

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра технологий обработки и защиты информации

Курсовой проект

Система брони площадок и покупки билетов
«ConcertSpace»

09.03.02 Информационные системы и технологии

Обработка изображений и машинное обучение

Обучающийся _____ *Янкевич А. А., 3 курс, д/о*

Обучающийся _____ *Капкин К. А., 3 курс, д/о*

Обучающийся _____ *Коротеева Е. Д., 3 курс, д/о*

Воронеж 2021

Содержание

Введение	3
1 Постановка задачи.....	4
2 Глоссарий	5
3 Анализ предметной области	6
3.1 Анализ существующих решений.....	6
3.1.1 Event Hall	6
3.1.2 Кассир.ру	7
3.1.3 Event Catalog.....	8
4 Анализ задачи	9
4.1 Варианты использования системы.....	9
4.2 Общая структура системы	11
4.3 Конфигурация системы.....	12
4.4 Взаимодействие компонентов системы.....	12
4.5 Взаимодействие в системе	15
4.6 Варианты состояния системы.....	16
4.7 Действия с системой.....	18
4.8 Развертывание системы.....	19
4.9 IDEF0.....	20
5 Анализ средств реализации.....	20

Введение

В настоящее время выступления любимых артистов вживую являются одним из главных желаний фанатов. Но насколько долго может затянуться организация малоизвестных артистов в небольших городах, если вы собираетесь сотрудничать с крупными площадками? А каков шанс, что с вами будут сотрудничать большие, известные площадки, если вы обыкновенный фанат с редкой возможностью, а не заядлый организатор с многолетним опытом за спиной? А еще мы знаем, что многим хочется маленького, душевного концерта в лофт-студии, а не заполненный Crocus City Hall с тысячной толпой. Для этого мы и создали наш ConcertSpace, для всех тех, кому надоело ждать и хочется чего-то «семейного», душевного и уютного.

Сайт должен облегчать процесс бронирования площадок, продажи билетов и покупки билетов. Основная функциональность сайта должна включать в себя:

- бронирование площадки на выбор
- покупка билетов на предоставляемые концерты
- возможность продажи билетов на сайте
- возможность организации концертов для любого пользователя

Данный групповой проект посвящен разработке такой системы: простой для восприятия человеком, но в то же время выполняющей все необходимые функции.

1 Постановка задачи

Данный проект является сервисом, упрощающим бронирование площадок, продажу и покупку билетов на концерты.

Данная система автоматизирует процесс организации и посещения концертов.

Система предназначена для:

- упрощения процесса организации концертов
- продажи и покупки билетов
- получения информации о доступных площадках, включая дату, вместимость, стоимость и доступные параметры для настройки; получения информации о концертах, включая дату, стоимость, описание и возможные виды билетов

Для достижения данной цели были выделены следующие подзадачи:

1. Разработка Front-end части сервиса;
2. Разработка Back-end части сервиса;
3. Создание связи между Front-end и Back-end частями приложения;
4. Разработка базы данных.

2 Глоссарий

ВИ – Вариант Ипользования или Use Case, описание см. https://ru.wikipedia.org/wiki/Сценарий_использования.

Система – Сервис помощи создания команды “ConcertSpace”, требования к которому указаны в данном документе.

Фреймворк – программная платформа, определяющая структуру программной системы.

Валидация – проверка вводимых пользователем данным на корректность.

Модерация – процесс контроля действий пользователей на соответствие принципам ресурса.

Клик – нажатие левой клавишей мыши на объект.

Front-end – клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.

Back-end –программно-аппаратная часть сервиса.

REST API – это стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов.

Github – крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

Веб-сервис, интернет-сервис, система, веб-приложение, проект – идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя.

Контент – наполнение веб-страницы.

Footer – нижняя часть веб-страницы.

3 Анализ предметной области

3.1 Анализ существующих решений

3.1.1 Event Hall

Является самой популярной и большой площадкой проведения концертов в городе Воронеже. На главной странице видна афиша ближайших мероприятий, но нет никакой информации для организаторов. При внимательном изучении страницы, можно найти кнопку «Организаторам», которая переносит на не кликабельный список номеров и почт, что значительно усложняет процесс организации.

Так же стоит заметить, что площадка Event Hall имеет вместимость в семь тысяч человек, что сразу же ставит крест на более-менее малоизвестных в Воронеже исполнителях.

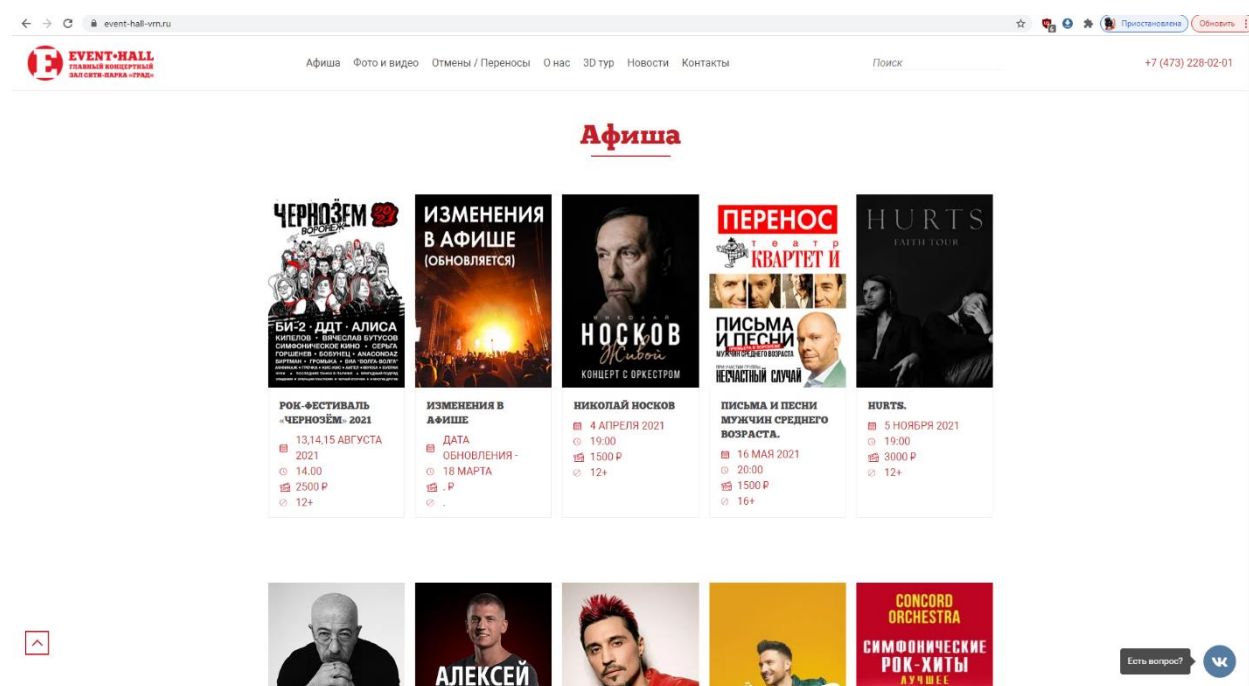


Рисунок 1 – Домашняя страница площадки EventHall

3.1.2 Кассир.ру

Всероссийский сервис по продаже билетов на различные мероприятия. Осуществляет продажу билетов не только на концерты, но и на другие культурные мероприятия во всех крупных городах России. На сайте достаточно трудно найти информация для организаторов, при достаточно внимательном изучении веб-страницы можно найти обозначенную мелким шрифтом кнопку «Организаторам», находящуюся в footer-е домашней страницы. Также стоит отметить, что круг лиц, имеющих шанс сотрудничать с веб-сервисом, ограничивается критериями, прописанными на странице «Организатором», что делает невозможным организацию концертов малоизвестных исполнителей волонтерами «фанатского фронта».

