

# МОДУЛЬ АВТОМАТИЗИРОВАННОГО АДМИНИСТРИРОВАНИЯ ПРОЕКТНЫМИ ЗАДАЧАМИ СОТРУДНИКОВ ПРОЕКТНОЙ КОМПАНИИ

А.А. Реннит

Научный руководитель: А.С. Фадеев, к.т.н.

Томский политехнический университет

E-mail: aar40@tpu.ru

## Введение

В современной экономике существуют организации, работы которых осуществляются на основе проекта. Проектное управление заключается в следующем: выбираются основные характеристики проектируемого оборудования или определяются конечные цели, а уже потом предпринимается попытка достичь их в соответствии с некоторыми временными и денежными ограничениями [1].

Целью работы является разработка информационного модуля для повышения эффективности планирования работ проектной компании.

## Архитектура программной системы

Программное приложение разделяется на две части: клиентская и серверная части. Клиентская часть состоит из компонента «Web client». Данные компоненты связаны с «Application server» с помощью интерфейса HTTP. Согласно диаграмме компонентов, представленной на рисунке 1, «Application server» реализовывает в себе бизнес-логику программного продукта.

Серверный компонент имеет один порт, который связывается с Nginx через HTTP протокол. В зависимости от URL в заголовке HTTP запроса Nginx либо отдает HTML страницу для веб-клиента, либо если URL удовлетворяет API, передает запрос на PHP сервер. PHP сервер взаимодействует с MS SQL сервером, а также с RabbitMQ по протоколу AMQP, также PHP взаимодействует с компонентом Socket.io по протоколу Web-socket.

Общий поток данных выглядит следующим образом: клиентские приложения отправляют запрос на API, PHP обрабатывает запрос, получает данные из базы данных, генерирует ответ и отправляет ответ на запрос. Затем PHP отправляет сообщение в Socket.io об изменениях каких-либо сущностей, данное сообщение приходит всем клиентским приложениям, которые установили соединение с Socket.io. Помимо этого, PHP помещает сообщение в очередь RabbitMQ, о необходимости вызова процедуры у любого из получателей сообщений. PHP считывает сообщение из очереди, выполняет какие либо действия и отправляет сообщение в Socket.io сервер.

## Инструменты разработки

Согласно архитектуре разрабатываемого приложения, необходимо разработать как клиентскую часть приложения, так и серверную.

Серверная часть приложения разработана на языке PHP с использованием фреймворка Symfony. Клиентская часть программного продукта была разработана на языке программирования JavaScript с использованием Nuxt.js и Vue.js

Для визуализации автоматизированного модуля администрирования проектных задач была выбрана библиотека теории графов с открытым исходным кодом, написанная на JS – Cytoscape.js. Данная библиотека предназначена для анализа и визуализации графиков.

Разработка модуля планирования работ

Согласно функциональным требованиям, каждая задача, которая отображается на сетевом графике в качестве вершины, должна называться определенным образом (Рисунок 1).



Рис.1. Отображение задачи на сетевом графике

Помимо отображения задач, необходимо визуализировать связи между ними. Связи могут быть двух типов: связь задач в одном разделе и связь задач в разных разделах. Сетевой график проектных работ, в котором узлы – это задачи, изображен на рисунке 2.

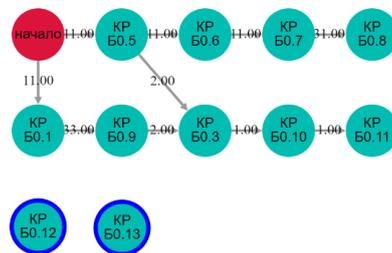


Рис. 2. Интерфейс работы приложения

Также библиотека cytoscape.js требует определенной структуры входных данных. Для приведения входных данных к необходимому формату был написан javascript код, который вызывается каждый раз при переходе пользователем на страницу разработанного

модуля. В связи с этим пользователю необходимо дождаться пока клиент обработает все данные, приведет их к необходимому формату и отрисует сетевой график. В процессе обработки данных пользователь не может взаимодействовать с другими элементами веб сайта, т.к. клиент приложения «занят» необходимыми вычислениями. Для оптимизации такого варианта использования в процессе реализации был применен механизм Web Workers.

Web Workers - это механизм, который позволяет скрипту выполняться в фоновом потоке, который отделен от основного потока веб-приложения. Преимущество заключается в том, ресурсоёмкие вычисления могут выполняться в отдельном потоке, позволяя запустить основной (обычно пользовательский) поток без блокировки и замедления [2].

Web Worker-ы предоставляют простое средство для запуска скриптов в фоновом потоке. Поток Worker'a может выполнять задачи без вмешательства в пользовательский интерфейс [2].

В результате вышеописанный код был вынесен в отдельный файл networkPlanning.js в качестве Web worker-a. Такая реализация позволила «освободить» пользовательский интерфейс от нагрузки, а также улучшить опыт взаимодействия с интерфейсом реализованного продукта.

Так как архитектура разработанного приложения преимущественно клиент-серверная, то предполагаются частые запросы на сервер. В процессе тестирования была обнаружена следующая уязвимость: при большом количестве запросов на сохранение данных в момент времени сервер в связи с большой загрузкой перестаёт отвечать. Для разрешения данной проблемы было принято решение использовать очередь сообщений

Очередь сообщений – программный компонент разработки программного обеспечения, который используется для межпроцессорного или межпоточкового взаимодействия в одном и том же процессе. Для обмена сообщений используется очередь.

Очереди сообщений предоставляют асинхронный протокол передачи данных, означая, что отправитель и получатель сообщения не обязаны взаимодействовать с очередью сообщений одновременно. Размещённые в очереди сообщения хранятся до тех пор, пока получатель не получит их [3].

В качестве «брокера сообщений» было использовано популярное программное решение RabbitMq.

Также для быстрого обновления данными с клиентом была использована JavaScript библиотека – SocketIO, в основном использующая протокол WebSocket.

## Заключение

В результате была разработан инструмент администрирования проектных работ проектной компании. Данный инструмент обладает следующим функционалом:

- Создание, удаление, обновление узлов.
- Создание, удаление, обновление ребер.
- Фильтрация графика по разделам.
- Сворачивания последовательных ребер.
- Позиционирования графа относительно точки начала.
- Отмены последнего действия.
- Отображение критического пути и его анимации.
- Подсветка несоединенных ребер.
- Предупреждения о создании цикла задач.
- Отображение карточки задачи.
- Быстрое создание задачи от текущей.

## Список литературы

1. Ахьюджа, Х. Н. Сетевые методы управления в проектировании и производстве [Текст] / Х.Н. Ахьюджа; перевод с англ. Б.С. Луныкова, В.М. Симонова, под ред. В.В. Калашникова. - М.: Мир, 1979. - 638 с.; Перевод изд.: Construction performance control by networks / H.N. Ahuja (New York etc., 1979); Дата обращения: 16.05.2019.
2. Использование Web Workers [Электронный ресурс] / Mozilla URL: [https://developer.mozilla.org/ru/docs/DOM/Using\\_web\\_workers](https://developer.mozilla.org/ru/docs/DOM/Using_web_workers), свободный. Яз. Рус. Дата обращения: 22.05.2019.
3. Очередь сообщений [Электронный ресурс] / Amazon URL: <https://aws.amazon.com/ru/message-queue>, свободный. Яз. Рус. Дата обращения: 20.05.2019