

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэкенд-разработка

Отчет

Домашняя работа №1
Технический дизайн микросервисов

Выполнил:
Чернышев Михаил Павлович
Группа К3441

Проверил:
Добряков Д. И.

Санкт-Петербург
2026 г.

Содержание

1	Описание системы	2
2	Общая архитектура решения	2
2.1	Описание микросервисов	2
2.2	Диаграмма архитектуры	2
2.3	Технологический стек	3
3	Диаграмма компонентов	3
3.1	User Service	3
3.2	Advertisement Service	4
3.3	Messages Service	5
4	Диаграммы баз данных	5
4.1	User Service DB	6
4.2	Message Service DB	7
4.3	Advertisement Service DB	7
5	Диаграммы пользовательских сценариев	8
5.1	Сценарий 1: Создание бронирования	8
5.2	Сценарий 2: Поиск объектов	9
5.3	Сценарий 3: Отправка сообщения	9
5.4	Сценарий 4: Оставление отзыва	10
6	Вывод	11

1 Описание системы

Проектируемая система — **сервис аренды недвижимости**, позволяющий пользователям искать, бронировать жильё и общаться с владельцами.

Основные функции:

- Регистрация и авторизация пользователей
- Каталог объектов недвижимости с фильтрацией
- Создание объявлений владельцами
- Бронирование объектов (аренда)
- Обмен сообщениями между арендаторами и владельцами
- Отзывы и рейтинги

2 Общая архитектура решения

2.1 Описание микросервисов

Сервис	Назначение
API Gateway	Единая точка входа, маршрутизация запросов, rate limiting
User Service	Авторизация, JWT токены, профили пользователей
Advertisement Service	Объекты недвижимости, фото, удобства, бронирования, отзывы
Messages Service	Чат между пользователями