На домашней странице сервиса помещен огромный баннер одного из мероприятий, что отвлекает внимание от других доступных вариантов.

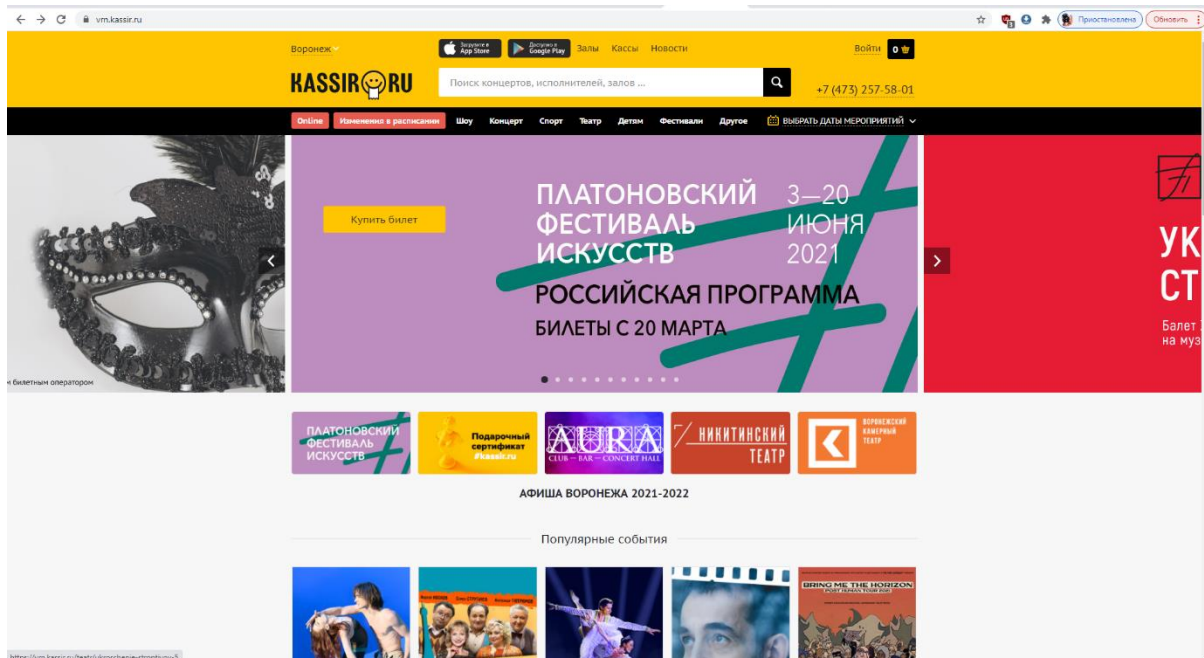


Рисунок 2 – Домашняя страница сервиса продажи билетов Кассир.ру

3.1.3 Event Catalog

Первая ссылка, выскакивающая при Google-запросе «Площадки под концерты» - веб-сервис помощи организации концертов Event Catalog. Хотелось бы отметить, что домашняя страница сайта выглядит сумбурной и совершенно не удобной для пользователя: мелкий шрифт, непонятная навигация, трудно разобраться, как именно осуществить поиск и бронирование. Пролистнуть ниже домашнюю страницу можно увидеть доступные в выбранном городе площадки, сначала можно подумать, что карточки совершенно не кликабельны, ведь переход не осуществляется ни при нажатии на название, ни при попытке развернуть краткое описание площадки. Переход можно осуществить при нажатии на маленькую иконку, расположенную рядом с названием, что совершенно не очевидно для обычного пользователя.

Также стоит отметить, что сервис не предоставляет услуги продажи билетов на мероприятия, а лишь сдает площадки под бронирование.

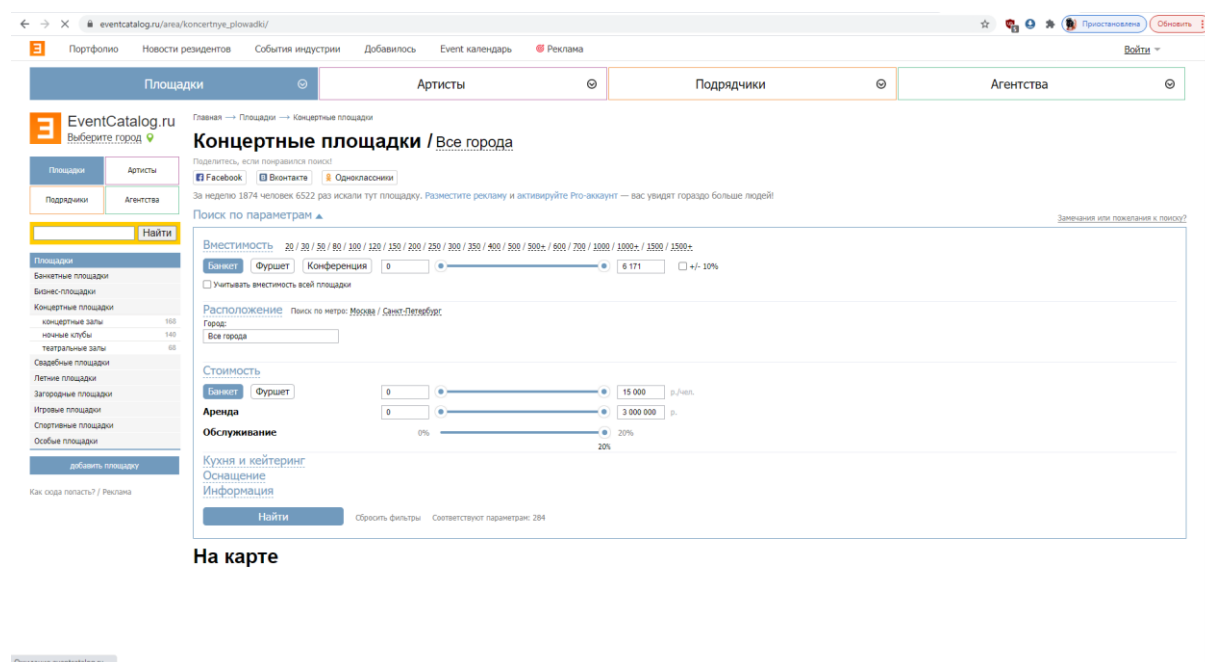


Рисунок 3 – Домашняя страница веб-сервиса Event Catalog

Живой концерт любимого исполнителя – одно из самых важных и долгожданных событий в жизни любого фаната. К сожалению, организация такого мероприятия не всегда легка и доступна, особенно в малых городах или для малоизвестных артистов. Мы создали ConcertSpace для того, чтобы любой, будь то профессиональный организатор или обычный фанат, имели возможность организовывать и продавать не просто живые выступления, а по-настоящему домашние и теплые фанатские встречи.

