

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

Факультет компьютерных наук

Кафедра программирования и информационных систем

Продажа электронных билетов на концерты на различных площадках.

Курсовая работа

09.03.02 Информационные системы и технологии

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

Руководитель _____ *В.С.Тарасов,*
преподаватель

Воронеж 2021

Содержание

Содержание	2
Введение	3
1. Постановка задачи	3
2. Анализ предметной области	4
2.1. Анализ аналогов	4
2.2. Анализ потребности	6
3. Продуктовые воронки	6
4. Диаграммы	7
4.1. Диаграмма прецедентов	7
4.2. Диаграмма классов	8
4.3. Диаграмма объектов	9
4.4. Диаграмма последовательностей	9
4.5. Диаграмма взаимодействий	14
4.6. Диаграмма состояний	15
4.7. Диаграмма активности	17
4.8. Диаграмма развертывания	19
4.9. Модель IDEF0	19
4.10. ER-диаграмма	21

Введение

С ростом прогресса в области информационных систем, у человека появилась возможность автоматизировать некоторые элементы обыденной жизни. С каждым днем возможность автоматизации обретает всё больше процессов.

Если говорить точнее, то иметь возможность продать что-то, не задействовав человека, а с помощью автоматизации более выгодно. Особенно, если это не какое-то глобальное производство, а веб-приложение.

Данная курсовая работа обзрывает процесс создания веб-приложения, специализацией которого является деятельность в сфере продажи билетов на концерты. Здесь и далее будет рассмотрены цели, задачи, специфика и результаты работы по созданию приложения.

1. Постановка задачи

Процесс создания каждого приложения несет в себе конкретную цель, не стало исключением и наше приложение. Его цель - выступить посредником между лицами, заинтересованными в продаже и покупке билетов, а также теми, кто хочет выступать, и теми, кто дает эту возможность. Если говорить подробнее, то это в первую очередь организация процесса покупки, то есть предоставление возможности продавцу продать билет, а также интерфейс для того, чтобы покупатель мог выбрать интересующее мероприятие. Не менее важным будет также позволить выступающему, непосредственно принадлежащему к выступлениям, участвовать в рассматриваемой системе.

Одной цели для приложения недостаточно, необходимы задачи, определение которых поможет определить путь развития приложения.

Основная задача данного проекта - создать приложение для продажи билетов, полностью соответствующее подготовленному техническому заданию.

Задачи, которые необходимо решить в ходе развития проекта:

- Собрать необходимую информацию об исследуемой предметной области;
- Провести анализ полученных данных;
- Разработать концепции работы приложения;
- Выбрать наилучший вариант и создать модель программы;
- Написать код приложения, отладить его, откорректировать до стабильной работоспособности;
- Тестирование и доработка приложения.

2. Анализ предметной области

2.1. Анализ аналогов

Разрабатываемое приложение не является единственным в своем роде, и у него есть аналоги, например Яндекс.Афиша, Afisha.ru, Kassir.ru, Сайт концертного зала «Event-Hall».

Яндекс.Афиша - это билетный сервис, работающий с 2005 года, от уже известной компании Яндекс.

Сильные стороны сервиса:

- Выбор места;
- Наличие рекомендательной системы;
- Есть особые разделы, например “Квесты”;
- Система скидок.

Слабые:

- Есть реклама, из-за чего просмотр содержимого сайта замедляется.

Как говорят авторы следующего сервиса, Afisha.ru - медийно-сервисная платформа о развлечениях и культурном досуге, которая была основана в апреле 1999 года.

Сильные стороны сервиса:

- Возможность оставлять отзывы;
- Работа с соц. сетями, на сайте есть ссылки на них;
- Возможность создать собственное мероприятие прямо на сайте.

Слабые:

- Интерфейс покупки билета для разных мероприятий отличается;

Kassir.ru является официальным билетным оператором событий федерального масштаба. Компания сотрудничает с многими промоутерскими компаниями России и мира.

Сильные стороны сервиса:

- Можно взять билет на онлайн-трансляцию;
- Наличие истории просмотров.

Слабые:

- Отсутствует возможность выбора фильма.

Сайт концертного зала «Event-Hall» является неплохим примером сервиса для продажи билетов в небольшом масштабе.

Сильные стороны сервиса:

- Сайт специализирован для своего концертного зала.

Слабые:

- Сайт предназначен только для одного концертного зала;
- Вход в личный кабинет недоступен на главном экране.

2.2. 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

Рассмотрим, какие шаги необходимо сделать, чтобы приобрести билет на концерт, т.е. чтобы дойти до основной функции приложения. Пользователь сначала должен выбрать требуемый концерт (т.е. либо выбрать его на главной странице, либо выбрать через поиск) (шаг 1), затем нажать на кнопку покупки билетов (шаг 2), выбрать места (шаг 3) и оплатить их (шаг 4). Таким образом всего за 4 шага пользователь может купить билет на желаемый концерт.

4. Диаграммы

4.1. Диаграмма прецедентов

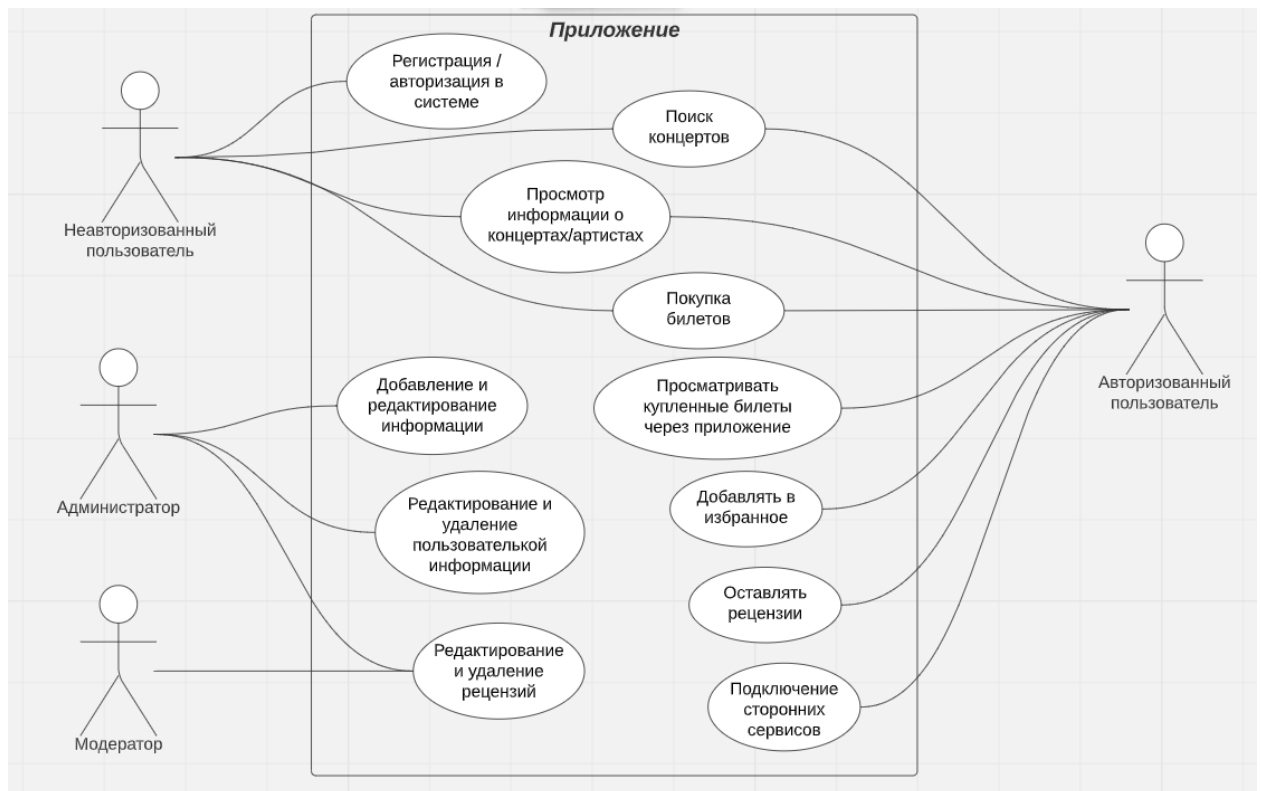


Рисунок 1 - Диаграмма прецедентов (use-case)

