бюджетное профессиональное образовательное учреждение Вологодской области

«Череповецкий лесомеханический техникум им. В.П. Чкалова»

Специальность **09.02.07** «Информационные системы и программирование»

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

**ПП по ПМ.05 Проектирование и разработка информационных систем**

Выполнил студент 3 курса группы ИС-31

Богданова Варвара Юрьевна

подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

место практики ООО «Малленом Системс»

Период прохождения:

с «08» 06 2025 г.

по «21» 06 2025 г.

Руководитель практики от

предприятия, должность:

Южакова Н.В., специалист по кадрам

подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

МПРуководитель практики от

техникума: Материкова А.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 года

г. Череповец

2025

Содержание

[ВВЕДЕНИЕ 3](#_Toc200904158)

[1 ОБЩАЯ ХАРАКТЕРИСТИКА КОМПАНИИ 4](#_Toc200904159)

[1.1 Общая информация (сфера деятельности) 4](#_Toc200904160)

[1.2 Роль информационных систем в работе организации 4](#_Toc200904161)

[1.3 Основные используемые технологии 5](#_Toc200904162)

[2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ 6](#_Toc200904163)

[2.1 Анализ требований и моделирование информационных систем 6](#_Toc200904164)

[2.2 Проектирование информационных систем 6](#_Toc200904165)

[2.3 Разработка информационных систем 6](#_Toc200904166)

[2.4 Тестирование информационных систем 7](#_Toc200904167)

[2.5 Внедрение, эксплуатация и сопровождение информационных систем 7](#_Toc200904168)

[3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ 8](#_Toc200904169)

[ЗАКЛЮЧЕНИЕ 17](#_Toc200904170)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#_Toc200904171)

[ПРИЛОЖЕНИЕ А 19](#_Toc200904172)

[ПРИЛОЖЕНИЕ Б 29](#_Toc200904173)

# ВВЕДЕНИЕ

# 

В современном мире информационные системы (ИС) играют ключевую роль в функционировании и развитии предприятий различного масштаба и отраслевой принадлежности. Эффективное проектирование и разработка ИС позволяет автоматизировать бизнес-процессы, оптимизировать управление ресурсами, повысить качество принимаемых решений и, в конечном итоге, укрепить конкурентные позиции компании. Место прохождения практики ООО «Малленом Системс». Сроки прохождения с 08.06.2025 по 21.06.2025

Цель: демонстрация приобретенных практических навыков и знаний в области проектирования, разработки, внедрения и сопровождения информационных систем, а также анализ полученного опыта и определение направлений для дальнейшего профессионального развития.

Задачи:

1. Провести анализ требований и моделирование информационных систем;
2. Выполнить проектирование информационных систем;
3. Произвести разработку информационных систем;
4. Провести тестирование информационных систем;
5. Осуществить внедрение, эксплуатацию и сопровождение информационных систем.

# ОБЩАЯ ХАРАКТЕРИСТИКА КОМПАНИИ

## Общая информация (сфера деятельности)

Малленом Системс - ведущая российская компания, специализирующаяся на разработке и внедрении систем компьютерного зрения, промышленной видеоаналитики и интеллектуальной обработки данных. Компания создает решения на базе технологий машинного зрения и искусственного интеллекта (машинное обучение, нейронные сети глубокого обучения).

Компания ориентирована на наукоемкие ИТ-проекты для повышения эффективности и безопасности промышленных предприятий в различных отраслях (транспорт, машиностроение, нефтегазовая, металлургическая, пищевая, фармацевтическая, алмазодобывающая, атомная и другие).

## Роль информационных систем в работе организации

Информационные системы играют ключевую роль в следующих аспектах работы Малленом Системс:

* Автоматизация: Автоматизация производственных процессов и промышленных предприятий.
* Видеоконтроль: Обеспечение видеоконтроля в различных областях, таких как транспорт и промышленность.
* Анализ данных: Интеллектуальная обработка данных для поиска закономерностей и прогнозирования.
* Безопасность: Повышение безопасности на промышленных предприятиях.
* Эффективность: Повышение эффективности технологических процессов и производства.
* Принятие решений: Предоставление информации для принятия обоснованных решений.
* Прослеживаемость: Обеспечение прослеживаемости продукции на различных уровнях (L1-L2).

## Основные используемые технологии

* Языки программирования: C++, Python (для машинного обучения и анализа данных), Java или C# (для разработки клиент-серверных приложений).
* Фреймворки: TensorFlow, PyTorch (для машинного обучения), OpenCV (для компьютерного зрения), .NET или Java frameworks (для разработки приложений).
* СУБД: PostgreSQL, MySQL, MS SQL Server (для хранения и управления данными).
* Системы контроля версий: Git (с использованием платформ, таких как GitHub, GitLab или Bitbucket).
* Инструменты проектирования: UML-диаграммы (для проектирования архитектуры систем), инструменты моделирования данных.
* Методологии разработки: Agile (Scrum, Kanban) - для гибкой и итеративной разработки, возможно, с элементами Waterfall для более крупных проектов.
* Системы управления задачами: Jira, Trello, Asana (для управления проектами и задачами).
* Кроме того, в компании используются собственные решения на базе нейронных сетей и детерминированных алгоритмов анализа изображений, а также алгоритмы от Cognex и Hikrobot.

# ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ

## Анализ требований и моделирование информационных систем

**Анализ требований** - часть процесса разработки программного обеспечения, которая включает сбор требований от заинтересованных сторон, выявление взаимосвязей и документирование. **Моделирование информационных систем**- процесс создания упрощённого представления реальной системы, процесса или явления с целью анализа, понимания, прогнозирования или автоматизации**. Анализ требований и моделирование взаимосвязаны**, так как моделирование помогает структурировать и визуализировать требования, а анализ требований, в свою очередь, определяет, какие функции и характеристики должна выполнять система.

## Проектирование информационных систем

Проектирование информационных систем (ИС) - процесс создания проекта-прототипа системы, который включает последовательное формирование и изменение согласованных моделей на всех этапах жизненного цикла ИС.

Цель проектирования - обеспечить требуемую функциональность системы, её адаптивность к изменяющимся условиям функционирования, пропускную способность, время реакции на запросы и другие характеристики.

## Разработка информационных систем

**Разработка информационных систем** - это процесс создания комплексного решения, которое включает программное обеспечение, базы данных, интерфейсы и другие компоненты для автоматизации и оптимизации бизнес-процессов заказчика.

