**Техническое задание**

1. Введение

Данное техническое задание определяет требования к подсистеме хранения данных для системы распознавания дефектов на текстиле. Система предназначена для автоматического выявления и классификации дефектов ткани на производственных линиях. Подсистема должна обеспечивать надежное, эффективное и масштабируемое хранение информации о выявленных дефектах, а также предоставлять API для доступа к этим данным для других компонентов Системы.

1. Цели и задачи

Цель: Разработать подсистему хранения данных, обеспечивающую надежное и эффективное хранение и обработку данных о дефектах ткани.

Задачи:

* + Обеспечить хранение данных о каждом выявленном дефекте, включая его характеристики (тип, размер, положение, изображение).
  + Обеспечить возможность добавления, удаления и обновления информации о дефектах.
  + Предоставить API для доступа к данным о дефектах для других компонентов Системы.
  + Обеспечить возможность поиска и фильтрации дефектов по различным критериям.
  + Обеспечить масштабируемость подсистемы для обработки больших объемов данных.
  + Обеспечить надежное хранение данных и возможность восстановления в случае сбоев.
  + Обеспечить разграничение прав доступа к данным.

1. Требования к функциональности
   1. Хранение данных о дефектах:

Подсистема должна хранить следующую информацию о каждом дефекте:

* Идентификатор дефекта (уникальный).
* Тип дефекта (например, "дыра", "пятно", "затяжка").
* Размер дефекта (ширина, высота, площадь).
* Положение дефекта на ткани (координаты).
* Изображение дефекта (ссылка на изображение или само изображение в формате, например, JPEG, PNG).
* Дата и время обнаружения дефекта.
* Дополнительные метаданные (например, комментарии оператора).
* Подсистема должна поддерживать хранение данных о различных типах дефектов с возможностью добавления новых типов в будущем.
  1. API для доступа к данным:

Подсистема должна предоставлять API для выполнения следующих операций:

* + - Добавление нового дефекта.
    - Получение информации о дефекте по идентификатору.
    - Удаление дефекта по идентификатору.
    - Обновление информации о дефекте.
    - Поиск дефектов по различным критериям (тип, размер, положение, дата, рулон и т.д.).
    - Получение статистики по дефектам.
  1. Поиск и фильтрация:

Подсистема должна обеспечивать возможность поиска и фильтрации дефектов по следующим критериям:

* + Тип дефекта.
    - Размер дефекта.
    - Положение дефекта.
    - Дата обнаружения.
    - Комбинация указанных критериев.
  1. Масштабируемость:

Подсистема должна быть спроектирована таким образом, чтобы ее можно было масштабировать для обработки больших объемов данных и увеличения нагрузки.

* 1. Надежность и восстановление:
* Подсистема должна обеспечивать надежное хранение данных и защиту от потери данных в случае сбоев.
* Необходимо предусмотреть механизмы резервного копирования и восстановления данных.
  1. Безопасность:
* Подсистема должна обеспечивать разграничение прав доступа к данным на основе ролей.
* API должен поддерживать аутентификацию и авторизацию.

1. Требования к производительности

* Время ответа на запросы API (на добавление, получение, обновление, удаление данных) должно быть не более 1 секунды в 95% случаев.
* Время выполнения поисковых запросов должно быть не более 2 секунд для запросов, возвращающих до 1000 результатов.
* Подсистема должна обеспечивать обработку не менее 100 запросов в секунду.

1. Требования к программному обеспечению и оборудованию

Программное обеспечение:

* + Операционная система: Windows.
  + Язык программирования: C#
  + База данных: PostgreSQL
  + Платформа разработки для реализации сервиса: ASP.NET.

Оборудование:

* + Серверы с достаточной вычислительной мощностью, объемом оперативной памяти и дискового пространства для хранения данных и обработки запросов.
  + Сеть с высокой пропускной способностью.

1. Требования к интерфейсам

* API: RESTful API
* Интерфейс администратора (GUI): Веб-интерфейс для управления подсистемой.

1. Требования к документации

* Техническое задание.
* Руководство пользователя и администратора.

1. Этапы разработки

Этап 1: Анализ требований и проектирование архитектуры.

Этап 2: Разработка API и базы данных.

Этап 3: Реализация функциональности добавления, удаления, обновления и поиска данных.

Этап 4: Разработка тестов.

Этап 5: Развертывание и настройка подсистемы.

Этап 6: Тестирование и исправление ошибок.

Этап 7: Написание документации.

1. Приемочные испытания

Приемочные испытания должны включать:

* Проверку соответствия функциональности требованиям технического задания.
* Проверку производительности и масштабируемости.
* Проверку надежности и безопасности.
* Проверку качества документации.