

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

**Отчет**

Домашняя работа 1: Технический дизайн микросервисов

Выполнил:  
Казарян Тигран

Группа К3441

Проверил:  
Добряков Д. И.

Санкт-Петербург

2025 г.

## Задача

Необходимо спроектировать набор следующих диаграмм:

общая архитектура решения (сервисы и их взаимосвязи, клиент-серверное взаимодействие);

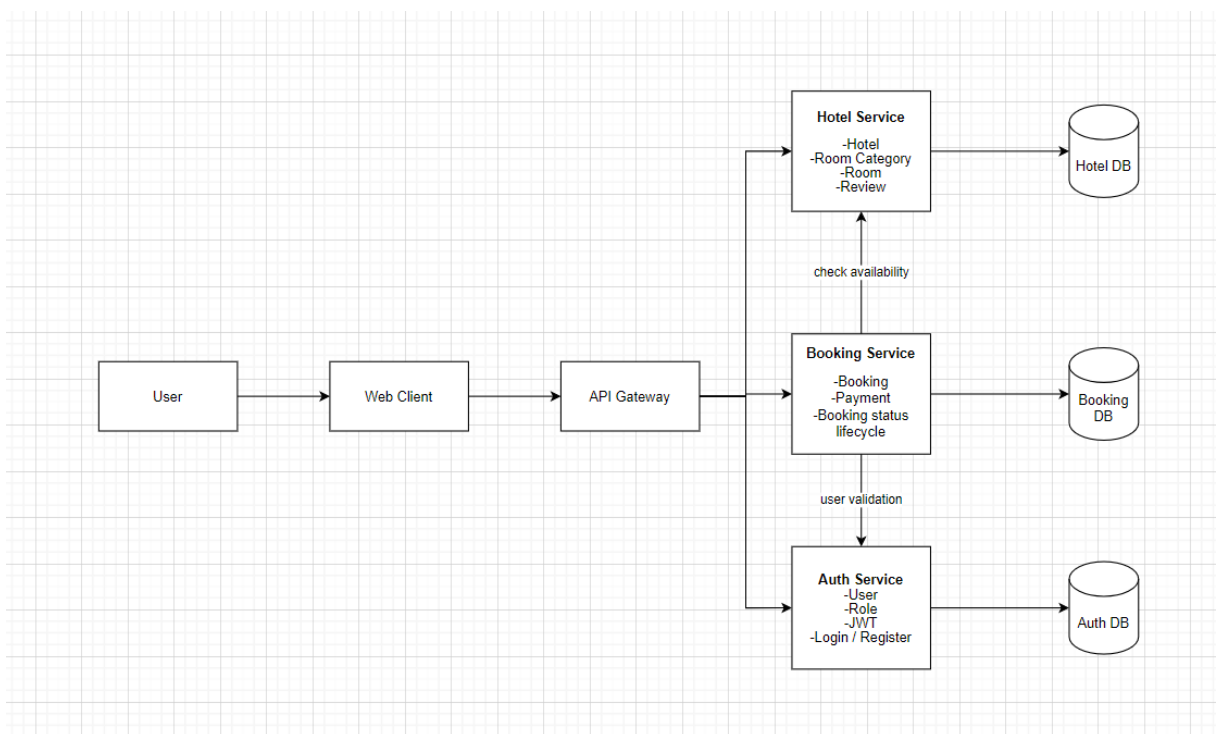
диаграмма компонентов;

диаграммы БД по каждому сервису;

диаграммы основных пользовательских сценариев (те сценарии, которые позволяют вашим приложением полноценно воспользоваться, пройти весь путь).

## Ход работы

### 1. Общая архитектура решения



Система бронирования отеля реализована в виде трёх сервисов, объединённых через API Gateway. Пользователь взаимодействует с системой через веб-браузер, при этом все клиентские запросы в первую очередь поступают в API Gateway, который отвечает за маршрутизацию запросов к соответствующим сервисам и не содержит собственной бизнес-логики.

Auth Service отвечает за регистрацию и аутентификацию пользователей системы. В рамках данного сервиса осуществляется хранение учетных данных пользователей, управление ролями и правами доступа, а также

выпуск и проверка JWT-токенов. Auth Service используется другими сервисами для подтверждения подлинности пользователя при выполнении операций, требующих авторизации. Сервис работает исключительно со своей базой данных AuthDB, в которой хранятся сведения о пользователях и их ролях.

