Исследование предметной области

Анализ деятельности предприятия

Компания ООО «Малленом Системс» является разработчиком систем технического зрения, машинного обучения и компьютерного моделирования для использования в различных деятельности.

«Малленом Системс» имеет большой опыт успешной реализации наукоемких IT-проектов в сфере транспорта, машиностроения, нефтегазовой, металлургической, пищевой, фармацевтической, алмазодобывающей, атомной и других отраслях промышленности [].

Деятельность работы компании:

* Технологический контроль производства продукции с помощью машинного зрения.
* Видеоконтроль и учет автомобильного и ж/д транспорта.
* Интеллектуальные системы транспортного моделирования и управления дорожным движением.
* Big Data (системы обработки больших массивов данных), машинное обучение и системы поддержки принятия решений.

В основе продуктов «Малленом Системс» лежат собственные разработки:

* Высокоэффективные алгоритмы распознавания номеров на транспорте.
* Технологии анализа изображений: обнаружение дефектов, проверка целостности, определение размеров и формы объектов, обнаружение и распознавание надписей.
* Уникальная система машинного обучения (с учителем и без учителя), позволяющая быстро и качественно строить модели сложных систем, выявлять закономерности в огромных массивах данных, реализовывать обучение и самообучение созданных моделей и продуктов.
* Компьютерная имитационная модель движущихся потоков на основе агентного дискретно-событийного подхода (позволяет моделировать сложные транспортные и логистические системы).
* Технология построения сложных систем контроля и управления с элементами искусственного интеллекта, использующая техническое зрение, машинное обучение и разработанные математические модели.

На данный момент компания владеет и развивает следующие автоматизированные системы контроля производства:

⎯ Прослеживание и контроль качества продукции системы: ВИСКОНТ.Гранулы, ВИСКОНТ.Трубы, ВИСКОНТ.Свекла и ВИСКОНТ.Капсулы;

⎯ Маркировки: ВИСКОНТ.Алко, ВИСКОНТ.Фарма и ВИСКОНТ.Молоко;

⎯ Контроль людей и событий: система EYECONT;

⎯ Видеоконтроль и учёт автотранспорта: Автомаршал, Автомаршал.Весовая, Автомаршал.Gate, Дорожный пристав и AVEDEX;

⎯ Видеоконтроль и учёт ж/д транспорта: система АРСИС [].

В данном разделе был произведен анализ деятельности компании ООО «Малленом Системс», в ходе которого были выделены основные направления разрабатываемых компанией систем. Каждое направление разработки напрямую связано с работой с видео и кадрами (изображениями).

Анализ предметной области

Компания ООО «Малленом Системс» работает со следующими видами информации: изображение, видео и текст.

Изображения, пример содержания информации иллюстрирован на рисунке 1. В него входят снимки объектов (люди, молочная продукция, гранулы, машины, дата элементы, представляющие из себя различные Data-Matrix, штрих-коды и т.д.) для дальнейшей обработки в разрабатываемых системах компании.



Изображение упаковок с Data-Matrix

Видео. Состоят из длительных видео материалов содержащие определенные объекты (люди, машины), также в это понятие входит прямые трансляции или различные видеопотоки.

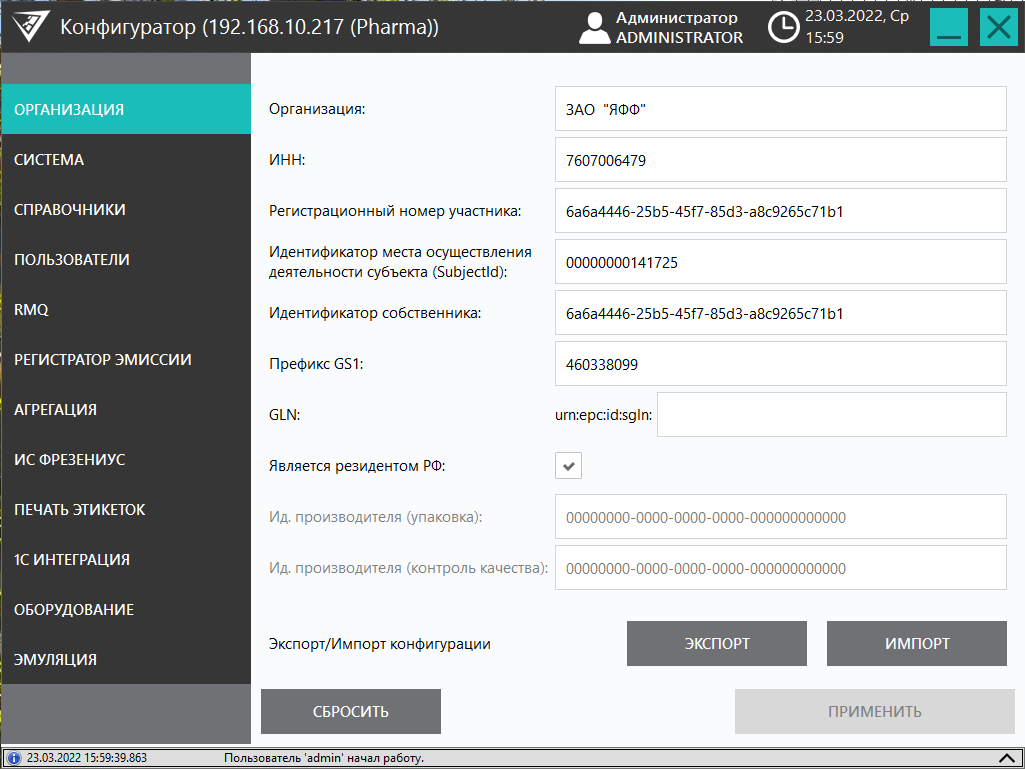
Пример содержания видео материала изображен на рисунке 2.



