бюджетное профессиональное образовательное учреждение Вологодской области

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

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

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

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

Выполнил студент 3 курса группы ИС-32 Щепетова Оксана Евгеньевна

подпись

место практики

наименование юридического лица, ФИО ИП Период прохождения:

с «06» июля 2025 г.

по «12» июля 2025 г.

Руководитель практики от предприятия должность

подпись МП

Руководитель практики от техникума: Материкова А.А.

Оценка:

« » 2025 года

г. Череповец 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ……………………………………………………………………...3

* Цель и задачи практики…………………………………………………….….3
* Краткое описание организации, где проходила практики………………………..3
* Сроки и место прохождения………………………………………………………..4

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ…………………………………4

* Общая информация………………………………………………………………….4
* Роль информационных систем в работе организации……………………………4
* Основные используемые технологии……………………………………………5

1. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ…6
   1. Анализ требований информационных систем…………………………………6
   2. Проектирование информационных систем……………………………………6
   3. Разработка информационных систем…………………………………………7
   4. Тестирование информационных систем…………………………………………8
   5. Внедрение, эксплуатация и сопровождение информационных систем………..8
2. ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ………………………………………………..……9

ЗАКЛЮЧЕНИЕ……………………………………………………………….…..…17

СПИСОК ИСПОЛЬЗОВАННЫХ…………………………………………………18 ИСТОЧНИКОВ…………………………………………………………………18

ПРИЛОЖЕНИЕ……………………………………………………………………19

ВВЕДЕНИЕ

* Цель и задачи практики.

Целью практики является приобретение у обучающегося практического опыта в проектировании и разработке информационных систем, а также освоение методов и инструментов, необходимых для успешного управления процессом разработки, тестирования и эксплуатации программных продуктов. Практика направлена на развитие навыков работы с информационными системами, понимание их структуры и функционирования, а также на формирование компетенций в рамках модуля ПМ.05 специальности 09.02.07 Информационные системы и программирование.

Задачи:

1. Ознакомление с процессами разработки информационных систем:

• Изучение этапов жизненного цикла разработки программного обеспечения.

• Понимание ролей и задач участников проекта.

2. Развитие практических навыков:

• Управление процессом разработки приложений с использованием современных инструментальных средств.

• Программирование в соответствии с техническим заданием и стандартами качества.

3. Анализ и проектирование:

• Осуществление анализа предметной области и выявление требований к информационной системе.

• Выбор моделей и методов обработки информации, соответствующих поставленным задачам.

4. Тестирование и оценка качества:

• Применение методик тестирования разрабатываемых приложений.

• Использование критериев оценки качества и надежности функционирования информационной системы.

5. Документация:

• Разработка проектной и эксплуатационной документации по информационной системе.

• Формирование отчетной документации по результатам выполненных работ.

6. Системный анализ и стандартизация:

• Изучение основных понятий системного анализа и методов контроля качества.

• Ознакомление с национальными и международными стандартами в области сертификации и обеспечения качества продукции.

7. Модификация и управление проектом:

• Модификация отдельных модулей информационной системы на основе анализа и интересов клиента.

• Управление проектом разработки модуля с использованием инструментальных средств.

* + Краткое описание организации, где проходила практика

Малленом Системс — это одна из ведущих компаний в России, занимающаяся разработкой и внедрением инновационных систем компьютерного зрения, а также промышленной видеоаналитики и интеллектуальной обработки данных. В центре её решений лежат современные технологии машинного зрения и искусственного интеллекта, включая алгоритмы машинного обучения и глубокие нейронные сети.раммировании в соответствии с требованиями технического задания, определения состава оборудования и программных средств разработки

информационной системы, разработке документации по эксплуатации информационной системы.

* + Сроки и место прохождения.

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

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ (ОРГАНИЗАЦИИ)

Общая информация (сфера деятельности)

Компания "Малленом Системс" специализируется на разработке и внедрении интеллектуальных систем обработки и анализа изображений, решений в области промышленного зрения, технического зрения и автоматизации. Основные направления деятельности включают:

* + разработку программного и аппаратного обеспечения для систем машинного зрения;
  + автоматизацию процессов контроля качества на производственных предприятиях;
  + создание интеллектуальных систем распознавания (например, номеров автомобилей, объектов на конвейерах и др.);
  + интеграцию решений в промышленные комплексы.

