ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Профессор департамента программной инженерии факультета компьютерных наук, к.т.н

_____ Е. М. Гринкруг
«_____» _____ 2020 г.

УТВЕРЖДАЮ
Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук

_____ В. В. Шилов
«_____» _____ 2020 г.

Инв. № подл | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата

СИСТЕМА УПРАВЛЕНИЯ ЗАДАНИЯМИ ПО АВТОМАТИЧЕСКОМУ СБОРУ ДАННЫХ ИЗ СЕТИ ИНТЕРНЕТ

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.13-01 ТЗ 01-1-ЛУ

Исполнитель: студент группы БПИ 174
_____ Д. Ю. Редникина
«_____» _____ 2020 г.

Москва 2020

УТВЕРЖДЕН
RU.17701729.04.13-01 ТЗ 01-1-ЛУ

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СИСТЕМА УПРАВЛЕНИЯ ЗАДАНИЯМИ ПО АВТОМАТИЧЕСКОМУ СБОРУ ДАННЫХ ИЗ СЕТИ ИНТЕРНЕТ

Техническое задание

RU.17701729.04.13-01 ТЗ 01-1-ЛУ

Листов 16

Москва 2020

Содержание

1 Введение

1.1 Наименование программы

1.1.1 Наименование программы на русском языке

Система управления заданиями по автоматическому сбору данных из сети Интернет

1.1.2 Наименование программы на английском языке

System for managing tasks of collecting data from the Internet

1.2 Краткая характеристика области применения

Технологии web-scraping используются как в науке, так и в бизнесе - многие люди чувствуют потребность в извлечении данных из HTML разметки интернет страниц. Существующие аналоги реализуют базовый функционал(сбор данных), но не предоставляют такие дополнительные возможности как периодический запуск или совместное редактирование. Многие аналоги (прим. scrapyd [7]) имеют ограниченный функционал. Главные возможности, которыми продукт обеспечит предполагаемых пользователей:

- Совместное управление запусками клаулеров
- Периодический запуск задач
- Сбор логов, ошибок
- Группировка краулеров,а также их запусков в проект
- Бесплатная функциональность

2 Основания для разработки

2.1 Документы, на основании которых ведется разработка

Приказ декана факультета компьютерных наук И.В. Аржанцева "Об утверждении тем, руководителей курсовых работ студентов образовательной программы «Программная инженерия» факультета компьютерных наук"№ 2.3-02/1112-04 от 11.12.2019.

2.2 Наименование темы разработки

Наименование темы разработки – «Система управления заданиями по автоматическому сбору данных из сети Интернет » («System for managing tasks of collecting data from the Internet »)

Программа выполняется в рамках темы курсовой работы в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия» Национального исследовательского университета «Высшая школа экономики», факультет компьютерных наук.

3 Назначение разработки

3.1 Функциональное назначение

Система будет применяться как средство управления проектами по созданию, редактированию и запуску веб краулеров для сбора данных в сети интернет. Продукт позволит следить за запусками в режиме реального времени, а также создавать периодические запуски по расписанию.

3.2 Эксплуатационное назначение

Программа будет использоваться как инструмент для самостоятельной или совместной работы над проектами для запуска, управления сбора данных с помощью веб краулеров в сети интернет.

Таким образом, программный продукт позволит создавать, запускать образы пауков (см. 7.1) для сбора, управления, логирования и дальнейшего экспорта данных в целях сбора, изучения и мониторинга данных (см. 7.1).

4 Требования к программе

4.1 Функциональные требования

1. **Авторизация**
   Чтобы использовать сервис, клиентская программа должна иметь возможность авторизоваться в системе с помощью REST API

   (a) Для регистрации пользователю нужно указать следующие данные

      i. Почта - уникальна для каждого зарегистрированного пользователя;
      ii. Имя - длина больше 1 символ;
      iii. Логин - длина больше 2 символов;
      iv. Пароль - длина больше 2 символов;

   (b) Для авторизации пользователя в системе должны быть указаны следующие данные

      i. Почта;
      ii. Пароль;

2. **Проекты**
   Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

   (a) Создание проекта со следующей информацией

      i. Имя проекта;
      ii. Описание проекта - опциональное поле;

   (b) Обновление метаданных о проекте (редактирование) могут быть обновлены только участником с минимальным уровнем дотупа `ReadAndWrite`. Следующие данные могут быть обновлены:

      i. Имя проекта;
      ii. Описание проекта;
      iii. Настройки проекта для запуска краулеров;
      iv. Аргументы для запуска краулеров проекта;

   (c) Обновление `egg` файла проекта (редактирование) – минимальный уровень доступа участника, обновляющий данные о проекте `ReadAndWrite`.

   (d) Удаление данных о проекте. Удалить проект может только владелец `Owner`.

   (e) Просмотр списка проектов (с пагинацией), к которым у пользователя есть как минимум `ReadOnly` доступ.

