

Министерство образования и науки Российской Федерации
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой
_____ Г.И. Радченко
«__ » 2018 г.

Разработка веб-сервиса для поиска
и анализа резюме на базе нейронных сетей

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090301.2018.131 ПЗ ВКР

Руководитель работы,
к.т.н., доцент
«Электронные
вычислительные машины»
_____ И.Л. Кафтанников
«__ » 2018 г.

Автор работы
студент группы КЭ-484
_____ Д.С. Жаворонков
«__ » 2018 г.

Нормоконтролёр, ст. преп. каф.
«Электронные вычислительные
машины»
_____ В.В. Лурье
«__ » 2018 г.

Оглавление

ВВЕДЕНИЕ.....	8
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	9
Выводы по разделу один	19
2 ТРЕБОВАНИЯ К РАЗРАБАТЫВАЕМОМУ ПРИЛОЖЕНИЮ.....	19
2.1 Требования к системе в целом.....	19
2.2 Функциональные требования	20
2.3 Требования к видам обеспечения	20
2.4 Требования к лингвистическому обеспечению	21
2.5 Варианты использования системы	22
Выводы по разделу два	22
3 АРХИТЕКТУРА СЕРВИСА	23
3.1 Компонент PostController.....	24
3.2 Компонент ParseController	25
3.3 Компонент ML	26
Выводы по разделу три	28
4 ВЫБОР ТЕХНОЛОГИИ ФОРМИРОВАНИЯ ОЦЕНКИ СОВПАДЕНИЯ.....	29
4.1 Корреляционный анализ.....	29
4.2 Регрессионные деревья.....	31
4.3 Искусственные нейронные сети	33
Выводы по разделу четыре	35
5 ОБОСНОВАНИЕ И ВЫБОР ПРОГРАММНЫХ СРЕДСТВ, НЕОБХОДИМЫХ ДЛЯ РЕАЛИЗАЦИИ.....	35
5.1 Серверная часть.....	36
5.2 Клиентская часть	37
5.3 База данных	37

Изм.	Лист	№ докум.	Подпись	Дата

5.3 Искусственная нейронная сеть	38
Выводы по разделу пять	40
6 РЕАЛИЗАЦИЯ СИСТЕМЫ.....	40
6.1 Реализация компонента ParseController.....	40
6.2 Реализация нейронной сети.....	46
6.3 Реализация базы данных.....	49
6.4 Реализация клиентской части.....	52
Выводы по разделу шесть	61
7 ПРИМЕРЫ РАБОТЫ	62
Выводы по разделу семь.....	63
ЗАКЛЮЧЕНИЕ	64
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	65

Изм.	Лист	№ докум.	Подпись	Дата

ЮУрГУ-090301.2018.135

Лист
7

ВВЕДЕНИЕ

В конце прошлого века руководители крупных компаний столкнулись с ситуацией, когда спрос на высококвалифицированных работников стал явно превышать предложение.

Проблема оптимального подбора кадров является актуальной для любого предприятия. От ее решения зависят качество и отдача каждого сотрудника, конечный успех предприятия в целом. Характерными особенностями данной проблемы являются субъективность и неопределенность. Субъективность выражается в том, что оценку и подбор персонала производят люди со своими стереотипами, симпатиями, предпочтениями, которые, иногда, могут идти вразрез с целями самого предприятия. Неопределенность проявляется в неточных формулировках требований типа «уверенное владение Microsoft Word», «опытный пользователь ПК», «умение работать в команде», допускающими трактовку в очень широких пределах. Решение такой задачи требует всесторонних знаний, но каким бы широким ни был кругозор человека, фактор субъективности полностью исключить не удастся.

Целью данной работы является реализация алгоритмов анализа рынка труда и, таким образом, разработка web-сервиса поиска и анализа резюме по заданной вакансии. Исходя из поставленной цели, задачами данного проекта являются:

- анализ существующие подходы и/или методы подбора персонала в условиях субъективности и неопределенности;
- рассмотрение возможности использование данных методов применительно к задаче;
- разработка искусственной нейронной сети для анализа резюме;
- реализация web-сервиса «СПАР».

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

В наше время активно развиваются сервисы для подбора персонала, нахождения работы. HeadHunter, avito, zarplata далеко не единственны сервисы предоставляющие эти услуги. Соответственно с ростом количества таких сервисов появляются стандарты формирования резюме, вакансий.

Для примера можно рассмотреть типовые резюме с разницей в хронологии 10 лет.

На рисунке 1.1 представлено «плохое» резюме. В нем не учтены стандарты по формированию, плохо систематизирована информация. Рекрутер, взглянув на такое резюме, сделает вывод раньше, чем его прочтет:

ФИОЛИЮБА ЛЮДМИЛА ВИКТОРОВНА
Дата рождения 01.06.64 г.
Marin.....a@mail.ru
Телефон :89507749444. 3733456
САНКТ — ПЕТЕРБУРГ .ул.Варшав..
Должность :КУРЬЕР
ОБРАЗОВАНИЕ СРЕДНЕЕ-СПЕЦИАЛЬНОЕ.
Опыт работы :01.09.01г.- 23.04.02 ООО АССОЦИАЦИЯ АиР
ЭКСПЕДИТОР.
13.05.02- 18.11.02 ООО Вега-Строй.
ЭКСПЕДИТОР.
28.07.03-12.03.04 ООО Коловрат МЕНЕДЖЕР ОТД.сбыта.
01.10.05- 06.11.09 КОМП. ЖИЗНЕЛЮБ МЕНЕДЖЕР по доставке.
ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ На данный момент я нахожусь в поиске работы
.Ищу подвижную энергичную работу .График с9-18-19ч.Семейное положение
ЗАМУЖЕМ 19лет.Имею дочь 17 лет

Рис. 1.1 — Образец «плохого» резюме.

Здесь мы можем наблюдать небрежность, грамматические ошибки, отсутствие фото, бессистемность. Такое резюме отталкивает.

Посмотрим пример современного резюме, отвечающей нормам на рисунках 1.2 и 1.3:

Изм.	Лист	№ докум.	Подпись	Дата	Лист	ЮУрГУ-090301.2018.135	9

Иван Иванов

Пол, дата рождения

Контактный телефон

Контактный эл. адрес

Проживает: город

Гражданство:

Готов \ не готов к переезду, готов \ не готов к командировкам

Желаемая должность и зарплата

Специалист по организации мероприятий

Маркетинг, реклама, PR

— Продвижение, специальные мероприятия

Занятость: полная занятость

График работы: полный день

Желательное время в пути до работы: не имеет значения

Опыт работы — 13 лет 8 месяцев

Октябрь 2016 — Салон связи
настоящее время Москва

Рис. 1.2 — Образец современного резюме.

Как видно из рисунка 1.2, основными отличительным чертами современного резюме является актуальность и информативность. Важное значение имеет тот факт, что, бросив взгляд на это резюме, работодатель отметит заинтересованность претендента, его предварительную подготовку, мониторинг других резюме и у него возникнет желание пригласить соискателя на собеседование.

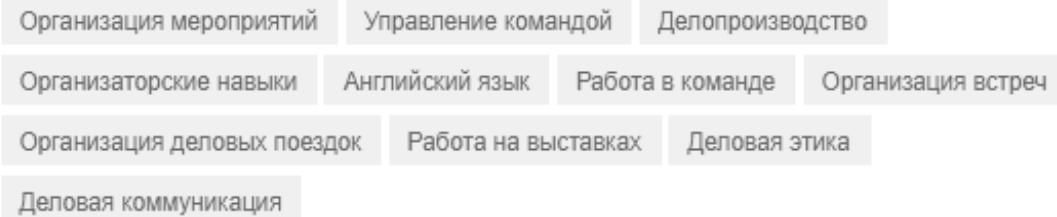
Изм.	Лист	№ докум.	Подпись	Дата

ЮУрГУ-090301.2018.135

Лист

10

Ключевые навыки



Рекомендации

по запросу

Обо мне

Опыт организации различных мероприятий: деловых и личных.

В период учебы в вузе участвовал в работе студенческого совета, организовывал различные мероприятия для студентов: спортивные, социальные, творческие.

Опыт организации корпоративных мероприятий.

Опыт участия в проекте по организации международной конференции.

Развитые организационные, коммуникативные и презентационные навыки.

Навыки продаж и ведения переговоров.

Знание основ документооборота и договорной работы.

Опыт проведения маркетинговых исследований рынка.

Умение работать в режиме многозадачности.

Рис. 1.3 — Ключевые навыки.

На рисунке 1.3 представлено резюме, в котором строго выдержаны требования к структуре, информация подана в правильной, логической последовательности. Дизайн также играет немаловажную роль.

Также, для анализа рынка труда, необходимо было рассмотреть и проанализировать множество вакансий, чтобы выявить метрики, чаще

Изм.	Лист	№ докум.	Подпись	Дата

всего встречающиеся в резюме, четко характеризующие претендента, и, одновременно, указанные в вакансиях.

Рассмотрим несколько примеров вакансий. На рисунках 1.4 и 1.5 представлена вакансия с сайта HeadHunter:

Senior Backend Developer | PHP программист | Ведущий web разработчик

от 85 000 руб. на руки

GILMON ✓

Челябинск, проспект Ленина, 55А



Откликнуться

Показать контакты



Требуемый опыт работы: 1–3 года

Полная занятость, полный день

Мы ищем профессионала, свободно владеющего стеком технологий PHP/MySQL.

Основные задачи, которые Вы будете решать:

- разработка и поддержка нового функционала;
- внедрение новых решений для автоматизации бизнеса (CRM, ERP);
- решение сложных алгоритмических задач;
- поддержка и развитие внутреннего API;

Рис. 1.4 — Вакансия PHP-программист с сайта HeadHunter (часть 1).

Анализируя информацию, представленную на рисунке 1.4 мы можем отчетливо видеть, что первой строкой указывают опыт работы и тип занятости. Дальше на рисунке претенденту предоставлена информация о задачах, которые ему необходимо будет решать.

Далее на рисунке 1.5 мы видим пункт «Требования к кандидату», в котором указываются профессиональные навыки, необходимые для

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ЮУрГУ-090301.2018.135

Лист

12

работы в данной компании. Этот пункт тоже очень важен, так как максимально характеризует претендента.

Требования к кандидату:

- опыт разработки на PHP от 2 лет;
- наличие портфолио или открытых проектов на GitHub;
- хорошее понимание концепции ООП, опыт её применения;
- умение читать и понимать чужой код;
- опыт работы с нереляционными базами данных и системами кеширования;
- последний год вы работали в аналогичной должности.

Плюсами будут:

- опыт работы с языками программирования JavaScript/Python/Java или любым другим;
- тесное знакомство с различными базами данных;
- знание английского языка;
- умение работать с кешированием, понимание целесообразности его применения;
- понимание работы RESTful сервисов.

Условия труда:

- современный офис в центре города;
- молодой и дружелюбный коллектив;
- все задачи ставятся в Trello, обсуждение проходит в Slack;
- развитая корпоративная культура;

Рис. 1.5 — Вакансия PHP-программист с сайта HeadHunter (часть 2).