## Тестирование информационных систем

Тестирование информационных систем - важный этап производства программного обеспечения, направленный на детальное исследование программного кода и выявление ошибок в работе систем.

Одна из главных целей тестирования - проверка соответствия работоспособности системы в целом или её отдельных модулей ожиданиям заказчика.

## Внедрение, эксплуатация и сопровождение информационных систем

**Внедрение информационных систем** — это процесс, который включает установку новых или обновление существующих систем в организации. Цель внедрения - повышение эффективности бизнес-процессов, улучшение качества принимаемых решений и оптимизация работы компании в целом.

Эксплуатация информационных систем включает в себя исправления в работе всех частей системы при возникновении сбоев, регистрацию этих случаев в журналах и накопление статистики о качестве работы всех компонентов системы.

Сопровождение информационных систем — этап, к которому выполняется переход уже после полного её внедрения. В него входят:

* Оперативное устранение возникающих проблем;
* Обеспечение защиты корпоративной информации;
* Добавление новых функций после изменения бизнес-процессов предприятия.

# ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

Задание 1:

1. Провести анализ систем управления проектами, основанные на методологии канбан.

Таблица 1 – Сравнительный анализ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Функция | Trello | Jira | Asana | Monday.com | Projecto |
| Канбан-доска | Да | Да | Да | Да | Да |
| Ограничение WIP | Да | Да | Да | Да | Да |
| Диаграмма Ганта | Нет | Да | Да | Да | Да |
| Управление ресурсами | Нет | Да | Да | Да | Да |
| Управление финансами | Нет | Нет | Нет | Да | Да |
| Отчетность | Ограниченная | Расширенная | Расширенная | Расширенная | Расширенная |
| Интеграции | Да | Да | Да | Да | Да |

1. Выбрать подходящую систему управления проектами.

Я выбрала систему управления Projecto. Ниже описаны несколько ключевых причин, почему стоит выбрать именно её:

* Единая платформа: Projecto объединяет Канбан, диаграммы Ганта, управление ресурсами, финансами, совместную работу и аналитику в одной платформе, что позволяет командам управлять проектами более эффективно.
* Комплексное управление ресурсами: Projecto позволяет эффективно распределять ресурсы между задачами, отслеживать их загрузку и оптимизировать использование, что особенно важно для команд с ограниченными ресурсами.
* Гибкость и настраиваемость: Projecto предлагает широкие возможности для настройки рабочих процессов, полей и отчетов, что позволяет адаптировать систему к потребностям конкретной команды и проекта.
* Отличная поддержка: Projecto предлагает качественную поддержку клиентов, включая документацию, обучающие материалы и отзывчивую команду поддержки

1. После выбора систем управления проектами, необходимо реализовать репозиторий с проектом будущей разрабатываемой подсистемы.

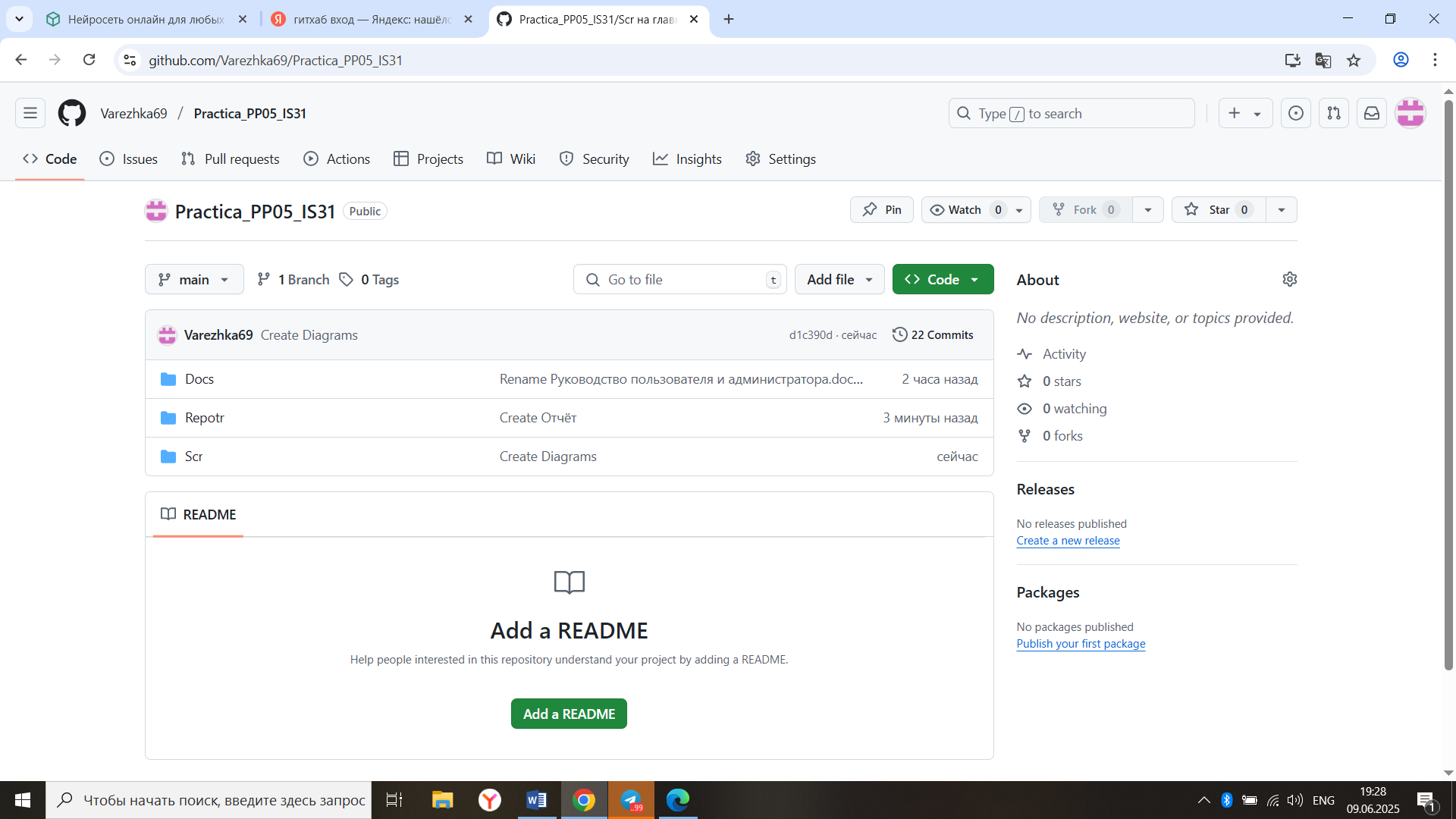


Рисунок 1 – Репозиторий

Задание 2:

* 1. Необходимо определится с направлением разработки.
  2. Для выбранной предметной области составить план работ, составить задачи и ввести их на доску канбан. Необходимо детально рассчитать сроки и количество задач, до окончания практики.

