МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

Мобильное приложение Vkatun для анализа и улучшения резюме с помощью ИИ

Курсовая работа

Направление: 09.03.04. Программная инженерия

Зав. Кафедрой _____ д. ф.-м. н, доцент С.Д. Махортов

Руководитель _____ ст. преподаватель В.С. Тарасов

Руководитель практики _____ Е.Д. Проскуряков

Обучающийся _____ А.С. Бондарев, 3 курс, д/о

Обучающийся _____ А.А. Аликин, 3 курс, д/о

Обучающийся _____ А.А. Васильев, 3 курс, д/о

Обучающийся _____ С.В. Кривоносова, 3 курс, д/о

Обучающийся _____ П.И. Мигачев, 3 курс, д/о

Обучающийся _____ Д.Э. Крумов, 3 курс, д/о

Воронеж 2025

СОДЕРЖАНИЕ

Определения, обозначения и сокращения

Термин	Определение
Android	операционная система с открытым исходным кодом, созданная для мобильных устройств на основе модифицированного ядра Linux
API	набор способов и правил, по которым различные программы общаются между собой и обмениваются данными
Buildx	расширение Docker CLI, предоставляющее улучшенные возможности для сборки мультиплатформенных Docker-образов (multi-arch) с поддержкой сложных сценариев билда
Dart	современный, объектно-ориентированный язык программирования, разработанный компанией Google. создавался как альтернатива JavaScript для веб-разработки
DeepSeek	нейросетевая технология или платформа, разработанная компанией DeepSeek, которая специализируется на создании решений на основе искусственного интеллекта (ИИ) и машинного обучения
Docker	платформа для разработки, доставки и запуска приложений в изолированных средах
Docker-образ	шаблон для создания контейнеров, содержащий файловую систему, зависимости, код и настройки приложения
embed.FS	встроенный тип в языке **Go**, позволяющий включать статические файлы (HTML, CSS, конфиги и др.) прямо в исполняемый бинарный файл этапе компиляции
Flutter SDK	набор инструментов для разработки кроссплатформенных приложений, позволяет разрабатывать приложения для мобильных платформ, веба, настольных операционных систем
GitHub	платформа для хостинга проектов на базе Git, которая обеспечивает возможность хранения кода, управления задачами, рецензирования кода и совместной работы над проектами

Gorilla Mux	HTTP-роутер и мультиплексор запросов для языка Go, предоставляющий богатый набор инструментов для обработки URL и маршрутизации. Входит в состав популярного набора инструментов Gorilla Web Toolkit
HTTPS	защищенная версия протокола HTTP, использующая шифрование для безопасной передачи данных
JSON	текстовый формат обмена данными, основанный на синтаксисе объектов JavaScript. JSON используется для представления структурированных данных в виде пар «ключ-значение» и массивов
LLM	LLM (Large Language Model, крупная языковая модель) — это тип искусственного интеллекта, основанный на нейронных сетях, который обучен на огромных объемах текстовых данных для понимания, генерации и обработки человеческого языка
MockDB	MockDB (от англ. Mock Database — «фиктивная база данных») — это тестовая или поддельная база данных, используемая в процессе разработки и тестирования программного обеспечения для имитации поведения реальной базы данных без необходимости работать с настоящими данными или полноценной БД
PGX	высокопроизводительный драйвер для взаимодействия с базой данных PostgreSQL, написанный на языке Rust. Используется для создания клиентов (например, API или сервисов) и/или расширений PostgreSQL
PostgreSQL	открытая и объектно-реляционная система управления базами данных (СУБД), которая поддерживает расширенные функции, такие как сложные запросы, транзакции, триггеры, хранимые процедуры и многое другое
REST API	архитектурный стиль для создания веб-сервисов, который использует стандартные HTTP-методы для взаимодействия с ресурсами. REST API основан на принципах REST

SSH	SSH (Secure Shell) — это криптографический сетевой протокол, который используется для безопасного удалённого доступа к компьютерам и серверам через небезопасную сеть, например Интернет
Zap	быстрый и структурированный логгер для языка программирования Go (Golang)
ИИ	область компьютерных наук, занимающаяся созданием систем, способных выполнять задачи, требующие человеческого интеллекта
Интерфейс приложения	визуальная и функциональная часть приложения, с которой взаимодействует пользователь
Метрики работы приложения	количественные показатели, которые используются для измерения и оценки производительности, стабильности, удобства использования и других аспектов работы приложения
Парсинг	процесс анализа и обработки структурированных данных (например, текста, HTML, XML, JSON) с целью извлечения полезной информации или преобразования данных в удобный для использования формат
Серверная часть	скрытая от пользователя составляющая веб-приложений или программного обеспечения, отвечающая за обработку данных, бизнес-логику, взаимодействие с базами данных и обеспечение работы клиентской части (Frontend)
СУБД	программное обеспечение, предназначенное для создания, хранения, управления и взаимодействия с базами данных (Система Управления Базами Данных)

Введение

На современном рынке труда конкуренция среди соискателей постоянно растёт, что требует от кандидатов всё более качественно составленных резюме. Однако далеко не каждый человек обладает навыками, позволяющими грамотно структурировать и преподнести информацию о своём опыте и компетенциях. Ошибки в оформлении, отсутствие ключевых деталей или плохая подача могут существенно снизить шансы на трудоустройство, даже при наличии хорошего опыта.

С развитием технологий и широким внедрением искусственного интеллекта появилась возможность автоматизировать и упростить процесс составления резюме.

Проект представляет собой мобильное приложение Vkatun, предназначенное для анализа и улучшения резюме с помощью ИИ. Приложение позволяет пользователю загружать свое резюме и получать персонализированные рекомендации по его улучшению.

В данной работе описывается реализация функциональных возможностей мобильного приложения Vkatun, включая выбор архитектурных решений, логику взаимодействия компонентов системы, а также средства реализации и особенности разработки серверной и клиентской частей.

1 Постановка задачи

1.1 Цели создания системы

Целями создания системы являются:

— Удовлетворенность пользователей качеством рекомендаций ИИ и процессом улучшения резюме после месяца использования приложения должна составлять не менее 7 из 10 баллов (где 1 – полностью не удовлетворен, 10 – полностью удовлетворен), что повысит вероятность их возвращения в приложение и увеличит лояльность к бренду;

— В период с августа по ноябрь 2025 года обеспечить привлечение первых 200 активных пользователей, что позволит начать формирование клиентской базы и создаст основу для будущего роста;

— В течение первых 3 месяцев после запуска приложения достичь уровня конверсии пользователей в платящих клиентов не менее 5%, что обеспечит регулярный доход и позволит протестировать модель монетизации.

1.2 Функциональные требования к разрабатываемой системе

Разрабатываемая система должна обеспечивать реализацию следующих функциональных возможностей:

— Загрузка существующего резюме для его последующего анализа и редактирования;

— Выдача рекомендаций по улучшению резюме на основе анализа содержания документа;

— Возможность ручного редактирования компонентов резюме в интерфейсе приложения;

— Функция экспорта готового резюме;

— Отображение метрик работы системы в интерфейсе администратора.

2 Анализ предметной области

2.1 Анализ рынка сервисов по улучшению резюме

В ходе исследования рынка сервисов по улучшению резюме с помощью ИИ анализа было выявлено 3 прямых конкурента. Рисунок 1 содержит результаты проведённого конкурентного исследования.

	VisualCV https://app.visualcv.com/	Teal https://app.tealhq.com/home	Jobscan https://www.jobscan.co/
Дизайн и пользовательский опыт	Дизайн средний, интерфейс не перегружен, приятно пользоваться	Дизайн хороший, интерфейс немного перегружен	Дизайн очень приятный, нет лишних элементов, интерфейс не перегружен
Создание резюме с нуля с помощью ИИ	Конструктор резюме позволяет редактировать каждый элемент, но написание резюме через ИИ нет	Конструктор резюме позволяет редактировать каждый элемент, но написание резюме через ИИ нет	Есть создание резюме с нуля с помощью генерации ИИ. Генерация хорошая
Улучшение существующего резюме через ИИ	Нет	Даются только общие оценки и общие рекомендации	Есть, реализовано очень хорошо. Оценка каждого компонента и выдача рекомендаций
Адаптация для русского рынка	Шаблоны и упор на международный рынок. Есть поддержка русского языка	Шаблоны и упор на международный рынок	Шаблоны и упор на международный рынок
Подбор подходящих вакансий	Нет	Есть, но очень непонятный	Есть, но не заработал
Уровень ИИ рекомендаций	Носят общий характер	Носят общий характер	Хороший
Стоимость	Бесплатная версия слишком ограничена. Платная подписка дает больше полезных ИИ возможностей, шаблонов, загрузку своего резюме. 25$ в месяц, 48$ в квартал	Бесплатная версия не слишком ограничена. Платная подписка дает больше полезных ИИ возможностей. 9$ в неделю, 29$ в месяц	Бесплатная версия в меру ограничена. Платная подписка дает больше полезных фич сервиса, рекомендаций ИИ, меньше ограничений и т.д. 50$ в месяц, 90$ в квартал, есть 2 недели пробного периода
База знаний	Есть полезные материалы, шаблоны, но их мало	Есть только интерактивная подготовка, полезных материалов нет	Очень много полезных материалов на разные темы
Помощь ИИ с сопроводительным письмом	Есть только шаблоны, писать резюме нужно самому	Есть, хорошо реализовано	Есть, хорошо реализовано
Интеграция с другими сервисами	Linkedin	Нет	Linkedin, работа с ATS

Рисунок 1 – Бенчмаркинг основных конкурентов

VisualCV — это онлайн-платформа для создания и редактирования профессиональных резюме, а также сопроводительных писем и портфолио (Рисунок 2). Пользователи могут выбрать один из множества доступных шаблонов для оформления документа, а также импортировать данные из LinkedIn для автоматического создания резюме. Система позволяет обновлять и редактировать резюме в реальном времени, а также отслеживать статистику по просмотрам резюме, что помогает оценить эффективность документа в поисках работы [1].

Платформа поддерживает экспорт резюме в различные форматы (PDF, Word и другие), а также позволяет настроить доступ к резюме с помощью

ссылок. В разделе сопроводительных писем пользователи могут создавать персонализированные письма для отправки работодателям.

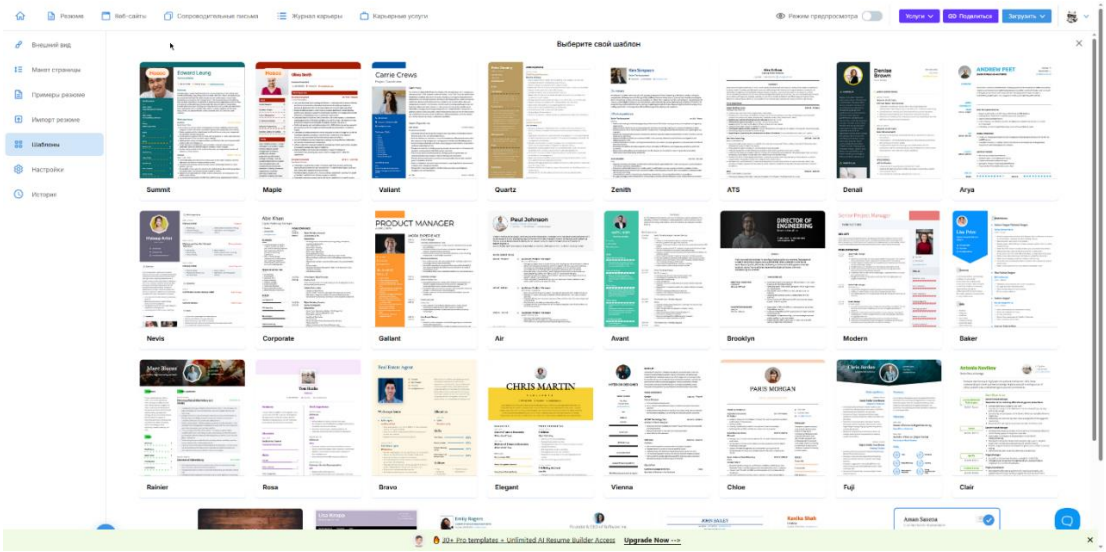

Рисунок 2 – Шаблоны резюме в VisualCV, доступные для пользователей

Teal — это сервис для создания и оптимизации резюме с использованием аналитики и рекомендаций по ключевым словам (Рисунок 3). Основная функция платформы — это анализ резюме с точки зрения автоматических систем отслеживания (ATS), которые применяются большинством работодателей для фильтрации резюме [2]. Teal помогает пользователям повысить шансы на прохождение через эти системы, предоставляя рекомендации по улучшению структуры документа и по включению нужных ключевых слов, соответствующих требованиям вакансий.

Платформа также включает в себя инструменты для отслеживания вакансий, планирования карьерных целей и создания индивидуальных стратегий для поиска работы. Одной из полезных функций является возможность получения анализа резюме по результатам его сравнения с описаниями вакансий, что помогает пользователю улучшить резюме для конкретных предложений работы.