4 Анализ задачи

4.1 Варианты использования системы

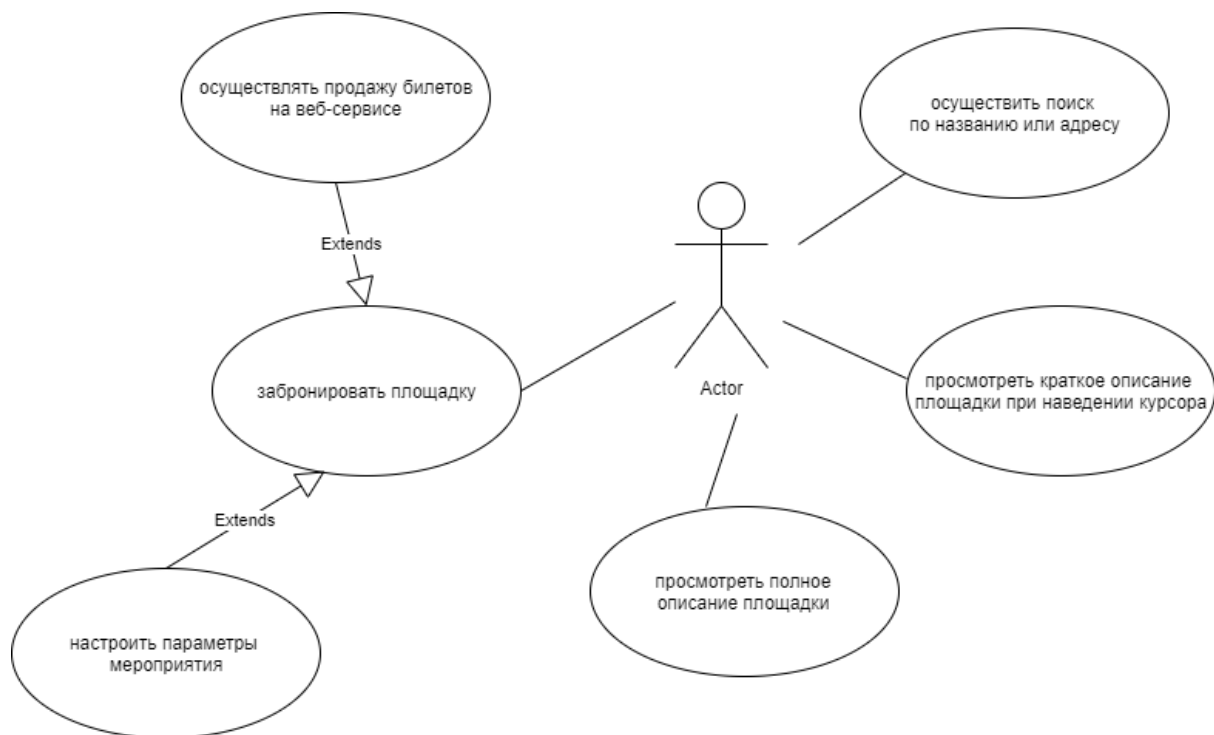


Рисунок 4 – Часть диаграммы прецедентов с актором Организатор.

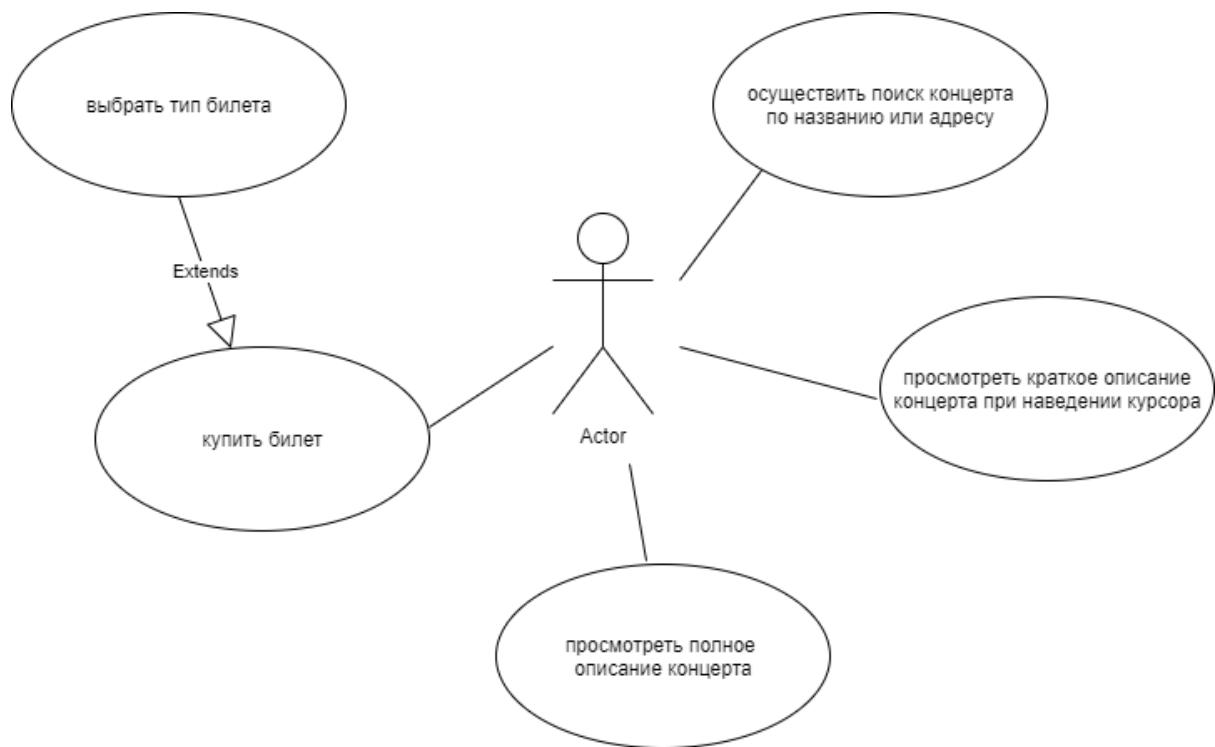


Рисунок 5 – Часть диаграммы прецедентов с актором Покупатель.

При взаимодействии с приложением выделяются 2 сценария: организатор, покупатель. Их роли наглядно изображены на рисунке.

