

# **“Foot Traffic Analysis for Optimal Restaurant Operations”**

*A*

## ***Project Report***

*Submitted in partial fulfillment of the*

*Requirements for the award of the degree of*

**Bachelor of Technology with specialization in Internet of Things and Smart Cities**

*Under the guidance of*

**Ms. Amber Hayat**

**Assistant Professor, Department of Systemics**

*Submitted By*

**Ishika Jain**

**SAPID: 500062973**

**ROLL No: R164217022**

**Krishan Kant**

**SAPID: 500062911**

**ROLL No: R164217027**

**Nihit Garg**

**SAPID: 500062280**

**ROLL No: R164217033**



**UNIVERSITY WITH A PURPOSE**

**SCHOOL OF COMPUTER SCIENCE**

**UNIVERSITY OF PETROLEUM AND ENERGY STUDIES**

**Bidholi via Prem Nagar, Dehradun**

**Uttarakhand, India**

**April-2020**



## **CANDIDATE'S DECLARATION**

We here by certify that the project work entitled “**Foot Traffic Analysis for Optimal Restaurant Operations**” in partial fulfillment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Internet Of Things and Smart Cities submitted to the School of Computer Science, Department of Systemics, University of Petroleum & Energy Studies, Dehradun, is an authentic record of our work carried out during a period from **February, 2020** to **May, 2020** under the supervision of **Ms. Amber Hayat**, Assistant Professor.

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

**(Ishika Jain, Krishan Kant, Nihit Garg)**

**(R164217022, R164217027, R164217033)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

**Date:** 06 April, 2019

Ms. Amber Hayat

**Project Guide**

**Dr. Neelu Jyoti Ahuja**

HOD, Department of Systemics

School of Computer Science

University of Petroleum & Energy Studies

Dehradun – 248007 (Uttarakhand)

---

## **ACKNOWLEDGEMENT**

We wish to express our deep gratitude to our guide Ms. Amber Hayat for all the advice, encouragement and constant support that he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We are also grateful to **Mr. Alok Jhaldiyal, our Activity Co-Ordinator** for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our friends for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

<b>Name</b>	Ishika Jain	Krishan Kant	Nihit Garg
<b>Roll No.</b>	R164217022	R164217027	R164217033

## **ABSTRACT**

An accurate foot traffic Analysis system can help retail organizations, physical stores, and restaurants to optimize their labor schedule and costs, and reduce food wastage and optimize their operations. Foot traffic data can provide actionable insights across virtually any industry including retail and restaurant, real estate, financial services, and insurance, hospitality and travel, sports and entertainment, and automotive. While these sectors may have unique challenges for reaching target audiences, data on consumer movement in the real world helps provide businesses with clarity into their preferences.

Based on foot traffic predictions, businesses can adjust staffing and product stock levels. The further a business can accurately forecast foot traffic into the future, the more it can optimize operations management, product management and in consequence grow profits. Therefore, an accurate forecast of foot traffic, by a Restaurant manager, can help increase customers' satisfaction, increase sales, and reduce food waste.

This project aims at developing, a set of data collection from the billing receipts of the Restaurants and analyzing system for Restaurant's foot traffic. Our data has been collected from the data of the billing receipts in the restaurants such as time of order, type of ordered food etc. This type of data is centrally processed and analyzed to predict that at what period of time in the day the crowd is maximum, average and minimum and what type of food is being mostly liked by the people at what time. This type of data is being process to predict this foot traffic in the time intervals of one month to experience much better analyzing of data.

.

## **TABLE OF CONTENTS**

<b>S. No.</b>	<b>Contents</b>	<b>Page No</b>
<b>1.</b>	<b>Introduction</b>	<b>1-2</b>
	1.1 A Brief Introduction	
	1.2 Background of The Study	
<b>2.</b>	<b>Problem Statement</b>	<b>3</b>
<b>3.</b>	<b>System Requirements</b>	<b>4</b>
	3.1 Hardware Requirements	
	3.2 Software Requirements	
<b>4.</b>	<b>Methodology</b>	<b>5-7</b>
	4.1 Flowchart	
	4.2 Use Case Diagram	
	4.3 E-R Diagram	
<b>5.</b>	<b>Pert chart</b>	<b>8</b>
<b>6.</b>	<b>Implementation</b>	<b>9-34</b>
<b>7.</b>	<b>Output Screen</b>	<b>35-38</b>
<b>8.</b>	<b>Result</b>	<b>39</b>
<b>9.</b>	<b>References</b>	<b>40</b>

# **1. INTRODUCTION**

## **1.1 A Brief Introduction**

Prior a business even decides to set up Restaurant in an area, the Business research team will study the foot traffic in the area during different times of the day and the week. Restaurants that are chain stores or franchises study traffic patterns in an area as well as median income, crime rates, and local foot traffic. If it's a larger Restaurant and is interested in the area, it might contract out the work. Consultants are often used to conduct surveys and analyses\ foot traffic patterns at different times of the year [1]. Footfall insights can help reinforce the effectiveness of marketing campaigns and encourage deals since they reveal where customers prefer or like to eat, as well as where they are likely to walk around once arriving at their destination. This information permits Restaurant managers to reliably understand when, where, and how to connect with the target audiences so they can make a personalized outreach campaign optimized for engagement.

Ways that detailed foot traffic data help the restaurants [2]?

- 1- Optimize the serving time of the Dishes.
- 2- Understand the consumer behavior patterns
- 3- Predict demands and traffic patterns.
- 4- Improving staff scheduling.
- 5- Analyze the performance of displays.

A study by Veeling [3] worked on improving foot traffic forecasting in 11 retail stores with neural networks. In the study, security gates are used to count people entering or exiting the physical stores. Beams placed at the gates are broken when people walk through them. To keep complexity low, only daily foot traffic is used in the prediction model. In this project, our aim is to analyze foot traffic based on historical data also we aim to analyze monthly and this contains the billing receipt data of every customer. After pre-processing the raw data, we built a model that gives total report of the last month to the manager of the restaurant such that they can optimize the restaurant operations.