Видео материал участка работ

Текст. Определенная для конкретной системы информация или данные, которые заносятся в АС.

Пример данных изображен на рисунке 3.



Пример блоков данных об организации в системе «Висконт.Фарма»

Форма одной из программы системы ВИСКОНТ.Фарма, где водятся и отображаются основные данные об организации (ИНН, наименование и т.д.).

В основном в компании работают с изображениями и видео. Видео в конечном результате рабиваются на карды, а кадры в свою очередь являются файлами типа изображения.

Графические используемые форматы изображений для обработки:

* Формат JPEG (Joint Photographic Expert Group). Изображения имеют расширения jpg, jpe, jpeg или jfif. Алгоритм JPEG позволяет сжимать изображение как с потерями, так и без потерь.
* Формат PNG (Portable Network Graphics). Данный формат использует сжатие без потерь.
* Формат BMP (Bitmap Picture). Изображения формата BMP имеют расширения .bmp, .dib и .rle. В формате BMP изображения храниться как есть или же с применением некоторых распространённых алгоритмов сжатия.

В данном разделе был произведен анализ предметной области квалификационной работы, определены основные виды, применяемой в обработке в системах компании, информации, а также представление об форматах графической информации (изображений). Данный анализ необходим для определения, с какими видами информации в итоге, разработанная подсистема будет работать.

Проблематика предметной области

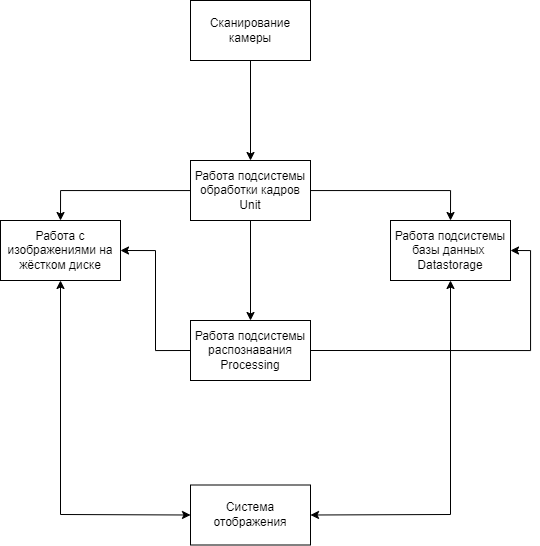
На сегодняшний момент процесс работы с данными в компании для распознавания объектов достаточно ресурсоемкий. В работе с различным видео и графической информацией, в компании возникает проблемы, связанные с избыточной информацией. Объем данной информации избыточно растет в процессе работы и все данные изображений с видеоустройств сохраняются на жесткий диск и базу данных. В свою очередь это приводит к излишним денежным затратам на покупку носителей информации, что сказывается на стоимость внедряемой системы.

С каждым снятым кадром с видеоустройства, система сохраняет большое количество несжатых данных изображений в память жесткого диска, при этом система не предусматривает гибкую работу с этими данными.

Система не предусматривает:

* сохранение в необходимом формате;
* cсжатие изображений для экономии диска;
* временное хранение или постоянное;
* удаление после истечения времени работы системы. После завершения работы системы информация об этих изображениях стирается, а изображения не удаляются с жесткого доска.

Для наглядной работы системы до автоматизации, стоит рассмотреть алгоритм, представленный на рисунке 4.



Алгоритм работы системы до автоматизации

До автоматизации работы с изображениями, процесс работы представляет из себя следующий алгоритм, где с видеоисточника «Сканирование камеры» поступает сигнал подсистеме обработке кадров «Unit», затем данная подсистема забирает изображение с камеры и передает на работу следующим подсистемам, таким как:

* Подсистема распознавания «Processing». В данном модуле на кадрах, с помощью нейронных сетей, распознаются необходимые для системы визуального контроля, элементы (отверстия на досках, шины для машин, расположение Data-Matrix).
* Подсистема базы данных «Datastorage». В модуле происходит работа, связанная с взаимодействием системы и базы данных. Сохраняются кадры или информация о них в подключенную базу данных.
* Модуль, работающий с жестким диском, сохраняет изображения на жесткий диск и не предусматривает больше никакую работы с ними.

После обработки и сохранения изображения в данных модулях, изображение попадает уже в среду визуализации данных «Система отображения», где конечный результат работы алгоритма, предоставляться пользователю системы.

