

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«**СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, естественно-научных
и гуманитарных дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О. В. Папина
подпись
«_____» _____ 2023 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

Разработка информационной системы рейтингования органов региональной и
муниципальной власти Республики Хакасия

Руководитель _____ ст. преподаватель В. И. Кокова
подпись, дата

Выпускник _____ П. Д. Михайлов
подпись, дата

Консультанты
по разделам:

Экономический _____ М. А. Бурева
подпись, дата

Нормоконтролер _____ А. Н. Кадычегова
подпись, дата

Абакан 2023

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, естественно-научных
и гуманитарных дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О. В. Папина
подпись
«_____» _____ 2023 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Абакан 2023

Студенту Михайлову Павлу Дмитриевичу

Группа ХБ 19-02

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Разработка информационной системы рейтингования органов региональной и муниципальной власти Республики Хакасия

Утверждена приказом по институту № 283 от 11.05.2023 г.

Руководитель ВКР: В. И. Кокова, ст. преподаватель, ХТИ – филиал СФУ

Исходные данные для ВКР: заказ Центра управления регионом Республики Хакасия

Перечень разделов ВКР:

1. Анализ предметной области. Выбор средств проектных решений.
2. Разработка информационной системы.
3. Определение совокупной стоимости владения проектом.

Перечень графического материала: нет

Руководитель ВКР

подпись

В. И. Кокова

Задание принял к исполнению

подпись

П. Д. Михайлов

«11» мая 2023 г.

РЕФЕРАТ

Выпускная квалификационная работа (ВКР) по теме «Разработка информационной системы рейтингования органов региональной и муниципальной власти Республики Хакасия» содержит 60 страниц текстового документа, 11 использованных источников, 33 иллюстрации, 5 таблиц, 5 формул.

ЧАТ-БОТ, ВЕБ-ПРИЛОЖЕНИЕ, TELEGRAM, TELEGRAM API, PHP, IDEF0, IDEF3, DFD, ТАБЛИЦЫ, ГРАФИКИ, MYSQL, БАЗЫ ДАННЫХ, ТСО, РИСКИ, РЕЙТИНГОВАНИЕ.

Объект выпускной квалификационной работы: деятельность Центра управления регионом Республики Хакасия (ЦУР РХ) по рейтингованию работы органов муниципальной и региональной власти.

Цель выпускной квалификационной работы: разработка информационной системы для автоматизации рейтингования органов региональной и муниципальной власти Республики Хакасия.

Для достижения цели необходимо выполнить следующие задачи:

- Проанализировать предметную область, деятельность ЦУР РХ и аналогичные программные продукты на рынке.
- Выбрать программные средства для разработки системы.
- Спроектировать и создать базу данных для системы.
- Разработать Telegram-бота для сбора рейтинга и формирования отчета.
- Разработать веб-приложение для формирования графиков и таблиц.
- Оценить затраты на реализацию информационной системы.

В результате была разработана информационная система, состоящая из Telegram-бота для сбора оценок и веб-приложения для наглядного отображения полученной информации. Был обоснован выбор языка программирования PHP и среды Sublime. По методике оценки совокупной стоимости проекта (ТСО) были рассчитаны затраты на разработку проекта.

SUMMARY

The graduation thesis is «IT system for rating regional and municipal authorities in Republic of Khakassia». It consists of 63 pages, 11 references, 33 figures, 5 tables, 5 formulae.

CHAT-BOT, WEB APPLICATION, TELEGRAM, TELEGRAM API, PHP, IDEF0, IDEF3, DFD, TABLES, GRAPHS, MYSQL, DATABASES, TCO, RISKS, RATING

The object of the graduation work is the activity of the Regional Management Center of the Republic of Khakassia (RMC RKh) in rating the work of municipal and regional authorities.

The purpose of the graduation work is to develop an IT system for automating the rating of regional and municipal authorities of the Republic of Khakassia.

Objectives:

- To analyze the subject area, the activity of the RMC RKh, and similar software products on the market.
- To select software tools for system development.
- To design and to create a database for the system.
- To develop a Telegram-bot for collecting ratings and generating reports.
- To develop a web application for creating graphs and tables.
- To calculate the costs of implementing the IT system.

The IT system has been developed; it consists of a Telegram-bot for collecting ratings and the web application for visualizing the collected information. The choice of the PHP programming language and the Sublime environment has been verified. The costs of the project development have been calculated through the Total Cost of Ownership (TCO) estimation methodology.

English language supervisor

signature, date

N. V. Chezybaeva

СОДЕРЖАНИЕ

Введение.....	6
1 Анализ предметной области и выбор средств разработки	8
1.1 Характеристика деятельности Центра управления регионом Республики Хакасия.....	8
1.2 Постановка цели и задач выпускной квалификационной работы	10
1.3 Концепция разработки чат-бота и веб-приложения по рейтингованию органов власти	10
1.4 Анализ аналогичных программных продуктов	12
1.5 Моделирование процесса рейтингования в нотации IDEF0.....	15
1.6 Моделирование действий пользователя в нотации IDEF3	19
1.7 Моделирование потоков данных в нотации DFD	20
1.8 Обоснование выбора средств разработки	23
1.9 Выводы по разделу «Анализ предметной области и выбор средств разработки».....	26
2 Разработка информационной системы рейтингования	27
2.1 Проектирование базы данных	27
2.2 Разработка Telegram-бота для сбора рейтинга.....	28
2.2.1 Реализация процесса сбора рейтинга	28
2.2.2 Формирование отчета в Excel	32
2.3 Разработка веб-приложения для отображения графиков и таблиц...	37
2.3.1 Определение структуры веб-приложения	37
2.3.2 Создание графиков с помощью Morris.js.....	39
2.3.3 Создание таблиц с помощью PHP	42

2.4 Выводы по разделу «Разработка информационной системы рейтингования».....	45
3 Определение совокупной стоимости владения проектом.....	46
3.1 Оценка затрат на средства разработки.....	46
3.2 Расчет капитальных затрат.....	47
3.3 Расчет эксплуатационных затрат.....	51
3.4 Расчет прямых расходов.....	53
3.5 Расчет совокупной стоимости владения системой.....	54
3.6 Оценка рисков реализации проекта	55
3.7 Выводы по разделу «Определение совокупной стоимости владения проектом».....	56
Заключение	58
Список использованных источников	60

ВВЕДЕНИЕ

Центр управления регионом Республики Хакасия (ЦУР РХ) является связующим звеном между органами власти, людьми и властью Республики Хакасия. Он создает информационные системы для удобного взаимодействия людей с органами власти, автоматизирует процессы создания отчета и сбора информации для высшего должностного лица и создает необходимые приложения для органов власти региона.

Одним из внутренних процессов ЦУР РХ является рейтингование органов власти – это процесс оценивания деятельности каждого органа власти в регионе и передача этой информации высшему должностному лицу (ВДЛ) для дальнейших мер.

Объектом ВКР является деятельность Центра управления регионом Республики Хакасия по рейтингованию работы органов муниципальной и региональной власти.

Необходимо создать систему, которая будет автоматизировать сбор оценок от специалистов ЦУР РХ по каждому органу власти, составлять отчет для ВДЛ и формировать график с таблицей для наглядного отображения работы органов власти в регионе. Создание такой системы позволит освободить человеческие ресурсы, выделявшиеся на эти процессы, а также увеличит скорость и точность работы с информацией за счет минимизации человеческого фактора.

Целью выпускной квалификационной работы является разработка информационной системы для автоматизации рейтингования органов региональной и муниципальной власти Республики Хакасия для выпускной квалификационной работы.

Для достижения цели необходимо выполнить следующие задачи:

- Проанализировать предметную область, деятельность ЦУР РХ и аналогичные программные продукты на рынке.
- Выбрать программные средства для разработки системы.

- Спроектировать и создать базу данных для системы.
- Разработать Telegram-бота для сбора рейтинга и формирования отчета.
- Разработать веб-приложение для формирования графиков и таблиц.
- Оценить затраты на реализацию информационной системы.

1 Анализ предметной области и выбор средств разработки

1.1 Характеристика деятельности Центра управления регионом Республики Хакасия

Центр управления регионом Республики Хакасия (ЦУР РХ) – специализированный пункт управления, предназначенный для обеспечения деятельности Главы Республики Хакасия – Председателя Правительства Республики Хакасия, Правительства Республики Хакасия при реализации функций мониторинга и координации обработки обращений и сообщений граждан, объединений граждан или юридических лиц (далее – обращения и сообщения); оценки, анализа и прогнозирования ситуации, стратегического, текущего и оперативного планирования, мониторинга и контроля исполнения управленческих решений [1].

ЦУР РХ включает в себя:

- 1) специально оборудованные помещения;
- 2) технические средства ЦУР РХ, включающие специализированный программно-технический комплекс средств обработки информации и коллективного отображения, телекоммуникационную инфраструктуру ЦУР РХ, систему защиты информации;
- 3) информационные системы, используемые для сбора, обработки, анализа и представления информации.

ЦУР РХ предназначен для решения следующих основных задач:

- 1) координация работы по мониторингу и обработке обращений и сообщений, направленных в исполнительные органы государственной власти Республики Хакасия, органы местного самоуправления муниципальных образований Республики Хакасия (далее – органы местного самоуправления) или должностным лицам указанных органов в письменной форме или в форме электронного документа;
- 2) мониторинг и контроль исполнения управленческих решений;

3) мониторинг исполнения планов и программ, а также достижения установленных ими значений целевых показателей;

4) обеспечение сбора, обработки и анализа информации, характеризующей общественно-политическую, социально-экономическую и информационную обстановку в Республике Хакасия и необходимой для планирования, подготовки и принятия управленческих решений;

5) обеспечение информационной поддержки принятия управленческих решений, проведение совещаний в очном и дистанционном режимах;

6) обеспечение необходимого уровня защиты информации, обрабатываемой в ЦУР РХ.

Миссия ЦУР РХ – сократить расстояние между жителями и властью, проблемой и её устранением.

ЦУР РХ оснащен современным аппаратным и официальным программным обеспечением, что позволяет поддерживать в рабочем состоянии сотни веб-приложений и ботов. Основным языком, на котором ведется разработка в ЦУРе РХ, является PHP. В качестве СУБД используется MySQL.

ЦУР РХ является структурным подразделением автономной некоммерческой организации (АНО) по развитию цифровых проектов в сфере общественных связей и коммуникаций «Диалог Регионы». Структура организации изображена на рисунке 1.

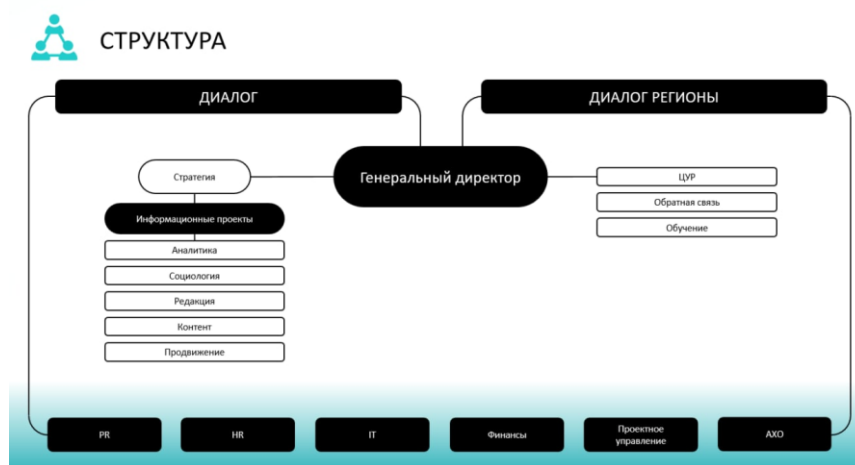


Рисунок 1 – Структура АНО «Диалог»

1.2 Постановка цели и задач выпускной квалификационной работы

Целью выпускной квалификационной работы является разработка информационной системы для автоматизации рейтингования органов региональной и муниципальной власти Республики Хакасия.

Для достижения цели необходимо выполнить следующие задачи:

- Проанализировать предметную область, деятельность ЦУР РХ и аналогичные программные продукты на рынке.
- Выбрать программные средства для разработки системы.
- Спроектировать и создать базу данных для системы.
- Разработать Telegram-бота для сбора рейтинга и формирования отчета.
- Разработать веб-приложение для формирования графиков и таблиц.
- Оценить затраты на реализацию информационной системы.

1.3 Концепция разработки чат-бота и веб-приложения по рейтингованию органов власти

Объектом исследования является деятельность Центра управления регионом Республики Хакасия по рейтингованию органов власти.

Органы власти делятся на две группы: ОМСУ и РОИВ.

ОМСУ – органы местного самоуправления, например, администрация города Абакана или администрация города Черногорска.

РОИВ – региональные органы исполнительной власти субъекта РФ, например, Министерство образования Республики Хакасия или Государственный комитет энергетики и тарифного регулирования Республики Хакасия.

Рейтингование происходит следующим образом: за каждым органом власти закрепляется сотрудник ЦУР РХ, активно взаимодействующий с ним и полу-

чающий информацию о результатах деятельности; исходя из результатов деятельности, сотрудник выставляет оценку работе закрепленного за ним органа власти по заранее установленным критериям, требуемым в отчете.

В конце месяца для каждого органа подсчитывается средняя оценка и строится график деятельности. Вся полученная статистика заносится в отчет, который затем передается высшему должностному лицу региона.

Подсчетом и сбором оценок деятельности органов власти занимается отдельный сотрудник ЦУР РХ, который формирует отчет ВДЛ вручную. Весь процесс происходит в чате Telegram, в котором сотрудники отправляют сообщения с наименованием оцениваемого органа власти, критерием, оценкой и описанием к ней. В конце каждого месяца ответственный за отчет сотрудник собирает эти сообщения, вручную заносит данные и рассчитывает сумму оценок, а затем строит графики. Сформировав отчет, сотрудник отправляет его ВДЛ.

Проблемой текущего процесса является высокая нагруженность сотрудника, формирующего отчет, и большая вероятность возникновения ошибок с заполнением или расчетами, поскольку сотрудник работает с большим объемом данных. Также сотрудникам, оценивающим деятельность, было бы куда удобнее не самим вводить наименования органов власти и критериев, а выбирать их из заранее готового списка. Таким образом, увеличится скорость и точность сбора и формирования статистики для ВДЛ.

Для устранения этих недостатков было решено создать систему для рейтингования органов власти и автоматического формирования отчета, графиков и таблиц.

Система будет состоять из двух частей: Telegram-бота и веб-приложения. Для сбора статистики выбран мессенджер Telegram, поскольку изначальный чат по сбору рейтинга так же находился в Telegram. Для отображения графиков и таблиц было решено создать веб-приложение, чтобы в любой момент можно было наглядно оценить текущую работу органов власти.

Основные функции системы:

1. Сбор оценок пользователей о деятельности органов власти.

2. Сохранение оценок в базе данных.
3. Формирование отчета за месяц, год и для печати.
4. Построение графиков на отдельной веб-странице.

Требования пользователей:

1. Возможность выбирать группу органов власти (ОМСУ или РОИВ), конкретный орган из выбранной ранее группы, а также критерии посредством кнопок.
2. Возможность вводить числовую оценку деятельности, а также оставлять отзыв.
3. Возможность просматривать статистику в виде графиков и таблиц на отдельной веб-странице.
4. Возможность остаться в той же среде (не менять мессенджер, все как раньше в Telegram, только с ботом).
5. Разграничение доступа для администраторов (людей, кто может формировать отчет) и обычных пользователей (кто может только оценивать).

