# Test Project Session 6

## IT Software Solution for Business

Independent Test Project Designer: Ramin Mohammaddoust

Independent Test Project Validator: Afshin Dehghani

# Introduction

In this session, you will enhance the Belle Croissant Lyonnais mobile app to handle custom orders. Customers will be able to personalize their orders with unique requests and modifications.

The focus is on:

- **Advanced Mobile Development:** Building complex features that cater to specific customer needs.
- **API Integration:** Connecting the app to both the existing product/order API.
- **User-Friendly Design:** Creating an intuitive interface for custom order creation.
- **Problem-Solving:** Handling special requests and ensuring smooth order processing.

# Contents

This session package includes the following materials:

1. **Session Instructions (PDF):** Detailed instructions outlining the tasks to be completed and deliverables expected for this session.
2. **Common Folder:** This folder contains additional resources such as the Belle Croissant Lyonnais logo, icons, style guide, and other design assets that can be used throughout the development of the application.
3. **API Documentation (PDF):** Documentation for the product/customer/order API, providing details about endpoints, request/response formats, and authentication.
4. **Custom Order Requirements (CSV):** A CSV file (custom_order_requirements.csv) detailing allowed customizations and their associated costs per product.

# Description of Project and Tasks

In this session, you will develop a backend system and enhance the Belle Croissant Lyonnais mobile application with advanced features for custom orders.

### Guidelines:

1. **Easy to Use:** Make the interface simple and easy for customers to understand.
2. **Looks Good:** Follow the Belle Croissant Lyonnais Style Guide for the design.
3. **Works Well:** Check that all parts of the application work correctly and without errors.
4. **Secure:** Protect customer data and ensure the application is safe to use.
5. **On Time:** Finish all tasks within the time limit.

### Technical Considerations:

1. **Database Design:** Create a normalized schema for custom order data on MS SQL Server.
2. **API Development:** Implement a RESTful API that integrates with the existing customer/product/order API.

3. **User Interface:** Develop intuitive screens for product catalog and customization.
4. **Data Validation:** Ensure user input is correct and complete, especially for customizations.
5. **Error Handling:** Display clear messages to the user if there are any problems.

### Additional Considerations:

- The application should work smoothly and quickly on provided mobile device.
- Use clear labels and instructions for all UI elements.
- Organize information in a way that is easy for customers to understand.
- Consider edge cases and potential errors in user input and data handling.

# Instructions to the Competitor

## 6.1 Backend Database and API for Custom Orders

### Objective:

Design and implement a backend system (database and API) that can handle custom orders for Belle Croissant Lyonnais, considering the information in the provided documents and ensuring integration with the existing product/order API.

### Tasks:

1. **Database Design:**
    - Create a database schema (Entity-Relationship Diagram - ERD) to store custom order data on the provided MS SQL Server.
    - Ensure the schema is normalized and follows good database design practices.
2. **Backend API Development:**
    - Design and implement a RESTful API using your preferred technology (.NET Web API recommended).
    - The API should provide endpoints for creating, retrieving, and managing custom orders.
    - Ensure the API seamlessly integrates with the existing product/order API.

### Deliverables:

- **Session6_DatabaseCredentials.txt:** Provide the connection string or credentials required to access your database (server name, database name, username, password).
- **Session6_API_Endpoints.txt:** A text file listing all implemented API endpoints, their corresponding HTTP methods, and required parameters.
- **Session6_CustomOrder_ERD.pdf:** A PDF file containing a screenshot of the implemented database in Microsoft SQL Server Management Studio (SSMS) representing the ERD of the database schema.

### Additional Notes:

- You are not required to implement authentication for this API.
- You can use the provided products_cleaned.csv and custom_order_requirements.csv files as references for product and customization data.

## 6.2 Product Catalog

### Objective:

Design and implement a user-friendly product catalog screen within the "Belle Croissant Lyonnais - Custom Order" mobile app, allowing customers to browse, search, filter, and select products for customization.

### Tasks:


1. **Fetch and Display Products:**
    - Fetch the list of available products from the main API endpoint.