Для того, чтобы выдержать независимость от платформы, иначе говоря – кроссплатформенность, рассмотрим еще несколько сайтов по подбору персонала. Пример вакансии с сайта avito.ru представлен на рисунках 1.6 и 1.7:

Изм.	Лист	№ докум.	Подпись	Дата	Лист	ЮУрГУ-090301.2018.135	13
------	------	----------	---------	------	------	-----------------------	----

WEB программист. html-верстальщик. Frontend-разраб

25 000 ₽

№ 1422413419, размещено 25 мая в 13:25 51 (+5)

 Добавить в избранное

 Добавить заметку



Сфера деятельности: IT, интернет, телеком

График работы: полный день

Опыт работы: более 1 года

Адрес: Челябинск, район Центральный, улица Татьяничевой, 9А
[Посмотреть карту](#)

Описание работодателя: в быстро развивающуюся сплоченную компанию требуется гений верстки и веб разработки

Обязанности:

Адаптивная вёрстка макетов, интеграция сайтов на CMS WordPress. Желание развиваться в сфере web программирования, поиск решений по задачам

[Показать телефон](#)

8 912 XXX-XX-XX

[Откликнуться](#)

Отвечает за несколько часов



Direct-romanoff ОOO

"Делориан"

Работодатель

На Avito с июля 2015

Завершено 16 объявлений

Контактное лицо

Алексей

Адрес:

Челябинск, район Центральный, улица Татьяничевой, 9А

Рис. 1.6 — Вакансия web-программист с сайта avito.ru (часть 1).

На рисунке 1.6 в примере с сайта avito.ru мы также видим, что опыт работы и тип занятости пишут на первом месте. Далее на рисунке 1.7 представлены обязанности, требования и условия.

адаптивная вёрстка макетов, интеграция сайтов на CMS WordPress. Желание развиваться в сфере web программирования, поиск решений по задачам связанным с WEB программированием.

ребования:

'веренные знания HTML5/CSS3, JavaScript, PHP, jQuery. Знаниеrossбраузерной верстки. Наличие опыта работы с различными CMS приветствуется, основная работа связана с версткой и интеграцией сайтов на CMS WordPress.

словия:

Полный рабочий день с 10-18, дружная и сплоченная команда, помимо коммерческой работы с сайтами заказчиков, возможность участия в интересных собственных проектов. Своевременная оплата труда 2 раза в месяц. Официальное трудоустройство. Возможность выполнять задачи по работе вне офиса при качественной реализации.

Рис. 1.7 — Вакансия web-программист с сайта avito.ru (часть 2).

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ЮУрГУ-090301.2018.135

Лист

14

Пример с сайта chelyabinsk.zarplata.ru изображен на рисунках 1.8 и

1.9:

The screenshot shows a job posting from the website chelyabinsk.zarplata.ru. The title of the job is '40 000 – 80 000 руб.' (40,000 – 80,000 rubles). The employer is 'ИНТЕРСВЯЗЬ' (InterSvyaz) located in 'Челябинск' (Chelyabinsk). The job requirements include a full working day, higher education, and full employment. The text states that InterSvyaz is looking for a WEB programmer and lists requirements such as responsibility for code, desire to learn new technologies, and preparedness for a trial period. It also mentions bonuses like official employment, a comfortable office in the North-West, fair pay, and regular qualification improvement.

40 000 – 80 000 руб.

ИНТЕРСВЯЗЬ

Челябинск

График работы: Полный рабочий день

Образование: Высшее

Занятость: Полная занятость

Компания «Интерсвязь» ищет в свои ряды WEB – программиста. Ты знаешь, что это такое; имеешь опыт работы, готов учиться и узнавать новое? Тогда ты точно нужен нам.

Мы просим не так уж и много:

- Умение нести ответственность за свой код. Как известно, мы в ответе за тех, кого написали;
- Огромное желание учиться и осваивать новые технологии;
- Готовность пройти испытательный срок, продолжительность которого будет зависеть от твоих навыков;

Рис. 1.8 — Вакансия web-программист с сайта chelyabinsk.zarplata.ru (часть 1).

Аналогично вышерассмотренным вакансиям тип занятости указан в первых пунктах. Также мы видим графу образование, которая тоже имеет немаловажный фактор.

На профессиональных навыков ТССС обязательно попадают:

- знание PHP и JavaScript;
- умение верстать веб-страницы;
- знание PHP-фреймворков (Yii будет большим плюсом);
- умение пользоваться системой контроля версий Git.

Всю эту совокупность навыков и качеств мы обменяем на следующие блага:

- Официальное трудоустройство, и все причитающиеся за это бонусы;
- Комфортный офис на Северо-западе;
- Достойная оплата труда.
- Регулярное повышение квалификации и обучение новым технологиям.

Благодарим всех соискателей, которые направят отклик на данную вакансию. Ваше резюме будет рассмотрено в течение ближайшего времени. Встречи или тел. интервью будут проведены только с кандидатами, максимально соответствующими требованиям требований.

Рис. 1.6 — Вакансия web-программист с сайта chelyabinsk.zarplata.ru (часть 2).

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ЮУрГУ-090301.2018.135

Лист
15

Таким образом, проанализировав множество резюме и вакансий, были выявлены метрики, которые, практически, полностью характеризуют претендента:

- Опыт работы.
- Тип занятости.
- Ключевые навыки.
- Требуемая заработная плата.
- Наличие и уровень образования.

На данный момент на рынке всемирной паутины присутствуют следующий варианты web-сервисов подбора кандидатов:

- «Оценка талантов»

Данный сервис разработан специалистами HeadHunter и предназначен для быстрого подбора персонала на открывшуюся вакансию. Можно выделить следующие недостатки:

- 1) Сроки подбора кандидата. Основная проблема данного сервиса, т.к. подбор осуществляется с помощью опросников к потенциальному кандидату, а затем вычисление эффективности на основе его ответов. Соответственно требуется не малое время для адекватного подбора;
- 2) Дороговизна услуги. Стоимость данной услуги имеет минимальную цену, которая, порой, превышает месячную заработную плату сотрудников данной специальности по региону.

На рисунке 1.4. представлены минимальные тарифы данного web-сервиса.

Изм.	Лист	№ докум.	Подпись	Дата



Стоимость найма и адаптации сотрудников для компании;
без использования инструментов оценки

40 250 рублей

Стоимость найма и адаптации сотрудников для компании;
с использованием инструментов оценки

37 000 рублей

Рис. 1.4 — Тарифы web-сервиса «Оценка талантов»

- «HR-SNIPER»

Данный сервис поможет написать вакансию в стиле компании, которая ищет кандидата. Вы отвечаете на вопросы, а система превращает ответы в «магическую» вакансию, привлекающую в 3,8 раза больше целевых кандидатов (по статистике сервиса). Очень удобный сервис, но тоже имеет ряд недостатков, которые были выявлены при личном использовании:

- 1) Сроки подбора кандидата. Данный сервис также использует анкеты с вопросами, что также увеличивает время работы сервиса;
- 2) Отсутствие эргономичности. Данный сервис использует однотипные, шаблонные тесты для соискателей. Это может отпугнуть ряд кандидатов, утомившихся проходить одинаковые тесты.

На рисунке 1.5 изображен скриншот из сервиса «HR-SNIPER».

Изм.	Лист	№ докум.	Подпись	Дата	Лист	ЮУрГУ-090301.2018.135	17
------	------	----------	---------	------	------	-----------------------	----

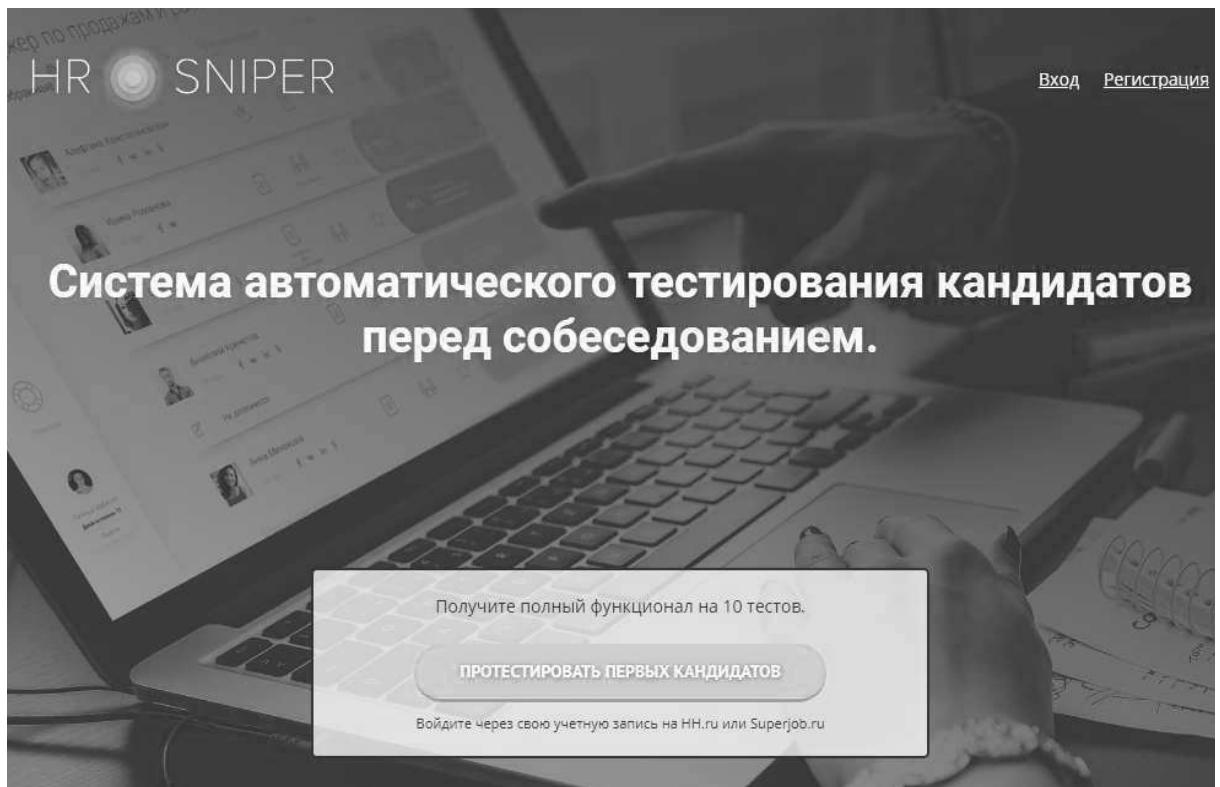


Рис. 1.5 — «HR-SNIPER»

- «FriendWork Recruiter»

Онлайн-сервис для автоматизации рекрутинга позволяет в один клик добавить резюме из работных сайтов, соцсетей и почты в базу. Был проведен опытный эксперимент использования, в ходе которого были выявлены следующие недостатки:

- 1) Отсутствие своевременной технической поддержки;
- 2) Отсутствие возможности размещений вакансий в режиме online;
- 3) Примитивность алгоритмов оценки.

На рисунке 1.6 изображен скриншот из сервиса «FriendWork Recruiter».

Изм.	Лист	№ докум.	Подпись	Дата	Лист	ЮУрГУ-090301.2018.135	18
------	------	----------	---------	------	------	-----------------------	----

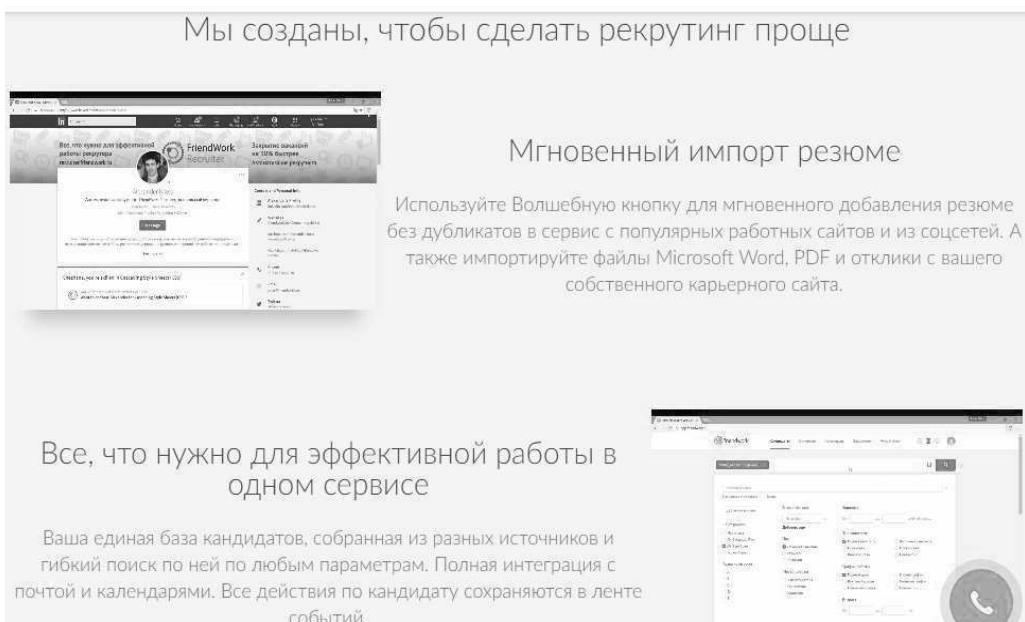


Рис. 1.6 — «FriendWork Recruiter»

Выводы по разделу один:

Таким образом, безупречного сервиса по подбору кандидатов на заданную вакансию на данный момент нет. В разрабатываемом проекте необходимо учесть недостатки данных аналогов.

