

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-\_\_\_\_\_

\_\_\_\_\_

подпись \_\_\_\_\_

место практики \_\_\_\_\_  
наименование юридического лица, ФИО ИП

Период прохождения:

с «\_\_\_» \_\_\_\_\_ 2025 г.

по «\_\_\_» \_\_\_\_\_ 2025 г.

Руководитель практики от

техникума: Материкова А.А.

\_\_\_\_\_

Оценка: \_\_\_\_\_

«\_\_\_» \_\_\_\_\_ 2025 года

Руководитель практики от

предприятия

должность \_\_\_\_\_

\_\_\_\_\_

подпись \_\_\_\_\_

МП

г. Череповец

2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
• Цель и задачи практики. ....	3
• Краткое описание организации, где проходила практика. ....	3
• Сроки и место прохождения. ....	3
ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ .....	3
• Общая информация. ....	3
• Роль информационных систем в работе организации. ....	4
• Основные используемые технологии. ....	4
2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ .....	4
2.1. Анализ требований информационных систем. ....	4
2.2. Проектирование информационных систем. ....	5
2.3. Разработка информационных систем. ....	5
2.4. Тестирование информационных систем. ....	6
2.5. Внедрение, эксплуатация и сопровождение информационных систем. ....	6
3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ .....	6
3.1. Задание 1. ....	6
3.2. Задание 2. ....	7
ЗАКЛЮЧЕНИЕ.....	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	16
ПРИЛОЖЕНИЕ .....	17

## ВВЕДЕНИЕ

- Цель и задачи практики.

Производственная практика нацелена на получение практического опыта в управлении процессами разработки информационных систем, разработке документации по эксплуатации информационной системы.

- разработке документации по эксплуатации информационной системы;
- Осуществлять постановку задачи по обработке информации.
- Выполнять анализ предметной области.
- Работать с инструментальными средствами обработки информации.

- Краткое описание организации, где проходила практика.

Малленом Системс – Российская компания в области разработки и внедрения систем компьютерного зрения, промышленной видео аналитики на основе технологий машинного зрения и искусственного интеллекта (машинное обучение, нейронные сети глубокого обучения) и интеллектуальной обработки данных.

- Сроки и место прохождения.

Срок прохождения практики с 25.05.25 по 7.06.25, ООО “Малленом Системс” практика проходила дистанционно.

## ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ

- Общая информация.

Малленом Системс была создана в 2011 году на базе команды ученых и программистов Санкт-Петербургского политехнического университета Петра Великого. Сегодня в компании более 100 сотрудников. Глубокие компетенции в сфере машинного зрения и большой опыт успешной реализации проектов на промышленных предприятиях позволяет успешно решать большой спектр задач в различных отраслях. В Центре исследований и разработки интеллектуальных систем ведется работа по созданию новых решений и развитию продуктов компании.

В основе разработанных в компании систем лежат как собственные решения на базе нейронных сетей и детерминированных алгоритмов анализа изображений, так и алгоритмы от мирового лидера в области машинного зрения – компании Cognex.

- Роль информационных систем в работе организации.

Информационные системы (ИС) играют ключевую роль в деятельности предприятия, обеспечивая автоматизацию, управление данными и поддержку принятия решений.

- Основные используемые технологии.
  - Платформа: .NET 9.0.5 (C#)
  - Платформа разработки: ASP.NET.
  - Реализации базы данных: Postgres или MS SQL.SERVER
  - Взаимодействия с базой данных: EntityFramework
  - Общшения с сервисом: REST API или SignalR.
  - Логирование: Serilog.
  - Система контроля версий: GitHub, Git, GitLab.

## 2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ

### 2.1. Анализ требований информационных систем.

Анализ требований является фундаментальным этапом разработки информационных систем, определяющим успех всего проекта. Этот процесс включает сбор, систематизацию и документирование функциональных и нефункциональных требований, а также выявление потенциальных противоречий между интересами заказчиков, пользователей и разработчиков.

Моделирование играет критическую роль в анализе требований, позволяя визуализировать процессы и структуры системы. Согласно методологиям, описанным в источниках, используются два типа моделей: предметные (физические прототипы) и знаковые (диаграммы и схемы).

