МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Мобильное приложение Vkatun для анализа и улучшения резюме на основе ИИ

Курсовой проект

по дисциплине

Технологии программирования

09.03.04 Программная инженерия

Информационные системы и сетевые технологии

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_ В.С. Тарасов, ст. преподаватель \_\_.\_\_.20\_\_

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

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.Д. Проскуряков, ассистент

Воронеж 2025

содержание

[Определения, обозначения и сокращения 3](#_Toc196058122)

[Введение 4](#_Toc196058123)

[1 Постановка задачи 5](#_Toc196058124)

[1.1 Цели создания системы 5](#_Toc196058125)

[1.2 Функциональные требования к разрабатываемой системе 5](#_Toc196058126)

[2 Анализ предметной области 7](#_Toc196058127)

[2.1 Анализ рынка сервисов по улучшению резюме 7](#_Toc196058128)

[3 Реализация 11](#_Toc196058129)

[3.1 Средства реализации 11](#_Toc196058130)

[3.2 Архитектура 11](#_Toc196058131)

[3.3 Серверная часть 11](#_Toc196058132)

[3.3.1 Функционал 11](#_Toc196058133)

[3.4 Клиентская часть 11](#_Toc196058134)

[3.4.2 Функционал 11](#_Toc196058135)

[3.4.3 Реализация интерфейсов 11](#_Toc196058136)

[Заключение 12](#_Toc196058137)

[Список использованных источников 13](#_Toc196058138)

Определения, обозначения и сокращения

TBD

|  |  |
| --- | --- |
| Термин | Определение |
| Android | операционная система с открытым исходным кодом, созданная для мобильных устройств на основе модифицированного ядра Linux |

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. 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. Функциональные требования к разрабатываемой системе

Разрабатываемая система должна обеспечивать реализацию следующих функциональных возможностей:

* Загрузка существующего резюме для его последующего анализа и редактирования;
* Генерация персонализированных рекомендаций по улучшению резюме на основе анализа структуры и содержания документа;
* Возможность ручного редактирования компонентов резюме в интерфейсе приложения;
* Функция экспорта готового резюме;
* Отображения метрик и статистики работы приложения в интерфейсе администратора.

1. Анализ предметной области
   1. Анализ рынка сервисов по улучшению резюме

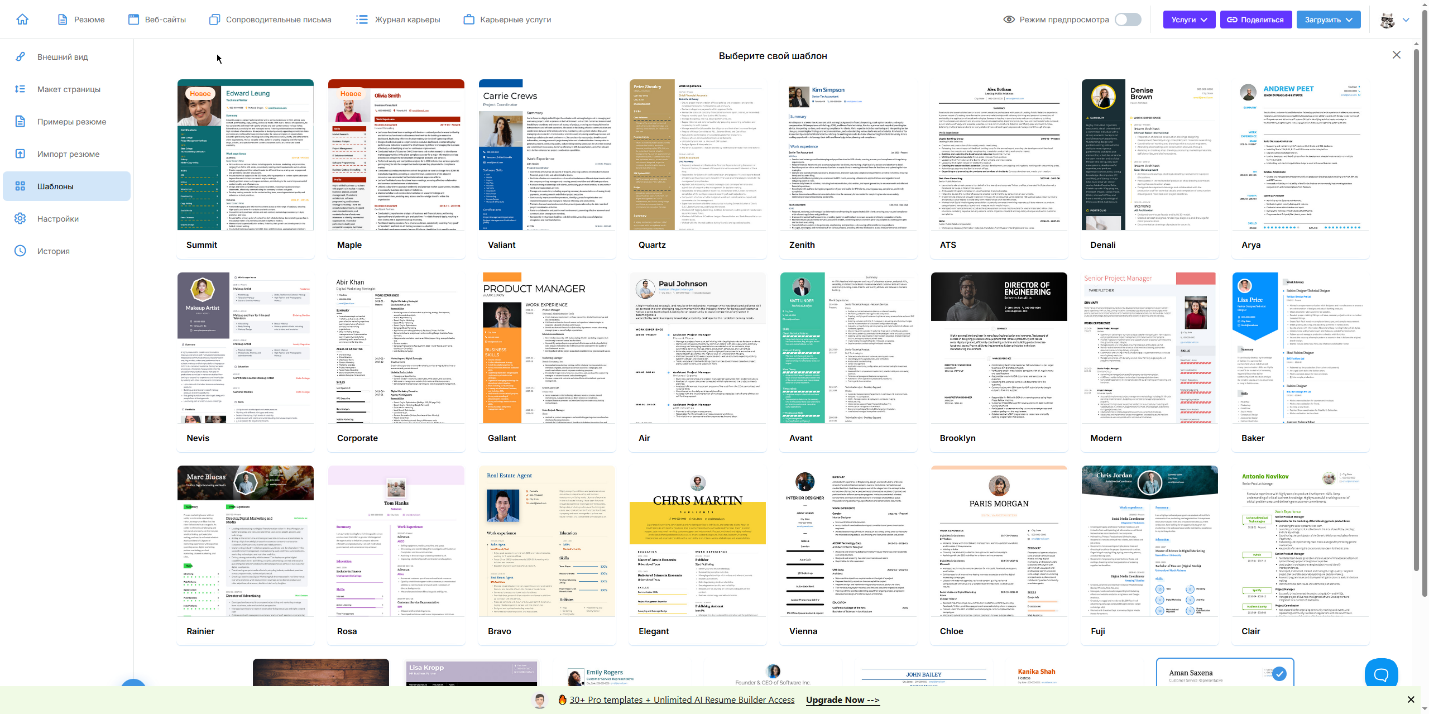
TBD (форматирование поменять картинок и ссылок на них в этом пункте)

