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

Пояснительная записка

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

Пояснительная записка

RU.17701729.04.13-01 ТЗ 01-1-ЛУ

Листов 29

Москва 2020

Содержание

1 Введение

1.1 Наименование программы

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

1.2 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 Назначение и область применения

2.1 Назначение программы

2.1.1 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.

2.1.2 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. 4.1.5) для сбора, управления, логирования и дальнейшего экспорта данных в целях сбора, изучения и мониторинга данных (см. 4.1.5).

2.1.3 Область 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 [1]) 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

3 Технические характеристики

3.1 Постановка задачи на разработку программы

Программа должна соответствовать требованиям, представленным в техническом задании:

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;

Существует потребность в отслеживании информации в интернете в автоматическом режиме:

– Совместное управление запусками
– Периодический запуск задач
– Просмотр логов, различных метрик
– Бесплатная платформа

Цель работы:

– Разработка инструмента для управления заданиями по автоматическому сбору данных из сети Интернет
  – создание запросов по созданию и запуску (в том числе и по расписанию) краулеров
  – предоставление запросов для доступа к данным, собираемым различными краулерами
  – real-time мониторинг работы краулеров
  – инфраструктура для разработки новых краулеров

Задачи работы:

– Разработать сервер с помощью play-framework на языке scala
– Написать тесты на разрабатываемый функционал
– Написать техническую документацию к разрабатываемому ПО

3.2 Описание алгоритмов и функционирования программы

3.2.1 Описание алгоритмов программы

3.2.1.1 Алгоритм проверки наличия доступа

В разрабатываемой системе практически каждый запрос имеет ограничение по доступу: изменение в проект могут вносить только участники с `ReadAndWrite` или `Owner` доступом, `ProjectId` и `CrawlerId` для запуска должны совпадать, удалять проект может человек с `Owner` правами и т.д.

Для обработки таких ситуаций было принято решение создать `SecurityService`, который реализовывает алгоритм работы с несколькими проверками сразу и обработкой ошибок доступа.

Была реализована функция выполнения *nextAction* - следующего действия в базе данных по результату предыдущей 2. С помощью функции flatMap можно делать композицию функции и таким образом соединять несколько проверок в единое целое. [6]

Целью данного механизма было желание добиться легкого в изменении способа проверки нескольких условий одновременно с выводом читабельной ошибки, если где-то в середине что-то пойдет не так.

```
        accessRight(ReadAndWrite) >> jobToProject >> jobToStatus(Running)
```

Листинг 1 — Нестинг функций

На примере JobsController можно увидеть, как легко использовать такой механизм для проверки нескольких условий.

```
def checkUserProjectAndJob(userId: UUID,
                           projectId: Long,
                           jobId: Long,
                           userAccess: MembershipAccessRight = Readonly):
                           Future[Option[JobExecution]] = {

    val jobExecutionAction =
      hasPermissionToAccessProject(projectId, userId, userAccess)
      .flatMap(checkResultAndPerformNextAction(
            SecurityService.UserAccessMessage,
            jobExecCorrespondsToProject(projectId, jobId)))
      .flatMap(res => checkResultAndPerformNextAction(
            SecurityService.JobExecutionToProjectMessage,
            DBIO.successful(res))(res))

    db.run(jobExecutionAction)
  }
```

Листинг 2 — Функция для проверки доступа пользователя и запуска

3.2.2 Описание схемы функционирования программы

Диаграмма вариантов использования (1) отображает ключевые прецеденты и главных акторов системы. Первостепенный актор - обычный пользователь, описание и профиль которого представлены в разделах 4.1.4 и 4.1.3. Второстепенные акторы системы - база данных postgres [5], а также scrapyd [1]. Вот основные прецеденты, которые инициирует обычный пользователь в рамках системы :

1. Авторизация;

2. **CRUD работ по запуску**;
3. CRUD периодических задач;
4. Просмотр и редактирование пауков;
5. **CRUD проектов**;

Прецеденты 5 и 2 будут расмотрены детально в этом разделе на странице 12. Область применения разрабатываемой системы продемонстрирована на рисунке 2.

