СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc199870844)

[1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ 4](#_Toc199870845)

[Общая информация. 4](#_Toc199870846)

[Роль информационных систем в работе организации. 5](#_Toc199870847)

[Основные используемые технологии 6](#_Toc199870848)

[2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ 8](#_Toc199870849)

[2.1 Анализ требований и моделирование информационных систем 8](#_Toc199870850)

[2.2 Проектирование информационных систем 8](#_Toc199870851)

[2.3 Разработка информационных систем 9](#_Toc199870852)

[2.4 Тестирование информационных систем 9](#_Toc199870853)

[2.5 Внедрение, эксплуатация и сопровождение информационных систем 10](#_Toc199870854)

[3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ 11](#_Toc199870855)

[ЗАКЛЮЧЕНИЕ 19](#_Toc199870856)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 20](#_Toc199870857)

[ПРИЛОЖЕНИЯ 21](#_Toc199870858)

# ВВЕДЕНИЕ

Цель практики заключается в получении практического опыта и знаний в сфере проектирования и разработки информационных систем с обработкой информации.

Задачи практики заключались в управлении процессами разработки приложений с использованием инструментальных средств; обеспечении сбора данных для анализа использования и функционирования информационной системы; применении методики тестирования разрабатываемых приложений; определении состава оборудования и программных средств разработки информационной системы с использованием алгоритмов обработки информации.

ООО "Малленом Системс" – ведущий российский разработчик программно-аппаратных решений в области **промышленного машинного зрения и интеллектуальной видеоаналитики**. Компания специализируется на создании высокотехнологичных систем, использующих алгоритмы компьютерного зрения (CV), машинного обучения (ML) и глубокого обучения (Deep Learning) для автоматизации критически важных задач контроля качества, безопасности, логистики и учета на промышленных предприятиях, транспортных узлах и объектах инфраструктуры. Портфель продуктов компании охватывает решения для идентификации объектов (продукция, вагоны, автотранспорт), контроля технологических процессов, ситуационной безопасности и анализа транспортных потоков.

Сроки практики с 25.05.2025 по 07.06.2025, место прохождения практики ООО “Малленом Системс” - Металлургов 21б.

# 1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ

## Общая информация.

Основная среда деятельности: Разработка, производство (интеграция аппаратной части), внедрение и сопровождение программно-аппаратных комплексов (ПАК) на основе компьютерного зрения и искусственного интелекта.

Ключевые технологические направления:

* Машинное зрение (Machine Vision) для промышленной автоматизации.
* Ситуационная видеоаналитика (Video Analytics, Video Content Analysis – VCA).
* Распознавание образов (Pattern Recognition), включая распознавание объектов, текста (OCR), лиц (при необходимости в рамках продуктов).
* Алгоритмы машинного обучения и глубокого обучения (нейронные сети) для анализа изображений и видео.

Ключевые продукты компании:

* ВИСКОНТ: Интеллектуальные системы машинного зрения для контроля качества продукции и прослеживаемости (track & trace).
* EYECONT: Система ситуационной видеоаналитики для контроля людей, действий и опасных событий на видео в реальном времени.
* АВТОМАРШАЛ: Система контроля доступа и учета автотранспорта на основе распознавания гос. номеров (ANPR/LPR).
* АВТОМАРШАЛ.ВЕСОВАЯ: АПК для автоматизации автомобильных весовых (интеграция распознавания номеров, управления шлагбаумами, сбора данных с весов).
* AVEDEX: ПО для подсчета и анализа автомобильного трафика по видео (классификация ТС, замер скорости, подсчет).
* АРДИС: Платформа и решения для идентификации вагонов (распознавание номеров), управления отгрузкой, коммерческого осмотра и контроля передвижения вагонов на железной дороге.

## Роль информационных систем в работе организации.

Я**дро бизнеса:** Информационные системы и алгоритмы ИИ **являются сутью продуктов** компании. Без них решения "Малленом Системс" не существовали бы.