## **1.2 Background of The Study**

Many types of businesses, including for- and non-profits, rely on people patronizing them. Without adfoot equate pedestrian traffic near your business, you may not hit the numbers you need. But getting people into the door is not the only reason you will want to evaluate foot

traffic data. Identifying pedestrian cycles gives you the information you need to optimize your operations. Whatever your sector, you will find a use for measuring traffic in the area [4].

There are many ways through which the one can give the benefits through their business but the modern strategies that are used is through the help of data science analysis and there are many research work that has been developed in this field. By research of Compliant IA[5]the Foot traffic analysis evolved from simply body counting to a more sophisticated set of analyses, akin to those available online. Now, data driven retailers can collect rich insights about not only how many people enter their store but also about their behavior while there.

Foot traffic is a critical barometer for retailers,” says David Barstow, SVP of product, Skyhook. “With so many signals competing for consumer attention, it’s important to understand how many consumers are visiting your stores relative to your competitors.” Retailers can “identify trends within their own CRM data to explain some of the changes but they can’t put together the entire story without understanding the flow of foot traffic,” he says. Foot traffic analysis ties everything together.

Our main aim of the project is to predict foot traffic based on historical data also we aim to analyze monthly and this contains the billing receipt data of every customer. After pre-processing the raw data, we built a model that gives total report of the last month to the manager of the restaurant such that they can optimize the restaurant operations. So the research work by

and study by Cortez et al. [6] it shown that they used one of the idea to analyse the foot traffic by the use of digital cameras to detect foot traffic at a sports store. Data mining methods are applied to build foot traffic forecasting models. The camera is linked with a human facial recognition system which counts the foot traffic and groups the traffic into three categories: all faces, female faces and male faces. The daily foot traffic combined with other factors like weather and special events are used to build a prediction model.

## **2. PROBLEM STATEMENT**

- In earlier days, without any computational power it is very difficult to analyze the whole progress of the running restaurant. There are many major problems ahead in operating the restaurants. Sometimes it is seen that the restaurant is established in the certain areas but it is no longer keep running because they do not have the proper reports about the operations that is being running in the restaurants.
- The aim of Foot Traffic analyzing is not only to detect the detect the crowd in the restaurants but it's main aim is to give the progress report to the restaurants such that they can analyze it and run optimal operations in the restaurants.
- We need to diminish the problems like:
  - 1) Lost sales tracking, which helps businesses plan inventory. If selling a certain type of dish was poor during peak foot traffic hours, that dish might not be worth selling.
  - 2) How much time spent in the restaurants or the average time spent by customers. Data showing time spent in the restaurants combined with the time of day, week, or year can provide critical insight to customer buying patterns and interests.
  - 3) Numbers and what type of dishes are buying by the customers.



### **3. SYSTEM REQUIREMENTS**

#### **3.1 Hardware Requirements:**

Any computing machine with following specifications:

Processor	:	INTEL I5 7 GEN
Hard Disk	:	1 TB
RAM	:	8 GB

#### **3.2 Software Requirements:**

- Apache-NetBeans IDE-11.3-bin-windows-x64
- Windows 10
- MySQL
- jdk-12\_windows-x64\_bin

## 4. METHODOLOGY

We're deploying a restaurant management system where we can manage Items, Order, Employee along with the prediction system where we can fetch data from the database as entered by the user and accordingly make suggestions.

For analysing the data we are using Greedy Algorithm specifically 0/1 Knapsack Problem. We're using this algorithm for calculating maximum and minimum frequency, for example frequency of number of dishes sold and then using the output for the prediction.

