

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»  
Кафедра №806 «Вычислительная математика и программирование»

**Итоговый проект  
по курсу «Базы данных»**

**Сервис по созданию и поиску резюме**

Выполнил: Н.В. Цирулев  
Группа: М8О-308Б-22  
Преподаватель: Малахов А.В.

Москва, 2024

## Оглавление

<b>Постановка задачи.....</b>	<b>3</b>
<b>Общее описание.....</b>	<b>4</b>
<b>Описание и схема моделей данных .....</b>	<b>5</b>
Схема моделей данных .....	5
Описание моделей данных .....	6
<b>Описание и схемы приложения .....</b>	<b>10</b>
Описание моделей данных .....	10
Use Case диаграмма для описания функционала сервиса .....	11
Дерево иерархии папок проекта .....	12
<b>QR код, ведущий на страницу проекта на github .....</b>	<b>13</b>

## **1. Постановка задачи**

Общие требования к итоговому проекту:

1. При реализации курсового проекта допускается только использование СУБД PostgreSQL.
2. Необходимо выбрать предметную область для создания базы данных. Выбранная предметная область должна быть уникальной для всего потока, а не только в рамках учебной группы.
3. Необходимо описать модели предметной области и уровня инфраструктуры и их назначение в рамках реализуемого проекта (минимальное количество моделей предметной области и уровня инфраструктуры - 5). Также необходимо выполнить проектирование логической структуры базы данных. Все таблицы, связанные с описанными моделями предметной области, должны находиться в 3NF или выше. База данных должна иметь минимум 7 таблиц.
4. Клиентское приложение должно быть в виде WEB или оконного приложения.
5. Необходимо организовать различные роли пользователей и права доступа к данным (например: администратор, редактор, рядовой пользователь). Клиентское приложение, взаимодействующее с базой данных, должно предоставлять функционал для авторизации пользователя по логину и паролю (хранение непосредственно пароля в базе данных запрещено, надо хранить HASH от этого пароля и проверять его).
6. При разработке функционала базы данных следует организовать логику обработки данных не на стороне клиента (Frontend), а на стороне

серверного приложения (Backend). Все обработки «SQL запросов», «работа бизнес-логики должны находиться в BACKEND части. FRONT только для отображения.

7. Запросы должны быть асинхронны. То есть, при нажатии на форму она не должна зависать. При нажатии на форму одним пользователем, другой должен иметь возможность свободно пользоваться приложением. То есть действия разных пользователей независимы.

