

УДК 004.4

**СИСТЕМА МОНИТОРИНГА И УПРАВЛЕНИЯ РАБОЧИМИ
ПЛОЩАДКАМИ
SYSTEM OF MONITORING AND MANAGEMENT OF WORKING
SERVERS**

П.Б. КОРНИЛОВ – бакалавр, Институт информационных технологий и радиоэлектроники, Кафедра ВТиСУ, группа ВТ-116, E-mail: acid186@mail.ru

М.А. ТРОФИМОВ – научный руководитель, старший преподаватель, Институт информационных технологий и радиоэлектроники, Кафедра ВТиСУ, E-mail: Mike_trofimov@mail.ru

P.B. KORNILOV – bachelor, Institute of information technologies and radio electronics, Department of VTiSU, group VT-116, E-mail: acid186@mail.ru

M.A. TROFIMOV – scientific advisor, Institute of information technology and radio electronics, Department of VTiSU, E-mail: Mike_trofimov@mail.ru

Аннотация: В статье кратко рассматриваются возможности проектируемой системы мониторинга и её отличия от аналогичных проектов, а также рассматриваются вопросы актуальности используемых в проекте технологий.

Abstract: This article briefly discusses the possibilities of a future monitoring system and its differences comparing to similar projects, and also discusses the relevance of the technologies used in the project.

Ключевые слова: фреймворк, мониторинг, система, Nuxt.js, JavaScript, WEB – приложение

Key words: framework, monitoring, system, Nuxt.js, JavaScript, WEB – application

С ростом и развитием каждой компании увеличивается число сотрудников, количество заказов, а также происходит ряд других изменений, но в то же время, например, в IT-компаниях возрастает запутанность между отделами и проектами. В растущей и активно развивающейся компании, специализирующейся на аутсорсинге по веб-разработке для digital-агентств и бизнеса, возникла необходимость в подобной системе из-за ряда факторов:

1. Разработчик тратит большое количество времени для получения сведений, может ли он работать на конкретном сервере или нет. Ответ может занимать до 1 ч, что очень долго, и задача простаивает;

2. Трудности в установлении сотрудника, который использовал сервер последним, что необходимо в случае ошибок или непредвиденных обстоятельств;

3. Прекращение засорения корпоративного чата по вопросам занятости серверов и использование его только в качестве чата для общения по деловым вопросам и вопросам организации компании.

В современном мире для всего есть готовое решение, система мониторинга не является исключением. Существует множество решений от разных компаний, но было принято решение отказаться от всевозможных готовых решений в пользу разработки собственной системы мониторинга и контроля за рабочими площадками по следующим причинам:

1. Создание только тех функций в приложении, которые нужны именно в данной компании;

2. Возможность модификации и масштабирования в будущем;

3. Бесплатное решение;

4. Повышение опыта начинающих разработчиков, назначенных на данный проект.

Согласно техническому заданию система должна предоставлять следующие возможности:

1. Просмотр всех площадок и всей информации, относящейся к данным площадкам, визуализация данных с сервера;
2. Возможность занять площадку – сообщить всем, что данный сервер занят, а также освободить площадку;
3. Вход в систему и выход из системы. Поддержание сессии;
4. Фильтрация площадок по различным параметрам – занятые, незанятые, по имени.

Для реализации подобного проекта можно воспользоваться одним из основных подходов в современной WEB–разработке:

1. Разработка приложения с помощью различных CMS–систем, например WordPress, Joomla и пр.;
2. Разработка с использованием фреймворков. Фреймворки – это программные продукты, реализующие многие полезные, часто используемые вещи и упрощающие создание приложений, например React, Vue.js, Angular и прочие;
3. Разработка с использованием стандартных инструментов в современной WEB–разработке, а именно: HTML, CSS, JavaScript.

Каждый из этих подходов обладает своими достоинствами и недостатками. Разработка на CMS–системе – это, по сути, настройка данной системы под собственные нужды, программирование применяется только в определенных случаях, используются для разработки сайтов с большой аудиторией.

Разработка с помощью стандартных инструментов – очень долгий процесс с вероятностью «изобрести велосипед» в ходе разработки. Используется в небольших проектах либо в модульных системах.

Разработка с использованием фреймворка – наиболее выгодное решение для реализации данной задачи, а именно с использованием Nuxt.js, по следующим причинам:

1. Использование функций фреймворка, а не создание их с нуля;
2. «Оптимизация из коробки»;
3. Комфортное использование и разработка;
4. Модульность;
5. Особенность профессии.

Реализация будет делиться на 2 части: Frontend и Backend. Все данные по проектам и площадкам будут приходить с сервера – части Backend. Frontend будет принимать эти данные и визуализировать их, а также предоставлять функционал взаимодействия с площадками, обеспечение авторизации в приложении, поддержание сессии, фильтрацию. В связи с этим в ходе реализации данного проекта можно выделить следующие моменты, которые необходимо продумать:

1. Способ передачи данных;
2. Хранение данных на стороне клиента и на сервере;
3. Визуализация данных, приходящих с сервера;

Пример предварительной диаграммы сущностей, хранящихся в БД на сервере, представлена на рисунке 1. Данные будут конвертироваться в JSON-объект и передаваться на Frontend-часть. Для организации Backend-части будет использована «CMS 1С-Битрикс: Управление сайтом», данная CMS используется в большинстве проектов компании, и разработчики концентрируют свои навыки разработки именно в данной CMS.