Роль информационных систем (ИС) в работе организации Информационные системы играют ключевую роль в деятельности

компании, поскольку она напрямую связана с разработкой и внедрением программных решений. Основные аспекты использования ИС:

* + Автоматизация процессов разработки — системы управления версиями, среды разработки и трекеры задач помогают командам эффективно взаимодействовать.
  + Разработка программного обеспечения — ИС используются на всех этапах: от сбора требований и проектирования до тестирования и внедрения.
  + Анализ изображений и видео — критически важный элемент продуктов компании, реализуемый средствами ИИ и компьютерного зрения.
  + Хранение и обработка данных — информационные системы обеспечивают надёжную работу с большими объёмами данных, получаемых от промышленных камер и сенсоров.
  + Контроль и мониторинг проектов — с помощью ИС ведётся управление проектами, задачами и версиями ПО.

Таким образом, ИС являются не вспомогательным, а центральным инструментом в деятельности предприятия.

Основные используемые технологии Языки программирования:

* C# — основной язык для разработки десктопных и встроенных решений.
* Python — применяется для задач машинного обучения, прототипирования и анализа данных.

СУБД:

* PostgreSQL, SQLite, MySQL — используются в зависимости от конкретного проекта.
* Также применяются файловые хранилища и специализированные форматы данных (например, для хранения изображений, логов, конфигураций).
* Системы контроля версий:
* Git (используется повсеместно).
* Репозитории размещаются в GitLab и GitHub.
* Фреймворки и библиотеки:
* .NET / .NET Core — основа серверных и клиентских приложений на C#.
* OpenCV — для компьютерного зрения.
* TensorFlow / PyTorch — в проектах, связанных с нейросетями и ИИ.
* Flask / FastAPI — для создания API на Python. Инструменты проектирования:
* Draw.io, Lucidchart, Enterprise Architect — для создания диаграмм UML и ER- моделей.
* PlantUML — для генерации схем на основе текстового описания.
* Методологии разработки:
* Agile / Scrum/ Kanban — гибкие методологии управления проектами.
* Часто используются итеративные подходы и CI/CD-практики. Системы управления задачами:
* Jira, Redmine, Trello, YouTrack — применяются для постановки задач, контроля прогресса и ведения документации.
* Также возможна интеграция с GitLab Issues.

1. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ
   1. Анализ требований к информационным системам

Анализ требований — критический начальный этап жизненного цикла информационной системы, определяющий успех разработки. Он включает сбор функциональных (что должна делать система) и нефункциональных (как она должна работать: надежность, производительность, безопасность) требований, а также выявление и устранение конфликтов между участниками проекта.

Для формализации и визуализации требований используется моделирование, помогающее понять архитектуру системы и выявить недостатки до кодирования. Применяются знаковые модели: диаграммы потоков данных (DFD), UML- диаграммы (вариантов использования, классов, последовательностей) и ER- модели. Могут также разрабатываться интерактивные прототипы UI.

Например, для системы учёта сотрудников ER-модель покажет сущности («Сотрудник», «Должность», «Отдел») и их связи, а DFD — процессы приёма и увольнения.

* 1. Проектирование информационных систем

Этап проектирования включает выбор архитектурного подхода и построение логической структуры системы. Разработчики могут использовать функционально- ориентированную модель, при которой система разбивается на иерархию функций, или объектно-ориентированную, где основное внимание уделяется данным и их взаимосвязям (объектам и классам).

Ключевое значение имеет выбор архитектуры. Примеры:

* + - Клиент-серверная архитектура — клиентские приложения обращаются к серверу за обработкой данных, что снижает требования к мощности клиентского оборудования.
    - Многоуровневая архитектура (например, трёхзвенная) — включает уровни представления, бизнес-логики и данных, что обеспечивает более высокий уровень безопасности, удобство масштабирования и управления кодом.
    - Подбор технологий происходит с учётом требований к системе:
    - Для работы со структурированными данными обычно применяются реляционные СУБД (PostgreSQL, MySQL);
    - При необходимости обрабатывать неструктурированные или гибкие форматы данных (например, JSON-документы) выбираются NoSQL-решения (MongoDB, Couchbase).

