

Санкт-Петербургский государственный университет

Береснёв Руслан Анатольевич

Выпускная квалификационная работа

Разработка веб-приложения для детекции дефектов ленты промышленного конвейера

Уровень образования: бакалавриат

Направление *02.03.03 «Математическое обеспечение и администрирование информационных систем»*

Основная образовательная программа *СВ.5162.2021 «Технологии программирования»*

Научный руководитель:
старший преподаватель кафедры системного программирования, к. т. н. Литвинов Ю. В.

Консультант:
заместитель технического директора, ООО «Системы компьютерного зрения»
Смирнова Е. А.

Рецензент:
генеральный директор, ООО «Системы компьютерного зрения» Степанов Д. Н.

Санкт-Петербург
2025

Saint Petersburg State University

Ruslan Beresnyov

Bachelor's Thesis

Development of a web application for detecting industrial conveyor belt defects

Education level: bachelor

Speciality *02.03.03 "Software and Administration of Information Systems"*

Programme *CB.5162.2021 "Programming Technologies"*

Scientific supervisor:
C.Sc., senior lecturer Y.V. Litvinov

Consultant:
Deputy technical director at Computer Vision Systems LLC E.A. Smirnova

Reviewer:
General manager at Computer Vision Systems LLC D.N. Stepanov

Saint Petersburg
2025

Оглавление

Введение	4
1. Постановка задачи	6
2. Обзор	7
2.1. Виды дефектов, встречающихся на производстве	7
2.2. Методы обнаружения дефектов ленты конвейера	8
2.3. Существующие системы дефектоскопии	11
2.4. Используемые технологии	19
3. Реализация	23
3.1. Архитектура сервера	23
3.2. Разделы веб-приложения	29
3.3. Особенности реализации	33
3.4. Контейнеризация с помощью Docker Compose	37
3.5. Тестирование бизнес-логики веб-приложения	38
4. Апробация	40
5. Заключение	41
Список литературы	42

Введение

Конвейеры широко используются во многих отраслях промышленности для транспортировки руды. Они автоматизируют процесс её доставки до нужного пункта и минимизируют использование человеческих ресурсов, что позволяет лишь осуществлять дистанционный контроль стабильности процесса.

В результате длительного использования и интенсивной работы на конвейерах часто возникают дефекты ленты, которые могут привести к её разрывам и простоям конвейера. Некоторые из наиболее распространённых дефектов включают в себя порывы, порезы, сквозные отверстия на ленте и её износ.

Простои конвейеров могут обойтись предприятиям в миллионы рублей упущенной прибыли. Кроме того, такие неисправности приводят к задержкам в производстве и поставках, что также негативно сказывается на репутации компании. Чтобы минимизировать риск дефектов и простоев конвейеров, компании должны проводить их регулярный осмотр и выполнять техническое обслуживание. На некоторых предприятиях осмотр каждого конвейера по всей длине до сих пор производится работниками вручную во время ежедневного обхода — на это может быть затрачено несколько часов, а также такой подход не позволяет увидеть все дефекты и отследить процесс их развития.

Для автоматизации процесса уже существуют и используются системы дефектоскопии, основанные на различных технических методах обнаружения неисправностей. Однако из-за санкций приобретение готового решения стало гораздо сложнее, а российские системы могут не быть заточены для работы именно на промышленных конвейерах, транспортирующих руду.

Компания ООО «Системы Компьютерного Зрения» получила заказ на создание программного обеспечения для производства, на котором имеется оборудование для получения снимков ленты конвейера. Компанией уже разработана библиотека для классификации дефектов по снимкам, которая вносит полученную информацию в базу данных, по-

этому в рамках данной работы необходимо создать полноценное удобное веб-приложение для работы с дефектами из базы данных и отправки оповещений. С помощью приложения работник предприятия будет уведомляться о появляющихся дефектах, сможет оперативно принимать необходимые меры и взаимодействовать с обнаруженными неисправностями, а также следить за общим состоянием ленты конвейера.

1. Постановка задачи

Целью работы является разработка полноценного веб-приложения к имеющейся базе данных для автоматизации дефектоскопии ленты конвейера. Были поставлены следующие задачи.

1. Провести обзор дефектов ленты и методов их обнаружения.
2. Провести обзор существующих аналогов, использующихся для дефектоскопии, и выявить их сильные стороны для сбора требований к собственному решению.
3. Спроектировать архитектуру веб-приложения.
4. Реализовать бизнес-логику и API сервисов согласно выявленным требованиям.
5. Реализовать пользовательский интерфейс системы.
6. Написать тесты для функциональности системы.
7. Провести контейнеризацию веб-приложения.

2. Обзор

В данном разделе описаны виды возникающих дефектов, технические методы их обнаружения, существующие решения для инспекции состояния ленты конвейера, а также использованные технологии для создания собственного веб-приложения.

2.1. Виды дефектов, встречающихся на производстве

В этом разделе обзора классифицированы неисправности, которые могут образоваться на ленте конвейера или же внутри её конструкции. Всего выделено восемь типов неисправностей, которые на данный момент распознаёт система компьютерного зрения.

1. Скол торца ленты (см. рис. 1a).
2. Расслоение ленты (см. рис. 1b).
3. Оголение троса каркаса (см. рис. 1c).
4. Трещина на поверхности (см. рис. 1d).
5. Задир ленты (см. рис. 1e).
6. Сквозное отверстие (см. рис. 1f).
7. Порез ленты (см. рис. 1g).
8. Износ поверхности ленты (см. рис. 1h).

В 4-ой главе книги [11] отмечено, что небольшие отклонения направления ленты приводят к сколам по торцам, застрявшие предметы в зоне загрузки материала вызывают порезы, истирание и задиры ленты, к сквозным отверстиям может привести падение острых предметов, а транспортировка горячих материалов может стать причиной расслоения ленты в некоторых местах. Стоит также отметить, что трещины на

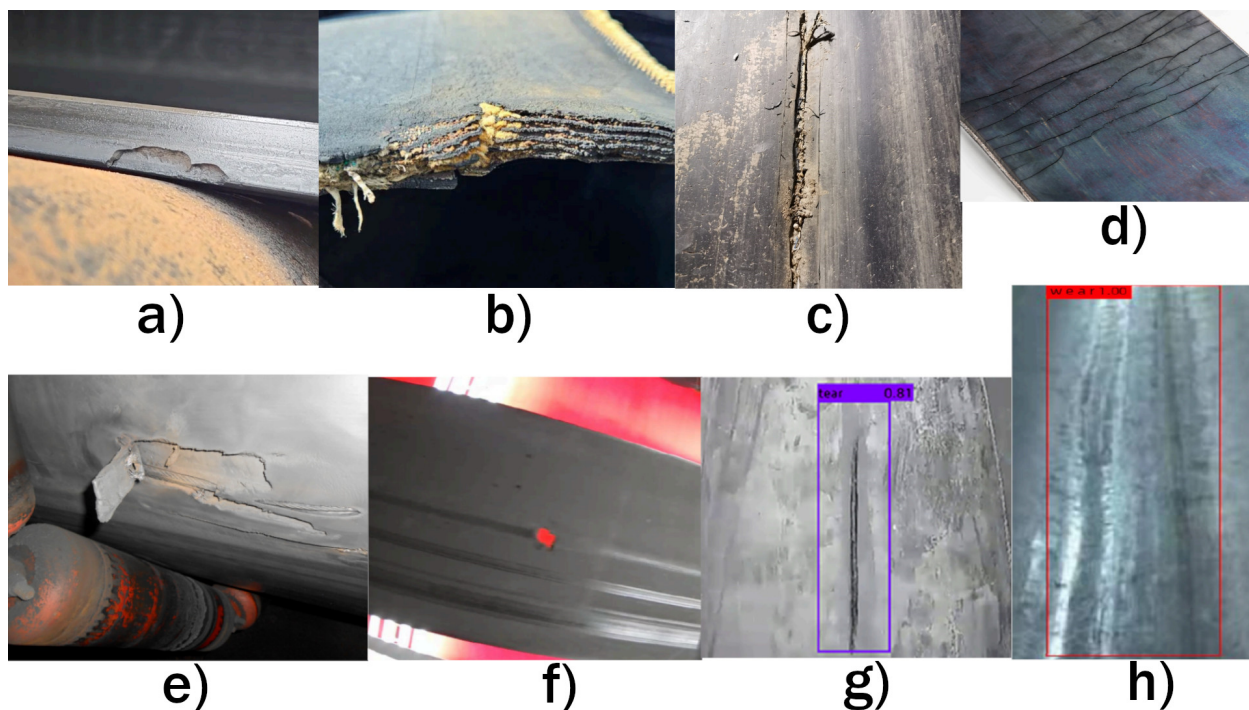


Рис. 1: Примеры возникающих дефектов

поверхности конвейера появляются из-за неравномерного физического напряжения, а при тяжёлом износе могут возникать желоба, которые при большой глубине приводят к оголению троса в каркасе ленты.

2.2. Методы обнаружения дефектов ленты конвейера

В данном разделе освещены различные технические методы, на основе которых проектируются устройства для выявления дефектов ленты конвейера. Местом поиска дефектов может быть как поверхность конвейерной ленты, так и её «каркас». Таким образом, цель обзора методов детекции — рассмотреть интересные решения, которые потенциально могут быть добавлены на производство, ведь одна система, например, специализируется на мониторинге поверхности ленты конвейера, а другая — на целостности её внутренней части.