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

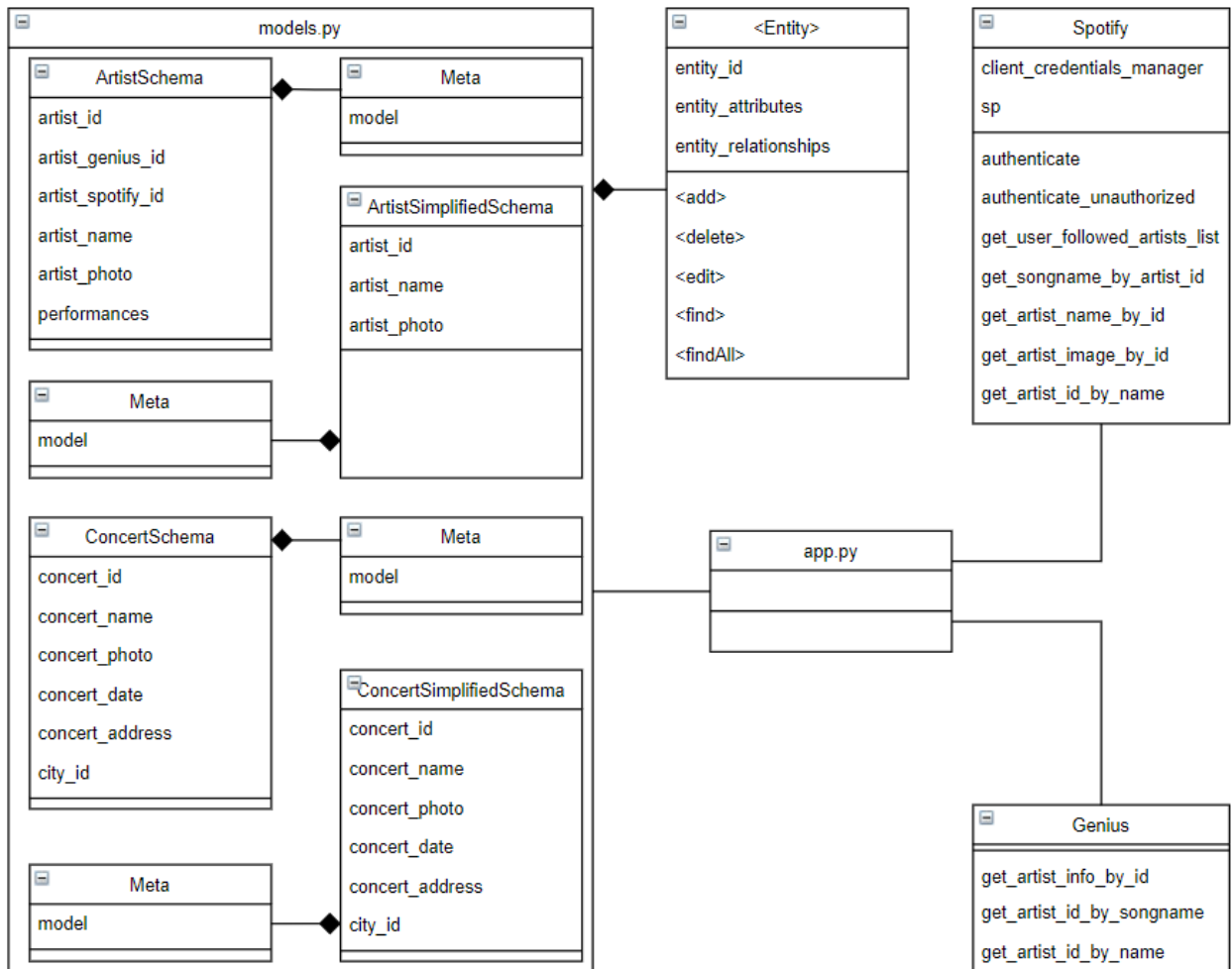


Рисунок 2 - Диаграмма классов приложения

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

4.3. Диаграмма объектов

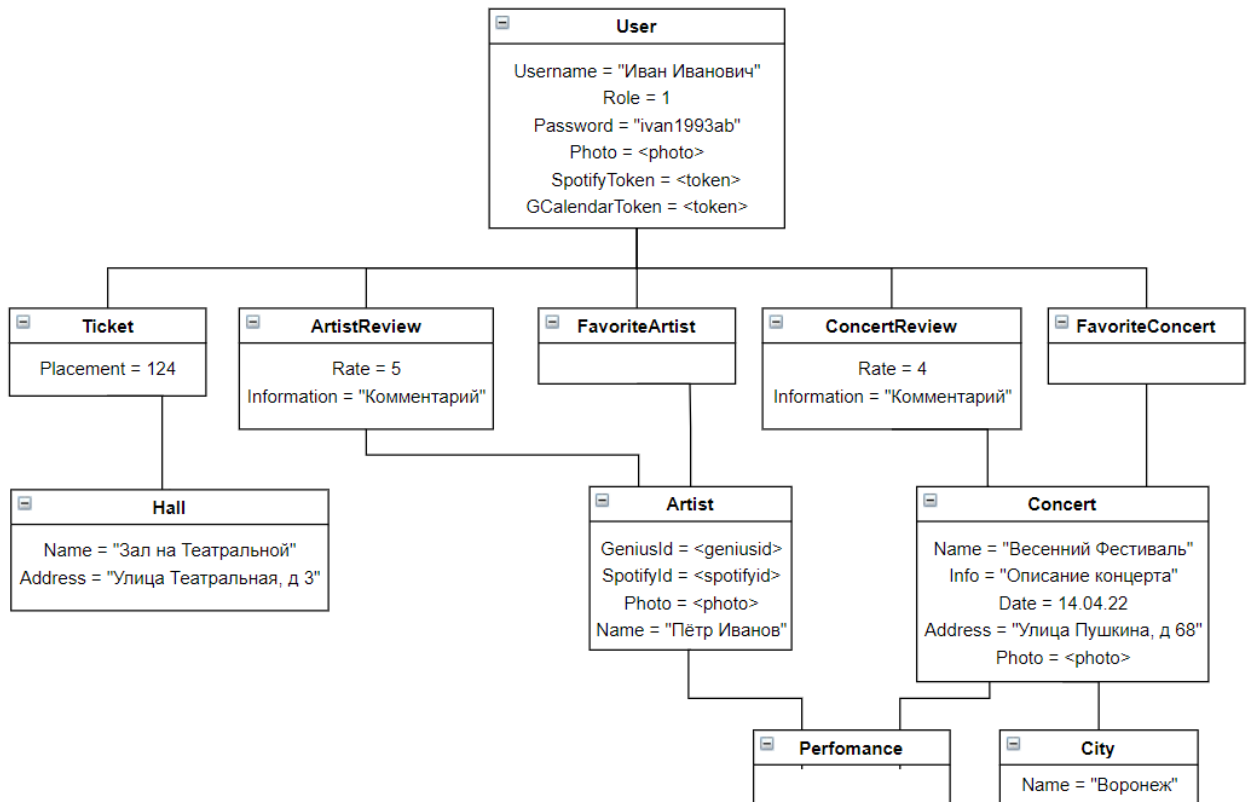


Рисунок 3 - Диаграмма объектов проекта

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 сценариев.

