[Date]

***By: Maryam Kareem (42876)***

*Restaurant Management System*

[Report]

**Abstract**

**Bachelors in Software Engineering**

**Iqra University**

**Project Report**

**Project Title**:     Restaurant Management System

**Author:**Maryam Kareem

**Abstract:**

  This Hamburger Restaurant Management is based on javaFx and IDE. This Platform is made for the online order purpose in which you must login first, add items in cart which you like to order and the billing page, and then your order is placed.

Running a restaurant is a big challenge now a days, so why not make the day-to-day processes easier by having a system that will help ease the workload for you? There are so many day-to-day processes that restaurants have to taking online orders. These can range from managing HR, monitoring employee attendance to preparing for payroll and to keep record of transactions and database. In current marketplace, there is a great value for food, restaurants and its management. There is day by day increasing number of restaurants and food places that are emerging today and also dealing online. It can be considered as a rapid growth in the field of business and food restaurants and its management system. The management system applied for every restaurant is different from the other one. Some restaurants may be bigger while the other may be smaller but every restaurant or hotel requires a management system and this is termed as Restaurant Management System. RMS that is, Restaurant Management Systems must have better serve its customers and aid employees with food and beverage transactions and controls online. Restaurant management System is database program that keeps record of all transaction carried out in the restaurant on daily bases online orders. restaurant management software are a new techniques to make online customers and to serve them at home. Every restaurant out there, whether small, medium-sized, or large, will benefit greatly by switching from manual restaurant management processes to automated or software-based online ordering system ones.

**Report Approved by Project Advisor**:                                                                                   Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Contents**

[1. REQUIREMENT ELICIATION: 4](#_Toc61569250)

[**Initial investigation:** 4](#_Toc61569251)

[**Study & Requirement Gathering** 4](#_Toc61569252)

[**INTRODUCTION** 5](#_Toc61569253)

[**PROCESS MODEL TO BE USED WITH THE REASON:** 5](#_Toc61569254)

[**Phases to be covered in Process model for Restaurant Management System:** 6](#_Toc61569255)

[**Over view of the system** 7](#_Toc61569256)

[**RESTAURANT MANAGEMENT SYSTEM:** 7](#_Toc61569257)

[**PROJECT Overview:** 7](#_Toc61569258)

[**ADMIN LOGIN PAGE:** 8](#_Toc61569259)

[**MAIN PAGE:** 9](#_Toc61569260)

[**BILL VIEW PAGE:** 10](#_Toc61569261)

[**Implementation Phase:** 11](#_Toc61569262)

[**Designing:** 11](#_Toc61569263)

[**Code:** 11](#_Toc61569264)

[**UML DIAGRAM OF RESTAURANT MANAGEMENT:** 27](#_Toc61569265)

[**SYSTEM REQUIREMENT SPECIFICATION:** 28](#_Toc61569266)

[**USECASE DIAGRAM:** 32](#_Toc61569267)

[**CONCLUSION:** 32](#_Toc61569268)

1. **REQUIREMENT ELICIATION:**

**Initial investigation:**

There are so many day-to-day processes that restaurants have to deal online order with. These can range from scheduling in employees, managing HR, monitoring employee for online ordering system attendance to preparing for payroll and automatically you can save the data easily on website.

Restaurant Management is the profession of managing a restaurant. Associate, bachelor, and graduate degree programs are offered in restaurant management by community colleges, junior colleges, and some universities in the United States. Management generally involves three major responsibilities: Administration, Front-of-the-House Management and Back-of-the-House Management. This system is developed to automate day to day activity of a restaurant. Restaurant is a kind of business that serves people all over world with readymade food. This system is developed to provide service facility to restaurant and also to the customer. This restaurant management system can be used by employees in a restaurant to handle the clients, their orders and can help them easily find free tables or place orders.

**Study & Requirement Gathering**:

In our current situation we face many problems while we order food (from customers point of view).

Doubt on waiters (from manager's/ management's point of view).

To revoke these types of problem we need this software. It is very logical to handle the above problems. This system will not be complex, very easy to understand (user friendly interface). In current marketplace, upper and middle-class restaurants will have much demand of this system

# **INTRODUCTION**

**Restaurant Management system**:

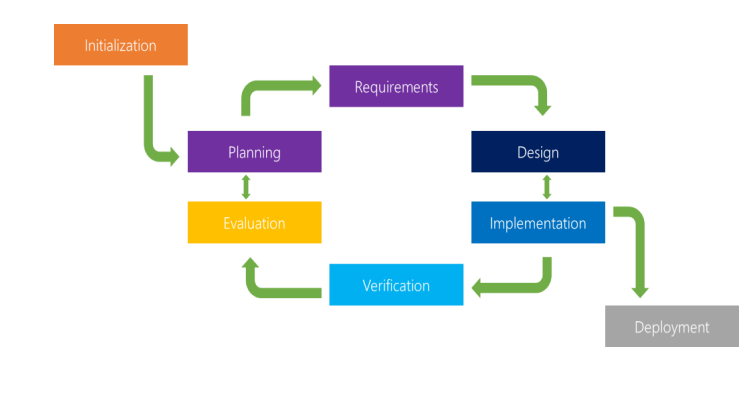
This restaurant management system can be used by employees in a restaurant to handle the clients, their orders and can help them easily find free tables. The restaurant online menu is organized by categorized . Each menu item has a name, price written beside it.