Например, диаграммы потоков данных помогают отобразить взаимодействие между компонентами системы, а ER-модели — структуру базы данных.

## 2.2. Проектирование информационных систем.

Проектирование ИС требует выбора между функционально-ориентированным и объектно-ориентированным подходами. Функционально-ориентированный подход фокусируется на декомпозиции системы на иерархические функции, что упрощает управление сложными процессами, но может привести к дублированию данных.

Важным аспектом проектирования является определение архитектуры системы. Например, при использовании клиент-серверной модели данные обрабатываются на сервере, что снижает нагрузку на клиентские устройства. Другой вариант многоуровневая архитектура (например, трёхзвенная), где презентационный, бизнес-логики и уровень данных разделены, что повышает безопасность и масштабируемость. Выбор технологий (СУБД, языки программирования) также влияет на архитектуру: реляционные базы данных подходят для структурированных данных, а NoSQL для неструктурированных.

## 2.3. Разработка информационных систем.

На этапе разработки ключевым вопросом является выбор методологии. Waterfall (каскадная модель) предполагает последовательное выполнение этапов, что подходит для проектов с четкими требованиями, но lacks гибкости. В contrast, Agile-методы (Scrum, Kanban) позволяют адаптироваться к изменениям через итеративную разработку и частые релизы.

Инструменты разработки включают системы контроля версий (Git), среды разработки (IDE) и фреймворки. Например, использование Git позволяет командам параллельно работать над разными ветками кода, минимизируя конфликты. Для веб-приложений популярны фреймворки типа React или Angular, обеспечивающие модульность и повторное использование компонентов. Важную роль играет документация: листинги кода, API-

спецификации и руководства пользователя, которые упрощают сопровождение системы.

#### 2.4. Тестирование информационных систем.

Тестирование направлено на обеспечение соответствия системы требованиям и выявление дефектов. Модульное тестирование проверяет отдельные компоненты (например, функции), тогда как интеграционное тестирование оценивает взаимодействие модулей.

Нагрузочное тестирование критично для оценки производительности: оно имитирует пиковые нагрузки, чтобы определить пределы пропускной способности. Тестирование безопасности включает проверку на уязвимости (SQL-инъекции, XSS), используя инструменты вроде OWASP ZAP. Автоматизация тестирования через CI/CD-конвейеры (Jenkins, GitLab CI) ускоряет выявление ошибок и сокращает время выхода на рынок.

#### 2.5. Внедрение, эксплуатация и сопровождение информационных систем.

Внедрение ИС включает установку, настройку и обучение пользователей. Важно обеспечить плавный переход от старой системы к новой, используя стратегии вроде параллельного запуска, где обе системы работают одновременно.

Эксплуатация требует мониторинга производительности и устранения инцидентов. Инструменты вроде Prometheus или Grafana помогают отслеживать метрики (время отклика, загрузку CPU). Сопровождение включает исправление ошибок, обновления и масштабирование. Регулярное резервное копирование и аварийное восстановление (Disaster Recovery) минимизируют риски потери данных.

### 3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

#### 3.1. Задание 1.

Было проведено изучение и анализ существующих систем управления проектами (СУП), была выбрана СУП shtab, он представляет из себя веб-сервис, который позволяет управлять проектами, самая важная функция это

создание доски канбан в которой была спланирована вся работа от начала до конца практики. Для удобства хранения всех документов и файлов для практики был создан GitHub репозиторий.

Скриншот сайта выбранной системы управления проектами представлен на рисунке 1.

– Краткое описание shtab:

Это российская система управления проектами и задачами с встроенным трекером времени, разработанная компанией Stik Pro. Она предназначена для координации работы команд и компаний любого масштаба от малого бизнеса до крупных корпораций и помогает организовать рабочие процессы.

Shtab позиционируется как российский аналог таких систем, как Trello, Notion и Asana.

– Краткое описание Канбан:

Это визуальный инструмент управления проектами и задачами, который помогает командам наглядно отслеживать процесс работы и повышать её эффективность. Она представляет собой доску, разделённую на колонки, каждая из которых соответствует определённому этапу рабочего процесса.

