Данные из *Kafka* параллельно с записью в *MongoDB* записываются в несколько потоков в *InfluxDB*.

Использование *InfluxDB* позволяет строить временные ряды метрик в *Grafana*. Это открывает возможности поиска аномалий в данных с помощью инструментов для *InfluxDB*.

Таким образом, на основе проведенного анализа архитектур скоростной передачи данных было выбрано оптимальное решение в виде архитектуры нулевой задержки. Данный вид архитектуры позволит передавать большие потоки данных за короткий промежуток времени без потери сообщений. Программные разработки, имеющие такую архитектуру, способны наиболее эффективно решать задачи по автоматизации мониторинга технологических процессов и рабочего оборудования промышленного производства в реальном времени.

Литература

1. Технология сегментированного кластера MongoDB // Русские Блоги / Информация для разработчиков, 2020. – Режим доступа: https://rus_sianblogs.com/ article/9754961584. – Дата доступа: 23.03.2023.
2. Spring Boot 3.0.4 / VMWare VMware, Inc. or its affiliates, 2023. – Режим доступа: https://spring.io/projects/spring-boot. – Дата доступа: 23.03.2023.

WEB-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ СКЛАДА НЕФТЕПРОДУКТОВ

А. В. Ёжиков

Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь

Научный руководитель Н. В. Ермалинская

Представлены результаты разработки функциональных требований, выбора технологий реализации и обоснования архитектуры проектируемого web-приложения для автоматизации работы склада нефтепродуктов.

Ключевые слова: web-приложение, учетные процессы, склад нефтепродуктов.

В связи с возрастающими количеством транспорта и потребностью в передвижении увеличиваются темпы потребления, а следовательно, и производства нефтепродуктов. Компании, имеющие свои нефтебазы и склады горюче-смазочных материалов, сталкиваются с тем, что большинство складов нефтепродуктов используют в своей работе оборудование и технологии, не обеспечивающие должный уровень технологической и экологической безопасности и точности учета. Внедрение систем комплексной автоматизации должно решить эти проблемы, повысить производственную и финансовую эффективность эксплуатируемых объектов и сократить до минимума экологические риски.

Таким образом, целью проводимых исследований является разработка web-приложения для автоматизации работы склада нефтепродуктов.

Одна из самых главных задач, которая осуществляется сотрудниками комплекса устройств или сооружений для хранения, приема и транспортировки нефти – это количественный учет нефтепродуктов. В обязанности специалистов по учетным мероприятиям входят: оформление документов во время приема нефтепродуктов и количественный анализ; при отгрузке или отправке нефтеперерабатывающей продукции

оформление соответствующих бумаг; количественный разбор остатков в емкостях, резервуарах нефтепродуктов; учет дефицита и избытка, а также документирование проверок, принимаемых и имеющихся в наличии нефтепродуктов.

Таким образом, в рамках проектируемого web-приложения автоматизации будут подлежать процессы учета: приема нефтепродуктов, доставляемых на склад; хранения нефтепродуктов в резервуарах и в тарных хранилищах; снабжения автозаправочных станций нефтепродуктами; контроля качества нефтепродуктов; содержания автомобильного парка в исправном состоянии и пр.

Функционально проектируемое web-решение должно выполнять следующие требования: 1) мониторинг количества нефтепродуктов в резервуарах; 2) заказ нефтепродуктов на НПЗ; 3) внесение записей о получении нефтепродуктов; 4) работа с заявками от заказчиков; 5) планирование рейсов; 6) списание нефтепродуктов; 7) формирование накладных; 8) внесение записей о проверках нефтепродуктов; 9) авторизация и аутентификация каждого работника при помощи логина и пароля; 10) внесение записей о ремонтах автоцистерн; 11) формирование сводного баланса по каждому виду топлива с указанием количества хранимого топлива на начало заданного периода, топлива поступившего за заданный период, топлива отпущенного за заданный период, а также хранимого топлива на окончание заданного периода; 12) наличие личного кабинета администратора с возможностью формирования списка работников (менеджер по учету, лаборант, начальник автопарка, директор склада нефтепродуктов) в соответствии с их функциями, а также всеми правами пользователей.

