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	ДОПУСТИТЬ К ЗАЩИТЕ
к.х.н., доцент каф.	Заведующий кафедрой ЭВМ
______________И.Д. Юшина	____________ Д.В. Топольский
«___» ___________2023 г.	«___» __________ 2023 г.

Разработка средств секвенирования файлов описания химических компонентов и системы поиска на основе полученных структур данных

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090401.2023.472 ПЗ ВКР

Руководитель работы,
к.т.н., доцент каф. ЭВМ
______________И.Л. Кафтанников
«___» ___________2023 г.

Автор работы,
студент группы КЭ-222
________________С.М. Рассказов
«___» ___________2023 г.

Нормоконтролёр,
ст. преподаватель каф. ЭВМ
________________С.В. Сяськов
«___» ____________2023 г.

Челябинск-2023

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 бакалавра

студенту группы КЭ-222
Рассказов Сергею Михайловичу,
обучающемуся по направлению
09.04.01 «Информатика и вычислительная техника»

1. **Тема работы: «**Разработка средств секвенирования файлов описания химических компонентов и системы поиска на основе полученных структур данных» утверждена приказом ректора от 25 апреля 2023 г. №753-13/12, приложение №308/10.

2. **Срок сдачи студентом законченной работы:** 1 июня 2023 г.

3. **Исходные данные к работе:**

3.1. Разрабатываемая информационная система должна обеспечивать взаимодействие с другими компонентами с помощью разработанного API.

3.2. При разработке системы необходимо использовать свободно распространяемую СУБД.

3.3. Входными данными для информационной системы являются файлы в форматах, определяемых заказчиком.

3.4. Входные данные должны подвергаться анализу и выделению информации, необходимой заказчику (парсинг).

3.5. Необходимо обеспечить возможность поиска в выделенной информации.

3.6. Необходимо выполнить исследование ускорения поисковых процедур при применении, полученных в результате ВКР, методов и алгоритмов.

4. **Перечень подлежащих разработке вопросов:**

4.1. Обзор аналогов;

4.2. Форматы файлов для хранения информации о структурах и свойствах материалов;

4.3. Определение требований;

4.4. Проектирование информационной системы;

4.5. Разработка.

5. **Дата выдачи задания:** 1 декабря 2022 г.

Руководитель работы ________________________________/*И.Л. Кафтанников*/

Студент ___________________________________/С.*М. Рассказов* /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Обзор аналогов	20.02.2023	
Форматы файлов для хранения информации о структурах и свойствах материалов	20.03.2023	
Определение требований	01.04.2023	
Проектирование информационной системы	10.04.2023	
Разработка	10.05.2023	
Написание текста работы, согласование с руководителем и сдача на нормоконтроль	16.05.2023	
Подготовка презентации и доклада	24.05.2023	

Руководитель работы ________________________________/*И.Л. Кафтанников*/

Студент ____________________________________/*С.М. Рассказов*/

АННОТАЦИЯ

С.М. Рассказов. Разработка средств секвенирования файлов описания химических компонентов и системы поиска на основе полученных структур данных. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2023, 73 с., 3 ил., библиогр. список – 42 наим.

В рамках выпускной квалификационной работы производится анализ наиболее известных и релевантных сервисов по предоставлению информации о химических элементах и х соединениях. Ставятся задачи: разработка информационной системы; поиск соответствующих инструментов. Выполнена разработка информационной системы по обработке файлов с химическими данными форматов CIF и OUT. Разработанная система трансформируется (упаковывается) в образ docker, что обеспечивает ее взаимодействие с образами базы данных, компонентами интерфейса и сервера обработки входящих запросов. Проведены два эксперимента по определению эффективности разрабатываемой системы по сравнению с обычными способами обработки информации.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Ускорение поиска, исследования и внедрения новых материалов с заданными функциональными свойствами является критически важной задачей развития промышленности и всей экономики стран в целом. Одним из путей решения этой задачи является создание развитой инфраструктуры информационного обеспечения специалистов, в первую очередь, распределенной виртуально интегрированной сети баз данных, содержащих информацию о свойствах веществ и материалов и технологиях их получения и обработки, а также систем компьютерного конструирования и моделирования материалов, доступных из Интернет специалистам самого разного профиля: научным работникам, инженерам, технологам, бизнесменам, госслужащим, студентам и т.д. В последние годы в развитых странах были выдвинуты и поддержаны правительствами инициативы, направленные на организацию инфраструктуры доступа к экспериментальным и расчетным данным о материалах [1, 2].

Химические соединения играют важную роль в современном мире, находя применение в различных отраслях промышленности, медицине, а также научных исследованиях. Поиск новых материалов и их свойств является важной задачей в науке и технологии. Существует два основных подхода в решении подобных задач.

Первый, это синтез новых материалов в лаборатории. По другому это метод «проб и ошибок» Этот метод заключается в том, чтобы изучать различные материалы и их свойства, а затем изменять их состав и структуру, чтобы получить желаемые свойства. Этот метод является довольно трудоемким, но может привести к открытию новых материалов с уникальными свойствами.

Преимущества:

– этот метод может быть полезным, когда недостаточно данных о свойствах материалов или когда нет ясного понимания того, какие свойства нужны.

Недостатки:

– этот метод может быть очень трудоемким и занимать много времени, особенно если нужно протестировать множество материалов;

– этот метод может быть дорогостоящим, если требуется большое количество материалов для тестирования;

– этот метод может быть неэффективным, если свойства материалов зависят от многих факторов и не могут быть изменены легко.

Другой подход заключается в использовании компьютерных моделей и цифровых двойников для предсказания свойств материалов. Это позволяет исследователям оптимизировать свойства материалов до их физического создания в лаборатории. Компьютерные модели представляют собой математические описания свойств материалов, которые строятся на основе физических законов и экспериментальных данных. Модели могут быть использованы для изучения различных свойств материалов, таких как механические, электрические, магнитные, тепловые и другие.

Цифровой двойник (Digital Twin) — это технология, которая позволяет создать виртуальную копию физического объекта, процесса или системы с использованием цифровых технологий. В материаловедении цифровой двойник может быть использован для моделирования и анализа свойств материалов и их поведения в различных условиях. Создание цифрового двойника включает в себя несколько этапов. Сначала собираются данные о физическом объекте, например, о его геометрии, химическом составе, механических свойствах и т.д. Эти данные затем используются для создания трехмерной модели объекта с помощью компьютерного программного обеспечения. Модель может быть настроена для учета различных условий, таких как температура, давление, вибрация и других.

Преимущества:

– компьютерное моделирование позволяет изучать свойства материалов без необходимости производить их физически. Это может сократить время и затраты на исследование материалов;

– компьютерное моделирование может использоваться для анализа свойств материалов на молекулярном уровне, что может помочь в определении оптимальной структуры материала;

– компьютерное моделирование может быть полезным, когда нужно прогнозировать свойства материалов в различных условиях.

Недостатки:

– компьютерное моделирование может не учитывать все аспекты реального мира, что может привести к неточным результатам;

– компьютерное моделирование может быть ограничено доступностью моделей или данных о свойствах материалов.

Одним из примеров использования цифрового двойника в материаловедении является моделирование поведения композитных материалов при механической нагрузке. Цифровой двойник может использоваться для анализа, как различные составы композитных материалов влияют на их механические свойства и как они будут вести себя при различных условиях нагрузки. Это позволяет исследователям и инженерам оптимизировать состав материала и проектировать более эффективные конструкции.

Таким образом, цифровой двойник в материаловедении является мощным инструментом, который позволяет ускорить и улучшить исследования и проектирование материалов и их свойств. Он также может использоваться для повышения качества производства и обеспечения безопасности и надежности материалов и изделий на основе них.

Цели и задачи работы будут поставлены после обзора аналогов разрабатываемой информационной системы.

1 ОБЗОР АНАЛОГОВ

1.1 Materials Project

Materials Project — это открытый проект, который предоставляет веб-доступ к вычисленной информации о известных и предсказанных материалах, а также мощным инструментам анализа для вдохновения и проектирования новых материалов. Проект предоставляет бесплатный онлайн-ресурс для научных исследователей, инженеров и всех, кто интересуется изучением свойств различных материалов [3].

Создатели проекта - группа ученых из разных университетов и научных организаций под руководством профессора Кристина Перссона из лаборатории Беркли. Проект был запущен в 2011 году и с тех пор стал одним из ведущих проектов в области материаловедения. Основным инструментом проекта является база данных более чем из 124 000 неорганических соединений и около 35 000 молекул. База данных содержит информацию о структуре материалов, их электронных и магнитных свойствах, термодинамических параметрах и других характеристиках.

Для обеспечения обработки такого большого количества данных используются суперкомпьютеры Национального научно-исследовательского центра энергетических исследований лаборатории Беркли (National Energy Research Scientific Computing Center (NERSC) [4].

В Materials Project используется NoSQL база данных MongoDB, которая использует индексацию на основе B-деревьев. Это позволяет обеспечить быстрый поиск по различным критериям, таким как химический состав, свойства материалов и т.д. Также в базе используется кэширование данных, что также способствует ускорению работы с базой.

Структура проекта состоит из нескольких компонентов: базы данных материалов, которая содержит информацию о химических соединениях и их свойствах; веб-интерфейса, который позволяет пользователям просматривать, фильтровать и скачивать данные; приложений для расчета различных характеристик материалов, таких как фазовые диаграммы, теплоемкость, электропроводность и т.д.; API для программного доступа к данным и функциям проекта.

Для наполнения базы данных используются компьютерные модели и цифровые двойники. Исследователи используют первоначальные данные о материалах, такие как их химический состав и кристаллическая структура, для создания моделей свойств материалов. Затем модели могут быть использованы для предсказания свойств материалов, которые еще не были экспериментально изучены. Проект также позволяет исследователям оптимизировать процесс синтеза новых материалов. С помощью компьютерных моделей и цифровых двойников исследователи могут определить оптимальные условия для синтеза материалов, которые имеют желаемые свойства.

Особенности проекта заключаются в том, что он является открытым и бесплатным для всех заинтересованных лиц; что он использует передовые вычислительные методы для моделирования материалов на основе первых принципов; что он постоянно обновляется и расширяется новыми данными и функциями; что он способствует сотрудничеству между учеными из разных областей знаний; что он помогает ускорить разработку новых материалов для различных приложений .

В основном, данные в Materials Project поступают от экспериментов, теоретических расчетов и анализа литературных источников. Эти данные затем проходят через качественную проверку, обработку и структурирование, прежде чем они станут доступными для пользователей.

Пользователи могут получить доступ к данным через интерфейс веб-сайта Materials Project или через API. Для получения доступа к данным через API, пользователи должны зарегистрироваться и получить API-ключ. После получения ключа, пользователи могут использовать запросы API, чтобы получать доступ к различным типам данных, таким как структуры, энергии формирования, свойства электронной структуры и многое другое.

Кроме того, пользователи могут загружать свои собственные данные в Materials Project, используя инструменты, такие как pymatgen, это Python-библиотека для работы с материалами и расчетами.

В этой базе данных идентификатор материала представляет собой его формулу, записанную в виде набора химических символов элементов, разделенных символом "-". Например, идентификатор для материала графит будет "C-". Если материал содержит несколько элементов, то их символы будут идти в порядке алфавитного списка. Например, идентификатор для материала CaF2 будет "Ca-F".

В базу данных Materials Project попадают различные части исходных данных, относящиеся к материалам и их свойствам. Несколько примеров того, что может попадать в базу данных:

– структуры материалов - геометрические описания атомной структуры материалов;

– энергии формирования - энергия, необходимая для формирования материала из элементарных составляющих;

– электронные свойства - электронные уровни и свойства, такие как ширина запрещенной зоны, плотность состояний и т.д.;

– механические свойства - свойства материалов, связанные с их механическим поведением, такие как упругость, твердость и т.д.;

– термодинамические свойства - свойства материалов, связанные с их термодинамическим поведением, такие как теплоемкость, теплопроводность и т.д.;

– информация об источниках - информация о источниках данных, используемых для получения и обработки исходных данных.

Кроме того, Materials Project также содержит информацию о реакциях на основе материалов, симуляциях квантовой механики и многом другом, связанных с материалами и их свойствами.

Каждый материал представлен в базе данных в виде записи, которая содержит информацию о свойствах материала. Каждое свойство материала также имеет свою собственную таблицу, которая связана с таблицей материалов.

Для хранения данных в базе данных Materials Project используется схема базы данных, основанная на отношениях. Каждая таблица базы данных представляет отдельный тип данных, а каждая запись в таблице представляет конкретный элемент данных.

Существует множество таблиц, используемых в базе данных Materials Project для хранения различных типов данных. Некоторые из наиболее распространенных таблиц включают в себя таблицы со структурами материалов, таблицы с электронными свойствами, таблицы с механическими свойствами, таблицы с термодинамическими свойствами и таблицы с информацией об источниках данных.

Структурирование данных в базе данных Materials Project позволяет быстро и эффективно хранить и получать информацию о материалах и их свойствах. Кроме того, структурирование данных также позволяет использовать мощные инструменты для анализа и обработки данных, такие как SQL-запросы и машинное обучение.

В базе данных Materials Project используется множество таблиц для хранения различных типов данных. Некоторые из наиболее распространенных таблиц включают в себя:

– таблица "materials" - содержит информацию о каждом материале в базе данных, включая его идентификатор, формулу, кристаллическую структуру, дополнительную информацию о материале и т.д.;

– таблица "material_properties" - содержит данные о свойствах материалов, таких как плотность, коэффициент линейного расширения, модуль Юнга и т.д.;

– таблица "xas" - содержит данные о рентгеновской спектроскопии поглощения (XAS) для материалов;

– таблица "bandstructure" - содержит данные о электронной структуре материалов, такие как зоны Бриллюэна, энергетические уровни, дисперсия и т.д.;

– таблица "dos" - содержит данные о плотности состояний (DOS) для материалов;

– таблица "elasticity" - содержит данные о механических свойствах материалов, таких как модуль Юнга, коэффициент Пуассона и т.д. ;

– таблица "piezo" - содержит данные о пьезоэлектрических свойствах материалов, таких как пьезоэлектрический коэффициент и диэлектрическая постоянная;

– таблица "surface_properties" - содержит данные о свойствах поверхности материалов, таких как поверхностная энергия, топология поверхности и т.д. ;

– таблица "thermo" - содержит данные о термодинамических свойствах материалов, таких как теплоемкость, энтальпия и энтропия;

– таблица "sources" - содержит информацию об источниках данных, которые были использованы для получения информации о материалах в базе данных;

– таблица "calculations" - содержит данные о расчетах, которые были выполнены для каждого материала в базе данных;

– таблица "structures" - содержит информацию о кристаллических структурах материалов.

