

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

РАБОТА ПРОВЕРЕНА

Рецензент

_____ 2023 г.
«__»_____

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой ЭВМ

_____ Д.В. Топольский

«__»_____ 2023 г.

Разработка клиентской части информационной системы для решения
поисковых задач материаловедения

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090401.2023.222 ПЗ ВКР

Руководитель работы,
к.т.н., доцент каф. ЭВМ

_____ И.Л. Кафтанников

«__»_____ 2023 г.

Автор работы,
студент группы КЭ-222

_____ А.С. Герасимов

«__»_____ 2023 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ

_____ С.В. Сяськов

«__»_____ 2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

_____ Д.В. Топольский

«___» _____ 2023 г.

ЗАДАНИЕ

на выпускную квалификационную работу магистра
студенту группы КЭ-222
Герасимову Александру Сергеевичу,
обучающемуся по направлению
09.04.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Разработка клиентской части информационной системы для решения поисковых задач материаловедения» утверждена приказом по университету от 25 апреля 2023 №753-13/12, приложение №308/10.
2. **Срок сдачи студентом законченной работы:** 1 июня 2023 г.
3. **Исходные данные к работе:**
Клиентский интерфейс информационной системы обеспечивает для пользователей следующие возможности:
 - поиск результатов исследований в области материаловедения по различным критериям;
 - скачивание результатов исследований на своё устройство, а также загрузка их со своего устройства в озеро данных;

- работа с различными файловыми форматами, хранящими результаты исследований;
- визуализация исследуемых веществ и материалов.

4. Перечень подлежащих разработке вопросов:

1. Анализ предметной области;
2. определение требований;
3. проектирование;
4. реализация;
5. тестирование.

5. Дата выдачи задания: 1 декабря 2022 г.

Руководитель работы _____ /И.Л. Кафтанников/

Студент _____ /А.С. Герасимов/

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Анализ предметной области	10.03.2023	
Определение требований	21.03.2023	
Проектирование	04.04.2023	
Реализация	25.04.2023	
Тестирование	07.05.2023	
Компоновка текста работы и сдача на нормоконтроль	16.05.2023	
Подготовка презентации и доклада	24.05.2023	

Руководитель работы _____ /И.Л. Кафтанников/

Студент _____ /А.С. Герасимов/

АННОТАЦИЯ

А.С. Герасимов. Разработка клиентской части информационной системы для решения поисковых задач материаловедения. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2023, 59 с., 20 ил., библиогр. список – 15 наим.

В рамках выпускной квалификационной работы производится анализ современной научно-технической, нормативной, методической литературы, затрагивающей научно-техническую проблему. Организуется разработка клиентской части веб-приложения для решения поисковых задач материаловедения. В ходе выполнения работы рассмотрены и решены проблемы, связанные с созданием оптимизированных клиентских приложений для работы с большим количеством информации.

Разработанная программа позволяет организовать работу с собственной базой исследований, а также ускорить процессы работы и исследований и повысить качество получаемых результатов, связанных с материаловедением, что является важным шагом в развитии направления материаловедения в университете.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	9
1.1. Обзор аналогов.....	12
1.2. Анализ основных технологических решений.....	17
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ	22
2.1. Функциональные требования.....	22
2.2. Нефункциональные требования.....	22
3. ПРОЕКТИРОВАНИЕ.....	24
3.1. Архитектура предлагаемого решения	24
3.2. Алгоритмы решения задачи	28
3.3. Описание данных.....	31
4. РЕАЛИЗАЦИЯ	36
5. ТЕСТИРОВАНИЕ.....	43
5.1. Методология тестирования	43
5.2. Проведение процедуры тестирования.....	43
ЗАКЛЮЧЕНИЕ	46
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	47
ПРИЛОЖЕНИЯ.....	49
ПРИЛОЖЕНИЕ А. Скрипт инициализации приложения	49
ПРИЛОЖЕНИЕ Б. Код конфигурации сборки приложения.....	50
ПРИЛОЖЕНИЕ В. Код конфигурации компиляции typescript	51
ПРИЛОЖЕНИЕ Г. Скрипт работы с файлами материаловедения.....	52

ВВЕДЕНИЕ

Материаловедение – это наука, которая изучает структуру, свойства и производственные применения материалов. Она имеет огромное значение в различных отраслях промышленности, таких как машиностроение, электроника, строительство, медицина и другие.

На современном уровне научных исследований использование баз данных стало рутинным инструментом, без которого невозможно планирование эксперимента, контроль актуальности и его новизны. Такой подход к разработке новых материалов в самых современных областях инженерии обусловлен переходом к 4-ой парадигме материаловедения, знаменующейся стартом использования больших массивов накопленных экспериментальных и теоретических данных и изображен на рисунке 1.

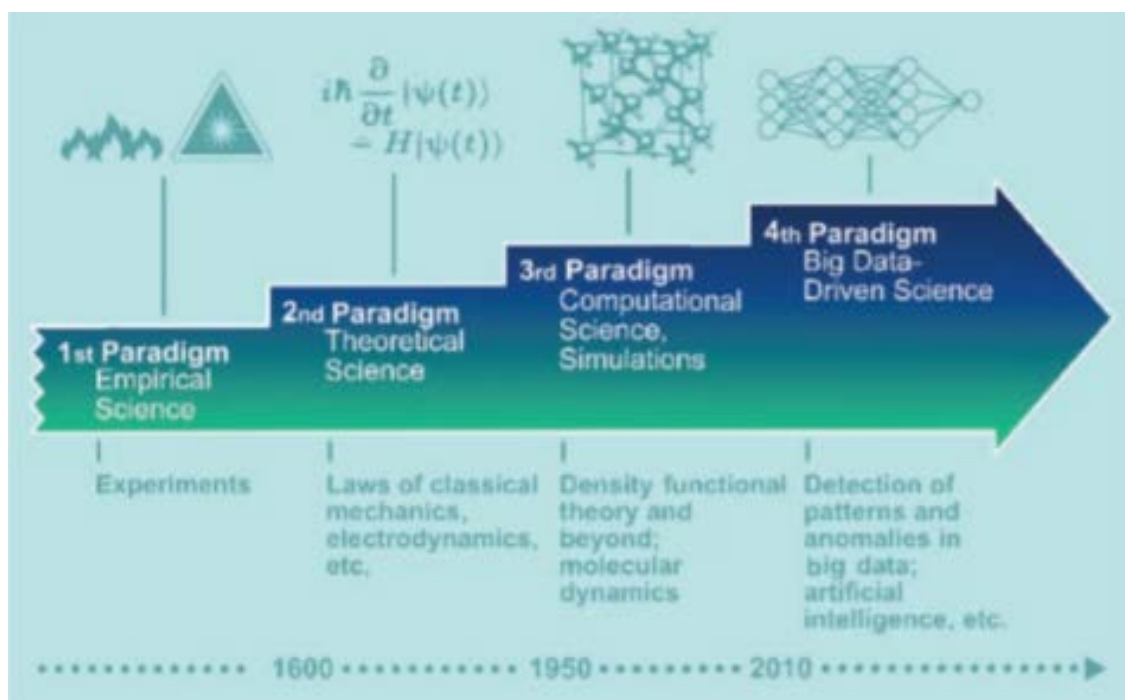


Рисунок 1 - Исторические парадигмы разработки и внедрения новых материалов [1]

Эта парадигма основана на упорядочении накопленного массива данных в терминах множества расчетных и экспериментальных дескрипторов, которые организованы в базу данных с возможностями поиска, аналитики и кластеризации методами искусственного интеллекта. Поиск взаимосвязей структура-свойство на больших массивах данных с учетом сотен дескрипторов позволяет простроить «карту» материалов, схематически изображенную на рисунке 2, представляющую собой объединение материалов с определенными свойствами, обладающих сходными значениями дескрипторов, определяющих эти свойства.

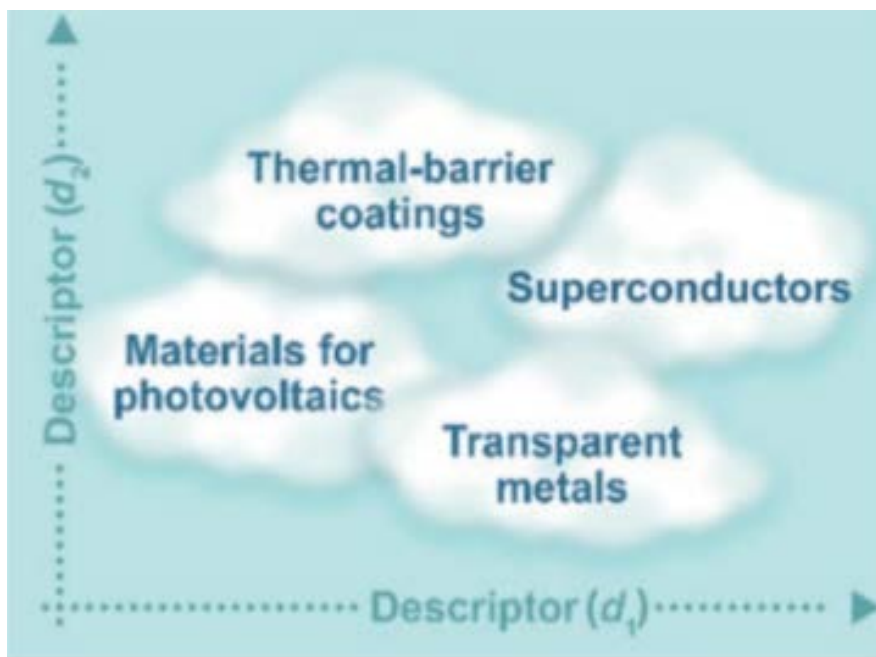


Рисунок 2 - Основа поиска искомых материалов (оптически прозрачных, суперпроводящих, для фотовольтаики, для термоизоляционных покрытий и т.д.) в многомерном пространстве дескрипторов [1]

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Исходя из того, что главной парадигмой современного материаловедения является установление следствий и связей свойств структур микро и наноуровней с макроуровнем, необходимым фактором дальнейшего развития становится разработка масштабируемых информационных систем для цифрового описания, преобразований, моделей, библиотек, структур, коммуникаций, баз данных и программно - аппаратных средств поддержки. Существенным фактором также является наличие сегодня наряду с традиционными научными исследованиями и технологиями формирование и развитие системной цифровой трансформации всех сфер деятельности, в частности технологий цифровых двойников.

Цифровой двойник — это цифровая (информационная, виртуальная) модель любых сущностей (объектов, систем, процессов, людей и т.п.).

С появлением новых технологий, возникла возможность создания цифровых двойников материалов, что позволяет ускорить процесс поиска и сравнения материалов.

Разработка цифровых двойников материалов – задача, отвечающая вызовам нашего времени. Цифровые двойники позволяют моделировать применение новых материалов и конструкций, выявлять экономические и экологические риски при внедрении инноваций, еще на этапе проектирования [2]. Отметим, что оптимальный выбор материала при проектировании продукта является важнейшим вопросом при реализации устойчивого производства [3]. Технология цифрового двойника может учитывать поведение изделия при нагрузках и оптимизировать процессы, связанные с их изготовлением и эксплуатацией, вплоть до отдельных компонентов материалов. Современные достижения в области цифровых и сенсорных технологий позволяют объектам сосуществовать вместе со своими физическими аналогами [4, 5].

