

3. Traffic Sign and Vehicle Detection Based on Improved YOLOv8 for Autonomous Driving / Zhongjie Huang, Lintao Li1, Gerd Christian Krizek, Linhao Sun // 2023 IEEE 6th International Conference on Industrial Cyber-Physical Systems (ICPS). – 2023. – P. 226–232.
4. Брехт, Э. А. Применение нейронной сети YOLO для распознавания дефектов / Э. А. Брехт, В. Н. Коншина // Интеллектуальные технологии на транспорте. – 2022. – № 2. – С. 41–47.
5. Kurachka, K. S. Localization of human percentages on X-ray images with use of Darknet YOLO / K. S. Kurachka, T. V. Luchshava, K. A. Panarin // Doklady BGUIR. – 2018. – Vol. 113, N 3. – P. 32–38.

АРМ НАЧАЛЬНИКА УЧАСТКА МОНТАЖА И НАЛАДКИ СИСТЕМ ОХРАННОЙ И ПОЖАРНОЙ СИГНАЛИЗАЦИИ БООО «СИСТЕМАВТОМАТИКА» С АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИЕЙ ВЫХОДНОЙ ДОКУМЕНТАЦИИ

А. Г. Александров

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

Научный руководитель М. И. Михайлов

Представлена разработка приложения, демонстрирующего автоматизированное рабочее место главного инженера по монтажу и наладке систем охранной и пожарной сигнализации для организации БООО «Системаавтоматика» с возможностью автоматической генерации документации.

Ключевые слова: web-приложение, автоматизация, учет, современные технологии, ремонт оборудования.

В современных условиях ручное заполнение документации при монтаже и наладке охранно-пожарных систем становится серьезной проблемой – оно отнимает много времени, повышает риск ошибок и усложняет контроль за выполнением работ. Разрозненное хранение актов, протоколов и отчетов приводит к путанице, дублированию данных и трудностям при поиске нужной информации. Централизованная система автоматической генерации и хранения документов не только ускоряет процессы, но и обеспечивает порядок в учете, исключает потерю данных и упрощает взаимодействие между сотрудниками.

Программный комплекс создается с применением современных технологий, обеспечивающих высокую производительность, масштабируемость и удобство дальнейшей поддержки.

Серверная часть автоматизированного рабочего места разработана на платформе *Node.js* с использованием фреймворка *NestJS*, что обеспечивает высокую производительность и масштабируемость системы [1, с. 22]. Архитектура приложения построена по модульному принципу, где каждый функциональный блок (работа с документами, управление пользователями, отчетность) выделен в отдельный модуль с четко определенными зависимостями. Это позволяет легко расширять функционал системы и поддерживать код в актуальном состоянии.

Для работы с данными используется *TypeORM* – современный *ORM*-фреймворк, который позволяет описывать сущности базы данных в виде *TypeScript*-классов с применением декораторов. *TypeORM* поддерживает автоматическую генерацию миграций, что особенно важно при работе в команде и поэтапном развитии системы. Фреймворк предоставляет *QueryBuilder* и репозиторийный паттерн работы с данными, что делает код более читаемым и поддерживаемым.

В качестве базы данных выбрана *SQLite* – легковесная встраиваемая СУБД. Ее главное преимущество заключается в простоте развертывания – база данных хранится в виде единого файла, не требует настройки сервера и при этом полностью соответствует принципам *ACID*. *SQLite* идеально подходит для данного проекта, так как система должна работать стабильно даже в условиях ограниченного доступа к сети.

Серверная часть реализует *RESTful API*, обеспечивающее взаимодействие между клиентским приложением и базой данных.

Клиентская часть системы построена на *React* с использованием *Redux* для управления состоянием и *Styled Components* для стилизации компонентов [2, с. 35]. Такой подход обеспечивает стабильную работу сложного интерфейса с множеством динамических элементов и форм. *Redux* выступает в качестве централизованного хранилища данных, где содержатся все сведения о документах, пользователях и настройках системы. Для обработки асинхронных операций применяется *redux-thunk*, что позволяет эффективно работать с *API* и выполнять сложные цепочки действий. Оптимизированные селекторы помогают избегать лишних перерисовок компонентов, поддерживая высокую производительность приложения даже при работе с большими объемами данных.

