**Restaurant Management System**

Contents

[Idea 1](#_Toc89715479)

[Goal of application 1](#_Toc89715480)

[The Main Characters 2](#_Toc89715481)

[UML Diagram 2](#_Toc89715506)

[Code: 3](#_Toc89715507)

[**Conclusion** 33](#_Toc89715515)

# Idea

Restaurant management is the profession of managing a restaurant. It includes the principles of OOP and using Java function of planning, organizing, staffing, directing, developing an attitude in food and beverage control systems and to efficiently and effectively plan menus at profitable prices, taking into consideration constraints and others.

The advantage of the system is that there is no need to hire a person for the same only a  
system is required to execute it. The Customer can work on the program and select the items  
which he/she wants to order.

# Goal of application

* It increases operational efficiency.
* It is designed to help you cost your recipes and track inventory saving your Money   
  and Time and maximizing profit.
* It helps the restaurant manager to manage the restaurant more effectively and   
  efficiently by computerizing Meal Ordering, Cart, and Restaurant Management.   
  Accounting.
* It increases the security.
* It avoids paper work.
* It is Simple to learn and easy to use.
* It is portable Application.

# The Main Characters

Restaurant Management System will the following features:

* Restaurant management-side features:
  + Map of orders in the restaurant
  + Assign place order
  + Edit in price
  + Read New order
  + Clean history
* Receipt-side features:
  + Assign items of order into price.
  + Time of order
  + Treat With Restaurant management
* Order-side features:
  + Calculate Available quantity of item.
  + Showing information of order.
* Menu-side features:
  + Showing All item Available in item.
  + Edit items in Menu.
  + Choice item.
* Item-side features:
  + Giving information of Item.
* Cart-side features:
  + Add items that taken in Card.
  + Conform order.
  + Calculate Total Price of order.
  + Payment

# UML Diagram

In the Figure UML Represent Restaurant Management System

Diagram

Description automatically generated

# Code:

# Project (Main) class:

package restaurant;

import java.text.DecimalFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.Scanner;

public class RestaurantManagement {

ArrayList<Item> fmenu = new ArrayList<>();

ArrayList<Item> Dmenu = new ArrayList<>();

ArrayList<Item> smenu = new ArrayList<>();

ArrayList<Receipt> allreceipt = new ArrayList<>();

Scanner input = new Scanner(System.in);

int count = 101;

public static void main(String[] args) {

clearScreen();

RestaurantManagement rest = new RestaurantManagement();

rest.start();

}

public void start() {

initData();

person logger = new cahier();

String staff = logger.login();

String option = "";

option = menu();

while (!option.trim().equalsIgnoreCase("Q")) {

if (option.trim().equalsIgnoreCase("1")) {

clearScreen();

PlaceOrder(staff);

} else if (option.trim().equalsIgnoreCase("2")) {

clearScreen();

EditPrice();

} else {

System.out.println("Invalid Input!");

}

option = menu();

}

System.out.println("");

System.out.println("Thanks for Using EL Khabar Management System!");

System.out.println("");

}

public void PlaceOrder(String staff) {

String choose1 = "";

String OrderID;

int success = 0;

while (!choose1.trim().equalsIgnoreCase("M")) {

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Make New Order \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Select from Options or Enter ‘M’ for Main Menu");

System.out.println("1. Start Order");

System.out.println("M. Go to main Menu");

choose1 = input.next();

if (choose1.trim().equalsIgnoreCase("1")) {

clearScreen();

OrderID = "SO" + count;

success = ordering(OrderID, choose1, staff);

}

if (choose1.trim().equalsIgnoreCase("M")) {

break;

}

}

}

public void EditPrice() {

String choose3 = "";

clearScreen();

while (!choose3.trim().equalsIgnoreCase("M")) {

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Edit Price \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Select from Options or Enter ‘M’ for Main Menu");

System.out.println("1. Food");

System.out.println("2. Drinks");

System.out.println("3. Side");

System.out.println("M. Go to main Menu");

