

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных наук

Кафедра программирования и информационных технологий

КУРСОВОЙ ПРОЕКТ

по теме

«Каталог музыкальных произведений, записанных в электронной форме.

Рекомендации для конкретного пользователя на основе его вкусов».

Зав. кафедрой	_____	С. Д. Махортов, д. ф.-м. н., доцент
Руководитель	_____	В. С. Тарасов, ст. п.
Обучающийся	_____	А. Г. Сапелкин, 3 курс

Воронеж, 2022

Содержание

Введение	4
1. Анализ предметной области	5
1.1. Терминология	5
1.2. Обзор аналогов	6
1.2.1. Spotify	7
1.2.2. VK Music	7
1.2.3. Deezer	8
1.3. Постановка задачи	9
1.3.1. Функциональные требования	9
1.3.2. Технические требования	10
1.3.3. Требования к интерфейсу	10
2. Реализация	11
2.1. Средства реализации	11
2.2. Архитектура приложения.....	13
2.2.1. База данных	13
2.2.2. Диаграмма прецедентов	16
2.2.3. Диаграммы классов.....	17
2.2.4. Диаграммы последовательности	18
2.2.5. IDEF-0 диаграмма	25
2.2.6. Диаграмма развертывания	26
2.2.7. Диаграмма состояний	27
2.3. Серверная часть.....	27

2.3.1. Основные положения	28
2.3.2. Реализация рекомендаций по предпочтениям	28
2.3.3. Развертка	29
2.3.4. Документация API	29
2.4. Клиентская часть	29
2.4.1. Основные положения	29
2.4.2. Графический пользовательский интерфейс	30
2.5. Аналитика	33
2.5.1. Метрики	33
2.6. Тестирование	35
Заключение	40
Список использованных источников	41

Введение

С развитием компьютерных технологий оцифровке подвергаются многие события и процессы. Музыканты набирают популярность с помощью распространения своих произведений в сети интернет посредством рекламы, а уже известные исполнители получают прибыль продавая авторские права различным сервисам для прослушивания музыкальных треков.

Для распространения оцифрованных музыкальных произведений исполнители зачастую пользуются сетью Интернет, но, разумеется, используя при этом различные платформы и приложения в качестве посредника, облегчающего работу по распространению.

Одним из удобных способов распространения музыкальных произведений является использование мобильного приложения для смартфона, так как и для пользователей, и для исполнителей, нет необходимости носить с собой громоздкие устройства.

Разумеется, количество распространенной таким образом музыки может быть слишком велико, чтобы пользователи могли быстро находить нужную им. Для повышения эргономичности приложения зачастую создают систему фильтров и рекомендаций по критериям.

Целью данной работы являлась разработка мобильного приложения, предназначенного для подбора и рекомендации новых произведений на основе предпочтений пользователя, которые он также может задать. Это позволит пользователям расширять свой музыкальный кругозор и открывать для себя новых исполнителей, новые жанры и новые композиции, а для исполнителей – распространять свои музыкальные произведения.

1. Анализ предметной области

1.1. Терминология

Термин	Определение
Интерфейс	Набор инструментов, позволяющий пользователю взаимодействовать с системой.
Чарт	Список самых популярных, на данный момент, песен.
Подкаст	Трансляция музыки или речевого контента в интернете по принципу тематической или жанровой радиостанции.
Авторизация	Предоставление определенному лицу прав на выполнение определенных действий.
Сервер	Обслуживающее устройство в системах автоматической обработки информации.
Модуль	Сложный узел, выполняющий самостоятельную функцию в техническом устройстве, а также вообще отделяемая, относительно самостоятельная часть какой-н. системы, организации.
Валидация	Процесс проверки данных различных типов по критериям корректности и полезности для конкретного применения.

Сплэш-скрин	Окно-заставка, которое появляется на экране во время загрузки приложения
Рендеринг	Процесс получения изображения по модели с помощью компьютерной программы.
Внешний ключ	Столбец (или группа столбцов), используемый в реляционной базе данных для связи данных между таблицами.
Главный ключ	Минимальный набор атрибутов, по значениям которых можно однозначно выбрать требуемый экземпляр сущности.
Альтернативный ключ	Атрибут (или группа атрибутов), несовпадающий с первичным ключом и уникально идентифицирующий экземпляр сущности.
Составной ключ	Комбинация двух или более столбцов в таблице, которые можно использовать для уникальной идентификации каждой строки в таблице.
Метрика	Инструмент веб-аналитики, который помогает получать наглядные отчеты, записи действий посетителей, отслеживать источники трафика.

1.2. Обзор аналогов

Начиная разработку нового музыкального приложения, стоит ознакомиться с уже существующими и рассмотреть их достоинства и недостатки. На

основании полученной информации можно сделать выводы о том, какие недочеты стоит взять на вооружение и не допускать в своей работе, чтобы интерфейс для пользователя был удобен, прост в использовании и управлении.

1.2.1. Spotify

Возможности приложения:

- прослушивание любых выбранных и доступных произведений;
- получение личных рекомендаций на основе избранных или уже прослушанных произведений;
- доступ к скачанным композициям в режиме offline;
- доступ к различным чартам и хит-парадам;
- доступ к радиостанциям;

Достоинства интерфейса:

- простой интерфейс;
- доступ с различных платформ;
- удобно в использовании;
- легко делиться музыкой с друзьями;
- большая база музыкальных произведений.

Недостатки:

- обязательная регистрация для доступа к функционалу приложения;
- ограниченные возможности бесплатной версии;
- отсутствуют тексты песен;
- часто всплывающая реклама.

В итоге, можно сказать, что это удобное и интуитивно понятное приложение с широкими возможностями, которое, однако, требует регистрации для доступа к функционалу, а в бесплатной версии ухудшает впечатления пользователя при использовании приложения.

1.2.2. VK Music

Возможности приложения:

- прослушивание любых доступных композиций;

- просмотр и воспроизведение музыки, которая нравится пользователям в списке друзей;
- просмотр текста некоторых композиций;
- доступ к скачанным трекам в режиме offline;
- доступ к редактированию текста композиции при выявлении ошибок;
- рекомендации на основе сохраненных треков;
- доступ к различным подборкам, чартам, плейлистам, сборникам.

Достоинства интерфейса:

- регулярное добавление новинок и обновление чартов;
- наличие возможности создания плейлистов;
- легко делиться музыкой с друзьями.

Недостатки:

- огромное количество повторяющихся композиций;
- часто всплывающая реклама;
- отсутствует возможность сортировки.

Таким образом, приложение обладает приятным интерфейсом и не менее приятным функционалом, однако композиции не отсортированы, а реклама портит общее впечатление о приложении.

1.2.3. Deezer

Возможности приложения:

- воспроизведение любых находящихся в библиотеке треков;
- воспроизведение сохраненных композиций в режиме offline;
- формирование своего плейлиста;
- получение рекомендаций, на основе избранных композиций;
- доступ к составленным плейлистам других пользователей;
- поиск нужных композиций;
- прослушивание интересных подкастов;
- просмотр текстов песен.

Теперь рассмотрим достоинства интерфейса приложения:

- большая музыкальная библиотека;
- наличие подкастов;
- рекомендации не только по трекам, но и по исполнителям, и по плейлистам;
- расширенный поиск, включающий не только поиск по автору и названию, но и по тексту композиции.

Недостатки:

- серьезные ограничения для бесплатной версии;
- обязательная регистрация/авторизация для использования;
- отсутствие возможности редактирования тестов песен;
- отсутствие чартов и хит-парадов, а также радиостанций.

Делая вывод, можно сказать, что у приложения приятный и интуитивный интерфейс, широкие возможности, в том числе рекомендации по разным критериям, однако все это значительно ограничено в бесплатной версии.

1.3. Постановка задачи

Целью данного проекта является разработка Android приложения «Музкат», которое предназначено для составления индивидуального списка музыкальных предпочтений для каждого пользователя, на основании которого можно получить рекомендации музыкальных произведений, что поможет пользователям этого приложения расширить свой кругозор в музыке, найти новые музыкальные произведения.

1.3.1. Функциональные требования

Для определения границ проекта и его основных задач были сформулированы следующие функциональные требования:

- возможность входа в аккаунт и регистрации неавторизованного пользователя с использованием логина и пароля;
- возможность выхода из аккаунта авторизованного пользователя;
- возможность просмотра информации о музыкальных записях: название, автор и жанр;

- возможность задания авторизированным пользователем предпочитаемых жанров и авторов;
- возможность авторизованного пользователя пополнять список доступных музыкальных записей, состоящих из названия произведения, автора и жанра.
- возможность авторизованного пользователя получения списка музыкальных произведений, с учетом заданных предпочтений.

1.3.2. Технические требования

Не менее важным также является и определение технических требований:

- все запросы клиента серверу и ответы сервера клиенту осуществляются по протоколу HTTP, если явно не указано иное;
- вход в аккаунт или регистрация осуществляется посредством отправки клиента серверу запроса о регистрации или входе с включенными в тело запроса введенного пользователем логина и пароля, где сервер проверяет и одобряет или отклоняет вход или регистрацию, отправляя соответствующие ответы пользователю;
- для хранения данных должна использоваться реляционная база данных;
- приложение должно быть реализовано под Android версии 6 или старше, с использованием языка Java;
- при добавлении новой информации о музыкальных записях или изменениях предпочтений авторизованного пользователя производится отсылка серверу запроса, содержащего всю необходимую информацию для изменения предпочтений или списка музыки;
- ведение статистики использования приложения в соответствии с описанными в следующей главе воронками конверсии.

1.3.3. Требования к интерфейсу

Для ясности намеченных задач и целей были выделены следующие требования к интерфейсу:

- возможность пропуска входа или регистрации неавторизованного пользователя с дальнейшим переходом к списку информации о музыкальных произведениях;
- возможность просмотреть причину неудачного входа в аккаунт или его регистрацию при, соответственно, попытке входа или регистрации;
- при выходе из аккаунта интерфейс обновляется в соответствии с функциональными требованиями, а список музыкальных произведений сбрасывается на клиентской части с последующим запросом нового списка у сервера.

2. Реализация

2.1. Средства реализации

В качестве средств реализации для разработки мобильного приложения использованы:

- Java 11 (клиент и сервер)
- Spring Framework (сервер)
- Spring Boot (сервер)
- Hibernate (сервер)
- Java Persistence API (сервер)
- Lombok (сервер)
- XML при создании экранов приложения и стилей (клиент)
- IntelliJ IDEA (сервер)
- Android Studio (клиент)

Перед очевидным аналогом – Kotlin – Java как язык программирования для android-приложений обладает преимуществом в размере сообщества разработчиков и информации по разработке приложений на этом языке, так как является более старым, что позволяет быстрее реализовывать решения и находить проблемы.

