**Практическая работа 21: Применение водопадной методологии**

Этап 2: Проектирование (System Design)

1. Общая архитектура системы

Архитектура системы будет состоять из следующих компонентов:

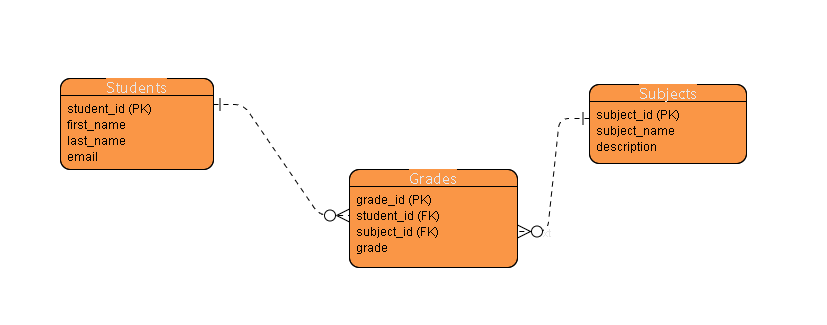
* Frontend: Веб-приложение, разработанное с использованием HTML, CSS и JavaScript. Оно будет обеспечивать интерфейс для студентов и преподавателей, а также хранить данные в локальном хранилище браузера (Local Storage) или использовать JSON-файлы для имитации работы с данными.

2. Стек технологий

* Frontend: HTML, CSS, JavaScript (возможно использование фреймворков, таких как React или Vue.js для улучшения взаимодействия).
* Хранение данных: Local Storage или JSON-файлы для хранения информации о студентах, предметах и оценках.

3. ER-диаграмма базы данных

Поскольку мы не используем бэкенд, данные будут храниться в локальном хранилище. ER-диаграмма будет выглядеть следующим образом:



Сущности:

* Студенты: id, имя, фамилия , email.
* Предметы: id, название, описание.
* Оценки: id, студент\_id, предмет\_id, оценка, дата.

Связи:

* Студенты могут иметь много оценок.
* Предметы могут иметь много студентов.

4. UML-диаграммы

* UML-диаграмма классов:
  + Классы: Student, Subject, Grade.
  + Связи между классами:
    - Student имеет множество Grade.
    - Subject может быть связан с множеством Student

Изображение выглядит как текст, диаграмма, снимок экрана, число

Контент, сгенерированный ИИ, может содержать ошибки.

* UML-диаграмма последовательностей:
  + Сценарий: Студент запрашивает свои оценки.
  + Участники: Студент, Frontend (интерфейс).

**Студент -> Frontend: Запросить оценки**

**Frontend -> Local Storage: Получить оценки студента**

**Local Storage -> Frontend: Вернуть оценки**

**Frontend -> Студент: Показать оценки**

* UML-диаграмма состояний:
  + Состояния: "Зарегистрирован", "Активен", "Завершен" для студента.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.