

Инженерная школа информационных технологий и робототехники
 Направление подготовки: 09.03.04 «Программная инженерия»
 Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка компоненты палитр для отображения геофизических данных для растровых изображений

УДК 004.932.4:004.925.5:550.3

Студент

Группа	ФИО	Подпись	Дата
8K71	Грабельников Роман Михайлович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Кузнецов Дмитрий Юрьевич	К.Т.Н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Маланина Вероника Анатольевна	К.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Черемискина Мария Сергеевна	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Чердынцев Евгений Сергеевич	К.Т.Н.		

Планируемые результаты обучения по ООП

Код результатов	Результат обучения (выпускник должен быть готов)
P1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
P6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
P7	Универсальные компетенции
P8	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P9	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P10	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
P11	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.04 «Программная инженерия»
Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8К71	Грабельникову Роману Михайловичу

Тема работы:

Разработка компоненты палитр для отображения геофизических данных для растровых изображений

Утверждена приказом директора (дата, номер)

№32-2/с от 01.02.2021

Срок сдачи студентом выполненной работы:

11.06.2021

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Объектом исследования является компонента палитр для отображения геофизических данных для растровых изображений, которая будет использоваться в приложениях для обработки и интерпретации данных вертикального сейсмического профиля.
Перечень подлежащих исследованию, проектированию и разработке вопросов	<ol style="list-style-type: none">1. Анализ предметной области2. Проектирование компоненты3. Разработка компоненты4. Тестирование компоненты5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

	6. Социальная ответственность
Перечень графического материала	1. Диаграмма Fishbone 2. Диаграммы UML 3. Рисунки, демонстрирующие интерфейс разработанной компоненты 4. Диаграмма Ганта
Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Маланина Вероника Анатольевна
Социальная ответственность	Черемискина Мария Сергеевна

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	28.02.2021
---	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Кузнецов Дмитрий Юрьевич	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К71	Грабельников Роман Михайлович		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.04 «Программная инженерия»
Уровень образования бакалавриат
Отделение информационных технологий
Период выполнения осенний / весенний семестр 2020/2021 учебного года

Форма представления работы:

Бакалаврская работы

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	11.06.2021
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
01.02.2021	Раздел 1. Анализ предметной области	10
01.03.2021	Раздел 2. Проектирование компоненты	25
29.03.2021	Раздел 3. Разработка компоненты	25
26.04.2021	Раздел 4. Тестирование компоненты	20
13.05.2021	Раздел 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
16.05.2021	Раздел 6. Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Кузнецов Дмитрий Юрьевич	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Чердынцев Евгений Сергеевич	к.т.н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8K71	Грабельников Роман Михайлович

Инженерная школа	Инженерная школа информационных технологий и робототехники	Отделение	Отделение информационных технологий
Уровень образования	Бакалавриат	Направление / специальность	09.03.04 «Программная инженерия»

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов работы: материально-технических, энергетических, финансовых, информационных и человеческих	Бюджет проекта – не более 320 тыс. руб., в т.ч. затраты по оплате труда – не более 206 560 руб.
2. Нормы и нормативы расходования ресурсов	Значение показателя интегральной ресурсоэффективности разработки – не менее 4 баллов из 5. Интегральный финансовый показатель разработки – не более 1,00. Интегральный показатель эффективности – не менее 4,4.
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Коэффициент отчислений на уплату во внебюджетные фонды – 30%. Коэффициент накладных расходов – 16%.

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения работы с позиции ресурсоэффективности и ресурсосбережения	Анализ конкурентных технических решений.
2. Планирование и формирование бюджета работы	Формирование плана и графика разработки: – определение структуры работ; – определение трудоемкости работ; – разработка плана-графика. Формирование бюджета затрат на работу: – заработная плата; – отчисления во внебюджетные фонды; – амортизационные расходы; – накладные расходы.

3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности работы	Выбор наиболее эффективного варианта исполнения разработки с позиции финансовой и ресурсной эффективности.
---	--

Перечень графического материала (с точным указанием обязательных чертежей):

<ol style="list-style-type: none"> 1. Оценочная карта конкурентных технических решений 2. Матрица SWOT 3. План-график проведения и бюджет работ 4. Оценка ресурсной и финансовой эффективности работ
--

Дата выдачи задания для раздела по линейному графику	28.02.2021
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Маланина Вероника Анатольевна	к.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К71	Грабельников Роман Михайлович		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

Группа	ФИО
8К71	Грабельников Роман Михайлович

Школа	Инженерная школа информационных технологий и робототехники	Отделение	Отделение информационных технологий
Уровень образования	Бакалавриат	Направление / специальность	09.03.04 «Программная инженерия»

Тема ВКР:

Разработка компоненты палитр для отображения геофизических данных для растровых изображений

Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объект исследования: компонента палитр для отображения геофизических данных для растровых изображений. Область применения: информационные системы и комплексы обработки и интерпретации геофизических данных.
--	--

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Правовые и организационные вопросы обеспечения безопасности: <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	Трудовой кодекс Российской Федерации. ГОСТ 22269-76 Рабочее место оператора. Взаимное расположение элементов рабочего места. ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования.
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	Опасные и вредные факторы: - отсутствие или недостаток необходимого освещения; - аномальные микроклиматические параметры воздушной среды; - повышенный уровень и другие неблагоприятные характеристики шума; - повышенное значение напряжения в электрической цепи; - короткое замыкание.
3. Экологическая безопасность:	Атмосфера: ртутное загрязнение. Гидросфера: утилизация бытовых отходов.

	Литосфера: утилизация компьютеров, смартфонов, другой оргтехники и бытовых отходов.
4. Безопасность в чрезвычайных ситуациях:	Возможные ЧС: пожары, наводнения, землетрясения, удары молнией, взрывы, террористические акты. Наиболее типичная ЧС: пожар.

Дата выдачи задания для раздела по линейному графику	28.02.2021
--	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Черемискина Мария Сергеевна	-		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К71	Грабельников Роман Михайлович		

РЕФЕРАТ

Выпускная квалификационная работа содержит 81 страницу, 17 рисунков, 19 таблиц, 1 приложение и 22 источника.

Ключевые слова: компонент, палитра, линия уровня, веб-приложение, проектирование, разработка.

Цель работы: разработка компоненты редактирования палитр для веб-приложения по обработке и интерпретации данных вертикального сейсмического профиля.

В первой главе представлено описание предметной области, анализ проблемы и требования к компоненте.

Вторая глава описывает процесс проектирования компоненты редактирования палитр.

В третьей главе описывается программная реализация компоненты редактирования палитр.

В четвертой главе описан процесс тестирования компоненты редактирования палитр.

Пятая глава представляет собой выполненное задание по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение», в котором отображены потенциал, планирование и эффективность решения.

Шестая глава представляет собой выполненное задание по разделу «Социальная ответственность», в котором рассмотрены организационные, правовые, производственные и экологические аспекты безопасности, а также безопасность в чрезвычайных ситуациях.

ОГЛАВЛЕНИЕ

СПИСОК ТЕРМИНОВ И СОКРАЩЕНИЙ.....	14
ВВЕДЕНИЕ.....	16
ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	18
1.1 Общие сведения	18
1.2 Постановка проблемы, цели и задач	20
1.3 Требования к компоненте	21
ГЛАВА 2. ПРОЕКТИРОВАНИЕ КОМПОНЕНТЫ.....	24
2.1 Диаграмма вариантов использования	24
2.2 Выбор средств разработки	27
2.2.1 Vue.js	27
2.2.2 TypeScript.....	27
2.3 Диаграмма классов.....	28
2.4 Диаграмма деятельности.....	29
2.5 Диаграмма компонентов	31
ГЛАВА 3. РАЗРАБОТКА КОМПОНЕНТЫ.....	32
3.1 Структура проекта	32
3.2 Разработка на Vue.js.....	33
3.3 Менеджер состояния Vuex	34
3.4 Библиотека компонентов пользовательского интерфейса PrimeVue	35
3.5 Реализация компонентов приложения	36
3.5.1 Компонент домашней страницы	36
3.5.2 Диалоговое окно линий уровня.....	37
3.5.3 Диалоговое окно масштабирования.....	38

3.5.4 Диалоговое окно редактирования линии.....	40
3.5.5 Диалоговое окно редактирования линий.....	40
3.5.6 Диалоговое окно редактирования цветовой карты	41
ГЛАВА 4. ТЕСТИРОВАНИЕ КОМПОНЕНТЫ	42
4.1 Составление тест-кейсов.....	42
4.2 Написание модульных тестов.....	43
4.3 Проведение тестирования	44
ГЛАВА 5. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ.....	47
5.1 Введение.....	47
5.2 Оценка коммерческого потенциала и перспективности проведения работы с позиции ресурсоэффективности и ресурсосбережения	47
5.2.1 Потенциальные потребители результата работы	47
5.2.2 Анализ конкурентных технических решений.....	48
5.2.3 SWOT-анализ	49
5.3 Планирование работ	51
5.3.1 Структура работ	51
5.3.2 Определение продолжительности этапов работ.....	51
5.4 Бюджет работы.....	55
5.4.1 Расчет основной заработной платы исполнителей.....	55
5.4.2 Расчет отчислений во внебюджетные фонды	56
5.4.3 Расчет амортизационных расходов	57
5.4.4 Расчет накладных расходов	58
5.4.5 Формирование бюджета затрат проекта.....	58

5.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности проекта.....	58
5.6 Выводы по разделу	61
ГЛАВА 6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ	62
6.1 Введение.....	62
6.2 Правовые и организационные вопросы обеспечения безопасности	62
6.3 Производственная безопасность	65
6.4 Экологическая безопасность.....	70
6.5 Безопасность в чрезвычайных ситуациях	71
6.6 Выводы по разделу	72
ЗАКЛЮЧЕНИЕ	73
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	74
ПРИЛОЖЕНИЕ А	77

СПИСОК ТЕРМИНОВ И СОКРАЩЕНИЙ

Веб-приложение – приложение, которое выполняется в браузере пользователя.

Десктопное (настольное) приложение – приложение, которое выполняется на компьютере.

Кроссплатформенность – способность приложения работать на разных аппаратных платформах и операционных системах.

Вертикальное сейсмическое профилирование (ВСП) – один из видов сейсморазведки, при использовании которого источник (приемник) размещается на поверхности, а приемник (источник) – в пробуренной скважине.

Палитра – элемент программного обеспечения (ПО) для работы с геофизическими данными, хранящий параметры отображения линий уровня и закраски между ними.

Линия уровня – одно из условных обозначений на карте или графике. Представляет собой линию, в каждой точке которой отображаемая величина имеет одинаковые значения.

Цветовая карта – компонент ПО, позволяющий редактировать параметры заливки цветового пространства.

UML – язык графического описания, используемый в области разработки ПО для моделирования бизнес-процессов и системного проектирования [1].

Фреймворк – программная платформа, облегчающая разработку ПО. Фреймворк задает каркас приложения и накладывает ограничения, позволяющие вести разработку более структурированно.

JavaScript – мультипарадигменный язык программирования, используемый в основном для разработки веб-приложений. JavaScript

поддерживает объектно-ориентированный, императивный и функциональный стили программирования.

TypeScript – язык программирования, расширяющий возможности JavaScript. Отличается от JavaScript возможностью строгого задания типов и наличием дополнительных функций.

Vue.js – JavaScript-фреймворк для создания веб-приложений.

Vuex – библиотека для Vue.js, предназначенная для управления состоянием приложения.

Тест-кейс – описание способа проверки работы ПО, последовательность действий, направленная на проверку какой-либо функции приложения.

ВВЕДЕНИЕ

Средства отображения геофизических данных в виде растровых изображений являются одними из важнейших инструментов в работе геофизиков. Визуализация геофизических данных необходима для их лучшего восприятия и более удобного анализа. Неотъемлемой частью подобных средств являются компоненты, позволяющие редактировать палитры – параметры отображения линий уровня и интервальной закрашки между ними.

Немаловажным аспектом в определении палитры является удобство работы с ней. Если пользователь подходит к определению палитры творчески, возникает необходимость редактирования многих параметров палитры: набора цветов, границ переходов от одного цвета к другому, масштабирования диапазона значений. В то же время должна оставаться возможность быстрого определения палитры – выбора одного из готовых вариантов.

На данный момент ведётся разработка веб-приложения для обработки и интерпретации данных вертикального сейсмического профиля. Мотивацией разработки собственного приложения послужили сложность и неудобство использования сторонних приложений для работы с геофизическими данными. Также важной причиной является то, что в сторонних приложениях отсутствует ряд алгоритмов обработки исходных данных, необходимых для преобразования данных к удобному для работы виду. Таким образом, целью данной работы является упрощение и повышение удобства работы с геофизическими данными. Основной задачей в рамках данной работы является реализация компоненты редактирования палитр для разрабатываемого приложения.

Для достижения цели необходимо выполнить следующие задачи:

- Анализ предметной области, аналогов разрабатываемого приложения и выявление лучших вариантов реализации компоненты редактирования палитр;
- Проектирование компоненты, разработка UML-диаграмм, описывающих ее функции;
- Разработка компоненты;
- Тестирование компоненты.

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Общие сведения

Сфера геофизических исследований является довольно узконаправленной. Поэтому на данный момент существует мало приложений, облегчающих работу ученых и инженеров этой области. Большинство программных продуктов представляют собой десктопные приложения, что заставляет пользователей использовать для работы только тот компьютер, на котором установлена соответствующая программа. Также десктопные решения накладывают ограничения на ПК, точнее на его операционную систему, так как большинство настольных программ не являются кроссплатформенными. Несомненно, существуют и веб-приложения для работы с геофизическими данными, но их функционал на сегодняшний день довольно мал.

Одним из аналогов рассматриваемого приложения является Golden Software Surfer [3]. Из преимуществ программы можно отметить широкий функционал – Surfer действительно является хорошим решением для выполнения самых разных задач. Недостатками являются отсутствие кроссплатформенности – программа доступна только для операционных систем семейства Windows – и высокий порог вхождения – новичку будет сложно разобраться с программой. Интерфейс программы Golden Software Surfer представлен на рисунке 1.

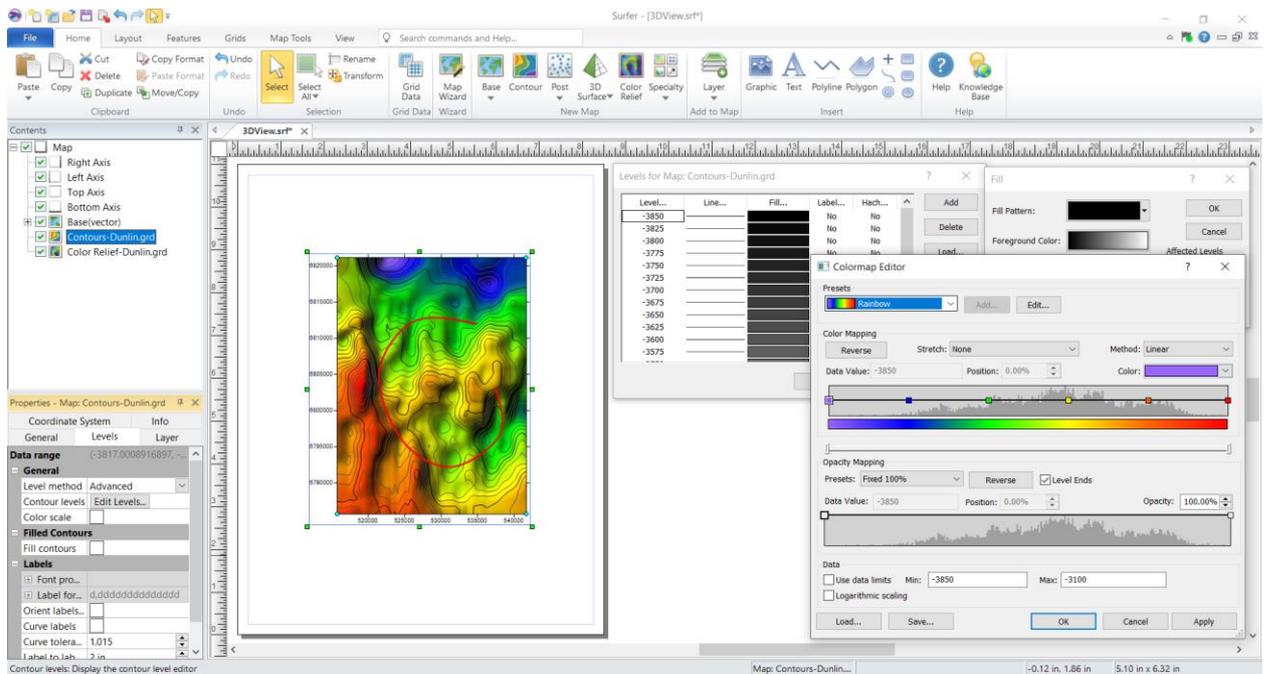


Рисунок 1 – Интерфейс программы Golden Software Surfer

В настоящее время можно заметить тенденцию использования веб-приложений вместо настольных программ. Главным преимуществом такого вида приложений является возможность доступа к приложению с любого устройства и из любого места, где есть доступ к интернету. Это обусловлено тем, что пользователю не нужно устанавливать на свой компьютер тяжеловесное ПО, необходим лишь браузер и подключение устройства к интернету. Также веб-приложения не требовательны к ресурсам, что позволяет пользоваться ими практически на любом устройстве. Исходя из всего вышеперечисленного было принято решение реализовать компоненту редактирования палитр в виде веб-приложения.

