

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет *компьютерных наук*

Кафедра *программирования и информационных систем*

Concert Hall

Веб-приложение для продажи билетов на музыкальные концерты

Курсовая работа

09.03.02 Информационные системы и технологии

Обучающийся _____ Е.П. Булавина, Ю.В. Петров, В.А. Горбушин, 3 курс

Руководитель _____ В.С. Тарасов, преподаватель

Воронеж 2021

Содержание

Содержание	2
Введение	3
1. Постановка задачи	4
2. Анализ предметной области	6
2.1. Глоссарий	6
2.2. Анализ аналогов	8
2.3. Анализ потребности	9
3. Продуктовые воронки	10
4. Диаграммы	12
4.1. Диаграмма прецедентов	12
4.2. Диаграмма классов	13
4.3. Диаграмма объектов	14
4.4. Диаграмма последовательностей	14
4.5. Диаграмма взаимодействий	19
4.6. Диаграмма состояний	20
4.7. Диаграмма активности	22
4.8. Диаграмма развертывания	24
4.9. Модель IDEF0	24
4.10. ER-диаграмма	26
5. Реализация	28
5.1. Средства реализации	28
5.2. Frontend-составляющая приложения	28
5.3. Backend-составляющая приложения	29
5.3.1. Работа с API	29
5.3.2. Наполнение клиентской части	29
5.4. Система авторизации	30
5.5. Описание функциональной части приложения и графического пользовательского интерфейса	30
6. Тестирование	41
Заключение	43
Список используемой литературы:	44

Введение

Развлечения всегда были неотъемлемой частью жизни человека. Они позволяют людям отвлечься от повседневной рутины, поднимают настроение, расслабляют. К этой сфере относится и музыка, которая, как известно, позволяет задать человеку практически любое настроение. Нет ничего удивительного в том, что люди оказались заинтересованы в публичных выступлениях любимых музыкантов. Так в свое время появились музыкальные концерты.

С развитием данной сферы количество как музыкальных коллективов, так и интересующихся ими людей только росло. Например, в одной только России в последние 10 лет в среднем 30 млн. человек ежегодно посещали 185 тыс. музыкальных концертов [1]. Людям оказались нужны способы максимально подробно узнавать о грядущих музыкальных концертах, что с развитием информационных технологий вышло на совершенно новый уровень: стали появляться билетные сайты-агрегаторы, предлагающие покупателям узнавать информацию об интересующих их концертах, а также приобретать билеты на них, не покидая свой дом. К примеру, в 2019 году по информации Moscow Ticketing Forum около 30% зрителей Москвы и Московской области приобретали билеты на музыкальные концерты через билетные сайты [2].

Сказанное выше позволяет сделать вывод, что разработка веб-приложения, позволяющего пользователям приобретать билеты на музыкальные концерты, имеет высокую актуальность по причине большой востребованности как людей в посещении концертов, так и в удобном дистанционном способе покупки билетов.

Следовательно, в данной курсовой работе будет описан процесс разработки веб-приложения для продажи билетов на музыкальные концерты.

1. Постановка задачи

Цель данной курсовой работы – разработка веб-приложения, позволяющего его посетителям и пользователям приобретать билеты на музыкальные концерты.

Задачи, которые необходимо решить в ходе развития проекта:

- Сбор информации об исследуемой предметной области;
- Проведение анализа полученных данных;
- Создание диаграмм, отражающих основные аспекты приложения;
- Разработка веб-приложения;
- Проведение тестирования веб-приложения;
- Размещение приложения на сервере.

Разработка веб-приложения разделена на разработку серверной и клиентской частей.

Разработка серверной части:

- Проработка и реализация базы данных;
- Реализация работы с используемыми API;
- Формирование запросов к базе данных.

Разработка клиентской части:

- Проработка дизайна веб-приложения;
- Формирование веб-страниц приложения;
- Реализация взаимодействия с серверной частью.

К приложению предъявляются следующие требования:

- Возможность просмотра информации о грядущих концертах (название концерта, место проведения концерта, дата и время проведения концерта, статус концерта);

- Возможность ознакомления с исполнителем посредством представленной на сайте информации о нем;
- Возможность приобретения билетов на выбранные концерты;
- Возможность написания рецензий на артистов и концерты, а также выставлять оценки в том случае, когда пользователь авторизован;
- Возможность получить контактные данные администрации приложения для связи (к примеру, для обсуждения внесения данных какого-либо концерта в приложение).

2. Анализ предметной области

2.1. Глоссарий

В курсовой работе будут использоваться следующие термины:

Аватар - Фотография или другое графическое изображение, используемое в учетной записи для персонализации пользователя.

Администратор - Специалист, уполномоченный управлять и администрировать определенный участок IT-среды.

Веб-приложение (сайт) - Клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера.

Карусель - Компонент слайд-шоу для цикличного повторения элементов.

Модератор - Административное лицо сервиса, которое действует от лица представителя веб-ресурса и выполняет определенный ряд задач, направленных на соблюдение пользователями правил данного ресурса.

Пользователь - Лицо, которое использует приложение для выполнения конкретной функции.

СУБД - Система управления базами данных. Комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными.

Android - Операционная система для мобильных устройств.

API - Программный интерфейс приложения. Описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

Bootstrap - Свободный набор инструментов для создания сайтов и веб-приложений.

CSS - Формальный язык, служащий для описания оформления внешнего вида документа, созданного с использованием языка разметки (HTML, XHTML, XML).

Flask - Фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2.

Genius - Сайт, позволяющий пользователям предоставлять аннотации и интерпретации текстов песен, новости, источники, стихи, документы.

Google Календарь - Сервис для планирования встреч, событий и дел, разработанный компанией Google.

HTML - Стандартизированный язык разметки веб-страниц воВсемирной паутине.

JavaScript - Мультипарадигменный язык программирования, используется как встраиваемый язык для программного доступа к объектам приложений.

jQuery - Набор функций JavaScript, фокусирующийся на взаимодействии JavaScript и HTML.

Linux - Семейство Unix-подобных операционных систем на базе ядра Linux.

PostgreSQL - Свободная объектно-реляционная система управления базами данных.

Python - Высокоуровневый язык программирования общего назначения.

REST - Архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

Songkick - Находящаяся в США служба поиска концертов, принадлежащая Warner Music Group.

Spotify - Интернет-сервис потокового аудио (стриминговый), позволяющий легально прослушивать музыкальные композиции, аудиокниги и подкасты, не скачивая их на устройство.

SQLAlchemy - Программная библиотека на языке Python для работы с реляционными СУБД с применением технологии ORM.

Windows 10 - Операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT.

2.2. Анализ аналогов

Для формулирования более конкретных требований к веб-приложению необходимо было провести анализ аналогов. К ним можно отнести такие сайты-агрегаторы как, например, Яндекс.Афиша, Афиша и сайт концертного зала «Event-Hall».

Яндекс.Афиша - это билетный сервис, работающий с 2005 года, созданный компанией Яндекс.

Сильные стороны сервиса:

- Выбор места;
- Наличие рекомендательной системы;

Слабые:

- Наличие назойливой рекламы, из-за которой просмотр содержимого сайта замедляется и становится менее комфортным.

Афиша — это медийно-сервисная платформа о развлечениях и культурном досуге, основанная в апреле 1999 года.

Сильные стороны сервиса:

- Возможность оставлять отзывы;

Слабые:

- Интерфейс покупки билета для разных мероприятий различается;

«Event-Hall» — сервис для продажи билетов в рамках одной торговой сети.

Сильные стороны сервиса:

- Сервис позволяет приобретать билеты напрямую у владельцев зала, в котором проходит концерт.

Слабые:

- Сервис предназначен только для одного концертного зала, т.е. мало количество доступных концертов, ограниченность по городу;

2.3. Анализ потребности

После изучения аналогов разрабатываемого проекта можно сделать вывод, что ни один из них не является идеальным решением поставленной задачи. Тем не менее, само большое количество различных сервисов, позволяющих приобретать билеты на музыкальные концерты, дает понять, что у людей есть потребность в этом. В связи с этим разработка проекта Concert Hall является актуальной, потому необходимо будет учесть недостатки аналогов и не допустить их.

3. Продуктовые воронки

Главной задачей приложения является продажа билетов на концерты. В связи с этим при помощи сервиса Яндекс.Метрика были добавлены следующие продуктовые воронки. Они позволят определить заинтересованность пользователей в различных составляющих веб-приложения

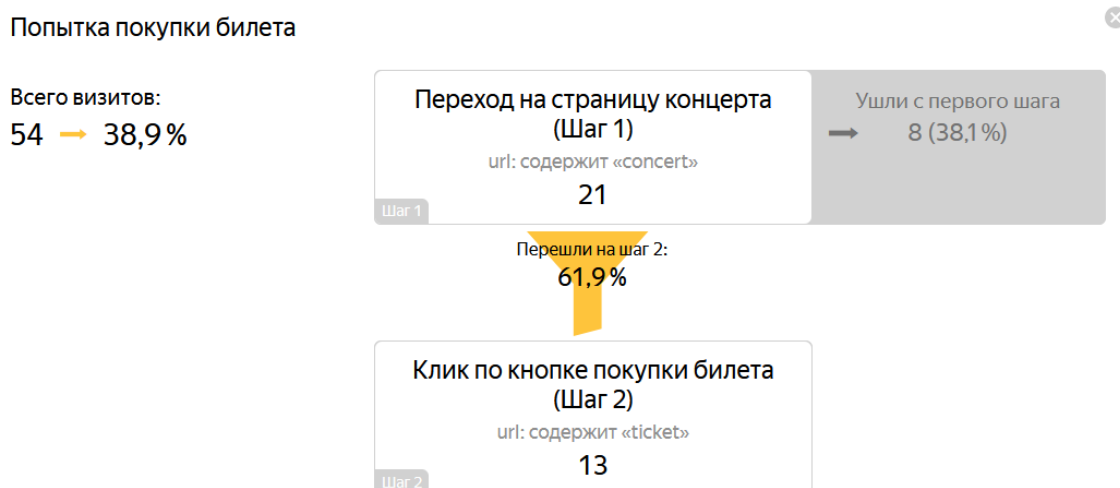


Рисунок 1 - Продуктовая воронка для попытки покупки билета

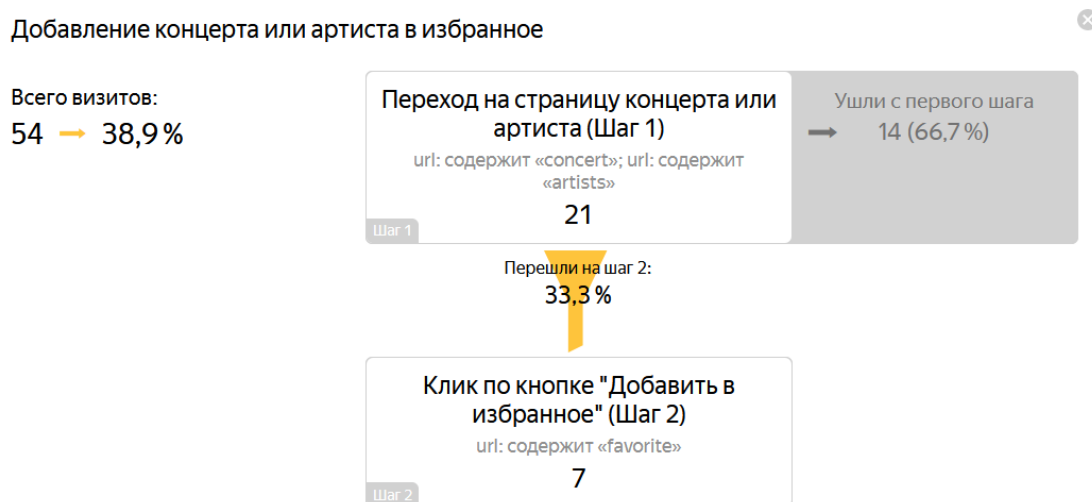
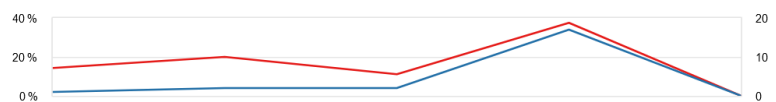


Рисунок 2 - Продуктовая воронка для добавления концерта или артиста в избранное

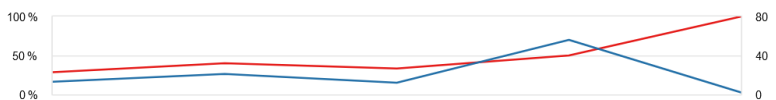
Поиск 🔍

Конверсия 23,3 %
Достижения цели 22
Целевые визиты 10



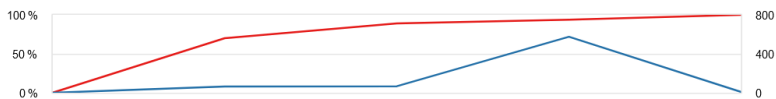
Просмотр артиста, участвующего в концерте 🎤

