

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»
Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

**ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД
ГЭК**

Руководитель образовательной
программы
09.03.04 Программная инженерия
_____ И. Н. Обабков

(подпись) (Ф.И.О.)
«01» _____ июня 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
Пояснительная записка

ТЕМА

**Формирование требований для автоматизации процесса по
актуализации заработной платы сотрудников для компании «Artsoft»**

Руководитель: Д.Б. Шадрин _____

Ст. Преподаватель _____

Нормоконтролер: В. Н. Васина _____

Обучающийся группы РИ-400015 Д. А. Старцев _____

Екатеринбург 2023

ЗАДАНИЕ

РЕФЕРАТ

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1 Анализ предметной области	8
1.1 Проблема заказчика	8
1.2 Анализ процесса формирования главного отчета	10
1.3 Анализ конкурентов.....	14
1.4 Календарный план	16
2. Анализ средств разработки	17
2.1 Архитектура приложения.....	17
2.2 База данных	19
2.3 Язык программирования	20
2.4 Выбор лингвистической модели	23
3. Техническое задание.....	25
4. Проектирование	26
4.1 Обновленный процесс сбора наглядной аналитики.....	26
4.2 User–flow	27
4.3 Проектирование базы данных сервиса «lk-user»	30
4.4 Проектирование клиент-серверного взаимодействия сервиса «lk-user»	31
4.5 Проектирование архитектуры «lk-user»	32
5. Реализация	35
5.1 Реализация клиентской части приложения	35
5.2 Реализация серверной части приложения	38
ЗАКЛЮЧЕНИЕ	40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	41
ПРИЛОЖЕНИЕ А (обязательное) Схема базы данных сервиса ЛК	43

ПРИЛОЖЕНИЕ Б (обязательное) Схема базы данных сервиса аналитики	44
ПРИЛОЖЕНИЕ В (обязательное) Техническое задание проекта	45
ПРИЛОЖЕНИЕ Г (обязательное) Содержание xlsx файла (отчета)	58
ПРИЛОЖЕНИЕ Д (обязательное) Аналитический запрос.....	62
ПРИЛОЖЕНИЕ Е (обязательное) Сущность аналитического запроса.	63
ПРИЛОЖЕНИЕ Ж (обязательное) Актуализация документов в кодовом представлении.....	64

ВВЕДЕНИЕ

В современном мире цифровизация и автоматизация проникают во все сферы деятельности компаний и повседневной жизни человека. Если раньше технологии касались только технического отдела, то сегодня все нуждается в автоматизации процессов для упрощения работы и повышения ее эффективности. В том числе профессиональная HR-деятельность, без которой не обойдется ни одна компания.

Часто компании прибегают к созданию HRM (Human Resources Management) отдела, который занимается стратегией обращения с группами сотрудников. Решения HRM должны основываться на определенных данных, сбор которых представляет собой большую работу, которая занимает много ресурсов.

В компании Artsofte В компании работает более 450 человек и оборот кадров всегда является острым вопросом для HR отдела.

Чтобы быть уверенными в своей позиции на рынке кадров: каждый квартал происходит анализ предложения на самом известном сайте по подбору вакансий — hh.ru.

Сбор и систематизация информации о вакансиях на популярном ресурсе hh.ru позволяет не только создать обширную базу данных, но и выделить ключевые тенденции в требованиях работодателей и ожиданиях соискателей. Это обеспечит эффективную адаптацию образовательных программ, обучения кадров и стратегий подбора персонала, что является актуальной задачей в условиях постоянно меняющегося рынка труда.

Чтобы уменьшить нагрузку на HR отдел и эффективно распределить его ресурс – было принято решение разработать сервис для автоматизации процесса актуализации заработной платы сотрудников.

Тем самым целью работы является аналитика процессов организации Artsofte и формирование требований к сервису автоматического анализа рынка – HRParser.

Для достижения цели поставлены следующие задачи:

- 1) изучение процессов организации встречи в «Artsofte»;
- 2) выявление и описание требований к системе у заказчика;
- 3) описание функциональных требований к системе;
- 4) описание сценариев;
- 5) передача технического задания в разработку.

Объектом исследования является процесс актуализации заработной платы сотрудников и возможности его автоматизации.

1 Анализ предметной области

1.1 Проблема заказчика

Перед тем как начать создавать продукт, необходимо провести анализ предметной области. Это включает в себя изучение бизнес-процессов компании, инструментов разработки компании, выявление и изучение проблем, общение с заказчиком, формулирование и проверку гипотез, а также разработку пользовательских историй для лучшего понимания опыта пользователя.

Эта часть работы будет фокусироваться на поиске и изучении проблемы, общении с потенциальными пользователями, и сборе требований. Затем на основе собранной информации будет составлено техническое задание, которое станет основой для разработки архитектуры, выбора инструментов и технологий разработки.

Существует процесс анализа рынка труда с последующей актуализацией заработной платы сотрудников.

Сама актуализация происходит по множественным причинам:

а) актуализация заработной платы позволяет компании прогнозировать будущие расходы на оплату труда. Это важно для предотвращения неожиданных финансовых нагрузок;

б) регулярная актуализация заработной платы позволяет сотрудникам видеть, что их труд и профессиональное развитие оцениваются. Это способствует повышению мотивации, уровня удовлетворенности и приверженности компании;

в) анализ рынка показывает актуальность вакансии: соотношение количества вакансий к заработной плате говорит о востребованности;

г) знание актуальной цены кадра позволяет компании оценить свою конкурентоспособность на рынке труда. А также сравнивать условия труда.

На рисунке 1 приведены главные задачи работодателей в условиях кризиса. График составлен на основе компиляции данных из исследования «TalentTech» в 2022 [1].

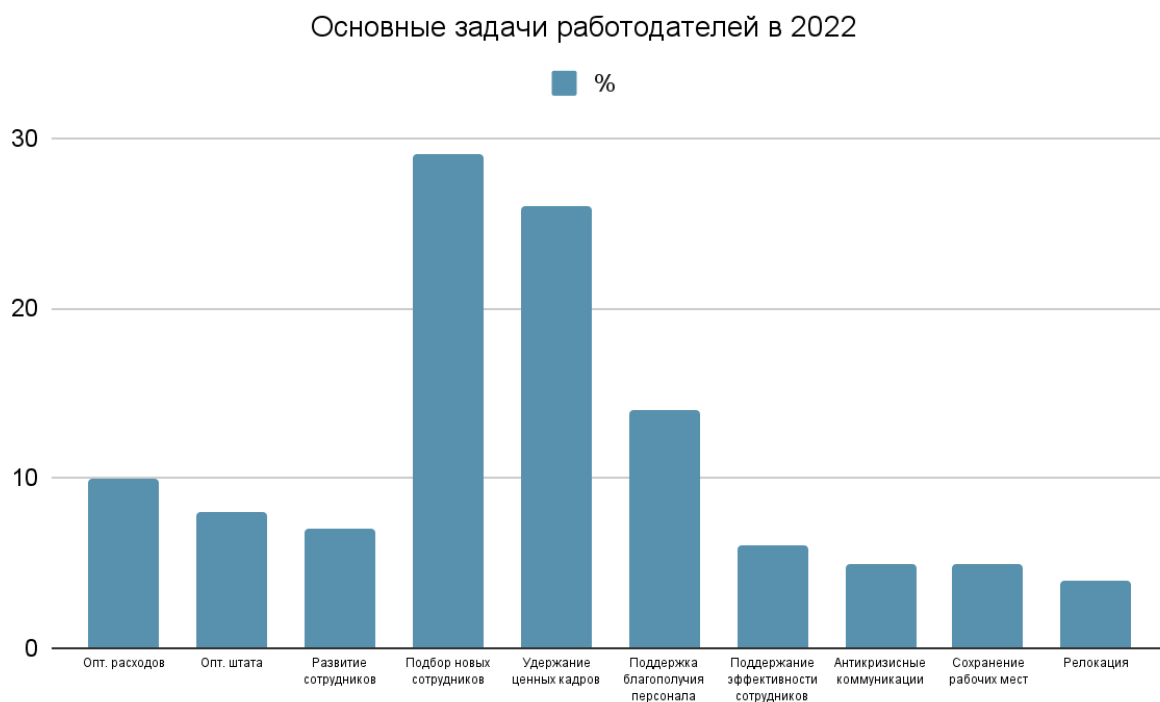


Рисунок 1 – Главные задачи работодателей в условиях кризиса по исследованию «TalentTech» в 2022 г.

Относительно этого исследования актуализация заработной платы связана со следующими пунктами:

- а) оптимизация расходов;
- б) оптимизация штата;
- в) подбор новых сотрудников;
- г) удержание ценных кадров;
- д) поддержание эффективности сотрудников;
- е) сохранение рабочих мест.

То есть целевой бизнес–процесс покрывает около 83% задач работодателя в условиях кризиса. Тогда следует чуть ли не обязательное наличие этого процесса в любой IT компании современного времени.

Совсем недавно рост онлайн–активности принес технологическим компаниям исторически высокие прибыли и вызвал бурное увлечение наймом сотрудников, чтобы удовлетворить растущий спрос. В этот период технологические компании предполагали, что такой взлет станет новым стандартом, что привело к активному расширению команд и быстрому увеличению размеров компаний. По статистике из исследования M.Tech [2] За время пандемии Amazon увеличил свою рабочую силу на 93%, Microsoft — на 53%, Meta — на 92%, Apple — на 20%, а Alphabet — на 60%.

У заказчика «Artsofte» имеется 120 ролей и более 450 сотрудников. Здесь под «ролью» понимается функциональная позиция в организации с определенным уровнем квалификации. Текущий бизнес–процесс формирования отчета об актуальности зарплат очень затратный по ресурсам компании. Это тратит время квалифицированного сотрудника. Приходится переносить сроки выполнения остальных задач, в связи с этим. Также если ускорить весь процесс, то компания будет иметь возможность чаще актуализировать заработную плату сотрудникам. Сейчас это происходит раз в полугодие.

Чтобы решить эту проблему нужно найти участок бизнес–процесса, который является самым трудозатратным и автоматизировать работу на этом участке.

1.2 Анализ процесса формирования главного отчета

Удобно будет описать процесс формирования итогового отчета с результирующей заработной платой через Customer Touchpoint Map (рисунок 2) – это визуальное представление всех важных моментов взаимодействия клиента с брендом или продуктом на протяжении всего пути от первого знакомства до последующих этапов, целью которой является оптимизация клиентского опыта.

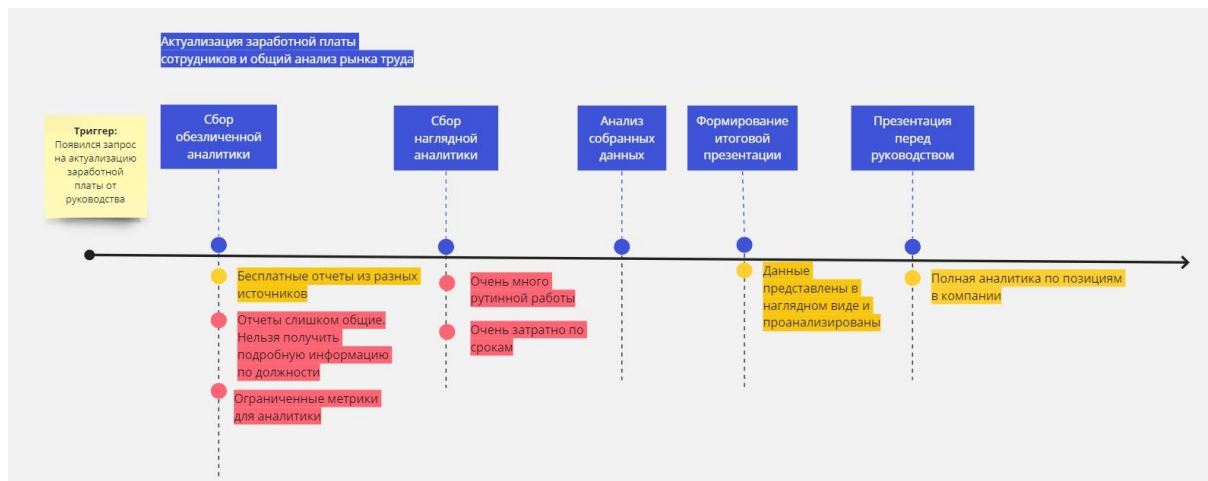


Рисунок 2 – Customer Touchpoint Map процесса создания итогового отчета

Процесс делится на 5 составляющих:

- 1) сбор обезличенной аналитики;
- 2) сбор наглядной аналитики;
- 3) анализ собранных данных;
- 4) формирование итоговой презентации;
- 5) презентация перед руководством.

