

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

Институт Математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем

(код и наименование направления подготовки, специальности)

Мобильные и сетевые технологии

(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Реализация алгоритма информационного поиска для информационной системы «Строительство Куйбышевской ГЭС»

Студент

В.Н. Вытоптова

(И.О. Фамилия)

(личная подпись)

Руководитель

А.П. Тонких

(Ученая степень, звание, И.О. Фамилия)

Консультант

М.В. Дайнеко

(Ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Аннотация

Тема выпускной квалификационной работы: «Реализация алгоритма информационного поиска для информационной системы «Строительство Куйбышевской ГЭС»».

Целью бакалаврской работы является реализация алгоритма информационного поиска в разработанной информационной системе.

Актуальность бакалаврской работы заключается в реализации алгоритма информационного поиска в уникальной информационной системе, так как она разработана на уникальных архивных данных, которые хранились прежде в бумажном формате исключительно в рамках музея. Во введении раскрывается актуальность исследования и даётся краткое описание проделанной работы.

В первой главе рассматриваются уже существующие варианты информационных систем, благодаря анализу которых будут поставлены задачи на реализацию.

Вторая глава содержит описание будущей системы, её концептуальной и технической реализации, а также математическую постановку задачи по разработке алгоритма, который будет работать внутри системы.

В третьей главе описывается процесс разработки системы и взаимодействия алгоритма с ней, представлен результат работы.

Результатом работы является информационная система, которая не имеет аналогов. Система содержит уникальные архивные данные, удобную классификацию информации и комфортный поиск интересующих статей.

Abstract

The present graduation work is devoted to implementing an information search algorithm for *Kuibyshev hydro-electric power station construction* information system.

The purpose of the research is to implement the information search algorithm in the information system under investigation.

The relevance of the research consists in implementing the information search algorithm in the unique information system, as it is developed based on the unique archival data that used to be stored in a paper format exclusively in the museum.

The introduction reveals the relevance of the research and provides a brief description of the work done.

The first chapter considers the existing variants of the information systems, carries out their analysis and sets the tasks to be solved.

The second chapter describes the future system, its conceptual and technical implementation, as well as the mathematical problem statement of the algorithm that works inside the system.

The third chapter deals with the process of developing the system and the interaction of the algorithm with it. This chapter also presents the results obtained.

The result of the research is the information system that has no analogues. The system contains the unique archived data, convenient information classification and a convenient search for articles.

Содержание

1 Теоретическое обоснование задачи для разработки.....	7
1.1 Описание исследуемой задачи для реализации	7
1.2 Обзор реализованных информационных систем.....	8
1.3 Формирование требований к новой системе.....	14
2 Проектирование информационной системы «Строительство Куйбышевской ГЭС»	16
2.1 Выбор технологии моделирования информационной системы.....	16
2.2 Построение диаграммы вариантов использования системы.....	17
2.3 Разработка интерфейса	21
2.4 Описание алгоритма информационного поиска	25
3 Программная реализация информационной системы «Строительство Куйбышевской ГЭС»	28
3.1 Организация сервера.....	28
3.2. Реализация интерфейса.....	31
3.3 Подгрузка элементов	33
3.4 Реализация алгоритма информационного поиска	37
Заключение	41
Список используемой литературы	42

Введение

На сегодняшний день существует множество информационных ресурсов, благодаря которым можно почерпнуть интересующую информацию. Однако привычная бумажная литература стала уходить на второй план, так как значительно уступает интерактивным информационным системам из-за практичности.

Безусловно, привычный книжный формат литературы ещё долго будет оставаться актуальным, так как в таких носителях хранятся уникальные знания, которые можно не найти на просторах информационных ресурсов. Но тем не менее существует ряд недостатков, благодаря которым начали развиваться альтернативные способы хранения или предоставления информации.

Одним из таких фундаментальных основополагающих недостатков стала мобильность. Электронные информационные ресурсы позволяют познакомиться с действительно редкой и полезной информацией в любой точке мира, не выходя из дома. Любая интерактивная информационная система будет значительно практичнее, так как такая система не сможет потеряться, благодаря современным способам дублирования или архивации данных. Однако стоит отметить один из самых положительных аспектов – это современность и яркость. Благодаря различным способам привычный книжный текст можно представить очень интересно, тем самым заинтересовав современных школьников и студентов.

Такие ресурсы благодаря ярким и интересным интерфейсам намного чаще привлекают людей, нежели книги. Интерактивная информационная система позволяет напрямую взаимодействовать с ней, получать интересующие знания, оставляя пользователя по-настоящему заинтересованным. Благодаря использованию аналитически собранных данных современные решения могут выглядеть очень ярко, интересно и в то

же время просто, так как пользователем может являться абсолютно любой человек с различными навыками владения компьютером.

Таким образом, **актуальность** бакалаврской работы заключается в необходимости разработки уникальной системы, которая будет сочетать в себе уникальную структуру и удобную навигацию, которая будет поддерживать поиск на странице.

Объектом исследования бакалаврской работы являются алгоритмы информационного поиска и уже существующие информационные системы.

Предметом исследования бакалаврской работы являются алгоритмы ранжирования web-страниц, программная реализация информационной системы.

Для достижения поставленной цели нужно решить следующие задачи:

1. Проанализировать всю научно-методическую литературу, которую необходимо классифицировать.
2. Проанализировать уже существующие варианты реализации подобных информационных систем.
3. Выбрать метод реализации информационного поиска.
4. Протестировать выбранный алгоритм.
5. Оценить эффективность реализованного алгоритма.

Практическая значимость работы заключается в разработке информационной системы «Строительство Куйбышевской ГЭС» с поддержкой алгоритма информационного поиска.

1 Теоретическое обоснование задачи для разработки

1.1 Описание исследуемой задачи для реализации

Каждый день мы сталкиваемся с различными потоками информации в том или ином виде. Информация может храниться как на физическом, так и на интерактивном носителе. Первостепенная цель этих носителей – сохранить и передать информацию, с которой может познакомиться пользователь. Один из крупнейших плюсов, с которым мы взаимодействуем на сегодняшний день, это возможность сохранять ценную историческую, библиографическую литературу без страха её потери, факта изнашивания, благодаря современным информационным технологиям.

Разработка информационной системы достаточно остро встречающийся вопрос на заре современных технологий. Государственные учреждения, в частности школы, налоговые и больницы, как одни из самых необходимых организаций в жизнедеятельности общества, всё больше переходят на взаимодействие с различными информационными ресурсами. Так же нельзя не подчеркнуть, что все общеобразовательные учреждения на сегодняшний день крайне озабочены методом подачи знаний. Всё чаще используются интерактивные доски, интерактивные игры, благодаря которым студент или ученик может познакомиться с информацией в новом и интересном игровом формате.

Немаловажная ценность информационных систем в образовательных учреждениях – это возможность сохранить ценную историческую информацию. Огромное количество рукописей, фотографий, статей и книг ежегодно теряют свой подлинный вид, большинство из них невозможно восстановить. Такие редкие статьи и фото, чаще всего, существуют лишь в единственном экземпляре, поэтому используя современные информационные технологии, можно переносить редкие документы в цифровой формат, чтобы

не только продлить им существование, но и чтобы каждый желающий мог работать с ними.

Реализованное программное обеспечение, позволяющее продемонстрировать исторически важные документы, широко применяется во всём мире. Стоит отметить, что каждая информационная система имеет первостепенную задачу – это ответы на вопросы пользователя в интересующей сфере. Такое программное обеспечение имеют схожую структуру, но отличается по своей реализации, так как каждая такая система выполняет свою индивидуальную задачу и имеет свои специальные возможности.

Для данной работы необходимо изучить уже разработанное программное обеспечение, которое несёт информативный характер, классифицируя информацию и предоставляя удобный интерфейс с комфортной навигацией.

1.2 Обзор реализованных информационных систем

Для реализации обозначенной проблемы необходимо найти и проанализировать уже существующие информационные ресурсы, чтобы опираясь на совокупность различных интерфейсов, навигации, эргономики проектирования сделать ряд определённых выводов и приступить к разработке.

Цель данной работы – разработать комфортную информационную систему «Строительство Куйбышевской ГЭС», которая будет перенесена на интерактивный стол музея Тольяттинского государственного университета, чтобы впоследствии любой пользователь с любыми навыками смог легко ориентироваться в навигации и комфортно изучать интересующий материал.

«Астрономическое ПО stellarium» – это бесплатное общедоступное программное обеспечение, представляющее собой бесплатный планетарий с открытым исходным кодом. Оно показывает реалистичное небо в 3D,

аналогичное наблюдаемому невооружённым глазом, биноклем или телескопом. Пример интерфейса представлен на рисунке 1.

