



**AppLine**  
Innovation

ООО «ИБС АППЛАЙН ИННОВАЦИИ»  
ОГРН 5177746371237, ИНН/ КПП 7731394794/ 773101001

Россия, 121205, г.Москва, Территория Сколково Инновационного  
Центра, Большой б-р, дом 42, строение 1, эт. 1 пом. 335 Р.М.08  
телефон/факс: +7 (495) 967 80 80  
[info@apline.ru](mailto:info@apline.ru), [www.apline-innovation.ru](http://www.apline-innovation.ru)

## **Описание системы сбора и ранжирования резюме «Скаут»**



## Оглавление

|      |   |   |
|------|---|---|
| 1.   | Описание продукта.....                                | 3 |
| 1.1  | Цель продукта.....                                    | 3 |
| 1.2  | Основные пользователи системы:.....                   | 3 |
| 1.3  | Основные сущности системы:.....                       | 3 |
| 1.4  | Задачи, выполняемые Системой.....                     | 3 |
| 2.   | Стек технологий.....                                  | 4 |
| 4.   | Функции системы.....                                  | 4 |
| 4.1. | Сбор резюме из различных источников.....              | 4 |
| 4.2. | Обработка резюме по различным профилям/вакансиям..... | 4 |
| 4.3. | Фильтрация и ранжирование резюме.....                 | 4 |
| 5.   | Архитектурные особенности системы.....                | 6 |
| 5.1  | Архитектурная диаграмма.....                          | 6 |
| 6.1  | Страница поиска.....                                  | 6 |
| 6.2  | Карточка кандидата.....                               | 8 |
| 7.   | Нефункциональные требования.....                      | 8 |



## 1. Описание продукта

**ИС Скаут** – инновационная скоринговая система на базе модуля машинного обучения, осуществляющая ранжирование и маркировку резюме кандидатов по степени их соответствия профилю поиска.

Система будет интегрирована как с внешними системами (источники резюме), так и с внутренними (заявки на поиск, описание профилей и т.д.).

Рекрутеры будут иметь возможность оставлять обратную связь по качеству маркировки резюме в режиме реального времени, что позволит реализовать процесс непрерывного самообучения ML-модели.

### 1.1 Цель продукта

**Цель системы** – частично автоматизировать работу рекрутера, существенно снизив количество рутины, связанной с первичной фильтрацией потока кандидатов. Как результат, повышение скорости подбора, а также рост качества и релевантности кандидатов.

В долгосрочной перспективе мы ожидаем, что система сократит среднее количество трудозатрат (времени и денег) на комплектацию открытых позиций в производственных командах, что позволит компании ИБС стать еще более технологичной и привлекательной для наших существующих и будущих клиентов.

### 1.2 Основные пользователи системы:

Рекрутеры – запускают процесс выгрузки из источников и скоринга резюме по ключевым словам, указанным в описании вакансии, берут в работу кандидатов, начиная с тех, у кого самый высокий рейтинг. Рекрутеры имеют возможность оставлять обратную связь для улучшения обучения ML модели.

### 1.3 Основные сущности системы:

1. ML модель
2. Модуль загрузки резюме из внешних источников через API
3. Модуль запуска ML модели и сохранения результатов
4. Хранилище резюме и результатов ранжирования
5. Серверные модули
6. Web UI приложение

### 1.4 Задачи, выполняемые Системой

Система должна выполнять следующие задачи:

1. Сбор резюме с рабочего ресурса hh.ru с учетом заданных критериев поиска и описания вакансии.
2. Фильтрация нерелевантных резюме.



3. Ранжирование релевантных резюме с помощью алгоритмов машинного обучения.
4. Возможность просмотреть карточку кандидата
5. Возможность сохранить резюме кандидата в формате rtf или pdf.
6. Возможность запросить контакты кандидата, если они скрыты.

## 2. Стек технологий

При реализации решения должен использоваться следующий технологический стек:

- Java, Spring
- База данных: PostgreSQL;
- React, Java Script;
- Python;
- Основные python-библиотеки:
  - Pymystem3;
  - Catboost;
  - Sklearn;
  - и другие.

## 4. Функции системы

### 4.1. Сбор резюме из различных источников

Функция обеспечивает возможность сбора резюме с рабочего сайта hh.ru. Взаимодействие с источником резюме осуществляется через API с учетом всех правил, требований и ограничений. Для снижения нагрузки на канал, система кеширует скаченные резюме во избежание повторного скачивания без необходимости.

### 4.2. Обработка резюме по различным профилям/вакансиям

Функционал сбора и ранжирования резюме работает одинаково эффективно вне зависимости от специализации кандидатов (Java, QA, Аналитик и т.д.) и учитывает индивидуальные требования, описанные в профиле поиска (описании вакансии).

Сбор резюме по различным профилям запускается в режиме реального времени с индикацией статуса выполнения на UI. Поиски, запущенные одновременно разными рекрутерами, умеют работать параллельно и не блокируют друг друга.

