|  |  |
| --- | --- |
| Component: Customer |  |
| SuperClasses: n/a | **SubClasses:** n/a |
| Responsibilities | **Collaborators** |
| Provide personal details (name, contact information) | n/a |
| Make reservations (‘makeReservation()’) | Reservation |
| Place orders {‘placeOrder()’) | Order |
| Receive updates on order status | Order |
| Save customer details to the database | n/a |

Explanation:

The Customer class represents the end-user of the restaurant information system. It holds essential details about the customers, including their contact information which is crucial for managing reservations and orders. The class allows customers to interact with the system, such as making and tracking their reservations and orders, ensuring a personalized and efficient customer service experience.

|  |  |
| --- | --- |
| Component: Reservation |  |
| SuperClasses: n/a | **SubClasses:** n/a |
| Responsibilities | **Collaborators** |
| Record reservation details (date, time, number of guests) | Customer, Table |
| Manage reservation statuses (booked, cancelled, confirmed) | Table (Conceptual) |
| Check and update table availability | Table (Conceptual) |
| Save reservation details to the database | n/a |

Explanation:

The Reservation class manages all aspects of booking a table at the restaurant. It is designed to hold the reservation's specifics and keep track of the status of each booking. Collaborating closely with the Customer class for inputs and the Table class for availability checks, it ensures that reservations are managed effectively, reducing the chance of overbooking, and improving the utilization of the restaurant's seating capacity.

|  |  |
| --- | --- |
| Component: Order |  |
| SuperClasses: n/a | **SubClasses:** n/a |
| Responsibilities | **Collaborators** |
| Track order items | MenuItem |
| Calculate and update total cost | Invoice |
| Manage order lifecycle (open, in preparation, served) | Kitchen |
| Communicate with Kitchen to update on order status | Kitchen |
| Save order details to the database | n/a |

Explanation:

The Order class is central to the restaurant's operations, tracking each customer's selection of menu items. It's responsible for calculating the total cost, incorporating any additions or deletions to the order, and managing the lifecycle of the order as it progresses from being placed to being served. The Order class collaborates closely with Customer to reflect their choices, MenuItem to detail the items ordered, Invoice to generate a bill for payment, and Kitchen to ensure that the meal is prepared according to the order specifications.

|  |  |
| --- | --- |
| Component: MenuItem |  |
| SuperClasses: n/a | **SubClasses:** n/a |
| Responsibilities | **Collaborators** |
| Maintain item details (name, description, price) | Order |
| Load menu items from the database | n/a |
| Provide information for order customizations | Order |

Explanation:

The MenuItem class acts as a digital representation of the restaurant's offerings, detailing each dish or drink, its price, and description. It's a vital part of the order process, providing the necessary information to both the customer placing the order and the kitchen preparing it. The class also includes functionality to update the availability of items, which helps in managing the inventory and informing customers about out-of-stock items.

|  |  |
| --- | --- |
| Component: Invoice |  |
| SuperClasses: n/a | **SubClasses:** n/a |
| Responsibilities | **Collaborators** |
| Generate itemized bill from orders (‘generateInvoice()’) | Order |
| Process payments (‘processPayment()’) | Payment |
| Record transaction details | Payment |
| Provide digital or printed receipts | Customer |

Explanation:

The Invoice class encapsulates financial transactions within the restaurant. It generates a detailed bill based on the customer's order and handles the payment process, ensuring accurate and secure transactions. The class is responsible for issuing receipts and maintaining a record of all financial activities, which is crucial for accounting and customer service.

|  |  |
| --- | --- |
| Component: Payment |  |
| SuperClasses: n/a | **SubClasses:** n/a |
| Responsibilities | **Collaborators** |
| Handle payment processing (credit card, cash, online) (‘makePayment()’) | Invoice |
| Validate payment methods and details | n/a |
| Securely manage and store transaction data | n/a |

Explanation:

The Payment class is responsible for all aspects of transaction processing within the restaurant information system. It supports various payment methods, ensuring flexibility and convenience for customers. The class collaborates with Invoice to finalize transactions and is designed to safeguard sensitive financial data, adhering to the highest security standards.

|  |  |
| --- | --- |
| Component: Kitchen |  |
| SuperClasses: n/a | **SubClasses:** n/a |
| Responsibilities | **Collaborators** |
| Prepare dishes as per order requests | Order |
| Load orders from the database | n/a |
| Communication with front of house regarding order status | n/a |
| Update order status | n/a |

Explanation:

The Kitchen class is at the heart of the restaurant's food preparation operations. It manages the flow of orders from the dining area to the kitchen staff, ensuring that meals are prepared correctly and efficiently. The class facilitates communication between the kitchen and the service staff, enabling timely updates about order statuses and contributing to a smooth dining experience.

|  |  |
| --- | --- |
| Component: Table |  |
| SuperClasses: n/a | **SubClasses:** n/a |
| Responsibilities | **Collaborators** |
| Track occupancy status | Reservation |
| Hold information about location and capacity | n/a |
| Be reserved or released based on customer arrivals/departures | Reservation |

Explanation:

The Table class is an abstraction of the physical dining space within the restaurant. It keeps track of each table's occupancy status and features such as capacity and location. By collaborating with the Reservation class, it helps to efficiently manage the restaurant's seating arrangements, thereby optimizing space utilization and customer flow.