В основном, цифровые двойники используются при оценке производительности сложных инженерных систем или их компонентов, где основное внимание уделяется точному описанию макромасштабной геометрии и показателей производительности на уровне компонентов. Например, существует быстрый способ создания цифрового двойника композитного материала, учитывающего диаметр, ориентацию и длину волокна. Происходит расчет механических свойств и сравнение результатов с данными, полученными путем микроскопии. Но такие решения из-за высокой дисперсии свойств образцов могут стать очень сложными, что может индуцировать нарушения или приводить к снижению пропускной способности и росту затрат. Эту проблему можно решить, если принимать во внимание физическое взаимодействие между образцами и системами обработки материалов [6, 7].

Цифровые двойники в индустрии на данный момент не используют в полной мере описание и архивирование данных о материалах, которые связаны с явлениями, происходящими на уровне атомно-молекулярных систем. Этот разрыв объясним разрозненным характером текущих исследований свойств материалов и свойств химических соединений и устоями в коммуникации сообществ по разработке продуктов. Тем не менее, очевидно, что успешное расширение цифровых двойников включением информации о свойствах материалов ведет к целостному проектированию производственного процесса в полностью интегрированных инновационных циклах.

Программные средства, работающие с цифровыми двойниками, должны с необходимой точностью воспроизводить свойства, изменения параметров, в состояниях, взаимодействиях, трансформации и деградации, и их последствия для оригинала и иметь возможность синхронизации с реальным объектом.

По мнению основателя направления "Цифровые двойники" (ЦД-ДТ) Майкла Гривса, профессора Мичиганского университета, составными частями ЦД (2002г.) являются:

- Физический продукт в реальном пространстве;
- виртуальный продукт в виртуальном пространстве;
- данные и информация, которые объединяют виртуальный и физический продукт.

К настоящему времени представления о структуре и функциональности ЦД существенно расширились. В составе ЦД могут (должны) присутствовать в целях задач, решаемых в сфере материаловедения) присутствовать:

- генераторы новых данных (приборные и результаты экспериментов и исследований приборные, аналитические);
- архивы предыдущих исследований (приведенные к текущим параметрам, форматам и шаблонам);
- архивы сторонних исследований (приведенные к текущим параметрам, форматам и шаблонам);
- аппаратно - программные средства агрегации массивов новых (текущих) данных конкретной проблемной области;
- аппаратно - программные средства агрегации массивов новых (текущих) данных нескольких проблемных областей;
- аппаратно - программные средства агрегации массивов новых (текущих) данных с архивными данными;
- теоретические модели (и средства работы с ними) исследуемых веществ, процессов и т.п.,
- физические и виртуальные средства разработки (генерации) новых моделей и модернизации предыдущих;

- средства получения данных от всех приборов и устройств лаборатории, их переформатирования, агрегации и передачи на сервера лаборатории (в случае закрытых исследований) или в облако, суперкомпьютер и иные средства высокопроизводительных датацентров.
- системы защиты информационной среды лаборатории и каналов связи;
- разнообразные программные средства моделирования, анализа как локальных наборов данных, так и модели и аппаратные средства работы в пространствах BigData и DataMining;
- современные математические пакеты анализа и прогнозирования свойств веществ;
- технические и программные средства и системы визуализации результатов исследований, трансформации визуальных представлений и их прогнозирования.

1.1 Обзор аналогов

На рынке программного обеспечения существует несколько сайтов, предоставляющих возможности работы с различными материалами и химическими соединениями.

Aflow.org — это онлайн-платформа для поиска и анализа материалов и химических соединений. Интерфейс системы представлен на рисунке 3. Ниже представлен обзор возможностей платформы:

- Поиск материалов: сайт позволяет искать материалы по различным параметрам, таким как химический состав, структура, свойства и т.д.;
- анализ свойств материалов: на сайте можно производить анализ свойств материалов, таких как механические, электрические и оптические свойства;
- сравнение материалов: платформа позволяет сравнивать различные материалы по их свойствам и параметрам;

- интеграция с другими инструментами: сайт интегрирован с другими инструментами, такими как программы для моделирования и расчета свойств материалов;
- аналитика и отчеты: на сайте доступна подробная аналитика по свойствам материалов, а также возможность создания отчетов;
- поддержка множества языков: сайт поддерживает множество языков, что позволяет пользователям из разных стран использовать его для поиска и анализа материалов.

The screenshot displays the AFLOW website interface. At the top, there is a navigation menu with links for HOME, CONSORTIUM, SEMINARS, SCHOOLS, FORUM, and SRC. Below this is a search bar with the text "Search Aflow" and a dropdown menu currently set to "ICSD only". To the right of the search bar is a green "Search" button indicating "(60389 entries)" and a "Display" button. The main content area features a "Periodic Table" with various filters on the left and right. The filters include "Property Filters" and "Periodic Table" categories such as Metals, Alkali Metals, Alkali Earths, Transition Metals, Lanthanides, Other Metals, Non-Metals, Boron Group, Carbon Group, Pnictogens, Chalcogens, and Halogens. There are also logical operators like "and", "not", "or", and "xor", and a "# of species" filter. The periodic table itself shows elements with their symbols and names, with some elements highlighted in orange. At the bottom of the page, there is a footer with logos for Duke University, University at Buffalo, Missouri S&T, NC State University, Penn State, and the University of Sand Drenth.

Рисунок 3 – Интерфейс сайта aflow

Онлайн-платформа Crystallography.net является базой данных кристаллических структур, которая содержит информацию о более чем 400 000 соединений. Она предоставляет уникальный контент для ученых и

исследователей в области химии и кристаллографии. Интерфейс системы представлен на рисунке 4.

- Функционал сайта позволяет пользователям искать и просматривать информацию о кристаллических структурах, а также загружать данные для анализа. Сайт имеет удобный интерфейс и интуитивно понятную навигацию, что делает его использование простым и удобным;

- одной из главных особенностей сайта является высокая точность и полнота данных, которые представлены в базе данных. Это позволяет пользователям получить наиболее достоверную информацию о кристаллических структурах;

- сайт также предоставляет доступ к различным функциям и инструментам, которые помогают пользователям в анализе данных. Например, пользователи могут использовать инструменты для поиска, по ключевым словам, фильтрации результатов по различным параметрам и сравнения структурных данных;

- сайт имеет высокую скорость работы и стабильность, что позволяет пользователям быстро получать доступ к информации и работать с базой данных без задержек и ошибок;

- существует возможности интеграции с другими инструментами и программами, что упрощает работу с данными и повышает эффективность исследований;

- сайт предоставляет аналитику и отчеты, которые помогают пользователям в анализе данных и принятии решений. Это делает его одним из наиболее полезных и ценных ресурсов для ученых и исследователей в области химии и кристаллографии;

- имеет высокий уровень поддержки пользователей, что позволяет пользователям получать помощь и поддержку в случае возникновения проблем

или вопросов. Это делает его надежным и удобным ресурсом для работы с кристаллическими структурами.

Information card for entry 7131035

7131034 << 7131035 >> 7131036

Preview

Unit cell parameters:
a=18.207Å
b=17.216Å
c=14.288Å
α=90.000°
β=121.068°
γ=90.000°

Coordinates: [7131035.cif](#)
Original paper (by DOI): [HTML](#)

Structure parameters

Common name	210610_FAK_KRD452
Formula	C40 H46 Ag B F4 N4
Calculated formula	C40 H46 Ag B F4 N4
Title of publication	An optically reversible room-temperature solid-state cobalt(II) photoswitch based on nitro-to-nitrito linkage isomerism
Authors of publication	Derez, Krystyna A.; Kamiński, Radosław; Kutnievska, Sylvia E.; Króweczyński, Adam; Schaniel, Dominik; Jarzemska, Katarzyna N.
Journal of publication	Chemical Communications
Year of publication	2022
Journal volume	58

Рисунок 4 – Интерфейс сайта <https://www.crystallography.net/>

NOMAD - является базой данных материалов, которая содержит информацию о более чем 100 000 структурных материалах. Она предоставляет уникальный функционал для исследования свойств материалов, в том числе кристаллических структур, и позволяет пользователям создавать и сохранять свои проекты исследований. Интерфейс системы представлен на рисунке 5.

– Одной из главных особенностей сайта является его интерактивный инструментальный, который позволяет пользователям искать и анализировать данные о материалах на основе различных параметров, таких как химический состав, структура, свойства и т.д;

– на сайте также доступны инструменты для расчета энергии связи, теплоемкости, магнитных свойств и других характеристик материалов;

– имеется удобный и понятный интерфейс, который позволяет пользователям легко найти нужную информацию. Кроме того, на сайте есть разделы с обучающими материалами, которые помогают пользователям лучше понимать основы кристаллографии и химии;

– бесплатность и открытость. Все данные на сайте доступны для свободного использования, что делает его идеальным инструментом для научных исследований в области материаловедения и кристаллографии.

The screenshot displays the NOMAD Encyclopedia website. At the top, there is a navigation bar with the NOMAD logo and 'The NOMAD Laboratory' on the left, and 'Guest (LOGIN)' with a user icon on the right. Below the navigation bar, the page title 'NOMAD Encyclopedia' is visible on the left, and 'Additional information' with a toggle switch is on the right. The main content area is divided into two columns. The left column contains filters for 'Structure' (System type: Bulk, 2D, 1D; Crystal system: Cubic, Hexagonal, Trigonal, Tetragonal, Orthorhombic, Monoclinic, Triclinic; Space group number: input field; Structure type: search field) and 'Properties' (Band gap (eV): min - max; Results containing...: Band structure, DOS, Thermal properties). The right column features a 'Composition' search bar with 'Clear' and 'Search' buttons. Below this is a search filter section with 'Element', 'Formula', and 'Name' tabs, and logical operators 'OR', 'AND', 'NOT', and parentheses. The central part of the page is a periodic table where elements are color-coded by groups: Alkali metals (orange), Alkaline earth metals (light orange), Transition metals (yellow), Post-transition metals (light yellow), Metalloids (pale yellow), Other nonmetals (light green), Halogens (green), Noble gases (light blue), Lanthanoids (blue), and Actinoids (dark blue). A legend at the bottom of the table lists these categories. Below the periodic table, there is a checkbox for 'Allow other elements' and a search result summary: 'Results (total: 2921785)' with navigation links '< prev page 1 / 116872 next >'. At the very bottom, there are labels for 'Space group', 'Space gr. int. symbol', 'Structure type', and 'N° calculations'.

Рисунок 5 – Интерфейс сайта <https://nomad-lab.eu/prod/rae/encyclopedia>

1.2 Анализ основных технологических решений

Основными технологиями, на которых строится клиентская часть любого веб-сайта, являются HTML (HyperText Markup Language) в качестве языка разметки, CSS (Cascading Style Sheets) в качестве языка стилей и JavaScript в качестве языка программирования. На данный момент нет других технологий, которые могли бы заменить этот стек технологий в области frontend-разработки. Такая ситуация обусловлена особенностями работы веб-браузеров, которые могут работать только с этими технологиями.