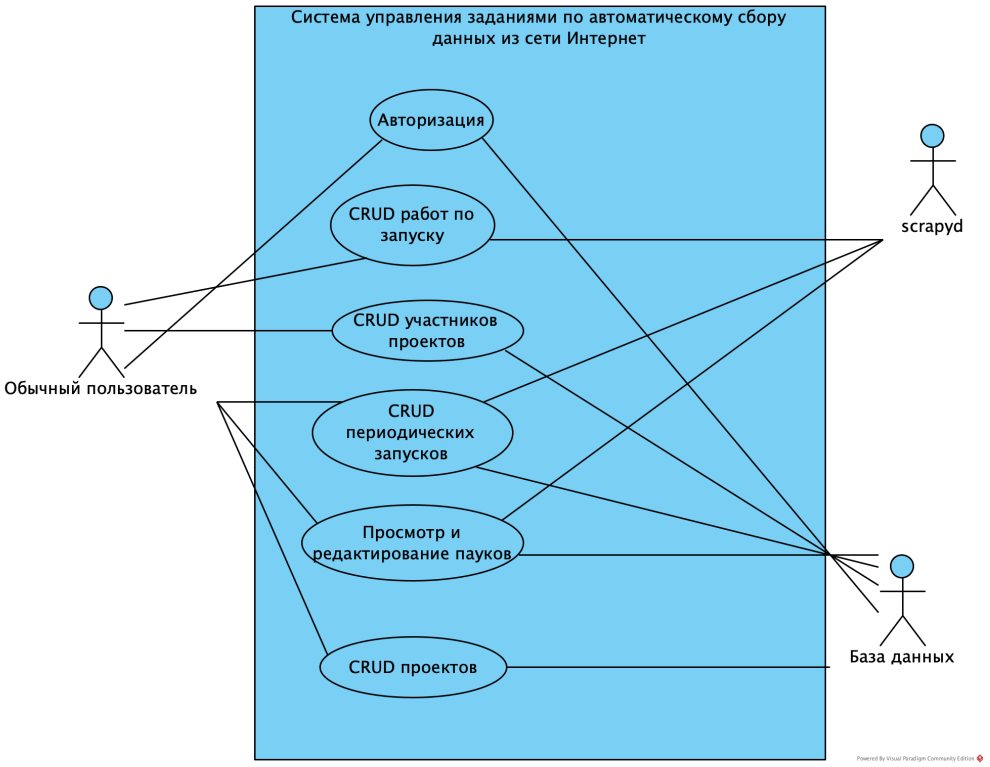

Рисунок 1 — Use-case диаграмма

приетв

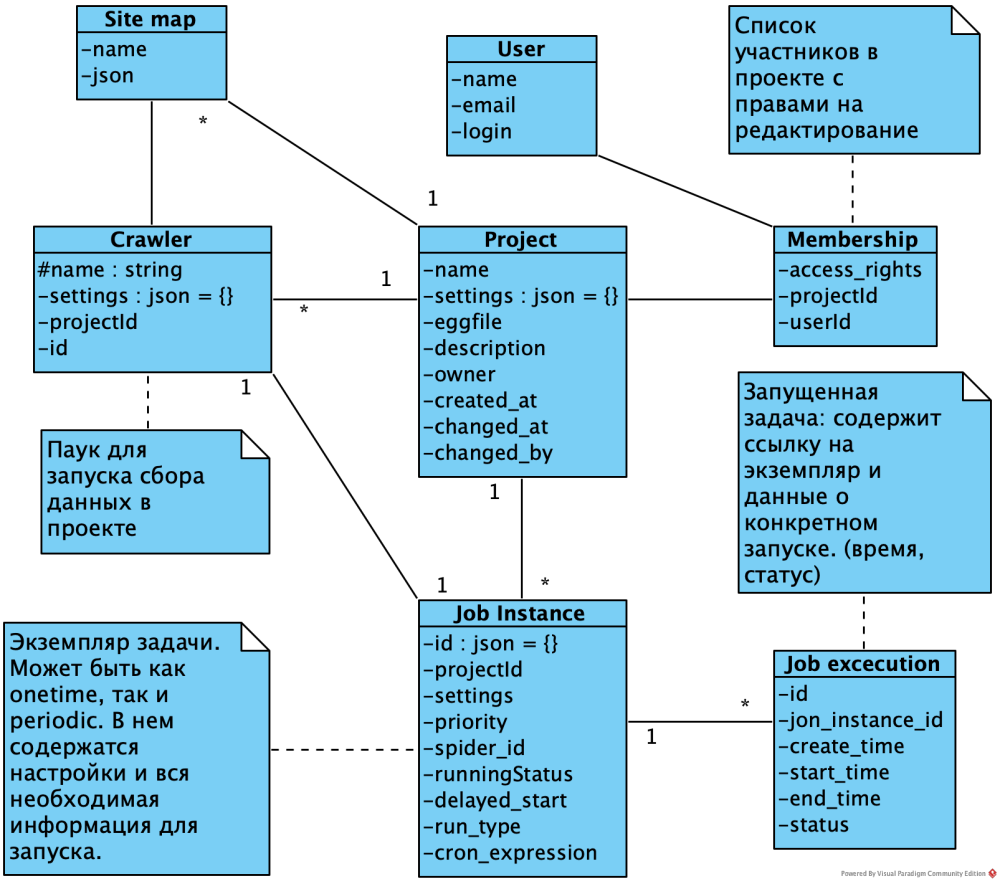

Рисунок 2 — Диаграмма области применения

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 ТЗ 01-1-ЛУ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Спецификации прецендентов