Языком для сервера была выбрана Java в связи с использованием Spring Framework [2], который обладает следующими преимуществами:

- Spring Framework может быть задействован на всех архитектурных слоях, применяемых при разработке клиент-серверных приложений;
- использует модель POJO при написании классов;
- позволяет свободно связывать модули и легко их тестировать;
- поддерживает декларативное программирование;
- избавляет от самостоятельного создания фабричных и синглтон-классов;
- поддерживает различные способы конфигурации;
- предоставляет сервис уровня middleware.

Spring Boot[3] обладает следующими преимуществами:

- не требует развертывания WAR-файлов;
- создает автономные приложения;
- помогает напрямую встроить в приложение Tomcat, Jetty или Undertow;
- не требует XML-конфигурации;
- направлен на уменьшение объема исходного кода;
- имеет дополнительную функциональность «из коробки»;
- простота запуска;
- простая настройка и управление.

Java Persistence API [4] обладает следующими преимуществами:

- отделяет SQL логику от бизнес-логики, что позволяет думать более абстрактно и ориентироваться на свои классы, а это, в свою очередь, также облегчает сопровождение;
- нет необходимости контролировать количество «?» в SQL-подобных запросах;
- есть совместимость с библиотеками валидации данных.

Главное преимущество Lombok [5] в том, что он значительно сокращает код, делает его более читаемым при помощи аннотаций, добавляя соответствующие им методы, контролируя уровни доступа. Это снимает рутинную работу с разработчика и ускоряет создание приложения.

Преимущество XML-разметки для создания экранов приложения перед ручным созданием элементов для экранов:

- легкая портируемость и мобильность без особой привязки к какому-либо коду;
- более краткая и удобная настройка элементов за счёт тегов и атрибутов;

IntelliJ IDEA [6] была выбрана как среда разработки, так как обладает следующими преимуществами:

- умное автодополнение кода;
- качественная и удобная отладка;
- поддержка системы контроля версий;
- интуитивный интерфейс;
- есть инструменты для безопасного рефакторинга и автогенерации кода.

И, наконец, Android Studio [7] обладает такими преимуществами:

- удобный дизайнер пользовательских интерфейсов, позволяющий облегчить визуальное проектирование приложения;
- XML редактор с подсветкой синтаксиса и умным автодополнением;
- поддержка системы контроля версий;
- поддержка эмуляции устройств;
- возможность проводить тестирование и анализ кода;
- высокая скорость сборки приложения;
- поддержка рендера средствами GPU.

2.2. Архитектура приложения

2.2.1. База данных

В качестве базы данных используется реляционная на основе СУБД [8].

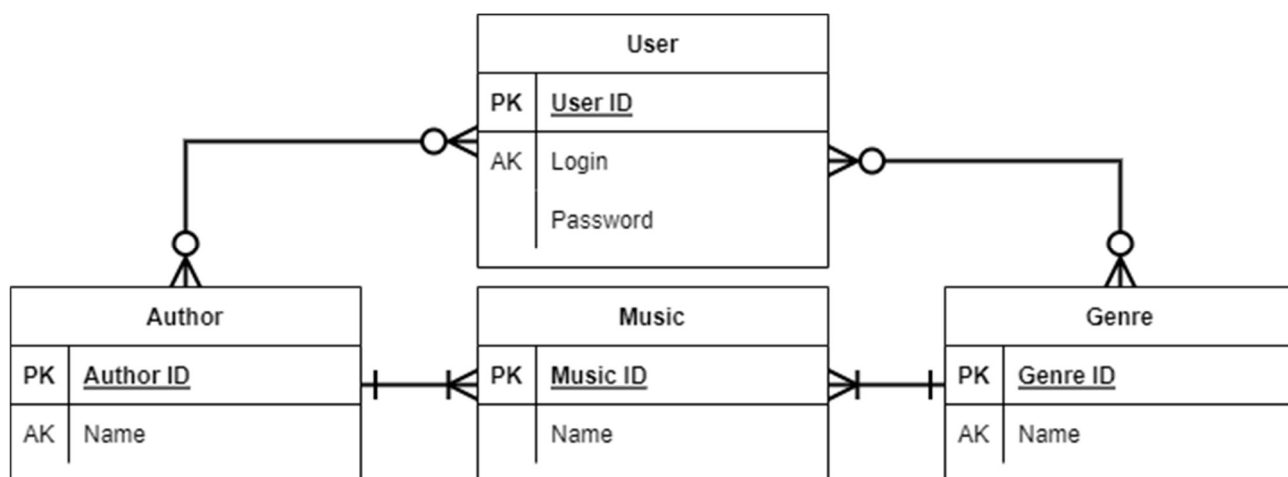


Рисунок 1 — ER-диаграмма базы данных

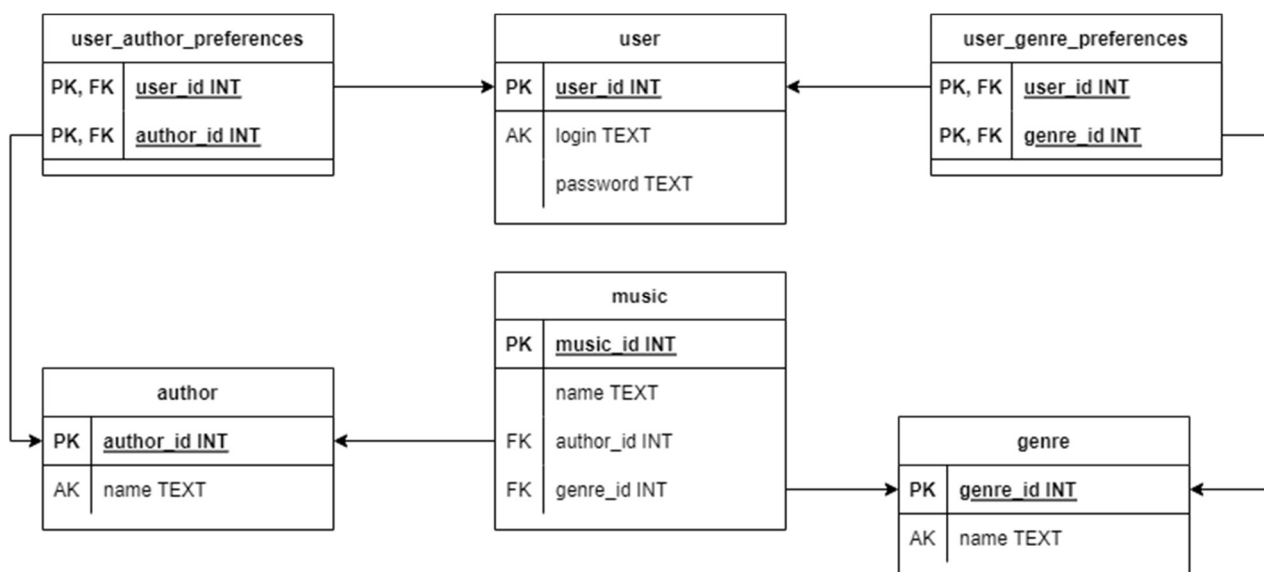


Рисунок 2 — Схема базы данных серверной части приложения.

Таблица «user» предназначена для хранения данных для авторизации пользователей, однажды успешно создавших аккаунт.

Она имеет такие атрибуты как:

- «user_id». Главный ключ. Домен: целые числа. Уникальный идентификатор пользователя. Назначается автоматически при помощи СУБД;
- «login». Альтернативный ключ. Домен: строки. Логин пользователя;
- «password». Домен: строки. Пароль пользователя.

Таблица «genre» предназначена для хранения информации о всех существующих в приложении жанрах.

Она имеет такие атрибуты как:

- «genre_id». Главный ключ. Домен: целые числа. Уникальный идентификатор жанра. Назначается автоматически при помощи СУБД;
- «name». Альтернативный ключ. Домен: строки. Название жанра.

Таблица «author» предназначена для хранения информации о всех существующих в приложении авторах, причем не в буквальном смысле, то есть автором может считаться исполнитель или группа.

Она имеет такие атрибуты как:

- «author_id». Первичный ключ. Домен: целые числа. Уникальный идентификатор автора. Назначается автоматически при помощи СУБД;
- «name». Альтернативный ключ. Домен: строки. Имя автора не в буквальном смысле, то есть здесь может быть фамилия, имя, отчество, псевдоним или их комбинация.

Таблица «music» предназначена для хранения информации о всех музыкальных произведениях, существующих в приложении.

Она имеет такие атрибуты как:

- «music_id». Первичный ключ. Домен: целые числа. Уникальный идентификатор музыкального произведения. Назначается автоматически при помощи СУБД;
- «name». Домен: строки. Название музыкального произведения;
- «author_id». Внешний ключ, ссылающийся на «author_id» таблицы «author». Домен: целые числа. Указывает на уникальный идентификатор автора данного музыкального произведения;
- «genre_id». Внешний ключ, ссылающийся на «genre_id» таблицы «genre». Домен: целые числа. Указывает на уникальный идентификатор жанра данного музыкального произведения.

Таблица «user_author_preferences» нужна для хранения информации о предпочитаемых пользователями авторах.

Она имеет такие атрибуты как:

— «user_id». Первичный, внешний и составной в паре с «author_id» ключ. Домен: целые числа. Уникальный идентификатор пользователя, имеющего в предпочтениях автора с уникальным идентификатором «author_id» данной таблицы;

— «author_id». Первичный, внешний и составной в паре с «user_id» ключ. Домен: целые числа. Уникальный идентификатор автора, которого в качестве одного из предпочитаемых выбрал пользователь с уникальным идентификатором «user_id» данной таблицы.

Таблица «user_genre_preferences» предназначена для хранения информации о предпочитаемых пользователями жанрах.

Она имеет такие атрибуты как:

— «user_id». Первичный, внешний и составной в паре с «genre_id» ключ. Домен: целые числа. Уникальный идентификатор пользователя, имеющего в предпочтениях жанр с уникальным идентификатором «genre_id» данной таблицы;

— «genre_id». Первичный, внешний и составной в паре с «user_id» ключ. Домен: целые числа. Уникальный идентификатор жанра, которого в качестве одного из предпочитаемых выбрал пользователь с уникальным идентификатором «user_id» данной таблицы.

2.2.2. Диаграмма прецедентов

Рассмотрим сначала диаграмма вариантов использования неавторизованного пользователя.

