

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники
Направление подготовки информационные системы и технологии
Отделение школы (НОЦ) информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка информационной системы формата "Вопрос-Ответ" (Q&A) с использованием Ruby on Rails

УДК 004.78:004.455

Студент

Группа	ФИО	Подпись	Дата
8И4А	Юрканский Даниил Яковлевич		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Суходоев М.С.	К.Т.Н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст. преподаватель ОСГН	Хаперская А.В.			

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОКД	Штейнле А.В.	К.М.Н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко И.В.	К.Т.Н.		

Томск – 2018 г.

РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Код результатов	Результат обучения (выпускник должен быть готов) Профессиональные и общепрофессиональные компетенции
P1	Применять базовые и специальные естественнонаучные и математические знания для комплексной инженерной деятельности по созданию, внедрению и эксплуатации геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием геоинформационных систем и технологий, информационных систем в бизнесе, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Выполнять комплексные инженерные проекты по созданию информационных систем и технологий, а также средств их реализации (информационных, методических, математических, алгоритмических, технических и программных).
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P6	Внедрять, эксплуатировать и обслуживать современные геоинформационные системы и технологии, информационные системы и технологии в бизнесе, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
	Универсальные (общекультурные) компетенции
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности
P8	Осуществлять коммуникации в профессиональной среде и в обществе в целом. Владеть иностранным языком (углублённый английский язык), позволяющим работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена команды, состоящей из специалистов различных направлений и квалификаций.
P10	Демонстрировать личную ответственность за результаты работы и готовность следовать профессиональной этике и нормам ведения комплексной инженерной деятельности.
P11	Демонстрировать знания правовых, социальных, экологических и культурных аспектов комплексной инженерной деятельности, а также готовность к достижению должного уровня физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности

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

Инженерная школа информационных технологий и робототехники
Направление подготовки информационные системы и технологии
Отделение школы (НОЦ) информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

В форме:

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

Студенту

Группа	ФИО
8И4А	Юрканскому Даниилу Яковлевичу

Тема работы:

Разработка информационной системы формата "Вопрос-Ответ" (Q&A) с использованием Ruby on Rails	
Утверждена приказом директора (дата, номер)	Приказ №2063/с от 23.03.2018 г.
Срок сдачи студентом выполненной работы:	

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе (наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).	1. Разработка информационной системы формата "Вопрос-Ответ" (Q&A) с использованием Ruby on Rails
Перечень подлежащих исследованию, проектированию и разработке вопросов (аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).	1. Провести анализ существующих решений Q&A систем. 2. Запланировать этапы разработки (итерации) 3. Разработать систему согласно итерациям и покрыть ее тестами. 4. Провести тестовое развертывание системы на реальном сервере.
Перечень графического материала (с точным указанием обязательных чертежей)	Презентация в формате *.ppt
Консультанты по разделам выпускной квалификационной работы (с указанием разделов)	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Хаперская А.В.
Социальная ответственность	Штейнле А.В.
Названия разделов, которые должны быть написаны на русском и иностранном языках:	
Заключение	

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

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Суходоев М.С.	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И4А	Юрканский Д.Я.		

Министерство образования и науки Российской Федерации

федеральное государственное автономное образовательное учреждение
высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа Информационных технологий и робототехники
Направление подготовки (специальность) Информационные системы и технологии
Уровень образования Бакалавриат
Отделение школы (НОЦ) Информационных технологий
Период выполнения весенний семестр 2017/2018 учебного года

Форма представления работы:

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

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

Студент

Срок сдачи студентом выполненной работы:	
--	--

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	Основная часть	75
	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
	Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Суходоев М.С.	К.Т.Н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко И.В.	К.Т.Н.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА

«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8И4А	Юрканский Даниил Яковлевич

Инженерная школа	Информационных технологий и робототехники	Отделение (НОЦ)	Информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	Информационные системы и технологии

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

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Работа с информацией, представленной в российских и иностранных научных публикациях, аналитических материалах, статистических бюллетенях и изданиях, нормативно-правовых документах.
2. Нормы и нормативы расходования ресурсов	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Оценка потенциальных потребителей исследования, SWOT-анализ.
2. Определение возможных альтернатив проведения научных исследований	Определение возможных альтернатив с помощью морфологического подхода.
3. Планирование и формирование бюджета научных исследований	Планирование этапов работ, определение трудоемкости работы и построение календарного графика.

Дата выдачи задания для раздела по линейному графику	06.02.2018
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст. преподаватель ОСГН	Хаперская А.В.			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И4А	Юрканский Даниил Яковлевич		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

Группа	ФИО
8И4А	Юрканский Даниил Яковлевич

Инженерная школа	Информационных технологий и робототехники	Отделение	Информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	Информационные системы и технологии

Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	<i>Разработать информационную систему формата "Вопрос- Ответ" (Q&A) с использованием фреймворка Ruby on Rails</i>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Производственная безопасность 1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения. 1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения.	1.1. Выявленные вредные факторы при разработке и эксплуатации проектируемого решения: – отклонение показателей микроклимата рабочей зоны; – недостаточная освещенность рабочей зоны; 1.2. Выявленные опасные факторы при разработке и эксплуатации проектируемого решения: – опасность поражения электрическим током; – пожароопасность.
2. Экологическая безопасность: Анализ воздействия объекта на окружающую среду; Разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.	Рассмотрены негативно влияющие на экологию факторы при эксплуатации компьютера. Решения по обеспечению экологической безопасности согласно нормативным документам.
3. Безопасность в чрезвычайных ситуациях	3.1. Перечень возможных ЧС, которые могут возникнуть при работе в помещении офиса. 3.2. Способы защиты от пожара и ликвидация последствий.
4. Правовые и организационные вопросы обеспечения безопасности	Организационные мероприятия по обеспечению безопасности трудящихся за персональным компьютером.

Дата выдачи задания для раздела по линейному графику	06.02.2018
--	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОКД	Штейнле А.В.	к.м.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И4А	Юрканский Даниил Яковлевич		

Реферат

Выпускная квалификационная работа 100с., 9 рис., 17 табл., 21 источник, 9 прил.

Ключевые слова: веб-приложение, информационная система, разработка через тестирование, гибкая методология разработки, система формата «вопрос-ответ», web application, information system, test driven development, agile, question&answer system.

Объект исследования: системы формата "Вопрос-ответ" и использование фреймворка Ruby on Rails для разработки информационной системы.

Цель работы – разработка системы формата "Вопрос-ответ" (Q&A), используя фреймворк Ruby on Rails.

Область применения: среда Интернет и различные сайты фирм и учреждений любого размера и назначения.

Степень внедрения: система готова к внедрению.

Экономическая значимость работы: выявлена стоимость ресурсов проведения научного исследования, затраты на зарплаты, отчисления и накладные расходы. Определена ресурсная (ресурсосберегающая), финансовая, бюджетная, социальная и экономическая эффективности исследования.

Обозначения и сокращения

RoR (Ruby on Rails) – фулстек фреймворк для разработки веб-приложений, написанный на динамическом языке программирования Ruby.

TDD (Test Driven Development) – техника разработки программного обеспечения, которая основывается на повторении очень коротких циклов разработки: сначала пишется тест, покрывающий желаемое изменение, затем пишется код, который позволит пройти тест, и под конец проводится рефакторинг нового кода к соответствующим стандартам.

BDD (Behaviour Driven Development) – разработка с опережающим описанием поведения программы. Сначала описывается в спецификации то, как должна себя вести программа, а затем пишется сама программа.

Q&A system (Question and Answers system) – система вопросов и ответов.

Оглавление

Введение	14
1 Теоретический анализ	16
1.1 Принцип работы системы вопросов-ответов	16
1.2 Обзор существующих решений	17
1.3 Анализ требований пользователей	18
1.4 Анализ средств разработки	20
2 Проектирование	22
2.1 Принцип работы фреймворка и структура системы	22
2.2 Проектирование системы	24
2.3 Планируемые итерации	26
3 Этапы реализации	29
3.1 Создание ресурса пользователей, ответов и вопросов	29
3.2 Управление ресурсами через AJAX и прикрепление файлов	31
3.3 Оценивание вопросов и ответов	36
3.4 Комментарии и динамическое обновление страницы через WebSockets... ..	39
3.5 Аутентификация через социальные сети и REST API.....	43
3.6 Поиск и фоновые задачи в системе	47
3.7 Тестирование системы в production окружении	49
4 Результаты работы.....	50
5 Финансовый менеджмент.....	53
5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения	53
5.1.1 Потенциальные потребители результатов исследования.....	53
5.1.2 Анализ конкурентных технических решений.....	53
5.1.3 Технология QuaD	54
5.1.4 SWOT-анализ.....	55
5.2 Планирование научно-исследовательских работ.....	56

5.2.1	Структура работ в рамках научного исследования	56
5.2.2	Определение трудоемкости выполнения работ	57
5.2.3	Разработка графика проведения научного исследования	59
5.3	Бюджет научно-технического исследования (НТИ).....	60
5.3.1	Расчет материальных затрат НТИ	60
5.3.2	Основная заработная плата исполнителей темы	61
5.3.3	Дополнительная заработная плата	63
5.3.4	Отчисления во внебюджетные фонды (страховые отчисления).....	64
5.3.5	Накладные расходы.....	65
5.3.6	Формирование бюджета затрат научно-исследовательского проекта	65
5.4	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	66
6	Социальная ответственность	70
6.1	Производственная безопасность.....	71
6.2	Вредные производственные факторы.....	72
6.2.1	Отклонение показателей микроклимата в помещении	72
6.2.2	Недостаточная освещенность рабочей зоны.....	74
6.3	Опасные производственные факторы	75
6.3.1	Опасность поражения электрическим током	75
6.3.2	Пожаровзрывобезопасность	77
6.4	Экологическая безопасность.....	78
6.4.1	Анализ воздействия продукта на окружающую среду.....	78
6.4.2	Решения по обеспечению экологической безопасности	78
6.5	Безопасность в чрезвычайных ситуациях	79
6.5.1	Перечень возможных ЧС при разработке и эксплуатации проектируемого решения	79
6.5.2	Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.....	80

6.6	Правовые и организационные вопросы обеспечения безопасности.....	82
6.6.1	Организационные мероприятия при компоновке рабочей зоны	83
6.6.2	Специфика влияния продукта на рабочий процесс	85
	Заключение.....	86
	Conclusion.....	87
	Список использованных источников.....	88
	Приложение А.....	91
	Приложение Б	92
	Приложение В.....	93
	Приложение Г	94
	Приложение Д.....	95
	Приложение Е	97
	Приложение Ж	98
	Приложение З.....	99
	Приложение И.....	100

Введение

Система вопросов и ответов – система, которая позволяет пользователям получить решение интересующей их проблемы. Спектр проблем зависит лишь от тематики сайта, где развернута данная система. Такой сервис выгодно отличается от обычной службы технической поддержки или простого поиска в интернете, он позволяет получить ответ 24 часа в сутки и различные решения от разных пользователей, которые не всегда выглядят очевидными и простыми со стороны обычного пользователя.

Подобные сервисы повсеместно распространены в интернете, их используют как полноценные системы, так и как дополнение к сайту. Они могут быть реализованы просто как темы на форуме, часто самые популярные вопросы выносятся в так называемый FAQ – Frequently Asked Questions (часто задаваемые вопросы). В качестве примеров полноценных систем можно привести такие популярные сервисы как `stackoverflow`, `otvet.mail.ru` и `askubuntu`.

Хорошая система должна быть легко масштабируема, гибка для изменений и актуальна на текущий момент времени, что означает использование современных технологий при ее разработке. В этой связи тема данной выпускной квалификационной работы – «Разработка информационной системы формата ‘Вопрос-Ответ’ (Q&A) с использованием Ruby on Rails».

Целью ВКР является создание сервиса формата вопросов и ответов с нуля, используя фреймворк Ruby on Rails. Задачей – разработать информационную систему для подобного сервиса.

Использование готового решения гораздо проще, чем разработка собственного с нуля, но оно не дает нужной гибкости и простоты расширения функционала, а разобраться в чужом коде всегда сложнее, чем собственном. Кроме того, не все готовые решения предоставляют REST API для разработки мобильных приложений, что крайне актуально в современном мире.

Более того, в процессе разработки планируется применить такие современные методологии как Agile и TDD/BDD, которые позволят произвести гибкую итерационную разработку и получить полностью покрытую тестами систему.

В связи с вышеперечисленным было решено разработать собственную информационную систему формата вопросов и ответов на фреймворке Ruby on Rails.

1 Теоретический анализ

1.1 Принцип работы системы вопросов-ответов

Типичный сайт-сервис формата вопросов-ответов предоставляет возможность любому пользователю задать вопрос и получить ответ, или же самому ответить на вопрос другого пользователя. Кроме того, существует возможность комментирования вопросов и ответов для небольших пояснений или уточнений. Автор вопроса может выбрать лучший ответ, каждый вопрос и ответ имеет свой рейтинг, полученный в результате голосований пользователя. Обычно данные системы также имеют гибкий поиск с различными фильтрами и облако тегов для метки темы вопросов.

На главной странице обычно выводится список последних вопросов, добавленных пользователями в системе, задавать вопрос и отвечать на него может только зарегистрированный пользователь (можно настроить авторизацию пользователей и по-другому, например, сделать различные роли и группы пользователей).

Аутентификация возможна как с помощью соц. сетей, так и обычная регистрация, зависит от конкретного решения и пожеланий заказчика.

Обычно последовательность взаимодействия такая:

1. пользователь задает вопрос;
2. другие пользователи отвечают на вопрос, помогают (или не помогают) решить проблему;
3. если пользователя удовлетворяют ответы, он выбирает ответ как правильный;
4. другие пользователи голосуют за вопрос и ответы.

В чем главный минус открытых готовых решений? Они либо предоставляют недостаточный функционал, либо слишком много лишнего, что значительно утяжеляет систему, делает ее громоздкой и медленной. Это все

зависит от конкретного технического задания и параметров системы, которые нужны заказчику.

1.2 Обзор существующих решений

Перед проектированием собственного решения необходимо проанализировать уже готовые решения для Q&A систем, выявить их достоинства и недостатки, провести сравнительный анализ.

1. OSQA

Написанное на Python и Django open-source решение для Q&A систем. Предоставляет два варианта продукта: AnswerHub – платный вариант для крупных корпораций, и само OSQA не такое большое, но и с ограниченным набором функционала. Например, бесплатное решение не имеет REST API, возможность подписки на вопрос и других возможностей [1]. Бесплатная версия не совсем соответствует нужным требованиям, с самостоятельным расширением и изменением могут возникнуть проблемы, поэтому такой вариант системы проигрывает самостоятельной разработке с нуля.

AnswerHub является платной опцией и предоставляет огромное число особенностей: аналитику, защиту от спама, мультиязычность, интеграцию с Slack и различными соц. сетями, альтер-его пользователей, фильтр плохих слов, закрепление постов и многое другое [1]. Система отлично подходит для крупных корпораций с большим бюджетом, но не отвечает требованиям в нашем случае.

2. Askbot

Также написанная на Python и Django Q&A система, имеет пробную бесплатную версию и несколько платных планов в зависимости от нужд сервиса. Имеет ограничение по числу пользователей, предоставляет REST API, но только для GET запросов и лишь малое количество, текущая версия API явно не подходит для создания мобильного приложения.

Для неограниченного числа пользователей необходим индивидуальный план ценообразования, причем точно больше 50\$ в месяц. Исходя из этого, можно

прийти к выводу, что данный вариант опять же проигрывает самостоятельной гибкой разработке с нуля, но вдобавок и AnswerHub решению, который выглядит значительно масштабнее для крупных корпораций [2].

3. Question2Answer

Q&A – система, написанная на PHP. Имеет следующие особенности: быстрый поиск, категории, теги, система репутации, оповещения по email, гибкое управление аккаунтами, включая личные сообщения и стену, аутентификация с помощью соц. сетей, интеграция с сайтами (например, на WordPress), различные языковые пакеты и имеет некоторое количество юнит-тестов.

Решение является бесплатным и не имеет конкретных ограничений, вполне подходит для небольших компаний, но не имеет REST API для создания мобильного приложения в будущем.

Поэтому разработка собственного решения с нуля, которое было бы полностью покрыто как юнит, так и интеграционными тестами выглядит предпочтительней этого варианта.

1.3 Анализ требований пользователей

Существует большое множество платных и бесплатных решений Q&A систем, некоторые из них приведены в разделе 1.2. В связи с этим собственный продукт должен быть конкурентоспособным и иметь некоторые выгодные отличительные особенности по сравнению с другими продуктами.

Как было выявлено, REST API имеет далеко не каждая Q&A система. Между тем, это важная часть современного веб-приложения, особенно если оно рассматривается и в плоскости веб-сервиса для реализации мобильных приложений. В связи с этим, можно с уверенностью утверждать, что разработанная в данном исследовании система обязательно должно иметь собственный REST API.

Как и типовые системы данного формата, система должна предоставлять следующий функционал:

- возможность задать вопрос;
- возможность ответить на вопрос;
- возможность прокомментировать вопрос или ответ;
- возможность проголосовать за вопрос/ответ;
- некая система поиска по вопросам, ответам и комментариям и/или облако тегов;
- возможность следить за новыми ответами через email рассылку.

Остальные черты системы являются исключительно индивидуальными и зависят от разработчиков и конкретных требований. Кому-то нужна система кармы или репутации, возможность входа через различные соц. сети, интеграция со сторонними платформами и сервисами или же интеграция со своим внутренним продуктом.

Для разрабатываемого приложения было принято включить следующие отличительные черты кроме основных перечисленных выше:

- аутентификация через соц. сети;
- ежедневная рассылка дайджеста новых вопросов в системе;
- появление новых вопросов, ответом и комментариев без обновления страницы.