Это только некоторые из таблиц, используемых в базе данных Materials Project. Каждая таблица представляет отдельный тип данных, и большинство таблиц связаны между собой для обеспечения более эффективной работы с данных.

Существуют также таблицы, связанные с расчетами энергий и свойств материалов, например:

– таблицы с расчетами энергий структур для различных методов, таких как GGA, GGA+U, LDA и других;

– таблицы с информацией о свойствах материалов, например, модулях упругости, теплопроводности, коэффициентах теплового расширения и т.д.;

– таблицы с информацией о фазовых переходах и термодинамических свойствах материалов при различных температурах и давлениях;

– таблицы с данными о магнитных свойствах материалов, таких как магнитная восприимчивость, магнитное моменты и т.д.;

Таблицы с результатами экспериментальных исследований в Materials Project содержат данные, полученные из различных экспериментальных методов анализа материалов. Некоторые из этих таблиц могут включать:

– рентгеноструктурный анализ: таблицы с рентгеноструктурными данными, полученными при анализе кристаллических материалов методом рентгеновской дифракции;

– электронная микроскопия: таблицы с данными, полученными при анализе материалов с помощью электронной микроскопии, такой как сканирующая электронная микроскопия (SEM), трансмиссионная электронная микроскопия (TEM) и другие;

– электронные свойства: таблицы с данными об электронных свойствах материалов, таких как проводимость, электронная плотность, энергетические уровни и т.д.

Данные из этих таблиц могут использоваться для получения более полной картины о материале, а также для сравнения экспериментальных данных с теоретическими расчетами.

Статистика использования проекта свидетельствует о его популярности и полезности. По данным на январь 2020 года, проект имел более 100 тысяч зарегистрированных пользователей из более чем 100 стран мира; более 10 миллионов запросов к базе данных в год; более 3000 цитирований в научных публикациях. Проект также получил ряд наград за свой вклад в науку и образование.

1.2 NOMAD

NOMAD (The NOn-covalent Molecular Interactions in Database) — это база данных для химиков, которая содержит информацию о молекулярных взаимодействиях, в основном, не-ковалентных, таких как водородные связи, взаимодействия ван-дер-Ваальса и ионно-дипольные взаимодействия. NOMAD является результатом многолетней работы исследователей в области химии и молекулярной биологии. Она содержит более чем 12 млн. записей о молекулярных структурах и около 3 млн записей о материалах. Общий объём загруженных файлов превышает 100 ТБ. Одной из особенностей является то, что она обновляется регулярно и содержит информацию о последних исследованиях в области молекулярных взаимодействий. База данных также содержит информацию о физических и химических свойствах молекул, таких как масса, температура плавления и кипения, показатель преломления и т.д. [5].

Цель – ускорение разработки и использования материалов с заданными функциональными свойствами. Программа стартовала в ноябре 2015 г. В рамках проекта ЕС HORIZON2020 (объём финансирования около 5 млн евро). Существенным недостатком NoMaD является ориентация на информационные ресурсы США (главным образом БД NIST по свойствам веществ и материалов) и информационные системы с расчётными данными. [6]. База данных NIST включает в себя разнообразные наборы данных, такие как физические константы, химические свойства веществ, спектральные данные, тепловые данные и другие. Эти данные используются в научных и инженерных исследованиях, промышленности и других областях.

В базе данных NOMAD можно осуществлять поиск по различным параметрам, таким как тип взаимодействия, тип молекулы, структурная формула и другие. Также в базе данных доступны инструменты для визуализации молекулярных структур и взаимодействий.

Открытый доступ позволяет использовать базу как исследователям, так и обычными пользователями, интересующимися молекулярной химией и биологией. Она также может быть использована для обучения и в образовательных целях.

Отдельной частью NOMAD является архив, содержащий информацию о симуляциях вещественных систем. В репозитории хранятся файлы архивов, содержащие информацию о симуляциях, в том числе координаты атомов, кинетические и потенциальные энергии, давление, температуру, а также метаданные, связанные с этими симуляциями, такие как описание методологии, используемой для симуляций, и результаты анализа данных. Эти данные могут быть использованы для исследования свойств материалов и молекул в различных условиях, таких как высокие давления и температуры, а также для разработки новых материалов и прогнозирования их свойств.

Репозиторий NOMAD Archive обеспечивает открытый доступ к этим данным и предоставляет мощный поисковый интерфейс, который позволяет искать данные с помощью различных параметров, таких как материал, метод симуляции, температура и другие. Также в репозитории доступны инструменты для анализа и визуализации данных.

В базе данных NOMAD используются следующие таблицы:

– system: содержит информацию о системах, включая уникальный идентификатор, состояние системы (например, газовая, твердая или жидкая), структуру системы (например, атомная или молекулярная), количество атомов, энергию, температуру, давление и другие параметры;

– calculation: содержит информацию о расчетах, включая уникальный идентификатор, идентификатор связанной системы, тип расчета (например, первопринципный, классический или гибридный), используемый метод расчета (например, функционал плотности или Монте-Карло), используемые базисные функции, точность расчета, время выполнения расчета и другие параметры;

– atom: содержит информацию о каждом атоме в системе, включая уникальный идентификатор, координаты, заряд, тип атома, магнитный момент и другие параметры;

– simulation_cell: содержит информацию о ячейке моделирования, включая размеры, форму, ориентацию, параметры кривизны и другие параметры;

– energy: содержит информацию об энергии системы, включая уникальный идентификатор, энергию каждого компонента системы (например, кинетическую, потенциальную или термодинамическую), общую энергию, дисперсию и другие параметры;

– force: содержит информацию о силе, действующей на каждый атом в системе, включая уникальный идентификатор, компоненты силы в трех измерениях, общую силу и другие параметры;

– stress: содержит информацию о тензоре напряжения в системе, включая уникальный идентификатор, компоненты тензора напряжения в трех измерениях, общее напряжение и другие параметры;

– electronic_structure: содержит информацию о свойствах электронной структуры системы, включая уникальный идентификатор, энергии состояний, электронную плотность, функцию волновой функции, магнитный момент и другие параметры;

– molecule: содержит информацию о молекуле, включая уникальный идентификатор, геометрию, вращательную константу, вибрационную энергию и другие параметры;

– kpoint_set: содержит информацию о наборе точек Бриллюэна для расчета электронной структуры;

– energy: содержит информацию об энергии системы, такую как полная энергия, энергия ферми, энергии состояний, энергия связи и т.д.;

– band_structure: содержит информацию о зонной структуре материалов, такую как энергии на граничных точках, индексы зон и т.д.;

– dos: содержит информацию о плотности состояний системы, такую как энергетический диапазон, плотность состояний, плотность состояний спина и т.д.

Эти таблицы взаимосвязаны и позволяют получить полное представление о химической системе и ее свойствах на основе результатов экспериментальных исследований и расчетов.

В этой базе данных идентификатор материала представляет собой уникальный идентификатор, присвоенный материалу в рамках этой базы данных. Обычно он состоит из буквенно-цифровой комбинации, которая может включать в себя информацию о структуре, исходных файлах и других метаданных. Для получения идентификатора материала в базе NOMAD, необходимо загрузить данные в базу данных и произвести поиск по параметрам материала, таким как химический состав, кристаллическая структура и другие характеристики.

Для NOMAD входными форматами являются файлы с расширениями:

– XYZ (координаты атомов в молекуле в формате XYZ);

– CIF (файлы кристаллических структур в формате CIF);

– VASPRUN.XML (результаты расчетов на основе первых принципов с помощью VASP);

– OUTCAR (файлы, содержащие дополнительные результаты расчетов на основе первых принципов с помощью VASP);

– CASTEP (.cell и .param файлы для CASTEP);

– CP2K (.inp файлы для CP2K);

– Gaussian (.log файлы для Gaussian);

– ORCA (.out файлы для ORCA);

– Quantum Espresso (.in и .out файлы для Quantum Espresso);

– NWChem (.nw файлы для NWChem);

– FHI-aims (.in файлы для FHI-aims).

Эти форматы позволяют описывать различные системы: от молекулярных до кристаллических, проводить расчеты на основе первых принципов, использовать различные программы для расчетов.

В базе данных NOMAD используется индексация на основе Apache Solr, которая позволяет быстро и эффективно выполнять поисковые запросы на больших объемах данных. Также в NOMAD используются различные методы оптимизации запросов и ускорения обработки данных, включая распределенные вычисления и многопоточность.

Apache Solr - это мощный и масштабируемый поисковый сервер, основанный на проекте Apache Lucene. Он предоставляет инструменты и возможности для создания поисковых систем и обработки больших объемов текстовых данных. Это ПО использует инвертированный индекс для эффективного поиска и ранжирования данных. Он может обрабатывать различные типы данных, включая текстовые документы, JSON, XML, CSV и другие. Solr обладает мощными возможностями фильтрации, обработки запросов и визуализации результатов поиска.

Solr поддерживает распределенную архитектуру, позволяющую масштабировать и обрабатывать большие объемы данных. Он может интегрироваться с другими инструментами и платформами, такими как Apache Hadoop, Apache Spark и другими, для обработки и анализа данных.

В целом, база данных NOMAD является важным инструментом для исследований в области химии, медицины, фармакологии и других наук. Она позволяет получать информацию о молекулярных взаимодействиях и структурах для дальнейшего анализа и использования в научных исследованиях.

Взаимосвязи данных в NOMAD

На данном этапе обработки данных в базе NOMAD происходит связывание результатов расчетов с конкретными материалами, на основе информации из их идентификаторов и метаданных. Это позволяет создать удобную структуру данных, позволяющую пользователю легко находить и анализировать нужную информацию.

Для связывания результатов расчетов с конкретными материалами используется идентификатор материала, который хранится в метаданных расчета. Этот идентификатор связывает результаты расчета с материалом и позволяет легко находить все расчеты, выполненные для конкретного материала.

Кроме того, в процессе связывания происходит обновление метаданных материалов на основе полученных результатов расчетов. Например, можно добавить информацию о структурных свойствах материала, полученных из результатов расчетов, в метаданные этого материала, чтобы пользователи могли легко найти нужные материалы при поиске по свойствам.

1.3 ICSD

The Inorganic Crystal Structure Database (ICSD) — это электронная база данных, которая содержит информацию о кристаллических структурах неорганических соединений, определенных экспериментально методами рентгеноструктурного анализа кристаллов. ICSD является важным инструментом для химиков, изучающих неорганическую химию и материаловедение.

База данных содержит более 200 000 записей, включающих данные о кристаллических структурах минералов, металлов, керамических материалов и других неорганических соединений. В базе данных можно найти информацию о молекулярной геометрии, межмолекулярных взаимодействиях, кристаллической решетке, термодинамических свойствах и других характеристиках соединений [7].

ICSD содержит экспериментальные данные, полученные из различных источников, включая публикации в научных журналах, конференции, отчеты и другие источники. Каждая запись в ICSD содержит информацию о структуре кристалла, включая координаты атомов, расстояния между атомами, углы связей и другие характеристики структуры. Также доступны дополнительные метаданные, такие как авторы, названия соединений, классификация материалов и т. д.

Система позволяет использовать поиск по различным критериям, таким как химический состав, кристаллическая структура, пространственная группа, свойства материала и другие параметры. Это позволяет исследователям и инженерам находить интересующие их материалы, а также проводить анализ и сравнение структурных данных. Стоит учитывать, что это коммерческий продукт и доступ к нему осуществляется по подписке. Однако, для академических исследований и образовательных целей доступ к ICSD может быть предоставлен через лицензии и подписки университетов и исследовательских организаций.

База данных ICSD (The Inorganic Crystal Structure Database) постоянно обновляется, и в ней содержится обширная коллекция структурных данных. Она включает данные о тысячах различных неорганических материалов, и это число постоянно растет вместе с пополнением базы новыми экспериментальными результатами и публикациями.

Одна из ключевых особенностей ICSD - это поддержка различных форматов данных, включая форматы файлов, такие как CIF (Crystallographic Information File), которые широко используются в кристаллографии. Формат CIF обеспечивает стандартную структуру данных, содержащую информацию о структуре кристалла, его атомах, симметрии и других свойствах.

ICSD также предоставляет удобный веб-интерфейс, который обеспечивает пользовательский доступ к базе данных. Через веб-интерфейс исследователи могут искать, просматривать и анализировать данные о структуре материалов, а также получать доступ к сопутствующей информации, включая статьи и ссылки на источники.

В целом, ICSD является ценным инструментом для исследователей и инженеров, которым требуется доступ к качественным и проверенным экспериментальным данным о структурах неорганических материалов. Она предоставляет важную информацию, которая способствует развитию научных исследований, промышленности и разработке новых материалов и технологий. ICSD широко используется в различных областях, таких как материаловедение, катализ, электроника, фармацевтика и другие, где важна информация о кристаллической структуре неорганических материалов для понимания их свойств и потенциальных применений.

1.4 Менее релевантные аналоги

1.4.1. OMDb

OMDb (Organic Materials Database) — это база данных, содержащая информацию о более чем 26 миллионах различных химических соединений. База данных наполняется из различных источников, таких как PubChem, ChEMBL и ChEBI. Это все базы данных, связанные с химией и химическими соединениями, которые содержат информацию о химической структуре, свойствах и активности соединений. Эта информация включает информацию о структуре молекул, свойствах, реакциях и метаболитах [8].

OMDB предоставляет разнообразную информацию о различных свойствах и характеристиках органических материалов, включая их структуру, физические и электронные свойства, оптические свойства, термическую стабильность, электропроводность и другие параметры. База данных включает данные, полученные из различных экспериментальных и теоретических источников, а также из публикаций и отчетов.

OMDb хранит данные в форматах MOL-файлов, SMILES и InChI. MOL-файлы — это текстовые файлы, содержащие информацию о молекуле, включая ее структуру, атомные координаты и другие свойства. SMILES (Simplified Molecular Input Line Entry System) — это текстовый формат, используемый для представления химических структур в виде строки символов. InChI (International Chemical Identifier) — это уникальный идентификатор для химических соединений. Эти файлы содержат информацию о молекуле, включая ее структуру, координаты атомов и другие свойства.