Продуктовые ИС: Каждый продукт (ВИСКОНТ, EYECONT, АВТОМАРШАЛ и т.д.) представляет собой сложную **специализированную информационную систему**, включающую:

* Модули захвата и предоработки видео/изображений.
* Ядро обработки (алгоритмы CV, ML/DL модели).
* Базы данных для хранения результатов анализа, эталонов, конфигураций.
* Пользовательские интерфейсы (веб, десктоп) для настройки, мониторинга и отчетности.
* Системы интеграции с внешними ИС (ERP, WMS, SCADA, системы безопасности).

Внутренние ИС: Для обеспечения собственной эффективности компания использует:

* Системы управления проектами: Управление R&D, внедренческими и интеграционными проектами (Jira, Redmine).
* CRM: Управление продажами сложных B2B-решений, поддержка клиентов.
* Системы контроля версий (VCS): Критически важны для разработки ПО и алгоритмов (GitLab, GitHub).
* C**истемы управления данными и обучения моделей:** Платформы для хранения и разметки изображений/видео, управления pipeline обучения нейросетей (возможно, внутренние разработки или специализированные инструменты - CVAT, LabelBox, MLflow, Kubeflow).
* Системы CI/CD: Автоматизация сборки, тестирования ПО и развертывания обновлений (GitLab CI/CD, Jenkins).
* C**истемы документооборота и Wiki:** Хранение технической документации, знаний (Confluence, внутренние Wiki).
* Системы коммуникации.

Роль: ИС (как продуктовые, так и внутренние) являются **абсолютно критической инфраструктурой** для создания, поставки и поддержки инновационных решений компании, обеспечивая ее технологическое лидерство.

## Основные используемые технологии

Языки программирования:

* Основные: Python (доминирует в CV/ML/DL), C++ (высокопроизводительные компоненты, интеграция с оборудованием).
* Вспомогательные: JavaScript/TypeScript (веб-интерфейсы), SQL.

Фреймворки и платформы общего назначения:

* Бэкенд: FastAPI, Django, Flask (Python); Node.js;
* Фронтенд: React.js, Vue.js, Angular.
* Микросервисы: Docker, Kubernetes (k8s) - для масштабируемости компонентов аналитики.

СУБД (Системы управления базами данных):

* Реляционные: PostgreSQL (основная для бизнес-данных, конфигураций), MySQL.

Системы контроля версий: Git (GitLab, GitHub).

Инструменты проектирования:

* Диаграммы: UML (PlantUML, draw.io), блок-схемы алгоритмов.
* Проектирование ИИ: Диаграммы архитектур нейросетей, карты данных.
* Моделирование данных: ER-диаграммы, интрументы СУБД.

Методологии разработки:

* Гибкие методологии (Agile): Scrum (для ПО), Kanban (для задач поддержки, оперативной разработки).

Другие ключевые компоненты:

* IDE: PyCharm, Visual Studio Code, Visual Studio, Jupyter Notebook/Lab.

2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ

## 2.1 Анализ требований и моделирование информационных систем

Цель: Понять что должна делать система, для кого и в каких условиях.

Задачи:

* Сбор, анализ, документирование и согласование требований от заинтересованных сторон (бизнес-пользователи, руководство, ИТ).
* Идентификация бизнес-процессов, проблем и возможностей.
* Создание моделей (Use Case, диаграммы деятельности, ER-диаграммы, BPMN и т.д.) для визуализации требований и будущей системы.
* Определение функциональных (что система делает) и нефункциональных (производительность, безопасность, удобство) требований.
* Формирование Технического Задания (ТЗ).

## 2.2 Проектирование информационных систем

Цель: Определить как система будет удовлетворять требованиям.

Задачи:

* Архитектурное проектирование: Выбор технологий, платформ, стиля архитектуры (монолит, микросервисы), разбиение системы на компоненты.
* Проектирование данных: Разработка структуры БД (логическая и физическая модель данных), выбор СУБД.
* Проектирование интерфейсов: Пользовательский интерфейс (UI/UX), программные интерфейсы (API).
* Проектирование безопасности: Меры по защите данных и системы.
* Детальное (низкоуровневое) проектирование: Спецификация классов, модулей, алгоритмов, схемы взаимодействия.
* Результат: Техническое Проектирование (ТП), архитектурные диаграммы, спецификации.