2 ТРЕБОВАНИЯ К РАЗРАБАТЫВАЕМОМУ ПРИЛОЖЕНИЮ

2.1 Требования к системе в целом

2.1.1 Требования к структуре и функционированию системы

- Данные СПАР должны храниться на одном сервере, к которому делегирован домен второго уровня, для доступа по сети Интернет.
- Работа с системой должна осуществляться через веб-интерфейс.

2.1.2 Требования к надежности системы

- Система должна стablyно работать в течение рабочего времени (с 11:00 до 20:00 мск). Если сервер возвращает код состояния HTTP отличный от 200 — это считается нарушением в работе.

Изм.	Лист	№ докум.	Подпись	Дата

2.2 Функциональные требования

- Обеспечить возможность анализа сайтов hh.ru, zarplata74.ru с целью нахождения наилучших кандидатов по выявленным предприятием метрикам;
- Обеспечить возможность динамического регулирования кол-ва характеризующих метрик;
- Обеспечить детальную информацию о кандидате (значения заданных метрик, контактные данные);
- Процентная визуализация соответствия соискателя заданной вакансии;
- Обеспечить наилучший подбор резюме на заданную вакансию;
- Анализ должен обеспечиваться на высоконагруженных сайтах по поиску резюме (до 2000 претендентов);
- Обеспечить возможность установления значений метрик по заданной вакансии;
- Обеспечить возможность поиска с использованием логических выражений на базе алгебры логики.

2.3 Требования к видам обеспечения

2.3.1 Требования к информационному обеспечению

- Хранение данных в системе должно быть построено на основе реляционной модели данных.
- Информация в базе данных системы должна сохраняться при возникновении аварийных ситуаций, связанных со сбоями электропитания.
- Система должна иметь бесперебойное электропитание, обеспечивающее её нормальное функционирование в течение 5 минут в случае отсутствия внешнего энергоснабжения, и 5 минут дополнительно для корректного завершения всех процессов.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ЮУрГУ-090301.2018.135

Лист
20

- Резервное копирование данных должно осуществляться на регулярной основе, в объемах, достаточных для восстановления информации в подсистеме хранения данных. Ответственность за резервное копирование несет Заказчик, однако инструмент резервного копирования должен быть встроен в систему Разработчиком.

- К контролю данных предъявляются следующие требования: система должна протоколировать все события, связанные с изменением хранимых данных, и иметь возможность в случае сбоя в работе восстанавливать свое состояние, используя ранее запротоколированные изменения данных.

2.4 Требования к лингвистическому обеспечению

- При реализации системы должны применяться следующие языки программирования: на серверной части MySQL и PHP, на клиентской части HTML5, jQuery, CSS3 и Twitter Bootstrap.

2.5 Варианты использования системы

Проведя анализ требований, была сформулирована модель вариантов использования разрабатываемой системы.

Можно выделить следующих акторов (действующий субъект):

1) Пользователь. Представляет собой работодателя, который взаимодействует с системой через интерфейс. Пользователь может загружать в систему эталонное резюме, получать результаты анализа в удобном для просмотра виде;

2) Искусственный интеллект. Представляет собой программу, которая ищет резюме и анализирует их на соответствие заданной вакансии.

На рисунке 2.1 расположена диаграмма прецедентов сервиса СПАР:

Изм.	Лист	№ докум.	Подпись	Дата	Лист	ЮУрГУ-090301.2018.135	21



Рисунок 2.1 Диаграмма прецедентов СПАР.

Можно выделить следующие варианты использования приложения:

- загрузить вакансию. Пользователь может загрузить в базу данных через специальный интерфейс в любом удобном для себя формате открывшуюся вакансию;
- получить результат. Пользователь в любой момент может получить актуальную информацию с заданного ресурса;
- изменить настройки. Пользователь может изменить настройки, параметры анализа, предусмотренные системой.

Выводы по разделу два:

Приведенные в данной главе требования были приняты к сведению как основные, которые необходимо реализовать в сервисе поиска и аналитики резюме «СПАР».

Изм.	Лист	№ докум.	Подпись	Дата

3 АРХИТЕКТУРА СЕРВИСА

В данном разделе будут рассмотрены элементы приложения, их логическое расположение, функционал.

Веб-сервис содержит в себе несколько компонентов. Компонент – элемент системы, который содержит в себе классы, объекты, функционал, тем самым разделяя систему на логические разделы. В данном проекте используется шаблон проектирования систем MVC. Дословный перевод модель-представление-контроллер обозначает разделение архитектуры на логические части, каждая из которых отвечает за определенный функционал, что позволяет гибко и независимо изменять отдельные части системы:

- 1) Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.
- 2) Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели.
- 3) Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

Для визуального представления данной архитектуры представлен рисунок 3.1:

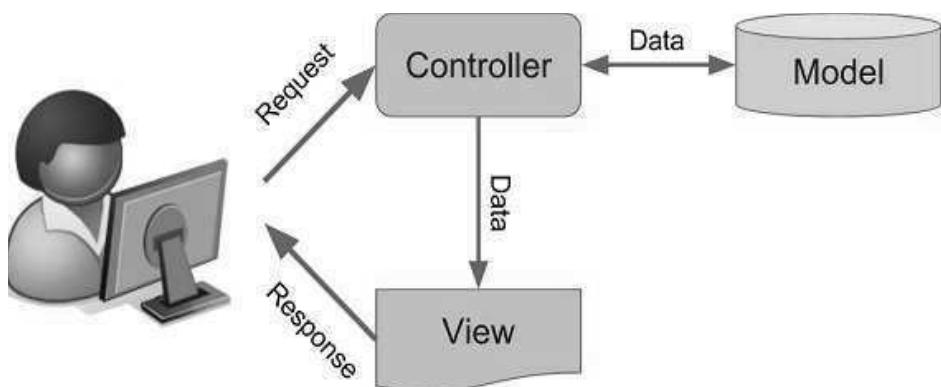


Рисунок 3.1 Архитектура MVC.

Архитектура приложения состоит из 3 основных компонентов:

Изм.	Лист	№ докум.	Подпись	Дата

- 1) PostController – компонент, отвечающий за отправку данных с клиента на сервер и обратно. Также он управляет основным ходом работы приложения, передавая команды в другие компоненты.
- 2) ParseController – компонент, взаимодействующий с базой данных, всеми моделями, представлениями, которые отвечают за логику получения и вывода данных.
- 3) AI – компонент, отвечающий за обработку данных с помощью искусственной нейронной сети.

На рисунке 3.2 представлена диаграмма компонентов приложения:

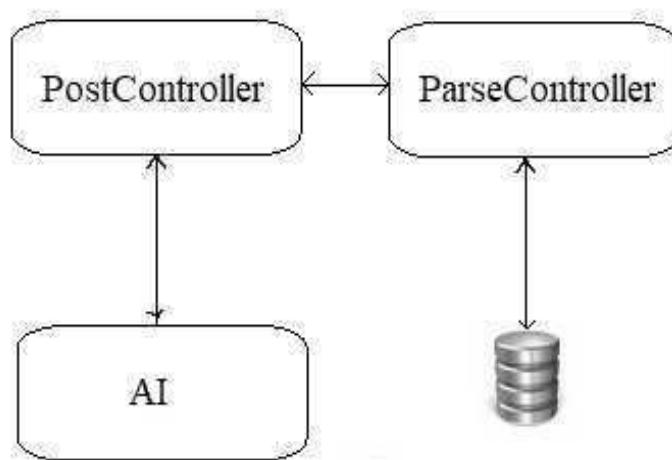


Рисунок 3.2 Диаграмма компонентов.

Рассмотрим подробнее компоненты приложения.

3.1 Компонент PostController

Данный компонент отвечает за связь клиентского интерфейса с серверным и выполняет такие функции:

- Анализ входных данных
- Компоновка выходных данных
- Подготовка и представление входных данных
- Выбрасывание исключений и ошибок
- Роутинг (маршрутизация) веб-сервиса

Изм.	Лист	№ докум.	Подпись	Дата

- Управление панелью администратора (разрабатывается)

Компонент содержит в себе один класс, именуемый Admin, который и выполняет все вышеперечисленные функции. Данный класс условно можно разделить на 3 логические части:

1) часть, отвечающая за различные проверки, ошибки, исключения, чтобы при нахождении оных вся система не рушилась, например:

- 404 ошибка
- Отсутствие данных в бд
- Некорректный запрос
- Попытка загрузки PHP скрипта на сервер и исполнение его там
- SQL - инъекции
- DDOS атака

2) часть, отвечающая за пересылку данных и содержит в себе следующий функционал:

- Формирование данных для отправки
- Формирование представлений из входных данных
- Правила задания метрик

3) часть, отвечающая за маршрутизацию. В ней содержится правила рендеринга, пополняемая константа маршрутов и методы для редиректов на соответствующие страницы, экшены.

3.2 Компонент ParseController

Данный компонент обеспечивает связь системы с базой данных, операции CRUD, настройка резервного копирования. Также он осуществляет парсинг сайтов с резюме

Компонент содержит в себе контроллер для связи с базой данных, который посредством двух моделей взаимодействует с бд:

Изм.	Лист	№ докум.	Подпись	Дата

1) Vacancy. Модель отвечает исключительно за работу с настройкой открывшейся вакансии и позволяет:

- Добавлять вакансию
- Редактировать вакансию
- Удалять вакансию
- Получать вакансию

2) ParserVac. Модель, которая работает с API различных сайтов трудоустройства и подготавливает данные для передачи их модели Vacancy. Модель реализует паттерн «стратегия», что позволяет ей работать с различными web-биржами труда.

3.3 Компонент ML

Данный компонент отвечает за обработку данных, которые переслал ему компонент ParseController, посредством библиотеки PHP-ML - Machine Learning library for PHP. Компонент реализует две основные задачи:

- Train (тренировать). Метод библиотеки для «обучения» нейронной сети.
- Predict (прогнозировать). Метод библиотеки анализирующий результат.

Приложение состоит из нескольких модулей, которые распределены в соответствии с уровнями стиля MVP (Рисунок 3.3).

Data уровень состоит из двух частей. Одна часть – это «черновые» классы, которые так или иначе обеспечивают получение или сохранение данных.

