Coffee Shop Simulation

1. System Overview

The **Coffee Shop Simulation** is a Java-based application that models a coffee shop's order management system. It incorporates the following key features:

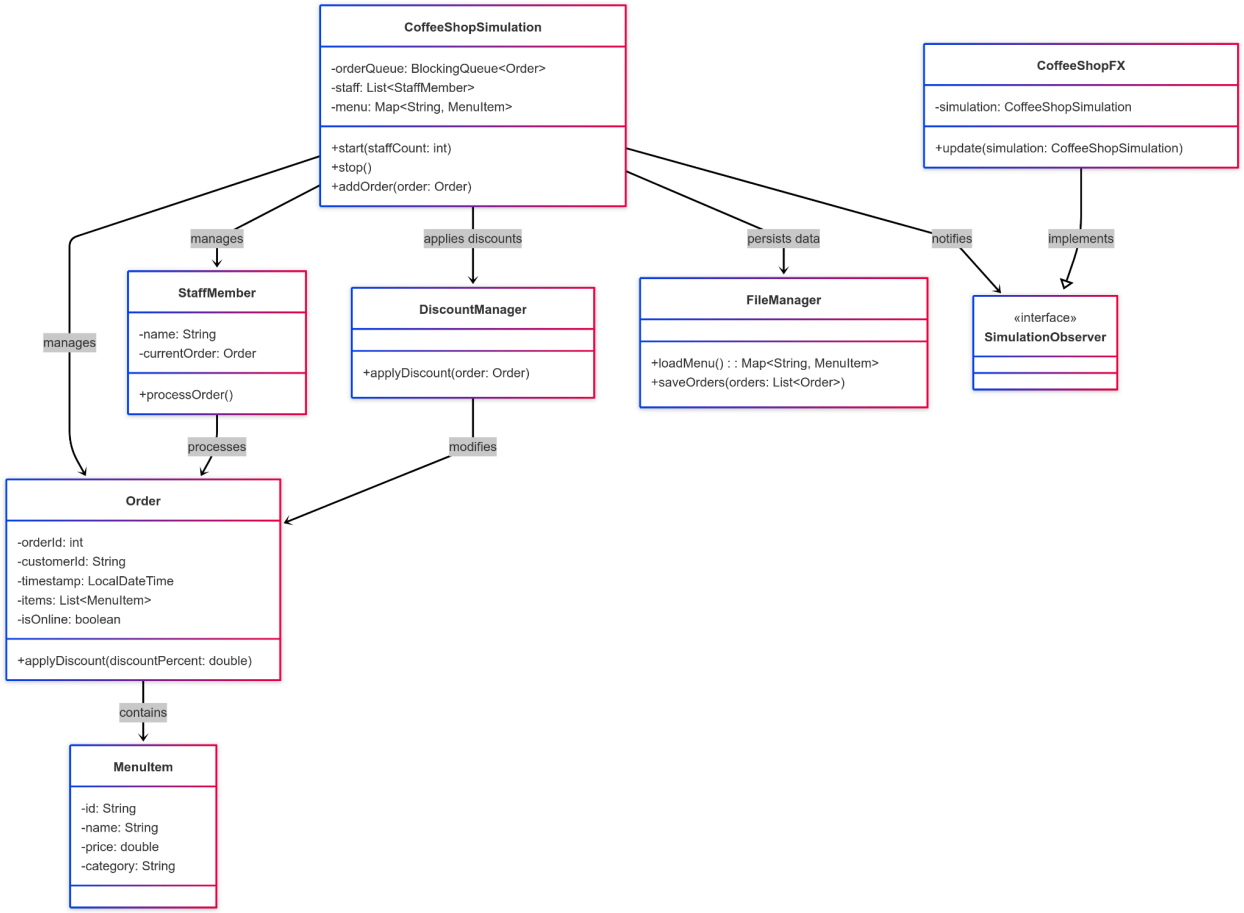
- **Order Processing**: Handles both in-store and online orders using a priority queuing system.
- **Staff Simulation**: Simulates concurrent order processing by multiple staff members.
- **Discount Rules**: Applies automatic discounts based on time, item count, and category combinations.
- **Graphical User Interface (GUI)**: Provides a real-time monitoring and control interface using JavaFX.
- **Persistence**: Utilizes CSV-based storage for maintaining data on orders and menu items.
- **Reporting**: Generates summary reports containing revenue, popular item, and customer statistic data.

2. Key Design Patterns

The following table outlines the design patterns used within the application:

Pattern	Purpose	Where Applied
Observer	Notifies the UI of simulation updates.	CoffeeShopSimulation ↔ CoffeeShopFX (via SimulationObserver)
Factory Method	Creates different types of orders.	Order constructors (for in-store and online orders)
Strategy	Enables dynamic discount calculation.	DiscountManager applies different discount rules.
Singleton	Ensures a single instance of FileManager.	Implicitly used for file operations.
Command	Encapsulates simulation actions (start/stop).	GUI buttons trigger encapsulated simulation commands.
Decorator	Extends order functionality (e.g., discounts).	Order.applyDiscount() modifies the order total by applying discounts.