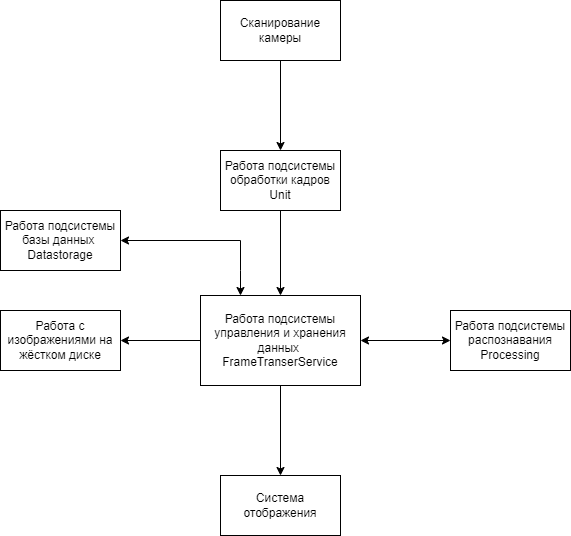
В ходе рассмотрения алгоритма, были обнаружены следующие недостатки системы:

* Много связей между модулей или подсистемами, не гибкая архитектура работы с передачей данных;
* Данные передаются всем подсистемам, что приводит к нагрузке на систему;
* Нет единого источника изображений, данные берутся как с жесткого диска, так и с базы данных, что приводит к долгому поиску в двух источниках данных о изображениях.

После данного анализа архитектуры и общей работы системы были приняты следующие решения:

* Оптимизировать и ускорить работу с данными;
* Сделать единое хранилище по работе с дынными изображения;
* Сделать независимым модулем от свей системы, для многократного использования.

На рисунке 5 приведен пример нового алгоритма системы, где появляется новая централизированная подсистема управления и хранения данных изображений «FrameTranserService».



Алгоритм работы системы после автоматизации

После автоматизации процессов управления и хранения данных изображений, алгоритм работы представляет из себя следующее, где после сканирование данных кадров и передачи в подсистему обработки кадров «Unit», далее данные изображений отправляются в подсистему управления и хранения данных «FrameTranserService», где сервис хранит у себя изображение. И при необходимости предоставляет изображения другим сервисам и подсистемам, таким как «Processing», «Datastorage» «Система отображения».

В данном разделе рассмотрены основные проблемы системы в работе с данными изображениями, рассмотрен алгоритм до автоматизации, где выделены его недостатки. И входе анализа был построен новый алгоритм работы системы, где устраняются недостатки старого алгоритма по работе с изображениями.

Описание пользователей и заинтересованных лиц

Заинтересованными лицами является как сама компания ООО «Малленом Системс», так и клиенты компании. Компания заинтересована в введении новых технологических решений в работе с данными изображений, так как компания в основном работает с данными видами информации.

В свою очередь клиенты заинтересованы в экономической части системы, где будет убрана необходимость закупать новые комплектующие для оптимальной работы системы в целом. Повышается удобство и скорость работы с данными, что так же значительно повлияет по производственные процессы и собственно, на прибыль клиента.

В данном разделе были выделены основные заинтересованные лица, для которых необходима данная разработка.

Постановка задачи

В связи с поставленными проблемами управления и хранения данных, компания ООО «Малленом Системс» решила разработать такую систему, которая позволит автоматизировать процессы взаимодействия данных о изображениях внутри системы, путем использования современных технологий для создания сервиса.

Требования к разрабатываемой автоматизированной системе регламентируется в составленном техническом задании, согласно ГОСТ 34.602 «Техническое задание на создание автоматизированной системы» в Приложении А.

Для осуществления был выбран вариант создать отельный Web-Api сервис по удаленной работе с данными, путем HTTP запросов. Данный вариант реализации упростит работу с изображениями и отделит данные процессы в отдаленный сервис.

Для создания сервиса требуется:

* провести проектирование архитектуры, включая проектирования структур данных и пользовательского интерфейса согласно требованиям;
* разработать веб-сервис управления и хранения данных;
* составить оптимальные тестовые сценарии.

В данном разделе была произведена постановка задачи на разрабатываемую систему, составлено техническое задание, по которому будет проектироваться и создаваться конечный продукт.

Теоретические сведения

Веб-сервис

Веб-сервис — это сетевая технология, обеспечивающая межпрограммное взаимодействие на основе веб-стандартов. []

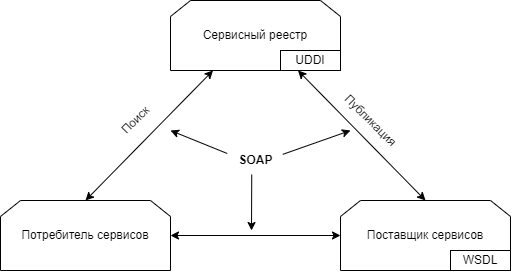
Веб-сервисы независимы от языка и платформы реализации. Они способны взаимодействовать друг с другом, а также с другими приложениями, обмениваясь сообщениями с помощью сетевых протоколов.

Стандартным протоколом для обмена информацией является SOAP, но также существуют и другие. Они будут рассмотрены в следующих пунктах.

Для описания интерфейсов веб-сервиса используется специальный язык описания - WSDL (Web Services Description Language).

Также веб-сервис содержит универсальный интерфейс распознавания, описания и интеграции - UDDI (Universal Discovery, Description and Integration), который используется для поиска существующих веб-сервисов и обеспечения доступа к ним.

Концепция веб-сервиса представлена на рисунке 6.



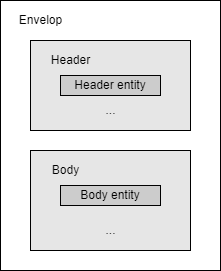
Концепция веб-сервиса

Протокол SOAP

SOAP (Simple Object Access Protocol) - протокол доступа к объектам ((компонентом распределенной вычислительной системы), предназначенный для обмена структурированной информацией.