Организатор:

- Просмотреть полное описание площадки
- Посмотреть краткое описание площадки
- Осуществить поиск по названию или адресу
- Забронировать площадку
- Осуществлять продажу билетов на веб-сервисе
- Настроить параметры концерта

Посетитель:

- Купить билет
- Посмотреть полное описание концерта
- Посмотреть краткое описание концерта
- Выбрать тип билета

– Осуществить поиск концерта по названию или адресу

4.2 Общая структура системы

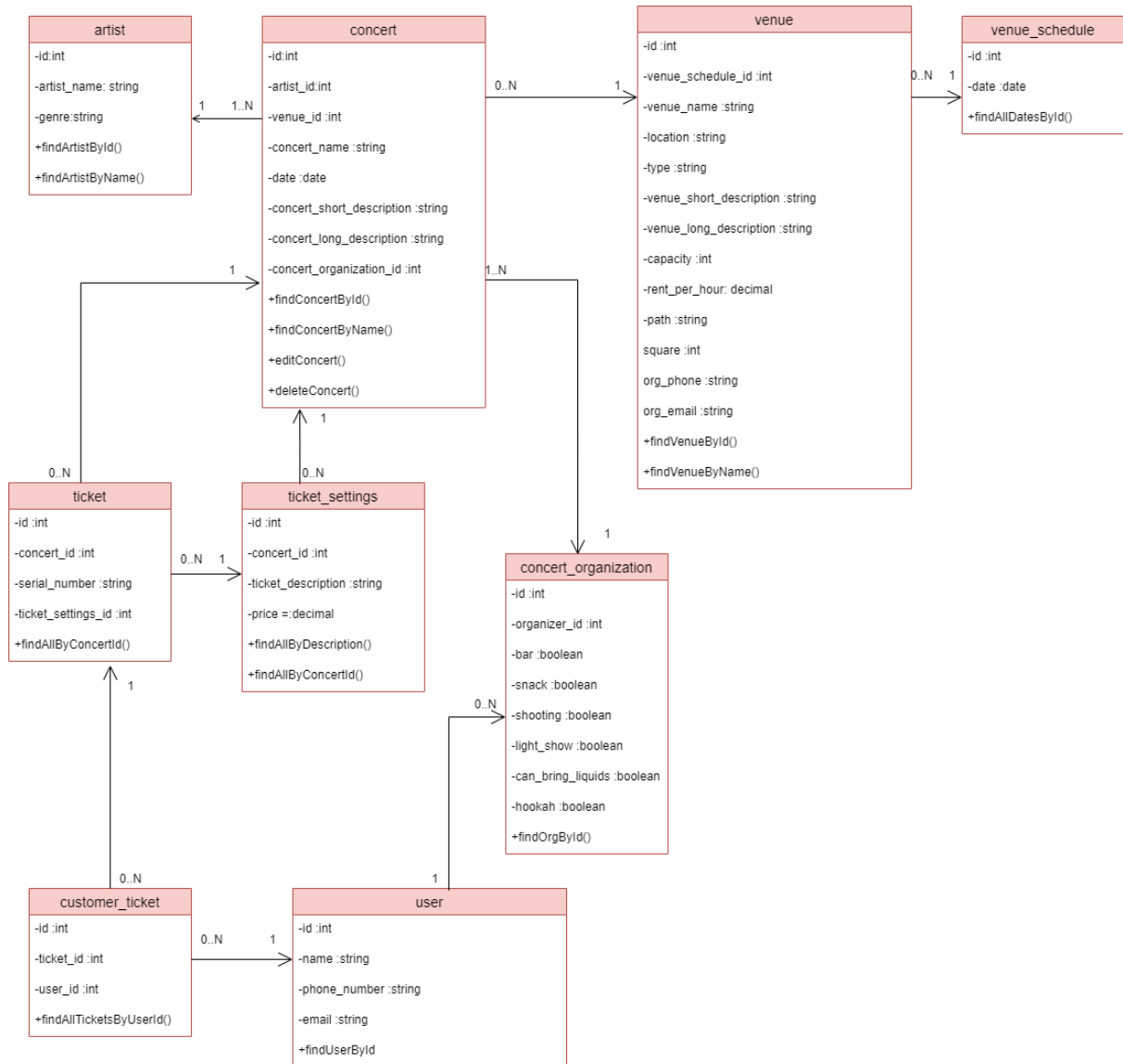


Рисунок 6 – Диаграмма классов.

На рисунке 6 изображена диаграмма классов. Она показывает набор классов и интерфейсов, а также их связи. Между всеми классами установлены взаимосвязи ассоциации.