Регистрация аккаунта

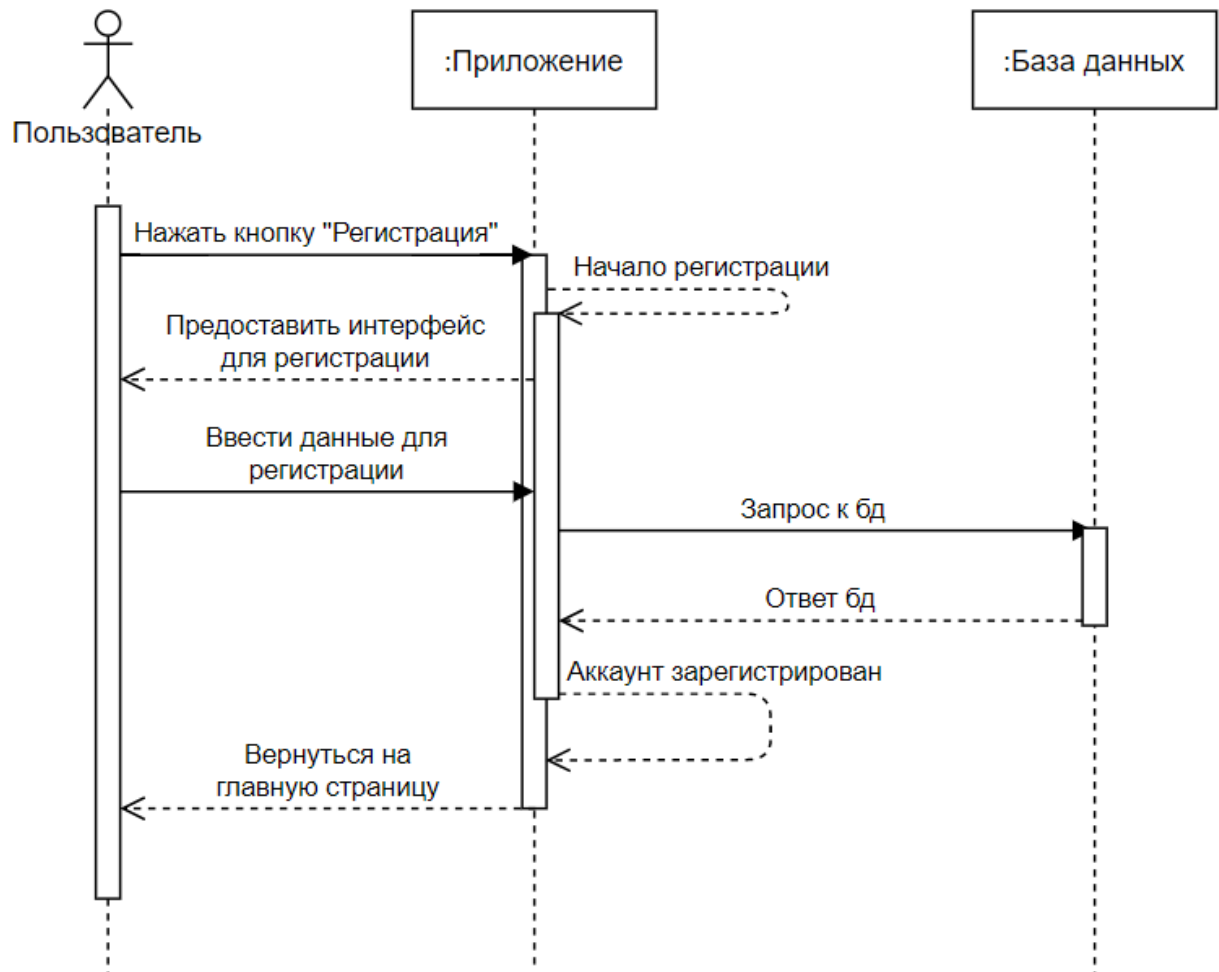


Рисунок 4 - Диаграмма последовательности для регистрации

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.

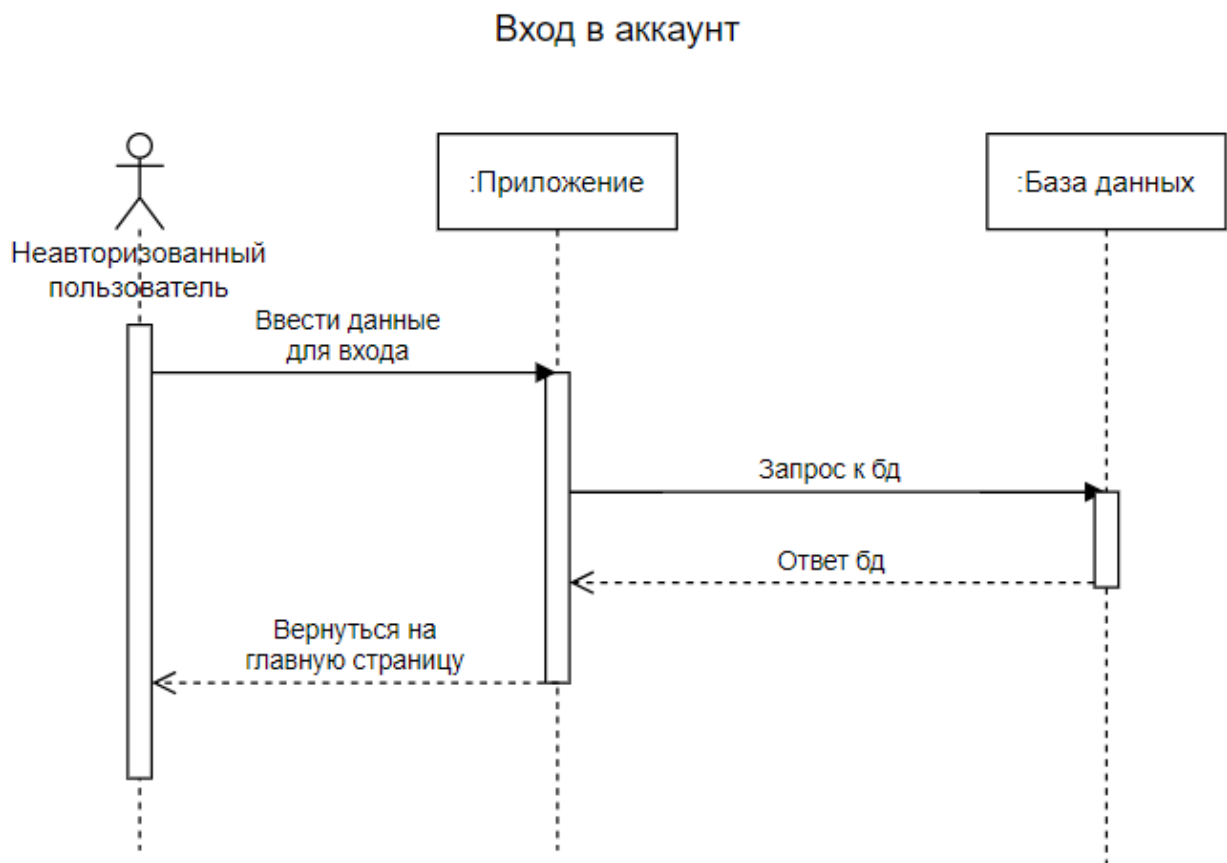


Рисунок 5 - Диаграмма последовательностей для авторизации

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.

