

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор

_____ Л.Б. Соколинский

«___»_____ 2024 г.

**Разработка фронтенда модуля по работе со строительными
объектами корпоративной системы
ООО «ЭнергоИнжиниринг»**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2024.308-295.ВКР

Научный руководитель,
доцент кафедры СП, к.ф.-м.н.
_____ А.Т. Латипова

Автор работы,
студент группы КЭ-401
_____ З.А. Баландин

Ученый секретарь
(нормоконтролер)
_____ И.Д. Володченко
«___»_____ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

УТВЕРЖДАЮ
Зав. кафедрой СП
_____ Л.Б. Соколинский
29.01.2024 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы бакалавра

студенту группы КЭ-401

Баландину Захару Алексеевичу,
обучающемуся по направлению

02.03.02 «Фундаментальная информатика и информационные технологии»

1. Тема работы (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)

Разработка фронтенда модуля по работе со строительными объектами
корпоративной системы ООО «ЭнергоИнжиниринг».

2. Срок сдачи студентом законченной работы: 03.06.2024 г.

3. Исходные данные к работе

3.1. Документация API бэкенд-сервера корпоративной системы
ООО «ЭнергоИнжиниринг».

3.2. Приказ ООО «ЭнергоИнжиниринг» №18 от 10.02.2024 г. «О разработке
корпоративной системы электронного документооборота» (дата обращения:
11.02.2024 г.).

3.3. Garaguso P.D. Vue.js 3 Design Patterns and Best Practices. // Packt Publish-
ing, 2023. – 296 p.

3.4. Документация Vue 3. [Электронный ресурс] URL: <https://vuejs.org/guide>
(дата обращения: 02.02.2024 г.).

3.5. Документация Vuex. [Электронный ресурс] URL: <https://vuex.vuejs.org/>
(дата обращения: 07.02.2024 г.).

4. Перечень подлежащих разработке вопросов

- 4.1. Выполнить анализ предметной области.
- 4.2. Спроектировать клиентскую часть модуля.
- 4.3. Реализовать клиентскую часть модуля.
- 4.4. Провести тестирование.

5. Дата выдачи задания: 29.01.2024 г.

Научный руководитель,
доцент кафедры СП, к.ф.-м.н.

А.Т. Латипова

Задание принял к исполнению

З.А. Баландин

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	7
1.1. Предметная область проекта	7
1.2. Средства разработки.....	9
2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ	11
2.1. Требования к проектируемой системе.....	11
2.2. Диаграмма вариантов использования.....	12
3. АРХИТЕКТУРА СИСТЕМЫ.....	14
3.1. Общее описание архитектуры системы.....	14
3.2. Описание API	16
3.3. Компоненты разрабатываемой системы.....	18
3.4. Диаграмма последовательности.....	19
4. РЕАЛИЗАЦИЯ СИСТЕМЫ	22
4.1. Реализация клиент-серверного взаимодействия.....	22
4.2. Реализация компонентов.....	27
4.3. Реализация компонентов-шаблонов	32
4.4. Реализация маршрутизации	34
4.5. Реализация валидации форм ввода данных	35
4.6. Реализация системы прав по работе с объектами	38
4.7. Реализация выбора темы приложения	40
5. ТЕСТИРОВАНИЕ СИСТЕМЫ.....	45
5.1. Тестирование системы.....	45
5.2. Опытное внедрение системы	49
ЗАКЛЮЧЕНИЕ	51
ЛИТЕРАТУРА.....	52
ПРИЛОЖЕНИЯ.....	55
Приложение А. Спецификация вариантов использования.....	55
Приложение Б. Жизненный цикл Vue-компонента.....	63
Приложение В. Скриншоты системы	64
Приложение Г. Скриншоты аналогичных проектов	67

ВВЕДЕНИЕ

Актуальность

На сегодняшний день множество компаний нуждаются в корпоративных системах, позволяющих автоматизировать, анализировать, отслеживать все процессы, происходящие в компании.

Деятельность ООО «ЭнергоИнжиниринг» имеет индивидуальные особенности, поэтому внедрение крупных готовых систем требует наличие особого штата сотрудников и дорогостоящего сопровождения для автоматизации всех процессов предприятия. Стоит отметить тенденцию ухода ряда корпоративных систем с российского рынка. Таким образом, тема выпускной квалификационной работы является актуальной.

Постановка задачи

Целью выпускной квалификационной работы является разработка фронтенда модуля по работе со строительными объектами корпоративной системы ООО «ЭнергоИнжиниринг».

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) выполнить анализ предметной области;
- 2) спроектировать клиентскую часть модуля;
- 3) реализовать клиентскую часть модуля;
- 4) провести тестирование.

Структура и содержание работы

Работа состоит из введения, пяти глав, заключения, списка литературы и приложения. Объем работы составляет 68 страниц, объем литературы – 28 источников.

В первой главе проводится анализ предметной области, анализ аналогов систем по работе со строительными объектами.

Во второй главе описываются функциональные и нефункциональные требования к разрабатываемой системе, рассмотрены варианты использования приложения.

