

Рис. 2. Архитектура приложения

В заключение можно отметить тот факт, что разработанное приложение полностью выполняет поставленные как функциональные, так и нефункциональные требования. Обнаружена потенциальная проблема разнovidного пользовательского интерфейса, с различием в предоставляемом функционале, решением которой является архитектура системы, учитывающая потенциальные проблемы системы.

## WEB-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ РАСПРЕДЕЛЕНИЯ НАГРУЗКИ ПО ПРЕПОДАВАТЕЛЯМ

Н. С. Заяц

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

Научный руководитель Т. Л. Романькова

*Описаны компоненты, необходимые для создания программного комплекса по автоматизации распределения нагрузки: архитектура программного комплекса, схема базы данных, технологии, а также способы организации их взаимодействия.*

**Ключевые слова:** автоматизация, документная база данных, контейнеризация, C#, Web API, REST, React.

В наши дни в независимости от занимаемой должности и сферы деятельности люди все чаще вынуждены документировать свою деятельность, составлять подробные планы. Это вызывается все большим усложнением рабочего процесса. Однако человеческие возможности ограничены, и дополнительная бумажная работа приводит к переработкам и стрессу.

Данная проблема встречается и в системе образования. Преподаватели вынуждены создавать, потом вручную перепроверять и, в случае наличия ошибки, переделывать большое количество документов. Конечно, подобные документы очень важны для организации работы в столь сложной организации, как университет. Из этого возникает объективная потребность в автоматизированной системе помощи преподавателям в составлении документации. Одним из таких документов является учебная нагрузка преподавателя.

Таким образом, главная цель работы – автоматизация распределения нагрузки по преподавателям.

Разрабатываемый программный продукт должен обеспечивать автоматизацию следующих бизнес-процессов:

- рассылка уведомлений о необходимости составления плана, о скором окончании сроков сдачи;
- автоматическая генерация excel-отчетов на основе обрабатываемых данных;
- автоматическая валидация распределения данных;
- возможность удаленной работы с данными;
- авторизация и аутентификация пользователей как по ролям, так и по кафедрам.

Вопрос доступности разрабатываемого приложения стоит очень остро. Большое количество преподавателей имеют средний или пожилой возраст, что усложняет работу с компьютером. Наличие разнообразных операционных систем с различным процессом установки программного обеспечения может создать дополнительные трудности, особенно при возникновении необходимости установки обновлений. Поэтому оптимальным решением в данной ситуации является создание web-приложения с выделенным сервером для обработки данных.

Так как задача автоматизации документооборота в рамках университета является комплексной, для ее решения разрабатывается несколько программных комплексов, которые будут взаимодействовать между собой. Уже существует база данных, которая хранит информацию, связанную с предметной областью университета. Поэтому приложение, разрабатываемое в данной работе, имеет необходимость в хранении только отчетов и тех данных, с которыми непосредственно сейчас ведется работа. Попытка нормализации подобных данных приведет к большому времени ожидания получения результатов, а использование подхода с двумя базами данных: реляционной и нереляционной не оправдано, так как каждый преподаватель будет работать только со своими данными и использование системы транзакций не имеет смысла. Поэтому оптимальным выбором в данной ситуации является использование нереляционной документной базы данных Mongo Db.

Структура представления отчета в базе данных приведена на рис. 1.

```
ObjectID id [primary key]
Date created_at,
ObjectID teacher_id,
Json object first_term {
  [
    String name_of_the_discipline,
    String faculty,
    String specialty,
    Int course,
    Int number_of_groups,
    Int number_of_students,
    Int number_of_lectures,
    Int number_of_laboratory_classes,
    Int number_of_exams,
    Int number_of_term_papers,
    Timestamp time_of_graduation_design
  ]
},
json object second_term [
  [
    String name_of_the_discipline,
    String faculty,
    String specialty,
    Int course,
    Int number_of_groups,
    Int number_of_students,
    Int number_of_lectures,
    Int number_of_laboratory_classes,
    Int number_of_exams,
    Int number_of_term_papers,
    Timestamp time_of_graduation_design
  ]
]
```

Рис. 1. Структура представления отчета в базе данных

Логическая модель данных предметной области, которые используются для формирования отчета, представлена на рис. 2.