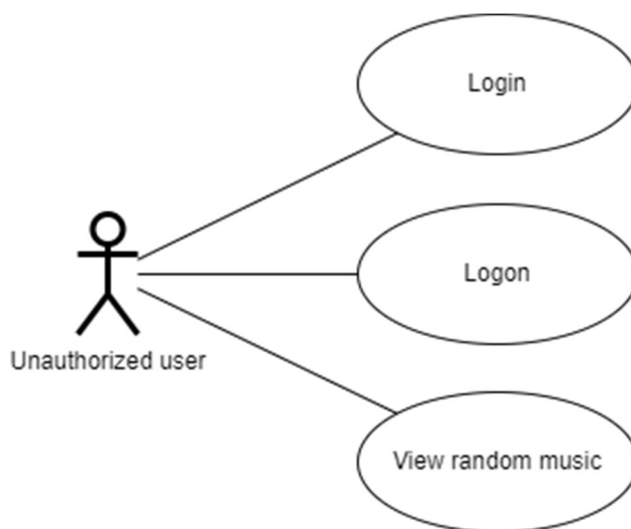


Рисунок 3 — Use-case диаграмма для неавторизованного пользователя

Также рассмотрим диаграмму использования для авторизованного пользователя.

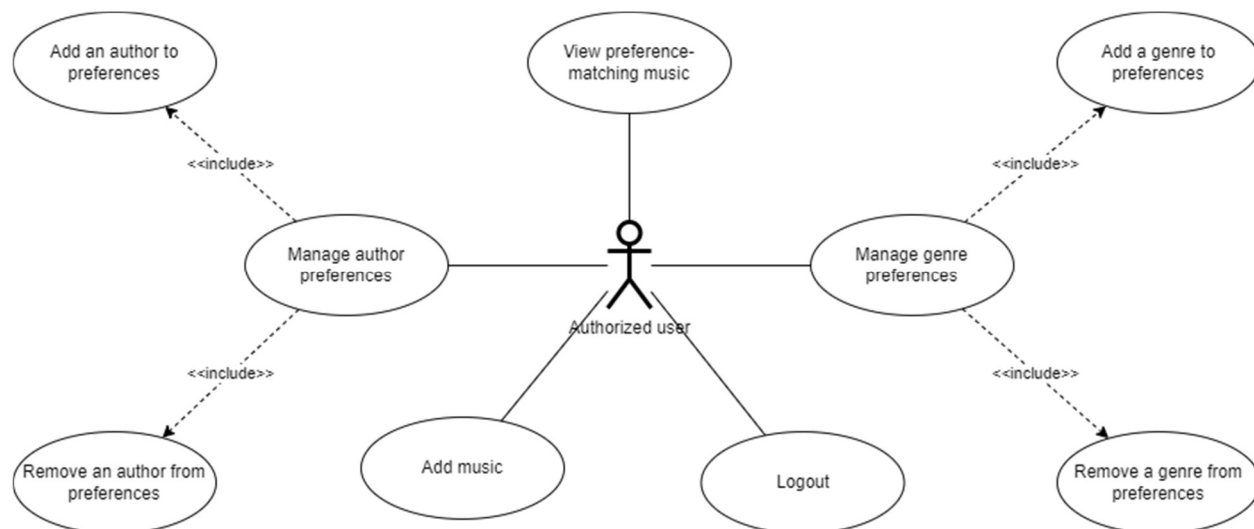


Рисунок 4 — Use-case диаграмма для авторизованного пользователя

2.2.3. Диаграммы классов

В данном пункте будут рассмотрены диаграммы только основных классов серверной части приложения.



Рисунок 5 — Диаграмма классов-сущностей серверной части приложения.

2.2.4. Диаграммы последовательности

Часть диаграмм опущена ввиду их тривиальности. Представлены только диаграммы, представляющие наибольший интерес.

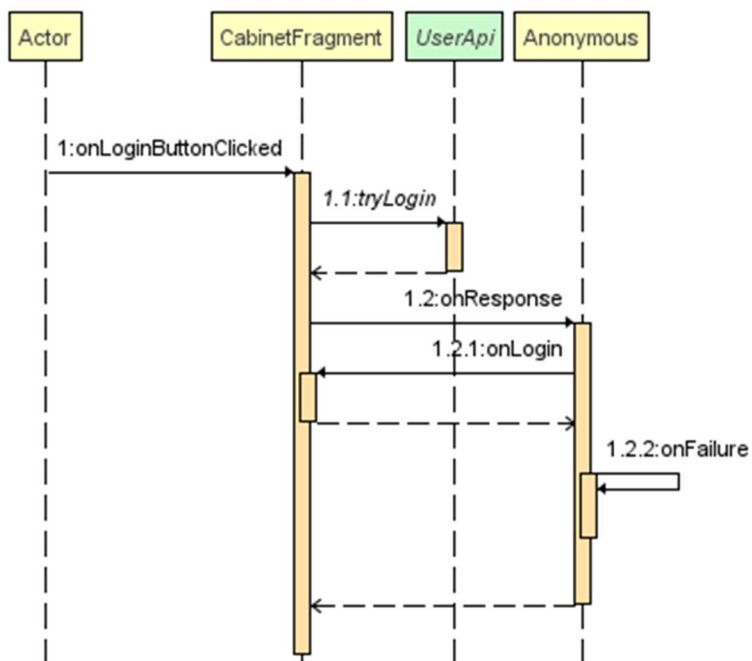


Рисунок 6 — Диаграмма последовательности входа в аккаунт.

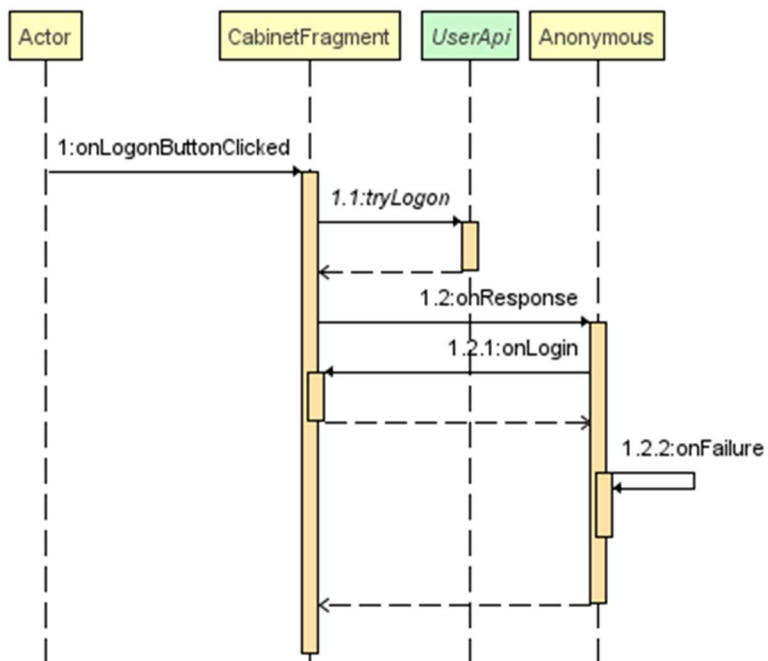


Рисунок 7 — Диаграмма последовательности регистрации.

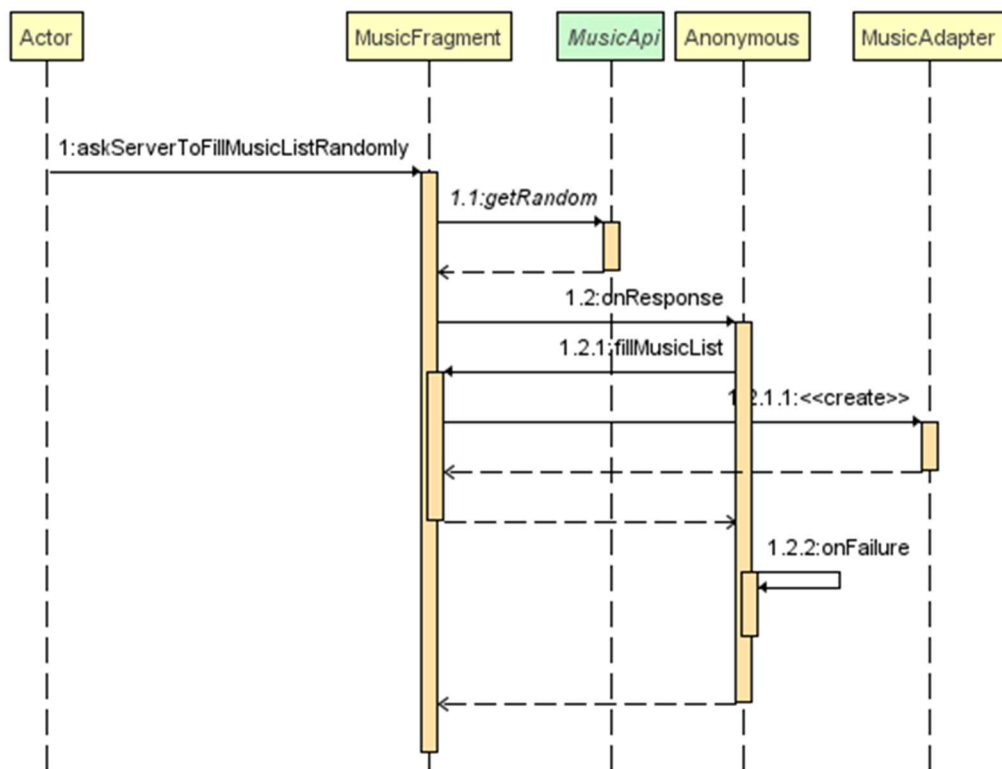


Рисунок 8 — Диаграмма последовательностей получения списка случайной музыки с сервера.

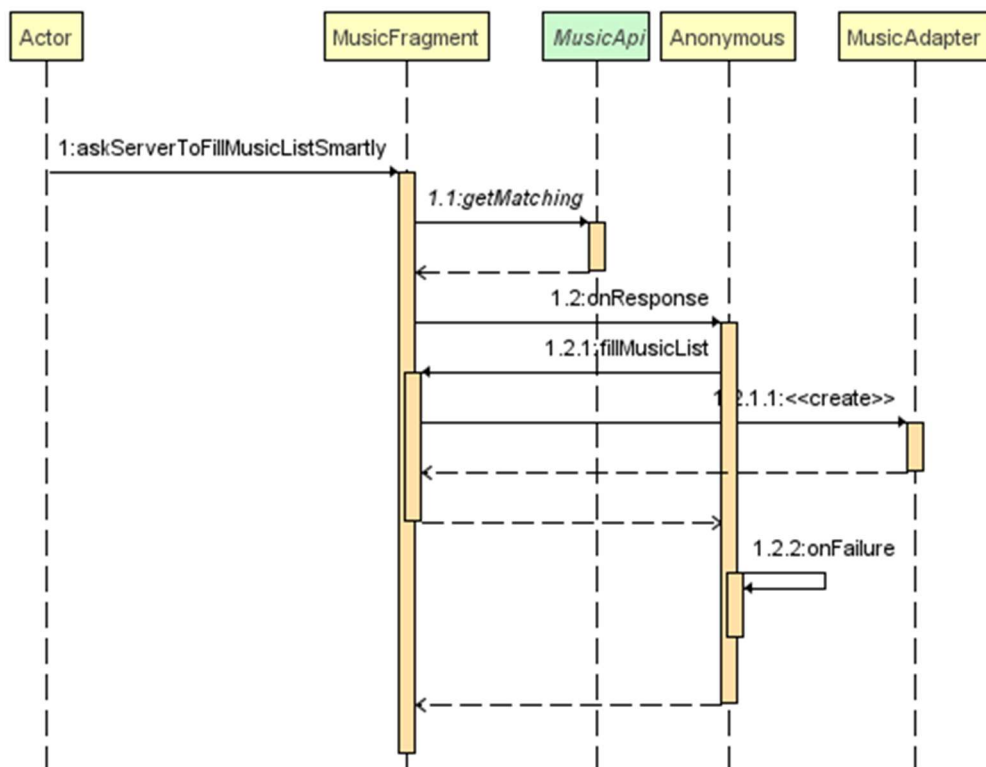


Рисунок 9 — Диаграмма последовательностей получения списка музыки с сервера, в соответствии с предпочтениями пользователей.