В итоге реализации проекта должны получиться:

- Telegram-бот, реализующий функцию выставления оценки через кнопки, сохраняющий всю информацию в базе данных, которую затем можно сформировать в отчет через отдельную кнопку.
- Веб-приложение для визуализации, создающее наглядные графики и таблицы с информацией о деятельности органов власти.

1.4 Анализ аналогичных программных продуктов

Одним из самых популярных инструментов для создания голосований (рейтингование является видом голосования с выставлением оценки) является сервис Google Forms от компании Google. Это онлайн сервис для создания форм для голосований, которые затем можно отправлять другим людям, чтобы узнать их мнение/оценку. Интерфейс сервиса представлен на рисунке 2.

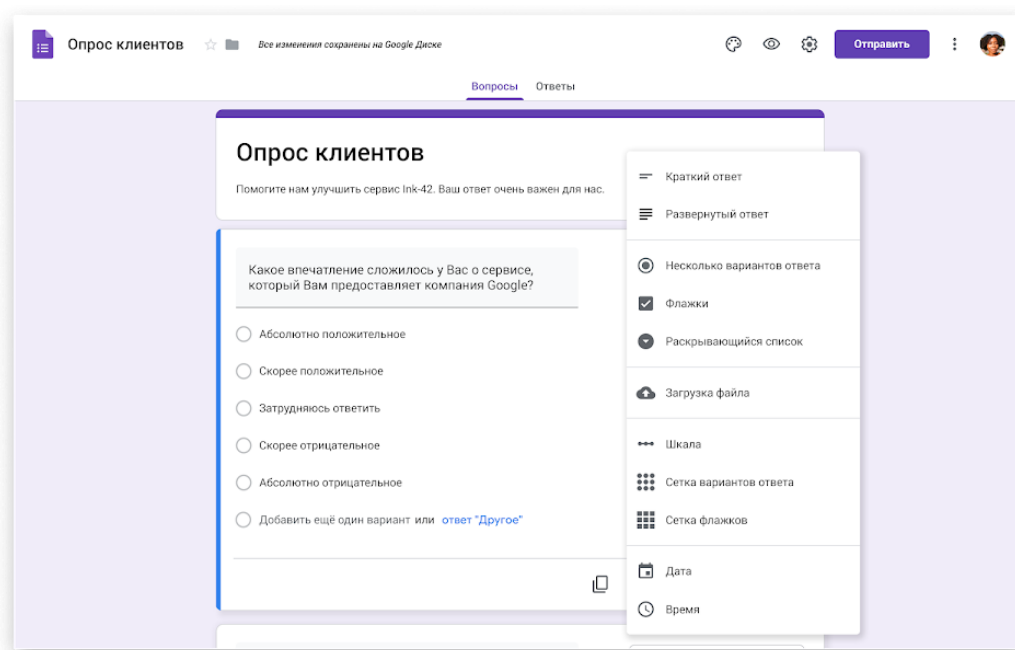


Рисунок 2 – Интерфейс Google Forms

Сервис позволяет выгружать данные в Excel таблицы, однако изменить вид таблицы, чтобы она подходила под требования официального отчета, нельзя. В данном случае важна структура таблицы, поскольку результаты формируются в отчет для высших должностных лиц. Также сервисом владеет компания Google, и все данные также хранятся на их серверах, поэтому использовать зарубежное программное обеспечение для автоматизации внутренних процессов взаимодействия органов власти не желательно. Из-за этих недостатков необходимо поискать другой вариант или разработать свой.

В качестве отечественного аналога рассмотрим сервис Testograf (рисунок 3).

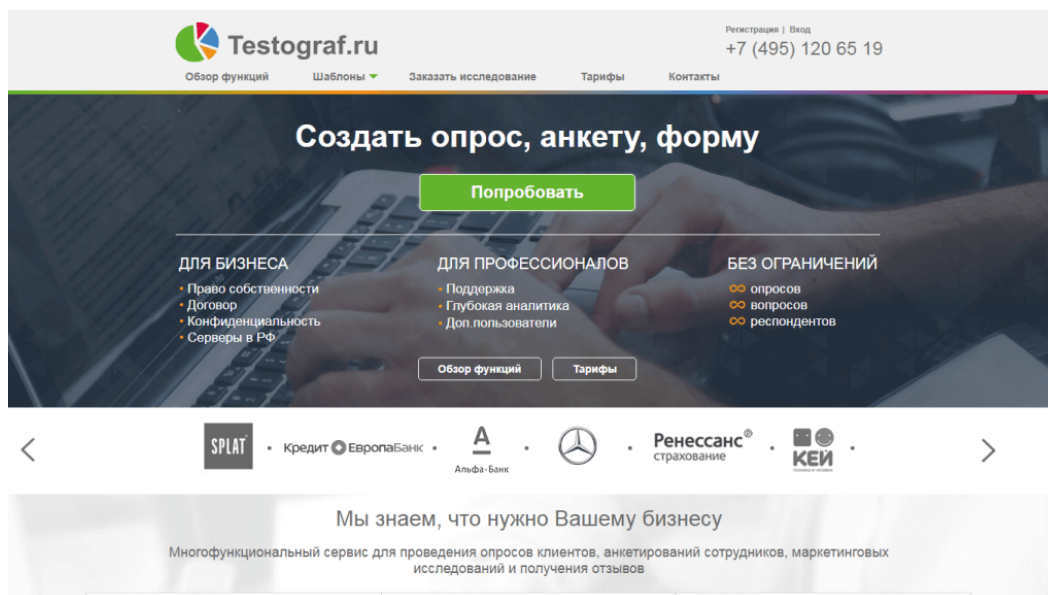


Рисунок 3 – Сервис создания опросов Testograf

Данный сервис зарегистрирован в России (домен ru), поэтому информация о данных вряд ли уйдет за пределы страны (проще будет договориться на выделение своих серверов). Однако и у этого варианта есть недостатки. Во-первых, все та же проблема с выводом в Excel нужного отчета. Во-вторых, данный сервис, в отличие от Google Forms, является платным, поэтому необходимо ежемесячно оплачивать подписку для полноценного использования. Рейтингование проходит регулярно каждый месяц, поэтому затраты на подписку на дистанции могут очень дорого стоить [2].

Таким образом, рассмотрев два готовых варианта, было сделано несколько выводов:

- во-первых, нет сервисов, позволяющих удобно формировать отчет в Excel;
- во-вторых, большинство бесплатных сервисов принадлежат зарубежным компаниям;
- в-третьих, затраты на подписку со временем превысят затраты на создание собственного программного продукта.

Поэтому было решено создать собственную систему для рейтингования органов власти.

1.5 Моделирование процесса рейтингования в нотации IDEF0

Нотация IDEF0 позволяет отобразить структуру процесса рейтингования, из каких действий он состоит и как эти действия связаны потоками информации.

В данном случае нотация будет использоваться для создания моделей «Как есть» и «Как будет» [3].

«Как есть» – модель исходного бизнес-процесса, создаваемая из исходных данных о процессе, собранных в результате работы. В результате моделирования можно выявить одно или несколько действий, которые являются лишними или могут быть автоматизированы, что позволит установить необходимость изменения процесса.

На рисунке 4 представлен первый уровень модели «Как есть».

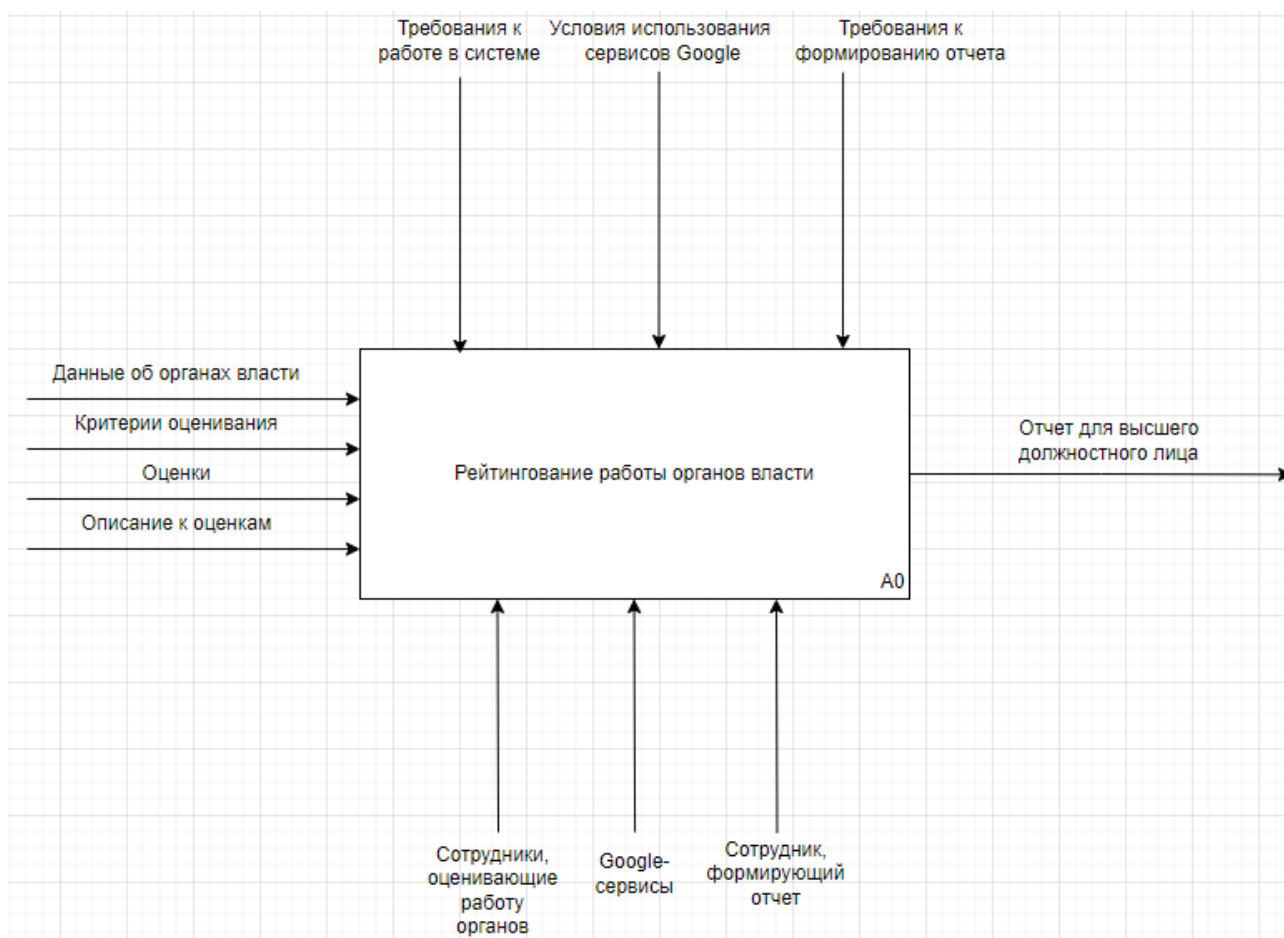


Рисунок 4 – Первый уровень модели «Как есть»

Декомпозиция представлена на рисунке 5.

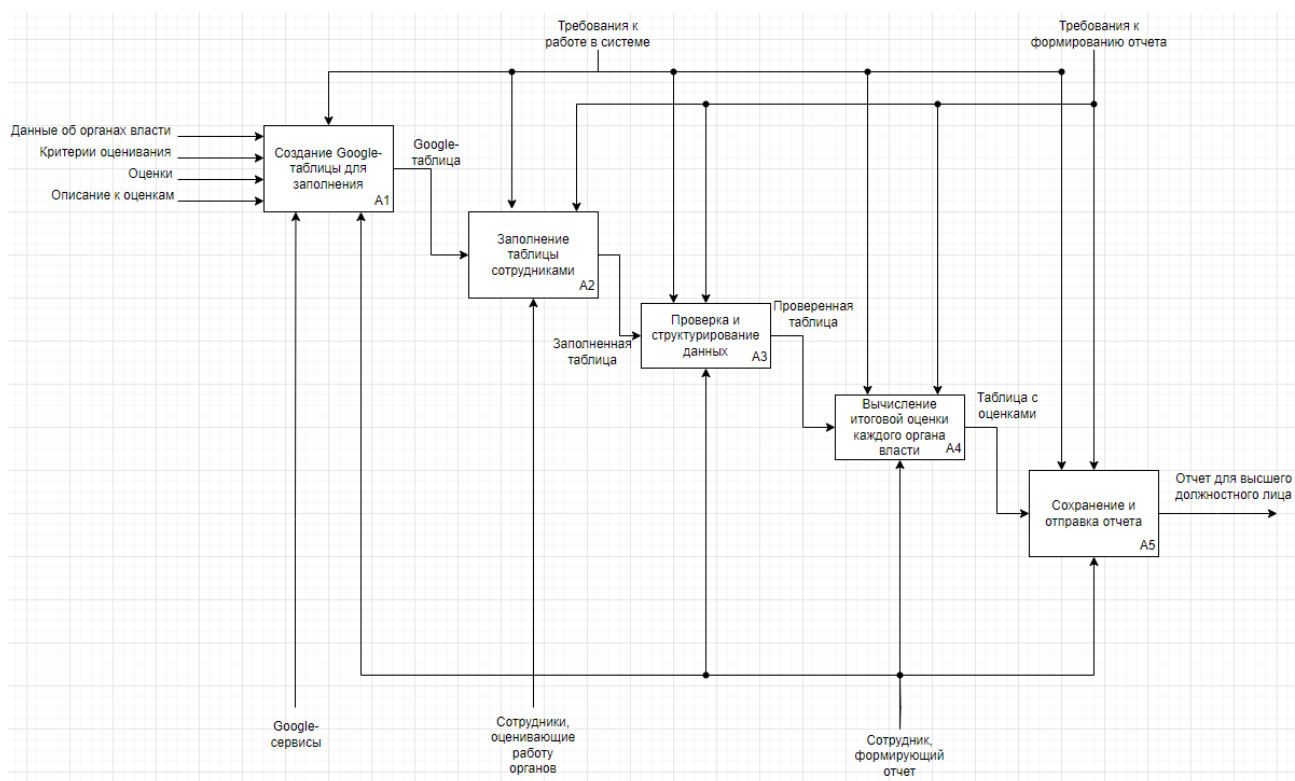


Рисунок 5 – Декомпозиция процесса A0 в модели «Как есть»

Из декомпозиции видно, что внутри бизнес-процесса почти каждое действие выполняет сотрудник, ответственный за формирование отчета для высшего должностного лица. Ему необходимо создать Google-таблицу, проверить правильность заполнения таблицы сотрудниками (например, не забыли ли они оставить описание к оценке), затем структурировать полученные данные и рассчитать итоговые оценки каждого органа власти.

Таким образом, на формирование отчета уходит очень много времени и не исключаются ошибки, связанные с человеческим фактором (опечатки, пропущенные сообщения, ошибки в вычислениях и т.п.). Можно увеличить число сотрудников для формирования отчета, однако это увеличит затраты и добавит проблем с синхронизацией работы нескольких сотрудников над одним отчетом. Построение графиков и таблиц также отсутствует в модели «Как есть», поскольку

ответственный за отчёт сотрудник занят формированием отчёта для ВДЛ. Необходимо автоматизировать этот процесс, чтобы система сама сохраняла сообщения, отбирала нужные по дате, рассчитывала итоговую оценку и строила графики с таблицами. Сотрудник будет меньше нагружен, поскольку ему останется лишь запросить отчет у программы, уменьшится шанс допустить ошибку и значительно увеличится скорость формирования отчета.

«Как будет» – модель нового бизнес-процесса, в которой действия по сбору оценок, их расчету и построению графиков были автоматизированы. Теперь данные действия выполняет система, а сотрудник назначается администратором системы и отслеживает корректность её работы. Первый уровень модели представлен на рисунке 6.