Menu Do = new Menu(Dmenu);

Menu fo = new Menu(fmenu);

Menu so = new Menu(smenu);

choose3 = input.next();

String type = " ";

if (choose3.trim().equalsIgnoreCase("1")) {

type = "food";

fo.displayItem();

fo.editItem();

System.out.println("Price Edited Succesfully!");

} else if (choose3.trim().equalsIgnoreCase("2")) {

type = "drink";

Do.displayItem();

Do.editItem();

System.out.println("Price Edited Succesfully!");

} else if (choose3.trim().equalsIgnoreCase("3")) {

type = "side";

so.displayItem();

so.editItem();

System.out.println("Price Edited Succesfully!");

} else if (choose3.trim().equalsIgnoreCase("M")) {

break;

}

}

}

public int ordering(String order, String type, String staff) {

String Ti = "";

int s = 0;

Menu ordermenu = new Menu();

ArrayList<Order> forder = new ArrayList<>();

ArrayList<Order> Dorder = new ArrayList<>();

ArrayList<Order> sorder = new ArrayList<>();

while (!Ti.trim().equalsIgnoreCase("B")) {

String title;

if ("1".equals(type)) {

title = "Start Order";

} else {

title = "";

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* " + title + " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

ordermenu.displayMenu();

Menu Do = new Menu(Dmenu);

Menu fo = new Menu(fmenu);

Menu so = new Menu(smenu);

Ti = input.next();

if (Ti.trim().equalsIgnoreCase("1")) {

fo.displayItem();

forder.addAll(fo.chooseItem());

System.out.print("Enter Anything to continue...");

input.next();

clearScreen();

} else if (Ti.trim().equalsIgnoreCase("2")) {

Do.displayItem();

Dorder.addAll(Do.chooseItem());

System.out.print("Enter Anything to continue...");

input.next();

clearScreen();

} else if (Ti.trim().equalsIgnoreCase("3")) {

so.displayItem();

sorder.addAll(so.chooseItem());

System.out.print("Enter Anything to continue...");

input.next();

clearScreen();

}

if (Ti.trim().equalsIgnoreCase("4")) {

int check = 1;

if (forder.size() == 0) {

if (Dorder.size() == 0) {

if (sorder.size() == 0) {

check = 0;

}

}

}

if (check != 0) {

Cart dn = new Cart(forder, Dorder, sorder);

Receipt rc = new Receipt();

rc = (Receipt) (dn.displayCart(order, staff));

System.out.println("\n-------------------------------------------------");

System.out.println("Confirm Order & Pay ? (y/n)");

String pay = input.next();

if ("y".equalsIgnoreCase(pay)) {

allreceipt.add(rc);

System.out.println("Order Paid Successfully");

}

} else {

System.out.println("You havent order anything!");

}

System.out.print("Enter Anything to continue...");

input.next();

Ti = "B";

clearScreen();

}

if (Ti.trim().equalsIgnoreCase("B")) {

break;

}

}

return (s);

}

public static void clearScreen() {

try {

new ProcessBuilder("cmd", "/c", "cls").inheritIO().start().waitFor();

} catch (Exception E) {

System.out.println(E);

}

}

public String menu() {

clearScreen();

System.out.println("Select any Option you want to perform. Enter ‘Q’ to quit the program ");

System.out.println("1. Make Order");

//System.out.println("2. Order Status");

System.out.println("2. Edit Price");

System.out.println("");

return input.next();

}

public void initData() {

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Welcome To Restaurant El khabar \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

fmenu.clear();

Dmenu.clear();

smenu.clear();

Item fm = new Item("1", "buratta pizza", 56.0);

fmenu.add(fm);

fm = new Item("2", "Pink pasta", 37.0);

fmenu.add(fm);

fm = new Item("3", "Spaghetti", 40.0);

fmenu.add(fm);

fm = new Item("4", "Rosemary salmon", 87.0);

fmenu.add(fm);

Item dm = new Item("1", "Cola", 5.0);

