3. Техническое задание.	Разработка и утверждение технического задания на создание АС.	

4. Разработка системы.	Разработка системы согласно техническому заданию.	
5. Защита курсовой работы.	5.1. Создание грамотной презентации и речи для защиты курсовой работы. . Защита курсовой работы.	

1.6. Порядок контроля и приемки системы

1.6.1. Состав, объем и методы испытаний системы и ее составных частей

Приложение должно пройти предварительные испытания, состоящие из отладки и минимального набора тестов.

В результате предварительных испытаний, должны быть исправлены недочеты, замечания на которые были получены в ходе предварительных испытаний.

Для проверки корректной работы внесённых изменений должны быть проведены повторные испытания разработанной системы.

1.6.2. Общие требования к приемке работ

В процессе приемки работ должна быть осуществлена проверка на соответствие требованиям настоящего «Технического задания». По результатам испытаний возможны доработки и исправления.

1.7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

При подготовке объекта автоматизации к вводу системы в действие следует выполнить:

- Обеспечение работ по вводу данных в систему;
- Развертывание системы на сервере;
- Настройка системы доступа и создание учетных записей.

1.8. Требования к документированию

По окончании работы предъявлена расчетно-пояснительная записка, в состав которой входят:

- Техническое задание;
- Научно-исследовательская часть;
- Проектно-конструкторская часть;
- Проектно-технологическая часть.

Также должна быть предоставлена графическая часть работы, выполненная в формате А1 на 2 листах, в которую входят:

- Демонстрационные чертежи;
- Алгоритмические схемы.

1.9. Источники разработки

- Гост 34.601-90;
- Гост 34.602-89.

2. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

2.1. Постановка задачи проектирования

Задачей данного проекта является разработка и проектирование веб-приложения, отвечающего следующим требованиям:

- Возможность отслеживания новых альбомов и музыкальных композиций;
- Возможность поиска интересных композиций по авторам, названиям альбомов и композиций;
- Возможность отслеживания недельных чартов текущей недели и предыдущих недель;
- Прослушивание музыкальных композиций.

База данных будет содержать информацию о пользователях, альбомах, исполнителях, песнях и недельных чартах.

Макет главной страницы представлен на Рисунке 2.1.

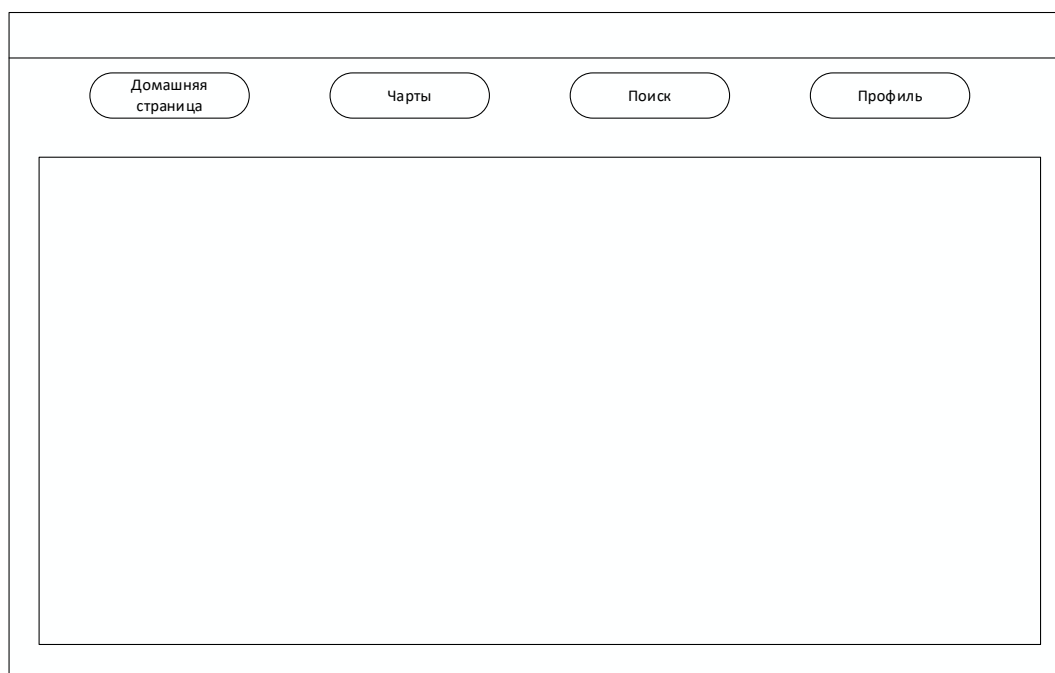


Рисунок 2.1 – Макет главной страницы

2.2. Описание предметной области

В данном приложении существуют песни, альбомы, исполнители и недельные чарты.

- У каждого альбома есть обложка, название, список песен, список исполнителей, список пользователей, которые добавили альбом в свою фонотеку и дата релиза;
- Каждая песня содержит обложку, название песни, свой альбом, список исполнителей, файл песни, список пользователей, которые добавили песню в свою фонотеку, дату релиза, количество прослушиваний(всего) и количество прослушиваний в рамках недельного чарта;
- Каждый исполнитель содержит фотографию, псевдоним и списки выпущенных альбомов и песен;
- Недельные чарты представляют собой дату проведения чарта и список песен, вошедших в каждый из них с позиционированием.

Благодаря внедрению панели администратора в систему облегчается взаимодействие с данными.

Описание и функционал со стороны администратора:

Администратор имеет доступ ко всем данным системы, а также может с легкостью добавлять, изменять, удалять их.

Описание и функционал со стороны пользователя:

- Отслеживание новинок недели, месяца, года;
- Просмотр чарта как текущей недели, так и предшествующих недель;
- Добавление альбомов и песен в личную фонотеку, для удобства (только для авторизованных пользователей);
- Поиск по ключевым словам(названиям альбомов, песен, именам исполнителей);
- Авторизация / регистрация в системе;
- Прослушивание песен.

Для данного проекта была сформирована UML диаграмма прецедентов для администратора (Рисунок 2.2.1) и для пользователя (Рисунок 2.2.2).