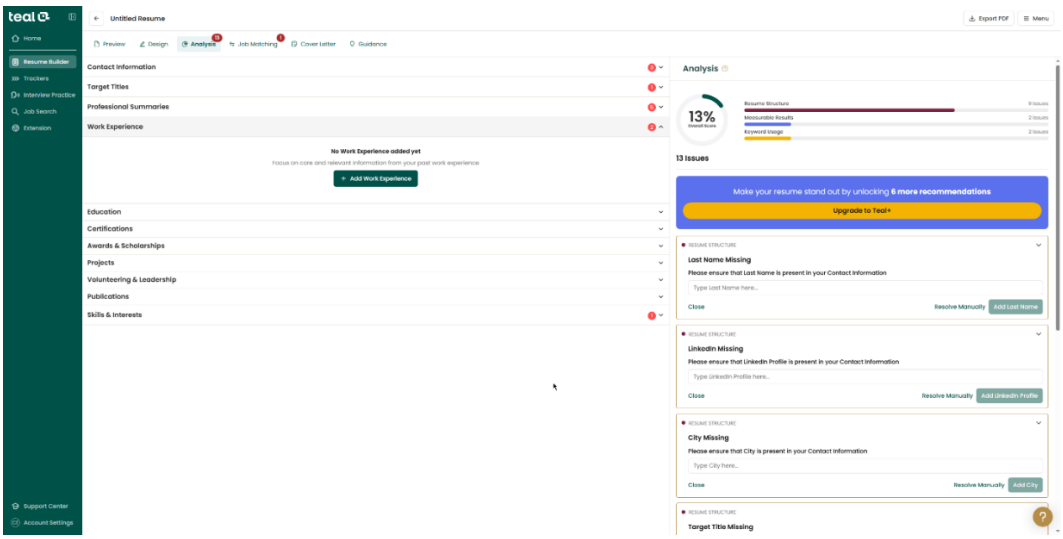

Рисунок 3 – Оценка резюме и рекомендации по улучшению Teal

Jobscan — это инструмент (Рисунок 4), ориентированный на улучшение резюме для соответствия требованиям систем автоматического отбора резюме (ATS). Платформа позволяет пользователям загружать свои резюме и вакансии для проведения анализа на совпадение ключевых слов и фраз, которые используются работодателями [3]. С помощью Jobscan можно оптимизировать резюме, подбирая наиболее подходящие ключевые слова и улучшая структуру документа в соответствии с требованиями вакансий.

Jobscan анализирует текст резюме и предоставляет подробные рекомендации, основанные на таких факторах, как форматирование, выбор ключевых слов и общая релевантность для вакансий. Сервис также включает инструмент для сравнения резюме с описанием вакансии, что помогает улучшить резюме и повысить шансы на отклик работодателя.

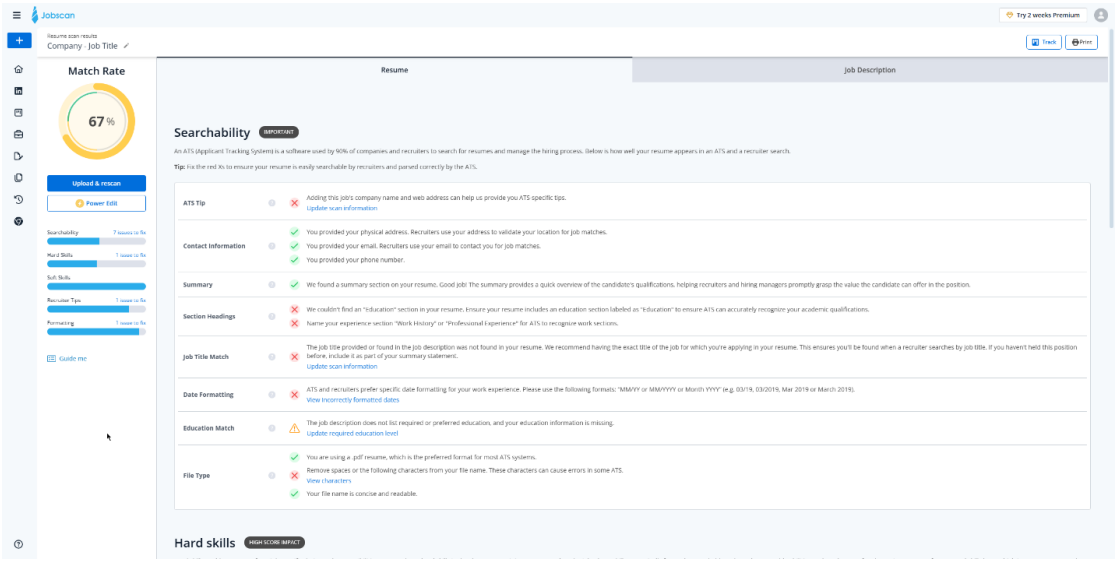

Рисунок 4 – Рекомендации ИИ по улучшению резюме Jobscan

2.2 Позиционирование системы

Разрабатываемое приложение направлено на пользователей из IT-сферы: разработчиков, тестировщиков, аналитиков, дизайнеров и других специалистов, которые ищут работу или хотят улучшить текущее резюме. В этом сегменте рынка особенно важно грамотно составленное резюме, соответствующее требованиям работодателей.

Таким образом, проект занимает нишу специализированных инструментов для улучшения резюме именно под IT, предоставляя функциональность, заточенную под требования технологического рынка труда.

3 Реализация

3.1 Средства реализации

3.1.1 Flutter

Для реализации клиентской части мобильного приложения использовались язык программирования Dart [4] версии 3.7 и Flutter SDK версии 3.29. Flutter представляет собой кроссплатформенный фреймворк [5], позволяющий разрабатывать мобильные, веб и десктопные приложения с использованием единой кодовой базы.

Использование Flutter обеспечивает ряд преимуществ:

— Ускоренная сборка пользовательских интерфейсов за счёт системы виджетов и объектно-ориентированного подхода;

— Высокая производительность благодаря компиляции в нативный код;

— Широкая экосистема сторонних библиотек, что упростило реализацию функциональных компонентов приложения, включая работу с сетью и хранением данных.

3.1.2 Go

Для разработки серверной части приложения используется язык программирования Go [6] версии 1.24. Go (или Golang) был выбран благодаря своей высокой производительности и встроенной поддержке параллелизма.

Go предоставляет встроенные механизмы работы с конкурентностью (goroutines), которые позволяют эффективно обрабатывать множество одновременных запросов без существенных затрат системных ресурсов.

3.1.3 PostgreSQL

Для хранения данных в проекте используется реляционная система управления базами данных PostgreSQL [7] версии 17.4. Данная СУБД выбрана благодаря своей надёжности и производительности. PostgreSQL обеспечивает устойчивую работу при высоких нагрузках.

3.1.4 Docker

Для развёртывания компонентов приложения используется система контейнеризации Docker версии 28.0.4. Docker позволяет упаковывать приложение и его зависимости в изолированные контейнеры, что обеспечивает переносимость между различными средами и упрощает процесс развёртывания.

Контейнеризация также повышает безопасность за счёт изоляции процессов: приложение работает в собственном окружении и не влияет на хост-систему. Для координации работы нескольких контейнеров используется инструмент Docker Compose версии v2.34.0-desktop.1.

3.2 Архитектура

В данной системе реализована архитектура, основанная на контейнеризации с использованием Docker Compose для управления компонентами: NGINX, Backend и PostgreSQL (Рисунок 5).

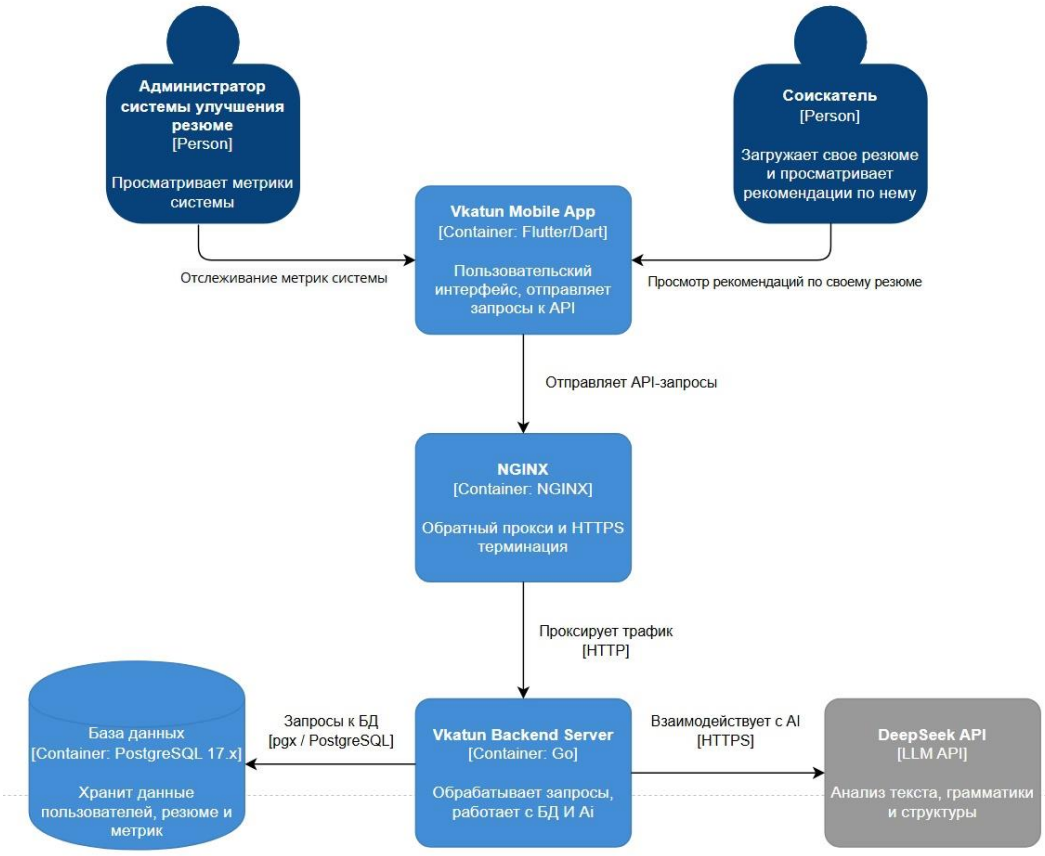

Рисунок 5 – Диаграмма контейнеров

В архитектуре приложения NGINX выполняет роль обратного прокси-сервера: принимает HTTP-запросы и перенаправляет их на бэкенд, реализованный на Go. Бэкенд обрабатывает загрузку резюме, формирует рекомендации, взаимодействует с базой PostgreSQL через драйвер pgx и отправляет запросы в DeepSeek API для анализа текста.

Передача данных между приложением и сервером осуществляется через REST API в формате JSON.

Для автоматизации используется GitHub Actions: при изменениях запускаются юнит-тесты, собирается Docker-образ и выполняется деплой.

Все действия проходят через сервер, который возвращает пользователю структурированный ответ и сохраняет метрики для администратора.

3.3 Серверная часть

Серверная часть системы Vkatun реализована на языке Go и обеспечивает всю основную бизнес-логику приложения: от загрузки резюме и их анализа до выдачи рекомендаций и работы с данными пользователя.

3.3.1 Компоненты

В системе используются следующие компоненты серверной части:

— API — маршрутизатор HTTP-запросов на базе Gorilla Mux. Отвечает за обработку входящих запросов от клиента, маршрутизацию по эндпоинтам, валидацию входных данных и возврат ответов в формате JSON;

— server — инициализационный слой, запускающий все компоненты: соединение с базой данных, миграции, логгер, HTTP-сервер;

— analyzer — модуль парсинга PDF-файлов и интеграции с языковой моделью (LLM);

— db — интерфейс доступа к PostgreSQL, реализованный на базе pgx; дополнительно для модульного тестирования реализована структура

MockDB на базе библиотеки testify, позволяющая эмулировать поведение базы данных без подключения к реальному хранилищу;

— models — структуры данных: User, Resume, Recommendation, DeepSeekRequest и др;

— logger — централизованная система логирования на базе zap;

— utils — утилиты для очистки текста, обработки markdown и служебных операций.

3.3.2 Хранение данных

Данные пользователей, резюме и метрик хранятся в PostgreSQL. Структура БД (Рисунок 6) включает:

— users — таблица с уникальным email, логином, хэшем пароля, датой создания и обновления;

— resume — таблица с основными полями: заголовок, контакты, должность, опыт, образование, навыки, раздел «О себе», user_id и временными метками;

— metrics — агрегированная таблица с количеством пользователей, резюме, внедрённых изменений и временной меткой последнего обновления.

Работа с базой данных реализована через модуль pgsql и пул соединений pgxpool. Все запросы выполняются через параметризованные SQL-запросы с возвратом структур моделей. Также в системе предусмотрены миграции схемы с использованием встроенного embed.FS, загружающего schema.sql при запуске.