Спецификация прецендента «CRUD запусков»

Таблица 1 — Краткая информация о прецеденте

Название	«CRUD запусков»
Аннотация	Любой пользователь, имеющий доступ к проекту `Owner` или `ReadAndWrite`, может создать запуск из имеющихся в проекте краулеров. Запуск может иметь приоритет, а также настройки и аргументы для скрейпинга. В результате запуска пользователю будут доступны логи и результаты сбора.
Автор документа	Редникина Д.Ю.
Рамки применения	Вся система
Уровень	Ключевая задача
Основной исполнитель	Обычный пользователь

Основной поток

1. Пользователь начинает прецендент: отправляет запрос на просмотр данных о задачах в статусе `finished`.
2. Система формирует JSON со списком задач в статусе `finished`, к которым у пользователя есть доступ.
3. Пользователь отправляет запрос на создание нового запуска, при этом указав id паука для запуска, приоритетность запуска, фргументы и настройки для запуска. Данные должны быть в формате JSON 6.
4. Система валидирует полученные данные.
5. Система отправляет запрос на scrapyd для создания запуска.
6. Система создает новый запуск в статусе `pending` в базе данных.
7. Система формирует JSON с созданным запуском и информацией о нем.

Альтернативные потоки

Альтернативный поток 1

Условие начала В шаге 3 основного потока пользователь отправил JSON в неверном формате.

1. Система валидирует данные.
2. Система возвращает статус-код `BAD_REQUEST` с сообщением об неверном формате введенных данных.

Альтернативный поток 2

Условие начала В шаге 3 основного потока пользователь отправил запрос на создание запуска с краулером, находящимся в проекте id, к которому пользователь имеет доступ `ReadOnly`.

1. Система валидирует права доступа пользователя к проекту id.
2. Система возвращает статус-код `FORBIDDEN` с сообщением «You don't have permission to run a job».

Альтернативный поток 3

Условие начала В шаге 3 пользователь отправляет `CrawlerId` паука, для которого хочет совершить запуск и `ProjectId` проекта, к которому имеет `Owner` или `ReadAndWrite` доступ. Паука с `CrawlerId` нет в проекте `ProjectId`.

1. Система валидирует права доступа к проекту и наличие паука в проекте с данным `projectId`.
2. Система возвращает статус-код `FORBIDDEN` с сообщением «Project and crawler doesn't match».

Альтернативный поток 4

Условие начала На шаге 3 пользователь отправляет запрос на отмену запуска `id`, который находится в статусе `Pending` или `Running`.

1. Система валидирует данные: доступ пользователя к отмене задачи, а также статус задачи.
2. Система отправляет запрос на `scrapyd` для отмены запущенной задачи.
3. Система валидирует полученный результат.
4. Система изменяет статус задачи на `Finished`.
5. Система отправляет статус-код `OK` и `id` отмененной задачи.

Альтернативный поток 5

Условие начала На шаге 3 пользователь отправляет запрос на отмену запуска `id`, который находится в статусе `Finished`.

1. Система валидирует данные: доступ пользователя к отмене задачи, а также статус задачи.
2. Система возвращает статус-код `FORBIDDEN` с сообщением «Job has already finished».

Альтернативный поток 6

Условие начала На шаге 3 пользователь отправляет запрос на удаление запуска `id`, который находится в статусе `Finished`.

1. Система валидирует данные: доступ пользователя к отмене задачи, а также статус задачи.
2. Система удаляет данные о задаче в базе данных.
3. Система возвращает статус-код `OK`.

Альтернативный поток 7

Условие начала На шаге 3 пользователь отправляет запрос на удаление запуска `id`, который находится в статусе `Pending` или `Running`.

1. Система валидирует данные: доступ пользователя к отмене задачи, а также статус задачи.
2. Система возвращает статус-код `FORBIDDEN` с сообщением «Job is not in finished state».

Таблица 2 — Пред- и постусловия для прецендента

Предусловия	Пользователь авторизован в системе.
Постусловия	В системе зафиксированы совершенные запуски.
Специальные требования	Пользователь должен быть знаком с технологией web-scraping.
Список технологий	База данных Postgresql. Scrapyd.
Приоритет	Высокий
Открытые проблемы	

Спецификация прецендента «CRUD проектов»

Таблица 3 — Краткая информация о прецденте

Название	«CRUD проектов»
Аннотация	Любой пользователь системы имеет возможность создать проект. Проект - это сущность для организации краулеров, а также их запусков и периодических задач в одно целое. Над проектом может работать как один человек, так и несколько.
Автор документа	Редникина Д.Ю.
Рамки применения	Вся система
Уровень	Ключевая задача
Основной исполнитель	Обычный пользователь

Основной поток

Диаграмма базового потока представлена на рисунке 3.

