МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра Информационных систем

Техническое задание

на разработку веб-приложения

«Сервис поиска отелей Bochka»

Исполнители

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Порядин

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Б.В. Исаченко

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Ю. Ткаченко

Заказчик

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.С. Тарасов

Воронеж 2024

СОДЕРЖАНИЕ

[1 Термины и сокращения 4](#_Toc167934045)

[2 Общие положения 5](#_Toc167934046)

[2.1 Полное наименование системы и ее условное обозначение 5](#_Toc167934047)

[2.2 Разработчики и заказчик 5](#_Toc167934048)

[2.3 Перечень документов, на основании которых создается приложение 6](#_Toc167934049)

[2.4 Плановые сроки начала и окончания работ по созданию приложения 6](#_Toc167934050)

[2.5 Цели и назначение создания приложения 7](#_Toc167934051)

[2.5.1 Цели создания приложения 7](#_Toc167934052)

[2.5.2 Назначение приложения 7](#_Toc167934053)

[3 Характеристика объекта автоматизации 8](#_Toc167934054)

[4 Требования к приложению 8](#_Toc167934055)

[4.1 Требования к приложению в целом 8](#_Toc167934056)

[4.2 Требования к функциям, выполняемым приложением 9](#_Toc167934057)

[4.3 Требования к видам обеспечения приложения 10](#_Toc167934058)

[4.3.1 Требования к структуре 10](#_Toc167934059)

[4.3.2 Требования к программному обеспечению сайта 11](#_Toc167934060)

[4.4 Общие технические требования к приложению 13](#_Toc167934061)

[4.4.1 Общие требования к оформлению и верстке страниц 13](#_Toc167934062)

[4.4.2 Требования к форматам и размеру данных в приложении 14](#_Toc167934063)

[4.4.3 Требования к защите информации 14](#_Toc167934064)

[5 Состав и содержание работ по созданию приложения 14](#_Toc167934065)

[6 Порядок оформления и предъявления заказчику результатов работы 15](#_Toc167934066)

[7 Пользовательские сценарии 16](#_Toc167934067)

[8 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие 19](#_Toc167934068)

[9 Требования к документированию 19](#_Toc167934069)

[10 Источники разработки 19](#_Toc167934070)

[Приложение А 20](#_Toc167934071)

[Приложение Б 21](#_Toc167934072)

[Приложение В 22](#_Toc167934073)

[Приложение Г 23](#_Toc167934074)

[Приложение Д 26](#_Toc167934075)

[Приложение Е 27](#_Toc167934076)

1. Термины и сокращения

В настоящей работе используются следующие термины и сокращения с соответствующими определениями:

* **Авторизация** – Предоставление определённому лицу прав на выполнение определённых действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий;
* **Авторизованный пользователь** – Пользователь, который успешно прошел процесс авторизации в приложении, предоставив свои учетные данные и подтвердив свою идентичность. Авторизованный пользователь имеет доступ ко всем основным функциям приложения;
* **Глубокая фильтрация** – Функциональность в веб-приложениях, которая позволяет пользователям настраивать широкий спектр параметров для поиска и отображения результатов, соответствующих их конкретным потребностям и предпочтениям;
* **Искусственный интеллект** – Набор технологий и алгоритмов, которые позволяют приложению анализировать большие объемы данных, выявлять закономерности и предсказывать поведение пользователей для персонализации рекомендаций и улучшения качества обслуживания;
* **Неавторизованный пользователь** – Пользователь, который еще не прошел процесс авторизации в приложении или не предоставил верные учетные данные для подтверждения своей идентичности. Неавторизованный пользователь имеет ограниченный доступ к функциям приложения;
* **Профиль (в веб-приложении)** – Учетная запись пользователя в веб-приложении, вход в которую осуществляется с помощью логина / номера телефона / e-mail и пароля. В учетной записи содержится информация о пользователе;
* **Сервер** – Это устройство, в частности компьютер, которое отвечает за предоставление услуг, программ и данных другим клиентам посредством использования сети;
* **СУБД** – Система управления базами данных. Комплекс программ, позволяющих создать базу данных (БД) и манипулировать данными;
* **Фреймворк** – Программные продукты, которые упрощают создание и поддержку технически сложных или нагруженных проектов. Фреймворк, как правило, содержит только базовые программные модули;
* **API** – Программный интерфейс приложения. Описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

1. Общие положения
   1. Полное наименование системы и ее условное обозначение

Полное наименование: Сервис поиска отелей с глубокой фильтрацией по времени заселения, местоположению и другим параметрам.

Условное обозначение приложения: Bochka – Сервис поиска отелей.

* 1. Разработчики и заказчик

Заказчик: старший преподаватель Тарасов Вячеслав Сергеевич, Воронежский Государственный Университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий.

Разработчики: 6 команда группы 3

Состав команды:

* студент Порядин Александр Вячеславович. Воронежский Государственный Университет, Факультет компьютерных наук, кафедра Информационных систем;
* студент Исаченко Богдан Вячеславович. Воронежский Государственный Университет, Факультет компьютерных наук, кафедра Информационных систем;
* студент Ткаченко Алексей Юрьевич. Воронежский Государственный Университет, Факультет компьютерных наук, кафедра Информационных систем.
  1. Перечень документов, на основании которых создается приложение

Данное приложение будет создаваться на основе следующих документов:

* Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 N 149-ФЗ (ред. от 12.12.2023);
* Федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ;
* Настоящее техническое задание, составленное в соответствии с ГОСТ 34.602 – 2020.
  1. Плановые сроки начала и окончания работ по созданию приложения

Плановый срок начала работ – март 2024 г.

Плановый срок окончания работ – июнь 2024 г.

* 1. Цели и назначение создания приложения
     1. Цели создания приложения

Целями создания приложения являются:

* Создание системы, которая позволит пользователям легко и удобно бронировать отели для размещения во время путешествий или поездок;
* Увеличение доходов заказчика за счет продажи гостиничных номеров через онлайн-платформу;
* Продажа гостиничных номеров для конкретной аудитории, предпочитающей определенные условия размещения, такие как питание, наличие специальных удобств и т.д., что позволяет удовлетворить разнообразные потребности клиентов;
* Обеспечение информативной карточки каждого отеля с подробным описанием и фотографиями, что помогает пользователям принимать осознанные решения.
  + 1. Назначение приложения

Сайт позволяет решать следующие задачи:

* Искать отели с глубокой фильтрацией по времени заселения, местоположению и другим параметрам;
* Просматривать подробную информацию о каждом отеле, включая фотографии, описание, услуги и прочее;
* Бронировать отель непосредственно через веб-сайт;
* Сохранять историю бронирования для последующего доступа;
* Создавать учётную запись пользователя;

1. Характеристика объекта автоматизации

Объектом автоматизации является процесс организации деятельности в сфере бронирования отелей.

1. Требования к приложению
   1. Требования к приложению в целом

Разрабатываемое приложение должно удовлетворять следующим основным требованиям:

* Приложение должно корректно работать в современных веб-браузерах;
* Приложение должно реализовывать основные функциональные задачи, соответствующие целям проекта;
* Созданное приложение должно иметь архитектуру, соответствующую шаблону клиент-серверного приложения, с разделением на back-end и front-end;
* Взаимодействие между back-end и front-end должно осуществляться посредством REST API.

