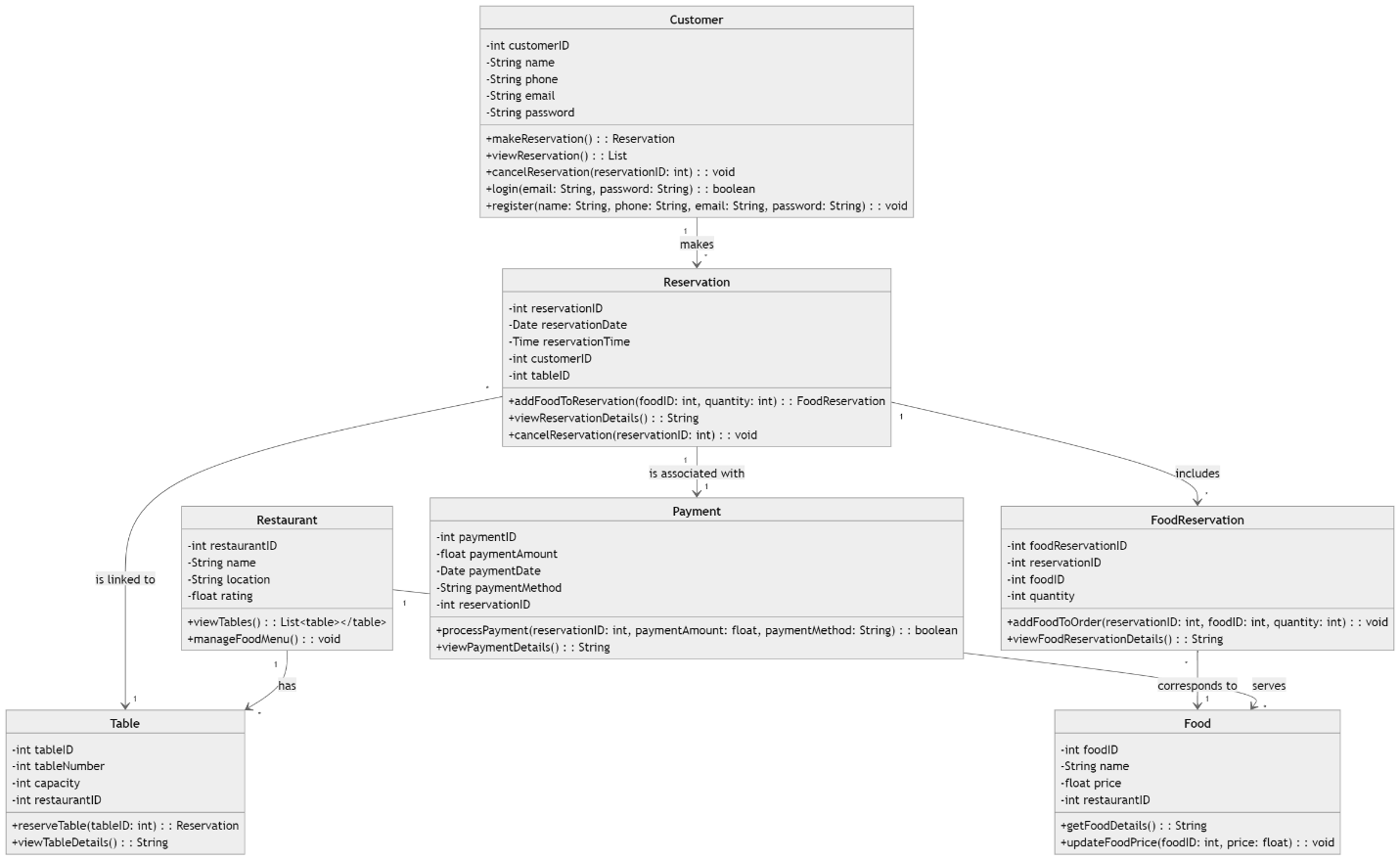
Class Diagram for Restaurnt reservation



**Class Diagram for Restaurant Reservation System**

**1. Customer**

* **Attributes:**
  + customerID: int
  + name: String
  + phone: String
  + email: String
  + password: String (for login and registration)
* **Methods:**
  + +makeReservation(): Reservation
  + +viewReservation(): List<Reservation>
  + +cancelReservation(reservationID: int): void
  + +login(email: String, password: String): boolean
  + +register(name: String, phone: String, email: String, password: String): void

**2. Restaurant**

* **Attributes:**
  + restaurantID: int
  + name: String
  + location: String
  + rating: float
* **Methods:**
  + +viewTables(): List<Table>
  + +manageFoodMenu(): void

**3. Table**

* **Attributes:**
  + tableID: int
  + tableNumber: int
  + capacity: int
  + restaurantID: int (Foreign Key)
* **Methods:**
  + +reserveTable(tableID: int): Reservation
  + +viewTableDetails(): String

**4. Reservation**

* **Attributes:**
  + reservationID: int
  + reservationDate: Date
  + reservationTime: Time
  + customerID: int (Foreign Key)
  + tableID: int (Foreign Key)
* **Methods:**
  + +addFoodToReservation(foodID: int, quantity: int): FoodReservation
  + +viewReservationDetails(): String
  + +cancelReservation(reservationID: int): void

**5. Food**

* **Attributes:**
  + foodID: int
  + name: String
  + price: float
  + restaurantID: int (Foreign Key)
* **Methods:**
  + +getFoodDetails(): String
  + +updateFoodPrice(foodID: int, price: float): void

**6. FoodReservation**

* **Attributes:**
  + foodReservationID: int
  + reservationID: int (Foreign Key)
  + foodID: int (Foreign Key)
  + quantity: int
* **Methods:**
  + +addFoodToOrder(reservationID: int, foodID: int, quantity: int): void
  + +viewFoodReservationDetails(): String

**7. Payment**

* **Attributes:**
  + paymentID: int
  + paymentAmount: float
  + paymentDate: Date
  + paymentMethod: String
  + reservationID: int (Foreign Key)
* **Methods:**
  + +processPayment(reservationID: int, paymentAmount: float, paymentMethod: String): boolean
  + +viewPaymentDetails(): String

**Relationships:**

1. **Customer ↔ Reservation**: A Customer can make multiple Reservations (one-to-many).
2. **Reservation ↔ Table**: Each Reservation is linked to a specific Table (many-to-one).
3. **Reservation ↔ FoodReservation**: A Reservation can include multiple FoodReservations (one-to-many).
4. **FoodReservation ↔ Food**: Each FoodReservation corresponds to a specific Food item (many-to-one).
5. **Reservation ↔ Payment**: Each Reservation is associated with a single Payment (one-to-one).
6. **Restaurant ↔ Table**: A Restaurant has multiple Tables (one-to-many).
7. **Restaurant ↔ Food**: A Restaurant serves multiple Food items (one-to-many).