

Общество с ограниченной ответственностью
КОРПОРАТИВНЫЕ ЭЛЕКТРОННЫЕ СИСТЕМЫ
(КЭЛС-центр)



**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
«ВЕБ-ИНТЕРФЕЙС ДЛЯ ПРОСМОТРА ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ»
(«ОКУЛЯР»)**

ОПИСАНИЕ ТЕХНИЧЕСКОЙ АРХИТЕКТУРЫ

2022 год

СОДЕРЖАНИЕ

1	Общие сведения.....	4
2	Структура программного обеспечения	4
2.1	Взаимодействие компонентов	4
2.2	Функциональная структура	5
3	Структура базы данных программного обеспечения	9
4	Применяемые технологии разработки программного обеспечения	15

Список сокращений и обозначений

API	– (англ. Application Programming Interface) программный интерфейс приложения
HTTP	– (англ. HyperText Transfer Protocol) протокол прикладного уровня передачи данных
PDM-система	– (англ. Product Data Management), система управления данными об изделии
БД	– база данных
ПО	– программное обеспечение
СУБД	– система управления базами данных

1 Общие сведения

Документ содержит описание технической архитектуры программного обеспечения (ПО) «Веб-интерфейс для просмотра технической документации» («ОКУЛЯР»). ПО «ОКУЛЯР» позволяет осуществлять поиск и просмотр в окне веб-браузера актуальной конструкторской и технологической документации на изделия, зарегистрированные в PDM-системе.

2 Структура программного обеспечения

2.1 Взаимодействие компонентов

ПО «ОКУЛЯР» состоит из следующих компонентов:

- «Веб-сервис» – выполняет функции авторизации пользователей в ПО, поиска и отображения документов, сохранения результатов поиска в базе данных;
- «Веб-интерфейс» – предоставляет функционал для ввода данных поиска, отображения результатов поиска, просмотра файлов документов;
- «Библиотека интеграции» – реализует подключение к PDM-системе, выполняет поиск объектов и документов, загружает файлы документов;
- «База данных» – выполняет функции по хранению информации с результатами поиска для последующего быстрого доступа.

На рисунке 2.1 показана схема взаимодействия компонентов ПО «ОКУЛЯР» и PDM-системы.

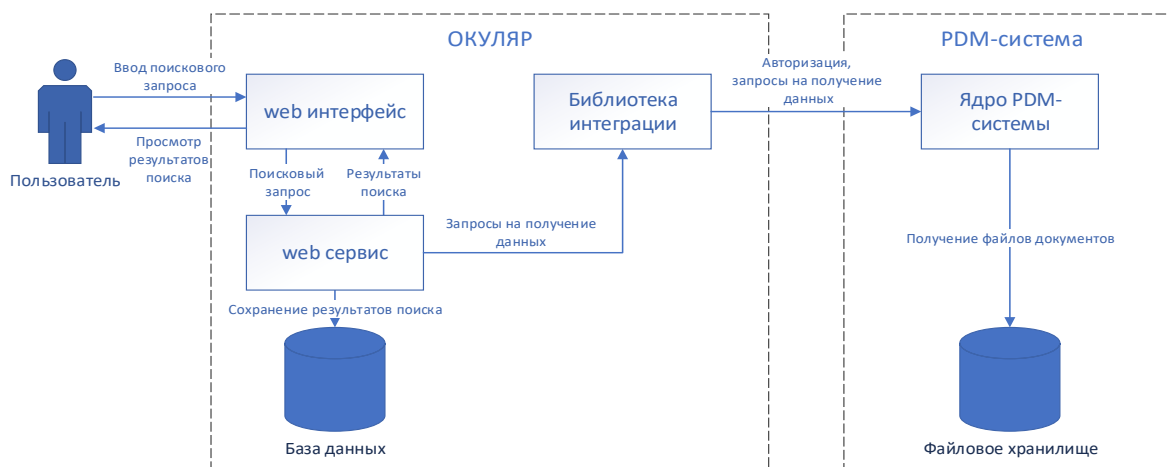


Рисунок 2.1 – Схема взаимодействия компонентов ПО «Окуляр»

2.2 Функциональная структура

На рисунке 2.2 показана функциональная структура ПО «ОКУЛЯР».