Например, классы работы с сетью, базы данных, кеша и т.д. Вторая часть является репозиториями, которые являются «прослойкой» между бизнес-логикой и моделью.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

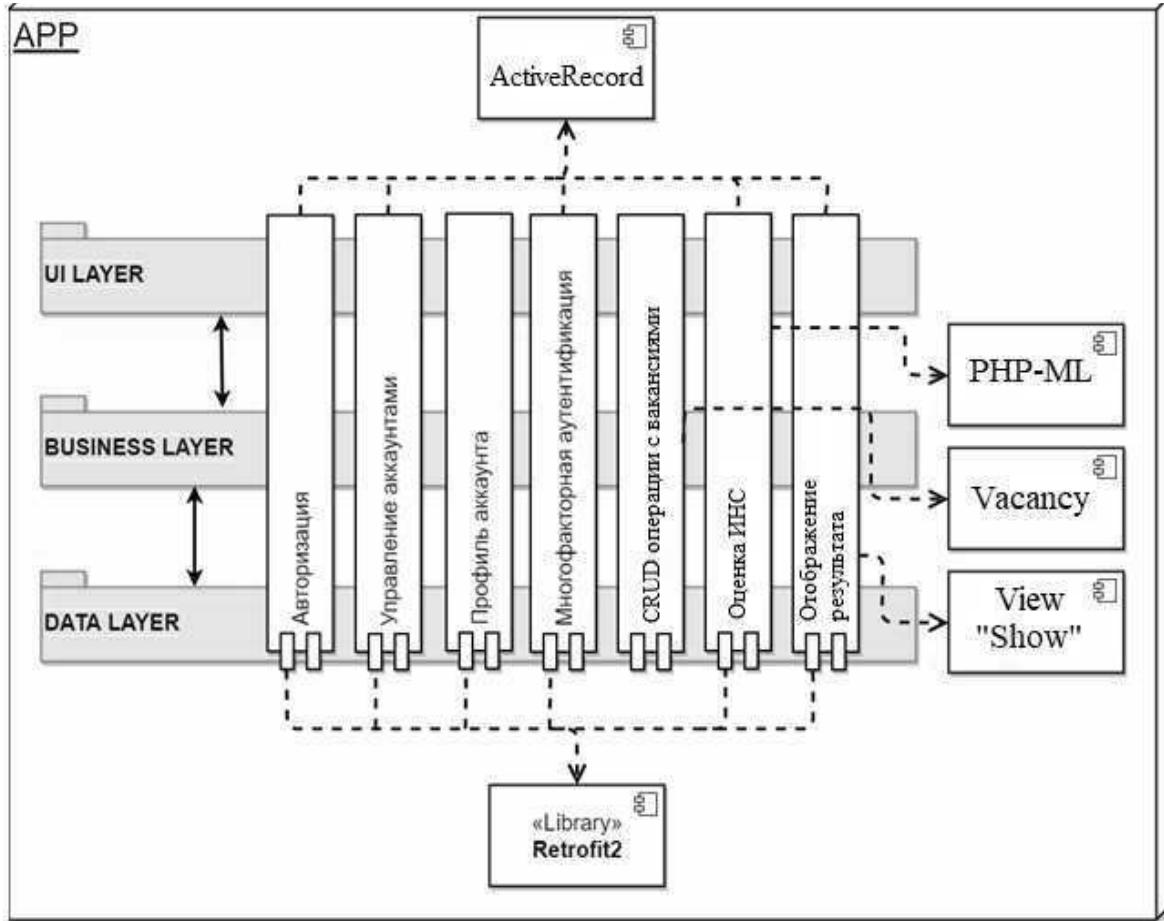


Рисунок 3.3 – Архитектура приложения в контексте MVP.

Таким образом, реализуется слабая зависимость логики от сущностей, что позволяет, с применением интерфейсов для репозитория, подменять реализацию того или иного «чернового» класса. Примером положительного момента такого разделения является замена ORM для работы с базой данных.

Таким образом, слой Data ответственен за получение и сохранение данных. Этот слой никаких ошибок не обрабатывает, объекты моделей данных не объединяет.

Уровень бизнес-логики. Обычно, каждый интерактор на данном уровне – это одна большая бизнес-фича, а методы – это конкретный бизнес-кейс.

Интерфейс интерактора открыт наружу следующему уровню – UI. Одна из обязательных функций данного уровня – это преобразования модели данных,

полученных из репозитория в модели данных для представления. Интеракторы обрабатывают ошибки: некоторые из них глушатся, некоторые «пробрасываются наверх». Таким образом, интерактор получает данные из репозитория, преобразует ответ в модель данных UI уровня и отдает их презентеру.

Следующий уровень – UI уровень. Также, как и уровень Data, разделен на две части: представление и презентер. Презентер является обработчиком конкретного представления, а именно:

- Обрабатывает взаимодействие пользователя с экраном;
- Показывает ошибки;
- Обновляет данные и т.д.

Таким образом, представление только передает действия пользователя и делает то, что говорит ему презентер.

Представление при создании создает ссылку на презентер, типом объекта которого является интерфейс презентера, и, при надобности, вызывает методы интерфейса. Так как презентер жестко прикреплен к представлению, то необходимо синхронизировать презентер с жизненным циклом окна с помощью методов bindView и unbindView.

Так как все уровни связаны между собой с помощью интерфейсов, это позволяет абстрагировать уровни друг от друга, что позволяет легко и без лишних действий подменять существующую функциональность, а также добавлять новую.

Выводы по разделу три:

В данном разделе были рассмотрены основные архитектурные компоненты приложения, на которых оно базируется. Были приведены основные задачи, которые решают данные компоненты, а также их взаимодействие между собой.

Изм.	Лист	№ докум.	Подпись	Дата

4 ВЫБОР ТЕХНОЛОГИИ ФОРМИРОВАНИЯ ОЦЕНКИ СОВПАДЕНИЯ

В данный момент существует множество вариантов формирования оценки результата. Рассмотрим подходящие для нашей задачи технологии, выделим плюсы и минусы, и на основе этих данных сделаем выбор.

4.1 Корреляционный анализ - метод, позволяющий обнаружить зависимость между несколькими случайными величинами.

Допустим, проводится независимое измерение различных параметров у одного типа объектов. Из этих данных можно получить качественно новую информацию - о взаимосвязи этих параметров.

Например, на рисунке 4.1 покажем ожидаемую заработную плату и опыт работы человека, каждое измерение представлено точкой в двумерном пространстве:

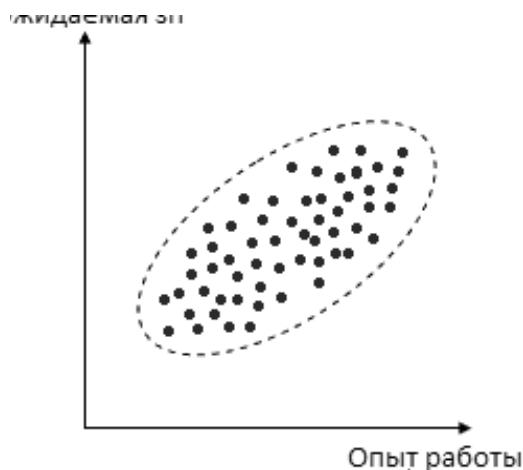


Рисунок 4.1. Зависимость ожидаемой заработной платы от опыта работы

Несмотря на то, что величины носят случайный характер, в общем наблюдается некоторая зависимость - величины коррелируют.

В данном случае это положительная корреляция (при увеличении одного параметра второй тоже увеличивается). Возможны также отрицательная корреляция и ее отсутствие.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

Важной характеристикой совместного распределения двух случайных величин является ковариация (или корреляционный момент). Ковариация является совместным центральным моментом второго порядка. Ковариация определяется как математическое ожидание произведения отклонений случайных величин:

$$\text{cov}_{XY} = \mathbf{M}[(X - \mathbf{M}(X))(Y - \mathbf{M}(Y))] = \mathbf{M}(XY) - \mathbf{M}(X)\mathbf{M}(Y),$$

где \mathbf{M} — математическое ожидание.

Методами корреляционного анализа решаются следующие задачи:

- Взаимосвязь. Есть ли взаимосвязь между параметрами?
- Прогнозирование. Если известно поведение одного параметра, то можно предсказать поведение другого параметра, коррелирующего с первым.
- Классификация и идентификация объектов. Корреляционный анализ помогает подобрать набор независимых признаков для классификации.

Преимущества:

- 1) Позволяет получить за очень короткое время множество данных для значительного числа испытуемых.
- 2) Метод можно применять в ряде особых случаев, в которых экспериментальный подход сопряжен с трудностями или даже невозможен (главным образом по этическим соображениям)
- 3) Позволяет получать информацию, основанную на более разнообразных выборках и более близкую к существующей в обществе реальности

Недостатки:

- 1) Не позволяет решить проблему, связанную с возможной интерпретацией зависимости, существующей между переменными.
- 2) Не может дать окончательное заключение о том, что две переменные причинно связаны.
- 3) Не дает объективных результатов при большом количестве переменных.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

4.2 Регрессионные деревья - средство поддержки принятия решений, использующееся в статистике и анализе данных для прогнозных моделей.

Структура дерева представляет собой «листья» и «ветки». На ребрах («ветках») дерева решения записаны атрибуты, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах — атрибуты, по которым различаются случаи.

Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение. Подобные деревья решений широко используются в интеллектуальном анализе данных. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной на основе нескольких переменных на входе.

В интеллектуальном анализе данных, деревья решений могут быть использованы в качестве математических и вычислительных методов, чтобы помочь описать, классифицировать и обобщить набор данных, которые могут быть записаны следующим образом:

$$(x, Y) = (x_1, x_2, x_3 \dots x_k, Y)$$

Зависимая переменная Y является целевой переменной, которую необходимо проанализировать, классифицировать и обобщить. Вектор X состоит из входных переменных X_1, X_2, X_3 и т. д., которые используются для выполнения этой задачи.

Пример использования представлен на рисунке 4.2:

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------



Преимущества:

- 1) Прост в понимании и интерпретации. Люди способны интерпретировать результаты модели дерева принятия решений после краткого объяснения.
- 2) Не требует подготовки данных. Прочие техники требуют нормализации данных, добавления фиктивных переменных, а также удаления пропущенных данных.
- 3) Позволяет работать с большим объемом информации без специальных подготовительных процедур. Данный метод не требует специального оборудования для работы с большими базами данных.

Недостатки:

- 1) Алгоритмы не могут обеспечить оптимальность всего дерева в целом.
- 2) Те, кто изучает метод дерева принятия решений, могут создавать слишком сложные конструкции, которые недостаточно полно представляют данные. Данная проблема называется переобучением. Для того, чтобы избежать данной проблемы, необходимо использовать Метод «регулирования глубины дерева».

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

3) Существуют концепты, которые сложно понять из модели, так как модель описывает их сложным путём. Данное явление может быть вызвано проблемами XOR, четности или мультиплексарности. В этом случае мы имеем дело с непомерно большими деревьями. Существует несколько подходов решения данной проблемы, например, попытка изменить репрезентацию концепта в модели (составление новых суждений), или использование алгоритмов, которые более полно описывают и репрезентируют концепт (например, метод статистических отношений, индуктивная логика программирования).

4) Для данных, которые включают категориальные переменные с большим набором уровней (закрытий), больший информационный вес присваивается тем атрибутам, которые имеют большее количество уровней.

4.3 Искусственные нейронные сети – это одно из направлений исследований в области искусственного интеллекта, основанное на попытках воспроизвести нервную систему человека. А именно: способность нервной системы обучаться и исправлять ошибки, что должно позволить смоделировать, хотя и достаточно грубо, работу человеческого мозга.

Нейронные сети, в отличие от статистических методов многомерного классификационного анализа, базируются на параллельной обработке информации и обладают способностью к самообучению, то есть получать обоснованный результат на основании данных, которые не встречались в процессе обучения.

Преимущества:

1) Нейрокомпьютеры особенно эффективны там, где нужно подобие человеческой интуиции, в частности к таким задачам относятся принятие

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

решений в процессе оценки финансового состояния некоторого экономического объекта.

2) Гибкость структуры нейронных сетей позволяет различными способами комбинировать простые составляющие нейрокомпьютеров – нейроны и связи между ними. За счет этого один нейрокомпьютер можно применять для решения различных задач, зачастую, никак не связанных между собой.