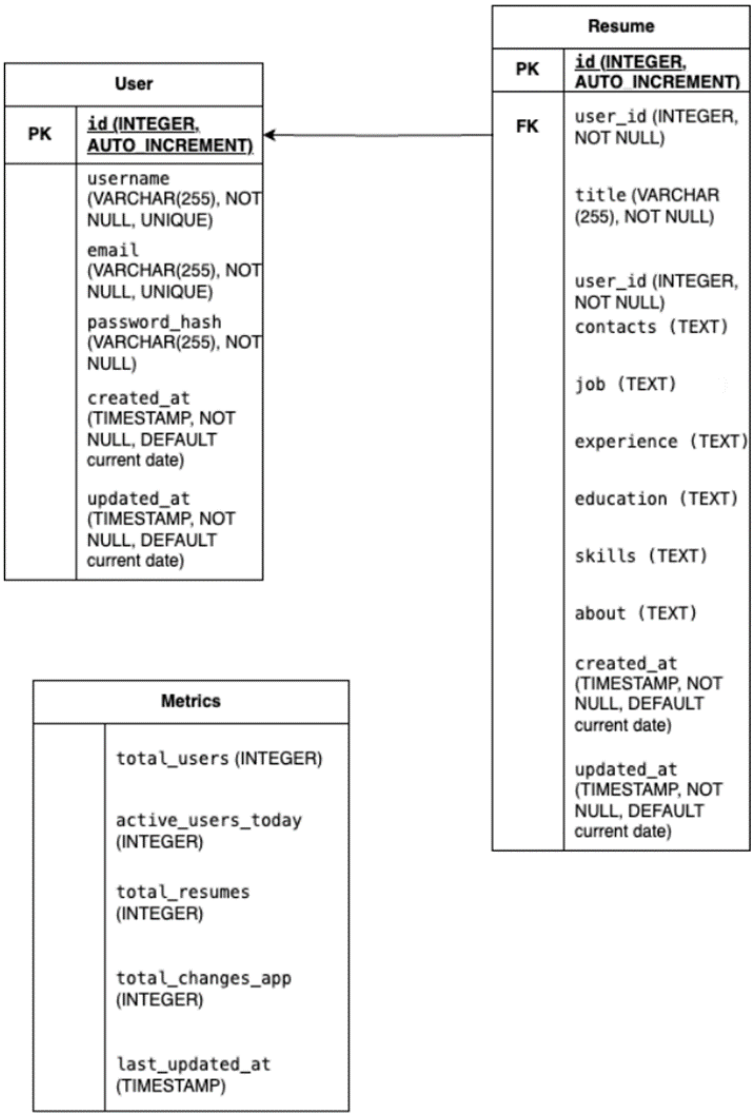

Рисунок 6 – Диаграмма сущностей

3.3.3 Загрузка резюме

Файл резюме в формате PDF загружается на сервер через эндпоинт /upload, после чего проходит последовательную обработку. Сначала осуществляется проверка формата файла и соблюдение ограничений по размеру. Далее выполняется извлечение текста с использованием библиотеки github.com/ledongthuc/pdf. Извлечённый текст очищается от невидимых символов, нормализуются пробелы, устраняются склеенные заголовки и восстанавливаются списки. Затем происходит интеллектуальная реструктуризация: текст автоматически разбивается на смысловые абзацы, заголовки выравниваются, списки получают маркировку. В завершение

предварительно обработанный текст отправляется в языковую модель DeepSeek с запросом на извлечение структурированной информации по шаблону ResumeInput.

Система автоматически обрабатывает резюме с различной структурой и форматированием. При парсинге извлекаются ключевые блоки. На выходе формируется JSON-объект, включающий поля: «Желаемая должность», «Контакты», «Опыт работы», «Образование», «Навыки», «О себе» и другие, необходимые для дальнейшей обработки.

LLM возвращает результат в markdown-обёртке (```json), который автоматически очищается и десериализуется в Go-структуру. Если пользователь авторизован – резюме сохраняется в базу, если нет — данные возвращается клиенту без сохранения.

Данный процесс представлен на диаграмме последовательности загрузки резюме (Рисунок 7).

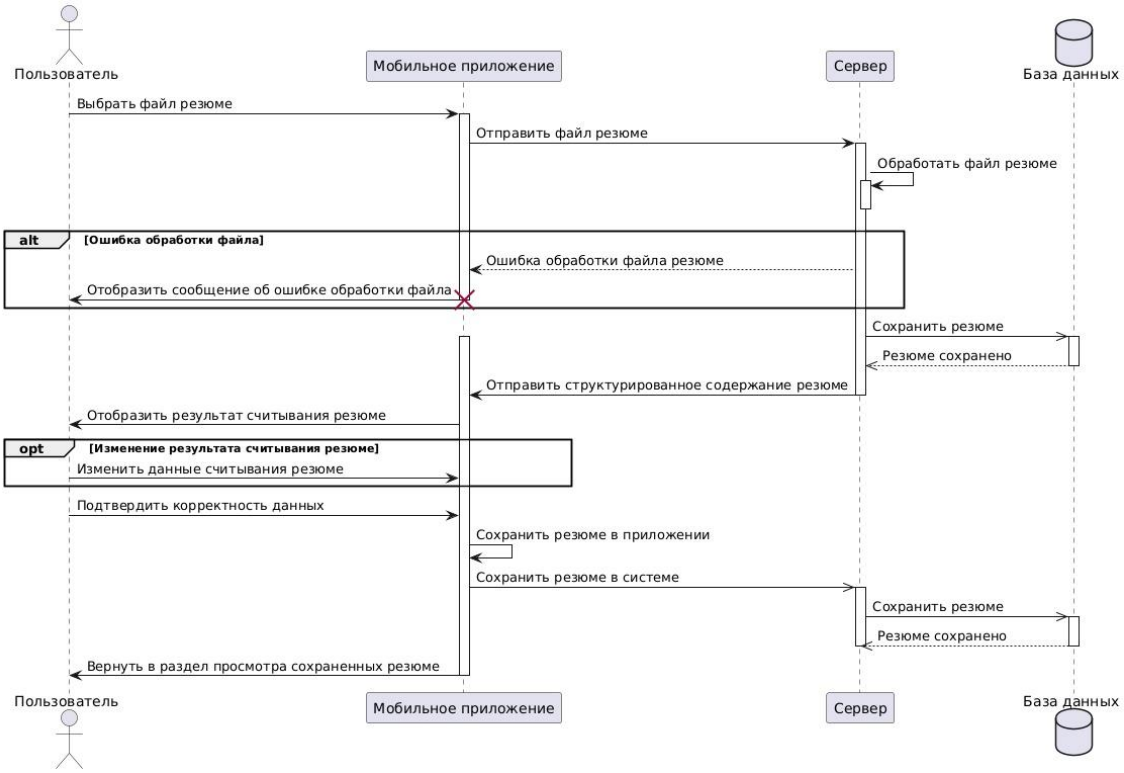

Рисунок 7 – Диаграмма последовательностей загрузки резюме

3.3.4 Выдача рекомендаций

Выдача рекомендаций происходит в несколько этапов. Сервер извлекает из резюме необходимый фрагмент текста, например раздел «О себе» или «Навыки». Далее формируется запрос с системной и пользовательской инструкцией в формате, аналогичном OpenAI Chat API, с указанием ролей system и user. Запрос отправляется в DeepSeek, после чего сервер получает ответ с разметкой, содержащей рекомендации. Из ответа удаляется markdown-обёртка ```json, затем результат парсится в структуры на языке Go. После этого пользователю возвращается список рекомендаций.

Поддерживаются следующие проверки:

— /resume/{id}/check/grammar — проверка орфографии, пунктуации и стилистики по всему резюме;

— /resume/{id}/check/grammar — проверка орфографии, пунктуации и стилистики по всему резюме;

— /resume/{id}/check/about — анализ раздела "О себе" и проверка наличия контактной информации;

— /resume/{id}/check/experience — оценка полноты и качества описания опыта работы;

— /resume/{id}/check/skills — фильтрация нерелевантных или подозрительных навыков.

Данный процесс представлен на диаграмме последовательности выдачи рекомендаций (Рисунок 8).

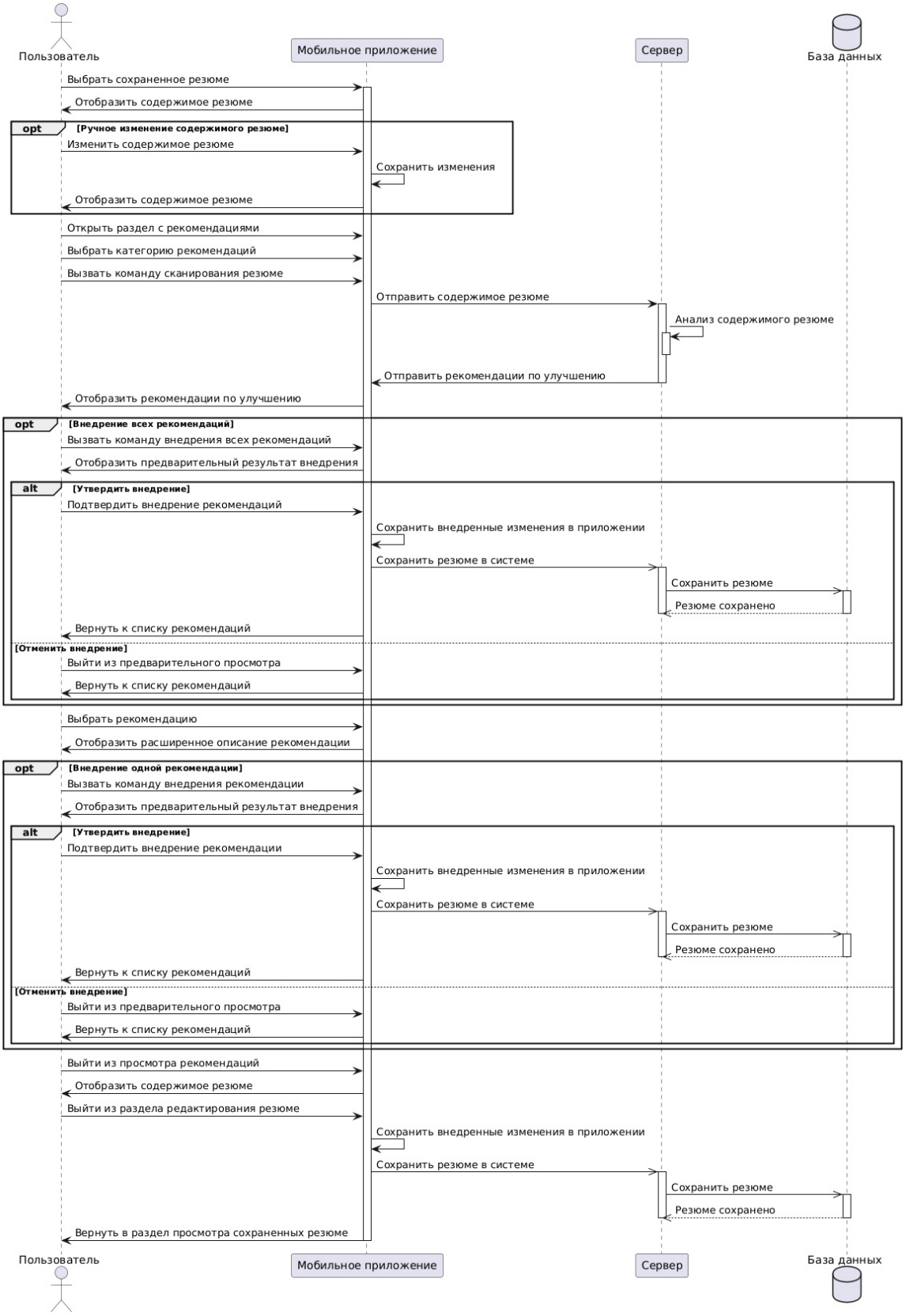

Рисунок 8 – Диаграмма последовательности выдачи рекомендаций

3.3.5 Редактирование резюме

Для редактирования резюме предусмотрены два отдельных эндпоинта:

— /resume/{id}/edit — обновление полной структуры резюме, включая все поля;

— /resume/{id}/edit/{section} — точечное редактирование конкретного раздела по его имени.

3.3.6 Авторизация в систему

Система авторизации реализована с использованием JWT (JSON Web Token). При входе пользователю выдается токен, содержащий его user_id и срок действия, установленный на 7 дней. Все защищённые маршруты требуют наличие действительного токена в заголовке Authorization: Bearer <token>.

Пароли обрабатываются с использованием библиотеки bcrypt. При регистрации пароль хешируется и сохраняется в базе данных. Во время аутентификации введённый пароль сравнивается с сохранённым хешем. Прямой доступ к исходному значению пароля отсутствует.

Реализованы маршруты для получения и изменения профиля пользователя: /profile, /profile/name, /profile/password. Все они защищены и доступны только при наличии валидного токена.

Для валидации данных и логирования используется библиотека zap. Все действия фиксируются централизованно.

3.3.7 Состав и конфигурация контейнерной среды

В текущей конфигурации docker-compose.yaml описаны следующие сервисы:

— pg — сервис базы данных PostgreSQL (версия 17.4-alpine3.21). Выполняется проброс порта на хост-машину (54322:5432), подключается внешний файл окружения .env для конфигурации, монтируется том postgres_volume для обеспечения персистентного

хранения данных. Также задан healthcheck, позволяющий отслеживать готовность БД к приёму соединений;

— backend — основной сервис приложения, сборка которого осуществляется из Dockerfile с использованием многоступенчатого подхода: на первом этапе в контейнере golang:1.24.0-alpine загружаются зависимости и производится компиляция исходного кода, после чего готовый бинарный файл переносится в минимальный образ alpine для последующего запуска;

— nginx — обратный прокси-сервер, сконфигурированный для обработки HTTP- и HTTPS-запросов. В контейнер монтируются конфигурационные и TLS-файлы (nginx.conf, nginx.crt, nginx.key) в режиме только для чтения.

Сервисы объединены в одну виртуальную сеть и связаны между собой через директиву depends_on, что гарантирует корректную последовательность запуска. Развёртывание всей инфраструктуры осуществляется единой командой: docker-compose up –build

3.3.8 CI/CD

В проекте используется GitHub Actions для автоматизации процессов сборки, тестирования и доставки обновлений. Настроен workflow backend_vkatun_ci.yaml, который запускается автоматически при выполнении push или pull request в ветки main или master. Сценарий включает этапы сборки и проверки backend-компонента и используется в качестве части CI/CD-процесса.

