Проектирование системы

Для проектирования сложной, современной автоматизирующей системы применяют унифицированный язык визуального моделирования Unified Modeling Language (UML).

Разработка контекстной диаграммы и описание сценариев использования системы

Для описания системы на концептуальном уровне используется контекстная диаграмма вариантов использования системы, изображенная на рисунке 1.

Она позволяет быстро, кратко и ёмко описать назначение и границы системы, выявить и устранить коллективные расхождения в их понимании, показать и договориться о её масштабе.

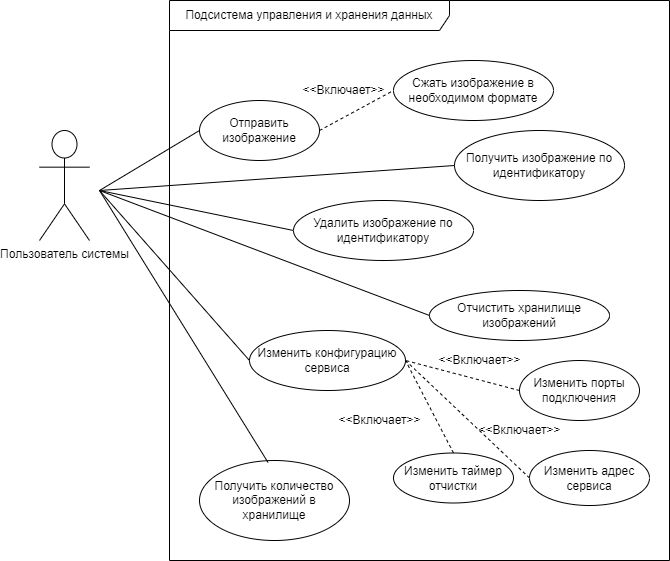


Диаграмма вариантов использования

Описания вариантов использования представлены в таблицах 1-10.

Вариант использования системы «Отправление изображение в хранилище»

|  |  |
| --- | --- |
| Название варианта | Отправление изображение в хранилище |
| Цель | Отправить новое изображение на сервис для хранения |
| Действующие лица | Пользователь системы |
| Краткое описание | Пользователь загружает изображение и отправляет запросом его на сервис, далее сервис сохраняет его в хранилище и выдает ему уникальный идентификатор для обращения к изображению |
| Тип варианта | Основной |

Вариант использования системы «Сжатие изображения»

|  |  |
| --- | --- |
| Название варианта | Сжатие изображения |
| Цель | Отправить новое изображение на сервис для хранения с параметром сжатия |
| Действующие лица | Пользователь системы |
| Краткое описание | Пользователь загружает изображение и отправляет запросом (с параметром сжатия) его на сервис, далее сервис обрабатывает изображение до нудного формата и сохраняет его в хранилище и выдает ему уникальный идентификатор для обращения к изображению |
| Тип варианта | Включающий |

Вариант использования системы «Получение изображение»

|  |  |
| --- | --- |
| Название варианта | Получение изображения |
| Цель | Получить изображение из хранилища |
| Действующие лица | Пользователь системы |
| Краткое описание | Пользователь отправляет запрос сервису с параметров, в котором содержится идентификатор изображения, и на выходе получает изображение с хранилища |
| Тип варианта | Основной |

Вариант использования системы «Удаление из памяти изображения»

|  |  |
| --- | --- |
| Название варианта | Удаление из памяти изображения |
| Цель | Убрать изображение из хранилища |
| Действующие лица | Пользователь системы |
| Краткое описание | Пользователь отправляет запрос сервису с параметров, в котором содержится идентификатор изображения. Сервис убирает из хранилища изображение. |
| Тип варианта | Основной |

Вариант использования системы «Очистка хранилища»

|  |  |
| --- | --- |
| Название варианта | Очистка хранилища |
| Цель | Очистить память хранилища от изображений |
| Действующие лица | Пользователь системы |
| Краткое описание | Пользователь отправляет запрос на очистку сервису. Сервис производит очистку хранилища. |
| Тип варианта | Основной |