Hotel Service нужен для хранения и предоставления информации об отеле и номерах. В нем содержится логика управления отелями, категориями номеров, отдельными номерами, а также отзывами пользователей. Hotel Service используется клиентским приложением для отображения каталога отеля и доступных номеров, а также сервисом бронирования для проверки доступности номеров на выбранные даты. Все данные, относящиеся к структуре отеля и отзывам, хранятся в собственной базе данных HotelDB.

Booking Service реализует основную бизнес-логику процесса бронирования. Он отвечает за создание и управление бронированиями, расчёт стоимости проживания, обработку платежей и хранение истории бронирований пользователей. При выполнении операций бронирования сервис взаимодействует с Hotel Service для проверки доступности номеров и с Auth Service для получения информации об авторизованном пользователе. Booking Service использует отдельную базу данных BookingDB, в которой хранятся данные о бронированиях и платежах.

Каждый сервис системы работает только со своей базой данных, что соответствует принципу изоляции данных в микросервисной архитектуре. Межсервисное взаимодействие осуществляется по HTTP через REST API, а все внешние запросы проходят через API Gateway.

## 2. Диаграмма компонентов

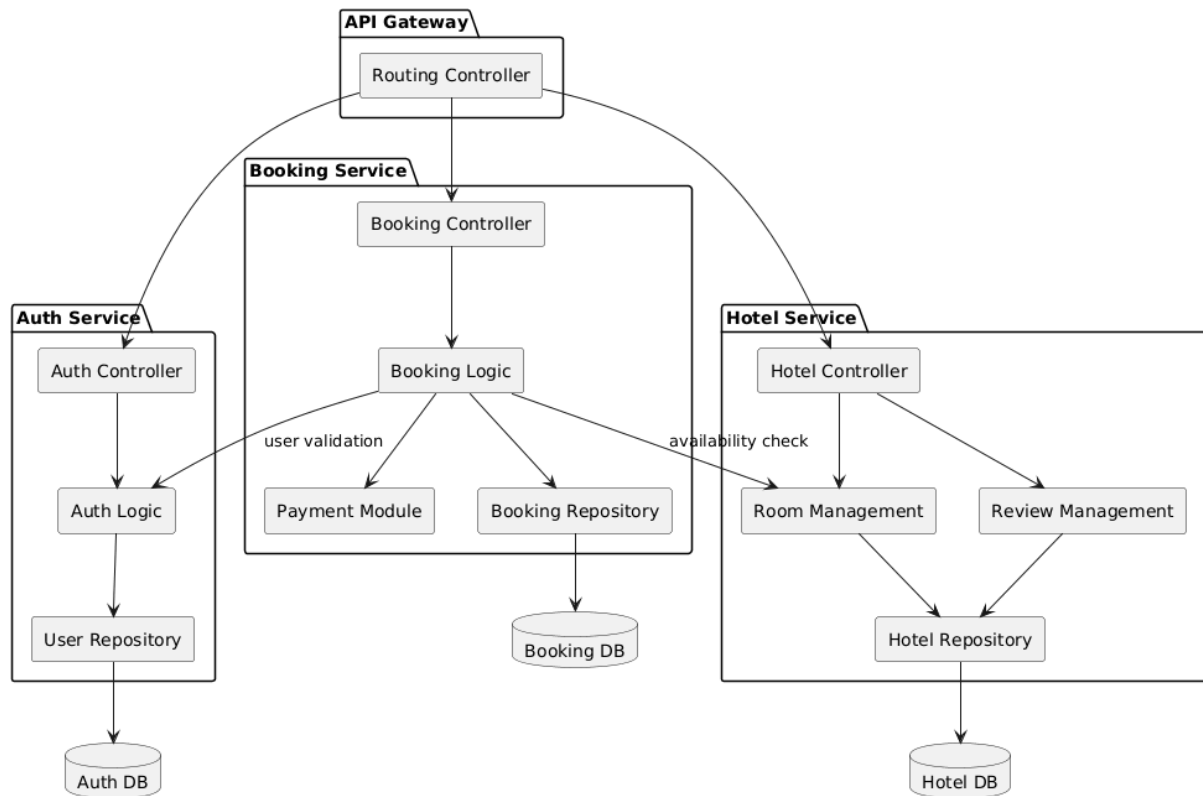


Диаграмма компонентов показывает внутреннюю структуру сервисов системы бронирования отеля и их взаимодействие. Все входящие запросы от клиента проходят через API Gateway, который выполняет маршрутизацию без реализации бизнес-логики.