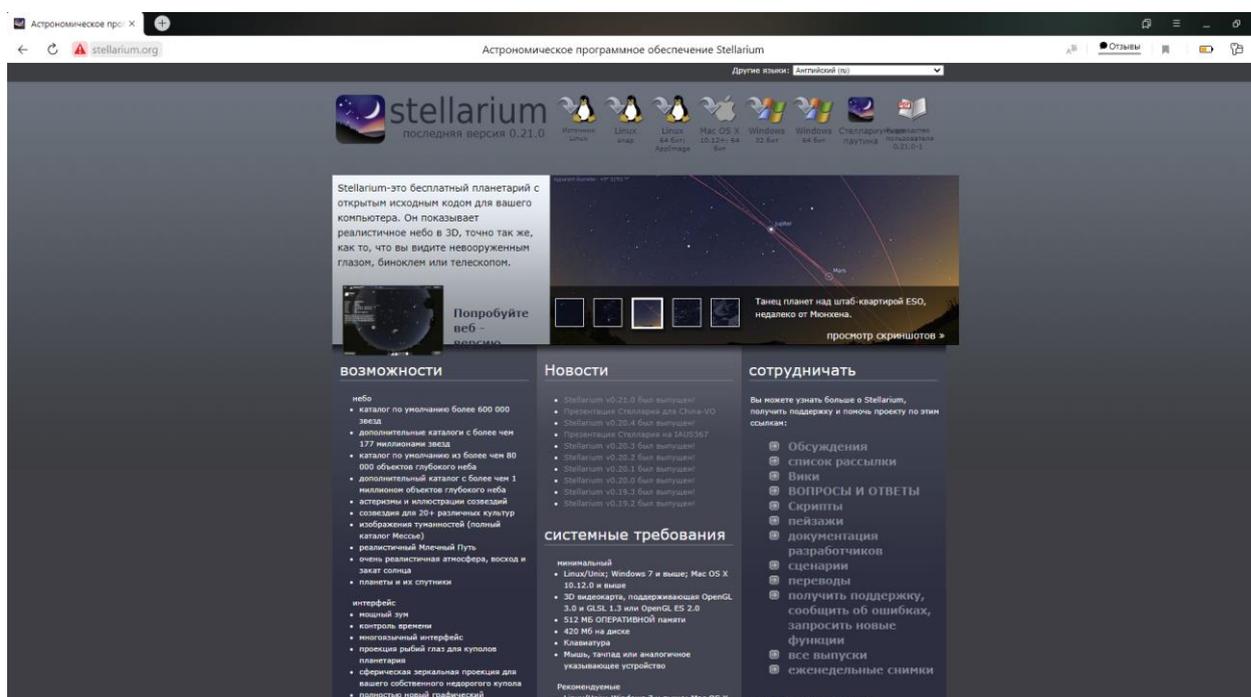


Рисунок 1 – Интерфейс ПО «stellarium»

Данная информационная система позволяет любому интересующемуся пользователю окунуться в мир астрономии, так же скачать звездную тему под любой тип операционной системы компьютера пользователя.

Стоит отметить, что ресурс бесплатный – сейчас это очень важный критерий, так как это значительно влияет на процент посещаемости, а также насколько человек будет заинтересован, потому что платные ресурсы отталкивают пользователей из-за обширных рекламных компаний и часто не оправдывают ожиданий и потраченных денег.

Stellarium является абсолютно бесплатным, но вызывает некоторое недоверие из-за своего простого интерфейса. Чаще всего такие web-сайты пользователи игнорируют, так как в настоящее время эргономика проектирования стремительно развивается, и пользователи привыкают к этому, а старые интерфейсы вызывают недоверие.

«Chem MSU» – это информационный ресурс, в котором размещены редкие учебные материалы по химии и нанотехнологии. Интерфейс представлен на рисунке 2.

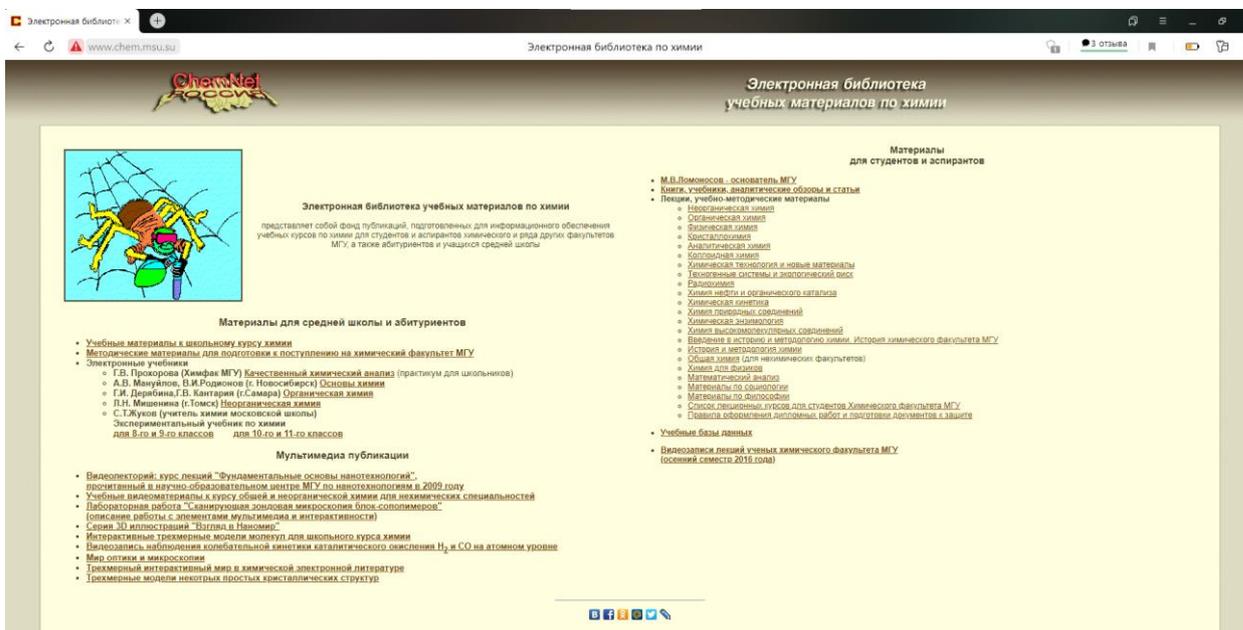


Рисунок 2 – Интерфейс ПО «Chem MSU»

Несмотря на то, что данный интерфейс является таким же простым, навигация здесь осуществляется намного удобнее. Также достоинством данной системы является общедоступность и наличие ценною информации.

Здесь сочетается совокупность лекционного материала преподавателей и аспирантов университета МГУ, доцентов и кандидатов наук, колоссальный набор информации по нанотехнологиям, наномиру, колебательной кинетике и другим физико-химическим отраслям.

«OnHistory» – обширный исторический информационный источник, включающий тематики войны, религии, науки, истории и даже катаклизмы. Обращает на себя внимание блок, содержащий наиболее популярные поисковые запросы. На сайте присутствует анимация. Интерфейс представлен на рисунке 3.



Рисунок 3 – Интерфейс ПО «OnHistory»

С точки зрения разработки это программное обеспечение реализовано немного сложнее. Здесь имеется анимация, а также активные ссылки, которые при наведении курсора меняют цвет. Однако в данной анимация в данном продукте крайне неудобная, поэтому при разработке необходимо учитывать эргономику использования. Интерфейс представлен на рисунке 4.



Рисунок 4 – Интерфейс программного обеспечения «OnHistory»

В целом, представленная информационная система имеет достаточно много достоинств, в том числе интересный интерфейс, комфортную навигацию, доступную и интересную информацию. В качестве недостатков можно указать слишком большое количество анимации, как активной, так и пассивной (пользователь может устать от этих всплывающих элементов), а также неудобное расположение подпунктов меню (пользователь может попасть не на желаемую страницу и в результате вовсе покинуть информационный ресурс).

Все вышеперечисленные источники являются информационными ресурсами с различной технической реализацией. При поиске информационных ресурсов по теме «Строительство Куйбышевской ГЭС» для сравнительного анализа, поисковая система не выдавала никаких подобных продуктов. Для сравнения рассмотрим один из информационных ресурсов, на котором хранится подобная информация.

Сайт, размещённый по адресу chronograph.livejournal.com, представляет собой исторический информационный ресурс, на котором хранится информация в формате фотоблога (рисунок 5).