Restaurant management can vary across multiple management styles, however, there is always one common denominator when it comes to setting goals: maximizing a restaurant’s profitability. In order to maximize a restaurant’s profitability, one has to always examine and understand a restaurant’s operational costs and how these relate to a restaurant’s productivity and efficiency in delivering quality service to its customers. Management takes a very important role in controlling and manipulating the balance of costs and profitability. An effective manager must always concern himself/herself with restaurant issues that pertain to inventory/stocking, pricing, order-taking, and much more. Oftentimes, a restaurant’s profitability either rises or falls depending on how well it is being managed.

Managing a restaurant online using a well-developed software minimizes the liabilities of mismanagement and productivity loopholes. The incorporation of a Restaurant Management Software in the managing of various business processes entails that your restaurant is competitive, innovative, well-managed, and up-to-date with the latest management and business trends.

## **PROCESS MODEL TO BE USED WITH THE REASON:**

The process model we have used here is iterative model. The iterative model is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. When discussing the iterative method, the concept of incremental development will also often be used liberally and interchangeably, which describes the incremental alterations made during the design and implementation of each new iteration.



### **Phases to be covered in Process model for Restaurant Management System:**

**• Planning & Requirements:**

As with most any development project, the first step is go through an initial planning stage to map out the specification documents, establish software or hardware requirements, and generally prepare for the upcoming stages of the cycle

**. • Analysis & Design:**

Once planning is complete, an analysis is performed to nail down the appropriate business logic, database models, and the like that will be required at this stage in the project. The design stage also occurs here, establishing any technical requirements (languages, data layers, services, etc) that will be utilized in order to meet the needs of the analysis stage.

**• Implementation:**

With the planning and analysis out of the way, the actual implementation and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.

**• Testing:**

Once this current build iteration has been coded and implemented, the next step is to go through a series of testing procedures to identify and locate any potential bugs or issues that have cropped up.

**• Evaluation:**

Once all prior stages have been completed, it is time for a thorough evaluation of development up to this stage. This allows the entire team, as well as clients or other outside parties, to examine where the project is at, where it needs to be, what can or should change, and so on.