### 4.1 Flowchart

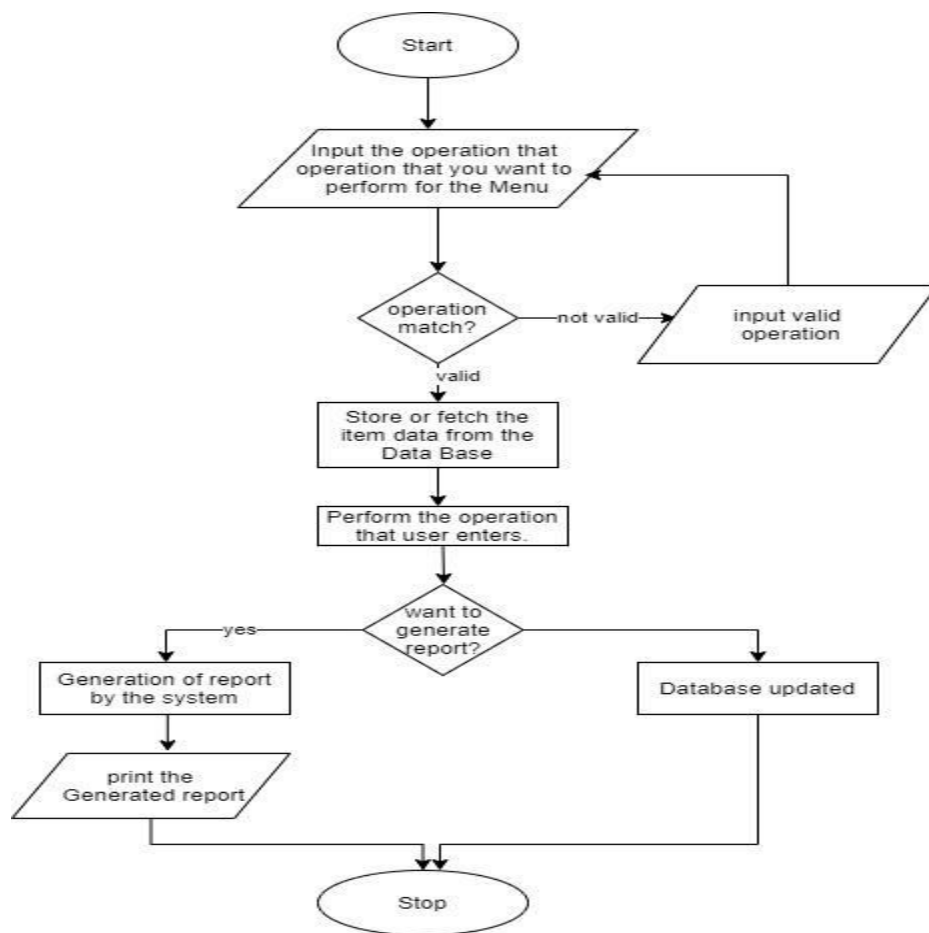


Figure 1: Flow chart

## 4.2 Use Case Diagram

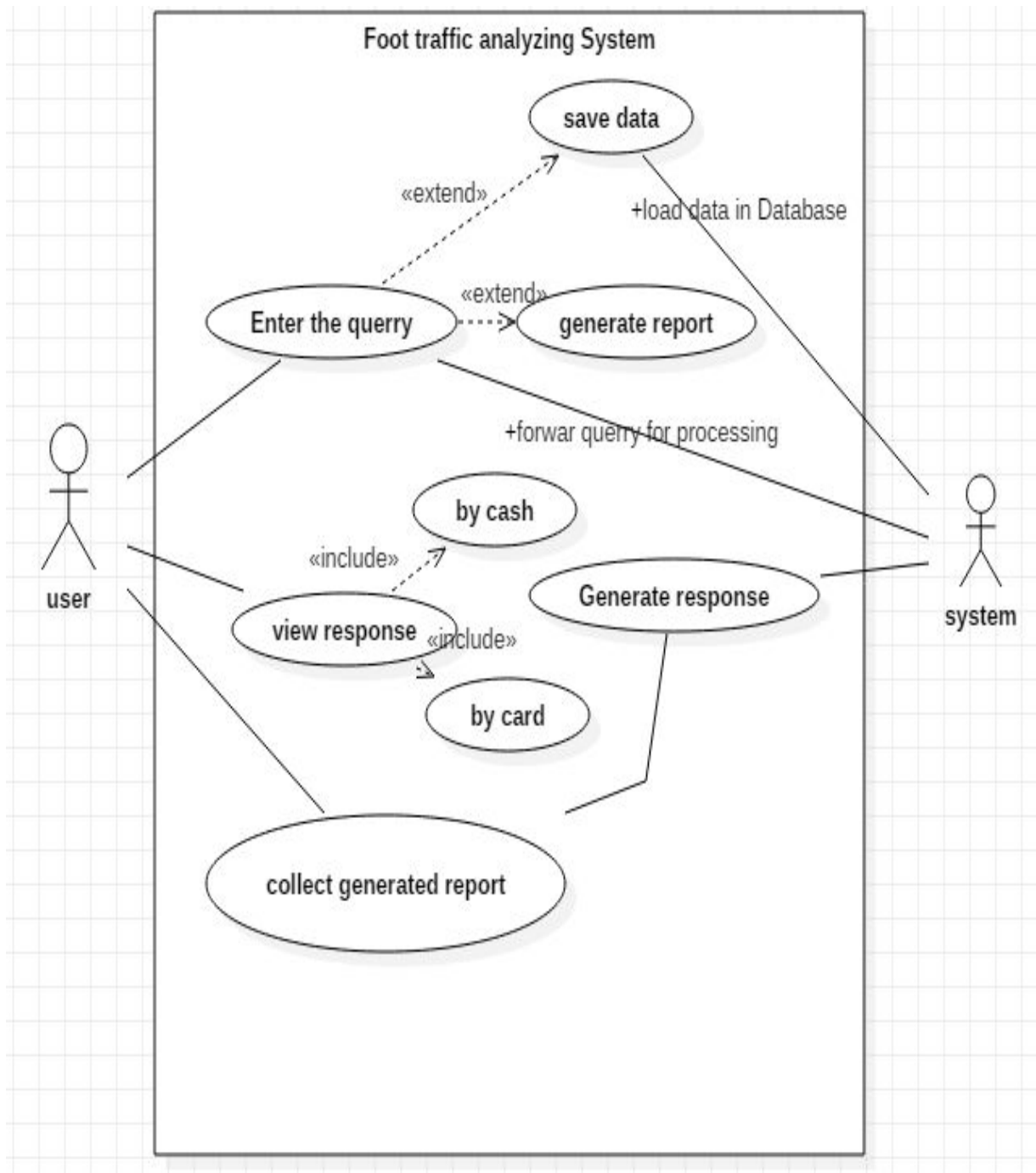
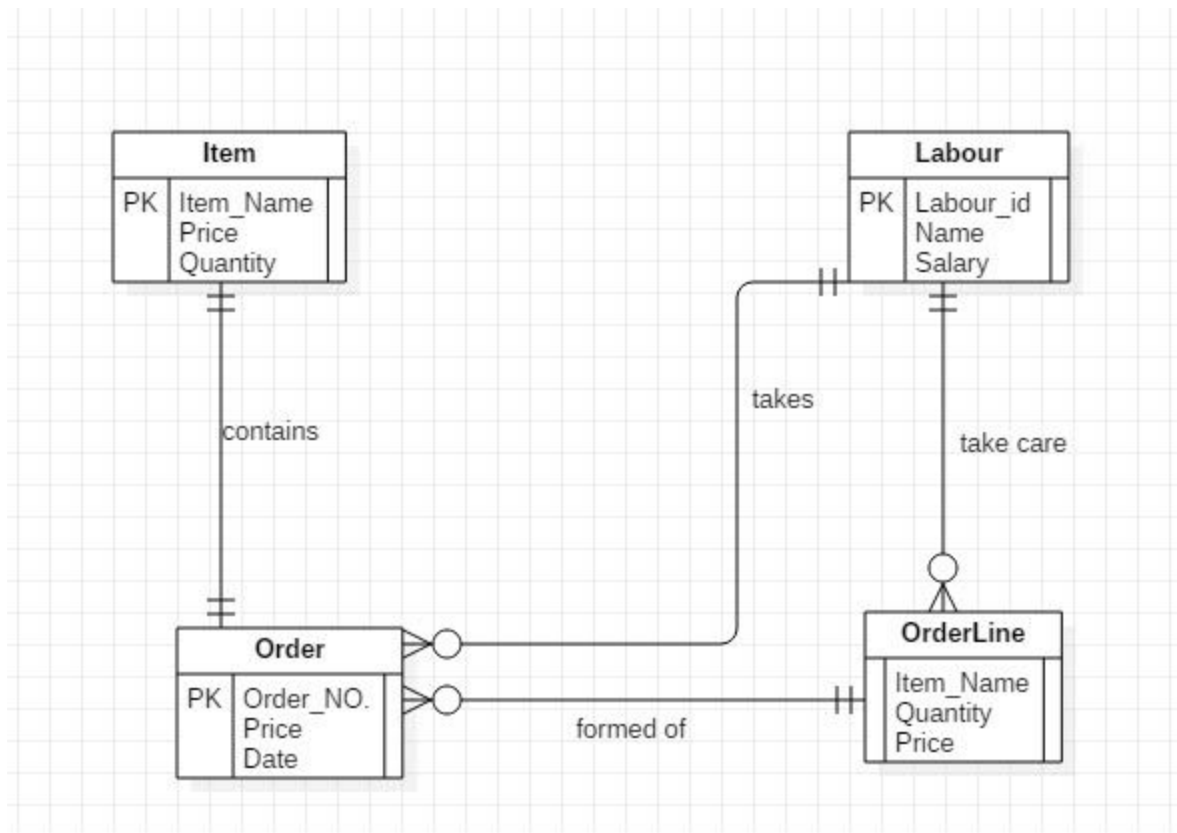
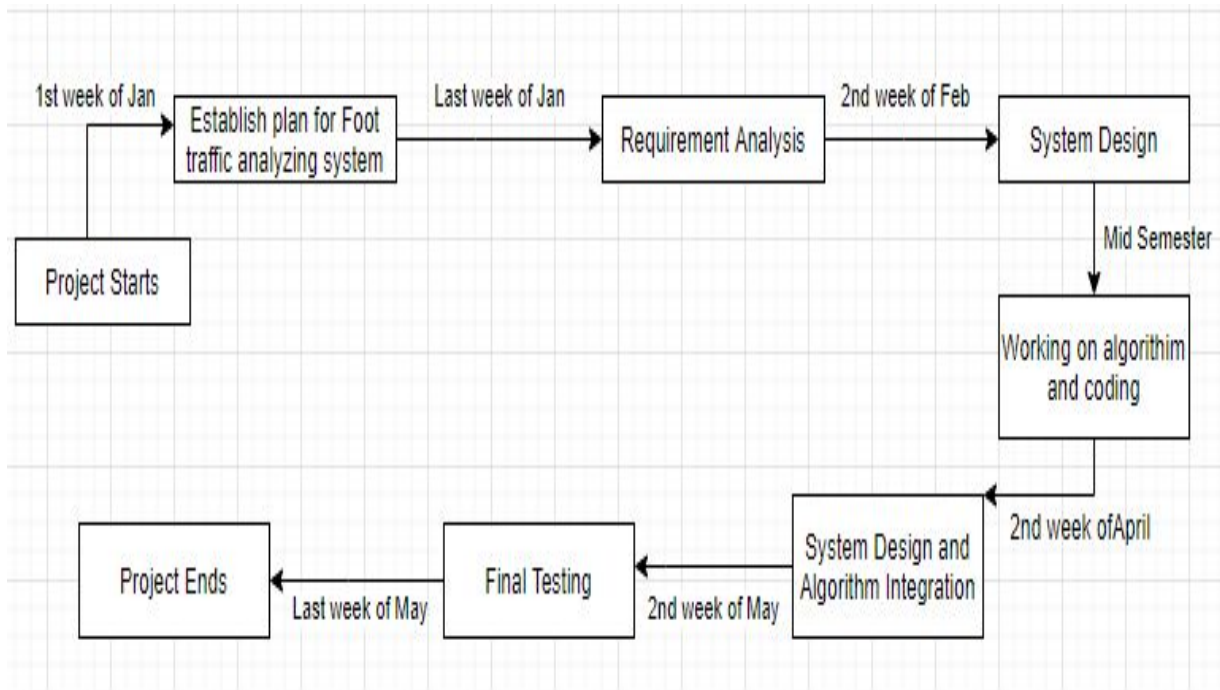