Проще говоря, SOAP — это протокол для взаимодействия с веб-сервисами, созданный на основе XML - расширяемом языке разметки. Можно сказать, что SOAP — это соглашение о строго форматированном XML-документе. Согласно ему, XML-документ должен содержать определенные элементы и пространства имён, а также специальные теги.

Сообщения SOAP оформляются в виде особой структуры, представленный на рисунке 7, которая называется конверт (Envelope). Она включает в себя заголовок (Header) - необязательный элемент и тело (Body).

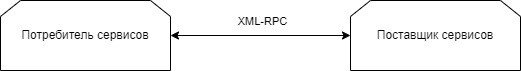


Структура SOAP-сообщений

Протокол XML-RPC

XML-RPC (Extensible Markup Language Remote Procedure Call) – протокол для вызова удаленных процедур. Он очень прост и эффективен использовании, однако, в отличие от SOAP, не предназначен для решения глобальных задач. Но несмотря на это, данный протокол используется достаточно широко во многих веб-разработках.

На рисунке 8 представлена концепция веб-сервера, использующего протокол XML-RPC.



Концепция XML-RPC

REST-архитектура

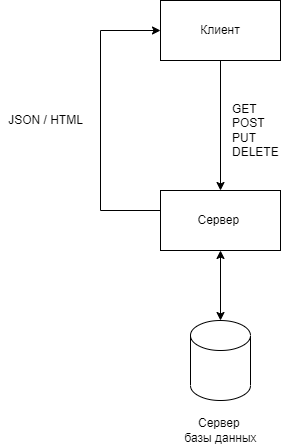
В отличие от SOAP и XML-RPC, REST (Representational State Transfer) нельзя назвать протоколом. Это стиль архитектуры взаимодействия компонентов в распределенной сети.

REST базируется на HTTP протоколе, и следовательно, может использовать все существующие наработки на базе HTTP.

Веб-сервис можно назвать RESTful сервисом, если он не нарушает обязательных требований к REST-архитектуре:

* архитектура сервиса должна быть приведена к модели клиент-сервер;
* на сервере не может хранится никакая информация о состоянии клиента, в период между его запросами;
* клиенты могут кэшировать (сохранять) ответы от сервера. Поэтому ответы сервера должны обозначаться как кэшируемые или некэшируемые, для предотвращения получения устаревших данных;
* интерфейсы REST-сервисов должны быть единообразны [].

На рисунке 9 представлена модель RESTful сервиса.



RESTful сервис

Веб-сервер

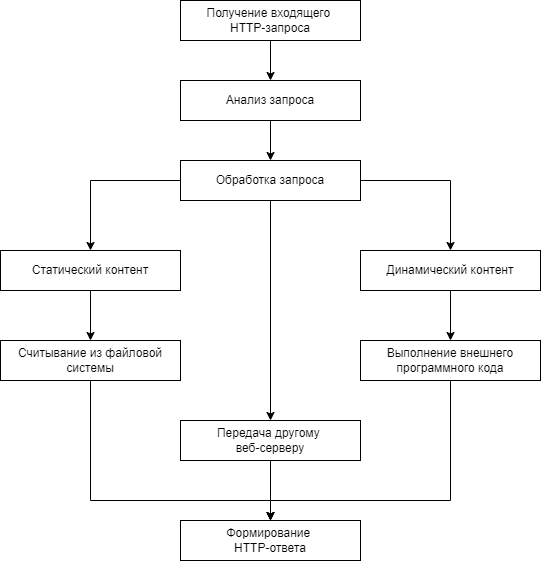
Веб-сервер - это сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы [].

Основные функции веб-сервера:

* управление соединением;
* прием и обработка запроса;
* разделение доступа к нескольким обслуживаемым наборам ресурсов;
* отдача статического содержимого;
* взаимодействие с приложениями для получения и дальнейшей отдачи клиенту динамического содержимого [].

При запуске веб-сервера ему присваивается номер порта – сетевой идентификатор. При этом говорят, что сервер слушает порт. На этот порт клиент посылает HTTP-запросы.

Алгоритм работы веб-сервера представлен на рисунке 10.



Алгоритм работы Веб-сервера

Работа данного алгоритма заключается в следующем, что на сервер поступает запрос от клиента, далее запрос анализируется: происходит аутентификация, обрабатывается адрес, проверяются виртуальные хосты. Затем происходит обработка запроса: если запрашивается статический контент, он считывается из файловой системы; если запрашивается динамический контент, то сервер обращается к программному коду и генерирует новые данные; сервер также может передавать запрос для обработки другому серверу. В конце происходит формирование HTTP-ответа и отправка его клиенту.

В данном разделе разобраны основные теоретические сведенья о используемых технологиях веб-сервера, протоколы передачи, архитектуры построения веб-сервиса, а также работа веб-сервера. Эта информация необходима, для общего понимания работы, разрабатываемой подсистемы управления и хранения данных.

Анализ существующих аналогов построения веб-сервиса

Apache

Apache — веб-сервер с открытым исходным кодом. Это значит, что пользователь может редактировать и адаптировать платформу под свои нужды. Apache работает на всех популярных операционных системах. Основным достоинством данного веб-сервера является его гибкость. Он включает в себя обширный функционал, при этом его можно расширить, добавив дополнительные модули.

Данный веб-сервер надежный и достаточно мощный веб-сервер, однако имеет и недостатки. Функционала настолько много, что возможно, он даже избыточен. В основном пользователь задействует всего лишь 10% от этих возможностей. Также, с точки зрения архитектуры, Арache работает по модели «процессов». Это означает, что для каждого соединения Аpасhe выделяет отдельный поток данных, что вызывает большую нагрузку [].

