

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки / специальности)

Бизнес-информатика

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка приложения для автоматизированного анализа отзывов
клиентов

Обучающийся

Сафонов К.С.

(Инициалы Фамилия)

_____ (личная подпись)

Руководитель

Климов В.С.

_____ (ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

Аннотация

Аннотация на дипломную работу студента Сафонов К.С. на тему: «Разработка приложения для автоматизированного анализа отзывов клиентов»

Работа включает в себя: 79 страниц, 6 таблиц, 66 рисунков, список использованных источников (23 позиция), 0 приложений.

Предмет исследования – является отдел веб разработок, связанных с развитием интернет ресурса с элементами отзывов.

Целью работы состоит разработка программного обеспечения для сбора пользовательских данных и анализ их в автоматическом режиме. После преобразование их в удобный вид графиков или таблиц.

Результатам исследования данной темы, была выполнена работа по разработке первичного дизайна, основного функционала программного обеспечения для темы, выполнено построение инструкции по эксплуатации данной темы и подведены итоги всей работы.

Небыли выполнены такие задачи, как построение итогового дизайна и разработка остальных разделов веб проекта.

Полученные результаты и практические навыки позволят улучшить последующее развитие данного и будущих проектов.

Содержание

Введение.....	4
1 Практический опыт в организации	5
1.1 Описание целей и задач предприятия.....	5
1.2 Составить организационную структуру предприятия с распределением ответственности, полномочий и взаимоотношениями между отделами предприятия	5
1.3 Провести анализ предметной области и взаимодействие практики с темой ВКР	15
2 Теоретическое изучение темы квалификационной работы.....	20
2.1 Моделирование бизнес–процессов предметной области для постановки задачи автоматизированного варианта решения	20
2.2 Построить функциональную модель «как есть», «как должно быть» ...	20
2.3 Проанализировать схожих работ и обосновать выбор решения по оптимизации и реинжиниринге.....	26
2.4 Проанализировать существующие разработки для решения обозначенной задачи.....	27
2.5 Описать функциональные требования к проекту и сформировать бизнес–цели и требования для ее реализации.....	33
2.6 Описание и обоснование решения по системной архитектуре проекта	34
3 Постройка и описание информационной модели.....	37
3.1 Построение концептуальной схемы приложения.....	37
3.2 Разработка программ для реализации алгоритмов обработки данных ..	39
3.3 Разработка интерфейса под все шаблоны	43
3.4 Описание модулей приложения АИС	46
3.5 Создание прототипа базы данных	47
3.6 Разработка контрольных примеров работы базы данных	58
3.7 Разработка инструкция для каждого пользователя	65
3.8 Предоставление контрольных примеров.....	71
3.9 Анализ данных отзывов.....	74
Заключение	77
Список используемой литературы и используемых источников	78

Введение

Начало данной выпускной квалификационной работы будет происходить на учебной практике в организации ООО «Палитра», ее назначение — это развитие навыков разработки своего программного обеспечения и показать на что годится ученик если он проходит практику в большой организации.

Целью прохождения учебной практики и конечно выпускной квалификационной работы — это проекта под названием разработка приложения для автоматизированного анализа отзывов клиентов.

Каждый проект является масштабным, и он делится на 3 этапа, анализ, разработка, тестирование. Поэтому задачи были поставлены под эти 3 этапа.

Были поставлены такие задачи как:

- анализ уже имеющихся разработок, и разбор, и модификация данных разработок;
- выяснение плюсов и минусов программного обеспечения;
- выявление критериев на разработку и составление отчета по убытку при их разработке;
- составление плана разработки;
- разработка программного обеспечения;
- тестирование программного обеспечения;

В ходе работ по разработки программного обеспечения, были выполнены такие основные работы как, сама разработка программного обеспечения, ее анализ, похожие инструменты и выявлены критерии на ее разработку, создание основной и резервной базы данных, разработан тестовый дизайн для простого лучшего восприятия проекта для всех пользователей, разработка основного функционала таких как, профиль, отчетность, сбор статистики, авторизация и регистрация, авто выставление прав доступа. При этом все эти функции протестированы и работают в нормальном режиме. Основные задачи, поставленные на учебную практику и самую выпускную квалификационную работу, были выполнены в полном объеме.

1 Практический опыт в организации

1.1 Описание целей и задач предприятия

Организация ООО «ПАЛИТРА» в которой я прохожу производственную практику занимает сферу в оптово розничных продажах. Данная организация занимает это место с 2007 года, и удерживает данную позицию по сегодняшний день.

Главной целью данной организации является обеспечение сторонних организаций всесторонней помощи.

В задачи данной организации входит:

- планирование и реализация открытия торговых мест;
- закупка и монтаж технического оборудования;
- создание и обеспечение базы данных;
- помощь и обучение новых сотрудников;
- сбор и анализ получаемых данных по открытым местам.

Благодаря данным задачам, новые организации получают помощь в различных сферах в зависимости от нужд. По этой причине данная организация занимает свою нишу в экономической сфере.

1.2 Составить организационную структуру предприятия с распределением ответственности, полномочий и взаимоотношениями между отделами предприятия

Моя организационная структура компании разделяется на несколько отделов, которые выполняют свои функции в определенных задачах. А некоторые и вовсе соединены договором и работают сами по себе.

В общую структуру входят следующие отделы:

- директор – главный руководитель всех отделов, через которого проходит большинство задач связанные с переговорами с другими организациями;
- заместитель директора, отвечающий за общие вопросы внутри организации. В них так же входят проблемы в организации и организационные вопросы;
- бухгалтерия – производит внутренний и внешний документооборот;
- юридический отдел – это юристы, которые занимаются документами связанные с законом;
- отдел кадров – занимается подбором персонала и организацией рабочего времени сотрудников;
- отдел продаж – это ИП организации, которые производят продажу различной продукции;
- IT отдел – занимается техническими вопросами и предоставляет отделу продаж нужное оборудования для работы;
- отдел закупок и логистики – занимается заказом товаров для ИП и перевозками в торговые точки транспортировкой между ними;
- склад – предназначен для хранения различных вещей и оборудования.
- административно–хозяйственный отдел – обеспечивает техническое обслуживание зданий, помещений, оборудования, а также проводит контроль текущих и капитальных ремонтов, осуществляет снабжение хозяйственным инвентарем и средствами механизации;
- маркетинг – задача данного отдела заключается в продвижении товаров и создании концептов для привлечения покупателей.

На этом обзор общей структуры заканчивается, а выглядит он следующий образом:



Рисунок 1 – Общая организационная структура

В ходе анализа моей организации было принято решение разделить организационные структуры из-за их обширности.

Ниже будут представлены все организационные структуры по отделам:

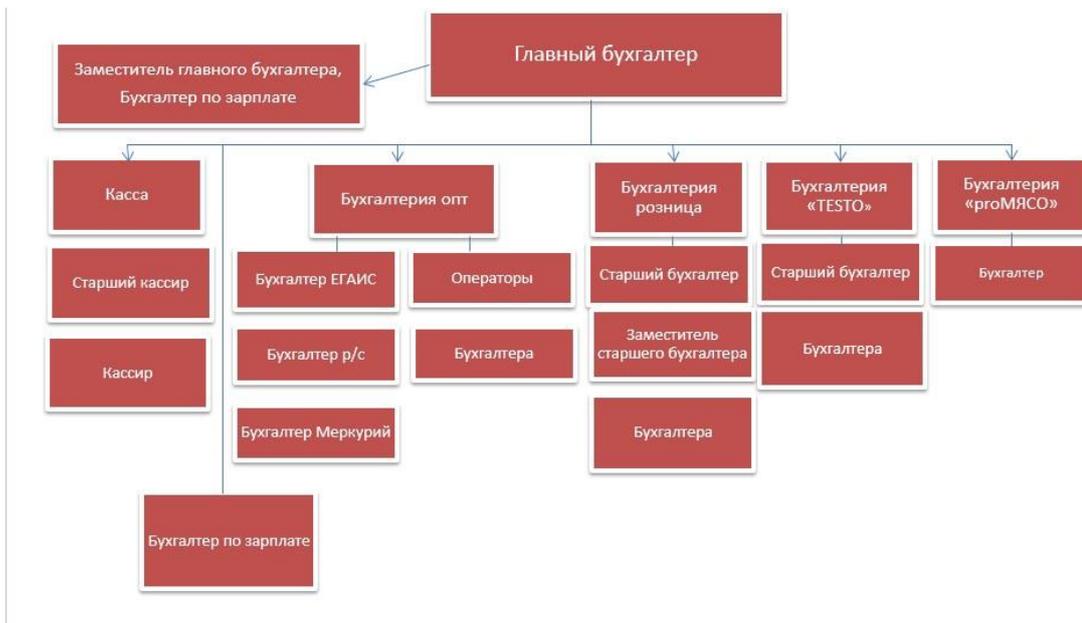


Рисунок 2 – Структурная схема отдела бухгалтерии



Рисунок 3 – Структурная схема отдела контроля

Так как этот отдел входит во внутреннюю структуру каждого ИП, я не стал включать его в общую структура, а просто вынес его отдельно.

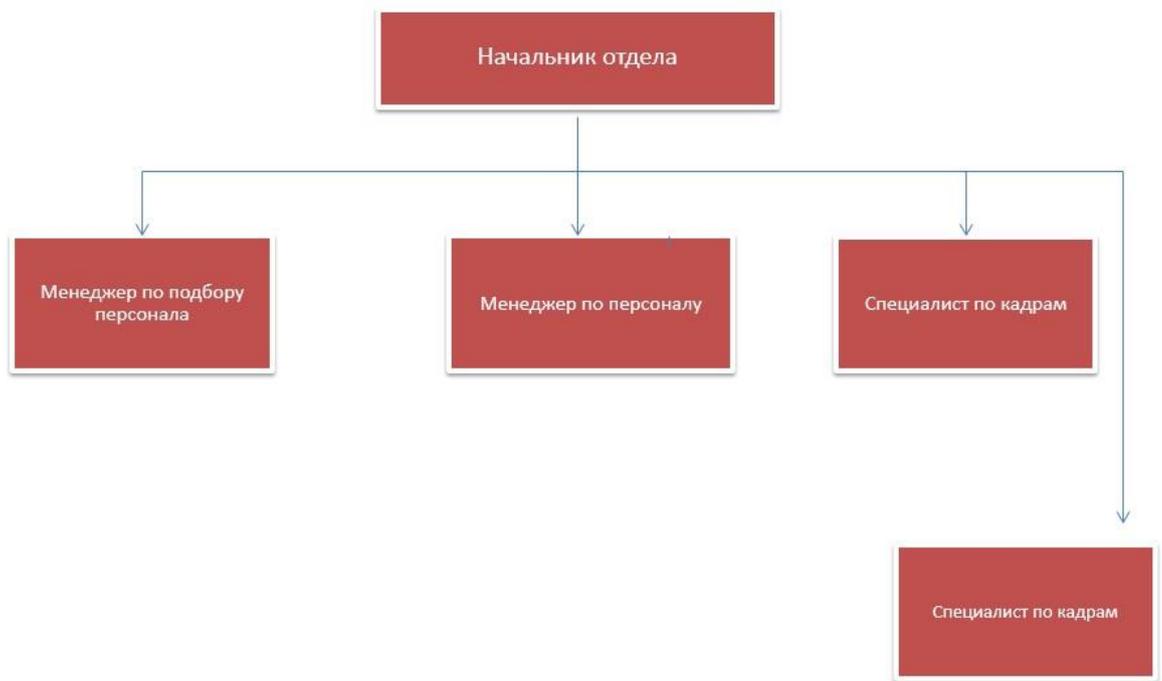


Рисунок 4 – Структурная схема отдела кадров



Рисунок 5 – Структурная схема IT отдела

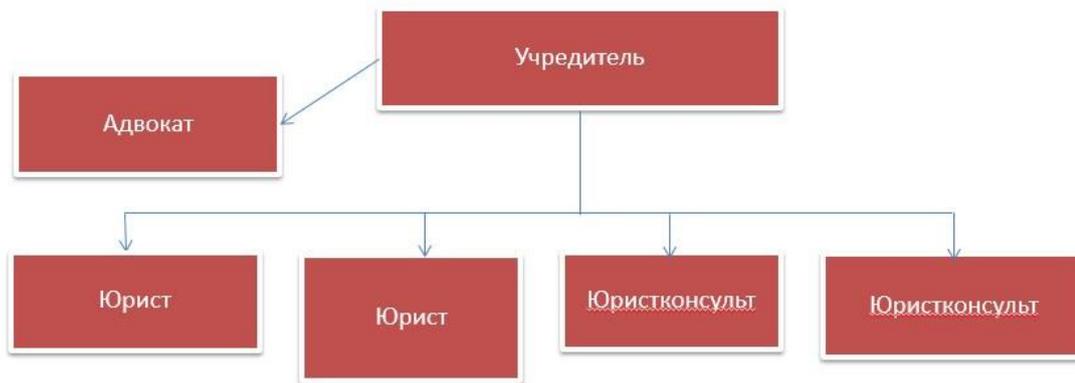


Рисунок 6 – Структурная схема юридического отдела

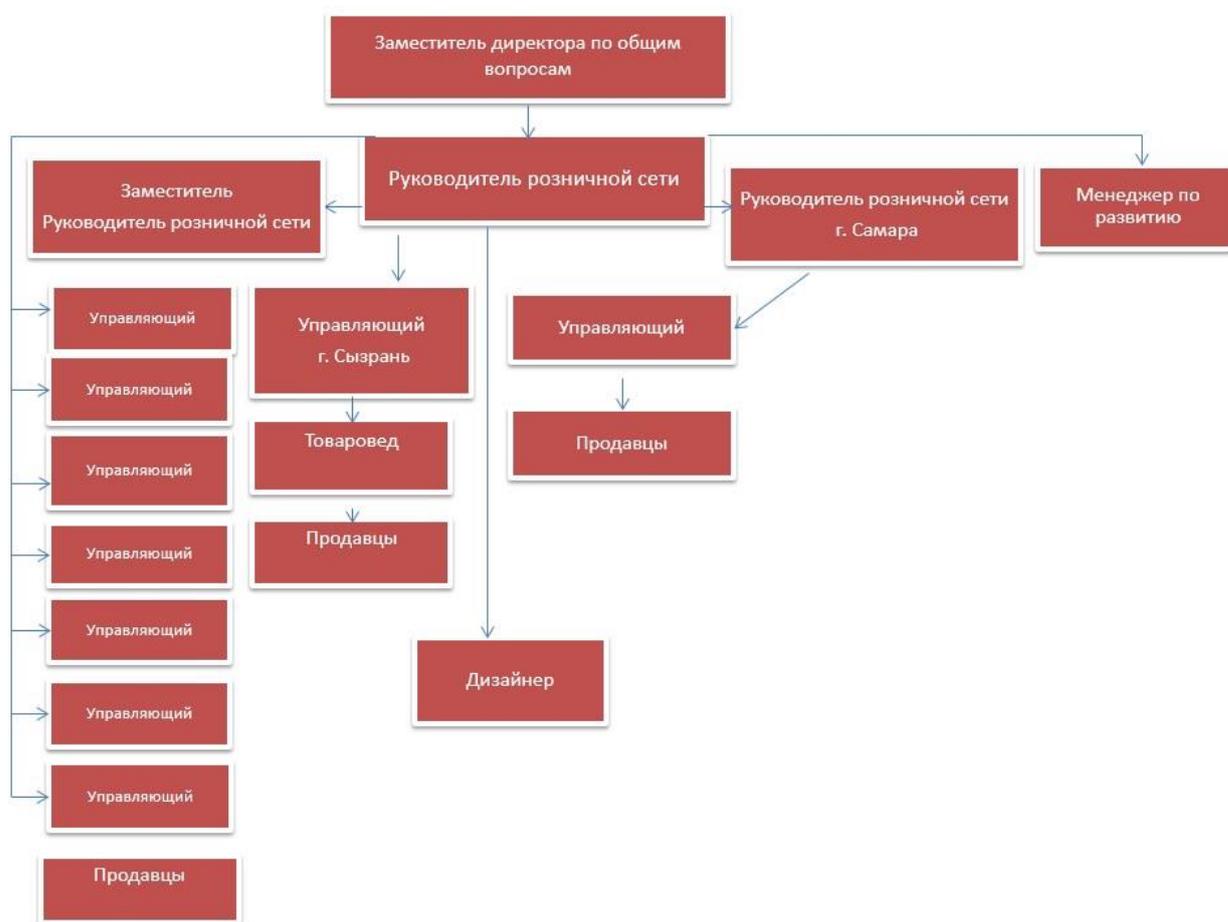


Рисунок 7 – Структурная схема розничной сети «ПИВО»

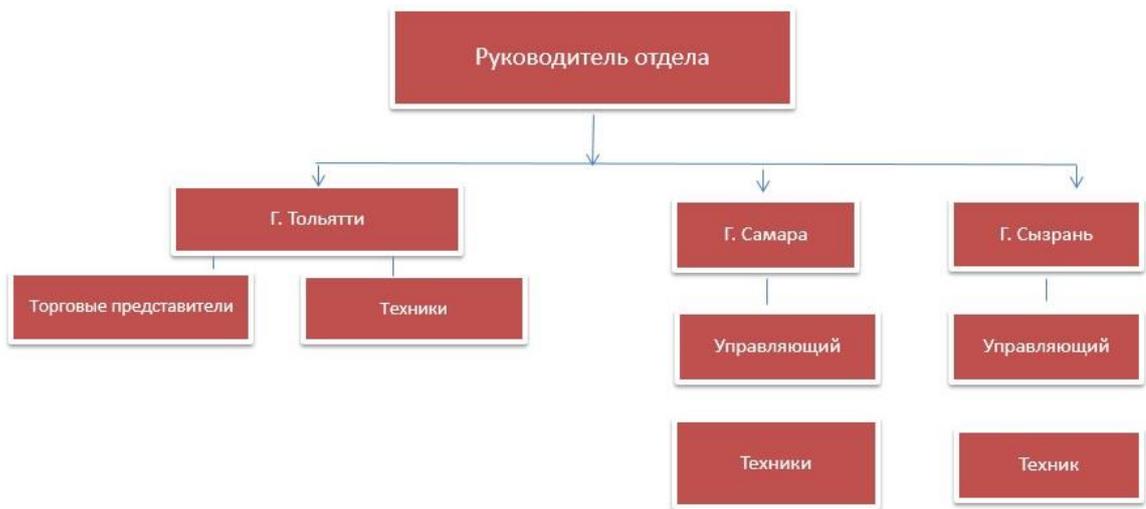


Рисунок 8 – Структурная схема оптовой сети «ПИВО»



Рисунок 9 – Структурная схема отдела закупок и логистики



Рисунок 10 – Структурная схема торговой сети пекарен

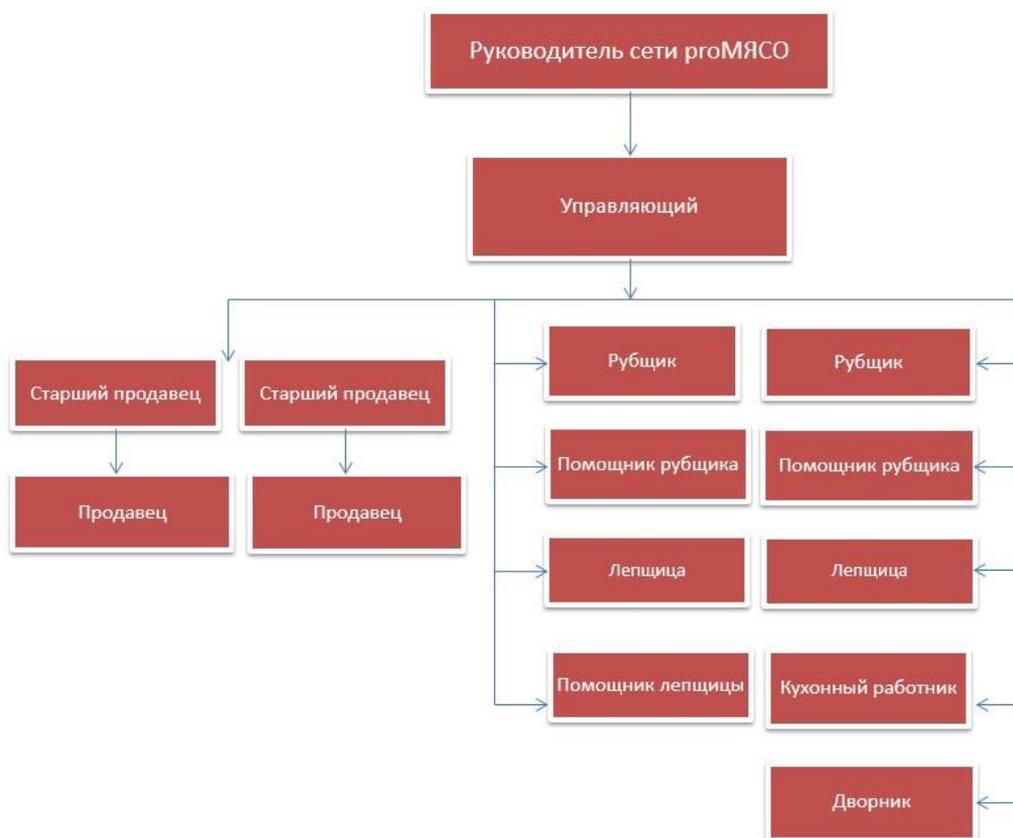


Рисунок 11 – Структурная схема торговой сеть мясных магазинов



Рисунок 12 – Структурная схема административно–хозяйственный отдел

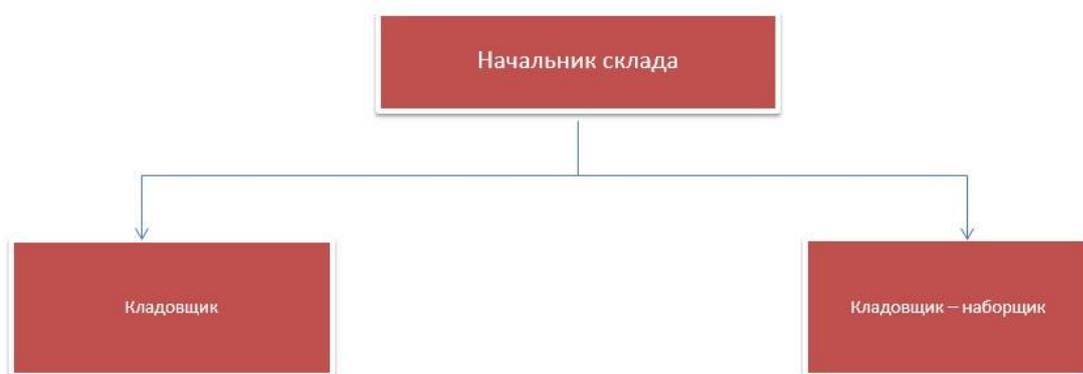


Рисунок 13 – Структурная схема склада

Это вся структура организации. Далее будем рассматривать документы и иные сущности, используемые при исполнении моделируемых бизнес–процессов и необходимые для моделирования документооборота, с описаниями их основного смысла.