Канбан-доски бывают физическими и цифровыми онлайн-сервисами. Структура доски канбан представлена на рисунке 2.

### 3.2. Задание 2.

Необходимо было выбрать направление разработки.

Была выбрана “Подсистема хранения данных для системы обнаружения трещин в стеклянных изделиях”, после была составлена доска канбан для данного этапа практики, необходимо было составить техническое задание (ТЗ) и подготовить руководство пользователя и администратора.

#### Техническое задание

На разработку подсистему хранения данных для системы обнаружения трещин в стеклянных изделиях

#### 1. Введение

Подсистема хранения данных предназначена для обработки, хранения и управления информацией о трещинах, обнаруженных в стеклянных изделиях. Система должна обеспечивать надежное хранение данных, поддержку операций добавления, удаления и обновления информации, а также предоставлять интерфейс для взаимодействия с другими компонентами системы обнаружения трещин.

## 2. Цели и задачи

- Создать сервис для хранения и управления данными о трещинах.
- Обеспечить возможность добавления, удаления и редактирования данных.
- Обеспечить быстрый и надежный доступ к информации.
- Реализовать API для интеграции с системой обнаружения трещин и внешними приложениями.
- Обеспечить безопасность и целостность данных.
- Обеспечить масштабируемость и отказоустойчивость подсистемы.

## 3. Функциональные требования

- Прием и сохранение данных о трещинах (тип, размер, координаты, время обнаружения, степень опасности и др.).
- Поддержка CRUD-операций (создание, чтение, обновление, удаление).
- Поиск и фильтрация данных по различным параметрам.
- Ведение истории изменений данных.
- Логирование всех операций с данными.

## 4. Нефункциональные требования

- Надежность: устойчивость к сбоям, средняя наработка до отказа не менее 7500 часов.
- Безопасность: аутентификация и авторизация пользователей, защита данных.
- Производительность: минимальные задержки при обработке запросов.



- Масштабируемость: возможность расширения объема данных и числа пользователей.

- Совместимость с платформой .NET 9.0.5 и технологиями ASP.NET, Entity Framework.

- Использование PostgreSQL или MS SQL Server в качестве СУБД.

- Логирование с использованием Serilog.

## 5. Архитектура и технологии

- Язык программирования: C#.

- Платформа: .NET 9.0.5.

- Веб-сервис: ASP.NET.

- База данных: PostgreSQL.

- ORM: Entity Framework.

- Взаимодействие через REST API.

- Логирование: Serilog.

## 6. Структура данных

- Идентификатор трещины.

- Тип трещины (поверхностная, сквозная и т.п.).

- Геометрические параметры (длина, ширина, глубина).

- Координаты расположения на изделии.

## 7. Требования к безопасности

- Контроль доступа на основе ролей.

- Шифрование данных при передаче.

- Защита от несанкционированного доступа.

- Регулярное резервное копирование.

## 8. План работ и сроки

- Анализ требований и проектирование – 1 неделя.

- Проектирование базы данных и API – 1 неделя.

- Разработка сервиса и интеграция с базой данных – 2 недели.

- Тестирование и отладка – 1 недели.

- Подготовка документации и внедрение – 1 неделя.

## 9. Документация

- Техническое задание.
- Руководство пользователя и администратора.
- ER-диаграмма базы данных.
- UML-диаграммы: варианты использования, последовательностей, компонентов, пакетов, деятельности.

### Руководство пользователя и администратора.

Подсистемы хранения данных для системы обнаружения трещин в  
стеклянных изделиях.

#### Руководство пользователя:

##### 1. Введение

Руководство предназначено для пользователей подсистемы хранения данных, которая обеспечивает управление информацией о трещинах в стеклянных изделиях. Подсистема позволяет добавлять, просматривать, редактировать и удалять данные о трещинах.

#### Основные функции пользователя

- Добавление данных о трещинах: ввод информации о новом дефекте (тип, размеры, координаты, дата обнаружения, степень опасности).
- Просмотр данных: поиск и фильтрация трещин по различным параметрам (дата, тип, изделие).
- Редактирование данных: обновление информации о существующих трещинах.
- Удаление данных: удаление устаревших или ошибочных записей.
- Экспорт данных: возможность выгрузки информации для дальнейшего анализа.