Figure 2 use case diagram of foot traffic analyzing system

### 4.3 E-R Diagram



## **5. Pert Chart**



*Figure 4 pert chart*

## **6. IMPLEMENTATION**

**CODE:**

**item.java:**

```
package restaurantsystem.model;

public class Item {
    private String name;
    private double price;
    private int quantity;

    public Item(String name, double price, int quantity) {
        this.name = name;
        this.price = price;
        this.quantity = quantity;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public double getPrice() {
```

```

        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public int getQuantity() {
        return quantity;
    }

    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }
}

```

**Order.java:**

```

package restaurantsystem.model;

public class Order {
    private int orderID;
    private double price;
    private String date;

    public Order(int orderID, double price, String date) {
        this.orderID = orderID;
    }
}

```

```
        this.price = price;
        this.date = date;
    }

    public int getOrderID() {
        return orderID;
    }

    public void setOrderID(int orderID) {
        this.orderID = orderID;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }
}
```



### **Login.java:**

```
package restaurantsystem.component.auth;

import javax.swing.JOptionPane;
import restaurantsystem.MainMenu;

public class Login extends javax.swing.JFrame {
    public Login() {
        initComponents();
        setDefaultUserName();
    }
    private void setDefaultUserName() {
        this.userNameField.setText("");
        this.passwordField.setText("");
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
    private void initComponents() {

        loginPanel = new javax.swing.JPanel();
        userNameLabel = new javax.swing.JLabel();
        userNameField = new javax.swing.JTextField();
        passwordLabel = new javax.swing.JLabel();
        loginButton = new javax.swing.JButton();
        passwordField = new javax.swing.JPasswordField();
```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

loginPanel.setBackground(new java.awt.Color(0, 0, 0));
loginPanel.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Restaurant
Management and Prediction System",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Lucida
Grande", 0, 13), new java.awt.Color(255, 255, 255))); // NOI18N
loginPanel.setForeground(new java.awt.Color(255, 255, 255));
loginPanel.setToolTipText("");
loginPanel.setName("");

userNameLabel.setBackground(new java.awt.Color(255, 153, 153));
userNameLabel.setForeground(new java.awt.Color(255, 255, 255));
userNameLabel.setText("User Name: ");

userNameField.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        userNameFieldActionPerformed(evt);
    }
});

passwordLabel.setForeground(new java.awt.Color(255, 255, 255));
passwordLabel.setText("Password :");

loginButton.setText("Log In");
loginButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        loginButtonActionPerformed(evt);
    }
});

javax.swing.GroupLayout loginPanelLayout = new
javax.swing.GroupLayout(loginPanel);
loginPanel.setLayout(loginPanelLayout);
loginPanelLayout.setHorizontalGroup(

loginPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(loginPanelLayout.createSequentialGroup()
        .addGap(31, 31, 31)