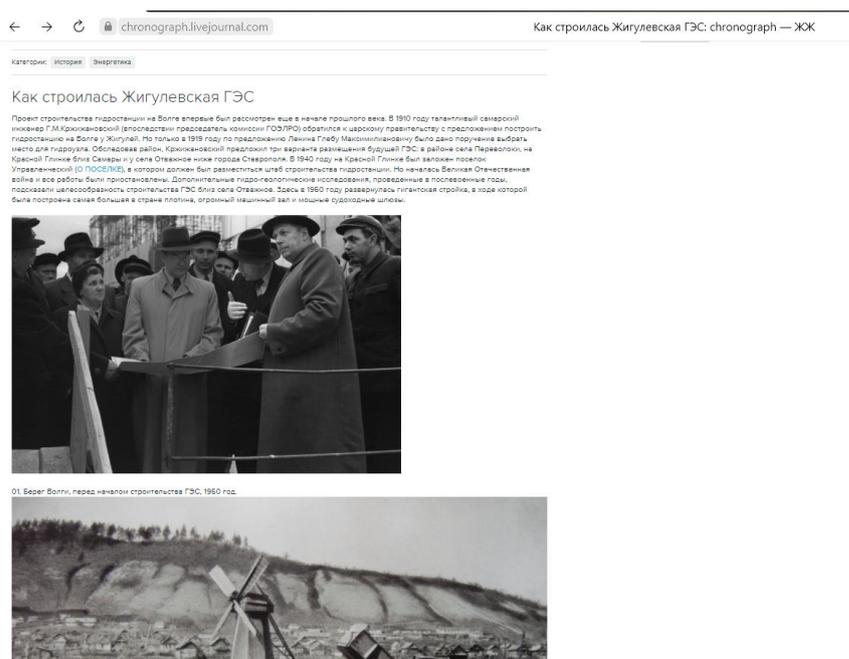


Рисунок 5 – Интерфейс фотоблога chronograph.livejournal.com

Блог представлен текстовым блоком с фотографиями с длинным скроллом. Стоит отметить отсутствие навязчивой рекламы, простоту интерфейса и структурированность, но в то же время присутствует чересчур большое количество картинок, которые нужно долго проматывать вниз, картинки часто не содержат подробных комментариев. Такой формат не очень хорошо воспринимается и может понравиться не каждому пользователю.

После анализа различных информационных систем можно сделать вывод, что единый метод разработки отсутствует. Какие-то ресурсы интегрированы с базами данных, другие имеют более простой интерфейс, но при этом хороший и информативный контент. В таблице 1 представлено сравнение ранее описанных информационных систем.

Таблица 1 – Сравнительная таблица перечисленных информационных систем

	Комфортная навигация	Информация о том, что содержит информационный ресурс	Научные материалы и статьи	Система поиска	Анимированные элементы
Stellarium	-	+	+	+	+
OnHistory	+	+	+/-	-	+
Chem MSU	+	+	+	-	-
Livejournal	-	-	+	-	-

На основе анализа рассмотренных информационных систем можно сделать вывод, что определённый шаблонный метод разработки программного обеспечения, результатом которого являлся бы идеальный интерфейс, отсутствует. Благодаря этому анализу были выделены несколько пунктов, которые должны обязательно учитываться при разработке:

- простой и комфортный интерфейс;
- удобная навигация по информационной системе;
- комфортная и понятная структуризация и классификация информации;
- представление уникальной информации о строительстве Куйбышевской ГЭС.

В результате подробного анализа различных методов реализации можно сделать вывод о необходимости разработки собственной информационной системы с удобной навигацией.

1.3 Формирование требований к новой системе

Задача, представленная к разработке – это создание уникальной информационной системы на музейных архивах, удобной для любого пользователя с абсолютно любыми навыками.

Благодаря данной системе будет решён ряд задач:

- так как информационный ресурс будет находиться в общем доступе, то он сможет знакомить потенциальных абитуриентов с интересующей информацией;
- музею получится структурировать и классифицировать всю информацию, которая хранится в архивах;
- данная разработка может стать платформой для дальнейшей модернизации проектов;
- данный интересный и непривычный формат подачи информации сможет привлечь внимание и интерес к истории своего региона.

Таким образом, стоит задача разработки новой системы, которая будет сочетать в себе современные технологические решения и доступ к уникальным архивным данным.

Выводы по главе

В данной главе были поставлены цели и задачи по реализации уникальной информационной системы. Был проведён сравнительный анализ, благодаря которому были разобраны альтернативные информационные системы, определены достоинства уже реализованного программного обеспечения для использования в проектируемой системе.

Произведённый анализ позволил сделать вывод о том, что какого-то единого подхода к разработке нет, так как каждая система создаётся, основываясь на возможности реализации с технической точки зрения и опираясь на конкретные пожелания заказчика, которые в свою очередь базируются на политике использования проектируемой системы, а также на ориентируются на уровень пользователей, которые будут взаимодействовать с системой.

Были также учтены и недостатки уже существующих систем. Интерфейс должен быть максимально лёгким и комфортным, система не должна быть перегружена анимированными элементами, должна иметь неброский минималистичный дизайн, удобную навигацию и структуризацию.

2 Проектирование информационной системы «Строительство Куйбышевской ГЭС»

2.1 Выбор технологии моделирования информационной системы

На основе сравнительного анализа была создана концептуальная модель информационной системы. На этапе создания концептуальной модели была разработана структурная схема будущего программного обеспечения.

Для представления и интерпретации будущей информационной системы воспользуемся UML. UML – унифицированный язык моделирования (Unified Modeling Language) или система обозначений, которую чаще всего используют для концептуальной структуризации, анализа и проектирования. Благодаря данному языку можно заранее визуализировать свой проект, его индивидуальную спецификацию и конструирование. Иными словами, данный язык можно охарактеризовать как международный язык рисования схем.

Для описания прототипной реализации информационной системы будут использоваться диаграммы вариантов использования (use-case diagram). Они являются отправной точкой в процессе технической реализации, так как предоставляют возможность заранее визуализировать будущую систему и способы коммуникации с ней. Основными элементами при работе с такими диаграммами будут являться элементы актёр (actor) и вариант использования (use case). Актёр – это роль, которую будет выполнять любой сторонний пользователь или любая сторонняя система при взаимодействии с проектируемой системой. Вариант использования – результат, к которому приведёт процесс взаимодействия актёра и системы.

UML является языком графическим описания, благодаря которому можно описать все протекающие процессы на этапе разработки. Для графического описания используются диаграммы, которые различаются по

типу и предназначению. Все диаграммы можно разделить на поведенческие и структурные. Задача поведенческих диаграмм – отобразить процессы, которые протекают внутри системы. В то же время структурные диаграммы описывают элементы, из которых состоит система. Благодаря обширному количеству диаграмм можно подойти к эргономике разработки со всех сторон: и со стороны разработчика, и со стороны пользователя, и со стороны графического дизайнера, и со стороны остальных участвующих в процессе разработки.

CASE-средства помогают не только визуализировать будущую информационную систему, но и структурировать корректный процесс разработки. Основная задача данного этапа – отделить процесс проектирования от написания программного кода.

2.2 Построение диаграммы вариантов использования системы

Для схематичного представления всех действий, которые могут совершаться пользователем, будет использована диаграмма вариантов использования. Задача данной диаграммы заключается в наиболее полном представлении взаимодействия пользователя с различными элементами и результатов этого взаимодействия. На рисунке 6 изображена диаграмма вариантов использования со стороны пользователя.

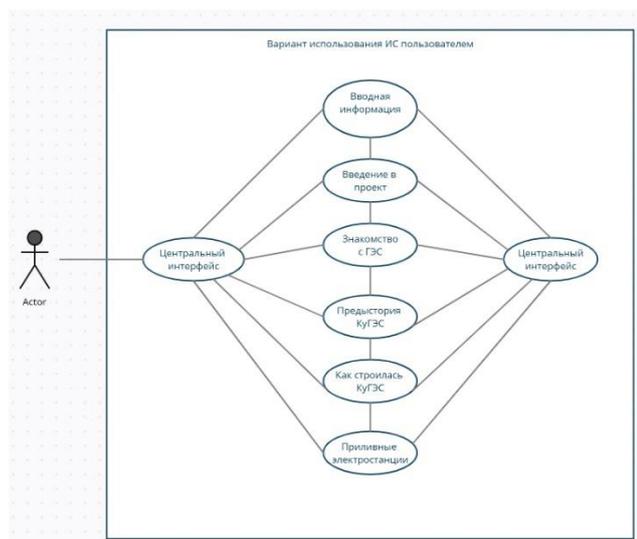


Рисунок 6 – Диаграмма вариантов использования информационной системы пользователем

Здесь представлены вариант взаимодействия пользователя с информационной системой. Основная концепция в блоке main (центральный информационный блок) это сочетание landing page + main block. Иными словами, это совокупность стартовой приветственной страницы и центрального блока, который будет содержать информацию. Задача стартовой страницы информационной системы заключается в предоставлении пользователю ненавязчивого интерфейса и удобной структуризации информации с её классификацией по категориям данных. Находясь на стартовой страницы (landing page), пользователь может взаимодействовать с любой вкладкой, учитывая, что информация будет обновляться только в центральном блоке, который будет содержать соответствующие статьи и фотографии. На рисунке 7 представлен макет страницы.

