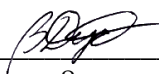


**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Научный руководитель,
приглашенный преподаватель
департамента программной инженерии


_____ В.А. Дударев
« 8 » мая 2022 г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия», кандидат
технических наук


_____ В.В. Шилов
« 8 » мая 2022 г.

**RESTFUL ВЕБ-СЕРВИС ДЛЯ РАБОТЫ С БАЗАМИ ДАННЫХ ПО СВОЙСТВАМ
НЕОРГАНИЧЕСКИХ ВЕЩЕСТВ И МАТЕРИАЛОВ**


Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.05.15-01 81 01-1-ЛУ

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	RU.17701729.05.15-01 81 01-1

Исполнитель:
студент группы БПИ205


_____ /Овчинникова П.А./
« 8 » мая 2022 г.

УТВЕРЖДЕНО
RU.17701729.05.15-01 81 01-1-ЛУ

**RESTFUL ВЕБ-СЕРВИС ДЛЯ РАБОТЫ С БАЗАМИ ДАННЫХ ПО СВОЙСТВАМ
НЕОРГАНИЧЕСКИХ ВЕЩЕСТВ И МАТЕРИАЛОВ**

Пояснительная записка

RU.17701729.05.15-01 81 01-1

Листов 45

<i>Подп. и дата</i>	
<i>Инв. № дубл.</i>	
<i>Взам. инв. №</i>	
<i>Подп. и дата</i>	
<i>Инв. № подл</i>	RU.17701729.05.15-01 81 01-1

2022

СОДЕРЖАНИЕ

АННОТАЦИЯ	5
ВВЕДЕНИЕ	7
1.1. НАИМЕНОВАНИЕ ПРОГРАММЫ.....	7
1.2. ДОКУМЕНТ, НА ОСНОВАНИИ КОТОРОГО ВЕДЕТСЯ РАЗРАБОТКА	7
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	8
2.1. НАЗНАЧЕНИЕ ПРОГРАММЫ	8
2.1.1. Функциональное назначение	8
2.1.2. Эксплуатационное назначение	8
2.2. КРАТКАЯ ХАРАКТЕРИСТИКА ОБЛАСТИ ПРИМЕНЕНИЯ	8
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	9
3.1. ПОСТАНОВКА ЗАДАЧИ НА РАЗРАБОТКУ ПРОГРАММЫ.....	9
3.2. ОПИСАНИЕ И ОБОСНОВАНИЕ АРХИТЕКТУРЫ ПРОГРАММЫ.....	11
3.2.1. Описание архитектуры программы.....	11
3.2.2. Обоснование архитектуры программы	13
3.3. ОПИСАНИЕ И ОБОСНОВАНИЕ АЛГОРИТМА РАБОТЫ ПРОГРАММЫ	14
3.3.1. Описание алгоритма работы программы	14
3.3.2. Обоснование алгоритма работы программы.....	19
3.4. ОПИСАНИЕ РАБОТЫ С БАЗОЙ ДАННЫХ.....	20
3.5. ОПИСАНИЕ И ОБОСНОВАНИЕ ВЫБОРА МЕТОДА ОРГАНИЗАЦИИ ВХОДНЫХ И	
 ВЫХОДНЫХ ДАННЫХ	22
3.6. ОПИСАНИЕ И ОБОСНОВАНИЕ ВЫБОРА СОСТАВА ТЕХНИЧЕСКИХ И ПРОГРАММНЫХ	
 СРЕДСТВ.....	25
3.6.1. Состав технических и программных средств	25
3.6.2. Обоснование выбора технических и программных средств	25
4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	26
4.1. ОРИЕНТИРОВОЧНАЯ ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ	26
4.2. ПРЕДПОЛАГАЕМАЯ ПОТРЕБНОСТЬ	26
4.3. ЭКОНОМИЧЕСКИЕ ПРЕИМУЩЕСТВА РАЗРАБОТКИ ПО СРАВНЕНИЮ С	
 ОТЕЧЕСТВЕННЫМИ И ЗАРУБЕЖНЫМИ ОБРАЗЦАМИ ИЛИ АНАЛОГАМИ	26
5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ	28
ПРИЛОЖЕНИЕ 1	30
ПРИЛОЖЕНИЕ 2	32
ПРИЛОЖЕНИЕ 3	33
ПРИЛОЖЕНИЕ 4	35
ПРИЛОЖЕНИЕ 5	40
ПРИЛОЖЕНИЕ 6	42
ПРИЛОЖЕНИЕ 7	45

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....46

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

Данный программный документ представляет собой пояснительную записку к программному проекту «Restful веб-сервис для работы с базами данных по свойствам неорганических веществ и материалов», предназначенного для упрощения процесса отслеживания обновлений в базах данных по свойствам неорганических веществ и материалов и автоматизации сохранения обновлений в метабазе.

Раздел «Введение» включает в себя наименование программы и документ, на основании которого ведётся разработка, с указанием организации, утвердившей данный документ.

В разделе «Назначение и область применения» содержатся функциональное и эксплуатационное назначение программы и краткая характеристика области её применения.

В разделе «Технические характеристики» присутствуют следующие подразделы: постановка задачи на разработку программы, описание функционирования программы, описание и обоснование алгоритма работы программы, описание и обоснование выбора метода организации входных и выходных данных, описание работы с базой данных, описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.

Программный документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [1];
2. ГОСТ 19.102-77 Стадии разработки [2];
3. ГОСТ 19.103-77 Обозначения программ и программных документов [3];
4. ГОСТ 19.104-78 Основные надписи [4];
5. ГОСТ 19.105-78 Общие требования к программным документам [5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

6
RU.17701729.05.15-01 81 01-1

Перед прочтением данного документа следует ознакомиться с терминологией, приведенной в Приложении 1.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы – «Restful веб-сервис для работы с базами данных по свойствам неорганических веществ и материалов».

Наименование программы на английском языке – «Restful web service for a databases on inorganic substances and materials properties».

1.2. Документ, на основании которого ведется разработка

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем тема курсового проекта.

Документ утвержден организацией: Национальный исследовательский университет «Высшая школа экономики», Факультет компьютерных наук, образовательная программа «Программная инженерия».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Назначение программы

2.1.1. Функциональное назначение

Функциональное назначение программы – предоставление универсального API для безопасной работы с базами данных, хранящих сведения о химических сущностях и их свойствах, просмотр метайнформации, хранящейся в метабазе, и сохранение обновлений из баз данных в метабазе.

2.1.2. Эксплуатационное назначение

Данный веб-сервис предназначен для упрощения процесса отслеживания обновлений в базах данных по свойствам неорганических веществ и материалов и автоматизации сохранения обновлений в метабазе.

Сервис может быть встроен во внешнюю систему, по запросу которой сервис будет отправлять запросы к базам данных и, при наличии обновлений в них, сохранять новые метаданные в метабазе. Другим случаем использования является ручной управление, при котором пользователь, пользуясь инфраструктурой веб-сервиса, сам отправляет запросы на поиск обновлений и их сохранение.