3. UML Class Diagram



4. Class Responsibilities

Core Classes

Class	Responsibility
CoffeeShopSimulation	Main controller; manages orders, staff, and the simulation lifecycle.
Order	Represents an order, including details such as ID, customer, items, and discounts.
MenuItem	Defines a menu item, including its ID, name, price, and category.
StaffMember	Represents a staff member, modeled as a thread that processes orders.
DiscountManager	Applies dynamic discounts based on defined business rules.

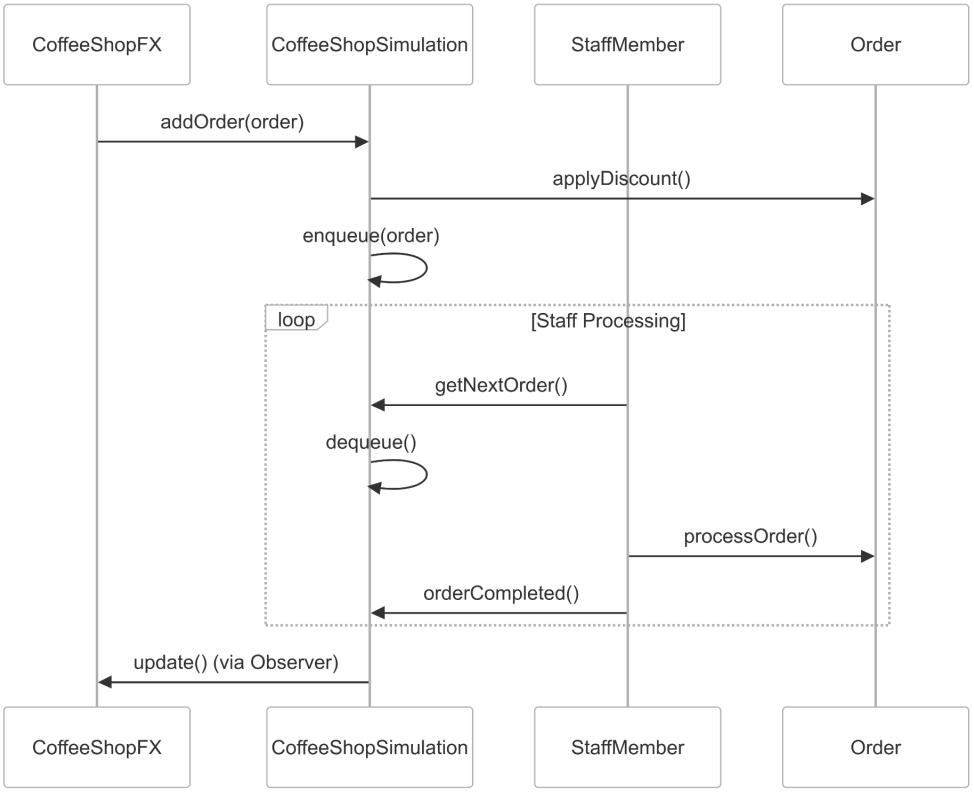
GUI Classes

Class	Responsibility
CoffeeShopFX	Main JavaFX window; displays the order queue and staff status.
NewOrderDialogFX	Dialog used for creating new orders within the application.

Utility Classes

Class	Responsibility
FileManager	Handles CSV input/output operations for orders and menu items.
SimulationObserver	Interface for UI updates, adhering to the Observer pattern.

5. Sequence Diagram: Order Processing



6. Discount Rules Implementation

The DiscountManager class employs the **Strategy Pattern** to apply various discount rules. The following logic is used:

```
Happy Hour Discount (8:00 AM - 10:00 AM):
if (orderTime.isBetween(8, 10)) {
  applyDiscount(0.15); // 15% discount
}
Bundle Discount (3+ Items):
if (order.getItems().size() >= 3) {
  applyDiscount(0.10); // 10% discount
}
Combo Discount (Coffee + Food):
if (order.hasCategory("Coffee") && order.hasCategory("Food")) {
  applyDiscount(0.20); // 20% discount
}
```

7. Data Persistence

File Structure

```
data/
├── menu.csv          (ID,Name,Price,Category)
├── orders.csv         (OrderID,CustomerID,Timestamp,ItemID,IsOnline)
├── coffee_shop_log.txt
└── coffee_shop_report.txt
```

CSV Format Examples

menu.csv

```
COF1,Espresso,2.50,Coffee
FD1,Croissant,3.00,Food
```

orders.csv

```
1000,C123,2023-10-01 09:00:00,COF1,true
```

8. Extending the System

Adding New Features

1. **New Discount Rules**: Implement additional discount strategies by extending the DiscountManager.
2. **Loyalty Program**: Modify the Customer class to track customer points and implement reward redemption.
3. **Advanced Reporting**: Add new methods to the FileManager class to generate custom reports.

Modifying the GUI

- **New UI Components**: Extend CoffeeShopFX with additional tabs or panels, such as an analytics dashboard.
- **Theming**: Customize the look and feel by modifying color constants (e.g., PRIMARY_COLOR, SECONDARY_COLOR).