Я выбрала тему «Подсистема хранения данных для системы распознавания дефектов на текстиле». После этого составила задачи и вывесила их на доску канбан.

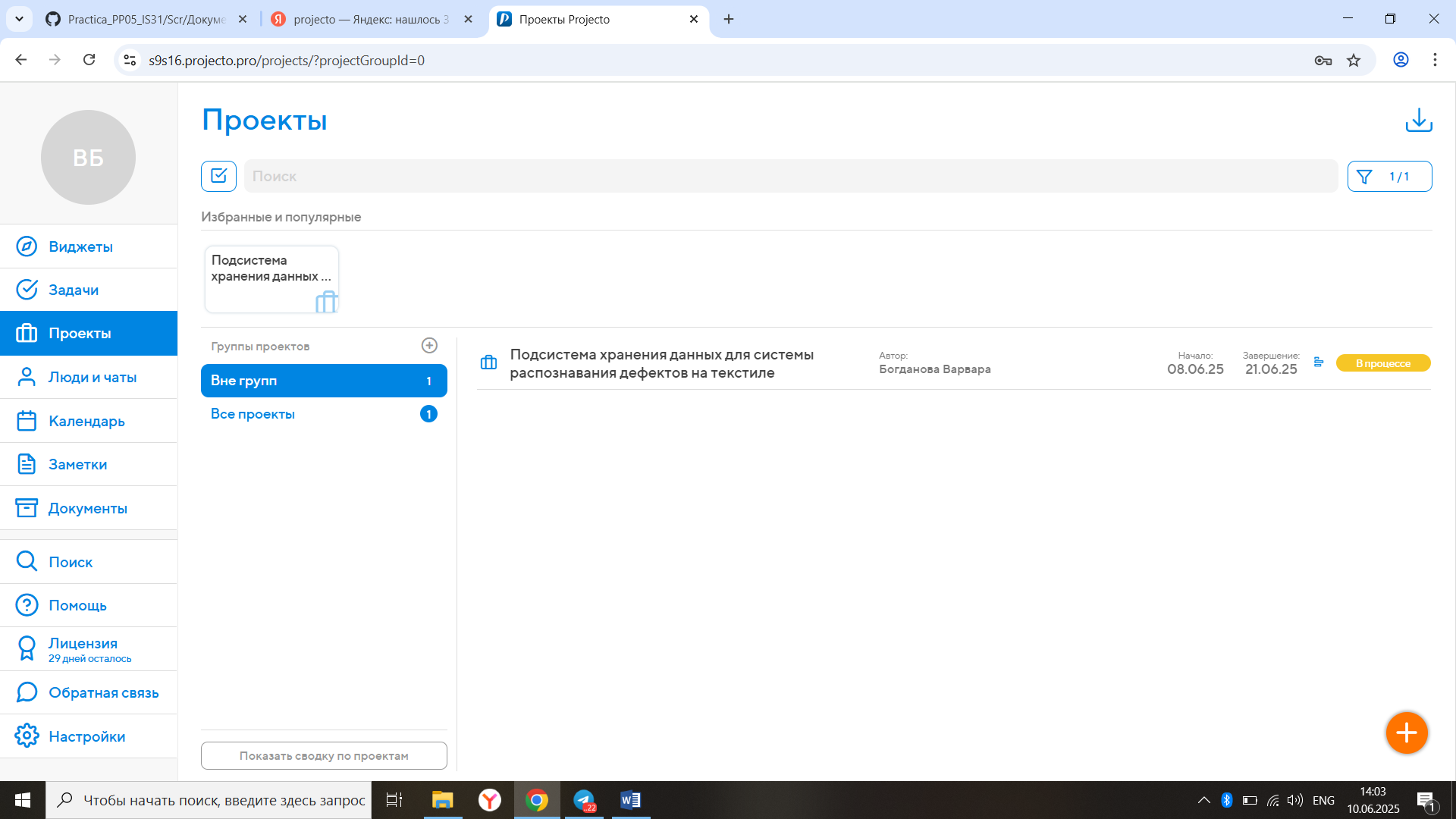


Рисунок 2 – Проект в Projecto

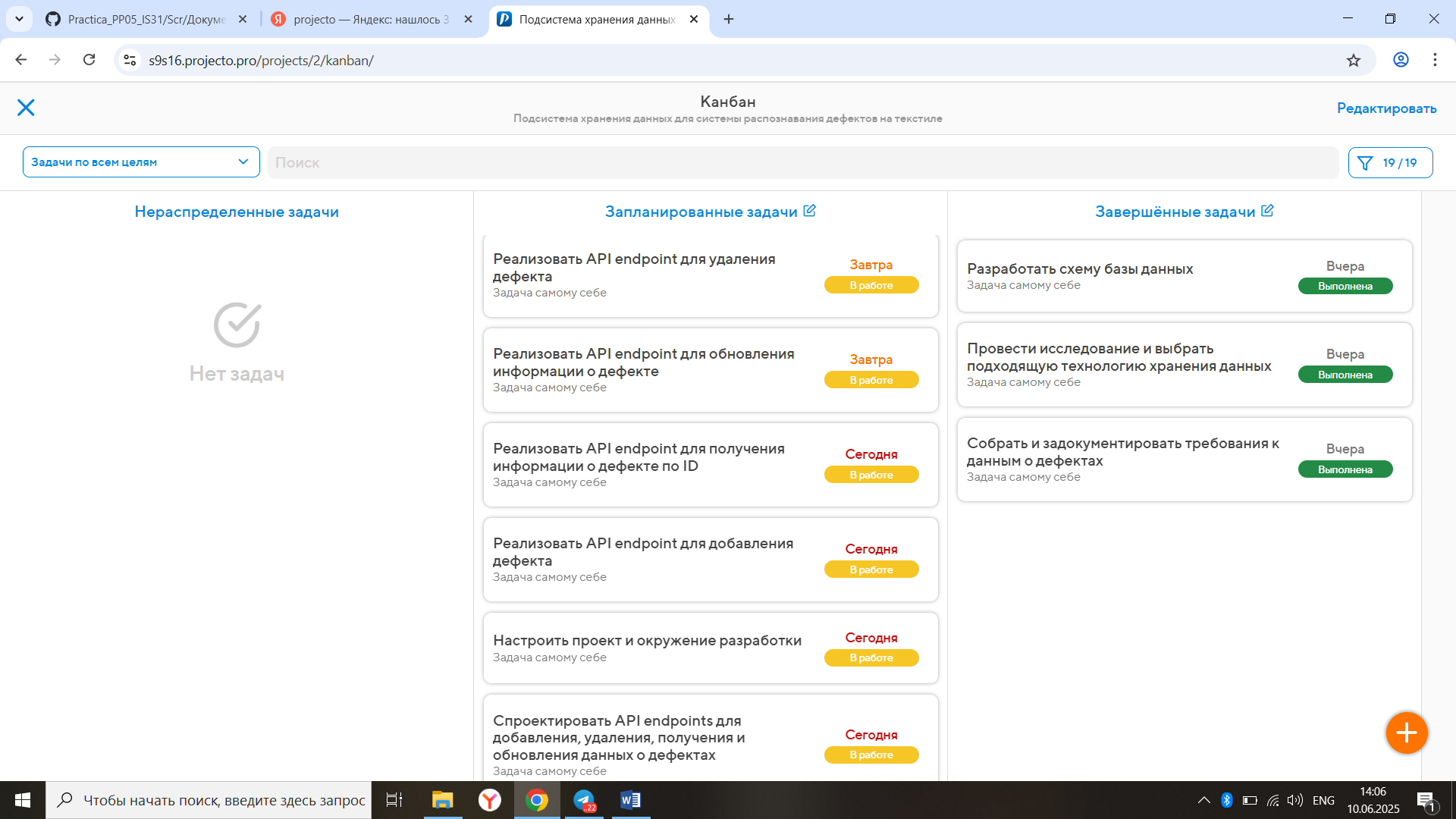


Рисунок 3 – Задачи в Projecto

* 1. Составить документацию для выбранной подсистемы:
     1. Техническое задание (Приложение А).
     2. Руководство пользователя и администратора (Приложение Б).
     3. ER диаграмма и её описание.

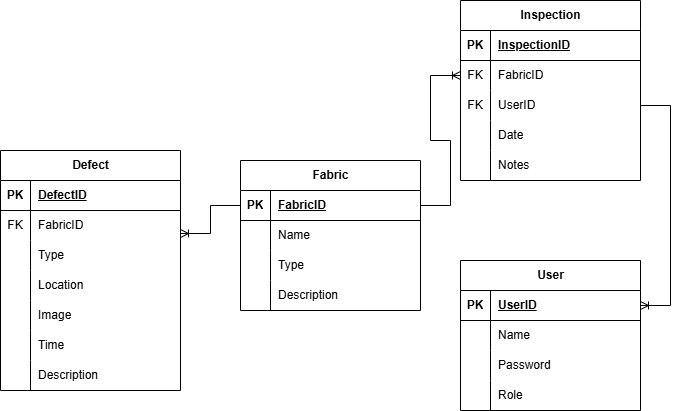


Рисунок 4 – ER диаграмма

Таблица 2 –Описание таблицы Defect

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Поле | Тип данных | Обязательный | Замечание |
| Уникальный идентификатор дефекта | DefectID | Число | Да | Автоматически генерируется |
| Уникальный идентификатор ткани | FabricID | Число | Да | Внешний ключ |
| Тип дефекта | Type | Текст | Да | Ограничение в 225 символов |
| Местоположение дефекта | Location | Текст | Да | Ограничение в 225 символов |
| Путь к файлу изображения дефекта | Image | Текст | Да | - |
| Время обнаружения дефекта | Time | Число | Да | - |
| Описание дефекта | Description | Текст | Нет | Ограничение в 225 символов |

Таблица 3 – Описание таблицы Fabric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Поле | Тип данных | Обязательный | Замечание |
| Уникальный идентификатор ткани | FabricID | Число | Да | Автоматически генерируется |
| Название ткани | Name | Текст | Да | Ограничение в 225 символов |
| Тип ткани | Type | Текст | Да | Ограничение в 225 символов |
| Описание ткани | Description | Текст | Нет | Ограничение в 225 символов |