Backend-разработка веб-сайтов может осуществляться с помощью множества ЯП, включая PHP, Python, Ruby, C#, Golang, JS и другие. В то же время, для frontend-разработки доступны только те ЯП, которые транслируются в JavaScript, например, TypeScript и CoffeeScript.

TypeScript — это язык программирования, который представляет собой надстройку над JavaScript, добавляющую статическую типизацию и другие возможности. Браузеры могут выполнять только код, написанный на JavaScript, поэтому написанный на TypeScript код, нужно транслировать в код на JavaScript, для этого используются различные инструменты сборки, один из самых популярных из них – это Vite, его описание будет представлено ниже. Использование TypeScript в проектах имеет следующие преимущества:

- Более безопасный код: статическая типизация позволяет обнаруживать ошибки на этапе компиляции, что уменьшает количество ошибок во время выполнения программы;
- улучшенная поддержка IDE: благодаря статической типизации и другим возможностям TypeScript, IDE может предоставить более точные подсказки и автодополнения;
- улучшенная читаемость кода: использование типов делает код более понятным и легко читаемым;

- более легкая поддержка кода: благодаря тому, что Typescript компилируется в JavaScript, его код может быть запущен на любой платформе, поддерживающей JavaScript;

- лучшая масштабируемость: Typescript позволяет создавать крупные проекты с множеством разработчиков, что облегчает совместную работу и поддержку кода;

- большое сообщество разработчиков: Typescript получил широкую поддержку от сообщества разработчиков, что обеспечивает доступность множества библиотек и инструментов для работы с ним.

Vite — это инструмент для сборки Web-приложений. Он позволяет объединять и упаковывать различные модули и зависимости в единую файловую структуру, которая может быть использована в браузере. Преимущества использования Vite [8]:

- Улучшенная производительность: Vite позволяет минимизировать количество запросов к серверу и уменьшить размер файлов, что ускоряет загрузку страницы;

- удобство работы с модулями: Vite позволяет использовать модульную систему CommonJS или ES6, что облегчает организацию кода и делает его более понятным;

- расширяемость: Vite имеет большое количество плагинов и лоадеров, которые позволяют расширять его функциональность и адаптировать под конкретные нужды проекта;

- кроссплатформенность: Vite может быть использован на любой платформе, поддерживающей JavaScript;

- поддержка современных технологий: Vite поддерживает использование современных технологий, таких как TypeScript, React, Vue.js, Angular и другие;

– легкость в использовании: Vite имеет простой и интуитивно понятный интерфейс, что делает его использование легким и удобным.

В сфере frontend-разработки имеется множество фреймворков и библиотек. Согласно опросу более чем 3500 профессиональных разработчиков о фреймворках [9] и ежегодному опросу среди разработчиков на stackoverflow [10], React от Facebook является лидером по популярности среди разработчиков. Также стоит обратить внимание на Angular и Vue, которые также пользуются большой популярностью у разработчиков. При изучении этих фреймворков следует обратить внимание на их преимущества и недостатки.

React — это библиотека JavaScript, которая была создана Facebook в 2013 году и предназначена для разработки интерфейсов веб-приложений. Она имеет множество преимуществ, таких как быстрый рендеринг интерфейсов благодаря React Virtual DOM, поддержка Progressive Web App (PWA), поддержка TypeScript, легкий синтаксис и большое сообщество разработчиков.

Однако у нее есть и недостатки, такие как отсутствие навязанной архитектуры кода, отсутствие единого стиля написания CSS-кода и смешивание логики приложений и их представлений, что может негативно сказаться на архитектуре [10].

Angular является одним из старейших JS-фреймворков, созданным в 2009 году. На сегодняшний день он полностью переписан на TypeScript и поддерживается компанией Google. Angular предоставляет широкий спектр инструментов для создания интерактивных веб-приложений.

Среди достоинств использования Angular следует отметить его поддержку TypeScript, благодаря инструментам, специально разработанным для работы с этим языком. Архитектура фреймворка оптимизирована для разработки на TypeScript и предназначена для создания больших, масштабируемых приложений.

Однако, использование Angular имеет некоторые недостатки, включая длительное время рендеринга интерфейсов по сравнению с другими фреймворками, одностороннюю привязку данных, обеспечивающую минимальный риск ошибок, связанных с изменением данных, высокое потребление оперативной памяти по сравнению с другими фреймворками, сложный синтаксис, связанный с TypeScript, а не с JavaScript, что может сделать порог входа выше, чем у других фреймворков, а также проблемы с миграцией при переходе с более старых версий на новые [11].

Vue — это фреймворк на языке JavaScript, который разработан для создания веб-интерфейсов. Он является проектом с открытым исходным кодом и появился на свет позже Angular и React. Создатели Vue постарались унаследовать лучшие качества обоих фреймворков и избавиться от их недостатков [12].

Vue обладает рядом преимуществ:

- Постоянно обновляемая документация, содержащая примеры использования;
 - использование виртуального DOM (как в React), что значительно повышает скорость рендеринга интерфейсов;
 - концепция однофайловых компонентов упрощает архитектуру приложений и переиспользование кода;
 - двусторонняя привязка данных, позволяющая легко и быстро изменять состояние интерфейсов веб-приложений;
 - малое использование памяти, по сравнению с другими фреймворками.
- Vue может использовать всего 20 КБ и при этом сохранять скорость и гибкость, что заметно улучшает производительность.

Тем не менее, у Vue есть и свои недостатки [12]:

- Проблемы с интеграцией в больших проектах;

- меньшее количество библиотек и инструментов для разработки больших проектов по сравнению с React и Angular.

Для выполнения проекта мы выберем фреймворк Vue, так как он обладает рядом ключевых преимуществ:

- В отличие от Angular, Vue обладает реактивностью, которая облегчает изменение состояния интерфейсов. Достаточно изменить данные в коде, и изменения автоматически отобразятся в интерфейсе, без необходимости постоянного вызова методов рендеринга при каждом изменении данных;

- в отличие от React, у Vue есть навязанная архитектура, которая упрощает расширение и переиспользование кода, делая его более понятным.

Вывод по разделу один.

Суть решаемой задачи: создать веб-сайт для поиска цифровых двойников материаловедения для баз данных на основе секвенированных файлов.

Для разработки системы будем использовать следующие технологии: HTML – для создания структуры страниц веб-сайта, CSS – для изменения внешнего вида веб-сайта, TypeScript – для программной логики веб-сайта, Vue – для создания архитектуры и простой работы с интерфейсами веб-сайта, Vite – для конфигурирования сборки проекта.

2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

2.1 Функциональные требования

На этапе проектирования системы были выявлены следующие функциональные требования:

- Возможность поиска материалов по различным химическим критериям (формула, название, свойства и т.д.);
- отображение информации о найденном материале;
- возможность визуализации информации о материале;
- фильтрация результатов поиска по различным параметрам (например, свойствам материала);
- возможность скачивать файлы из озера данных;
- возможность загружать свои файлы в озеро данных.

2.2 Нефункциональные требования

На этапе проектирования системы были выявлены следующие нефункциональные требования:

- Доступность веб-сайта круглые сутки;
- интуитивно понятный интерфейс;
- поддержка функционала на следующих веб-браузерах и операционных системах: Веб-браузеры: Edge (версия 16 и выше), Firefox (версия 52 и выше), Chrome (версия 57 и выше), Safari (версия 10.1 и выше), Opera (версия 44 и выше). Использование других веб-браузеров может привести к неправильному отображению функционала или его неполной поддержке. Операционные системы: Windows, Linux, Mac OS. Различия между операционными системами не должны влиять на работу системы, поскольку за это отвечают веб-браузеры.

Следовательно, система может работать на других операционных системах, если они поддерживают указанные выше веб-браузеры;

– оптимизация системы для работы на разных разрешениях и пропорциях экранов мониторов. Рекомендуемое разрешение для работы с системой - не менее 1280 x 720.

3 ПРОЕКТИРОВАНИЕ

3.1 Архитектура предлагаемого решения

В данной главе будет представлена архитектура разрабатываемого веб-приложения для поиска цифровых двойников материаловедения на основе секвенированных файлов. Основным фреймворком, используемым в разработке клиентской части, является Vue.js.

1. Общая архитектура

Веб-приложение представляет собой клиент-серверное решение. Клиентская часть реализована на Vue.js, а серверная - на Python с использованием фреймворка Flask. Серверная часть отвечает за обработку запросов, а клиентская - за отображение данных и взаимодействие с пользователем.

REST API (Representational State Transfer Application Programming Interface) — это стиль архитектуры программного обеспечения, который определяет правила для создания веб-сервисов. REST API позволяет клиентам обращаться к веб-серверу для получения или изменения ресурсов, используя протокол HTTP. Каждый ресурс идентифицируется уникальным URI, и клиенты могут выполнять операции чтения, записи, обновления и удаления, используя стандартные методы HTTP, такие как GET, POST, PUT и DELETE [13].

Одним из главных преимуществ REST API является его простота и универсальность. Он может использоваться для обмена данными между различными языками программирования, платформами и устройствами. Кроме того, REST API легко масштабируется и может обрабатывать большие объемы запросов. Еще одним преимуществом является возможность кэширования данных на стороне клиента, что позволяет уменьшить нагрузку на сервер и ускорить работу приложения. Интерфейс REST API модели представлен на рисунке 6.

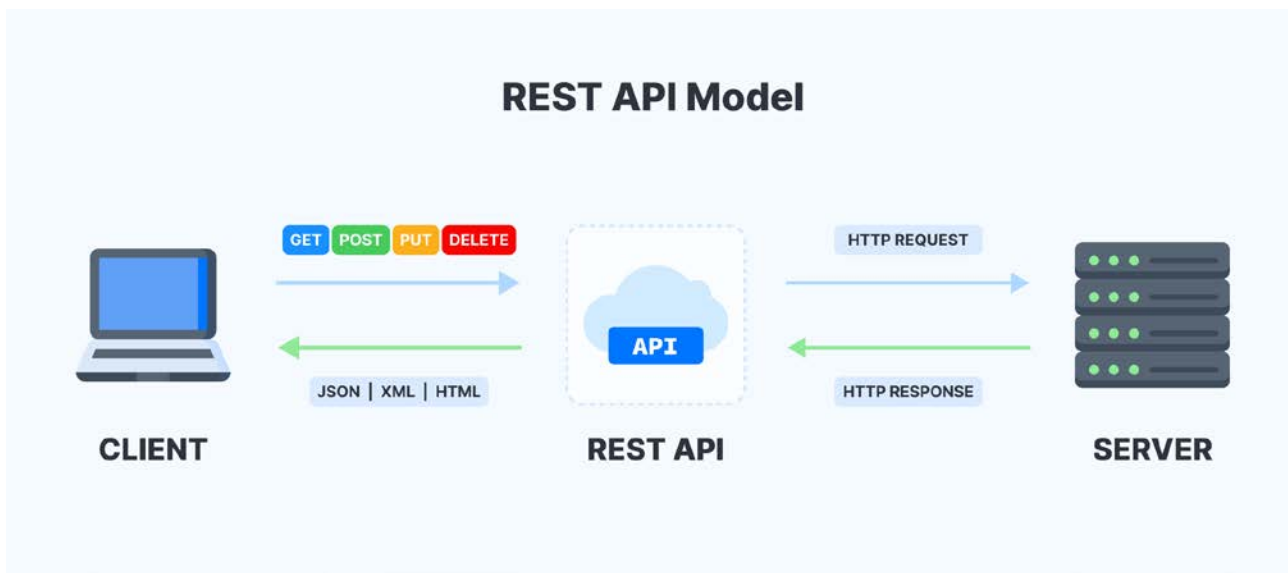


Рисунок 6 – REST API Модель

2. Архитектура клиентской части

В современном мире веб-разработки существуют различные подходы к созданию клиентской части веб-приложений, основные из них: SPA (Single Page Application), SSR (Server Side Rendering) и SSG (Static Site Generation) [14].