В ходе CI/CD выполняются три действия. В job run-go-unit-tests запускается сборка проекта и выполнение всех модульных тестов с помощью команды go test. В job image-build-and-push происходит авторизация в Docker Registry по адресу cr.selcloud.ru, затем создаётся Docker-образ с использованием Buildx, тегируется по значению переменной GITHUB_SHA и

загружается в реестр. В job deploy-image копируются файлы docker-compose.yaml, .env и nginx.conf на сервер в Selectel Cloud. Подключение выполняется по SSH с использованием ключа из GitHub Secret SSHKEY, после чего выполняется перезапуск контейнеров. Переменные окружения передаются в файл .env автоматически.

3.4 Клиентская часть

Клиентская часть реализована на Flutter с использованием многослойной архитектуры, обеспечивающей разделение между компонентами. Основой управления состоянием выбран Provider в сочетании с ChangeNotifier. Для работы с локальным хранилищем используется SharedPreferences (для гостевого режима) и SQLite (для авторизованных пользователей), обеспечивающие хранение данных даже при отсутствии интернет-соединения.

Поведение резюме в различных состояниях отражено на диаграмме состояний (Рисунок 9).

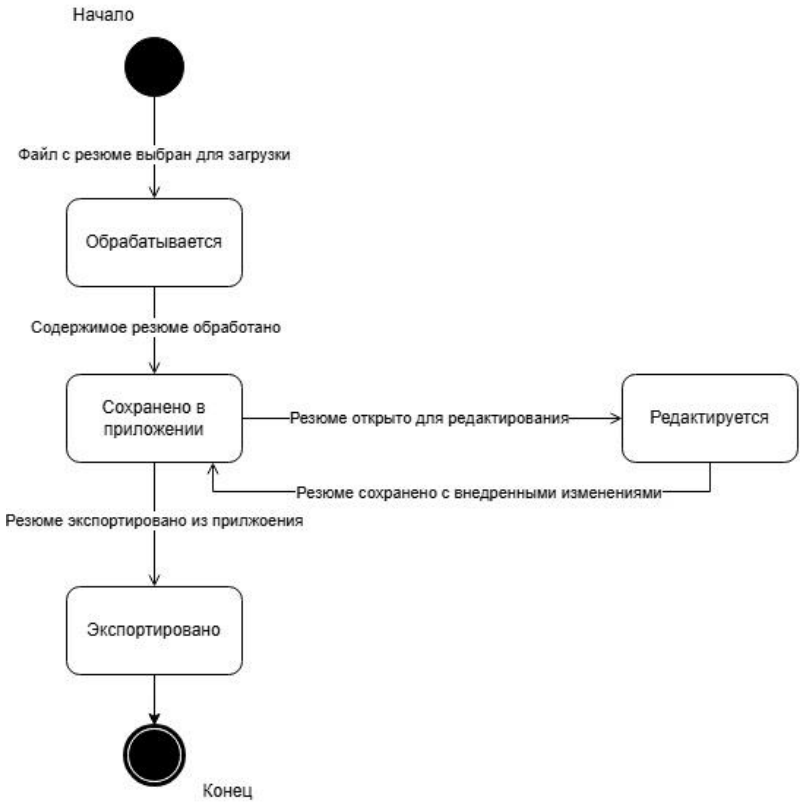

Рисунок 9 – Диаграмма состояний

3.4.1 Обработка резюме

Процесс загрузки резюме начинается с выбора PDF-файла через модифицированный интерфейс библиотеки file_picker. Приложение проверяет выбранный файл на соответствие формату PDF. Если файл не подходит по формату, он не отображается в выборе. Если пользователь достиг лимита по количеству загруженных резюме, отображается уведомление об ограничении.

После валидации файл отправляется на сервер через multipart/form-data с использованием библиотеки Dio. Во время загрузки пользователь видит индикатор прогресса. Если сервер успешно обрабатывает файл и возвращает корректный JSON-ответ, данные преобразуются в экземпляр класса Resume, который добавляется в список резюме пользователя. Дополнительно JSON сохраняется на устройстве с помощью SharedPreferences.

Если сервер возвращает ошибку или происходит сбой на клиентской стороне, пользователь получает уведомление об ошибке.

После успешной загрузки пользователь переходит на страницу ResumeViewPage. Там доступно ручное редактирование полей. Если были внесены изменения, резюме сохраняется: для неавторизованного пользователя — только локально, для авторизованного — синхронизируется с сервером и также сохраняется на устройстве.

3.4.2 Система рекомендаций ИИ

Интерфейс рекомендаций реализован в виде кастомного Dialog, использующего анимацию SlideTransition. Для отображения поверх основного контента применяется OverlayEntry. Окно поддерживает жесты drag, позволяющие пользователю закрыть его вручную.

Для управления состоянием используется TabController, обеспечивающий переключение между категориями, и PageView для горизонтальной навигации. Обновление состояния реализовано через ValueNotifier.

Сами рекомендации отображаются внутри кастомного ListView с разделителями между элементами. Смена содержимого сопровождается анимацией AnimatedSwitcher, обеспечивающей плавное обновление интерфейса. Компонент адаптируется под разные размеры экранов и плотность пикселей.

3.4.3 Внедрение рекомендаций

Класс ApplyCorrections представляет собой StatefulWidget, предназначенный для визуального сравнения оригинального и исправленного резюме с возможностью применения автоматических исправлений. Компонент используется на этапе после анализа ошибок и позволяет пользователю просматривать различия и подтверждать изменения перед сохранением.

При инициализации в методе initState() создаётся копия исходного резюме и автоматически применяются все исправления с помощью метода _applyCorrection(). Для синхронизации прокрутки между двумя панелями используется метод _syncScroll(), который вычисляет относительное положение и синхронизирует его между оригинальной и изменённой версией. Прокрутка реализована через вызов jumpTo, что обеспечивает мгновенное перемещение без анимации.

Метод _applyCorrection() выполняет поиск строк, содержащих исходный текст с ошибкой (errorText), и заменяет их на предложенное исправление (suggestion).

Вся структура адаптируется под размеры экрана: высота AppBar вычисляется динамически, а для отображения текстовых блоков используется RichText, что обеспечивает гибкое форматирование. Компонент применяется как финальный этап взаимодействия с рекомендациями и служит инструментом для окончательного внедрения исправлений в резюме.

3.5 Реализация функциональных возможностей системы

3.5.1 Роли пользователей в системе

В системе реализовано три основные роли пользователей, каждая из которых обладает определённым набором действий (Рисунок 10).

Неавторизованный пользователь — это гость системы, не прошедший авторизацию. Ему доступны следующие функции: загрузка своего резюме в приложение, удаление резюме из приложения, изменение компонентов резюме, скачивание резюме, создание аккаунта и вход в аккаунт. Также неавторизованный пользователь может просматривать ограниченный список рекомендаций ИИ. Все действия доступны только с одним резюме. Загрузка и работа с несколькими документами недоступна.

Авторизованный пользователь — это зарегистрированный пользователь, выполнивший вход в систему. Он имеет доступ ко всем функциям неавторизованного пользователя, а также может сохранять в приложении несколько резюме, просматривать полный список рекомендаций ИИ, внедрять в резюме рекомендации ИИ и изменять данные своего аккаунта.

Администратор системы — это авторизованный пользователь с расширенными правами. Дополнительно к возможностям авторизованного пользователя он может просматривать метрики приложения.

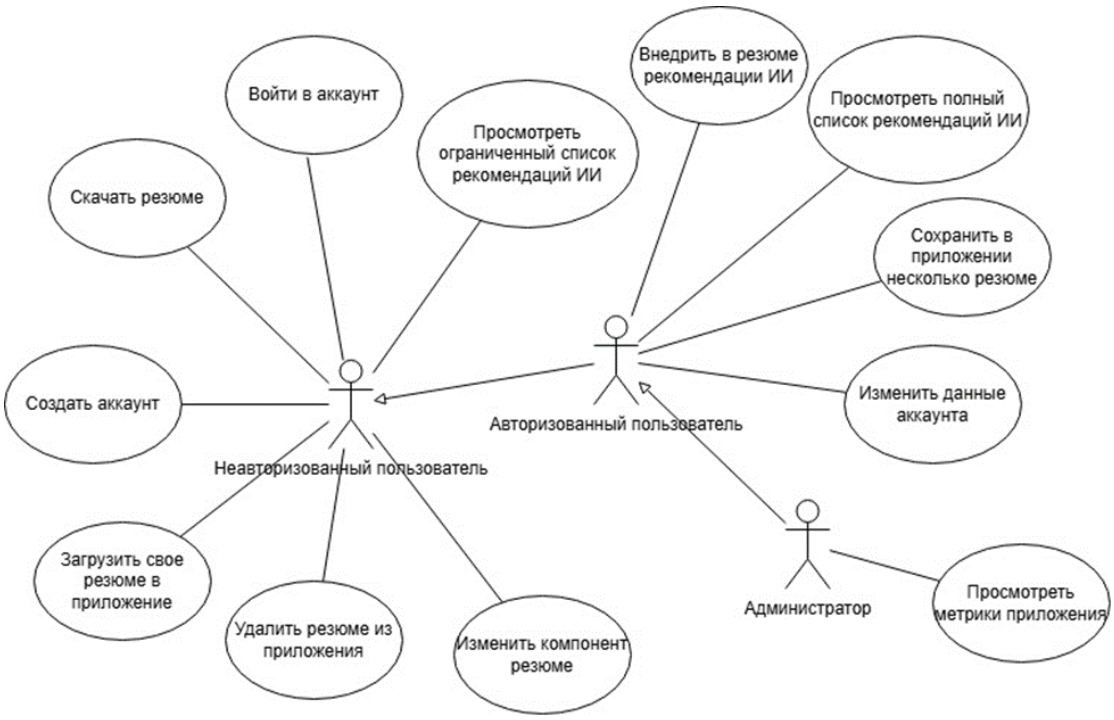

Рисунок 10 – Диаграмма прецендентов

3.5.2 Функциональные возможности

3.5.2.1 Загрузка резюме

Система поддерживает загрузку файлов в формате PDF, включая версии 1.4 и выше. При добавлении резюме приложение обрабатывает только документы, содержащие структурированные блоки: контактные данные, опыт работы, образование, навыки и раздел «О себе». Обработка выполняется только для резюме, составленных на русском языке. В случае отсутствия одного или нескольких указанных разделов, соответствующая информация не распознается и не отображается.

После выбора и загрузки файла система приступает к его обработке и выводит пользователю результат распознавания. Если файл не является PDF-документом или при его обработке возникла ошибка, пользователю отображается сообщение об ошибке.

После успешной обработки содержимое резюме распознается и отображается в приложении в виде структурированных данных. В случае, если некоторые данные были распознаны с ошибками, пользователь получает

возможность вручную внести корректировки через соответствующие поля ввода.

После подтверждения корректности внесённых данных система сохраняет резюме в системе. Загруженное и отредактированное резюме остаётся доступным в приложении и может использоваться для дальнейшей работы.

Порядок выполнения действий при загрузке резюме отображён на диаграмме активности загрузки резюме (Рисунок 11).

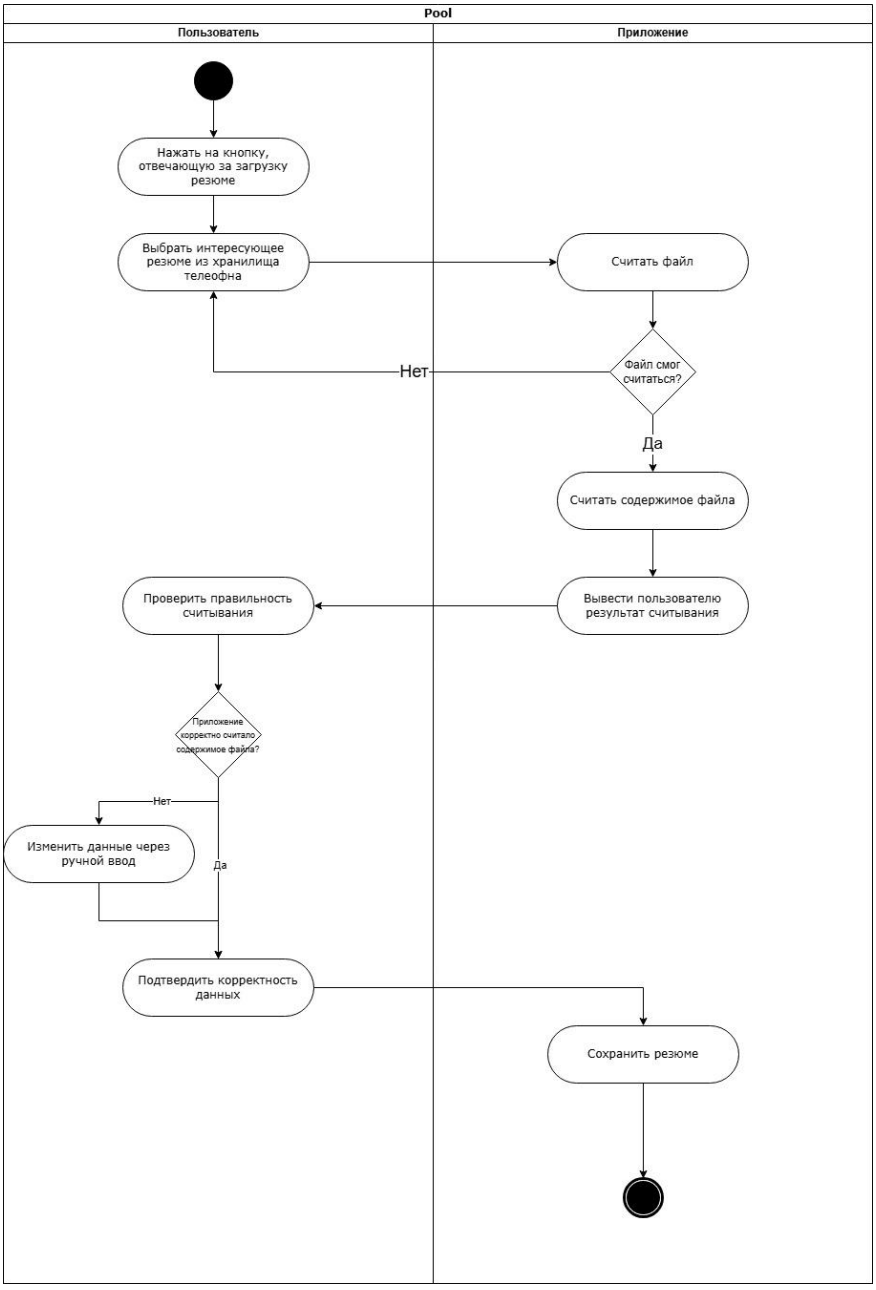

Рисунок 11 – Диаграмма активности загрузки резюме

3.5.2.2 Доступ к сохраненным в приложении резюме

Система предоставляет пользователю возможность просматривать ранее загруженные резюме и выполнять с ними различные действия. Пользователю доступен список всех загруженных документов, при этом общее количество резюме, хранимых на одного пользователя, ограничено пятнадцатью.

