
TEXT BOOK RESTAURANT, BAR & CATERING MANAGEMENT



By :

ADIL SISWANTO, SST.Par., M.Par

**FAKULTAS ILMU ADMINISTRASI
UNIVERSITAS BRAWIJAYA**

2014

FOREWORD

By giving thanks to the presence of Allah the Almighty, finally the Textbook for the course “TEXT BOOK RESTAURANT, BAR & CATERING MANAGEMENT” has been launched. This textbook is prepared to meet the needs of students of the Department of Business Administration, Tourism Studies Program, Universitas Brawijaya, semester II. to understand and appreciate the concept of professional ethics in general.

“Practice makes perfect”, that this textbook is not enough just to read. Therefore, as supporters of this textbook, students are expected to read appropriate materials from various sources, whether from literature or the Internet related to real applications of "RESTAURANT, BAR & CATERING MANAGEMENT" in the real world of work, so that what is presented in the textbook is not only "like an ivory tower".

Don't forget to say thank you to fellow lecturers of Universitas Brawijaya, for their constructive suggestions. To the students of the Department of Business Administration, Tourism Studies Program, who have provided additional insight and understanding to the Compilers, the Compilers convey high appreciation. The author conveys his sincere gratitude to his beloved wife who with all her understanding and encouragement has made the compiler feel confident that the making of this textbook will never be in vain.

Readers' criticisms and suggestions for the improvement and completeness of this textbook are really expected by the authors. Finally, the authors hope that this Textbook will provide benefits and benefits for the readers, as well as contribute to the development of knowledge treasures. Insya Allah.

Wassalamualaikum warahmatullahi wabarakaatuh.

Malang, 09 January 2014

Adil Siswanto

1. INTRODUCTION

Restaurant Management System is an application which will help restaurant to optimized and control over their restaurants. For the waiters, it is making life easier because they don't have to go kitchen and give the orders to chef easily. For the management point of view, the manager will able able to control the restaurant by having all the reports to hand and able to see the records of each employees and orders.

This application helps the restaurants to do all functionalities more accurately and faster way. Restaurant Management System reduces manual works and improves efficiency of restaurant. This application is helping restuarants to maintain the stock and cash flows and there are many more functionalities, like.

- To store records.
- Control orders and services.
- Billings.
- Control staff and their shifting.
- Control multiple branches.
- Helps Manager to control each part of the restaurant.

The main goal is to maintain the restaurant's functions in an effective and accurate manner and also it is reducing the use of manual entries. This software helps restaurant's to maintain day to day records in system. It is keeping a proper record of the database.

1.1 PROBLEM STATEMENT:

Many restaurants is storing all of their data in manual way. They have huge number of customers daily. So because large number of customers, they need the help of some features so they can maintain and stores the records accurately. For managers it is difficult to view the tables, orders, kitchen, reception and the counter simultaneously.

They need full-fledged software to maintain their day to day transactions, orders and also regular update on records, cash transaction, daily staffs reports, customers feedbacks etc. In the existing system, entering all the details are done manually, it is taking lots of time and also there are chances for mistakes.

1.2 DOMAIN STUDY

Restaurant management is the profession of managing a restaurant. The management is mainly divided into three broader section.[\[1\]](#)

- **Floor management:** 'Floor management' includes managing staff who give services to customers and allocate the duties of opening and closing restaurant. The manager is responsible for making sure his or her staff is following the service standards and health and safety regulations. The manager is the most important person in the front-of-the-house environment, since it is up to him or her to motivate the staff and give them job satisfaction. The manager also looks after and guides the personal well-being of the staff, since it makes the work force stronger and more profitable.
- **Kitchen Management:** 'Kitchen Management' includes the managing staff working in the kitchen, especially the head chef. The kitchen is the most important part of the business and the main reason customers patronize the restaurant. Managing the kitchen staff helps to control food quality. As most commercial kitchens are a closed environment, the staff may become bored or tired from the work. Without proper management, this often results in an inconsistent food product. Kitchen management involves most importantly, cost control and budgeting. Meeting KPI's are a must for a restaurant to survive.