Кроме того, для разработчика в процессе расширения системы, добавления новых модулей, очень важно, чтобы эти изменения происходили как можно безболезненней. Ручное тестирование всего функционала системы после внесения изменений, это не очень выгодный вариант. Гораздо лучше, иметь покрытую тестами систему, тесты же помогают проверить весь функционал за считанные минуты и выявить проблемные зоны. Благодаря такому подходу разработки, можно добавлять новые модули и не бояться сломать старое.

1.4 Анализ средств разработки

Система должна обладать гибкостью, быть современной, иметь REST API и покрыта тестами, исходя из этого необходимо выбрать, какие технологии будут использоваться для разработки.

Среди возможных вариантов для разработки системы были рассмотрены следующие варианты:

1. C# и ASP.NET фреймворк;
2. PHP и Laravel фреймворк;
3. Java и Spring фреймворк;
4. Ruby on Rails фреймворк.

Ввиду наличия опыта разработки веб-приложений с использованием фреймворка Ruby on Rails, в качестве языка для разработки информационной системы был выбран Ruby, современный динамический полностью объектно-ориентированный язык программирования, который еще называют «английский для программистов» за его простоту чтения кода. Ruby on Rails – современный фулстек фреймворк, включающий в начальный комплект несколько модулей для работы как с бэкенд, так и фронтенд частью (например, модуль для задач по расписанию, работы с WebSockets, рассылкой email) и соответствующей MVC и RESTful архитектуре [3].

Кроме того, RoR включает в себя библиотеку i18n для интернационализации приложения, которая позволяет переводить систему на разные языки, формировать множественное число и предоставляет ряд других возможностей [4].

Благодаря тому, что Ruby интерпретируемый язык программирования, сервер приложения можно запустить в режиме консоли, что сильно помогает в процессе разработки и тестирования программы.

В состав фреймворка также входит модуль для тестирования приложения, вдобавок к нему существует несколько библиотек для написания тестов в RoR [4].

Для разработки были выбраны TDD/BDD и Agile методологии. Стоит отметить, что в данном случае никого заказчика нет, но Agile представляет собой гибкий итеративный подход к разработке, в котором используются пользовательские истории, динамические требования к продукту, который изменяются в результате новых обстоятельств [5]. Agile методология предполагает реализацию от простого к сложному, где реализуется только то, что необходимо на данный момент. Это очень хорошо соответствует духу фреймворка Ruby on Rails, который построен на принципах Agile.

Разработка через тестирование имеет примерно следующий вид: сначала пишется тест, где описывается желаемый алгоритм действия кода и результат, который хотят получить, затем пишется сам код, на третьем этапе выполняется рефакторинг теста и самого кода. Данная парадигма называется RGR – Red, Green, Refactirong, что означает тест сначала не проходит, потом написан код и тест проходит (зеленый) и наконец выполняется рефакторинг [6].

Подход к написанию тестов через описание поведения с точки зрения пользователей, называется Behavior Driven Development (BDD). BDD тесты понятны для простого пользователя, а не только для разработчика, они написаны на интуитивно понятном уровне с помощью английского языка [7].

Такой подход отлично подходит для фреймворка Ruby on Rails, который имеет несколько вариантов для написания тестов подобного характера, как юнит, так и интеграционных.

2 Проектирование

2.1 Принцип работы фреймворка и структура системы

RoR – фулстек фреймворк, имеющий клиент-серверную архитектуру, в качестве клиента выступает любое устройство, на котором запущено приложение в веб-браузере, сервером является само приложение, а точнее сервер приложения – Puma, Unicorn или Passenger [4].

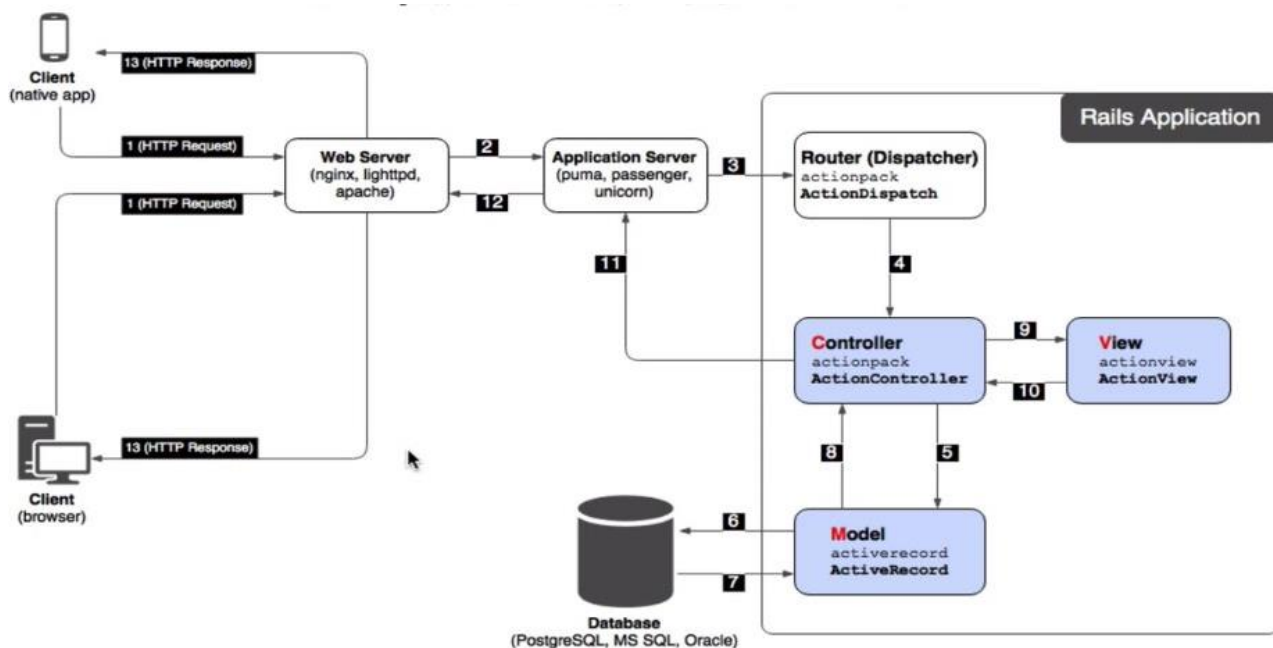


Рисунок 1 – обработка запросов в RoR

Клиент-серверное взаимодействие происходит по следующей схеме:

1. клиент отправляет http запрос веб-серверу (nginx, apache, etc.);
2. далее этот запрос передается одному из экземпляров (инстансов) сервера приложений в зависимости от нагрузки на них;
3. после этого запрос передается модулю ActionDispatch;
4. ActionDispatch отвечает за маршрутизацию, т.е. за передачу запроса нужному контроллеру;
5. контроллер обрабатывает запрос и отдает его модели, которая является надстройкой над БД;
6. модель, уровень которой является ORM и реализован с помощью Active Record, запрашивает нужные данные из БД;

7. в БД выполняется SQL запрос на получение или изменение данных, затем эти данные передаются обратно модели;
8. затем модель отдает нужные данные контроллеру;
9. после контроллер отправляет данные на уровень представления (View);
10. как только данные переданы View, происходит обратный путь, теперь запрос идет обратно к контроллеру для формирования ответа;
11. от контроллера ответ переходит серверу приложений;
12. сервер приложений передает его веб-серверу;
13. и, в конце концов, веб-сервер передает сформированный ответ клиенту.

Как видно из представленного выше пути взаимодействия, при проектировании системы нужно будет разработать три уровня абстракции: Model – уровень абстракции БД и бизнес-логики, Controller – уровень логики приложения и передача данных во View, View – представление данных в пригодном для пользователя виде.

Контроллеры содержат стандартный набор CRUD операций с ресурсами, которые в Rails называются new (GET), create (POST), edit (GET, PUT), destroy (DELETE), также там размещаются методы и для нестандартных операций с ресурсами. Модели определяют связи на уровне моделей, различные проверки и методы для доступа к данным, также в моделях присутствует вся сложная бизнес-логика системы (если она не помещена в другом разделе приложения, например, в lib или app/[directory_name]).

Уровень представлений может быть написан с помощью различных шаблонизаторов, например, ERB или SLIM, которые затем переводят этот код в HTML, при написании представлений используются так называемые helpers (хелперы), которые выполняют бизнес-логику или точнее сказать приведение данных в конечный вид для пользователя, потому что в MVC паттерне на уровне View не должны выполняться дополнительные сложные операции над данными для выдачи пользователю [4].

В качестве СУБД была выбрана PostgreSQL – одна из самых популярных систем управления базами данных в мире, которая зачастую используется в RoR приложениях.

REST API в системе будет также реализован с помощью контроллеров, только ответ будет отдаваться в формате JSON.

Фоновые задачи и задачи по расписанию будут реализованы с помощью модуля ActiveJob и библиотек (гемов) для Rails.

Для оформления клиентского интерфейса будет использована CSS-библиотека Bootstrap 4, один из ее последних релизов.

2.2 Проектирование системы

Поскольку была выбрана итеративная методология разработки «от простого к сложному», то на этапе проектирования системы необходимо обдумать базовый функционал – самую первую итерацию (пользовательскую историю).

Для системы формата «Вопрос-ответ» базовым функционалом является возможность задавать вопрос и отвечать на вопрос.

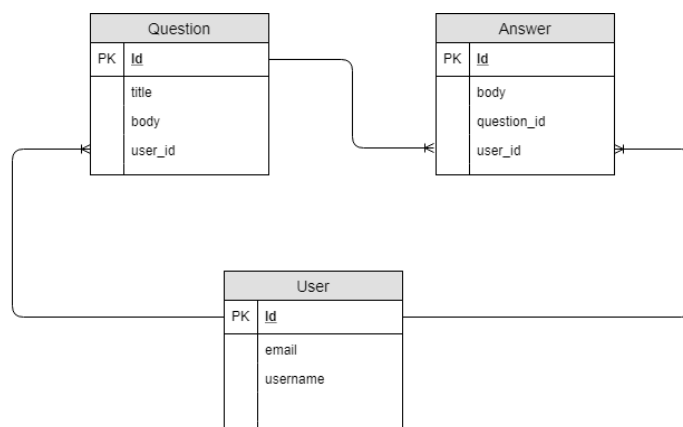


Рисунок 2 – начальная схема БД

На первой итерации системы необходимы лишь три таблицы в БД:

- User с полями типа string email и username, стоит отметить что для аутентификация будет использована библиотека Devise, которая добавляет множество полей для User, которые не указаны на схеме. Например: confirmed_at, encrypted_password, current_sign_in_ip,

reset_password_token, created_at, updated_at и многие другие. Поля created_at и updated_at являются так называемыми timestamps, которые желательны для каждого ресурса в базе данных. В дальнейшем в эту таблицу будут добавлены и другие поля, которые затрагивают следующие итерации;

- Question с полями типа string title, body и user_id – внешний ключ для связи с таблицей User;
- Answer с полем string body и внешними ключами user_id и question_id. Question и answer также будут содержать timestamps, т.е. поля created_at и updated_at. Они будут расширены новыми полями в будущих итерациях.

Кроме того, на первой итерации необходимо дать возможность пользователю удалять свой ответ и запретить гостю создавать вопросы и ответы к ним.

Стоит сразу определиться с ролями, которые будут в системе и какие действия будут им доступны. Всего в системе будет три роли: гость, администратор, который управляет всеми ресурсами, и обычный пользователь, управляющий только своими ресурсами. То есть у аутентифицированного пользователя есть права на следующие действия:

- редактирование и удаление своего вопроса;
- редактирование и удаление своего ответа;
- возможность задать вопрос;
- возможность добавить комментарий;
- редактирование и удаление своего комментария;
- возможность голосовать (ставить рейтинг) за чужие вопросы и ответы;
- если пользователь является автором вопроса, то он может выбирать лучший ответ;
- возможность подписаться на уведомления/отписаться от уведомлений о новых ответах;

- возможность прикреплять файлы к вопросам и ответам.

У гостя есть право на «чтение» ресурсов – просмотр главной страницы с вопросами, просмотр самого вопроса, комментариев и ответов к нему. Также он может осуществлять поиск в системе. Другие действия ему не доступны.

Диаграмма вариантов использования для трех ролей в системе представлена в приложении 3.

2.3 Планируемые итерации

Составим примерный список итераций с историями, которые нужно выполнить после базовой:

1. пользователь может редактировать свой вопрос и ответ. Данная функция должна быть реализована с помощью AJAX, т.е. используя асинхронный запрос к серверу без обновления страницы. Также должна быть возможность прикреплять файлы к вопросам и ответам, пользователь может удалить прикрепленный файл. Изменение файлов доступно при редактировании ресурса (вопроса или ответа).

На этой итерации также планируется выполнить реализации возможности выбора лучшего ответа автором вопроса, данная функция должна быть реализована с помощью AJAX без обновления страницы;

2. добавление вопросу и ответу рейтинга, который формируется в результате голосования пользователями. Каждый пользователь может проголосовать за ресурс лишь однажды, но при этом имеет возможность отменить свой голос для повторного голосования.

Также должно быть реализовано посредством AJAX;

3. добавить возможность комментировать вопросы и ответы. Появление новых вопросов, ответов и комментариев на странице без ее обновления посредством WebSockets;

4. добавить возможность аутентификации через социальные сети – Facebook, Twitter. Начать разработку REST API:

– получение пользователя владельца токена;

- получение всех пользователей кроме владельца;
- получение списка вопросов и ответов;
- получение вопроса и ответа отдельно;
- создание вопроса и ответа.

Таблица 1 – предполагаемый REST API

URL	Метод	Описание
Ресурс пользователей		
/me	GET	Возвращает владельца токена
/index	GET	Возвращает список всех пользователей кроме владельца токена
Ресурс вопросов		
/index	GET	Возвращает список всех вопросов
/show	GET	Возвращает конкретный вопрос
/create	POST	Создание вопроса
Ресурс ответов		
/index	GET	Возвращает список всех ответов
/show	GET	Возвращает конкретный ответ
/create	POST	Создание ответа

5. в системе должна быть реализована ежедневная рассылка email подписанным пользователям в виде дайджеста со списком новых вопросов в системе за день.

Пользователь может подписаться/отписаться на уведомления о новых ответах.

Пользователь может выполнить поиск в системе по ответам, вопросам, комментариям и пользователям.

В приведенных выше итерациях ничего не написано о написании тестов к приложению, предполагается, что все перечисленные истории будет протестированы с помощью модульных и интеграционных тестов.

В production окружении у реально развернутой на сервере системе также должно быть обновление индекса поиска и бэкап базы данных по расписанию. Эти действия планируется выполнить после последней итерации в процессе тестового развертывания системы.

3 Этапы реализации

3.1 Создание ресурса пользователей, ответов и вопросов

На первой итерации были созданы таблицы в БД для ресурса пользователей, ответов и вопросов. Создание таблиц и нужных полей в RoR происходит с помощью команд в терминале, которые имеет следующий вид:

```
rails generate(g) migration create_questions title body
```

В результате будет сформирован следующий файл:

```
class CreateQuestions < ActiveRecord::Migration[5.1]
  def change
    create_table :questions do |t|
      t.string :title
      t.string :body
      t.timestamps
    end
  end
end
```

После этого необходимо выполнить команду: `rails db:migrate`. Данная миграция создаст таблицу `questions` с полями `title`, `body`, `created_at` и `updated_at`. Соответствующая миграция была выполнена и для создания таблицы `answers`.

Поскольку для аутентификации используется библиотека (гем) `Devise`, то миграция создается самой библиотекой после ее установки. Для использования конкретных возможностей библиотеки в модели пользователя перечисляются нужные свойства – `devise :database_authenticatable, :registerable, :confirmable`.

После выполнения необходимых миграций создаются модели, где определяются ассоциации на уровне ORM, добавляются необходимые валидации и скоупы (статические методы модели) для получения данных из БД.

Например, для пользователя в модели прописываются следующие строки кода:

```
has_many :questions
has_many :answers
validates :username, presence: true, uniqueness: true, length: { minimum: 5 }
```

Первые две определяют ассоциации на уровне моделей и добавляют для каждого пользователя методы `user.answers` и `user.questions`, которые возвращают связанные с этим пользователем вопросы и ответы.

Третья строка создает валидацию (проверку) на уровне моделей при инициализации нового пользователя (регистрации), имя пользователя должно быть уникальным и иметь длину не менее 5 символов.

Для проверки наличия нужных ассоциаций в процессе проектирования пишутся тесты к моделям, которые располагаются в файле `app/сpec/models/model_name_spec.rb`. Например, чтоб проверить наличия ассоциации между пользователем и вопросов с ответами, используется следующие выражения:

```
it { should have_many(:questions) }
it { should have_many(:answers) }
```

Для теста валидации тела ответа:

```
it { should validate_length_of(:body).is_at_least(5) }
```

Стандартный контроллер состоит из CRUD операций, в Rails контроллере за это отвечает методы `index`, `new`, `create`, `edit`, `update` и `destroy`.

Метод `index` просто возвращает список всех вопросов: `@questions = Question.all`.

Для удаления идет проверка, является ли пользователь автором вопроса:

```
current_user.author_of?(@answer)
```

Сам метод написан в модели `User`, принимает в качестве аргумента объект и просто сравнивает `id` пользователя с `user_id` объекта — `id == item.user_id`.

Для тестирования контроллеров проверяется, влияет ли запрос с нужными параметрами на определенные свойства, например, число всех вопросов в системе:

```
expect { post :create, params: { question: attributes_for(:question) } }.to change(Question, :count).by(1)
```

Тестирование остальных методов происходит в похожей манере, при редактировании после PUT запроса идет проверка, равен ли новый заголовок в вопросе заданному — `expect(question.title).to eq 'new title'`.

Для запрета гостям производить любые действия кроме чтения с вопросами и ответами, на уровне представлений идет проверка, существует ли текущий пользователь, если такого пользователя нет, то возвращается nil и гость не видит ссылки для создания, редактирования и удаления объектов.

