**Restaurant Order Management System**

# Project Structure

**ProjectName:**- RestaurantOrderManagementSystem

**Packages:**

1.src/main/java

2.src/main/resources

3.src/test/java

4.src/test/resources

## Maven Dependencies-

1.log4j

2.lombok

3.mysql connector

4.protobuf

**Logs**-

Restaurant-order-management-system.log

# Packages and Classes Under It

1.src/main/java

* org.techhub.clientApp

RestaurantOrderManagementSystem.java

* org.techhub.Model
  1. CustomerModel.java
  2. StaffModel.java
  3. UserModel.java
  4. MenuModel.java
  5. DishModel.java
  6. TableModel.java
  7. OrderModel.java
  8. BillModel.java
* org.techhub.Repository
  1. DBConfig.java
  2. DBUser
  3. CustomerRepository.java(Interface)
  4. CustomerRepositoryImp.java
  5. StaffRepository.java(Interface)
  6. StaffRepositoryImp.java
  7. MenuRepository.java(Interface)
  8. MenuRepositoryImp.java
  9. DishRepository.java(Interface)
  10. DishRepositoryImp.java
  11. TableRepository.java(Interface)
  12. TableRepositoryImp.java
  13. OrderRepository.Java(Interface)
  14. OrderRepositoryImp.java
  15. Query.java(for Sql Queries)
* org.techhub.Service
  1. CustomerService.java(Interface)
  2. CustomerServiceImp.java
  3. StaffService.java(Interface)
  4. StaffServiceImp.java
  5. MenuService.java(Interface)
  6. MenuServiceImp.java
  7. DishService.java(Interface)
  8. DishServiceImp.java
  9. TableService.java(Interface)
  10. TableServiceImp.java
  11. OrderService.java(Interface)
  12. OrderServiceImp.java

2.src/main/resources(Files)

* dbconfig.properties
* log4j.properties

# Methods with ClassNames

## 1.RestaurantOrderManagementSystem

## 1. public static boolean login(staffService,CustomerService,Email,Password):- This Method is Used to validate if the Logged User isCustomer/Cashier,Admin or any other staffMember if Yes it will return true else false.

1. **private static boolean registerCustomer(CustmoreService,Scanner):-**this method returns true if the customer is registered successfully else returns false.

1. **private static boolean registerStaff(StaffService,Scanner):-**this method returns true if staff registered successfully else returns false.

1. **private static void showFoodMenu(MenuService):-**this method shows all food menus present in MenuMaster table.

1. **private static void showDishes(MenuService,DishService,Scanner):-**this method will display all dishes present in given menuType.

1. **private static OrderModel placeOrder(sc,dishService,tableNo):**This method will take dishId and quantity as input as returns OrderModel object.

## 7. private static void generateBill(List<OrderModel>OrderService):- this method will accept the list of orders ,calculates and displays the bill.

## 2.CustomerRepositoryImp

1. **public boolean isRegisteredCustomer(CustomerModel):**this method takes CustomerModel as Input ,registers the Customer and returns true if customer registered successfully else returns false.

1. **public Optional<List<CustomerModel>> getAllCustomers():-**this method retrives the list of customermaster and returns the list of customers..if not customer found returns empty optional.

1. **public String getCustomerByEmail(String):**this method searches the customer in customermaster by email and returns the email if found else returns null.

1. **public boolean deleteCustomer(String):-**this method deletes the customer by Email and returns true if deleted else returns false.
2. **Public CustomerModel getCustomer(String):-** this method takes email as input parameter and returns customemodel .

3.**StaffRepositoryImp**

1. **public boolean isRegisteredStaff(StaffModel):-** this method takes StaffModel as Input ,registers the Staff and returns true if Staff registered successfully else returns false.

1. **public Optional<List<StaffModel>> getAllStaff():-**this method retrives the list of staffmaster and returns the list of staff.if not staff found returns empty optional.

1. **public String getStaffByEmail(String):-** this method searches the staff in staffmaster by email and returns the email if found else returns null.

1. **public boolean deleteStaff(String):-**this method deletes the staff by Email and returns true if deleted else returns false.

1. **public String validateUser(String, String):**this method aceepts email and password validates if the user is valid/Registered and returns the role else returns null.

4.**MenuRepositoryImp**

1. **public boolean addMenuType(MenuModel):-**this method accepts the MenuModel as parameter ,adds menu in MenuMaster returns true if menu added else returns false.

1. **public Optional<List<MenuModel>> showMenus():-**this method accepts list of MenuModel as input and returns all Menus from Menumaster else returns Empty optional.