При выборе конкретного резюме пользователь может перейти к его редактированию через интерфейс приложения. Также доступна функция удаления резюме, которая сопровождается обязательным подтверждением перед окончательным удалением документа. Кроме того, система позволяет экспортировать выбранное резюме в формате PDF.

3.5.2.3 Редактирование резюме

Система предоставляет пользователю возможность вручную редактировать текст в отдельных разделах резюме. В разделе ФИО допускается ввод текста длиной до 100 символов, состоящего из русских букв и пробелов. В поле желаемой должности разрешено использовать до 100 символов, включая русские буквы, пробелы и дефис.

В разделе «Опыт работы» пользователь может указать название компании и должность — каждое поле до 100 символов. Даты работы вводятся в формате ММ.ГГГГ – ММ.ГГГГ или с указанием «по настоящее время». Описание обязанностей ограничено 1000 символами. Всего можно добавить до 20 мест работы.

Раздел «Образование» позволяет указывать учебные заведения (до 5 записей), где на каждую запись предусмотрено до 300 символов для названия учебного заведения, до 100 символов для специальности и диапазон годов обучения в формате ГГГГ – ГГГГ.

В блоке «Навыки» пользователь может ввести до 20 навыков, каждый длиной от 1 до 50 символов. Навыки разделяются запятыми или точками с запятой. В разделе «О себе» допускается ввод текста до 500 символов.

После внесения изменений система предоставляет возможность сохранить обновлённую версию резюме. Все изменения отображаются при последующем просмотре резюме.

3.5.2.4 Просмотр рекомендаций ИИ

При открытии раздела редактирования резюме система автоматически отображает рекомендации, сгенерированные искусственным интеллектом. Все рекомендации разделены по категориям и оформлены по единому шаблону. Каждая из них включает три части: элемент с ошибкой, описание проблемы и предложенный вариант исправления.

Система выделяет отдельную категорию — исправление ошибок. В неё входят орфографические, грамматические, пунктуационные и стилистические ошибки. Орфографические ошибки представляют собой опечатки и неверное написание слов. Например, слово «безчувственный» система распознаёт как ошибочное, объясняет, что допущена ошибка в написании, и предлагает верный вариант — «бесчувственный». Грамматические ошибки касаются неправильного построения предложений, нарушений в согласовании слов и форм глаголов. Пунктуационные ошибки — это отсутствие или неправильное размещение знаков препинания, таких как запятые или двоеточия. Стилистические ошибки охватывают канцеляризмы, тяжёлые обороты и недостаточную выразительность текста. Все исправления соответствуют нормам современного русского языка.

Следующая категория рекомендаций касается структуры резюме. Система проверяет, насколько логично оформлен текст. Если в разделе, например, «Опыт работы» отсутствует информация, или весь текст идёт сплошным абзацем без деления на смысловые блоки, система предлагает внести необходимые изменения.

Категория улучшения содержания охватывает смысловое наполнение разделов. Если описание опыта работы слишком короткое (менее 20 слов), система предлагает его расширить. В разделе «Навыки» проверяется

уместность указанных умений. Если пользователь написал в желаемой должности «Разработчик», а в списке навыков указал, например, «игра на гитаре» или «фотография», система определяет их как нерелевантные и рекомендует удалить. Рекомендации по навыкам применяются только в тех случаях, когда указана должность «Разработчик», и анализ касается языков JavaScript, Python, PHP, Ruby, Java, C#, Go, C++, Kotlin, Swift, Objective-C, Dart.

3.5.2.5 Внедрение рекомендаций

Система предоставляет авторизованному пользователю возможность внедрять рекомендации, сгенерированные искусственным интеллектом, если они поддерживаются для автоматического внесения изменений.

Пользователь может внедрять рекомендации из следующих категорий: исправление ошибок, улучшение структуры (разбивка текста на абзацы) и улучшение содержания (удаление нерелевантных навыков).

При просмотре рекомендаций пользователь может выбрать любую из поддерживаемых. Перед применением изменений система показывает предварительный просмотр результата. После подтверждения изменения автоматически вносятся в резюме.

Порядок выполнения действий при внедрении рекомендаций показан на диаграмме активности просмотра и внедрение рекомендаций ИИ (Рисунок 12).

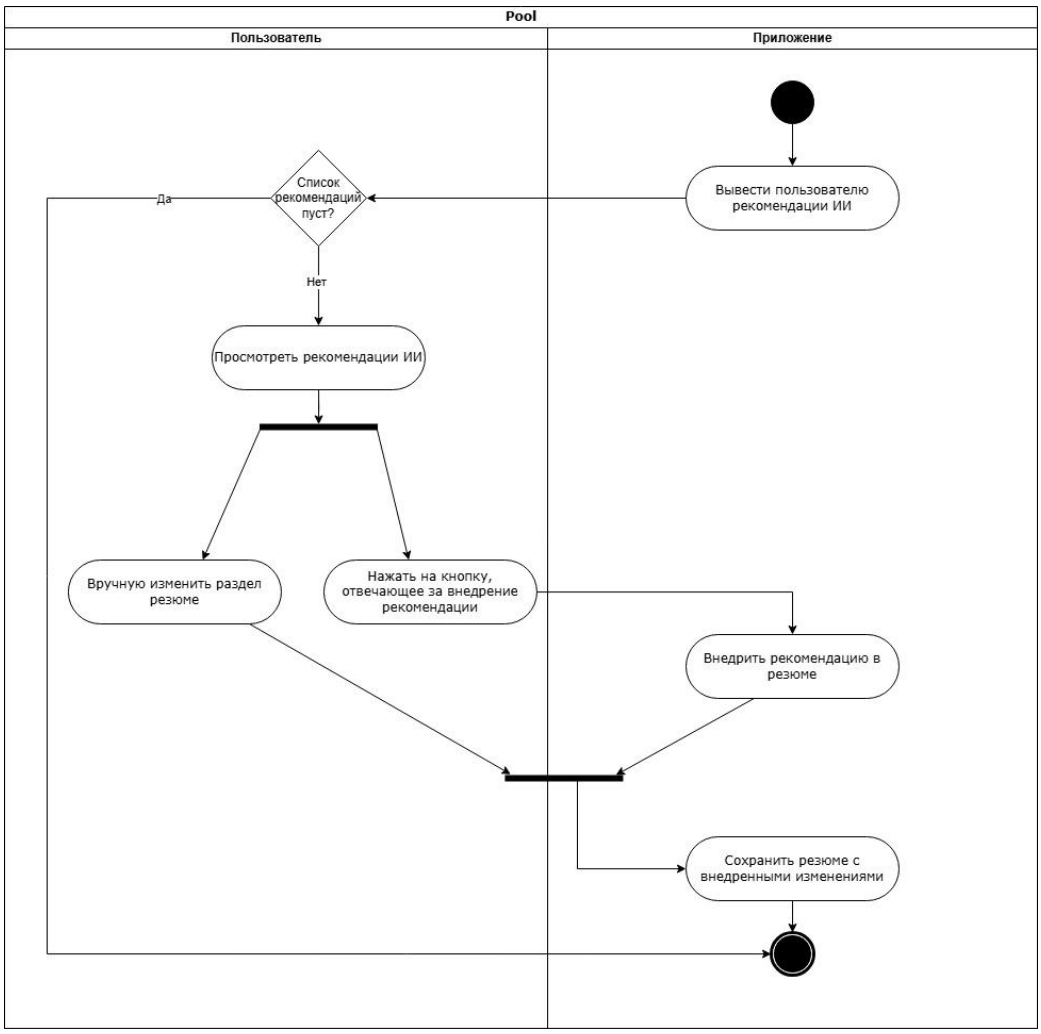

Рисунок 12 – Диаграмма активности просмотра и внедрения рекомендаций ИИ

Если рекомендация относится к исправлению ошибок (орфографических, пунктуационных, грамматических или стилистических), система производит автоматическую замену ошибочного текста на предложенный исправленный вариант.

Если рекомендация связана с разбивкой текста на абзацы, система форматирует текст, добавляя соответствующие разрывы.

Если рекомендация касается удаления нерелевантных навыков, система автоматически удаляет такие навыки из раздела «Навыки».

3.5.2.6 Экспорт резюме

Система позволяет пользователю экспортировать резюме в формате PDF. При экспорте формируется файл, включающий все данные,

представленные в приложении: ФИО, контактные данные, желаемая должность, опыт работы, образование, навыки и раздел «О себе».

Пользователь выбирает нужное резюме через интерфейс приложения, после чего система инициирует процесс формирования PDF-документа. По завершении этого процесса пользователю отображается кнопка для скачивания сформированного файла.

В случае успешного экспорта система выводит уведомление о завершении операции и предоставляет доступ к файлу. Если в ходе экспорта возникает ошибка, пользователю отображается соответствующее сообщение с указанием проблемы.

Содержимое экспортируемого файла полностью соответствует актуальной версии резюме, отображённой в приложении. Все изменения, внесённые пользователем до момента экспорта, учитываются при формировании итогового документа.

3.5.2.7 Авторизация пользователя в личный кабинет

Система реализует функциональность авторизации и регистрации пользователей. При входе в приложение пользователь вводит адрес электронной почты и пароль в соответствующие поля формы. Если данные введены корректно, система авторизует пользователя и автоматически перенаправляет его в личный кабинет. В случае неверного ввода учетных данных приложение уведомляет пользователя об ошибке и предлагает повторить попытку.

Если у пользователя отсутствует учетная запись, система предоставляет возможность создать новый аккаунт. В процессе регистрации необходимо указать имя пользователя длиной от 3 до 30 символов, уникальный email длиной от 3 до 100 символов и пароль от 8 до 25 символов, содержащий минимум одну цифру и один специальный символ. После успешной регистрации пользователь автоматически авторизуется и получает доступ к личному кабинету.

Система выполняет проверку всех введённых данных. При несоответствии email стандартному формату, несоблюдении требований к сложности пароля, использовании уже зарегистрированного email или попытке входа с некорректными данными система отображает соответствующее сообщение об ошибке и не допускает продолжения действия.

После успешного входа или завершения регистрации пользователь перенаправляется в личный кабинет, где может воспользоваться полным набором функций приложения.

Последовательность действий при авторизации и регистрации пользователя представлена на диаграмме активности авторизации в систему (Рисунок 13).

