

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

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка бэкэнд-части приложения для составления пищевого рациона
УДК 004.451:004.75:004.455:613.2

Студент

Группа	ФИО	Подпись	Дата
8К82	Хрякова Юлия Евгеньевна		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Саврасов Фёдор Витальевич	к.т.н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Гасанов Магеррам Али оглы	д.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева Ирина Леонидовна	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

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

Планируемые результаты освоения ООП по направлению 09.03.04

«Программная инженерия»

Код компетенции	Наименование компетенции
Универсальные компетенции	
УК(У)-1	Способность осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способность определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способность осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способность осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах)
УК(У)-5	Способность воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах
УК(У)-6	Способность управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни
УК(У)-7	Способность поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности
УК(У)-8	Способность создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций
УК(У)-9	Способность проявлять предприимчивость в профессиональной деятельности, в т.ч. в рамках разработки коммерчески-перспективного продукта на основе научно-технической идеи
Общепрофессиональные компетенции	
ОПК(У)-1	Способность использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с прикладной математикой и информатикой
ОПК(У)-2	Способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию образовательного контента, прикладных баз данных
ОПК(У)-3	Способность приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии
ОПК(У)-4	Способность понимать сущность и значение информации в развитии современного общества, осознавать опасность и угрозу, возникающие в этом процессе, соблюдать основные требования информационной безопасности
ОПК(У)-5	Способность использовать основные методы, способы и средства получения, хранения, переработки информации и навыки работы с компьютером как со средством управления информацией

ОПК(У)-6	Способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
ОПК(У)-7	Способность использовать в своей профессиональной деятельности знание иностранного языка
ОПК(У)-8	Способность критически переосмысливать накопленный опыт, изменять при необходимости направление своей деятельности
ОПК(У)-9	Способность получить организационно-управленческие навыки при работе в научных группах и других малых коллективах исполнителей
Профессиональные компетенции	
ПК(У)-1	Способность работать в составе научно-исследовательского и производственного коллектива и решать задачи профессиональной деятельности
ПК(У)-2	Способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения
ПК(У)-3	Способность осуществлять целенаправленный поиск информации о новейших научных и технологических достижениях в информационно-телекоммуникационной сети "Интернет" и в других источниках
ПК(У)-4	Способность собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям
ПК(У)-5	Способность понимать, совершенствовать и применять современный математический аппарат
ПК(У)-6	Способность приобретать и использовать организационно-управленческие навыки в профессиональной и социальной деятельности
ПК(У)-7	Способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы
ПК(У)-8	Способность к реализации решений, направленных на поддержку социально-значимых проектов, на повышение информационной грамотности населения, обеспечения общедоступности информационных услуг
ПК(У)-9	Способность к организации педагогической деятельности в конкретной предметной области (прикладная математика и информатика)

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

УТВЕРЖДАЮ:
 Руководитель ООП
 _____ Чердынцев Е.С.
 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

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

Студенту:

Группа	ФИО
8K82	Хряковой Юлии Евгеньевне

Тема работы:

Разработка бэкэнд-части приложения для составления пищевого рациона

Утверждена приказом директора (дата, номер)

Срок сдачи студентом выполненной работы:	06.06.2022
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Объектом разработки в исследовательской работе является приложение для составления рациона. Режим работы: клиент-сервер. Особые требования к продукту: кроссбраузерность, адаптивный дизайн, модульная структура.
Перечень подлежащих исследованию, проектированию и разработке вопросов	<ol style="list-style-type: none"> 1. Исследование предметной области; 2. Проектирование системы; 3. Разработка системы; 4. Анализ результатов разработки веб-приложения; 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение;

	6. Социальная ответственность.
Перечень графического материала	Презентация в формате *.pptx

Консультанты по разделам выпускной квалификационной работы

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Гасанов Магеррам Али оглы
Социальная ответственность	Мезенцева Ирина Леонидовна

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	28.03.2022
---	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Саврасов Фёдор Витальевич	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К82	Хрякова Юлия Евгеньевна		

Школа Инженерная школа информационных технологий и робототехники
 Кафедра Отделение информационных технологий
 Направление подготовки 09.03.04 Программная инженерия
 Уровень образования Бакалавр
 Период выполнения Весенний семестр 2021/2022 учебного года

Форма представления работы:

Бакалаврская работа

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

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	6 июня 2022 г.
--	----------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	Написание основной части	60
	Написание части финансового менеджмента, ресурсоэффективности и ресурсосбережения	20
	Написание части социальной ответственности	20

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Саврасов Фёдор Витальевич	К.Т.Н.		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Чердынцев Евгений Сергеевич	К.Т.Н.		

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

Студенту:

Группа	ФИО
8К82	Хряковой Юлии Евгеньевне

Школа	ИШИТР	Отделение школы (НОЦ)	Отделение информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

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

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Оклад руководителя – 47317 руб. Оклад инженера – 20000 руб.
2. Нормы и нормативы расходования ресурсов	Премимальный коэффициент руководителя 30%; премиальный коэффициент инженера 20%; доплаты и надбавки руководителя 30%; доплаты и надбавки руководителя 30%; дополнительной заработной платы 12%; накладные расходы 16%; районный коэффициент 1,3.
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 потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	SWOT-анализ разработанной стратегии
2. Планирование и формирование бюджета научных исследований	Определение структуры работы. Расчет трудоемкости выполнения работ. Подсчет бюджета исследования
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Рассчитать показатели финансовой эффективности, ресурсоэффективности и эффективности исполнения

Перечень графического материала (с точным указанием обязательных чертежей):

<ol style="list-style-type: none"> Оценка конкурентоспособности технических решений Матрица SWOT Альтернативы проведения НИ График проведения и бюджет НИ Оценка ресурсной, финансовой и экономической эффективности НИ
--

Дата выдачи задания для раздела по линейному графику	03.03.2022
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Гасанов Магеррам Али оглы	д.э.н.		03.03.2022

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К82	Хрякова Юлия Евгеньевна		03.03.2022

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
8K82		Хряковой Юлии Евгеньевне	
Школа	ИШИТР	Отделение (НОЦ)	Отделение информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Тема ВКР:

Разработка бэкэнд-части приложения для составления пищевого рациона	
Исходные данные к разделу «Социальная ответственность»:	
<p>Введение</p> <ul style="list-style-type: none"> - Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. - Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации 	<p>Объект исследования: платформа разработки серверной части приложения Node.js, фреймворк для разработки клиентской части приложения React. Область применения: информационные технологии, разработка веб-приложений. Рабочая зона: комната, учебная аудитория. Размеры помещения: 8*10 м. Количество и наименование оборудования рабочей зоны: персональный компьютер с установленными Node.js и React, периферия.</p>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<p>1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения:</p> <ul style="list-style-type: none"> - специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; - организационные мероприятия при компоновке рабочей зоны. 	<p>ГОСТ 12.2.032-78. Рабочее место при выполнении работ сидя; ГОСТ 21889-76. Система "Человек-машина". Кресло человека-оператора; Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022).</p>
<p>2. Производственная безопасность при разработке проектного решения:</p> <ul style="list-style-type: none"> - Анализ выявленных вредных и опасных производственных факторов 	<p>Вредные производственные факторы в соответствии с ГОСТ 12.0.003-2015:</p> <ul style="list-style-type: none"> • отсутствие или недостаток необходимого естественного освещения; • эмоциональные перегрузки; • статические физические перегрузки; • перенапряжение зрительных анализаторов; • монотонность трудового процесса. <p>Опасные производственные факторы в соответствии с ГОСТ 12.0.003-2015:</p> <ul style="list-style-type: none"> • электрический ток. <p>Требуемые средства коллективной и индивидуальной защиты от выявленных факторов:</p>

	<ul style="list-style-type: none"> • удобное посадочное место; • изоляция электроники от попадания пыли и влаги; • источник искусственного освещения.
3. Экологическая безопасность при разработке проектного решения:	<p>Воздействие на литосферу: утилизация офисной мебели и компьютерных комплектующих.</p> <p>Воздействие на атмосферу: выброс вредных веществ на предприятиях по производству комплектующих.</p> <p>Воздействие на гидросферу: атмосферные осадки, приносящие в гидросферу загрязнения от производства комплектующих.</p>
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения:	<p>Возможные ЧС: пожары, наводнения, аварии на коммунальных системах жизнеобеспечения, аварии на электроэнергетических системах.</p> <p>Наиболее типичная ЧС: авария на коммунальной системе жизнеобеспечения.</p>
Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева Ирина Леонидовна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8K82	Хрякова Юлия Евгеньевна		

Реферат

Выпускная квалификационная работа содержит 88 страниц, 31 рисунок, 21 таблицу и 16 литературных источников.

Ключевые слова: рацион питания, веб-приложение, архитектура приложения, Node.js.

Объектом исследования является веб-приложение для составления пищевого рациона.

Цель работы – спроектировать и разработать бэкэнд-часть приложения для составления пищевого рациона.

В процессе исследования проводилось изучение конкурентных решений на российском рынке. В результате исследования было выявлено, что существующие программные продукты имеют закрытый исходный код, являются платными или имеют ограниченный функционал, и, как следствие, не пользуются высокой популярностью у целевой аудитории.

Результатом работы является спроектированная и реализованная бэкэнд-часть веб-приложения.

В первом разделе представлено описание предметной области и функций разрабатываемой системы.

Во втором разделе проводится декомпозиция основных процессов, выбор средств программной разработки, концептуальное и логическое проектирование базы данных.

Третий раздел содержит информацию о способах и технологиях реализации серверной части приложения, а также описание ее архитектуры.

В четвертом разделе описываются результаты разработки системы и итоговый функционал.

Оглавление

Перечень терминов и условных обозначений	14
Введение	15
Глава 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ	16
1.1. Общая информация	16
1.2. Постановка задачи	18
1.3. Общее описание функций системы	21
Выводы по главе	22
Глава 2. ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ	23
2.1. Проектирование информационной системы	23
2.1.1. Диаграмма BPMN	23
2.2. Выбор программно-технических средств	25
2.2.1. Выбор языка программирования серверной части	25
2.2.2. Выбор СУБД	26
2.3. Проектирование базы данных	27
2.3.1. Концептуальное проектирование	27
2.3.2. Логическое проектирование	34
2.3.2.1. Логическая схема предметной области	34
Выводы по главе	36
Глава 3. РАЗРАБОТКА СИСТЕМЫ	37
3.1. Архитектура системы	37
3.2. Обработка изображений	40
3.3. Авторизация и регистрация	41
3.4. Пользователи	42
3.5. Измерения	43
3.6. Рецепты	43
3.7. Цель	43

3.8. Уровень активности	44
3.9. Продукт	44
3.10. Рацион	45
3.11. Прием пищи	45
3.12. Категория продукта	45
3.13. Деплой приложения	45
Выводы по главе	47
Глава 4. АНАЛИЗ РЕЗУЛЬТАТОВ РАЗРАБОТКИ И ТЕСТИРОВАНИЕ	48
4.1. Результат создания базы данных	48
4.2. Тестирование API сервера	48
4.2.1. Авторизация и регистрация	48
4.2.2. Получение данных о пользователе	50
4.2.3. Создание приема пищи	51
4.2.4. Создание рациона	52
4.2.5. Создание рецепта	53
4.2.6. Удаление из базы данных	54
Выводы по главе	55
ГЛАВА 5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	56
5.1. Оценка перспективности проведения исследований	56
5.1.1. Анализ конкурентных технических решений	56
5.2. SWOT-анализ	58
5.3. Планирование работ по научно-техническому исследованию	62
5.3.1. Структура работ в рамках научного исследования	62
5.3.2. Определение трудоемкости выполнения работ	63
5.3.3. Разработка графика проведения научного исследования	64
5.4. Бюджет технического проекта	67
	12

5.4.1. Расчет материальных затрат	68
5.4.2. Расчет затрат на оборудование	68
5.4.3. Основная заработная плата исполнителей	69
5.4.4. Расчет дополнительной заработной платы	70
5.4.5 Отчисления во внебюджетные фонды	71
5.4.6. Накладные расходы	71
5.4.7. Формирование бюджета затрат научно-исследовательского проекта	72
5.5. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	73
Выводы по разделу	74
Глава 6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ	75
6.1. Правовые и организационные вопросы обеспечения безопасности	76
6.2. Производственная безопасность	77
6.2.1. Анализ опасных и вредных факторов и обоснование мероприятий по снижению их воздействия	78
6.2.1.1. Отсутствие или недостаток необходимого естественного освещения	78
6.2.1.2. Эмоциональные перегрузки	79
6.2.1.3. Перенапряжение зрительных анализаторов	80
6.2.1.4. Статические физические перегрузки	80
6.2.1.5. Монотонность трудового процесса	80
6.2.1.6. Электрический ток	81
6.3. Экологическая безопасность	82
6.4. Безопасность в чрезвычайных ситуациях	83
Выводы по разделу	85
Заключение	86
Список литературы	87
Приложение А	89
Приложение Б	94
	13

Перечень терминов и условных обозначений

1. **Веб-сервер** – сервер, который принимает HTTP-запросы и возвращает HTTP-ответы клиентской части приложения.
2. **Аутентификация** – процесс проверки подлинности введенных данных (например, записанные в базу данные сравниваются для подтверждения идентификации пользователя).
3. **HTTP** – HyperText Transfer Protocol – протокол передачи данных, изначально в формате HTML, в настоящее время служит для передачи произвольных данных.
4. **HTML** – язык разметки для просмотра веб-страниц в браузере.
5. **URL** – адрес, выданный отдельному ресурсу в интернете, указывающий его местоположение.
6. **CRUD** – аббревиатура для обозначения четырех операций при работе с данным (create, read, update, delete).
7. **SQL** – язык программирования, применяемый для выполнения CRUD-операций с данными в реляционной базе данных.
8. **СУБД** – разновидность программного обеспечения, позволяющего создавать базы данных и управлять ими.
9. **JSON** – легковесный формат обмена данными, хранящий в себе методы разбора объектной нотации языка программирования JavaScript.
10. **Фреймворк** – программный продукт, который определяет структуру создаваемого приложения для облегчения разработки.

Введение

В настоящее время все больше людей сталкиваются с необходимостью следить за своим питанием. Это обуславливается, во-первых, растущим интересом к спорту на фоне снижения физической активности и перехода к сидячему образу жизни. Для достижения своей цели, например, набора мышечной массы, недостаточно только регулярных занятий спортом, важно также следить за тем, чтобы организм спортсмена получал необходимое количество калорий и макронутриентов [1] и имел ресурсы для наращивания мышечной массы и сжигания жиров.

Во-вторых, существует категория людей, вынужденных следить за питанием по состоянию здоровья, в таком случае необходимо составлять свой рацион в соответствие со списками запрещенных и рекомендованных продуктов.

Помимо двух названных выше категорий людей, любому человеку будет полезно следить за своим питанием для предотвращения различных заболеваний, например, ожирения или гастрита, вызванных несбалансированным питанием [2], недопущения ухудшения физической формы, а также для формирования сознательности в отношении питания.

Все вышеперечисленное требует внушительного количества временного ресурса для самостоятельного поиска информации о необходимом количестве КБЖУ, диетах, количества макронутриентов и калорий в продуктах, а также составления рациона вручную.

На данный момент в мире существует тенденция цифровизации инструментов и методов решения различных проблем, коснувшаяся и составления рациона питания.

Глава 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Общая информация

Поставленная выше проблема имеет высокую актуальность, в связи с чем было принято решение о создании веб-приложения Rationalist для помощи людям в сокращении временных затрат на составление рациона питания. Для первоначального анализа требований к разрабатываемому ПО был проведен обзор конкурентных решений на российском рынке.

Первым конкурентом является «ЗОЖНИК – дневник зожника», являющийся кроссплатформенным приложением для ведения здорового образа жизни, большая часть функционала которого предназначена для спортсменов. Приложение включает в себя такие функции как: подсчет КБЖУ по приемам пищи за день, составление индивидуального дневника питания и тренировок, добавление своих продуктов и рецептов. Данное приложение не лишено недостатков. Опираясь на рейтинг в 4,4 балла, составленный на основе пользовательских оценок в магазине приложений, можно выделить несколько ключевых недостатков [3]:

- Ограниченный бесплатный функционал (доступно создание не более 3-х рационов питания, отсутствие доступа к базе полезных продуктов).
- Нестабильная работа системы у многих пользователей, проблемы при загрузке из магазина приложений.
- Малое количество продуктов, доступных для составления рациона, в связи с чем пользователи вынуждены добавлять их самостоятельно.
- Отсутствие рекомендаций по диетам для пользователей с заболеваниями ЖКТ.
- Закрытый исходный код.

Пример добавления продуктов в рацион представлен на рисунке 1.