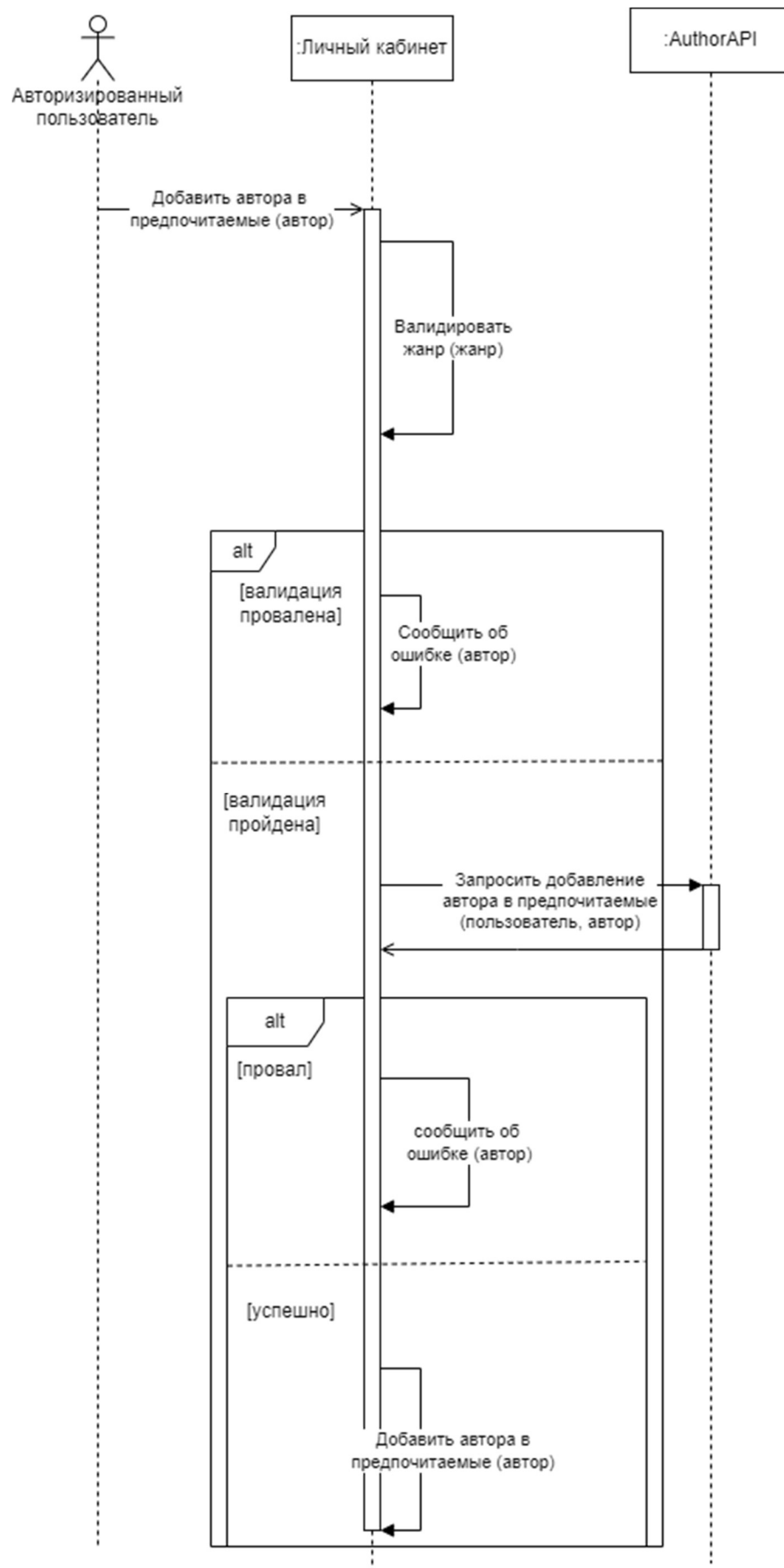


Рисунок 10 — Диаграмма последовательности добавления автора в предпочитаемые

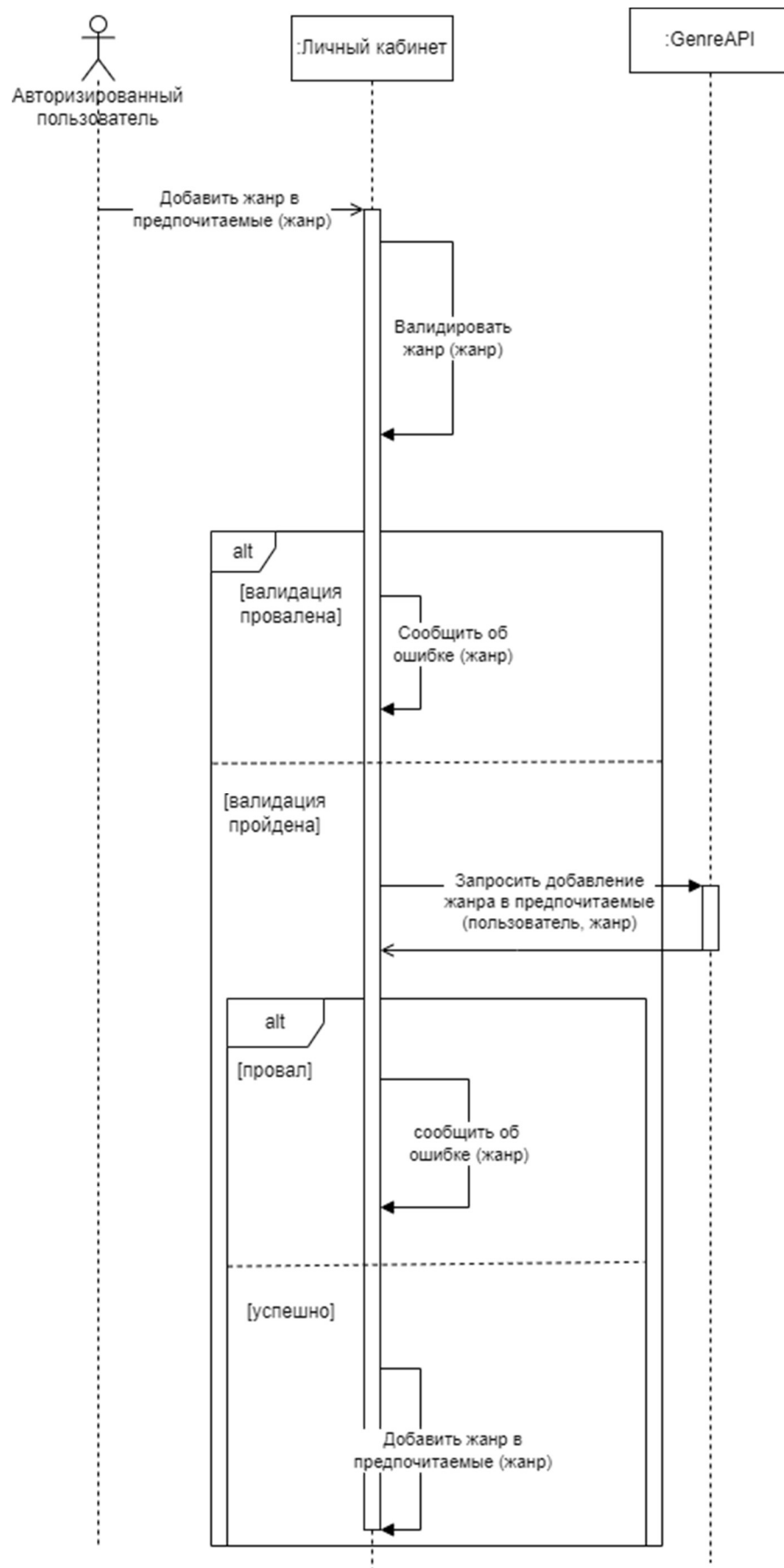


Рисунок 11 — Диаграмма последовательности добавления жанра в предпочитаемые.

