TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных систем и технологий
Направление подготовки 09.04.02 Информационные системы и технологии
Отделение школы (НОЦ) Отделение информационных технологий

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Проектирование и реализация адаптивного реактивного клиент-серверного приложения для сервиса доставки еды

УДК 004.451:004.415.2:004.85

Студент:

Группа	ФИО	Подпись	Дата
8ИМ01	Ким Станислав		

Руководитель ВКР:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Копнов М. В.	к.т.н., доцент		

Консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОИТ ИШИТР	Коровкин В. А.			

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГТ ШБИП	Былкова Татьяна Васильевна	к. э. н.		

По разделу «Социальная ответственность»:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ООДШБИП	Федоренко Ольга Юрьевна	д.м.н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Савельев Алексей Олегович	к.т.н., доцент		

Томск – 2022 г.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

по направлению 09.04.02 «Информационные системы и технологии»

Код результатов	Результат обучения (выпускник должен быть готов)
Универсальные компетенции	
УК(У)-1	Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий.
УК(У)-2	Способен управлять проектом на всех этапах его жизненного цикла.
УК(У)-3	Способен организовывать и руководить работой команды, вырабатываякомандную стратегию для достижения поставленной цели.
УК(У)-4	Способен применять современные коммуникативные технологии, в том числена иностранном(ых) языке(ах), для академического и профессионального взаимодействия.
УК(У)-5	Способен анализировать и учитывать разнообразие культур в процессе межкультурного взаимодействия
УК(У)-6	Способен определять и реализовывать приоритеты собственной деятельностии способы ее совершенствования на основе самооценки
Общепрофессиональные компетенции	
ОПК(У)-1.	Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте
ОПК(У)-2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, длярешения профессиональных задач
ОПК(У)-3	Способен анализировать профессиональную информацию, выделять в нейглавное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями
ОПК(У)-4	Способен применять на практике новые научные принципы и методыисследований
ОПК(У)-5	Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных
Профессиональные компетенции	
ПК (У)-1	Способен проектировать сложные пользовательские интерфейсы; анализировать эргономические характеристики программных продуктов
ПК (У)-2.	Способен управлять программнотехническими, технологическими и человеческими ресурсами
ПК (У)-3.1	Способен управлять развитием БД
ПК (У)-3.2	Способен управлять работами по сопровождению и проектами создания (модификации) ИС, автоматизирующих задачи организационного управления и бизнеспроцессы
ПК (У)-4	Способен осуществлять руководство разработкой комплексных проектов навсех стадиях и этапах выполнения работ
ПК (У)-5	Способен проектировать и организовывать учебный процесс по образовательным программам с использованием современных образовательных технологий

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных систем и технологий
Направление подготовки 09.04.02 Информационные системы и технологии
Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП
_____ _____ Савельев А.О.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

Студенту:

Группа	ФИО
8ИМ01	Ким Станислав

Тема работы:

Проектирование и реализация адаптивного реактивного клиент-серверного приложения для сервиса доставки еды	
Утверждена приказом директора (дата, номер)	23.05.2022 г. № 143-32/с

Срок сдачи студентом выполненной работы:	

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Разработка направлена на проектирование и реализацию адаптивного реактивного клиент-серверного приложения для сервиса доставки еды
Перечень подлежащих исследованию, проектированию и разработке вопросов	Изучение документации о создании веб-приложений; Изучение и освоение проектных решений и инструментов для реализации веб-приложения; Реализация веб-приложения на фрэймворках Vue.js3 и Laravel9; Создание компонентов приложения; Создание и связывание REST API сервиса с клиентской частью:

Перечень графического материала	Сценарий использования; Диаграмма архитектуры приложения; Модель базы данных; Таблицы с используемыми API;

Консультанты по разделам выпускной квалификационной работы

Раздел	Консультант
Финансовый менеджмент	Былкова Татьяна Васильевна
Социальная ответственность	Федоренко ОльгаЮрьевна
Раздел на иностранном языке	Сидоренко Татьяна Валерьевна

Названия разделов, которые должны быть написаны на русском и иностранном языках:

Разделы на русском: проектирование приложения, реализация приложения, финансовый менеджмент, ресурсоэффективность и ресурсосбережение; социальная ответственность

Раздел на английском: Overview on technologies used in the project

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Копнов М. В.	к.т.н., доцент		

Консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОИТ ИШИТР	Коровкин В. А.			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ01	Ким Станислав		

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных систем и технологий
Направление подготовки 09.04.02 Информационные системы и технологии
Уровень образования Магистратура
Отделение школы (НОЦ) Отделение информационных технологий
Период выполнения 2021/2022 учебный год

Форма представления работы:

Магистерская диссертация

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
21.02.2022	Анализ предметной области	10
27.02.2022	Освоение проектных решений и инструментов	10
14.03.2022	Проектирование проекта	15
20.05.2022	Разработка проекта	35
26.05.2022	Тестирование проекта	10
03.06.2022	Финансовый менеджмент	10
03.06.2022	Социальная ответственность	10

СОСТАВИЛ:
Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Копнов М. В.	к.т.н., доцент		

Консультант ВКР:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОИТ ИШИТР	Коровкин В. А.			

СОГЛАСОВАНО:
Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Савельев Алексей Олегович	к.т.н., доцент		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА

«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8ИМ01	Ким Станиславу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Магистратура	Направление/специальность	Информационные системы и технологии

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Стоимость материальных ресурсов определялась по средней рыночной стоимости. Оклады в соответствии с окладами сотрудников организации.
2. Нормы и нормативы расходования ресурсов	Районный коэффициент - 30%
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Коэффициент отчислений на уплату во внебюджетные фонды 30%.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Оценка коммерческого и инновационного потенциала НТИ	Провести предпроектный анализ
2. Разработка устава научно-технического проекта	Представить Устав научного проекта магистерской работы
3. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	Разработать план управления НТИ
4. Определение ресурсной, финансовой, экономической эффективности	Определить интегральный финансовый показатель разработки Определить интегральный показатель ресурсоэффективности разработки Определить интегральный показатель эффективности
Перечень графического материала*:*	
1. Оценка конкурентоспособности технических решений *2.* Матрица SWOT *3.* Перечень работ и продолжительность их выполнения *4.* Оценка экономической эффективности НИ *5.* График проведения и бюджет НТИ *6.* Оценка ресурсной, финансовой эффективности НТИ	

Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Былкова Т. В.	к. э. н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ01	Ким Станислав		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ИМ01	Ким Станислав

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Магистратура	Направление/специальность	Информационные системы и технологии

Тема ВКР:

Проектирование и реализация адаптивного реактивного клиент-серверного приложения для сервиса доставки еды	
Исходные данные к разделу «Социальная ответственность»:	
Введение – Характеристика объекта исследования (вещество,материал, прибор, алгоритм, методика) и области его применения. – Описание рабочей зоны (рабочего места) приразработке проектного решения/при эксплуатации	Объект исследования: Адаптивное реактивное клиент-серверное приложение для сервиса доставки еды. Область применения: Предприятия, бизнес. Рабочее место. Оборудование, используемое на рабочем месте: персональный компьютер, дополнительный монитор, мышь, зарядное устройство компьютера. Помещение офисного типа имеет следующие характеристики: площадь 32 квадратных метра, естественное и искусственное освещение, естественная вентиляция, отопление батареями.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности <u>при эксплуатации</u>: – специальные (характерныепри эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	- Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ.(ред. От 24.04.2020). - СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредностидля человека факторов среды обитания. - ГОСТ 12.0.003-2015 Опасные и вредные производственные факторы. Классификация. Переченьопасных и вредных факторов. - ГОСТ 12.1.005-88 Система стандартов безопасности труда (ССБТ). Общие санитарно-гигиенические требования к воздуху рабочей зоны. - СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95. - СН 2.2.4/2.1.8.562-96. Шум на рабочих местах, в помещениях жилых, общественных зданий и натерритории жилой застройки.

	- ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности - ГОСТ 12.1.003-83 Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности. - ГОСТ 12.2.032-78 ССБТ Рабочее место при выполнении работ сидя. Общие эргономические требования. - ГОСТ Р 50923-96. Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения - ТОИ Р-45-084-01 Типовая инструкция по охране труда при работе на персональном компьютере. - ГОСТ Р 12.1.019-2017 ССБТ Электробезопасность. Общие требования и номенклатура видов защиты. - ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов. - ГОСТ 12.1.004-91 Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Общие требования. - ГОСТ 17.4.3.04-85 Охрана природы (ССОП). Почвы. Общие требования к контролю и охране от загрязнения.
2. Производственная безопасность <u>при эксплуатации</u>: – Анализ выявленных вредных и опасных производственных факторов – Расчет уровня опасного или вредного производственного фактора	Вредные факторы: - Повышенная или пониженная температура и относительная влажность воздуха. - Превышение уровня шума. - Отсутствие или недостаток освещения. - Психофизиологические факторы Опасные факторы: - Поражение электрическим током; - Короткое замыкание; - Статическое электричество. Средства индивидуальной защиты: - Маски; - Перчатки; - Очистители кожи; - Спиртовые салфетки для обработки ПК. Средства коллективной защиты: - Средства защиты от поражения электрическим током; - Средства защиты от повышенного уровня статического электричества; - Средства защиты от воздействия биологических факторов Расчет по фактору искусственного освещения.

3. Экологическая безопасность <u>при эксплуатации</u>	Разработка не оказывает влияние на атмосферу и гидросферу. Селитебная зона – должно быть обеспечено выполнение требований в области охраны окружающей среды. Негативное воздействие на литосферу происходит при утилизации компьютера и периферийных устройств (принтеры, МФУ, веб-камеры, наушники, колонки, телефоны), люминесцентных ламп, макулатуры.
4. Безопасность в чрезвычайных ситуациях <u>при эксплуатации</u>	- Возможные ЧС: пожары, грозы, ураганы, оползни. - Наиболее типичная ЧС: пожар.

Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Группа	ФИО	Подпись	Дата
Профессор ООД ШБИП	Федоренко Ольга Юрьевна		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ01	Ким Станислав		

РЕФЕРАТ

Выпускная квалификационная работа содержит 92 страницы, 19 рисунков, 20 таблиц, 35 источников, 3 приложения.

Работа посвящена разработке пользовательского адаптивного реактивного клиент-серверного приложения на фреймворке Vue.js для сервиса доставки еды.

Ключевые слова: VUE, JavaScript, SPA, REST API, веб-клиент.

Объектом исследования является разработка веб-приложения.

Предметом исследования является процесс разработки клиент-серверного приложения.

Цель работы – проектирование и реализация адаптивного реактивного клиент-серверного приложения для сервиса доставки еды.

В результате исследования были спроектировано и реализовано веб-приложение согласно выявленным требованиям и архитектуре системы.

Степень внедрения: частичная.

Область применения: сервис по доставке еды.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В данной работе применены следующие обозначения и сокращения:

SPA – Single Page Application;

API – Application Programming Interface;

REST – Representational State Transfer;

HTTP – HyperText Transfer Protocol;

JSON – JavaScript Object Notation;

БД – база данных.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

С каждым годом темп жизни людей увеличивается. Одной из проблем, с которой сталкиваются многие, становится нецелесообразная трата времени на осуществление заказа еды. Данная проблема актуальна для людей большинства профессий, у которых время на обеденный перерыв ограничено, и скорость приготовления блюд является важнейшим критерием выбора заведения для обеда. Однако из-за больших очередей и долгого ожидания приготовления блюд существует риск не уложиться по времени.

Решением данной проблемы может стать возможность предварительного заказа еды, который позволит получить уже приготовленное блюдо без долгого ожидания. Поэтому все сервисы сферы питания стараются внедрить в свои бизнес-процессы возможность создания заказа онлайн. Такой способ позволяет, с одной стороны, пользователям избегать очередей и экономить время, а другой, экономить время сервисов на приеме заказов.

Разрабатываемый проект предназначен для предварительного заказа еды в сети кафе. Потребителя приложения можно разделить на две категории: бизнес как потребитель и конечные пользователи, осуществляющие заказы еды. Основным сегментом потребителей, среди конечных пользователей, являются люди, которые имеют ограниченное время на прием пищи, что не позволяет стоять в очередях и тратить время. Разрабатываемый проект позволяет уменьшить время на реализацию заказа, что приводит к сокращению времени у клиента на получение заказа. Второй группой потребителя приложения рассматривается бизнес. Для бизнеса разрабатываемый проект решает следующие проблемы: собирает статистику и отчетность по каждому филиалу сервиса и позволяет проследить динамику изменения продаж.

При реализации проекта должны быть изучены и использованы современные подходы к проектированию Single Page Application (SPA) веб-приложений. Значение термина SPA кроется внутри него самого. SPA — это приложение, которое размещается всего на одной веб-странице с целью обеспечения более гибкого взаимодействия с пользователем, аналогичного настольному приложению, которое загружается за один раз, в отличии от традиционных многостраничных подходов в разработке веб-проектов. Это экономит большое количество времени загрузки проекта.

В ходе данного проекта должна быть реализована архитектура и разработка адаптивного SPA-приложения – сервиса по доставке еды с наличием:

- возможности получения и отправки данных из базы (название секций, информация о товарах, информация о пользователях);
- секций, которые позволяют распределить товары по категориям, и их редактированием;
- корзины, в которой в зависимости от способа получения товара появляются либо удаляются соответствующие поля;
- возможности добавления товаров в корзину и оформления заказа, с подсчитанной стоимостью, путем отправки данных на сервер;
- валидации данных форм;
- компонента для входа с проверкой в панель администрирования;
- компонента в виде таблиц для панели администрирования с данными о товарах, названиях секций, товарах определенной секции, администраторах и заказах, которые могут быть отредактированы.

Для достижения поставленной цели в рамках выпускной квалификационной работы необходимо решить следующие основные задачи:

1. создание маршрутов к компонентам приложения средствами фреймворка Vue.js;
2. провести работу данными, необходимыми для работы сервиса, посредством написания запросов при использовании библиотеки Axios;
3. использовать паттерн управления состоянием Vuex;
4. использовать библиотеку VeeValidate для валидации данных;
5. разработать REST API сервис для работы с клиентом;
6. разработать модель хранения данных для сервиса по доставке еды.

1. Проектирование приложения

1.1 Требования к проектируемому приложению

Правильно сформулированные и задокументированные требования необходимы для корректного видения концепции и возможностей приложения.

Для разрабатываемого приложения были сформулированы и выделены следующие общие требования:

- проект должен быть разработан на языке JavaScript ES6;
- в качестве фреймворка для написания клиента использовать Vue.js 3;
- клиентское приложение должно отображать данные о товарах, секциях, по которым распределены товары, контактные данные о сервисе по доставке еды, панели администрирования;
- клиентское приложение должно предоставлять возможность оформления заказа для различных вариантов;
- клиентское приложение должно предоставлять возможность способа выбора оплаты и получения заказа;
- в качестве фреймворка для написания REST API сервиса использовать Laravel 9;
- в качестве базы данных должна использоваться «MariaDb»;
- REST API сервис должен принимать HTTP-запросы с различными параметрами и отсылать ответы на них;
- получаемые данные должны проходить процесс валидации;
- данные в HTTP-запросах передаются в формате JSON.

В рамках выпускной квалификационной работы реализуются следующие компоненты:

- основная контентная часть с наличием товаров;

– основная контентная часть с наличием секций, по которым распределены товары;

– страница с контактными данными;

– страницу с картами, для предоставления информации о местоположении сервисов;

– корзина товаров;

– форма для оформления заказа с возможностью переключения между способами получения заказа и способами оплаты;

– регистрация, авторизация, аутентификация;

– панель администрирования с данными, о товарах, названиях секций, товарах определенной секции, администраторах и заказах, которые могут быть добавлены, удалены и отредактированы;

– REST API сервис.

Для описанных выше компонентов был составлен следующий перечень функциональных требований:

– клиентское приложение должно отображать информацию о доступных секциях товаров;

– клиентское приложение должно отображать информацию о доступных товарах соответствующей секции;

– клиентское приложение должно предоставлять возможность добавления и удаления данных о названии, изображении, весе, описании, стоимости, количестве товаров в корзине;

– клиентское приложение должно предоставлять возможность изменения количества товаров в корзине;

– клиентское приложение должно предоставлять возможность оформления заказа;

– клиентское приложение должно предоставлять возможность выбора способа получения и оплаты заказа;

– REST API сервис должен обеспечивать возможность регистрации, входа пользователя в аккаунт и выход из него;

– REST API сервис должен использовать систему ролевого разграничения прав для предоставления доступа пользователям к методам;

– REST API сервис должен иметь следующие роли:

1. пользователь;
2. администратор;

– в качестве системы аутентификации должны использоваться средства «Laravel Sanctum»;

– в качестве средства аутентификации должен использоваться «Bearer Token»;

– REST API сервис должен обеспечивать следующие возможности работы с пользователями:

1. регистрация администратора;
2. вход администратора в систему;
3. выход администратора из системы;
4. возвращать список администраторов;

– REST API сервис должен обеспечивать следующие возможности работы с контентом:

1. получать названия секций;
2. получать данные о товарах;
3. получать данные с товарами, которые соответствуют определенной секции;
4. создавать новый продукт;
5. обновлять данные о продукте:
6. удалять данные о продукте;
7. создавать новую секцию;
8. обновлять данные о секции;
9. удалять созданные секции;

– данные, переданные клиентской частью должны сохранятся на серверном хранилище;

– информация, переданная между клиентским приложением и REST API сервисом должна быть в виде JSON-файлов.