У приложения есть следующие перспективы развития:

* Создание мобильной версии приложения для удобного доступа к услугам в любое время и из любой точки;
* Внедрение технологий искусственного интеллекта для персонализации рекомендаций и улучшения качества обслуживания пользователей;
* Добавление блога, который будет содержать полезные статьи, советы и рекомендации по выбору отелей, путешествиям и туризму в целом. Это позволит привлечь новых пользователей за счет предоставления ценной информации и увеличит интерес к приложению.
  1. Требования к функциям, выполняемым приложением

Разрабатываемое приложение должно соответствовать следующим функциональным требованиям:

Неавторизованный пользователь должен обладать возможностью:

* Авторизоваться/зарегистрироваться в приложении;
* Получать информацию о предложениях отелей с глубокой фильтрацией по различным критериям, таким как местоположение, цена, удобства и другие параметры;
* Просматривать детальную информацию о каждом отеле, включая фотографии и описания;
* Выполнять поиск отелей по различным критериям.

Авторизованный пользователь (в роли клиента) должен обладать возможностью:

* Получать информацию о предложениях отелей с глубокой фильтрацией по различным критериям, таким как местоположение, цена, удобства и другие параметры;
* Просматривать детальную информацию о каждом отеле, включая фотографии и описания;
* Выполнять поиск отелей по различным критериям;
* Просматривать свою историю бронирования;
* Возможность бронировать отель;

Авторизованный пользователь (в роли админа) должен обладать возможностью:

* Добавлять информацию об отеле, включая описание, фотографии, цены и доступные удобства;
* Обновлять информацию об отеле, в том числе актуализация цен и доступности номеров;
  1. Требования к видам обеспечения приложения
     1. Требования к структуре

Для Frontend:

Сервис должен быть реализован в соответствии с архитектурным паттерном Module — паттерн, который используется для организации кода в отдельные модули или компоненты. Цель использования паттерна Module - избежать конфликтов и обеспечить лучшую структурированность, масштабируемость и повторное использование кода. Каждый модуль содержит свою собственную область видимости, что позволяет исключить конфликты между переменными или функциями из разных модулей.

Для Backend:

Приложение должно быть реализовано в соответствии с подходом MVC (Model – View – Controller) — паттерн разработки, разделяющий архитектуру приложения на три модуля: модель (Model), представление или вид (View), контроллер (Controller).

* Model – это основная логика приложения. Отвечает за данные, методы работы с ними и структуру программы. Модель реагирует на команды из контроллера и выдает информацию и/или изменяет свое состояние. Она передает данные в представление;
* View – отвечает за визуализацию информации, которую он получает от модели. View отображает данные на уровне пользовательского интерфейса. Например, в виде таблицы или списка. Представление определяет внешний вид приложения и способы взаимодействия с ним;
* Controller – обеспечивает взаимодействие с системой: обрабатывает действия пользователя, проверяет полученную информацию и передает ее модели. Контроллер определяет, как приложение будет реагировать на действия пользователя. Также контроллер может отвечать за фильтрацию данных и авторизацию.

Требования к веб-интерфейсу:

* Веб-приложение должно быть доступно и корректно работать в современных веб-браузерах на настольных компьютерах и мобильных устройствах;
* Реализация пользовательского интерфейса должна быть интуитивно понятной, удобной и эстетически привлекательной для пользователей;
* Веб-интерфейс должен быть разработан с использованием фреймворка React.js версии 17 или выше.

Требования к серверной части:

* Серверная часть приложения должна обрабатывать запросы и обеспечивать взаимодействие с базой данных и другими внешними сервисами;
* Серверная часть должна быть реализована на языке программирования Java версии 17 или выше;
* Для обработки HTTP-запросов и создания API должен использоваться фреймворк Spring Boot.

В качестве системы управления базами данных необходимо использовать PostgreSQL версии 13 или выше.

* + 1. Требования к программному обеспечению сайта

Для реализации серверной части приложения будут использоваться следующие средства:

* Язык программирования Java;
* Фреймворк Spring Boot;
* СУБД PostgreSQL;
* Инструмент для создания документации API Swagger.

Для реализации клиентской части приложения будут использоваться следующие средства:

* Язык программирования JavaScript;
* Фреймворк React.

Для развёртывания приложения будут использоваться следующие средства:

* Docker для автоматизации развёртывания;
* Nginx для обеспечения поддержки SSL и проксирования запросов к back-end приложению.

Инструменты для ведения документации:

* Miro – платформа для совместной работы распределенных команд;
* Swagger – фреймворк для спецификации REST API;
* Figma – онлайн-сервис для дизайнеров, веб-разработчиков и маркетологов. Он предназначен для создания прототипов сайтов или приложений, иллюстраций и векторной графики.

Дополнительный инструментарий:

* Git – распределённая система управления версиями;
* GitHub – платформа разработки программного обеспечения с открытым исходным кодом, представляющая систему управления репозиториями программного кода для Git;
* GitHub Projects – визуальный инструмент, обеспечивающий эффективность командной работы на любом проекте.

В качестве преимуществ выбранных технологий можно отметить следующее:

Для Java и Spring Boot:

* Готовые решения для реализации RESTful архитектуры;
* Удобные инструменты для работы с PostgreSQL

Для PostgreSQL:

* Функциональность;
* Высокая надежность и производительность;
* Бесплатное и открытое ПО.

Для JavaScript и React:

* Кросс-платформенность;
* Поддержка разными браузерами.
  1. Общие технические требования к приложению
     1. Общие требования к оформлению и верстке страниц

Все страницы сайта должны быть выполнены в едином стиле, соответствующем тематике отельного бронирования. Цветовая палитра и стили шрифтов должны быть гармонично подобраны и привлекательны для пользователей. Приложение должно содержать разработанный логотип, отражающий его назначение и стиль.

Необходимо корректное и одинаковое отображение страниц сайта в

следующих браузерах:

* Google Chrome 122.0.6261.128/129;
* Yandex Browser 23.11.3.955;
* Microsoft Edge 121.0.2277.83;
* Safari 16.5.2;
* Mozilla Firefox 123.0.1.

Верстка сайта должна быть адаптирована под популярные разрешения экранов, чтобы обеспечить удобство использования и приятный внешний вид для всех пользователей.

* + 1. Требования к форматам и размеру данных в приложении
* Сайт должен поддерживать загрузку изображений в форматах JPEG, PNG с разрешением, соответствующим различным типам контента и требованиям дизайна;
* Текстовые данные должны быть представлены в формате HTML для обеспечения удобства редактирования и форматирования контента.
  + 1. Требования к защите информации

Для обеспечения безопасности информации будет использоваться механизм JWT-токенов. Даже в случае получения злоумышленником такого токена, который предоставляет доступ ко всем функциям приложения, его действие будет ограничено заданным периодом времени, после чего токен станет недействительным и потребуется получить новый.

