

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

Агрегатор новостей с возможностью их добавления для ЦУР Республики
Хакасия

Руководитель _____ доцент, кандидат. физ-мат.наук А.Н. Таскин
подпись, дата

Выпускник _____ Д.А. Кириллов
подпись, дата

Консультанты
по разделам:

Экономический _____ М.А. Буреева
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 Кириллову Данилу Алексеевичу

Группа ХБ 19-02

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Агрегатор новостей с возможностью их добавления для ЦУР Республики Хакасия

Утверждена приказом по институту № 283 от 11.05.2023 г.

Руководитель ВКР: А. Н. Таскин, канд. физ-мат. наук, ХТИ – филиал СФУ

Исходные данные для ВКР: заказ ХТИ – филиала СФУ.

Перечень разделов ВКР:

1. Анализ предметной области.
2. Описание разработки информационной системы.
3. Оценка экономической эффективности разработки и внедрения информационной системы для профориентационного тестирования.

Перечень графического материала: нет

Руководитель ВКР

А. Н. Таскин

Задание принял к исполнению

Д. А. Кириллов

«11» мая 2023г

РЕФЕРАТ

Выпускная квалификационная работа по теме «Управление IT-проектом «Разработка Агрегатора новостей с возможностью их добавления для ЦУР Республики Хакасия» содержит 91 страницу текстового документа, 25 иллюстраций, 6 использованных источников.

НОВОСТИ, СРЕДСТВА МАССОВОЙ ИНФОРМАЦИИ, НОТАЦИЯ ДРАКОН, IDEF3, DFD, ВЕБ-ПРИЛОЖЕНИЕ, PHP, MYSQL, XAMPP, RSS, HTML, CSS

Предмет: разработка новостного агрегатора на основе парсинга новостных RSSлент.

Объект: деятельность Центра управления регионом Республики Хакасия в области информационных технологий.

Цель: создание новостного агрегатора, который будет позволять пользователям получать актуальные новости разных тематик и от разных источников на одной платформе.

В результате работы была проанализирована деятельность ЦУР, спроектировано и разработано веб приложение, включая клиентскую и серверную часть, а также была выполнена оценка экономической эффективности.

SUMMARY

The theme of the graduation thesis is «IT Project Management: News Aggregator for Regional Center of IT Management in Republic of Khakassia». It consists of 91 pages, 25 figures, 6 references.

NEWS, MASS MEDIA, DRAGON NOTATION, IDEF3, DFD, WEB APPLICATION, PHP, MYSQL, XAMPP, RSS, HTML, CSS

Subject: development of a news aggregator based on news feed parsings.

Object: activity of the Regional Center of IT Management in the Republic of Khakassia.

Purpose: to create a news aggregator to receive current news on various topics from different sources on a single platform.

Results: the activity of the Regional Center of IT Management has been analyzed; the web application has been designed, including the client and server parts, and the calculation of economic efficiency has been conducted.

СОДЕРЖАНИЕ

Введение.....	7
1 Анализ предметной области	9
1.1 Описание и анализ основной деятельности Центра Управления Регионом республики Хакасия	9
1.2. Анализ новостных ресурсов	13
1.3 Анализ существующих решений на рынке	15
1.4 Бизнес-процессы реализуемого приложения.....	21
Вывод по разделу «Анализ предметной области».....	28
2 Описание разработки Агрегатора.....	28
2.1 Определение архитектуры и средств разработки	28
2.2 Процесс разработки	30
2.2.1 Анализ новостных сайтов	30
2.2.2 Проектирование базы данных	31
2.2.3 Разработка парсера новостных сайтов.....	33
2.2.4 Разработка агрегатора новостей на основе парсера	35
2.2.4.1 Загрузка новостей в базу данных	37
2.2.4.2 Отображение новостей	38
2.2.4.3 Авторизация пользователя.....	39
2.2.4.5 Аутентификация пользователя	40
2.2.4.6 Создание RSS версии.....	40
2.2.4.7 Поиск новости	41
2.2.4.8 Удаление новости	42
2.2.4.9 Содержание новости.....	43
2.2.4.10 Администрирование	44
2.2.4.11 Конфигурация подключения	46
2.2.5 Запуск сервера с помощью XAMPP.....	47
Вывод по разделу «Описание разработки Агрегатора».....	48
3 Оценка совокупной стоимости и рисков системы.....	49
3.1 Расчёт затрат на разработку проекта	49
3.2 Расчёт эксплуатационных затрат	54
3.3 Расчёт показателей экономического эффекта.....	57
3.4 Маркетинговое сопровождение разрабатываемого продукта.....	58
3.5 Расчет совокупной стоимости владения системой.....	58
3.6 Оценка рисков реализации проекта	59
Вывод по разделу «Оценка совокупной стоимости и рисков системы».	61
Заключение	62
Список использованных источников	64
Приложение А	65
Приложение Б.....	67
Приложение В	70
Приложение Г.....	73
Приложение Д	74

Приложение Е.....	77
Приложение Ж.....	78
Приложение И.....	80
Приложение К.....	82

ВВЕДЕНИЕ

В наше время информационные технологии играют важную роль в жизни. Одной из важных областей в этой сфере является агрегация новостей. Агрегаторы новостей позволяют пользователям получать доступ к новостям разных источников в одном месте, что значительно облегчает процесс получения информации.

Цель данной работы - разработка парсера новостных сайтов для агрегатора новостей. Результатом работы будет являться функциональный парсер, который сможет извлекать информацию из новостных сайтов и сохранять ее в базу данных.

В процессе разработки парсера будут использованы современные технологии, такие как PHP, MySQL, HTML.

Предмет: разработка новостного агрегатора на основе парсинга новостных rss лент.

Объект: деятельность Центра управления регионом Республики Хакасия в области информационных технологий.

Цель: создание новостного агрегатора, который будет позволять пользователям получать актуальные новости разных тематик и от разных источников на одной платформе.

Задачи:

1. Изучить основную деятельность Центра управления регионом Республики Хакасия в области информационных технологий.
2. Разработать концепцию новостного агрегатора на основе парсинга новостных rss лент.
3. Проанализировать предметную область и существующие решения на рынке.
4. Определить бизнес-процессы реализуемого приложения.
5. Определить архитектуру и средства разработки: HTML, CSS, JavaScript, MySQL и PHP.

6. Разработать парсер новостных сайтов и агрегатор новостей на его основе.

7. Реализовать загрузку новостей в базу данных, отображение новостей, авторизацию и аутентификацию пользователей, поиск и удаление новостей, администрирование и конфигурацию подключения.

8. Запустить сервер с помощью XAMPP.

9. Сделать выводы по результатам выполненной работы.

1 Анализ предметной области

1.1 Описание и анализ основной деятельности Центра Управления Регионом республики Хакасия

С декабря 2020 года по поручению президента во всех субъектах РФ были созданы Центры управления регионами (ЦУР), единые пункты мониторинга и обработки поступающих от жителей жалоб в разных сферах жизни региона. Кроме того, на основе поступающих в ЦУР данных формируют аналитические материалы, характеризующие социально-экономическую ситуацию в области, и вырабатывают предложения по дальнейшему развитию региона.

Центр управления регионом — это формируемый в субъекте РФ проектный офис, создание и деятельность которого регламентируются нормативным правовым актом субъекта. ЦУР осуществляет:

- координацию работ по мониторингу и обработке всех видов обращений и сообщений, поступающих в органы и организации (по любым каналам), а также публикуемых в социальных сетях, мессенджерах и других средствах электронной массовой коммуникации (ЭМК);
- взаимодействие с гражданами через соцсети, мессенджеры и иные средства ЭМК по направлениям и тематикам деятельности ЦУР;
- оперативное реагирование на обращения;
- предоставление дополнительной информации в целях территориального и стратегического планирования развития регионов.

Учитываются обращения не только граждан РФ (в том числе проживающих за пределами территории страны), но и иностранцев и лиц без гражданства, а также юридических лиц любых организационно-правовых форм вне зависимости от места их государственной регистрации.

ЦУР является проектом по взаимодействию исполнительной власти с населением. Жалобы граждан формируются по отраслевым блокам (здравоохранение, образование, транспорт, энергетика, жилищно-коммунальное

хозяйство, дороги) и направляются ответственным для оперативного реагирования.

В центрах структурируется и анализируется большой массив информации, который затем можно использовать для исключения «слабого звена» и корректировки управленческих решений. Появление ЦУР повышает оперативность реакции региональных властей на существующие проблемы, а также позволяет предотвращать их возникновение в будущем благодаря анализу и системам мониторинга.

ЦУР играет важную роль в достижении цифровой трансформации страны — одной из национальных целей развития России на период до 2030 года.

Цифровизация обратной связи, рост понимания и доверия между гражданами и властью – одна из главных задач «Диалога».

Для мониторинга и работы с вопросами и обращениями людей используются федеральные системы «Инцидент Менеджмент» и «Госуслуги. Решаем вместе» – ключевые инструменты прозрачности. Такая схема работы помогает государству оперативно слышать граждан на разных цифровых площадках, решать их проблемы, координировать нацпроекты и программы помощи.

Управление социологии занимается исследованием общественного мнения. Специалисты направления реализуют целый комплекс задач, связанных со сбором, обработкой и анализом обратной связи от граждан в интернете. В работе управления используются уникальные авторские методики и современные информационные системы. Они позволяют не только эффективно проводить онлайн-исследования, но и разрабатывать собственные инструменты и аналитические продукты.

Центр как принимает жалобы по конкретным вопросам, так и анализирует информацию из различных публичных источников по актуальным для граждан вопросам. Исходя из практики работы ЦУР, жители обращаются по таким темам, как:

- некачественный ремонт дорог;
- установка мусорных контейнеров;
- школьное питание;
- низкие темпы строительства социальных объектов;
- качество и доступность медицинской помощи;
- вакцинация от коронавируса;
- проблемы с социальными пособиями.

После принятия жалобы сотрудники дают промежуточный ответ о направлении информации в ответственный орган, далее сообщают о принятом решении. Весь массив поступившей информации анализируется и предоставляется региональным властям в виде аналитического отчета.

Согласно постановлению президиума правительства республики Хакасия от 04.09.2020 № 120-п организация имеет следующую структуру (таблица 1).

Таблица 1 – Структура ЦУР

Наименование органа / должность	Роль
Аппарат Главы Республики Хакасия – Председателя Правительства Республики Хакасия	координатор
Государственный комитет цифрового развития и связи Республики Хакасия	технический куратор
Руководитель обособленного подразделения Центра компетенций ЦУР в Республике Хакасия (по согласованию)	руководитель
Руководители/заместители руководителей исполнительных органов государственной власти Республики Хакасия по тематикам: жилищно-коммунальное хозяйство, твердые коммунальные отходы, энергетика, образование, здравоохранение, дороги, транспорт, социальная защита	ответственные за отраслевые блоки по тематикам
Руководитель пресс-службы Главы Республики Хакасия – Председателя Правительства Республики Хакасия и Правительства Республики Хакасия	ответственный за работу с масс-медиа
Сотрудники исполнительных органов государственной власти/подведомственных учреждений Республики Хакасия по тематикам: жилищно-коммунальное хозяйство, твердые коммунальные отходы, энергетика, образование, здравоохранение, дороги, транспорт, социальная защита	руководители отраслевых блоков Центра управления регионом Республики Хакасия по тематикам
Сотрудники обособленного подразделения Центра компетенций ЦУР в Республике Хакасия (по согласованию)	руководители направлений Центра управления регионом Республики Хакасия

Также, согласно отчетам, в периоде с 04.2021 по 10.2021 ЦУР Республики Хакасия обработало более пяти тысяч заявок и отличился в области скорости обработки первичного ответа. Тогда как аналогичный средний показатель по

России составлял порядка пяти часов, в республике Хакасия он равен 2 часа 46 минут.

1.2. Анализ новостных ресурсов

За последние десятилетия рынок по предоставлению информации, претерпел грандиозные изменения. От традиционных газет и радио, люди все чаще узнают новости посредством телевидения и интернета [1].

Согласно социологическому исследованию, проводимому в городе Новосибирск в 2019 году, газеты и радио продолжают терять роль средств информирования [1].

Стоит отметить, что телевидение также постепенно теряет свои позиции: если в 2015 году 96% респондентов узнавали о новостях в стране, регионе и прочем с помощью телевизионного вещания, то к 2019 году этот процент снизился почти вдвое, достигнув отметки в 52% [1].

Единственные тенденции роста присутствуют только у интернета: от 49% в 2015 году, до 68% в 2019 году; и социальных сетей: от 15% до 27% [1].

Диаграмма исследования показана на рисунке 1

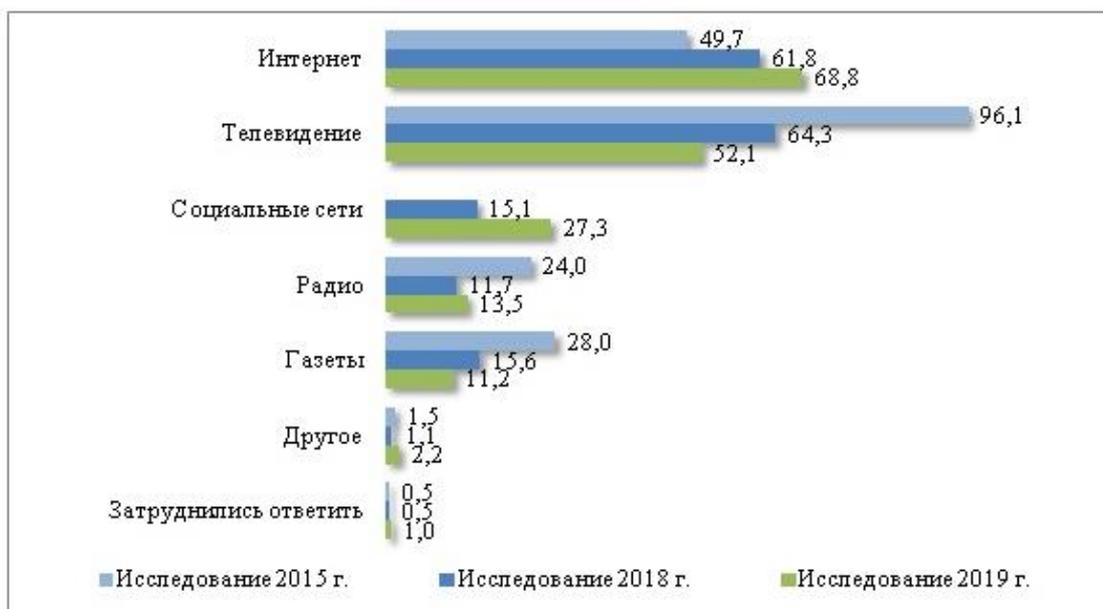


Рисунок 1 – Диаграмма исследования

Законодательство Российской Федерации определяет два вида средств массовой информации (СМИ): печатные и непечатные. В свою очередь к первым относятся: газеты, журналы и прочее, ко вторым: телеэфиры, радиопередачи, и интернет СМИ. Согласно реестру средств массовой информации (таблица 2) на 2020 год насчитывается 22,77 тысяч непечатных СМИ.

