Минобрнауки россии

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных *наук*

Кафедра информационные системы и технологии

Разработка музыкального приложения “MusicZone”

Курсовой проект

*09.03.02 Информационные системы и технологии*

*Программная инженерия в информационных системах*

Зав. кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С. Д. Махортов, д.ф.- м.н.,доцент \_\_.\_.20\_\_

Обучающийся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Р.С. Шевцов, 3 курс, д/о

Обучающийся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_М.В. Гончаренко, 3 курс, д/о

Обучающийся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_М.С. Артемьев, 3 курс, д/о

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.С. Тарасов, ст. преподаватель

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_И.В.Клейменов, ассистент

Воронеж 2023

**Содержание**

[Введение 4](#_Toc136355095)

[1 Постановка задачи 5](#_Toc136355096)

[1.1 Цели создания системы 5](#_Toc136355097)

[1.2 Требования к разрабатываемой системе 5](#_Toc136355098)

[1.3 Задачи проекта 5](#_Toc136355099)

[1.3.1 Для не зарегистрированного пользователя 5](#_Toc136355100)

[1.3.2 Для зарегистрированного пользователя 5](#_Toc136355101)

[1.3.3 Для администратора 5](#_Toc136355102)

[2 Анализ предметной области 6](#_Toc136355103)

[2.1 Терминология 6](#_Toc136355104)

[2.2 Обзор аналогов 8](#_Toc136355105)

[2.2.1 Яндекc.Музыка 8](#_Toc136355106)

[2.2.2 YouTubeMusic 9](#_Toc136355107)

[2.2.3 Spotify 11](#_Toc136355108)

[2.3 Моделирование системы 12](#_Toc136355109)

[2.3.1 Диаграмма в стиле методологии IDEF0 12](#_Toc136355110)

[2.3.2 Диаграмма прецедентов 13](#_Toc136355111)

[2.3.3 Диаграммы классов 14](#_Toc136355112)

[2.3.4 Диаграммы последовательности 15](#_Toc136355113)

[2.3.5 Диаграмма развертывания 19](#_Toc136355114)

[2.3.6 Диаграмма состояния 19](#_Toc136355115)

[2.3.7 Диаграмма объектов 19](#_Toc136355116)

[2.3.8 Диаграммы активности 20](#_Toc136355117)

[2.3.9 Диаграмма сотрудничества 21](#_Toc136355118)

[3 Реализация 21](#_Toc136355119)

[3.1 Средства реализации 21](#_Toc136355120)

[3.2 Реализация базы данных 23](#_Toc136355121)

[3.2.1 ER-диаграмма 23](#_Toc136355122)

[3.2.2 Физическая модель базы данных 24](#_Toc136355123)

[3.3 Реализация клиентской части 24](#_Toc136355124)

[3.3.1 Форма для поиска музыки 24](#_Toc136355125)

[3.3.2 Форма экрана плеера 25](#_Toc136355126)

[3.3.3 Форма экрана авторизации 26](#_Toc136355127)

[3.3.4 Форма экрана регистрации 27](#_Toc136355128)

[3.3.5 Форма экрана восстановления пароля 27](#_Toc136355129)

[3.3.6 Форма экрана личной страницы 29](#_Toc136355130)

[3.3.7 Форма экрана редактирования профиля 30](#_Toc136355131)

[3.3.8 Форма экрана загруженной музыки 31](#_Toc136355132)

[3.3.9 Форма экрана добавления музыки 31](#_Toc136355133)

[3.3.10 Форма экрана редактирования музыки 32](#_Toc136355134)

[3.4 Серверная часть 33](#_Toc136355135)

[3.4.1 Основные положения 33](#_Toc136355136)

[4 Тестирование 34](#_Toc136355137)

[Заключение 35](#_Toc136355138)

Введение

В современном мире мобильные приложения стали неотъемлемой частью нашей повседневной жизни, предоставляя нам доступ к различным сервисам и удобствам на расстоянии нажатия кнопки. Одной из популярных категорий мобильных приложений являются музыкальные приложения, которые с успехом сочетают в себе развлекательную и практическую функции.

Музыкальные приложения – это программные приложения для мобильных устройств, предназначенные для прослушивания, организации и обмена музыкой. Они позволяют пользователям наслаждаться музыкой в любое время и в любом месте, предлагая широкий спектр функций, включая потоковую передачу музыки, создание плейлистов, поиск новых артистов и треков, радиостанции и многое другое. Благодаря мобильным музыкальным приложениям, музыка стала более доступной и персонализированной, позволяя нам настроиться на нужное настроение или создать собственный саундтрек к повседневным моментам нашей жизни.

В данной курсовой работе рассматривается процесс разработки собственного мобильного музыкального приложения

В рамках работы будут рассмотрены различные аспекты разработки мобильного музыкального приложения, начиная с анализа предметной области, определения его концепции и основных функциональных требований. Затем будет изучено проектирование пользовательского интерфейса и пользовательского опыта, с учетом современных тенденций и личных практик в этой области. Важное внимание будет уделено выбору и интеграции соответствующих технологий и API для обеспечения необходимых функций, таких как передача музыки, поиск треков и другие.

1. Постановка задачи
   1. Цели создания системы

Целью данной работы является создание мобильного музыкального приложения для поиска и прослушивания музыки.

* 1. Требования к разрабатываемой системе
* Обеспечение авторизации и аутентификации пользователей
* Использование механизмов защиты от SQL-инъекций
* Использование протокола передачи данных HTTP
* Приложение должно быть построено на трехуровневой архитектуре
  1. Задачи проекта
     1. Для не зарегистрированного пользователя
* Поиск музыки
* Прослушивания найденной музыки
  + 1. Для зарегистрированного пользователя
* Поиск музыки
* Прослушивание найденной музыки
* Добавить музыку на личную страницу
  + 1. Для администратора
* Загрузка, редактирование и удаление музыки со стороны администратора приложения
* Поиск музыки
* Прослушивания найденной музыки
* Добавить музыку на личную страницу

1. Анализ предметной области
   1. Терминология

* Мобильное приложение — программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для конкретной платформы (iOS, Android, WindowsPhone и т. д.)
* Android-приложение — программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для платформы Android
* Клиент — это аппаратный или программный компонент вычислительной системы, посылающий запросы серверу
* Сервер — выделенный или специализированный компьютер для выполнения сервисного программного обеспечения
* База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД)
* HTTP — это протокол, позволяющий получать различные ресурсы, например HTML-документы. Протокол HTTP лежит в основе обмена данными в Интернете
* SQL-запросы — это наборы команд для работы с реляционными базами данных
* Дизайн-макет — это схематичное изображение финальной идеи с указанием всех деталей. В нем указываются концепция, шрифты, тексты, изображения, расположение всех элементов и общая картина продукта
* Аутентификация — процедура проверки подлинности, например, проверка подлинности пользователя путем сравнения введенного им пароля с паролем, сохраненным в базе данных
* Авторизация — предоставление определенному лицу или группе лиц прав на выполнение определенных действий
* Android — это операционная система с открытым исходным кодом, созданная для мобильных устройств на основе модифицированного ядра Linux
* Фреймворк — программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта
* SQL-инъекция — внедрении в запрос произвольного SQL-кода, который может повредить данные, хранящиеся в БД или предоставить доступ к ним
* HTTPS — расширение протокола HTTP для поддержки шифрования в целях повышения безопасности
* Пользователь — человек, который использует приложение
* Аккаунт или учетная запись — это персональная страница пользователя или личный кабинет, который создается после регистрации на сайте
* Frontend — клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса
* Backend — программно-аппаратная часть сервиса, отвечающая за функционирование его внутренней части
* REST — архитектурный стиль взаимодействия компонентов распределённого приложения в сети
* API — описание взаимодействия одной компьютерной программы с другой
* Трек — это любая звуковая дорожка, электронная музыкальная композиция (мелодия, вокал)
* Песня — это композиция, состоящая из поэтического текста и мелодии. Они выполняют равные смысловые функции
  1. Обзор аналогов

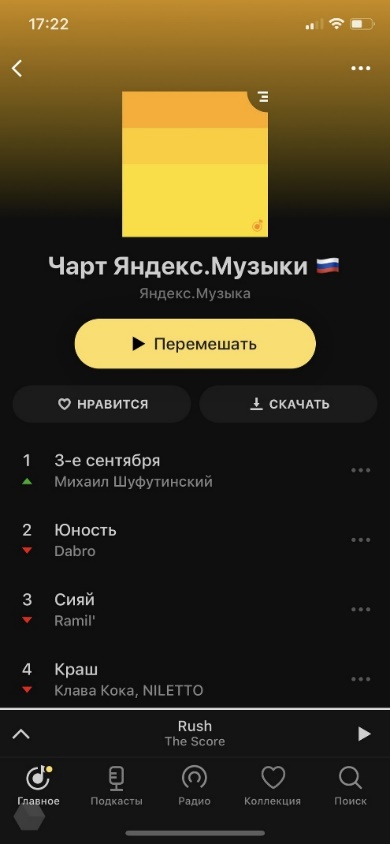
Разрабатывая приложение, основной задачей которого является хранение, распространение и прослушивание музыки, необходимо рассматривать разработку с точки зрения актуальности и уникальности проекта. Для оценки этих качеств необходимо прибегнуть к рассмотрению аналогов разрабатываемого приложения, адекватно оценивая все положительные и негативные черты того или иного продукта.

