MAP523 – Mobile App Development iOS
Assignment 2

Instructions:

- Your application will be tested using an **iPhone 11.** It is your responsibility to ensure that your application runs properly on this simulator.
- Besides implementing the required functionality, submissions are required to use the correct coding conventions used in class, professional organization of the code, alignment, clarity of names is all going to be part of the evaluation. Comments for code snippets are recommended but not required.
- When you are ready to submit, upload a **zip file** containing your entire project code to the assessment dropbox. The name of your zip file should be **PIZZA_FIRSTNAME_USERNAME.zip**. Replace FIRST NAME and USERNAME with your first name and userid. **7zip and rar files are not accepted.**

Task:

You have been hired by your local pizza store to build a mobile application that enables customers to order pizza. Using IOS, develop a two screen application that enables the user to enter their order details and generate a receipt indicating the total cost of their order.

The store sells the following types of pizzas. Pizzas must be modelled in the app as a custom Pizza class.

Pizza Name	Description	Price - Small	Price - Medium	Price - Large
Spicy Pulled Pork Pizza	Slow-roasted pulled pork with a special spicy and smoky BBQ pizza sauce.	15.50	17.50	21.50
Vegetarian Fiesta Pizza	Roasted red peppers, caramelized onions, sundried organic tomatoes, feta, and spinach, on a thin crust with pesto sauce.	15.50	17.50	21.50
The Original	Pepperoni, cheese, green onions. Served with extra tomato sauce and three types of cheese.	15.50	17.50	21.50

The application must also provide the following UI components:

1. A **label** that displays the name of the application (example: My Pizza Store)
2. Provide a **picker control** that enables the user to select the type of pizza they want. When the user chooses a pizza, the app's user interface should update to **display the description** of the pizza. HINT: The UIPickerViewDelegate provides a function with the parameter *didSelectRow (ie:* pickerView(:didSelectRow)). This function is automatically called when the user selects an item in a picker view. By default, the picker should be set to **The Original**.
3. Provide a **segmented control** to allow the user to choose the size of the pizza. Sizes are small, medium, large. By default, the control should be set to **Medium**.

4. Provide a **stepper control** to allow the user to choose the quantity of pizzas. The maximum number of pizzas that can be purchased is 5. To successfully place an order, the **user must select a minimum of 1 pizza**. By default, the control is set to **1 pizza**.

5. Provide a **button** labelled **START OVER**. Tapping this button enables the user to clear/reset their order. When the button is pressed, the app should reset all UI controls to their default values. Any displayed receipts should be cleared/hidden from the user.

6. Provide a **button** labelled PLACE ORDER. When the button is tapped, display an **AlertController (popup)** with an **OK** and **CANCEL** button. The popup should ask the user "Are you sure you are ready to place this order?". If the user taps CANCEL, then dismiss the popup and do not perform any other operations.

   If the user taps OK, send the selected Pizza, size, and quantity to a second screen.

   On the second screen, use the received information to calculate and display a receipt for the order. The receipt should display:
   - Name of the pizza that was ordered
   - Size of pizza
   - Quantity
   - Subtotal
   - Tax (13% of subtotal price)
   - Final total

Grading Criteria

- User Interface: All required UI controls are provided. Design of the application is clean and polished. Text is legible.
- Functionality: Implemented per requirements.
- App compilation: App compiles and does not crash.