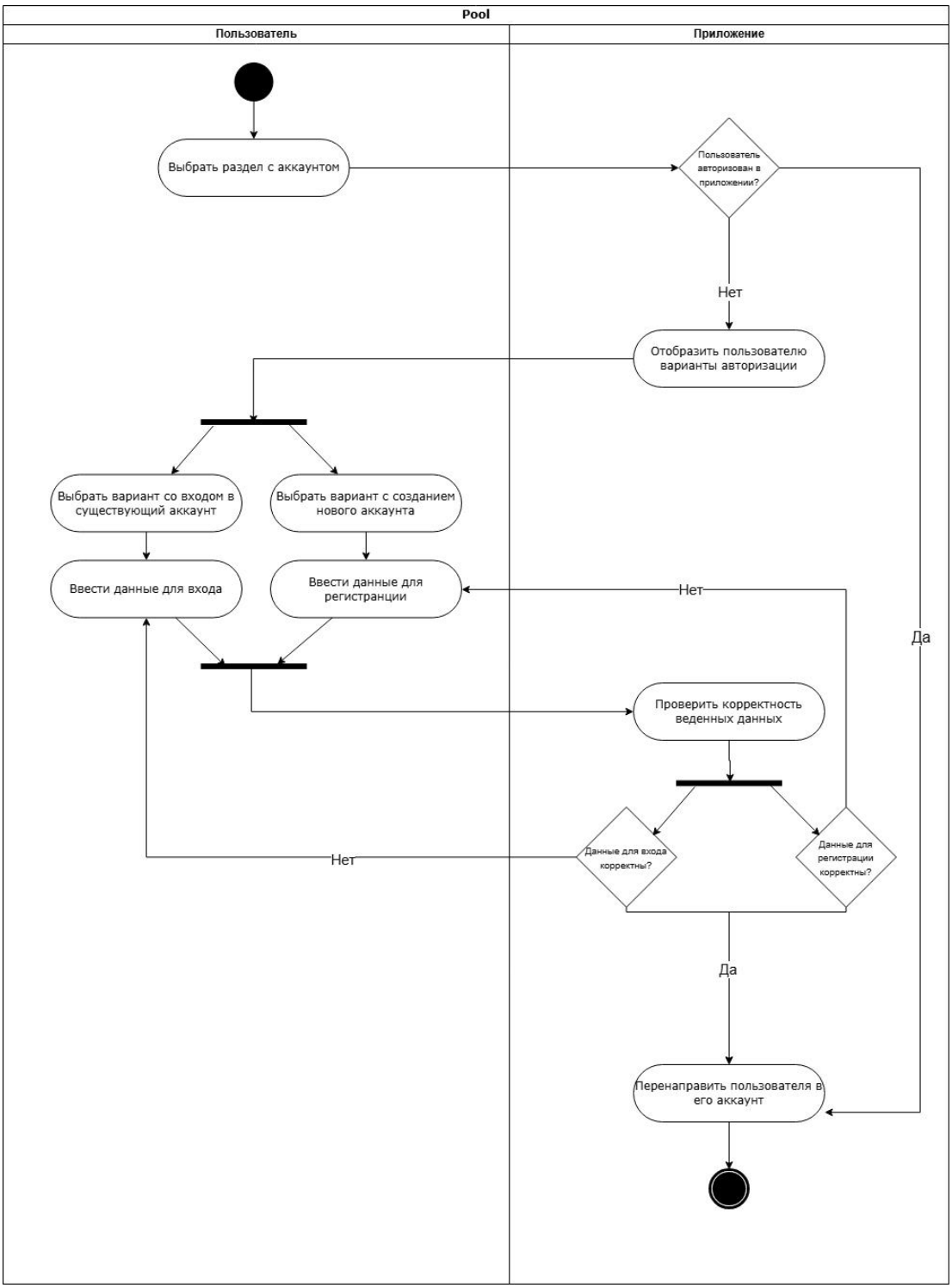

Рисунок 13 – Диаграмма активности авторизации в систему

3.5.2.8 Управление данными аккаунта

Система предоставляет пользователю возможность управления данными аккаунта через интерфейс личного кабинета. В соответствующем разделе отображаются имя пользователя и электронная почта, указанные при регистрации.

Пользователь может изменить имя и пароль с помощью специальной формы редактирования. Для имени действует ограничение длины от 3 до 30 символов. Пароль должен содержать от 8 до 25 символов и включать как минимум одну цифру и один специальный символ. После ввода новых данных требуется подтверждение их сохранения.

Перед сохранением система выполняет проверку корректности введённой информации. В случае обнаружения ошибок — таких как несоблюдение требований к длине или составу имени и пароля — отображается уведомление с указанием проблемы, которую необходимо исправить.

При успешном редактировании данные обновляются в системе и остаются актуальными при последующих входах в аккаунт. Обновлённая информация становится доступной пользователю сразу после сохранения.

3.5.2.9 Администрирование приложения

Система предоставляет администратору доступ к основным метрикам работы приложения через интерфейс. В приложении отображаются следующие показатели: общее количество загруженных пользователями резюме, количество активных пользователей за последние 24 часа, общее число зарегистрированных аккаунтов, а также процент рекомендаций, которые были приняты и внедрены пользователями.

Администратор может просматривать данные за различные периоды времени, включая последние 24 часа, последнюю неделю и последний месяц. Метрики обновляются автоматически и отображаются в актуальном виде при каждом обращении к административному разделу.

3.5.2.10 Onboarding

Onboarding запускается один раз при первом запуске приложения и кратко знакомит пользователя с основными экранами и возможностями: загрузкой резюме и получением рекомендаций. После завершения onboarding пользователь переходит к основному сценарию работы.

3.5.3 Режимы функционирования системы

В онлайн-режиме система предоставляет пользователю полный функционал, требующий подключения к интернету. Пользователь может загружать резюме, которое отправляется на сервер для обработки. После обработки система отображает рекомендации, сформированные алгоритмами искусственного интеллекта на основе содержимого резюме. Пользователь получает возможность редактировать резюме с учётом предложенных рекомендаций и сохранять обновлённую версию. Обработка документа и генерация рекомендаций выполняются на серверной стороне, поэтому при отсутствии интернет-соединения анализ и обновление данных становятся недоступными.

В оффлайн-режиме система предоставляет доступ к ранее загруженным и сохранённым в приложении резюме. Пользователь может открывать такие резюме, просматривать их содержимое, а также вносить изменения. При этом анализ с использованием ИИ недоступен.

3.5.4 Ограничение и показатели значений

Система обеспечивает отклик от сервера на запрос мобильного приложения для получения рекомендаций по улучшению резюме в течение не более 5 минут. После отправки запроса приложение получает обработанные рекомендации в установленный промежуток времени.

Обработка загруженного пользователем резюме выполняется в течение не более 1 минуты. Отсчёт начинается с момента выбора файла пользователем и завершается после успешной обработки и передачи данных в приложение.

Формирование экспортированного файла резюме выполняется системой не более чем за 1 минуту. Отсчёт начинается с момента запуска пользователем операции экспорта и завершается после создания файла и предоставления его для скачивания.

При парсинге сложных или плохо структурированных PDF могут возникать проблемы с распознаванием текста, если используются

нестандартные шрифты или разметка. Это приводит к искажению или отсутствию некоторых данных в результате неправильного распознавания.

Система предназначена для работы исключительно с резюме пользователей, ищущих работу в сфере IT. Резюме, относящиеся к другим сферам или сильно отклоняющиеся от типичных форматов, могут быть трудными для правильной интерпретации и генерирования рекомендаций.

3.6 Тестирование

В ходе тестирования приложения Vkatun использовались ручное тестирование, юнит-тестирование и автоматизированные сценарии. Были проверены все основные функции как на стороне пользователя, так и администратора.

Ручное тестирование проводилось через Postman и охватывало базовые сценарии работы API. Юнит-тестирование выполнялось с использованием моков и охватывало обработчики. Автоматизированные тесты были реализованы в Postman. Проверялись как позитивные, так и негативные сценарии: успешные операции, ошибки, валидация данных и авторизация.

Пользовательский интерфейс тестировался вручную. Были проверены действия: регистрация, вход в систему, загрузка и отображение резюме, переходы между страницами, экспорт резюме, внедрение ИИ-рекомендаций и взаимодействие с сервером.

Все действия выполняются в соответствии с заданной логикой. Ошибки отображаются, переходы между экранами работают стабильно, запросы к API обрабатываются корректно. Элементы интерфейса реагируют на состояния без сбоев.

Проведённое тестирование подтвердило, что все основные функции системы работают согласно требованиям.

3.7 Реализация интерфейсов системы

3.7.1 Отображение сохраненных резюме и взаимодействие с ними

Авторизованный пользователь на главном экране может выбрать интересующее его сохранённое резюме, зажав соответствующий элемент. Во всплывающем окне появляется список действий, которые пользователь может выполнить с резюме: «Редактировать резюме», «Экспорт резюме», «Удалить резюме» (Рисунок 14).

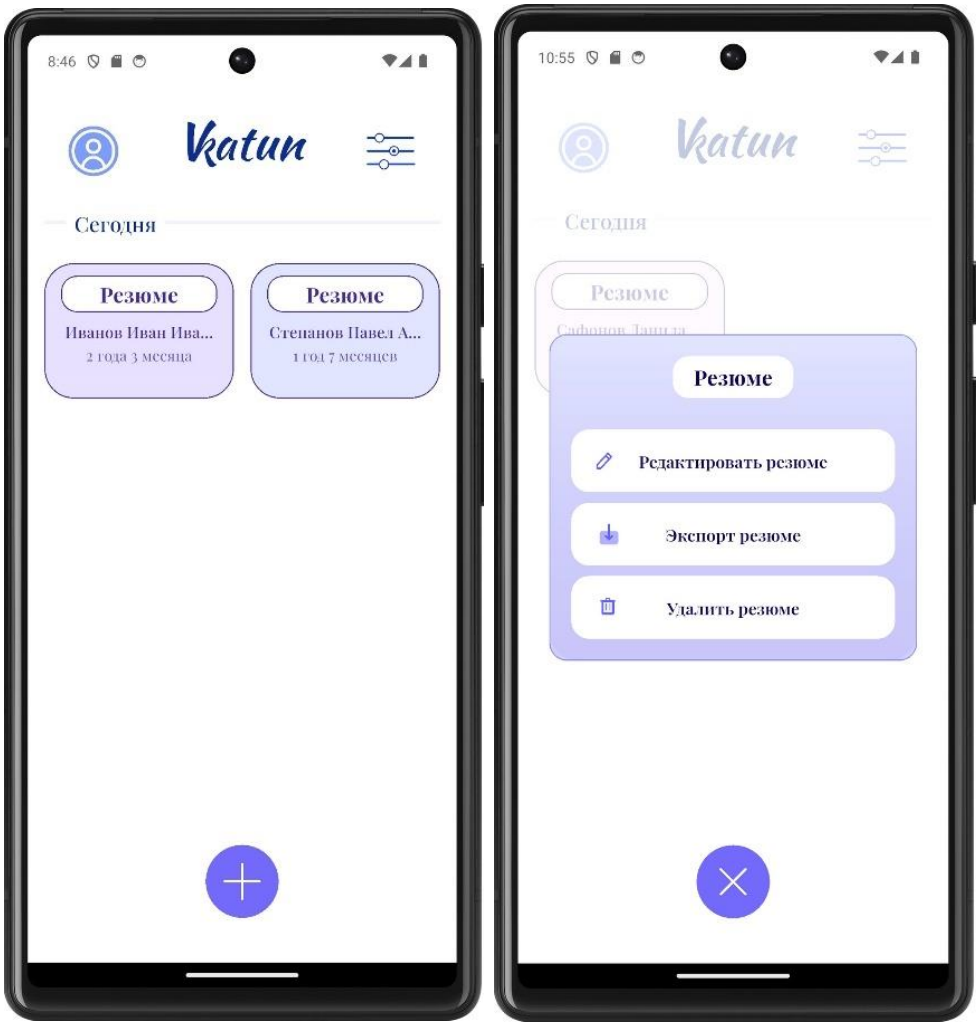

Рисунок 14 – Отображение сохраненных резюме и взаимодействие с ними

3.7.2 Загрузка резюме

Чтобы загрузить резюме, пользователь на главном экране может нажать кнопку загрузки, после чего попадает в хранилище файлов и выбирает нужное резюме. Затем открывается экран предварительного редактирования, где

пользователь может внести правки в соответствующие разделы и сохранить резюме, нажав кнопку подтверждения (Рисунок 15).

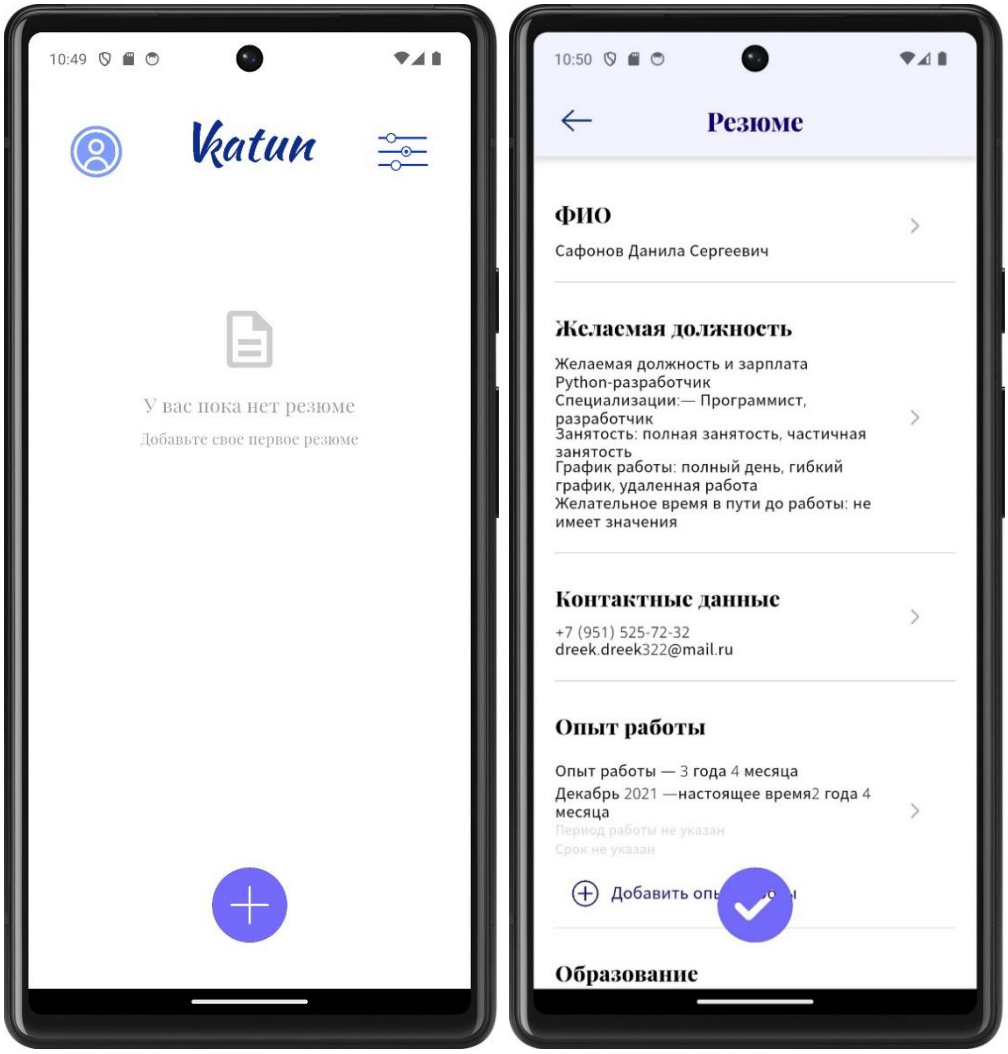

Рисунок 15 – Загрузка резюме

3.7.3 Редактирование резюме

На экране редактирования резюме пользователь может редактировать его разделы. На следующем экране он может внести правки в соответствующие поля и сохранить изменения (Рисунок 16).