Конверсия 41,9 %
Достижения цели 104
Целевые визиты 18



Просмотр страницы концерта 🎫

Конверсия 72,1 %
Достижения цели 714
Целевые визиты 31



Переход в меню настроек ⚙️

Конверсия 53,5 %
Достижения цели 221
Целевые визиты 23



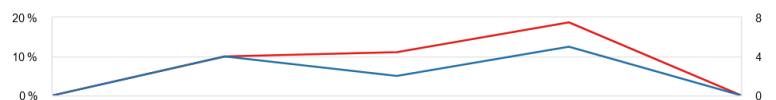
Смена аватара пользователя 👤

Конверсия 20,9 %
Достижения цели 20
Целевые визиты 9



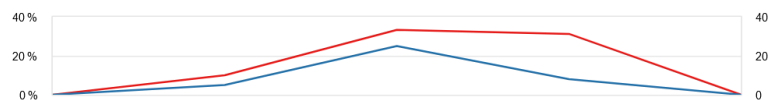
Добавление в избранное 📌

Конверсия 11,6 %
Достижения цели 11
Целевые визиты 5



Попытка привязки к Spotify 🎧

Конверсия 20,9 %
Достижения цели 38
Целевые визиты 9



Попытка привязки к Google Календарь 📅

Конверсия 25,6 %
Достижения цели 31
Целевые визиты 11

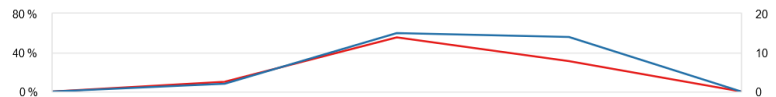


Рисунок 3 - Конверсии на Яндекс.Метрике

4. Диаграммы

4.1. Диаграмма прецедентов

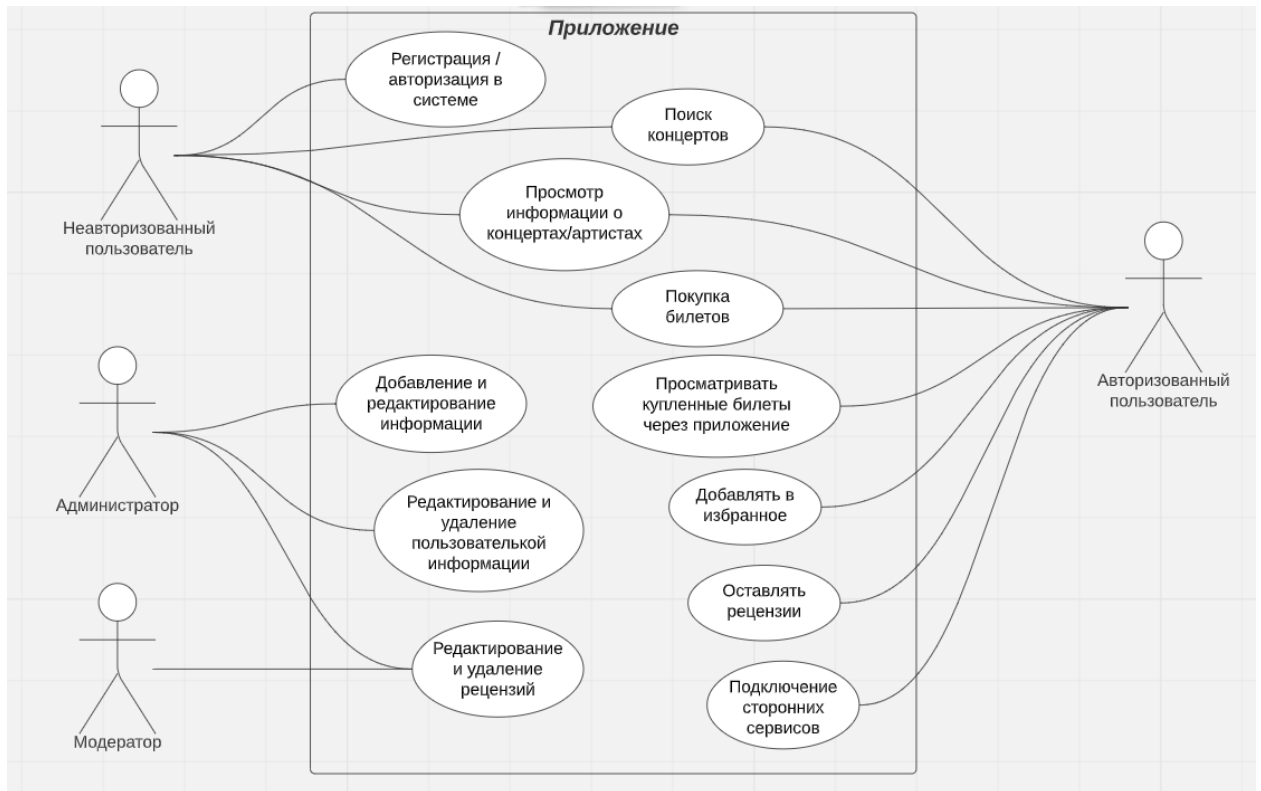


Рисунок 4 - Диаграмма прецедентов (use-case)

На данной диаграмме представлены функции актеров (пользователей системы) в системе.

Неавторизованный пользователь: регистрация/авторизация в системе, поиск концертов, просмотр информации о концертах/артистах, покупка билетов;

Авторизованный пользователь (дополнительно к функциям неавторизованного пользователя): просмотр купленных билетов через приложение, возможность добавлять в избранное концерты и артистов, возможность оставлять рецензии на концерты и артистов, подключение сторонних сервисов;

Модератор: редактирование и удаление рецензий;

Администратор (дополнительно к функциям модератора): добавление и редактирование информации, редактирование и удаление пользовательской информации.

4.2. Диаграмма классов

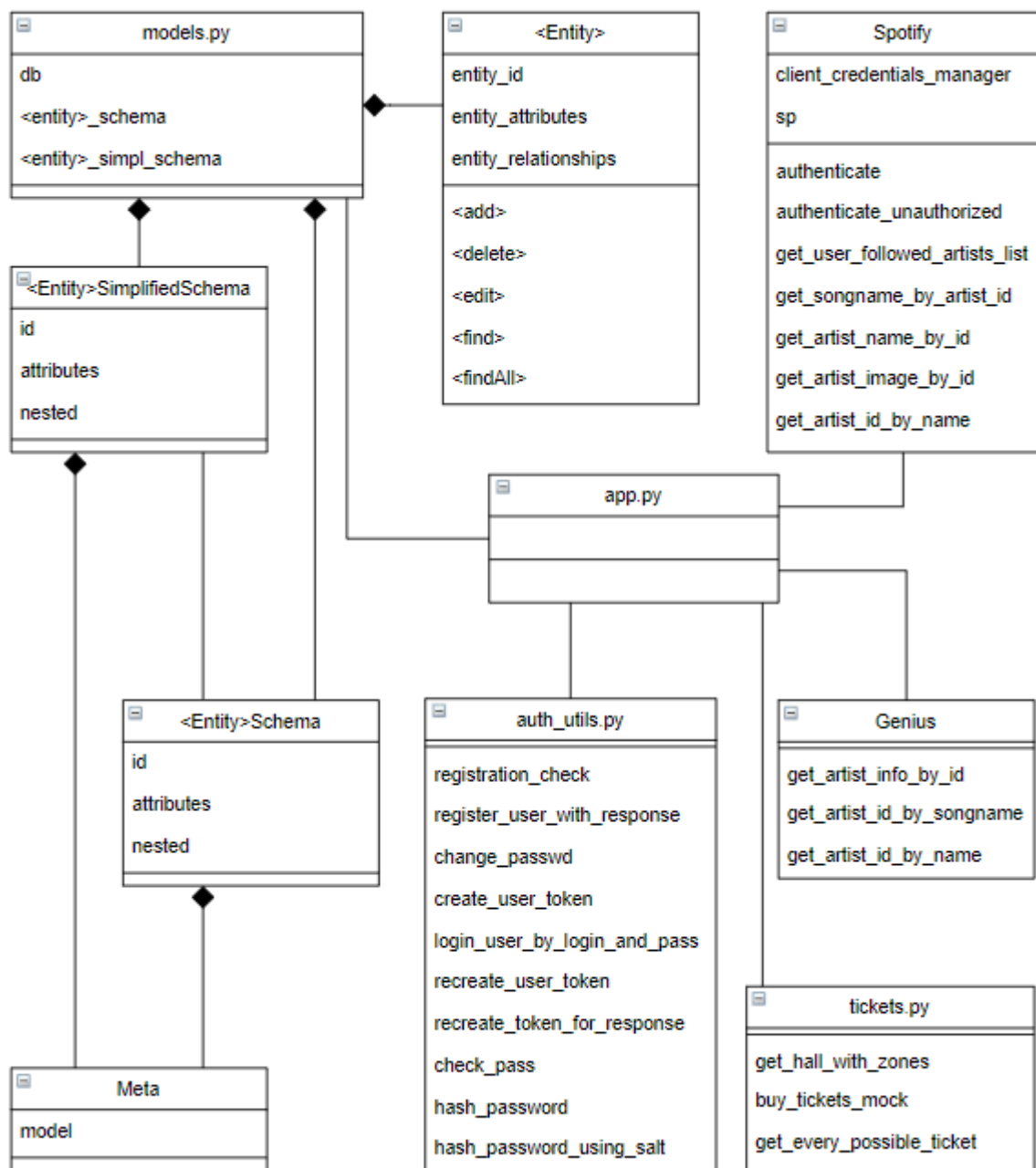


Рисунок 5 - Диаграмма классов приложения

В данной диаграмме представлены классы системы, их свойства, атрибуты и функции. Взаимодействие между модулями производится посредством контроллера.

4.3. Диаграмма объектов

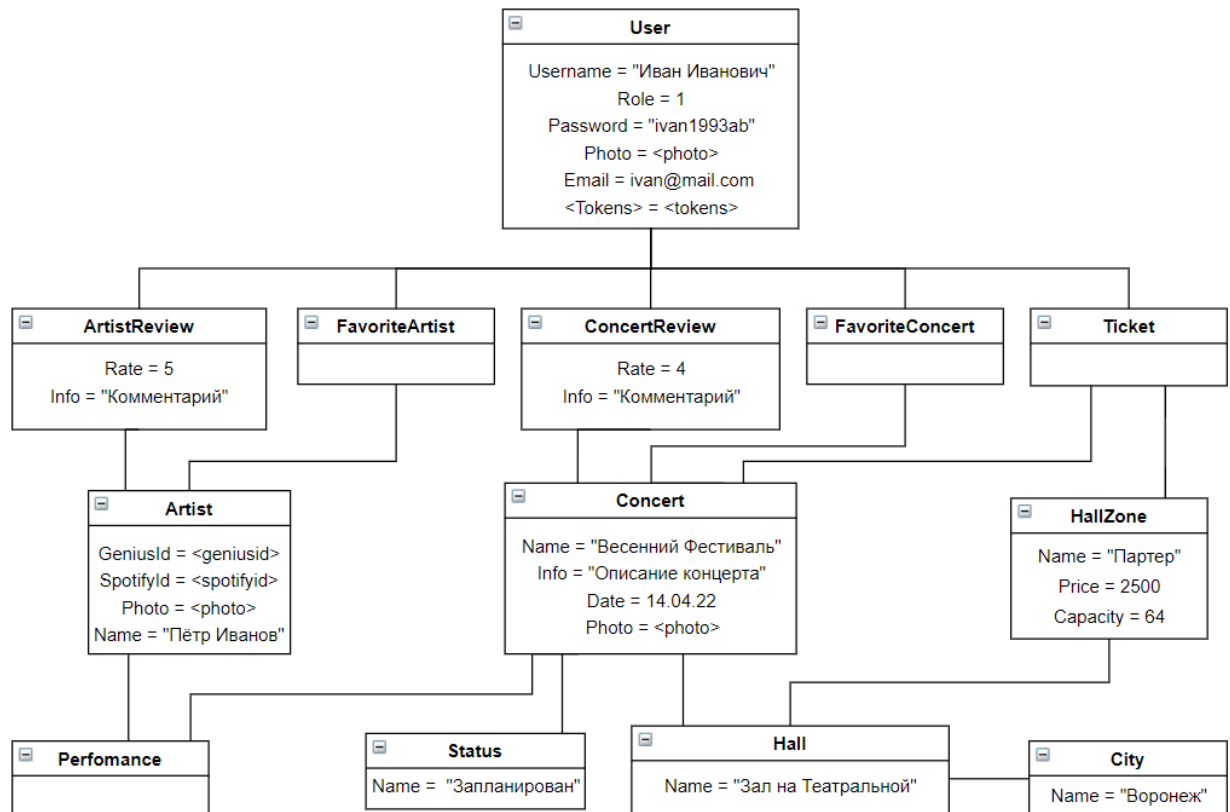


Рисунок 6 - Диаграмма объектов проекта

Диаграмма объектов показывает множество объектов - экземпляров классов и отношений между ними в какой-то момент времени. Выступает в роли дополнения к диаграмме классов.

4.4. Диаграмма последовательностей

Диаграмма последовательностей служит для описания взаимодействия объектов системы по мере их проявления в сценарии. В данном случае имеем 5 основных сценариев.

Регистрация аккаунта

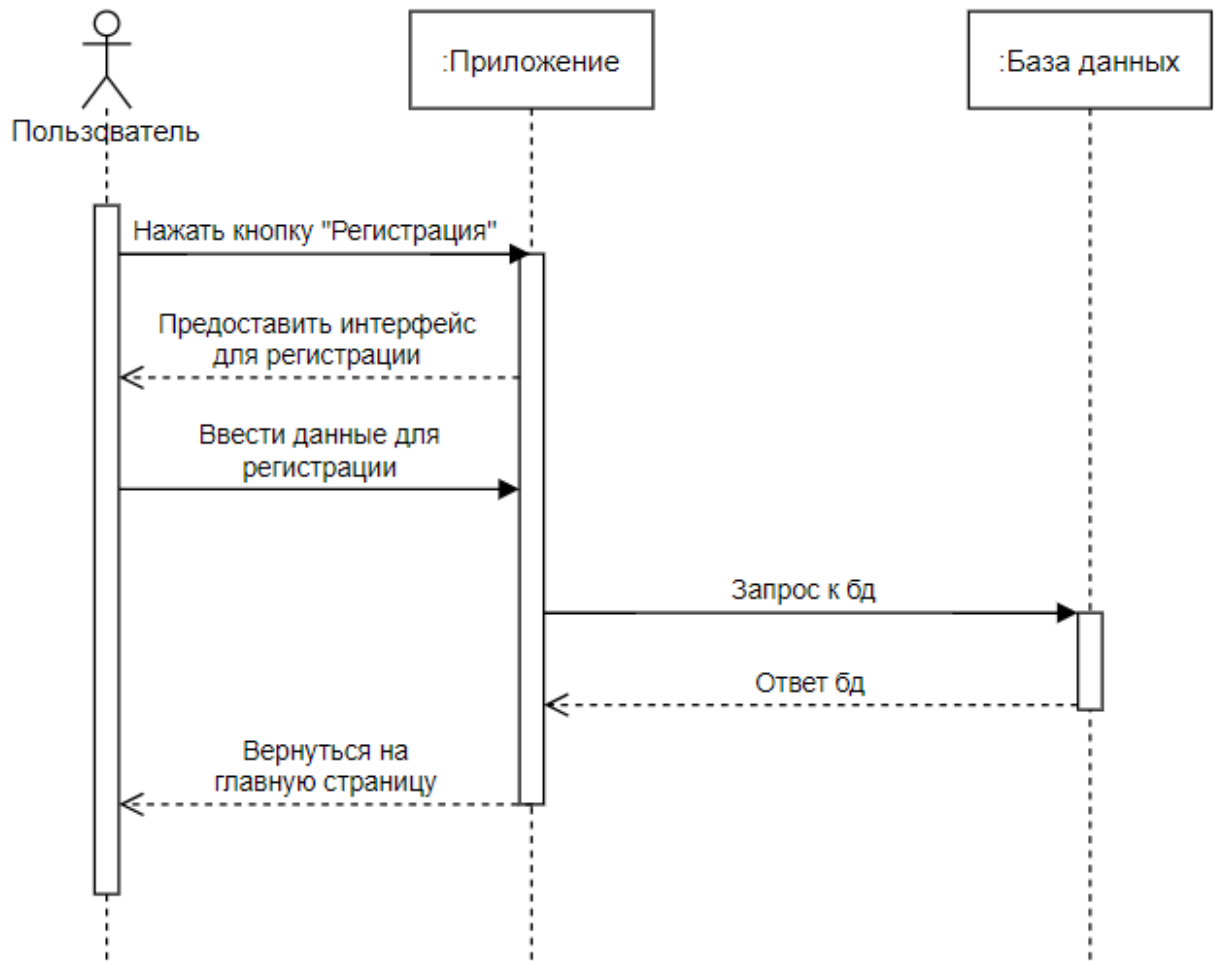


Рисунок 7 - Диаграмма последовательности для регистрации

Незарегистрированный пользователь после нажатия на кнопку регистрации получает от приложения форму для ввода данных регистрации. После отправки этих данных приложение делает запрос к БД и в случае успешного ответа (т.е. данные введены корректно и не совпадают с существующим) пользователь получает зарегистрированный аккаунт.

