

Практическое занятие 1:

Вводная часть:

Предполагается 3 практических работы и курсовая работа, они разделены на 5 контрольных работ в СДО – первые 3 это проекты, ещё 2 – это курсовая.

Как будут проходить практики:

На каждую практическую работу нужно сформировать команды из 4 человек (должно быть ровно 4 человека).

В составе команды должны быть распределены следующие роли:

1. Тимлид (project manager)
2. Аналитик
3. Фронтэнд-разработчик
4. Бэкенд-разработчик

На каждую роль можно поставить только одно человека.

В течение семестра роли в команде должны будут меняться, чтобы каждый мог побывать в разных ролях.

В связи с тем, что на 3 проекта 4 человека, предполагается, что каждый участник побудет в роли тимлида и аналитика, а также фронтэнд или бэкенд разработчика, однако, ротация ролей остаётся на усмотрение команды.

Формирование команд:

Команды формируются из 4 человек. В индивидуальных случаях, если кол-во студентов не делится на 4, допускается сделать группу из 3 или 5 человек, но количество таких групп должно быть минимальным.

Не допускается создание команд из 1-2 человек – должно быть не менее 3 и не более 5.

В случае, если в команде 3 человека, то один человек занимает сразу 2 роли, в случае 5 человек, 5й участник команды выбирает роль второго фронтэнд или бэкенд-разработчика.

Проектная работа

В рамках каждой практической работы каждая проектная команда выбирает тему проекта, производит анализ, проектирование, разработку и тестирование программного продукта с последующим написанием отчёта и защитой работы в конце срока выполнения.

Отчёт состоит из следующих пунктов:

1. Введение (1 страница): Описание целей и задач практической работы.
2. Обзор литературы (1 страница): Изучение и анализ доступной литературы и источников по теме проектирования интерфейсов и макетов.
3. Методология (1-2 страницы): Описание подхода, выбранного студентом для выполнения практической работы, объяснение принятых решений и используемых инструментов.
4. Создание макетов (1-2 страницы): Представление разработанных макетов пользовательских интерфейсов, а также их обоснование с точки зрения применения принципов проектирования.

5. Сравнительный анализ инструментов (2-3 страницы): Обзор и оценка различных инструментов для создания макетов и проектирования интерфейсов.

6. Пользовательское тестирование (2-3 страницы): Описание процесса тестирования макетов среди пользователей и анализ полученной обратной связи.

7. Выводы (1 страница): Обобщение результатов работы, анализ полученных данных и выводы по теме исследования.

8. Список использованных источников (1 страница): Перечисление литературы, статей, ресурсов и инструментов, которые были использованы студентом в процессе работы.

Задачи участников команды в отчёте:

Тимлид контролирует общий процесс выполнения работы, пишет введение, заключение, список источников, а также обзор литературы. Тимлид также создаёт GitHub/GitVerse репозиторий проекта и формирует доску задач и распределяет задания в любом удобном сервисе – Teamly, Week, Notion и т.д. – на усмотрение команды.

Аналитик описывает предметную область, методологию, инструменты, формирует UseCase диаграмму, диаграмму пользовательских историй (UserStoryMap) и диаграмму классов (UML), определяет доступный функционал для неавторизованного пользователя (гостя), авторизованного пользователя (клиента) и администратора (продавца/владельца сервиса).

Фронтэнд-разработчик формирует макеты (например, в Figma), выполняет сравнительный анализ инструментов для разработки фронтэнда (библиотеки, фреймворки), выполняет реализацию и тестирование фронта.

Бэкенд-разработчик строит ErD диаграмму, проектирует базу данных, выполняет сравнительный анализ инструментов для разработки бэкенда и

базы данных (библиотеки, фреймворки, СУБД), составляет список API, выполняет реализацию и тестирование бэкенда.

По итогу тимлид соединяет части всех участников команды в единый отчёт и загружает отчёт команды в СДО.

Защита проектов

По итогу выполнения проекта проводится его защита. Защиту выполняет вся команда, путём демонстрации отчёта, демонстрации работы проекта и презентация.

Критерии выбора темы проекта

В качестве проекта выбирается веб или мобильное приложение, с обязательной связью с бэкендом по REST API, тема проекта – свободная. Примеры тем: Система управления складом, система управления магазином и т.п.

Не допускаются такие темы, как телеграмм-бот, десктоп приложение (без связи по REST API), веб-приложение без REST API (Razer Pages, Spring MVC и т.п.), консольные приложения и т.п.

Главная цель – должен быть явно выраженный фронтэнд и явно выраженный бэкенд, как отдельные, самостоятельные проекты, выполняемые разными участниками команды.

Обратите внимание, что в системе должно быть минимум 3 роли:

1. Гость (зашёл на сайт, не зарегистрирован) – может просматривать ограниченный набор страниц (главная, товары, о магазине, но не может формировать корзину).
2. Клиент (зарегистрированный пользователь) – может всё то же самое, что и Гость, дополнительно может формировать корзину, создавать, отменять и отслеживать заказы.

3. Администратор (владелец) – может всё то же самое, что и клиент
(ОБРАТИТЕ ВНИМАНИЕ: роль администратора РАСШИРЯЕТ возможности обычного пользователя, т.е. администратор может тоже покупать в своём же магазине). Дополнительно может редактировать кол-во доступного товара, цены, состав и статусы заказов.

В случае, если тема не связана с магазином, то требования аналогичны выбранной предметной области.

Работа над проектами

На выполнение проекта даётся 4 недели. Срок сдачи первого проекта – 23 февраля (4 неделя).

Этапы работ:

1 неделя – выбор и утверждение темы, моделирование диаграмм, выбор инструментария, составление макетов, списка API, список функциональных возможностей системы и распределение функционала по ролям.

2 неделя – промежуточный отчёт о выполненной работе, корректировка предложенных решений. Начало разработки.

3 неделя – промежуточные результаты разработки, демонстрация черновика отчёта, тестирование системы.

4 неделя – демонстрация отчёта, программы, защита проекта.

Дополнительно

Изучение работы с диаграммами будет рассмотрено на лекции. Также можно самостоятельно изучить на следующих ресурсах:

UseCase - <https://nationalteam.worldskills.ru/skills/proektirovaniye-use-case-diagrammy-opredelenie-funktionalnykh-vozmozhnostey-sistemy/>

UseCase - <https://habr.com/ru/articles/566218/>

Диаграмма классов: <https://habr.com/ru/articles/511798/>

Диаграмма классов: https://flexberry.github.io/ru/gpg_class-diagram.html

ER-диаграмма: <https://practicum.yandex.ru/blog/chto-takoe-er-diagramma/>

ER-диаграмма: <https://habr.com/ru/companies/timeweb/articles/916824/>

UserStoryMap: <https://crmgroup.ru/glossary/user-story-mapping/>

UserStoryMap: <https://practicum.yandex.ru/blog/chto-takoe-user-story-mapping/>

UserStoryMap: <https://habr.com/ru/companies/abdigital/articles/699950/>