

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

Факультет компьютерных наук

Кафедра программирования и информационных систем

Разработка стримингового музыкального веб-приложения «Musicman»

Курсовая работа

09.03.04 Программная инженерия

Разработка программного обеспечения

Зав. кафедрой _____ С.Д. Махортов, д. ф.-м. н., профессор

Обучающийся _____ А.Р. Сашина, 3 курс, д/о

Обучающийся _____ З.С. Казмиров, 3 курс, д/о

Обучающийся _____ Е.М. Охрямкин, 3 курс, д/о

Руководитель _____ В.С. Тарасов, старший преподаватель

Воронеж 2023

Содержание

Введение.....	3
1 Постановка задачи.....	4
1.1 Постановка задачи	4
1.2 Средства реализации.....	4
2 Анализ предметной области	6
2.1 Терминология (глоссарий).....	6
2.2 Обзор аналогов	6
2.2.1 Spotify	6
2.2.2 Deezer	7
2.2.3 Яндекс Музыка.....	8
2.2.4 Youtube Music	9
2.3 Анализ поставленной задачи	11
2.3.1 Диаграмма вариантов использования	11
2.3.2 Диаграмма последовательности	12
2.3.3 Диаграмма классов	13
2.3.4 Диаграмма развертывания	15
3 Реализация.....	17
3.1 Реализация серверной части приложения	17
3.2 Реализация клиентской части приложения	18
Заключение	19
Список использованных источников	20

Введение

В современном мире стриминговые музыкальные сервисы являются неотъемлемой частью нашей жизни и получают все большую популярность. Они предоставляют пользователю удобный доступ к миллионам аудиозаписей в любое время и в любом месте, что позволяет наслаждаться музыкой в комфортных условиях.

Цель данной курсовой работы - разработка веб приложения музыкального стримингового сервиса, позволяющего пользователю в удобной форме наслаждаться любимой музыкой, создавать собственные плейлисты и делиться ими с другими пользователями.

В работе будет рассмотрен процесс разработки приложения, включая его проектирование, функциональность, дизайн и архитектуру системы. А также анализ конкретной среды, исследование потребностей и предпочтений пользователей и определение пути улучшения пользовательского опыта.

Для реализации проекта будут использованы современные технологии и инструменты, необходимые для создания качественного веб приложения. Работа будет состоять из нескольких этапов, включая анализ потребностей пользователей, проектирование интерфейса и архитектуры, разработку функциональности и интеграцию с внешними сервисами, а также тестирование и оптимизацию приложения для улучшения его производительности и удобства использования.

Результатом данной работы будет функциональное и удобное веб приложение музыкального стримингового сервиса, удовлетворяющее потребности пользователей и соответствующее современным требованиям в области веб разработки.

1 Постановка задачи

1.1 Постановка задачи

Стриминговый сервис предназначен для прослушивания песен с возможностью настройки приложения под нужды пользователя.

Основными целями создания системы являются:

- возможность регулировать настройки прослушивания песен;
- показ песен пользователю на основе его музыкальных предпочтений;
- возможность рекомендации песни другим пользователям.

1.2 Средства реализации

Система должна состоять из сервера приложения, реляционной базы данных, клиентской части.

Основной используемый стек технологий:

Back-end (серверная часть):

- Java 17;
- Spring Framework [2];
- PostgreSQL [4], Liquibase[5].

Java является кроссплатформенным языком, т.е. для запуска достаточно иметь Java Virtual Machine. Самым популярным фреймворком для Java является Spring. Основным преимуществом является огромное наличие компонент и внутренних библиотек, которые уже реализованы, а значит позволяет быстро и качественно написать код.

В качестве базы данных была выбрана Postgres, так как умеет работать с различными типами данных и позволяет ускорять запросы с помощью индексов. Для управления и применения изменений в базу данных будет использоваться библиотека Liquibase. Основным преимуществом является поддержка написания миграционных файлов в виде yaml или xml файлов.

Front-end (клиентская часть):

- CSS3 + HTML5;

- React.js [3], TypeScript;
- Effector [6];
- Material UI [7].

CSS3 + HTML5 были выбраны потому, что это самый современный стандарт вёрстки и разметки. Он поддерживается большинством браузеров и предоставляет множество новых свойств, упрощающих разработку.

React является популярным фреймворком для разработки клиентской части. Главным его преимуществом является виртуальная объектная модель документа (DOM), занимающая мало места. А значит позволяет быстрее обновлять страницу с изменениями и повышает производительность приложения.

Язык TypeScript был выбран, потому что это компилируемый в JavaScript код, который нивелирует недостатки слабой типизации JavaScript и берёт множество проверок безопасности кода на себя во время трансляции в JavaScript код.

Effector является стейт-менеджером, который хорошо взаимодействует с React-ом и позволяет легко разделять работу с данными по разным хранилищам (декомпозиция). Material UI является библиотекой готовых компонент для React, которые обладают приятным дизайном.

2 Анализ предметной области

2.1 Терминология (гlossарий)

Администратор сайта – специалист, осуществляющий информационную поддержку сайта, управление контентом.

Веб-браузер (браузер) - клиентская программа, поставляемая третьими сторонами и позволяющая просматривать содержимое веб-страниц.

Контент – совокупность информационного наполнения веб-сайта.

Неавторизованный пользователь (гость) – человек, который может авторизоваться в системе, если был зарегистрирован ранее, или пройти регистрацию.

Пользователь – человек, который зарегистрирован в системе и имеет доступ к личному кабинету и основному функционалу системы.

Система администрирования – закрытая от посетителей часть сайта. Управляется администратором.

Эквалайзер – программа, позволяющая регулировать громкость отдельных зон частотного диапазона и выравнивать амплитудно-частотную характеристику звукового сигнала.

Плейлист – подборка аудиоконтента.

2.2 Обзор аналогов

2.2.1 Spotify

Spotify – это один из самых популярных музыкальных стриминговых сервисов в мире. Этот сервис имеет огромную библиотеку с миллионами песен и плейлистов, которые можно слушать как онлайн, так и офлайн. Spotify имеет простой и интуитивно понятный интерфейс, а также множество функций, таких как персонализированные рекомендации, функция обнаружения новой музыки, плейлисты, созданные другими пользователями, и многое другое.