Веб-сайт OMDb предоставляет инструменты и API, которые позволяют искать молекулы по их идентификаторам, находить похожие молекулы и загружать молекулы на сайт. Функционал также позволяет визуализировать данные и анализировать результаты. Пользователи могут просматривать структуры материалов, строить диаграммы, графики и проводить другие операции для более детального анализа и интерпретации данных.

Доступ к базе может быть ограничен и требовать подписки или лицензии для полного доступа к базе данных и ее функциональности. Однако, для академических исследований и образовательных целей могут быть предоставлены ограниченные бесплатные варианты доступа.

OMDB является ценным ресурсом для исследователей, работающих в области органической химии, материаловедения и смежных областей. Она обеспечивает доступ к качественным и проверенным данным о свойствах органических материалов, что способствует разработке новых материалов, проектированию устройств и оптимизации их свойств для различных приложений.

1.4.2. PDB

The Protein Data Bank (PDB) - база данных о белках и других биомолекулах. Это цифровой репозиторий, содержащий трехмерные структуры биологических макромолекул, в основном белков и нуклеиновых кислот, которые были экспериментально определены с помощью рентгеновской кристаллографии, ядерного магнитного резонанса и других экспериментальных методов. PDB является ценным ресурсом для исследователей, изучающих структуру и функцию макромолекул в биологии, биохимии и биофизике.

PDB был создан в 1971 году в рамках сотрудничества Национальных институтов здравоохранения (NIH), Национального фонда науки (NSF) и Департамента энергетики (DOE) в США. В настоящее время он управляется Всемирным банком данных о белках (wwPDB), который является партнерством между организациями в США, Европе и Азии [9].

В этой базе содержится более 180 000 структур биологических макромолекул, что делает его крупнейшим общедоступным репозиторием таких структур. Структуры депонируются исследователями со всего мира, которые делают свои данные общедоступными, чтобы облегчить дальнейшие исследования и открытия в этой области.

Каждой структуре присваивается уникальный идентификатор, известный как PDB ID, который состоит из четырех буквенно-цифровых символов. PDB ID используется для идентификации структуры и ссылки на нее в научных публикациях [10].

Кроме атомных координат макромолекул, в PDB также содержится дополнительная информация о структуре, такая как экспериментальные методы, использованные для ее определения, исходный организм макромолекулы и функциональные аннотации, связанные с ней. Эта дополнительная информация помогает исследователям лучше понимать структуру и функцию биологических макромолекул.

1.4.3. CSD/CCDC

База данных CSD (Cambridge Structural Database) или CCDC (Cambridge Crystallographic Data Centre) содержит структурные данные о кристаллических соединениях, полученные из различных источников, включая публикации, научные журналы, базы данных и лабораторные исследования [11].

CSD/CCDC является наиболее авторитетной и обширной базой данных структурных данных кристаллических соединений. Она включает в себя информацию о молекулярных структурах, атомных координатах, связях, симметрии и других химических свойствах кристаллов [12].

База данных CSD/CCDC предоставляет исследователям возможность:

– поиска структурных данных по различным критериям, таким как химический состав, свойства соединения, авторы и другие;

– анализа и сравнения структур, включая изучение сходств и различий между разными кристаллическими соединениями;

– визуализации структурных данных в трехмерном формате для более наглядного представления молекулярных структур;

– получения информации о физических и химических свойствах соединений, таких как длины и углы связей, симметрия кристаллов и т. д.

– извлечения полезной информации для своих исследований и проектов, такой как структурные шаблоны, химические тренды, прогнозирование свойств и другие.

CSD/CCDC активно используется в химической и фармацевтической индустрии, а также в научных исследованиях и академической сфере. Она представляет собой ценный инструмент для химиков и кристаллографов, позволяющий легко получать доступ к огромной базе данных структурных данных и проводить разнообразные анализы и исследования в области кристаллографии.

1.5 Этапы обработки данных в рассмотренных системах

Обычно путь прохождения файла от загрузки до получения результатов обработки химиком может быть следующим:

– загрузка файла в базу данных. Это может происходить автоматически при помощи скриптов или вручную, когда химик сам загружает файл;

– предварительная обработка данных. В этом этапе может происходить удаление дубликатов, проверка на ошибки формата, приведение данных к единому виду и т.д.;

– сопоставление существующих данных. Если в базе уже есть похожий файл или данные из этого файла уже были обработаны ранее, то новый файл может быть сопоставлен с уже имеющимися данными. Это позволяет избежать дублирования обработки и сократить время на обработку новых данных;

– обработка данных. На этом этапе происходит обработка данных, которая может включать в себя расчет различных параметров, поиск структуры, анализ свойств материалов и т.д.;

– анализ результатов обработки. Полученные результаты обработки могут быть анализированы химиком, который проводит дополнительные эксперименты и проверки, чтобы убедиться в правильности результатов;

– публикация результатов. Полученные результаты могут быть опубликованы в научных статьях или доступны для общественности на различных платформах;

Выводом является сравнительная таблица (Таблица 1). В ней отражены самые основные по мнению автора дынные об использовании сервисов химической информатики.

Таблица 1 – Сравнительная таблица аналогов

Имя проекта	**Основной тип данных**	**Доступ**	**Данные**	**Архитектура**	**Основное преимущества**	**Основной недостаток**
Materials Project	Кристаллические материалы	Открытый доступ	Расчетные и экспериментальные	Микросервисная архитектура (WEB интерфейс, API, DB)	Помощь в подборе методов расчётов, огромный инструментарий	Возможны проблемы с работой из-за большой загруженности
NOMAD	Неорганические соединения	Открытый доступ	Расчетные и экспериментальные	API есть, распределенная БД	алгоритмы машинного обучения	Модули аналитики на внешних серверах
ICSD	неорганические соединения	Подписка\ лицензия	Экспериментальные	Закрыто (коммерция)	Довольно полная картина материалов	Доступ
OMDb	Молекулы и органические материалы	Открытый доступ,	Опытные и исследовательские данные	API есть, остальное неизвестно	Агрегатор данных	Данные только MOL-файлов, SMILES и InChI
PDB	Белки и биомолекулы.	Открытый доступ,	В основном опытные	Микросервисная архитектура (WEB интерфейс, API, DB)	Ссылка на исходный организм макромолекулы	Используются другие типы данных
CSD	Органические и неорганически	Лицензия, ограниченная функциональн	Экспериментальные	Закрыто (коммерция)	Авторитетность Кембриджский банк	Доступ

1.6 Постановка задачи

Цель данной работы состоит в разработке информационной системы для хранения экспериментальных, теоретических и расчётных данных, связанных с химическими соединениями. Для дальнейшего их использования при создании цифровых двойников материалов. А также повысить эффективность и точность исследований в области химии.

Входными данными для ИС являются файлы форматов CIF (Crystallographic Information File) и OUT полученные с помощью программы Crystal 17. Эти файлы представляют из себя последовательные текстовые документы в кодировках ASCII и UTF-8 соответственно. В них содержится информация об атомах, молекулах, кристаллах, их связях, свойствах и служебная информация об условиях экспериментов, авторах и т.п. Из полученных файлов информационная система должна выявлять наиболее полезные данные и структурировать их для последующего их анализа и выполнения процедур поиска.

Задачи в рамках данной работы:

– анализ требований пользователей: необходимые функции, алгоритмы взаимодействия, масштабируемость архитектуры;

– разработка парсера для CIF файлов;

– разработка парсера OUT файлов, использующихся для оптимизации химических элементов;

– разработка серверной части: создание API для взаимодействия с другим программным обеспечением, функции загрузки файлов и хранения исходных. Защита от дублирования данных;

– проектирование базы данных - определение структуры базы данных;

– модуль взаимодействия с базой данных;

– проведение экспериментов по сравнению результатов при работе с обработанными и необработанными данными;

– предложить дальнейшие этапы развития информационной системы.

2 ФОРМАТЫ ФАЙЛОВ ДЛЯ ХРАНЕНИЯ ИНФОРМАЦИИ О СТРУКТУРАХ И СВОЙСТВАХ МАТЕРИАЛОВ

2.1 CIF (Crystallographic Information File)

Химия — это большая наука с множеством направлений и ветвлений. Помимо направлений существует еще множество подходов, разные взгляды на теле или иные расчеты и эксперименты. Всё это дает нам множество различных форматов данных, которые заточены, каждый под свою цель и имеют свои особенности. В настоящее время используются десятки различных форматов, и охватить их все в рамках одной работы это слишком сложная задача. Так как данная работа нацелена помочь исследователям из «Лаборатории многомасштабного моделирования многокомпонентных функциональных материалов» Южно-Уральского Государственного Университета, то форматы файлов для начала разработки был рекомендованы её сотрудниками.

Файлы CIF — это формат для хранения информации о кристаллических структурах и молекулах. Они используются в кристаллографии и материаловедении для описания трехмерной структуры кристаллических соединений и полимеров. Файл этого формата содержит информацию о расположении атомов в кристаллической решетке, типах атомов, их взаимном расположении, а также другие данные, такие как температура, давление симметрия, термодинамические параметры, длины связей и т.д. Данные из файлов могут быть использованы для моделирования и анализа кристаллических структур [13].

Существуют рекомендации по формированию CIF файлов, они описаны в документе «A Guide to CIF for Authors» [14]. Этот документ, создан Международной ассоциацией кристаллографии (International Union of Crystallography, IUCr). В нем описывается структура CIF-файла, правила формирования заголовков и форматирования данных, а также рекомендации по выбору ключевых слов и терминов. Эти рекомендации не содержат строгой регламентации всей информации. То есть, часть информации может быть упущена или наоборот добавлена конкретным автором. Документ содержит также примеры правильного форматирования и оформления данных в CIF-файле. Авторы, работающие с данными кристаллографии, могут использовать "A Guide to CIF for Authors" в качестве руководства по написанию и форматированию файлов CIF, чтобы обеспечить правильное представление и обмен данных с другими учеными и базами данных.

Некоторые правила, описанные в документе:

– файл CIF должен быть написан на языке ASCII и иметь расширение CIF;

– файл должен содержать заголовок, который указывает на авторов, название кристалла и дату создания файла;

– в CIF-файле должны использоваться ключевые слова, определенные в международном словаре CIF, который описывает стандартные термины для описания свойств кристаллов. (Речь идет не о переменных, а о базовом наборе ключевых слов);

– ключевые слова должны быть написаны в верхнем регистре;

– табличные данные в CIF-файле должны быть организованы в блоки, разделенные ключевым словом «loop_»;

– значения данных в блоке могут быть записаны как отдельные значения, так и в виде массивов данных;

– в CIF-файле должны использоваться только допустимые символы ASCII. Все символы, которые не являются допустимыми, должны быть заменены на эквиваленты в кодировке ASCII;

– имена переменных должны быть короткими и описательными, их необходимо выбирать таким образом, чтобы они были легко понятны другим пользователям;

– в CIF-файле не должно быть повторяющихся данных. Если данные повторяются в нескольких блоках, то их необходимо вынести в отдельный блок;

– в CIF-файле должны быть записаны все экспериментальные параметры и значения, а также условия, при которых был проведен эксперимент.

2.2 OUT

Файлы OUT — являются результатом работы программных пакетов по расчету электронной структуры материалов. Формат стал популярным из-за постоянного развития в области химии и материалов. Старые форматы уже не могут покрывать все требования современных исследований. В них содержится информация о различных параметрах, таких как: энергии связей, оптимизированные координаты атомов, а также спектры и другие расчетные данные. Эти данные могут быть использованы для предсказания свойств материалов, исследования реакций и других расчетов в области химии и материаловедения.

OUT файлы не имеют стандарта, как CIF. Программное обеспечение, генерирующее такие файлы, может определить свои собственные правила для формирования выходных данных. Также стоит отметить, что этим форматом пользуются не только программы для химических расчетов, но и большое количество совершенно разнонаправленных программ, они используют его для хранения своих логов. Чтобы правильно интерпретировать такие файлы, нужно смотреть на документацию конкретной программы, которая его сгенерировала. Некоторые программы, например, Gaussian (программа для расчёта матрицы Гесса), генерирует результат в OUT файл, но если эту задачу выполнить с помощью другой программы, то данные в выходном файле могут отличаться по своей форме и структуре, хотя смысл остается прежним.

В лаборатории ЮУрГУ используется программа CRYSTAL 17, именно она генерирует OUT файлы, которые рассматриваются в данной работе [15]. CRYSTAL – это компьютерная программа для расчета электронной структуры молекул и кристаллических твердых тел. Она использует методики теории функционала плотности (DFT) и методы теории функционала плотности на основе плоских волн (PW). Она может решать широкий спектр задач в области теоретической и вычислительной химии, физики твердого тела, катализа и других областей науки и техники.

Программа позволяет проводить расчеты на основе различных уровней теории, включая гибридные функционалы, локальные функционалы и функционалы с использованием металлов. Она может работать с различными типами систем, включая органические и неорганические молекулы, металлические и полупроводниковые материалы, а также с гетероструктурами и кластерами. Имеет удобный и гибкий пользовательский интерфейс, который позволяет легко настраивать параметры расчета, загружать структуры и анализировать результаты. Она также предоставляет мощные инструменты для визуализации и анализа данных. CRYSTAL 17 является одной из наиболее популярных программ для расчета электронной структуры и часто используется в исследованиях, связанных с дизайном новых материалов, исследованием реакционных механизмов, катализом и других областях науки и техники.

OUT файлы, полученные с помощью CRYSTAL 17, можно разделить на следующие категории:

Оптимизация - OUT файл содержит информацию о структуре, которая является оптимизированной в рамках заданных критериев, таких как энергия, силы, градиенты и др. Результаты оптимизации могут включать энергетические параметры, координаты атомов, величины сил и градиентов.

Рамановский расчета - OUT файл содержит информацию о частотах колебаний, интенсивностях и поляризациях связанных с модами колебаний в системе. Эти данные позволяют анализировать молекулярные и кристаллические колебания, связанные с определенными модами.

Расчет матрицы Гесса – OUT файл после расчета содержит информацию о вторых производных потенциальной энергии по отношению к атомным координатам. Это позволяет определить геометрические и энергетические свойства системы, такие как частоты колебаний, собственные значения и векторы хессиана, а также информацию о потенциальных энергиях и силовых константах.