В них входит:

- нормативные внутренние документы – это документы содержащий любую комбинацию рекомендаций, требований или правил, которым должен удовлетворять пакет программ. Эти документы являются основой нормативной базы организации. К ним относятся приказы, распоряжения, положения и другие виды документов;
- отчеты о выполнении работы – это исходящие документы для клиентов компании, которые содержат в себе регламент работ и стадии выполнения;

– договора – это документы, которые подтверждают некоторое соглашение сторон (заказчика и исполнителя), для определенных действий;

– технические задания – это документ, описывающий все нюансы и требования для проводимых работ. Оно составляется предварительно, перед началом всех работ.

Структура бизнес – процессов, отражающую их иерархию от более общих групп к частным бизнес–процессам представлена ниже:



Рисунок 14 – Структура бизнес–процессов по иерархии от общих к частным группам

На этом пояснение об организационной структуре закончили и теперь переходим к анализу предметной области и взаимодействию моей организации с выпускной квалификационной работой.

1.3 Провести анализ предметной области и взаимодействие практики с темой выпускной квалификационной работы

Анализ предметной области является важным аспектом каждого проекта, без него не один проект не может быть изучен и не одна задача не может быть построена. В работу практики входит изучение предметной области организации из чего складываются объекты исследования при разработке автоматизированной системы по теме самого ВКР. Данная практика позволит изучить и достичь нужных целей в исследовании и разработке того или иного проекта. В ходе чего будет производиться анализ проекта на тему «Разработка приложения для автоматизированного анализа отзывов клиентов» для данной организации. В ходе исследования предмет исследования может постоянно менять в зависимости от проекта, в моем случае предметом исследования было выбрано отдел, занимающийся проектированием веб-приложений. А внутри данной предмета был выбран узкий объект-исследования — это веб-разработка. Который будет взаимодействовать с моим проектом. Отдел же в свою очередь делиться на несколько групп, любой занимающиеся тем или иным видом деятельности. А сама структура данного отдела будет показана на рисунке 15

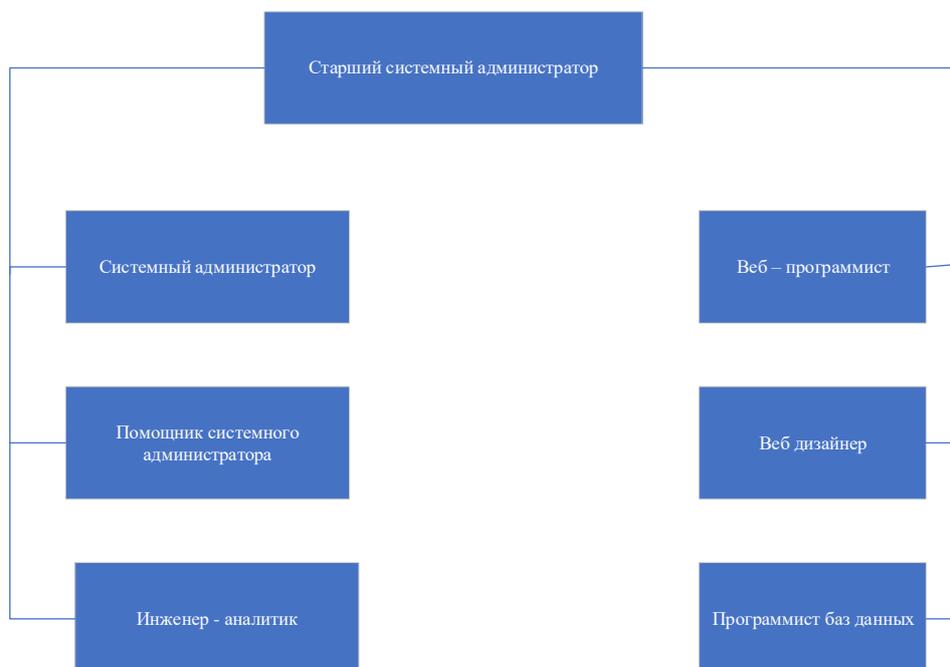


Рисунок 15 – Структурная схема IT отдела

Так же немного расскажу про сам отдел, чем он занимается и какие перспективы могут за пойти. Так как изучение данного предмета будет влечь за собой глубокое погружение в работу и полезность в изучении некоторых смежных сфер. Начало исследования будет происходить из описания вида деятельности каждой группы лиц, в ходе которых будет описано кратное описание его виды работы.

Начинается все со старшего системного администратора, который занимается руководством данного отдела, выставляет задачи и контролирует их работу. Далее идет системный администратор в его работу будет входить настройка и обслуживание различного сетевого оборудования и обычных персональных компьютеров, помощник при этом помогает ему в том, чего не успевает сам специалист.

В следующих рядах входит инженер аналитик, веб–программист, программист баз данных и дизайнер.

Инженер аналитик, производит расчет данных о рейтинге на сетевых ресурсах, собирает информацию о пользователей и клиентах, и производит актуальный план работы по улучшению сервиса. Веб программист разрабатывает сайты и их компоненты, производит компоновку различных

элементов для сайта. Дизайнер разрабатывает дизайн и структуру сайта. А программист баз данных, проектирует систему взаимодействия данных внутри сайта и других сетевых ресурсов.

Следующим шагом будет проектирование концептуальной модели.

Данная модель в большинстве случаев имеет высший приоритет так как процесс создает фундамент для проектирования проекта. Ниже будет произведены наиболее известные нотации моделирования.

В самые популярные концептуальные модели входят:

- UML;
- IDEF0;
- ARIS.

Каждый из данной технологии показывает свои плюсы и минус, а теперь поговорим отдельно о каждом.

UML – является распространенным графическим языком для описания объектно–ориентированного моделирования. Который очень тесно связан с разработкой программных продуктов.

UML модель в основном эксплуатируется для представления абстрактных моделей различных разрабатываемых систем.

IDEF0 – является методом функционального моделирования бизнес–процессов. И может представлять схему взаимодействия внутренних информационных потоков с показательной структурой их систем.

Напоследок представляю вам ARIS еще один метод для моделирования бизнес–процессов, который намного сложнее своих предшественников, из–за нестандартного интерфейса. Является разработкой немецкой компании Software AG.

Далее представляю вам таблицу критериев, по которым была выбрана методология предоставлена в таблице 1.

Таблица 1 Сравнительный анализ методологий

Критерии сравнения	UML	IDEF0	ARIS
Легкое изучение	Легкое	Легкое	Сложное
Удобство в установке	Легкое	Легкое	Среднее
Удобство использования	Среднее	Легкое	Сложное
Система хранения данных	Файловое хранение данных	Файловое хранение данных	Система объектной базы данных
Функции декомпозиции	Неограниченные	Неограниченные	Неограниченные
Подход к проектированию системы	Функциональный	Объектно–ориентированный	Процессный

После проведения анализа, который был показан на таблице 1, можно сделать определенные выводы, что метод IDEF0 является самых подходящим из предложенных и ее установка занимает меньше всего времени, по сравнению с остальными, по этой причине будет производиться разбор проекта по IDEF0 модели. При этом данная методология удобна в использовании и изучении. Следующим по порядку является UML, а после уже ARIS. Они оба сложнее первой нотации как в установке, так и в использовании и изучении. При этом часто возникают сложности в восприятии данных моделей.

Основным функция метода IDEF0 является предоставление графической схемы с описанием взаимодействия внутренних процессов при этом каждый ее блок описывает определенную функцию бизнес–процесса. При этом каждая из сторон имеет свои задачи.

Вывод главы 1

В данной главе были описаны цели и задачи моей организации, в которой я выполнял квалификационную работу. При описании цель и задач, я так же задумался и о реализации своих задач в квалификационной работе и это помогло мне их придумать и разделить их на этапы. Описал подробно структуру всех отделов, каждый отдел имеет свои плюсы и минусы, какие-то показали из себя полезными для реализации своего работы, какие-то не повлияли вообще не как. Но каждый отдел поднес хоть какую-то информацию по реализации моей работы. При этом всё было описано внедрение моей организации в мою выпускную квалификационную работу, где были описаны важные элементы по реализации программы. При этом была описана предметная область и реализовано подробное описание методологий со всеми плюсами и минусами, какая хороша для моей работы, а какую лучше не использовать. В последствии описания данных методологий было принято использовать такие методологии как IDEF0 и UML.

2 Теоретическое изучение темы квалификационной работы

2.1 Моделирование бизнес–процессов предметной области для постановки задачи автоматизированного варианта решения

В настоящее время каждый проект базируется на [основе 3 уровней](#) системы, концептуальную, логическую и физическую. Такой подход позволяет сократить время разработки и реализации любого проекта. А также существуют некоторые варианты совершенствования таких уровней:

– в первую очередь для совершенствования требуется произвести описание методов реализации процессов в той или иной организации, а вторым и пунктом к этому определение сложности в развитии;

– второй этап — это использование практического опыта, при постановке путей преодоления трудностей в процессе его проектирования, с применением корректировок в тех элементах, где трудностей и препятствия не обнаруживаются.

– последним этапом для совершенствования заключается в способе реализации процессов благодаря современным технологиям. При котором ввод данных в информационную систему повышает эффективность исследования.

2.2 Построить функциональную модель «как есть», «как должно быть»

После создания фундамента, начинается процесс проектирования проекта. Первым этапом служит строительство функциональной модели «как есть», а после «как должно быть».

[Моделирование бизнес](#) – процесса — это взаимодействие различных задач по средством выполнение действий, которые направленные на создание

определенного проекта. Визуально оно представляется в виде блок–схемы бизнес–процессов.

Модель бизнес–процесса «Как есть» представляет собой модель уже созданной или же с проектируемой системы, которая используется на данный момент. Данная модель показывает используемые в данный момент процессы и их потоки в данной информационной или же внутренней системе. В данной модели показываются слабые и сильные стороны организации и организационных процессов. Так же данную модель можно произвести с помощью CASE – средств с используемыми внутренними инструментами для показа ее внутренних потоков и процессов. При создании данной модели важно представить как можно подробнее реальные потоки бизнес – процесса.

Инструмент, который я буду использовать называется Allfusion Process Modeler.

После первичного ознакомления плавно переходим к построению данной модели с описанием основных первичных данных и самих процессов. На рисунке 16 будет показана контекстная диаграмма «Как есть».

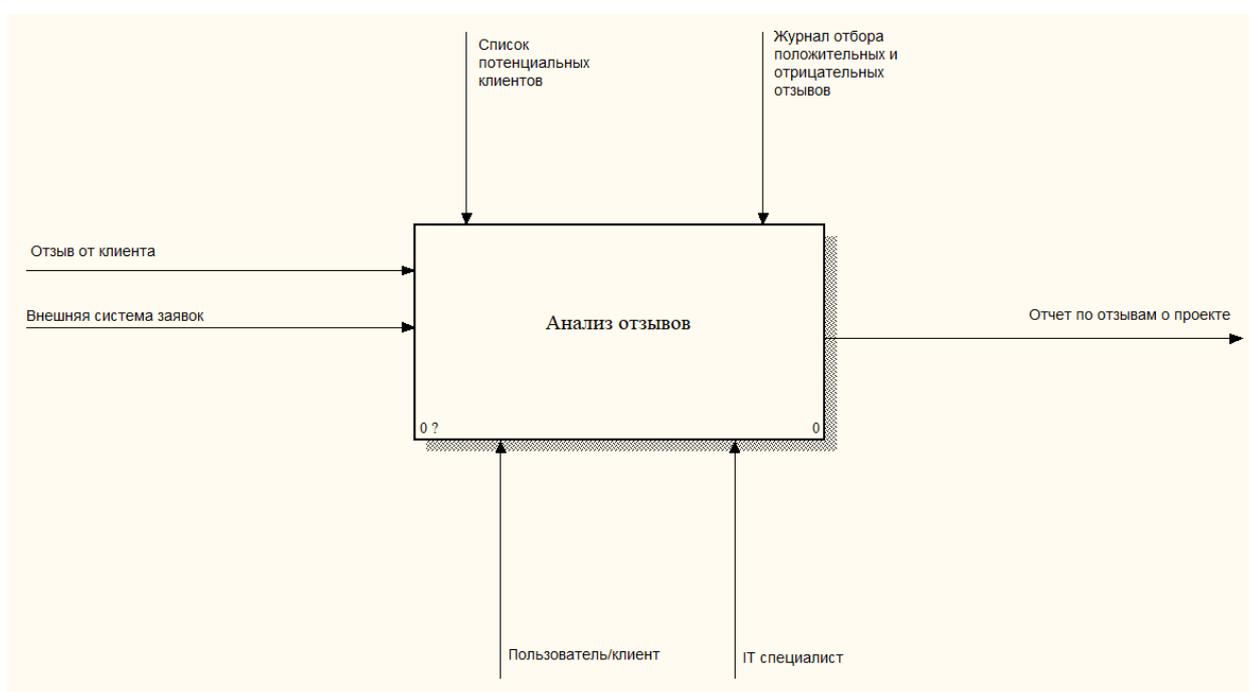


Рисунок 16 – Контекстная диаграмма «Как есть»

Давайте теперь рассмотрим более подробно процесс А0 «Анализ отзывов», на вход поступают такие входные данные как, отзыв от клиента,

система заявок из вне с информацией по клиенту и ее мнение о работе. На выходе получаем отчет по отзывам о проекте.

Следующим шагом для лучшего ознакомления с главным процессом разделим их на под процессы или же произведем его декомпозицию, которая будет показана на рисунке 17.

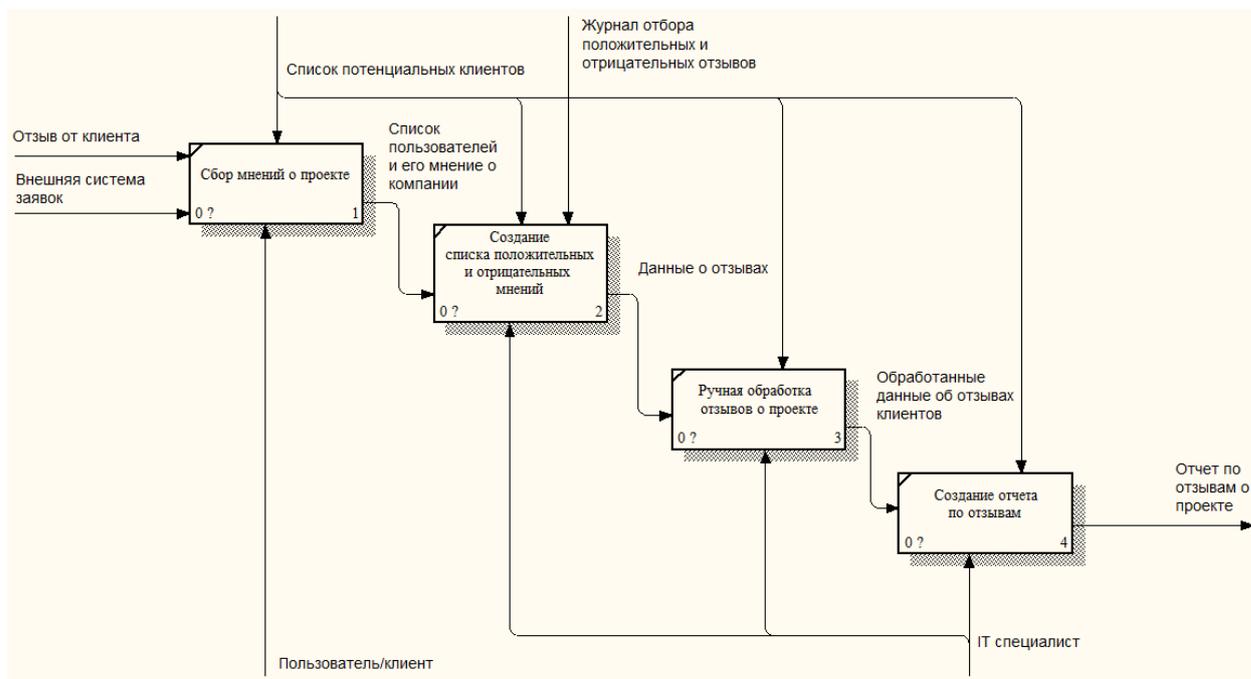


Рисунок 17 – Декомпозиция контекстной диаграммы «Как есть»

После представления декомпозиции контекстной модели «Как есть», рассмотрим отдельный процесс по подробнее:

- процесс А1 «Сбор мнений о проекте», в данном процессе имеется ввиду сбор различных записей, в том числе и комментарии к выполненным заявкам. Проводится тщательный отбор по мнениям высказанные от клиентов;
- процесс А2 «Создание списка положительных и отрицательных мнений», данный процесс прямо сообщает нам о том, что производится разделение между положительными и отрицательными отзывами о проекте;
- процесс А3 «Ручная обработка отзывов о проекте» подразумевает оценку каждого отзыва вручную, при этом проводится проверка на ложные отзывы;
- процесс А4 «Создание отчета по отзывам» в данном процессе приводятся выводы по полученным данным и создается отчет о рейтинге

проекта, приводятся примеры самых положительных и отрицательных отзывов.

По итогам выше указанных процессов, делаем выводы что процессы А1, А2, А3, А4 могут быть автоматизированы с целью упрощения и экономии времени на другие задачи, а также, уменьшение затрат на материальные нужды.

Теперь проведу декомпозицию некоторых основных процессов, требующих автоматизации, процесс А2 «Создание списка положительных и отрицательных мнений», А3 «Ручная обработка отзывов о проекте» которая будет показано на рисунке 18 и 19

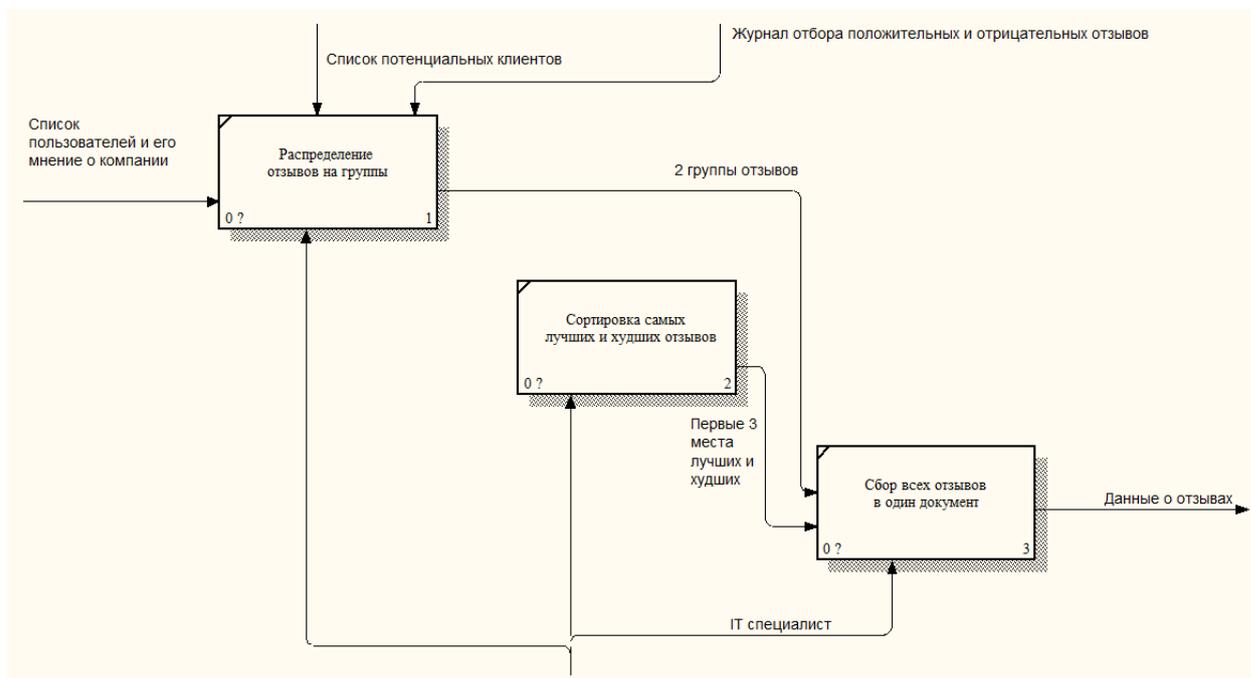


Рисунок 18 Декомпозиция процесса А2 «Создание списка положительных и отрицательных мнений»

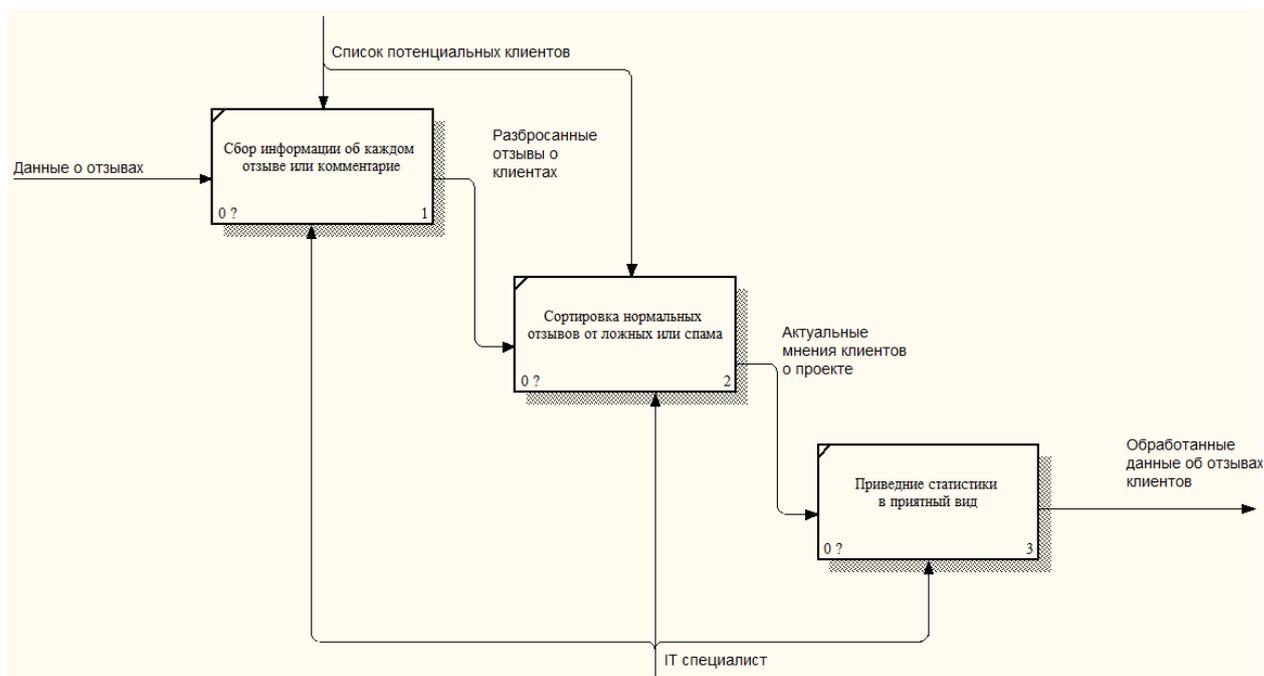


Рисунок 19 – Декомпозиция процесса АЗ «Ручная обработка отзывов о проекте»

Как только провели разбор концептуальной модели «как есть» перейду к построению модели «как должно быть». Построение данной модели делается на основе концептуальной модели «как есть» с именениями, которые улучшают и оптимизируют систему в проекте. Данный этап производит анализ структуры уже имеющегося проекта и производится устранения недостатков. В ходе которого производится автоматизация некоторых модулей и процессов. Из-за чего риски на разработку могут уменьшиться в разы. При описании данной концептуальной модели будут показаны изменения. Синим цветом будет отображаться элементы, которые больше не будут использоваться в системе, а красным, те, которые изменились или же добавились.