1. Пользователь начинает прецендент.
2. Система формирует JSON со списком проектов, к которым у пользователя есть доступ.
3. Пользователь отправляет запрос на создание нового проекта, при этом указав название и опциональное описание проекта. Данные должны быть в формате JSON 3.
4. Система валидирует полученные данные (название и описание проекта).
5. Система создает новый проект с указанными данными и присваивает пользователю `Owner` права на доступ к проекту.
6. Система формирует JSON с созданным проектом и информацией о нем.

Альтернативные потоки

Альтернативный поток 1

Условие начала В шаге 3 основного потока пользователь отправил JSON в неверном формате.

1. Система валидирует данные.
2. Система возвращает статус-код `BAD_REQUEST` с сообщением об неверном формате введенных данных.

Альтернативный поток 2

Условие начала В шаге 3 основного потока пользователь отправляет запрос об изменении одного из вернувшихся на предыдущем шаге 2 проекта. Пользователь имеет `ReadAndWrite` права на редактирование этого проекта.

1. Пользователь отправляет JSON с данными, которые должны быть изменены о проекте, в формате 4 или 5.
2. Система валидирует доступ пользователя к изменяемому проекту и данные.
3. Система фиксирует внесенные пользователем изменения.
4. Система отображает статус-код операции `OK`.

Альтернативный поток 3

Условие начала В шаге 3 основного потока пользователь отправляет запрос об изменении одного из вернувшихся на предыдущем шаге 2 проекта. Пользователь имеет `ReadOnly` права на редактирование этого проекта.

1. Пользователь отправляет JSON с данными, которые должны быть изменены о проекте, в формате.
2. Система валидирует доступ пользователя к изменяемому проекту и данные.
3. Система возвращает статус-код операции `FORBIDDEN` с сообщением «You don't have permission to change project».

Альтернативный поток 4

Условие начала В шаге 3 основного потока пользователь отправляет запрос об удалении проекта. У пользователя `Owner` доступ к проекту.

1. Пользователь отправляет запрос с `id` проекта, который хочет удалить.
2. Система валидирует права доступа к проекту и наличие проекта с данным `id`.
3. Система удаляет проект из БД.
4. Система возвращает статус-код `OK`.

Альтернативный поток 5

Условие начала В шаге 3 основного потока пользователь отправляет запрос об удалении проекта. У пользователя `ReadOnly` или `ReadAndWrite` доступ к проекту.

1. Пользователь отправляет запрос с `id` проекта, который хочет удалить.
2. Система валидирует права доступа к проекту и наличие проекта с данным `id`.
3. Система возвращает статус-код `FORBIDDEN` с сообщением «You don't have permission to delete project».

Альтернативный поток 6

Условие начала В шаге 3 основного потока пользователь отправляет запрос об удалении проекта. У пользователя могут быть любые права доступа.

1. Пользователь отправляет запрос с `id` проекта, который хочет удалить.
2. Система валидирует права доступа к проекту и наличие проекта с данным `id`.
3. Система возвращает статус-код `FORBIDDEN` с сообщением «Project doesn't exist».

Таблица 4 — Пред- и постусловия для прецендента

Предусловия	Пользователь авторизован в системе.
Постусловия	В системе зафиксированы произведенные изменения проектов.
Специальные требования	Пользователь должен быть знаком с технологией web-scraping.
Список технологий	База данных Postgresql.
Приоритет	Высокий
Открытые проблемы	