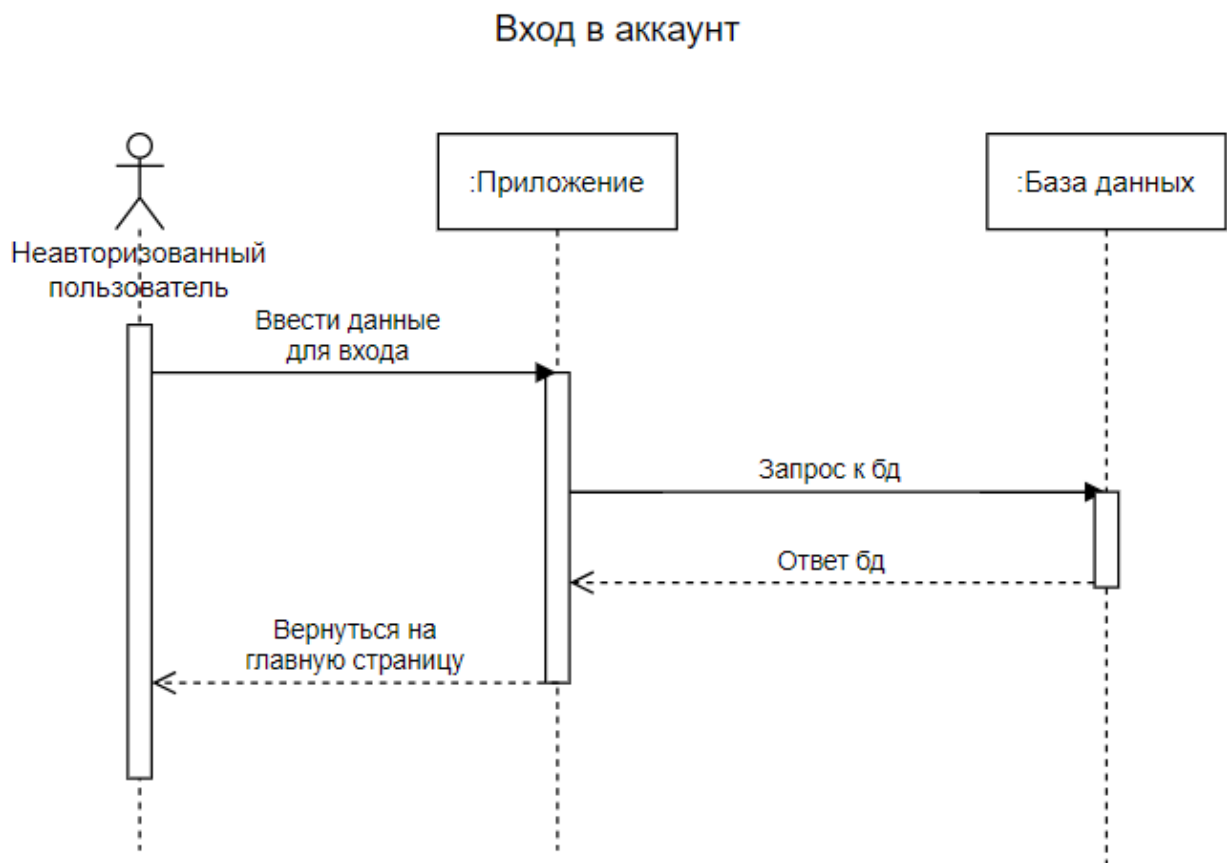


Рисунок 8 - Диаграмма последовательностей для авторизации

Неавторизованный пользователь вводит данные для входа (логин и пароль) в форму приложения, которое отправляет запрос в БД и в случае положительного ответа позволяет пользователю войти в приложение (с возвратом на главную страницу), в противном случае будет предложено ввести данные заново.

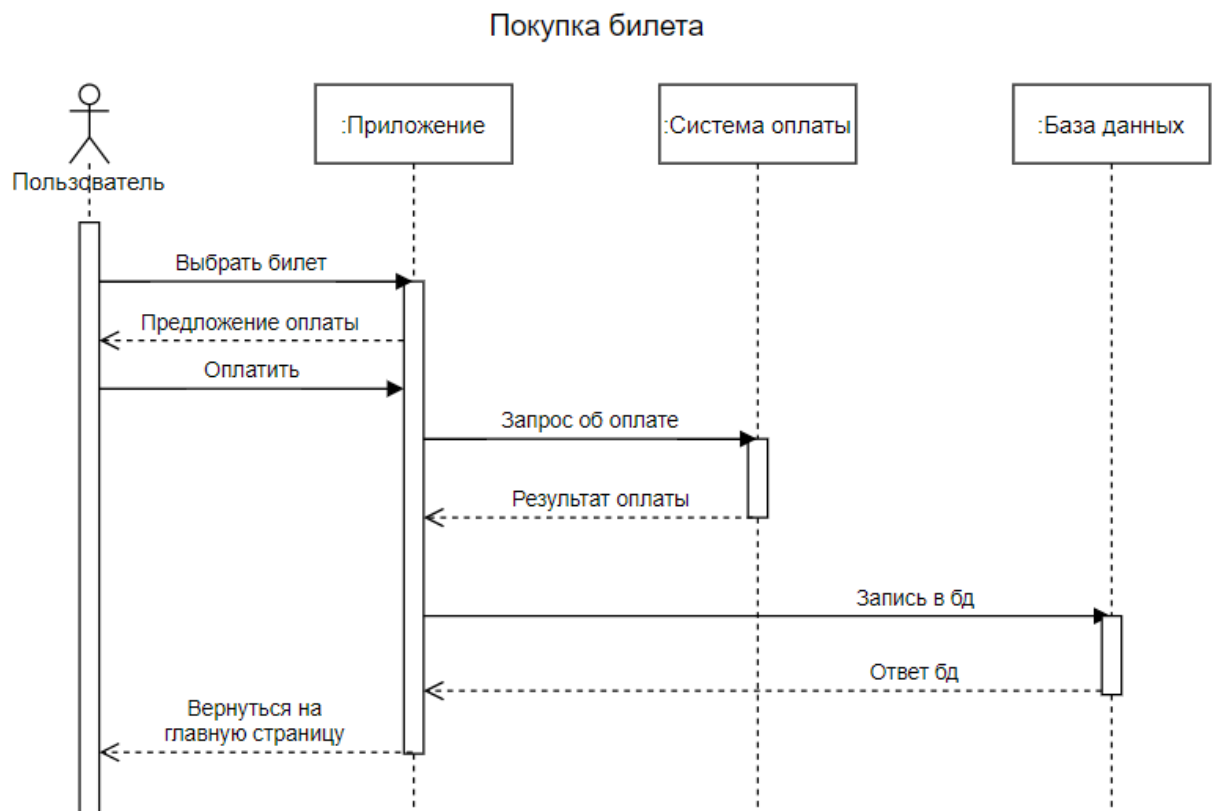


Рисунок 9 - Диаграмма последовательностей для покупки билета

Пользователь выбирает необходимые билеты, после чего приложение предлагает пользователю оплатить покупку. Пользователь вводит данные для оплаты в приложение, которое посылает запрос в систему оплаты. В случае положительного ответа в БД записывается информация о приобретенных билетах.

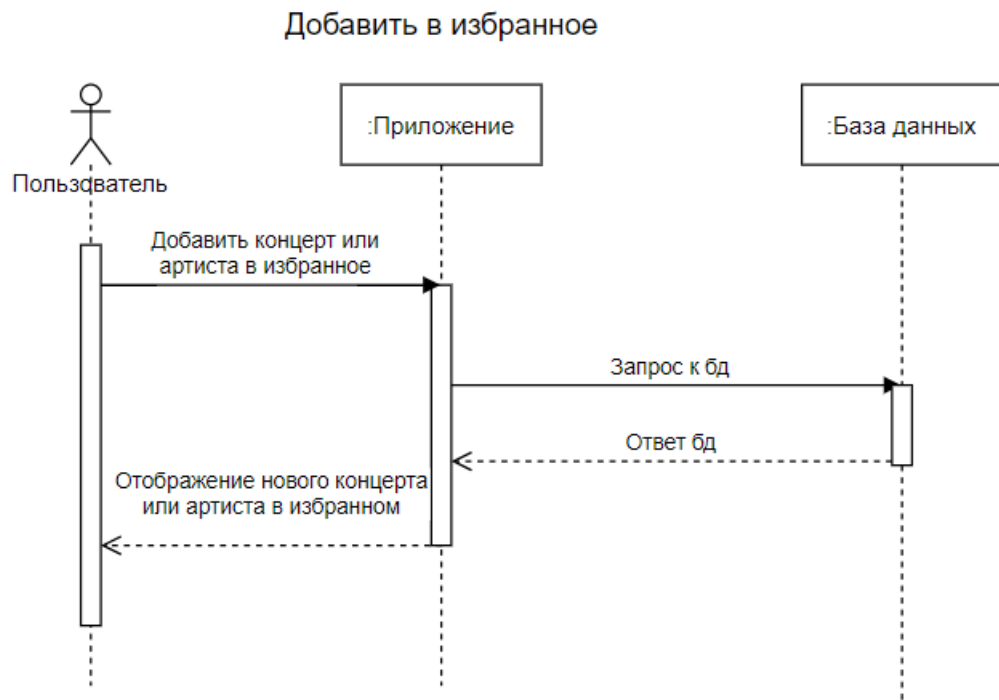


Рисунок 10 - Диаграмма последовательностей для добавления элемента в избранное

Пользователь выбирает концерт или артиста, чтобы добавить в список избранного. После нажатия на соответствующую кнопку приложения производится запрос на запись данной информации в БД в случае удачного ответа запись производится.

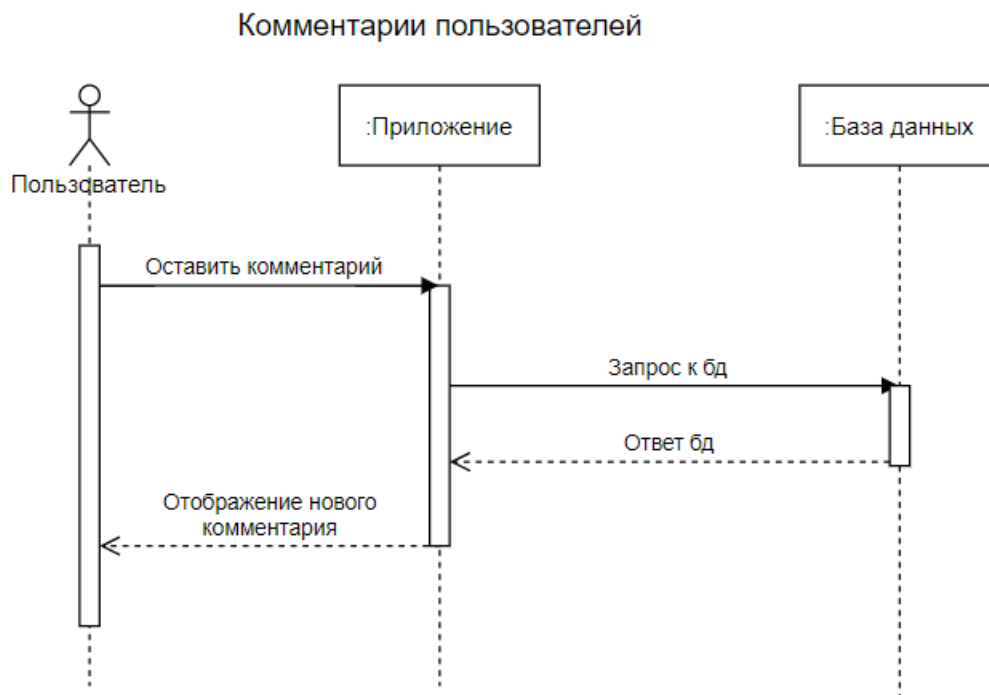


Рисунок 11 - Диаграмма последовательностей для добавления комментариев

Пользователь вводит текст комментария в соответствующем поле приложения и выбирает оценку для выбранного концерта/артиста. После нажатия на кнопку отправки комментария приложение делает запрос на запись в БД и в случае успешного ответа комментарий добавляется.

4.5. Диаграмма взаимодействий

Данная диаграмма определяет возможные способы взаимодействия пользователей с системой. Позволяет видеть все взаимодействия запросов в системе, служит дополнением к диаграмме последовательностей

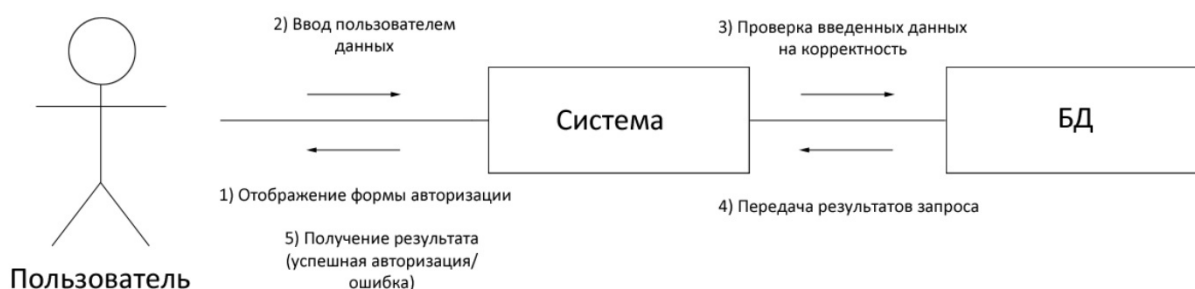


Рисунок 12 - Диаграмма взаимодействия для авторизации

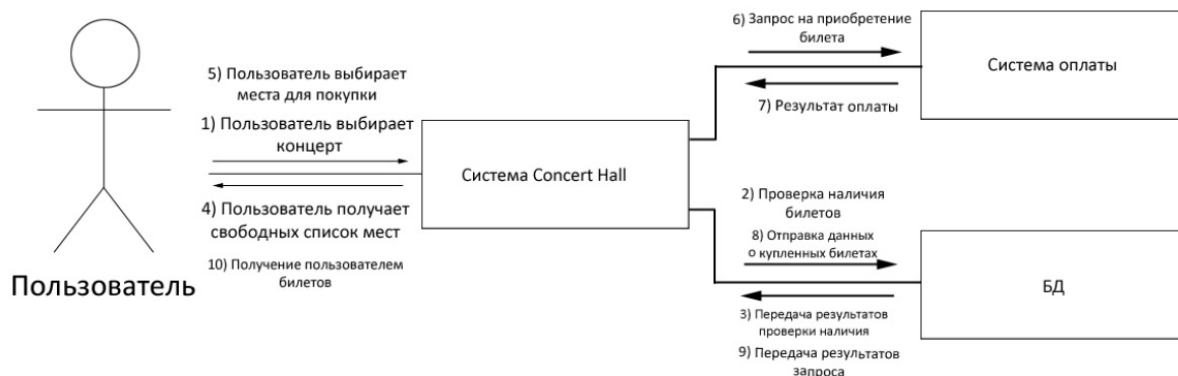


Рисунок 13 - Диаграмма взаимодействия для покупки билета

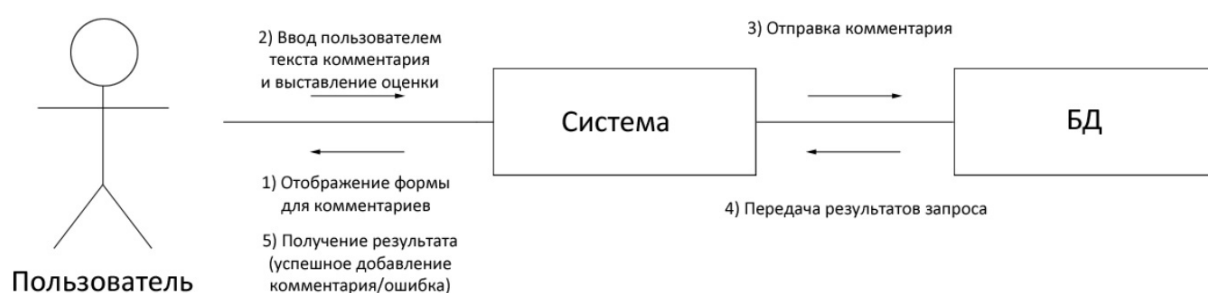


Рисунок 14 - Диаграмма взаимодействия для добавления комментария

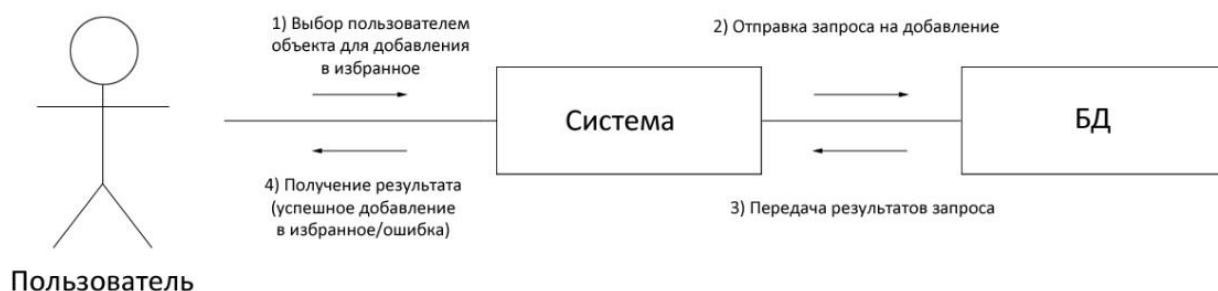


Рисунок 15 - Диаграмма взаимодействия для добавления элемента в избранное

4.6. Диаграмма состояний

Данные диаграммы описывают переход объектов из одних состояний в другие.