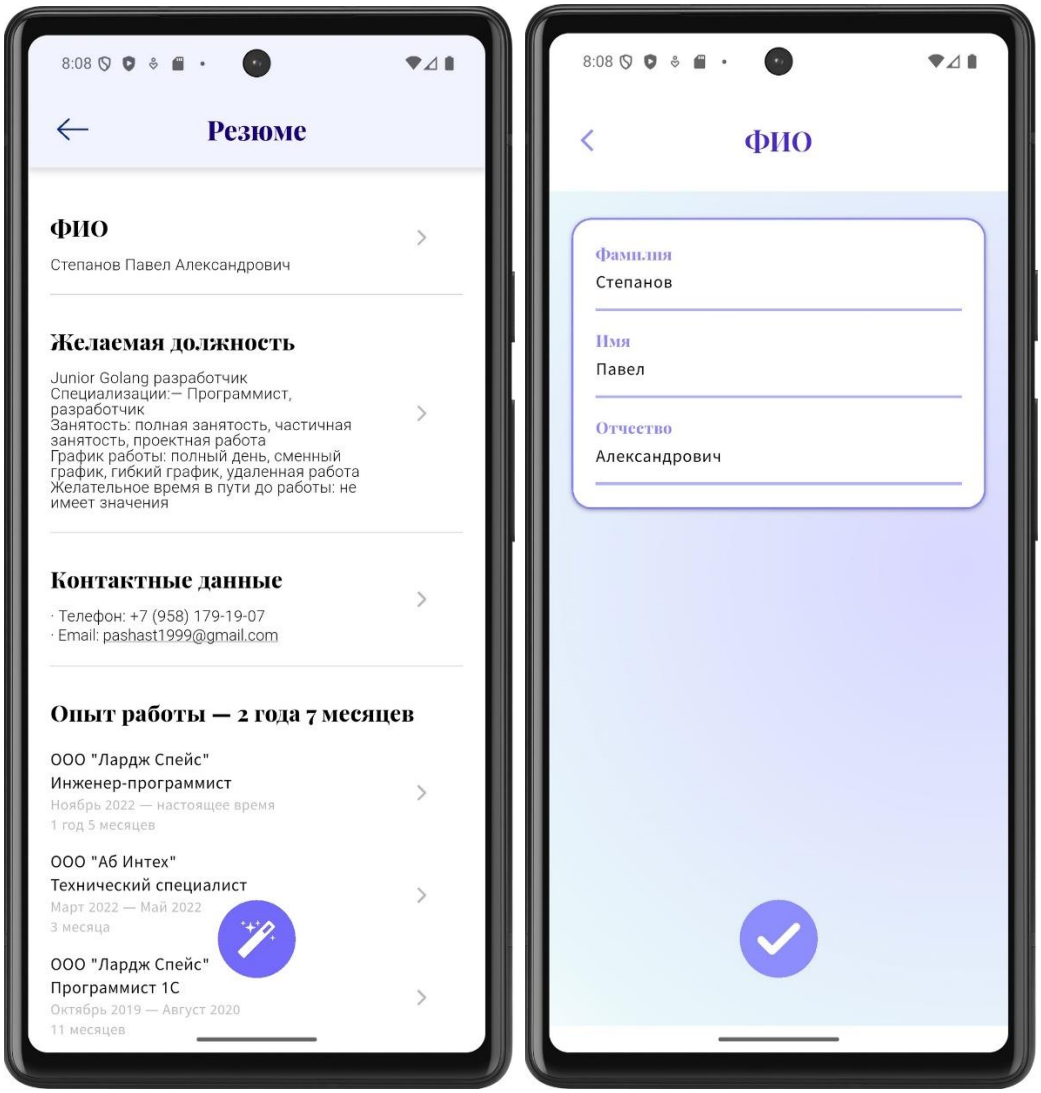

Рисунок 16 – Раздел редактирования резюме

3.7.4 Просмотр и внедрение рекомендаций

При просмотре рекомендаций пользователь может просканировать свое резюме на наличие ошибок (Рисунок 17).

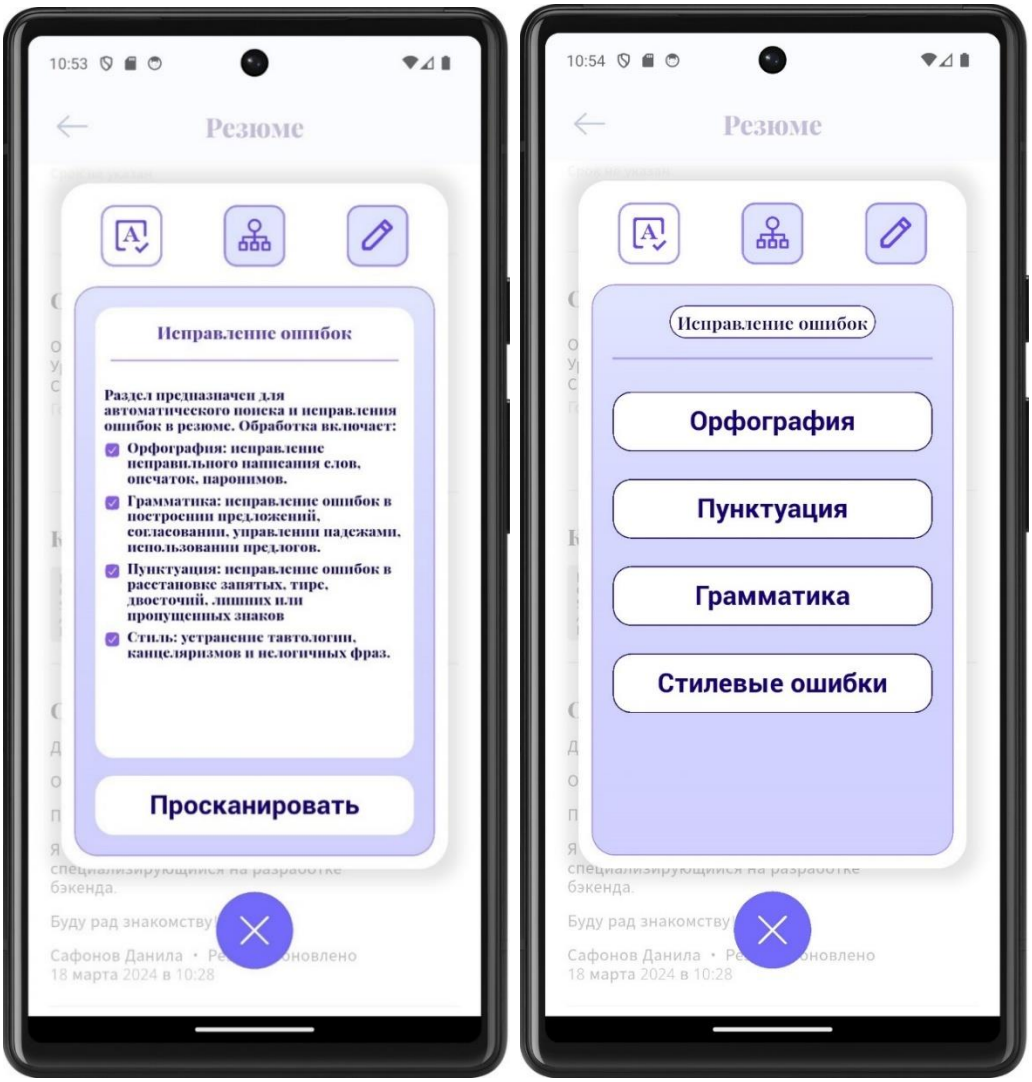

Рисунок 17 – Описание раздела рекомендаций и взаимодействие с ним

После сканирования пользователь может ознакомиться со списком рекомендаций и внедрить их. Во время внедрения одновременно отображаются текущий вариант резюме и вариант с внесёнными исправлениями (Рисунок 18).

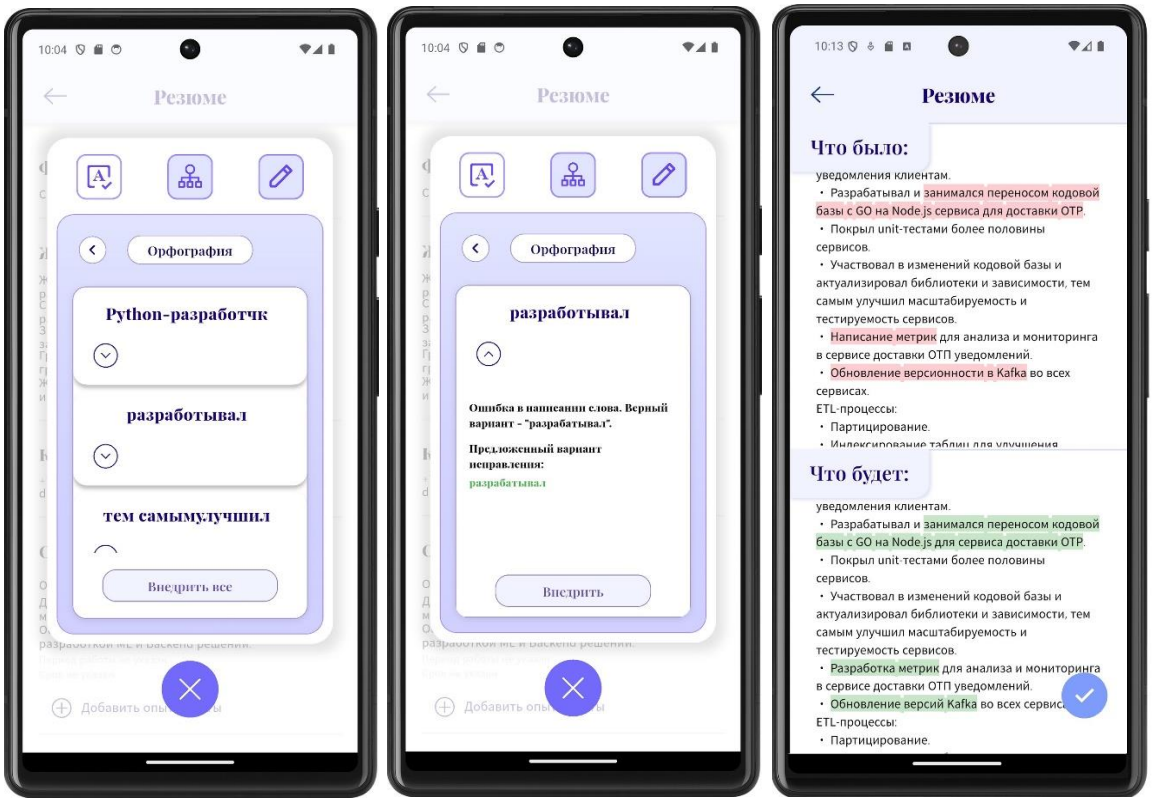

Рисунок 18 – Просмотр и внедрение рекомендаций

3.7.5 Авторизация в личный кабинет и управление данными аккаунта

Пользователь может авторизоваться в системе либо через вход, либо через регистрацию. Если выбран вариант регистрации, на экране создания аккаунта необходимо заполнить поля «Имя пользователя», «Адрес электронной почты» и другие, после чего нажать кнопку «Создать аккаунт». Если пользователь выбирает вход, то на экране входа он должен заполнить поля «Email» и «Пароль» и нажать кнопку «Войти» (Рисунок 19).