Dmenu.add(dm);

dm = new Item("2", "7up", 5.0);

Dmenu.add(dm);

dm = new Item("3", "Orange juice", 15.0);

Dmenu.add(dm);

dm = new Item("4", "Mojito", 25.0);

Dmenu.add(dm);

Item sm = new Item("1", "Dynamite shrimp", 39.0);

smenu.add(sm);

sm = new Item("2", "Mac & cheese balls", 45.0);

smenu.add(sm);

sm = new Item("3", "Tiramisu", 42.0);

smenu.add(sm);

sm = new Item("4", "Molten Chocolate ", 19.0);

smenu.add(sm);

}

}

# Receipt class:

package project;

import java.text.DecimalFormat;//\*\*\*\*

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.ArrayList;

import java.util.Scanner;

public class Receipt {

String orderId;

String staff;

String orderTime;

String orderDate;

ArrayList<String> name = new ArrayList<>();

ArrayList<Integer> qty = new ArrayList<>();

ArrayList<Double> price = new ArrayList<>();

Double Total;

Scanner input = new Scanner(System.in);

public Receipt() {

}

public Receipt(String Oid, String staff, String orderTime, String orderDate, ArrayList<String> name, ArrayList<Integer> qty, ArrayList<Double> price, Double Total) {

this.orderId = Oid;

this.staff = staff;

this.orderTime = orderTime;

this.orderDate = orderDate;

this.name = name;

this.qty = qty;

this.price = price;

this.Total = Total;

}

public String getOrderId() {

return orderId;

}

public String getStaff() {

return staff;

}

public String getOrderTime() {

return orderTime;

}

public String getOrderDate() {

return orderDate;

}

public ArrayList<String> getName() {

return name;

}

public ArrayList<Integer> getQuantity() {

return qty;

}

public ArrayList<Double> getPrice() {

return price;

}

public Double getTotal() {

return Total;

}

public void Order() {//\*\*\*\*

DecimalFormat df = new DecimalFormat("#.##");//\*\*\*\*

System.out.println(" ");

System.out.println("Order ID: " + orderId + " " + orderTime);//\*\*\*\*

System.out.println("Staff Name: " + staff + " " + orderDate);

int count = 1;

System.out.printf("%2s %20s %5s %10s", "No", "NAME", "QTY", "PRICE");//\*\*\*\*

System.out.println();

for (int i = 0; i < name.size(); i++) {

System.out.println("-------------------------------------------------");

System.out.format("%2d %20s %5s %10s", count, (name.get(i)), (qty.get(i)), df.format((price.get(i)) \* (qty.get(i))));//\*\*\*\*

System.out.println();

count++;

}

System.out.println("\n-------------------------------------------------");

System.out.println("\nTOTAL " + df.format(Total));

System.out.println("\n-------------------------------------------------");

System.out.println("Order Status: Delivered");

System.out.println(" ");

}

}

# Person class:

package project;

import java.util.Scanner;

abstract class person {

String name;

int age;

String Address;

abstract String login();

}

class cashier extends person{

@Override

public String login() {

Scanner input = new Scanner(System.in);

String[] worker = new String[3];

worker[0] = "admin";//\*\*\*\*

String[] pwd = new String[3];

pwd[0] = "123";//\*\*\*\*

String uname = "", pass;

int temp = 0;

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Login \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("");

while (temp == 0) {

int invalidname = 0;

int invalidpass = 0;

System.out.println("Enter Username: ");

uname = input.next();

System.out.println("Enter Password: ");

pass = input.next();

for (int i = 0; i < worker.length; i++) {

if (uname.equalsIgnoreCase(worker[i])) {

if (pass.equals(pwd[i])) {

System.out.println("Login Succesful!");

System.out.println("Welcome " + worker[i]);

uname = worker[i];

temp = 1;

invalidname = 0;

System.out.println("Enter anything to continue...");

input.next();

} else {

invalidpass = 1;

}

} else {

invalidname++;