Диаграмма состояния для пользователя

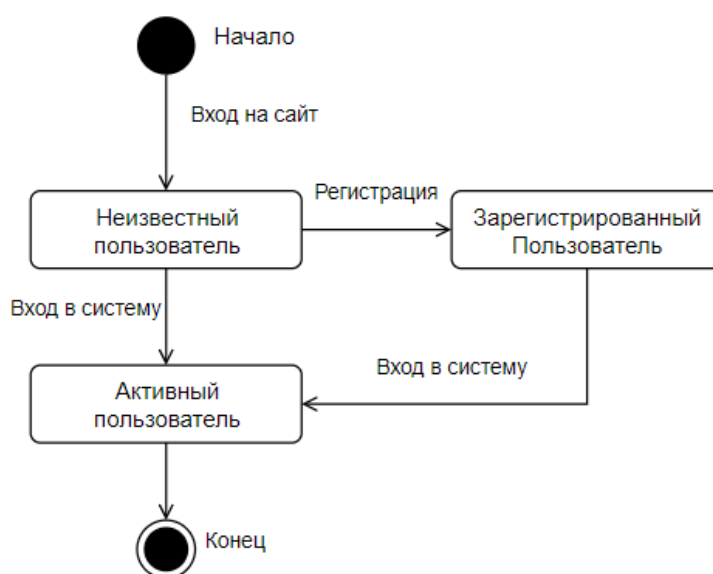


Рисунок 16 - Диаграмма состояний для авторизации

Диаграмма состояния для концерта

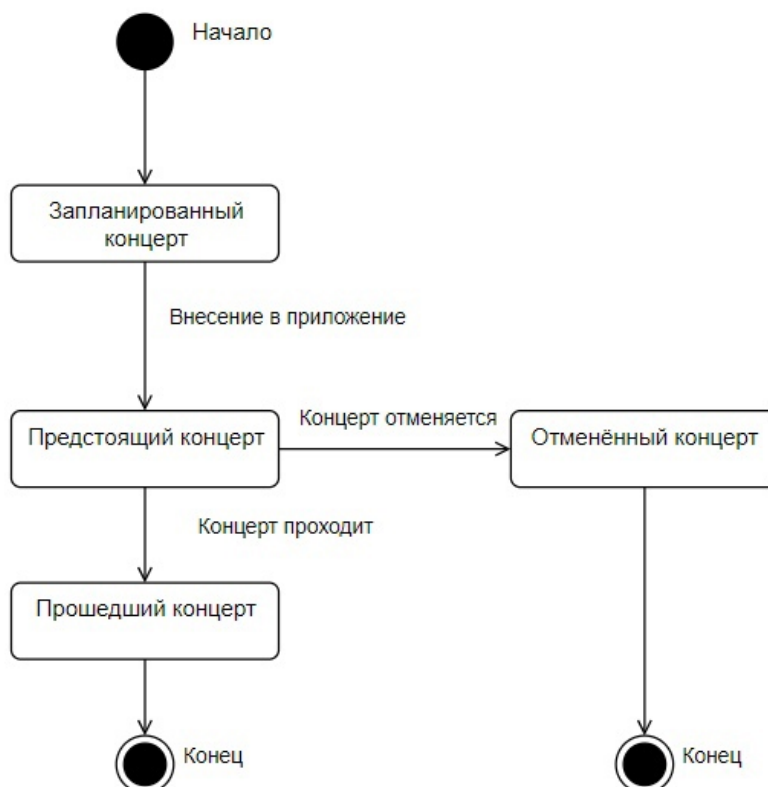


Рисунок 17 - Диаграмма состояний для концерта

Диаграмма состояния для билета



Рисунок 18 - Диаграмма состояний для билета

4.7. Диаграмма активности

Диаграмма активности описывает возможную последовательность объектов системы. Данная диаграмма показывает, что пользователь может, не авторизовавшись, либо покинуть приложение, либо купить билет, либо авторизоваться. Авторизованный пользователь может помимо покупки билетов также оставлять комментарии к концертам и артистам, а также добавлять их в свой список избранного.

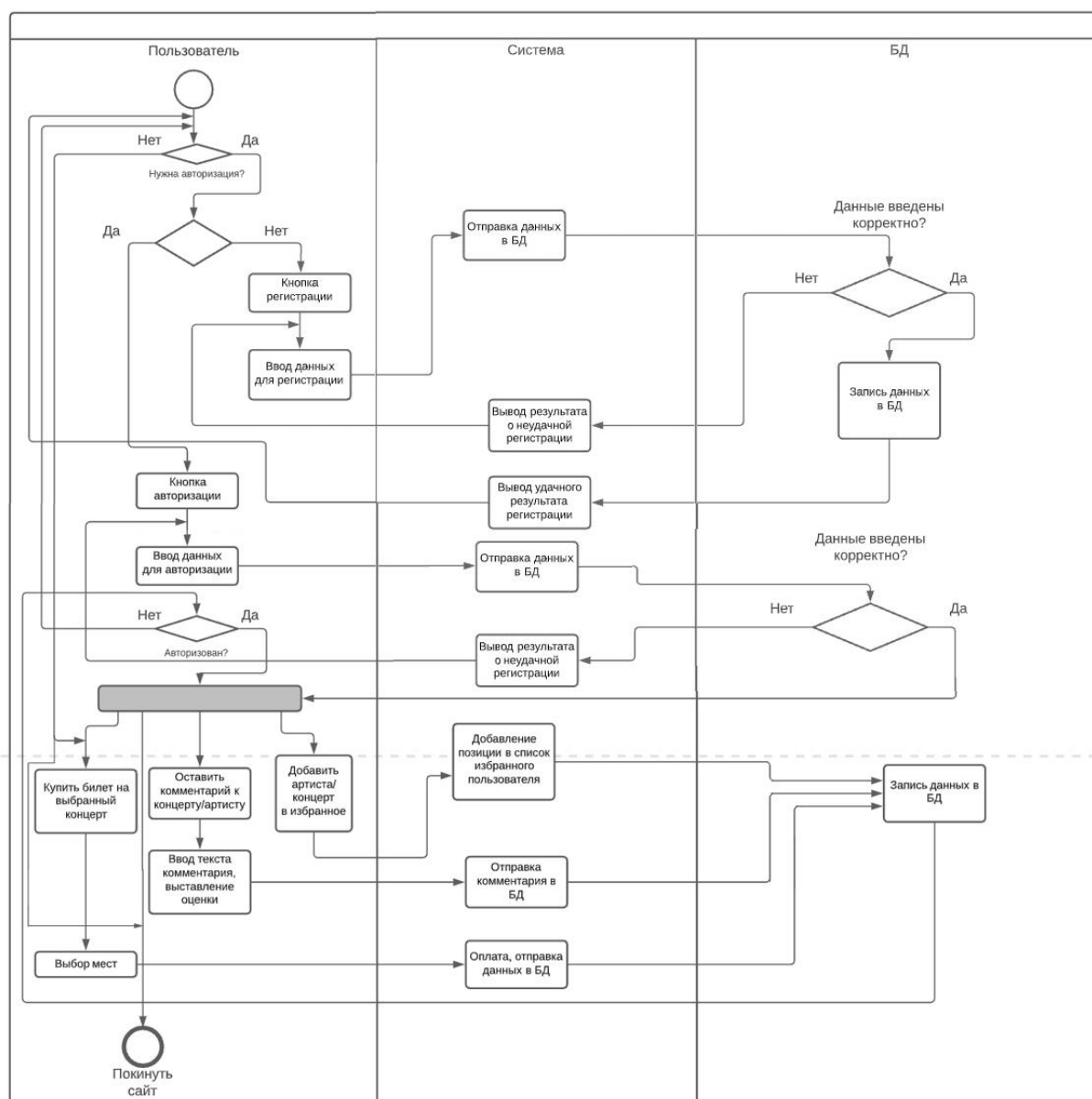


Рисунок 19 - Диаграмма активности для приложения

4.8. Диаграмма развертывания

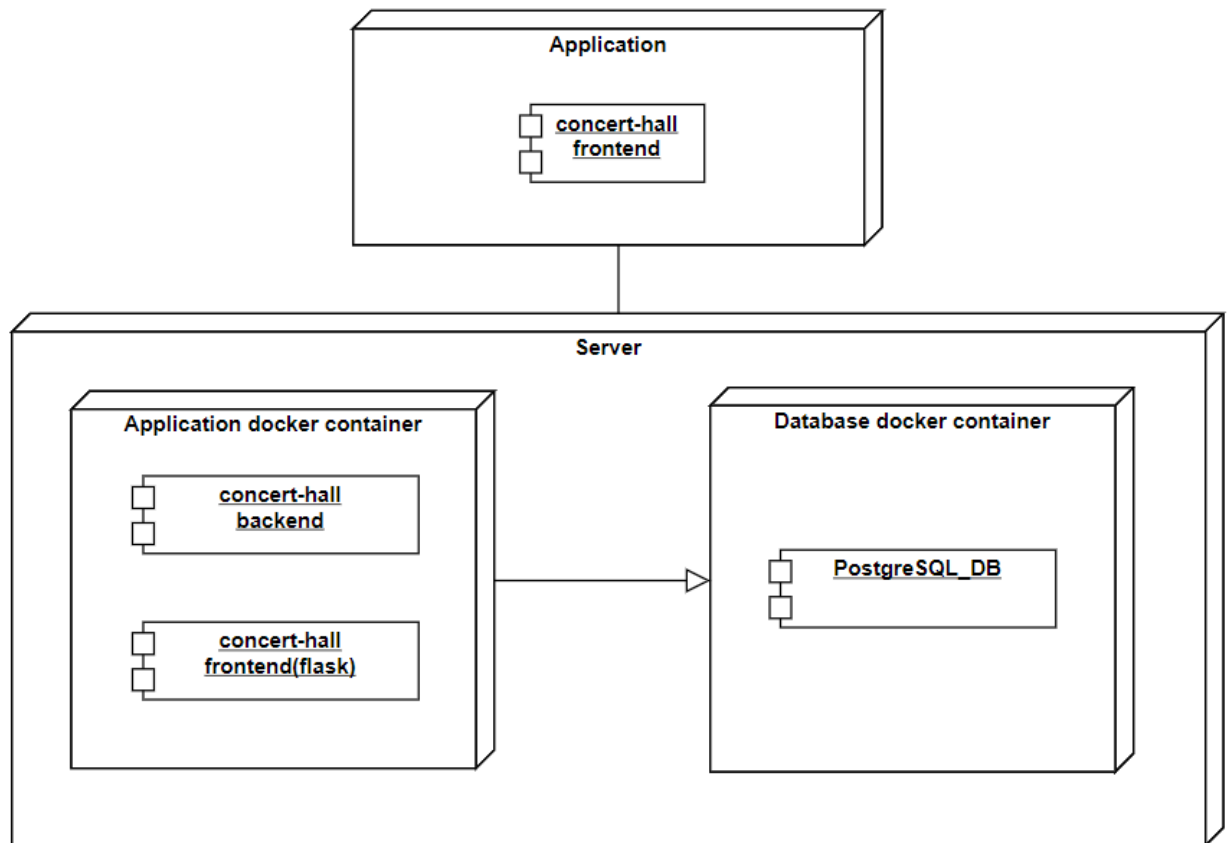


Рисунок 20 - Диаграмма развертывания для приложения

Данная диаграмма определяет аппаратные компоненты, их функционирование, а также их взаимодействие друг с другом.

4.9. Модель IDEF0

IDEF0 модель отображает структуру и функции приложения, а также потоки информации и материальных объектов, преобразуемые этими функциями.