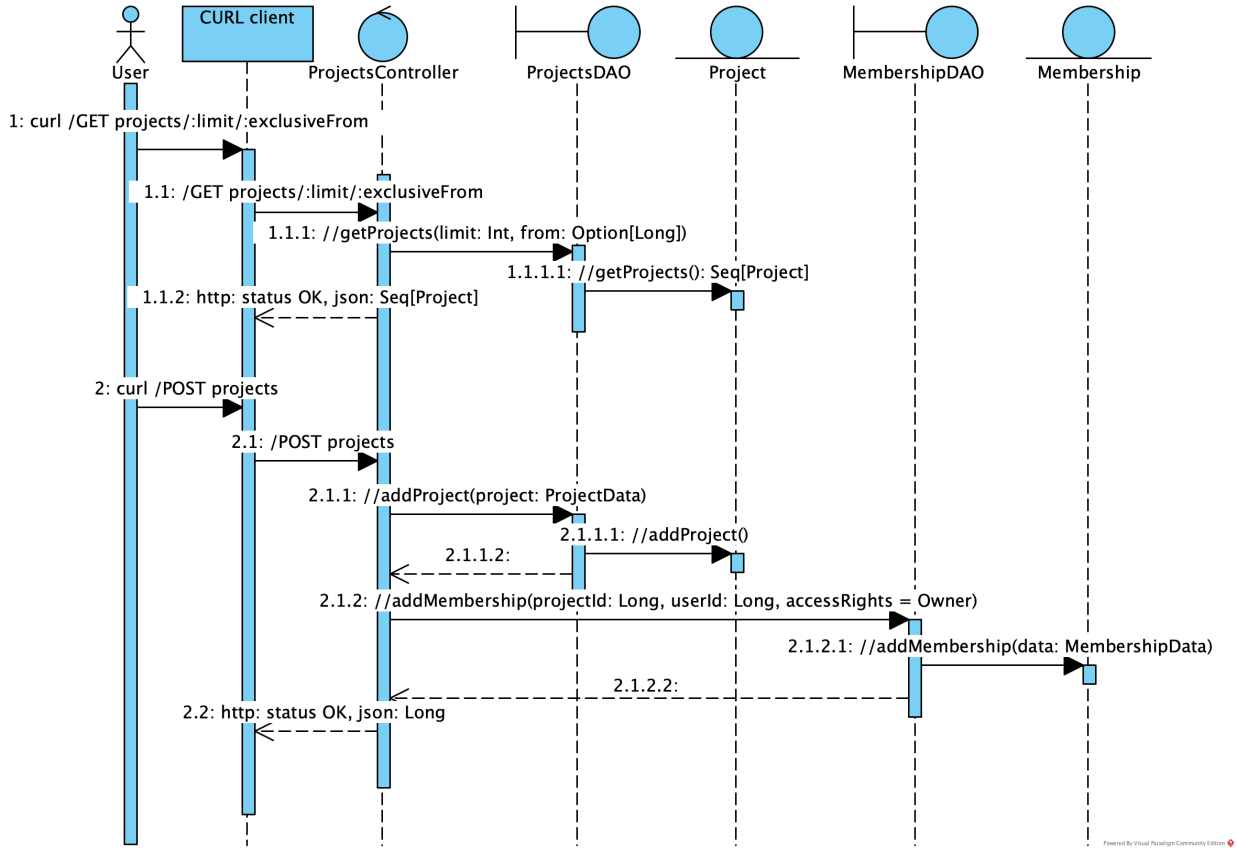

Рисунок 3 — Базовый поток прецендента «CRUD проектов»

3.2.3 Возможные взаимодействия программы с другими программами

Разрабатываемая программа взаимодействует с scrapyd [1]. Базовый функционал этого сервера используется для непосредственного запуска пауков.

3.3 Описание и обоснование выбора метода организации входных и выходных данных

Так как разрабатываемая программа является серверным приложением, то для формата входных данных был выбран JSON формат. Такая организация ввода позволяет быстро производить ввод данных, и препятствует ошибочному и некорректному заполнению данными полей базы данных.

Для организации выходных данных также был выбран JSON формат. Это позволяет клиентам легко интерпретировать и отображать данные, так как JSON - самый популярный формат для сериализации передаваемых данных в клиент-серверных приложениях [7].

3.4 Описание и обоснование выбора состава технических и программных средств

3.4.1 Состав технических и программных средств

При разработке программного продукта использовались следующие технические и программные средства:

- Язык разработки: Scala 2.13.1
- Фреймворк: Play-framework 2.6.13 [11]
- Среда разработки: IntelliJ IDEA 2019.2 (Ultimate Edition)
- Dependency manager: sbt
- MacBook (Retina, 12-inch, Early 2016)
- Операционная система: macOS Catalina Version 10.15.2
- Библиотеки, использованные при разрабоке:
  - play-silhouette 5.0.3
  - postgresql 42.2.8
  - play-slick 3.0.1
  - slick-pg 0.18.0
  - slick-pg_play-json 0.18.0
  - scala-guice 4.2.6
  - swagger-play2 1.6.1
  - scalacheck 1.13.5
  - scalatestplus-play 3.1.2
  - akka-quartz-scheduler 1.8.2-akka-2.6.x

3.4.2 Обоснование выбора библиотек

3.4.2.1 slick / play-slick / slick-pg

Slick - это современная библиотека для совершения запросов к базе данных и доступа к ней, реализованная на языке Scala. Она позволяет работать с данными из БД почти так же, как если использовать коллекции из стандартной библиотеки Scala, и в то же время дает полный контроль над тем, когда происходит доступ к базе данных и какие данные передаются. Преимущество библиотеки заключается в том, что можно писать запросы к базе данных на type-safe Scala вместо SQL, получая таким образом статическую проверку совместимости типов, безопасность во время компиляции и композиционность в Scala. Также в Slick есть расширяемый компилятор запросов, который может генерировать код для разных бэкэндов.

Play-slick - это библиотека, которая позволяет интегрировать Slick [10] в жизеннный цикл приложения в Play-framework [11], а также поддерживает play-эволюции [2].

Slick-pg - надстройка над Slick [10], которая позволяет работать с Jsonb типом и enum в Postgresql [5].

