

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

Аксёнова Ксения Максимовна

подпись _____

место практики

наименование юридического лица, ФИО ИП

Период прохождения:

с «25» мая 2025 г.

по «07» июня 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 |
| 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ИС ООО “МАЛЛЕНОМ СИСТЕМС” | 4 |
| 1.1 Основные модели построения информационных систем, их структура, особенности и области применения | 5 |
| 1.2 Платформы для создания, исполнения и управления ИС..... | 7 |

ВВЕДЕНИЕ

Место прохождения практики ООО “Малленом Системс”. Сроки прохождения производственной практики: 25.05.2025 по 07.06.2025.

Цель

Производственная практика направлена на формирование у обучающегося общих и профессиональных компетенций, приобретение практического опыта и реализуется в рамках модуля ПМ.05 специальности 09.02.07 Информационные системы и программирование.

Задачи

1. Формирование общих компетенций.
2. Формирование профессиональных компетенций: проектирование и разработка информационных систем, выполнять анализ предметной области, основные модели построения информационных систем, платформы для создания, исполнения и управления информационной системой.
3. Приобретение практического опыта в: управлении процессом разработки приложений с использованием инструментальных средств, обеспечении сбора данных для анализа использования и функционирования информационной системы, программировании в соответствии с требованиями технического задания, определения состава оборудования и программных средств разработки информационной системы, разработке документации по эксплуатации информационной системы.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ИС ООО «МАЛЛЕНОМ СИСТЕМС»

Предметная область информационной системы предприятия ООО «Малленом Системс», охватывает различные аспекты деятельности компании, включая бизнес-процессы, организационную структуру, взаимодействие с клиентами и поставщиками, управление ресурсами и финансами, автоматизацию операций и поддержку принятия решений руководством.

Ключевые элементы предметной области ИС ООО «Малленом Системс»

1) Объекты предметной области:

1. Сотрудники и подрядчики
2. Клиенты и поставщики
3. Товары и услуги
4. Контракты и договоры
5. Технические устройства и оборудование
6. Процессы закупки, производства, реализации и послепродажного обслуживания
7. Финансовые показатели и бюджетирование
8. Документы, регламентирующие внутренние правила и внешние обязательства
9. Аналитическая информация и отчёты

2) Отдел продаж и маркетинга:

1. Производственный отдел
2. Бухгалтерия и финансовый отдел
3. IT-отдел
4. Административно-хозяйственная служба
5. Юридический отдел

3) Требования к информационной инфраструктуре:

Структурированная локальная сеть и доступ к облачным сервисам;

Высокопроизводительные серверы и рабочие станции;

Современные базы данных и хранилища больших объёмов информации;

Средства резервирования и восстановления данных;
Антивирусные программы и системы мониторинга сетевых атак;
Централизованное администрирование и поддержка работоспособности систем.

Проведённый анализ показал, что предметная область информационных систем компании отличается высокой степенью сложности и разнообразием решаемых задач. Эффективность использования ИС определяется уровнем соответствия их функционала целям и задачам компании, качеством подготовки кадров, надёжностью аппаратуры и соблюдением норм законодательства.

1.1 Основные модели построения информационных систем, их структура, особенности и области применения

Основные модели построения информационных систем (ИС), применяемые предприятиями, характеризуются разными принципами организации и областями применения:

1. Классическая централизованная модель («клиент-сервер»)

Структура: Все вычислительные мощности сосредоточены на центральном сервере, клиенты подключаются дистанционно.

Особенности: Легкость управления, высокая степень контроля над системой, но возможна низкая скорость реакции на запросы при большом числе пользователей.

Применение: Большие корпоративные системы с высокими требованиями к безопасности и целостности данных (банковская сфера, крупные промышленные комплексы).

2. Распределённая модель («сервер-приложение-клиент»)

Структура: Часть вычислений распределяется между серверами и клиентскими устройствами.

Особенности: Выше гибкость и устойчивость к сбоям отдельных узлов, нагрузка равномерно распределяется, увеличивается общая пропускная способность.

Применение: Интернет-магазины, онлайн-игры, системы коллективной работы (Google Docs, Trello).

3. Многоуровневая модель

Структура: Система делится на несколько независимых слоев (обычно три: веб-сервер, прикладной слой, база данных). Каждый уровень решает свою задачу отдельно.

Особенности: Более устойчива к изменениям, легче поддерживать и модернизировать отдельные уровни, повышает безопасность.

Применение: Банковские сервисы, медицинские информационные системы, государственные порталы.

4. Сервис-ориентированная архитектура (SOA)

Структура: Основана на предоставлении сервисов (API), каждый из которых реализует отдельную функциональность и доступен другим компонентам системы.

Особенности: Гибкость, независимость компонентов, упрощается интеграция разнородных систем, легко расширяется новыми функциями.

Применение: Крупные коммерческие платформы, финансовые учреждения, телеком-компании.