}

}

if (invalidname == worker.length) {

if (invalidpass == 0) {

System.out.println("Wrong Username!");

}

}

if (invalidpass == 1) {

System.out.println("Password Mismatch!");

}

}

return (uname);

}

}

# Order class:

package project;

package restaurant;

import java.util.ArrayList;

public class Order {

private ArrayList<String> OName = new ArrayList<>();

protected ArrayList<Integer> quantity = new ArrayList<>();

private ArrayList<Double> Oprice = new ArrayList<>();

Order() {

}

public Order (ArrayList<String> OName, ArrayList<Integer> quantity, ArrayList<Double> Oprice) {

this.OName = OName;

this.quantity = quantity;

this.Oprice = Oprice;

}

public ArrayList<String> getOName() {

return OName;

}

public ArrayList<Integer> getQuantity() {

return quantity;

}

public ArrayList<Double> getPrice() {

return Oprice;

}

}

# Menu class:

package project;

import java.util.ArrayList;

import java.util.Scanner;

public class Menu {

private ArrayList<Item> item = new ArrayList<>();

Scanner input = new Scanner(System.in);//\*\*\*

Menu() {

}

public Menu(ArrayList<Item> item) {

this.item = item;

}

public ArrayList<Item> getItem()

{

return item;

}

public ArrayList<Item> editItem() {

String chooseP = "y";

int index;

while (!"n".equalsIgnoreCase(chooseP)) { //\*\*\*

int found=0;

double nprice;

System.out.println("Enter the item ID: ");

String cusId = input.next();

for (int i = 0; i < item.size(); i++) {

if (cusId.equalsIgnoreCase(item.get(i).getId())) {

index = i;

System.out.println(" ");

System.out.println("You have selected " + item.get(index).getName());

System.out.println("Current Price: " + item.get(index).getPrice() );

System.out.println("Enter The New Price For This Item: ");

nprice = input.nextDouble();

item.get(index).price = nprice;

}

else

{

found++;

}

}

if (found == item.size())

{

System.out.println("Item didn't exist!");

}

System.out.println("Do you want to edit price of another item from here ? (y/n)");

chooseP = input.next();

}

return (item);

}

public ArrayList<Order> chooseItem() {

ArrayList<Order> ordertemp = new ArrayList<>();

ArrayList<String> Cname = new ArrayList<>();

ArrayList<Integer> Cqty = new ArrayList<>();

ArrayList<Double> Cprice = new ArrayList<>();

String chooseP = "y";

int index;

while (!"n".equalsIgnoreCase(chooseP)) {

int found=0;

System.out.println("Enter the item ID: ");

String cusId = input.next();

int qty;

for (int i = 0; i < item.size(); i++) {

if (cusId.equalsIgnoreCase(item.get(i).getId())) {

index = i;

System.out.println("How many " + item.get(index).getName() + " you want ? :");

qty = input.nextInt();

Cname.add(item.get(index).getName());

Cprice.add(item.get(index).getPrice());

Cqty.add(qty);

}

else

{

found++;

}

}

if (found == item.size())

{

System.out.println("Item didn't exist!");

}

System.out.println("Do you want to add another item from here ? (y/n)");

chooseP = input.next();

}

Order dummy = new Order(Cname, Cqty, Cprice);

ordertemp.add(dummy);

return (ordertemp);

}

public void displayMenu() {

System.out.println("Select from Options or Enter ‘B’ to go back.");

System.out.println("1. Food");

System.out.println("2. Drinks");

System.out.println("3. Side");

System.out.println("4. Cart");

System.out.println("B. Go Back");

}

public void displayItem() {

System.out.println("");

int count = 1;

System.out.printf("%2s %10s %15s %10s", "No", "ID", "NAME", "PRICE");

System.out.println();

for (int i = 0; i < item.size(); i++) {

System.out.format("%2d %10s %15s %10s", count, item.get(i).getId(), item.get(i).getName(), item.get(i).getPrice()); //formatting

System.out.println();

count++;