Другие расчеты – помимо представленных выше, crystal также может генерировать другие out файлы с расчетами электронной структуры, спектроскопических и магнитных свойств, а также некоторые другие. Но в общем объеме информации они все вместе занимают незначительную часть, поэтому в текущей реализации их поддержка не подразумевается.

В файлах оптимизации содержится следующая информация:

– энергия: Файл оптимизации структуры обычно содержит информацию об энергии системы после оптимизации. Это может быть общая энергия системы, энергия электронов, энергия решетки и другие связанные энергетические параметры;

– координаты атомов: Файл содержит координаты атомов после процедуры оптимизации. Это могут быть трехмерные координаты каждого атома в системе, указанные в определенной единице измерения, такой как ангстремы;

– силы и градиенты: Файл может содержать информацию о силах и градиентах, которые были использованы в процессе оптимизации структуры. Это могут быть силы, действующие на каждый атом, а также значения градиентов энергии по отношению к каждому атомному координату;

– параметры оптимизации: Файл может содержать информацию о параметрах, используемых в процессе оптимизации структуры. Это могут быть параметры сходимости, критерии остановки, шаги оптимизации и другие настройки, которые определяют процесс оптимизации;

– информация о симметрии: Файл может содержать информацию о симметрии системы после оптимизации. Это может быть указано с использованием точных симметричных операций, группы пространственной группы и других связанных параметров;

Кроме перечисленных элементов, файл оптимизации структуры может содержать другую дополнительную информацию, в зависимости от параметров и настроек, указанных в расчете.

В файлах Рамановского расчета содержится следующая информация:

– частоты колебаний: Файлы Рамановского расчета содержат информацию о частотах колебаний, связанных с модами колебаний в системе. Это могут быть значения частот, указанные в см^-1 (сантиметры в минус первой степени) или другой соответствующей единице измерения;

– интенсивности: Файлы могут содержать информацию об интенсивностях связанных с модами колебаний. Интенсивности отражают относительную силу каждой моды колебания и могут быть выражены в арбитражных единицах или других единицах интенсивности;

– поляризация: Файлы могут содержать информацию о поляризации связанных с модами колебаний. Поляризация указывает направление колебаний в пространстве и может быть описана с использованием соответствующих символов поляризации, таких как A, B, C и других;

– амплитуды колебаний: Файлы могут содержать информацию о величине амплитуд колебаний, связанных с каждой модой. Это может быть числовое значение, указывающее величину колебательных движений каждого атома в системе;

– разрешенные и запрещенные моды: Файлы могут указывать, какие моды являются разрешенными и могут наблюдаться в Рамановском спектре, а также какие моды являются запрещенными и не наблюдаются. Это связано с симметрией системы и выбором допустимых переходов;

Кроме перечисленных элементов, файлы Рамановского расчета могут содержать другую дополнительную информацию, включая спектральные интенсивности, ширины линий, фазовые факторы и другие связанные параметры.

В файлах расчета матрицы Гесса содержится следующая информация:

– размерность матрицы: Файлы содержат информацию о размерности матрицы Гесса, которая определяется количеством атомов в системе и количеством координат, связанных с каждым атомом. Размерность матрицы может быть выражена числом атомов или размером матрицы;

– значения элементов матрицы: Файлы содержат числовые значения элементов матрицы Гесса. Это значения вторых производных энергии по отношению к координатам атомов, которые характеризуют силы и скорости изменения энергии системы при изменении положения атомов;

– симметрия матрицы: Файлы могут содержать информацию о симметрии матрицы Гесса, если применялись соответствующие симметричные методы расчета. Это может быть указано с использованием соответствующих символов симметрии или других связанных параметров;

– массы атомов: Файлы могут содержать информацию о массах атомов, используемых при расчете матрицы Гесса. Массы атомов влияют на значения элементов матрицы и учитываются при оценке вибрационных свойств системы;

Кроме перечисленных элементов, файлы расчета матрицы Гесса могут содержать другую дополнительную информацию, в зависимости от параметров и настроек, указанных в расчете.

Все получаемые данные могут быть использованы для интерпретации экспериментальных данных, понимания механизмов химических реакций и проектирования новых материалов с определенными свойствами.

Особо следует отметить, что из-за отсутствия ограничений по формированию таких файлов они содержат значительно больше информации и далеко не вся из них полезна. Crystal 17 в том числе обладает функционалом оптимизации и выходным форматом является OUT файл с множеством итераций, полезными из которых являются первая и последняя, что может составлять всего 3% от всего файла. Остальное является служебной информацией, которая нужна крайне редко. Также существуют другие файлы для хранения химической информации, ниже представлен краткий обзор на них. В данной работе их обработка также не выполнена, но при дальнейшем поддержании проекта планируется добавить и их поддержку.

2.3 PDB (Protein Data Bank)

Формат Protein Data Bank (PDB) является стандартом для хранения информации о трехмерной структуре белков, нуклеиновых кислот и других биомолекул [16]. Подробное описание PDB формата:

– заголовок (HEADER): Строка, начинающаяся с "HEADER", содержит информацию о файле, включая дату, авторов и заголовок структуры;

– идентификаторы (COMPND, SOURCE): Строки, начинающиеся с "COMPND" и "SOURCE", содержат информацию об идентификации биомолекулы, такую как ее название, организм и источник;

– атомы (ATOM, HETATM): Строки, начинающиеся с "ATOM" и "HETATM", содержат информацию об атомах в структуре, включая их идентификаторы, типы, координаты XYZ, температурные факторы и др.;

– связи (CONECT): Строки, начинающиеся с "CONECT", определяют связи между атомами в структуре;

– кристаллическая решетка (CRYST1): Строка, начинающаяся с "CRYST1", содержит информацию о параметрах кристаллической решетки, такие как размеры ячейки и углы между основными векторами;

– аннотации (REMARK): Строки, начинающиеся с "REMARK", содержат дополнительные аннотации и комментарии к структуре, такие как методы экспериментального определения, обработка данных и т.д.;

– информация об авторах (AUTHOR, JRNL): Строки, начинающиеся с "AUTHOR" и "JRNL", содержат информацию об авторах и публикациях, связанных с структурой.;

– формат файла (FORMAT): Строка, начинающаяся с "FORMAT", определяет версию формата PDB и правила форматирования;

PDB файлы обычно содержат одну или несколько биомолекул. В случае комплексных структур, каждая биомолекула может быть описана отдельными записями в файле.

PDB файлы также могут содержать дополнительные разделы, такие как SEQRES (последовательность аминокислот или нуклеотидов), HELIX (информация о спиральных участках), SHEET (информация о бета-листах) и др.

2.4 MOL (MOLfile)

Формат MOL (Molecular File Format) является текстовым форматом, используемым для представления информации о химической структуре молекулы [17]. Подробное описание MOL формата:

– строка с заголовком (Header): Первая строка файла содержит информацию о заголовке молекулы, обычно это имя или идентификатор молекулы;

– количество атомов и связей (Counts): Во второй строке содержится два числа - количество атомов в молекуле и количество связей между атомами;

– координаты атомов (Atom Block): Следующие строки описывают атомы в молекуле. Каждая строка содержит информацию об атоме, включая его атомный символ, координаты XYZ, и, возможно, дополнительные параметры, такие как заряд и масса;

– информация о связях (Bond Block): после блока с атомами следуют строки, описывающие связи между атомами. Каждая строка указывает индексы связанных атомов и тип связи;

– блок дополнительных данных (Properties Block): В MOL файле может присутствовать блок с дополнительными данными, в котором указываются различные свойства молекулы, такие как их идентификаторы, имена, дескрипторы и другая информация;

MOL формат позволяет представить химическую структуру молекулы и основные параметры, связанные с ней. Он широко используется в химической информатике и программном обеспечении для обмена и хранения данных о молекулах.

Кроме основной информации, MOL файлы также могут содержать дополнительные поля, такие как информация о зарядах, массах, стереохимии, изомерии и другие параметры, в зависимости от программного обеспечения и контекста использования формата.

2.5 XYZ

Формат XYZ (Cartesian Coordinate File Format) является простым текстовым форматом, используемым для представления трехмерных координат атомов без дополнительной информации о молекуле [18]. Подробное описание формата XYZ:

– количество атомов (Number of atoms): Первая строка файла содержит целое число, указывающее количество атомов в молекуле;

– комментарий (Comment line): Вторая строка файла содержит комментарий, который может содержать дополнительную информацию о молекуле или файле. Эта строка не является обязательной и может быть пропущена;

– координаты атомов (Atom coordinates): Следующие строки содержат информацию об атомах. Каждая строка содержит атомный символ и координаты X, Y и Z атома, разделенные пробелами или табуляцией. Координаты могут быть выражены в ангстремах или других подходящих единицах измерения;

Формат XYZ обычно используется для быстрого и простого обмена координатами атомов и отображения молекул в 3D-программах или визуализаторах. Однако он не содержит дополнительной информации о связях, типах атомов, зарядах и других параметрах, поэтому может быть ограничен для более сложных задач химического моделирования и анализа.

2.6 SDF (Structure-Data File)

SDF (Structure-Data File) - это формат файлов, используемый для хранения информации о химических соединениях. SDF-файлы обычно содержат молекулярные структуры соединений, а также связанные с ними данные, такие как свойства, идентификаторы, дополнительные атрибуты и т. д. Формат SDF является одним из наиболее распространенных и широко используемых форматов для обмена и хранения химических данных.

SDF-файлы состоят из набора блоков, каждый из которых представляет отдельное химическое соединение. Блоки имеют следующую структуру: заголовок, блок структуры и свойства.

SDF-файлы могут содержать несколько блоков, представляющих различные химические соединения. Это позволяет удобно хранить и обмениваться информацией о большом числе соединений в одном файле.

Существует множество форматов химических данных, каждый из них нужен для определенных целей. В данном случае были выбраны форматы CIF и OUT, так как они востребованы заказчиком, а именно лабораторией многомасштабного моделирования многокомпонентных функциональных материалов.

3 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

3.1 Технические

1) Возможность работы системы на серверах Южно-Уральского государственного Университета.
2) Отсутствие необходимости специфического серверного оборудования.

3.2 Функциональные

1) Извлечение полезных данных из CIF файлов.
2) Извлечение полезных данных из OUT файлов.
3) Структурирование полученных данных.
4) Внесение полученной информации в советующие таблицы базы данных.
5) Поиск по полученным данным в БД
6) Для взаимодействия с интерфейсом использовать API.

3.3 Системные

Возможность масштабирования при больших объёмах загружаемой информации. Объём данных не регламентируется, так как напрямую зависит от количества загружаемых данных.

3.4 Подсистемы

1) Парсер CIF файлов;
2) Парсер OUT файлов;
3) Модуль взаимодействия с базой данных;
4) Application Programming Interface (программный интерфейс приложения.

3.4.1. Требования к подсистеме «Парсер CIF файлов»

1) Выделение полезных данных из исходного файла;
2) Разбиение полезной информации на заранее объявленные группы;
3) Строгое сопоставление параметра к группе. В одной группе может содержаться неограниченное количество параметров;
4) Выходными данными является структура в формате JSON, содержащая все группы, входящие в них параметры и дополнительно координаты атомов вещества.

3.4.2. Требования к подсистеме «Парсер OUT файлов»

1) Определение внутренней структуры OUT файла.

2) Для каждого из типов OUT файлов необходимо выделить значимые части.

3) Выходными данными является JSON файл.

3.4.3. Требования к подсистеме «Модуль взаимодействия с БД»

1) Данные полученные из парсеров должны заполнять собой таблицы в базе данных для осуществления последующего поиска по ней.

3.4.4. Требования к подсистеме API

Необходимо реализовать следующие запросы:

«/api/upload_cif_file/» - в запросе передаются файлы в бинарном виде, после чего им присваивается идентификатор и происходит парсинг. Полученные данные заносятся в соответствующую таблицу в базе данных.

«/api/upload_out_file/» - в запросе передаются файлы в бинарном виде, после чего им присваивается идентификатор и происходит парсинг. Полученные данные заносятся в соответствующую таблицу в базе данных.

«/api/get_columns/» - для работы с CIF файлами реализовать передачу названий групп, в которых будут располагаться полученные из парсера данные.

«/api/output_data_cif/» - запрос, ответом на который передаются значения из таблицы с необходимыми (переданными в запросе) данными.

«/api/output_data_out/» - запрос, ответом на который передаются значения из таблицы с необходимыми (переданными в запросе) данными.

«/api/download_file/» - запрос, ответом на который является возвращение бинарного файла, загруженного в систему.

3.5 Нефункциональные требования системы

Простота в разработке и доработке. Для этого использовать популярную связку язык программирования и фреймворк. Возможны вставки на других языках программирования.

Выбрать базу данных, распространяющуюся по открытой лицензии.

4 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

4.1 Процедура обработки данных

Пользователь загружает файлы в форматах CIF и OUT. Далее эти файлы передаются на соответствующие парсеры.

Оба парсера выделяют полезную информацию из исходных данных. Для CIF файлов информация дополнительно разделяется на группы. После чего данные помещаются в БД. Исходному файлу присваивается внутренний идентификатор системы и остается возможность его загрузить в случае, если исследователю будет нужна более подробная информация, а не только выделенная парсером.

4.2 Архитектура информационной системы

Схематичное представление информационной системы представлено на рисунке 1.