- Present products in an organized grid and visually appealing manner.
- For each product, display:
  - Name
  - Image (from Product Images folder)
  - Price
  - Heart icon (initially empty) to indicate favorite status

2. **Search and Filter:**
   - Implement a search bar to allow users to search for products by name.
     - The search should be case-insensitive and match partial words.
     - Display search results in real-time as the user types.
     - Provide an option to clear the search query.
   - Allow users to filter products by category.
     - Display filter options for all available categories.
     - Enable multiple category selections for filtering.
     - Update the displayed products based on the selected filters and search query.
3. **Favorites:**
   - When a user taps the "Heart" icon, add or remove the product from their favorites list.
   - Store the favorites list locally on the device.
   - Provide a "Favorites" filter option to display only the user's saved favorites.
4. **Navigation to Details:**
   - When a product is selected, navigate to the Product Details and Customization screen (Task 6.3).

## Deliverables:

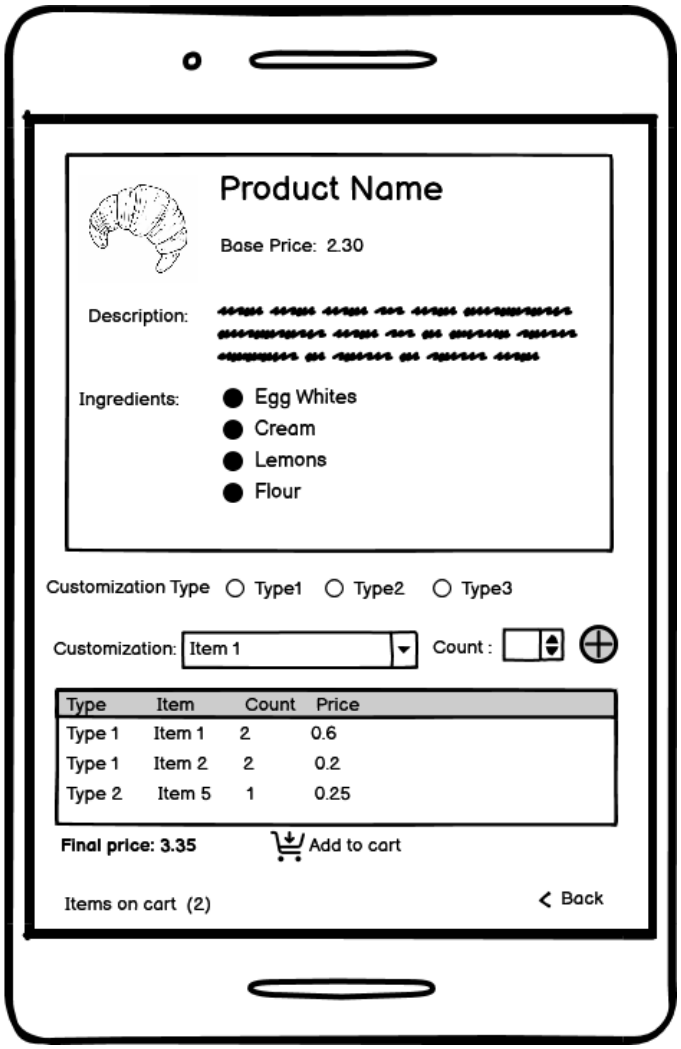
- Implement the Product Catalog screen in the "Belle Croissant Lyonnais - Custom Order" mobile app.

# 6.3 Product Details and Customization

## Objective:

Design and implement a user interface screen within the Belle Croissant Lyonnais mobile app to display detailed product information and allow customers to customize their orders.

## Tasks:




1. **Display Product Details:**
    - Fetch and display the following details for the selected product from the main API:
        - Product name
        - Image (from Product Images folder)
        - Base Price (Price from the main API)
        - Description
        - Ingredients (formatted as a bulleted list for each ingredient)

2. **Customization Options:**
    - Fetch and display the available customization options for the selected product from the database created in task 6.1.
    - Group customizations by type (e.g., Glaze, Topping, Filling).
    - Use appropriate UI elements for each customization type (e.g., checkboxes, radio buttons, dropdowns, text fields).
    - Implement logic to:
        - Display only the customization options that are applicable to the selected product category.
        - Enforce prerequisites and quantity limits for each option (as defined in the created database in task 6.1).
        - Calculate and display the updated price of the product in real time as customizations are selected.
3. **Add to Cart:**
    - Display the final customized product details (name, customizations, quantity, price).
    - "Add to Cart" button:
        - Validate that all required customizations (if any) are selected.
        - Add the customized product to the user's shopping cart.
        - Update the cart icon in the app bar to reflect the number of items in the cart.

## Deliverables:

- Integrate the Product Details and Customization screen into the "Belle Croissant Lyonnais - Custom Order" mobile app.