    .addGroup(loginPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(loginButton)
        .addGroup(loginPanelLayout.createSequentialGroup()

    .addGroup(loginPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(userNameLabel,
            javax.swing.GroupLayout.PREFERRED_SIZE, 89,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(passwordLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(loginPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

```

```

        .addComponent(userNameField)
        .addComponent(passwordField,
javax.swing.GroupLayout.DEFAULT_SIZE, 116, Short.MAX_VALUE))))
        .addContainerGap(57, Short.MAX_VALUE))
    );
    loginPanelLayout.setVerticalGroup(

loginPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(loginPanelLayout.createSequentialGroup()
            .addGap(23, 23, 23)

        .addGroup(loginPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(userNameLabel)
            .addComponent(userNameField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)

        .addGroup(loginPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(passwordLabel)
            .addComponent(passwordField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addComponent(loginButton)
            .addContainerGap(34, Short.MAX_VALUE))
    );

```

```

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(45, 45, 45)
            .addComponent(loginPanel, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(46, Short.MAX_VALUE))
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(54, 54, 54)
            .addComponent(loginPanel, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGap(67, Short.MAX_VALUE))
        );

loginPanel.getAccessibleContext().setAccessibleName("");

pack();
} // </editor-fold> // GEN-END: initComponents

private void loginButtonActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_loginButtonActionPerformed
    if (userNameField.getText().equalsIgnoreCase("123") &&
passwordField.getText().equalsIgnoreCase("123")) {

```

```

        JOptionPane.showMessageDialog(this, "Access granted");
        this.dispose();
        MainMenu mainMenu = new MainMenu();
        mainMenu.setVisible(true);
    } else {
        userNameField.setText("");
        passwordField.setText("");
        JOptionPane.showMessageDialog(this, userNameField.getText() + " Access Denied");
    }

}

private void userNameFieldActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_userNameFieldActionPerformed

    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {

        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        }
    }
}

```

```

    }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    }

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(() -> {
        new Login().setVisible(true);
    });
}

private javax.swing.JButton loginButton;
private javax.swing.JPanel loginPanel;

```

```
private javax.swing.JPasswordField passwordField;
private javax.swing.JLabel passwordLabel;
private javax.swing.JTextField userNameField;
private javax.swing.JLabel userNameLabel;

}
```

**Labour.java:**

```
package restaurantsystem.model;

public class Labour {
    private String id;
    private String name;
    private double salary;

    public Labour(String id, String name, double salary) {
        this.id = id;
        this.name = name;
        this.salary = salary;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
```



```

        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }
}

```

**ViewItem.java:**

```

package restaurantsystem.component.Prediction;

import java.awt.event.ActionEvent;
import java.util.*;

import restaurantsystem.service.ItemService;

```

```

class ViewItem extends javax.swing.JFrame {

    private final ItemService itemService;

    public ViewItem() {
        initComponents();
        this.itemService = new ItemService();
        performFileRelatedTask();
    }

    public void performFileRelatedTask() {
        Hashtable<String,Integer> itemQuantity=new Hashtable<String,Integer>();

        ArrayList<String> itemN=new ArrayList<String>();
        ArrayList<Integer> itemQ=new ArrayList<Integer>();

        itemService.getAll().forEach((item) -> {

            itemQuantity.put(item.getName(),item.getQuantity());

        });

        itemService.getAll().forEach((item) -> {

            itemN.add(item.getName());
            itemQ.add(item.getQuantity());
        });
    }
}

```

```

    });

    int max=0;
    String itemName="";
    String name="";
    int count=0;
    String maxItem="";

    for(Map.Entry m:itemQuantity.entrySet()){
        if((int)m.getValue()>max) {
            max=(int)m.getValue();
            itemName=(String) m.getKey();

        }
    }

    //text.setText("Order with Max Quantity\t"+itemName);

    for(int i=0;i<itemN.size();i++)
    {
        name=itemN.get(i);
        int freq = Collections.frequency(itemN, name);
        if(freq>count) {
            count=freq;
            maxItem=name;
        }
    }
}

```

```

StringBuilder fullnames = new StringBuilder();

fullnames.append("Prediction:"+maxItem+" will be the most ordered in future")
    .append("\n")
    .append("Suggestion:"+maxItem+" can be prepared in large quantity to reduce
wait time");

// System.out.println("Most ordered Food\n"+maxItem);
//text.setText("Prediction:"+maxItem+" will be the most ordered in future");
//text.setText("Suggestion:"+maxItem+"can be prepared in large quantity to reduce wait
time");
text.setText(fullnames.toString());

}

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN:initComponents
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    text = new javax.swing.JTextArea();
    backButton = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();

```

```

jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(204,204,204));

text.setBackground(new java.awt.Color(192,192,192));
text.setColumns(20);
text.setRows(5);
jScrollPane1.setViewportView(text);

backButton.setText("Back");
backButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        backButtonActionPerformed(evt);
    }
});

jLabel1.setText("Item Prediction");

jLabel2.setText("");

jLabel3.setText("");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(layout.createSequentialGroup())