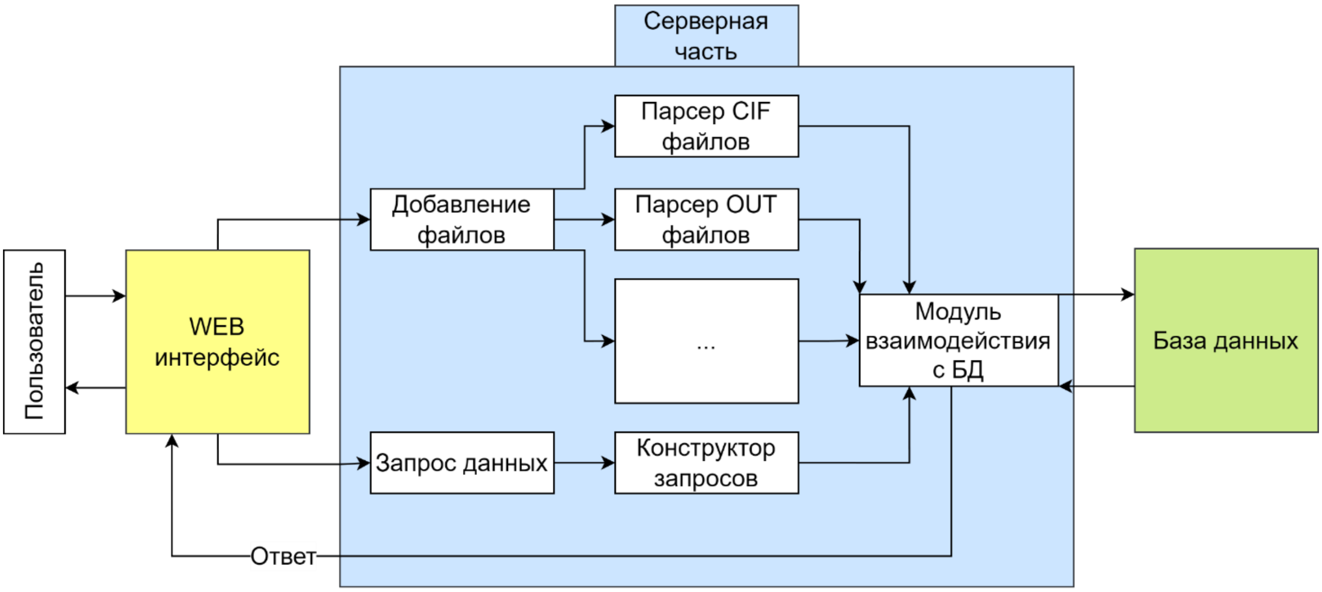

Рисунок 1 - Архитектура информационной системы

4.3 Выбор языка программирования

Язык программирования — это формальный язык, предназначенный для написания компьютерных программ. Языки программирования делятся на различные типы в зависимости от их применения и назначения.

Для разработки веб-сервиса можно использовать следующие языки программирования:

4.3.1. Python

Python — это высокоуровневый язык программирования, который был разработан в конце 1980-х годов Гвидо ван Россумом. Он является одним из самых популярных языков программирования в мире и широко используется в различных областях, таких как наука о данных, искусственный интеллект, веб-разработка и многое другое [19].

Особенности Python:

– простой и интуитивный синтаксис. Python имеет простой и понятный синтаксис, который делает его легко читаемым и понятным для новых пользователей;

– высокая скорость разработки. Python имеет множество библиотек и фреймворков, которые ускоряют процесс разработки, делая его быстрее и эффективнее;

– мультипарадигмальный язык. Python поддерживает различные стили программирования, такие как объектно-ориентированный, функциональный и структурный;

– переносимость. Python работает на множестве платформ, таких как Windows, Mac OS и Linux, что делает его переносимым и доступным для использования на различных устройствах.

Большое сообщество. Python имеет огромное сообщество разработчиков, которые создают сторонние библиотеки и фреймворки, расширяя его возможности. Два самых популярных фреймворка это Django и Flask.

4.3.2. Ruby

Ruby — это интерпретируемый, объектно-ориентированный язык программирования, который был разработан в Японии в 1995 году Юкихиро Мацумото. Ruby имеет простой и интуитивный синтаксис, который делает его популярным среди начинающих программистов. Он также обладает множеством библиотек и фреймворков, которые упрощают процесс разработки, делая его быстрее и эффективнее [20].

Особенности Ruby:

– простой и понятный синтаксис. Ruby имеет простой и интуитивный синтаксис, который делает его легко читаемым и понятным для новых пользователей;

– динамическая типизация. Ruby является динамически типизированным языком, что означает, что тип переменной определяется во время выполнения программы, а не во время компиляции;

– объектно-ориентированный язык. Ruby полностью объектно-ориентированный язык программирования, что означает, что все в Ruby является объектом;

– мультипарадигмальный язык. Ruby поддерживает различные стили программирования, такие как функциональный, объектно-ориентированный и процедурный;

– большое сообщество. Ruby имеет огромное сообщество разработчиков, которые создают сторонние библиотеки и фреймворки, расширяя его возможности.

Ruby также широко используется для разработки веб-приложений, благодаря своим фреймворкам, таким как Ruby on Rails. Ruby on Rails (RoR) — это открытый и бесплатный фреймворк, который был разработан для создания крупных веб-приложений. Он предоставляет разработчикам множество готовых компонентов и инструментов для работы с базами данных, аутентификации пользователей, управления контентом и других задач, связанных с веб-разработкой. RoR также имеет простой и интуитивный синтаксис, который делает его популярным среди программистов и ускоряет процесс разработки.

Таким образом, Ruby — это мощный и гибкий язык программирования, который может быть использован для создания различных приложений, в том числе и веб-сервисов, с помощью его фреймворка Ruby on Rails. PHP - язык программирования, который был создан специально для разработки веб-сервисов. PHP позволяет легко создавать динамические веб-страницы и работать с базами данных.

4.3.3. Java

Java — это высокоуровневый, объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (теперь часть Oracle Corporation). Java создавался как язык программирования, который может быть запущен на любой платформе, независимо от аппаратного и программного обеспечения. Это достигается благодаря использованию виртуальной машины Java (JVM), которая интерпретирует байт-код, созданный из исходного кода Java [21].

Java был создан в 1995 году Джеймсом Гослингом и его командой. Язык был разработан с целью облегчения разработки программного обеспечения, уменьшения ошибок в программном коде и упрощения портирования приложений между разными платформами.

Особенности Java:

– кроссплатформенность. Java код может быть запущен на любой платформе, поддерживающей виртуальную машину Java;

– объектно-ориентированность. Java полностью объектно-ориентированный язык программирования, что означает, что все в Java является объектом;

– мультипарадигмальный язык. Java поддерживает различные стили программирования, такие как объектно-ориентированный, процедурный и функциональный;

– безопасность. Java обеспечивает высокий уровень безопасности благодаря своей системе безопасности, которая позволяет запускать код в защищенной среде;

– большое сообщество. Java имеет огромное сообщество разработчиков, которые создают множество библиотек и фреймворков, расширяя его возможности.

Java также широко используется для создания веб-приложений и веб-сервисов. Для разработки веб-приложений на Java используется множество фреймворков, таких как Spring, Struts и JavaServer Faces (JSF). Фреймворк Spring, например, предоставляет множество инструментов для создания веб-приложений и веб-сервисов, включая управление зависимостями, автоматическую конфигурацию и интеграцию с базами данных.

Таким образом, Java является мощным и гибким языком программирования, который может быть использован для создания различных приложений, в том числе и веб-сервисов, с помощью множества доступных библиотек и фреймворков.

4.3.4. C#

C# — это объектно- и компонентно-ориентированный язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО [22].

Написание Веб сервиса на этом языке подразумевает платформу ASP.NET

ASP.NET (Active Server Pages для .NET) — платформа разработки веб-приложений, в состав которой входит: веб-сервисы, программная инфраструктура, модель программирования, от компании Майкрософт. ASP.NET входит в состав платформы .NET Framework и является развитием более старой технологии Microsoft ASP.

.NET Framework — это программная платформа, выпущенная компанией Microsoft, которая подходит для разных языков программирования.

Считается, что платформа .NET Framework явилась ответом компании Microsoft на набравшую к тому времени большую популярность платформу Java. ASP.NET основывается на Common Language Runtime: разработчики могут писать код для ASP.NET, используя практически любые языки программирования, некоторые из которых входят в комплект .NET Framework (C#, Visual Basic.NET и JScript .NET), а другие могут быть установлены дополнительно (IronRuby, IronPython, PHP, Perl, Smalltalk, Haskell и др.).

Некоторые особенности ASP.NET:

1 компилируемый код выполняется быстрее, а большинство ошибок отлавливается ещё на стадии разработки;

2 расширяемый набор элементов управления и библиотек классов, ускоряющий разработку;

3 возможность кэширования всей страницы, её частей или данных, используемых на странице;

4 возможность разделения визуальной части и бизнес-логики по разным файлам, есть возможность выделять часто используемые шаблоны пользовательских элементов управления, таких как меню сайта, наличие master-страниц для задания шаблонов оформления, поддержка AJAX (расширение ASP.NET AJAX);

5 расширяемые модели событий, обработки запросов и серверных элементов управления;

6 поддержка CRUD-операций при работе с таблицами через GridView;

7 возможно создание веб-приложений, которые реализуют шаблон Model-View-Controller (ASP.NET MVC Framework).

.NET достаточно широко распространён в сфере разработки внутрикорпоративных программных продуктов, но в веб-разработке всё же встречается относительно редко, как и другие программные продукты корпорации Microsoft. Использование .NET «тянет» за собой покупку и иного ПО от корпорации Microsoft (серверной ОС, СУБД и т.п.). Технология достаточно дорогая в разработке и сопровождении: кроме затрат на покупку лицензий на необходимое ПО существенный вклад в бюджет проектов вносят высокие зарплаты разработчиков.

Было принято решение использовать Python, так как:

– Python имеет довольно простой синтаксис;

– множество открытых библиотек для работы с данными;

– есть опыт разработки на этом языке;

– возможность подключения интерактивной машины для взаимодействия с базой напрямую;

– рекомендация заказчика.

Фреймворком был выбран Flask, так как:

Flask — это микрофреймворк для создания веб-приложений на языке программирования Python. Он был создан Армином Роначером в 2010 году и стал одним из самых популярных фреймворков для веб-разработки на Python. Flask имеет минимальные зависимости и позволяет разработчикам создавать веб-приложения с минимальным количеством кода.

В Flask нет жестких правил и конвенций, поэтому разработчики имеют полную свободу выбора структуры и архитектуры своих приложений. Фласк использует подход "без батареек", который означает, что он не содержит заранее определенных шаблонов проектирования или библиотек, необходимых для разработки приложения. Вместо этого Flask предоставляет основные инструменты, необходимые для создания веб-приложения, такие как маршрутизация запросов, обработка HTTP-запросов и ответов, работа с формами, сессиями и базами данных.

Несколько преимуществ использования Flask для создания веб-приложений:

– легковесность - имеет небольшой размер и минимальные зависимости, что делает его идеальным выбором для создания небольших проектов и прототипов;

– гибкость - не навязывает жестких правил и конвенций, поэтому разработчики имеют полную свободу выбора структуры и архитектуры своих приложений;

– легкость в освоении - имеет простой и интуитивно понятный синтаксис, что делает его легко осваиваемым для новых разработчиков;

– расширяемость - имеет большое сообщество разработчиков, которые создают сторонние расширения, плагины и библиотеки, которые могут использоваться в приложениях Flask. Это значительно расширяет возможности фреймворка и позволяет разработчикам создавать более сложные веб-приложения;

– быстрота разработки - благодаря простоте и гибкости Flask, разработка веб-приложений может быть выполнена очень быстро и эффективно;

– хорошая документация - Flask имеет хорошо организованную и понятную документацию, что делает его привлекательным.

4.4 Выбор среды разработки

IDE (или интегрированная среда разработки) — это программа, предназначенная для разработки программного обеспечения. Как следует из названия, IDE объединяет несколько инструментов, специально предназначенных для разработки. Эти инструменты обычно включают редактор, предназначенный для работы с кодом. Например, подсветка синтаксиса и автодополнение, инструменты сборки, выполнения и отладки; и определённую форму системы управления версиями.

Большинство IDE поддерживают множество языков программирования и имеют много функций, из-за чего могут быть большими, занимать много времени для загрузки и установки и требуют глубоких знаний для правильного использования.

С другой стороны, есть редакторы кода, которые представляют собой текстовый редактор с подсветкой синтаксиса и возможностями форматирования кода. Большинство хороших редакторов кода могут выполнять код и использовать отладчик, а лучшие даже могут взаимодействовать с системами управления версиями. По сравнению с IDE, хороший редактор кода, как правило, легковесней и быстрее, но зачастую ценой меньшей функциональности.

4.4.1. Eclipse

Eclipse является бесплатной программной платформой с открытым исходным кодом, контролируется организацией Eclipse Foundation. Написана на языке программирования Java и основной целью её создания является повышение продуктивности процесса разработки программного обеспечения [23].

Для разработки на языке программирования python есть специальные плагины, самый популярный из них PyDev.

PyDev — надстройка над Eclipse, превращающая его в интегрированную среду разработки на Python.

Основные возможности:

– автопродолжение кода (Code completion);

– подсветка синтаксиса (Syntax highlighting);

– анализ кода (Code analysis);

– возможность перехода к определению функции (Go to definition);

– возможность рефакторинга (Refactoring);

– отладчик (Debugger);

– удалённый отладчик (Remote debugger);

– интерактивная консоль (Interactive console);

– интеграция с unittest-ами (Unittest integration);

– покрытие кода тестами (Code coverage).

Преимущества: если есть опыт работы с этой IDE, то установка плагина и его освоение будет крайне простым и быстрым.

Недостатки: если пользователь только начинает изучать Python или разработку в целом, Eclipse может стать непосильной ношей.

4.4.2. Sublime Text

Sublime Text, написан инженером из Google с мечтой о лучшем текстовом редакторе. Является весьма популярным редактором кода для python. Доступен на всех платформах. Sublime Text имеет встроенную поддержку редактирования Python-кода, а также богатый набор расширений, называемых пакетами, которые расширяют возможности синтаксиса и редактирования [24].

Установить дополнительный Python-пакет может быть непросто — все пакеты Sublime Text написаны на Python, поэтому для установки пакетов сообщества зачастую может потребоваться выполнить Python-скрипт непосредственно в редакторе.

Преимущества: у Sublime Text большое количество поклонников. Как редактор кода, Sublime Text быстрый, лёгкий и имеет хорошую поддержку.

Недостатки: Sublime Text не является бесплатным, хотя вы можете использовать пробный период сколько угодно. Установка расширений может создать множество проблем. Кроме того, в редакторе нет поддержки отладки и запуска кода.

4.4.3. PyCharm

Одной из лучших полнофункциональных IDE, предназначенных именно для Python, является PyCharm. Существует как бесплатный open-source

(Community), так и платный (Professional) варианты IDE. PyCharm доступен на Windows, Mac OS X и Linux [25].

PyCharm «из коробки» поддерживает разработку на Python напрямую — откройте новый файл и начинайте писать код. Вы можете запускать и отлаживать код прямо из PyCharm. Кроме того, в IDE есть поддержка проектов и системы управления версиями.