1.2 Сценарий использования

Сценарий использования – описание поведения системы при взаимодействии ее с кем-либо и чем-либо из внешней среды.

Наличие эффективных сценариев использования позволяет значительно упростить процесс последующей реализации приложения. Необходимые для реализации приложения сценарии представлены в таблице 1.

Таблица 1 – Сценарий использования

Сценарий 1. Предоставление информации о сервисе по доставке еды	
Действующие лица	Пользователь, клиентское приложение, REST API сервис
Цель	Получить и отобразить основную информацию о сервисе по доставке еды
Предусловие	Пользователь ввел адрес сервиса по доставке еды в поисковую строку браузера и перешел по нему.
Успешный сценарий: 1. Клиентское приложение отображает все необходимые данные о сервисе 2. Пользователю доступен весь функционал приложения	
Результат	Пользователь получил информацию о сервисе и ему доступен весь функционал приложения
Альтернативные сценарии: 3. Нестабильная работа хостинга 4. Нет доступа к базе данных	
Результат	Пользователь не получил информацию о сервисе и ему не доступен функционал приложения
Сценарий 2. Получение списка доступных товаров	
Действующие лица	Пользователь, клиентское приложение, REST API сервис

Цель	Получить данные о доступных товарах
Предусловие	Пользователь перешел в раздел секций с товарами
Успешный сценарий: 1. Клиентское приложение отправляет запрос на сервер с целью получить данные о всех доступных товарах. 2. REST API сервис возвращает клиентскому приложению необходимые данные	
Результат	Данные о доступных товарах успешно возвращены
Альтернативные сценарии: 3. Нет доступа к базе данных 4. REST API сервис возвращает сообщение о неудачной попытке получения данных.	
Результат	Данные о доступных товарах не были успешно возвращены
Сценарий 3. Получение списка доступных секций	
Действующие лица	Пользователь, клиентское приложение, REST API сервис
Цель	Получить данные о доступных секциях
Предусловие	Пользователь перешел в раздел секций с товарами
Успешный сценарий: 1. Клиентское приложение отправляет запрос на сервер с целью получить данные о всех доступных товарах. 2. REST API сервис возвращает клиентскому приложению необходимые данные	
Результат	Данные о доступных секциях успешно возвращены
Альтернативные сценарии: 3. Нет доступа к базе данных 4. REST API сервис возвращает сообщение о неудачной попытке получения данных.	
Результат	Данные о доступных секциях не были успешно возвращены
Сценарий 4. Добавление доступных товаров в корзину	
Действующие лица	Пользователь, клиентское приложение, REST API сервис
Цель	Пользователь: добавить товар в корзину
Предусловие	Пользователь инициировал добавление товара в корзину
Успешный сценарий: 1. Происходит передача данных о товаре с компонента секции в компонент корзины	
Результат	Товар добавлен в корзину
Альтернативные сценарии: 2. Получены не все данные о товаре	

Результат	Некорректное отображение данных о товаре в корзине
Сценарий 5. Изменение данных о товарах, добавленных в корзину	
Действующие лица	Пользователь, клиентское приложение, REST API сервис
Цель	Пользователь: изменить данные о товарах в корзине
Предусловие	Пользователь данные о товарах в корзине
Успешный сценарий: 1. Пользователь изменил количество товаров либо удалил товар из корзины	
Результат	Информация о количестве товаров в корзине успешно изменена, либо данные удалены из корзины
Альтернативные сценарии: 2. Получены некорректные данные о товаре	
Результат	Информация о количестве товаров в корзине ну была успешно изменена, либо данные не были удалены из корзины
Сценарий 6. Оформление заказа	
Действующие лица	Пользователь, клиентское приложение, REST API сервис
Цель	Оформить заказ с выбором способа получения и способа оплаты заказа, после чего получить данные о нем
Предусловие	Пользователь инициировал оформление заказа
Успешный сценарий: 1. Пользователь заполняет данные, необходимые для получения товара 2. Пользователь выбирает и заполняет поля, зависящие от выбора способа получения и оплаты заказа 3. Клиентское приложения проверяет данные формы методами валидации 4. Пользователь отправляет данные о заказе нажав на кнопку «сделать заказ»	
Результат	Данные о заказе получены и выведены, заказ успешно оформлен
Альтернативные сценарии: 5. Введенные данные не проходят валидацию 6. Серверное приложение возвращает сообщение о причине провала валидации 7. Нет доступа к базе данных 8. REST API сервис возвращает сообщение о неудачной попытке изменения данных.	
Результат	Данные о заказе не получены и не выведены, заказ не оформлен

Сценарий 7. Осуществление связи с сервисом по доставке	
Действующие лица	Пользователь, клиентское приложение
Цель	Связаться с сервисом по доставке
Предусловие	Пользователь нажал на кнопки, посредством которых можно перейти на социальные сети либо связаться с сервисом посредством мобильной связи.
Успешный сценарий: 1. Пользователь нажимает на кнопки с изображением социальных сетей сервиса 2. Пользователь нажимает на кнопку с номером телефона и связывается с сервисом посредством мобильного звонка	
Результат	Пользователь связался с сервисом по доставке
Альтернативные сценарии: 3. Страницы социальных сетей недоступны 4. Невозможность дозвониться до сервиса	
Результат	Пользователь не связался с сервисом по доставке
Сценарий 8. Получение информации о местоположении предложенных сервисов	
Действующие лица	Пользователь, клиентское приложение
Цель	Получить информацию о местоположении сервиса с применением карт
Предусловие	Пользователь перешел на страницу «Контакты»
Успешный сценарий: 1. Клиентское приложение отображает данные о сервисе и карты для получения данных о местоположении соответствующего сервиса. 2. Пользователь сориентирован в положении сервиса относительно своего положения	
Результат	Пользователь получил информацию о сервисе
Альтернативные сценарии: 3. Недоступность сервиса по отображению карт	
Результат	Пользователь не получил информацию о сервисе
Сценарий 9. Регистрация администратора	
Действующие лица	Пользователь, клиентское приложение, REST API сервис
Цель	Создать учетную запись администратора
Предусловие	Пользователь перешел на страницу «Вход и регистрация»

Успешный сценарий: 1. Пользователь открывает форму регистрации и заполняет необходимые поля. 2. Клиентское приложение отправляет данные на сервер 3. REST API сервис проверяет отправленные данные, записывает их в базу данных и возвращает приложению токен для аутентификации пользователя.	
Результат	Учетная запись администратора успешно создана
Альтернативные сценарии: 4. Введенные данные пользователя не проходят валидацию 5. REST API сервис возвращает сообщение о причине провала валидации 6. Нет доступа к базе данных 7. REST API сервис возвращает сообщение о неудачной попытке создания записи.	
Результат	Учетная запись администратора не создана
Сценарий 10. Авторизация пользователя	
Действующие лица	Пользователь, клиентское приложение, REST API сервис
Цель	Пользователь: войти на платформу и получить доступ к ее возможностям. REST API сервис: идентифицировать пользователя и его права
Предусловие	Пользователь зашел на сайт
Успешный сценарий: 1. Пользователь открывает форму входа и заполняет необходимые поля. 2. Клиентское приложение отправляет данные на сервер 3. REST API сервис проверяет отправленные данные и возвращает приложению токен для аутентификации пользователя.	
Результат	Пользователь успешно авторизирован и может работать с системой
Альтернативные сценарии: 4. Введенные данные пользователя не проходят валидацию 5. REST API сервис возвращает сообщение о причине провала валидации 6. Нет доступа к базе данных 7. REST API сервис возвращает сообщение о неудачной попытке создания записи и записывает данные об ошибке в файл журнала.	
Результат	Учетная запись пользователя не создана

Сценарий 11. Выход администратора из аккаунта	
Действующие лица	Администратор, клиентское приложение, REST API сервис
Цель	Администратор: выйти из аккаунта на платформе REST API сервис: обновить информацию о администраторе
Предусловие	Администратор инициировал выход из платформы
Успешный сценарий: 1. Клиентское приложение отправляет запрос на сервер 2. REST API сервис удаляет все токены аутентификации пользователя	
Результат	Администратор успешно вышел из аккаунта, его токены аутентификации удалены
Альтернативные сценарии: 3. Нет доступа к базе данных 4. REST API сервис возвращает сообщение о неудачной попытке изменения данных.	
Результат	Администратор не вышел из аккаунта
Сценарий 12. Получение списка администраторов	
Действующие лица	Пользователь, клиентское приложение, REST API сервис
Цель	Администратор: получить список пользователей
Предусловие	Администратор инициировал получение списка пользователей
Успешный сценарий: 1. Клиентское приложение отправляет запрос на сервер с параметрами о количестве записей. 2. REST API сервис проверяет отправленные данные и возвращает клиентскому приложению необходимый список пользователей	
Результат	Список администраторов отправлен на клиентское приложение
Альтернативные сценарии: 3. Введенные данные администратора не проходят валидацию 4. REST API сервис возвращает сообщение о причине провала валидации 5. Нет доступа к базе данных 6. REST API сервис возвращает сообщение о неудачной попытке изменения данных.	
Результат	Список администраторов не отправлен

Сценарий 13. Добавление, удаление и редактирование данных о товаре	
Действующие лица	Администратор, клиентское приложение, REST API сервис
Цель	Администратор: добавить новый товар либо обновить или удалить информацию о них
Предусловие	Администратор инициировал добавление нового товара либо обновление или удаление данных о товаре
Успешный сценарий: 1. Клиентское приложение получило необходимые данные о товаре 2. Клиентское приложение отправляет данные серверному приложению 3. REST API сервис проверяет отправленные данные и вносит необходимые изменения	
Результат	Добавлены, удалены или обновлены данные о товаре
Альтернативные сценарии: 4. Отправленные данные не проходят валидацию 5. REST API сервис возвращает сообщение о причине провала валидации 6. Нет доступа к базе данных 7. REST API сервис возвращает сообщение о неудачной попытке получения данных.	
Результат	Внесенные данные о товаре не были добавлены, удалены или обновлены
Сценарий 14. Добавление, удаление и редактирование данных о секциях	
Действующие лица	Администратор, клиентское приложение, REST API сервис
Цель	Администратор: добавить новую секцию либо обновить или удалить информацию о ней
Предусловие	Администратор инициировал добавление новой секции либо обновление или удаление данных о ней
Успешный сценарий: 1. Клиентское приложение получило необходимые данные о секции 2. Клиентское приложение отправляет данные серверному приложению 3. REST API сервис проверяет отправленные данные и вносит необходимые изменения	
Результат	Добавлены, удалены или обновлены данные о секции
Альтернативные сценарии: 4. Отправленные данные не проходят валидацию	

5. Нет доступа к базе данных 7. REST API сервис возвращает сообщение о неудачной попытке получения данных.	
Результат	Данные о секции не добавлены, удалены либо изменены
Сценарий 15. Добавление и удаление товаров из секций	
Действующие лица	Администратор, клиентское приложение, REST API сервис
Цель	Администратор: добавить или удалить товар из секции
Предусловие	Администратор инициировал добавление или удаление товара из секции
Успешный сценарий: 1. Клиентское приложение получило необходимые о товаре из секции 2. Клиентское приложение отправляет данные серверному приложению 3. REST API сервис вносит необходимые изменения	
Результат	Добавлены или удалены товары из секции
Альтернативные сценарии: 4. Нет доступа к базе данных 5. REST API сервис возвращает сообщение о неудачной попытке изменения данных.	
Результат	Не добавлены или не удалены товары из секции
Сценарий 16. Просмотр сведений о заказе	
Действующие лица	Администратор, клиентское приложение, REST API сервис
Цель	Администратор: просмотреть сведения о полученном заказе
Предусловие	Администратор инициировал просмотр таблицы с данными о заказах
Успешный сценарий: 1. Клиентское приложение отправило данные о заказе 2. REST API сервис получил данные о заказе 3. Клиентское приложение отобразило данные о заказе в таблице заказов	
Результат	Администратор просмотрел данные о заказе
Альтернативные сценарии: 4. Нет доступа к базе данных 5. REST API сервис возвращает сообщение о неудачной попытке получения или отправки данных.	
Результат	Администратор не просмотрел данные о заказе

2. Реализация приложения

2.1 Используемые технологии

2.1.1 HTML

HTML также известен, как стандартизированный язык разметки документов. Данный язык применяется для задания структуры веб-страницы и отображения контента. Описание разметки подавляющего большинства сайтов осуществляется с использованием данного языка. Язык понятен браузерам; текст, который интерпретируется в результате, выводится на экраны устройств [1].

2.1.2 CSS

CSS — дословно расшифровывается как каскадные таблицы стилей, данный язык отвечает за визуальное представление документов.

В основном применение находит как средство оформления внешнего вида веб-страниц, написанных при помощи языка HTML.

CSS предоставляет возможность для определения шрифта, цвета, размеров, расположения каких-либо блоков и других задач оформления внешнего вида веб-страниц. Основной задачей, которую решает CSS, является разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой же веб-страницы. Именно это разделение делает разработку более структурированной, и соответственно более приятной к восприятию [2].

2.1.3 SCSS

Написание CSS, когда таблица стилей становится огромной, превращается в более сложный и трудоемкий процесс. И вот в таком случае используется препроцессор. SCSS позволяет использовать функции недоступные в самом CSS, например, переменные, вложенности, миксины,

наследование и другие приятные вещи, возвращающие удобство написания CSS [3].

2.1.3 JavaScript

Мультипарадигменный язык программирования. Это полноценный динамический язык программирования, который применяется к HTML документу, и может обеспечить динамическую интерактивность на веб-сайтах. Он не предоставляет низкоуровневых средств работы с памятью, процессором, так как изначально был ориентирован на браузеры, в которых это не требуется.

Что же касается остальных возможностей – они зависят от окружения, в котором запущен JavaScript. В браузере JavaScript умеет делать всё, что относится к манипуляции со страницей, взаимодействию с посетителем [4].

2.1.4 API

API (Application Programming Interface — «программный интерфейс приложения») — описание способов, используя которые, можно производить взаимодействия между двумя программами. Обычно входит в описание какого-либо интернет-протокола, программного каркаса или стандарта вызовов функций операционной системы. Часто реализуется отдельной программной библиотекой или сервисом операционной системы [5]. Существуют несколько видов API. В данном проекте реализуется работа с REST API. REST API взаимодействует при помощи HTTP запросов, выполняя стандартные функции: создание, обновление, чтение, удаление записей в ресурсе.

2.1.5 Axios

При необходимости получать и отображать данные из API используется Axios – это библиотека с открытым исходным кодом, позволяющая делать HTTP-запросы [6].

2.1.6 Vue.js

Vue — это прогрессивный фреймворк для создания пользовательских интерфейсов [7]. Vue.js можно внедрять постепенно. Это означает, что при переходе к другому проекту необязательно переписывать его полностью. этим он отличается от своих аналогов. Его ядро в первую очередь решает задачи уровня представления (view), что упрощает интеграцию с другими библиотеками и существующими проектами. Vue подходит для небольших проектов, которым нужно добавить немного реактивности. При всем этом, Vue легко масштабируется и полностью подходит для создания объемных проектов. Данный фреймворк используют такие компании как Alibaba, Twitter, Facebook и EuroNews.

2.1.7 Vue-router

Vue-Router — это пакет JavaScript, который позволяет настроить маршрутизацию для одностраничных приложений. Использование SPA имеет множество преимуществ, но одним из основных недостатков является то, что все компоненты веб-страницы доставляются, добавляются или удаляются с помощью JavaScript без загрузки дополнительных HTML-страниц с сервера. В этом суть SPA, но главная проблема заключается в возможности перемещаться по «страницам», к которым пользователи привыкли на большинстве веб-сайтов [8]. Эту задачу как раз-таки решает vue-router.

2.1.8 Vuex

Vuex — паттерн управления состоянием и библиотека для приложений на Vue.js [9]. Он является централизованным хранилищем данных для всех компонентов приложения с нормами, которые гарантируют, что состояние будет подвергнуто изменениям только предсказуемым образом.

2.1.9 VeeValidate

VeeValidate – библиотека для валидации, которая содержит в себе 35 встроенных валидаторов, перевод ошибок на английский язык, директивы и компоненты для валидации.