Таблица 1: Микросервисы системы

2.2 Диаграмма архитектуры



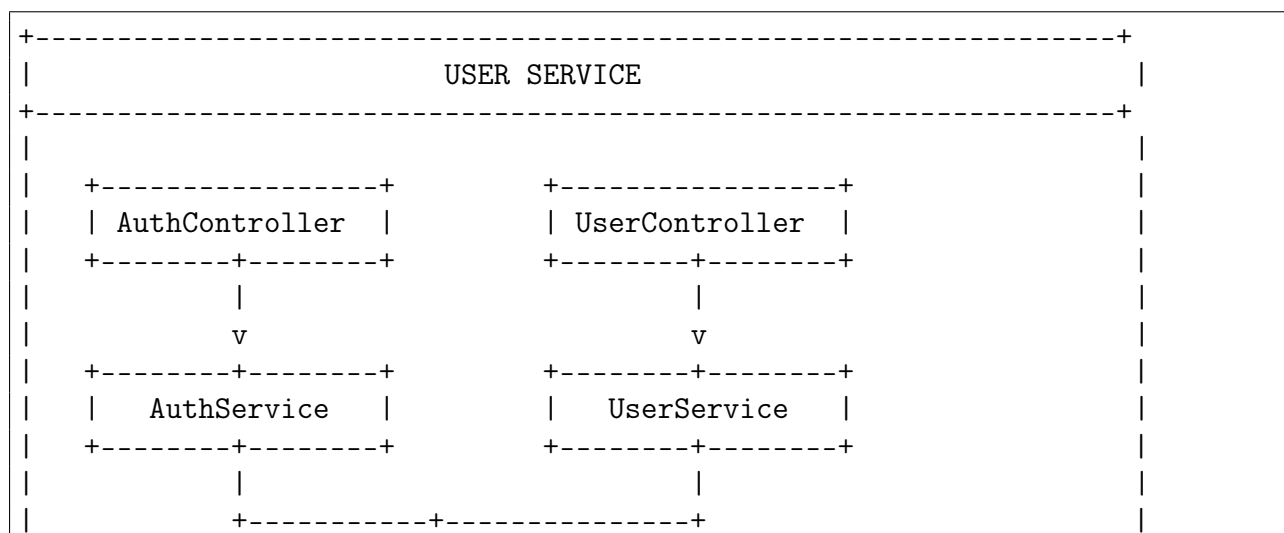
Рис. 1: Общая архитектура решения

2.3 Технологический стек

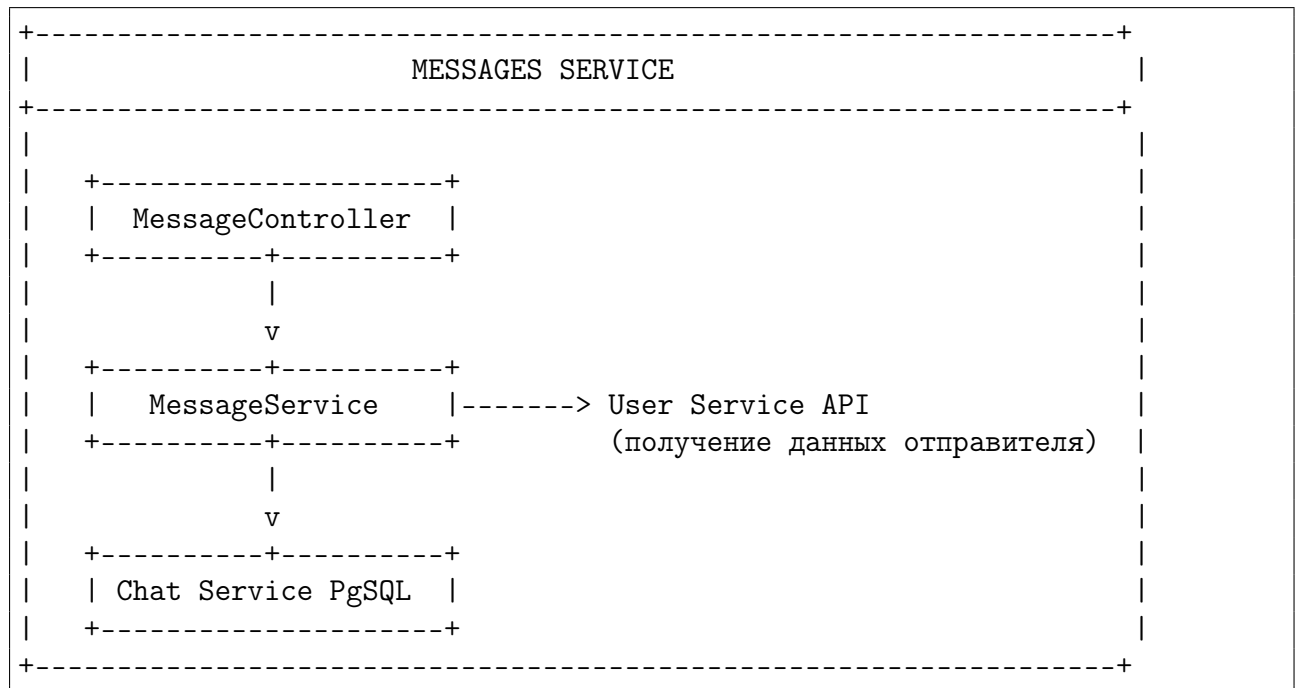
- **API Gateway:** Nginx / Kong
- **Сервисы:** Node.js (NestJS) / Go
- **База данных:** PostgreSQL
- **Кэш:** Redis
- **Контейнеризация:** Docker, Kubernetes