3.4.2.2 play-silhouette % tests

Библиотека Silhouette - это библиотека аутентификации для приложений Play Framework [11], которая поддерживает несколько методов аутентификации, включая OAuth1, OAuth2, OpenID, CAS, Credentials, Basic Authentication или пользовательские схемы аутентификации [8]. При проектировании приложения было принято решение использовать механизм аутентификации через Cookie - библиотека прекрасно подошла. Более того, существует удобный механизм для тестирования api-запросов, написанный с помощью `SecuredActions` библиотеки Silhouette - Silhouette-Testing. Также библиотека поддерживает механизм авторизациии через социальные сети - таким образом в будущем при расширении функционала приложения можно будет легко встроить механизм такой авторизации. [9].

3.4.2.3 akka-quartz-scheduler

Для запуска периодических задач были рассмотрены следующие библиотеки, подходящие для использования в Play-framework [11]. Основное требование к библиотеке - запуски по cron-expression на долговременный период.

Таблица 5 — Рассмотренные библиотеки для периодического запуска

Библиотека	**Плюсы**	**Минусы**
Scala's tasks	Поддерживается	The Akka Scheduler не предназначен для использования на длительные периоды запусков, также не гарантирована точность запусков.

[2]Эволюции не использовались при разработки системы

Monix + cron4s	Хорошая и популярная библиотека.	Очень много зависимостей. Пришлось бы писать много кода для соединения с cron4s.
akka-quartz-scheduler	Иделаьно подходит под задачи проекта. Есть работа с cron-expression.	Не работает на более новых версиях Play-framework [11] ($>$ 2.6.13)
fs2	Библиотека для асинхронных запусков в функциональном стиле.	Не предназначена для данного вида задач, придется писать обертку. Нет работы с cron-expression.
play-akkjobs	На официальном сайте Play-framework ссылка на эту библиотеку как на хороший scheduler	Не поддерживается

По итогам сравнения была выбрана библиотека akka-quartz-scheduler.

3.4.3 Обоснование выбора языка программирования

3.4.3.1 Scala

Scala объединяет объектно-ориентированное и функциональное программирование в одном емком языке высокого уровня. Статические типы Scala помогают избежать ошибок в сложных приложениях, а среда выполнения JVM позволяет создавать высокопроизводительные системы с легким доступом к огромным экосистемам библиотек.

3.4.4 Обоснование выбора шаблона проектирования

3.4.4.1 MVC

Так как разработка велась в фреймворке Play [11], то был использован подход с паттерном MVC [12] для работы с HTTP-запросами. MVC представляет Controllers, Views[3], Actions - для работы с api-запросами.

3.4.4.2 Слоистая архитектура проектирования

При разработке использовался архитектурный подход - layers. Все приложение поделено на несколько слоев (см. рисунок 4)