2.2. Краткая характеристика области применения

«Restful веб-сервис для работы с базами данных по свойствам неорганических веществ и материалов» применим в системах, где используются несколько баз данных и есть необходимость в обобщении, хранящейся в них информации, с помощью метабазы. Основная цель разрабатываемой программы – автоматизация процессов, связанных с импортированием метаданных в метабазу, и упрощение процесса отслеживания обновлений в базах данных по свойствам неорганических веществ и материалов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку программы

Разрабатываемый программный продукт должен выполнять следующие задачи:

1. Разделение инфраструктуры веб-сервиса на два микросервиса: сервиса для базы данных и сервиса для метабазы;
2. Реализация функционала сервиса для базы данных:
 - 2.1. Возможность проверить текущее состояние работы сервиса. Проверяется:
 - i. Состояние подключения к базе данных;
 - ii. Проверка исполнения sql-скрипта для получения представления о метаданных таблицы (sql-скрипт указывается при установке сервиса для базы данных, более подробно об этом в пункте 3.4).
 - 2.2. Получение метаданных о свойствах, хранящихся в базе данных. В состав метаданных о свойствах входит следующая информация:
 - i. Идентификатор базы данных, откуда было взято данное свойство;
 - ii. Идентификатор свойства в базе данных;
 - iii. Название свойства;
 - iv. Описание свойства;
 - v. Ссылка на шаблон страницы о свойстве во Всемирной паутине;
 - vi. Статус обновления свойства относительно других таблиц.
 - 2.3. Получение метаданных о системах, хранящихся в базе данных. В состав метаданных о свойствах входит следующая информация:
 - i. Идентификатор базы данных, откуда была взята данная система;
 - ii. Уникальный идентификатор системы в базе данных;
 - iii. Количество элементов в системе;
 - iv. Статус обновления системы относительно других таблиц;
 - v. Дата внесения в базу данных этой системы;
 - vi. Строковое представление элементов;
 - vii. Информация о системе;
 - viii. Описание системы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 2.4. Ограничение доступа к функционалу 2.2 и 2.3 посредством аутентификации по JWT-токену.
3. Реализация функционала сервиса для метабазы:
- 3.1. Ограничение доступа к функционалу сервиса посредством Cookie аутентификации. Задача по ограничению функционала включает в себя следующие подзадачи:
- 3.1.1. Получение формы с пользовательских данных для прохождения аутентификации. Пользовательские данные включают в себя следующую информацию:
- i. Имя пользователя;
 - ii. Пароль.
- 3.1.2. Проверка полученных данных и аутентификация пользователя в зависимости от результатов проверки;
- 3.1.3. Возможность выхода из текущего аккаунта.
- 3.2. Возможность просмотра домашней страницы сервиса, с отображением перечня доступных функций (пункты 3.3 – 3.6);
- 3.3. Возможность просмотра просмотр подключенных сервисов (в виде таблицы);
- 3.4. Возможность просмотра сохраненных в метабазе метаданных о свойствах (в виде таблицы);
- 3.5. Возможность просмотра сохраненных в метабазе метаданных о системах (в виде таблицы);
- 3.6. Возможность обновления метаинформации в метабазе через сбор метаданных у сервисов для баз данных и просмотр внесенных обновлений (в виде двух таблиц: по системам и по свойствам);
- 3.7. Генерация JWT-токена при обращении к сервисам для баз данных для прохождения аутентификации.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.2. Описание и обоснование архитектуры программы

3.2.1. Описание архитектуры программы

Веб-сервис использует принцип микросервисной архитектуры программного обеспечения [12]: архитектура веб-сервиса состоит из двух типов микросервисов, взаимодействующих между собой – сервис для метабазы и сервис для базы данных. В контексте использования веб-сервиса метабаза – это место сохранения обновления, а базы данных по свойствам неорганических веществ и материалов – это источники обновления. Поэтому с точки зрения архитектуры сервис для метабазы может быть только в единственном экземпляре, в то время как сервисов для баз данных может быть несколько. Каждый сервис устанавливается на сервер с соответствующей ему базой данных для обеспечения доступа к информации из баз данных.

Рассмотрим процесс взаимодействия микросервисов с точки зрения общей архитектуры на примере запроса на обновления метабазы (рис. 1).

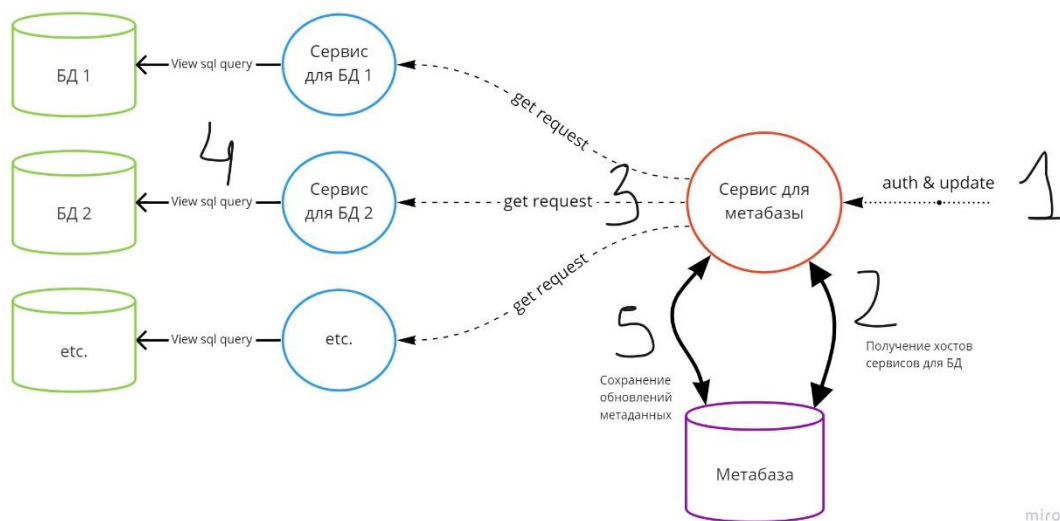


Рисунок 1. Общая архитектура работы веб-сервиса на примере запроса на обновление метабазы.

В общем виде алгоритм работы веб-сервиса следующий:

1. Пользователь проходит аутентификацию в сервисе для метабазы и отправляет запрос на обновление метабазы;
2. Сервис для метабазы запрашивает у метабазы хосты сервисов для баз данных;
3. Сервис для метабазы отправляет сервисам запросы на получение метаданных о свойствах и системах;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. Каждый сервис для базы данных принимает входящий запрос, обновляет представление о метаданных с помощью исполнения sql-скрипта, отправляет к своей базе данных запрос на получение метаданных и возвращает полученные метаданные сервису для метабазы;

5. Сервис для метабазы обнаруживает обновления и сохраняет их в метабазу.

Таким образом работа веб-сервиса разделяется на независимые компоненты, каждый из которых отвечает за выполнение своей задачи: сервисы для баз данных собирают информацию из баз данных, а сервис для метабазы анализирует полученную информацию, выделяет обновления и сохраняет их.

Архитектура веб-сервиса соответствует стилю REST [13]. В рамках клиент-серверного приложения это позволяет системам легко обмениваться данными и дает возможность масштабировать сервис в будущем. Рассмотрим работу веб-сервиса с точки зрения пользователя (рис. 2).

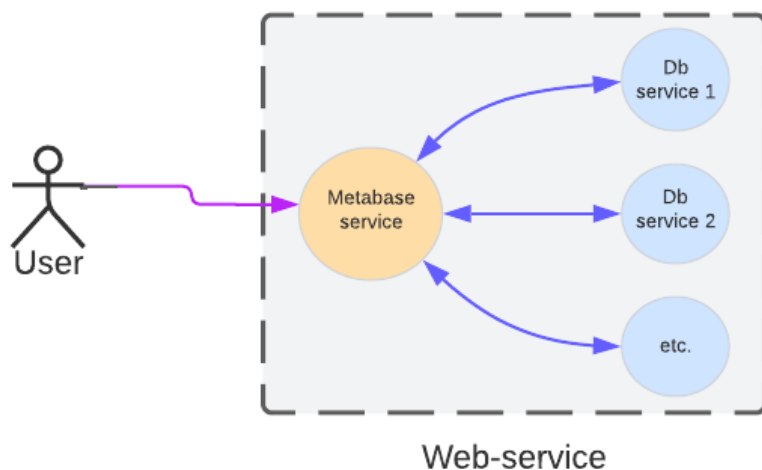


Рисунок 2. Веб-сервис с точки зрения пользователя.

При работе с веб-сервисом пользователь, по сути, взаимодействует только с сервисом для метабазы. Поскольку взаимодействие между компонентами уже настроено при разработке веб-сервиса, пользователю не нужно вдаваться в детали работы веб-сервиса, он может просто пользоваться его функционалом.