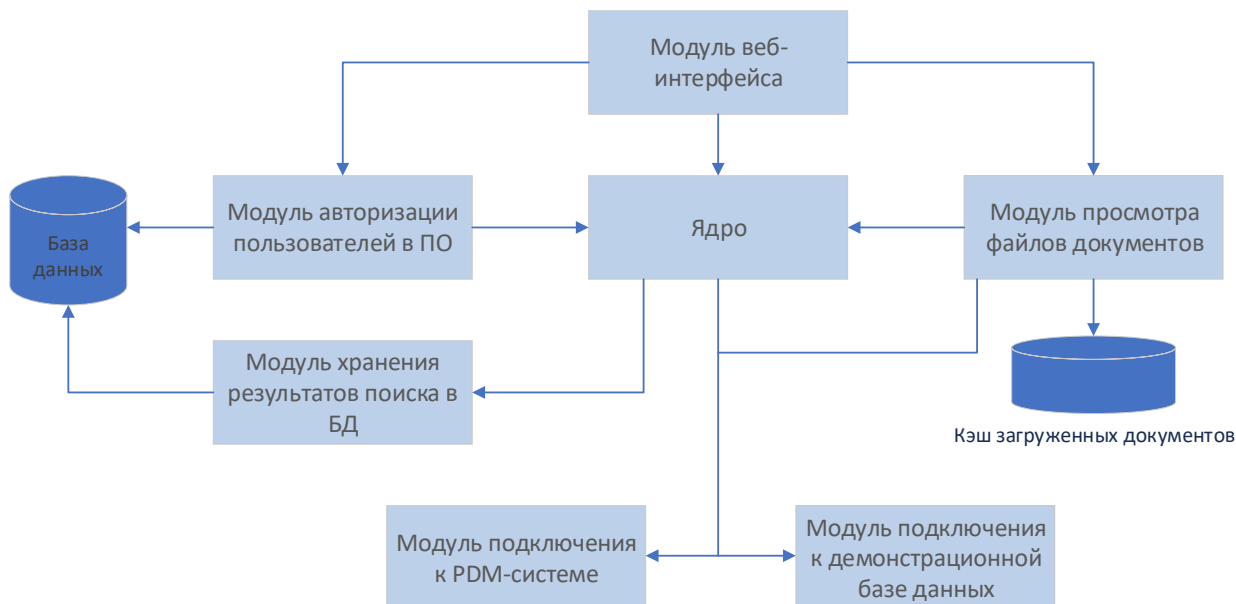


Рисунок 2.2 – Функциональная структура ПО «ОКУЛЯР»

ПО «ОКУЛЯР» включает следующие функциональные модули:

- «Ядро системы» – представляет собой веб-сервис, в котором реализован веб-сервер «Kestrel», обеспечивающий подключение пользователей по протоколу HTTP. Веб-сервер реализует следующий API контракт:
 - Контроллер аутентификации доменных пользователей (рисунок 2.3);

DomainAuthentication		
POST	/api/DomainAuthentication/DomainUserLogin	Авторизация доменного пользователя
POST	/api/DomainAuthentication/DomainUserRegister	Регистрация доменного пользователя
POST	/api/DomainAuthentication/UserLogoff	Выход пользователя

Рисунок 2.3 – Методы API «DomainAuthntication»

- Контроллер поиска изделий (рисунок 2.4);

Searching			^
GET	/api/Searching/Get	Получение результатов поиска для текущего пользователя	✓ 🔒
POST	/api/Searching/Post	Сохранение результатов поиска	✓ 🔒
GET	/api/Searching/CheckFinish	Проверка заершения поиска	✓ 🔒
POST	/api/Searching/FindApplicability	Поиск применимости	✓ 🔒
GET	/api/Searching/CheckApplicabilityFinish	Проверка заершения поиска применимости	✓ 🔒

Рисунок 2.4 – Методы API «Searching»

- Контроллер изделий (рисунок 2.5);

Products			^
GET	/api/Products/GetComposition	Получение состава изделий	✓ 🔒
GET	/api/Products/GetProducts	Получение списка изделий	✓ 🔒
GET	/api/Products/GetApplicability	Получение применимости	✓ 🔒
GET	/api/Products/GetProduct/{id}	Получение изделия	✓ 🔒
GET	/api/Products/GetDocuments/GetDocuments	Получение документов	✓ 🔒
GET	/api/Products/GetProductParents/ProductParents	Получение родительских изделий	✓ 🔒
GET	/api/Products/GetDocumentFiles/DocumentFiles	Получение списка файлов документа	✓ 🔒
POST	/api/Products/EnableComposition	Включение отображения применимости	✓ 🔒
GET	/api/Products/DownloadFile	Загрузка файла документа	✓ 🔒

Рисунок 2.5 – Методы API «Products»

- Контроллер управления пользователями (рисунок 2.6);