Таким образом, за первую итерацию были созданы нужные таблицы в БД, модели ресурсов, контроллеры для CRUD операций и представления. Добавлено ограничение для редактирования и удаления ресурса только автором и запрет гостю на выполнение любых операций кроме чтения. В дальнейшем логика, связанная с авторизацией пользователя, будет вынесена в отдельный класс, чьей ответственностью будет только разграничение прав.

3.2 Управление ресурсами через AJAX и прикрепление файлов

На второй итерации было необходимо сделать создание ответа, а также редактирование ответа и вопроса посредством AJAX. Добавить возможность прикреплять файлы к вопросам и ответам, а также дать автору возможность выбирать лучший ответ.

Теперь частичное представление для создания ответа имеет разрешение .js.erb, что позволяет вернуть пользователю ответ в формате JS, а не простом html. Представление содержит следующий код:

```
$('.answers-errors').html("<%= j render 'shared/errors', resource: @answer %>");
$('.answers').append("<%= j render @answer %>");
$('#answer_body').val('');
```

При создании нового ответа в случае возникновения ошибок, они вставляются в блок answers-errors, новый ответ добавляется в конец блока answers, а значение формы очищается.

Для тестирования правильного поведения системы со стороны пользователя в браузере пишутся специальные интеграционные (acceptance) тесты. Пишется название сценария, если действие происходит посредством ajax запроса, тогда нужно указать специальную опцию *js: true*, после этого выполняются необходимые действия и проверяется, совпадает ли результат с

ожидаемым. Пример интеграционного теста для создания ответа к вопросу аутентифицированным пользователем представлен ниже:

```
scenario 'Authenticated user adds answer to the question', js: true do
  sign_in(user)

  visit question_path(question)

  fill_in 'Body', with: 'My unique answer'
  click_on 'Answer'

  within '.answers' do
    expect(page).to have_content 'My unique answer'
  end
end
```

Такое тестирование не только позволяет тестировать поведение системы, но и показать заказчику, что все работает как нужно. Благодаря тому, что тесты написаны интуитивно понятным языком на английском, даже не имеющий к программированию никого отношения человек, сможет понять, к чему относится данный тест.

Редактирование вопросов и ответов также было реализовано через AJAX, точнее говоря, это называется AJAJ, поскольку отдается JavaScript, а не XML. В представлении ответа для ссылки редактирования было убрано какое-то либо действия по умолчанию, обработка нажатия теперь происходит на стороне клиента:

```
= link_to 'Edit', '', class: 'edit-answer-link', data: { answer_id: answer.id }
$(document).on('turbolinks:load', function() {
  $(document).on('click', ".edit-answer-link", function(e) {
    e.preventDefault();
    $(this).hide();
    var answer_id = $(this).data('answerId')
    $('#form#edit_answer_' + answer_id).show()
  })

  $(document).on('click', ".edit-question-link", function(e) {
    e.preventDefault();
    $(this).hide();
    $('#form.edit_question').show()
  })
})
```

Как видно из кода выше, теперь при нажатии на кнопку открывается форма, где вводятся новые значения пользователем. В представлении для метода

PUT обновления ответа, теперь идет проверка на существование ошибок и в зависимости от их наличия, выполняется нужное действие:

```
<% if @answer&.errors.present? %>
  $('#answer-update-errors-<%= @answer.id %>').html("<%= j render 'shared/errors',
resource: @answer %>");
<% else %>
  $('#.answer-<%= @answer.id %>').html('<%= j render @answer %>');
<% end %>
```

При написании интеграционного теста для редактирования ответа, описываются различные возможные сценарии и ожидаемый результат – обычный пользователь не видит ссылку на редактирования, автор видит ссылку и может редактировать, при вводе неверных значений должна появиться ошибка, гость также не видит ссылку. Ниже приведен сценарий для автора ответа:

```
describe 'Author user' do
  before do
    sign_in(answer.user)
    visit question_path(question)
  end

  scenario 'sees link edit' do
    within '.answers' do
      expect(page).to have_link 'Edit'
    end
  end

  scenario 'tries to edit his/her answer', js: true do
    click_on 'Edit'
    within '.answers' do
      fill_in 'Answer', with: 'edited answer'
      click_on 'Update Answer'

      expect(page).to_not have_content answer.body
      expect(page).to have_content 'edited answer'
      expect(page).to_not have_selector 'textarea'
    end
  end

  scenario 'invalid attributes for answer', js: true do
    click_on 'Edit'
    within '.answers' do
      fill_in 'Answer', with: ''
      click_on 'Update Answer'

      expect(page).to have_content answer.body
      expect(page).to have_content "Body can't be blank"
    end
  end
end
```

До сценария сначала производится вход в систему и переход на страницу вопроса. Затем проверяются три сценария, видит ли автор ссылку на

редактирования, само редактирование и сценарий ввода неверных атрибутов для ответа. Такого рода тесты были написаны для проверки редактирования и удаления ответов и вопросов.

После этого была реализована возможность выбора лучшего ответа автором вопроса. Для этого в таблицу БД Answer с помощью миграции было добавлено поле best типа Boolean со значением по умолчанию false.

Сам метод установки вопроса лучшим реализован в модели Answer, это уже относится к бизнес-логике системы и не имеет отношения к контроллеру, который реализует только сам PUT запрос на изменение ресурса.

```
def set_best!  
  Answer.transaction do  
    question.best_answer.update!(best: false) if question.has_best_answer?  
    self.update!(best: true)  
  end  
end
```

Тело метода заключено в транзакцию, поскольку происходит изменения сразу двух ресурсов, а не одного ответа. Сначала у старого лучшего ответ, если он есть, поле best задается false. Затем у текущего ответа best задается true.

В файле представления также отдается JavaScript в ответ на запрос, происходит рендеринг всех ответов на странице, после этого без перезагрузки новый лучший ответ отображается первым:

```
$('.answers').html('<%= j render @question.answers %>');
```

Код теста этого метода для случая, если пользователь – автор вопроса, представлен ниже:

```
describe 'Author of the question' do  
  before do  
    sign_in(question.user)  
    visit question_path(question)  
  end  
  
  scenario 'Sees choose best link' do  
    within ".answer_#{answer.id}" do  
      expect(page).to have_link 'Set best'  
    end  
  end  
  
  scenario 'Sets answer as best and displays first', js: true do  
    within ".answer_#{answer.reload.id}" do  
      click_on 'Set best'  
      expect(page).to_not have_link 'Set best'  
      expect(page).to have_css('svg.octicon-check', count: 1)  
    end  
  end  
end
```

```

    within ".answers" do
      expect(answer.body).to appear_before(first_answer.body)
    end
  end

  scenario 'Sets new answer as best', js: true do
    within ".answer_#{answer.reload.id}" do
      click_on 'Set best'
      expect(page).to_not have_link 'Set best'
      expect(page).to have_css('svg.octicon-check', count: 1)
    end

    within ".answer_#{previous_best_answer.reload.id}" do
      expect(page).to have_link 'Set best'
      expect(page).to_not have_css('svg.octicon-check', count: 1)
    end
  end
end
end

```

Как видно из теста, сначала происходит вход в систему и переход на страницу вопроса. Затем поочередно проверяется, видит ли автор ссылку для задания ответа лучшим, появляется ли ответ первым в списке при нажатии, и последний сценарий – проверка на новый лучший ответ, у старого должна пропасть иконка лучшего ответа, а новый соответственно должен ее получить, при этом ссылка ‘Set best’ у него не отображается.

В контроллере ответов также добавляется метод PUT set_best, который вызывает метод set_best! из модели Answer, до этого находится связанный с ответом вопрос, чтобы дополнительно проверить, является ли пользователь автором:

```

def set_best
  @question = @answer.question
  @answer.set_best! if current_user.author_of?(@question)
end

```

Следующим шагом являлась реализация добавление файлов к вопросам и ответам. Для этого использовалась библиотека carrierwave, которое позволяет добавлять файлы к ресурсам, настраивать их различным образом и многое другое.

В первую очередь создается отдельная модель для вложений, связь с ответами и вопросами у нее является полиморфной, т.к. файл может

принадлежать либо вопросу, либо ответу. Также в модели подключается загрузчик от библиотеки carrierwave:

```
class Attachment < ApplicationRecord
  belongs_to :attachable, polymorphic: true, optional: true

  mount_uploader :file, FileUploader
end
```

В самом загрузчике определяется папка, куда попадают прикрепленные файлы – "uploads/#{model.class.to_s.underscore}/#{mounted_as}/#{model.id}".

Для вложений также были созданы соответствующие представления, контроллер и тесты к ним. Например, контроллер содержит только метод для удаления вложения:

```
class AttachmentsController < ApplicationController
  before_action :authenticate_user!

  def destroy
    @attachment = Attachment.find(params[:id])
    @resource = @attachment.attachable
    @attachment.destroy if current_user.author_of?(@resource)
  end
end
```

А само добавления реализовано через nested forms и контроллеры вопросов и ответов, где в разрешенные параметры запроса добавляется файл – `params.require(:question).permit(:title, :body, attachments_attributes: [:file])`.

В итоге за вторую итерацию было реализовано управление вопросами и ответами через AJAX, возможность прикрепления файлов к ним и возможность выбора лучшего ответа автором вопроса.

3.3 Оценивание вопросов и ответов

На третьей итерации было необходимо добавить пользователю возможность голосовать за вопрос или ответ: ставить «понравится» и «не понравится». При этом пользователь может голосовать только один раз, но имеет возможность отменить свой голос и переголосовать.

Поскольку и ресурс ответов, и ресурс вопросов, может иметь рейтинг и методы реализации его одинаковы, то для соблюдения принципа DRY (Don't

repeat yourself), т.е. для исключения повторения кода, используется расширения модуля ActiveSupport – Concerns. Концерны представляют собой модуль Ruby, который расширен модулем ActiveSupport::Concerns. Концерны можно использовать как для моделей, так и для контроллеров. В тестировании для соблюдения DRY используют так называемые «shared examples», они также просто подключаются в файле теста, не дублируя сами методы.

Основная логика расположена в модуле концернов Votable, тело метода обернуто в транзакции, поскольку изменения происходят не только в объекте вопроса/ответа, но и создается (удаляется) связанный с ним голос. Например, код метода увеличение рейтинга (голосования «за») представлен ниже:

```
def rating_up!(user)
  return false unless votable?(user)

  self.transaction do
    self.votes.create!(user: user, value: 1)
    self.rating += 1
    self.save!
  end
end
```

В первой строке метода проверяется, является ли ресурс доступным для голосования, т.е. голосовал ли уже за него пользователь или нет. Эта проверка не входит в ответственность метода, но на данном этапе разработки она сделана здесь, в дальнейшем для этого будет использован отдельный класс, связанный с авторизацией.

Концерн контроллера содержит три публичных PUT метода для повышения рейтинга, понижения и отмены голоса. Ответ пользователю в них отдается в формате JSON, который обрабатывается на стороне клиента и в зависимости от результата динамически формирует содержание страницы. Пример метода повышения рейтинга для контроллера представлен ниже:

```
def rating_up
  respond_to do |format|
    if @votable.rating_up!(current_user)
      format.json { render json: @votable }
    else
      format.json { render json: @votable.id, status: :forbidden }
    end
  end
end
```

```
end
end
end
```

Подключение концернов к моделям и контроллерам происходит с помощью ключевого слова `include`.

Обработка запроса на стороне клиента выполнена с помощью CoffeeScript, в зависимости от ответа, выполняются различные действия. `Ajax:success` выполняется в случае удачного завершения запроса к серверу, `ajax:error` – в случае неудачного. Например, обработка голосования происходит следующим образом:

```
vote_question = ->
  $('.vote_question').bind 'ajax:success', (e) ->
    question = $.parseJSON(e.detail[2].responseText)
    $('.question_rating').html(question.rating)
  .bind 'ajax:error', (e) ->
    $('.vote_error_question').html("You have already voted!")
```

Если ответ от сервера приходит удачный, т.е. в основном 2xx код, тогда из полученного ответа формата JSON получается рейтинг и вставляется в соответствующий html элемент. Если же неудачный – выводится информационное сообщение.

Рассмотрим один из тестов (shared examples) для концерна контроллеров:

```
shared_examples_for 'voted' do
  let(:controller) { described_class }
  let(:votable) {
create(controller.to_s.underscore.split('_')[0].singularize.to_sym) }
  let!(:user) { create(:user) }

  before { sign_in(user) }

  describe 'PATCH #rating_up' do
    before { patch :rating_up, params: { id: votable.id, format: :json } }

    it 'gets json response' do
      expect(response.header['Content-Type']).to include 'application/json'
      expect(response.body).to eq (assigns(:votable)).to_json
    end

    it 'returns id and forbidden status if rating_up! returns false' do
      patch :rating_up, params: { id: votable.id, format: :json }
      expect(response.header['Content-Type']).to include 'application/json'
      expect(response.body).to eq assigns(:votable).id.to_s
      expect(response.status).to eq 403
    end
  end
end
```

Переменная `votable` получается динамически в зависимости от того, для какого контроллера происходит тестирование, т.е. либо `AnswersController` или `QuestionsController` в данном случае. Значение контроллера получается с помощью метода `described_class`, который возвращает класс (контроллера).

В тесте сначала происходит вход пользователя в систему, поскольку только аутентифицированному пользователю разрешено выполнение запросов к серверу. После этого проверяется правильное поведение запроса в случае удачного и неудачного его выполнения.

Затем в нужных файлах для тестирования контроллеров просто прописывается строка `it_behaves_like 'voted'`.

Также, как и на всех итерациях были написаны интеграционные тесты, тестирующие поведения системы с точки зрения системы, но их описание исключено из исследовательской работы ввиду большого количества.

Таким образом, на третьей итерации была добавлена возможность голосования пользователя за вопросы и ответы, а также возможность отмены своего голоса и повторного голосования.

3.4 Комментарии и динамическое обновление страницы через WebSockets

На четвертой итерации нужно было добавить пользователю возможность комментировать вопросы и ответы, а также появление новых вопросов, ответов и комментариев без обновления страницы посредством технологии `WebSockets`.

Таблица комментария состоит из тела, внешнего ключа для связи с пользователем и полиморфной связью с комментируемым ресурсом: два специальных поля в таблице `commentable_id`, `commentable_type`. В самой модели комментария с помощью `belongs_to` и опции `polymorphic: true` обозначается полиморфная связь с другими моделями:

```
belongs_to :commentable, polymorphic: true, optional: true
```

Комментарии имеют собственный контроллер, который содержит три основных метода – POST create, PUT update и DELETE destroy. Их отправка также осуществляется через AJAX.

Все методы и пользовательский интерфейс, связанный с комментариями, был протестирован с помощью юнит и интеграционных тестов. Тесты не слишком отличаются от уже представленных ранее в этой главе, поэтому конкретных примеров по тестированию комментариев здесь не показано.

После этого было реализовано динамическое появление комментариев, ответов и вопросов на странице с помощью WebSockets.

Веб-сокеты (Web Sockets) – это передовая технология, которая позволяет создавать интерактивное соединение между клиентом (браузером) и сервером для обмена сообщениями в режиме реального времени. Веб-сокеты, в отличие от HTTP, позволяют работать с двунаправленным потоком данных, что делает эту технологию совершенно уникальной [7].

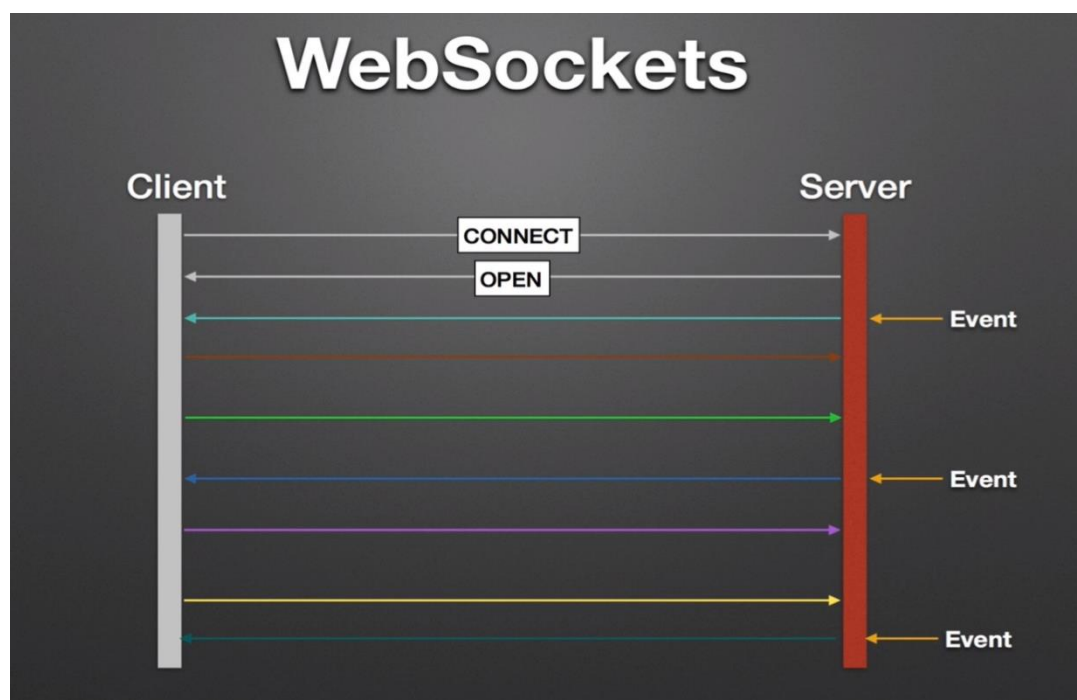


Рисунок 3 – технология WebSockets

В RoR за серверную часть работы с WebSockets отвечает модуль ActionCable, который работает по технологии Pub/Sub, т.е. издатель-подписчик.

