**Спецификация требований к программному обеспечению**

Информация о документе:  
  
Идентификатор документа: SRS-2023-HH  
Редакция: 1.0  
Дата: 2023-02-06  
Автор: Алексей Боровой

[**1. Введение**](#_gjdgxs) **3**

[Цели проекта](#_30j0zll) 3

[Предполагаемая аудитория и последовательность восприятия](#_1fob9te) 3

[Масштаб проекта](#_3znysh7) 3

[Ссылки на источники](#_2et92p0) 4

[**2. Общее описание**](#_tyjcwt) **4**

[Видение продукта](#_3dy6vkm) 4

[Функциональные требования:](#_wkhkbv95vao1) 5

[Классы и характеристики пользователей](#_2s8eyo1) 5

[Среда функционирования продукта (операционная среда)](#_3rdcrjn) 5

[Рамки, ограничения, правила и стандарты](#_26in1rg) 5

[**3. Функциональность системы**](#_1ksv4uv) **5**

[Функциональный блок 1. Сбор данных](#_44sinio) 5

[Функциональный блок 2. Очистка и предобработка информации для помещения в базу данных](#_z337ya) 7

[Функциональный блок 3. Создание базы данных](#_3j2qqm3) 7

[Функциональный блок 4. Помещение обработанной информации в базу данных](#_1y810tw) 7

[Функциональный блок 5. Создание интерактивного дашборда.](#_4i7ojhp) 7

[Функциональный блок 6. Отладка. Поиск и устранение ошибок и неисправностей в работе приложения.](#_1ci93xb) 7

[**4. Требования к внешним интерфейсам**](#_3whwml4) **7**

[**5. Нефункциональные требования**](#_2bn6wsx) **7**

[**6. Прочее**](#_qsh70q) **8**

# Введение

В этом документе представлена полная спецификация требований к программному обеспечению для проекта, направленного на разработку приложения для сбора, обработки, хранения и визуализации информации о вакансиях по специальностям “Аналитик” (Data Analyst) (все направления), “Инженер данных” (Data engineer), “Дата саентист” (Data Scientist).

## Цели проекта

* Получение исчерпывающей и наглядной информации о распределении вакансий по направлениям аналитики данных (в количественном и процентном выражении).
* Определение для каждого направления аналитика данных ключевых навыков, подсчет количественного выражения упоминаний тех или иных навыков в вакансиях, составление рейтинга таких навыков по частоте упоминания.
* Определение распределения вакансий по: географии, отраслям бизнеса, уровню подготовки специалистов (грейд).

## Предполагаемая аудитория и последовательность восприятия

Предполагаемой аудиторией (конечными пользователями) приложения являются соискатели на вакантные должности, такие как:

* Аналитик (“Data Analyst”) (с учетом всех направлений);
* Инженер данных (“Data engineer”);
* Дата саентист (“Data Scientist”) (в т.ч. ML-инженер, ML-researcher и т.д.).

## Масштаб проекта

Данный проект позволит конечному пользователю получить актуальную информацию о вакансиях в области аналитики, размещенных на сайте hh.ru.

В данном проекте задействована рабочая группа, состоящая из 5 (пяти) сотрудников: лидера команды (Team Lead), системного аналитика, двух дата инженеров и аналитика данных. Бюджет проекта - 0 руб. Сроки проекта - 3 месяца.

Границы проекта: от сайта-агрегатора вакансий ([HH.ru](http://hh.ru)) до конечного потребителя визуальной информации на интерактивном дашборде.

## Ссылки на источники

Вся информация, а также рабочие файлы размещаются в Git-репозитории: <https://github.com/mgarkunov/hh_ru_analytics>

# Общее описание

## Видение продукта

В настоящее время на сайтах-агрегаторах каждый день размещается множество вакансий по различным направлениям бизнеса, промышленности, торговли, услуг и проч. В виду того, что инструментов, которыми должен обладать соискатель на желаемую должность весьма большое количество, такой соискатель может испытывать трудности с определением перечня необходимых к изучению технологий. Данный проект позволит получить распределение по необходимым инструментам и технологиям в зависимости от искомой должности и выделить наиболее популярные у работодателей. Это позволит соискателю чётко определить для себя приоритет в изучении технологий.

Кроме того, в данном проекте проводится аналитика распределения технологий в зависимости от географического расположения работодателя, что позволит сегментировать технологии по населенным пунктам.

В проекте предусматривается систематическая выгрузка данных с сайта-агрегатора, очистка и аккумулирование информации в базе данных, после чего данные размещаются в интерактивном обновляемом дашборде.



## Функциональные требования:

## Классы и характеристики пользователей

Конечный потребитель продукта (пользователь) - соискатель, заинтересованный в получении вакантной должности, либо человек, занимающийся повышением своих навыков, стремящийся получить исчерпывающую информацию по наиболее популярным навыкам, отображаемым в описании вакансии на сайте [HH.ru](http://hh.ru) .

Всего потенциальных пользователей - неограниченное количество.

Предполагается, что основным средством работы с интерактивным дашбордом у пользователей будет являться персональный компьютер.

## Среда функционирования продукта (операционная среда)

Доступ конечного потребителя к интерактивному дашборду должен быть обеспечен из любого веб-браузера и с любого устройства (персональный компьютер, планшет, смартфон), имеющего доступ к сети интернет.