1. **public int getMenuIdByMenuType(String):-**this method accepts menuType as input searches for menuId and returns the menuId if Found else returns -1.

1. **public String searchMenuType(String):-**this method accepts menu Type as input and returns the menuType is found else returns null.

1. **public boolean deleteMenuType(String):-**this method accepts menu type as input and returns true if menutype is deleted else returns false.

## 4.DishRepositoryImp

1. **public boolean addBulkDishes(String,String):-**this method accepts the menuType and filename as Parameter if filename matches with menuType.csv adds bulk dishes into the given menu and returns true else returns false.

1. **public List<DishModel> showAllDishes(String):-**this method accepts menuType as input and returns list of dishes of given menuType.

1. **public String searchDishByName(String , String):-**this method accepts menutype and dishname as input and returns the dishname if found else returns null.

1. **public boolean deleteDishByName(String type, String,dishName)-**this method accepts menuType and dishName as input ,deletes the dish and returns true if found else returns false.

1. **public DishModel getDish(int):-**this method accepts dishId as input and returns dishModel if found else returns null.
2. **public boolean addDish(String,DishModel):-** this method accepts menu type and dishmodel as input and adds single dish to dishmaster and returns true if dish added else return false.

## 5.TableRepositoryImp

1. **public boolean reserveTable(int,String):-**this method accepts tablenumber and Email as input and reserves the table if available and returns true if reserved else returns false.

1. **public boolean showAvailableTables():-**this table shows all available tables and returns true if found else returns false.

## 6.OrderRepositoryImp

1. **public List<OrderModel> orderPlaced(OrderModel):-**this method accepts the OrderModel as input and returns the list of OrderModel if Order Placed successfully else returns null.

1. **public Optional<List<OrderModel>> viewAllOrders():-**this method returns list all orders from OrderModel if found else returns empty Optional.

1. **public List<OrderModel> ViewOrderByTableNo(int):-**this method accepts tableNo as input and returns list of all orders matching to the tableNo else returns null.

1. **public boolean generateBill(BillModel):-**this method accepts BillModel as input and generates bill for particular order if generated returns true else returns false.
2. **public List<OrderModel> getOrdersByOrderNo(int):-** this method accepts order no as input and returns list of pending orders from ordermaster.
3. **public boolean completePayment(double,int):-** this method accepts total bill amount and tableNo as input , updates bill status from billmaster,updates order status from ordermaster and updates status from tablemaster based on bill amount and table no returns true if updation is successfull else return false.

**DATABASE**

**Database Name**-RestaurantOrdermanagementSystem

**Tables**

1.UserMaster

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| UserId | Int | No | Primary Key | Null | Auto\_increment |
| UserName | Varchar(200) | No |  | Null |  |
| Email | Varchar(200) | No | Unique | Null |  |
| Contact | Varchar(10) | Yes |  | Null |  |
| Password | Varchar(200) | No |  | Null |  |
| Address | Varchar(200) | Yes |  | Null |  |
| Role | Varchar(200) | Yes |  | Null |  |

2.Login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| loginId | Int | No | Primary key | Null | Auto\_increment |
| Email | Varchar(200) | No | unique | Null |  |
| Password | Varchar(200) | No |  | Null |  |
| Role | Varchar(200) | No |  | Null |  |

3.MenuMaster

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| menuId | Int | No | Primary Key | Null | Auto\_increment |
| menuType | Varchar(200) | No | Unique | Null |  |

4.DishMaster

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| dishId | Int | No | Primary Key | Null | Auto\_increment |
| dishName | Varchar(200) | No | Unique | Null |  |
| Price | Int | Yes |  | Null |  |
| Category | Varchar(200) | Yes |  | Veg |  |

5.TableMaster

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| tableNo | Int | No | Primary key | Null | Check |
| custId | int | Yes | Foreign key | Null |  |
| Status | Int(10) | Yes | Default | 0 |  |

6.OrderMaster

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| orderId | Int | No | Primary key | Null | Auto\_increment |
| dishName | Varchar(200) | Yes |  | Null |  |
| Quantity | Int | Yes |  | 0 |  |
| Price | Int | No |  | Null |  |
| tableNo | int | Yes | Foreign key | Null |  |
| OrderStatus | Varchar(200) | Yes | Default | Pending |  |

7.Bill

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| billId | Int | No | Primarykey | Null | Auto\_increment |
| orderId | Int | Yes | Foreign key | Null |  |
| billDate | Date | Yes |  | Null |  |
| subtotal | Decimal(10,2) | Yes |  | Null |  |
| Gst | Decimal(10,2) | Yes |  | Null |  |
| serviceCharges | Decimal(10,2) | Yes |  | Null |  |
| totalBill | Decimal(10,2) | Yes |  | Null |  |
| BillStatus | Varchar(200) | Yes | Default | Unpaid |  |