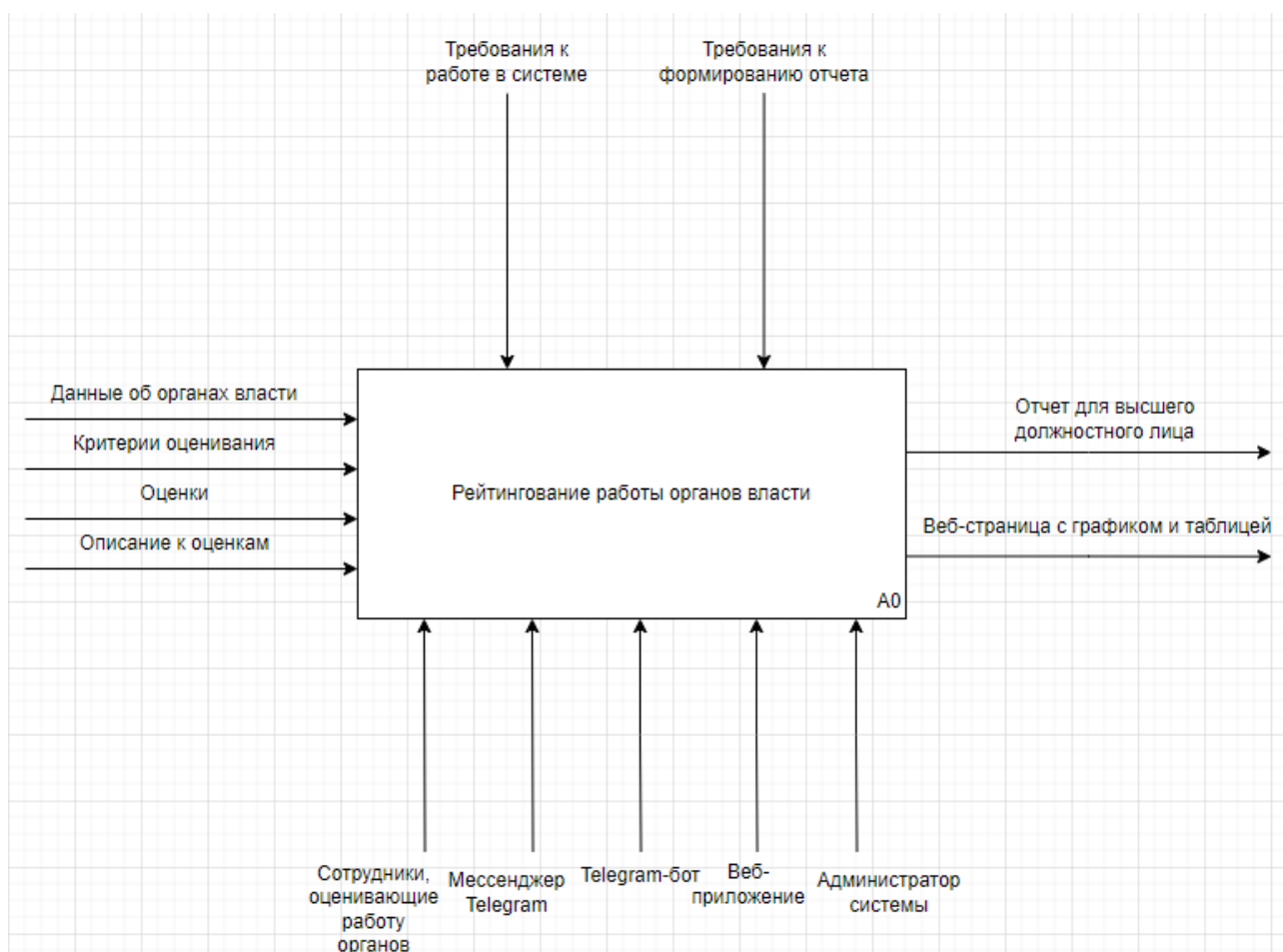


Рисунок 6 – Первый уровень модели «Как будет»

Декомпозиция представлена на рисунке 7.

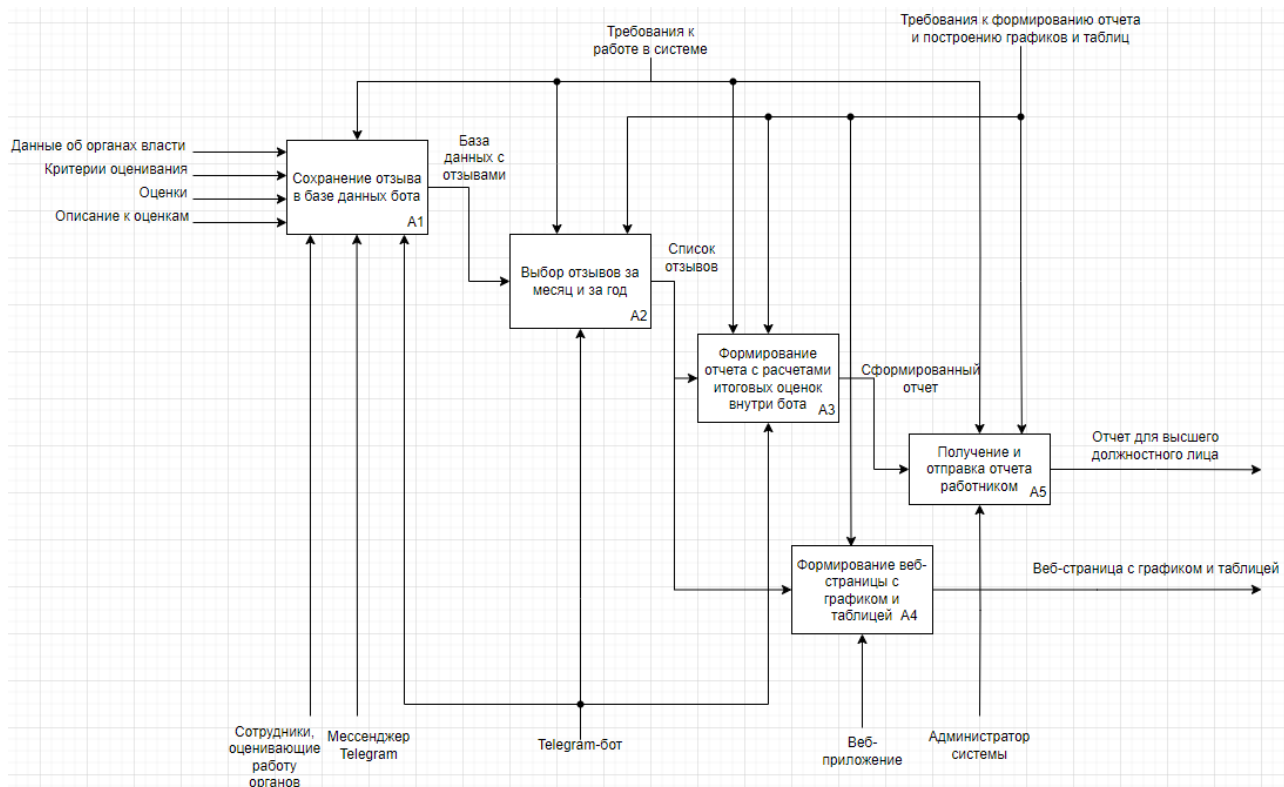


Рисунок 7 – Декомпозиция бизнес-процесса в модели «Как будет»

В результате все действия по внесению данных в таблицу и расчета итоговых оценок были совмещены в одно – формирование отчета. Это действие выполняет Telegram-бот, при этом сотрудник больше никак не участвует в данном процессе, а лишь получает необходимый отчет. За построение графиков и таблиц теперь отвечает веб-приложение, которое формирует веб-страницу по собранным данным от Telegram-бота. Также появилась возможность выбирать период отчетности. Раньше, чтобы сформировать отчет за год, необходимо было взять все предыдущие отчеты и сформировать один общий. Теперь же необходимо просто выбрать даты и бот сам достанет все сообщения за указанный период в кратчайшие сроки.

Только у администратора есть доступ к формированию отчета и открытию/закрытию периодов рейтингования.

1.6 Моделирование действий пользователя в нотации IDEF3

Моделирование действий пользователя позволяет ещё на стадии планирования разобраться в устройстве проектируемой системы, понять, как будет работать система, взаимодействие её компонентов [4].

Одной из таких моделей является модель действий пользователя в системе. В ней отражены всевозможные действия пользователя в системе. С помощью такой модели можно легко отследить последовательность действий, что из чего следует. Такая модель особенно важна в случае с ботом, поскольку необходимо сохранять правильный порядок при взаимодействии системы с пользователем, выводить данные или просто ввести исходные данные только тогда, когда будет необходимо. Модель была построена с помощью нотации IDEF3 и отражает работу пользователя в Telegram-боте (рисунок 8).

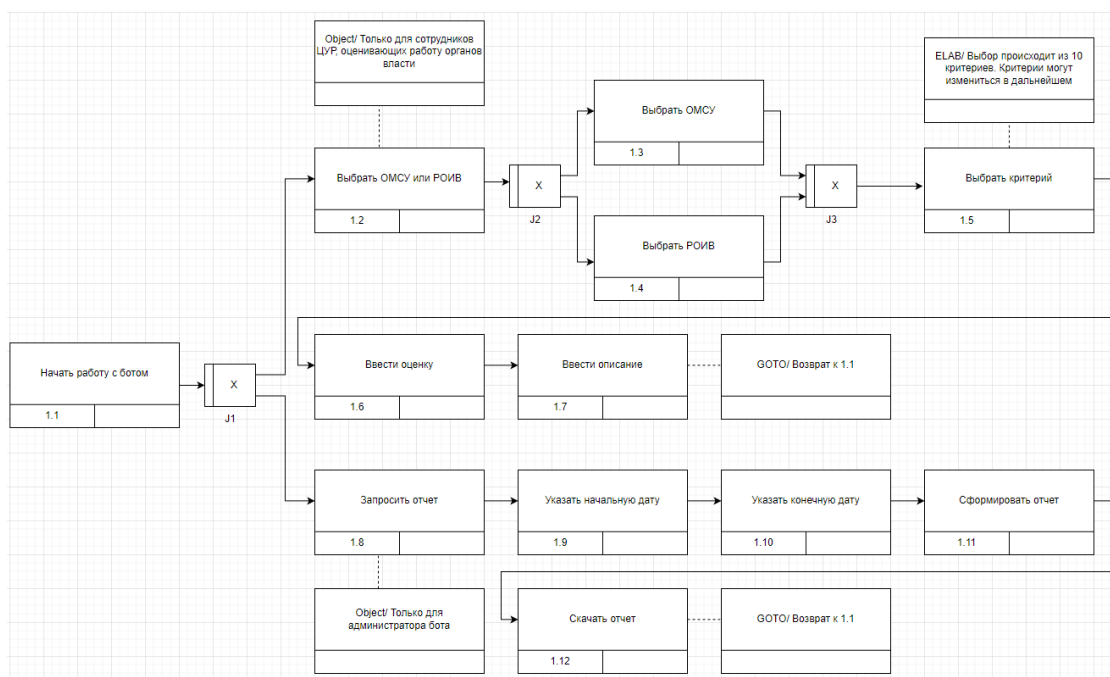


Рисунок 8 – Модель действий пользователя при работе с ботом

Любое взаимодействие с системой начинается с входа в систему, в случае с ботом это начать чат с ботом. Далее есть два варианта действий пользователя:

выставить оценку о деятельности органа власти или запросить отчет по уже сохраненным отзывам.

Для запроса необходимо обладать правами администратора (права будут устанавливаться как поле в базе данных). Для получения отчета необходимо указать дату – период времени для формирования отчета. Затем система сформирует отчет и отдаст пользователю для скачивания. Далее система возвращается в исходную позицию с возможностью выбрать действие.

Деятельность органов власти оценивают сотрудники ЦУР, закрепленные за каждым органом в регионе. Сначала пользователь выбирает один из двух видов органов власти: региональный (РОИВ) или местного самоуправления (ОМСУ). Данный выбор позволяет сократить время поиска конкретного органа власти, поскольку они разделены по видам. Затем из предложенного списка выбрать орган власти, деятельность которого предстоит оценить, далее выбрать критерий из перечня (перечень устанавливается требованиями к отчету), ввести оценку и описание (описание отвечает на вопрос: «почему именно такая оценка?»). После ввода описания отзыв считается завершенным и сохраняется системой, пользователя возвращает обратно к выбору, где он может оставить ещё одну оценку, если это потребуется. Сотрудник может выставить несколько оценок по одну и тому же критерию, поскольку за период рейтингования оцениваемый орган власти может провести много важных работ, результаты которых необходимо внести в отчет. Ответственность за «накрутку» оценок лежит на самих пользователях, поэтому в случае «накрутки» оценок нарушителя можно будет найти по записям в базе данных.

1.7 Моделирование потоков данных в нотации DFD

Диаграмма потоков данных (DFD) позволяет отследить то, как система взаимодействует с внешними сущностями, а также то, как протекает информация внутри неё.

В данном случае внешними сущностями являются сотрудники ЦУР РХ, оценивающие деятельность органов власти, и администратор системы, запрашивающий и получающий отчеты о деятельности органов власти в регионе.

От сотрудников боту при нажатии кнопок в чате и отправки сообщений поступают вид органа власти, наименование, оценка и описание к ней. Бот после сохранения отзыва возвращает пользователю сообщение об успешном сохранении.

Администратор в конце каждого месяца запрашивает отчет у бота, скачивает его и отправляет ВДЛ.

Контекстная модель бота изображена на рисунке 9.

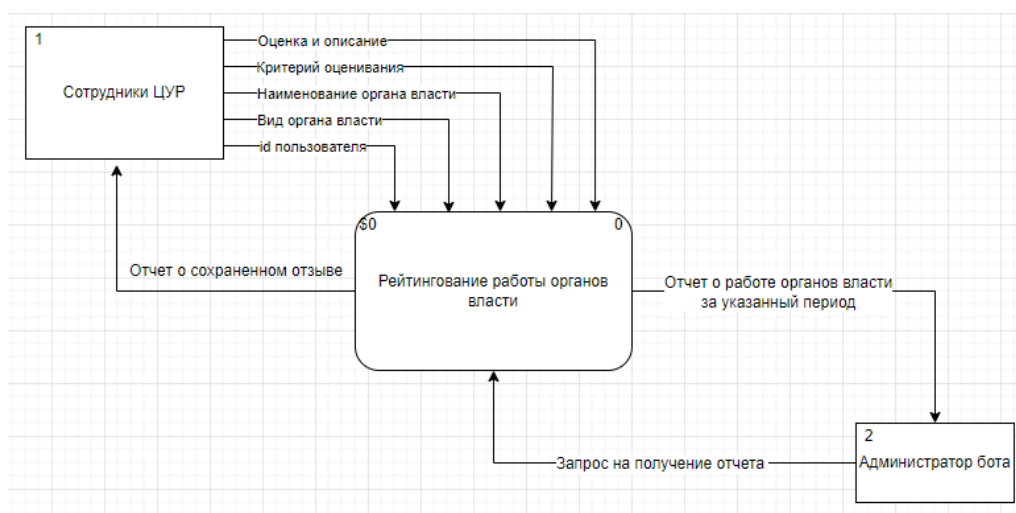


Рисунок 9 – Контекстная модель чат-бота по рейтингованию

Далее стоит рассмотреть, как протекает информация внутри бота. При запуске бота от пользователя поступает id его чата с ботом (на схеме изображено id пользователя, поскольку для бота этот id позволяет идентифицировать нужный чат с пользователем). С помощью этого id бот сверяет данные в базе доступных пользователей и возвращает найденного. В случае если пользователь не найден, из базы данных ничего не возвращается и бот не отвечает и не выполняет команды в данном чате. Для авторизованного пользователя создается пустая форма для отзыва, в которую по ходу диалога будут вноситься данные. Заполнение

формы начинается с ввода вида органа власти (ОМСУ или РОИВ). По выбранному виду бот запрашивает перечень органов власти. Для этого в базе данных есть специальная таблица с информацией об органах власти в регионе. Из полученного перечня пользователь выбирает орган власти, который будет оценивать. Затем, аналогично, пользователь выбирает критерий из перечня, составленного на основе таблицы в базе данных. Все результаты заносятся в форму, в которую остается ввести оценку и описание. Затем эта форма собирается в отзыв, который сохраняется в базе данных. Пользователю отправляется сообщение о том, что отзыв сохранен.