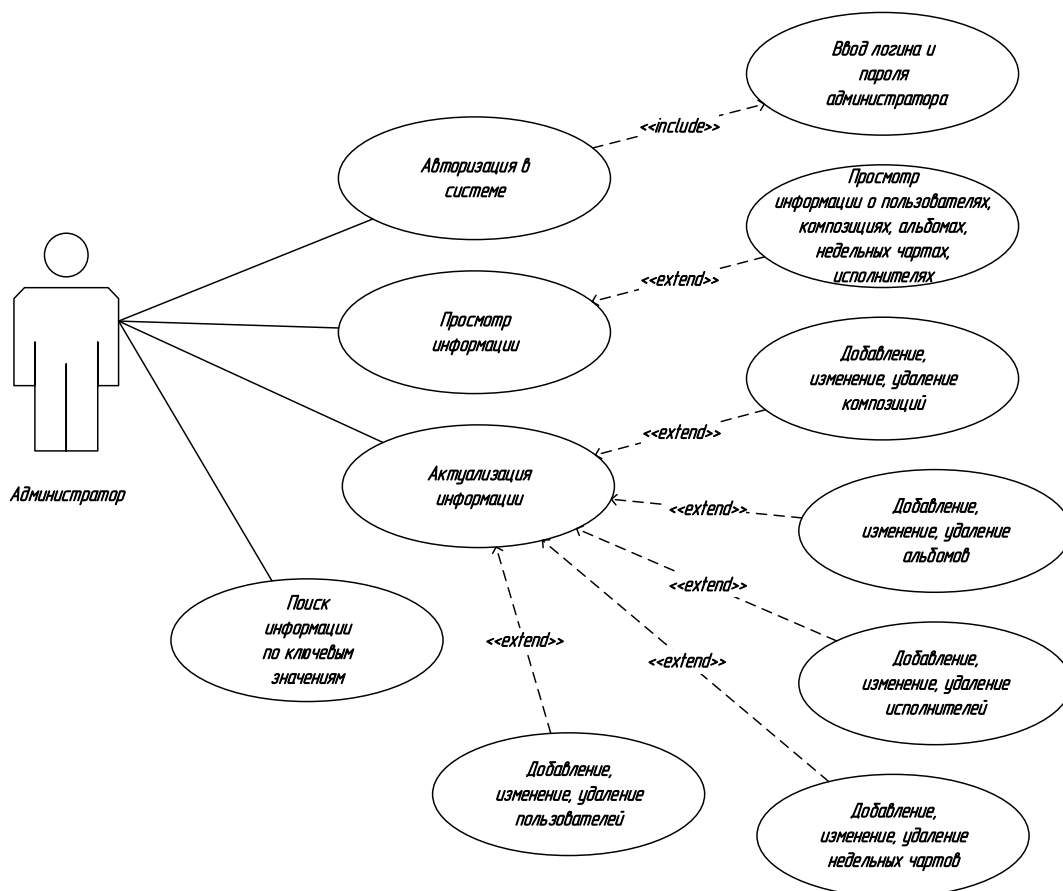


Рисунок 2.2.1 – UML диаграмма прецедентов для администратора

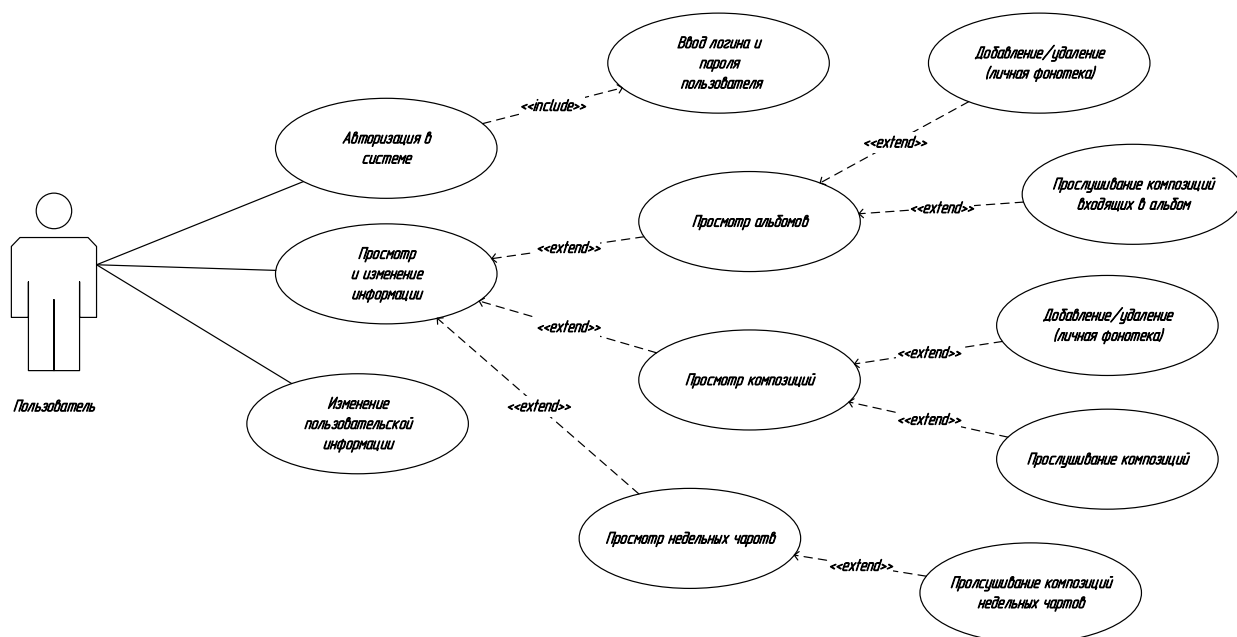


Рисунок 2.2.2 – UML диаграмма прецедентов для пользователя

2.3. Анализ аналогов

SoundCloud

Главная задача – это прослушивание музыки, плеер интегрирован в приложение. Здесь есть возможность потокового прослушивания аудиозаписей и создания плейлистов неограниченного объёма. [3]

Главная страница SoundCloud представлена на Рисунке 2.3.1.

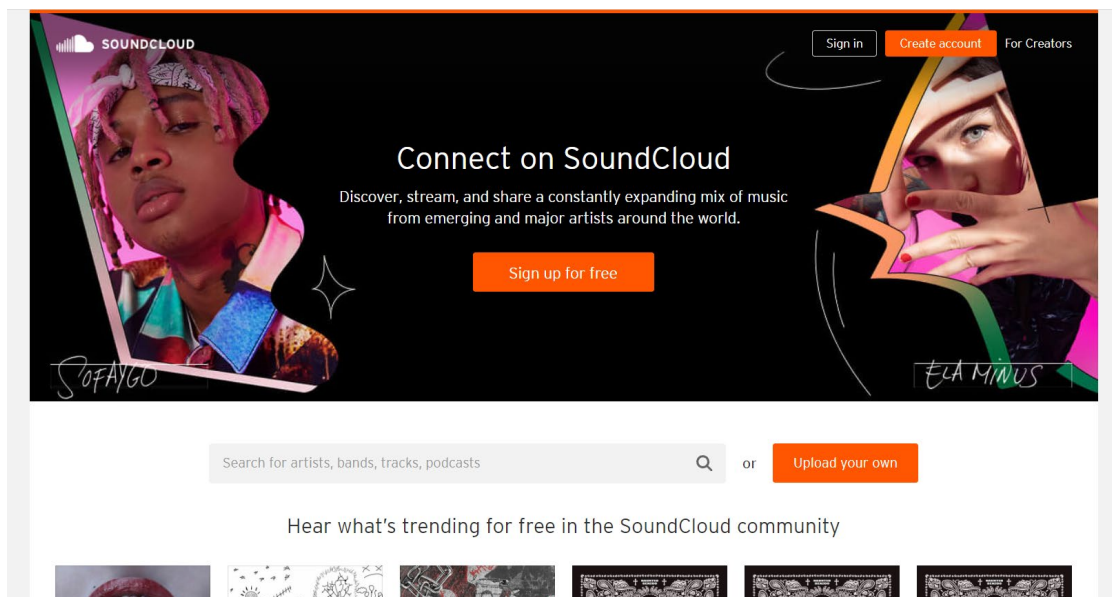


Рисунок 2.3.1 - Главная страница SoundCloud [1]

Функционал:

- Проигрывание станций – каждый пользователь сервиса может публиковать свои аудиоподборки, которые будут доступны широкой публики. Чтобы послушать композиции, достаточно кликнуть по станции.
- Обзор – на данной странице пользователю предлагаются чарты от SoundCloud, а также фильтр композиций по жанрам.
- Создание плейлистов – музыкальная социальная сеть позволяет формировать свои собственные плейлисты.
- Воспроизведение похожих дорожек.

Плюсы:

- Уникальность композиций.
- Возможность прослушивания + составления собственного альбома
- Отображение статистики прослушиваний, просмотров.
- Возможность комментировать альбомы и песни.

Минусы:

- Ограничение по аккаунту.
- Ограничение количества скачиваний.

Spotify

Spotify – это сервис для стриминга различного контента (миллионов треков, подкастов и видеороликов) от авторов со всего мира.

Основной функционал, такой как воспроизведение музыки – бесплатный. Также можно оформить Premium-подписку. [\[5\]](#)