# **Over view of the system**

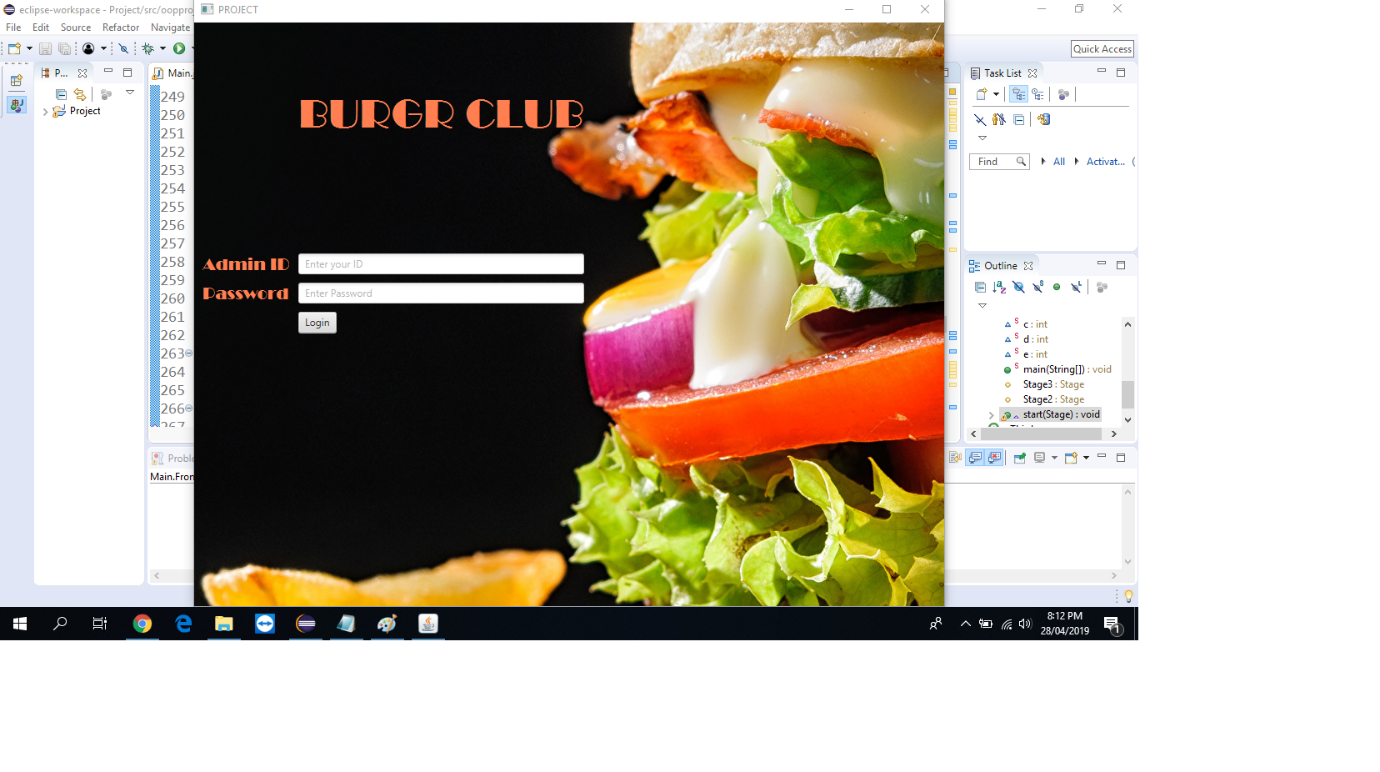
## **RESTAURANT MANAGEMENT SYSTEM:**

## **PROJECT Overview:**

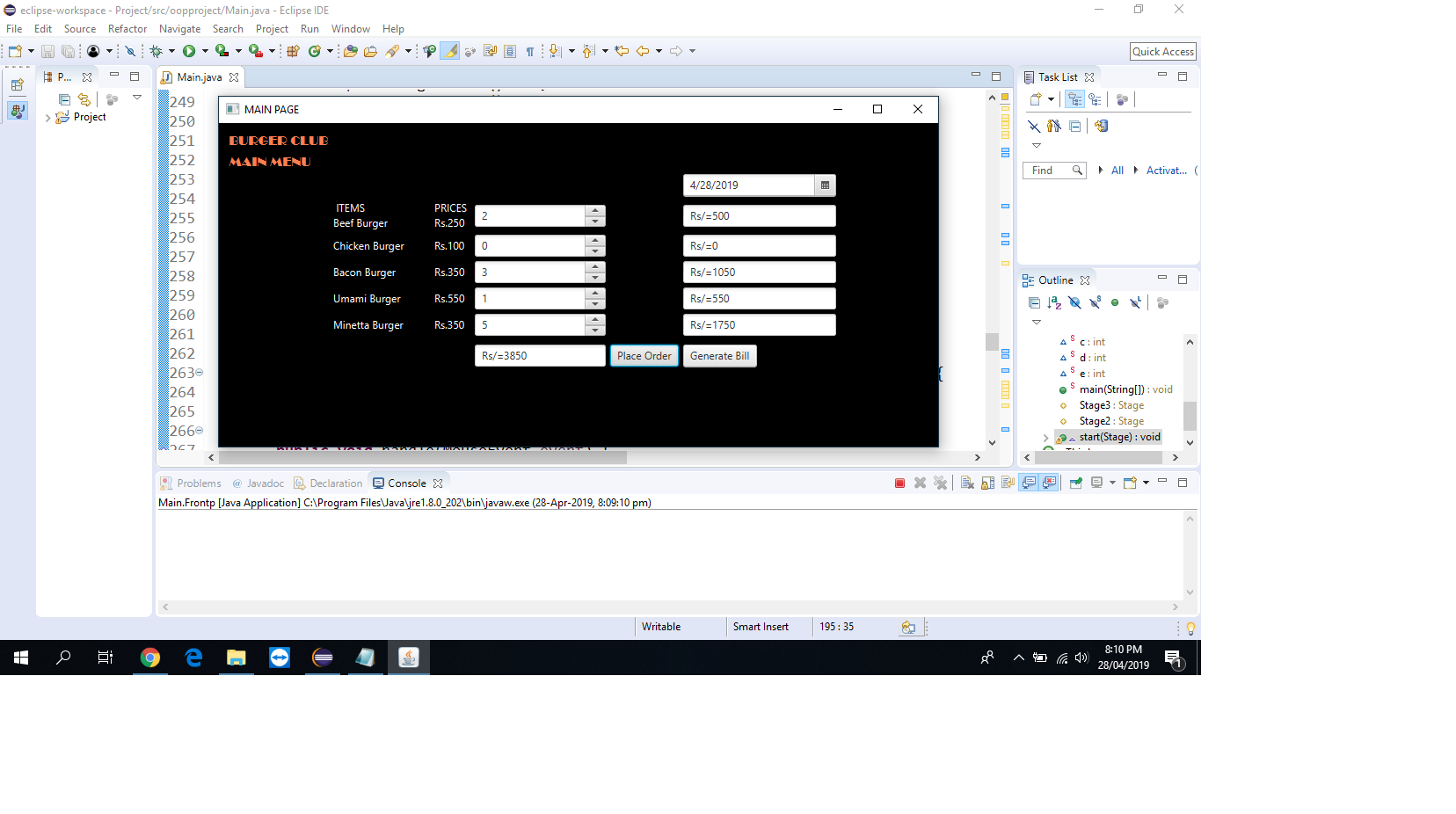
Project is based on javaFx and IDE which is used is ESCLIPSE. It consists of two forms which are:

* Admin Login Page
* Main Page
* Bill Page

### **ADMIN LOGIN PAGE:**



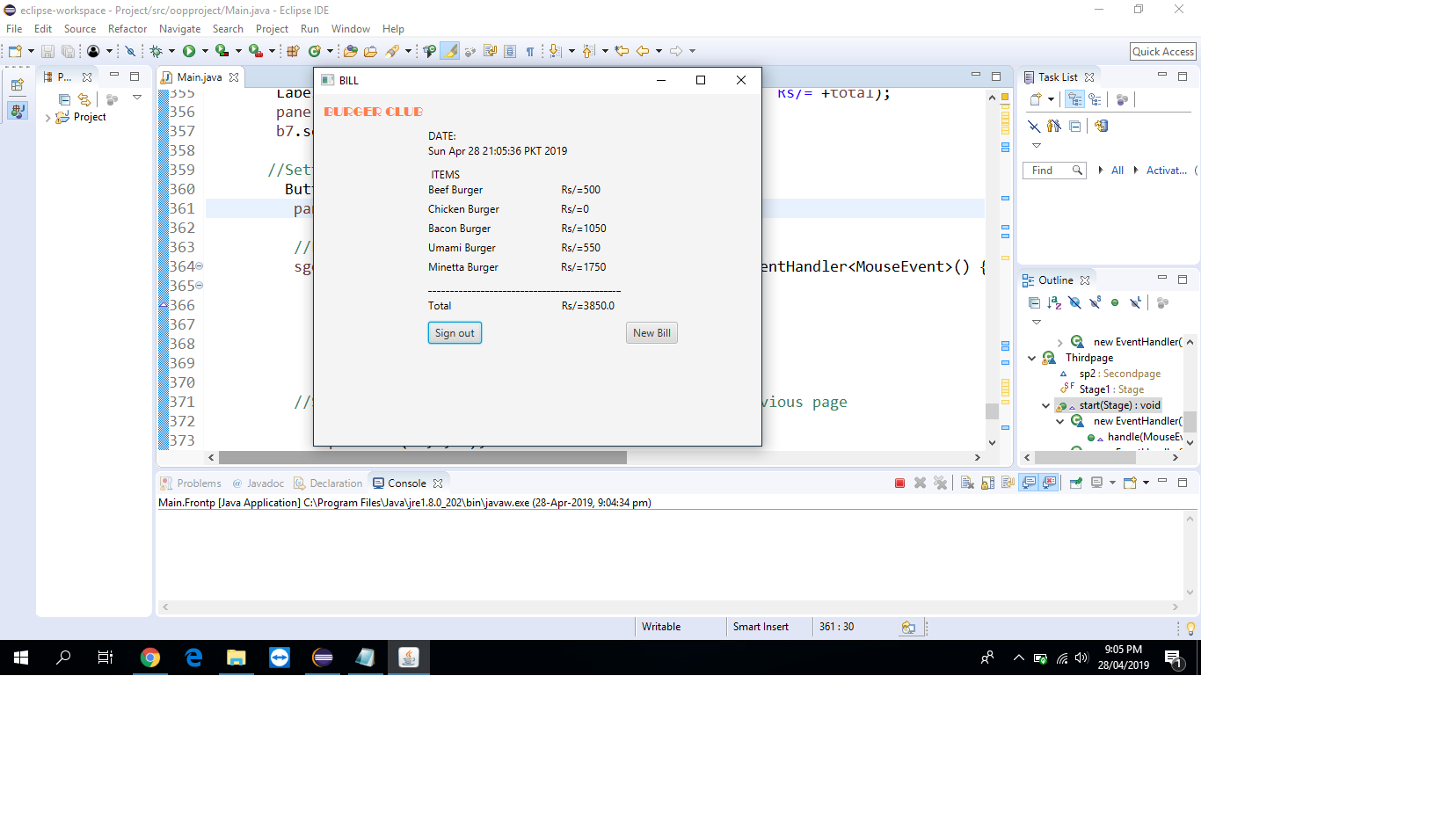
* Here user input ID and Password and open the main page for use.