SPA — подход, при котором весь контент загружается один раз при запуске приложения, а затем обновляется динамически без перезагрузки страницы. Этот подход использует JavaScript для обработки пользовательских действий и обновления содержимого страницы. Он позволяет создавать более быстрые и отзывчивые приложения, так как не требуется перезагрузка страницы при каждом запросе.

SSR — подход, при котором страницы генерируются на сервере и отправляются на клиент для отображения. Этот подход используется, когда нужно обеспечить лучшую производительность для приложения, так как многие операции выполняются на сервере. Однако, SSR имеет свои ограничения, так как

он может быть медленнее, чем SPA, и требует дополнительной инфраструктуры на стороне сервера.

SSG — подход, при котором страницы создаются на стадии сборки и сохраняются в виде статических файлов. Этот подход используется, когда нужно обеспечить более быструю загрузку страниц и улучшить SEO. Однако, SSG не подходит для приложений, которые требуют динамического контента и частых обновлений.

При разработке проекта был выбран подход SPA, потому что это позволяет создавать более быстрые и отзывчивые приложения, которые не требуют перезагрузки страницы при каждом запросе. SPA также обеспечивает более простую и эффективную разработку, так как весь код приложения находится в одном месте и может быть повторно использован в различных частях приложения. Схема SPA подхода представлена на рисунке 7.

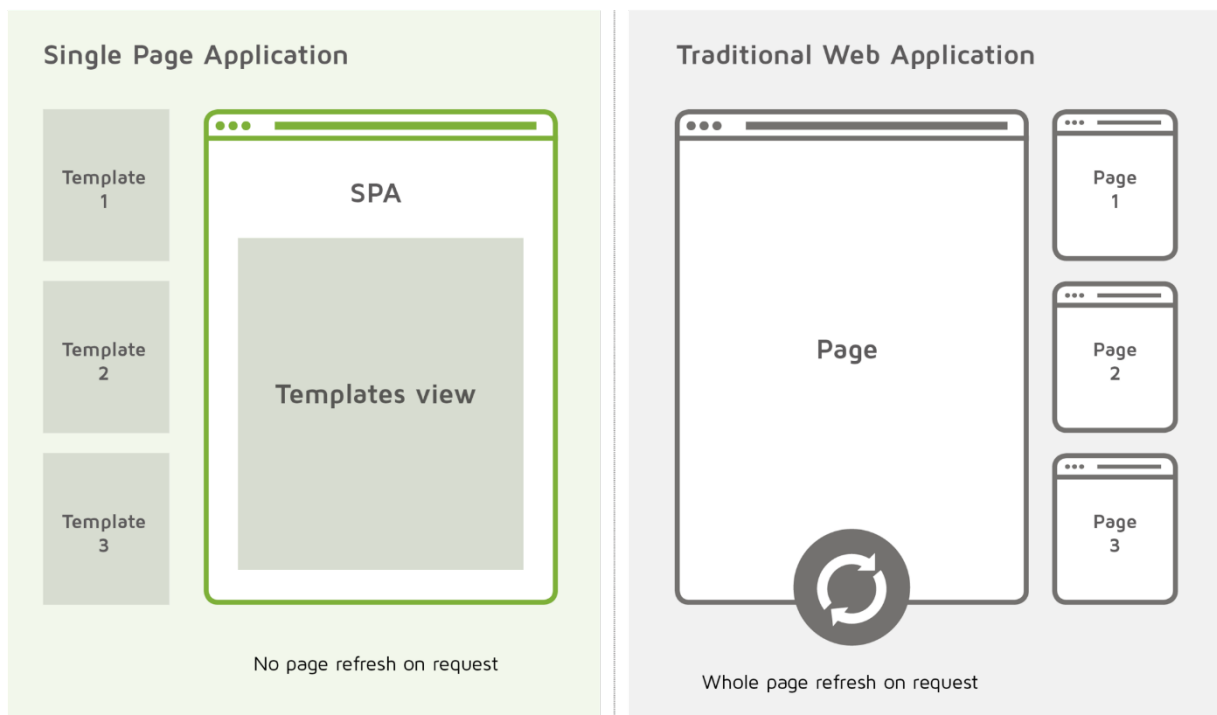


Рисунок 7 – SPA vs традиционный подходы

Клиентская часть веб-приложения состоит из компонентов, написанных на Vue.js, отвечающих за состояние и отображение клиентских интерфейсов.

- Компонент авторизации пользователя, открывающий доступ к функционалу системы;

- компонент поиска цифровых двойников. Отвечает за получение запросов от пользователя и отправку их на сервер для обработки. Результаты поиска отображаются в таблице;

- компонент таблицы результатов поиска. Отображает найденные цифровые двойники и позволяет пользователю выбирать нужный материал;

- компонент отображения выбранного цифрового двойника. Отображает детальную информацию о выбранном материале;

- компонент загрузки цифрового двойника. Позволяет загрузить с компьютера пользователя файлы, содержащие информацию о цифровом двойнике и отправить их на сервер.

3. Архитектура серверной части

Серверная часть приложения написана на Python с использованием фреймворка Flask. Она отвечает за обработку запросов от клиента, поиск цифровых двойников и взаимодействие с базой данных.

Основными модулями серверной части являются:

- Модуль обработки запросов. Получает запросы от клиента и передает их на обработку соответствующему модулю;

- модуль поиска цифровых двойников. Осуществляет поиск цифровых двойников на основе секвенированных CIF, OUT и других файлов;

- модуль взаимодействия с базой данных. Отвечает за сохранение и получение данных из базы данных.

3.2 Алгоритмы решения задачи

Каждый компонент отвечает за отображение отдельной части интерфейса приложения. Все компоненты связаны между собой и обмениваются данными через Vuex, взаимодействие с сервером организовано с помощью Axios.

Vuex — централизованное хранилище состояния для приложений, написанных на фреймворке Vue.js. Он решает множество задач, связанных с управлением состоянием приложения.

Все данные приложения хранятся в одном месте, что упрощает управление состоянием и предотвращает дублирование данных. Это позволяет избежать ошибок, связанных с несогласованными данными в разных частях приложения.

Vuex также позволяет легко отслеживать изменения состояния приложения и реагировать на них. Когда состояние приложения изменяется, Vuex автоматически обновляет все компоненты, которые зависят от этого состояния. Это упрощает разработку и уменьшает количество ошибок.

Другим важным достоинством Vuex является возможность использования модулей. Модули позволяют разбить хранилище на более мелкие части, что упрощает управление состоянием приложения и делает код более организованным. Схема работы Vuex представлена на рисунке 8.

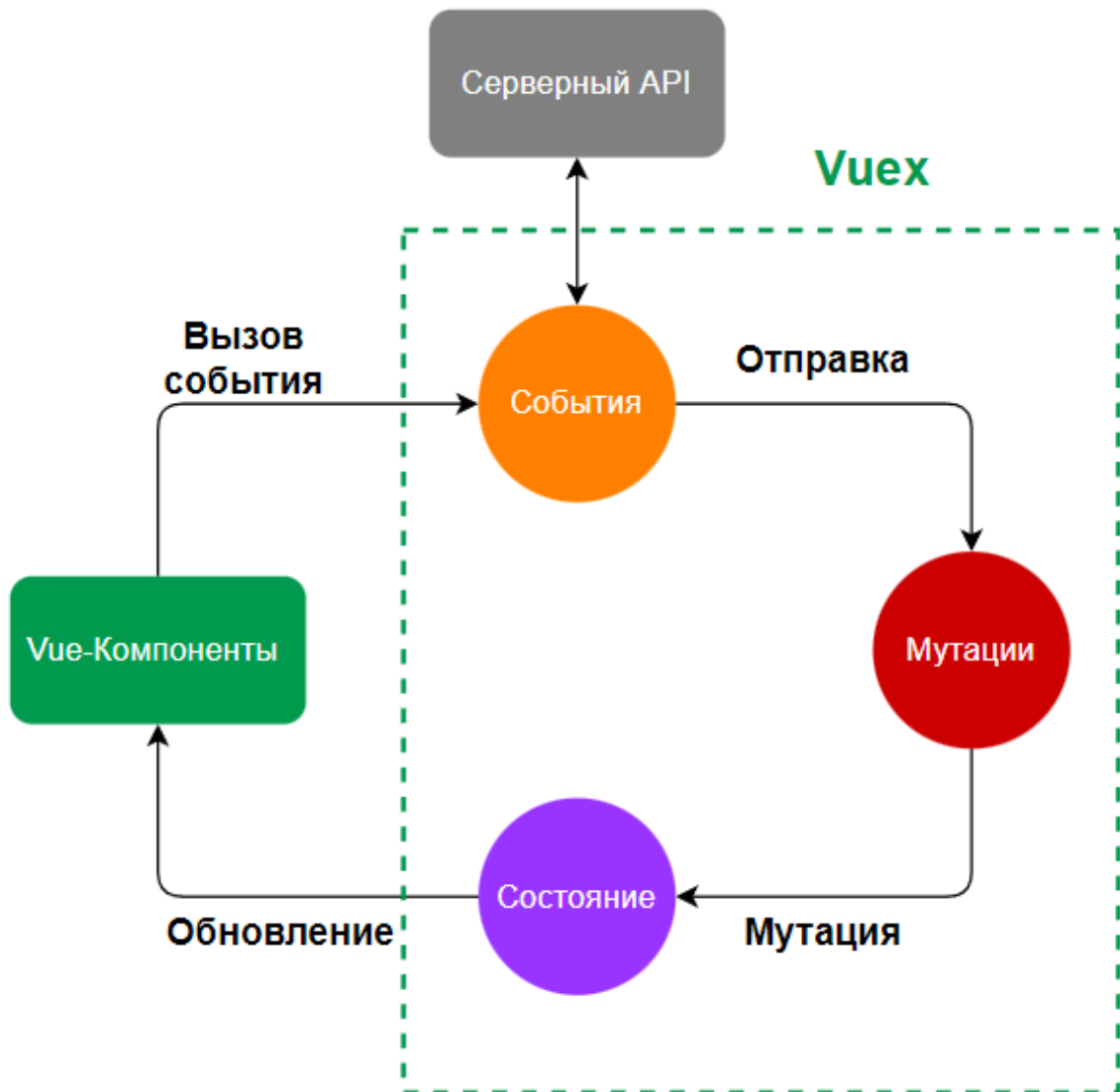


Рисунок 8 – Архитектура Vuex

Axios — это библиотека JavaScript для выполнения HTTP-запросов из браузера или Node.js. Axios предоставляет простой и удобный API для выполнения запросов на сервер и обработки ответов. Axios поддерживает все основные HTTP-методы, включая GET, POST, PUT и DELETE, а также может отправлять запросы с параметрами, заголовками и телом запроса.