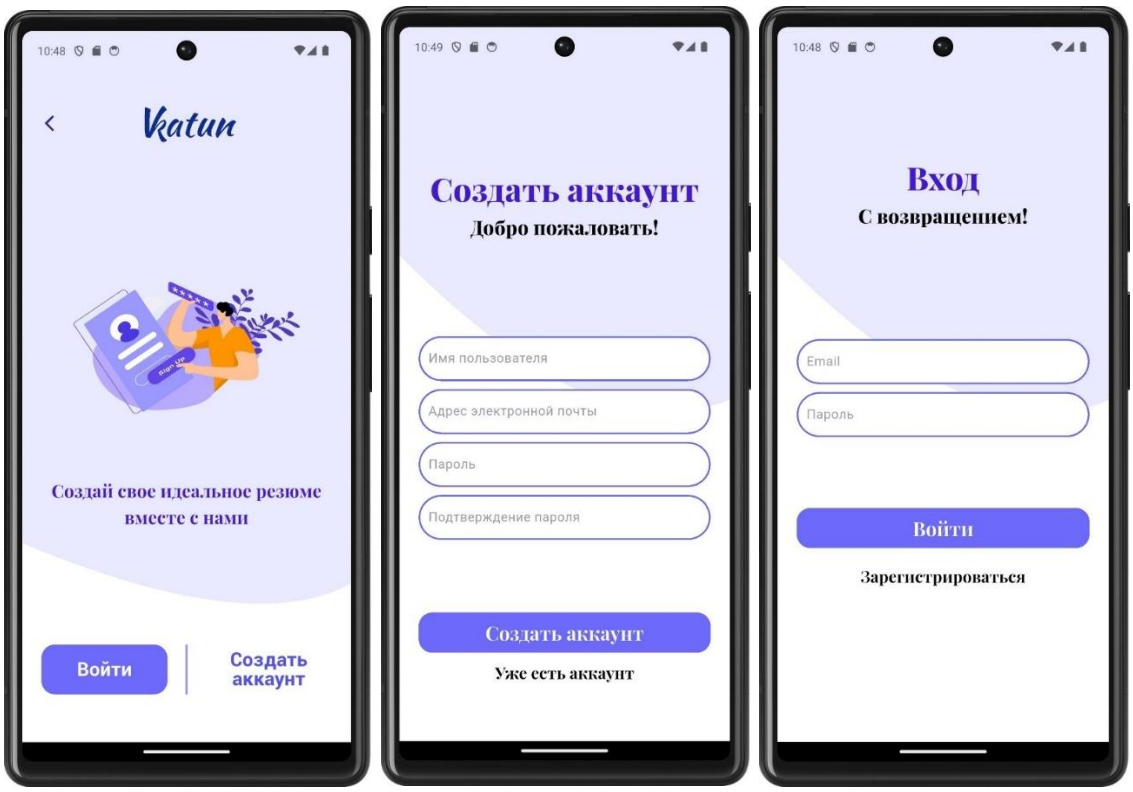

Рисунок 19 – Авторизация в систему

На экране управление данными аккаунта пользователь может изменить данные аккаунта, а также выйти из него (Рисунок 20).

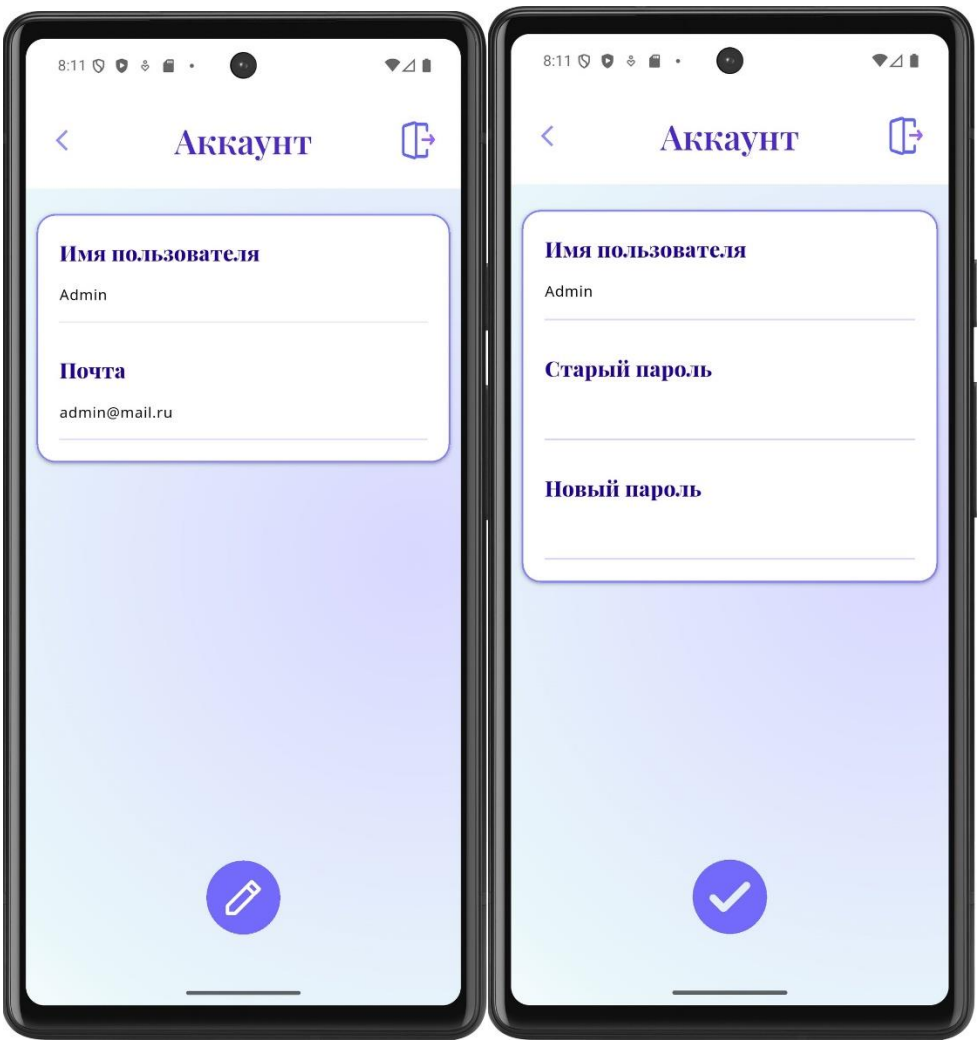

Рисунок 20 – Управление данными аккаунта

3.7.6 Администрирование

Администратор приложения может получить доступ к системным метрикам, нажав соответствующую кнопку в разделе «Администратор». В разделе метрик отображаются данные с возможностью выбора временного интервала (Рисунок 21).

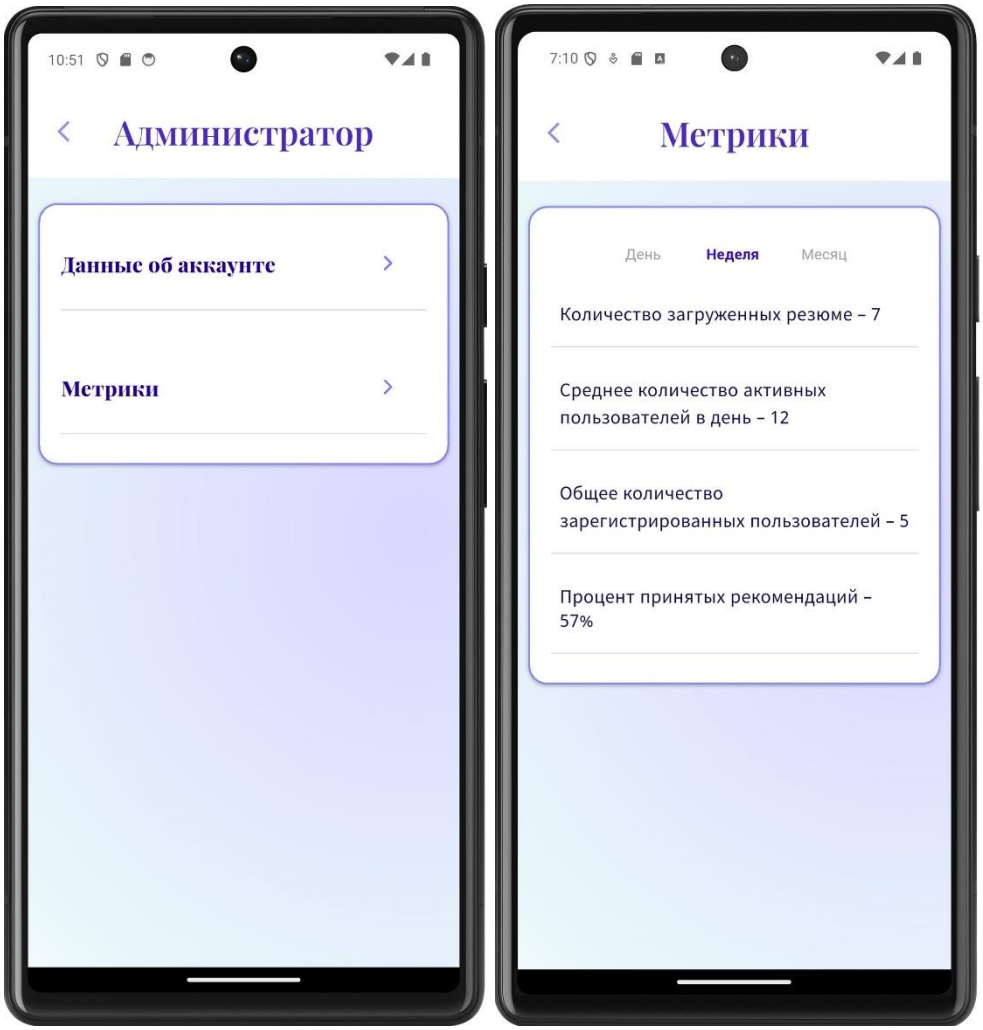

Рисунок 21 – Просмотр системных метрик

4 Аналитика

Для тестирования работоспособности приложения доступ к нему был предоставлен студентам факультета компьютерных наук. Приложение было запущено 12 пользователями. На основе собранных данных была построена воронка по сценарию «Работа с рекомендациями».

Из общего числа пользователей 83,33% загрузили резюме, 33,33% перешли к просмотру рекомендаций, и только 25% внедрили их в резюме (Рисунок 22). Это показывает, что значительная часть пользователей останавливается на этапе после загрузки файла. Можно сделать вывод, что элементы, связанные с отображением рекомендаций и применением изменений, требуют доработки с точки зрения понятности и доступности в пользовательском интерфейсе.

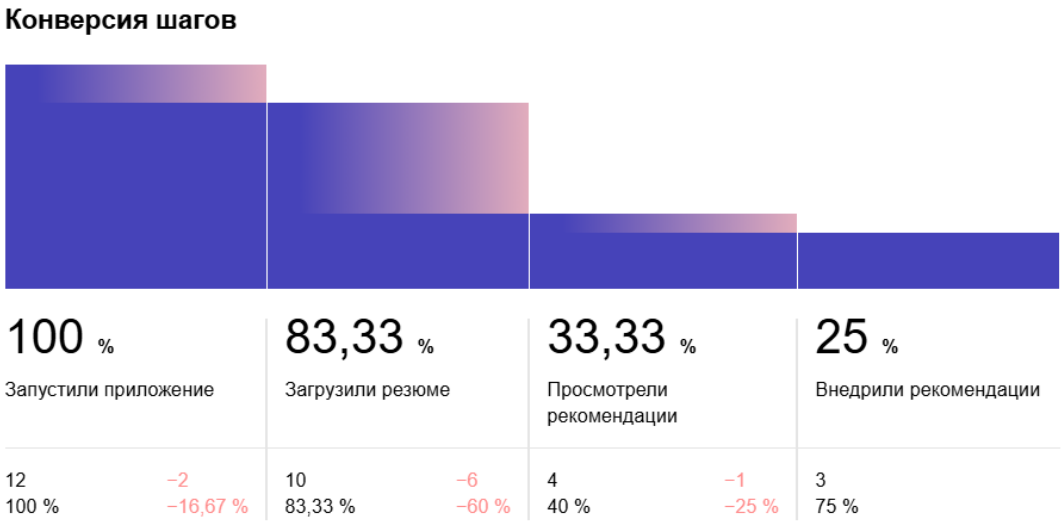

Рисунок 22 – Воронка просмотра и внедрения рекомендаций

Заключение

В ходе выполнения курсового проекта была реализована основная функциональность мобильного приложения Vkatun, предназначенного для анализа и улучшения резюме с помощью ИИ.

Были выполнены следующие задачи:

— Реализована загрузка и хранение пользовательских резюме;

— Разработан функционал генерации рекомендаций по улучшению резюме;

— Обеспечена возможность ручного редактирования резюме внутри приложения;

— Реализован экспорт готового резюме в виде документа.

В процессе реализации были использованы инструменты командной разработки, сформирована архитектура системы и обеспечена работа основных компонентов.

Разработанное приложение готово к использованию и может быть расширено в будущем для поддержки подписочной модели монетизации и дополнительных функций.

Список использованных источников

1. Электронный сервис VisualCV [Электронный ресурс]. – Режим доступа: URL: https://www.visualcv.com/ru/ – (Дата обращения 17.03.2025)
2. Электронный сервис Teal [Электронный ресурс]. – Режим доступа: URL: https://www.tealhq.com/tools/resume-builder – (Дата обращения 17.03.2025)
3. Электронный сервис Jobscan [Электронный ресурс]. – Режим доступа: URL: https://www.jobscan.co/ – (Дата обращения 17.03.2025)
4. Документация языка программирования Dart версии 3.7 [Электронный ресурс]. – Режим доступа: https://dart.dev/docs – (Дата обращения 27.03.2025)
5. Документация Flutter SDK версии 3.29 [Электронный ресурс]. – Режим доступа: https://docs.flutter.dev/ – (Дата обращения 05.04.2025)
6. Документация языка программирования Go версии 1.24 [Электронный ресурс]. – Режим доступа: https://go.dev/doc/ – (Дата обращения 27.03.2025)
7. Документация СУБД PostgreSQL версии 17.4 [Электронный ресурс]. – Режим доступа: https://www.postgresql.org/docs/ – (Дата обращения 27.03.2025)