Дополнительно учитываются особенности среды исполнения — например, использование микросервисной архитектуры в облачных решениях.

* 1. Разработка информационных систем

В процессе разработки необходимо определить методологию управления проектом. Для проектов с чётко установленными требованиями применяется каскадная модель (Waterfall), где этапы (анализ, проектирование, кодирование, тестирование, внедрение) выполняются последовательно. Однако этот подход слабо адаптируется к изменяющимся условиям.

Современные практики предпочитают гибкие методологии (Agile, Scrum, Kanban), при которых продукт создаётся итерационно с постоянной обратной связью от пользователей. Это позволяет своевременно вносить корректировки и ускоряет выход промежуточных версий.

Инструменты разработки включают:

Системы контроля версий — например, Git, обеспечивающий совместную работу над кодом и отслеживание изменений;

* + - Среды разработки (IDE) — такие как Visual Studio, PyCharm, VS Code;
    - Фреймворки — для веб-разработки популярны Django (Python), ASP.NET (C#), React, Angular, которые ускоряют создание интерфейсов и бизнес-логики.

Крайне важна техническая и пользовательская документация: спецификации A PI, примеры использования, инструкции для администраторов и конечных пользователей. Она облегчает ввод новых разработчиков в проект, поддержку и дальнейшее развитие системы.

* 1. Тестирование информационных систем

Цель тестирования — подтвердить соответствие разработанной системы требованиям и выявить возможные дефекты. Тестирование проводится на нескольких уровнях:

* + - Модульное — проверка работы отдельных функций или классов.
    - Интеграционное — оценка взаимодействия между модулями.
    - Системное и приёмочное — проверка всей системы на соответствие бизнес- целям.

Нагрузочное тестирование особенно важно для систем с высокой посещаемостью. Оно позволяет определить, сколько пользователей система выдержит одновременно, и при каких условиях начинаются сбои.

Безопасность тестируется с использованием специализированных инструментов — например, OWASP ZAP или Burp Suite, позволяющих находить уязвимости, такие как SQL-инъекции, межсайтовое выполнение скриптов (SS) и др.

Для повышения эффективности применяются средства автоматизированного тестирования и непрерывной интеграции/доставки (CI/CD), например, Jenkins, GitLab CI, GitHub Actions. Эти инструменты автоматически запускают тесты при каждом обновлении кода, что помогает оперативно выявлять ошибки.

* 1. Внедрение, эксплуатация и сопровождение информационных систем

Фаза внедрения предполагает установку и конфигурацию системы, обучение пользователей и технического персонала. Один из подходов — параллельный запуск, при котором старая и новая системы работают одновременно, снижая риск ошибок при переходе.

Во время эксплуатации проводится мониторинг работы системы, сбор логов и метрик, таких как загрузка процессора, объём памяти, отклик API. Популярные инструменты мониторинга — Prometheus, Grafana, Zabbi. Сопровождение включает:

* + - устранение найденных ошибок;
    - выпуск обновлений;
    - оптимизацию производительности;
    - обеспечение масштабируемости по мере роста нагрузки;
    - регулярное резервное копирование данных и реализацию плана аварийного восстановления (Disaster Recovery).

Это обеспечивает стабильную работу ИС в долгосрочной перспективе и минимизирует возможные потери данных при сбоях.

1. ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

## Задание 1. Анализ и выбор системы управления базами данных (СУБД)

### 1.1 Анализ популярных СУБД

### 1.2 Обоснование выбора PostgreSQL

| СУБД | Лицензия | ACID | Расширя-ть | Надёжн-ть | JSON | Репликация | Платформа |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PostgreSQL | BSD | Да | Да | Высокая | Да | Встроенная | Кросс |
| MySQL | GPL | Да | Частично | Высокая | Огранич. | Дополн. | Кросс |
| MariaDB | GPL | Да | Да | Высокая | Да | Да | Кросс |
| MS SQL | Проп. | Да | Частично | Высокая | Да | Да | Win, Linux |
| Oracle | Проп. | Да | Да | Очень высокая | Да | Да | Кросс |
| SQLite | Public | Да | Нет | Средняя | Да | Нет | Встр. |

Для выполнения задач в рамках производственной практики была выбрана система управления базами данных **PostgreSQL**. Этот выбор обусловлен следующими причинами:

1. **PostgreSQL** является свободно распространяемой и полностью бесплатной СУБД с открытым исходным кодом.
2. Система поддерживает транзакционность и гарантирует соответствие требованиям **ACID**, что обеспечивает высокий уровень надёжности.
3. PostgreSQL позволяет эффективно обрабатывать сложные запросы, поддерживает **оконные функции**, **CTE**, **полнотекстовый поиск** и другие современные механизмы работы с данными.
4. Поддержка **масштабируемости**, **репликации** и **шардирования** делает её подходящей для систем, работающих в режиме реального времени.
5. PostgreSQL обладает мощными средствами работы с **JSON**, **массивами**, **геоданными** и другими полуструктурированными форматами.
6. Расширяемость позволяет добавлять собственные функции, типы данных и операторы, что особенно полезно в задачах технического контроля.
7. Активное сообщество, регулярные обновления и богатая экосистема делают PostgreSQL зрелым и надёжным решением для промышленныВывод

Изучив популярные СУБД, было принято решение использовать **PostgreSQL**. Она оптимально подходит для задач, связанных с автоматизацией, хранением больших объёмов структурированных и полуструктурированных данных, аналитикой и интеграцией с другими информационными системами.

## Задание 2. Выбор предметной области

Выбранная предметная область:  
 **База данных для хранения результатов автоматического обнаружения дефектов автомобильных кузовов на конвейере.**

### Обоснование выбора

Современные автопроизводства активно внедряют технологии **автоматического контроля качества** на этапе сборки и окраски кузовов. Системы технического зрения, тепловизоры, ультразвуковые и лазерные датчики в режиме реального времени фиксируют следующие параметры:

* наличие трещин, царапин, вмятин, сколов, неровностей покрытия;
* координаты и размеры обнаруженных дефектов;
* этап или участок обнаружения (сварка, покраска, сборка);
* дата, время и смена;
* идентификатор кузова и производственной партии;
* результаты автоматической классификации (тип дефекта, уровень критичности);
* изображения или видеозаписи дефектов.

Актуальность хранения и анализа этих данных обоснована следующими факторами:

* необходимость документального подтверждения качества выпускаемой продукции;
* выявление системных проблем (например, ошибка в роботе-сварщике или неисправность распылителя краски);
* оперативное принятие решений об устранении дефектов;
* интеграция с ERP/MES-системами и системами планирования производства;
* возможность аналитики, построения отчётности, визуализации на панели управления.

Для этих задач требуется надёжная и гибкая база данных, обеспечивающая:

* высокую скорость записи и чтения;
* устойчивость к сбоям и защиту от потерь данных;
* хранение большого массива информации (в том числе изображений/видео по ссылке);
* гибкую фильтрацию и отчётность по параметрам;
* разграничение прав доступа по ролям (оператор, инженер, администратор).

Выбранная СУБД **PostgreSQL** соответствует всем этим требованиям, благодаря:

* поддержке полнотекстового поиска, индексирования и геометрических типов данных (например, для хранения координат дефекта);
* расширенной работе с временными рядами;
* возможности масштабирования и резервного копирования;
* модульной архитектуре и поддержке сторонних расширений (например, PostGIS, pg\_partman, timescaledb).

### Вывод:

База данных для системы обнаружения дефектов кузовов автомобилей на конвейере — это технологически и практически значимая предметная область, отвечающая задачам современной промышленной автоматизации. PostgreSQL позволяет реализовать такую систему с учётом всех требований к надёжности, масштабируемости и аналитике.

1. Задание 3. Составление Технического задания на разработку базы данных

Техническое задание на разработку базы данных (База данных для системы обнаружения дефектов автомобильных кузовов на конвейере)

### **a. Описание целей и задач БД**

**Цель:**  
Создание реляционной базы данных для хранения, анализа и отслеживания результатов автоматического контроля качества кузовов автомобилей на конвейерной линии.

**Задачи:**

* Автоматизация сбора, хранения и анализа данных о дефектах кузовов;
* Обеспечение удобного поиска и фильтрации данных по дате, типу дефекта, кузову и др.;
* Интеграция с системами визуального контроля, промышленными камерами, датчиками и ERP/MES-системами;
* Поддержка формирования отчётов и аналитики для производственного контроля;
* Обеспечение разграниченного доступа к данным по ролям.