Сбор обезличенной аналитики – на этом этапе собираются данные по зарплатам из открытых источников. Данные в таких метриках уже структурированы. На данный момент лучшими из бесплатных решений являются:

- <https://career.habr.com/salaries>,
- <https://rit.work/jobs>.

Отчеты со сторонних источников, хотя и предоставляют обширные обзоры, вносят свои собственные ограничения в виде ограниченного объема данных и отсутствия возможности проверки их достоверности. Несмотря на эти ограничения, собранный датасет играет важную роль в процессе валидации аналитики на следующем этапе исследования.

На основе анализа этих обезличенных данных мы можем выделить ключевые тенденции и закономерности, которые будут служить основой для более глубоких исследований на следующих этапах. Важно подчеркнуть, что,

несмотря на отсутствие возможности проверки конкретных вакансий, полученные данные предоставляют ценный обзор зарплатного уровня в индустрии и являются неотъемлемой частью проверочного этапа нашего исследования.

Стоит отдельно отметить, что hr не контролирует метрики, которые ему предоставляют сторонние ресурсы по анализу рынка, в связи с чем тратятся дополнительные ресурсы на анализ полученных датасетов.

Сбор наглядной аналитики – на этом этапе формируется отчет похожий на тот, что на предыдущем шаге, однако в этом отчете мы можем сами настроить нужные нам метрики, брать только нужные нам данные. Отчет формируется вручную по каждой роли.

По схеме процесса «Сбор данных для наглядной аналитики» (рисунок 3) можно понять, что работа распараллеливается на нескольких человек по причине большого объема ручной работы.

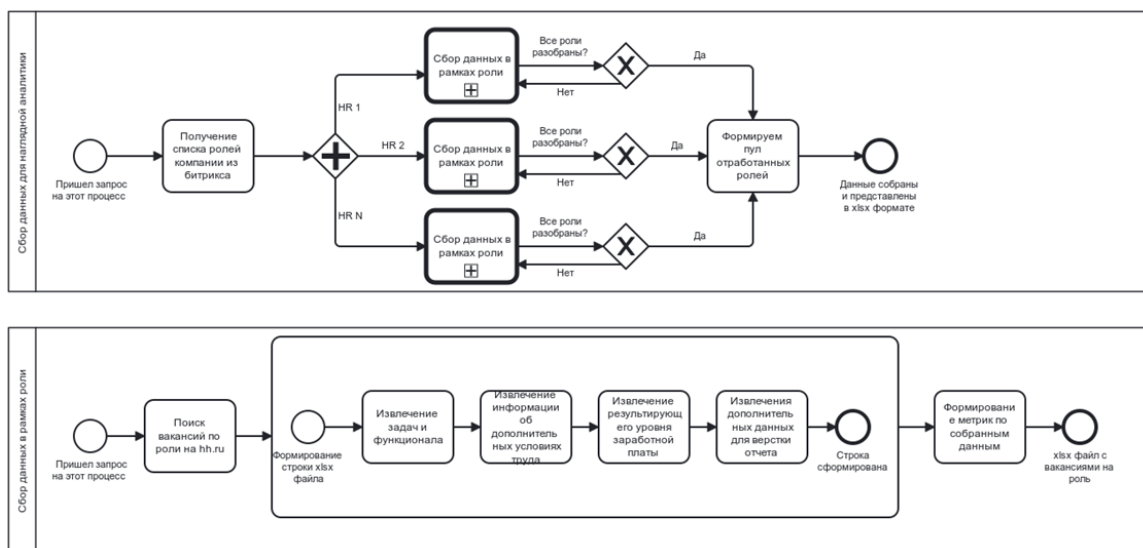


Рисунок 3 – BPMN диаграмма процесса «Сбор наглядной аналитики»

Работа каждого юнита (схема Сбор данных в рамках роли) заключается в том, что он заходит на сайт hh.ru, начинает читать вакансии и механически переносить данные со страницы вакансий в excel таблицу – построчно формируя отчет по одной роли. На это уходит большинство времени.

Как можно заметить по рисунку 4 – сбор данных для этапа «Наглядная аналитика #2» занимает около 74% от суммарного времени всех этапов. Именно здесь происходит «утечка» времени. Следовательно, именно этот этап и стоит подвергать автоматизации.

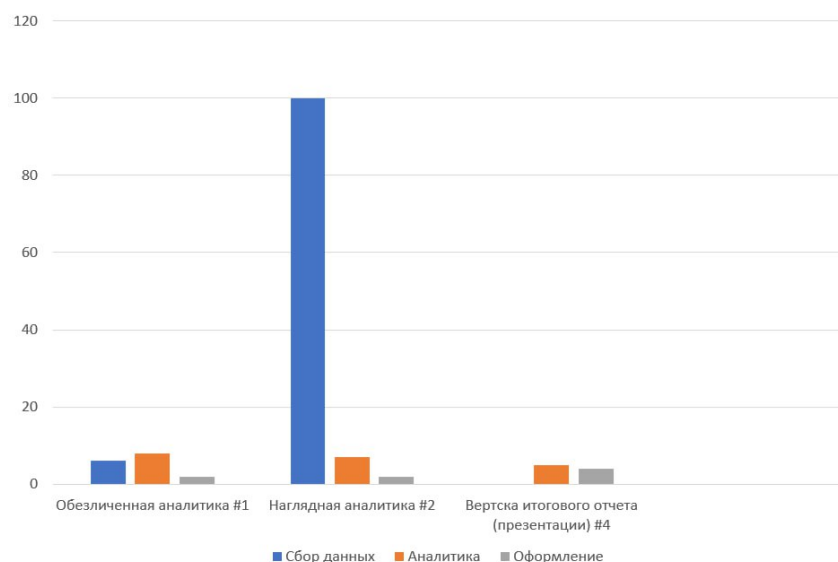


Рисунок 4 – Распределение времени среди трёх этапов создания итогового отчёта

Анализ собранных данных представляет собой выявление необходимых для итогового отчета данных:

- а) зарплатные ожидания;
- б) анализ уровня конкурентоспособности;
- в) тенденции рынка;
- г) адаптация стратегии найма;
- д) оптимизация зарплат внутри компании.

Формирование итоговой презентации – здесь данные, полученные на предыдущем этапе, представляются в упорядоченном виде для упрощения восприятия аналитических данных. Здесь происходит верстка презентации, размещение смысловых акцентов. Этот этап нужен для подведения итогов аналитической работы.

Презентация перед руководством – на этом этапе отчет презентуют перед руководством, которое будет принимать решение по оптимизации бюджета на следующее полугодие.

Таким образом, для автоматизации процесса актуализации заработной платы сотрудников необходимо: разработать систему, которая автоматически собирает данные с hh.ru и формирует отчет по запросу пользователя. Тем самым сократив время всего процесса примерно на две трети.

1.3 Анализ конкурентов

Главным преимуществом нашего продукта по сравнению с конкурентами следует считать возможность формировать специфические метрики. Далее приводится список ресурсов, которые уже используются для анализа рынка. Также приводится список их преимуществ и недостатков.

- а) Habr Карьера – электронный ресурс [3];
- б) HeadHunter – электронный ресурс [4];
- в) Rit.work – электронный ресурс [5];
- г) Роснавык – электронный ресурс [6];
- д) Зарплатомер (SuperJob) – электронный ресурс [7].

Система получения данных на Habr–карьера заключается в следующем. Пользователь, который хочет узнать ситуацию на рынке труда – должен указать свой уровень заработной платы и должность (профессию). Таким образом пользователь попадает в общий набор данных, который и предоставляется к анализу на этом портале.

То есть – ничто не мешает пользователю внести случайные данные, проверить это не выйдет, т.к. хабр предоставляет только итоговые данные, которые являются обезличенными.

Итоговая информация представляет собой набор квалификаций (Lead, Senior, Middle, Junior, Intern) – и соответствующее распределение зарплат.

Также имеется возможность фильтрации по местоположению, специализации, профессиональным навыкам, компаниям, типу занятости и формату работы.

HeadHunter – самый популярный сайт для работодателей и соискателей в СНГ. Источником данных являются реальные заказчики рабочей силы. Каждую вакансию можно проверить на подлинность.

Итоговой информацией здесь является набор вакансий, которые можно свободно фильтровать по различным критериям.

Его недостаток лишь в том, что данные представлены как есть, то есть в слабо упорядоченном виде и наложить дополнительный фильтр на данные или собрать метрику (например медиану по выборке) – нельзя. Отчего нужно создавать свои инструменты анализа по данным, которые HeadHunter предоставляет.

Rit.work – это HeadHunter, но только для IT-вакансий. Здесь своя база данных, преимущество над HeadHunter`ом в том, что помимо тех же фильтров, что и на вышеупомянутой платформе, есть вывод метрики по отфильтрованным данным (перцентили, график – зарплата по месяцам, график – количество/зарплата).

Количество вакансий по сравнению с ХХ заметно меньше и обновляются они менее динамично, чем на самом популярном поисковике СНГ.

Роснавык обладает хорошей визуализацией данных и удобными отчетами. Однако нельзя «провалиться» в вакансию и узнать её источник. Ограниченное количество фильтров для одного запроса.

Также есть платные аналитические отчеты по согласованному списку профессиональных областей (до 4ех штук) по 50000 рублей за одну штуку.

Источники информации: Работа России, HeadHunter, SuperJob, Работа.ру. Проверки на дубли не ведется, отсюда можно получить погрешности при анализе данных.

SuperJob – удобные полезные отчеты по запрошенной профессии. Обезличенные данные. Высокая цена.

Результаты данного анализа можно представить в виде таблицы 1, где Парсер – название нашего сервиса.

Таблица 1 – Сравнительный анализ конкурентов

Сервис	Обезличенность	Стоимость	Метрики	Наполнение
Habr Карьера	1	5	3	2
HeadHunter	5	5	1	5
Rit.work	5	5	3	2
Роснавук	1	3	5	3
SuperJob	1	1	4	5
Парсер	5	4	5	5

1.4 Календарный план

Основные этапы практики приведены в таблице 2.

Таблица 2 – Этапы практики

Этап практики	Дата
Анализ бизнес–процессов компании	17.02 – 24.02
Анализ требований к системе	25.02 – 29.02
Описание технического задания и построение архитектуры системы	29.02 – 24.03
Предлагаемое техническое решение	24.03 – 05.04

2. Анализ средств разработки

2.1 Архитектура приложения

Программный продукт представляет собой два сервиса, связанных друг с другом технологией REST. Это означает, что каждый запрос от клиента к серверу должен содержать всю необходимую информацию, без использования данных о предыдущих запросах.

На рисунке 5 изображена схема двух сервисов, которые являют собой систему обработки вакансий с hh.ru и обеспечивают вывод информации в виде xlsx документов.

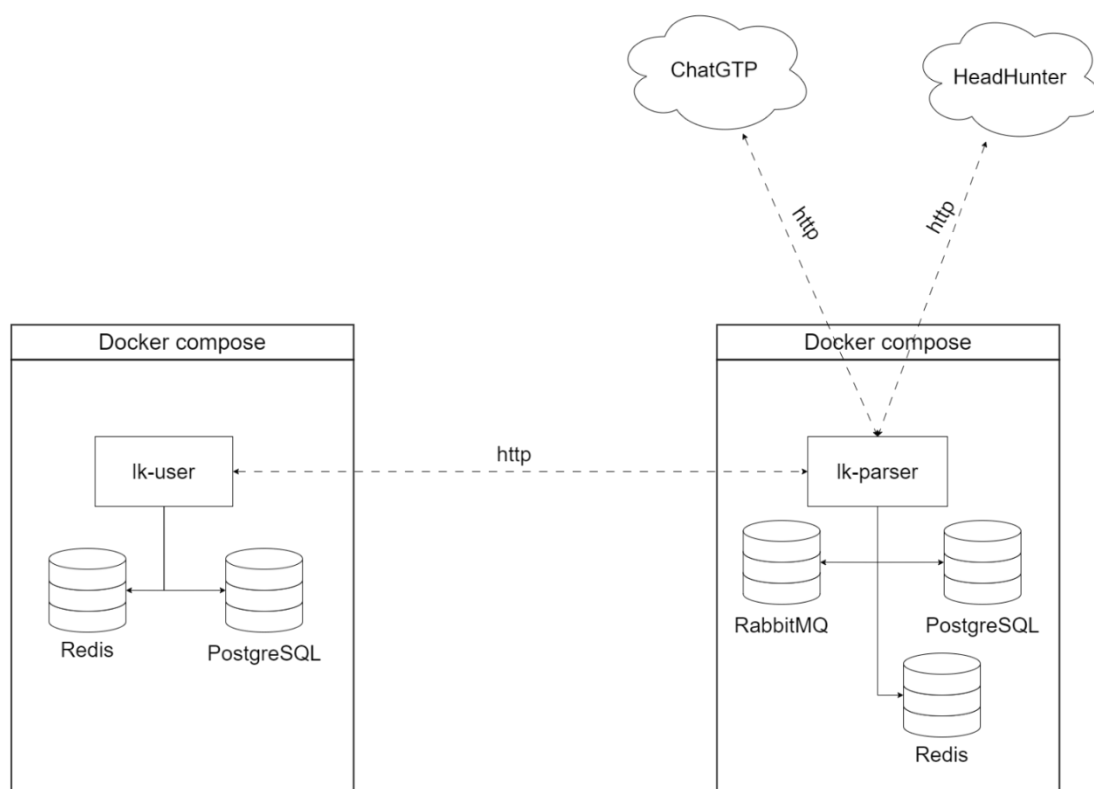


Рисунок 5 – Архитектура взаимодействия сервисов

В рамках данного продукта моя зона ответственности – клиентская сторона личного кабинета пользователя (lk-user). Иными словами – верстка шаблонов, дизайн интерфейсов, добавление динамики при помощи средств

JavaScript, внедрение JavaScript файлов в монолит при помощи фреймворка Webpack для соблюдения принципов чистого кода. В этом сервисе не представлено никакой сложной логики. Просто набор интерфейсов и база данных для хранения сопутствующих сущностей.