Head chefs must instill and teach money management to apprentices. This is as important as teaching the art and skills of cookery.

- **Administration:** 'Administration' includes stock controlling, scheduling rotations, budgeting the labor costs, balancing cost and profit according to seasonality, surveying and hiring staff, and maintenance of the commercial kitchen equipment.

2.SYSTEM REQUIREMENTS

2.1 LITERATURE REVIEW

Sources of Information

We have been to many restaurants, to understand their process of maintaining database and the level of efficiency they have in their system and drawbacks of their existing systems. After visiting many such centres and stores we thought of developing an application which will overcome the drawbacks of the existing systems.

Primary sources:

- Web Sites
- Discussion with owners of many restaurants.
- Suggestions from friends.

Secondary sources:

- Reference Materials

2.2 EXISTING SYSTEM:

2.2.1 Description:

Many Restaurants stores and maintain their day to day transactions manually. But some of them are having automation system which is helping them to store the data. But such restaurants are storing the information about the orders and the customer information. They don't have facility to store the information of feedbacks and favorite orders of customers over some period of time.

Restaurants are having standalone applications so at one time, they have the facility of many screens or many operations which is happening at one time. So they are storing them and then at last, the restaurant managers will able to see the data of last day.

The software which restaurants are using is very costly and their maintenance which is very high.

The software which we tried and got the information was called “iMagic Restaurant Reservation”. This software which is free for download but the restaurant has to pay after some time.

- **iMagic Restaurant Reservation:**

iMagic Restaurant Reservation aims to save time by letting you manage reservations on a computer and, in the process, lets you keep track of regular customers. For such a high price tag, we weren't that impressed by its design and brief trial period; however, it gets the job done.

The user interface is pretty plain, but it's intuitive and easy to navigate even without the Help feature. Should you need it, it also contains helpful tutorials and tips for getting started and pinpointing any issues you might have. Command buttons for viewing and making reservations line the top of the window, along with buttons for viewing customer history and reports. We were able to jump right in and quickly create a new reservation. It was simply a matter of clicking the Make Reservation button and inputting all the required information, including name, time of reservation, and duration. There are also fields for entering the customer's address, phone, e-mail, and business information. It's not required, but will certainly come in handy for taking advantage of the customer history and report features. Our reservation immediately appeared in the program's main screen. Using the buttons at the bottom of the window, we were able to edit the entry, change the status (arrived, departed, seated, at bar, etc.), and even add a message.

Though we would have liked to have seen a longer trial period than the 10 uses offered, we were still able to get a grip on the program's functions and navigation. If you're looking for a way to enter and manage reservations, this is a good tool to have on-hand.

2.2.1 Drawbacks:

- This software is basically used only for reservation means table booking. So if we want to just order some food or store any feed backs then it wont be any helpful. At last the restaurants have to store by themselves which will became no use of software.
- The user interface of the application is also not that much attractive. It is made in .net so the restaurant must have to install any windows operating system to install this application.
- So from the restaurants point of view, they are able to store only one kind of information. There is no security feature also.
- If any of party order is cancelling at the last moment, it will make a variation in the already created records and also will causes the wastage of foods.
- Many of the systems will not store the budget details for a long time.
- This will creates lot of mistakes like misspellings, calculation problems, duplicate entries etc.
- It is difficult for Managers to supervise all the sides of restaurants like kitchen, floor, and counter simultaneously.
- There is no functionality to get the updated details at all time to the Owners and Managers from all the branches.

2.3 PROPOSED SYSTEM:

The proposed system helps in many ways. It helps to do billing very easily. Account maintenance also becomes easier. They can keep track of their purchases of inventories, staffs details, customer feedback, sales of foods, and account details etc. The software is provided with the facilities to find out the favorite food of the customers, and the seasonal foods, or customers to add or modify and delete their feedbacks and suggestions. It helps in managing data of different types of orders like party order, home delivery or the normal order. Managing data of daily customers, managing data of staffs, managing data of daily expenses. It eliminates the drawbacks of existing system and also includes some more features.