1. Состав и содержание работ по созданию приложения

Состав и содержание работы по созданию приложения включает в себя следующие этапы:

* Сбор необходимой информации, анализ предметной области, анализ конкурентов, определение функциональных и нефункциональных требований для создания приложения;
* Создание сценариев использования веб-сервиса по поиску отелей, разработка интерфейса, разработка модели БД;
* Разработка рабочего приложения, в которую входит написание программного кода и корректирование работы проекта;
* Тестирование и отладка: проверка работоспособности веб-сервиса согласно установленным требованиям, проверка корректности документации. Устранение обнаруженных ошибок;
* Проведение тестовой эксплуатации веб-сервиса с небольшим числом пользователей. Доработка и улучшение сервиса на основе обратной связи.

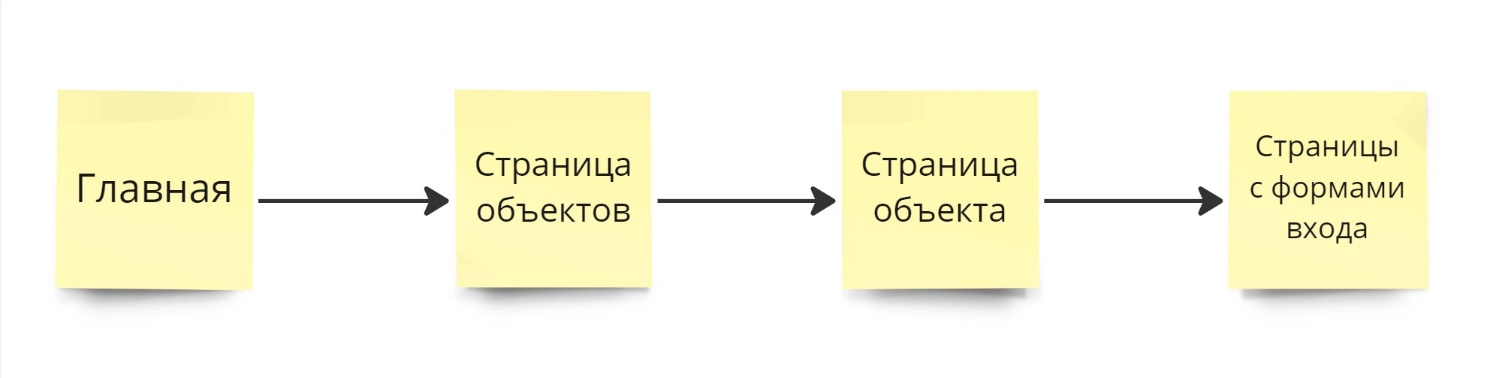
1. Порядок оформления и предъявления заказчику результатов работы

Предварительные отчеты по работе будут проводится во время рубежных аттестаций:

* 1 аттестация (середина марта 2024) – создан репозиторий на GitHub, распределены задачи проекта в канбан-доске GitHub Projects, создан проект Miro с UML диаграммами и пользовательскими сценариями. Написано техническое задание, а также создана презентация по проекту, видео и написано сопроводительное письмо.
* 2 аттестация (конец апреля 2024) – разработан интерфейс приложения. Реализована логика сайта и взаимодействие базы данных с сервером;
* 3 аттестация (начало июня 2024) – проведена отладка и доработка кода, проведено тестирование по работе системы, предоставлен курсовой проект, выполнены завершающие работы по доработке приложения, предоставлена готовая система.

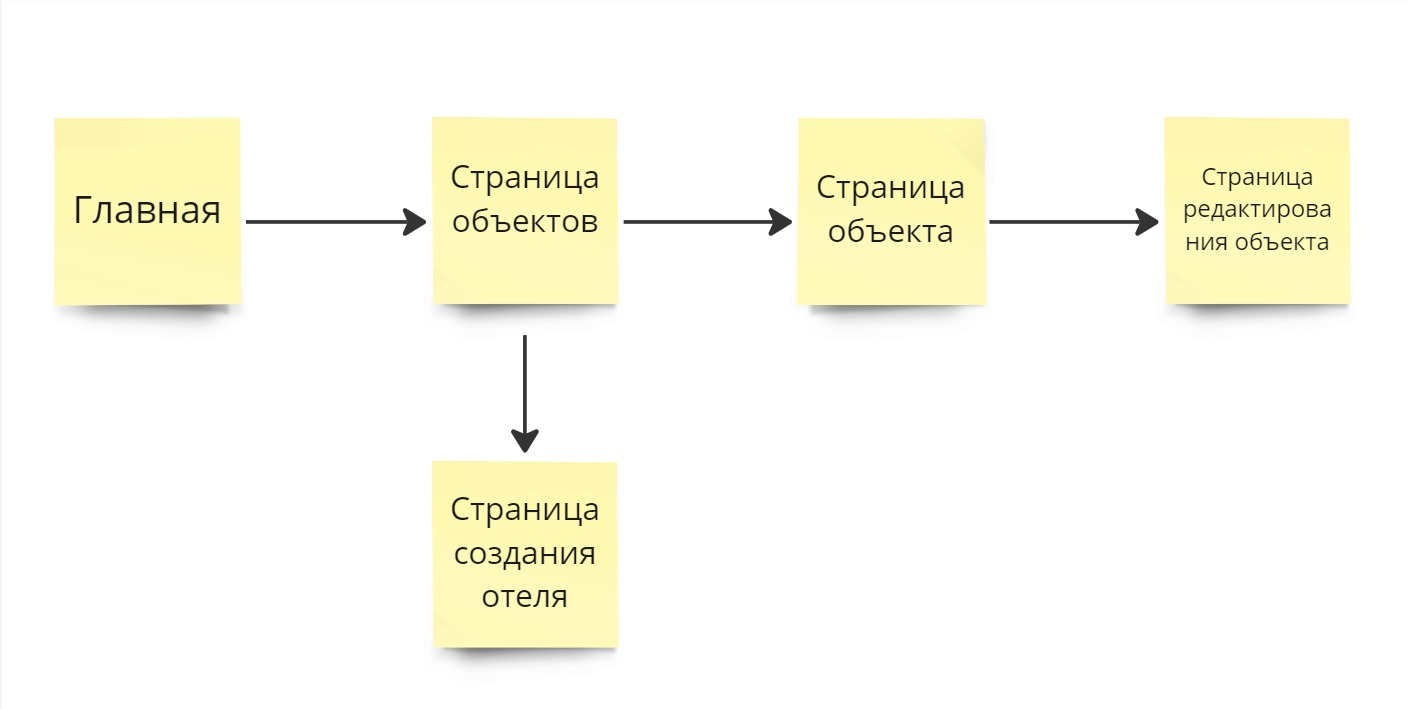
1. Пользовательские сценарии

Для неавторизованного пользователя:



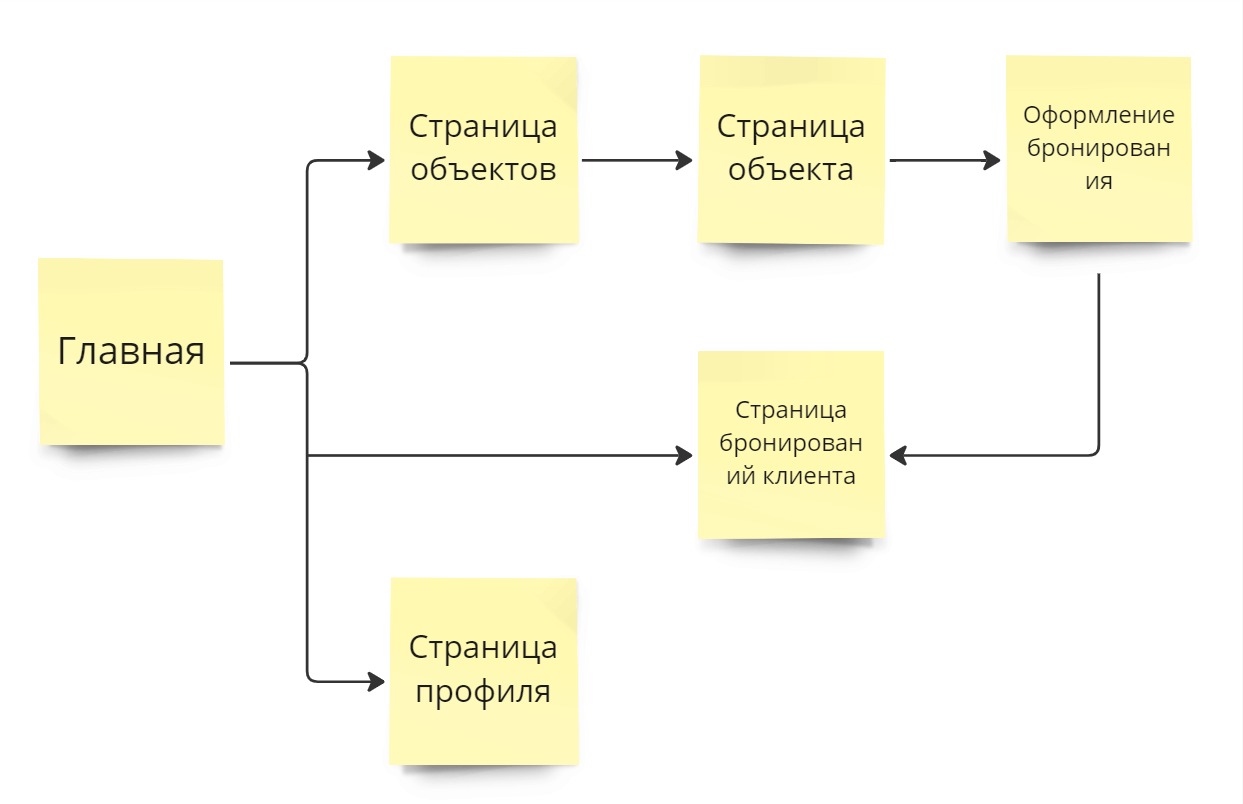
1. Сценарий для неавторизованного пользователя

Для авторизованного пользователя (в роли админа):



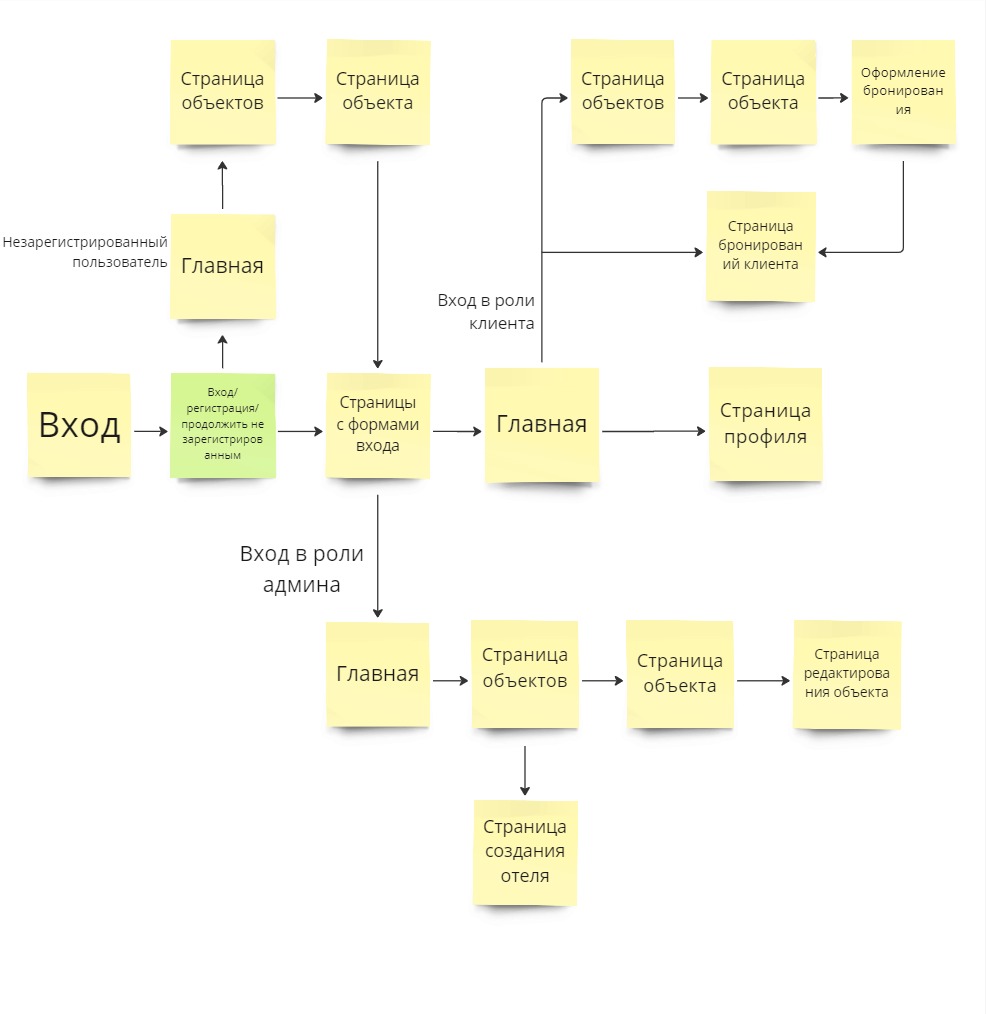
1. Сценарий для админа

Для авторизованного пользователя (в роли клиента):



1. Сценарий для клиента

Общий сценарий пользования:



1. Общий сценарий
2. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие

Перед внедрением автоматизированной системы необходимо удостовериться в соответствии программно-технических средств требованиям, на которых будет развернуто программное обеспечение системы. Также следует провести предварительное тестирование и опытную эксплуатацию системы.