Сервис для метабазы разработан с использованием паттерна MVC (Model-View-Controller) [14]. Данный паттерн подразделяет данные программы и ее логику на три компонента с точки зрения функционального назначения. В паттерне присутствуют следующие компоненты:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. Модель (model) включает в себя сущности, используемые в программе и над которыми производятся некоторые действия. В результате совершения действий модель может менять свое состояние;
2. Представление (view) – данный компонент отвечает за отображение данных пользователю и за реагирование на намерения пользователя совершить какие-либо действие;
3. Контроллер (controller) обеспечивает выполнение действий пользователя, передавая модели некоторое новое состояние.

Схематичное изображение работы паттерна MVC изображено на рис. 3.

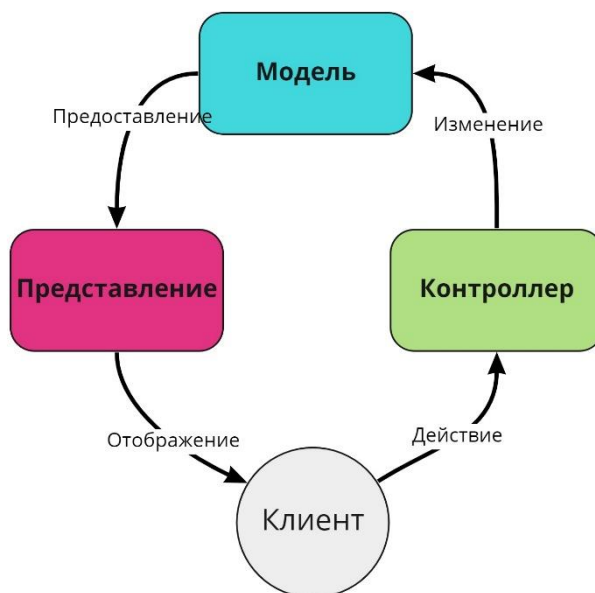


Рисунок 3. Схема работы паттерна MVC.

3.2.2. Обоснование архитектуры программы

«Restful веб-сервис для работы с базами данных по свойствам неорганических веществ и материалов» обеспечивает взаимодействие метабазы и баз данных, хранящих сведения о химических веществах и их свойствах. Поскольку метабаза и базы данных могут храниться на разных серверах, для обеспечения взаимодействия инфраструктуру веб-сервиса необходимо разделить на два микросервиса: сервис для метабазы и сервис для базы данных. Помимо этого, микросервисный подход к архитектуре закладывает возможности масштабирования веб-сервиса в будущем, а также снижает нагрузку на каждый отдельный сервер.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.3. Описание и обоснование алгоритма работы программы

3.3.1. Описание алгоритма работы программы

Архитектура веб-сервиса состоит из двух типов микросервисов, взаимодействующих между собой: сервиса для метабазы и сервиса для базы данных.

Алгоритм работы сервиса для метабазы может быть описан блок-схемами, представленными на рисунках 4-5.

На рисунке 4 представлен процесс аутентификации [15] пользователя в сервисе для метабазы.

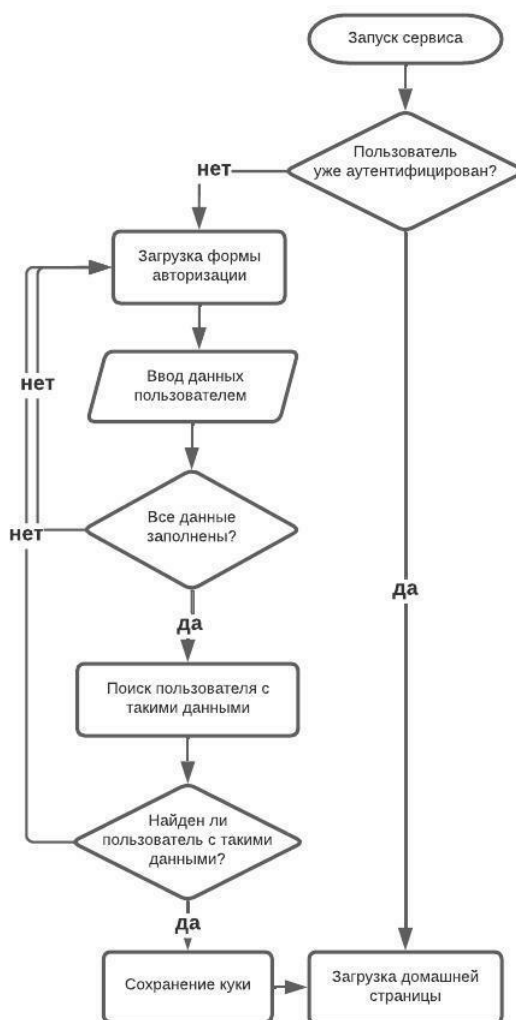


Рисунок 4. Блок-схема аутентификация в сервисе для метабазы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Процесс аутентификации может быть разбит на следующие этапы:

1. После запуска сервиса, в случае если пользователь еще не аутентифицирован, загружается представление с формой для аутентификации.
2. Пользователь должен указать в форме логин (имя пользователя) и пароль.
3. После отправки формы проводится проверка заполненности полей формы: если одно из полей пусто, пользователя возвращают на страницу с формой для аутентификации с указанием незаполненных полей; если все данные указаны, проводится их проверка.
4. При проверке введенные данные сверяются с данными уже зарегистрированных пользователей (они десериализуются из файла, указанного при установке сервиса, более подробно об этом в пункте 3.5): если нашелся пользователь с такими данными, значит аутентификация пройдена успешно, куки о текущей сессии сохраняются, и пользователя перенаправляют на домашнюю страницу сервиса. В случае если нет зарегистрированного пользователя с данными, идентичными введенным, пользователь будет перенаправлен на страницу с формой для аутентификации с сообщением о неправильности введенных данных.

На рисунке 5 показан алгоритм работы функционала сервиса для метабазы.

На домашней странице пользователю доступно несколько функций сервиса:

- a) Просмотр активных на данный момент сервисов для баз данных;
- b) Просмотр метаданных, хранящихся в метабазе, о свойствах;
- c) Просмотр метаданных, хранящихся в метабазе, о системах;
- d) Обновление метабазы.

В зависимости от выбранной функции дальнейший алгоритм работы может различаться.

При выборе функции о просмотре метаданных из метабазы алгоритм работы следующий:

1. Запрос к метабазе на получение метаданных о свойствах/системах;
2. Возврат html-страницы с таблицей, отображающей метаданные.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