3) Нейронные сети позволяют создать эффективное программное обеспечение для высокопараллельных компьютеров. Создавая математическое обеспечения на базе нейронных сетей, можно для широкого класса задач решить проблему эффективности одновременного решения параллельных задач. Кроме того, параллельная обработка информации обеспечивает высокую скорость решения задач.

4) Решение задач в условиях неопределенности – способность обучения нейронной сети позволяет решать задачи с неизвестными закономерностями и зависимостями между входными и выходными данными, что позволяет работать с неполными данными. Кроме того, взаимоотношения между величинами заранее не устанавливаются, поскольку метод предусматривает изучение существующих взаимосвязей на готовых моделях.

5) Устойчивость к шумам во входных параметрах – нейронная сеть может самостоятельно определять неинформативные для анализа параметры и производить их отсев, в связи с чем, пропадает необходимость дополнительного анализа информационного вклада входных данных.

6) Адаптивность к изменениям окружающей среды – нейронные сети могут быть переучены в новых условиях окружающей среды, описываемых незначительными колебаниями параметров этой среды.

Недостатки:

1) Сложность разработки.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

- 2) Дороговизна исследований.
- 3) Свежесть идеи (мало документации).

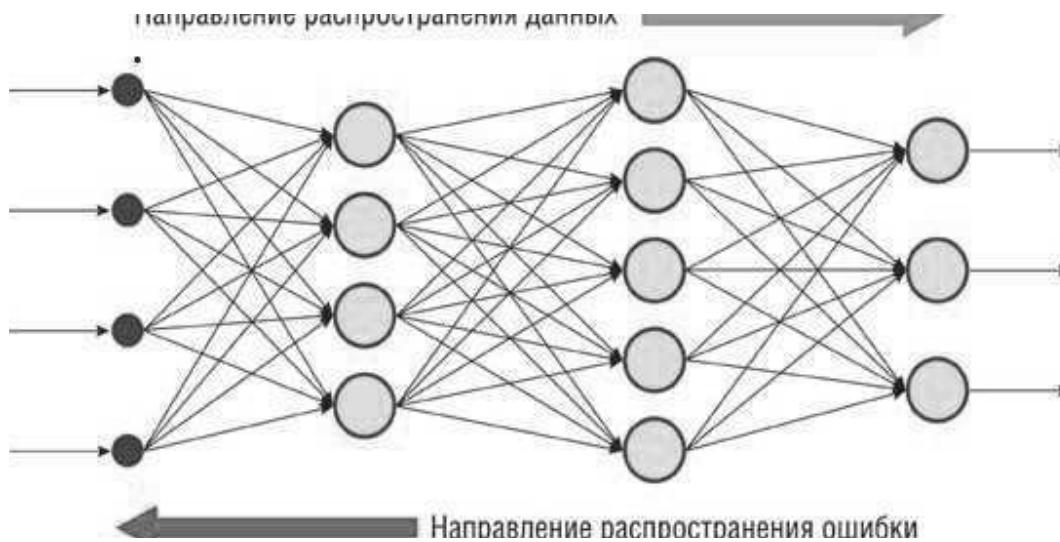


Рисунок 4.3 Схематичное изображение искусственной нейронной сети.

Раннее аппаратные средства могли реализовать лишь примитивную ИНС, состоящую из малого количества слоев и решать с помощью нее мелкие задачи, однако в наше время техника позволяет создать и использовать нейронные сети для решения сложнейших алгоритмов, сильно уменьшая сложность решения.

Выводы по разделу четыре:

Таким образом, был сделан выбор в пользу искусственной нейронной сети. Выбор аргументирован возможностью решения задач в условиях неопределенности, объективностью этих решений и независимостью от условий окружающей среды.

5 ОБОСНОВАНИЕ И ВЫБОР ПРОГРАММНЫХ СРЕДСТВ, НЕОБХОДИМЫХ ДЛЯ РЕАЛИЗАЦИИ

Существует множество вариантов разработки web-сервисов. Как правило, средства разработки определяются конечными целями проекта и его техническими требованиями. Перечислим самые важные критерии:

1. Размер и тип проекта

Изм.	Лист	№ докум.	Подпись	Дата

2. Сложность проекта
3. Скорость разработки
4. Стоимость специалистов
5. Доступность специалистов
6. Доступные инструменты разработки
7. Наличие готовых решений
8. Гибкость решения
9. Наличие широкого сообщества
10. Отказоустойчивость решения
11. Тренд его развития
12. Наличие подробной документации
13. Стоимость поддержки
14. Требования к нагрузкам
15. Требования к безопасности
16. Кроссплатформенность
17. Возможности интеграции с другими решениями

Исходя из этих критериев, прежде всего определимся со стеком технологий, который будет использоваться для реализации проекта. Для этого рассмотрим самые популярные варианты, которые активно используются на рынке и успели себя зарекомендовать.

5.1 Серверная часть:

1. PHP — его используют в основном для простых и средних проектов. Очень много коробочных решений. С выходом последней версии языка под номером 7, он получил действительно мощные возможности.
2. Python — современный язык, разработка на нем быстрая и качественная. Используют его для средних и больших проектов.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

3. Ruby — современный язык, разработка на нем также быстрая. Его используют в основном для разработки простых и средних проектов, часто разрабатывают стартапы.

4. Java — разработка на нем очень долгая и дорогая. Его используют в основном для больших проектов со специфическими требованиями.

5. C# — аналог Java, также используют для больших проектов, часть в сфере Fin Tech.

6. JS — очень быстро развивается, тренд последних лет. Огромное количество наработок, и можно писать все, что угодно, даже игры. Его используют для средних и больших проектов, но действительно мощные возможности этот язык получил недавно, потому примеров больших проектов пока мало, специалисты самые дорогие, и найти их сложнее всего.

5.2 Клиентская часть:

1. JavaScript - мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили.

2. HTML+CSS

5.3 СУБД:

1. MySQL –

Преимущества:

- Простота в работе
- Богатый функционал
- Бесплатное использование
- Масштабируемость
- Скорость

Недостатки:

- Проблемы с надежностью
- Медленная разработка

Изм.	Лист	№ докум.	Подпись	Дата

2. PostgreSQL

Преимущества:

- Бесплатное использование
- Большое количество дополнений
- Расширение функционала

Недостатки:

- Низкая производительность
- Проблемы с размещением на хостинге
- Малое сообщество

3. MS SQL

Преимущества:

- Большое количество возможностей вialectе языка запросов SQL

Недостатки:

- Низкая производительность
- Неудобный интерфейс
- Платное использование

5.4 ИНС

Искусственная нейронная сеть (ИНС) – это математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования сетей нервных клеток живого организма. Существует множество библиотек и фреймворков для разных языков программирования, позволяющих построить нейросеть с требуемыми параметрами. Рассмотрим некоторые из них:

1) Synaptic.js

Изм.	Лист	№ докум.	Подпись	Дата

Synaptic.js – это мощный фреймворк для построения нейросетей практически любой сложности, написанный на языке JavaScript. Он имеет открытый исходный код и распространяется по лицензии MIT.

Synaptic.js имеет несколько встроенных архитектур, что позволяет очень быстро сравнивать и изучать различные алгоритмы машинного обучения. Имеются хорошо написанное введение в нейронные сети, ряд практических примеров и множество других учебников по машинному обучению.

Но Synaptic.js имеет серьезный недостаток: так как в фреймворке отсутствует серверная часть и все функции исполняются в браузере, на ее запуск тратятся ресурсы компьютера пользователя.

2) TensorFlow

TensorFlow - фреймворк для глубокого машинного обучения, разрабатываемый в Google Brain. С помощью него можно создать нейросеть практически любого требуемого типа. Реализация нейросети, которая будет генерировать тексты - довольно «тяжелая» задача для сервера, однако, Tensorflow, в отличии от других библиотек, может работать как на CPU, так и GPU, что существенно ускорит работу и обучение нейросети. Данный фреймворк отлично подходит для генерации какой-либо информации.

3) PHP ML

PHP ML – это фреймворк для работы с искусственным интеллектом. Фреймворк разрабатывается свободным сообществом разработчиков на языке программирования PHP. PHP ML имеет довольно хорошую оптимизацию, описание встроенных классов и методов. Данный фреймворк имеет открытый исходный код, а также является одним из самых развивающихся на данный момент.

Изм.	Лист	№ докум.	Подпись	Дата

Кроме рассмотренных вариантов, существует бесчисленное множество технологий для разработки web-сервисов, но большинство из них имеют существенные недостатки.

Выводы по разделу пять:

Таким образом, был сделан выбор в пользу стека технологий:

1. PHP (фреймворк Yii2)
2. JavaScript + HTML + CSS
3. Mysql

Выбор аргументирован простотой данных технологий, их бесплатным использованием, техническими требованиями к данному приложению. Также весомое значение имеет большое количество русской документации, большого сообщества.

6 РЕАЛИЗАЦИЯ СИСТЕМЫ

6.1 Реализация компонента ParseController

Парсер веб-сайтов отвечает за сбор данных по заданным метрикам с определенного ресурса. Блок-схема общего алгоритма работы данной части системы представлена на рис. 6.1

В начале работы парсер скачивает код различных типов страниц сайта (главная, каталог, карточка товара) вместе со всеми зависимостями (картинки, стили и скрипты) и сохраняет их во временные папки на сервере. Это сделано для того, чтобы снизить количество обращений к удаленному ресурсу при разборе кода, соответственно ускорив работу парсера.

Для разбора исходного кода сайтов была использована библиотека SimpleHtmlDom, превращающая html-код в иерархическую структуру из объектов. В SimpleHtmlDom также есть методы, позволяющие искать содержимое тегов и атрибутов на странице с помощью масок и регулярных выражений.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

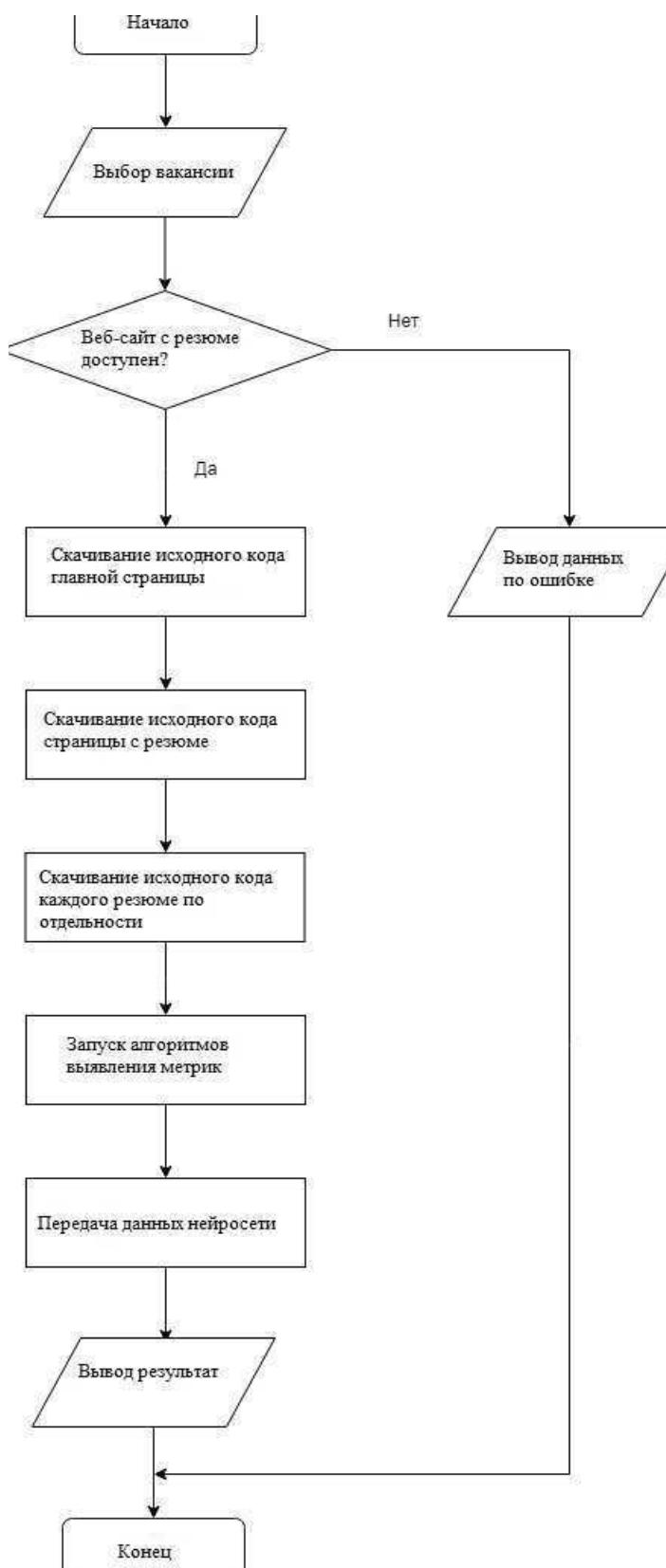


Рис. 6.1 – Общий алгоритм работы парсера

Парсер состоит из нескольких классов, которые отвечают за сбор отдельных групп метрик.