### **MAIN PAGE:**

* To show the date the user click on date and the present date will appear.
* Here menu is mentioned which is random, it can be changed from backend.
* Whatever the customer ordered user just have to put the quantity of that item, then click on place order button, by clicking it the subtotal and total price will be automatically calculated and shown in their text fields.
* Generate button will generate a bill of order.

### **BILL VIEW PAGE:**

****

* Here bill of order generated.
* New bill button will open the main page again for more orders.
* Sign out button will perform the signing out function it closes the program.

# **Implementation Phase:**

# **Designing:**

In this project we did all the designing phase according to user’s need and also we did checking starting from the login page and till the last bill page

# **Code:**

package project;

import java.util.Date;

import java.util.Scanner;

import javax.swing.event.ChangeEvent;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.event.Event;

import javafx.event.EventHandler;

import javafx.geometry.Pos;

import javafx.scene.Node;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.DatePicker;

import javafx.scene.control.Label;

import javafx.scene.control.Labeled;

import javafx.scene.control.PasswordField;

import javafx.scene.control.Spinner;

import javafx.scene.control.TextField;

import javafx.scene.effect.DropShadow;

import javafx.scene.effect.InnerShadow;

import javafx.scene.effect.Reflection;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.Background;

import javafx.scene.layout.BackgroundFill;

import javafx.scene.layout.BackgroundImage;

import javafx.scene.layout.BackgroundPosition;

import javafx.scene.layout.BackgroundRepeat;

import javafx.scene.layout.BackgroundSize;

import javafx.scene.layout.GridPane;

import javafx.scene.paint.Color;

import javafx.scene.text.Font;

import javafx.scene.text.Text;

import javafx.stage.Stage;

public class Main {

**// Class of first page**

public static class Frontp extends Application {

//Declare id and password instance

String Admin = "Maryam";

String Pass = "123456";

String checkAdmin, checkPass;

//Object of second page in page for calling secondpage

Secondpage sp=new Secondpage();

public static void main(String[] args) {

// TODO Auto-generated method stub

launch(args);

}

@Override

public void start(Stage Stage1) throws Exception {

// TODO Auto-generated method stub

**//Setting of firstpage + styling**

GridPane pane=new GridPane();

Scene scene = new Scene(pane,900,700);

Stage1.setScene(scene);

Stage1.show();

Stage1.setTitle("PROJECT");

Label heading1= new Label("BURGR CLUB");

heading1.setFont(Font.font ("broadway", 50));

heading1.setAlignment(Pos.TOP\_LEFT);

heading1.setTextFill(Color.CORAL);

pane.setHgap(10);

pane.setVgap(10);

pane.add(heading1, 2,8 );

//Background settings

Image image=new Image("file:C:\\Users\\Maryam\\Desktop\\Resturaunt management system\\Hamburger\_Buns\_French\_fries\_Chili\_pepper\_558301\_3840x2400.jpg");

BackgroundImage backgroundImage = new BackgroundImage(image, BackgroundRepeat.NO\_REPEAT, BackgroundRepeat.NO\_REPEAT, BackgroundPosition.CENTER, null);

Background background = new Background(backgroundImage);

pane.setBackground(background);

//Label setting

Label adminid = new Label("Admin ID");

adminid.setFont(Font.font ("broadway", 20));

adminid.setAlignment(Pos.CENTER\_LEFT);

adminid.setTextFill(Color.CORAL);

pane.add(adminid, 1, 22);

Label password = new Label("Password");

password.setFont(Font.font ("broadway", 20));

password.setAlignment(Pos.CENTER\_LEFT);

password.setTextFill(Color.CORAL);

pane.add(password, 1, 23);

//Textfield settings

final TextField txt = new TextField();

txt.setPromptText("Enter your ID");

txt.setAlignment(Pos.CENTER\_LEFT);

pane.add(txt, 2, 22);

final PasswordField pas = new PasswordField();

pas.setPromptText("Enter Password");

pas.setAlignment(Pos.CENTER\_LEFT);

pane.add(pas, 2, 23);

//login Button settings

Button btn=new Button("Login");

final Label lblMsg = new Label();

btn.setAlignment(Pos.CENTER);

pane.add(btn, 2, 24);

pane.add(lblMsg, 2, 25);

//Function of login button

btn.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent arg0) {

// TODO Auto-generated method stub

checkAdmin = txt.getText().toString();

checkPass = pas.getText().toString();

if(checkAdmin.equalsIgnoreCase(Admin)&& checkPass.equals(Pass)){

lblMsg.setText("Congratulations!");

lblMsg.setTextFill(Color.GREEN);

try

{

sp.start(Stage1);

} catch (Exception e)

{

e.printStackTrace();

}

}

else{

lblMsg.setText("Incorrect user or password");

lblMsg.setTextFill(Color.RED);

}

}

});