Ниже на рисунке 20,21 будет представлена концептуальная модель «как должно быть».

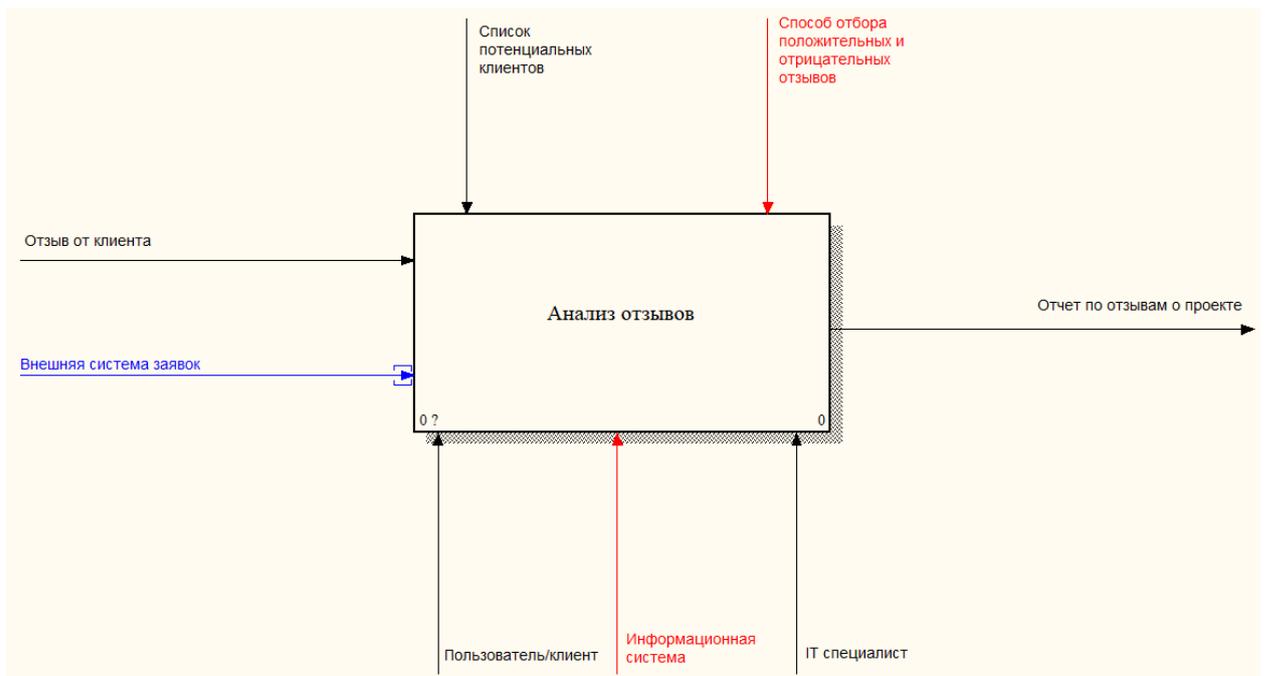


Рисунок 20 – Контекстная диаграмма «Как должно быть»

С помощью информационной системы IT специалисты смогут не только собирать информацию в любое время, но и разгрузить себя для выполнения других важных задач. Так как процесс сбора информации и фильтрации данных будет производить информационная система.

С информационной системой будут работать напрямую только IT специалисты, а косвенно пользователи и клиенты, так как обычные пользователи смогут только оставлять отзывы о проекте.

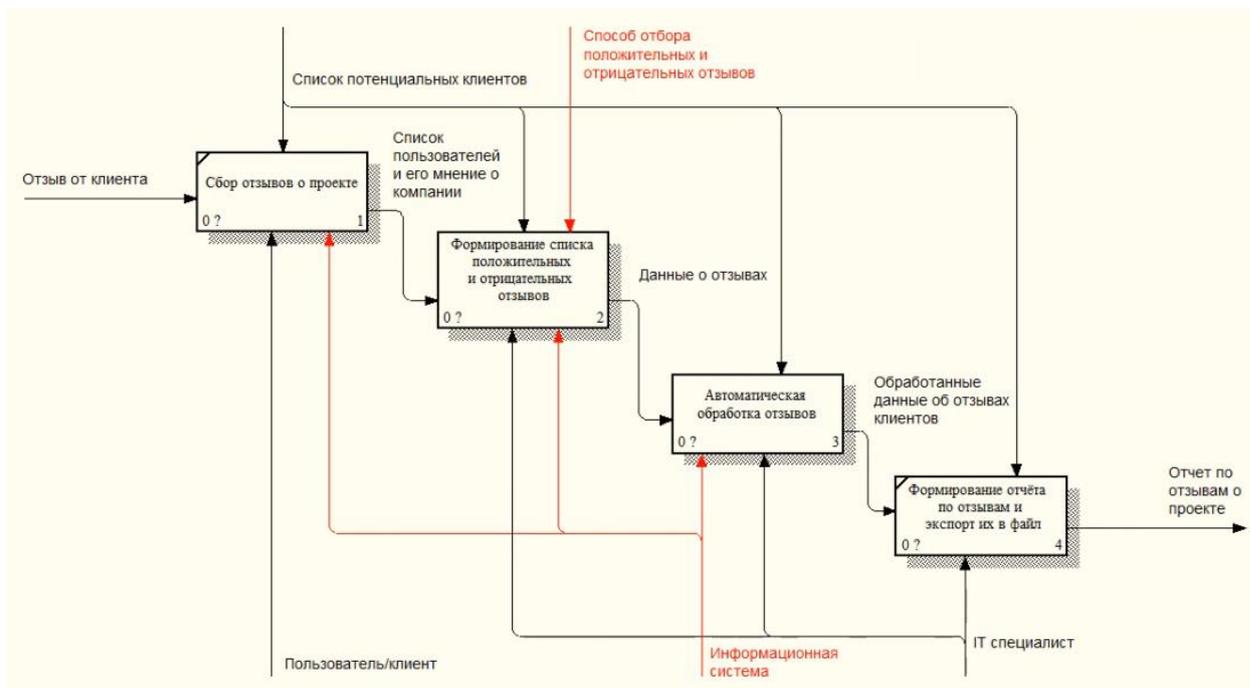


Рисунок 21 – Декомпозиция контекстной диаграммы «Как должно быть»

В декомпозиции контекстной модели «Как должно быть» видно, как изменилась система под воздействием информационной системы. В ходе чего вся информация, которая поступает обрабатывается самой системой, при этом не как не затрагивает ИТ персонал. А сами специалисты могу в простой форме сформировать отчет о уже обработанных данных.

2.3 Проанализировать схожих работ и обосновать выбор решения по оптимизации и реинжиниринге

В ходе проведения анализа процессов А2 «Создание списка положительных и отрицательных мнений» и А3 «Ручная обработка отзывов о проекте» входящие в главный бизнес–процесс А0 «Анализ отзывов» были выделены следующие недостатки:

- необходимость во в ручном исследовании и обработки мелкой работы в больших масштабах, при этом организовывать фильтрацию и правильный порядок построение отчетности;
- неэффективный документооборот, который выражается ведение списка потенциальных клиентов. Они постоянно меняются, и корректировка

данных может занимать длительное время, при этом актуальная информация всегда нужна для получения максимальной выгоды для проекта.

Ход разработки для каждого подобного проекта почти одинаковый. Отличия могут возникать только в ходе самой разработки. Так как отзыв это обычная поле, которое передает информацию на главный ресурс или же в базу, для последующего его анализа. Поэтому анализ других работ затратит много времени в пустую, из-за чего, показывать их я не буду.

Анализ производился исключительно по уже разработанным проектам и из них можно произвести некоторые выводы, о том, что для данного проекта требуется обязательная автоматизацию всех этапов.

Так как я выше описал недостатки, но плавно перейдем к требованиям для решения данного проекта. Для выявления определенных требований будет использоваться система модели FURPS+. Расшифровывается она как functionality usability reliability performance supportability. Каждая расшифровка в данной аббревиатуре, является требованием для реализации данного проекта. А знак + означает, что данная система имеет еще 4 дополнительных программы и которые делаться на несколько групп, таких как implementation, interface, design, physical.

На этом требования как должен работать проект сформирован. И теперь я плавно буду переходить к уже существующим проектам с описанием их работы. Проанализирую их и сделаю вывод по данному этапу.

2.4 Проанализировать существующие разработки для решения обозначенной задачи

Для анализа существующих разработок, нужно вначале определить некоторые критерии, по которым будет производиться выбор той или иной работы. Для этого я произвел анализ и выбрал критерии для будущего своего проекта, который будет включать в себя:

– работа с интерфейсом в системе не должна задавать вопросов и иметь простой функционал;

– информационная система должна автоматически собирать информацию с сайта;

– полученные данные должны обрабатываться и извлекать из них актуальную информацию о клиентах;

– информационная система должна иметь внутреннюю систему отчетов с выгрузкой необходимых данных на сторонний носитель;

– данная система должна быть мобильная, так как статистика о проекте будет формироваться в любое доступное время.

Такие критерии были поставлены мной для реализации своего проекта, из чего можно теперь перейти к сравнительной характеристике схожих работ.

Ниже будут представлены аналоги информационной системы глобального анализа отзывов в сети.

Первым в списке пойдет:

– яндекс метрика, его пользовательский интерфейс представлен на рисунке 1.7;

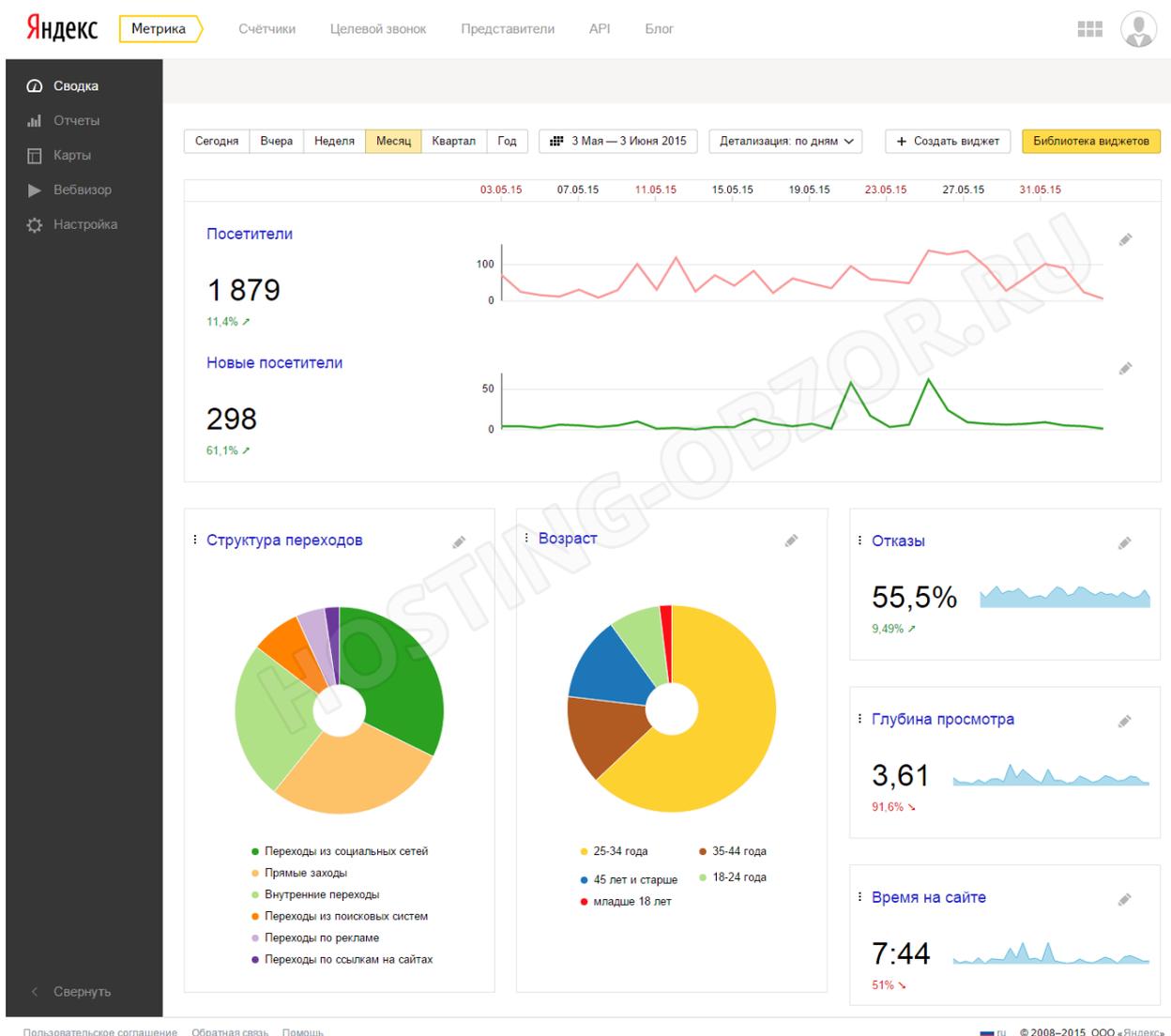


Рисунок 22 Главная страницы Яндекс метрики.

Она располагается в одном из разделов Яндекса, но также можно перейти по сетевому адресу <https://metrika.yandex.ru>, представляет собой сервис по сбору и аналитике данных поступающих с определенного сайта. Данный сервис обладает рядом преимуществ:

- широкое разнообразие различных параметров для анализа аудитории;
- проще сделана настройка целей;
- обладает широкими возможностями по представлению информации в виде графиков;
- повышенный способ анализа сервисов из трафика «Яндекса».

Но, к сожалению, имеет и ряд минусов, которые затрагивают следующие элементы:

- функционал является условно бесплатным;
- нет возможности сделать аналитику по положительным или же отрицательным отзывам.
- babkee, пользовательский интерфейс представлен на рисунке 23;

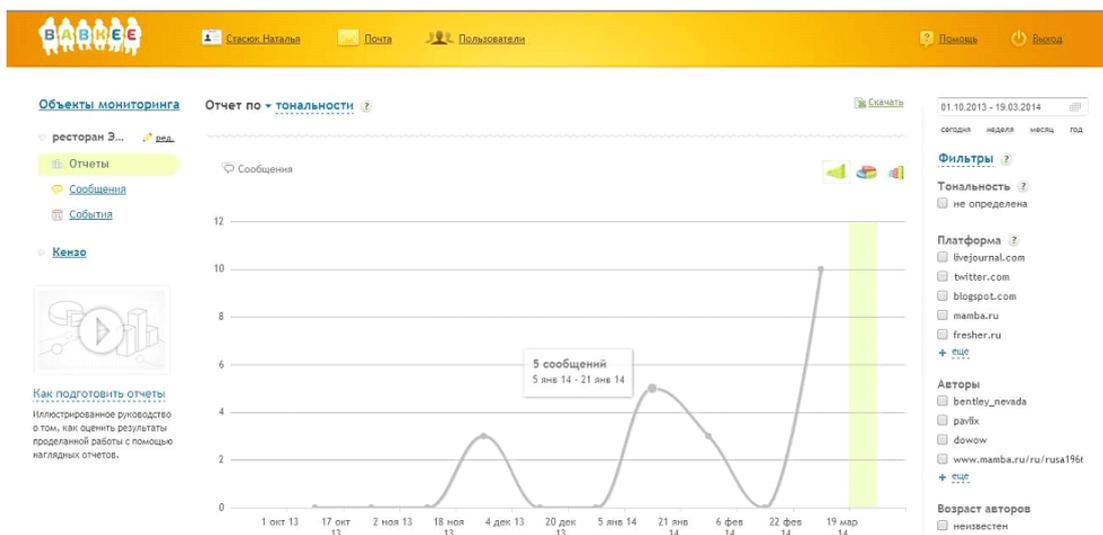


Рисунок 23 Страница мониторинга сервиса Babkee

Она располагается в открытом доступе и также на него можно перейти по сетевому адресу <https://www.babkee.ru/>, представляет собой сервис по сбору и аналитике данных, поступающих с определенного сайта. Из сервисов обладает рядом преимуществ:

- предоставление всего функционала совершенно бесплатно;
- уникальные параметры оценки значимости контента, в том числе и различных сообщений;
- формирование отчетности на основе данных об авторе сообщения;
- имеет настройку фильтров и умеет запоминать уже готовую настройку;
- позволяет экспортировать собранные данные в документы;
- умеет обрабатывать дубликаты.

Но также обладает некоторыми минусами:

- отсутствие разделение между положительными и отрицательными отзывами;

- отсутствует кроссплатформенность;
- мало количество графиков и нет функционала их настроек.
- IQBuzz, пользовательский интерфейс представлен на рисунке 24.

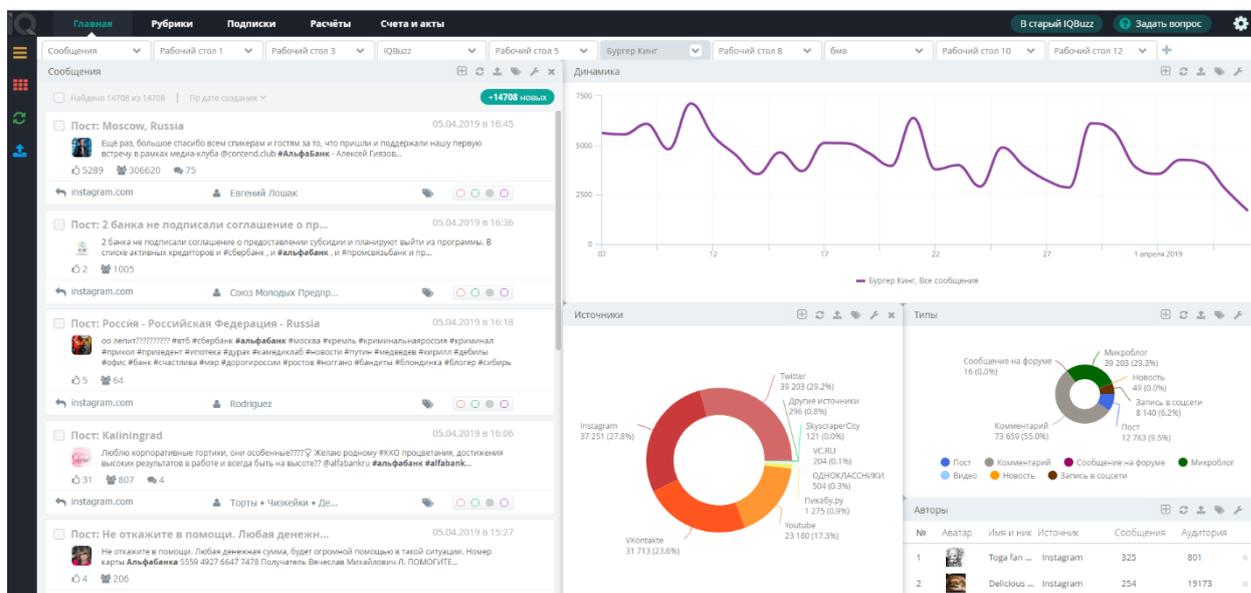


Рисунок 24 Общая страница мониторинга сервиса IQBuzz.

Она располагается в открытом доступе и также на него можно перейти по сетевому адресу <https://iqbuzz.pro>, представляет собой сервис по сбору и аналитике данных, поступающих с определенного сайта. Из сервисов обладает рядом преимуществ:

- имеет высокую аналитику в различных сферах, от анализа обычного подсчета пользователей, до глубокого анализа всего доступного материала;
- предоставление помощи в оказании различных услуг;
- имеет большой функционал в ряде графических диаграмм;
- имеет функцию сравнения с похожими проектами.

Из минусов могу сообщить:

- сложность в усвоении данного интерфейса;
- платная услуги данного продукта. К этому могу добавить еще и большая стоимость.

Результат анализа информационных систем по сбору и анализу данных представлена в таблице 2.

Таблица 2 – Сравнение существующих аналогов информационной системы.

Требования	Яндекс метрики	Babkee	IQBuzz
Просмотр общего списка отзывов	+	+	+
Распределение между положительными и отрицательными отзывами	–	–	–
Создание диаграмм	+	+	+
Экспорт в документы	–	+	–
Анализ данных пользователей/клиентов	–	+	+
Система уведомлений	–	–	+
Кроссплатформенность	–	–	–
Программный продукты	–	–	–
Сбор данных с сайта	+	+	+
Итого:	3/9	5/9	5/9

Исходя из разбора данной таблицы 2, делаем вывод, что каждая система имеет свои плюсы и минусы, но они не выполняют все критерии информационной системы.

После проведения сравнительного анализа делаем вывод, что необходимо спроектировать и разработать новую информационную систему, а также описать цель ее создания и основные требования к будущей информационной системы.

2.5 Описать функциональные требования к проекту и сформировать бизнес–цели и требования для ее реализации

Исходя из анализа проведенном в вышестоящем пункте, делаем вывод о том, что ни одна система не может подходить необходимым требованиям. Из чего складывается задача спроектировать информационную систему анализа отзывов.

Сформулируем теперь цель и основные требования к будущей информационной системы.

Целью создания информационной системы является предоставление простого и удобного инструмента с анализом отзывах о клиентах/пользователях в веб варианте.

Для достижения поставленной цели были сформированы некоторые требования к будущей информационной системе:

- интуитивно понятный интерфейс;
- выгрузка данных в отдельный файл;
- сбор, обработка и распределение отзывов по группам;
- формирование отчета о рейтинге сайта;
- возможность интеграции с системой личного кабинета;
- кроссплатформенность.

Теперь опишу каждый пункт требования в отдельности, почему я выбрал именно их и для чего ни служат.

Первый из списка идет интуитивно понятный интерфейс. Каждый интерфейс для пользователя должен быть простым и удобным, чтобы не было вопросов даже у человека, который не разу не сидел за компьютером. Далее идет выгрузка данных в отдельный файл, данный критерий служит для удобства анализа данных. Сбор и обработка является главным критерием, без него не может быть не какого анализа данных. Потом проект должен иметь возможность сформировать отчет об имеющейся информации. Так же должны быть личные кабинеты, для распределения доступов к тем или иным объектам.