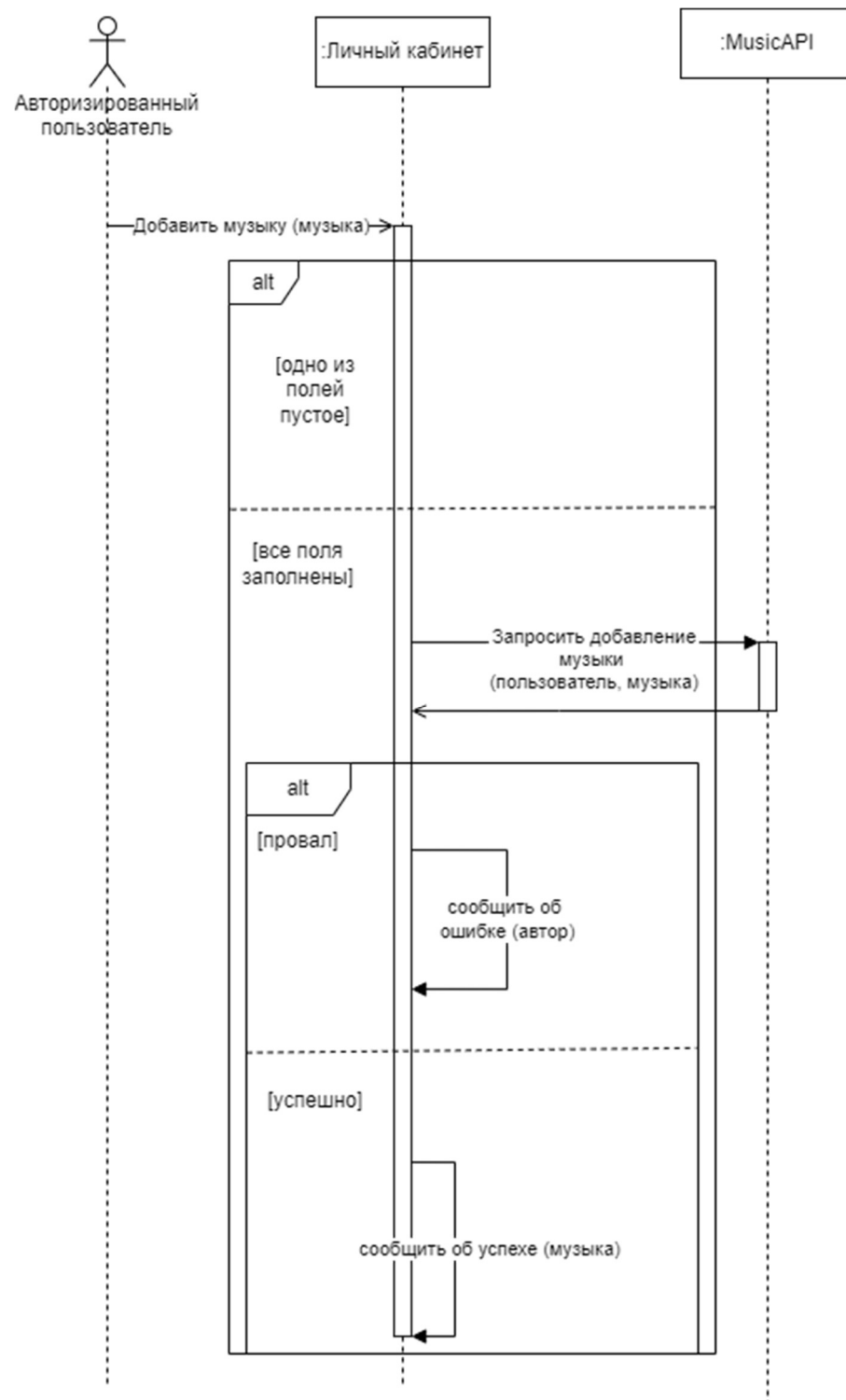


Рисунок 12 — Диаграмма последовательности добавления новой музыки.

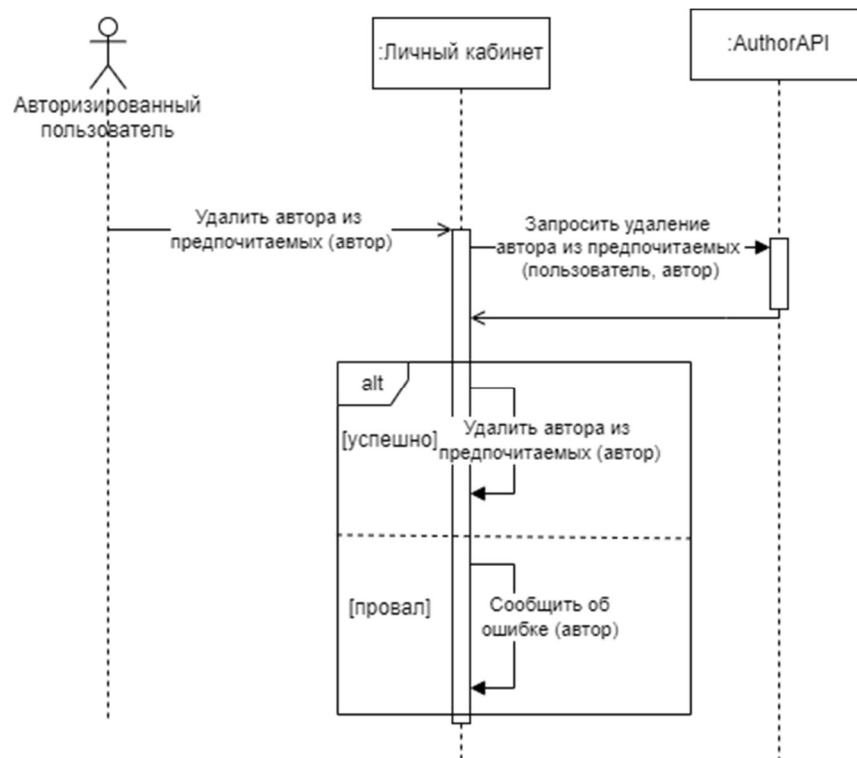


Рисунок 13 — Диаграмма последовательности удаления автора из предпочитаемых.

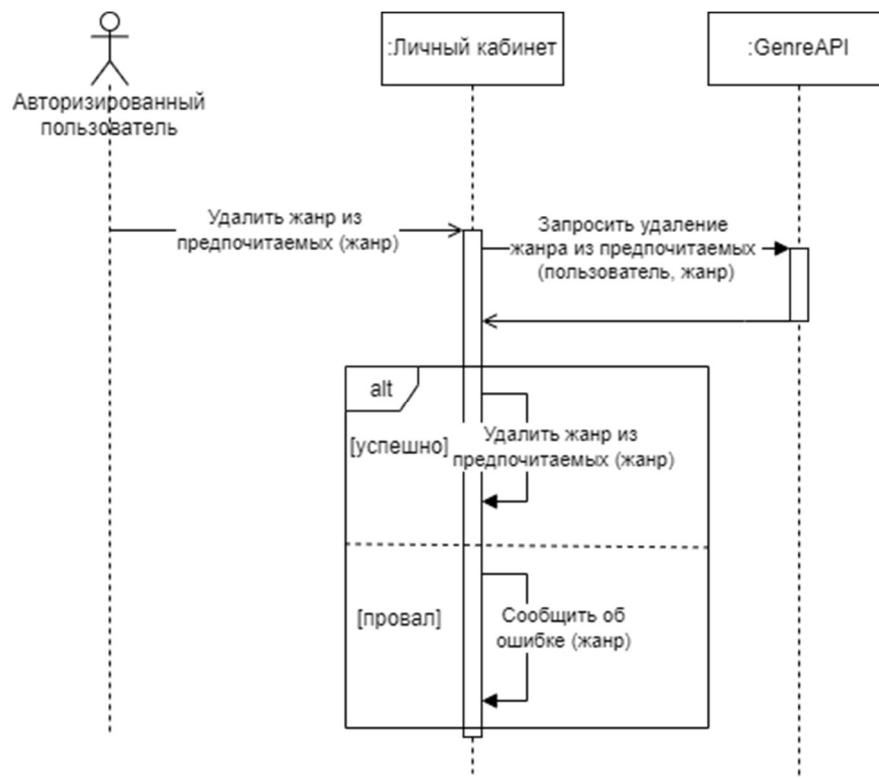


Рисунок 14 — Диаграмма последовательности удаления жанра из предпочитаемых.

2.2.5. IDEF-0 диаграмма

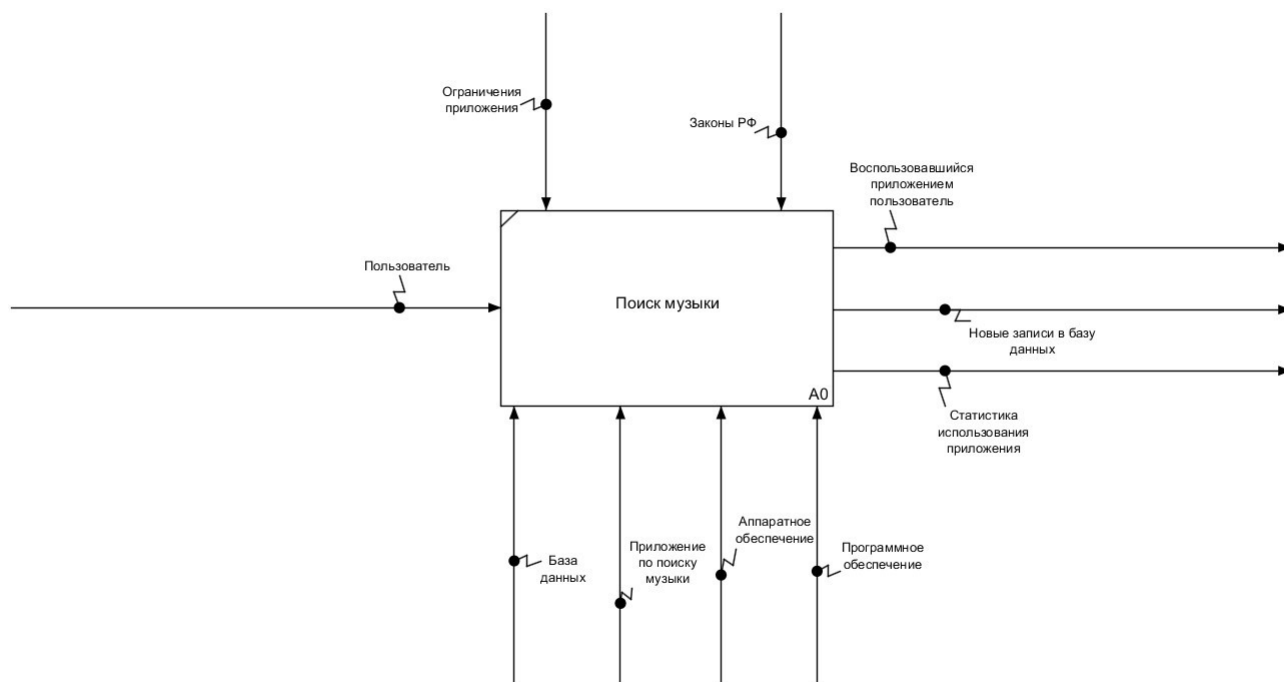


Рисунок 15 — IDEF-0 диаграмма приложения.