2.1.10 Laravel

Laravel – это бесплатный PHP фреймворк с открытым исходным кодом, для разработки веб-приложений по архитектурному шаблону Model-View-Controller. Ядро Laravel является надёжным с точки зрения производительности, возможно расширение платформы, используя множество дополнений. Laravel также прекрасно интегрируется с другими сторонними библиотеками и платформами, что позволяет создавать высокомасштабируемые приложения. Для долгосрочных задач имеется возможность поставить их в очередь для асинхронного выполнения в фоновом режиме, что ещё больше повышает производительность [10].

2.1.11 Laravel Sanctum

Laravel Sanctum предлагает легковесную систему аутентификации для SPA (одностраничных приложений), мобильных приложений и простых API на основе токенов. Sanctum позволяет каждому пользователю вашего приложения создавать несколько токенов API для своей учетной записи. Этим токенам могут быть предоставлены полномочия, которые определяют, какие действия токенам разрешено выполнять [11].

2.1.12 Swagger

Swagger — это набор инструментов, которые помогают описывать API. Благодаря ему пользователи и машины лучше понимают возможности REST API без доступа к коду [12]. С помощью Swagger можно быстро создать документацию и отправить ее другим разработчикам или клиентам.

2.2 Программная реализация компонентов

Фреймворк Vue опирается на компонентный подход. Все приложение делится на многократно используемые части – компоненты (рисунок 1). Компонент может представлять собой не только одну вещь, но и какую-то часть приложения, которая должна работать и выглядеть везде единообразно.

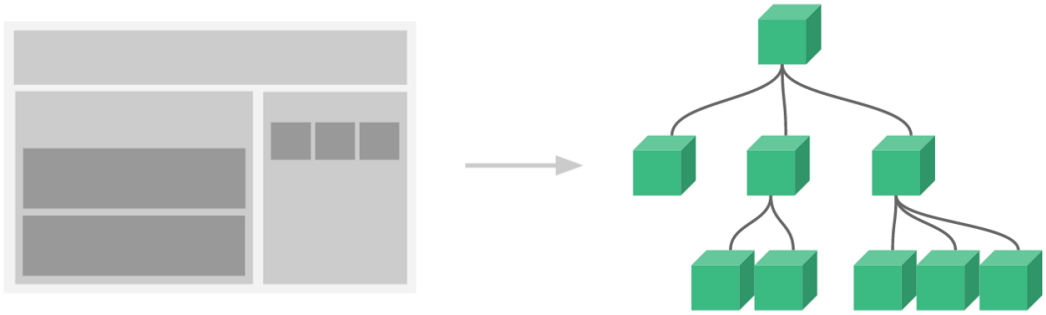

Рисунок 1 – Разделение страницы на компоненты

Компонентный подход предоставляет возможность избегания смешивания кода и чёткого выстраивания архитектуры приложения. Страницу, содержащую сложную структура и большое количество кода всегда можно разбить на более простые составляющие. Каждую из таких частей при выделении в компонент проще поддерживать, а при необходимости повторять разбиение внутри компонента на ещё меньшие части.

Для реализации архитектуры на Vue.js изначально она была спроектирована в виде следующей диаграммы (рисунок 2).

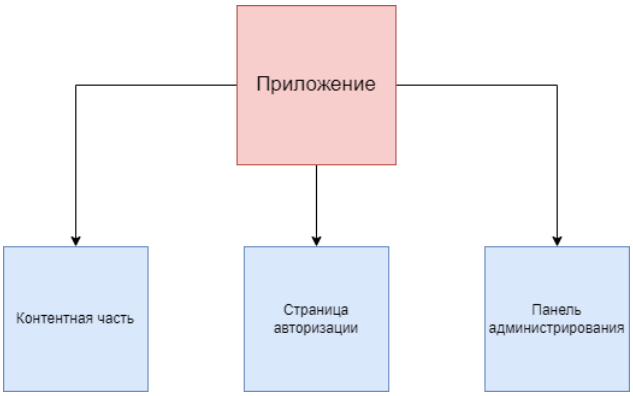

Рисунок 2 – Архитектура приложения

Далее архитектура разрабатываемого приложения была спроецирована на Vue.js (рисунок 3).

В проектируемом приложении есть корневой элемент App.vue. В котором находится компонент-шаблон AppLayout, в который в свою очередь

будут подставляться еще три шаблона страниц: для входа, основной секции и панели администрирования.

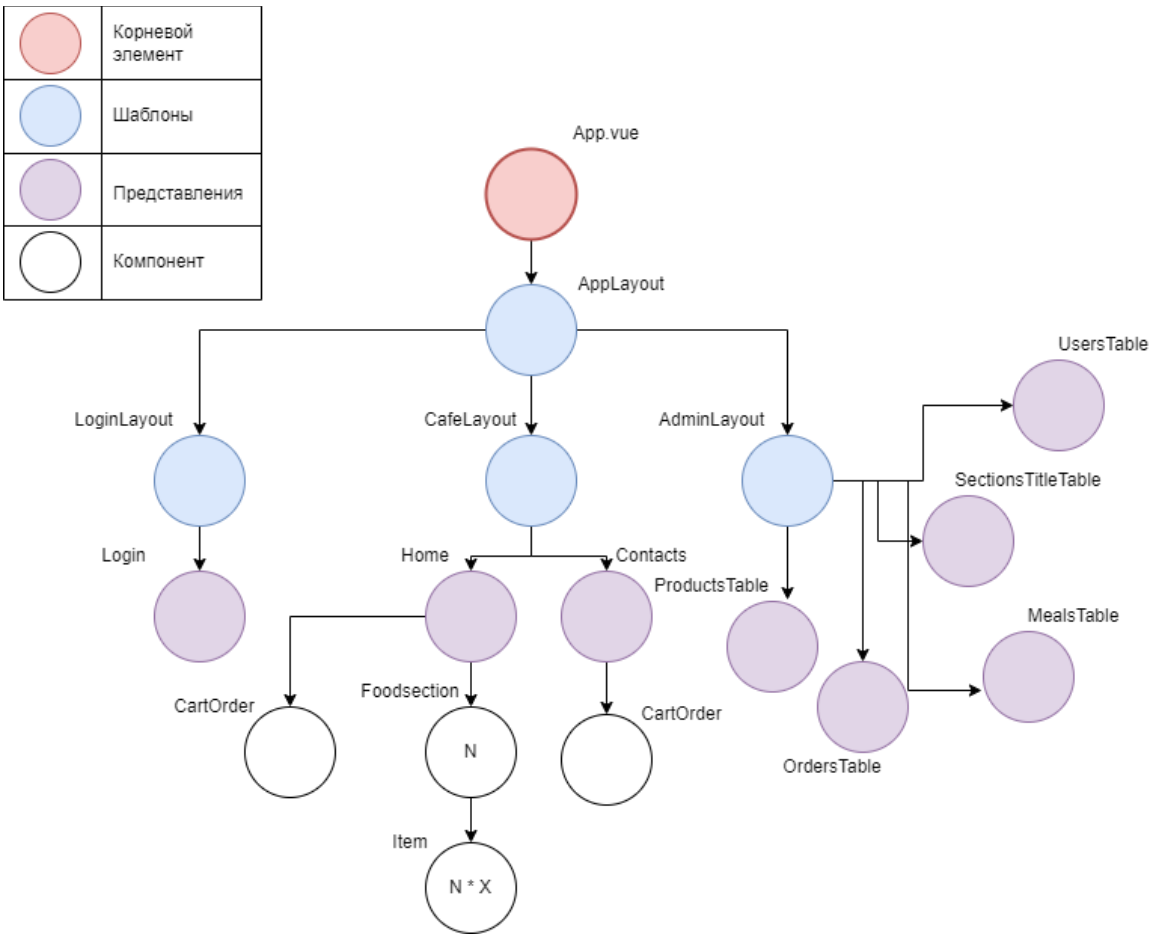

Рисунок 3 – Архитектура приложения

В каждом из шаблонов находятся соответствующие ему компоненты – представления. Как видно из диаграммы в представлении Home есть компонент Foodsection, который был описан единожды, но может быть вызван и использован нужное количество раз. В каждой Foodsection находятся компоненты Item, которые также повторно вызываются и используются соответствующее количество раз.

После проектирования архитектуры приложения следует процесс создания компонентов. Наличие нестандартных тегов в разметке является знаком присутствия компонентов (например, <foodsection></foodsection>). Однако, в отличии от обычного тега, компонент может содержать не только

необходимую разметку, но и своеобразную логику своей работы и использование других компонентов.

В начале следует создать динамический компонент-шаблон для всего приложения AppLayout, в который будут подставляться далее созданные компоненты-шаблоны (рисунок 4).

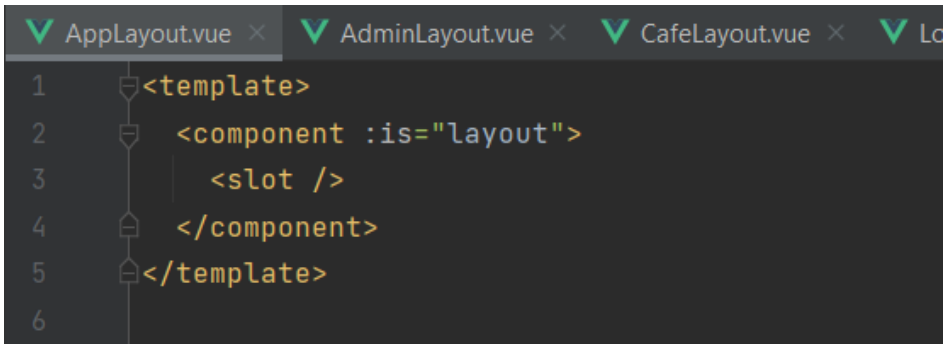

Рисунок 4 – Компонент AppLayout

Созданы три шаблона, помимо основного – это Cafe, Login и Admin layout. Они представляют собой обертку для трех основных типов страниц проекта, страницы кафе, логина и администрирования соответственно. Например, для компонентов Home, где находится основная контентая часть, и для компонента Contacts, где находится информация с контактами, должна быть одна обертка страницы – шаблон CafeLayout. Данный шаблон состоит из одинаковых верхней и нижней частей (header и footer, соответственно), которые должны отображаться только на двух вышеуказанных компонентах (рисунок 5-6).

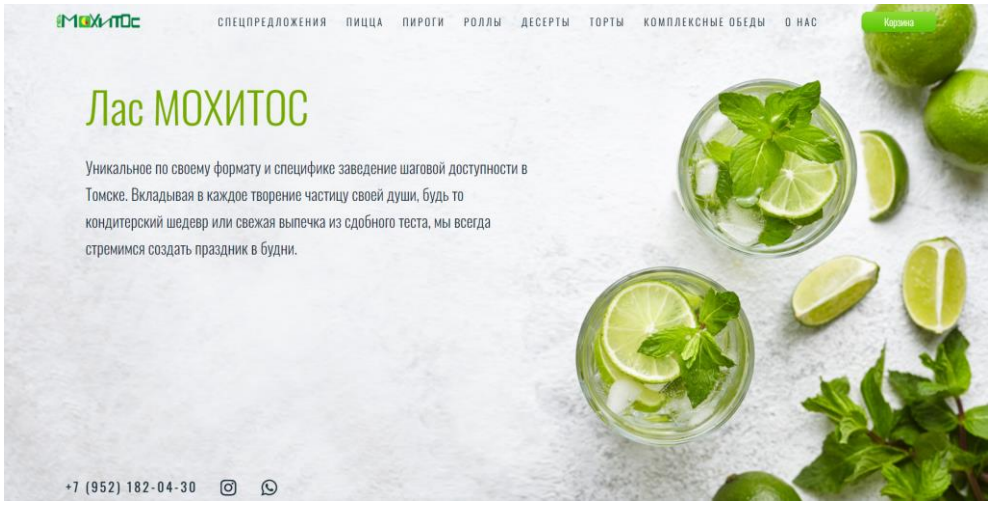

Рисунок 5 – компонент Home

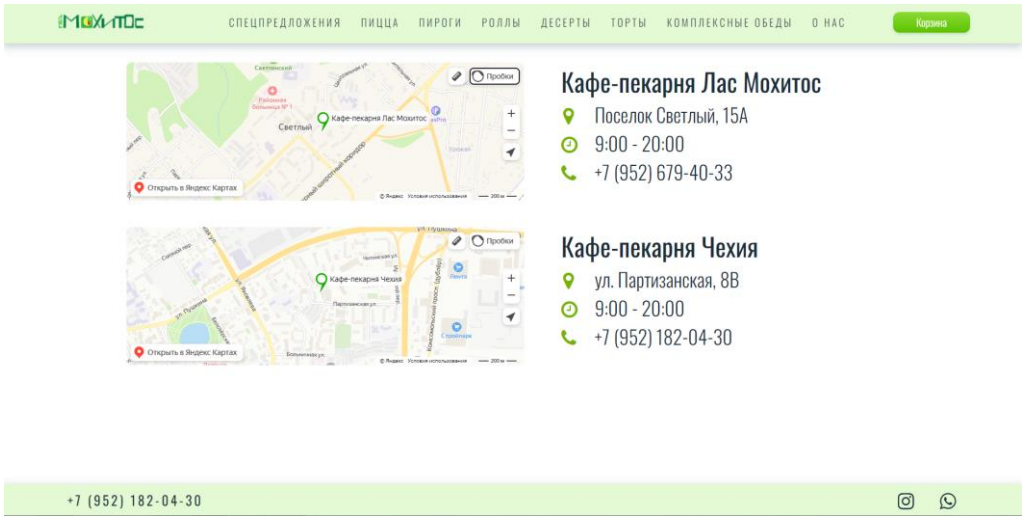

Рисунок 6 – компонент Contacts

Таким же образом были реализованы и использованы два оставшихся шаблона.

Для реализации работы с данными был использован Vuex – это библиотека, которая позволяет управлять состоянием. Она была создана командой Vue для предоставления возможности управления данными в проектах соответствующего фреймворка. Библиотека предоставляет централизованный способ управления данными, которые используются в приложении, и позволяет применять к ним необходимые манипуляции. Vuex создает хранилище, в которое входят состояния, геттеры, мутации и действия. Для того, чтобы обновить или изменить состояние, нужно совершить мутацию

– изменение исходной структуры данных. Для выполнения асинхронной задачи, нужно выполнить действие. Действие в случае корректной работы совершают мутацию данных, которая изменяет состояние, что влечет за собой обновление представление (рисунок 7).

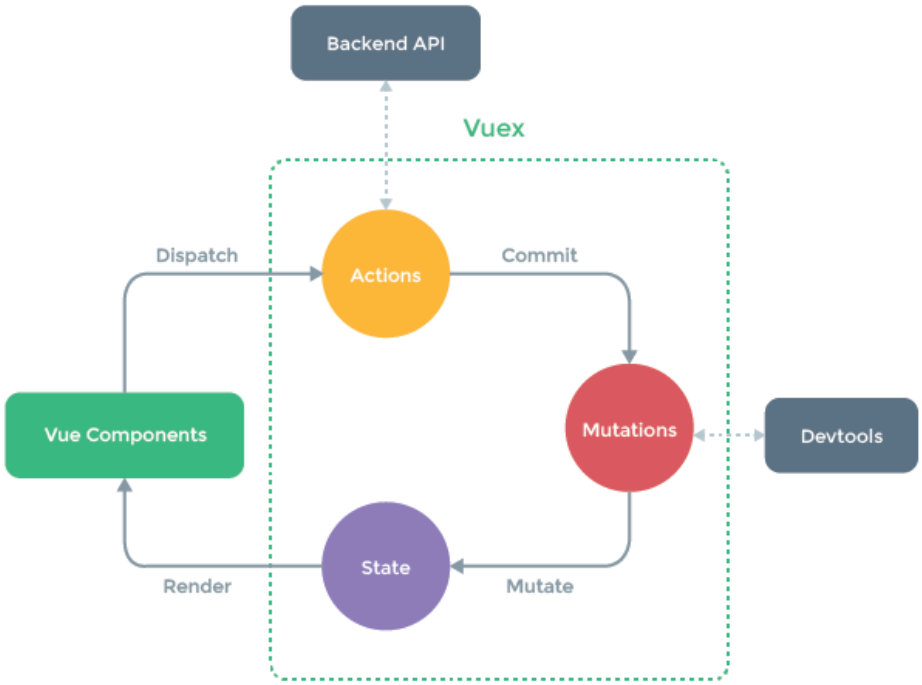

Рисунок 7 – стадии Vuex

Vuex не навязывает каких-то особых ограничений на используемую структуру кода. Но несмотря на это, библиотека требует соблюдения нескольких высокоуровневых принципов:

1. Глобальное состояние приложения должно находиться в глобальном хранилище;
2. Единственной возможностью изменения состояния является выполнение синхронных транзакций – мутации;
3. Асинхронные операции инкапсулируются в действия или их комбинации.