### **b. Требования к функциональности**

Система должна обеспечивать:

1. Регистрацию каждого кузова, поступающего на контроль (уникальный идентификатор, дата и время);
2. Хранение данных о результатах визуального и сенсорного контроля;
3. Фиксацию каждого выявленного дефекта с указанием:
   * типа дефекта (вмятина, трещина, царапина, покрас, недокрас и т.д.),
   * координат расположения,
   * степени серьёзности;
4. Возможность хранения фотографий или видео дефекта (в виде ссылок или путей);
5. Формирование отчётов по сменам, участкам, типам дефектов, операторам;
6. Поиск и фильтрация по любому из параметров;
7. Контроль доступа по ролям (оператор, инженер, контролёр, администратор).

### **c. Требования к данным**

**Типы данных:**

* Уникальный идентификатор кузова;
* Марка автомобиля и модель;
* Технические параметры (номер конвейера, смена, оператор, участок);
* Дата и время проверки;
* Тип, описание и координаты дефекта;
* Фото/видео (ссылки);
* Статус контроля (принят/забракован);
* Исполнитель контроля (человек или автоматическая система).

**Объёмы:**

* До 5 000 кузовов в день;
* В среднем по 2–5 дефектов на кузов;
* Хранение данных за минимум 3 года (~20–30 млн записей);
* Фото/видео — по ссылке на хранилище (например, файловый сервер или объектное хранилище).

**Источники данных:**

* Камеры и сенсоры автоматического контроля;
* Ручной ввод оператором (при подтверждении/уточнении);
* Системы производственного учёта (MES, ERP).

### **d. Ограничения**

**Производительность:**

* Минимальная задержка при записи дефектов с линии;
* Мгновенный поиск по основным полям (дата, тип дефекта, ID кузова).

**Безопасность:**

* Ролевая модель доступа (оператор, инженер, администратор);
* Аудит действий пользователей;
* Защита от изменения исторических данных.

**Масштабируемость:**

* Возможность горизонтального масштабирования БД;
* Возможность добавления новых конвейеров, линий, моделей автомобилей.

### e. Сроки и этапы разработки

| Этап | Содержание | Срок |
| --- | --- | --- |
| 1 | Анализ требований, проектирование структуры БД и сущностей | 1 день |
| 2 | Создание схемы БД, реализация таблиц и связей в PostgreSQL | 1 день |
| 3 | Импорт тестовых данных, написание основных SQL-запросов | 1 день |
| 4 | Реализация системы ролей и прав доступа (администратор, оператор) | 1 день |
| 5 | Тестирование БД, проверка целостности данных, подготовка отчета | 1 день |
| **Итого** | **Полный цикл разработки** | **5 рабочих дней** |

ER-диаграмма: Визуализирует взаимосвязи данных (сущностей), их характеристики и взаимодействия в базе данных.

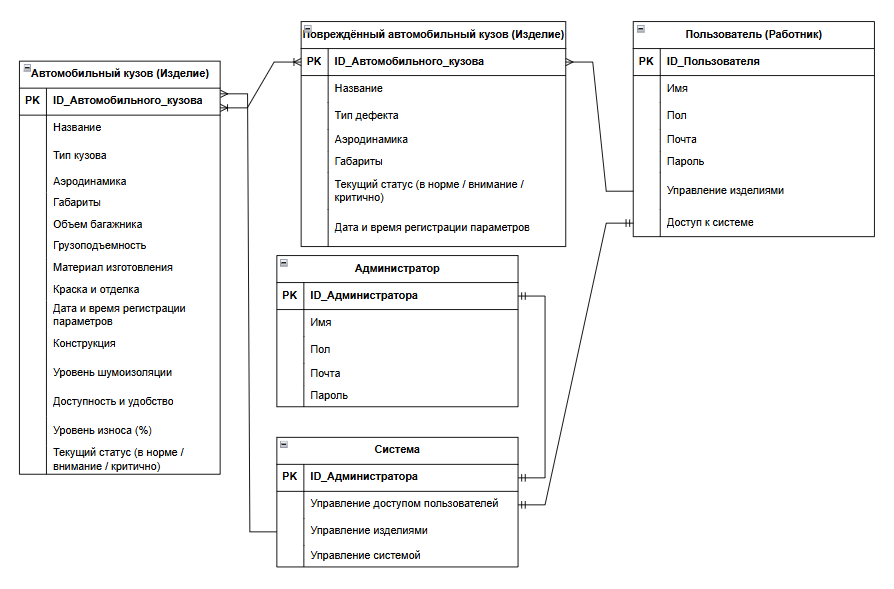


Рисунок 1 − ER диаграмма

* 1. Диаграмма вариантов использования.

