

Схема разрабатываемого web-приложения приведена на рис. 3.

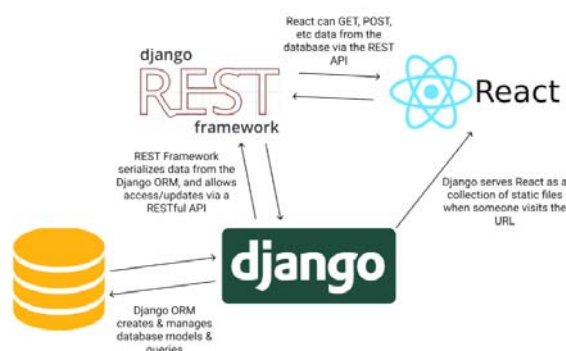


Рис. 3. Общая схема web-приложения

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-фреймворк Django (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.

В странах СНГ широко распространены 1С приложения. Основное преимущество 1С – это гибкость настроек готовых (типовых) программ под специфику любого предприятия. Любой программный продукт легко адаптируется и дорабатывается под нужды пользователей. Программа 1С позволяет быстро обрабатывать все первичные документы, поступающие от поставщиков и покупателей, рассчитывать зарплату сотрудникам, организовать систему платежей в соответствии с графиком платежей на предприятии, вести учет основных средств и нематериальных активов. Вся первичная информация быстро структурируется и затем может быть использована в разных отчетах. Каждая новая версия программы 1С сводит к минимуму ручной труд и дает возможность автоматизированной работы всей бухгалтерии. Например, можно создавать новые печатные формы, формы отчетов и обработки, интегрировать программу 1С с сайтами, интернет-магазинами, другими учетными системами.

Выделим минусы 1С приложений:

- медлительность некоторых вычислений. Нельзя вести динамический учет;
- программа в целом универсальна. Но существует около 5 % узкопрофильных предприятий, для которых она не подойдет;
- повышенная ресурсоемкость. Система относится к категории «прожорливых продуктов». Она использует внушающее количество системных ресурсов. Чтобы все нормально функционировало, ПК должен обладать достаточно высокими характеристиками;
- сложность интерфейса. В целом программа является простой, но чем больше пользователю нужно производить расчетов, тем сложнее она будет казаться, ведь это напрямую связано с количеством информации, которая поступает в систему. Соответственно, возрастает количество документов и отчетов;
- отсутствует свобода творчества. Если пользователь нестандартно выполнит проводку, либо решит задачу не так, как нужно, можно столкнуться с серьезными проблемами. В программу встроен конкретный механизм с перечнем процедур, и шаг влево и шаг вправо подразумевает сбой.

Поэтому многие предприятия принимают решение перейти на более новые технологии, которые предоставляют более широкий функционал.

Для перехода на новые технологии нужно сохранить данные, которые были использованы в старом приложении. Также часто требуется поддерживать старое приложение. Поэтому для перехода нужен коннектор между 1С Приложением и новым приложением.

Рассмотрим задачи кафедры:

- автоматизация распределения нагрузки по преподавателям;
 - автоматизация ведения плана преподавателя;
 - автоматизация учета материального обеспечения кафедры.
- Список функциональных требований к приложению:
- получение данных из 1С приложения;
 - синхронизация данных с базой 1С;
 - преобразование данных в нужный вид;
 - просмотр, добавление, изменение, удаление информации об университете;
 - наличие личного профиля преподавателя с информацией об пользователе – логин, пароль (его возможность изменения);
 - предоставление разнovidной статистики по университету.

Список нефункциональных требований к приложению:

- хранение и обработка большого количества данных;
- полиморфный пользовательский интерфейс;
- привилегированный доступ к ресурсам.

На рис. 1 представлена схема приложения для управления информационным обеспечением задач кафедры.

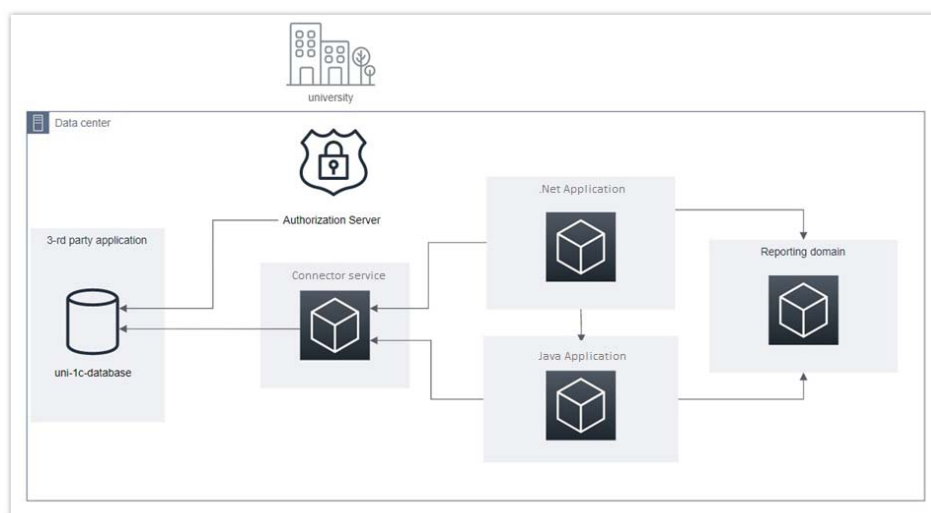


Рис. 1. Схема приложения для управления информационным обеспечением задач кафедры

Микросервисная архитектура – вариант сервис-ориентированной архитектуры программного обеспечения, направленный на взаимодействие насколько это возможно небольших, слабо связанных и легко изменяемых модулей – микросервисов, получивший распространение в середине 2010-х гг. в связи с развитием практик гибкой разработки и DevOps.