Вариант использования системы «Получение информации о хранилище»

|  |  |
| --- | --- |
| Название варианта | Получение информации о хранилище |
| Цель | Получить информацию о количестве изображений в хранилище сервиса |
| Действующие лица | Пользователь системы |
| Краткое описание | Пользователь отправляет запрос на информацию. Сервис отправляет данные о хранилище пользователю. |
| Тип варианта | Основной |

Вариант использования системы «Изменение конфигурации сервиса»

|  |  |
| --- | --- |
| Название варианта | Изменение конфигурации сервиса |
| Цель | Поменять настройки сервиса |
| Действующие лица | Пользователь системы |
| Краткое описание | Пользователь открывает конфигурацию и меняет необходимые ему параметры системы |
| Тип варианта | Основной |

Вариант использования системы «Изменение порта сервиса»

|  |  |
| --- | --- |
| Название варианта | Изменение порта сервиса |
| Цель | Изменить параметр порта для подключения |
| Действующие лица | Пользователь системы |
| Краткое описание | Пользователь открывает конфигурацию и меняет порт подключения к сервису |
| Тип варианта | Включающий |

Вариант использования системы «Изменение адреса сервиса»

|  |  |
| --- | --- |
| Название варианта | Изменение порта сервиса |
| Цель | Изменить параметр адреса для подключения |
| Действующие лица | Пользователь системы |
| Краткое описание | Пользователь открывает конфигурацию и меняет адрес подключения к сервису |
| Тип варианта | Включающий |

Вариант использования системы «Изменение таймера хранилища»

|  |  |
| --- | --- |
| Название варианта | Изменение времени таймера очистки |
| Цель | Изменить время таймера хранения изображения |
| Действующие лица | Пользователь системы |
| Краткое описание | Пользователь открывает конфигурацию и меняет время таймера |
| Тип варианта | Включающий |

Типичный ход событий позволяет увидеть наглядное представление общения пользователя с системой. Описание типичного хода событий при «Отправка изображения» представлено в таблице 11.

Описание типичного хода событий при «Отправка изображения»

|  |  |
| --- | --- |
| Действия пользователя | Отклик системы |
| Загружает изображение и отправляет POST запросом на сервер |  |
|  | Получает запрос на добавление с изображением, сохраняет в хранилище, формирует ответ с идентификатором и отправляет клиенту |
| Получает ответ от сервера с идентификатором сохраненного изображения |  |

Описание типичного хода событий при «Получение изображения» представлено в таблице 12.

Описание типичного хода событий при «Получение изображения»

|  |  |
| --- | --- |
| Действия пользователя | Отклик системы |
| Отправляет GET запрос с параметром виде идентификатора изображения на сервер |  |
|  | Получает запрос на отправку изображения. Ищет по полученному идентификатору изображение, далее формирует ответ с изображением и отправляет клиенту |
| Получает ответ от сервера с изображением |  |

Описание типичного хода событий при «Отчистка хранилища» представлено в таблице 13.

Описание типичного хода событий при «Отчистка хранилища»

|  |  |
| --- | --- |
| Действия пользователя | Отклик системы |
| Отправляет DELETE запрос |  |
|  | Получает запрос на очистку хранилища. Очищает хранилище и формирует ответ, далее отправляет клиенту |
| Получает ответ от сервера о выполнении операции |  |

Описание типичного хода событий при «Удаление изображения» представлено в таблице 14.