}

}

}

//**Class of second page**

class Secondpage extends Application

{

// declaration of variables as static for subtotal

static int a=0,b=0,c=0,d=0,e=0;

public static void main(String[] args) {

// TODO Auto-generated method stub

launch(args);

}

//for calling page 2 and page3

protected Stage Stage3;

protected Stage Stage2;

@Override

public void start(Stage Stage2) throws Exception {

// TODO Auto-generated method stub

//Setting of secondpage + styling

GridPane pane=new GridPane();

Scene scene = new Scene(pane,800,600);

Stage2.setScene(scene);

Stage2.setTitle("MAIN PAGE");

Label heading1= new Label(" BURGER CLUB");

heading1.setFont(Font.font ("broadway", 15));

heading1.setTextFill(Color.CORAL);

pane.setHgap(5);

pane.setVgap(5);

pane.add(heading1, 0, 2);

//Heading settings

Label heading2= new Label(" MAIN MENU");

heading2.setFont(Font.font ("broadway", 15));

heading2.setTextFill(Color.CORAL);

pane.add(heading2, 0, 3);

**//Background settings**

BackgroundFill background = new BackgroundFill(Color.BLACK,null,null);

Background back = new Background(background);

pane.setBackground(back);

//Date settings//Setting of firstpage + styling

Date date=new Date();

DatePicker dp=new DatePicker();

dp.setPromptText("Date");

pane.add(dp, 4,4);

//Settings of LAbels of food item along with their PRICES

Label b1=new Label(" ITEMS\t \t PRICES \nBeef Burger\t Rs.250");

pane.add(b1, 1,5);

b1.setTextFill(Color.WHITE);

Label b2=new Label("Chicken Burger\t Rs.100");

pane.add(b2, 1, 6);

b2.setTextFill(Color.WHITE);

Label b3=new Label("Bacon Burger\t Rs.350");

pane.add(b3, 1, 7);

b3.setTextFill(Color.WHITE);

Label b4=new Label("Umami Burger\t Rs.550");

pane.add(b4, 1, 8);

b4.setTextFill(Color.WHITE);

Label b5=new Label("Minetta Burger\t Rs.350");

pane.add(b5, 1, 9);

b5.setTextFill(Color.WHITE);

//Textfields for subtotal

TextField st1 = new TextField();

st1.setPromptText("SubTotal");

st1.setPrefWidth(0);

pane.add(st1, 4,5);

TextField st2= new TextField();

st2.setPromptText("SubTotal");

st2.setPrefWidth(0);

pane.add(st2, 4,6);

TextField st3= new TextField();

st3.setPromptText("SubTotal");

st3.setPrefWidth(0);

pane.add(st3, 4,7);

TextField st4= new TextField();

st4.setPromptText("SubTotal");

st4.setPrefWidth(0);

pane.add(st4, 4,8);

TextField st5= new TextField();

st5.setPromptText("SubTotal");

st5.setPrefWidth(0);

pane.add(st5, 4,9);

//Spinners for quantity input

Spinner<Integer> spinner1 = new Spinner<>(0, 100, 0);

pane.add(spinner1, 2,5);

Spinner<Integer> spinner2 = new Spinner<>(0, 100, 0);

pane.add(spinner2,2,6);

Spinner<Integer> spinner3 = new Spinner<>(0, 100, 0);

pane.add(spinner3,2,7);

Spinner<Integer> spinner4 = new Spinner<>(0, 100, 0);

pane.add(spinner4,2,8);

Spinner<Integer> spinner5 = new Spinner<>(0, 100, 0);

pane.add(spinner5,2,9);

spinner1.addEventHandler(MouseEvent.MOUSE\_CLICKED, new EventHandler<Event>() {

@Override

public void handle(Event event) {

//System.out.println(getValue.spinner1);

pane.add(new TextField("Rs/="+a), 4,5);

pane.add(new TextField("Rs/"+spinner1.getValue()\*250), 4, 5);

}

});

spinner2.addEventHandler(MouseEvent.MOUSE\_CLICKED, new EventHandler<Event>() {

@Override

public void handle(Event event) {

pane.add(new TextField("Rs/="+b), 4,6);

pane.add(new TextField("Rs/"+spinner2.getValue()\*100), 4, 6);

}

});

spinner3.addEventHandler(MouseEvent.MOUSE\_CLICKED, new EventHandler<Event>() {

@Override

public void handle(Event event) {

pane.add(new TextField("Rs/="+c), 4,7);

pane.add(new TextField("Rs/"+spinner3.getValue()\*350), 4, 7);

}

});

spinner4.addEventHandler(MouseEvent.MOUSE\_CLICKED, new EventHandler<Event>() {

@Override

public void handle(Event event) {

pane.add(new TextField("Rs/="+c), 4,8);

pane.add(new TextField("Rs/"+spinner4.getValue()\*550), 4, 8);