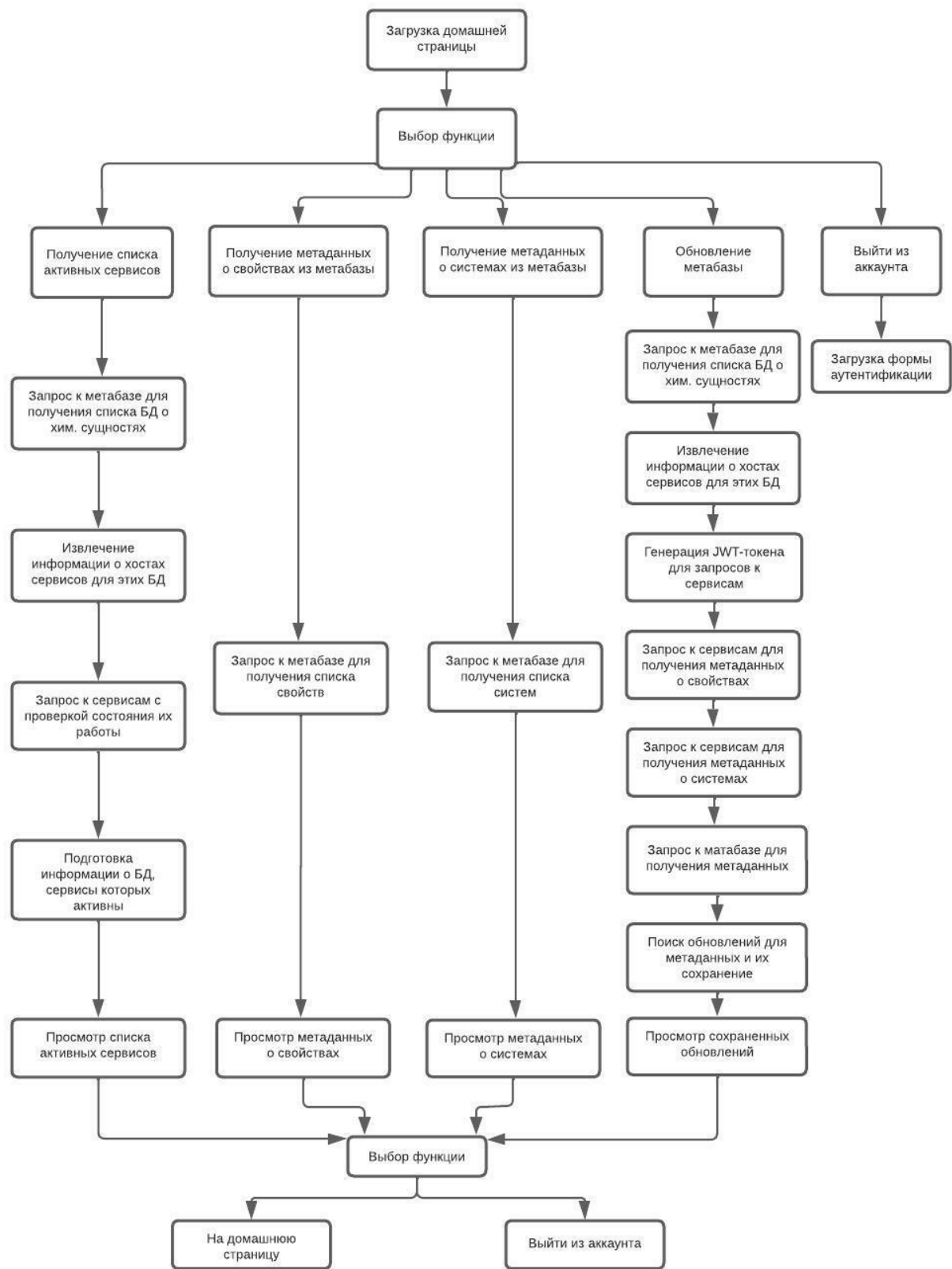


Рисунок 5. Блок-схема работы функционала сервиса для метабазы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

При выборе функции о просмотре подключенных на данный момент баз данных алгоритм работы следующий:

1. Запрос к метабазе на получение информации о базах данных по свойствам неорганических веществ и материалов;
2. Выделение из полученной информации данных о хостах сервисов для баз данных;
3. Запрос каждому сервису на проверку состояния его работы, анализ ответов на запросы и сохранение информации о сервисах, которые активны;
4. Подготовка информации о базах данных, сервисы которых активны;
5. Возврат html-страницы с таблицей, отображающей информацию о базах данных, сервисы которых активны.

При выборе функции обновления метабазы алгоритм работы следующий:

1. Запрос к метабазе на получение информации о базах данных по свойствам неорганических веществ и материалов;
2. Выделение из полученной информации данных о хостах сервисов для баз данных;
3. Генерация JWT-токена [16] для запросов к сервисам для баз данных. Токен необходим для совершения запросов, поскольку необходимо пройти аутентификацию и авторизацию в сервисах для получения доступа к некоторым эндпоинтам;
4. Каждому сервису для базы данных отправляются следующие запросы (с отправкой JWT-токена):
 - 4.1. Запрос на получение метаданных о свойствах;
 - 4.2. Запрос на получение метаданных о системах.
5. Запрос к метабазе на получение метаданных о свойствах и системах;
6. Сопоставление результатов запросов к сервисам для баз данных и к метабазе, обнаружений обновлений и их сохранение. Обновления ищутся сначала среди свойств, затем среди систем по следующему принципу:
 - 6.1. Поиск среди метаданных от сервисов для баз данных данных, которых нет в метабазе. Сохранение новых элементов в метабазу;
 - 6.2. Поиск отличающихся полей метаданных между метабазой и сервисов для баз данных. Сохранение отличий для элементов в метабазу.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

7. Возврат html-страницы с таблицами, отображающими информацию об обновлениях метаданных о свойствах и системах соответственно.

Алгоритм работы сервиса для базы данных может быть описан блок-схемой, изображенной на рисунке 6.

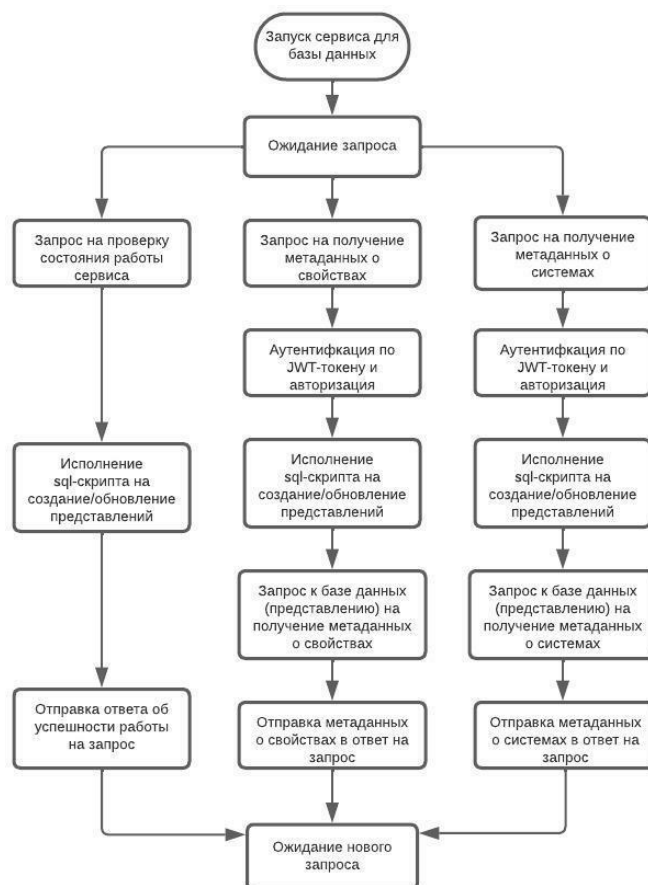


Рисунок 6. Блок-схема алгоритма работы сервиса для базы данных.

Сервис для базы данных принимает и обрабатывает следующие запросы:

- а) Запрос на проверку состояния работы сервиса;
- б) Запрос на получение метаданных о свойствах из базы данных;
- с) Запрос на получение метаданных о системах из базы данных.

В зависимости от запроса алгоритм работы сервиса может различаться.

При поступлении запроса на проверку состояния алгоритм работы следующий:

1. Исполнение sql-скрипта на создание/обновления представлений;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. Отправка ответа на запрос: если в ходе исполнения скрипта возникли ошибки, в ответе на запрос будет сообщено об ошибках в работе сервиса; если ошибок не возникло – сервис работает успешно.

При поступлении запроса на получение метаданных о свойствах/системах алгоритм работы следующий:

1. Аутентификация и авторизация исполнителя запроса по JWT-токену;
2. Исполнения sql-скрипта на создание/обновление представлений;
3. Запрос к представлению о свойствах/системах в базе данных и получение метаданных;
4. Отправка метаданных в ответ на запрос. В случае возникновения ошибок в ходе алгоритма в ответ на запрос будет возвращено значение null (пустое значение).

После исполнения запросов сервис ожидает поступления дальнейших запросов.

3.3.2. Обоснование алгоритма работы программы

Сервис для базы данных устанавливается для каждой базы данных отдельно. Такой подход с индивидуальной настройкой дает веб-сервису масштабируемость, так как взаимодействие между сервисами для метабазы и баз данных становится унифицированным, и в таблицу-источников обновлений можно вносить любое количество баз данных по свойствам неорганических веществ и материалов вне зависимости от внутреннего устройства базы данных.