На стороне фреймворка создается канал и происходит вещание какого-то события, при этом клиент подписывается на поток при заходе на страницу, например, при заходе на страницу определенного вопроса, пользователь становится подписчиком канала и определенного потока для новых комментариев и новых ответов вопроса. Соединение остается все время открытым, пинг происходит каждую секунду, чтобы проверить доступность канала [7]. При срабатывании какого-то события, происходит вещания со стороны сервера, сообщение обрабатывается на клиентской стороне и преобразуется в вид, доступный для пользователя.

Например, название потока для новых ответов вопроса формируется следующим образом – "question:#{data['question_id'].to_i}:answers".

Намного интереснее происходит создание подписки на стороне клиента с помощью JS и CoffeeScript:

```
$(document).on('turbolinks:load', ->
  return followQuestion() if App.answers_subscribe

App.answers_subscribe = App.cable.subscriptions.create('AnswersChannel', {
  collection: -> $('.answers')

  connected: ->
    @followQuestion()

  followQuestion: ->
    questionId = @collection().data('questionId')
    if questionId
      App.answers_subscribe.perform 'follow', question_id: questionId
    else
      App.answers_subscribe.perform 'unfollow'

  received: (data) ->
    @collection().append JST["templates/answer"](answer:
$.parseJSON(data['answer']))
  })))
```

Для канала AnswersChannel создается подписка, при соединении выполняется метод followQuestion, в котором проверяется существует ли 'questionId' у коллекции, если да, тогда происходит подписка к определенному потоку. При получении события от потока в блок 'answers' вставляется обработанное с помощью специального шаблонизатора JST информация, событие

приходит в формате JSON и представляет собой объект ответа: тело, автора, дату создания.

В данном случае интересными являются интеграционные тесты для такого интерактивного поведения, поскольку нужно проверить это стороны двух пользователей или в случае тестов сессий, одна гостевая, другая – аутентифицированного пользователя. Библиотеки для тестирования в RoR легко позволяют это сделать:

```
context 'multiple sessions' do
  scenario "comment appears on another user's page", js: true do
    Capybara.using_session('user') do
      sign_in(user)
      visit question_path(question)
    end

    Capybara.using_session('guest') do
      visit question_path(question)
    end

    Capybara.using_session('user') do
      within ".question" do
        click_on 'Add comment'
        fill_in 'create_comment_body', with: 'new comment'
        click_on 'Comment'

        expect(page).to have_content 'new comment'
      end
    end

    Capybara.using_session('guest') do
      within ".question" do
        expect(page).to have_content 'new comment'
      end
    end
  end
end
```

В тесте используются две сессии – ‘user’ и ‘guest’, с пользовательской происходит добавление комментария, на гостевой проверяется его наличие.

Таким образом, за четвертую итерации была добавлена возможность комментирования вопросов и ответов, а также интерактивное взаимодействие между клиентом и сервером – отображение новых вопросов, ответов и комментариев через ВебСокеты.

3.5 Аутентификация через социальные сети и REST API

На пятой итерации было необходимо добавить пользователям возможность аутентифицироваться через социальные сети Facebook и Twitter. Также нужно было разработать REST API для системы, который позволяет получать всех пользователей (или отдельного владельца токена), все вопросы (отдельный вопрос), все ответы (отдельный ответ), а также создавать вопросы и ответы.

Для аутентификации использовались библиотеки `omniauth`, `omniauth-facebook` и `omniauth-twitter`, которые используют протокол открытой аутентификации OAuth 2.0. Он предоставляет несколько вариантов для «поточков» – методов реализации, в данном случае используется Authorization Code Flow [8].

У пользователя может быть несколько авторизаций (хотя понятие авторизация не совсем подходит в данном случае), каждая авторизация состоит из названия провайдера (строка), идентификатора, поля `confirmed` (boolean) и внешнего ключа пользователя. Для входа в систему через социальную сеть был разработан следующий алгоритм:

- сначала происходит поиск по имеющимся авторизациям в системе, если найдена авторизация по провайдеру и `uid`, тогда возвращается связанный пользователь и происходит вход в систему.
- если такой авторизации нет в системе, тогда возможны два варианта в зависимости от того, возвращает соц. сеть `email` пользователя или нет:
 - при наличии `email` идет поиск пользователя в системе, если такой пользователь существует, то создается связанная с ним авторизация и осуществляется вход в систему. При этом пользователю сначала необходимо подтвердить авторизацию, письмо с соответствующей ссылкой будет отправлено ему на почту.

Если пользователя с таким адресом почты нет в системе, тогда создается новый пользователь вместе с авторизацией. Для последующего входа в систему пользователю также сначала нужно подтвердить адрес электронной почты;

- когда социальная сеть не возвращает адрес почты пользователя, тогда происходит переадресация пользователя на страницу для ввода своего email, после этого повторяются предыдущие шаги: либо создается авторизация для имеющегося пользователя, либо создается новый пользователь вместе с авторизацией. Пользователю также необходимо подтвердить адрес своей почты перед входом в систему.

Ко всем связанным с аутентификацией методам моделей и контроллеров также были написаны модульные и интеграционные тесты.

Кроме того, на пятой итерации вся логика, связанная с авторизацией (правами доступа пользователя) была вынесена из моделей и контроллеров в отдельный класс. Для этого использовалась специальная библиотека CanCanCan. Создается специальный класс Ability, в котором определяется, что может делать конкретная группа пользователей, например, для пользователя были определены следующие возможности:

```
can :read, :all
can :create, [Question, Answer, Comment]
can [:update, :destroy], [Question, Answer, Comment], user: user

can :destroy, Attachment, attachable: { user: user }

can [:set_best], Answer, best: false, question: { user: user }

can [:rating_up, :rating_down], Votable do |votable|
  votable.user != user && votable.votes.where(user: user).blank?
end

can [:cancel_vote], Votable do |votable|
  votable.votes.where(user: user).present?
end
```

Сначала обозначается метод и объект метода, затем либо через запятую, либо в блоке пишется условие, при котором действие может быть выполнено.

Например, пользователь может выбрать ответ лучшим, если он не является лучшим на данный момент и у вопроса связанный объект пользователя совпадает с текущим пользователем.

В представлениях для отображения ссылок только авторизованным пользователям используется специальный метод `if can?` Действие, Объект – `if can? :create, Comment`.

В таблицу `Users` было добавлено логическое поле `admin` для разграничения прав между обычным пользователем и администратором. Администратор может выполнять все действия – `can :manage, :all`.

Для создания в системе собственной REST API использовалась библиотека `doorkeeper`, которое реализует `Authorization Code Flow` между системой и внешними приложениями. В БД были добавлены нужные для работы с библиотекой таблицы.

Контроллеры для REST API находятся в отдельной папке `API/V1`, т.е. первая версия API. Все данные отдается в формате `json`. Для их сериализации используется библиотека `ActiveModelSerializer`, там прописываются поля, которые должны быть включены в ответ и связанные с ними объекты, например для ответа:

```
class SeparateAnswerSerializer < ActiveModel::Serializer
  attributes :id, :body, :created_at, :updated_at, :question_id
  has_many :comments
  has_many :attachments, serializer: AttachmentSerializer
end
```

В результате ответ содержит поля `id`, `body`, `created_at`, `updated_at` и `question_id`. А также связанные с ним комментарии и вложения.

Для проверки корректной работы REST API также были написаны юнит тесты к контроллерам, например, тест для проверки работы создания ответов авторизованным на это пользователем через REST API выглядит так:

```
describe 'POST /create' do
  context 'authorized' do
    let(:access_token) { create(:access_token) }
    let!(:question) { create(:question) }
    let(:user) { User.find(access_token.resource_owner_id) }
  end
end
```

```

before { post "/api/v1/questions/#{question.id}/answers", params: { format:
:json, answer: attributes_for(:answer), access_token: access_token.token } }

it 'returns 201 status' do
  expect(response.status).to eq 201
end

%w(id body created_at updated_at).each do |attr|
  it "contains #{attr}" do
    expect(response.body).to
be_json_eq(question.answers.last.send(attr.to_sym).to_json).at_path("answer/#{attr}")
  end
end
end
end
end

```

Проверяется, что ответ имеет нужный статус 201 Created – создан и что ответ пришел в правильном формате, содержит необходимые атрибуты, которые прописаны в сериалайзере и при этом значения полей равны созданным в БД.

В итоге перечень REST API содержит все методы, перечисленные в Таблице 2.

Таблица 2. REST API системы

URL	Метод	Описание
Ресурс пользователей		
/me	GET	Возвращает владельца токена
/index	GET	Возвращает список всех пользователей кроме владельца токена
Ресурс вопросов		
/index	GET	Возвращает список всех вопросов
/show	GET	Возвращает конкретный вопрос
/create	POST	Создание вопроса
Ресурс ответов		
/index	GET	Возвращает список всех ответов
/show	GET	Возвращает конкретный ответ
/create	POST	Создание ответа

В результате за пятую итерацию была добавлена возможность входа через Facebook и Twitter, реализована начальная и единственная на данный момент версия REST API. Также вся логика, связанная с авторизацией, была вынесена в отдельную абстракцию, которая за это отвечает.

3.6 Поиск и фоновые задачи в системе

На шестой итерации было необходимо добавить возможность поиска по вопросам, ответам, комментариям и пользователям в системе. Реализовать функционал подписок на новые ответы к вопросам, а также ежедневную рассылку дайджеста новых вопросов в системе.

Для выполнения фоновых задач в системе используется модуль RoR ActiveJob и один из адаптеров на усмотрение разработчика, в данном случае использовалась библиотека Sidekiq. Для формирования задач по расписанию используется библиотека whenever, которая добавляет задачи в crontab системы.

В базу данных была добавлена таблица subscriptions, которая состоит из полей id, user_id и question_id. Пользователь может подписаться на вопрос только один раз, поэтому был добавлен составной индекс по полям user_id и question_id с ограничением уникальности — `add_index :subscriptions, [:question_id, :user_id], unique: true`.

Был добавлен контроллер для управления подписками с двумя методами для создания и удаления подписок, авторизация пользователей также выполнена в классе Ability.

После добавления нового ответа, вызывается обратный вызов, в которой идет выполнение фоновой задачи — `NewAnswerEmailDistributionJob.perform_later(self.question, self)`.

`Perform_later` означает, что задача поставлена в очередь, а не исполняется в данный момент. В самой фоновой задаче происходит поиск всех подписчиков вопроса и для каждого выполняется отправка письма по почте в порядке очереди:

```
def perform(question, answer)
  question.subscribers.find_each do |subscriber|
    AnswerMailer.distribution(subscriber, answer).try(:deliver_later)
  end
end
```

Для рассылки ежедневного дайджеста используется библиотека `whenever`, в специальный файл расписания помещается задача и время, когда она должна быть исполнена:

```
every 1.days, at: '4:00 am' do
  runner "DailyDigestJob.perform_now"
end
```

В данном случае происходит выполнение фоновой задачи `DailyDigestJob` каждый день в 4 утра. В самой задаче для каждого пользователя системы происходит исполнение метода `DailyMailer digest` – `DailyMailer.digest(user).try(:deliver_later)`.

В этом методе происходит поиск всех вопросов, созданных за сегодня, затем список этих вопросов в пригодном для пользователя виде отправляется ему на почту.

Все методы, связанные с отправкой сообщений, также были протестированы с помощью интеграционных тестов, которые тестирует, как и отправку сообщений, так и их просмотр со стороны пользователя.

Поиск в системе был реализован с помощью движка полнотекстового поиска `SphinxEngine` и библиотеке для Rails `ThinkingSphinx`. Полнотекстовый поиск – автоматизированный поиск документов, при котором поиск ведётся не по именам документов, а по их содержимому, всему или существенной части [9].

Для каждого из документов (ресурсов) составляется свой индекс, в котором помещается, что нужно индексировать, т.е. по каким полям будет вестись поиск в индексе.

Поиск может быть произведен как по всем документам, так и по определенным: вопросам, ответам, комментариям и пользователям. Логика, связанная с поиска вынесена в отдельный класс в папке `services`. При каждом новом поиске создается новый объект класса `Search`, который содержит в себе объект поиска и тело поиска (запрос). В зависимости от объекта выполняется разный поиск.

Также для поиска было применено фрагментарное кеширование, которое позволяет выдавать пользователю страницу результатов поиска, сохраненную в кеше, при условии, что она является актуальной:

```
def search_cache(object, body, results)
  size = results.size
  max_updated_at = results.max_by(&:updated_at).updated_at.try(:utc).try(:to_s)
  "#{object}/#{body}-#{size}-#{max_updated_at}"
end
```

Для этого применяется особый ключ (идентификатор), который состоит из объекта поиска, тела, размера выдаваемой коллекции результата и времени последнего обновленного объекта.

Таким образом, за шестую итерацию была добавлена возможность подписки на уведомления о новых ответах к вопросам, ежедневная рассылка списка новых вопросов в системе за день и поиск в системе по всем объектам или только вопросам, ответам, комментариям и пользователям.

3.7 Тестирование системы в production окружении

Для тестирования работы в production окружении она была размещена на AWS (Amazon Web Services). Деплой приложения выполнялся с ручной настройкой, используя Capistrano. В качестве веб-сервера был выбран Nginx, а в качестве сервера приложения Unicorn.

В реальном окружении было необходимо решить несколько задач:

- обновление индекса поиска по расписанию;
- бэкап БД по расписанию;
- настройка сервера для отправки электронных писем;
- мониторинг самого приложения и различных программ, которое оно использует.

Обновление индекса и бэкап БД по расписанию был выполнен также с помощью библиотеки `whenever`, как было показано в разделе 3.6. Для отправки писем был использован агент пересылки сообщений `Exim`.

Для мониторинга была использована программа `monit`, которая предоставляет понятный веб-интерфейс для мониторинга программ.

4 Результаты работы

Разработанная система формата «Вопрос-ответ», представляющая собой Ruby on Rails веб-приложение, позволяет пользователю задавать вопрос для решения какой-либо проблемы и получать на него ответы других пользователей. Также имеется возможность комментировать ответы и вопросы.

Каждый пользователь может голосовать за вопросы и ответы, а автор выбирать лучший ответ. Новые ответы, вопросы и комментарии появляются на странице без обновления с помощью технологии WebSockets.

Для системы был разработан REST API функционал, который позволяет использовать систему в качестве сервиса и в дальнейшем при его доработке возможно создание мобильных приложений.

Пользователь имеет возможность подписаться (отписаться) на уведомления о новых ответах на интересующий его вопрос. В системе производится ежедневная рассылка новых вопросов за день всем зарегистрированным пользователям.

Поиск в системе был реализован с помощью двигателя (системы) полнотекстового поиска Sphinx.

Пользовательский интерфейс и примеры работы системы показаны на рисунках 4-9.

Итоговая схема БД представлена в приложении Ж.

Диаграмма классов представлена в приложении И.