Внешний вид диаграммы вариантов использования рисунок 6.

* 1. Диаграмма последовательностей.

Внешний вид диаграммы последовательности рисунок 7.

* 1. Диаграмма компонентов.



Рисунок 2 − Диаграмма компонентов

1. Диаграмма пакетов.

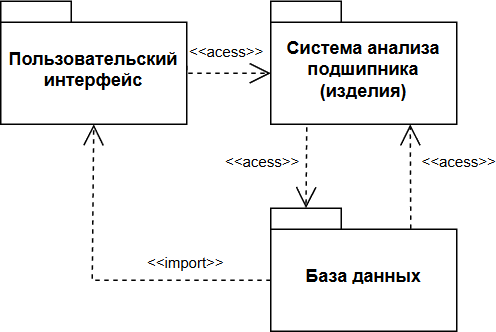


Рисунок 3 − Диаграмма пакетов

1. Диаграмма деятельности.

Внешний вид диаграммы деятельности рисунок 8.

ЗАКЛЮЧЕНИЕ

Прохождение практики в компании ООО «Малленом Системс» стало для меня ценным этапом профессионального развития и дало возможность значительно расширить спектр прикладных знаний в области проектирования и создания информационных систем. В рамках практики я не только углубила теоретические знания, полученные в учебном процессе, но и на практике освоила работу с различными инструментами проектирования, научилась строить ER-диаграммы, разрабатывать техническую документацию и структурировать задачи в канбан-системах.

Особенно важным для меня стало знакомство с корпоративными подходами к управлению проектами. Использование Kanban-досок позволило лучше понять, как внутри компаний выстраиваются процессы планирования и реализации задач, как организуется командная работа и каким образом осуществляется контроль сроков. Благодаря этому опыту я научилась самостоятельно ставить задачи, приоритизировать их, следить за сроками выполнения и оперативно вносить коррективы в план действий.

Практика дала мне возможность погрузиться в разработку реального подсистемного решения — системы хранения данных для мониторинга состояния подшипников. Я научилась проектировать базу данных на основе требований к системе, составлять грамотное техническое задание, разрабатывать структурные схемы и документировать процессы. Такой опыт особенно ценен, поскольку он приближает к условиям работы в ИТ- индустрии, где от специалиста требуется не только знание языков программирования, но и понимание архитектурных принципов, стандартов документации и умение работать в команде.

Полученные знания и навыки стали важным шагом на пути к моей профессиональной самореализации. Я уверена, что опыт, приобретенный в ходе практики, послужит прочной основой для моей будущей карьеры в области информационных технологий и поможет успешно решать реальные производственные задачи.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методы анализа требований [Электронный ресурс]/ - Режим доступа: https://logrocon.ru/news/requirements\_analysis2

2. Графический редактор диаграмм [Электронный ресурс]/ Режим доступа: https://[www.drawio.com/](http://www.drawio.com/)

3. Методы моделирования и модели разработки ИС [Электронный ресурс]/ - Режим доступа: https://inftis.narod.ru/pis/pis-p3-1.htm

4. Основные этапы проектирования ИС [Электронный ресурс]/ - Режим доступа: https://scilead.ru/article/714-osnovnie-etapi-proektirovaniya- informatsionnik

5. Тестирование информационных систем [Электронный ресурс]/ - Режим доступа: https://at-consulting.ru/testirovanie-informacionnyh-sistem

6. Разработка и внедрение информационных систем [Электронный ресурс]/ - Режим доступа: https://infocom-s.ru/sozdanie-informacionnyh-sistem