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



— Бенчмаркинг основных конкурентов

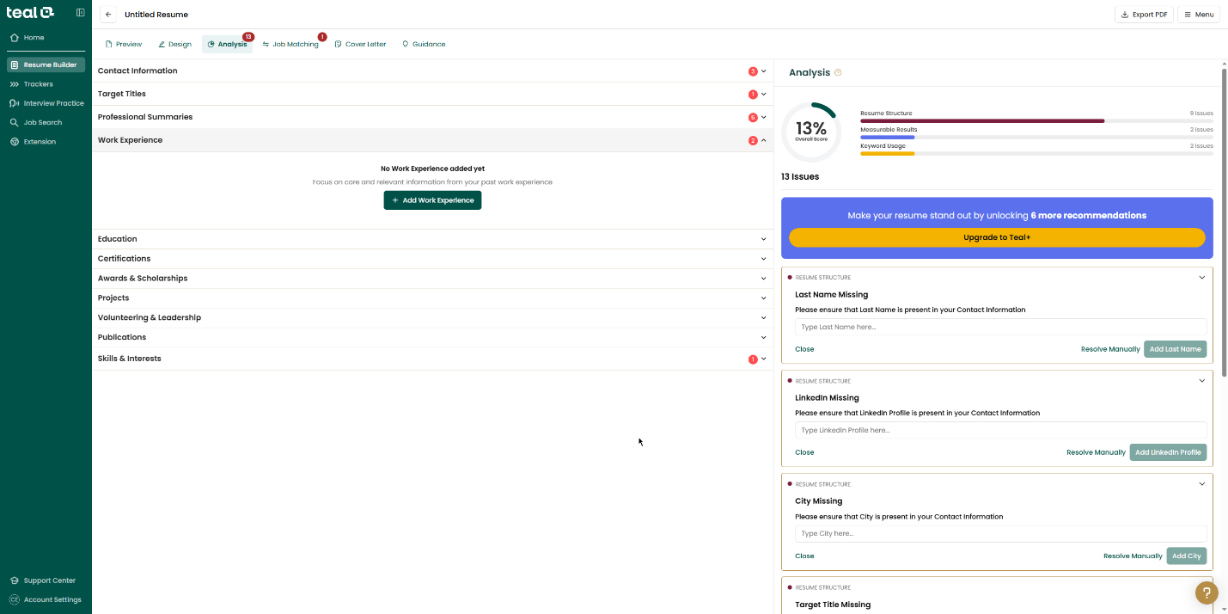
VisualCV — это онлайн-платформа для создания и редактирования профессиональных резюме, а также сопроводительных писем и портфолио (Рисунок А.2 ). Пользователи могут выбрать один из множества доступных шаблонов для оформления документа, а также импортировать данные из LinkedIn для автоматического создания 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.

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.

— Шаблоны резюме в VisualCV, доступные для пользователей

Teal — это сервис для создания и оптимизации резюме с использованием аналитики и рекомендаций по ключевым словам (Рисунок А.3 ). Основная функция платформы — это анализ резюме с точки зрения автоматических систем отслеживания (ATS), которые применяются большинством работодателей для фильтрации резюме. 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.