## **2. Общее описание работы**

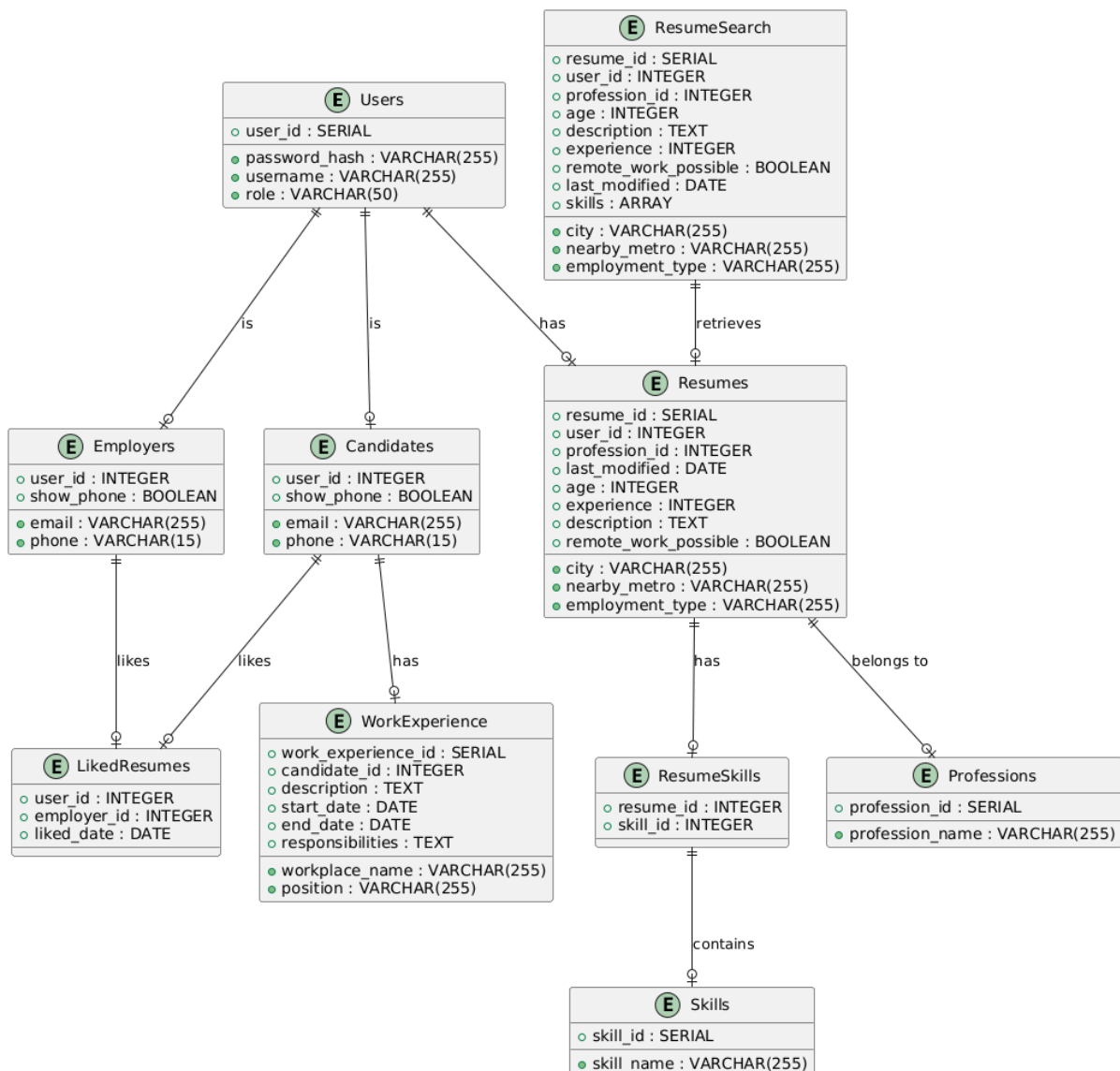
В данном проекте был реализован сервис создания и поиска резюме для соискателей и работодателей. Сервис представлен в виде веб-приложения на Streamlit.

Была реализована авторизация пользователей с разделением ролей на администраторов сайта, работодателей и кандидатов. Работодатель имеет возможность просматривать все резюме пользователей, сохранять наиболее релевантные и связываться с владельцем резюме любым удобным ему способом. Кандидаты имеют возможность выкладывать в общий доступ свое резюме и редактировать в любой удобный момент.

Система автоматически редактирует базу данных и с большим количеством пользователей обеспечивает более удобный поиск резюме для пользователей сайта благодаря системе тегов с подсказками, сортировке объявлений и фильтрации. Участие администраторов сайта требуется только лишь для добавления новых специальностей.

### 3. Описание и схема моделей данных

#### 3.1. Схема моделей данных



## 3.2. Описание моделей данных

### Таблица Users

Эта таблица предназначена для хранения информации о пользователях системы. Она включает уникальные данные о каждом пользователе, такие как хэш пароля, имя пользователя и его роль в системе (кандидат, работодатель или администратор). Роль пользователя критична для разграничения доступа в системе: например, кандидаты могут создавать резюме, а работодатели — просматривать их и взаимодействовать с кандидатами.

### Таблица Professions

Таблица профессий хранит перечень возможных профессий для кандидатов. Это необходимо для того, чтобы связывать резюме с конкретной профессией, а также для удобного поиска по профессиям в процессе взаимодействия с системой.

### Таблица Resumes

Таблица резюме хранит подробности о резюме кандидатов. Каждое резюме связано с пользователем (кандидатом) и конкретной профессией. Помимо базовой информации, такой как возраст, опыт работы, город, эта таблица содержит описание типа занятости и возможность удаленной работы, что важно для работодателей, ищущих подходящих кандидатов.

### Таблица Employers

Таблица работодателей хранит дополнительные контактные данные работодателей, такие как email и телефон, а также флаг, который указывает, следует ли показывать телефон в публичном доступе. Эти данные необходимы для связи между работодателями и кандидатами, особенно для более тесного взаимодействия.