С учетом особенностей организации и ведения учетных процессов на складе нефтепродуктов проектирование информационной системы должно предусматривать выделение в ней нескольких уровней: *нижнего* – уровня микроконтроллеров и управляющих устройств; *среднего* – АРМ операторов и программно-аппаратных средств, управляющих нижним уровнем; *верхнего* – программных средств, обеспечивающих прием и регистрацию текущей информации о реализации и приеме нефтепродуктов, документирование основных товарных операций, автоматическое формирование учетных операций и пересчет «книжных» остатков нефтепродуктов, формирование отчетных документов о движении нефтепродуктов за выбранный период на основании зарегистрированных учетных операций, подготовку и передачу информации во внешнюю учетную систему.

В данном случае рассматривается разработка верхнего уровня такой комплексной системы управления, а именно web-приложения для учета движения, приемки и сбыта продукции склада нефтепродуктов.

Для возможного дальнейшего расширения приложения, добавления нового функционала, а также объединения приложения в цельную автоматизированную систему нефтяной компании наиболее подходящим является архитектурный стиль *REST*, являющийся набором принципов взаимодействия компьютерных систем, основанный на методах протокола *HTTP*. Также к преимуществам *REST* можно отнести гибкость, которая дает возможность обрабатывать различные типы запросов и форматы данных. На рис. 1 приведена схема работы *REST API*.

Для реализации серверной части приложения был выбран *Django* – это фреймворк для создания web-приложений с помощью языка программирования *Python*. *Django* позволяет быстро создавать безопасные и поддерживаемые web-приложения. *Django* помогает избежать многих распространенных ошибок безопасности, предоставляя фреймворк, разработанный для обеспечения автоматической защиты сайта. *Django* предоставляет безопасный способ управления учетными записями пользователей и паролями, избегая распространенных ошибок, таких как размещение инфор-

мации о сеансе в файлы *cookie*, где она уязвима (вместо этого файлы *cookie* содержат только ключ, а фактические данные хранятся в базе данных) или непосредственное хранение паролей вместо хэша пароля [1].

Фреймворк *Django* реализует архитектурный паттерн *Model-View-Template* (*MVT*), который по смыслу является аналогом более распространенного паттерна *MVC*.

Для написания *RESTAPI* используется библиотека *Django REST Framework*, которая ускорит и упростит разработку. *Django REST Framework* предоставляет разработчику весь необходимый набор инструментов для создания *REST*-сервисов на основе *Django*. По сути, *DRF* – это коллекция предустановленных классов. *DRF* дает инструменты для решения штатных задач, возникающих при создании *REST API* [2].

В качестве базы данных выбрана *PostgreSQL*. Основными преимуществами данной базы данных является то, что она бесплатная и не уступает платным аналогампо предоставляемым функциональным возможностям.

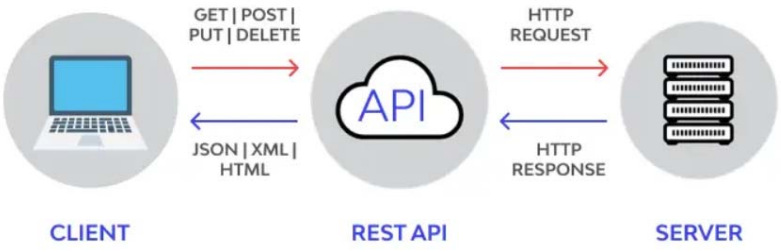

Рис. 1. Схема работы *RESTAPI*

Для реализации авторизации и аутентификации в приложении используется механизм на основе *JWT* токенов. Схема аутентификации представлена на рис. 2.