## 2.3 Разработка информационных систем

Цель: Написание кода компонентов системы согласно проекту.

Задачи:

* Написание исходного кода на выбранных языках программирования.
* Создание базы данных.
* Разработка пользовательского интерфейса.
* Интеграция отдельных модулей и компонентов.
* Настройка среды разработки и сборка системы.
* Применение методологий разработки.
* Использование систем контроля версий (Git).

2.4 Тестирование информационных систем

Цель: Проверить, соответствует ли разработанная система требованиям, проекту, и выявить дефекты (баги) перед выпуском.

Задачи:

* Планирование тестирования: Определение стратегии, объемов, требований к тестовым данным и средам.
* Разработка тестов: Создание тест-кейсов, тест-скриптов, сценариев автоматизации.
* Выполнение тестов: Ручное и/или автоматизированное прогон тестов.
* Фиксация и отслеживание дефектов.
* Анализ результатов и отчетность.

## 2.5 Внедрение, эксплуатация и сопровождение информационных систем

Цель: Перевести систему в рабочее состояние, обеспечить ее стабильную работу и развитие.

Задачи:

* Внедрение (Deployment):
  + Планирование развертывания (миграция данных, обучение пользователей, переход с старой системы).
  + Установка системы на промышленную среду (продакшн).
  + Окончательное приемочное тестирование.
  + Запуск системы в эксплуатацию.
* Эксплуатация (Operation):
  + Администрирование системы (мониторинг производительности, резервное копирование, управление пользователями).
  + Обеспечение доступности и отказоустойчивости.
  + Техническая поддержка пользователей.
* Сопровождение (Maintenance):
  + Корректирующее: Исправление обнаруженных в эксплуатации ошибок.
  + Адаптирующее: Приспособление системы к изменениям в окружении (ОС, СУБД, законы).
  + Профилактическое: Улучшение сопровождаемости кода, рефакторинг.
* Планирование вывода системы из эксплуатации.

# 3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

В ходе производственной практики поставлена задача изучить и анализировать систем управления проектами, основанные на методологии канбан, после этого определиться с направлением разработки подсистемы хранения данных. Я выбрал разработку подсистемы хранения данных для системы контроля целостности труб, в конце составил план работ и оформил документацию.

Канбан — это гибкая методология управления проектами и процессами, фокусирующаяся на визуализации рабочего потока, ограничении незавершённой работы (WIP) и оптимизации потока задач.

Основные принципы Канбан:

* Визуализация Работы (Kanban Board): Все задачи представлены в виде карточек на доске с колонками, отражающими этапы рабочего процесса (например, "To Do", "In Progress", "Testing", "Done"). Это даёт мгновенную картину состояния проекта.
* Ограничение Незавершённой Работы (Work In Progress - WIP Limits): Устанавливается максимальное количество задач, которые могут одновременно находиться на каждом этапе (в каждой колонке). Это предотвращает перегрузку команды, выявляет узкие места и заставляет завершать начатое.
* Управление Потоком (Flow Management): Цель — сделать поток задач максимально плавным и предсказуемым. Анализируется, где задачи застревают (бутылочные горлышки), как долго задачи находятся в системе (время цикла) и сколько задач завершается за период времени (пропускная способность).
* Эволюционное Улучшение (Kaizen): Канбан не требует резких изменений. Процесс улучшается постепенно, на основе данных (метрик) и наблюдений за потоком.

Преимущества Канбан-Систем:

* Гибкость и Адаптивность: Легко реагировать на изменения приоритетов и входящих задач.
* Повышение Прозрачности: Все видят статус работ, проблемы и загрузку.
* Сокращение Времени Цикла: Фокус на завершении задач и устранении узких мест.
* Предсказуемость: На основе исторических данных (Throughput, Cycle Time) можно точнее прогнозировать сроки.
* Снижение Перегрузки Команды: WIP-лимиты защищают команду от многозадачности.
* Низкий Порог Входа: Легко внедрить поверх существующего процесса.

