**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**](#_heading=h.gjdgxs) **3**

[Цели проекта](#_heading=h.30j0zll) 3

[Предполагаемая аудитория и последовательность восприятия](#_heading=h.1fob9te) 3

[Масштаб проекта](#_heading=h.3znysh7) 3

[Ссылки на источники](#_heading=h.2et92p0) 4

[**2. Общее описание**](#_heading=h.tyjcwt) **4**

[Видение продукта](#_heading=h.3dy6vkm) 4

[Функциональные требования:](#_heading=h.1t3h5sf) 5

[Классы и характеристики пользователей](#_heading=h.4d34og8) 5

[Среда функционирования продукта (операционная среда)](#_heading=h.2s8eyo1) 5

[Рамки, ограничения, правила и стандарты](#_heading=h.17dp8vu) 5

[**3. Функциональность системы**](#_heading=h.3rdcrjn) **5**

[Функциональный блок 1. Сбор данных](#_heading=h.26in1rg) 5

[Функциональный блок 2. Очистка и предобработка информации для помещения в базу данных](#_heading=h.lnxbz9) 8

[Функциональный блок 3. Создание базы данных](#_heading=h.35nkun2) 8

[Функциональный блок 4. Создание интерактивного дашборда.](#_heading=h.44sinio) 13

[Функциональный блок 5. Отладка. Поиск и устранение ошибок и неисправностей в работе приложения.](#_heading=h.2jxsxqh) 13

[**4. TO\_DO\_LIST:**](#_heading=h.4i7ojhp) **13**

# 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

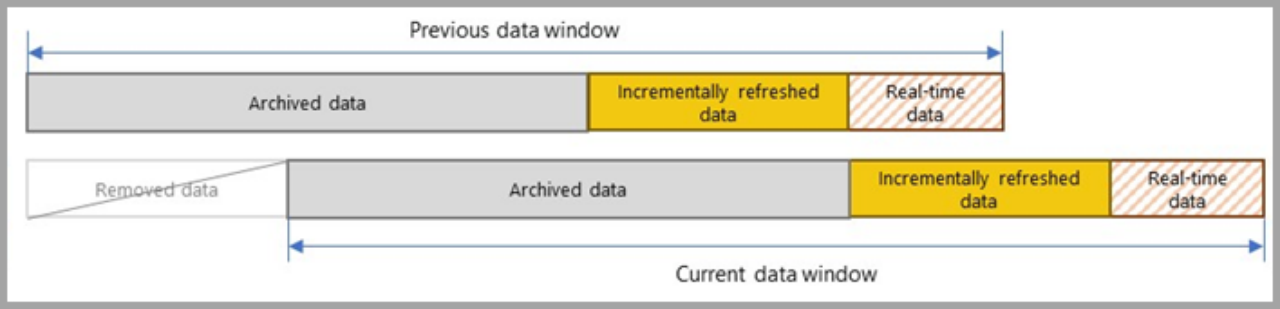
* Описание и приоритет.

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. По итогам запроса скрипт должен получить данные за последние 16 дней и в последующем загружать данные каждый день, с периодичностью 1 час. Чтобы исключить повторы, старт выгрузки начинать в 01:00:01, 02:00:01, 03:00:01 и т.д., до 00:00:01. А 00:00:00, 01:00:00, 02:00:0 включать в предыдущий диапазон. Подобный формат используется для того, чтобы уйти от ограничения в 2000 вакансий за одну загрузку.



1. Скрипт параллельно должен выполнять 2 запроса:
2. По profession role. Перечень id professional role представлен ниже:

* 157,
* 156,
* 10,
* 150,
* 165,
* 164,
* 148,
* 163,
* 134;
* 40 (это значение подразумевает название специальности “Другое”).

1. По поисковому запросу. Поисковый запрос представлен ниже:

[СГЕНЕРИРОВАННЫЙ ПОИСКОВЫЙ ЗАПРОС]

1. Скрипт должен формировать цикличные запросы (вывод по 100 строк). Отслеживание количества полученных строк с данными производится целочисленным делением количества строк на 100. Сохранение в базу данных происходит по 100 строк за 1 обращение к БД. При количестве строк более 100, скрипт должен делить данные на объем по 100 строк и производить циклические запросы к API и БД.
2. При первичном получении информации по запросу (в обоих запросах из п.9), скрипт должен выводить 5 следующих полей:

* **id,**
* **name,**
* **type,**
* **published\_at,**
* **created\_at.**

После получения базовых полей, скрипт должен преобразовывать данные из пространственной таблицы в плоскую и сохранять поля в базу данных.

1. Скрипт должен уметь подключаться к сформированной базе данных, обрабатывать возможные ошибки подключения/сохранения данных.
2. При сохранении первичных данных должна производиться проверка на дублирование данных, уже находящихся в базе данных.