И сам проект должен работать на всех устройствах. На этом требования к проекту описаны и перейдем к следующему этапу разработки.

2.6 Описание и обоснование решения по системной архитектуре проекта

На сегодняшний день существуют множество архитектур, которые обладают различными полезными функциями, и каждая из них используется в различных целях. Основой для любой информационной системы является архитектура клиент – сервер. Данная архитектура разделяется на несколько групп, из-за чего каждая группа выполняет свои конкретные задачи, при этом они хорошо взаимодействуют друг с другом. Архитектура клиент–сервер будет показана на рисунке 25.

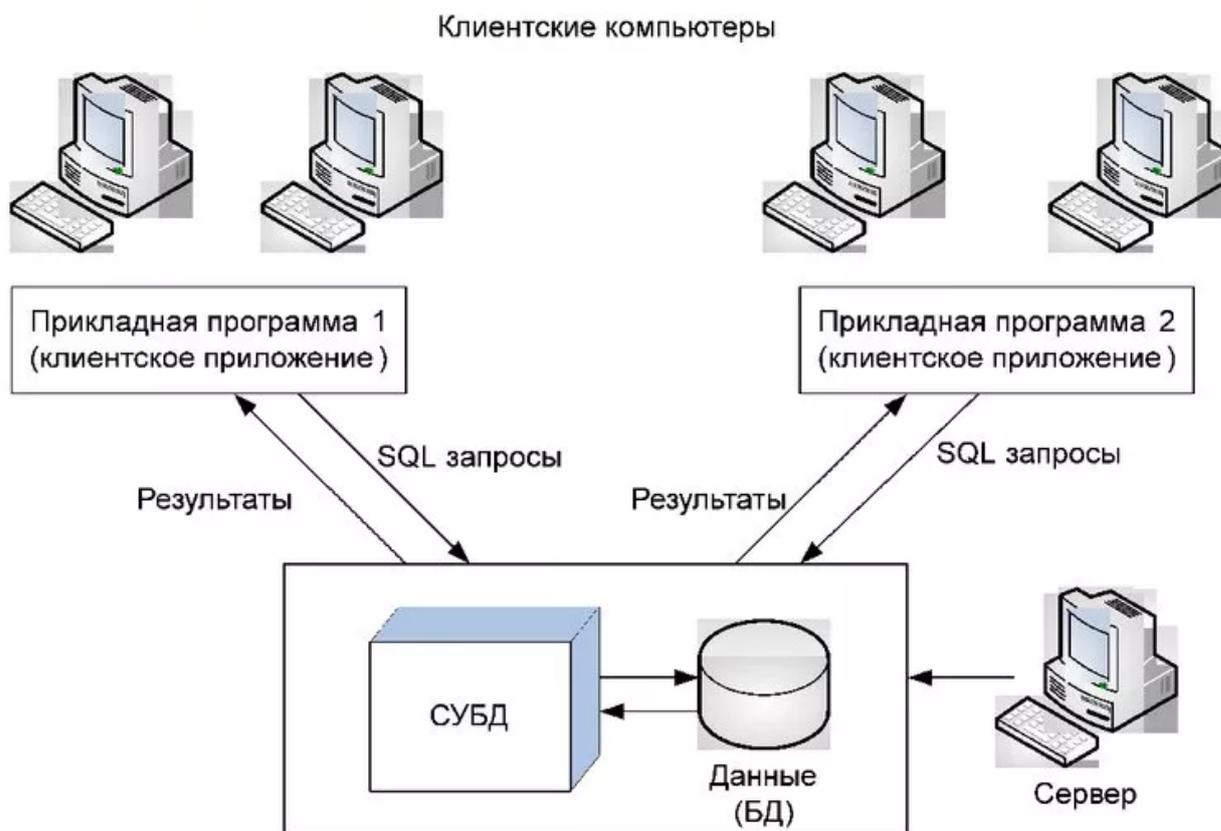


Рисунок 25 – Архитектура клиент–сервер

Теперь перейду к описанию этих групп, чтобы поближе познакомиться с ними. Архитектура клиент–сервер разделяется на 3 группы. Первая является клиентом, вторая группа является сервером, и третья будет базой данных.

Первое группа клиент – она представляет программный продукт или же сам персональный компьютер, который посылает запросы на их обработку и получения.

Вторая группа сервер – Это физическое оборудование, которое принимает и обрабатывает данные с помощью множества алгоритмов, которые были заложены в данной оборудовании и использует множества методов шифрования для лучшей защиты.

Третья группа база данных – Представляет собой хранилище для записи тех или иных данных. При этом все управление происходит под СУБД.

Данная архитектура довольно гибкая и позволяет использовать ее во многих областях, а вариативность настроек позволит реализовать даже сложные проекты при очень хорошей защите информации. Это архитектура так же обладает высокой скоростью обмена данных, при всем этом технические характеристики клиента могут быть ниже по многим показателям. По этой причине была выбрана данная архитектура.

Вывод главы 2

Во второй главе были спроектированы диаграммы как «Как есть» и «Как должно быть». В них очень конкретно показаны и объяснены различия до и после введения автоматизированной информационной системы. Были представлены так же готовые инструменты для аналитики многих данных такие как количество посещения, возраст людей, их интересы и тому подобное. К сожалению, не одна из данных инструментов не повлекло за собой нужных мне интересов, поэтому по требованиям они не подошли. По этой причине я решил разработать свое программное обеспечения для соблюдения требований. При всем при этом была выбрана качественная архитектура АИС и она будет иметь вид клиент–сервер. Такие решения позволит не только развить мои навыки, но и углубиться в работу архитектуры и этих сервисов, так как они будут являться примером для разработки собственной.

3 Постройка и описание информационной модели

3.1 Построение концептуальной схемы приложения

Приложение будет происходить в веб варианте, по этой причине каких–либо глубоких погружений в построение схем не будет.

Для реализации данного проекта нужно спроектировать концептуальную схему проекта. Концептуальная модель моего проекта будет показана на рисунке 26.

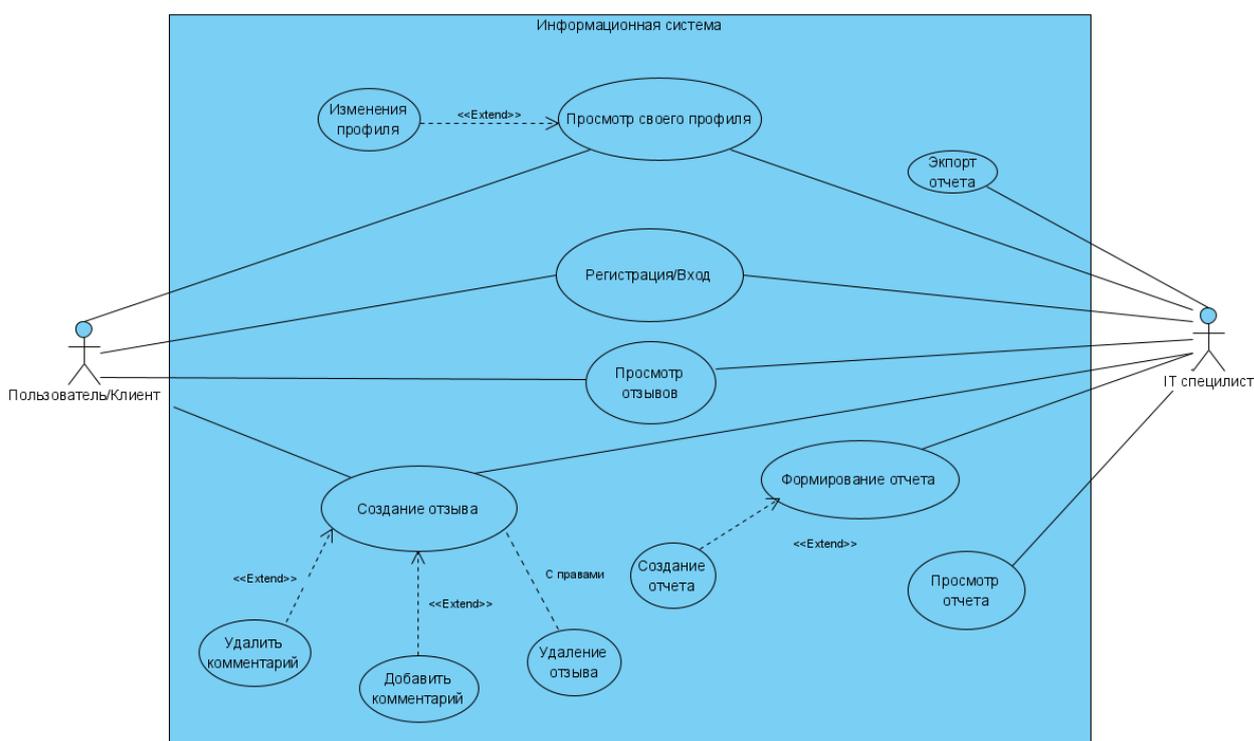


Рисунок 26 – Концептуальная схема приложения

На рисунке 3.1 представлены два актера, или по–другому действующие лица:

– IT специалист – пользователь веб–приложения с особыми правами, который анализирует данные, собранные при помощи информационной системы и формирует отчеты. Также занимается модерацией самих отзывов.

– пользователь/клиент – пользователь веб–приложения, который оценивает данный сайт и оставляет на нем отзыв о данном проекте, при этом может добавлять других пользователей в свой список контактов для будущего

их переписывания, и оставлять комментарии как над своим отзывом, так и под другими. При этом участвуя в социальной коммуникации с другими пользователями.

В таблице 3 отражена характеристика прецедентов, которые были описаны в диаграмме вариантов использования.

Таблица 3 – Краткая характеристика прецедентов

Прецедент	Характеристика
Регистрация/Вход	Регистрация и вход пользователя в системе с последующим созданием его личного профиля с определёнными правами.
Создание отзыва	Пользователь вносит данные и отправляет их в базу с последующим показанием внесённых данных на страницу
Добавление комментария	Пользователи дополняют уже имеющуюся запись, новыми, при помощи отдельной вкладки для комментариев.
Удаление комментария	Пользователь может свободно удалить свой комментарий при помощи соответствующей кнопки.
Удаление отзыва	Пользователь с определёнными правами может удалить свой отзыв.
Просмотр отзывов	Любой пользователь может просмотреть все выложенные отзывы на странице.
Просмотр профиля	Пользователь просматривает свой профиль.

Продолжение таблицы 3

Изменение профиля	Пользователь может изменять свои личные данные в своем профиле.
Формирования отчета	Для IT пользователей имеется отдельная вкладка с отчетами, для сбора анализа по отзывам.
Создание отчета	Отчет автоматически создается при выполнении определенной команде от IT пользователя
Просмотр отчета	Только IT пользователь может просматривать отчеты об отзывах
Экспорт отчета	Любой отчет, который формируется может быть сохранены в файл на компьютер.

После составления концептуальной схемы я перехожу к следующему этапу. Это выбор программного обеспечения для реализации темы ВКР.

3.2 Разработка программ для реализации алгоритмов обработки данных

[Для разработки программного обеспечения информационной системы](#) требуется для начало выбрать инструмент разработки. В инструменты входят такие среды разработки как: Visual Studio, eclipse, Code::Blocks и текстовые редакторы visual code, Notepad++. Их так же называют IDE системами.

Ниже я буду кратко описывать перечисленные программы, чтобы из них выбрать более удобную и функциональную для своего проекта.

Первым из приведенных это Visual Studio – интегрированная среда разработки, которая может похвастаться своими широкими возможностями к расширению за счет плагинов, данная среда разрабатывается известной

компании Microsoft. Имеет большой функционал по созданию различного программного обеспечения от консольных до веб – разработок.

Второй кандидат является eclipse – популярная на данный момент среда разработки, которая находится в открытом доступе и является фаворитом в языке программирования JAVA.

Третьей средой будет популярный Code::Blocks, еще одна свободная среда разработки, но ее плюсы это кроссплатформенность, и популярная за счет различных сторонних подключаемых модулей.

Далее идет текстовые редакторы одним из кандидатов является всем известных Notepad++ – это текстовый редактор для windows, имеет простой интерфейс и функционал, но имеет функцию подключения различных плагинов.

И последним идет visual code – Новейшая разработка от Microsoft, представляющий себя текстовым редактором с функционал подключения различных плагинов и встроенным управлением git control. При этом является кроссплатформенным.

После описание я провел анализ в табличной форме для выбора подходящей среды разработки, она будет показана на таблице 4.

Таблица 4 – Сравнительный анализ технологий разработки программного обеспечения

Критерии оценки	Visual Studio	Eclipse	Code::Blocks	Notepad++	Visual Code
Кроссплатформенность	–	–	–	+	+
Бесплатность	–	+	+	+	+
Быстродействие	–	–	+	+	+
Подсветка кода	+	+	+	+	+
Авто дополнение кода	+	+	+	+	+
Навигация по коду					
Подсветка ошибок	+	+	+	+	+

Продолжение таблицы 4

Отладка кода	+	+	+	–	+
Внутренний эмулятор	+	–	–	–	+
Всего	5/8	5/8	6/8	6/8	8/8

Исходя из таблицы 4, я сделал вывод, что текстовые редакторы не обладают нужным количеством функций, а многие из них являются платными и не могут предоставить больших возможностей. Поэтому я выбрал достаточно популярную и удобную среду разработки visual code. Является очень удобным инструментом для реализации многих проектов. Следующим шагом нужен выбор самой СУБД, а именно программу, которая будет хранить всю имеющую информацию. MySQL, Microsoft SQL server, Oracle, PostgreSQL, MariaDB, Microsoft Access.

Немного опишу каждую из них.

Для этого я провел ряд исследований по некоторым вариантам. В них входит такие СУБД как:

MySQL – Из всех популярных видов баз данных, MySQL является самой популярной из всех что есть на данный момент, по ней имеются очень много инструкция и документации, что делает ее очень легкой в усвоении. При этом она имеет большой функционал и высокую гибкость в работе с данными.

Microsoft SQL server – Данная СУБД разработана от компании Microsoft. В основном используется для запросов Transact– SQL. Данные запросы являются расширенным языком обычного SQL от компании Microsoft.

Oracle – Это СУБД разработанная компанией Oracle, она поддерживает множество технологий для реализации объектно–ориентированного подходов к организации баз данных. И имеет высокую популярность в этой сфере.

PostgreSQL – Одна из самых конкурирующих СУБД является как раз PostgreSQL, по причине того, ориентирована на полное соответствие стандартам SQL. И часто используется в коммерческих целях.

MariaDB – Является ответвлением от системы управления данных MySQL. Данная СУБД является гибридом таких СУБД как Oracle и MySQL.

В ходе обдумывания какую СУБД выбрать были сформулированы некоторые основные критерии, по которым был сделан выбор СУБД:

- кроссплатформенность;
- бесплатность;
- быстродействие;
- удобное использование и легкое обучение;
- наличие документации по работе с СУБД
- распространённость.

Результат данного анализа и выбора будет представлен на таблице 5.

Таблица 5 – Сравнительный анализ СУБД

Критерии Оценки	MySQL	Microsoft SQL server	Oracle	PostgreSQL	MariaDB
кроссплатформенность	+	–	–	+	–
бесплатность	+	–	–	–	+
быстродействие	+	+	+	+	+
удобное использование и легкое обучение	+	+	+	–	+
наличие документации по работе с СУБД	+	+	–	–	+
распространённость	+	+	+	+	–
Общая сумма баллов:	6	4	3	3	4

По этой таблице видно, что многие СУБД, имеют узкий спектр возможностей, а некоторые могут использоваться везде, но не имеют нужного функционала. А порой стоят достаточно дорого. По этой причине я выбрал самый распространённый и бесплатный вариант MySQL. Который будет

стоять на хостинге и обрабатывать всю имеющую информацию. О хостинге поговорю чуть позже.

3.3 Разработка интерфейса под все шаблоны

Разработка интерфейса является важной частью каждого проекта. В ходе этого этапа я буду представлять различные схемы и прототип основных шаблонов, которые будут использоваться в данном теме.

Интерфейс будет представлен веб варианте из-за чего функционал и вид данного проекта будет представлен в виде веб странице со множеством функций. Каждый элемент данной странице будет выполнять ту или иную работу и многие элементы будут взаимодействует друг с другом.

Разработка интерфейса будет проходить в несколько этапов:

- анализ уже имеющихся разработок;
- создание и анализ макета;
- внедрение функционала в макет;
- тестирование и исправление макета.

Это основные функции, которые потребуется в создании интерфейса.

Каждый интерфейс имеет так же определенные критерии для ее разработки в моем случае, интерфейс должен обладать компактностью, простотой, не иметь лишней информации на экране. При этом сам интерфейс должен иметь частичную анимацию.

Перейдем теперь к проектированию самого интерфейса. Анализ интерфейсов был уже произведен в пункте 2.4. По этой причине я не буду его показывать, а кратко опишу результаты. Многие интерфейсы, которые я анализировал, имел вкладку входа, страницу аналитики и конечно меню с переходом к той или иной вкладке. Поэтому была спроектирована схема, нескольких страниц, которые покажут функционал той или иной вкладки. Ниже на рисунке 27, 28, 29, 30, 31 будут представлены схемы основных страниц.

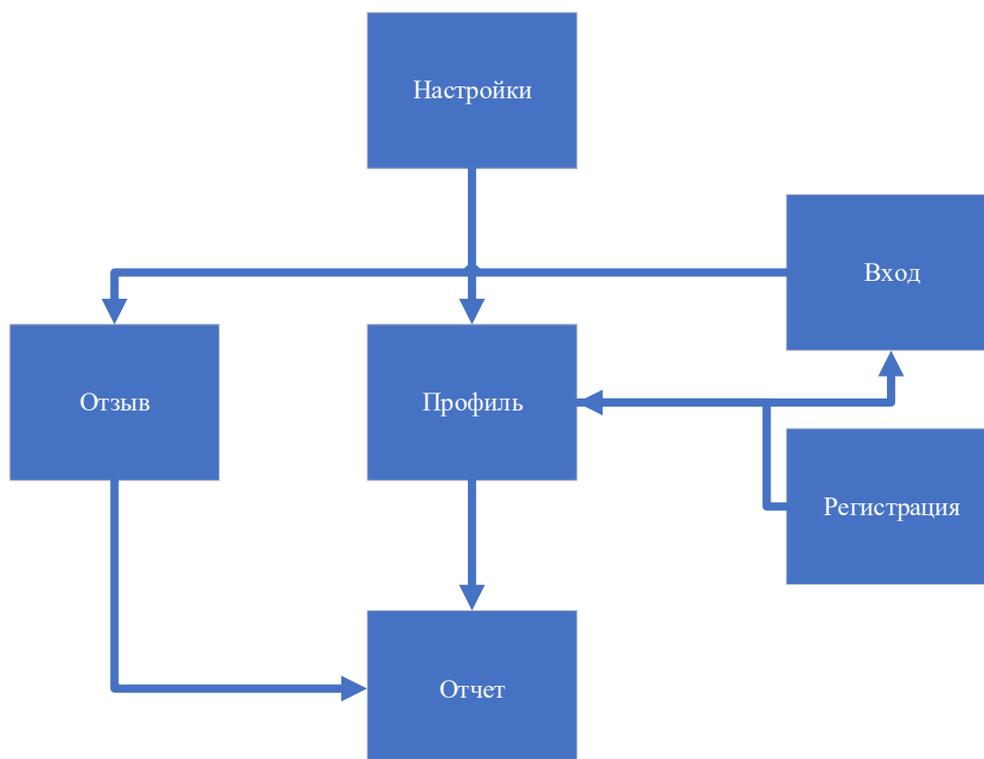


Рисунок 27 – Схема взаимодействия основных модулей в АИС

Данная схема представляет все виды страниц, которые будут показаны в данном разделе, а описание этой схемы будут представлены в пункте 3.2.4.

Теперь покажу разработанный интерфейс по каждому шаблону.

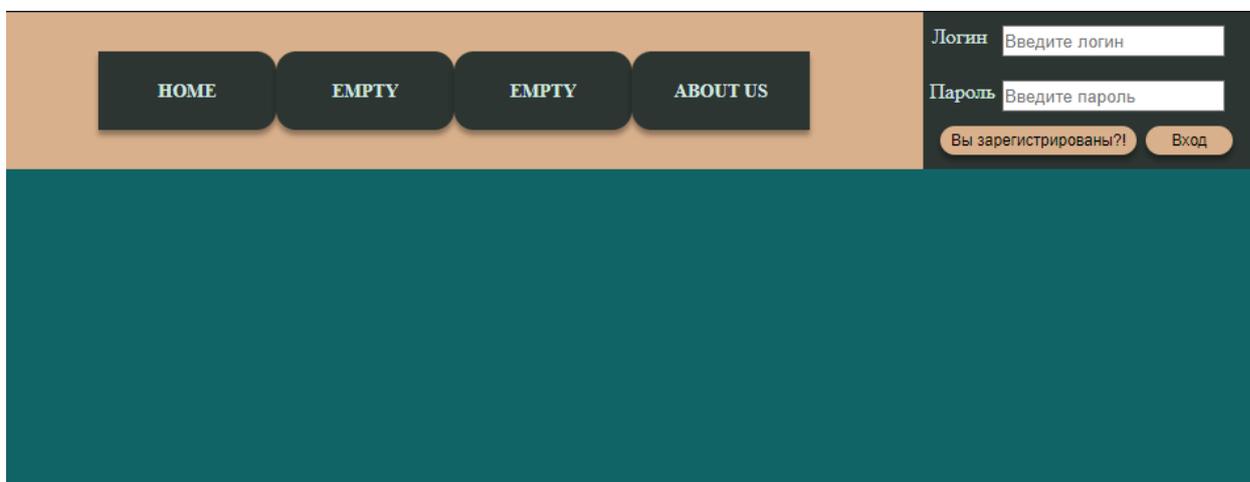


Рисунок 28 – Главная страница

Так выглядит интерфейс главной страницы, из которых показаны некоторые модули АИС.