Второй сервис (сервис аналитики – lk-parser) – представляет собой парсер, генератор отчетов, калькулятор аналитической информации, прокси запросов к лингвистической модели и также хранилище данных для аналитики.

Взаимодействие между сервисами осуществляется посредством протокола HTTP, где необходимая информация передается в формате JSON, что соответствует принципам архитектуры REST. Каждый сервис оперирует собственным набором баз данных, обеспечивая тем самым высокую степень автономности. Это означает, что в случае неполадок или отказа одного из сервисов, другие могут продолжать свою работу без существенного влияния на общую функциональность системы.

Redis, представляющая собой NoSQL базу данных, служит для эффективного управления кэшем данных в распределенных средах. Она обеспечивает высокую производительность за счет хранения данных в оперативной памяти и использования сложных алгоритмов кэширования. Это позволяет уменьшить нагрузку на основные базы данных и сократить временные задержки при обращении к данным.

PostgreSQL, в свою очередь, является реляционной базой данных, предназначенной для долгосрочного и структурированного хранения данных. Ее гибкий и мощный SQL-движок обеспечивает эффективное хранение и обработку данных в различных сценариях, от малых приложений до крупных корпоративных систем.

RabbitMQ, являясь брокером сообщений, выполняет роль посредника между различными компонентами системы, обеспечивая надежную и отказоустойчивую передачу сообщений. Его механизмы хранения сообщений гарантируют сохранность данных даже в случае сбоев или отказов в работе

сервисов, обеспечивая непрерывность работы системы и сохранение целостности данных. Каждый сервис находится в изолированном Docker контейнере, что позволяет обеспечить максимальную безопасность и устойчивость системы. Контейнеризация с помощью Docker обеспечивает изоляцию каждого сервиса, предотвращая взаимное воздействие между ними и минимизируя возможность уязвимостей. Кроме того, такой подход упрощает развертывание, масштабирование и управление сервисами, делая процесс более эффективным и прозрачным.

2.2 База данных

Для хранения полезных данных была предложена следующая схема базы данных сервиса личных кабинетов (рисунок А.1).

База данных сервиса личного кабинета состоит из 6 сущностей:

- а) пользователь – чтобы обеспечивать ограниченный доступ к ресурсу;
- б) миграции – для удобства развертывания приложения и актуализации базы данных при разработке;
- в) hh–регион – регионы хранятся в базе данных чтобы не запрашивать их с hh при каждом формировании аналитического запроса;
- г) hh–индустрия – аналогично региону;
- д) hh–подиндустрия – аналогично региону, связана с hh–индустрией первичным ключом;
- е) аналитический запрос – аналитический запрос хранится на стороне личного кабинета чтобы можно было переотправлять запросы при неудачном выполнении.

На рисунке Б.1 представлена база данных сервиса lk–parser, ниже представлено её описание.

База данных сервиса аналитики опирается на 5 сущностей (плюс 3 вспомогательных).

Сущность вакансии (vacancy) – аналитическая единица отчета, именно эта сущность обогащается данными при парсинге и дополнительной обработке.

Также в отдельную таблицу (order_excel_file_param) записываются параметры формируемых файлов для более удобного обращения к ним при скачивании.

Formula_setting и calculation_results служат для сохранения применяемых к вакансии формул при обработке.

Также существует таблица, дублирующая таблицу из сервиса личного кабинета – parser_order. Это сделано для репликации данных, чтобы в случае чего не было потерь важной информации.

2.3 Язык программирования

Было принято решение разрабатывать сервис личных кабинетов связкой языков JavaScript и PHP. В частности, фреймворков jQuery, Symfony.

Выбор использования связки JavaScript (jQuery) + PHP Symfony с интеграцией Webpack и Bootstrap обусловлен несколькими ключевыми факторами, которые обеспечивают эффективное производство и достижение целей сервиса:

Хотя jQuery многими считается устаревшей технологией – она не теряет своих многочисленных плюсов, которые перечислены в официальной документации [8]:

а) простота использования – jQuery предлагает простой синтаксис, что делает работу с DOM и событиями веб-страницы более интуитивной;

б) кроссбраузерная совместимость обеспечивает однородное поведение на различных браузерах, что делает выполнение кода стабильным;

в) богатая библиотека функций предлагает широкий спектр функций и плагинов для анимации, AJAX запросов, управления CSS и других задач, упрощая разработку;

г) эффективность позволяет создавать компактный и эффективный код за счёт методов выбора элементов DOM и массовой манипуляции элементами;

д) активное сообщество и поддержка – jQuery имеет огромное сообщество разработчиков и обширную документацию, обеспечивая доступ к множеству ресурсов для обучения и развития.

Такой же участи удостоен PHP, как язык, которого принято считать устаревшим, однако нужно просто правильно его использовать, тогда его недостатки становятся его плюсами. Вот, что пишут про этот язык сами его создатели в документации [9]:

а) PHP легко изучить и использовать для создания веб-приложений;

б) поддержка различных баз данных и богатый функционал делают PHP мощным инструментом для разработки;

в) PHP успешно применяется в создании крупных и масштабируемых веб-приложений;

г) большое сообщество и ресурсы PHP обеспечивает доступ к поддержке и развитию.

Тогда стоит приступить к обсуждению взаимодействия связки этих технологий.

Производительность и нагрузка: Учитывая, что сервис не испытывает высокой нагрузки и выполняет относительно простые задачи, выбор легковесных технологий таких как JavaScript (jQuery) и PHP Symfony позволяет достичь необходимого уровня производительности без излишней сложности и затрат.

Экономичность: Использование открытых и бесплатных технологий, таких как JavaScript (jQuery) и PHP Symfony, а также фреймворка Bootstrap, помогает сократить расходы на лицензирование и разработку. Это особенно важно, учитывая дешевизну производства.

Удобство разработки и поддержки: Symfony предоставляет мощные инструменты для разработки веб-приложений, обеспечивая высокую производительность и удобство в работе с базой данных и другими аспектами

приложения. Следуя официальной документации Webpack [10], интеграция с Webpack позволяет управлять зависимостями JavaScript и оптимизировать их загрузку, а Bootstrap, судя по документации [11], предоставляет готовые компоненты и стили для быстрой разработки пользовательского интерфейса.

Выбранный стек технологий обеспечивает достаточную гибкость и масштабируемость для возможного расширения функциональности и увеличения нагрузки в будущем

Таким образом, использование связки JavaScript (jQuery) + PHP Symfony с интеграцией Webpack и Bootstrap обеспечивает оптимальное сочетание экономичности, производительности, удобства разработки и гибкости, соответствуя потребностям и возможностям нашего сервиса.

Сервис «lk-parser» - является чисто backend приложением, для реализации которого была выбрана связка spring framework с java.

Так как проект может располагаться на сервере и должен быть доступен удаленно, было выбрано четыре языка программирования для сравнения: PHP (Symfony), JavaScript (Node), Go, Java (Spring), производительности языков программирования в веб-апп задачах основаны на сравнении из статьи «Производительность I/O бэкэнда: Node vs. PHP vs. Java vs. Go» [12].

Для визуализации данных, ниже приведена таблица 3 – таблица сравнения этих языков.

Таблица 3 – Сравнение языков программирования

Критерий	PHP	Java	Go	JavaScript
Производительность	3	4	5	4
Экосистема	4	5	3	4
Асинхронное программирование	0	1	1	1
Сложность изучения	5	2	1	5
Масштабируемость	3	4	5	4
Сообщество	5	5	2	3
Поддержка	3	5	5	4
Итог	23	26	22	25

2.4 Выбор лингвистической модели

Важной частью автоматизации является нейросеть по вычленению слов по заданному контексту. Есть много различных решений этой конкретной задачи. Нужно выбрать ту, которая устроила бы заказчика и была легко интегрируема в наш сервис.

Есть два подхода к реализации работы нейросети в сервисе:

- облачное решение,
- компиляция кода нейросети.

Компилирование нейросети ведет к высокой ресурсной нагрузке. Более подробно это рассмотрено в одноименной статье [13], судя по ней – вычисления происходят на графическом ядре. Отсюда следуют высокие расходы на инфраструктуру проекта. (В сравнении со стоимостью облачного решения)

Помимо поддержки инфраструктуры, для скомпилированной нейросети нужно иметь большой датасет для тренировки этой самой модели.

количество примеров в датасете должно быть около 50тыс единиц для более–менее валидной работы этой самой нейросети. Чем больше – тем лучше. Обработка одной вакансии занимает 1–2 минуты. Таким образом сбор датасета займет около 1250 часов.

Также, натренированная нейросеть не учитывает свободный словесный запрос по типу: «если в твоей выборке будет вакансия на тимлида, то не включай эту вакансию в выборку, просто исключи её.»

Нейросеть, которая не является большой лингвистической моделью – не способна воспринимать такие просьбы пользователя.

Также использование ChatGPT обусловлено дешевизной разработки, то есть это уже готовое решение, которое свободно интегрируется посредством открытого API, которую OpenAI предоставляет свободно, судя по документации [14].

Т.к. прямые запросы в ChatGPT из России невозможны – нужно использовать проху для запросов. Таким образом выбор пал на neugoarі, где стоимость запроса в 500 токенов (примерно столько занимает запрос одной вакансии) равна 0,02 рубля, исходя из их прейскуранта [15]. То есть при среднем объеме отчета в 600–1000 вакансий (запросов) и 120 должностях. Получаем расход, соответствующий диапазону $600 * 120 * 0.02 \leq x \leq 1000 * 120 * 0.02$ или $1440 \leq x \leq 2400$ рублей за одну реализацию процесса по актуализации заработной платы сотрудников «Artsoftе»

Сервер с видеокартой для реализации LLM нейросети (или же упрощенной NLP – Natural Language Processing) стоит в месяц минимум 7500рублей (самый дешевый тариф на hostkey.ru). Есть возможность брать сервера посуточно (около 400 рублей в сутки, что примерно равно использованию прокси), но тогда все запросы нужно будет формировать заранее и исключается возможность «экстренного парсинга», также это накладывает лишние расходы на тестирование и разработку продукта.

3. Техническое задание

Техническое задание представлено в приложении Б.1 и далее. Ключевая часть реализации, которая передана программисту серверной части сервиса «lk-parser» представлена в приложении В.5.

4. Проектирование

4.1 Обновленный процесс сбора наглядной аналитики

Существующий процесс представлен на рисунке 3. Его подробное описание выглядит так: собираем список должностей для анализа. Обычно этих ролей около 1, в связи с этим работа по обработке разделяется на несколько человек. Каждый HR берет из «стопки должностей» одну и формирует по ней xlsx файл: сначала по названию должности подается запрос в hh.ru, чтобы найти актуальные вакансии по этому названию. Каждая вакансия соответствует строке xlsx файла. Сама строка формируется путем сбора следующей информации из описания вакансии:

- 1) извлечение задач и функционала;
- 2) извлечение информации о дополнительных условиях труда;
- 3) извлечение результирующего уровня заработной платы,
- 4) извлечение дополнительных данных для верстки отчета.

После того, как по каждой вакансии сформирована строка – xlsx файл считается законченным. И берутся следующие должности до тех пор, пока они не закончатся.

Тогда определим слабые стороны старого процесса:

- а) над обработкой ролей работает несколько человек;
- б) каждая отдельная вакансия обрабатывается в несколько сложных этапов;
- в) основное время уходит на рутинные действия, которые легко автоматизировать.