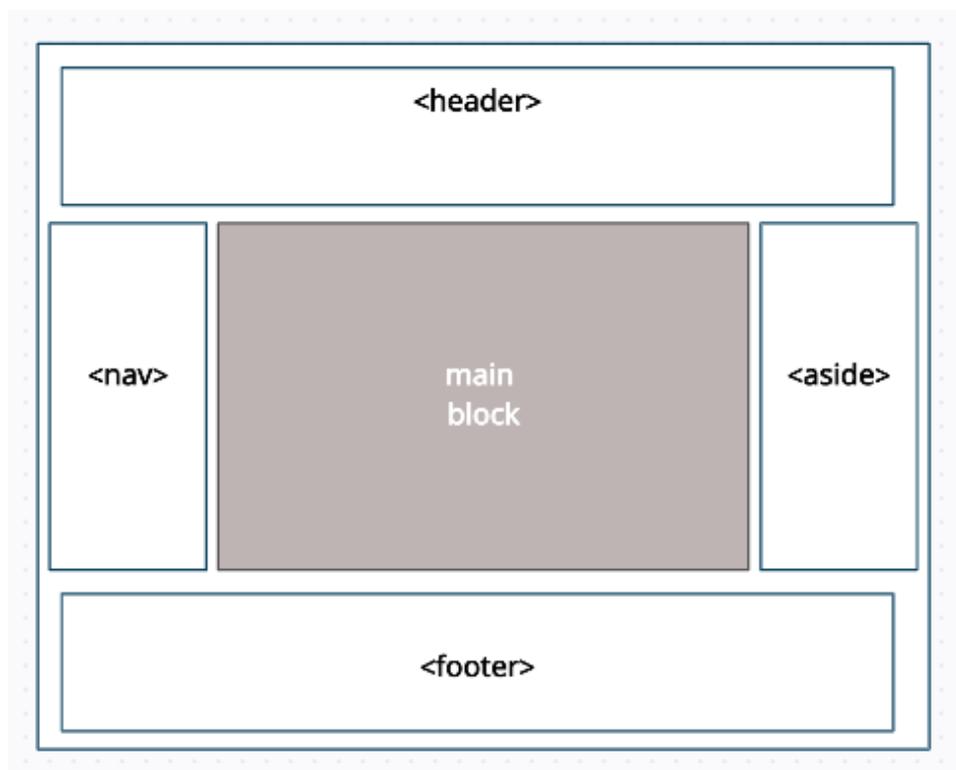


Рисунок 7 – Макет страницы

На рисунке 8 представлена диаграмма взаимодействия разработчика с информационной системой.

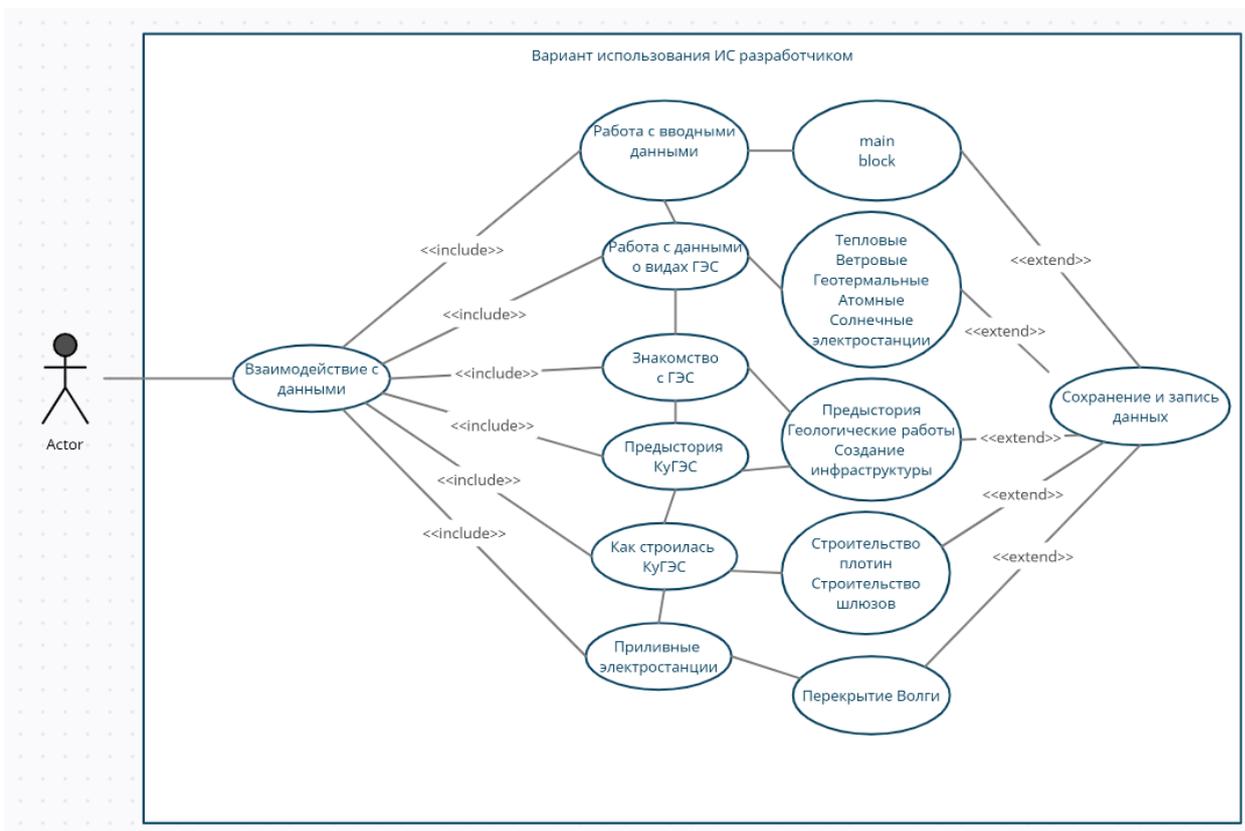


Рисунок 8 – Диаграмма вариантов использования информационной системы разработчиком

На данной диаграмме показаны функции, с которыми будет сталкиваться разработчик в процессе реализации программного кода. Помимо взаимодействия с данными необходимо реализовать интерфейс и подгрузку элементов.

Основная концептуальная идея заключается во взаимодействии с данными, которые будут располагаться в центральном блоке. Проектируемая информационная система должна располагать информацией в центральном блоке, а также заранее определить место для фотографии и позаботиться о функциональной подгрузке элементов.

2.3 Разработка интерфейса

Интерфейс – совокупность технических, программных и методических протоколов, правил, соглашений для средств сопряжения пользователей с устройствами и программами в вычислительной системе, а также устройств с другими устройствами и программами. Интерфейс, в широком смысле слова, представляет собой способ взаимодействия между объектами. Интерфейс в техническом смысле слова задаёт параметры, процедуры и характеристики взаимодействия объектов.

Различают несколько видов интерфейсов, а именно интерфейс пользователя, программный интерфейс и физический интерфейс. Интерфейс пользователя сочетает в себе все возможные варианты взаимодействия пользователя и программной среды. В то же время программный интерфейс используется только для описания использования взаимодействия между программами. Физический интерфейс можно представить как эргономичное размещение устройств, так как в этом случае идёт речь о физическом взаимодействии объектов. В качестве примера физического интерфейса можно привести компьютерные порты.

Для концептуальной визуализации в данной работе необходимо представить, как будет выглядеть пользовательский интерфейс. Затем нужно предоставить макет заказчику и после его утверждения приступить к разработке. Основная задача прототипного пользовательского интерфейса составить некоторый «диалог» между системой и пользователем. Диалог подразумевает под собой обмен информационными ресурсами между компьютером и пользователем в режиме реального времени. Это диалог должен быть направлен на решение какой-либо конкретной задачи, в данном случае эта задача носит образовательный характер.

Чаще всего словосочетание «пользовательский интерфейс» воспринимается как «макет» или «внешний вид» разрабатываемого приложения. Однако разработка пользовательского интерфейса

подразумевает не только разработку визуальной составляющей проектируемого программного обеспечения, но и проектирование программных компонентов, с которыми будет взаимодействовать пользователь. На этапе разработки интерфейса необходимо определить технические возможности, позволяющие достигнуть желаемого результата и создать некий «образ», от которого дальше будет отталкиваться разработчик.