Разрабатываемая компонента представляет собой компоненту приложения для анализа, интерпретации и визуализации геофизических данных. Компонента позволяет редактировать палитры – параметры отображения линий уровня и интервальной закрашки между линиями при отображении геофизических данных в виде растровых изображений. Основными функциями компоненты являются:

- Добавление и удаление линий уровня;
- Редактирование значений линий уровня;

- Редактирование линий и закраски между линиями;
- Масштабирование диапазона значений линий уровня;
- Установка цветов линий и цветов закраски между линиями с помощью цветовой карты.

1.2 Постановка проблемы, цели и задач

Для выявления и более полного понимания проблемы необходимо проанализировать причины возникновения проблемы. Подходящим инструментом для этого является диаграмма «Fishbone», представленная на рисунке 2.



Рисунок 2 – Диаграмма «Fishbone»

Исходя из указанных на диаграмме причин проблемой является сложный и неудобный процесс работы с геофизическими данными. Главной причиной проблемы является использование стороннего ПО для анализа и визуализации геофизических данных. Во-первых, сторонние приложения не распространяются свободно, то есть не являются бесплатными. Во-вторых, у стороннего ПО может быть сложный и непонятный интерфейс, с которым будет тяжело разобраться пользователю, впервые использующему это ПО. В-

третьих, в сторонних программах могут отсутствовать функции, необходимые для решения специфичной задачи.

Таким образом, целью работы является повышение качества и удобства обработки геофизических данных. Для достижения поставленной цели необходимо разработать веб-приложение для анализа и отображения этих данных. В данной работе рассматривается компонента такого приложения, представляющая собой редактор палитр. Следовательно, главной задачей работы будет создание компоненты веб-приложения, позволяющей создавать новые палитры и редактировать уже существующие с возможностью экспортировать палитры в файлы. В эту задачу входит:

- Проектирование компоненты палитр;
- Разработка компоненты палитр;
- Тестирование компоненты палитр.

1.3 Требования к компоненте

Для более полного и однозначного понимания возможностей разрабатываемой компоненты необходимо составить список функциональных требований:

1. В приложении должна быть возможность загрузить файл палитры для ее редактирования.
2. В приложении должна быть возможность создать новую палитру.
3. В приложении должна быть возможность добавления линии уровня.
4. В приложении должна быть возможность удаления линий уровня.
5. В приложении должна быть возможность редактирования значения линии уровня.
6. В приложении должна быть возможность редактирования линии:
 - 6.1. Выбрать стиль линии;
 - 6.2. Выбрать цвет линии;
 - 6.3. Задать ширину линии.

7. В приложении должна быть возможность редактирования цвета закрашки между линиями.
8. В приложении должна быть возможность масштабирования палитры:
 - 8.1. Задать минимум значений;
 - 8.2. Задать максимум значений;
 - 8.3. Задать интервал значений;
 - 8.4. Выбрать возможность масштабирования линий;
 - 8.5. Выбрать возможность масштабирования цветов закрашки между линиями.
9. В приложении должна быть возможность редактирования всех линий сразу:
 - 9.1. Выбрать стиль линий;
 - 9.2. Выбрать цвет линий;
 - 9.3. Задать цвета линий с помощью диалогового окна редактирования цветовой карты (см. требование 11);
 - 9.4. Задать ширину линий.
10. В приложении должна быть возможность редактирования всех цветов закрашки между линиями сразу:
 - 10.1. Задать цвета закрашки с помощью диалогового окна редактирования цветовой карты (см. требование 11).
11. В диалоговом окне редактирования цветовой карты должны быть следующие возможности:
 - 11.1. Выбор готового варианта цветовой карты из следующих:
 - 11.1.1. Градации серого;
 - 11.1.2. Радуга.
 - 11.2. Редактирование цветовой карты должно осуществляться с помощью слайдера – линии с точками:
 - 11.2.1. Добавление точки на слайдер.
 - 11.2.2. Удаление точки со слайдера.
 - 11.2.3. Возможные действия при выборе точки слайдера:

- 11.2.3.1. Редактирование значения положения точки в условных единицах;
 - 11.2.3.2. Редактирование значения положения точки в процентах;
 - 11.2.3.3. Выбор цвета точки.
- 11.3. Выбор типа цветовой карты из следующих вариантов:
- 11.3.1. Градиентная;
 - 11.3.2. Интервальная.
- 11.4. Инвертирование цветовой карты – зеркальное отображение точек слайдера относительно центра слайдера.
12. В приложении должна быть возможность сохранения палитры в файл.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ КОМПОНЕНТЫ

Для проектирования компоненты использовались диаграммы UML, так как UML является удобным и легким для восприятия языком графического описания. В нем используется большое количество диаграмм, описывающих как структуру приложения, так и его поведение.

2.1 Диаграмма вариантов использования

На диаграмме вариантов использования отображаются действующие лица, варианты использования проектируемого решения и отношения между элементами диаграммы. Эта диаграмма описывает доступные функции системы. Диаграмма вариантов использования для компоненты редактирования палитр представлена на рисунке 3.

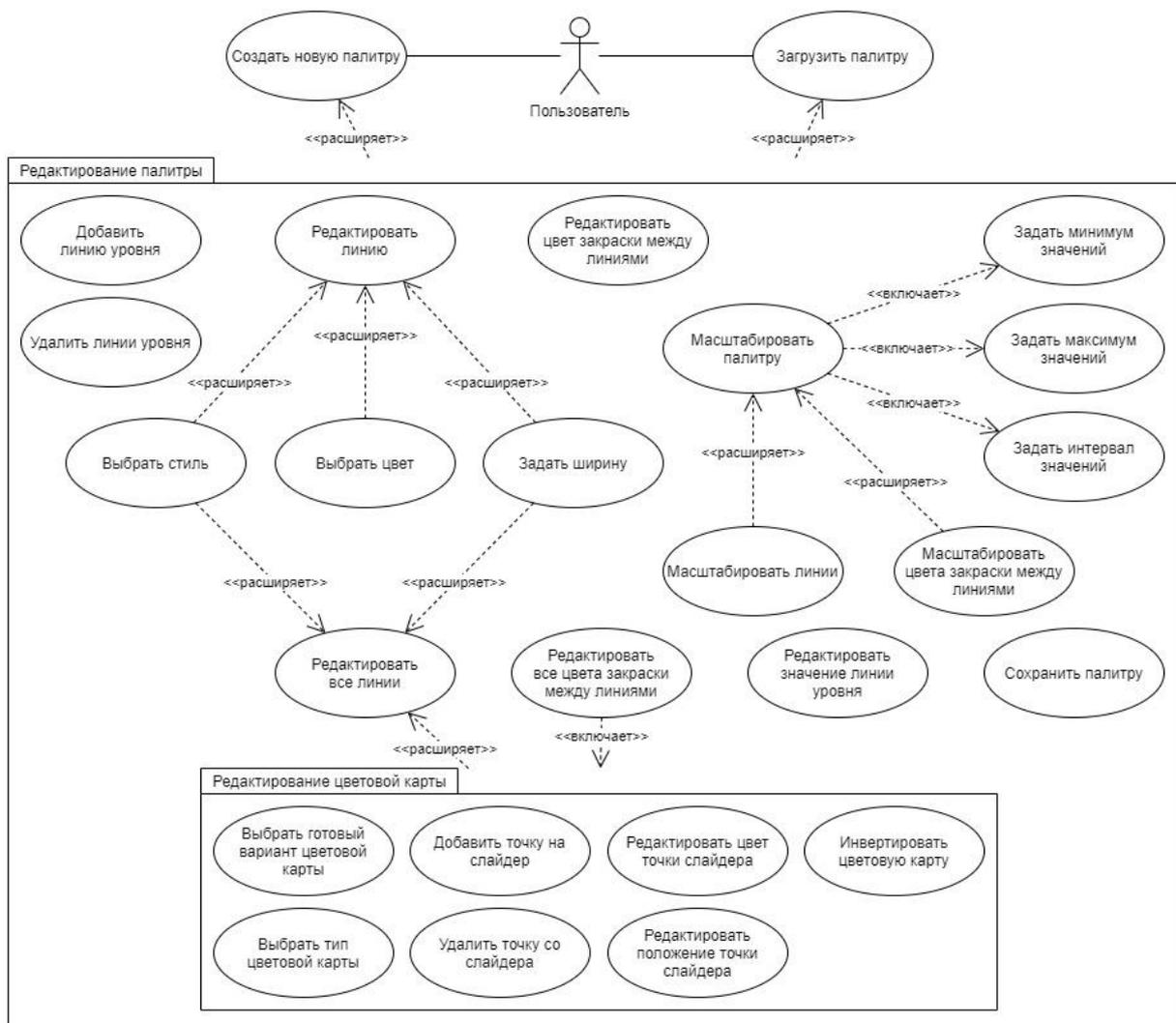


Рисунок 3 – Диаграмма вариантов использования

Для лучшего понимания вариантов использования их подробное описание приведено в таблице 1.

Таблица 1 – Описание вариантов использования

Вариант использования	Описание
Создать новую палитру	Создать новую палитру.
Загрузить палитру	Загрузить файл палитры для редактирования.
Добавить линию уровня	Добавить линию уровня.
Удалить линии уровня	Выбрать и удалить линии уровня.
Редактировать линию	Редактировать параметры линии (стиль, цвет, ширина).
Редактировать все линии	Редактировать параметры всех линий (стиль, цвет, ширина).
Выбрать стиль	Выбрать стиль линии из следующих: сплошная, пунктирная, точечная, невидимая.
Выбрать цвет	Выбрать цвет с помощью палитры выбора цвета.
Задать ширину	Задать ширину линии в пикселях.
Редактировать цвет закрашки между линиями	Редактировать цвет закрашки между выбранной линией уровня и следующей.
Редактировать все цвета закрашки между линиями	Редактировать все цвета закрашки между линиями.
Редактировать значение линии уровня	Редактировать значение линии уровня.
Масштабировать палитру	Преобразовать значения линий уровня в соответствии с заданными границами диапазона значений и интервалом (шагом). При необходимости недостающие линии уровня добавляются, а лишние – удаляются.
Задать минимум значений	Задать минимум значений.
Задать максимум значений	Задать максимум значений.
Задать интервал значений	Задать интервал значений.

Масштабировать линии	Задать для новых линий параметры (стиль, цвет, ширина) старых линий пропорционально первоначальным значениям линий уровня.
Масштабировать цвета закрашки между линиями	Задать цвета закрашки между новыми линиями как у старых линий пропорционально первоначальным значениям линий уровня.
Выбрать готовый вариант цветовой карты	Выбрать готовый вариант цветовой карты из следующих: «Градации серого», «Радуга».
Выбрать тип цветовой карты	Выбрать тип цветовой карты из следующих: «Градиентная», «Интервальная».
Редактирование цветовой карты	Редактирование цветовой карты для задания цветов всех линий или цветов закрашки между линиями. Если выбран тип палитры «Градиентная», то цвет линии определяется как смешение цветов ближайших к линии по значению точек слайдера в соответствующей пропорции. Если выбран тип палитры «Интервальная», то цвет для линии уровня со значением, меньшим значения следующей точки слайдера, задаётся как цвет предыдущей точки слайдера.
Добавить точку на слайдер	Добавить точку на слайдер.
Удалить точку со слайдера	Удалить точку со слайдера.
Редактировать цвет точки слайдера	Выбрать цвет точки слайдера.
Редактировать положение точки слайдера	Изменить положение точки слайдера перетаскиванием, заданием точного значения или заданием значения в процентах.
Инвертировать цветовую карту	Зеркально отразить точки слайдера относительно середины слайдера.

2.2 Выбор средств разработки

В соответствии с функциональными требованиями и вариантами использования проектируемого решения были выбраны следующие средства и технологии, с помощью которых будет разрабатываться приложение.

2.2.1 Vue.js

Фреймворк Vue.js является отличным решением для разработки компоненты редактирования палитр по ряду причин:

- Данный фреймворк обладает всеми функциями современных фронтенд-фреймворков.
- Существует множество библиотек, дополняющих и расширяющих функциональные возможности Vue. Несколько таких библиотек (Vuex, PrimeVue) также будет использоваться при разработке.
- Vue является быстрым и легким. Сам фреймворк занимает всего 20 КБ [4].

Также использование Vue.js обусловлено тем, что в компоненте редактирования палитр нет ресурсозатратных операций, требующих использования серверной части приложения для расчетов.

2.2.2 TypeScript

TypeScript – язык программирования для разработки веб-приложений, расширяющий возможности JavaScript. TypeScript, в отличие от JavaScript, является типизированным языком, что позволяет использовать все преимущества и возможности объектно-ориентированного программирования. В то же время в TypeScript сохраняется гибкость JavaScript. Кроме типизации, TypeScript добавляет в язык некоторые новые возможности. Например, обобщенные структуры (generics) – функции, классы и интерфейсы, способные работать не с одним определенным типом данных, а с разными. Кроме того, TypeScript вводит структуры перечислений (enum), которые также являются полезным и в некоторых случаях незаменимым инструментом программирования [5].

2.3 Диаграмма классов

На диаграмме классов отображаются сущности и отношения между ними. На рисунке 4 представлена диаграмма классов для компоненты редактирования палитр. При создании этой диаграммы учитывалась специфика приложения. Так как для разработки приложения выбран фреймворк Vue.js и языка программирования TypeScript, приложение будет отличаться от классического объектно-ориентированного приложения. Классы и интерфейсы в нем используются только для описания сущностей, а вся бизнес-логика реализуется в компонентах Vue и в модулях Vuex.