[3]Так как разрабатывалось серверное приложение, то views не использовались

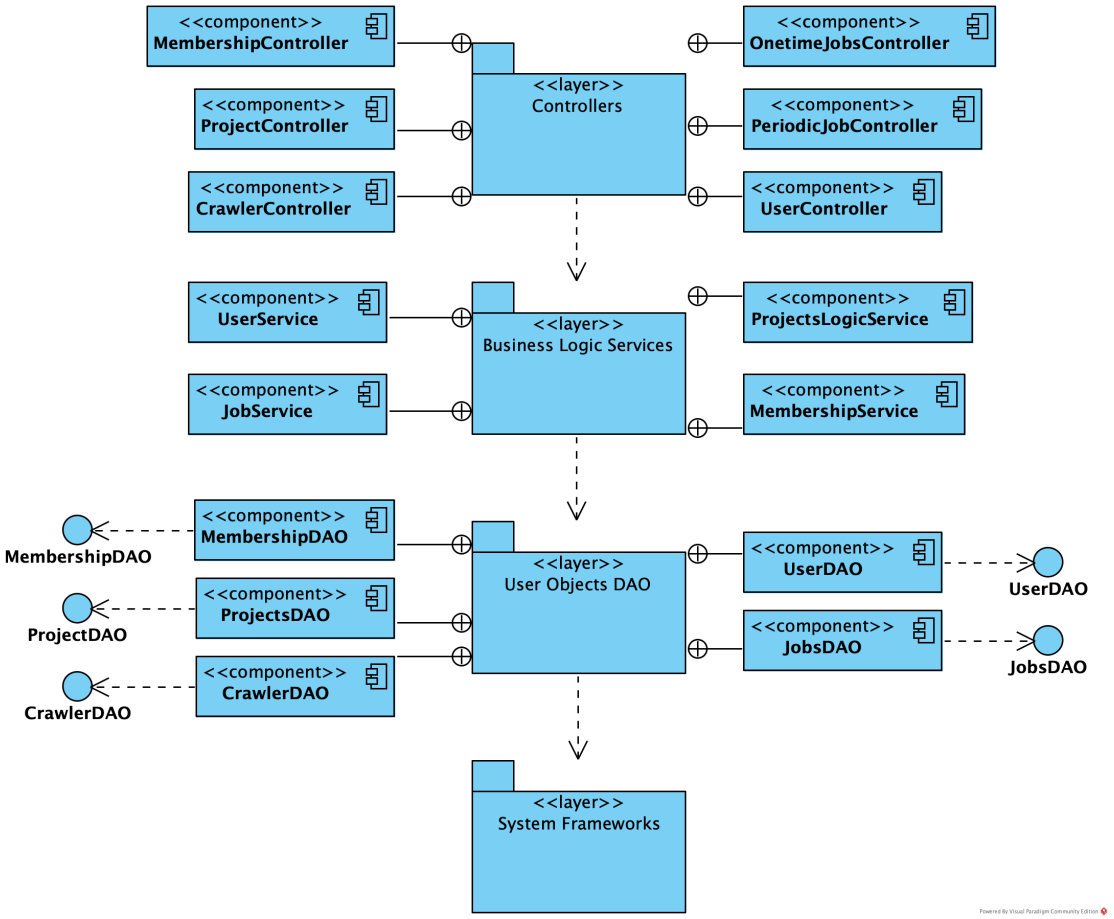

Рисунок 4 — Пример слоистости в приложении

3.4.5 Обоснование выбора базы данных

3.4.5.1 PostgreSQL

PostgreSQL[5] – свободная объектно-реляционная система управления базами данных с открытым исходным кодом.

База данных позволяет хранить большое количество связанной и структурированной информации и эффективно производить манипуляции с ней.

Postgres также отличается хорошей поддержкой сериализации и индексации произвольных JSON [7] объектов, что сильно повышает гибкость в ее применении.

Взаимодействие с базой данных производится на диалекте языка SQL. В рамках разработки большинство взаимодействия производилось посредством библиотеки play-slick ([10]) для строгой типизации и уменьшения ошибок со стороны программиста.

4 Технико-экономические показатели

4.1 Предполагаемая потребность

4.1.1 Устройство рынка

Рынок веб-скрейпинга составляют как коммерческие продукты, а также их open source аналоги. Web-scraping - это технология и методы, используемые стартапами, небольшими и крупными компаниями, которые делают возможным быстрое извлечение данных и информации из сети интернет и их обработку. Также технологии извлечения данных широко применяются в науке, образовании учебными, студентами и программистами.

Таким образом, заинтересованных лиц на рынке выделить сложно - слишком большому кругу компаний сейчас требуется воспользоваться извлечением информации из интернета. Разрабатываемый продукт планируется сделать бесплатной площадкой для использования технологий web-scraping и доступной любому пользователю.

4.1.2 Пользовательская среда

Продукт будет применяться пользователями в рабочих, научных и исследовательских целях. Продуктом пользоваться можно будет как водиночку: запуск краулеров и сбор данных не требует дополнительных человеческих ресурсов, но также можно будет предоставлять доступ для наблюдения и редактирования запусков другим людям.

Задачи, которые решают пользователи данной системой давольно обширные. Это может быть мелкая проблема, например как «сбор данных о книгах с сайта», а вот примеры наиболее популярных крупных проблем:

– Сбор данных о продуктах и ценах для сравнения

– Сбор списков недвижимости

– Сбор данных для исследований

Перечисленный список задач компаниям и ученым приходится делать вручную или с использованием уже существующих аналогов - с этим они сталкиваются на регулярной основе. На данный момент существует много платных веб-сервисов, а также open source расширение для браузера.

4.1.3 Список пользователей

Роль пользователя продукта	**Описание**	**Способ работы с продуктом**	**Представители интересов в процессе разработки**
Обычный пользователь	Компания, заинтересованная в сборе данных; программист; студент;	Применение для сбора данных в учебных целях/ рабочих целях (мониторинг сайтов)/исследованиях	Представитель заказчика

Таблица 6 — Список пользователей

4.1.4 Профили пользователей

Категория пользователя	*Обычный пользователь*
Описание	Пользователь, заинтересованный в сборе данных с помощью технологии web-scraping
Представители	Компания, заинтересованная в сборе данных; стартап; программист; студент; научный сотрудник; аналитик
Уровень компетентности	Определенный уровень знаний в области сбора данных и написании пауков для сбора данных, пользователь ПК
Обязанности	– Использование продукта в целях сбора данных – Мониторинг логов, ошибок – Запуск периодических работ и мониторинг результатов
Критерий удовлетворенности продуктом	Продукт удовлетворяет потребности в мониторинге и сборе данных
Степень вовлеченности	Полная вовлеченность
Ожидаемые артифакты	Логи, собранные данные

Таблица 7 — Профили пользователей

4.1.5 Экономические преимущества по сравнению с отечественными и зарубежными аналогами