К программным элементам, которые позволяют пользователю взаимодействовать с информационной системой, относятся:

- набор задач пользователя, которые он решает благодаря данной информационной системе;
- навигация между блоками системы;
- визуальный дизайн;
- средства, благодаря которым информация будет отображаться на экране;
- устройства и технологии ввода данных;
- информационный поиск.

Далее необходимо определить функциональную границу между технической средой выполнения и визуальным проектом.

Для разработки макета воспользуемся графическим онлайн-редактором для проектирования и дизайна интерфейсов Figma. Среда разработки представлена на рисунке 9.

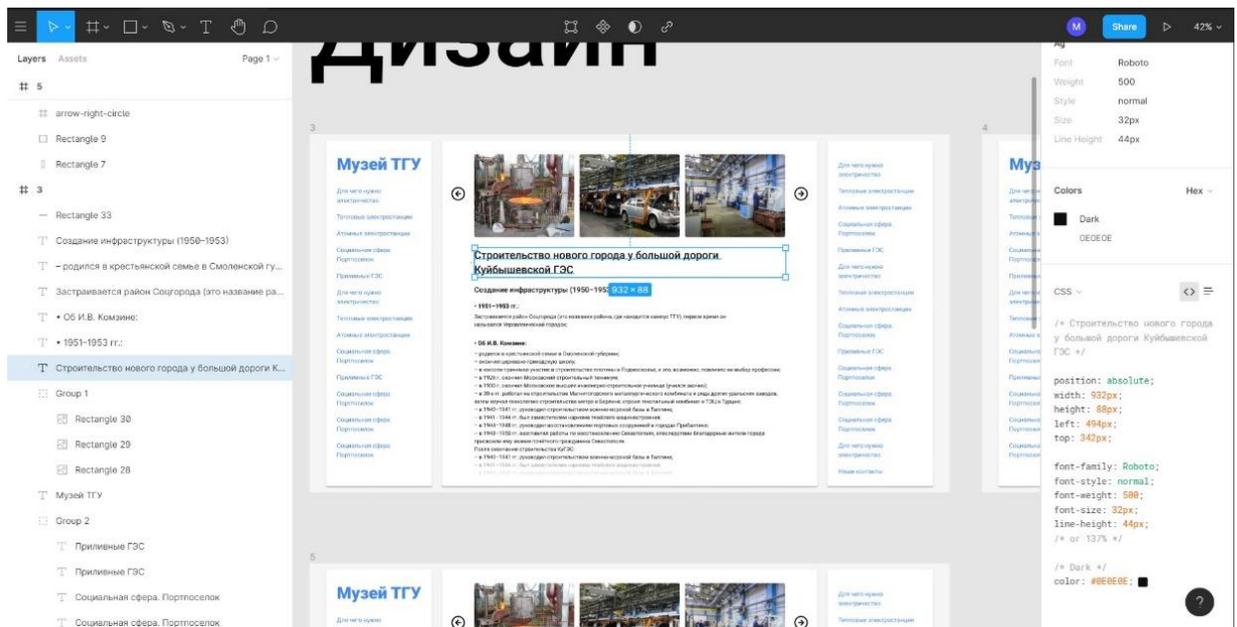


Рисунок 9 – Среда разработки для проектирования интерфейсов Figma

Данный онлайн-редактор создан специально для frontend-разработчиков, которые специализируются на разработке пользовательских интерфейсов. Figma отличается от других редакторов преимущественно простым интерфейсом, в котором учитываются все данные для разработчика. Подключая различные дополнительные плагины, можно адаптировать эту среду под создание коммерческих дизайнов, логотипов или презентаций.

Сначала необходимо определиться со спецификациями элементов, с которыми будет взаимодействовать пользователь. Чтобы глаз человека не так уставал от восприятия ярких элементов при длительном взаимодействии, был сделан выбор в пользу спокойных бело-серых оттенков и минималистичного шрифта. Гамма, на которой будет основан проект, будет состоять преимущественно из серых тонов с синими оттенками, так как именно в них выполнен макет логотипа музея ТГУ. Спецификация элементов представлена на рисунке 10.

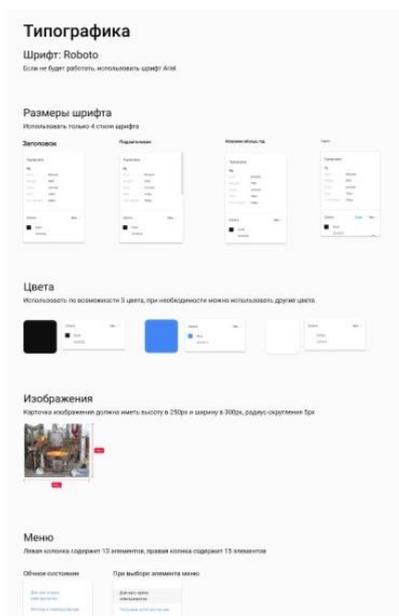


Рисунок 10 – Спецификация элементов макета ИС

При создании макета необходимо было учесть эргономику элементов системы, опираясь на пожелания заказчика. Данная информационная система будет содержать достаточно большое количество информации, для работы с которой необходимо разработать комфортную для пользователя систему навигации. На макете, представленном на рисунке 11, визуализирован фрагмент информационного блока, который содержит элементы, с которыми по желанию может взаимодействовать пользователь, и ключевую информацию, относящуюся к тематике блока, которая может быть изменена для центрального блока.



Рисунок 11 – Макет разрабатываемой информационной системы

На основе представленного макета будет разрабатываться уникальная информационная система в соответствии с индивидуальным дизайном, сочетающим эргономику элементов взаимодействия пользователя с системой и возможность программной реализации.

Основная цель разработки пользовательского интерфейса заключалась в разработке максимально комфортной среды для пользователя с минимальными навыками пользования компьютером.

2.4 Описание алгоритма информационного поиска

Задача алгоритма информационного поиска заключается в выдаче ориентировочного перечня ответов в ответ на запрос пользователя. В случае проектируемой информационной системы перечень ответов представляет собой совокупность данных и web-страниц.

Данный алгоритм работает на основе внутренних и внешних ссылок и степени их важности. Степень важности зависит от тегов web-страницы или количества ключевых слов, которые она содержит.

Для классического ранжирования используется метод PageRank. Этот метод используется для оценки важности сайта, его просматриваемости, после чего в зависимости от его посещаемости он занимает своё место в поисковой системе.

Для математического представления данного алгоритма необходимо представить переходную матрицу, у которой существует предел (1):

$$p^T(t + 1) = p^T(t)P, \quad (1)$$

где

$p^T(t + 1)$ – стационарное распределение вероятности, которое определяется методом простых итераций;

$p^T(t)$ – матрица перехода, на основе которой определяется вероятность, что пользователь перейдёт по подобранной ссылке;

P – переходная матрица.

Чтобы представить матрицу перехода, используется уравнение Колмогорова-Чепмена. Затем задаётся предел, на основе которого рассчитывается, с какой вероятностью пользователь перейдёт по ссылке.

В настоящее время метод PageRank заложен во все поисковые алгоритмы в упрощённом, усовершенствованном или усложнённом формате. В классическом виде формула расчёта выглядит как (2):

$$PR = (1 - d) + d \sum_{i=1}^n \frac{PR_i}{C_i}, \quad (2)$$

где

PR – вычисляемый с помощью алгоритма PageRank показатель для рассматриваемой страницы;

d – коэффициент затухания, который определяет вероятность того, что пользователь, зашедший на страницу, перейдёт по одной из ссылок, а не прекратит поиск;

n – общее количество страниц;

PR_i – показатель PageRank i -й страницы, ссылающейся на рассматриваемую страницу;

C_i – общее число ссылок на i -й странице.

Так как алгоритм PageRank используется для оценки качества ресурса во всемирной сети, для проектируемой системы он будет реализован в упрощённом формате. В данном случае ссылочное ранжирование будет основываться на теге заголовка подгружаемой страницы и ключевых словах. На основе совокупности этих данных будет подбираться наиболее подходящий вариант под поисковый запрос. Интерфейс пользователя будет содержать поле для ввода ключевых слов и выпадающий список, на основе которых будет подгружаться перечень ответов на запрос пользователя.

Выводы по главе

В данной главе было рассмотрено концептуальное моделирование, на основе которого рассматривается диалог взаимодействия проектируемой информационной системы с пользователем. Для описания диалога была выбрана методология UML, а именно диаграммы вариантов использования use-case.

Были представлены варианты взаимодействия пользователя с системой, её желаемая эргономика и функциональность. Так же был представлен вариант взаимодействия разработчика с системой, его функциональные возможности и задачи.