Таблица 2 – Количество СМИ в тысячах

Вид СМИ	2017	2018	2019	2020
Всего СМИ	77,58	72,07	65,85	65,0
Печатных СМИ	52,63	47,88	42,88	42,23

В феврале 2021 года фонд «Общественное мнение» провел опрос среди полутора тысяч человек в 104 населенных пунктах, расположенных в 53 регионах Российской Федерации. Данный опрос показал, что 56% россиян читают новости [2].

Исходя из всего вышеперечисленного, можно сказать, что средства массовой информации являются важной частью жизни многих людей. Также по данным государственного реестра можно судить об обильном количестве самих средств. СМИ бывают как узкопрофильные, как например «Матч» или «Форбс», так и широкого профиля: «РИА», «РТ» и прочие. Также, помимо агентств имеющих в подчинении журналистов, корреспондентов, операторов, интервьюеров и других работников, которые могут непосредственно собирать новости для агентства, существуют СМИ, специализирующиеся на сборе новостей со сторонних ресурсов, к подобному виду относится «Лента». Лента собирает новости с различных новостных ресурсов, но подавляющее их большинство – федеральные. Если взять рынок информационных агентств по республике, то он существенно беднее федерального. Говоря о новостных агрегаторах, специализирующихся на данном регионе, можно судить о полном их отсутствии. Отсюда и возникает актуальность разработки новостного агрегатора, специализирующегося на новостях республике Хакасия [3,4].

Реализуемая система должна решать ряд следующих задач:

- Оперативное информирование населения республики Хакасия по региону
- Информирование пользователей новостями о самом агрегаторе.

Данный агрегатор должен обладать следующим функционалом:

- Выполнение запросов к интернет СМИ республики Хакасия;
- Формирование базы данных, на основе полученных запросов;
- Формирование HTML документов, основанную на базе данных;
- Элементами администрирования сайтами такими как:
 - Добавление собственных новостей;
 - Редактирование и/или удаление собственных новостей.

Исходя из выше сказанного, была сформирована цель моделирования бизнес-процесса, а также определены задачи.

1.3 Анализ существующих решений на рынке

Как было сказано в предыдущей главе, рынок интернет СМИ является довольно обширным. Для проводимого анализа существующих проектных решений были использованы как русскоязычные новостные порталы, так и англоязычные.

Для удобства пользователя необходимо, чтобы реализуемый проект отвечал существующим стандартам пользовательского интерфейса и пользовательского сценария, которые были выявлены в данном разделе.

Первый этап подразумевает анализ пользовательского сценария пользователя новостных порталов.

На втором этапе выявлялись стандарты пользовательского интерфейса, такие как: заглавная страница, развертка новости, и основные функциональные блоки данных элементов.

В ходе анализа пользовательского сценария были просмотрены такие интернет СМИ как: «ТАСС», «РИА», «Times», «РБК» и прочие. На основе просмотра интерфейсов данных ресурсов была построена следующая диаграмма, продемонстрированная на рисунке 2.

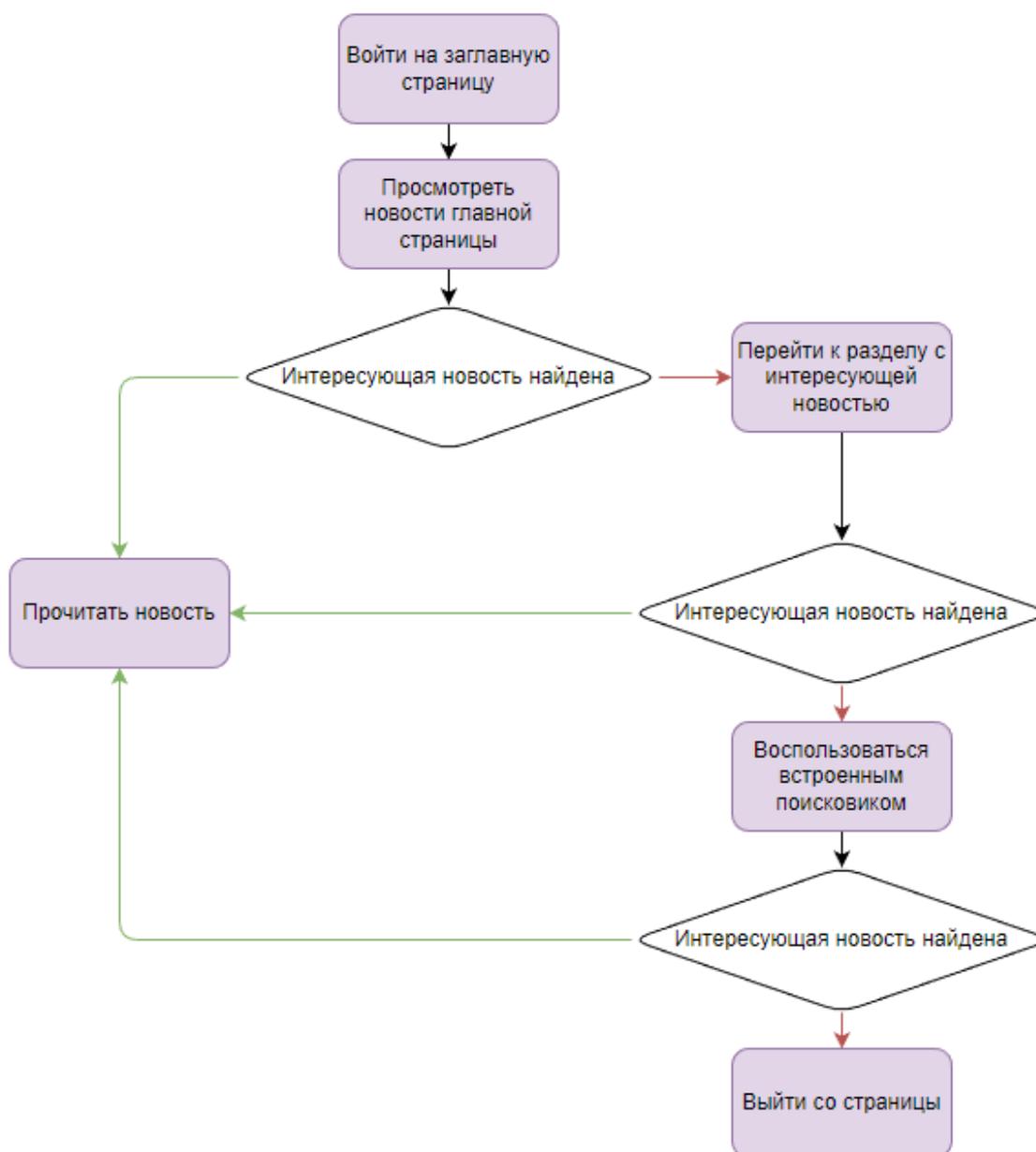


Рисунок 2 – Диаграмма в нотации «Дракон» пользовательского сценария

На втором этапе анализа были выбраны источники, описанные выше, проанализированы заглавные страницы и статьи с точки зрения пользовательского интерфейса.

Главные страницы отражают последние и «горячие» новости, опционально присутствуют изображения новостей. Также в большинстве новостных порталов в «шапке» (под шапкой здесь и далее подразумевается навигационная панель приложения, расположенная сверху) находятся новостные тематики, такие как: спорт, политика, искусство и другие. Демонстрация заглавных страниц «Тасс» и «Lenta.ru», произведена на рисунках 3,4.

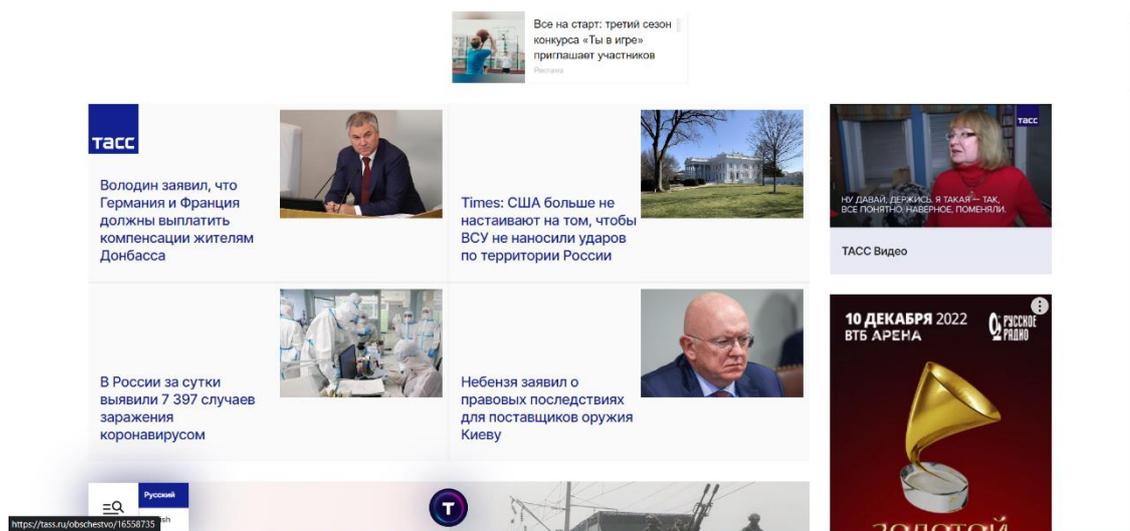


Рисунок 3 – Заглавная страница ТАСС

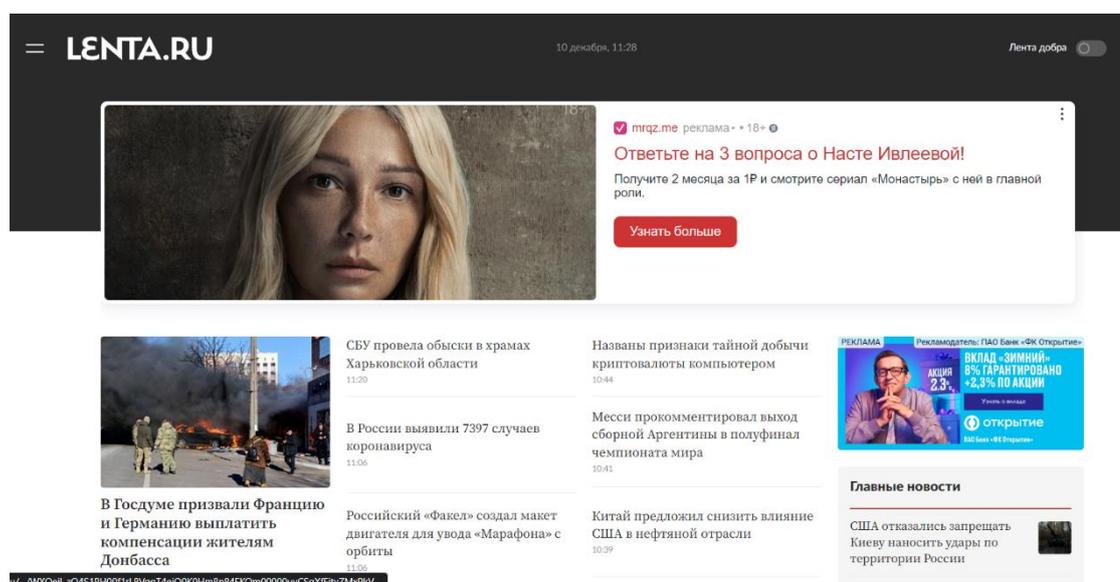


Рисунок 4 – Заглавная страница Лента.ру

Также стоит отметить, что ряд новостных порталов полностью избегают изображений на заглавных страницах, оставляя их только в статьях. Подобный подход можно наблюдать у «РБК». Заглавная страница данного новостного агентства изображена на рисунке 5

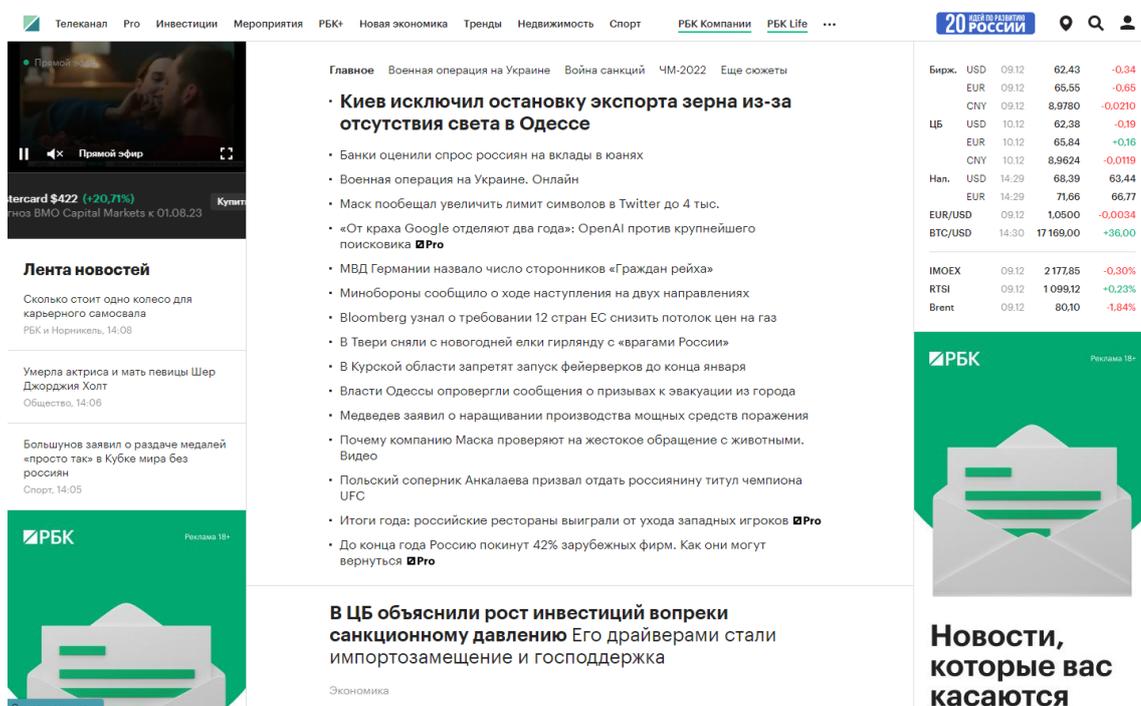


Рисунок 5 – Заглавная страница РБК

Новостные статьи соблюдают общую структуру, строящуюся на следующей последовательности:

- заголовок;
- краткая аннотация статьи;
- изображение (является опциональным решением);
- источники (данный структурный элемент, может отражать как репортера, участвовавшего при снятии репортажа, либо автора статьи, так и ссылку на другое средство массовой информации, опубликовавшей данный материал;
- дата публикации новости.