Следовательно, в новом процессе эти недостатки должны быть исправлены. Тогда новый процесс должен выглядеть как на рисунке 12:

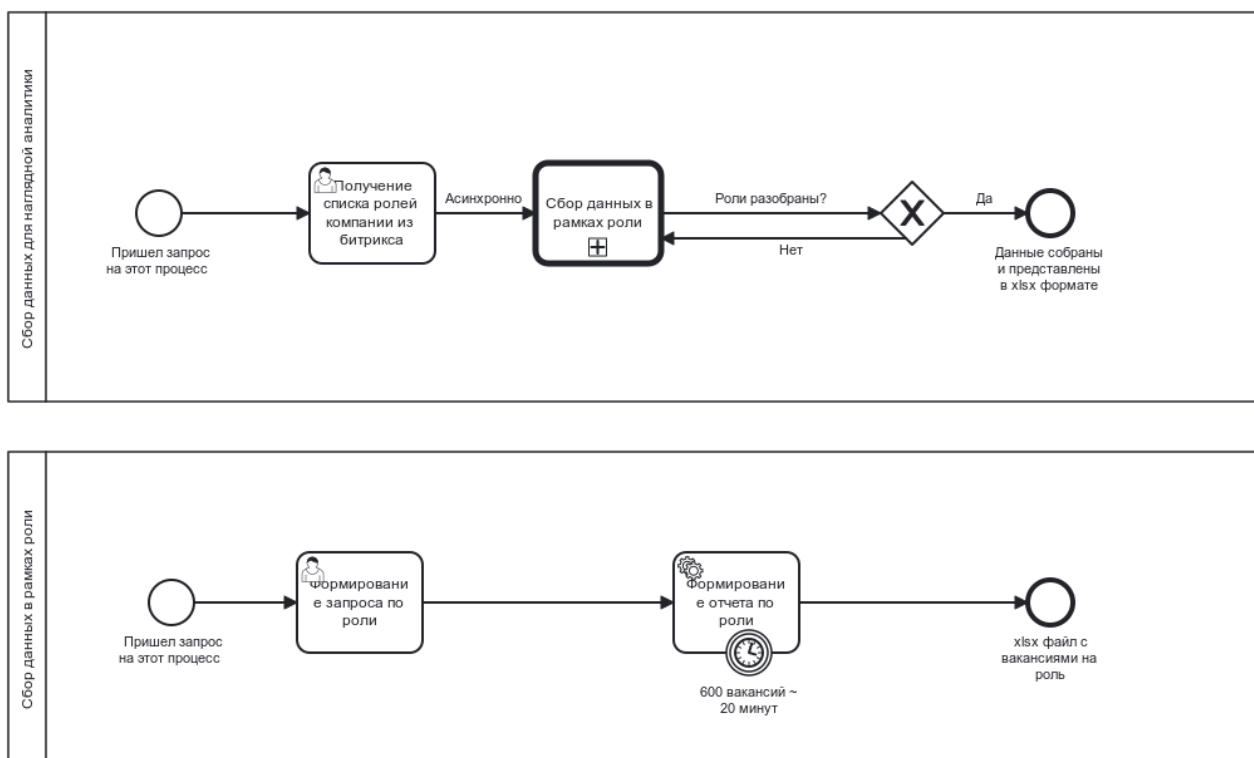


Рисунок 12 – обновленный процесс получения отчета

Теперь процесс выглядит так: получаем выгрузку должностей из битрикса, выполняем запрос на формирования отчета по каждой должности. Далее запросы попадают в сервис формирования отчетов. (Там, по определенному запросу в LLM, автоматизируются ручные действия по вычленению данных из текста.) Ожидаем выполнения всех запросов. Данные собраны и представлены в xlsx формате, содержание отчета представлено в приложении В.1.

4.2 User-flow

Для демонстрации работы с продуктом было принято решение воспользоваться руководством по созданию user-flow [17]. User-flow (поток пользователя) представляет собой последовательность действий, которые пользователь выполняет при взаимодействии с продуктом или сервисом (рисунок 13). Это концептуальная модель, которая описывает путь

пользователя от начальной точки (например, посещение веб–сайта или запуск приложения) до достижения конечной цели (например, совершение покупки, регистрация аккаунта и т. д.).

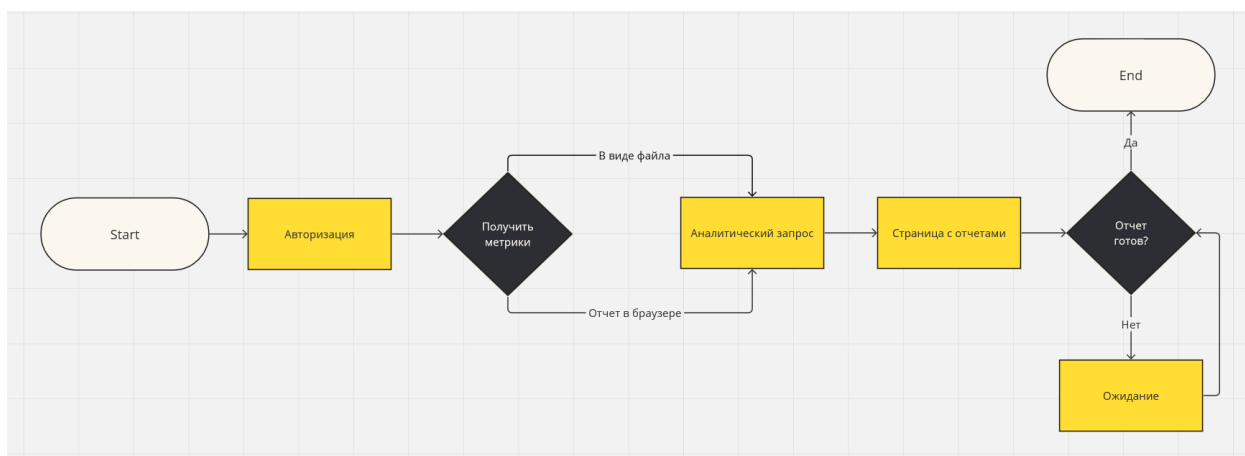


Рисунок 13 – user flow получения отчета

Принцип работы с нашим решением довольно прост и не требует дополнительного обучения персонала: чтобы получить отчет необходимо войти в личный кабинет, используя логин и пароль, далее нужно зайти во вкладку Request (рисунок 14) и заполнить нужные для запроса поля.

Рисунок 14 – форма запроса

Запрос описан в таблице Г.1 Стоит отметить, что в каждом поле с мультиселектом реализован поиск по подстроке и добавлены кнопки для

выбора всех вариантов. Также из-за большого списка индустрий с hh.ru было принято решения добавить кнопку для их сокрытия чтобы не перегружать интерфейс.

После того, как запрос был отправлен на сервер, можно зайти во вкладку Documents (рисунок 15) и увидеть статус документа:

- а) START_PARSING – парсинг начат;
- б) COMPLETE – парсинг окончен;
- в) GENERATE_DOCUMENT – генерация документа;
- г) COMPLETE_WITH_DOCUMENT – генерация документа окончена;
- д) PARSING_ERROR – ошибка при выполнении.

HrParser

Documents

Request

Хэш	Запрос	Ссылка на скачивание	Статус	Дата
dcb240ce	1С	Скачать файл	COMPLETE_WITH_DOCUMENT	2024-04-02 05:41:17
e2c71183	java программист	Скачать файл	COMPLETE_WITH_DOCUMENT	2024-04-01 13:57:11
e2c9e37d	java программист	Скачать файл	COMPLETE_WITH_DOCUMENT	2024-04-01 13:41:58
a08f493a	java программист	Скачать файл	COMPLETE_WITH_DOCUMENT	2024-04-01 13:37:31
0e8232f9	java программист		PARSING_ERROR	2024-04-01 13:35:57
a43142db	java программист	Скачать файл	COMPLETE_WITH_DOCUMENT	2024-04-01 13:32:12
a8ac6dc3	php программист	Скачать файл	COMPLETE_WITH_DOCUMENT	2024-04-01 13:19:48
9ec11585	php программист	Скачать файл	COMPLETE_WITH_DOCUMENT	2024-04-01 13:15:30
b9dade22	php программист	Скачать файл	COMPLETE_WITH_DOCUMENT	2024-03-30 14:09:17
60f68b32	JavaScript разработчик	Скачать файл	COMPLETE_WITH_DOCUMENT	2024-03-29 21:15:53

[« Previous](#) [1](#) [2](#) [Next »](#)

Рисунок 15 – страница со списком отчетов

Также при нажатии на хэш можно посмотреть запрос более детально (рисунок 16), что позволяет получить дополнительную информацию о конкретном запросе или его результате. Это особенно полезно для анализа данных и выявления паттернов или аномалий в информации.

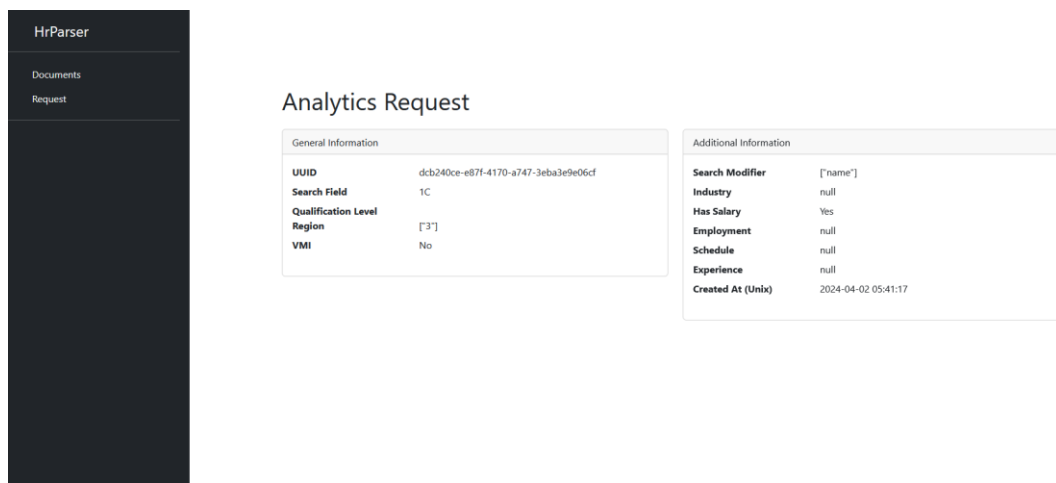


Рисунок 16 – страница с содержанием запроса

Таким образом user-flow укладывается в три интуитивных интерфейса и не требует привлечения дополнительных ресурсов на обучение персонала. Главным преимуществом этого порядка действий является то, что не нужно дожидаться выполнения одного запроса, чтобы начинать формировать следующий.

4.3 Проектирование базы данных сервиса «lk-user»

База данных кратко описана в пункте 2.2. В этом разделе будет описана ключевая сущность – сущность аналитического запроса.

Сущность аналитического запроса (analytics_request) служит для хранения запросов пользователей. В таблице Д.1 приведены поля этой сущности с пояснениями:

4.4 Проектирование клиент-серверного взаимодействия сервиса «lk-user»

Межсервисное взаимодействие частично представлено на рисунке 5. В этом пункте описано взаимодействие внутри сервиса «lk-user» и часть межсервисного взаимодействия.

На рисунке 17 можно заметить, что lk-user frontend и lk-user backend общаются путем «внутреннего API». Это означает, что обмен данными происходит путем шаблонизации HTML документов при помощи библиотеки twig. То есть без HTTP запросов к бекенду.