##### 2. Вход в систему

- Для доступа к системе пользователь должен пройти аутентификацию, используя предоставленные учетные данные.
- В случае утери пароля необходимо обратиться к администратору.

##### 3. Использование интерфейса

- В главном меню выберите нужный раздел (например, «Добавить трещину», «Просмотр данных»).

- Для добавления трещины заполните форму с обязательными полями.

- Для поиска используйте фильтры по дате, типу трещины и другим параметрам.

- Для редактирования или удаления выберите нужную запись и используйте соответствующие кнопки.

#### 4. Работа с API (для интеграции)

- Система предоставляет REST API для выполнения CRUD-операций с данными.

- Для работы с API необходим токен аутентификации.

- Документация по API включает описание эндпоинтов, форматов запросов и ответов.

#### 5. Рекомендации по безопасности

- Не передавайте учетные данные третьим лицам.

- Завершайте сеанс работы после окончания работы с системой.

- Сообщайте администратору о любых подозрительных действиях.

Руководство администратора:

##### 1. Введение

Руководство, предназначенное для администраторов, которые отвечают за настройку, поддержку и безопасность подсистемы хранения данных, а также за управление учетными записями пользователей.

##### 2. Установка и настройка

- Установка сервиса на сервере с платформой .NET 9.0.5.

- Настройка базы данных (PostgreSQL).

- Конфигурация подключения Entity Framework к базе данных.

- Настройка логирования через Serilog.

- Обеспечение резервного копирования данных.

##### 3. Управление пользователями

- Создание, изменение и удаление учетных записей пользователей.

- Назначение ролей и прав доступа (например, пользователь, администратор).

- Сброс паролей и управление политиками безопасности.

#### 4. Мониторинг и обслуживание

- Мониторинг работы сервиса и базы данных.

- Анализ логов для выявления ошибок и подозрительной активности.

- Проведение регулярного резервного копирования и тестирование восстановления.

- Обновление программного обеспечения и базы данных.

#### 5. Безопасность

- Настройка аутентификации и авторизации.

- Обеспечение защиты данных от несанкционированного доступа.

- Контроль доступа на уровне ролей.

- Реагирование на инциденты безопасности.

#### 6. Работа с API

- Управление доступом к REST API.

- Контроль использования API и ограничение доступа при необходимости.

#### 7. Техническая поддержка пользователей

- Помощь пользователям при возникновении проблем.

- Обучение пользователей работе с системой.

- Ведение документации по инцидентам и их решению.

После составления документов нужно составить ER – диаграмму и 5 UML диаграмм.