При соблюдении данных норм, предоставляется возможность использования любой структуры проекта. При значительном увеличении размеров хранилища, необходимо выносить действия, мутации и геттеры в отдельные файлы – модули.

Использование Vuex совсем не принуждает к вынесению всего состояния в хранилище. Однако перемещение большей части состояния во Vuex, позволяет сделать мутации более явными и удобными для отладки, но обратной стороной этого подхода является приведение к многословности и нецелесообразному усложнению логики. В случаях, когда часть состояния относится только к одному компоненту, логичнее оставить его в качестве локального состояния.

Для компонента Home нужно использовать повторяющийся компонент Foodsection. Количество его использований зависит от количества и названий категорий, которые хранятся на сервере. В процессе создания веб-приложения неоднократно появляется необходимость в получении и отображении данных из API. Существует несколько альтернативных способов для решения данного вопроса, к примеру, библиотеки Fetch, Supaeragent, Request. Но одним из популярных решений является использование axios, так как используя данную библиотеку, нет необходимости преобразовывать передаваемые данные в формат JSON, axios делает это сам. Посредством axios-запроса к базе данных, в которой хранятся названия секций и получая данные, подставляем их в соответствующее место компонента Foodsection (приложение Б). То есть, если на сервере хранятся 5 категорий, то данный компонент будет вызван соответствующее количество раз (рисунок 8-9).

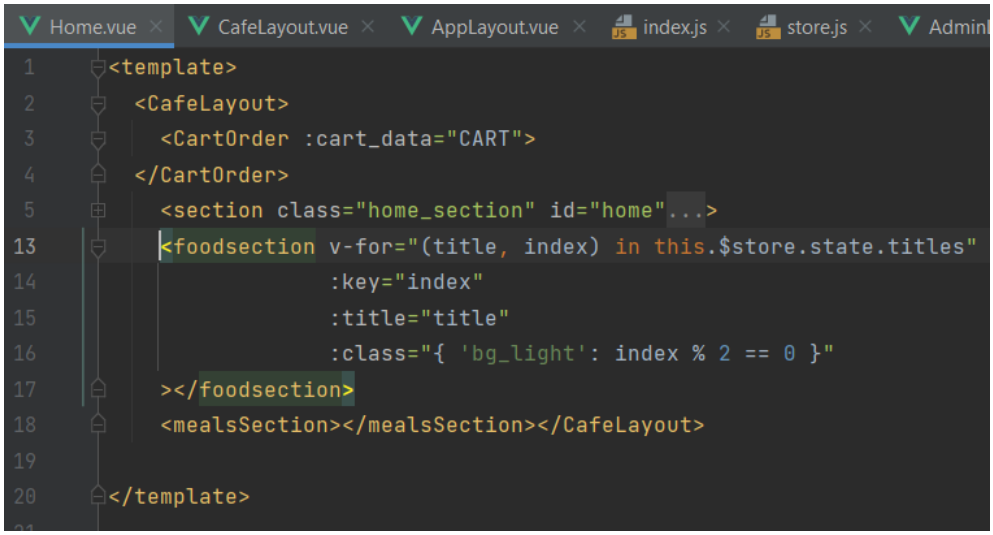

Рисунок 8 – использование компонента foodsection

Рисунок 9 – получение данных посредством Axios

Затем таким же образом происходит получение данных о товарах. Каждый товар представлен json-файлом, в котором хранятся свойства каждого товара. Получая их и используя строковую интерполяцию, свойства подставляются в нужные места компонента (приложение В).

Необходимыми компонентами для работы приложения являются корзина и форма для отправки данных о заказе. Компонент корзина при отсутствии в нем заказов выдает сообщение о том, что корзина пуста и перейти к форме отправки заказа невозможно. Также в корзине предоставляется возможность изменения количества добавленного товара.

В компоненте форма для заполнения данных о заказе указаны поля:

- имя;
- телефон;
- способ получения заказа: в кафе либо доставка;
- если выбран способ «в кафе», то предоставляется выбор названия заведения, для оформления заказа;
- если выбран способ «доставка», то появляются поля адрес и желаемое время доставки;
- способ оплаты заказа: наличные либо картой.

В проекте есть всего лишь одна HTML-страница, а необходимость в навигации между «страницами» веб-сайта остается. Для решения данной задачи был использован Vue-router. При помощи технологий фреймворка, происходит компоновка приложения из компонентов. Добавляя Vue Router, происходит сопоставление компонентов с маршрутами. После чего он понимает где их нужно отобразить. Пользовательский интерфейс реальных приложений обычно представлен многоуровневой иерархией компонентов. Столь же обычно и соответствие сегментов URL некоторой структуре вложенности компонентов.

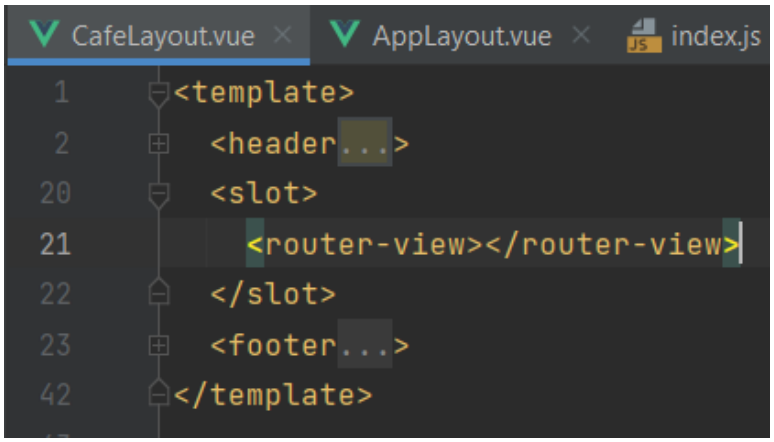

Рисунок 10 – компонент, использующий router-view

Здесь <router-view> — это точка, в которой будет отображён компонент, соответствующий маршруту верхнего уровня. Аналогичным образом,

отображаемый там компонент может и сам содержать вложенный <router-view> (рисунок 10).

Для отображения компонентов в этой вложенной точке, нам понадобится опция children в конфигурации конструктора VueRouter (рисунок 11).

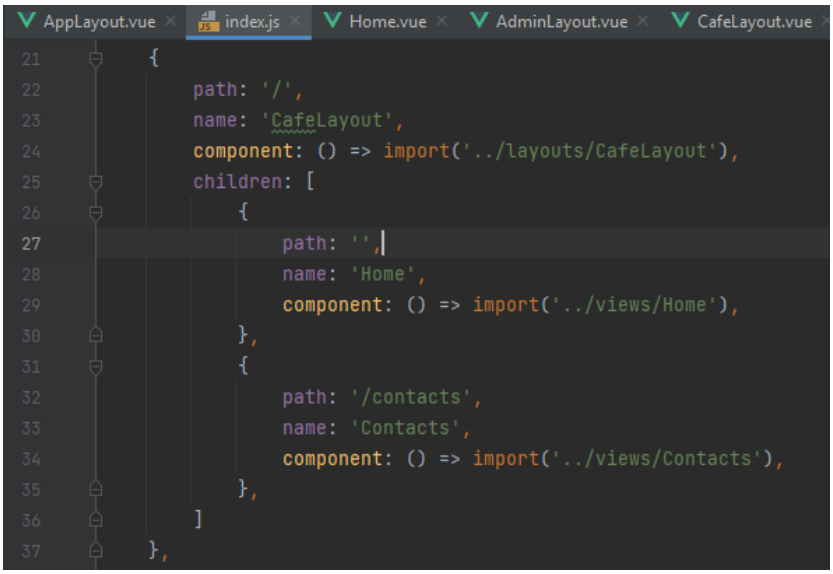

Рисунок 11 – указание адресов

Опция children принимает обычный массив объектов конфигурации маршрутов, такой же, как и сам routes. Таким образом, вложенность путей в теории по глубине ничем не ограничена.

Остальные пути были описаны по такому же принципу.

В шаблоне для аутентификации были применены методы библиотеки VeeValidate для проверки введенных данных (рисунок 12).

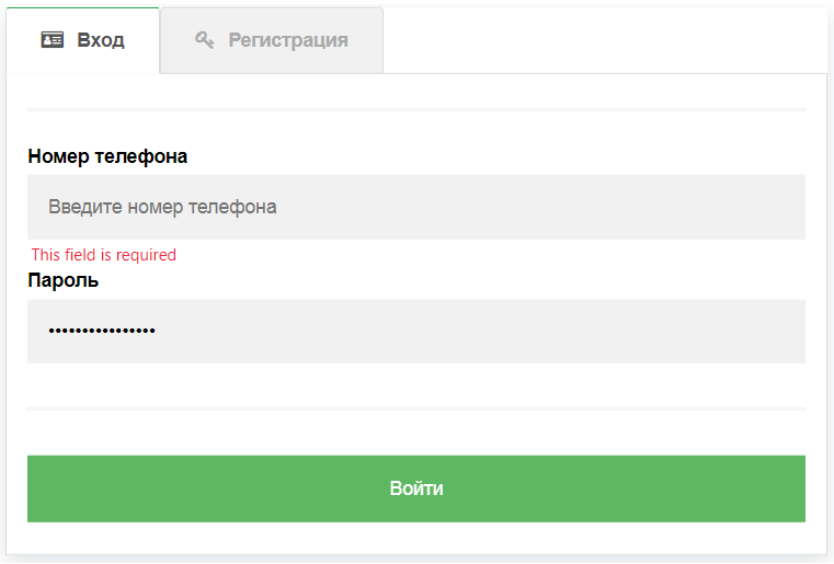

Рисунок 12– Валидация данных посредством VeeValidate

Также была разработана панель администрирования, в которой присутствуют динамически изменяющиеся таблицы, которые связаны с базой данных посредством axios-запросов (рисунок 13). В данных таблицах реализована возможность добавления, удаления и редактирования записей (рисунок 14).

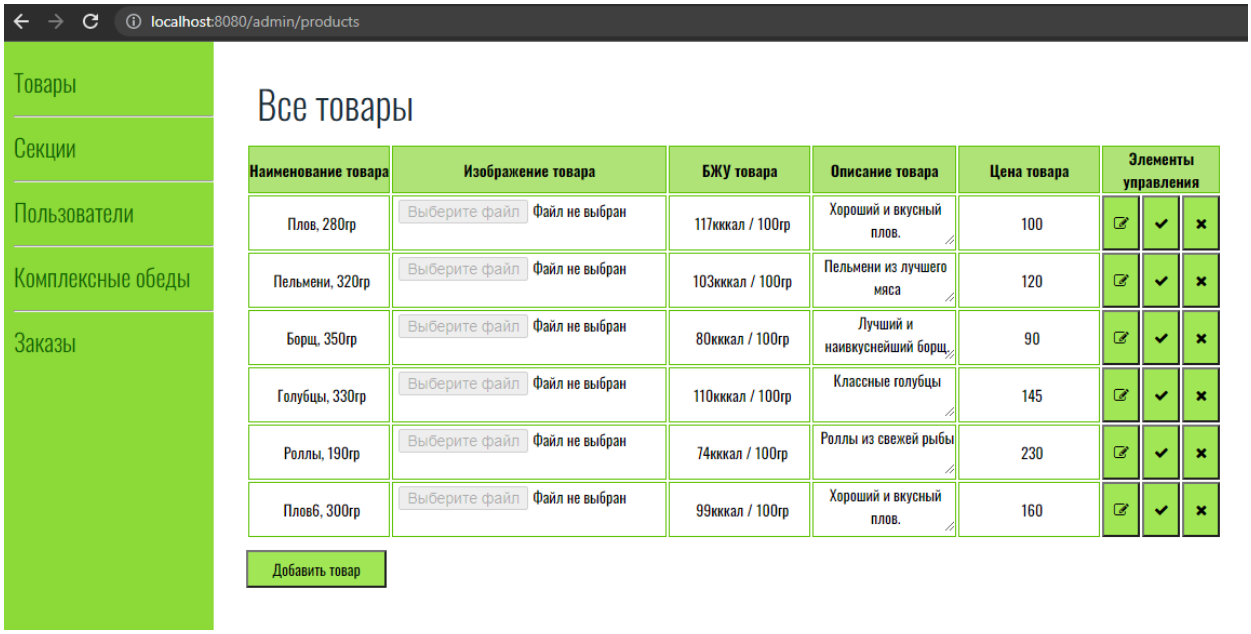

Рисунок 13 – Таблица «Товары»

Была создана форма «Добавление товара» с двусторонней привязкой данных посредством директивы v-model. Что означает, что данные, введенные в форму, сразу же добавляются в созданную модель.

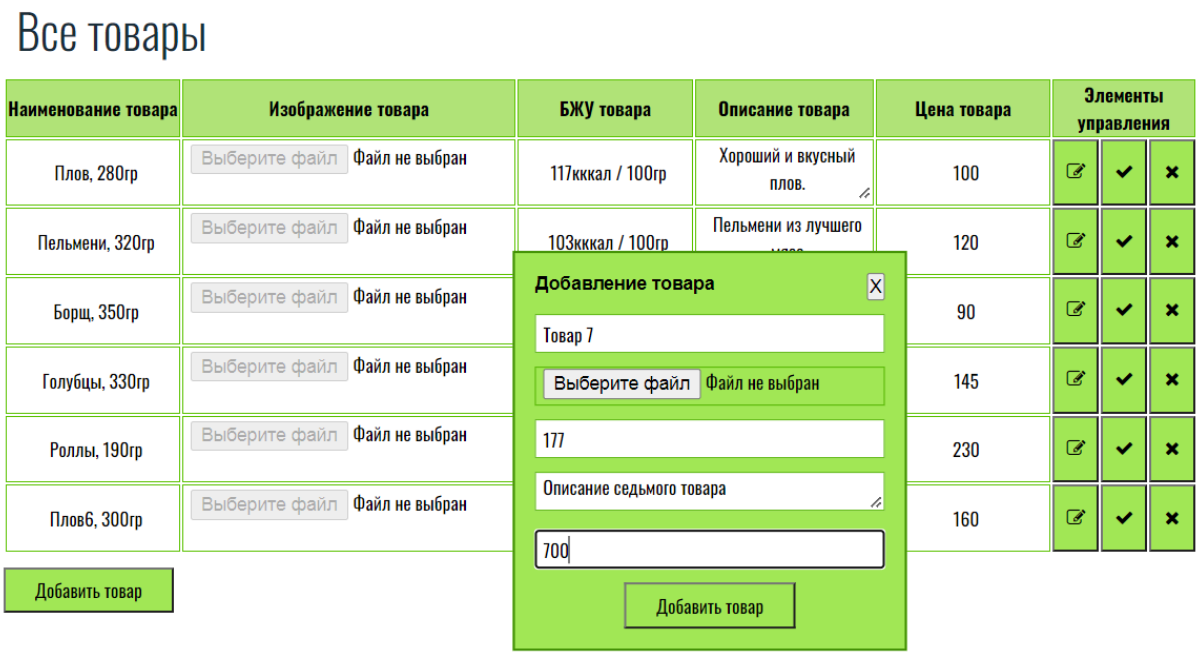

Рисунок 14 – Добавление товара

После нажатия на кнопку «Добавить товар» модель с данными формы отправляется в массив товаров.

Изначально все поля записи недоступны для редактирования. У каждого товара присутствует возможность изменения его данных.

Также реализована возможность удаления записи.

Данный проект должен автоматически подстраиваться под любые размеры экранов (рисунок 15-16). Количество переходов на сайт с мобильных устройств увеличивается с каждым годом, и чтобы эффективно обрабатывать этот трафик, нужно предлагать пользователю сайты под все типы устройств с удобным интерфейсом. Ввиду этого была переорганизована структура проекта и реализована адаптивная верстка посредством написания media-запросов.