Сервис для метабазы использует cookie аутентификацию [15, 16]. Такой вариант аутентификации удобен при разных способах использования веб-сервиса: как при ручном варианте, когда пользователю достаточно заполнить форму аутентификации, так и при автоматическом вызове сервиса, когда при генерируемом запросе достаточно указать query-параметры в URL запроса.

Сервис для базы данных использует JWT аутентификацию. Данный вид аутентификации хорош тем, что при нем не требуется хранение дополнительных данных о выданных сессиях, он удобен при выполнении запросов к сервису (достаточно указать токен в заголовках запроса) и этот вид аутентификации довольно прост в реализации.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.4. Описание работы с базой данных

В «Restful веб-сервис для работы с базами данных по свойствам неорганических веществ и материалов» используемые базы данных подразделяются на два типа: метабазу и базу данных по свойствам неорганических веществ и материалов. В веб-сервисе может быть только одна метабаза, куда будут сохраняться обновления. Баз данных по свойствам неорганических веществ и материалов может быть несколько, оттуда будут браться обновления [17, 18]. При разработке данного веб-сервиса в качестве метабазы использовалась база данных Metabase, в качестве баз данных по свойствам неорганических веществ и материалов использовались базы данных BandGap и Crystal. Опишем работу с базами данных на их примере.

База данных Metabase содержит 24 таблицы. Однако в работе веб-сервиса используются только 3 таблицы, связанных между собой по ключу DBID (рис. 7):

1. Таблица «DBInfo» — таблица с данными о базах данных по свойствам неорганических веществ и материалов;
2. Таблица «PropertiesInfo» — таблица с метаданными о свойствах неорганических веществ и материалов;
3. Таблица «SystemInfo» — таблица с метаданными о системах неорганических веществ и материалов.

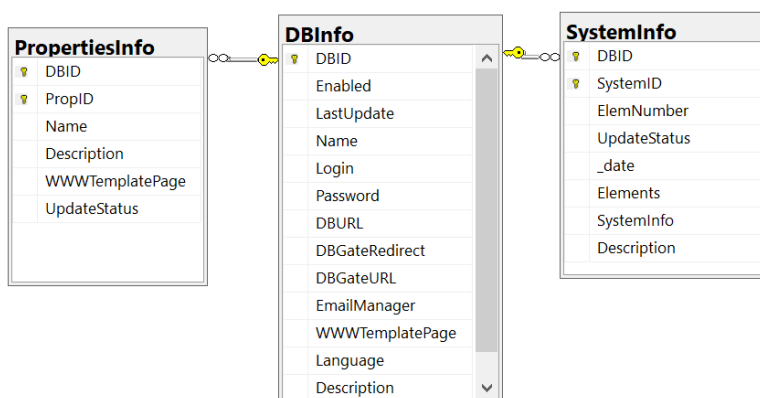


Рисунок 7. Используемые таблицы базы данных Metabase.

База данных BandGap содержит 11 таблиц. Для подключения базы данных к сервису использовались 2 таблицы (рис. 8):

1. Таблица «Substances» — таблица с данными о неорганических веществах;
2. Таблица «_PropertiesConv» — таблица с метаданными о свойствах неорганических веществ и материалов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

PropertiesConv	Substances
<div> <div>⚙</div> <div>NOMPROP</div> <div>UpdateStatus</div> <div>NAZVPROP</div> <div>HTML</div> </div>	<div> <div>⚙</div> <div>SubstanceID</div> <div>NumElements</div> <div>El1</div> <div>X1</div> <div>El2</div> <div>X2</div> <div>El3</div> <div>X3</div> <div>El4</div> <div>X4</div> <div>Elements</div> <div>Compound</div> <div>_date</div> </div>

Рисунок 8. Используемые таблицы базы данных BandGap.

База данных Crystal содержит 41 таблицу. Для подключения базы данных к сервису использовались 2 таблицы (рис. 9):

1. Таблица «HeadTabl» — таблица с данными о системах неорганических веществ и материалов;
2. Таблица «Properties» — таблица с метаданными о свойствах неорганических веществ и материалов.

HeadTabl	Properties
<div> <div>⚙</div> <div>HeadClue</div> <div>System</div> <div>Expert</div> <div>Help</div> <div>Class</div> </div>	<div> <div>⚙</div> <div>NOMPROP</div> <div>NAZVPROP</div> <div>HTML</div> <div>TableName</div> <div>Y</div> </div>

Рисунок 9. Используемые таблицы базы данных Crystal.

Для работы с базами данных используется СУБД SQL Server Management. Обращение к базе данных для подключения к БД, создания и обновления представлений, а также получения данных из таблиц происходит с помощью объектно-ориентированной технологии доступа к данным ADO.NET Entity Framework [12] и взаимодействия с SQL Server через сущности пространства имен Microsoft.Data.SqlClient.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.5. Описание и обоснование выбора метода организации входных и выходных данных

Входными данными являются:

а) Сервис для метабазы:

1. При установке сервиса пользователю необходимо указать строку подключения к метабазе MetabaseConnection в файле appsetting.json в разделе ConnectionStrings;
2. В процессе использования сервиса пользователю необходимо пройти аутентификацию. Для этого пользователю необходимо указать имя пользователя и пароль.

б) Сервис для базы данных:

1. При установке сервиса пользователю необходимо указать строку подключения к базе данных DatabaseConnection в файле appsetting.json в разделе ConnectionStrings;
2. При установке сервиса пользователю необходимо sql-скрипт DatabaseQuery.sql, результатам работы которого будет создание/обновление представлений SystemInfo и PropertiesInfo о метаданных систем и свойств в данной таблице.

Представление SystemInfo должно содержать следующие столбы:

- i. DBID: идентификатор базы данных, откуда была взята данная система; тип int, не может принимать значение null;
- ii. SystemId: уникальный идентификатор системы в базе данных; тип int, не может принимать значение null;
- iii. ElemNumber: количество элементов в системе; тип int (значение по умолчанию – 0), не может принимать значение null;
- iv. UpdateStatus: статус обновления системы относительно других таблиц, тип int (значение по умолчанию – 1), не может принимать значение null;
- v. _date; дата внесения в базу данных этой системы; тип datetime (значение по умолчанию – дата создания/обновления представления), не может принимать значение null;
- vi. Elements: строковое представление элементов, представленных в этой системе; тип varchar(32), не может принимать значение null;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- vii. SystemInfo: информация о системе; тип varchar(256), не может принимать значение null;
- viii. Description: описание системы, тип varchar(256) (значение по умолчанию – “”) (пустая строка)), может принимать значение null.

Представление PropertiesInfo должно содержать следующие столбцы:

- i. DBID: идентификатор базы данных, откуда было взято данное свойство; тип int, не может принимать значение null;
- ii. PropId: уникальный идентификатор свойства в базе данных; тип int, не может принимать значение null;
- iii. Name: название свойства; тип varchar(256), не может принимать значение null;
- iv. Description: описание свойства; тип text (значение по умолчанию – “”) (пустая строка)), не может принимать значение null;
- v. WWWTemplatePage: ссылка на шаблон страницы о свойстве во Всемирной паутине; тип varchar(256) (значение по умолчанию – “”) (пустая строка)), не может принимать значение null;
- vi. UpdateStatus: статус обновления свойства относительно других таблиц; тип int (значение по умолчанию – 1), не может принимать значение null.

Такой подход к алгоритму установки сервиса был выбран с целью обеспечения масштабируемости веб-сервиса. Масштабируемость обеспечивается за счет того, что есть возможность подключить любую базу данных по свойствам неорганических веществ и материалов, достаточно лишь написать sql-скрипт, описывающий создание/обновление представление SystemInfo и PropertiesInfo.