IIS

IIS (Internet Information Services) — веб-сервер от компании Microsoft, и поэтому, данная платформа работает только с операционными системами Windows. Но несмотря на это, IIS является вторым по полярности на рынке, после Apache. Он обладает достаточно широким функционалом, надёжен, и безопасен, так как его поддержкой официально занимается Microsoft. Однако он уступает в производительности своему конкуренту.

Nginx

Nginx — достаточно популярный веб-сервер, который часто используют в качестве обратного прокси-сервера. Он способен обрабатывать большое количество соединений, при этом расходуя минимум системных ресурсов. Nginx можно назвать одним из самых производительных веб-серверов, он лучше всех отдаёт статические данные среди популярных серверов. Однако он не имеет возможности самостоятельно обрабатывать запросы к динамическому контенту.

Node.js

Node.js — платформа для разработки распределенных сетевых приложений, таких как веб-сервер. Данная платформа предназначена для использования на всех типах операционных систем; она достаточно проста, доступна и стабильна.

Node.js основан на движке V8, а он является одним из самых производительных для JavaScript на данный момент. За счёт использования данного движка, код выполняется намного быстрее. Кроме того, еще одно преимущество V8 — это эффективное управление памятью.

Также для Node.js существует очень много библиотек и инструментов, которые активно развиваются. Однако это можно отнести и к недостаткам – так как нет каких-либо основных библиотек или инструментов - у каждого есть множество альтернатив, свои плюсы и минусы, которые не всегда можно понять из документации. Как итог - не просто сделать выбор в сторону той или иной библиотеки, и не всегда этот выбор оказывается правильным.

Node.js — относительная молодая платформа, поэтому для многих стандартных задач нет какого-то готового и законченного решения.

ASP.NET Core

Платформа ASP.NET Core — разработана корпорацией Microsoft и предназначена для разработки различных веб-приложений.

Основным преимуществом данной технологии является её кроссплатформенность. Так как она основана на .NET Core, то приложение, разработанное на этой платформе, может быть развёрнуто не только на операционных системах семейства Windows, но и на всех других популярных операционных системах.

ASP.NET Core состоит из нескольких компонентов, благодаря чему можно легко расширять базовый функционал фреймворка.

Главный недостаток данной платформы состоит в том, что она была выпущена не так давно. Она активно развивается, и дополняется в настоящее время. Поэтому многие привычные фреймворки и пакеты еще не умеют работать c ASP.NET Core.

Kestrel

Kestrel — это кроссплатформенный веб-сервер, который обычно используется с приложениями на основе ASP.Net core. Его можно использовать как отдельный веб-сервер, так и с обратным прокси-сервером, таким как IIS, Apache или Nginx. Использование обратного прокси-сервера не обязательно, но всё же желательно. Так как он получает НТТР запросы из сети и направляет их в Kestrel после первоначальной обработки. Это позволяет облегчить распределение нагрузки, обеспечить безопасность и поддержку SSL.

В данном разделе были рассмотрены популярные аналоги построения веб-сервера, рассмотрены как преимущества той или иной технологии, так и недостатки. Анализ позволит выделить более удобный и удачный выбор веб-сервера для разработки системы управления и хранения данных по распознаванию объектов.

Анализ средств для разработки системы

Для разработки веб-сервера согласно техническому заданию, было решено использовать платформу ASP.NET Core, сервер Kestrel, и язык программирования C#.

ASP.NET Core появилась относительно недавно, однако уже успела стать достаточно популярной в IT-сообществах. Одним из главных достоинств платформы является её кроссплатформенность, а это значит, что приложение может быть развёрнуто на всех популярных операционных системах. Также преимуществом ASP.NET является встроенный IOC-контейнер.

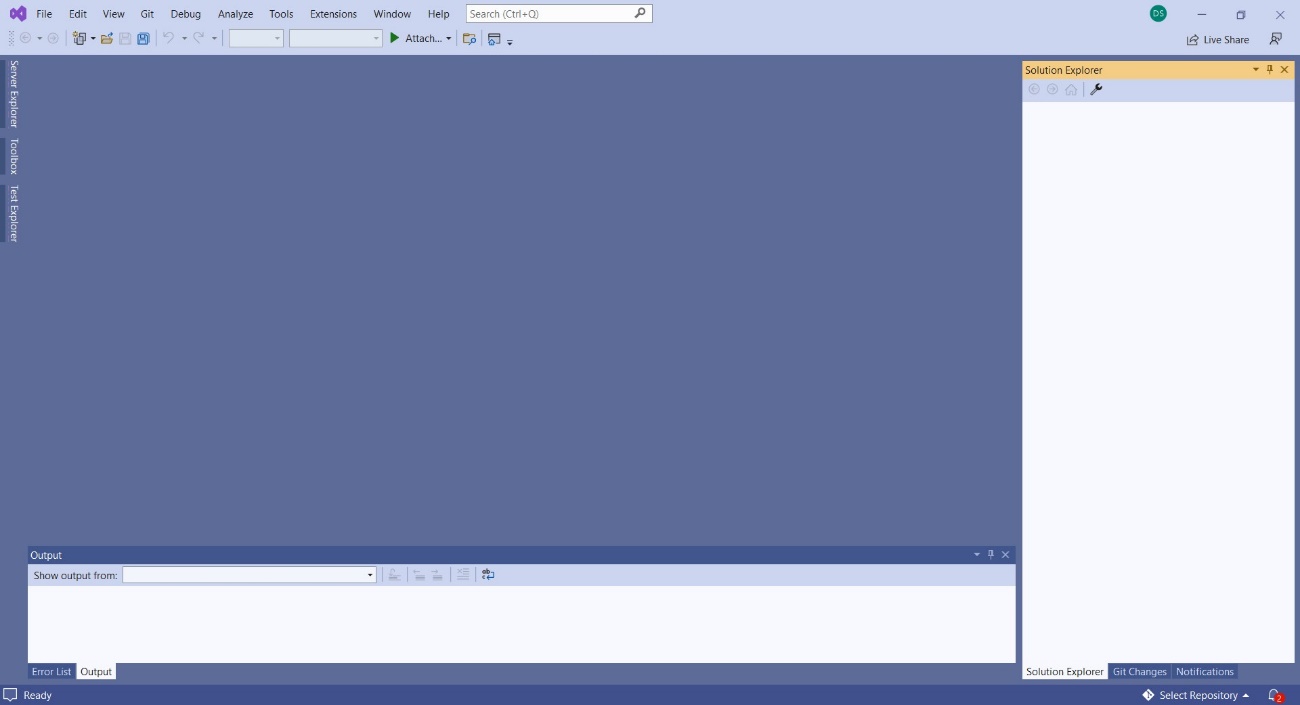
IOC-контейнер — это фреймворк, которая позволяющая упростить и автоматизировать написание кода, позволяющий реализовывать внедрение зависимостей более лаконичным способом, избавляя от рутины.