1. Vacancy

Класс отвечает исключительно за работу с метриками открывшейся вакансии. Класс используя данные о вакансии проводит поиск по сайтам. Основные методы данного класса:

1.1 getSalary

Метод, получающий данные о желаемой заработной плате претендента. Единственным параметром данной функции является \$id, в который загружается структура DOM-дерева (DOM – это представление документа в виде дерева объектов, доступное для изменения) рассматриваемого кандидата. С помощью метода find(), который является частью библиотеки SimpleHtmlDom, происходит поиск тегов span с атрибутом class «resume-block_salary», в которых строка. Поиск осуществляется по маске, в которой указывается название тега (span или любой произвольный тег), атрибут в котором содержится нужный класс и строка, которая должна содержаться в атрибуте (в данном примере это расширение файла «resume-block_salary»). Также метод проверяет указал ли претендент заработную желаемую заработную плату и, если не указал, сообщает об этом. Исходный код метода представлен на рис. 6.2:

```
private function getSalary($id)
{
    $salary = $id->find('span.resume-block_salary')[0]->plaintext;
    if ($salary == " ") $salary = " Не указан";
    return $salary;
}
```

Рис. 6.2 – Листинг метода 1.1 getSalary

1.2 getExperience

Изм.	Лист	№ докум.	Подпись	Дата

Метод предоставляет информацию об опыте работы претендента. Механизм получения данных схож с методом getSalary. Также присутствует проверка на отсутствие опыта работы. Листинг метода представлен на рис. 6.3;

```
private function getExperience($id)
{
    $experience = $id->find('.resume-block__title-text')[1]->
        plaintext;
    if ($experience != '') {
        $experience = str_replace(['Опыт работы'], '', $experience);
    }
    else {
        $experience = "Без опыта";
    }
    return $experience;
}
```

Рис. 6.3 – Листинг метода getExperience

1.3 getSkills

Метод получающий список навыков претендента. Поиск ведется с помощью масок способом, описанным выше. После получения массива объектов, навыки записываются в массив путем перебора. Исходный код метода представлен на рис. 6.4;

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

```

private function getSkills($id)

{
    $skillsArr = [];
    $skills = $id->find('.Bloko-TagList-Text');
    foreach ($skills as $s) {
        $skillsArr[] = $s->plaintext;
    }
    return $skillsArr;
}

```

Рис. 6.4 – Листинг метода 1.1 getSkills

2. Employee

Класс, реализующий CRUD операции (создание, получение, обновление и удаление данных). Класс наследуется от ActiveRecord (специальный класс для работы с базой данных). Основные методы данного класса:

2.1 addEmployee

Этот метод добавляет нового претендента в базу данных. Принимает входным параметром массив с данным о претенденте. Создается новый объект ActiveRecord. К его свойствам, соответствующим полям таблицы Employee, присваиваются значения пришедшие от парсера. Метод обернут в структуру try – catch, что позволяет отлавливать внештатные ситуации и выбрасывать исключения. Листинг метода представлен на рис. 6.5:

Изм.	Лист	№ докум.	Подпись	Дата

```

public static function addEmployee($post)
{
    try {
        $employee = new Employee();
        $employee->firstName = $post['firstName'];
        $employee->lastName = $post['lastName'];
        $employee->href = $post['href'];
        $employee->salary = $post['salary'];
        $employee->experience = $post['experience'];
        $employee->skills = $post['skills'];
        $employee->save();
        return json_encode(['status' => 'ok']);
    } catch (Exception $e) {

        throw new Exception("not add", 500);
    }
}

```

Рис. 6.5 – Листинг метода addEmployee

2.2 getEmployee

Метод, получающий информацию о конкретном работнике. Поиск в базе данных осуществляется по уникальному идентификатору. Листинг данного метода представлен на рис. 6.6:

```

public static function getEmployee($id)
{
    $employee = Employee::find()
        ->where(['id' => $id])
        ->one();
    return json_encode($employee);
}

```

Рис. 6.6 – Листинг метода 2.2 getEmployee

Изм.	Лист	№ докум.	Подпись	Дата

2.3 updateEmployee

Этот метод добавляет обновляет информацию по существующему претенденту в базе данных. Принимает входным параметром массив с новыми данным о претенденте. Листинг метода представлен на рис. 6.7:

```
public static function updateEmployee($post)
{
    try {
        $employee = Employee::findOne($post['id']);
        $employee->firstName = $post['firstName'];
        $employee->lastName = $post['lastName'];
        $employee->href = $post['href'];
        $employee->salary = $post['salary'];
        $employee->experience = $post['experience'];
        $employee->skills = $post['skills'];
        $employee->save();
        return json_encode(['status' => 'ok']);
    } catch (Exception $e) {
        throw new Exception("not add", 500);
    }
}
```

Рис. 6.7 – Листинг метода 2.3 updateEmployee

2.4 deleteEmployee

Этот метод удаляет претендента из базы данных по уникальному id. Листинг метода представлен на рис. 6.7:

```
public static function deleteEmployee($id)
{
    try {
        $employee = Employee::findOne($id);
        $employee->delete();
        return json_encode(['status' => 'ok']);
    } catch (Exception $e) {
        throw new Exception("not add", 500);
    }
}
```

Рис. 6.7 – Листинг метода 2.4 deleteEmployee

Изм.	Лист	№ докум.	Подпись	Дата

6.2 Реализация нейронной сети

Задачей нейронной сети в разрабатываемом программном продукте является анализ собранных парсером данных и вычисление значения предполагаемой целесообразности. Нейронная сеть представляет собой совокупность нейронов, которые составляют слои. В каждом слое нейроны между собой никак не связаны, но связаны с нейронами предыдущего и следующего слоев. Информация поступает с первого на второй слой, со второго — на третий и т.д.

Количество слоев и нейронов в них определяют точность и достоверность получаемых результатов при решении задач, т. е. чем больше слоев и нейронов на каждом слое — тем меньше ошибок и выше надежность работы сети. Однако, если построить слишком большую сеть, то можно столкнуться с уменьшением производительности и увеличением сложности модели. Блок-схема общего алгоритма работы данной части системы представлена на рис. 6.13.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------



Рис. 6.13 – Общий алгоритм работы нейросети

Работа с нейросетью осуществляется посредством трех основных классов:

6.2.1 NeuroNetwork

Данный класс инкапсулирует работу с библиотекой PHP ML. Перечень методов:

1. __construct

Конструктор класса, отвечающий за создание экземпляра нейросети (объект класса LeastSquares). Класс LeastSquares является частью библиотеки PHP ML и представляет собой реализацию нейросети, построенной по модели с множественной линейной регрессией.

Изм.	Лист	№ докум.	Подпись	Дата

2. train

Отвечает за обучение нейросети;

3. predict

Отвечает за получение значения рассчитанной конверсии.

Листинг класса NeuroNetwork представлен на рис. 6.14;

```
1 <?php
2
3 require_once __DIR__ . '/vendor/autoload.php';
4
5 class NeuroNetwork
6 {
7     public function __construct($config)
8     {
9         $this->config = $config;
10
11         $this->networkModel = new Phpml\Regression\LeastSquares();
12     }
13
14     public function train(array $samples, array $targets)
15     {
16         return $this->networkModel->train($samples, $targets);
17     }
18
19     public function predict(array $data)
20     {
21         return $this->networkModel->predict($data);
22     }
23 }
```

Рис. 6.14 – Листинг конструктора класса NeuroNetwork

6.2.2 GetTrainingData

Класс, осуществляющий поиск данных для тренировки нейросети. В задачи этого класса входит формирование списка резюме полностью удовлетворяющих вакансии. Класс содержит единственный метод runParser.

Изм.	Лист	№ докум.	Подпись	Дата

6.2.3 NeuroLearn

Задачей этого класса является обучение нейронной сети. Класс содержит открытую сессию `tf.Session`, который загружает обучающие выборки, созданные объектами класса `GetTrainingData` в нейронную сеть. Листинг класса представлен на рис. 6.15.

```
1 <?php
2
3 class NeuroLearn
4 {
5     $this->trainingData = [];
6
7     public function goTrain(string $query)
8     {
9         if ($dh = opendir('sites')) {
10
11             while (($file = readdir($dh)) !== false){
12                 $data = json_decode(file_get_contents('sites/'.$file));
13
14                 $this->trainingData[] = $data;
15
16                 unset('sites/'.$file);
17             }
18
19             closedir($dh);
20
21             if ($this->trainingData) {
22                 $network = new NeuroNetwork();
23
24                 return $network->train($this->trainingData);
25             }
26
27             return null;
28         }
29     }
30 }
```

Рис. 6.15 – Листинг класса `NeuroLearn`

6.3 Реализация базы данных

База данных должна хранить в себе следующие данные: информацию об эталонной вакансии, метрики найденных кандидатов и состояния нейронной сети (так называемые «веса», которые сохраняются в JSON формат и позволяют пользоваться обученной нейронной сетью даже при перезагрузках и сбоях

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

системы). Для этого было разработано 3 таблицы: Pretender, Standart, Weight, которые изображены на рисунках 6.16-6.18:

Pretender			
	<code>id</code>	<code>integer</code>	
	<code>first_name</code>	<code>varchar(100)</code>	
	<code>last_name</code>	<code>varchar(100)</code>	
	<code>experience</code>	<code>float(100)</code>	
	<code>salary</code>	<code>decimal(100)</code>	
	<code>skills</code>	<code>text</code>	
	<code>href</code>	<code>text</code>	
	Add field		

Рис. 6.16 – Таблица Pretender.

Столбец `id`, являющийся первичным ключом таблицы Pretender, автоинкрементный. Также в таблицы хранятся символьные колонки имя, фамилия, численный тип с плавающей запятой отвечает за опыт работы, заработка плата имеет тип десятичной дроби. Навыки претендента, как и ссылка на его резюме, хранятся в колонках с текстовым типом.

Standart			
	id	integer	
	salary	decimal	
	skills	text	
	expirience	float	
	education	float	
	Add field		

Рис. 6.17 – Таблица Standart.

weight			
	id	integer	
	condition	text	
	Add field		

Рис. 6.18 – Таблица Weight.

6.4 Реализация клиентской части.

Клиентская часть передает серверу данные о запрашиваемой вакансии, который необходимо анализировать, а также визуализирует итоговые результаты. Блок-схема основного алгоритма данной части системы представлена на рис. 6.17.