Auth Service включает контроллер аутентификации, компонент бизнес-логики и репозиторий пользователей, обеспечивающие регистрацию, вход в систему и работу с базой данных AuthDB.

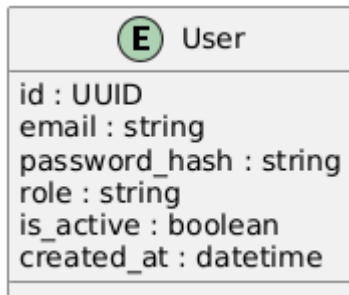
Hotel Service отвечает за управление данными об отеле, номерах и отзывах и использует собственную базу данных HotelDB.

Booking Service реализует основную бизнес-логику бронирования, включая управление бронированиями и платежами, и работает с базой данных BookingDB.

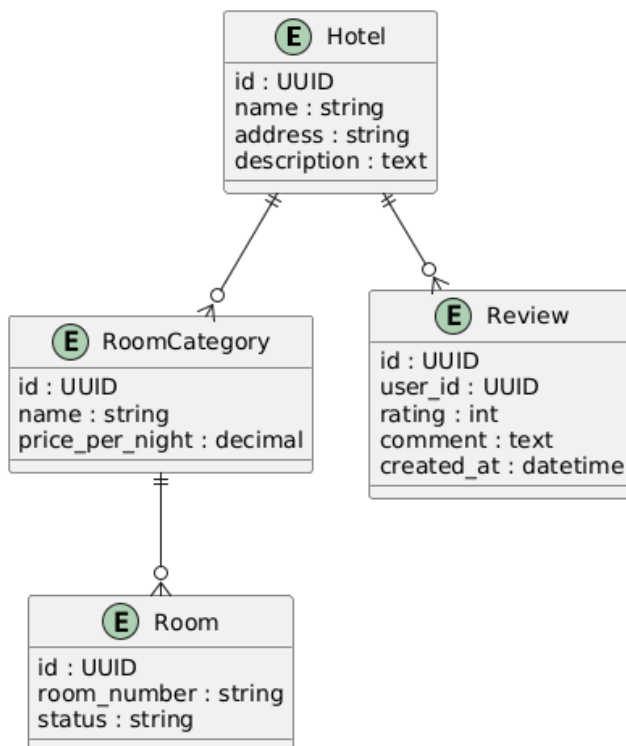
Взаимодействие между сервисами осуществляется только на уровне бизнес-логики, при этом прямой доступ к базам данных между сервисами отсутствует