Описание типичного хода событий при «Удаление изображения»

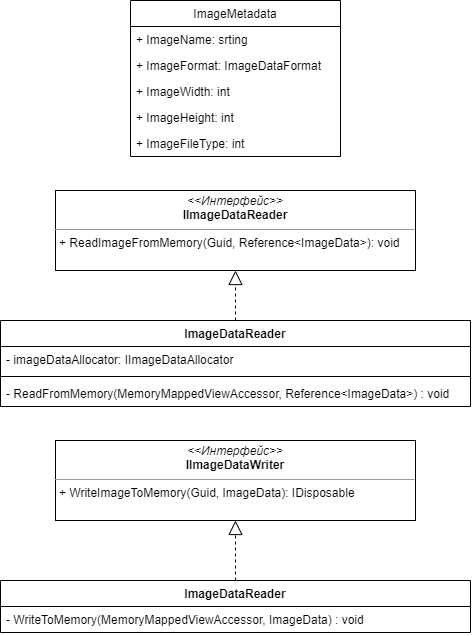
|  |  |
| --- | --- |
| Действия пользователя | Отклик системы |
| Отправляет DELETE запрос с параметром виде идентификатора изображения на сервер |  |
|  | Получает запрос удаление изображения. Ищет по полученному идентификатору изображение, далее удаляет и формирует ответ и отправляет клиенту |
| Получает ответ от сервера о выполнении операции |  |

В данной главе был произведён полный анализ вариантов использования системы с подробным их описанием.

Построение аналитической и структурной диаграммы классов

Диаграмма классов демонстрирует структуру иерархии классов системы, их кооперации, поля, методы, взаимосвязи между ними. Аналитическая диаграмма классов, показывающая иерархию классов с их взаимосвязями.

Диаграмма классов библиотеки записи изображений в память, представлена на рисунке 6. Данная библиотека предназначена для работы с изображениями в памяти.



Структурная диаграмма классов библиотеки записи данных в память

Описание классов библиотеки представлено в таблице 20.

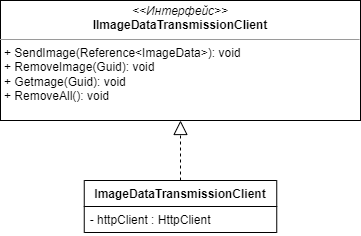
Описание классов библиотеки записи данных

|  |  |
| --- | --- |
| Класс | Описание |
| ImageMetadata | Класс, представляющий из себя модель данных изображений |
| IImageDataReader | Интерфейс, описывающий чтение изображения |
| ImageDataReader | Класс, реализующий функционал чтения изображения из памяти |
| IImageDataWriter | Интерфейс, описывающий запись изображения |
| ImageDataReader | Класс, реализующий функционал записи изображения в памяти |

Описание методов клиентской библиотеки

|  |  |
| --- | --- |
| Метод | Описание |
| ReadImageFromMemory | Метод, читающий изображение из памяти устройства |
| ReadFromMemory | Метод, читающий уже изображение в ячейках временной памяти (ОЗУ). Отличие от метода «ReadImageFromMemory» заключается в том, что «ReadImageFromMemory» только формирует образ изображения, которое стоит достать из памяти, а «ReadFromMemory» достает из памяти |
| WriteImageToMemory | Метод, записывающий изображение в память устройства |
| WriteToMemory | Метод, аналогично методу «ReadFromMemory» записывает в ячейки временной памяти изображение |

Диаграмма классов клиентской библиотеки, представлена на рисунке 7. Библиотека предназначена для работы с сервисом, то есть представляет из себя клиентскую часть системы.



Структурная диаграмма классов клиентской библиотеки

Описание классов клиентской библиотеки представлено в таблице 21.