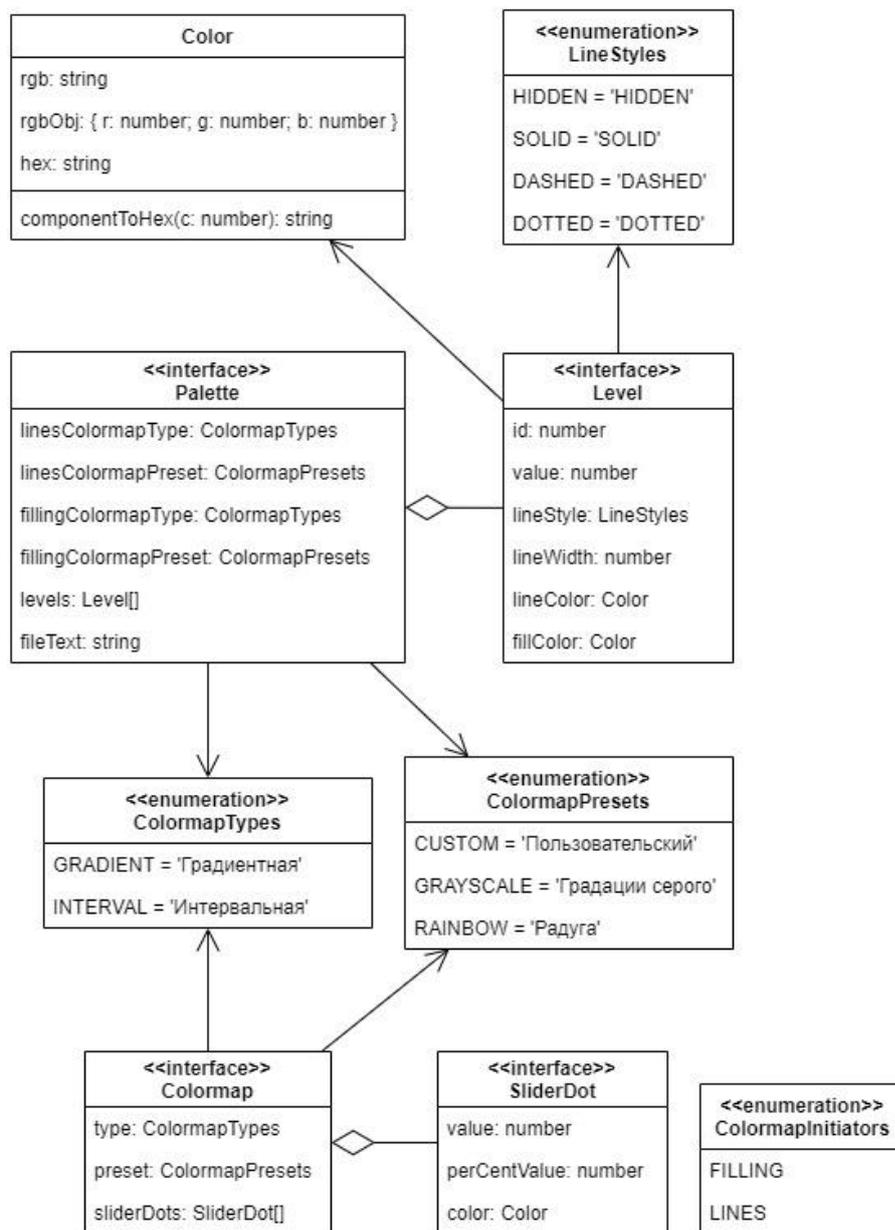


Рисунок 4 – Диаграмма классов

2.4 Диаграмма деятельности

Диаграмма деятельности позволяет представить поведение какой-либо операции, выполняемой в проектируемой программе. Она отображает алгоритмическую и логическую реализацию этой операции.

Таким образом, имеет смысл рассмотреть на диаграмме деятельности один из самых сложных с точки зрения реализации вариантов использования. Для компоненты редактирования палитр им является масштабирование палитры. Этот вариант использования описан в виде диаграммы деятельности на рисунках 5-6.

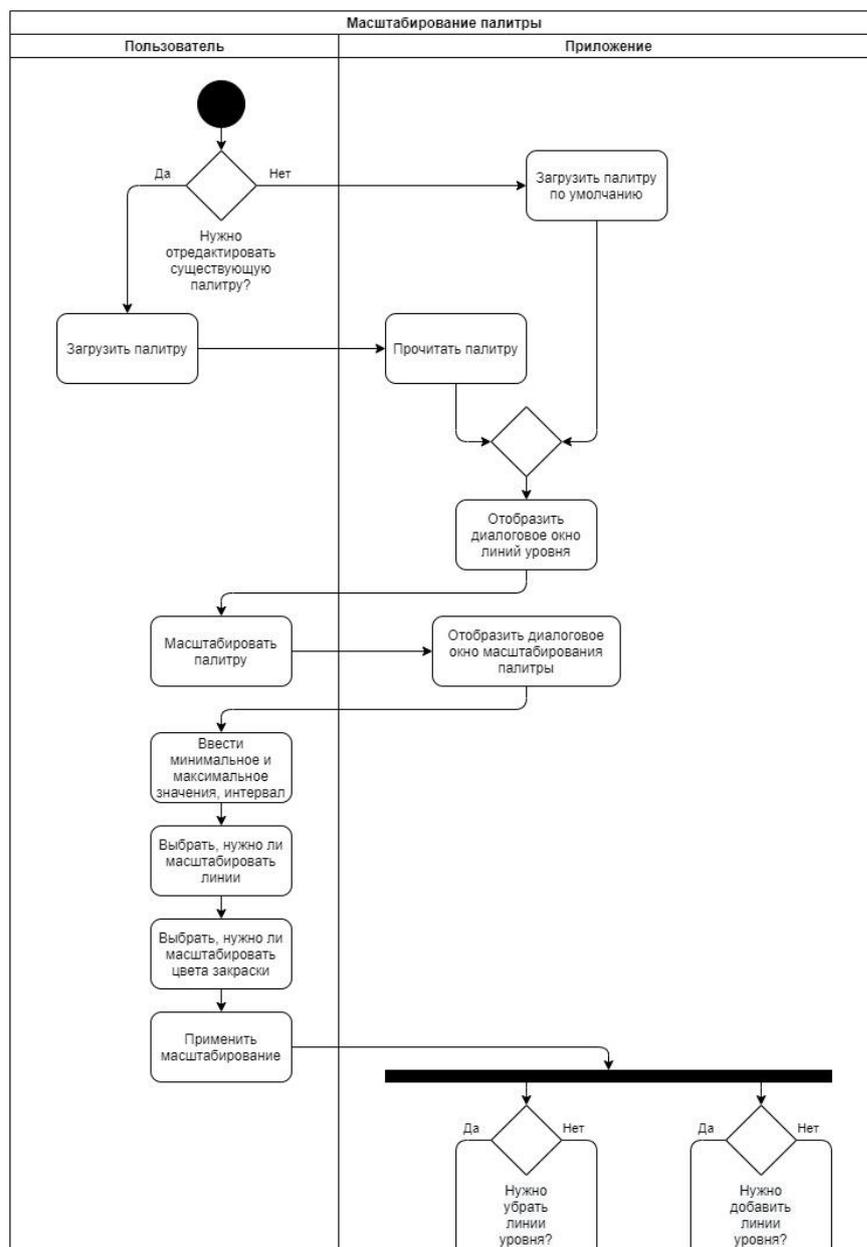


Рисунок 5 – Диаграмма деятельности

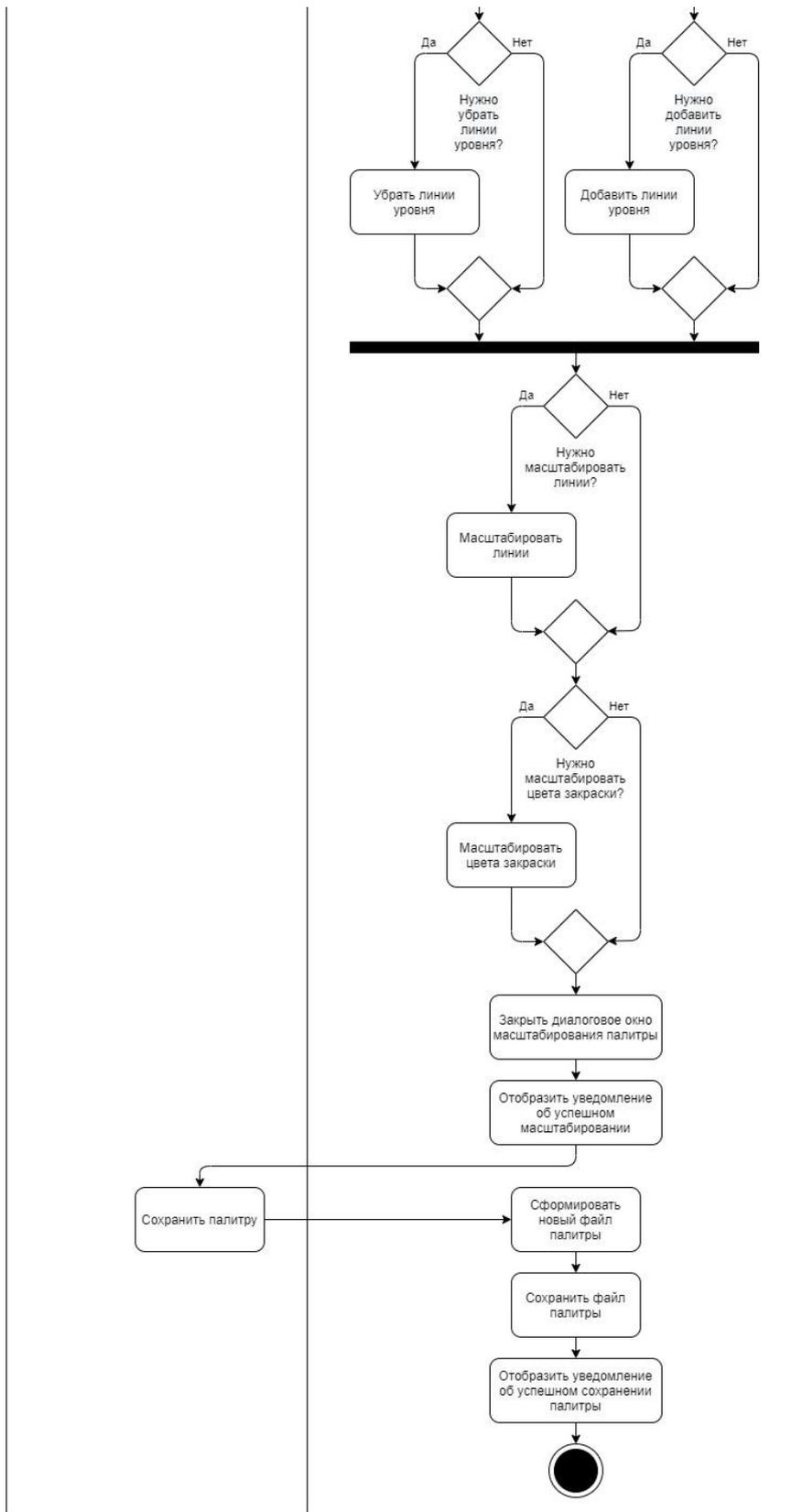


Рисунок 6 – Диаграмма деятельности (продолжение)

2.5 Диаграмма компонентов

Диаграмма компонентов нужна для определения состава компонентов и модулей приложения. Эта диаграмма позволяет перейти от логического представления о проектируемом решении к физическому, так как на ней отображается физическая структура программы: файлы с исходным кодом, исполняемые файлы, используемые библиотеки и отношения между ними. На рисунке 7 представлена диаграмма компонентов для проектируемого приложения.

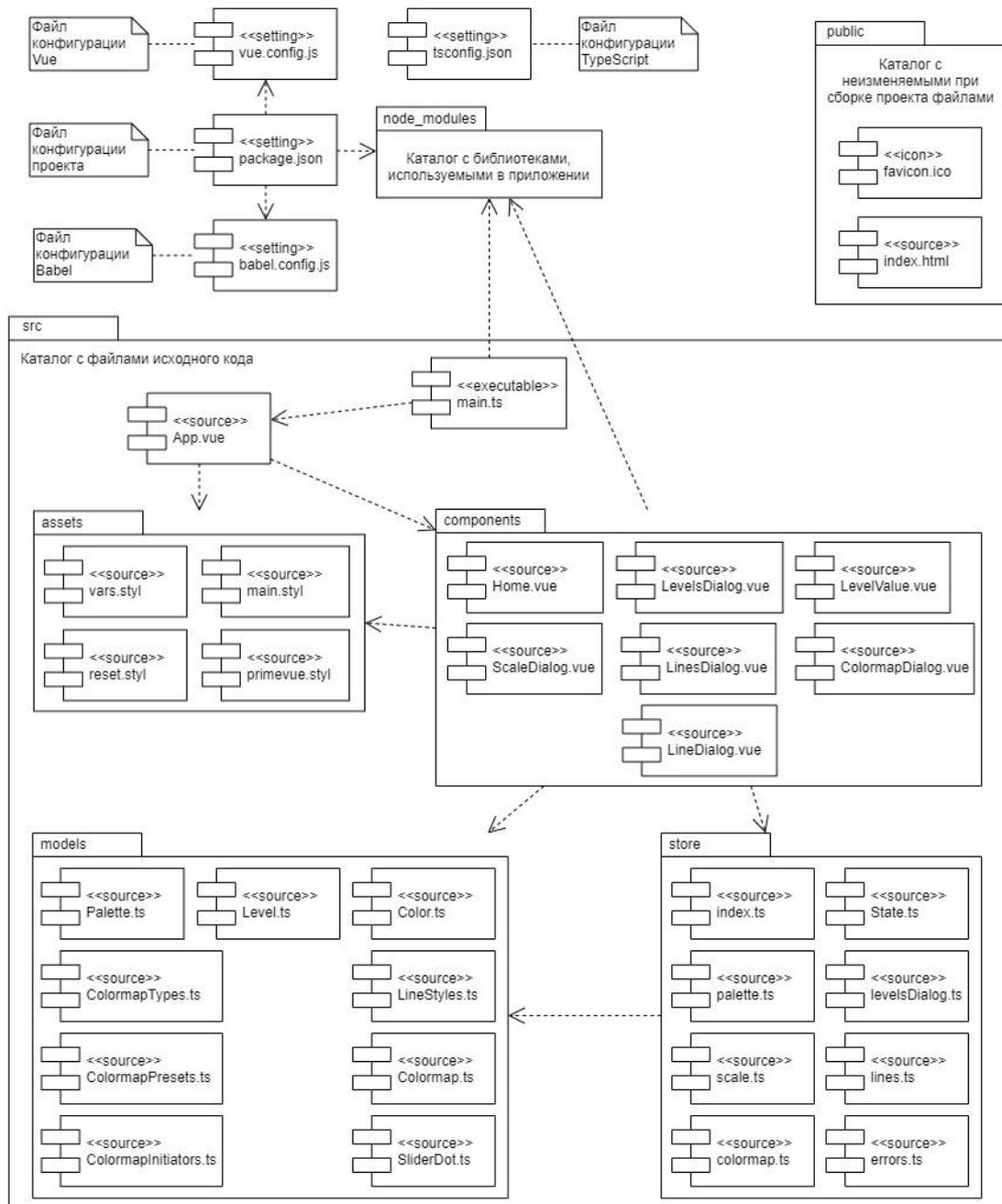


Рисунок 7 – Диаграмма компонентов

ГЛАВА 3. РАЗРАБОТКА КОМПОНЕНТЫ

3.1 Структура проекта

Структура проекта, в котором используется Vue.js, Vuex и TypeScript представлена на рисунке 8.

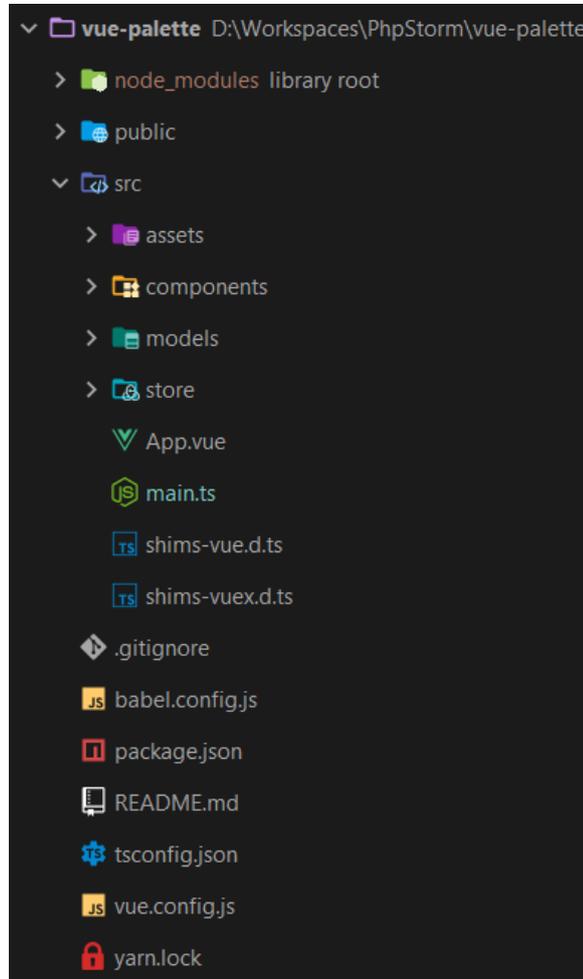


Рисунок 8 – Структура проекта

Описание основных файлов и каталогов проекта приведено в списке:

- node_modules – каталог, в котором хранятся используемые в проекте библиотеки;
- public – в этом каталоге находятся неизменяемые во время сборки проекта файлы – index.html и favicon.ico;
- src – каталог с исходным кодом приложения:
 - assets – в этом каталоге находятся файлы стилей, используемые в нескольких компонентах приложения;

- `components` – каталог с компонентами `Vue`;
 - `model` – каталог, в котором хранятся модели – классы, интерфейсы и перечисления;
 - `store` – каталог файлов `Vuex`, в них реализуется бизнес-логика приложения;
 - `App.vue` – основная компонента приложения `Vue`, в которой подключаются другие компоненты;
 - `main.ts` – исполняемый файл приложения, в нем подключаются библиотеки и происходит инициализация приложения;
 - `shims.vue.d.ts` и `shims.vuex.d.ts` – файлы конфигурации, позволяющие совмещать `TypeScript` с `Vue` и `Vuex`;
- `babel.config.js` – файл конфигурации `Babel`;
 - `package.json` – файл конфигурации проекта, в нем указаны подключенные библиотеки и скрипты, необходимые для разработки и сборки проекта;
 - `tsconfig.json` – файл конфигурации компилятора `TypeScript`;
 - `vue.config.js` – файл конфигурации `Vue`.