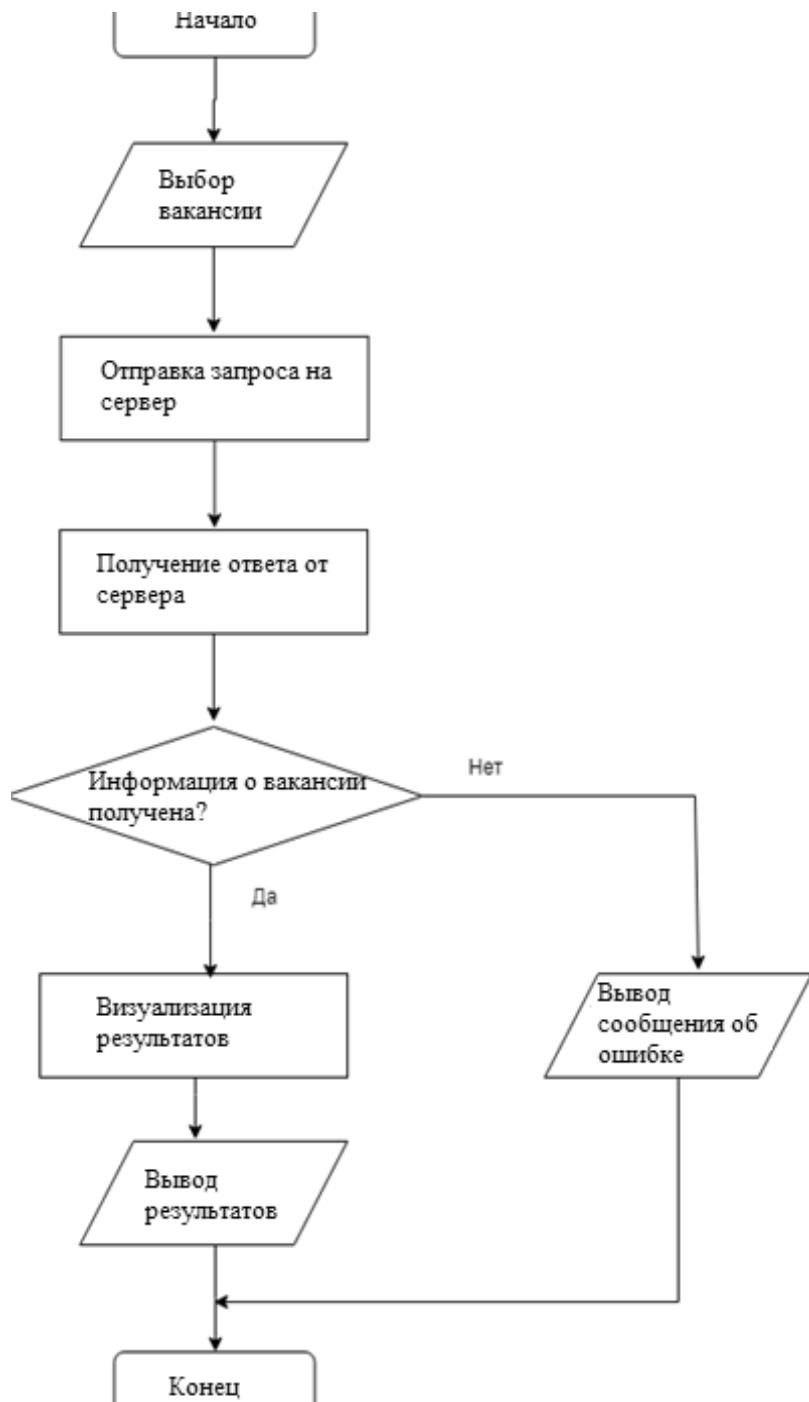


Рис. 6.19 – Блок-схема основного алгоритма клиентской части.

Основные классы клиентской части:

1. GetAnalysis

Задачей этого класса является выполнение запросов к серверу. Перечень методов:

1.1 load

Данный метод проверяет на валидность введенную пользователем информацию и, если она прошла проверку, отправляет асинхронный запрос на сервер. Если же пользователь вводит некорректный адрес, появляется сообщение об ошибке;

1.2 onSuccess

Обработчик события, который вызывается, если сервер не вернул ошибок. Данная функция декодирует результаты анализа, переданные сервером, и запускает алгоритм, визуализирующий результаты в удобный для пользователя вид;

1.3 onError

Обработчик событий, который срабатывает в случае, если сервер вернул ошибку. Задачей данного метода является вывод текста ошибки на экран.

Для реакции на действия посетителя и внутреннего взаимодействия скриптов существуют события.

Событие – это сигнал от браузера о том, что что-то произошло. Существует много видов событий. Список самых часто используемых:

События мыши:

1. click – происходит, когда кликнули на элемент левой кнопкой мыши
2. contextmenu – происходит, когда кликнули на элемент правой кнопкой мыши
3. mouseover – возникает, когда на элемент наводится мышь

Изм.	Лист	№ докум.	Подпись	Дата

4. mousedown и mouseup – когда кнопку мыши нажали или отжали
5. mousemove – при движении мыши

События на элементах управления:

1. submit – посетитель отправил форму <form>
2. focus – посетитель фокусируется на элементе, например, нажимает на input

Клавиатурные события:

1. keydown – когда посетитель нажимает клавишу
2. keyup – когда посетитель отпускает клавишу

На рис. 6.20 и 6.21 представлен исходный код класса GetAnalysis

```

1 import { Component, OnInit, Injectable } from '@angular/core';
2 import { ApiService } from '../../../../../app.api-service';
3 import { MatDialog, MatDialogRef, MatDialogConfig } from '@angular/material';
4 import { EditEducationModal } from '../../../../../modals/editEducation.component';
5 import { EditCarModal } from '../../../../../modals/editCar.component';
6 import { EditAddressModal } from '../../../../../modals/editAddress.component';
7 import { AddAddressModal } from '../../../../../modals/addAddress.component';
8 import { NewAvatarLoaderModal } from '../../../../../modals/avatar.component';
9 import { AppState } from '../../../../../app.service';
10 import { AdressModal } from '../../../../../modals/address/';
11
12 import * as _ from "lodash";
13
14 import { Observable } from 'rxjs/Rx';
15
16 @Component({
17   selector: 'profile-edit',
18   templateUrl: 'edit.html',
19   styleUrls: ['style.css']
20 })
21 @Injectable()
22 export class GetAnalysis implements OnInit{
23   public busy: boolean;
24   public success: string = '';
25   public timer;
26   public date
27

```

Рис. 6.20 – Листинг класса GetAnalysis (часть 1)

Изм.	Лист	№ докум.	Подпись	Дата

```

public validation() {
    if (
        this.obj.type === ''
        || this.obj.year === ''
        || this.obj.descr === ''
        || this.obj.expirience === ''
    ){
        this.obj.error = 'Заполните все поля';
    }
    else if(parseInt(this.obj.expirience) === this.mask || parseInt(this.obj.expirience) > this.mask){
        this.obj.error = 'Опыт работы должен быть введен в формате маски';
    }
    else{
        this.obj.error = '';
        this.sendRequest('');
    }
}

```

Рис. 6.20 – Листинг класса GetAnalysis (часть 2)

6. AppState

Класс-сервис, отвечающий за разделение логики клиентской части.

Основные методы:

6.1 get

Реализует получение метрик. Для удобства реализация вынесена в сервис;

6.2 getHtml

Генерирует на основе переданного массива данных и шаблона html-код.

Исходный код класса представлен на рисунках 6.21 и 6.22:

Изм.	Лист	№ докум.	Подпись	Дата

```

2 import { Injectable } from '@angular/core';
3 import { Subject } from 'rxjs';
4
5 export type InternalStateType = {
6   [key: string]: any
7 };
8
9 @Injectable()
0 export class AppState {
1   public _state: InternalStateType = {};
2   private vacSubrscribeManager;
3
4   constructor() {
5     this.vacSubrscribeManager = new Subject();
6   }
7
8   public vac() {
9     return this.vacSubrscribeManager;
0   }
1   public vacUpdated() {
2     this.vacSubrscribeManager.next(Date.now());
3   }
4   public set state(value) {
5     throw new Error('do not mutate the `state` directly');
6   }
7   public get state() {
8     return this._state = this._clone(this._state);
9   }

```

Рис. 6.21 – Листинг класса AppState (часть 1).

```

30   public get(prop?: any) {
31     const state = this.state;
32     return state.hasOwnProperty(prop) ? state[prop] : state;
33   }
34
35   public set(prop: string, value: any) {
36     // internally mutate our state
37     return this._state[prop] = value;
38   }
39
40   private _clone(object: InternalStateType) {
41     // simple object clone
42     return JSON.parse(JSON.stringify(object));
43   }
44 }
45

```

Рис. 6.22 – Листинг класса AppState (часть 2).

Изм.	Лист	№ докум.	Подпись	Дата

Также в клиентскую часть входят представления в контексте MVC (view).

Ниже приведена реализация html шаблонов некоторых из них:

Главная страница клиентского приложения представлена на рисунке 6.23:

```
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title></title>
    <link rel="stylesheet" href="css/bootstrap.css">
    <link rel="stylesheet" href="main.css">
    <link href="assets/css/bootstrap-responsive.css" rel="stylesheet">
</head>
<body>
    <div class="container-fluid">

        <form method="post" action="index.php?r=parse/parse">
            <div class="form-group search">
                <label for="first">Выберите вакансию из списка</label>
                <select class="form-control" name="job" id = "first">
                    <option value="Программист PHP">Программист PHP</option>
                    <option value="Программист Java">Программист Java</option>
                    <option value="Программист C#">Программист C#</option>
                </select>

                <button type="submit" class="btn btn-primary">Найти</button>
            </div>
        </form>
    </div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.js">
```

Рис. 6.23 – Листинг html кода главной страницы приложения.

Для придания информации читабельного и воспринимаемого вида использовались css-таблицы. Пример представлен на рисунке 6.24.

Асинхронное отображение информации реализовано с помощью технологии AJAX. Ajax — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

```

        background-image:url(img/neiro.jpg) ;
    }
    .nav li{
        display: inline-block;
        padding-left: 20px;
    }
    .search{
        margin-top: 200px;
        margin-left: 300px;
        margin-right: 300px;
        text-align: center;
    }

#first{
    width: 400px;
    margin-right: auto;
    margin-left: auto;
    margin-bottom: 30px;
}

.search button{
    margin-top: 5px;
}
.search label{
    color: white;
    font-size: 25px;
    margin-bottom: 10px;
}

```

Рис. 6.24 – Css стили для главной страницы.

HTML-шаблон по отображению результатов запроса представлен на рисунке 6.25. Для визуализации процентного соответствия кандидата заданной вакансии используется технология CANVAS. CANVAS, это элемент HTML 5, который предназначен для создания растрового изображения при помощи JavaScript.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

```

ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);
?>
<div class="vacs-container">

<?php

foreach ($dataset as $d) {

    echo "<div class='vac-container'>
        <b>Ссылка на соискателя:</b><a target='_blank'
        href='".$d['v']."'>$d[h]</a>
        <br>
        <b class='skillsId'>Ключевые навыки:</b>
        <div class='skillsResult'>";
        foreach ($d['skillList'] as $s) {
            echo "<span class='skill'>$s</span>";
        }
        echo "</div>";
        echo "<b>Ожидаемый уровень зарплаты:</b>$d[zarplata]<br>";
        echo "<b>Опыт работы:</b>$d[opit]<br>";
        echo "<b>Процент совпадения:</b> <br> ";
        echo "<div class='columns'>
            <div class='dial blue' data-width='180' data-lineWidth='41'>".$d['neiroPercent']."
        </div>
        </div>";
        echo "</div>";

}

```

Рис. 6.25 – Листинг html кода отображения результатов запроса.