3 Диаграмма компонентов

3.1 User Service



3.3 Messages Service



4 Диаграммы баз данных

Диаграмма базы данных

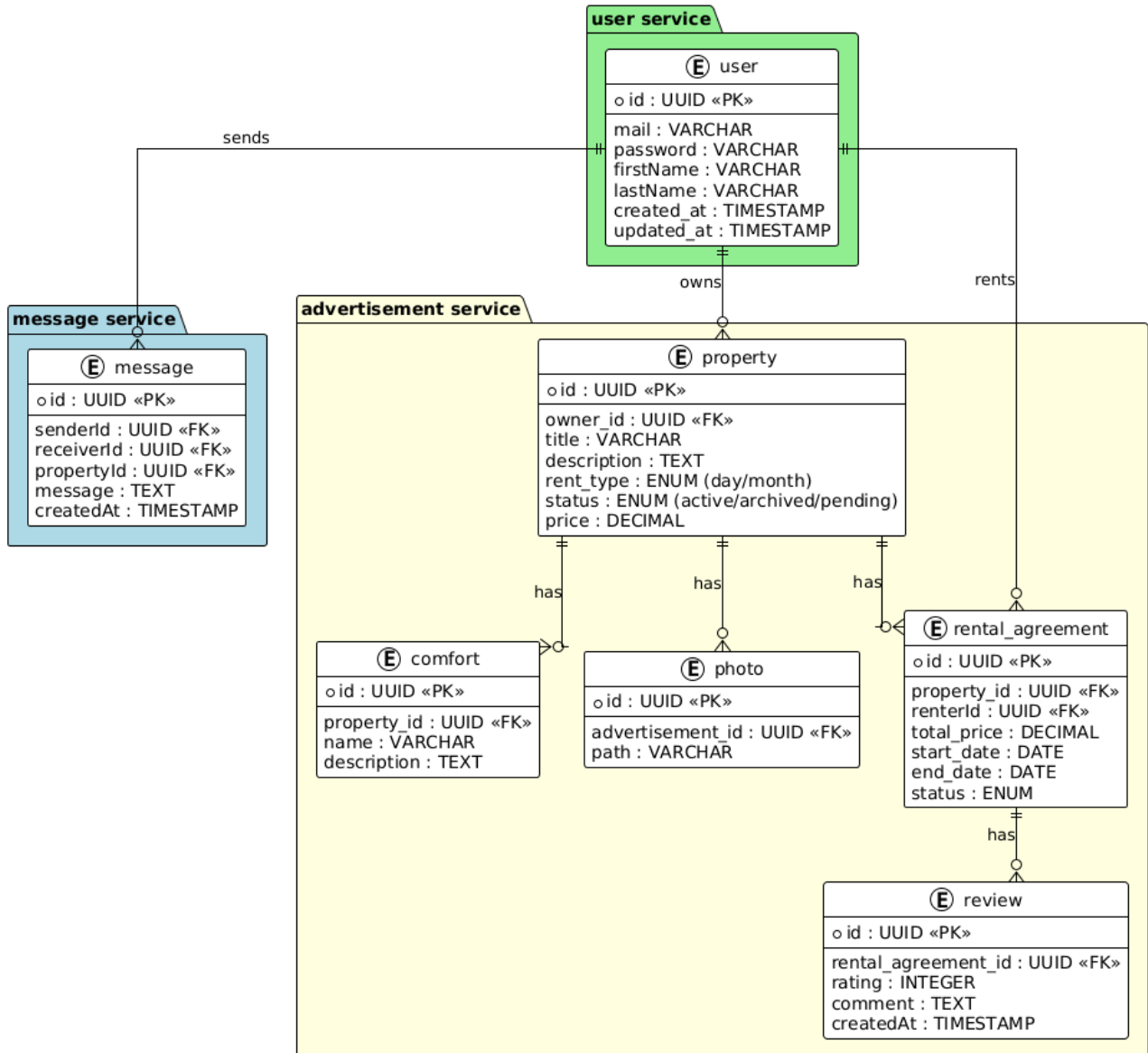


Рис. 2: ER-диаграмма базы данных

4.1 User Service DB

user	
PK	id
	mail
	password
	firstName
	lastName
	created_at
	updated_at

4.2 Message Service DB

+-----+			
	message		
+-----+			
	PK	id	
+-----+			
	FK	senderId	--> user.id
	FK	receiverId	--> user.id
	FK	propertyId	--> property.id
		message	
		createdAt	
+-----+			

4.3 Advertisement Service DB

+-----+			+-----+			+-----+				
property			rental_agreement			review				
+-----+			+-----+			+-----+				
PK	id		<--+	PK	id		<--+	PK	id	
+-----+				+-----+				+-----+		
FK	owner_id		+--	FK	property_id		+--	FK	rental_	
title				FK renterId				agreement_id		
description				total_price				rating		
rent_type				start_date				comment		
status				end_date				createdAt		
price				status				+-----+		
+-----+				+-----+						
+-----+				+-----+						
comfort				photo						
+-----+				+-----+						
PK	id			PK	id					
+-----+				+-----+						
FK	property_id			FK	advertisement_id					
name				path						
description				+-----+						
+-----+				+-----+						

5 Диаграммы пользовательских сценариев

5.1 Сценарий 1: Создание бронирования



Рис. 3: Сценарий создания бронирования

Описание сценария:

1. Пользователь отправляет запрос на создание бронирования
2. API Gateway проверяет авторизацию через User Service
3. Advertisement Service проверяет доступность дат
4. Создаётся запись о бронировании в БД
5. Messages Service отправляет уведомление владельцу
6. Пользователь получает подтверждение