Инструменты для Канбан:

* Физические Доски: Стикеры и белая доска. Отлично подходят для локальных команд.
* Цифровые Инструменты: Jira (с Kanban-проектом или плагинами): Мощный, гибкий, популярный в разработке ПО.
* Trello: Очень простой и интуитивный, хорош для старта и небольших команд.
* Azure DevOps Boards: Интеграция с экосистемой Microsoft.
* Kanbanize, Leankit: Специализированные инструменты с продвинутыми метриками (CFD) и функциями.
* Asana, Monday.com: Универсальные инструменты с режимами Канбан.
* Выбор инструмента зависит от сложности процессов, размера команды, распределённости, потребности в интеграциях и глубине анализа метрик.

Моя подсистема состоит из сервиса, который обрабатывает данные труб, может удалять и добавлять их.

План работ в днях:

* 1-3: разработка подсистемы и добавление библиотек к ней
* 4-5: проверка подсистемы на наличие ошибок и неисправностей
* 6-7: составление тз, руководства пользователя и администратора, диаграмм
* 8 день: Составление отчета

Техническое задание (ТЗ)

Система: Подсистема хранения данных для системы контроля целостности труб (Pipe Integrity Storage System).

1.1 Введение

* Назначение: Централизованное управление данными трубопроводов с функциями CRUD, фильтрации, архивации и аудита изменений.
* Функции:
  + Централизованное управление данными о трубах (добавление, удаление, редактирование).
  + Обеспечение целостности и доступности данных для системы мониторинга.
* Цель: Aвтоматизация процессов учета труб и контроля их технического состояния для предотвращения аварий. Снижение рисков ЧП за счет мониторинга целостности труб.

1.2 Требования к функционалу:

* Добавление объекта: Прием данных через API.
* Удаление объекта: Удаление по ID номеру.
* Поиск/Фильтрация: Поиск по параметрам: названию, длине, диаметру, статусу.
* Экспорт данных: Выгрузка в формате JSON.

1.3 Структура объекта:

[  
  {  
    "name": "PipeA",  
    "length": 100.0,  
    "diameter": 25.0,  
    "status": "ok"  
  },  
  {  
    "name": "PipeB",  
    "length": 150.5,  
    "diameter": 30.0,  
    "status": "damaged"  
  },  
  {  
    "name": "PipeC",  
    "length": 90.0,  
    "diameter": 20.0,  
    "status": "needs\_repair"  
  }  
]

1.4 Нефункциональные требования

Производительность:

* Обработка 1000+ запросов/мин.
* Отклик API <400 мс.

Безопасность:

* Шифрование данных при передаче.

Хранение:

* База данных: SQLite.

1.5 Технологии

* Бэкенд: Python (FastAPI), БД: SQLite + SQLAlchemy.

1.6 Этапы разработки

* Проектирование API.
* Реализация CRUD-операций.
* Интеграция с хранилищем файлов.
* Тестирование (unit, нагрузочное).
* Документация

Руководство пользователя и администратора:

Для пользователя (API)

Базовый URL: http://127.0.0.1:8000/docs

Добавление объекта:

Curl -X POST "http://127.0.0.1:8000/pipes?name=MainPipe&length=120.5&diameter=30.2&status=ok"

Удаление объекта:

curl -X DELETE http://127.0.0.1:8000/pipes/1

Для администратора

Установка (Uvicorn):

uvicorn main:app –reload

ER – диаграмма (Рисунок 1 в приложении стр 21)

Сущности:

* PIPE: Характеристики труб
* ENGINEER: Персонал для обслуживания
* MONITORING\_SYSTEM: Автоматические системы контроля
* MAINTENANCE\_RECORD: Журнал обслуживания

Ключевые атрибуты:

* status в PIPE: Текущее состояние трубы
* action\_type в MAINTENANCE\_RECORD: Тип проведенных работ
* last\_active в MONITORING\_SYSTEM: Контроль работоспособности системы

UML диаграммы:

Диаграмма вариантов использования (Рисунок 2 в приложении стр 21)

* Исполнители: Администратор, Инженер, Система мониторинга
* Ключевые сценарии: Управление данными труб, отчетность.