Выходными данными являются:

а) Сервис для метабазы:

1. В ответ на запрос о получении формы для аутентификации сервис должен предоставить форму аутентификации с полями для заполнения имени пользователя и пароля и кнопкой, при нажатии на которую форма будет отправлена;
2. В ответ на запрос об аутентификации пользователя (проверки введенных данных) есть два варианта выходных данных:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- i. При корректности введенных данных и успешной аутентификации пользователь будет перенаправлен на домашнюю страницу сервиса, где в виде списка будет отображен функционал сервиса для метабазы;
 - ii. При некорректности введенных данных пользователю будет отправлена форма для аутентификации с отображением ошибок аутентификации (сообщение о незаполненности одного из полей или сообщение о некорректности данных).
3. В ответ на запрос на получение списка активных сервисов пользователю должна быть отправлена страница с таблицей, содержащей информацию о хосте подключенного сервиса и информацией о базе данных, к которой подключен активный сервиса;
 4. В ответ на запрос о получении метаданных о свойствах из метабазы пользователю должна быть отправлена страница с таблицей, содержащей информацию о метаданных свойств;
 5. В ответ на запрос о получении метаданных о системах из метабазы пользователю должна быть отправлена страница с таблицей, содержащей информацию о метаданных систем;
 6. В ответ на запрос об обновлении метабазы пользователю должна быть возвращена страница со следующей информацией:
 - i. В случае если ни один из сервисов для базы данных не активен, на странице должны быть сообщения о том, что не удалось собрать обновления из баз данных, связанных с этими сервисами;
 - ii. В случае если хотя бы один из сервисов для базы данных был доступен, на странице должны быть отображены две таблицы: таблица, содержащая обновления метаданных о свойствах; таблица, содержащая обновления метаданных о системах. Также на странице должны быть отображены сообщения о неудачах сбора обновлений из баз данных, связанных с неактивными сервисами, если таковые имеются.
 7. В ответ на запрос о выходе из системы пользователю должна быть отправлена форма для аутентификации.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.6. Описание и обоснование выбора состава технических и программных средств

3.6.1. Состав технических и программных средств

Для нормального функционирования программы требуется компьютер, оснащенный следующими техническими компонентами:

1. VGA-совместимые видеоадаптер и монитор;
2. 200 мб свободного дискового пространства;
3. 512 мб ОЗУ или более;
4. Периферийные устройства: клавиатура, мышь или тачпад;
5. Доступ к сети интернет.

Для работы программы необходим следующий состав программных средств:

1. ОС Windows 10;
2. Microsoft .NET 5.0;
3. SQL Server 2019 Express Edition и выше;
4. Веб-браузер с поддержкой HTML5 и CSS3 [21, 22].

3.6.2. Обоснование выбора технических и программных средств

При количестве памяти ОЗУ, не соответствующего требованиям, ресурсов компьютера может не быть недостаточно для корректной работы программы, что повлечет аварийное завершение работы. Для сохранения промежуточных данных требуется не менее 200 Мб свободного места на жестком диске. Мышь и клавиатура необходимы для обеспечения взаимодействия пользователя и веб-сервиса.

Для разработки программного продукта был сделан выбор в пользу языка программирования C# [19, 20] с использованием фреймворка ASP.NET Core, так как это является удобным решением с точки зрения кроссплатформенной веб-разработки.

SQL Server требуется для обеспечения и поддержки работы веб-сервиса и баз данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Ориентировочная экономическая эффективность

В рамках данной работы расчет экономической эффективности не предусмотрен.

4.2. Предполагаемая потребность

«Restful веб-сервис для работы с базами данных по свойствам неорганических веществ и материалов» может быть использован для автоматизации процесса сохранения метаданных в метабазу. Таким образом, он поможет отслеживать появление новых данных при работе с несколькими базами данных.

4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Были выявлены следующие аналоги программного продукта «Restful веб-сервис для работы с базами данных по свойствам неорганических веществ и материалов»:

1) Amazon SimpleDB [24]:

Amazon SimpleDB – это хранилище данных NoSQL, которое полностью обеспечивает администрирование баз данных. Продукт упрощает процесс работы с базами данных, поскольку разработчику достаточно сделать запросы веб-сервисов и сохранить полученные, а Amazon SimpleDB сделает все остальное.

Данное решение обеспечивает высокую доступность к данным, гибкость к формату хранения данных, а также данная система является легко масштабируемой. Гибкость обеспечивается тем, что Amazon Simple DB – это нереляционная база данных. Однако в таком случае найти необходимую информацию внутри базы сложнее. Еще одним минусом использования этого хранилища данных может являться стоимость его использования.

2) Native Oracle XML DB Web Services [25]:

Сервис предоставляет возможность создания запросов SQL к базе данных Oracle и получения результатов в виде данных XML. Минусы данного веб-сервиса: сервис работает только с СУБД Oracle, а также сервис не использует REST подход к архитектуре (как нужно в нашем случае).

3) Oracle REST Data Service [26]:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Oracle REST Data Service — это приложение, которое позволяет создавать RESTful API для баз данных Oracle, используя SQL. По сути, ORDS представляет собой HTTP-интерфейс между внешним миром и БД Oracle. Аналогично предыдущему сервису, Oracle REST Data Service совместим только с СУБД от Oracle.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.
11. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
12. Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга. – СПб.: Питер, 2019. – 544 с.: ил.
13. Masse M. REST API Design Rulebook. – USA.: O'Reilly Media, Inc., 2011. – 114 с.: ил.
14. Lakshmiraghavan B. Pro ASP.NET Web API Security. Securing ASP.NET Web API. – USA.: Apress, 2013. – 416 с.: ил.
15. Madden N. API Security in Action. – USA.: Manning, 2020. – 577 с.: ил.
16. Фримен А. ASP.NET Core MVC 2 с примерами на C# для профессионалов. 7-е изд.: Пер. с англ. - СПб.: ООО "Диалектика", 2019. - 1008 с.: ил. – Парал. тит. англ.
17. Дейт, К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильямс", 2005. — 1328 с.: ил. — Парал. тит. англ.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

18. Фиайли К. SQL: Пер. с англ. – М.: ДМК Пресс. – 456 с.: ил. (Серия «Quick Start»).
19. Джепикс Ф., Троелсен Э. Язык программирования C# 7 и платформы .NET и .NET Core. Пер. с англ. – Артеменко Ю. Н., 2018. – 1328 с.
20. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#. 3-е изд. - СПб.: Питер, 2012.- 928 с.
21. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. — СПб.: Питер, 2012. — 304 с.: ил.
22. Макфарланд Д. Большая книга CSS3. 3-е изд. — СПб.: Питер, 2014. — 608 с.: ил. — (Серия «Бестселлеры O'Reilly»).
23. Документация по Entity Framework [Электронный ресурс] / Microsoft. – Режим доступа: <https://docs.microsoft.com/ru-ru/ef/>, свободный. (Дата обращения: 13.02.2022).
24. Продукт Amazon SimpleDB [Электронный ресурс] / Amazon AWS. – Режим доступа: <https://aws.amazon.com/ru/simpliedb/>, свободный. (дата обращения: 25.12.2021)
25. Продукт Native Oracle XML DB Web Services [Электронный ресурс] / Oracle docs. – Режим доступа: https://docs.oracle.com/database/121/ADXDB/xdb_web_services.htm#ADXDB5676, свободный. (дата обращения: 25.12.2021)
26. Продукт Oracle REST Data Services [Электронный ресурс] / Oracle. – Режим доступа: <https://www.oracle.com/ru/database/technologies/appdev/rest.html>, свободный. (дата обращения: 25.12.2021)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 1