Users		
GET	/api/Users/Get	Получение перечня пользователей
GET	/api/Users/Get/{id}	Получение пользователя
GET	/api/Users/AllUserStatuses	Получение статусов пользователей
PUT	/api/Users/Put	Сохранение пользователя
POST	/api/Users/Post	Добавление пользователя
DELETE	/api/Users/Delete	Удаление пользователя
POST	/api/Users/PasswordReset	Сброс пароля
GET	/api/Users/GetUserAction	Получение действий пользователя
GET	/api/Users/GetUserAction2	Получение действия пользователя
GET	/api/Users/GetUserAction3	Получение действий пользователя для таблицы

Рисунок 2.6 – Методы API «Users»

- «Модуль веб-интерфейса» – формирует пользовательские формы, обеспечивающие выполнение следующих действий пользователя:
 - авторизация;
 - ввод данных для поиска;
 - просмотр истории поиска;
 - просмотр структуры состава изделий;
 - просмотр перечня документации;
 - просмотр файлов загруженных документов.
- «Модуль авторизации пользователей в ПО» – реализует следующий функционал:
 - проверка зарегистрированных пользователей, при этом пользователи могут быть доменными или внешними. Доменные пользователи авторизуются автоматически, внешним пользователям требуется ввести логин/пароль;

- ведение записи журнала действий пользователей. Вся информация хранится в базе данных;
- «Модуль подключения к PDM» – при обращении к PDM-системе для поиска или загрузки файлов документов, модуль создает подключение к PDM-системе в пуле подключений. При этом отслеживается и ограничивается количество подключений в пуле. Таким образом, при одновременном подключении множества пользователей происходит ограничение запрашиваемых данных в ожидании свободных подключений. В случае нехватки доступных подключений в пуле, пользовательская сессия будет ожидать освобождения. После выполнения операции с доступом к PDM-системе, подключение закрывается, освобождая тем самым место в пуле подключений. Это необходимо для контроля за нагрузкой на доступ к данным PDM-системы, т.к. подключение фактически производится через один аккаунт. Пример пула подключений показан на рисунке 2.7;

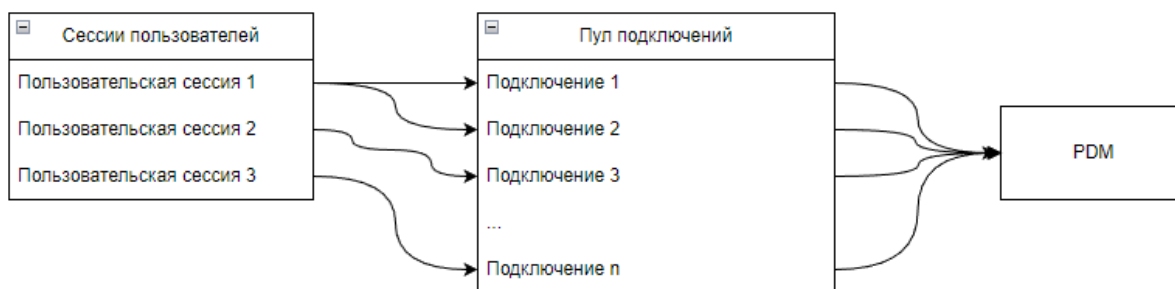


Рисунок 2.7 – Пул подключений к PDM-системе

Модуль подключается к PDM-системе, выполняет непосредственные запросы на получение объектов, документов, файлов документов, проверяет статусы объектов. После получения данных из PDM-системы, модуль сохраняет их в базе данных, предоставляя возможность отобразить результаты запроса без повторной загрузки из PDM-системы;

- «Модуль подключения к демонстрационной базе данных» – данный модуль получает объекты, документы, файлы документов из демонстрационной базы данных (БД);

- «Модуль просмотра файлов документов» – отображает файл документа в специальном окне, запрещающем скачивание файлов. Данное окно предоставляет возможность постраничного отображения содержимого документа, масштабирования содержимого документа. Модуль загружает файлы из кэша документов, куда они загружаются по запросу модулем подключения к PDM-системе или модулем подключения к демонстрационной базе данных;
- «Модуль хранения результатов поиска в БД» – сохраняет результаты поиска в базе данных для последующего отображения их пользователям, тем самым уменьшая нагрузку на модуль получения данных. При первом запуске ПО модуль выполняет миграцию данных, создавая тем самым саму базу данных, а также создавая необходимую схему данных и заполняя предварительно таблицы БД.

3 Структура базы данных программного обеспечения

Структура базы данных ПО показана на рисунке 3.1.