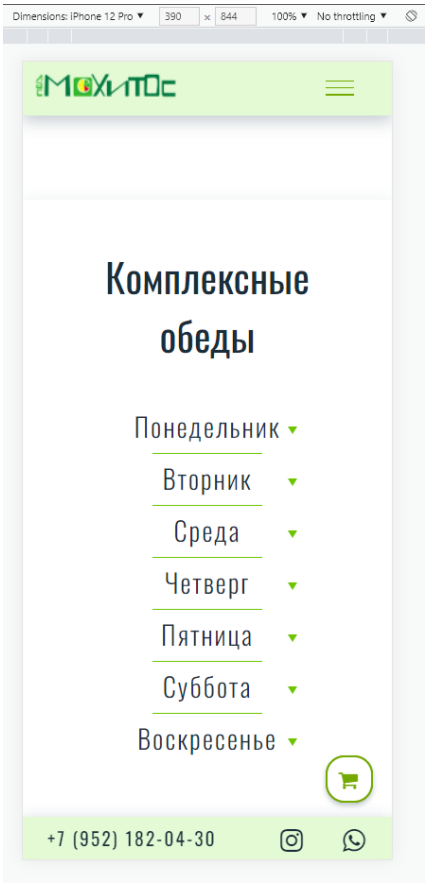

Рисунок 15– Отображение проекта на Iphone 12 Pro

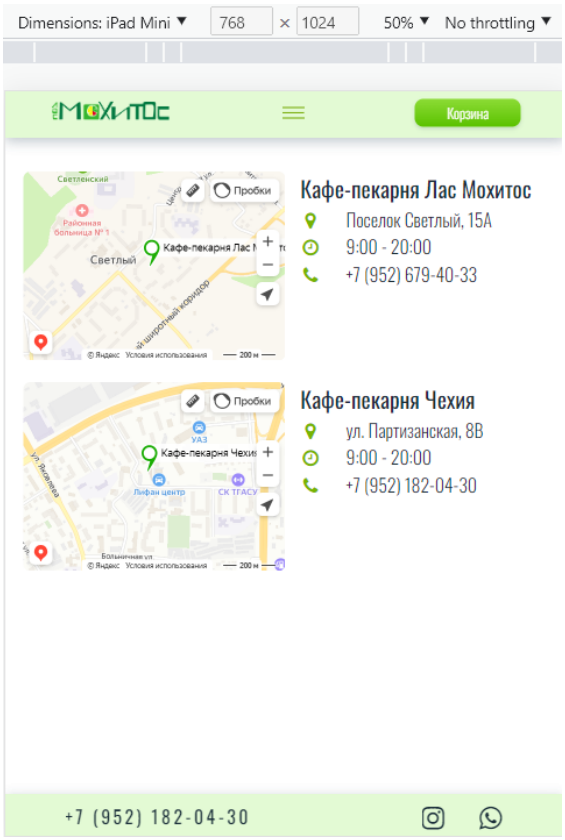

Рисунок 16 – Отображение проекта на Ipad mini

2.3 Реализация REST API сервиса

В рамках работы был разработан REST API сервис на основе фреймворка Laravel 9. REST API сервис — это легкий, обслуживаемый и масштабируемый сервис, построенный на архитектуре REST. Веб-служба REST предоставляет вызывающему клиенту API из приложения безопасным, единообразным способом без сохранения состояния. Вызывающий клиент может выполнять предопределенные операции, используя сервис REST. Основным протоколом для REST является HTTP.

Была разработана база данных при использовании системы управления реляционными базами данных MariaDB (рисунок 17).

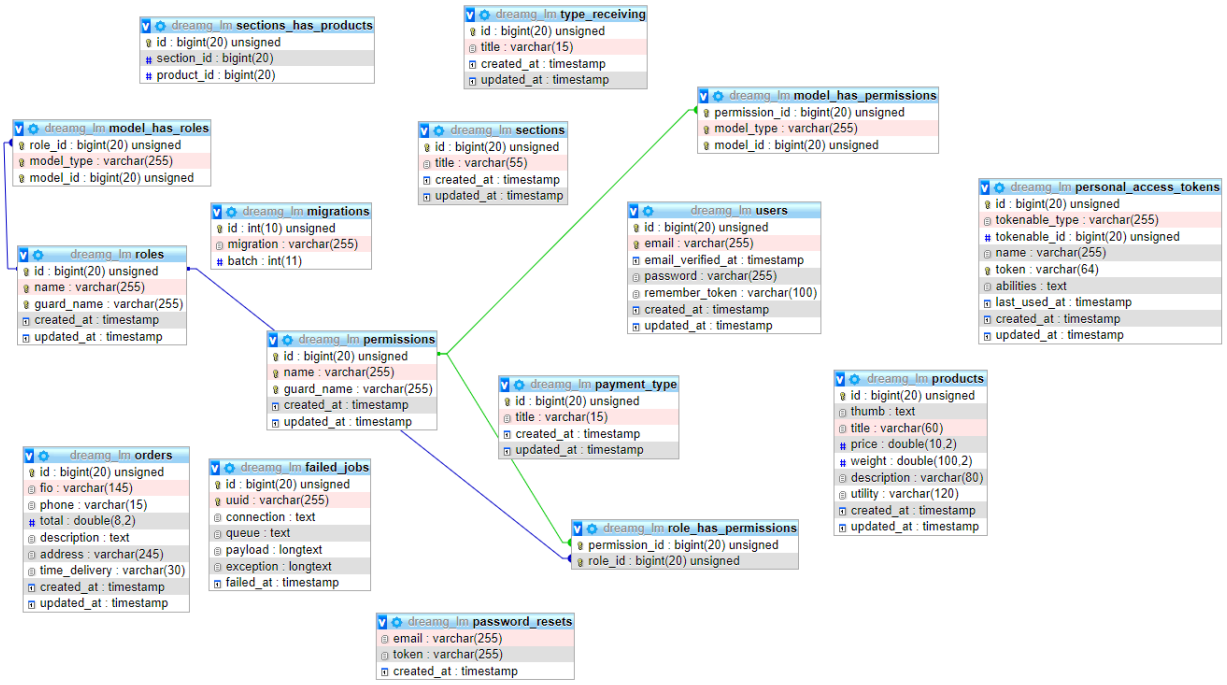

Рисунок 17 – База данных

В настоящее время наиболее популярным форматом веб-сервисов является REST API. Этот формат заключается в том, что сервер предоставляет разработчикам специальный файл, написанный на языке Open API, в котором описаны функции, которые может выполнять данный сервер. В файле описан формат всех возможных запросов, перечень параметров каждого запроса,

формат ответа на каждый запрос, подробное описание всех возвращаемых данных.

Был использован инструмент Swagger, с помощью которого можно визуализировать интерфейс, описанный в файле, и попытаться выполнить запросы с помощью веб-страницы. Разработанные API указаны в таблице 2.

Таблица 2 – Используемые API

API	Метод	Назначение
/v1/database.getsections	GET	Получение названий секций
/v1/database.getproducts	GET	Получение данных о товарах
/v1/database.getproduct	POST	Передача id товара секции
/v1/registration	POST	Регистрация пользователя в системе
/v1/login	POST	Вход пользователя в систему
/v1/logout	POST	Выход пользователя из системы

После авторизации доступны следующие методы для добавления, обновления и удаления информации (таблица 3).

Таблица 3 – API доступные после авторизации

API	Метод	Назначение
/v1/product.create	POST	Создание нового продукта
/v1/product.update	PUT	Обновление данных о продукте
/v1/product.delete	DELETE	Удаление созданного продукта
/v1/section.create	POST	Создание новой секции
/v1/section.update	PUT	Обновление данных о секции
/v1/section.delete	DELETE	Удаление созданной секции
/v1/user	POST	Получение информации о текущем пользователе
/v1/order.create	POST	Отправка данных о заказе

3. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

3.1 Предпроектный анализ

Разрабатываемый проект предназначен для автоматизации предварительного заказа еды в сети кафе. Потребителей приложения можно разделить на две категории: бизнес как потребитель и конечные пользователи, осуществляющие заказы еды. Основным сегментом потребителей, среди конечных пользователей, являются люди, которые имеют ограниченное время на прием пищи, что не позволяет стоять в очередях и тратить время. Разрабатываемый проект позволит сократить время на обслуживание заказа, что приведет к сокращению времени у пользователя на получение заказа. Второй группой потребителей приложения рассматривается бизнес. Для бизнеса разрабатываемый проект решает следующие проблемы: собирает статистику и отчетность по каждому кафе и позволяет проследить динамику изменения продаж, предоставляет конечному пользователю информацию о точках продаж.

В качестве конкурентных проектов были выбраны веб-приложения сервисов по доставке еды «Хинкали-Гали», «АланаМама» и «Ням-Ням».

Для оценки конкурентоспособности следует составить оценочную карту (таблица 3.1). Анализ конкурентных технических решений определяется по формуле:

$$К = \sum В_i \cdot Б_i, \quad (1)$$

где К – средневзвешенное значение показателя качества и перспективности научной разработки;

$В_i$ – вес показателя (в долях единицы);

$Б_i$ – средневзвешенное значение i-го показателя.

Таблица 3.1 – Оценочная карта конкурентных технических решений

Критерии оценки	Вес критерия	Баллы		Конкурентоспособность	
		Разработанный сервис по доставке еды	Сервис по доставке еды «Хинкали-Гали»	Сервис по доставке еды «АланаМама»	Сервис по доставке еды «Ням-Ням»
Удобство в эксплуатации	0,2	8	7	6	7
Качество интерфейса	0,15	7	4	5	6
Простота эксплуатации	0,15	9	7	8	7
Кроссплатформенность	0,15	9	7	8	8
Функциональность	0,15	7	6	8	8
Предполагаемый срок использования	0,1	9	9	9	9
Конкурентоспособность продукта	0,1	8	6	7	8
Итого	1	0,81	0,65	0,715	0,745

Значение показателя К является 81%, что позволяет говорить о перспективах разработки и качестве проведенного исследования.

Для выявления достоинств и недостатков продукта с целью улучшения качества следует провести SWOT анализ (таблица 3.2).

Таблица 3.2 – SWOT анализ

	Внутренние факторы		
Внешние факторы		Сильные стороны: Интуитивно понятный интерфейс Адаптивность под все виды устройств. Возможность по расширению приложения	Слабые стороны: Требовательность к системе хранения данных.
	Возможности: Сбор статистики и отчетности. Прослеживание динамики изменения продаж Увеличение клиентской базы.	Проект предоставляет все данные, необходимые для повышения конкурентоспособности сервиса по доставке еды, что соответственно приводит к увеличению числа потребителей.	Реализовать маркетинговый план по привлечению аудитории.
	Угрозы: Сбои или перебои на серверах (неработоспособность приложения).	Неработоспособность приложения – это лишь потенциально возможная ситуация, т.к. «падение» серверов очень редкое явление.	Приобретение более надежного хостинга.

Для улучшения качества продукта можно принять некоторые меры. Такие как постоянное совершенствование, внедрение новых технологий и поддержка проекта.

Угрозой, на которую практически невозможно повлиять, являются сбои или перебои на серверах. Решением данной проблемы является переход на более надежный сервер.

3.2 Планирование управления научно-техническим проектом.

Для организации и систематизации работы выпускника был сформирован план работ. Данный этап обеспечил своевременное и эффективное выполнение задания выпускной квалификационной работы. Для осуществления разработки, был сформирован ряд работ и назначены исполнители для каждого этапа работы (таблица 3.3).

Таблица 3.3 – Структура работ в рамках научного исследования

№	Наименование работы	Исполнители работы
1	Выбор научного руководителя магистерской работы	Ким С.В.
2	Составление и утверждение темы магистерской работы	Коровкин В.А, Ким С.В.
3	Составление календарного плана-графика выполнения магистерской работы	Коровкин В.А
4	Подбор и изучение литературы по теме магистерской работы	Ким С.В.
5	Анализ предметной области	Ким С.В.
6	Проектирование веб-приложения	Ким С.В.
7	Разработка веб-приложения	Ким С.В.
8	Тестирование веб-приложения	Коровкин В.А, Ким С.В.
9	Согласование выполненной работы с научным руководителем	Коровкин В.А, Ким С.В.
10	Выполнение других частей работы (финансовый менеджмент, социальная ответственность)	Ким С.В.

Для того чтобы определить трудоемкость работ, используются следующие показатели:

ожидаемое значение трудоемкости;

продолжительность каждой работы;

продолжительность выполнения i-ой работы в календарных днях;

коэффициент календарности.

Для определения ожидаемого значения продолжительности работ $t_{ож}$ применяется метод двух оценок t_{min} и t_{max}:

$$t_{ож} = \frac{3 \cdot t_{min} + 2 \cdot t_{max}}{5}, \qquad (2)$$

где t_{min} – минимальная трудоемкость работ, чел/дн.;

t_{max} – максимальная трудоемкость работ, чел/дн.

Используя следующую формулу, определяется продолжительность каждой работы в рабочих днях T_{pn}:

$$T_{рn} = \frac{t_{ожn}}{Ч_n}, \qquad (3)$$

где $T_{рn}$ – продолжительность одной работы, раб.дни,

$t_{ожn}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дни,

$Ч_n$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

Расчет продолжительности этапа работ в календарных днях ведется по формуле:

$$T_{кд} = T_{рд} \cdot K_{кл}, \quad (4)$$

где $T_{кд}$ – продолжительность выполнения i-й работы в календарных днях,

$T_{рд}$ – продолжительность выполнения i-й работы в рабочих днях,

$K_{кл}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$K_{кл} = \frac{T_{кал}}{T_{кал} - T_{вд} - T_{пр}}, \qquad (5)$$

где $T_{кал}$ – количество календарных дней в году,

$T_{вд}$ – количество выходных дней в году,

$T_{пр}$ – количество праздничных дней в году.

Согласно производственному календарю (для 6-дневной рабочей недели) в 2022 году 365 календарных дней, из них 66 выходных или праздничных дней, следовательно, $K_{\text{кл}} = 1{,}22$.

Расчеты по трудоемкости выполнения работ представлены в таблице 3.4. Диаграмма Гантта, построенная по рассчитанным показателям, представлена в таблице 3.5.

Таблица 3.4 – Временные показатели осуществления разработки

№	Наименование работы	Исполнители работы	Трудоемкость работ, чел.-дни			Длительность работ, дни	
			t_{min}	t_{max}	$t_{ож}$	$T_р$	$T_к$
1	Выбор научного руководителя магистерской работы	Ким С.В.	1	2	1,4	1	1
2	Составление и утверждение темы магистерской работы	Коровкин В.А.	4	7	5,2	5	6
		Ким С.В.	4	7	5,2	5	6
3	Составление календарного плана-графика выполнения магистерской работы	Коровкин В.А.	2	2	2	2	2
4	Подбор и изучение литературы по теме магистерской работы	Ким С.В.	10	15	12	12	14
5	Анализ предметной области	Ким С.В.	10	15	12	12	14
6	Проектирование веб-приложения	Ким С.В.	7	10	8,2	8	9
7	Разработка веб-приложения	Ким С.В.	30	45	36	40	47
8	Тестирование веб-приложения	Ким С.В.	3	5	3,8	3	3
		Коровкин В.А.	3	5	3,8	3	3
9	Согласование выполненной работы с научным руководителем	Ким С.В.	4	7	5,2	5	5
		Коровкин В.А.	4	7	5,2	5	5
10	Выполнение других частей работы (финансовый менеджмент, социальная ответственность)	Ким С.В.	8	10	8,8	8	9
11	Подведение итогов, оформление работы	Ким С.В.	3	5	3,8	4	4
Итого		Коровкин В.А.				15	16
		Ким С.В.				98	112

Таблица 3.5 – График Гантта

Этап	$T_к$		февраль			март			апрель			май		
	НР	И	10	20	30	40	50	60	70	80	90	100	110	120
1	1	–												
2	6	6												
3	2	–												
4	–	14												
5	–	14												
6	–	9												
7	–	47												
8	3	3												
9	5	5												
10	–	9												
11	–	4												

НР И

3.3 Бюджет научно-технической разработки.

Статья затрат «Материальные затраты» включает в себя затраты на приобретение канцелярских принадлежностей на сумму 1240 рублей и оплату электроэнергии на сумму 455 рублей. Общая сумма материальных затрат составляет 1695 руб.

В статью затрат «Специальное оборудование для научных (экспериментальных) целей» входят суммы, необходимые на обеспечение амортизации используемого оборудования.

В качестве оборудования выступает компьютер стоимостью 60000 рублей и монитор стоимостью 8000 рублей. Итоговая стоимость составляет 68000 рублей.

Расчет амортизации персонального компьютера, используемого при написании работы: первоначальная стоимость персонального компьютера составляет 40000 рублей; срок полезного использования для офисных машин – 3 года; планируется использовать персональный компьютер для написания ВКР в течение 4 месяцев. Тогда:

- норма амортизации:

$$\text{Ан} = \frac{1}{n} \cdot 100\% = \frac{1}{3} \cdot 100\% = 33{,}33\ \%, \quad (6)$$

- годовые амортизационные отчисления:

$$\text{Аг} = 40000 \cdot 0{,}33 = 13200 \text{ руб.}, \quad (7)$$

- ежемесячные амортизационные отчисления:

$$\text{Ам} = \frac{13200}{12} = 1100 \text{ руб.}, \quad (8)$$

- итоговая сумма амортизации основных средств:

$$\text{А} = 1100 \cdot 4 = 4400 \text{ руб.} \quad (9)$$

Итоговая сумма затрат на амортизацию составила 4400 руб.

Статья расходов «Основная заработная плата исполнителей темы» включает основную заработную плату с учетом премий и доплат для исполнителей проекта: студента и научного руководителя.