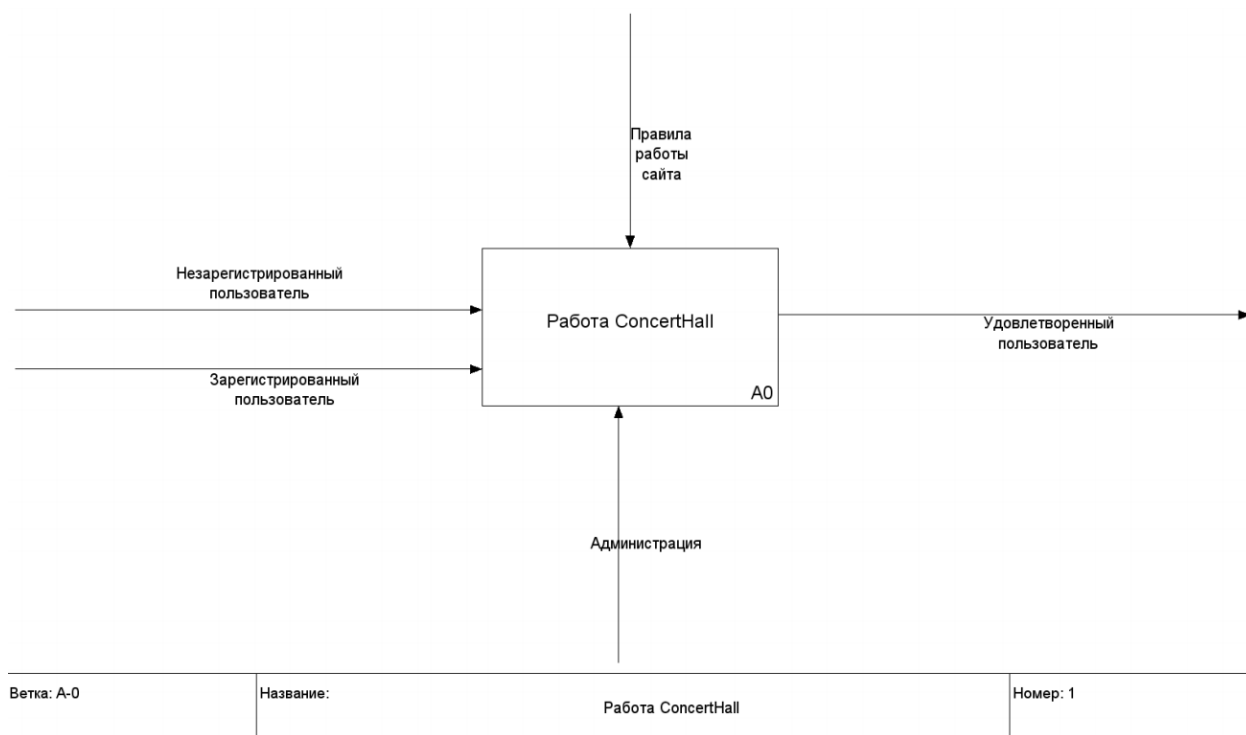


Рисунок 21 - Модель IDEF0 – уровень A-0

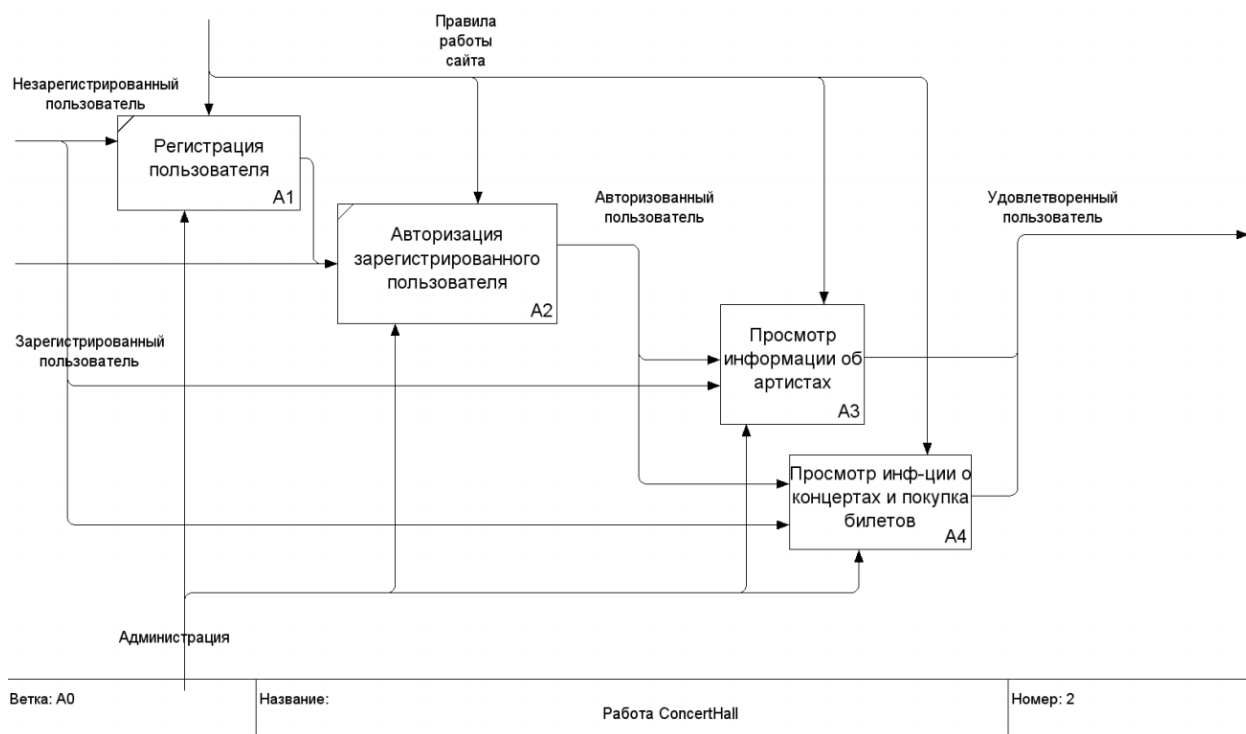


Рисунок 22 - Модель IDEF0 – уровень A0

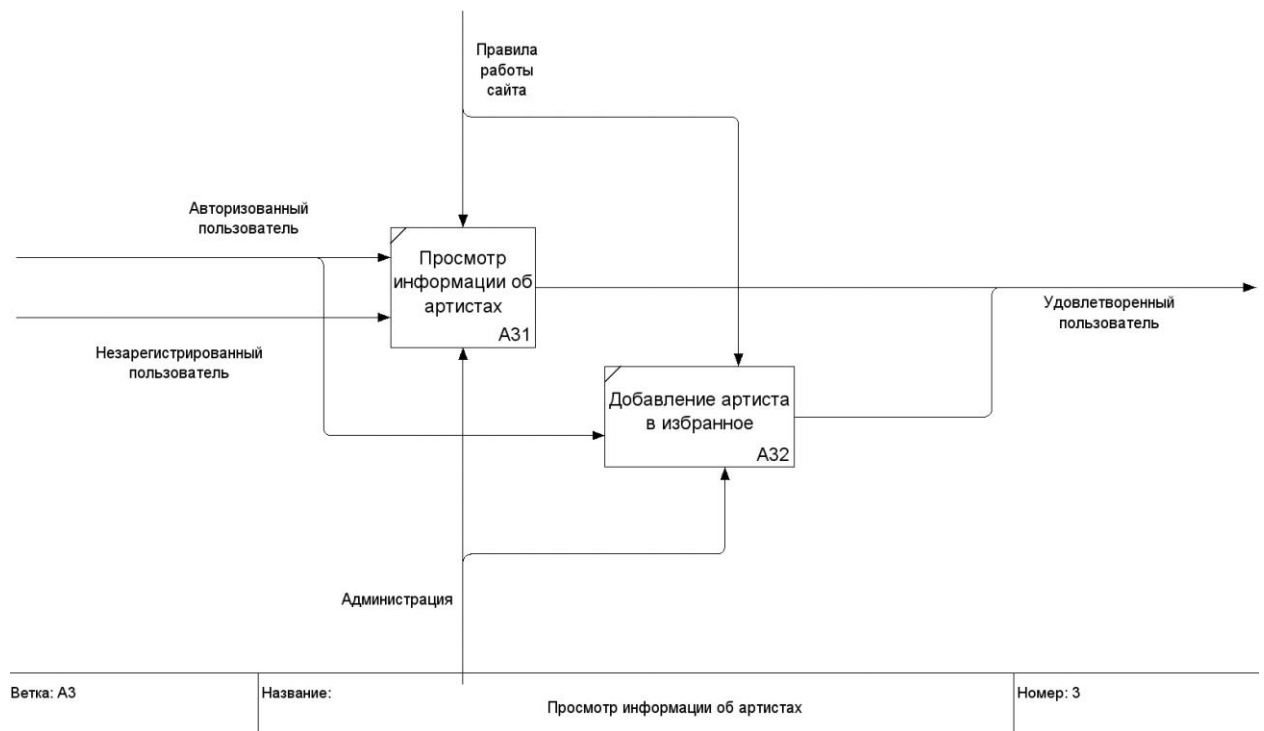


Рисунок 23 - Модель IDEF0 – уровень A3 (просмотр информации об артистах)

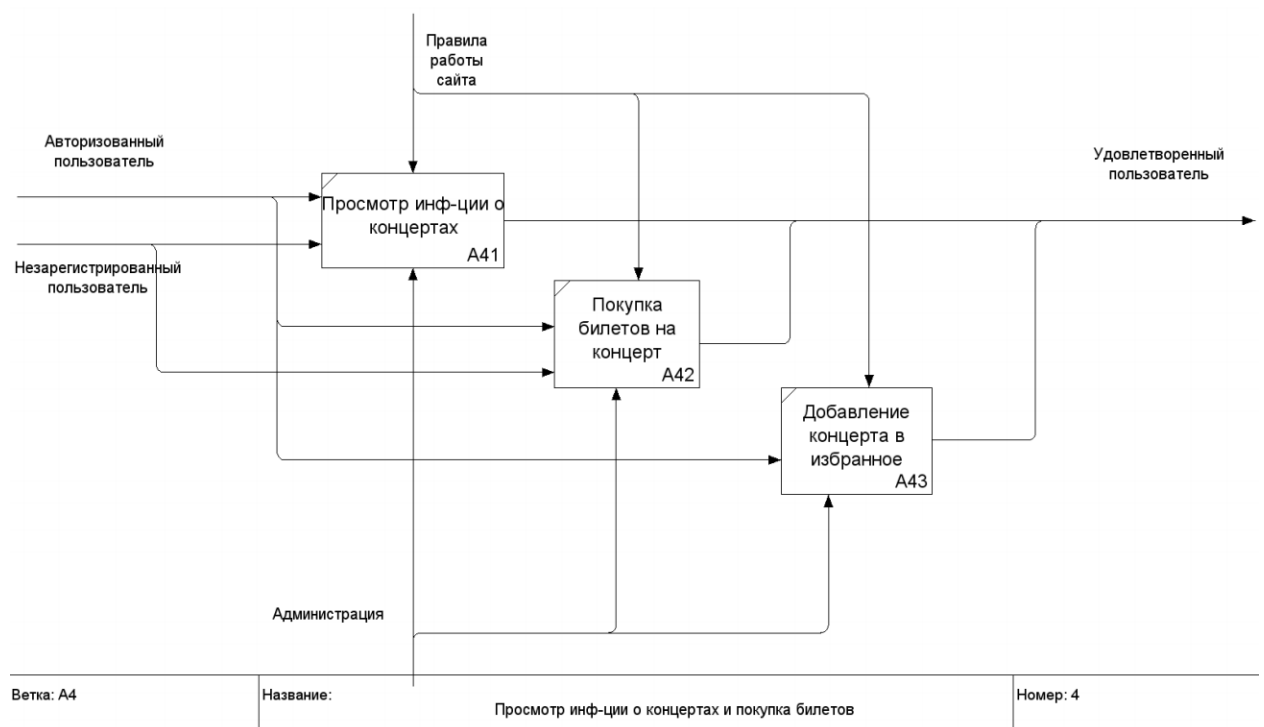


Рисунок 24 - Модель IDEF0 – уровень A4 (просмотр информации о концертах и покупка билетов)

4.10. ER-диаграмма

Данная диаграмма является представлением базы данных в виде наглядных графических диаграмм.

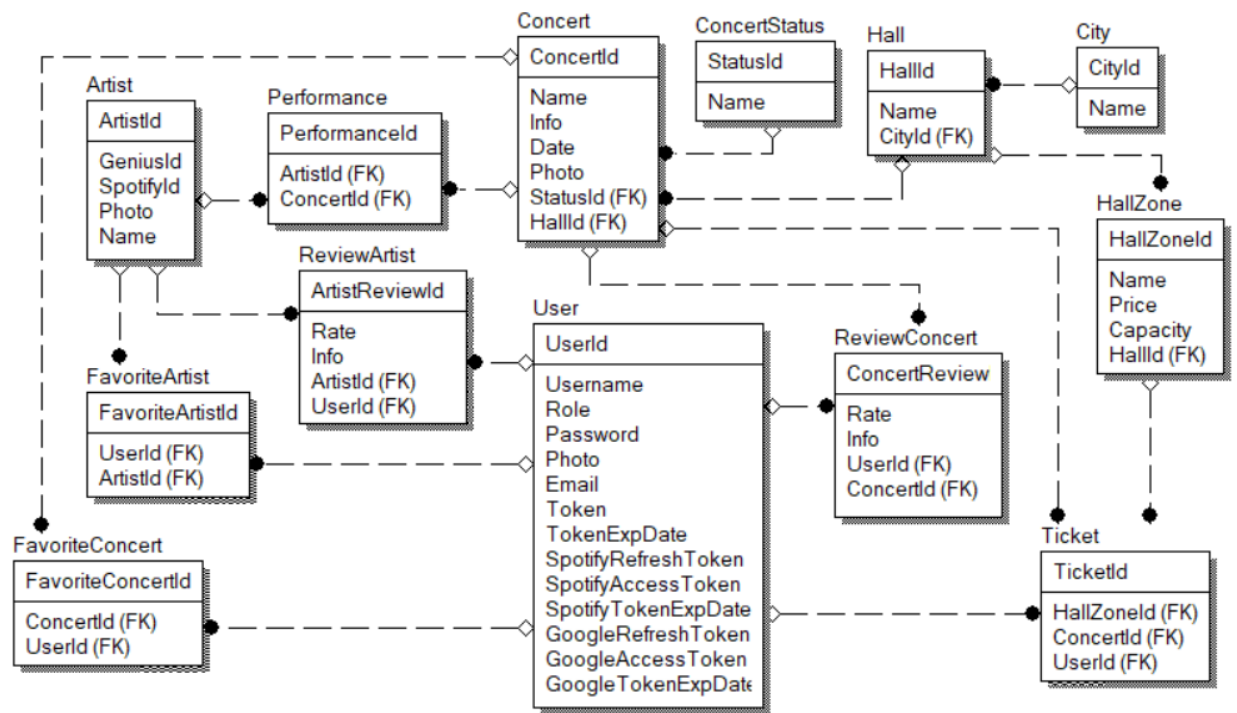


Рисунок 25 - ER-диаграмма проекта

5. Реализация

5.1. Средства реализации

В качестве средств реализации были использованы такие как:

- Язык программирования Python по причине большого количества библиотек, позволяющих реализовать требуемую функциональность приложения;
- Фреймворк Flask для языка Python, позволяющий создавать веб-приложения. Выбран был по причине большого количества пакетов-расширений, которые дают возможность реализовать необходимые в ходе решения задачи элементы приложения;
- База данных PostgreSQL по причине своей объектно-реляционной модели, а также большому количеству графических интерфейсов взаимодействия с ней, к примеру, IntelliJ Idea Datagrip;
- Шаблонизатор Jinja2, значительно упрощающий верстку страниц веб-приложения;
- Google OAuth 2.0, используемый для анонимного и надежного подключения аккаунта пользователя разрабатываемого веб-приложения к Google Календарь;
- Spotify OAuth 2.0, используемый для анонимного подключения аккаунта пользователя разрабатываемого веб-приложения к его аккаунту в сервисе Spotify.

Для развертывания приложения использованы платформа Heroku и программное обеспечение Docker, предназначенное для автоматизации развертывания приложения.

5.2. Frontend-составляющая приложения

Клиентская часть приложения реализована на языке HTML с использованием языка Javascript и библиотеки JQuery. Для оформления

страниц использовался фреймворк Bootstrap. В блоке комментариев для задания рейтинга рецензии используется плагин Bootstrap Star Rating от Krajee.

Также в клиентской части приложения предусмотрена проверка корректности ввода данных, несмотря на наличие ее в серверной составляющей. Сделано это было для дополнительной защиты.