### 3. Схемы баз данных

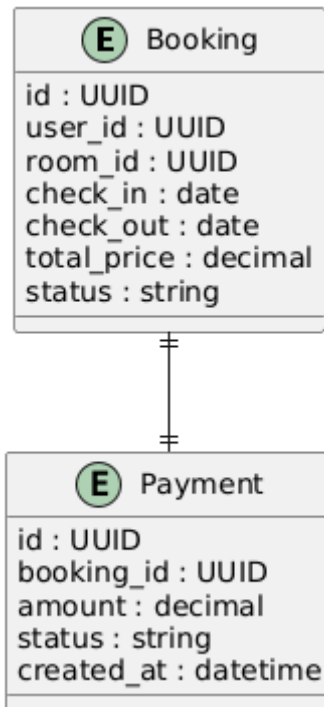
#### Auth Service — AuthDB



#### Hotel Service — HotelDB



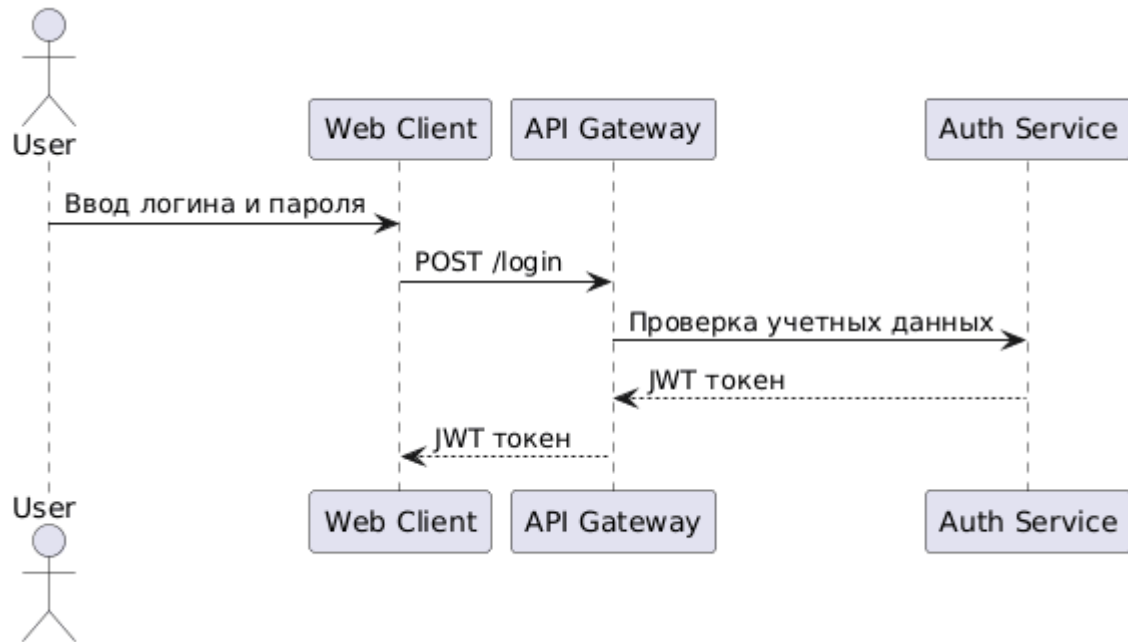
## Booking Service — BookingDB



#### 4. Пользовательские сценарии

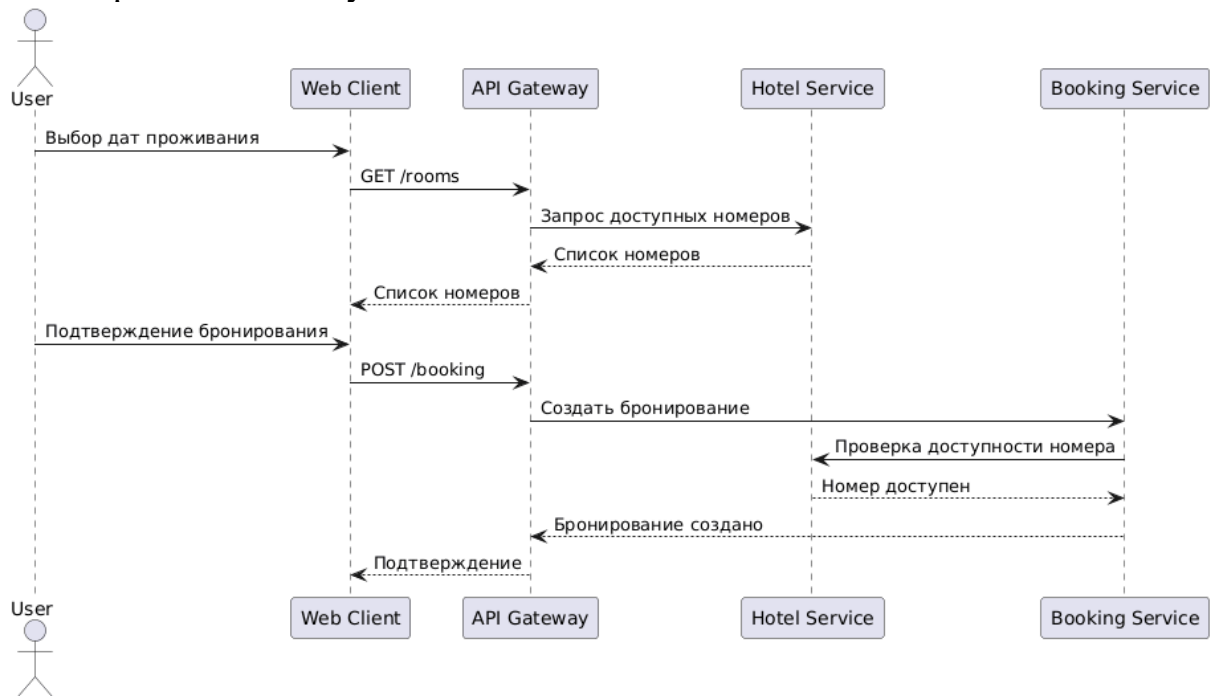
##### Сценарий: Авторизация пользователя