Месячный оклад руководителя ТПУ с должностью ассистента составляет 21760 рубля (без учета районного коэффициента, но с учетом премиальных и надбавок), для студента был взят оклад ассистента без научной степени – 21760 рублей.

В таблице 3.6 показаны количества календарных, нерабочих и праздничных дней, дней, пришедшихся на потерю рабочего времени и действительный годовой фонд рабочего времени.

Таблица 3.6 – Баланс рабочего времени (для 6-дневной недели)

Показатели рабочего времени	Дни
Календарные дни	365
Нерабочие дни (праздники/выходные)	66
Потери рабочего времени (отпуск/невыходы по болезни)	56
Действительный годовой фонд рабочего времени	243

Количество месяцев работы без отпуска принимается за 10,4 (с учетом длительности отпуска в 48 дней). Тогда, зная месячную заработную плату, можно рассчитать среднедневную заработную плату:

$$З_{дн}^{рук} = \frac{З_м \cdot М}{F_д} = \frac{21760 \cdot 10{,}4}{243} = 931{,}29 \text{ руб.}, \quad (10)$$

$$З_{дн}^{студ} = \frac{З_м \cdot М}{F_д} = \frac{21760 \cdot 10{,}4}{243} = 931{,}29 \text{ руб.} \quad (11)$$

Расчет основной заработной платы осуществляется по формуле:

$$З_{осн} = З_{дн} \cdot Т_р \cdot (1 + К_{пр} + К_д) \cdot К_р, \quad (12)$$

где $З_{дн}$ – среднедневная заработная плата, руб.,

$Т_р$ – продолжительность работ, выполняемых работником, раб. дни,

$К_{пр}$ – премиальный коэффициент,

$К_д$ – коэффициент доплат и надбавок,

$К_р$ – районный коэффициент.

Результаты соответствующих расчетов приведены в таблице 3.7.

Таблица 3.7 – Расчет основной заработной платы

Исполнители	$З_{дн}$, руб.	$К_{пр}$	$К_д$	$К_р$	$Т_р$	$З_{осн}$, руб.
Студент	931,29	0,3	0,2	1,3	98	177969,51

Научный руководитель	931,29	0,3	0,2	1,3	15	27240,23
Итого						205209,75

Зная основную заработную плату, можно рассчитать дополнительную заработную плату в размере 12 % от основной:

$$З_{доп} = k_{доп} \cdot З_{осн}, \quad (13)$$

где $k_{доп}$ – коэффициент дополнительная заработная плата,

$З_{осн}$ – основная заработная плата.

Таблица 3.8 – Расчет дополнительной заработной платы

Исполнители	$k_{доп}$	$З_{осн}$, руб.	$З_{доп}$, руб.
Студент	0,12	177969,51	21356,34
Научный руководитель	0,12	27240,23	3268,82
Итого			24625,16

Отчисления во внебюджетные фонды рассчитываются как:

$$З_{внеб} = k_{внеб} \cdot (З_{осн} + З_{доп}), \quad (14)$$

где $k_{внеб}$ – коэффициент внебюджетные фонды; в 2022 г., в соответствии с Федеральным законом для учреждений, осуществляющих образовательную и научную деятельность, используется пониженная ставка – 30%,

$З_{осн}$ – основная заработная плата,

$З_{доп}$ – дополнительная заработная плата.

Таблица 3.9 – Расчет страховых отчислений

Исполнители	$k_{внеб}$	$З_{доп}$	$З_{осн}$	$З_{внеб}$
Студент	0,30	21356,34	177969,51	59797,755
Научный руководитель	0,30	3268,82	27240,23	9152,715
Итого				68950,47

Накладные расходы учитывают прочие затраты, не попавшие в предыдущие статьи расходов: оплата услуг связи, электроэнергии и т.д.

Их величина определяется согласно следующей формуле:

$$З_{накл} = k_{накл} \cdot (\text{сумма статей расходов}), \quad (15)$$

где $k_{накл}$ – коэффициент накладных расходов, принятый за 16 %.

Таблица 10 – Расчет накладных расходов

Статьи затрат	Сумма, руб.
Материальные затраты	1695

Затраты на амортизацию	4400
Затраты на основную заработную плату	205209,75
Затраты на дополнительную заработную плату	24625,16
Затраты на отчисления во внебюджетные фонды	68950,47
Накладные расходы	48573,66

Результаты составления итогового бюджета разработки представлены в таблице 3.11.

Таблица 3.11 – Бюджет затрат на разработку

Наименование	Сумма, руб	Удельный вес, %	Сумма, руб	Удельный вес, %	Сумма, руб	Удельный вес, %
	Текущая разработка		Аналог 1		Аналог 2	
Материальные затраты	1695	0,46	3045	0,62	3045	0,70
Затраты на амортизацию	4400	1,2	4400	0,89	4400	1,02
Затраты на основную заработную плату	205209,75	58,06	297940,89	60,57	248803,79	57,54
Затраты на дополнительную заработную плату	24625,16	6,97	35752,91	7,27	29561,62	6,84
Затраты на отчисления во внебюджетные фонды	68950,47	19,51	84366,64	17,15	81765,13	18,91
Накладные расходы	48573,66	13,74	66351,12	13,49	64803,59	14,99
Итого	353453,57	100	491856,5	100	432379,13	100

3.4 Оценка сравнительной эффективности исследования

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

Интегральный финансовый показатель разработки определяется по следующей формуле:

$$I_{\text{финр}}^{\text{исп}.i} = \frac{\Phi_{pi}}{\Phi_{max}}$$

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

Φ_{pi} – стоимость i-го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги).

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить по следующей формуле:

$I_{pi} = \sum a_i \cdot b_i$

где: I_{pi} – интегральный показатель ресурсоэффективности для i-го варианта исполнения разработки;

a_i – весовой коэффициент i-го варианта исполнения разработки;

b_i^a, b_i^p – бальная оценка i-го варианта исполнения разработки, устанавливается экспертным путём по выбранной шкале оценивания;

n – число параметров сравнения.

Расчёт интегрального показателя ресурсоэффективности приведён в форме таблицы (таблице 3.12).

Таблица 3.12 – Сравнительная оценка характеристик вариантов исполнения проекта

ПО Критерии	Весовой коэффициент параметра	Текущий проект	Аналог 1	Аналог 2
1. Выход продукта)	0,1	5	4	3
2. Удобство в эксплуатации	0,25	5	5	3
3. Надежность	0,2	4	4	3
4. Безопасность	0,15	4	4	4
5. Простота эксплуатации	0,15	5	4	4
6. Возможность автоматизации данных	0,15	4	3	5
Итого	1	27	24	22

$I_m^p = 5*0,1 + 5*0,25 + 4*0,2 + 4*0,15 + 5*0,15 + 4*0,15 = 4,50$

$I_1^{\text{А}} = 4*0,1 + 5*0,25 + 4*0,2 + 4*0,15 + 4*0,15 + 3*0,15 = 4,1$

$I_2^{\text{А}} = 3*0,1 + 3*0,25 + 3*0,2 + 4*0,15 + 4*0,15 + 5*0,15 = 3,6$

Интегральный показатель эффективности разработки $I_{\text{финр}}^p$и аналога $I_{\text{финр}}^{\text{а}}$определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{\text{финр}}^p = \frac{I_m^p}{I_{\text{ф}}^p}; I_{\text{финр}}^{\text{а}} = \frac{I_m^{\text{а}}}{I_{\text{ф}}^{\text{а}}}$$

Сравнение интегрального показателя эффективности текущего проекта и аналогов позволит определить сравнительную эффективность проекта.

Сравнительная эффективность проекта определяется по формуле:

$$\text{Э}_{\text{ср}} = \frac{I_{\text{финр}}^p}{I_{\text{финр}}^{\text{а}}}$$

где: $\text{э}_{\text{ср}}$ – сравнительная эффективность проекта;

$I_{\text{финр}}^p$ – интегральный показатель разработки;

$I_{\text{финр}}^{\text{а}}$ – интегральный технико-экономический показатель аналога.

Сравнительная эффективность разработки по сравнению с аналогами представлена в таблице 3.13.

Таблица 3.13 – Сравнительная эффективность разработки

№ п/п	Показатели	Разработка	Аналог 1	Аналог 2
1	Интегральный финансовый показатель разработки	0,72	1	0,88
2	Интегральный показатель ресурсоэффективности разработки	4,5	4,1	3,6
3	Интегральный показатель эффективности	11,84	16,4	18
4	Сравнительная эффективность вариантов исполнения	1	0,72	0,66

3.5 Выводы по разделу

В ходе работы над разделом был проведен предпроектный анализ, включающий анализ конкурентных решений и SWOT-анализ. Анализ

конкуретных решений показал преимущество разрабатываемого подхода по сравнению с описанными в литературе. SWOT-анализ позволил определить потенциальные пути улучшения разработки: перспективными можно считать улучшение предлагаемого в данной работе подхода путем применения новых методов машинного обучения и понижение порога вхождения для рядовых пользователей. Также в ходе инициации проекта был составлен календарный план проекта, проведена оценка трудоемкости работ. График-план проекта был представлен в виде диаграммы Гантта. Был сформирован бюджет разработки, включающий материальные затраты, затраты на амортизацию, основную заработную плату исполнителям, дополнительную заработную плату, отчисления во внебюджетные фонды и накладные расходы. Общий бюджет разработки составил 353453,57 руб. Было проведено сравнение эффективности выполнения для данной разработки и ее прямого аналога на основании сравнения значений интегральных показателей эффективности.

4. Социальная ответственность

Целью работы является проектирование и реализация адаптивного клиент-серверного SPA-приложения. В качестве предметной области был выбран небольшой сервис по доставке еды.

Сфера применения разработки – данная работа решает бизнес задачи сервиса по доставке еды.

Разработка этого проекта велась с использованием персонального компьютера на нескольких рабочих площадках, расположенных в Томском политехническом университете, которые обладают характеристиками офисного типа.

Поскольку данная работа по разработке проекта напрямую связана с использованием компьютерного оборудования, необходимо обеспечить соблюдение трудового законодательства, а также законов, защищающих окружающую среду от возможных вредных последствий, доказанных использованием компьютера.

4.1 Правовые и организационные вопросы обеспечения безопасности

К теме трудового права, основным источником информации является трудовой кодекс Российской Федерации. Данный официальный документ даёт регулирования в сфере, связанной с заработной платы, рутинного труда, особенности регулирования труда женщин, регулирования труда детей, регулирования труда лиц с ограниченными возможностями и другие. Трудовой кодекс, помимо основных положений отношений между работником и работодателем, содержит руководящие принципы по вопросу безопасности труда, которые также должны применяться в процессе реализации магистерских диссертаций. Следует подчеркнуть, что российское законодательство запрещает дискриминацию по любым признакам и принудительный труд.

На вопрос о правилах, которые контролируют режим работы, в пределах Российской Федерации, режим, широко распространённый в делопроизводстве – это рабочая неделя, которая состоит из 5 дней работы и двух дней, что соответствуют концу недели, в течение рабочего времени в рабочий диапазон, который берет начало в 9:00 и имеет своё завершение в 17:00, следует уточнить, что в это время работы есть отдых, который длится с 13:00 до 14:00. Кроме того, в соответствии с правилами нормальное рабочее время не может превышать 40 часов в неделю, хотя дети в возрасте до 16 лет могут работать, рабочее время в течение рабочей недели для детей в возрасте до 16 лет не должно превышать 24 часа, для молодых людей в возрасте от 16 до 18 лет число рабочих часов в течение рабочей недели не может превышать 35 часов, эти правила применяются также к инвалидам I и II группы [13].

При анализе должности, которую занимает работодатель, правила Российской Федерации обязывают его обеспечить надлежащие условия, чтобы его сотрудники могли безопасно выполнять свою работу. Информация, изложенная в нормативных актах Российской Федерации, содержит

подробную информацию о характеристиках рабочего места, исправности используемых машин, исправности технологического оборудования, своевременном предоставлении необходимой технической документации, надлежащем качестве используемых материалов, требуемом качестве используемых инструментов и т. д. Ко всему вышеописанному следует также добавить условия, необходимые в области здоровья и безопасности на производстве.

Рассматривая тему выполнения диссертационной работы как трудовой деятельности, мы должны учитывать, что это вид работы, предполагающий использование компьютера или также называемый персональные электронно-вычислительные машины (ПЭВМ), помимо персонального компьютера основными элементами рабочего места программиста являются: письменный стол, рабочее кресло (Кресло), дополнительный дисплей, дополнительная клавиатура, дополнительная мышь. Место, в котором развивается эту работу, необходимо обеспечить правильные условия для выполнения поставленных задач, рабочие операции должны выполняться в зоне действия поля двигателя, выполнение рабочих операций необходимо убедиться в зоне легкого доступа, как это указывают нормы [14, 15] также должна быть обеспечена оптимальная осанка человека, правильное расположение и порядок рабочего места, свобода трудового передвижения, использование оборудования, отвечающего требованиям эргономики и инженерной психологии, все это обеспечивает более эффективный трудовой процесс, снижает усталость и предотвращает риск профессиональных заболеваний.

В Российской Федерации основным документом, регулирующим любую работу с компьютерами (или ПЭВМ), является ТОИ Р-45-084-01 Типовая инструкция по охране труда при работе на персональном компьютере, этот документ содержит информацию, регулирующую условия и организацию, а также руководящие принципы санитарно-эпидемиологического регулирования, в дополнение к этому официальному документу используются

также правила, регулирующие рабочее место пользователя, такие как: ПЭВМ:: ГОСТ 12.2.032-78 ССБТ «Рабочее место при выполнении работ сидя. Общие эргономические требования» и ГОСТ Р 50923-96. «Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения» [16-17].

Помещения, используемые в качестве рабочего пространства в Томском политехническом университете, обеспечивают комфортную и безопасную среду для работы с компьютером, конструкции рабочих мест учитывают требования расстояния до настольных компьютеров, но при использовании персонального компьютера пользователь обязан уважать расстояние от глаз до монитора портативного компьютера, который должен быть 650 мм. При работе с персональным компьютером это позволяет поворачивать экран в горизонтальной и вертикальной плоскости с фиксацией в указанном положении для обеспечения переднего наблюдения монитора. В заключение, при выполнении этой работы были приняты во внимание рекомендуемые меры по обеспечению правильного размещения персонального компьютера на поверхности рабочего стола, помимо расположения компьютера, была также учтена правильная калибровка яркости и контрастности монитора для повышения комфорта при работе с персональным компьютером. К вышеизложенному следует добавить, что, используя описанную информацию о том, что рекомендуемое время отдыха должно составлять 10-15 минут после каждых 45-60 минут работы, было решено делать 10-минутный отдых каждые 50 минут работы, в течение этого времени отдыха выполняются упражнения для глаз и минуты физической подготовки. Сидячее положение - это положение, выбранное в работе с компьютерами, в сидячем положении основная нагрузка ложится на мышцы, поддерживающие позвоночник и голову. В связи с этим, когда вы сидите в течение длительного времени, необходимо время от времени менять фиксированные рабочие позы.

4.2 Производственная безопасность

Чтобы обеспечить безопасность студента во время разработки магистерской диссертации, выполняя свою работу перед персональным компьютером, необходимо проанализировать возможные вредные и опасные воздействия, которые могут возникнуть в зависимости от среды, в которой студент работает. Когда воздействие человека может привести к заболеванию, оно считается вредным фактором, вместо этого, когда воздействие человека может привести к травме, оно считается опасным фактором.

Для проведения анализа вредных и опасных факторов, которые могут повлиять на развитие этой работы в качестве источника берётся ГОСТ 12.0.003-2015 «Опасные и вредные производственные факторы. Классификация» [18]. Анализируя тип выполняемой работы, в данном случае ориентированной на разработку программного обеспечения, целесообразно учитывать вредные и опасные физические и психофизические факторы производства, этот тип анализа идеально подходит для работ, связанных с использованием компьютеров, ниже рассматриваются физические факторы. Список факторов показан в таблице 4.1.

Таблица 4.14 – Возможные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Разработка	Изготовление	Эксплуатация	
1. Повышенная или пониженная температура и относительная влажность воздуха.	+	+	+	СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания [31]. ГОСТ 12.1.005-88 Система стандартов безопасности труда (ССБТ). Общие санитарно-гигиенические требования к воздуху рабочей зоны [20].