4.3 Конфигурация системы

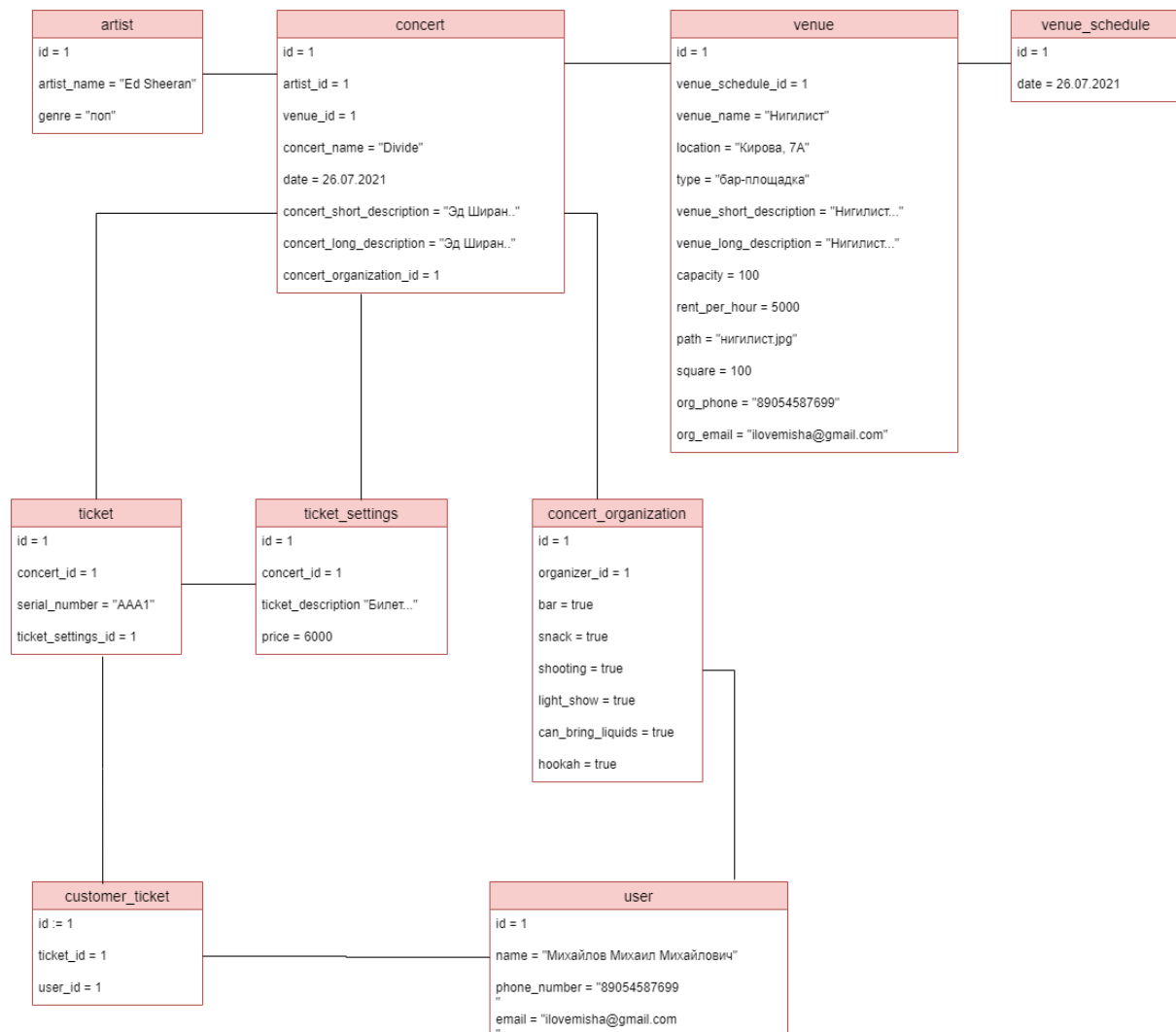


Рисунок 7 – Диаграмма объектов.

4.4 Взаимодействие компонентов системы

На следующих рисунках изображены диаграммы последовательности для сценариев пользователей системы, на которых изображены упорядоченные во времени взаимодействия объектов.

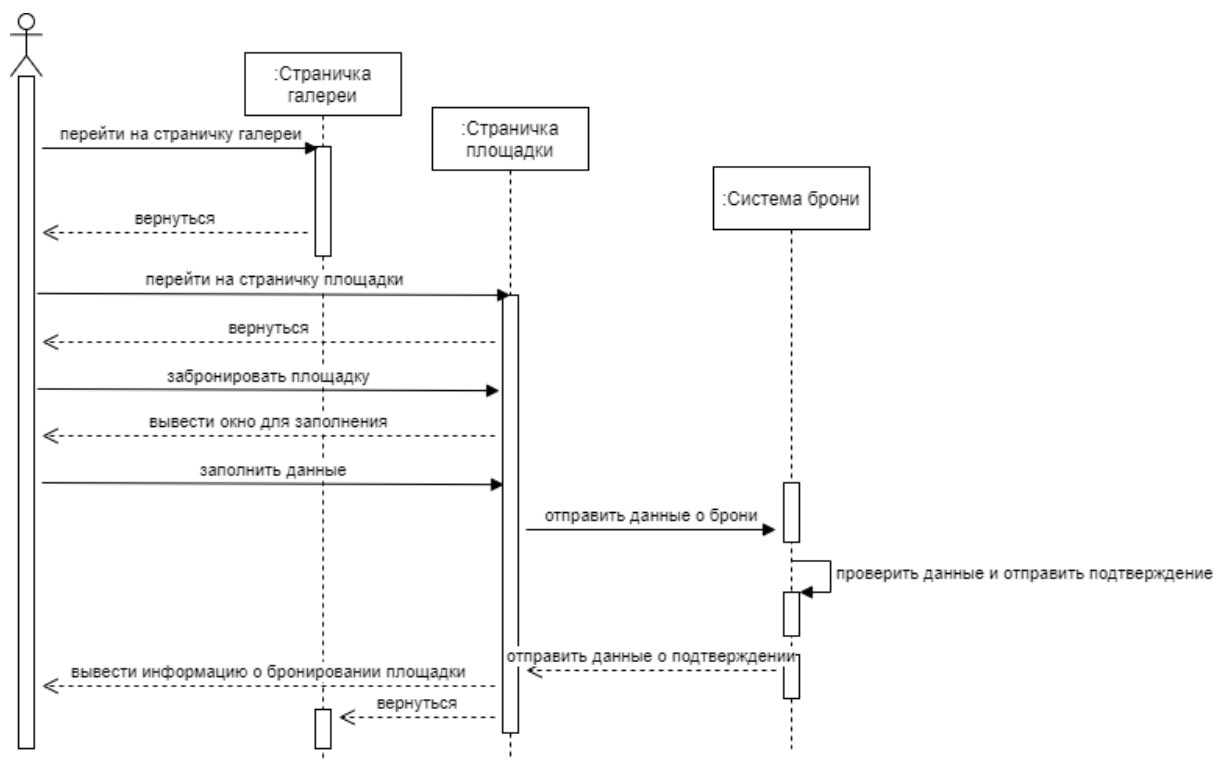


Рисунок 8 – Диаграмма последовательностей организатора.