Пользователь вводит учетные данные в клиентском приложении, после чего запрос на вход передается через API Gateway в сервис авторизации. Auth Service выполняет проверку переданных данных и, в случае успешной аутентификации, формирует JWT-токен, который возвращается пользователю через API Gateway и используется для дальнейших авторизованных запросов.



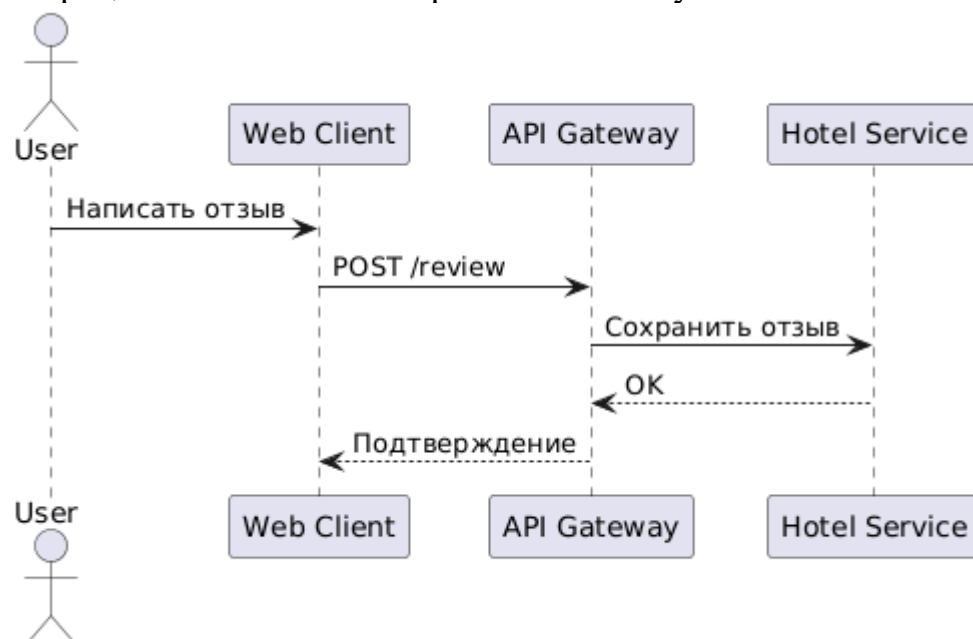
### Сценарий: Поиск и бронирование номера

Пользователь выбирает даты проживания, после чего клиентское приложение запрашивает через API Gateway список доступных номеров у сервиса управления отелем. После выбора номера пользователь подтверждает бронирование, и запрос передается в сервис бронирования. Booking Service взаимодействует с Hotel Service для проверки доступности номера, после чего создаёт бронирование и возвращает пользователю подтверждение о его успешном создании.



### Сценарий: Добавление отзыва

После ввода текста отзыва в клиентском приложении запрос передается через API Gateway в сервис управления отелем. Hotel Service сохраняет отзыв в базе данных и возвращает подтверждение успешного выполнения операции пользователю через API Gateway.





## **Вывод**

В ходе работы была спроектирована архитектура системы бронирования отеля на основе микросервисного подхода. Система состоит из набора сервисов, объединённых через API Gateway, что обеспечивает единый вход для клиентских запросов и корректное клиент–серверное взаимодействие. Каждый сервис использует собственную базу данных, что позволяет изолировать данные и снизить связанность компонентов системы.

Диаграмма компонентов отражает внутреннюю структуру сервисов и их взаимодействие, демонстрируя разделение ответственности между обработкой запросов, бизнес-логикой и доступом к данным. Диаграммы баз данных показывают структуру хранимых данных и границы ответственности каждого сервиса. Диаграммы последовательности описывают основные пользовательские сценарии работы с системой и подтверждают корректность взаимодействия компонентов при выполнении ключевых операций.