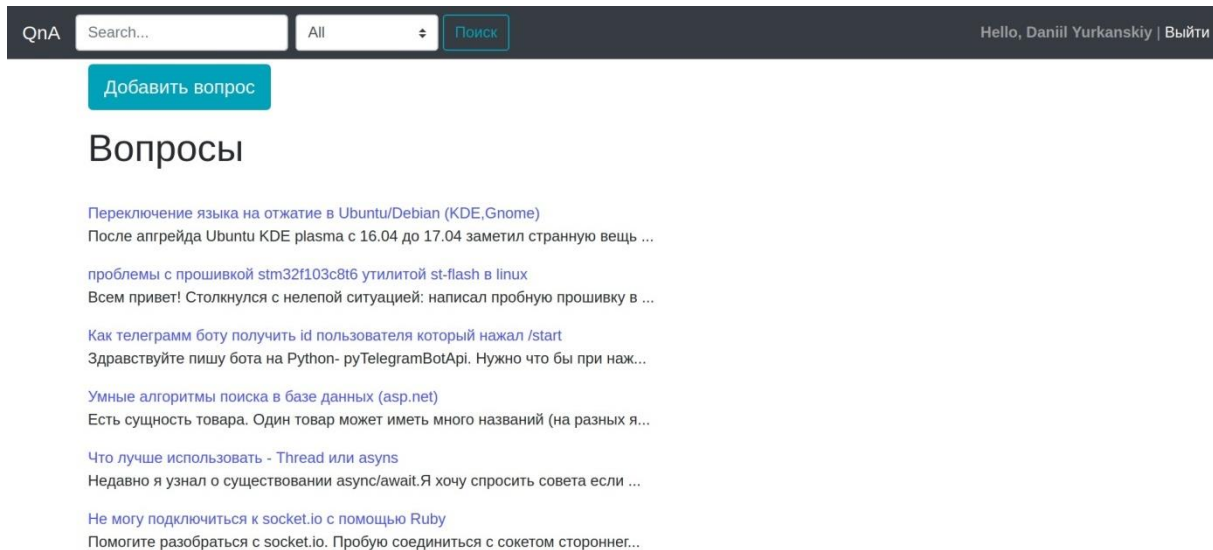


Рисунок 4 – Главная страница приложения

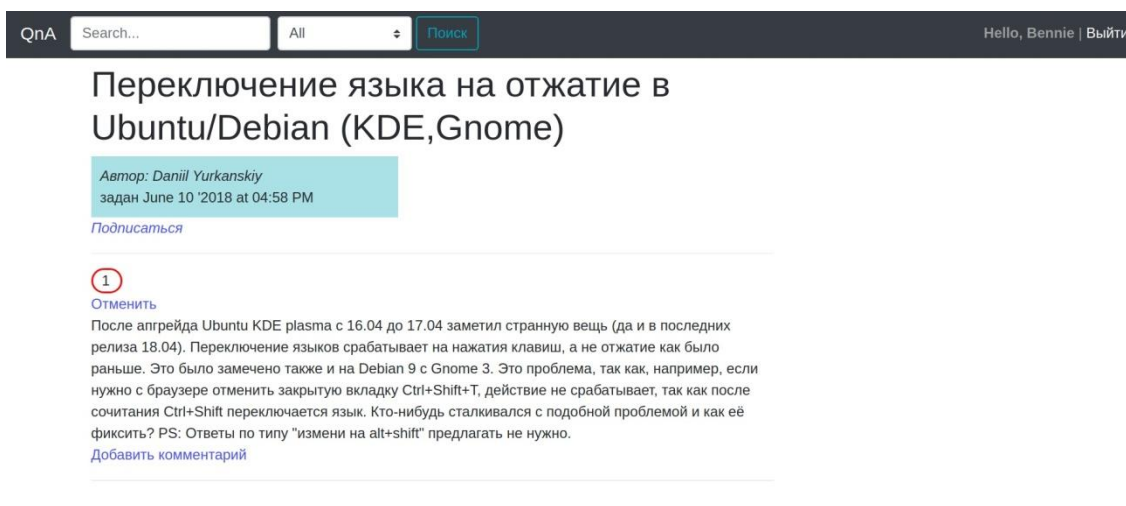


Рисунок 5 – Страница вопроса

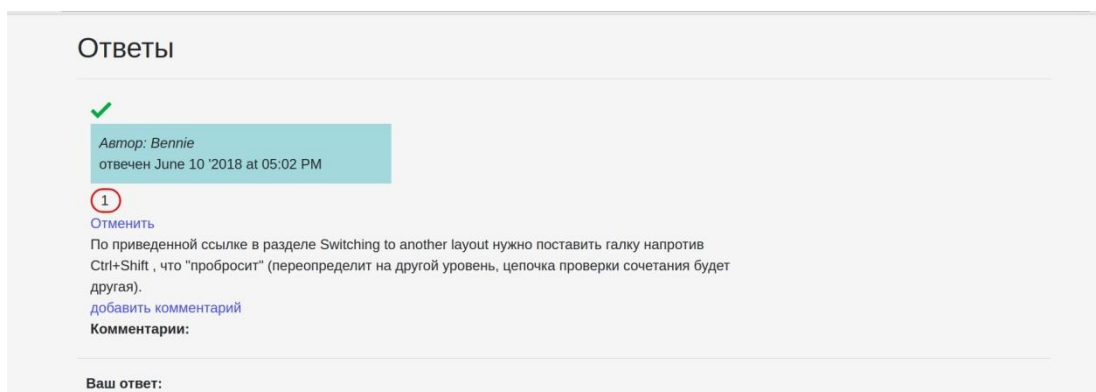


Рисунок 6 – Выбор лучшего ответа

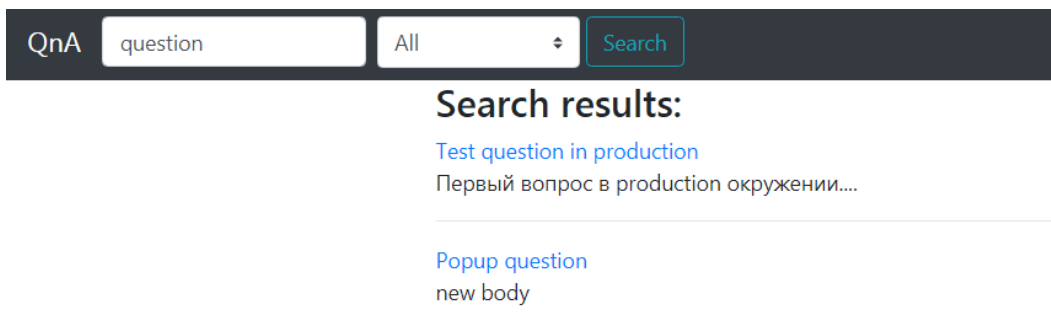
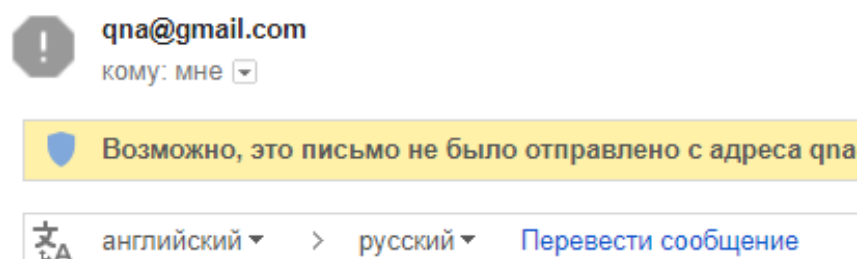


Рисунок 7 – Поиск на сайте в production



Hi, DanielYouCan!

Here is the daily digest for 2018-03-30 :

- [Почему...](#)

Рисунок 8 – Пример полученного дайджеста

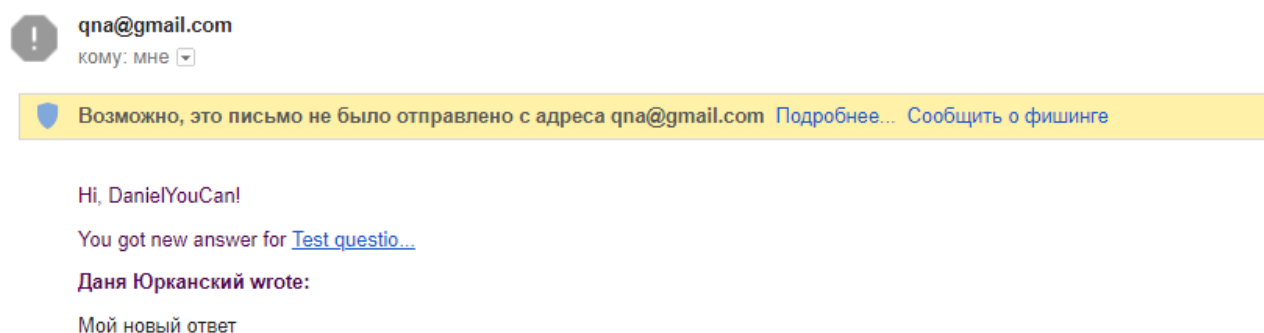


Рисунок 9 – Уведомление о новом ответе к вопросу

5 Финансовый менеджмент.

5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

5.1.1 Потенциальные потребители результатов исследования

Результаты исследования могут быть использованы многими категориями потребителей: веб-мастерами, рекламодателями, владельцами сайтов, коммерческими организациями и любыми физическими лицами.

5.1.2 Анализ конкурентных технических решений

Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения позволяет провести оценку сравнительной эффективности научной разработки и определить направления для ее будущего повышения. Данный метод позволяет определить слабые и сильные стороны и вывить подходящую стратегию развития компании и предлагаемого продукта [10].

Данный анализ с помощью оценочной карты приведен в приложении А.

Критерии для сравнения и оценки ресурсоэффективности и ресурсосбережения, приведенные в таблице 1, состояются из выбранных определенных пунктов сравнения с учетом их технических и экономических критериев для каждого этапа разработки.

Позиция разработки и конкурентов оценивается по каждому критерию по пятибалльной шкале, где 1 – самая слабая позиция, а 5 – самая сильная. Веса показателей в сумме должны составлять 1.

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum V_i \times B_i ,$$

где **K** – конкурентоспособность научной разработки или конкурента;

V_i – вес показателя (в долях единицы);

B_i – балл i -го показателя.

Экспертная оценка основных технических и экономических характеристик конкурентных программных решений показывает, что разрабатываемая информационная система формата вопросов и ответов является конкурентоспособной по сравнению с представленными аналогами.

Основными недостатками конкурентных программных продуктов являются достаточно сложный пользовательский интерфейс и высокая цена.

5.1.3 Технология QuaD

Технология QuaD (QUality ADvisor) представляет собой гибкий инструмент измерения характеристик, описывающих качество новой разработки и ее перспективность на рынке и позволяющие принимать решение целесообразности вложения денежных средств в научно-исследовательский проект. Использование данной технологии позволяет повысить точность и достоверность результатов исследования рынка [10].

В соответствии с технологией QuaD каждый показатель оценивается по сто-балльной шкале, где 1 – особо слабая позиция, а 100 – самая сильная.

Веса выбранных критериев в сумме должны составлять 1 [11].

Оценочная карта по технологии QuaD приведена в приложении Б.

Оценка качества и перспективности по технологии QuaD определяется по формуле:

$$P_{\text{ср}} = \sum V_i \times B_i$$

где $P_{\text{ср}}$ – средневзвешенное значение показателя качества и перспективности научной разработки;

V_i – вес показателя (в долях единицы);

B_i – средневзвешенное значение i -го показателя.

Разработка информационной системы формата «Вопрос-ответ» является перспективной, так как значение итогового показателя оценки перспективности равно 78 по технологии QuaD.

5.1.4 SWOT-анализ

Для наглядного представления всех возможностей, угроз, отображения слабых и сильных сторон компании и продукта, эффективности использования ресурсов и работы предприятия наиболее удобно использовать методику сжатого анализа маркетинговой информации - SWOT-анализ. SWOT-анализ представляет собой комплексный анализ научно-исследовательского проекта. Его применяют для исследования внешней и внутренней среды проекта, и проводится в несколько этапов.

Первый этап заключается в описании сильных и слабых сторон проекта, в выявлении возможностей и угроз для реализации проекта, которые проявились или могут появиться в его внешней среде.

Таблица 3. Матрица SWOT

	Сильные стороны научно-исследовательского проекта: С1. Удобный и простой интерфейс С2. Хорошая масштабируемость С3. Экономия ресурсов	Слабые стороны научно-исследовательского проекта: Сл1. Простой дизайн Сл2. Отсутствие профессиональной команды разработчиков Сл3. Относительная простота реализации – использованные технологии
Возможности: В1. Быстрое развертывание В2. Возможность расширение проекта В3. Возможность внедрения в систему ТПУ		
Угрозы: У1. Отсутствие финансирования У2. Высокие требования пользователей У3. Сбои в работе сервера и программы		

В ходе второго этапа была построена интерактивная матрица проекта, представленная в приложении В. минимизировать ошибки, контролировать процесс создания и проверять работу системы в соответствии с поставленными требованиями.

В рамках третьего этапа должна быть составлена итоговая матрица SWOT-анализа, которая приводится в бакалаврской работе (таблица 4).

Таблица 4. SWOT анализ

	Сильные стороны научно-исследовательского проекта: С1. Удобный и простой интерфейс С2. Хорошая масштабируемость С3. Экономия ресурсов	Слабые стороны научно-исследовательского проекта: Сл1. Простой дизайн Сл2. Отсутствие профессиональной команды разработчиков Сл3. Относительная простота реализации – использованные технологии
Возможности: В1. Быстрое развертывание В2. Возможность расширение проекта В3. Возможность внедрения в систему ТПУ	1. Внедрение в систему ТПУ 2. Разработка приложения на мобильные устройства	1. Доработка пользовательского интерфейса 2. Взгляд со стороны профессионалов – устранение недостатков 3. Использование современных технологий для пользовательской части (фронтенда)
Угрозы: У1. Отсутствие финансирования У2. Высокие требования пользователей У3. Сбои в работе сервера и программы	1. Поиск финансирования 2. Улучшения согласно требованиям пользователей 3. Повышение отказоустойчивости	1. Внедрение платных услуг в системе 2. Развитие собственных решений

Таким образом, в результате SWOT-анализа были рассмотрены сильные и слабые стороны проекта, выявлены возможные направления дальнейшей разработки программного продукта и рассмотрены варианты минимизации влияния угроз, которые могут этому помешать. Для повышения эффективности работы приложения и увеличения приносимой им прибыли необходимо улучшить интерфейс приложения, ввести платные услуги, реализовать интеграцию в системе ТПУ, повысить отказоустойчивость и разработать приложения на мобильные устройства.

5.2 Планирование научно-исследовательских работ

5.2.1 Структура работ в рамках научного исследования

Планирование научно-исследовательских работ заключается в упорядочивании стадий процесса исследования, таких как определение тематики научной деятельности, сбор и структурирование информации, выбор концепции деятельности, оформление и изложение полученных результатов научного исследования и др. для повышения эффективности исследования и разработок в

целом. Для выполнения научных исследований была сформирована научная группа, состоящая из научного руководителя и студента.

Перечень этапов НИР и распределение исполнителей представлен в таблице 5.

Таблица 5. Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб.	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания	Студент, руководитель проекта
Выбор технологий для проведения исследования	2	Подбор и анализ материалов для предметной области	Студент
	3	Выработка концепции планируемой системы	Студент, руководитель проекта
	4	Выбор технологии для проведения исследования	Студент
	5	Календарное планирование работ	Студент, руководитель проекта
Разработка и тестирование информационной системы	6	Выбор среды разработки	Студент
	7	Выбор методологий разработки	Студент
	8	Первичное проектирование системы	Студент
	9	Разработка и тестирование системы	Студент
Развертывание и доработка системы	10	Развертывание системы на реальном сервере	Студент
	11	Отладка, доработка интерфейса	Студент
Оформление отчета по НИР (комплекта документации по ОКР)	12	Составление пояснительной записки	Студент, руководитель проекта

5.2.2 Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаях образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Для определения ожидаемого значения продолжительности работ $t_{ож}$ применяется вероятностный метод – метод двух оценок t_{min} и t_{max} . Исходными

данными являются следующие вероятностные оценки продолжительности каждой работы:

- оптимальная или минимальная (t_{\min}) оценка продолжительности работ при выгодном стечении обстоятельств;
- пессимистическая или максимальная (t_{\max}) оценка продолжительности работы в предположении, что она будет выполнена при наиболее неблагоприятных условиях.

На основании вероятностных оценок усредняют продолжительности работ (метод усреднения), и вероятностные графики рассматриваются как детерминированные, т.е. ожидаемые (средние) значения продолжительности работ $t_{ож}$, которые определяются по формуле:

$$t_{ожi} = \frac{3t_{мини} + 2t_{маxi}}{5}$$

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

$t_{мини}$ – минимально возможная трудоемкость выполнения заданной i -ой работы (оптимистическая оценка: в предположении наиболее благоприятного стечения обстоятельств), чел.-дн.;

$t_{маxi}$ – максимально возможная трудоемкость выполнения заданной i -ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.

По ожидаемой трудоемкости работ определяется продолжительность каждой работы в рабочих днях T_p , учитывая параллельное выполнение работ в группах. Продолжительность одной работы определяется по формуле:

$$T_{pi} = \frac{t_{ожi}}{Ч_i}$$

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

$t_{ожi}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.

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

В приложении Г представлены расчеты $t_{ожі}$ и T_{pi} для каждой из работ.

5.2.3 Разработка графика проведения научного исследования

Для наглядного изображения планирования научно-исследовательских работ наиболее удобным и доступным инструментом является диаграмма Ганта, представляющая собой отрезки, соответствующие длительности выполнения каждого этапа научного исследования. Диаграмма Ганта позволяет визуально оценить последовательность и длительность этапов научного исследования, а также длительность всего научного исследования.

Для удобства построения графика, длительность каждого из этапов работ из рабочих дней переведем в календарные дни. Для этого воспользуемся формулой:

$$T_{ki} = T_{pi} \times k_k$$

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

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

k_k – коэффициент календарности.

Коэффициент календарности определяется по формуле:

$$k_k = \frac{T_k}{T_k - T_{\text{вых}} - T_{\text{пр}}}$$

где T_k – количество календарных дней в году;

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

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

$$k_k = \frac{365}{365 - 118} = 1,5$$

Рассчитанные значения в календарных днях по каждой работе T_{ki} округлим до целого числа.

Результаты расчетов представлены в приложении Д для 3х вариантов разработки информационной системы с использованием наиболее распространенных сочетаний программных средств:

Исполнение 1 – Данный проект - «Информационная система формата "Вопрос- Ответ" (Q&A) с использованием Ruby on Rails».

Исполнение 2 – с использованием C# и Asp.net MVC и негибкой методологией.

Исполнение 3 – Java и Spring

По данным расчетам, программа будет разработана (длительность в календарных днях):

- в первом исполнении 122 дней;
- во втором исполнении 140 дней;
- в третьем исполнении 143 дней.

Следовательно, можно сделать вывод, что в первом исполнении работы будет выполнена быстрее.

На основании приложения Д построен календарный план-график, представленный в приложении Е.

5.3 Бюджет научно-технического исследования (НТИ)

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

- материальные затраты НТИ;
- расчет амортизации;
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- накладные расходы.

5.3.1 Расчет материальных затрат НТИ

Расчет материальных затрат осуществляется по формуле:

$$Z_M = (1 + k_T) \times \sum_{i=1}^m C_i \times N_{расxi}$$

где m – количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{расxi}$ – количество материальных ресурсов i -го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м² и т.д.);

C_i – цена приобретения единицы i -го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м² и т.д.);

k_T – коэффициент, учитывающий транспортно-заготовительные расходы.

Для разработки данного продукта необходимы следующие материальные ресурсы:

- ноутбук;
- принтер.

Расчет материальных затрат представлен в таблице 6.

Таблица 6. Материальные затраты

Наименование	Ед. изм.	Кол-во			Цена за ед., руб.			Затраты на материалы, руб		
		Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Ноутбук	шт.	1	1	1	35290	37700	39100	35290	37700	39100
Принтер		1	1	1	8200	8500	9000	8200	8500	9000
Итого:								43490	46200	48100

5.3.2 Основная заработная плата исполнителей темы

В настоящую статью включается основная заработная плата научных и инженерно-технических работников, рабочих макетных мастерских и опытных производств, непосредственно участвующих в выполнении работ по данной теме. Под заработной платой в соответствии с ТК РФ ст. 129 понимают вознаграждение

за труд в зависимости от квалификации работника, сложности, количества, качества и условий выполняемой работы, также компенсационные и стимулирующие выплаты.