Таблица 4 – Описание таблицы User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Поле | Тип данных | Обязательный | Замечание |
| Уникальный идентификатор пользователя | UserID | Число | Да | Автоматически генерируется |
| Имя пользователя для входа в систему | Name | Текст | Да | Ограничение в 225 символов |
| Пароль пользователя | Password | Текст | Да | Ограничение в 225 символов |
| Роль пользователя в системе | Role | Текст | Да | Ограничение в 225 символов |

Таблица 5 – Описание таблицы Inspection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Поле | Тип данных | Обязательный | Замечание |
| Уникальный идентификатор проверки | InspectionID | Число | Да | Автоматически генерируется |
| Уникальный идентификатор ткани | FabricID | Число | Да | Внешний ключ |
| Уникальный идентификатор пользователя | UserID | Число | Да | Внешний ключ |
| Дата проведения проверки | Date | Число |  | - |
| Заметки | Notes | Текст | Нет | Ограничение в 225 символов |

* + 1. Диаграмма вариантов использования

Диаграмма вариантов использования демонстрирует взаимодействие между пользователями и подсистемой хранения данных.



Рисунок 5 – Диаграмма вариантов использования

* + 1. Диаграмма последовательностей

Диаграмма последовательности детализирует взаимодействие между компонентами системы при добавлении дефекта, демонстрируя порядок вызовов и передачу данных.

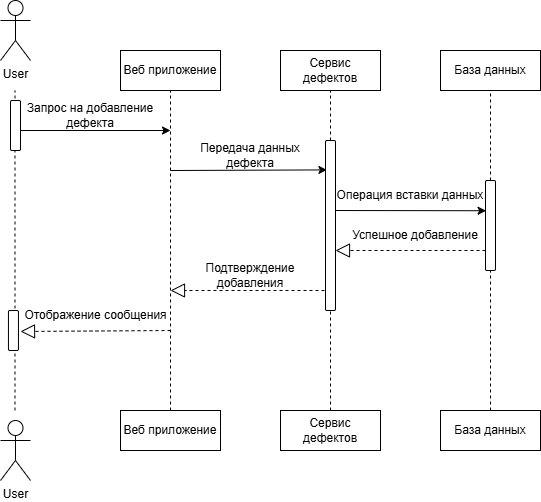


Рисунок 6 – Диаграмма последовательности

* + 1. Диаграмма компонентов

Диаграмма компонентов отображает структуру подсистемы хранения данных, показывая основные компоненты и их взаимодействие.