Помимо мировой практики разработки продуктов данного направления, необходимо учитывать специфику текущего времени, когда некоторая продукция, сервис и т.д. может быть не доступна на территории страны, которая является целевой аудиторией разрабатываемого продукта.

* + 1. Яндекc.Музыка

Яндекс.Музыка — это онлайн-сервис, предоставляющий простой и удобный доступ к миллионам аудио-треков, подборкам, плейлистам, радиостанциям и музыкальным подкастам. Яндекс.Музыка рекомендует своим пользователям новинки и композиции, исходя из их личных предпочтений, возраста, настроения и времени суток. Также сервис позволяет создавать персональные плейлисты, синхронизировать музыку между разными устройствами. Яндекс.Музыка доступна в большинстве стран и на большинстве релевантных устройств.

Яндекс.Музыка обладает широким спектром предоставляемых услуг и с точки зрения авторского контента, выплачивая хорошие royalty



1. Интерфейс страницы «Яндекс.Музыка»

Недостатки:

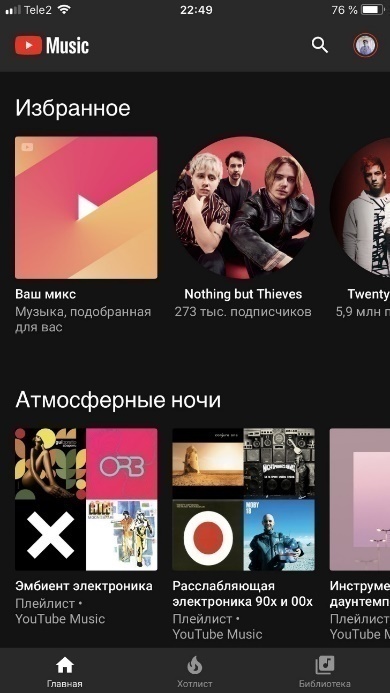
* Ограниченный доступ к музыкальным трекам
* Высокая стоимость подписки на сервис
* Рекламные блоки на платформе могут быть раздражающими
* Неудобство в использовании приложения на некоторых устройствах
* Ограниченный доступ к облаку и потеря данных
* Неполное совпадение исполнителей, альбомов и треков
* Не удается найти все композиции и исполнителей
* Невозможность загрузить личную музыку в плейлисты
* Вирусные подборки, появляющиеся в рекомендациях, не соответствующие запросам пользователя
  + 1. YouTubeMusic

YouTubeMusic — это онлайн-музыкальный сервис, представленный компанией YouTube в 2018 году. Сервис предоставляет своим пользователям доступ к миллионам песен и музыкальным видео, а также плейлистам, радиостанциям и рекомендациям на основе предпочтений пользователя.

Стоит отметить, что на фоне многих известных зарубежных стриминговых сервисов именно продукт Google отличается хорошо работающими рекомендациями, что идеально могут скрасить вечер слушателя. В плане работы с рекомендациями данный продукт конкурирует с такими стриминговыми сервисами как Spotify, Dizzer и т.д.

YouTubeMusic доступен для использования на различных устройствах, таких как компьютеры, смартфоны и планшеты, и поддерживает различные операционные системы, включая Android и iOS. Он также интегрирован с другими сервисами Google, такими как GooglePlay Музыка.

Сервис предлагает возможность прослушивания музыки как в режиме онлайн, так и офлайн, в зависимости от настроек пользователя. Он также включает функции автоматического плейлиста и адаптивного режима, который позволяет пользователям настроить оптимальное качество звука в зависимости от доступной связи с интернетом.



1. Интерфейс страницы «YouTubeMusic»

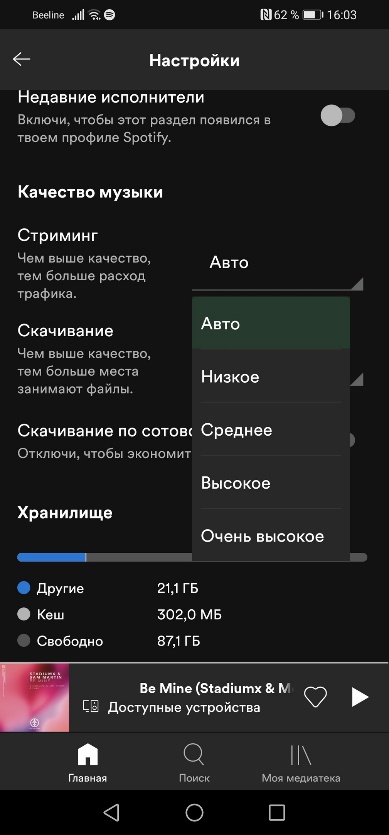
Недостатки:

* Громоздкий интерфейс
* Плохая работа с карточками музыкантов
* Создание темы исполнителя без возможности привязать актуальную страницу исполнителя в YouTube
* Различное качество звука от трека к треку
* Отсутствие некоторых функций, таких как создание временного плейлиста
  + 1. Spotify

Spotify — это стриминговый сервис, предлагающий доступ к миллионам песен, подкастам и аудиокнигам. Spotify позволяет пользователям создавать плейлисты, слушать музыку в режиме онлайн или офлайн (сохранение в кэш), делиться треками с друзьями, искать новых исполнителей и многое другое. Spotify также имеет мобильное приложение и многочисленные интеграции с другими платформами и устройствами.

На данный момент Spotify — это флагман от мира стриминговых сервисов. Его алгоритмы подборки очень точны, а работа с артистами, посредством взаимодействия через цифровой музыкальный дистрибьютор, самая удобная из существующих на данный момент.

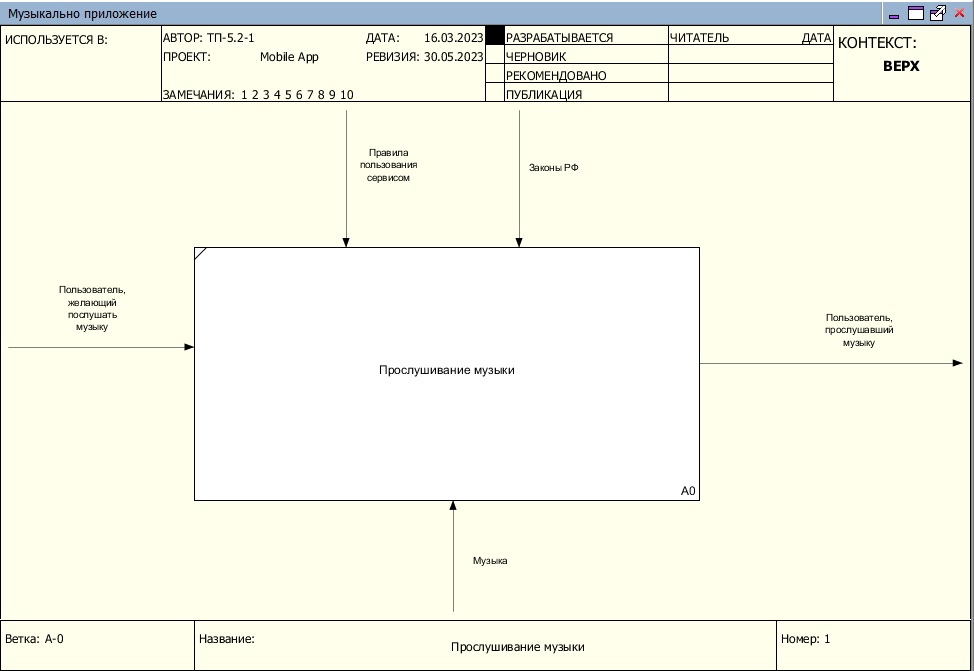
Spotify предлагает широкий спектр услуг по распространению музыки определенных исполнителей, включая треки даже малоизвестных исполнителей по жанрам и стилю подходящие под образец в подборки, которые рекомендуются пользователям с соответствующими музыкальными предпочтениями.