Использование Vue.js, Vuex, Vite и Axios позволяет создавать масштабируемые и высокопроизводительные веб-приложения.

Основной компонентом в системе является компонент, реализующий интерфейс для поиска и работы с химическими файлами, например, CIF и OUT. Главной задачей было сделать этот компонент, расширяемым, так чтобы он мог работать как с CIF, так и с OUT файлами, а в будущем и с другими форматами. Благодаря использованию концепции параметрического полиморфизма получилось сделать компонент максимально переиспользуемым. Примеры того, как выглядят реализации компонентов для работы с CIF и OUT файлами представлены на листингах 1 и 2 соответственно.

Листинг 1 – Код Vue-компонента по работе с CIF-файлами

```
<template>
  <file-type-component :file-type="fileTypes.CIF "></file-type-component>
</template>

<script setup lang="ts">
import {fileTypes} from "@/types/enums";
</script>
```

Листинг 2 – Код Vue-компонента по работе с OUT-файлами

```
<template>
  <file-type-component :file-type="fileTypes.OUT "></file-type-component>
</template>

<script setup lang="ts">
import {fileTypes} from "@/types/enums";
</script>
```

Как видно из листингов, вся логика по работе с различными материалами инкапсулирована в компоненте `FileTypeComponent.vue`, далее в этот компонент подаётся тип файла, в данном случае CIF или OUT и уже внутри компонента реализуются соответствующие входному типу файла, отображение интерфейса и сетевые запросы.

Пример одного из таких разрешений в зависимости от того, с какими файлами идет работа представлен на листинге 3.

Листинг 3 – Код FileTypeComponent.Vue, реализующий скачивание файла

```
const handleUpload = () => {
  fileUploading.value = true

  const formData = new FormData()
  fileList.value.forEach((file, id) => {
    formData.append('files', file.raw)
  })

  Api[fileTypeName].uploadFile(formData)
  .then(response => {
    fileList.value.forEach((file, id) => {
      if (response.data[file.name].status === 'successfully') {
        file.status = 'success'
      } else {
        file.status = 'fail'
      }
    })
  })
  .catch(error => {
    console.error(error)
  })
  .finally(() => {
    fileUploading.value = false
  })
}
```

В зависимости от – fileTypeName, переменной, хранящей информацию, о том, с каким документом идет работа, берётся соответствующий API и вызывается нужный метод, отправляющий сетевой запрос с загруженными файлами.

3.3 Описание данных

Входными данными для системы являются действия пользователя, направленные на взаимодействие с интерфейсом системы, один из примеров таких взаимодействий, это ввод поисковых параметров в формы на рисунке 9.

Поиск CIF-файлов

id

Поле

filename

Поле

content

Режим поиска Частичный CIF Полный CIF

Рисунок 9 – Ввод параметров поиска

Также входными данными для системы являются файлы, которые пользователь может загружать в систему и отправлять на сервер для хранения. На данный момент это текстовые форматы .cif и .out. CIF и OUT файлы в химии — это форматы файлов, используемые для хранения данных о кристаллах.

В CIF (Crystallographic Information File) файлах хранятся данные о структуре, кристаллической решетке, атомных координатах, термодинамических параметрах и других свойствах кристалла. Пример содержимого CIF файла представлен на рисунке 10.


```

#####
#####
#
#           Cambridge Crystallographic Data Centre
#                   CCDC
#
#####
#####
#
# If this CIF has been generated from an entry in the
Cambridge
# Structural Database, then it will include bibliographic,
chemical,
# crystal, experimental, refinement or atomic coordinate data
resulting
# from the CCDC's data processing and validation procedures.
#
#####
#####

data_CSD_CIF_BAPREQ
_audit_creation_date 2003-09-24
_audit_creation_method CSD-ConQuest-V1
_database_code_CSD BAPREQ
_database_code_depnum_ccdc_archive 'CCDC 175853'
_chemical_formula_sum 'C20 H36 N12 O8 S4'
_chemical_formula_moiety
;
C12 H24 O6,2(C4 H6 N6 O1 S2)
;
_journal_codен_Cambridge 119
_journal_volume 647
_journal_year 2003
_journal_page_first 129
_journal_name_full 'J.Mol.Struct. '
loop_
_publ_author_name
"M.S.Fonari"
"Yu.A.Simonov"
"V.Ch.Kravtsov"
"J.Lipkowski"
"E.V.Ganin"
"A.A.Yavolovskii"
_chemical_name_systematic
;
1,4,7,10,13,16-Hexaoxacyclooctadecane
bis(1-((4-amino-1,2,5-thiadiazol-3-yl) carbonyl)
thiosemicarbazide) clathrate

```

Рисунок 10 – Содержимое CIF файла

В OUT файле хранятся результаты расчетов по кристаллической структуре. Формат файла OUT зависит от программы, используемой для расчета. OUT файл может содержать результаты расчета энергии, оптимизации геометрии и других

параметров, а также различную вспомогательную информацию, касающегося оборудования и системы. Пример содержимого OUT файла представлен на рисунке 11.

```

*****
*****
*** K POINTS COORDINATES (OBLIQUE COORDINATES IN UNITS OF IS =
4)
  1-R( 0 0 0)  2-C( 1 0 0)  3-R( 2 0 0)  4-C( 0 1
0)
  5-C( 1 1 0)  6-C( 2 1 0)  7-R( 0 2 0)  8-C( 1 2
0)
  9-R( 2 2 0) 10-C( 0 0 1) 11-C( 1 0 1) 12-C( 2 0
1)
 13-C( 3 0 1) 14-C( 0 1 1) 15-C( 1 1 1) 16-C( 2 1
1)
 17-C( 3 1 1) 18-C( 0 2 1) 19-C( 1 2 1) 20-C( 2 2
1)
 21-C( 3 2 1) 22-R( 0 0 2) 23-C( 1 0 2) 24-R( 2 0
2)
 25-C( 0 1 2) 26-C( 1 1 2) 27-C( 2 1 2) 28-R( 0 2
2)
 29-C( 1 2 2) 30-R( 2 2 2)

  DIRECT LATTICE VECTORS COMPON. (A.U.)      RECIP. LATTICE VECTORS
  COMPON. (A.U.)
      X          Y          Z          X          Y
Z
  18.4588449    0.0000000    0.0000000    0.3403889
0.0000000    0.0695004
  0.0000000    20.7567519    0.0000000    -0.0000000
0.3027056    -0.0000000
  -5.4056440    0.0000000    26.4749681    0.0000000
0.0000000    0.2373255

  DISK SPACE FOR EIGENVECTORS (FTN 10)      118879488 REALS

  SYMMETRY ADAPTION OF THE BLOCH FUNCTIONS ENABLED
  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT gordsh1      TELAPSE      5.07
TCPU      0.16

  MATRIX SIZE: P(G) 2051552, F(G) 355280, P(G) IRR 258582, F(G)
IRR 89505
  MAX G-VECTOR INDEX FOR 1- AND 2-ELECTRON INTEGRALS 33

  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT INPUT      TELAPSE      5.36
TCPU      0.44

  NEIGHBORS OF THE NON-EQUIVALENT ATOMS

```

Рисунок 11 – Содержимое OUT файла

Промежуточные данные в системе представляют собой информацию, которая хранится на клиенте и подготавливается для отправки на сервер. Например, если пользователь назначил параметры поиска материала, или загрузил какие-нибудь файлы, информация об этом сохраняется на клиенте, подготавливается и затем отправляется на сервер для обработки. клиенте, которая затем отправляется на сервер для поиска материалов.

Выходные данные в системе представляют собой одностраничное приложение, состоящее из одной HTML-страницы, подключающей CSS и JavaScript сборки, содержащие весь код проекта, оптимизированный для доставки пользователю. Внутри реализована логика навигации между страницами приложения как в стандартных веб-приложениях, но в рамках одного HTML-файла, что позволяет открывать новые страницы без перезагрузки браузера, а с помощью динамически загружаемой верстки страницы.

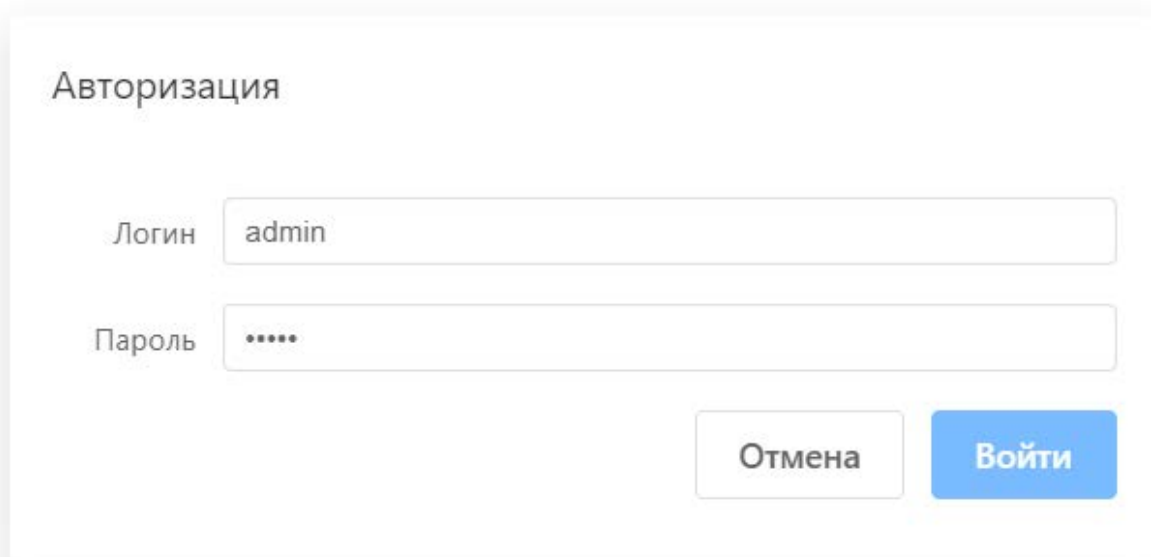
Список страниц в приложении:

- Страница авторизации пользователя;
- страница работы с OUT файлами;
- страница работы с CIF файлами.

4 РЕАЛИЗАЦИЯ

В данной главе рассмотрены реализация интерфейсов и основные достижения, полученные в результате выполнения выпускной квалификационной работы.

Первый интерфейс, с которым сталкивается пользователь, это интерфейс авторизации в системе. Здесь пользователь вводит логин и пароль, если они корректны, то пользователь получает доступ к функционалу системы. Интерфейс авторизации пользователя представлен на рисунке 12.



Авторизация

Логин admin

Пароль *****

Отмена Войти

Рисунок 12 – Окно авторизации пользователя

В случае, если пользователь попытается ввести неверные данные для входа в систему, пользователь не пройдет авторизацию и не получит доступ к системе, это продемонстрировано на рисунке 13.



Авторизация

Логин

Пароль

Рисунок 13 – Отказ в доступе к функционалу системы

Далее пользователю становятся доступны странице по работе с CIF и OUT файлами. Интерфейсы у них выглядят одинаково, отличаются только данные на странице. Поэтому рассмотрим интерфейсы на примере работы с CIF файлами. Интерфейс с начальным состоянием страницы для работы с CIF файлами представлен на рисунке 14.