Преимущества: это среда разработки для Python с поддержкой огромного функционала, хорошо взаимодействующего друг с другом. Также среда может похвастаться большим и отзывчивым сообществом. В ней «из коробки» можно редактировать, запускать и отлаживать Python-код.

Недостатки: PyCharm может медленно загружаться, а настройки по умолчанию, возможно, придётся подкорректировать для существующих проектов.

В результате анализа сред разработки, выбрана PyCharm. Она предоставляет возможность создания виртуального окружения и локального виртуального сервера, что позволяет не устанавливать дополнительного ПО, такого как nginx или apache. Также имеет все возможности текстового редактора файлов форматов HTML, CSS и другие.

4.5 Выбор базы данных и системы управления базой данных

База данных является одним из важнейших компонентов веб сервиса. От неё зависит сложность разработки, быстродействие всей системы и отказоустойчивость проекта. В данной работе планируется использовать реляционную базу данных. На сегодняшний день существует множество различных баз данных, рассмотрим несколько из них: MySql, PostgreSql, Access, MariaDB, SQLite.

MySQL – это «Open Source» проект, который почти во всех случаях может быть использован бесплатно, разработчики лишь рекомендуют приобрести лицензию если эта разработка приносит деньги в бизнесе. Это способствует развитию проекта. Также плюсами этого проекта, является то, что MySQL может быть запущен как на персональных компьютерах под управления операционной системы MS Windows, так и на серверном оборудований с высокой мощностью или малых серверах развернутых на Linux дистрибутивах [26]. Система управления базой данных обладает отличной переносимостью. Она зарекомендовала себя благодаря тому, что многие большие проекты такие как: Facebook, Google, Twitter используют её.

PostgreSQL - это мощная объектно-реляционная база данных с открытым исходным кодом, активная разработка которой насчитывает более 30 лет, что принесло ей прочную репутацию за надежность, функциональность и производительность.

В официальной документации можно найти огромное количество информации, описывающей, как установить и использовать PostgreSQL [27]. С подробной таблицей характеристик можно ознакомиться на официальном сайте разработчика

Microsoft Office Access или просто Microsoft Access — реляционная система управления базами данных (СУБД) корпорации Microsoft. Входит в состав пакета Microsoft Office. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA, в самой Access можно писать приложения, работающие с базами данных. Подходит скорее для учебных проектов или проектов малого бизнеса, язык VBA является устаревшим, и в целом СУБД для серьезных проектов подходит не в полной мере из-за своих ограничений [28].

MariaDB Server - одна из самых популярных реляционных баз данных с открытым исходным кодом. Она сделана оригинальными разработчиками MySQL и имеет открытый исходный код. MariaDB входит в состав большинства облачных решений и используется по умолчанию во многих дистрибутивах Linux.

MariaDB Server основан на ценностях производительности, стабильности и открытости. Недавние новые функции включают расширенную кластеризацию с Galera Cluster 4, функции совместимости с Oracle Database и Temporal Data Tables, позволяющие запрашивать данные в том виде, в котором они были в любой момент в прошлом [29].

SQLite – компактная СУБД с открытым исходным кодом. Основным отличием является архитектура. В основном СУБД являются клиент-серверными приложениями, то есть база данных работает как отдельный процесс, а приложение к нему подключается. SQLite является встраиваемой СУБД, то есть является отдельным файлом до 140 ТБ. Обычно её используют в мобильной разработке. Также её можно использовать для небольших учебных проектов, но стоит учитывать ограниченный функционал. При создании проекта Django в среде PyCharm SQLite является базой по умолчанию [30].

В результате исследования баз данных была составлена сравнительная таблица (Таблица 2).

Таблица 2 – Сравнение исследуемых баз данных

Критерий / Название	**Открытый код**	**Гибкость запросов**	**Надежность**	**Возможность дописать ядро**
MySQL	+	+	+	-
PostgreSQL	+	+	+	+
Microsoft Access	-	-	-	-
MariaDB	+	-	-	-
SQLite	+	-	-	-

В результате анализа баз данных и их взаимодействие с языком программирования python была выбрана PostgreSQL.

5 РАЗРАБОТКА

5.1 Разработка парсера CIF файлов

При разработке парсера CIF файлов были рассмотрены уже имеющиеся инструменты. Анализ найденных библиотек показал, что в каждой из них имеются свои недостатки. Сравнительный анализ представлен в таблице 3.

Таблица 3 – Результат тестирования библиотек

Название	Преимущество	Проблема
Pyemaps [31]	Последняя дата релиза 27 нояб. 2022 г.	«AttributeError: module 'pyemap' has no attribute 'parse'»
Pyemap [32]	Используется в проекте eMAP	«[Errno 13] Permission denied» Плохо подходит для CIF, больше для PDB
Nomad exampleparser [33]	Набор инструментов для создания парсера по стилю и функционалу похожему на парсер в проекте NOMAD	«ERROR: After October 2020 you may experience errors when installing or updating packages. » ModuleNotFoundError: No module named 'resource'
ChemSpiPy [34]	Большой функционал, рекомендации от пользователей	Обработка на облаках проекта
cif-parsers [35]	Инструмент обещает простоту в использовании	Неверный формат входных данных, требует .gz
Pycifrw [36]	It was developed at the Australian National Beamline Facility (ANBF)	Библиотека для работы с файлами tar.gz
Pymatgen [37]	Существует интеграция с проектом Materials Project	ERROR: Failed building wheel for pymatgen. Только Python 3.8

По результатам исследования по согласованию с заказчиком была определена необходимость разработки собственного парсера CIF файлов.

Для эффективного поиска информации, она должна быть структурирована, это позволяет строить алгоритм по работе не над всей информацией, а только над той, в которой может содержаться искомая информация. В первоначальном виде cif это последовательный текстовый файл, и так как он разрабатывался в прошлом веке, он имеет некоторые моральные устаревания. Например, длина строки не превышает 80 символов, использование символов только таблицы ASCII.

На основе данных содержащихся в инструкции по формированию файлов, было выявлено что абсолютное большинство свойств начинаются со служебного символа «_» далее название свойства (параметра) и его значение через пробел. Например, _chemical_formula_iupack C34 H22 N4 O1 S1. Парсер был составлен на основе этой информации, то есть из файла извлекаются строки, представляющие собой полезную информацию в виде свойство (параметр): значение. Помимо этого, к инструкции прилагается эталонный cif файл с названием «example.cif». В нём перечислены основные группы имеющейся информации, их было решено взять за основу алгоритма деления. Перечень групп представлен в таблице 4.

Таблица 4 - Перечень групп параметров

№	Название группы	Содержание
1	default_group	Данные о дате создания документа
2	submission_details	Данные об авторах
3	processing_summary	Данные о журнале,в котором опубликовано исследование
4	title_and_author	Понятное название и аннотация
5	text	Данные к рисункам
6	chemical_data	Брутто формула, данные о симметрии, параметрах исследуемого кристалла и т.д.
7	refinement_data	Уточняющие данные, схема весов, параметры матриц, коэффициенты
8	atomic_coordinates_and_ displacement_parameters	Данные об атомах, их расположение и т.д.
9	molecular_geometry	Данные молекулярной геометрии

При рассмотрении множества файлов периодически возникают названия, не содержащиеся в example, при разработке эта группа была названа «unknown_group». Это связано с тем, что правила не строгие и разные исследователи могут дополнять или изменять имеющиеся имена параметров.

Для удобства распределения свойств (параметров) по группам было принято решение создать файл, описывающий структуру групп. Он представляет собой словарь, где каждому названию группы соответствует массив входящих в неё параметров (Листинг 1). Благодаря этому, можно довольно просто менять структуру файлов для их последующего более правильного хранения, а значит и поиска. Часть групп не содержит свойств, но они оставлены, так как конфигурация может еще меняться. После того, как система будет переведена на стадию тестирования можно будет удобно скорректировать принадлежность того или иного параметра к группе.

Листинг 1 - содержание файла structure.json

```
{"structure":
	{
	"default": ["_audit_creation_date"],
	"SUBMISSION DETAILS": ["_publ_contact_author_name", "_publ_contact_author_phone",
"_publ_contact_author_fax", "_publ_contact_author_email", "_publ_requested_journal",
"_publ_requested_category", "_publ_requested_coeditor_name"],
	"PROCESSING SUMMARY  (IUCr Office Use Only)": ["_journal_date_recd_electronic",
"_journal_date_to_coeditor", "_journal_date_from_coeditor", "_journal_date_accepted",
"_journal_date_printers_first", "_journal_date_printers_final", "_journal_date_proofs_out",
"_journal_date_proofs_in", "_journal_coeditor_name", "_journal_coeditor_code",
"_journal_techeditor_code", "_journal_coden_ASTM", "_journal_name_full", "_journal_year",
"_journal_volume", "_journal_issue", "_journal_page_first", "_journal_page_last",
"_journal_suppl_publ_number", "_journal_suppl_publ_pages"],
	"TITLE AND AUTHOR LIST":[""],
	"TEXT": ["_publ_section_figure_captions"],
	"Chemical Data": ["_chemical_name_common", "_chemical_melting_point",
"_chemical_formula_iupac", "_symmetry_Int_Tables_number", "_chemical_formula_moiety",
"_chemical_formula_sum", "_chemical_formula_weight", "_chemical_compound_source",
"_chemical_absolute_configuration", "_symmetry_cell_setting",
"_symmetry_space_group_name_Hall", "_cell_length_a", "_cell_length_b", "_cell_length_c",
"_cell_angle_alpha", "_cell_angle_beta", "_cell_angle_gamma", "_cell_volume",
"_cell_formula_units_Z", "_cell_measurement_temperature", "_cell_measurement_reflns_used",
"_cell_measurement_theta_min", "_cell_measurement_theta_max", "_exptl_crystal_size_max",
"_exptl_crystal_size_mid", "_exptl_crystal_size_min", "_exptl_crystal_density_meas",
"_exptl_crystal_density_diffrn", "_exptl_crystal_density_method", "_exptl_crystal_F_000",
"_exptl_absorpt_coefficient_mu", "_exptl_absorpt_correction_type",
"_exptl_absorpt_process_details", "_exptl_absorpt_correction_T_min",
"_exptl_absorpt_correction_T_max", "_diffrn_ambient_temperature",
"_diffrn_radiation_wavelength", "_diffrn_radiation_type", "_diffrn_radiation_source",
"_diffrn_radiation_monochromator", "_diffrn_measurement_device_type",
"_diffrn_measurement_method", "_diffrn_reflns_number", "_diffrn_reflns_av_R_equivalents",
"_diffrn_reflns_limit_h_min", "_diffrn_reflns_limit_h_max", "_diffrn_reflns_limit_k_min",
"_diffrn_reflns_limit_k_max", "_diffrn_reflns_limit_l_min", "_diffrn_reflns_limit_l_max",
"_diffrn_reflns_theta_min", "_diffrn_reflns_theta_max", "_diffrn_reflns_theta_full",
"_diffrn_measured_fraction_theta_max", "_diffrn_measured_fraction_theta_full",
"_reflns_number_total", "_reflns_number_gt", "_reflns_threshold_expression",
"_computing_data_collection", "_computing_cell_refinement", "_computing_data_reduction",
"_computing_structure_solution", "_computing_structure_refinement",
"_computing_molecular_graphics", "_computing_publication_material"],
	"Refinement Data": ["_refine_ls_structure_factor_coef", "_refine_ls_matrix_type",
"_refine_ls_weighting_scheme", "_atom_sites_solution_primary",
"_atom_sites_solution_secondary", "_refine_ls_extinction_method",
"_refine_ls_extinction_coef", "_refine_ls_number_reflns", "_refine_ls_number_parameters",
"_refine_ls_number_restraints", "_refine_ls_R_factor_all", "_refine_ls_R_factor_gt",
"_refine_ls_wR_factor_ref", "_refine_ls_wR_factor_gt", "_refine_ls_goodness_of_fit_ref",
"_refine_ls_restrained_S_all", "_refine_diff_density_max", "_refine_diff_density_min"],
	"Atomic Coordinates and Displacement Parameters":[""],
	"Molecular Geometry":[""],
	"AtomicCoordinates":[""]
	}
}
```

Результат этапа

Результатом данного этапа стал алгоритм парсинга CIF файлов, который решает все поставленные задачи. Сначала на вход поступает файл «structure.json», который формирует экземпляр класса Structure. Этот класс содержит перечень групп и принадлежащих им параметров. Благодаря гибкой структуре конфигурации параметров и их принадлежности к группам можно будет корректировать хранящиеся данные. Далее вызывается конструктор класса CifFile с двумя параметрами, это: экземпляр класса structure и путь до бинарного файла. После чего происходят процедуры парсинга: выделаются необходимые параметры и координаты атомов, для каждого из них создаются свои объекты для последующей работы отдельно с ними.

Так как обработка файла происходит только в момент загрузки, то полное его переписывание в массив строк не является очень трудоемкой задачей для сервера, по крайней мере пока файлов одномоментно не загружается тысячи и более.

Блок схема алгоритма представлена на рисунке 2.