1. Интерфейс страницы «Spotify»

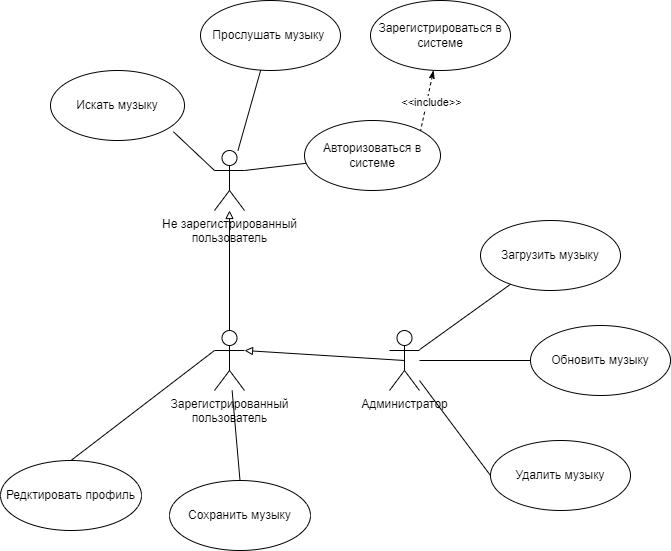
Недостатки:

* Возможно сохранение не более 10 тысяч треков
* Spotify не доступен в некоторых странах мира, что ограничивает число потенциальных пользователей
* Ограниченный доступ к музыке, в частности, к новым альбомам известных артистов
* Spotify потребляет намного больше интернет-трафика, чем другие сервисы
  1. Моделирование системы
     1. Диаграмма в стиле методологии IDEF0



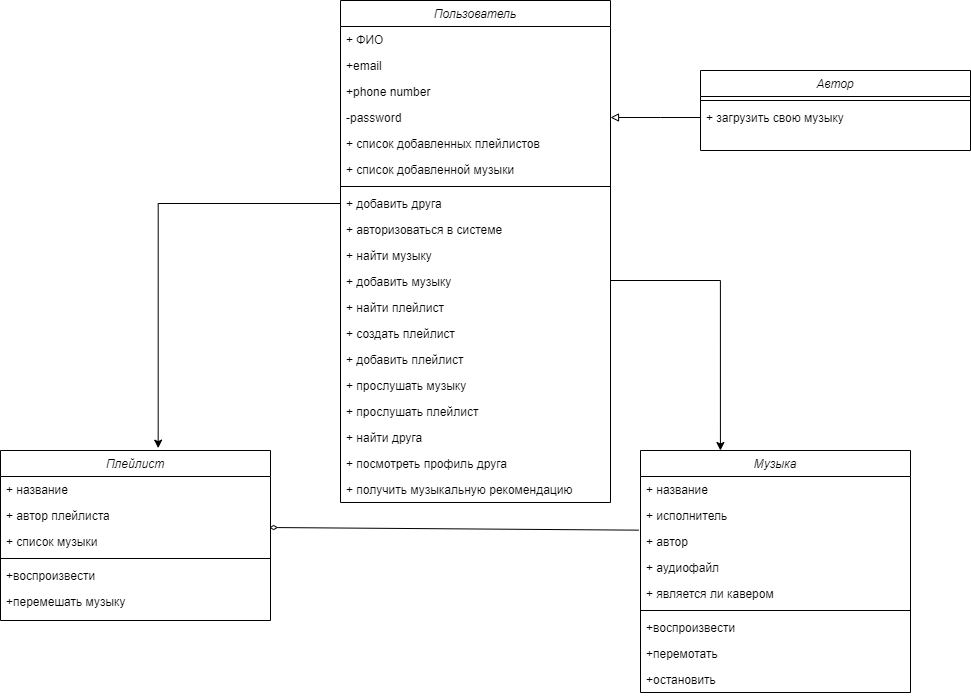
1. Диаграмма в стиле методологии IDEF0
   * 1. Диаграмма прецедентов

Рассмотрим полную диаграмму для использования приложения разными типами пользователей (неавторизованный, авторизованный, исполнитель). В данном случае необходимость составления диаграммы прецедентов продиктована прежде всего тем, что use-case диаграмма — это инструмент для моделирования системы и понимания ее функциональности и потребностей пользователей. Они помогают в определении основных действий, которые пользователь должен совершить в системе, чтобы достичь определенных целей. Они также позволяют определить возможные риски и проблемы, которые могут возникнуть в ходе использования системы.



1. Use-Case диаграмма пользования приложением
   * 1. Диаграммы классов

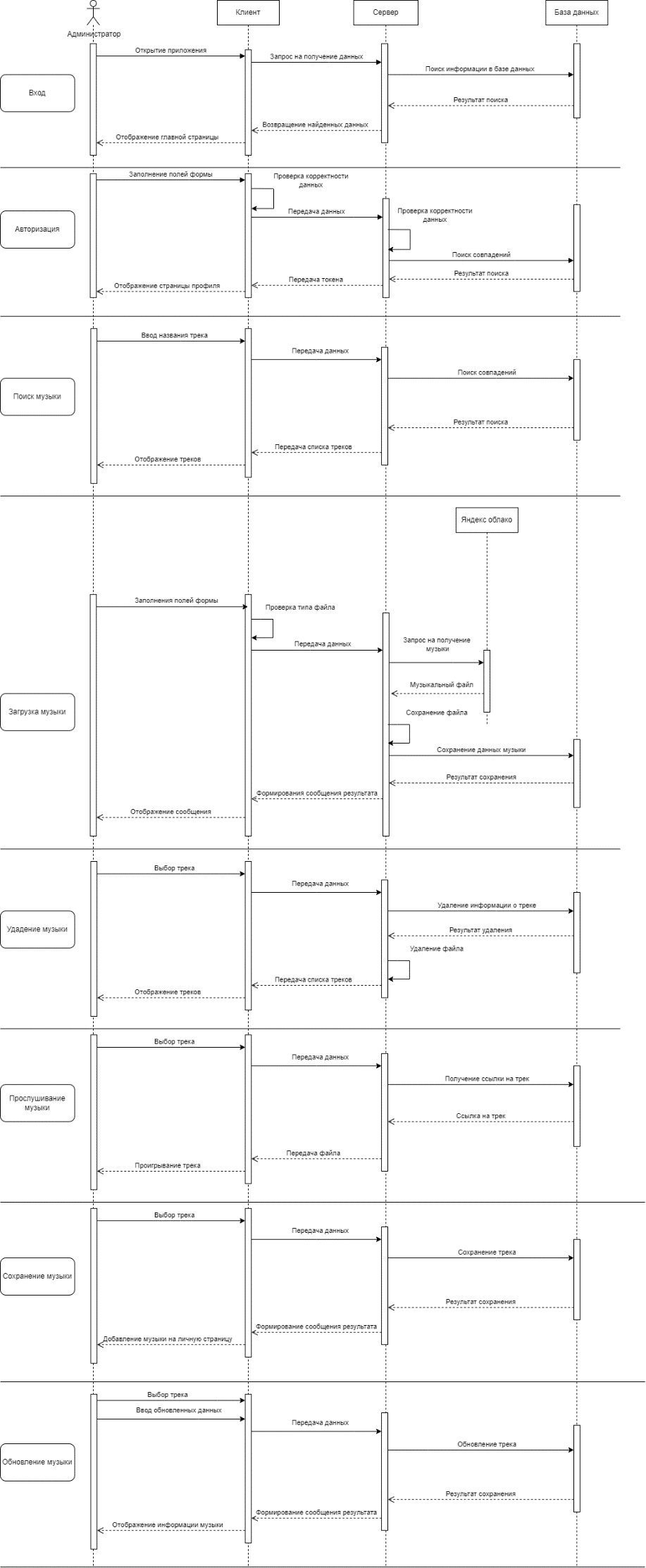
В данном пункте будут рассмотрены диаграммы классов-сущностей серверной части приложения. Диаграмма классов позволяет легко понять структуру проекта и отношения между классами. Это позволяет всем участникам проекта быть на одной волне, что позволяет сократить количество ошибок и повторных разъяснений в процессе разработки, помимо этого классы, которые созданы на диаграмме, обычно хорошо описываются, что позволяет легче отлаживать код и тестировать приложение в целом.



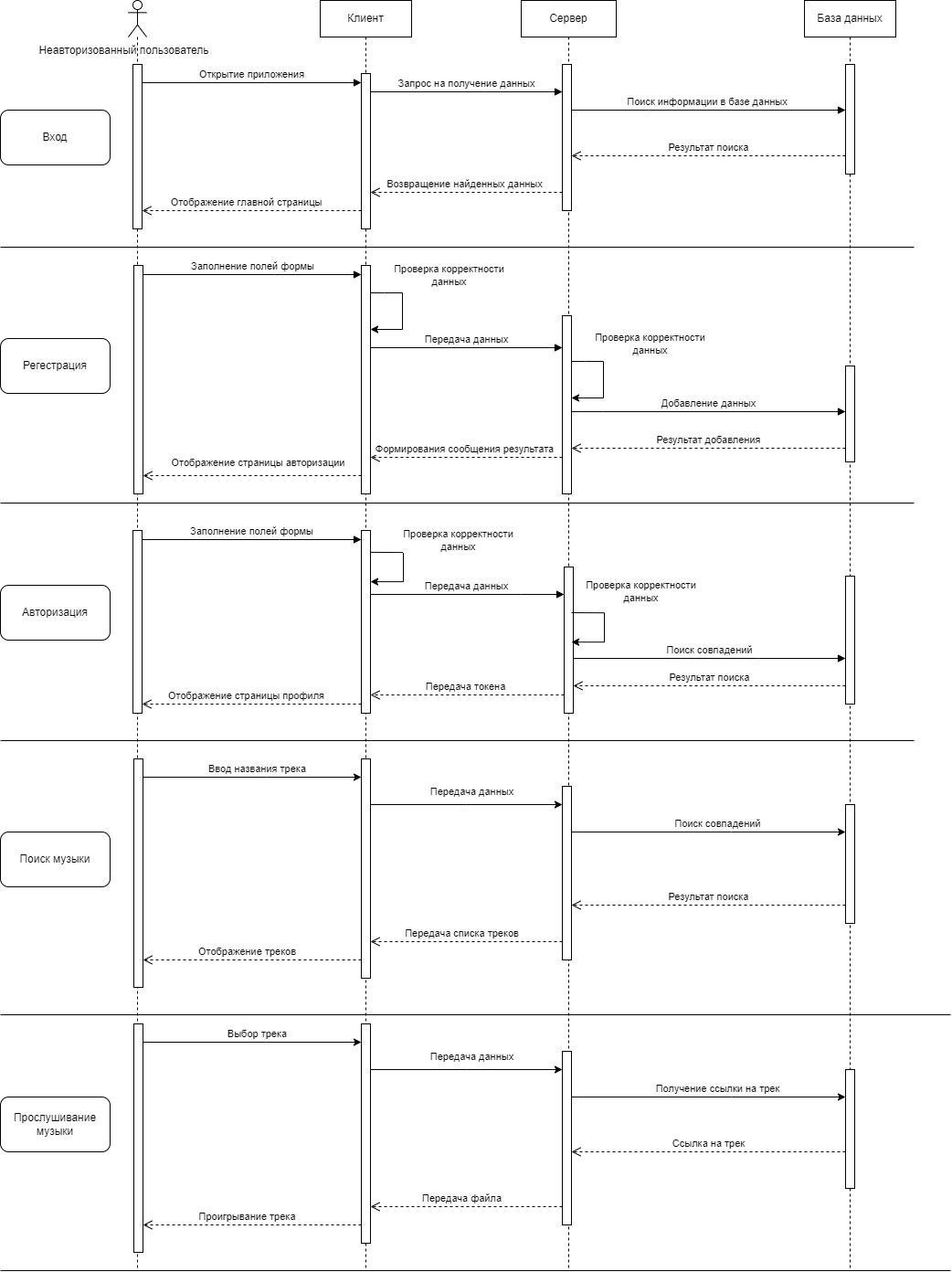
1. Диаграмма классов-сущностей серверной части приложения.
   * 1. Диаграммы последовательности

Диаграмма последовательности является важным инструментом для проекта, который помогает более глубоко понимать процесс, улучшать его эффективность и упрощать взаимодействие.

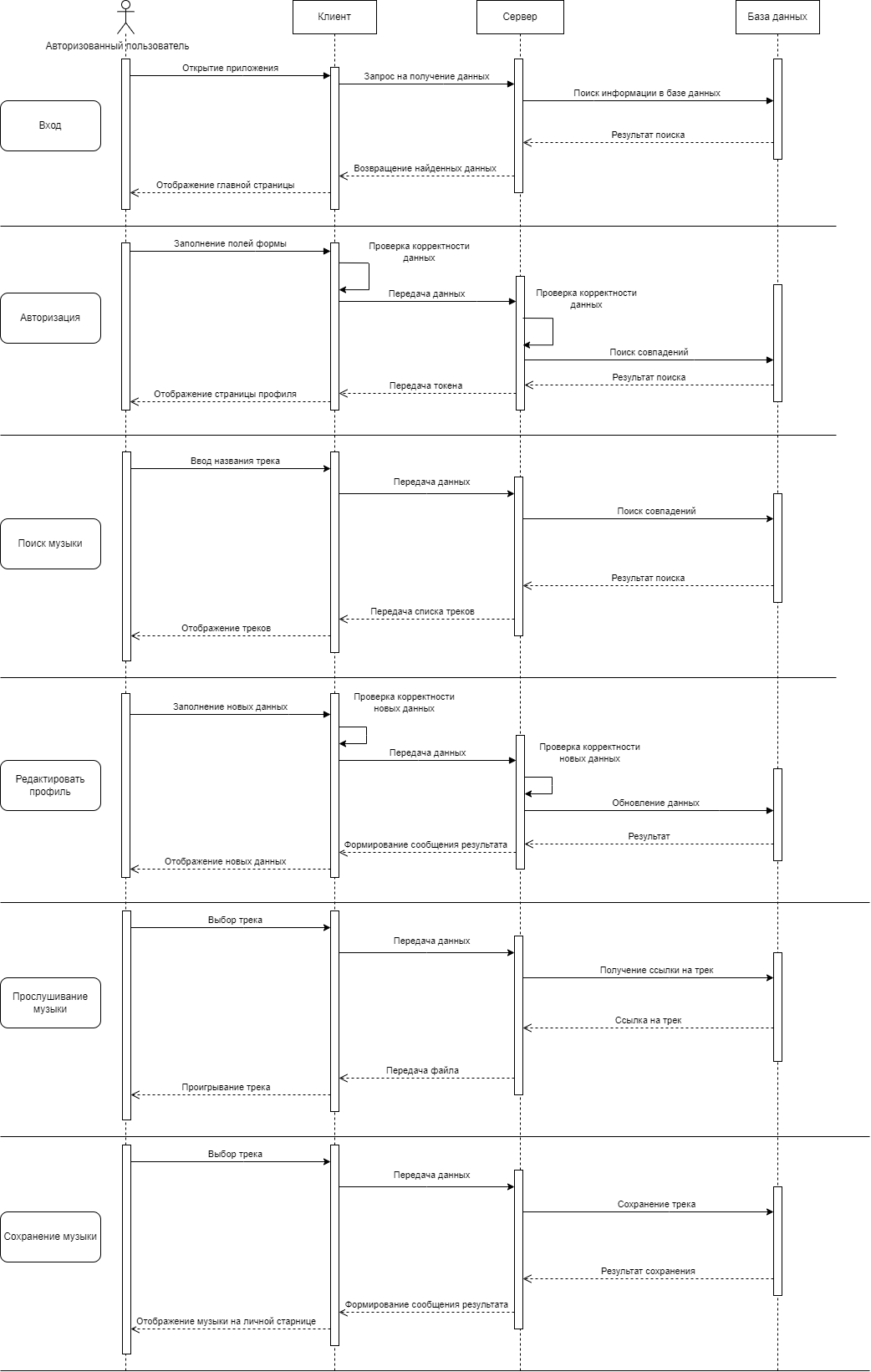
Рассмотрим диаграмму последовательности администратора приложения.



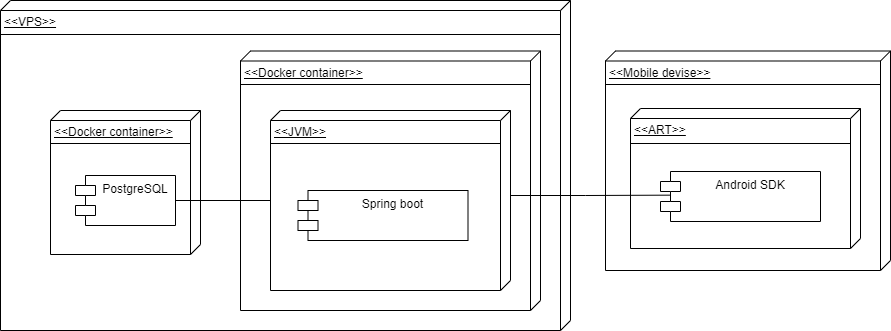
1. Диаграмма последовательности администратора



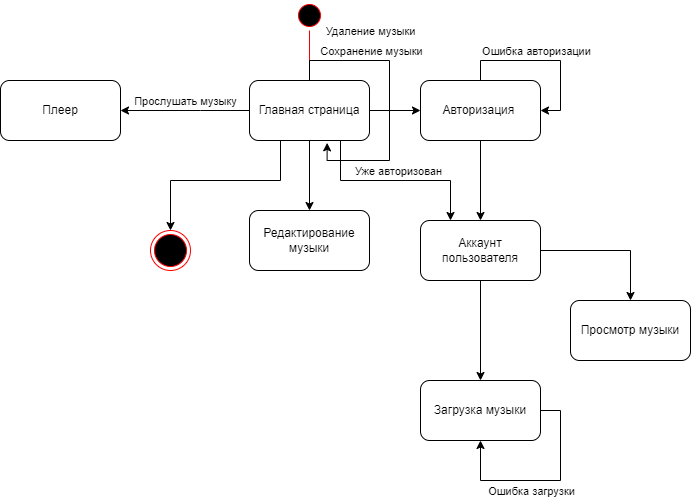
1. Диаграмма последовательности неавторизированного пользователя



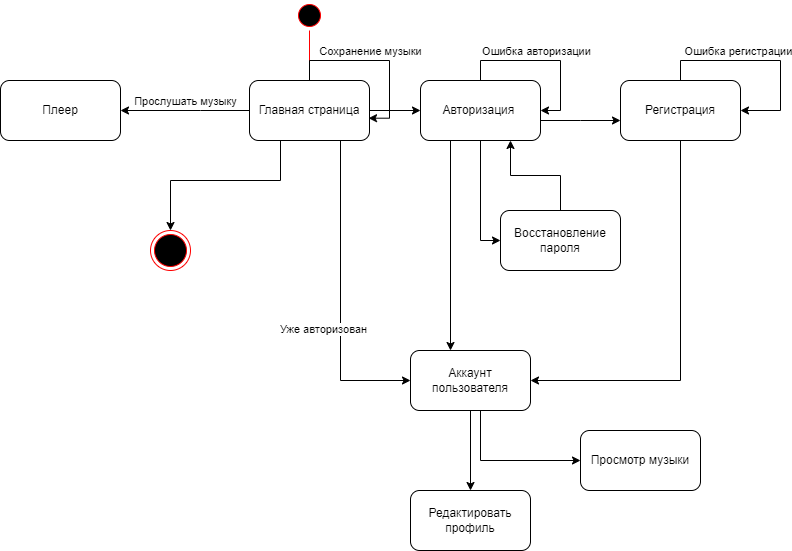
1. Диаграмма последовательности авторизованного пользователя
   * 1. Диаграмма развертывания



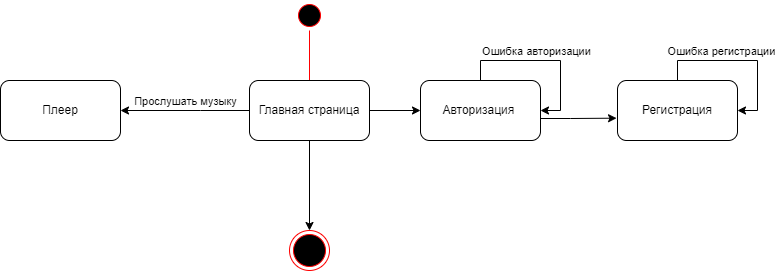
1. Диаграмма развертывания приложения
   * 1. Диаграммы состояния



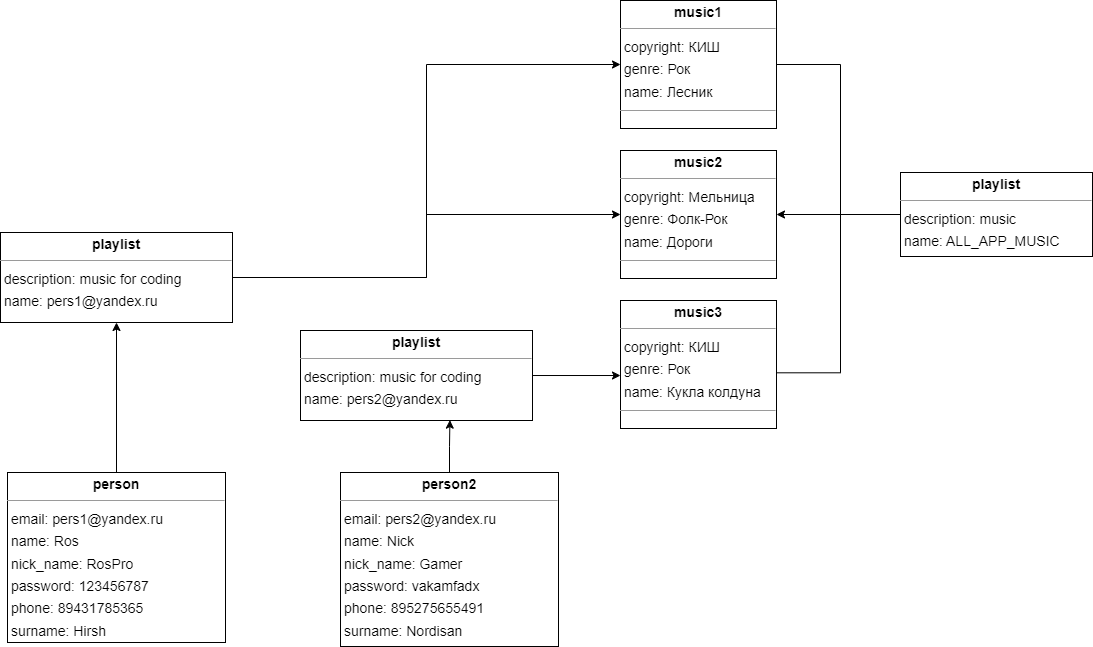
1. Диаграмма состояния администратора



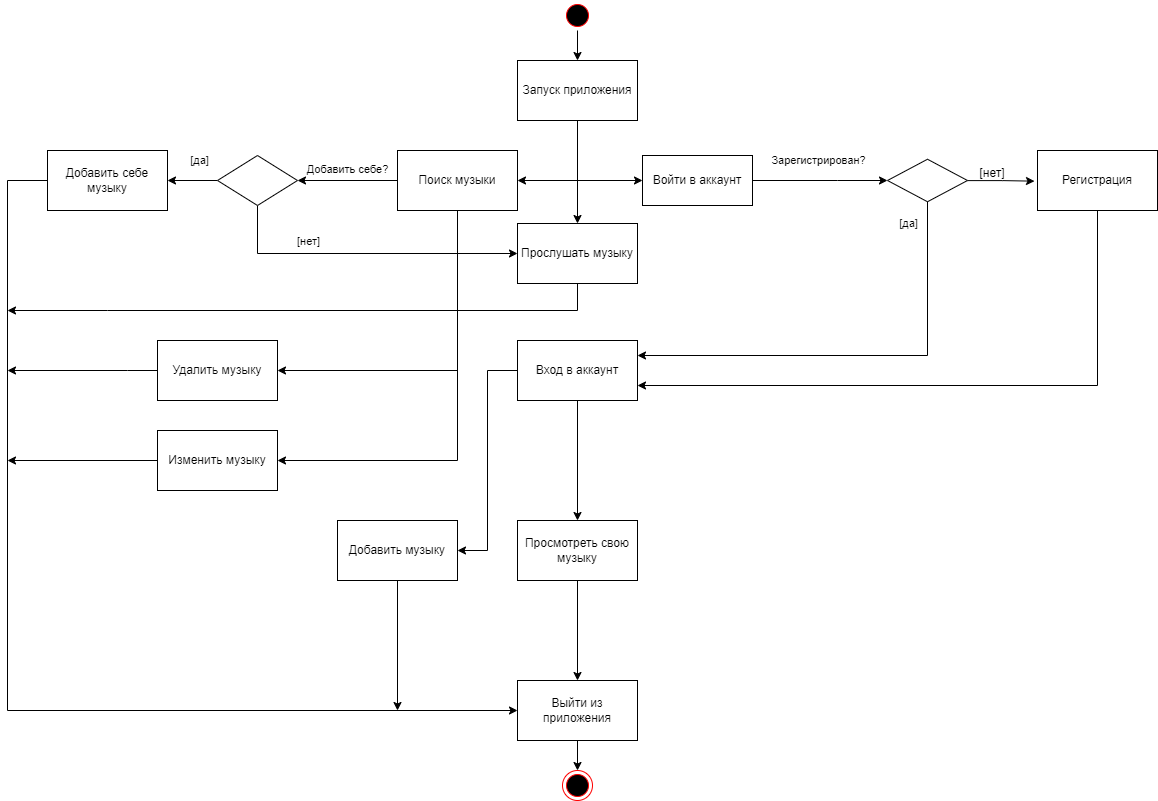
1. Диаграмма состояния неавторизованного пользователя



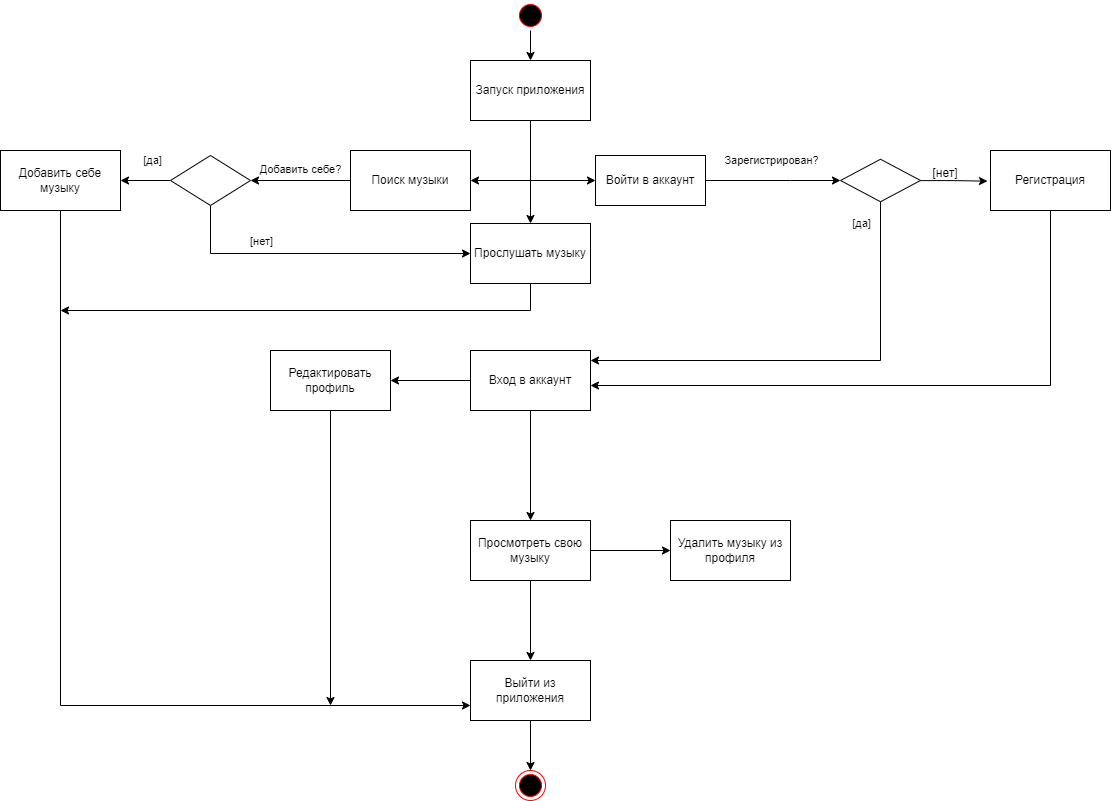
1. Диаграмма состояния авторизованного пользователя
   * 1. Диаграмма объектов



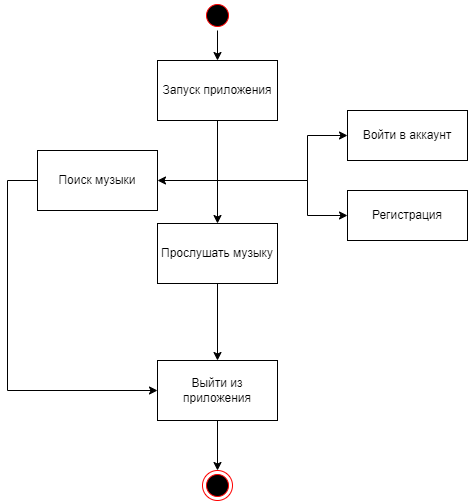
1. Диаграмма объектов
   * 1. Диаграммы активности



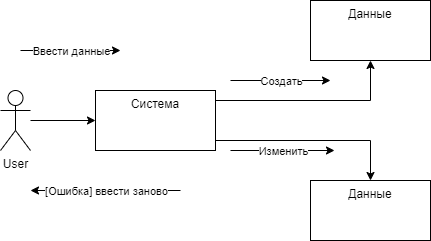
1. Диаграмма активности администратора



1. Диаграмма активности авторизованного пользователя



1. Диаграмма активности неавторизованного пользователя
   * 1. Диаграмма сотрудничества



1. Диаграмма сотрудничества
2. Реализация
   1. Средства реализации

Ниже приведен перечень используемых технологий, который в ходе разработки может расширяться.

Backend

* Java
* SpringBootFramework
* PostgreSQL
* FlyWay
* Docker

Frontend:

* Android SDK

Инструменты для ведения документации:

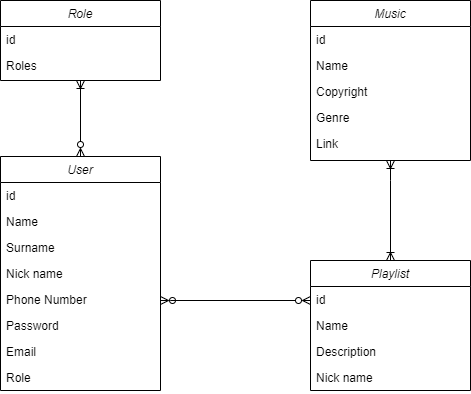
* Miro
* Swagger
* Draw.io
* Ramus
* Figma

Дополнительный инструментарий:

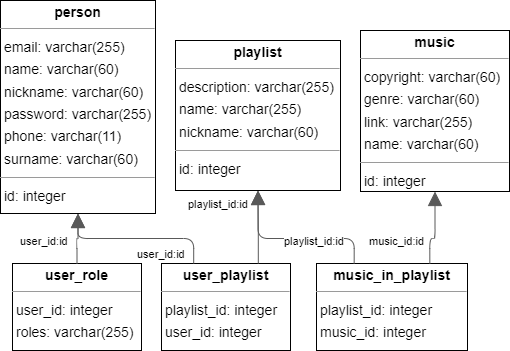
* Git
* GitHub
* Trello
  1. Реализация базы данных

Для хранения данных была выбрана база данных PostgreSQL. Она является продуктом с открытым исходным кодом, который поддерживается многими серверами. Присутствует поддержка различных типов данных. PostgreSQL поддерживает множественные типы данных, такие как числа разной точности, тексты с различными кодировками, изображения, звуки, видео, XML-документы, JSON-объекты и многие другие.

* + 1. ER-диаграмма



1. ER-диаграмма базы данных
   * 1. Физическая модель базы данных



1. Физическая модель базы данных
   1. Реализация клиентской части

Для реализации клиентской части приложения было выбрано cсочетание Android SDK и языка Java. Этот набор обеспечивает разработчикам множество возможностей для создания мобильных приложений под Android. Они могут использовать SDK для доступа к аппаратным возможностям устройства, взаимодействия с различными сервисами и API, разработки пользовательского интерфейса и многого другого. Java, в свою очередь, предоставляет надежный и мощный язык программирования, который позволяет разработчикам создавать сложные и высокопроизводительные приложения.

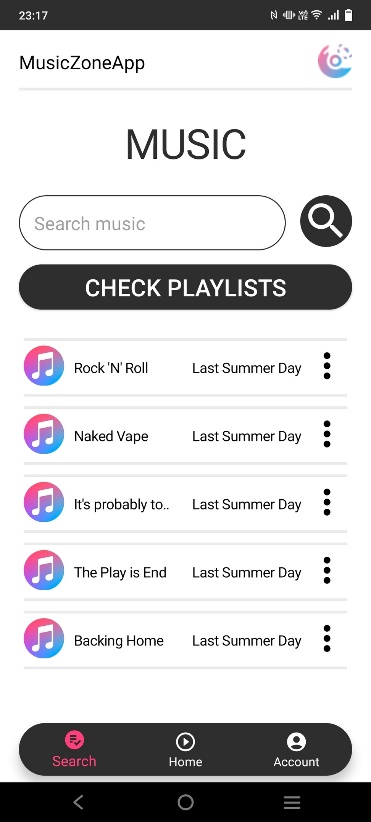
* + 1. Форма для поиска музыки

Пользователь (авторизованный и не авторизованный) имеет возможность найти музыку или для прослушивания по названию или автору. В качестве ответа на его запрос будет выведен список треков. Пользователь может нажать на трек, чтобы прослушать его.

Зарегистрированный пользователь имеет возможность добавить найденную музыку себе в профиль.

Администратор имеет возможность редактировать или удалить музыку.

Снизу расположена панель с кнопками, при нажатии на соответствующую кнопку пользователь переходит на страницы: поиск/главная, прослушивание музыки, личная страница.

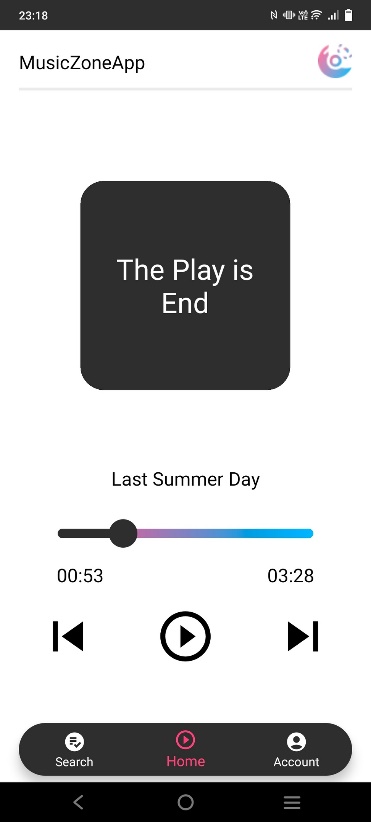


1. Форма экрана поиска музыки
   * 1. Форма экрана плеера

На данном экране расположена панель с название трека и его описанием, слайдер, отображающий текущее время трека и кнопки:

* перемотка трека
* стоп
* играть
* следующий трек

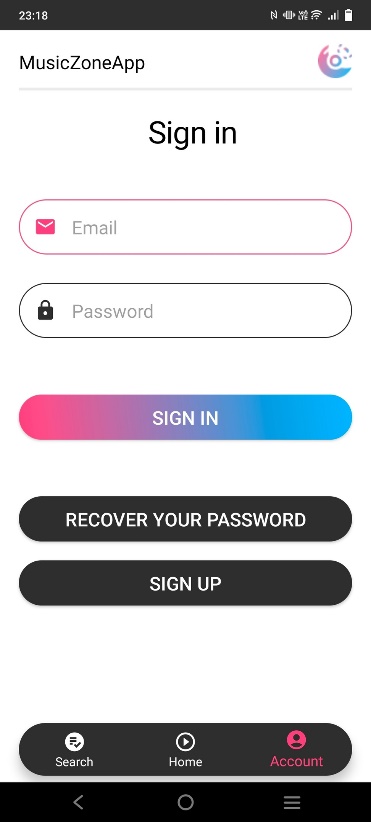
Снизу расположена панель с кнопками, при нажатии на соответствующую кнопку пользователь переходит на страницы: поиск/главная, прослушивание музыки, личная страница.



1. Форма экрана плеера
   * 1. Форма экрана авторизации

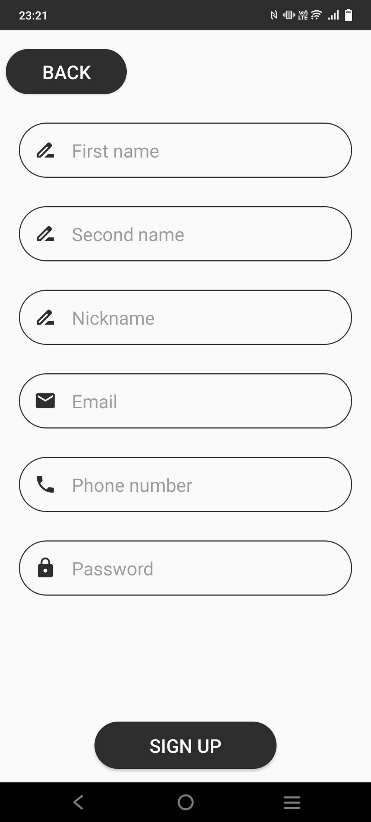
На данном экране отображены поля с вводом логина и пароля к аккаунту, кнопка войти в аккаунт, кнопка зарегистрироваться, а также кнопка восстановление пароля.

Снизу расположена панель с кнопками, при нажатии на соответствующую кнопку пользователь переходит на страницы: поиск/главная, прослушивание музыки, личная страница.



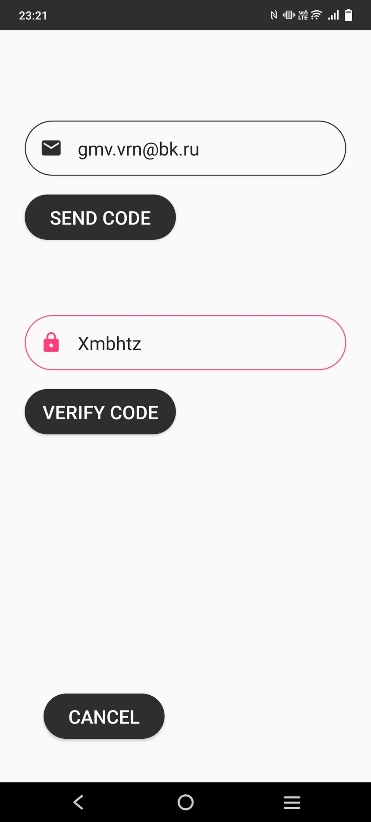
1. Форма экрана авторизации
   * 1. Форма экрана регистрации

На данном экране отображены поля для ввода имени, фамилии, псевдоним, почты, номера телефона, пароль. Ниже расположена кнопка регистрации.

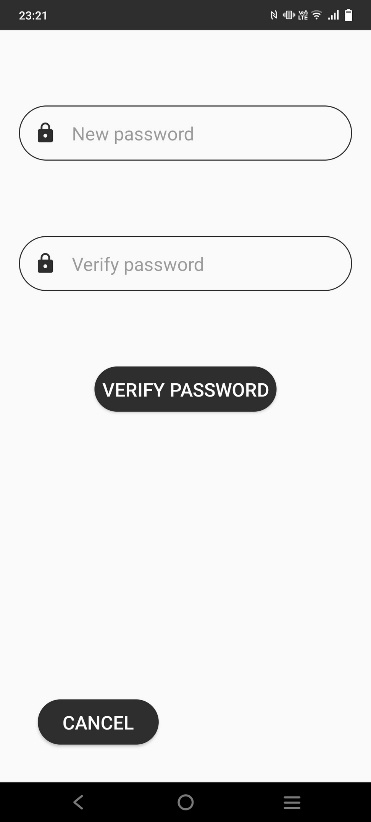


1. Форма экрана регистрации
   * 1. Форма экрана восстановления пароля

На данном экране отображены поля для ввода адреса электронной почты и кнопка отправки кода. После введения данных появляется форма для ввода полученного кода, и кнопка для подтверждения кода. При успешной проверке кода, появляется форма для заполнения нового пароля, и кнопка подтверждения нового пароля. В случае неудачи пользователь будет возвращен на начальную форму восстановления пароля.



1. Форма экрана восстановления пароля

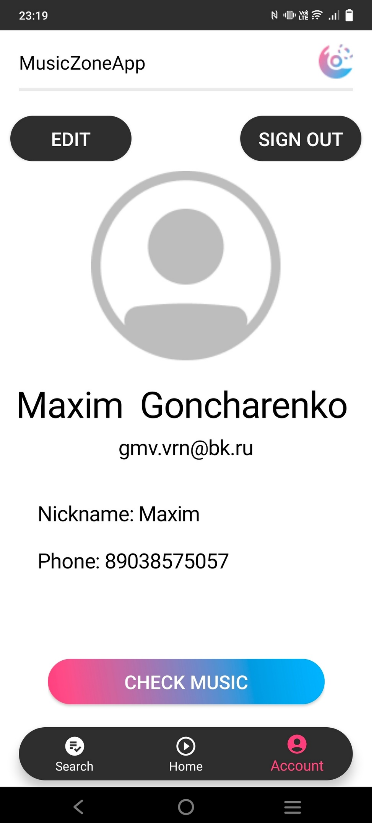


1. Форма экрана ввода нового пароля
   * 1. Форма экрана личной страницы

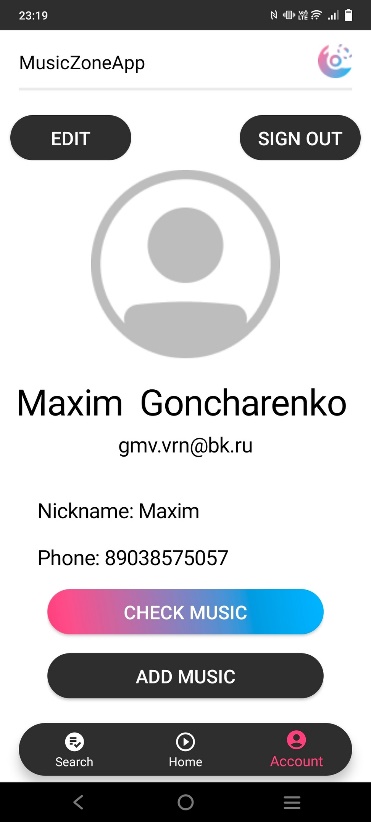
На данном экране указана информация о пользователе (имя, фамилия, псевдоним, почта и номер телефона) рядом расположена кнопка редактирования пользователя и выхода из аккаунта. Ниже находится кнопка для просмотра добавленной музыки.

Снизу расположена панель с кнопками, при нажатии на соответствующую кнопку пользователь переходит на страницы: поиск/главная, прослушивание музыки, личная страница.

Если пользователь является администратором, то на экране появляется кнопка загрузки музыки.

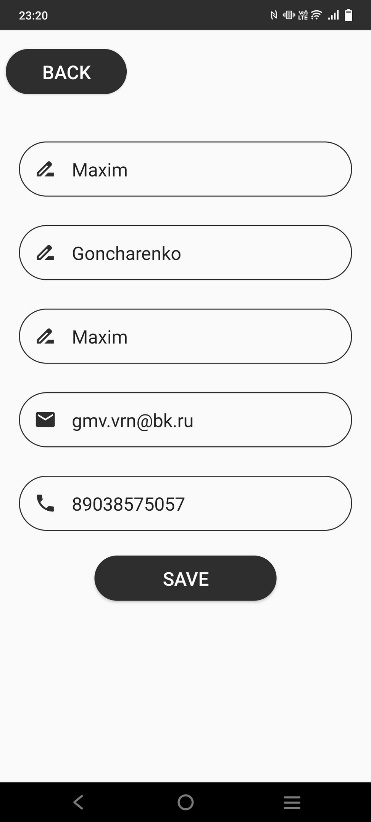


1. Форма экрана личной страницы



1. Форма экрана личной страницы администратора
   * 1. Форма экрана редактирования профиля

На данном экране отображены поля для ввода имени, фамилии, псевдоним, почты, номера телефона, пароль с уже введенными соответствующими данными. Ниже расположена кнопка для сохранения данных.



1. Форма экрана редактирования профиля
   * 1. Форма экрана загруженной музыки

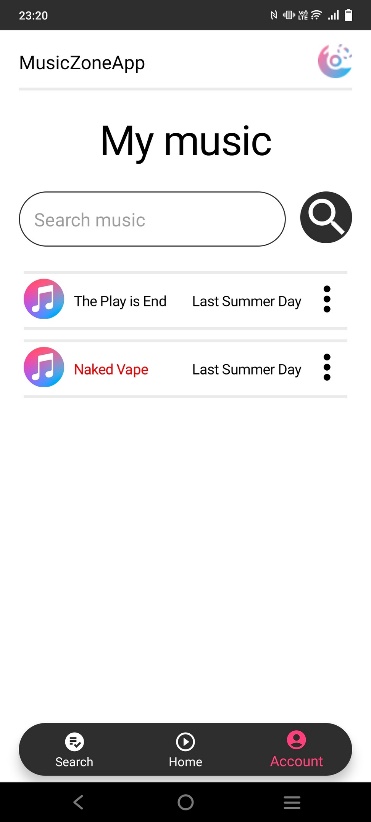
На данном экране расположены список загруженной музыки со стороны администратора приложения, и кнопка для добавления музыки.

Снизу расположена панель с кнопками, при нажатии на соответствующую кнопку пользователь переходит на страницы: поиск/главная, прослушивание музыки, личная страница.



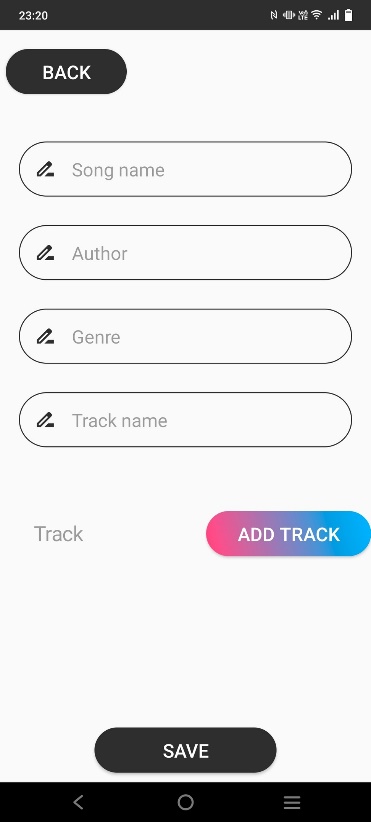
1. Форма экрана загрузки музыки
   * 1. Макет экрана добавленной музыки

На этой странице отображается добавленная музыка пользователем. Снизу расположена панель с кнопками, при нажатии на соответствующую кнопку пользователь переходит на страницы: поиск/главная, прослушивание музыки, личная страница.



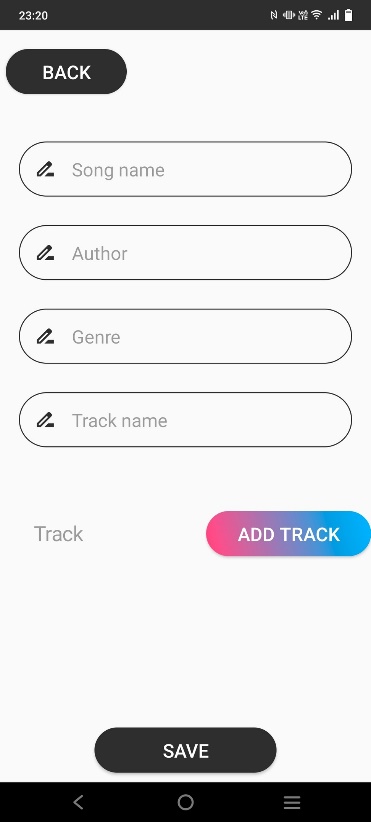
1. Форма экрана пользовательской музыки
   * 1. Форма экрана добавления музыки

На данном экране расположены поля для заполнения названия трека, автор, жанр музыки и загрузить музыкальный файл, кнопки для сохранения и выхода.



1. Форма экрана добавления музыки
   * 1. Форма экрана редактирования музыки

На экране расположены поля уже заполненными соответствующими данными для редактирования (название музыки, автор, жанр музыки) и загруженный музыкальный файл. В нижней части находятся кнопки для сохранения и выхода.



1. Макет экрана редактирования музыки
   1. Серверная часть

В качестве языка был выбран строго типизированный объектно-ориентированный язык программирования Java. Он остается очень популярным языком программирования в этой мобильной разработки благодаря своим мощным возможностям и широкому спектру инструментов для разработки. К тому же существует огромное количество фреймворков и библиотек, написанных на Java, которые в перспективе можно легко интегрировать в проект.

Серверная часть проекта реализована с помощью фремворка Spring Boot и имеют архитектуру, основанную на модели MVC (Model-View-Controller). Контроллеры обрабатывают входящие запросы и отправляют ответы клиенту. Они являются интерфейсом между клиентом и бизнес-логикой приложения. Сервисы предоставляют бизнес-логику приложения. Репозитории отвечают за взаимодействие с базой данных.

Все эти компоненты взаимодействуют друг с другом, чтобы обеспечить функциональность приложения. Контроллеры получают запросы от клиента, направляют их в сервисы, которые используют репозитории для доступа к данным в базе данных, а затем возвращают ответы в контроллеры, которые отправляют их клиенту.

Также применяется программная платформа Docker для быстрой разработки, тестирования и развертывания приложений.

1. Тестирование
2. Продуктовые воронки

Заключение

При разработке мобильного приложения в рамках курсового проекта и анализа приложений для рекомендации музыки, было замечено, что необходимо создать новое приложение, лишенное проблем предыдущих аналогов. С целью решения этой проблемы были определены задачи, которые помогли создать клиент-серверное приложение "Muzic Zone". Оно предназначено для прослушивания музыкальных произведений пользователям.