## Таблица Candidates

Таблица кандидатов включает контактную информацию тех пользователей, которые хотят быть найдены работодателями. Подобно таблице работодателей, она хранит email и телефон кандидата, а также флаг отображения телефона, что дает возможность регулировать уровень конфиденциальности контактных данных.

## Таблица LikedResumes

Эта таблица фиксирует, какие резюме понравились работодателю. С помощью связи между работодателями и кандидатами она позволяет работодателям отслеживать интересующие их резюме, а кандидатам — видеть, какие работодатели заинтересовались их профилем.

## Таблица Skills

Таблица навыков необходима для того, чтобы описывать умения и профессиональные качества кандидатов. Навыки помогают работодателям искать кандидатов с нужными квалификациями и могут быть связаны с конкретными резюме через таблицу ResumeSkills.

## Таблица ResumeSkills

Таблица связи между резюме и навыками позволяет создавать точные и детализированные профили кандидатов, связывая их резюме с набором навыков. Это дает возможность работодателям искать кандидатов по конкретным навыкам, таким как знание определенных технологий или методов работы.

## Таблица WorkExperience

Таблица опыта работы хранит информацию о том, где и как долго работал кандидат. Эта таблица предоставляет подробности о предыдущих рабочих местах, что позволяет работодателям оценить опыт кандидата и его профессиональную подготовленность.

---

## 2. Триггер и Функция

### Функция update\_last\_modified

Эта функция автоматически обновляет дату последнего изменения резюме, когда оно обновляется. Это необходимо для отслеживания актуальности информации в системе, чтобы всегда можно было видеть, когда в последний раз вносились изменения в резюме.

```
CREATE OR REPLACE FUNCTION update_last_modified()
RETURNS TRIGGER AS $$
BEGIN
    NEW.last_modified = CURRENT_DATE;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER update_resume_date
BEFORE UPDATE ON Resumes
FOR EACH ROW
EXECUTE FUNCTION update_last_modified();
```

### Триггер update\_resume\_date

Триггер автоматически срабатывает перед обновлением резюме и вызывает функцию, которая обновляет поле last\_modified. Это гарантирует, что дата последнего изменения будет актуальной и точной, без необходимости вручную обновлять это поле каждый раз.

---



### 3. Представление

#### Представление ResumeSearch

```
CREATE OR REPLACE VIEW ResumeSearch AS
SELECT r.resume_id, r.user_id, r.profession_id, r.age, r.description, r.experience, r.city, r.nearby_metro,
       r.employment_type, r.remote_work_possible, r.last_modified, p.profession_name,
       array_agg(s.skill_name) FILTER (WHERE s.skill_name IS NOT NULL) AS skills
FROM Resumes r
JOIN Professions p ON r.profession_id = p.profession_id
LEFT JOIN ResumeSkills rs ON r.resume_id = rs.resume_id
LEFT JOIN Skills s ON rs.skill_id = s.skill_id
GROUP BY r.resume_id, p.profession_name;
```

Это представление предоставляет агрегированную информацию по резюме кандидатов, включая не только основные данные (возраст, опыт работы, описание), но и связанные с ними навыки. Оно объединяет несколько таблиц, чтобы создать удобный и функциональный интерфейс для поиска кандидатов, что значительно упрощает работу как для работодателей, так и для системы в целом. Например, представление помогает находить резюме с определенным набором навыков или с указанием типа занятости, что делает поиск более точным и эффективным.

---

Эти таблицы и механизмы работают вместе, чтобы создать гибкую систему для управления резюме, поиска кандидатов и взаимодействия между кандидатами и работодателями. База данных предлагает как хранение данных, так и функциональные возможности для их обработки и анализа, обеспечивая удобный и эффективный процесс для всех пользователей.

