

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа №2

Выполнил:

Максимова Ольга

Группа К3342

Проверил:

Добряков Д. И.

Санкт-Петербург

2026 г.

Задача

1. Опишите все функциональные и нефункциональные требования к вашему бэкенд-приложению
2. Выберите формат реализации API, которого вы будете придерживаться: REST API или Backend For Frontend
3. Спроектируйте все эндпоинты вашего API в формате OpenAPI-схемы с учётом реализованной диаграммы БД
4. Опишите тело каждого запроса и тело каждого ответа, продумайте возможные ошибки, возвращаемые из API

Ход работы

Аутентификация:

Auth		^
POST	/api/auth/login	 ✓
POST	/api/auth/logout	✓
POST	/api/auth/register	✓

User		^
GET	/api/users	✓
GET	/api/users/{id}	✓
PATCH	/api/users/{id}	✓
DELETE	/api/users/{id}	✓
GET	/api/users/{id}/conversations	✓
GET	/api/users/{id}/rents	✓

GET /api/users: возвращает всех пользователей

GET /api/users/{id}: возвращает публичные данные пользователя

PATCH /api/users/{id}: позволяет обновить данные пользователя

DELETE /api/users/{id}: позволяет удалить данные пользователя

GET /api/users/{id}/conversations: возвращает все активные чаты с сообщениями

GET /api/users/{id}/rents: возвращает две категории: где он арендатор и где арендодатель

Accommodation



GET	/api/accommodations	▼
POST	/api/accommodations	▼
GET	/api/accommodations/{id}	▼
PATCH	/api/accommodations/{id}	▼
DELETE	/api/accommodations/{id}	▼
POST	/api/accommodations/{id}/photos	▼
DELETE	/api/accommodations/{id}/photos/{photoId}	▼

GET /api/accommodations: возвращает список объектов недвижимости

POST /api/accommodations: создает новый объект недвижимости

GET /api/accommodations/{id}: возвращает полную информацию: объект, фото, удобства, условия

PATCH /api/accommodations/{id}: обновляет поля объявления

DELETE /api/accommodations/{id}: удаляет объект

POST /api/accommodations/{id}/photos: Добавляет фото к объявлению

DELETE /api/accommodations/{id}/photos/{photoId}: удаляет конкретное фото

Rents



GET	/api/rents	✓
POST	/api/rents	✓
GET	/api/rents/accommodation/{accomId}	✓
GET	/api/rents/{id}	✓
PATCH	/api/rents/{id}	✓

GET /api/rents: список всех аренд

POST /api/rents: добавление объекта

GET /api/rents/accommodation/{accomId}: все аренды конкретного объекта

GET /api/rents/{id}: информация о конкретной аренде с данными объекта и участников

PATCH /api/rents/{id}: обновление статуса

Messages



GET	/api/messages	✓
POST	/api/messages	✓
GET	/api/messages/conversation	✓
GET	/api/messages/{id}	✓
PATCH	/api/messages/{id}	✓
DELETE	/api/messages/{id}	✓

Payment



GET	/api/payments	✓
POST	/api/payments	✓
GET	/api/payments/rent/{rentId}	✓
GET	/api/payments/{id}	✓

Также прописаны вероятные ошибки. Например:

```
@error
model Error {
  code: int32;
  message: string;
  exception?: string;
  exc_type?: string;
}

model Unauthorized extends Error {
  code: 401;
  message: "Unauthorized: Invalid credentials or session expired";
}

model Forbidden extends Error {
  code: 403;
  message: "Forbidden: You do not have permission";
}
```

Вывод

В результате выполнения данной практической работы были спроектированы endpoints API в формате Rest API.