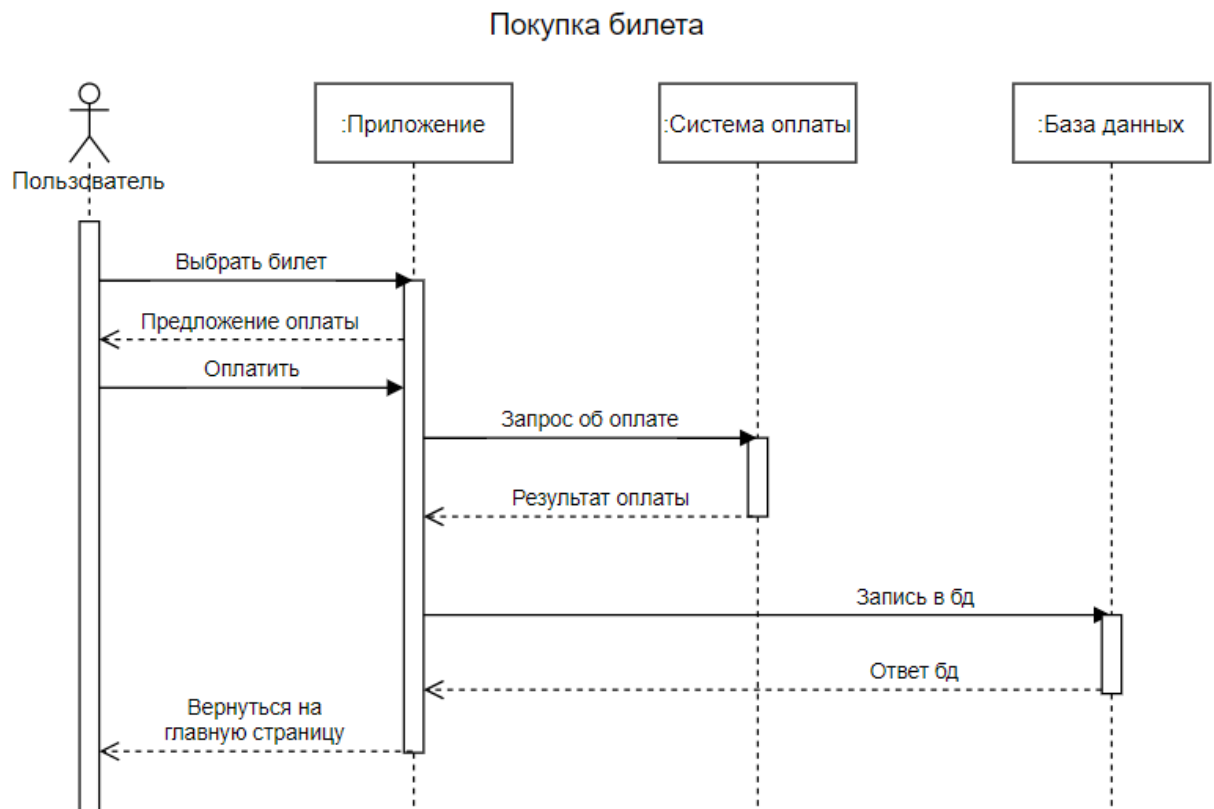


Рисунок 6 - Диаграмма последовательностей для покупки билета

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.

Добавить в избранное

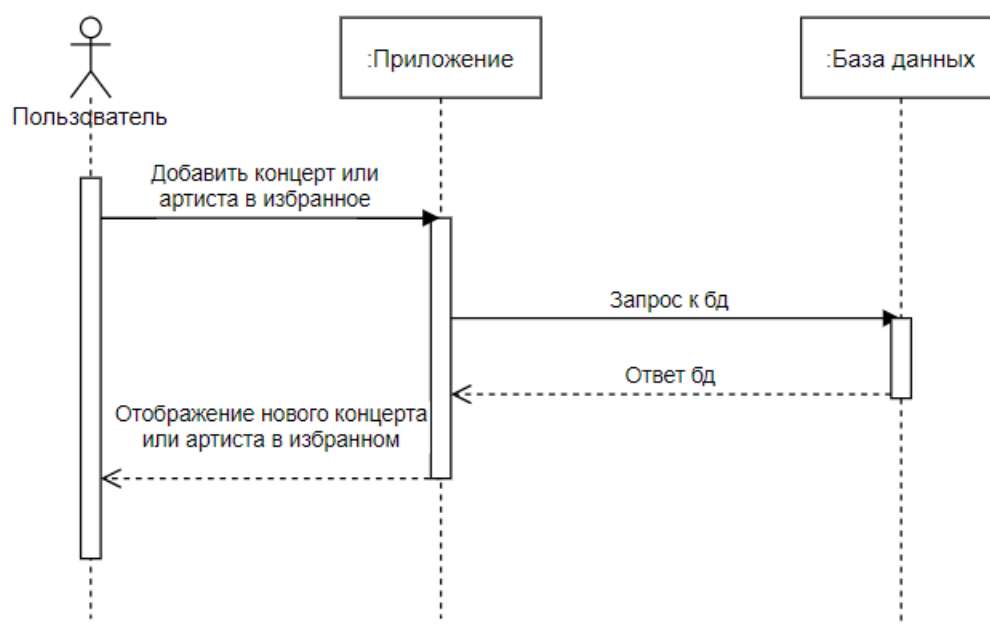


Рисунок 7 - Диаграмма последовательностей для добавления элемента в избранное

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.