В третьей главе описывается архитектура системы, файловая система проекта, описание API. Приведены диаграмма компонентов, описывающая компоненты системы и взаимосвязи между ними, и диаграмма последовательности, визуализирующая взаимодействие компонентов системы и их порядок.

В четвертой главе описана реализация системы: клиент-серверное взаимодействие, управление состоянием приложения через глобальное хранилище, реализация Vue-компонентов, шаблонов, маршрутизации в приложении, системы прав, валидации форм ввода данных, а также реализация светлой и темной тем.

В пятой главе описан процесс кросс-браузерного, функционального тестирования, приведен протокол тестирования основных аспектов работы разрабатываемой системы. Описан опыт внедрения приложения на предприятии.

В заключении приведены основные результаты работы и направления дальнейшей работы.

В приложении А представлена спецификация вариантов использования.

В приложении Б представлен жизненный цикл Vue-компонента.

В приложении В представлены скриншоты системы.

В приложении Г представлены скриншоты аналогичных проектов.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Предметная область проекта

ООО «ЭнергоИнжиниринг» – строительная компания, предлагающая полный спектр услуг и работ в сфере строительства, осуществляющая строительство объектов любой сложности [1]. Одни из направлений деятельности компании: строительство современных мини-ТЭС (тепловые электростанции), электрических станций и подстанций, линий электропередач любой сложности, выполнение пусконаладочных работ.

Для строительной компании очень важно оптимизировать бизнес-процессы, иметь единую информационную базу, улучшать взаимодействие как сотрудников и руководителей, так и сотрудников разных отделов

Компанией было принято решение разработать собственную корпоративную систему, объединяющую различные сферы бизнес-процессов в строительстве, удовлетворяя специфике работы предприятия. Цель, предметная область разрабатываемой системы, а также постановка задач описаны в приказе о разработке корпоративной системы. Исходя из текста приказа, необходимо разработать и внедрить корпоративную систему с модулем по работе со строительными объектами, используя веб-технологии.

Анализ аналогичных проектов

Gectaro

Gectaro [2] – российский облачный сервис для автоматизации управления строительством и ремонтом. Система имеет модули создания смет, ведения финансового учета, составления графика работ, управления снабжением, контроля стройки, ведения складского учета, справочника расценок, документов и актов, управления отчетами и постановки задач.

Существует интеграция с множеством сервисов – AmoCRM, Google Drive, Trello, 1С и другие, импорты из Гранд-сметы, Смета.ру и других программ.

Сервис имеет приятный современный интерфейс, система доступна пользователям через браузер, данные хранятся в облаке. Существует отдельное мобильное приложение для прорабов. Интерфейс системы представлен на рисунке 1.

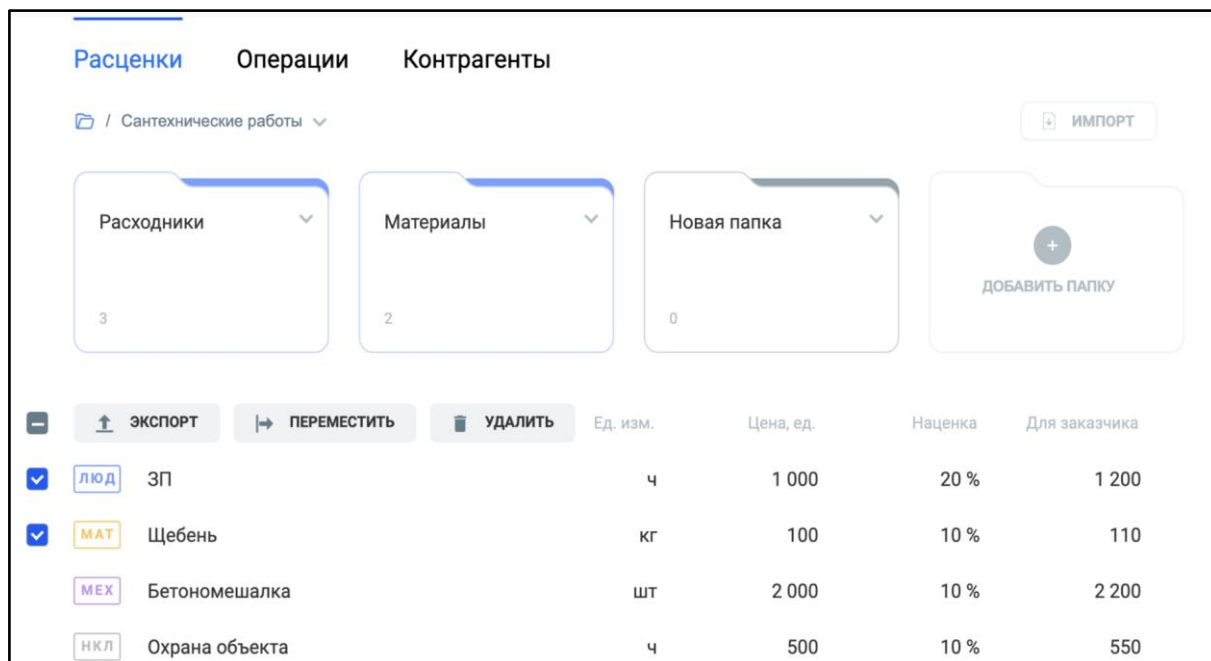


Рисунок 1 – Интерфейс Gecstaro

Ключевой минус – ограничение на количество пользователей. Самый дорогой тариф «Профи» предлагает до 30 пользователей, что недостаточно для нашего предприятия. Программные модули не полностью охватывают деятельность предприятия.