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(300, 300, 300)
            .addComponent(backButton))
        .addGroup(layout.createSequentialGroup()
            .addGap(53, 53, 53)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 301,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(layout.createSequentialGroup()
            .addGap(9, 9, 9)
            .addComponent(jLabel1)
            .addGap(54, 54, 54)
            .addComponent(jLabel2)
            .addGap(63, 63, 63)
            .addComponent(jLabel3))))))
        .addContainerGap(74, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap(28, Short.MAX_VALUE)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(jLabel1)
        .addComponent(jLabel2)
        .addComponent(jLabel3))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 137,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(42, 42, 42)
        .addComponent(backButton)
        .addGap(50, 50, 50))
    );

    pack();
    setLocationRelativeTo(null);
} // </editor-fold> // GEN-END: initComponents

private void backButtonActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_backButtonActionPerformed
    PredictionManagement im = new PredictionManagement();
    im.setVisible(true);
    this.setVisible(false);
} // GEN-LAST: event_backButtonActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    try {

```

```

        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(ViewItem.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ViewItem.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(ViewItem.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(ViewItem.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(() -> {

```



```

        new ViewItem().setVisible(true);
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton backButton;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextArea text;
// End of variables declaration//GEN-END:variables

public void actionPerformed(ActionEvent e) {

}

}

```

### **ViewLabour.java:**

```

package restaurantsystem.component.Prediction;

import java.util.ArrayList;

import restaurantsystem.service.LabourService;
public class ViewLabour extends javax.swing.JFrame {

```

```

private final LabourService labourService;

/**
 * Creates new form ViewLabour
 */
public ViewLabour() {
    initComponents();
    this.labourService = new LabourService();
    performFileRelatedTask();
}
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN:initComponents
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    text = new javax.swing.JTextArea();
    backButton = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setBackground(new java.awt.Color(0, 153, 204));

    text.setEditable(false);
    text.setColumns(20);
    text.setRows(5);

```

```
jScrollPane1.setViewportView(text);
```

```
backButton.setText("Back");
```

```
backButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        backButtonActionPerformed(evt);  
    }  
});
```

```
jLabel1.setText("Labour Prediction");
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());  
getContentPane().setLayout(layout);
```

```
layout.setHorizontalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(layout.createSequentialGroup()  
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                    .addGroup(layout.createSequentialGroup()  
                        .addGap(315, 315, 315)  
                        .addComponent(backButton))  
                    .addGroup(layout.createSequentialGroup()  
                        .addGap(54, 54, 54)  
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,  
301, javax.swing.GroupLayout.PREFERRED_SIZE))  
                    .addGroup(layout.createSequentialGroup()  
                        .addGap(71, 71, 71)
```

```

        .addComponent(jLabel1)
        .addGap(64, 64, 64)
        .addComponent(jLabel2)
        .addGap(49, 49, 49)
        .addComponent(jLabel3)))
    .addContainerGap(45, Short.MAX_VALUE))
);

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel1)
            .addComponent(jLabel2)
            .addComponent(jLabel3))
            .addGap(5, 5, 5)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 148,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 60,
Short.MAX_VALUE)
            .addComponent(backButton)
            .addGap(50, 50, 50))
);

pack();
setLocationRelativeTo(null);
} // </editor-fold> // GEN-END: initComponents
private void performFileRelatedTask() {

```

```

StringBuilder stringBuilder = new StringBuilder();
    ArrayList<Integer> labourN=new ArrayList<Integer>();
    int size=0;

    labourService.getAll().forEach((item) -> {

        labourN.add((Integer.parseInt(item.getId())));

    });
    size=labourN.size();
    if(size<5) {
        stringBuilder.append("Prediction:"+" There are less no of staff members")
            .append("\n")
            .append("Suggestion:"+" We can add more staff members for reducing the wait
time");
    }
    if(size>10) {
        stringBuilder.append("Prediction:"+" There are more number of staff members than
required")
            .append("\n")
            .append("Suggestion:"+" We can remove some staff members as it isn't required");
    }
    text.setText(stringBuilder.toString());
}

private void backButtonActionPerformed(java.awt.event.ActionEvent evt)
{
    GEN-FIRST:event_backButtonActionPerformed
        PredictionManagement lm = new PredictionManagement();

```

```

        lm.setVisible(true);
        this.setVisible(false);
    } //GEN-LAST:event_backButtonActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(ViewLabour.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
    } catch (InstantiationException ex) {

```

```
java.util.logging.Logger.getLogger(ViewLabour.class.getName()).log(java.util.logging.Level.  
SEVERE, null, ex);
```

```
    } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(ViewLabour.class.getName()).log(java.util.logging.Level.  
SEVERE, null, ex);
```

```
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(ViewLabour.class.getName()).log(java.util.logging.Level.  
SEVERE, null, ex);
```

```
    }
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

```
java.awt.EventQueue.invokeLater(() -> {
```

```
    new ViewLabour().setVisible(true);
```

```
});
```

```
}
```

```
// Variables declaration - do not modify//GEN-BEGIN:variables
```

```
private javax.swing.JButton backButton;
```

```
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JLabel jLabel2;
```