В качестве IDE была выбрана Visual Studio 2022 Community от Microsoft.

Visual Studio 2022 — более быстрая, более производительная и упрощенная версия, предназначенная для учащихся, а также пользователей, которые создают решения промышленного масштаба [].

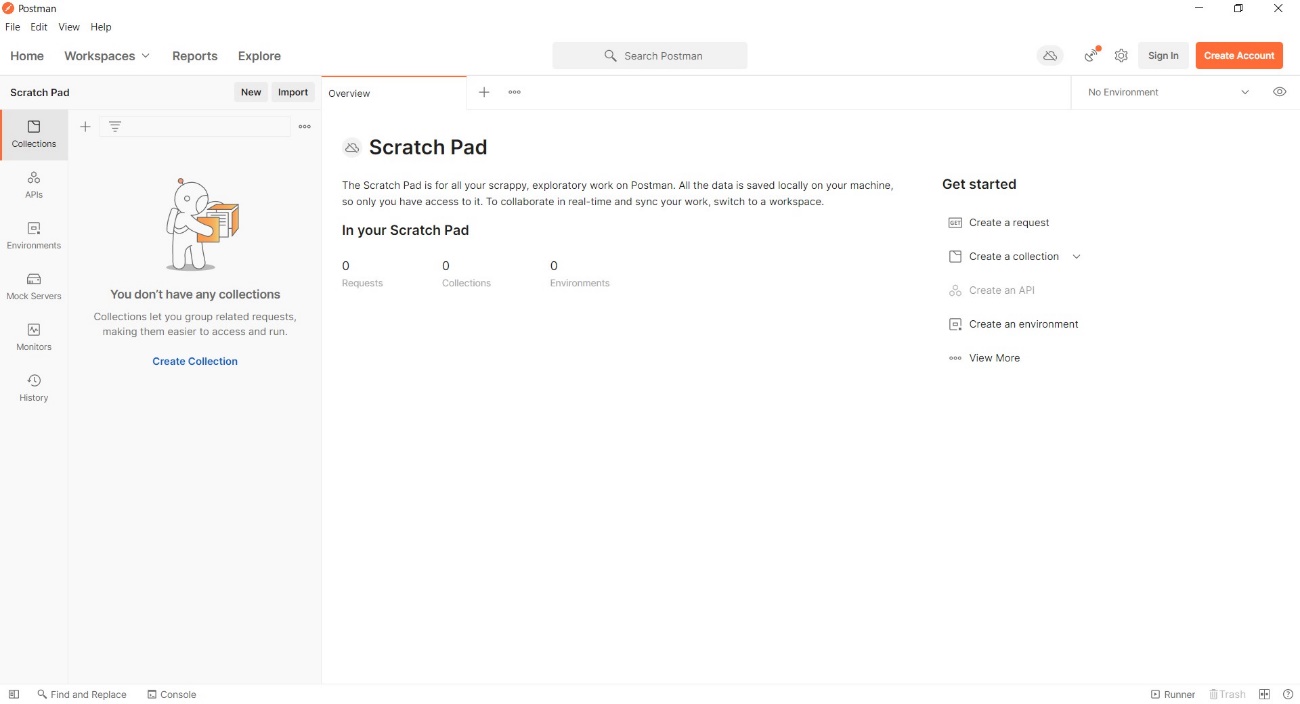
Так как на сегодняшний день она предлагает самый широкий спектр функциональных возможностей для разработки на языке С#. Имеет множество устанавливаемых модулей (расширений), облегчающие написание программного кода, путем подцветки синтаксиса, ошибок и предложений по улучшению написанного кода.

Графический интерфейс среды разработки Visual Studio 2022 Community представлена на рисунке 12.



Cредf разработки Visual Studio 2022 Community

Для разработки сервиса, была использована программа Postman API Platform, изображен на рисунке 14.