2.2.6. Диаграмма развертывания

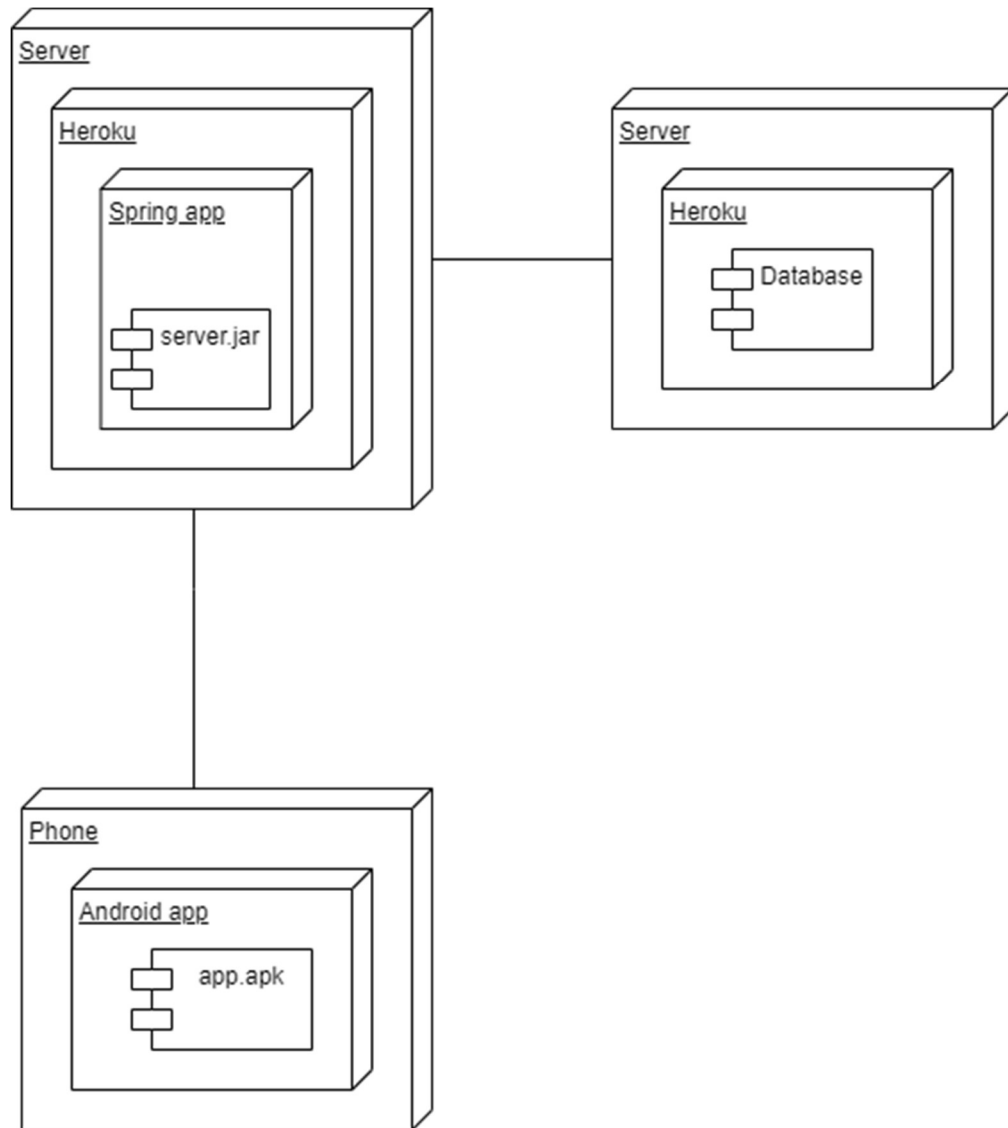


Рисунок 16 — Диаграмма развертывания приложения

2.2.7. Диаграмма состояний

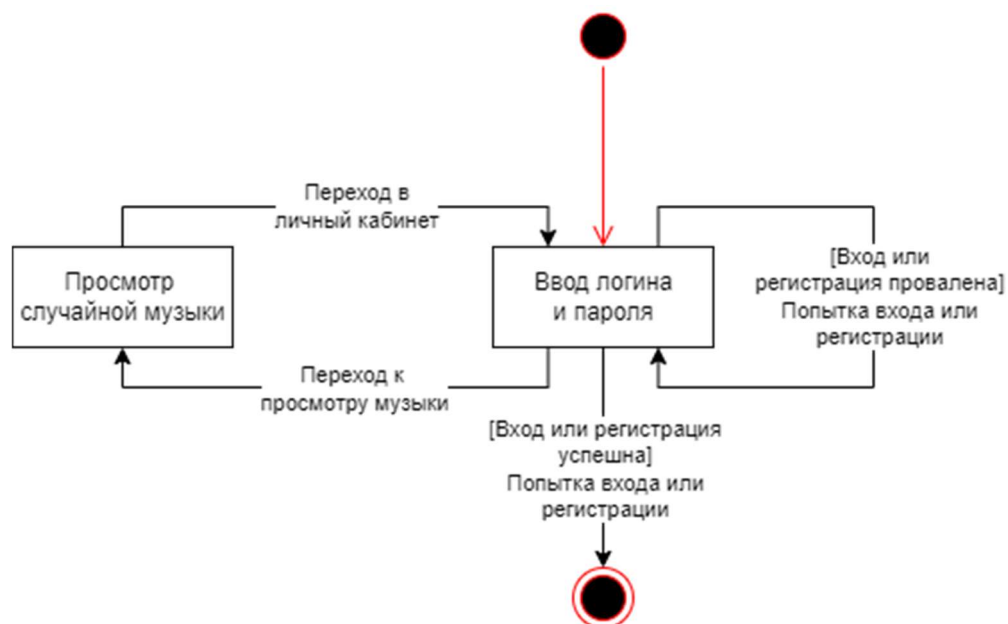


Рисунок 17 — Диаграмма состояний неавторизованного пользователя.

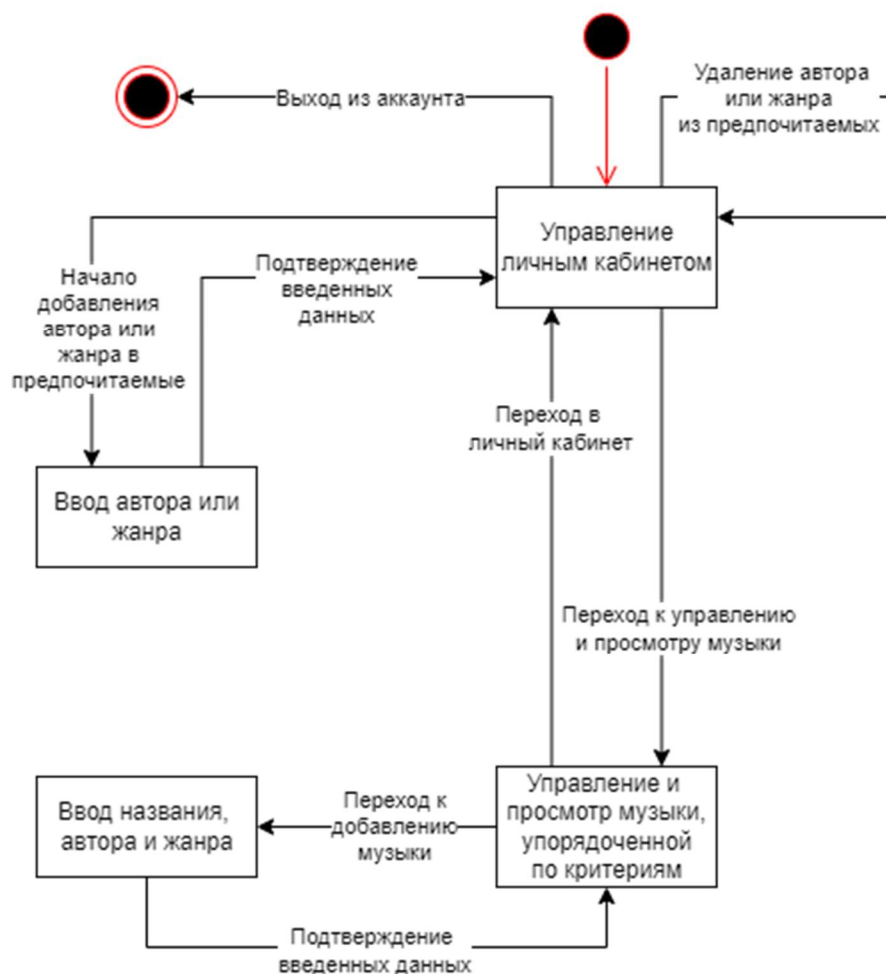


Рисунок 18 — Диаграмма состояний авторизованного пользователя.

2.3. Серверная часть

2.3.1. Основные положения

Основной библиотекой сервера является Java Spring Framework.

Запросы разных типов поступают на соответствующие ранее упомянутой базе данных контроллеры, которые передают обработку бизнес логики сервисам. Сервисы же, в свою очередь, взаимодействуют с репозиториями и сущностями. По окончании работы метода сервиса результат возвращается в контроллер, а далее – в тело-ответ сервера.

2.3.2. Реализация рекомендаций по предпочтениям

Одной из основных задач данного курсового проекта была реализация создания рекомендаций музыкальных произведений, основанная на предпочтениях пользователей по автору и жанру.

Рекомендации по собственным предпочтениям пользователя по автору и жанру были реализованы в одном SQL-запроса к базе данных:

```
SELECT *  
FROM music  
WHERE genre_id IN (:aids) OR author_id IN (:gids)  
ORDER BY (CASE  
    WHEN author_id IN (:aids)  
        AND genre_id IN (:gids) THEN 1  
    ELSE 2  
END)  
LIMIT :amt
```

В места «:aids» и «:gids» будут подставлены соответственно главные ключи предпочитаемых пользователем авторов и жанров. В «:amt» будет передано максимальное количество музыкальных произведений, которое может вернуть этот запрос серверу.

Данный запрос одновременно совершает выборку соответствующих предпочтениям музыкальных произведений, совершает их сортировку в зависимости от степени соответствия и ограничивает количество выдаваемых серверу результатов.

Реализация же рекомендаций по предпочтениям пользователей с похожими предпочтениями более сложна и объемна для представления ее в данном параграфе.

Ознакомиться с полным алгоритмом подбора рекомендованной музыки, а также документации, можно, перейдя по ссылке с соответствующей меткой [9].

2.3.3. Развертка

При развертке серверной части приложения использовалась облачная платформа Heroku, включая используемую базу данных. По этим причинам сервер может оказаться недоступным при поступлении запроса, но будет запущен в течение приблизительно одной минуты.

2.3.4. Документация API

Серверная часть имеет несколько точек взаимодействия с клиентами, представленными различными запросами на различные URL.

Для подробного описания таких точек использовалась ранее упомянутая библиотека Swagger.

Документация же развернута на облачной платформе Heroku и доступна в используемых источниках с соответствующей меткой [10].

2.4. Клиентская часть

2.4.1. Основные положения

В качестве клиента выступает запущенное приложение на мобильном устройстве с операционной системой Android шестой версии или старше. Запросы отправляются на сервер при помощи ранее описанной библиотеки Retrofit.

Основу интерфейса составляют два фрагмента – личный кабинет и список музыки. Остальные экраны, такие как экран добавления музыки, предпочитаемого автора или жанра, представлены отдельной активностью, то есть Activity.

Списки музыки, предпочитаемых авторов и жанров представлены списком RecyclerView, дающим возможность описать каждый элемент списка со своей

логикой. В списке музыки, в частности, присутствует плавающая кнопка `FloatingActionButton`, по нажатию которой открывается активность добавления новой музыкальной записи. Для предпочитаемых авторов и жанров же используется обычная кнопка `Button` после соответствующих списков.