Диаграмма последовательностей (Рисунок 3 в приложении стр 22)

Сценарий: обновление статуса трубы

1. Инженер > Система: Запрос на обновление статуса (ID трубы, новый статус)
2. Система > База данных: Поиск трубы по ID
3. База данных > Система: Данные трубы
4. Система > База данных: Обновить статус
5. База данных > Система: Подтверждение успешного обновления
6. Система > Инженер: Уведомление "Статус обновлён"

Диаграмма компонентов (Рисунок 4 в приложении стр 22)

Компоненты:

* API-cервис: Бизнес-логика (обновление статуса, генерация отчётов)
* Хранилище данных: Доступ к БД
* База данных: Таблица Pipe.

Диаграмма пакетов (Рисунок 5 в приложении стр 22)

Иерархия:

* UI: Интерфейсы пользователя.
* Services: Бизнес-логика.
* Storage: Работа с данными.
* Models: Сущности предметной области.

Диаграмма деятельности (Рисунок 6 в приложении стр 23)

Процесс: Обновление статуса трубы

1. Начало процесса:
   * Инженер выбирает трубу (вводит ID)
   * Запрашивает изменение статуса
2. Параллельные проверки:
   * Валидация нового статуса (допустимые значения)
3. Решение:
   * Если данные некорректны → возврат ошибки
   * Если корректны → создание команды обновления
4. Выполнение обновления:
   * Запись в журнал аудита
   * UPDATE запрос в БД
   * Проверка результата выполнения
5. Завершение:
   * Успех: обновление кэша и уведомление
   * Ошибка: возврат сообщения об ошибке

Внешний вид моей подсистемы и панели добавления трубы представлен в приложении на рисунках 7-8(стр 23-24).

Программный код показан в приложении на рисунках 9-14 (стр 24-26).

# ЗАКЛЮЧЕНИЕ

В рамках производственной практики мной было выполнено проектирование и разработка информационной системы. В задачи входило изучение и анализ систем управления проектами, основанных на методологии канбан, а также реализация проекта со своей подсистемой хранения данных.  
Работа над проектом дала мне возможность глубже разобраться в процессах проектирования, разработки и создании подсистем хранения данных.

Практика в ООО "Малленом Системс" способствовала совершенствованию моих навыков проектирования и анализа, а также обеспечила ценный опыт работы с системами управления проектами и взаимодействии с досками канбан.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. <https://gstar.ru/products/geostar-pkrs-101>
2. <https://en.wikipedia.org/wiki/Kanban_(development)#:~:text=Kanban%20(Japanese%3A%20%E7%9C%8B%E6%9D%BF%2C%20meaning,handling%20of%20system%2Dlevel%20bottlenecks>.
3. <https://www.unisender.com/ru/blog/luchshie-servisy-kanban-dosok/>
4. <https://ru.wikipedia.org/wiki/GitHub>
5. <https://ru.wikipedia.org/wiki/SQLite>
6. <https://ru.wikipedia.org/wiki/SQLAlchemy>
7. <https://habr.com/ru/articles/557014/>
8. <https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_(UML)>
9. <https://ru.wikipedia.org/wiki/ER-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C>
10. <https://ru.wikipedia.org/wiki/Python>
11. <https://ru.wikipedia.org/wiki/FastAPI>
12. <https://serverspace.ru/support/glossary/serilog/?utm_source=google.com&utm_medium=organic&utm_campaign=google.com&utm_referrer=google.com>