Стилизация реализована через *Styled Components*, что дает возможность создавать компоненты с инкапсулированными стилями и динамически изменять их параметры в зависимости от текущего состояния или полученных *props*. Этот подход обеспечивает модульность и переиспользуемость *UI*-элементов, а также позволяет легко настраивать внешний вид системы через единую тему. Все стили написаны с использованием современного *CSS-в-JS* синтаксиса, что исключает конфликты классов и упрощает поддержку кодовой базы. Особое внимание уделено адаптивности интерфейса, который корректно отображается на устройствах с разными разрешениями экрана.

Пользовательский интерфейс включает комплексные формы ввода данных с валидацией, интерактивные таблицы с сортировкой и пагинацией, а также систему уведомлений о выполнении операций. Для работы с *API* применяется *axios* с настроенными перехватчиками запросов, которые автоматически добавляют токены авторизации и обрабатывают возможные ошибки. Реализована *JWT*-аутентификация с автоматическим обновлением токенов, что обеспечивает безопасность работы с системой. Все действия пользователя фиксируются в хранилище *Redux*, что значительно упрощает процесс отладки и анализа поведения приложения. Логика ролевого доступа динамически изменяет интерфейс в зависимости от прав текущего пользователя, скрывая или показывая соответствующие элементы управления.

Сервис создания документов представляет собой ключевой функциональный модуль системы, автоматизирующий процесс генерации выходных форм. В основе работы сервиса лежит механизм шаблонной обработки, где заранее подготовленные образцы документов содержат специальные метки-заменители (плейсхолдеры), которые в реальном времени заполняются актуальными данными из базы данных.

При инициализации процесса создания документа система загружает соответствующий шаблон из хранилища, одновременно выполняя запросы к базе данных для получения необходимой информации. Данные могут включать сведения об объекте, параметрах оборудования, исполнителях работ и других сущностях, связанных с конкретным монтажным проектом. Шаблоны документов поддерживают сложную структуру с условными блоками – некоторые разделы могут автоматически включаться или исключаться из итогового документа в зависимости от полученных данных.

Разрабатываемое приложение является эффективным решением для автоматизации рабочих процессов предприятия. Оно сочетает в себе удобство использования, гибкость и масштабируемость, что делает его идеальным инструментом для предприятий, занимающихся монтажом и обслуживанием систем охранной и пожарной сигнализации.

Таким образом, благодаря автоматической генерации документов и удаленному доступу, приложение значительно сокращает временные затраты на рутинные задачи и повышает производительность сотрудников. Использование современных технологий обеспечивает высокую надежность и возможность дальнейшего расширения функционала, что делает приложение не только актуальным, но и перспективным решением для бизнеса.

Литература

1. Янг, А. Node.js в действии / А. Янг, Б. Мек, М. Кантелон. – СПб. : Питер, 2018. – 432 с.
2. Стефанов, С. React.js. Быстрый старт / С. Стефанов. – СПб. : Питер, 2017. – 304 с.

ВИЗУАЛИЗАЦИЯ ДВИЖЕНИЯ ЭЛЕМЕНТОВ ГИДРАВЛИЧЕСКОГО КЛАПАНА С ПРИМЕНЕНИЕМ ПРОГРАММЫ «Blender»

А. Д. Левкина

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

Научный руководитель Е. П. Поздняков

Рассмотрена визуализация движения диафрагмы гидравлического клапана с применением программы «Blender». Отмечено, что сложное движение зафиксированного в сборке элемента воспроизводится при помощи внедрения виртуального скелета, состоящего из отдельных частей арматуры, в трехмерную модель.

Ключевые слова: анимация, арматура, виртуальный скелет, диафрагма, сложное движение.

Целью работы является создание сложного движения диафрагмы гидравлического клапана при помощи внедрения виртуального скелета в программе «Blender». Для создания мультипликации, рекламы, фильмов и другого анимационного цифрового контента применяются специализированные программные продукты. Системы автоматизированного проектирования (САПР) также обладают средствами для визуализации движения механизмов, приборов, устройств, узлов машин.

Для создания анимации работы механизма (в частности, демонстрация деформации диафрагмы при работе клапана) был использован программный продукт «Blender» в связи с тем, что в САПР SolidWorks возможно задать анализ на прочность с конечной демонстрацией деформации детали, но модуль анимации в ПО недостаточно развит и заточен на основные операции демонстрации деталей и сборок, такие как: сборка-разборка механизма с передвижением моделей деталей сборочного механизма по прямой траектории, вращение модели или сборки для демонстрации общего вида модели [2].

«Blender» – профессиональное свободное и открытое программное обеспечение для создания трехмерной компьютерной графики, включающее в себя средства моделирования, скульптинга, анимации, симуляции, рендеринга, постобработки и мон-