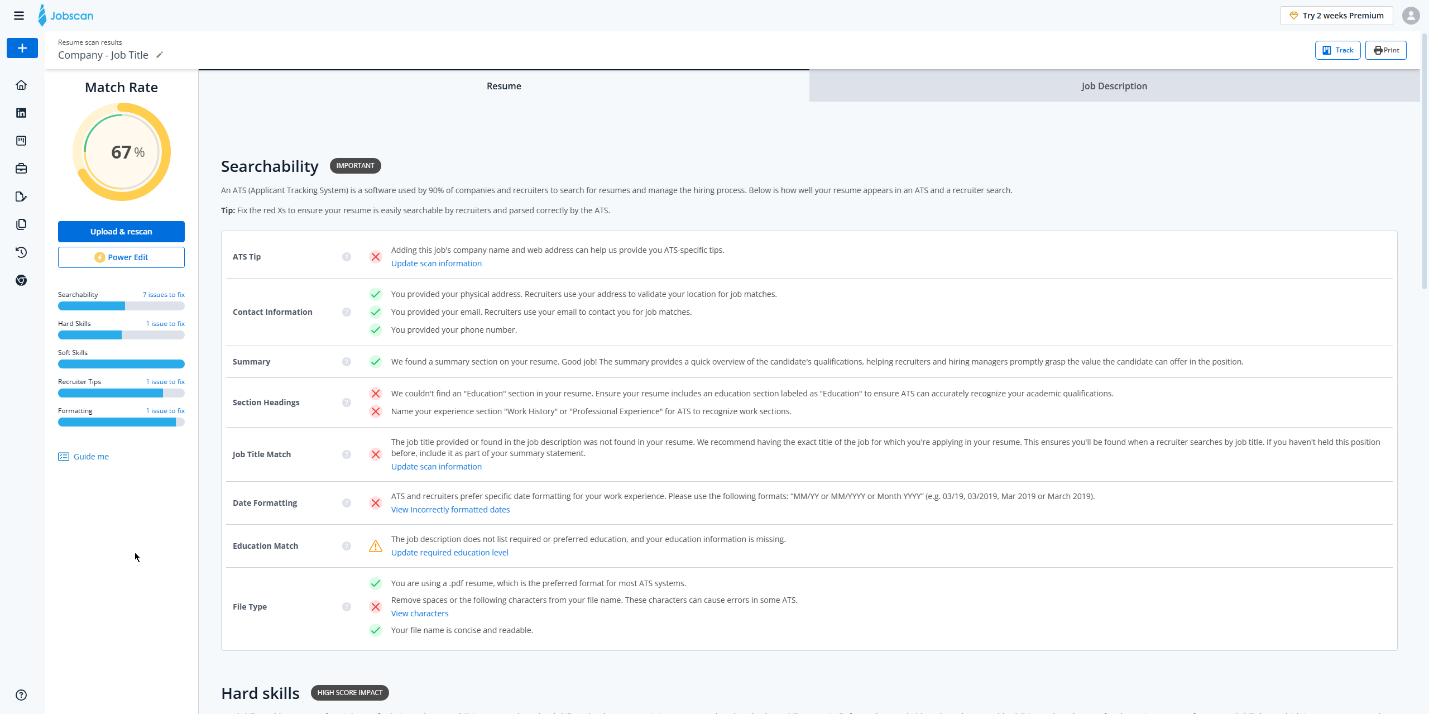
Платформа также включает в себя инструменты для отслеживания вакансий, планирования карьерных целей и создания индивидуальных стратегий для поиска работы. Одной из полезных функций является возможность получения анализа резюме по результатам его сравнения с описаниями вакансий, что помогает пользователю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



— Оценка резюме и рекомендации по улучшению в Teal

Jobscan — это инструмент (Рисунок А.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. 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.



— Рекомендации ИИ по улучшению резюме в Jobscan

* 1. Позиционирование проекта

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 и стандартам отрасли.

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

1. Реализация
   1. Средства реализации
      1. Flutter

Для реализации клиентской части мобильного приложения использовались язык программирования Dart версии 3.7 и Flutter SDK версии 3.29. Flutter представляет собой кроссплатформенный фреймворк, позволяющий разрабатывать мобильные, веб- и десктопные приложения с использованием единой кодовой базы. Это позволило ускорить процесс разработки и обеспечить поддержку Android-платформы в рамках данного проекта.