Рисунок 7 – Диаграмма компонентов

* + 1. Диаграмма пакетов

Диаграмма пакетов демонстрирует логическую организацию компонентов в подсистеме, упрощая понимание структуры и зависимостей между модулями.



Рисунок 8 – Диаграмма пакетов

* + 1. Диаграмма деятельности

Диаграмма деятельности описывает поток действий при работе с подсистемой хранения данных, показывая процесс добавления дефекта от начала до конца.

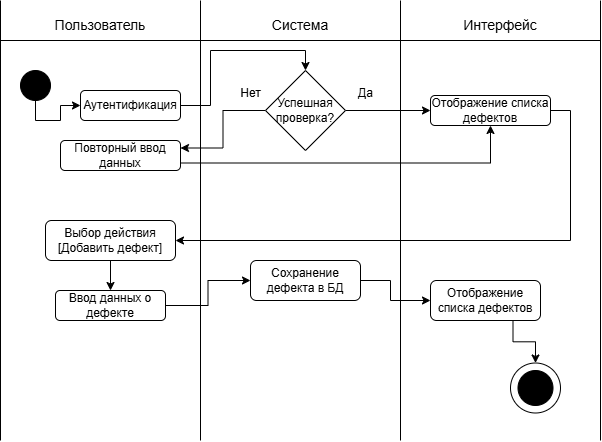


Рисунок 9 – Диаграмма деятельности

Задание 3:

Необходимо после этапа проектирования разработать выбранную подсистему.

Для начала я создала базу данных и таблицы в SQL Server, далее создала новый проект «**TextileDefectTracker» и добавила необходимые пакеты. Создала классы, соответствующие таблицам в базе данных, а также** **класс TextileDefectContext с классом DbContext - это класс, который представляет базу данных и позволяет взаимодействовать с ней. После этого я реализовала основные методы для работы с данными, для этого создала класс DefectService.** Далее **запустила приложение и проверила, что в консоли отображаются сообщения об успешном создании ткани и дефекта.**

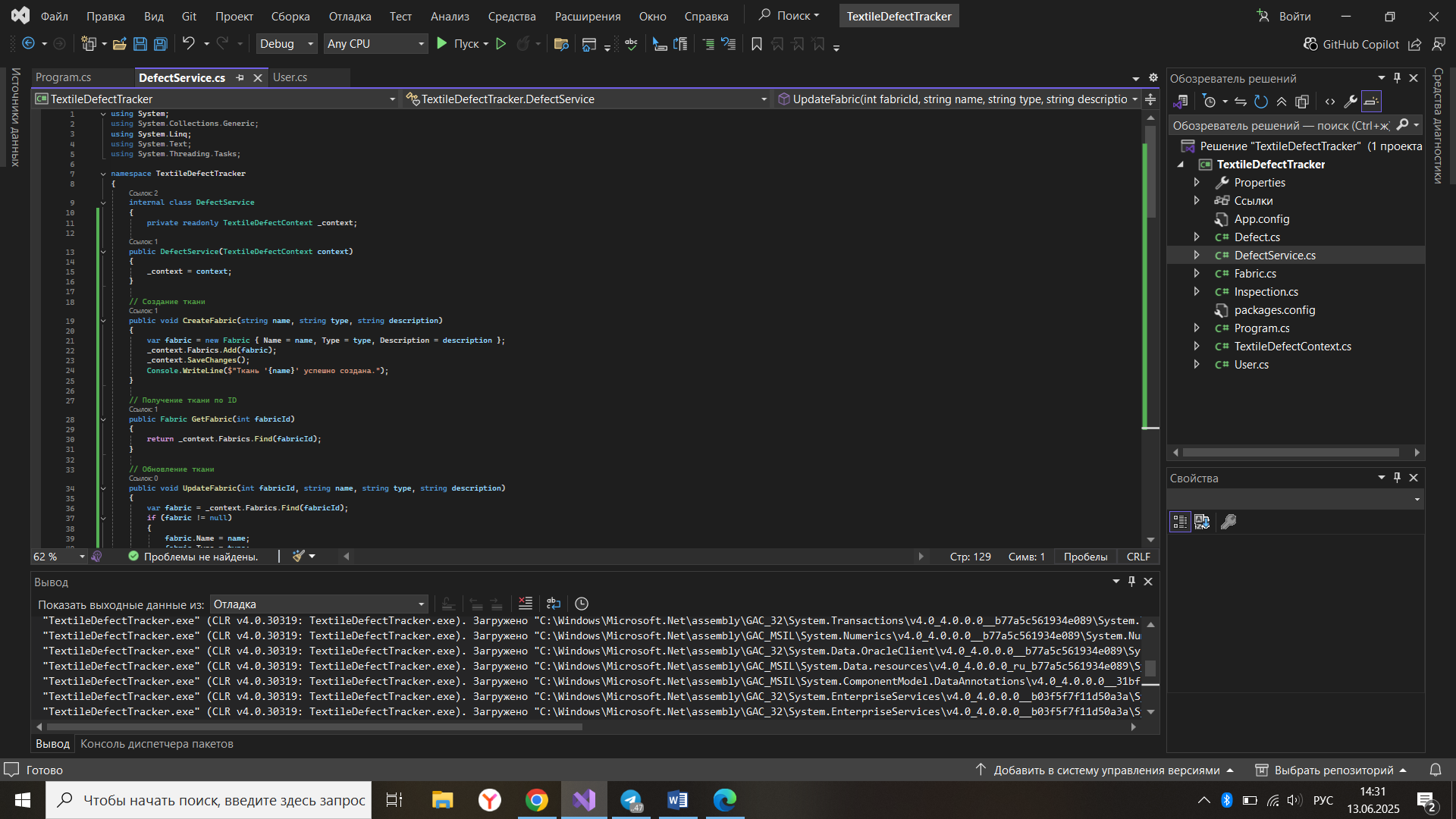


Рисунок 10 – Код программы

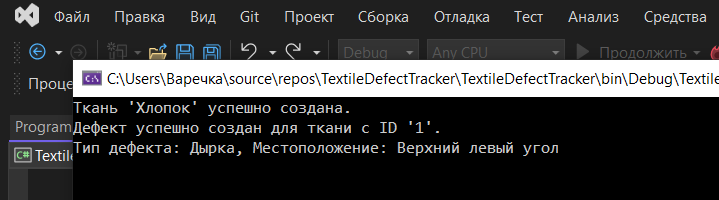


Рисунок 11 – Отображение сообщений в консоли

# ЗАКЛЮЧЕНИЕ

В ходе прохождения производственной практики были успешно достигнуты поставленные цели и задачи. Полученные практические навыки и знания в области проектирования, разработки, внедрения и сопровождения информационных систем позволили закрепить теоретическую базу, полученную в процессе обучения, и применить ее на практике при разработке модуля ИС.

