

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бек-энд разработка

Отчет

Домашняя работа 1: Технический дизайн микросервисов

Выполнил:

Петухов Семён

К3439

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

### I. Общая архитектура решения

**Тип:** Диаграмма архитектуры системы

**Описание:** Микросервисная архитектура для системы поиска работы.

**Состав сервисов:**

- **User-service** — управление пользователями, авторизация, роли (соискатель / работодатель).
- **Resume-service** — управление резюме, опытом, навыками, образованием.
- **Vacancy-service** — управление компаниями, вакансиями, требованиями.
- **Application-service** — подача откликов и хранение мотивационных писем.
- **Gateway/API Gateway** — точка входа для клиента, маршрутизация запросов.
- **Frontend (Client)** — пользовательский интерфейс (веб-клиент).
- **Database Layer** — отдельная БД на каждый сервис.

## Взаимодействие:

Клиент → API Gateway → соответствующий сервис → своя база данных.  
Сервисы общаются между собой по REST (например, Application-service запрашивает данные из Resume-service и Vacancy-service).

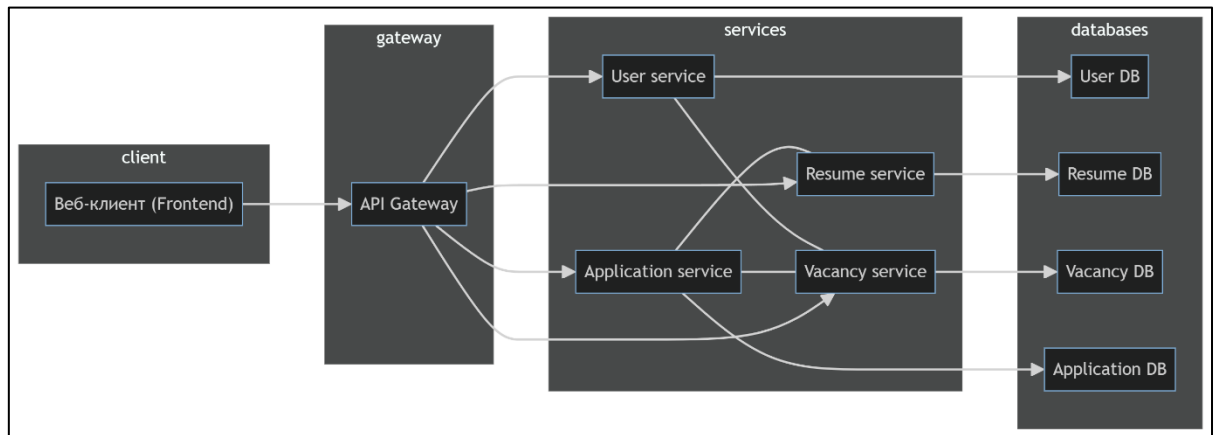


Рисунок 1 – Общая архитектура решения

## II. Диаграмма компонентов

### Основные компоненты внутри сервисов:

#### User-service

- Контроллеры: AuthController, UserController
- Репозиторий: UserRepository
- Модель: User
- Взаимодействует с Resume-service и Application-service

#### Resume-service

- Контроллеры: ResumeController, SkillController, EducationController
- Модели: Resume, WorkExperience, Skill, ResumeSkills, Education
- Репозитории для каждой сущности

#### Vacancy-service

- Контроллеры: VacancyController, CompanyController

- Модели: Vacancy, VacancySkills, Company
- Репозитории для каждой сущности

### Application-service

- Контроллеры: ApplicationController, MotivationLetterController
- Модели: Application, MotivationLetter
- Взаимодействует с Vacancy-service и Resume-service

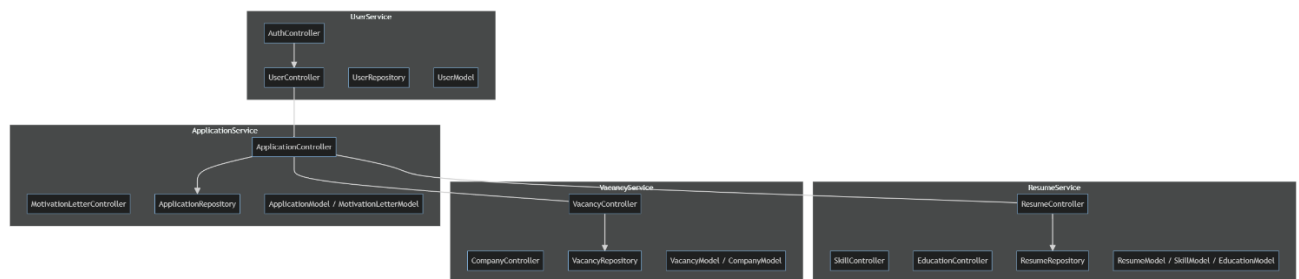


Рисунок 2 – Диаграмма компонентов

## III. Диаграммы баз данных (ERD)

### 1. User-service DB

Хранит информацию о пользователях и компаниях.

Пользователь (User) имеет логин, email, пароль и роль — соискатель или работодатель.

Если пользователь — работодатель, он связан с компанией (Company), у которой есть название, описание и местоположение.

Связь: один пользователь принадлежит одной компании, компания может иметь много пользователей.