5. Микросервисная архитектура

Структура: Вместо монолитной системы используется набор небольших самостоятельных микросервисов, выполняющих ограниченный круг функций.

Особенности: Независимое развёртывание каждого сервиса, простота масштабирования и адаптации к изменениям.

Применение: Электронная коммерция, облачные платформы, социальные сети (Amazon, Netflix).

1.2 Платформы для создания, исполнения и управления ИС

1. Платформы разработки (IDE и среды программирования)
Инструментальные среды для написания, компиляции и отладки программного кода:

Примеры: Visual Studio (Microsoft), IntelliJ IDEA (JetBrains), Eclipse (Eclipse Foundation), PyCharm (Python-разработка).

Назначение: Быстрая разработка приложений и сокращение времени вывода продукта на рынок.

2. Платформы управления жизненным циклом приложений (ALM)
Средства для координации разработки, тестирования, деплоя и поддержки программных продуктов:

Примеры: Jira (Atlassian), GitLab CI/CD, TeamCity (JetBrains), Jenkins.

Назначение: Координация команд разработчиков, тестировщиков и DevOps-инженеров, автоматизация сборки и развертывания приложений.

3. Системы управления конфигурациями и изменениями (CMDB и CMMS)
Программы для фиксации текущего состояния инфраструктуры и её изменения:

Примеры: BMC Remedy, IBM Maximo, HP Universal Configuration Management Database (UCMDB).

Назначение: Контроль версий и конфигурации технических активов, устранение конфликтов при изменениях и предотвращение неисправностей.

4. Средства управления знаниями и контентом (ECM) Используются для структурированного хранения, индексации и оперативного поиска корпоративной информации:

Примеры: Alfresco, SharePoint (Microsoft), OpenText Content Suite.

Назначение: Обеспечение доступности актуальной информации для сотрудников, ускорение совместной работы и принятие обоснованных решений.

5. BI-платформы (Business Intelligence) Решения для сбора, анализа и представления данных в удобоваримом виде:

Примеры: Power BI (Microsoft), Tableau, QlikView.

Назначение: Получение инсайтов из большого объема данных, формирование отчетности и поддержка стратегического менеджмента.

ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

Задание 1

1) Изучение и анализ системы управления проектами, основанные на методологии канбан.

Первоначально Kanban — это система организации работы на производстве, которая появилась в конце 50-х годов в Японии. Мастера, ответственные за разные этапы производства, крепили на общую доску карточки с работами, которые нужно было выполнить.

Сейчас, говоря о Kanban, чаще подразумевают гибкую методологию для управления задачами в IT-сфере, например, в командах разработки, службы поддержки, производства контента.

Существует несколько популярных инструментов, основанных на методологии Канбан:

Trello: Один из самых простых и понятных инструментов, идеально подходящий для небольших команд и индивидуальных проектов. Позволяет визуально организовывать задачи и отслеживать прогресс.

Jira: Более мощная платформа, используемая крупными организациями. Имеет гибкую настройку рабочих процессов и интеграции с множеством сторонних приложений.

GitLab: Бесплатный облачный сервис с открытым исходным кодом, включающий возможности CI/CD, Wiki и трекеры багов. Подходит для полного жизненного цикла разработки ПО.

Azure DevOps: Платформа от Microsoft, обеспечивающая полный цикл разработки программного обеспечения, включая проекты, контроль версий и автоматизацию сборки.

2) Выбор подходящей системы управления проектами.

В данной работе была выбрана система GitLab, потому что есть возможность объединения с аккаунтом на GitHub для удобства работы над проектом.

3) После выбора систем управления проектами, необходимо реализовать репозиторий с проектом будущей разрабатываемой системы. На рисунке 1 представлен репозиторий с проектом.

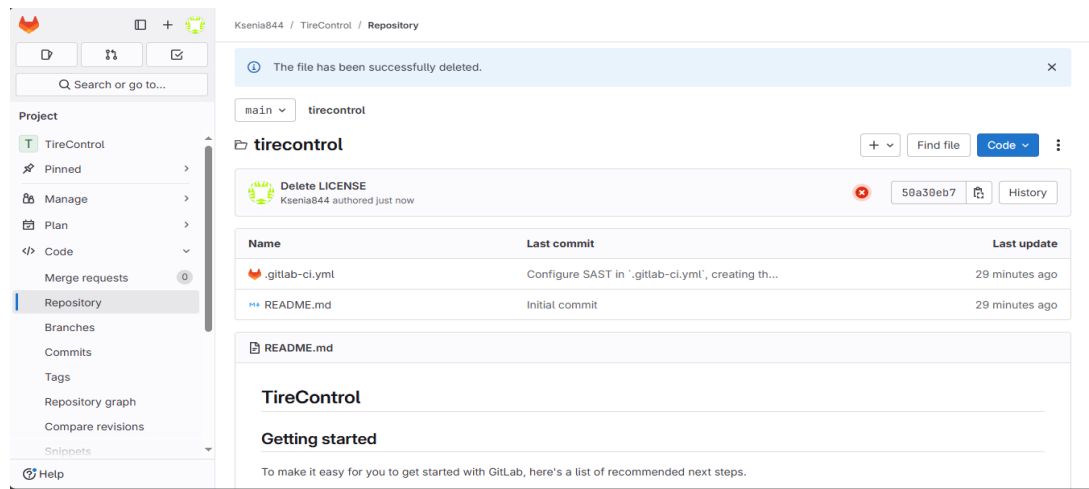


Рисунок 1 - Проект tirecontrol в репозитории GitLab

Задание 2

1) Определение с направлением разработки.