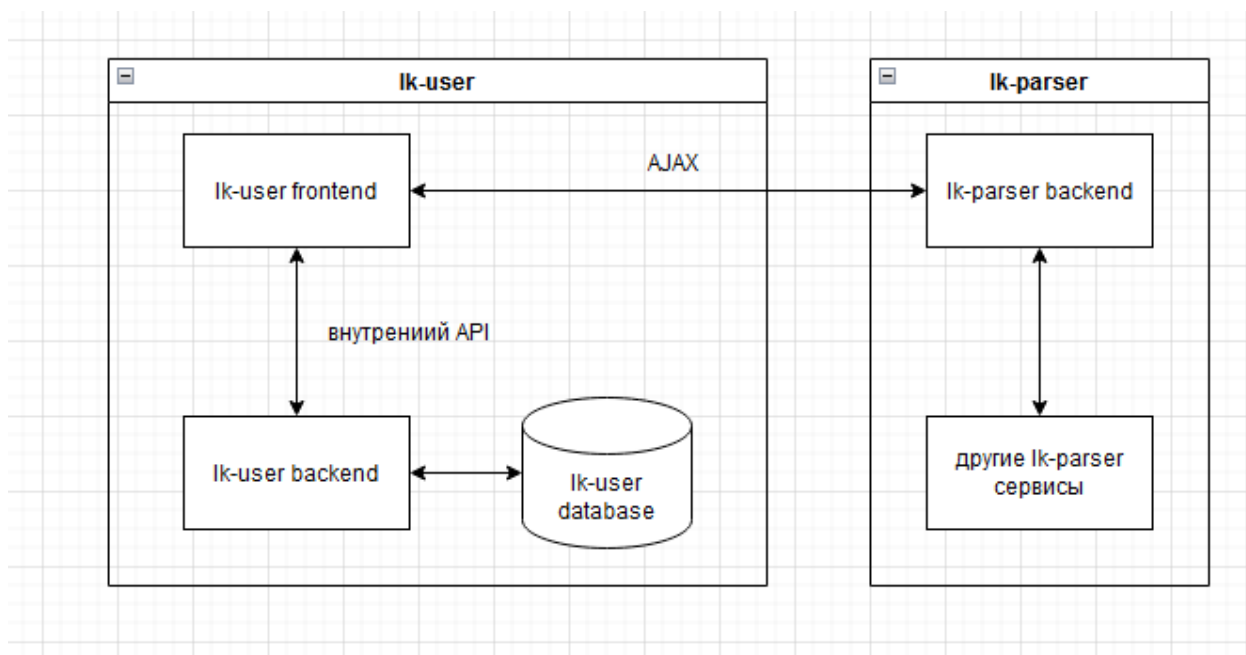


Рисунок 17 – Внутреннее и внешнее взаимодействие «lk-user» и «lk-parser»

А внешнее взаимодействие как раз происходит путем HTTP запросов при помощи AJAX запросов, которые являются частью библиотеки jQuery.

Таким образом на фронте не нужно реализовывать методы авторизации, не нужно отдельно обрабатывать валидацию форм. Т.к. все это реализовано на стороне бекенда Symfony Bundle, и попадают на клиентскую часть через twig

шаблоны. JavaScript скрипты нужны для обеспечения динамики форм. И внешних запросов.

4.5 Проектирование архитектуры «lk-user»

Под архитектурой здесь понимается компоновка файлов внутри проекта. Скелет всего проекта представлен на рисунке 18

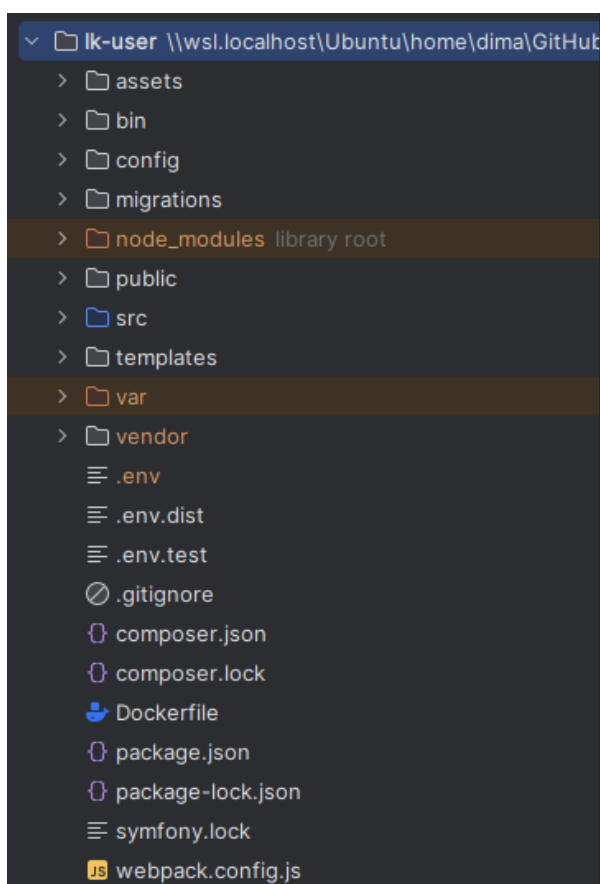


Рисунок 18 – Основной скелет проекта

Функциональность такого расположения файлов – следующая:

- а) assets – тут располагается клиентская часть приложения;
- б) src – тут располагается серверная часть приложения;
- в) public – тут располагается статика приложения;
- г) var, vendor, node_modules – служат для хранения библиотек;
- д) .env – файлы окружения;

е) composer.json, package.json – соответственно конфигурация php, javascript;

Далее прилагается скелет серверной части приложения. На рисунке 19 предоставляется содержание папки src. Само содержание подчиняется стандартному скелету Symfony – проекта.

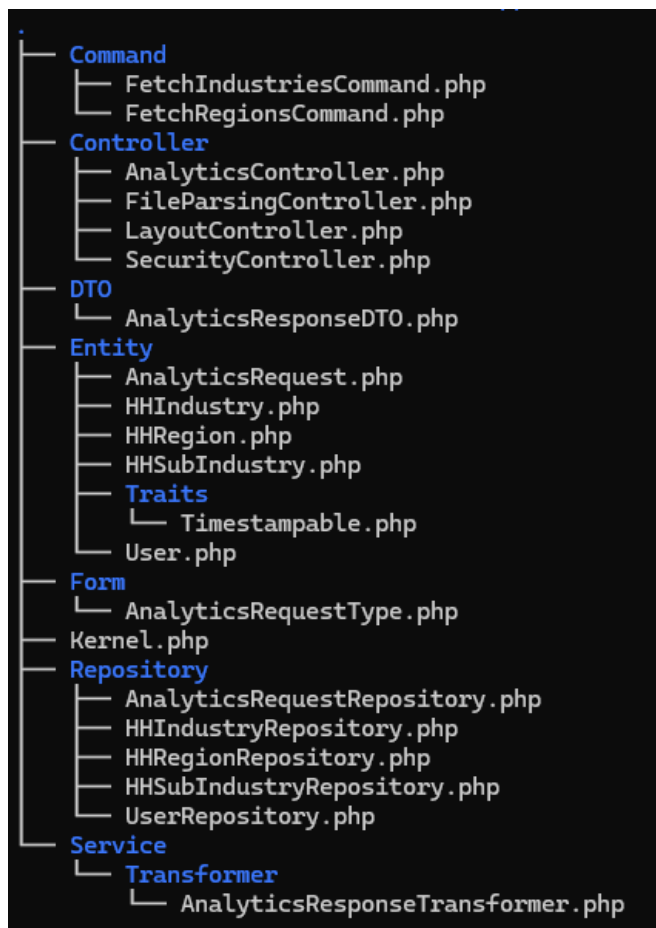


Рисунок 19 – Содержание папки src

Функциональность такого расположения файлов – следующая:

а) command – здесь лежат команды для актуализации индустрий и регионов с hh.ru;

б) controller – здесь располагаются контроллеры для различных действий пользователя;

в) DTO – здесь лежат объекты для унификации передачи данных;

г) entity – эти файлы являются моделями для базы данных;

д) form – формы запросов;

е) repository – через репозитории происходит доставка данных из базы данных;

ж) service – различные дополнительные сервисы.

Также в папке assets на рисунке 20 – располагаются JavaScript скрипты: из них (и шаблонов из папки templates) и состоит клиентская часть приложения.

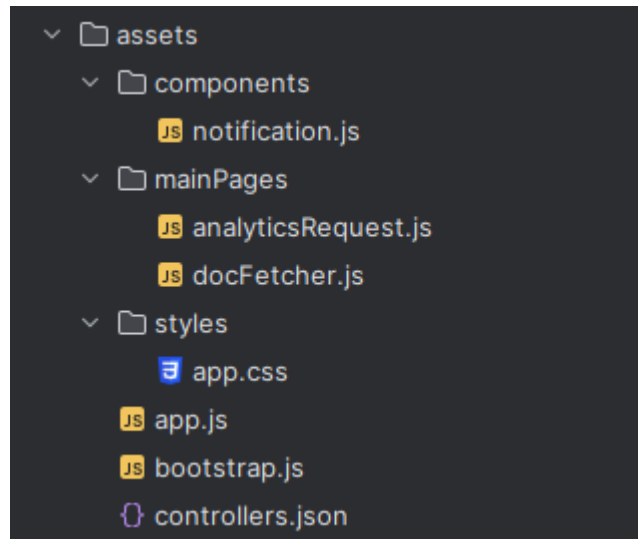


Рисунок 20 – папка assets

Webpack предоставляет удобную маршрутизацию файлов внутри проекта, таким образом можно настроить удобный доступ до файлов внутри шаблонов. Название папок говорит само за себя:

а) components – здесь лежат компоненты, которые следует переиспользовать в разных частях скриптов;

б) mainPages – здесь лежат скрипты для главных страниц приложения;

в) styles – здесь располагаются стили приложения;

г) app.js – точка входа в приложение;

д) bootstrap.js – интеграция UI/UX библиотеки bootstrap;

е) controllers.json – конфигурация маршрутизации контроллеров.

5. Реализация

5.1 Реализация клиентской части приложения

Специфика реализации клиентской части приложения в этом сервисе является то, что клиентская часть не является отдельным приложением со своим сервером. А просто сбор статики рядом с бекендом на Symfony 6. Это сделано для удешевления разработки и простоты поддержки. Чтобы запустить клиентскую часть «рядом» были предприняты следующие шаги:

Написан Dockerfile (рисунок 21), в котором отдельно учтено условие, когда на сервере отсутствует nodejs (node_modules). Таким образом достаточно запустить Dockerfile и не разворачивать модули отдельно.



```
1 FROM node:14-alpine
2
3 WORKDIR /lkuser
4
5 COPY package.json ./
6 COPY package-lock.json ./
7
8 RUN npm install
9
10 #prod
11 #CMD ["npm", "run", "build"]
12
13 #dev
14 CMD [ "-d", "node_modules" ] && npm run dev-server || npm ci && npm run dev-server
```

Рисунок 21 – содержание Dockerfile клиентской части приложения «lk-user»

Теперь чтобы Javascript собирался как статические файлы рядом с php, который тоже собирается как статические файлы, необходимо использовать сборщик модулей. В этом проекте был выбран Webpack — это бесплатный сборщик модулей с открытым исходным кодом для JavaScript. Он создан в первую очередь для JavaScript, но может преобразовывать внешние ресурсы,

такие как HTML, CSS и изображения, если включены соответствующие загрузчики. Webpack принимает модули с зависимостями и генерирует статические ресурсы, представляющие эти модули.

Настройка Webpack происходит автоматически при помощи библиотеки «@symfony/webpack-encore». То есть после установки командой `composer require symfony/webpack-encore-bundle` – файл конфигурации не менялся.

Важными частями клиентской части являются: обновление статуса документов, страница формирования аналитического запроса.

Обновление статуса документов реализована через long polling – используется в веб-приложениях реального времени для достижения практически мгновенной связи между клиентом и веб-сервером. Это особенно полезно в приложениях чата и обмена сообщениями, где обновления в реальном времени имеют решающее значение.

Запрос по документам на странице отправляется в «lk-parser» и записи на странице обновляются без обновления страницы. Метод, который отправляется каждые 5 секунд отображен в приложении Е.1. В самом методе вставляется HTML код по ключевым тегам на странице в зависимости от статуса документа.