Использование Flutter обеспечивает ряд преимуществ:

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

— Высокая производительность благодаря компиляции в нативный код и использованию графического движка Skia, рендерящего интерфейс с аппаратным ускорением;

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

Таким образом, выбранный стек технологий позволил быстро развернуть клиентскую часть и реализовать ключевые сценарии взаимодействия пользователя с системой

* + 1. Go

Для разработки серверной части приложения используется язык программирования Go версии 1.24. Go (или Golang) был выбран благодаря своей высокой производительности, встроенной поддержке параллелизма и лаконичному синтаксису, что делает его оптимальным решением для построения высоконагруженных сервисов.

Go предоставляет встроенные механизмы работы с конкурентностью (goroutines), которые позволяют эффективно обрабатывать множество одновременных запросов без существенных затрат системных ресурсов. Это особенно важно для приложений, взаимодействующих с большим количеством пользователей или выполняющих интенсивную сетевую активность.

* + 1. PostgreSQL

Для хранения данных в проекте используется реляционная система управления базами данных PostgreSQL версии 17.4. Данная СУБД выбрана благодаря своей надёжности, производительности и активной поддержке со стороны сообщества. PostgreSQL обеспечивает устойчивую работу при высоких нагрузках и подходит для проектов с активным ростом пользовательской базы.

Система поддерживает такие механизмы, как репликация и шардирование, что даёт возможность масштабировать инфраструктуру без необходимости перехода на другие технологии хранения данных. Это особенно актуально при расширении функциональности приложения и увеличении количества пользователей в будущем.

* + 1. Docker

Для развёртывания компонентов приложения используется система контейнеризации Docker версии TBD. Docker позволяет упаковывать приложение и его зависимости в изолированные контейнеры, что обеспечивает высокую 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, который позволяет описывать конфигурацию сервисов и их взаимодействие в декларативной форме, обеспечивая удобство масштабирования и поддержки инфраструктуры.

TBD (наверняка еще нужно дополнить)

* 1. Архитектура

TBD (про архитектуру, описать диаграммы)

Архитектура системы построена с учётом разделения ролей и упрощения взаимодействия между пользователями и сервисом. Основной акцент сделан на автоматическом анализе загружаемых резюме и генерации рекомендаций по их улучшению.

Система включает два типа пользователей:

Соискатель, загружающий своё резюме для анализа и получения персонализированных рекомендаций.

Администратор, отслеживающий ключевые метрики системы с целью мониторинга её стабильности и эффективности.

Взаимодействие между компонентами организовано по принципу централизованной обработки. Все действия пользователей проходят через центральную систему, которая отвечает за приём, обработку и анализ данных. Система формирует обратную связь для соискателя в виде рекомендаций по улучшению структуры и содержания резюме. Параллельно она собирает данные о своей работе и предоставляет статистику администратору для контроля за качеством и производительностью.

Диаграмма архитектуры на уровне контекста демонстрирует эти взаимодействия и роль каждого участника. Она иллюстрирует, как пользователи обмениваются данными с системой и какую функциональную нагрузку система выполняет в ответ.

TBD (добавить диаграмму)

* 1. Серверная часть

TBD (расписать вообще про сервисы, что как и с чем взаимодействует и каким образом, про хранение данных, развертывание и т.п.)

* 1. Клиентская часть

TBD (про структуру приложения и логику работы)

* 1. Реализация функциональных возможностей системы
     1. Роли пользователей в системе

В системе реализовано три основные роли пользователей, каждая из которых имеет различный уровень доступа и набор доступных функций:

* Неавторизованный пользователь — гость системы, не прошедший процесс авторизации.
* Авторизованный пользователь — зарегистрированный пользователь прошедший процесс авторизации.
* Администратор системы — авторизованный пользователь с расширенными правами.
  + 1. Функциональные возможности
    2. Погрешности

TBD (больше про тестирование, погрешности там и т.п.)

* 1. Реализация интерфейсов системы
     1. Экраны неавторизованного пользователя
     2. Экраны авторизованного пользователя
     3. Экраны администратора системы

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 TBD:

— 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;

— Выполнена интеграция клиентской и серверной частей приложения.

В процессе реализации были использованы инструменты командной разработки, сформирована архитектура системы, обеспечена корректная работа основных компонентов, а также проведено юнит-тестирование.

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

Список использованных источников

TBD (дополнить + отформатировать)

Официальная документация языка программирования Dart версии 3.7;

Официальная документация Flutter SDK версии 3.29;

Официальная документация языка программирования Go версии 1.24;

Официальная документация СУБД PostgreSQL версии 17.4.

<https://www.visualcv.com/ru/>

<https://www.jobscan.co/>

<https://www.tealhq.com/tools/resume-builder>

вигерс разработка требований к программному обеспечению (книга)

#### TBD