

```

int minRange = 90; //минимальное расстояние для реагирования
int rangeToSlowDown = 320; //дистанция, на которой стоит начать замедляться
int minVelocity = 25; //минимальная скорость
int maxVelocity = 55; //максимальная скорость
int curVelocity = 25; //текущая скорость

Drivetrain.setDriveVelocity(minVelocity, percent);

```

Рисунок 4 – Объявление некоторых переменных и установка начальной скорости робота

Действующий по разработанной программе робот способен объехать практически любое препятствие, не выходящее за рамки ограничений его комплектующих.

УДК 004.4

ИНФОРМАЦИОННАЯ СИСТЕМА УПРАВЛЕНИЯ УЧЕБНЫМИ ПЛАНАМИ

**Казakov В.Е., к.т.н., доц., Ринейский К.Н., начальник ЦИТ, Демидов Д.Д., выпускник,
Карнилов М.С., асс.**

*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Реферат. В статье представлен обзор разработки клиентской части информационной системы управления учебными планами, представлены предпосылки его внедрения в учреждение образования и круг решаемых им задач.

Ключевые слова: front-end, учебный план, JavaScript, React.

Учебный план – нормативный документ, регламентирующий общее направление и основное содержание подготовки специалиста, последовательность и интенсивность, сроки изучения учебных дисциплин, основные формы организации обучения, формы и сроки проверки знаний и умений учащихся. В УО «ВГТУ» разработан REST-сервис, обеспечивающий хранение данных учебных планов [1].

Разработана клиентская часть данной информационной системы, которая представляет собой браузерное приложение, не требующее инсталляции. Приложение разработано на языке TypeScript на основе фреймворка React.js [2]. Применялись также: библиотека стилизации компонентов emotion и фреймворк material UI; средство организации централизованного хранилища приложения redux; в качестве средства валидации форм использовалась связка библиотек yup и formic; библиотека доступа к REST-сервисам axios, библиотека управления уведомлениями hot-toast, асинхронность обработки запросов реализована с помощью библиотеки thunk, а также средство автоматизации создания селекторов reselect.

Особое внимание было уделено выбору библиотеки реализации табличного визуального компонента. Создание таблиц является одной из наиболее распространенных задач веб-разработки. Вместе с тем, создание качественной таблицы, которая будет удовлетворять всем требованиям пользователей, может быть сложной задачей.

Были исследованы три популярные библиотеки для создания таблиц на базе React: AG Grid, DataGrid и Material React Table. Все они предоставляют множество функций для работы с данными, таких как сортировка, фильтрация, группировка и агрегация. Они также поддерживают виртуализацию и могут работать с большими объемами данных. AG Grid и DataGrid имеют платные версии, которые предоставляют дополнительные функции, такие как экспорт данных, поддержка графиков и диаграмм, а также поддержка различных форматов данных. В то время как Material React Table имеет только бесплатную версию, которая предоставляет основные функции. Исходя из требований к табличному компоненту, удобства использования и скорости освоения была выбрана библиотека AG Grid [3].

AG Grid предоставляет множество возможностей для настройки таблицы и ее поведения;

поддерживает различные типы данных, такие как числа, строки, даты, изображения и т.д.; позволяет настраивать внешний вид таблицы, включая шрифты, цвета и стили; имеет множество функций для работы с данными, таких как сортировка, фильтрация, группировка и агрегация; поддерживает функции поиска и выделения строк. AG Grid может работать с большими объемами данных, благодаря своей оптимизированной архитектуре и встроенной поддержке виртуализации.

При проектировании интерфейса учитывались требования сотрудников учебного отдела. Разработанная форма соответствует внешнему виду документа учебного плана (рис. 1), что облегчает обучение работе с приложением.

№ п/п	Наименование учебного дисциплины (курса, проекта/практической работы)	Семестр	Часы	Курсовые проекты/практические работы	Итого	Курсовые проекты/практические работы	Итого	Итого	Итого	1 семестр (17 недель)		2 семестр (15 недель)	
										1	2	1	2
1	Теоретический курс	1	10	8	2790	1429	541	370	284	232			
	Социально-гуманитарные дисциплины I												
111	История Беларуси государственности	1	0	0	80	89	33	22	12	22			
112	Философия	1	1	0	220	80	84	0	0	84			

Рисунок 1 – Вид страницы редактирования учебного плана

Приложение реализует следующие основные функции: управление специальностями; управление учебными планами, включая редактирование модулей, дисциплин, семестров обучения, учебных практик; автоматический подсчёт основных показателей учебных планов: количество звонковой нагрузки, зачётных единиц, курсовых и иных работ по семестрам (рис. 2).

2	Количество часов учебных занятий	0	0	0	0	0	0	0	0	0	0	0	
3	Фактическое количество зачетных единиц	0	0	0	0	0	0	0	0	0	0	0	
4	Число курсовых работ/проектов	0	0	0	0	0	0	0	0	0	0	0	
	Количество часов учебных занятий	12	30	8	2790	1429	543	370	284	232	953	518	23
	Количество часов в неделю											33	31
	Количество курсовых проектов											0	0
	Количество курсовых работ											0	1
	Количество экзаменов											4	3
	Количество зачетов											3	4

Рисунок 2 – Вид страницы редактирования учебного плана

Форма учебного плана содержит набор ячеек, требующих возможности редактирования, а также набор ячеек, содержащих значения, вычисляемые автоматически, поэтому при проектировании формы было принято решение разместить в редактируемых ячейках элементы управления, включающие режим редактирования с последующим подтверждением или отменой внесённых изменений.

Подтверждённые изменения сразу же передаются в сервис для сохранения в базе данных, что повышает оперативность обновления информации, но, с другой стороны, несколько увеличивает сетевой трафик. Такое увеличение не будет критическим, поскольку, при исследовании бизнес-процесса управления планами было выяснено, что количество одновременно работающих с

системой пользователей не будет превышать пяти. Также снижает объём передаваемой по сети информации предложенный формат обмена: пакет включает только изменённые данные.

При изменении значений часов и форм контроля в семестрах дисциплин, а также добавления и удаления дисциплин необходим пересчёт всех итоговых показателей учебного плана, что влечёт за собой увеличение затрат времени на вычисления. Задача была решена с использованием селекторов. Селектор – это функция, которая принимает состояние Redux в качестве аргумента и возвращает данные, полученные из этого состояния для обновления визуального компонента. Селектор не пересчитывается, пока не изменится состояние части хранилища, за которым он следит. Таким образом, при изменении данных в общем хранилище приложения, селектор не допустит перерисовки связанного с ним визуального компонента, если эти изменения не затронули отображаемые с компоненте данные.

Использование сотрудниками учебного отдела данного модуля информационной системы университета позволит сформировать банк данных для обеспечения информацией другие части системы, такие как модуль распределения нагрузки по кафедрам, управления нагрузкой заведующим кафедрой, распределение аудиторного фонда.

Список использованных источников

1. Казаков, В. Е. Разработка back-end приложения «Учебно-методический отдел (учебные планы)» / М. С. Карнилов, В. Е. Казаков // Материалы докладов 55-й Международной научно-технической конференции преподавателей и студентов: в 2 т. / УО «ВГТУ». – Витебск, 2022. – Т. 2. – с. 5–7
2. Сайт «react.dev» [Электронный ресурс] <https://react.dev/learn> – Дата доступа: 12.05.2023.
3. Сайт «ag-grid.com» [Электронный ресурс] / Spring Framework – Режим доступа: <https://www.ag-grid.com/> – Дата доступа: 12.05.2023.

УДК 004.4

ЭКСПОРТ ДАННЫХ ИЗ FRONT-END ПРИЛОЖЕНИЯ В XLSX-ФОРМАТ

*Поздняков К.В., студ., Жизневский В.А., к.ф.-м.н., доц., Казаков В.Е., к.т.н., доц.
Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Реферат. В статье представлен обзор программных средств, предназначенных для извлечения данных из front-end приложения в формат *xlsx*.

Ключевые слова: импорт, XLSX, JavaScript, библиотека.

Существует несколько способов экспортировать данные из интерфейсного приложения в формат файла Excel (XLSX).

Вот несколько методов:

1. Использование библиотеки JavaScript. Существует множество доступных библиотек JavaScript, таких как *xlsx* и др, которые могут помочь вам экспортировать данные из вашего интерфейсного приложения в файл Excel. Эти библиотеки предоставляют API для преобразования данных JSON в формат XLSX и могут быть интегрированы в ваше приложение.

2. Использование серверного API. Если ваше интерфейсное приложение взаимодействует с серверным API, вы можете отправить запрос в API для экспорта данных в файл Excel. Затем серверный API может сгенерировать файл Excel и отправить его обратно во внешнее приложение.

3. Использование стороннего сервиса. Доступно несколько сторонних сервисов, которые позволяют экспортировать данные в формат Excel. Эти службы обычно предоставляют API, которые могут быть интегрированы в ваше приложение.

Рассмотрим библиотеку *xlsx* – это популярная библиотека JavaScript, используемая для работы с файлами Excel. Библиотека предоставляет ряд функций для работы с файлами Excel, включая:

1. Чтение и запись файлов Excel в различных форматах, таких как XLSX, XLS и CSV.
2. Работа с отдельными листами, строками и ячейками в файлах Excel.
3. Преобразование между различными типами данных и типами данных Excel.