# ПРИЛОЖЕНИЯ

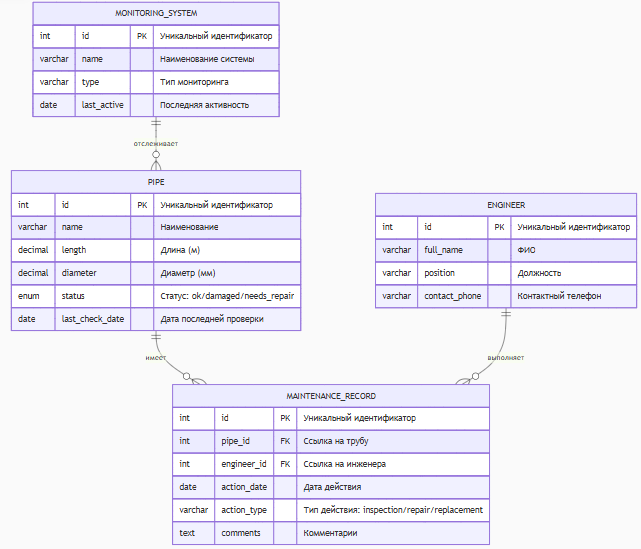


Рисунок 1- ER диаграмма

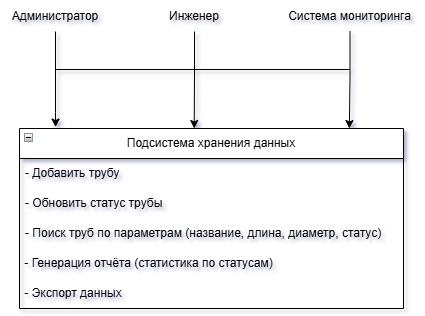


Рисунок 2 - Диаграмма вариантов использования

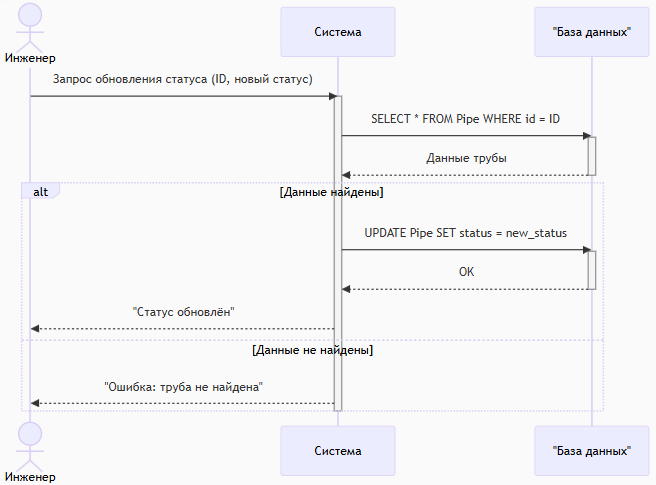


Рисунок 3 - Диаграмма последовательности

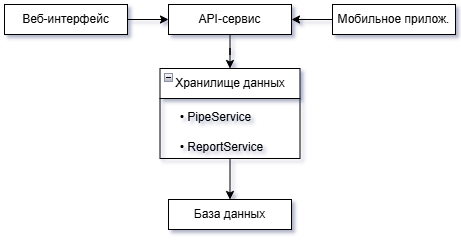


Рисунок 4 - Диаграмма компонентов

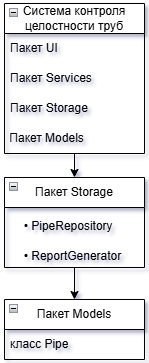