5.2 Сценарий 2: Поиск объектов

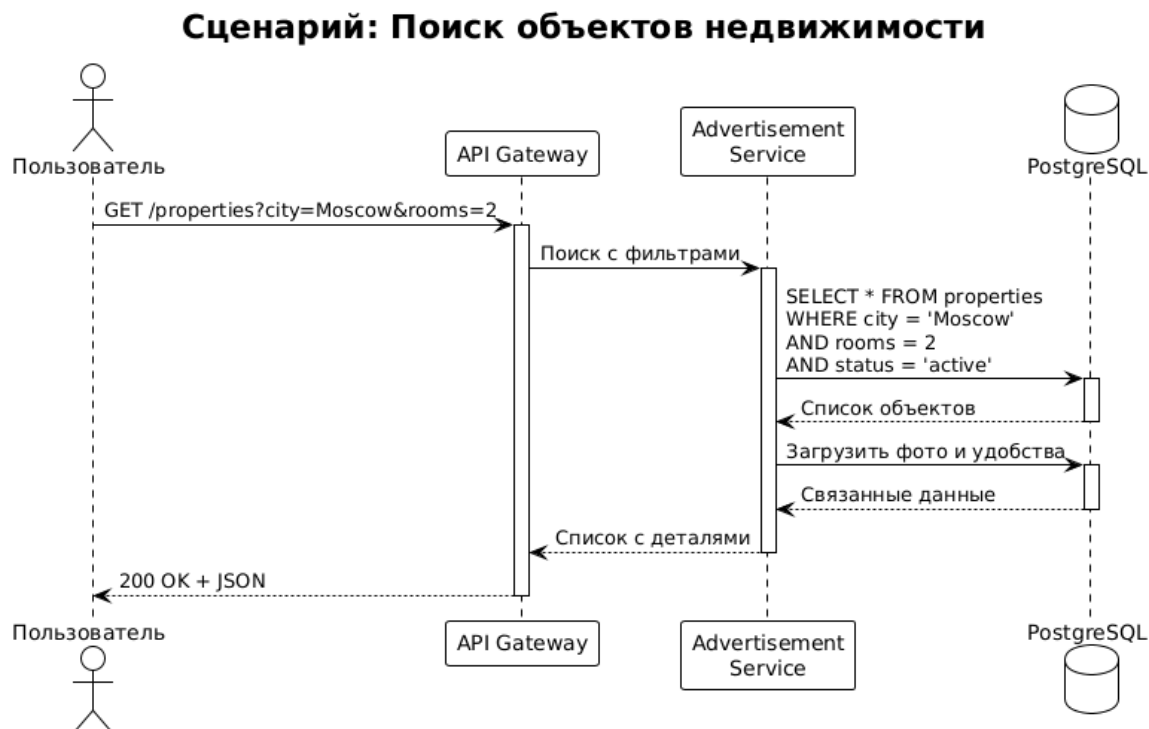


Рис. 4: Сценарий поиска объектов

Описание сценария:

1. Пользователь отправляет запрос с фильтрами (город, комнаты, цена)
2. API Gateway перенаправляет в Advertisement Service
3. Выполняется SQL-запрос с условиями фильтрации
4. Загружаются связанные данные (фото, удобства)
5. Пользователь получает список объектов

5.3 Сценарий 3: Отправка сообщения

Описание сценария:

1. Арендатор отправляет сообщение владельцу через Messages Service
2. Messages Service обращается к User Service для проверки получателя
3. Сообщение сохраняется в БД
4. Владелец получает уведомление (push/email)

5.4 Сценарий 4: Оставление отзыва

Описание сценария:

1. После завершения аренды пользователь может оставить отзыв
2. Advertisement Service проверяет, что аренда действительно была
3. Отзыв сохраняется и привязывается к rental_agreement
4. Обновляется средний рейтинг объекта

6 Вывод

В ходе выполнения домашней работы была спроектирована микросервисная архитектура для сервиса аренды недвижимости.

Разработанные диаграммы:

- **Общая архитектура** — 3 микросервиса (User, Advertisement, Messages) с API Gateway
- **Диаграммы компонентов** — внутренняя структура каждого сервиса (Controllers, Services, DB)
- **Диаграммы БД** — ER-диаграммы для каждого сервиса с таблицами и связями
- **Сценарии** — создание бронирования, поиск объектов, отправка сообщений, отзывы

Ключевые архитектурные решения:

- Разделение на независимые сервисы по бизнес-доменам
- Каждый сервис имеет свою базу данных (Database per Service)
- API Gateway для централизованной маршрутизации
- Межсервисное взаимодействие через REST API

Приложение: PlantUML код диаграмм

Диаграммы созданы с использованием PlantUML. Исходные файлы находятся в папке `diagrams/`:

- `architecture.puml` — архитектура
- `database.puml` — ER-диаграмма
- `sequence_order.puml` — сценарий бронирования
- `sequence_search.puml` — сценарий поиска

Для компиляции используйте <https://www.plantuml.com/plantuml/uml/> или расширение PlantUML для VS Code/IntelliJ.