

WebLab#1

Depends on: undefined

Subject: Анализ/проектирование

Season: 2024

1. Взять за основу большое приложение с базой данных (например, ЛР по ППО, курсовой проект по БД, личный пет-проект и т.д.). Рекомендуется брать уже существующий проект, чтобы не тратить время на разработку бизнес-логики и логики взаимодействия с БД.

Можно взять любую новую идею, которая предполагает работу с 3-5 сущностями в базе данных (или ином хранилище, в т.ч. удаленном), и 2-3 формы взаимодействия с пользователем.

Если идея больше, то допускается формирование групп в 2 человека (с кратным увеличением сложности и объема задачи). Эту ситуацию необходимо согласовать с преподавателем.

С этим приложением будут связаны все последующие лабораторные работы.

2. В .md файле подготовить (или переиспользовать существующие материалы), кратко:
 - a. Цель работы, решаемая проблема/предоставляемая возможность
 - b. Краткий перечень функциональных требований
 - c. Use-case диаграмма системы
 - d. BPMN диаграмма основных бизнес-процессов
 - e. Примеры описания основных пользовательских сценариев
 - f. ER-диаграмма сущностей
 - g. Диаграмма БД
 - h. Компонентная диаграмма системы

- i. Экраны будущего web-приложения на уровне черновых эскизов. Задача данного упражнения - понять, как с приложением должен взаимодействовать пользователь для упрощения проектирования API. Это могут быть классические wireframes, черновики от руки, наброски в PAINT/псевдографике/Figma.

Примечание: здесь не требуется полноценный UI/UX дизайн, требуется примерное распределение функциональности по экранам и возможным компонентам управления.

Примечание

Рекомендации на случай выбора новой темы (рекомендуется продолжать свой проект с ППО):

Требования к приложению.

1. Обязательно использование классического WEB-стека: http, html, css, ts. При этом Backend может быть реализован на C#, Java, Kotlin, Scala, C++, TypeScript, Delphi, Go, Rust, Python (по согласованию с преподавателем и в случае следования всем рекомендациям из методических указаний, публикуемых в курсе ППО).
2. Помимо классического web-приложения, может быть рассмотрено web-приложение, упаковываемое в десктопное и мобильное, а так же telegram bot с WEB UI.
3. Не менее 2-3 экранов с данными. Под “Экраном” понимается не просто html “об авторе”, а целостная страница с данными. Пример такого приложения: Интернет магазин (страница списка товаров, детальный просмотр товара, корзина), платформа, маркетплейс, умный дом и тд.
4. На каждом экране должна быть минимум одна пользовательская активность (кроме пассивного просмотра информации). Продолжая пример с интернет-магазином: Поиск товара и добавление в корзину на списке товаров, Добавление в корзину при детальном просмотре, Покупка в корзине.
5. В проекте должны быть данные. Это могут быть данные, хранящиеся в виде файлов в файловом хранилище, может быть SQL/NoSQL база данных, может быть сервис данных, а может быть и внешняя система, с которой ваш проект взаимодействует по некому API (например, vk, twitter и т.д.). В любом случае в приложении должен быть выделен слой по работе с такими данными.