## Рамки, ограничения, правила и стандарты

1. Код, разрабатываемый на языке  Python, должен соответствовать требованиям PEP-8 ([PEP 8 – Style Guide for Python Code](https://peps.python.org/pep-0008/));
2. Запросы к сайту hh.ru при помощи официального API должны выполняться согласно требованиям документации к данному API ([GitHub - hhru/api: HeadHunter API: документация и библиотеки](https://github.com/hhru/api));
3. Проектируемая база данных должна соответствовать требованиям официальной документации разработчика баз данных  PostgreSQL ([15: PostgreSQL 15.1 Documentation](https://www.postgresql.org/docs/current/index.html));
4. Визуализация должна быть выполнена в соответствии с требованиями документации Data Lens ([Yandex DataLens | Yandex Cloud - Документация](https://cloud.yandex.ru/docs/datalens/));

# Функциональность системы

## Функциональный блок 1. Сбор данных

* Описание и приоритет.

Данный блок представляет собой сбор данных с сайта [HH.ru](https://hh.ru/) через предлагаемый сайтом API с помощью запросов к сайту в среде Python 3 и обработку полученных данных.

Приоритет - высокий.

* Функциональные требования:

1. Среда выполнения процедуры: Python 3, version 3.8+. Используемые библиотеки: Pandas, Requests, Datetime.
2. Скрипт должен получать данные о вакансиях с сайта [HH.ru](https://hh.ru/) с помощью его API. Данные должны быть возвращены в формате, который определяется работой API (JSON).
3. Аутентификация API: Скрипт должен аутентифицироваться с API [HH.ru](https://hh.ru/), используя необходимые учетные данные (токен или API-ключи), либо производить запросы и получение данных без аутентификации (в случае, если API это позволяет). Необходимо предусмотреть оба способа.
4. Скрипт должен отправлять запрос на сайт [HH.ru](https://hh.ru/) в соответствии с требованиями документации на  API.
5. Скрипт должен обрабатывать любые ошибки, возникающие во время получения или обработки данных. Это может включать в себя повторное выполнение неудачных запросов или протоколирование сообщений об ошибках для последующего анализа.
6. Скрипт должен обрабатывать получаемые данные в формате JSON, полученные от API [HH.ru](https://hh.ru/), и преобразовывать их в формат плоской таблицы с целью последующего сохранения таких данных в базу данных. Скрипт должен уметь производить определенные операции над данными, такие как: фильтрация, агрегирование или преобразование данных перед их сохранением в базе данных.
7. Скрипт должен быть разработан с учетом производительности, чтобы обеспечить эффективное получение и обработку данных, минимизируя затрачиваемые ресурсы. Скрипт должен запускаться и выполнять все заложенные в него функции на серверах с минимально-допустимыми для запуска Python версии 3.8 и более.
8. Все шаги в скрипте должны быть задокументированы и иметь описание работы. Все функции должны иметь описание работы, параметров и проч. Это гарантирует, что другие пользователи смогут понять и эффективно использовать скрипт.
9. Запрос на сайт [HH.ru](https://hh.ru/) должен обладать следующими параметрами (ниже указаны только параметры запроса, требующие уточнения, остальные параметры остаются “по-умолчанию”):

* page - номер страницы (значения принимаются равными 1, 2, 3…..n);
* per\_page - количество элементов на страницу (значение принимается равным 100);
* text - текст фильтра (указывается наименование вакансии (только одной));
* area - регион поиска (значение принимается равным 113 (Россия));
* date\_from - дата, которая ограничивает снизу диапазон дат публикации вакансий. Значение указывается в формате ISO 8601: YYYY-MM-DDTHH;
* date\_to - дата, которая ограничивает сверху диапазон дат публикации вакансий. Значение указывается в формате ISO 8601: YYYY-MM-DDTHH;
* responses\_count\_enabled - включение счётчика откликов (параметр принимается true);

1. Блок-схема работы скрипта:

<https://lucid.app/lucidchart/e2b6384b-d3dd-4804-a22b-042175237634/edit?viewport_loc=550%2C274%2C1597%2C804%2C0_0&invitationId=inv_587c0399-d1d3-4ee5-a9f7-e42652ef5c6e>

(После завершения работы над блок-схемой будет вставлено изображение)

## Функциональный блок 2. Очистка и предобработка информации для помещения в базу данных

## Функциональный блок 3. Создание базы данных

## Функциональный блок 4. Помещение обработанной информации в базу данных

## Функциональный блок 5. Создание интерактивного дашборда.

## Функциональный блок 6. Отладка. Поиск и устранение ошибок и неисправностей в работе приложения.

# Требования к внешним интерфейсам

**Интерфейсы пользователя (UX)**

**Программные интерфейсы**

**Интерфейсы оборудования**

**Интерфейсы связи и коммуникации**

# Нефункциональные требования

**Требования к производительности**

**Требования к сохранности (данных)**

**Требования к качеству программного обеспечения**

**Требования к безопасности системы**

**Требования на интеллектуальную собственность**

# Прочее

**Приложение А: Глоссарий**

**Приложение Б: Модели процессов и предметной области и другие диаграммы**

**Приложение В: Список ключевых задач**

# TO\_DO\_LIST:

1. Вставить блок Расима со структурой БД
2. Вставить блок Расима с инкрементацией
3. Описать работу скрипта при наличии ошибки загрузки в БД
4. Обновляемый скрипт (получить дату последней загрузки данных?)
5. Инкремент с нахлестом (1 день)
6. Дедубликация?