2.3.1 Advantages:

- RMS will stores all the records permanently and also gives the raises and deviations in the status of the restaurant in the markets.
- All the records of the current and ex-staffs will be stored in the database.
- All the shift timings of staffs, salary, extra working time and charge etc. are also updated daily.
- Managers can view the kitchen side, floor side and the counter side details simultaneously and also can get details and updates from various branches.
- Any cancelations of parties are also updated.
- All the expenses per day will updated in the database daily.
- Also keeps the record of food items prepared and the sales of food and also the record of balance food.
- Less use of manual work.
- Profits and losses will be updated every month and also shows the variations.
- All the food order details like the order types (normal, home delivery, party order etc.) are stored daily.

2.4 FUNCTIONAL SPECIFICATION

Restaurant Management is having many modules, which make the software more efficient and user friendly. The modules make the maintenance of the database easier.

Every module is divided on the basis of the senarios. The main three senarios are

- Kicthen
- Manager
- Floor

The different modules in this project are described below:

- Food Items
- Order.
- Staffs.
- Cancellations.
- Expenses
- Home delivery

Menu Module:

In this module the number of food, their prices, offers etc. like details will stores.

- Insert food item.
- Insert price.
- Delete food item.
- Delete price.
- Modify price.

Order Module:

In this module the type of order, the amount etc like details will stores.

- Insert Order.
- Insert type of order.
- Insert price.
- Delete order.
- Delete price.
- Modify price.

Staffs Module:

In this module the number of staffs, their names, salary details, shift timings etc will stores.

- Insert staff.
- Delete staff.

Cancelations Module:

In this module the canceled foods details are stores.

- Insert Canceled order.
- Delete canceled order.

Expenses Module:

In this module the expenses of the money are stores.

- Insert Expenses.
- Delete Expenses.
- Modify Expenses.

Home Delivery:

In this module, there will section which will store the data about home delivery orders.

- Add order.
- Add bill.
- Delete order.
- Delete bill.

2.5 SOFTWARE TOOL SPECIFICATION:

Net Beans 7:

Most developers recognize the NetBeans IDE as the original free Java IDE. It is that, and much more! The NetBeans IDE provides support for several languages (PHP, JavaFX, C/C++, JavaScript, etc.) and frameworks.

NetBeans is an open-source project dedicated to providing rock solid software development products (theNetBeans IDE and the NetBeans Platform) that address the needs of developers, users and the businesses that rely on NetBeans as a basis for their products.

The NetBeans IDE is written in Java and can run anywhere a compatible JVM is installed, including Windows, Mac OS, Linux, and Solaris. A JDK is required for Java development functionality, but is not required for development in other programming languages.

The NetBeans platform allows applications to be developed from a set of modular software components called *modules*. Applications based on the NetBeans platform (including the NetBeans IDE) can be extended by third party developers.

The NetBeans Platform is a reusable framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally-signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

- User interface management (e.g. menus and toolbars)
- User settings management
- Storage management (saving and loading any kind of data)
- Window management
- Wizard framework (supports step-by-step dialogs)
- NetBeans Visual Library
- Integrated Development Tools

Netbean IDE is a free, open-source, cross-platform IDE with built-in-support for Java Programming Language.

MySQL Workbench:

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, and much more. MySQL Workbench is available on Windows, Linux and Mac OS.

MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. It includes everything a data modeler needs for creating complex ER models, forward and reverse engineering, and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort.

MySQL Workbench delivers visual tools for creating, executing, and optimizing SQL queries. The SQL Editor provides color syntax highlighting, reuse of SQL snippets, and execution history of SQL. The Database Connections Panel enables developers to easily manage database connections. The Object Browser provides instant access to database schema and objects.

MySQL Workbench provides a visual console to easily administer MySQL environments and gain better visibility into databases. Developers and DBAs can use the visual tools for configuring servers, administering users, and viewing database health.