В результате была определена совокупность желаемых характеристик с точки зрения технической стороны разработки, с точки зрения концептуально-логической модели и с точки зрения дизайна пользовательского интерфейса. Был разработан макет, удовлетворяющий требованиям, указанным в первой главе, сочетающий в себе современную подачу контента, минималистичный приятный дизайн и удобную навигацию. Для поиска информации был рассмотрен математический алгоритм PageRank, на основе которого система будет сразу перенаправлять пользователя по его искомому запросу. Была рассмотрена математическая формула, на основе которой работает алгоритм PageRank, а также формула, позволяющая реализовать алгоритм PageRank на основе взаимодействия пользователя с web-сайтом.

3 Программная реализация информационной системы «Строительство Куйбышевской ГЭС»

3.1 Организация сервера

Данная информационная система будет реализована в формате веб-сайта на локальном хостинге. В данной главе будут рассмотрены все элементы, которые будут составлять разрабатываемую информационную систему, то есть и frontend-, и backend-элементы.

Для удобства настройки и работы сервера была использована бесплатная операционная система Ubuntu 20.4 LTS. Установка системы производилась с внешнего носителя – USB-флэш-карты – с заранее записанным образом Ubuntu. Для установки применялась загрузка с live-CD для первоначальной настройки. Интерфейс программы загрузки представлен на рисунке 12. На рисунке показан процесс создания разделов на диске.

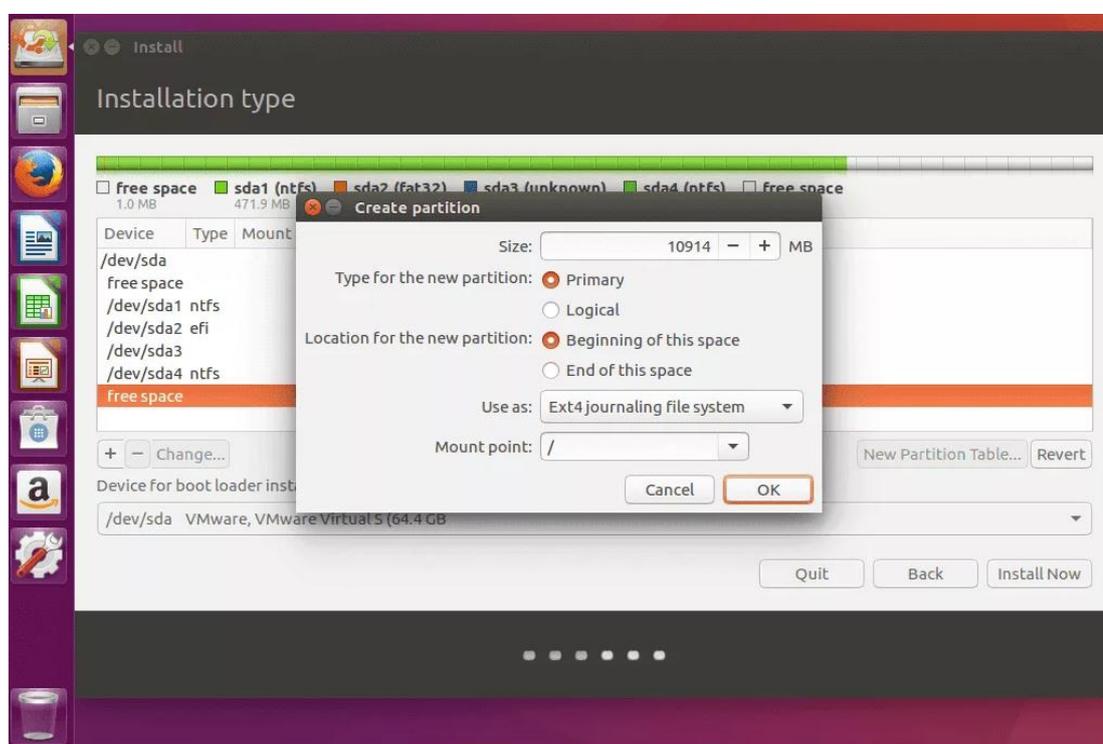


Рисунок 12 – Интерфейс программы установки операционной системы Ubuntu

Рисунок 14 – Скриншот процесса завершения установки программного сервера Apache2

После успешной установки web-сервера операционная система может передавать браузеру данные о сайте: изображения и сами страницы, также аудио- и видеоматериалы. Однако этого недостаточно для работы сайта, поскольку не настроены конфигурационные файлы сервера.

Для того чтобы их настроить, необходимо перейти в конфигурационные файлы Apache2, которые находятся по адресу `/etc/apache2`. В терминальном редакторе Nano необходимо открыть файл `apache.conf`. На рисунке 15 показана команда для открытия текстового редактора.

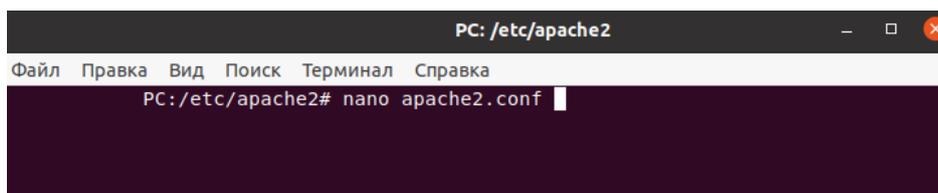


Рисунок 15 – Команда для открытия текстового редактора

Для удобства разработки место хранения файлов сайта необходимо сменить. Вместо «`/var/www/`» нужно прописать «`/home/www/`» (рисунок 16). Это требуется для того, чтобы было более удобно изменять файлы в процессе редактирования сайта.

```
Файл  Правка  Вид  Поиск  Терминал  Справка
GNU nano 4.8                               apache2.conf                               Изменён
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /home/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

#<Directory /srv/>
#     Options Indexes FollowSymLinks
#     AllowOverride None
#     Require all granted
#</Directory>
```

Рисунок 16 – Скриншот окна редактора nano Apache2

Затем необходимо сохранить этот файл и перезагрузить сервер командой «sudo apache2 restart». Это необходимо для того, чтобы настройки применились.

Теперь рассмотрим каждый функциональный файл, который позволяет информационной системе работать корректно.

3.2. Реализация интерфейса

Файл стилей style.css – это файл, в котором заданы значения параметров отображения информации, в том числе параметры длины и ширины полей, количества и формы блоков, их размер, цвет, шрифт и толщина различных рамок. Все эти параметры отвечают за визуальное отображение контента системы. Фрагмент кода представлен на рисунке 17.

```
body {  
  padding: 0;  
  margin: 0;  
  height: 1080px;  
}  
  
h5 {  
  text-align: center;  
  margin: 5px auto;  
}  
  
ul {  
  list-style-type: none;  
  margin: 0px 2px;  
}  
  
li {  
  margin-top: 10px;  
}  
  
.center_container {  
  display: flex;  
  height: 1080px;  
}
```

Рисунок 17 – Фрагмент кода файла стилей style.css

Для визуального оформления контента системы необходимо определить блоки, которые будут содержать навигацию, а также центральный блок. Интерфейс основывался на макете, который был спроектирован в предыдущей главе. Спецификация элементов системы представлена на рисунке 18.

```

/* затемняющий фон */
#overlay {
  display: none;
  background: #000;
  position: fixed;
  top: 0;
  left: 0;
  height: 100%;
  width: 100%;
  opacity: 0.5;
  z-index: 9990;
}

/* кнопка закрытия */
#close-popup {
  width: 30px;
  height: 30px;
  background: #FFFFFF;
  border: 1px solid #AFAFAF;
  border-radius: 15px;
  cursor: pointer;
  position: absolute;
  top: 15px;
  right: 15px;
}

#close-popup i {
  width: 30px;
  height: 30px;
  background: url(https://codernote.ru/files/cross.png) no-repeat center center;
  background-size: 16px 16px;
  display: block;
}

```

Рисунок 18 – Фрагмент кода, содержащий спецификацию элементов системы

На рисунке 17 показаны стилизация внешнего элемента <body> с выравниванием текста по центру, размером и расположением блоков на экране, заполнение цветом центральных и второстепенных блоков. На рисунке 18 показано заполнение фоновым цветом и оформление кнопки, которая будет выполнена в виде логотипа музея ТГУ. Основопологающая задача данного файла – определение визуальных характеристик frontend-элементов.

3.3 Подгрузка элементов

Все элементы, необходимые для работы информационной системы, находятся в папке на сервере, благодаря чему данный web-сайт может работать без подключения к сети Интернет. Папки, содержащие элементы информационной системы, представлены на рисунке 19.