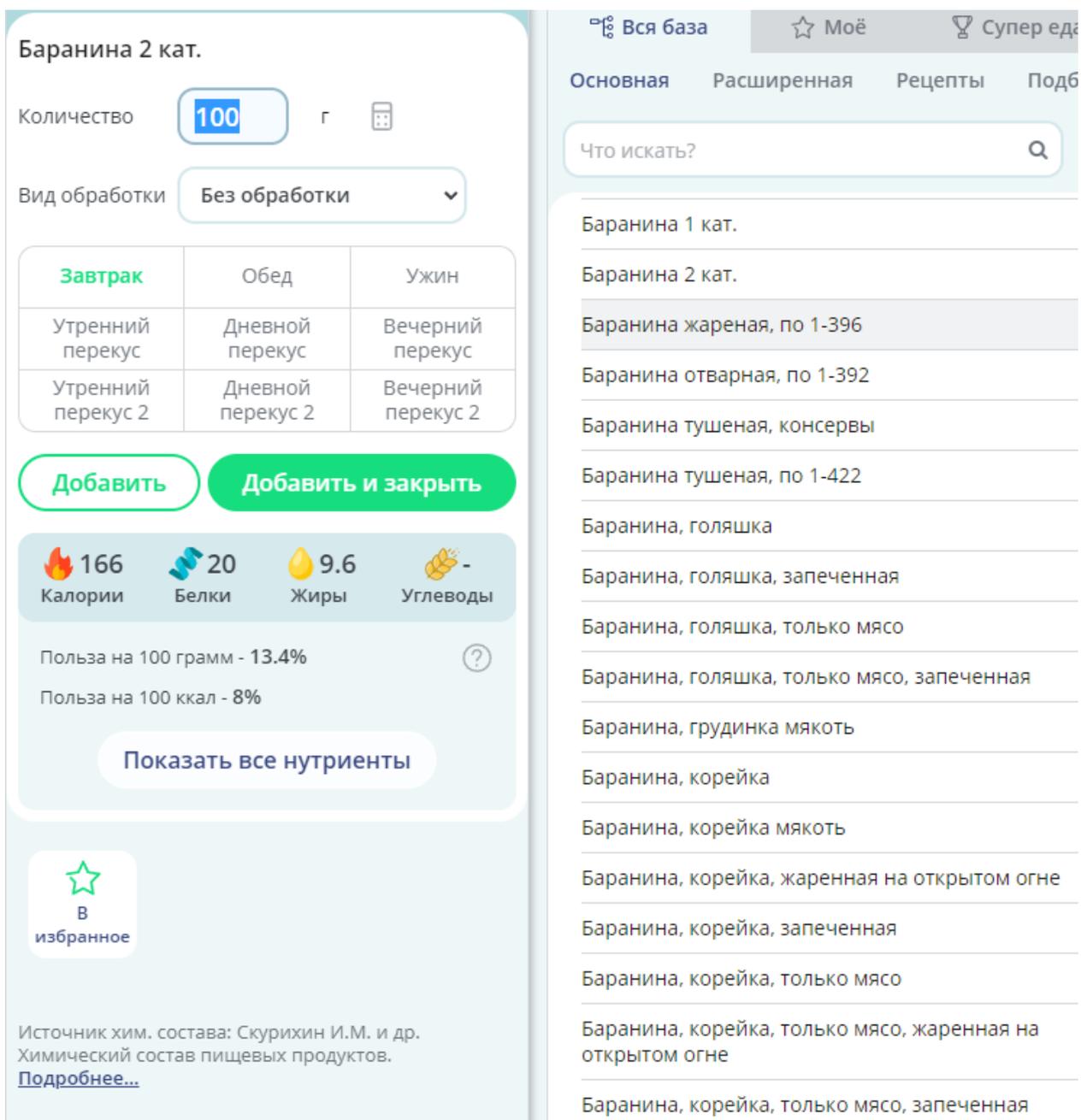


Рисунок 1. Приложение "Зожник". Меню добавления продукта

Вторым конкурентом является веб-приложение «3x15». Данный сервис имеет гораздо меньший функционал:

- Отсутствие авторизации и регистрации, и, как следствие, личного кабинета с сохраненными данными пользователя и рационами.
- Перегруженность ресурса за счет размещения всей информации на одной странице.
- Отсутствие системы отслеживания прогресса физических показателей.

Пример составления рациона представлен на рисунке 2.

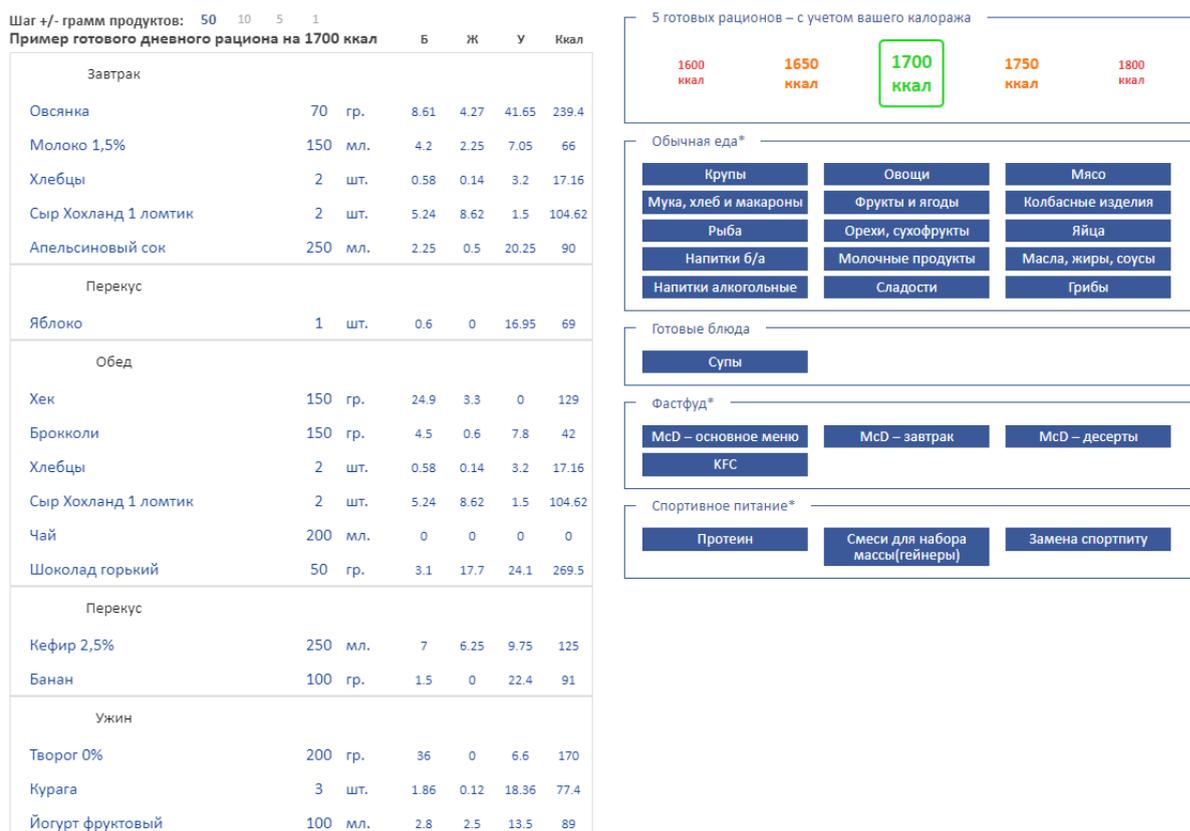


Рисунок 2. Приложение "3x15". Составление рациона

На основании вышеприведенного анализа было принято решение о создании собственного веб-приложения для составления рациона.

1.2. Постановка задачи

Разрабатываемое веб-приложение является многокомпонентным, так как сочетает в себе программно-аппаратную (серверную) и клиентскую части приложения. Система позволяет пользователю как составлять рационы питания вручную, самостоятельно выбирая продукты, так и получать готовый рацион на основе диеты, которой придерживается пользователь, а также создавать собственные рецепты и просматривать рецепты других пользователей.

Расчет необходимого количества калорий производится на основе личных данных пользователя (рост, вес, возраст, пол, уровень активности, цель) по следующим формулам:

- Формула Миффлина-Сан Жеора
- Формула Харриса-Бенедикта
- Формула калорийности ВОЗ

Пользователь будет видеть результаты вычисления по 3-м формулам, а также средний результат, который в дальнейшем будет использоваться для составления рациона.

Расчет БЖУ производится на основе необходимого количества калорий для пользователя [4] и норм, приведенных в таблице 1.

Таблица 1. Суточная норма макронутриентов

Макронутриенты	Набор массы	Поддержание веса	Похудение
Белки	25-35 %	25-35 %	40-50 %
Жиры	15-25 %	25-35 %	30-40 %
Углеводы	40-60 %	30-50 %	30-50 %

Подсчет содержания КБЖУ в составленном рационе будет произведен на основе таблиц содержания КБЖУ в продуктах и их массе, добавленной пользователем в рацион.

Рекомендации по составлению рациона при наличии заболевания, подразумевающего соблюдение диеты, будут составлены на основе лечебных столов (диет) по Певзнеру [5], приведенных в таблице 2.

Таблица 2. Столы по Певзнеру

Диета	Показания
Стол № 0	Операции на органы пищеварения
Стол № 1	6-12 месяцев после обострения язвенной болезни, гастриты с повышенной кислотностью
Стол № 1a	Обострения хронического гастрита с повышенной кислотностью и язвенной болезни

Стол № 1б	Затихание хронических гастритов с повышенной кислотностью и обострения язвенной болезни
Стол № 2	Хронические колиты и гастриты с пониженной кислотностью
Стол № 3	Атонические запоры
Стол № 4	Острые заболевания кишечника в период продолжающегося поноса
Стол № 4а	Колиты с процессами брожения
Стол № 4б	Хронические колиты затухающей стадии
Стол № 4в	Острые заболевания кишечника, хронические заболевания кишечника
Стол № 5	Заболевания желчного пузыря, печени и желчевыводящих путей вне обострения
Стол № 5а	Хронические панкреатиты
Стол № 6	Подагра, почечнокаменная болезнь
Стол № 7	Хронические заболевания почек без явлений почечной недостаточности
Стол № 7а	Острые почечные заболевания
Стол № 7б	Затихание острого воспаления в почках
Стол № 8	Ожирение
Стол № 9	Сахарный диабет в легкой и средней формах
Стол № 10	Заболевание сердечно-сосудистой системы с недостаточностью кровообращения
Стол № 11	Туберкулезы, истощение после инфекций, травм и операций
Стол № 12	Функциональное заболевание нервной системы
Стол № 13	Острые инфекционные заболевания
Стол № 14	Фосфатурия
Стол № 15	Заболевания, не требующих лечебных диет

В зависимости от наличия диеты пользователя, продукты, доступные для добавления в рацион будут отмечены: запрещенные – красным, рекомендованные – зеленым.

Результатом процессов описанной предметной области будет являться сокращение временных затрат на составление рациона пользователем.

1.3. Общее описание функций системы

В первую очередь, для пользователей необходимо определить роли и возможности этих ролей.

В процессе работы было выделено 3 роли:

- неавторизованный пользователь – имеет возможность создания рациона (без сохранения);
- авторизованный пользователь – имеет возможность создать рацион, просмотреть свои рационы, ввести свои данные (рост, вес, пол, возраст, и т.п.), измерения и их динамику, создать собственный рецепт и посмотреть свои рецепты и рецепты других пользователей, а также сменить данные для авторизации;
- администратор – имеет возможность изменить формулы расчета КБЖУ, добавить продукты в базу данных, изменить данные о продуктах, добавить диету, изменить данные о диете.

Система должна позволять неавторизованному пользователю возможности регистрации и авторизации, создание рациона питания без сохранения, а также просмотр информации о сервисе. При этом пользователь также будет вводить личные данные для разового составления рациона.

Авторизованный пользователь должен иметь возможности редактирования личных данных, создания рационов, просмотра собственных рационов, ввода веса и измерений и отслеживания их динамики по датам, создания собственных рецептов: ввод описания рецепта, фотографии готового блюда, этапы приготовления, состава, итогового КБЖУ, возможность просмотреть свои рецепты и рецепты других пользователей.

Выводы по главе

В данном разделе была выявлена целевая аудитория сервиса, проведен анализ конкурентов, выделены сильные и слабые стороны их продуктов. Были определены роли пользователей и соответствующий им функционал веб-приложения.

Глава 2. ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ

2.1. Проектирование информационной системы

2.1.1. Диаграмма BPMN

Наибольшей трудностью в реализации обладает разработка процесса создания рациона, большое количество взаимодействий пользователя с системой требует детальной декомпозиции для лучшего понимания процесса, и, как следствие, избежание ошибок при написании программной части как со стороны клиента, так и со стороны сервера.

С помощью диаграммы BPMN описан процесс создания рациона. Инициатором составления рациона может выступать как неавторизованный, так и авторизованный пользователь.

Диаграмма в нотации BPMN представлена на рисунке 3.

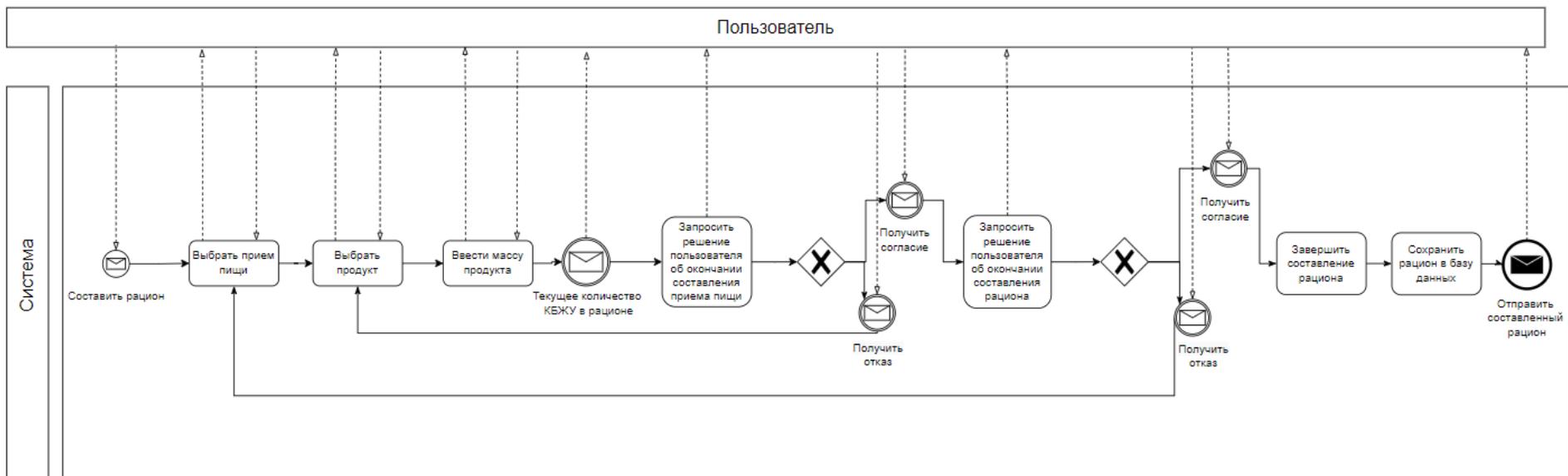


Рисунок 3. Процесс составления рациона пользователем

Описанный на диаграмме процесс в своей работе основывается на двух циклах. После инициализации пользователем составления рациона следует первый (внешний) цикл.

Данный цикл отвечает за составление рациона путем добавления приемов пищи столько раз, сколько это будет необходимо пользователю. Выход из этого цикла производится тогда, когда система получает от пользователя подтверждение об окончании процесса.

Во внутреннем цикле описан процесс добавления продуктов в один прием пищи до тех пор, пока система не получит от пользователя подтверждение об окончании составления одного приема пищи. Внутри этого цикла пользователь выбирает продукт, указывает его массу, после чего система отображает рассчитанное КБЖУ в массе выбранного продукта, а также обновляет информацию об общем количестве КБЖУ в рационе, после чего пользователь может прекратить составление приема пищи и вернуться во внешний цикл или продолжить составление, вернувшись в начало внутреннего.

2.2. Выбор программно-технических средств

2.2.1. Выбор языка программирования серверной части

Веб-сервер – сервер, принимающий HTTP-запрос от клиентов, в нашем случае – веб-браузера, и отправляющий им HTTP-ответы, содержащий HTML-страницу, изображения, файлы или другие данные. Также веб-сервер может выполнять и дополнительные задачи, такие как:

- Аутентификация пользователей.
- Использование защищенных протоколов передачи данных (HTTP Security – HTTPS).
- Ведение журнала запросов.

В ходе анализа языков разработки было принято решение использовать один язык разработки для серверной и клиентской части, так как, во-первых, если разработка ведется в команде, то разработчики могут

проводить перекрестное code-review и более глубоко понимать решения задач друг друга, и, во-вторых, если разработка ведется одним человеком, то он может вести разработку обеих частей приложения на едином языке программирования.

Так как для разработки клиентской части, в большинстве случаев, выбирается язык программирования JavaScript, то, исходя из вышеописанного, для серверной части приложения следует остановить свой выбор на этом же языке. По данным опроса пользователей ресурса «Stack Overflow» за 2020 год [6], среди самых популярных технологий для серверной разработки только одна использует JavaScript – Node.js, соответственно, именно эта технология была выбрана для создания серверной части приложения.

Также, среди преимуществ Node.js можно выделить использование движка Google V8 JavaScript, компилирующий функции, написанные на JS, в машинный код, обеспечивая ускорение и упрощение реализации. Помимо этого, Node.js асинхронно обрабатывает запросы, не блокируя потоки данных при поступлении нового запроса, что сокращает время отклика в несколько раз [7].

Также для платформы Node.js необходима библиотека Nodemon, которая в автоматическом режиме перезапускает сервер после внесения изменений, избавляя разработчика от процесса перезапуска сервера в ручном режиме.

2.2.2. Выбор СУБД

В качестве СУБД была выбрана свободно распространяемая объектно-реляционная система PostgreSQL, так как, во-первых, она изначально поддерживает большое разнообразие типов данных, включая JSON, hstore и XML, и, во-вторых, поддерживает управление параллельным доступом посредством многоверсионности (MVCC), что означает

предоставление одновременного доступа к базе данных множеству агентов на чтение и запись без блокирования процессов [8].