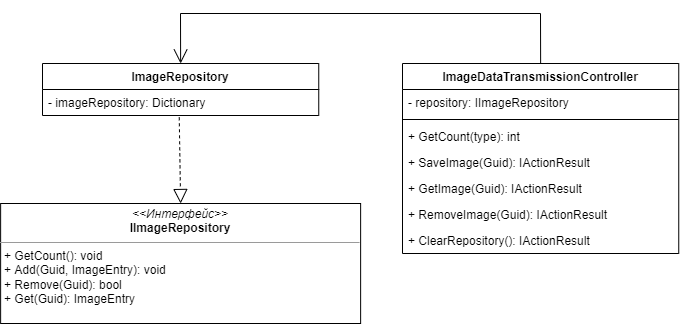
Описание классов клиентской библиотеки

|  |  |
| --- | --- |
| Класс | Описание |
| ImageDataTransmissionClient | Класс, взаимодействия клиента с сервисом |
| IImageDataTransmissionClient | Интерфейс, описывающий обращение клиента к сервису |

Описание методов клиентской библиотеки

|  |  |
| --- | --- |
| Метод | Описание |
| SendImage | Метод клиента, отправляет изображение сервису для хранения |
| RemoveImage | Метод клиента, отправляет запрос на удаление изображение из репозитория |
| Getmage | Метод клиента, отправляет запрос на получение изображение из репозитория |
| RemoveAll | Метод клиента, отправляет запрос на полную очистку репозитория |

Диаграмма классов сервиса, представлена на рисунке 8. Описывает работу самого сервиса, далее контроллера, которому обращается клиентская часть, а в данной диаграмме описываются классы по работе с репозиторием изображения.



Структурная диаграмма классов сервиса

Описание классов сервиса представлено в таблице 21. Методы сервиса описаны в таблице 22.

Описание классов сервиса

|  |  |
| --- | --- |
| Класс | Описание |
| ImageDataTransmissionController | Класс, отвечающий за работу контроллера для работы сервиса |
| IImageRepository | Интерфейс, описывающий работу репозитория изображений |
| ImageRepository | Класс, реализующий функционал работы репозитория хранения изображений |

Описание методов

|  |  |
| --- | --- |
| Метод | Описание |
| GetCount | Метод отдает количество объектов в хранилище |
| Add | Метод добавляет объект в хранилище |
| Remove | Метод удаляет выбранный объект из хранилища |
| Get | Метод отдает объект из хранилища |
| SaveImage | Метод контроллера, сохраняющий изображение в хранилище |
| GetImage | Метод контроллера, отдает изображение из хранилища |
| RemoveImage | Метод контроллера, удаляющий изображение из хранилища |
| ClearRepository | Метод контроллера, отчищает хранилище от объектов |

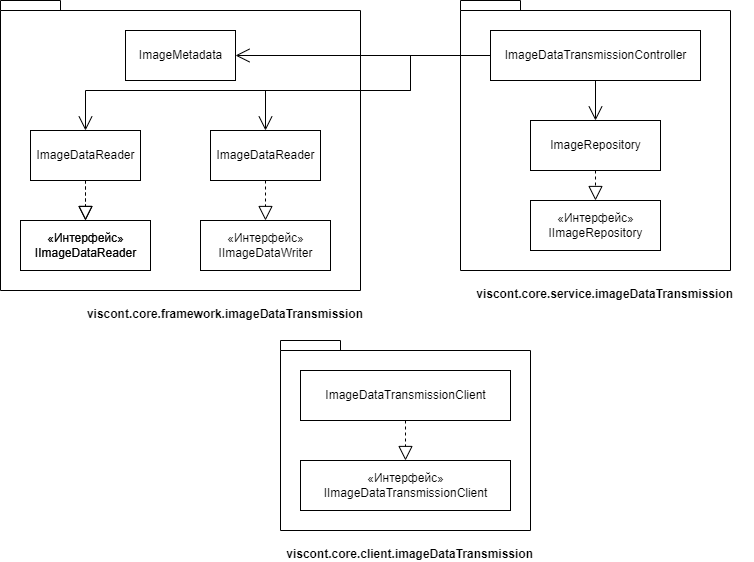
Диаграммы пакетов системы

Диаграмма пакетов служит, в первую очередь, для организации элементов в группы по какому-либо признаку с целью упрощения структуры и организации работы с моделью системы.