Materials Data

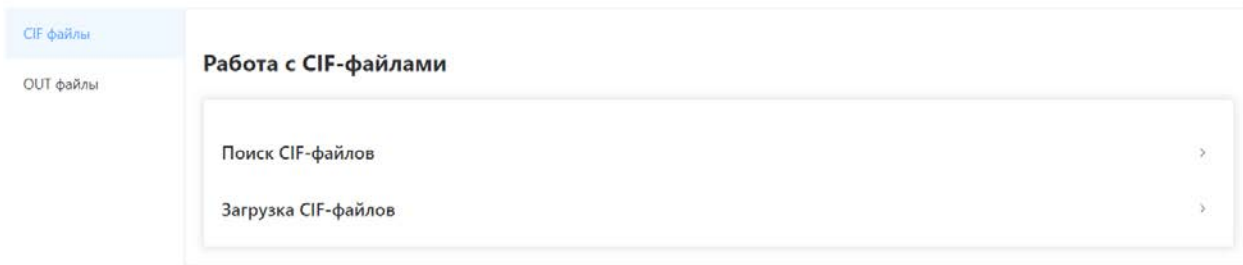


Рисунок 14 – Начальное состояние страницы по работе с CIF-файлами

Интерфейс страницы содержит набор вложенных и скрытых интерфейсов. Раскрытие вкладки “Поиск CIF-файлов” открывает форму для создания поискового запроса к озеру данных материаловедения, как это продемонстрировано на рисунке 15.

Поиск CIF-файлов

- id
- filename
- default_group
- submission_details
- processing_summary
- title_and_author
- text
- chemical_data
- refinement_data
- atomic_coordinates_and_displacement_parameters
- molecular_geometry
- visual_data
- atomic_coordinates
- unknown_group

Режим поиска Частичный CIF Полный CIF

Поиск

Рисунок 15 – Форма создания поискового запроса

Поиск ведется по различным группам секвенированного файла. Можно создавать комбинированный запрос из нескольких параметров, таким образом отфильтровать нужные файлы по этим параметрам, как это представлено на рисунке 16.

visual_data ▼

Поле

atomic_coordinates ▼

Поле

unknown_group ▼

Поле

Режим поиска Частичный CIF Полный CIF 🔍 Поиск

Рисунок 16 – Комбинированный поиск по различным критериям

Результаты поиска отображаются в виде таблицы. Также присутствует информация о времени выполнения запроса к озеру данных и общее количество результатов поиска, этот интерфейс представлен на рисунке 17.

id	filename	default_group	submission_details	processing_summary	title_and_author	text	chemical_data	Дополнительно
c2b7074f-5b b1-47d3-975 2-4ba77efbdd eb	BAPREQ.sea rch2.cif	[\"_audit_creati on_date\": \"20 03-09-24\"]	[\"Null\": 0]	[\"_journal_volum e\": \"647\", \"_j ournal_year\": \" 2003\", \"_jour nal_page_first\": \"129\", \"_jour nal_name_full\": \"J.Mol.St ruct.\"]	[\"Null\": 0]	[\"Null\": 0]	▶ Длинный текст	
0a457651-10 92-4cf0-a2c3 -69c940e381 d7	BAWHEM0 1search2.c if	[\"_audit_creati on_date\": \"20 05-12-21\"]	[\"Null\": 0]	[\"_journal_volum e\": \"44\", \"_j ournal_year\": \" 2005\", \"_jour nal_page_first\": \"7194\", \"_jour nal_name_full\": \"Inorg.Ch em.\"]	[\"Null\": 0]	[\"Null\": 0]	▶ Длинный текст	
266089a-8a 0b-4453-9a1 e-02a3e7f5 bdd	BEMBE8.sea rch2.cif	[\"_audit_creati on_date\": \"20 04-06-01\"]	[\"Null\": 0]	[\"_journal_volum e\": \"42\", \"_j ournal_year\": \" 2003\", \"_jour nal_page_first\": \"5863\", \"_jour nal_name_full\": \"Angew.C hem.Int.Ed.\"]	[\"Null\": 0]	[\"Null\": 0]	▶ Длинный текст	
41fe36d8-14 0f-4cd6-90d 8-fb0eacbd2 98a	BETHA201.s earch2.cif	[\"_audit_creati on_date\": \"20 02-10-24\"]	[\"Null\": 0]	[\"_journal_volum e\": \"66\", \"_j ournal_year\": \" 2001\", \"_jour nal_page_first\": \"8954\", \"_jour nal_name_full\": \"J.Org.Ch em.\"]	[\"Null\": 0]	[\"Null\": 0]	▶ Длинный текст	
40914d4a-db 66-4452-bcd c-f021dd44f ec	BAPREQ.sea rch2.cif	[\"_audit_creati on_date\": \"20 03-09-24\"]	[\"Null\": 0]	[\"_journal_volum e\": \"647\", \"_j ournal_year\": \" 2003\", \"_jour nal_page_first\": \"129\", \"_jour nal_name_full\": \"J.Mol.St ruct.\"]	[\"Null\": 0]	[\"Null\": 0]	▶ Длинный текст	

◀ 1 2 3 4 5 ▶

Рисунок 17 – Результаты поиска файлов

Пользователь предоставляется возможность загружать файлы материаловедения в озеро данных, чтобы в будущем с ними работать. Для этого существует интерфейс загрузки файлов, продемонстрированный на рисунке 18.

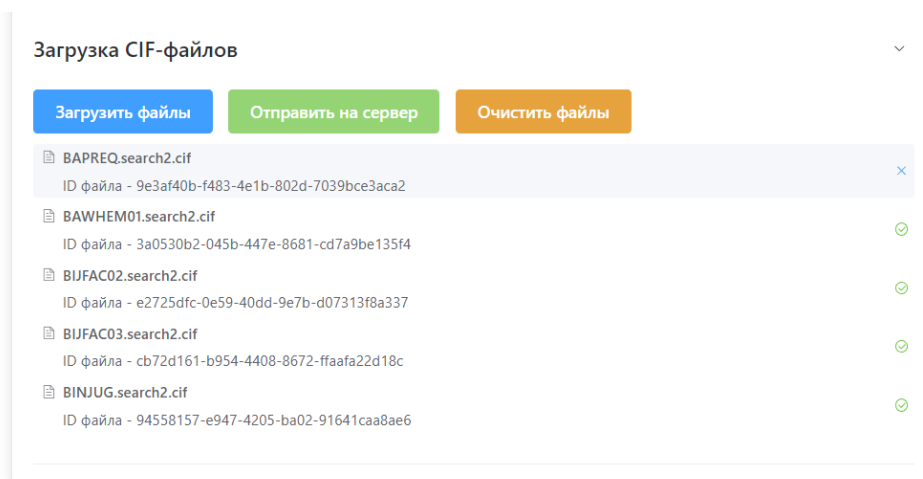


Рисунок 18 – Загрузка файлов на озеро данных

Также пользователь может скачивать файлы материаловедения на локальный компьютер пользователя по клику на кнопку в колонке “Дополнительно” напротив нужного файла, интерфейс этой работы представлен на рисунке 19.

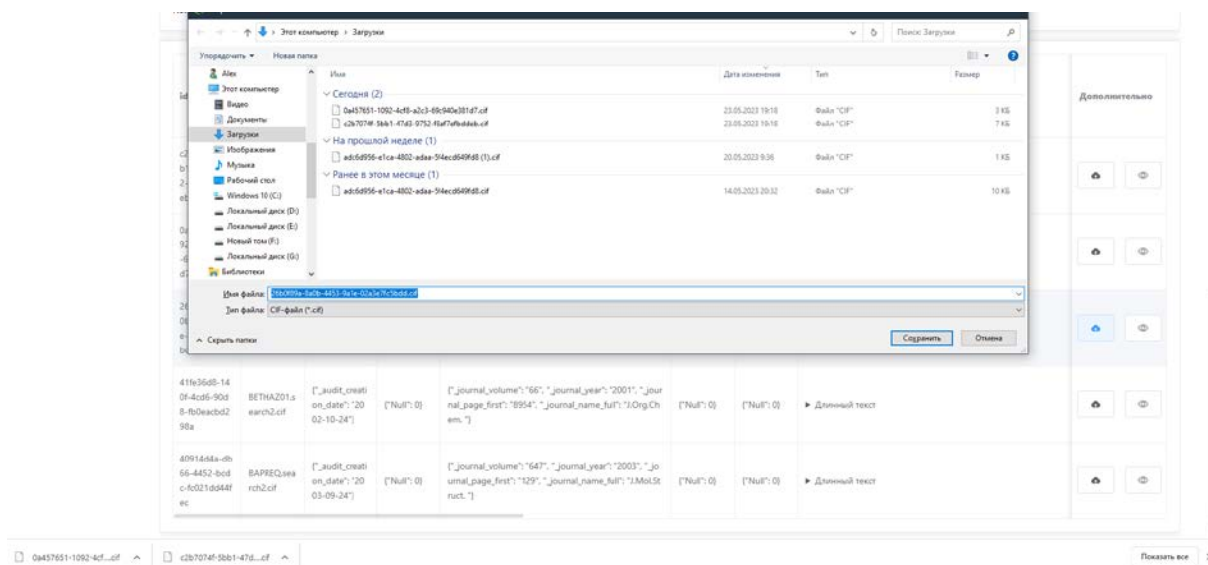


Рисунок 19 – Скачивание файлов материаловедения на компьютер пользователя

Помимо кнопки для скачивания файла в колонке “Дополнительно” есть кнопка визуализации SMILES кодов, соответствующих выбранному файлу.

SMILES (Simplified Molecular Input Line Entry System) — это строковое представление молекулы, которое используется в химических приложениях для хранения и передачи информации об атомах, связях и других химических свойствах молекулы. SMILES представляет собой одну строку символов ASCII, которая описывает топологию молекулы и связи между атомами. SMILES используется в многих химических приложениях, включая базы данных молекул, программы для построения молекул и виртуального скрининга, а также в системах распознавания химических структур.