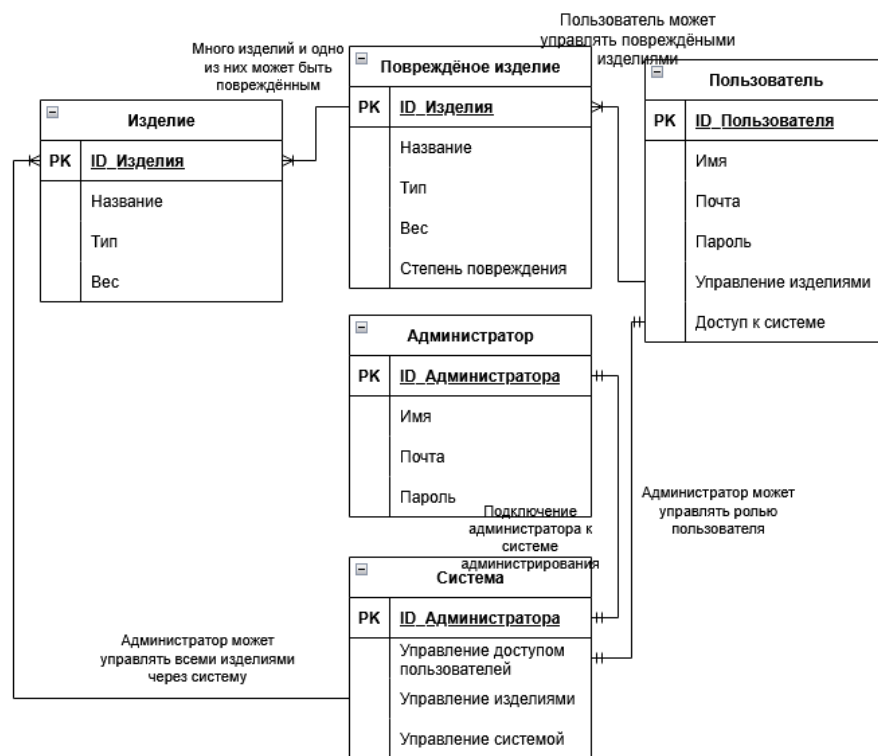


Рисунок 1 – Вид ER диаграммы

1. Диаграмма вариантов использования.

Внешний вид диаграммы вариантов использования рисунок 6.

2. Диаграмма последовательностей.

Внешний вид диаграммы последовательности рисунок 7.

3. Диаграмма компонентов.

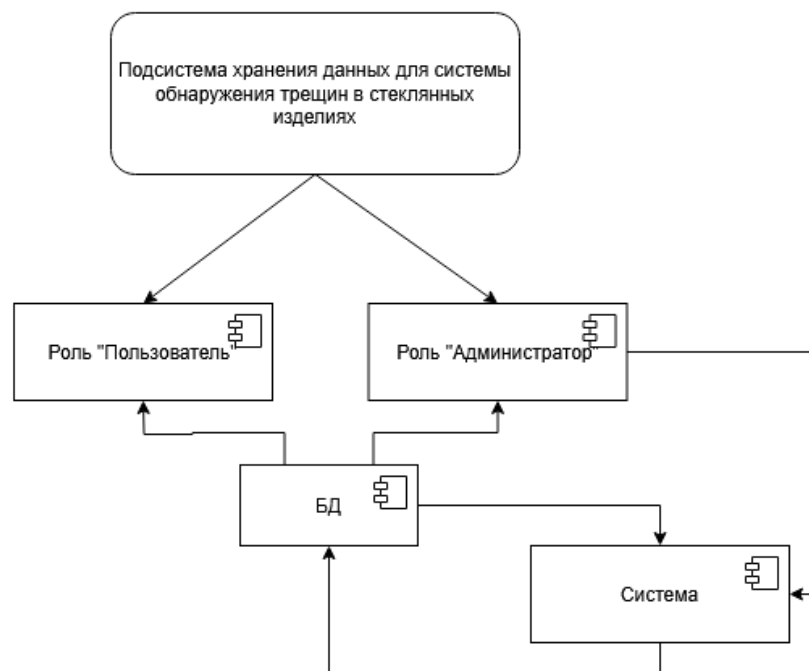


Рисунок 2 – Вид диаграммы компонентов

4. Диаграмма пакетов.

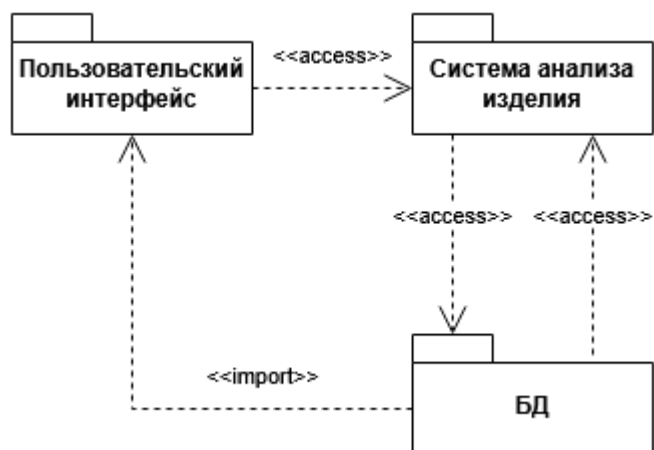


Рисунок 3 – Вид диаграммы пакетов

5. Диаграмма деятельности.

Внешний вид диаграммы деятельности рисунок 8.