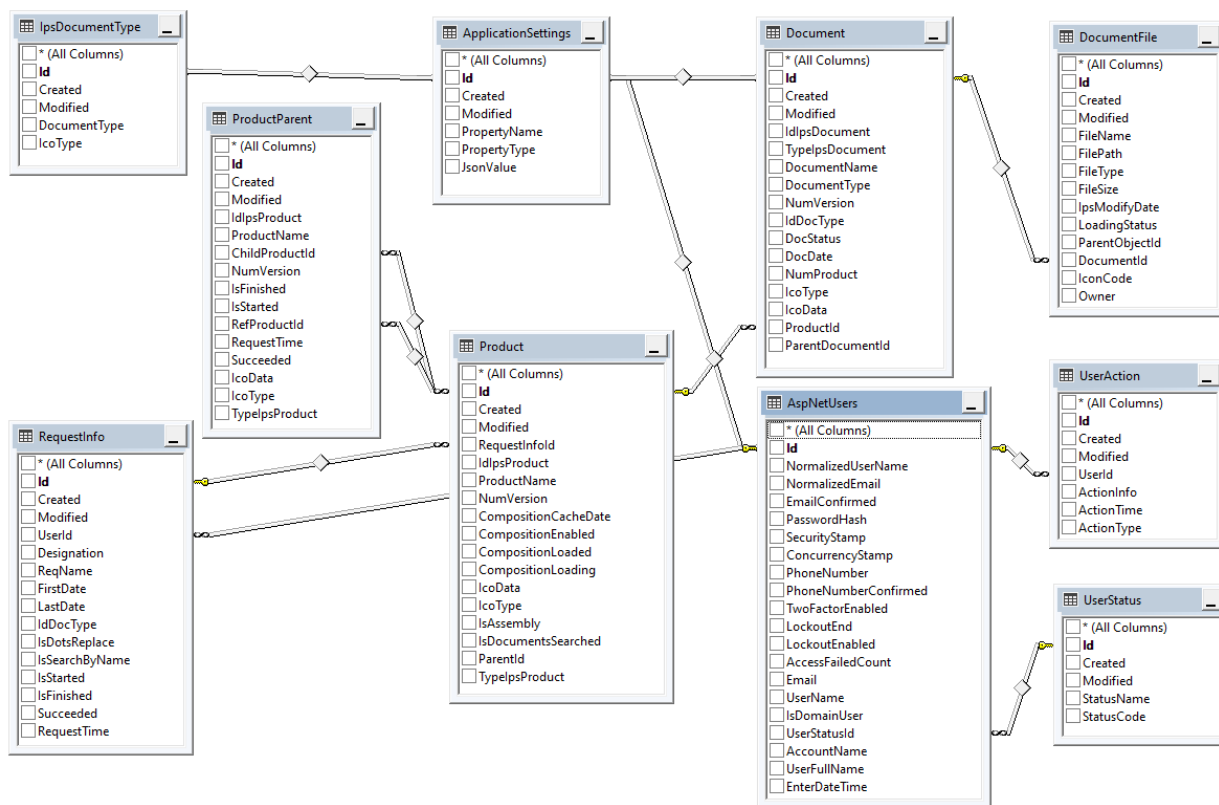


Рисунок 3.1 – Структура базы данных ПО «ОКУЛЯР»

ПО «ОКУЛЯР» поддерживает работу со следующими системами управления базами данных (СУБД):

- PostgreSQL,
- Microsoft SQL Server.

В таблице «RequestInfo», структура которой показана в 3.1, хранится информация о поисковых запросах пользователей.

Таблица 3.1 – Таблица «RequestInfo»

Поле	Комментарий
UserId	ИД пользователя, выполнившего запрос
Designation	Обозначение
ReqName	Наименование
FirstDate	Дата ввода с
LastDate	Дата ввода по
IdDocType	Тип документа
IsDotsReplace	Признак замены точки на тире при поиске
IsSearchByName	Признак поиска по имени
IsStarted	Признак того, что поиск запущен
IsFinished	Признак того, что поиск завершен
Succeeded	Признак успешности поиска
RequestTime	Время поиска

В таблице «Product», структура которой показана в 3.2, хранится информация о найденном изделии.

Таблица 3.2 – Таблица «Product»

Поле	Комментарий
RequestInfoId	ИД запрос на поиск
ParentId	ИД родительского изделия
IdIpsProduct	ИД изделия в PDM
TypeIpsProduct	Тип изделия в PDM
ProductName	Наименование
NumVersion	Номер версии
IcoType	Тип иконки
IcoData	Вlob иконки
IsAssembly	Признак сборки
IsDocumentsSearched	Признак того, что документы найдены для изделия
CompositionEnabled	Признак включенного состава
CompositionLoaded	Признак, что состав загружен
CompositionLoading	Признак, что состав загружается
CompositionCacheDate	Дата загрузки состава изделия

В таблице «Document», структура которой показана в 3.3, хранится информация о документах.