1. Требования к документированию

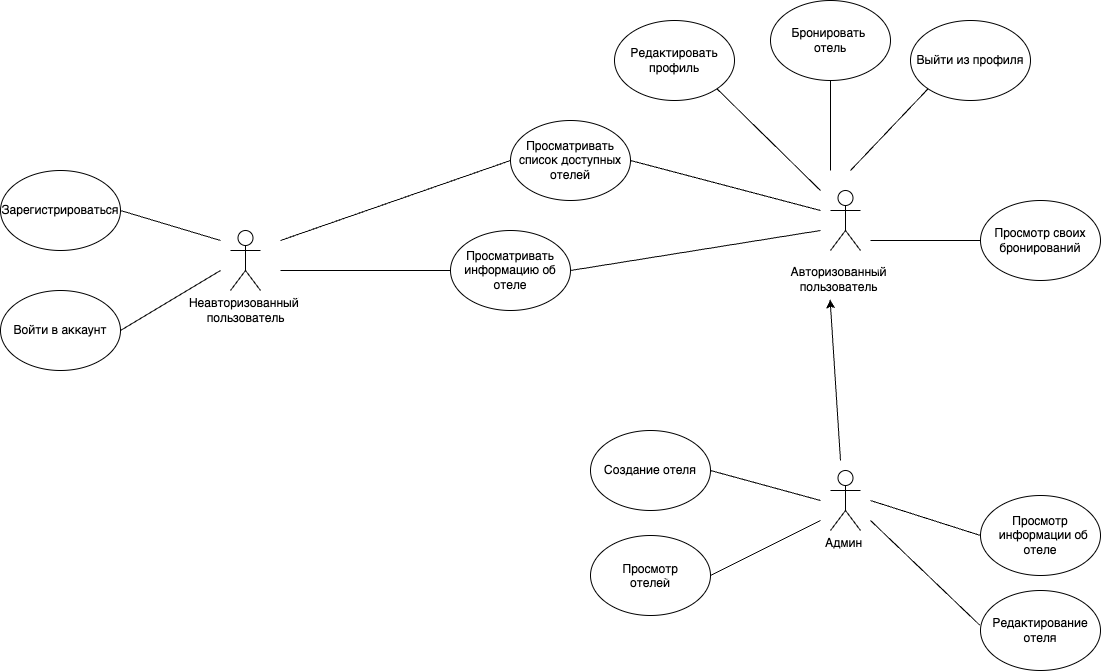
Перечень документов, подлежащих разработке:

* Курсовой проект.

1. Источники разработки

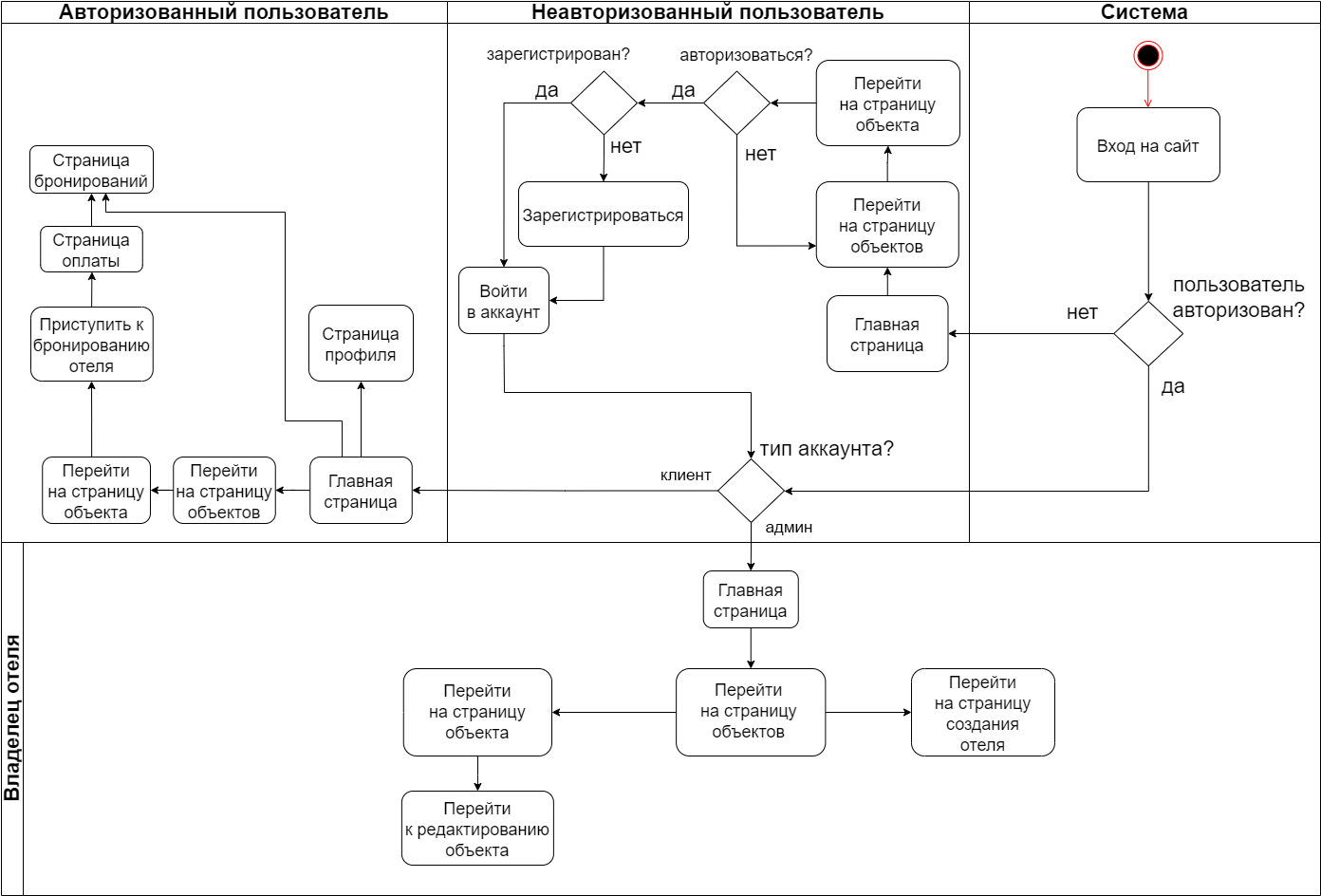
* ГОСТ 34.602 – 2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы;
* ГОСТ 34.601 – 90. Автоматизированные системы. Стадии создания;
* Система бронирования отелей Booking (<https://www.booking.com>).