## ЗАКЛЮЧЕНИЕ

Практика в ООО “Малленом Системс” позволила мне улучшить навык проектирования и разработки информационных систем, а также получить много нового опыта работы с информационными системами, рассмотреть их с разных сторон и разработки разного вида диаграмм и документации.

Работа с системами управления проектами показала мне то, как работают компании, было интересно поставить себе цель и выполнить её к нужному сроку.

Опыт, полученный на практике, очень значим для понимания того, как устроены многие процессы при проектировании ИС в реальных условиях.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Система управление проектами [Электронный ресурс]/ - Режим доступа: <https://shtab.app/>
2. Графический редактор диаграмм [Электронный ресурс]/ Режим доступа: <https://www.drawio.com/>
3. Методы моделирования и модели разработки ИС [Электронный ресурс]/ - Режим доступа: <https://inftis.narod.ru/pis/pis-p3-1.htm>
4. Тестирование информационных систем [Электронный ресурс]/ - Режим доступа: <https://at-consulting.ru/testirovanie-informacionnyh-sistem>
5. Разработка и внедрение информационных систем [Электронный ресурс]/ - Режим доступа: <https://infocom-s.ru/sozдание-informacionnyh-sistem>
6. Основные этапы проектирования ИС [Электронный ресурс]/ - Режим доступа: <https://scilead.ru/article/714-osnovnie-etapi-proektirovaniya-informatsionnik>
7. Разработка требований к ИС [Электронный ресурс]/ - Режим доступа: <https://searchinform.ru/services/outsource-ib/zaschita-informatsii/razrabotka-trebovanij-po-ib/razrabotka-trebovanij-k-informatsionnoj-sisteme/>
8. Примеры UML диаграмм [Электронный ресурс]/ - режим доступа: <https://practicum.yandex.ru/blog/uml-diagrammy/>
9. Методы анализа требований [Электронный ресурс]/ - Режим доступа: [https://logrocon.ru/news/requirements\\_analysis2](https://logrocon.ru/news/requirements_analysis2)
10. Методы проектирования ИС [Электронный ресурс]/ - Режим доступа: <https://studfile.net/preview/9057964/page:87/>



