

Интеграция с внешними системами позволяет веб-приложению взаимодействовать с другими корпоративными учетными системами. Также предусмотрена возможность импорта и экспорта данных в форматах *Excel*, *CSV* и *PDF*, что расширяет возможности работы с отчетами и дает возможность адаптировать информацию под нужды разных подразделений.

Таким образом, разработка информационно-аналитического веб-приложения для управления автотранспортом РУП «Гомельэнерго» является важным шагом в автоматизации процессов учета и анализа эксплуатации автопарка. Применение современных технологий позволяет не только повысить прозрачность процессов, но и существенно сократить затраты на эксплуатацию транспортных средств, особенно за счет более точного контроля расхода топлива и своевременного проведения технического обслуживания.

Кроме того, система способствует улучшению качества принятия решений благодаря удобной аналитике и отчетности, что положительно влияет на общую эффективность работы предприятия. Внедрение веб-приложения помогает минимизировать риски, связанные с неисправностями автотранспорта, и оптимизировать процессы управления ресурсами. В дальнейшем, с развитием системы, можно будет интегрировать дополнительные модули для мониторинга состояния автомобилей с помощью датчиков и других современных технологий.

В целом, использование информационно-аналитического веб-приложения позволит РУП «Гомельэнерго» повысить эффективность работы автопарка, снизить эксплуатационные расходы и увеличить степень автоматизации процессов, что, в свою очередь, способствует устойчивости работы и конкурентоспособности компании на рынке энергетических услуг.

Л и т е р а т у р а

1. Бучек, Г. ASP.NET / Г. Бучек. – СПб. : Питер, 2002. – 509 с.
2. Лок, Э. ASP.NET Core в действии / Э. Локк ; пер. с англ. Д. А. Беликова. – М. : ДМК-Пресс, 2021. – 906 с.
3. Смит, Дж. П. Entity Framework Core в действии / Дж. П. Смит ; пер. с англ. Д. А. Беликова. – М. : ДМК-Пресс, 2022. – 690 с.
4. Тузовский, А. Ф. Проектирование и разработка web-приложений : учеб. пособие для СПО / А. Ф. Тузовский. – М. : Юрайт, 2019. – 218 с.
5. Гоф, Дж. Проектирование архитектуры API / Дж. Гоф, Д. Брайант, М. Оберн. – Астана : АЛИСТ, 2024. – 288 с.
6. Павловская, Т. А. C#. Программирование на языке высокого уровня / Т. А. Павловская. – СПб. [и др.] : Питер, 2014. – 432 с.

ПРОГРАММНЫЙ КОМПЛЕКС С МОБИЛЬНЫМ ПРИЛОЖЕНИЕМ ДЛЯ УЧЕТА ПОСЕЩАЕМОСТИ СТУДЕНТОВ ГГТУ им. П. О. СУХОГО

Д. С. Волчек

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

Научный руководитель О. Д. Асенчик

Разработан программный комплекс, автоматизирующий учет посещаемости студентов, который включает графический интерфейс, адаптированный для работы как на стационарных компьютерах, так и на мобильных платформах, что обеспечивает опера-

тивное и достоверное фиксирование данных. Комплекс обладает возможностью расширения и адаптации для применения в различных образовательных учреждениях.

Ключевые слова: мобильные приложения, API, ASP .NET, учет посещаемости.

Ручной учет посещаемости студентов является трудоемким и подверженным ошибкам процессом. Оперативный доступ к информации для принятия управленческих решений при таком способе учета также затруднен. Автоматизация этого процесса с помощью программного комплекса позволит упростить работу преподавателей и администрации, снизить вероятность ошибок и сэкономить время. Система учета посещаемости мотивирует студентов более ответственно относиться к посещению занятий, так как их присутствие или отсутствие будет фиксироваться объективно и оперативно. Это способствует повышению успеваемости и дисциплины.

В настоящее время учет посещаемости студентов в большинстве образовательных учреждений, включая университеты, осуществляется с использованием различных подходов и технологий. Однако многие из этих решений имеют ограничения и не полностью отвечают современным требованиям. На рынке присутствуют мобильные приложения для учета посещаемости, такие как Attendance Radar [1], My Attendance Tracker [2] и другие, но они часто имеют ограниченную функциональность, слабую интеграцию с другими системами университета и не всегда адаптированы под специфику конкретного учебного заведения. Программный комплекс разрабатывался с учетом специфики ГГТУ им. П. О. Сухого и интегрирован с текущими системами управления учебным процессом. Это обеспечит централизованное управление данными и упростит работу преподавателей и администрации.