2.3. Проектирование базы данных

2.3.1. Концептуальное проектирование

Анализируя предметную область, определим ключевые сущности:

- «Пользователь»
- «Данные пользователя для входа»
- «Рецепт»
- «Продукт»
- «Рацион»
- «Прием пищи»
- «Категория продукта»
- «Шаг рецепта»
- «Измерение»
- «Уровень активности»
- «Цель»
- «Роль»

Вышеперечисленные сущности имеют следующие отношения:

- «Данные пользователя для входа» содержат «Роль».
- «Пользователь» добавляет «Измерение».
- «Пользователь» выбирает «Цель».
- «Пользователь» имеет «Данные пользователя для входа».
- «Пользователь» выбирает «Уровень активности».
- «Пользователь» создает «Рецепт».
- «Пользователь» создает «Рацион».
- «Рецепт» включает «Шаг рецепта».
- «Рецепт» содержит «Продукт».
- «Продукт» имеет Категорию продукта.
- «Рацион» содержит «Прием пищи».

- «Рацион» содержит «Продукт».
- «Прием пищи» содержит «Продукт».

Далее опишем связи между сущностями, где сущности обозначены прямоугольником, а связи – ромбом.

1. «Данные пользователя для входа» содержат «Роль» (рисунок 4). Связь М:1, т.к. одни данные пользователя для входа могут содержать только одну роль, но одна роль может принадлежать нескольким данным пользователей. Одна роль может не принадлежать ни одним данным пользователей, но данные пользователя должны обязательно иметь роль, поэтому сущность «Роль» имеет обязательный класс принадлежности, а сущность «Данные пользователя для входа» – необязательный.



Рисунок 4. Связь "«Данные пользователя для входа» содержат «Роль»"

2. «Пользователь» добавляет «Измерение» (рисунок 5). Связь М:М, т.к. пользователь может добавлять несколько измерений, и измерения могут принадлежать нескольким пользователям. У пользователя может не быть измерений, поэтому сущность «Измерение» имеет необязательный класс принадлежности. «Измерение» может не принадлежать ни одному пользователю, поэтому у сущности «Пользователь» необязательный класс принадлежности.



Рисунок 5. Связь "«Пользователь» добавляет «Измерение»"

3. «Пользователь» выбирает «Цель» (рисунок 6). Связь М:1, т.к. один пользователь может выбрать только одну цель, а одна цель может быть выбрана несколькими пользователями. «Пользователь» обязан иметь цель,

поэтому сущность «Пользователь» имеет обязательный класс принадлежности, а определенная цель может быть не выбрана ни одним пользователем, поэтому сущность «Цель» имеет необязательный класс принадлежности.



Рисунок 6. Связь "«Пользователь» выбирает «Цель»"

4. «Пользователь» имеет «Данные пользователя для входа» (рисунок 7). Связь 1:1, т.к. один пользователь может иметь только одни данные для входа, также как и данные для входа могут принадлежать только одному пользователю, поэтому сущности «Пользователь» и «Данные пользователя для входа» имеют обязательный класс принадлежности.



Рисунок 7. Связь "«Пользователь» имеет «Данные пользователя для входа»"

5. «Пользователь» выбирает «Уровень активности» (рисунок 8). Связь M:1, т.к. один пользователь может выбрать только один уровень активности, а один уровень активности могут выбрать несколько пользователей. «Пользователь» обязан иметь уровень активности, поэтому сущность «Пользователь» имеет обязательный класс принадлежности, а определенный уровень активности может быть не выбран ни одним пользователем, поэтому сущность «Уровень активности» имеет необязательный класс принадлежности.



Рисунок 8. Связь "«Пользователь» выбирает «Уровень активности»"

6. «Пользователь» создает «Рецепт» (рисунок 9). Связь 1:М, т.к. один пользователь может создать много рецептов, а один рецепт может быть создан только одним пользователем. «Пользователь» может не создавать рецепты, но рецепт обязательно создан пользователем, поэтому сущность «Пользователь» имеет обязательный класс принадлежности, а сущность «Рецепт» – необязательный.



Рисунок 9. Связь "«Пользователь» создает «Рецепт»"

7. «Пользователь» создает «Рацион» (рисунок 10). Связь 1:М, т.к. один пользователь может создать много рационов, а один рацион может быть создан только одним пользователем. «Пользователь» может не создавать рационы, но рацион обязательно должен быть создан пользователем, поэтому сущность «Пользователь» имеет обязательный класс принадлежности, а сущность «Рацион» – необязательный.



Рисунок 10. Связь "«Пользователь» создает «Рацион»"

8. «Рецепт» включает «Шаг рецепта» (рисунок 11). Связь 1:М, т.к. один рецепт может иметь несколько шагов, а шаг может принадлежать только одному рецепту. «Рецепт» обязательно имеет шаги, а шаг не может существовать без рецепта, поэтому сущности «Рецепт» и «Шаг рецепта» имеют обязательный класс принадлежности.



Рисунок 11. Связь "«Рецепт» имеет «Шаг рецепта»"

9. «Рецепт» содержит «Продукт» (рисунок 12). Связь М:М, т.к. один рецепт может включать в себя несколько продуктов, и продукт может принадлежать нескольким рецептам. «Рецепт» обязательно содержит продукты, а продукт может не принадлежать ни одному рецепту, поэтому сущность «Продукт» имеет обязательный класс принадлежности, а «Рецепт» – необязательный.



Рисунок 12. Связь "«Рецепт» содержит «Продукт»"

10. «Продукт» имеет Категорию продукта (рисунок 13). Связь М:1, т.к. продукт может иметь только одну категорию, а категория продукта может принадлежать нескольким продуктам. «Продукт» обязательно имеет категорию, и категория продукта обязательно принадлежит хотя бы одному продукту, следовательно, сущности «Продукт» и «Категория продукта» имеют обязательный класс принадлежности.



Рисунок 13. Связь "«Продукт» имеет Категорию продукта"

11. «Рацион» содержит «Прием пищи» (рисунок 14). Связь М:М, т.к. рацион может содержать несколько приемов пищи, и прием пищи может принадлежать нескольким рационам. «Рацион» обязательно содержит в себе прием пищи, но определенный прием пищи не обязательно содержится в рационе, таким образом, сущность «Рацион» имеет необязательный класс принадлежности, а сущность «Прием пищи» – обязательный.



Рисунок 14. Связь "«Рацион» содержит «Прием пищи»"

12. «Рацион» содержит «Продукт» (рисунок 15). Связь М:М, т.к. рацион может содержать несколько продуктов, и продукт может принадлежать нескольким рационам. «Рацион» обязательно содержит в себе продукт, но определенный продукт не обязательно содержится в рационе, таким образом, сущность «Рацион» имеет необязательный класс принадлежности, а сущность «Продукт» – обязательный.



Рисунок 15. Связь "«Рацион» содержит «Продукт»"

13. «Прием пищи» содержит «Продукт» (рисунок 16). Связь М:М, т.к. прием пищи может содержать несколько продуктов, и продукт может принадлежать нескольким приемам пищи. «Прием пищи» не обязательно содержит в себе продукт, и определенный продукт не обязательно содержится в приеме пищи, таким образом, сущности «Прием пищи» и «Продукт» имеют необязательный класс принадлежности.



Рисунок 16. Связь "«Прием пищи» содержит «Продукт»"

После описания полученных связей получается ER-диаграмма в нотации Мартина, представленная на рисунке 17.

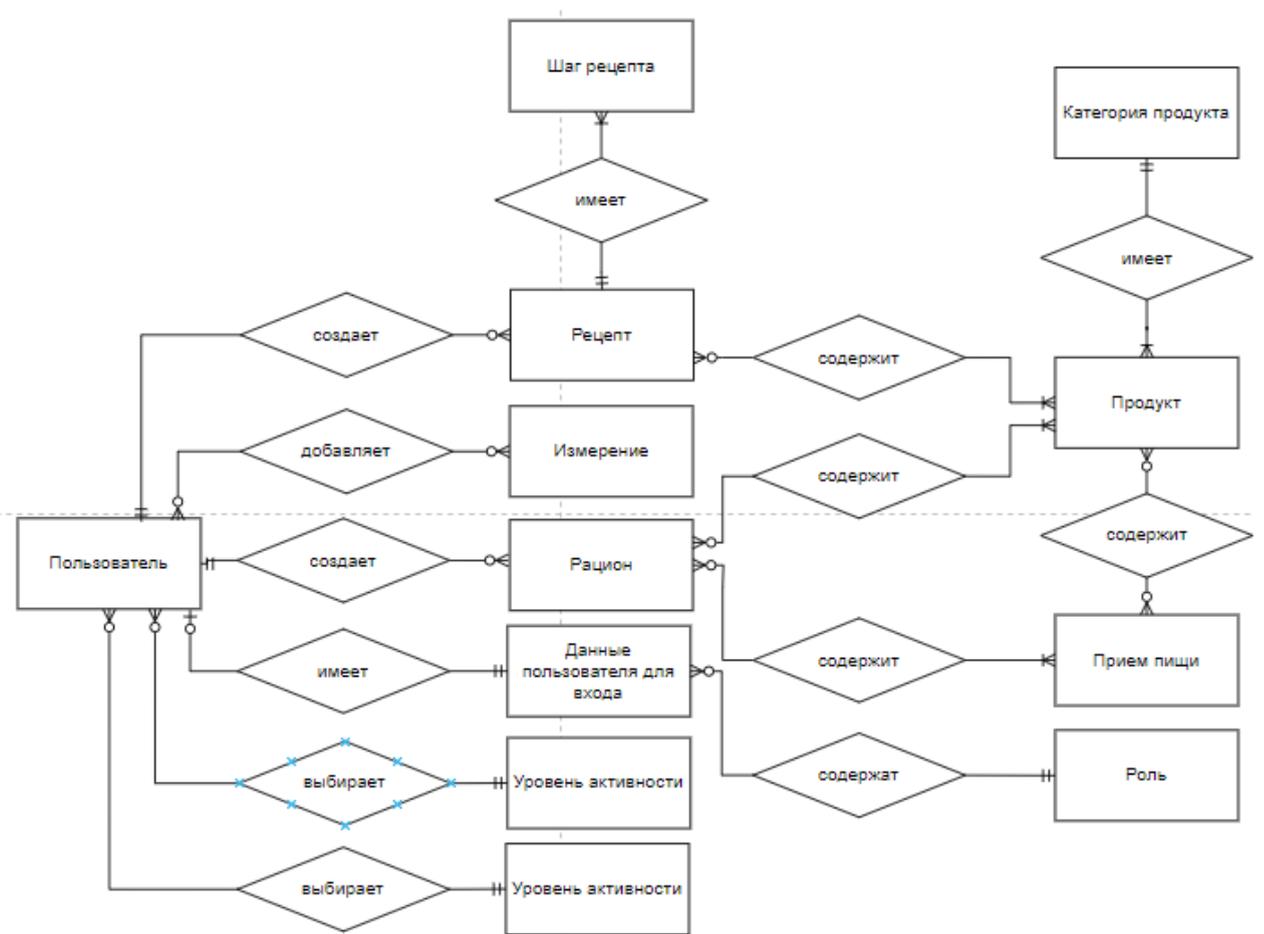


Рисунок 17. ER-диаграмма в нотации Мартина

2.3.2. Логическое проектирование

2.3.2.1. Логическая схема предметной области

После создания ER-диаграммы была спроектирована логическая модель предметной области в нотации Мартина. Логическая модель расширяет концептуальную с помощью определения для сущностей их атрибутов, уточняет состав сущностей и взаимосвязи между ними.

В процессе построения концептуальной диаграммы было выявлено 5 связей между сущностями вида М:М, нормализуем данные отношения путем введения промежуточных таблиц:

- «Пользователь_измерение» («Пользователь» и «Измерение»)
- «Рецепт_продукт» («Рецепт» и «Продукт»)
- «Рацион_прием_продукт» («Рацион», «Прием пищи» и «Продукт»)

Спроектированная логическая модель предметной области представлена на рисунке 18.

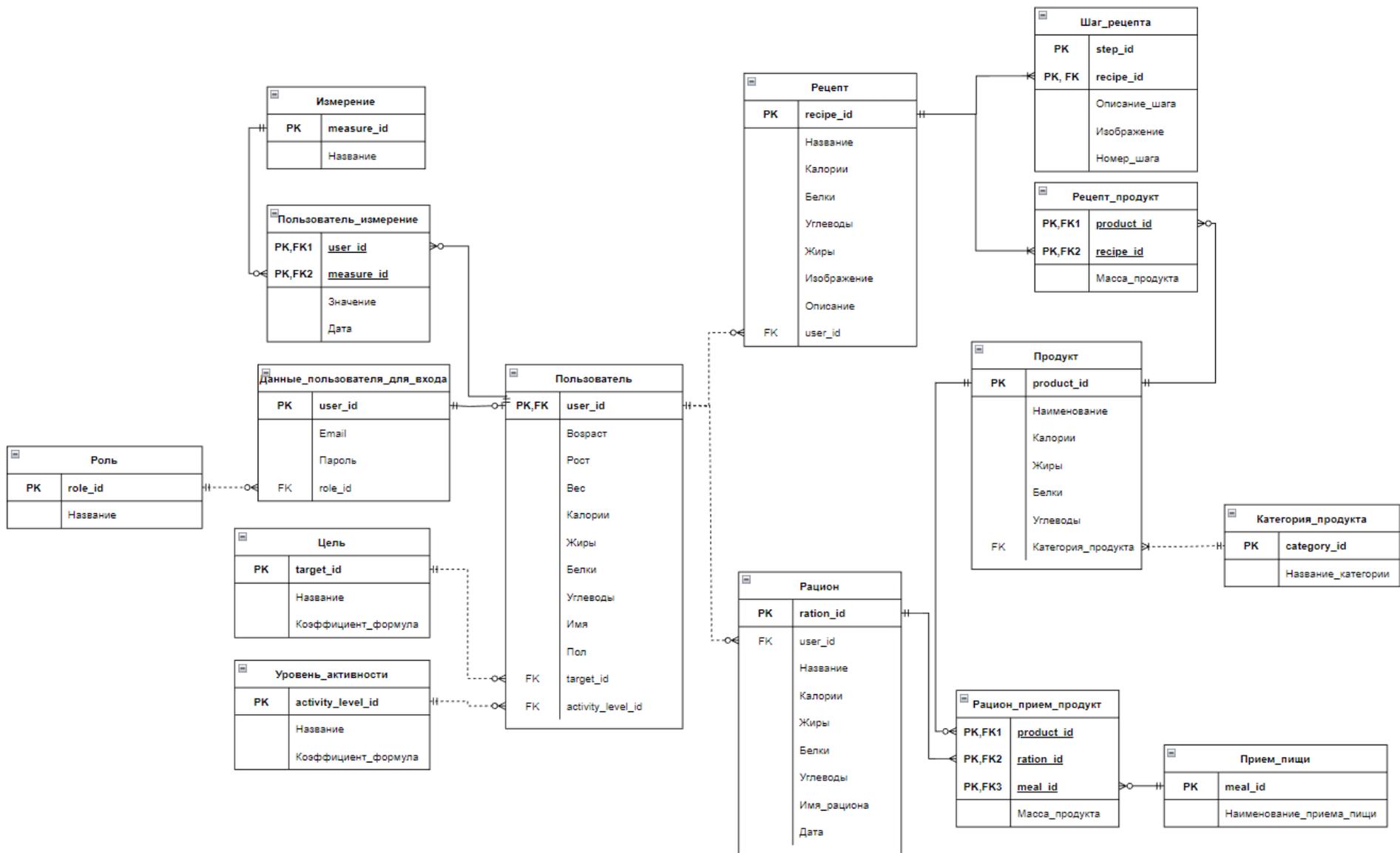


Рисунок 18. Логическая модель предметной области

На данной модели можно выделить 2 типа связей: идентифицирующая и неидентифицирующая. Идентифицирующая связь – это связь, при которой экземпляр сущности-потомка однозначно определяется своей связью с сущностью-родителем, такая связь обозначается сплошной линией, в то время как неидентифицирующая – пунктирной, в ней экземпляр подчиненной сущности не зависит от родительской.

Выводы по главе

В данном разделе была проведена декомпозиция основного процесса веб-приложения, спроектирована серверная часть приложения, а также обоснован выбор технологий разработки и СУБД. Помимо этого, были составлены концептуальная и логическая модели предметной области.

Глава 3. РАЗРАБОТКА СИСТЕМЫ

3.1. Архитектура системы

Так как серверная часть веб-приложения разрабатывалась на Node.js и фреймворке Express, то итоговая разметка содержит в себе следующие значащие каталоги:

- `config;`
- `controllers;`
- `error;`
- `middleware;`
- `models;`
- `routes;`
- `static;`
- `calculations.`

Каждый из вышеперечисленных каталогов несет свою уникальную функцию, описанную далее.

`Config` – каталог с одним файлом – `config.js`, необходимым для выгрузки веб-приложения на сервер (деплой) с описанием пути подключения к базе данных (диалект, протокол, хост, название, имя пользователя и пароль).

`Controllers` – каталог, содержащий js-файлы, отвечающие за CRUD-операции. Ниже приведен список контроллеров и пояснение, с какими сущностями базы данных они проводят вышеописанные операции:

- `UserController` – «Пользователь» и «Данные пользователя для входа»;
- `measureController` – «Измерение»;
- `recipeController` – «Рецепт» и «Шаг рецепта»;
- `targetController` – «Цель»;
- `activityLevelController` – «Уровень активности»;
- `productController` – «Продукт»;

- `rationController` – «Рацион»;
- `mealController` – «Прием пищи»;
- `categoryController` – «Категория продукта».