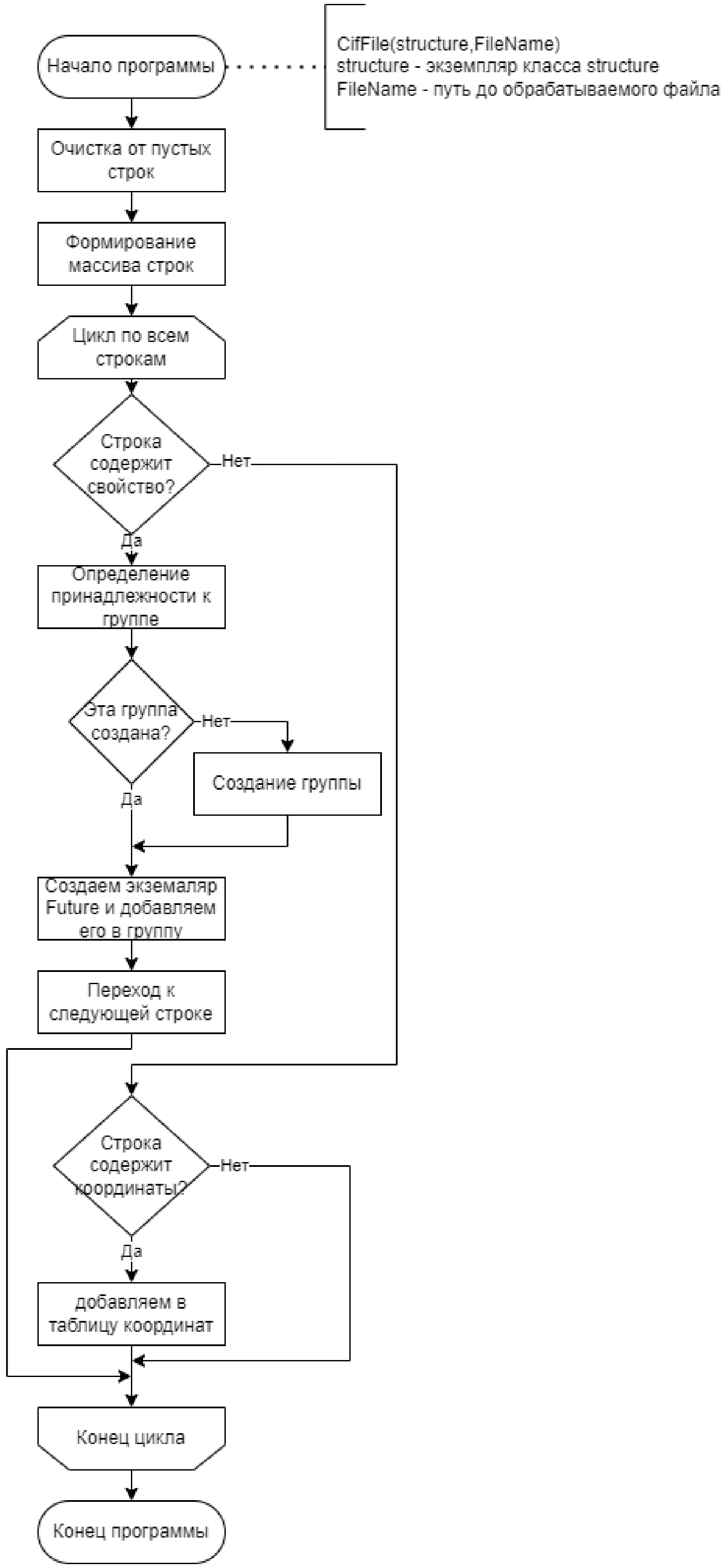

Рисунок 2 - Блок схема алгоритма парсера CIF файлов

5.2 Подключение фреймворка FLASK

На данном этапе нужно было продемонстрировать работоспособность идеи разделения на группы и численно определить ускорения. Для этого было принято решение разработать базовое приложение на языке программирования Python с использованием фреймворка Flask, которое позволило бы загружать CIF файлы, обрабатывать их и заполнять полученными данными базу данных. После чего осуществить процедуру поиска и замерить разницу во времени между последовательными файлами и полными.

Результат этапа

Созданы функции для взаимодействия с базой данных, а именно заполнение и поиск данных.

Был реализован весь необходимый функционал для генерации WEB интерфейса с помощью Flask, это позволило бы заказчику в удобном и понятном формате проверить работоспособность. В приложении отображалось поле поиска и результаты, полученные из базы данных, а также время выполнения запроса. Внешний вид интерфейса представлен на рисунке 3.

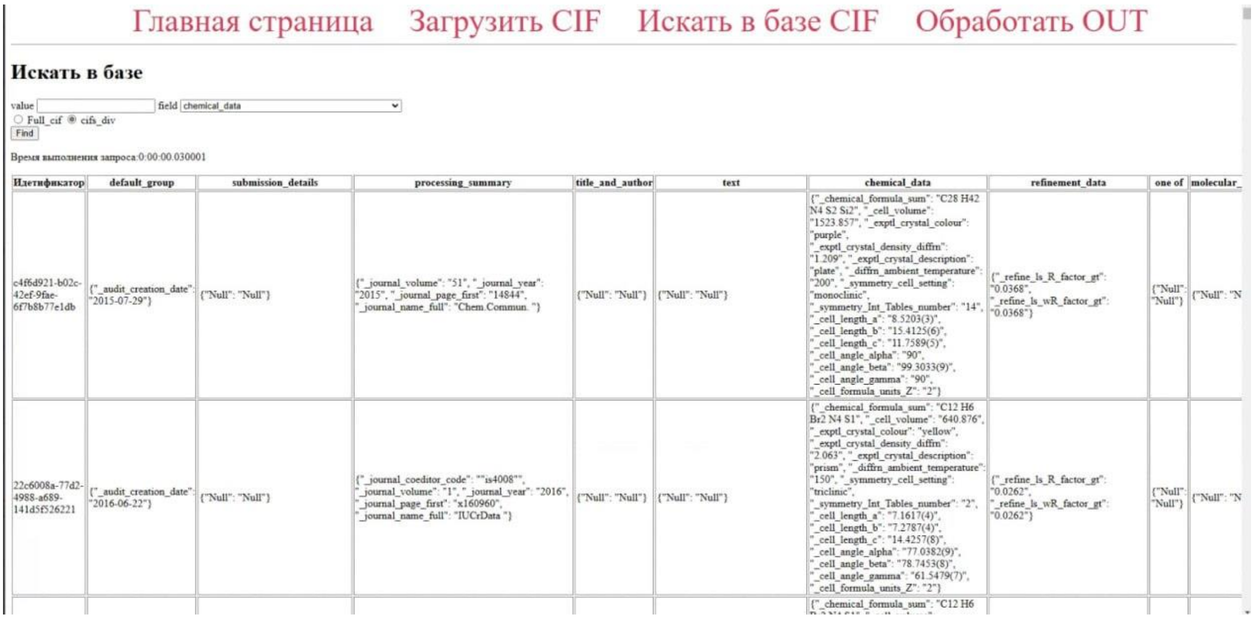

Рисунок 3 – Внешний вид базового интерфейса

После разработки базового интерфейса был проведен эксперимент, в котором поиск осуществлялся двумя способами. Первый это поиск по всему исходному документу, хранящемуся в базе данных. Алгоритм внутри базы данных ищет искомую подстроку в хранящихся данных. Второй это поиск в разделенном документе, тогда пользователь выбирает группу, в которой стоит осуществлять поиск.

Для примера был проведен эксперимент, который показал превосходство разделенной информации по сравнению с последовательной. Данные хранились в локально расположенной базе данных PostgreSQL, на твердотельном накопители. При работе не обычном жестком диске предполагается большее преимущество. Также при большем количестве данных в базе разделение будет давать всё больший и больший эффект.

Поиск осуществлялся в группе «Chemical Data», по брутто формуле: «C34 H22 N4 O1». Результаты экспериментов представлены в таблице 5.

Таблица 5 – Результаты эксперимента с CIF файлами

Количество элементов в базе	**Время поиска в полных файлах**	**Время поиска в разделенных файлах**	**Ускорение**
433	0:00:00.011000	0:00:00.001000	В 11 раз
4753	0:00:00.101000	0:00:00.004430	В 22 раза

5.3 Создание API

После демонстрации первых результатов заказчику стало понятно, что внешний вид — это отдельная задача и хотелось отделить ее от серверной части для снижения нагрузки. Решением стало разработать API интерфейс, чтобы можно было писать абсолютно независимый модуль «frontend» части.

Результат

В результате написан модуль управления приложением. В результате было написано несколько обработчиков входящих запросов. Они представлены ниже:

Загрузка cif файлов с клиента на сервер с адресом /api/upload_cif_file/. Входными данными является массив с названием «files», который содержит бинарные файлы. Пример выходных данных представлен в листинге 2.

Листинг 2 – Пример ответа на запрос /api/upload_cif_file/

```
{
    "DIVTUA.search2.cif": {
```

```
        "error": "",
        "filename": "DIVTUA.search2.cif",
        "id": "97a08db7-db06-4a03-b6c9-2eaf85588fff",
        "status": "successfully"
    },
    "DIWBAP.search2.cif": {
        "error": "уже есть в базе",
        "filename": "DIWBAP.search2.cif",
        "id": "9401bbb6-839c-43f2-9390-c0bbd027370c",
        "status": "error"
    }
}
```

Получения списка групп CIF файлов с адресом /api/get_columns/ Примеры входных и выходных данных представлены в листингах 3 и 4.

Листинг 3 – Пример запроса /api/get_columns/

```
{
  "fileType": "cif",
  "dataViewFormat": "divided"
}
```

Листинг 4 – Пример выходных данных на запрос /api/get_columns/

```
{
  "fileType": "CIF",
  "dataViewFormat": "divided",
  structure: [...]
}
```

Получение данных из базы данных, полученной из cif файлов с адресом /api/output_data_cif/ Примеры входных данных и выходных данных представлены в листингах 5 и 6.

Листинг 5 – Пример входных данных для запроса /api/output_data_cif/

```
{
    "count":20,
    "delimetr":1,
    "dataViewFormat":"divided",
    "structure":{
        "id":"",
        "filename":"",
        "default_group": "2014",
        "submission_details": "",
        "processing_summary": "",
        "title_and_author":"",
        "text": "",
        "chemical_data": "",
        "refinement_data":"",
        "atomic_coordinates_and_displacement_parameters":"",
        "molecular_geometry":"",
        "atomic_coordinates":"",
        "unknown_group":""
        }
}
```

Листинг 6 – Пример выходных данных для запроса /api/output_data_cif/

```
{
    "count": 20,
    "dataViewFormat": "divided",
    "data_from_DB": [
        {
```

```
            "id":"3d57fdfc-692f-40e1-a0e8-2b266d97d512",
            "filename":"PEBWOI.search2.cif",
            ......
        },
        {
            "id":"3d57fdfc-692f-40e1-a0e8-2b266d97d512",
            "filename":"PEBWOI.search2.cif",
            ......
        }
    ],
    "delimetr": 1,
    "execution_time": "0:00:00.001004",
    "filetype": "cif",
    resultsCount: 9,
}
```

Загрузка OUT файлов с клиента на сервер с адресом /api/upload_out_file/. Входными данными является массив с названием «files», который содержит бинарные файлы. Пример выходных данных представлен в листинге 7

Листинг 7 – Пример выходных данных запроса /api/upload_out_file/

```
{
    "1242.out": {
        "error": "",
        "filename": "1242.out ",
        "id": "97a08db7-db06-4a03-b6c9-2eaf85588fff",
        "status": "successfully"
    },
    "54546.out": {
        "error": "уже есть в базе",
        "filename": "54546.out.cif",
        "id": "9401bbb6-839c-43f2-9390-c0bbd027370c",
        "status": "error"
    }
}
```

Получение данных из БД, полученной из out файлов с адресом /api/output_data_out/ Примеры входных и выходных данных представлены в листингах 8 и 9.

Листинг 8 – Пример входных данных для запроса /api/output_data_out/

```
{
    "count":20,
    "delimetr":1,
    "dataViewFormat":"divided",
    "structure":{
        "keyWordTypeCalc":"",
        "SpaceGroup":"",
        "CrystalCellParam":"",
        "CountFreeAtoms":"",
        "TableFreeAtoms":"",
        "TypeCalc":"",
        "Atomonly":"",
        "TableBasisSet":"",
        "NonParse":"",
        "EAU":"",
        "BandGap":"3.5423",
        "Volume":"",
        "Density":"",
        "ParamsCellOpt":"",
```

```
        "OptCoordAtoms":""
        }
}
```

Листинг 9 – Пример выходных данных запроса /api/output_data_out/

```
{
  "count": 5,
  "delimeter": 5,
  "fileType": "CIF",
  "dataViewFormat": "divided",
  "structure": {}
}
```

Выгрузка любых файлов с сервера имеет адрес /api/download_file/ Входными данными является поле с ключом «id» и собственно идентификатором файла. Выходными данными в случае успеха является файл в бинарном виде. В случае ошибки возвращается поле с ключом «error» содержащее подробности ошибки.

5.4 Разработка парсера OUT файлов оптимизации

Началом этого этапа, как и с CIF файлами был поиск уже имеющихся инструментов, но таких библиотек и программных продуктов найдено не было, поэтому модуль разрабатывался самостоятельно.

OUT файлы оптимизации как правило являются очень длинными, в них записано множество итераций от исходных значений до финальных. Вся информация представлена в кодировке ASCII.

Результат этапа

Был разработан алгоритм для парсинга OUT файлов, содержащих расчет оптимизации, который решает все поставленные задачи.

5.5 Подключение OUT парсера в общую систему

Так как функции для взаимодействия с базой данных для CIF файлов были уже разработаны, то они были скорректированы для текущих данных. Тоже самое касается и веб интерфейса, больших отличий в данных не было, поэтому небольшое изменение алгоритмов дало необходимый эффект.

Результат этапа

Проведена процедура поиска. Поиск осуществлялся также двумя способами, но по другим параметрам. Для файлов оптимизации очень важным параметром является ширина запретной зоны «band_gap» именно по нему и осуществлялся поиск. Величиной поиска являлось число с плавающей запятой «3.4231» Результаты экспериментов представлены в таблице 6.

Таблица 6 – Результаты эксперимента с OUT файлами

Количество элементов в базе	**Время поиска в полных файлах**	**Время поиска в разделенных файлах**	**Ускорение**
404	0:00:00.010998	0:00:00.000998	В 11 раз
4004	0:00:00.091005	0:00:00.002002	В 45 раз

5.6 Применение SMILES для визуализации химических компонентов

Отдельной задачей по обработке CIF файлов является получение SMILES кода из файла.

SMILES (Simplified Molecular Input Line Entry System) — это текстовый формат для представления химических структур в виде строки символов. Он предоставляет удобный способ записи и обмена информацией о молекулах.

SMILES код состоит из последовательности символов, представляющих атомы, связи и другие химические свойства молекулы. Он основан на принципе, что атомы и связи могут быть представлены с использованием символов, основанных на их химических символах и валентностях.

Был проведен поиск имеющихся инструментов и выделено два наиболее перспективных, это библиотека RDkit [38] и программный продукт OpenBabel [39].

RDKit (The RDKit: Open-Source Cheminformatics) - это мощная библиотека для химического информатики, разработанная и поддерживаемая Open Source-сообществом. Она предоставляет широкий набор инструментов и алгоритмов для анализа химических структур, виртуального скрининга, моделирования молекул и других задач в области химической информатики. [38]

Взаимодействие с этой библиотекой наладить не удалось по причинам, связанным с поддержкой современных инструментов загрузки: pip, anaconda, setup-tool.