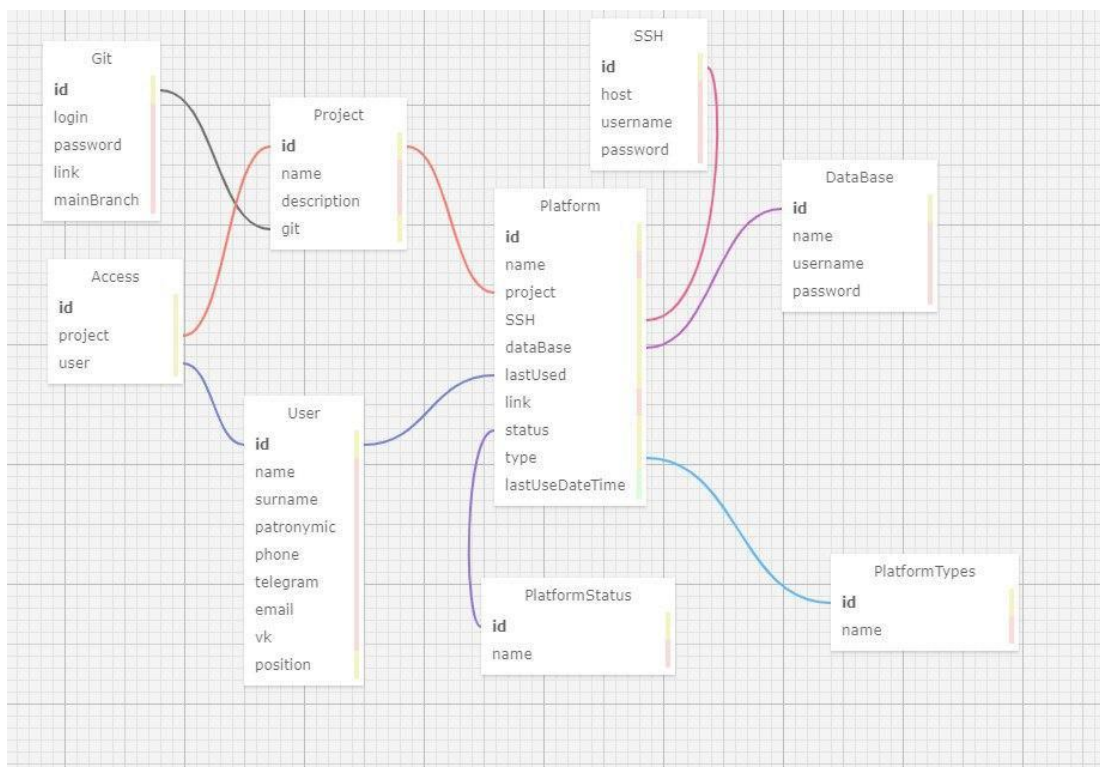


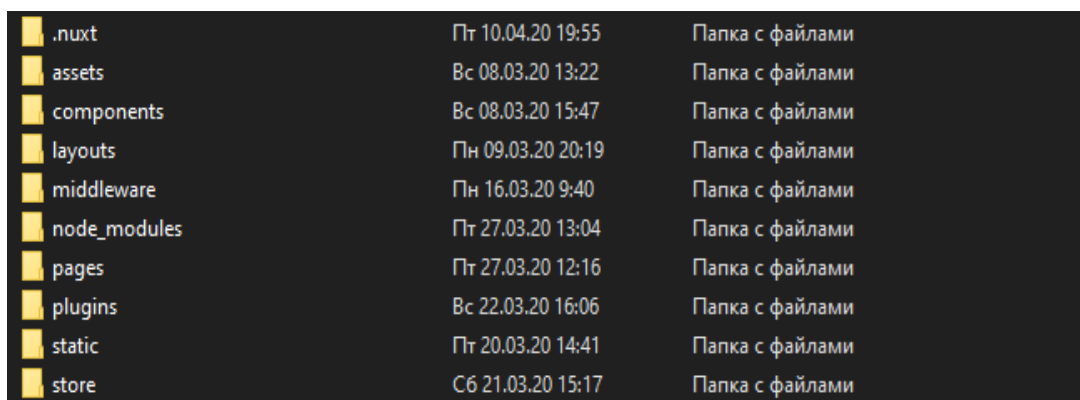
Рисунок 1 – Предварительная диаграмма сущностей

Способ передачи данных: Поскольку в качестве фреймворка был выбран Nuxt.js, можно использовать его свойство модульности для обеспечения работы различных дополнительных функций. Например, модуль «Axios» – безопасный HTTP–клиент для обмена данными, отлично подойдет для обмена данными между сервером и клиентом посредством AJAX–запросов. Данный модуль позволяет гибко настроить его использование по сравнению с другими плагинами, например, настраивать заголовки для сервера, использовать прокси, определять базовый URL–адрес и т.д., что делает его выбор очевидным.

Способ хранения данных: Nuxt.js является фреймворком для Vue.js, который сочетает в себе плагин Vuex. Данный плагин позволяет организовать глобальное хранилище для всего приложения. Преимущество данного плагина заключается в том, что при изменении данных в Vuex

данные, которые полагаются на него, будут реактивно изменены на странице в браузере.


Визуализация данных: основной задачей данного проекта является визуализация данных, приходящих с сервера и предоставление функционала взаимодействия пользователя с площадками. Данный фреймворк предоставляет удобную среду для разработки, каждая папка в структуре проекта имеет собственное предназначение. Структура папок проекта представлена на рисунке 2.



.nuxt	Пт 10.04.20 19:55	Папка с файлами
assets	Вс 08.03.20 13:22	Папка с файлами
components	Вс 08.03.20 15:47	Папка с файлами
layouts	Пн 09.03.20 20:19	Папка с файлами
middleware	Пн 16.03.20 9:40	Папка с файлами
node_modules	Пт 27.03.20 13:04	Папка с файлами
pages	Пт 27.03.20 12:16	Папка с файлами
plugins	Вс 22.03.20 16:06	Папка с файлами
static	Пт 20.03.20 14:41	Папка с файлами
store	Сб 21.03.20 15:17	Папка с файлами

Рисунок 2 – Структура папок проекта

Каждая страница в проекте представляет собой файл с расширением vue. Структура данного файла представлена на рисунке 3.



```
1  <template>
2    <div>
3
4    </div>
5  </template>
6
7  <script>
8    export default {
9
10   }
11  </script>
12
13  <style lang="">
14
15  </style>
16
```

Рисунок 3 – Структура файла с расширением vue

Каждая секция отвечает за свой аспект в разработке. Секция «template» отвечает за создание HTML-кода, «script» – для программирования и работы с модулями, «style» – для стилизации компонентов и элементов HTML. Nuxt.js предоставляет возможность модульной разработки HTML-кода – компонент можно использовать сколько угодно раз, что значительно упрощает разработку и снижает время реализации.

Пример страницы авторизации представлен на рисунке 4. Предварительная основная страница представлена на рисунке 5.



Рисунок 4 – Страница авторизации

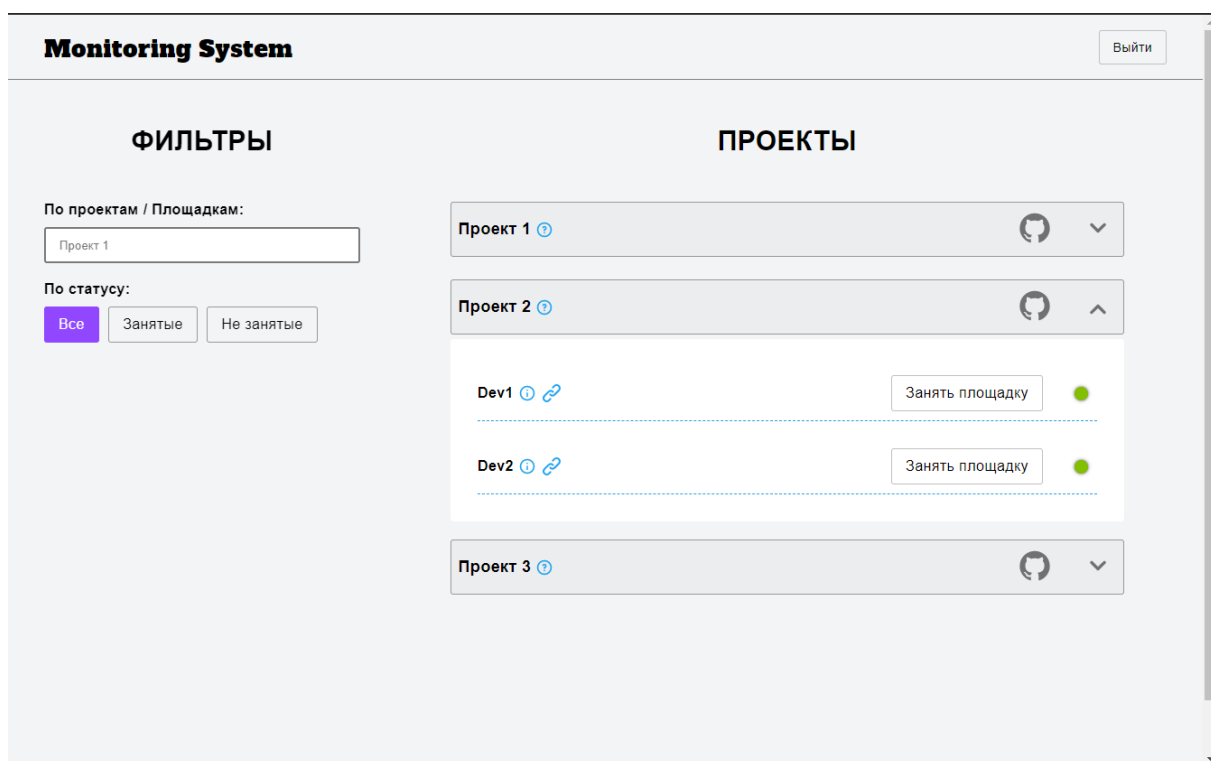


Рисунок 5 – Предварительная основная страница.

По окончании работ над данным проектом критерием оценки будет служить соответствие реализованных функций функциям ТЗ, а именно:

1. Просмотр всех площадок и всей информации, относящейся к данным площадкам, визуализация данных с сервера;
2. Возможность занять площадку – сообщить всем, что данный сервер занят, а также освободить площадку;
3. Вход в систему и выход из системы. Поддержание сессии;
4. Фильтрация площадок по различным параметрам – занятые, свободные, по имени.

Список используемой литературы:

1. Nuxt.js – официальная документация [Электронный ресурс], - <https://nuxtjs.org/>

2. Vue – Официальная документация [Электронный ресурс], -
<https://vuejs.org/>