Листинг контроллера `UserController` приведен в приложении А.

Вышеописанная структура способствует упрощению разработки и поддержке кода путем повышения уровня модульности и декомпозиции.

Каталог `error` содержит в себе файл `ApiError.js`, который отвечает за создание объектов-ошибок для будущего их использования в контроллерах при отправке сообщения об ошибке на клиентскую часть приложения.

Каталог `middleware` содержит в себе файлы промежуточного кода, которые выполняются до того, как начнет выполняться основной код.

`AuthMiddleware` – фильтр, который отвечает за проверку того, авторизован ли пользователь на момент, когда он отправляет запрос на сервер. Если пользователь не авторизован, то фильтр возвращает ошибку и не дает тем самым начаться, например, операции создания рецепта.

`CheckRoleMiddleware` – фильтр, отвечающий за проверку соответствия роли пользователя с необходимой для выполнения кода контроллера, отвечающего, например, за добавление продукта в базу данных. В данном случае вышеописанный код не позволит пользователю выполнить данную функцию, если его роль не содержит прав администратора.

`ErrorHandlingMiddleware` – фильтр, отвечающий за обработку ошибок перед их отправкой на клиентскую часть.

Каталог `models` содержит файл `models.js`, в котором описывается структура создаваемой базы данных, позволяя не создавать таблицы вручную с помощью СУБД, а перечислить названия сущностей и ее полей с помощью библиотеки `Sequelize`. Данная библиотека дает разработчику возможность

взаимодействия с базой данных на языке приложения JavaScript, без использования SQL.

Листинг фильтра AuthMiddleware приведен ниже:

```
const jwt = require('jsonwebtoken')
module.exports = function (req, res, next) {
  if (req.method === 'OPTIONS') {
    next()
  }
  try {
    const token = req.headers.authorization.split(' ')[1]
    if(!token) {
      return res.status(401).json({message: 'Не авторизован'})
    }
    const decoded = jwt.verify(token,
process.env.SECRET_KEY)
    if (decoded.role !== role)
      return res.status(403).json({message: "Нет доступа"})
    req.userCredentials = decoded
    next()
  } catch (e) {
    res.status(401).json({message: 'Не авторизован'})
  }
}
```

Каталог routes содержит в себе файлы, создающие для клиентской части приложения пути, по которым необходимо отправить HTTP-запрос, чтобы получить HTTP-ответ. Данные файлы используют функции, созданные в каталоге controllers и экспортируют готовые пути в основной файл каталога routes – index.js.

Файл calculations отвечает за расчет необходимого КБЖУ пользователя и содержит в себе 5 функций:

- `voz` – расчет калорий по формуле ВОЗ;
- `benedict` – расчет калорий по формуле Харриса-Бенедикта;
- `mifflin` – расчет калорий по формуле Миффлина-Сан Жеора;

- `calculateCalories` – экспортируемая функция, рассчитывает калории с помощью трех вышеперечисленных функций и вычисляет среднее арифметическое значение, а также возвращает 4 полученных результата (три – из формул, четвертое – среднее арифметическое);
- `calculatePFC` – экспортируемая функция, которая рассчитывает необходимое БЖУ на основе среднего арифметического количества калорий и цели пользователя.

Листинг метода `mifflin` приведен ниже:

```
const mifflin = (age, weight, coefficient, height, gender,
target) => {
  if (gender === 'male') {
    return (10 * weight + 6.25 * height - 5 * age + 5) *
coefficient * target
  } else {
    return (10 * weight + 6.25 * height - 5 * age - 161) *
coefficient * target
  }
}
```

Каталог `static` содержит в себе статические файлы, хранимые на серверной части приложения, в нашем случае – изображения.

Файл `index.js` является исполняемым файлом серверной части, который отвечает за запуск сервера и доступ ко всем вышеописанным функциям.

3.2. Обработка изображений

Обработка изображений на серверной части приложения была реализована с помощью библиотек `Uuid` и `Express-fileupload`. Доступ к изображению, отправленному на сервер, предоставляется с помощью `Express-fileupload` из объекта `req.files`, после чего функцией `v4` библиотеки `Uuid` создается уникальное имя, и затем изображение

отправляется в папку `static`. В базе данных хранится не само изображение, а его уникальное сгенерированное имя, которое впоследствии будет отправлено на клиентскую часть для возможности его отображения.

3.3. Авторизация и регистрация

Для повышения уровня безопасности и приватности пользователя, пароль, введенный при регистрации, сначала будет зашифрован и только после этого сохранен в базу данных. Шифрование реализовано с помощью библиотеки `bcrypt`.

Аутентификация разработана с помощью JWT (JSON web token) для обработки системой данных о пользователе в удобном виде. Таким образом, когда действие, совершаемое на клиентской части приложения требует подтверждения аутентификации пользователя, в заголовке HTTP-запроса будет отправлен веб-токен, содержащий данные о пользователе, который будет расшифрован на серверной части приложения, и, в зависимости от того, аутентифицирован ли пользователь, действие либо будет выполнено, либо будет возвращена ошибка.

Пример структуры JWT представлен на рисунке 19.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwicm9sZSI6ImlnVzZlIiLCJ1c2VybmFtZSI6IkpNyZWhbnNzIiwicGFzc3dvcmQiOiJiODllyWFjN2U2MTQxNzU2M0MwI3MTBiNzI3NzY4Mjk0ZDBlNmEyNzdicCJ9.hWYD2Ha1w4vpTz1Lk6aNVU4H_SryPM7huhXRnczWhxE
```

HEADER: ALGORITHM & TOKEN TYPE
<pre>{ "alg": "HS256", "typ": "JWT" }</pre>
PAYLOAD: DATA
<pre>{ "sub": "1234567890", "role": "user", "username": "Crelanss", "password": "b89eaac7e61417341b710b727768294d0e6a277bp" }</pre>
VERIFY SIGNATURE
<pre>HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload), your-256-bit-secret) <input type="checkbox"/> secret base64 encoded</pre>

Рисунок 19. Структура JWT

Листинг метода авторизации пользователя приведен ниже:

```
async login (req, res, next) {
  const {email, password} = req.body
  const user = await User.findOne({where: {email}})
  if (!user) {
    return next(ApiError.internal('Пользователь не найден'))
  }
  let comparePassword = bcrypt.compareSync(password,
user.password)
  if (!comparePassword) {
    return next(ApiError.internal('Указан неверный пароль'))
  }
  const token = generateJwt(user.id, user.email, user.role,
user.name)
  return res.json({token})
}
```

3.4. Пользователи

Для работы с пользователем необходимо реализовать следующие методы:

- зарегистрировать пользователя;
- авторизовать пользователя;
- редактировать данные о пользователе;
- рассчитать КБЖУ пользователя;
- вернуть одного пользователя;
- проверить авторизованность пользователя.

Контроллер `UserController` будет принимать три типа HTTP-запросов: PUT, GET и POST. Ниже описаны действия при обработке запросов с префиксом `user`:

GET-запросы:

- `/auth` – проверка авторизованности пользователя;
- `/` - возвращение пользователя;

POST-запросы:

- `/registration` – зарегистрировать пользователя;
- `/login` – авторизовать пользователя;
- `/cpfc` – КБЖУ пользователя;

PUT-запросы:

- `/` – редактировать данные о пользователе.

3.5. Измерения

Для управления измерениями пользователя необходимо реализовать следующие методы:

- получить данные об измерениях пользователя;
- редактировать данные об измерениях пользователя.

3.6. Рецепты

Для управления рецептами пользователя необходимо реализовать следующие методы:

- создать рецепт;
- получить рецепт по идентификатору;
- редактировать рецепт;
- получить все рецепты;
- удалить рецепт.

Листинг контроллера `recipeController` приведен в приложении Б.

3.7. Цель

Для управления целями необходимо реализовать следующие методы:

- добавить цель;
- изменить цель;
- получить все цели;
- получить цель по идентификатору;
- удалить цель.

3.8. Уровень активности

Для управления уровнями активности необходимо реализовать следующие методы:

- добавить уровень активности;
- редактировать уровень активности;
- получить уровень активности по идентификатору;
- получить все уровни активности;
- удалить уровень активности.

3.9. Продукт

Для управления продуктами активности необходимо реализовать следующие методы:

- добавить продукт;
- редактировать данные о продукте;
- получить продукт по идентификатору;
- получить все продукты;
- удалить продукт.

Листинг роутера `productRouter` представлен ниже:

```
const Router = require('express')
const router = new Router()
const productController =
require('../controllers/productController')
const checkRole = require('../middleware/checkRoleMiddleware')
router.post('/', checkRole('administrator'),
productController.create)
router.get('/', productController.getAll)
router.get('/:id', productController.getOne)
router.delete('/:id', checkRole('administrator'),
productController.delete)
router.put('/:id', checkRole('administrator'),
productController.updateProductInfo)
module.exports = router
```

3.10. Рацион

Для управления рационом пользователя необходимо реализовать следующие методы:

- создать рацион;
- редактировать рацион;
- получить рацион по идентификатору;
- получить все рационы пользователя;
- удалить рацион;

3.11. Прием пищи

Для управления приемами пищи необходимо реализовать следующие методы:

- создать прием пищи;
- редактировать прием пищи;
- получить прием пищи по идентификатору;
- получить все приемы пищи;
- удалить прием пищи.

3.12. Категория продукта

Для управления категориями продуктов необходимо реализовать следующие методы:

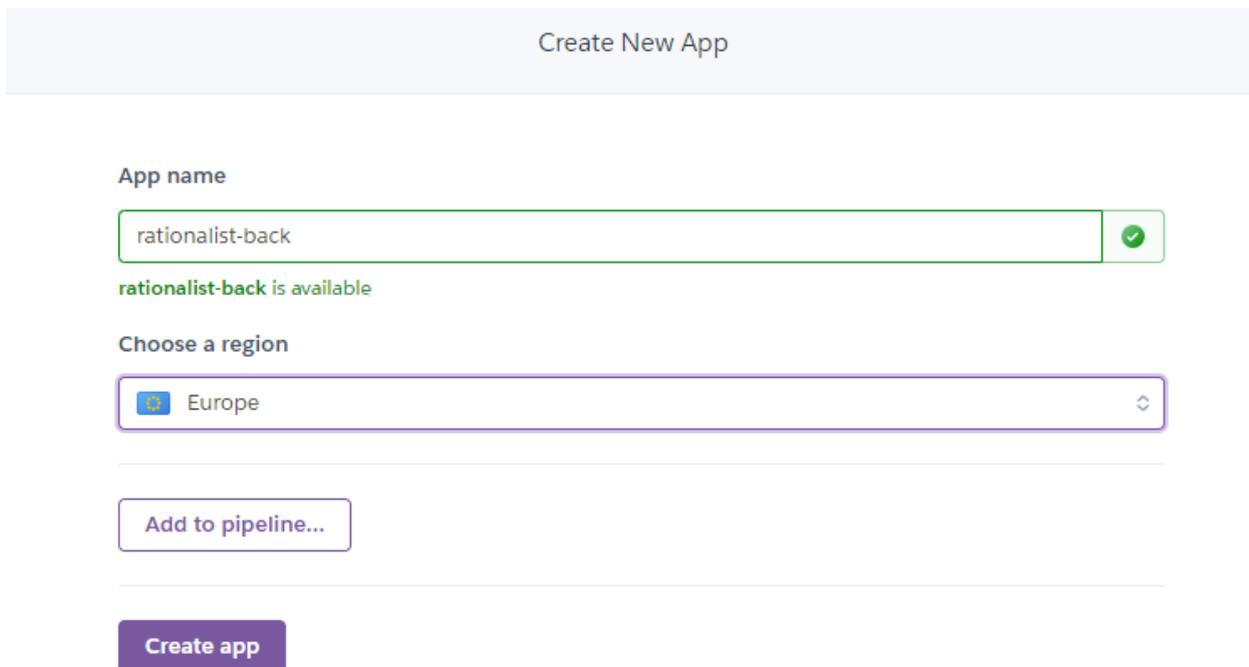
- создать категорию продукта;
- редактировать категорию продукта;
- получить категорию продукта по идентификатору;
- получить все категории продуктов;
- удалить категорию продукта.

3.13. Деплой приложения

В качестве платформы для размещения серверной части приложения была выбрана облачная PaaS (Platform as a Service) платформа Heroku,

которая позволяет в автоматизированном режиме загружать программные решения, в будущем развертываемые по сгенерированному URL.

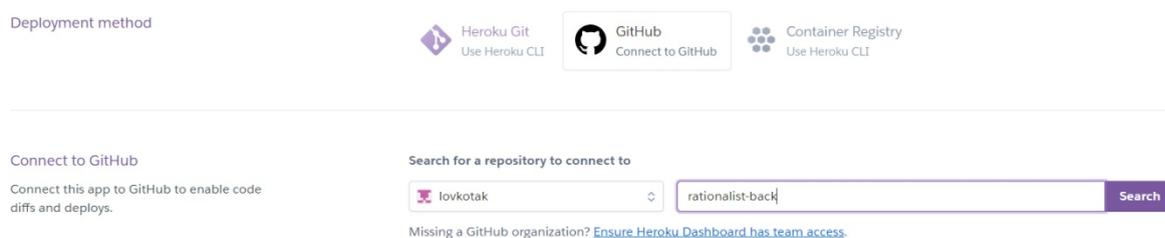
Для того, чтобы задеплоить серверную часть, необходимо создать приложение в Heroku (рисунок 20).



The screenshot shows the 'Create New App' interface on Heroku. At the top, there is a header 'Create New App'. Below it, the 'App name' section has a text input field containing 'rationalist-back' and a green checkmark icon to its right. Below the input field, it says 'rationalist-back is available'. The 'Choose a region' section has a dropdown menu with 'Europe' selected. Below these sections, there is a button labeled 'Add to pipeline...' and a larger purple button labeled 'Create app'.

Рисунок 20. Создание приложения в Heroku

После создания приложения, необходимо связать Git-репозиторий и приложение, авторизовавшись через GitHub, после чего указать имя репозитория. Данный процесс продемонстрирован на рисунке 21.



The screenshot shows the 'Deployment method' section of the Heroku app creation process. It lists three options: 'Heroku Git Use Heroku CLI', 'GitHub Connect to GitHub', and 'Container Registry Use Heroku CLI'. The 'Connect to GitHub' section is expanded, showing a search for a repository to connect to. The search field contains 'rationalist-back' and a 'Search' button. Below the search field, there is a link: 'Missing a GitHub organization? Ensure Heroku Dashboard has team access.'

Рисунок 21. Подключение Git-репозитория

Следующим шагом необходимо создать базу данных, установив расширение для Heroku под названием «Heroku Postgres» и связать ее с существующим приложением (рисунок 22).

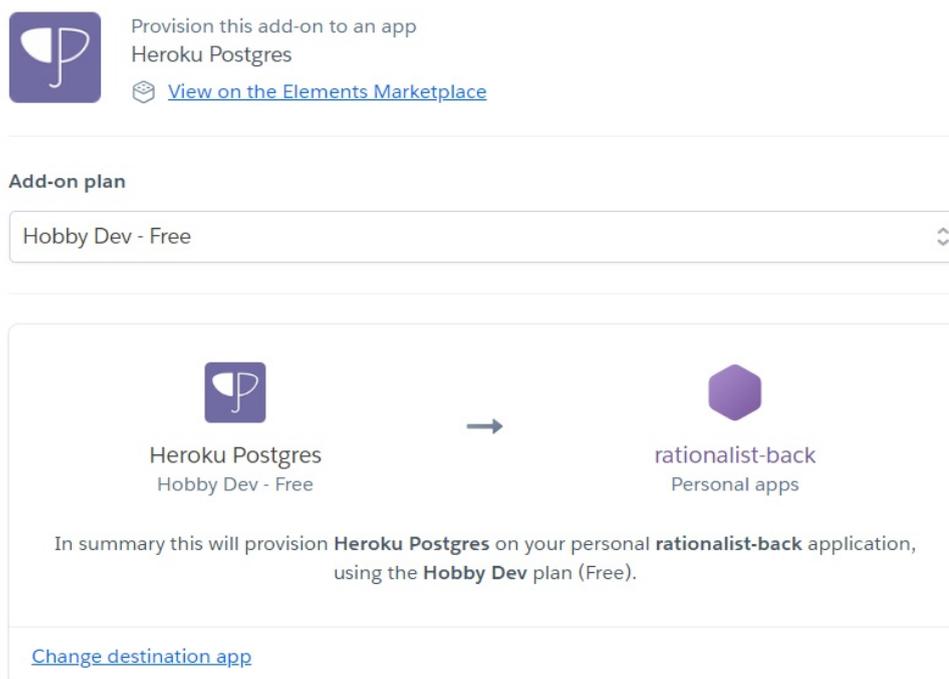


Рисунок 22. Подключение базы данных

Когда база данных подключена и приложение создано, необходимо лишь проверить корректность настроенного файла `config.js`, с помощью которого библиотека `Sequelize` будет связывать приложение и облачную базу данных. Выполнив этот шаг, на серверную часть приложения получится отправлять запросы не только с локальной сети, а с любого устройства, введя сгенерированный URL для приложения.

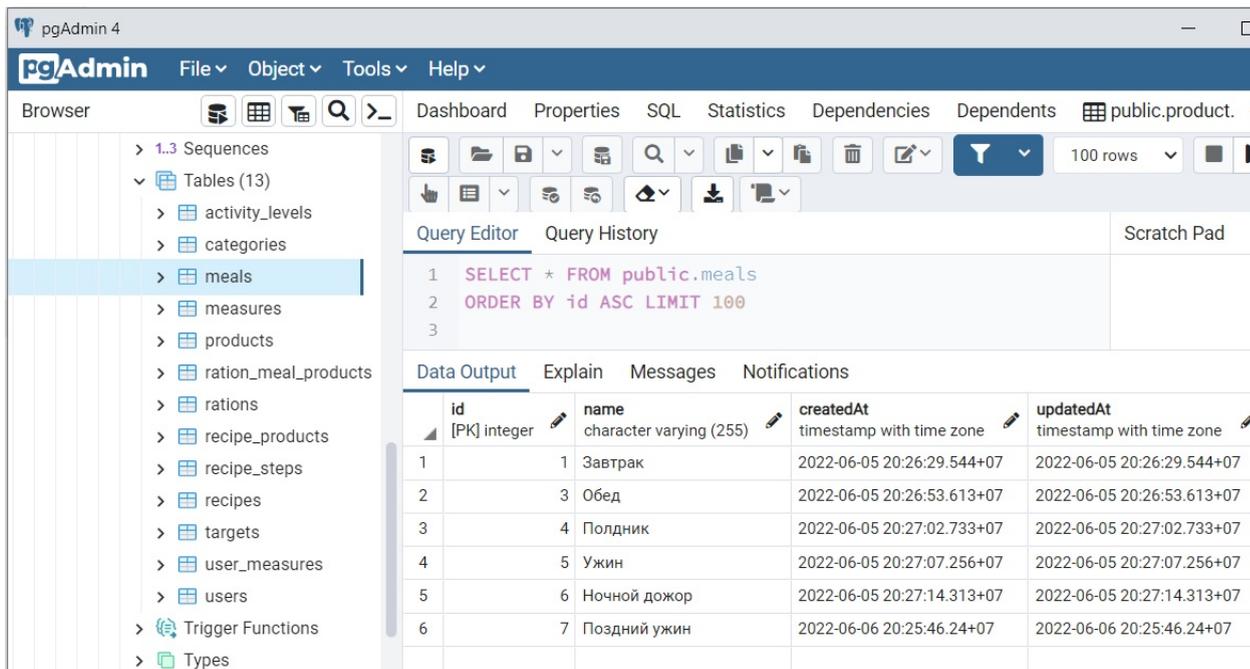
Выводы по главе

В данной главе были описаны детали разработки веб-приложения. Для серверной части были описаны файловая структура и архитектура, а также более подробно рассмотрены обработка изображений, авторизация, регистрация, CRUD-операции с сущностями базы данных, деплой приложения.

Глава 4. АНАЛИЗ РЕЗУЛЬТАТОВ РАЗРАБОТКИ И ТЕСТИРОВАНИЕ

4.1. Результат создания базы данных

В процессе разработки серверной части веб-приложения для составления пищевого-рациона была создана база данных с помощью СУБД PostgreSQL и библиотеки Sequelize. Созданная база данных была заполнена тестовыми данными. Результат представлен на рисунке 23.



The screenshot shows the pgAdmin 4 interface. The left sidebar shows a tree view of the database structure, with 'meals' selected under 'Tables (13)'. The main window displays the 'Query Editor' with the following SQL query:

```
1 SELECT * FROM public.meals
2 ORDER BY id ASC LIMIT 100
3
```

Below the query editor, the 'Data Output' tab shows a table with the following data:

id	name	createdAt	updatedAt
1	Завтрак	2022-06-05 20:26:29.544+07	2022-06-05 20:26:29.544+07
2	Обед	2022-06-05 20:26:53.613+07	2022-06-05 20:26:53.613+07
3	Полдник	2022-06-05 20:27:02.733+07	2022-06-05 20:27:02.733+07
4	Ужин	2022-06-05 20:27:07.256+07	2022-06-05 20:27:07.256+07
5	Ночной дождор	2022-06-05 20:27:14.313+07	2022-06-05 20:27:14.313+07
6	Поздний ужин	2022-06-06 20:25:46.24+07	2022-06-06 20:25:46.24+07

Рисунок 23. Сущность «Приемы пищи» с тестовыми данными

4.2. Тестирование API сервера

Ручное тестирование серверной части производилось с помощью платформы Postman, позволяющей отправлять HTTP-запросы и получать HTTP-ответы. На каждый контроллер была создана коллекция запросов, в каждой из которых были протестированы методы, описанные в контроллере.

4.2.1. Авторизация и регистрация

На рисунке 24 представлен процесс регистрации пользователя.

Также была проведена проверка случаев ввода некорректных данных при попытке авторизации, результат представлен на рисунке 26.

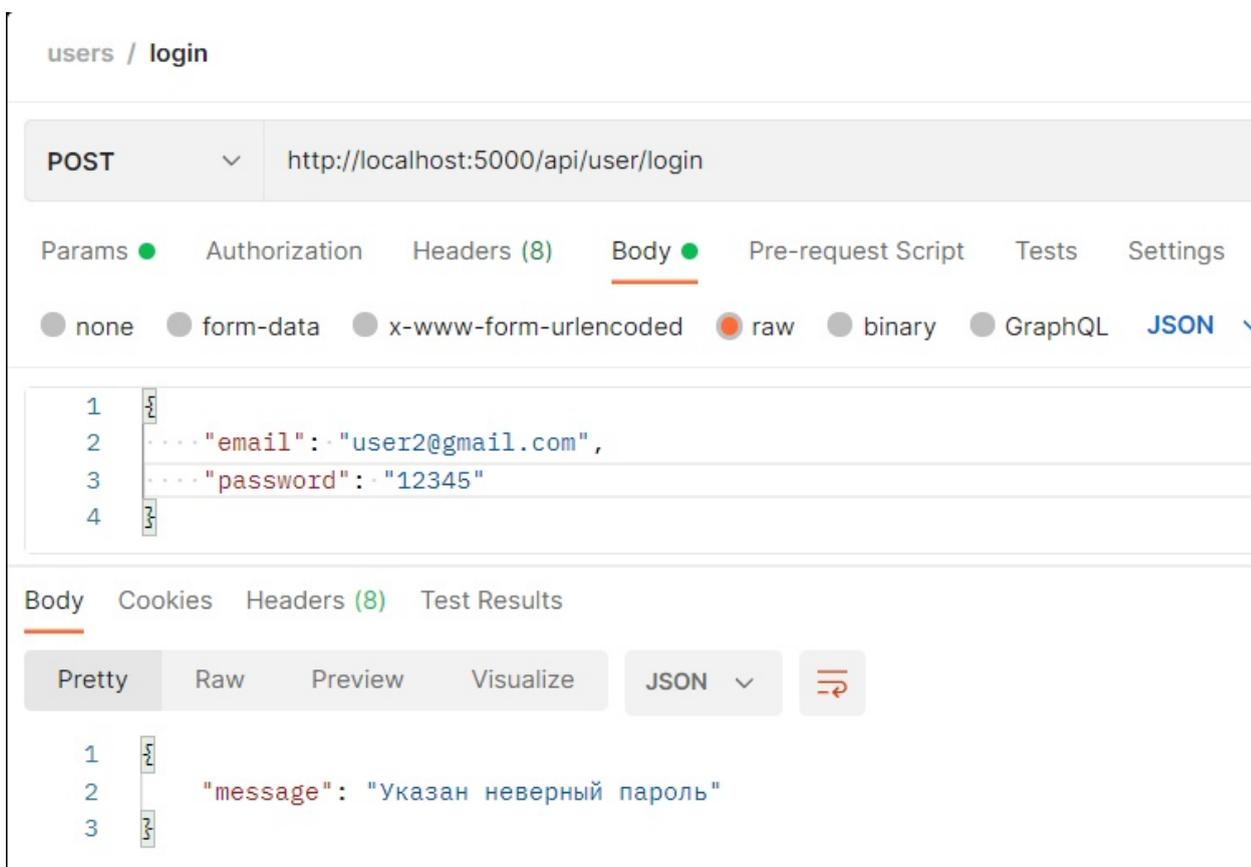


Рисунок 26. Попытка авторизации пользователя при введении некорректных данных

4.2.2. Получение данных о пользователе

Для отображения информации в личном кабинете пользователя, необходимо получить с серверной части приложения все данные о нем с помощью GET-запроса. Так как в GET-запросах нет тела запроса, данные, по которым необходимо получить информацию о пользователе из базы данных, передавалась через JWT, который содержался в заголовке запроса, он расшифровывался на серверной части и по ID пользователя, полученного из расшифрованного вида токена, производился поиск в БД. Пример получения данных о пользователе представлен на рисунке 27.

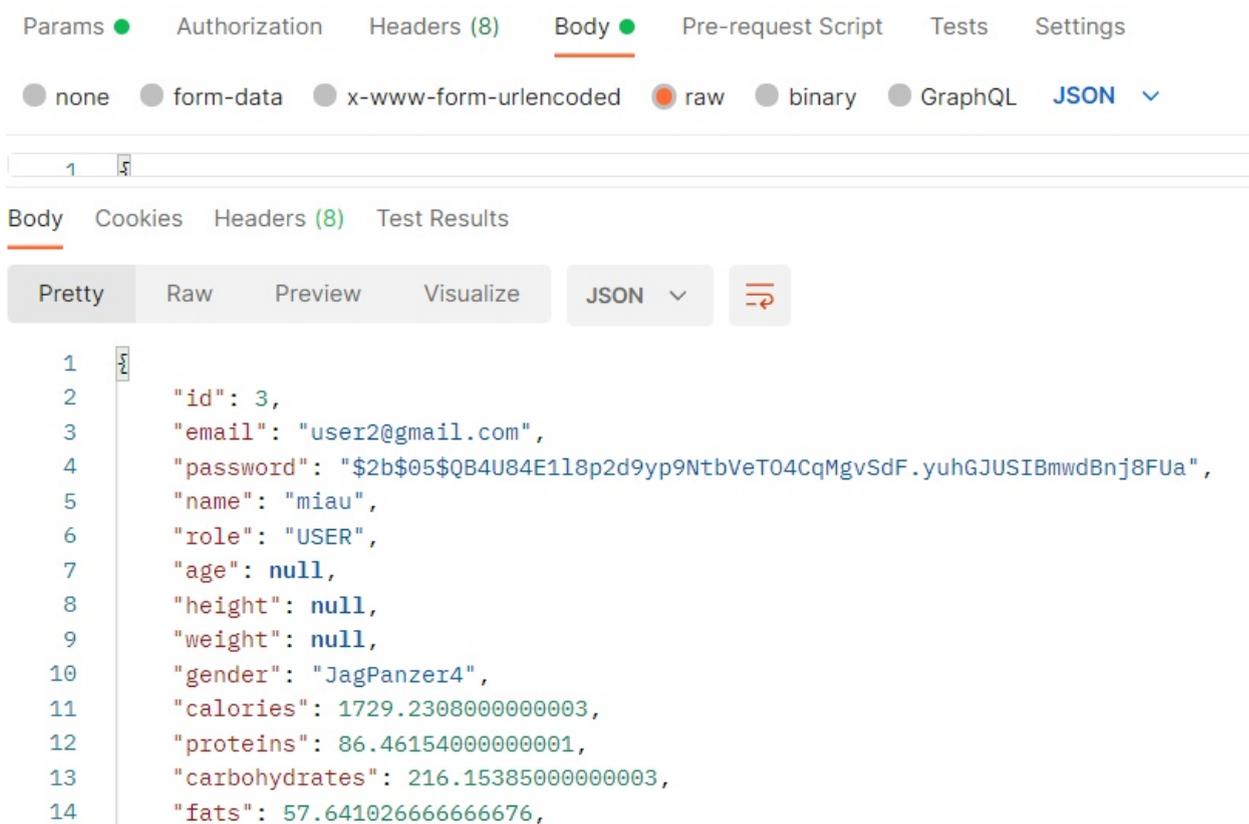


Рисунок 27. Получение данных о пользователе

4.2.3. Создание приема пищи

Создание цели пользователя, уровня активности пользователя, категории продукта, и приема пищи, происходит по одному сценарию, представленном на рисунке 28.

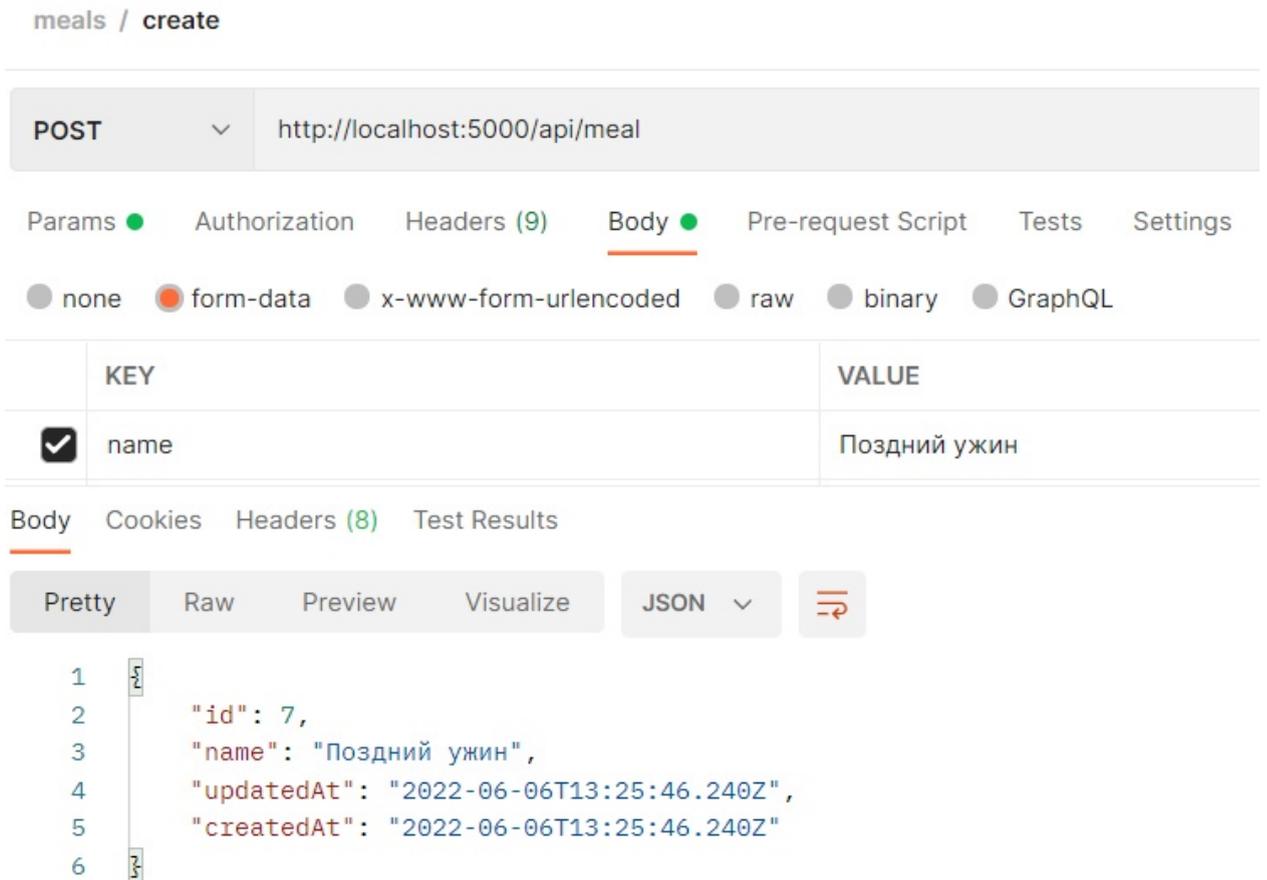


Рисунок 28. Создание приема пищи

4.2.4. Создание рациона

При создании рациона, на серверную часть отправляются данные о содержании КБЖУ в рационе, список продуктов, приемы пищи, и название рациона. Процесс создания рациона продемонстрирован на рисунке 29.

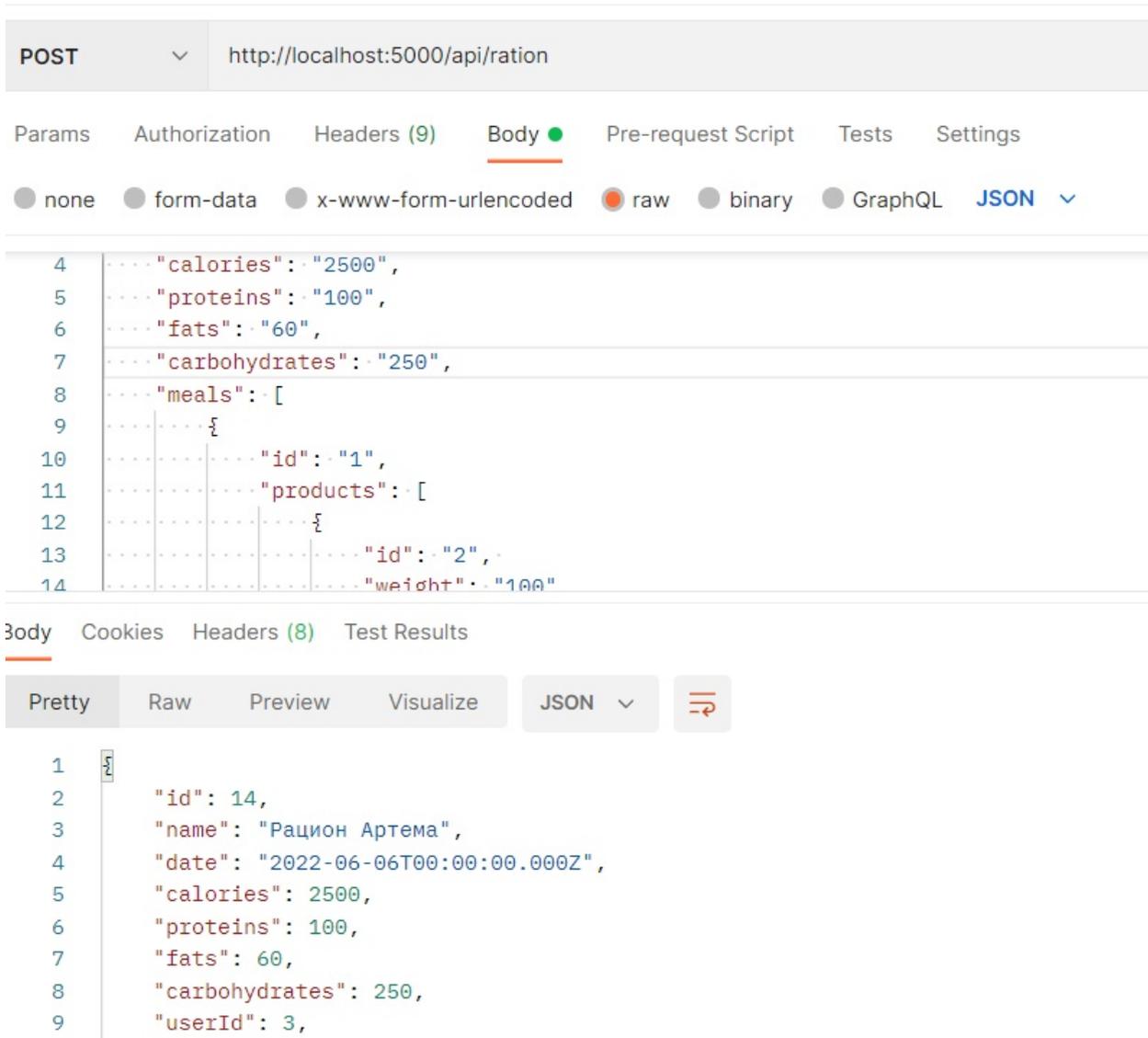


Рисунок 29. Создание рациона

4.2.5. Создание рецепта

При создании рецепта, с клиентской части отправляются данные о КБЖУ рецепта, список продуктов, изображение рецепта, его название и описание. Процесс создания рецепта продемонстрирован на рисунке 30.

The screenshot shows a REST client interface for a POST request to `http://localhost:5000/api/ration`. The request body is a JSON object with the following structure:

```

4     .... "calories": "2500",
5     .... "proteins": "100",
6     .... "fats": "60",
7     .... "carbohydrates": "250",
8     .... "meals": [
9     ..... {
10     .....   "id": "1",
11     .....   "products": [
12     .....     {
13     .....       "id": "2",
14     .....       "weight": "100"

```

Below the request body, the response is shown in the 'Test Results' tab, rendered in a 'Pretty' JSON format:

```

1  {
2    "id": 14,
3    "name": "Рацион Артема",
4    "date": "2022-06-06T00:00:00.000Z",
5    "calories": 2500,
6    "proteins": 100,
7    "fats": 60,
8    "carbohydrates": 250,
9    "userId": 3,

```

Рисунок 30. Создание рецепта

4.2.6. Удаление из базы данных

Все сущности БД поддерживают функцию удаления, доступ к которой предоставляется с помощью DELETE-запроса. Процесс удаления приема пищи представлен на рисунке 31.

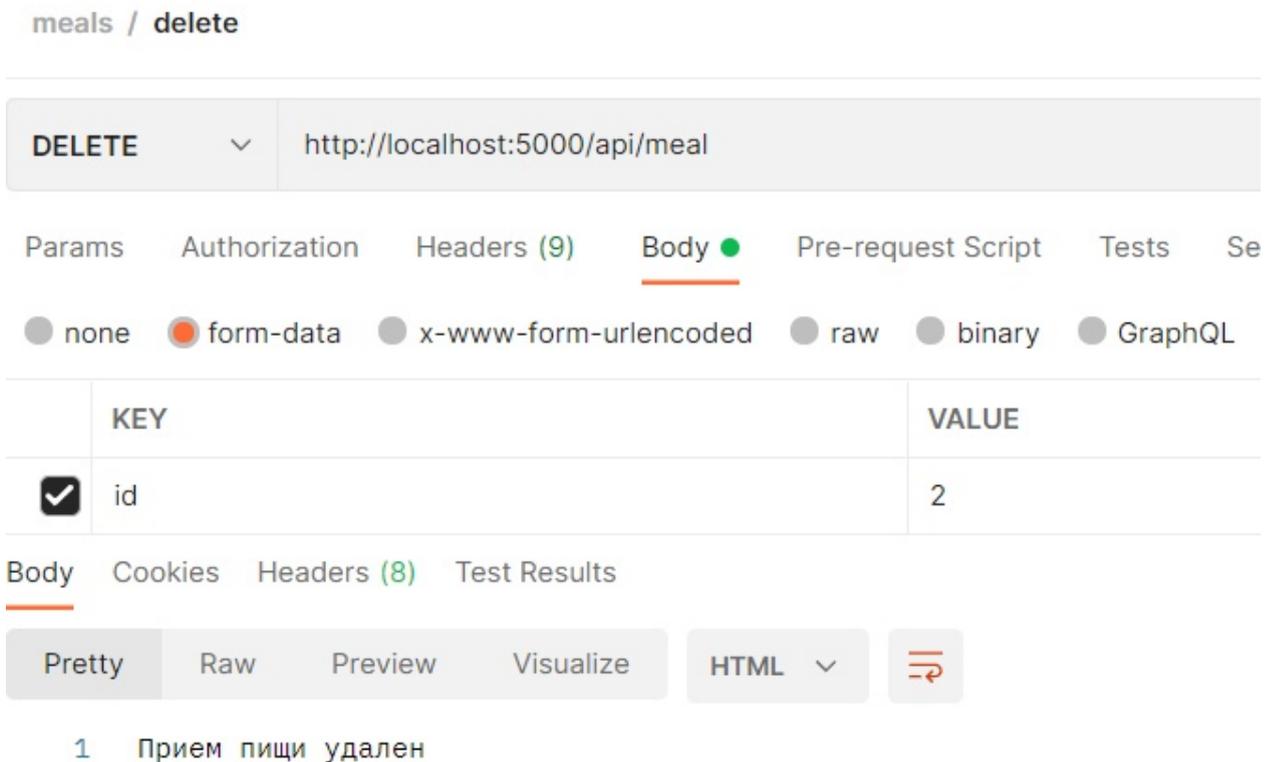


Рисунок 31. Удаление приема пищи

Выводы по главе

В ходе работы по созданию серверной части веб-приложения для составления рациона была разработана архитектура на основании рекомендаций и стандартов технологий Node.js. Реализованы конечные точки запросов для регистрации, авторизации, создания рациона, ведения и отслеживания измерений, управления продуктами, категориями продуктов, уровнями активности, целями для администратора, а также создания и просмотра рецептов. Была проведена демонстрация функционала бэкэнд-части приложения как со стороны администратора (создание категории продуктов), так и со стороны авторизованного пользователя (составление рациона).

ГЛАВА 5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Целью выпускной квалификационной работы является проектирование и разработка бэкэнд-части приложения для составления пищевого рациона.

В текущем разделе необходимо определить продолжительность работ и произвести расчет трудовых затрат проекта. Также необходимо эффективно организовать производство для уменьшения экономических затрат. Для эффективной организации производства необходимо экономически обосновать все инженерные решения.

Для достижения поставленной цели будут выполнены следующие задачи:

- оценка конкурентоспособности технических решений;
- проведение SWOT-анализа для выявления сильных и слабых сторон проекта;
- планирование проведения работ с построением диаграммы Ганта;
- расчет бюджета проекта;
- оценка ресурсной, финансовой и экономической эффективности.

5.1. Оценка перспективности проведения исследований

5.1.1. Анализ конкурентных технических решений

Детальный анализ конкурирующих разработок, существующих на рынке, необходимо проводить систематически, поскольку рынки пребывают в постоянном движении. Такой анализ помогает вносить коррективы в научное исследование, чтобы успешнее противостоять своим соперникам. Важно реалистично оценить сильные и слабые стороны разработок конкурентов.

В качестве конкурирующих разработок были выбраны: ЗОЖНИК – дневник зожника, на карте обозначен как БК1, 3X15 – обозначен на карте как БК2. Данные приложения не находятся в одинаковом ценовом диапазоне, так

как первое из перечисленных приложений распространяется по платной модели подписки, а второе – полностью бесплатное. На данный момент они являются единственными конкурентами на рынке в указанной предметной области. Собственная разработка указана на карте как Б. Оценочная карта представлена в таблице 3.

Позиция выбранной разработки и альтернативных вариантов оценивается по каждому показателю экспертным путем по пятибалльной шкале, где 1 – наиболее слабая позиция, а 5 – наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1 (100%).

Таблица 3 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес	Баллы			Конкурентоспособность		
		Б	Б _{К1}	Б _{К2}	К	К _{К1}	К _{К2}
Технические критерии оценки ресурсоэффективности							
Стабильность работы	0,15	4	4	3	0,6	0,6	0,45
Производительность клиентского интерфейса	0,15	5	2	5	0,75	0,3	0,75
Графический дизайн	0,2	4	5	2	0,8	1	0,4
Удобство для пользователей	0,2	5	3	4	1	0,6	0,8
Функциональность	0,15	5	5	2	0,75	0,75	0,3
Экономические критерии оценки эффективности							
Стоимость возможности использования полного функционала	0,15	5	2	5	0,75	0,3	0,75
Итого	1	28	21	21	4,65	3,55	3,45

Пример оценки конкурентоспособности приведем для нашего приложения:

$$K = \sum(\text{Вес} \cdot \text{Б})$$

$$K = 0,15 \cdot 4 + 0,15 \cdot 5 + 0,2 \cdot 4 + 0,2 \cdot 5 + 0,15 \cdot 5 + 0,15 \cdot 5 = 4,65;$$

где K_{K1} , K_{K2} – конкурентоспособности приложений ЗОЖНИК и 3X15, соответственно; Вес – вес критерия (в долях единицы, в сумме равняется 1), выбирается экспертным путем;

Согласно оценочной карте наиболее конкурентоспособным, в данной ситуации, является мобильное приложение ЗОЖНИК с показателем конкурентоспособности 3,55 условных единиц, однако уступает показателю конкурентоспособности нашей разработки – 4,65. Основными конкурентными преимуществами нашей разработки являются: ценовая политика, удобство и производительность графического интерфейса.

5.2. SWOT-анализ

SWOT-анализ применяют для исследования внешней и внутренней среды проекта. Он проводится в несколько этапов. Первый этап помогает выявить сильные и слабые стороны проекта, также возможности и угрозы.

Сильные стороны — это факторы, которые положительно сказываются на развитии проекта. Сюда обычно включают все, что превращает функционирование в успешную и конкурентную работу.

Слабые стороны– это недостаток, упущение или ограниченность научно-исследовательского проекта, которые препятствуют достижению его целей. Это то, что плохо получается в рамках проекта или где он располагает недостаточными возможностями или ресурсами по сравнению с конкурентами.

Возможности включают в себя любую предпочтительную ситуацию в настоящем или будущем, возникающую в условиях окружающей среды проекта: тенденцию, изменение или предполагаемую потребность, которая поддерживает спрос на результаты проекта и позволяет руководству проекта улучшить свою конкурентную позицию.

Угроза представляет собой любую нежелательную ситуацию, тенденцию или изменение в условиях окружающей среды проекта, которые

имеют разрушительный или угрожающий характер для его конкурентоспособности в настоящем или будущем. В качестве угрозы может выступать барьер, ограничение или что-либо еще, что может повлечь за собой проблемы, разрушения, вред или ущерб, наносимый проекту.

На первом этапе SWOT анализа в таблице 4 были описаны сильные и слабые стороны проекта, выявлены возможности и угрозы реализации веб-приложения.

Таблица 4 – Матрица SWOT анализа

Сильные стороны	Возможности во внешней среде
С1. Бесплатное использование; С2. Эргономичный дизайн; С3. Простота применения для конечного пользователя.	В1. Попадание продукта на первые строчки поисковых запросов браузера; В2. Тренд на здоровое питание;
Слабые стороны	Угрозы внешней среды
Сл1. Использование серверов начального уровня производительности; Сл2. Медленный старт из-за ограниченности рекламного бюджета; Сл3. Отсутствие мобильной версии.	У1. Усовершенствование конкурентных решений; У2. Отсутствие спроса на продукт.

Второй этап состоит в выявлении соответствия сильных и слабых сторон проекта внешним условиям окружающей среды. Это соответствие или несоответствие должны помочь выявить степень необходимости проведения стратегических изменений. В рамках данного этапа необходимо построить интерактивную матрицу проекта. Ее использование помогает разобраться с различными комбинациями взаимосвязей областей матрицы SWOT. Возможно использование этой матрицы в качестве одной из основ для оценки вариантов стратегического выбора. Каждый фактор помечается либо знаком «+» (означает сильное соответствие сильных сторон возможностям), либо

знаком «-» (что означает слабое соответствие); «0» – если есть сомнения в том, что поставить «+» или «-». Интерактивная матрица проекта представлена в табл. 5 и 6.

Таблица 5 – Интерактивная матрица сильных и слабых сторон и возможностей

Возможности проекта	Сильные стороны			Слабые стороны		
		C1	C2	C3	Сл1	Сл2
B1	+	+	+	+	-	0
B2	+	-	-	-	-	+

Таблица 6 – Интерактивная матрица сильных сторон и слабых сторон и угроз

Угрозы проекта	Сильные стороны			Слабые стороны		
		C1	C2	C3	Сл1	Сл2
У1	+	+	+	0	0	+
У2	+	+	+	-	+	+

Анализ интерактивных таблиц представляется в форме записи сильно коррелирующих сильных сторон и возможностей или слабых сторон и возможностей:

- B1B2C1; B1C2C3;

- B1Сл1; B2Сл3;

- У1У2С1С2С3;

- У2Сл2; У1У2Сл3.

Самой большой угрозой для проекта является малый спрос из-за отсутствия мобильной версии, что может быть нивелировано созданием адаптивной версии веб-приложения для работы на всех типах устройств.

В рамках третьего этапа составляется итоговая матрица SWOT-анализа, представленная в таблице 7.

Таблица 7 – Итоговая матрица SWOT-анализа

	<p>Сильные стороны:</p> <p>С1. Бесплатное использование;</p> <p>С2. Эргономичный дизайн;</p> <p>С3. Простота применения для конечного пользователя.</p>	<p>Слабые стороны:</p> <p>Сл1. Использование серверов начального уровня производительности;</p> <p>Сл2. Медленный старт из-за ограниченности рекламного бюджета;</p> <p>Сл3. Отсутствие мобильной версии.</p>
<p>Возможности:</p> <p>В1. Попадание продукта на первые строчки поисковых запросов браузера;</p> <p>В2. Тренд на здоровое питание;</p>	<p>Веб-приложение может попасть на первые строчки поисковых запросов в браузере за счет бесплатного доступа и простоты применения.</p>	<p>Исходя из слабых сторон и возможностей, можно предположить экспоненциальный рост числа пользователей, наблюдается сильная зависимость развития от первоначальной активности, проект может завязнуть на начальном этапе.</p>
<p>Угрозы:</p> <p>У1. Усовершенствование конкурентных решений;</p> <p>У2. Отсутствие спроса на продукт.</p>	<p>Инновационные технические решения конкурентов могут понизить спрос на использование нашего приложения, несмотря на бесплатность сервиса.</p>	<p>Отсутствие мобильной версии может заставить пользователей обратить внимание на продукты, у которых имеются и мобильная, и веб-версия.</p>

5.3. Планирование работ по научно-техническому исследованию

5.3.1. Структура работ в рамках научного исследования

Проектирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках проектирования серверной части веб-приложения;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения технического проекта.

Порядок составления этапов и работ от разработки технического задания и до оформления итогового отчета, распределение исполнителей по данным видам работ приведен в таблице 8.

Таблица 8 – Перечень этапов работ при проектировании

Основные этапы	№ работы	Содержание работы	Исполнитель
Разработка темы и задания	1	Составление календарного плана графика выполнения бакалаврской работы	инженер; науч. рук.
	2	Разработка ТЗ	инженер; науч. рук.
Выбор направления исследования	3	Подбор и изучение литературы по теме бакалаврской работы	инженер; науч. рук.
	4	Анализ предметной области	инженер; науч. рук.
Теоретические исследования	5	Выбор программных решений для разработки серверной части приложения	инженер
	6	Исследование архитектурных подходов	инженер
Процесс разработки	7	Разработка концептуального дизайна приложения	инженер
	8	Разработка серверной части	инженер
	9	Тестирование	инженер
	10	Доработка, устранение выявленных ошибок	инженер
Заключительный этап	11	Составление отчета о проделанной работе и оценка эффективности полученных результатов	инженер; науч. рук.
	12	Защита дипломного проекта	инженер

5.3.2. Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ.

Трудоемкость выполнения проекта оценивается экспертным путем в человеко-днях и носит вероятностный характер, который зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости $t_{ожі}$ используется следующая формула:

$$t_{\text{ож}i} = \frac{3t_i + 2t_i}{5}, \quad (5.1)$$

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

t_i – минимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.;

t_i – максимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.;

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях T_p , учитывающая параллельность выполнения работ по нескольким исполнителями.

$$T_{pi} = \frac{t_{\text{ож}i}}{Ч_i}, \quad (5.2)$$

где T_{pi} – продолжительность одной работы, раб.дн.;

$t_{\text{ож}i}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.;

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

5.3.3. Разработка графика проведения научного исследования

Наиболее удобным и наглядным представлением проведения научных работ является построение ленточного графика в форме диаграммы Ганта.

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

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

$$T_{ki} = T_{pi} \cdot k_{\text{кал}}, \quad (5.3)$$

где T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{pi} – продолжительность выполнения i -й работы в рабочих днях;

$k_{\text{кал}}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - (T_{\text{вых}} + T_{\text{пр}})}, \quad (5.4)$$

где $T_{\text{кал}}$ – количество календарных дней в году;

$T_{\text{вых}}$ – количество выходных дней в году;

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

Расчет коэффициента календарности:

Таблица 9 – Временные показатели научного исследования

Номер работы	Трудоёмкость работ, чел.-дн.			Длительность работ, дн.			
	t _{min}	t _{max}	t _{ож}	Тр		Тк	
				науч. рук.	инженер	науч. рук.	инженер
1	1	2	1,4	0,7	0,7	1,036	1,036
2	3	7	4,6	2,3	2,3	3,404	3,404
3	5	10	7	3,5	3,5	5,18	5,18
4	3	7	4,6	2,3	2,3	3,404	3,404
5	7	12	9	0	9	0	13,32
6	2	4	2,8	0	2,8	0	4,144
7	7	14	9,8	0	9,8	0	14,504
8	20	25	22	0	22	0	32,56
9	1	2	1,4	0	1,4	0	2,072
10	1	3	1,8	0	1,8	0	2,664
11	7	10	8,2	0	8,2	0	12,136
12	1	1	1	0,5	0,5	0,74	0,74
Итого	58	97	73,6	9,3	64,3	13,764	95,164

На основе полученной таблицы построим календарный план-график (таблица 10).

Таблица 10. Календарный план-график проведения работ

Номер работы	Исполнитель		Продолжительность выполнения работ												
			февраль		март			апрель			май				
			2	3	1	2	3	1	2	3	1	2	3		
1	инженер; науч. рук.	1,036	син	оранж											
2	инженер; науч. рук.	3,404	син	оранж											
3	инженер; науч. рук.	5,18	син	оранж											
4	инженер; науч. рук.	3,404		син	оранж										
5	инженер	13,32		син	син										
6	инженер	4,114				син									
7	инженер	14,504				син	син	син							
8	инженер	32,56							син	син	син	син			
9	инженер	2,072											син		
10	инженер	2,664											син		
11	инженер; науч. рук.	12,136											син	син	оранж
12	инженер	0,74													син

син. – инженер,

оранж. – науч. рук.

5.4. Бюджет технического проекта

При планировании бюджета ТП должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета ТП используется следующая группировка затрат по статьям:

- материальные затраты ТП;
- затраты на оборудование;
- амортизационные отчисления;

- заработная плата исполнителей темы;
- отчисления во внебюджетные фонды;
- накладные расходы.

5.4.1. Расчет материальных затрат

При планировании бюджета научно-техническое исследование должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением.

Таблица 11 – Материальные затраты

Наименование	Цена за ед., руб.	Кол-во, шт.	Сумма, руб.
Кронштейн для монитора	1500	2	3000
Хостинг для сервера	1200	1	1200
Итого			4200

5.4.2. Расчет затрат на оборудование

В данную статью включают все затраты, связанные с приобретением специального оборудования (приборов, контрольно-измерительной аппаратуры, стендов, устройств и механизмов), необходимого для проведения работ. Определение стоимости спецоборудования производится по действующим прейскурантам.

При выполнении научно-исследовательской работы использовался персональный компьютер. Его стоимость представлена в таблице 12.

Таблица 12 – Расчет бюджета затрат на приобретение оборудования

Наименование оборудования	Количество, шт.	Цена за 1 ед., руб.	$Z_{\text{обор}}$, руб.
Персональный компьютер	1	120 000	120 000
Итого:			120 000 рублей

5.4.3. Основная заработная плата исполнителей

Статья включает расчет основной заработной платы работников, непосредственно занятых выполнением проекта, то есть инженера и научного руководителя.

Основная заработная плата рассчитывается по следующей формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} \cdot T_p, \quad (5.5)$$

где $Z_{\text{осн}}$ – основная заработная плата одного работника, руб.; $Z_{\text{дн}}$ – среднедневная заработная плата работника, руб.; T_p – продолжительность работ, выполняемых научно-техническим работником, раб. дн.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_m \cdot M}{F_d}, \quad (5.6)$$

где Z_m – месячный должностной оклад работника, руб.; M – количество месяцев работы без отпуска в течение года (10,4 для 6-дневной недели); F_d – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн.

Таблица 13 – Баланс рабочего времени (для 6-дневной недели)

Показатели рабочего времени	Дни
Календарные дни	365
Нерабочие дни (праздники/выходные)	118
Потери рабочего времени (отпуск/невыходы по болезни)	59
Действительный годовой фонд рабочего времени	188

Месячный должностной оклад работника:

$$Z_m = Z_{\text{тс}} \cdot (1 + k_{\text{пр}} + k_d) \cdot k_p, \quad (5.7)$$

где $Z_{\text{тс}}$ – заработная плата по тарифной ставке, руб.; $k_{\text{пр}}$ – премиальный коэффициент (0,3); $k_{\text{д}}$ – коэффициент доплат и надбавок (0,2-0,5); $k_{\text{р}}$ – районный коэффициент (для Томска – 1,3).

Таким образом, расчёт основной заработной платы представлен в таблице 14.

Таблица 14 – Расчёт основной заработной платы

Исполнители	$Z_{\text{тс}}$, руб.				$Z_{\text{м}}$, руб.	М		$Z_{\text{дн}}$, руб.		$Z_{\text{осн}}$, руб.
Инженер	20000	0,3	0,2	1,3	39000	10,4	188	2157	72,3	155 951
Научный руководитель	47317	0,3	0,2	1,3	92268	10,4	188	5104	9,3	47 467
Итого: 203 418 рублей										

5.4.4. Расчет дополнительной заработной платы

Затраты по дополнительной заработной платы исполнителей темы учитывают величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций.

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{\text{доп}} = k_{\text{доп}} \cdot Z_{\text{осн}}, \quad (5.8)$$

где $k_{\text{доп}}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12–0,15). Расчет дополнительной заработной платы приведен в таблице 15.

Таблица 15 – Расчет дополнительной заработной платы

Исполнители	$Z_{\text{осн}}$, руб.	$k_{\text{доп}}$	$Z_{\text{доп}}$, руб.
Инженер	155 951	0,12	18 714
Руководитель	47 467	0,12	5 696
Итого: 24 410 рублей			

5.4.5 Отчисления во внебюджетные фонды

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

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$Z_{\text{внеб}} = K_{\text{внеб}} \cdot (Z_{\text{осн}} + Z_{\text{доп}}), \quad (5.9)$$

где $K_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды, равный 0,3. Представим подробный расчет отчислений во внебюджетные фонды для инженера:

$$Z_{\text{внеб}} = 0,3 * (155 951 + 18 714) = 52 400 \text{ рублей}$$

Представим отчисления во внебюджетные фонды в табличной форме (таблица 16).

Таблица 16 – Отчисления во внебюджетные фонды

Исполнители	$Z_{\text{осн}}$, руб.	$Z_{\text{доп}}$, руб.	$k_{\text{в}}$	$Z_{\text{внеб}}$, руб.
Инженер	155 951	18 714	0,3	52 400
Руководитель	47 467	5 696	0,3	15 949
Итого: 68 348 рублей				

5.4.6. Накладные расходы

Накладные расходы учитывают прочие затраты организации, не включенные в предыдущие статьи расходов: печать и ксерокопирование

материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы и т.д. Величина коэффициента накладных расходов принимается в размере 16%. Их величина определяется по следующей формуле:

$$Z_{\text{накл}} = \left(\sum \text{статей} \right) \cdot k_{\text{нр}}, \quad (5.10)$$

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

Величину коэффициента накладных расходов можно взять в размере 16%.

В предыдущие статьи расходов не включены: оплата интернета ($Z_{\text{инт}} = 3000$ руб), оплата ПО для разработки ($Z_{\text{по}} = 10\,000$ руб).

Накладные расходы составили:

$$Z_{\text{накл}} = (4200 + 120000 + 203418 + 24410 + 68348 + 3000 + 10000) \cdot 0,16 = 69340$$

Запишем результат в таблице 17.

Таблица 17 – Накладные расходы

$Z_{\text{мат}}$, руб.	$Z_{\text{обор}}$, руб.	$Z_{\text{осн}}$, руб.	$Z_{\text{доп}}$, руб.	$Z_{\text{инт}}$, руб.	$Z_{\text{по}}$, руб.	$Z_{\text{внеб}}$, руб.
4200	120 000	203 418	24 410	3 000	10 000	68 348
Итого: 69 340 рублей						

5.4.7. Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат проектной работы является основной для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается проектной организацией в

качестве нижнего предела затрат на разработку. Данные бюджета затрат приведены в таблице 18.

Таблица 18 – Бюджет затрат

Наименование	Сумма, руб.	Удельный вес, %
Материальные затраты	4200	0,86
Затраты на оборудование	120 000	24,5
Затраты на основную заработную плату	203 418	41,54
Затраты на дополнительную заработную плату	24 410	4,98
Отчисления во внебюджетные фонды	68 348	13,96
Накладные расходы	69 340	14,16