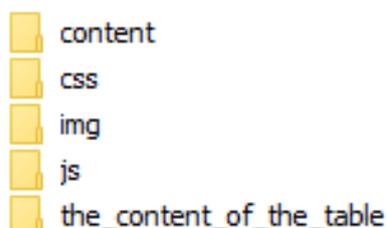


Рисунок 19 – Папки для элементов информационной системы

В папках находятся следующие элементы:

- img (фотографии и изображения);
- content (структурированная информация);
- css (файл style.css, который отвечает за оформление);
- js (javascript-код для обеспечения подгрузки элементов в центральном блоке).

Для полноценной работы информационной системы необходимо обеспечить подгрузку элементов и текстового блока, которые будут меняться в зависимости от выбранной пользователем категории.

Подгрузка элементов информационной системы будет осуществляться с помощью JavaScript-библиотеки jQuery. Это многофункциональная библиотека, которая позволяет более компактно представлять код, что значительно сокращает время на поиск и устранение ошибок.

Это не единственное достоинство этой библиотеки. Она обладает простым и понятным синтаксисом, открытым кодом, а также возможностью взаимодействовать с любыми браузерами. Если код, написанный

исключительно на JavaScript, не поддерживает некоторые элементы, необходимо использовать полифилы.

На рисунке 20 представлен скриншот, на котором показан фрагмент файла, содержащего библиотеку jQuery, которая обеспечивает подгрузку элементов центрального блока с информацией с помощью языка JavaScript.

```
713
714     function U(e, t) { return t.toUpperCase() }
715
716     function X(e) { return e.replace(/ms-/g, U) }
717     var V = function(e) { return 1 === e.nodeType || 9 === e.nodeType || !+e.nodeType };
718
719     function G() { this.expando = S.expando + G.uid++ }
720     G.uid = 1, G.prototype = {
721       cache: function(e) { var t = e[this.expando]; return t || (t = {}, V(e) && (e.nodeType ? e[this.expando] = t : Object.defineProperty(e,
722         set: function(e, t, n) {
723           var r, i = this.cache(e);
724           if ("string" == typeof t) i[X(t)] = n;
725           else
726             for (r in t) i[X(r)] = t[r];
727           return i
728         },
729       get: function(e, t) { return void 0 !== t ? this.cache(e) : e[this.expando] && e[this.expando][X(t)] },
730       access: function(e, t, n) { return void 0 !== t || t && "string" == typeof t && void 0 !== n ? this.get(e, t) : (this.set(e, t, n), void
731       remove: function(e, t) { var n, r = e[this.expando]; if (void 0 !== r) { if (void 0 !== t) { n = (t = Array.isArray(t) ? t.map(X) : (t
732       hasData: function(e) { var t = e[this.expando]; return void 0 !== t && !S.isEmptyObject(t) }
733     };
734     var Y = new G,
735         Q = new G,
736         J = /^(?:(?!(\w|W)*\|)\[?[\w|W]*\])?$/,
737         K = /[A-Z]/g;
738
739     function Z(e, t, n) {
740     S.extend({ hasData: function(e) { return Q.hasData(e) || Y.hasData(e) }, data: function(e, t, n) { return Q.access(e, t, n) }, removeData:
```

Рисунок 20 – Скриншот, демонстрирующий фрагмент файла библиотеки jQuery

Особенность использования данной библиотеки заключается не только в её функциональных возможностях, но и в наличии уникальной get-функции, которая позволяет отобразить отдельные блоки информации в центральной консоли без необходимости перезагрузки всей страницы. Содержимое файла скрипта подгрузки представлен на рисунке 21.

```
1  function load_content(url_content) {
2    $.get(url_content, function(data) {
3      $('#middle').html('');
4      $('#middle').append(data);
5    });
6  }
```

Рисунок 21 – Скриншот содержимого файла скрипта подгрузки

Благодаря данной библиотеке элементы на сайте воспроизводятся динамически, что делает разработку простой и удобной. В результате выполнения get-функция формируется блок, который получает данные по указанной пользователем ссылке и на основе полученных данных подгружает необходимый контент.

Рассмотрим случай, при котором нет необходимости использовать функционал языка js. В таком случае информационная система представляла бы из себя набор статичных html-документов, которые бы менялись между собой, полностью перегружая сайт, то есть производились бы переходы по ссылкам. В этом случае разработка бы сильно усложнилась, так как при изменении одного элемента на одной из 30 html-страниц, пришлось бы менять все остальные страницы, что крайне трудоёмко и занимает значительное количество времени при большом количестве страниц. Также от такого подхода сильно страдает эргономика, так как система может вести себя непредсказуемо – например самовольно сворачиваться или менять режим с полноэкранный на оконный, что в данном случае недопустимо.

Помимо подгружаемых текстовых элементов в центральном блоке, стояла задача реализации небольшой галереи изображений. Суть создания галереи заключалась в возможности масштабировать изображение при взаимодействии с ним, то есть увеличивать интересующий элемент. Для этого необходимо написать скрипт, который позволил бы работать с любыми изображениями на странице, увеличивать и уменьшать изображение, а также выводить его в центр main-блока.

Такой скрипт должна выполнять:

- отцентровку изображения в main-блоке;
- при клике на изображении масштабировать его и затемнять фон;
- при клике на любом участке интерфейса изображение должно свернуться.

Для начала необходимо вывести изображение. Фрагмент скрипта представлен на рисунке 22.

```
<div class="image">
  
</div>
```

Рисунок 22 – Скриншот скрипта для вывода изображения

Затем необходимо описать визуальные настройки. Скрипт соответствующей функции представлен на рисунке 23.

```
<head>
<script type="text/javascript">
$(document).ready(function() {
// Обработка клика на картинке
$('.img_block img').click(function() {
// Получаем адрес картинки
var imgAddr = $(this).attr("src");
// Задаем свойство SRC картинке, которая в скрытом диве.
$('#img_big_block img').attr({src: imgAddr});
// Показываем скрытый контейнер
$('#img_big_block').fadeIn('slow');
});
// Обрабатывает клик по большой картинке
$('#img_big_block').click(function() {
$(this).fadeOut();
});
});
</script>
```

Рисунок 23 – Скриншот скрипта функции масштабирования

Процесс настройки включает выбор размера желаемого изображения, размера увеличенной картинки или фотографии, настройку цвета затемняющего фона.

3.4 Реализация алгоритма информационного поиска

Задача алгоритма информационного поиска заключается в сравнении ключевых слов, которые пользователь вводит с клавиатуры. На основе этих слов будет происходить подбор страницы, которая с наибольшей вероятностью подходит под введённый запрос. Так как при взаимодействии с поисковым полем, адрес страницы будет меняться, скрипт будет учитывать не только ключевые слова, но и все URL-адреса, после чего отобразит желаемый результат в центральном блоке.

Алгоритм также будет предлагать вариант выпадающего списка с ключевыми словами, на основе которых пользователь сможет выбрать интересующий его фрагмент. Фрагмент кода скрипта, реализующего информационный поиск, представлен на рисунке 23.

```
<script type="text/javascript">
var lastResFind=""; // последний удачный результат
var copy_page=""; // копия страницы в исходном виде
function TrimStr(s) {
    s = s.replace( /\s+/g, '' );
    return s.replace( /\s$/g, '' );
}
function FindOnPage(inputId) { //ищет текст на странице, в параметр передается ID поля для ввода
    var obj = window.document.getElementById(inputId);
    var textToFind;

    if (obj) {
        textToFind = TrimStr(obj.value); //обрезаем пробелы
    } else {
        alert("Введенная фраза не найдена");
        return;
    }
    if (textToFind == "") {
        alert("вы ничего не ввели");
        return;
    }

    if (document.body.innerHTML.indexOf(textToFind)=="-1")
        alert("ничего не найдено, проверьте правильность ввода!");

    if (copy_page.length>0)
        document.body.innerHTML=copy_page;
    else copy_page=document.body.innerHTML;

    document.body.innerHTML = document.body.innerHTML.replace(eval("/name="+lastResFind+"/g1")," "); //стираем предыдущие якоря для скрала
    document.body.innerHTML = document.body.innerHTML.replace(eval("/"+textToFind+"/g1"),
    ["<a name="+textToFind+" style='background:red'>"+textToFind+"</a>"]); //Заменяем найденный текст ссылками с якорем;
    lastResFind=textToFind; // сохраняем фразу для поиска, чтобы в дальнейшем по ней стереть все ссылки
    window.location = "#"+textToFind; //перемещаем скрол к последнему найденному совпадению
}
</script>
```

Рисунок 23 – Код скрипта, реализующего информационный поиск

Необходимо также оформить стиль окна ввода, с которым будет взаимодействовать пользователь. Требуется настроить размер окна, цвет, оформление и поле ввода. После HTML-настроек окно поиска можно отображать на центральной странице. Оформление поля ввода представлено на рисунке 24.