Плюсы:

- Большая библиотека с миллионами песен и плейлистов;
- Простой и удобный интерфейс (см. Рисунок 1);

- Персонализированные рекомендации и функция обнаружения новой музыки;
- Доступно на большинстве устройств, включая смартфоны, компьютеры, телевизоры и т.д.;
- Функция офлайн-воспроизведения песен.

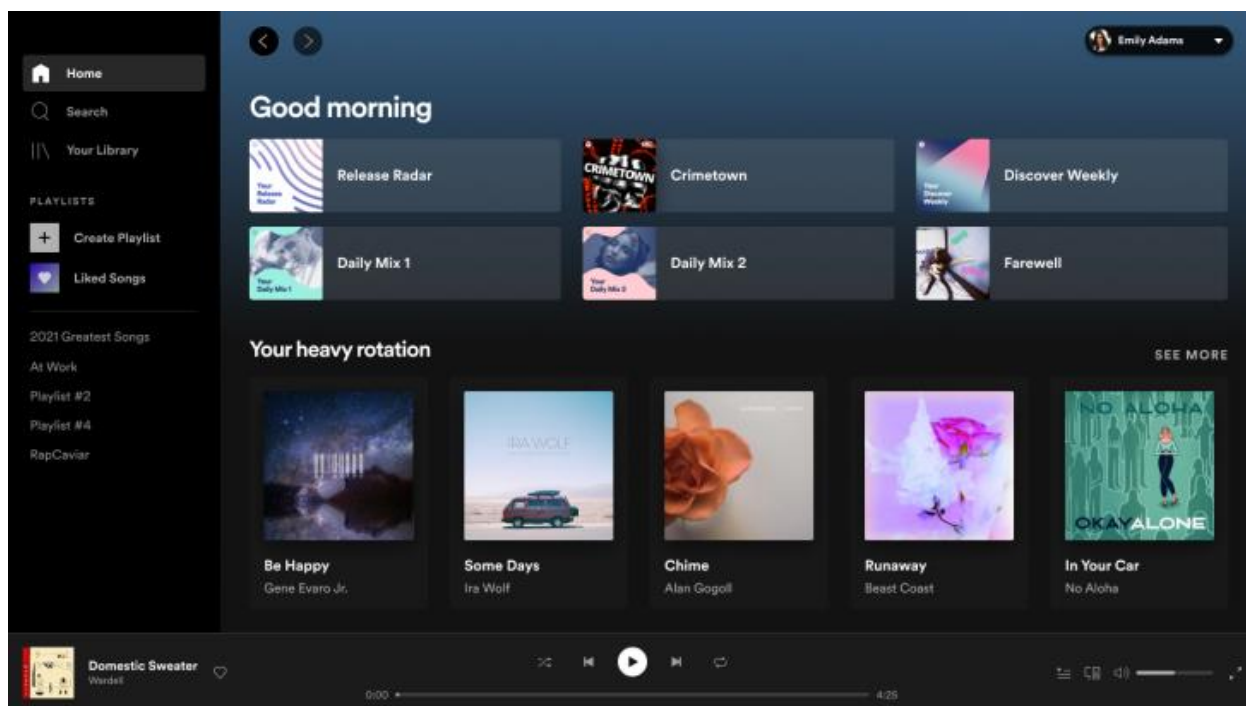


Рисунок 1 - Главная страница Spotify

Минусы:

- Реклама для бесплатной версии;
- Ограниченный доступ к новым альбомам для бесплатной версии;
- Некоторые пользователи могут считать плату за премиум-версию слишком высокой.

2.2.2 Deezer

Deezer – это музыкальный стриминговый сервис, который доступен в более чем 180 странах. Этот сервис также имеет огромную библиотеку с миллионами песен и плейлистов, а также множество функций, таких как подборки песен на основе настроения, автоматическая настройка на любимые исполнители, подкасты и многое другое.

Плюсы:

- Огромная библиотека с миллионами песен и плейлистов;
- Множество функций, таких как подборки песен на основе настроения, автоматическая настройка на любимые исполнители, подкасты и многое другое;
- Доступно на большинстве устройств, включая смартфоны, компьютеры, телевизоры и т.д. (см. Рисунок 2);
- Качество звука выше, чем у конкурентов;
- Функция офлайн-воспроизведения песен.