Приложение А



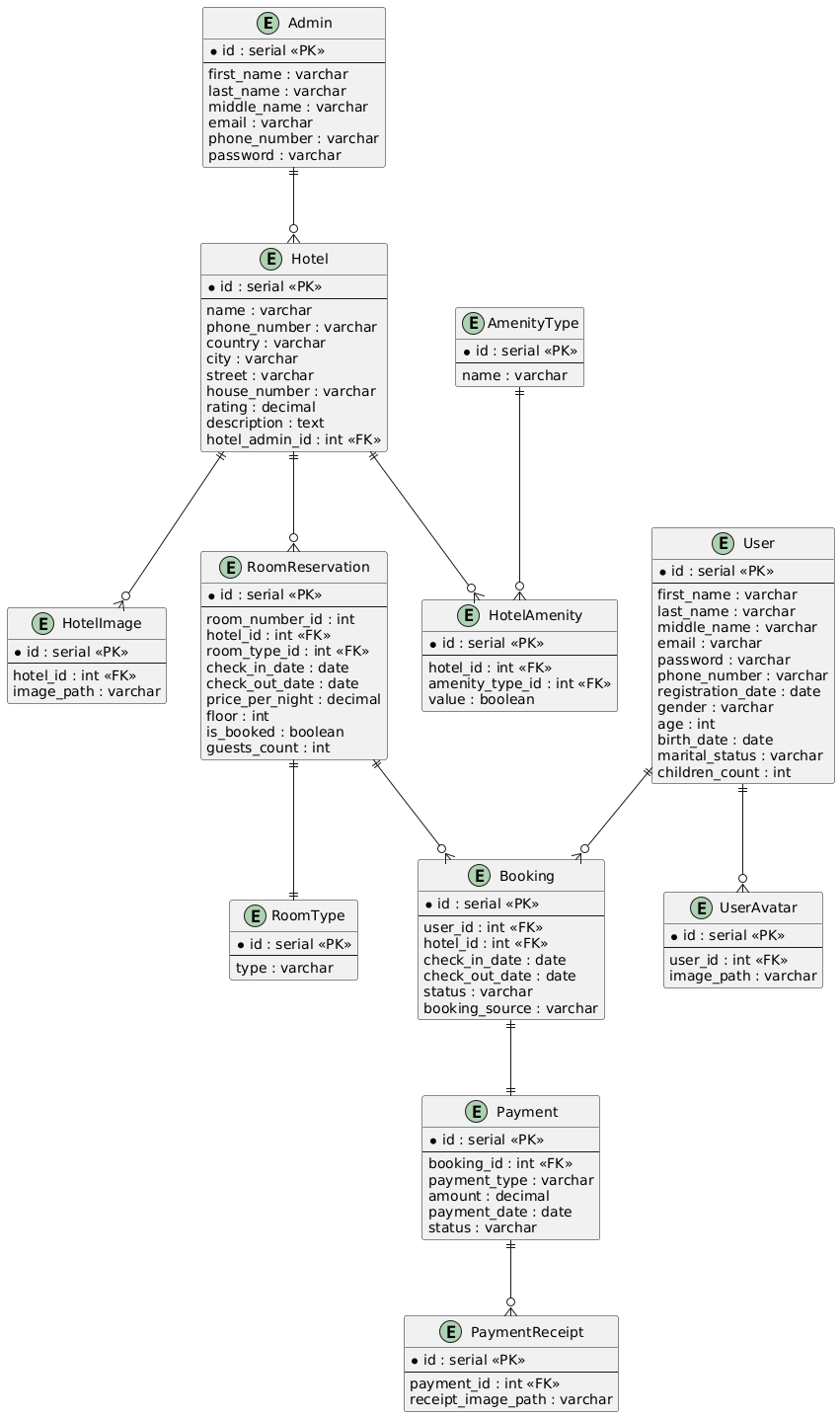
1. Use Case диаграмма

Приложение Б



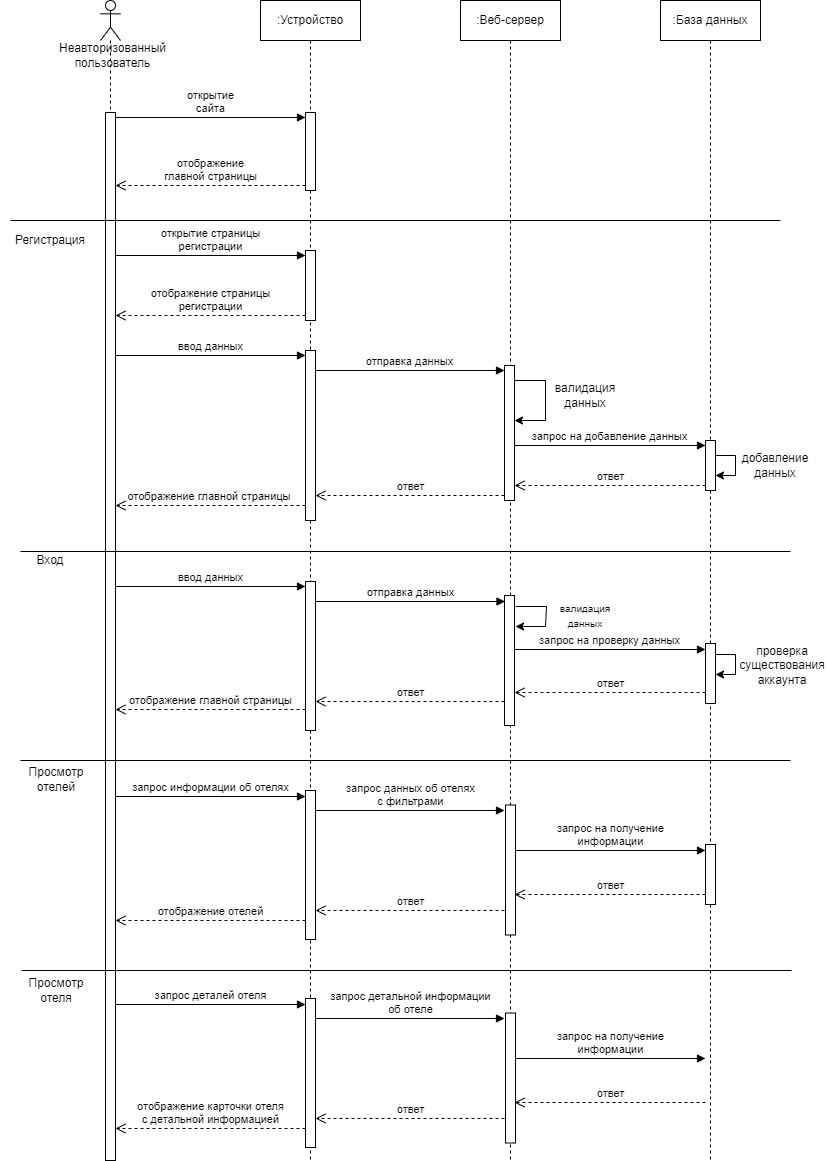
1. Activity диаграмма

Приложение В

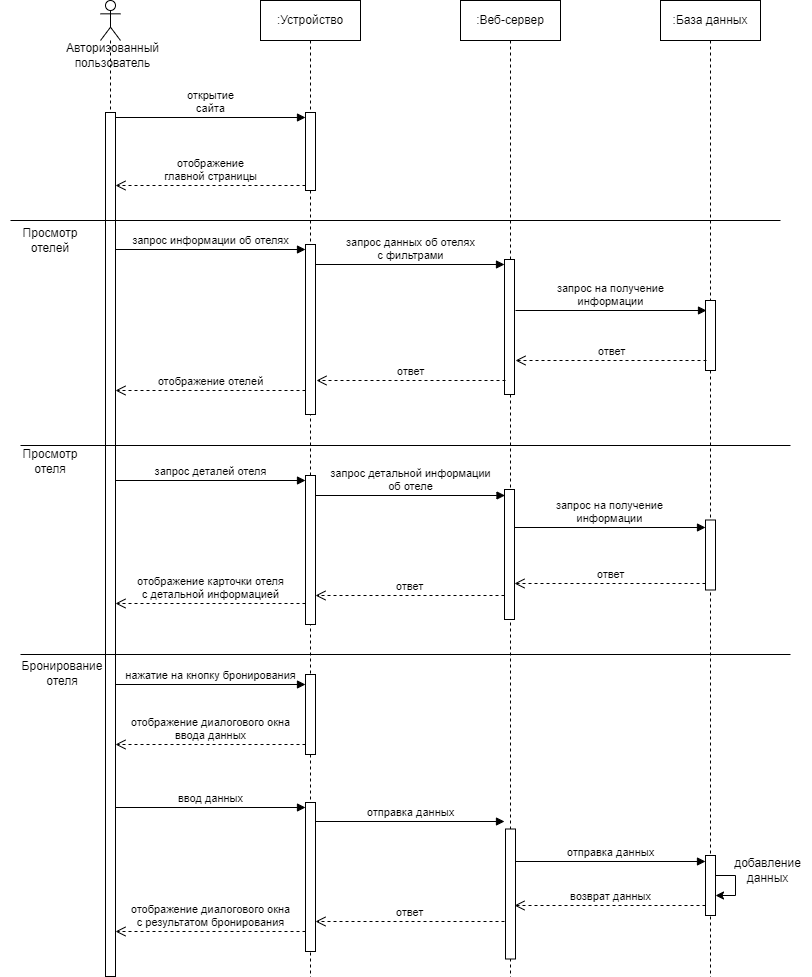


1. ER-диаграмма