}

});

spinner5.addEventHandler(MouseEvent.MOUSE\_CLICKED, new EventHandler<Event>() {

@Override

public void handle(Event event) {

pane.add(new TextField("Rs/="+c), 4,9);

pane.add(new TextField("Rs/"+spinner5.getValue()\*350), 4, 9);

}

});

//Textfield of Total

final TextField t = new TextField();

t.setPromptText("Total");

t.setPrefWidth(0);

pane.add(t, 2,11);

//Setting of Place order button

Button Place = new Button("Place Order");

pane.add(Place,3,11);

//Function of place order button

Place.addEventHandler(MouseEvent.MOUSE\_CLICKED,new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event) {

a=spinner1.getValue()\*250;

b=spinner2.getValue()\*100;

c=spinner3.getValue()\*350;

d=spinner4.getValue()\*550;

e=spinner5.getValue()\*350;

pane.add(new TextField("Rs/="+a\*2), 4,5);

pane.add(new TextField("Rs/="+b), 4,6);

pane.add(new TextField("Rs/="+c), 4,7);

pane.add(new TextField("Rs/="+d), 4,8);

pane.add(new TextField("Rs/="+e), 4,9);

pane.add(new TextField("Rs/="+(a+b+c+d+e)), 2,11);

}

});

//Settings of Generate bill button

Button bill = new Button("Generate Bill");

pane.add(bill,4,11);

//Function of Generate bill button

bill.addEventHandler(MouseEvent.MOUSE\_CLICKED,new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event) {

// TODO Auto-generated method stub

Thirdpage tp=new Thirdpage();

try {

tp.start(Stage2);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}});

}

}

//Class of Third page

class Thirdpage extends Application

{

//Object for calling second page

Secondpage sp2=new Secondpage();

protected static final Stage Stage1 = null;

@Override

public void start(Stage Stage3) throws Exception {

// TODO Auto-generated method stub

//Setting of Thirdpage + styling

GridPane pane=new GridPane();

Scene scene = new Scene(pane,600,400);

Stage3.setScene(scene);

Stage3.setTitle("BILL");

Label heading1= new Label(" BURGER CLUB");

heading1.setFont(Font.font ("broadway", 15));

heading1.setTextFill(Color.CORAL);

pane.setHgap(5);

pane.setVgap(5);

pane.add(heading1, 0, 2);

Secondpage sp2=new Secondpage();

//To print date

Date date=new Date();

Label b0=new Label("DATE:\n"+date);

pane.add(b0, 1, 4);

b0.setTextFill(Color.BLACK);

//Labels of items with their prices for print

Label b1=new Label(" ITEMS\nBeef Burger\t Rs/="+sp2.a);

pane.add(b1, 1,6);

b1.setTextFill(Color.BLACK);

Label b2=new Label("Chicken Burger\t Rs/="+sp2.b);

pane.add(b2, 1, 7);

b2.setTextFill(Color.BLACK);

Label b3=new Label("Bacon Burger \t Rs/="+sp2.c);

pane.add(b3, 1, 8);

b3.setTextFill(Color.BLACK);

Label b4=new Label("Umami Burger\t Rs/="+sp2.d);

pane.add(b4, 1, 9);

b4.setTextFill(Color.BLACK);

Label b5=new Label("Minetta Burger\t Rs/="+sp2.e);

pane.add(b5, 1, 10);

b5.setTextFill(Color.BLACK);

Label b6=new Label("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

pane.add(b6, 1, 11);

b6.setTextFill(Color.BLACK);

double total=0;

total=sp2.a+sp2.b+sp2.c+sp2.d+sp2.e;

Label b7=new Label("Total\t Rs/="+total);

pane.add(b7, 1, 12);

b7.setTextFill(Color.BLACK);

//Settings of sign out button

Button sgout = new Button("Sign out");

pane.add(sgout,1,14);

//Function of sign out button

sgout.addEventHandler(MouseEvent.MOUSE\_CLICKED,new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event) {

Stage3.close();

}

});

//Settings of new bill button which on click open previous page

Button nb = new Button("New Bill");

pane.add(nb,2,14);

//Function of new bill button

nb.addEventHandler(MouseEvent.MOUSE\_CLICKED,new EventHandler<MouseEvent>() {

@Override

public void handle(MouseEvent event) {

try {

sp2.start(Stage3);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

});

}

}

//THE END

# **UML DIAGRAM OF RESTAURANT MANAGEMENT:**

+ Application

+ Frontp

+Thirdpage

String : Admin;

String : Pass;

String : checkAdmin;

String : checkPass;

+Secondpage

static int: a=0;

