

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

**Отчет**

**Домашняя работа 1: Технический дизайн микросервисов**

**Выполнил:**  
**Фирсов Илья**

**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);

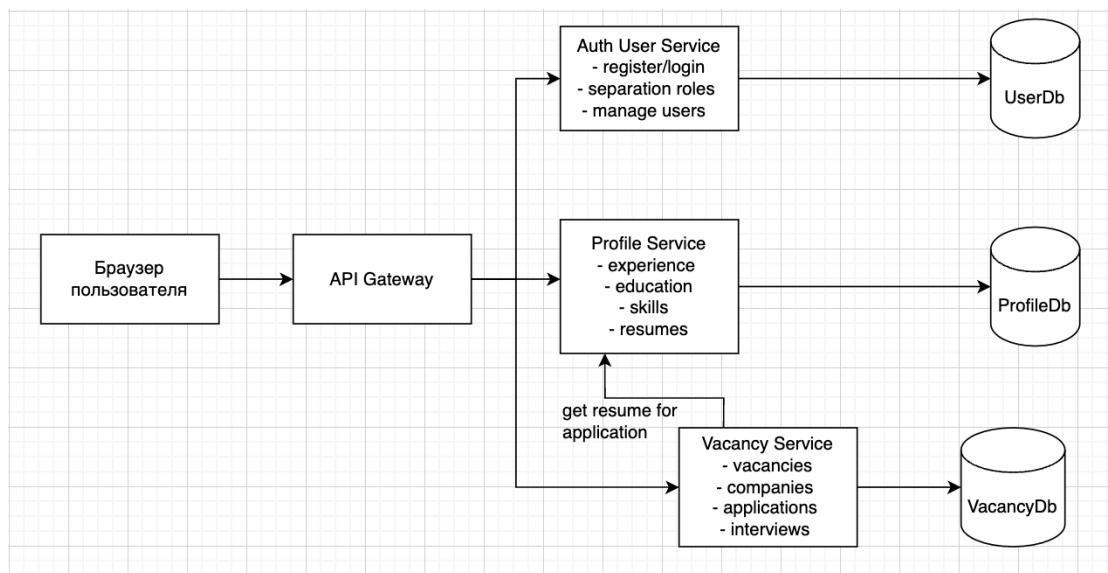
диаграмма компонентов;

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 реализована в виде трёх сервисов, объединённых через API Gateway. Пользователь работает через браузер, все запросы сначала попадают в Gateway, после чего маршрутизируются в нужный сервис.

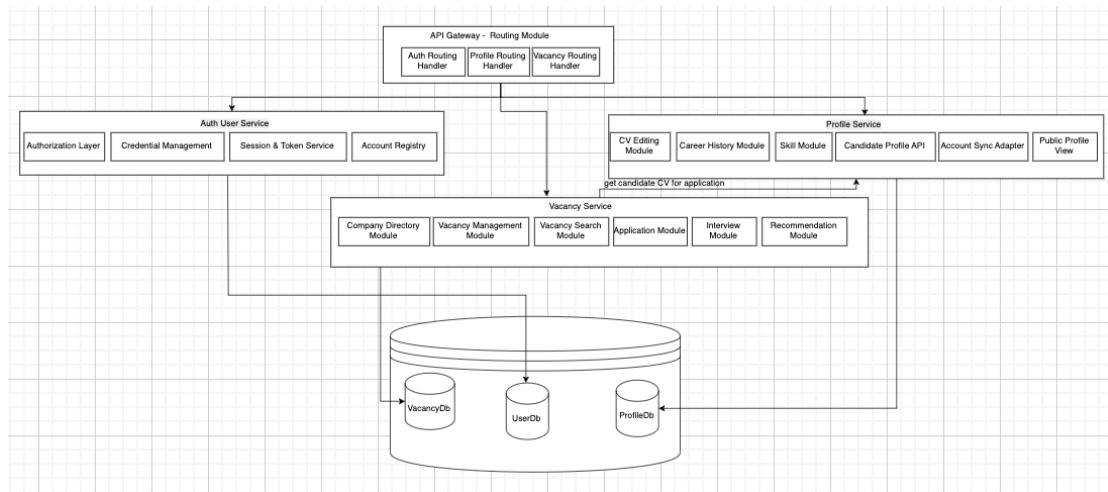
Auth User Service отвечает за регистрацию, вход, управление пользователями, роли, блокировки и выпуск токенов. Сервис также хранит связь пользователя с компанией, что позволяет сразу определять принадлежность пользователя к работодателю.

Profile Service отвечает только за данные резюме: навыки, опыт работы, образование и сводную информацию по кандидату. Profile Service используется как отдельный источник данных для вакансий.