Система состоит из трех пакетов:

* Пакет классов «viscont.core.framework.imageDataTransmission». Расположены классы, отвечающие за работу (чтение и запись) данных в ОЗУ и вид данных (модели);
* Пакет «viscont.core.service.imageDataTransmission». Расположены классы, отвечающие за работу сервиса, контролер, позволяющий обрабатывать запросы и формировать ответы, работу с репозиторием изображений.
* Пакет классов «viscont.core.client.imageDataTransmission». Расположены классы, отвечающие за работу клиентской части системы, подключение к сервису, передачи и получение данных.

Пакеты классов системы продемонстрированы на изображении 20.



Пакеты классов системы

Также на диаграмме пакетов отображены отношения классов друг с другом. Где наглядно видно, сервис напрямую работает с библиотекой записи, предназначенной для записи данных в ОЗУ устройства, а клиентская библиотека не зависима от остальных пакетов.

Проектирование базы данных

В системе будет использоваться распространенный вид базы данных − реляционная. Представляет из себя табличное представление данных, что в свою очередь является простой и доступной к пониманию формой подачи информации для пользователя. К большим достоинствам относится лёгкость манипуляции с данными и их обработкой.

История кадров распознавания отображена в таблице «CognexRecognitions», и состоит из следующих атрибутов:

* Идентификатор распознавания;
* Идентификатор операции обработки;
* Идентификатор синхронизации;
* Время отметки;
* Дополнительные данные о кадре;
* Номер линии (расположение камеры);
* Название камеры;
* Роль камеры;
* Данные изображение;
* Идентификатор изображения;
* Последовательность приобретений индекс;
* Количество кадров для приобретений;
* Количество кадров.

Физическая диаграмма представлена на рисунке 5.

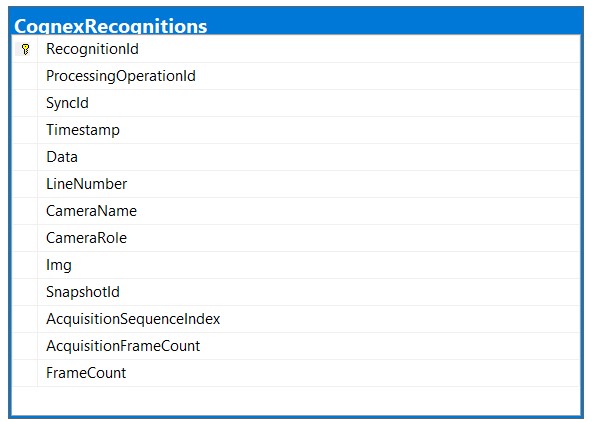


Рисунок 5 — Физическая диаграмма базы данных

В таблице 3 приведено подробное описание сущности «CognexRecognitions» в базе данных.

Таблица 2 - Описание полей таблицы «CognexRecognitions»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Вид | Тип | Описание |
| RecognitionId | PK | uniqueidentifier | Идентификатор распознавания |
| ProcessingOperationId |  | uniqueidentifier | Идентификатор операции обработки |
| SyncId |  | uniqueidentifier | Идентификатор синхронизации |
| Timestamp |  | datetime2(7) | Время отметки |
| Data |  | nvarchar(MAX) | Дополнительные данные о кадре |
| LineNumber |  | int | Номер линии (расположение камеры) |
| CameraName |  | nvarchar(MAX) | Название камеры |
| CameraRole |  | nvarchar(MAX) | Роль камеры |
| Img |  | varbinary(MAX) | Данные изображение |
| SnapshotId |  | uniqueidentifier | Идентификатор изображения |
| AcquisitionSequenceIndex |  | bigint | Последовательность приобретений индекс |
| AcquisitionFrameCount |  | bigint | Количество кадров для приобретений |
| FrameCount |  | bigint | Количество кадров |