Первичная проверка происходит по уникальному id вакансии. В случае отсутствия id вакансии в базе данных производится запись в БД.

В случае, если id вакансий имеется в БД, происходит вторичная проверка по дате размещения вакансии (поле **published\_at**). Если id вакансии совпадает, но дата размещения вакансии отличается, вакансия, имеющая более раннюю дату публикации (новая, полученная при работе скрипта) перезаписывается (как наиболее актуальная).

В случае, если уникальный id вакансии, дата публикации вакансии совпадают с уже имеющимися в БД, такая вакансия игнорируется и не сохраняется в БД.

1. При записи в БД скрипт должен формировать дополнительное поле **tech\_change\_date** (тип **timestamp** [ (p) ] [ without time zone ]) с записью текущей даты и времени обращения к БД.

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

Скрипт должен обращаться к базе данных с первичными (базовыми полями), получать базовые поля (см. п 11 функционального блока 1) и производить запрос расширенных данных по каждой вакансии.

После этого скрипт должен преобразовывать пространственную таблицу в плоскую и производить запись преобразованных данных в базу данных.

## Функциональный блок 3. Создание базы данных

Данные могут выгружены в 2-х вариациях: 37 столбцов и 96 столбцов. Основная разница в том, что в таблице с 37 столбцами «обрезано» выдаются столбцы **requirement** и **responsibility**. Что бы убрать момент неточности информации в данных, будем выгружать вторую таблицу

В базу данных PostgreSQL данные загружать по типам, представленным ниже в таблице:

| **Имя** | **Тип** | **Тип в БД** | **Описание** |
| --- | --- | --- | --- |
| id | string | INT PRIMARY KEY AUTO\_INCREMENT | Идентификатор вакансии |
| description | string | VARCHAR(300) | Описание вакансии, содержит html |
| branded\_description | string или null | VARCHAR(300) | Брендированное описание вакансии |
| key\_skills | array | VARCHAR(300) | Информация о ключевых навыках, заявленных в вакансии. Список может быть пустым. |
| key\_skills[].name | string | VARCHAR(300) | название ключевого навыка |
| schedule | object | VARCHAR(300) | График работы. Элемент справочника schedule |
| schedule.id | string | INT | Идентификатор графика работы |
| schedule.name | string | VARCHAR(300) | Название графика работы |
| accept\_handicapped | boolean | BOOLEAN(TRUE, FALSE, UNKNOWN) | Указание, что вакансия доступна для соискателей с инвалидностью |
| accept\_kids | boolean | BOOLEAN(TRUE, FALSE, UNKNOWN) | Указание, что вакансия доступна для соискателей от 14 лет |
| experience | object | VARCHAR(300) | Требуемый опыт работы. Элемент справочника experience |
| experience.id | string | INT | Идентификатор требуемого опыта работы |
| experience.name | string | VARCHAR(300) | Название требуемого опыта работы |
| address | object или null | VARCHAR(300) | Адрес вакансии |
| alternate\_url | string | VARCHAR(300) | Ссылка на представление вакансии на сайте |
| apply\_alternate\_url | string | VARCHAR(300) | Ссылка на отклик на вакансию на сайте |
| code | string или null | INT | Внутренний код вакансии работодателя |
| department | object или null | VARCHAR(300) | Департамент, от имени которого размещается вакансия (если данная возможность доступна для компании). Работодатели могут запросить справочник департаментов. |
| department.id | string | INT | Идентификатор департамента |
| department.name | string | VARCHAR(300) | Название департамента |
| employment | object или null | VARCHAR(300) | Тип занятости. Элемент справочника employment. |
| employment.id | string | INT | Идентификатор типа занятости |
| employment.name | string | VARCHAR(300) | Название типа занятости |
| salary | object или null | VARCHAR(300) | Оклад |
| salary.from | number или null | INT | Нижняя граница вилки оклада |
| salary.to | number или null | INT | Верхняя граница вилки оклада |
| salary.gross | boolean или null | BOOLEAN(TRUE, FALSE, UNKNOWN) | Признак того что оклад указан до вычета налогов. В случае если не указано - null. |
| salary.currency | string | VARCHAR(50) | Идентификатор валюты оклада (справочник currency). |
| archived | boolean | BOOLEAN(TRUE, FALSE, UNKNOWN) | Находится ли данная вакансия в архиве |
| name | string | VARCHAR(100) | Название вакансии |
| insider\_interview | object или null | VARCHAR(300) | Интервью о жизни в компании |
| area | object | VARCHAR(300) | Регион размещения вакансии |
| area.id | string | INT | Идентификатор региона |
| area.name | string | VARCHAR(300) | Название региона |
| area.url | string | VARCHAR(300) | Url получения информации о регионе |
| initial\_created\_at | string | VARCHAR(300) | Дата и время создания вакансии |
| created\_at | string | VARCHAR(300) | Дата и время публикации вакансии |
| published\_at | string | VARCHAR(300) | Дата и время публикации вакансии |
| employer | object или null | VARCHAR(300) | Короткое представление работодателя. Описание полей смотрите в информации о работодателе. Может не прийти в случае, если вакансия анонимная |
| employer.blacklisted | boolean | BOOLEAN(TRUE, FALSE) | Добавлены ли все вакансии работодателя в список скрытых |
| response\_letter\_required | boolean | BOOLEAN(TRUE, FALSE) | Обязательно ли заполнять сообщение при отклике на вакансию |
| type | object | VARCHAR(300) | Тип вакансии. Элемент справочника vacancy\_type. |
| type.id | string | INT | Идентификатор типа вакансии |
| type.name | string | VARCHAR(300) | Название типа вакансии |
| has\_test | boolean | BOOLEAN(TRUE, FALSE) | Информация о наличии прикрепленного тестового задании к вакансии. В случае присутствия теста - true. |
| response\_url | string или null | VARCHAR(300) | На вакансии с типом direct нельзя откликнуться на сайте hh.ru, у этих вакансий в ключе response\_url выдаётся URL внешнего сайта (чаще всего это сайт работодателя с формой отклика). |
| test | object или null | VARCHAR(300) | Информация о прикрепленном тестовом задании к вакансии. В случае отсутствия теста — null. В данный момент отклик на вакансии с обязательным тестом через API невозможен. |
| test.required | boolean | BOOLEAN(TRUE, FALSE) | Обязательно ли заполнение теста для отклика |
| contacts | object или null | VARCHAR(300) | Контактная информация |
| billing\_type | object | VARCHAR(300) | Биллинговый тип вакансии. Элемент справочника vacancy\_billing\_type. |
| billing\_type.id | string | INT | Идентификатор биллингового типа вакансии |
| billing\_type.name | string | VARCHAR(300) | Название биллингового типа вакансии |
| allow\_messages | boolean | BOOLEAN(TRUE, FALSE) | Включена ли возможность соискателю писать сообщения работодателю, после приглашения/отклика на вакансию |
| premium | boolean | BOOLEAN(TRUE, FALSE) | Является ли данная вакансия премиум-вакансией |
| driver\_license\_types | array | VARCHAR(50) | Список требуемых категорий водительских прав. Список может быть пустым. |
| driver\_license\_types[].id | string | VARCHAR(50) | Категория водительских прав. Элемент справочника driver\_license\_types |
| accept\_incomplete\_resumes | boolean | BOOLEAN(TRUE, FALSE) | Разрешен ли отклик на вакансию неполным резюме |
| working\_days | object или null | VARCHAR(50) | Рабочие дни. Элемент справочника working\_days |
| working\_days.id | string | VARCHAR(50) | Идентификатор рабочих дней |
| working\_days.name | string | VARCHAR(50) | Название рабочих дней |
| working\_time\_intervals | object или null | VARCHAR(50) | Временные интервалы работы. Элемент справочника working\_time\_intervals |
| working\_time\_intervals.id | string | VARCHAR(50) | Идентификатор временного интервала работы |
| working\_time\_intervals.name | string | VARCHAR(50) | Название временного интервала работы |
| working\_time\_modes | object или null | VARCHAR(50) | Режимы времени работы. Элемент справочника working\_time\_modes |
| working\_time\_modes.id | string | VARCHAR(50) | Идентификатор режима времени работы |
| working\_time\_modes.name | string | VARCHAR(50) | Название режима времени работы |
| accept\_temporary | boolean или null | BOOLEAN(TRUE, FALSE, UNKNOWN) | Указание, что вакансия доступна для соискателей с временным трудоустройством |
| professional\_roles | array | VARCHAR(300) | Массив объектов профролей. Список может быть пустым. |
| languages | array или отсутствует | VARCHAR(300) | Список языков. Элементы справочника languages. |
| languages[].id | string | INT | Идентификатор языка. |
| languages[].name | string | VARCHAR(50) | Название языка. |
| languages[].level | object | VARCHAR(50) | Уровень знания языка. Элемент справочника language\_level. |
| languages[].level.id | string | VARCHAR(50) | Идентификатор уровня знания языка. |
| languages[].level.name | string | VARCHAR(50) | Название уровня знания языка. |

## Функциональный блок 4. Создание интерактивного дашборда.

## Функциональный блок 5. Отладка. Поиск и устранение ошибок и неисправностей в работе приложения.

# TO\_DO\_LIST:

1. Вставить блок с поисковым запросом
2. Описать работу скрипта при наличии ошибки загрузки в БД
3. Обновляемый скрипт (получить дату последней загрузки данных?)
4. Инкремент с нахлестом (1 день)