Общий бюджет	489 716	100

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

Оценка сравнительной эффективности исследования основывается на определении интегрального показателя ресурсоэффективности, который имеет следующий вид:

$$I_p = \sum a_i \cdot b_i \quad (5.11)$$

где I_p – интегральный показатель ресурсоэффективности; a_i – весовой коэффициент i -го параметра; b_i – балльная оценка i -го параметра, которая устанавливается экспертным путем по выбранной шкале оценивания.

Расчет интегрального показателя ресурсоэффективности представлен в таблице 19.

Таблица 19 – Оценка характеристик исполнения проекта

Критерий	Весовой коэффициент	Балльная оценка разработки
Способствует сокращению временных затрат на составление пищевого рациона	0,15	5
Удобство в эксплуатации (соответствует требованиям потребителей)	0,25	4
Энергосбережение	0,15	5
Надежность и помехоустойчивость	0,25	4
Возможность сопровождения и расширения функционала	0,2	5
Итого	1	4,5

Интегральный показатель ресурсоэффективности составил 4,5 баллов из 5 возможных, что свидетельствует об эффективности реализации проекта.

Выводы по разделу

В рамках раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» был оценен коммерческий потенциал разработки, потенциальные потребители результатов исследования, приведен анализ конкурентных решений. Основываясь на результатах проведенного в данном разделе анализа, можно сделать вывод, что проект является конкурентоспособным и перспективным. Стоимость разработки НИ составила 489 716 руб.

Глава 6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Разработка информационной системы (ИС) производится группой работников, состоящей из двух человек – руководителя и студента. Работа посвящена проектированию и разработке бэкэнд-части веб-приложения для составления пищевого рациона. Данное решение будет применяться людьми для сокращения временных затрат при составлении рациона, а также поможет людям с заболеваниями, при которых назначены диеты, соблюдать их.

Разработка информационной системы велась в городе Томске.

Выпускная квалификационная работа по разработке бэкэнд-части веб-приложения для составления рациона выполнялась в ходе преддипломной практики в жилом помещении. Рабочее место представляет собой комнату, характеристики которой приведены ниже:

- Ширина – 8,0 м, длина – 10,0 м, высота – 3,0 м;
- Площадь – 80 м^2 ;
- Объем – 240 м^3 ;
- В помещении имеется естественная вентиляция, дверь, окно, щели;
- В помещении установлено искусственное освещение, имеется естественное освещение.

В данном помещении максимальное количество сотрудников в одну смену – 2. В среднем на одного сотрудника приходится 40 м^2 площади и около 120 м^3 объема помещения. Данное помещение удовлетворяет санитарным нормам, согласно которым на одного работника должно приходиться не менее 6 м^2 площади и 24 м^3 объема.

6.1. Правовые и организационные вопросы обеспечения безопасности

Трудовой кодекс РФ устанавливает права и обязанности работника и работодателя, регулирует вопросы охраны труда, профподготовки, переподготовки и повышения квалификации, трудоустройства, социального партнерства [9]. Закрепляются правила оплаты и нормирования труда, порядок разрешения трудовых споров.

Работа в офисе относится ко второй категории тяжести труда – работы выполняются при оптимальных условиях внешней производственной среды и оптимальной величине физической, умственной и нервно-эмоциональной нагрузки.

Согласно основным положениям Трудового Кодекса РФ, в частности, рабочее время не должно превышать 40 часов в неделю. Возможно сокращение рабочего времени:

- не более 24 часа в неделю – для работников младше 16 лет;
- не более 35 часов – для работников, чей возраст в диапазоне от 16 до 18 лет, а также для инвалидов I и II группы;
- не больше 36 часов в неделю – для работников, работающих на рабочих местах с вредными условиями для жизни.

Температура в помещении должна быть оптимальной и в пределах от 20 до 24 градусов по Цельсию, с влажностью воздуха от 40 до 60 процентов и скоростью движения воздушного потока от 0,2 до 0,3 метров в секунду [10].

Вид трудовой деятельности за компьютерным устройством, в рамках выполнения выпускной квалификационной работы, относится к группе В – работа в режиме диалога с компьютерным устройством. Категория данной трудовой деятельности соответствует III (до 6 часов непосредственной работы за компьютером).

Рабочее место является частью рабочей зоны и представляет собой место постоянного или временного пребывания работника в процессе

трудовой деятельности. Рабочее место должно удовлетворять следующим требованиям [11]:

- обеспечивать возможность удобного выполнения работ;
- учитывать физическую тяжесть работ;
- учитывать технологические особенности процесса выполнения работ.

Согласно закону об информации, информационных технологиях и о защите информации, работодатель не вправе собирать, хранить, использовать и распространять информацию о частной жизни работников без их письменного согласия [9].

6.2. Производственная безопасность

Условия труда, в которых разрабатывается серверная часть веб-приложения, в том числе устройства, с помощью которых осуществляется проектирование и разработка решения, могут спровоцировать появление вредных и опасных факторов производства.

Согласно ГОСТ 12.0.003–2015 «Система стандартов безопасности труда (ССБТ)». Опасные и вредные производственные факторы. Классификация» при проектировании и разработке проектного решения могут возникнуть факторы, представленные в таблице ниже [12]:

Таблица 20 – Возможные опасные и вредные производственные факторы на рабочем месте инженера-программиста

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
Отсутствие или недостаток необходимого естественного освещения	СП 52. 13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95* [13]
Эмоциональные перегрузки	Трудовой кодекс Российской Федерации от 30.12.2001 №197-ФЗ (ред. от 01.03.2022) [9]
Перенапряжение зрительных анализаторов	-
Статические физические перегрузки	Приказ Минтруда России от 29 октября 2021 г. № 774н «Об утверждении общих требований к организации безопасного рабочего места»
Монотонность трудового процесса	-
Электрический ток	ГОСТ 12.1.038–82 ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов [14]

Производственный фактор является вредным в случае, если его воздействие на работающего в определенных условиях приводит к заболеванию или снижению работоспособности. Опасными считаются производственные факторы, воздействие которых на работающего в конкретных условиях может привести к травмам, а также другим внезапным резким ухудшениям здоровья.

6.2.1. Анализ опасных и вредных факторов и обоснование мероприятий по снижению их воздействия

6.2.1.1. Отсутствие или недостаток необходимого естественного освещения

Такой вредный фактор как недостаточная освещенность рабочей зоны возникает вследствие отсутствия должного количества источников освещения

в рабочей зоне. Недостаточная освещенность снижает работоспособность, значительно влияет на здоровье работников, а именно на их качество зрения.

В СП 52.13330.2016 [13] зрительная работа сотрудника, работающего с ПК охарактеризована как работа разряда Б – высокой точности (наименьший эквивалентный размер объекта различения – 0,3-0,5 мм), подразряда 1 (относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее 70%). Для снижения влияния фактора недостаточной освещенности на рабочем месте необходимо, чтобы уровень естественного освещения и яркость экрана персонального компьютера были приблизительно одинаковыми, так как яркий свет в зоне периферийного зрения заметно увеличивает глазное напряжение и приводит к быстрой утомляемости, а также коэффициент естественного освещения должен быть равен 4,2 и 1,5 при верхнем или комбинированном освещении и при боковом освещении соответственно. Путем решения проблемы недостаточной освещенности помещения может стать, например, расширение оконного проема.

6.2.1.2. Эмоциональные перегрузки

Эмоциональные перегрузки – состояние организма человека, которое проявляется после совершения работы и приводит к снижению его эффективности труда на определенный промежуток времени.

Чрезмерное эмоциональное напряжение может нести за собой возможные нарушения работы центральной нервной системы, а также развитие заболеваний сердечно-сосудистой системы. Помимо этого, характерными последствиями эмоциональных перегрузок являются нарушения сна, снижение работоспособности и концентрации.

Для снижения воздействия данного фактора на человека, необходимо придерживаться следующих правил: при вводе данных, редактировании программ, чтении информации с экрана непрерывная продолжительность работы не должна превышать 4-х часов при 8-часовом рабочем дне. Через

каждый час работы необходимо делать перерыв на 5-10 минут, а через два часа на 15 минут.

Также работнику необходимо соблюдать режим сна, вести здоровый образ жизни, заниматься спортом.

6.2.1.3. Перенапряжение зрительных анализаторов

Работа с использованием компьютеров предполагает длительное наблюдение за происходящим на экране монитора, это влечет за собой постоянное напряжение зрительного анализатора. Чтобы снизить нагрузку, необходимо обеспечить дополнительные перерывы для работников. В то же время, со стороны работника, необходимо совершать гимнастику для глаз для расслабления глазных мышц, а также следить за тем, чтобы время непрерывной работы за компьютером не превышало допустимого.

6.2.1.4. Статические физические перегрузки

Статические физические перегрузки возникают из-за неправильного оснащения рабочего места, например, стул на рабочем месте не позволяет регулировать высоту и наклон спинки.

Такого рода перегрузки отражаются, преимущественно, на опорно-двигательный аппарат и функциональные системы организма (сердечно-сосудистую, дыхательную и др.). Статические перегрузки могут приводить к таким заболеваниям, как: остеохондроз, сколиоз, варикозное расширение вен, а также общее снижение иммунитета.

Чтобы компенсировать этот фактор, необходимо руководствоваться приказом Минтруда России об организации безопасного рабочего места. Рабочее место должно быть оборудовано креслом с возможностью регулировки его высоты и наклона спинки [11].

6.2.1.5. Монотонность трудового процесса

Многие виды работы требуют от работника длительного выполнения однообразных действий или непрерывной и устойчивой концентрации

внимания. Поэтому монотонность является достаточно серьезным негативным фактором

В условиях монотонной работы с организмом человека могут произойти такие изменения как:

- изменение функционального состояния центральной нервной системы;
- снижение уровня бодрствования;
- нарушение автоматизма деятельности и способности к переключениям;
- изменение биологического ритма.

Так как работа разработчика программных систем связана только с работой на ПК, она является монотонной. Такая работа требует непрерывной концентрации внимания на протяжении длительного времени и является однообразной.

Для снижения уровня монотонности можно проводить следующие мероприятия:

- внедрение режима труда и отдыха;
- чередование операций и темпа их выполнения;
- частые, но кратковременные перерывы во время работы;
- выполнять физических упражнений в течение перерывов; менять положение выполнения работ (стоя – сидя);
- посещение специальных помещений психологической разгрузки и отдыха.

6.2.1.6. Электрический ток

Поражение электрическим током является одним из опасных факторов на рабочем месте. Результатом воздействия на организм человека электрического тока могут быть электротравмы, электрические удары и даже смерть.

При работе с компьютером, при прикосновении к его составляющим, могут возникнуть токи статического напряжения, которые в свою очередь, имеют свойство притягивать пыль и мелкие частицы к экрану. Пыль на экране ухудшает видимость, а при подвижности воздуха может попасть на поверхность кожи лица и в легкие, что вызывает заболевание кожи и дыхательных путей. В таблице 21 представлены предельно допустимые значения напряжения прикосновения и тока на рабочем месте разработчика-программиста, согласно ГОСТу 12.1.038-82 [14].

Таблица 21 – Предельно допустимые значения напряжения прикосновения и тока

Род тока	Напряжения прикосновения, В	Ток, мА
Переменный, 50 Гц	2,0	0,3
Постоянный	8,0	1,0

В качестве мер защиты нужно использовать оградительные устройства, устройства автоматического контроля и сигнализации, изолирующие устройства и покрытия, устройства защитного заземления.

6.3. Экологическая безопасность

При выполнении выпускной квалификационной работы, могут быть связаны негативно влияющие на экологию причины, сопутствующие эксплуатации компьютера. Показателями негативного влияния являются отходы, имеющие место на этапе производства компьютера и комплектующих, а также отходы, связанные с неполной их утилизацией. Кроме того, компьютерная техника является комплексом устройств, потребляющих электроэнергию, в связи с чем, нерациональное их использование может быть также расценено, как необоснованная нагрузка на окружающую среду [15].

Эксплуатация компьютерной техники может сопровождаться следующими негативными факторами влияния на окружающую среду:

- локальное повышение электромагнитного и радиоактивного фона;
- повышение интенсивности звукового фона;
- образование твердых отходов (компьютерный лом, бумага и т.п.) и жидких отходов (сточные воды);
- неоправданное потребление электроэнергии.

Производство компьютерных комплектующих загрязняет атмосферу путем выброса вредных веществ на производственных предприятиях, а также воздействует на гидросферу загрязненными атмосферными осадками.

Ниже изложены 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.

6.4. Безопасность в чрезвычайных ситуациях

Поскольку в каждом офисе существует канализация, системы отопления и водоснабжения, которые зависят от работы коммунальных служб, то авария на коммунальной системе жизнеобеспечения была выбрана как наиболее вероятная ЧС.

Аварии такого типа редко сопровождаются гибелью людей, однако создают существенные препятствия жизнедеятельности и работе, особенно в холодное время года.

Аварии на электроэнергетических системах могут привести к долговременным перерывам электроснабжения.

Аварии на канализационных системах способствуют массовому выбросу загрязняющих веществ и ухудшению санитарно-эпидемиологической обстановки.

Аварии в системах водоснабжения нарушают обеспечение населения водой или делают воду непригодной для питья.

Аварии на тепловых сетях в зимнее время года приводят к невозможности работы в не отапливаемых помещениях.

При возникновении ЧС, в первую очередь, следует сообщить об аварии диспетчеру Ремонтно-эксплуатационного управления или Жилищно-эксплуатационной конторы и попросить вызвать аварийную службу.

При скачках напряжения в электронной сети офиса необходимо незамедлительно обесточить все электронные приборы и выдернуть вилки из розеток для предотвращения пожара.

При исчезновении в водопроводной системе воды необходимо закрыть все открытые до этого краны.

При отключении отопительной системы, для обогрева помещения рекомендуется использовать обогреватели не самодельного, а заводского изготовления в качестве меры предосторожности от возгорания.

Рабочее помещение, представленное для выполнения ВКР, согласно СанПиН 1.2.3685-21, можно отнести к категории В (пожароопасное).

На основании Федерального закона от 22.07.2008 N 123-ФЗ (ред. от 30.04.2021) "Технический регламент о требованиях пожарной безопасности" классом возможного пожара является класс А: горение твердых веществ и материалов [16].

К первичным средствам пожаротушения относятся все виды переносных и передвижных огнетушителей, оборудование пожарных кранов, ящики с порошковыми составами (песок, перлит и т.п.), а также огнестойкие ткани (асбестовое полотно, кошма, войлок и т.п.).

Первичные средства пожаротушения должны размещаться в легкодоступных местах и не должны быть помехой и препятствием при эвакуации персонала из помещений.

Выводы по разделу

В результате проведенного анализа был рассмотрен процесс разработки системы с правовой, экологической, производственной точек зрения, а также обеспечения безопасности при чрезвычайных ситуациях. Рабочее место соответствует всем необходимым нормам. По электробезопасности помещение относится к категории безопасное. По тяжести труда в соответствии с СанПиН 1.2.3685-21 работа при разработке проектного решения относится к категории I. Рабочее помещение, представленное для выполнения ВКР, согласно СанПиН 1.2.3685-21, можно отнести к категории В (пожароопасное). Согласно постановлению об утверждении критериев отнесения объектов, оказывающих негативное воздействие на окружающую среду с изменениями на 7 октября 2021 года, рабочее помещение относится к объектам IV категории – объекты, оказывающие минимальное негативное воздействие на окружающую среду.

Заключение

Составление рациона питания позволяет человеку поддерживать баланс макронутриентов в организме, достигать поставленных целей в спорте, а также придерживаться определенной диеты без риска навредить здоровью. Перенос этого процесса в цифровое пространство позволит человеку сократить затраты временного ресурса, и, как следствие, еще эффективнее придерживаться здорового образа жизни.

Основные результаты проведенной работы:

- Выполнен анализ доступных на российском рынке аналогов с похожим функционалом.
- Спроектирована архитектура веб-приложения.
- Проведена реализация серверной части веб-приложения.
- Описан функционал реализованного веб-приложения.

Как следствие, было разработано веб-приложение для составления рациона питания. При разработке использовался язык JavaScript и такие технологии как Node.js и Express, в качестве СУБД использовалась PostgreSQL.

Список литературы

1. Befit [Электронный ресурс] – Как вести расчет БЖУ (белков, жиров и углеводов) и для чего это надо? / – Свободный доступ из сети Интернет. Схема доступа: <https://letbefit.ru/blog/kak-vesti-raschet-bzhu-belkov-/> (дата обращения: 02.04.2022).
2. Органическая продукция без глютена [Электронный ресурс] – что такое сбалансированное питание, и с чем его едят? / – Свободный доступ из сети Интернет. Схема доступа: <https://bio-xxi.ru/blog/chto-takoe-sbalansirovannoe-pitanie-i-s-chem-ego-edyat/#>. (дата обращения: 02.04.2022).
3. Google Play [Электронный ресурс] – Дневник зожника / – Свободный доступ из сети Интернет. Схема доступа: <https://play.google.com/store/apps/details?id=com.tvoydnevnik> (дата обращения: 07.04.2022).
4. Alserstroy [Электронный ресурс] – Формула правильного похудения – как рассчитать баланс БЖУ / – Свободный доступ из сети Интернет. Схема доступа: <https://alserstroy.ru/formula-pravilnogo-pohudeniya-kak-rasschitat-balans-bzhu-skolko/> (дата обращения: 07.04.2022).
5. ЦОМид Магнитогорск [Электронный ресурс] – Лечебные столы (диеты) / – Свободный доступ из сети Интернет. Схема доступа: https://cmd74.ru/lechebnye-stoly-diety/#_1 (дата обращения: 07.04.2022).
6. Stack Overflow [Электронный ресурс] – Stack Overflow Developer Survey 2020 / – Свободный доступ из сети Интернет. Схема доступа: <https://insights.stackoverflow.com/survey/2020> (дата обращения: 15.04.2022).
7. CoderNet [Электронный ресурс] – Плюсы и минусы веб-разработки на Node.js / – Свободный доступ из сети Интернет. Схема доступа: https://codernet.ru/articles/web/plyusyi_i_minusyi_razrabotki_veb-prilozhenij_na_nodejs/ (дата обращения: 17.04.2022).

8. Завтра облачно [Электронный ресурс] – PostgreSQL или MySQL, как из этих СУБД лучше впишется в ваш проект / – Свободный доступ из сети Интернет. Схема доступа: <https://mcs.mail.ru/blog/postgresql-ili-mysql-kakaya-iz-etih-relyacionnyh-subd> (дата обращения: 17.04.2022).
9. Трудовой кодекс Российской Федерации от 30.12.2001 №197-ФЗ (ред. от 01.03.2022) – Текст: непосредственный.
10. ГОСТ 12.1.005–88 Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны (правовые нормы) – Текст: непосредственный.
11. Приказ Минтруда России от 29 октября 2021 г. № 774н «Об утверждении общих требований к организации безопасного рабочего места» – Текст: непосредственный.
12. ГОСТ 12.0.003–2015 Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация (с Поправками) ГОСТ от 09 июня 2016 г. № 12.0.003-2015 – Текст: непосредственный.
13. СП 52.13330.2016. Естественное и искусственное освещение: утвержден Приказом Министерства строительства и жилищно-коммунального хозяйства Российской Федерации от 7 ноября 2016 г. N 777/ введен с 8 мая 2017 г. – Текст: непосредственный.
14. ГОСТ 12.1.038–82 ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов (с Изменением N 1) – Текст: непосредственный.
15. ГН 2.2.5.2308-07 «Ориентировочные безопасные уровни воздействия (ОБУВ) вредных веществ в воздухе рабочей зоны».
16. Федеральный закон от 22.07.2008 N 123-ФЗ (ред. От 30.04.2021) «Технический регламент о требованиях пожарной безопасности» – Текст: непосредственный.

Приложение А

```
const ApiError = require('../error/ApiError');
const bcrypt = require('bcrypt')
const {calculateCalories, calculatePFC} =
require('../calculations/calculations')
const jwt = require('jsonwebtoken')
const {User, Product, ActivityLevel, Target} =
require('../models/models')

const generateJwt = (id, email, role, name) => {
  return jwt.sign(
    {id, email, role, name},
    process.env.SECRET_KEY,
    {expiresIn: '24h'}
  )
}

class UserController {
  async registration(req, res, next) {
    const {email, password, name} = req.body

    let role = 'USER'
    if (!email || !password || !name) {
      return next(ApiError.badRequest('Некорректно введены данные'))
    }

    const candidate = await User.findOne({where: {email}})
    if (candidate) {
      return next(ApiError.badRequest('Пользователь с таким email уже
существует'))
    }

    const hashPassword = await bcrypt.hash(password, 5)

    if (email == 'envyeah@gmail.com') {
      role = 'administrator'
    }
  }
}
```

```

}

const user = await User.create({
  email,
  password: hashPassword,
  role,
  name

})

const token = generateJwt(user.id, email, role, name)
return res.json({token})
}

async login(req, res, next) {
  const {email, password} = req.body
  const user = await User.findOne({where: {email}})
  if (!user) {
    return next(ApiError.internal('Пользователь не найден'))
  }
  let comparePassword = bcrypt.compareSync(password,
user.password)
  if (!comparePassword) {
    return next(ApiError.internal('Указан неверный пароль'))
  }
  const token = generateJwt(user.id, user.email, user.role,
user.name)
  return res.json({token})
}

async check(req, res, next) {
  const token = generateJwt(req.user.id, req.user.email,
req.user.role, req.user.name)
  return res.json({token})
}

```

```

async getOne(req, res) {
  const {id} = req.params
  const user = await User.findOne({where: {id}})
  return res.json(user)
}

async setCpfc(req, res, next) {
  try {
    const {age, weight, coefficientId, height, gender, targetId} =
req.body
    const alCoef = await ActivityLevel.findOne({
  where: {
    id: coefficientId
  }
})
    const targetCoef = await Target.findOne({
  where: {
    id: targetId
  }
})
    const calories = calculateCalories(age, weight,
alCoef.coefficient, height, gender, targetCoef.coefficient)
    const pfc = calculatePFC(calories, targetCoef.name)

    const token = req.headers.authorization.split(' ')[1]
    const decoded = jwt.verify(token, process.env.SECRET_KEY)

    const user = await User.update({
  calories: calories.average,
  fats: pfc.fats,
  proteins: pfc.proteins,
  carbohydrates: pfc.carbohydrates
},
{
  where: {
    id: decoded.id

```

```

    }
  })

  const userToShow = await User.findOne({
    where: {
      id: decoded.id
    }
  })

  return res.json(userToShow)

} catch (e) {
  return next(ApiError.badRequest(e.message))
}
}

async updateUserData(req, res, next) {
  try {
    const {email, name, age, weight} = req.body
    const token = req.headers.authorization.split(' ')[1]
    const decoded = jwt.verify(token, process.env.SECRET_KEY)

    if (email) {
      const candidate = await User.findOne(
        {
          where: {
            email: email
          }
        }
      )
      if (candidate) {
        return next(ApiError.badRequest('email занят'))
      }
    }

    const oldUser = await User.findOne({

```

```
where: {
  id: decoded.id
}
})
await User.update({
  id: oldUser.id,
  email: email,
  password: oldUser.password,
  role: oldUser.role,
  name: name,
  age: age,
  weight: weight
},
{
  where: {
    id: oldUser.id
  }
})

return res.send('Пользователь обновлен')

} catch (e) {
return next(ApiError.badRequest(e.message))
}
}
}

module.exports = new UserController()
```

Приложение Б

```
const {Recipe, Product, RecipeProduct, RecipeStep} =
require('../models/models')
const ApiError = require('../error/ApiError');
const uuid = require('uuid')
const path = require('path')
const jwt = require('jsonwebtoken')

class RecipeController{
  async create(req, res, next) {
    try{
      const token = req.headers.authorization.split(' ')[1]
      const decoded = jwt.verify(token, process.env.SECRET_KEY)

      let {name, calories, proteins, fats, carbohydrates,
description, products, steps} = req.body
      const {image} = req.files
      let fileName = uuid.v4() + ".jpg"
      image.mv(path.resolve(__dirname, '..', 'static',
fileName))

      const recipe = await Recipe.create({
        name,
        calories,
        proteins,
        fats,
        carbohydrates,
        description,
        userId: decoded.id,
        image: fileName
      })

      products = JSON.parse(products)
      for (const p of products) {
        let productId = await Product.findAll({
```

```

where: {
  id: p.id
}
})
await RecipeProduct.create({
  recipeId: recipe.id,
  productId: productId[0].dataValues.id,
  weight: p.weight
})
}
steps = JSON.parse(steps)
for (const s of steps) {
  await RecipeStep.create({
    recipeId: recipe.id,
    step_description: s.step_description,
    step_number: s.step_number
  })
}
return res.json(recipe)
} catch (e){
  next(ApiError.badRequest(e.message))
}
}
async getAll(req, res){
  const recipes = await Recipe.findAll()
  return res.json(recipes)
}
async getOne(req, res){
  const {id} = req.params
  const recipe = await Recipe.findOne({where: {id}})
  return res.json(recipe)
}
}

module.exports = new RecipeController()

```