Раз в месяц администратор запрашивает отчет у бота. Из базы данных запрашиваются все отзывы за месяц и за прошедший год, которые затем помещаются в файл. Далее происходит автоматический расчет итоговых оценок. Полученные данные формируются в отчет для ВДЛ, который затем отправляется на скачивание администратору.

Модель потоков данных внутри системы изображена на рисунке 10.

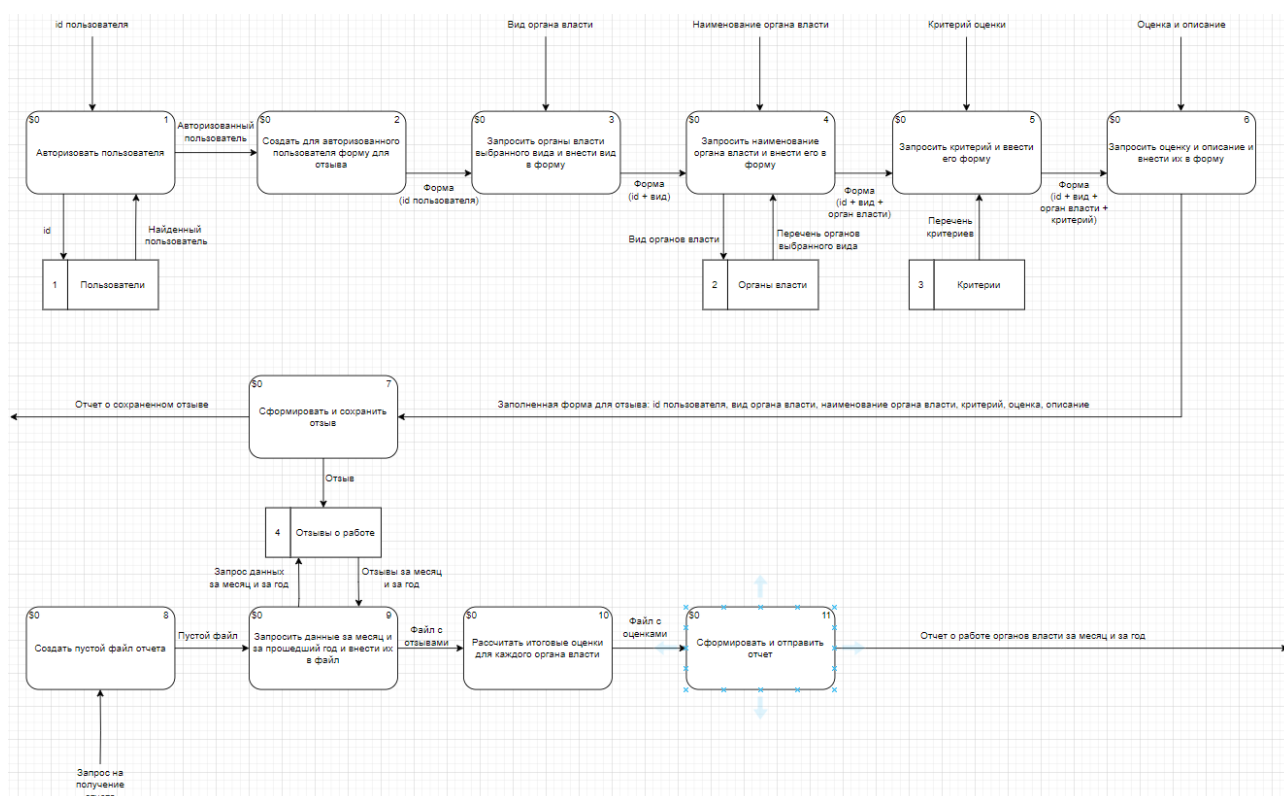


Рисунок 10 – Модель потоков данных

1.8 Обоснование выбора средств разработки

Для создания Telegram-бота и веб-приложения был выбран язык программирования PHP. Существует несколько преимуществ разработки Telegram-ботов на PHP по сравнению с другими языками программирования. Во-первых, PHP имеет длительную историю и широкую поддержку, что означает, что есть много инструментов, библиотек и фреймворков, которые могут быть использованы при разработке Telegram ботов. Во-вторых, PHP обеспечивает быстрый и легкий доступ к базам данных, что позволяет быстро сохранять и извлекать данные из бота. В-третьих, PHP имеет простой и понятный синтаксис, что упрощает разработку и поддержку кода. Кроме того, PHP предоставляет множество возможностей для работы с HTML и CSS, что упрощает разработку веб-приложений на данном языке.

Помимо сравнения с другими языками, выбор пал на PHP по причине активного использования данного языка в IT-инфраструктуре ЦУР РХ. Большинство ботов и веб-приложений, хранящихся на серверах ЦУР РХ, разработаны на данном языке, поэтому, чтобы не нарушать общую инфраструктуру и обеспечить удобную интеграцию проекта было решено выбрать этот язык.

Из аналогичных соображений была выбрана СУБД MySQL, поскольку она также является частью IT-инфраструктуры ЦУР РХ. На фоне других СУБД MySQL одна из самых легких систем управления базами данных, при этом является бесплатной и модифицируемой. Поддерживает стандарты SQL и ACID, а также является достаточно надежной и стабильной, что позволяет минимизировать риск потери или утечки данных.

Следующим этапом является выбор сред разработки. От выбора сред разработки зависит качество выпускаемого программного продукта и время, которое будет затрачено на его создание. Современная среда разработки способна сама ещё на стадии написания кода находить возможные ошибки и предлагать решения. Проведем сравнительную характеристику самых популярных инструментов

для работы с языком программирования PHP и системой управления базами данных MySQL.

Самыми популярными решениями для PHP являются PHPStorm, NetBeans и Sublime. Характеристики для сравнения выделить однозначно не получится, поскольку каждая среда обладает уникальными функциями, которых нет у других. Поэтому все достоинства и недостатки приведены в таблице 1.

Таблица 1 – Сравнение IDE для PHP

	PHPStorm	NetBeans	Sublime
Преимущества	<ul style="list-style-type: none"> – Анализ кода (дописывание). – Удобная работа с фреймворками. – Использование лучших практик для исправления кода. – Встроенный дебаггер. – Популярность среди разработчиков. 	<ul style="list-style-type: none"> – Анализ кода (дописывание). – Цена (бесплатный). – Встроенный контроль версий. – Множество функций для работы с вебom. 	<ul style="list-style-type: none"> – Скорость. – Цена (бесплатный). – Поддержка плагинов (можно переделать под себя). – Удобство работы с кодом. – Возможность изменить редактор под себя.
Недостатки	<ul style="list-style-type: none"> – Цена. – Скорость (слишком много лишнего функционала). 	<ul style="list-style-type: none"> – Скорость (медленный). – Плохая работа с большими файлами. 	<ul style="list-style-type: none"> – Требуется установка плагинов для становления полноценной IDE.

Исходя из сравнения видно, что все рассматриваемые инструменты обладают примерно равным числом преимуществ и недостатков.

Среды PHPStorm и NetBeans являются профессиональными средами для работы с PHP, поскольку имеют множество встроенных инструментов для работы с языком. Однако PHPStorm является платной программой и используется опытными разработчиками в больших компаниях, которые покрывают расходы на использование столь продвинутых инструментов. В данном случае столь точный инструмент излишен и дорог, поэтому PHPStorm не подходит.

Среды NetBeans и Sublime являются бесплатными, поэтому могут подойти для небольших проектов, однако NetBeans больше подходит для веб-разработчиков. В нем есть множество удобных инструментов для создания серверной части приложения, однако в случае с написанием Telegram-бота они будут только мешать. Из-за излишних инструментов скорость работы самой программы может пострадать. Также, пользователи отмечают, что NetBeans плохо работает с большими файлами, поэтому в случае с большими отчетами о рейтинге могут возникнуть проблемы и непредвиденные риски, чего хотелось бы избежать [5].

Таким образом, остается только Sublime – текстовый редактор с возможностью установить много плагинов для работы. Изначально Sublime поставляется как редактор и, следовательно, не так перегружен, что увеличивает скорость работы программы. С установкой плагинов на нем можно спокойно компилировать и запускать программы.

Для удобного администрирования базы данных необходимо выбрать приложение, предназначенное для работы с MySQL. Самыми популярными являются RНРMyAdmin, Navicat и Workbench. Сравнение этих приложений представлено в таблице 2.

Таблица 2 – Сравнение приложений для администрирования БД

	РНРMyAdmin	Navicat	Workbench
Преимущества	<ul style="list-style-type: none"> – Цена (бесплатный). – Управление БД без ввода запросов. – Удобная панель Администрирования. – Интенсивное развитие. 	<ul style="list-style-type: none"> – Визуальный конструктор запросов. – Миграция, синхронизация данных. – Инструмент для планирования задач. – Скорость. 	<ul style="list-style-type: none"> – Цена (бесплатный). – Графическое представление БД. – Удобное создание связей. – Возможность восстановить структуру БД.
Недостатки	<ul style="list-style-type: none"> – Скорость. – Меньше функционала, чем у других. 	<ul style="list-style-type: none"> – Цена. 	<ul style="list-style-type: none"> – Примитивный буфер обмена. – Нет конструктора запросов.

По результатам сравнения два из трех приложения бесплатные. Платный аналог отбросим, поскольку другие приложения обладают достаточным функционалом и нет смысла тратиться на дорогостоящий инструмент. На выбор остаются PHPMyAdmin и Workbench. Выбор пал на приложение PHPMyAdmin, поскольку оно обладает удобным графическим интерфейсом и позволяет работать с базой без ручного ввода запросов [6].

1.9 Выводы по разделу «Анализ предметной области и выбор средств разработки»

В данном разделе была проанализирована деятельность Центра управления регионом Республики Хакасия, его состав и задачи. Для выпускной квалификационной работы были поставлены цель и задачи для её достижения.

Перед началом проектирования были проанализированы уже имеющиеся программные продукты на рынке, подходящие для автоматизации процесса рейтингования. В ходе анализа выяснилось, что для внутреннего процесса необходимо разработать собственную информационную систему, поскольку предприятие работает с конфиденциальной информацией.

Для проектирования системы было решено построить 4 модели: модели «Как было» и «Как будет» в нотации IDEF0, модель действий пользователя в системе в нотации IDEF3 и модель потоков данных в нотации DFD. Данные модели позволяют по-разному взглянуть на систему, оценить объем работы до самой разработки и отследить возможные ошибки, которые могут возникнуть при работе в системе.

Перед разработкой было проведено сравнение популярных сред и средств для разработки информационных систем. По результатам сравнения было решено выбрать язык программирования PHP и среду разработки Sublime. Решающим фактором при выборе языка стало активное использование PHP в IT-инфраструктуре ЦУР РХ, что позволит быстро внедрить систему. В качестве среды разработки был выбран Sublime, поскольку он обладает достаточным

функционалом для реализации проекта и позволяет пользователю настроить среду так, как ему будет удобно.

2 Разработка информационной системы рейтингования

2.1 Проектирование базы данных

Разработку проекта стоит начать с проектирования базы данных. База данных будет использоваться в обеих частях проекта: бот будет использовать её для хранения оценок, списка пользователей, списка рейтингов и наименования органов власти; веб-приложение будет запрашивать данные о рейтинговании для построения графиков.

База данных будет содержать 4 сущности (таблицы): «Пользователи», «Таблица отзывов», «История рейтингования» и «Органы власти». Информационно-логическая модель базы данных изображена на рисунке 11.

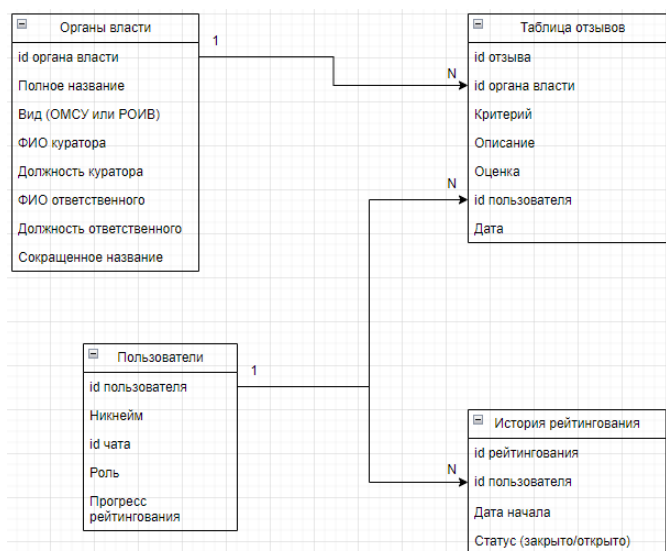


Рисунок 11 – Информационно-логическая модель базы данных

Таблица «Органы власти» предоставлена заказчиком, поскольку она уже используется в работе и содержит всю необходимую информацию об органах власти.

«Таблица отзывов» содержит информацию о всех оставленных отзывах. Отзывом является совокупность оценки и описания, внесенная сотрудником по выбранному органу власти и критерию. Данная таблица является результатом работы бота по сбору рейтинга и источником данных для отчета и веб-приложения.

Таблица «Пользователи» хранит информацию о пользователях бота. Никнейм и id чата в Telegram, роль (администратор или оценивающий), а также текущий прогресс рейтингования. Под прогрессом рейтингования подразумевается то, на каком этапе сейчас находится пользователь (выбирает орган власти, ставит оценку, вводит описание). Данное поле необходимо для корректной работы бота.

Таблица «История рейтингования» содержит информацию о каждом проведенном рейтинговании: его id, никнейм запустившего, дату начала и статус (открыто или закрыто) [7].

2.2 Разработка Telegram-бота для сбора рейтинга

2.2.1 Реализация процесса сбора рейтинга

Сбор рейтинга происходит поэтапно: сначала приходит сообщение о начале рейтингования. Далее пользователь выбирает сначала вид органа власти (ОМСУ или РОИВ), потом конкретный орган власти, критерий. Все предыдущие выборы осуществляются нажатием нужной кнопки в чате, далее необходимо отправить сообщение с оценкой в чат. Оценка может быть от -1 до 2 и должна быть целым числом. После оценки требуется ввести описание, после чего отзыв сохраняется и пользователю предлагается оставить ещё один отзыв.