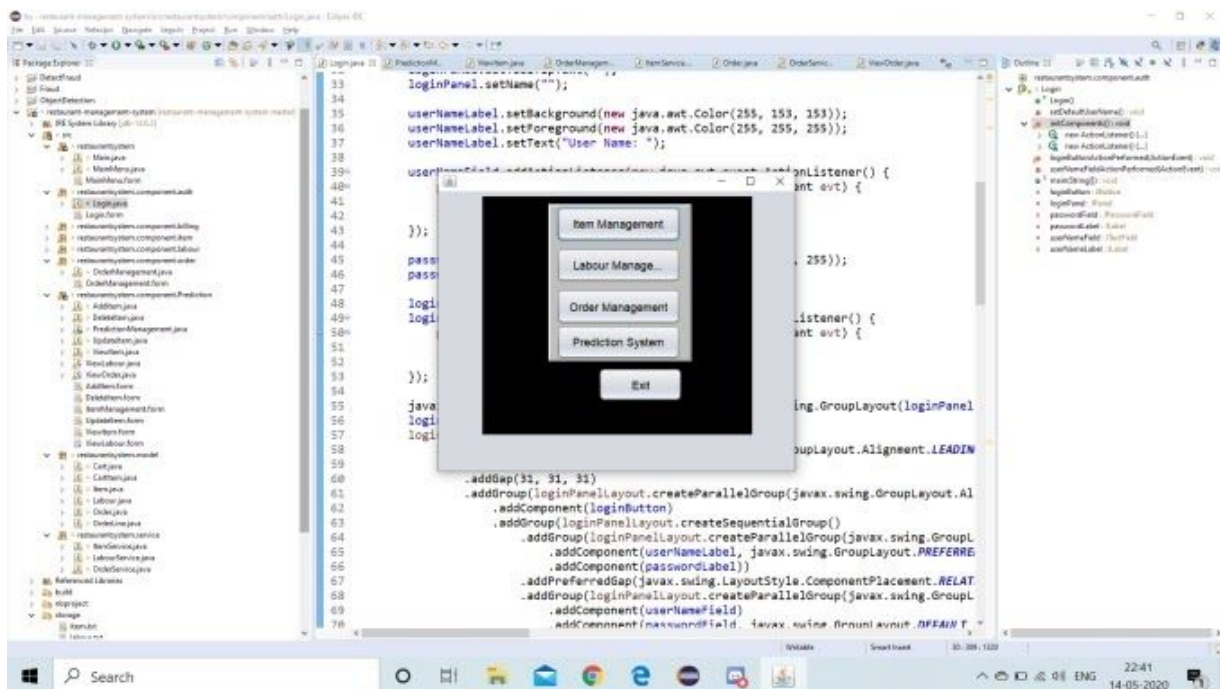
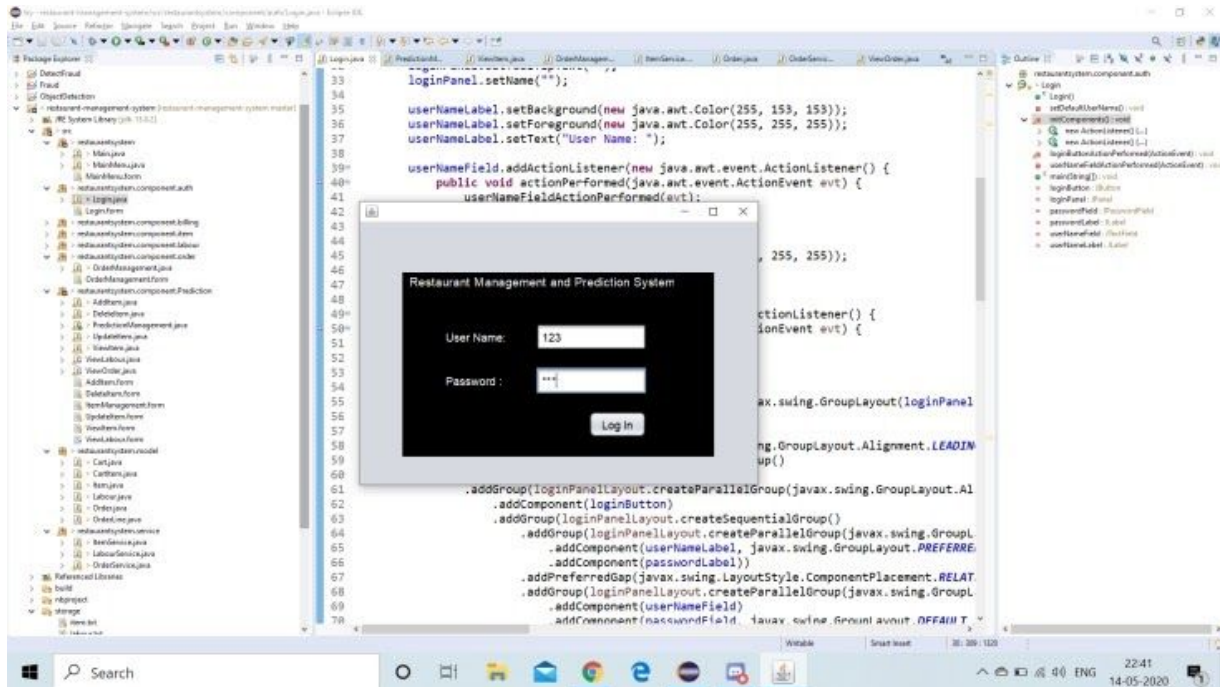
```
private javax.swing.JLabel jLabel3;
```

```
private javax.swing.JScrollPane jScrollPane1;
```

```
private javax.swing.JTextArea text;
```