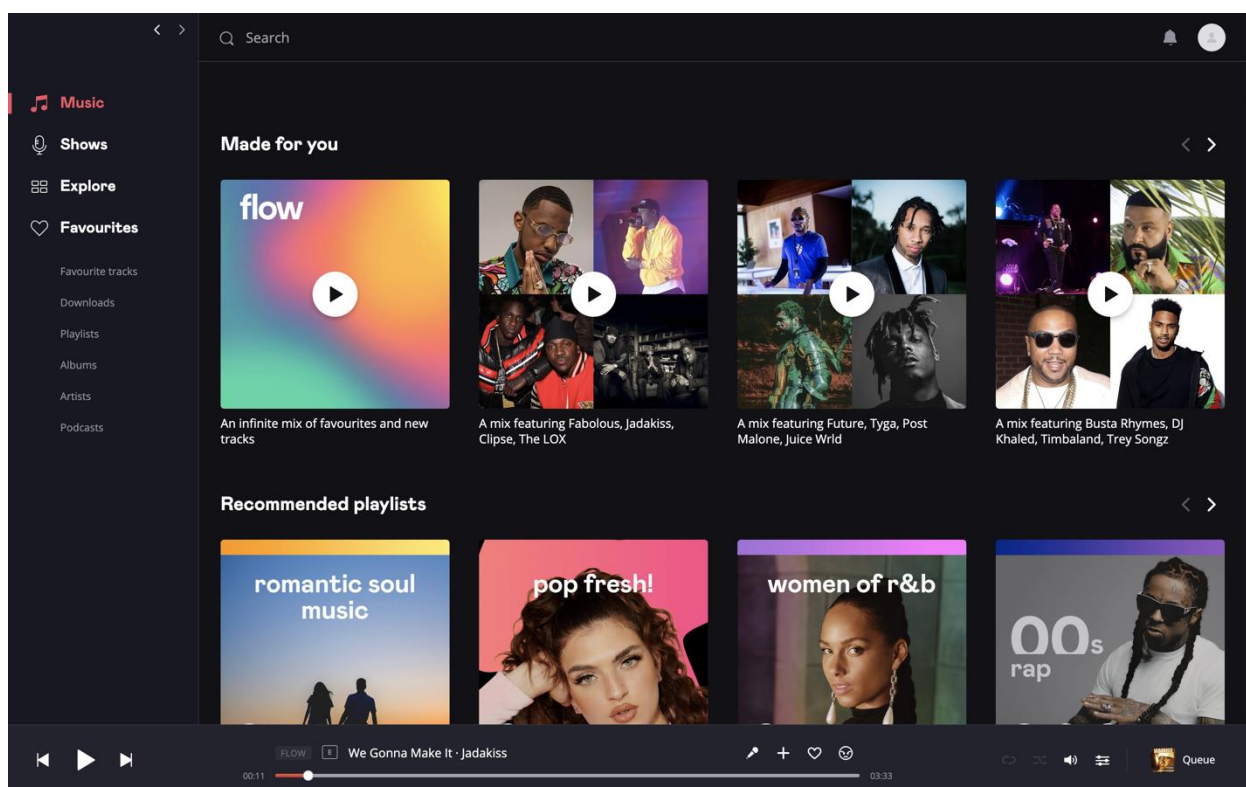


Рисунок 2 - Главная страница Deezer

Минусы:

- Некоторые пользователи могут считать плату за премиум-версию слишком высокой;
- Не все песни доступны во всех странах;
- Реклама для бесплатной версии.

2.2.3 Яндекс Музыка

Яндекс Музыка – это музыкальный стриминговый сервис, разработанный Яндексом. Сервис имеет огромную библиотеку с миллионами песен и плейлистов, а также множество функций, таких как рекомендации на основе настроения, адаптивный плейлист, функция персонализации и многое другое.

Плюсы:

- Большая библиотека с миллионами песен и плейлистов;
- Множество функций, таких как рекомендации на основе настроения, адаптивный плейлист, функция персонализации и многое другое;
- Доступно на большинстве устройств, включая смартфоны, компьютеры, телевизоры и т.д. (см. Рисунок 3);
- Бесплатный доступ к музыке для пользователей Яндекс.Плюс;
- Функция офлайн-воспроизведения песен.

Минусы:

- Некоторые пользователи могут считать интерфейс сложным и запутанным;
- Ограниченный доступ к новым альбомам для бесплатной версии;
- Реклама для бесплатной версии;
- Некоторые пользователи могут испытывать проблемы с качеством звука.

2.2.4 Youtube Music

Youtube Music – это музыкальный стриминговый сервис, разработанный Google. Сервис имеет библиотеку с миллионами песен и плейлистов, а также множество функций, таких как персонализированные рекомендации, офлайн-воспроизведение и многое другое.

Плюсы:

- Большая библиотека с миллионами песен и плейлистов;

- Множество функций, таких как персонализированные рекомендации, офлайн-воспроизведение и многое другое;
- Доступно на большинстве устройств, включая смартфоны, компьютеры, телевизоры и т.д. (см. Рисунок 4);
- Бесплатный доступ к музыке для пользователей Youtube Premium;
- Интеграция с другими сервисами Google, такими как Google Assistant и Google Home.

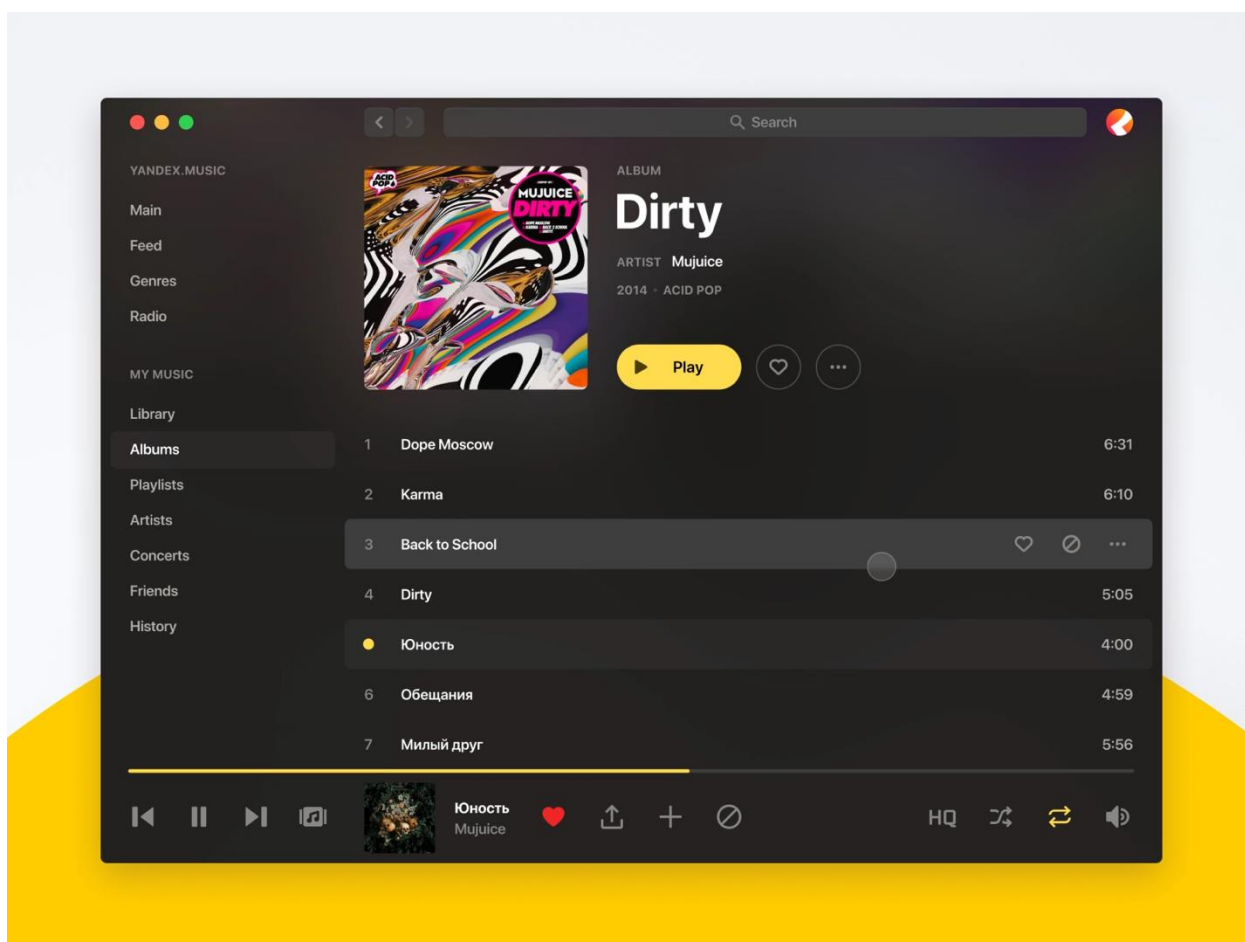


Рисунок 3 - Главная страница Яндекс Музыки

Минусы:

- Некоторые пользователи могут считать интерфейс сложным и запутанным;
- Высокая стоимость премиум-подписки;
- Ограниченный доступ к новым альбомам для бесплатной версии;
- Реклама для бесплатной версии.

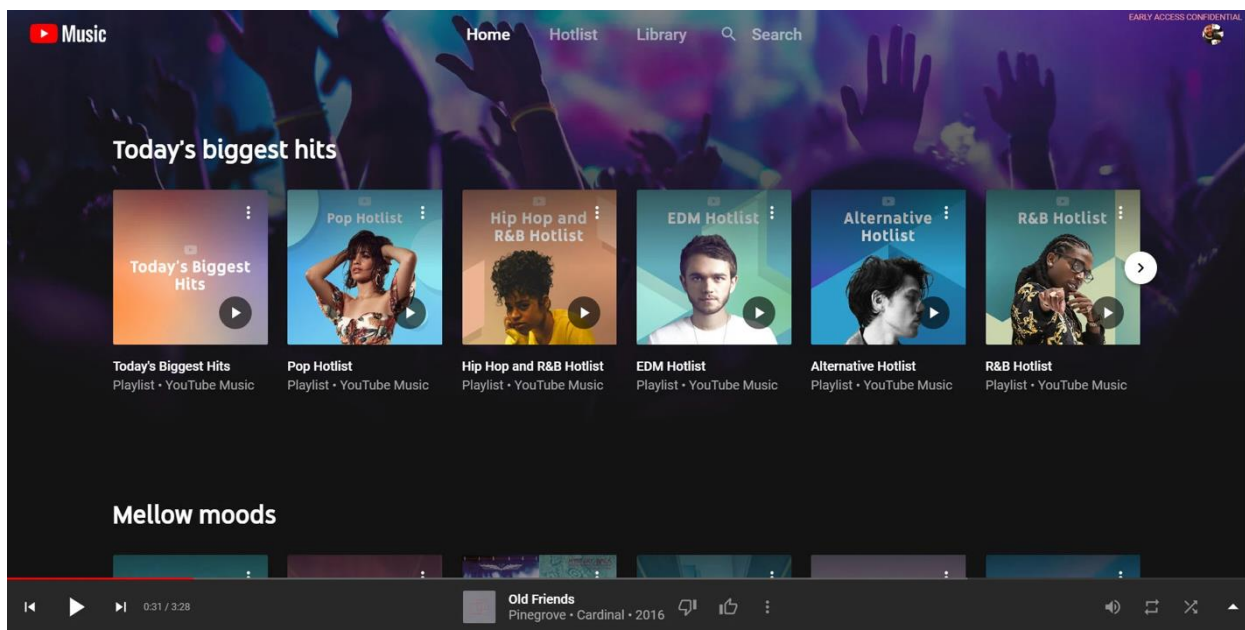


Рисунок 4 - Главная страница Youtube Music

2.3 Анализ поставленной задачи

2.3.1 Диаграмма вариантов использования

Ниже приведено подробное описание диаграммы использования (см. Рисунок 5 и Рисунок 6). Система используется следующими группами пользователей:

- Неавторизованный пользователь;
- Авторизованный пользователь;
- Администратор.

Неавторизованный пользователь может:

- Ограниченно просматривать контент (без возможности прослушать композицию);
- Сортировать контент.

Авторизованный пользователь может:

- Сортировать контент;
- Управлять своим аккаунтом. В управлении входит регистрация, вход в систему, редактирование профиля, а также возможность добавлять других зарегистрированных пользователей в друзья или удалять их;

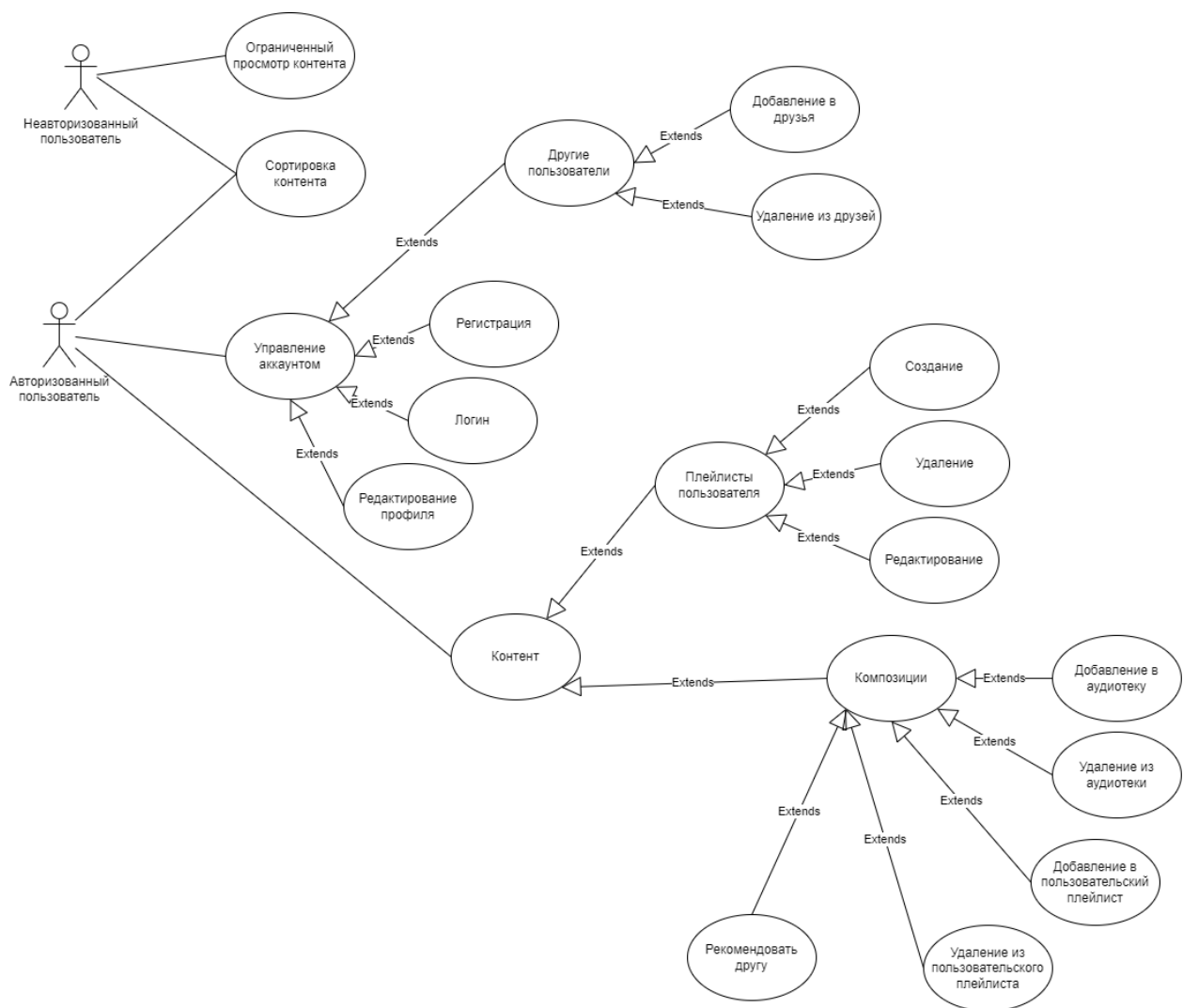


Рисунок 5 - Диаграмма вариантов использования (часть 1)

— Просматривать контент. Пользователь может создавать, удалять и редактировать свои плейлисты. Также авторизованный пользователь может добавлять в аудиотеку или личный плейлист или удалять оттуда композиции, рекомендовать другу композицию.

Администратор может:

- Удалять пользователей из системы в случае нарушения соблюдения правил платформы;
- Управлять контентом: добавлять жанры, исполнителей и композиций, а также удалять плейлисты пользователей в случае нарушений правил платформы.

2.3.2 Диаграмма последовательности

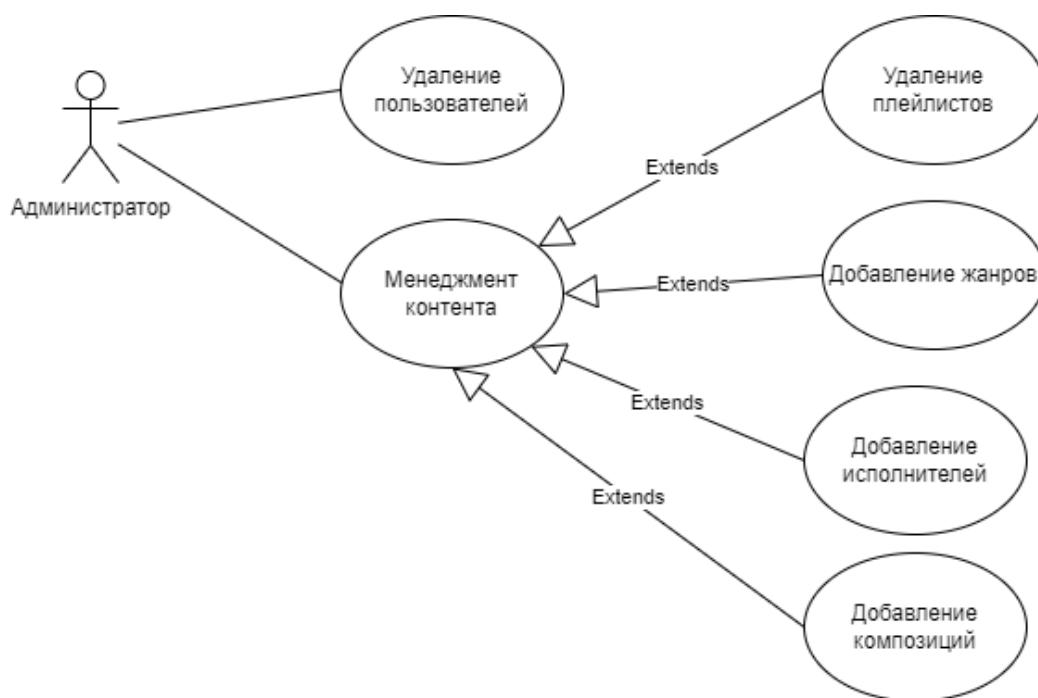


Рисунок 6 - Диаграмма вариантов использования (часть 2)

Одной из основной целью приложения является показ пользователю композиций на основе его музыкальных предпочтений. Это достигается не только за счет возможности фильтрации контента по жанрам и исполнителю, но и с помощью учета статистики пользователя. Далее рассматривается диаграмма последовательности при нажатии прослушивания композиции (см. Рисунок 7).

При нажатии кнопки прослушивания композиции с вебсайта посылается запрос на сервер, где происходит обработка. В случае если запрос был некорректен (например, запрос отправил неавторизованный пользователь или запрашивалась несуществующая композиция), пользователю демонстрируется экран с ошибкой. В противном случае перед получением файла композиции из хранилища в базу данных записывается информация о жанре композиции и уникальный идентификатор пользователя, а также в отношение «история» заносится информация о песне.

2.3.3 Диаграмма классов

Для работы сервера используются основные сущности, изображенные на диаграмме классов (см. Рисунок 8). Далее следует описание классов:

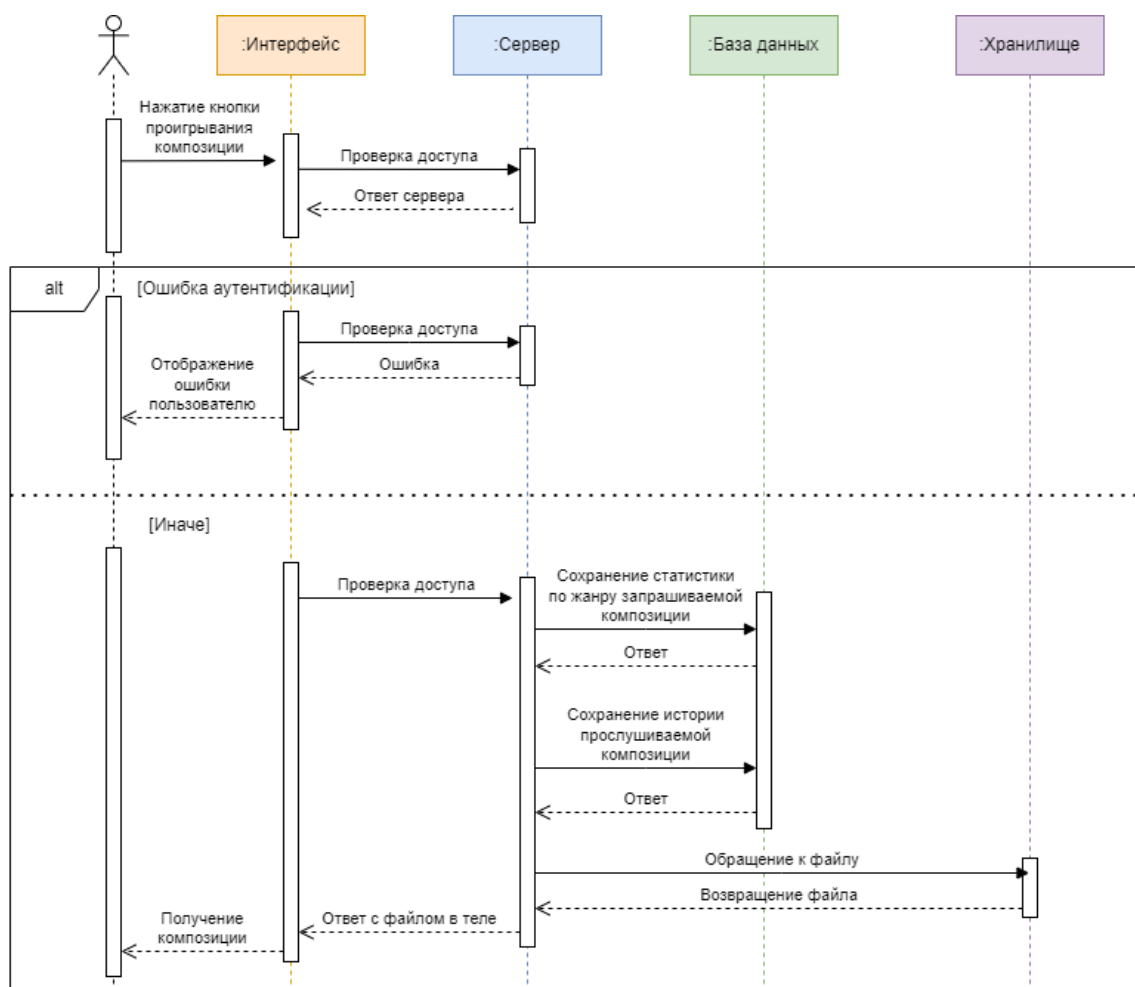


Рисунок 7 - Диаграмма последовательности при нажатии прослушивании композиции

- User – класс пользователя;
- Genre – класс жанра;
- Singer – класс исполнителя;
- RefreshToken – класс токена для обновления access токена, нужный для отправки авторизованных запросов;
- Verification – класс для хранения информации о верификации пользователя;
- Statistic – класс статистики пользователя по прослушиваемым жанрам;
- Song – класс композиции;
- History – класс истории прослушанных композиций;

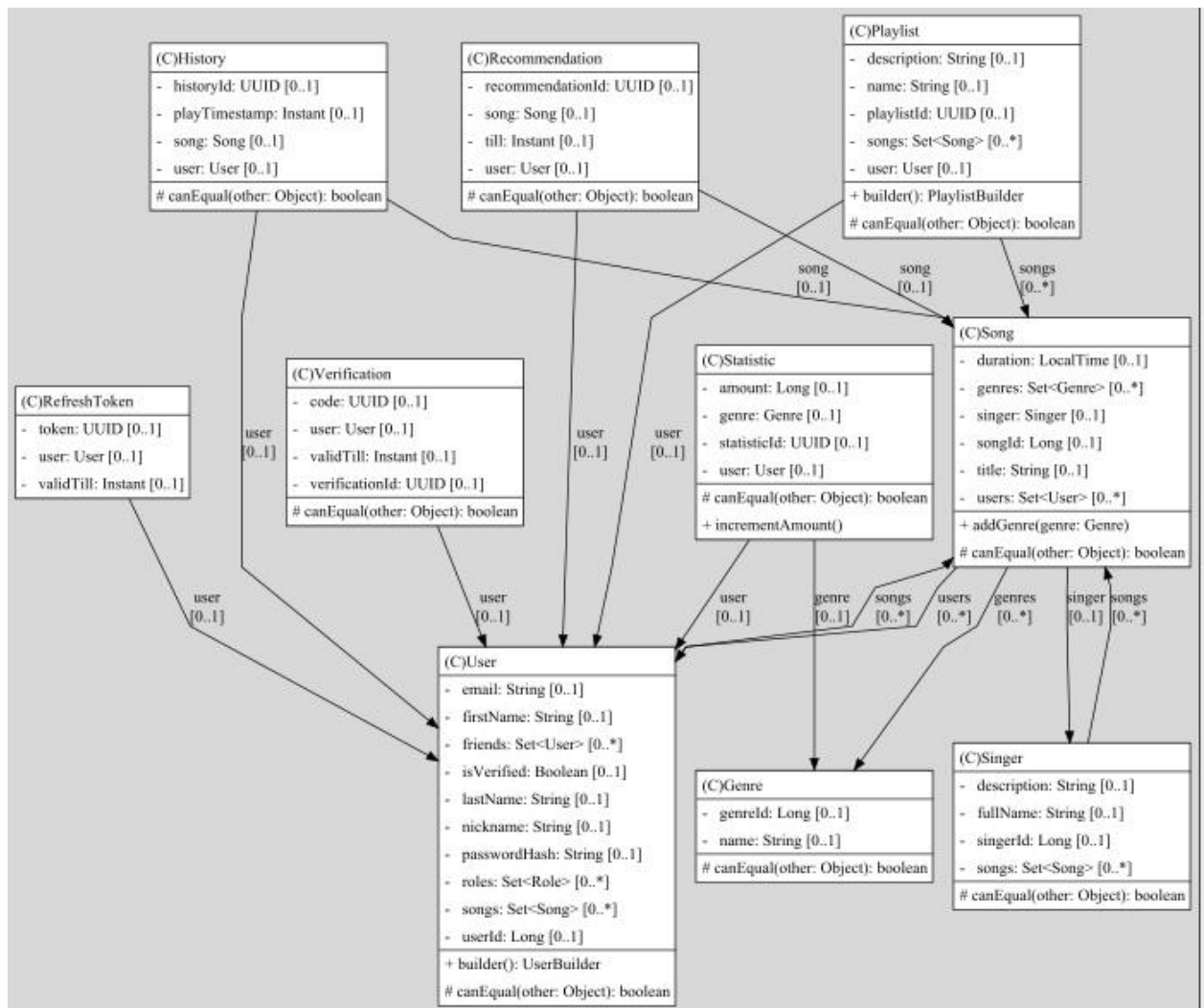


Рисунок 8 - Диаграмма классов

- Recommendation – класс для хранения информации о рекомендованной песни другим пользователем;
- Playlist – класс плейлиста.

2.3.4 Диаграмма развертывания

На рисунке 9 изображена диаграмма развертывания приложения. Основными компонентами приложения являются сервер, который используют базу данных, и клиентская часть с веб-браузером. Для работы серверной и клиентской частей необходим выход в интернет.

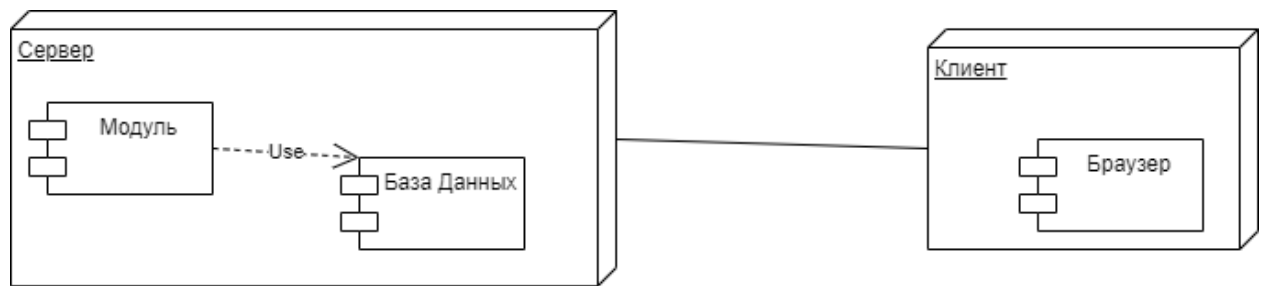


Рисунок 9 - Диаграмма развертывания

3 Реализация

3.1 Реализация серверной части приложения

Серверная часть приложения является монолитом. Организацию кода можно увидеть на Рисунке 10. Каждый класс помещен в соответствующую его предназначению папку, таким образом структурируя код.