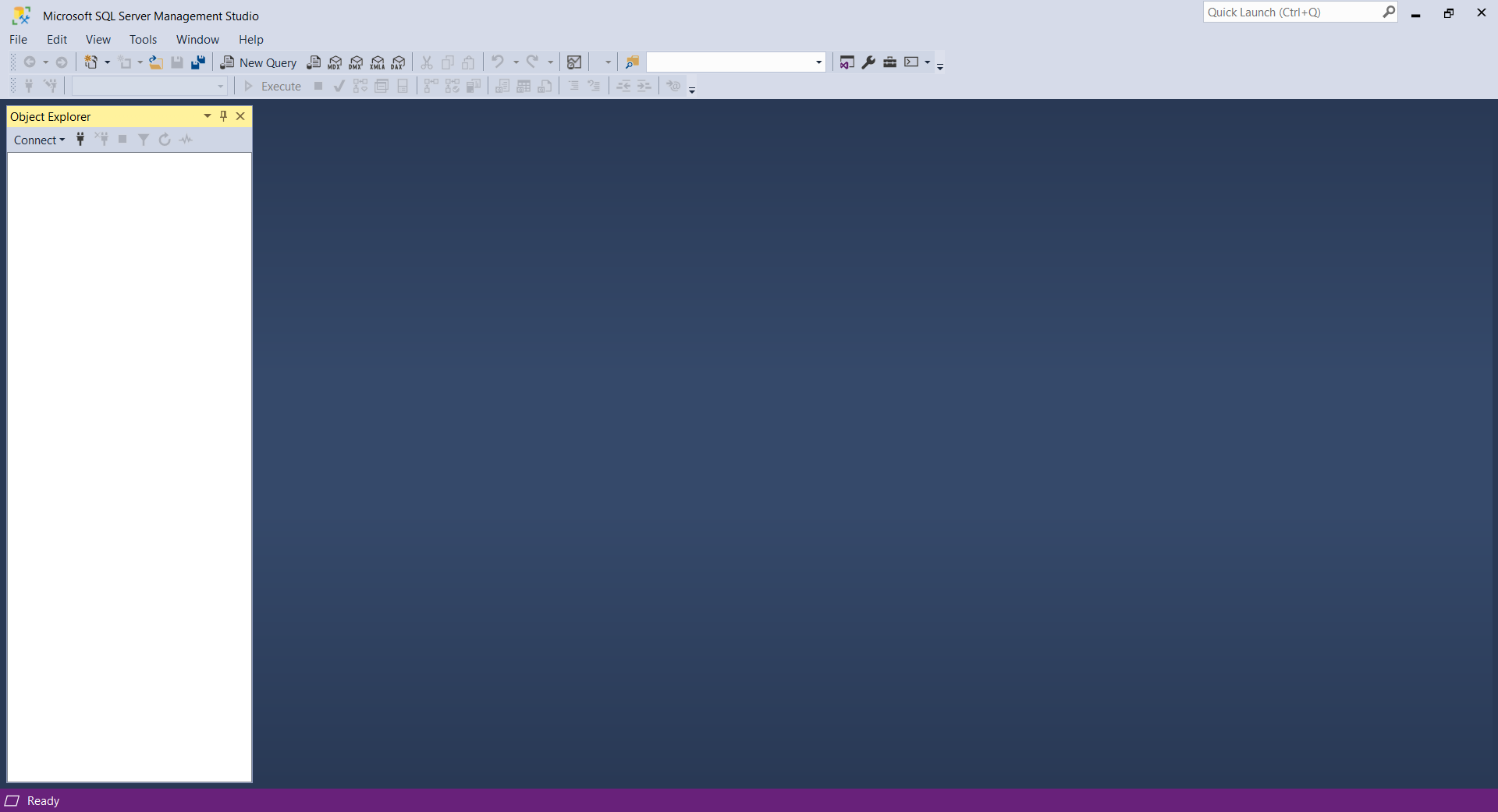


Программа Postman API Platform

Postman — это классическое приложение, способное выполнять запросы API к любому API HTTP. Он используется для тестирования и изучения API-интерфейсов.

В качестве СУБД была выбрана MS SQL. Она хорошо работает в связке с выбранной платформой и позволяет полностью удовлетворить требования к построению структуры базы данных для приложения.

Для работы с базой данных используется программное обеспечение SQL Server Management Studio (SSMS), представлена на рисунке 15.



SQL Server Management Studio

SSMS — это интегрированная среда для управления любой инфраструктурой SQL. Используется для доступа, настройки, администрирования и разработки всех компонентов SQL Server. Среда SSMS предоставляет единую комплексную служебную программу, которая сочетает в себе обширную группу графических инструментов с рядом многофункциональных редакторов скриптов для доступа к SQL Server для разработчиков и администраторов баз данных всех профессиональных уровней[].

Для обработки информации в базах данных будет использоваться фреймворк — Entity framework (EF). EF позволяет взаимодействовать с базой данных за счёт функций языка C# и сущностей, а не таблиц. В обычном случае взаимодействие происходит под действием прямых SQL запросов к базе данных. EF заменяет часть с написанием SQL запросов на использование сущностей – классов с их функционалом.

Работа с данными LINQ — набор технологий, создающих и использующих возможности интеграции запросов непосредственно в язык C#. Технологии LINQ превращают запросы в удобную языковую конструкцию, которая применяется аналогично классам, методам и событиям[].