3.2 Разработка на `Vue.js`

Во `Vue` реализован способ разработки на основе компонентов. Он заключается в разработке отдельных компонентов, которые могут использоваться повторно в нескольких местах приложения. При этом в каждом компоненте есть три блока: `template`, `script` и `style`. Первый блок предназначен для разметки страницы, второй – для реализации JavaScript- или TypeScript-логики, которая может использоваться в первом блоке, и третий блок – для написания стилей. В листинге 1 приведен пример компонента `Vue`.

Листинг 1 – Компонент `Vue`

```
<template>
  <div class="home">
    <...>
  </div>
</template>
```

```

<script lang="ts">
import {defineComponent} from 'vue'
import LevelsDialog from '@/components/LevelsDialog.vue'

export default defineComponent({

  name: 'Home',

  components: {
    LevelsDialog
  },

  methods: {
    async onSelectFile(event: any) {
      <...>
    }
  }
})
</script>

<style scoped lang="stylus">
.home
  display flex
  align-items center
  flex-direction column
  min-height 100vh
  padding 50px 0
  <...>
</style>

```

Также во Vue присутствует система реактивности, позволяющая динамически изменять отображаемые на веб-странице данные. Для этого в компонентах Vue реализован механизм, предоставляющий разработчику возможность использовать в блоке `template` данные и методы из следующих свойств блока `script`: `data`, `methods` и `computed`. Реактивность Vue делает управление данными простым и интуитивно понятным для разработчика, что положительно влияет на его продуктивность и скорость выполнения проекта.

3.3 Менеджер состояния Vuex

Vuex представляет собой библиотеку для управления состоянием приложения. Vuex позволяет разным компонентам получать доступ к одним данным или вызвать методы, изменяющие одни данные [6].

Во Vuex, так же как и во Vue, используются файлы с определенной структурой. Основные блоки файла Vuex: `state`, `mutations`, `actions` и `getters`. В первом блоке хранятся данные приложения, во втором – методы, изменяющие

данные блока state, в третьем – методы, работающие с данными и методами других файлов Vuex и серверной частью приложения, если она предусмотрена, и в четвертом – методы, позволяющие компонентам Vue получить доступ к данным из блока state. В листинге 2 продемонстрирована структура файла Vuex.

Листинг 2 – Файл Vuex

```
import {State} from '@/store/State'
<...>

export default {

  state: () => ({
    showLineDialog: false,
    editableLevelLine: null,
    isLinesDialog: false
  }),

  mutations: {
    <...>

    HIDE_LINE_DIALOG(state: State) {
      state.isLinesDialog = false
      state.showLineDialog = false
      state.editableLevelLine = null
    }
  },

  actions: {
    <...>

    HIDE_LINE_DIALOG({commit}) {
      commit('HIDE_LINE_DIALOG')
    }
  },

  getters: {
    showLineDialog: (state: State) => state.showLineDialog,
    editableLevelLine: (state: State) => state.editableLevelLine,
    isLinesDialog: (state: State) => state.isLinesDialog
  }
}
```

3.4 Библиотека компонентов пользовательского интерфейса

PrimeVue

Библиотеки компонентов пользовательского интерфейса созданы для упрощения и ускорения разработки. PrimeVue является одной из таких библиотек. Это – мощный, но простой в использовании инструмент. В этой

библиотеке присутствует порядка 90 компонентов, таких как элементы веб-форм, кнопки, иконки, таблицы, списки, диалоговые окна, меню, графики и многие другие. Из всего многообразия представленных компонентов в проекте были использованы следующие: FileUpload, Button, Dialog, DataTable, ColorPicker, InputNumber, Checkbox, Toast, Dropdown, Panel, Divider и RadioButton [7].

3.5 Реализация компонентов приложения

3.5.1 Компонент домашней страницы

При посещении страницы приложения пользователь в первую очередь видит компонент домашней страницы. На нем пользователь может выбрать один из двух вариантов использования: создать новую палитру или загрузить файл существующей палитры и редактировать его. Интерфейс данного компонента представлен на рисунке 9.

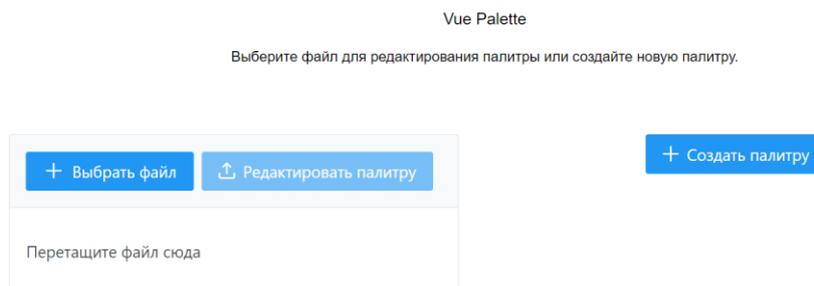


Рисунок 9 – Компонент домашней страницы

В случае редактирования палитры необходимо загрузить файл определенного формата. В первой строке файла должны быть описаны тип и выбранный готовый вариант цветовой карты для цветов линий, во второй строке – то же самое, но для цветов закраски. Далее в файле должны указываться данные о самих линиях уровня. Пример содержания такого файла представлен в листинге 3.

Листинг 3 – Файл палитры

```
lines:type=GRADIENT;preset=CUSTOM
filling:type=INTERVAL;preset=CUSTOM
value lineStyle lineWidth lineColor fillColor
10 solid 2 RGB(0,0,0) RGB(255,255,255)
15 solid 8 RGB(0,0,0) RGB(255,0,0)
20 solid 2 RGB(255,0,0) RGB(0,255,0)
35 solid 2 RGB(0,255,0) RGB(0,0,255)
55 solid 2 RGB(0,0,255) RGB(0,0,0)
60 dotted 2 RGB(0,0,0) RGB(255,255,255)
75 solid 8 RGB(0,0,0) RGB(255,0,0)
85 solid 2 RGB(255,0,0) RGB(0,255,0)
100 solid 2 RGB(0,0,0) RGB(255,255,255)
```

3.5.2 Диалоговое окно линий уровня

При нажатии на кнопку создания или редактирования палитры открывается диалоговое окно линий уровня. Оно представлено на рисунке 10. В нем отображаются данные о линиях уровня: значение, линия и цвет закрашки.

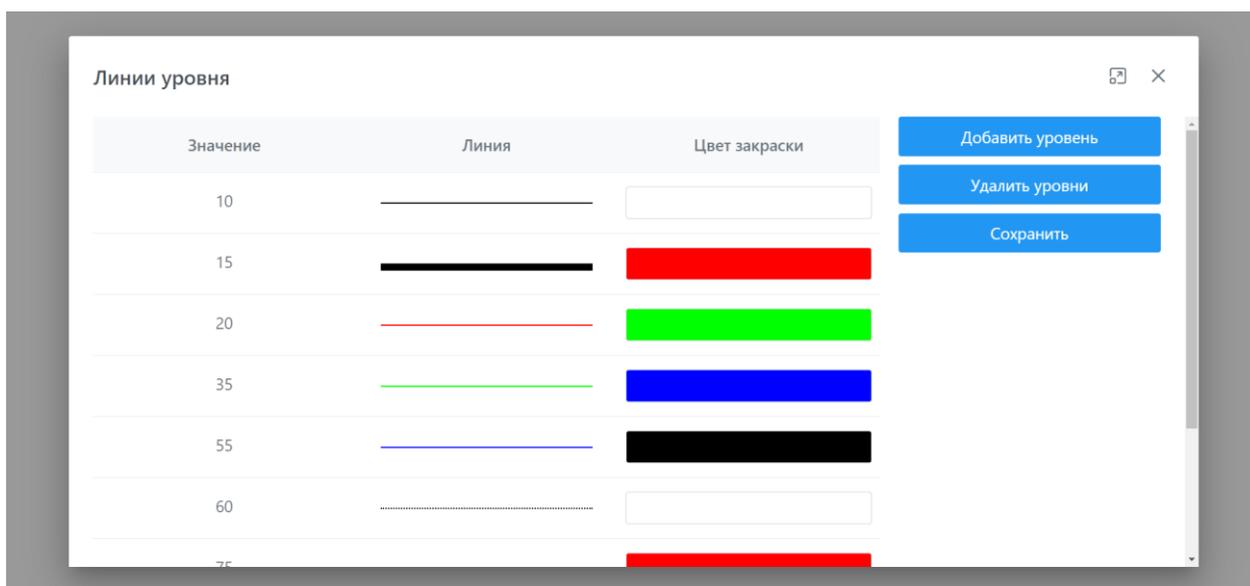


Рисунок 10 – Диалоговое окно линий уровня

Также в этом диалоговом окне реализованы следующие функции:

- Редактирование значения линии уровня. Для этого необходимо нажать на значение определенной линии, после чего можно будет ввести новое значение.

- Редактирование цвета закрашки. Для этого необходимо нажать на цвет закрашки определенной линии, после чего откроется палитра выбора цвета.
- Добавление линии уровня. Для этого необходимо нажать на кнопку «Добавить уровень».
- Удаление линий уровня. Для этого необходимо нажать на кнопку «Удалить уровни», после чего напротив каждого уровня появится поле для выбора этого уровня (рисунок 11). После выбора линий уровня и нажатия кнопки «Подтвердить» выбранные линии будут удалены.

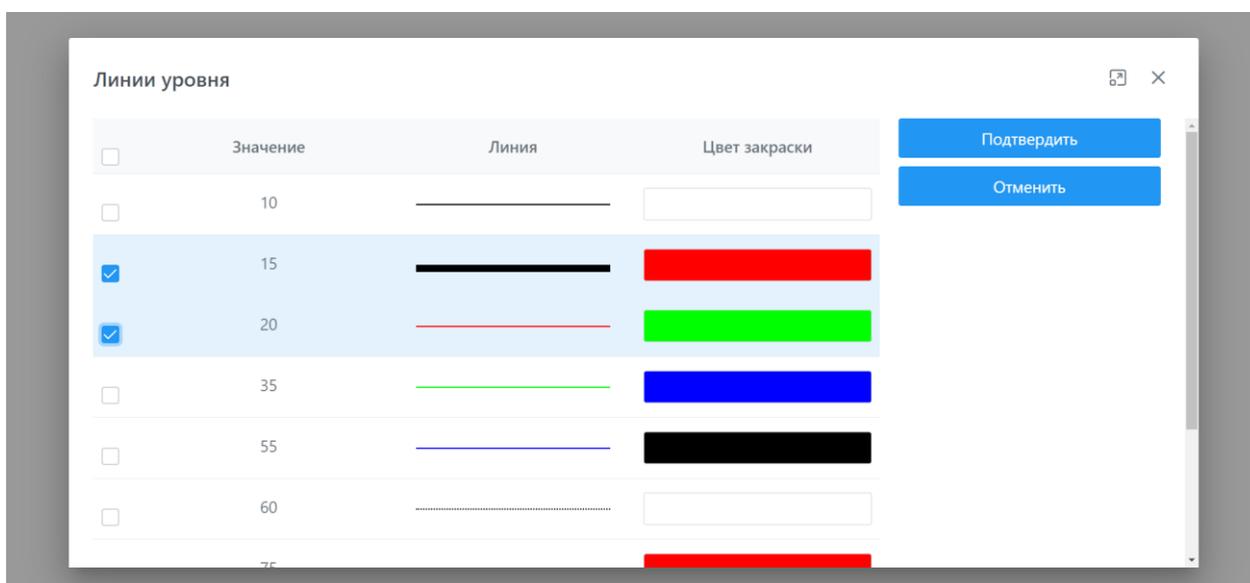


Рисунок 11 – Удаление линий уровня

3.5.3 Диалоговое окно масштабирования

Диалоговое окно масштабирования открывается при нажатии на кнопку «Значение» в строке заголовка таблицы. Для масштабирования линий уровня необходимо ввести граничные значения и интервал и выбрать, нужно ли при масштабировании значений масштабировать линии и цвета закрашки между линиями. Интерфейс этого компонента представлен на рисунке 12.

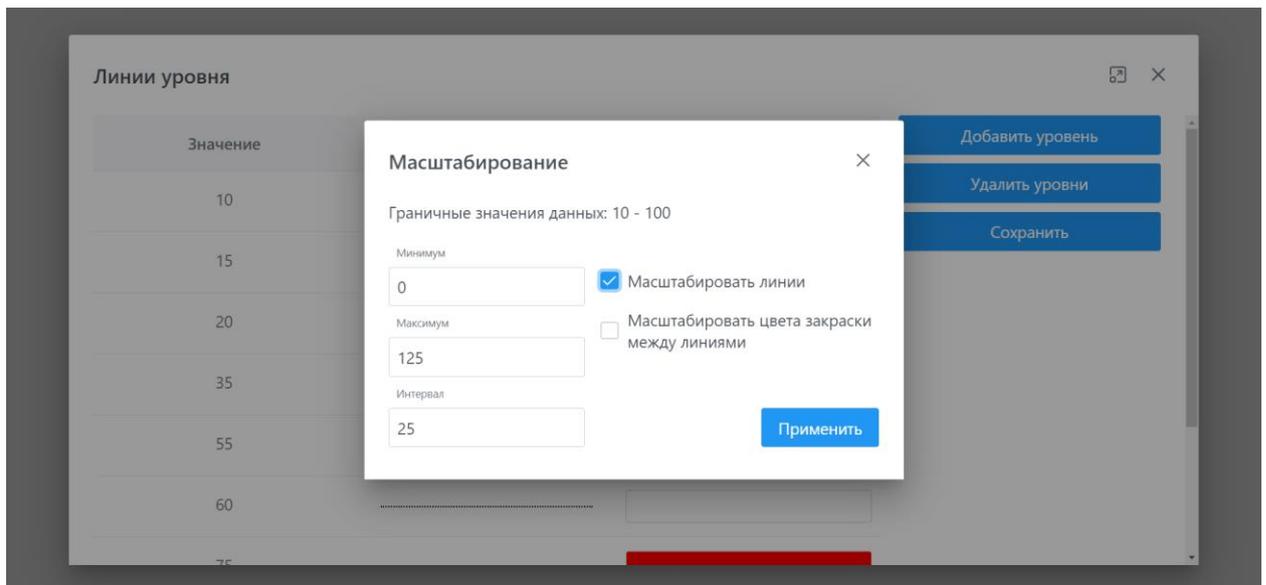


Рисунок 12 – Диалоговое окно масштабирования

Для реализации масштабирования уровней был разработан алгоритм поиска соответствующих уровней и выставления им параметров (приложение А). При масштабировании линий уровня создается новый набор линий уровня, после чего для новых уровней устанавливаются признаки линий и цвета закраски соответствующих старых уровней. Для реализации поиска соответствующих уровней используется модифицированный алгоритм двоичного поиска. Он представлен в листинге 4. Также этот алгоритм используется в компоненте цветовой карты, который рассмотрен в пункте 3.5.6.

Листинг 4 – Алгоритм двоичного поиска

```
export const binarySearch = <T extends { value: number }>(data: T[], target: number,
start: number, end: number): T => {
  if (end < 1) {
    return data[0]
  }

  const middle = Math.floor((start + (end - start) / 2))
  if (target == data[middle].value) {
    return data[middle]
  }

  if (end - 1 === start) {
    return Math.abs(data[start].value - target) > Math.abs(data[end].value - target)
? data[end] : data[start]
  }

  if (target > data[middle].value) {
```

```

return binarySearch(data, target, middle, end)
}

if (target < data[middle].value) {
return binarySearch(data, target, start, middle)
}
}

```

3.5.4 Диалоговое окно редактирования линии

Диалоговое окно редактирования линии открывается при нажатии на линию в колонке с заголовком «Линия». В этом диалоговом окне можно выбрать стиль и цвет линии, задать ее ширину. Интерфейс данного компонента представлен на рисунке 13.