Цифровое управление строительством

Цифровое управление строительством [3] – облачная платформа для контроля строительства и автоматизации работы заказчика, подрядчиков и строительного контроля. Одним из продуктов платформы является управление строительными проектами и планирование. Он включает в себя ведение смет, календарно-сетевых графиков с поддержкой диаграммы Ганта, автоматизацию работы с документами и стройматериалами. Также продукт включает задачник.

Огромным достоинством системы является хранение предыдущих версий документов, смет с возможностью их сравнения.

Система обладает удобным современным веб-интерфейсом, который представлен на рисунке 8 приложения Г.

MacroERP

MacroERP [4] – программное обеспечение для управления строительством. Интерфейс системы представлен на рисунке 9 приложения Г.

Система автоматизирует процессы управления строительными проектами и включает ведение графика проекта, контроль качества работ, документооборот, управление себестоимостью проекта, финансами, материально-техническими ресурсами, снабжением.

У системы существует интеграция с профильными отраслевыми продуктами – 1С, ГрандСмета, РиК.

Анализ аналогичных проектов показал, что текущие решения обладают удобством, автоматизацией, интеграцией различных сфер в области строительства. Однако эти программные не отвечают специфике работы и реализации бизнес-процессов предприятия. Решения покрывают лишь часть работы того или иного отдела компании: для других отделов либо нет решения, либо оно недостаточно автоматизирует их процессы, что приводит к отсутствию мотивации у сотрудников работать в этой системе.

1.2. Средства разработки

Для разработки модуля по работе со строительными объектами корпоративной системы используются следующие технологии.

1. HTML 5 [5] – язык гипертекстовой разметки, используемый для структурирования и отображения веб-страниц.

2. CSS 3 [6] – язык стилей, применяемый для представления внешнего вида веб-страницы. Для повышения эффективности и удобства написания стилей используется препроцессор Sass [7].

3. JavaScript [8] – мультипарадигменный язык программирования, используемый в веб-разработке для создания интерактивных веб-сайтов.

4. Vue.js 3 [9] – JavaScript-фреймворк для создания пользовательских интерфейсов. Используется для создания реактивных одностраничных и прогрессивных веб-приложений.

5. Axios [10] – HTTP-клиент, основанный на JavaScript, использующийся для отправки, перехвата запросов на сервер и обработки его ответов.

6. Vuex [11] – библиотека для Vue.js, реализующая глобальное управление состоянием приложения.

7. Vue Router [12] – библиотека для Vue.js, реализующая маршрутизацию в приложении.

8. Vuelidate [13] – библиотека для Vue.js, используемая для валидации данных – проверки значений на соответствие правилам, определенных предметной областью, тем самым упрощая создание различных форм.

9. Vuetify [14] – UI-фреймворк для разработки пользовательских интерфейсов на Vue.js. Фреймворк предлагает реализации различных компонентов: от кнопок и форм ввода данных до таблиц с фильтрацией, сортировкой и пагинацией; поддержку различных наборов иконок, например, Material Design Icons. Кроме того, Vuetify является CSS-фреймворком, который поддерживает возможность внедрения в приложение светлую и темную тему.

Вывод по первой главе

В первой главе описана предметная область проекта. Анализ аналогичных проектов показал разнообразие готовых решений, способных автоматизировать бизнес-процессы компаний, однако не отвечающих специфике работы ООО «ЭнергоИнжиниринг». Также определены средства разработки системы.

2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

2.1. Требования к проектируемой системе

Функциональные требования к проектируемой системе

Функциональные требования определяют то, что хочет видеть заказчик в качестве результата – необходимое ему поведение системы. В результате анализа были выделены функциональные требования к разрабатываемому веб-приложению.

1. Должна быть реализована система авторизации и аутентификации.
2. Должна быть реализована система создания, редактирования строительных объектов.
3. Должен быть реализован жизненный цикл строительного объекта с отслеживанием прогресса его освоения по договору.
4. Должна быть реализована система прав, основанная на роли пользователя.
5. Должна быть реализована панель администратора для просмотра пользователей с возможностью их редактирования.

Нефункциональные требования к проектируемой системе

Нефункциональные требования предъявляют качественные характеристики проектируемой системы – как должна проявлять система поведение, определенное в функциональных требованиях.

Разрабатываемое веб-приложение имеет следующие нефункциональные требования.

1. Приложение должно иметь приятный и понятный пользователю интерфейс.
2. Приложение должно быть написано на Vue.js 3.
3. Приложение должно взаимодействовать с внутренним API.
4. Приложение должно быть расширяемым.
5. Приложение должно быть кроссплатформенным.

3. АРХИТЕКТУРА СИСТЕМЫ

3.1. Общее описание архитектуры системы

Современные JavaScript-фреймворки, в частности, Vue.js, с помощью которого разрабатывается система, закладывают основу архитектуры, которая в дальнейшем применяется и расширяется разработчиком.

Веб-приложение на Vue.js строится из компонентов – переиспользуемых экземпляров Vue. Каждый компонент инкапсулирует в себе верстку и логику, позволяющую получать данные, обрабатывать и представлять их. Компоненты могут входить в состав других компонентов, при этом выстраивается связь «родитель-потомок». Родители могут передавать потомку входные параметры, называемые «пропсами» (от англ. props), а потомки могут возвращать данные родительскому компоненту.

Однако если необходимо передать в несвязанные друг с другом компоненты одни и те же данные или использовать их в связанных, но достаточно высокой уровнем вложенности, компонентах, то возникает проблема в том, что разработчику придется прокидывать данные от корневого компонента до необходимого, при этом не во всех компонентах, куда передаются данные, он может быть необходим. Связанность между компонентами растет, читаемость, масштабируемость кода падает. Чтобы избежать этой проблемы, в приложении будет использоваться Vuex – библиотека для Vue.js, позволяющая организовывать управление состоянием. Благодаря данной технологии выносятся данные в глобальное хранилище, откуда оно доступно всем компонентам. Также Vuex задает свой паттерн управления глобальными данными, реализация которого будет рассмотрена в 4 главе.

Компоненты входят также в состав страниц приложения. Страницы представляют из себя компоненты: они так же, как и компоненты, инкапсулируют в себе верстку, скрипты, стили, однако их рендеринг происходит, когда пользователь перейдет по заданному для каждой страницы URL-адресу.

Файловая система

Файловая система разрабатываемого приложения следующая.

1. Папка `node_modules` – содержит установленные зависимости для проекта.

2. Папка `public` – содержит единственный файл – `index.html`. В разметке этого файла находится элемент с классом «`app`». В этот элемент Vue будет генерировать верстку и реактивно ее изменять, тем самым реализуя одностраничное приложение (SPA).

3. Папка `src` – содержит исходный код приложения.

Структура директории `src` следующая.

1. Папка `assets` – содержит статические изображения, а также шрифты.

2. Папка `components` – в данной директории содержатся Vue-компоненты.

3. Папка `layouts` – содержит компоненты, представляющие собой шаблоны. Их удобно использовать, если есть определенные компоненты, верстка, которые могут повторяться на разных страницах.

4. Папка `helpers` – содержит вспомогательные функции, которые импортируются в компоненты. `assets` – содержит статические изображения, а также шрифты.

5. Папка `plugins` – содержит в себе установленные для проекта плагины с различными настройками.

6. Папка `router` – в нем расположен файл, отвечающий за маршрутизацию в приложении.

7. Папка `store` – глобальное хранилище, реализованное с помощью библиотеки `Vuex`. В нем располагаются отдельные модули, отвечающие за различные части разрабатываемой системы.

8. Папка `views` – в данной директории расположены страницы – компоненты, которые передаются в роутер и отображаются при заданном для каждой страницы URL-адресе.

Модульная архитектура

С ростом числа компонентов, связей между ними, а также ростом бизнес логики появляется проблема масштабируемости фронтенд-приложения: папка `components` содержит в себе десятки, а то и сотни компонентов, каждый из которых отвечает за разный модуль и, более того, одни содержат в себе бизнес-логику, а другие нет: в одной папке на одном и том же уровне могут находиться как модальное окно, которое собирает данные, отправляет на сервер, обрабатывает ответ, так и элемент интерфейса, например, кнопка, лишенная бизнес-логики. Такой подход возможен в небольших приложениях, однако если в разрабатываемой системе множество модулей, необходимо заранее задумываться об архитектуре фронтенд-приложения.

В разрабатываемой системе используется модульная архитектура. Папка `components` содержит в себе подкаталоги, каждый из которых содержит компоненты, отвечающие строго за определенный модуль в системе. Выделяется отдельно слой `App`, который представлен соответствующей папкой: здесь содержатся компоненты, как правило, являющиеся элементами графического интерфейса, они лишены бизнес-логики, либо имеют ее минимальную составляющую, при этом данные компоненты менее всего поддерживаются изменениям, и они используются во всем приложении, в том числе, в компонентах, входящих в самостоятельные модули. Остальные папки разбиты по определенным модулям: например, каталог `BuildingObjects` содержит компоненты, отвечающие только за работу с модулем `Объекты`. В этих компонентах максимально сосредоточена логика, связанная с получением, хранением, обработкой данных, связанных со строительными объектами.

3.2. Описание API

Для реализации разрабатываемой системы используется серверный API. Клиент-серверное взаимодействие определяется архитектурным сти-

stance) с заданными конфигурациями такими как URL-адрес бэкенда, HTTP-заголовками используются в Vuex – официальной библиотеке для Vue.js, которая реализует глобальное хранилище состояния Vue-приложения. В глобальном хранилище сконцентрирована бизнес-логика приложения: клиент-серверное взаимодействие, состояние, описывающее предметную область разрабатываемой системы и правила, характеризующее его изменение, а также вычисляемые свойства. Через вспомогательные функции `mapState`, `mapGetters`, `mapMutations` и `mapActions` реализуется доступ к состоянию, геттерам, мутациям и действиям глобального хранилища из Vue-компонента. В компонентах сконцентрирована логика представления.

Таким образом, благодаря использованию глобального хранилища происходит разделение логики представления, которая применяется в компонентах, и бизнес-логики.

Для визуального представления описанных выше компонентов, а также зависимостей между ними используется UML-диаграмма компонентов, представленная на рисунке 3.

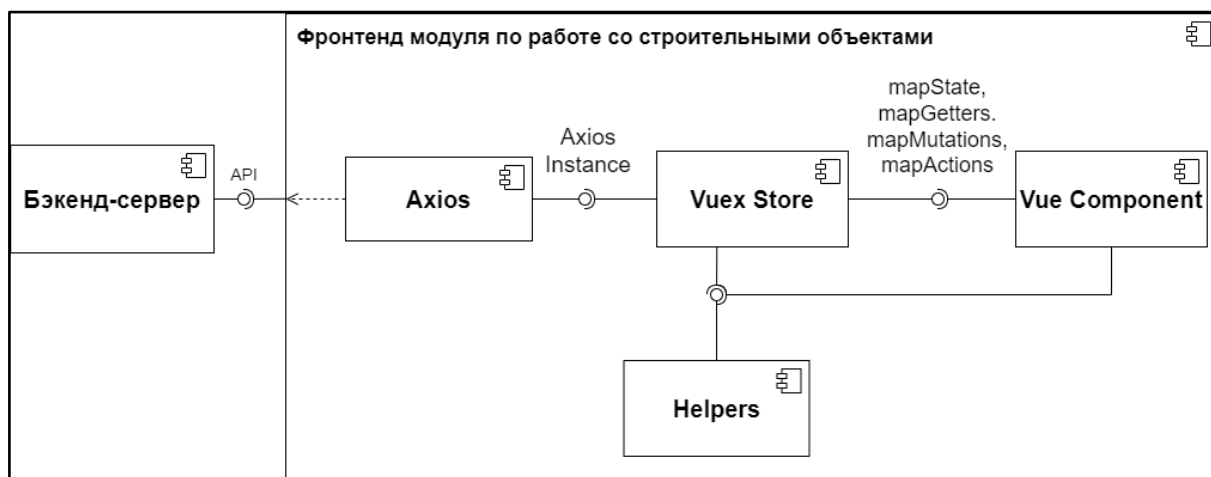


Рисунок 3 – Диаграмма компонентов системы

3.4. Диаграмма последовательности

Для визуализации взаимодействия компонентов разрабатываемой системы и их порядка составлена UML-диаграмма последовательности. На

редается обратно в модуль хранилища Vuex. После возвращения фокуса управления, Vuex обновляет состояние – добавляет в список строительных объектов новый созданный объект и возвращает результат компоненту модального окна создания строительного объекта. Анимация загрузки завершается, модальное окно создания строительного объекта закрывается, пользователь получает сообщение о результате создания объекта. На странице строительных объектов обновляется список объектов в соответствии с измененным состоянием в модуле Vuex.

Вывод по третьей главе

С учетом требований к разрабатываемой системе, а также выбранных технологий, была спроектирована архитектура системы. Клиент-серверное взаимодействие определяется архитектурным стилем REST-API. В основе Vue.js приложения лежит взаимодействие Vue-компонентов, разделенных на модули, с глобальным хранилищем Vuex. Для визуального представления компонентов системы и их зависимостей разработана UML-диаграмма компонентов, а для визуализации взаимодействия компонентов разрабатываемой системы – диаграмма последовательности.

4. РЕАЛИЗАЦИЯ СИСТЕМЫ

4.1. Реализация клиент-серверного взаимодействия

Создание экземпляра Axios

Для реализации клиент-серверного взаимодействия в системе используется Axios – HTTP-клиент, который позволяет отправлять запросы на сервер. Библиотека крайне удобна тем, что она основана на промисах [18] – специальном объекте языка JavaScript, который используется для обработки асинхронного кода.

Чтобы каждый раз не задавать различные параметры запроса (URL-адрес, заголовки и др.), библиотека позволяет создавать экземпляры Axios с пользовательской конфигурацией [19]. Создание экземпляра Axios представлено в листинге 1.

Листинг 1 – Создание экземпляра Axios

```
const instance = axios.create({
  baseURL: process.env.VUE_APP_API_URL,
  headers: {
    "Content-Type": "application/json",
  },
});
```

Создание экземпляра происходит с помощью метода `create`, где задается базовый URL-адрес сервера и заголовок, который говорит о том, что общение клиента и сервера будет происходить с помощью JSON. Далее можно использовать этот экземпляр для отправки запросов, но при этом уже не нужно будет указывать адрес, а также передавать стандартные заголовки.

Реализация глобального хранилища

Как говорит официальная документация, Vuex является не только библиотекой для Vue.js, но и паттерном (шаблоном) управления состоянием приложения. Vuex задает свои правила. Для логической организации и масштабируемости хранилище разделено на модули – каждый модуль соответствует сущности, определяемой серверным API.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы был разработан фронтенд модуля по работе со строительными объектами корпоративной системы ООО «ЭнергоИнжиниринг».

При этом были решены следующие задачи:

- 1) выполнен анализ предметной области;
- 2) спроектирована клиентская часть модуля;
- 3) реализована клиентская часть модуля;
- 4) проведено тестирование.

Направления дальнейшей работы

Направления дальнейшей работы и практической разработки следующие:

- 1) миграция с JavaScript на TypeScript [26];
- 2) миграция с Vuex на Pinia [27];
- 3) миграция с Vue CLI на Vite [28];
- 4) миграция с Options API на Composition API;
- 5) разработка новых модулей;
- 6) адаптация системы под мобильные устройства;
- 7) история последних действий ответственных по объекту.

ЛИТЕРАТУРА

1. ООО «ЭнергоИнжиниринг». [Электронный ресурс] URL: <http://energoinz.ru/> (дата обращения: 19.04.2024 г.).
2. Gectaro. [Электронный ресурс] URL: <https://gectaro.com/> (дата обращения: 19.04.2024 г.).
3. Цифровое управление строительством. [Электронный ресурс] URL: <https://tsus.ru/> (дата обращения: 19.04.2024 г.).
4. MacroERP. [Электронный ресурс] URL: <https://macroerp.ru/> (дата обращения: 19.04.2024 г.).
5. HTML. [Электронный ресурс] URL: <https://developer.mozilla.org/ru/docs/Web/HTML> (дата обращения: 20.03.2024 г.).
6. CSS: каскадные таблицы стилей. [Электронный ресурс] URL: <https://developer.mozilla.org/ru/docs/Web/CSS> (дата обращения: 20.03.2024 г.).
7. Sass. [Электронный ресурс] URL: <https://sass-lang.com/> (дата обращения: 20.03.2024 г.).
8. JavaScript. [Электронный ресурс] URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата обращения: 20.03.2024 г.).
9. Что такое Vue? [Электронный ресурс] URL: <https://ru.vuejs.org/guide/introduction.html> (дата обращения: 20.03.2024 г.).
10. Axios. Введение. [Электронный ресурс] URL: <https://axios-http.com/ru/docs/intro> (дата обращения: 14.04.2024 г.).
11. What is Vuex? [Электронный ресурс] URL: <https://vuex.vuejs.org> (дата обращения: 14.04.2024 г.).
12. Vue Router. [Электронный ресурс] URL: <https://vue-router.ru.netlify.app/introduction.html> (дата обращения: 14.04.2024 г.).
13. Vuelidate. [Электронный ресурс] URL: <https://vuelidate-next.netlify.app/> (дата обращения: 20.03.2024 г.).

14. Vuetify. [Электронный ресурс] URL: <https://vuetifyjs.com/en/> (дата обращения: 20.03.2024 г.).
15. Использование диаграммы вариантов использования UML при проектировании программного обеспечения. [Электронный ресурс] URL: <https://habr.com/ru/articles/566218> (дата обращения: 14.02.2024 г.).
16. Разбираемся, что такое REST API. [Электронный ресурс] URL: <https://vc.ru/u/941519-olegs-sustenkovs/315926-razbiraemsysya-cto-takoe-rest-api> (дата обращения: 12.04.2024 г.).
17. Introducing JSON. [Электронный ресурс] URL: <https://www.json.org/json-en.html> (дата обращения: 14.04.2024 г.).
18. Промисы. [Электронный ресурс] URL: <https://learn.javascript.ru/promise-basics> (дата обращения: 14.04.2024 г.).
19. Экземпляр Axios. [Электронный ресурс] URL: <https://axios-http.com/ru/docs/instance> (дата обращения: 14.04.2024 г.).
20. Однофайловые компоненты Vue.js. [Электронный ресурс] URL: <https://v3.ru.vuejs.org/ru/guide/single-file-component.html> (дата обращения: 17.03.2024 г.).
21. Хуки жизненного цикла. [Электронный ресурс] URL: <https://ru.vuejs.org/guide/essentials/lifecycle> (дата обращения: 14.04.2024 г.).
22. Слоты. [Электронный ресурс] URL: <https://ru.vuejs.org/guide/components/slots> (дата обращения: 19.04.2024 г.).
23. Примеси. [Электронный ресурс] URL: <https://v3.ru.vuejs.org/ru/guide/mixins.html> (дата обращения: 14.04.2024 г.).
24. Раз и навсегда про темную тему или почему вам «на самом деле» нужна темная тема. [Электронный ресурс] URL: <https://habr.com/ru/articles/695468> (дата обращения: 08.04.2024 г.).
25. Vuetify Theme configuration. [Электронный ресурс] URL: <https://vuetifyjs.com/en/features/theme> (дата обращения: 08.04.2024 г.).
26. TypeScript. [Электронный ресурс] URL: <https://www.typescriptlang.org> (дата обращения: 08.04.2024 г.).