### 4.3. Фильтрация и ранжирование резюме

В рамках каждого поиска система выгружает из источника набор резюме, согласно заданным критериям фильтрации. Максимальное количество резюме в одном наборе не должно превышать определённый лимит, редактируемый в наборе свойств системы (например, не более 100 резюме за один поиск).



Затем список резюме ранжируется по степени соответствия описанию вакансии с помощью ML модели. На первое место в списке попадают кандидаты с наилучшим соответствием. ML модель должна эффективно работать для всего спектра специализаций и требований к кандидатам.

В основе ранжирующего модуля лежит анализ естественного текста. Для анализа естественного текста используются BERT-подобные языковые модели. Особенность такого подхода является отсутствие ограниченного набора характерных слов и возможность понимать любое слово в тексте вакансии или резюме. Модели преобразовывают естественный язык текста вакансии / резюме в векторное представление. Это векторное представление используется для ранжирования резюме кандидатов относительно текста вакансии. Таким образом одно и то же резюме может иметь свой ранг в зависимости от рассматриваемой вакансии. Модели обучались на нашем размеченном датасете вакансий-резюме, используя видеокарту. Работает ранжирующая модель на видеокарте, что увеличивает скорость работы. Так же применен механизм кеширования результатов, что исключает повторные вычисления и увеличивает скорость работы модуля.

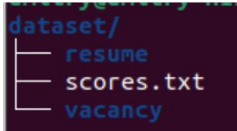
Сводная таблица особенностей ML модели:

|                                 | ИС Скаут   |
|---------------------------------|--|
| Модель                          | <ol style="list-style-type: none"><li>1. Векторизация резюме (Embedder)</li><li>2. Векторизация вакансии (Embedder)</li><li>3. Кодирование резюме / вакансии (языковая модель)</li><li>4. Ранжирование</li></ol>   |
| Словарь модели                  | Не фиксирован, слова не ограничены   |
| Разнообразие вакансий           | Любые вакансии   |
| Учет описания вакансии в модели | Учитывается  |
| Результат                       | <p>Список ранжированных резюме для данной вакансии без разбивки на категории. Сверху к низу идут от самых подходящих резюме к менее подходящим.</p> <p><i>В данном подходе словарь не фиксирован. Это значит, что наша модель сможет распознать любые слова (даже опечатки в словах). Соответственно, модель поймет новые слова (Capital markets, kotlin etc) и сможет их учитывать при подборе вакансий, но качество подбора возможно будет хуже.</i></p> |



Подготовка данных для обучения (разметка) и само обучение осуществляется экспертами в ручном режиме. Обучение ML модели осуществляется с определенной периодичностью.

Обучающие данные будем хранить в папке dataset, дерево директории показано ниже:



В папке resume находятся резюме, названия которых уникальны на латинском языке (без кириллицы). В папке vacancy лежат вакансии с уникальными названиями на латинице. В файле scores.txt лежат размеченные данные (пример ниже):

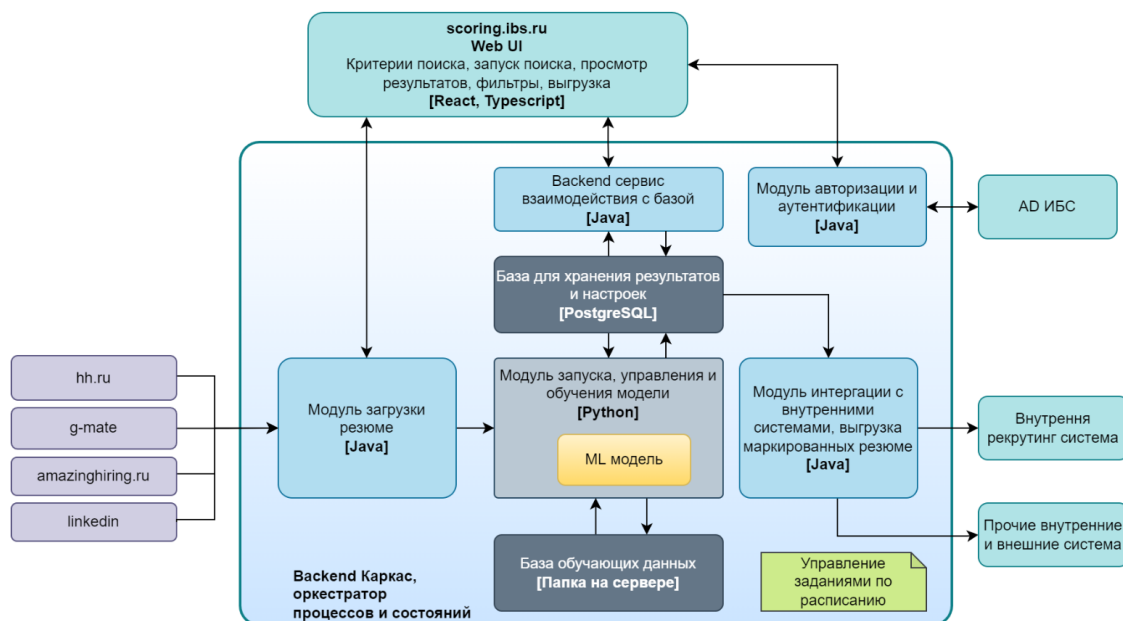
```
1 unique_name_resume_1, unique_name_vacancy_1, score
```

В каждой строчке файла находится имя\_файла\_резюме, имя\_файла\_вакансия и балл соответствия от 1 до 4. Данные записываем через запятую.