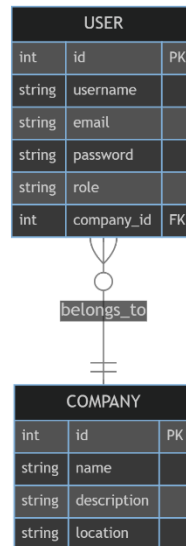


Рисунок 3 – User-service DB

## 2. Resume-service DB

Содержит данные о резюме и профессиональном опыте соискателей.

Резюме (Resume) связано с пользователем и включает имя, дату рождения, опыт, навыки, зарплату и образование.

Отдельно хранятся опыт работы (Work\_experience), навыки (Skill) и уровень образования (Education).

Для связи резюме и навыков используется промежуточная таблица Resume\_Skills.

Основные связи:

- одно резюме может иметь много записей об опыте,
- одно резюме может включать множество навыков (многие ко многим),
- одно образование может относиться к нескольким резюме.

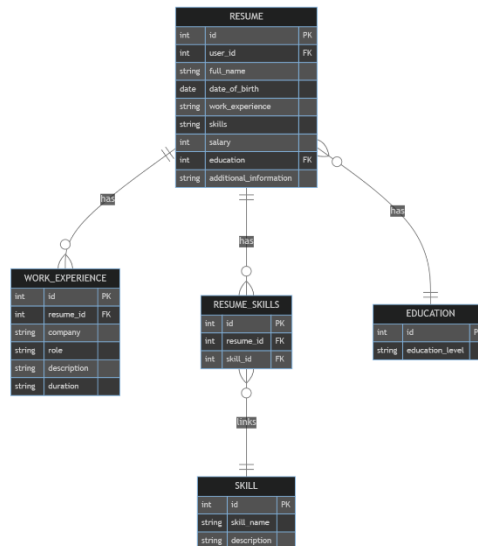


Рисунок 4 – Resume-service DB

### 3. Vacancy-service DB

Хранит вакансии и данные о компаниях.

Вакансия (Vacancy) содержит название, описание, требования, зарплату, опыт и принадлежит компании.

Через таблицу Vacancy\_Skills вакансия связывается с набором навыков, необходимых кандидату.

Компания может иметь несколько вакансий, а один навык может встречаться в разных вакансиях.

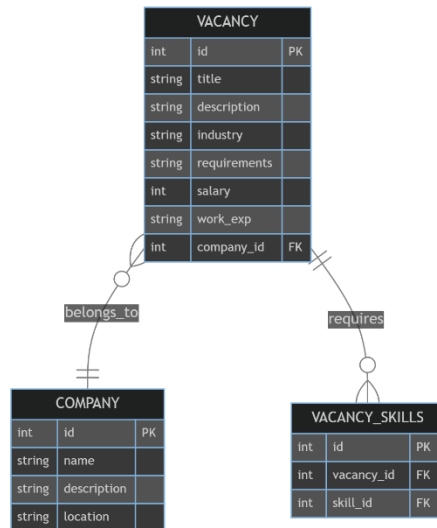


Рисунок 5 – Vacancy-service DB

#### 4. Application-service DB

Отвечает за отклики соискателей и мотивационные письма.

Отклик (Application) связывает пользователя, его резюме и вакансию, содержит статус и дату подачи.

Мотивационное письмо (Motivation\_letter) связано с пользователем и вакансией, содержит заголовок и текст.

Связи: один отклик может включать одно мотивационное письмо, а пользователь может отправить несколько откликов на разные вакансии.

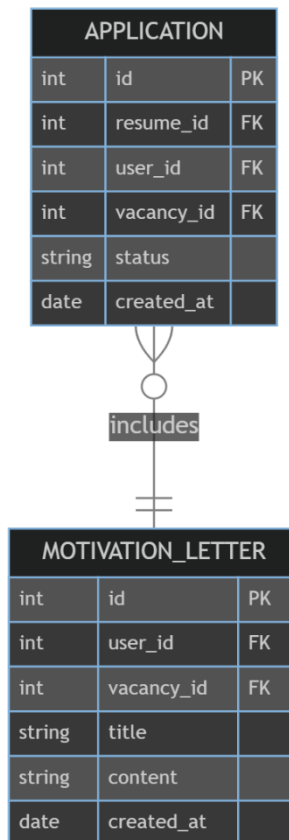


Рисунок 6 – Application-service DB

#### IV. Диаграммы пользовательских сценариев (Use Case)

##### Основные сценарии:

##### 1. Регистрация и вход пользователя

- Актор: Пользователь
- Действие: Регистрация → Вход → Получение токена доступа.

##### 2. Создание резюме

- Актор: Соискатель
- Действие: Создание резюме → Добавление навыков, опыта, образования.

##### 3. Создание вакансии



- Актор: Работодатель
- Действие: Создание компании → Публикация вакансии → Добавление требований.

#### 4. Поиск и отклик на вакансию

- Актор: Соискатель
- Действие: Поиск вакансий → Просмотр деталей → Отправка отклика + мотивационного письма.

#### 5. Просмотр откликов

- Актор: Работодатель
- Действие: Просмотр откликов на свои вакансии → Изменение статуса заявки.



Рисунок 6 – Use Case

## Вывод

В ходе работы была спроектирована структура базы данных для системы поиска работы, включающая пользователей, компании, резюме, вакансии и отклики.

Все сущности распределены по отдельным микросервисам, что обеспечивает гибкость, масштабируемость и простоту сопровождения системы.