В рамках практики был выполнен анализ предметной области, позволивший детально изучить специфику функционирования системы и определить требования к разрабатываемому модулю. На основе анализа была разработана проектная документация, включающая в себя описание функциональных и нефункциональных требований, архитектурное решение и проект базы данных.

Также была разработана документация по эксплуатации модуля ИС, обеспечивающая возможность его дальнейшего использования и сопровождения. Сформированная отчётная документация отражает все этапы прохождения практики, результаты выполненной работы и полученные навыки.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. <https://metanit.com/sharp/tutorial/3.46.php>
2. <https://app.diagrams.net/>
3. <https://www.mallenom.ru/?ysclid=m4l2ditzeb835497887>
4. <https://practicum.yandex.ru/blog/uml-diagrammy/>
5. https://projecto.pro/

ПРИЛОЖЕНИЕ А

|  |  |
| --- | --- |
| СОГЛАСОВАНО | УТВЕРЖДАЮ |
|  | Субъект предпринимательской  деятельности  Богданова Варвара Юрьевна  \_\_\_\_\_\_\_\_\_\_\_\_ (подпись)  «08» 06 2025 г.  (м.п.) |

**Подсистема хранения данных для системы распознавания дефектов на текстиле**

Техническое задание

(приложение к договору № \_\_\_\_)

Череповец

2025

Содержание

[1 Термины и определения 21](#_Toc200470172)

[2 Общие сведения 21](#_Toc200470173)

[2.1 Назначение документа 21](#_Toc200470174)

[2.2 Наименование Исполнителя и Заказчика 22](#_Toc200470175)

[2.3 Основание для разработки подсистемы 22](#_Toc200470176)

[2.4 Плановые сроки начала и окончания работ по созданию подсистемы 22](#_Toc200470177)

[2.5 Порядок оформления и предъявления результатов работ 22](#_Toc200470178)

[3 Назначение и цели создания подсистемы 22](#_Toc200470179)

[3.1 Назначение подсистемы 22](#_Toc200470180)

[3.2 Цели создания подсистемы 23](#_Toc200470181)

[3.2.1 Основные цели создания подсистемы 23](#_Toc200470182)

[3.2.2 Целевая аудитория 23](#_Toc200470183)

[4 Требования к подсистеме 23](#_Toc200470184)

[4.1 Требования к подсистеме в целом 23](#_Toc200470185)

[4.1.1 Требования к структуре и функционированию подсистемы 23](#_Toc200470186)

[4.1.2 Требования к персоналу 24](#_Toc200470187)

[4.1.3 Требования к сохранности информации 24](#_Toc200470188)

[4.1.4 Требования к разграничению доступа 24](#_Toc200470189)

[4.2 Требования к функциям, выполняемым подсистемой 24](#_Toc200470190)

[4.2.1 Основные требования 24](#_Toc200470191)

[4.2.1.1 Структура подсистемы 25](#_Toc200470192)

[4.2.2 Требования к функциональным возможностям 25](#_Toc200470193)

[4.3 Требования к видам обеспечения 26](#_Toc200470194)

[4.3.1 Требования к информационному обеспечению 26](#_Toc200470195)

[4.3.1.1 Требования к хранению данных 26](#_Toc200470196)

[4.3.1.2 Требования к языкам программирования 26](#_Toc200470197)

[4.3.2 Требования к лингвистическому обеспечению 26](#_Toc200470198)

[4.3.3 Требования к программному обеспечению 26](#_Toc200470199)

[4.3.4 Требования к аппаратному обеспечению 27](#_Toc200470200)

[5 Состав и содержание работ по созданию подсистемы 27](#_Toc200470201)

[6. Порядок контроля и приемки подсистемы 28](#_Toc200470202)

[6.1 Виды, состав, объем и методы испытаний 28](#_Toc200470203)

[6.2 Общие требования к приемке подсистемы 28](#_Toc200470204)

[7. Требования к составу и содержанию работ по вводу подсистемы в эксплуатацию 28](#_Toc200470205)

# Термины и определения

Специальные термины, используемые в настоящем ТЗ, приведены в таблице ниже. Прочая техническая терминология понимается в соответствии с действующими стандартами и рекомендациями международных органов, ответственных за вопросы стандартизации в сети Интернет.

Таблица 6 – Термины и определения

|  |  |
| --- | --- |
| Термин | Описание |
| Подсистема хранения данных (ПДХ) | Комплекс программно-аппаратных средств, обеспечивающий надежное хранение, обработку и предоставление данных |
| Система распознавания дефектов (СРД) | Программно-аппаратный комплекс, предназначенный для автоматического выявления дефектов на текстиле |
| Дефект | Несоответствие текстуры текстильного полотна установленным стандартам качества |
| Заказчик | Компания, заказывающая разработку веб-приложения |
| Исполнитель | Компания, выполняющая разработку веб-приложения |
| API (Application Programming Interface) | Интерфейс прикладного программирования |
| UI (User Interface) | Пользовательский интерфейс |
| Администратор | Пользователь с правами управления системой |
| Пользователь | Посетитель сайта, использующий функционал бронирования |

# Общие сведения

# Назначение документа

Данное техническое задание определяет требования к разработке подсистемы хранения данных для системы распознавания дефектов на текстиле. ТЗ является основой для разработки, тестирования и приемки подсистемы.

# Наименование Исполнителя и Заказчика

Заказчик

ООО «Малленом Системс» Адрес:423250, г. Череповец, ул. Металлургов, д. 21, кв. 2

Исполнитель

ИП Богданова Варвара Юрьевна Адрес: 423250, г.Череповец, ул. Вахитова, д. 1, кв. 2

# Основание для разработки подсистемы

Основанием для разработки подсистемы является Договор №\_\_ от \_\_\_\_\_\_\_\_\_\_\_ между Исполнителем и Заказчиком.