27. Pinia. [Электронный ресурс] URL: <https://pinia-ru.netlify.app> (дата обращения: 08.04.2024 г.).

28. Vite. [Электронный ресурс] URL: <https://vitejs.dev> (дата обращения: 25.04.2024 г.).

Приложение Б. Жизненный цикл Vue-компонента

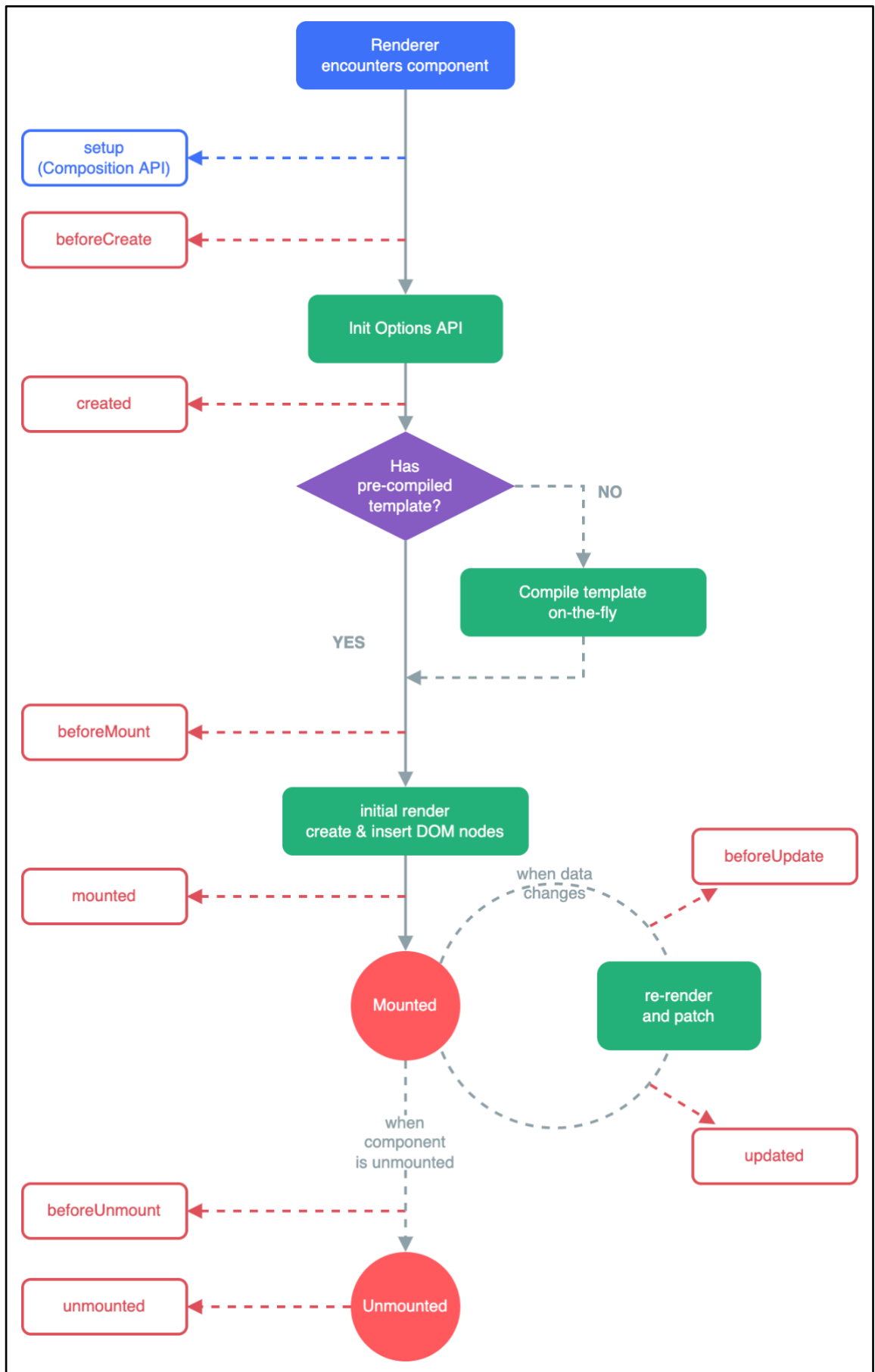


Рисунок 1 – Жизненный цикл Vue-компонента

Приложение Г. Скриншоты аналогичных проектов

Строительство ЦЦСР
Проект для демонстрации

123.820 млн из 101.086 млн

График производства работ Аналитика Версии графика Стадии реализации

График производства работ
Версия: График СМР - Актуальная

Наименование	Объем	Единицы	Сумма	Вес	План Начало	План Окончание	Объем	Факт Начало	Факт Окончание	Объем	Прогноз	Прогноз Окончание	Откл. от дир.	Откл. от опер.	Откл. динам.
Объект	100	%	90 393 771.88	0	12.11.2021	18.05.2030	100 + X	31.08.2023	13.03.2024	53.258...	+ X	31.08.2023	24.06.2032	-768	-6
1 Подготовительный период	100	%	139 873.17	0.04%	12.11.2021	13.09.2029	100 + X	01.09.2023	19.12.2023	46.966...	+ X	01.09.2023	23.10.2031	-770	-20
1.1 Глава 1. "Подготовка территории строительства"	100	%	139 873.17	100	12.11.2021	13.09.2029	100 + X	01.09.2023	19.12.2023	46.966...	+ X	01.09.2023	23.10.2031	-770	-20
1.1.1 ЛС-01-01-01	100	%	139 873.17	100	12.11.2021	13.09.2029	100 + X	01.09.2023	19.12.2023	46.966...	+ X	01.09.2023	23.10.2031	-770	-20
1.1.1.1.1 НС Белое	100	%	119 943.72	1.761%	12.11.2021	13.09.2029	100 + X	01.09.2023	19.12.2023	44.247...	+ X	01.09.2023	23.10.2031	-770	-18
1.1.1.1.1 Срезка кустарника и мелколесья в грунтах естест	0.072	га	413.26	1.761%	12.11.2021	13.09.2029	0.072 + X	дд.мм.гггг	дд.мм.гггг	...		21.12.2023	23.10.2031	-770	27
1.1.1.1.2 Разработка грунта с перемещением до 10 м булы	0.1596	1000 м3	100 000	14.98	12.11.2021	13.09.2029	0.1596 + X	22.09.2023	23.09.2023	0.001596	X	22.09.2023	28.12.2023	2086	-35
								01.09.2023	04.09.2023	0.0266	X				
								01.09.2023	02.09.2023	0.019152	X				
1.1.1.1.3 При перемещении грунта на каждые последующи	0.1596	1000 м3	3 256.24	13.87	12.11.2021	13.09.2029	0.1596 + X	04.09.2023	05.09.2023	0.01596	X	01.09.2023	12.10.2023	2163	-74
								22.09.2023	23.09.2023	0.001596	X				
								04.10.2023	12.10.2023	0.096292	+ X				
1.1.1.1.4 Разработка грунта с погрузкой на автомобили-са	0.01976	1000 м3	2 276.88	9.705	12.11.2021	13.09.2029	0.01976 + X	01.09.2023	02.09.2023	0.002371	X				