Рисунок 5 - Диаграмма пакетов

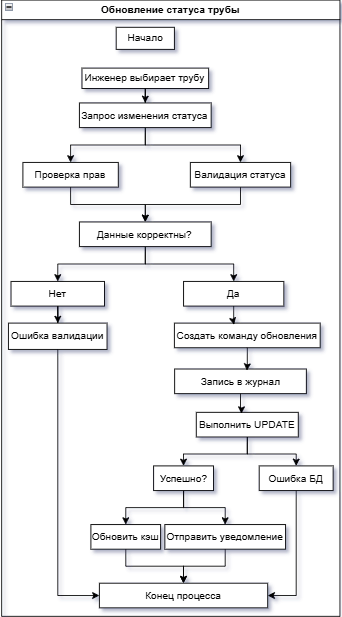


Рисунок 6 - Диаграмма деятельности

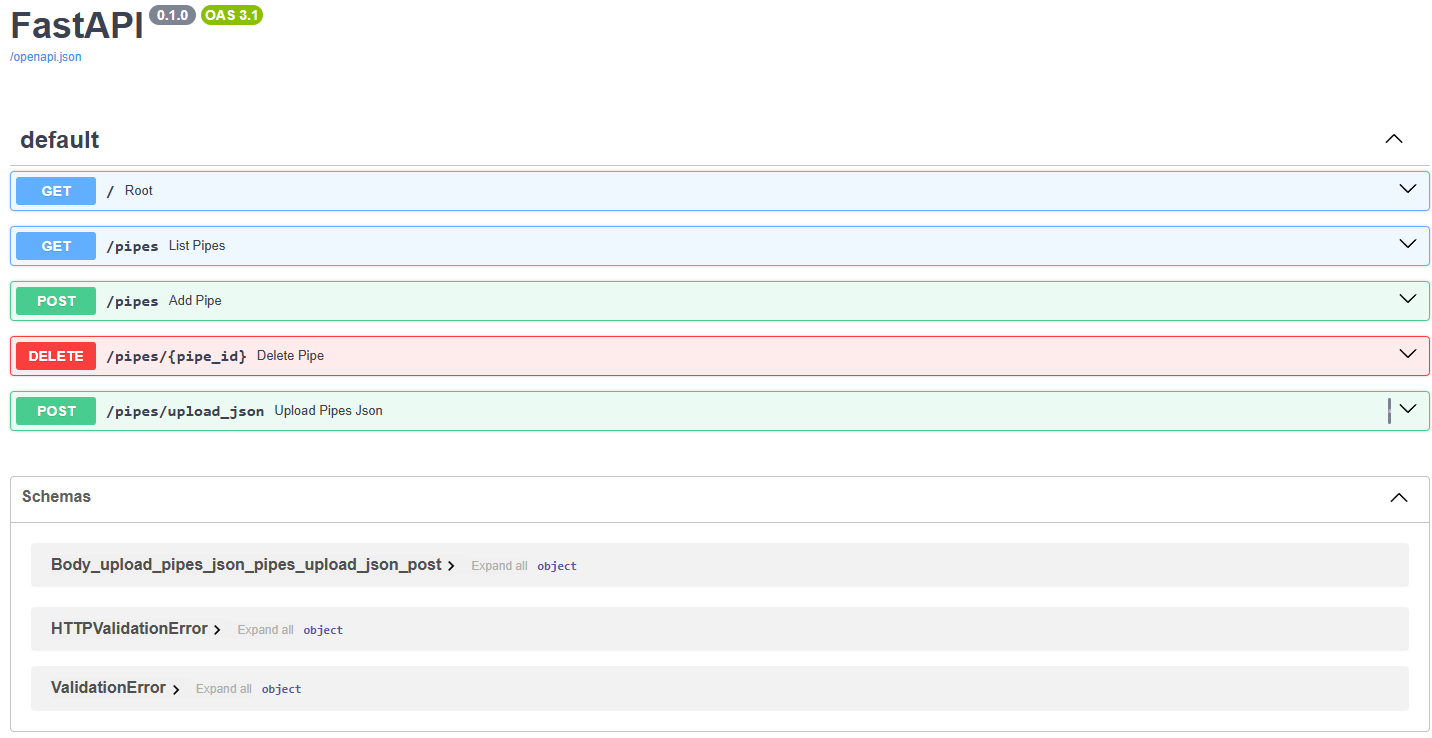


Рисунок 7 - Внешний вид подсистемы



Рисунок 8 - панель с добавлением трубы

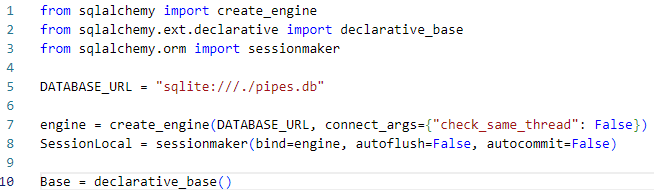


Рисунок 9 - Код файла database.py

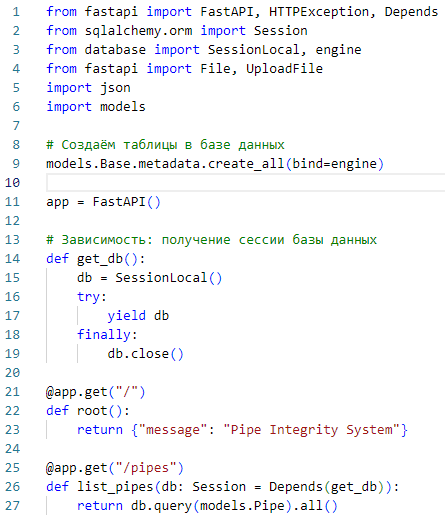