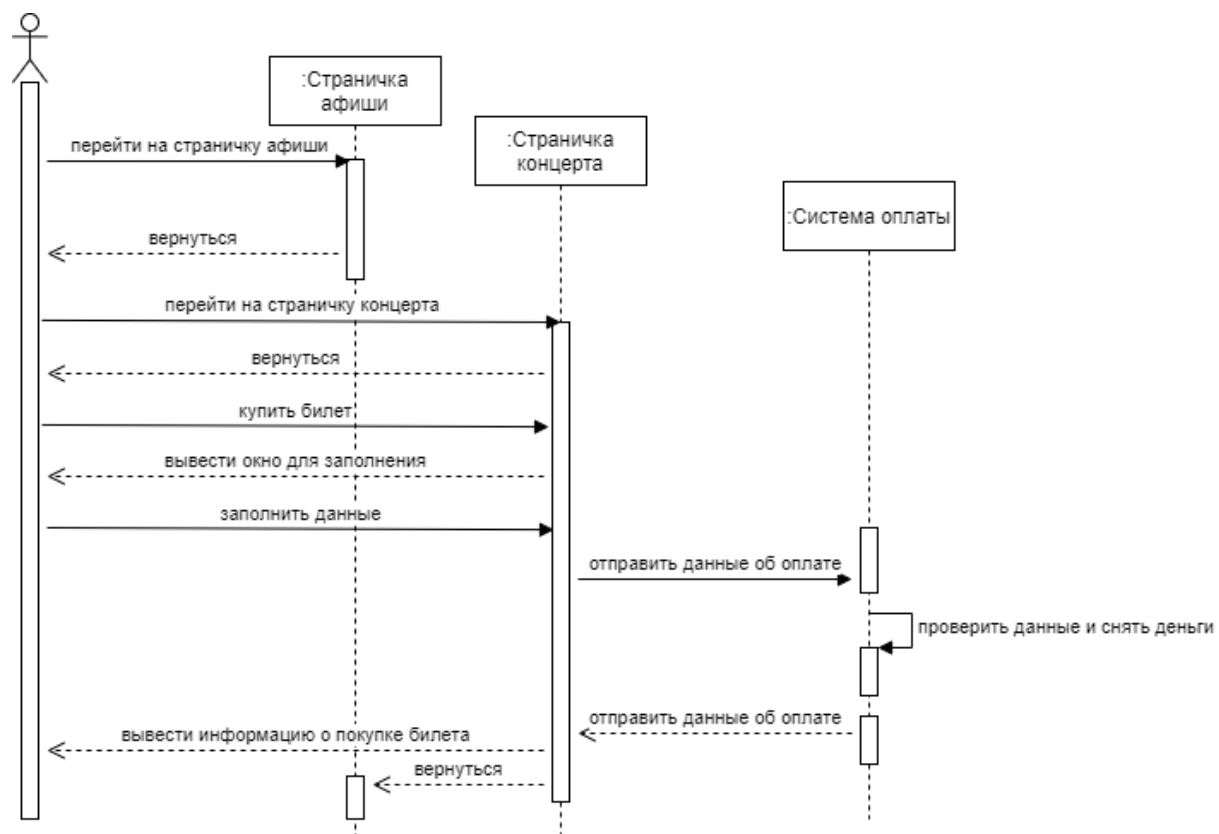


Рисунок 9 – Диаграмма последовательностей покупателя.

4.5 Взаимодействие в системе

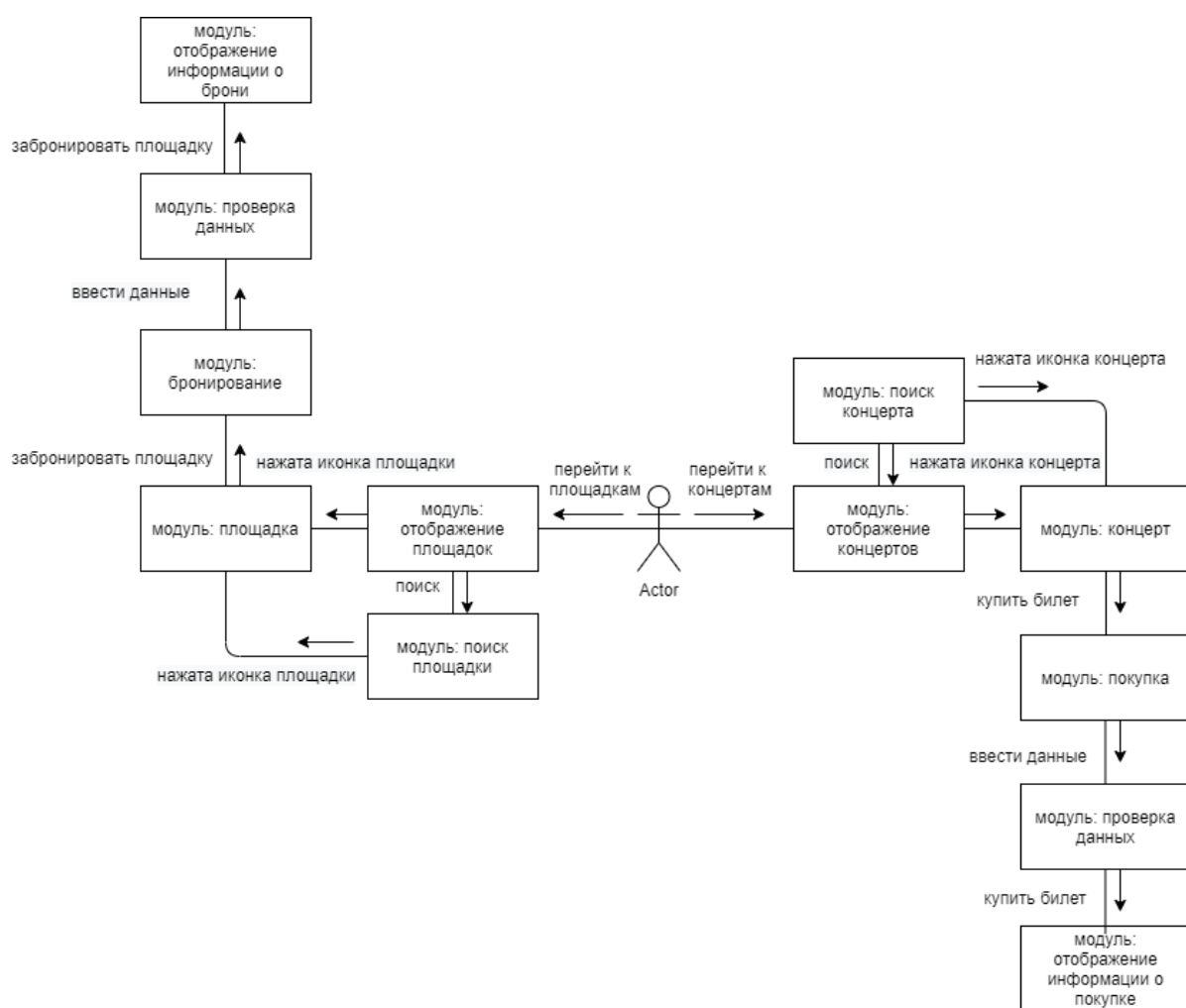


Рисунок 10 – Диаграмма взаимодействия.

На рисунке 10 изображена диаграмма взаимодействия, которая представляет взаимодействие между объектами системы и сообщения, которыми они обмениваются.

