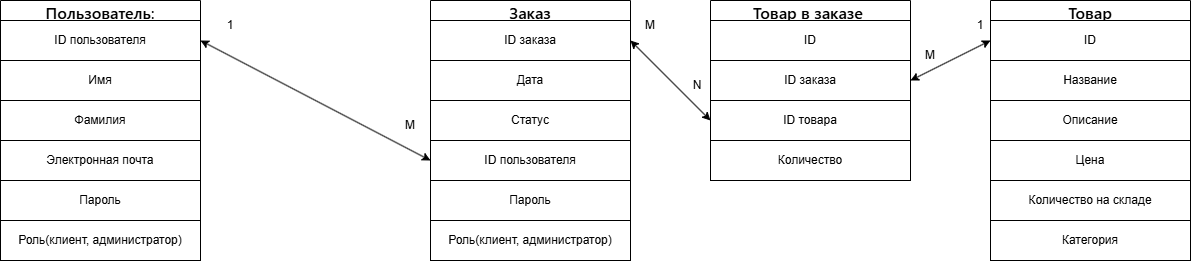
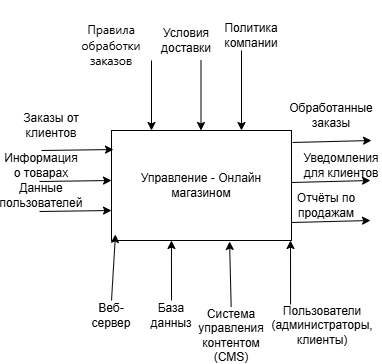
1 – Расписать ТЗ по вашей теме

**Название проекта:** Онлайн-магазин "MebDom"  
  
**1. Введение**  
  
**1.1 Цели проекта:**  
Создание веб-приложения для реализации товаров для дома (мебель, бытовая техника, текстиль и т.д.) через интернет. Онлайн-магазин должен предоставлять пользователям удобный интерфейс для поиска, выбора и покупки товаров с различными условиями доставки и оплаты.  
  
**1.2 Задачи проекта:**  
- Разработка интуитивно понятного и привлекательного дизайна сайта.  
- Обеспечение функционала поиска и фильтрации товаров с минимальными задержками.  
- Интеграция надежной системы обработки заказов и платежей.  
- Построение системы управления товарами с возможностью добавления, редактирования и удаления.  
- Интеграция с системами доставки и автоматизированным учетом товаров на складе.  
  
  
  
**2. Общие сведения**  
  
**2.1 Платформа:**  
- Веб-приложение, доступное через современные браузеры (Chrome, Firefox, Safari).  
- Адаптивный дизайн для работы на мобильных устройствах и планшетах.  
  
**2.2 Технологии:**  
- Серверная часть: JavaScript (Node.js).  
- Клиентская часть: HTML5, CSS3, JavaScript React   
- База данных: PostgreSQL.  
- Хостинг: Облачный AWS  
- Интеграция с платежными системами   
- Интеграция с системами доставки   
  
  
**3. Функциональные требования**  
  
**3.1 Пользовательский функционал:**  
  
- **Регистрация и авторизация:**  
- Вход на сайт через электронную почту и пароль.  
- Восстановление пароля через электронную почту.  
- Возможность авторизации через социальные сети (Facebook, Google).  
  
- **Каталог товаров:**  
- Отображение списка товаров с фотографиями, названиями и ценами.  
- Функция фильтрации по категориям (мебель, бытовая техника, текстиль и т.д.) и характеристикам (цвет, размер, цена).  
- Реализация многоуровневой навигации (категории и подкатегории товаров).  
  
- **Поиск товаров:**  
- Поиск товаров по ключевым словам с подсказками.  
- Сортировка товаров по цене, новизне, популярности и рейтингу.  
  
- **Карточка товара:**  
- Подробная информация о выбранном товаре: фотографии, описание, характеристики, цена.  
- Кнопка "Добавить в корзину" и возможность оформления заказа непосредственно из карточки товара.  
- Отзывы и рейтинги покупателей.  
  
- **Корзина:**  
- Возможность просмотра добавленных товаров с изменением количества и удалением.  
- Расчет общей стоимости корзины, включая информацию о скидках и акциях.  
  
- **Оформление заказа:**  
- Этапы оформления заказа: выбор способа доставки и оплаты, ввод данных для доставки (ФИО, адрес, телефон).  
- Возможность применения промокодов для получения скидок.  
- Подтверждение заказа и отображение статуса (на обработке, отправлен, завершен).  
  
- **Личный кабинет:**  
- Просмотр и управление личной информацией (изменение пароля, контактной информации).  
- Просмотр истории заказов, статуса текущих заказов и возможность повторного заказа.  
  
- **Поддержка пользователей:**  
- Чат для быстрого обращения в службу поддержки.  
- База знаний и раздел "Часто задаваемые вопросы" (FAQ).  
  
**3.2 Административный функционал:**  
  
- **Управление товарами:**  
- Возможность добавления, редактирования и удаления товаров с загрузкой изображений.  
- Управление категориями товаров: создание и удаление категорий.  
- Изменение статуса наличия товара (в наличии, под заказ, временно отсутствует).  
  
- **Управление заказами:**

- Просмотр всех заказов с возможностью изменения их статуса (обрабатывается, отправлен, завершен, отменен).  
- Возможность отправки уведомлений пользователям о статусе их заказов.  
  