2. Превышение уровня шума	+	+	+	ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности [21].
3. Отсутствие или недостаток освещения.	+			СП 52.13330.2016 Естественное и искусственное освещение [33]. Актуализированная редакция СНиП 23-05-95* (с Изменением N 1).
4. Психофизиологические факторы (монотонность труда, умственное и эмоциональное напряжение, перенапряжение зрительных анализаторов)	+	+	+	МР 2.2.9.2311 – 07 «Профилактика стрессового состояния работников при различных видах Профессиональной деятельности» [32].
5. Поражение электрическим током	+		+	ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов (с Изменением N 1) [25].
6. Короткое замыкание	+	+	+	ГОСТ Р 50571.4.44-2019. Электроустановки Низковольтные. Часть 4.44. Защита для обеспечения безопасности. Защита от резких отклонений напряжения и электромагнитных возмущений [33]
7. Статическое электричество	+	+	+	ГОСТ 12.1.044-89. Система стандартов безопасности труда. Пожаровзрывоопасность веществ и материалов [34] ГОСТ 12.1.045-84 ССБТ. Электростатические поля. Допустимые уровни на рабочих местах и требования к проведению контроля [35].

Температура и относительная влажность воздуха

Проводя анализ микроклимата, особенно в связи с его температурой и относительной влажностью, мы знаем, что длительное воздействие на человека плохих условий на уровне микроклимата ухудшает самочувствие работника, снижает его производительность и может привести к заболеваниям, поэтому на рабочем месте следует уделять большое внимание контролю микроклимата.

Оптимальные микроклиматические условия устанавливаются в соответствии с критериями оптимального теплового и функционального состояния человека. Допустимые значения показателей микроклимата устанавливаются в тех случаях, когда по технологическим, техническим и экономическим причинам оптимальные значения не могут быть предоставлены. Допустимые значения не вызывают вреда или расстройств здоровья, но могут привести к общим и локальным ощущениям теплового дискомфорта, напряжения механизмов терморегуляции, ухудшения самочувствия и снижения работоспособности.

Параметры микроклимата включают: температуру воздуха, температуру поверхностей, относительную влажность воздуха, скорость воздуха. Оптимальные значения зависят от сезона, а также от физических усилий, прилагаемых работником. При разработке этой работы, поскольку это разработка программного обеспечения, вся работа выполняется сидя без систематических физических усилий.

Мы берём в качестве ссылки значения, указанные в ГОСТ 12.1.005-88 [20], которые показаны в таблице 4.2.

Таблица 4.15 – Оптимальные и допустимые величины показателей микроклимата на рабочих местах производственных помещений

Оптимальные значения характеристик микроклимата			
Период года	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	22 - 24	40 - 60	0,1
Теплый	23 - 25	40 - 60	0,1
Допустимые значения характеристик микроклимата			
Холодный	18 - 25	не более 75	не более 0,1
Теплый	20 - 28	55	0,1 - 0,2

Превышение уровня шума

При анализе шумов, которые могут повлиять на работников, возможными источниками шума являются: рабочее оборудование, компьютерные вентиляторы, копиры и кондиционеры.

Шум оказывает негативное влияние на организм человека: снижает работоспособность, повышает усталость, влияет на органы слуха и центральную нервную систему, снижает внимание.

Уровень шума на рабочем месте программистов не должен превышать 50дБА, а в залах обработки информации на вычислительных машинах 65дБА [21,22]. Необходимые ограничения пространства показаны в следующей таблице.

Таблица 4.16 – Допустимые значения уровней звукового давления в октавных полосах частот и уровня звука, создаваемого ПЭВМ.

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

В рабочих пространствах, используемых в Томском политехническом университете являются пространства, в которых очень мало шума, особенно учебные комнаты, которые являются местами, которые чаще всего использовались.

Отсутствие или недостаток освещения

Ещё один важный момент, который следует учитывать, - это освещение на рабочем месте, недостаточное освещение негативно влияет на зрение работника, кроме того, это приводит к быстрой усталости, снижает его работоспособность, поскольку вызывает физические дискомфорты, такие как головные боли и бессонница. С учётом правил СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95, минимальное освещение на рабочих местах не должно отличаться от нормализованного среднего освещения в помещении более чем на 10% [23].

Таблица 4.17 – Требования к освещению помещений промышленных предприятий

Помещения	Рабочая поверхность и плоскость нормирования КЕО и освещенности, и высота плоскости над полом	Искусственное освещение					Естественное освещение	
		Освещенность рабочих поверхностей, лк		Объединенный показатель дискомфорта UGR, не более	Коэффициент пульсации освещенности	Индекс цветопередачи источников света Ra	КОЕ * $e_н$, %	
		При комбинированном освещении	При общем освещении				При верном или комбинированном освещении	При боковом освещении
Кабинеты и рабочие комнаты, офисы, представительства	Г-0,8	400/200	300	21	15	80	3,0	1,0

**КОЕ - коэффициента естественной освещенности.*

Также согласно СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95 уровень освещения на поверхности рабочего стола при работе с ПЭВМ должен быть в диапазоне от 300 до 500 лк [23].

Выполненная работа имеет тип разработки программного обеспечения, поэтому она включает в себя большую работу визуального типа, упущение этого фактора может привести к заболеваниям зрения.

Расчёт искусственного освещения

Дано помещение с размерами: длина А = 8 м, ширина В = 4 м, высота Н = 3,0 м. Высота рабочей поверхности $h_{рп}$ = 0,65 м. Требуется создать освещенность Е = 300 лк.

Коэффициент отражения стен ρ_c = 30 %, потолка $\rho_п$ = 50 %. Коэффициент запаса k =1,5 ; коэффициент неравномерности Z = 1,1. Светильники типа ОД, λ = 1,5. h_c = 0,05 м.

Определение расчетную высоту:

$$h = H - h_c - h_{\text{рп}} = 3{,}0\text{м} - 0{,}05\text{м} - 0{,}65\text{м} = 2{,}3\text{м}$$

Расстояние между светильниками:

$$L = \lambda \cdot h = 1{,}5 \cdot 2{,}3\text{м} = 3{,}45\text{м}$$

Расстояние от крайнего ряда светильников до стены:

$$\frac{L}{3} = 3{,}45\text{м}/3 = 1{,}15\text{м}$$

Определение количество рядов светильников и количество светильников в ряду:

$$n_{\text{ряд}} = \frac{(\text{В} - \frac{2}{3}L)}{L} + 1 = \frac{(4 - \frac{2}{3} \cdot 3{,}45)}{3{,}45} + 1 \approx 1$$

$$n_{\text{св}} = \frac{(\text{А} - \frac{2}{3}L)}{l_{\text{св}} + 0{,}5} = \frac{(8 - \frac{2}{3} \cdot 3{,}45)}{1{,}53 + 0{,}5} \approx 3$$

Светильники необходимо размещать в 1 ряд. В каждом ряду можно установить 3 светильников типа ПВЛ мощностью 80 Вт (с длиной 1,53 м), при этом разрывы между светильниками в ряду составят 50 см. Изображение в масштабе план помещения и размещения на нем светильников отображается на рисунке 1. Учитывая, что в каждом светильнике установлено две лампы, общее число ламп в помещении N = 6.

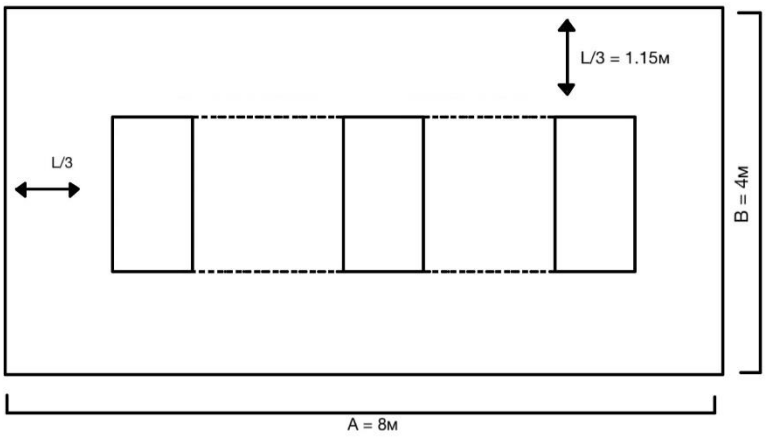

Рисунок 4.17 – План помещения и размещения светильников с люминесцентными

Находим индекс помещения

$$i = \frac{S}{h(A+B)} = \frac{32}{2{,}3(8+4)} = 1{,}16$$

Коэффициент использования светового потока

$$\eta = 0{,}48$$

Определяем потребный световой поток ламп в каждом из рядов:

$$\Phi = \frac{\text{Ен} \cdot S \cdot K_З \cdot Z}{N_л \cdot \eta} = \frac{300 \cdot 32 \cdot 1{,}5 \cdot 1{,}1}{6 \cdot 0{,}48} = 5500$$

Ближайшая стандартная лампа – ЛТБ 80 Вт с потоком 5200 лм.

Проверка выполнения условия:

$$-10\% \leq \frac{\Phi_{\text{л.станд}} - \Phi_{\text{л.расч}}}{\Phi_{\text{л.станд}}} \cdot 100\% \leq +20\%$$

Результат: $-10\% \leq -5{,}77\% \leq +20\%$

Определение электрическую мощность осветительной установки:

$$P = 6 \cdot 80 = 4800\text{Вт}$$

Электрический ток

Поражение электрическим током является опасным фактором производства, и, поскольку при разработке программного обеспечения программист находится в контакте с электронным оборудованием, следует обратить внимание на потенциальные проблемы безопасности, связанные с электричеством. Нормы электробезопасности на рабочем месте

регламентируются ГОСТ Р 12.1.019-2017 ССБТ Электробезопасность. Общие требования и номенклатура видов защиты [23].

ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов, вопросы требований к защите от поражения электрическим током освещены в ГОСТ Р 12.1.019-2009 ССБТ [25,26].

Поражение человеческого организма электрическим током может быть разнообразным. Разряд, проходящий через ткани, оказывает на него тепловое, электролитическое, биологическое и динамическое действие.

Напряжения прикосновения и токи, протекающие через тело работника при неаварийном режиме электроустановки, не должны превышать следующие значений:

- При 50 Гц переменного тока – 2,0 В и 0,3 мА, соответственно.
- При 400 Гц переменного тока – 3,0 В и 0,4 мА
- При постоянном токе – 8,0 В и 1,0 мА.

Существуют также защитные меры, которые указывают нормальные условия (защита от прямого контакта), которые будут приняты во внимание, это достигается за счёт основной защиты, такой как: безопасное расположение токоведущих частей, размещение их вне зоны досягаемости частями тела, конечностями; предупредительная световая, звуковая сигнализации, блокировки безопасности, знаки безопасности; основная изоляция; и другие технические мероприятия.

Помещение, где расположено рабочее место оператора ПЭВМ, относится к помещениям без повышенной опасности ввиду отсутствия следующих факторов: сырость, токопроводящая пыль, токопроводящие полы, высокая температура, возможность одновременного прикосновения человека к имеющим соединение с землёй металлоконструкциям зданий, технологическим аппаратам, механизмам и металлическим корпусам электрооборудования.

4.3 Экологическая безопасность

В настоящее время давление во всех странах на заботу об окружающей среде привело к разработке законов, которые помогают заботиться о экосистеме. Одним из способов помочь в уходе за окружающей средой является переработка компьютерного оборудования, хотя это очень сложный процесс из-за сложности его классификации, последующей гомогенизации и отправки для повторного использования.

Переработка макулатуры представляет собой многоэтапный процесс, цель которого заключается в восстановлении бумажного волокна и, зачастую, других компонентов бумаги (таких как минеральные наполнители) и использование их в качестве сырья для производства новой бумаги [27].

В нормативном документе ТОИ Р-45-084-01 Типовая инструкция по охране труда при работе на персональном компьютере даются следующие общие рекомендации по снижению опасности для окружающей среды, исходящей от компьютерной техники:

- Применять оборудование, соответствующее санитарным нормам и стандартам экологической безопасности;

- Применять расходные материалы с высоким коэффициентом использования и возможностью их полной или частичной регенерации;

- Отходы в виде компьютерного лома утилизировать;

- Использовать экономичные режимы работы оборудования.

Люминесцентные лампы, используемые для искусственного освещения на рабочих местах, также требуют специального удаления, поскольку они содержат от 10 до 70 мг ртути, которая является чрезвычайно опасным химическим веществом и может вызывать отравление живых существ, а также загрязнение атмосферы, гидросферы и литосферы. Срок службы этих ламп составляет около 5 лет, после чего они должны быть доставлены для обработки в специальные пункты приёма. Во время разработки и написания ВКР

образовывался мусор, такой как: канцелярские принадлежности, бумажные отходы [28].

Согласно ГОСТ-у 17.4.3.04-85, особое внимание следует уделять почвам, прилегающим к предприятиям и объектам промышленности, жилищно-коммунального и сельского хозяйств, транспорта, которые по характеру своей деятельности могут загрязнять почву посредством выбросов, сбросов, отходов, стоков и осадков сточных вод [30].

Охрана почв от загрязнения должна осуществляться с учетом утилизации и захоронения выбросов, сбросов, отходов, стоков и осадков сточных вод с соблюдением мер по предотвращению загрязнения почв.

4.4 Безопасность в чрезвычайных ситуациях

Наиболее вероятным случаем чрезвычайной ситуации в офисе является пожар, поскольку внутри офисов много бумаги, пыли и других легковоспламеняющихся материалов. Возникновение пожара может происходить из-за нескольких факторов, среди которых мы имеем:

- Возникновение короткого замыкания в проводке из-за неисправностей проводки или электрических соединений и электрических распределительных панелей;

- Пожар в устройствах компьютерной техники из-за нарушения изоляции или неисправности самого оборудования;

- Пожар мебели или полов из-за нарушения правил пожарной безопасности, а также ненадлежащего использования дополнительных электроприборов и электроустановок;

- Возгорание устройств искусственного освещения.

Правила, нормализующие события, связанные с пожарами, мы находим в ГОСТ 12.1.004-91 Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Общие требования [29].

Во всех местах, используемых для разработки этой работы, есть нормативная вывеска, ведущая к аварийным выходам, также схемы эвакуации расположены в видимых местах, эти схемы содержат информацию о аварийных выходах и инструкции, которые должны соблюдаться в случае аварийной эвакуации.

Приведена схема эвакуации четвёртого этажа 20-го здания Томского политехнического университета (рисунок 4.2).

Рисунок 4.2 – Схема эвакуации четвёртого этажа 20-го здания Томского политехнического университета

4.5 Выводы по разделу

После анализа политики Российской Федерации по вопросам, о безопасности работников и работодателей, можно сделать вывод, что рабочие места, используемые для выполнения данной работы, соответствуют необходимым правилам, поскольку соблюдаются нормы безопасности, при использовании компьютеров, рабочие места не приводят к ухудшению здоровья ученика, кроме того, установки, в которых выполняется работа, отвечают нормам и правилам, регулирующим соответствующие параметры микроклимата, освещения и электрической безопасности, наконец,

экологическая безопасность на предприятии соответствует действующим нормативным документам.

Со стороны студента, чтобы избежать негативного воздействия на здоровье во время выполнения работы, устанавливается план запланированных перерывов, включающих небольшие физические и визуальные упражнения, которые помогают расслабиться.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы была достигнута заявленная цель, а именно спроектировано и реализовано адаптивное клиент-серверное SPA-приложение по доставке еды.

В ходе выполнения работы были достигнуты следующие результаты:

1. Составление требований к разрабатываемому проекту;
2. Знакомство с фреймворком Vue.js, основными механизмами работы с моделями данных. Для отрисовки и взаимодействия с компонентами были использованы директивы vue: v-bind, v-on, v-if, v-for, v-model, transition;
3. Спроектированы компоненты приложения: представлена схема компонентов;
4. С использованием vue-router были объявлены, как основные маршруты, так и вложенные (nested routes). Осуществлена динамическая компоновка страницы;
5. Для отправки запросов GET/POST/UPDATE/DELETE использовалась библиотека axios, которая позволила не только получать успешные ответы с сервера в формате json, но и разбирать ошибки;
6. Для синхронизации данных, которые должны быть общими во всех (или многих) компонентах веб-приложения было использовано «централизованное хранилище» Vuex. Были созданы соответствующие состояния, геттеры, мутации и действия для определения, получения и редактирования этих данных. Учитывая эти моменты, по мере роста и масштабировании приложения, хранилище может существенно увеличиться. Чтобы решить эту задачу Vuex предоставляет возможность разделить хранилище на модули, где каждому модулю будет соответствовать собственное состояние, мутации, действия, геттеры, а также возможность использования встроенных подмодулей;

7. Для корректного заполнения форм были использованы методы библиотеки VeeValidate;

8. Разработан REST API сервис для работы с клиентом;

9. Разработана модель хранения данных для сервиса по доставке еды.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. HTML [Электронный ресурс] / Языки разметки веб-страниц – URL: http://bourabai.kz/dhtml/index.html (Дата обращения 21.03.2022)
2. CSS [Электронный ресурс] / Что такое CSS – URL: http://www.a-sd.ru/glossariy/c_d/c_s_s.html (Дата обращения 21.03.2022)
3. SCSS [Электронный ресурс] / основы Sass – URL: https://sass-scss.ru/guide/ (Дата обращения 23.03.2022)
4. JavaScript [Электронный ресурс] / Программирование на JavaScript – URL: https://itgen.io/programmirovanie-na-javascript (Дата обращения 30.03.2022)
5. API [Электронный ресурс] / API – URL: https://traceway.ru/support/glossary/api/ (Дата обращения 04.04.2022)
6. Axios [Электронный ресурс] / Используем axios для доступа к API – URL: https://ru.vuejs.org/v2/cookbook/using-axios-to-consume-apis.html (Дата обращения 12.04.2022)
7. Vue.js [Электронный ресурс] / Что такое Vue.js? – URL: https://ru.vuejs.org/v2/guide/index.html (Дата обращения 12.04.2022)
8. Vue-router [Электронный ресурс] / Роутинг – URL: https://ru.vuejs.org/v2/guide/routing.html (Дата обращения 12.04.2022)
9. Vuex [Электронный ресурс] / Что такое Vuex? – URL: https://vue3js.cn/vuex/ru/ (Дата обращения 14.05.2022)
10. Laravel [Электронный ресурс] / 8 лучших PHP Framework для веб-разработчиков – URL: https://www.hostinger.ru/rukovodstva/8-luchshih-php-framework-dla-web-razrabotchikov/ (Дата обращения 12.05.2022 г.)
11. Laravel Sanctum [Электронный ресурс] / Пакет Laravel Sanctum – URL: https://laravel.su/docs/8.x/sanctum (Дата обращения 24.05.22)

12. Swagger [Электронный ресурс] / Swagger: что это и как с ним работать? – URL: https://highload.today/swagger-api/ (Дата обращения 26.05.2022 г.)