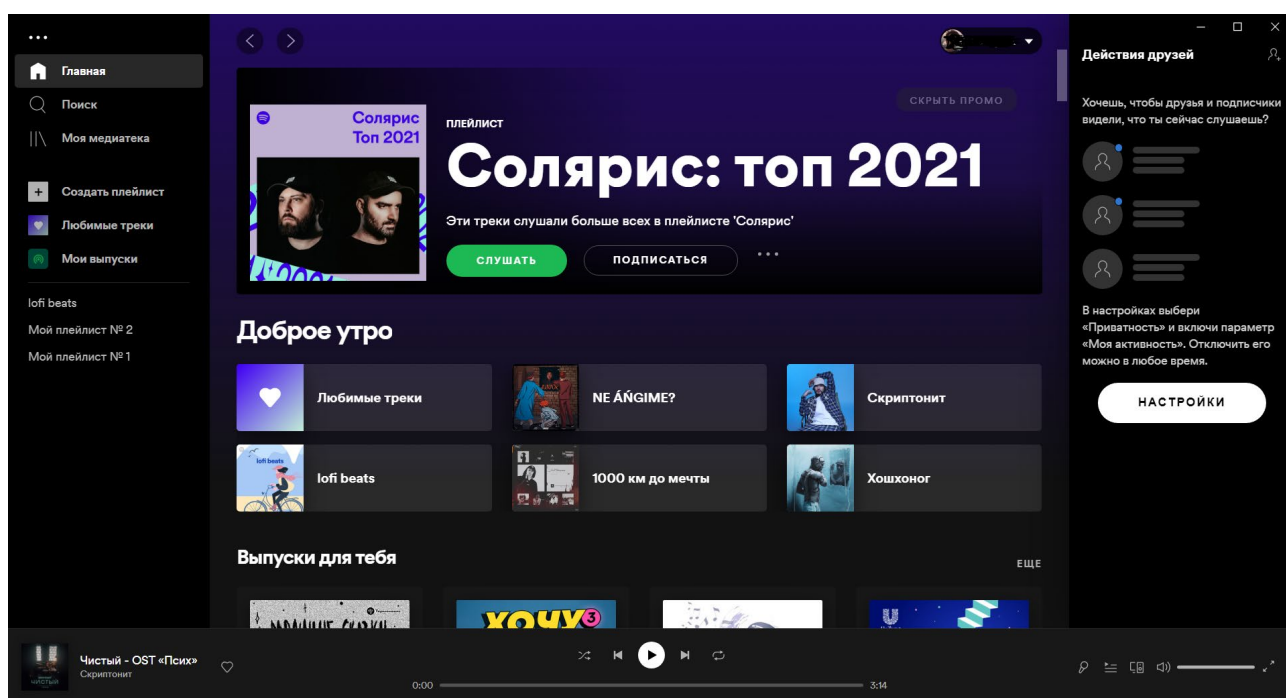


Рисунок 2.3.2 – Главная страница Spotify для ПК [\[2\]](#)

Функционал:

- Получение рекомендаций на основе своих предпочтений.
- Создание своих коллекций музыки и подкастов.

- Главное преимущество – алгоритмы подбора музыки, которые практически идеально подстраиваются под вкусы слушателей.

Сервис Spotify доступен на разных устройствах, включая компьютеры, телефоны и планшеты, а также динамики, телевизоры и автомобильные аудиосистемы.

Преимущества:

- Добавление подписчиков.
- Создание плейлистов.
- Наличие официального приложения для смартфона и компьютера. И основные плюсы Spotify связаны с тем, что программу можно без проблем установить на любое устройство.
- Рекомендации.
- Высокое качество звучания.

Недостатки:

- Большое количество ограничений в бесплатной версии.

2.4. Перечень задач, подлежащих решению в процессе разработки.

- Определить состав и структуру данных;
- Разработать пользовательский интерфейс;
- Определить архитектуру приложения;
- Разработать и реализовать базу данных;
- Разработать функционал WEB-приложения;
- Протестировать и проверить разработанную систему;
- Исправить выявленные ошибки.

2.5. Обоснование выбора инструментов и платформы для разработки

1) В качестве среды разработки была выбрана Microsoft Visual Studio 2019

Microsoft Visual Studio 2019— это набор инструментов разработки, основанных на использовании компонентов, и других технологий для создания мощных, производительных приложений. Кроме того, среда Visual Studio оптимизирована для совместного проектирования, разработки и развертывания корпоративных решений. Visual Studio предоставляет средства для проектирования, разработки и отладки.

Среда разработки Visual Studio представляет собой полный набор средств разработки для создания настольных приложений. Visual C# использует единую интегрированную среду разработки (IDE), которая позволяет совместно использовать средства и упрощает создание решений на базе нескольких языков.

Почему именно эта версия продукта, потому что в нем есть такие удобные нововведения как:

1. Visual Studio IntelliCode повышает эффективность разработки программного обеспечения с помощью искусственного интеллекта (ИИ). Для создания рекомендаций IntelliCode анализирует 2000 проектов с открытым кодом на GitHub

2. Рефакторинг В C#. Есть много новых удобных возможностей рефакторинга, которые помогают упорядочить код. Они отображаются как предложения со значком лампочки и включают такие действия, как перемещение элементов в интерфейс или базовый класс, настройку пространств имен в соответствии со структурой папок.

2) Основной фреймворк разработки – ASP.NET Core с использованием Entity Framework:

ASP.NET Core - это кросс-платформенный фреймворк с открытым исходным кодом, разработанный совместно компанией Microsoft и ее сообществом. Это тотальное превращение ASP.NET, которое объединяет структуру MVC и Web API в единый фреймворк.[\[6\]](#)

Основные преимущества ASP.NET Core:

- Архитектура MVC. ASP.NET Core упрощает разработку, компиляцию и тестирование разработчиками в модели, представлении или контроллере.
- Функциональность Razor Pages. Razor Pages - это новый элемент ASP.NET Core, который делает программные сценарии, основанные на веб-страницах, более производительными. С технической точки зрения, Razor Pages - это модель кодирования, основанная на веб-страницах, которая упрощает создание веб-интерфейса. С применением Razor Pages каждая веб-страница становится автономной с компонентом View, код также четко налажен.
- Улучшение совместной работы и кросс-платформенной поддержки. ASP.NET Core - это кросс-платформенный фреймворк, что означает, что приложения, построенные с использованием этого фреймворка, могут работать в операционных системах Windows, Linux и Mac. Кроме того, разработчики также могут свободно выбирать свои ОС разработки.

3. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ

3.1. Разработка алгоритмов обработки информации

Данное приложение имеет трехуровневую клиент-серверную архитектуру с использованием MVC (Model-View-Controller) (Рисунок 3.1).

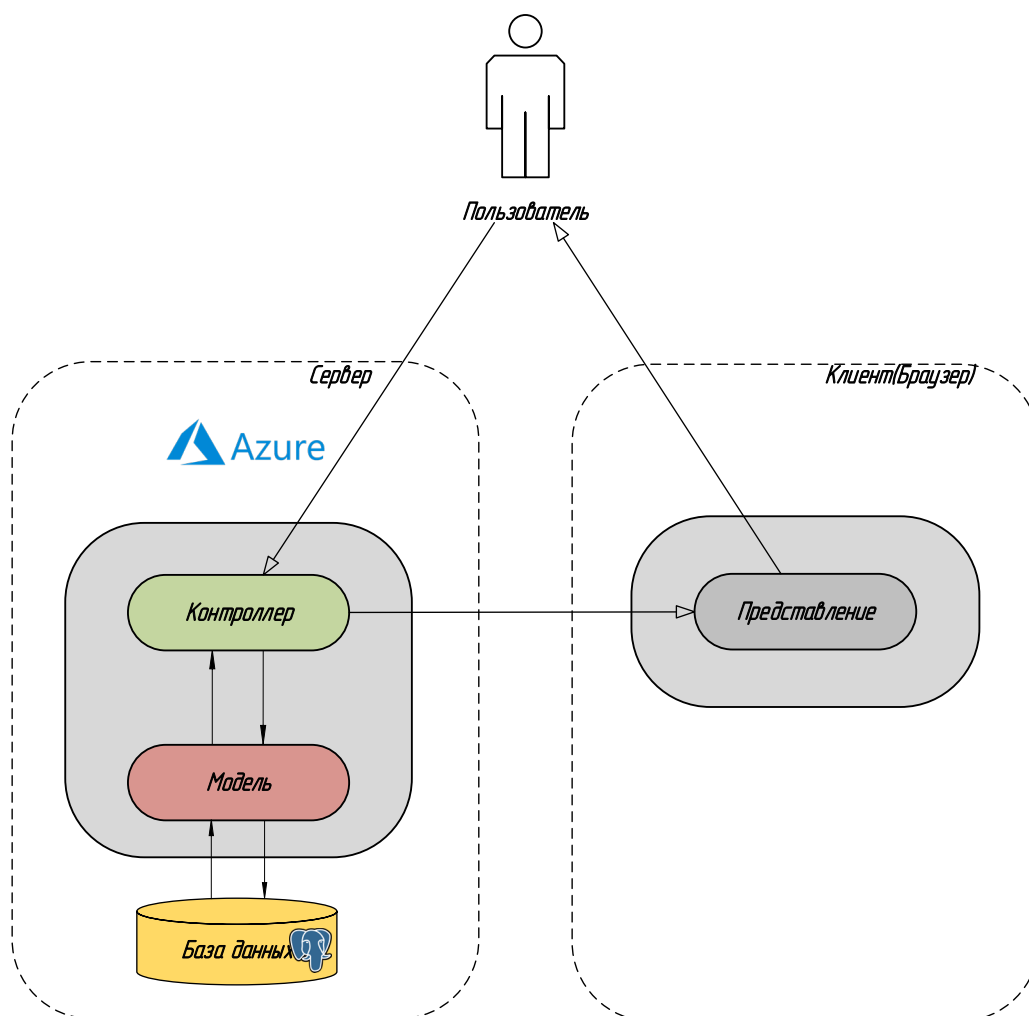


Рисунок 3.1 – MVC архитектура

База данных хранит и исходные данные заполненные администратором и данные пользователей, а также композиции, альбомы, исполнителей и недельные чарты. Всё взаимодействие с базой данных происходит через контроллер, который в свою очередь обращается к модели, а уже модель взаимодействует с базой данных. После получения данных, контроллер передает данную модель представлению на клиентской стороне.