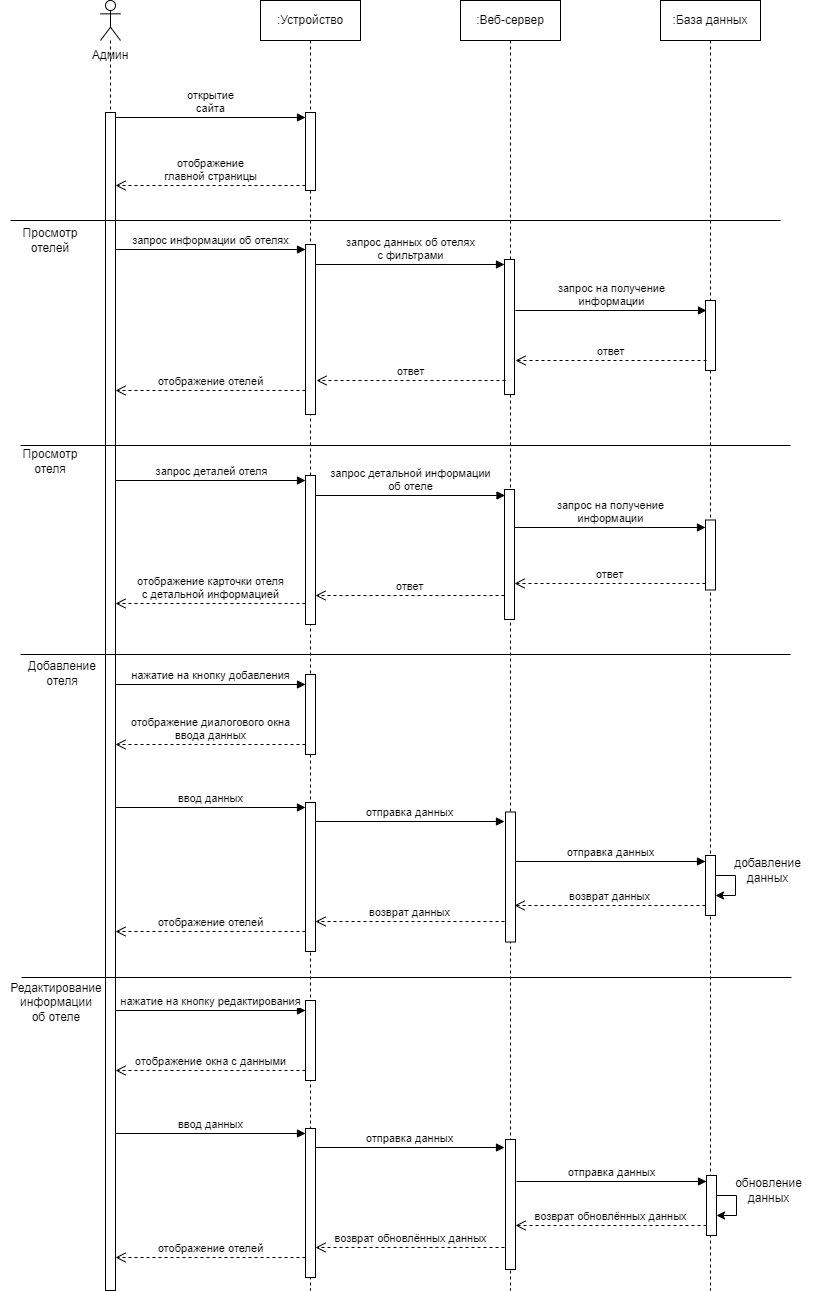
Приложение Г



1. Sequence диаграмма неавторизованного пользователя

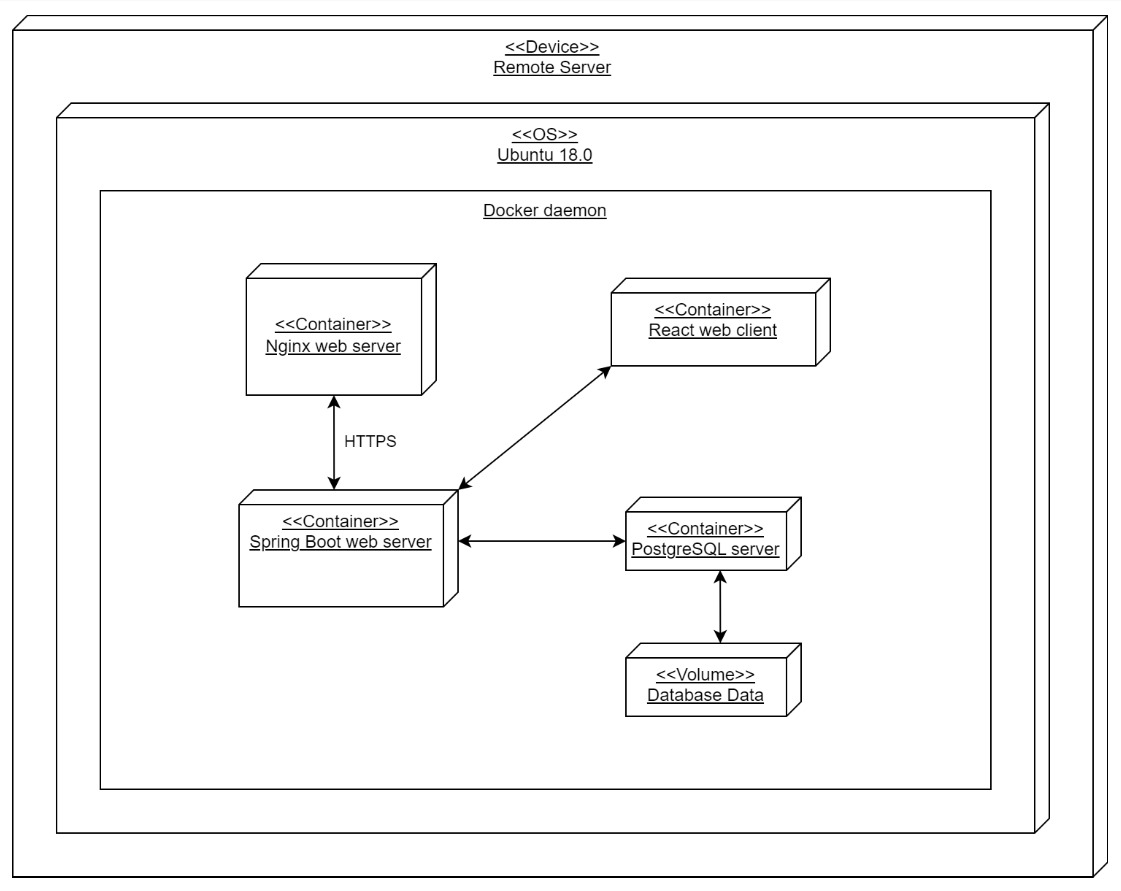


1. Sequence диаграмма клиента



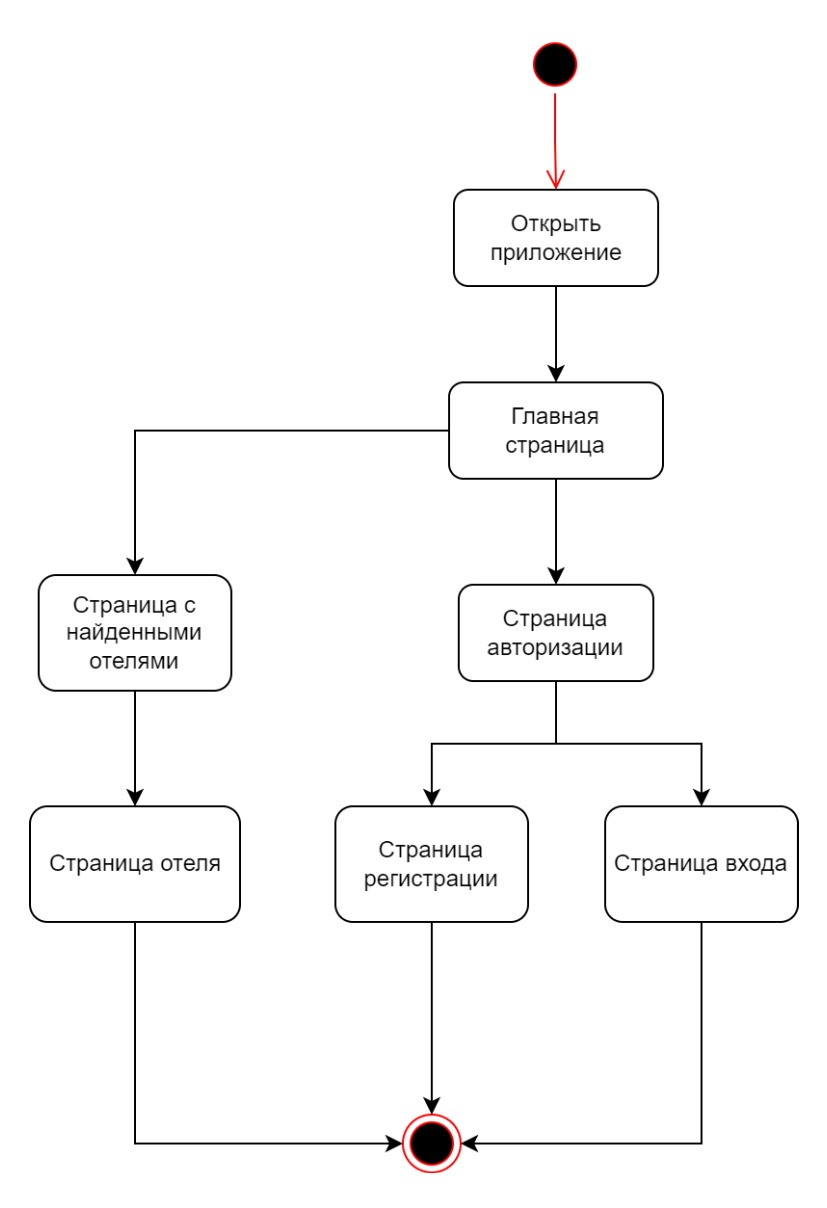
