3 Разработка проектной документации на разработку информационной системы соответствии с

Требованиями заказчика.

Это ключевой этап учебной практики (УП.05), на котором все собранные ранее данные превращаются в формальный документ — **Техническое Задание (ТЗ)** или **Проектную Документацию**, являющуюся руководством к действию для разработчиков.

Разберем процесс и структуру документации в соответствии с требованиями заказчика.

**Разработка проектной документации на информационную систему (УП.05)**

**Цель:** Создать комплекс документов, который однозначно определяет, *ЧТО* должно быть разработано, *КАК* система будет функционировать и при каких условиях, служа основой для заключения договора на разработку и приемки готового продукта.

**Структура проектной документации (Основные разделы)**

Для учебного проекта (УП.05) рекомендуется следующая структура, основанная на реальных стандартах (ГОСТ 34, ГОСТ 19).

**1. Вводная часть (Общие положения)**

* **1.1. Наименование системы:** Полное и сокращенное название ИС.
* **1.2. Основание для разработки:** Приказ по учебному заведению, номер задания на практику.
* **1.3. Назначение системы:** Краткое описание того, для чего создается система и какие бизнес-пробемы она решает (например: "Автоматизация процесса обработки заказов в интернет-магазине").
* **1.4. Цели и задачи создания системы:**
  + **Цели:** Общие стратегические результаты (Повысить скорость обслуживания клиентов на 30%).
  + **Задачи:** Конкретные, измеримые действия (Реализовать функционал ведения каталога товаров, учета складских остатков, оформления онлайн-заказов).
* **1.5. Перечень стадий и этапов разработки:** План-график работ (например: Техническое проектирование, Разработка, Тестирование, Внедрение).

**2. Требования к системе**

Это самый важный и объемный раздел.

* **2.1. Функциональные требования (Что система ДОЛЖНА делать):**
  + Структурируются по модулям или ролям пользователей.
  + **Формат:** Можно оформить в виде таблицы или списка сценариев использования (Use Case).
  + *Пример для роли "Менеджер":*
    - ФР.1. Добавление нового клиента в справочник.
    - ФР.2. Просмотр и редактирование карточки товара.
    - ФР.3. Формирование отчета "Продажи за период".
* **2.2. Требования к надежности:**
  + Время безотказной работы (например, 99.5% в рабочее время).
  + Время восстановления после сбоя (не более 30 минут).
* **2.3. Требования к безопасности:**
  + Разграничение прав доступа по ролям (Гость, Пользователь, Администратор).
  + Требования к паролям (минимальная длина, сложность).
  + Обязательность аутентификации (входа в систему).
  + Шифрование конфиденциальных данных.
* **2.4. Требования к эргономике и интерфейсу:**
  + "Интерфейс должен быть интуитивно понятен для пользователя с базовыми навыками работы с ПК".
  + "Время обучения новогого сотрудника работе с системой не должно превышать 2 часов".
* **2.5. Требования к совместимости и интеграции:**
  + "Система должна интегрироваться с сайтом компании через REST API".
  + "Система должна поддерживать экспорт отчетов в форматах XLSX и PDF".

**3. Описание системы**

* **3.1. Архитектура системы:** Схема, показывающая основные компоненты системы (клиентское веб-приложение, сервер приложений, база данных) и связи между ними.
* **3.2. Программно-технические средства:**
  + **Серверная часть:** ОС (Windows Server/Linux), СУБД (MySQL, PostgreSQL), сервер приложений (Node.js, Apache Tomcat).
  + **Клиентская часть:** Веб-браузер (Chrome, Firefox последних версий).
* **3.3. Структура базы данных:** Модель данных (ER-диаграмма) с описанием основных таблиц (например: Users, Products, Orders).

**4. Состав и содержание работ по созданию системы**

* Перечень этапов, которые должна выполнить команда разработчиков.
* *Пример:*
  1. Разработка пользовательского интерфейса.
  2. Проектирование базы данных.
  3. Реализация серверной логики.
  4. Интеграция с внешними системами.
  5. Тестирование (модульное, интеграционное, системное).

**5. Порядок контроля и приемки системы**

* **5.1. Виды испытаний:**
  + **Предварительные испытания:** Проводятся разработчиком.
  + **Приемочные испытания:** Проводятся заказчиком (преподавателем) по настоящему документу.
* **5.2. Порядок приемки:** "Система считается принятой, если успешно пройдены все приемочные испытания, описанные в Протоколе испытаний, и подписан Акт приемки".

**6. Экономическая эффективность (см. предыдущий ответ)**

* Краткий расчет затрат, экономии и срока окупаемости.

**Процесс разработки документации (Пошаговый план для УП.05)**

1. **Анализ и структурирование исходных данных:**
   * Возьмите все, что собрали на предыдущем этапе (интервью, опросы, документы).
   * Сгруппируйте информацию по будущим разделам ТЗ.
2. **Формализация требований:**
   * Превратите разрозненные пожелания в строгие, проверяемые требования.
   * *Плохо:* "Система должна быть быстрой".
   * *Хорошо:* "Время отклика системы при открытии списка товаров не должно превышать 2 секунд при 100 одновременных пользователях".
3. **Проектирование архитектуры и моделей:**
   * Нарисуйте схему архитектуры (можно в draw.io, Lucidchart).
   * Создайте ER-диаграмму базы данных.
4. **Написание текста документа:**
   * Используйте четкий, однозначный технический язык.
   * Избегайте двусмысленностей. Каждое требование должно быть понято одинаково и вами, и заказчиком, и разработчиком.
5. **Согласование и утверждение:**
   * **Это ключевой момент практики.** Представьте черновик документа вашему руководителю (заказчику).
   * Получите обратную связь, внесите правки.
   * Получите официальное утверждение (подпись на титульном листе). После этого ТЗ становится неизменным ориентиром.

**Типичные ошибки и рекомендации для УП.05**

* **Ошибка:** Требования в виде размытых пожеланий.
  + **Решение:** Использовать формат "Система должна предоставлять возможность [роли] выполнять [действие]".
* **Ошибка:** Отсутствие критериев приемки.
  + **Решение:** По каждому ключевому требованию должен быть понятный способ проверки (например, для требования "Добавление клиента" критерием будет: "В форме вводятся данные, по нажатию кнопки 'Сохранить' запись появляется в общем списке клиентов").
* **Ошибка:** Игнорирование нефункциональных требований (безопасность, надежность).
  + **Решение:** Выделить для них отдельный раздел.
* **Рекомендация:** Используйте визуализацию. Одна схема стоит десяти страниц текста.
* **Рекомендация:** С самого начала договоритесь о формате документа (Google Docs, Word) и средствах проектирования (диаграммы).

**Итог:** Качественно разработанная проектная документация — это не просто "отчет для галочки", а реальный инструмент управления проектом. На защите УП.05 она продемонстрирует ваше умение системно мыслить, формализовать требования и профессионально подходить к созданию сложных объектов.