## 5. Архитектурные особенности системы

Архитектура системы реализует микросервисный подход. Интеграция различных компонент системы (серверная часть, ML, UI) осуществляется через API.

### 5.1 Архитектурная диаграмма



### 6.1 Страница поиска



Описание вакансии

Обязательные:

- Уверенное знание React 17 (Hooks/Redux/Thunk) и TypeScript, с практическим опытом проектирования и разработки.
- Хорошее знание JavaScript (ES5+) и особенностей языка.
- Понимание алгоритмов и структур данных.
- Навыки верстки, понимание CSS/SCSS.
- Опыт работы от 4 лет

Желательные:

- Опыт профилирования, оптимизации и рефакторинга JS (TS) кода.
- Опыт работы с webpack, web sockets, service workers.
- Уверенное владение git
- Понимание принципов проектирования и архитектуры программного обеспечения.

Ключевые теги: #ReactJS #TypeScript

Критерии поиска

java spring

ПОИСК

Искать по пунктам резюме

в опыте работы

Выберите регионы

Новгородская область

Найдено и обработано 9 резюме

| Имя кандидата                               | Рейтинг           | Регион           | Должность             | Обновлено  |
|---|-------------------|------------------|-----------------------|------------|
| <a href="#">Контакты не открыты</a>         | 7.950734615325928 | Великий Новгород | Java-разработчик      | 4/24/2023  |
| <a href="#">Контакты не открыты</a>         | 7.950227737426758 | Великий Новгород | Java-разработчик      | 10/20/2023 |
| <a href="#">Контакты не открыты</a>         | 7.947101593017578 | Великий Новгород | Java-разработчик      | 10/2/2023  |
| <a href="#">Абраменков Евгений</a>          | 7.942526340484619 | Великий Новгород | Инженер-программист   | 11/25/2023 |
| <a href="#">Контакты не открыты</a>         | 7.940363407135010 | США              | Middle Java Developer | 7/27/2021  |
| <a href="#">Zhdanov Alexander Alexander</a> | 7.939565658569336 | Valday           | Python programmer     | 2/17/2023  |



## 6.2 Карточка кандидата

### Контакты не открыты

Java-разработчик

29 декабря 1986 г. (36 лет)

ОТКРЫТЬ КОНТАКТЫ

#### Опыт работы

Сбербанк - Технологии

октябрь 2021 - январь 1970 (51 год 9 месяцев)

Middle Java-разработчик (Ведущий ИТ-инженер)

Системная интеграция, автоматизации технологических и бизнес-процессов предприятия, ИТ-консалтинг  
Разработка, рефакторинг и интегрирование в многофункциональную платформу rest микросервиса с несколькими версиями API: - использование AOP и других возможностей инфраструктуры Spring - валидация с помощью Javax аннотаций - реализация мониторинга по стандарту Prometheus - визуализация данных через Grafana - журналирование с помощью библиотеки логирования logback, реализующая SLF4J - тестирование с помощью Mockito, MockWebServer - применение Lombok - использование в производственном процессе Nexus CI/CD, Bitbucket, Docker, Jenkins, Jira

JavaOnlineProjects

январь 2021 - сентябрь 2021 (8 месяцев)

Java-разработчик

Участие в разработке Spring 5/JPA enterprise-приложения с авторизацией и правами доступа на основе ролей на стеке: Maven, Spring MVC, Spring Security, REST (Jackson), Java 8 Stream API: - реализация сохранения в базы Postgres и HSQLDB на основе Spring JDBC, JPA(Hibernate) и Spring-Data-JPA - реализация и тестирование REST и AJAX контроллеров (более 130 JUnit тестов) - реализация клиента на Bootstrap (css/js), datatables, jQuery + plugins.

BENTELER

сентябрь 2010 - сентябрь 2020 (10 лет)

CAE Engineer

Автокомпоненты, запчасти (производство)  
Расчёт, проектирование и разработка компонентов автомобильных подвесок

#### Ключевые навыки

## 7. Нефункциональные требования

- Система должна быть совместима с российскими ОС (Альт, Астра Линукс и т.д.).
- Стабильная работа до 20 потоков загрузки одновременно
- Стабильная работа до 20 потоков ранжирования одновременно
- Время ранжирования одного резюме не должно превышать 2-х секунд (при расчетной нагрузке сервиса)