

# тестового проекта

## Занятие 2

### ИТ-программное решение для бизнеса

Независимый дизайнер тестового проекта: Рамин Мохаммаддуст

Независимый валидатор тестового проекта: Афшин Дехгани

## Введение

На этом занятии вы займетесь разработкой программного решения для пекарни Belle Croissant Lyonnais, которая планирует модернизировать свою деятельность и расширить спектр предоставляемых услуг. Вы соберете требования, создадите архитектуру системы, модель данных и дизайн пользовательского интерфейса в соответствии с потребностями пекарни.

На этом занятии оцениваются ваши способности по следующим направлениям:

- Сбор и анализ требований различных заинтересованных сторон.
- Разработка масштабируемой и гибкой программной системы.
- Создание интуитивно понятных и удобных для пользователя интерфейсов.
- Эффективное информирование о технических решениях.

## Содержание

Данный учебный пакет содержит следующие материалы:

1. **Session Instructions (PDF):** Подробные инструкции и критерии оценки с четкими наглядными пособиями и схемами.
2. **Belle Croissant Lyonnais Business Model Canvas (PDF):** Одностраничная визуальная сводка о ключевых партнерах пекарни, деятельности, ресурсах, ценностных предложениях, взаимоотношениях с заказчиками, каналах сбыта, сегментах заказчиков, структуре затрат и потоках доходов.
3. **User Personas (PDF):** Подробные описания типичных пользователей (сотрудников пекарни, менеджера, заказчиков), включая их цели, потребности, уязвимые места и технические навыки.
4. **Session Instructions (PDF):** Подробные инструкции с описанием задач, которые необходимо выполнить, и ожидаемых результатов для этого занятия.
5. **Common Folder:** Эта папка содержит дополнительные ресурсы, такие как логотип, значки, руководство по стилю и другие элементы дизайна Belle Croissant Lyonnais, которые могут быть использованы при разработке приложения.

## Описание проекта и задач

На этом занятии вы разработаете основу для программного решения Belle Croissant Lyonnais:

### Методические рекомендации:

1. **Легкость для понимания:** Сделайте схемы и рисунки понятными, с интуитивно понятной конструкцией.
2. **Комплексность:** Охватите все основные аспекты системы.
3. **Точность:** Убедитесь, что все представления соответствуют предоставленной бизнес-модели и персонажам пользователей.
4. **Стандартизация:** Используйте стандартные для отрасли обозначения и методологии.

5. **Своевременность:** Выполняйте все задания в отведенные сроки.

#### Технические особенности:

1. **Анализ требований:** определение ключевых участников, вариантов использования и системных взаимодействий.
2. **Проектирование ERD:** Создайте нормализованную диаграмму, представляющую объекты, атрибуты и взаимосвязи.
3. **Дизайн UI:** Разработайте каркас как для настольных (для персонала), так и для мобильных (для заказчиков) интерфейсов.
4. **Разработка API:** Укажите конечную точку RESTful для управления заказчиками.
5. **Использование инструмента:** Используйте Microsoft Visio для создания диаграмм.

#### Дополнительные факторы:

- Сосредоточьтесь на основной функциональности и расставьте приоритеты по наиболее важным элементам.
- Учитывайте как текущие, так и будущие потребности пекарни.
- Убедитесь, что все диаграммы и рисунки визуально привлекательны и просты в интерпретации.
- Придерживайтесь принципов RESTful и лучших практик при разработке API.
- При разработке пользовательского интерфейса учитывайте предоставленные данные о персонажах пользователей.

## Инструкции для участника

### 2.1 Выявление и анализ требований

#### Цель:

Продемонстрировать способности эффективно собирать и анализировать требования к новой программной системе Belle Croissant Lyonnais и визуально представлять их с помощью стандартной диаграммы.

#### Задачи:

1. Ознакомьтесь с предоставленными документами "Belle Croissant Lyonnais Business Model Canvas" и "User Personas".
2. Определите ключевых вовлеченных в систему участников (пользователей и внешние системы), а также их взаимодействие.
3. Определите варианты использования (действия или процессы), которые каждый участник выполняет в системе.
4. Создайте диаграмму вариантов использования с помощью Microsoft Visio, которая визуально представляет идентифицированных участников, варианты использования и их взаимосвязи.

#### Результаты:

- **Имя файла:** Session2\_UseCaseDiagram.pdf

- **Тип файла:** PDF-файл (.pdf)

#### Дополнительные примечания:

- Сосредоточьтесь на основной функциональности системы и расставьте приоритеты в наиболее важных вариантах использования.
- Учитывайте как текущие, так и будущие потребности пекарни, как указано в документах Business Model Canvas и User Personas.
- Убедитесь, что диаграмма визуально привлекательна и легко поддается интерпретации.
- Не включайте в диаграмму никаких объяснений, обоснований или интерпретаций.

