Restaurant Management System

This Restaurant Management System project is a Java-based console application integrated with PostgreSQL using JDBC. The system is designed to handle core operations of a restaurant, including table bookings, order placement, billing, and payment. It is structured using a layered architecture, with packages organized under `com.restaurant`. The `model` package contains entity classes that represent database tables such as `Table`, `Item`, `Order`, `OrderItem`, and `Bill`. The `dao` package handles direct database access logic, while the `service` package implements the business logic by calling methods from the DAO layer. Although the current application is console-driven, it follows a controller-like flow within the `Main` class and is easily extensible to REST or GUI interfaces. Database configuration is managed through a dedicated class in the `config` package.

The application begins with a menu-driven interface in `Main.java` that allows the user to perform multiple actions, such as viewing and booking available tables, placing orders, generating bills, and processing payments. The table booking flow works by listing unbooked tables from the database. Once a user selects a table, the system updates its status to booked. During order placement, the system allows selection of menu items and their quantities. It inserts a new order record and links it to the selected table, followed by inserting multiple order items tied to the order ID. Each new order is initialized with a status of `Pending`. In a real-world scenario, this status could later be updated to `Prepared` by kitchen staff, simulating kitchen workflow.

Once an order is marked as prepared, a bill can be generated. The system calculates the total based on the ordered items and creates a new bill record with a default `Unpaid` status. Upon receiving payment, the system updates the bill's status to `Paid` and records the payment timestamp. The flow is designed to allow multiple interactions until the user explicitly exits the application, giving it a seamless and repeatable transactional structure.

To run the system, the evaluator must first create the PostgreSQL database named `restaurant_db` and execute the provided SQL script to create the necessary tables (`tables`, `items`, `orders`, `order_items`, `bills`). Sample data is also provided to populate the tables with initial values. The database credentials must be updated in the `DBConnection.java` class. Once the database is set up, the project can be opened in any Java IDE (e.g., IntelliJ, Eclipse), and the `Main` class can be run to start interacting with the application. No third-party libraries are used—only standard JDBC and core Java utilities.

All source code is available in the GitHub repository at:
GitHub

The evaluator is advised to test flows such as booking a table, placing an order, marking it prepared, generating a bill, and completing the payment to experience the full functionality of the system.