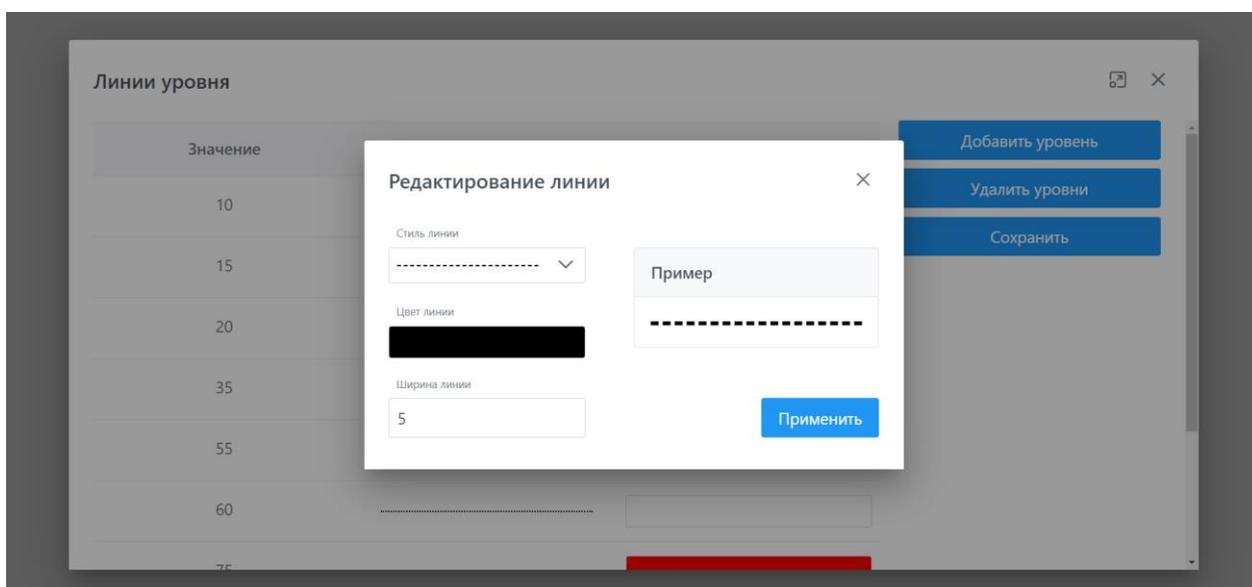


Рисунок 13 – Диалоговое окно редактирования линии

3.5.5 Диалоговое окно редактирования линий

Диалоговое окно редактирования линии открывается при нажатии на кнопку «Линия» в строке заголовка таблицы. Этот компонент очень похож на диалоговое окно редактирования одной линии, но в нем можно выбрать только определенные параметры для задания всем линиям. Также цвет линий можно задать с помощью цветовой карты, компонент которой рассмотрен в пункте 3.5.6. Интерфейс данного диалогового окна представлен на рисунке 14.

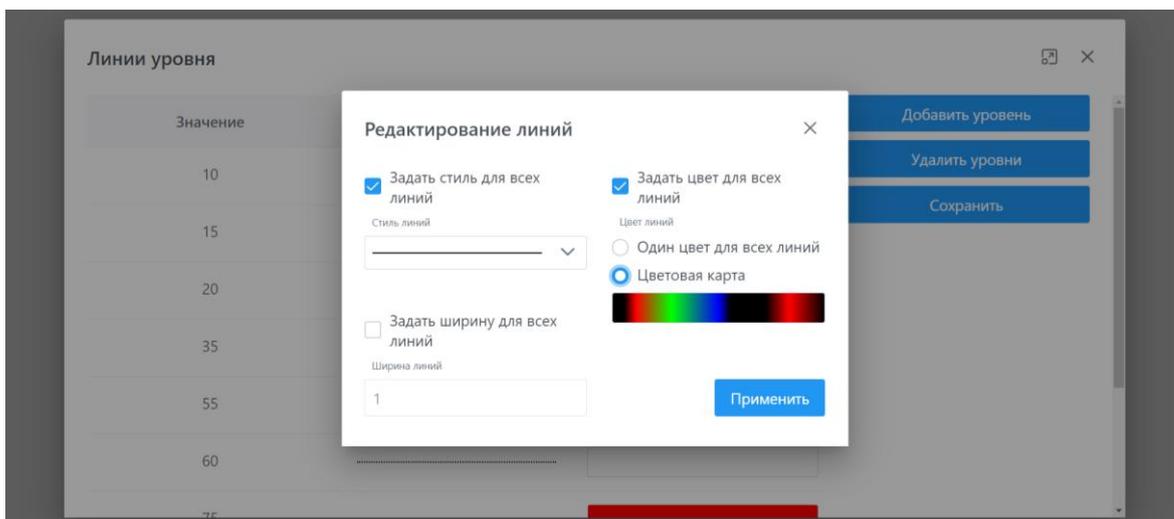


Рисунок 14 – Диалоговое окно редактирования линий

3.5.6 Диалоговое окно редактирования цветовой карты

Диалоговое окно редактирования цветовой карты используется для редактирования цветов закрашки между линиями и для редактирования цветов самих линий. Для редактирования цветов закрашки необходимо нажать на кнопку «Цвет закрашки» в строке заголовка таблицы, а для редактирования цветов линий – на кнопку редактирования цвета в диалоговом окне редактирования линий при выбранном способе задания цветов линий «Цветовая карта». Интерфейс диалогового окна редактирования цветовой карты представлен на рисунке 15.

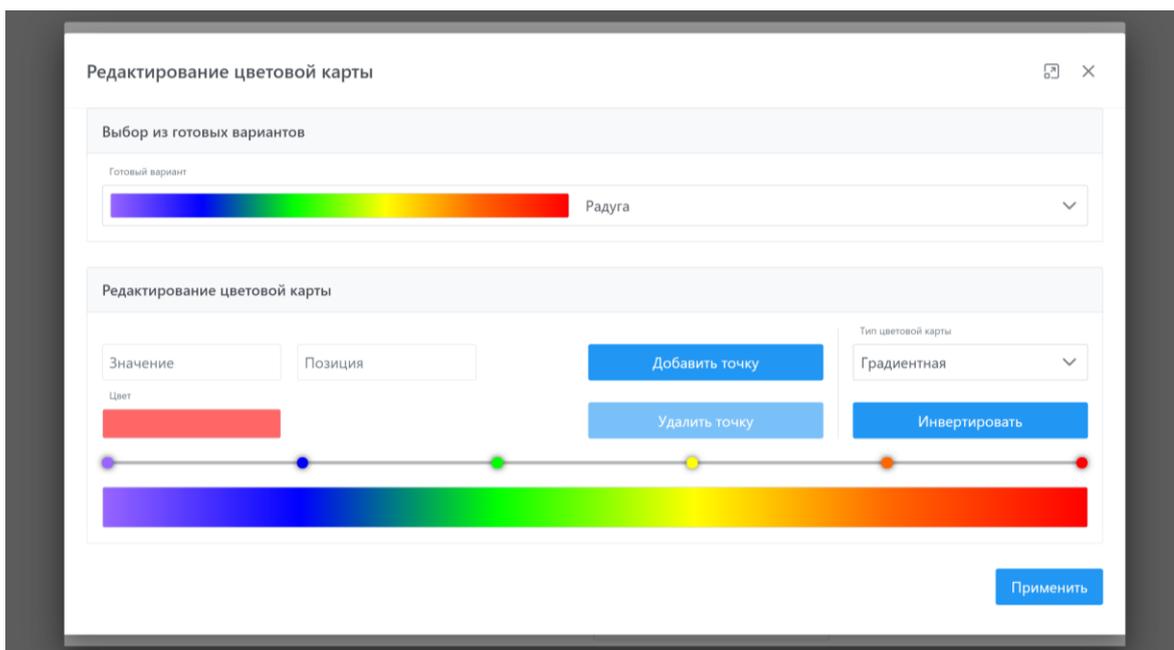


Рисунок 15 – Диалоговое окно редактирования цветовой карты

ГЛАВА 4. ТЕСТИРОВАНИЕ КОМПОНЕНТЫ

Тестирование является одним из основных этапов создания программного обеспечения. Во время тестирования отлавливается и исправляется значительная часть ошибок, возникающих при эксплуатации программного продукта. Во время тестирования проверяется соответствие реального поведения приложения ожидаемому, то есть описанному в требованиях к приложению. Также при наличии в программе сложных методов необходимо реализовать для них модульные тесты, чтобы тщательно проверить работу этих методов на разных наборах данных.

4.1 Составление тест-кейсов

На основе требований к разрабатываемой компоненте, описанных в пункте 1.3, были составлены тест-кейсы. Тест-кейс представляет собой документацию, описывающую последовательность шагов, которые необходимо выполнить в приложении для достижения определенного результата. Каждый тест-кейс описывает одну функцию приложения и содержит следующие атрибуты [8]:

- Идентификатор;
- Название;
- Предусловия;
- Шаги;
- Ожидаемый результат.

Для тестирования компоненты редактирования палитр были составлены тест-кейсы по ключевым функциональным требованиям, указанным в пункте 1.3. В качестве примера в таблице 2 приведен тест-кейс для проверки требования 6 (в приложении должна быть возможность редактирования линии).

Таблица 2 – Пример тест-кейса

Идентификатор	6
Название	Редактирование линии
Результат	
Действие	Ожидаемый результат
Предусловия	
Открыть домашнюю страницу приложения, нажать кнопку «Создать палитру»	Открыто диалоговое окно линий уровня
Шаги	
<ol style="list-style-type: none"> 1. Нажать на линию в колонке с заголовком «Линия» 2. Выбрать стиль линии, отличный от уже выбранного 3. Выбрать цвет линии, отличный от уже выбранного 4. Ввести ширину линии, отличную от установленной 5. Нажать кнопку «Применить» 	Отображается диалоговое окно редактирования линии. В диалоговом окне можно выбрать стиль (невидимая, сплошная, пунктирная, точечная), цвет (с помощью палитры выбора цвета), ввести ширину линии (больше 0 и меньше 50). При изменении параметров меняется отображение примера линии. При нажатии кнопки «Применить» диалоговое окно закрывается, в диалоговом окне линий уровня меняется отображение отредактированной линии в соответствии с выбранными параметрами.

4.2 Написание модульных тестов

Для написания модульных тестов использовалась JavaScript-библиотека Jest совместно с официальной библиотекой модульного тестирования для Vue.js – Vue Test Utils. Jest ориентирована на простоту, поэтому ее достаточно легко освоить и начать использовать [9].

Для создания теста необходимо создать файл с именем следующего вида: <название теста>.test.js или <название теста>.spec.js. При использовании TypeScript расширение файла меняется с js на ts. Для написания тестов

необходимо использовать функции, которые предоставляет Jest. Основные из представленных в списке:

- describe(name: number | string | Function | FunctionLike, fn: EmptyFunction): void – определяет логический блок для написания тестов;
- test(name: string, fn?: ProvidesCallback, timeout?: number): void – одна из основных и наиболее часто используемых функций, она нужна для написания самого теста;
- expect<T = any>(actual: T): JestMatchers<T> – эта функция необходима для вызова тестируемого метода;
- toBe<E = any>(expected: E): R, toBeCloseTo(expected: number, numDigits?: number): R, toBeGreaterThan(expected: number): R, toEqual<E = any>(expected: E): R и т.д. – функции сопоставления, они сравнивают результат выполнения тестируемого метода с ожидаемым.

4.3 Проведение тестирования

В рамках тестирования были проверены все функциональные требования к разрабатываемой компоненте. В ходе тестирования было выявлено несколько ошибок, в том числе ошибка при задании цветов закрашки при помощи цветовой карты. Результат тестирования этого требования представлен в таблице 3.

Таблица 3 – Результат тестирования требования

Идентификатор	10.1.1
Название	Редактирование цветов закрашки при использовании градиентной цветовой карты
Результат	В случае, если крайняя правая точка слайдера находится не в крайнем правом положении (позиция 100%), цвета закрашки всех линий уровня, чье

	значение больше значения крайней правой точки слайдера, становятся черными вне зависимости от цвета последней точки слайдера.
Действие	Ожидаемый результат
Предусловия	
Открыть домашнюю страницу приложения, нажать кнопку «Создать палитру»	Открыто диалоговое окно линий уровня
Шаги	
<ol style="list-style-type: none"> 1. Нажать на кнопку «Цвет закраски» в строке заголовка таблицы 2. Установить тип цветовой карты «Градиентная» 3. Добавить несколько точек на слайдер путем нажатия кнопки «Добавить точку» 4. Удалить несколько точек со слайдера путем нажатия кнопки «Удалить точку» или клавиши «Esc» при выбранной точке 5. Изменить положение и цвет нескольких точек на слайдере 6. Нажать кнопку «Применить» 	Отображается диалоговое окно редактирования цветовой карты. В диалоговом окне можно выбрать готовый вариант цветовой карты, отредактировать цветовую карту: добавить точку, удалить точку, изменить положение и цвет точки, инвертировать цветовую карту. При изменении параметров меняется отображение градиента, расположенного под слайдером. При нажатии кнопки «Применить» диалоговое окно закрывается, в диалоговом окне линий уровня меняются цвета закраски в соответствии с отредактированной цветовой картой.

Модульные тесты были написаны для сложных методов, которые используются в приложении. Одним из таких методов является метод двоичного поиска. Код написанных тестов для этого метода представлен в листинге 5. Результат выполнения тестов отображен на рисунке 16.

Листинг 5 – Модульные тесты для метода двоичного поиска

```
import {binarySearch} from '@store/scale'

const TEST_ARR = [
```

```

    { value: -50 }, { value: -40 }, { value: -30 }, { value: -20 }, { value: -10 },
    { value: 0 }, { value: 10 }, { value: 20 }, { value: 30 }, { value: 40 }, { value:
50 }
]

const TEST_TARGET_NEG = -17
const TEST_TARGET_ZERO = 0
const TEST_TARGET_POS = 22

describe('Binary search, normal values:', () => {
  test('negative target', () => {
    expect(binarySearch(TEST_ARR, TEST_TARGET_NEG, 0, TEST_ARR.length - 1).value)
      .toBeCloseTo(TEST_TARGET_NEG - 3)
  })
  test('zero target', () => {
    expect(binarySearch(TEST_ARR, TEST_TARGET_ZERO, 0, TEST_ARR.length - 1).value)
      .toBeCloseTo(TEST_TARGET_ZERO)
  })
  test('positive target', () => {
    expect(binarySearch(TEST_ARR, TEST_TARGET_POS, 0, TEST_ARR.length - 1).value)
      .toBeCloseTo(TEST_TARGET_POS - 2)
  })
})

describe('Binary search, small values:', () => {
  const smallArr = TEST_ARR.map(it => ({value: it.value / 1000000}))
  <...>
})

describe('Binary search, large values:', () => {
  const largeArr = TEST_ARR.map(it => ({value: it.value * 1000000}))
  <...>
})

```

```

PASS tests/unit/binarySearch.spec.ts (5.232s)
  Binary search, normal values:
    ✓ negative target (7ms)
    ✓ zero target (1ms)
    ✓ positive target
  Binary search, small values:
    ✓ negative target
    ✓ zero target (1ms)
    ✓ positive target
  Binary search, large values:
    ✓ negative target
    ✓ zero target (1ms)
    ✓ positive target

Test Suites: 1 passed, 1 total
Tests:       9 passed, 9 total
Snapshots:  0 total
Time:        8.002s
Ran all test suites.
Done in 12.09s.

```

Рисунок 16 – Результат выполнения модульных тестов

ГЛАВА 5. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

5.1 Введение

В работе рассматривается компонента программного обеспечения, позволяющая редактировать палитры – параметры отображения линий уровня и интервальной закрашки между линиями при отображении геофизических данных в виде растровых изображений. Проведем комплексное описание и анализ финансово-экономических аспектов разработанной компоненты.

5.2 Оценка коммерческого потенциала и перспективности проведения работы с позиции ресурсоэффективности и ресурсосбережения

5.2.1 Потенциальные потребители результата работы

Для анализа потребителей результатов работы необходимо рассмотреть целевой рынок и провести его сегментирование.