Тестирование системы происходит двумя способами, используеться модульное (компонентное) тестирование, для него используется специальная среда тестирования NUnit, также используется метод покрытия, где составляются определенные сценарии тестирования, и фиксируется конечный результат (успешно или не успешно).

В качестве системы контроля версии продукта был выбран Git — распределенная система управления версиями. Полнофункциональные локальные репозитории упрощают работу в автономном режиме или в удаленном расположении. При создании системы необходимо фиксировать свою работу локально, а затем синхронизировать свою копию репозитория с копией на сервере.

В качестве удаленного сервера, согласно техническому заданию, используется собственный сервер GitLab.

GitLab — веб-инструмент жизненного цикла DevOps с открытым исходным кодом, представляющий систему управления репозиториями кода для Git с собственной вики, системой отслеживания ошибок, CI/CD пайплайном и другими функциями [].

Инструмент для работы с репозиторием используется Gitter, представлен на рисунке 16.



Графический интерфейс Gitter

Программное обеспечение Gitter — визуальный клиент системы управления версиями Git, разработанное компанией Mallenom Systems.

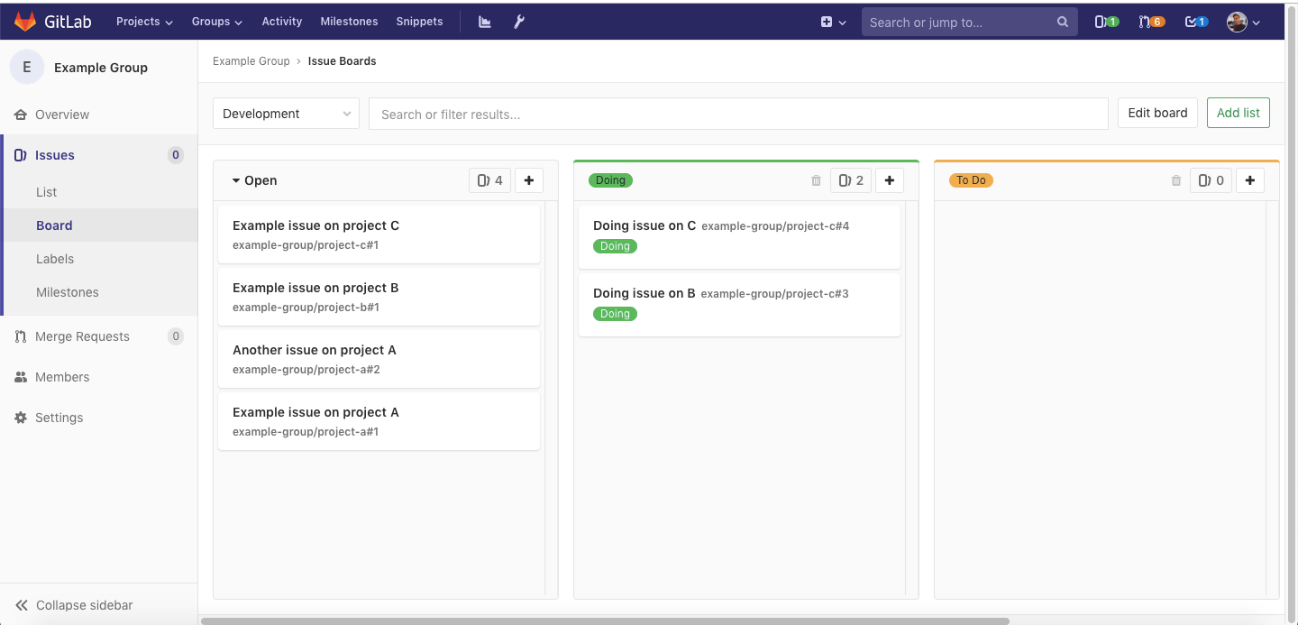
В данном разделе рассматриваются основные средства для разработки подсистемы управления и хранения данных изображений «FrameTranserService». Основным языком программирования выступает C#, среда разработки Visual Studio Community, система управления базами данных Microsoft Sql Server, система контроля версий Git.

Анализ методологии разработки системы

В компании Mallenom Systems при разработке программного обеспечения используют гибкую методологию «Agile».

Методика Аgile — это итеративный подход к управлению проектами и разработке программного обеспечения, позволяющий разработчикам ускорить разработку программных продуктов. Преимуществом является в том, что нет необходимости выпускать весь продукт целиком, разработка выполняется в рамках небольших, но удобных инкрементов. Требования, планы и результаты постоянно проходят проверку на актуальность, благодаря чему разработчики могут быстро реагировать на изменения.

Для реализации принципов Аgile компания использует популярный подход Kanban. Инструмент для работы с данной методологией используется сервис Gitlab, графический интерфейс доски представлен на рисунке 17.



Графический интерфейс сервиса Gitlab

Предполагает обсуждение производительности в режиме реального времени и полную прозрачность рабочих процессов. Рабочие задачи визуально представлены на доске Kanban, что позволяет разработчикам видеть состояние каждой задачи в любой момент времени.

На каждом этапе создания системы, создаются определенные задачи, которые в свою очередь могут разбиваться на подзадачи. Задачи ставят в основном руководители проектов, а разбивают на подзадачи сами разработчики, для удобного распределения нагрузки в период разработки программного обеспечения.

В данном разделе рассмотрена, используемой компанией, методология «Agile», выделены ее главные преимущества. Зная методологию разработки, позволит ускорить и повысить качество разрабатываемой системы.