								04.09.2023	05.09.2023	0.001976	X	01.09.2023	14.01.2024	2069	3
								22.09.2023	23.09.2023	0.000198	+ X				
1.1.1.1.5 Перевозка грузов автомобилями-самосвалами гр	34.58	1 т груза	13 440.08	57.28	12.11.2021	13.09.2029	34.58 + X	19.12.2023	19.12.2023	10	+ X	19.12.2023	24.12.2023	2090	-2
								01.09.2023	04.09.2023	0.12	X				
								01.09.2023	02.09.2023	0.0864	X				
1.1.1.1.6 Планировка площадей бульдозерами мощностью	0.72	1000 м2	557.26	2.375	12.11.2021	13.09.2029	0.72 + X	04.09.2023	05.09.2023	0.072	X	01.09.2023	22.12.2023	2092	-73
								22.09.2023	23.09.2023	0.0072	X				

Рисунок 8 – Цифровое управление строительством

Демонстрация

Заявки Недвижимость Финансы Контакты Снабжение Стройка Документы Проекты Отчёты Звонки Аналитика

Поиск

Недвижимость / Дома/строения / # 134647 / Передача ключей ЖК Сказка 1 дом

Передача ключей ЖК Сказка 1 дом

Категория:
 Менеджер:
 Клиент:
 Дата подписания акта передачи: От До
 Опции:

№	П	Э	Отделка	Покупатель	Уведомление	Даты готовности объекта		Осмотр	Осмотр с покупателем
						Плановая	Факт		
кв. 1 почт.№1 #134649	1	1	Без отделки	Васютин Вася +7930 (691) 973-4888 Сделка проведена • Рег. 25.09.2021 Задолженность 1 900 000	<input type="button" value="+ Добавить"/>	<input type="button" value="⊕"/>	<input type="button" value="⊕"/>	31.01.2022, 10:50	<input type="button" value="+ Осмотр"/>
кв. 3 почт.№3 #134651	1	1	Без отделки	Пончикова Муся Пусиковна 541 (658) 416-5454 Сделка проведена Задолженность 319 001	<input type="button" value="+ Добавить"/>	<input type="button" value="⊕"/>	<input type="button" value="⊕"/>	<input type="button" value="⊕"/>	<input type="button" value="⚠ Есть замечания"/> <input type="button" value="+"/> <input type="button" value="🗨 Описание замечаний"/>
кв. 5 почт.№5 #134653	1	2	Чистовая	Пончикова Муся Пусиковна 541 (658) 416-5454 Сделка проведена	<input type="button" value="+ Добавить"/>	<input type="button" value="⊕"/>	<input type="button" value="⊕"/>	<input type="button" value="⊕"/>	<input type="button" value="✓ Нет замечаний"/> <input type="button" value="+"/> <input type="button" value="🗨 итоговый без замечаний"/>

Шахматка
 Состав
 Редактирование
 Фото и документы (20)
 Прайс
 Финансы
 Заметки (0)
 Постажные планировки
 План дома
 Планировки объектов
 Смена контакта
 Смена менеджера
 Смена компании
 Клонировать
 Паспорт объекта
 Счета дома

Рисунок 9 – Интерфейс MacroERP