Данная концепция соблюдена, как в полном решении сайта, так и в обогатённых сводках сайта, о чем более подробно будет сказано далее. Статьи показаны на рисунках 6, 7.



Рисунок 6 – статья на сайте РИА

TheUpshot

The (Incomplete) Revolution in Counting Abortions

Researchers know more than before, but incomplete data is still an issue as more women try to sidestep restrictions.

Give this article 8



The first of two pills taken for a medical abortion at a Planned Parenthood clinic in October in Kansas City, Kan. Charlie Riedel/Associated Press



By Claire Cain Miller and Margot Sanger-Katz

Dec. 8, 2022

Among women in the United States who no longer have an abortion clinic nearby, many are turning to abortion pills ordered

Рисунок 7– статья на сайте NY Times

RSS (англ. Rich Site Summary — обогащённая сводка сайта) — семейство XML-форматов, предназначенных для описания лент новостей, анонсов статей, изменений в блогах и т. п.

Общая структура RSS (рисунок 8), как вид XML-форматов предназначена для описания объектов. Так, например, можно предоставить список животных с указанием вида, возраста и имени.

```
<item>
  <name>Лиза</name>
  <type>Goat</type>
  <age>13</age>
</item>

<item>
  <name>Джек</name>
  <type>Dog</type>
  <age>13</age>
</item>

<item>
  <name>Соня</name>
  <type>Cow</type>
  <age>13</age>
</item>
```

Рисунок 8 – пример RSS документа

Помимо обычной структуры файлов предоставленных в HTML формате большая часть новостных порталов обладает RSS версией. Следующим образом (рисунок 9) выглядит rss версия «Лента.ру». Так как для rss нет общих стандартов наименования, одни и те же элементы на разных сайтах при разном названии могут обозначать одно и то же. Одним из таких примеров является поле «Autor» и «Source», которые в большинстве случаев обозначает ссылку на изначальный материал. Несмотря на расхождения в наименовании, структура объектов в большинстве случаев повторяет структуру статей, описанную выше.

```

<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom">
  <channel>
    <language>ru</language>
    <title>Lenta.ru : Новости</title>
    <description>Новости, статьи, фотографии, видео. Семь дней в неделю, 24 часа в сутки.</description>
    <link>https://lenta.ru</link>
    <image>
      <url>https://lenta.ru/images/small_logo.png</url>
      <title>Lenta.ru</title>
      <link>https://lenta.ru</link>
      <width>134</width>
      <height>22</height>
    </image>
    <atom:link rel="self" type="application/rss+xml" href="http://lenta.ru/rss/news"/>
  </channel>
  <item>
    <guid>https://lenta.ru/news/2022/12/11/brit/</guid>
    <author>Никита Абрамов</author>
    <title>Великобритания пессимистично отнеслась к перспективам переговоров Москвы и Киева</title>
    <link>https://lenta.ru/news/2022/12/11/brit/</link>
    <description>
      <![CDATA[Глава министерства иностранных дел Великобритании Джеймс Клеверли пессимистично оценил перспективы переговоров между Россией и Украиной. Слот Клеверли, президент России Владимир Путин пока не готов вести переговоры «в духе доброй воли».]]>
    </description>
    <pubDate>Sun, 11 Dec 2022 15:07:22 +0300</pubDate>
    <enclosure url="https://icdn.lenta.ru/images/2022/12/11/14/2022121114475821314/pic_baf954f785781380b2a29f34a2c9b503.jpg" type="image/jpeg" length="38622"/>
    <category>Мир</category>
  </item>
  <item>
    <guid>https://lenta.ru/news/2022/12/11/gazotrav/</guid>
    <author>Елизавета Новикова</author>
    <title>Российская семья погибла из-за отравления угарным газом в Подольске</title>
    <link>https://lenta.ru/news/2022/12/11/gazotrav/</link>
    <description>
      <![CDATA[В однокомнатной квартире в Подольске обнаружили тела супругов и их 18-летнего сына без следов насильственной смерти. По предварительным данным трех человек будет известна после проведения ряда экспертиз.]]>
    </description>
    <pubDate>Sun, 11 Dec 2022 14:57:34 +0300</pubDate>
    <enclosure url="https://icdn.lenta.ru/images/2022/12/11/14/20221211144710293/pic_1f88528ff9e5e83785852f7924920fcd.jpg" type="image/jpeg" length="25163"/>
    <category>Силловые структуры</category>
  </item>
  <item>
    <guid>https://lenta.ru/news/2022/12/11/ankalaevcomment/</guid>
    <author>Ксения Курашкина</author>
    <title>Анкалаев объяснил слова о возможном уходе из UFC</title>
    <link>https://lenta.ru/news/2022/12/11/ankalaevcomment/</link>
    <description>
      <![CDATA[Российский боец Магомед Анкалаев опроверг слова о том, что он уходит из UFC. «После боя я сказал, что больше не хочу драться в этом городе из UFC», – заявил он. Ранее Анкалаев сразился за титул чемпиона с Яном Блаховичем. Схватка окончилась ничей.]]>
    </description>
    <pubDate>Sun, 11 Dec 2022 14:54:00 +0300</pubDate>
    <enclosure url="https://icdn.lenta.ru/images/2022/12/11/14/20221211140949753/pic_621b981cef412828457dca80336bee8f.jpg" type="image/jpeg" length="30136"/>
  </item>

```

Рисунок 9 – RSS версия сайта Lenta.ru

Несмотря на то, что RSS-версии сайтов являются распространенной практикой, реализуемый проект не подразумевает ее наличие. Поэтому данное решение не будет включено в конечный продукт. Тем не менее, стоит отметить, что рассмотрение наличия данного нюанса является одним из основополагающих для реализации конечного продукта, о чем будет рассказано в последующих главах.

1.4 Бизнес-процессы реализуемого приложения

Говоря о реализуемом приложении, необходимо разобраться в структуре его работы. Так как реализуемое приложение разрабатывается, как и с серверной точки зрения, так и с клиентской точки зрения, бизнес-процессы будут расписаны как с точки зрения сервера, так и с точки зрения конечного пользователя.

Общая структура действий серверной части такова:

- обратится к rss версиям новостных версий сайтов;
- получить новые данные;
- конвертировать наименования объектов и полей под стандарты приложения;
- отправить новые данные в базу данных;
- формировать новые html документы на основании базы данных.

Теперь рассмотрим каждую часть по-отдельности.

Для первого этапа существуют два варианта реализации: мониторинг страниц в некоторый промежуток времени и непрерывный мониторинг страниц. Существенная разница в реализации первого и второго варианта заключается лишь в промежутке времени, через который цикл проверки сайтов переходит на следующую итерацию. В первом случае промежуток времени – это константная величина, например 5 минут или 1 час, во втором – цикл мониторинга начинается сразу же после завершения предыдущей итерации цикла. В среднем на новостной портал регионального либо же муниципального уровня приходится не больше 30 новостей в день, то есть в среднем новости появляются раз 28,8 минут. В свою очередь одна итерация цикла проверки новостей, при отсутствии новых данных, порядка пары секунд (В данном случае подразумевается, что цикл начинается заново, как только дойдет до существующей новости в базе. А также не учитывается пропускная способность интернета и других параметров). Таким образом непрерывный цикл за сутки повторится более 43 000 раз. Несмотря на то, что данный цикл не является ресурсозатратным, потребность в таком количестве итераций отсутствует. Поэтому в данном решении будет использован первый вариант.

Второй этап как было сказано ранее завязан на записи данных с сайта и проверки каждой новости на существующую в базе. Существенных разбирательств данный пункт не требует, за исключением одного момента связанным со следующим пунктом, поэтому данный нюанс будет рассказан при рассмотрении следующего этапа.

Как уже было сказано в предыдущей главе, rss структура каждого отдельного сайта отличается друг от друга. Несмотря на то, что каждая новость подвергается конвертированию, последняя новость сайта также сохраняется в первоизданном виде. Данная особенность связана с предыдущим этапом: получением новых данных путем проверки на наличие новости. Дабы избежать обратного конвертирования новостей из стандартов реализуемого приложения и структурой новостного сайта, являющегося источником, последняя скопированная новость сохраняется для сверки в следующей итерации проверки.

Четвертый этап производится после конверсии данных. Данные в новой структуре отправляются в базу данных.

Если предыдущие пункты описывали логику сервера в контакте получения новых данных, то данный этап отражает формирование данных для пользователя на основании, запрашиваемого пользователем URL адреса (Например, example.ru/1).

DFD диаграмма бизнес-процессов сервера продемонстрированы на рисунках 10, 11.



Рисунок 10 – DFD диаграмма

Декомпозиция процесса 1



Рисунок 11 – Декомпозиция процесса 1

ER диаграмма предварительной базы данных продемонстрирована на рисунке 12

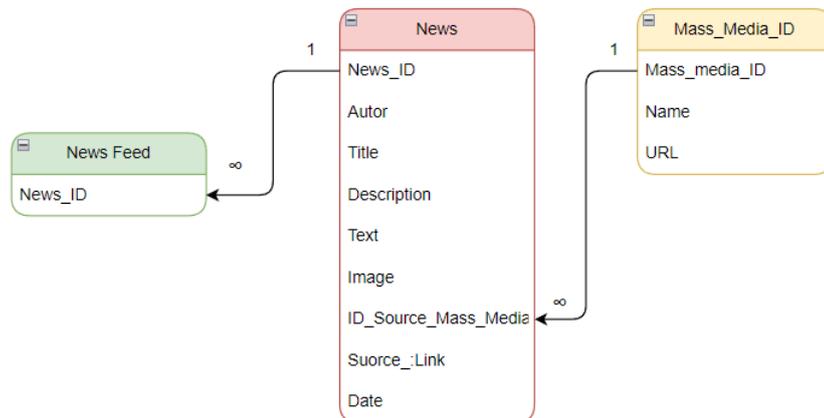


Рисунок 12 – ER Диаграмма предварительной Базы данных

Реализуемая база данных подразумевает наличие трех функционально обязательных таблиц названных: «News», «Mass_Media», «News_Feed».

Первая таблица – сборник новостей, отражающих номер статьи «News_ID», автора «Autor», заголовок статьи «Title», описание/аннотацию статьи «Description», текст «Text», картинку «Image», уникальный

идентификатор новостного агентства, являющегося источником новости «ID_Source_Mass_Media», ссылку на источник «Source_Link» и дату «Date».

News_ID – ключ и автоитерируемое поле с целочисленным значением. Является как обозначением новости в базе данных, так и частью URL согласно которому будет формироваться HTML документ со статьей.

Autor – строковое поле, принимающее по умолчанию значение NULL. Обозначает автора статьи если таковой имеется.

Title – строковое поле, обязательное к заполнению. Отражает заголовок статьи.

Description – строковое поле, обязательное к заполнению. Отражает аннотацию/описание статьи.

Text – строковое поле, обязательное к заполнению, отражающее текст статьи.

Image – строковое поле, принимающее по умолчанию значение NULL. MySQL не имеет возможности сохранять изображения, поэтому данное поле хранит в себе URL ссылку на существующее изображение в сети интернет. Не для каждой новости есть изображение, поэтому данное поле является необязательным.

ID_Source_Mass_Media – внешний ключ для таблицы Mass_Media, принимает целочисленные значения. Обязательно к заполнению.

Source_Link – строковое поле, обязательное к заполнению. Отражает URL изначальной статьи.

Date – поле принимающее в качестве параметра полную дату (ДД.ММ.ГГ ЧЧ:ММ). Дата написания статьи.

Сущность Mass_Media – справочник всех интернет СМИ, из которых производится сохранение новостей. Имеет три поля: Mass_Media_ID – первичный ключ, автоитерируемое поле с целочисленным значением; Name – название СМИ; URL – строковое поле, обозначающее url адрес интернет СМИ.

Прежде чем перейти к описанию сущности News_Feed, необходимо описать его необходимость. Предположим, что у нас есть два интернет-агентства

с которых мы собираем новости. Сначала сервер проверяет новости с одного сайта и сохраняет их в базе данных, после переходит ко второму. Предположим, что база данных пустая и из первого агентства мы сохранили 10 новостей, со второго еще 10. Соответственно первая запись статьи из второго агентства в сущность News будет иметь первичный ключ «11». Но время ее написания меньше, чем у записи с первичным ключом «3». Новостные порталы отображают первыми новейшие статьи. Для отображения новейших статей и предназначена сущность News_Feed имеющая в себе лишь одно поле News_ID. Данная сущность обновляется процедурно, после добавления сервером последней новой записи, сортируя новости по дате написания. Хотя SQL позволяет сортировать записи по определенному полю и полю, приминающее дату, отлично для этого подходит, но сортировка данной таблицы по данному полю обозначала бы, что данная сортировка происходила бы каждый раз, при запросе пользователем главной страницы. С каждой добавленной записью данная сортировка становится ресурсозатратнее.

Демонстрация бизнес-процессов «как будет» произведена с помощью диаграмм в нотации IDF0 и показана на рисунках 13 - 16.



Рисунок 13 – IDEF0 «Как было»

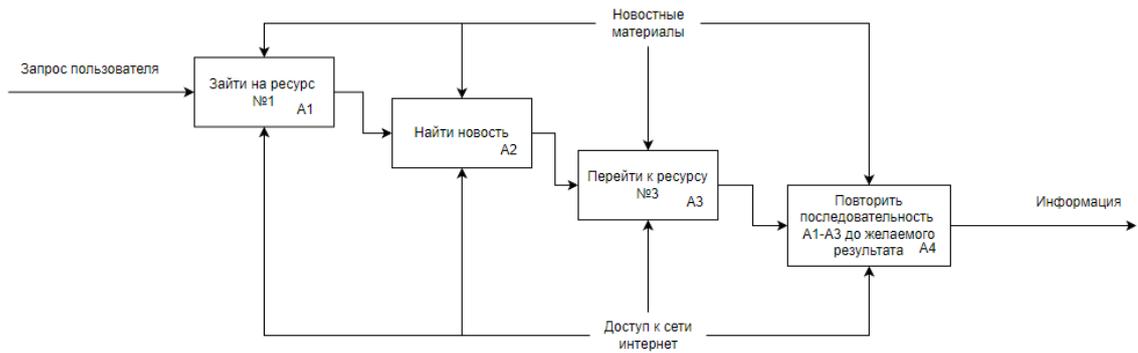


Рисунок 14– Декомпозиция блока A0



Рисунок 15 – IDEF0 «Как будет»

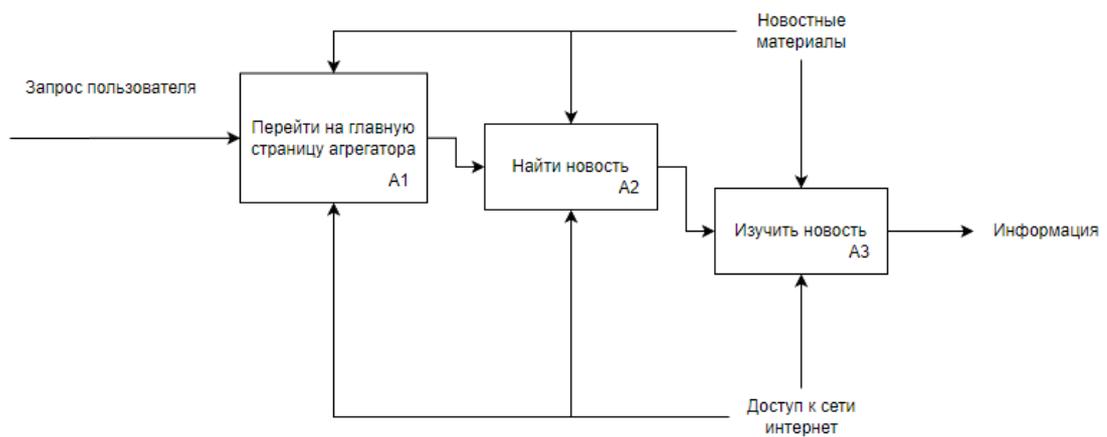


Рисунок 16 – Декомпозиция блока A0

Вывод по разделу «Анализ предметной области»

В разделе был проведён анализ ЦУР. Исходя из полученных данных было выполнено моделирование бизнес-процесса для разработки веб приложения «Агрегатор новостей РХ»:

- разработана концепция для веб приложения;
- выполнен анализ аналогичный программных продуктов и обоснована необходимость собственной разработки;
- выполнено структурное моделирование для описания бизнес-процесса, подлежащего автоматизации (*IDF0, IDEF3, DFD, Дракон*);
- разработана *ER*-диаграмма базы данных;
- выполнен сравнительный анализ и обоснован выбор программных средств веб приложения;
- разработано техническое задание на создание веб приложения профориентационного тестирования;
- выполнено технико-экономическое обоснование, на основе которого была доказана эффективности данной разработки.

Материалы данной курсовой работы будут использоваться для разработки веб приложения «Агрегатор новостей РХ», а также будут включены в выпускную квалификационную работу.

2 Описание разработки Агрегатора

2.1 Определение архитектуры и средств разработки

Архитектура разрабатываемой системы включает три слоя, это поможет существенно разгрузить web-сервер, что будет способствовать увеличению скорости работы ИС.

Механизм взаимодействия продемонстрирован на рисунке 17.



Рисунок 17 – IDEF3 Диаграмма

Модель взаимодействия web-приложения на рисунке 18.



Рисунок 18 – Модель взаимодействия

Для разработки приложения были выбраны следующие средства программной разработки:

- *HTML* – язык гипертекстовой разметки, который позволяет расположить на странице элементы (текст, изображения, ссылки и т. д.) [6];
- *CSS* – таблицы стилей, которые позволяют придать каждому элементу страницы нужный вид[7];
- *JavaScript* – скрипт, позволяющий управлять элементами на странице[8];
- *MySQL* – СУБД, хранилище данных сайта[9]
- *PHP* – язык, который формирует страницы и выполняет логику работы сайта на серверной стороне[10].

2.2 Процесс разработки

2.2.1 Анализ новостных сайтов

В современном мире новости являются важным источником информации для многих людей. Интернет-агрегаторы новостей сегодня позволяют получать сразу несколько источников новостей на одной странице. Однако, чтобы создать такой агрегатор, необходимо собирать новости с различных новостных сайтов.

В этой работе рассматривается процесс разработки парсера новостных сайтов для агрегатора новостей. Парсер новостных сайтов - это программное обеспечение, которое собирает информацию с различных новостных сайтов и предоставляет ее в удобном для дальнейшей обработки формате.

Перед тем, как начать разработку парсера, необходимо изучить структуру HTML-кода страниц новостных сайтов и выделить теги, содержащие необходимую информацию, такую как заголовок новости, ее краткое описание, ссылку на полный текст новости и т.д.

Затем необходимо создать соединение с базой данных MySQL и создать таблицу для хранения собранных новостей. Для удобства использования собранных данных в агрегаторе новостей необходимо сохранять данные в формате RSS или JSON.

Далее следует реализовать процесс сбора данных с новостных сайтов с использованием языка программирования PHP и библиотеки Simple HTML DOM. Simple HTML DOM - это библиотека PHP для работы с HTML-кодом, которая позволяет извлекать информацию из HTML-страниц с помощью CSS-селекторов.

После сбора данных с новостных сайтов и сохранения их в базе данных MySQL, необходимо реализовать процесс отображения собранных новостей на странице агрегатора новостей. Для этого можно использовать язык программирования PHP и технологии AJAX и jQuery.

Таким образом, в данной работе был рассмотрен процесс разработки парсера новостных сайтов для агрегатора новостей с использованием языка программирования PHP и базы данных MySQL. Создание такого парсера может значительно упростить процесс получения новостей с различных источников

2.2.2 Проектирование базы данных

Проектирование базы данных является важным шагом в разработке агрегатора новостей. В базе данных будет храниться информация о новостях, включая заголовки, описания, ссылки, даты публикации и другие данные, необходимые для управления новостями.

Одним из способов получения новостей является использование формата RSS (Really Simple Syndication). RSS - это формат обмена данными, позволяющий сайтам предоставлять свои новости в виде XML-файлов. Каждый файл RSS содержит множество новостей в формате, удобном для парсинга.

База данных будет использоваться в веб-приложении, которое будет позволять пользователям просматривать новости, а также регистрироваться и авторизовываться для доступа к дополнительным функциям.

Для создания данной базы данных были определены основные сущности и их атрибуты, которые будут храниться в базе данных. Основные сущности - это новости и пользователи. Для каждой из них была создана отдельная таблица с соответствующими атрибутами.

Данная база данных позволит хранить большое количество информации о новостях и пользователях, а также будет обеспечивать удобный и быстрый доступ к этой информации. Она будет поддерживать автоматическое обновление новостей.

Для создания базы данных использовался язык SQL и система управления базами данных MySQL. Кроме того, были использованы принципы нормализации баз данных для обеспечения эффективной работы с базой данных.

Таблица "news" будет содержать следующие поля:

"id" - уникальный идентификатор новости типа целое число (INT) с длиной 6 символов (6) и без знака (UNSIGNED). Поле будет автоматически увеличиваться (AUTO_INCREMENT) для каждой новой записи. Оно будет являться первичным ключом (PRIMARY KEY) таблицы.

"title" - заголовок новости типа переменная длины (VARCHAR) с максимальной длиной 255 символов (255). Поле не может быть пустым (NOT NULL).

"description" - описание новости типа текст (TEXT). Поле не может быть пустым (NOT NULL).

"link" - ссылка на новость типа VARCHAR с максимальной длиной 255 символов (255). Поле не может быть пустым (NOT NULL).

"pubDate" - дата и время публикации новости типа дата/время (DATETIME). Поле может быть пустым.

"ourNews" – флаг, указывающий на собственные новости. Поле может быть пустым

Таблица "users" будет содержать следующие поля:

"id" - уникальный идентификатор пользователя типа целое число (INT) с длиной 6 символов (6) и без знака (UNSIGNED). Поле будет автоматически увеличиваться (AUTO_INCREMENT) для каждой новой записи. Оно будет являться первичным ключом (PRIMARY KEY) таблицы.

"username" - имя пользователя типа VARCHAR с максимальной длиной 255 символов (255). Поле не может быть пустым (NOT NULL).

"password" - хэш пароля типа VARCHAR с максимальной длиной 255 символов (255). Поле не может быть пустым (NOT NULL).

Таким образом, эти две таблицы будут представлять собой основу базы данных, которая будет хранить информацию об источниках новостей, пользователях и самих новостях.

Также будет создана таблица Black_List полностью повторяющая структуру сущности News для хранения удаленных новостей и использоваться в качестве проверочных данных для парсинга.

2.2.3 Разработка парсера новостных сайтов

Разработка парсера новостных сайтов - это процесс создания программного обеспечения, которое собирает и анализирует данные новостных сайтов, а затем сохраняет их в базу данных или использует для создания агрегатора новостей.

Для создания парсера необходимо определить список новостных сайтов, с которых нужно получать данные, их структуру и форматирование. Кроме того, необходимо определить, какие данные необходимо сохранить (например, заголовки, описания, даты, авторы и т. д.) и какие инструменты будут использоваться для анализа и сохранения этих данных.

В процессе разработки парсера можно использовать различные языки программирования, библиотеки и инструменты. Одним из популярных языков программирования для создания парсеров является Python, а также существует множество библиотек и инструментов для работы с HTML и XML-страницами, таких как Beautiful Soup, lxml, Requests и другие.

При разработке парсера важно учитывать юридические и этические аспекты, такие как авторские права и политику конфиденциальности. Некоторые новостные сайты могут запрещать использование их данных без разрешения, поэтому необходимо быть внимательным и следовать правилам использования.

Одним из примеров парсера новостных сайтов может быть парсер, который собирает данные с нескольких новостных сайтов, фильтрует их по ключевым словам и сохраняет в базу данных для использования в агрегаторе новостей.

После проектирования базы данных и выбора новостных сайтов для парсинга, начинается разработка парсера. Парсер является программным модулем, который загружает данные с веб-страницы и извлекает из них необходимую информацию.

Для разработки парсера можно использовать различные языки программирования, такие как Python, PHP, Java и другие. В данном случае мы будем использовать язык PHP.

Разработка парсера будет проходить в несколько этапов:

1. Загрузка веб-страницы.
2. Извлечение информации из страницы.
3. Сохранение информации в базу данных.

Для загрузки веб-страницы используется функция `file_get_contents()`, которая позволяет получить содержимое веб-страницы в виде строки. Например:

```
$url = 'https://example.com/'; $html = file_get_contents($url);
```

Далее, для извлечения информации из страницы используются регулярные выражения или библиотеки для парсинга HTML-кода, такие как Simple HTML DOM Parser. Регулярные выражения позволяют находить и извлекать информацию из HTML-кода, используя шаблоны поиска.

Пример извлечения заголовка новости и ее ссылки с помощью регулярных выражений:

```
preg_match('/<h2  
class="entry-title"><a href="(.*?)">(.*?)</a></h2>/',  
$html, $matches);  
$title = $matches[2];  
$link = $matches [1];
```

Для более сложных случаев, когда необходимо извлекать информацию из множества страниц, можно использовать библиотеку Simple HTML DOM Parser. Она позволяет использовать CSS-селекторы для поиска и извлечения информации из HTML-кода. Пример:

```

require_once('simple_html_dom.php');
$url = 'https://example.com/';
$html = file_get_html($url); // Находим все заголовки новостей
foreach ($html->find('.entry-title') as $entry) {
    $title = $entry->plaintext;
    $link = $entry->find('a', 0)->href;
    // Сохраняем информацию в базу данных
    $sql = "INSERT INTO news (title, link) VALUES ('$title', '$link')";
    $result = $db->query($sql); }

```

Кроме того, парсер также осуществляет очистку текста от ненужных символов, форматирование и сохранение данных в базу данных. В результате парсера, агрегатор получает актуальные новости от различных источников, что позволяет пользователю быстро и удобно получать доступ к новостным материалам.

В следующей части курсовой работы будет описан процесс создания парсера новостных сайтов, включая использованные технологии и инструменты, структуру и логику работы, а также примеры кода.

2.2.4 Разработка агрегатора новостей на основе парсера

Разработка агрегатора новостей на основе парсера предполагает создание системы, которая будет автоматически собирать новости с различных источников и предоставлять их пользователю в удобном виде.

Одним из ключевых компонентов такой системы является парсер, который будет извлекать данные из различных источников новостей. Для этого можно использовать библиотеки для парсинга HTML-кода, такие как Simple HTML DOM, Goutte или Beautiful Soup.

После того, как данные извлечены, они могут быть сохранены в базу данных, чтобы обеспечить быстрый доступ к ним. Затем, можно разработать интерфейс для пользователя, который позволит просматривать новости, фильтровать их по категориям или ключевым словам, и т.д.

Для улучшения качества агрегации новостей можно использовать алгоритмы машинного обучения, которые будут анализировать новости и на основе этого предлагать пользователю наиболее релевантные новости.

В целом, разработка агрегатора новостей на основе парсера представляет собой интересный и актуальный проект, который может быть использован для автоматизации процесса сбора и предоставления новостей различным пользователям.

Для создания базы данных MySQL необходимо выполнить несколько простых шагов:

1. Установить MySQL на компьютер, если его еще нет.
2. Запустить MySQL и зайти в консольную оболочку. Это можно сделать, например, через командную строку или терминал, введя команду `mysql -u username -p`, где `username` - имя пользователя, имеющего права на создание базы данных.

3. Создать базу данных. Для этого нужно ввести команду `CREATE DATABASE databasename`, где `databasename` - название базы данных, которую нужно создать. Например, `CREATE DATABASE mynewsdb`.

4. Создать таблицы в базе данных. Для создания таблиц необходимо использовать команду `CREATE TABLE tablename (column1 datatype, column2 datatype, column3 datatype, ...)` и указать название таблицы и ее столбцы (поля) с указанием типа данных для каждого поля.

5. Например, `CREATE TABLE news (id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY, title VARCHAR(255) NOT NULL, description TEXT NOT NULL, link VARCHAR(255) NOT NULL, pubDate DATETIME)`.

6. Настроить таблицы базы данных, добавив индексы, ограничения и т.д. Это можно сделать, используя команды `ALTER TABLE`.

7. Заполнить таблицы базы данных. Для этого можно использовать команду `INSERT INTO tablename (column1, column2, column3, ...) VALUES (value1, value2, value3, ...)`, где `tablename` - название таблицы, `column1`, `column2`,

column3, ... - названия столбцов, а value1, value2, value3, ... - значения для каждого столбца.

Таким образом, создание базы данных на MySQL не является сложной задачей, но требует знания SQL и умения работать с консольной оболочкой.

В рамках разработки агрегатора новостей на основе парсера необходимо разработать следующие компоненты:

1. Парсеры для каждого новостного сайта, которые извлекают информацию о новостях из HTML-страниц.
2. Скрипты для сохранения информации о новостях в базу данных MySQL.
3. Скрипты для выборки и отображения новостей на сайте-агрегаторе.
4. Дизайн сайта-агрегатора новостей.
5. Также для удобства разработки и тестирования следует подготовить окружение разработки на локальном компьютере или виртуальной машине, установив на нее веб-сервер, базу данных MySQL и необходимые библиотеки и зависимости для работы скриптов.

2.2.4.1 Загрузка новостей в базу данных

Код в файле index.php отвечает за постоянный мониторинг новостных RSS-лент и добавление новостей в базу данных.

Первым шагом задаётся переменная \$n, которая определяет время ожидания между проверками лент (в минутах). Далее происходит подключение к базе данных, описанной в файле config.php.

Затем начинается бесконечный цикл, который считывает URL-адреса RSS-лент из файла url.txt и начинает проходить по каждой из них.

Для каждой ленты происходит загрузка её содержимого и парсинг XML-документа с помощью функции simplexml_load_string.

Затем в цикле `foreach` обрабатываются все новостные элементы (теги `<item>`) и извлекаются нужные данные, такие как заголовок, описание, ссылка и дата публикации.

Эти данные затем используются для формирования SQL-запроса, который добавляет новость в базу данных с помощью функции `mysqli_query`.

Если при парсинге происходят ошибки (например, отсутствует какое-то поле в XML-документе), то они игнорируются блоком `try-catch`.

После прохода всех лент выполняется команда `sleep`, которая заставляет скрипт ждать указанное количество минут перед следующей проверкой лент. Таким образом, приложение будет работать непрерывно, мониторя новости и добавляя их в базу данных.

2.2.4.2 Отображение новостей

За отображение новостей отвечает файл `FirstPage.php`.

Код на языке PHP представляет собой скрипт для отображения новостей на веб-странице.

Первым шагом в коде выполняется проверка, авторизован ли пользователь на сайте. Затем определяется текущая страница, на которой отображаются новости. Если была отправлена форма поиска, то выполняется SQL запрос для выборки новостей, содержащих указанный в форме запрос. Если форма не была отправлена, то выполняется SQL запрос для выборки 10 последних новостей на текущей странице.

Результат SQL запроса выводится в цикле `while` с помощью функции `fetch_assoc()`, которая получает очередную строку результата запроса в виде ассоциативного массива. Для каждой новости на странице отображается ее заголовок, краткое содержание и кнопка для удаления новости (если пользователь авторизован).

Если запрос не дал результатов, то выводится сообщение о том, что новостей для отображения нет.

Если форма поиска была отправлена, то выводится кнопка для возврата к последним новостям на сайте. Если же форма поиска не была отправлена, то в зависимости от номера текущей страницы выводятся ссылки на предыдущую и следующую страницы (если они есть) или ссылка для возврата к последним новостям на сайте (если текущая страница - вторая).

Главная страница продемонстрирована на рисунке 19.

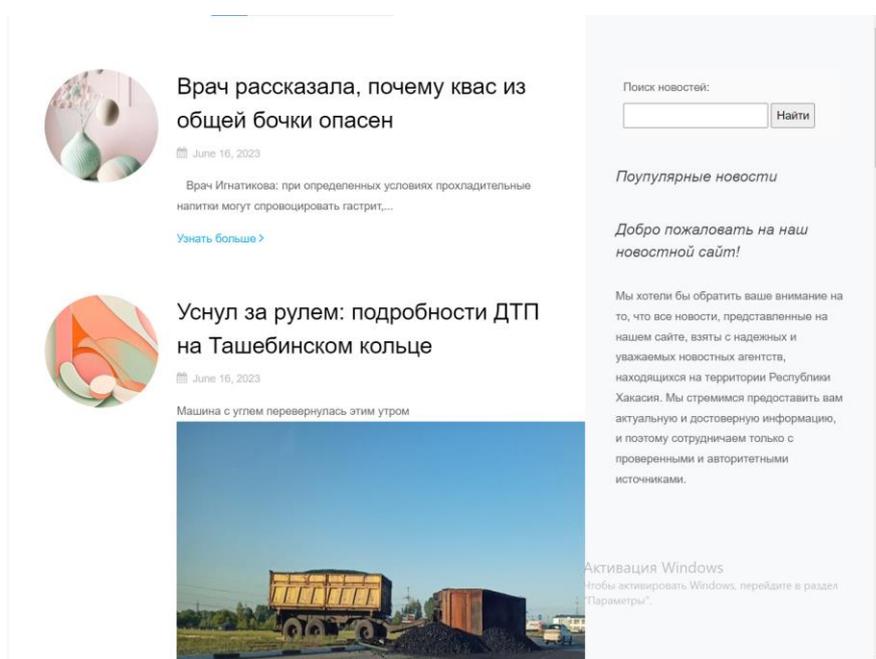


Рисунок 19 – Заглавная страница

2.2.4.3 Авторизация пользователя

Код файла login.php представляет собой скрипт для авторизации пользователей на веб-сайте. Сначала происходит проверка, был ли пользователь уже авторизован, и если да, то происходит перенаправление на главную страницу. Затем происходит проверка наличия данных из формы входа, и если они были введены, то происходит соединение с базой данных и выполнение SQL-запроса для выборки пользователя из базы данных. Если пользователь

найден, то имя пользователя сохраняется в сессии, и происходит перенаправление на главную страницу. Если пользователь не найден, то выводится сообщение об ошибке.

Форма входа содержит поля для ввода имени пользователя и пароля, а также кнопку "Войти", при нажатии на которую отправляются данные формы на сервер для обработки.

2.2.4.5 Аутентификация пользователя

Данный код представляет собой скрипт, который отвечает за аутентификацию пользователя. Он закрывает текущую сессию пользователя и перенаправляет его на страницу входа.

Первые три строки удаляют все переменные сессии и уничтожают сессию, которая была создана при авторизации пользователя. Затем происходит перенаправление пользователя на страницу входа с помощью функции `header()`, которая отправляет HTTP-заголовок для перенаправления на указанную страницу.

Функция `exit` останавливает выполнение скрипта после перенаправления, чтобы избежать возможных ошибок и дополнительной обработки запросов.

2.2.4.6 Создание RSS версии

Код в файле `RSS.php` PHP создает RSS-ленту, которая содержит последние 10 новостей, опубликованных на сайте.

Код начинается с подключения к базе данных, используя файл конфигурации `"config.php"`. Затем выполняется SQL-запрос для выборки последних 10 новостей из таблицы `"news"` и результат сохраняется в переменной `$result`.

Затем создается XML-документ в формате RSS, начиная с элемента `"rss"`. Заголовок и описание ленты определяются в соответствии с переменными `$title`

и \$description, которые могут быть изменены на любые другие значения. Далее, элементы "channel" содержат информацию о ленте, такую как название, описание, ссылку и язык.

Затем в цикле while для каждой новости создается элемент "item". Заголовок, описание, ссылка и дата публикации каждой новости сохраняются в соответствующих переменных. Все значения обрабатываются функцией htmlspecialchars() для предотвращения возможных проблем с безопасностью. Наконец, элементы "item" добавляются в XML-документ.

В конце кода закрываются теги "channel" и "rss".

2.2.4.7 Поиск новости

PHP код в файле search.php отвечает за поиск новостей на сайте по заданному поисковому запросу. Далее в тексте проведен разбор алгоритма данного файла.

Сначала в коде определяются параметры соединения с базой данных с помощью подключения файла config.php. Затем проверяется, был ли выполнен поисковый запрос, используя функцию isset() для проверки существования переменной \$_GET['q'] и функцию empty() для проверки того, что переменная не пуста.

Если запрос был выполнен, то заданный поисковый запрос сохраняется в переменную \$search_query. Далее выполняется SQL-запрос к базе данных, который выбирает все новости, в заголовке которых содержится заданный поисковый запрос, с помощью оператора LIKE и фильтра %. Результаты запроса сохраняются в переменную \$result.

Если результаты поиска были найдены (число строк в \$result больше 0), то цикл while используется для вывода найденных новостей на экран. Каждая новость выводится в виде HTML-элементов <h2> и <p>, соответственно, для заголовка и описания новости. Для каждой новости также добавляется ссылка на страницу с полной новостью, указывающая на news_item.php?id= с id новости в

качестве параметра. Если пользователь авторизован на сайте, то для каждой новости выводится форма для удаления новости с помощью кнопки «Удалить».

Если результаты поиска не были найдены (число строк в $\$result$ равно 0), то выводится соответствующее сообщение. Если поисковый запрос не был выполнен (переменная $\$_GET['q']$ не существует или является пустой), то выводится сообщение "Введите поисковый запрос".

После того, как поиск завершен, соединение с базой данных закрывается с помощью метода $\$conn->close()$.

Поиск новости по запросу «Жара» показан на рисунке 20.

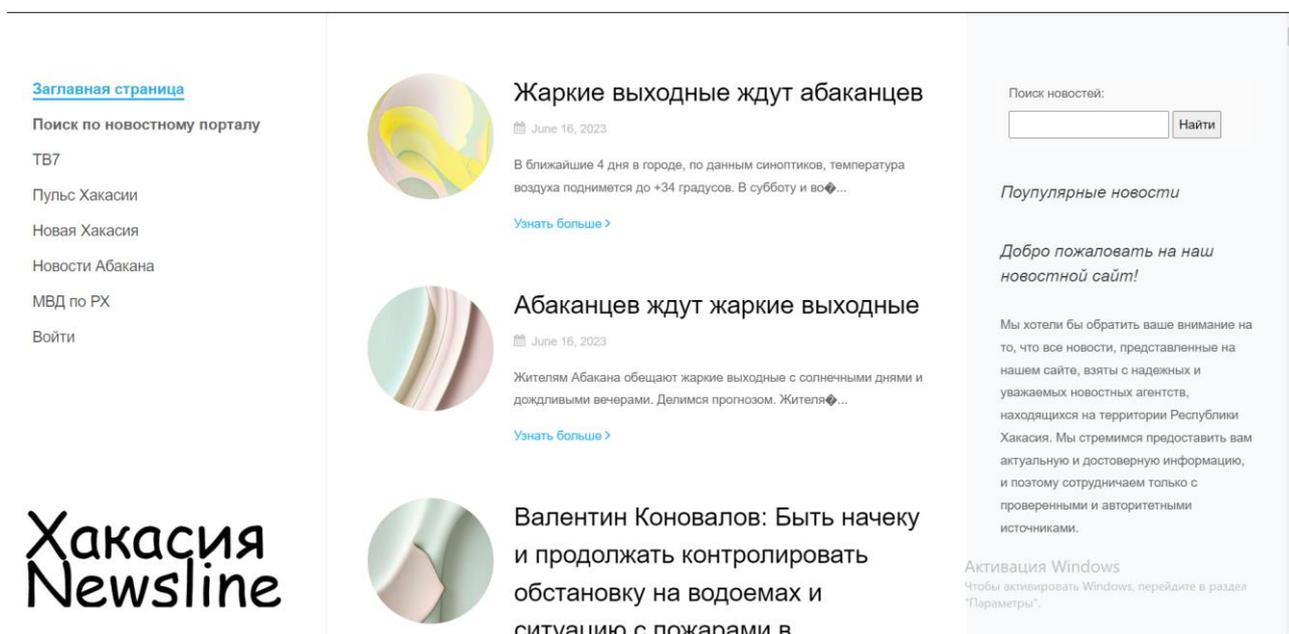


Рисунок 20 – Результат поиска

2.2.4.8 Удаление новости

Данный PHP-скрипт (delete.php) отвечает за удаление новости из базы данных. Он также использует параметры соединения с базой данных, указанные в файле config.php, и проверяет, был ли пользователь авторизован, прежде чем выполнить удаление.

Первым шагом скрипт начинает сессию PHP с помощью функции `session_start()`. Затем он подключает файл `config.php` с параметрами соединения с базой данных.

Далее скрипт проверяет, был ли пользователь авторизован, с помощью сессии PHP. Если пользователь авторизован, скрипт получает ID новости, которую нужно удалить, с помощью `$_POST['news_id']`. Этот ID передается из формы удаления новости в HTML-коде.

После получения ID новости, скрипт выполняет SQL-запрос для удаления новости из базы данных с помощью команды `DELETE`. SQL-запрос использует значение `$news_id` для определения, какую новость нужно удалить.

Если удаление прошло успешно, скрипт выводит сообщение "Новость успешно удалена" и перенаправляет пользователя на страницу входа с помощью функции `header("Location: login.php")`. Если же произошла ошибка удаления, скрипт выводит сообщение об ошибке.

В конце скрипт закрывает соединение с базой данных с помощью функции `$conn->close()`. Если пользователь не авторизован, скрипт перенаправляет его на страницу входа с помощью функции `header("Location: login.php")`.

2.2.4.9 Содержание новости

PHP скрипт `news_item.php` отвечает за отображение содержания новости на странице. Он использует параметр `id`, переданный в URL, для выборки соответствующей новости из базы данных и вывода ее содержимого на экран.

Сначала происходит проверка наличия параметра `id` в URL. Если он передан, то он используется для выполнения SQL запроса для выборки новости с соответствующим ID из базы данных. Результаты запроса сохраняются в переменной `$result`.

Затем проверяется, есть ли хотя бы одна строка в результате запроса. Если да, то это означает, что новость была найдена, и ее содержание выводится на экран.

Заголовок новости выводится в теге h2, а содержание новости – в теге p. Если пользователь уже авторизован, скрипт отображает кнопку удаления новости, которая отправляет POST запрос на другой скрипт, ответственный за удаление новости из базы данных.

Если параметр id отсутствует в URL или результат запроса не содержит ни одной строки, скрипт выводит сообщение о том, что новость не найдена.

В конце скрипт закрывает соединение с базой данных и завершает свою работу.

Пример содержания новости показан на рисунке 21.

[Заглавная страница](#)

Поиск по новостному portalу

ТВ7

Пульс Хакасии

Новая Хакасия

Новости Абакана

МВД по РХ

Войти

Жаркие выходные ждут абаканцев

В ближайшие 4 дня в городе, по данным синоптиков, температура воздуха поднимется до +34 градусов. В субботу и воскресенье в республиканской столице +28, +29. К понедельнику столбик термометра поднимется еще на 2 градуса. Самым жарким на неделе станет вторник. А вот со среды в Абакане прогнозируются ливни... они вновь принесут прохладу.

В ближайшие 4 дня в городе, по данным синоптиков, температура воздуха поднимется до +34 градусов. В субботу и воскресенье в республиканской столице +28, +29. К понедельнику столбик термометра поднимется еще на 2 градуса. Самым жарким на неделе станет вторник. А вот со среды в Абакане прогнозируются ливни... они вновь принесут прохладу.

Рисунок 21 – Содержание новости

2.2.4.10 Администрирование

Код admin.php представляет собой файл PHP, который отвечает за добавление новостей и пользователей на сайт.

Первым шагом в коде начинается сессия PHP с помощью функции session_start(). Затем, в файле подключается файл конфигурации config.php, который содержит данные для подключения к базе данных.

Далее следует проверка, был ли пользователь авторизован. Если пользователь не авторизован, то он перенаправляется на страницу входа, используя функцию `header("Location: login.php")`.

Далее в коде содержатся два блока для обработки формы добавления новости и формы добавления пользователя соответственно. Обработка формы производится с помощью метода POST.

При обработке формы добавления новости, из формы извлекаются данные заголовка новости, описания новости и загруженного файла. Файл загружается на сервер с помощью функции `move_uploaded_file()`, которая помещает загруженный файл в указанную папку.

Затем данные новости добавляются в базу данных MySQL с помощью запроса `INSERT INTO`, который вставляет новую строку в таблицу `news`.

При обработке формы добавления пользователя из формы извлекаются имя пользователя, пароль и подтверждение пароля. Далее происходит проверка на заполненность всех полей формы и соответствие пароля и подтверждения пароля.

Если проверки прошли успешно, то пароль шифруется с помощью функции `password_hash()` и данные пользователя добавляются в базу данных MySQL с помощью запроса `INSERT INTO`.

Наконец, соединение с базой данных закрывается с помощью метода `close()`.

В конце файла находится HTML-код, который отображает формы добавления новостей и пользователей. Форма добавления новостей содержит поля для ввода заголовка, описания и загрузки файла, а форма добавления пользователей содержит поля для ввода имени пользователя и пароля.

Обе формы отправляют данные на этот же файл для обработки данных и добавления их в базу данных.

Окно администратора продемонстрировано на рисунке 22

[Заглавная страница](#)

[Поиск по новостному portalу](#)

[ТВ7](#)

[Пuls Хакасии](#)

[Новая Хакасия](#)

[Новости Абакана](#)

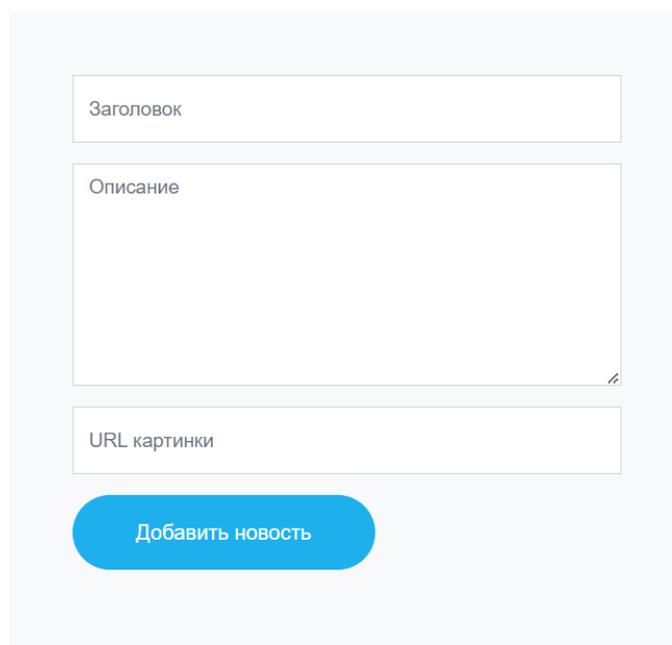
[МВД по РХ](#)

[Кабинет администратора](#)

[Выйти](#)

Хакасия Newslіne

Contact Information



Заголовок

Описание

URL картинки

Добавить новость

Рисунок 22 – Форма добавления новости

2.2.4.11 Конфигурация подключения

Код файла `config.php` представляет собой конфигурацию подключения к базе данных MySQL. Он содержит следующие переменные:

1. `$host`: имя хоста, на котором запущен сервер базы данных.
2. `$username`: имя пользователя базы данных, с правами на подключение и выполнение запросов.
3. `$password`: пароль пользователя базы данных.
4. `$dbname`: имя базы данных, к которой происходит подключение.

Далее, функция `mysqli_connect()` используется для установления соединения с базой данных, передавая ей в качестве параметров переменные `$host`, `$username`, `$password` и `$dbname`.

Если соединение не установлено (например, из-за неправильных учетных данных), функция `mysqli_connect_error()` вызывается для вывода сообщения об ошибке подключения.

В данном случае, если подключение установлено успешно, объект соединения `$conn` будет использоваться в других скриптах для выполнения запросов к базе данных.

Все файлы `php` представлены в приложениях от А до И

2.2.5 Запуск сервера с помощью ХАМРР

Для отображения РНР страниц используется веб-сервер, который обрабатывает запросы от клиентских браузеров и отдает в ответ запрошенные РНР файлы. Обычно в качестве веб-сервера для РНР используется Apache, но могут использоваться и другие серверы, такие как Nginx или Microsoft IIS.

Для работы с РНР и веб-сервером в процессе разработки можно использовать локальный сервер, например, ХАМРР, который включает в себя все необходимые компоненты: веб-сервер Apache, СУБД MySQL и интерпретатор РНР.

При запросе РНР файла, сервер запускает интерпретатор РНР, который обрабатывает код в файле и генерирует динамическую веб-страницу, которая отдается обратно клиентскому браузеру в виде HTML, CSS, JavaScript и других ресурсов. В итоге браузер отображает содержимое страницы, которая была сгенерирована на сервере.

ХАМРР - это набор утилит для создания и запуска веб-сервера, который включает в себя Apache, MySQL, РНР и другие компоненты, необходимые для работы веб-приложений на локальном компьютере. Для запуска сервера с помощью ХАМРР необходимо выполнить следующие шаги:

1. Установить ХАМРР на компьютер. После установки запустите ХАМРР Control Panel.
2. Запустить Apache и MySQL, выбрав соответствующие опции в ХАМРР Control Panel.

3. После запуска серверов можно протестировать работу сервера, открыв веб-браузер и вводя адрес "http://localhost/" в адресной строке. Если все работает правильно, откроется стартовая страница XAMPP.

4. Далее необходимо разместить свой сайт в каталоге "htdocs", который находится в папке, где установлен XAMPP. Можно создать свой проект в этом каталоге или скопировать уже готовый проект.

5. После размещения проекта в папке "htdocs" можно открыть свой сайт в веб-браузере, введя адрес "http://localhost/название_папки_с_проектом" в адресной строке.

6. При работе с базами данных необходимо создать базу данных и пользователей с правами доступа к базе данных в phpMyAdmin, который можно запустить, выбрав соответствующую опцию в XAMPP Control Panel.

Это основные шаги для запуска сервера с помощью XAMPP. В дополнение к этому, можно настроить различные параметры сервера, такие как порты, на которых он слушает запросы, параметры PHP и другие параметры, в зависимости от проекта и требований.

Вывод по разделу «Описание разработки Агрегатора»

В рамках работы были рассмотрены такие аспекты, как создание сайта на языке PHP, подключение к базе данных MySQL, а также установка и настройка сервера XAMPP.

Были рассмотрены основные принципы разработки динамических веб-сайтов на PHP, такие как работа с переменными, условиями, циклами и функциями, а также особенности работы с базами данных.

Для создания сайта использовался стек технологий, включающий в себя языки HTML, CSS и JavaScript, а также PHP для создания динамических страниц. Были рассмотрены основные принципы работы с каждым из этих языков, а также методы их взаимодействия между собой.

В ходе проекта была создана база данных для хранения информации о пользователях и новостях, а также была написана программа на PHP для добавления новостей и пользователей в базу данных.

3 Оценка совокупной стоимости и рисков системы

3.1 Расчёт затрат на разработку проекта

Капитальные вложения, связанные с автоматизацией обработки информации, рассчитываются по формуле

$$K = K_n + K_p, \quad (1)$$

где K_n – капитальные вложения на проектирование, руб.;

K_p – капитальные вложения на реализацию проекта, руб.

Предпроизводственные затраты представляют собой единовременные расходы на разработку обеспечивающих или функциональных систем или элементов на всех этапах проектирования, а также затраты на их усовершенствование, т.е. на проведение обследования и обработку материалов исследования, разработку технического задания, разработку технического и рабочего проекта системы и ее опытного внедрения. Сюда включаются затраты на разработку алгоритмов и программ, стоимость разработок по привязке типовых проектных решений (ТПР) и пакетов прикладных программ (ППП) к конкретному объекту автоматизации. Суммарные затраты на проектирование системы и ее разработку и отладку на компьютере определяются по формуле

$$K_n = ((1 + W_d)(1 + W_c) + W_n) \sum_{i=1}^m Z_{oi} + C_M + M_e, \quad (2)$$

где m – количество работников, участвующих в разработке проекта;

Z_{oi} – затраты на основную заработную плату работника i -й категории, руб.;

W_d – коэффициент, учитывающий дополнительную заработную плату в долях к основной заработной плате ($W_d = 0,4$ и состоит из коэффициента отпускных, равного $0,1$, и районного коэффициента – $0,3$);

W_c – коэффициент, учитывающий отчисления на социальные нужды, в долях к сумме основной и дополнительной заработной платы разработчиков. В соответствии с Федеральным законом 379-ФЗ от 03.12.2011г, страховые взносы в Пенсионный фонд в долях единицы – $0,22$, страховые взносы в ФСС – $0,029$, страховые взносы в ФФОМС – $0,051$, страховые взносы на производственный травматизм – $0,002$. Итого: $W_c = 0,302$;

W_n – коэффициент, учитывающий накладные расходы организации, в долях к основной заработной плате разработчиков (принимается по фактическим данным, $W_n = 0,6$);

C_M – затраты на программное обеспечение;

M_v – затраты на использование машинного времени.

Затраты на основную заработную плату работника i -й категории рассчитывают по формуле

$$Z_{oi} = Z_{дни} t_i, \quad (3)$$

где $Z_{дни}$ – среднедневная заработная плата работника i -й категории, руб./дн.;

t_i – количество дней, отработанных работником i -й категории.

Для оплаты труда программиста будет использована тарифная оплата труда, соответствующая минимальной оплате труда (МРОТ) в Республике Хакасия. Срок работы программиста – 68 неполных рабочих дня. На работу над проектом программист будет выделять 3 часа в день. Тогда на работу будет затрачен 1 месяц. Расчет основной заработной платы разработчика проекта приведен в таблице 3.

Таблица 3 – Основная заработная плата разработчика

Должность	Должностной оклад, руб.	Средняя дневная ставка, руб.	Затраты на заработную плату разработчика, руб.
Программист	16242	738,27	33835

Ввиду того, что проектируемая информационная система должна быть запрограммирована и отлажена с помощью компьютеров, к суммарным затратам на разработку добавляются затраты на использование машинного времени, вычисляемые по формуле

$$M_B = t_{MB} S_{MЧ} K_M = 204 * 4,6 = 938 \text{ руб.} \quad (4)$$

где t_{MB} – машинное время компьютера, необходимое для разработки программного продукта; $t_{MB} = 204$ час..

Работа выполняется на ноутбуке на базе процессора Ryzen 3 3200U. Стоимость ноутбука составляет 45 000 руб.

Тогда стоимость 1 часа машинного времени:

$$S_{MЧ} = (45000/5)/247/8 = 4,6 \text{ руб./час.};$$

K_M – коэффициент мультипрограммности (показывает долю машинного времени, отводимого непосредственно на работу над проектом); $K_M = 1$.

Программное обеспечение, приобретенное в процессе выполнения работы, и их стоимость приведены в 3.4.

Таблица 4 – Перечень используемого программного обеспечения

Наименование ПО	Единица измерения	Требуемое количество	Цена за единицу, руб.	Сумма, руб.
Операционная система Windows 10	шт.	1	14990	14990
Figma	шт.	1	0	0
Google Chrome	шт.	1	0	0
Visual Studio Code	шт.	1	0	0
Итого				14990

Затраты на программное обеспечение:

$$C_m = 204 * (14990/8/247/8) = 194 \text{ руб.}$$

Смета затрат на разработку представлена в таблице 5.

Таблица 5 – Затраты на разработку

Статьи затрат	Сумма, руб.
Основная заработная плата	33835
Затраты на ПО	194
Затраты на машинное время	938
Прочие расходы	1049
Итого	36016

Диаграмма затрат на разработку также показана на рисунке 23

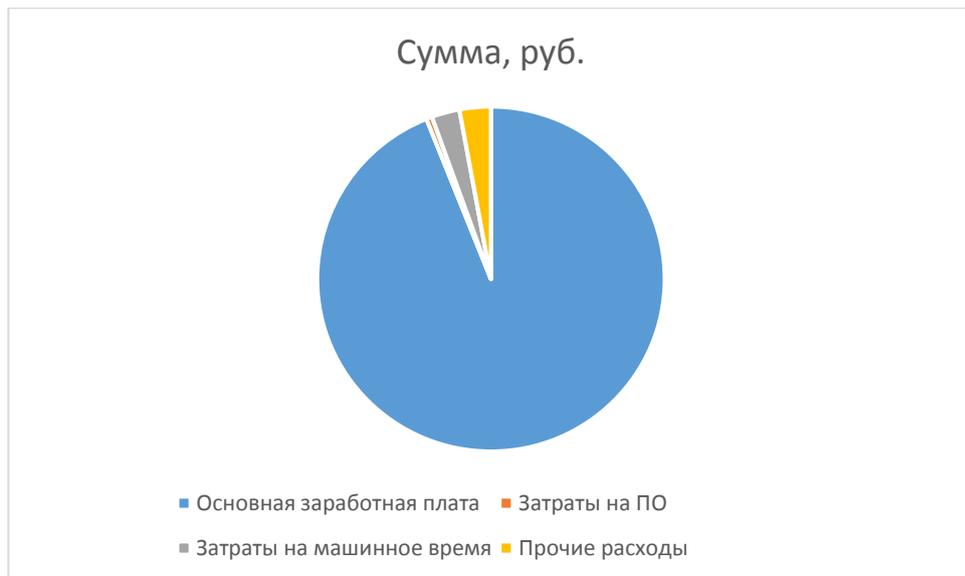


Рисунок 23 - Затраты на разработку

Капитальные вложения на реализацию проекта рассчитывают по формуле

$$K_p = K_o + K_{зд} + K_{пп} + K_{св} + K_{иб} + K_{пк}, \quad (5)$$

где K_o – затраты на основное и вспомогательное оборудование, руб.;

$K_{пп}$ – затраты на приобретение типовых разработок, пакетов, руб.;

$K_{св}$ – затраты на прокладку линий связи, руб.;

$K_{иб}$ – затраты на создание информационной базы данных, руб.;

$K_{пк}$ – затраты на подготовку и переподготовку кадров, руб.

В связи с тем, что для внедрения системы, рассматриваемой в данном проекте, не было затрат, связанных с прокладкой линии связи, затрат на основное и вспомогательное оборудование, то данные затраты для внедрения системы не учитывают. Также не принимаются в расчет затраты по подготовке и переподготовке кадров, затраты на создание информационной базы и затраты на приобретение типовых разработок.

Таким образом, при внедрении системы, рассматриваемой в данном проекте, затраты на его реализацию определяются затратами на оборудование и материалы. В оборудование входит ноутбук на базе процессора Ryzen 3 3200U. Стоимость ноутбука составляет 45 000 руб.

Амортизация на период внедрения и опытной эксплуатации (1 неделя) составит:

$$K_0 = 45\,000/5/52 = 173 \text{ рубля.}$$

Таким образом, суммарные затраты на разработку и внедрение веб-приложения составляют 36189 рублей.

3.2 Расчёт эксплуатационных затрат

К эксплуатационным (текущим) затратам относятся затраты, связанные с обеспечением нормального функционирования проекта. В перечень данных затрат входит ведение информационной базы, эксплуатация комплекса технических средств, эксплуатация систем программно-математического обеспечения, реализация технологического процесса обработки информации по задачам, а также эксплуатация системы в целом.

Текущие затраты рассчитываются по формуле

$$Z_{\text{тек}} = Z_{\text{зп}} + C_a + Z_э + C_{\text{рем}} + Z_m + C_{\text{проч}}, \quad (6)$$

где $Z_{\text{зп}}$ – затраты на заработную плату основную дополнительную с отчислениями во внебюджетные фонды, руб.;

C_a – амортизационные отчисления от стоимости оборудования и устройств системы, руб.;

$Z_э$ – затраты на силовую энергию, руб.;

$V_{\Sigma \text{р}}$ – затраты на текущий ремонт оборудования и устройств системы, руб.;

Z_m – затраты на материалы и машинные носители, руб.;

$C_{\text{проч}}$ – прочие расходы, руб.

Эксплуатацию разработанной системы осуществляют специалисты ЦУРа. Затраты на заработную плату основную и дополнительную с отчислениями на социальные нужды персонала рассчитываются по формуле

$$C_{\text{зп}} = \sum_{i=1}^m (t_i Z_i (1 + W_d)(1 + W_c)), \quad (7)$$

где t_i – время эксплуатации системы i -м работником, дни;

Z_i – среднедневная заработная плата i -го работника, руб./день.

В период эксплуатации с Агрегатором новостей будет работать сотрудник ЦУРа. Его работа будет заключаться в формировании и добавлении новостных статей. Данная работа занимает до 3 часов в неделю. Тогда затраты на заработную плату сотрудника в год составят:

$$C_{\text{зп}} = 29587 / 176 * 3 * 52 * 1,302 = 34\ 145 \text{ руб.}$$

Сумма амортизационных отчислений рассчитывается по формуле

$$C_a = \sum_{j=1}^n \frac{C_{bj} a_j g_j t_j}{F_{\text{эф}} j} \quad (8)$$

где C_{bj} – балансовая стоимость j -го вида оборудования, руб.;

t_j – время работы j -го вида оборудования, час;

$F_{\text{эф}} j$ – эффективный фонд времени работы оборудования в год, час;

a_j – норма годовых амортизационных отчислений для j -го вида оборудования;

g_j – количество единиц оборудования j -го вида.

Эффективный фонд времени работы оборудования можно вычислить по формуле

$$F_{\text{эф}} = D_p \times H_3, \quad (9)$$

где D_p – количество рабочих дней в году. $D_p = 247$ дней.

H_3 – норматив среднесуточной загрузки, час./день, $H_3 = 3$.

Таким образом, эффективный фонд времени работы оборудования для веб-приложения составляет 741 час.

Данные для расчёта:

$a_j = 0,2$ (используется ускоренная амортизация 20-30%);

$g_i = 1$;

$t_{j1,2} = 156$ час.;

$C_{b1} = 0$ руб. Планируется, что веб приложение будет работать на сервере ЦУРа, поэтому дополнительные затраты не требуются.

$C_{b2} = 45000$ руб. Для работы проекта необходим компьютер, на который будет установлена программа.

Таким образом, сумма амортизационных отчислений для веб приложения составит $C_{a1} = 45000 * 0,2 * \frac{156}{741} = 1895$ руб.

Затраты на текущий ремонт оборудования незначительные, так как работа выполняется на рабочем оборудовании ЦУРа.

Затраты на программное обеспечение, потребляемое в течение года, составляют 1% от балансовой стоимости основного оборудования. В данном случае их стоимость будет равна 0, так как для их эксплуатации необходима операционная система, которая была приобретена ранее и не имеет ограничений по сроку использования. Для веб-приложения также необходим браузер, который по умолчанию является бесплатным.

Прочие расходы составляют 3% от прямых расходов. Годовые эксплуатационные затраты предоставлены в таблице 6.

Таблица 6 – Годовые эксплуатационные затраты

Статьи затрат	Затраты на проект, руб.
Основная и дополнительная зарплата с отчислениями во внебюджетные фонды	34 145
Амортизационные отчисления	1895
Прочие расходы	1081
Итого	37 121

Также диаграмма показана эксплуатационных затрат на рисунке 24.

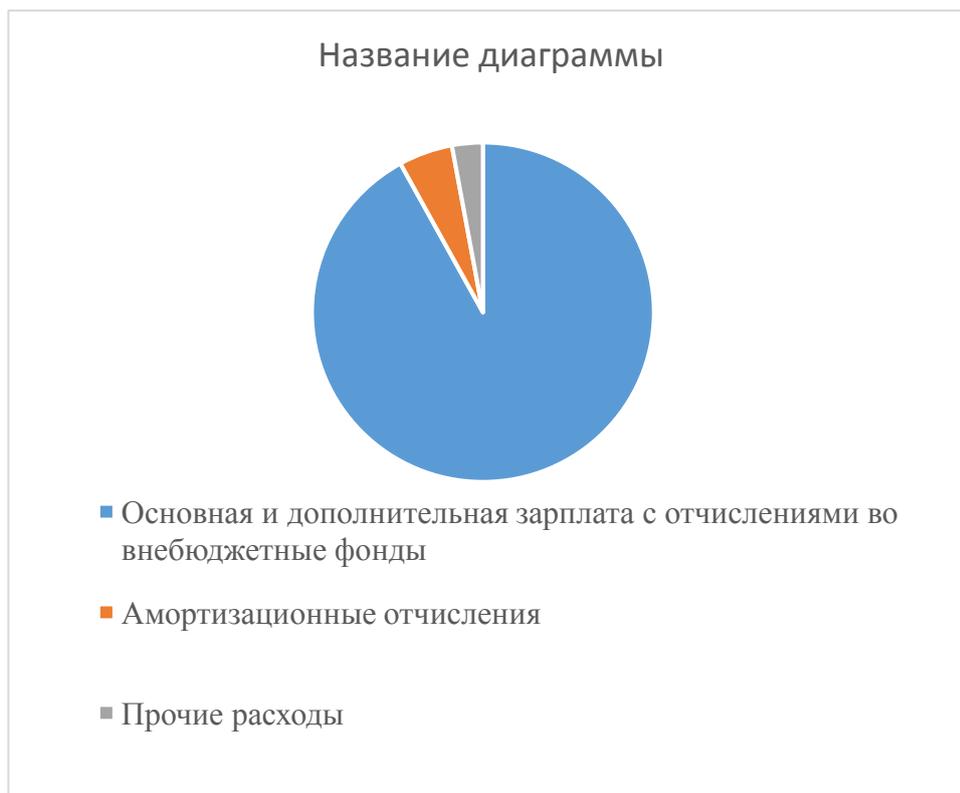


Рисунок 24 - Затраты на проект

3.3 Расчёт показателей экономического эффекта

Разрабатываемый продукт будет поставляться на некоммерческой основе, поэтому экономическая эффективность отсутствует. Программный продукт не предназначен для получения денежных доходов.

3.4 Маркетинговое сопровождение разрабатываемого продукта

Агрегатор новостей РХ разработан для ЦУР в соответствии с техническим заданием. Дальнейшему продвижению программный продукт не подлежит, так как является заказным продуктом. Коммерческому распространению данный проект не подлежит.

3.5 Расчет совокупной стоимости владения системой

Расчет ТСО проводится по формуле

$$TCO = DE + IC_1 + IC_2, \quad (10)$$

где DE – прямые расходы;

IC_{1,2} – косвенные расходы первой и второй групп соответственно.

IC_{1,2} – затраты на проведение обучения и заполнение информационной базы внесены в капитальные, других косвенных затрат нет.

$$IC_{1,2} = 0$$

Расчеты прямых затрат проводятся по формуле

$$DE = DE_1 + DE_2 + DE_3 + DE_4 + DE_5 + DE_6 + DE_7 + DE_8, \quad (11)$$

где DE₁ – капитальные затраты;

DE₂ – расходы на управление ИТ;

DE₃ – расходы на техническую поддержку АО и ПО;

DE₄ – расходы на разработку прикладного ПО внутренними силами;

DE₅ – расходы на аутсорсинг;

DE₆ – командировочные расходы;

DE₇ – расходы на услуги связи;

DE8 – другие группы расходов.

DE1 = 36 189 руб.

DE2 = 34 145 руб.

DE3 = 1895 руб.

DE4 = 0, так как проект не подразумевает расходы на разработку прикладного ПО внутренними силами.

DE5 = 0, так как передачи проекта аутсорсинговой компании не подразумевается. Проект поддерживается техническим специалистом.

DE6 = 0, так как нет необходимости отправлять сотрудников в командировку.

DE7 = 0, так как стоимость услуг связи уже рассчитаны в другой группе расходов и находятся в DE3 (Слс = 0 руб.).

DE8 = 1081 рубль.

DE = 73 310 руб.

ТСО = 73 310 руб.

3.6 Оценка рисков реализации проекта

Для предотвращения возможных проблем в процессе реализации проекта используется практика оценки рисков. Эта методика включает анализ потенциальных проблем, их вероятности возникновения, уровня влияния на проект, а также разработку стратегий решения. При грамотной оценке рисков возможные проблемы не будут неожиданными, так как будут заранее продуманы пути их решения.

1. Вероятность возникновения риска соответствия низкая, так как в начальной стадии проекта было проведено моделирование ИС, в рамках которого был разработан необходимый функционал и пользовательские сценарии взаимодействия. Тем не менее, необходимо учитывать возможность недопонимания с заказчиком, что может повлиять на риск соответствия.

2. Риск задержки сроков реализации проекта имеет средний уровень вероятности. Разработчик проекта оценивает сложность реализации и необходимый срок на этапе планирования. Тем не менее, разработка информационной системы – это сложный и многогранный процесс, который может столкнуться с неожиданными проблемами, поэтому необходимо предусмотреть дополнительное время на их решение.

3. Вероятность возникновения риска эксплуатации низкая, так как сотрудники проводят необходимые технические работы по мере необходимости.

4. Вероятность возникновения риска потери данных программного продукта средняя, так как система опирается на внешние источники данных.

В таблице 8 приведен анализ потенциальных рисков проекта.

Таблица 8 – Перечень рисков

Группа рисков	Перечень рисков проекта	Уровень влияния	Вероятность	Возможные варианты решения
Риск соответствия	Недостаточное взаимопонимание между заказчиком и разработчиком	Средний	Низкая	Консультация разработчика с заказчиком
Риск задержки сроков реализации проекта	Непредвиденные расходы, срыв сроков внедрения	Средний	Средняя	Заложить дополнительное время на решение непредвиденных технических проблем
Риск эксплуатации	Непредвиденный сбой работы сервера	Высокий	Низкая	Проведение технических работ по обслуживанию сервера
Риск потери данных проекта	Непредвиденная поломка компьютера, на котором выполняется процесс разработки	Низкий	Средняя	Подготовить среду разработки для резервного копирования проекта

Вывод по разделу «Оценка совокупной стоимости и рисков системы»

В данном разделе были выполнены расчеты затрат и проведена оценка рисков, связанных с реализацией проекта создания системы. Суммарные капитальные затраты на разработку ИС составили 36 189 рублей, что соответствует средней стоимости проектов данного уровня. Эксплуатационные затраты на использование ИС оценены в размере 37 121 рубля в год.

Агрегатор новостей РХ разработан в полном соответствии с техническим заданием выданным Центром Управления Региона Республики Хакасии. Программный продукт не будет коммерчески распространяться, поскольку является не является коммерческим.

ЗАКЛЮЧЕНИЕ

В рамках данной выпускной квалификационной работы (ВКР) был проведен анализ предметной области, описана и проанализирована основная деятельность Центра Управления Регионом республики Хакасия. Был проведен анализ новостных ресурсов и существующих решений на рынке.

Основным результатом работы является разработка Агрегатора новостей. В разделе "Описание разработки Агрегатора" были определены архитектура и средства разработки, а также описан процесс разработки. Был проведен анализ новостных сайтов, проектирование базы данных и разработка парсера новостных сайтов. Далее был разработан агрегатор новостей на основе парсера, включающий в себя функциональности загрузки новостей в базу данных, отображения новостей, авторизации и аутентификации пользователя, создания RSS версии, поиска новостей, удаления новостей, отображения содержания новости, администрирования и конфигурации подключения. Для запуска сервера использовался XAMPP.