Форма формирования запроса имеет multiselect поля, которые должны быть реализованы через checkbox, что означает, что из множества пунктов, должно быть выбрано несколько одновременно. Для реализации этой функциональности была выбрана библиотека select2. Также эта библиотека поддерживает текстовый поиск, что сильно улучшает пользовательский опыт. Пример некоторых полей приведен на рисунке 22.

```

1 dstartsev
$(document).ready(function() : void {
  $('#searchModifier').select2({
    width: '100%',
    placeholder: 'Выберите модификатор поиска',
    allowClear: true,
    closeOnSelect: false,
    theme: "bootstrap-5"
  });

  $('.toggle-all-industries').click(function() : void {
    $('.industry-group').toggle();
  });

  $('.toggle-industry').click(function() : void {
    var target = $(this).data('target');
    $(target).toggle();
  });

  $('#region').select2({
    width: '100%',
    placeholder: 'Выберите регион',
    allowClear: true,
    closeOnSelect: false,
    theme: "bootstrap-5"
  });
});

```

Рисунок 22 – пример реализации select2 селектора

Ошибки при отправке формы обрабатываются на стороне Symfony. И выводится через JavaScript. Ошибки автоматически обрабатываются в зависимости от типа, дополнительно наложено ограничение на пустое поле поиска. Вид формы приведен на рисунке 23.

```

2 usages 1 dstartsev
class AnalyticsRequestType extends AbstractType
{
  no usages 1 dstartsev
  public function buildForm(FormBuilderInterface $builder, array $options): void
  {
    $builder
      ->add( child: 'searchField', type: TextType::class, [
        'constraints' => [
          new NotBlank(['message' => 'Поле поиска не должно быть пустым']),
        ],
      ])
      ->add( child: 'qualificationLevel', type: TextType::class)
      ->add( child: 'region', type: CollectionType::class)
      ->add( child: 'vmi', type: CheckboxType::class)
      ->add( child: 'searchModifier', type: CollectionType::class)
      ->add( child: 'submit', type: SubmitType::class)
      ->add( child: 'industry', type: CollectionType::class)
      ->add( child: 'hasSalary', type: CheckboxType::class)
      ->add( child: 'employment', type: TextType::class)
      ->add( child: 'schedule', type: TextType::class)
      ->add( child: 'experience', type: TextType::class)
    ;
  }
}

```

Рисунок 23 – форма аналитического запроса

5.2 Реализация серверной части приложения

На рисунке 19 изображён скелет проекта. Если идти сверху вниз по файлам проекта, то функциональность они будут нести следующую:

Command – в проекте реализованы две команды, которые шлют запросы в hh.ru и актуализируют (сохраняют в базу данных) список регионов и список индустрий. Пример команды изображен на рисунке 24.

```
no usages 1 dstartsev
protected function execute(InputInterface $input, OutputInterface $output): int
{
    $io = new SymfonyStyle($input, $output);
    $io->title('Fetch industries');

    try {
        $industriesResponse = $this->httpClient
            ->request(['method' => 'GET', 'uri' => self::BASE_URI])
            ->getBody()
            ->getContents();

        $industries = json_decode($industriesResponse, ['associative' => true]);

        $io->info('Fetching industries...');

        if ($input->getOption('name' => 'refresh')) {
            $this->HHIndustryRepository->deleteAll();
        }

        foreach ($industries as $industry) {
            $hhIndustry = new HHIndustry();
            $hhIndustry->setName($industry['name']);
            $hhIndustry->setHhId($industry['id']);
            $subIndustries = $industry['industries'];
            foreach ($subIndustries as $subIndustry) {
                $hhSubIndustry = new HHSubIndustry();
                $hhSubIndustry->setName($subIndustry['name']);
                $hhSubIndustry->setHhId($subIndustry['id']);
                $hhSubIndustry->setHhIndustry($hhIndustry);
                $this->HHSubIndustryRepository->add($hhSubIndustry);
                $io->writeln('messages: Added SubIndustry: ' . $subIndustry['name']);
            }

            $io->writeln('messages: Added Industry: ' . $industry['name']);
        }

        $io->info('Industries fetched successfully!');
    }
}
```

Рисунок 24 – команда по сбору индустрий

Controller – контроллеры существуют чтобы обозначит маршрутизацию в приложении. В контроллерах реализованы все url – пути, которые присутствуют в приложении.

Entity – служат для реализации представления моделей в базе данных. В «lk-user» используется DoctrineORM, таким образом инициализация сущности выглядит как на рисунке 25.

```
namespace App\Entity;

use App\Repository\HHIndustryRepository;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;

14 usages  ▲ dstartsev
#[ORM\Entity(repositoryClass: HHIndustryRepository::class)]
class HHIndustry
{
    1 usage
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    private ?int $id = null;

    2 usages
    #[ORM\Column(length: 255, nullable: true)]
    private ?string $hhId = null;

    2 usages
    #[ORM\Column(length: 255, nullable: true)]
    private ?string $name = null;

    5 usages
    #[ORM\OneToMany(mappedBy: 'hhIndustry', targetEntity: HHSubIndustry::class)]
    private Collection $hHSubIndustries;

    1 usage  ▲ dstartsev
    public function __construct()
    {
        $this->hHSubIndustries = new ArrayCollection();
    }
}
```

Рисунок 25 – сущность индустрии

Благодаря подходу с описанием сущностей при помощи DoctrineORM – разработчик имеет возможность создавать автоматические миграции. Тем самым актуализируя базу данных среди разработчиков при каждом изменении.

ЗАКЛЮЧЕНИЕ

Разработка и внедрение системы HRParser позволило компании Artsofte решить проблему актуализации заработной платы сотрудников, оптимизировать расходы и повысить эффективность работы HR отдела. Автоматизация данного процесса стала ключевым фактором для улучшения управления персоналом и обеспечения конкурентоспособности компании на рынке труда

Анализ предметной области и выявление проблемы позволили сфокусироваться на разработке инновационного решения, которое значительно упростило и ускорило процесс актуализации заработной платы. Теперь HR отдел имеет возможность чаще проводить анализ рынка труда и обновлять данные о заработной плате сотрудников, что способствует более точному прогнозированию расходов компании и повышению мотивации персонала.

При разработке системы были выполнены следующие задачи:

- а) изучение процессов организации встречи в компании Artsofte;
- б) выявление и описание требований к системе у заказчика;
- в) описание функциональных требований к системе;
- г) описание сценариев использования;
- д) подготовка и передача технического задания в разработку.

Внедрение HRParser открывает новые перспективы для компании Artsofte, позволяя ей оперативно реагировать на изменения на рынке труда, улучшать условия работы сотрудников и обеспечивать устойчивое развитие в условиях динамичного бизнес-окружения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. TalentTech: офиц. сайт. – Москва. – URL: <https://main.talenttech.ru/> (дата обращения: 05.04.24)
2. Хабр (М.Тех – М.Видео–Эльдорадо): офиц. сайт. – Москва. – URL <https://habr.com/ru/companies/mvideo/articles/732966/> (дата обращения 05.04.24).
3. Информация по конкурентному сервису Хабр–карьера: офиц. сайт. – Москва. – URL: <https://career.habr.com> (дата обращения 05.04.24).
4. Информация по конкурентному сервису hh.ru: офиц. сайт. – Москва. – URL: <https://hh.ru> (дата обращения 05.04.24).
5. Информация по конкурентному сервису Rit.work: офиц. сайт. – Москва. – URL: <https://rit.work> (дата обращения 05.04.24).
6. Информация по конкурентному сервису Роснавук: офиц. сайт. – Москва. – URL: <https://lk.rosnavuk.ru> (дата обращения 05.04.24).
7. Информация по конкурентному сервису SuperJob: офиц. сайт. – Москва. – URL: <https://www.zarplatomer.ru> (дата обращения 05.04.24).
8. jQuery: офиц. сайт. – Сан-Франциско. – URL: <https://jquery.com/> (дата обращения 05.04.24).
9. PHP: офиц. сайт. – Сан-Франциско. – URL: <https://www.php.net/> (дата обращения 05.04.24).
10. Webpack: офиц. сайт. – Сан-Франциско. – URL: <https://webpack.js.org/> (дата обращения 05.04.24).
11. Build fast, responsive sites with Bootstrap: офиц. сайт. – Кембридж. – URL: <https://getbootstrap.com/> (дата обращения 05.04.24).
12. Производительность I/O бэкэнда Node vs. PHP vs. Java vs. Go : сайт - URL: <https://habr.com/ru/companies/vk/articles/329258/> Дата обращения 05.04.2024).

13. Локальные нейросети (генерация картинок, локальный chatGPT):
офиц. сайт. – Москва. – URL: <https://habr.com/ru/articles/727834/> (дата обращения 05.04.24).
14. ChatGPT openAPI: офиц. сайт. – Сан-Франциско. – URL:
<https://chat.openai.com/> (дата обращения 05.04.24).
15. HeadHunterAPI: офиц. сайт. – Москва. – URL:
https://github.com/hhru/api/blob/master/docs/authorization_for_application.md/ (дата обращения 05.04.24).
16. User flow: как создаются популярные приложения и сайты: офиц. сайт.
– Москва. – URL: <https://habr.com/ru/articles/496760/> (дата обращения 05.04.24).

ПРИЛОЖЕНИЕ А **(обязательное)** **Схема базы данных сервиса ЛК**

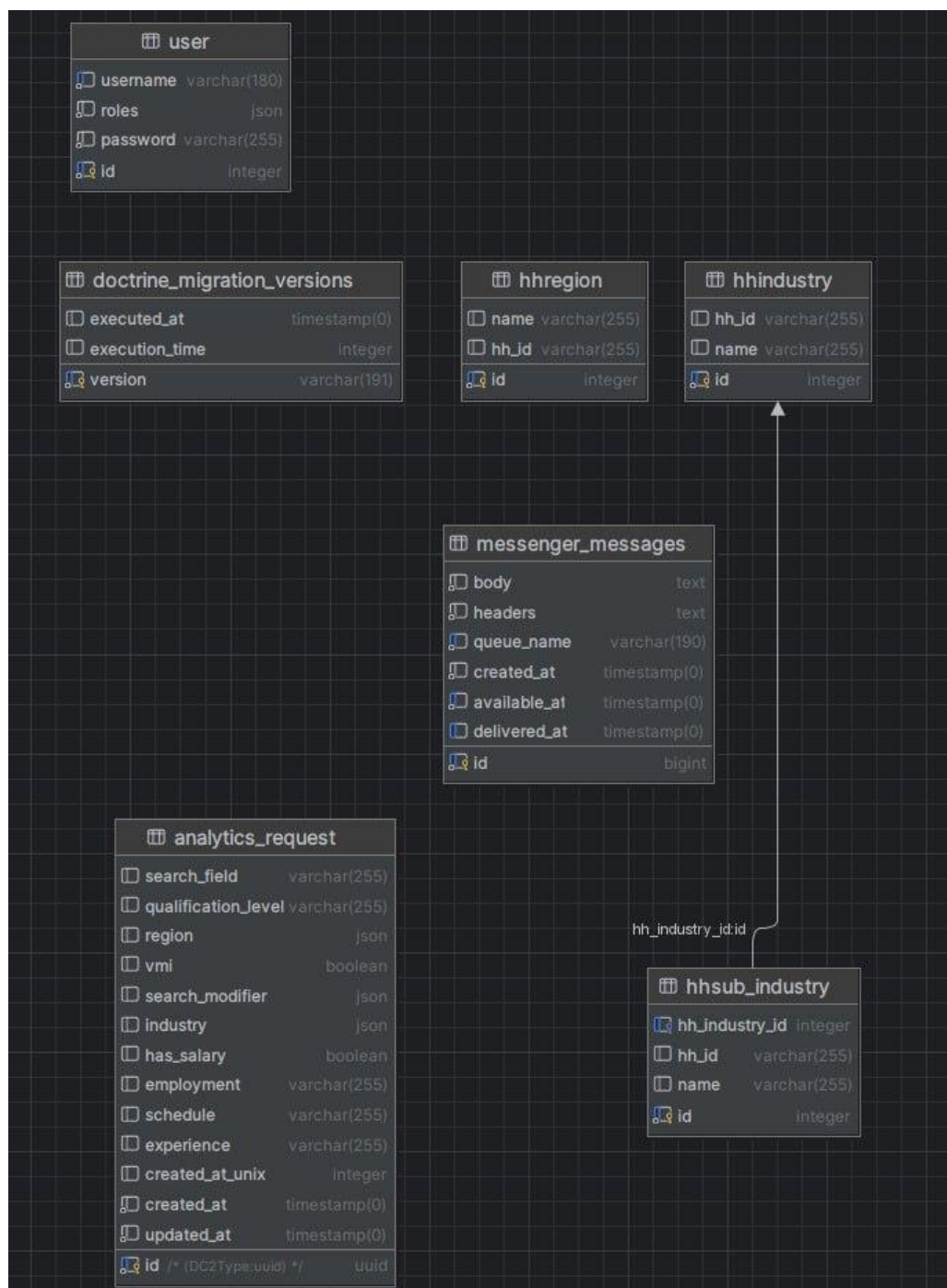


Рисунок А.1 – База данных сервиса личных кабинетов

(обязательное)