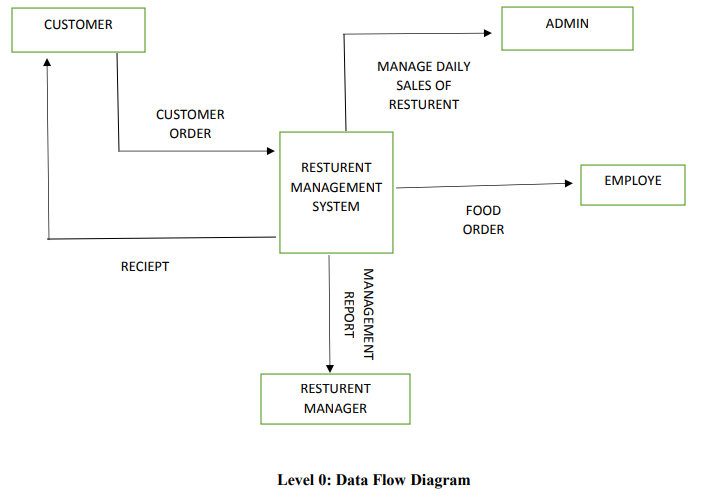
int :b=0;

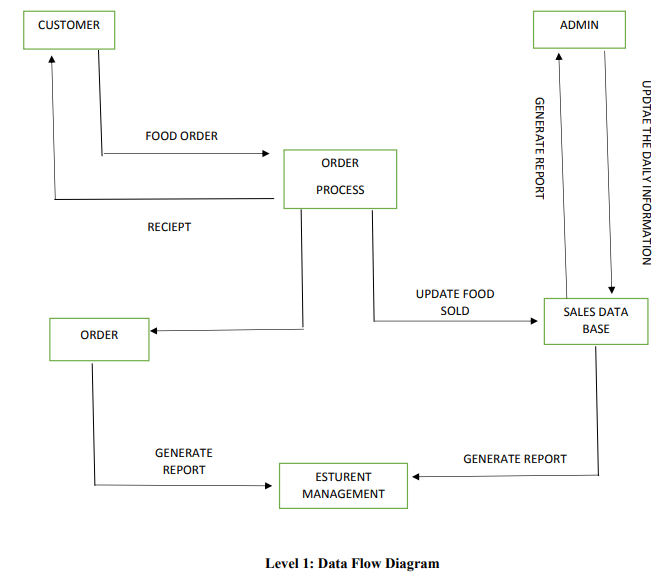
int :c=0;

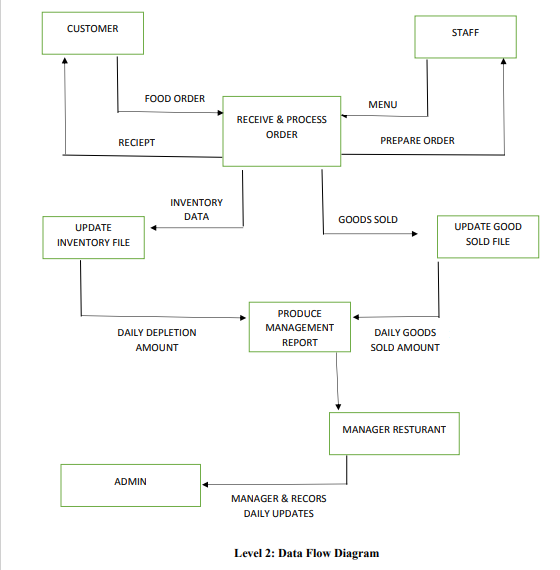
int: d=0;

int :e=0;

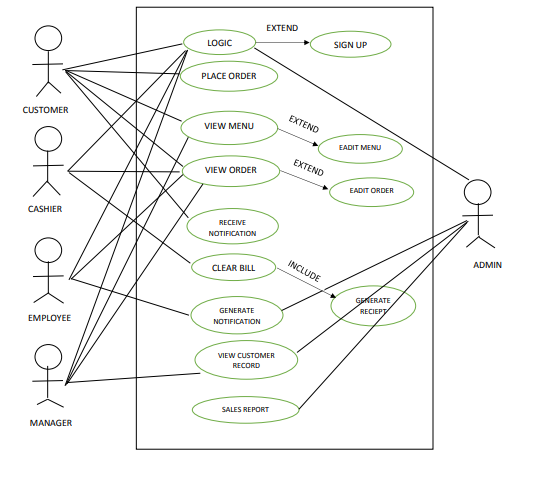
# **SYSTEM REQUIREMENT SPECIFICATION:**







# **USECASE DIAGRAM:**



# **CONCLUSION:**

We were able to create a Online Restaurant Management systems records. This system is able to place order and save customers data and billing records whenever needed easily. Data entering of customers and employees are also included in this system along with the order and the billing process. Customers, restaurant records and employees are interconnected in order to maintain the process. This system can also be further improved adding many other features and including the other systems as well. Finally, we believe that we were able to launch an effective online ordering system to the restaurant causing the restaurant to perform well in the future regarding the billing and ordering online.

# GITHUB LINK:

https://github.com/Maryam7861/Restaurant-Mangagement-system.git