В микросервисной архитектуре приложение разбивается на ряд независимо развертываемых сервисов, которые взаимодействуют с помощью API-интерфейсов. Благодаря такому подходу каждый отдельный сервис можно развертывать и масштабировать независимо от других. В результате команды могут быстрее и чаще предоставлять объемные и сложные приложения. В отличие от монолитного приложения, с микросервисной архитектурой команды могут быстрее внедрять новые возможности и вносить изменения, при этом им не приходится переписывать большие фрагменты существующего кода.

На рис. 2 представлена схема базы данных для управления информационным обеспечением задач кафедры.

При разработке использована микросервисная архитектура. Для развертывания приложения использована практика непрерывной доставки и контейнеры.

Непрерывная доставка – это практика разработки программного обеспечения, когда при любых изменениях в программном коде выполняется автоматическая сборка, тестирование и подготовка к окончательному выпуску. Непрерывная доставка является одним из основополагающих принципов разработки современных приложений, поскольку расширяет практику непрерывной интеграции за счет того, что все изменения кода после стадии сборки развертываются в тестовой и/или в рабочей среде.

Контейнеры – хорошая альтернатива аппаратной виртуализации. Они позволяют запускать приложения в изолированном окружении, но при этом потребляют намного меньше ресурсов.

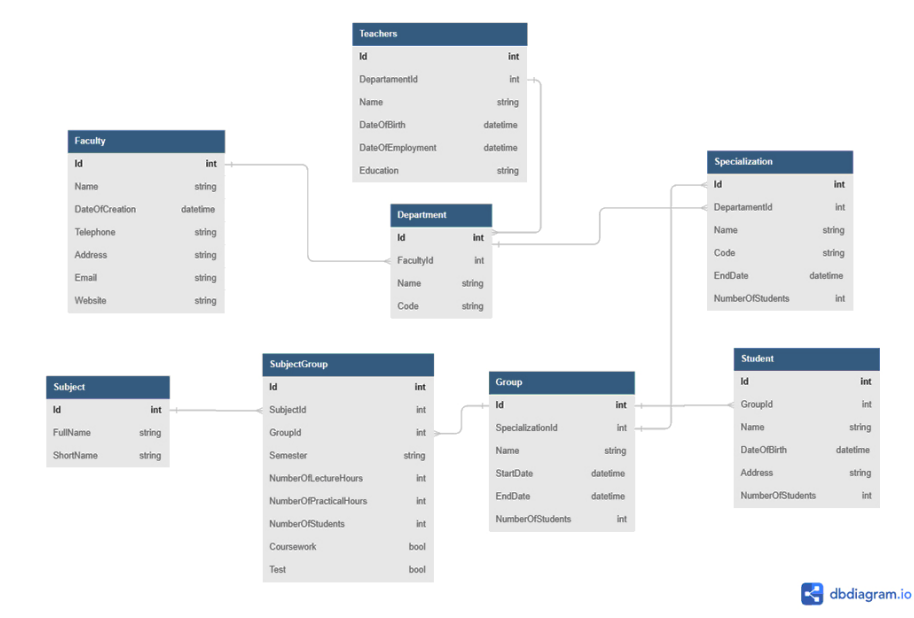


Рис. 2. Схема базы данных для управления информационным обеспечением задач кафедры

Таким образом, был разработан web-сервис, который получает данные из 1С базы данных университета и предоставляет данные в json. Данное приложение может быть использовано в учреждениях образования для автоматизации процессов обучения. Приложение позволит существенно упростить учебный процесс.

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ АВТОМАТИЗАЦИИ ВЕДЕНИЯ ИНДИВИДУАЛЬНОГО ПЛАНА ПРЕПОДАВАТЕЛЕЙ

Р. О. Езепенко

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

Научный руководитель Т. Л. Романькова

Описаны особенности разработки программного комплекса для автоматизации ведения индивидуального плана преподавателей с разнovidным пользовательским интерфейсом; выявление функциональных и нефункциональных требований к приложению; обнаружение и дальнейшее объяснение особенности разработки компонентов пользовательского интерфейса, применяя Back-end for front-end подход.

Ключевые слова: автоматизация, микросервисы, BFF, .NET, Android application.

Процессы автоматизации в современном мире находят применение в различных сферах жизни исфера образования – не является исключением. В рамках данной работы приведен пример автоматизации ведения индивидуального плана преподавателя, с решением типовых задач, характерных для справочных систем.