Рисунок 10 - Код файла main.py часть 1

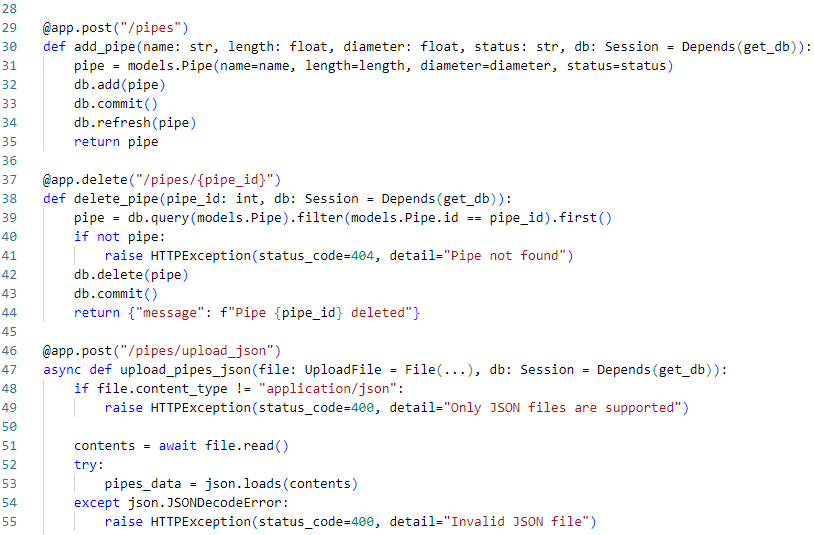


Рисунок 11 - Код файла main.py часть 2

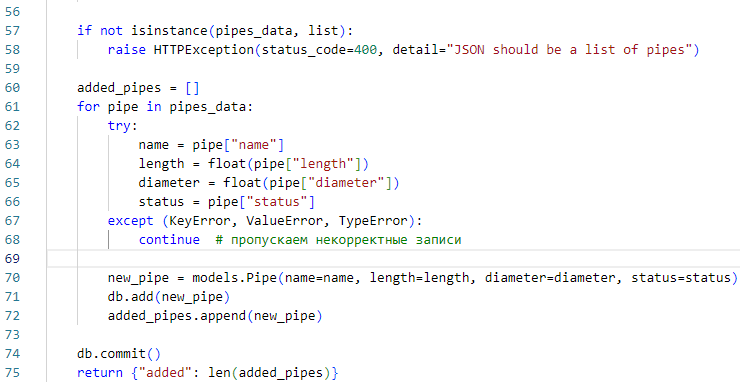


Рисунок 12 - Код файла main.py часть 3

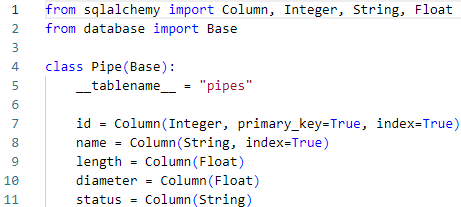


Рисунок 13 - Код файла models.py

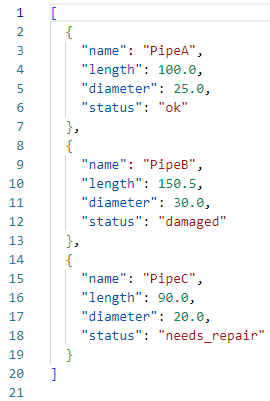


Рисунок 14 - Код файла сохраненных труб pipes.json