2.2.1. Анализ 3D-модели поверхности

В [30] указано, что для построения 3D-модели поверхности конвейера используются лазеры, проецирующие опорную линию поперёк ленты, а затем камера с частотой в несколько сотен герц сканирует лазерный рисунок. Из искажений опорных линий отдельных кадров строятся 3D-изображения, которые вместе затем формируют 3D-модель профиля поверхности. Появившиеся таким образом отклонения от плоскости на 3D-модели позволяют автоматически обнаружить поверхностный дефект ленты конвейера и выявить его тип.

Преимущества данного подхода заключаются в невосприимчивости метода к цвету и текстуре ленты, а также в автоматическом обнаружении дефекта и остановке конвейера благодаря высокочастотной камере. Таким образом, обнаружение дефектов может производиться автоматически в условиях плохой освещённости. Недостатком является невозможность контролировать состояние внутренней части ленты — например, целостность металлических стержней каркаса.

2.2.2. Датчики, основанные на принципе магнитной индукции

Один из методов, указанный в [3], позволяет обнаружить механически повреждённые или подвергшиеся коррозии металлические стержни каркаса: специальное устройство регистрирует конфигурацию магнитного поля, образуемого исправными стержнями, и преобразует её в цифровые сигналы или 2D-изображение, а далее сравнивает с ней в режиме реального времени текущую конфигурацию магнитного поля. Неисправные металлические стержни вызывают изменения в магнитном поле, и если разница в конфигурациях превышает пороговое значение, это свидетельствует о повреждении каркаса ленты. Преимуществом данного метода является обнаружение дефектов в режиме реального времени.

Похожий метод упоминается в [1] и демонстрируется в [6], но отличается тем, что используется для обнаружения порезов на поверхности ленты конвейера: специальная магнитная сетка в виде нескольких

полос встраивается внутрь ленты, и при возникновении пореза на месте расположения сетки изменяется её магнитное поле, что фиксируется при прохождении повреждённой сетки через датчик. Существенный недостаток данного метода — необходимость размещать множество сеток по длине конвейера, при этом всё равно не вся поверхность ленты сможет контролироваться.

2.2.3. Датчики, основанные на рентгеновском излучении

Как отмечено в [32], на одну из сторон нижней ленты конвейера устанавливается источник, который облучает ленту, а с другой её стороны находятся фотодиодные матрицы, принимающие излучение и преобразующие дошедшие частицы в электрические сигналы. Таким образом, формируется проекция стержней каркаса. Затем электрические сигналы усиливаются и преобразуются в цифровые, а после данного этапа полученное изображение обрабатывается алгоритмами компьютерного зрения, в результате чего выявляются дефекты.

Механизм обнаружения дефектов заключается в количестве радиации, получаемой приёмником. Если стержни каркаса растянуты, разорваны или подверглись коррозии, то интенсивность проникновения рентгеновского излучения в этом месте будет выше, так как поглощается меньшее количество лучей.

Преимущество данного метода заключается в возможности обнаружения дефектов внутри ленты в режиме реального времени, однако, как отмечено в [3], недостатками являются невозможность обнаруживать порезы поверхности и опасность радиационного излучения для человека.

2.2.4. Методы получения и обработки 2D-изображений

Для получения изображений поверхности ленты в [3] сравниваются методы с использованием как промышленных камер, ориентированных на видимый спектр и чувствительных к количеству света и текстуре поверхности, так и более дорогих инфракрасных камер, которые за счёт

большей длины волн позволяют обнаруживать дефекты (например, порезы) в условиях плохой освещённости или сильной запылённости. Более того, в [15] описан метод раннего обнаружения порезов ленты: с помощью алгоритмов компьютерного зрения выделяется яркая область снимка, которая образовалась на месте с повышенной температурой, что означает истирание поверхности некоторым предметом в реальном времени. Такой способ позволяет автоматически остановить конвейер и предотвратить появление обширного пореза. Также существует метод [14], обрабатывающий изображение, полученное при совмещении обычного и инфракрасного снимков. Данный метод сочетает преимущества обоих подходов: высокое качество изображения, полученного обычной камерой, и возможность раннего обнаружения дефектов поверхности ленты с помощью камеры инфракрасного спектра.

Полученное изображение подготавливается с помощью алгоритмов шумоподавления, преобразования в оттенки серого, выравнивания гистограммы и т.д., а затем происходит обнаружение дефектов методами компьютерного зрения, а также с помощью нейронных сетей, основанных на глубоком обучении. Нейронные сети предоставляют более современный способ детекции и позволяют классифицировать дефекты по типам.

2.3. Существующие системы дефектоскопии

Для анализа функциональности и сбора требований к собственному решению были отобраны аналоги, в каждом из которых рассматривались функции обнаружения дефектов и работы с ними: история появления, информация о дефектах, система отчётов, уведомления в критичных случаях и т.д.

Поиск осуществлялся в пределах первых двух страниц в поисковой системе Google с помощью запросов «Conveyor belt defect detection system» и «Система обнаружения дефектов ленты конвейера». Отбирались системы, имеющие хотя бы минимальную информацию в открытом доступе. Также поиск возможной информации проводился сред-

ствами ResearchGate и GoogleScholar.

Таким образом, цель данного обзора — выявить полезную функциональность, понять сильные стороны конкретных реализаций, проанализировать пользовательский интерфейс и на основе полученных данных спроектировать и создать собственное веб-приложение.

2.3.1. Vizorlabs

Vizorlabs — российская система компьютерного зрения, одним из инструментов которой является веб-приложение для работы с поверхностными дефектами, обнаруженными посредством 2D-лидаров. В руководстве администратора [33] подробно представлены функциональность и информация о пользовательском интерфейсе.

В разделе «Список конвейеров» указаны параметры конвейеров и лидаров, относящихся к ним. При выборе конвейера можно посмотреть конкретный отрезок ленты, где схематично в виде цветowych ячеек показаны место и степень повреждения (либо наличие износа), а также поперечный профиль ленты в виде цифровой проекции.

В разделе «Статистика» собрана краткая статистика дефектов за период по всем конвейерам, а также отображена сортируемая таблица с информацией о всех выявленных дефектах: id, дата, конвейер, тип дефекта (износ, порез, сквозной дефект), фото. По каждому дефекту из списка можно посмотреть подробную информацию в отдельном окне. Доступна функция экспорта — как для всей таблицы, так и для отдельного дефекта.

В разделе «История изменений» представлена информация по прогонам за период в табличном виде. Для каждого прогона отражены id, дата и количество выявленных дефектов по типам.

В приложении есть отдельная функциональность для администратора с настройками данных о пользователях и их ролях в системе. Аутентификация пользователей осуществляется посредством электронной почты. В системе также присутствует возможность уведомлять пользователей о появившихся дефектах с помощью письма на электронную почту, либо сообщением в Telegram. Данные о дефекте можно отправить

в формате pdf, xls или docx.

Сильные стороны приложения заключаются в наличии подробной статистики по каждому дефекту и возможности схематично просмотреть всю ленту по секторам, а также в детальной настройке сервисов для отправки уведомлений и формирования отчётов.

2.3.2. Continental Industry

Continental Industry — немецкая корпорация, выпустившая серию продуктов для инспекции конвейерных лент [7]. Одни системы нацелены на обнаружение неисправностей каркаса ленты (CordProtect), другие — на обнаружение поверхностных дефектов (RipProtect Plus, SurfaceProtect), а некоторые рассчитаны практически на весь спектр дефектов (TotalProtect, MultiProtect). Принцип работы данных систем основан на рентгеновском излучении, магнитной индукции и анализе с помощью лазерных лучей.

Программное обеспечение доступно в веб-приложении или в мобильном приложении. По каждому дефекту предоставляется информация о его виде, критичности, размере и положении на ленте. Также можно скачивать отчёты с информацией о дефектах или отслеживать статус ленты онлайн: в приложении отображается цветовая карта изношенности ленты и прогнозируется её оставшееся время жизни.

Преимущества продуктов Continental являются ориентированность на различные конвейерные ленты и широкий спектр обнаружения дефектов за счёт различных технических методов.

2.3.3. ML Sense

ML Sense — российская система, основанная на использовании машинного зрения и нейронных сетей, разработанная для контроля качества продукции на производствах конвейерного типа. На сервере (под управлением ОС Linux) анализируется поступающий с камеры видеопоток, а далее обнаруженные неисправности передаются в приложение оператора, также создавая оповещение.

В приложении определения целостности конвейерной ленты [18] доступны следующие функции:

- просмотр снимков ленты и обнаруженных неисправностей;
- классификация неисправностей на «отсутствие соединительной заклёпки» и «наличие прочих дефектов поверхности»;
- отображение положения и размера неисправностей;

Как указано в [17], сильной стороной системы является наличие базы данных с размеченными дефектами и возможность самообучения для классификации неисправностей.

2.3.4. Fenner Dunlop

Fenner Dunlop — нидерландская компания, предоставляющая две системы автоматического мониторинга конвейеров, Rip Ranger и Eagle Eye, а также приложение BIRDSi для работы с ними [10] (как на мобильные устройства, так и на компьютеры).

Rip Ranger предназначен для обнаружения порезов ленты с помощью встроенных в ленту индуктивных петель. Eagle Eye предназначен как для обнаружения дефектов металлического каркаса, так и для обнаружения порезов на поверхности. BIRDSi позволяет отслеживать в реальном времени все события, происходящие с лентой, а также её текущий статус. В приложении сохраняется история всех возникших ранее дефектов и поддерживаются мгновенные оповещения по SMS или электронной почте в случае появления новых неисправностей. Присутствует возможность добавлять заметки к каждому конвейеру.

Преимущество BIRDSi заключается в возможности автоматически отслеживать состояние сразу всех конвейеров через главную панель.