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. Разрабатываемая платформа предоставит бесплатный доступ к слудующему функционалу:

– Совместное управление запусками клаулеров

– Периодический запуск задач

– Сбор логов, ошибок

– Группировка краулеров,а также их запусков в проект

– Бесплатная функциональность

На рынке представлены следующие аналоги разрабатываемому планировщику заданий по запуску краулеров:

Системы с открытым исходным кодом

scrapymon

Простенький интерфейс для обзора задач и пауков над scrapyd. Нет поддержки планировщика. Нет дополнительного функционала, кроме базового запуска задач. Работает поверх scrapyd.

gerapy

Работает поверх scrapyd. Api для запуска задач предоставляет cron-формат для периодического запуска.

scrapydweb

У продукта есть lgpl-лицензия, есть парсинг логов, удобный UI-интерфейс. Api для запуска задач предоставляет cron-формат для периодического запуска. Работает поверх scrapyd.

ScrapyKeeper

У продукта есть парсинг логов, удобный UI-интерфейс. Есть архивация версий и доступ к проектам. Api для запуска задач предоставляет cron-формат для периодического запуска. Работает поверх scrapyd.

Проприетарные системы

Scrapinghub.com

Платная платформа, которая предоставляет возможность запускать периодические задачи, разрешает совместный доступ к проектам.

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

Листинги

```
{
    "name": "some name",
    "description": "optional description"
}
```

Листинг 3 — JSON for POST /projects

```
{
    "name": "some name",
    "description": "optional description",
    "spiderSettings": "{}",
    "spiderArgs": "{}"
}
```

Листинг 4 — JSON for PUT /project

```
{
    "eggFile": <egg file>
}
```

Листинг 5 — JSON for PUT /deploy

```
{
    "crawlerId": "CrawlerName",
    "priority": "Normal",
    "args": "{}",
    "settings": "{}"
}
```

Листинг 6 — JSON for POST /jobs

ПРИЛОЖЕНИЕ С

Описание классов, структур, методов, полей

Контроллеры

ApplicationController.scala	API методы для доступа к авторизации
CrawlersController.scala	API методы для доступа к паукам
JobsController.scala	API методы для доступа к запускам
MembershipController.scala	API методы для доступа к участникам проекта
PeriodicJobsController.scala	API методы для доступа к периодическим запускам
ProjectsController.scala	API методы для доступа к проектам
SignInController.scala	API методы для доступа к авторизации
SignUpController.scala	API методы для доступа к регистрации

Сервисы

JobService.scala	Сервис для управления запусками
MembershipService.scala	Сервис для управления
ProjectService.scala	Сервис для управления проектами
ScrapydService.scala	Сервис для отправки запросов в scrapyd [1]
SecurityService.scala	Сервис для управления безопасностью доступа
UpdaterService.scala	Сервис для управления обновлениями и синхронизацией со scrapyd [1]
UserService.scala	Сервис для управления пользователями

DAO

CrawlersDAO.scala	Класс паук-объект базы данных
JobDAO.scala	Класс запуск-объект базы данных
MembershipDAO.scala	Класс участник-объект базы данных
PasswordDAO.scala	Класс пароль-объект базы данных
ProjectDAO.scala	Класс проект-объект базы данных

Список источников

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

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

[3] ScalaTest+Play [Электронный ресурс] URL:`http://www.scalatest.org/plus`(Дата обращения: 16.04.2020, режим доступа: свободный)

[4] Testing - silhouette [Электронный ресурс] URL:`https://www.silhouette.rocks/docs/testing` (Дата обращения: 16.04.2020, режим доступа: свободный)

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

[6] FlatMap [Электронный ресурс] URL:`https://www.scala-lang.org/api/current/scala/collection/View/FlatMap.html` (Дата обращения: 16.04.2020, режим доступа: свободный)

[7] JSON - Использование [Электронный ресурс] URL:`https://ru.wikipedia.org/wiki/JSON` (Дата обращения: 16.04.2020, режим доступа: свободный)

[8] Silhouette - документация [Электронный ресурс] URL: `https://www.silhouette.rocks/docs` (Дата обращения: 16.04.2020, режим доступа: свободный)

[9] Silhouette-testing - документация [Электронный ресурс] URL: `https://www.silhouette.rocks/docs/testing` (Дата обращения: 16.04.2020, режим доступа: свободный)

[10] Slick [Электронный ресурс] URL: `https://scala-slick.org`(Дата обращения: 16.04.2020, режим доступа: свободный)

[11] Play-framework [Электронный ресурс] URL:`https://www.playframework.com` (Дата обращения: 16.04.2020, режим доступа: свободный)

[12] Package play.mvc [Электронный ресурс] URL: `https://www.playframework.com/documentation/2.6.0/api/java/play/mvc/package-summary.html` (Дата обращения: 16.04.2020, режим доступа: свободный)

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