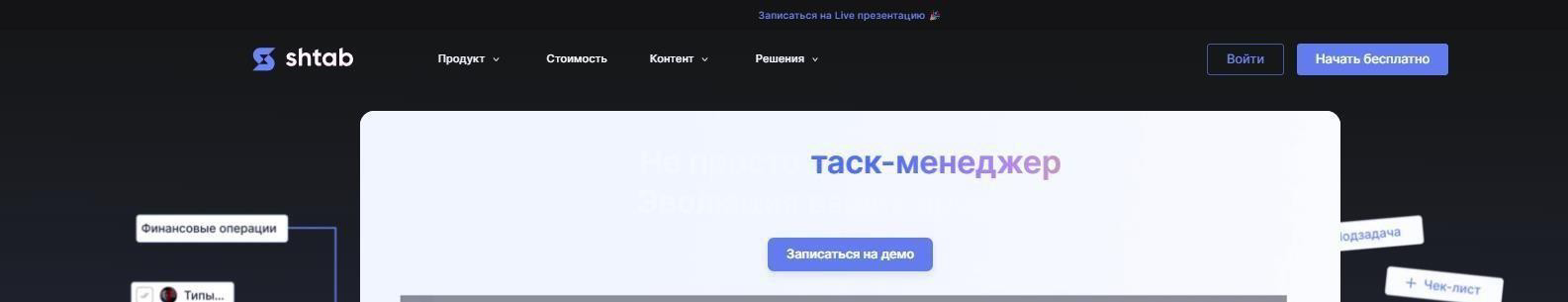
7. Примеры UML диаграмм [Электронный ресурс]/ - режим доступа: https://practicum.yande.ru/blog/uml-diagrammy/?

8. 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/

9. Методы проектирования ИС [Электронный ресурс]/ - Режим доступа: https://studfile.net/preview/9057964/page:87/

10. Система управления проектами [Электронный ресурс]/ - Режим доступа: https://shtab.app/