**Join Tables**:

1.MenuDishJoin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| menuId | int | Yes | Foreign Key | Null |  |
| dishId | Int | Yes | Foreign key | Null |  |

# Procedures Used In Project-

1.To Insert Records In CustomerMaster and User Table When Customer Is Registered Delimiter ;;

create procedure addUser(cname varchar(200),mail varchar(200),cont varchar(20),pass varchar(200),addr varchar(200),role varchar(200)) begin

insert into customermaster values(‘0’,cname,mail,cont,pass,addr,role); insert into User values(‘0’,mail,pass,role); end

;;

2.To Delete the Records from CustomerMaster and User when Cuctomer deletes the Account.

Delimiter ;;

Create procedure deleteCustomer(mail varchar(200))

Begin

Delete from customerMaster Where Email=mail;

Delete from user where Email=mail;

End

;;

3.To Insert Record into StaffMaster and User Table when Staff Registers Successfully.

Delimiter ;;

Create procedure addStaff(sname varchar(200),mail varchar(200),cont varchar(20),pass varchar(200),addr varchar(200),role varchar(200))

Begin

Insert into StaffMaster Values(‘0’,sname,mail,cont,pass,addr,role);

Insert into User values(‘0’,mail,pass,role);

End

;;

4.To Delete the Record from StaffMaster and User when Staff Deletes the account.

Delimiter ;;

Create procedure deleteStaff(mail varchar(200))

Begin

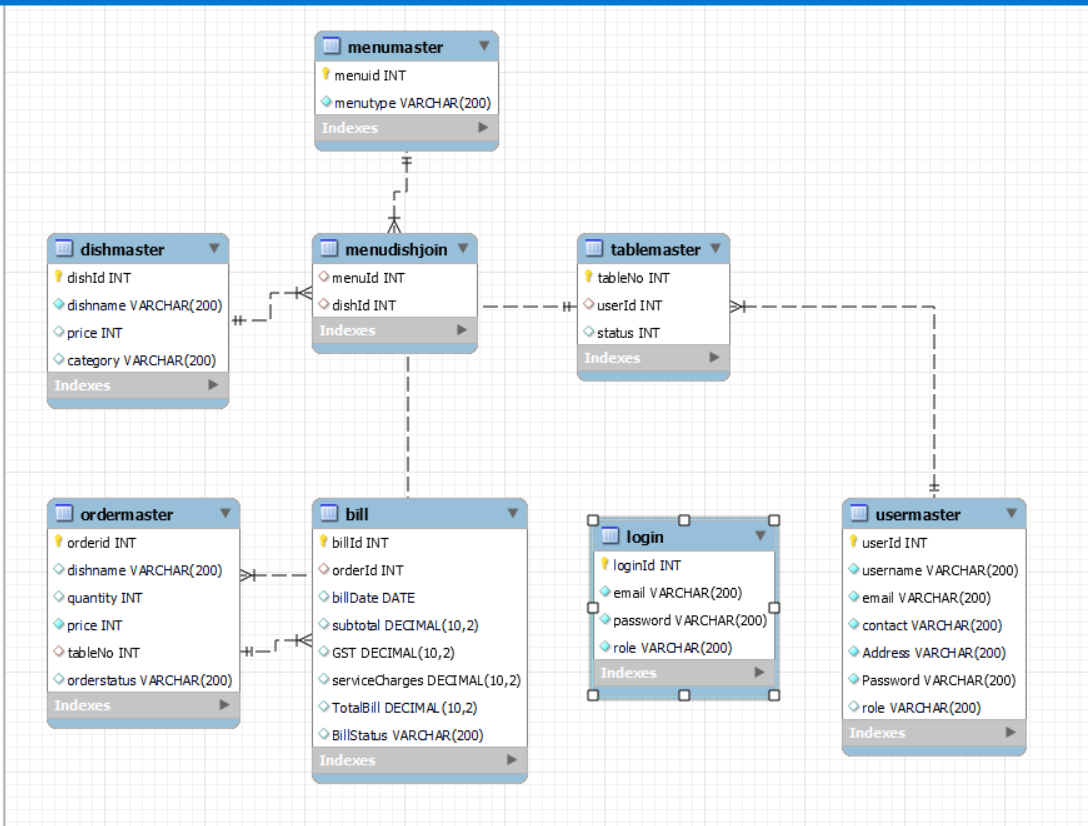
Delete from staffMaster Where Email=mail;

Delete from User where Email=mail;

End

;;

**ER DIAGRAM :**

****