# Лабораторная работа №1

Тема: Программная реализация сервиса для автоматизации отбора кандидатов на основе обработки естественного языка из резюме и вакансий

1. Техническое задание по ГОСТ 19.201-78

1. Общие сведения

1.1. Наименование системы: Сервис автоматизации отбора кандидатов «RecruitAI».

1.2. Шифр темы: М24-534.

1.3. Заказчик: Национальный исследовательский ядерный университет «МИФИ».

1.4. Разработчик: Студент группы М24-534 Носов Артём Иванович.

1.5. Сроки разработки: 18.09.2024 – 22.12.2024.

1.6. Источники финансирования: Бюджет НИР университета.

2. Назначение и цели создания системы

2.1. Назначение: Автоматизация процесса подбора кандидатов через анализ резюме и вакансий с использованием NLP.

2.2. Цели:

- Сокращение времени первичного отбора на 50%.

- Повышение точности сопоставления кандидатов и вакансий до 85%.

- Интеграция с HR-платформами (HH.ru, LinkedIn).

3. Характеристика объекта автоматизации

3.1. Объект: Процесс рекрутмента, включающий обработку резюме (текст), анализ требований вакансий, ранжирование кандидатов.

3.2. Условия эксплуатации:

- Онлайн-доступ 24/7.

- Поддержка до 1000 одновременных запросов.

- Совместимость с ОС Windows/Linux, СУБД PostgreSQL.

4. Требования к системе

4.1. Требования к системе в целом:

- Микросервисная архитектура (REST API).

- Режимы работы: автоматический анализ, ручная корректировка результатов.

- Надежность: время восстановления после сбоя ≤ 5 мин.

- Безопасность: шифрование данных (AES-256), аутентификация через OAuth 2.0.

- Эргономика: веб-интерфейс с поддержкой мобильных устройств.

4.2. Требования к функциям:

- Загрузка резюме (PDF, DOCX).

- Семантический анализ текста (BERT).

- Ранжирование кандидатов по соответствию вакансии.

- Генерация отчетов (PDF, Excel).

- Про параллельность

4.3. Требования к видам обеспечения:

- Программное: Python 3.10, PyTorch, FastAPI.

- Техническое: 8 ГБ ОЗУ, 4 ядра CPU, GPU (для NLP).

- Информационное: база резюме и вакансий в PostgreSQL, векторное хранение в Weaviate.

5. Состав и содержание работ

5.1. Этапы разработки:

- Анализ требований (08.09.2024 – 29.09.2024).

- Проектирование архитектуры (06.10.2024 – 27.10.2024).

- Реализация и тестирование (01.12.2024 – 22.12.2024).

5.2. Документы: Технический проект, руководство пользователя, тестовые сценарии.

6. Порядок контроля и приемки

6.1. Виды испытаний:

- Функциональное тестирование (модульное, интеграционное).

- Нагрузочное тестирование (до 10 000 запросов/час).

6.2. Приемка: утверждение заказчиком после успешного тестирования.

2. Концептуальная модель

2.1. Таблица прецедентов

| Прецедент | Акторы | Описание |

|-------------------------------|----------------------|-------------------------------------------------------------------------|

| Загрузка резюме | Рекрутер, Кандидат | Загрузка резюме в формате PDF/DOCX в систему. |

| Анализ текста | Система | Извлечение навыков, опыта и образования с использованием BERT. |

| Сопоставление с вакансиями | Система | Расчет уровня соответствия кандидата требованиям вакансии. |

| Ранжирование кандидатов | Рекрутер | Просмотр списка кандидатов, отсортированных по релевантности. |

| Генерация отчета | Рекрутер | Экспорт результатов в PDF/Excel. |

2.2. Диаграмма прецедентов

```plaintext

[Кандидат] --> (Загрузка резюме)

[Рекрутер] --> (Загрузка вакансии)

[Система] --> (Анализ текста)

[Система] --> (Сопоставление с вакансиями)

[Рекрутер] --> (Ранжирование кандидатов)

[Рекрутер] --> (Генерация отчета)

```

2.3. Диаграмма концептуальных классов

```plaintext

+----------------+ +----------------+ +----------------+

| Резюме | | Вакансия | | Кандидат |

|----------------| |----------------| |----------------|

| - текст | | - требования | | - имя |

| - навыки |<----->| - навыки |<----->| - опыт |

| - опыт | | - зарплата | | - контакты |

+----------------+ +----------------+ +----------------+

```

2.4. Системное взаимодействие (клиент-сервер)

```plaintext

[Клиент] --> (REST API) --> [Сервер] --> [NLP-обработчик] --> [СУБД]

| |

v v

[Ранжирование] [Векторная БД]

```

3. Дополнения для повышения оценки

- Сетевые технологии: Использование Kubernetes для масштабирования микросервисов.

- Принятие решений: Алгоритм ранжирования с учетом мягких навыков (лидерство, коммуникабельность) через анализ тональности текста.