4.6 Варианты состояния системы

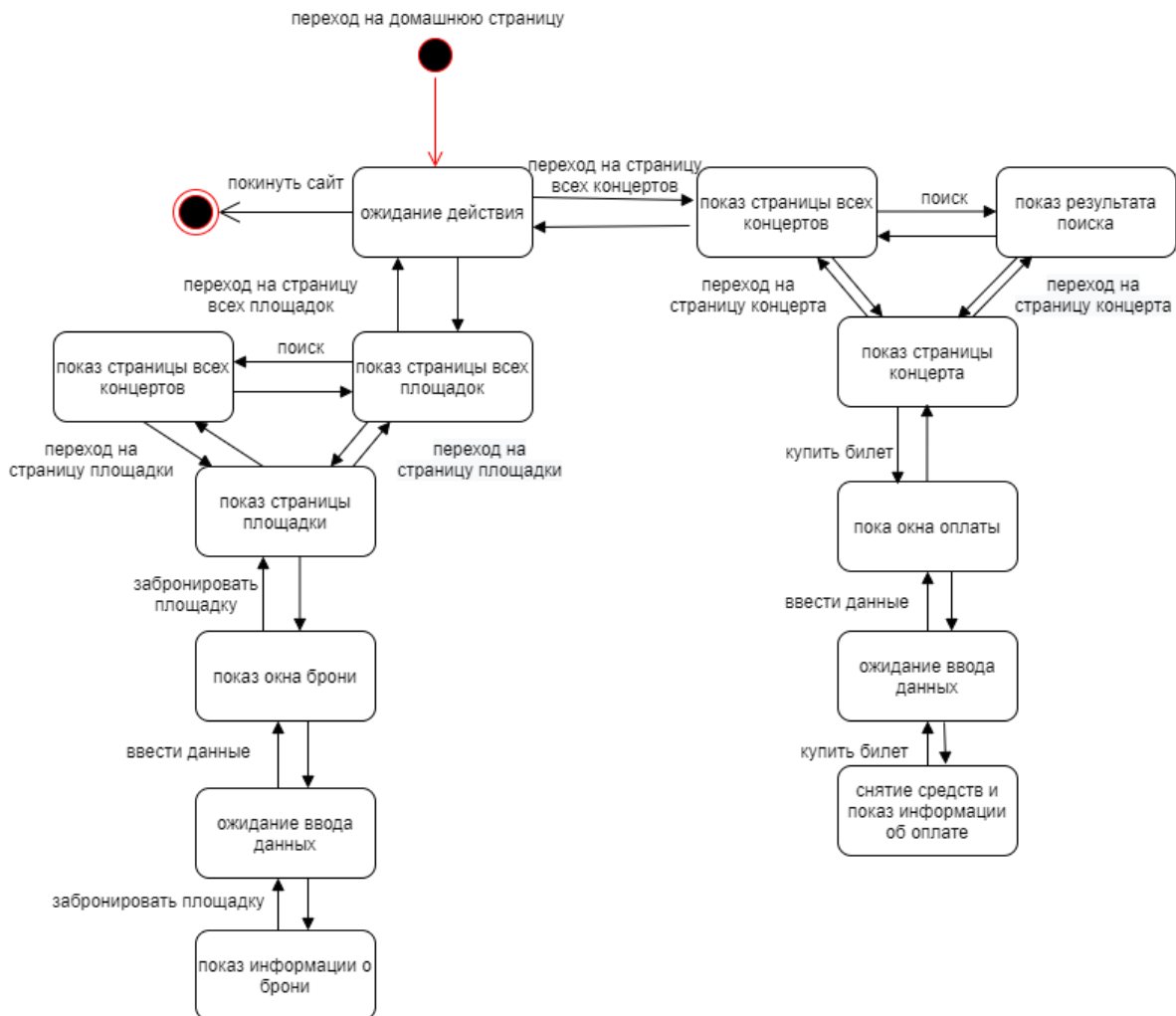


Рисунок 11 – Диаграмма состояний.

Диаграмма состояний, изображенная на рисунке 11, отражает возможные состояния системы. При переходе на главную страницу сайта, система находится в ожидании выбора действия. В зависимости от выбора гостя возможны следующие возможные цепочки событий:

- Переход на страницу всех площадок
- Переход на страницу всех концертов
- Переход на страницу карточки концерта
- Переход на страницу карточки площадки
- Поиск концерта

- Поиск площадки
- Переход на страницу брони
- Переход на страницу покупки билета

Если пользователь системы выбирает переход на страницу всех концертов, система переходит в состояние показа данных всех концертов из базы данных системы. Далее возвращается в состояние ожидания действия.

Если пользователь системы выбирает переход на страницу всех площадок, система переходит в состояние показа данных всех площадок из базы данных системы. Далее возвращается в состояние ожидания действия.

Если пользователь системы выбирает переход на страницу карточки концерта, система переходит в состояние показа данных концерта из базы данных системы. Далее возвращается в состояние ожидания действия.

Если пользователь системы выбирает переход на страницу карточки площадки, система переходит в состояние показа данных площадки из базы данных системы. Далее возвращается в состояние ожидания действия.

Если пользователь системы выбирает поиск концерта, система переходит в состояние показа результата поиска концерта в базе данных. Далее возвращается в состояние ожидания действия.

Если пользователь системы выбирает поиск площадки, система переходит в состояние показа результата поиска площадки в базе данных. Далее возвращается в состояние ожидания действия.

Если пользователь системы выбирает переход на страницу брони, система переходит в состояние показа полей для заполнения данных, как только данные заполнены и проверены, а бронь подтверждена, система переходит в состояние ожидания действия.

Если пользователь системы выбирает переход на страницу покупки, система переходит в состояние показа полей для заполнения данных, как

только данные заполнены и проверены, а покупка подтверждена, система переходит в состояние ожидания действия.

4.7 Действия с системой

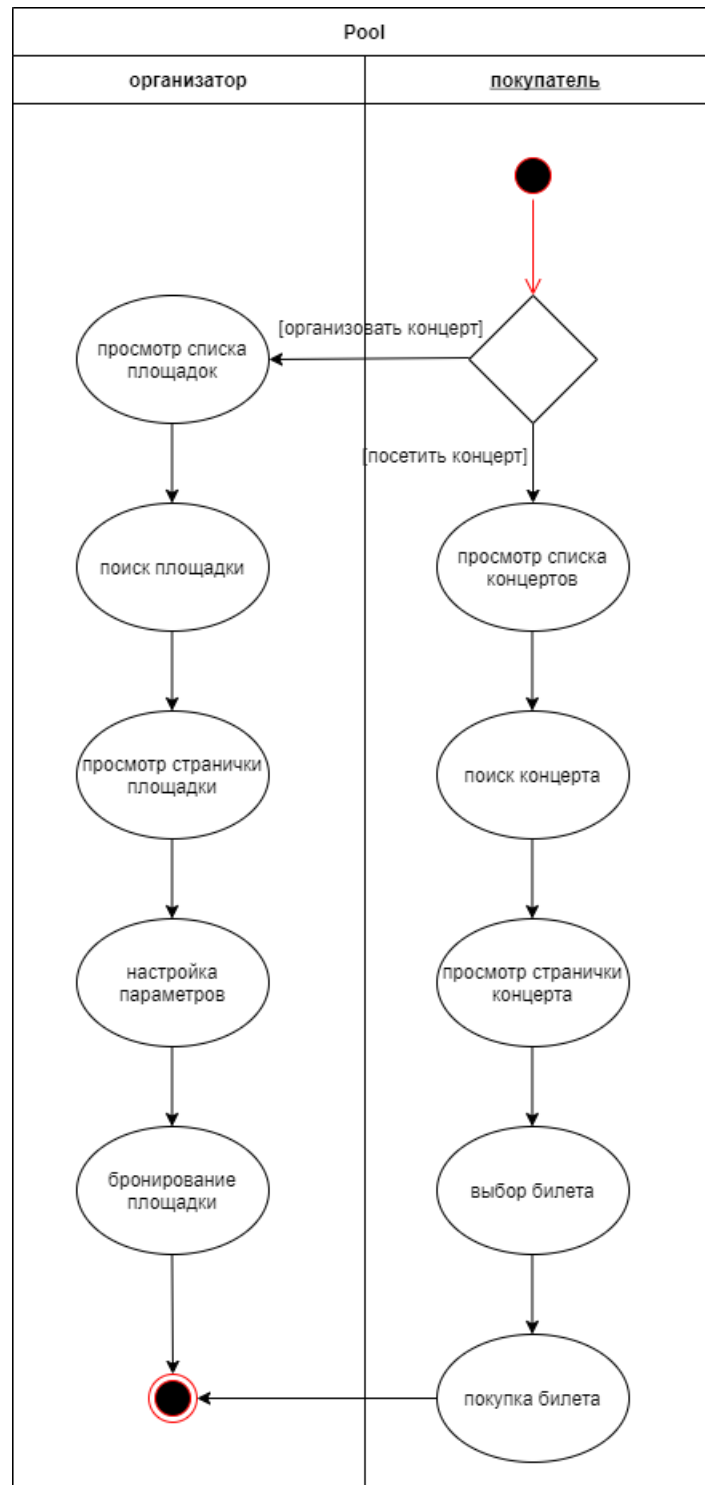


Рисунок 12 – Диаграмма активностей.

