МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет *компьютерных наук*

Кафедра *информационных систем*

Каталог музыкальных произведений

*Курсовой проект*

*по дисциплине*

*Технологии программирования*

*09.03.02. Информационные системы и технологии*

*Информационные системы и сетевые технологии*

Преподаватель *\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.С. Тарасов, ассистент* \_\_.\_\_.20\_\_

Обучающиеся \_\_\_\_\_\_\_\_\_\_\_\_\_\_ *К.Г. Чилиби, А.А Пилюгин*

*3 курс, д/о*

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *И.Ю. Иванов, преподаватель*

Воронеж2019

**Содержание:**

[**Введение** 3](#_Toc10699097)

[**Постановка задачи** 3](#_Toc10699098)

[**Обоснование выбора средств разработки** 6](#_Toc10699099)

[**Модульная схема проекта** 7](#_Toc10699100)

[**Анализ предметной области** 8](#_Toc10699101)

[**Диаграммы** 15](#_Toc10699102)

[ER-диаграмма 15](#_Toc10699103)

[Диаграмма классов 16](#_Toc10699104)

[Диаграмма активности 17](#_Toc10699105)

[Диаграмма последовательности 18](#_Toc10699106)

[Use Case 19](#_Toc10699107)

[Диаграмма состояний 20](#_Toc10699108)

[Диаграмма объектов 21](#_Toc10699109)

[Диаграмма развертывания 22](#_Toc10699110)

[**План тестирования** 23](#_Toc10699111)

[**Заключение** 30](#_Toc10699112)

[**Отчет о проделанной работе** 31](#_Toc10699113)

# **Введение**

В современном мире все сложнее находить сайты для удобного прослушивания музыки.

Поэтому целью нашей работы является создание такого сервиса, который позволит пользователю находить и слушать музыкальные произведения без особого труда.

# **Постановка задачи**

**Условия задачи:**

Разрабатываемый проект представляет собой приложение для прослушивания музыки. Готовая система обеспечит возможность просмотра и изменения списков музыкальных произведений, а также загрузки и сохранения музыкальных композиций.

**Сроки выполнения работы:** февраль 2019 года – июнь 2019 года.

**Основание для разработки:** современные образовательные учреждения требуют меньшей бюрократии для более гибкого и совершенного учебного процесса.

**Характеристика объекта разработки:** Объектом разработки является система каталога музыкальных произведений Готовое приложение подразумевает возможность поиска и прослушивание и скачивание песен, через сайт.

**Цель**: разработать приложение, реализующее каталог музыкальных произведений.

**Сфера использования**: музыкальный бизнес и связанные с ним услуги.

**Основные требования**:

1. Регистрация и авторизация пользователей.

2. Возможность просмотра:

Для обычного пользователя, гостя и администратора - списков музыкальных композиций(все пользователи могут слушать песни и скачивать их, отличие состоит в том, что администратор и авторизованный пользователь могут добавлять песни в список любимых композиций).

Для администратора – всех списков.

3. Возможность добавления, удаления, редактирование информации администратором.

**Входная информация:** название музыкальных произведений, имена исполнителей, сами музыкальные композиции.

**Требования к программному обеспечению:** Система должна работать в современных версиях браузеров Google Chrome, Mozilla Firefox.

**Безопасность**: Для выполнения всех операций, пользователь должен являться администратором.

# **Обоснование выбора средств разработки**

Для реализации данного проекта был выбран язык программирования Python. Основание – огромное количество библиотек и достаточно высокая расширяемость, широкое распространение в среде разработчиков.

Использование Python помогает крайне удобно выстроить взаимодействие между back-end и front-end частями проекта, а также реализовать все необходимые модули приложения.

Веб-фреймворк Flask был выбран в качестве основы реализуемого проекта. Основание – легковесность и гибкость, данный фреймворк не имеет жесткой структуры и позволяет выбирать модули под конкретные задачи и устанавливать их по мере необходимости.