Бизнес логика приложения находится в методах моделей (сущностей) и осуществляется в контроллерах. В представлениях на стороне пользователя нет бизнес логики.

Основное преимущество архитектуры MVC – модульность. Из-за этого свойства очень просто создавать множество представлений для одинаковых моделей. Так же с данной архитектурой намного проще отлаживать код (если, например, возникла ошибка в обработке данных, то сразу отбрасывается модель и представление, и ошибка ищется в контроллере).

Основные компоненты серверной части приложения (MVC):

- Модель – объектная модель предметной области (например: модель Заказа), включает в себя данные и методы работы с этими данными, реагирует на запросы из контроллера, возвращая данные и/или изменяя своё состояние. При этом модель не содержит в себе информации о способах визуализации данных или форматах их представления, а также не взаимодействует с пользователем напрямую.
- Представление – отвечает за отображение информации (визуализацию). Одни и те же данные могут представляться различными способами и в различных форматах. Например, коллекцию объектов при помощи разных представлений можно представить на уровне пользовательского интерфейса как в табличном виде, так и списком (как, например, представление товаров категории для администратора и для пользователя).
- Контроллер – обеспечивает связь между пользователем и системой, использует модель и представление для реализации необходимой реакции на действия пользователя. Как правило, на уровне контроллера осуществляется фильтрация полученных данных и авторизация — проверяются права пользователя на выполнение действий или получение информации.

Приложением-клиентом, в следствии такой архитектуры, является веб-браузер, что дает множество плюсов:

- Кроссплатформенность. Для взаимодействия с приложением не требуется дополнительное ПО, нужен только веб-браузер, который существует на всех операционных системах (в том числе и мобильных).
- Простота. Пользователям намного проще разобраться с веб-приложением, чем, например, с десктопным приложением.
- Удобная поддержка приложения. Так как все приложение находится на сервере, то пользователям не нужно вручную устанавливать или ждать обновления.

3.2. Разработка алгоритмов обработки информации

Основные нетривиальные задачей разработки системы является создание алгоритмов обработки информации.

В системе были реализованы интерфейсы репозиторий и их реализации для каждой из сущностей. Данные интерфейсы включают в себя все возможные операции по манипуляции данными.

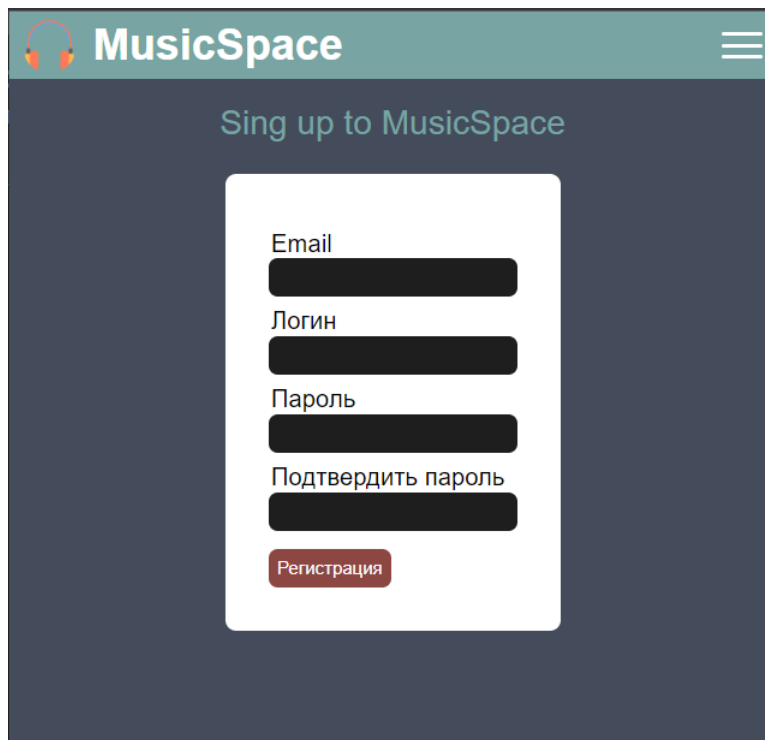
Репозитории предоставляют минимально необходимые данные о сущностях, но если есть необходимость получения дополнительной информации о сущности, как например в список песен, для каждой песни получить списки пользовательских фонотек, то в дело вступает Менеджер данных (DataManager) позволяющий расширить сущность и получить вложенные данные.

Также DataManager служит для предотвращения избыточности данных посредством контроля создаваемых сущностей.

3.3. Разработка интерфейса взаимодействия пользователя с системой

Были разработаны макеты веб-страниц приложения:

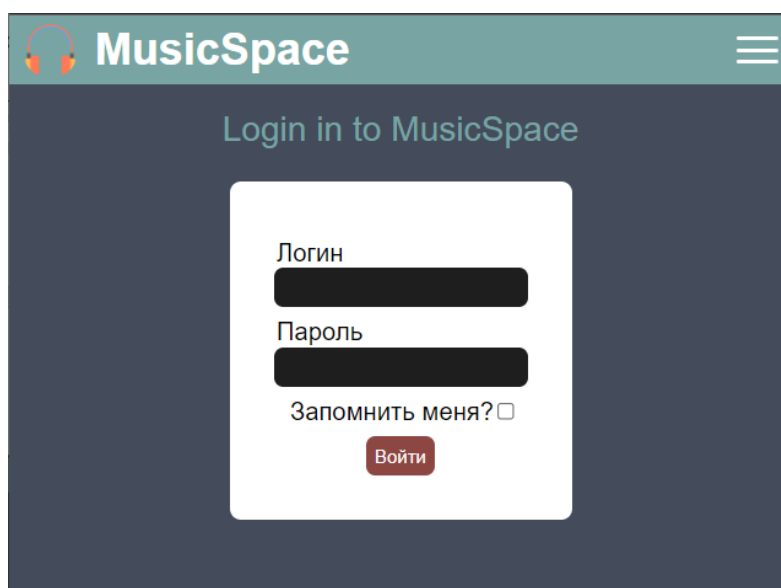
- Макет страницы регистрации (Рисунок 3.3.1).



Mockup of the MusicSpace registration page. The header features the MusicSpace logo with a headphones icon and a hamburger menu. The main heading is "Sing up to MusicSpace". The registration form includes fields for Email, Login, Password, and Confirm Password, followed by a "Регистрация" button.

Рисунок 3.3.1 – Макет страницы регистрации

- Макет страницы авторизации (Рисунок 3.3.2).



Mockup of the MusicSpace login page. The header features the MusicSpace logo with a headphones icon and a hamburger menu. The main heading is "Login in to MusicSpace". The login form includes fields for Login and Password, a "Запомнить меня?" checkbox, and a "Войти" button.

Рисунок 3.3.2 – Макет страницы авторизации

- Макет главной страницы (Рисунок 3.3.3).