Приложение способно запоминать логин и пароль пользователей при входе в аккаунт. При перезапуске приложения, сохраненные в `SharedPreferences` логин и пароль, будут загружены и применены при запуске приложения автоматически.

2.4.2. Графический пользовательский интерфейс

При входе в приложение пользователь, как правило, видит сплэш-скрин, который исчезает в момент готовности приложения к работе.



Рисунок 19 — Сплэш-скрин

После чего пользователь попадает на экран авторизации. Если возможно, авторизация происходит автоматически.

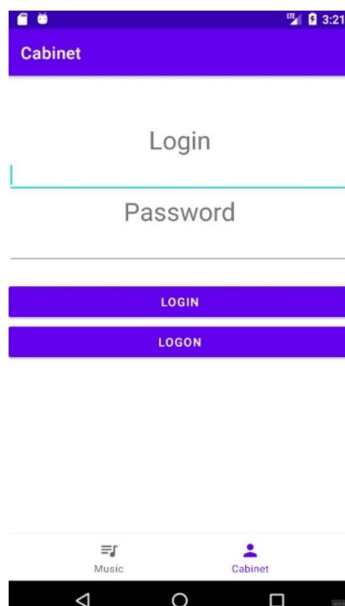


Рисунок 20 — Экран авторизации

Пользователь может пропустить авторизацию и перейти сразу в список со случайными музыкальными произведениями.

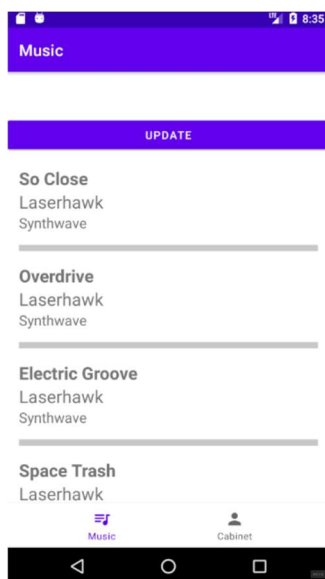


Рисунок 21 — Экран просмотра случайной музыки

Авторизовавшийся пользователь оказывается в личном кабинете, где может настроить предпочтения.

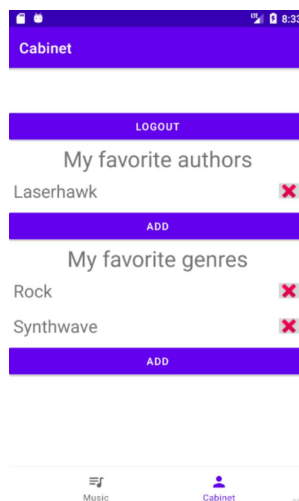


Рисунок 22 — Экран личного кабинета с настройками предпочтений

Пользователь может настроить предпочтения путем нажатия крестика напротив нужного предпочтения или нажав «Add».

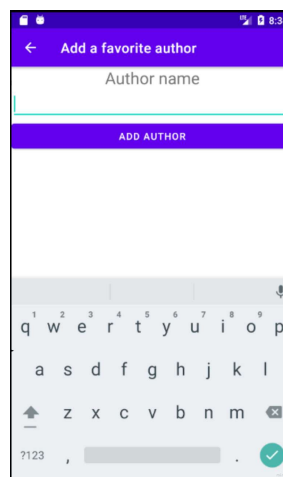


Рисунок 23 — Экран добавления предпочтения

Авторизовавшийся пользователь также может перейти в список с музыкальными произведениями, которые будут отсортированы в соответствии с его предпочтениями.

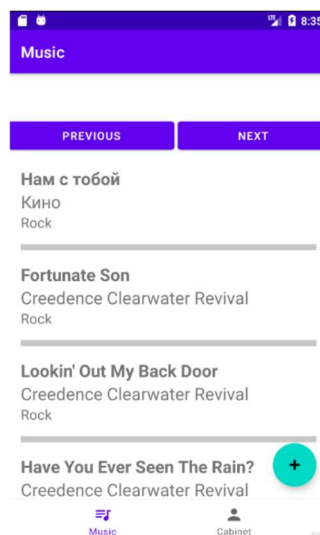


Рисунок 24 — Экран просмотра отсортированной по предпочтениям музыки

Нажав на плавающую кнопку «+» авторизованный пользователь может добавить новое музыкальное произведение.

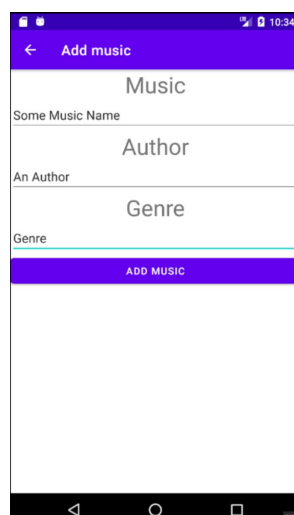


Рисунок 25 — Экран добавления нового музыкального произведения

2.5. Аналитика

2.5.1. Метрики

В качестве системы для сбора данных об использовании приложения пользователями выступает AppMetrica от Яндекс, так как позволяет наиболее быстро и удобно настроить метрики для мобильного приложения, обладает интуитивным интерфейсом и понятным руководством по использованию.

Были составлены три воронки конверсии, первой из которых является воронка «Использование предпочтений», отслеживание шагов которой

позволяет увидеть использование пользователями основной возможности приложения по настройке предпочтений.

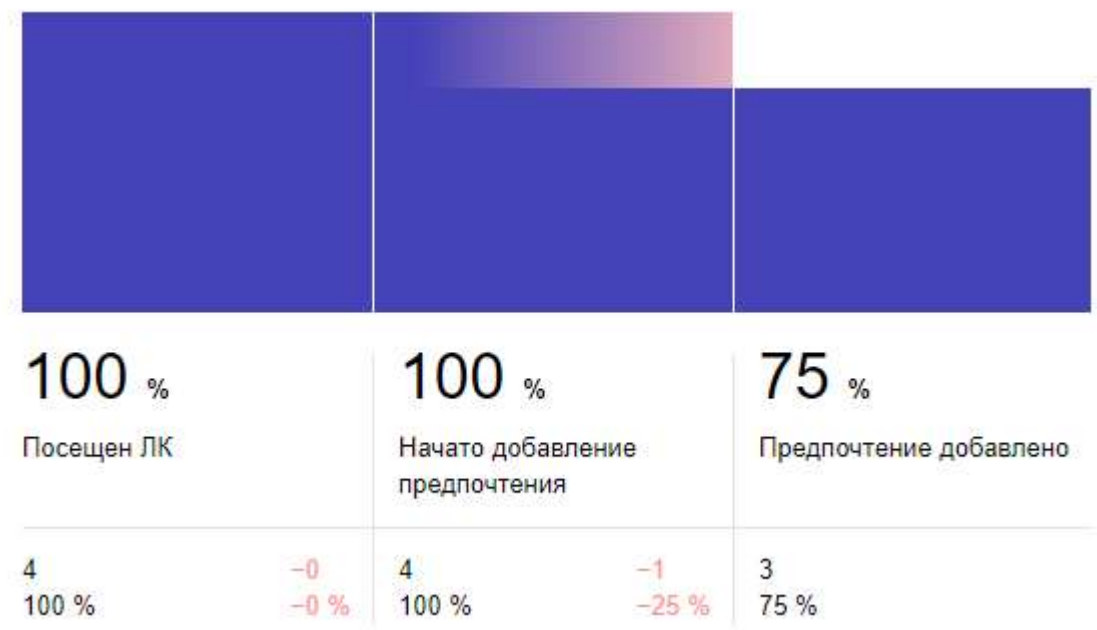


Рисунок 26 — Воронка «Использование предпочтений»

Воронка «Аутентификация» полезна для анализа возникающих проблем аутентификации у пользователей, например неправильно введенный пароль, восстановление которого сейчас не предусмотрено, или же сложности в интерфейсе.

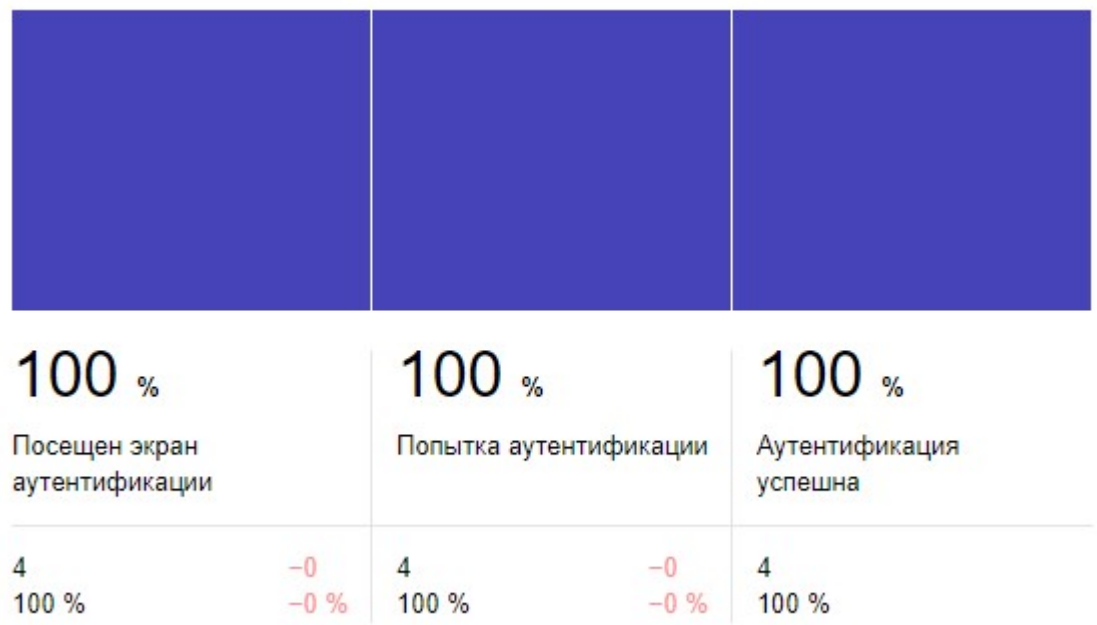


Рисунок 27 — Воронка «Аутентификация»

Последней воронкой является «Добавление музыки», которая рассчитана на прослеживание динамики пополнения списка с музыкой.