Vacancy Service является самым нагруженным по логике. Он обрабатывает компании, вакансии, требования к навыкам, отклики, интервью, сотрудников и рекомендации. Все бизнес-процессы подбора сосредоточены в этом сервисе.

Каждый сервис работает только со своей базой данных. UserDb используется для авторизации, ProfileDb - для резюме, VacancyDb - для всего цикла найма. Межсервисное взаимодействие происходит только по HTTP.

## 2. Диаграмма компонентов



API Gateway содержит единый routing-модуль, который направляет запросы в Auth, Profile или Vacancy сервис. Gateway также используется как точка валидации токена перед передачей запроса дальше.

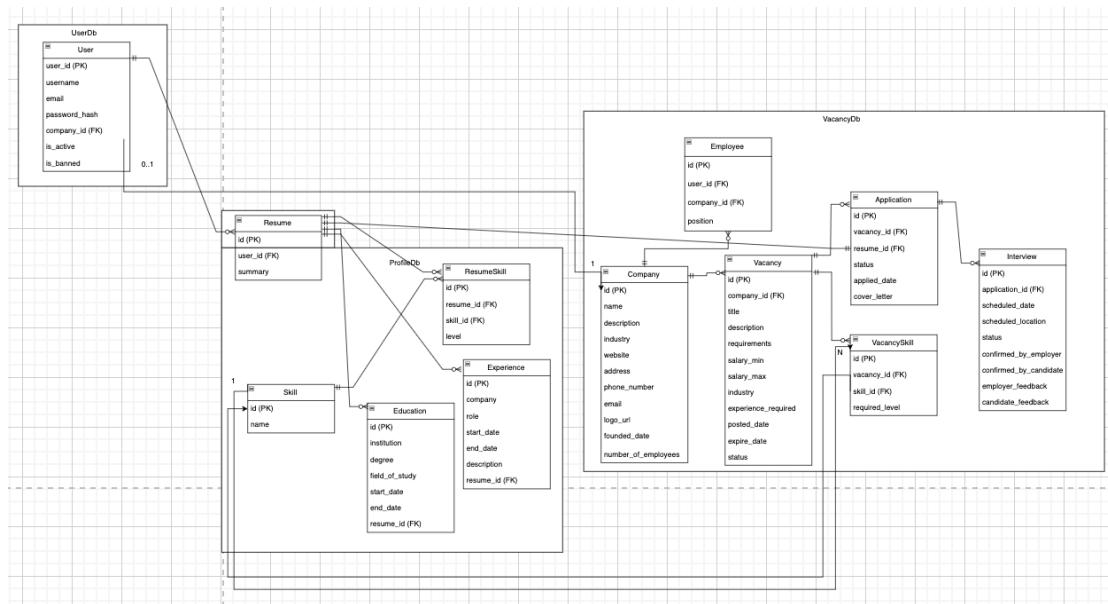
Auth User Service состоит из слоя авторизации, управления учётными данными, сервиса токенов и модуля регистрации аккаунтов. Он не участвует в бизнес-логике вакансий и резюме, а используется только для контроля доступа.

Profile Service включает модули редактирования резюме, навыков, истории карьеры и образования. Также в нём присутствует публичный просмотр профиля и адаптер синхронизации с аккаунтом пользователя.

Vacancy Service состоит из справочника компаний, управления вакансиями, поиска, модуля откликов, модуля интервью и модуля рекомендаций. здесь формируется весь процесс найма от публикации вакансии до интервью.

Vacancy Service запрашивает резюме кандидата из Profile Service при создании отклика. Auth User Service используется всеми сервисами для проверки прав пользователя.