Таблица 3.3 – Таблица «Document»

Поле	Комментарий
IdIpsDocument	ИД документа в PDM
TypeIpsDocument	Тип документа в PDM
DocumentName	Наименование документа
DocumentType	Тип

Поле	Комментарий
NumVersion	Номер версии
IdDocType	ИД типа документа
DocStatus	Статус документа
DocDate	Дата документа
NumProduct	Номер документа
IcoType	Тип иконки
IcoData	Blob иконки
ProductId	ИД изделия
ParentDocumentId	ИД родительского документа

В таблице «DocumentFile», структура которой показана в 3.4, хранятся файлы документов.

Таблица 3.4 – Таблица «DocumentFile»

Поле	Комментарий
FileName	Имя файла
FilePath	Путь
FileType	Тип файла
FileSize	Размер
IpsModifyDate	Дата модификации
LoadingStatus	Статус загрузки файла
DocumentId	ИД документа
IconCode	Код иконки
Owner	Владелец файла
ViewerType	Тип просмотрщика

В таблице «ProductParent», структура которой показана в 3.5, хранится список родительских элементов изделий.

Таблица 3.5 – Таблица «ProductParent»

Поле	Комментарий
IdIpsProduct	ИД изделия в PDM
ProductName	Наименование изделия
ChildProductId	ИД потомка-изделия
NumVersion	Номер версии
TypeIpsProduct	Тип изделия в PDM
IcoType	Тип иконки
IcoData	Blob иконки
IsStarted	Признак запуска поиска
IsFinished	Признак завершения поиска
Succeeded	Успех поиска
RequestTime	Время выполнения запроса
RefProductId	Ссылка на изделие

В таблице «AspNetUsers», структура которой показана в 3.6, хранится список пользователей ПО.

Таблица 3.6 – Таблица «AspNetUsers»

Поле	Комментарий
Id	ИД пользователя
Email	Почта
UserFullName	Полное имя пользователя

Поле	Комментарий
AccountName	Имя аккаунта
IsDomainUser	Признак того, что пользователь доменный
UserStatusId	Статус пользователя
EnterDateTime	Дата/время последнего входа

В таблице «UserAction», структура которой показана в 3.7, хранится журнал действий пользователя.

Таблица 3.7 – Таблица «UserAction»

Поле	Комментарий
UserId	ИД пользователя
ActionInfo	Информация о действии
ActionTime	Время
ActionType	Тип действия

В таблице «UserStatus», структура которой показана в 3.8, хранится справочник статусов пользователя.

Таблица 3.8 – Таблица «UserStatus»

Поле	Комментарий
Id	ИД статуса
StatusName	Наименование статуса
StatusCode	Код статуса

В таблице «IpsDocumentType», структура которой показана в 3.9, хранится справочник типов документов.

Таблица 3.9 – Таблица «IpsDocumentType»

Поле	Комментарий
Id	ИД типа
DocumentType	Тип документа
IcoType	Тип иконки

Демонстрационная база данных ПО «ОКУЛЯР» расположена в папке «DemoData» и содержит:

- «Documents» – папка с файлами документов,
- «Icons» – папка с файлами иконок изделий, документов и файлов,
- «products.json» – файл, в котором содержится список изделий, документов и файлов,
- «products.json» – файл, в котором содержится список типов изделий и документов.

4 Применяемые технологии разработки программного обеспечения

Основной сервис ПО «ОКУЛЯР» реализован на C# .Net Core 3.1. Поставляется в виде Self-contained дистрибуции .NET Core приложений.

Библиотека «IpsConnector», осуществляющая подключение к PDM-системе, реализована на C# .Net Framework 4.6.1. Поставляется в виде COM-библиотеки DLL.

Библиотека «IpsConnectorDemo», осуществляющая подключение к демонстрационным данным, реализована на C# .Net Framework 4.6.1. Поставляется в виде COM-библиотеки DLL.

Библиотека «IpsWrapper», предоставляющая интерфейс для подключения библиотек «IpsConnector» и «IpsConnectorDemo», реализована на C# .Net Framework 4.6.1. Поставляется в виде библиотеки DLL.

Инсталлятор приложения осуществляет развертывание приложения в заданную папку, а также регистрацию библиотек IpsConnector и IpsConnectorDemo. Реализован с помощью приложения InnoSetup.