Основная заработная плата рассчитывается по формуле:

$$C_{\text{осн/зп}} = \sum_{i=1}^n t_i \times C_{\text{зп}_i}$$

где n – количество видов работ;

t_i – затраты труда на выполнение i -го вида работ, в днях;

$C_{\text{зп}_i}$ – среднедневная заработная плата работника, выполняющего i -ый вид работы, руб/день.

Среднедневная заработная плата рассчитывается по формуле:

$$C_{\text{зп}_i} = \frac{D \times K \times M_p}{F_0}$$

где D – месячный должностной оклад работника;

K – коэффициент, учитывающий коэффициент по премиям и районный коэффициент ($K=1,3$);

M_p – количество месяцев работы без отпуска в течение года;

F_0 – действительный годовой фонд рабочего времени работника, в днях.

При отпуске 28 дней $M_p=11,08$.

$$\text{Руководитель: } C_{\text{зп}} = \frac{25000 \times 1,3 \times 11,08}{251} = 1434,6$$

$$\text{Студент: } C_{\text{зп}} = \frac{4200 \times 1,3 \times 11,08}{251} = 241$$

Результаты расчета действительного годового фонда проведены в таблице

7.

Таблица 7. Годовой фонд рабочего времени

Показатели рабочего времени, дни	Исполнение 1	Исполнение 2	Исполнение 3
Календарное число дней в году	365	365	365
Количество нерабочих дней Выходные	104	104	104

Праздники (фактически по каждому году)	13	13	13
Планируемые потери отпуска	28	28	28
Действительный годовой фонд	251	251	251

Расчет затрат на основную заработную плату приведен в таблице 8. При этом затраты на оплату труда первого студента-дипломника определяются как оклад студента 3230,76 руб., а оклад руководителя проекта (ассистент) составляет 19230,76. Коэффициент К, учитывающий коэффициент по премиям и районный коэффициент равен 1,3, а M_p равно 11,08.

Таблица 8. Расчет основной заработной платы

Исполнители	Среднедневная заработная плата $C_{зн}$ (руб.)			Трудоемкость (t _i), чел-дни			Затраты на основную зарплату (руб.)		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Руководитель	1434,6			10	11	11	14346	15780,6	15780,6
Студент	241			81	93	95	19521	22413	22895
Итого							33867	38193,6	38675,6

5.3.3 Дополнительная заработная плата

Дополнительная заработная плата включает заработную плату за не отработанное рабочее время, но гарантированную действующим законодательством.

Расчет дополнительной заработной платы ведется по формуле:

$$Z_{\text{доп}} = k_{\text{доп}} \times Z_{\text{осн}}$$

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

$k_{\text{доп}}$ равен 0,15. Результаты по расчетам дополнительной заработной платы сведены в таблицу 9.

Таблица 9. Затраты на дополнительную заработную плату

Исполнители	Основная зарплата(руб.)			Коэффициент доп. заработной платы ($k_{доп}$)	Дополнительная зарплата(руб.)		
	Исп.1	Исп.2	Исп.3		Исп.1	Исп.2	Исп.3
Руководитель	14346	15780,6	15780,6	0,15	2151,9	2367,09	2367,09
Студент	19521	22413	22895		2928,15	3361,95	3434,25
Итого					5080,05	5729,04	5801,34

5.3.4 Отчисления во внебюджетные фонды (страховые отчисления)

В данной статье расходов отображены обязательные отчисления в соответствии с установленными законодательством Российской Федерации нормами органов 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 формулы:

$$Z_{внеб} = k_{внеб} \times (Z_{осн} + Z_{доп})$$

где $k_{внеб}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

На 2018 г. в соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30%. На основании пункта 1 ст.58 закона №212-ФЗ для учреждений осуществляющих образовательную и научную деятельность в 2014 году водится пониженная ставка – 27,1%.

Отчисления во внебюджетные фонды представлены в таблице 10.

Таблица 10. Отчисления во внебюджетные фонды

Исполнители	Основная зарплата(руб.)			Дополнительная зарплата(руб.)		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Руководитель	14346	15780,6	15780,6	2151,9	2367,09	2367,09
Студент	19521	22413	22895	2928,15	3361,95	3434,25
Коэффициент отчислений во внебюджетные фонды	30%					
Итого						
Исполнение 1	11684,1					
Исполнение 2	13176,8					
Исполнение 3	13343,1					

5.3.5 Накладные расходы

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

$$Z_{\text{накл}} = (\text{сумма статей } 1 \div 7) \times k_{\text{нр}}$$

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

Величина коэффициента накладных расходов равна 50%.

$$\text{Исполнение 1: } \frac{43490}{7} \times 0,5 = 3106,4$$

$$\text{Исполнение 2: } \frac{46200}{7} \times 0,5 = 3300$$

$$\text{Исполнение 3: } \frac{48100}{7} \times 0,5 = 3435,7$$

5.3.6 Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно-исследовательской работы (темы) является основой для формирования бюджета затрат проекта, который при

формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку научно-технической продукции.

Определение бюджета затрат на научно-исследовательский проект по каждому варианту исполнения приведен в таблице 11.

Таблица 11. Расчет бюджета затрат НИИ

Наименование статьи	Сумма, руб		
	Исп.1	Исп.2	Исп.3
Материальные затраты НИИ	43490	46200	48100
Затраты по основной заработной плате исполнителей темы	33867	38193,6	38675,6
Затраты по дополнительной заработной плате исполнителей темы	5080,05	5729,04	5801,34
Отчисления во внебюджетные фонды	11684,1	13176,8	13343,1
Накладные расходы	3106,4	3300	3435,7
Бюджет затрат НИИ	97227,55	106599,44	109355,74

Вывод: Таким образом бюджет затрат на научно – исследовательский проект по каждому варианту исполнения составляет: 97227,55; 106599,44; 109355,74 рублей. Наиболее оптимальным является вариант первого исполнения – 97227,55 рублей.

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

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

Интегральный финансовый показатель разработки определяется по формуле:

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

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

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

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

Расчет:

$$I_{\text{финр}}^{\text{исп.1}} = \frac{97227,55}{109355,74} = 0,89$$

$$I_{\text{финр}}^{\text{исп.2}} = \frac{106599,44}{109355,74} = 0,975$$

$$I_{\text{финр}}^{\text{исп.3}} = \frac{109355,74}{109355,74} = 1$$

Таким образом полученная величина интегрального финансового показателя разработки отражает соответствующее численное удешевление стоимости разработки в размах (значение меньше единицы, но больше нуля).

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

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

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

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i – балльная оценка i -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания.

Расчет интегрального показателя ресурсоэффективности приведен в таблице 12

Таблица 12. Сравнительная оценка характеристик вариантов исполнения проекта

Объект исследования Критерии	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
---------------------------------	-------------------------------------	-------	-------	-------

Стабильность системы	0,2	4	4	5
Удобство в эксплуатации (соответствует требованиям потребителей)	0,2	5	3	4
Производительность	0,2	5	3	3
Энергосбережение	0,1	4	3	2
Надежность	0,2	4	3	4
Эффективность	0,1	5	4	3
Итого:	1	4,5	3,3	3,5

После подсчета интегрального показателя ресурсоэффективности, можно сказать, что «Исполнитель 1» справляется лучше конкурентов следуя критериям по реализации ИС, его ресурсоэффективность составляет 4,5.

Интегральный показатель эффективности вариантов исполнения разработки ($I_{испi}$):

$$I_{исп1} = \frac{I_{p-исп1}}{I_{финр}^{исп1}} = \frac{4,5}{0,89} = 5,06$$

$$I_{исп2} = \frac{I_{p-исп2}}{I_{финр}^{исп2}} = \frac{3,3}{0,975} = 3,38$$

$$I_{исп3} = \frac{I_{p-исп3}}{I_{финр}^{исп3}} = \frac{3,5}{1} = 3,5$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволяет определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных. Сравнительная эффективность проекта ($\mathcal{E}_{ср}$) рассчитывается по формуле:

$$\mathcal{E}_{ср} = \frac{I_{исп1}}{I_{исп2}}$$

Сравнительная эффективность разработки, представлена в таблице 13.

Таблица 13. Сравнительная эффективность разработки

Показатели	Исп.1	Исп.2	Исп.3
Интегральный финансовый показатель разработки	0,89	0,975	1

Интегральный показатель ресурсоэффективности разработки	4,5	3,3	3,5
Интегральный показатель эффективности	5,06	3,38	3,5
Сравнительная эффективность вариантов исполнения	1,5	1	1,03

Сравнение значений интегральных показателей эффективности позволяет понять и выбрать более эффективный вариант решения поставленной в бакалаврской работе технической задачи с позиции финансовой и ресурсной эффективности. Наиболее целесообразным вариантом из предложенных является вариант 1.

Вывод: в ходе выполнения главы, касающейся финансовый менеджмент, ресурсосбережение и ресурсную эффективность, была выявлена стоимость ресурсов проведения научного исследования, затраты на зарплаты, отчисления и накладные расходы. Определена ресурсная (ресурсосберегающая), финансовая, бюджетная, социальная и экономическая эффективности исследования, которая выявила, что исполнение 1 является самым эффективным из трех имеющихся вариантов. Проведена оценка потребителей, проведен QuaD-анализ, который позволил посчитать средневзвешенное значение показателя качества и перспективности научной разработки, P_{cp} составило 78 – это позволяет утверждать, что данная разработка является перспективной. Благодаря SWOT-анализу были показаны сильные и слабые стороны информационной системы, ее возможности и угрозы. Был составлен календарный план-график ВКР.

6 Социальная ответственность

Целью выпускной квалификационной работы является разработка информационной системы формата «Вопрос-ответ» с использованием фреймворка Ruby on Rails. Для разработки системы проводились различные теоретические исследования, анализ и структуризация полученных данных, а также проектирование и программирование системы с использованием вычислительной техники.

Так как выполнение работы заключалось в разработке приложения, то в качестве рабочего места будет рассмотрено рабочее место оператора персональной электронной вычислительной машины (ПЭВМ).

Использование средств вычислительной техники, накладывает целый ряд вредных факторов на человека, что впоследствии снижает производительность его труда и может привести к существенным проблемам со здоровьем сотрудника.

Обеспечение производственной и экологической безопасности является необходимым условием реализации любых проектов, в том числе конструкторских и исследовательских. В общем, обеспечение безопасности предполагает создание безопасных и благоприятных рабочих условий для лиц, задействованных в работе над проектом, а также условий, обеспечивающих экологическую безопасность окружающей среды.

Первичным этапом в задаче обеспечения безопасности труда является выявление и анализ вредных и опасных факторов труда оператора ПЭВМ, возможных причин потенциальных аварий и пожаров, производственных травм, профессиональных заболеваний. Следующими этапами в задачи обеспечения безопасности труда являются разработка мероприятий по защите вредных и опасных факторов, оценка условий труда и микроклимата рабочей среды.

Поэтому данный раздел посвящен анализу вредных и опасных факторов производственной среды для операторов ПЭВМ, разработке программ по минимизации воздействия вредоносного и опасного влияния выявленных факторов, а также программ по снижению вредных воздействий на окружающую

среду, экономии невозможных ресурсов и защите в чрезвычайных ситуациях.

6.1 Производственная безопасность

Для обеспечения производственной безопасности необходимо проанализировать воздействия на человека вредных и опасных производственных факторов, которые могут возникать при разработке или эксплуатации проекта.

Производственные условия на рабочем месте характеризуются наличием различных опасных и вредных производственных факторов, оказывающих негативное влияние на работников.

Производственный фактор считается вредным, если воздействие этого фактора на работника может привести к его заболеванию. Производственный фактор считается опасным, если его воздействие на работника может привести к его травме [12].

Вредные факторы характеризуются потенциальной опасностью для здоровья, в частности способствуют развитию каких-либо заболеваний, приводят к повышенной утомляемости и снижению работоспособности. При этом, вредные факторы проявляются при определенных условиях таких как интенсивность и длительность воздействия. Опасные производственные факторы способны моментально оказать влияние на здоровье работника: привести к травмам, ожогам или к резкому ухудшению здоровья работников в результате отравления или облучения.

В таблице 14 представлены возможные вредные и опасные факторы, возникающие при работе за ПЭВМ.

Таблица 14. Вредные и опасные факторы, возникающие при работе за ПЭВМ

Наименование видов работ	Факторы	Нормативные документы
Вредные факторы		
Работа за ПЭВМ	Отклонение показателей микроклимата (температуры и влажности воздуха)	СанПиН 2.2.2/2.4.1340-03 СанПиН 2.2.4.548-96
	Недостаточная освещенность рабочей зоны	СанПиН 2.2.2/2.4.1340- 03
Опасные факторы		
Работа за ПЭВМ	Опасность поражения электрическим током	ГОСТ 12.1.038–82
	Пожаровзрывоопасность	ФЗ от 22.07.2008 №123-ФЗ

6.2 Вредные производственные факторы

6.2.1 Отклонение показателей микроклимата в помещении

Одним из необходимых благоприятных условий труда является обеспечение в помещениях нормальных условий микроклимата, оказывающих существенное влияние на тепловое самочувствие человека. Микроклимат в производственных помещениях, зависит от особенностей технологического процесса, а также внешних условий (категории работ, периода года, условий вентиляции и отопления).

К параметрам, характеризующим микроклимат в производственных помещениях, относятся:

- температура воздуха (t , °C);
- температура поверхностей (t , °C);
- относительная влажность воздуха (ϕ , %);
- скорость движения воздуха (v , м/с);
- интенсивность теплового облучения (I , Вт/м²).

В производственных помещениях для работы с ПЭВМ происходит постоянное выделение тепла самой вычислительной техникой, вспомогательными приборами и средствами освещения. Поскольку оператор

расположен в непосредственной близости с источниками выделения тепла, то данный фактор является одним из важнейших вредных факторов производственной среды оператора ПЭВМ, а высокая температура воздуха способствует быстрому перегреву организма и быстрой утомляемости [13].

Влажность оказывает большое влияние на терморегуляцию организма. Так, например, высокие показатели относительной влажности (более 85 %) затрудняют терморегуляцию снижая возможность испарения пота, низкие показатели влажности (менее 20 %) вызывают пересыхание слизистых оболочек человека [14].

Работа программиста относится к категории Ia, которые производятся сидя и сопровождаются незначительным физическим напряжением. Интенсивность энерготрат организма для данной категории работ составляет до 120 ккал/ч (до 139 Вт).

Оптимальные значения показателей микроклимата на рабочих местах производственных помещений согласно СанПиН 2.2.4.548-96 для категории работ Ia представлены в таблице 15 [15].

Таблица 15. Оптимальные величины показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia	22 – 24	60 – 40	0,1
Теплый	Ia	21 – 23	60 – 40	0,1

Оптимальные микроклиматические условия обеспечивают общее и локальное ощущение теплового комфорта при минимальном напряжении механизмов терморегуляции, не вызывают отклонений в состоянии здоровья, создают предпосылки для высокого уровня работоспособности и являются предпочтительными на рабочих местах.

В таблице 16 приведены допустимые величины показателей микроклимата на рабочих местах производственных помещений согласно СанПиН 2.2.4.548-96 для категории работ Ia [15].

Таблица 16. Допустимые величины показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia	20 – 25	15 – 75	0,1
Теплый	Ia	21 – 28	15 – 75	0,1 – 0,2

Допустимые микроклиматические условия не вызывают повреждений или нарушений состояния здоровья, но могут приводить к возникновению общих и локальных ощущений теплового дискомфорта, напряжению механизмов терморегуляции, ухудшению самочувствия и понижению работоспособности.

Согласно требованиям СанПиН 2.2.2/2.4.1340–03, в кабинете поддерживается температура равная 19–20 С°, при относительной влажности в 55–58%. Для этого в помещении проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ [16].

6.2.2 Недостаточная освещенность рабочей зоны

Недостаточная освещенность рабочей зоны является вредным производственным фактором, возникающим при работе с ПЭВМ, уровни которого регламентируются СП 52.13330.2011.

Работа с компьютером подразумевает постоянный зрительный контакт с дисплеем ПЭВМ и занимает от 80 % рабочего времени.

Недостаточный уровень освещенности в помещении приводит к снижению остроты зрения, головным болям, снижению концентрации внимания и, как следствие, к ухудшению производительности труда.

Причиной недостаточной освещенности являются недостаточность естественного освещения, недостаточность искусственного освещения, пониженная контрастность.

Рабочее помещение должно иметь как естественное, так и искусственное освещение. Коэффициент естественного освещения должен быть не менее 1,2%.

Согласно СанПиН 2.2.2/2.4.1340-03 освещенность на поверхности рабочего стола в зоне размещения документа должна быть 300 – 500 лк, что может достигаться установкой местного освещения, не создающего бликов на поверхности экрана. Освещенность поверхности экрана не должна превышать 300 лк. Яркость светящихся поверхностей (окон, светильников), находящихся в поле зрения должна быть не более 200 кд/м². Для источников искусственного освещения следует применять люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). Коэффициент пульсации при работе с ПЭВМ не должен превышать 5%.

Следует ограничивать отраженную блескость на рабочих поверхностях (экран, стол, клавиатура) за счет правильного выбора и расположения светильников, яркость бликов на экране не должна превышать 40 кд/м². Светильники местного освещения должны иметь непросвечивающий отражатель.

6.3 Опасные производственные факторы

6.3.1 Опасность поражения электрическим током

Поражение электрическим током является опасным производственным фактором и, поскольку оператор ПЭВМ имеет дело с электрооборудованием, то вопросам электробезопасности на его рабочем месте должно уделяться много внимания.

Опасность поражения человека электрическим током оценивается величиной тока I (А), проходящего через его тело, или напряжением прикосновения U (В). Степень опасного воздействия на человека электрического тока зависит от рода и величины напряжения тока, частоты электрического тока, пути тока через тело человека, продолжительности его воздействия на организм человека, а также условий внешней среды.