### 3. Диаграммы баз данных (вариант 1, без пересказа таблиц)



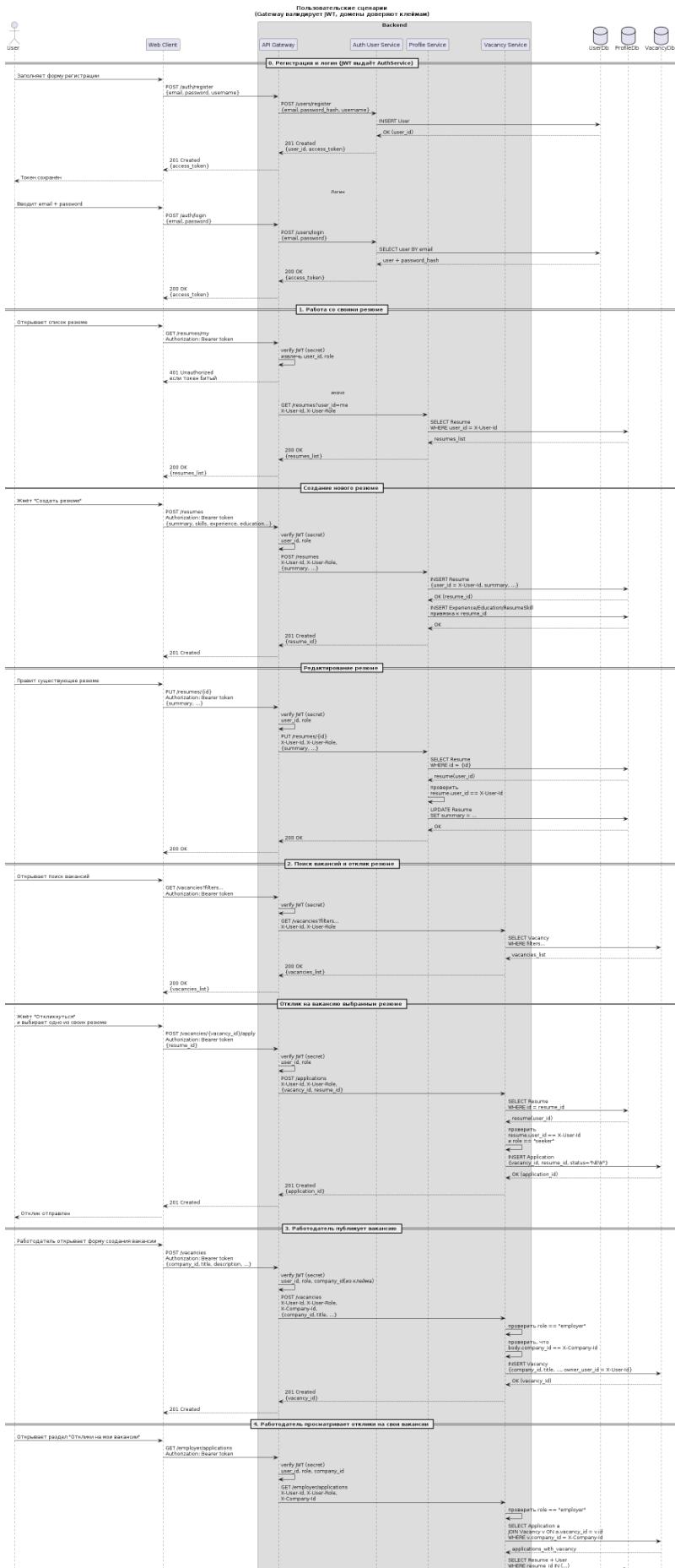
UserDb используется только для хранения аккаунтов, статусов пользователей, ролей, признаков блокировки и связи пользователя с компанией. Эта база не содержит бизнес-данных.

ProfileDb хранит только структуру резюме кандидатов. Все навыки, опыт работы и образование существуют только в рамках Profile Service. Эта база не знает ничего о вакансиях, компаниях и откликах.

VacancyDb хранит все объекты подбора: компании, вакансии, требования к навыкам, отклики, интервью и сотрудников. В ней же хранится связь между вакансиями и резюме через Application.

Разделение баз данных исключает прямое пересечение данных между сервисами. Все связки выполняются через идентификаторы и запросы между сервисами.

## **4. Пользовательские сценарии**



Пользователь начинает работу с регистрацией. Данные отправляются в Auth User Service через API Gateway. После создания аккаунта пользователь проходит вход и получает токен, который используется для всех дальнейших запросов.

После входа пользователь переходит к работе с резюме. Все операции с резюме выполняются через Profile Service. Создание, редактирование и хранение резюме происходят только в ProfileDb.

При поиске вакансий запросы отправляются в Vacancy Service. Сервис возвращает список вакансий из VacancyDb. После выбора вакансии пользователь отправляет отклик, при этом Vacancy Service получает резюме кандидата из Profile Service и создаёт запись Application.

Работодатель публикует вакансию через Vacancy Service. Перед созданием вакансии выполняется проверка, что пользователь привязан к компании. Вакансия сохраняется в VacancyDb.

После появления откликов работодатель просматривает список кандидатов. Vacancy Service получает данные резюме через Profile Service и отображает их вместе с откликом. Далее может быть создано интервью, которое также сохраняется в VacancyDb.

## **Вывод**

В этой системе основной упор сделан на сервис вакансий как на центральное ядро системы. Profile Service выступает как внешний поставщик резюме, а Auth User Service используется только для контроля доступа. Такая схема удобна для расширения логики подбора: интервью, рекомендации и работа с сотрудниками добавляются без изменения сервиса профилей. Архитектура жёстко разделяет данные пользователей, резюме и вакансий, что упрощает поддержку и масштабирование системы.