Рисунок 24 – Скриншот поля ввода поиска

Была настроена форма окна, поле поиска, вставлен текст и лупа, чтобы пользователю было комфортно ориентироваться и взаимодействовать с данной строкой. Интерфейс разработанной информационной системы представлен на рисунках 25, 26.



Рисунок 25 – Интерфейс главной страницы системы «Строительство Куйбышевской ГЭС»

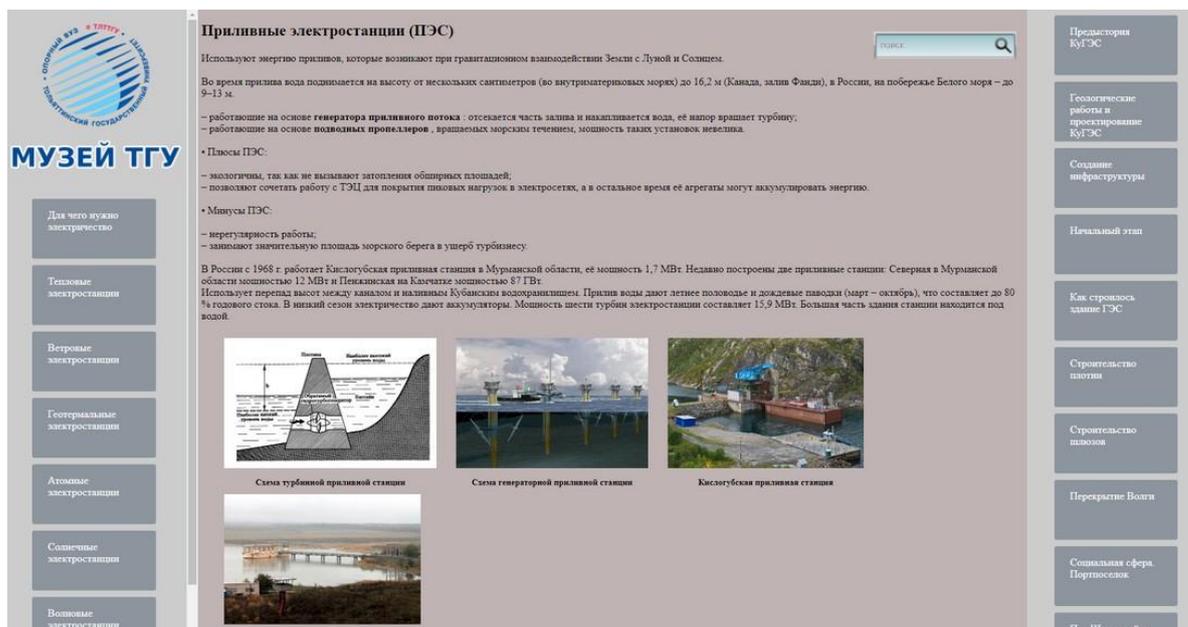


Рисунок 26 – Интерфейс одной из страниц системы «Строительство Куйбышевской ГЭС»

Выводы по главе

В данной главе были рассмотрены этапы frontend- и backend-разработки. Были выполнены первоначальные настройки сетевого администрирования.

Для корректной работы информационной системы был организован сервер. Концептуальной задумкой была реализации системы не только для локального пользователя, но при необходимости и для глобального.

Были реализованы визуальный профиль, центральный и второстепенные блоки, их оформление, а также сам контент.

Заключительным этапом была разработка backend-элементов, которые отвечают за подгрузку элементов данной системы. Были описаны функции, благодаря которым осуществлялась замена информации в центральном блоке в зависимости от выбранной вкладки, а также загрузка и масштабирование изображений. Была реализована функция для информационного поиска, а также настроено и оформлено поле, в которое вводится поисковый запрос.

Заключение

В ходе выполнения выпускной квалифицированной работы была создана информационная система «Строительство Куйбышевской ГЭС» с поддержкой алгоритма информационного поиска. Реализованная система предназначена для просмотра исторических данных с удобной классификацией по определённым категориям.

В дальнейшем при необходимости данная система может расширять свои функциональные возможности, так как она была реализована для интерактивного стола с возможностью модернизации.

В процессе реализации данного проекта были решены все поставленные задачи. Был осуществлён перевод информации с физических бумажных носителей в цифровой формат для возможности дальнейшего сохранения ценных исторических документов, были выполнены frontend- и backend-задачи, были созданы и согласованы с заказчиком дизайн-макеты, а впоследствии сотрудникам музея были представлены варианты разработки данной системы.

Завершающим этапом разработки был выбор средств и методов. Были проведены первоначальные настройки сервера для корректной работы информационной системы. Также был реализован алгоритм информационного поиска, который помогает быстро и комфортно ориентироваться на странице, чтобы пользователь оперативно получить интересующую его информацию.

Таким образом, все поставленные задачи были реализованы. Данная система уникальна и не имеет похожих аналогов, что позволяет ей в дальнейшем развиваться.

Список используемой литературы

1. Александров, Д.В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы: Учебное пособие / Д.В. Александров. – М.: ФиС, 2011. – 224 с.
2. Алексеев, А.П. Введение в Web-дизайн: учебное пособие / А.П. Алексеев / – М.: СОЛОН-ПРЕСС, 2008. – 192 с.
3. Барановская, Т.П. Информационные системы и среды. 2014. – 654 с.
4. Бородаев, Д.А. WEB-сайт как объект графического дизайна: монография / Д.А. Бородаев. – Харьков, 2004. – 232 с.
5. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, А. Джекобсон. – М.: ДМК, 2015. – 432 с.
6. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. – М.: Финансы и статистика, 1998. – 176 с.
7. Гранд, М. Шаблоны проектирования в Java / М. Гранд. – М.: Новое знание, 2004. – 559 с.
8. Грин, Д. Математические методы анализа алгоритмов / Д. Грин, Д. Кнут. – М.: Мир, 1987. – 120 с.
9. Гультияев, А.К. Проектирование и дизайн пользовательского интерфейса / А.К. Гультияев, В.А. Машин. – М.: Корона-Принт, 2010. – 350 с.
10. Кит, Д. HTML5 для веб-дизайнеров / Д. Кит. – М.: Манн, Иванов и Фербер, 2012. – 214 с.
11. Журавлёва, Т.Ю. Информационные технологии: учебное пособие / Т.Ю. Журавлёва – Саратов: Вузовское образование, 2018. – 72 с.
12. Киммел, П. UML. Основы визуального анализа и проектирования / П. Киммел. – М.: ИТ Пресс, 2008. – 272 с.

13. Левитин, А.В. Алгоритмы: введение в разработку и анализ / А.В. Левитин. – М.: Издательский дом «Вильямс», 2006. – 576 с.
14. Носова, Л.С. Case-технологии и язык UML: учебно-методическое пособие / Л.С. Носова – Челябинск, Саратов: Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. – 67 с.
15. Окулов, С.М. Алгоритмы обработки строк. Учебное пособие / С.М. Окулов. Издательство: БИНОМ. Лаборатория знаний, 2012. – 256 с.
16. Программирование и основы алгоритмизации: Учеб. пособие / В.Г. Давыдов. – 2-е изд., стер. – Издательство: Высш. шк., 2005. – 447 с.
17. Сырых, Ю.А. Современный веб-дизайн. Настольный и мобильный / Ю.А. Сырых. - М.: Диалектика, Вильямс, 2014. – 108 с.
18. Уолтер, А. Эмоциональный веб-дизайн / А. Уолтер. – М.: Манн, Иванов и Фербер, 2012. – 140 с.
19. A re-introduction to JavaScript (JS tutorial) : на англ. яз. : [Электронный ресурс] / MDN web docs : moz://a – Режим доступа: https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_reintroduction_to_JavaScript (дата обращения 11.03.2021).
20. Codecademy [Электронный ресурс]: <https://www.codecademy.com/> (дата обращения: 22.04.2021).
21. JavaScript Tutorial : на англ. яз. : [Электронный ресурс] / www.w3schools.com : THE WORLD'S LARGEST WEB DEVELOPER SITE – Режим доступа: <https://www.w3schools.com/js/default.asp> (дата обращения 11.03.2021).
22. JQuery Tutorial : на англ. яз. : [Электронный ресурс] / www.w3schools.com : THE WORLD'S LARGEST WEB DEVELOPER SITE – Режим доступа: <https://www.w3schools.com/jquery/default.asp> (дата обращения 11.03.2021).

23. Mohit Malhotra. Everything About Software Architecture / [Электронный ресурс] URL: <https://medium.com/swlh/everything-aboutsoftware-architecture-dfd2b9351ef4> (дата обращения: 27.03.2021).