## ПРИЛОЖЕНИЕ

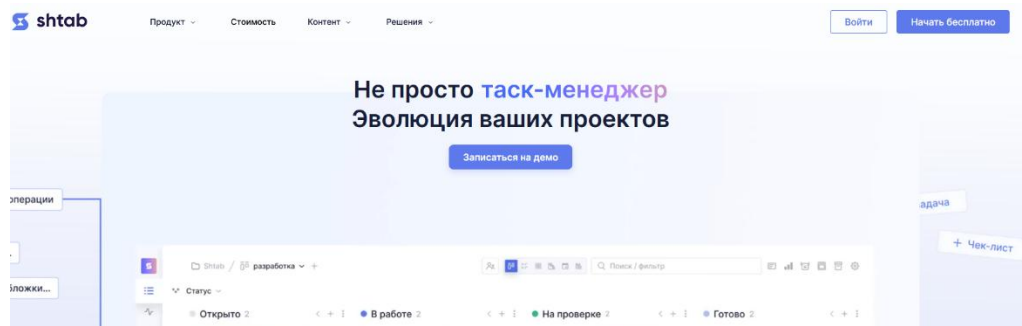


Рисунок 4 – Скриншот сайта выбранной системы управления проектами

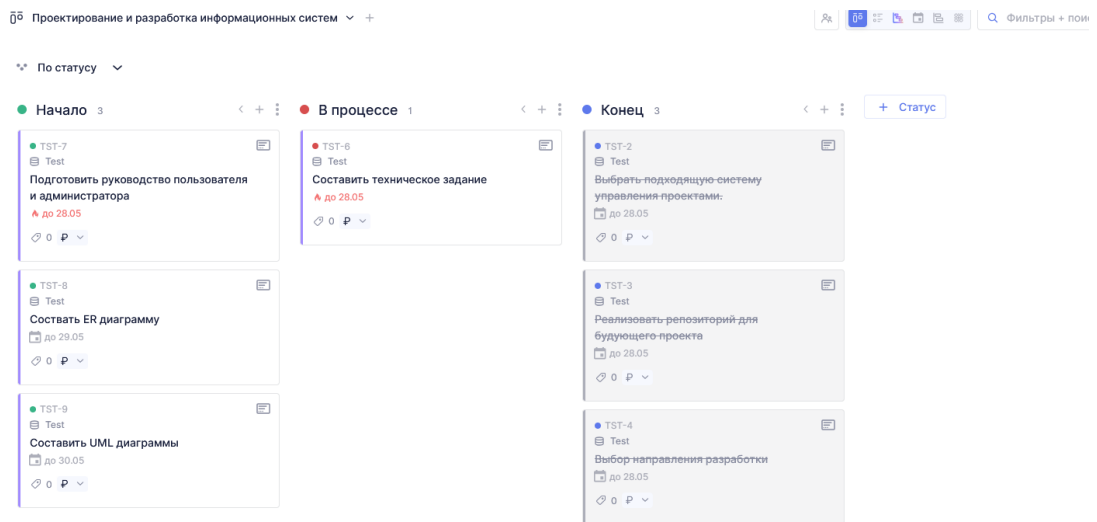


Рисунок 5 – Доска канбан

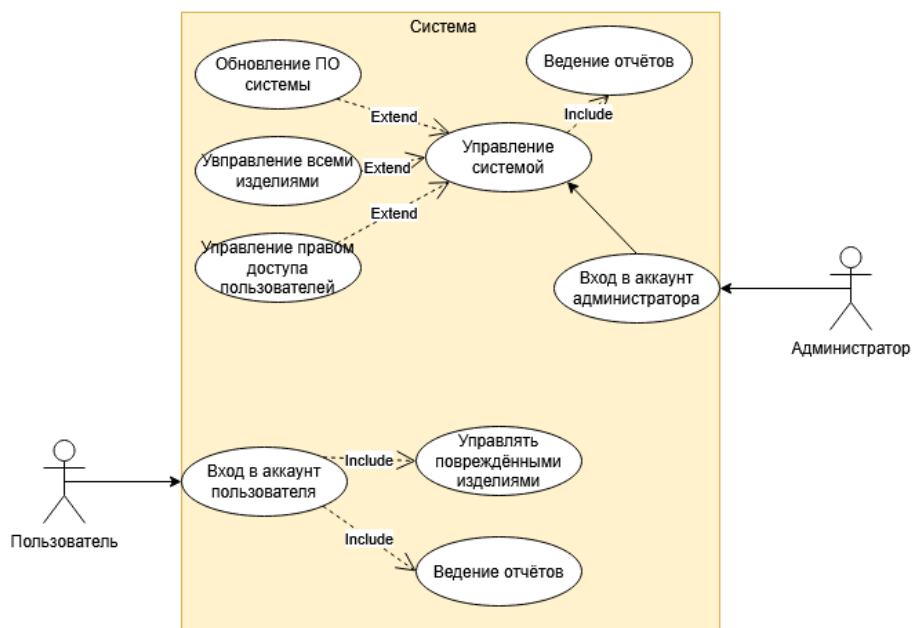