ТЕРМИНОЛОГИЯ

Таблица 1

Термин	Определение
Веб-сервис	Идентифицируемая уникальным веб-адресом программная система со стандартизированными интерфейсами.
API	Описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой. API может представлять собой набор классов, процедур, функций, структур или констант. Расшифровка: Application Programming Interface.
HTTP	Прикладной протокол для передачи гипертекстовых документов, таких как HTML. Расшифровка: Hypertext Transfer Protocol.
REST	Стиль архитектуры программного обеспечения для распределенных систем, как правило, используется для построения веб-служб. Расшифровка: Representational state transfer.
Микросервисная архитектура	Подход, при котором единое приложение строится как набор небольших сервисов, каждый из которых работает независимо от других и коммуницирует с остальными используя протоколы, такие как HTTP.
Метаданные (метаинформация)	Информация о другой информации, или данные, относящиеся к дополнительной информации о содержимом или объекте.
База данных	Упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе.
Метабаза	База данных, хранящая информацию о метаданных.
Аутентификация	Процесс проверки подлинности учетных данных пользователя для его идентификации.
Авторизация	Процесс проверки прав пользователя и определения возможности доступа к различным ресурсам.
Cookie (куки)	Небольшой фрагмент данных, отправленный веб-сервером и хранимый на компьютере пользователя.
Cookie-based аутентификация	Вид аутентификации, при котором для идентификации пользователя используются файлы cookie.
Токен аутентификации	Нечто, свидетельствующее об аутентичности и используемое для идентификации его владельца.
JSON	Текстовый формат обмена данными, основанный на JavaScript. Расшифровка: JavaScript Object Notation.
JWT-токен	Открытый стандарт для создания токенов доступа, основанный на формате JSON. Расшифровка: JSON Web Token.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 1

JWT аутентификация	Вид аутентификации, при котором для идентификации пользователя используются JWT-токены.
ADO.NET Entity Framework	Объектно-ориентированная технология доступа к данным.
Microsoft SQL Server	Система управления реляционными базами данных, разработанная корпорацией Microsoft.
Нереляционная база данных	База данных, в которой не используется табличная схема строк и столбцов.
NoSql	Обозначение широкого класса разнородных систем управления базами данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 2

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КОМПОНЕНТОВ
ИНФРАСТРУКТУРЫ ВЕБ-СЕРВИСА**

Описание и функциональное назначение компонентов инфраструктуры веб-сервиса.

Таблица 2

Название компонента	Тип	Назначение
Service for metabase	Микросервис	Сервис, устанавливаемый на сервер с метабазой. Отвечает за взаимодействие с пользователями, сервисами для баз данных и метабазой.
Service for database	Микросервис	Сервис, устанавливаемый на сервер с базой данных, для обеспечения взаимодействия с ней.
EntityLib	Библиотека классов	Библиотека объединяет классы, которые являются общими для микросервисов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 3

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

Описание и функциональное назначение классов.

Таблица 3

Компонент	Класс	Назначение
Service for metabase (Сервис для метабазы)	AccountController	Класс-контроллер, реализующий аутентификацию и авторизацию пользователей в сервисе для метабазы.
	MetabaseController	Класс-контроллер, отвечающий на запросы, которые предполагают взаимодействие с метабазой.
	JwtAuthenticationOptions	Класс, описывающий конфигурацию JWT-токена с точки зрения сервиса для метабазы для взаимодействия с сервисом для базы данных.
	MetabaseContext	Класс, который определяет контекст данных, используемый для взаимодействия с метабазой.
	MetabaseDb	Класс, который содержит информацию о базе данных по свойствам неорганических веществ и материалов, представленной в таблице DBInfo в метабазе.
	MetabasDbDto	Класс, который используется для передачи информации о базе данных в ответ на запросы пользователя.
	ServiceUser	Класс, который содержит информацию о зарегистрированном пользователе, сохраненной в файле.
	ServiceUserDto	Класс, который используется для передачи информации о пользователе при аутентификации.
	SpecialHttpClient	Модифицированный HTTP-клиент для отправки запросов сервису для базы данных.
	TableDto	Класс, который используется для передачи информации, используемой при построении таблицы. Html таблица является ответом на запросы пользователя.
	TokenGenerator	Класс, который отвечает за генерацию JWT-токена, используемого при запросах к сервису для базы данных.
	Program	Класс, запускающий процесс выполнения сервиса для метабазы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 3

Компонент	Класс	Назначение
Service for database (Сервис для базы данных)	DatabaseContoller	Класс-контроллер, отвечающий на запросы, которые предполагают взаимодействие с базой данных.
	AuthenticationOptions	Класс, описывающий конфигурацию JWT-токена с точки зрения сервиса для базы данных.
	DatabaseContext	Класс, который определяет контекст данных, используемый для взаимодействия с базой данных.
	Program	Класс, запускающий процесс выполнения сервиса для базы данных.
EntityLib (Библиотека сущностей)	FullPropertyEqualityComparer	Класс-компаратор, сравнивающий сущности MetabaseProperty по более широкому списку свойств.
	FullSystemEqualityComparer	Класс-компаратор, сравнивающий сущности MetabaseSystem по более широкому списку свойств.
	MetabaseProperty	Класс, описывающий метаданные о свойствах.
	MetabaseSystem	Класс, описывающий метаданные о системах.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 4

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ КЛАССОВ СЕРВИСА ДЛЯ МЕТАБАЗЫ

Таблица 4.1

Описание полей, методов и свойств класса AccountController				
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
Login	public	IActionResult	-	Get-запрос на получение формы для аутентификации.
Login	public	Task<IActionResult>	ServiceUserDto userDto	Аутентификация пользователя согласно полученным данным.
Logout	public	Task<IActionResult>	-	Выполнение выхода из аккаунта (текущей сессии).
SignIn	private	Task	ServiceUser user	Выполнение входа в аккаунт.
GetUser	private	ServiceUser?	string username, string password	Поиск зарегистрированного пользователя с такими данными.

Таблица 4.2

Описание полей, методов и свойств класса MetabaseController				
Поля				
Наименование	Модификатор доступа	Тип аргумента	Назначение	
_metabaseContext	private	MetabaseContext	Контекст взаимодействия с метабазой.	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
Index	public	IActionResult	-	Запрос на открытие домашней страницы сервиса.
Update	public	Task<IActionResult	-	Запрос на обновление метабазы.
GetActivatedServices	public	Task<IActionResult	-	Запрос на получение активных сервисов.
GetProperties	public	Task<IActionResult	-	Запрос на получение метаданных о свойствах из метабазы.
GetSystems	public	Task<IActionResult	-	Запрос на получение метаданных о системах из метабазы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 4.2

Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
UpdateProperties	private	Task<List<Metabase Property>>	HttpClient client, string dbServiceUrl	Обновление метаданных о свойствах в метабазе.
UpdateSystems	private	Task<List<Metabase System>>	HttpClient client, string dbServiceUrl	Обновление метаданных о системах в метабазе.

Таблица 4.3

Описание полей, методов и свойств класса JwtAuthenticationOptions				
Поля				
Наименование	Модификатор доступа	Тип аргумента	Назначение	
Issuer	public	string	Описание издателя токена.	
Audience	public	string	Описание потребителя токена.	
Key	public	string	Ключ для шифрации токена.	
Lifetime	public	int	Время жизни токена.	
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
GetSymmetricSecurityKey	public	Symmetric SecutiryKey	-	Получение ключа безопасности для генерации токена.

Таблица 4.4

Описание полей, методов и свойств класса MetabaseContext			
Свойства			
Наименование	Модификатор доступа	Тип аргумента	Назначение
PropertiesInfo	private	string	Описывает таблицу с метаданными о свойствах.
SystemInfo	private	string	Описывает таблицу с метаданными о системах.
DBInfo	private	string	Описывает таблицу о базах данных по свойствам неорганических веществ и материалов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 4.4

Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
OnModelCreating	protected	void	ModelBuilder modelBuilder	Конфигурация контекста взаимодействия с базой данных.

Таблица 4.5

Описание полей, методов и свойств класса MetabaseDb			
Свойства			
Наименование	Модификатор доступа	Тип аргумента	Назначение
DBID	public	int	Уникальный идентификатор базы данных по свойствам неорганических веществ и материалов.
Enabled	public	Boolean	Работает ли база данных в данный момент.
LastUpdate	public	DateTime	Время последнего обновления базы данных.
Name	public	string	Название базы данных.
Login	public	string	Логин для подключения к базе данных.
Password	public	string	Пароль для подключения к базе данных.
DBURL	public	string	Ссылка на страницу о базе данных.
DBGateURL	public	string	Ссылка на подключение к базе данных.
EmailManager	public	string	Электронный адрес менеджера базы данных.
WWWTemplatePage	public	string	Шаблон страницы о базе данных во Всемирной паутине.
Language	public	string	Язык информации, хранимой в базе данных.
Description	public	string	Описание базы данных.
DBServiceHost	public	string?	Хост сервиса для этого экземпляра базы данных.