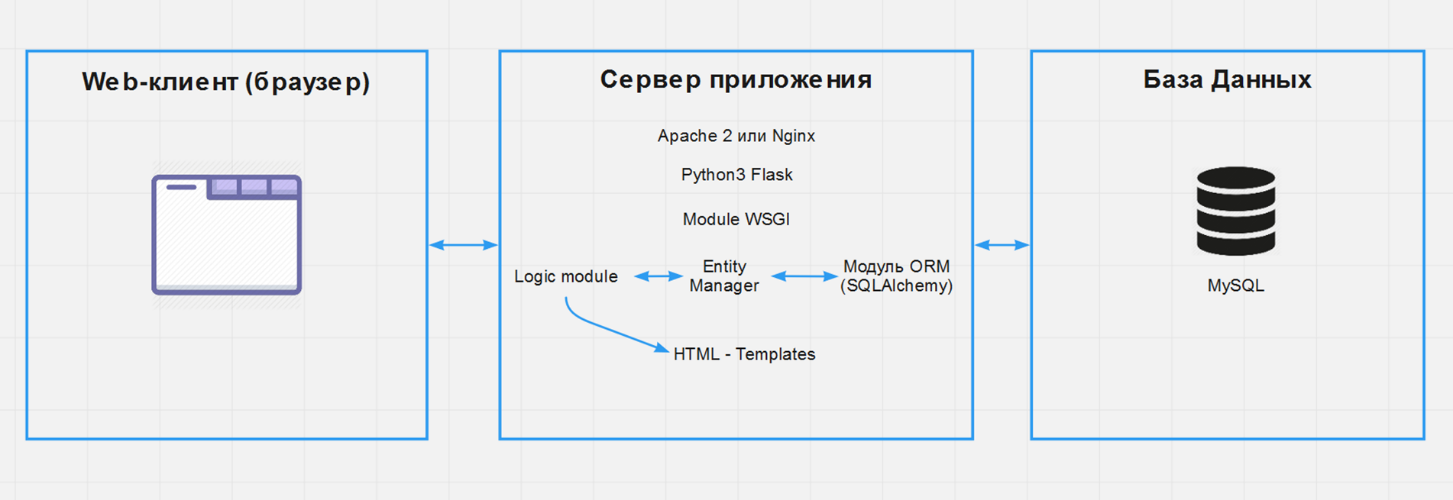
Язык программирования Python удобен для работы с базами данных.

В данном проекте используется фреймворк Flask, для организации работы базы данных был выбрана библиотека SQLAlchemy. В качестве СУБД была выбрана SQLITE. **SQLITE**- это самая распространенная полноценная серверная СУБД. SQLITE очень функциональная, свободно распространяемая СУБД, которая успешно работает с различными сайтами и веб приложениями. Также к числу ее очевидных достоинств относится: простота в работе, масштабируемость, удобство в работе с большими объемами данных и высокая производительность.

При разработке front-end части были использованы широко распространенные HTML (в соответствии со стандартом 4.0) и CSS.

Реализация на языке Python позволила легко организовать связь back-end части (логика проекта, функционирование базы данных) с front-end частью (веб-формы HTML, динамические структуры).

# **Модульная схема проекта**



SQLITE

Рисунок 1 Общая функциональная схема приложения

На рисунке 1 представлена модульная схема проекта, описывающая общую архитектуру реализовываемого приложения.

Пользователи отправляют запросы, используя динамические поля и элементы страниц HTML. Front-end часть передает пакеты данных.

В back-end части реализована логика приложения. Она выполняется на сервере. Он обрабатывает запрос и обращается к базе данных, после возвращает ответ в front-end часть, где информация поступает пользователям.

**Анализ предметной области**

1. Область применения.

Областью применения данного приложения является сфера услуг, непосредственно связанная с современной музыкальной индустрией, выходящей на новые рынки сбыта и использующая новейшие интернет-технологии.

1. Общие положения.
2. Настоящее положение устанавливает требования к сайту, предоставляющему услуги, связанные с прослушиванием музыкальных произведений
3. При разработке настоящего Положения за основу принята модель, реализованная на основе разработок популярных стриминговых музыкальных сервисов, таких как Apple Music и Spotify.
4. Переход на современную Интернет-систему упростит ведение бизнеса как для музыкальных лейблов, так и для исполнителей. Обеспечит лучшую рекламу новых музыкальных альбомов.

# **Диаграммы, используемые в данной работе**

## ER-диаграмма

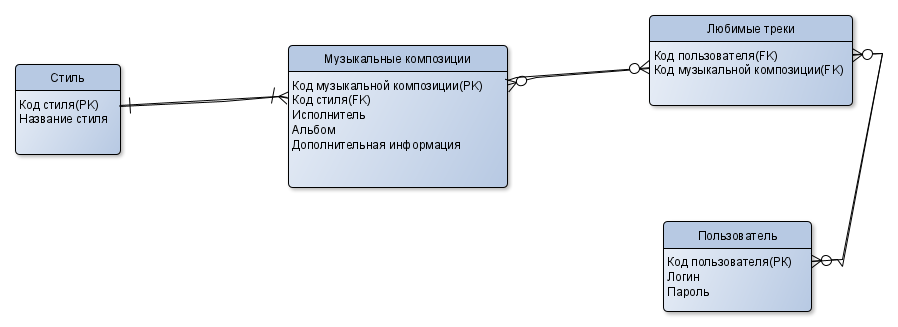


Рисунок 2 ER диаграмма

## Диаграмма классов

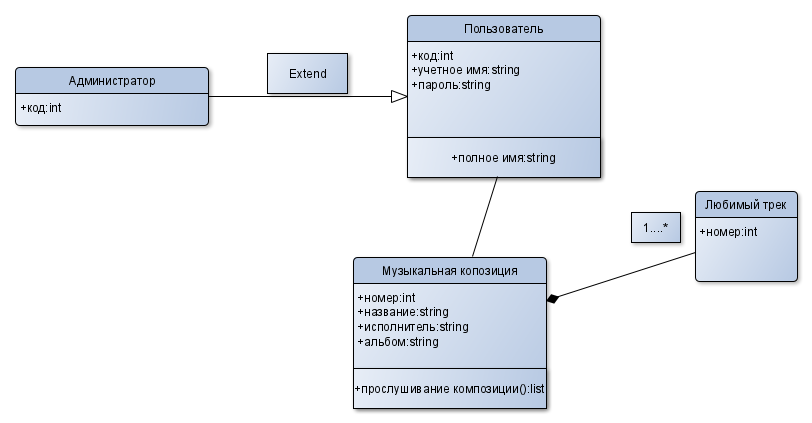


Рисунок 3 Диаграмма классов

## Диаграмма активности

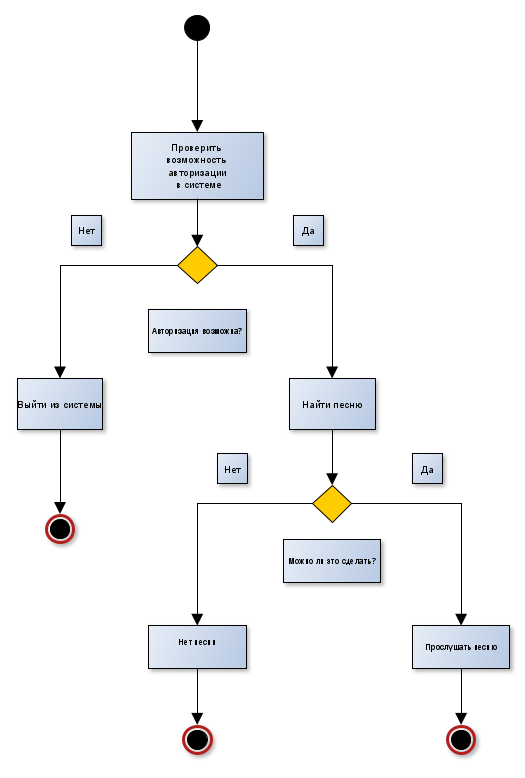


Рисунок 4 диаграмма активности

## Диаграмма последовательности

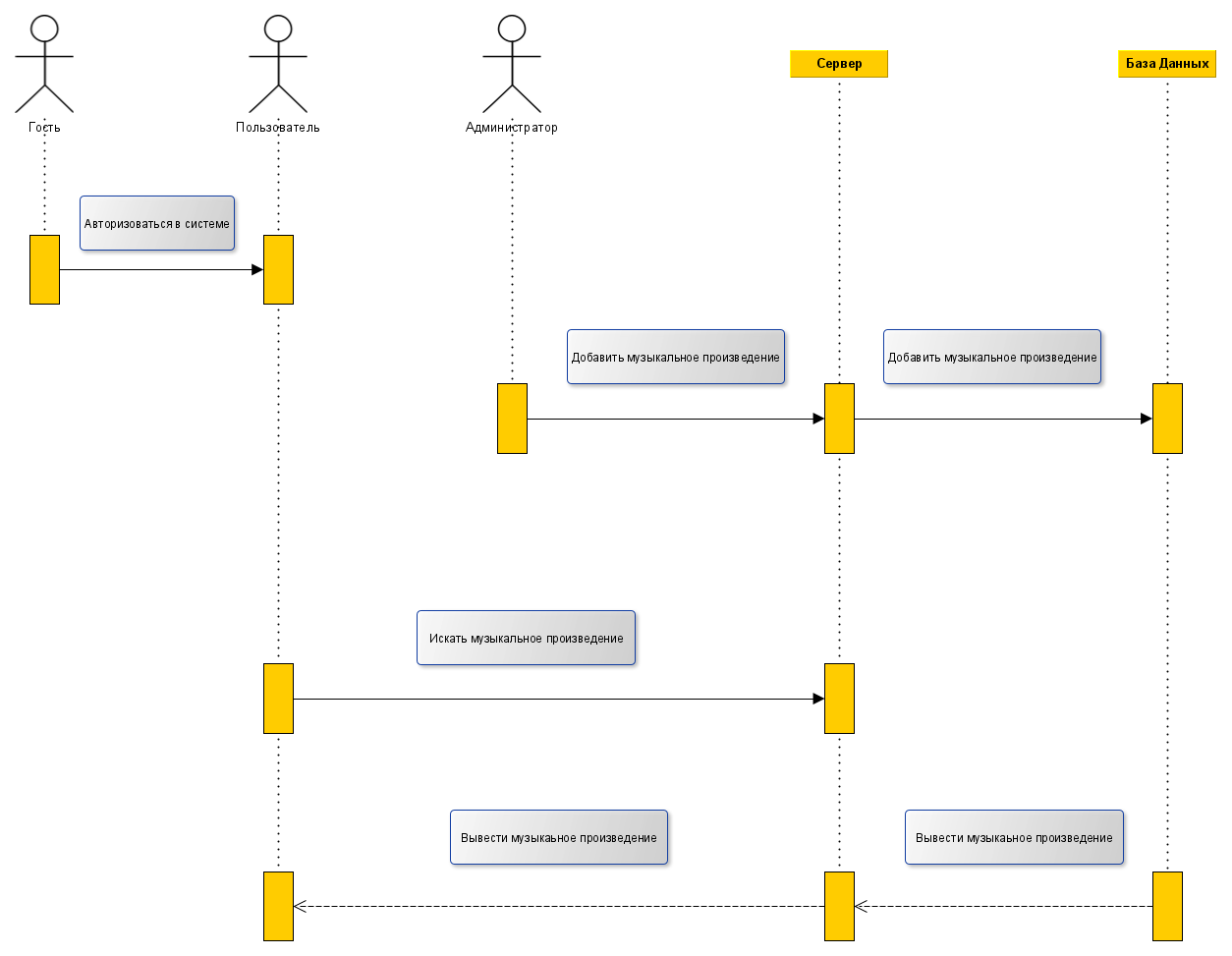


Рисунок 5 диаграмма последовательности

## Use Case

Диаграмма вариантов использования объясняет функционал системы. C системой взаимодействуют 3 пользователя – администратор, пользователь и гость. Действия каждого актера характеризуются в зависимости от его роли.

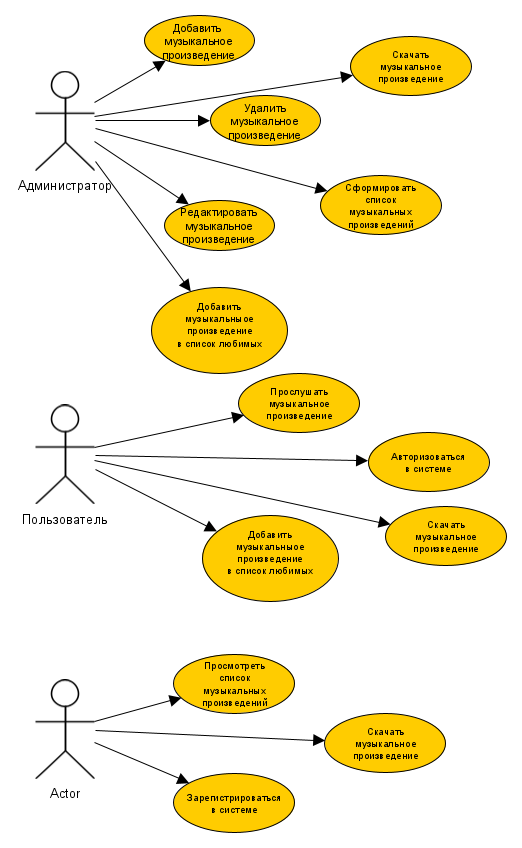


Рисунок 6 диаграмма вариантов использования

Диаграмма состояний

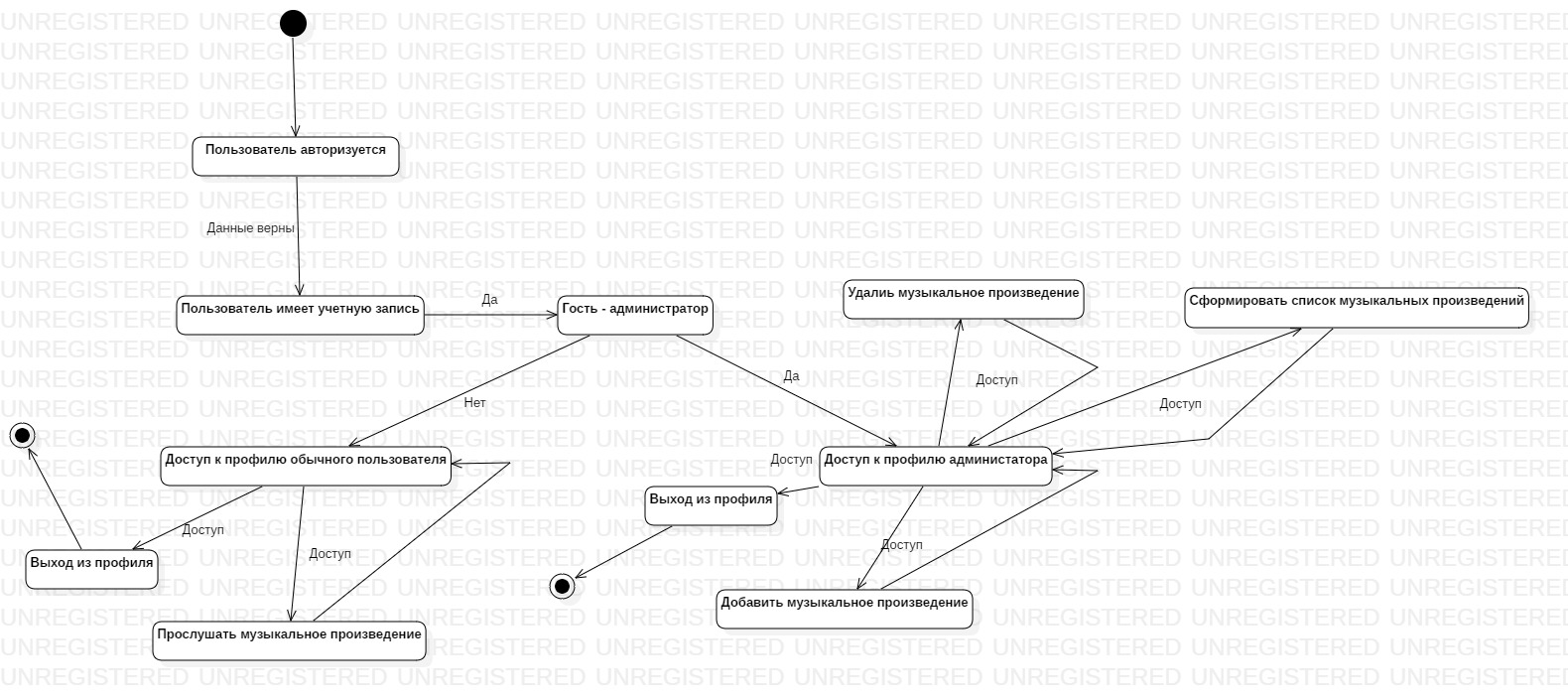


Рисунок 7 диаграмма состояний

## Диаграмма объектов

Данная диаграмма отражает состояние системы в определенный момент времени. Тут отображены объекты, соответствующие классам, но уже с заданными конкретными атрибутами.