Css-код представлен на рисунках 6.26 и 6.27. Цвета диаграмм, созданных с помощью CANVAS также вынесены в css-стили. Подобная визуализация процентных данных, улучшает восприимчивость, делает лаконичной поданную информацию. Диаграммы — это хороший способ документирования. Диаграммы позволяют быстро разобраться в сложных системах, а также служить справочными картами, позволяющими в них ориентироваться. Такие карты часто генерируются непосредственно из существующих систем, решая вечную проблему актуальности документации: диаграмма базы данных генерируется из самой БД, диаграмма классов — из исходного текста и т. п.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

```

2     background-image: url(img/neiro.jpg);
3     text-align:center;
4
5   }
6   .columns {
7     display: flex;
8     align-items: center;
9     > div {
10       flex: 1;
11       text-align: center;
12       padding: 20px;
13     }
14   }
15
16 .dial {
17   border-color: #22262f;
18   color: #689F38;
19   display: inline-block;
20   text-align: left;}
21 p {
22   text-align:center;
23   font-weight: bold;
24   color: #000;|
25   white-space: nowrap;
26   position: relative;
27   overflow: hidden;
28   z-index: 1;
29   margin: 0;
30 }

```

Рис. 6.26 – Css стили для страницы с отображением результата анализа (часть 1).

CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

```
... .vac-container {  
56     width: 450px;  
57     margin-left: auto;  
58     margin-right: auto;  
59     margin-top: 15px;  
60     margin-bottom: 20px;  
61     background: #eee;  
62     padding: 11px;  
63     padding-left: 15px;  
64     padding-right: 15px;  
65     border: 1px solid #E0E0E0;  
66     border-radius: 10px;  
67     display: inline-block;  
68     margin-left: 15px;  
69     margin-right: 15px;  
70     vertical-align: top;  
71     min-height: 587px;  
72     background: #fff;  
73     box-shadow: 0 0 10px #000;  
74     text-align:left;  
75     opacity: .9;  
76     position: relative;  
77 }  
78  
79 .vac-container .columns {  
80     margin-top: 20px;  
81     margin-left: auto;  
82     margin-right: auto;  
83     max-width: 222px;  
84 }  
85 }
```

Рис. 6.26 – Css стили для страницы с отображением результата анализа (часть 2).

Выводы по разделу шесть:

В данном разделе были отображены все функции, разработанные в проекте, была предоставлена реализация основных компонентов системы. Рассмотрены разработанные алгоритмы, отвечающие за серверную часть и

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

работы искусственной нейронной сети. Представлена реализация серверной и клиентской части, искусственной нейронной сети. На основании данной главы можно сделать выводы, что разработанные компоненты функционируют в соответствии с требованиями, представленными в главе 2.

7 ПРИМЕРЫ РАБОТЫ

В данной главе будут визуализированы результаты работы сервиса, также будут проверена работоспособность.

На рисунке 7.1 располагается страница работодателя, выбирающего конкретную вакансию, по которой он хочет получить претендентов:

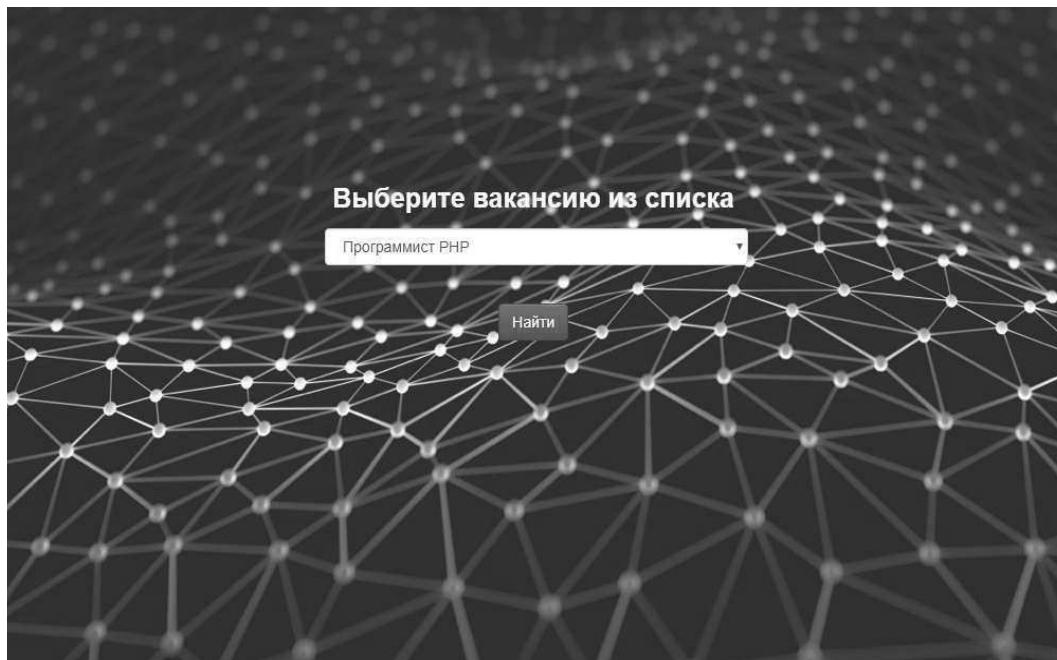


Рис. 7.1 – Главная страница пользователя.

На рисунках 7.2 и 7.3 отображены результаты запроса. Отображены ключевые навыки, опыт работы, желаемая заработка плата. Присутствует ссылка на соискателя. Также визуализирована процентная оценка:

Изм.	Лист	№ докум.	Подпись	Дата



Рис. 7.2 – Результаты поиска.

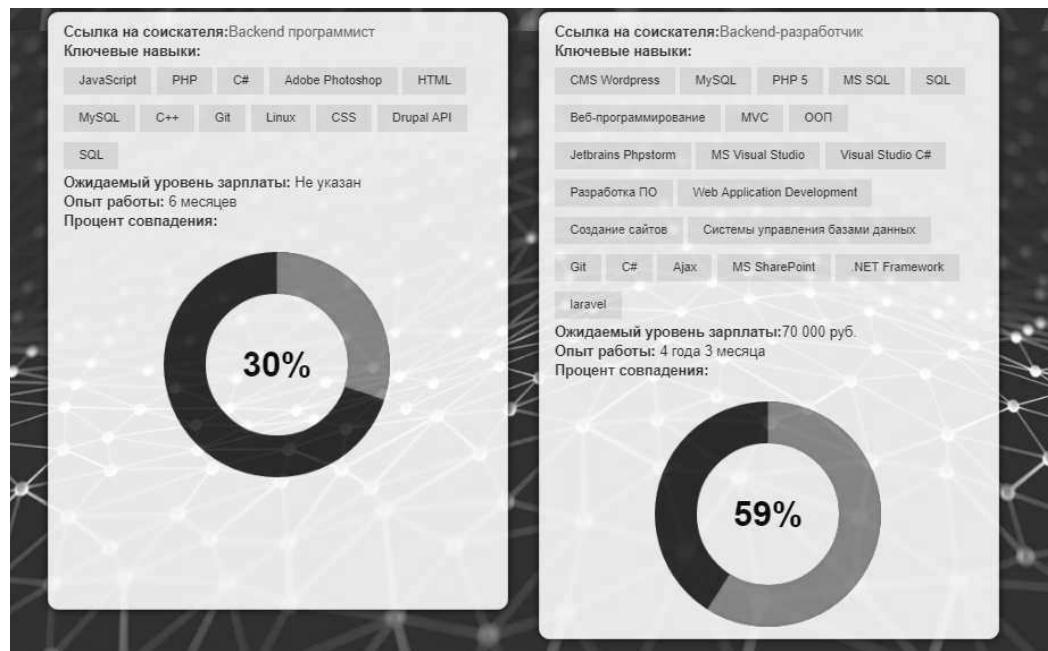


Рис. 7.3 – Результаты поиска.

Выводы по разделу семь:

В данном разделе были визуализированы результаты работы веб-сервиса.

Изм.	Лист	№ докум.	Подпись	Дата

ЗАКЛЮЧЕНИЕ

В результате работы были разработаны основные алгоритмы поиска и анализа резюме, разработан веб-сервис.

Для достижения цели были поставлены и решены следующие задачи:

- Рассмотрены различные варианты резюме и выявлены метрики, максимально характеризующие кандидата;
- Проанализированы существующие аналоги;
- Спроектирована схема базы данных;
- Разработан алгоритм работы серверной части;
- Разработан алгоритм работы искусственной нейронной сети;
- Реализован веб-сервис.

Пояснительная записка отображает описание всех процессов разработки веб-сервиса по поиску и анализу резюме, начиная с рассмотрения аналогов и выбора платформы для разработки, до реализации системы.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ЮУрГУ-090301.2018.135

Лист
65

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Neural Network Design. [Электронный ресурс]. – Режим доступа: <https://playground.tensorflow.org/>. – (дата обращения 12.04.2018).
2. Введение в архитектуры нейронных сетей [Электронный ресурс]. – Режим доступа: <https://habr.com/company/oleg-bunin/blog/340184/>. – (дата обращения 20.03.2018).
3. Алистер, К. О. Современные методы описания функциональных требований к системе. – Ч. 1. Линейные системы. Учебное пособие / К. О. Алистер - Москва: Изд. Лори, 2002. – 76 с.
4. Карл, И. Вигерс. Разработка требований к программному обеспечению / И. Карл, И. Вигерс. – 2-е изд., перераб. и доп. – М.: Microsoft, 204. – 504 с., ил.
5. Гарольд Лоусон. Путешествие по системному ландшафту / Гарольд Лоусон. – Москва: Издательский центр ДМК, 2013. – 93 с
6. PHP [Электронный ресурс]. – Режим доступа: <http://php.net/manual/ru/langref>. – (дата обращение 15.03.2018).
7. Yii PHP Framework [Электронный ресурс]. – Режим доступа: <https://www.yiiframework.com/>. – (дата обращения 12.04.2017).
8. Composer — менеджер зависимостей для PHP [Электронный ресурс]. – Режим доступа: <https://habr.com/post/145946/>. – (дата обращение 20.03.2018).
9. What is Unreal Engine 4 ? [Электронный ресурс]. – Режим доступа: <https://www.unrealengine.com/>. – (дата обращение 18.03.2018).
10. Angular [Электронный ресурс]. – Режим доступа <https://angular.io/guide/quickstart>. – (дата обращение 25.03.2018).
11. Qt Graphics Framework [Электронный ресурс]. – Режим доступа: [http://doc.qt.io/qt-5/graphicsview.html/](http://doc.qt.io/qt-5/graphicsview.html). – (дата обращение 25.03.2018).
12. What is SFML ? [Электронный ресурс]. – Режим доступа: <https://www.sfml-dev.org/grl-whatis/> – (дата обращение 25.03.2018).

Изм.	Лист	№ докум.	Подпись	Дата

ЮУрГУ-090301.2018.135

Лист
66

13. Space Invaders. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/SpaceInvaders/> – (дата обращения 13.04.2018).
14. Radiant. [Электронный ресурс]. – Режим доступа: <http://www.hexag.net/radiant/>. – (дата обращения 13.04.2018).
15. Архитектура REST. [Электронный ресурс]. – Режим доступа: <https://habr.com/post/38730/>. – (дата обращения 13.04.2018).
16. Beyond Space. [Электронный ресурс]. – Режим доступа: <http://store.steampowered.com/app/297111/>. – (дата обращения 13.04.2017).
17. Банда четырех: паттерны проектирования. [Электронный ресурс]. – Режим доступа: <https://www.ozon.ru/context/detail/id/2457392/>. – (дата обращения 13.04.2018).
18. Диаграмма вариантов использования как концептуальное представление бизнес-системы в процессе ее разработки. [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/32/32/lecture/1004/>. – (дата обращения 10.02.2018).

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------