5.3. Backend-составляющая приложения

Серверную часть приложения можно разделить на две части: часть, ответственная за работу с API, и часть, ответственная за представление и наполнение клиентской части.

5.3.1. Работа с API

В данном приложении задействованы Genius API, Spotify API, а также Google Calendar API.

Genius API используется для получения информации об артистах. Зная имя артиста, можно получить его ID в базе данных Genius, а из нее информацию о нем, если такой артист вообще там есть.

Spotify API позволяет пользователю видеть в своем списке рекомендаций грядущие концерты тех артистов, на которых он подписан в сервисе Spotify, если таковые будут присутствовать. Также при помощи данного API в базу данных заносятся фотографии артистов.

Google Calendar API позволяет реализовать привязку аккаунта Google пользователя веб-приложения, чтобы он мог добавлять интересующие его грядущие концерты в свой календарь

5.3.2. Наполнение клиентской части

Для заполнения клиентской части используется фреймворк Flask, который позволяет заполнять определенные шаблоны страниц необходимыми в текущий момент данными (к примеру, страницы артистов и концертов). Для работы с шаблонами страниц также используется шаблонизатор Jinja2, с помощью которого была сделана базовая страница, благодаря чему на всех

страницах приложения используются одинаковые CDN и стили. Также при помощи Jinja2 на каждой странице есть одинаковые header и footer.

Также данная часть связывает клиентскую часть и API. Реализовано это следующим образом: пользователь из клиентской части посылает запросы к API, которые передают на сервер информацию в необработанном виде, которая уже на нем обрабатывается и передается в клиентскую часть.

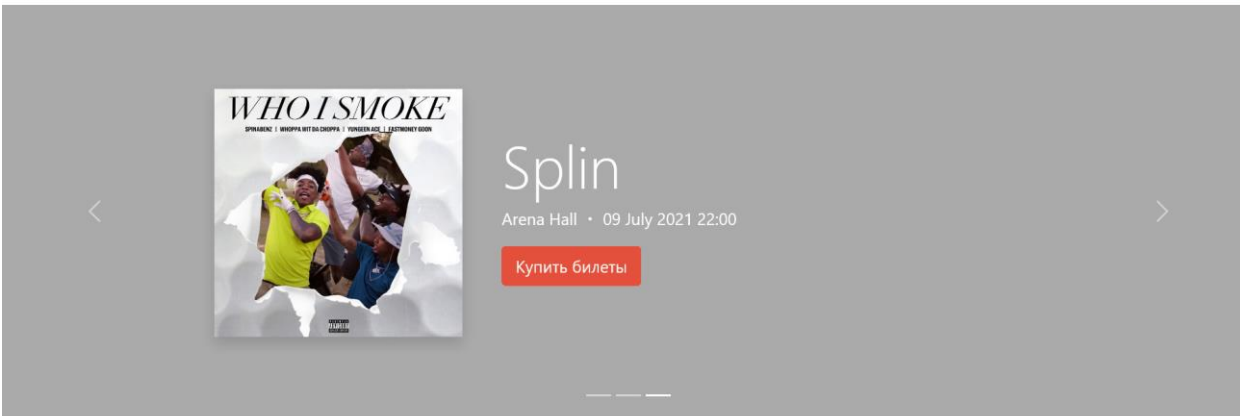
5.4. Система авторизации

При регистрации или авторизации пользователю выдается случайно генерируемый токен (случайный набор байтов, обладающий длиной набора 32 байта), который всегда является уникальным для каждого пользователя, позволяя однозначно его идентифицировать. Время жизни токена – 1 сутки. После окончания жизни данного токена его можно обновить, получив новый. Обновлять токен можно неограниченное число раз.

Пользовательские пароли подвергаются шифрованию методом SHA-256, в базе данных хранится зашифрованный его вариант в формате hash@salt. «salt» – это случайно сгенерированная на основе пароля строка, представляющая из себя 16 байт информации в текстовом виде. При применении «salt» пароль делится на две части, а «salt» на четыре. Из этого собирается строка. В начале мы используем первые четыре символа «salt», затем добавляем первую половину пароля, к этому добавляем символы «salt» с четвертого по двенадцатый, затем прибавляем вторую половину пароля, после чего используются последние четыре символа «salt». На полученную строку действуем алгоритмом шифрования SHA-256. Байты, получившиеся в результате алгоритма, переводятся в шестнадцатеричную строку. Данная строка является хешем, который и заносится в базу данных.

5.5. Описание функциональной части приложения и графического пользовательского интерфейса

На рисунке 26 изображена главная страница веб-приложения.



Грядущие концерты

Artist	Venue	Date and Time
Pharaoh	Arena Hall	20 June 2021 14:00
Slot	Aura	24 June 2021 13:30
Splin	Green Theater / Зеленый театр	09 July 2021 22:00
Хаски	Arena Hall	01 August 2021 22:00
Eluveitie	Arena Hall	04 September 2021 16:00
GSPD	Arena Hall	05 September 2021 13:00

Рисунок 26 - Главная страница

В том случае, если пользователь авторизовался и привязал к аккаунту в приложении аккаунт сервиса Spotify, он также может видеть на главной странице блок с концертами, предложенными на основе его подписок в сервисе (рисунок 27).

Рекомендованное



Рисунок 27 - Блок с рекомендациями Spotify

При помощи навигационной панели, содержащейся на всех страницах, неавторизованный пользователь может выбрать свое местоположение, а также перейти на страницы регистрации (рисунок 28) и авторизации (рисунок 29).

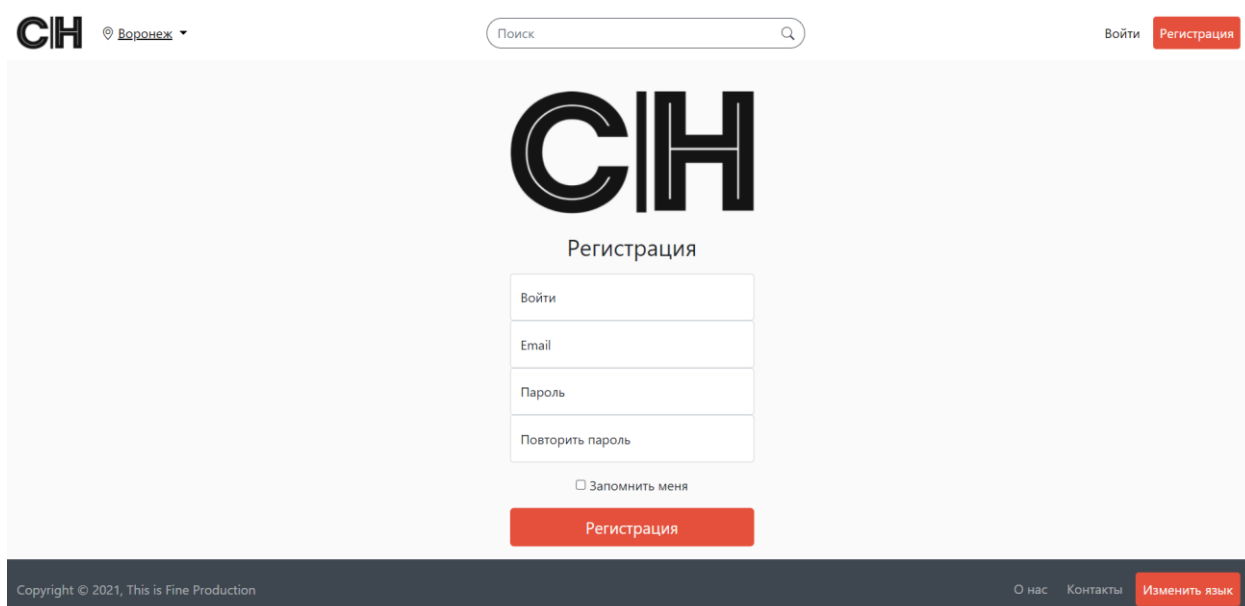
The registration page of the SN website. At the top, there is a navigation bar with the "SN" logo, a location dropdown menu set to "Воронеж", a search bar with the placeholder "Поиск", and links for "Войти" and "Регистрация". The main content area features a large "SN" logo and the heading "Регистрация". Below this is a registration form with four input fields: "Войти", "Email", "Пароль", and "Повторить пароль". There is a checkbox labeled "Запомнить меня" and a red "Регистрация" button. The footer contains copyright information "Copyright © 2021, This is Fine Production", links for "О нас" and "Контакты", and a red button for "Изменить язык".

Рисунок 28 - Страница регистрации

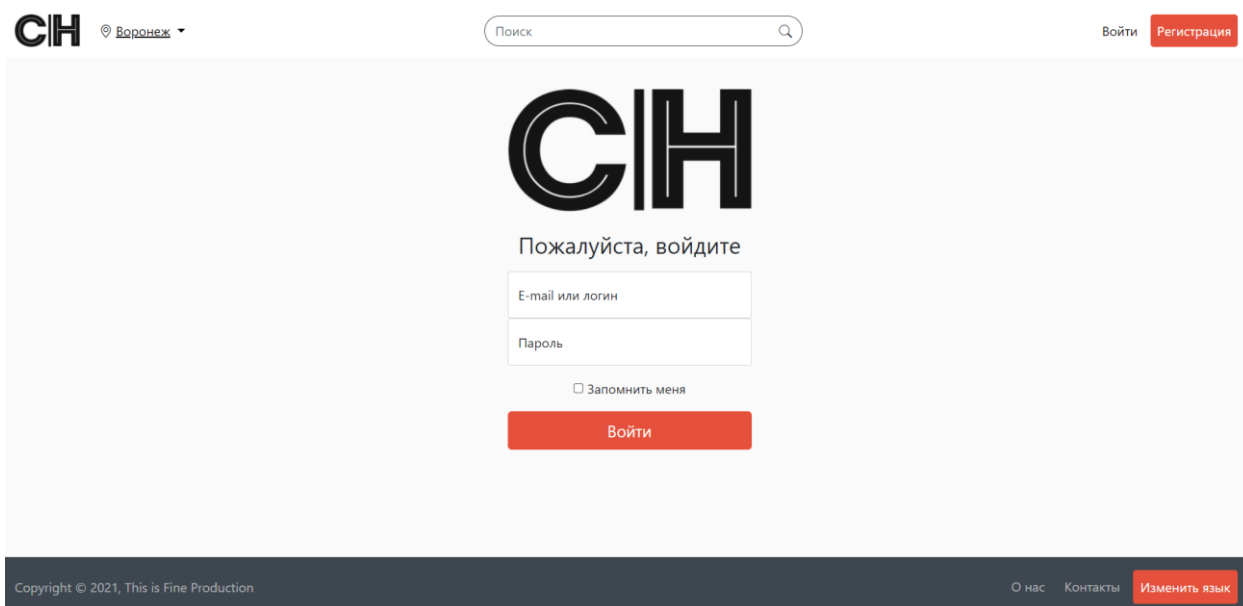


Рисунок 29 - Страница авторизации

Авторизованный пользователь может просмотреть список купленных им билетов, а также после клика на иконку профиля увидеть выпадающее меню (рисунок 30), из которого он может перейти на страницы избранного контента (концерты и артисты) и настроек профиля.

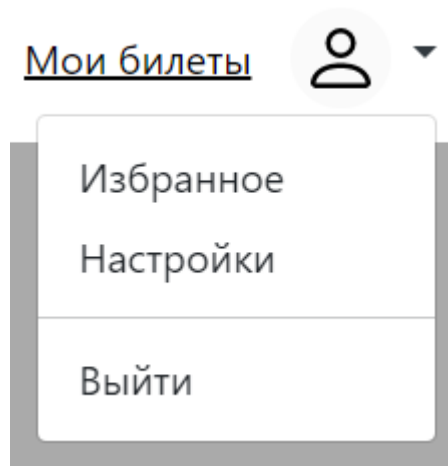


Рисунок 30 - Выпадающее меню авторизованного пользователя

На странице, изображенной на рисунке 31, пользователь может просматривать купленные им билеты на концерты.

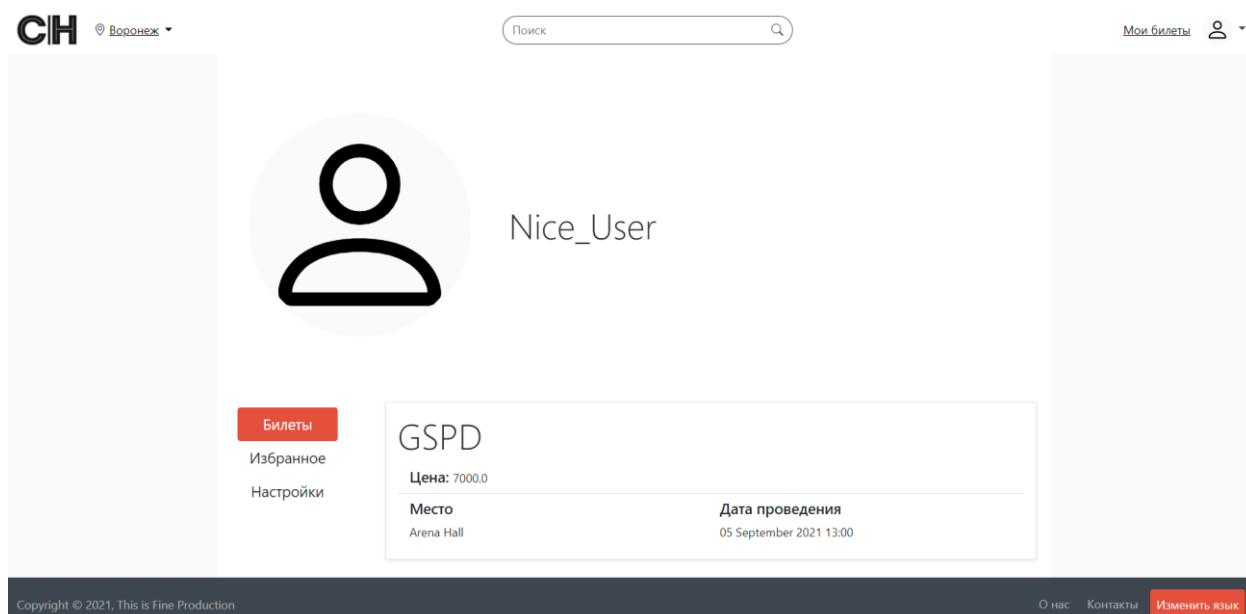


Рисунок 31 - Страница приобретенных пользователем билетов

На странице избранного (рисунок 32) пользователь может просмотреть концерты и артистов, которых он добавил в свой список избранного.