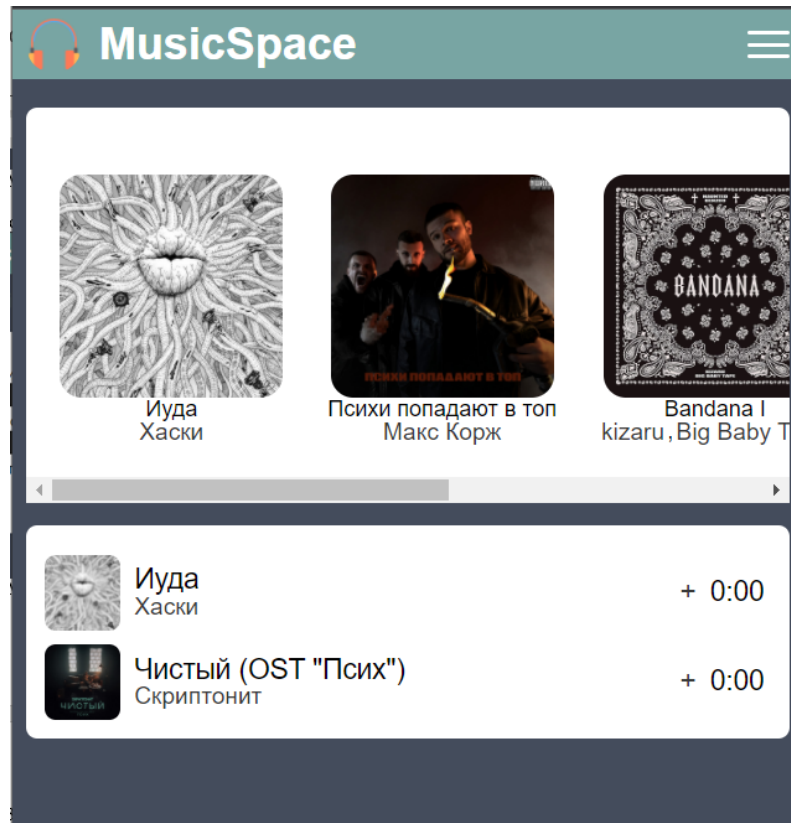


Рисунок 3.3.3 – Макет главной страницы

- Макет страница недельных чартов (Рисунок 3.3.4).

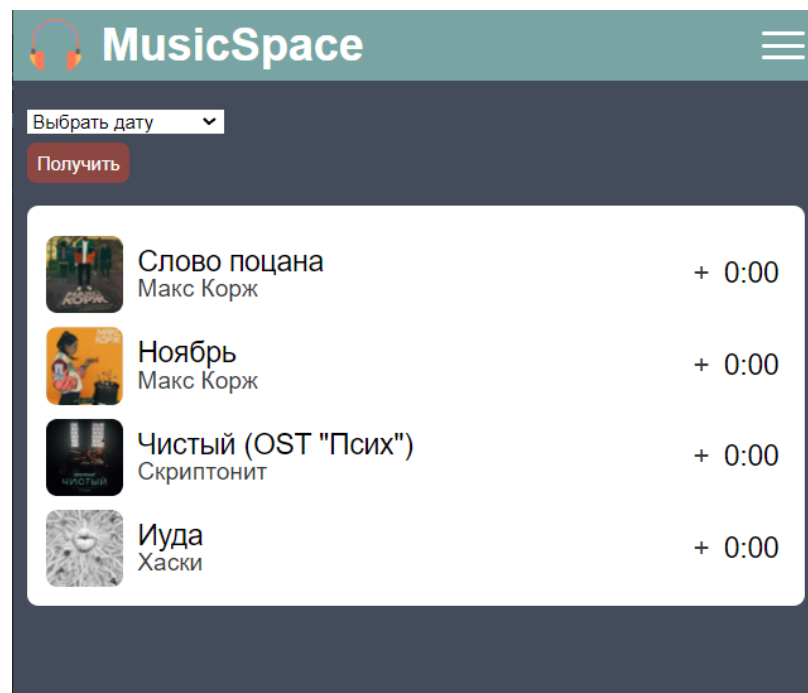


Рисунок 3.3.4 – Макет страницы недельных чартов

- Макет бургер-меню страницы (Рисунок 3.3.5).

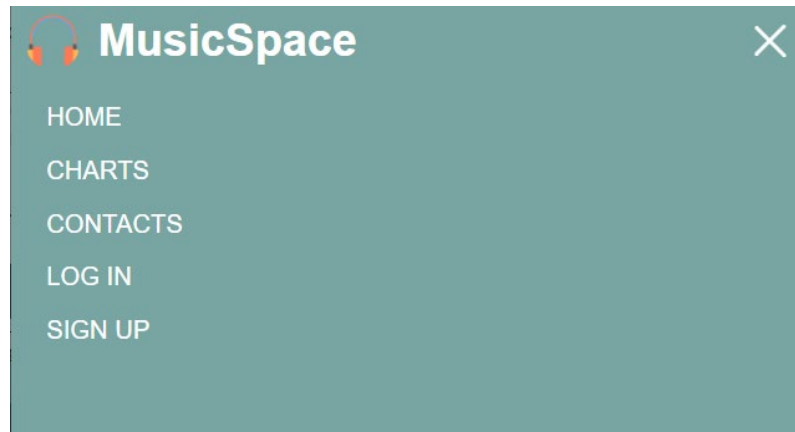


Рисунок 3.3.5 – Макет бургер-меню страницы

- Макет профиля пользователя (Рисунок 3.3.6).

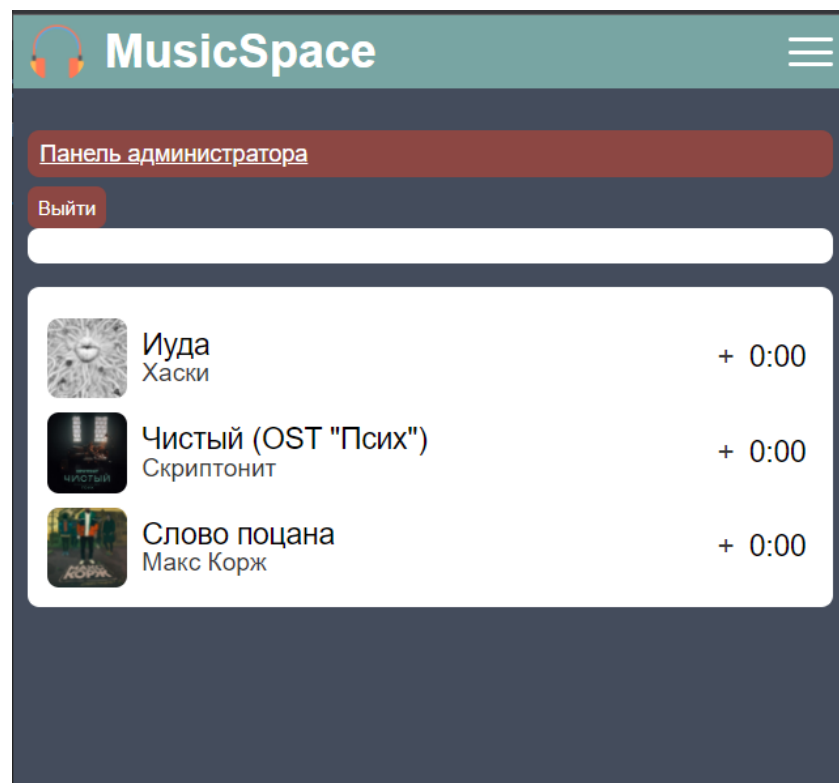


Рисунок 3.3.6 – Макет профиля пользователя

3.4. Разработка архитектуры приложения

Было решено создавать именно веб-приложение, так как оно более универсально и практично для конечного пользователя. Веб-приложением можно пользоваться из любого браузера на любой операционной системы.

Для данного проекта подошла бы любая платформа разработки веб-приложения, была выбрана, как уже указывалось в исследовательской части, ASP.NET Core.

Передача информации между компонентами внутри приложения осуществляется, как уже говорилось, с использованием MVC (Model-view-controller). Контроллеры используют REST архитектуру. В некоторых случаях (например при фильтрации) используется модель поведения PRG (Post-Redirect-Get), которая защищает пользователя от повторной отправки данных веб-форм, и значительно улучшает восприятие приложения пользователем (Рисунок 3.4.1).

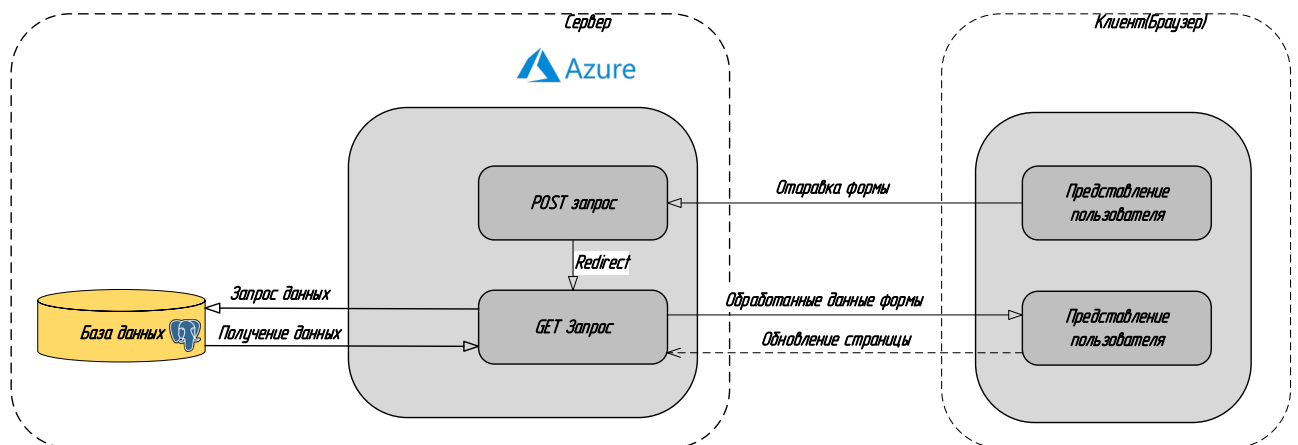


Рисунок 3.4.1 - Модель поведения PRG

Так же стоит отметить использование модели асинхронного программирования (async await), что позволяет серверу делать асинхронные запросы к базе данных, ускоряя отклик сервера.

4. ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

4.1. Проектирование начального и тестового наполнения базы данных.

Процедура автоматизации.

При разработке системы встал вопрос об удобном заполнении системы данными. Для тестового наполнения базы данных был использован осуществленный механизм добавления через само веб-приложение (вкладка “Администрирование”). Причиной выбора такого решения стало удобство выбора списка объектов для заполнения других объектов, например заполнение списка исполнителей для альбомов и композиций.

Данная автоматизация позволяет без особых навыков и усилий взаимодействовать с базой данных, без прямого взаимодействия с СУБД.

4.2. Технологические решения, поддерживающие эксплуатационный цикл программы.

Данное приложение можно улучшить в будущем многими способами. Так как разработка приложения была ограничена небольшим количеством времени, то большинство алгоритмов были реализованы не идеально. Большинство из них можно оптимизировать. Алгоритмы взаимодействия с базой данных можно усовершенствовать, добившись наилучшей оптимизации.

В данном приложении многие страницы не в полной мере адаптивны, и неверно отображаются на некоторых моделях мобильных устройств.

Также в данное приложение можно добавить систему комментирования альбомов и композиций, ленивую загрузку страницы (Lazy load), которая позволит пользователю быстрее получить требуемую информацию.

4.3. Порядок развертывания системы.

Данное приложение было развернуто в облачной платформе Azure компании Microsoft. Данное действие является весьма простым и быстрым в реализации, из-за использования Visual Studio(Рисунок 4.3.1).

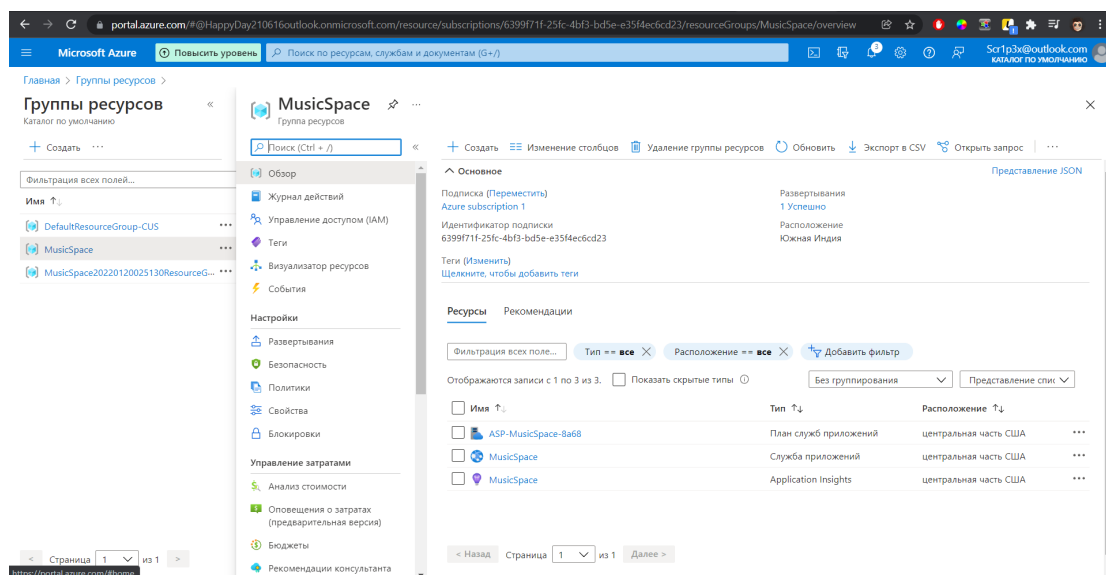


Рисунок 4.3.1 – Профиль опубликованного приложения

Для корректной публикации в Azure была создана группа ресурсов для приложения и базы данных. База данных была так же настроена на связь с Entity Framework для корректного применения миграций.

4.4. Разработка руководства пользователя и руководства администратора.

Руководство администратора:

Для возможности администрирования веб приложения требуется выполнить авторизацию с логином и паролем пользователя, имеющего доступ к администрированию сайта (Рисунок 4.4.1). Если нужно установить данную роль пользователя, то требуется сделать это в СУБД. В дальнейшем же можно добавлять роли в самом приложении имея доступ к администрированию.

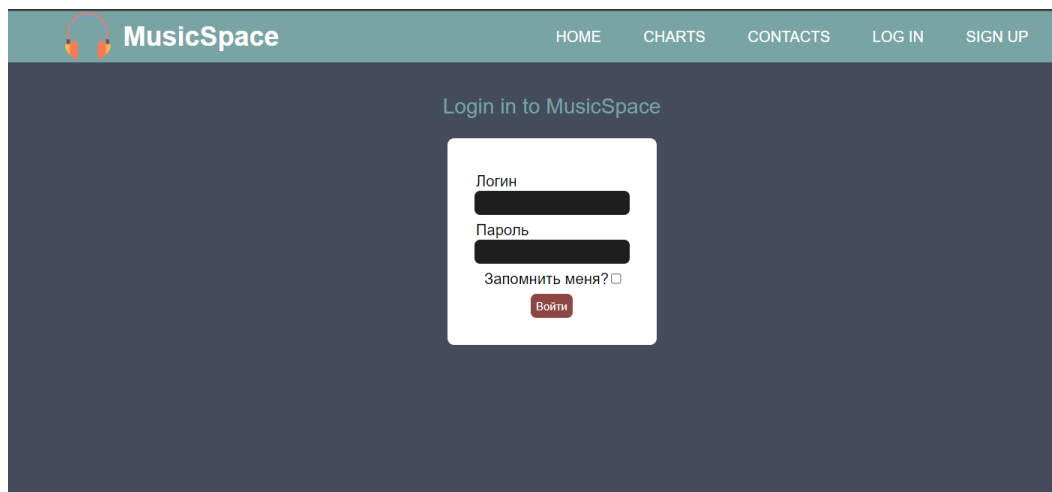


Рисунок 4.4.1 – Форма авторизации

1) После авторизации в профиле администратора появится кнопка “Панель администратора” (Рисунок 4.4.2 и Рисунок 4.4.3). При нажатии на кнопку появится несколько опций выбора. Подробнее о них далее.

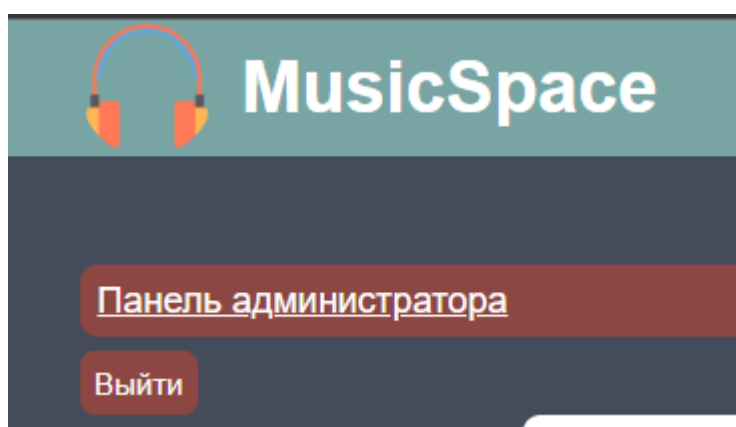


Рисунок 4.4.2 – Кнопка панели администратора

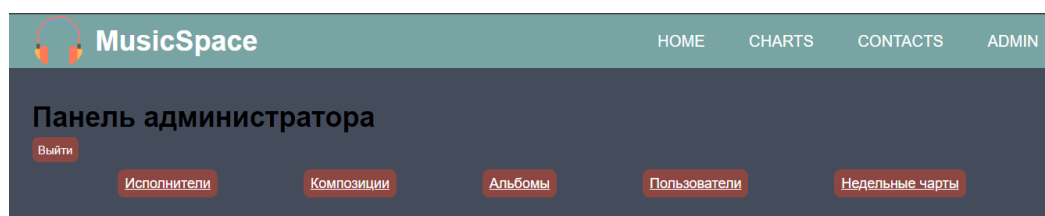


Рисунок 4.4.3 – Панель администратора

2) Кнопка Исполнители (Рисунок 4.4.3) – предоставляет таблицу всех имеющихся в базе данных исполнителей, а также все возможные действия по изменению данных исполнителей (Рисунок 4.4.4).