- **Управление пользователями:**  
- Просмотр всех пользователей с фильтрацией по активным, неактивным и удаленным аккаунтам.  
- Возможность блокировки/разблокировки пользователей.  
  
- **Отчеты и аналитика:**  
- Генерация отчетов по продажам (ежедневные, ежемесячные).  
- Анализ популярности товаров и категорий, а также эффективность промоакций.  
  
  
  
**4. Нефункциональные требования**  
  
**4.1 Производительность:**  
- Время загрузки страницы не должно превышать 2 секунд.  
- Система должна поддерживать не менее 1000 пользователей одновременно.  
  
**4.2 Безопасность:**  
- Хранение паролей с использованием шифрования (например, bcrypt).  
- Регулярные обновления компонентов системы для защиты от уязвимостей и атак.  
- Защита от DDoS-атак и использование HTTPS.  
  
**4.3 Используемость:**  
- Простой и интуитивно понятный интерфейс.  
- Обеспечение доступности для пользователей с ограниченными возможностями (соответствие стандартам WCAG).  
- Поддержка мультиязычности (по мере необходимости).  
  
**4.4 Техническая поддержка и документация:**  
- Предоставление документации API и руководства по использованию для администраторов.  
- Процесс обновления и обслуживания системы без остановки работы.  
  
  
  
**5. Этапы разработки**  
  
**5.1 Подготовительный этап:**  
- Сбор требований и анализ рынка (1 месяц).  
- Определение целевой аудитории и формирование концепции.  
  
**5.2 Дизайн:**  
- Проектирование пользовательского интерфейса (UI/UX) и создание макетов страниц (2 месяца).  
- Прототипирование и получение отзывов от потенциальных пользователей.  
  
**5.3 Разработка:**  
- Этап 1: Реализация пользовательского функционала (3 месяца).  
- Этап 2: Реализация административного функционала (2 месяца).  
- Этап 3: Интеграция платежных систем и служб доставки (1 месяц).  
  
**5.4 Тестирование:**  
- Проведение функционального, нагрузочного и безопасности тестирования (2 месяца).  
- Исправление выявленных ошибок и доработка системы.  
  
**5.5 Запуск:**   
- Официальный запуск онлайн-магазина (через 8 месяцев с начала проекта).  
  
  
  
**6. Бюджет и ресурсы**  
- Оценка бюджета на проект: [1200000].  
- Ресурсы:  
- Команда разработчиков (2-3 программиста, 1 дизайнер, 1 тестировщик).  
- Обслуживание серверов и программного обеспечения.  
  
**7. Ожидаемые результаты**  
- С запуском онлайн-магазина ожидается увеличение продаж на 30% в течение первого года.  
- Уровень удовлетворенности клиентов не менее 85% по результатам опросов.  
- Постоянное обновление ассортимента и улучшение функционала на основе анализа отзывов пользователей.

2 – Составить схемы IDEF0/Er диаграммму







3 – описать какие языки. Технологии следует использовать и почему.

Серверная часть

**JavaScript (Node.js)**:  
  - **Почему**:  
    - Позволяет использовать JavaScript как на клиентской, так и на серверной стороне, что упрощает разработку и понимание кода.  
    - Высокая производительность при обработке асинхронных запросов.  
    - Большое количество доступных пакетов и модулей с помощью npm (Node Package Manager).

Клиентская часть

- **HTML/CSS**:  
  - **Почему**:  
    - Эти языки являются стандартами для создания веб-страниц. HTML используется для разметки содержимого, а CSS — для оформления и стилизации.  
    - Поддержка всех современных браузеров.  
  
- **JavaScript (с библиотеками/фреймворками)**:  
  - **Почему**:  
    - Используется для создания интерактивных элементов на страницах.  
    - Библиотеки, такие как React или Vue.js:  
      - **React**: Позволяет создавать сложные пользовательские интерфейсы с помощью компонентов, имеет активное сообщество и большое количество готовых решений.  
      - **Vue.js**: Легкий в усвоении, подходит для небольших и средних приложений. Позволяет быстро разработать визуальную часть.

База данных

- **PostgreSQL**:  
  - **Почему**:  
    - Мощная реляционная база данных, поддерживающая отсутствие схемы и сложные запросы.  
    - Хорошая поддержка транзакций, безопасной работы с данными и различных типов данных.

**Кэширование и производительность:**  
- **Redis** :  
  - **Почему**:  
    - Используются для кэширования часто запрашиваемых данных, что значительно ускоряет доступ к информации и улучшает производительность приложения.

**Системы управления версиями:**  
- **Git (с использованием GitHub или GitLab)**:  
  - **Почему**:  
    - Обеспечивает контроль версий кода и упрощает совместную работу над проектом.  
    - Позволяет легко откатывать изменения и управлять разными версиями проекта.

**Интеграция с платежными системами:**

**СБП, СПФС:**  
  - **Почему**:  
    - Платежные платформы, предлагающие удобные API для интеграции, множество вариантов оплаты и поддержку безопасности транзакций.

**Хостинг и развертывание:**  
- **AWS**:  
  - **Почему**:  
    - Позволяют масштабировать приложение, обеспечивая надежную работу в интернет-среде.

4 – Создайте репозиторий git на рабочем столе

5 – Создайте удаленный репозиторий на github загрузите в него вашу работу.