

Project Requirements for the Pizza Store Application

Objective:

To use an Object-Oriented approach to develop a comprehensive pizza store application that streamlines recipe management, inventory control, menu customization, and order processing, empowering the store owner to manage their business effectively.

Features:

1. Recipe Management:
 - a. Create, edit, and delete pizza recipes.
 - b. Store detailed information about each recipe, including ingredients and quantities.
 - c. Categorize recipes based on pizza type (e.g., vegetarian, meat lovers, specialty).
 - d. Implement search functionality to easily find specific recipes.
2. Ingredient Inventory Management:
 - a. Maintain an inventory of all pizza ingredients.
 - b. Track ingredient usage based on recipe requirements.
 - c. Generate alerts for low-stock ingredients to ensure adequate supply.
3. Standard Menu Management:
 - a. Create and manage a standard menu of pizzas.
 - b. Display each pizza with its name, description, ingredients, and price.
 - c. Categorize pizzas based on type, size, or other relevant criteria.
4. Customer-Customized Pizza Ordering:
 - a. Provide customers with the option to create their own pizzas.
 - b. Offer a selection of pizza bases, sauces, toppings, and additional ingredients.
 - c. Allow customers to customize the quantity of each ingredient.
 - d. Display the total price of the customized pizza based on ingredient choices.
5. Side Dish Management:
 - a. Create and manage a menu of side dishes.
 - b. Display each side dish with its name, description, and price.
 - c. Categorize side dishes based on type (e.g., appetizers, desserts, beverages).
6. Order Processing:
 - a. Capture customer orders for both standard pizzas and customized pizzas.
 - b. Display order details, including pizza selection and side dishes.
 - c. Generate order slips for kitchen staff to prepare orders accurately.
7. User Interface and Accessibility:
 - a. Design a user-friendly interface that is easy to navigate for store owners.

Currently, the store owner relies on paper forms to record customer orders, as illustrated below. This form can serve as a guide for identifying various objects and encapsulating all the necessary information into one or more classes. However, it is important to note that the objects are NOT constrained by the information defined on the order form. Therefore, you should also consider all the additional requirements mentioned previously. To achieve reusability, flexibility, and extensibility, your design should adhere to object-oriented principles and concepts such as Abstraction and Encapsulation. To ensure data persistence, files will be utilized for data storage. Consequently, a 3-layer architecture is recommended, with classes categorized into Presentation, Business, and Data layers.

Choose your pizza(s), call, fax or bring this order to
 SFBU Voice or Fax 510 - 555 - 7777
MINIMUM SUGGESTED NOTICE 48 HOURS

Delivery Date

Delivery Time

DeliWorks Feature Combinations

Qty	16" Extra Large, 12 slices - all start with sauce & Mozzarella		@
<input type="checkbox"/>	Pepperoni	Pepperoni	15.50
<input type="checkbox"/>	Hawaiian	Ham, Red Onion, Pineapple	18.50
<input type="checkbox"/>	Deluxe	Pepperoni/Bacon/Mushrooms/Olives/Peppers/Onion	19.50
<input type="checkbox"/>	Meat Lovers	Pepperoni, Ham, Bacon	18.50
<input type="checkbox"/>	Vegetarian	Mushrooms, Olives, Onion, Peppers, Tomato	19.50
<input type="checkbox"/>	BBQ Chicken	Chicken, Red Onion, BBQ Sauce, Cheddar	19.50

BUILD YOUR OWN OPTIONS Ex. Large 16", 12 slices

Basic with Sauce and Mozzarella \$14.00 each item add 1.75

Pizza 1

<input type="checkbox"/>	Pepperoni
<input type="checkbox"/>	Ham
<input type="checkbox"/>	Bacon
<input type="checkbox"/>	Mushrooms
<input type="checkbox"/>	Onion
<input type="checkbox"/>	Black Olives
<input type="checkbox"/>	Tomato Slices
<input type="checkbox"/>	Extra Cheese
<input type="checkbox"/>	Pineapple

Pizza 2

<input type="checkbox"/>	Pepperoni
<input type="checkbox"/>	Ham
<input type="checkbox"/>	Bacon
<input type="checkbox"/>	Mushrooms
<input type="checkbox"/>	Onion
<input type="checkbox"/>	Black Olives
<input type="checkbox"/>	Tomato Slices
<input type="checkbox"/>	Extra Cheese
<input type="checkbox"/>	Pineapple

Pizza 3

<input type="checkbox"/>	Pepperoni
<input type="checkbox"/>	Ham
<input type="checkbox"/>	Bacon
<input type="checkbox"/>	Mushrooms
<input type="checkbox"/>	Onion
<input type="checkbox"/>	Black Olives
<input type="checkbox"/>	Tomato Slices
<input type="checkbox"/>	Extra Cheese
<input type="checkbox"/>	Pineapple

Qty

SIDES

<input type="checkbox"/>	Caesar Salad
<input type="checkbox"/>	Tossed Salad
<input type="checkbox"/>	Asst. Pop
<input type="checkbox"/>	Asst. Juices
<input type="checkbox"/>	Water
<input type="checkbox"/>	Cookies

@

2.29
2.29
1.35
1.45
1.00
1.00

Name

Company

Phone

email

Submission Requirements:

- Use Case Diagram and Class Diagram: Submit the use case diagram and class diagram in either PDF or PNG format.
- Python Source Code and Data Files: Provide the complete Python source code and any associated data files required for program execution.
- Program Execution Instructions: Include clear and detailed instructions on how to run the program. Ensure that testers can execute the program without entering all data manually. Verify that all necessary data is stored in the data files.
- Test Cases and Results: Submit a comprehensive document containing your test cases and corresponding test results. Each test case should have a detailed description, the expected outcome, and the actual outcome, including screenshots if applicable.