2.3.5. Roxon

Roxon — финский производитель двух устройств для мониторинга состояния конвейерных лент: HX170 (ориентирован только на верхнюю

поверхность ленты) и НХ270 (позволяет следить за обеими сторонами ленты). Принцип работы основан на построении 3D-модели поверхности и её анализе.

В [30] и [31] показаны функциональность и пользовательский интерфейс системы НХ270. При обнаружении дефекта возникает звуковой сигнал и отправляется уведомление на электронную почту оператора, а при критических дефектах лента автоматически останавливается. Пользователь может сам настроить критичность дефекта в зависимости от его параметров. Для каждого дефекта отображается информация о его типе, уровне критичности, дате и времени возникновения, расположении на ленте, длине, ширине, площади и глубине.

Из главного меню приложения можно выбрать конкретный конвейер и далее просматривать информацию по его событиям. В середине экрана отображается фото ленты с местом дефекта, а также его параметры. Сбоку доступна панель со списком появившихся на конвейере дефектов, каждый из которых можно открыть отдельно или удалить. Из приложения оператор может остановить конвейер, а также направить выбранный дефект на станцию инспекции и починки. Это осуществляется прокручиванием ленты до нужной точки. Помимо информации о дефектах в приложении имеется отдельное меню [29] для просмотра износа ленты и её толщины по всей длине в виде графиков.

Сильной стороной приложения является подробность информации о каждом возникающем дефекте.

2.3.6. BeltSpy

BeltSpy — австралийская система, основанная на компьютерном зрении. Система распознаёт три вида дефектов: дефекты двух типов соединений ленты и дефекты поверхности.

В [5] показан подробный обзор функциональности и пользовательского интерфейса системы. В приложении на главной странице схематично отображён весь конвейер с пометками дефектов на нём и общей информацией о параметрах конвейера, а также есть краткая сводка количества обнаруженных дефектов. Вся подробная информация фик-

сируется на боковой панели в виде таблиц. В одной таблице указаны произошедшие события: тип обнаруженного дефекта и его критичность, дальность в метрах относительно начального положения ленты и комментарии; в другой таблице хранится список снимков всей ленты в разные моменты времени.

Система не обнаруживает неисправности автоматически. После того, как сделан снимок всего конвейера, оператор может провести инспекцию дефектов. При инспекции дефекты соединений ленты сразу регистрируются системой, а места с подозрением на поверхностные дефекты оператор последовательно просматривает сам и может отметить критичность дефекта и добавить к нему комментарий, тогда дефект будет зарегистрирован в таблице событий. Конвейер может быть остановлен автоматически, когда выбранный оператором дефект достигнет станции починки.

Сильной стороной системы является возможность создавать снимки всей ленты конвейера и детально просматривать каждый из них.

2.3.7. Fagus GreCon BELTVIEW

Fagus GreCon BELTVIEW — немецкий продукт, основанный на оптическом методе контроля состояния ленты конвейера. Система сканирует поверхность и создаёт её цифровую карту для дальнейшего анализа возможных дефектов.

Работа с дефектами осуществляется через десктопное приложение на базе Windows. Как указано в [9], в приложении имеется следующая функциональность:

- графическое отображение поверхности ленты;
- отображение графиков толщины и износа ленты;
- задание пороговых значений для обнаружения неисправностей;
- подробное отображение каждого дефекта;

- отправка оповещения оператору и автоматическая остановка конвейера в случае возникновения критических дефектов;
- сохранение всех неисправностей в базе данных с информацией о положении, размерах и классификации;
- возможность выгрузки настраиваемых pdf-отчетов с регулярным интервалом.

2.3.8. CBGuard

CBGuard — немецкая компания, разработавшая систему «Life Extender» [4] для обнаружения поверхностных дефектов и определения толщины ленты посредством лазерного сканирования, а также выявления внутренних дефектов посредством рентгеновских лучей.

В десктопном приложении на базе Windows в реальном времени отображается видео с помеченными дефектами и информацией об их типе, критичности (от 1 до 3) и местоположении. При возникновении отклонений от заданных параметров оператору отправляется SMS-оповещение, он может вручную остановить конвейер и приблизить найденный дефект. В правой части интерфейса отображаются две таблицы: найденные дефекты различных видов и информация о соединительных звеньях. В приложении доступна функция выгрузки детальных отчётов со всеми дефектами.

2.3.9. Open-source веб-приложение для Amazon

Помимо проприетарных систем были рассмотрены и open-source решения. На GitHub по запросу «Conveyor belt defect detection» было показано пять репозиторийев, из которых релевантным является только один [39]. В остальных непустых репозиториях либо была разработана система компьютерного зрения, либо решение не ориентировано на промышленные конвейеры для транспортировки руды.

В [39] создано веб-приложение с системой компьютерного зрения, разработанное для Amazon. Система детекции дефектов реализована

на Python с использованием библиотек OpenCV и NumPy, а для веб-приложения был использован фреймворк Flask. Для хранения данных применена СУБД SQLite.

В репозитории опубликовано подробное руководство [38] по системе. В приложении доступна информация о прямоте ленты конвейера, количестве порезов, и оценка общего качества поверхности ленты. Каждый день в виде таблицы автоматически генерируются отчёты с актуальным состоянием конвейера. Полученные значения сравниваются с бенчмарками (введённые пользователем пороговые значения), а затем отображается статистика по параметрам ленты, а также её снимки. В системе используются роли, так что только авторизованные пользователи могут менять настройки бенчмарков и выполнять определённые действия. Помимо функциональности для пользователей система предоставляет журналы со всеми операциями, что позволяет отслеживать возникающие ошибки.

2.3.10. Выводы из обзора существующих систем дефектоскопии

В каждой системе присутствуют различные полезные функции, многие из которых являются общими для рассмотренных систем. Все особенности можно объединить в следующие:

- просмотр ленты в режиме реального времени, либо просмотр фотографий;
- общее состояние конвейерной ленты и/или её толщина;
- просмотр истории возникших дефектов в виде таблиц или списков;
- информация об отдельном дефекте: его вид, критичность, фотография, время возникновения, положение на ленте;
- система оповещений оператора предприятия через E-mail/Telegram/уведомления в приложении/звуковые сигналы;

- выгрузка отчётов о найденных дефектах и состоянии конвейера в форматах pdf/xls/docx;
- функции остановки конвейера и перемещения места с дефектом к станции починки;
- настройки параметров критичности для дефектов, а также настройки параметров уведомлений и ролей в системе;
- пользовательский интерфейс с информацией, разделённой по вкладкам.

Были рассмотрены две российские системы, однако стоит отметить, что заказчик предпочёл собственное решение, поэтому собранные особенности были учтены и использованы при создании веб-приложения.

2.4. Используемые технологии

В данном разделе представлены ключевые технологии, использованные для создания веб-приложения, а также причины выбора данных технологий.

2.4.1. FastAPI

FastAPI [36] — достаточно современный веб-фреймворк, появившийся в 2018 году, поэтому был выбран как инструмент для создания API на Python. Более того, по количеству скачиваний [21] уступает только фреймворку Flask [22], являясь одним из самых популярных веб-фреймворков. FastAPI имеет документацию с множеством примеров и подробным туториалом, разбитым на темы, в котором постепенно создаётся веб-приложение. Важными особенностями FastAPI являются автоматическая генерация документации формата OpenAPI по существующим эндпоинтам.

2.4.2. PostgreSQL

PostgreSQL [19] — СУБД, которая на данный момент уже используется в проекте для выгрузки информации в базу данных из библиотеки компьютерного зрения на C++, поэтому на стороне Python взаимодействие происходит с ней же.

2.4.3. SQLAlchemy

SQLModel [37] — библиотека на основе SQLAlchemy и Pydantic от создателя FastAPI для взаимодействия с СУБД с использованием механизма ORM (Object-Relational Mapping). ORM позволяет не работать с базой данных напрямую, а обращаться к моделям, представляющим таблицы, и объектам моделей, представляющим записи в этих таблицах. Pydantic же предоставляет возможность валидировать входные данные запросов в соответствии с заранее созданными JSON-подобными моделями, а также использовать данные модели для ORM. Библиотека SQLAlchemy была выбрана, потому что, как и FastAPI, имеет подробную документацию, а также легко интегрируется с FastAPI.

2.4.4. Gmail API

Google API используется для взаимодействия с разными сервисами Google. Через Google Cloud Console можно выбрать необходимый сервис и создать проект с ним, а далее настроить аутентификацию с использованием учётных данных. Gmail API [12] предоставляет функциональность для работы с почтовым ящиком в Google-аккаунте пользователя, в частности, возможность отправлять письма. Для аутентификации используется стандарт OAuth, позволяющий не задействовать такие данные, как логин и пароль. При первом запросе к API создаётся сервер авторизации, на котором пользователь выбирает аккаунт, на этот аккаунт выдаются определённые права (например, только чтение писем), а после формируется токен, который в дальнейшем служит для автоматической аутентификации и авторизации.

2.4.5. Python-telegram-bot

В приложении используется библиотека `python-telegram-bot` [35] для создания бота и работы с Telegram API в Python. Данная библиотека была выбрана как одна из самых популярных для создания Telegram-ботов на Python. Для сравнения, она в несколько раз чаще используется [24], чем другая достаточно популярная библиотека `aiogram` [20].

2.4.6. Reportlab

Для создания pdf-документов в пределах сервиса выгрузки отчётов была задействована библиотека `reportlab` [28]. С её помощью размещены таблицы, фотографии и по-разному отформатированный текст. Среди аналогов данная библиотека была выбрана за её возможность более гибко и удобно настраивать все элементы на холсте внутри pdf-документа, чем, например, с помощью `fpdf2` или `weasyprint`. Более того, `reportlab` имеет самое большое количество скачиваний [25], если сравнивать с `fpdf2` [23] и `weasyprint` [26].

2.4.7. React

Для создания пользовательского интерфейса веб-приложения была использована JavaScript-библиотека `React` [27]. С помощью `React` были разработаны переиспользуемые JSX-компоненты, на которых строится каждый раздел пользовательского интерфейса. Был выбран среди `Vue.js` и `Angular`, так как в требованиях был указан как предпочтительный.

2.4.8. Node.js

`Node.js` [16] — среда выполнения JavaScript, которая была задействована для запуска веб-сервера.

2.4.9. Axios

Axios [2] — HTTP-клиент для легконастраиваемого выполнения запросов к серверу из браузера. Был использован для создания отдельных для каждого сервиса axios-экземпляров, которые позволяют получать доступ к API лишь по названию эндпоинта, что улучшает читаемость и сокращает код, по сравнению со стандартной функцией `fetch`.

2.4.10. Material UI

Material UI [13] — библиотека React-компонентов от Google, имеющая более 20 миллионов скачиваний в месяц [34] и используемая в дизайне многих популярных сервисов. Перед созданием пользовательского интерфейса также были рассмотрены библиотеки Mantine и Chakra UI, однако Material UI была выбрана из-за субъективных предпочтений в дизайне, несмотря на то, что Mantine и Chakra UI являются более новыми библиотеками.

2.4.11. Docker

Docker [8] — инструмент для сборки всей логики приложения в контейнеры с отдельной средой, с помощью которых можно быстро запустить веб-приложение.

3. Реализация

3.1. Архитектура сервера

В данном подразделе представлена имеющаяся на данный момент архитектура приложения в виде диаграммы компонентов и их описаний, а также приведены реализованные эндпоинты в рамках API веб-приложения.

3.1.1. Компоненты и сервисы сервера

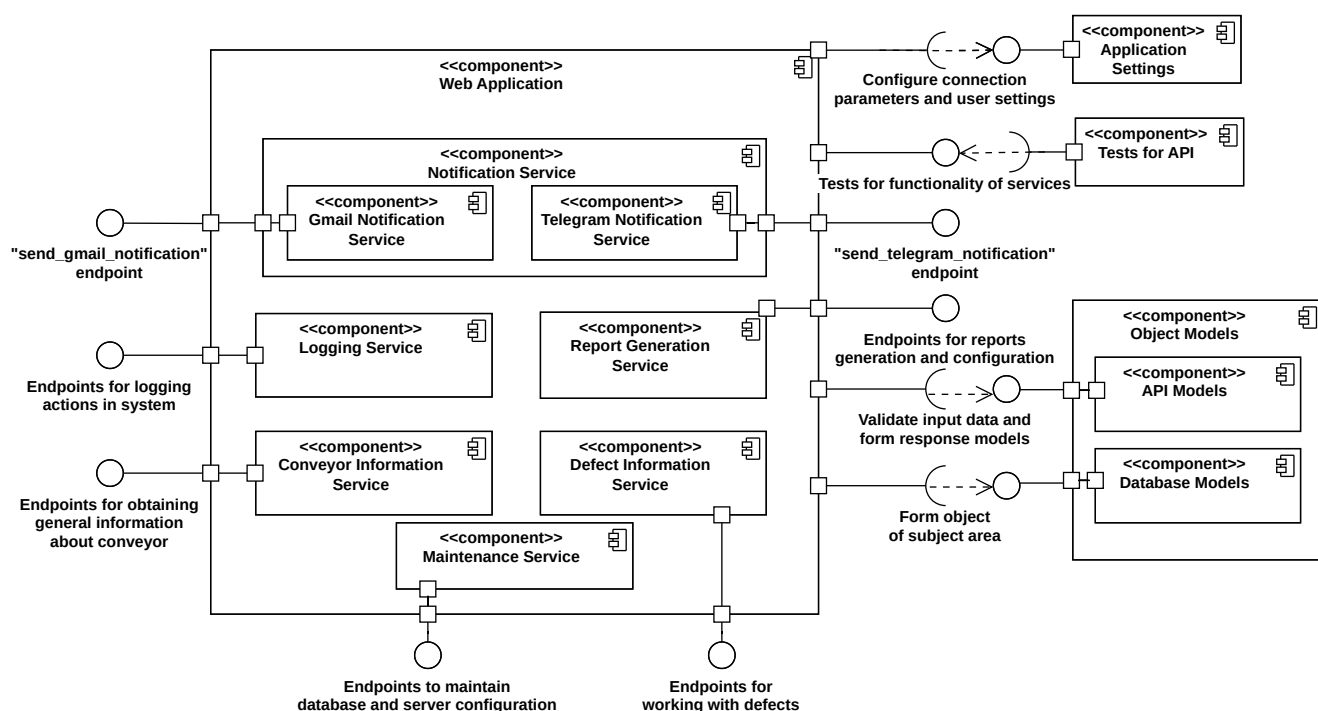


Рис. 2: Диаграмма компонентов

На рис. 2 главным компонентом является Web Application, представляющий основную бизнес-логику приложения, все эндпоинты сервера и реализующий взаимодействие с внешней средой. Внутри данного компонента находятся шесть сервисов: сервис для отправки оповещений, сервис для работы с дефектами, сервис контроля общего состояния ленты конвейера, сервис логирования событий, сервис генерации и выгрузки отчётов, а также сервис поддержки базы данных и всего приложения.

- **Notification Service** реализует два способа отправки оповещений: через сообщение от Telegram-бота и через письмо со специального Gmail-аккаунта. В обоих случаях оповещения будут отправлены оператору предприятия для быстрого реагирования на появление нового дефекта, а также для получения созданных отчётов.
- **Defect Information Service** позволяет получать отфильтрованные по различным параметрам дефекты из базы данных, менять их критичность или удалять их. По каждому дефекту также предоставляется подробная информация и предыдущие дефекты в цепочке прогрессирования.
- **Conveyor Info Service** предоставляет информацию об общем статусе конвейера, его параметрах, а также количестве дефектов в каждой группе критичности.
- **Logging Service** отвечает за создание записей логов всех действий, происходящих в приложении, а также позволяет работать с ними — например, получить все логи действий конкретного типа.
- **Report Generation Service** предоставляет возможность сформировать отчёт по всем дефектам, по конкретному дефекту или об общем статусе конвейера и его параметрах в формате .pdf/.csv.
- **Maintenance Service** позволяет мгновенно пересоздать базу данных, заполнить её необходимой информацией, добавлять тестовые дефекты, а также предоставляет возможность проверить работоспособность приложения и загрузить пользовательские настройки.

Компонент API Models содержит модели для валидации тела запроса, а также формирования ответов на запросы к API приложения, а в Database Models находятся модели для объектного представления таблиц базы данных посредством механизма ORM. Модели в API Models

наследуются от класса `BaseModel` из библиотеки `pydantic`, а модели в `Database Models` наследуются от класса `SQLModel` из библиотеки `sqlmodel`, который, в свою очередь, наследуется от `BaseModel`. Таким образом, компонент `Object Models` является абстракцией над данными компонентами.

Ещё одним компонентом является `Application Settings`, представляющий конфигурацию для подключения и параметров, которые не могут быть опубликованы в репозитории. Например, такими параметрами являются токен Telegram-бота и данные пользователя при подключении к СУБД.

`Tests for API` не является компонентом, реализующим непосредственно какую-либо функциональность бизнес-логики. Данный компонент отвечает за тестирование сервисов сервера.

3.1.2. Реализованные эндпоинты

В рамках API веб-приложения были реализованы следующие эндпоинты:

- `/` — получить информацию о веб-приложении;
- `/notification` — получить информацию о сервисе оповещений;
- `/notification/with_telegram` — отправить оповещение в виде сообщения в Telegram;
- `/notification/with_gmail` — отправить оповещение в виде письма на указанную почту Gmail;
- `/defect_info` — получить информацию о сервисе работы с дефектами;
- `/defect_info/count` — получить общее количество дефектов, а также количества по группам критичности;
- `/defect_info/all` — получить подробную информацию по всем имеющимся дефектам;

- **/defect_info/id=<defect_id>** — получить подробную информацию о дефекте по его id;
- **/defect_info/type=<defect_type>** — получить информацию по всем дефектам указанного типа;
- **/defect_info/critical** — получить информацию по всем дефектам критического уровня;
- **/defect_info/extreme** — получить информацию по всем дефектам около-критического (экстремального) уровня;
- **/defect_info/by_period** — получить информацию по всем дефектам за некоторый период (в запрос могут быть переданы два необязательных параметра типа datetime: начало и конец периода);
- **/defect_info/filtered** — получить список дефектов, отфильтрованных сразу по всем параметрам (тип, критичность, промежуток времени);
- **/defect_info/id=<current_defect_id>/previous** — получить подробную информацию о предыдущей вариации текущего дефекта по его id;
- **/defect_info/id=<current_defect_id>/chain_of_previous** — получить подробную информацию обо всех предыдущих вариациях текущего дефекта в его цепочке прогрессии;
- **/defect_info/id=<defect_id>/set_criticality** — задать новый уровень критичности для дефекта по его id (нормальный/-экстремальный/критический);
- **/defect_info/id=<defect_id>/delete** — удалить запись о дефекте по его id;
- **/conveyor_info** — получить информацию о сервисе, предоставляющем текущее состояние конвейера и его базовые параметры;

- **/conveyor_info/parameters** — получить информацию о таких базовых параметрах конвейера как длина, ширина и толщина ленты;
- **/conveyor_info/status** — получить информацию о текущем состоянии конвейера (нормальное/экстремальное/критическое);
- **/conveyor_info/create_record** — создать новую запись в таблице состояний конвейера;
- **/conveyor_info/change_parameters** — обновить технические параметры конвейера;
- **/logs** — получить информацию о сервисе логирования;
- **/logs/all** — получить все записи логов;
- **/logs/id=<log_id>** — получить определённую запись лога по id;
- **/logs/type=<log_type>** — получить записи лога только определённого типа;
- **/logs/create_record** — создать новую запись лога с помощью параметров log_type и log_text;
- **/logs/id=<log_id>/delete** — удалить определённую запись лога по id;
- **/logs/delete_all** — полностью очистить лог;
- **/report** — получить информацию о сервисе для создания отчётов;
- **/report/all/pdf** — выгрузить отчёт по всем дефектам в pdf-формате;
- **/report/id=<defect_id>/pdf** — выгрузить отчёт по конкретному дефекту в pdf-формате;

- **/report/conveyor/pdf** — выгрузить отчёт с информацией о параметрах конвейера и его актуальном статусе в pdf-формате;
- **/report/all/csv** — выгрузить отчёт по всем дефектам в csv-формате;
- **/report/id=<defect_id>/csv** — выгрузить отчёт по конкретному дефекту в csv-формате;
- **/report/conveyor/csv** — выгрузить отчёт с информацией о параметрах конвейера и его актуальном статусе в csv-формате;
- **/maintenance/** — получить информацию о сервисе обслуживания;
- **/maintenance/get_events** — подписаться на события сервера (Server-Sent Events);
- **/maintenance/check_server** — проверить доступность сервера;
- **/maintenance/check_database** — проверить доступность базы данных;
- **/maintenance/create_tables** — создать или пересоздать все таблицы базы данных;
- **/maintenance/fill_database** — заполнить базу данных необходимыми и тестовыми данными;
- **/maintenance/add_test_defect** — добавить тестовый дефект в базу данных;
- **/maintenance/make_relation** — создать связь между двумя дефектами;
- **/maintenance/remove_relation** — удалить связь между двумя дефектами;

- `/maintenance/get_user_notification_settings` — получить настройки уведомлений пользователя;
- `/maintenance/update_user_notification_settings` — обновить настройки уведомлений пользователя.

3.2. Разделы веб-приложения

В данном подразделе продемонстрированы секции клиентской части, а также описана их функциональность. Переключение между секциями осуществляется через меню (см. рис. 3).

Типа	Criticality	Timestamp
hole	extreme	01.01.2025 - 00:00:00
rope	critical	02.01.2025 - 00:00:00
wear	normal	01.02.2025 - 00:00:00
wear	normal	01.02.2025 - 00:00:00
wear	normal	01.02.2025 - 00:00:00
wear	normal	01.02.2025 - 00:00:00
wear	normal	01.02.2025 - 00:00:00
wear	normal	01.02.2025 - 00:00:00
wear	normal	01.02.2025 - 00:00:00
wear	normal	01.02.2025 - 00:00:00
wear	normal	01.02.2025 - 00:00:00
wear	normal	01.02.2025 - 00:00:00
wear	normal	01.02.2025 - 00:00:00
wear	normal	01.02.2025 - 00:00:00

Рис. 3: Раскрывающееся меню веб-приложения

3.2.1. Работа с дефектами

Здесь отображается краткая информация о каждом дефекте в виде таблицы (см. рис. 4). Записи можно фильтровать по типу, критичности и принадлежности настраиваемому временному интервалу. Каждый дефект имеет критичность “normal”/“extreme”/“critical” и выделяется соответствующим цветом. Также по всем дефектам можно выгрузить отчёт (в формате .pdf или .csv), который будет выглядеть похожим образом.

При нажатии на строку таблицы открывается вкладка с подробной информацией о выбранном дефекте (см. рис. 5). Здесь отображаются

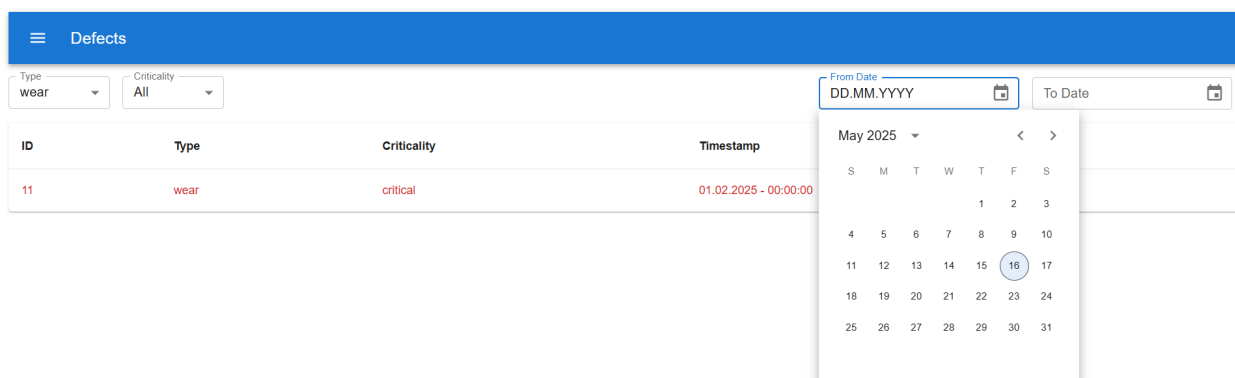


Рис. 4: Раздел “Defects”

его id, параметры, критичность и фотография. Есть опции для изменения критичности дефекта и его удаления. Также стоит отметить, что если дефект постепенно прогрессировал, для него отображается цепочка всех предыдущих вариаций, к каждой из которых можно перейти, чтобы изучить подробнее. В данной вкладке можно выгрузить отчёт со всей информацией о дефекте.

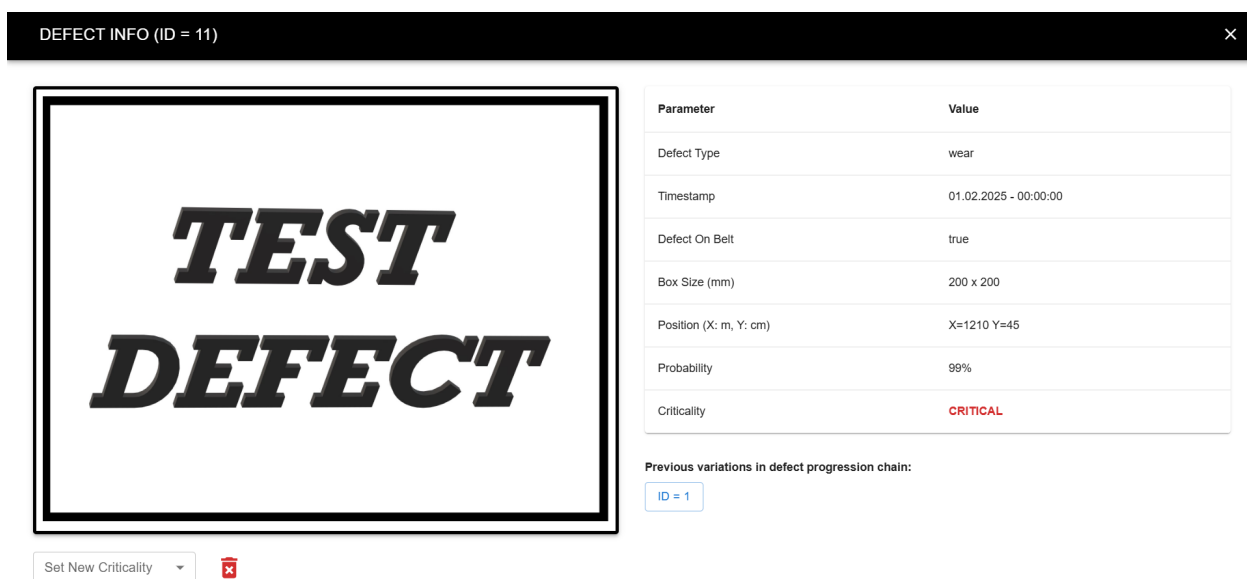


Рис. 5: Раскрывающаяся вкладка с подробной информацией о дефекте

3.2.2. Информация о конвейере

В данном разделе представлены две карточки (см. рис. 6): технические параметры конвейера и его текущий статус, а также сводка по общему количеству дефектов и количеству в каждой группе критичности.

При нажатии на кнопку “Change Parameters” появляется диалоговое окно для задания новых значений.

Как и в разделе “Defects”, здесь так же доступна выгрузка отчётов. При нажатии на кнопки они становятся неактивными на пару секунд, а в это время на сервере генерируется отчёт с информацией о конвейере в формате .pdf или .csv, а затем по умолчанию отправляется пользователю в Telegram и на Gmail.

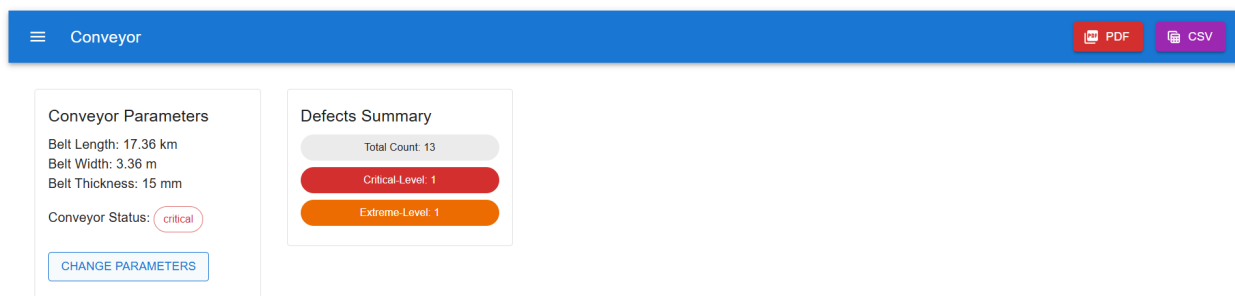


Рис. 6: Раздел “Conveyor”

3.2.3. Журнал логов

В журнале логов отображаются все более-менее значимые события, происходящие в системе (см. рис. 7). Это может быть, например, сведение о новом дефекте или же запись о созданном отчёте. Записи можно фильтровать по типу и отобразить, например, только определённое количество, по умолчанию равное десяти. Каждому типу присвоена собственная иконка для лучшей читаемости. В отличие от дефектов, логи можно удалить сразу все.

3.2.4. Настройки приложения

В разделе с настройками имеются две секции: настройки оповещений и настройки базы данных (см. рис. 8). В первой секции пользователь может выбрать, каким образом хочет получать созданные отчёты и оповещения о новых дефектах. Помимо этого есть возможность сразу же протестировать систему оповещений. Во второй секции находятся опции для пересоздания всех таблиц базы данных и заполнения

Logs			
Type	All	Show latest	10
Icon	Log	Type	Timestamp
	Report of the all defects in .pdf format has successfully generated	REPORT_INFO	22.05.2025 - 16:06:06
	Log record with id=22 has removed successfully	ACTION_INFO	22.05.2025 - 16:05:50
	Base parameters of the conveyor were updated	INFO	22.05.2025 - 16:05:03
	Notification with the subject "NEW NORMAL-LEVEL DEFECT ON THE CONVEYOR!" was successfully sent via Gmail to the address rusber03102003@gmail.com	MESSAGE	22.05.2025 - 16:03:02
	Notification with the text "NEW NORMAL-LEVEL DEFECT ON THE CONVEYOR!" id = 15 timestamp = 2025-02-01 00:00:00 type = wear is_on_belt = True box_width_in_mm = 200 box_length_in_mm = 200 longitudinal_position = 1210000 transverse_position = 450 probability = 99 criticality = normal" was successfully sent via Telegram to the user @razber3	MESSAGE	22.05.2025 - 16:02:58
	Set current general status of conveyor: "critical"	STATE_OF_DEVICES	22.05.2025 - 16:02:57
	New normal-level defect with id=15 has appeared on the conveyor!	WARNING	22.05.2025 - 16:02:57

Рис. 7: Раздел “Logs”

их необходимыми данными после этого. Также пользователь может посмотреть на реакцию приложения, добавив тестовый дефект.

Settings

Notification Settings

On new defect

Send reports

SEND TEST TG-MESSAGE

SEND TEST GMAIL-LETTER

Database Settings

RECREATE DB

FILL DB

ADD TEST DEFECT

Рис. 8: Раздел “Settings”

3.2.5. Диалоговые окна

В веб-приложении реализованы три вида диалоговых окон: оповещение о новом дефекте на конвейере (см. рис. 9), возникшая ошибка при выполнении запроса (см. рис. 10) и подтверждение действия (см. рис. 11): удаление дефектов, логов или пересоздание таблиц базы данных.

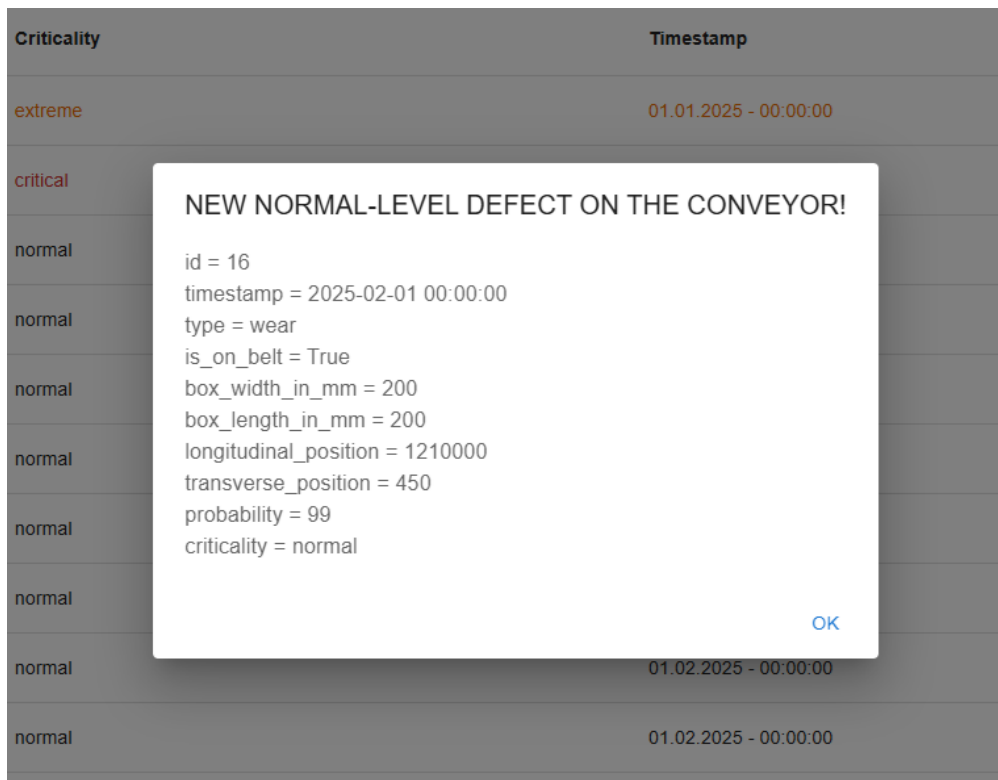


Рис. 9: Диалоговое окно, отображающееся при появлении нового дефекта

3.3. Особенности реализации

В данном разделе освещены технические аспекты реализации имеющихся сервисов, клиент-серверного взаимодействия и параметров приложения.

3.3.1. Отправка оповещений через Telegram

При первой отправке оповещения через Telegram API с использованием токена бота собираются все сообщения, отправленные ему за последние сутки, из сообщений берётся и сохраняется в .env-файл id чата с пользователем, который последним отправлял сообщение, а при следующих вызовах эндпоинта `/notification/with_telegram` id чата берётся уже из .env-файла.

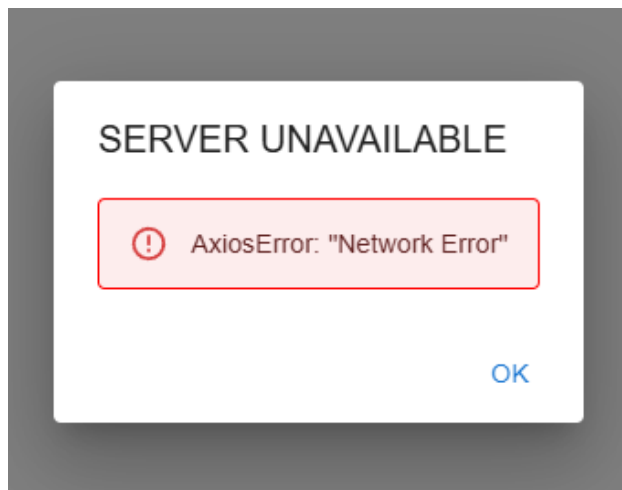


Рис. 10: Диалоговое окно, отображающееся при возникновении ошибки

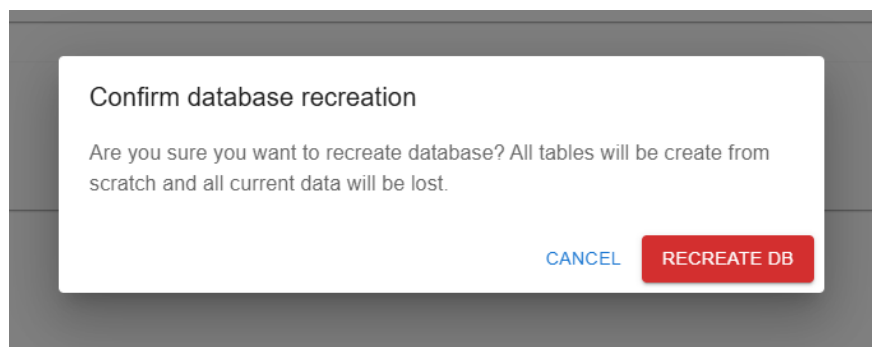


Рис. 11: Диалоговое окно, отображающееся для подтверждения операции

3.3.2. Отправка оповещений через Gmail

Письма с информацией о возникших дефектах и отчёты отправляются с Gmail-аккаунта, созданного специально для веб-приложения. На данном аккаунте в сервисе Google Cloud Console создан проект для взаимодействия с Gmail API. Проект предоставляет секретные данные пользователя (“client_secret.json”), которые используются при первой аутентификации через протокол OAuth 2.0.

При первом запросе к эндпоинту `/notification/with_gmail` необходимо выбрать созданный Gmail-аккаунт и пройти аутентификацию на запущенном в браузере специальном сервере. После аутентификации из данных в файле “client_secret.json” создаётся файл “token.json” с токеном, из которого при дальнейших запросах формируются учётные данные пользователя (credentials) и, таким образом, происходит авто-

матическая аутентификация. Стоит отметить, что токен действителен в течение одного часа, однако автоматически обновляется с помощью refresh-токена из файла “client_secret.json” после истечения. Когда и refresh-токен станет недействительным, пользователю необходимо будет снова пройти аутентификацию.

Файл “token.json” уже создан и должен быть помещён вместе с файлом “client_secret.json” в корень репозитория. Таким образом, благодаря токену приложение имеет возможность отправлять письма на почту пользователя без необходимости прохождения аутентификации каждый раз.

3.3.3. Работа с базой данных и модели таблиц

Для получения данных о конвейере, дефектах, логах, пользовательских настройках необходимо с сервера установить подключение к СУБД PostgreSQL. Для этого используется функция `create_engine` из библиотеки `sqlmodel`, создающая единый объект, сконфигурированный для работы с PostgreSQL, который затем используется в других файлах для создания сессий подключения к базе данных.

Каждый запрос к базе данных происходит в рамках сессии `Session`. Таблицы базы данных, с которыми происходит содержательная работа всех сервисов, кроме сервиса отправки оповещений, представлены в отдельном файле в виде моделей, наследующихся от класса `SQLModel` из библиотеки `sqlmodel`. Работа с внешними ключами осуществляется с помощью класса `Relationship` из `sqlmodel` для обращения к записи, на которую указывает внешний ключ, как к объекту, а не через `id`.

3.3.4. Механизм оповещения сервера базой данных

При создании таблиц базы данных сервис поддержки также создаёт триггер на добавление записей в таблицу с дефектами. При запуске сервера в файле “db_listener.py” создаётся задача, постоянно ожидающая сообщения от базы данных в отдельном канале, в который при появлении нового дефекта триггером отправляется информация о данном

дефекте с помощью встроенной в PostgreSQL функции “pg_notify()”. Сервер принимает информацию, создаёт из неё модель дефекта и отправляет оповещение через Telegram и Gmail. На клиент оповещение передаётся с помощью технологии Server-Sent Events.

3.3.5. Механизм оповещения клиента с помощью Server-Sent Events

Как только на клиенте отрендерится в первый раз основной компонент веб-приложения (“App.js”), с помощью хука вызывается эндпоинт `/maintenance/get_events`, который «подписывает» клиента на события сервера, создавая очередь сообщений (класс `asyncio.Queue`) между ними для односторонней отправки. Как только сервер получает оповещение от базы данных, в очередь сообщений кладётся информация о новом дефекте и, таким образом, отсылается клиенту. Клиент принимает сообщения с помощью стандартного JavaScript-класса `EventSource`, и при появлении нового оповещения отображает диалоговое окно с информацией о дефекте.

3.3.6. Особенности реализации сервиса для работы с дефектами

Дефекты со временем прогрессируют, если их не устранять, поэтому для каждого дефекта предусмотрена собственная цепь развития. В ней фиксируются все параметры дефекта в различные периоды времени. Данная цепь представляет собой односвязный список, который строится по записям в модели `Relation`, в которой всего два столбца — `id_current` и `id_previous`, которые являются внешними ключами, указывающими на модель `Defect`.

Также стоит отметить, что изменение критичности дефекта может повлиять на изменение общего состояния конвейера, а при удалении дефекта происходит обновление его цепи прогрессии.

3.3.7. Механизм логирования действий

Каждое нетривиальное действие в системе логируется (кроме успешного выполнения GET-запросов). Для этого вызывается функция, в которую передаются тип и текст новой записи лога.

3.3.8. Проверка доступности сервера и базы данных

В сервисе поддержки есть эндпоинты `/maintenance/check_server` и `/maintenance/check_database`, которые клиент вызывает каждые 10 секунд для проверки работоспособности веб-приложения. Если сервер или база данных недоступны, появляется диалоговое окно с сообщением об ошибке.

3.3.9. Файлы конфигурации и секретные данные

Для хранения таких данных, как, например, токен Telegram-бота или имя пользователя, используется файл `“.env”`, который необходимо создать до первого запуска приложения по шаблону класса `Settings`, валидирующего структуру файла `“.env”`. Класс `Settings` находится в файле конфигурации.

3.4. Контейнеризация с помощью Docker Compose

Веб-приложение было контейнеризировано с использованием `Docker Compose` (см. рис. 12), в котором были созданы три контейнера: для базы данных, сервера и клиента. Контейнеры могут взаимодействовать друг с другом благодаря внутренней сети `Docker Compose`.

Контейнер с базой данных создан из `docker`-образа `“postgres:17”` с использованием внутреннего хранилища, благодаря которому данные сохраняются даже после перезапуска контейнера.

Контейнер с `FastAPI`-сервером создан из `docker`-образа `“python:3.13”` с установкой дополнительных зависимостей. Помимо ORM-запросов к базе данных сервер «слушает» канал `“new_defect”`, и при появлении оповещения обрабатывает данные о новом дефекте.

Контейнер с клиентом создан из docker-образа “node:22”. Веб-сервер с помощью HTTP-клиента “axios” получает необходимую информацию с backend-сервера, рендерит JSX-файлы в статичные .js-файлы, а затем отправляет их браузеру вне контейнера для отображения пользовательского интерфейса. Также клиент с помощью Server-Sent Events фоново ожидает от backend-сервера оповещения о новых дефектах.

Стоит отметить, что docker-контейнеры не развёрнуты в облачном сервисе, поэтому веб-приложение работает только локально и доступно в браузере по адресу **http://localhost:3000/**.

3.5. Тестирование бизнес-логики веб-приложения

Для сервиса работы с дефектами были созданы модульные тесты, однако чтобы их запустить, необходимы локально запущенный FastAPI-сервер и отдельная тестовая база данных.

Тесты запускаются с помощью python-библиотеки “pytest”. Перед запуском всех тестов таблицы тестовой базы данных автоматически пересоздаются сервисом поддержки.

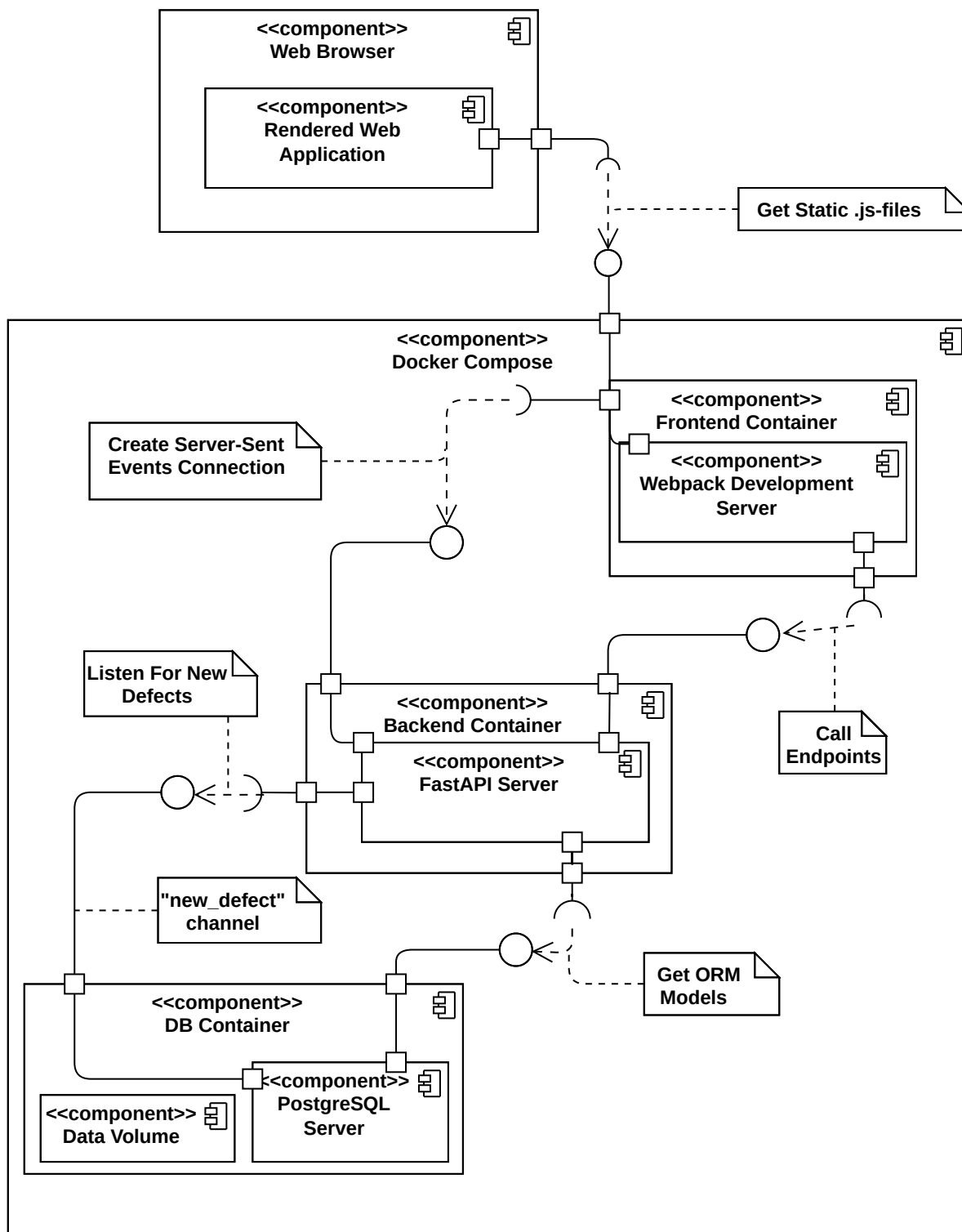


Рис. 12: Диаграмма компонентов всего веб-приложения

4. Апробация

Апробация веб-приложения не проведена.

5. Заключение

В рамках выпускной квалификационной работы были получены следующие результаты:

- проведён обзор дефектов ленты конвейера и методов их обнаружения;
- проведён обзор существующих систем, используемых для дефектоскопии;
- спроектирована архитектура веб-приложения;
- реализовано полноценное веб-приложение;
- написаны модульные тесты для сервиса работы с дефектами;
- проведена контейнеризация веб-приложения.

Ссылка на исходный код — <https://github.com/RuslanBeresnev/Conveyor-Belt-Monitoring-Web-Application> (дата обращения: 24.05.2025).

Список литературы

- [1] Analysis of Devices to Detect Longitudinal Tear on Conveyor Belts / Alexander Zakharov, Boris Geike, Alexander Grigoryev, Alla Zakharova // [E3S Web Conf.](#) — 2020. — Vol. 174. — P. 03006. — URL: <https://doi.org/10.1051/e3sconf/202017403006> (дата обращения: 24 мая 2025 г.).
- [2] Axios. HTTP-клиент для браузера и node.js на основе Promise. — URL: <https://axios-http.com/ru/> (дата обращения: 24 мая 2025 г.).
- [3] Belt Tear Detection for Coal Mining Conveyors / Xiaoqiang Guo, Xinhua Liu, Hao Zhou et al. // [Micromachines.](#) — 2022. — Vol. 13, no. 3. — URL: <https://www.mdpi.com/2072-666X/13/3/449> (дата обращения: 24 мая 2025 г.).
- [4] CBGuard. CBGuard: Conveyor Belt Monitoring. — URL: https://www.cbguard.info/index_html_files/CBGuard%20Brochure%20EN.pdf (дата обращения: 3 декабря 2024 г.).
- [5] Chereisky Artem. BeltSpy Conveyor Belt Inspection and Monitoring System. — URL: <https://www.youtube.com/watch?v=D9cEmZBurNE> (дата обращения: 3 декабря 2024 г.).
- [6] Continental Industry. CONTI MultiProtect: High-performance rip detection system with cord and splice monitoring functionality for conveyor belts. — URL: <https://www.continental-industry.com/en/solutions/conveyor-belt-systems/conveyor-services/digital-solutions/belt-monitoring-systems/products/range-of-products/conti-multiprotect> (дата обращения: 16 ноября 2024 г.).
- [7] Continental Industry. Continental Belt Monitoring System: Conveyor Belt Protection Devices and Inspection Services. — URL: <https://www.continental-industry.com/getmedia/b83f4263-4076-4ab5-a57>

[6-cac7fd4a37b3/Continental-Belt-Monitoring-Systems_en.pdf](#)

(дата обращения: 25 ноября 2024 г.).

- [8] Docker. Develop faster. Run anywhere. — URL: <https://www.docker.com/> (дата обращения: 24 мая 2025 г.).
- [9] Fagus GreCon. GreCon BELTVIEW: High-resolution scanner to determine the surface of conveyor belts. — URL: <https://www.fagus-grecon.com/en/measuring-technology/products/beltview/> (дата обращения: 3 декабря 2024 г.).
- [10] Fenner Dunlop. Conveyor diagnostics and monitoring. — URL: https://fennerdunlopamericas.com/app/uploads/fda-conveyor-diagnostics__2023__digital.pdf (дата обращения: 27 ноября 2024 г.).
- [11] Foundations 4: The Practical Resource for Cleaner, Safer, More Productive Dust Material Control / R. Todd Swinderman, Andrew D. Marti, Larry J. Goldbeck et al. — Neponset, IL, USA : Martin Engineering Company, 2009. — ISBN: 0971712115.
- [12] Gmail API. Python quickstart. — URL: <https://developers.google.com/gmail/api/quickstart/python> (дата обращения: 3 января 2025 г.).
- [13] Google. Material UI is an open-source React component library that implements Google’s Material Design. It’s comprehensive and can be used in production out of the box. — URL: <https://mui.com/material-ui/> (дата обращения: 24 мая 2025 г.).
- [14] Integrative binocular vision detection method based on infrared and visible light fusion for conveyor belts longitudinal tear / Tiezhu Qiao, Lulu Chen, Yusong Pang et al. // *Measurement*. — 2017. — Vol. 110. — P. 192–201. — URL: <https://www.sciencedirect.com/science/article/pii/S0263224117304189> (дата обращения: 24 мая 2025 г.).
- [15] Longitudinal tear early-warning method for conveyor belt based on infrared vision / Yang Yi, Hou Chengcheng, Qiao Tiezhu et al. //

- Measurement. — 2019. — Vol. 147. — P. 106817. — URL: <https://www.sciencedirect.com/science/article/pii/S0263224119306748> (дата обращения: 24 мая 2025 г.).
- [16] Node.js. Node.js is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts. — URL: <https://nodejs.org/en/> (дата обращения: 24 мая 2025 г.).
- [17] NordClan. ML Sense: готовое решение для контроля качества продукции на производствах конвейерного типа на основе машинного зрения и нейросетей. — URL: <https://mlsense.nordclan.com/> (дата обращения: 26 ноября 2024 г.).
- [18] NordClan. ML Sense: определение целостности конвейерной ленты. — URL: <https://mlsense.nordclan.com/opredeleniya-celostnosti-konveyernoy-lenty-kompyuternym-zreniem-i-neyrosetyu> (дата обращения: 26 ноября 2024 г.).
- [19] PostgreSQL Foundation. PostgreSQL: The World's Most Advanced Open Source Relational Database. — URL: <https://www.postgresql.org/> (дата обращения: 3 января 2025 г.).
- [20] PyPI. aiogram downloading stats. — URL: <https://pypistats.org/packages/aiogram> (дата обращения: 3 января 2025 г.).
- [21] PyPI. fastapi downloading stats. — URL: <https://pypistats.org/packages/fastapi> (дата обращения: 3 января 2025 г.).
- [22] PyPI. flask downloading stats. — URL: <https://pypistats.org/packages/flask> (дата обращения: 3 января 2025 г.).
- [23] PyPI. fpdf2 downloading stats. — URL: <https://pypistats.org/packages/fpdf2> (дата обращения: 24 мая 2025 г.).
- [24] PyPI. python-telegram-bot downloading stats. — URL: <https://pypistats.org/packages/python-telegram-bot> (дата обращения: 3 января 2025 г.).

- [25] PyPI. reportlab downloading stats. — URL: <https://pypistats.org/packages/reportlab> (дата обращения: 24 мая 2025 г.).
- [26] PyPI. weasyprint downloading stats. — URL: <https://pypistats.org/packages/weasyprint> (дата обращения: 24 мая 2025 г.).
- [27] React. The library for web and native user interfaces. — URL: <https://react.dev/> (дата обращения: 24 мая 2025 г.).
- [28] ReportLab. ReportLab Docs. — URL: <https://docs.reportlab.com/> (дата обращения: 16 марта 2025 г.).
- [29] Roxon. Roxon HX270 - Wear monitoring. — URL: https://qsgrp.com/infusions/page_documents/documents/roxon_hx270_wear_monitoring.pdf (дата обращения: 3 декабря 2024 г.).
- [30] Roxon. Roxon HX270 for critical conveyors. — URL: https://qsgrp.com/infusions/page_documents/documents/roxon_hx270_for_critical_conveyors.pdf (дата обращения: 16 ноября 2024 г.).
- [31] Sperosens. Sperosens Roxon HX270. — URL: <https://www.youtube.com/watch?v=QKtc4HWS5r4> (дата обращения: 3 декабря 2024 г.).
- [32] Study on nondestructive detection system based on X-ray for wire ropes conveyer belt - art. no. 66220L / Changyun Miao, Boya Shi, Peng Wan, Jie Li // [Proceedings of SPIE - The International Society for Optical Engineering](#). — 2008. — 03. — Vol. 6622.
- [33] Vizorlabs. Программное обеспечение “Система видеоаналитического мониторинга состояния шахтных конвейерных лент, оценки объема и гранулометрического состава пород”. — URL: <https://vizorlabs.ru/upload/iblock/f44/fci4vqdn8iubuhkvybher9poasmu58qg/KONVEYRY-Rukovodstvo-administratora.docx.pdf> (дата обращения: 24 ноября 2024 г.).
- [34] npm. Material UI downloading stats. — URL: <https://www.npmjs.com/package/@mui/material> (дата обращения: 24 мая 2025 г.).

- [35] python-telegram bot. python-telegram-bot: We have made you a wrapper you can't refuse. — URL: <https://python-telegram-bot.org/> (дата обращения: 3 января 2025 г.).
- [36] tiangolo. FastAPI framework, high performance, easy to learn, fast to code, ready for production. — URL: <https://fastapi.tiangolo.com/> (дата обращения: 3 января 2025 г.).
- [37] tiangolo. SQLAlchemy, SQL databases in Python, designed for simplicity, compatibility, and robustness. — URL: <https://sqlmodel.tiangolo.com/> (дата обращения: 3 января 2025 г.).
- [38] wyniemko. Conveyor Belt Monitoring System Manual. — URL: <https://github.com/wyniemko/prod-amazon-belt/blob/main/Conveyor%20Belt%20Monitoring%20System%20Manual.pdf> (дата обращения: 4 декабря 2024 г.).
- [39] wyniemko. prod-amazon-belt. — URL: <https://github.com/wyniemko/prod-amazon-belt/tree/main> (дата обращения: 4 декабря 2024 г.).