# Плановые сроки начала и окончания работ по созданию подсистемы

Начало работ: 08.06.2025

Окончание работ: 21.06.2025

Содержание и длительность отдельных этапов работ приведены в разделе 5 настоящего ТЗ. Сроки, состав и очередность работ являются ориентировочными и могут изменяться по согласованию с Заказчиком.

# Порядок оформления и предъявления результатов работ

Работы по созданию подсистемы производятся и принимаются поэтапно. По окончании каждого из этапов работ, перечисленных в разделе 5 настоящего ТЗ, Исполнитель представляет Заказчику соответствующие результаты, и стороны подписывают Акт сдачи-приемки работ.

# Назначение и цели создания подсистемы

# Назначение подсистемы

ПХД предназначена для обеспечения надежного, масштабируемого и безопасного хранения и обработки данных, генерируемых системой распознавания дефектов на текстиле. Она должна обеспечивать эффективный доступ к изображениям, метаданным и результатам анализа для различных пользователей и приложений.

# Цели создания подсистемы

# Основные цели создания подсистемы

* Обеспечение надежного хранения больших объемов изображений текстильных полотен.
* Обеспечение быстрого доступа к изображениям и метаданным для обучения, анализа и отчетности.
* Поддержка масштабируемости системы для обработки растущих объемов данных.
* Обеспечение безопасности данных и разграничение прав доступа.

# Целевая аудитория

* Операторы СРД.
* Аналитики данных.
* Контролеры качества.
* Администраторы системы.

# Требования к подсистеме

# Требования к подсистеме в целом

# Требования к структуре и функционированию подсистемы

* ПХД должна быть выполнена в виде модульной архитектуры, обеспечивающей гибкость и расширяемость.
* ПХД должна поддерживать распределенное хранение данных для обеспечения масштабируемости и отказоустойчивости.
* ПХД должна обеспечивать автоматическое резервное копирование данных.
* ПХД должна обеспечивать возможность восстановления данных после сбоев.
* ПХД должна предоставлять API для интеграции с другими системами.
* ПХД должна поддерживать мониторинг состояния и производительности.

# Требования к персоналу

Для эксплуатации ПХД требуется персонал, обладающий знаниями в области:

* + Администрирования операционных систем (Linux/Windows).
  + Администрирования баз данных (PostgreSQL, MySQL, NoSQL).
  + Программирования на C#.
  + Обеспечения информационной безопасности.

# Требования к сохранности информации

* ПХД должна обеспечивать сохранность данных в течение не менее 30 лет.
* Должны быть предусмотрены меры защиты от потери данных в результате аппаратных сбоев, ошибок программного обеспечения, вирусных атак и несанкционированного доступа.
* ПХД должна обеспечивать контроль целостности данных.

# Требования к разграничению доступа

* ПХД должна обеспечивать разграничение прав доступа к данным для различных пользователей и групп пользователей.
* Должны быть реализованы механизмы аутентификации и авторизации пользователей.

# Требования к функциям, выполняемым подсистемой

# Основные требования

* Прием изображений и метаданных от СРД.
* Хранение изображений и метаданных в структурированном виде.
* Обеспечение быстрого поиска и извлечения изображений по различным критериям (дата, тип дефекта, координаты, и т.д.).
* Предоставление API для доступа к данным из других систем.
* Управление правами доступа к данным.
* Мониторинг состояния и производительности системы.
* Резервное копирование и восстановление данных.

# Структура подсистемы

ПХД должна включать следующие компоненты:

* Интерфейс приема данных: Компонент, принимающий изображения и метаданные от СРД.
* Хранилище данных: Компонент, обеспечивающий хранение изображений и метаданных.
* API: Интерфейс для доступа к данным из других систем.
* Система управления правами доступа: Компонент, обеспечивающий разграничение прав доступа к данным.
* Система мониторинга: Компонент, обеспечивающий мониторинг состояния и производительности системы.
* Система резервного копирования и восстановления: Компонент, обеспечивающий резервное копирование и восстановление данных.

# Требования к функциональным возможностям

Прием и обработка данных:

* + Поддержка различных форматов изображений (JPEG, PNG, TIFF, и т.д.).
  + Автоматическая обработка метаданных.
  + Валидация входящих данных.

Хранение данных:

* + Поддержка различных типов хранилищ (файловая система, СУБД, облачное хранилище).
  + Индексирование данных для быстрого поиска.
  + Сжатие данных для экономии места.

Поиск и извлечение данных:

* + Поиск изображений по метаданным (дата, тип дефекта, координаты, и т.д.).
  + Возможность фильтрации и сортировки результатов поиска.
  + Предоставление данных в различных форматах (JSON, XML, и т.д.).

Управление правами доступа:

* + Создание и управление пользователями и группами пользователей.
  + Назначение прав доступа к различным данным.

Мониторинг:

* Мониторинг использования ресурсов системы (CPU, RAM, Disk).
  + Мониторинг производительности системы
  + Уведомления о событиях (ошибки, перегрузки, и т.д.).

Резервное копирование и восстановление:

* + Автоматическое резервное копирование данных.
  + Возможность восстановления данных из резервной копии.
  + Поддержка различных стратегий резервного копирования

# Требования к видам обеспечения

# Требования к информационному обеспечению

# Требования к хранению данных

Необходимо обеспечить хранение следующих типов данных:

* + Изображения текстильных полотен в высоком разрешении.
  + Метаданные изображений (размер, формат, дата создания и т.д.).
  + Результаты анализа изображений (тип дефекта, координаты, вероятность).
  + Журналы событий.

Общий объем хранимых данных должен составлять не менее 5 ТБ. Необходимо обеспечить возможность масштабирования хранилища данных по мере увеличения объема данных.

# Требования к языкам программирования

Для подсистемы должен использоваться язык С#

# Требования к лингвистическому обеспечению

Приложение должен быть выполнен на русском языкe. Должна быть предусмотрена возможность добавления второго языка на любой из страниц приложения.

# Требования к программному обеспечению

* Операционная система: Windows
* СУБД: MySQL
* Веб-сервер: Apache
* Необходимые библиотеки и фреймворки: TensorFlow

# Требования к аппаратному обеспечению

* Требования к аппаратному обеспечению определяются исходя из предполагаемой нагрузки и выбранных технологий.
* Необходимо обеспечить достаточное количество вычислительных ресурсов (CPU, RAM) и дискового пространства.
* Необходимо обеспечить отказоустойчивость системы путем использования RAID-массивов, резервных источников питания и других средств.