}

}

}

# Item class:

package project;

public class Item {

private String id;

private String name;

protected double price;

Item() {

}

public Item(String id, String name, double price) {

this.id = id;

this.name = name;

this.price = price;

}

public String getId() {

return id;

}

public String getName() {

return name;

}

public double getPrice() {

return price;

}

# Cart class:

package project;

package restaurant;

import java.text.DecimalFormat;

import java.util.ArrayList;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Scanner;

public class Cart {

private ArrayList<String> name = new ArrayList<>();

private ArrayList<Integer> qty = new ArrayList<>();

private ArrayList<Double> price = new ArrayList<>();

Scanner input = new Scanner(System.in);

public Cart(ArrayList<Order> fcart, ArrayList<Order> Dcart, ArrayList<Order> scart) {

ArrayList<String> iname = new ArrayList<>();

ArrayList<Integer> iqty = new ArrayList<>();

ArrayList<Double> iprice = new ArrayList<>();

for (int i = 0; i < fcart.size(); i++) {

iname.addAll(fcart.get(i).getOName());

iqty.addAll(fcart.get(i).getQuantity());

iprice.addAll(fcart.get(i).getPrice());

}

for (int i = 0; i < Dcart.size(); i++) {

iname.addAll(Dcart.get(i).getOName());

iqty.addAll(Dcart.get(i).getQuantity());

iprice.addAll(Dcart.get(i).getPrice());

}

for (int i = 0; i < scart.size(); i++) {

iname.addAll(scart.get(i).getOName());

iqty.addAll(scart.get(i).getQuantity());

iprice.addAll(scart.get(i).getPrice());

}

this.name = iname;

this.qty = iqty;

this.price = iprice;

}

public Cart() {

}

public ArrayList<String> getName() {

return name;

}

public ArrayList<Integer> getQuantity() {

return qty;

}

public ArrayList<Double> getPrice() {

return price;

}

double getTotal() {

double subtotal = 0.0;

for (int i = 0; i < name.size(); i++) {

double amt;

amt = price.get(i) \* qty.get(i);

subtotal = subtotal + amt;

}

return (subtotal);

}

public String getTime() {

String jam;

SimpleDateFormat formatter = new SimpleDateFormat("HH:mm:ss");

Date datey = new Date();

jam = formatter.format(datey);

return (jam);

}

public String getDate() {

String hari;

SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");

Date today = new Date();

hari = formatter.format(today);

return (hari);

}

public Object displayCart(String id, String staff) {

DecimalFormat df = new DecimalFormat("#.##");

String orderTime = getTime();

String orderDate = getDate();

System.out.println("Order ID: " + id + " " + orderTime);

System.out.println("Staff Name: " + staff + " " + orderDate);

int count = 1;

System.out.printf("%2s %20s %5s %10s", "No", "NAME", "QTY", "PRICE");

System.out.println();

for (int i = 0; i < name.size(); i++) {

System.out.println("-------------------------------------------------");

System.out.format("%2d %20s %5s %10s", count, name.get(i), qty.get(i), df.format((price.get(i) \* qty.get(i))));

System.out.println();

count++;

}

double stotal = getTotal();

double taxtotal = (stotal \* 0.15);

double ftotal = stotal + taxtotal;

System.out.println("\n-------------------------------------------------");

System.out.println("\nBefore Tax TOTAL " + df.format(stotal));

System.out.println("TAX(%15) " + df.format(taxtotal));

System.out.println("Total " + df.format(ftotal));

Receipt dummy = new Receipt(id, staff, orderTime, orderDate, name, qty, price, ftotal);

return (dummy);

}

}

# **Conclusion**

The Restaurant Management System is used to grow the restaurant's business in order to fulfil orders.

It makes it easier for us to deal with customers and reduces the amount of work we have to do.as well as less time spent in use For the customer, there are no delays. There is a main screen that contains a list of foods from which the customer can select and show

He will be informed of the total cost of the order and will be able to connect to the main program.