Кроме того, была проведена оценка экономической эффективности разработки и внедрения информационной системы для профориентационного тестирования. Было обосновано целесообразность разработки проекта, произведен расчет затрат на разработку проекта, эксплуатационных затрат и показателей экономического эффекта. Также было предложено маркетинговое сопровождение разрабатываемого продукта.

В результате выполнения работы была создана информационная система в виде Агрегатора новостей, которая может быть полезна Центру Управления Регионом республики Хакасия для эффективного сбора и анализа новостной информации. Разработанный продукт имеет потенциал для использования в других организациях и сферах деятельности.

Таким образом, разработка и внедрение Агрегатора новостей представляют собой перспективное решение, способствующее повышению эффективности работы Центра Управления Регионом республики Хакасия и

других организаций. Результаты работы могут быть использованы в дальнейших исследованиях и разработках в области информационных систем и новостной аналитики.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аудитория средств массовой информации города Новосибирск / Новосибирск.ру : [сайт]. - 2019. - URL: <https://novosibirsk.ru/upload/press/research-smi-nsk-2019/smi-nsk-2019.pdf> (дата обращения: 10.10.22).
2. Федеральное агентство новостей / Фонд общественного мнения : [сайт]. - URL: <https://fom.ru/SMI-i-internet/14538> (дата обращения: 10.10.22).
3. За пять месяцев Центр управления регионом Хакасии обработал более 6 тысяч обращений / R-19.ru : [сайт]. - URL: <https://r-19.ru/news/politika/114417/> (дата обращения: 10.10.22).
4. О создании и функционировании Центра управления регионом Республики Хакасия [сайт] / R-19.ru. - URL: <https://r-19.ru/documents/140/105469.html> (дата обращения: 10.10.22).
5. RSS / Википедия : [сайт]. - URL: <https://ru.wikipedia.org/wiki/RSS> (дата обращения: 10.10.22).
6. HTML. В Wikipedia: свободной энциклопедии [сайт]. URL: <https://ru.wikipedia.org/wiki/HTML> (дата обращения: 10.10.22).
7. CSS. В Wikipedia: свободной энциклопедии [сайт]. URL: <https://ru.wikipedia.org/wiki/CSS> (дата обращения: 10.10.22).
8. JavaScript. В Wikipedia: свободной энциклопедии [сайт]. URL: <https://ru.wikipedia.org/wiki/JavaScript> (дата обращения: 10.10.22).
9. PHP. В Wikipedia: свободной энциклопедии [сайт]. URL: <https://ru.wikipedia.org/wiki/PHP> (дата обращения: 10.10.22).
10. MySQL. В Wikipedia: свободной энциклопедии [сайт]. URL: <https://ru.wikipedia.org/wiki/MySQL> (дата обращения: 10.10.22).

ПРИЛОЖЕНИЕ А

Программный код Index.php

```
<?php
$n=10;
require_once('config.php');

while (true) {
    // Read URLs from urls.txt
    $urls = file('C:\xampp\htdocs\url.txt', FILE_IGNORE_NEW_LINES);

    foreach ($urls as $url) {
        // Get contents of RSS feed
        $rss = file_get_contents($url);

        // Parse XML
        $xml = simplexml_load_string($rss);

        // Loop through news items
        foreach ($xml->channel->item as $item) {
            try {
                // Extract data
                $title = $item->title;
                $description = $item->description;
                $content = $item->children('content', true)->encoded;
                $content .= $item->children('yandex', true)->{'full-text'};
                $description = (string) $description . ' . (string)$content;

                $link = $item->link;
```

```
$pubDate = date('Y-m-d H:i:s', strtotime($item->pubDate));

// Insert data into database
$sql = "INSERT INTO news (title, description, link, pubDate) VALUES
('$title', '$description', '$link', '$pubDate)";
mysqli_query($conn, $sql);
} catch (\Throwable $th) {
    continue;
}
}
}

// Wait n minutes
sleep(60 * $n);
}
?>
```

ПРИЛОЖЕНИЕ Б

Программный код FirstPage.php

```
<?php
// Начать сессию PHP
session_start();

// Установить параметры соединения с базой данных
require_once('config.php');

// Проверить, был ли пользователь авторизован
$isLoggedIn = isset($_SESSION["username"]);

// Определить текущую страницу
$page = isset($_GET["page"]) ? $_GET["page"] : 1;

// Если форма поиска отправлена, выполнить поиск
if (isset($_POST["search"])) {
    $searchQuery = $_POST["searchQuery"];
    $sql = "SELECT * FROM news WHERE title LIKE '%$searchQuery%'
ORDER BY pubDate DESC";
} else {
    // Выполнить SQL запрос для выборки 10 новостей на текущей странице
    $start = ($page - 1) * 10;
    $sql = "SELECT * FROM news ORDER BY pubDate DESC LIMIT $start,
10";
}

$result = $conn->query($sql);
```

```

?>

<!DOCTYPE html>
<html>
<head>
  <title>Заголовок страницы</title>
</head>
<body>
  <h1>Новости Хакасии</h1>
  <?php if ($isLoggedIn) { ?>
    <p>Вы вошли как <?php echo $_SESSION["username"]; ?>. <a
href="logout.php">Выйти</a></p>
  <?php } else { ?>
    <p><a href="login.php">Войти</a></p>
  <?php } ?>

  <form method="post">
    <label for="searchQuery">Поиск новостей:</label>
    <input type="text" name="searchQuery" id="searchQuery">
    <input type="submit" name="search" value="Найти">
  </form>

  <?php if ($result->num_rows > 0) { ?>
    <?php while($row = $result->fetch_assoc()) { ?>
      <div>
        <h2><a href="news_item.php?id=<?php echo $row['id']; ?>"><?php
echo $row["title"]; ?></a></h2>
        <p><?php echo "<p>" . substr($row["description"], 0, 200) . "...</p>";
?></p>

```

```

        <?php if ($isLoggedIn) { ?>
            <form method="post" action="delete_news.php">
                <input type="hidden" name="news_id" value="<?php echo
$row["id"]; ?>">
                <input type="submit" value="Удалить">
            </form>
        <?php } ?>
    </div>
    <?php } ?>
<?php } else { ?>
    <p>Нет новостей для отображения</p>
<?php } ?>

<?php if (isset($_POST["search"])) { ?>
<p><a href="firstpage.php">Показать последние новости</a></p>
    <?php } else { ?>
        <?php if ($result->num_rows == 10) { ?>
            <?php if ($page >= 2) { ?>
                <p><a href="firstpage.php?page=<?php echo $page - 1;
?>">Предыдущие новости</a></p>
                <?php } ?>
                <p><a href="firstpage.php?page=<?php echo $page + 1;
?>">Следующие новости</a></p>
            <?php } elseif ($page == 2) { ?>
                <p><a href="firstpage.php">Показать последние новости</a></p>
            <?php } ?>
        <?php } ?>

```

ПРИЛОЖЕНИЕ В

Программный код Login.php

```
<?php
// Начать сессию PHP
session_start();

// Проверить, был ли пользователь уже авторизован
if (isset($_SESSION["username"])) {
    // Пользователь уже авторизован, перенаправить на главную страницу
    header("Location: firstpage.php");
    exit;
}

// Проверить, были ли введены данные из формы входа
if (isset($_POST["username"]) && isset($_POST["password"])) {
    // Установить параметры соединения с базой данных
    require_once('config.php');

    // Очистить данные из формы входа
    $username = mysqli_real_escape_string($conn, $_POST["username"]);
    $password = mysqli_real_escape_string($conn, $_POST["password"]);

    // Выполнить SQL запрос для выборки пользователя из базы данных
    $sql = "SELECT * FROM users WHERE username='$username' AND
password='$password'";
    $result = $conn->query($sql);
```

```

// Проверить, был ли найден пользователь
if ($result->num_rows == 1) {
    // Пользователь найден, сохранить имя пользователя в сессии
    $_SESSION["username"] = $username;

    // Перенаправить на главную страницу
    header("Location: firstpage.php");
    exit;
} else {
    // Пользователь не найден, вывести сообщение об ошибке
    $error_message = "Неверное имя пользователя или пароль";
}

// Закрывать соединение с базой данных
$conn->close();
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Вход в систему</title>
</head>
<body>
    <h1>Вход в систему</h1>
    <?php
        if (isset($error_message)) {
            echo "<p style='color:red'>" . $error_message . "</p>";
        }
    ?>

```

```
<form method="POST" action="login.php">
  <label for="username">Имя пользователя:</label><br>
  <input type="text" id="username" name="username"><br>
  <label for="password">Пароль:</label><br>
  <input type="password" id="password" name="password"><br><br>
  <input type="submit" value="Войти">
</form>
</body>
</html>
```

ПРИЛОЖЕНИЕ Г

Программный код login.php

```
<?php
// Начать сессию PHP
session_start();

// Удалить все переменные сессии
session_unset();

// Удалить сессию
session_destroy();

// Перенаправить на страницу входа
header("Location: login.php");
exit;
?>
```

ПРИЛОЖЕНИЕ Д

Программный код RSS.php

```
<?php
// Подключение к базе данных
require_once('config.php');

// Выборка всех новостей
$sql = "SELECT * FROM news ORDER BY pubDate DESC LIMIT 10";
$result = $db->query($sql);

// Проверка на наличие новостей
if ($result->num_rows == 0) {
    die("Новостей нет");
}
?>

<?php
// Создание заголовка и описания ленты
$title = "Последние новости";
$description = "Это лента последних новостей с нашего сайта.";

// Отображение заголовка и описания ленты в XML формате
echo "<?xml version='1.0' encoding='UTF-8'?>
<rss version='2.0'>
<channel>
<title>{$title}</title>
<description>{$description}</description>
```

```

    <link>http://example.com</link>
    <language>ru-ru</language>
";
?>

<?php
// Создание заголовка и описания ленты
$title = "Последние новости";
$description = "Это лента последних новостей с нашего сайта.";

// Отображение заголовка и описания ленты в XML формате
echo "<?xml version='1.0' encoding='UTF-8'?>
<rss version='2.0'>
    <channel>
        <title>{$title}</title>
        <description>{$description}</description>
        <link>http://example.com</link>
        <language>ru-ru</language>
";
?>

<?php
// Создание элементов "item" для каждой новости
while ($row = $result->fetch_assoc()) {
    $title = htmlspecialchars($row["title"]);
    $description = htmlspecialchars($row["description"]);
    $link = "http://example.com/news.php?id={$row["id"]}";
    $pubDate = date("r", strtotime($row["pubDate"]));

    echo "<item>
        <title>{$title}</title>

```

```
<description>{$description}</description>
<link>{$link}</link>
<pubDate>{$pubDate}</pubDate>
</item>
";
}
?>
<?php
// ЗАКРЫТИЕ ТЕГОВ "channel" и "rss"
echo "</channel>
</rss>";
?>
```

ПРИЛОЖЕНИЕ Е

Программный код config.php

```
<?php
$host = "localhost";
$username = "root";
$password = "0000";
$dbname = "News";

$conn = mysqli_connect($host, $username, $password, $dbname);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
?>
```

ПРИЛОЖЕНИЕ Ж

Программный код DeleteNews.php

```
<?php
// Начать сессию PHP
session_start();

// Установить параметры соединения с базой данных
require_once('config.php');

// Проверить, был ли пользователь авторизован
if (isset($_SESSION["username"])) {
    // Получить ID новости, которую нужно удалить
    $news_id = $_POST['news_id'];

    // Выполнить SQL запрос для удаления новости с заданным ID
    $sql = "DELETE FROM news WHERE id = '$news_id'";
    if ($conn->query($sql) === TRUE) {
        echo "Новость успешно удалена";
        header("Location: login.php");
    } else {
        echo "Ошибка удаления новости: " . $conn->error;
    }

    // Закрыть соединение с базой данных
    $conn->close();
} else {
```

```
// Если пользователь не авторизован, перенаправить его на страницу  
входа  
header("Location: login.php");  
exit();}?>
```

ПРИЛОЖЕНИЕ И

Программный код Search.php

```
<?php
// Установить параметры соединения с базой данных
require_once('config.php');

// Создать соединение с базой данных

// Проверить, был ли выполнен поисковый запрос
if (isset($_GET['q']) && !empty($_GET['q'])) {
    $search_query = $_GET['q'];

    // Выполнить SQL запрос для выборки новостей по заданному
    поисковому запросу
    $sql = "SELECT * FROM news WHERE title LIKE '%$search_query%'
ORDER BY pubDate DESC";
    $result = $conn->query($sql);

    // Вывести результаты поиска на экран
    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            echo "<h2>" . $row["title"] . "</h2>";
            echo "<p>" . $row["description"] . "</p>";
            // Добавить ссылку на полную новость
            echo "<p><a href='news_item.php?id=" . $row["id"] . "'>Читать
далее</a></p>";
            // Добавить кнопку удаления записи, если пользователь авторизован
```

```

        if (isset($_SESSION["username"])) {
            echo '<form method="post"><input type="hidden" name="id"
value="" . $row["id"] . "><input type="submit" name="delete"
value="Удалить"></form>';
        }
    }
} else {
    echo "Нет результатов для отображения";
}
} else {
    echo "Введите поисковый запрос";
}

// Закрывать соединение с базой данных
$conn->close();
?>

```

ПРИЛОЖЕНИЕ К

Программный код NewsItem.php

```
<?php
// Начать сессию PHP
session_start();

// Установить параметры соединения с базой данных
require_once('config.php');

// Проверить, был ли передан параметр id новости
if (isset($_GET['id'])) {
    // Получить id новости из параметра GET
    $news_id = $_GET['id'];

    // Выполнить SQL запрос для выборки новости по её id
    $sql = "SELECT * FROM news WHERE id = $news_id";
    $result = $conn->query($sql);

    // Вывести информацию о новости на экран
    if ($result->num_rows > 0) {
        $row = $result->fetch_assoc();
        echo "<h2>" . $row["title"] . "</h2>";
        echo "<p>" . $row["description"] . "</p>";

        // Проверить, был ли пользователь авторизован
        if (isset($_SESSION["username"])) {
            // Пользователь уже авторизован, показать кнопку удаления
            echo "<form method='post' action='delete_news.php'>";
            echo "<input type='hidden' name='news_id' value='$news_id'>";
            echo "<input type='submit' value='Удалить'>";
            echo "</form>";
        }
    } else {
```

```
        echo "Новость не найдена";
    }

    // Закрыть соединение с базой данных
    $conn->close();
} else {
    echo "Неверный запрос";
}
?>
```

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

Отпечатано в одном экземпляре.

Библиография 5 наименований.

Один экземпляр сдан на кафедру.

« ____ » _____ 2023 г.

_____ ФИО