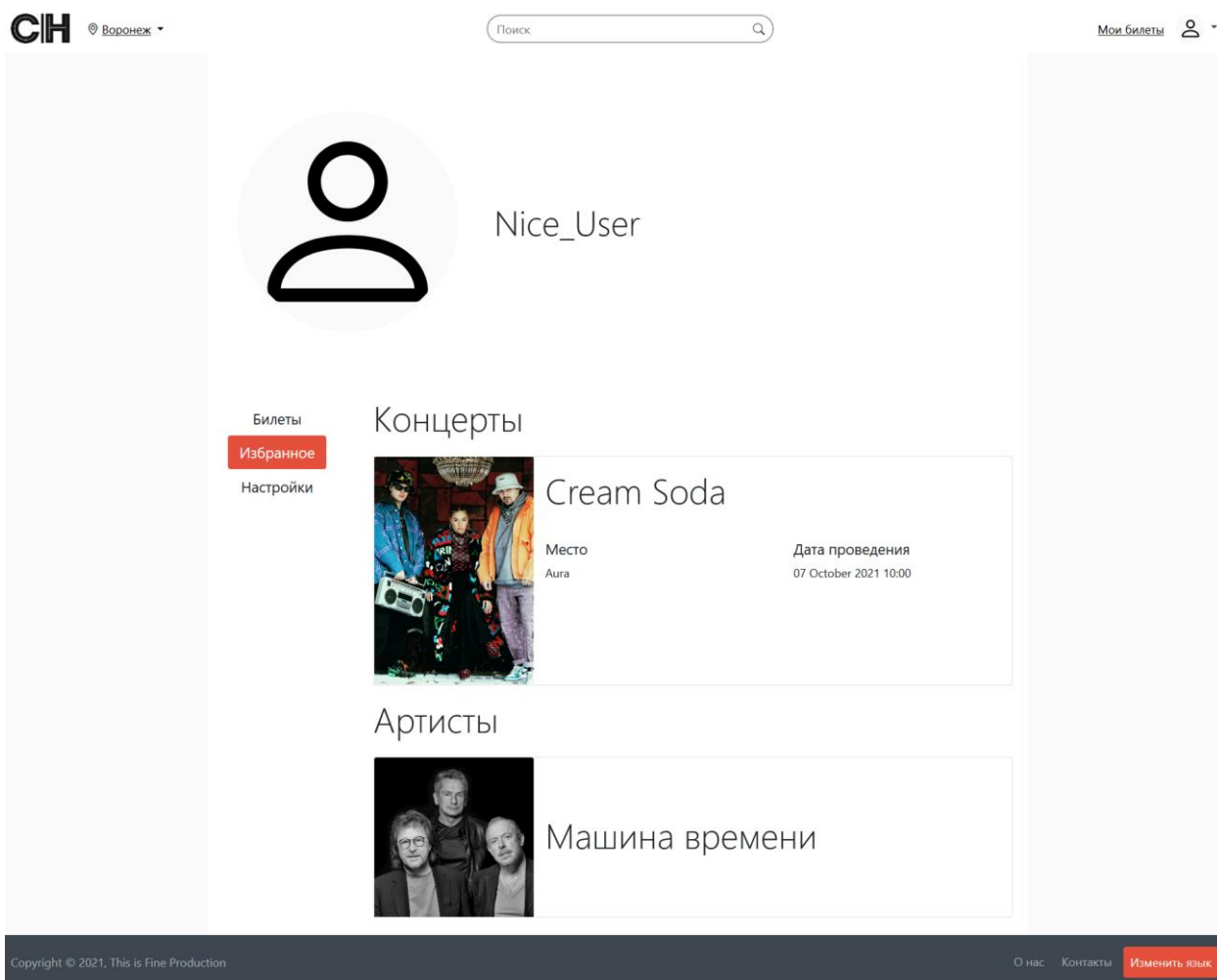


Рисунок 32 - Страница избранного пользователя

На странице настроек (рисунок 33) пользователь может подключить свои аккаунты в Spotify и Google, изменить свой аватар, а также сменить пароль.

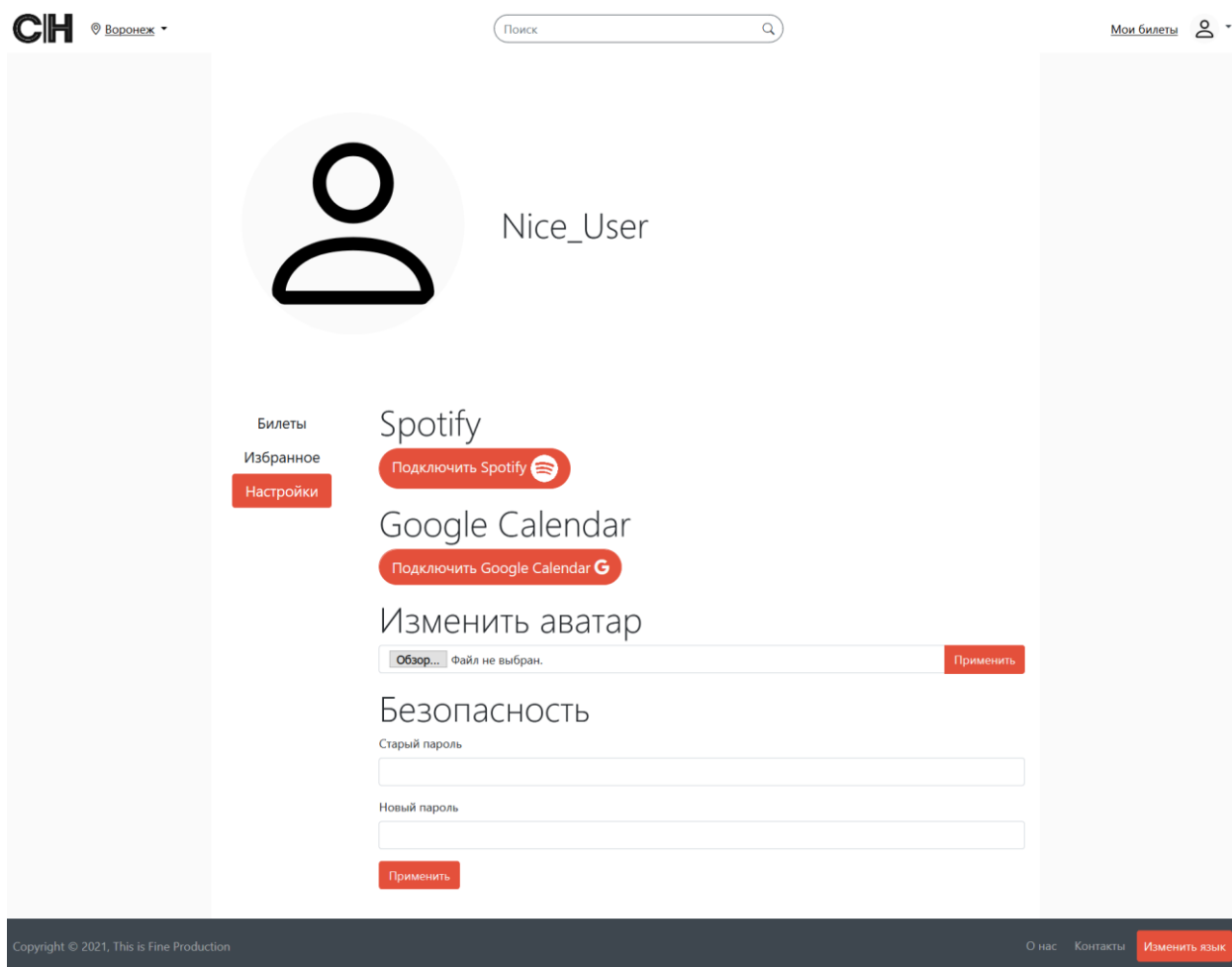


Рисунок 33 - Страница настроек пользователя.

На странице концерта (рисунок 34) пользователь может посмотреть информацию о концерте, место и дату проведения, список выступающих артистов, а также добавить данный концерт в свой список избранного. В случае, если концерт еще не завершился, пользователь может купить на него билет. Если концерт уже завершен, пользователь может оставить рецензию на него.

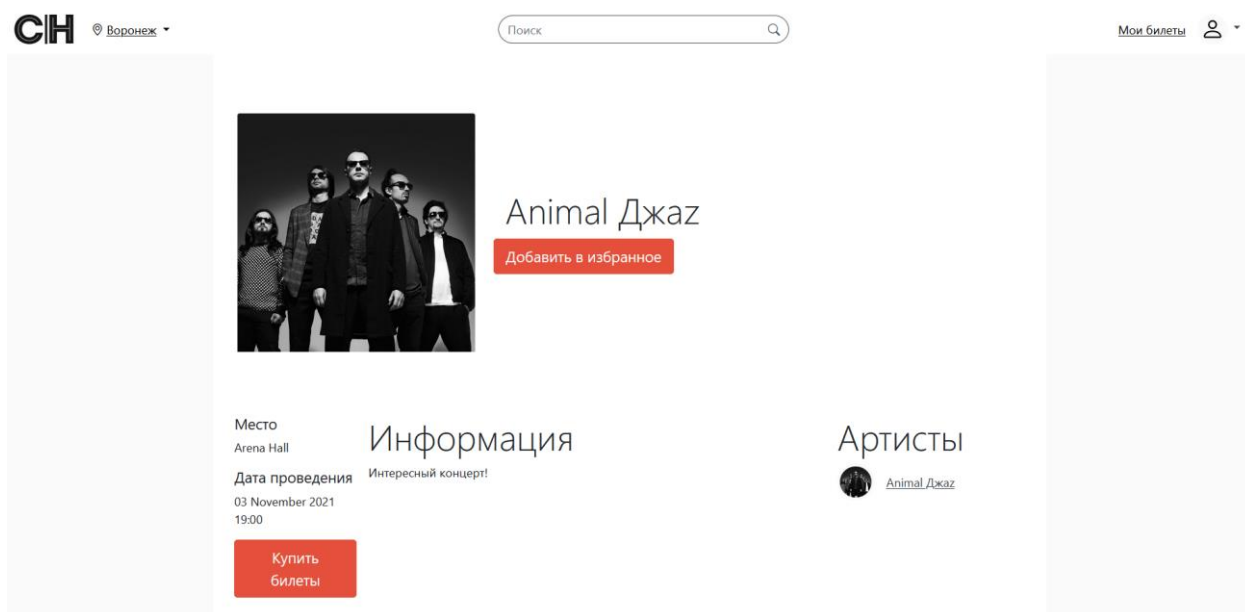


Рисунок 34 - Страница концерта

На странице покупки билетов (рисунок 35) пользователь может выбрать, в какой зоне зала он хочет приобрести билеты, а также их количество. Цены на билеты зависят от зоны в зале, количество зон зависит от конкретного концертного зала.

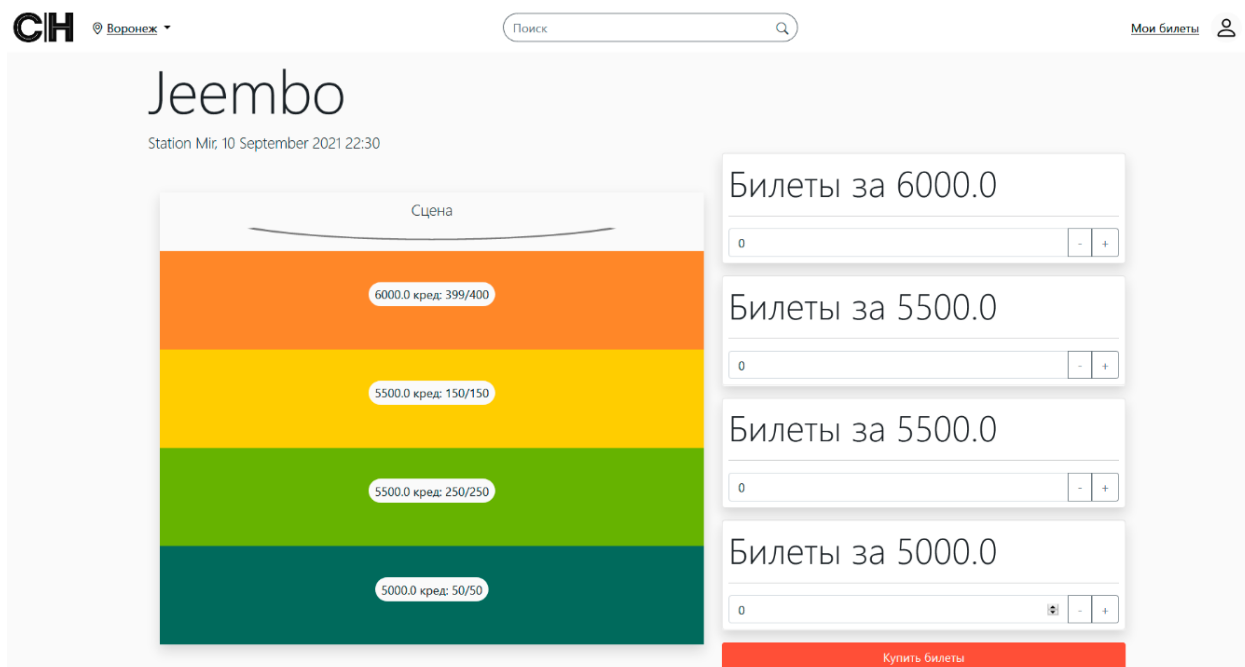


Рисунок 35 - Страница покупки билетов на концерт.

На странице артиста (рисунки 36-38) пользователь имеет возможность посмотреть список концертов с участием данного исполнителя (как

прошедших, так и грядущих), пользовательские рецензии на них, а также информацию о нем, полученную при помощи Genius API.

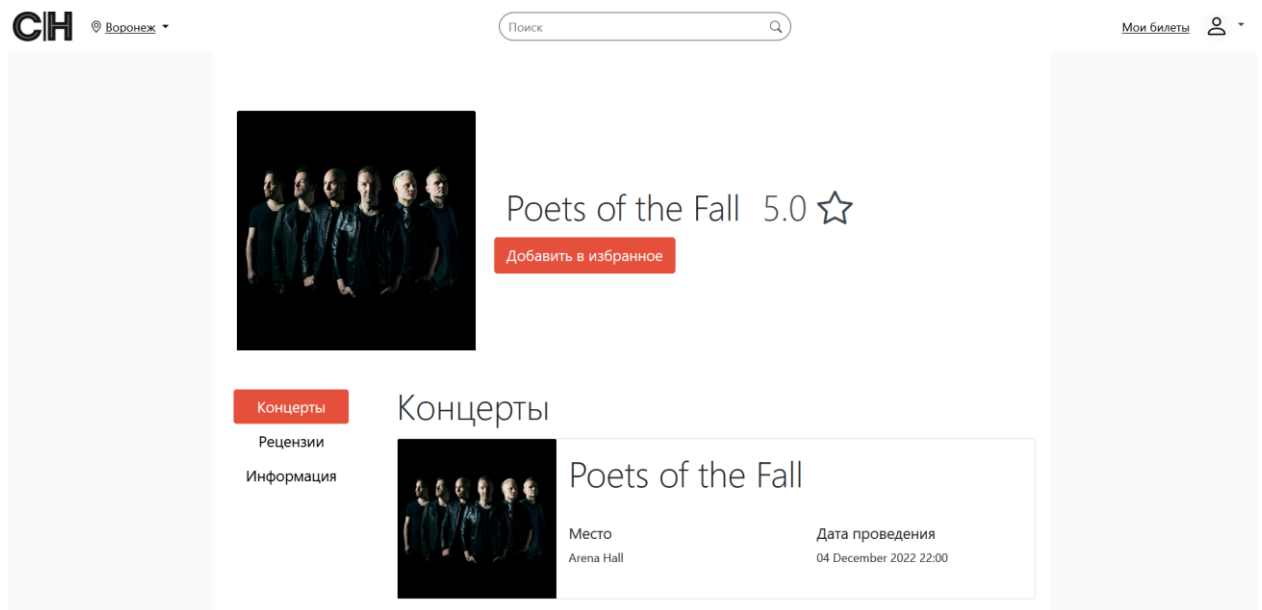


Рисунок 36 - Страница артиста (список концертов)