Рисунок 6 – Диаграмма вариантов использования

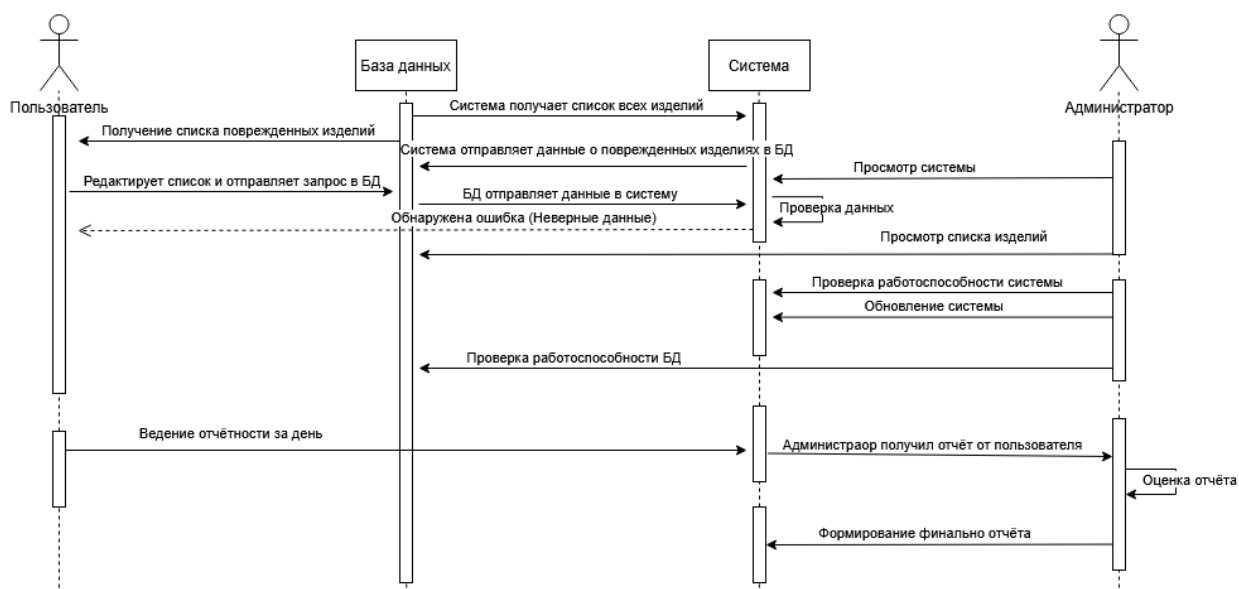


Рисунок 7 – Диаграмма последовательности

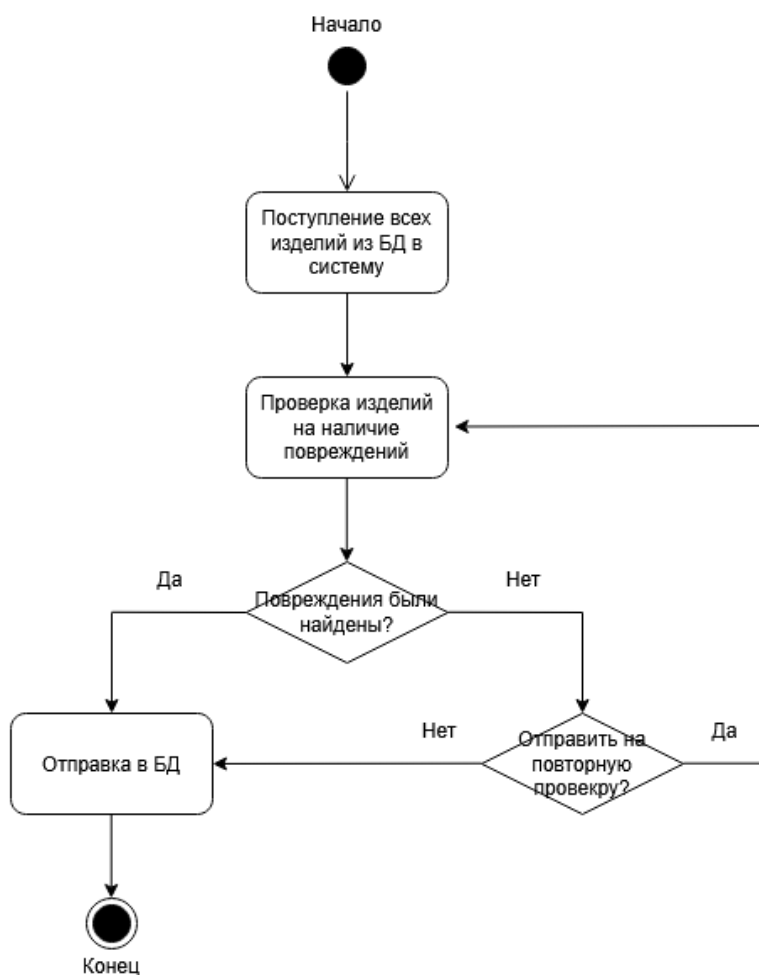


Рисунок 8 – Диаграмма деятельности