## **4. Описание и схемы приложения**

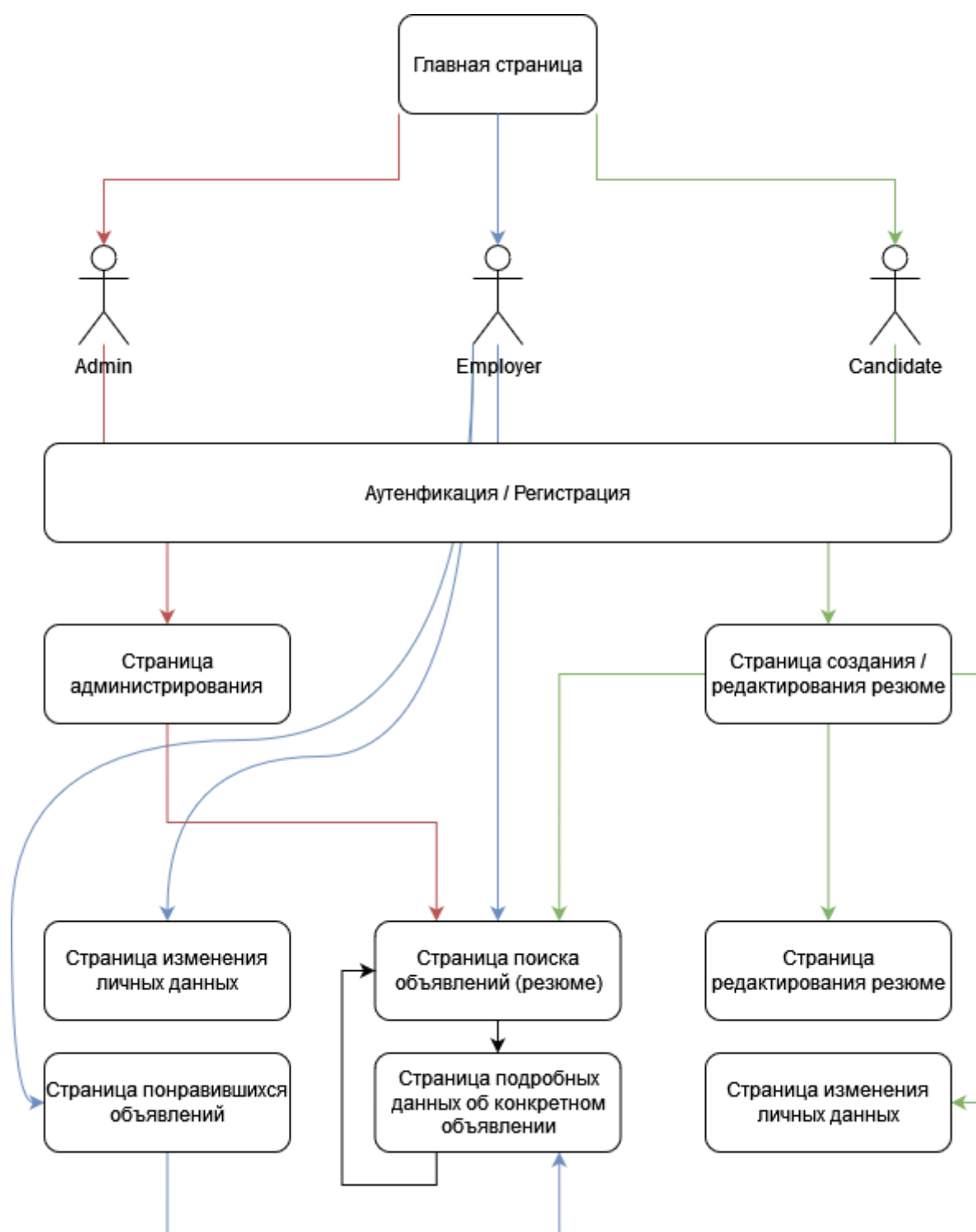
### **4.1. Краткое описание приложения**

Пользователь заходит на главную страницу приложения. Ему предлагается выбрать роль, которая сохраняется в сессии. Далее пользователь с ролью администратор сразу попадает на страницу входа, рядовой пользователь вводит адрес электронной почты или номер телефона, если пользователь с такими данными существует, то его перекидывает на страницу входа, иначе предлагается зарегистрироваться.

После аутентификации пользователь попадает на сайт. Пользователь с ролью “Employer” имеет право добавлять понравившиеся резюме в избранное, но не имеет права создавать свое резюме. Пользователь с ролью “Candidate” имеет право создавать свое резюме и редактировать, но не имеет функционала добавлять резюме в избранное.

Просматривать страницу с объявлениями (резюме) и переходить по ним, чтобы увидеть более подробную информацию, может как зарегистрированный пользователь с любой ролью, так и любой незарегистрированный пользователь.

## 4.2. Use Case диаграмма для описания функционала сервиса



### 4.3. Дерево иерархии папок проекта



## 5. QR код, ведущий на страницу проекта на github