Open Babel - это программная библиотека и набор инструментов для химической информатики с открытым исходным кодом. Он предоставляет возможности для чтения, записи, конвертации и манипулирования химическими структурами и данными. Open Babel поддерживает большое количество форматов файлов и обеспечивает интерфейсы для различных языков программирования, включая C++, Python, Java и другие. Его поддержка прекратилась несколько лет назад, но в скомпилированном виде он решает все поставленные задачи, а именно генерация SMILES кодов и генерация изображений на основе этих кодов.

Результат этапа

В парсер CIF файлов добавилась процедура генерации SMILES кодов, также эти данные были внесены в файл structure.json и стали передаваться с другими параметрами через API.

5.7 Подготовка docker-образов для портирования на сервер

Для перехода информационной системы на этап тестирования, необходимо ее разместить на серверах с публичным адресом, в данном случае это один серверов Южно-Уральского Государственного университета.

Чтобы упростить поддержку было принято решение в создании docker образов для каждой из частей приложения. Также для удобства использования необходимо было написать docker-compose файл.

Docker - это открытая платформа для автоматизации развертывания, упаковки и запуска приложений в контейнерах. Контейнеры представляют собой изолированные среды, в которых могут работать приложения и их зависимости, обеспечивая консистентность и переносимость приложений между различными средами выполнения.

Docker образ - это состояние, или снимок, контейнера, который содержит все необходимые компоненты для запуска приложения в изолированном окружении. Образ является основной строительной единицей в Docker и служит основой для создания и запуска контейнеров [40].

Образ Docker содержит файловую систему, которая включает в себя все файлы и зависимости, необходимые для работы приложения. Он также включает в себя метаданные, такие как конфигурационные параметры, сетевые настройки, команды запуска и другую информацию, которая определяет, как контейнер будет работать.

Образы Docker являются неизменяемыми и имеют версионирование, что означает, что после их создания они не могут быть изменены, а только пересозданы с новой версией. Образы могут быть созданы с использованием Dockerfile, который содержит инструкции по построению образа, такие как установка зависимостей, копирование файлов, настройка среды выполнения и т.д. Docker контейнер - это запущенный экземпляр Docker образа. Контейнер представляет собой изолированное окружение, в котором выполняется приложение и его зависимости.

Контейнеры Docker используются для упаковки и доставки приложений с их необходимыми компонентами (библиотеки, зависимости, конфигурационные файлы и т.д.) вместе с самим приложением. Каждый контейнер работает независимо от других контейнеров и хост-системы, обеспечивая изолированное окружение выполнения.

Docker Compose - это инструмент, который позволяет определять и запускать множество связанных контейнеров Docker вместе как единое приложение. Он использует файл конфигурации в формате YAML для определения сервисов, сетей, томов и других аспектов приложения [41].

Docker Compose упрощает оркестрацию и управление контейнеризованными приложениями, особенно в случаях, когда необходимо запускать несколько контейнеров и устанавливать между ними связи и зависимости. Вместо запуска и настройки каждого контейнера вручную, Docker Compose позволяет определить все необходимые компоненты в одном файле и запустить их с помощью одной команды. Результат

Результат

По окончанию данного этапа были настроены два образа, это образ базы данных postresql и образ обработки запроса nginx [42]. Созданы два своих образа, в одном из которых располагается «backend» приложение разрабатываемое в рамках данной работы (Листинг 10) и «frontend» приложение разрабатываемое в рамках другой работы, но тоже данного проекта (Листинг 11). Конфигурация nginx сервера представлена в листинге 12. Описание docker образа записано в «Dockerfile» внутри каждого из приложений.

Листинг 10 – Параметры образа backend контейнера

```
FROM python:3.11
# Установка зависимостей Python
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
# Копирование исходного кода приложения
COPY . /app
WORKDIR /app
# Установка переменной окружения для Flask
ENV FLASK_APP=app.py
ENV DB_host=postgresql
ENV DB_port=5432
ENV DB_user=postgres
ENV DB_password=password
ENV DB_dbname=postgres
RUN pip3 install psycopg2-binary
# Установка команды запуска приложения
CMD ["flask", "run", "--host=0.0.0.0"]
```

Листинг 11 – Параметры образа frontend приложения

```
FROM node
WORKDIR /client
COPY ./package*.json ./
RUN npm install
RUN npm install -g http-server
COPY . .
RUN npm run build
EXPOSE 5173
CMD ["npm", "run", "dev", "--", "--host"]
```

Листинг 12 – Параметры настройки сервера nginx

```
server {
    listen 80;
       client_max_body_size 100M;
    location / {
        proxy_pass http://parser_front:5173/;
    }

    location /api {
        proxy_pass http://web_cif_parser:5005/api;
    }
}
```

Также для взаимодействия всех 4 модулей был написан docker-compose файл (Листинг 13).

Листинг 13 – Параметры docker-compose.yaml

```
version: '2.1'
services:
  postgresql:
    restart: always
    image: postgres
    environment:
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=password
      - POSTGRES_DB=postgres
    volumes:
      - /data:/var/lib/postgresql/data
    ports:
      - 5433:5432
  web_cif_parser:
    restart: always
    image: ******/backend:latest
```

```yaml
    environment:
      - DB_host=postgresql
      - DB_port=5432
      - DB_user=postgres
      - DB_password=password
      - DB_dbname=postgres
    ports:
      - 5005:5005
    depends_on:
      - postgresql
    entrypoint: ["python3", "app.py"]
  parser_front:
    restart: always
    image:  ******/frontend:latest
    environment:
      - VITE_APP_API_URL=http://*****.susu.ru:5005/api
    ports:
      - 5173:5173
    depends_on:
      - web_cif_parser
  nginx:
    restart: always
    image: :*****/my_nginx
    ports:
      - 80:80
    depends_on:
      - web_cif_parser
```

ЗАКЛЮЧЕНИЕ

Цель данной работы состояла в разработке информационной системы, которая должна позволять удобно и структурировано хранить наиболее полезные данные, осуществлять получение этих полезных данных из исходных файлов и осуществлять поисковые процедуры среди полученной информации.

В рамках данной работы были решены следующие задачи:

– подбор архитектуры системы и способов ее реализации;

– разработка парсера для определенных заказчиком файлов;

– разработка серверной части: создание API для взаимодействия с другим программным обеспечением, функции загрузки файлов и хранения исходных. Защита от дублирования данных;

– проектирование базы данных - определение структуры базы данных;

– модуль взаимодействия с базой данных;

– проведение экспериментов по сравнению результатов при работе с обработанными и необработанными данными;

Цель достигнута, поставленные задачи были решены в необходимой мере, заказчик получил продукт и претензий не имеет.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Киселева, Н. Н. Информационная структура современного неорганического материаловедения / Н. Н. Киселева // XVII Всероссийская с международным участием школа - семинар по структурной макрокинетике для молодых ученых имени академика А.Г. Мержанова : Сборник научных материалов, Черноголовка, 16–18 октября 2019 года. – Черноголовка: Федеральное государственное бюджетное учреждение науки Институт структурной макрокинетики и проблем материаловедения им. А.Г. Мержанова Российской академии наук, 2019. – С. 13-16. – DOI 10.24411/9999-004A-2019-10001. – EDN GQBQDA.

2 Киселева, Н. Н. Информационная инфраструктура современного материаловедения - проекты и результаты / Н. Н. Киселева // Энергия: экономика, техника, экология. – 2017. – № 7. – С. 2-14. – EDN YZMWXN.

3 Materials Project – Home [Электронный ресурс] URL: https://materialsproject.org/ (дата обращения: 09.03.2023)

4 Materials Project Q&A with Persson // News Center [Электронный ресурс]. - Дата публикации: 8 мая 2020 г. - URL: https://newscenter.lbl.gov/2020/05/08 /materials-project-qa-persson/ (дата обращения: 09.03.2023)

5 NOMAD Lab [Электронный ресурс]. - URL: https://nomad-lab.eu/nomad-lab/ (дата обращения: 11.03.2023)

6 National Institute of Standards and Technology (NIST) [Электронный ресурс]. - URL: https://www.nist.gov/ (дата обращения: 11.03.2023)

7 Inorganic Crystal Structure Database (ICSD) [Электронный ресурс]. - URL: https://www.lib.tpu.ru/html/icsd (дата обращения: 20.03.2023)

8 Organic Materials Database (OMDb) [Электронный ресурс]. - URL: https://omdb.mathub.io/ (дата обращения: 20.03.2023)

9 Research Collaboratory for Structural Bioinformatics (RCSB) Protein Data Bank [Электронный ресурс]. - URL: https://www.rcsb.org/ (дата обращения: 22.03.2023)

10 Bernstein, H. J., & Westbrook, J. D. (2001). The Protein Data Bank: a computer-based archival file for macromolecular structures. Methods in Enzymology, 331, 236-248.

11 Cambridge Crystallographic Data Centre (CCDC). Cambridge Structural Database (CSD) [Электронный ресурс]. - URL: https://www.ccdc.cam.ac.uk /solutions/software/csd/ (дата обращения: 29.03.2023)

12 Allen, F. H., et al. The Cambridge Structural Database: a quarter of a million crystal structures and rising. Acta Crystallographica Section B: Structural Science, 2002. - С. 380-388.

13 Hall, S. R., Allen, F. H., & Brown, I. D. The Crystallographic Information File (CIF): a new standard archive file for crystallography. Acta Crystallographica Section A: Foundations of Crystallography,1991. -С. 655-685.

14 The CIF (Crystallographic Information File) - A Brief Guide [Электронный ресурс]. - URL: https://www.iucr.org/__data/assets/pdf_file/0019 /22618/cifguide.pdf (дата обращения: 05.04.2023)

15 Crystallography Group, University of Turin. CRYSTAL [Электронный ресурс]. - URL: https://www.crystal.unito.it/ (дата обращения: 08.04.2023)

16 PDB Westbrook, J., & Fitzgerald, P. M. D. (2009). The PDB format, mmCIF, and other data formats. Methods in Molecular Biology, 572, 243-262.

17 MOL Gasteiger, J., & Engel, T. (Eds.). (2003). Chemoinformatics: A Textbook. Wiley-VCH.

18 XYZ format Saxe, J. D., & Wilkins, D. K. (1997). MOLPRO: A simple tool for drawing chemical structures. Journal of Chemical Information and Modeling, 37(6), 1019-1023.

19 Python Reitz, K., & Schlusser, T. (2016). The Hitchhiker's Guide to Python: Best Practices for Development. O'Reilly Media.

20 Ruby Pine, C. (2013). Learn to Program (2nd Edition). Pragmatic Bookshelf.

21 Java Хорстманн К. Полный справочник Java. — 11-е издание. — ДМК Пресс, 2021.

22 C# Шилдт Г. Искусство программирования на языке C#. — Питер, 2018.

23 The Community for Open Innovation and Collaboration | The Eclipse Foundation [Электронный ресурс]. - URL: https://www.eclipse.org/ (дата обращения: 20.04.2023)

24 Sublime Text - Text Editing, Done Right [Электронный ресурс]. - URL: https://www.sublimetext.com/ (дата обращения: 20.04.2023)

25 PyCharm: the Python IDE for Professional Developers by JetBrains [Электронный ресурс]. - URL: https://www.jetbrains.com/pycharm/ (дата обращения: 20.04.2023)

26 Белл, Ш., Купер, М., Майтнер, И. Learning MySQL: Handle Your Data with the World's Most Popular Open Source Database. — O'Reilly Media, 2006. — 602 с.

27 PostgreSQL: Documentation: 15: PostgreSQL 15.3 Documentation [Электронный ресурс]. - URL: https://www.postgresql.org/docs/15/index.html (дата обращения: 20.04.2023)

28 Система управления БД MS Office Access. – Информатика [Электронный ресурс]. - URL: http://psk68.ru/files/metod/uchebnik_Informatika /access.html (дата обращения: 20.04.2023)

29 MariaDB Foundation - MariaDB.org [Электронный ресурс]. - URL: https://mariadb.org/ (дата обращения: 20.04.2023)

30 SQLite Home Page [Электронный ресурс]. - URL: https://www.sqlite.org /index.html (дата обращения: 20.04.2023)

31 pyemaps PyPI [Электронный ресурс]. - URL: https://pypi.org/project/ pyemaps/#description (дата обращения: 29.04.2023)

32 gayverjr/pyemap: Python package aimed at automatic identification of electron and hole transfer pathways in proteins. [Электронный ресурс]. - URL: https://github.com/gayverjr/pyemap (дата обращения: 29.04.2023)

33 How to write a parser — NOMAD Repository and Archive documentation [Электронный ресурс]. - URL: https://nomad-lab.eu/prod/rae/docs /parser.html (дата обращения: 29.04.2023)

34 Getting Started — ChemSpiPy 2.0.0 documentation [Электронный ресурс]. - URL: https://chemspipy.readthedocs.io/en/latest/guide/gettingstarted.html (дата обращения: 29.04.2023)

35 Protein Data Bank Japan / tools / cif-parsers · GitLab documentation [Электронный ресурс]. - URL: https://gitlab.com/pdbjapan/tools/cif-parsers (дата обращения: 01.05.2023)

36 (IUCr) PyCifRW [Электронный ресурс]. - URL: https://www.iucr.org /resources/cif/software/pycifrw (дата обращения: 01.05.2023)

37 Introduction — pymatgen 2023.5.10 documentation [Электронный ресурс]. - URL: https://pymatgen.org/ (дата обращения: 03.05.2023)

38 RDkit [Электронный ресурс]. - URL: https://www.rdkit.org/ (дата обращения: 20.05.2023)

39 Open Babel [Электронный ресурс]. - URL: https://openbabel.org /wiki/Main_Page (дата обращения: 20.05.2023)

40 Docker overview | Docker Documentation [Электронный ресурс]. - URL: https://docs.docker.com/get-started/overview/ (дата обращения: 23.05.2023)

41 Docker Compose overview | Docker Documentation [Электронный ресурс]. - URL: https://docs.docker.com/compose/ (дата обращения: 23.05.2023)

42 Nginx [Электронный ресурс]. - URL: https://nginx.org/ (дата обращения: 23.05.2023)