Визуализация SMILES кодов

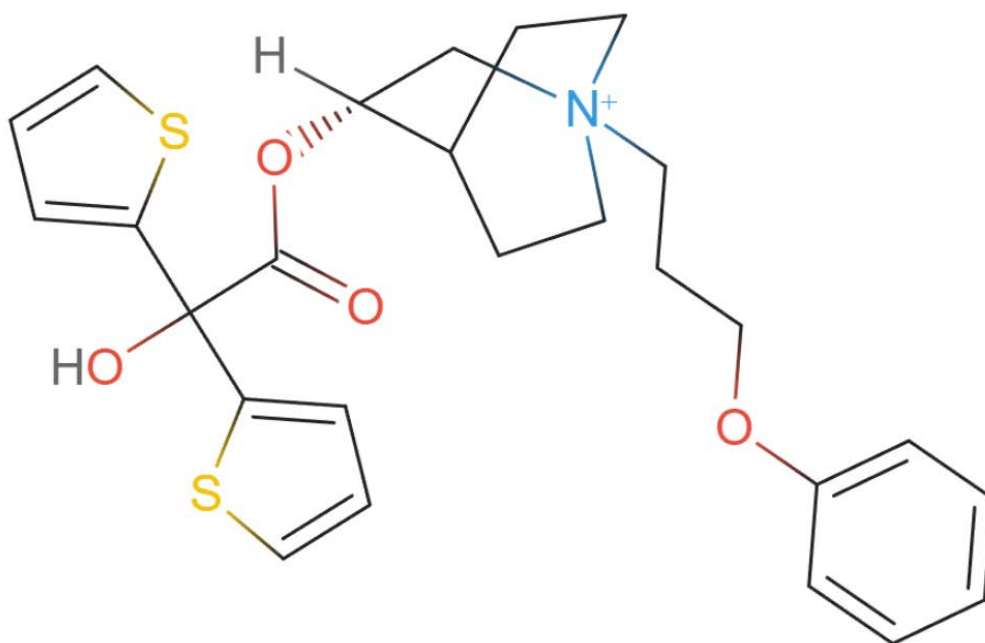


Рисунок 20 – Визуализация SMILES кодов

5 ТЕСТИРОВАНИЕ

5.1 Методология тестирования

На данный момент ПО отлажено, развернуто и сдано в эксплуатацию. В рамках тестирования было проведено альфа-тестирование продукта.

Альфа-тестирование — это форма приемочного тестирования, которое проводится на стадии разработки приложения. Оно направлено на выявление возможных проблем и ошибок в приложении до запуска его на рынке. Я и разработчик серверной части приложения провели альфа-тестирование нашего приложения чтобы убедиться в его качестве и функциональности, что оно соответствует требованиям и ожиданиям пользователей.

Во время выполнения квалификационной работы были использованы различные методологии тестирования ПО.

В рамках модульного тестирования была протестирована функциональность всех модулей системы, в рамках интеграционного тестирования была проверена корректность взаимодействия клиентской и серверной частей приложения.

5.2 Проведение процедуры тестирования

После завершения разработки приложения было проведено функциональное тестирование с целью проверки выполнения функциональных требований, то есть способности программного обеспечения в определенных условиях решать задачи пользователей [15]. Набор функциональных тестов представлен в таблице 1.

Таблица 1 – Тестирование системы

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
1	Неверная авторизация	1. Открыть систему. 2. Ввести неверные данные в поля авторизации.	Система должна выдать сообщение об ошибке авторизации и отказать в доступе к функционалу.	Да.
2	Верная авторизация	1. Открыть систему. 2. Ввести верные данные в поля авторизации.	Система должна страницу авторизации на страницу работы с материалами	Да.
3	Проверка появления поисковых групп	1. Открыть систему. 2. Открыть страницу по работе с материалами.	Система должна добавить поисковые группы для определенного файла на страницу	Да.
4	Проверка возможности загрузить файл материаловедения	1. Открыть систему. 2. Открыть страницу по работе с материалами. 3. В интерфейсе загрузки файлов, загрузить файлы и нажать кнопку отправки на сервер	Файлы должны отправиться файлы на сервер и сохраниться там.	Да.
5	Проверка работы поиска без поисковых параметров	1. Открыть систему. 2. Открыть страницу по работе с материалами. 3. Нажать кнопку поиска материалов	Система должна вывести все доступные в озере данных данные	Да.
6	Проверка работы поиска с поисковыми параметрами	1. Открыть систему. 2. Открыть страницу по работе с материалами. 3. Нажать кнопку поиска материалов.	Система должна вывести отфильтрованные по этим критериям данные	Да.
7	Проверка работы поиска с режимом поиска файлов в полном формате	1. Открыть систему. 2. Открыть страницу по работе с материалами. 3. Поменять режим поиска 4. Нажать кнопку поиска материалов.	Система должна поменять поисковые группы, а также вывести все доступные в озере данных данные в другом формате отображения	Да.

Продолжение таблицы 1

8	Проверка возможности скачать файлы с сервера	<ol style="list-style-type: none"> 1. Открыть систему. 2. Открыть страницу по работе с материалами. 3. Нажать кнопку поиска материалов. 4. Нажать кнопку скачать напротив одного из материалов. 	Браузер должен скачать выбранный файл на компьютер пользователя.	Да.
9	Проверка визуализации SMILES кодов	<ol style="list-style-type: none"> 1. Открыть систему. 2. Открыть страницу по работе с материалами. 3. Нажать кнопку поиска материалов. 4. Нажать кнопку визуализации напротив одного из материалов. 	Данные по файлу в таблице должны соответствовать визуальному отображению на странице	Да.

ЗАКЛЮЧЕНИЕ

В рамках данной работы была разработана клиентская часть информационной системы для решения поисковых задач материаловедения.

Были решены поставленные задачи, а именно:

- Произведен анализ предметной области;
- выбраны и обоснованы технологии реализации проекта;
- определены функциональные и нефункциональные требования к системе;
- выполнено проектирование системы;
- реализованы интерфейсы по поиску информации материаловедения;
- реализованы интерфейсы по скачиванию и загрузке файлов материаловедения;
- реализован интерфейс визуализации SMILES-кодов;
- проведены модульное и интеграционное тестирования.

В настоящее время клиентская часть собрана в Docker-контейнер и вместе с Docker-контейнером, содержащим серверную часть проекта, развернуты на сервере ЮУрГУ и доступна пользователям.

Архитектура приложения была создана с учетом продолжения разработки, сделана масштабируемой и конфигурируемой. В будущем можно будет продолжать разработку и добавление нового функционала.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Draxl, C., & Scheffler, M. NOMAD: The FAIR concept for big data-driven materials science. *MRS Bulletin* // Published online by Cambridge University Press Sept. 2018, p. 676-682.
2. Digital twin: the Age of Aquarius in Construction and Real Estate. [Электронный ресурс]. – URL: <https://www.constructionplacements.com/digital-twin-the-age-of-aquarius-in-construction-and-real-estate/> (дата обращения 16.03.2023).
3. Xiang, F. Digital twin driven green material optimal-selection towards sustainable manufacturing / F. Xiang // *Procedia Cirp*. – 2019. – P. 1290-1294. doi: 10.1016/j.procir.2019.04.015.
4. Mei, H. Recent Advances in Piezoelectric Wafer Active Sensors for Structural Health Monitoring Applications. / H. Mei, M. Haider, R. Joseph, A. Migot, V. Giurgiutiu // *Sensors*. – 2019. – Т.81. – P. 383. doi:10.3390/s19020383.
5. Ullo, S.L. Advances in Smart Environment Monitoring Systems Using IoT and Sensors. / S.L. Mei, G.R. Sinha // *Sensors*. – 2020. – Т.95. – P. 121-125. doi:10.3390/s20113113.
6. Constantin Bauer, Math2Market GmbH. The Digital Twin – A comparative study of material simulation on μ CT-scanned and modelled microstructures / The 4th International Congress on 3D Material Science, 2018.
7. Glatt M. Modeling and implementation of a digital twin of material flows based on physics simulation. / M. Glatt // *Journal of Manufacturing Systems*. – 2021. – Т. 58. – P. 231-245. doi: 10.1016/j.jmsy.2020.04.015.
8. Vite - Next Generation Frontend Tooling. [Электронный ресурс]. – URL: <https://vitejs.dev/> (дата обращения: 18.03.2023).
9. State of Frontend 2022. [Электронный ресурс]. – URL: <https://tsh.io/state-of-frontend/#developers> (дата обращения: 19.03.2023).

10. Web Frameworks. [Электронный ресурс]. – URL: <https://survey.stackoverflow.co/2022/#section-most-popular-technologies-web-frameworks-and-technologies> (дата обращения: 19.03.2023).
11. Quick Start. [Электронный ресурс]. – URL: <https://react.dev/learn> (дата обращения: 21.03.2023).
12. The web development framework for building the future. [Электронный ресурс]. – URL: <https://angular.io/> (дата обращения: 23.03.2023).
13. What is a REST API? [Электронный ресурс]. – URL: <https://www.ibm.com/topics/rest-apis> (дата обращения: 24.03.2023).
14. An approachable, performant and versatile framework for building web user interfaces. [Электронный ресурс]. – URL: <https://vuejs.org/> (дата обращения: 25.03.2023).
15. Бурбин, А.В. Тестирование программного обеспечения. Роль тестирования в методологиях разработки программного обеспечения. А.В. Мурзин // Инновации. Наука. Образование, 2021. – №. 43. – С. 1267–1273.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Скрипт инициализации приложения

```
<template>
  <el-container>
    <el-header height="100">
      <h1>Materials Data</h1>
    </el-header>
    <el-container>
      <el-aside width="200px" v-show="router.currentRoute.value.path !==
'/'">
        <side-bar-menu></side-bar-menu>
      </el-aside>
      <el-main>
        <router-view></router-view>
      </el-main>
    </el-container>
  </el-container>
</template>

<script setup lang="ts">
import SideBarMenu from "@/components/SideBarMenu.vue";
import { useRouter } from 'vue-router'

const router = useRouter()
</script>

<style scoped>

</style>
```

ПРИЛОЖЕНИЕ Б

Код конфигурации сборки приложения

```
import { defineConfig } from 'vite'
import vue from '@vitejs/plugin-vue'
import AutoImport from 'unplugin-auto-import/vite'
import Components from 'unplugin-vue-components/vite'
import { ElementPlusResolver } from 'unplugin-vue-components/resolvers'
import { fileURLToPath, URL } from 'url';

export default defineConfig({
  plugins: [
    vue(),
    AutoImport({
      resolvers: [ElementPlusResolver()],
    }),
    Components({
      resolvers: [ElementPlusResolver()],
    }),
  ],
  resolve: {
    alias: [
      { find: '@', replacement: fileURLToPath(new URL('./src', import.meta.url)) }
    ],
  },
})
```

ПРИЛОЖЕНИЕ В

Код конфигурации компиляции typescript

```
{
  "compilerOptions": {
    "target": "ESNext",
    "useDefineForClassFields": true,
    "module": "ESNext",
    "moduleResolution": "Node",
    "strict": true,
    "jsx": "preserve",
    "resolveJsonModule": true,
    "isolatedModules": true,
    "esModuleInterop": true,
    "lib": ["ESNext", "DOM"],
    "skipLibCheck": true,
    "noEmit": true,
    "baseUrl": ".",
    "allowJs": true,
    "checkJs": false,
    "paths": {
      "@/*": ["src/*"]
    }
  },
  "include": [
    "src/**/*.ts",
    "src/**/*.tsx",
    "src/**/*.vue"
  ],
  "exclude": [
    "node_modules",
    "dist"
  ],
  "references": [{ "path": "./tsconfig.node.json" }]
}
```