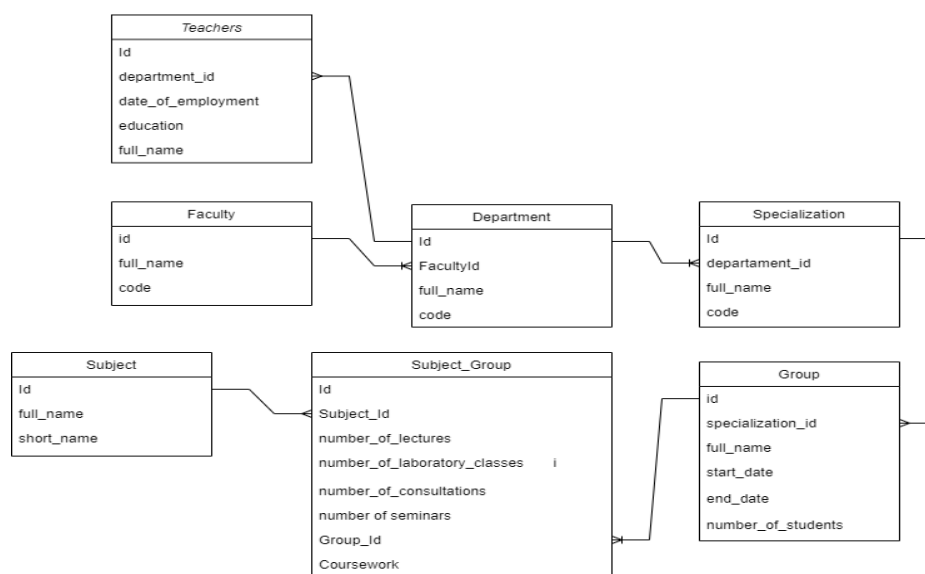


Рис. 2. Логическая модель данных предметной области

В качестве сервера обработки данных используется ASP.NET Core Web API приложение, спроектированное на основе REST принципов. Данный подход позволяет создать понятный интерфейс, который будет легко расширить, а за счет использования платформы .Net возможно предоставлять интерфейс посредством Nuget пакетов.

Так как в данном web-приложении многие элементы, такие как заголовки таблицы и элементы меню, будут обновляться редко, в отличие от рабочей области с данными, из которых создается отчет, то для увеличения производительности следует использовать одностраничное приложение. Данный подход позволит снизить не только интернет трафик пользователя и время ожидания, но еще существенно уменьшит нагрузку на сервер.

Для построения подобного приложения был выбран язык программирования JavaScript и фреймворк React, так как они имеют широкое распространение и качественную документацию, что ведет к сокращению времени на разработку.

Для отправки уведомлений пользователю используется Google SMTP клиент. Он обладает всем необходимым функционалом, а количество бесплатных обращений покрывает потребности университета.

Система авторизации и аутентификации основывается на использовании JWT-токенов, так как это позволяет использовать единую систему авторизации между всеми приложениями университета.

Так как предполагается необходимость дальнейшей поддержки разрабатываемого приложения, то для упрощения процесса сборки, тестирования и развертывания, необходимо использование процессов непрерывной интеграции и развертывания. Для данных целей были выбраны такие инструменты, как GitLab Runner и Docker Compose.

GitLab Runner интегрирован в систему управления версиями GitLab и обладает широкой функциональностью, а Docker Compose позволит запустить собранный программный комплекс на компьютере-сервере автоматически.

Схема программного комплекса изображена на рис. 3.

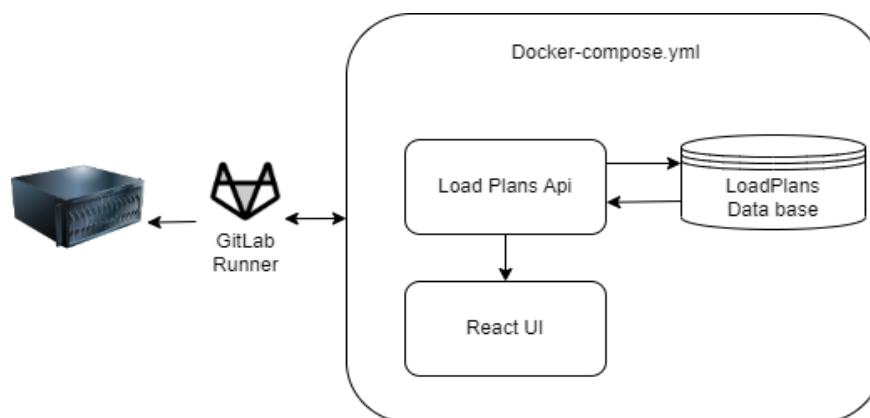


Рис. 3. Схема программного комплекса

Таким образом, в рамках данной работы создается приложение, которое обеспечивает авторизацию и аутентификацию пользователей, возможность удаленного создания нагрузочных планов на преподавателей с валидацией в реальном времени и последующий экспорт отчетов в формате Excel, отправку уведомлений и напоминаний о необходимости формирования нагрузки преподавателей и скором окончании срока сдачи.

Главными преимуществами данного приложения являются невысокая цена разработки первоначальной версии продукта, простота дальнейшей эксплуатации и доработки.

## АНАЛИЗ АУДИОСИГНАЛА С ПРИМЕНЕНИЕМ АЛГОРИТМА БЫСТРОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ

М. А. Макеев

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

Научный руководитель В. Н. Шибeko

*Рассмотрены такие важнейшие процессы обработки и анализа аудиосигнала, как эквалализация, компрессия, денойзинг и исследование аудиосигнала. Также была реализована программа-анализатор входного аудиосигнала с использованием алгоритма быстрого преобразования Фурье. С помощью данной программы на непосредственном примере продемонстрирована важность такого инструмента в различных сферах жизни человека.*

**Ключевые слова:** процесс денойзинга, компрессия аудио-сигнала, быстрое преобразование Фурье, музыка, характеристика звучания, скорость алгоритмов.

Бой часов, пение птиц, речь человека и музыка – все эти явления можно обобщить одним словом «звук». Физика трактует звук как порождаемые некоторым источником колебания воздуха, на которые реагируют наши слуховые органы чувств.