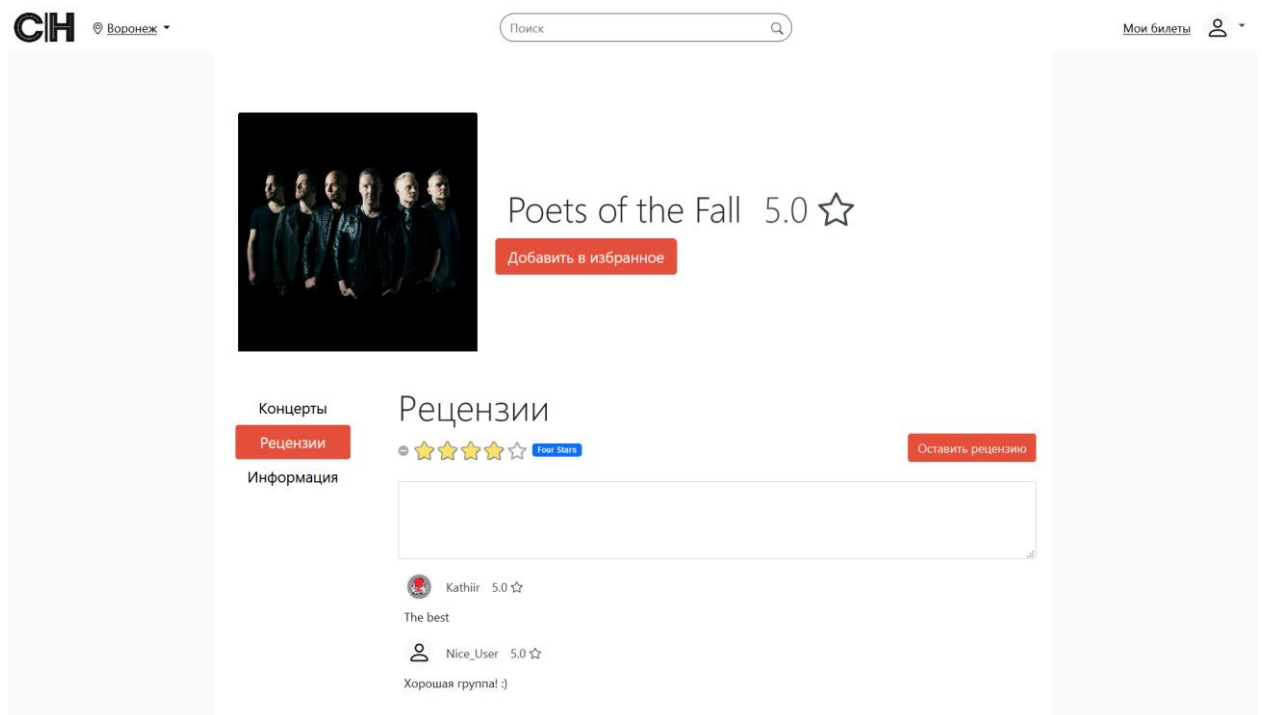


Рисунок 37 - Страница артиста (список рецензий)



Рисунок 38 - Страница артиста (информация)

В нижней части каждой страницы (footer) обнаружить кнопку переключения языка (русский/английский), а также ссылки на страницу с информацией о команде и контактные данные администрации (рисунок 39).

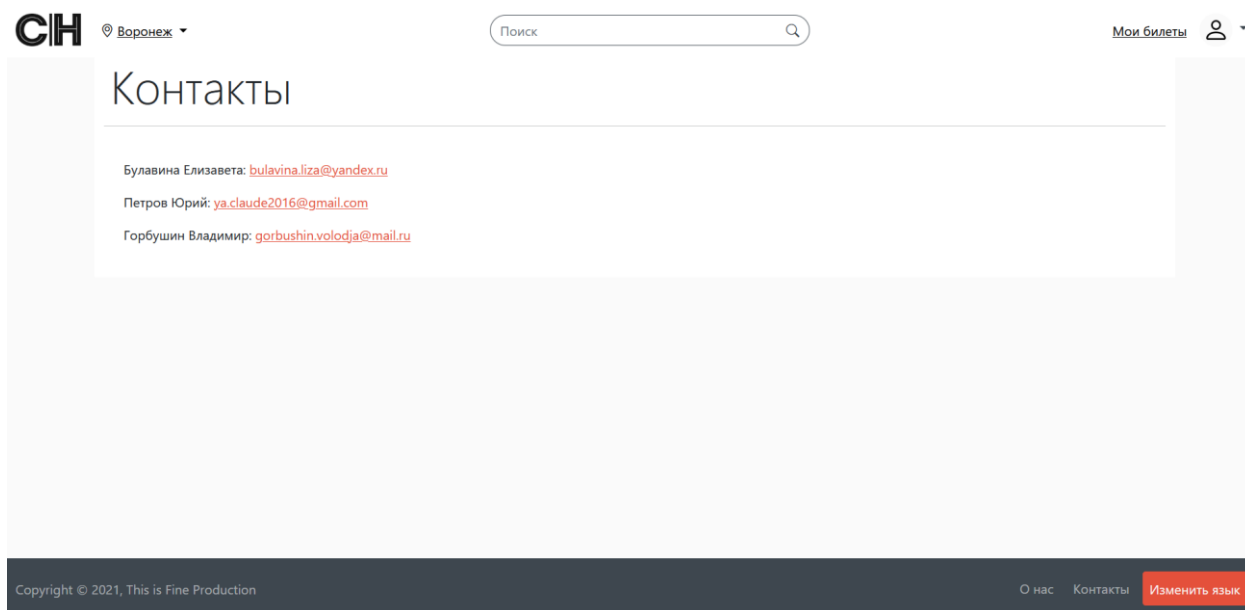


Рисунок 39 - Страница с контактной информацией

Как было упомянуто выше, пользователь может воспользоваться поиском по сайту через строку поиска в навигационной панели. Поиск может проводиться как среди артистов, так и среди концертов (рисунки 40-41).

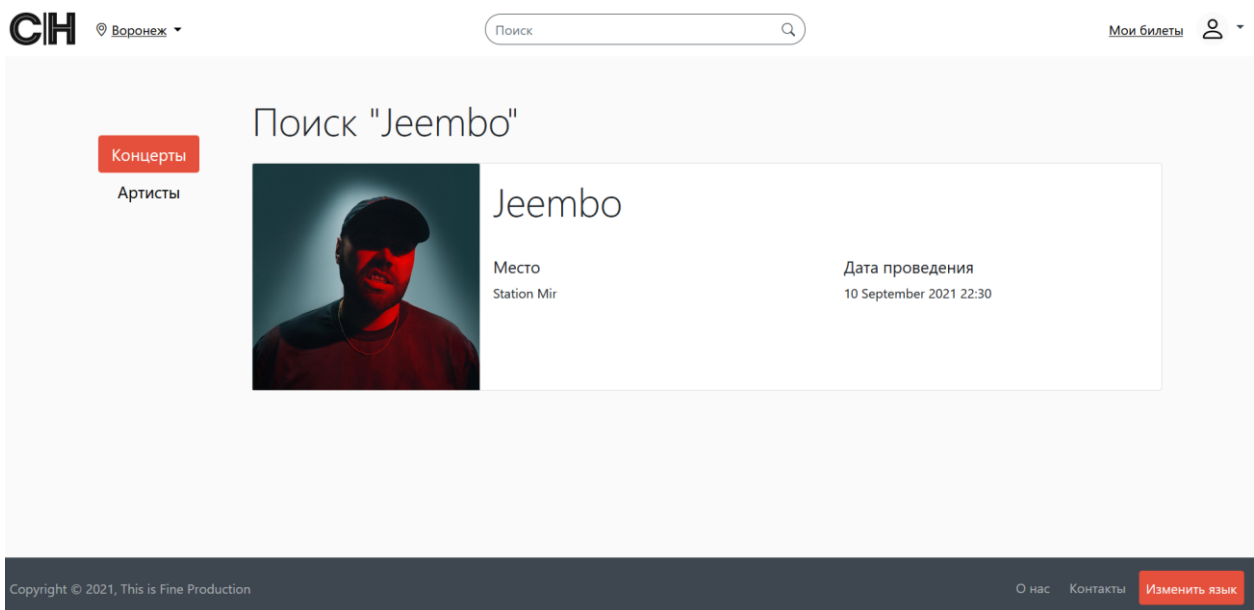


Рисунок 40 - Результаты поиска по концертам

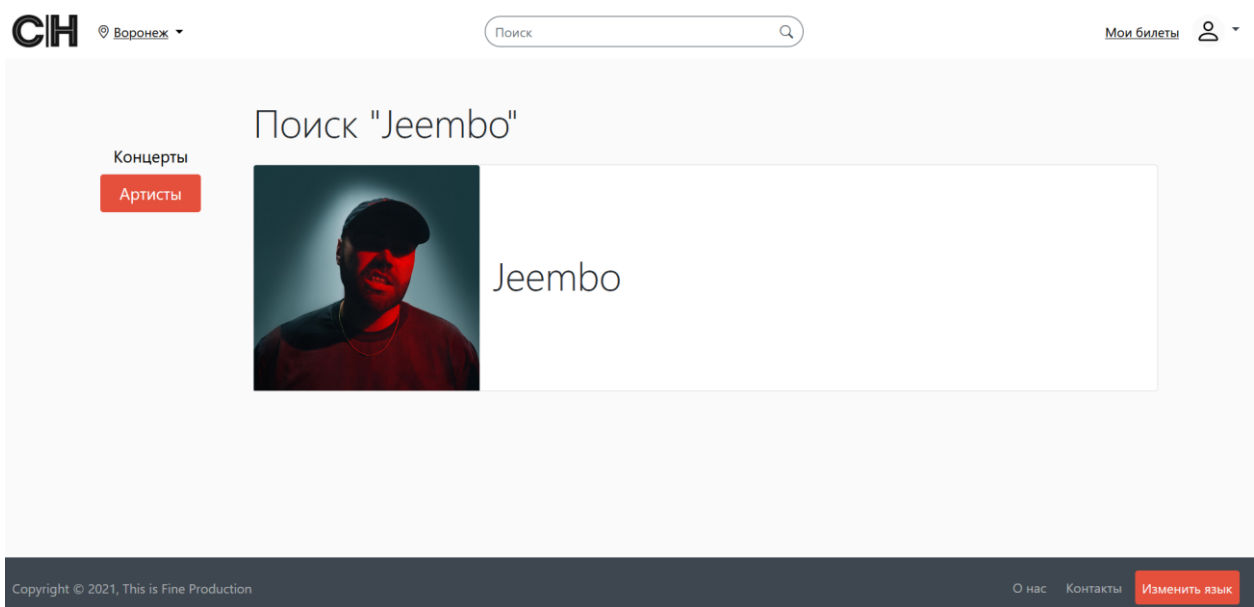


Рисунок 41 - Результаты поиска по артистам

6. Тестирование

После работ по созданию приложения было произведено тестирование его backend-сервисов.

Тип запроса	Функция	Результат тестирования
POST	Регистрация пользователя	Успешно
POST	Авторизация пользователя	Успешно
POST	Деавторизация пользователя	Успешно
POST	Смена пароля	Успешно
GET	Получение краткой информации о пользователе (через токен)	Успешно
GET	Получение информации о профиле пользователя (через токен)	Успешно
POST	Изменение аватара пользователя	Успешно
POST	Изменение токена пользователя	Успешно
DELETE	Удаление токена пользователя	Успешно
GET	Получение списка концертов по подпискам пользователя в Spotify (в случае привязки аккаунта)	Успешно
POST	Добавление концерта в Google Calendar (в случае привязки аккаунта)	Успешно
POST	Добавление элемента в список избранного	Успешно
DELETE	Удаление элемента из списка избранного	Успешно
GET	Проверка элемента на нахождение в списке избранного	Успешно
GET	Получение списков избранных артистов и концертов	Успешно

POST	Покупка билетов	Успешно
GET	Получение всей информации о купленных пользователем билетах	Успешно
GET	Получение всех рецензий для концерта или артиста	Успешно
POST	Добавление рецензии на артиста	Успешно
POST	Добавление рецензии на концерт	Успешно
GET	Получение артиста по его ID	Успешно
GET	Поиск артиста по его имени	Успешно
GET	Получение концерта по ID	Успешно
GET	Поиск концерта по его названию	Успешно
GET	Получение списка всех концертов	Успешно
GET	Получение всей информации о концертном зале	Успешно
GET	Получение всей информации о билетах на концерт	Успешно

Таблица 1 - Список проведенных тестов запросов

Заключение

В ходе выполнения данной курсовой работы были выполнены следующие задачи:

- Собрана необходимая информация об исследуемой предметной области;
- Проведен анализ полученных данных;
- Созданы диаграммы, отражающие основные аспекты приложения;
- Разработано веб-приложение;
- Проведено тестирование веб-приложения;
- Приложение было размещено на сервере.

Цель курсовой работы считается достигнутой, разработанное веб-приложение удовлетворяет поставленным перед ним требованиям:

- Пользователь может просматривать информацию о грядущих концертах (название концерта, место проведения концерта, дата и время проведения концерта, статус концерта);
- Пользователь может ознакомиться с исполнителем на основе представленной на сайте информации о нем;
- Пользователь имеет возможность приобретать билеты на выбранные им концерты;
- Пользователь может писать рецензии на артистов и концерты, а также выставять оценки, когда пользователь авторизован;
- Пользователь может получить контактные данные администрации приложения для обратной связи.

Список используемой литературы:

1. Анализ рынка театральных и концертных залов в России в 2016-2020 гг, оценка влияния коронавируса и прогноз на 2021-2025 гг [файл] – URL: https://businessstat.ru/images/demo/concert_and_theatre_russia_demo_businessstat.pdf (дата обращения: 18.03.2021 – 18.06.2021)
2. Moscow Ticketing Forum №2 2019 [файл] – URL: https://ticketingforum.ru/files/digest/MTF_digest_2019.pdf (дата обращения: 18.03.2021 – 18.06.2021)
3. Bootstrap 5.0 Documentation [сайт] – URL: <https://getbootstrap.com/docs/5.0/> (дата обращения: 18.03.2021 – 18.06.2021)
4. Jinja2 Documentation [сайт] – URL: <https://jinja.palletsprojects.com/en/2.11.x/> (дата обращения: 18.03.2021 – 18.06.2021)
5. Flask Documentation [сайт] – URL: <https://flask-doc.readthedocs.io/en/latest/> (дата обращения: 18.03.2021 – 18.06.2021)
6. Spotify API Documentation [сайт] – URL: <https://developer.spotify.com/documentation/> (дата обращения: 18.03.2021 – 18.06.2021)
7. Genius API Documentation [сайт] – URL: <https://docs.genius.com/> (дата обращения: 18.03.2021 – 18.06.2021)
8. Google Calendar for Developers [сайт] – URL: <https://developers.google.com/calendar/api/quickstart/python?hl> (дата обращения: 18.03.2021 – 18.06.2021)
9. Google OAuth Documentation [сайт] – URL: <https://developers.google.com/gdata/docs/auth/oauth> (дата обращения: 18.03.2021 – 18.06.2021)