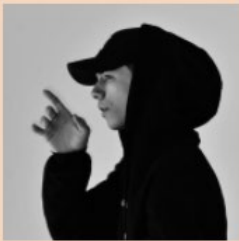
Аватар	Имя исполнителя	
	Big Baby Tape	Изменить Подробнее Удалить
	FACE	Изменить Подробнее Удалить
	kizaru	Изменить Подробнее Удалить
	Macan	Изменить Подробнее Удалить

Рисунок 4.4.4 – Таблица исполнителей

3) Кнопка композиции (Рисунок 4.4.3) – предоставляет таблицу всех имеющихся в базе данных песен, а также все возможные действия по изменению данных композиций (Рисунок 4.4.5).

Обложка	Название песни	Название альбома	Исполнители песни	Количество прослушиваний	
	Иуда	Иуда	Хаски	0	Изменить Подробнее Удалить
	Слово поцана	Домашний	Макс Корж	0	Изменить Подробнее Удалить
	Ноябрь	Домашний	Макс Корж	0	Изменить Подробнее Удалить
					Изменить Подробнее Удалить

0:00 | Иуда Хаски | 0:00

Рисунок 4.4.5 – Таблица композиций

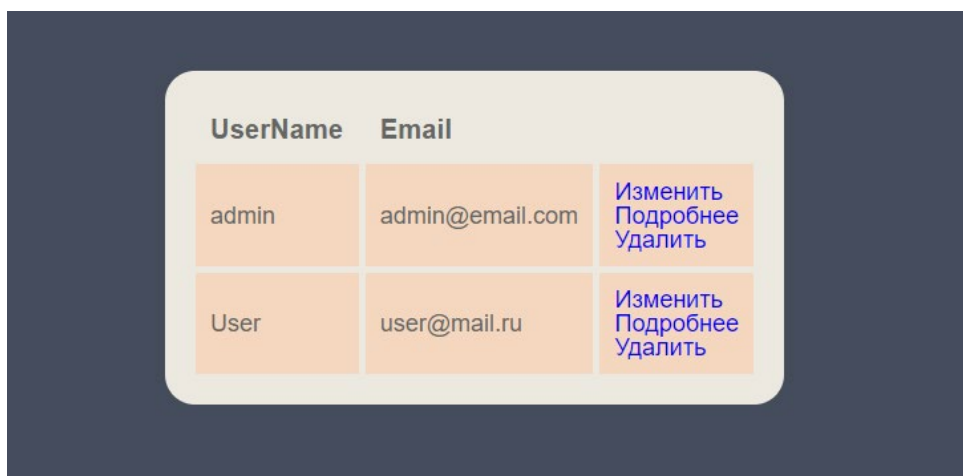
4) Кнопка Альбомы (Рисунок 4.4.3) – представляет таблицу всех имеющихся в базе данных альбомов, а также все возможные действия по изменению данных альбомов (Рисунок 4.4.6).

Обложка	Название альбома	Исполнители	Дата релиза	
	Домашний	Макс Корж	27.06.2017	Изменить Подробнее Удалить
	Уроборос: Улица 36	Скриптонит	16.12.2017	Изменить Подробнее Удалить
	Иуда	Хаски	23.02.2019	Изменить Подробнее Удалить
				Изменить

0:00 | Иуда Хаски | 0:00

Рисунок 4.4.6 – Таблица альбомов

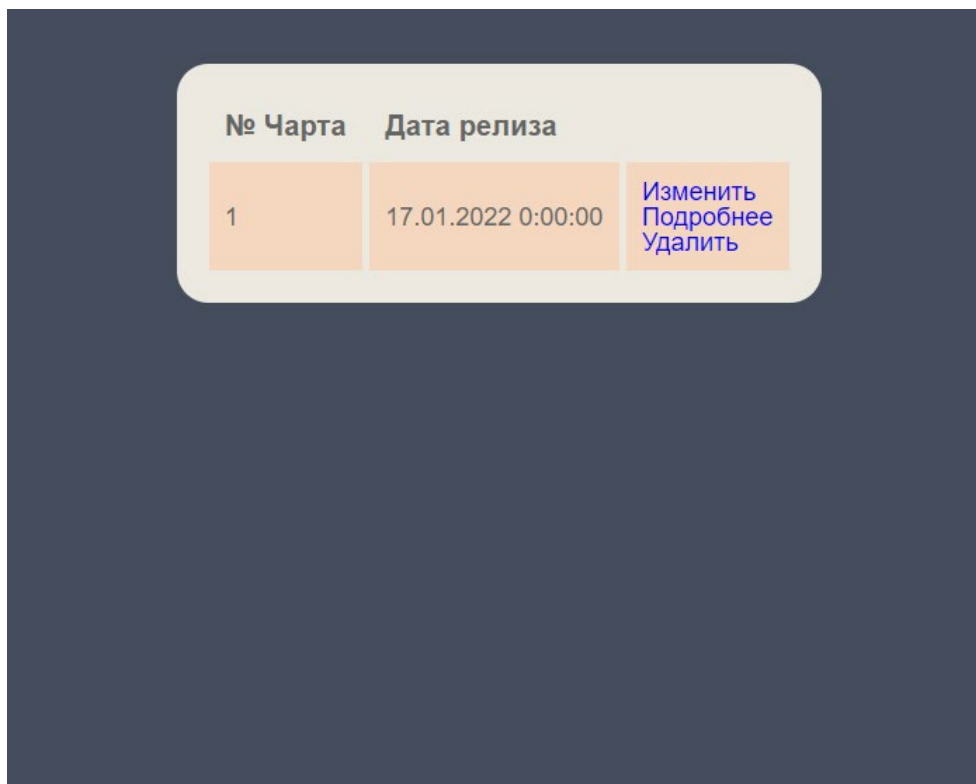
5) Кнопка пользователей (Рисунок 4.4.3) – представляет таблицу всех имеющихся в базе данных пользователей, а также все возможные действия по изменению данных пользователей (Рисунок 4.4.7).



UserName	Email	
admin	admin@email.com	Изменить Подробнее Удалить
User	user@mail.ru	Изменить Подробнее Удалить

Рисунок 4.4.7 – Таблица пользователей

6) Кнопка недельные чарты (Рисунок 4.4.3) – отображает таблицу всех имеющихся в базе данных недельных чартов, а также все возможные действия по изменению данных недельных чартов (Рисунок 4.4.8).



№ Чарта	Дата релиза	
1	17.01.2022 0:00:00	Изменить Подробнее Удалить

Рисунок 4.4.8 – Таблица недельных чартов

Руководство пользователя:

Для возможности прослушивания композиций главной страницы и недельных пользователь может не проходить авторизацию, но если пользователю потребуется воспользоваться личным хранилищем альбомов и композиций, то он будет вынужден пройти авторизацию с логином и паролем пользователя (Рисунок 4.4.1). Если же пользователь ранее не регистрировался на платформе, ему необходимо пройти регистрацию.

1) В ходе регистрации пользователю необходимо ввести электронную почту, логин (по которому он в дальнейшем будет производить авторизацию) и пароль (Рисунок 4.4.9). Всем регистрирующимся пользователям система автоматически присваивает роль “Пользователь”.