Процесс запуска рейтингования начинается с ввода администратором команды «/oren». Код обработки команды представлен на рисунке 12.

```

$bot->command('open', function ($message) use ($bot) {
    $userId = $message->getChat()->getId();
    if (checkAuthorize($userId)) {
        $ratingId = time();
        if (createRating($ratingId, $userId, $bot)) {
            fwrite ($fp, 'id: '.$ratingId.' userId: '.$userId.PHP_EOL);
            massSend($ratingId, $bot);
        }
    } else {
        $bot->sendMessage($userId, "Вы не можете создавать опросы.");
    }
});

```

Рисунок 12 – Код обработки команды «/open»

В представленном коде бот получает id чата из введенного сообщения, проверяет может ли пользователь запускать рейтингование или нет. Если нет, то выдается сообщение, что пользователь не может создавать опросы. Иначе, в таблицу «История рейтингования» записывается никнейм запустившего и дата, а также запускается метод massSend(), который посылает всем пользователям сообщение о начале опроса. Код метода представлен на рисунке 13.

```

function massSend($ratingId, $bot) {
    $fp = fopen('error_012593.log', 'a');
    fwrite($fp, 'mass_send'.PHP_EOL);
    global $mysqli;
    $query = "UPDATE Mikh_rating_users SET mode = ''";
    $mysqli->query($query);
    $query = "SELECT * FROM Mikh_rating_users"; // WHERE type='0'; // WHERE type='0'
    $result = $mysqli->query($query);
    //fwrite($fp, 'result: '.serialize($result).PHP_EOL);
    foreach ($result as $key=>$val) {
        $GLOBALS['replyKeyboardRoivOmsu'] = new \TelegramBot\Api\Types\Inline\InlineKeyboardMarkup (
            [
                [
                    ['callback_data' => 'one_roiv', "text" => "РОИВ"],
                    ['callback_data' => 'one_omsu', "text" => "ОМСУ"]
                ],
                [
                    ['callback_data' => 'one_krasivoe', "text" => "Отчеты показываете?"]
                ]
            ],
            false,
            true
        );
        $document = new \CURLFile("Памятка.pdf");
        $bot->sendDocument($val['chatId'], $document, "Посмотрите в памятку.");
        $bot->sendMessage($val['chatId'], "Пожалуйста, оцените взаимодействие с РОИВ и ОМСУ.", false, null, null, $GLOBALS['replyKeyboardRoivOmsu']);
    }
    fclose($fp);
};

```

Рисунок 13 – Код метода massSend

В данном методе сначала происходит очистка текущего прогресса рейтингования всех пользователей в таблице, чтобы они могли начать оценивать с самого начала. Затем из базы данных запрашивается список всех пользователей,

кто будет оценивать (кто не администратор). Создается клавиатура с двумя кнопками: «ОМСУ» и «РОИВ». Данное сообщение вместе с памяткой отправляется каждому полученному из базы данных пользователю.

При создании клавиатуры каждой кнопке дается название, которое будет видно пользователю, а также `callback_data` (данные обратного вызова) – информация для программы, какая кнопка была нажата. В дальнейшем по этой информации можно в другом участке кода определить, какая кнопка была нажата и какое действие необходимо выполнить дальше. Пример обработки информации при нажатии кнопки «РОИВ» представлен на рисунке 14.

```
if ((strpos($callbackQuery->getData(), 'roiv') != false) && (!isset($data['step1']))) {
    fwrite($fp, 'step1: '.$data['step1'].PHP_EOL);
    $uData = array('step1' => 'РОИВ');
    setUdata($chatId, $uData);
    $query = "SELECT * FROM Mikh_organv1 WHERE idvid_ov='1'";
    $result = $mysqli->query($query)->fetch_all();
    $count = ceil(count($result)/2);
    $keyboardOne = array();
    $i = 0;
    for ($j=0; $j < $count; $j++) {
        $tmpArr = array();
        for ($k=0; $k<2; $k++) {
            $callback = 'two_organ('.$result[$i][0].')';
            if (($result[$i][11]) != '') {
                array_push($tmpArr, array('callback_data' => $callback, 'text' => $result[$i][11]));
            }
            $i++;
        }
        array_push($keyboardOne, $tmpArr);
        if ($f == 3) break;
    }
    array_push($keyboardOne, array(['callback_data' => 'btn_drop', "text" => "Отмена"]));
    $GLOBALS['replyKeyboardOrgan'] = new \TelegramBot\Api\Types\Inline\InlineKeyboardMarkup (
        $keyboardOne,
        false,
        true
    );
    $bot->sendMessage($chatId, 'Выберите РОИВ', false, null, null, $GLOBALS['replyKeyboardOrgan']);
    $bot->deleteMessage($chatId, $mId);
}
```

Рисунок 14 – Код обработчика для кнопки «РОИВ»

В представленном коде сначала проверяется то, что была нажата кнопка «РОИВ», и что пользователь находится на нужном этапе рейтингования (на случай, если он нажмет эту кнопку в другой момент времени). Информация о том, что пользователь выбрал «РОИВ» записывается в базу данных в поле «Прогресс рейтингования». Следующим шагом программы будет отображение всех органов власти данного вида (РОИВ). Для их получения в базу данных в таблицу «Органы

власти» отправляется запрос с получением всех РОИВ. Затем создается массив кнопок, в котором для каждого органа создается кнопка со своим названием и обратными данными (callback). Данный массив записывается как клавиатура в следующее сообщение и отправляется пользователю. Предыдущее сообщение удаляется, чтобы не перегружать историю сообщений в чате.

Аналогичным образом происходит дальнейший процесс рейтингования. Пользователь нажимает на кнопку, затем срабатывает обработчик, который записывает выбор в «Прогресс рейтингования» пользователя и отправляет сообщение со следующим выбором.

Так происходит до тех пор, пока от пользователя ожидается нажатие кнопки. Когда пользователь доходит до выставления оценки, то теперь не получится с помощью `callback_data` отследить то, какое действие выбрал пользователь. В данном случае помогает поле «Прогресс рейтингования», в котором записана вся история выборов пользователя до этого. Если пользователь прошел все этапы (выбрал вид органа власти, сам орган власти и указал критерий), то от него ожидается оценка. Если он поставил оценку, то дальше ожидается описание оценки. Поэтому в коде при получении сообщения от пользователя происходит сначала проверка, есть ли открытое рейтингование, затем проверка на то, находится ли пользователь сейчас на нужном этапе, а потом проверка корректности введенных данных. Под корректностью подразумевается, например, правильно введенная оценка. Она должна быть целым числом от -1 до 2 включительно. Иначе пользователю нужно сообщить, что он ввел число неправильно. Код обработчика сообщений представлен на рисунке 15.

```

$bot->on(function ($update) use ($bot) {
    $fp = fopen('error_012593.log', 'a');
    fwrite($fp, 'update'.PHP_EOL);
    $message = $update->getMessage();
    $chatId = $message->getChat()->getId();
    $postText=$message->getText();
    $userName = $message->getChat()->getUsername();
    $data = getUdata($chatId);
    if (checkUser($chatId)) {
        if ((isset($data['step3']) && (isset($data['step4']))) {
            fwrite($fp, 'step4 start'.PHP_EOL);
            $uData = array('step4' => $postText);
            $data = array_merge($uData, $data);
            if ((is_numeric($postText) && ($postText >= -1) && ($postText <= 2)) {
                setUdata($chatId, $data);
                $bot->sendMessage($chatId, 'Пожалуйста, добавьте описание к поставленной отметке', false, null, null, $GLOBALS['dropRating']);
            } else {
                $bot->sendMessage($chatId, 'Оценка не соответствует условию.');
            }
        }
        elseif ((isset($data['step4']) && (isset($data['step5']))) {
            fwrite($fp, 'step5'.PHP_EOL);
            $uData = array('step5' => $postText);
            $data = array_merge($uData, $data);
            setUdata($chatId, $data);
            $bot->sendMessage($chatId, 'Ваш отзыв успешно сохранен!');
            $bot->sendMessage($chatId, 'Вы можете оставить ещё один, нажав на кнопку "Ещё"', false, null, null, $GLOBALS['replyKeyboardAgain']);
            addRecord($chatId, $userName);
        } else {
            $bot->sendMessage($chatId, 'Команда не верна.');
        }
    }
}, function() {
    return true;
});

```

Рисунок 15 – Код обработчика сообщений от пользователя

Таким образом проходит процесс выставления оценки деятельности органов власти. После ввода описания оценки пользователю предлагается оставить ещё один отзыв, нажав кнопку «Ещё». Нажатие кнопки обновляет прогресс пользователя и отправляет сообщение с просьбой выбрать «ОМСУ» или «РОИВ».

Чтобы завершить текущий опрос администратор должен ввести команду «/close», которая изменит статус текущего опроса в базе данных, и при любом действии пользователя ему будет выдаваться сообщение о том, что на данный момент открытых опросов нет.

2.2.2 Формирование отчета в Excel

Для формирования отчета используется библиотека PhpOffice. С её помощью можно создавать объекты файлов из Microsoft Office и редактировать их [8].

Требуемый отчет является файлом Excel и состоит из трех других отчетов (листов).

Первый лист – это отчет за текущий месяц. В качестве столбцов выступают критерии оценивания, а в качестве строк – наименования органов власти. На пересечении находится сумма поставленных оценок и комментарии к деятельности органа власти по данному критерию. Также в отчете содержится отдельный столбец с суммарной оценкой для каждого органа власти. Эта оценка складывается из всех оценок, полученных по всем критериям.

Пример сгенерированного отчета за апрель можно посмотреть на рисунке 16.

A	B	C	D	E	F	G	H	I		
ТИП	Наименование	Месяц	Итоговый результат	Госпбюджет (РОИВ ОМСУ и подведы)	Среднее время ответов в ИМ	Инфа о решённых проблемах в соц	Редакция_сбор и подготовка	Продвижение ЦУ		
РОИВ	Министерство строительства и жилищно-коммунального хозяйства Республики Хакасия	Апр	3	1	2	(2)Отлично! cracaziabra 2023-04-02 16:38:16				
РОИВ	Министерств образования и науки Республики Хакасия	Апр	-1	-1						
РОИВ	Министерство природных ресурсов и экологии Республики Хакасия	Апр	0							
РОИВ	Министерство труда и занятости населения Республики Хакасия	Апр	0							

Рисунок 16 – Пример сгенерированного отчета за апрель

Для создания такого отчета с помощью кода необходимо «пробежаться» по каждой ячейке и заполнить её нужным значением. Пробег осуществляется с помощью циклов. Первым шагом является выставление названий столбцов. Их фиксированное количество, поэтому они проставляются прямо в коде. Затем, создается цикл с количеством итераций, равным количеству органов власти. Список органов власти запрашивается из базы данных. Для каждого органа власти заполняется строка. Из базы данных запрашивается сумма оценок, отзывы и критерии по текущему органу власти за месяц. Далее, с помощью конструкции switch-case по критерию выбирается сумма оценок и формируется список отзывов к оценкам. Код получился достаточно объемным из-за конструкции switch-case и процесса заполнения каждой ячейки. Часть кода представлена на рисунке 17.

```

$sheet->setCellValue('B1', 'Наименование');
$sheet->setCellValue('C1', 'Месяц');
$sheet->setCellValue('D1', 'Итоговый результат');
$sheet->setCellValue('E1', 'Госпблизки (РОИВ ОМСУ и подведы)');
$sheet->setCellValue('F1', 'Среднее время ответов в ИМ');
$sheet->setCellValue('G1', 'Инфа о решённых проблемах в соц сетях');
$sheet->setCellValue('H1', 'Редакция сбор и подготовка информации');
$sheet->setCellValue('I1', 'Продвижение информации ЦУРа в соц сетях');
$sheet->setCellValue('J1', 'Зелёная зона в кв.рейтинге и обратная связь');
$sheet->setCellValue('K1', 'Деятельность по формированию ДК');
$sheet->setCellValue('L1', 'Угрозы и риски');
$sheet->setCellValue('M1', 'Таргет и онлайн-социология');
$sheet->setCellValue('N1', 'Прочие направления');
global $mysqli;
$query = "SELECT idov,nazv,idvid_ov FROM Mikh_organv1 WHERE idvid_ov <= '2'";
$result = $mysqli->query($query);
$count = $result->num_rows;
$criteriaArray = array();
$i=2;
for ($startDate = $start; $startDate <= $end; $startDate++){
    $curDate = sprintf("%02d", $startDate);
    fwrite($fp, $curDate.PHP_EOL);
    $record = array();
    $date = date("Y");
    $from = $date."-".$curDate-01." 00:00:00";
    $to = $date."-".$curDate-32." 23:59:59";

    // $bot->sendMessage($chatId, $count);
    if ($count>0) {
        $result->data_seek(0);
        foreach ($result as $key=>$value) {
            // $bot->sendMessage($chatId, $value['nazv']);
            $queryOrgan = "select a.id, a.company, a.criteria, a.description, a.userId, a.userName, a.score, sum(b.score), a.date
            from Mikh_rating_table a
            left join Mikh_rating_table b
            on b.company = '".$value['nazv']."' and a.criteria = b.criteria
            where a.company = '".$value['nazv']."' and a.`date` >='".$from."' and a.`date` <='".$to."'
            and b.`date` >='".$from."' and b.`date` <='".$to."'
            group by a.id;";

            $resultOrgan = $mysqli->query($queryOrgan);
            foreach ($resultOrgan as $orgKey=>$orgVal) {
                switch ($orgVal['criteria']) {
                    case 'Госпблизки (РОИВ ОМСУ и подведы)':
                        $record[$i][0]['sum'] = $orgVal['sum(b.score)'];
                        $record[$i][0][10]['description'] = $record[$i][0][10]['description'].chr(13).chr(10).('.'.$orgVal['score'].').'.$orgVal['description'].'.$orgVal['userName']
                        .'.'.$orgVal['date'];
                }
            }
        }
    }
}

```

Рисунок 17 – Часть кода метода по созданию отчета за месяц

Второй лист итогового отчета содержит отчет о деятельности органов власти за текущий год. По структуре он аналогичен отчету за месяц: критерии расположены по столбцам, а органы власти – по строкам. Отличие в том, что теперь количество строк определяется не только количеством органов власти, но и количеством прошедших за этот год месяцев. Таким образом, названия органов власти могут повторяться в виде строк, однако месяц будет указан иной. Переход по месяцам внутри отчета показан на рисунке 18.

32	ОМСУ	Администрация Усть-Абаканского района	Jan	0	
33	ОМСУ	Администрация Ширинского района	Jan	0	
34	РОИВ	Правительство РХ	Jan	0	
35	РОИВ	Министерство строительства и жилищно-коммунального хозяйства Республики Хакасия	Feb	0	
36	РОИВ	Министерств образования и науки Республики Хакасия	Feb	0	
37	РОИВ	Министерство природных ресурсов и экологии Республики Хакасия	Feb	0	

Рисунок 18 – Переход по месяцам внутри отчета за год

Поскольку годовой отчет похож на отчет за месяц, то с программной стороны будет проще воспользоваться уже готовым кодом по формированию отчета за месяц, но ввести возможность выбора даты. Для этого циклы пробегов по ячейкам необходимо поместить в ещё один цикл – пробег по месяцам. По каждому месяцу будет заполняться свой отчет за месяц, однако каждый следующий такой отчет будет начинаться сразу после предыдущего на следующей строке. Достаточно просто не обнулять номер строки, на которой сейчас находится запись, и начинать с неё формировать следующий отчет за месяц.

Таким образом, второй лист будет формироваться тем же методом, что и первый, но будет выставлена дата начала – 1 месяц года. Отчет за месяц в аргументах метода будет содержать две одинаковые даты – текущий месяц. Код этого метода был представлен на рисунке 17.

Отчет для печати располагается на третьем листе. Он отличается от двух предыдущих, поскольку в нем не учитываются критерии оценивания, отдельные оценки и комментарии. На этом отчете органы власти поделены на две группы (ОМСУ и РОИВ) и в качестве столбцов выступают прошедшие месяца текущего года. Цель отчета – отобразить итоговый рейтинг органов власти за каждый из прошедших месяцев. Также ячейки перекрашиваются в цвета в зависимости от оценки внутри: если меньше 0, то в красный; если от 0 до 2 включительно, то в желтый; иначе – в зеленый. Пример отчета изображен на рисунке 19.