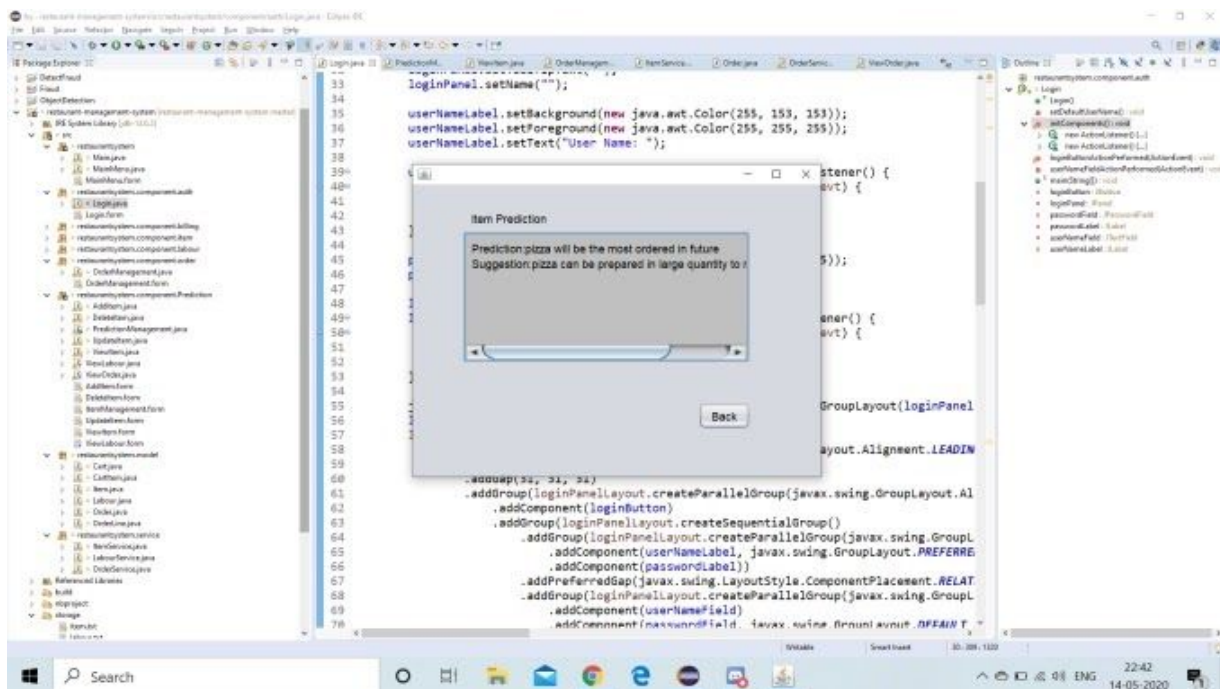
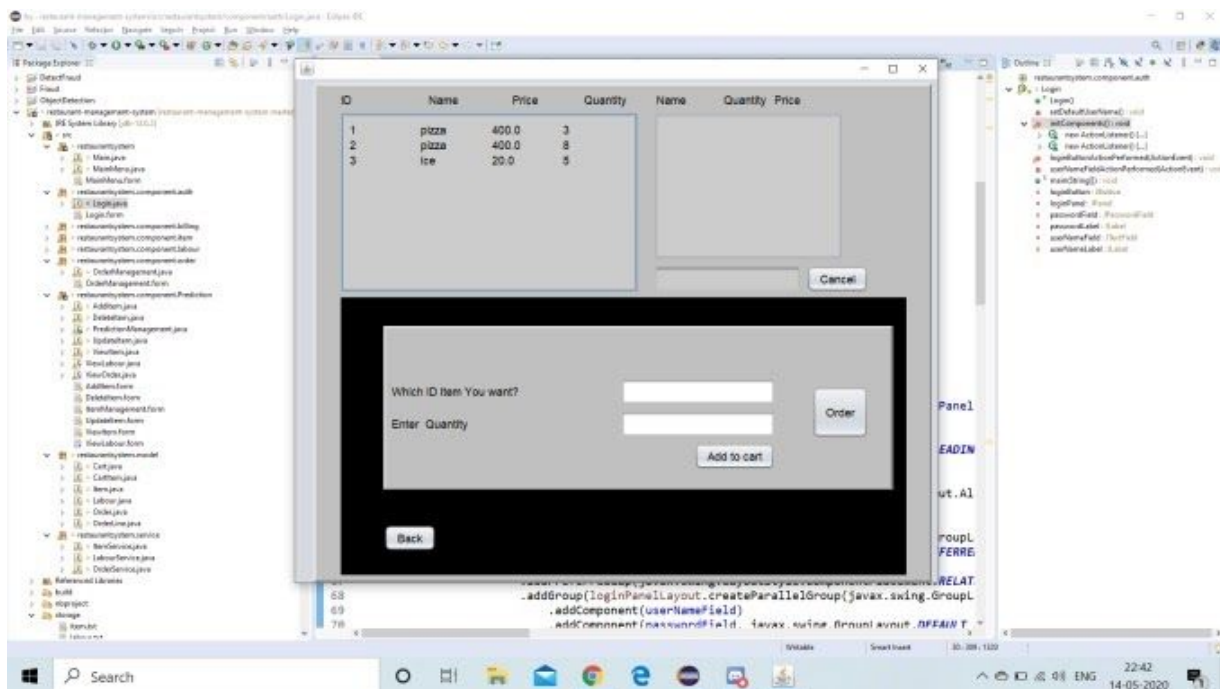
```
// End of variables declaration//GEN-END:variables}
```

## 7. Output Screen











## **8. Result**

This application is working as a restaurant management system which takes care of Item list in the menu, Employees working in the restaurant, orders and queue in which orders are placed.

In addition to this every time an order is placed or an item is added to the menu database would be updated. Also with that data and prediction system most ordered dish and suggestions for better operations are displayed along with suggestions related to the workforce.

Using these suggestions user can alter his restaurant operations and yield better profit.

## **REFERENCES**

- [1.]Foot Traffic by Will Kenton, updated in: - Updated Jan 29, 2020  
url: <https://www.investopedia.com/terms/f/foot-traffic.asp>
  
- [2.] McKinsey & Company, Consumer Marketing Analytics Center, “Creating Competitive Advantage from Big data in Retail,” June 2012.  
url:<https://www.intel.com/content/dam/www/public/us/en/documents/solution-briefs/sc-analytics-retail-foot-traffic-analytics-solution-brief.pdf>
  
- [3.]Veeling, B.: Improving Visitor Traffic Forecasting in Brick-and-Mortar Retail Stores with Neural Networks. Bachelor thesis, Universiteit Twente (2014)
  
- [4.] By Traf-Sys Posted on – 13 September 2019  
url: <https://www.trafsys.com/how-to-measure-foot-traffic/>
  
- [5.] The Evolution of foot traffic analysis and how to use it in 2020,” Posted on December 17 2019, By Compliant IA”.  
url: <https://www.trafsys.com/how-to-measure-foot-traffic/>
  
- [6.] Cortez, P., Matos, L.M., Pereira, P.J., Santos, N., Duque, D.: Forecasting store foot traffic using facial recognition, time series and support vector machines. In: International Joint Conference SOCO’16-CISIS’16-ICEUTE’16: San Sebasti’an, Spain, October 19th-21st, 2016 Proceedings, pp. 267–276. Springer International Publishing (2017),  
url:[http://dx.doi.org/10.1007/978-3-319-47364-2\\_26](http://dx.doi.org/10.1007/978-3-319-47364-2_26)