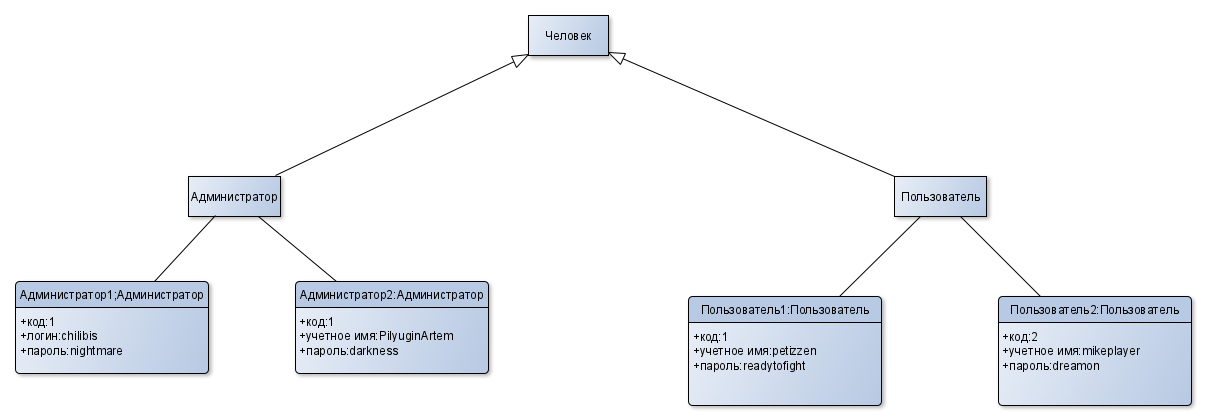
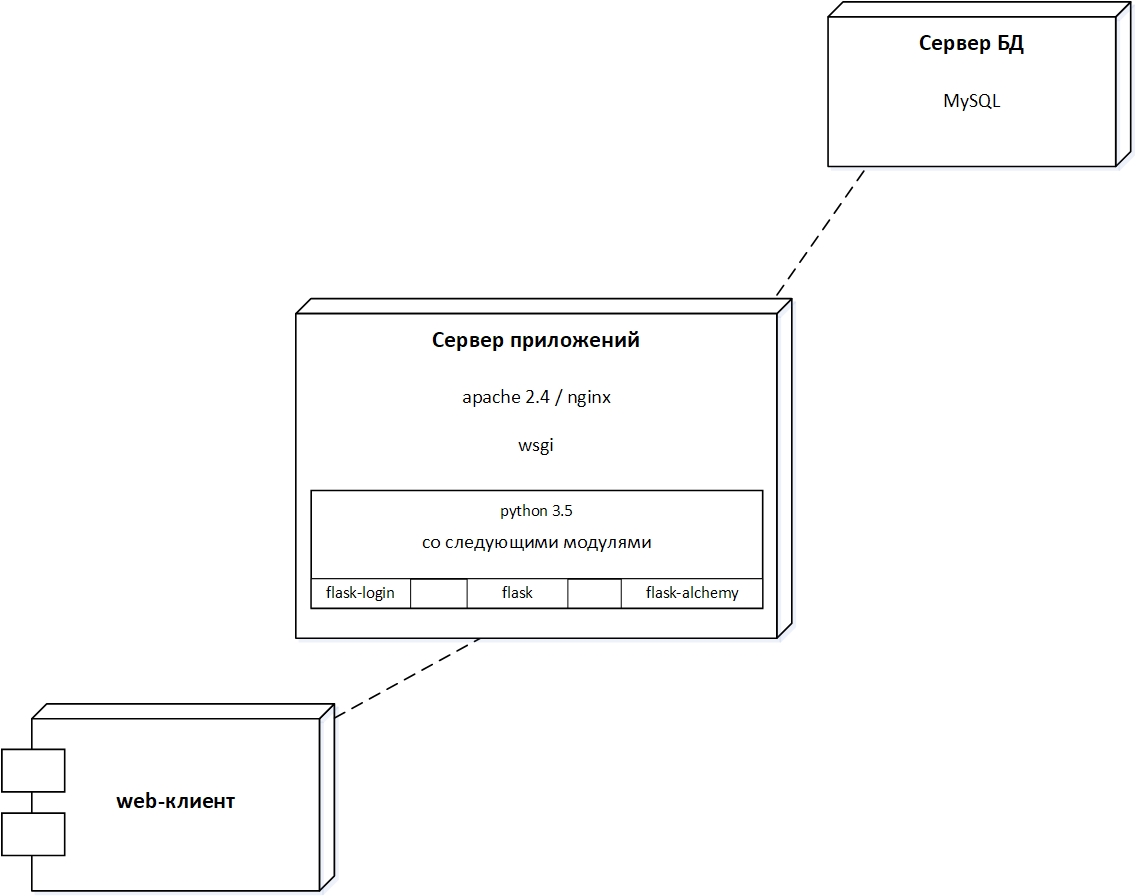