Наименование	Jan	Feb	Mar	Apr	May	Наименование	Jan	Feb	Mar	Apr	May
Министерство строительства и жилищно-коммунального хозяйства Республики Хакасия	0	0	0	2	0	Администрация города Абакана	0	0	0	0	2
Министерство образования и науки Республики Хакасия	0	0	0	-1	0	Администрация города Абаза	0	0	0	0	0
Министерство природных ресурсов и экологии Республики Хакасия	0	0	0	0	0	Администрация города Саяногорска	0	0	0	0	0
Министерство труда и социальной защиты Республики Хакасия	0	0	0	0	0	Администрация города Сорска	2	0	1	-1	1
Министерство здравоохранения Республики Хакасия	0	0	0	2	1	Администрация города Черногорска	2	1	1	1	3
Министерство транспорта и дорожного хозяйства Республики Хакасия	0	0	0	0	0	Администрация Алтайского района	0	0	0	0	0
Министерство финансов Республики Хакасия	0	0	0	0	0	Администрация Аскинского района	0	0	0	0	0
Министерство экономического развития Республики Хакасия	0	0	0	0	0	Администрация Бейского района	-1	0	5	2	1
Министерство сельского хозяйства Республики Хакасия	0	0	0	0	0						

Рисунок 19 – Пример отчета для печати

Отчет сильно отличается от двух предыдущих, поэтому для его формирования был выделен отдельный метод. В методе из базы данных запрашиваются органы власти по видам (ОМСУ и РОИВ) и помещаются в два массива. По этим массивам в дальнейшем будут заполняться строки. Сначала формируется таблица для РОИВ. В самой первой строке прописываются месяцы, а в самом первом столбце – наименования органов власти. Затем для каждого месяца из базы данных запрашивается орган власти и сумма оценок, которые затем вводятся в соответствующую ячейку. С помощью библиотеки на основании оценки ячейке устанавливается цвет. Таблица для ОМСУ формируется аналогичным образом, только начинается формирование с отступом в один столбец от предыдущей таблицы. Весь код состоит из более чем 900 строк, на рисунке 20 представлена только часть кода.

```
function createExcelToPrint($bot, $sheet, $end) {
    $fp = fopen('error_012593.log', 'a');
    $sheet -> getStyle('A') -> getAlignment() -> setWrapText(true);
    fwrite($fp, 'StartFormingPrintExcel'.PHP_EOL);
    $startCol = 'B';
    $i = 2;
    $monthNum = 1;
    $sheet -> setCellValue('A1', 'Наименование');
    global $mysqli;
    $query = "SELECT nazv, idvid_ov FROM Mikh_organv1 WHERE idvid_ov = '2'";
    $omsu = $mysqli -> query($query);
    $omsuCount = $omsu -> num_rows;
    $query = "SELECT nazv, idvid_ov FROM Mikh_organv1 WHERE idvid_ov = '1'";
    $roiv = $mysqli -> query($query);
    $roivCount = $roiv -> num_rows;
    foreach ($roiv as $key => $value) {
        $sheet->setCellValue('A' . $i, $value['nazv']);
        $i++;
    }
    for ($col = $startCol; $monthNum <= $end; $col++) {
        $curDate = sprintf("%02d", $monthNum);
        $date = date("Y");
        $from = $date . "-" . $curDate . "01 00:00:00";
        $to = $date . "-" . $curDate . "31 23:59:59";
        $query = "select company, sum(score) from Mikh_rating_table where date >='". $from ."' and date <='". $to ."' and company_type = 'РОИВ' group by company";
        $result = $mysqli -> query($query);
        $sheet -> setCellValue($col . '1', date('M', mktime(0,0,0, $monthNum, 1, 2020)));
        $result = $mysqli -> query($query) -> fetch_all();
        // fwrite($fp, implode($scores).PHP_EOL);
        for ($row = 2; $row <= $roivCount+1; $row++) {
            $score = 0;
            $compName = $sheet -> getCell("A" . $row) -> getValue();
            foreach ($result as $key => $cts) {
                if ($cts[0] == $compName) {
                    $score = $cts[1];
                    break;
                }
            }
            $color = '00ffffcc';
            if ($score >= 5) $color = '0092d050';
            if ($score < 0) $color = '00ff0000';
            $sheet -> setCellValue($col . $row, $score);
            $sheet -> getStyle($col . $row) -> getFill() -> setFillType(\PhpOffice\PhpSpreadsheet\Style\Fill::FILL_SOLID) -> getStartColor() -> setARGB($color);
        }
        $monthNum++;
    }
}
```

Рисунок 20 – Часть кода по созданию отчета для печати

В результате пользователю в виде сообщения в Telegram возвращается итоговый отчет в виде Excel-файла, содержащий 3 листа. Как это выглядит в самом чате можно увидеть на рисунке 21.

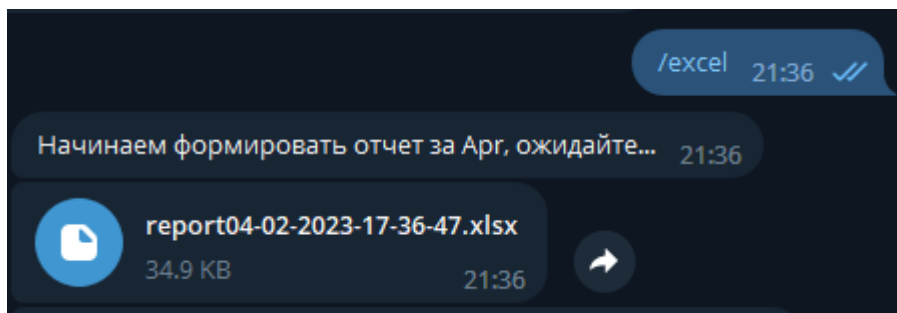


Рисунок 21 – Сообщение с файлом в ответ на команду /excel

2.3 Разработка веб-приложения для отображения графиков и таблиц

2.3.1 Определение структуры веб-приложения

Поскольку веб-приложение будет содержать несколько элементов (график, таблицу, кнопки переключения, названия), то необходимо заранее определить, как эти элементы будут располагаться на экране.

За основу для сайта был взят шаблон от заказчика. Поскольку ЦУР РХ давно занимается разработкой различных систем для работы с органами власти, то у них уже есть сайт, где все эти системы можно разместить и распределить по разделам. В один из таких разделов в дальнейшем будет внедрена разрабатываемая система, а именно веб-приложение (бот размещается в Telegram). Шаблон включает в себя заголовок, логотип и боковое меню с другими разделами (рисунок 22).

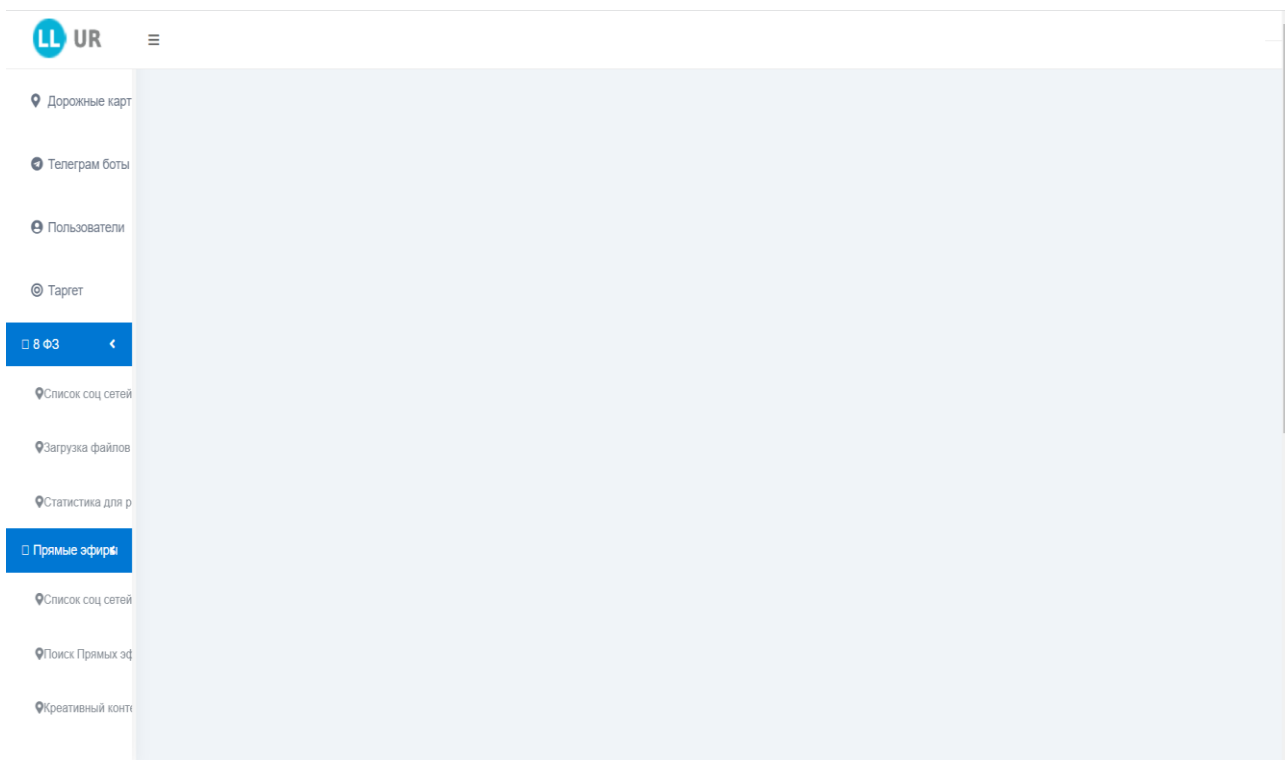


Рисунок 22 – Шаблон сайта от заказчика

Для расстановки элементов введем табличную систему классов. Вдоль горизонтали будут располагаться строки (блоки, обозначенные классом row). В каждой строке будет один или больше столбцов (класс col). Таким образом можно расположить несколько элементов в строку и несколько строк на страницу. В первой строке будет располагаться название графика, во второй – график и кнопки, в третьей – название таблицы, а в четвертой – сама таблица (рисунок 23).


```

<!-- Блок с графиком -->
<div class="row">
  <div class="col-xl-3 col-lg-3 col-md-3 col-sm-6 m-b-3">
    <div class="card">
      <div class="card-body">
        <div id="line"></div>
      </div>
    </div>
  </div>
<!-- Кнопки -->
<div style="max-width: 25%;">
  <form action="" method="POST" id="orb_form">
    <fieldset >
    </fieldset>
  </form>
</div>
</div>

<!-- Название таблицы и таблица -->

<div class="row">
  <div class="col-xl-3 col-lg-3 col-md-3 col-sm-6 m-b-3">
    <p style="margin-bottom: 2px; font-size: 1.5rem;">Таб
  </div>
</div>

<div class="row">
  <div class="col-xl-3 col-lg-3 col-md-3 col-sm-6 m-b-3">
    <div class="card">
      <div class="card-body">
        <table class="table table-hover table-bordered">

```

Рисунок 23 – Код формирования структуры веб-приложения

2.3.2 Создание графиков с помощью Morris.js

Morris.js – это библиотека для быстрого создания графиков с помощью JavaScript. Создание графиков происходит путем передачи данных и настроек внешнего вида графика в специальный конструктор библиотеки. Для работы Morris.js также необходимо подключить две других библиотеки: jQuery и Raphael [9].

Для отображения графиков в Morris.js необходимо передать данные. Эти данные запрашиваются из той же базы данных, с которой работает Telegram-бот. Доступ к базе данных осуществляется через phpMyAdmin, поэтому для получения данных будет использоваться код на языке PHP, который будет формировать

строку с данными в нужном для библиотеки формате, а затем передавать её в конструктор. Код для получения данных из базы и формирования строки для библиотеки представлен на рисунке 24.

```
$query = "select b.socrnazv from Mikh_rating_table_old a left join Mikh_organvl b on a.company = b.nazv
b.socrnazv";
$result = $mysqli -> query($query);
$companies = "";

while ($row = mysqli_fetch_array($result)) {
    $socrnazv = $row["socrnazv"];
    if ($socrnazv == null) continue;
    $socrnazv = trim($socrnazv);
    $socrnazv = str_replace(' ', '_', $socrnazv);
    $companies .= "'".$socrnazv."'";
}
$companies = substr($companies, 0, -2);

$chart_data = '';
for ($month = 1; $month <= date('n'); $month++) {
    $curDate = sprintf("%02d", $month);
    $query = "select b.socrnazv, sum(a.score) from Mikh_rating_table_old a left join Mikh_organvl b on b.n
    <= '2023-".$curDate."-32 23:59:59' group by a.company";
    $result = $mysqli -> query($query);
    $date = date("Y").'-' . $curDate;
    $chart_data .= "{ date: '" . $date . "', ";
    while ($row = mysqli_fetch_array($result)) {
        $socrnazv = $row["socrnazv"];
        if ($socrnazv == null) continue;
        $socrnazv = trim($socrnazv);
        $socrnazv = str_replace(' ', '_', $socrnazv);
        $chart_data .= $socrnazv . " : ".$row["sum(a.score)].", ";
    }
    $chart_data = substr($chart_data, 0, -2);
    $chart_data .= "}, ";
}
$chart_data = substr($chart_data, 0, -2);
?>
```

Рисунок 24 – Код для запросов к базе данных и формирования строки

Полученную строку с данными необходимо передать в конструктор библиотеки. Строка содержит в себе данные в виде «Ключ: значение». По ключам можно настроить какие данные будут отображаться на каждой оси, а значения библиотека сама возьмет из данной строки по этим же ключам. Также в конструкторе можно настроить поведение графика: будет ли он переходить от значения к значению плавно, какая будет толщина линии, какие будут цвета и т.д. Итоговый конструктор для графика представлен на рисунке 25.

```

<script>

Morris.Line({
  element: 'line',
  locale: 'ru',
  data: [
    <?php echo $chart_data; ?>],
  xkey: 'date',
  ykeys: [<?php echo $companies; ?>],
  labels: [<?php echo $companies; ?>],
  fillOpacity: 0.9,
  behaveLikeLine: true,
  lineWidth: 3,
  smooth: true,
  hideHover: 'auto',
  parseTime: false,
  resize: true
});

</script>

```

Рисунок 25 – Код конструктора для графика

Код для формирования графика готов, однако в процессе написания и тестирования проекта было выяснено, что органы власти стоит разделить по видам на ОМСУ и РОИВ и отображать отдельно. Поскольку суммарно органов власти 33 (20 РОИВ и 13 ОМСУ), то на одном графике они будут перемешиваться друг с другом и график получится ненаглядным и запутанным. Поэтому, обговорив с заказчиком, было принято решение разделить график на два, отдельно для каждого вида: РОИВ и ОМСУ. Для переключения между графиками были добавлены две кнопки с названиями видов органов власти. С программной стороны кнопки посылают POST запрос по нажатию с названием кнопки. Страницка перезагружается, это же приложение получает этот запрос и устанавливает выбранный вид органов власти. Затем он подставляется в запрос к базе данных и по итогу запрашиваются только нужные данные. На рисунках 26 и 27 представлены графики для ОМСУ и РОИВ соответственно.