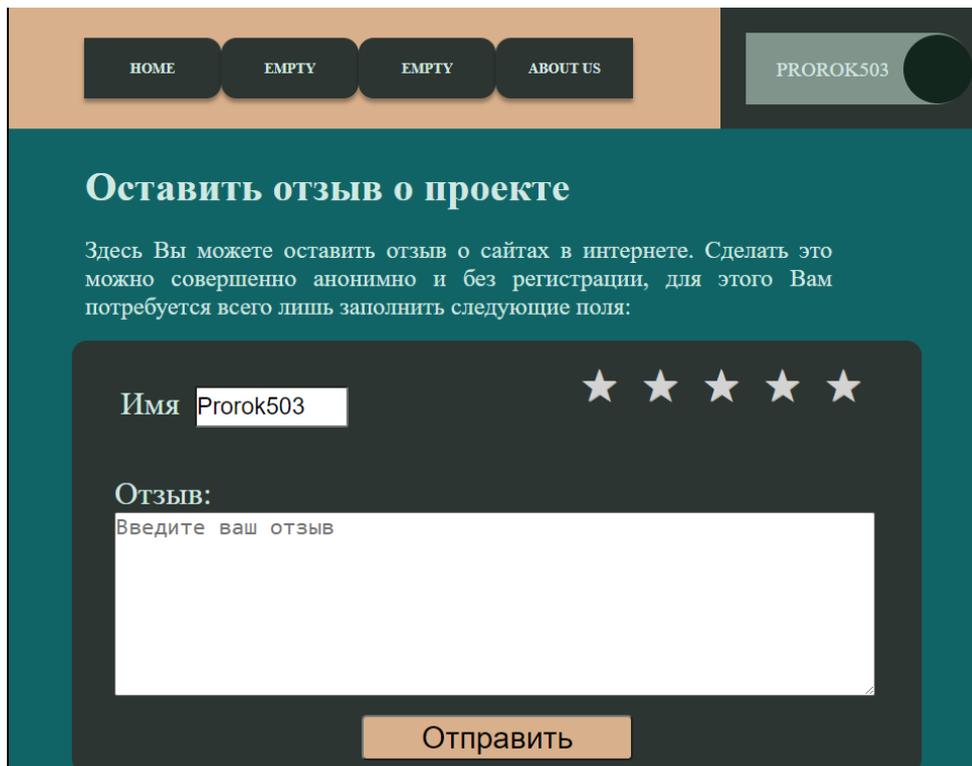


Рисунок 29 – Страница с оставлением отзыва

Здесь представлена страница с отзывами, который имеет поле для ввода нужных для оставления отзыва данных. Ниже будут представлены оставшиеся модули данного проекта, такие как профиль и отчет.

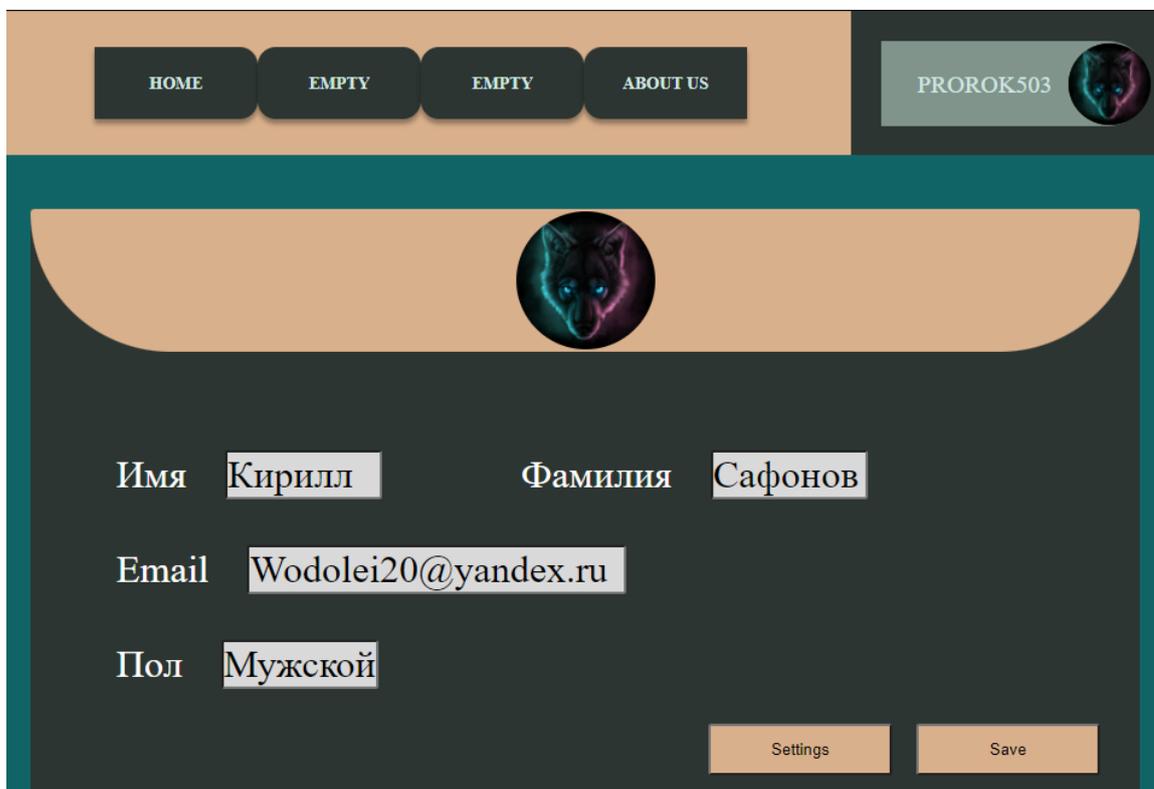


Рисунок 30 – Страница профиля пользователя

The screenshot shows a web application header with navigation buttons: HOME, EMPTY, EMPTY, and ABOUT US. On the right, there is a user profile for PROROK503 with a circular avatar. Below the header, there is a button labeled 'Сформировать' (Generate) above a table with the following structure:

Дата	Кол-во М.	Кол-во Ж.	Кол-во положительных	Кол-во отрицательных	Всего отзывов
Данные	Данные	Данные	Данные	Данные	Данные
Данные	Данные	Данные	Данные	Данные	Данные
Данные	Данные	Данные	Данные	Данные	Данные

Рисунок 31– Страница с отчётом об отзыве

Начальная оболочка программы была сделано, теперь перейдем к описанию ее модулей и походу к формированию базы данных.

3.4 Описание модулей приложения АИС

Каждый модуль имеет свой функционал и обращается с каким-либо запросом, или же передает определенные данные для работы другого модуля. Перейдем к описанию основных модулей, все модули представлены как структура веб страниц:

- профиль – представляет собой страницу с данными о зарегистрированном пользователе, в котором будут хранится такие первичные данные как имя, фамилия, пол, пароль, емейл. При этом данный модуль имеет внутренний функционал редактирования данных.

- регистрация – представляет собой небольшой фрагмент с внесением первичных данных от пользователя и отправкой их в информационную систему для дальнейшей обработки и записи этих данных в информационную таблицу, каждая запись является уникальной и при повторном внесении тех или иных главных элементов выдает ошибку о существующей записи.

– вход или же авторизация – в данном модуле происходит валидация данных и процесс входа в саму систему. Данные считывают из информационной системы и происходит вход благодаря чему, открывают некоторые функция, которые доступны только зарегистрированным пользователя, например редактирование профиля.

– настройки – представляет собой небольшой блок с уже имеющейся информации о зарегистрированном пользователе, и определенных функциях его личного профиля В данному блоке может происходить изменение личной информации или же данных для входа в систему.

– отзывы – представляет собой модуль, который позволяет пользователю оставить мнение о проекте, при авторизации, добавляет новые функции на подобии изменения или же удаление отзыва.

– отчёт – данный модуль является чисто информационным и представляется в виде таблицы с определенными данными. Каждый специалист IT может использовать данный отчёт и после проанализировать его. Благодаря чему сможет предоставить актуальные данные об проекте.

3.5 Создание прототипа базы данных

База данных — это хранилище с обработкой данных. Существует множества инструментов для реализации баз данных и для правильного выбора вначале нужно анализировать их и после произвести выбор с последующим проектированием его.

Выбор СУБД необходимо делать с тщательной осторожностью, так как бывают разные СУБД, которые еще могут быть недоработаны или же вообще не подойти для разрабатываемой информационной системы. Буду рассматривать только популярные проверенные СУБД. После их перечисления проведем краткий их анализ. Для анализа были выбраны следующие СУБД: MySQL, Microsoft SQL server, Oracle, PostgreSQL, MariaDB, Microsoft Access.

MySQL – Из всех популярных видов баз данных, MySQL является самой популярной из всех что есть на данный момент, по ней имеются очень много инструкция и документации, что делает ее очень легкой в усвоении. При этом она имеет большой функционал и высокую гибкость в работе с данными.

Microsoft SQL server – Данная СУБД разработана от компании Microsoft. В основном используется для запросов Transact– SQL. Данные запросы являются расширенным языком обычного SQL от компании Microsoft.

Oracle – Это СУБД разработанная компанией Oracle, она поддерживает множество технологий для реализации объектно–ориентированных подходов к организации баз данных. И имеет высокую популярность в этой сфере.

PostgreSQL – Одна из самых конкурирующих СУБД является как раз PostgreSQL, по причине того, ориентирована на полное соответствие стандартам SQL. И часто используется в коммерческих целях.

MariaDB – Является ответвлением от системы управления данными MySQL. Данная СУБД является гибридом таких СУБД как Oracle и MySQL.

В ходе обдумывания какую СУБД выбрать были сформулированы некоторые основные критерии, по которым был сделан выбор СУБД:

- кроссплатформенность;
- бесплатность;
- быстродействие;
- удобное использование и легкое обучение;
- наличие документации по работе с СУБД
- распространённость.

Результат сравнительного анализа СУБД приведен в таблице 6

Таблица 6 – Сравнительный анализ СУБД

Критерии Оценки	MySQL	Microsoft SQL server	Oracle	PostgreSQL	MariaDB
кроссплатформенность	+	–	–	+	–
бесплатность	+	–	–	–	+
быстродействие	+	+	+	+	+
удобное использование и легкое обучение	+	+	+	–	+
наличие документации по работе с СУБД	+	+	–	–	+
распространённость	+	+	+	+	–
Общая сумма баллов:	6	4	3	3	4

Исходя из таблицы 6, делаем вывод в том, что СУБД MySQL подходит для резолюции данной информационной системы больше всего при доступных экономических финансах.

После исследования и выбора СУБД нужно произвести построение логической и физической модели данных, для того чтобы точно представлять, что входит в тот или иной модуль, для этого было произведено проектирование логической и физической модели данных, они будут представлены на рисунках 32, 33.

Логическая модель данных представляет собой небольшую структуру базы данных, но без прямой ее привязки. Данная модель покажет взаимосвязь между сущностями и их атрибуты.

Разработанная логическая модель данных представлена на рисунке 32.

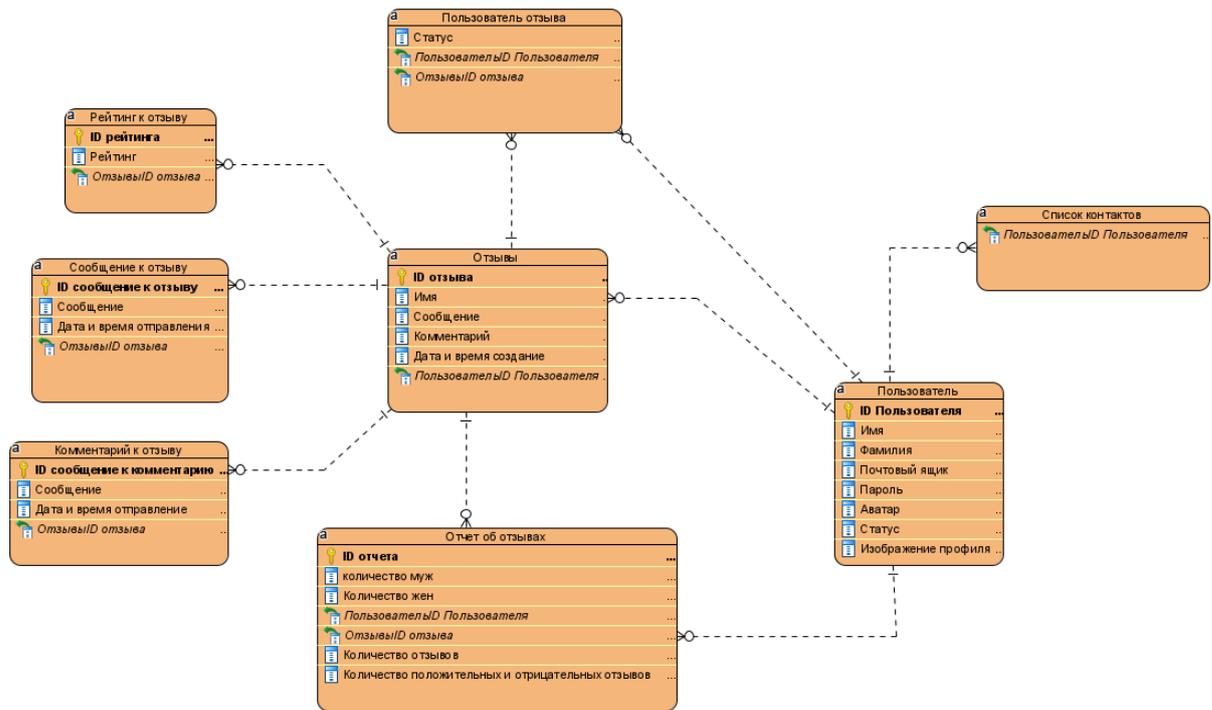


Рисунок 32 – Логическая модель данных

Логическая модель данных, изображенная на рисунке 32, представляет 8 сущностей:

- сущность «Пользователь отзывов», описывает какие пользователи и с каким статусом относятся к тем или иным отзывам;
- сущность «Отзывы», хранит в себе такую информацию как имя отправителя, набранное сообщение, комментарий отправителя, дату и время создания отзыва и комментарий;
- сущность «Отчет об отзывах», хранит и выводит такую информацию как количество людей и их пол, общее количество отзывов, количество положительных и отрицательных отзывов. И описывает какие отзывы были добавлены в отчет, и какие пользователи оставлял их;
- сущность «Пользователь», содержит данные о пользователе, такие как имя, фамилия, почтовый ящик, пол, пароль;
- сущность «Список контактов», описывает какие пользователи были добавлены в контакты у другого пользователя;
- сущность «Сообщение к отзыву», описывает какие сообщения и их время создание были присвоены тому или иному отзыву;

- сущность «Комментарий к отзыву» описывает какие сообщения комментария и их время создание были присвоены тому или иному отзыву;
- сущность «Рейтинг к отзыву», описывает рейтинг того или иного отзыва в системе.

Физическая модель данных представлять такую же схему, как и логическая, за исключением того, что внутри сущностей описываются их параметры. Физическая модель находится на рисунке 33.

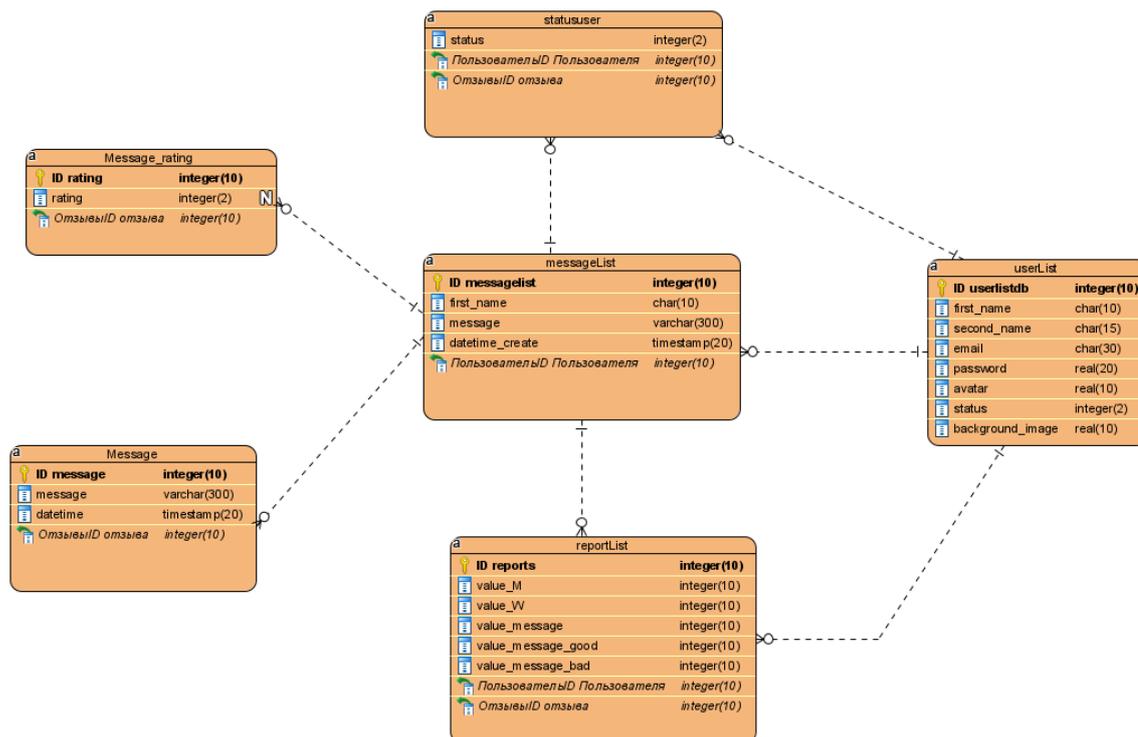


Рисунок 33 – Физическая модель данных

Физическая модель состоит из нескольких сущностей, в них входит: «userList», «statususer», «messageList», «reportList», «Message», «Message_rating». Все сущности при этом обладают отношением «один ко многим».

Теперь для реализации самой базы нужно разобраться с классовой системой. Нужно построить диаграмму классов, чтобы видеть базу данных в целом, для последующего построения базы. Данная диаграмма будет показана на рисунке 34.

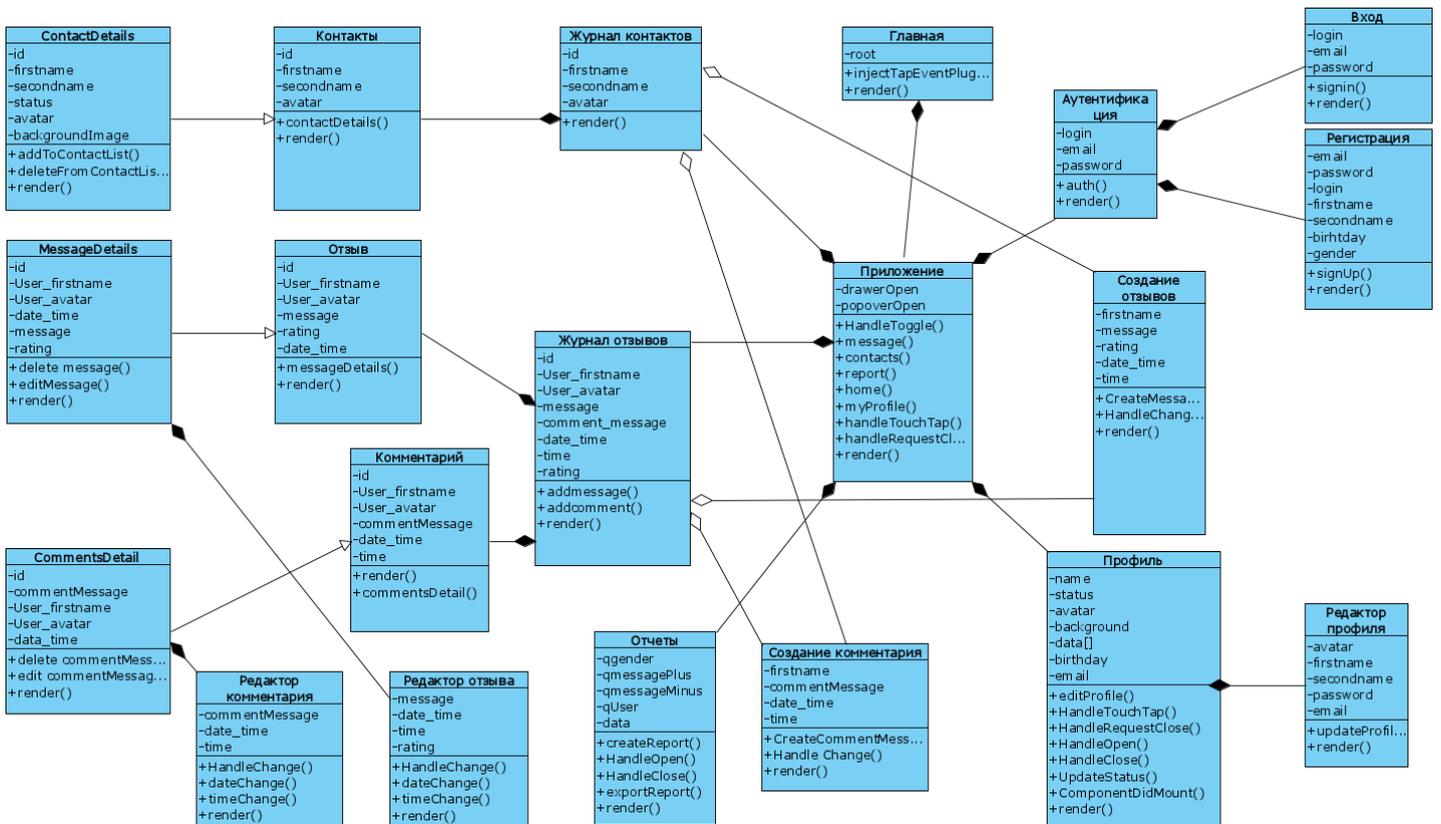


Рисунок 34 – диаграмма классов

После выполнения анализа и проектирование диаграмм, можно переходить к созданию самой базы данных. Она будет реализована на хостинге при помощи развернутого сервера apache. Apache – это кроссплатформенное программное обеспечение, которое позволит разместить веб-сервер.

Для создания СУБД MySQL, требуется веб-сервер, чтобы все данные СУБД где-то хранились и взаимодействовали с сайтом, над которым будет производиться взаимодействие. Веб-сервер можно создать с помощью 2 вариантов:

- поднятие веб-сервера на локальном компьютере;
- воспользоваться системой хостинга.

Каждый вариант имеет свои плюсы и минусы, но я коротко скажу в чем отличие, чтобы не развивать много воды, так как мы разрабатываем базу, а не устанавливаем сервер. Отличия просты, в первом случае все происходит абсолютно бесплатно, но имеет ограничения по выходу в общий доступ, а во

втором случае придётся немного заплатить, но сайт с базой данных будет открыта для просмотра любому.

Я воспользовался таким сервисом как REG.RU. Он позволяет не только арендовать хостинг, но и выбрать название домена. Если что поясню про название хостинг и домен.

Хостинг – Это сервис, который позволяет разместить свой материал физическим или юридическим лицам.

Домен – это название сайта для лучшего поиска той или иной страницы в сети.

Так вот сервис REG.RU позволил мне поднять свой сервер, на котором будет находится мой материал по сайту и базой данных. Поднятая база данных под управлением phpMyAdmin будет показана на рисунке 35.