3. **Участники проектов**
   Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

   (a) Просмотр информации об участниках проекта;

      i. Имя, почта, логин участника;
      ii. Статус участника в проекте (`ReadOnly`, `ReadAndWrite` или `Owner`);

(b) Обновление статуса участника проекта. Это действие совершать может только владелец проекта;

(c) Удаление участника из проекта. Данное действие может совершать только владелец проекта;

(d) Добавление нового участника с указанными правами на редактирование. Данное действие может совершать только владелец проекта;

4. **Краулеры**
Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

(a) Просмотр списка краулеров проекта;

(b) Редактирование информации о краулере для последующих запусков. Следующая информация может быть изменена

i. Настройки краулера для запуска;

ii. Аргументы для запуска;

5. **Запуски краулеров**
Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

(a) Просмотр списка запусков в определенном статусе (`Pending`, `Running` или `Finished`) с пагинацией, совершенных в проектах, к которым у пользователя есть как минимум `ReadOnly` доступ;

(b) Редактирование запуска - остановка запуска, перевод его в состояние `Finished`. Операция может быть применена только к запускам в состоянии `Running` или `Pending`;

(c) Удаление запуска - удаление всех данных о запуске из базы данных. Операция может быть применена только к запускам в состоянии `Finished`;

(d) Создание запуска со следующей информацией

i. Краулер, с которым происходит запуск;

ii. Настройки запуска – это могут быть как и предопределенные настроки на `scrapyd` [1], так и собственные настройки;

iii. Аргументы запуска – аргументы для запуска краулера, которые передаются через командную строку;

iv. Описание запуска;

6. **Периодические запуски**
Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

(a) Просмотр списка периодических запусков с пагинацией;

(b) Редактирование следующей информации о периодическом запуске

i. Настройки будущих запусков – это могут быть как и предопределенные настроки на `scrapyd`, так и собственные настройки;

[1] `http://doc.scrapy.org/en/latest/topics/settings.html`

ii. Аргументы будущих запусков – аргументы для запуска краулера, которые передаются через командную строку;
iii. Краулер, с помощью которого будет совершен запуск;
iv. cron-expression расписания запуска;

(c) Удаление периодического запуска;

(d) Отмена последующих запусков - перевод периодической задачи в состояние `Disabled`;

(e) Возобновление запусков - перевод периодической задачи в состояние `Enabled`;

(f) Создание периодического запуска со следующими данными
i. Название;
ii. Описание – опциональное;
iii. Краулер;
iv. Приоритетность, влияющая на очередь запусков (`Low`, `Normal` или `High`);
v. Статус (`Enabled` или `Disabled`);
vi. Настройки будущих запусков – это могут быть как и предопределенные настроки на `scrapyd`, так и собственные настройки;
vii. Аргументы будущих запусков – аргументы для запуска краулера, которые передаются через командную строку;

4.2 Требования к формату входных и выходных данных

1. В качестве входных данных сервер принимает REST [4] запросы от клиентских приложений, в теле которых передаются сериализованные в формате JSON [3] данные.
2. Сервер обрабатывает JSON [3] ответы от сервера `scrapyd` [7].
3. Сервер принимает информацию от базы данных PostgreSQL [6].

4.3 Условия эксплуатации

4.3.1 Климатические условия

Климатические условия должны сопадать с климатическими условиями эксплуатации устройства.

4.3.2 Требования к пользователю

Пользователь должен быть ознакомлен с документами «Руководство программиста «Система управления заданиями по автоматическому сбору данных из сети Интернет » и «Руководство пользователя «Система управления заданиями по автоматическому сбору данных из сети Интернет », а также разбираться в терминологии 7.1.

4.4 Требования к составу и параметру технических средств

Минимальный состав технических компонент, необходимый для нормального функционирования программы:

1. компьютер оснащенный процессором не ниже Intel Pentium/Celeron, или совместимый с ними с тактовой частотой не ниже 1,3 ГГц;
2. 1 Гб ОЗУ или более;
3. жесткий диск с объемом свободной памяти не менее 4 ГБ;
4. клавиатура;
5. доступ в интернет.

4.5 Требования к информационной и программной совместимости

Для нормального функционирования программы требуется компьютер, оснащенный следующими программными компонентами:

1. Ubuntu Server 18.04.2 LTS [8];
2. PostgreSQL 11 [6];
3. scrapyd [7];
4. Scala 2.12.6 [9];

4.6 Требования к маркировке и упаковке

Приложение должно быть доступно для установки из архива проекта, при скачивании из системы LMS НИУ ВШЭ [2].