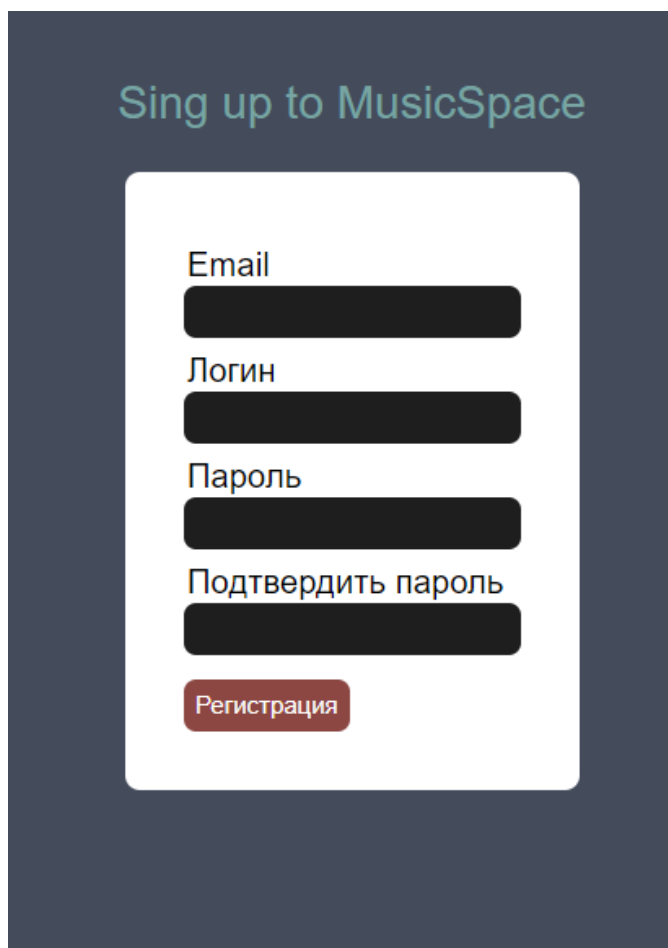
The image shows a registration form titled "Sing up to MusicSpace" in a light blue font. The form is contained within a white rounded rectangle on a dark blue background. It includes four input fields with labels: "Email", "Логин" (Login), "Пароль" (Password), and "Подтвердить пароль" (Confirm password). Each label is in a dark blue font, and the corresponding input field is a dark blue rounded rectangle. At the bottom of the form is a red button with the text "Регистрация" (Registration) in white.

Рисунок 4.4.9 – Форма регистрации

2) После авторизации/регистрации пользователю становится доступным добавление альбомов и музыкальных композиций в личное хранилище (профиль), который отображен на Рисунке 4.4.10.

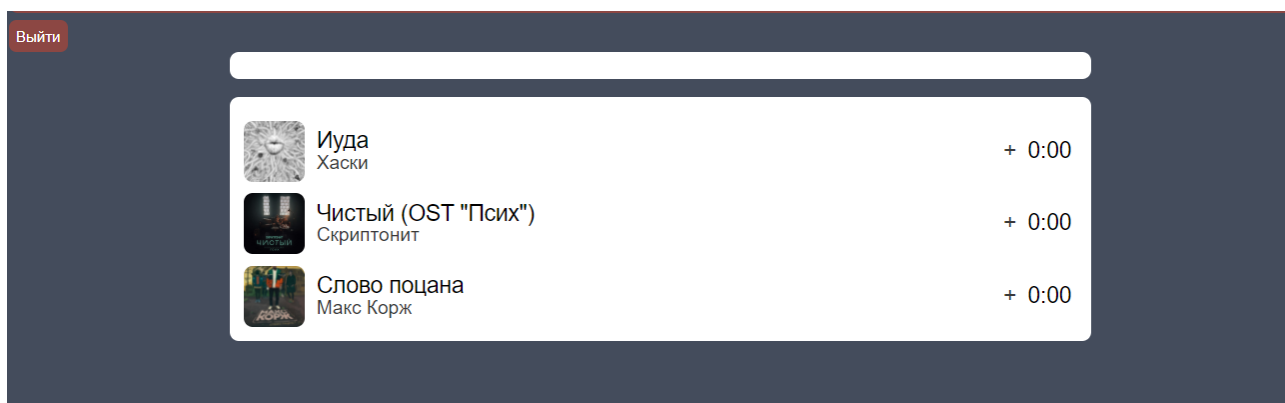


Рисунок 4.4.10 – Профиль пользователя

3) Для просмотра и прослушивания композиций недельных чартов пользователю необходимо перейти на вкладку главного меню “Charts” (Рисунок 4.4.3). После этого пользователю станут доступны композиции текущего недельного чарта в убывающем порядке количества недельных прослушиваний (Рисунок 4.4.11).

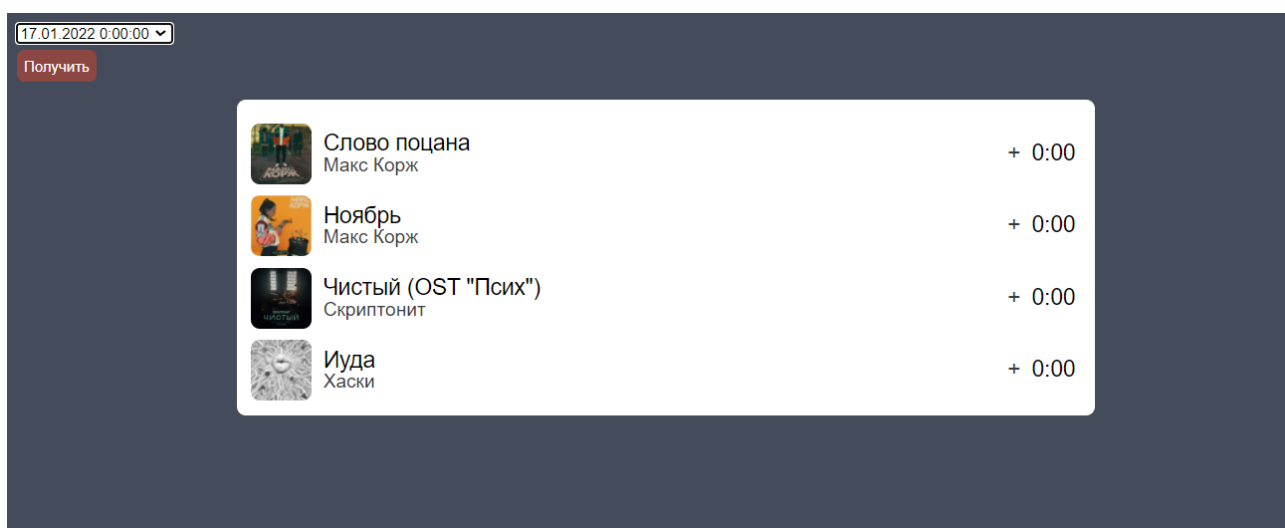


Рисунок 4.4.11 – Страница недельных чартов

4) Для отслеживания предыдущих чартов пользователю необходимо выбрать интересующую дату в выпадающем списке в левом верхнем углу страница (Рисунок 4.4.11) и нажать на кнопку “Получить”.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы было разработано веб-приложение для прослушивания музыки.

Работа выполнена в несколько этапов: была выбрана архитектура, СУБД и разработана структура системы, реализовано веб-приложение.

Были сформированы навыки по разработке и реализации программного приложения с базой данных.

Были реализованы все поставленные задачи.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1) <https://soundcloud.com/>
- 2) <https://www.spotify.com/ru-en/>
- 3) <https://www.audiomania.ru/content/art-5640.html>
- 4) <https://docs.microsoft.com/ru-ru/dotnet/csharp/linq/>
- 5) <https://ru.wikipedia.org/wiki/Spotify>
- 6) <https://tqm.com.ua/likbez/article/pochemu-net-ru>
- 7) Шнырёв, С.Л. Базы данных: учебное пособие для вузов - М. : НИЯУ МИФИ, 2011. — 224 с. — Режим доступа: <http://e.lanbook.com/book/75809>
- 8) Ревунков, Г.И. Базы и банки данных - М.: МГТУ им. Н.Э. Баумана, 2011. — 68 с. — Режим доступа: <http://e.lanbook.com/book/52425>
- 9) Ревунков, Г.И. Проектирование баз данных - М.: МГТУ им. Н.Э. Баумана, 2009. — 20 с. — Режим доступа: <http://e.lanbook.com/book/52390>
- 10) Кудрявцев, К.Я. Создание баз данных: учебное пособие — М.: НИЯУ МИФИ, 2010. — 155 с. — Режим доступа: <http://e.lanbook.com/book/75822>
- 11) Сидоров В.Н., Сломинская Е.Н., Полникова Т.В., Макарова О.Ю. Оформление графической части выпускной квалификационной работы. Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2016.