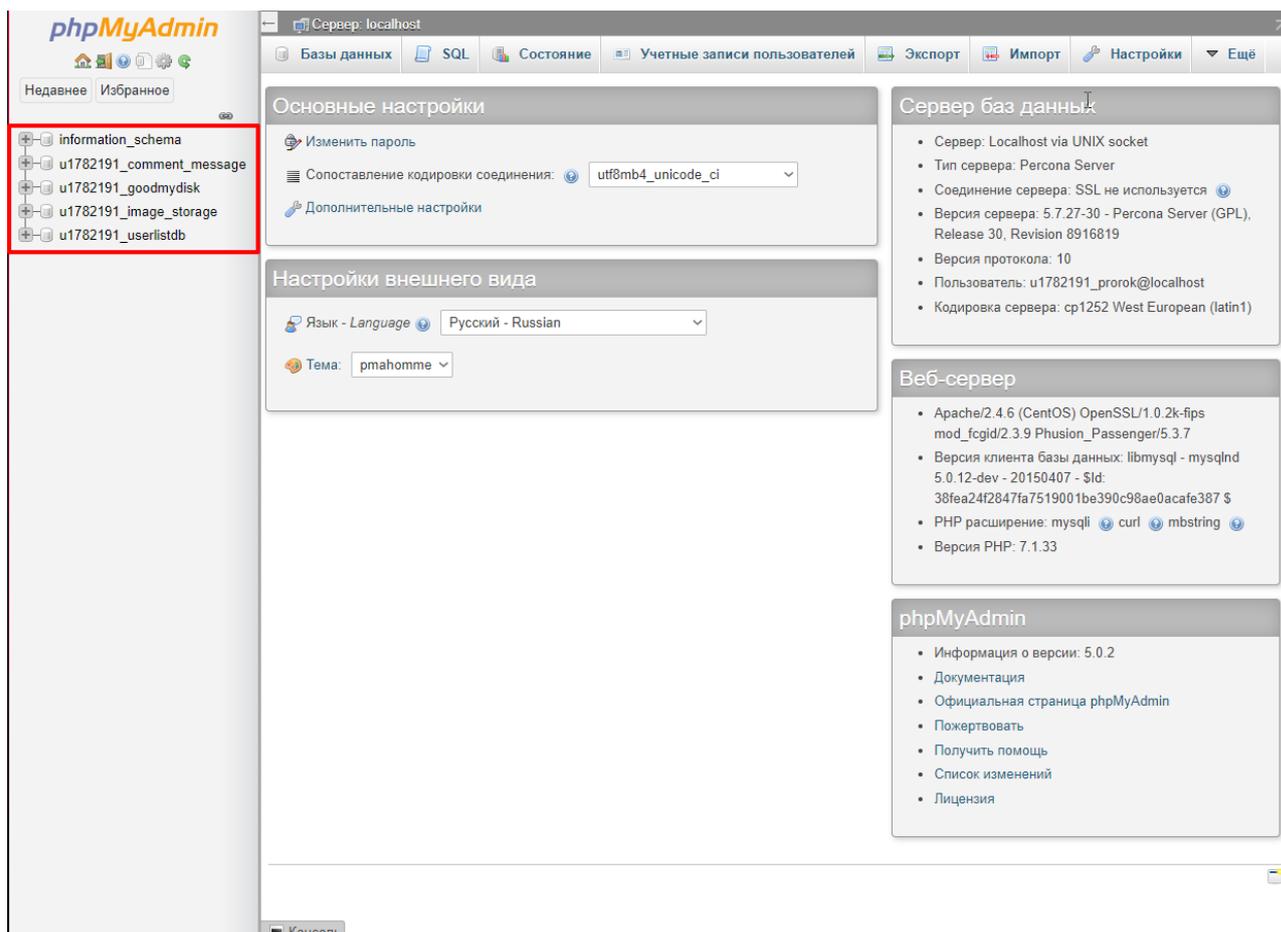


Рисунок 35 – База данных под управлением phpMyAdmin

Каждая база данных состоит из таблиц, в которых с определенной степенью защиты шифруется и защищаются данные от посторонних лиц. Для

лучшего введения базы данных я разделил информацию по нескольким базам данных, чтобы было легче разобраться с поступающими данными. В каждой базе данных имеются свои таблицы и свои настройки.

[Создание базы данных](#) происходит очень легко, так как база данных происходит на хостинге, то есть отличия создание данных баз. Если бы [phpMySql](#) был поставлен на локальном компьютере, то создание базы происходит через сам интерфейс, достаточно просто выбрать название базы и все. Но на хостинге происходит немного иначе, так как общий доступ к серверу ограничен, то нужно подавать заявку на создание базы через сайт REG.RU. Ниже будут представлены изображения вкладок для последующего создания базы данных.

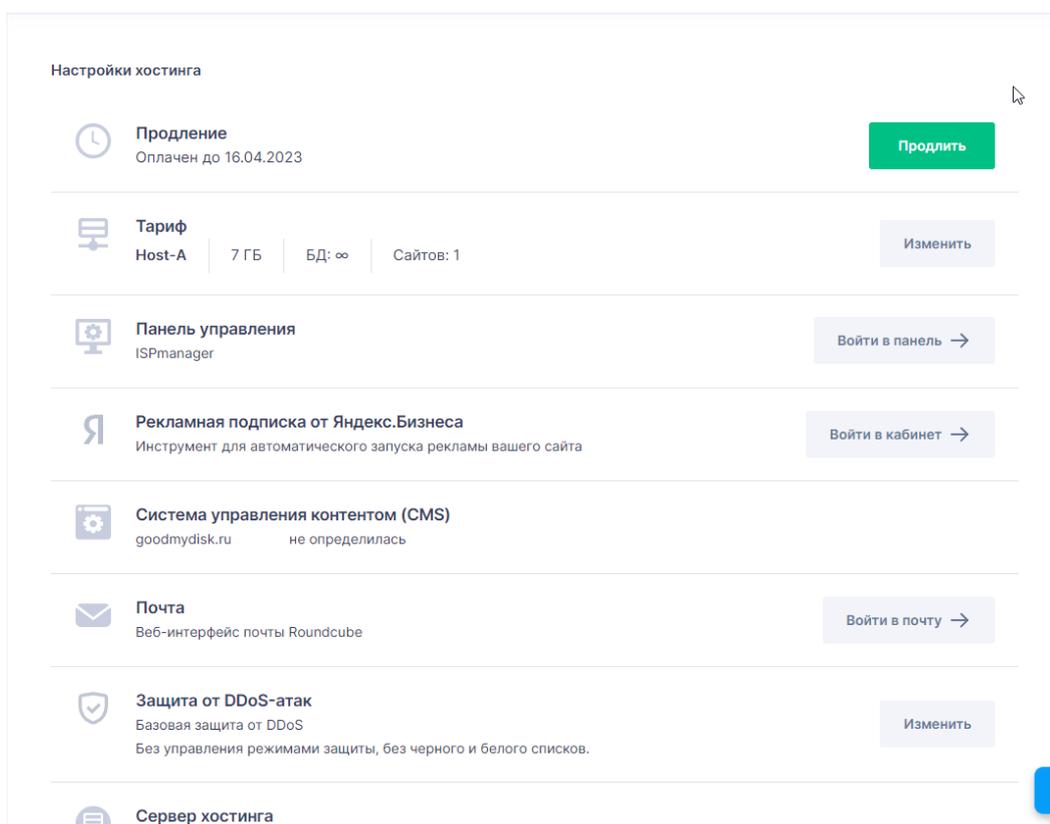


Рисунок 36 – Меню настройки хостинга

[Переходи во вкладку панель управления](#), так как большинство изменений и управление происходит там.

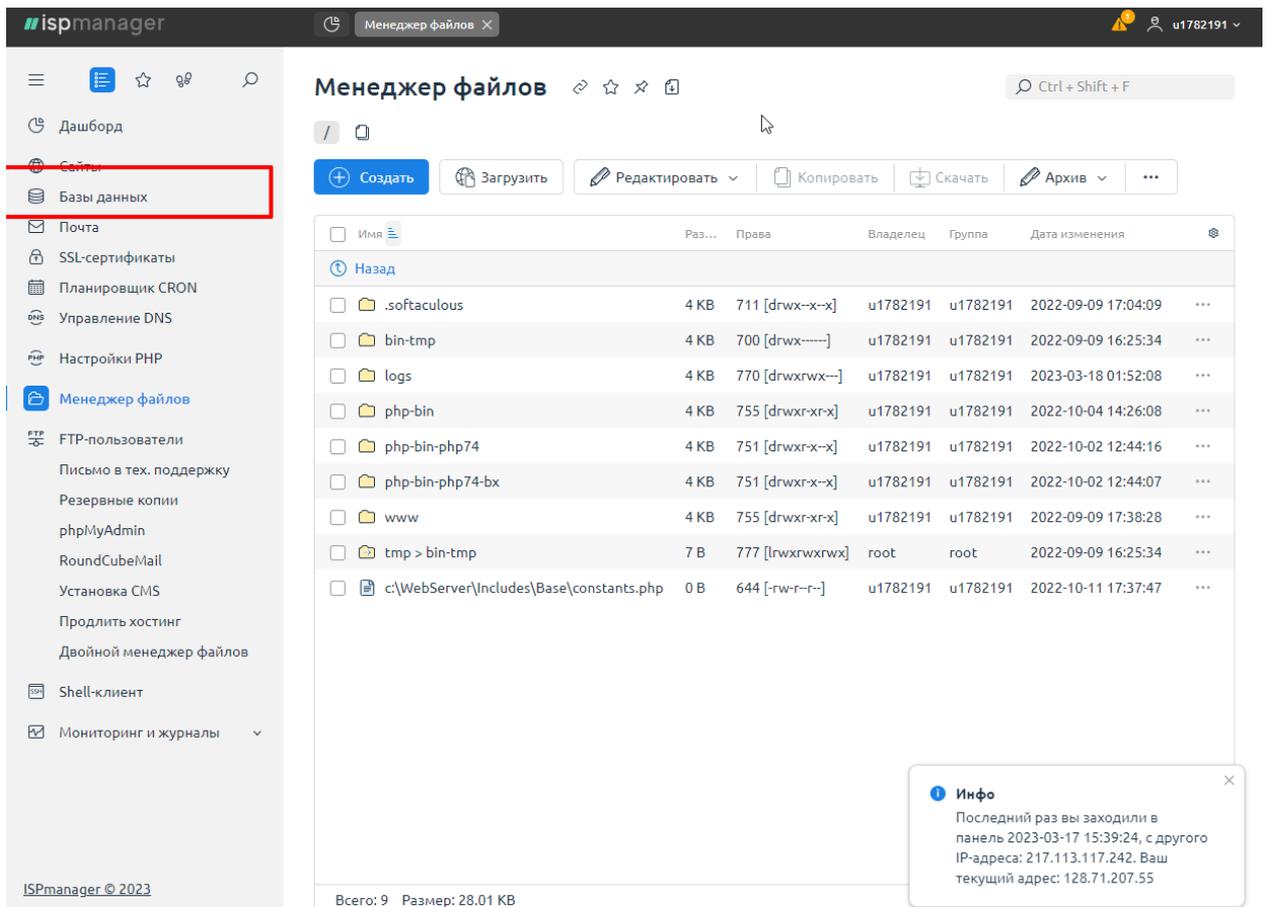


Рисунок 37 – Меню управления хостингом ispmanager

В этом меню нужно перейти по вкладке базы данных и там будут показаны все базы данных, которые были созданы во время проектирования.

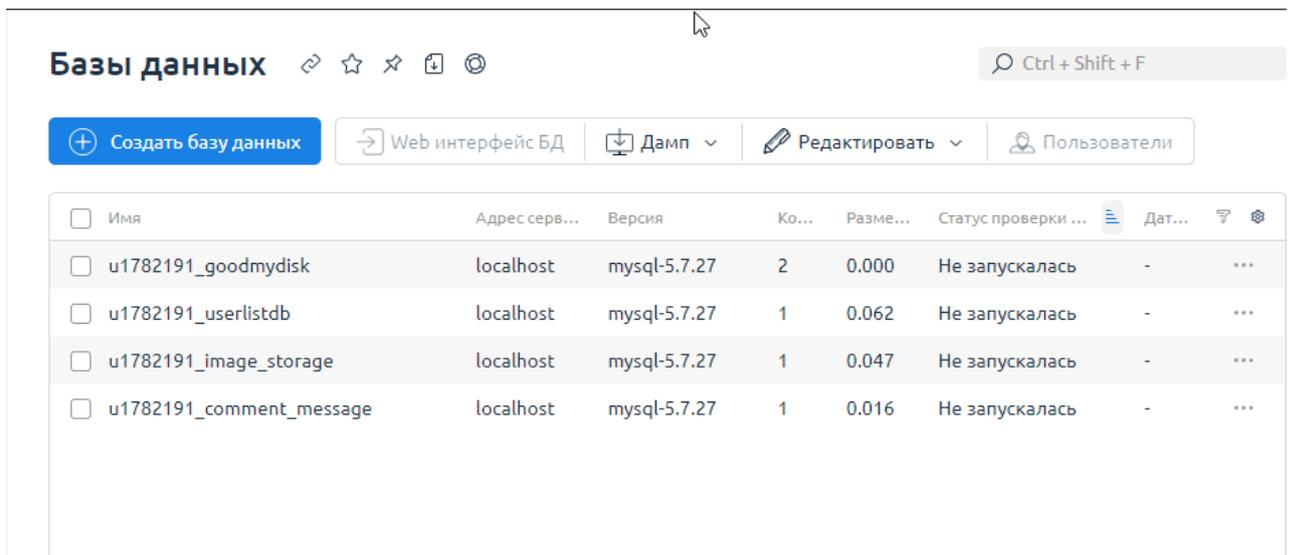
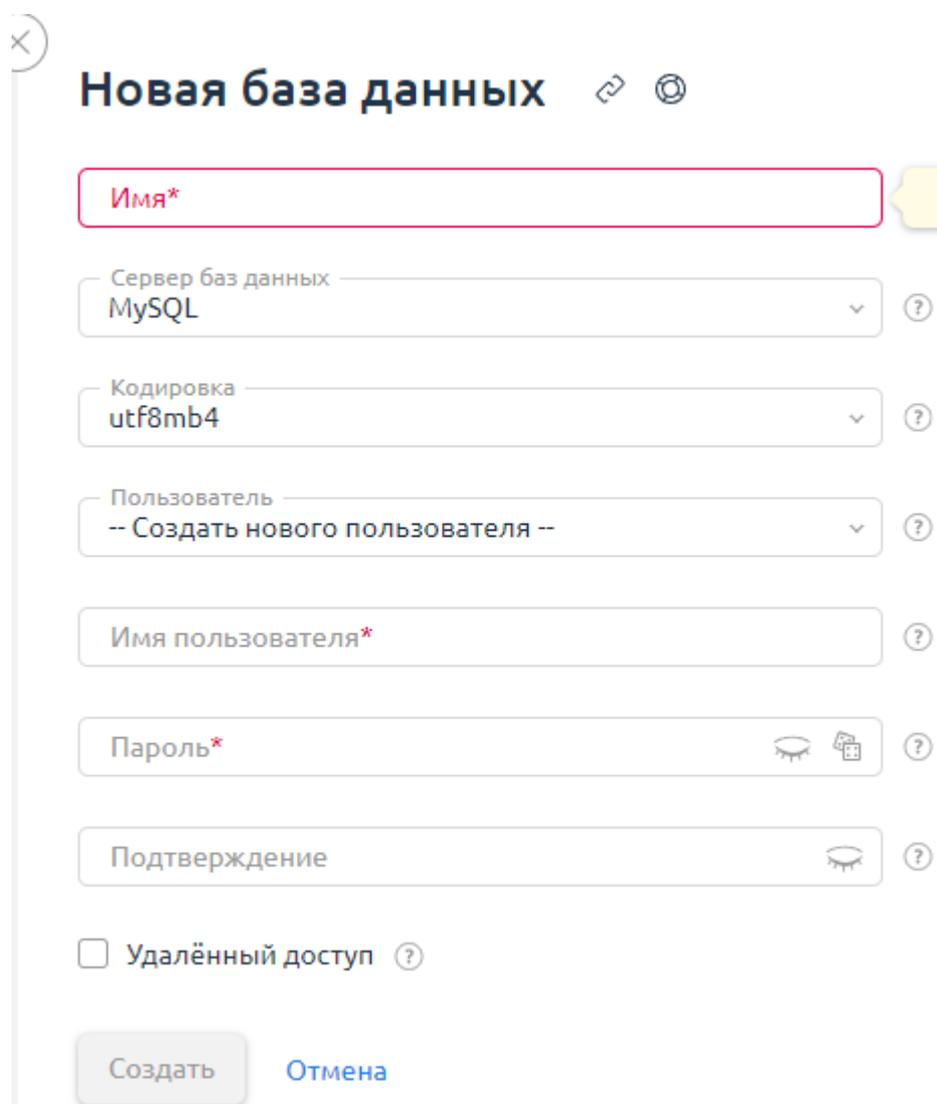


Рисунок 38 – Вкладка информации о базах данных в меню управления ispmanager

После нажатия на кнопку создать базу данных появляются поля для ввода необходимых данных. После ввода отправляется запрос на создание базы данных с нужно уже информации.



Новая база данных

Имя*

Сервер баз данных
MySQL

Кодировка
utf8mb4

Пользователь
-- Создать нового пользователя --

Имя пользователя*

Пароль*

Подтверждение

Удалённый доступ

Создать Отмена

Рисунок 39 – Вкладка создание базы данных

Период создание базы данных на сервисе REG.RU занимает около дня. После успешного выполнения запроса, он появляется в списке баз данных, который показан на рисунке 39.

Теперь перейдем к разработки таблиц в этих самых баз данных. В интерфейсе phpMyAdmin создание таблиц происходит достаточно просто, по уже анализированной диаграмме классов, создаем таблицы. Переходим в phpMyAdmin и после переходим в любую базу данных и нажимаем кнопку «Новая».

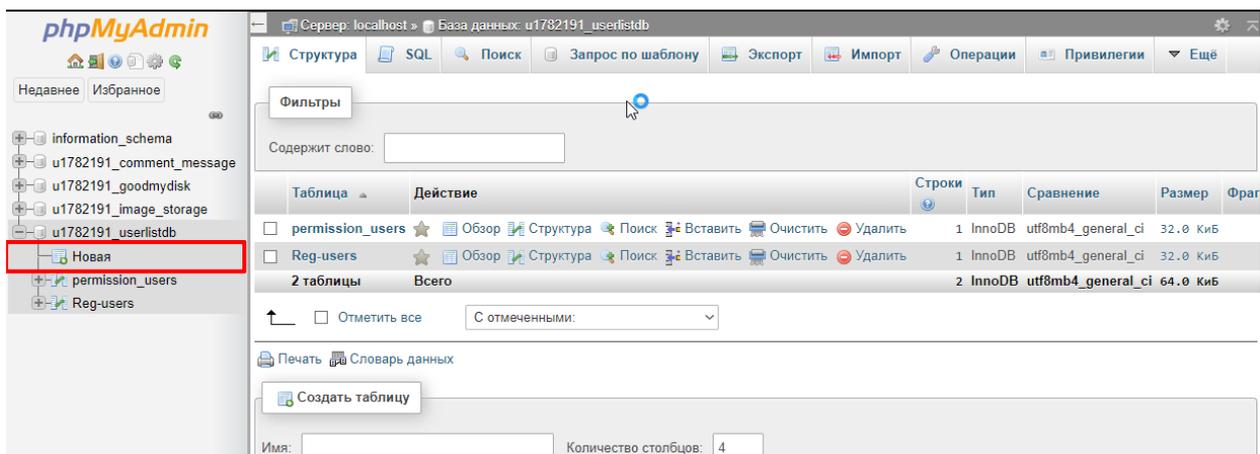


Рисунок 40 – Открытое меню базы данных в phpMyAdmin

После перехода на вкладку создания, нужно указать такие данные как:

- имя;
- тип;
- значение (Если имеется);
- атрибут;
- индекс;
- A_I – это пункт отвечает за автоматическую нумерацию столбцов.

Как только будут указаны все имеющиеся данные, нужно нажать на кнопку сохранить. И таблица создастся. Ниже будет показано изображение пункта создания таблиц.

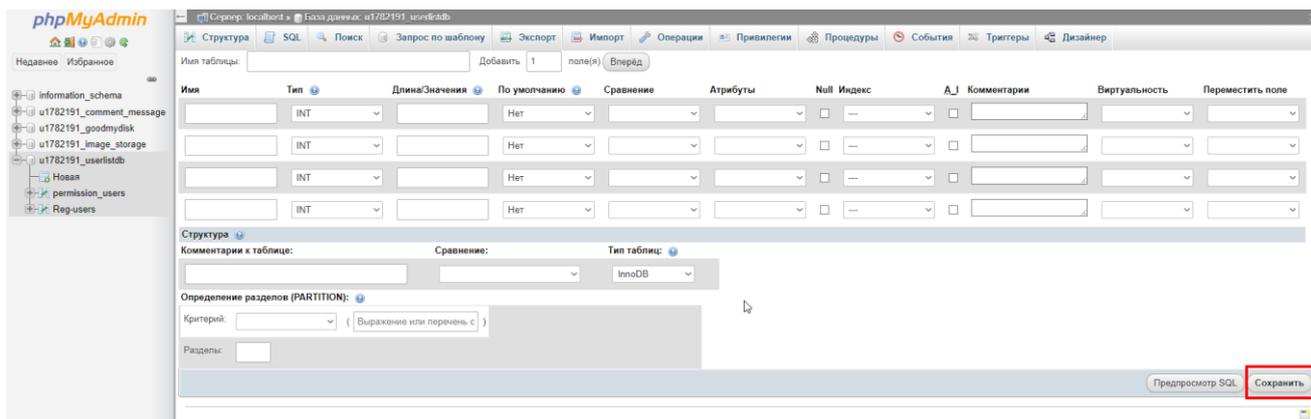


Рисунок 41 – Страница создание таблиц в СУБД

Подведем итоги по созданию прототипа базы данных. Было создано в СУБД 4 базы данных, каждая из которых выполняет свою роль, и каждая хранит в себе множество таблиц с теми или иными функциями. Далее я плавно

перехожу к работе самой базы данных и взаимодействия этих данных друг с другом.

3.6 Разработка контрольных примеров работы базы данных

Данный раздел будет посвящен разработки функционала данного проекта, будет предоставлены критерии и требования ко всем видам функционала и предоставлен программный код к ним. После будет сделано небольшое описание для будущего развития проекта.

Начну я с пояснения функционала каждого модуля и описание критериев по их разработки. Всего на данный момент имеется 6 модулей. Каждый модуль использует базу данных, и они могут взаимодействовать друг с другом между собой с помощью связи таблиц.

Всего будет описано 6 модулей при этом я их разделю на 3 группы:

- профиль;
- вход и регистрация;
- аналитика.

Пояснение каждого модуля были описаны в разделе 3.2.4. Далее перейдем к описанию критериев для базы данных по порядку.

Первым будет модуль вход. Критерии для этого модуля просты, должна быть простая авторизация с минимальным количеством входов. Модуль связывается с базой данных, проверяет введенные данные с зарегистрированными и после ставит условия, если все данные верны, то создает сессию, которая так же подтягивает все данные из профиля. Если данные не верны выдает ошибку на экран. Какие будут критерии по интерфейсу, данного модуля:

- должно быть 2 вида входа в систему, первая компактная, вторая в отдельной страничке;
- все кнопки и поля должны быть небольшого размера;
- дизайн должен быть аккуратным и не разрежающим.