Целевой рынок – сегменты рынка, на котором будет продаваться в будущем разработка. Сегмент рынка – это особым образом выделенная часть рынка, группы потребителей, обладающих определенными общими признаками. Сегментирование – это разделение покупателей на однородные группы, для каждой из которых может потребоваться определенный товар (услуга) [10].

Целевым рынком для компоненты палитр будут компании, занимающиеся анализом и моделированием поверхностей, визуализацией ландшафта, обработкой и интерпретацией данных вертикального сейсмического профилирования. Таким образом, продукт можно продавать как в Томске, так и по всей Томской области и России.

Для сегментирования рынка целесообразно использовать два наиболее значимых критерия: размер компании-заказчика и тип программного обеспечения.

В таблице 4 представлено сегментирование рынка ПО с компонентами редактирования палитр. Занятые ниши обозначены символом «+». Занимающие эти ниши приложения рассмотрены в подразделе 2.2.

Таблица 4 – Сегментирование рынка

Критерии		Размер компании		
		Крупные	Средние	Мелкие
Тип ПО	Десктопное	+	+	+
	Веб-приложение			

Таким образом, есть три свободные ниши. Разрабатываемая компонента является веб-приложением и ориентирована на крупные, средние и мелкие компании.

5.2.2 Анализ конкурентных технических решений

Компонента редактирования палитр не является готовым решением, которое можно встроить в имеющийся программный продукт. Поэтому именно среди компонент нет конкуренции, но подобные компоненты присутствуют во многих программах для отображения ландшафта и других поверхностей. Следовательно, необходимо рассмотреть конкурентов приложений для отображения поверхностей, в состав которых входят компоненты палитр.

Анализ конкурентов представлен в виде оценочной карты, которая приведена в таблице 5. В таблице использованы следующие обозначения:

- ф – фактический разрабатываемый продукт;
- к1 – конкурент 1 – Golden Software Surfer [2];
- к2 – конкурент 2 – QGIS [11].

Таблица 5 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Бф	Бк1	Бк2	Кф	Кк1	Кк2
Технические критерии оценки ресурсоэффективности							
Повышение производительности труда пользователя	0,3	4	4	4	1,2	1,2	1,2
Удобство в эксплуатации	0,1	5	3	4	0,5	0,3	0,4
Функциональная мощность	0,3	3	5	4	0,9	1,5	1,2
Качество интеллектуального интерфейса	0,05	4	2	3	0,2	0,1	0,15
Кроссплатформенность	0,1	5	1	3	0,5	0,1	0,3
Экономические критерии оценки эффективности							
Цена	0,1	5	2	5	0,5	0,2	0,5
Предполагаемый срок эксплуатации	0,05	5	5	5	0,25	0,25	0,25
Итого	1				4,05	3,65	4

Анализ конкурентоспособности показал, что разрабатываемая компонента опережает технические и экономические показатели эффективности аналогичных компонент в других программных продуктах. Основными преимущественными факторами являются кроссплатформенность и удобство в эксплуатации. Наиболее слабым фактором оказалась функциональная мощность.

5.2.3 SWOT-анализ

SWOT-анализ представляет собой комплексный анализ проекта, который применяют для исследования его внешней и внутренней среды. В ходе SWOT-анализа необходимо выявить внутренние и внешние факторы среды проекта, которые подразделяются на:

- Strengths (сильные стороны);

- Weaknesses (слабые стороны);
- Opportunities (возможности);
- Threats (угрозы).

В результате проведения SWOT-анализа формируется матрица SWOT.

Она представлена в таблице 6.

Таблица 6 – Матрица SWOT

	Сильные стороны:	Слабые стороны:
	<ol style="list-style-type: none"> 1. Кроссплатформенность 2. Удобство в эксплуатации 3. Интуитивно-понятный пользовательский интерфейс 	<ol style="list-style-type: none"> 1. Мало функциональных возможностей 2. Отсутствие интеграции с программами отображения поверхностей
<p>Возможности:</p> <ol style="list-style-type: none"> 1. Расширение функциональных возможностей 2. Интеграция с программами отображения поверхностей 3. Тенденция использования веб-приложений 	<p>Благодаря кроссплатформенности и удобству компонента является доступным инструментом не только для пользователей специализированных программ, но и для неопытных пользователей</p>	<p>Расширение функциональных возможностей и интеграция компоненты с программами отображения поверхностей может повысить конкурентоспособность проекта</p>
<p>Угрозы:</p> <ol style="list-style-type: none"> 1. Высокая производительность десктопных приложений конкурентов 2. Быстрое изменение трендов технологий веб-приложений 	<p>Существуют альтернативные более производительные приложения, но они доступны на ограниченном количестве платформ</p>	<p>Оптимизация работы компоненты и интеграция ее с программами отображения поверхностей поможет избежать самого негативного сценария</p>

В результате проведения SWOT-анализа выявлены сильные и слабые стороны проекта, его возможности и угрозы. Были рассмотрены возможные варианты продвижения продукта на рынке в зависимости от внешних условий.

5.3 Планирование работ

5.3.1 Структура работ

Процесс проведения работ в рамках проекта имеет определенную структуру и может быть разбит на определенные этапы. Перечень этапов и распределение исполнителей по ним приведены в таблице 7. Обозначения в таблице:

- НР – научный руководитель;
- Р – разработчик.

Таблица 7 – Этапы работ по проекту

№ этапа	Название этапа	Исполнитель
1	Выдача задания	НР
2	Формирование и утверждение технического задания	НР Р
3	Разработка календарного плана	Р
4	Анализ предметной области	Р
5	Проектирование компоненты	Р
6	Разработка компоненты	Р
7	Тестирование компоненты	Р
8	Согласование результатов работы с научным руководителем	НР Р
9	Подведение итогов, оформление работы	Р

5.3.2 Определение продолжительности этапов работ

Трудовые затраты образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников проекта.

Трудоемкость выполнения работы оценивается экспертным путем в человеко-днях и носит вероятностный характер, т.к. зависит от множества трудно учитываемых факторов. Для определения ожидаемого значения трудоемкости $t_{ож i}$ используется следующая формула:

$$t_{ож i} = \frac{3t_{мин i} + 2t_{маx i}}{5},$$

где $t_{ож i}$ – ожидаемая трудоемкость выполнения i -ой работы, чел.-дн.;

$t_{мин i}$ – минимально возможная трудоемкость выполнения заданной i -ой работы (оптимистическая оценка), чел.-дн.;

$t_{маx i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы (пессимистическая оценка), чел.-дн.

Количество рабочих дней для выполнения заданной i -ой работы будет равно ожидаемой трудоемкости выполнения i -ой работы, так как проект выполняет один исполнитель.

Для удобства построения графика проведения работ длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$T_{к i} = t_{ож i} \cdot k_{кал},$$

где $T_{к i}$ – продолжительность выполнения работы в календарных днях;

$k_{кал}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{кал} = \frac{T_{кал}}{T_{кал} - T_{вд} - T_{пд}},$$

где $T_{кал}$ – календарные дни;

$T_{вд}$ – выходные дни;

$T_{пд}$ – праздничные дни.

Для 2021 года коэффициент календарности будет равен:

$$k_{кал} = \frac{365}{365 - 118} = 1,478.$$

Рассчитанная продолжительность всех работ представлена в таблице 8. Также на основе этой таблицы построен календарный план-график (рисунок 17).

Таблица 8 – Продолжительность работ

№ этапа	Этап	Исполнитель	Трудоемкость работ			Длительность работ в календарных днях
			t_{\min} , чел.-дн.	t_{\max} , чел.-дн.	$t_{\text{ож}}$, чел.-дн.	
1	Выдача задания	НР	2	4	2,8	4
2	Формирование и утверждение технического задания	НР	4	8	5,6	8
		Р	2	4	2,8	4
3	Разработка календарного плана	Р	2	4	2,8	4
4	Анализ предметной области	Р	6	10	7,6	11
5	Проектирование компоненты	Р	8	12	9,6	14
6	Разработка компоненты	Р	32	42	36	53
7	Тестирование компоненты	Р	6	10	7,6	11
8	Согласование результатов работы с заказчиком	НР	4	8	5,6	8
		Р	4	8	5,6	8
9	Подведение итогов, оформление работы	Р	2	4	2,8	4
Итого		НР			14	20
		Р			74,8	109

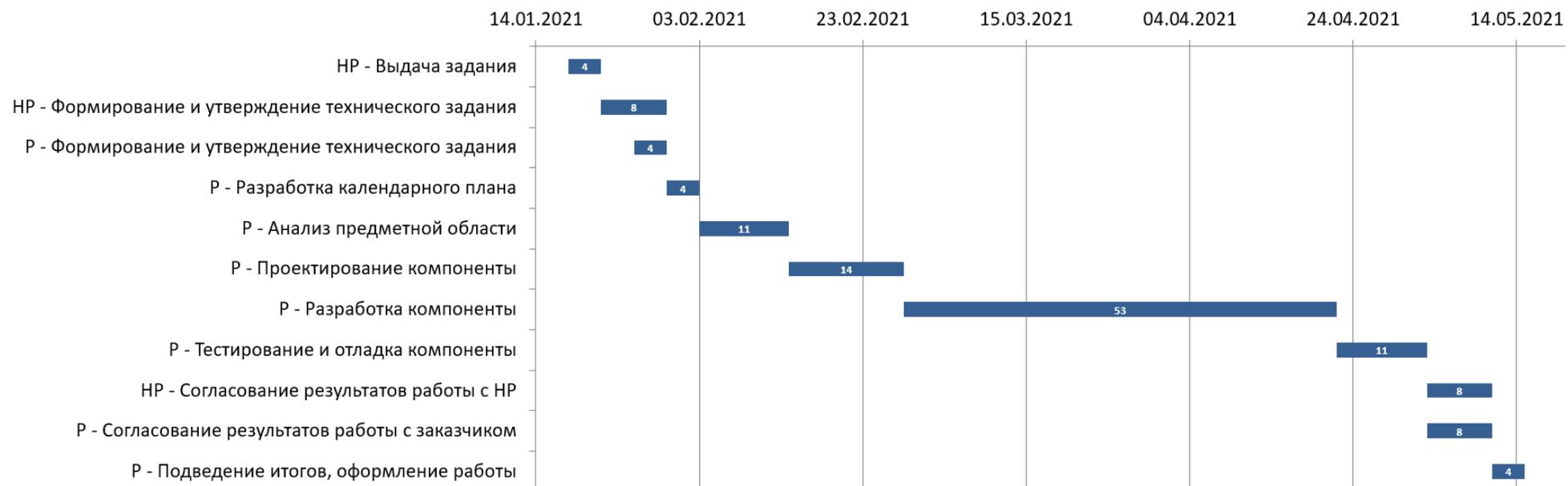


Рисунок 17 – Календарный план-график проведения работ

5.4 Бюджет работы

Рассчитаем бюджет работы в соответствии со следующими статьями затрат:

- материальные затраты;
- затраты на специальное оборудование для научных (экспериментальных) работ;
- основная заработная плата исполнителей;
- дополнительная заработная плата исполнителей;
- отчисления во внебюджетные фонды (страховые отчисления);
- затраты на научные и производственные командировки;
- амортизационные расходы;
- контрагентные расходы;
- накладные расходы.

При этом следует учитывать специфику продукта. Так как работа представляет собой проектирование и разработку программного обеспечения, материальные затраты и затраты на специальное оборудование будут равны нулю. Также для проектирования и разработки компоненты палитр нет необходимости совершения командировок, отклонения от нормальных условий труда и выполнения работ сторонними организациями, поэтому затраты на командировки, дополнительная заработная плата исполнителей и контрагентные расходы тоже не будут учитываться.

5.4.1 Расчет основной заработной платы исполнителей

Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя.

Расчет основной заработной платы приведен в таблице 9. Затраты времени в рабочих днях взяты из таблицы 8. Средний оклад младшего фронтенд-разработчика в Томске составляет приблизительно 30 000 руб.,

научного руководителя – 60 000 руб. Данные взяты с сайта для поиска работы «HeadHunter» [12].

Таблица 9 – Расчет основной заработной платы

Исполнитель	Оклад, руб./мес.	Среднее количество рабочих дней в месяце	Среднедневная ставка, руб./раб. день	Заплаты времени, дни	Районный коэффициент	Фонд з/платы, руб.
НР	60 000	20,58	2 915,45	20	1,3	75 801,75
Р	30 000	20,58	1 457,73	109	1,3	206 559,77
Итого:						282 361,52

5.4.2 Расчет отчислений во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования, пенсионного фонда и медицинского страхования от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$З_{внеб} = k_{внеб} \cdot З,$$

где $k_{внеб}$ – коэффициент отчислений на уплату во внебюджетные фонды;

$З$ – заработная плата.

В соответствии с таблицей 10 установлен размер страховых взносов, равный 30%.

Таблица 10 – Тарифы страховых взносов

Направление взносов	Лимит базы, рублей	Ставка, %
ПФР	В пределах 1 292 000	22
	Сверх 1 292 000	10
ФСС	В пределах 912 000	2,9 (1,8 за иностранных сотрудников со статусом временного пребывания)
	Сверх 912 000	Не уплачивается
ФФОМС	Не ограничена	5,1

Таким образом, затраты на отчисления во внебюджетные фонды будут составлять:

$$З_{внеб} = 0,3 \cdot 282\,361,52 = 84\,708,46.$$

5.4.3 Расчет амортизационных расходов

В данной статье расходов рассчитывается амортизация используемого оборудования за время выполнения проекта. Для расчета амортизационных расходов используется формула:

$$З_{ам} = \frac{N_{ам} \cdot Ц_{об} \cdot t_{рф}}{F_d},$$

где $N_{ам}$ – годовая норма амортизации единицы оборудования;

$Ц_{об}$ – балансовая стоимость единицы оборудования с учетом транспортно-заготовительных расходов;

$t_{рф}$ – фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта;

F_d – действительный годовой фонд времени работы соответствующего оборудования.

При этом годовая норма амортизации компьютера равна 0,4, так как срок полезного использования компьютера в среднем равен 2,5 годам ($1 / 2,5 = 0,4$). Балансовая стоимость компьютера на 2021 год составляет 50 000 руб. Фактическое время работы компьютера взято из таблицы 5 как ожидаемая

трудоемкость работ разработчика, переведённая в часы. Действительный годовой фонд времени работы компьютера равен 1752 $((365 - 118 - 28) \cdot 8)$.

Таким образом, амортизационные расходы составят:

$$Z_{\text{ам}} = \frac{0,4 \cdot 50000 \cdot (112 + 598,4)}{1752} = 8\,109,59.$$

5.4.4 Расчет накладных расходов

Накладные расходы учитывают затраты, не попавшие в предыдущие статьи расходов: печать и ксерокопирование, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$Z_{\text{накл}} = \text{сумма статей} \cdot k_{\text{нр}},$$

где $k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы.

Величину коэффициента накладных расходов можно взять в размере 16%. Тогда накладные расходы составят:

$$Z_{\text{накл}} = (282\,361,52 + 84\,708,46 + 8\,109,59) \cdot 0,16 = 60\,028,73.$$

5.4.5 Формирование бюджета затрат проекта

Определение бюджета затрат на работу представлено в таблице 11.

Таблица 11 – Бюджет работы

Наименование статьи	Сумма, руб.
Основная заработная плата исполнителей	282 361,52
Отчисления во внебюджетные фонды	84 708,46
Амортизационные расходы	8 109,59
Накладные расходы	60 028,73
Итого:	435 208,3

5.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности проекта

При расчёте показателей эффективности оценим 3 варианта исполнения компоненты:

1. С использованием фреймворка и дополнительных библиотек;
2. С использованием фреймворка;
3. Нативное веб-приложение.

Определение эффективности происходит на основе расчета интегрального показателя эффективности проекта. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный финансовый показатель разработки определяется как:

$$I_{\text{финр}}^{\text{исп } i} = \frac{\Phi_{p i}}{\Phi_{\text{max}}},$$

где $I_{\text{финр}}^{\text{исп } i}$ – интегральный финансовый показатель разработки;

$\Phi_{p i}$ – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения проекта

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{p i} = \sum a_i \cdot b_i,$$

где $I_{p i}$ – интегральный показатель ресурсоэффективности;

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i – бальная оценка i -го варианта исполнения разработки.