Комментарии пользователей

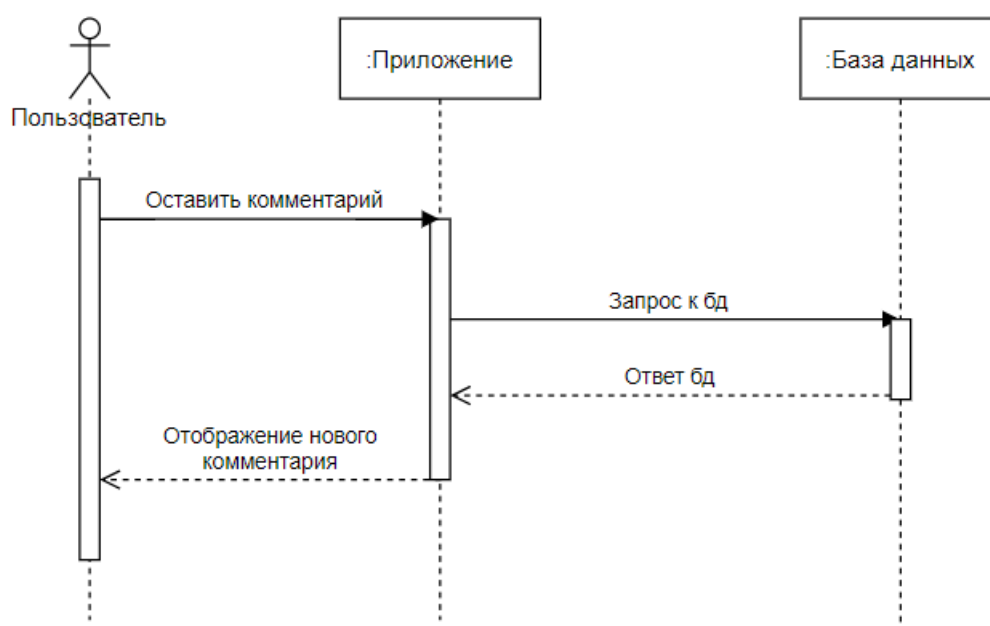


Рисунок 8 - Диаграмма последовательностей для добавления комментариев

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

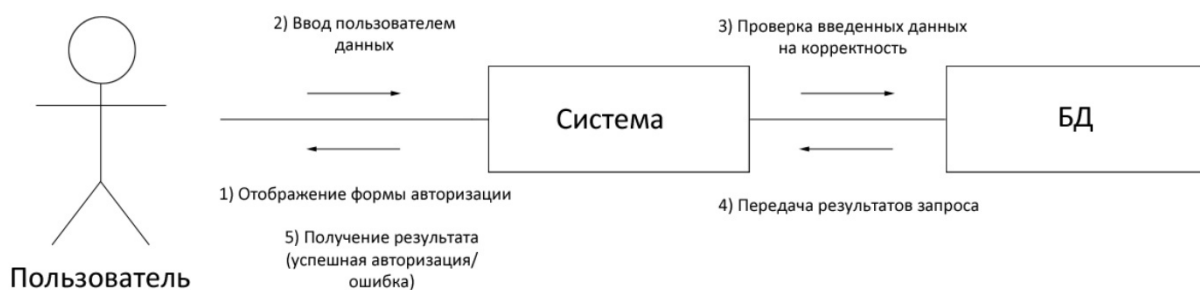


Рисунок 9 - Диаграмма взаимодействия для авторизации

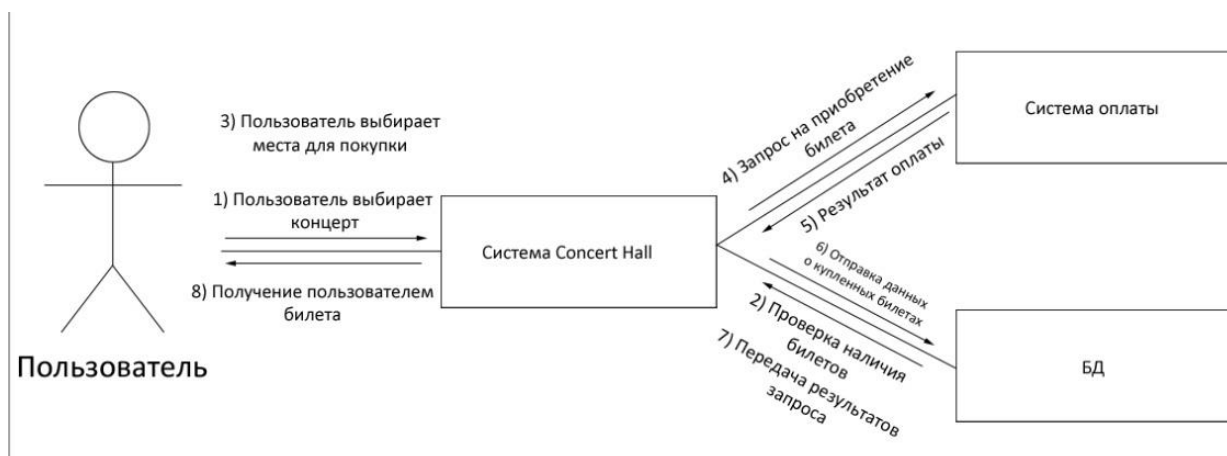


Рисунок 10 - Диаграмма взаимодействия для покупки билета

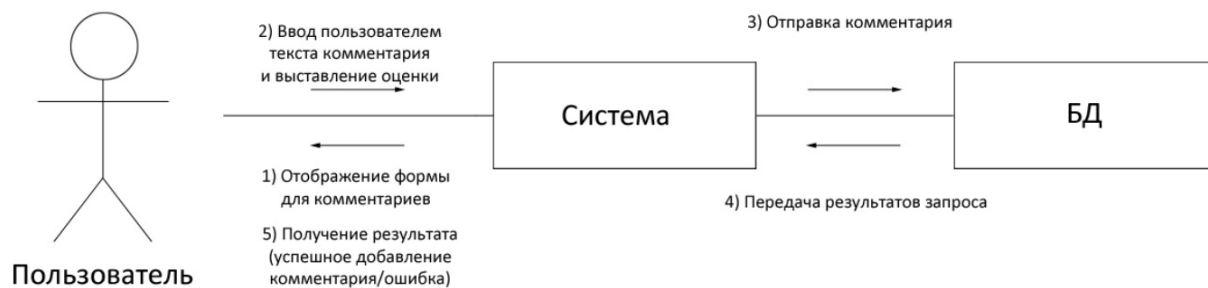


Рисунок 11 - Диаграмма взаимодействия для добавления комментария

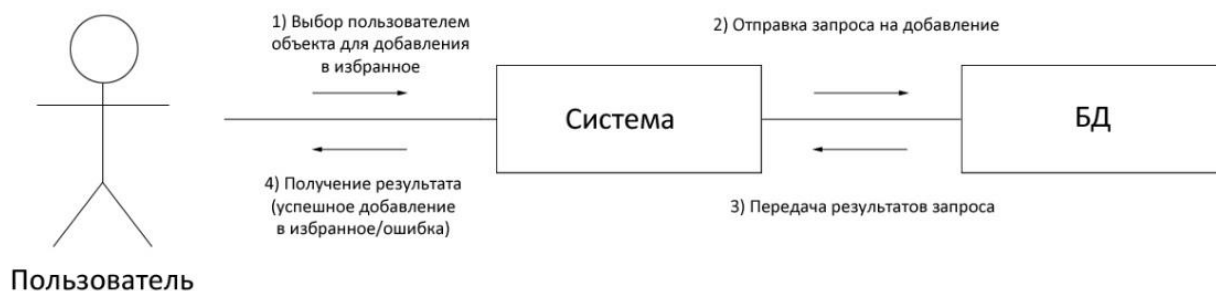


Рисунок 12 - Диаграмма взаимодействия для добавления элемента в избранное

4.6. Диаграмма состояний

Данные диаграммы описывают переход объектов из одних состояний в другие.