Таблица 4.6

Описание полей, методов и свойств класса MetabaseDbDto			
Свойства			
Наименование	Модификатор доступа	Тип аргумента	Назначение
DBID	public	int	Уникальный идентификатор базы данных по свойствам неорганических веществ и материалов.
Name	public	string	Название базы данных.
EmailManager	public	string	Электронный адрес менеджера базы данных.
Language	public	string	Язык информации, хранимой в базе данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 4.6

Наименование	Модификатор доступа	Тип аргумента	Назначение	
Description	public	string	Описание базы данных.	
ServiceHost	public	string?	Хост сервиса для этого экземпляра базы данных.	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
FromMetabaseDb	public	MetabaseDbDto	MetabaseDb metabaseDb	Получение объекта MetabaseDbDto из объекта MetabaseDb.

Таблица 4.7

Описание полей, методов и свойств класса ServiceUser				
Свойства				
Наименование	Модификатор доступа	Тип аргумента	Назначение	
Username	public	string	Имя пользователя.	
Password	public	string	Пароль пользователя.	
Role	public	string	Роль пользователя в сервисе.	

Таблица 4.8

Описание полей, методов и свойств класса ServiceUserDto				
Свойства				
Наименование	Модификатор доступа	Тип аргумента	Назначение	
Username	public	string	Имя пользователя.	
Password	public	string	Пароль пользователя.	

Таблица 4.9

Описание полей, методов и свойств класса SpecialHttpClient				
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
GetHttpClient	public	HttpClient	-	Создание HTTP клиента со специальной конфигурацией.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 4.10

Описание полей, методов и свойств класса TableDto<T>				
Свойства				
Наименование	Модификатор доступа	Тип аргумента	Назначение	
Columns	public	ICollection<PropertyInfo>	Столбцы будущей таблицы. По своей сути это рефлексивно извлеченные свойства класса <T>.	
Items	public	Items<T>	Строки будущей таблицы (информация о каких-либо объектах).	
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
BuildModel	public	TableDto<T>	ICollection<T> properties	Построение модели таблицы на основе информации об объектах.

Таблица 4.11

Описание полей, методов и свойств класса TokenGenerator				
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
GetToken	public	string	IEnumerable<Claim> claims	Генерация JWT-токена.

Таблица 4.12

Описание полей, методов и свойств класса Program				
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
Main	public	void	string[] args	Запуск консольного приложения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 5

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ КЛАССОВ СЕРВИСА ДЛЯ БАЗЫ ДАННЫХ

Таблица 5.1

Описание полей, методов и свойств класса DatabaseController				
Поля				
Наименование	Модификатор доступа	Тип аргумента	Назначение	
_dbContext	private	DatabaseContext	Контекст взаимодействия с базой данных.	
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
Check	public	Task<IActionResult>	-	Запрос на проверку состояния работы сервиса.
GetProperties	public	Task<ICollection<MetabaseProperty>?>	-	Запрос на получение метаданных о свойствах из базы данных.
GetSystems	public	Task<ICollection<MetabaseSystem>?>	-	Запрос на получение метаданных о системах из базы данных.
ExecuteSqlQuery	private	Task	-	Выполнение sql-скрипта.

Таблица 5.2

Описание полей, методов и свойств класса AuthenticationOptions				
Поля				
Наименование	Модификатор доступа	Тип аргумента	Назначение	
Issuer	public	string	Описание издателя токена.	
Audience	public	string	Описание потребителя токена.	
Key	public	string	Ключ для шифрации токена.	
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
GetSymmetricSecurityKey	public	SymmetricSecutiryKey	-	Получение ключа безопасности для генерации токена.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 5.3

Описание полей, методов и свойств класса MetabaseContext				
Свойства				
Наименование	Модификатор доступа	Тип аргумента	Назначение	
PropertiesInfo	private	string	Описывает представление с метаданными о свойствах.	
SystemInfo	private	string	Описывает представление с метаданными о системах.	
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
OnModelCreating	protected	void	ModelBuilder modelBuilder	Конфигурация контекста взаимодействия с базой данных.

Таблица 5.4

Описание полей, методов и свойств класса Program				
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
Main	public	void	string[] args	Запуск консольного приложения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 6
ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И
СВОЙСТВ КЛАССОВ БИБЛИОТЕКИ СУЩНОСТЕЙ

Таблица 6.1

Описание полей, методов и свойств класса MetabaseProperty				
Свойства				
Наименование	Модификатор доступа	Тип аргумента	Назначение	
DBID	public	int	Идентификатор базы данных, откуда было взято данное свойство.	
PropId	public	int	Идентификатор свойства в базе данных.	
Name	public	string	Название свойства.	
Description	public	string	Описание свойства.	
WWWTemplatePage	public	string	Шаблон страницы о свойстве во Всемирной паутине.	
UpdateStatus	public	string	Статус обновления свойства относительно других таблиц.	
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
Equals	public	bool	Object other	Сравнение объектов MetabaseProperty по значениям следующих свойств: DBID, PropId.
GetHashCode	public	int	-	Получение хэш-кода объекта по значениям следующих свойств: DBID, PropId.
Modify	public	void	MetabaseProperty property	Обновление значений свойств экземпляра в зависимости от значений свойств объекта property.

Таблица 6.2

Описание полей, методов и свойств класса MetabaseSystem			
Свойства			
Наименование	Модификатор доступа	Тип аргумента	Назначение
DBID	public	int	Идентификатор базы данных, откуда была взята данная система.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 6.2

Наименование	Модификатор доступа	Тип аргумента	Назначение	
SystemId	public	int	Идентификатор системы в базе данных.	
ElemNumber	public	int	Количество элементов в системе.	
UpdateStatus	public	int	Статус обновления системы относительно других таблиц.	
_date	public	DateTime	Дата внесения в базу данных этой системы.	
Elements	public	string	Строковое представление элементов, представленных в системе.	
SystemInfo	public	string	Информация о системе.	
Description	public	string?	Описание системы.	
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
Equals	public	bool	Object other	Сравнение объектов MetabaseSystem по значениям следующих свойств: DBID, SystemId.
GetHashCode	public	int	-	Получение хэш-кода объекта по значениям следующих свойств: DBID, SystemId.
Modify	public	void	MetabaseSystem system	Обновление значений свойств экземпляра в зависимости от значений свойств объекта system.

Таблица 6.3

Описание полей, методов и свойств класса FullPropertyEqualityComparer				
Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
Equals	public	bool	MetabaseProperty? x, MetabaseProperty? y	Сравнение объектов MetabaseProperty по значениям следующих свойств: DBID, PropId, Name, Description, UpdateStatus, WWWTemplatePage.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 6.3

Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
GetHashCode	public	int	MetabaseProperty obj	Получение хэш-кода объекта по значениям следующих свойств: DBID, PropId, Name, Description, UpdateStatus, WWWTemplatePage.

Таблица 6.4

Описание полей, методов и свойств класса FullSystemEqualityComparer

Методы				
Наименование	Модификатор доступа	Тип аргумента	Аргументы	Назначение
Equals	public	bool	MetabaseSystem x, MetabaseSystem y	Сравнение объектов MetabaseSystem по значениям следующих свойств: DBID, SystemId, ElemNumber, UpdateStatus, Elements, SystemInfo, Description.
GetHashCode	public	int	MetabaseSystem obj	Получение хэш-кода объекта по значениям следующих свойств: DBID, SystemId, ElemNumber, UpdateStatus, Elements, SystemInfo, Description.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 7

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ
SQL-СКРИПТОВ**

Описание и функциональное назначение SQL-скриптов

Таблица 7

Скрипт	Назначение
BandGap	Создание или обновление представлений PropertiesInfo и SystemInfo в базе данных BandGap.
Crystal	Создание или обновление представлений PropertiesInfo и SystemInfo в базе данных Crystal.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата