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 ИНФОРМАЦИОНЫХ СИСТЕМ**

Выполнил студент 3 курса группы ИС-31

Авдонина Елизавета Максимовна

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

место практики: ООО «Малленом Системс»

Период прохождения:

с «08» июня 2025 года

по «21» июня 2025 года

Руководитель практики от

предприятия, должность:

Южакова Н.В., специалист по кадрам

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

МПРуководитель практики от

техникума: Материкова А.А.

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

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

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc200362279)

[1 ОБЩАЯ ХАРАКТЕРИСТИКА КОМПАНИИ 4](#_Toc200362280)

[1.1 Общая информация (сфера деятельности) 4](#_Toc200362281)

[1.2 Роль информационных систем в работе организации 4](#_Toc200362282)

[1.3 Основные используемые технологии 5](#_Toc200362283)

[2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ 6](#_Toc200362284)

[2.1 Анализ требований и моделирование информационных систем 6](#_Toc200362285)

[2.2 Проектирование информационных систем 6](#_Toc200362286)

[2.3 Разработка информационных систем 7](#_Toc200362287)

[2.4 Тестирование информационных систем 7](#_Toc200362288)

[2.5 Внедрение, эксплуатация и сопровождение информационных систем 8](#_Toc200362289)

[3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ 9](#_Toc200362290)

[ЗАКЛЮЧЕНИЕ 18](#_Toc200362291)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 19](#_Toc200362292)

[ПРИЛОЖЕНИЯ 20](#_Toc200362293)

# ВВЕДЕНИЕ

Производственная практика является важным компонентом образовательного процесса, который способствует качественной подготовке специалистов и их успешной интеграции в профессиональное сообщество.Она помогает студентам познакомиться с культурой работы в компании, включая понимание внутренней структуры, процедур и этики. Практика проходила в ООО «Малленом Системс» с 08.06.2025 по 21.06.2025.

Целью практики было освоение основного вида деятельности по направлению «Проектирование и разработка информационных систем» и формирование соответствующих общих и профессиональных компетенций.

В рамках выполнения поставленной цели, были сформулированы следующие 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.

В процессе практики был изучен ключевой цикл проектирования и разработки информационных систем в компании, а также применены ранее приобретенные знания на практике при создании реальной информационной системы.

# 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

Проектирование и разработка информационных систем — это комплексный процесс создания ИС, включающий в себя этапы анализа, проектирования, кодирования, тестирования и внедрения

## Анализ требований и моделирование информационных систем

Анализ требований и моделирование информационных систем — это начальная фаза, в которой выясняется, зачем вообще нужна система, что она должна делать, для кого и как. Главная цель - понять потребности заказчика и пользователей и зафиксировать их в виде четких и понятных требований. Затем эти требования преобразуются в модели, чтобы визуализировать систему и убедиться, что все поняли ее одинаково.

1. Сбор требований: Интервью с заказчиками, опросы пользователей, изучение существующих документов и систем.
2. Анализ требований: Выявление противоречий, неясностей, приоритезация требований.
3. Спецификация требований: Документ, в котором подробно описаны все требования к системе (функциональные и нефункциональные).
4. Моделирование: Создание диаграмм и схем, которые показывают структуру системы, ее поведение, взаимодействие с другими системами. Примеры: диаграммы вариантов использования (use case), диаграммы классов, диаграммы последовательностей.

## Проектирование информационных систем

На основе требований разрабатывается план создания системы. Определяется архитектура системы, структура базы данных, интерфейсы пользователя, логика работы отдельных модулей.

1. Архитектурное проектирование: Выбор архитектурного стиля (например, клиент-сервер, микросервисы), определение основных компонентов системы и их взаимодействия.
2. Проектирование базы данных: Разработка схемы базы данных, определение таблиц, полей, связей, индексов.
3. Проектирование интерфейса пользователя (UI/UX): Разработка внешнего вида, навигации, логики работы интерфейса, чтобы он был удобным и понятным для пользователей.
4. Проектирование модулей: Определение функциональности, входных и выходных данных для каждого модуля системы.

## Разработка информационных систем

Реализация проекта на практике. Написание кода, создание базы данных, разработка интерфейса, интеграция всех компонентов в единую систему.

1. Написание кода: Программирование модулей и компонентов системы на выбранных языках программирования.
2. Создание базы данных: Реализация спроектированной структуры базы данных.
3. Разработка интерфейса пользователя: Создание графического интерфейса, который соответствует спроектированным макетам.
4. Интеграция: Объединение всех разработанных компонентов в единую работающую систему.

## Тестирование информационных систем

Проверка того, что система работает правильно и соответствует требованиям. Выявление и исправление ошибок.

1. Разработка тестовых сценариев: Создание набора тестов, которые проверяют различные аспекты системы.
2. Модульное тестирование: Проверка отдельных модулей.
3. Интеграционное тестирование: Проверка взаимодействия между модулями.
4. Системное тестирование: Проверка системы в целом.
5. Приемочное тестирование: Проверка системы заказчиком перед внедрением.

## Внедрение, эксплуатация и сопровождение информационных систем

Ввод системы в эксплуатацию, обучение пользователей, обеспечение ее стабильной работы, исправление ошибок и внесение изменений.

1. Развертывание: Установка системы на серверах или в облаке.
2. Миграция данных: Перенос данных из старых систем в новую систему.
3. Обучение пользователей: Проведение тренингов и предоставление инструкций.
4. Техническая поддержка: Решение проблем и вопросов пользователей.
5. Сопровождение: Исправление ошибок, внесение изменений, добавление новых функций.

# ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

Задание 1:

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

Обзор популярных систем управления проектами, основанных на Kanban, с их плюсами и минусами:

Trello:

* Плюсы: Простой в использовании, интуитивно понятный интерфейс, бесплатный тарифный план, интеграция с множеством других инструментов.
* Минусы: Ограниченные возможности для сложных проектов, нет встроенной поддержки для отслеживания времени.

Jira:

* Плюсы: Мощный, настраиваемый, хорошо подходит для крупных команд и сложных проектов, широкие возможности отслеживания и отчетности.
* Минусы: Более сложный в освоении, более дорогой, требует настройки.

Asana:

* Плюсы: Гибкий, подходит для различных типов проектов, хорошие возможности для совместной работы, интеграция с другими инструментами.
* Минусы: Может быть перегружен функциями для небольших проектов.

Microsoft Planner:

* Плюсы: Интегрирован с Microsoft 365, простой в использовании, подходит для небольших команд.
* Минусы: Ограниченные возможности по сравнению с другими системами.

ClickUp:

* Плюсы: Очень настраиваемый, множество функций, бесплатный тарифный план с большим количеством возможностей.
* Минусы: Может быть перегружен функциями, требуются усилия для настройки под конкретные нужды.

GitHub Projects/GitLab Issues:

* Плюсы: Тесно интегрированы с репозиториями кода, идеально подходят для разработки программного обеспечения, бесплатные для публичных проектов.
* Минусы: Ориентированы в основном на разработку, менее удобны для других типов проектов.

YouTrack:

* Плюсы: Разработан специально для команд разработчиков, мощные инструменты отслеживания и отчетности, интеграция с другими инструментами JetBrains.
* Минусы: Ориентирован в основном на разработку, может быть сложным в освоении.

1. Выбрать подходящую систему управления проектами.

Учитывая требования и задачи проекта (подсистема хранения данных для мониторинга износа режущих инструментов, сервис для обработки данных, удаление/добавление инструментов), я выбрала Jira, так как она обеспечивает высокую степень настраиваемости, необходимую для управления сложной системой с различными типами данных и взаимодействиями. Ее возможности отслеживания и отчетности будут полезны для мониторинга прогресса разработки и выявления проблем.

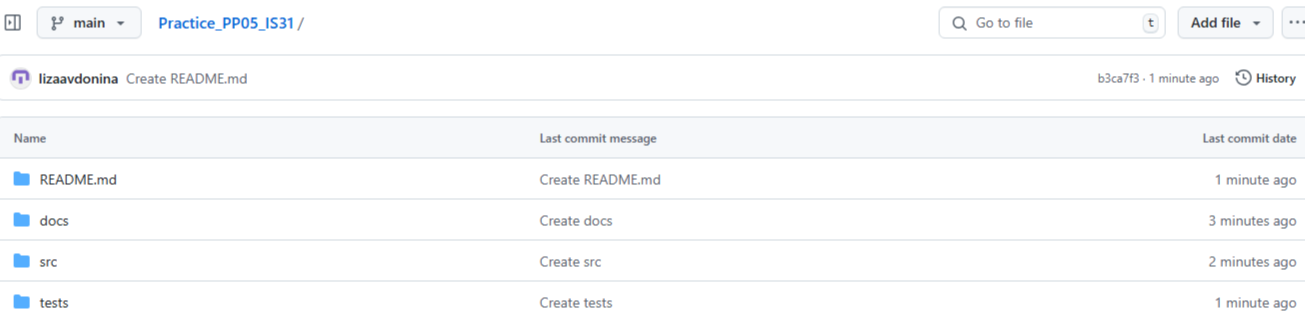
1. После выбора систем управления проектами, необходимо реализовать репозиторий с проектом будущей разрабатываемой подсистемы. 

Рисунок 1 – репозиторий «Practice\_PP05\_IS31»

Задание 2:

1. Необходимо определится с направлением разработки

Моя тема: Подсистема хранения данных для мониторинга износа режущих инструментов. Состоит из сервиса, который обрабатывает данные о состоянии инструментов, может удалять, добавлять их.

1. Для выбранной предметной области составить план работ, составить задачи и ввести их на доску канбан (из задания 1). Необходимо детально рассчитать сроки и количество задач, до окончания практики.

**Этапы и задачи:**

Неделя 1: Анализ, выбор технологий, проектирование базы данных

Backlog:

* Анализ предметной области и требований (1 день).
* Выбор технологий (0.5 дня).
* Разработка ER-диаграммы базы данных (1 день).
* Определение структуры таблиц, полей, связей (0,5 день).
* Написание скриптов для создания базы данных (0.5 дня).
* Создание репозитория проекта (0.5 дня).
* Настройка Kanban-доски.

To Do: Выберите задачи из Backlog и переместите сюда.

Разработка API и моделей данных

Backlog:

* Создание моделей данных (Entity Framework) (0,5 дня).
* Разработка API для добавления, удаления, обновления инструментов (1 день).
* Реализация логирования (Serilog) (0.5 дня).
* Написание unit-тестов для API (1 день).

To Do: Выберите задачи из Backlog и переместите сюда.

Неделя 2: Разработка логики сервиса и интеграция с базой данных

Backlog:

* Реализация логики сервиса для обработки данных о состоянии инструментов (1 день).
* Интеграция сервиса с базой данных (1 день).
* Написание интеграционных тестов (1 день).

To Do: Выберите задачи из Backlog и переместите сюда.

Документация и тестирование

Backlog:

* Написание технического задания (1 день).
* Подготовка руководства пользователя (1 день).
* Подготовка руководства администратора (1 день).
* Разработка UML-диаграмм (варианты использования, последовательности, компоненты, пакеты, деятельности) (1 день).
* Подготовка отчета.

To Do: Выберите задачи из Backlog и переместите сюда.

1. Составить документацию для выбранной подсистемы:

a. Техническое задание.

1. Введение

Настоящее техническое задание (ТЗ) определяет требования к разработке подсистемы хранения данных для мониторинга износа режущих инструментов. Подсистема является частью общей системы мониторинга, предназначенной для сбора, обработки, хранения и анализа данных о состоянии режущих инструментов с целью оптимизации производственных процессов, повышения эффективности использования инструментов и снижения затрат на их замену.

2. Цели и задачи

Цель: Создание надежной, масштабируемой и эффективной подсистемы для хранения данных о состоянии режущих инструментов.

Задачи:

* Обеспечение возможности хранения данных о различных типах режущих инструментов.
* Предоставление API для добавления, удаления, обновления и получения данных об инструментах.
* Обеспечение высокой доступности и отказоустойчивости.
* Обеспечение безопасности данных.
* Оптимизация хранения данных для повышения производительности запросов.
* Интеграция с существующей системой мониторинга (если применимо).
* Возможность масштабирования для поддержки растущего объема данных и количества инструментов.

3. Требования к функциональности

3.1. Хранение данных:

Подсистема должна обеспечивать хранение следующих данных о каждом режущем инструменте:

* Идентификатор инструмента (ID): Уникальный идентификатор инструмента.
* Тип инструмента: (например, фреза, сверло, резец и т.д.).
* Производитель: Наименование производителя инструмента.
* Модель: Модель инструмента.
* Материал: Материал, из которого изготовлен инструмент.
* Дата установки: Дата установки инструмента на станок.
* Текущее значение износа: Числовое значение, характеризующее степень износа инструмента (например, в процентах, миллиметрах и т.д.).
* История значений износа: Временной ряд значений износа, зафиксированных в определенные моменты времени.
* Тип износа: (например, абразивный, адгезионный, усталостный и т.д.).
* Состояние инструмента: (например, в работе, на хранении, требует замены, списан).
* Станок: ID станка, на котором установлен инструмент.
* Информация о заточке**:** (Дата, количество заточек, параметры заточки).

3.2. API:

Подсистема должна предоставлять API для выполнения следующих операций:

* Создание нового инструмента: Добавление информации о новом инструменте в систему.
* Получение информации об инструменте: Получение информации об инструменте по его ID.
* Обновление информации об инструменте: Обновление информации об инструменте (например, текущего значения износа, состояния и т.д.).
* Удаление инструмента: Удаление информации об инструменте из системы.
* Получение списка инструментов: Получение списка инструментов, удовлетворяющих заданным критериям (например, по типу, состоянию, станку и т.д.).
* Получение истории износа инструмента: Получение временного ряда значений износа для заданного инструмента.

3.3. Требования к безопасности:

* Авторизация и аутентификация: Доступ к API должен быть защищен авторизацией и аутентификацией.
* Шифрование данных: Конфиденциальные данные должны быть зашифрованы при хранении и передаче.
* Защита от несанкционированного доступа: Должны быть предусмотрены меры защиты от несанкционированного доступа к данным.

3.4. Требования к надежности:

* Отказоустойчивость: Подсистема должна быть отказоустойчивой и обеспечивать минимальное время простоя в случае сбоев.
* Резервное копирование: Должно быть предусмотрено резервное копирование данных с возможностью восстановления в случае необходимости.
* Мониторинг: Должны быть предусмотрены механизмы мониторинга состояния подсистемы и оповещения о возникающих проблемах.

4. Требования к реализации

Технологии (C#):

* Платформа: .NET 9.0.5 (C#)
* Сервис: ASP.NET Core Web API
* База данных: MS SQL Server
* ORM: Entity Framework Core
* API: REST API Логирование: Serilog

5. Требования к интеграции

* Подсистема должна легко интегрироваться с существующими компонентами системы мониторинга.
* Интеграция должна осуществляться через API.
* Необходимо предоставить примеры интеграции.

6. Тестирование

Необходимо провести всестороннее тестирование подсистемы, включая:

* Функциональное тестирование.
* Тестирование производительности.
* Тестирование безопасности.
* Тестирование надежности.
* Интеграционное тестирование.

7. Приемочные испытания

* Приемочные испытания должны проводиться на основе разработанных тестовых сценариев.
* Критерии приемки должны быть четко определены и задокументированы.

8. Документация

В процессе разработки необходимо подготовить следующую документацию:

* Техническое задание.
* Архитектурный проект.
* Схема базы данных.
* Описание API.
* Руководство пользователя.
* Руководство администратора.
* Результаты тестирования.

9. Сроки разработки

2 недели

b. Подготовить руководство пользователя и администратор

1. Введение

Настоящее руководство пользователя и администратора предназначено для предоставления информации о настройке, управлении и использовании подсистемы хранения данных для мониторинга износа режущих инструментов. Подсистема состоит из сервиса, который обрабатывает данные о состоянии инструментов, предоставляет API для добавления, удаления и обновления информации об инструментах.

2. Целевая аудитория

* Пользователи: Операторы станков, инженеры-технологи, специалисты по техническому обслуживанию, использующие API Подсистемы для получения данных об инструментах и внесения изменений.
* Администраторы: Системные администраторы, отвечающие за установку, настройку, обслуживание и мониторинг Подсистемы.

3. Обзор системы

Подсистема хранения данных предназначена для централизованного хранения и управления информацией о режущих инструментах, используемых в производственном процессе. Подсистема позволяет отслеживать состояние инструментов, прогнозировать износ, планировать замену и оптимизировать использование инструментов.

4. Руководство пользователя

4.1. Доступ к API:

* Подсистема предоставляет API для взаимодействия с другими компонентами системы мониторинга.
* Доступ к API осуществляется по протоколу HTTP/HTTPS.
* Для аутентификации и авторизации используется OAuth 2.0.

4.2. Основные операции API:

* Получение информации об инструменте (GET /tools/{id})
* Создание нового инструмента (POST /tools)
* Обновление информации об инструменте (PUT /tools/{id})
* Удаление инструмента (DELETE /tools/{id})
* Поиск инструментов (GET /tools?type={type}&status={status})

4.3. Обработка ошибок:

В случае возникновения ошибок API возвращает код ошибки HTTP и сообщение об ошибке в формате JSON.

5. Руководство администратора

5.1. Установка подсистемы:

Требования к оборудованию:

* Указать минимальные и рекомендуемые требования к серверу.

Требования к программному обеспечению:

* Указать необходимые версии операционной системы, базы данных, веб-сервера.

Процесс установки:

* Подробно описать шаги по установке и настройке Подсистемы, включая установку необходимых пакетов, настройку базы данных и веб-сервера.

5.2. Обслуживание и мониторинг

Основные параметры для мониторинга:

* CPU Usage.
* Memory Usage.
* Disk Space Usage.
* Network Traffic.

Резервное копирование:

* Необходимо регулярно создавать резервные копии базы данных и конфигурационных файлов.
* Рекомендуется использовать автоматическое резервное копирование с помощью cron.

5.3. Устранение неполадок:

Ошибка подключения к базе данных:

* Проверьте правильность строки подключения к базе данных в файле конфигурации.
* Проверьте, запущен ли сервер базы данных.
* Проверьте, доступен ли сервер базы данных с сервера, на котором работает подсистема.

Ошибка аутентификации:

* Проверьте правильность предоставленных учетных данных.
* Проверьте, правильно ли настроена аутентификация в файле конфигурации.

6. Безопасность

* Регулярно обновляйте программное обеспечение подсистемы и операционной системы.
* Используйте надежные пароли для доступа к подсистеме и базе данных.
* Настройте брандмауэр для защиты от несанкционированного доступа.
* Регулярно создавайте резервные копии данных.
* Обучите пользователей правилам безопасности.

c. ER диаграмму, описать ее.

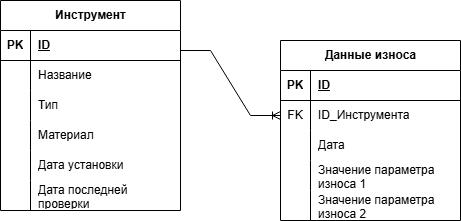


Рисунок 2 - ER диаграмма

Общая структура:

Диаграмма представляет собой модель данных, описывающую инструменты и данные об их износе. Она состоит из двух сущностей: Инструмент (Tool) и ДанныеИзноса (WearData), связанных отношением "один ко многим".

Сущности и атрибуты:

Сущность "Инструмент" (Tool):

Представляет собой физический инструмент.

Содержит следующие атрибуты:

* ID: Уникальный идентификатор инструмента
* Название: Название инструмента.
* Тип: Тип инструмента (например, сверло, фреза, токарный резец).
* Материал: Материал, из которого изготовлен инструмент.
* Дата установки: Дата установки инструмента в оборудование или начала его использования.
* Дата последней проверки: Дата последней проверки состояния инструмента.

Сущность "ДанныеИзноса" (WearData):

Представляет собой запись данных об износе конкретного инструмента.

Содержит следующие атрибуты:

* ID: Уникальный идентификатор записи об износе (вероятно, первичный ключ).
* ИнструментID: Идентификатор инструмента, к которому относится запись износа (внешний ключ, ссылающийся на ID в таблице Инструмент).
* Дата: Дата, когда были сняты данные об износе.
* Значение параметра износа 1: Численное значение первого параметра, характеризующего износ (например, глубина износа, ширина износа).
* Значение параметра износа 2: Численное значение второго параметра, характеризующего износ (например, изменение шероховатости поверхности).