Рисунок 8 диаграмма объектов

## Диаграмма развертывания



SQLITE

Рисунок 9 диаграмма развертывания

# **План тестирования**

Таблица 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Предпосылка** | **Шаг** | **Отметка о прохожде-нии** | **Комментарий** |
| 1 | Добавление нового музыкального произведения | Авторизация в системе | 1. Ввести основную информацию о композиции 2. Загрузить музыкальную композицию на сайт. | пройдено |  |
| 2 | Добавление музыкального произведения в список. | Заполнение данных о музыкальной композиции | 1. Открыть поиск списков музыкальных композиций студента 2. Проверить наличие загруженной музыкальной композиции в списке | пройдено |  |
| 3 | Прослушивание музыкальной композиции | Загрузка музыкальной композиции | 1. Авторизация 2. Используя поиск найти музыкальную композицию  3. Проверить возможность ее прослушивания | пройдено |  |
| 4 | Удаление музыкальной композиции | Заполнение данных о музыкальных композициях и создание списков музыкальных произведений. | 1. Открыть список музыкальных композиций. 2. Выбрать музыкальную композицию 3. Удаление самой музыкальной композиции 4. Удаление музыкальной композиции из списков 5. Удаление всех данных о музыкальной композиции | пройдено |  |

# **Заключение**

В заключении, итогом проделанной работы является полностью реализованное приложение «Каталог музыкальных произведений». В соответствии с описанными в техническом задании требованиями, выполнен следующий функционал:

Система удовлетворяет следующим требованиям:

1. Прослушивание музыкальных композиций.

2. Возможность редактирования списков музыкальных композиций.

3. Возможность скачивания треков.

Данное приложение полностью удовлетворяет требованиям, излагаемым в техническом задании. Заданный функционал отлажен и работает.

Гость может просматривать списки музыкальных композиций.

Пользователь может прослушивать музыкальные композиции.

Администратор формирует список музыкальных композиций, загружает на сайт музыкальные композиции и добавляет данные о них.

Окончательным результатом является простое, удобное и интуитивно понятное приложение, удовлетворяющее потребностям всех пользователей данного сайта.

# **Отчет по проделанной работе**

Составление диаграмм между участниками команды было распределено следующим образом:

1. Диаграмма состояний – Константин Чилиби

2. Диаграмма объектов – Артем Пилюгин

3. Диаграмма развертывания – Константин Чилиби

4. Диаграмма классов – Константин Чилиби

5. ER-диаграмма - Артем Пилюгин

7. Диаграмма Use Case – Константин Чилиби

8. Диаграмма активности – Артем Пилюгин

9. Диаграмма последовательностей – Артем Пилюгин

Работа с приложением была реализована следующим образом:

1. ORM модели для работы с БД – Константин Чилиби

2. Модуль авторизации в системе – Константин Чилиби

3. Модуль регистрации – Артем Пилюгин

4. Модуль Администратора- Артем Пилюгин

5. Модуль поиска – Константин Чилиби

6. Модуль музыкальных композиций – Артем Пилюгин