Представляемое приложение предоставляет удобный интерфейс для студентов и преподавателей, включая возможность отслеживания посещаемости в режиме реального времени, уведомления о пропущенных занятиях и интеграцию с расписанием. Приложение разработано с учетом потребностей студентов и преподавателей, обеспечивая простоту использования, минимальное время обучения и высокую скорость работы.

Предлагаемый программный комплекс с мобильным приложением предлагает инновационный подход, сочетающий в себе удобство, функциональность и интеграцию с существующими системами. Это делает его актуальным и востребованным решением для современного университета.

Цель проекта – создание кроссплатформенного решения для учета посещаемости.

На рис. 1 приведена схема взаимодействия клиента с сервером. Исходя из схемы видно, как пользователь взаимодействует с сервером. При вводе своего логина и пароля пользователь входит в приложение, проходя через процесс аутентификации. В зависимости от его роли он видит разные страницы и ему доступны разные действия. В отправляемой системе токене содержатся имя и роль пользователя. И при отправке запроса на выполнение какой-либо операции произойдет проверка роли в токене пользователя и в зависимости от роли пользователя ему будет выдан тот или иной ответ от сервера. В приложении присутствует несколько ролей, а именно: декан, заместитель декана, преподаватель и староста. Декан может добавлять и изменять любую информацию, в том числе он имеет доступ к любой странице и изменению ролей других пользователей. Заместитель декана выполняет те же функции, но не может изменять роли пользователей. Преподаватель может редактировать посещения на своих занятиях, а староста – только у своей группы, иные функции им недоступны.

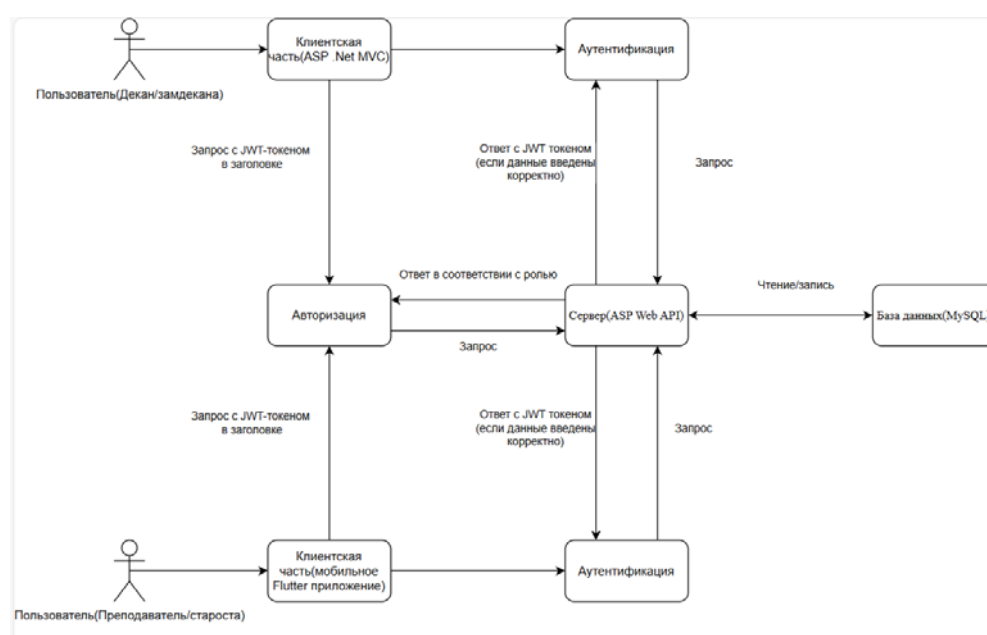


Рис. 1. Схема взаимодействия между клиентом сервером

Программная архитектура основана на многослойной архитектуре и разделен на несколько слоев: слой данных, слой сервера, слой клиента. Такая архитектура помогает разделить ответственность между различными частями приложения, что делает код более поддерживаемым и масштабируемым. Она также облегчает тестирование и внесение изменений в отдельные компоненты без влияния на другие части системы.

Для хранения и обработки данных о структурах баз данных использовалась реляционная база данных MySQL для надежного и эффективного хранения всех необходимых данных о сущностях. Для работы с базой данных используется ORM (Object-Relational Mapping) фреймворк Entity-Framework (EF) для работы с базами данных в C# и .NET. Он упрощает взаимодействие с базой данных.