ПРИЛОЖЕНИЕ Г

Скрипт работы с файлами материаловедения

```
<template>
  <el-container direction="vertical" v-loading="groupsLoading">
    <el-card id="materials-card">
      <el-row>
        <h2>Работа с {{ fileTypeNames }}-файлами</h2>
      </el-row>
      <el-card>
        <el-row class="m-t-15">
          <el-collapse v-model="activeNames" class="collapse-container">
            <el-collapse-item name="Поиск">
              <template #title>
                <p class="collapse-header">
                  Поиск {{ fileTypeNames }}-файлов
                </p>
              </template>
              <search-component :structure="structure" />
            </el-collapse-item>
            <el-row :gutter="25" class="m-t-15" align="middle">
              <el-col :span="4">
                <h3>Режим поиска</h3>
              </el-col>
              <el-col :span="12">
                <el-radio-group v-model="selectedMode">
                  <el-radio size="large" label="divided">
                    Частичный {{ fileTypeNames }}
                  </el-radio>
                  <el-radio size="large" label="full" >
                    Полный {{ fileTypeNames }}
                  </el-radio>
                </el-radio-group>
              </el-col>
              <el-col :span="8" align="end">
                <el-button
                  type="primary"
                  :icon="Search"
                  size="large"
                  @click="getMaterialsData"
                >
                  Поиск
                </el-button>
              </el-col>
            </el-row>
          </el-collapse-item>
          <el-collapse-item name="Загрузка" class="m-t-15">
            <template #title>
              <p class="collapse-header">
                Загрузка {{ fileTypeNames }}-файлов
              </p>
            </template>
          </el-collapse-item>
        </el-card>
      </el-container>
```

```

    <el-upload
      v-model:file-list="fileList"
      ref="uploadRef"
      class="upload-demo m-t-15"
      multiple
      :auto-upload="false"
      :action="fileApiActionUrl"
      v-loading="fileUploading"
    >
      <template #default="{ fileList }">
        <div class="el-upload el-upload--text">
          <el-button type="success"
            @click="handleUpload" class="m-r-15">
            Отправить на сервер
          </el-button>
        </div>
        <div class="el-upload el-upload--text">
          <el-button type="warning" @click="clearFiles"
            class="m-r-15">
            Очистить файлы
          </el-button>
        </div>
      </template>
      <template #trigger>
        <el-button type="primary" class="m-r-
15">Загрузить файлы</el-button>
      </template>
    </el-upload>

    </el-collapse-item>
  </el-collapse>
</el-row>
</el-card>
<template v-if="resultsVisible">
  <el-row :gutter="25" class="m-t-15" align="middle">
    <el-col>
      Время выполнения запроса: {{executionTime}}
    </el-col>
  </el-row>
  <el-row :gutter="25" class="m-t-15" align="middle">
    <el-col>
      Количество результатов: {{resultsCount}}
    </el-col>
  </el-row>
</template>
</el-card>

<el-card
  class="m-t-15"
  v-loading="materialsDataLoading"
  v-if="materials.length > 0"
>
  <el-table :data="materials" border>
    <el-table-column
      v-for="key in groupsNames"

```

```

      :key="key"
      :prop="key"
      :label="key"
      :width="calculateSize(materials[0][key], 100, 800)"
      resizable
    >
    <template #default="scope">
      <template v-if="scope.row[key].length > 500">
        <details>
          <summary>Длинный текст</summary>
          {{scope.row[key]}}
        </details>
      </template>
      <template v-else>
        {{scope.row[key]}}
      </template>
    </template>
  </el-table-column>
  <el-table-column label="Дополнительно" width="150" fixed="right"
v-if="groupsNames.length > 0">
  <template #default="scope">
    <el-row :gutter="25" align="middle">
      <el-col :span="12" class="centered-content">
        <el-button
          size="default"
          @click="handleDownload(scope.$index,
scope.row, scope)"
          :icon="UploadFilled"
          :loading-icon="Eleme"
          :loading="tableDownloadButtons[scope.$index].loading"
        ></el-button>
      </el-col>

      <el-col :span="12" class="centered-content">
        <el-button
          size="default"
          @click="viewMaterialInfo(scope.row)"
          :icon="View"
        ></el-button>
      </el-col>
    </el-row>
  </template>
</el-table-column>
</el-table>
</el-card>
<el-row class="m-t-15" align="middle" justify="center">
  <el-pagination
    v-model:current-page="delimetr"
    :page-size="materialsCount"
    layout="prev, pager, next"
    :total="resultsCount"
    hide-on-single-page
  />
</el-row>

```

```

    <el-card class="m-t-15" v-show="showMaterialInformation" id="smiles-card-
ref">
      <el-collapse v-model="showSmiles">
        <el-collapse-item name="Визуализация SMILES">
          <template #title>
            <p class="collapse-header">
              Визуализация SMILES кодов
            </p>
          </template>
          <material-information
            :smilesCode="smilesCode"
            :sizes="canvasSizes"
          />
        </el-collapse-item>
      </el-collapse>
    </el-card>
  </el-container>
</template>

<script setup lang="ts">
import {Elleme, Search, UploadFilled, View} from '@element-plus/icons-vue'
import {computed, h, nextTick, onMounted, reactive, Ref, ref, toRaw, unref, watch}
from 'vue'
import {API} from '@api'
import {IGetGroupsRequest} from '@types/Requests'
import {IGetGroupsResponse} from '@types/Responses'
import {dataViewFormats, fileTypeNames} from '@types/enums'
import SearchComponent from '@components/SearchComponent.vue'
import type {UploadInstance, UploadUserFile} from 'element-plus'
import {calculateSize} from '@helpers/Utils'
import MaterialInformation from '@components/MaterialInformation.vue'

const props = defineProps({
  fileType: {
    type: String as () => fileTypeNames,
    required: true
  },
})

const fileTypeNames = unref(props.fileType).toUpperCase()
const activeNames = ref('')
const uploadRef = ref<UploadInstance>()
const fileList = ref<UploadUserFile[]>([])
const
      fileApiActionUrl
      =
`import.meta.env.VITE_APP_API_URL}/upload_${fileTypeNames.toLocaleLowerCase()}_
file/`
const fileUploading = ref(false)

const handleUpload = () => {
  fileUploading.value = true

  const formData = new FormData()
  fileList.value.forEach((file, id) => {
    formData.append('files', file.raw)
  })
}

```

```

API[fileTypeName].uploadFile(formData)
  .then(response => {
    fileList.value.forEach((file) => {
      if (response.data[file.name].status === 'successfully') {
        file.status = 'success'
      } else {
        file.status = 'fail'
        file.error = response.data[file.name].error
      }
    })

    const uploadList = document.querySelector('.el-upload-list')
    const uploadChildren = Array.from(uploadList!.children)
    const files = response.data
    const fileKeys = Object.keys(files);
    uploadChildren.forEach((el, id) => {
      const oldDiv = document.getElementById(`upload-list__item-info-
${id}`)

      if (oldDiv) {
        oldDiv.remove()
      }
      const div = document.createElement('div')
      const file = files[fileKeys[id]]
      div.classList.add('upload-list__item-info')
      div.id = `upload-list__item-info-${id}`
      div.innerHTML = `

ID файла - ${file.id}</div>`
      if (file.error) {
        div.innerHTML = div.innerHTML.concat(`

Ошибка загрузки
файла - ${file.error}</div>`)
      }
      el.append(div)
    })
  })
  .catch(error => {
    console.error(error)
  })
  .finally(() => {
    fileUploading.value = false
  })
}

const clearFiles = () => {
  uploadRef.value!.clearFiles()
}

const showMaterialInformation = ref(false);
const smilesCode =
ref('OC(C(=O)O[C@H]1C[N+]2(CCCOC3=CC=CC=C3)CCC1CC2)(C1=CC=CS1)C1=CC=CS1');
let canvasSizes = reactive({ width: null, height: null })
const showSmiles = ref('Визуализация SMILES')

const viewMaterialInfo = (row) => {

  const regex = /{"smiles": "([^"]+)"/
  const match = row.visual_data ? row.visual_data.match(regex) : null


```



```

const element = document.getElementById('smiles-card-ref')

if (match) {
  smilesCode.value = match[1]
  showMaterialInformation.value = true
  if (element) {
    element.scrollToView({ behavior: 'smooth' })
  }
}

const materialInformation = computed(() => {
  if (showMaterialInformation.value) {
    return h(MaterialInformation, {
      smilesCode: smilesCode.value,
    });
  }
  return null;
});

watch(smilesCode, () => {
  materialInformation.value;
});

const selectedMode = ref('divided') as Ref<dataViewFormats>
const groupsLoading = ref(true)

const materialsDataLoading = ref(false)
const materialsCount = ref(5)
const delimetr = ref(1)
let structure: Record<any, any> = reactive({})
const materials = ref([])

onMounted(() => {
  const element = document.getElementById('materials-card')
  canvasSizes = { width: element.offsetWidth, height: element.offsetHeight }
})

const tableDownloadButtons: Record<any, any> = ref([])
for (let i = 0; i < materialsCount.value; i += 1) {
  tableDownloadButtons.value.push({loading: false})
}

const handleDownload = async (index: number, row: any, scope: any) => {
  tableDownloadButtons.value[index].loading = true;

  const request: any = {
    id: row.id
  }

  const response = await API.OTHER.downloadFile(request)

  const href = URL.createObjectURL(response.data);

  // create "a" HTML element with href to file & click

```

```

const link = document.createElement('a');
link.href = href;
const filename = `${row.id}.${fileName.toLowerCase()}`;
link.setAttribute('download', filename);
document.body.appendChild(link);
link.click();

// clean up "a" element & remove ObjectURL
document.body.removeChild(link);
URL.revokeObjectURL(href);

tableDownloadButtons.value[index].loading = false;
}

const resultsVisible = ref(false)
const executionTime = ref('0:00:00.000000')
const resultsCount = ref(0)
const groupsNames = ref([])

const getGroups = async () => {
  groupsLoading.value = true
  const request: IGetGroupsRequest = {
    fileType: fileName.toLowerCase(),
    dataViewFormat: selectedMode.value,
  }

  try {
    // @ts-ignore
    const response: IGetGroupsResponse = await API.OTHER.getGroups(request)
    groupsNames.value = response.data.structure
    structure = reactive(groupsNames.value.reduce((acc, group) => {
      acc[group] = ''
      return acc
    }, {}))

    activeNames.value = ''
    await nextTick()
    activeNames.value = 'Поиск'
  } catch (e) {
    console.error(e);
  }

  groupsLoading.value = false
}

const getMaterialsData = async () => {
  materialsDataLoading.value = true
  const request: any = {
    count: materialsCount.value,
    delimetr: delimetr.value,
    dataViewFormat: selectedMode.value,
    structure: toRaw(structure)
  }

  try {

```

Окончание приложения Г

```
    const response = await API[fileTypeName].getMaterialData(request)
    executionTime.value = response.data.execution_time
    resultsCount.value = response.data.resultsCount
    materials.value = response.data.data_from_DB
    resultsVisible.value = true
  } catch (e) {
    console.error(e);
  }
}

materialsDataLoading.value = false
}

onMounted(() => {
  getGroups()
})

watch(delimetr, async () => {
  showMaterialInformation.value = false
  await getMaterialsData()
})

watch(selectedMode, () => {
  showMaterialInformation.value = false
  materials.value = []
  resultsCount.value = 0
  resultsVisible.value = false
  getGroups()
})

</script>

<style>
.upload-list__item-info {
  margin-left: 24px;
  color: var(--el-text-color-regular);
  display: inline-flex;
  text-align: center;
  align-items: center;
  padding: 0 4px;
  transition: color var(--el-transition-duration);
  font-size: var(--el-font-size-base);
}
</style>
```