13. Трудовой кодекс Российской Федерации: от 30 декабря 2001 г. N 197-ФЗ (ТК РФ) [Электронный ресурс] / Консультант Плюс. URL: https://www.consultant.ru/document/cons_doc_LAW_34683/ (дата обращения 28.05.2022).

14. ГОСТ 12.2.033-78. Рабочее место при выполнении работ стоя: от 26 апреля 1978 г. № 1100.

15. ГОСТ 12.2.032-78. Рабочее место при выполнении работ сидя: от 26 апреля 1978 г. № 1102.

16. ТОИ Р-45-084-01 Типовая инструкция по охране труда при работе на персональном компьютере.

17. ПЭВМ: ГОСТ 12.2.032-78 ССБТ «Рабочее место при выполнении работ сидя. Общие эргономические требования».

18. ГОСТ Р 50923-96. «Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения».

19. ГОСТ 12.0.003-2015 «Опасные и вредные производственные факторы. Классификация»

20. ГОСТ 12.1.005-88 Система стандартов безопасности труда (ССБТ). Общие санитарно-гигиенические требования к воздуху рабочей зоны.

21. ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности.

22. СН 2.2.4/2.1.8.562–96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории застройки.

23. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95.

24. ГОСТ Р 12.1.019-2017 ССБТ Электробезопасность. Общие требования и номенклатура видов защиты

25. ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ).

26. ГОСТ Р 12.1.019-2009 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты.

27. ГОСТ Р 53692-2009 Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов.

28. ГОСТ 12.3.031-83 «Работы со ртутью. Требования безопасности»

29. ГОСТ 12.1.004-91 Система стандартов безопасности труда (ССБТ). Пожарная безопасность. Общие требования

30. ГОСТ 17.4.3.04-85 Охрана природы (ССОП). Почвы. Общие требования к контролю и охране от загрязнения.

31. СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания.

32. МР 2.2.9.2311 – 07 «Профилактика стрессового состояния работников при различных видах Профессиональной деятельности»

33. ГОСТ Р 50571.4.44-2019. Электроустановки Низковольтные. Часть 4.44. Защита для обеспечения безопасности. Защита от резких отклонений напряжения и электромагнитных возмущений

34. ГОСТ 12.1.044-89. Система стандартов безопасности труда. Пожаровзрывоопасность веществ и материалов

35. ГОСТ 12.1.045-84 ССБТ. Электростатические поля. Допустимые уровни на рабочих местах и требования к проведению контроля.

Приложение А

(обязательное)

Overview on Technologies used in the project

Студент:

Группа	ФИО	Подпись	Дата
8ИМ01	Ким Станислав		

Руководитель ВКР:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Копнов М.В.	к.т.н.		

Консультант-лингвист отделения иностранных языков ШБИП:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИЯ ШБИП	Сидоренко Т.В.	к.пед.н.		

Introduction

Modern approaches to develop Single Page Application (SPA) are under study. SPA is an application placed on one web page to provide for a user more flexible interaction, similar to a desktop application that loads at a time, unlike traditional multi-page approaches used in web projects development. This saves much loading time.

To get acquainted with the concept of SPA and the use it in solving client-server tasks, the goal was set: to develop and implement an adaptive client-server SPA application. A small food delivery service was chosen as the subject area.

To achieve this goal within the framework, it is necessary to solve the following tasks:

1. introducing the Vue.js framework and the main mechanisms for working with the data models,
2. developing a web application components: user pages and administration panel,
3. creating of routing,
4. working with the API requests using Axios library,
5. using the Vuex state management pattern,
6. using the VeeValidate library to validate forms.

Project Description

The project is intended for pre-ordering food in a cafe network. The consumer of the application can be divided into two categories: business as a consumer and end users who make food orders. The main segment of consumers, among users, are people who have limited time for meals, which does not allow them to stand in queues and waste time. An example of such a segment is students. As a rule, students have little time for meals, breaks between couples come at the same time and queues accumulate in fast food cafes, which leads to a lack of time. The project will reduce the time for order maintenance, which will lead to a reduction in the user's time to receive the order. The second group of consumers of the application is business. For business, the project solves the following problems: collects statistics and reports on each cafe and allows to track the dynamics of sales changes, provides the end user with information about sales.

In the course of this project, the architecture and development of an adaptive SPA application - a food delivery service with the availability of:

• the ability to receive and send data from the database (section names, product information, user information) should be implemented,

• sections of products distributed by category,

• basket, in which, depending on the method of receiving the orders, the corresponding fields appear,

• the ability to add products to the cart and place an order, with the calculated cost, by sending data to the server,

• validation of form data,

• sections for logging in to the administration panel with verification,

• sections of tables with all the necessary data that can be edited for the administration panel.

A.1 Overview on Technologies used in the project

A.1.1 HTML

HTML is known as a markup language. Almost all of web pages contain a description of the markup in this language. This language is understandable to browsers; the text that is interpreted as a result is displayed on device screens. This is the code that is used to structure and display a web page and its content. For example, content can be structured within multiple paragraphs, bulleted lists, or using images and data tables.

A.1.2 CSS

CSS — literally stands for cascading style sheets, this language is responsible for the visual representation of documents.

It is mainly used as a means of designing the appearance of web pages written using the HTML language.

CSS provides the ability to determine the color, font, location of any blocks and other tasks of presenting the appearance of web pages. The main purpose for which CSS was developed was to separate the description of the logical structure of a web page (which is produced using HTML or other markup languages) from the description of the appearance of the same web page. This separation makes the development more structured, and therefore more pleasant to perceive.

A.1.3 Bootstrap

Bootstrap is an open and free HTML, CSS and JS framework that is used by web developers to layout responsive website designs and web applications.

It allows you to make up websites several times faster than using "pure" CSS and JavaScript. And in our world, time is a very valuable resource. Another aspect of it is accessibility.

A.1.4 SCSS

Writing CSS, when the stylesheet becomes huge, turns into a more complex and time-consuming process. And in this case, a preprocessor is used. SCSS allows you to use functions that are not available in CSS itself, for example, variables,

nesting, mixins, inheritance and other nice things that return the convenience of writing CSS.

A.1.5 JavaScript

Multi-paradigm programming language. It is a full-fledged dynamic programming language that is applied to an HTML document, and can provide dynamic interactivity on websites. It does not provide low-level means of working with memory, processor, as it was originally focused on browsers in which this is not required.

As for the other features, they depend on the environment in which JavaScript is running. In the browser, JavaScript can do everything related to page manipulation, interaction with the visitor

JavaScript was supposed to be the "younger brother" of Java. However, at the moment everything is different: JavaScript has grown a lot, and now it is a completely independent language, with its own specification.

A.1.6 API

API (Application Programming Interface) is a description of the ways (a set of classes, procedures, functions, structures or constants) that one computer program can interact with another program. It is usually included in the description of an Internet protocol, a software framework, or a standard for calling operating system functions. It is often implemented by a separate software library or an operating system service. There are several types of APIs. This project implements work with the REST API. The REST API interacts using HTTP requests, performing standard functions: creating, updating, reading, deleting records in a resource.

A.1.7 Axios

Axios is used to receive and display data from the API – it is an open source library that allows you to make HTTP requests.

A.1.8 Vue.js

Vue is a progressive framework for creating user interfaces. Unlike monolith frameworks, Vue is designed to be suitable for gradual implementation. Its core

primarily solves the problems of the view level, which simplifies integration with other libraries and existing projects. On the other hand, Vue is also fully suitable for creating complex single-page applications. The Vue framework relies on a component-based approach. The entire application is divided into reusable parts – components (Figure 19). A component can represent not only one thing, but also some part of the application that should work and look the same everywhere.

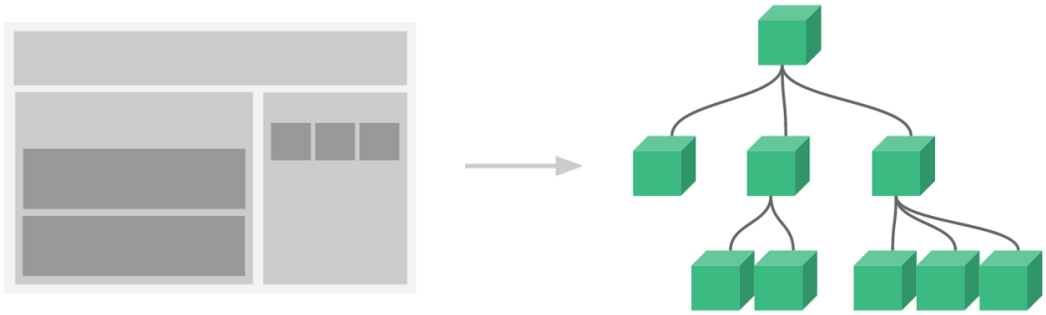

Figure 19 – Splitting a page into components

A.1.9 Vue-router

Vue-Router is a JavaScript package that allows to configure routing for single-page applications. Using SPA has many advantages, but one of the main disadvantages is that all components of a web page are delivered, added or removed using JavaScript without downloading additional HTML pages from the server. This is the essence of SPA, so the main problem is the ability to navigate through the "pages" that users are used to on most websites. Vue-router solves this problem.

A.1.10 Vuex

Vuex is a state management pattern and library for Vue.js. It is a centralized data repository for all components of the application with rules ensuring that the state can only be changed in a predictable way. Using Vuex does not mean that you need to take out all the state in the repository. Although moving most of the state to Vuex will make mutations more explicit and easier to debug, it can also lead to

verbosity and unnecessary complication of logic. If part of the state refers to only one component, it is better to leave it as a local state.

A.1.11 VeeValidate

VeeValidate is a validation library that contains 35 built–in validators, error translation into English, directives and components for validation. The library supports 41 languages for errors. Installing and using it with the right localization is achieved in just a couple of steps.

Приложение Б

(обязательное)

Листинг кода, описывающий работу Vuex

Файл store.js:

```
import { createStore } from 'vuex'
import axios from "axios";

export const store = createStore({
    state:{
        status: '',
        token: localStorage.getItem('token') || '',
        user: {},
        items: [],
        isOrderHidden: false,
        cart: [],
        titles: [],
        users: [],
    },
    mutations:{
        auth_request(state){
            state.status = 'loading'
        },
        auth_success(state, token, user){
            state.status = 'success'
            state.token = token
            state.user = user
        },
        auth_error(state){
            state.status = 'error'
        },
        logout(state){
            state.status = ''
            state.token = ''
        },
        SET_ITEMS_TO_STATE: (state, items) => {
            state.items = items;
        },
        SET_PRODUCT_TO_STATE: (state, item) =>{
            let cloneItem = {...item}
            state.items.push(cloneItem);
        },
        ATTR_TOGGLE_ITEM: (state, item) =>{

            item.isReadonly = !item.isReadonly;
            console.log(item);
        },
        REMOVE_PRODUCT_FROM_STATE: (state, item) =>{
            state.items = state.items.filter(i => i.name !== item.name)
        },
        SET_CART: (state, item) => {
            if (state.cart.length){
                let isItemExists = false;
                state.cart.map(function(product){
                    if(product.name === item.name){
                        isItemExists = true;
                        product.quantity++
```

```
                    }
                })
                if(!isItemExists){
                    state.cart.push(item)
                }
            } else{
                state.cart.push(item)
            }
        },
        DELETE_FROM_CART: (state, index) => {
            state.cart.splice(index, 1);
        },
        CHANGE_MODAL_VALUE: (state) => {
            state.isModalHidden = !state.isModalHidden;
        },
        INCREMENT: (state, index) => {
            state.cart[index].quantity++
        },
        DECREMENT: (state, index) => {
            if(state.cart[index].quantity > 1){
                state.cart[index].quantity--
            }
        },
    },
    actions:{
        login({commit}, user){
            return new Promise((resolve, reject) => {
                commit('auth_request')
                axios({url: 'http://lm.perimeter.games/api/v1/login', data:
user, method: 'POST' })
                    .then(resp => {
                        const token = resp.data.token
                        const user = resp.data.user
                        localStorage.setItem('token', token)
                        axios.defaults.headers.common['Authorization'] =
token
                        commit('auth_success', token, user)
                        resolve(resp)
                    })
                    .catch(err => {
                        commit('auth_error')
                        localStorage.removeItem('token')
                        reject(err)
                    })
            })
        },
        register({commit}, user){
            return new Promise((resolve, reject) => {
                commit('auth_request')
                axios({url: 'http://localhost:3000/register', data: user,
method: 'POST' })
                    .then(resp => {
                        const token = resp.data.token
                        const user = resp.data.user
                        localStorage.setItem('token', token)
                        axios.defaults.headers.common['Authorization'] =
token
                        commit('auth_success', token, user)
                        resolve(resp)
                    })
                    .catch(err => {
```

```
                        commit('auth_error', err)
                        localStorage.removeItem('token')
                        reject(err)
                    })
            })
        },
        logout({commit}){
            return new Promise((resolve) => {
                commit('logout')
                localStorage.removeItem('token')
                delete axios.defaults.headers.common['Authorization']
                resolve()
            })
        },
        GET_TITLES_FROM_API({commit}){
            return axios('
http://lm.perimeter.games/api/v1/database.getsections', {
                method: "GET",
            })
                .then((titles) =>{
                    commit('SET_TITLES_TO_STATE', titles.data);
                    console.log(titles.data);
                    return titles;
                })
                .catch((error) =>{
                    console.log(error)
                    return error;
                })
        },
                DECREMENT_CART_ITEM({commit}, index){
            commit('DECREMENT', index)
        },
        INCREMENT_CART_ITEM({commit}, index){
            commit('INCREMENT', index)
        },
        DELETE_FROM_CART({commit}, index){
            commit('DELETE_FROM_CART', index);
        },
        TOGGLE_ORDER_VALUE:({commit}) =>{
            commit('CHANGE_ORDER_VALUE');
        }
    },
    getters:{
        isLoggedIn: state => !!state.token,
        authStatus: state => state.status,
        TITLES(state){
            return state.titles;
        },
        ITEMS(state){
            return state.items;
        },
        ORDER_VALUE(state){
            return state.isOrderHidden;
        },
        CART(state){
            return state.cart;
        }
    }
})
```

Приложение В

(обязательное)

Листинг кода, описывающий компонент item

Файл item.vue

```
<template>
  <div class="item">
    <img :src=" require('../assets/images/'/* + item_data.thumb*/)"
alt="Изображение товара">
    <div class="item_info">
      <div class="item_description">
        <div class="item_title"><h4>{{item_data.title}}</h4></div>
        <div class="item_bju">{{item_data.utility}}</div>
        <div class="item_desc">{{item_data.description}}</div>
        <div class="item_price">{{item_data.price}} ₽</div>
      </div>
      <div class="item_controller">
        <button class="item_btn"
        @click="addToCart">
          В корзину
        </button>
      </div>
    </div>
  </div>
</template>

<script>
import {mapActions} from "vuex";

export default {
  name: 'item',
  data(){
    return{
    }
  },
  props: {

    item_data: {
      type: Object,
      default() {
```

```
        return{}
      }
    }
  },
  methods: {
    ...mapActions([

    ]),
    addToCart(){
      this.$emit('addToCart',this.item_data)
    },
  }
}
</script>

<style>
</style>
```