# Состав и содержание работ по созданию подсистемы

Состав и содержание работ по созданию ПХД должны включать следующие этапы:

Анализ требований и проектирование:

* Уточнение требований к ПХД.
* Разработка архитектуры ПХД.
* Проектирование схемы данных.
* Разработка интерфейсов.

Разработка программного обеспечения:

* Написание кода ПХД.
* Тестирование кода.
* Документирование кода.

Интеграция и тестирование:

* Интеграция ПХД с СРД и другими системами.
* Проведение комплексного тестирования.

Ввод в эксплуатацию:

* Установка и настройка ПХД на серверах Заказчика.
* Обучение персонала Заказчика.
* Передача документации Заказчику.

Гарантийное обслуживание:

* Устранение дефектов, выявленных в процессе эксплуатации.
* Консультации по вопросам эксплуатации.

# Порядок контроля и приемки подсистемы

# Виды, состав, объем и методы испытаний

Сдача-приемка выполненных работ должна осуществляться при предъявлении Исполнителем комплектов соответствующих документов и завершаться оформлением акта сдачи-приемки, подписанного Исполнителем и утвержденного Заказчиком. Испытания подсистемы проводятся силами Исполнителя.

# Общие требования к приемке подсистемы

Приемка подсистемы должна проводиться приемочной комиссией, в состав которой должны входить представители Заказчика и Исполнителя, в течение трех рабочих дней после завершения работ. Результаты работы комиссии должны оформляться актом, подписанным членами комиссии и утверждённым Заказчиком.

# Требования к составу и содержанию работ по вводу подсистемы в эксплуатацию

* Установка и настройка ПХД на серверах Заказчика.
* Миграция данных из существующих систем (при необходимости).
* Обучение персонала Заказчика.
* Разработка инструкций по эксплуатации ПХД.
* Тестирование ПХД в реальных условиях эксплуатации.
* Подписание акта ввода в эксплуатацию.

# ПРИЛОЖЕНИЕ Б

Руководство пользователя и администратора

1. Введение

Руководство предназначено для пользователей и администраторов подсистемы хранения данных системы распознавания дефектов на текстиле. Оно содержит информацию о функциональности Подсистемы, ее настройке и эксплуатации.

1. Обзор подсистемы

Подсистема предназначена для хранения и управления данными о дефектах ткани, выявленных Системой. Она предоставляет API для доступа к этим данным другим компонентам Системы.

1. Роли пользователей

* Пользователь (Оператор): Имеет доступ к чтению данных о дефектах, добавлению новых дефектов, обновлению информации о существующих дефектах и удалению дефектов.
* Администратор: Имеет полный доступ ко всем функциям Подсистемы, включая управление пользователями, настройку системы, резервное копирование и восстановление данных.

1. Руководство пользователя (Оператора)
   1. Доступ к API:
   * Для доступа к API Подсистемы необходимо использовать HTTP-клиент.
   * Необходимо получить токен аутентификации от администратора.
   1. Добавление нового дефекта:
   * Используйте метод POST/defects.
   * Необходимо предоставить данные о дефекте в формате JSON в теле запроса.
   * В случае успешного добавления, API вернет код 201 (Created) и JSON с данными о созданном дефекте, включая его идентификатор.
   1. Получение информации о дефекте:
   * Используйте метод GET/defects/{defect\_id}, где {defect\_id} - идентификатор дефекта.
   * В случае успешного получения, API вернет код 200 (OK) и JSON с данными о дефекте.
   * В случае, если дефект не найден, API вернет код 404 (Not Found).
   1. Обновление информации о дефекте:
   * Используйте метод PUT/defects/{defect\_id}, где {defect\_id} - идентификатор дефекта.
   * Необходимо предоставить данные о дефекте в формате JSON в теле запроса.
   * В случае успешного обновления, API вернет код 200 (OK) и JSON с обновленными данными о дефекте.
   * В случае, если дефект не найден, API вернет код 404 (Not Found).
   1. Удаление дефекта:
   * Используйте метод DELETE/defects/{defect\_id}, где {defect\_id} - идентификатор дефекта.
   * В случае успешного удаления, API вернет код 204 (No Content).
   * В случае, если дефект не найден, API вернет код 404 (Not Found).
   1. Поиск дефектов:
   * Используйте метод GET/defects.
   * Можно использовать параметры запроса для фильтрации результатов. Например: /defects?type=дыра - поиск дефектов типа "дыра".

* API вернет код 200 (OK) и JSON с массивом дефектов, соответствующих критериям поиска.

1. Руководство администратора
   1. Установка и настройка:
   * Настройка базы данных: Необходимо настроить подключение к базе данных в конфигурационном файле Подсистемы.
   * Настройка API: Необходимо настроить параметры API (порт, хост, параметры безопасности) в конфигурационном файле Подсистемы.
   1. Управление пользователями:
   * Администратор имеет право создавать, удалять и изменять права доступа пользователей.
   * Управление пользователями осуществляется через API или через веб-интерфейс администратора.
   * При создании пользователя необходимо указать его логин, пароль и роль.
   1. Резервное копирование и восстановление данных:
   * Регулярное резервное копирование данных является важной частью администрирования Подсистемы.
   * Необходимо настроить еженедельное автоматическое резервное копирование базы данных.
   1. Мониторинг:

* Администратор должен отслеживать состояние Подсистемы (загрузку сервера, использование дискового пространства, количество ошибок).
* Рекомендуется использовать инструменты мониторинга, такие как Prometheus и Grafana.
  1. Обновление программного обеспечения:
* Необходимо регулярно обновлять программное обеспечение Подсистемы для получения новых функций и исправления ошибок безопасности.
* Перед обновлением необходимо сделать резервную копию данных.

1. Устранение неполадок

* Ошибка подключения к базе данных: Проверьте параметры подключения к базе данных в конфигурационном файле.
* Ошибка аутентификации: Проверьте правильность токена аутентификации. Убедитесь, что токен не истек.
* Ошибка сервера (500): Проверьте логи сервера на наличие ошибок. Обратитесь к разработчикам.
* Некорректные данные: Проверьте формат и правильность передаваемых данных.

1. Безопасность

* Не храните пароли в открытом виде.
* Используйте надежные пароли.
* Регулярно обновляйте программное обеспечение для исправления уязвимостей.
* Ограничьте доступ к Подсистеме только авторизованным пользователям.
* Используйте шифрование для защиты данных при передаче по сети.