Расчет интегрального показателя ресурсоэффективности приведен в таблице 12.

Таблица 12 – Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Весовой коэффициент параметра	И1	И2	И3
Простота разработки	0,15	5	3	4
Расширяемость функционала	0,3	5	2	3
Производительность	0,2	3	3	3
Простота поддержки	0,15	5	3	3
Надежность	0,2	4	4	4
Итого:	1	4,4	2,9	3,35

Интегральный показатель эффективности вариантов исполнения разработки определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{\text{исп } i} = \frac{I_{\text{р-исп } i}}{I_{\text{финр } i}},$$

где $I_{\text{исп } i}$ – интегральный показатель эффективности варианта исполнения;

$I_{\text{р-исп } i}$ – интегральный показатель ресурсоэффективности;

$I_{\text{финр } i}$ – интегральный финансовый показатель.

Сравнительная эффективность варианта исполнения проекта определяется формулой:

$$\mathcal{E}_{\text{ср}} = \frac{I_{\text{исп } i}}{I_{\text{исп } 0}},$$

$\mathcal{E}_{\text{ср}}$ – сравнительная эффективность варианта исполнения проекта;

$I_{\text{исп } i}$ – интегральный показатель эффективности варианта исполнения проекта;

$I_{\text{исп } 0}$ – интегральный показатель эффективности лучшего варианта исполнения проекта.

Сравнение показателей эффективности вариантов исполнения проекта представлено в таблице 13. Значения интегрального финансового показателя для первого исполнения примем как 0,7, для второго – 0,9, а для третьего – 1, так как разработка компоненты без использования сторонних библиотеки более трудоемка и, следовательно, увеличит бюджет проекта, а разработка без использования фреймворка повышает трудоемкость в разы.

Таблица 13 – Сравнительная эффективность разработки

№	Показатели	И1	И2	И3
1	Интегральный финансовый показатель разработки	0,7	0,9	1
2	Интегральный показатель ресурсоэффективности разработки	4,4	2,9	3,35

3	Интегральный показатель эффективности	6,29	3,22	3,35
4	Сравнительная эффективность вариантов исполнения	1	0,51	0,53

Таким образом, первый вариант исполнения является наиболее эффективным.

5.6 Выводы по разделу

В ходе работы над данным разделом были выявлены потенциальные потребители продукта, проанализированы конкурентные решения, проведен SWOT-анализ. Также было проведено планирование работ, рассчитан бюджет и эффективность проекта.

Основываясь на результатах проведенного анализа, можно сделать вывод, что разработка компоненты редактирования палитр является конкурентоспособной и перспективной.

ГЛАВА 6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

6.1 Введение

В работе рассматривается компонента программного обеспечения, позволяющая редактировать палитры – параметры отображения линий уровня и интервальной закрашки между линиями при отображении геофизических данных в виде растровых изображений. Таким образом, пользователями разработанной компоненты будут геологи, работающие на своих рабочих местах за компьютером. Следовательно, необходимо более подробно рассмотреть защиту человека от технических систем и технологий, а именно защиту пользователей компьютерной техники. Работы с применением разрабатываемой компоненты будут проводиться в Томске в офисных помещениях.

6.2 Правовые и организационные вопросы обеспечения безопасности

В трудовом кодексе РФ содержатся основные положения отношений между организацией и сотрудниками, включая оплату и нормирование труда, выходных, отпуска и так далее [13].

В соответствии со ст. 129 ТК РФ работодателем устанавливается тарифная ставка – фиксированный размер оплаты труда работника за выполнение нормы труда определенной сложности (квалификации) за единицу времени без учета компенсационных, стимулирующих и социальных выплат. При невыполнении норм труда порядок и размер оплаты труда может меняться. Ст. 155 ТК РФ регламентирует порядок оплаты труда при невыполнении норм труда в зависимости от того, по чьей вине нормы труда не были выполнены (таблица 14).

Таблица 14 – Варианты оплаты труда при невыполнении трудовых норм

Причина невыполнения нормы труда	Порядок оплаты
По вине работодателя	В размере не ниже средней заработной платы работника, рассчитанной пропорционально фактически отработанному времени
По причинам, не зависящим от работодателя и работника	Не менее двух третей тарифной ставки, оклада, рассчитанных пропорционально фактически отработанному времени
По вине работника	Оплата в соответствии с объемом фактически выполненной работы

Работа в офисе относится ко второй категории тяжести труда – работы выполняются при оптимальных условиях внешней производственной среды и при оптимальной величине физической, умственной и нервно-эмоциональной нагрузки. Продолжительность рабочего дня работников не должна превышать 40 часов в неделю (ст. 91 ТК РФ).

Для сохранения здоровья и обеспечения оптимальной работоспособности пользователей ПЭВМ на протяжении рабочей смены должны устанавливаться регламентированные перерывы. Ст. 109 ТК РФ предусматривает предоставление работникам в течение рабочего времени специальных перерывов, обусловленных технологией и организацией производства и труда. Виды этих работ, продолжительность и порядок предоставления таких перерывов устанавливаются правилами внутреннего трудового распорядка. В отличие от перерывов для отдыха и питания, предусмотренных ст. 108 ТК, они предоставляются в течение рабочего времени, включаются в рабочее время и оплачиваются.

При восьмичасовом рабочем дне перерывы составляют 15 минут через два часа после начала работы и через два часа после перерыва на обед для сотрудников, чья деятельность связана со считыванием информации с экрана.

В соответствии с ГОСТ 22269-76 «Рабочее место оператора. Взаимное расположение элементов рабочего места» на рабочем месте необходимо учитывать [14]:

- рабочую позу пользователя;
- пространство для размещения пользователя;
- возможность обзора элементов рабочего места;
- возможность обзора пространства за пределами рабочего места;
- возможность ведения записей, размещения используемой документации и материалов.

Взаимное расположение элементов рабочего места должно обеспечивать возможность осуществления всех необходимых движений для эксплуатации компьютера. Пользователь должен хорошо видеть монитор и слышать звуковые сигналы, подаваемые операционной системой и используемыми программами. Взаимное расположение элементов рабочего места должно способствовать оптимальному режиму труда и отдыха, снижению утомления, предупреждению появления ошибочных действий.

ГОСТ 12.2.032-78 «Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования» устанавливает общие эргономические требования к рабочим местам при выполнении работ в положении сидя [15].

Рабочее место при выполнении работ сидя – рабочий стол. При работе за столом должно быть обеспечено оптимальное положение работающего, которое достигается регулированием высоты рабочей поверхности, сиденья и пространства для ног. При нерегулируемой высоте рабочей поверхности – регулированием высоты сиденья и подставки для ног. В этом случае высота рабочей поверхности должна составлять 630 мм для женщин, 680 мм для мужчин и 655 мм для женщин и мужчин.

Клавиатура и компьютерная мышь должны располагаться на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю.

Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно перпендикулярно к нормальной линии взгляда в 15 градусов вниз от горизонтали.

6.3 Производственная безопасность

Условия труда, в которых эксплуатируется компонента палитр для отображения геофизических данных, могут спровоцировать появление вредных и опасных факторов производства.

При выполнении работ на персональном компьютере согласно ГОСТ 12.0.003-2015 «Система стандартов безопасности труда. Опасные и вредные производственные факторы. Классификация» могут иметь место следующие факторы, представленные в таблице 15 [16].

Таблица 15 – Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ		Нормативные документы
	Разра- ботка	Эксплу- атация	
1. Отсутствие или недостаток необходимого освещения	+	+	СНиП 23-05-95* «Естественное и искусственное освещение» [17]
2. Аномальные микроклиматические параметры воздушной среды	+	+	СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений» [18]
3. Повышенный уровень и другие неблагоприятные характеристики шума	+	+	СН 2.2.4/ 2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки» [19]

4. Повышенное значение напряжения в электрической цепи; короткое замыкание	+	+	ГОСТ 12.1.038-82 «Система стандартов безопасности труда. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов» [20]
--	---	---	---

1. Отсутствие или недостаток необходимого освещения.

Для пользователя компьютера недостаточная освещенность рабочей зоны является одной из причин нарушения зрительной функции. Кроме того, этот вредный фактор может приводить к снижению эффективности труда и ухудшать общее самочувствие.

Искусственное освещение в помещениях для эксплуатации ПК должно осуществляться системой общего равномерного освещения. В случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения). Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. При этом столы следует располагать так, чтобы свет от окон падал на рабочую поверхность стола слева.

СНиП 23-05-95* «Естественное и искусственное освещение» регламентирует требования к освещению помещений промышленных предприятий (КЕО, нормируемая освещенность, объединенный показатель дискомфорта и коэффициент пульсации освещенности). Так как работа за компьютером является работой средней точности (IV разряд зрительной работы), то показатели освещения рабочего места пользователя ПК следует принимать по таблице 16.

Таблица 16 – Нормируемые параметры освещения

Подразряд зрительной работы	Контраст объекта с фоном	Характеристика фона	Искусственное освещение				
			Освещенность, лк			Сочетание нормируемых величин показателя ослепленности и коэффициента пульсации	
			при системе комбинированного освещения		при системе общего освещения		
			всего	в том числе от общего			P
а	Малый	Темный	750	200	300	40	20
б	Малый Средний	Средний Темный	500	200	200	40	20
в	Малый Средний Большой	Светлый Средний Темный	400	200	200	40	20
г	Средний Большой »	Светлый » Средний	–	–	200	40	20
Естественное освещение				Совмещенное освещение			
КЕО $e_H, \%$							
при верхнем или комбинированном освещении		при боковом освещении		при верхнем или комбинированном освещении		при боковом освещении	
4		1,5		2,4		0,9	

2. Аномальные микроклиматические параметры воздушной среды.

Как сказано в СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений», показатели микроклимата должны обеспечивать сохранение теплового баланса человека с окружающей средой и поддержание оптимального или допустимого теплового состояния организма.

По уровню энергозатрат работу за компьютером можно отнести к категории Ia, так как при такой сидячей малоактивной деятельности человек тратит очень мало энергии – менее 139 Вт (120 ккал/ч). Для данной категории будут соответствовать показатели микроклимата, представленные в таблице 17.

Таблица 17 – Оптимальные параметры микроклимата

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	22-24	21-25	60-40	0,1
Теплый	23-25	22-26	60-40	0,1

3. Повышенный уровень и другие неблагоприятные характеристики шума.

В соответствии с СН 2.2.4/ 2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки» необходимо рассмотреть категорию работников, занимающихся творческой деятельностью, руководящей работой, научной деятельностью, конструированием и проектированием, программированием, преподаванием и обучением на рабочих местах в помещениях дирекции, проектно-конструкторских бюро, в лабораториях для теоретических работ и обработки данных, потому что эксплуатация разрабатываемой компоненты палитр предполагается на рабочих местах за компьютером в офисных помещениях. Предельно допустимые уровни звукового давления в октавных полосах частот, уровни звука и эквивалентные уровни звука для данной категории представлены в таблице 18.

Таблица 18 – Уровни звукового давления

Уровни звукового давления, дБ, в составных полосах со среднегеометрическими частотами, Гц									Уровни звука и эквивалентные уровни звука, дБА
31,5	63	125	250	500	1000	2000	4000	8000	
86	71	61	54	49	45	42	40	38	50

Если какое-либо оборудование превышает нормативные уровни шума, то его необходимо размещать в других помещениях. Также для снижения уровня шума можно установить звукоизолирующие перегородки между рабочими столами, использовать звукопоглощающие материалы для отделки стен и потолка.

4. Повышенное значение напряжения в электрической цепи; короткое замыкание.

Для питания компьютерной техники используется электричество. Поэтому при работе за компьютером необходимо соблюдать правила ГОСТ 12.1.038-82 «Система стандартов безопасности труда. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов». Несоблюдение этих правил может привести к поражению электрическим током при прикосновении к токоведущим частям, находящимся под напряжением, на которых остался заряд или появилось напряжение.

В соответствии с ГОСТ 12.1.038-82 «Система стандартов безопасности труда. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов» напряжения прикосновения и токи, протекающие через тело человека при нормальном (неаварийном) режиме электроустановки, не должны превышать значений, указанных в таблице 19. Указанные допустимые значения напряжений прикосновения и токов установлены для путей тока от одной руки к другой и от руки к ногам.

Таблица 19 – Допустимые значения напряжения прикосновения и тока

Род тока	U, В	I, мА
	не более	
Переменный 50 Гц	2,0	0,3
Переменный 400 Гц	3,0	0,4
Постоянный	8,0	1,0

При работе за компьютером самым доступным устройством для защиты от перепадов напряжения и от повышенного значения напряжения в электрической цепи является сетевой фильтр с предохранителем, который может полностью обесточить подключенные устройства, если будет серьезный сдвиг фаз. Также можно использовать стабилизаторы напряжения и источники бесперебойного питания. Кроме того, мерами защиты от воздействия электрического тока могут послужить устройства автоматического контроля и сигнализации, изолирующие устройства и покрытия, устройства защитного заземления.

6.4 Экологическая безопасность

На рабочем месте за компьютером выявлен предполагаемый источник загрязнения окружающей среды. При поломке предметов вычислительной техники (компьютеров) и оргтехники образуются отходы.

Вышедшие из строя ПЭВМ и сопутствующая оргтехника относятся к IV классу опасности и подлежат специальной утилизации. Для оказания наименьшего влияния на окружающую среду необходимо проводить специальную процедуру утилизации ПЭВМ и оргтехники, при которой более 90% отправится на вторичную переработку и менее 10% будут отправлены на свалки. При этом она должна соответствовать процедуре утилизации ГОСТ Р 53692-2009 «Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов» [21].

Кроме вышедших из строя ПЭВМ, в ходе деятельности организации образуются бытовые отходы (канцелярские отходы, искусственные источники освещения), которые должны быть утилизированы в соответствии с определенным классом опасности или переработаны, чтобы не оказывать негативное влияние на состояние окружающей среды. Так, отработанные ртутьсодержащие и люминесцентные лампы, относящиеся к I классу, должны быть переданы в специализированные организации, которые обеспечивают условия для накопления и возможного хранения указанных опасных отходов. Канцелярские отходы, относящиеся к V классу, считаются неопасными и не требуют специальной утилизации.

6.5 Безопасность в чрезвычайных ситуациях

В качестве возможных чрезвычайных ситуаций на рабочем месте за компьютером можно выделить: пожары, наводнения, землетрясения, удары молнией, взрывы, террористические акты.

С учетом специфики работы и наличием вычислительной техники в помещении наиболее типичной чрезвычайной ситуацией будет пожар, под которым понимается неконтролируемый процесс горения, возникший из-за возгорания вычислительной техники и угрожающий жизни и здоровью работников.

Причинами возгорания при работе с компьютером могут быть:

- короткое замыкание;
- неисправность компьютера или электросетей;
- небрежность оператора при работе с компьютером;
- воспламенение ПЭВМ из-за перегрузки.

В связи с этим, согласно ГОСТ 12.1.004-91 «Пожарная безопасность. Общие требования», при работе с компьютером необходимо соблюдать следующие нормы пожарной безопасности [22]:

- для предохранения сети от перегрузок запрещается одновременно подключать к сети количество потребителей, превышающих допустимую нагрузку;
- работы за компьютером проводить только при исправном состоянии оборудования, электропроводки;
- иметь средства для тушения пожара (огнетушитель);
- установить количество, размеры и соответствующее конструктивное исполнение эвакуационных путей и выходов;
- обеспечить возможность беспрепятственного движения людей по эвакуационным путям.

Хорошим вариантом предотвращения возгорания может стать прокладка всех видов кабелей в металлических газонаполненных трубах. При появлении пожара, любой, увидевший пожар должен: незамедлительно заявить о данном в пожарную службу по телефонному номеру 01 или 112, заявить о происшествии и соблюдать покой.

В случае возникновения пожара в здании автоматически срабатывают датчики пожаротушения, и звуковая система оповещает всех сотрудников о немедленной эвакуации из здания, после чего сотрудники направляются на выход в соответствии с планом эвакуации при пожарах и других ЧС.

6.6 Выводы по разделу

В данном разделе были рассмотрены вопросы правового и организационного обеспечения безопасности, производственной и экологической безопасности, а также безопасности в чрезвычайных ситуациях.

В результате работы над разделом можно сделать вывод, что рабочее место, за которым и будет эксплуатироваться разрабатываемая компонента палитр, удовлетворяет всем нормам и правилам безопасности. Также следует отметить необходимость готовности сотрудников к возникновению чрезвычайных ситуаций.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была разработана компонента редактирования палитр, используемая в программах анализа и отображения геофизических данных.