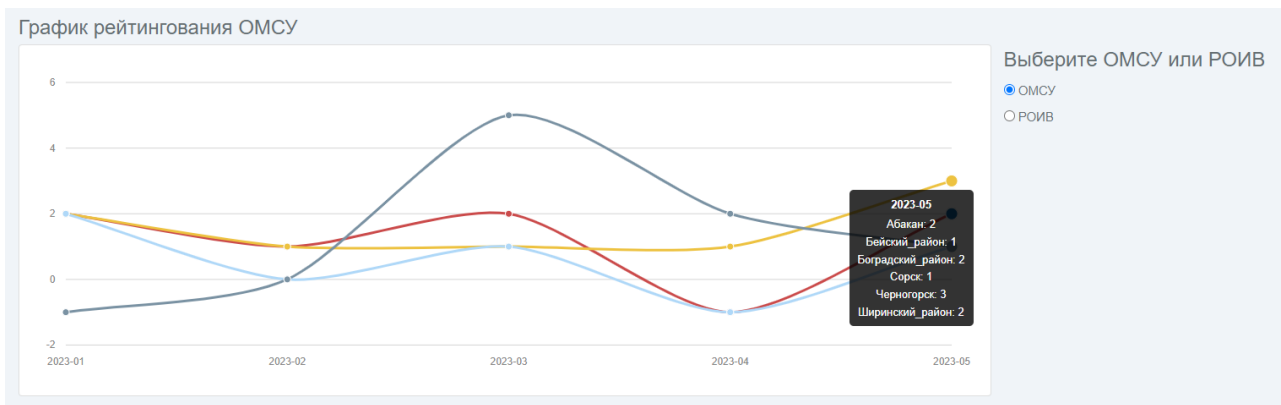


Рисунок 26 – График для органов власти вида ОМСУ

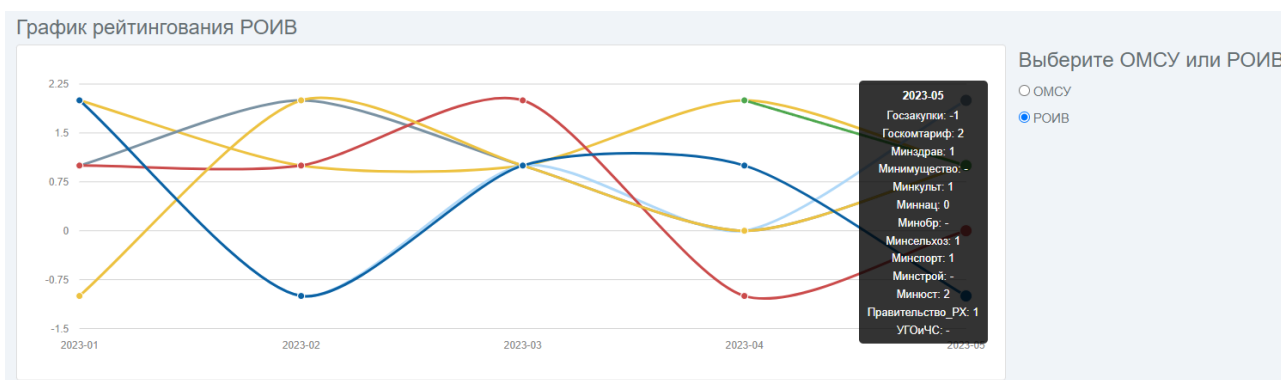


Рисунок 27 – График для органов вида РОИВ

2.3.3 Создание таблиц с помощью РНР

Для создания таблиц было решено не использовать сторонних библиотек, поскольку сами таблицы изначально поддерживаются в HTML и с помощью РНР можно формировать HTML-код с нужными данными [10].

Однако просто ввести готовую форму HTML, а потом просто заполнить её данными из РНР тоже не выйдет, поскольку количество строк и столбцов в таблице не фиксировано. Подобно графику, таблица также будет отображать данные только для выбранного вида органов власти (ОМСУ или РОИВ), от этого будет меняться количество строк. В качестве столбцов будут выступать прошедшие месяцы текущего года, поэтому их количество также будет изменяться. Таким образом, необходимо написать РНР код, который будет формировать нужное количество строк и столбцов и заполнять их данными из базы.

Помимо этого, в отличие от графиков, в таблице должны быть представлены не только суммарные оценки органа власти за месяц, а также средние значения за весь прошедший год и номера критериев, по которым были выставлены оценки за каждый месяц.

Исходя из этих условий, запрос в базу данных для таблицы будет немного отличаться от похожего запроса для бота. Значение каждой ячейки можно представить как запрос в базу по конкретному органу власти, за конкретный период времени, с суммой оценок и всеми критериями. Помимо простого запроса необходимо в цикле пройти по всей таблице. Внешним циклом будет проход по органам власти, а внутренним – по месяцам. Код внутреннего цикла с запросом представлен на рисунке 28.

```
for ($i = 1; $i <= date('n'); $i++) {
    $curDate = sprintf("%02d", $i);
    $date = date("Y");
    $from = $date."-$curDate-01 00:00:00";
    $to = $date."-$curDate-31 23:59:59";
    $query = "select sum(score), GROUP_CONCAT(DISTINCT criteria) as crit from Mikh_rating_table
        where company = '". $comp.'" and date >='". $from.'" and date <='". $to.'" group by company";

    $result = $mysqli -> query($query);
    $row = mysqli_fetch_array($result);
    $monthScore = isset($row[0]) ? $row[0] : 0;
    $crits = explode(',', $row[1]);
    $numbers = '';
    if (isset($row[1])) {
        $numbers .= '№';
        foreach ($critlist as $key => $value) {
            if (in_array($key, $crits)) $numbers .= $value.", ";
        }
        $numbers = substr($numbers, 0, -2);
        $numbers .= ' ';
    }
    $averageScore+=$monthScore;
    $color = "Khaki";
    if ($monthScore < 1) $color = "LightCoral";
    if ($monthScore > 2) $color = "LimeGreen";
    echo "<td style =\"background-color: ".$color.";\>". $monthScore." ".$numbers."</td>";
}
```

Рисунок 28 – Код внутреннего цикла заполнения таблицы

Внешний цикл проходит по всем органам власти выбранного вида (рисунок 29).

```

foreach ($companies as $key => $value) {

    $comp = $value['nazv'];
    $averageScore = 0;

    echo "<tr><th scope=\"row\">".$rowCount."</th>";
    echo "<th scope=\"row\">".$comp."</th>";
}

```

Рисунок 29 – Код внешнего цикла заполнения таблицы

Формирование таблицы происходит с помощью тегов: <tr> для строки, <th> для столбца. Помимо числовых значений, ячейки таблицы также раскрашены в три цвета в зависимости от суммарной оценки за месяц: красный – оценка меньше 0, желтый – оценка от 0 до 2 включительно, зеленый – оценка больше 2. Таблица для ОМСУ представлена на рисунке 30.

Таблица рейтингования ОМСУ

#	Орган исполнительной власти	Jan	Feb	Mar	Apr	May	Jun	Среднее значение за период
1	Администрация города Абакана	0	0	0	0	2 (№5)	0	1
2	Администрация города Абаза	0	0	0	0	0	0	0
3	Администрация города Саяногорска	0	0	0	0	0	0	0
4	Администрация города Сорска	2 (№4)	0 (№2)	1 (№7)	-1 (№2)	1 (№6)	0	1
5	Администрация города Черногорска	2 (№2)	1 (№7)	1 (№7)	1 (№3)	3 (№3, 5)	0	2
6	Администрация Алтайского района	0	0	0	0	0	0	0
7	Администрация Аскизского района	0	0	0	0	0	0	0
8	Администрация Бейского района	-1 (№6)	0 (№5)	3 (№3, 5)	2 (№6)	1 (№5)	0	2
9	Администрация Боградского района	0	0	0	0	2 (№5)	0	1
10	Администрация Орджоникидзевского района	0	0	0	0	0	0	0
11	Администрация Таштыпского района	0	0	0	0	0	0	0
12	Администрация Усть-Абаканского района	0	0	0	0	0	0	0
13	Администрация Ширинского района	2 (№6)	1 (№4)	2 (№3)	-1 (№6)	2 (№7)	0	1

Рисунок 30 – Таблица рейтингования для ОМСУ

Итоговый внешний вид веб-приложения показан на рисунке 31.

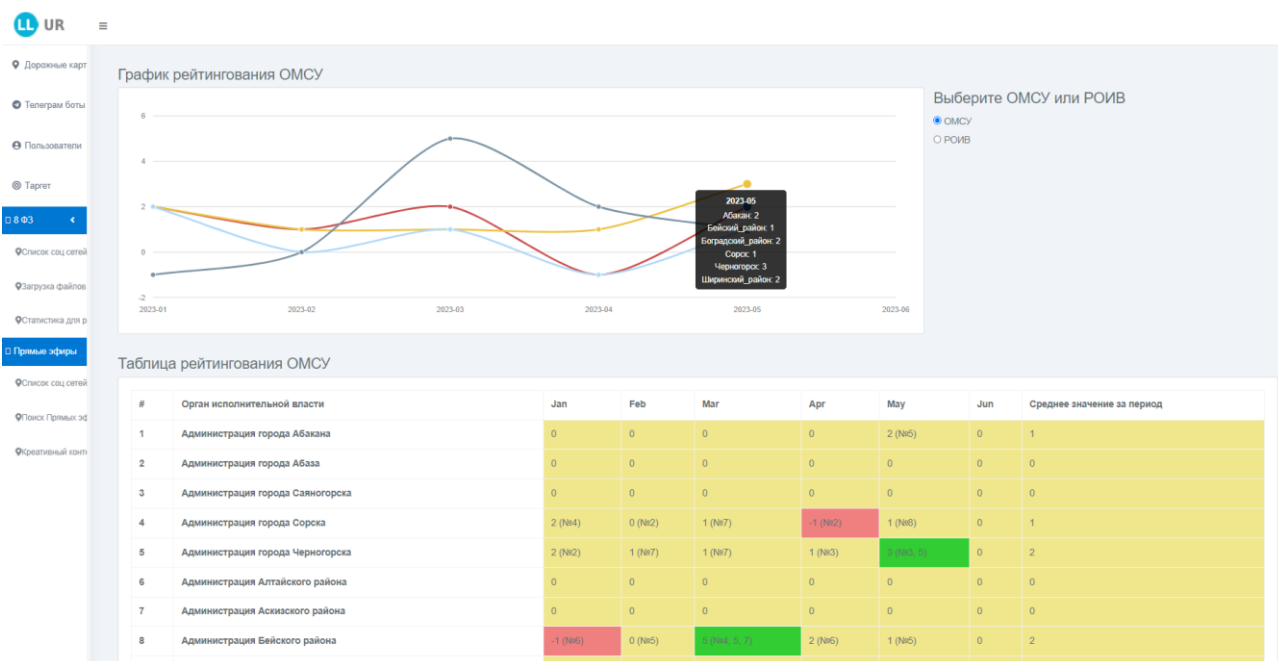


Рисунок 31 – Итоговый внешний вид веб-приложения по рейтингованию

2.4 Выводы по разделу «Разработка информационной системы рейтингования»

В результате процесса разработки была спроектирована и создана база данных, состоящая из 4 таблиц. С созданной базой данных взаимодействуют две части системы: Telegram-бот и веб-приложение.

Telegram-бот был написан на языке программирования PHP с использованием Telegram API – набором инструментов от разработчиков Telegram для работы с чат-ботами. Функционала Telegram API оказалось достаточно, чтобы реализовать процесс сбора оценок, однако для формирования отчёта понадобилась дополнительная библиотека – PhpOffice. Данная библиотека позволяет с помощью PHP создавать файлы из Microsoft Office (Word, Excel и т.д.). Таким образом полученные данные получилось сформировать в отчёт в виде Excel-файла.

Для создания веб-приложения также использовался язык PHP, с помощью которого осуществлялось получение данных из базы и расчёт итоговых оценок. Полученные данные были представлены в виде графика с помощью библиотеки Morris.js и в виде таблицы с помощью формирования html-элементов через PHP.

3 Определение совокупной стоимости владения проектом

3.1 Оценка затрат на средства разработки

Для реализации проекта потребуется персональный компьютер (ПК), а также программное обеспечение для разработки. Время на разработку проекта – 2 месяца.

Для разработки необходимо выделить ПК. Поскольку разработка не включает в себя сложных графических элементов, то необходимости в мощной видеокарте нет. Комплектующие для компьютера представлены в таблице 3.

Таблица 3 – Стоимость комплектующих для разработки

Комплектующие	Название	Основные характеристики	Стоимость, руб.	Срок, мес.
Процессор	AMD Ryzen 7 5700G	4 ядра, встроенное видеоядро Vega 6	14000	2
Материнская плата	GIGABYTE B450M H	Сокет AM4	4200	2
Оперативная память	Crucial Ballistix	Объем: 8ГБ DDR4	3600	2
Монитор	Lenovo ThinkVision	Диагональ: 21.5 дюйма, TN матрица	8200	2

Проект будет разрабатываться на языке программирования PHP, в качестве системы управления базами данных будет использован MySQL. На компьютере будет установлена операционная система Windows 10 Home. Названия и стоимость программ отображены в таблице 4.

Таблица 4 – Программное обеспечение для разработки

ПО	Название	Стоимость, руб.	Сроки, мес.
Операционная система	Windows 10 Home	17699	2
Среда для разработки на PHP	Sublime Text	бесплатно	2
Приложение для работы с MySQL	PHPMyAdmin	бесплатно	2

3.2 Расчет капитальных затрат

Расчет капитальных затрат проводится по следующей формуле

$$K = K_{\text{пр}} + K_{\text{тс}} + K_{\text{лс}} + K_{\text{по}} + K_{\text{ио}} + K_{\text{об}} + K_{\text{оэ}}, \quad (1)$$

где $K_{\text{пр}}$ – затраты на проектирование информационной системы;

$K_{\text{тс}}$ – затраты на технические средства управления;

$K_{\text{лс}}$ – затраты на создание линий связи локальных сетей;

$K_{\text{по}}$ – затраты на программные средства;

$K_{\text{ио}}$ – затраты на формирование информационной базы;

$K_{\text{об}}$ – затраты на обучение персонала;

$K_{\text{оэ}}$ – затраты на опытную эксплуатацию.

Расчет стоит начать с затрат на проектирование системы (проектные затраты), поскольку данные затраты включают в себя ряд других. Формула для расчета проектных затрат

$$K_{\text{пр}} = K_{\text{зп}} + K_{\text{свт}} + K_{\text{ипс}} + K_{\text{проч}}, \quad (2)$$

где $K_{\text{зп}}$ – затраты на заработную плату разработчикам;

$K_{\text{свт}}$ – затраты на средства вычислительной техники;

$K_{\text{инс}}$ – затраты на инструментальные программные средства;

$K_{\text{проч}}$ – прочие проектные затраты.

Разрабатываемый проект состоит из двух частей: Telegram-бота и веб-приложения. Обе части разрабатываются на одном и том же языке PHP, поэтому одного программиста, владеющего данным языком, хватит для разработки системы. Минимальный размер оплаты труда (МРОТ) по стране составляет 16242 рубля, ниже этой суммы платить разработчику нельзя. Суммарно разработка всего проекта длится 2 месяца, по 4 часа в день. То есть на разработку в общей сложности будет затрачен 1 полный рабочий месяц. Таким образом нагрузка будет распределена равномерно и платить программисту больше МРОТ нет необходимости.

Таким образом, систему будет разрабатывать один программист с окладом 16242 рубля.

Для расчета зарплаты программиста стоит учитывать районный и северный коэффициенты. В республике Хакасия оба коэффициента равны 30% от оклада. Суммарная надбавка составит $30 + 30 = 60\%$.

Зарплата программиста с учетом коэффициентов составит $16242 * 1,6 = 25987$ рублей в месяц. Разработка продлится 2 месяца по 4 часа в день.