Диаграмма состояния для пользователя

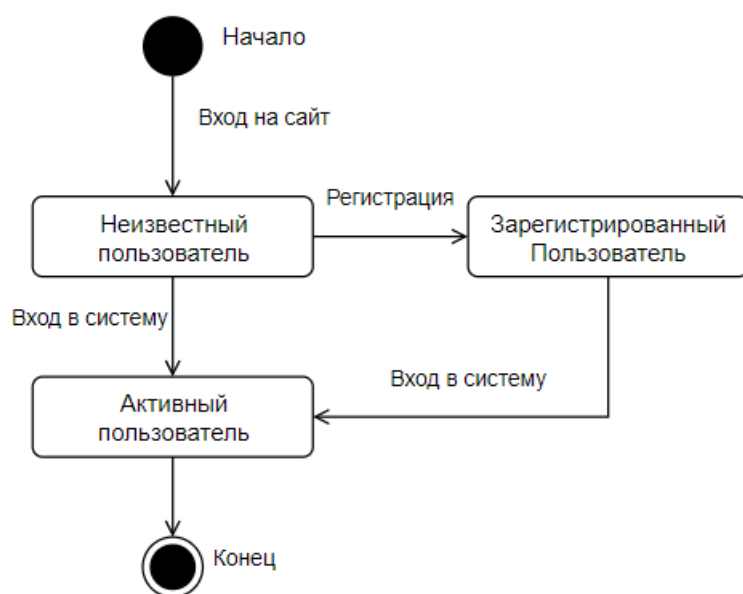


Рисунок 13 - Диаграмма состояний для авторизации

Диаграмма состояния для концерта

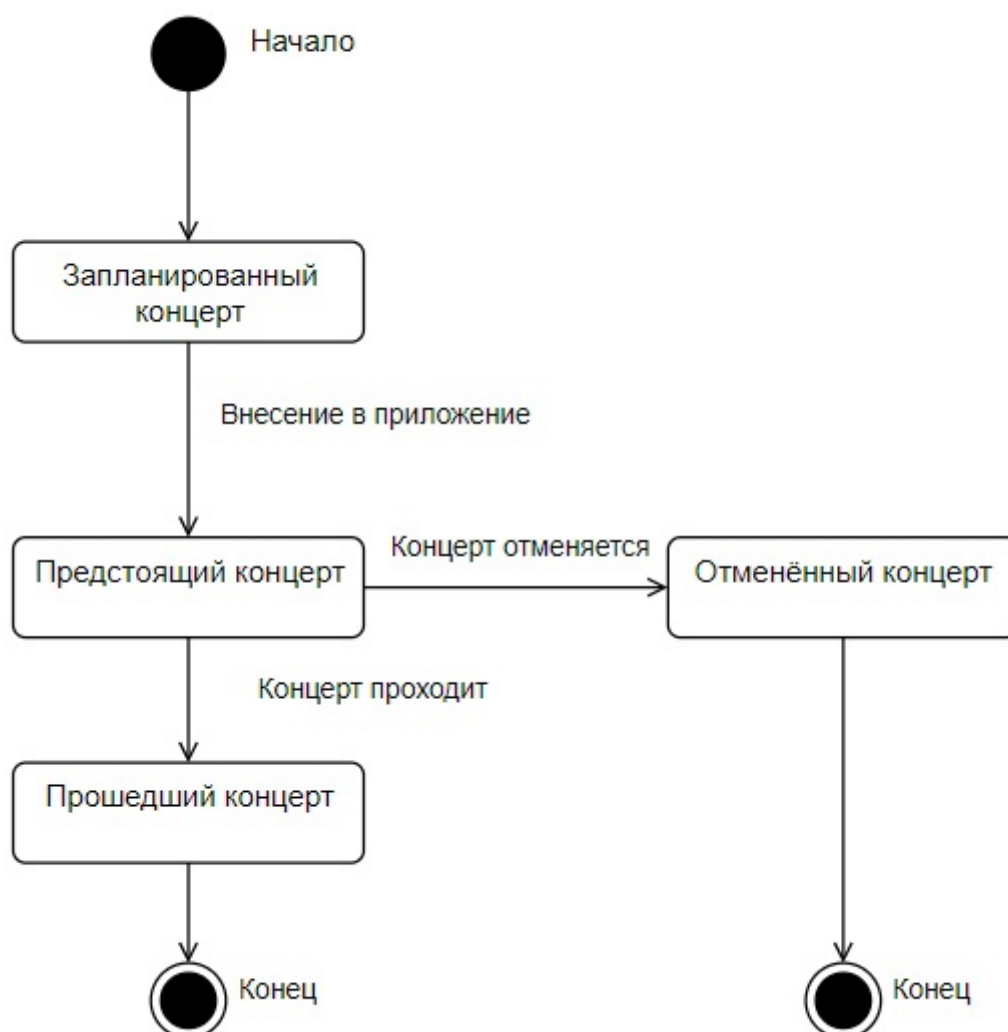


Рисунок 14 - Диаграмма состояний для концерта

Диаграмма состояния для билета



Рисунок 15 - Диаграмма состояний для билета

4.7. Диаграмма активности

Диаграмма активности описывает возможную последовательность объектов системы. Например, в данной диаграмме мы видим, 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.

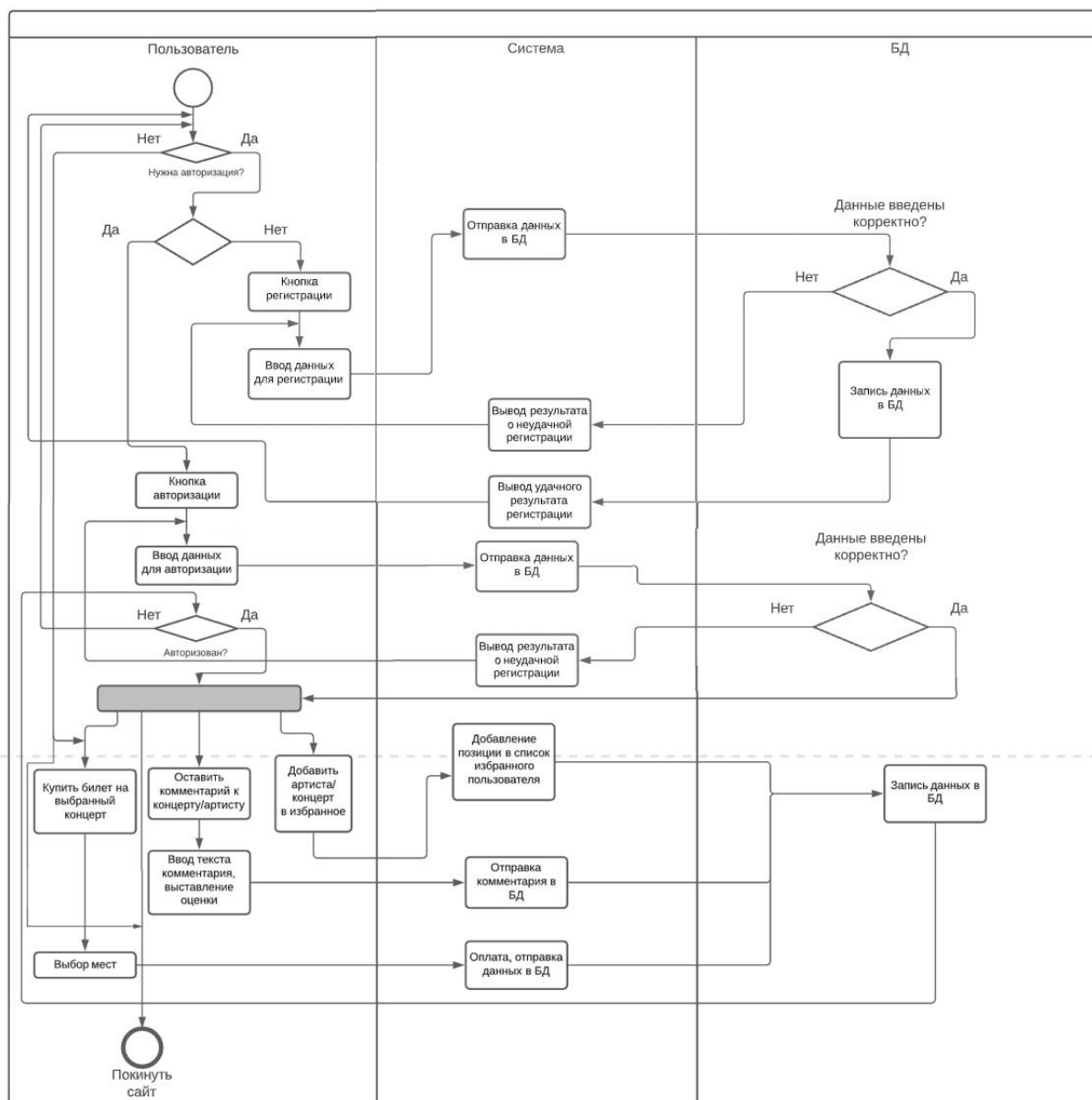


Рисунок 16 - Диаграмма активности для приложения

4.8. Диаграмма развертывания

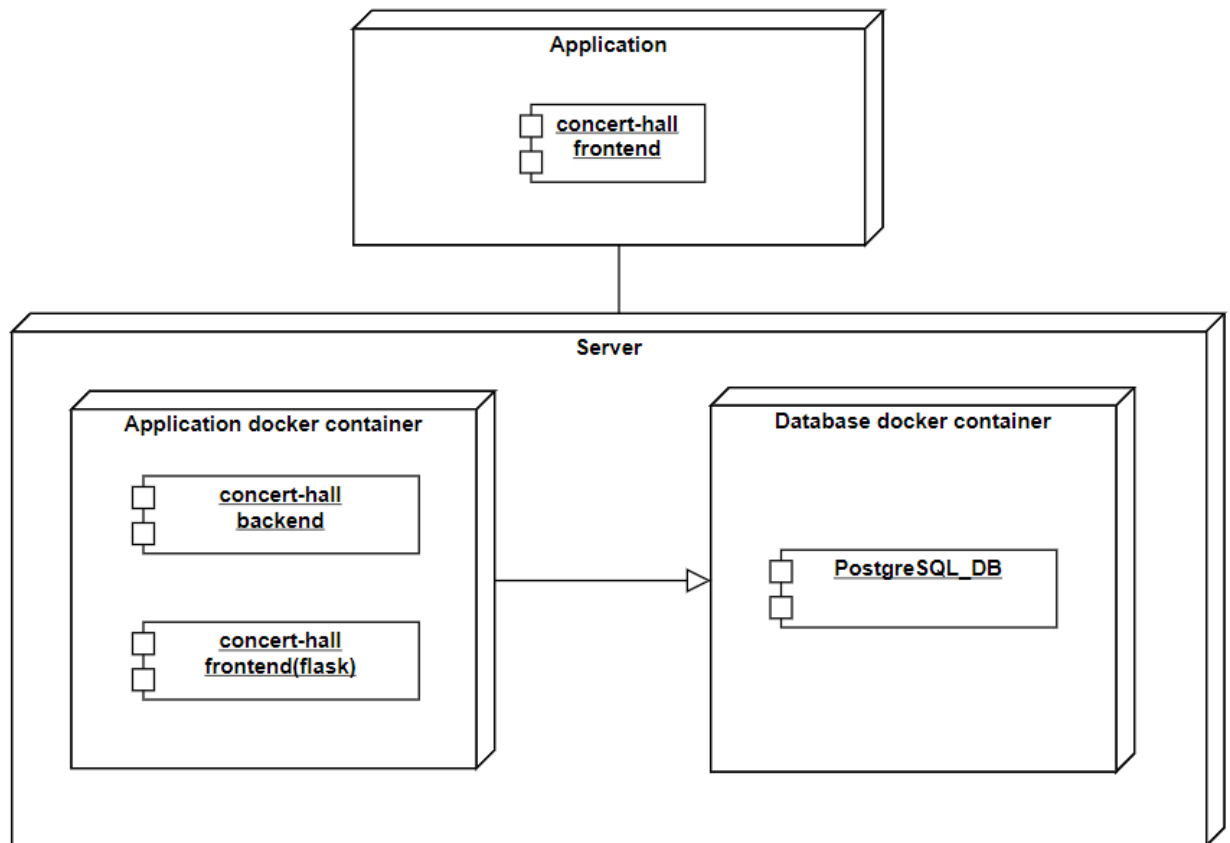


Рисунок 17 - Диаграмма развертывания для приложения

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.