Электрический ток, протекая через тело человека, производит термическое, механическое и световое воздействие – электролитическое разложение жидкости (в том числе и крови), судорожное сокращение мышц,

разрыв тканей и поражение глаз.

Работа с ПЭВМ является опасной с точки зрения поражения током, так как практически во всех частях компьютера течет электрический ток.

Поражение электрическим током при работе в ПЭВМ возможно при наличии оголенных участков на кабеле, нарушении изоляции распределительных устройств и от токоведущих частей компьютера в случае их пробоя и нарушении изоляции, при работе с ПЭВМ во влажной одежде и влажными руками.

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

К мероприятиям по предотвращению возможности поражения электрическим током относятся:

- при производстве монтажных работ необходимо использовать только исправный инструмент, аттестованный службой КИПиА;
- с целью защиты от поражения электрическим током, возникающим между корпусом приборов и инструментом при пробое сетевого напряжения на корпус, корпуса приборов и инструментов должны быть заземлены;
- при включенном сетевом напряжении работы на задней панели должны быть запрещены;
- все работы по устранению неисправностей должен производить квалифицированный персонал;
- необходимо постоянно следить за исправностью электропроводки [16].

Согласно ГОСТ 12.1.038-82 на рабочем месте программиста допускаются уровни напряжений прикосновения и токов, представленные в таблице 17 [17].

Таблица 17. Предельно допустимые напряжения прикосновения и токи

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

Значения напряжения прикосновения и токов приведены при продолжительности воздействия не более 10 минут в сутки.

6.3.2 Пожаровзрывобезопасность

Возникновение пожара является опасным производственным фактором, т.к. пожар на предприятии наносит большой материальный ущерб, а также часто сопровождается травмами и несчастными случаями.

Пожарная безопасность представляет собой единый комплекс организационных, технических, режимных и эксплуатационных мероприятий по предупреждению пожаров и взрывов.

В помещениях с ПЭВМ повышен риск возникновения пожара из-за присутствия множества факторов: наличие большого количества электронных схем, устройств электропитания, устройств кондиционирования воздуха; возможные неисправности электрооборудования, освещения, или неправильная их эксплуатация может послужить причиной пожара.

Согласно ФЗ от 22.07.2008 №123-ФЗ для устранения возможных причин возникновения пожаров необходимо проводить следующие мероприятия:

- Организационные мероприятия:
 - противопожарный инструктаж обслуживающего персонала;
 - обучение персонала техники безопасности;
 - разработка инструкций, плакатов, планов эвакуации.
- Эксплуатационные мероприятия:
 - соблюдение эксплуатационных норм оборудования;
 - выбор и использование современных автоматических средств тушения пожаров.

- Технические мероприятия:
 - профилактический осмотр и ремонт оборудования;
 - соблюдение противопожарных мероприятий при устройстве электропроводок, оборудования, систем отопления, вентиляции и освещения.

6.4 Экологическая безопасность

6.4.1 Анализ воздействия продукта на окружающую среду

Разработанный программный продукт, не наносит вреда окружающей среде ни на стадиях его разработки, ни на стадиях эксплуатации. Однако, средства, необходимые для его разработки и эксплуатации могут наносить вред окружающей среде.

Согласно СанПиН 2.2.1/2.2.1.1200-03 объект, на котором производилась разработка продукта, а также объекты, на которых будет производиться его использование операторами ПЭВМ относятся к предприятиям пятого класса, размер санитарной зоны для которых равен 50 м.

Основными факторами, оказывающими негативные действия на экологию, являются факторы, связанные с производством и эксплуатацией компьютерной техники. В частности, отходы и выбросы, имеющие место на этапе производства компьютеров, а также отходы, связанные с неполной их утилизацией.

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

- локальное повышение электромагнитного и радиоактивного фона;
- неоправданное потребление электроэнергии (связано с использованием компьютера не на полную мощность в течение всего его время работы) и прочее.

6.4.2 Решения по обеспечению экологической безопасности

При разработке любых автоматизированных систем возникает необходимость утилизировать производственные отходы, в качестве которых в данном случае выступают бумажные отходы (макулатура) и неисправные детали персональных компьютеров, плат, контроллеров.

Согласно СанПиН 2.1.7.1322-03 бумажные отходы должны передаваться в соответствующие организации для дальнейшей переработки во вторичные бумажные изделия. Неисправные комплектующие персональных компьютеров должны передаваться либо государственным организациям, осуществляющим вывоз и уничтожение бытовых и производственных отходов, либо организациям, занимающимся переработкой отходов. Важнейшим этапам обращения с отходами является их сбор, а в дальнейшем переработка, утилизация и захоронение [18].

6.5 Безопасность в чрезвычайных ситуациях

6.5.1 Перечень возможных ЧС при разработке и эксплуатации проектируемого решения

Согласно ГОСТ Р 22.0.07-95 возможными чрезвычайными ситуациями могут быть:

- техногенные: взрывы, пожары, обрушение помещений, аварии на системах жизнеобеспечения;
- природные: наводнения, ураганы, бури, природные пожары;
- экологические: разрушение озонового слоя, кислотные дожди;
- биологические: эпидемии, пандемии;
- антропогенные: война, терроризм.

Общие правила поведения при чрезвычайных ситуациях:

1) Не паниковать и не поддаваться панике. Призывать окружающих к спокойствию.

2) По возможности немедленно позвонить по телефону «01», сообщить что случилось, указать точный адрес места происшествия, назвать свою фамилию и номер своего телефона.

3) Включить устройства передачи звука (радио, телевизор), а также прослушать информацию, передаваемую через уличные громкоговорители и громкоговорящие устройства. В речевом сообщении будут озвучены основные рекомендации и правила поведения.

4) Выполнять рекомендации специалистов (сотрудников полиции, медицинских работников, пожарных, спасателей).

5) Не создавать условия, которые препятствуют и затрудняют действия сотрудников полиции, медицинских работников, спасателей, пожарных.

Наиболее характерной для объекта, где размещаются рабочие помещения, оборудованные ПЭВМ, чрезвычайной ситуацией является пожар.

Причинами возникновения данного вида ЧС могут являться:

- возникновением короткого замыкания в электропроводке;
- возгоранием устройств ПЭВМ из-за неисправности аппаратуры;
- возгоранием устройств искусственного освещения;
- возгоранием мебели по причине нарушения правил пожарной безопасности, а также неправильного использования дополнительных бытовых электроприборов и электроустановок.

Помещение для работы операторов ПЭВМ по системе классификации категорий помещений по взрывопожарной и пожарной опасности относится к категории Д (из 5-ти категорий А, Б, В1-В4, Г, Д), т.к. относится к помещениям с негорючими веществами и материалами в холодном состоянии [19].

6.5.2 Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.

Основные правила пожарной безопасности указаны в документе **«Правила противопожарного режима в РФ» от 25 апреля 2012г.**

Пожарная безопасность подразумевает надлежащее состояние объекта с исключением возможности возникновения очага возгорания (пожара) и его распространения в пространстве. Обеспечение пожарной безопасности – приоритетная задача для любого предприятия. Создание системы защиты регламентировано законом и нормативными документами различных ведомств.

Каждый сотрудник организации должен быть ознакомлен с инструкцией по пожарной безопасности, пройти инструктаж по технике безопасности и строго соблюдать его.

Запрещается использовать электроприборы в условиях, не

соответствующих требованиям инструкций изготовителей, или имеющие неисправности, которые в соответствии с инструкцией по эксплуатации могут привести к пожару, а также эксплуатировать электропровода и кабели с поврежденной или потерявшей защитные свойства изоляцией.

Электроустановки и бытовые электроприборы в помещениях по окончании рабочего времени должны быть обесточены (вилки должны быть вынуты из розеток). Под напряжением должны оставаться дежурное освещение и пожарная сигнализация. Недопустимо хранение легковоспламеняющихся, горючих и взрывчатых веществ, использование открытого огня в помещениях офиса.

Перед уходом из служебного помещения работник обязан провести его осмотр, закрыть окна, и убедиться в том, что в помещении отсутствуют источники возможного возгорания, все электроприборы отключены и выключено освещение. С периодичностью не реже одного раза в три года необходимо проводить замеры сопротивления изоляции токоведущих частей силового и осветительного оборудования.

Работник при обнаружении пожара или признаков горения (задымление, запах гари, повышение температуры и т.п.) должен:

- немедленно прекратить работу и вызвать пожарную охрану по телефону «01», сообщив при этом адрес, место возникновения пожара и свою фамилию;
- принять по возможности меры по эвакуации людей и материальных ценностей;
- отключить от сети закрепленное за ним электрооборудование;
- приступить к тушению пожара имеющимися средствами пожаротушения;
- сообщить непосредственному или вышестоящему начальнику и оповестить окружающих сотрудников;
- При общем сигнале опасности покинуть здание согласно «Плану эвакуации людей при пожаре и других ЧС».

Для тушения пожара применять ручные углекислотные огнетушители (типа ОУ-2, ОУ-5), находящиеся в помещениях офиса, и пожарный кран внутреннего противопожарного водопровода. Они предназначены для тушения начальных возгораний различных веществ и материалов, за исключением веществ, горение которых происходит без доступа воздуха. Огнетушители должны постоянно содержаться в исправном состоянии и быть готовыми к действию. Категорически запрещается тушить возгорания в помещениях офиса при помощи химических пенных огнетушителей (типа ОХП-10) [20].

6.6 Правовые и организационные вопросы обеспечения безопасности

Регулирование отношений между работником и работодателем, касающихся оплаты труда, трудового распорядка, особенности регулирования труда женщин, детей, людей с ограниченными способностями и проч., осуществляется законодательством РФ, а именно **трудовым кодексом РФ**.

Режим труда и отдыха предусматривает соблюдение определенной длительности непрерывной работы на персональном компьютере (ПК) и перерывов, регламентированных с учетом продолжительности рабочей смены, видов и категории трудовой деятельности.

Вид трудовой деятельности на персональном компьютере в рамках данной работы соответствует группе В – творческая работа в режиме диалога с ПК, категория трудовой деятельности – I (до 2 часов непосредственной работы на ПК).

При 8-часовой рабочей смене и работе на ПК, соответствующей описанным выше критериям необходимо через 1,5- 2,0 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва устраивать регламентированные перерывы продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

Продолжительность непрерывной работы на ПК без регламентированного перерыва не должна превышать 2 часа.

Эффективными являются нерегламентированные перерывы

(микропаузы) длительностью 1-3 минуты.

Регламентированные перерывы и микропаузы целесообразно использовать для выполнения комплекса упражнений и гимнастики для глаз, пальцев рук, а также массажа. Комплексы упражнений целесообразно менять через 2-3 недели. Продолжительность рабочего дня не должна быть меньше указанного времени в договоре, но не больше 40 часов в неделю. Для работников до 16 лет – не более 24 часов в неделю, от 16 до 18 лет и инвалидов I и II группы – не более 35 часов.

Возможно установление неполного рабочего дня для беременной женщины; одного из родителей (опекуна, попечителя), имеющего ребенка в возрасте до четырнадцати лет (ребенка-инвалида в возрасте до восемнадцати лет). Оплата труда при этом производится пропорционально отработанному времени, без ограничений оплачиваемого отпуска, исчисления трудового стажа и других прав.

При работе в ночное время продолжительность рабочей смены сокращается на один час. К работе в ночную смену не допускаются беременные женщины; работники, не достигшие возраста 18 лет; женщины, имеющие детей в возрасте до трех лет, инвалиды, работники, имеющие детей-инвалидов, а также работники, осуществляющие уход за больными членами их семей в соответствии с медицинским заключением, матери и отцы-одиночки детей до пяти лет.

Организация обязана предоставлять ежегодный отпуск продолжительностью 28 календарных дней. Дополнительные отпуска предоставляются работникам, занятым на работах с вредными или опасными условиями труда, работникам имеющим особый характер работы, работникам с ненормированным рабочим днем и работающим в условиях Крайнего Севера и приравненных к нему местностях [21].

6.6.1 Организационные мероприятия при компоновке рабочей зоны

Большое значение для профилактики статических физических перегрузок имеет правильная организация рабочего места человека, работающего с ПЭВМ. Рабочее место должно быть организовано в

соответствии с требованиями стандартов, технических условий и (или) методических указаний по безопасности труда. Оно должно удовлетворять следующим требованиям:

- обеспечивать возможность удобного выполнения работ;
- учитывать физическую тяжесть работ;
- учитывать размеры рабочей зоны и необходимость передвижения в ней работающего;
- учитывать технологические особенности процесса выполнения работ.

Невыполнение требований к расположению и компоновке рабочего места может привести к получению работником производственной травмы или развития у него профессионального заболевания. Рабочее место программиста должно соответствовать требованиям **СанПин 2.2.2/2.4.1340-03**.

Конструкция оборудования и рабочего места при выполнении работ в положении сидя должна обеспечивать оптимальное положение работающего, которое достигается регулированием высоты рабочей поверхности, высоты сидения, оборудованием пространства для размещения ног и высотой подставки для ног. Схемы размещения рабочих мест с персональными компьютерами должны учитывать расстояния между рабочими столами с мониторами: расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м. Клавиатура должна располагаться на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю. Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно перпендикулярно к нормальной линии взгляда (нормальная линия взгляда 15 градусов вниз от горизонтали). Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5 - 2,0 м [16].

6.6.2 Специфика влияния продукта на рабочий процесс

Разрабатываемый в ходе выполнения ВКР программный продукт может использоваться для помощи при решении проблем пользователей на любых интернет-ресурсах. Постоянный мониторинг за системой не требуется, система мониторинга позволяет своевременно оповещать администратора о наличии проблем, что значительно снижает количество времени, потраченное администратором. Это позволяет говорить о высокой автономности системы.

Заключение

В результате выполнения выпускной квалификационной работы была достигнута ее цель – разработка информационной системы формата «Вопрос-ответ», используя язык Ruby и фреймворк Ruby on Rails.

Система соответствует всем выявленным в процессе анализа требованиям, разработка велась по TDD, BDD и Agile методологиям. Это позволило покрыть всю систему автоматизированными тестами, а также провести разработку итерационного характера. Также система была протестирована в реальном окружении, выполнены задачи по резервному копированию, обновлению индекса поиска и мониторингу, а также первоначальной настройке сервера.

В дальнейшем планируется разработать уникальный дизайн и сделать пользовательский интерфейс гораздо удобнее, добавить облако тегов, расширенный поиск и систему репутации.

Conclusion

Main task was achieved as a result of completing bachelor project: to develop Q&A information system using Ruby on Rails framework.

The system meets all requirements that were identified in process of analysis, TDD, BDD and Agile methodologies were used in development. This allows covering all system with automated tests and completing iterative development. The system was tested in production environment, following tasks were accomplished there: to update search index and to backup database by timetable, to monitor system using special utility, and set up initial server configuration.

In the future, it is planned to make unique design with user-friendly interface, to add tag cloud, advanced search and user reputation system.

Список использованных источников

1. Compare OSQA and AnswerHub [Электронный ресурс] / AnswerHub. URL: <http://www.dzonesoftware.com/products/answerhub-versus-osqa>, свободный. Яз. Англ. Дата обращения: 20.04.2018.
2. Create your own Q&A [Электронный ресурс] / Askbot. URL: <https://askbot.com/>, свободный. Яз. Англ. Дата обращения: 20.04.2018.
3. О Ruby [Электронный ресурс] / Язык программирования Ruby ruby-lang. URL: <https://www.ruby-lang.org/ru/about/>, свободный. Яз. Рус. Дата обращения: 20.04.2018.
4. Ruby on Rails по-русски [Электронный ресурс] / Rusrails. URL: <http://rusrails.ru/>, свободный. Яз. Рус. Дата обращения: 20.04.2018.
5. Разработка через тестирование [Электронный ресурс] / Википедия. URL: https://ru.wikipedia.org/wiki/Разработка_через_тестирование, свободный. Яз. Рус. Дата обращения: 25.04.2018.
6. BDD – рабочий метод или TDD в модной обертке? [Электронный ресурс] / Habrahabr. URL: <https://habr.com/company/jugru/blog/329468/>, свободный. Яз. Рус. Дата обращения: 25.04.2018.
7. Асинхронный веб или что такое веб-сокеты [Электронный ресурс] / Tproger. URL: <https://tproger.ru/translations/what-are-web-sockets/>, свободный. Яз. Рус. Дата обращения: 25.04.2018.
8. OAuth [Электронный ресурс] / Википедия. URL: <https://ru.wikipedia.org/wiki/OAuth>, свободный. Яз. Рус. Дата обращения: 25.04.2018.
9. Полнотекстовый поиск [Электронный ресурс] / Википедия. URL: https://ru.wikipedia.org/wiki/Полнотекстовый_поиск, свободный. Яз. Рус. Дата обращения: 25.04.2018.

10. Кузьмина Е.А, Кузьмин А.М. Методы поиска новых идей и решений "Методы менеджмента качества" №1 2003 г.
11. Кузьмина Е.А, Кузьмин А.М. Функционально-стоимостный анализ. Экскурс в историю. "Методы менеджмента качества" №7 2002 г.
12. Охрана труда [Электронный ресурс] / Безопасность жизнедеятельности. URL:
<http://www.grandars.ru/shkola/bezopasnostzhiznedeyatelnosti/ohranatruda.html>, свободный. Яз. Рус. Дата обращения: 10.05.2018.
13. Ефремова О. С. Требования охраны труда при работе на персональных электронно-вычислительных машинах. – 2-е изд., перераб. и доп. – М. : Издательство «Альфа-Пресс», 2008. Яз. Рус. Дата обращения: 10.05.2018.
14. Назаренко О. Б. Безопасность жизнедеятельности: учебное пособие / О. Б. Назаренко, Ю. А. Амелькович; Томский политехнический университет. – 3-е изд., перераб. и доп. – Томск: Изд-во Томского политехнического университета, 2013. Яз. Рус. Дата обращения: 10.05.2018.
15. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений. Яз. Рус. Дата обращения: 10.05.2018
16. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. Яз. Рус. Дата обращения: 10.05.2018.
17. ГОСТ 12.1.038–82 Система стандартов безопасности труда. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов. Яз. Рус. Дата обращения: 10.05.2018.
18. ГОСТ Р 55090-2012 Ресурсосбережение. Обращение с отходами. Рекомендации по утилизации отходов бумаги. Яз. Рус. Дата обращения: 10.05.2018.
19. НПБ 105-03 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности [Электронный

ресурс] / Электронный фонд правовой и нормативно-технической документации.