1. Sequence диаграмма админа

Приложение Д

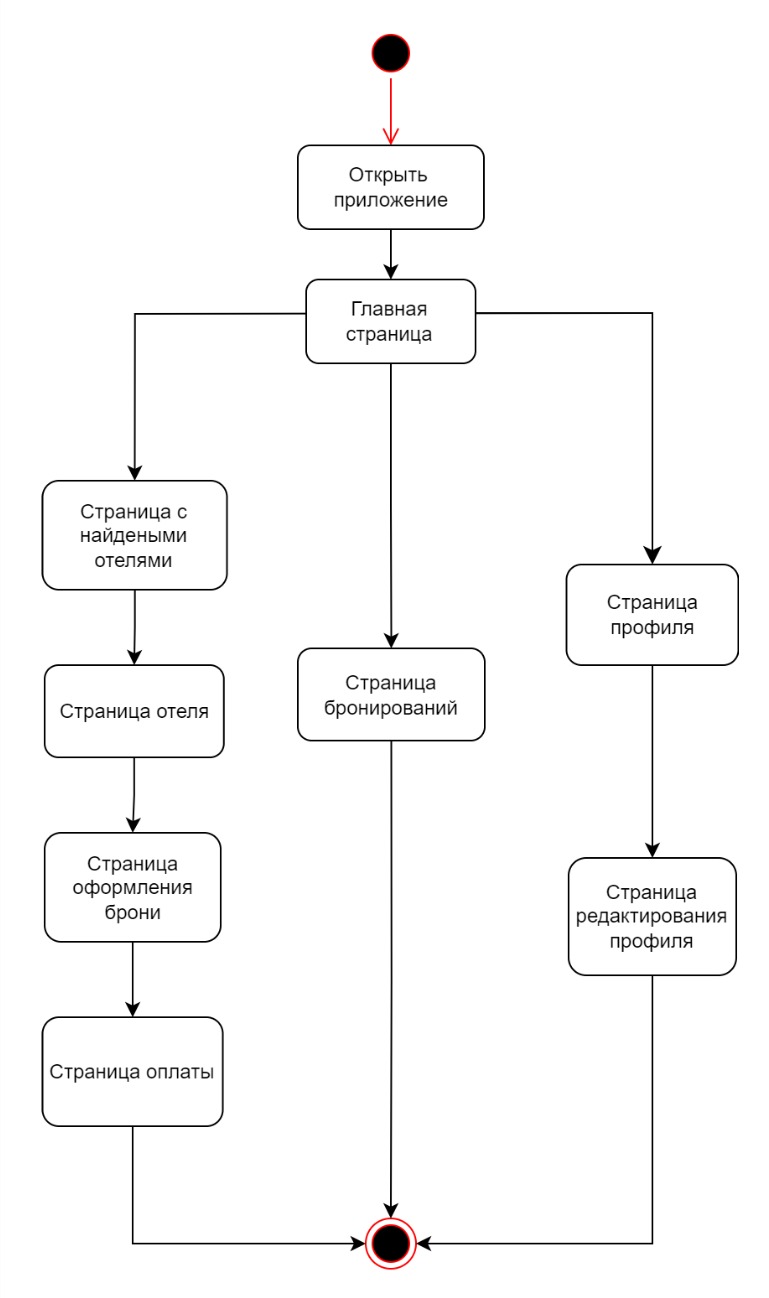


1. Deployment диаграмма

Приложение Е



1. Statechart диаграмма неавторизованного пользователя



1. Statechart диаграмма клиента



1. Statechart диаграмма админа