5 Требования к программной документации

Состав программной документации должен включать в себя следующие компоненты:

1. Техническое задание «Система управления заданиями по автоматическому сбору данных из сети Интернет » (ГОСТ 19.201-78)
2. Программа и методика испытаний «Система управления заданиями по автоматическому сбору данных из сети Интернет » (ГОСТ 19.301-78)
3. Пояснительная записка «Система управления заданиями по автоматическому сбору данных из сети Интернет » (ГОСТ 19.404-79)
4. Руководство оператора «Система управления заданиями по автоматическому сбору данных из сети Интернет » (ГОСТ 19.505-79)
5. Текст программы «Система управления заданиями по автоматическому сбору данных из сети Интернет » (ГОСТ 19.401-78)

Вся документация должна быть составлена согласно ЕСПД (ГОСТ 19.101-77, 19.104-78, 19.105-78, 19.106-78 и ГОСТ к соответствующим документам (см. выше)) [1]. Все документы сдаются в электронном виде в составе курсовой работы LMS НИУ ВШЭ.

Пояснительная записка «Система управления заданиями по автоматическому сбору данных из сети Интернет » должна быть проверена на плагиат ($< 40\%$ заимствований). Документ, подтверждающий проверку Пояснительной записки сдается в печатном виде вместе с подписанным отзывом от научного руководителя.

6 Технико-экономические показатели

6.1 Предполагаемая потребность

Программа будет использоваться программистами, которые используют web-scraping (7.1) для отслеживания изменений, скачивания данных из сети интернет.

6.2 Ориентировочная экономическая эффективность

Полная функциональность главного аналога [5] не доступна для бесплатного использования.

Разрабатываемая система будет бесплатной и будет иметь англоязычный интерфейс.

7 Стадии и этапы разработки

7.1 Необходимые стадии разработки, этапы и содержание работ

1. *Техническое задание:*
   (a) Этапы разработки:
      i. Обоснование необходимости разработки программы;
      ii. Постановка задачи;
      iii. Сбор исходных материалов;
      iv. Выбор и обоснование критериев эффективности и качества разрабатываемой программы;
      v. Обоснование необходимости проведения научно-исследовательских работ;
   (b) Разработка и утверждение технического задания:
      i. Определение требований к программе;
      ii. Определение стадий, этапов и сроков разработки программы и документации на неё;
      iii. Согласование и утверждение технического задания;
2. *Технический проект:*
   (a) Разработка технического проекта:
      i. Уточнение структуры входных и выходных данных;
      ii. Разработка алгоритма решения задачи;
      iii. Определение формы представления входных и выходных данных;
      iv. Разработка структуры программы;
      v. Окончательное определение конфигурации технических средств.
   (b) Утверждение технического проекта:
      i. Разработка пояснительной записки;
      ii. Согласование и утверждение технического проекта.
3. *Рабочий проект:*
   (a) Разработка программы:
      i. Программирование и отладка программы.
   (b) Разработка программной документации:
      i. Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77 [1].
   (c) Испытания программы:
      i. Разработка, согласование и утверждение порядка и методики испытаний;
      ii. Корректировка программы и программной документации по результатам испытаний.

ПРИЛОЖЕНИЕ А

Используемые понятия и определения

Web scraping – это сбор данных с различных интернет-ресурсов. Общий принцип его работы можно объяснить следующим образом: некий автоматизированный код выполняет GET-запросы на целевой сайт и получая ответ, парсит HTML-документ, ищет данные и преобразует их в заданный формат.

Проект – сущность для объединения и предоставления доступа к запускам/краулерам/периодическим задачам.

Веб краулер – программа, являющаяся составной частью поисковой системы и предназначенная для перебора страниц Интернета с целью занесения информации о них в базу данных поисковика. Неотъемлемая часть проекта. Именно с помощью пауков пользователь может “краулить” сайты для сбора необходимой информации.

Запуск – единоразовый запуск краулера с настройками и аргументами, указанными для этого запуска.

Периодический запуск – запуск с множеством настроек, повторяющийся в определенные периоды времени (запуски по cron-expression).

ПРИЛОЖЕНИЕ В

Список источников

[1] Единая система программной документации – М.: ИПК, Издательство стандартов, 2000, 125 стр.

[2] LMS [Электронный ресурс] URL: `https://lms.hse.ru` (Дата обращения: 16.05.2020, режим доступа: свободный)

[3] JSON [Электронный ресурс] URL: `https://www.json.org` (Дата обращения: 16.05.2020, режим доступа: свободный)

[4] Representational state transfer URL: `https://en.wikipedia.org/wiki/Representational_state\_transfer` (дата обращения: 2020.12.14).

[5] Scrapinghub URL: https://scrapinghub.com (дата обращения: 2019.12.14).

[6] Postgresql [Электронный ресурс] URL:`https://www.postgresql.org` (Дата обращения: 16.04.2020, режим доступа: свободный)

[7] Github scrapyd/scrapyd [Электронный ресурс] URL: `https://github.com/scrapy/scrapyd` (Дата обращения: 16.04.2020, режим доступа: свободный)

[8] Ubuntu [Электронный ресурс] URL: `https://www.ubuntu.com` (Дата обращения: 16.04.2020).

[9] Scala [Электронный ресурс] URL: `https://www.scala-lang.org` (Дата обращения: 16.04.2020).

Лист регистрации изменений									
Изм.	Номера листов				Всего листов в документе	№ документа	Входящий № сопроводит. докум. и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					