Схема базы данных сервиса аналитики

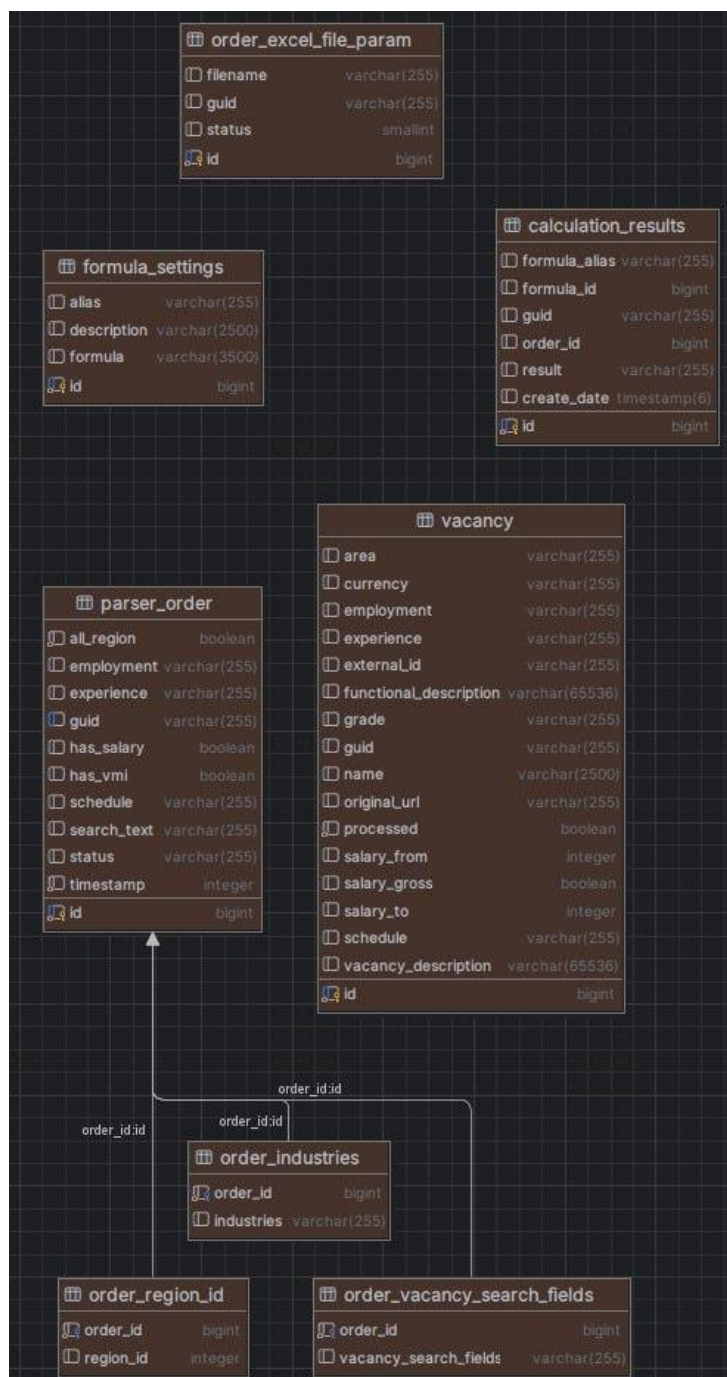


Рисунок Б.1 – База данных сервиса аналитики

ПРИЛОЖЕНИЕ В

(обязательное)

Техническое задание проекта

Техническое задание

1 Введение

Наименование программы – «Парсер вакансий».

Краткая характеристика области применения – система «Парсер вакансий» предназначена для сбора вакансий с сайта hh.ru по сформированному запросу. Система получает на вход запрос от оператора. На выход система выдаёт отчёт в формате [xlsx](#). Отчёт представляет собой набор вакансий с дополнительными аналитическими данными.

2 Основания для разработки

Основанием для разработки является договор от 01.02.2024. Договор утвержден Директором ООО «Бизерра.ру» Ивановым Иваном Ивановичем, именуемым в дальнейшем Заказчиком, и Петровым Петром Петровичем (самозанятый), именуемым в дальнейшем исполнителем, 01.02.2024.

Согласно Договору, Исполнитель обязан разработать и установить систему «Парсер вакансий» на оборудовании Заказчика не позднее 12.04.2024, предоставить исходные коды и документацию к разработанной системе не позднее 12.04.2024.

Наименование темы разработки – «Разработка системы Парсер вакансий для сбора аналитических данных по рынку труда». Условное обозначение темы разработки (шифр темы) – «Парсер-v1.0».

3 Назначение разработки

Программа будет использоваться HR-отделом компании «[Artsoft](#)». Одной группой пользователей: оператор-HR.

Рисунок В.1 – Первая страница технического задания

Продолжение ПРИЛОЖЕНИЯ В

Функциональное назначение – для оператора-HR`а система позволяет формировать выборку вакансий по, определенному в запросе, фильтру. Помимо набора вакансий выборка включает себя следующие данные: медиана по выборке, мода по выборке, средняя по выборке, перцентили по 25%, дополнительная информация по заработной плате.

Эксплуатационное назначение – программа должна эксплуатироваться HR-отделом. Запущенная по предоставленным реквизитам для входа. Дополнительного обучения персонала не требуется.

4 Требования к программе

4.1 Требования к составу выполняемых функций

После запуска программы пользователю отображается форма ввода логина и пароля, показанная на рисунке 6.

Please sign in

Username

Username



Password

Password

☐ Remember me

Sign in

Рисунок 1 – форма ввода логина и пароля

Система не различает права пользователей. Базовая авторизация нужна для предоставления защищённого доступа к ресурсу.

Рисунок В.2 – Вторая страница технического задания

Продолжение ПРИЛОЖЕНИЯ В

Для оператора система предоставляет следующие возможности:

Просмотр списка отчётов – оператор имеет возможность посещения страницы со списком отчетов (вкладка «Документы»), где отчёт однозначно определяется полем «Хэш», вид страницы представлен на рисунке 7.



Хэш	Запрос	Ссылка на скачивание	Статус	Количество	Дата
63180291	Java программист	Скачать файл	COMPLETED WITH DOCUMENT	81	2024-04-18 17:08:55
375eb67d	PHP-программист	Скачать файл	COMPLETED WITH DOCUMENT	30	2024-04-18 17:01:43
e45ca05b	Аналитик	Скачать файл	COMPLETED WITH DOCUMENT	129	2024-04-12 20:03:14
428d6617	Аналитик	Скачать файл	COMPLETED WITH DOCUMENT	2000	2024-04-12 20:01:22
e0b63704	Аналитик		PENDING ERROR	0	2024-04-08 20:40:35
82d899c3	Аналитик	Скачать файл	COMPLETED WITH DOCUMENT	774	2024-04-08 08:18:51
18b76312	Аналитик		PENDING	15299	2024-04-08 08:18:08
8de511ac	ТС разработчик		PENDING ERROR	0	2024-04-06 14:21:46
db2d40ae	ТС	Скачать файл	COMPLETED WITH DOCUMENT	250	2024-04-02 05:41:17
62c71183	Java программист	Скачать файл	COMPLETED WITH DOCUMENT	172	2024-04-01 13:57:11

Рисунок 2 – вид страницы «Документы»

Также оператор может определить нужный отчет по полю «Запрос», где отображается содержание запроса из поля запроса. Пользователь может скачать файл при наличии ссылки на скачивание, нажав на кнопку «Скачать файл». Пользователь может видеть статус выполнения запроса, количество вакансий по запросу и дату формирования запроса.

Запуск формирования отчета по собранной выборке – оператор может запустить генерацию файла, если количество вакансий в выборке его устраивает. Система не запускает генерацию файла автоматически чтобы не перегружать систему. Статус и вид кнопки «Сгенерировать файл» при завершённом поиске вакансий представлены на рисунке 8.

Рисунок В.3 – Третья страница технического задания

Продолжение ПРИЛОЖЕНИЯ В

Хэш	Запрос	Ссылка на скачивание	Статус	Количество	Дата
42f1264a	Java программист	Сгенерировать файл	COMPLETE	61	2024-05-03 16:06:37

Рисунок 3 – Вид отчёта, когда поиск вакансий по запросу закончен

Просмотр запроса по UUID выборки – при нажатии на поле «Хэш» пользователь может увидеть запрос, на основе которого сформирована выборка. Вид отправленного запроса можно увидеть на рисунке 9.

Analytics Request

General Information

UUID

42f1264a-3238-49eb-8a5d-658212474996

Поле поиска

Java программист

Уровень квал

Регион

Москва Московская область

ДМС

Нет

Additional Information

Модификатор поиска

Только с зарплатой

Да

Занятость

Полная занятость

График

Полный день

Опыт

От 1 года до 3 лет

Создан

2024-05-03 17:21:16

Индустрии

Все

Рисунок 4 – Информация о запросе после отправки запроса

Запуск поиска выборки – чтобы начать формирование выборки необходимо заполнить форму на вкладке «Запрос». Вид страницы представлен на рисунке 10. Если введены некорректные значения — то соответствующее пол в интерфейсе оператора выделяется красным цветом.

ИИ-Partner

Дополнительно

Запрос

Создать/поискать код вакансии

Поиск вакансии

Только с заработной платой

Уровень квалификации

Регион

Выборка по региону

ДМС

Выборка по ДМС

Модификатор поиска

Выборка модификатор поиска

Список индустрий

Правильно, неправильно, ошибка КИВ

Выборка индустрии

Информационная технология, компьютерная техника, инженерия

Выборка индустрии

Экономика, производство, услуги, транспорт, строительство и торговля

Выборка индустрии

Активна Windows
2024-05-03 16:06:37

Рисунок 5 – страница с формой запроса

Рисунок В.4 – Четвертая страница технического задания

Продолжение ПРИЛОЖЕНИЯ В

4.2 Требования к источникам данных

Вакансии в систему поступают с сайта hh.ru, посредством запросов, через API, предоставленным hh.ru.

Информацию по вакансиям, которую hh.ru не предоставляет, система получает из интеграции с нейросетью «ChatGPT». Данные, которые определяются нейросетью описаны в колонке «Определяется нейросетью» в таблице Г.1.

4.3 Требования к реализации

«Парсер вакансий» представляет собой два сервиса «lk-user» и «lk-parser», соответственно - модуль личного кабинета и модуль аналитики.

Модуль личного кабинета должен взаимодействовать с модулем аналитики по средством REST API, с использованием протокола HTTP/HTTPS.

На стороне модуля личных кабинетов должна быть реализована функциональность по макетам из пункта 4.1.

На стороне модуля аналитики должны быть реализованы методы CRUD – действий для сущности запроса. Также реализовано сохранение запросов и вакансий в СУБД.

Также необходимы методы для создания и получения xlsx файлов, содержание которых описано в таблице Г.1.

Если в вакансии указана зарплата «от» и «до», то для аналитических расчетов следует брать среднюю по этим величинам.

Должна быть реализована интеграция с нейросетью для обогащения данных вакансии, которые нельзя получить через интеграцию с hh.ru. Данные, которые определяются нейросетью описаны в колонке «Определяется нейросетью» в таблице Г.1.

Интеграция должна происходить по следующему протоколу:

- а) нужно проксировать запросы через сервис neuroapi.host;

Рисунок В.5 – Пятая страница технического задания

Продолжение ПРИЛОЖЕНИЯ В

б) нужно предоставить интерфейс смены api-ключа neuroapi.host для системного администратора при истечении токенов.

в) нужно получать данные из нейросети путем запроса, содержание которого приведено на рисунке 11.

```
Ты получил на вход заголовок вакансии и описание вакансии.
Тебе нужно дать ответ в JSON и только в виде JSON вида:
{
  "VMI": true/false,
  "functional": "some_functional_by_description",
  "keySkills": {"skill1", "skill2", "skill3"},
  "grade": "enum: junior, middle, senior, teamlead"
},
где VMI – наличие ДМС в вакансии, functional – чем предстоит заниматься на рабочем месте (функциональность вакансии), keySkills – набор ключевых навыков из вакансии в виде массива, grade – определи уровень квалификации сотрудника по описанию вакансии и подставь нужный уровень из списка: junior, middle, senior, teamlead.
ОЧЕНЬ ВАЖНО! Если заголовок не соответствует описанию вакансии верни ответ в виде JSON вида:
{
  "success": false
}
И ничего больше.
```

Рисунок 6 – запрос к лингвистической модели

4.4 Требования к организации входных и выходных данных

Валидация данных выполняется на стороне сервера. Описание полей запроса приведено в таблице Д.1.

Вывод системы представляет собой xlsx файл – таблицу, содержание которой представлено в таблице Г.1.

4.5 Требования к временным характеристикам

После отправки запроса на сервер, новая информация на клиентах отображается не позднее, чем через 5 секунд. Длительность хранения данных в СУБД регулируется на стороне администратора СУБД.

Рисунок В.6 – Шестая страница

Продолжение ПРИЛОЖЕНИЯ В

5 Требования к надежности

Вероятность безотказной работы системы должна составлять не менее 99.99% при условии исправности сети (связи приложений оператора и посетителя с базой данных).

5.1 Требования к обеспечению надежного (устойчивого) функционирования программы

В связи с тем, что в базе данных хранятся данные о совершенных клиентами покупках (финансовая информация) — базу данных стоит резервировать (резервирование замещением).

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- а) организацией бесперебойного питания технических средств;
- б) использованием лицензионного программного обеспечения;
- в) регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
- г) регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

Рисунок В.7 – Седьмая страница технического задания

Продолжение ПРИЛОЖЕНИЯ В

5.2 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 10 минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

5.3 Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине не следует перегружать систему большим числом пустых запросов. И не следует использовать систему в не штатном режиме.