URL: <http://docs.cntd.ru/document/1200032102>, свободный. Яз. Рус. Дата обращения: 10.05.2018.

20. ППБ 01–03. Правила пожарной безопасности в Российской Федерации. – М.: Министерство Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий, 2003.

21. Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 03.07.2016) (с изм. и доп., вступ. в силу с 01.01.2017) [Электронный ресурс] / Консультант Плюс. URL: https://www.consultant.ru/document/cons_doc_LAW_34683/, свободный. Яз. Рус. Дата обращения: 10.05.2018.

Приложение А

Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы				Конкурентоспособность			
		Б _ф	Б _{к1}	Б _{к2}	Б _{к3}	К _ф	К _{к1}	К _{к2}	К _{к3}
Надежность	0,1	4	4	5	4	0,4	0,4	0,5	0,4
Удобство в эксплуатации	0,15	5	4	3	4	0,75	0,6	0,45	0,6
Безопасность	0,1	4	4	5	5	0,4	0,4	0,5	0,5
Потребность ресурсов памяти	0,1	4	4	2	4	0,4	0,4	0,2	0,4
Простота интерфейса	0,1	5	3	3	3	0,5	0,3	0,3	0,3
Экономические критерии оценки ресурсоэффективности									
Конкурентоспособность продукта	0,1	5	4	5	4	0,5	0,4	0,5	0,4
Область применения	0,1	4	3	4	5	0,4	0,3	0,3	0,4
Уровень проникновения на рынок	0,1	4	5	4	4	0,4	0,5	0,4	0,4
Цена	0,05	5	3	3	2	0,5	0,15	0,15	0,1
Срок выхода на рынок	0,1	4	4	4	3	0,4	0,4	0,4	0,3
Итого	1					4,65	3,85	3,7	3,8

Приложение Б

Оценочная карта для сравнения конкурентных технических решений по технологии QuaD

Критерии оценки	Вес критерия	Баллы	Макс. балл	Относительное значение (3/4)	Средневзвешенное значение
Показатели оценки качества разработки					
Надежность	0,07	80	100	0,8	0,056
Безопасность	0,1	75	100	0,75	0,075
Потребность в ресурсах памяти	0,1	100	100	1	0,1
Удобство эксплуатации	0,1	90	100	0,9	0,09
Удобство интерфейса	0,07	70	100	0,7	0,049
Модифицируемость	0,13	90	100	0,9	0,117
Показатели оценки коммерческого потенциала разработки					
Конкурентоспособность продукта	0,1	90	100	0,9	0,09
Уровень проникновения на рынок	0,1	65	100	0,65	0,065
Финансовая эффективность научной разработки	0,05	75	100	0,75	0,0375
Перспективность рынка	0,1	90	100	0,9	0,09
Цена	0,08	100	100	1	0,008
Итого	1	925	1100	9,25	0,7775

Приложение В

Интерактивная матрица проекта

Сильные стороны проекта				
		Удобный и простой интерфейс	Хорошая масштабируемость	Экономия ресурсов
Возможности проекта	Быстрое развертывание	+	+	0
	Возможность расширения	0	+	-
	Возможность внедрения в систему ТПУ	0	0	-
Угрозы проекта				
	Отсутствие финансирования	+	0	+
	Высокие требования пользователей	-	+	-
	Сбои в работе программы и сервера	0	-	+
Слабые стороны проекта				
		Простой дизайн	Отсутствие профессиональной команды разработчиков	Относительная простота реализации
Возможности проекта	Быстрое развертывание	+	0	+
	Возможность внедрения в систему ТПУ	-	0	0
	Возможность расширения	0	-	+
Угрозы проекта	Отсутствие финансирования	+	+	+
	Высокие требования пользователей	-	-	0
	Сбои в работе программы и сервера	-	+	0

Приложение Г

Перечень работ и подсчет показателей

	t_{\min}	t_{\max}	$t_{\text{ожид}}$	Ψ	T_{pi}
Составление и утверждение технического задания	3	6	4,2	2	2,1
Подбор и анализ материалов для предметной области	2	5	3,2	1	3,2
Выработка концепции планируемой системы	3	6	4,2	2	2,1
Выбор технологии для проведения исследования	1	2	1,4	2	0,7
Календарное планирование	1	2	1,4	2	0,7
Выбор среды разработки	1	2	1,4	1	1,4
Выбор методологий разработки	1	2	1,4	1	1,4
Первичное проектирование системы	2	4	2,8	1	2,8
Разработка и тестирование системы	45	75	57	1	57
Развертывание системы на реальном сервере	1	3	1,8	1	1,8
Отладка, доработка интерфейса	2	5	5,2	1	5,2
Составление пояснительной записки	7	14	9,8	2	4,9

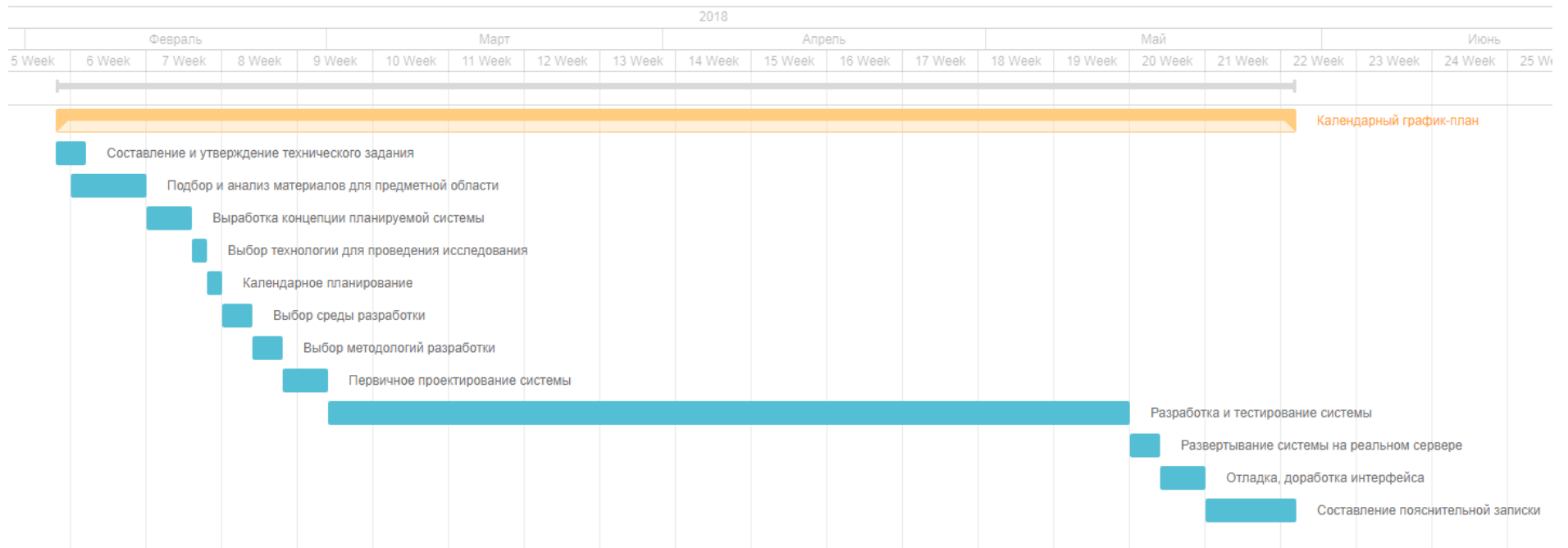
Приложение Д

Временные показатели проведения научного исследования

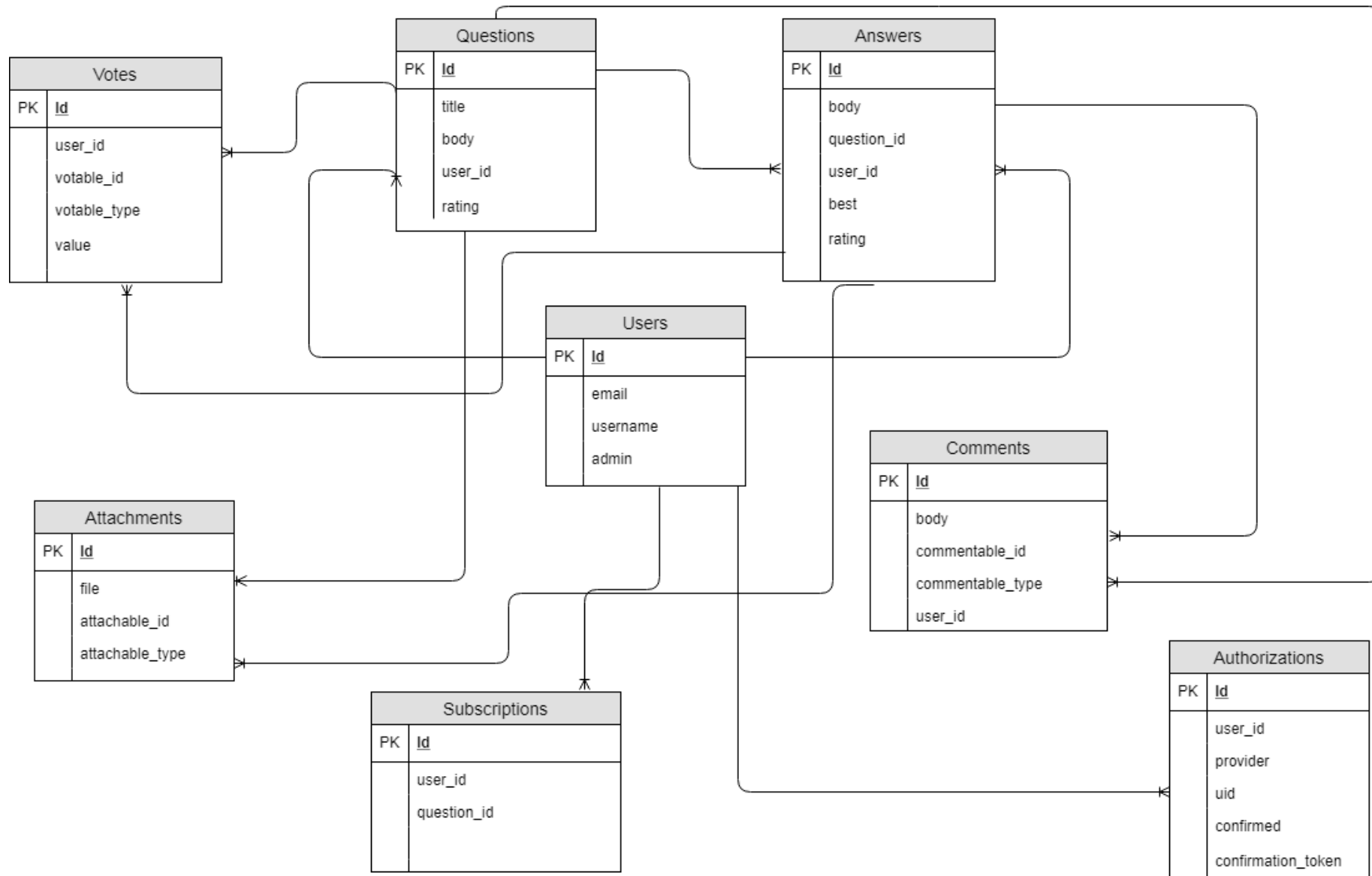
Название работы, исполнитель и	Трудоёмкость работ									Длительность работ (раб. дни)			Длительность работ (кал. дни)		
	T _{min} , чел-дни			T _{max} , чел-дни			T _{ож} , чел-дни								
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Составление и утверждение технического задания (С, Р)	3	3	3	6	6	6	4,2	4,2	4,2	2,1	2,1	2,1	3,15	3,15	3,15
Подбор и анализ материалов для предметной области (С)	2	3	3	5	6	6	3,2	4,2	4,2	3,2	4,2	4,2	4,8	6,3	6,3
Выработка концепции планируемой системы (С, Р)	3	4	5	6	7	8	4,2	5,2	6,2	2,1	2,6	3,1	3,05	3,9	4,65
Выбор технологии для проведения исследования (С, Р)	1	1	1	2	2	2	1,4	1,4	1,4	0,7	0,7	0,7	1,05	1,05	1,05
Календарное планирование (С, Р)	1	1	1	2	2	2	1,4	1,4	1,4	0,7	0,7	0,7	1,05	1,05	1,05
Выбор среды разработки (С)	1	1	1	2	2	2	1,4	1,4	1,4	1,4	1,4	1,4	2,1	2,1	2,1
Выбор методологий разработки (С)	1	1	1	2	2	2	1,4	1,4	1,4	1,4	1,4	1,4	2,1	2,1	2,1
Первичное проектирование системы (С)	2	3	4	4	6	8	2,8	4,2	5,6	2,8	4,2	5,6	4,2	6,3	8,4
Разработка и тестирование системы (С)	45	50	50	75	85	85	57	64	64	57	64	64	85,5	96	96
Развертывание системы на реальном сервере (С)	1	2	2	2	3	3	1,8	3	3	1,8	3	3	2,7	4,5	4,5

Отладка, доработка интерфейса (С)	2	3	3	5	6	6	3,2	4,2	4,2	3,2	4,2	4,2	4,8	6,3	6,3
Составление пояснительной записки (С, Р)	7	7	7	14	14	14	9,8	9,8	9,8	4,9	4,9	4,9	7,35	7,35	7,35
Итого													121,85	140,1	142,95

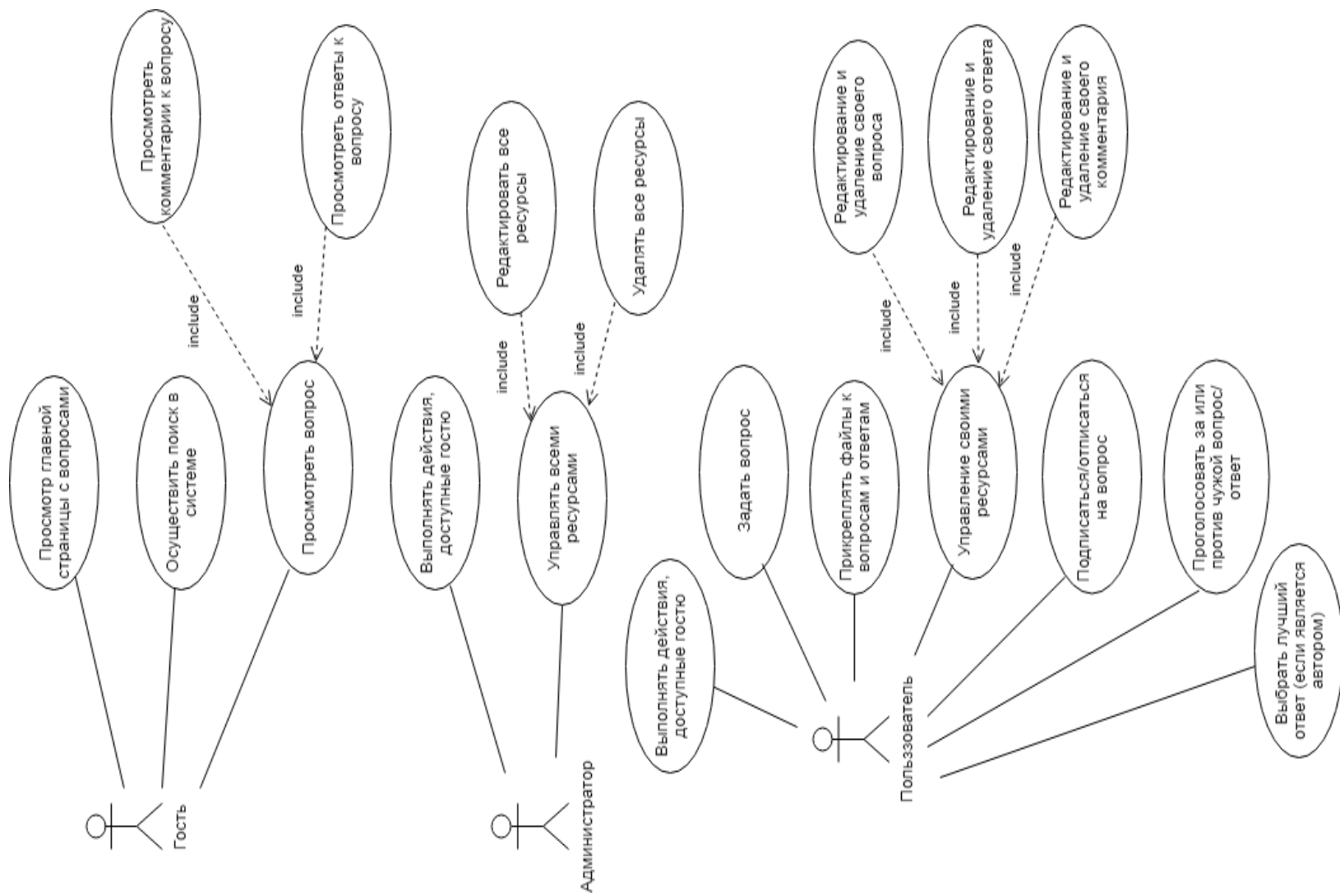
Приложение Е



Приложение Ж



Приложение 3



Приложение II