ПРИЛОЖЕНИЕ



Скриншот сайта выбранной системы управления проектами

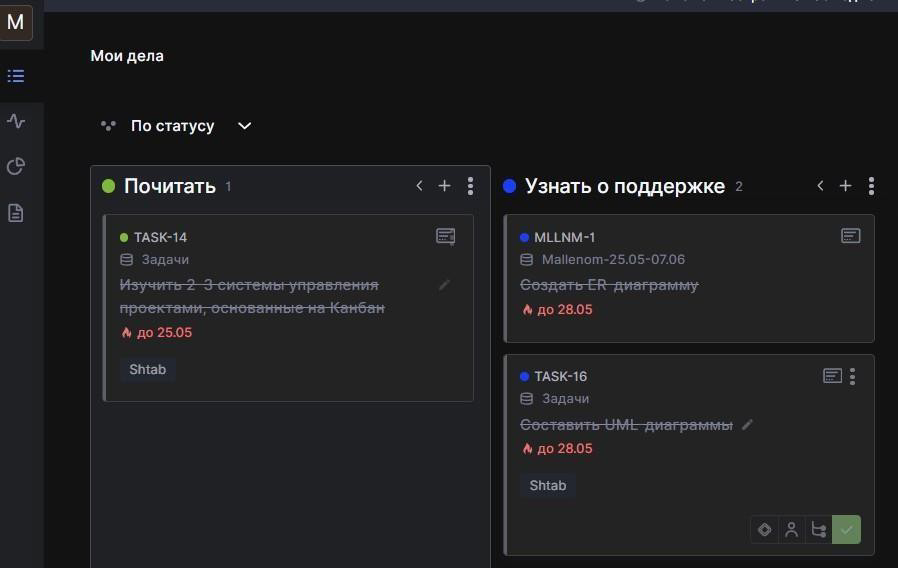


Рисунок 5 − Доска канбан

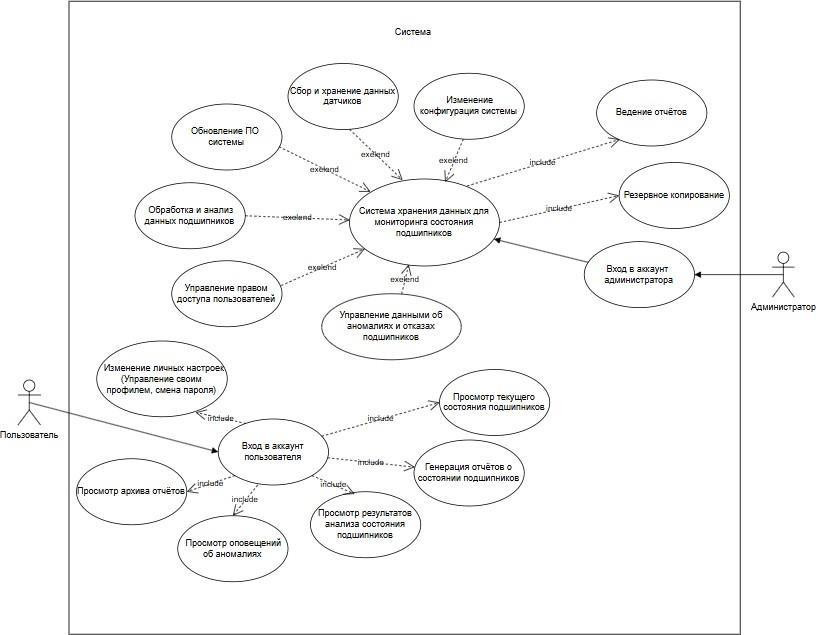


Рисунок 6 − Диаграмма вариантов использования

Рис уно к 4

−

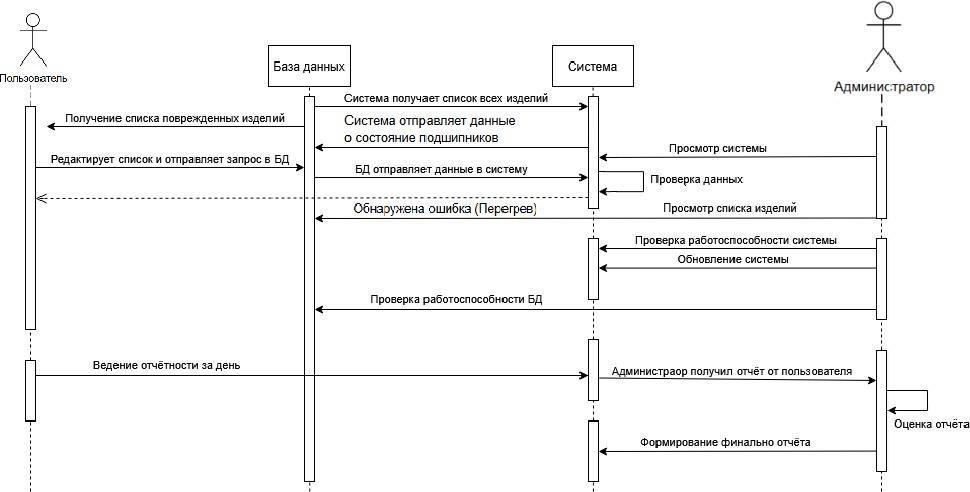


Рисунок 7 − Диаграмма последовательности

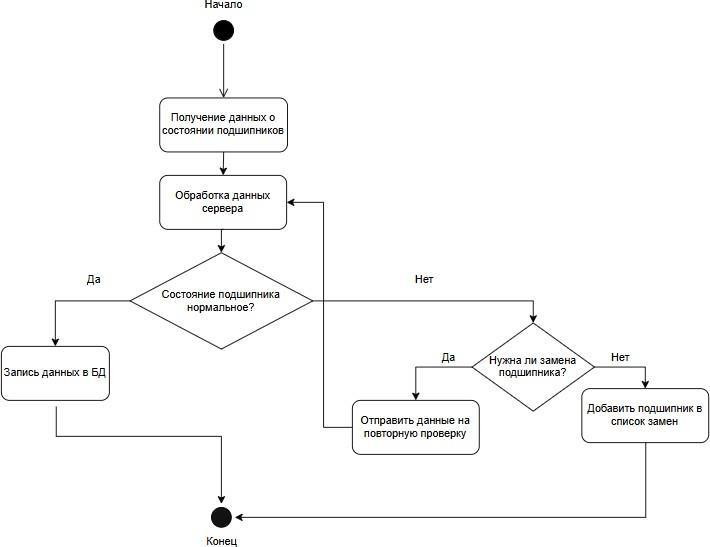


Рисунок 8 − Диаграмма деятельности