На рисунке 12 изображена диаграмма активностей, которая показывает поток последовательности действий для перехода от одной деятельности к другой.

4.8 Развертывание системы

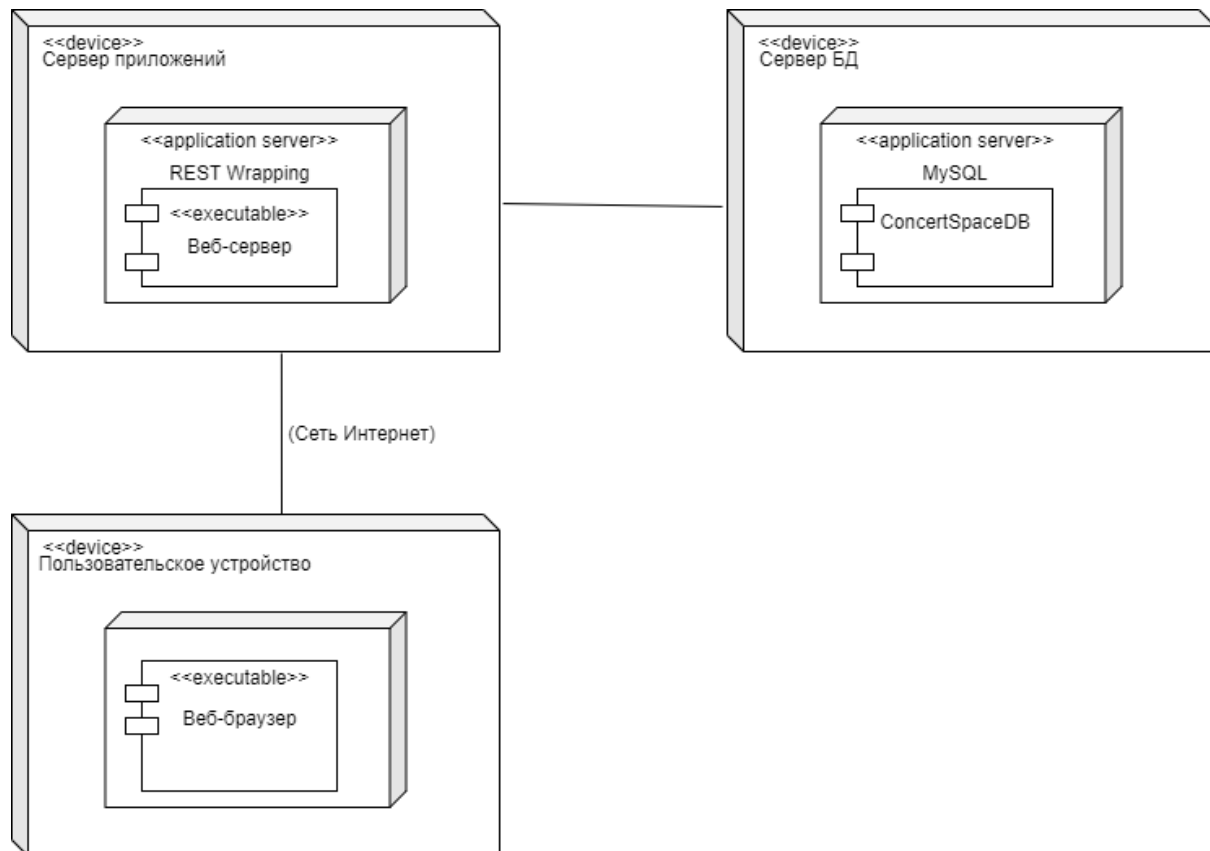


Рисунок 13 – Диаграмма развертывания.

На рисунке 13 изображена диаграмма развертывания, показывающая топологию системы и распределение компонентов по ее узлам, а также соединения – маршруты передачи информации.

4.9 IDEF0

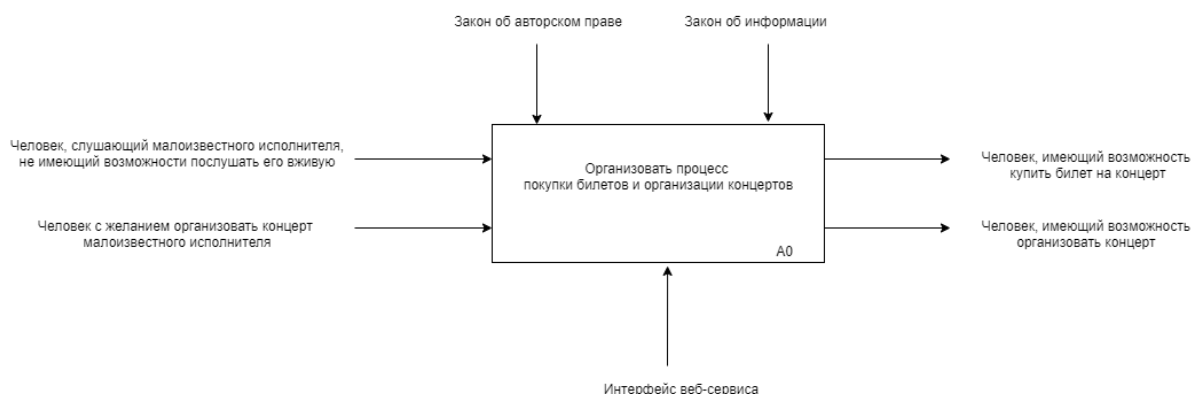


Рисунок 14 – IDEF0

На рисунке 14 изображена IDEF0 диаграмма.

5 Анализ средств реализации

В качестве средств реализации приложения были выбраны следующие технологии:

- ReactJS – JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. Его цель — предоставить высокую скорость, простоту и масштабируемость.

- Material UI – стиль графического дизайна интерфейсов программного обеспечения и приложений. Стиль расширяет идею «карточек», более широким применением строгих макетов, анимаций и переходов, отступов и эффектов глубины (света и тени).

- MqSQL – свободная реляционная система управления базами данных. Поддерживает архитектуру клиент-сервер. Преимуществами являются быстрота, надежность и легкость использования.

- В качестве языка разработки использовался язык Java.

- Шаблон проектирования MVC выбран по причине разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компоненты: модель, представление и контроллер.