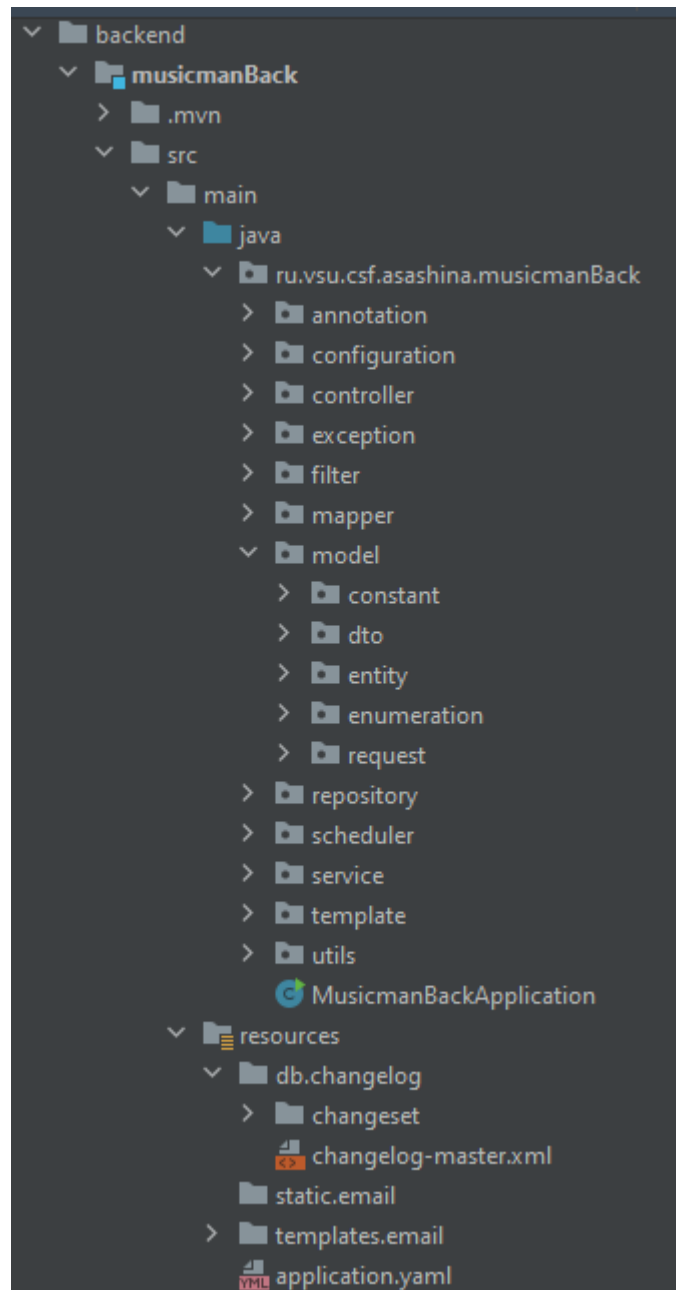


Рисунок 10 - Структура кода серверной части приложения

Для соблюдения принципов SOLID и удобочитаемости [8] разработка классов следовала цепочке model (сущности базы данных и приложения) – repository (класс, отвечающий за взаимодействие с базой данных) – service

(класс с бизнес-логикой) – controller (класс, к которому происходит обращение по HTTP-запросу извне).

Также была реализована Swagger документация, содержащая информацию о наборе запросов, которые можно послать приложению. Пример документации показан на Рисунке 11.

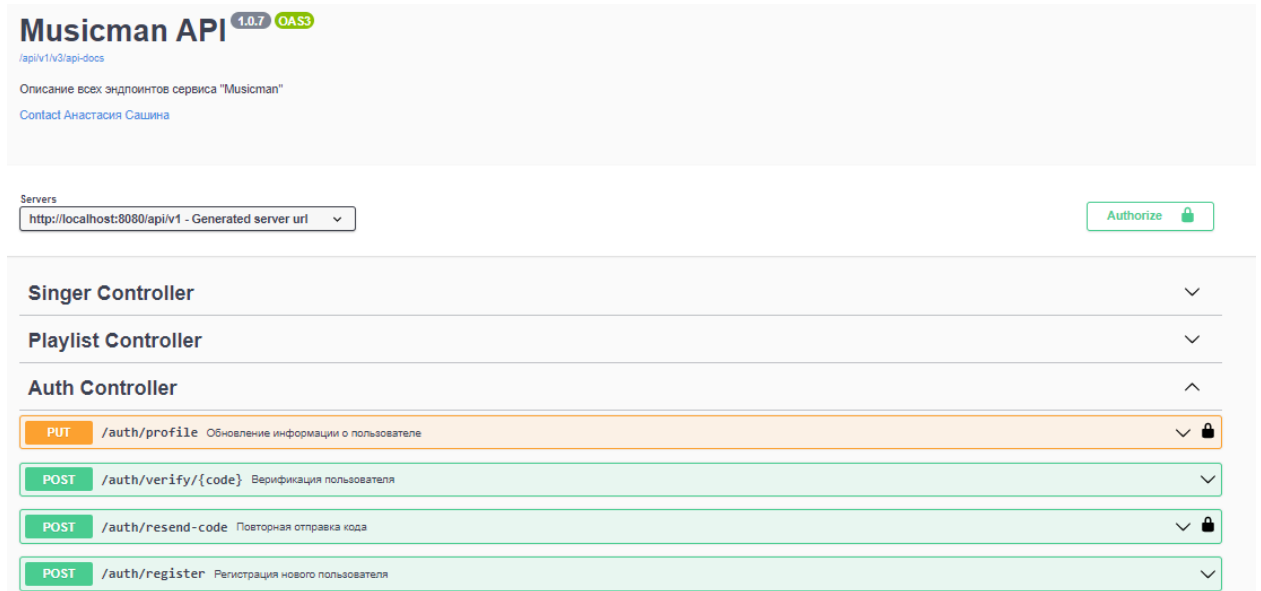


Рисунок 11 - Swagger документация

3.2 Реализация клиентской части приложения

Заключение

В ходе выполненной работы было реализовано приложение Musicman для прослушивания музыки. Веб-приложение обладает визуальным интерактивным интерфейсом, который необходим для взаимодействия пользователя с приложением и позволяет выводить на экран всю нужную информацию. Были выполнены следующие задачи:

- Разработана серверная часть приложения, развернутая на виртуальной машине;
- Разработана клиентская часть, необходимая для взаимодействия с логикой приложения;
- С помощью API была реализована связь между клиентской и серверной частями;
- Приложение учитывает статистику по композициям для рекомендации, осуществляет логику по возможно рекомендации песни и другу и использует эквалайзер.

Список использованных источников

1. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы ГОСТ 34.602-2020 — Взамен ГОСТ 34.602-89; введен 01.01.2020
2. Официальный сайт Spring: сайт. — URL: <https://spring.io/> (дата обращения: 21.03.2023). — Текст: электронный.
3. Официальный сайт React.js: сайт. — URL: <https://react.dev/> (дата обращения — 21.03.2023). — Текст: электронный.
4. Официальный сайт PostgreSQL: сайт. — URL: <https://www.postgresql.org/> (дата обращения: 21.03.2023). — Текст: электронный.
5. Официальный сайт Liquibase: сайт. — URL: <https://www.liquibase.org/> (дата обращения: 21.03.2023). — Текст: электронный.
6. Официальный сайт Effector: сайт. — URL: <https://effector.dev/> (дата обращения: 21.03.2023). — Текст: электронный.
7. Официальный сайт Material UI сайт. — URL: <https://mui.com/> (дата обращения: 21.03.2023). — Текст: электронный.
8. Мартин Р. Чистый код: создание, анализ и рефакторинг / Р. Мартин — СПб. : Питер, 2022 — 464с.