Restaurant Reservation System

Presented by :

Ahmed Hossam 20100867

Ahmed Hany 20100547

Classes structure and relationships

Structure

- RestaurantReservation is the main class that represents the restaurant reservation system. It maintains a list of tables, menu items, and orders. It also has a reference to the DiscountInterface interface, which allows for applying different discounts based on the membership level.
- The class has a Singleton pattern implementation, ensuring that only one instance of RestaurantReservation can exist.
- It provides methods for adding tables (addTable), menu items (addMenuItem), and creating orders (createOrder).
- The getAvailableTables method returns a list of available tables based on the provided date and party size.
- The getMenuItems method returns the list of menu items available in the restaurant.
- The getOrders method returns the list of orders placed.
- DiscountInterface is an interface that defines the contract for calculating discounts. It is implemented by the classes GoldMember, SilverMember, and BronzeMember, which represent different membership levels in the restaurant.
- Table represents a table in the restaurant. It has properties such as tableNumber, capacity, and isAvailable. The class provides methods for reserving a table (reserveTable), freeing a table (freeTable), checking if a table is available (isAvailable), and getting the table's capacity (getCapacity).
- MenuItem represents a menu item in the restaurant. It has properties like name, description, and price. The class provides methods to retrieve these properties (getName, getDescription, getPrice).
- Order represents an order placed by a customer. It includes a list of menu items (items), the table associated with the order (table), and the total price (totalPrice).

- The Order class has methods for adding (addMenuItem) and removing (removeMenuItem) menu items from the order. These methods update the total price.
- The getItems method returns the list of menu items in the order.
- The getTotalPrice method returns the total price of the order.
- Person is a base class representing a general person. It has properties such as name, phoneNumber, and email.
- Customer is a derived class that extends the Person class. It adds no additional properties but represents a customer in the system.

Relationships

1. Composition : The RestaurantReservation class has a composition relationship with the Table class. It maintains a list of tables (tables), which are created and managed within the RestaurantReservation class.

   The RestaurantReservation class also has a composition relationship with the MenuItem class. It maintains a list of menu items (menu), which are created and managed within the RestaurantReservation class.

   The Order class has a composition relationship with the MenuItem class. It maintains a list of menu items (items), representing the items included in the order.
2. Association : The RestaurantReservation class has an association with the Order class. It maintains a list of orders (orders), representing the orders placed in the restaurant.

   The RestaurantReservation class has an association with the DiscountInterface interface. It has a reference to the discountInterface object, which allows for applying different discounts based on the membership level.

   The Order class has an association with the Table class. It has a reference to the table (table) associated with the order.
3. Inheritance: The GoldMember, SilverMember, and BronzeMember classes inherit from the DiscountInterface interface. They provide specific implementations of the discount calculation methods.

   The Customer class inherits from the Person class, representing a specific type of person.

GUI And How It Works

The Restaurant Reservation System is a JavaFX application that allows users to view available tables, create orders, and apply discounts. It provides an interactive interface for managing reservations and menu items in a restaurant.

Key Features:

1. View Available Tables: Users can check the availability of tables based on the party size they enter. The system displays a list of available tables or a message if no tables are available.
2. Create Order: Users can create orders by selecting a table and choosing menu items. The system calculates the total price, applies discounts based on the selected membership level, and displays an order summary.
3. Discount Calculator: The system includes a DiscountCalculator class that handles discount calculations. It offers different discount rates based on the selected membership level (Gold, Silver, Bronze, or Non-Member).
4. Data Persistence: The system uses text files to store data for tables, menu items, and orders. The file paths are specified as constants in the code.

Implementation Overview:

The HelloApplication class extends the Application class and serves as the entry point for the application.

The start method initializes the RestaurantReservation instance, adds initial tables and menu items, and creates the main application window.

The createScene method creates the main scene with buttons for viewing tables, creating orders, and exiting the application.

The displayAvailableTables method shows a new window with the available tables based on the party size entered by the user.

The getPartySize method prompts the user to enter the party size and returns the entered value.

The createOrder method allows the user to select a table and menu items to create an order. It also applies discounts and displays the order summary.

The selectMenuItems method presents the user with a list of menu items to choose from. It allows multiple selections and returns the selected items.

The displayOrderSummary method shows the order summary in a separate window.

The DiscountCalculator class handles discount calculations based on the selected membership level. It prompts the user to select a membership level and returns the calculated discounted price.