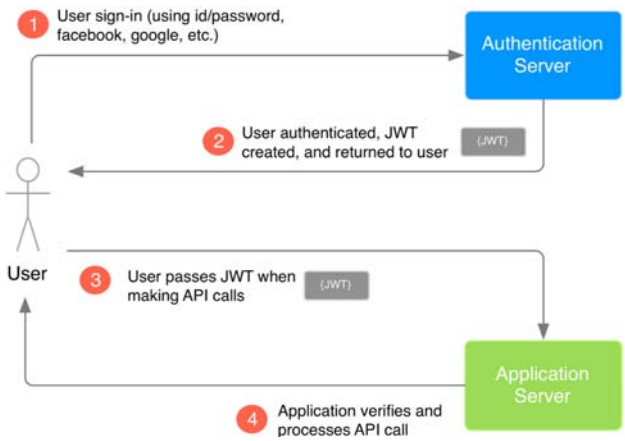

Рис. 2. Схема аутентификации на основе *JWT* токенов

Для разработки пользовательского интерфейса решено использовать *single-pageapplication*. При использовании данного подхода web-приложение состоит из одной *HTML*-страницы. Приложение подключается к серверу только один раз, а затем просто динамически подгружает и обновляет данные. Для создания интерфейса, следуя *SPA* подходу, была выбрана *JavaScript*-библиотека с открытым исходным кодом *React*.

Схема разрабатываемого web-приложения приведена на рис. 3.

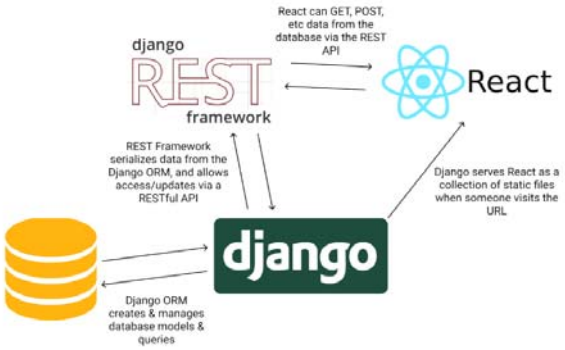

Рис. 3. Общая схема web-приложения

Практическое использование проектируемого web-приложения позволит обеспечить повышение эффективности учета, приемки и сбыта нефтепродуктов за счет автоматизации ключевых учетных процессов: контроля остатков в резервуарах, формирования сводного баланса и иных оперативных документов, внесения записей о результатах проверками качества нефтепродуктов, содержания в исправном состоянии автомобильного парка. Разработанное web-приложение может использоваться как самостоятельно, так и быть интегрировано в автоматизированный программный комплекс нефтяной компании.

Литература

1. Веб-фреймворкDjango (Python) / Сайт проекта по обучению веб-разработке MDN. – 2023. – Режим доступа: https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Introduction. – Дата доступа: 03.04.2023.
2. Django REST Framework / Сайт проекта по обучению веб-разработке «Runebook.разработчик». – 2023. – Режим доступа: https://runebook.dev/ru/docs/django_rest_framework. – Дата доступа: 04.04.2023.

ВЗАИМОДЕЙСТВИЕ С 1С ПРИЛОЖЕНИЕМ В РАЗРАБОТКЕ WEB-СЕРВИСА ДЛЯ УПРАВЛЕНИЯ ИНФОРМАЦИОННЫМ ОБЕСПЕЧЕНИЕМ ЗАДАЧ КАФЕДРЫ

А. Ю. Пищук

Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь

Научный руководитель Т. Л. Романькова

На многих предприятиях используются 1С приложения. Часто требуется разработать приложения на более новых технологиях. Для этого требуется разработать коннектор между 1С базой данных и новым приложением для сохранения данных. Целью данной работы является разработка web-сервиса, который получает данные из 1С базы данных университета и предоставляет данные в json.

Ключевые слова: 1С приложение, коннектор, микросервисная архитектура, контейнеры.