6 Условия эксплуатации

Программа (клиент) запускается в браузере оператора-HR`а. База данных находится на сервере заказчика, как и серверная часть приложения. Должна существовать устойчивая связь по сети между клиентами и базой данных.

Для бесперебойной работы системы не следует отклоняться от адекватных запросов к системе.

Рисунок В.8 – Восьмая страница технического задания

Продолжение ПРИЛОЖЕНИЯ В

6.1 Требования к видам обслуживания

По истечении, прохождении в негодность, или создании токенов для интеграций – следует производить процесс выпуска токенов.

Для hh.ru процесс выпуска токенов описан в Официальной документации hh.ru [1] – электронный ресурс.

Для neiroapi.host процесс выпуска токенов описан в Официальной документации neiroapi.host[2] – электронный ресурс.

6.2 Требования к численности и квалификации персонала

При установке и настройке системы необходим системный администратор. В процессе эксплуатации с программой работают оператор-НР.

Системный администратор должен иметь высшее профильное образование. В перечень задач, выполняемых системным администратором, должны входить:

- а) развертывание Docker контейнеров;
- б) настройка СУБД;
- в) настройка сертификатов для https соединения;
- г) настройка интеграций (заполнение валидных ключей).

Пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом установленных браузеров.

7 Требования к составу и параметрам технических средств

Состав технических средств:

- а) Компьютер оператора, включающий в себя:

Рисунок В.9 – Девятая страница технического задания

Продолжение ПРИЛОЖЕНИЯ В

- а. процессор x86/64 с тактовой частотой, не менее 1 ГГц;
 - б. оперативную память объемом, не менее 1 Гб;
 - в. видеокарту, монитор, мышь, клавиатура.
- б) Сервер для СУБД и серверной части, включающий в себя:
- а. процессор x86/64 (4 ядра) с тактовой частотой, не менее 2 ГГц;
 - б. оперативную память объемом, не менее 4 Гб;

8 Требования к программной документации

Предварительный состав программной документации:

- а) техническое задание;
- б) программа и методика испытаний (пункт 11);
- в) описание процесса выпуска токенов (пункт 4.3);

9 Техничко-экономические показатели

Программа "Парсер вакансий" призвана упростить процесс анализа рынка для компаний, которые не готовы вкладывать значительные ресурсы в эту задачу. Вероятнее всего, она будет особенно полезна для небольших IT-фирм. Хотя функционал программы в целом совпадает с аналогами, её главное отличие заключается в возможности создания индивидуализированных выборок вакансий в соответствии с потребностями заказчика.

Зачастую бесплатное распространение программы может создать высокий спрос на неё. Это объясняется тем, что она предоставляет возможность значительно упростить и ускорить процесс анализа вакансий. Тем не менее, для обеспечения экономической выгоды может быть введена платная установка системы. Такой подход позволит компании получать прибыль и одновременно удовлетворять потребности клиентов в высококачественном инструменте анализа рынка труда.

Рисунок В.10 – Десятая страница технического задания

Продолжение ПРИЛОЖЕНИЯ В

10 Стадии и этапы разработки

Разработка должна быть проведена в три стадии:

- а) техническое задание;
- б) технический (и рабочий) проекты;
- в) внедрение.

На этапе разработки «Техническое задание» необходимо провести процесс согласования и утверждения данного документа.

На этапе "Технический (и рабочий) проект" требуется завершить следующие этапы работы:

- а) разработка программы;
- б) испытания программы.

На этапе "Внедрение" необходимо осуществить процесс разработки "Подготовки и передачи программы".

Содержание работ по этапам:

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- а) определение требований к программе;
- б) определение и уточнение требований к техническим средствам;
- в) постановка задачи;
- г) определение стадий, этапов и сроков разработки программы;
- д) согласование и утверждение технического задания.

На этапе разработки программы необходимо провести работы по программированию и отладке программного кода.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

- а) разработка, согласование и утверждение порядка и методики испытаний;
- б) проведение приемо-сдаточных испытаний;
- в) корректировка программы по результатам испытаний.

Рисунок В.11 – Одиннадцатая страница технического задания

Продолжение ПРИЛОЖЕНИЯ В

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы в эксплуатацию на объектах заказчика.

11 Порядок контроля и приемки

Приемка системы осуществляется заказчиком после завершения всех этапов разработки и испытаний программы.

Перед приемкой системы заказчику предоставляется отчет о выполнении работ по техническому заданию, включая перечень выполненных задач, описание процесса развертывания программы, а также результаты проведенных испытаний.

Заказчик осуществляет проверку выполнения требований к техническим средствам, программной совместимости, а также соответствия функциональности программы заявленным характеристикам.

Приемка системы считается завершенной после подтверждения заказчиком соответствия программы заявленным требованиям и успешного прохождения приемо-сдаточных испытаний.

После приемки системы системный администратор заказчика обязуется обеспечить ее эксплуатацию в соответствии с документацией и рекомендациями разработчика.

В случае выявления несоответствий или недостатков заказчик вправе предъявить требования по их устранению разработчику, после чего проводится повторная проверка и приемка системы.

Рисунок В.12 – Двенадцатая страница технического задания

Продолжение ПРИЛОЖЕНИЯ В

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. NeuroAPI: офиц. сайт. – Москва. – URL: <https://neuroapi.host/> (дата обращения 05.04.24).
2. HeadHunterAPI: офиц. сайт. – Москва. – URL: https://github.com/hhru/api/blob/master/docs/authorization_for_application.md (дата обращения 05.04.24).

Рисунок В.13 – Список использованных источников технического задания

ПРИЛОЖЕНИЕ Г
(обязательное)
Содержание xlsx файла (отчета)

Таблица Г.1 – Содержание xlsx файла (отчета)

Столбец	Пояснение	Определяет ся нейросетью	Пример
1	2	3	4
Ссылка на вакансию в hh.ru	Прямая ссылка на страницу вакансии на hh.ru	Нет	https://ekaterinburg.hh.ru/vacancy/90947747
Набор навыков из словаря hh.ru	Ключевые навыки, которые работодатель ожидает у соискателя, взятые из словаря hh.ru	Да	Python, SQL, JavaScript, аналитическое мышление, коммуникабельность
Название вакансии	Официальное название вакансии, с площадки hh.ru	Нет	Специалист по разработке программного обеспечения
Грейд	Уровень квалификации, указанный в вакансии	Да	Junior

Продолжение ПРИЛОЖЕНИЯ Г

Продолжение таблицы Г.1

1	2	3	4
Задача и функционал	Описание основных задач и обязанностей, которые предстоит выполнять сотруднику в данной вакансии	Нет	Разработка и поддержка программного обеспечения, участие в проектировании систем, тестирование и оптимизация кода
Описание вакансии	Описание вакансии, предоставленное работодателем	Нет	Мы ищем опытного специалиста с углубленными знаниями в области программирования для участия в проектах по созданию высокотехнологичных решений

Продолжение ПРИЛОЖЕНИЯ Г

Продолжение таблицы Г.1

Требования в вакансии	Опыт работы	Нет	Опыт работы от 3 лет, знание Python и опыт работы с базами данных, высшее техническое образование
Наличие ДМС	Информация о наличии медицинской страховки	Да	Да
Рабочее время	Информация о наличии рабочем графике	Да	Гибкий график
Удаленная работа	Информация о возможности удаленной работы	Нет	Да
Резул льтирующий уровень заработной платы	Диапазон заработной платы, который предоставляется для данной вакансии	Нет	80 000 – 120 000

Продолжение ПРИЛОЖЕНИЯ Г

Продолжение таблицы Г.1

Название компании	Название компании, предлагающей вакансию	Нет	ArtSofte
ID компании	ID компании с сайта hh.ru	Нет	1234567

ПРИЛОЖЕНИЕ Д
(обязательное)
Аналитический запрос

Таблица Д.1 – содержание аналитического запроса

Название поля	Предназначение
Текст поиска	Аналогичен «умному поиску» с hh.ru (описание есть в https://ekaterinburg.hh.ru/article/1175)
Уровень квалификации сотрудника	Поиск производится только среди этого уровня квалификации
Регион поиска	Позволяет получить выборку вакансий по определённому региону
Наличие ДМС	Предоставляет вакансии только с наличием ДМС
Поля поиска	Модифицирует поле «Текст поиска», предоставляя опции: поиск в названии, поиск в названии компании, поиск в описании вакансии
Список индустрий	Возможность отфильтровать результат запроса по списку индустрий hh.ru
Вакансии только с заработной платой	При включенной опции вакансии без указания заработной платы будут исключены из отчета
Тип занятости	Возможность отфильтровать результат запроса по полям: полная занятость, частичная занятость, проектная работа/разовое задание, волонтерство, стажировка, оформление по ГПХ или по совместительству
График работы	Возможность отфильтровать результат запроса по полям: полный день, сменный график, гибкий график, удаленная работа. вахтовый метод
Опыт работы	Возможность отфильтровать результат запроса по полям: не имеет значения, нет опыта, от 1 года до 3 лет, от 3 до 6 лет, более 6 лет

ПРИЛОЖЕНИЕ Е

(обязательное)

Сущность аналитического запроса

Таблица Е.1 – содержание аналитического запроса

Название поля	Тип поля	Предназначение
ID	uuid	Строка для однозначной идентификации сущности
search_field	varchar(255)	Строка для хранения поля поиска
qualification_level	varchar(255)	Строка для хранения уровня квалификации
region	json	Объект для хранения сущности региона
vmi	boolean	Булево значение для обозначения наличия ДМС
search_modifier	json	Объект для хранения сущности модификатора поиска
industry	json	Объект для хранения сущности индустрии
has_salary	boolean	Булево значение для обозначения наличия зарплаты в вакансии
employment	varchar(255)	Строка для хранения типа занятости
schedule	varchar(255)	Строка для хранения графика работы
experience	varchar(255)	Строка для хранения опыта работы
created_at_unix	integer	Unix представление времени создания сущности

ПРИЛОЖЕНИЕ Ж

(обязательное)

Актуализация документов в кодовом представлении

```
var statusColorMapping : Map<String, String> = {
    'START_PARSING': 'bg-warning text-dark',
    'PARSING': 'bg-warning text-dark',
    'POST_PROCESSING': 'bg-warning text-dark',
    'MATHEMATICS': 'bg-warning text-dark',
    'PARSING_ERROR': 'bg-danger',
    'COMPLETE': 'bg-success',
    'START_CREATE_DOCUMENT': 'bg-warning text-dark',
    'COMPLETE_WITH_DOCUMENT': 'bg-success',
    'CREATING_DOCUMENT': 'bg-warning text-dark'
};

1+ usages 1 dstartsev
function updateData() : void {
    $.ajax({
        type: 'POST',
        url: '${API_URL}:${API_PORT}/api/v1/analytics/orders',
        contentType: 'application/json',
        data: JSON.stringify(postData),
        success: function (responseData) : void {
            responseData.data.forEach(function (item, index) : void {
                var tableRow = $('tr[data-id="' + item.guid + '"]');
                var statusCell : any | JQuery = tableRow.find('#status-' + item.guid);
                var vacancyCount : any | JQuery = tableRow.find('#count-' + item.guid);

                statusCell.text(item.status);
                vacancyCount.text(item.vacancyCount);

                var classes = statusColorMapping[item.status];
                if (classes) {
                    statusCell.html('<span class="badge ' + classes + '">' + item.status + '</span>');
                }

                if (item.filePath) {
                    if (item.status === 'COMPLETE_WITH_DOCUMENT') {
                        var downloadLink : string = '<a class="btn btn-primary" href="' + item.filePath + '" download>Скачать';
                        tableRow.find('#downloadLink-' + item.guid).html(downloadLink);
                    }
                } else {
                    if (item.status === 'COMPLETE') {
                        var generateButton : string = '<button class="generate-button btn btn-primary" data-guid="' + item.guid + '">Генерация';
                        tableRow.find('#downloadLink-' + item.guid).html(generateButton);
                    }
                }
            });

            $('button.generate-button').on('click', function (e) : void {
                e.preventDefault();
                var guid = $(this).data('guid');
                var button = $(this);

                button.removeClass('btn-primary').addClass('btn-success');

                button.prop('disabled', true).text('Генерация...');

                generateFile(guid);
            });

            compareRequests(prevRequestData, responseData);

            prevRequestData = responseData;

            firstRequest = false;
        },
        error: function (error) : void {
            console.error('Ошибка при выполнении AJAX-запроса:', error);
        }
    });
};
```

Рисунок Ж.1 – скрипт на обновление запросов