Работодателю также необходимо учесть обязательные отчисления во внебюджетные фонды (30,2% от зарплаты).

$$K_{\text{зп}} = 25987 * 1,302 = 33835 \text{ руб.}$$

Стоимость компьютера для разработки по данным из таблицы 3 составляет $14000 + 4200 + 3600 + 8200 = 30000$ рублей. Норма амортизационных отчислений в год составляет 25%. Поскольку разработка ведется 2 месяца, то процент за это время составит $25 / 12 * 2 = 4,16\%$. При расчете затрат на средства вычислительной техники учитывается только время работы техники, то есть амортизация.

$$K_{\text{свт}} = 30000 * 0,0416 = 1248 \text{ руб.}$$

Затраты на инструментальные программные средства состоят только из затрат на амортизацию операционной системы, поскольку остальное программное обеспечение бесплатное (данные из таблицы 4). Предполагается, что операционную систему необходимо будет обновить через 5 лет, следовательно, амортизация в год составит 20%, за 2 месяца $20\% / 12 * 2 = 3,33 \%$.

$$K_{\text{ипс}} = 17699 * 0,0333 = 590 \text{ руб.}$$

Прочие затраты включают в себя затраты на поддержание компьютера и программного обеспечения, а также затраты на электричество и интернет. Данные затраты принято оценивать от 3% до 5% от суммы остальных затрат. В данном случае затраты составят 5%.

$$K_{\text{проч}} = (33835 + 1248 + 590) * 0,05 = 1784 \text{ руб.}$$

Подставим значения в формулу (1) для расчета проектных затрат.

$$K_{\text{пр}} = 33835 + 1248 + 590 + 1784 = 37457 \text{ руб.}$$

$K_{\text{тс}}$ – затраты на технические средства управления составят 0 рублей, поскольку используется компьютер предприятия.

$K_{\text{лс}}$ – затраты на линии связи. Так как у ЦУР РХ есть свой сервер, то затраты на линии связи отсутствуют.

$$K_{\text{лс}} = 0 \text{ руб.}$$

$K_{\text{по}}$ – затраты на программные средства состоят из затрат на Telegram и операционную систему. Операционная система уже установлена на компьютерах предприятия, а Telegram – бесплатное приложение. Поэтому затраты равны нулю.

$K_{\text{ин}}$ – затраты на формирование информационной базы. Информационная база состоит из списков органов власти, критериев и пользователей системы. Все эти данные уже внесены в базу предприятия, но на этапе внедрения потребуется ее актуализация. На данный вид работы у ответственного сотрудника ЦУР РХ будет затрачен примерно 1 рабочий день. Поэтому затраты составляют

$$K_{\text{ин}} = 25987 / 22 * 1,302 = 1538 \text{ руб.}$$

$K_{\text{об}}$ – затраты на обучение персонала. Пользователи системы имеют большой опыт работы с ботами Telegram, поскольку данная система используется для автоматизации многих внутренних процессов. Поэтому обучение персонала не требуется, затраты равны 0.

$K_{\text{оп}}$ – затраты на опытную эксплуатацию. Система будет тестироваться 7 дней по 2 часа. Суммарное время составит 14 часов или же 2 рабочих дня.

$$K_{\text{оп}} = 25987 / 22 * 2 * 1,302 = 3076 \text{ руб.}$$

Подставляем значения в формулу (1) для расчета капитальных затрат:

$$K = 37457 + 1538 + 3076 = 42071 \text{ руб.}$$

Состав капитальных затрат представлен в виде диаграммы на рисунке 32.



Рисунок 32 – Состав капитальных затрат

3.3 Расчет эксплуатационных затрат

Для расчета эксплуатационных затрат используется следующая формула

$$C = C_{зп} + C_{ао} + C_{то} + C_{лс} + C_{ни} + C_{проч}, \quad (3)$$

где $C_{зп}$ – зарплата персонала, работающего с информационной системой;

$C_{ао}$ – амортизационные отчисления;

$C_{то}$ – затрата на техническое обслуживание;

$C_{лс}$ – затраты на использование глобальных сетей;

$C_{ни}$ – затраты на носители информации;

$C_{проч}$ – прочие затраты.

$C_{зп}$ – для поддержания работы системы необходимо раз в месяц проводить работу над обнаруженными ошибками. Работы будут проводиться полный рабочий день.

$$C_{зп} = 25987 / 22 * 12 * 1,302 = 18456 \text{ руб.}$$

C_{ao} – амортизационные отчисления не будут учитываться, поскольку используется компьютер компании.

$C_{то}$ – затраты на техническое обслуживание. Аналогично с амортизационными отчислениями – не учитываются, т.к. компьютер компании.

$C_{лс}$ – затраты на использование глобальных сетей. Дополнительных затрат на использование глобальных сетей нет.

$C_{ни}$ – затраты на носители информации. Поскольку система расположена на компьютерах компании, то затраты на носители учитываться не будут.

$C_{проч}$ – прочие затраты. Прочие затраты составят 5% от суммы всех затрат.

$$C_{проч} = 18456 * 0,05 = 923 \text{ руб.}$$

Рассчитаем эксплуатационные затраты по формуле (3)

$$C = 18456 + 923 = 19379 \text{ руб.}$$

Перечень эксплуатационных затрат представлен в виде диаграммы на рисунке 33.



Рисунок 33 – Структура эксплуатационных затрат

3.4 Расчет прямых расходов

Прямые расходы считаются по следующей формуле

$$DE = DE_1 + DE_2 + DE_3 + DE_4 + DE_5 + DE_6 + DE_7 + DE_8, \quad (4)$$

где DE_1 - капитальные затраты;

DE_2 - расходы на управление ИТ;

DE_3 - расходы на техническую поддержку АО и ПО;

DE_4 - расходы на разработку прикладного ПО внутренними силами;

DE_5 - расходы на аутсорсинг;

DE_6 - командировочные расходы;

DE_7 - расходы на услуги связи;

DE_8 - другие группы расходов.

DE_1 – считалось ранее в капитальных затратах и составляет 42071 руб.

DE_2 – расходы на управление составляют эксплуатационные затраты на оплату труда персонала ($C_{зп}$) и составляют 18456 руб.

DE_3 – расходы на техническую поддержку включают в себя затраты на техническое обслуживание ($C_{то}$) и амортизационные отчисления ($C_{ао}$). Данные затраты равны нулю, поэтому $DE_3 = 0$.

DE_4 – для системы не требуется прикладное ПО, поэтому расходы равны 0.

DE_5 – расходы на аутсорсинг равны нулю, поскольку системе не требуются внешние источники.

DE_6 – командировки для сотрудников не предусмотрены, поэтому расходы равны нулю.

DE_7 – расходы на услуги связи состоят из затрат на использование глобальных сетей ($C_{лс}$) и затрат на носители информации ($C_{ни}$). Данные затраты равны нулю, поэтому $DE_7 = 0$.

DE_8 – прочие расходы примерно составляют 3% от всех других расходов ($DE_1 - DE_7$).

$$DE_8 = 0,03 * (42071+18456) = 1816 \text{ руб.}$$

Подставим полученные значения в формулу (4).

$$DE = 42071+18456+1816 = 62343 \text{ руб.}$$

3.5 Расчет совокупной стоимости владения системой

Для расчета затрат на реализацию проекта используется формула

$$TCO = DE + IC_1 + IC_2, \tag{5}$$

где DE – прямые расходы;

IC_1 – косвенные расходы первой группы;

IC_2 – косвенные расходы второй группы.

Прямые расходы уже были посчитаны ранее – $DE = 62343$ руб.

Косвенные расходы первой группы (IC_1) – это расходы на исправление ошибок в системе, вызванных неправильным проектированием системы. Поскольку при создании системы были учтены все риски, данные расходы равны нулю.

Косвенные расходы второй группы (IC_2) – это расходы на ошибки в организации функционирования системы. Данные расходы также составят 0 рублей, поскольку выявленные недостатки в организации функционирования системы были исправлены на этапе опытной эксплуатации [11].

Таким образом, рассчитав все возможные затраты, можно провести расчет совокупной стоимости владения системой по формуле (5).

$$TCO = 62343 + 0 + 0 = 62343 \text{ руб.}$$

3.6 Оценка рисков реализации проекта

Риски реализации проекта – это всевозможные ситуации, которые могут помешать, замедлить или вовсе остановить создание проекта. Необходимо учесть как можно больше таких рисков, составить для них решения или пути обхода. Перечень рисков представлен в таблице 5.

Таблица 5 – Риски реализации проекта

Название риска	Описание риска	Уровень влияния	Вероятность возникновения	Возможные решения
Риск соответствия	Несоответствие требованиям заказчика	Высокий	Низкая	Консультации с заказчиком, смена методологии разработки на Agile/Scrum.
Операционный риск	Непредвиденные расходы	Средний	Средняя	Оформление доп. договора
Риск денежных потоков	Задержка финансирования	Высокий	Низкая	Авансовая оплата
Риск блокировки Telegram	Блокировка мессенджера Telegram на территории государства	Высокий	Низкая	Сделать систему более гибкой и адаптивной для быстрой миграции в другой мессенджер

Риск соответствия имеет высокий уровень влияния, поскольку несоответствие требованиям может привести к сильным изменениям в проекте. Вероятность возникновения низкая, поскольку есть точное техническое задание, а также проводятся регулярные консультации с заказчиком с целью уточнения деталей.

Операционный риск имеет средний уровень влияния, поскольку непредвиденные расходы могут замедлить создание продукта. Поскольку расходы являются непредвиденными, то вероятность возникновения будет средней.

Риск денежных потоков связан с задержкой финансирования, что может серьезно повлиять на реализацию проекта, поэтому степень влияния высокая. Вероятность возникновения низкая, поскольку время на разработку выделено немного и стоимость проекта небольшая.

Риск блокировки Telegram может оказать сильнейшее влияние на проект, поскольку используются инструменты для работы с Telegram API и программный код бота пишется для этого мессенджера, поэтому степень влияния высокая. Вероятность низкая, поскольку Telegram независимая разработка, которая не нарушает законы страны.

3.7 Выводы по разделу «Определение совокупной стоимости владения проектом»

Совокупная стоимость проекта составила 62343 рубля, из которых 42071 рубль – капитальные затраты, а 18456 рублей – эксплуатационные затраты и риски. Затраты на проект получились не слишком большими, несмотря на то что проект состоит из двух частей: Telegram-бота и веб-приложения. Основную часть составили капитальные затраты. В капитальные затраты была включена оплата труда программиста за 2 месяца работы по 4 часа в день и стоимость оборудования и программного обеспечения для создания системы. Эксплуатационные затраты получились меньше, поскольку система была разработана на языке программирования PHP для уже сформированной ИТ-инфраструктуры ЦУР. Были учтены возможные риски, в том числе возможная, но маловероятная, блокировка мессенджера Telegram.

Поскольку данный проект создавался для автоматизации внутреннего процесса, то получать прибыль от продажи заказчик не будет. Выгоду от создания и

использования системы составляет оптимизация и ускорение внутренних процессов предприятия.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы в Центре управления регионом Республики Хакасия была разработана информационная система рейтингования органов региональной и муниципальной власти Республики Хакасия, состоящая из Telegram-бота для сбора рейтинга и формирования отчета и веб-приложения для построения графиков и таблиц рейтингования.

Сначала была разработана концепция IT-проекта, проанализированы аналогичные программные продукты и обоснована необходимость собственной разработки, поскольку программные продукты конкурентов не подходили под особенности процесса.

Для хранения информации была спроектирована реляционная база данных. База данных используется для работы сразу с чат-ботом и веб-приложением. Также база данных учитывает различные группы пользователей, с помощью чего можно закрыть доступ неизвестным лицам, а также выделить администраторов системы с доступом к расширенному функционалу бота.

Был обоснован выбор сред разработки для требуемых языка программирования PHP и СУБД MySQL. Были оценены самые популярные решения и выбраны наиболее эффективные для данного проекта.

Разработан Telegram-бот, с набором встроенных клавиатур для упрощения процесса оценивания для пользователей, а также с возможностью запрашивать отчет для администраторов. Бот автоматически запрашивает информацию из базы данных и формирует отчет по всем требованиям.

Было создано веб-приложение для отображения информации о рейтинговании в виде графиков и таблиц. На графике можно проследить тенденцию работы органов власти, посмотреть оценки всех органов власти за любой прошедший месяц. В таблице каждая ячейка покрашена в цвет в зависимости от оценки органа власти за месяц. Таким образом, можно быстро увидеть, какой орган власти работает успешно, а какой отстает. Также в таблице выведено среднее значение за прошедший год для каждого органа власти.

Сначала была предусмотрена возможность отображения оценок всех органов власти разом на одном графике, из-за чего читабельность и наглядность графиков и таблиц сильно снижались. Поэтому в приложении присутствуют кнопки для переключения между видами органов власти: ОМСУ и РОИВ, тем самым сохраняя наглядность графиков и таблиц.

Проведена оценка затрат на реализацию проекта по методике расчёта совокупной стоимости владения. В результате стоимость проекта составила 62343 руб. В стоимость включены затраты на разработку и внедрение системы, а также учтены возможные риски и их влияние на разрабатываемую систему. Поскольку система разрабатывалась для автоматизации внутренних процессов, то прибыль от продажи проекта не рассчитывалась.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Правительство Республики Хакасия // r-19.ru : [сайт]. – URL: <https://r-19.ru/documents/140/105469.html> (дата обращения: 23.05.23).
2. Сервис опросов, тестов и аналитики Testograf // testograf.ru : [сайт]. – URL: <https://www.testograf.ru/> (дата обращения: 24.05.23).
3. IDEF0. Знакомство с нотацией и пример использования // infostart.ru : [сайт]. – URL: <https://infostart.ru/1c/articles/640962/> (дата обращения: 24.05.23).
4. Методология IDEF3 // itteach.ru : [сайт]. – URL: <https://itteach.ru/bpwin/metodologiya-idef3> (дата обращения: 25.05.23).
5. Лучшие IDE для PHP // gb.ru : [сайт]. – URL: https://gb.ru/posts/php_ides (дата обращения: 25.05.23).
6. Лучшие инструменты для разработки и администрирования MySQL // habr.com : [сайт]. – URL: <https://habr.com/ru/post/142385/> (дата обращения: 26.05.23).
7. Проектирование базы данных MySQL // pro-prof.com : [сайт]. – URL: <https://infostart.ru/1c/articles/640962/> (дата обращения: 28.05.23).
8. Создание Excel документа на PHP // nicknixer.ru : [сайт]. – URL: <https://nicknixer.ru/programmirovanie/sozdanie-excel-dokumenta-na-php-generaciya-xls-fajlov/> (дата обращения: 28.05.23).
9. How to use Morris.js chart with PHP & Mysql // webslesson.info : [сайт]. – URL: <https://www.webslesson.info/2017/03/morrisjs-chart-with-php-mysql.html> (дата обращения: 28.05.23).
10. Вывод HTML-таблицы с функцией сортировки на PHP + MySQL // snipp.ru : [сайт]. – URL: <https://snipp.ru/php/sort-table-php> (дата обращения: 29.05.23).
11. Оценка совокупной стоимости владения информационной системой // studme.org : [сайт]. – URL: https://studme.org/184950/informatika/otsenka_sovokupnoy_stoimosti_vladieniya_informatsionnoy_sistemoy (дата обращения: 30.05.23).

Выпускная квалификационная работа выполнена мной самостоятельно. Используемые в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография 11 наименований.

Один экземпляр сдан на кафедру.

« ____ » _____ 2023 г.

_____ Михайлов Павел Дмитриевич
подпись