Связь:

* Тип связи: "Один ко многим" (One-to-Many).
* Направление связи: Один инструмент может иметь много записей в таблице ДанныеИзноса. Каждая запись в ДанныеИзноса относится только к одному конкретному инструменту.
* Значение связи: Эта связь позволяет отслеживать историю износа каждого инструмента, регистрируя параметры износа через определенные промежутки времени.

В целом, эта ER-диаграмма моделирует систему, которая отслеживает информацию об инструментах и ​​собирает данные об их износе, что позволяет анализировать состояние инструментов, прогнозировать их срок службы и оптимизировать процессы обслуживания.

d. UML, из них:

i. Диаграмма вариантов использования

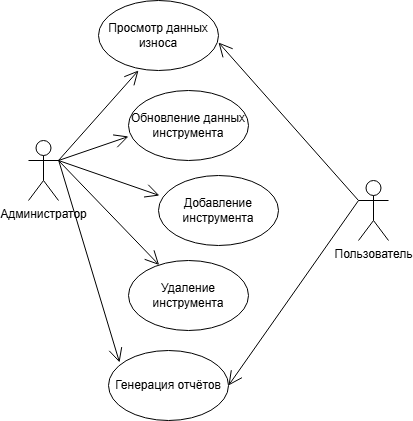


Рисунок 3 – диаграмма вариантов использования

ii. Диаграмма последовательностей

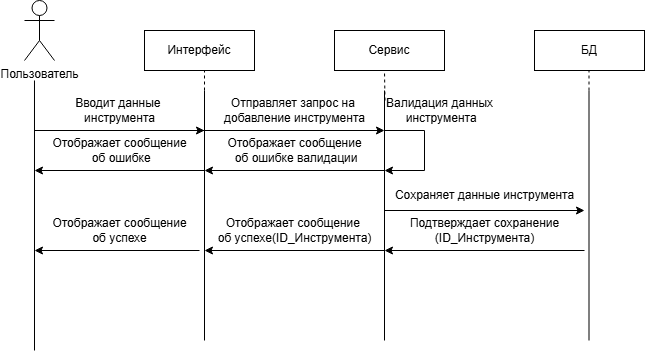
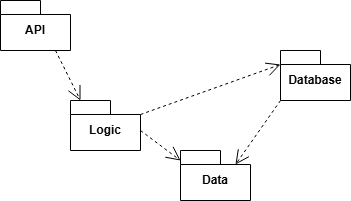


Рисунок 4 – диаграмма последовательностей

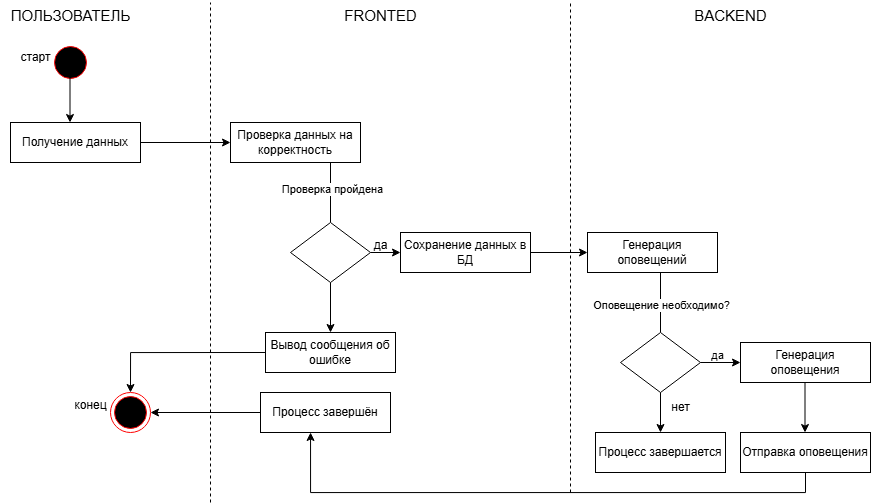
iii. Диаграмма компонентов



iv. Диаграмма пакетов



v. Диаграмма деятельности



# 

# ЗАКЛЮЧЕНИЕ

э

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

# ПРИЛОЖЕНИЯ