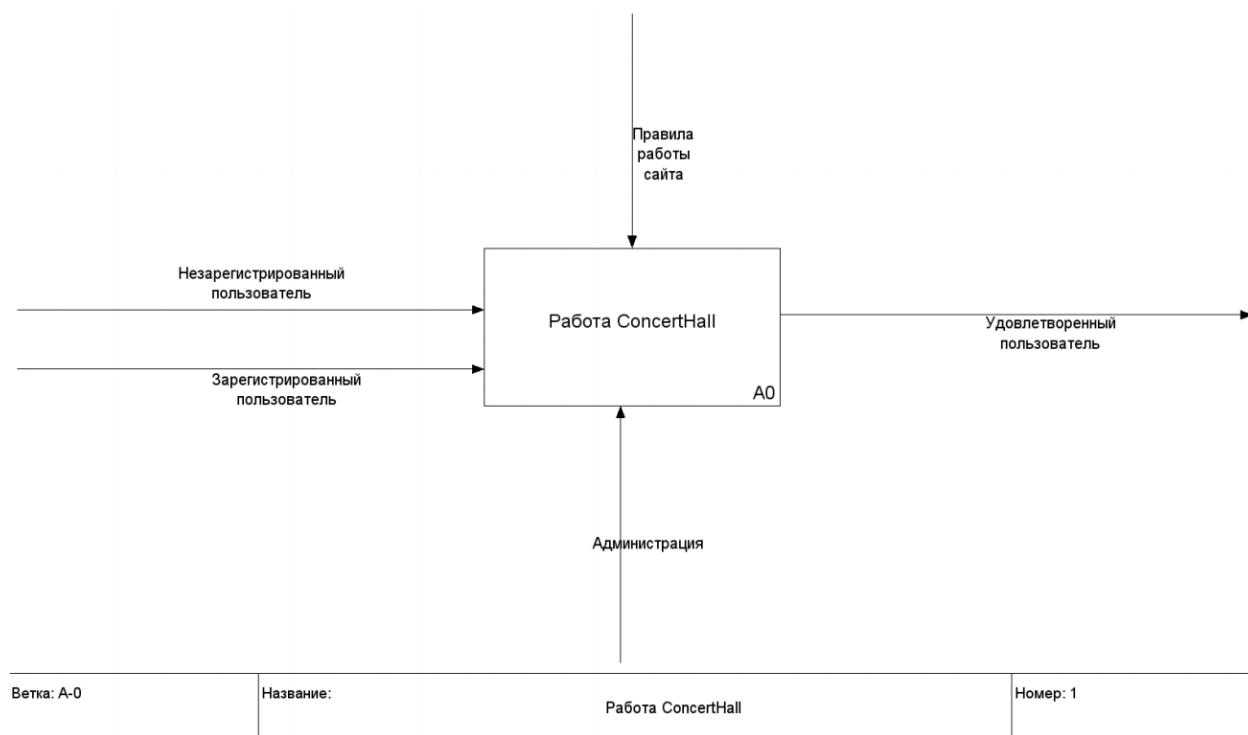


Рисунок 18 - Модель IDEF0 – уровень A-0

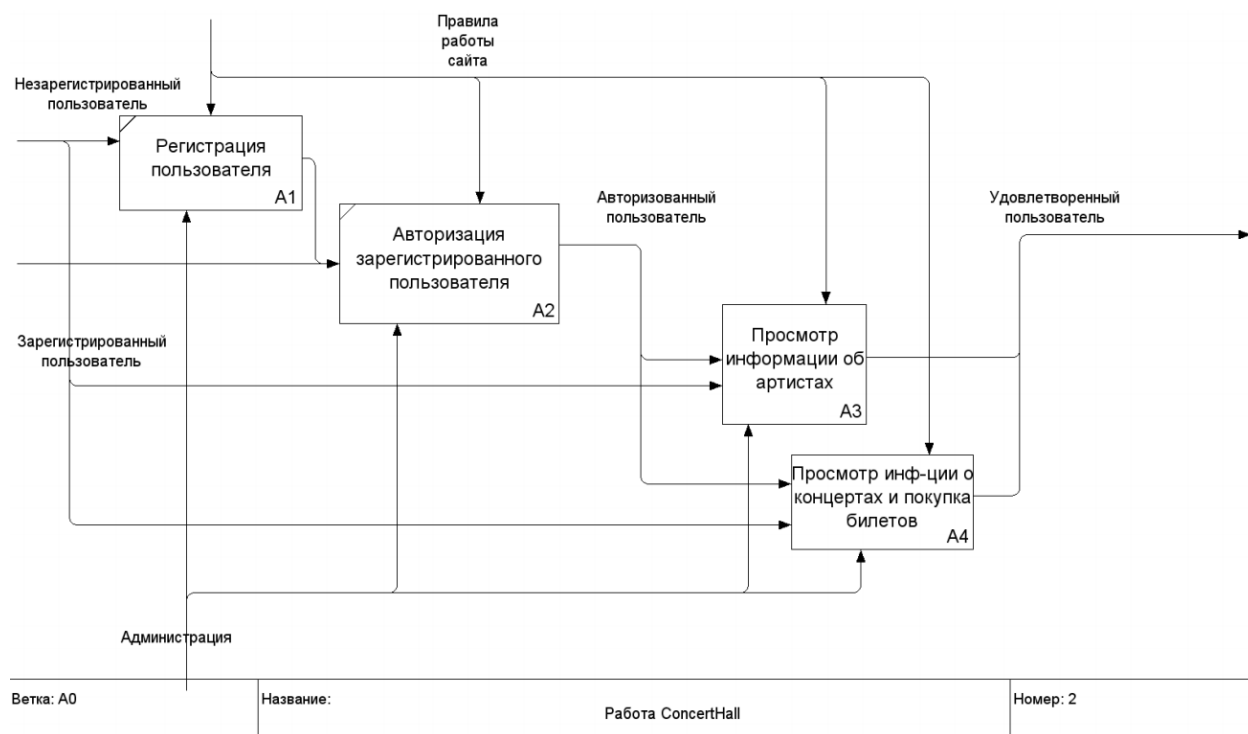


Рисунок 19 - Модель IDEF0 – уровень A0

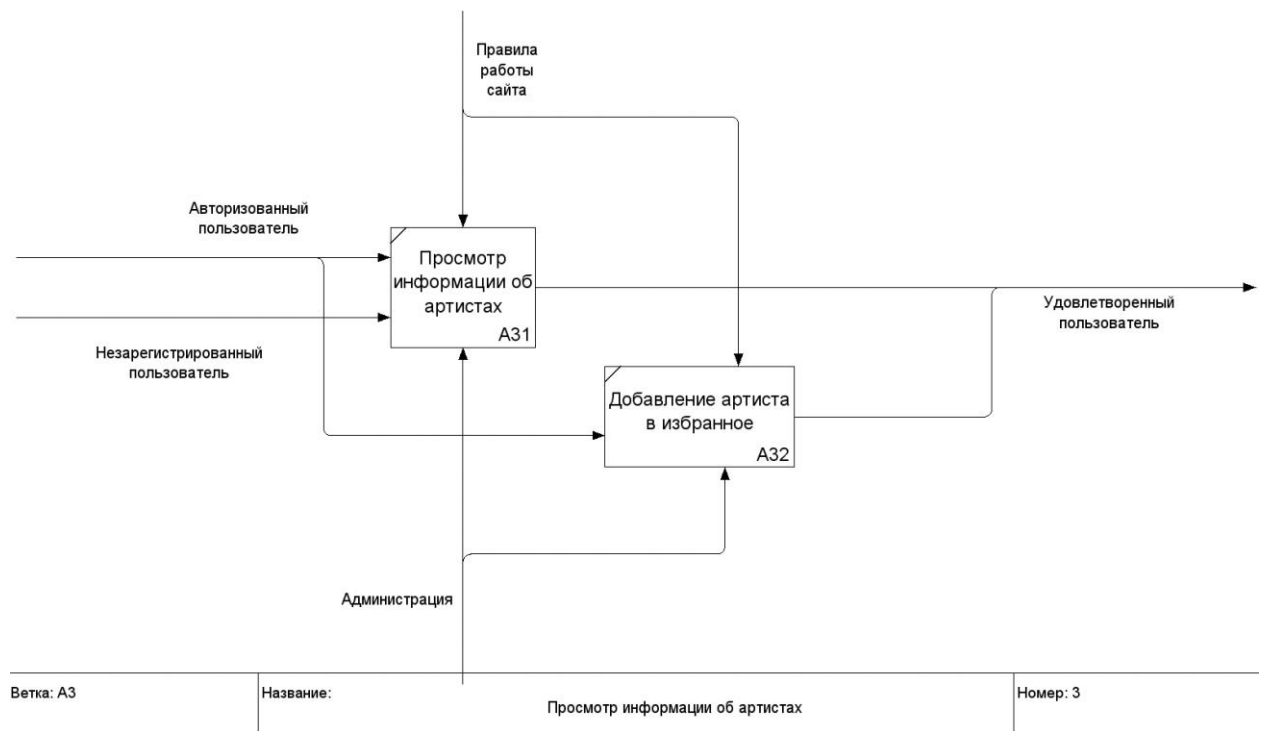


Рисунок 20 - Модель IDEF0 – уровень A3 (просмотр информации об артистах)