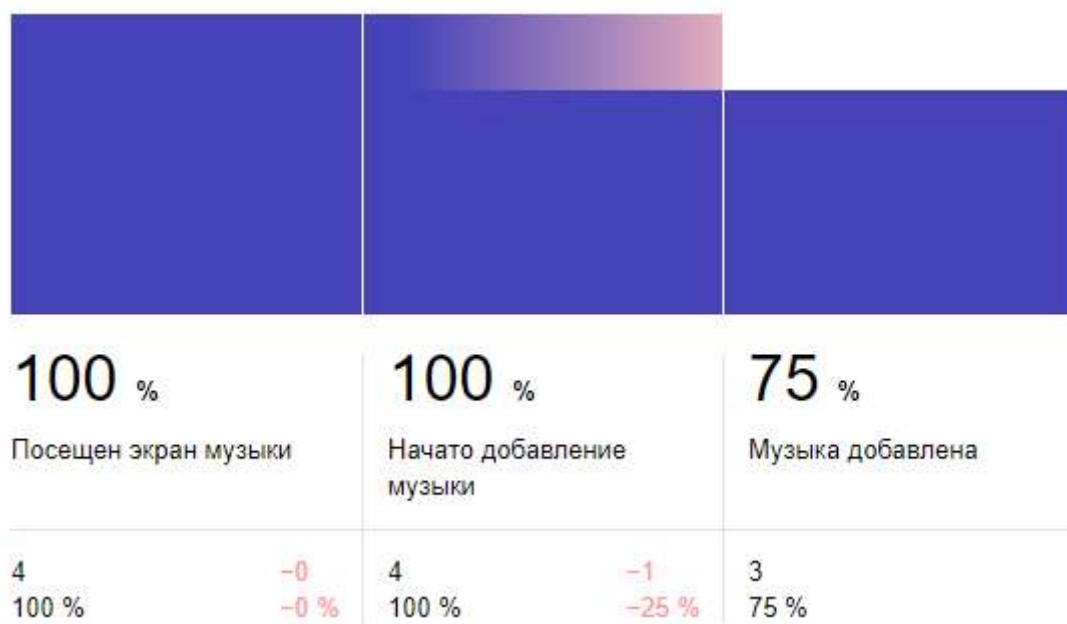


Рисунок 28 — Воронка «Добавление музыки»

2.6. Тестирование

Тестирование приложения осуществлялось при помощи ручного тестирования по заранее составленным тест-кейсам.

Шаг	Ожидаемый результат
Ввести заведомо неправильный логин и пароль	Введенные данные отображаются в полях ввода
Нажать кнопку входа	Кнопка входа пропадает и заменяется на круг загрузки
Дождаться результата входа	Появление всплывающего текста (тоста) на экране, сообщающего о провале попытки входа

Рисунок 29 — Тест-кейс «Поведение системы при провале авторизации»

Шаг	Ожидаемый результат
Ввести заранее полученный логин и пароль	Введенные данные отображаются в полях ввода
Нажать кнопку входа	Кнопка входа заменяется на колесо загрузки
Дождаться результата входа	Экран авторизации заменяется на экран личного кабинета с предпочтениями

Рисунок 30 — Тест-кейс «Вход в аккаунт»

Шаг	Ожидаемый результат
Повторить шаги «Входа в аккаунт»	Экран авторизации заменяется на экран личного кабинета с предпочтениями
Перейти на экран просмотра рекомендованной музыки	Экран просмотра случайной музыки заменен на экран просмотра рекомендованной музыки
Вернуться в личный кабинет	Экран авторизации все еще заменен на экран личного кабинета
Нажать на кнопку выхода из аккаунта	Экран личного кабинета заменяется на экран авторизации
Перейти на экран просмотра случайной музыки	Экран просмотра рекомендованной музыки был заменен на экран просмотра случайной музыки

Рисунок 31 — Тест-кейс «Выход из аккаунта»

Шаг	Ожидаемый результат
Перейти на экран просмотра случайной музыки	Экран просмотра случайной музыки содержит записи без возможности добавления новой

Нажать на кнопку «Update»	Музыкальные произведения перемешаны в случайном порядке или появились новые музыкальные записи на экране
---------------------------	--

Рисунок 32 — Тест-кейс «Просмотр случайной музыки»

Шаг	Ожидаемый результат
Повторить шаги «Входа в аккаунт»	Экран авторизации заменен на экран личного кабинета
Нажать на кнопку «Add» в предпочитаемых авторах	Переход на экран добавления предпочитаемого автора
Ввести существующего и не добавленного в предпочтения автора	Введенные данные отображаются в поле ввода
Нажать кнопку добавления предпочитаемого автора	Кнопка добавления предпочитаемого автора заменена на колесо загрузки
Дождаться результата попытки добавления предпочитаемого автора	Попадание на экран личного кабинета с добавленным на предыдущих шагах автором
Повторить шаги данного тест-кейса для предпочитаемого жанра, за исключением последнего шага	Результат аналогичен полученному при тестировании добавления предпочитаемого автора, за исключением того, что был добавлен предпочитаемый жанр

Рисунок 33 — Тест-кейс «Добавление предпочтений»

Шаг	Ожидаемый результат
Повторить шаги «Входа в аккаунт»	Экран авторизации заменен на экран личного кабинета
Нажать на кнопку «Add» предпочитаемых авторов	Попадание на экран добавления предпочитаемого автора

Ввести не существующего автора в поле для ввода	Введенные данные отображаются в поле для ввода
Нажать на кнопку добавления предпочитаемого автора	Кнопка добавления предпочитаемого автора заменена на колесо загрузки
Дождаться результата попытки добавления предпочитаемого автора	Получение ошибки добавления предпочитаемого автора, попадание на экран личного кабинета
Повторить шаги данного тест-кейса для предпочитаемого жанра, за исключением последнего шага	Результат аналогичен полученному при тестировании добавления предпочитаемого автора, за исключением того, что была получена ошибка, связанная с жанром

Рисунок 34 — Тест-кейс «Поведение системы при провальной попытке добавления предпочтений»

Шаг	Ожидаемый результат
Повторить шаги «Входа в аккаунт»	Экран авторизации заменен на экран личного кабинета
Нажать на красный крестик напротив случайного автора	Крестик заменяется на колесо загрузки
Дождаться результата попытки удаления предпочитаемого автора	Предпочитаемый автор больше не существует в списке предпочитаемых авторов
Повторить шаги данного тест-кейса для предпочитаемого жанра, за исключением последнего шага	Результат аналогичен полученному при тестировании удаления предпочитаемого автора, за исключением того, что был убран предпочитаемый жанр

Рисунок 35 — Тест-кейс «Удаление предпочтений»

Шаг	Ожидаемый результат
-----	---------------------

Повторить шаги «Входа в аккаунт»	Экран авторизации заменен на экран личного кабинета
Перейти на экран просмотра рекомендованной музыки	Экран просмотра рекомендованной музыки содержит музыкальные записи, отсортированные в порядке соответствия добавленным предпочтениям согласно техническому заданию

Рисунок 36 — Тест-кейс «Просмотр рекомендованной музыки»

Шаг	Ожидаемый результат
Повторить шаги «Входа в аккаунт»	Экран авторизации заменен на экран личного кабинета
Перейти на экран со списком рекомендованных музыкальных произведений	Экран случайной музыки был заменен на экран рекомендованной музыки с возможностью добавления новых музыкальных произведений
Нажать на плавающую кнопку «+»	Переход на экран добавления нового музыкального произведения
Ввести название музыки, автора и жанр	Введенные данные отображаются в полях для ввода
Нажать на кнопку добавления новой музыки	Кнопка добавления музыкального произведения заменяется на колесо загрузки
Дождаться результата попытки добавления музыкального произведения	Переход на экран просмотра рекомендованной музыки
При помощи кнопок «Next» и «Previous» отыскать добавленную музыку	Музыкальное произведение найдено и отображено в списке музыкальных произведений

Рисунок 37 — Тест-кейс «Добавление музыки»

Заключение

В ходе разработки данного мобильного приложения в рамках курсового проекта, и, в частности, в ходе анализа приложений по рекомендации музыки, была выявлена потребность в создании нового приложения, лишенного недостатков проанализированных аналогов.

Для этого были поставлены задачи, в соответствии с которыми было создано клиент-серверное приложение «Музкат», предназначенное для рекомендации музыкальных произведений пользователям, причем, для зарегистрированных пользователей, рекомендации основываются на заданных ими предпочтениях по автору или жанру, включая рекомендации по жанрам и авторам пользователей, имеющих похожие предпочтения.

Список использованных источников

1. Безопасность баз данных под управлением Oracle [Электронный ресурс]. Режим доступа: URL: https://www.aladdin-rd.ru/company/pressroom/articles/bezopasnost_baz_dannyh_pod_upravleniem_oracle. – Заглавие с экрана. – (Дата обращения 26.05.2022).
2. Java Spring Framework [Электронный ресурс]. Режим доступа: URL: <https://ask-dev.ru/info/6467/what-are-the-pros-and-cons-of-the-assorted-java-web-frameworks>. – Заглавие с экрана. – (Дата обращения 26.05.2022).
3. Преимущества и недостатки использования Spring Boot [Электронный ресурс]. Режим доступа: URL: <https://javarush.ru/groups/posts/3380-kofe-breyk-75-preimujshestva-i-nedostatki-ispoljhzovanija-spring-boot-funkcii-dlja-strok-v-java>. – Заглавие с экрана. – (Дата обращения 26.05.2022).
4. JPA и гибернация [Электронный ресурс]. Режим доступа: URL: <https://javascope.com/introducing-jpa-and-hibernate-4hgsc-00778adc/>. – Заглавие с экрана. – (Дата обращения 26.05.2022).
5. Lombok преимущества и недостатки [Электронный ресурс]. Режим доступа: URL: <https://www.codetd.com/ru/article/6990413>. – Заглавие с экрана. – (Дата обращения 26.05.2022).
6. IntelliJ IDEA [Электронный ресурс]. Режим доступа: URL: <https://habr.com/ru/post/112749/>. – Заглавие с экрана. – (Дата обращения 26.05.2022).
7. Анализ среды разработки мобильных приложений android studio. [Электронный ресурс]. Режим доступа: URL: <https://ilyarm.ru/analiz-sredy-razrabotki-mobilnyh-prilozhenii-android-studio-razrabotka.html>. – Заглавие с экрана. – (Дата обращения 26.05.2022).
8. Чем PostgreSQL лучше других SQL баз данных. [Электронный ресурс]. Режим доступа: URL: <https://habr.com/ru/post/282764/>. – Заглавие с экрана. – (Дата обращения 26.05.2022).

9. GitHub [Электронный ресурс]. Режим доступа: URL: <https://github.com/Lizurt/Muzkat>. – Заглавие с экрана. – (Дата обращения 06.06.2022).
10. Heroku [Электронный ресурс]. Режим доступа: URL: <https://muzkat-server.herokuapp.com/swagger-ui/index.html>. – Заглавие с экрана. – (Дата обращения 06.06.2022).