По функционалу данного модуля:

- все кнопки должны функционировать, должным образом;
- все поля не должны расширяться и должны быть статичные, так же поля не должны быть заблокированы.

Следующий модуль идет регистрация. Для регистрации требования более тонкие. Во-первых, всего 4 поля:

- логин;
- email;
- пароль;
- имя пользователя.

При этом первые три являются обязательными. Следующий критерий — это вводимые данные. Имя может быть любое, а вот логин не должен быть меньше 3 символов, при этом он может совмещать в себя и буквы и цифры, специальные символы не допустимы. А вот пароль может и должен иметь все виды символов. Обязательные символы должны быть такие как:

- хотя бы один спец символ;
- хотя бы одна цифра;
- хотя бы одна заглавная буква.

Пароль должен иметь не менее 8 символов. Должен так же присутствовать фильтр слов, чтобы не было слов со ненормативной лексикой.

Благодаря такой безопасности, украсть информацию будет достаточно сложно. Следующим пунктом в данном модуле будет дизайн. Он должен быть достаточно прост. Регистрация будет происходить на отдельной страничке, и вся форма должна быть строго по центру, для простоты ввода данных.

Все введенные данные должны быть записаны в определенную таблицу, для последующего их использования.

[Профиль важный модель](#) для его разработки требуется соблюсти некоторые требования. Профиль должен работать только при рабочей сессии, то есть, если не было произведена авторизация, то какие-либо данные не должны загружаться на страницу. А сам профиль должен иметь несколько

пунктов взаимодействия между модулями. Профиль взаимодействует с базой за счет модуля входа, и модулем настройки. Каждый из модулей позволяет управлять той или иной функцией профиля. Например, вход позволяет создать сессию и прогрузить данные того пользователя, который авторизовался. А настройки позволяют изменить имеющиеся данные пользователя на другие значения. Профиль должен иметь всю необходимую информацию о пользователе, его имя, логин, дата рождения, пол, аватар. Каждый пользователь должен изменять основные данные своего профиля, кроме логина, он является статичным и не изменяемым, так как поиск информации происходит по данному имени. Пол должен вылезать в виде списка, для лучшего и правильного ввода информации. Все поля должны иметь гармоничную структуру, не должно оставаться много пространства. А также должны быть кнопки управления профилем, таких как сохранение новых данных и системы отчетов, если есть определенные права доступа. Вид данного профиля может быть свободным, но должен совпадать по тематике основной страницы.

Последней группой будет аналитика в нее входят такие модули как отзыв и отчет. Модуль отзыва должен обеспечивать полноценную читаемость и иметь все нужные кнопки и поля для ввода данных. При этом если существует сессия, то некоторые данные такие как имя должны подставляться в поля для последующего удобства ввода данных. Так же будут ограничения по правам, если у пользователя есть права администратора, то у него открывается новая кнопка на переход к страничке отчета, благодаря которой он может сформировать отчет за ближайшее время. При этом в отчете показывается такая информация как:

- количество человек в день;
- количество мужчин и женщин;
- кол-во положительных и отрицательных отзывов;
- средний рейтинг в день.

На этом описание требований подошел к концу. Далее будет показан программный код модулей.

```
<!DOCTYPE html>
<html lang="ru">
<head>
  You, позавчера * Страница с отзывом (Без вкладшей.)
  <meta charset="UTF-8">
  <title>menuProfile</title>
  <link rel="stylesheet" href="/Assets/CSS/newMenu.css">
  <link rel="stylesheet" href="/Assets/CSS/Style_message.css">
  <link rel="stylesheet" href="/Assets/CSS/figure.css">
</head>
<body>
<div class = "newMenu">
  <div class="block_menu">...
  </div>
  <div class = "body">
    <div class = "text_message">
      <h2>Оставить отзыв о проекте</h2>
      <p>Здесь Вы можете оставить отзыв о сайтах в интернете. Сделать это можно совершенно анонимно и без
    </div>
    <div class = "create-message">
      <div class = "block-create-message">
        <div class = "name-and-rating"><p>Имя<input id = "anonims_name"><p></div>
        <div class="rating-area">
          <input type="radio" id="star-5" name="rating" value="5">
          <label for="star-5" title="Оценка «5»"></label>
          <input type="radio" id="star-4" name="rating" value="4">
          <label for="star-4" title="Оценка «4»"></label>
          <input type="radio" id="star-3" name="rating" value="3">
          <label for="star-3" title="Оценка «3»"></label>
          <input type="radio" id="star-2" name="rating" value="2">
          <label for="star-2" title="Оценка «2»"></label>
          <input type="radio" id="star-1" name="rating" value="1">
          <label for="star-1" title="Оценка «1»"></label>
        </div>
        <div class = "box_message">
          <p>Отзыв:</p>
          <textarea name = "message" cols = "" rows = "" placeholder = "Введите ваш отзыв"></textarea>
        </div>
        <div class = "message_button"><p><input type = "submit" value = "Отправить"></p></div>
      </div>
    </div>
    <div class = "prewive-message"></div>
  </div>
</div>
</body>
```

Рисунок 42 – Программный код страницы с отзывами

```

<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>menuProfile</title>
  <link rel="stylesheet" href="/Assets/CSS/newMenu.css">
  <link rel="stylesheet" href="/Assets/CSS/report.css">
</head>
<body>
  <div class = "newMenu">
    <div class="block_menu">...
    </div>
    <div class = "body_table">
      <div class = "menu_footer">
        <p><input type = "button" value = "Сформировать"></p>
      </div>
      <div class = "table">
        <table>
          <tr>
            <th>Дата</th>
            <th>Кол-во М.</th>
            <th>Кол-во Ж.</th>
            <th>Кол-во положительных</th>
            <th>Кол-во отрицательных</th>
            <th>Всего ОТЗЫВОВ</th>
          </tr>
          <tr>...
          </tr>
          <tr>...
          </tr>
          <tr>...
          </tr>
          <tr>...
          </tr>
        </table>
      </div>
    </div>
  </div>
</body>
</html>

```

Рисунок 43 – Программный код страницы с таблицей отчёта

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>menuProfile</title>
  <link rel="stylesheet" href="/Assets/CSS/newMenu.css">
  <link rel="stylesheet" href="/Assets/CSS/Profile.css">
</head>
<body>
<div class = "newMenu">
  <div class="block_menu">...
  </div>
  <div class = "body_profile">
    <div class = body_footer>
      
    </div>
    <div class = "body_center_settings">
      <div class = "body_settings">
        <p><span>Имя<input type = "text" value = "Кирилл"></span><span class = "second_name">Фамилия<input type = "text"
        <p>Email<input class = "email" type = "text" Value = "Modolei20@yandex.ru"></p><br>
        <p>Пол<input type = "text" Value = "Мужской"></p>
      </div>
    </div>
    <div class = "button">
      <input type = "button" value ="Settings"><input type = "submit" value ="Save">
    </div>
  </div>
</div>
</body>
```

Рисунок 44 – Программный код страницы профиля пользователя
Далее приложу программный код взаимодействия СУБД с интерфейсом.

```
<?php
require("../includes/base/Base_userlistdb");
$link = mysqli_connect(DB_SERVER,DB_USER, DB_PASS, DB_NAME);
mysqli_set_charset($link, 'utf8');

if (!$link) {
  die('Ошибка соединения: ' . mysql_error());
}
echo 'Успешно соединились';
?>
```

Рисунок 45 – Программный код подключения к СУБД

```
<?php
unset($_SESSION['logged_username']);
session_destroy();
header("location: ../index.php");
?>
```

Рисунок 46 – Программный код выхода из сессии.

```

<?php
$first_name = filter_var(trim($_POST['first_name']));
$username = filter_var(trim($_POST['username']));
$email = filter_var(trim($_POST['email']));
$password = filter_var(trim($_POST['password']));
$name_image = 'kartinki_volka_na_avu_04.jpg';
?>
<?php
if (mb_strlen($username) < 5 || mb_strlen($username) > 90) {
    echo "Недопустимая длина логина";
    exit();
}
else if (mb_strlen($first_name) < 5) {
    echo "Недопустимая длина имени.";
    exit();
}
//Проверка длины значений
?>
<?php
$password = md5($password."thisisforhabr");
$mysql = new mysqli('localhost', 'u1782191_prorok', 'zV8dF3tD7m', 'u1782191_userlistdb');
mysqli_set_charset($mysql, 'utf8');

$result = $mysql->query("SELECT * FROM `Reg-users` WHERE `username` = '$username'");

$user = $result->fetch_assoc();

if (!empty($user)) {
    echo "Данный логин уже зарегистрирован!";
    exit();
}
$mysql->query("INSERT INTO `Reg-users` (`first_name`,`username`,`email`,`password`,`name_image`) VALUES ('".$first_name."','".$username."','".$email."','".$password."','".$name_i

$mysql->close();
?>
<?php
$new_url = "https://goodmydisk.ru/authorization/login.php";
header('Location: '.$new_url, true, 302);
exit();
?>

```

Рисунок 47 – Программный код создания пользователя

```

<?php
$getsql = mysqli_query($mysql, "SELECT * FROM `Reg-users` WHERE `username` = '$username'");
$result = mysqli_fetch_array($getsql);
$_SESSION['logged_username'] = $result ['username'];
$_SESSION['logged_first_name'] = $result ['first_name'];
$_SESSION['logged_email'] = $result ['email'];
$_SESSION['logged_name_image'] = $result ['name_image'];
$_SESSION['id_session'] = session_id();
?>

```

Рисунок 48 – Программный код сессии

```

<?php
$username = filter_var(trim($_POST['username']));
$password = filter_var(trim($_POST['password']));
?>
<?php
$password = md5($password."thisisforhabr");

$mysql = new mysqli('localhost', 'u1782191_prorok', 'zV8dF3tD7m', 'u1782191_userlistdb');
mysqli_set_charset($mysql, 'utf8');

$sql = $mysql->query("SELECT * FROM `Reg-users` WHERE `username` = '$username' AND `password` = '$password'");

if (mysqli_num_rows($sql) > 0) {
    include("/var/www/u1782191/data/www/goodmydisk.ru/Includes/session.php");
    $mysql->close();
    header('location: /index.php');
}
else {
    echo "Ошибка авторизации";
}
?>

```

Рисунок 49 – Программный код авторизации

```

<?php require '../message/connect.php'; ?>
<!DOCTYPE html>
<html lang="ru">

<head>
  <meta charset="UTF-8">
  <title>Облакаго для личного пользования</title>
  <meta name="description" content="Тестовое облакаго">
  <link rel="stylesheet" href="\Assets\css\Style.css">
  <link rel="stylesheet" href="/Assets/CSS/Style_test_message.css">
</head>

