

В. А. Андреев

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Е. А. Ружицкая**, канд. физ.-мат. наук, доцент

РЕАЛИЗАЦИЯ КЛИЕНТСКОЙ ЧАСТИ МОДУЛЯ ДЛЯ МОНИТОРИНГА ЗАКАЗОВ ПРИ РАЗРАБОТКЕ СИСТЕМЫ УПРАВЛЕНИЯ АВТОМОЙКАМИ

Чтобы клиенты автомобильной мойки могли отслеживать готовность своего заказа в зале ожидания, был разработан модуль по мониторингу готовых заказов с использованием фреймворка Angular.

Основными частями модуля является секция, отвечающая за аудио-визуальное оповещение и секция, которая занимается логикой вывода на экран поступающих в очередь заказов.

При разработке первой секции использовался Web Audio API. Он предоставляет разработчикам возможность управлять источниками аудио внутри JavaScript, не используя никаких внешних проигрывателей. В случае описываемого приложения данный API использовался для проигрывания определённого аудио-файла. Для этого создаётся экземпляр класса `Audio`, параметром в конструкторе задаётся имя аудио-файла. Для проигрывания этого аудио необходимо у экземпляра класса вызвать метод `play()`.

В качестве фона на мониторе предусмотрено использование как графического изображения, так и проигрывание видео-потока с помощью YouTube API. Чтобы использовать безопасно такой источник видеоряда внутри `src`-атрибута соответствующего HTML-тега, в контроллере Angular необходимо передавать ссылку на поток видео в функцию `$sce.trustAsResourceUrl`. Настройка этих параметров производится пользователем на странице установок для модуля мониторинга заказов во всплывающем окне.

Вторая секция, отвечающая за логику построения очереди, отвечает за размещение объявления о готовности заказа в порядке очереди. При реализации необходимо учитывать проблему синхронизации с сервером. Для избегания проблемы дублированных данных используется функция сравнения очередей заказов: первая очередь – это поступающая с сервера через определённый интервал времени, вторая очередь – это та, которая уже находится в обработке на клиенте. С помощью функции `reservationQueuesDiff (newQueue, oldQueue)` происходит их сравнение по определённым заданным признакам. Таким

образом, при обновлении данных и наличии различий при сравнении, новый заказ будет добавлен в очередь для объявления.

А. В. Антоненко

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Е. А. Ружицкая**, канд. физ.-мат. наук, доцент

WEB-ПРИЛОЖЕНИЕ «ИГРЫ ММО» С ИСПОЛЬЗОВАНИЕМ СТЕКА MERN

Web-приложение «Игры ММО», целью которого является отображение каталога игр, было разработано с помощью стека MERN.

Стек MERN – это JavaScript-стек, разработанный для упрощения процесса разработки. MERN включает в себя четыре компонента с открытым исходным кодом: MongoDB, Express, React и Node.js. Эти компоненты обеспечивают комплексную среду для работы разработчиков.

MongoDBNoSQL это нереляционная документно-ориентированная база данных. MongoDB не требует схем, в то время как традиционные реляционные базы данных имеют типичный дизайн схемы, основанный на столбцах и таблицах.

Express является базовой платформой web-приложений, структурой web-приложения для Node.js. Чтобы упростить задачу написания кода сервера, используются Express, вместо того, чтобы писать полный код web-сервера вручную на Node.js.

React – это библиотека JavaScript для создания пользовательских интерфейсов. React использует язык программирования JavaScript для создания повторяющихся или условных элементов DOM.

Node.js – это кроссплатформенная среда выполнения JavaScript. Он построен на движке Chrome V8 JavaScript. Node.js предназначен для создания масштабируемых сетевых приложений и может выполнять код JavaScript вне браузера.

Данное приложение было разработано с целью отображение каталога игр. В нем существует разграничение прав доступа к данным: администратор может добавлять, изменять и удалять игры, а обычные пользователи не имеют к этому функционалу доступа.

Пользователи без привилегий могут воспользоваться поиском игры, по таким параметрам как, жанру, стилистике, региону, модели оплаты, типу игры, году выпуска, читать ее описание, смотреть