## 2.2 Разработка диаграммы Entity-Relationship Diagram (ERD)

### Цель:

Разработать нормализованную диаграмму сущностей и связей (Entity-Relationship Diagram (ERD)), которая моделирует сущности данных, атрибуты и взаимосвязи для программного решения Belle Croissant Lyonnais.

### Задачи:

1. **Определение сущностей:** На основе функциональных требований и диаграммы классов из задачи 2.2 определите основные сущности, необходимые для системы.
2. **Определение атрибутов:** Для каждого объекта перечислите его соответствующие атрибуты (элементы данных), включая их типы данных и любые ограничения.
3. **Установление связей:** Определите связи между сущностями (например, один к одному, один ко многим, многие ко многим) и укажите количество элементов (например, 1, 0.., 1..) для каждой связи.
4. **Нормализация:** Убедитесь, что ERD нормализована по крайней мере до третьей нормальной формы (3NF), чтобы устранить избыточность и обеспечить целостность данных.
5. **Создание ERD:** Нарисуйте нормализованную ERD с помощью Microsoft Visio, следуя стандартным обозначениям нотации Chen или Crow's Foot .

### Результаты:

- **Имя файла:** Session2\_ERD.pdf
- **Тип файла:** PDF-файл (.pdf)

#### Дополнительные примечания:

- Сосредоточьтесь на основных объектах данных и взаимосвязях, которые необходимы для функционирования системы.
- Для получения дополнительной информации и понимания потребностей пекарни в данных обратитесь к документам Belle Croissant Lyonnais Business Model Canvas и "User Personas" .
- Оценка будет сосредоточена на технической корректности и ясности ERD, а не на конкретной технологии базы данных или деталях реализации.

## 2.3 Дизайн пользовательского интерфейса (UI)

### Цель:

Разработать интуитивно понятные и визуально привлекательные шаблоны для программного решения Belle Croissant Lyonnais, удовлетворяющих потребности как персонала (для настольных компьютеров), так и заказчиков (для мобильных устройств).

### Задачи:

Разработайте шаблоны для следующих экранов интерфейса персонала, которые будут развернуты на рабочем столе:

- **Панель мониторинга:** Обзор продаж, запасов, заказов и данных о заказчиках. Включите ключевые показатели и визуализации.
- **Управление заказами:** Представление, создание, редактирование и обработка заказов (в магазине и онлайн).
- **Управление запасами:** Представление и обновление уровней запасов, отслеживание использования ингредиентов, а также составление отчетов.
- **Управление заказчиками:** Просмотр профилей заказчиков, статуса программы лояльности и истории покупок.

### Результаты:

- **Имя файла:** Session2\_Wireframes\_Staff.pdf
- **Тип файла:** PDF-файл (.pdf)

### Дополнительные примечания:

- Сосредоточьтесь на основной функциональности и работе с пользователем, а не на сложных деталях дизайна.
- Учитывайте индивидуальные особенности пользователей, чтобы адаптировать UI к их потребностям и ожиданиям.

## 2.4 Разработка API

### Цель:

Разработать конечную точку RESTful API специально для управления заказчиками в рамках программного решения Belle Croissant Lyonnais.

### Задача:

1. Проанализируйте функциональные требования, связанные с управлением заказчиками, из задачи 2.1.
2. Разработайте единую конечную точку API, соответствующую принципам и рекомендациям RESTful.
  - Укажите URL-адрес конечной точки (например, /api/customers).
  - Выберите подходящий HTTP-метод(ы) для поддерживаемых операций (например, GET, POST, PUT, DELETE).

- Определите обязательные и необязательные параметры для каждой операции, включая типы данных и правила проверки.
- Опишите структуру данных ожидаемого ответа и типы данных для каждой операции (формат JSON).

**Результаты:**

- **Имя файла:** Session2\_CustomerAPI\_Design.txt
- **Тип файла:** Текстовый файл (.txt)

**Дополнительные примечания:**

- Данная задача сосредоточена исключительно на разработке конечной точки API, а не на ее реализации.
- Пример, приведенный в Sample-API.pdf, — это всего лишь один из возможных способов создания конечной точки. Участникам рекомендуется предлагать свои собственные решения, соответствующие принципам RESTful.
- Оценка будет сосредоточена на правильности и ясности проектирования конечной точки.