API (Application Programming Interface) разработан при помощи ASP.NET Web API [3]. Он является частью экосистемы ASP.NET и позволяет разработчикам создавать RESTful (REST) веб-сервисы, которые возвращают данные в форматах, таких как JSON или XML. Create, Read, Update и Delete (CRUD) операции выполняются с помощью паттерна «Репозиторий», что позволяет отделить логику работы с данными от бизнес-логики и делает приложение более гибким. API можно использовать для создания любых приложений и на любых платформах.

Графический интерфейс десктопного Web-приложения разработан на ASP.NET MVC – это фреймворк для создания веб-приложений с использованием шаблона MVC. Он предназначен для генерации HTML-страниц, которые отображаются в браузере. Используется разработанная API для выполнения CRUD операций.

Для разработки мобильного клиента был использован фреймворк Flutter [4]. Это фреймворк с открытым исходным кодом для создания кроссплатформенных приложений, разработанный компанией Google. Flutter использует Dart – современный язык программирования, также разработанный Google. Код Dart компилируется в нативный код, что обеспечивает высокую производительность. Мобильное приложение также будет работать с API и иметь свою систему авторизации, завязанную на API.

Важным компонентом разработанного Flutter приложения является библиотека BLoC (Business Logic Component) [5]. Она помогает разделять бизнес-логику от пользовательского интерфейса, что делает код более читаемым, тестируемым и поддерживаемым. BLoC основан на реактивном программировании и использует потоки (Streams) для передачи данных между слоями приложения. BLoC – это инструмент для управления состоянием в Flutter-приложениях. Он помогает создавать чистый, модульный и тестируемый код, что особенно важно для крупных проектов.

Разработанный программный комплекс [6] может быть интегрирован с существующими системами управления учебным процессом, что позволит централизованно управлять данными и анализировать успеваемость студентов. Мобильное приложение обеспечивает удобный доступ к информации о посещаемости для студентов и преподавателей, а также используется непосредственно для учета. Студенты смогут отслеживать свою посещаемость в режиме реального времени, а преподаватели и руководство факультета – оперативно вносить данные и получать отчеты.

Л и т е р а т у р а

1. Attendance Radar. – URL: <https://attendanceradar.com/> (дата обращения: 18.02.2025).
2. Moodle. – URL: <https://ru.wikipedia.org/wiki/Moodle?> (дата обращения: 18.02.2025).
3. Создание веб-API с помощью ASP.NET Core. – URL: <https://learn.microsoft.com/ru-ru/aspnet/core/web-api/?view=aspnetcore-9.0> (дата обращения: 18.02.2025).
4. Find your way with Flutter. – URL: <https://docs.flutter.dev/get-started/fundamentals> (дата обращения: 19.02.2025).
5. Flutter BLoC для начинающих. – URL: <https://medium.com/flutter-community/flutter-bloc-for-beginners-839e22adb9f5> (дата обращения: 19.02.2025).
6. Attendance-app: GitHub. – URL: <https://github.com/DenisVolchekkk/Attendance-app/tree/main/diplom> (дата обращения: 19.02.2025).

ПРОЕКТИРОВАНИЕ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ ФОРМИРОВАНИЯ ПЛАНОГРАММ НА ОСНОВЕ АНАЛИЗА ТЕКСТОВЫХ ДОКУМЕНТОВ

Ю. Д. Евженко

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

Научный руководитель К. С. Курочка

Предложены концепция и архитектура экспертной системы, способной анализировать текстовые данные для извлечения критериев размещения товаров. Разработка направлена на повышение точности и скорости формирования планограмм, снижение трудозатрат и минимизацию ошибок, связанных с ручной интерпретацией текстовых инструкций.

Ключевые слова: экспертная система, планограмма, анализ текста, большие языковые модели, обработка естественного языка, автоматизация мерчандайзинга.

Планограмма – это визуальная схема выкладки товаров на полках торгового оборудования, являющаяся ключевым инструментом мерчандайзинга. От корректности и эффективности планограммы напрямую зависят показатели продаж, восприятие бренда и удовлетворенность покупателей. Традиционно процесс формирования планограмм осуществляется вручную специалистами-мерчандайзерами. Однако этот процесс становится все более сложным из-за обилия требований, поступающих из различных источников: договоров с поставщиками, корпоративных стандартов,