<body>
<?php include("../includes/main_header.php"); ?>
<div class="message_contant">
  <div class="box_form_message">
    <div class="form_message">
      <form action="sendMessage.php" method="post" name="form">
        <p class="is-h">Автор:</><br><input name="author" type="text" class="is-input" id="author"></p>
        <p class="is-h">Текст сообщения:<br><textarea name="message" rows="5" cols="50" id="message"></textarea></p>
        <input name="js" type="hidden" value="no" id="js">
        <button type="submit" id='click' name="button" class="is-button">Отправить</button>
      </form>
    </div>
    <div class="clear">
    </div>
    <div id="commentBlock">
    <p>Комментарии к статье</p>
    <?php
    $result = $mysql->query("SELECT * FROM `messages`"); /*Получаем все данные из таблицы*/
    $comment = $result->fetch_assoc(); /* В результирующий массив */
    do{echo "<div class='comment' style='border: 1px solid gray; margin-top: 1%; border-radius: 5px; padding: 0.5%;'>Автор: <strong>".$comment['author'].</strong><br>"
    }while($comment = $result->fetch_assoc());
    ?>
    </div>
  </div>
</div>
</div>
</div>
<?php include("../includes/main_footer.php"); ?>
</body>
</html>

```

Рисунок 50 – Программный код создание отзыва

После окончания разработки самой оболочки, базы данных и других элементов моего программного продукта, перейдем к написанию инструкция для каждого вида пользователя.

3.7 Разработка инструкция для каждого пользователя

Входе этого раздела будет предоставлены инструкция для 3 групп пользователей:

- нейтральные;
- авторизованные;
- администраторы.

Каждая группа будет иметь схожие порядок действий, но будут отличаться лишь некоторые моменты.

Первая группа является нейтральной из-за чего многое на данном сервисе будет недоступно, максимум что они могут сделать, это оставить отзыв на сайте.

Для начало каждый пользователь переходит по ссылке <https://goodmydisk.ru/>. Данная ссылка позволяет попасть на главную страницу.

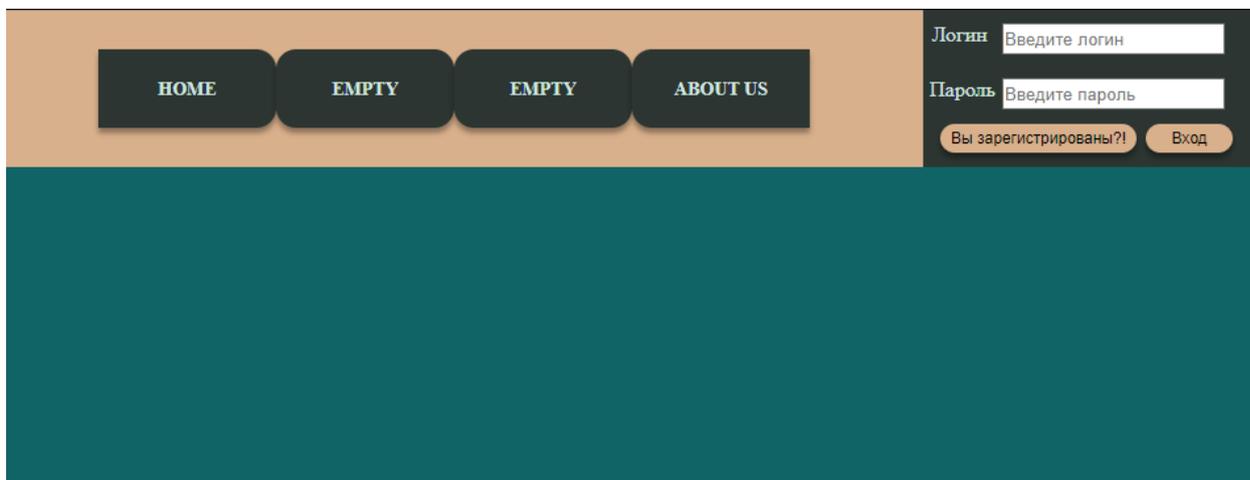


Рисунок 51 – Главная страница

Далее для такого пользователя есть 2 варианта событий. Первый это зарегистрироваться и получить больше прав. И второй это просто оставить отзыв. Рассмотрим вначале первый вариант.

Для регистрации, нужно нажать на кнопку «Вы зарегистрированы?!» на главной странице, чтобы перейти на страницу регистрации, которая будет выглядеть следующим образом.

Регистрация

Полное имя

E-mail

Имя пользователя

Пароль

Уже зарегистрированы? [Зарегистрироваться](#)

[Введите имя пользователя!](#)

© 2014 1stwebdesigner. Все права защищены.

Рисунок 52 – Страница с регистрацией нового пользователя

После успешного ввода данных и нажатии на кнопку «Зарегистрироваться», перейдет на поле входа в систему. На рисунке 53 показан данная страница.

Вход

Имя пользователя

Пароль

Еще не зарегистрированы? [Log In](#)

[Регистрация!](#)

© 2014 1stwebdesigner. Все права защищены.

Рисунок 53 – Страница с авторизацией

А дальше идет этап уже для авторизированной группы, который можно будет посмотреть ниже. А сейчас рассмотрим 2 вариант, если пользователь не хочешь регистрироваться на данном сайте. После просмотра того или иного

контента на странице, он может оставить отзыв, который находится в верхней меню.



Рисунок 54 – Главное меню с контекстным меню

После открытия меню нужно перейти по вкладке «Leave a review». Это позволит перейти на страницу с оставлением отзыва.

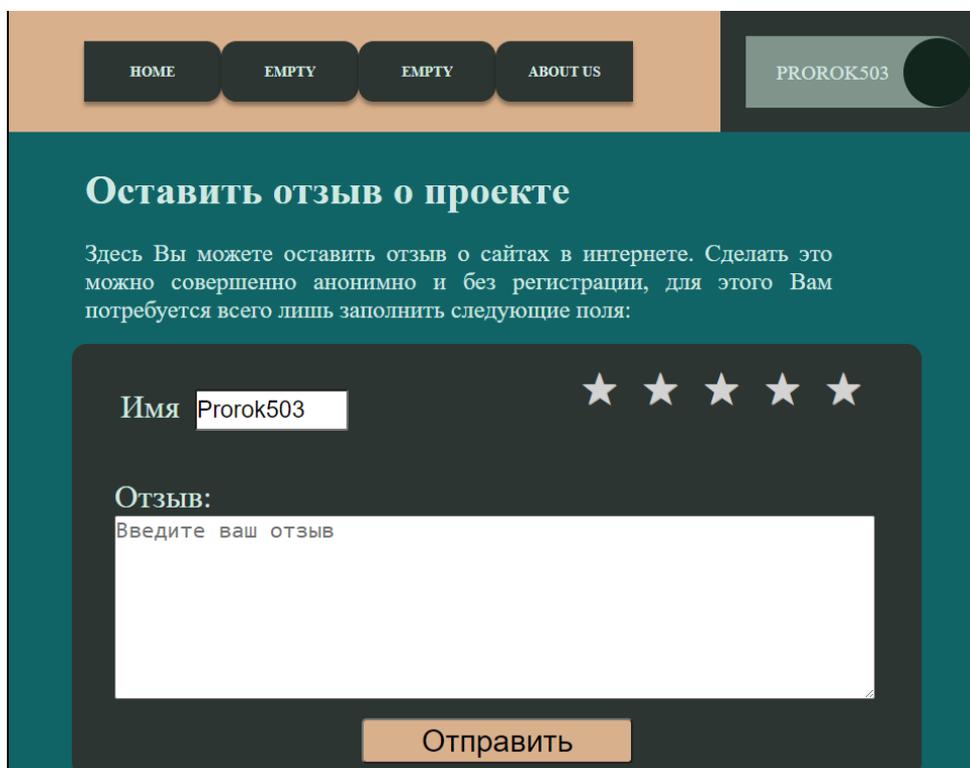


Рисунок 55 – Страница с оставлением отзыва

Здесь можно заполнить форму для отзыва и отправить ее в базу данных, которая в свою очередь запишет информацию в таблицы. Как видно на изображении 55 в поле имя введены данные пользователя, который будет показан в отзыве, кто отправил. У анонимных пользователей, будет пустое поле, они могут нечего не писать и тогда программа автоматически подставит слово «аноним». Или же пользователь может вручную вписать свое имя. На этом инструктаж по анонимным пользователям завершен.

Перейдем ко второй группе авторизованные. У авторизованного пользователя есть свои плюсы, во-первых, у него есть свой профиль, во-вторых, имя на странице отзыва подставляется автоматически. У авторизованных пользователей отличий в переходе между вкладками нет, единственный момент у них добавляется профиль, который мы и обсудим.

После входа в систему, через отдельную страничку или же через главное меню у авторизованного пользователя появляется его аватарка в правом верхнем углу экрана.

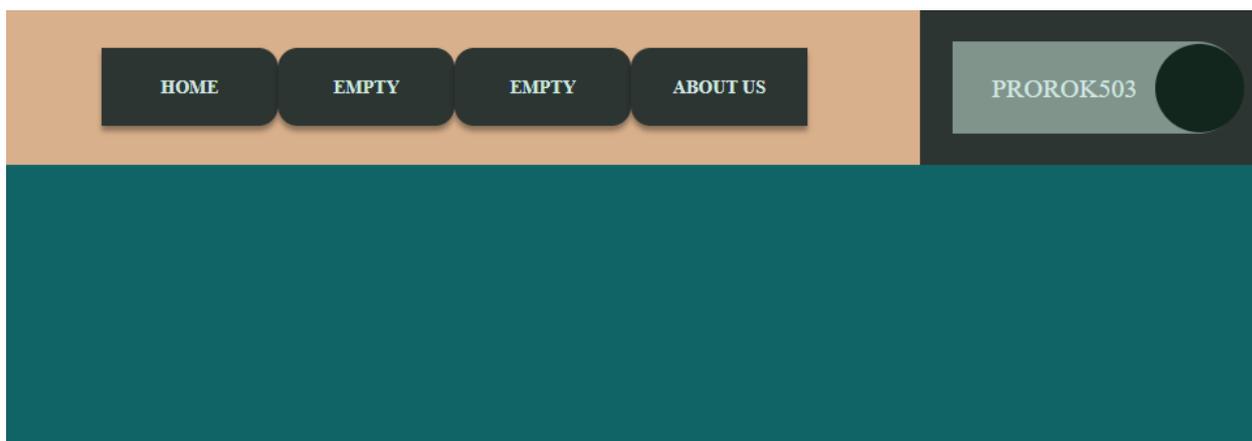


Рисунок 56 – Страница с веденным пользователем

После нажатия на аватарку открывается контекстное меню.



Рисунок 57 – Контекстное меню на главной странице.

И пользователь может произвести несколько действия, перейти в свой профиль, перейти в настройки профиля, перейти в контакты, и выйти из профиля.

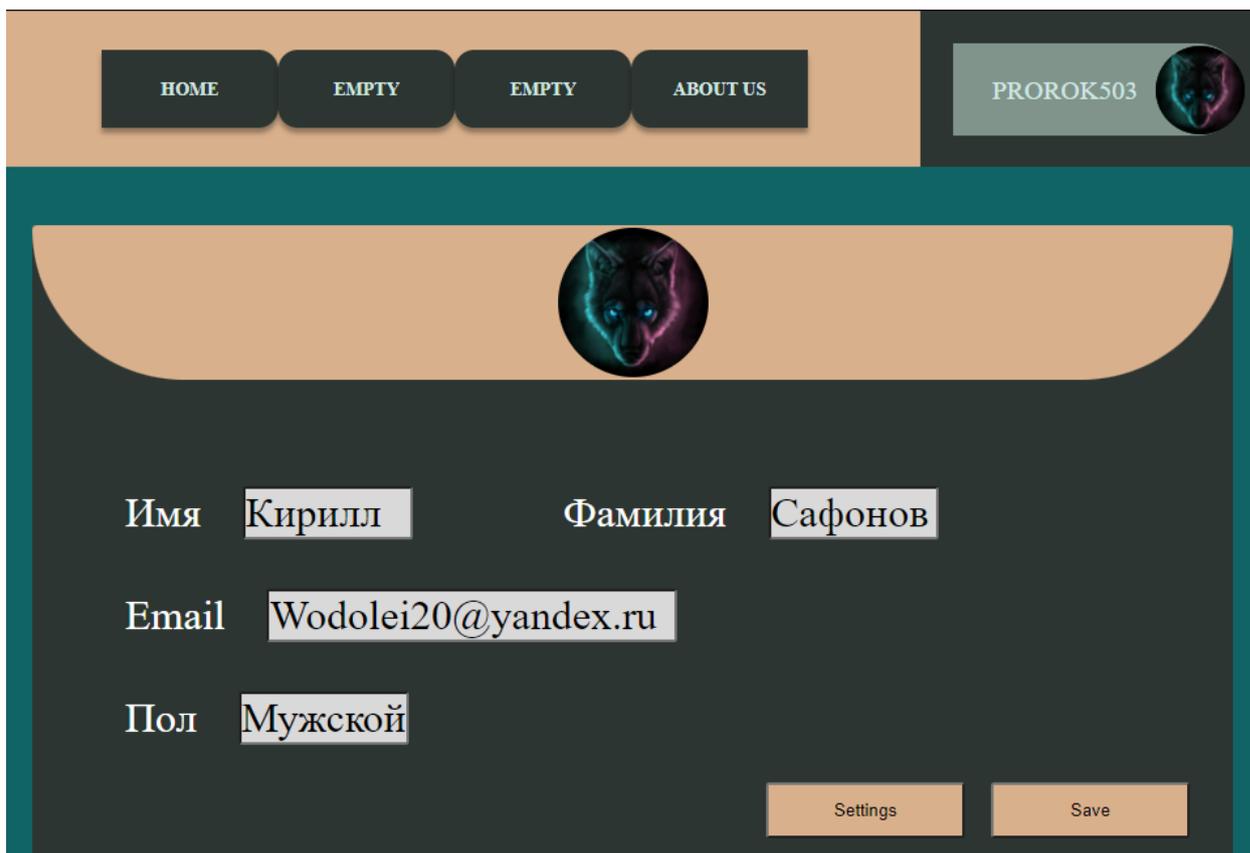


Рисунок 58 – Страница профиля пользователя

Как только специалист или пользователь перейдет в свой профиль он сможет изменять свои личные данные. При изменении своих данных и нажатии на кнопку «Save» произойдет замещение измененных данных на новые введенные. При нажатии на кнопку «Settings» меню переходит на другую страницу, где будут показаны множество функций. У специалиста будет добавлена дополнительная кнопка и при нажатии на кнопку «Report». Будет открываться страницу с отчетом при котором и будут формироваться данные об отзывах. На этом инструктаж по авторизированному пользователю завершен.

У пользователя с правами администрации будет добавлена кнопка «Report», которая отвечает за то, что позволяет отправиться на страницу с отчетом, которая будет показана ниже.



Рисунок 59 – Страница с отчётом об отзыве

И после пользователь формирует отчет с определенными данными. Ну вот и весь функционал страницы.

3.8 Предоставление контрольных примеров

Теперь покажу готовые варианты работы СУБД и пользователя. Начнем с создания пользователя.

Когда пользователь вводит данные то эти данные переходят в базу данных.

Регистрация

Полное имя

E-mail

Имя пользователя

Пароль

Уже зарегистрированы?
[Введите имя пользователя!](#)

Рисунок 60 – страница регистрация с введенными данными

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Ещё

✓ Отображение строк 0 - 1 (2 всего, Запрос занял 0.0006 сек.)

`SELECT * FROM `Reg-users``

Профилирование [[Построчное редактирование](#)] [[Изменить](#)] [[Анализ SQL запроса](#)] [[Создать PHP-код](#)] [[Обновить](#)]

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице | Сортировать по ключу: Ниодного

+ Параметры

		id_user	first_name	username	email	password	name_image		
<input type="checkbox"/>	Изменить	Копировать	Удалить	27	Kirill	Prorok604	wodolei20@yandex.ru	a82aa39ba4400555709640ebe43e4a74	kartinki_volka
<input type="checkbox"/>	Изменить	Копировать	Удалить	28	Кирилл	prorok503	wodolei20@yandex.ru	7d64e1e855fe9bd83cb269b8cfbb89f9	kartinki_volka

Рисунок 61 – База данных с введёнными данными в форме регистрации.

Данные профиля берут с данной таблицы, поэтому дублировать информацию я не буду.

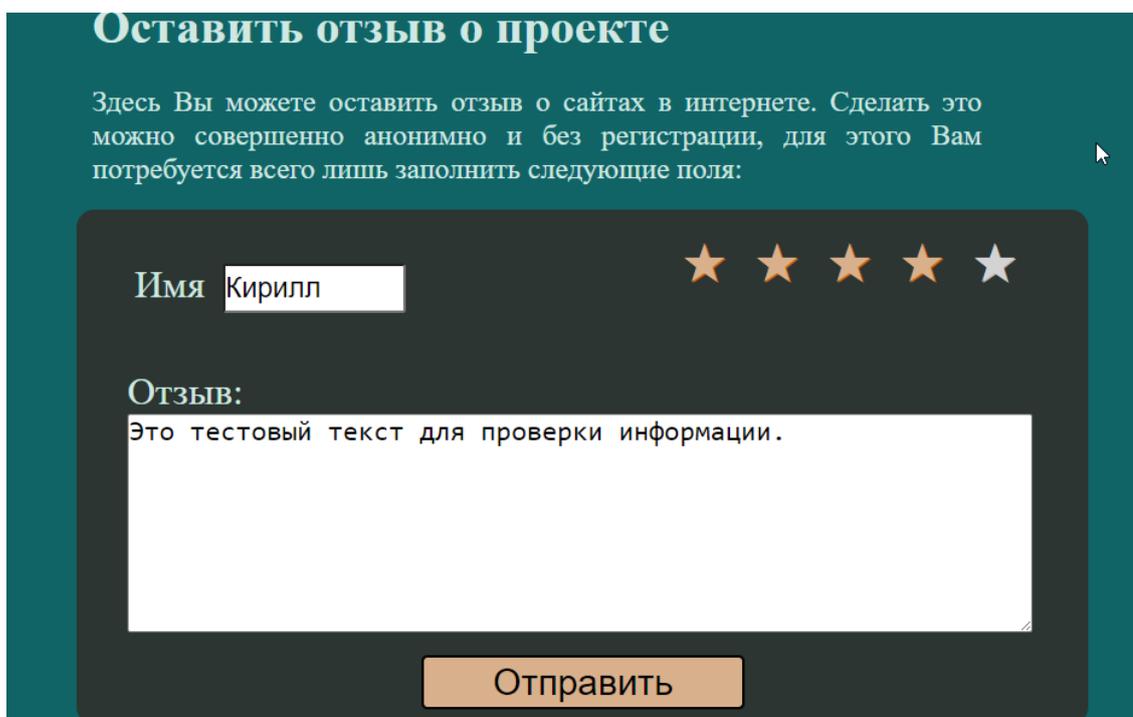


Рисунок 62 – Тестовый набор для отправки данных в СУБД

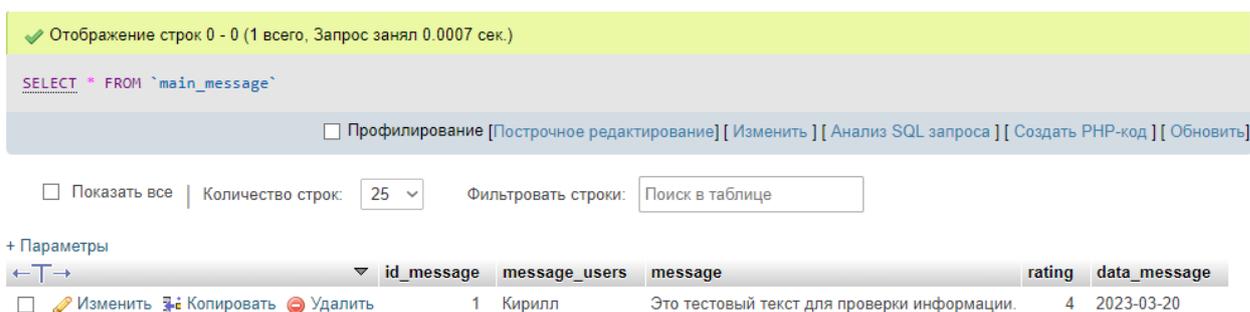


Рисунок 63 – Добавление данных в базу данных из формы отзывов.

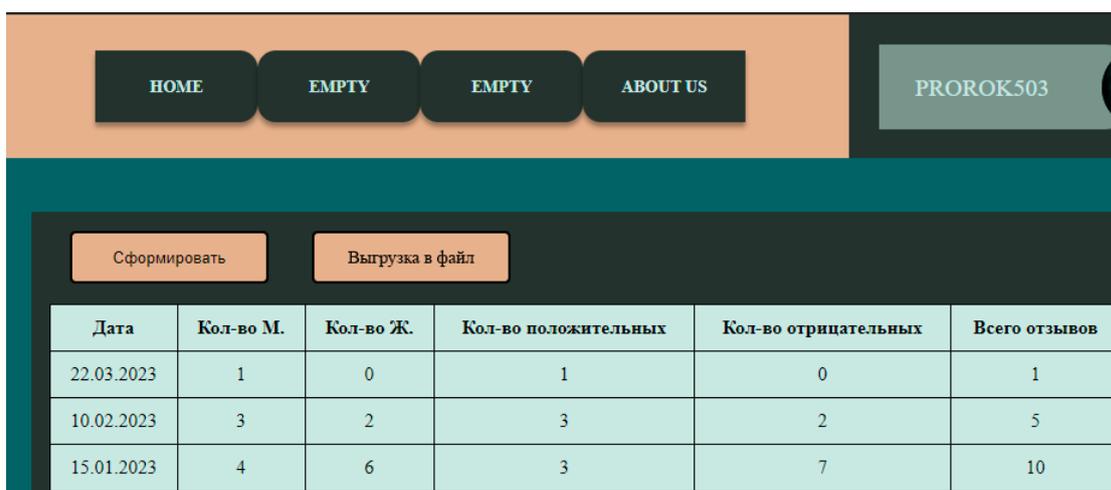


Рисунок 64 – Формируется данные за счет суммирование данных всех из таблиц.

Данный раздел наглядно показал принцип работы базы данных. В последствии в нее будет записано много различных данных, как для самой программы, так и для анализа.

3.9 Анализ данных отзывов

Анализ будет происходить с помощью выгрузки данных в отдельный файл, для последующей их обработки в программе MS Excel.

При нажатии на кнопку выгрузка в файл происходит перенос данных из базы данных в отдельный файл. После есть файл Excel, который собирает данных из файлов, которые выгружаются отдельно и происходит обновление данных.

Ниже на рисунке 65 будет отображаться интерфейс данного файла в excel.

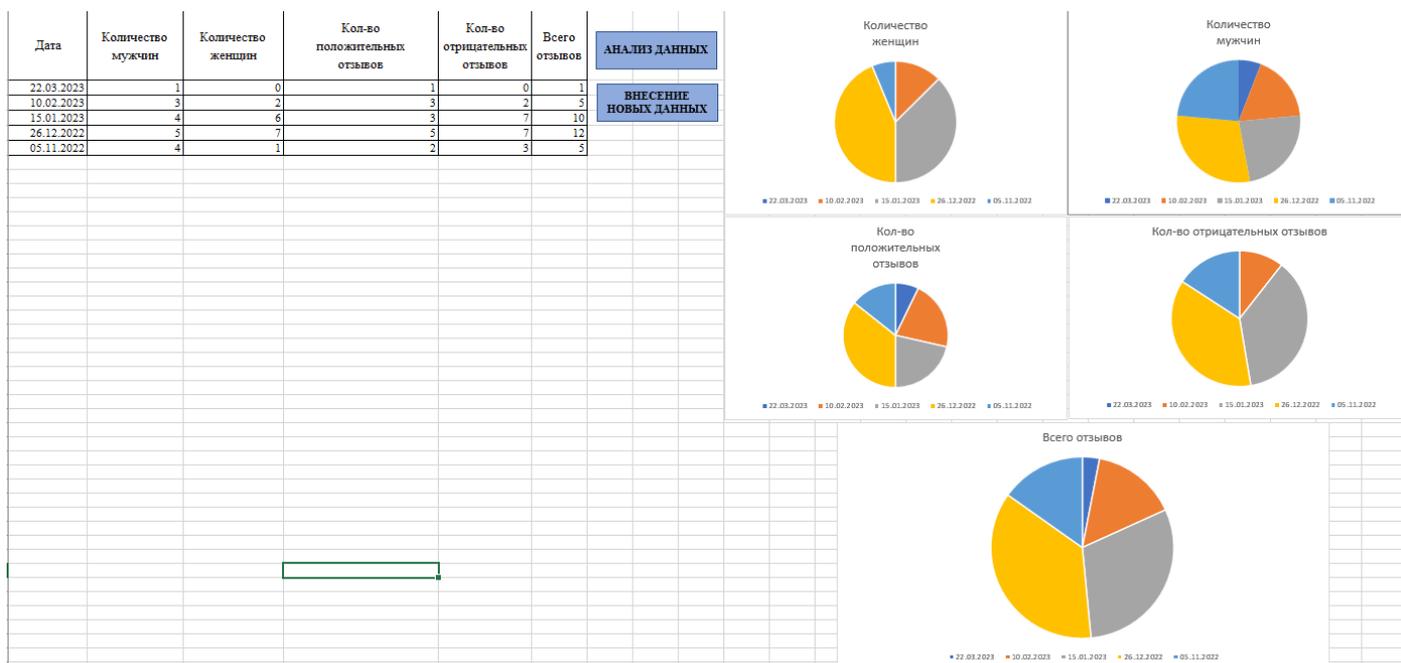


Рисунок 65 – Интерфейс анализа отзывов в excel.

Далее будет представлены диаграммы уже проанализированных ОТЗЫВОВ ПО МЕСЯЦАМ.

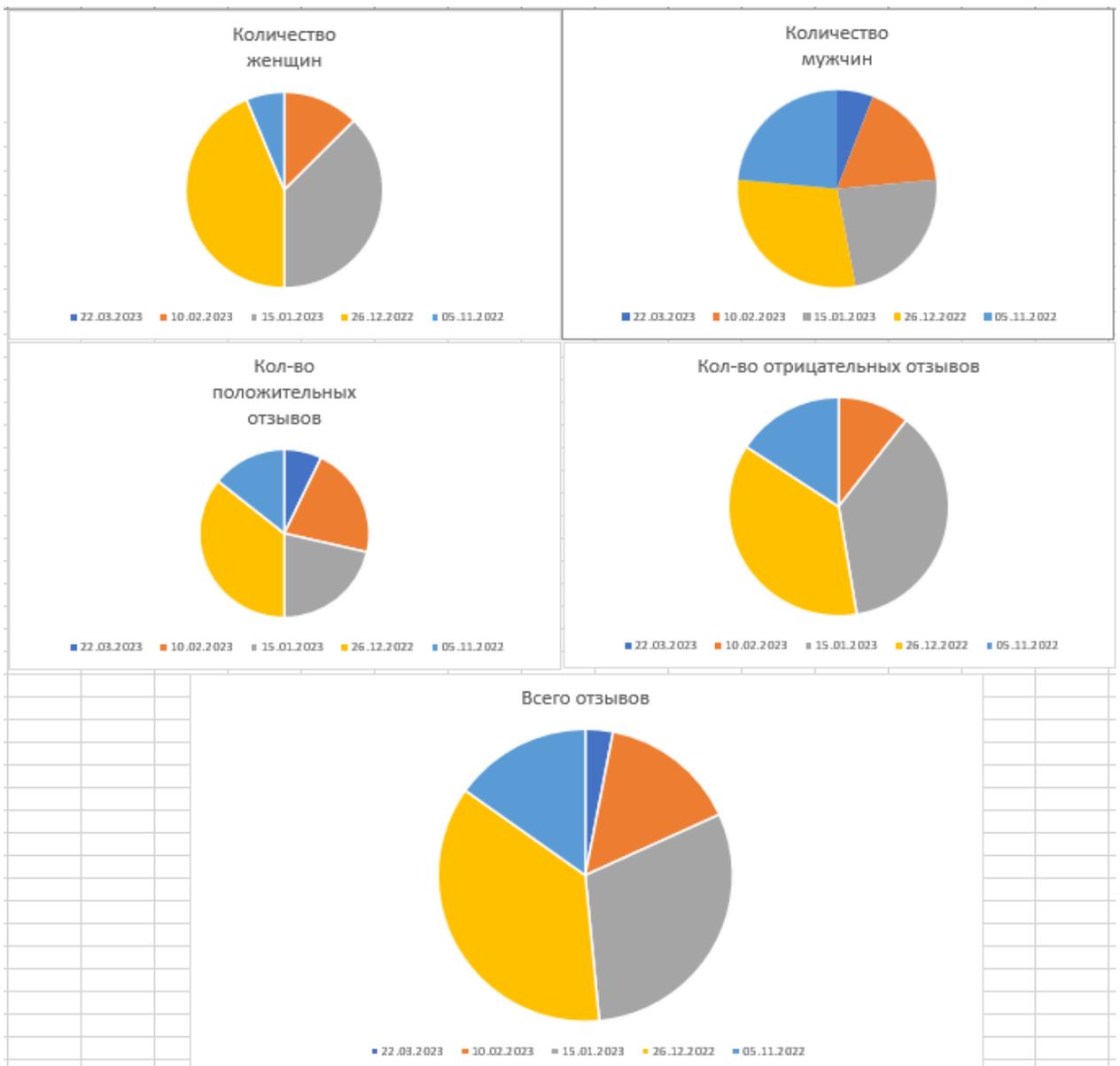


Рисунок 66 – Диаграммы анализа отзывов по месяцам.

Вывод главы 3

В ходе третьей главы была спроектирована концептуальная схема приложения на ней было подробно представлена схема взаимодействия всех модулей моей работы, после была выбрана и разработана база данных MySQL которая была протестирована в работе, при этом были представлены используемые инструменты для создания приложения и был разработан начальный интерфейс, который будет постепенно дорабатываться. Создана инструкция для каждого вида пользователя с определенными правами. Каждый пользователь ограничен только своими правами, из-за чего не будет проблем с неправильной работой всех функций. А после произведен небольшой эксперимент по анализу данных отзывов в программе MS excel. Который показал эффективно все данные, которые были получены из базы данных моей программы.

Заключение

Подведем итоги разработки программного обеспечения под названием разработка приложения для автоматизированного анализа отзывов клиентов.

В ходе прохождения учебной практики были выполнены такие задания как описание организационную структуру компании, в которой проходит учебную практику, анализ предметной области разработанного проекта, описание структуры отдела в котором проходил практику, проектирование концептуальной и функциональной модели проекта, анализ и разбор имеющихся проектов и создание критериев на разработку нового программного обеспечения. При всем этом были поставлены конкретные цели для разработки моей выпускной квалификационной работы, которая представлена в виде этапов.

В ходе разработки были описаны и выбраны архитектуры проекта, разработаны схемы программы, и функциональные элементы, на который подробно показаны все виды моей работы. При этом была выбрана среда разработки и предоставлены примеры работы программного обеспечения. Так же могу добавить, что среда разработки была качественно выбиралась, так как хотелось использовать все самые удобные и функциональные инструменты.

В ходе практики не были выполнены такие задачи как конечный выбор дизайна, так как для его составления нужно гораздо больше времени и требуется некоторого опроса или статистики от настоящих пользователей. Поэтому половина функций находится на стадии разработки таких как вкладок фото, видео, музыки и различных файлов. Безопасность информации находится на минимальном уровне, по этой причине пользоваться в открытом доступе нет возможности.

Конечный продукт еще находится в далеком будущем, но навыки, приобретенные в ходе данной практики уже закреплены и позволят развить этот проект в что-то новое при должной сноровке и времени. И будущие проекты будут выполнены гораздо быстрее и практичнее. Так же могу добавить, что данная работа будет иметь очень важный элемент моей жизни.

Список используемой литературы и используемых источников

1. Балдин К.В, Уткин В.Б. Информационные системы в экономике М.: Дашков и К, 2016. 395 с.
2. Вендров А.М. CASE–технологии. Современные методы и средства проектирования информационных систем. М.: Финансы и статистика, 1998. 176 с.
3. Гаевский А.Ю., А.Ю. Гаевский, В.А. Романовский. Создание Web–страниц и Web–сайтов. HTML и JavaScript М. : Триумф, 2015. 464 с.
4. Дакетт Джон HTML и CSS. Разработка и дизайн веб–сайтов Москва : Наука, 2014. 480 с.
5. Дронов В.А. PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web–сайтов М.: БХВ–Петербург, 2016. 820 с.
6. Колисниченко Д. PHP 5/6 и MySQL 6. Разработка Web–приложений М.: БХВ–Петербург, 2014. 560 с.
7. Клифтон Ян., Мовчан Д. А. Проектирование пользовательского интерфейса в Android М. : ДМК Пресс, 2017. 452 с.
8. Джеймс Ли Brent Уэр. Использование Linux, Apache, MySQL и PHP для разработки WEB–приложений М.: Вильямс, 2004. 429 с.
9. Малыхина М.П. Базы данных: основы, проектирование, использование СПб.: ВHV, 2007. 528 с.
10. Маркин А.В. Разработка отчетов в информационных системах М.: Диалог–МИФИ, 2012. 312 с.
11. Маклафлин Б. PHP и MySQL. Исчерпывающее руководство. СПб.: Питер, 2013. 512 с.
12. Морган С., Райан Б., Хорн Ш., Бломсма М.. Разработка распределенных приложений на платформе Microsoft .Net Framework. СПб.: Питер, 1–е издание, 2008 год, 608 с.
13. Перлова О.Н. Проектирование и разработка информационных систем М.: Академия, 2018. 272 с.

14. Прохоренок Н.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web–мастера СПб.: БХВ–Петербург, 2010. 912с.
15. Рудинский И. Д. Технология проектирования автоматизированных систем обработки информации и управления: учеб. пособие Москва: Горячая линия – Телеком, 2011. 304 с.
16. Семакин И.Г. Основы программирования М.: Издательский центр "Академия", 2008. 432с.
17. Сырых Ю. А. Современный веб–дизайн. Настольный и мобильный М. : Диалектика, 2019. 384 с.
18. Ташков П.А. Веб–мастеринг: HTML, CSS, JavaScript, PHP, CMS, графика, раскрутка – СПб.: Питер, 2010. 512 с.
19. Хоган Б. HTML5 и CSS3. Веб–разработка по стандартам нового поколения СПб.: Питер, 2014. 320 с.
20. Хомоненко А.Д. Базы данных: Учебник для высших учебных заведений СПб.: КОРОНА принт, 2009.– 672 с.
21. Шкрыль А. PHP – это просто. Програмируем для Web–сайта СПб.: БХВ–Петербург, 2006. 387 с.
22. Штайнер Г. HTML/XML/CSS. Справочник. Лаборатория Базовых Знаний, 2001. 436 с.
23. Янк К. PHP и MySQL. От новичка к профессионалу М.: Эксмо, 2013. 384 с.