В ходе выполнения работы было выполнено:

- Проанализирована предметная область, изучены аналоги разрабатываемого приложения и выявлены лучшие варианты реализации компоненты редактирования палитр;
- Спроектирована компонента, разработаны UML-диаграммы, описывающие ее функции;
- Разработана компонента;
- Протестирована компонента.

Основным результатом работы является созданная компонента редактирования палитр. При внедрении ее в разрабатываемое веб-приложение по обработке и интерпретации данных вертикального сейсмического профиля разработанная компонента позволит повысить качество и удобство обработки геофизических данных.

В результате проведения финансового анализа были выявлены потенциальные потребители, проанализированы конкурентные решения, проведено планирование работ, рассчитан бюджет и эффективность проекта.

В ходе анализа данных социальной ответственности были рассмотрены вопросы правового и организационного обеспечения безопасности, производственной и экологической безопасности, а также безопасности в чрезвычайных ситуациях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. UML: [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/UML> (дата обращения: 08.05.2021).
2. Объектно-ориентированный подход к анализу и проектированию информационных систем: [Электронный ресурс]. URL: <https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture> (дата обращения: 08.05.2021).
3. Surfer: [Электронный ресурс]. URL: <https://www.goldensoftware.com/products/surfer> (дата обращения: 08.05.2021).
4. Vue.js: [Электронный ресурс]. URL: <https://ru.vuejs.org/> (дата обращения: 10.05.2021).
5. The TypeScript Handbook: [Электронный ресурс]. URL: <https://www.typescriptlang.org/docs/handbook/intro.html> (дата обращения: 10.05.2021).
6. Vuex: [Электронный ресурс]. URL: <https://vuex.vuejs.org/ru/> (дата обращения: 10.05.2021).
7. PrimeVue: [Электронный ресурс]. URL: <https://www.primefaces.org/primevue/> (дата обращения: 10.05.2021).
8. Пишем максимально эффективный тест-кейс: [Электронный ресурс]. URL: <https://habr.com/ru/post/246463/> (дата обращения: 13.05.2021).
9. Jest: [Электронный ресурс]. URL: <https://jestjs.io/> (дата обращения: 13.05.2021).
10. Финансовый менеджмент, ресурсоэффективность и ресурсо-сбережение: учебно-методическое пособие / И.Г. Видяев, Г.Н. Серикова, Н.А. Гаврикова, Н.В. Шаповалова, Л.Р. Тухватулина З.В. Креницына; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2014. – 36 с.
11. QGIS: [Электронный ресурс]. URL: <https://www.qgis.org/ru/site/> (дата обращения: 24.04.2021).

12. HeadHunter: [Электронный ресурс]. URL: <https://tomsk.hh.ru/> (дата обращения: 25.04.2021).
13. Трудовой кодекс Российской Федерации: федеральный закон от 30.12.2001 № 197-ФЗ: [принят Государственной Думой 21 декабря 2001 года]. – Доступ из справ.-правовой системы Гарант. – Текст: электронный.
14. ГОСТ 22269-76 Рабочее место оператора. Взаимное расположение элементов рабочего места: введен в действие постановлением Государственного комитета стандартов Совета Министров СССР от 22 декабря 1976 г. N 2798: дата введения 1978-01-01. – URL: <https://docs.cntd.ru/document/1200012834> (дата обращения: 17.04.2021). – Текст: электронный.
15. ГОСТ 12.2.032-78 Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования: введен в действие постановлением Государственного комитета стандартов Совета Министров СССР от 26 апреля 1978 г. N 1102: дата введения 1979-01-01. – URL: <https://docs.cntd.ru/document/1200003913> (дата обращения: 17.04.2021). – Текст: электронный.
16. ГОСТ 12.0.003-2015 Система стандартов безопасности труда. Опасные и вредные производственные факторы. Классификация: введен в действие приказом Федерального агентства по техническому регулированию и метрологии от 9 июня 2016 г. N 602-ст: дата введения 2017-03-01. – URL: <https://docs.cntd.ru/document/1200136071> (дата обращения: 18.04.2021). – Текст: электронный.
17. СНиП 23-05-95* Естественное и искусственное освещение: приняты и введены в действие постановлением Минстроя России от 2 августа 1995 г. N 18-78: дата введения 1996-01-01. – URL: <https://docs.cntd.ru/document/871001026> (дата обращения: 18.04.2021). – Текст: электронный.
18. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений: утверждены и введены в действие

постановлением Госкомсанэпиднадзора России от 1 октября 1996 г. N 21: дата введения с момента утверждения. – URL: <https://docs.cntd.ru/document/901704046> (дата обращения: 18.04.2021). – Текст: электронный.

19. СН 2.2.4/ 2.1.8.562-96 Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки: утверждены и введены в действие постановлением Госкомсанэпиднадзора России от 31 октября 1996 г. N 36: дата введения с момента утверждения. – URL: <https://docs.cntd.ru/document/901703278> (дата обращения: 18.04.2021). – Текст: электронный.
20. ГОСТ 12.1.038-82 Система стандартов безопасности труда. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов: введен в действие постановлением Государственного комитета СССР по стандартам от 30.07.82 N 2987: дата введения 1983-07-01. – URL: <https://docs.cntd.ru/document/5200313> (дата обращения: 19.04.2021). – Текст: электронный.
21. ГОСТ Р 53692-2009 Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов: утвержден и введен в действие приказом Федерального агентства по техническому регулированию и метрологии от 15 декабря 2009 г. N 1092-ст: дата введения 2011-01-01. – URL: <https://docs.cntd.ru/document/1200081740> (дата обращения: 20.04.2021). – Текст: электронный.
22. ГОСТ 12.1.004-91 Пожарная безопасность. Общие требования: утвержден и введен в действие постановлением Государственного комитета СССР по управлению качеством продукции и стандартам от 14.06.91 N 875: дата введения 1992-07-01. – URL: <https://docs.cntd.ru/document/9051953> (дата обращения: 21.04.2021). – Текст: электронный.

ПРИЛОЖЕНИЕ А

Листинг А.1 – Программная реализация масштабирования уровней

```
import {State} from '@store/State'
import {Level} from '@models/Level'
import {LineStyle} from '@models/LineStyle'
import {Color} from '@models/Color'
import {Palette} from '@models/Palette'

export interface ScaleInfo {
  scaleMin: number;
  scaleMax: number;
  scaleInterval: number;
  isScaleLines?: boolean;
  isScaleFilling?: boolean;
}

interface ScalableLevel extends Level {
  scaleValue: number;
}

export const countPerCent = (value: number, min: number, max: number): number =>
(value - min) / (max - min) * 100

export const countValue = (perCent: number, min: number, max: number): number => min
+ (max - min) * perCent / 100

export const formatLevelsToScalable = (levels: Level[]): ScalableLevel[] => {
  const scalableLevels: ScalableLevel[] = []
  levels.forEach(level => {
    scalableLevels.push({
      ...level,
      scaleValue: countPerCent(level.value, levels[0].value, levels[levels.length -
1].value)
    } as ScalableLevel)
  })
  return scalableLevels
}

const findSuitableLevels = (levels: ScalableLevel[], minVal: number, maxVal: number):
ScalableLevel[] => {
  const suitableLevels: ScalableLevel[] = []
  for (let i = 0; i < levels.length; i++) {
    if (levels[i].scaleValue < minVal) {
      continue
    } else if (levels[i].scaleValue > maxVal) {
      break
    }
    suitableLevels.push(levels[i])
  }
  return suitableLevels
}

const scaleLevels = (newLevels: Level[], oldLevels: Level[], scaleLines: boolean,
scaleFilling: boolean): Level[] => {
  const scalableNewLevels = formatLevelsToScalable(newLevels)
  const scalableOldLevels = formatLevelsToScalable(oldLevels)
```

```

const scaledLevels = []

let midVal2 = scalableOldLevels[0].scaleValue + (scalableOldLevels[1].scaleValue -
scalableOldLevels[0].scaleValue) / 2
let suitableNewLevels = findSuitableLevels(scalableNewLevels,
scalableOldLevels[0].scaleValue, midVal2)
suitableNewLevels.forEach(level => {
  if (scaleLines) {
    level.lineColor = scalableOldLevels[0].lineColor
    level.lineStyle = scalableOldLevels[0].lineStyle
    level.lineWidth = scalableOldLevels[0].lineWidth
  }
  if (scaleFilling) {
    level.fillColor = scalableOldLevels[0].fillColor
  }
  scaledLevels.push(level)
})

for (let i = 0; i < scalableOldLevels.length - 2; i++) {
  const midVal1 = scalableOldLevels[i].scaleValue +
    (scalableOldLevels[i + 1].scaleValue - scalableOldLevels[i].scaleValue) / 2
  midVal2 = scalableOldLevels[i + 1].scaleValue +
    (scalableOldLevels[i + 2].scaleValue - scalableOldLevels[i + 1].scaleValue) / 2
  suitableNewLevels = findSuitableLevels(scalableNewLevels, midVal1, midVal2)
  suitableNewLevels.forEach(level => {
    if (scaleLines) {
      level.lineColor = scalableOldLevels[i + 1].lineColor
      level.lineStyle = scalableOldLevels[i + 1].lineStyle
      level.lineWidth = scalableOldLevels[i + 1].lineWidth
    }
    if (scaleFilling) {
      level.fillColor = scalableOldLevels[i + 1].fillColor
    }
    scaledLevels.push(level)
  })
}

const midVal1 = scalableOldLevels[scalableOldLevels.length - 2].scaleValue +
  (scalableOldLevels[scalableOldLevels.length - 1].scaleValue -
  scalableOldLevels[scalableOldLevels.length - 2].scaleValue) / 2
suitableNewLevels = findSuitableLevels(scalableNewLevels, midVal1,
  scalableOldLevels[scalableOldLevels.length - 1].scaleValue)
suitableNewLevels.forEach(level => {
  if (scaleLines) {
    level.lineColor = scalableOldLevels[scalableOldLevels.length - 1].lineColor
    level.lineStyle = scalableOldLevels[scalableOldLevels.length - 1].lineStyle
    level.lineWidth = scalableOldLevels[scalableOldLevels.length - 1].lineWidth
  }
  if (scaleFilling) {
    level.fillColor = scalableOldLevels[scalableOldLevels.length - 1].fillColor
  }
  scaledLevels.push(level)
})

scaledLevels.forEach(level => delete level.scaleValue)
return scaledLevels
}

export const binarySearch = <T extends { value: number }>(data: T[], target: number,

```

```

start: number, end: number): T => {
  if (end < 1) {
    return data[0]
  }
  const middle = Math.floor((start + (end - start) / 2))
  if (target == data[middle].value) {
    return data[middle]
  }
  if (end - 1 === start) {
    return Math.abs(data[start].value - target) > Math.abs(data[end].value - target)
? data[end] : data[start]
  }
  if (target > data[middle].value) {
    return binarySearch(data, target, middle, end)
  }
  if (target < data[middle].value) {
    return binarySearch(data, target, start, middle)
  }
}
}

```

```

export default {
  state: () => ({
    showScaleDialog: false,
    scaleMin: null,
    scaleMax: null,
    scaleInterval: 25,
    isScaleLines: false,
    isScaleFilling: false
  )),
  mutations: {
    SHOW_SCALE_DIALOG(state: State) {
      state.showScaleDialog = true
    },
    HIDE_SCALE_DIALOG(state: State) {
      state.showScaleDialog = false
    },
    SET_SCALE_LIMITS(state: State, scaleInfo: ScaleInfo) {
      state.scaleMin = scaleInfo.scaleMin
      state.scaleMax = scaleInfo.scaleMax
      state.scaleInterval = scaleInfo.scaleInterval
      state.isScaleLines = scaleInfo.isScaleLines
      state.isScaleFilling = scaleInfo.isScaleFilling
    }
  },
  actions: {
    SHOW_SCALE_DIALOG({commit}) {
      commit('SHOW_SCALE_DIALOG')
    },
    HIDE_SCALE_DIALOG({commit}) {
      commit('HIDE_SCALE_DIALOG')
    },
  },
}

```

```

    SET_SCALE_LIMITS({commit, getters}, payloadScaleInfo?: ScaleInfo) {
      let payloadMin, payloadMax, payloadInterval, payloadIsScaleLines,
      payloadIsScaleFilling
      if (payloadScaleInfo) {
        payloadMin = payloadScaleInfo.scaleMin
        payloadMax = payloadScaleInfo.scaleMax
        payloadInterval = payloadScaleInfo.scaleInterval
        payloadIsScaleLines = payloadScaleInfo.isScaleLines
        payloadIsScaleFilling = payloadScaleInfo.isScaleFilling
      } else {
        payloadMin = getters.levels[0].value
        payloadMax = getters.levels[getters.levels.length - 1].value
        payloadInterval = getters.scaleInterval
        payloadIsScaleLines = false
        payloadIsScaleFilling = false
      }
      const scaleInfo = {
        scaleMin: payloadMin - payloadMin % payloadInterval,
        scaleMax: payloadMax + payloadInterval - payloadMax % payloadInterval,
        scaleInterval: payloadInterval,
        isScaleLines: payloadIsScaleLines,
        isScaleFilling: payloadIsScaleFilling
      } as ScaleInfo
      commit('SET_SCALE_LIMITS', scaleInfo)
    },

    SCALE_LEVELS({commit, dispatch, getters}, payloadScaleInfo: ScaleInfo) {
      dispatch('SET_SCALE_LIMITS', payloadScaleInfo)

      let newLevels: Level[] = []
      const scaleMin = getters.scaleMin
      const scaleMax = getters.scaleMax
      const scaleInterval = getters.scaleInterval
      for (let i = 0; i < (scaleMax - scaleMin) / scaleInterval; i++) {
        newLevels.push({
          id: i,
          value: scaleMin + scaleInterval * i,
          lineStyle: LineStyles.SOLID,
          lineWidth: 1,
          lineColor: new Color('RGB(0,0,0)'),
          fillColor: new Color('RGB(255,255,255)')
        } as Level)
      }

      const oldLevels: Level[] = getters.levels
      if (getters.isScaleLines && getters.isScaleFilling) {
        // Масштабируем и линии, и цвета закраски
        newLevels = scaleLevels(newLevels, oldLevels, true, true)
      } else if (getters.isScaleLines) {
        // Масштабируем только линии
        newLevels = scaleLevels(newLevels, oldLevels, true, false)
        let previousOldLevel: Level = null
        oldLevels.forEach(oldLevel => {
          const suitableNewLevel = binarySearch(newLevels, oldLevel.value, 0,
            newLevels.length - 1)
          if (!(
            previousOldLevel &&
            Math.abs(previousOldLevel.value - suitableNewLevel.value) <

```

```

Math.abs(oldLevel.value - suitableNewLevel.value)
  )) {
    suitableNewLevel.fillColor = oldLevel.fillColor
  }
  previousOldLevel = oldLevel
})
} else if (getters.isScaleFilling) {
  // Масштабируем только цвета закраски
  newLevels = scaleLevels(newLevels, oldLevels, false, true)
  let previousOldLevel: Level = null
  oldLevels.forEach(oldLevel => {
    const suitableNewLevel = binarySearch(newLevels, oldLevel.value, 0,
newLevels.length - 1)
    if (!(
      previousOldLevel &&
      Math.abs(previousOldLevel.value - suitableNewLevel.value) <
Math.abs(oldLevel.value - suitableNewLevel.value)
    )) {
      suitableNewLevel.lineColor = oldLevel.lineColor
      suitableNewLevel.lineStyle = oldLevel.lineStyle
      suitableNewLevel.lineWidth = oldLevel.lineWidth
    }
    previousOldLevel = oldLevel
  })
}
}

const newPalette: Palette = getters.palette
newPalette.levels = newLevels
commit('SET_PALETTE', newPalette)
dispatch('SET_SCALE_LIMITS')
}
},

getters: {
  showScaleDialog: (state: State) => state.showScaleDialog,
  scaleMin: (state: State) => state.scaleMin,
  scaleMax: (state: State) => state.scaleMax,
  scaleInterval: (state: State) => state.scaleInterval,
  isScaleLines: (state: State) => state.isScaleLines,
  isScaleFilling: (state: State) => state.isScaleFilling
}
}
}

```