В качестве проекта для разработки выбрана тема: “Подсистема хранения данных для системы контроля шин”. Состоит из сервиса, который обрабатывает данные шин, может удалять, добавлять их.

2) Для выбранной предметной области необходимо составить план работ, задачи и ввести их на доску канбан (из задания 1). Требуется детально рассчитать сроки и количество задач, до окончания практики. Доска канбан изображена на Рисунке 2.

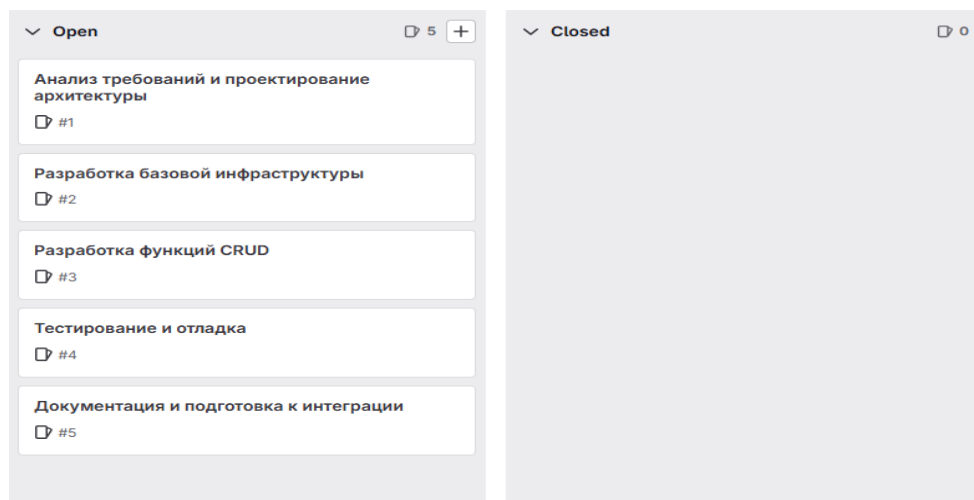


Рисунок 2 - Доска канбан

3) Составление документации для выбранной подсистемы:

1. Техническое задание.

Техническое задание

1. Название подсистемы

Подсистема хранения данных для системы контроля шин («ШинХран»)

2. Описание задачи

Разработать специализированную подсистему для хранения и эффективного доступа к информации о состоянии автомобильных шин, обеспечивающую надежную обработку данных, возможность внесения изменений и удалений.

3. Основные функциональные требования

Хранение характеристик шин: маркировка, производитель, размер, глубина протектора, дата установки, километраж пробега.

Возможности поиска, сортировки и фильтрации шин по указанным критериям.

Возможность создания, обновления и удаления записей о шинах.

4. Инфраструктурные требования

Платформа сервера: Microsoft Windows Server 2022+

Язык программирования: C#

Платформа разработки: .NET Core 9.0.5

Веб-платформа: ASP.NET Core

СУБД: MS SQL Server

ORM-инструменты: Entity Framework Core

Протокол коммуникации: SignalR

Логирование: Serilog

5. Специфические требования

Средства экспорта данных в CSV/XLSX для удобного последующего анализа.

Данные передаются по протоколу HTTPS для повышения безопасности.

6. Ограничения

Скорость интернет-соединения для нормальной работы не менее 5 Мбит/с.

Максимальное число одновременных активных сессий составляет 100 пользователей.

2. Подготовить руководство пользователя и администратора.

3. ER диаграмму, описать ее.

4. UML, из них:

Диаграмма вариантов использования

Показывает взаимодействие пользователей (актеров) с системой и основные сценарии её использования. Отображает роли и функции системы с точки зрения пользователя.

Диаграмма последовательностей

Иллюстрирует порядок взаимодействий между объектами системы во времени для выполнения конкретной задачи или сценария. Показывает обмен сообщениями и последовательность действий.

Диаграмма компонентов

Отображает структуру системы в виде компонентов (модулей, сервисов) и их взаимосвязи. Используется для моделирования архитектуры системы и связи между её частями.

Диаграмма пакетов

Организует модель системы в логические группы (пакеты), показывая их иерархию и связи между ними. Помогает структурировать большие системы.

Диаграмма деятельности

Моделирует бизнес-процессы или рабочие потоки, показывая последовательность действий, ветвления, параллельные процессы и условия перехода.