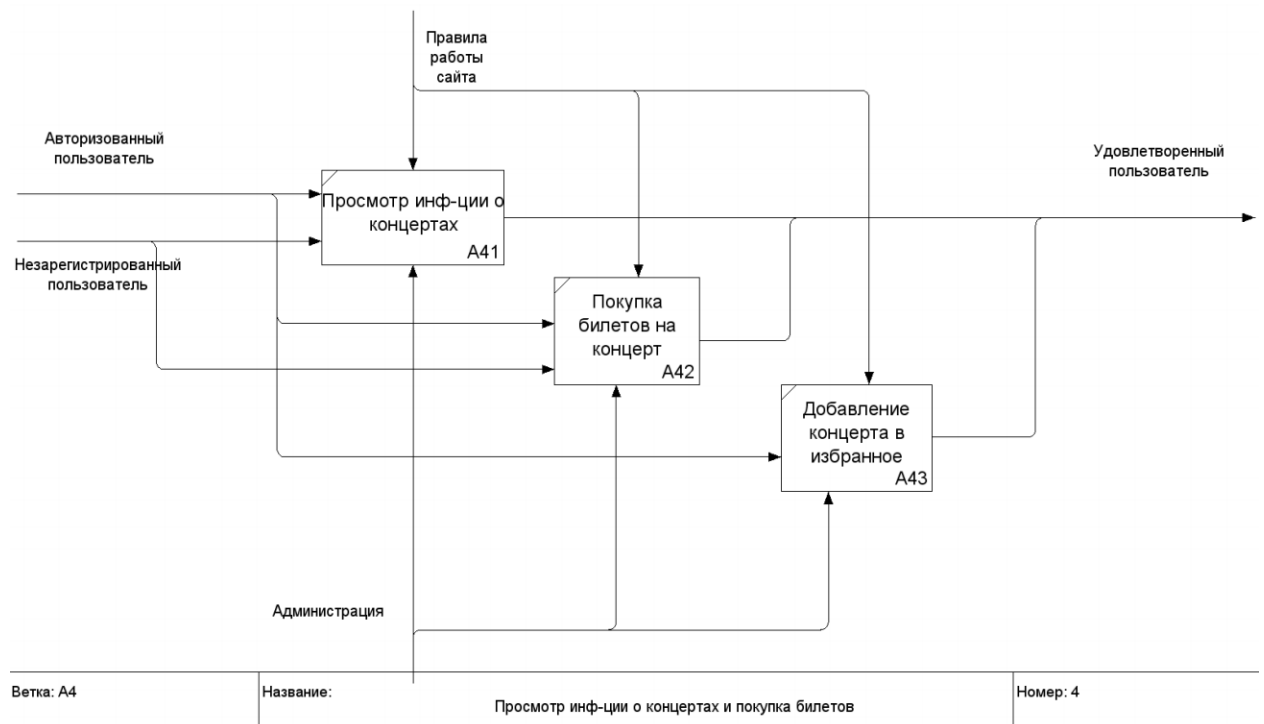


Рисунок 21 - Модель IDEF0 – уровень A4 (просмотр информации о концертах и покупка билетов)

4.10. ER-диаграмма

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

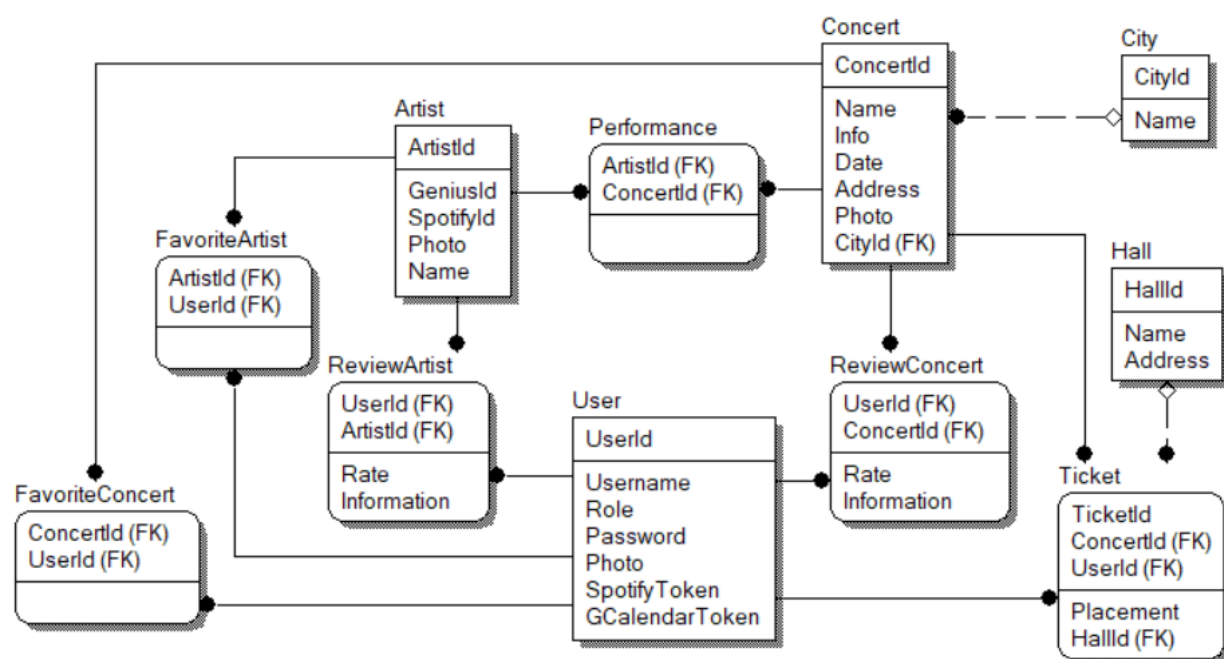


Рисунок 22 - ER-диаграмма проекта