2.6 HARDWARE REQUIREMENT:

- Processor: 800MHz Intel Pentium III or equivalent or new
- Disk Space: 50 MB or more

2.7 SOFTWARE REQUIREMENT:

- Operating System: No particular OS required because Java is platform independent.
- Software: JRE, MySQL

3. DESIGN SPECIFICATION

3.1 MODULER DESIGN:

Resturant Management System, we have divided mainly in three modules.

1. Manager Side:
 - It includes stock controlling, scheduling rotations, labour costs, balancing cost and profit according to seasonality, surveying and hiring staff, and maintenance of the commercial kitchen equipment.

2. Kitchen Side
 - It will keep the records of orders. So chef's will able to see and give the orders to waiters.

3. Floor Side
 - There will be details about orders, details about waiter and billing facility will be there.

3.2 SYSTEM SPECIFICATION

3.2.1 DATAFLOW DIAGRAM:

0 Level :

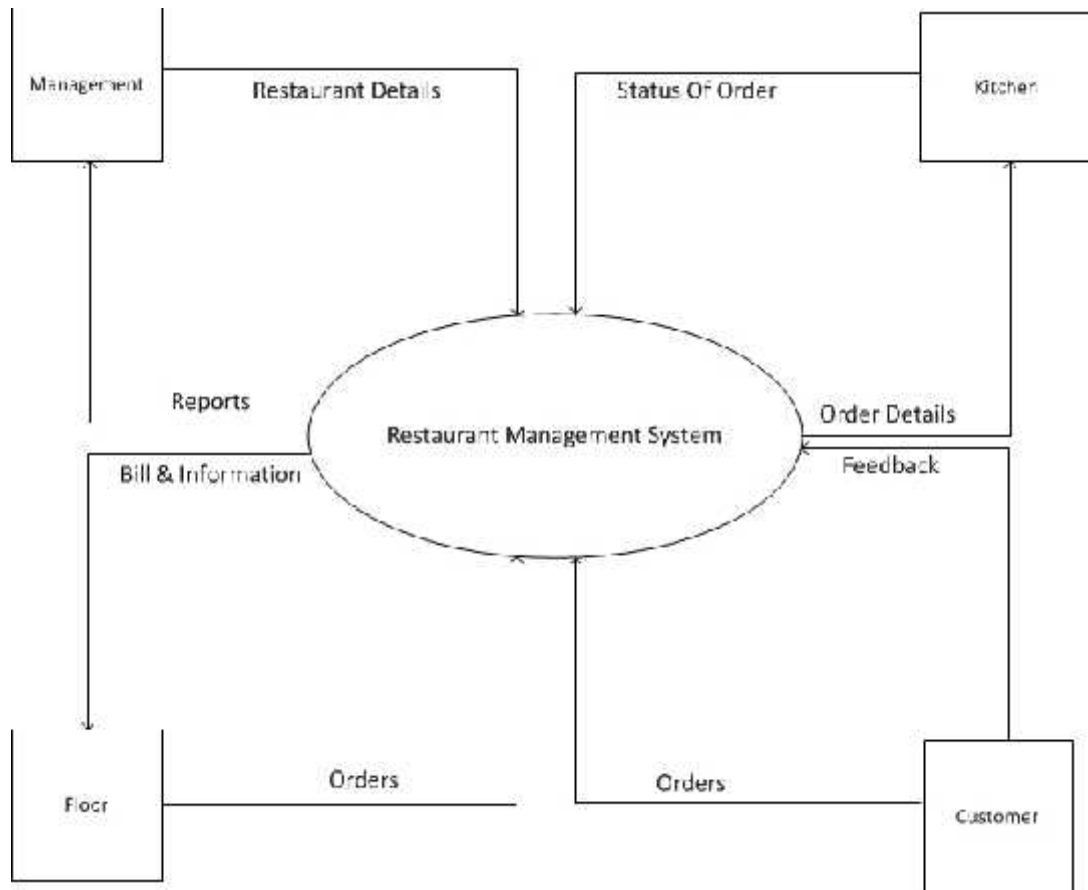


Fig 3.1

1 Level:

- Login Process:

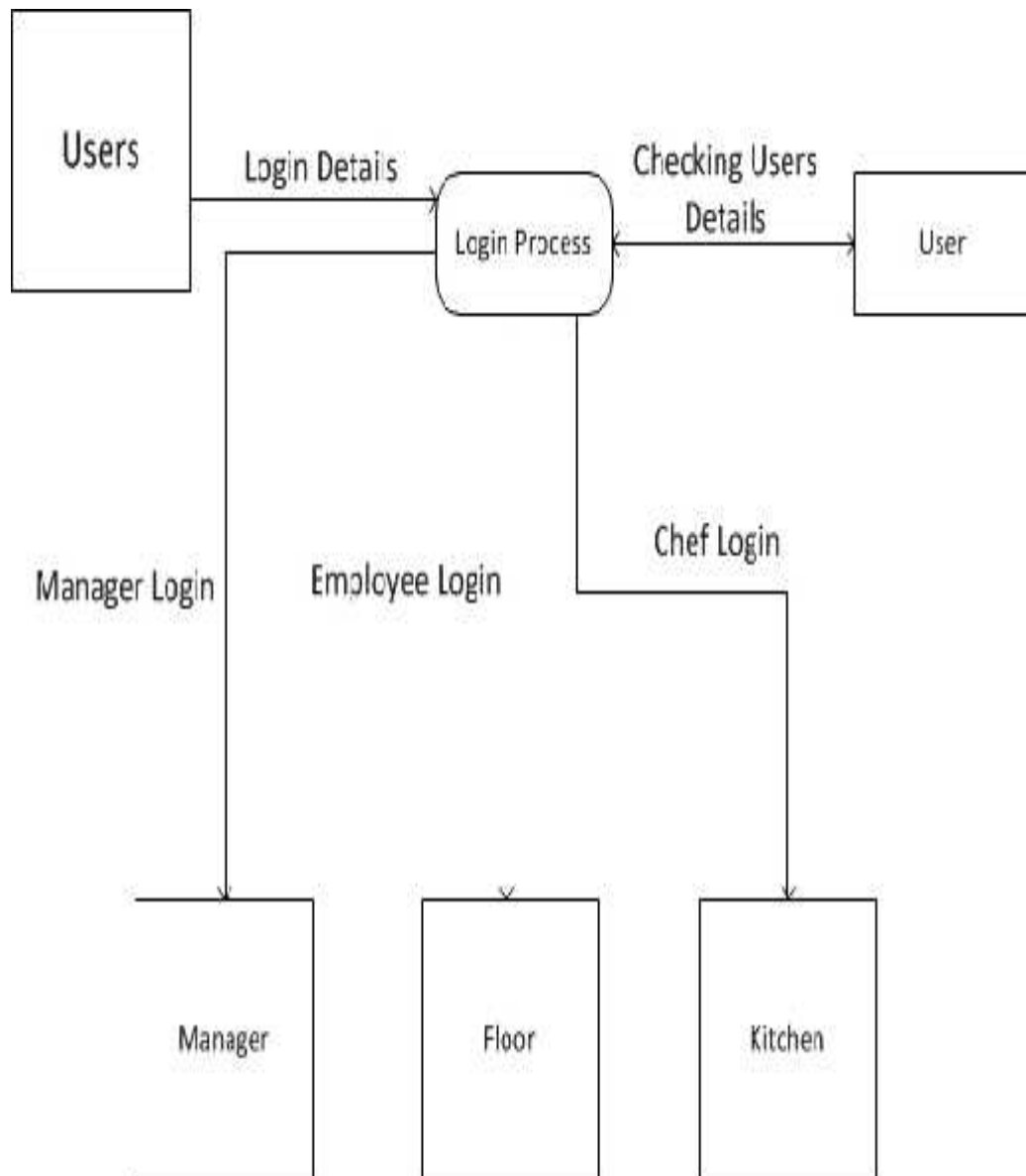


Fig 3.2

- Manager Side:

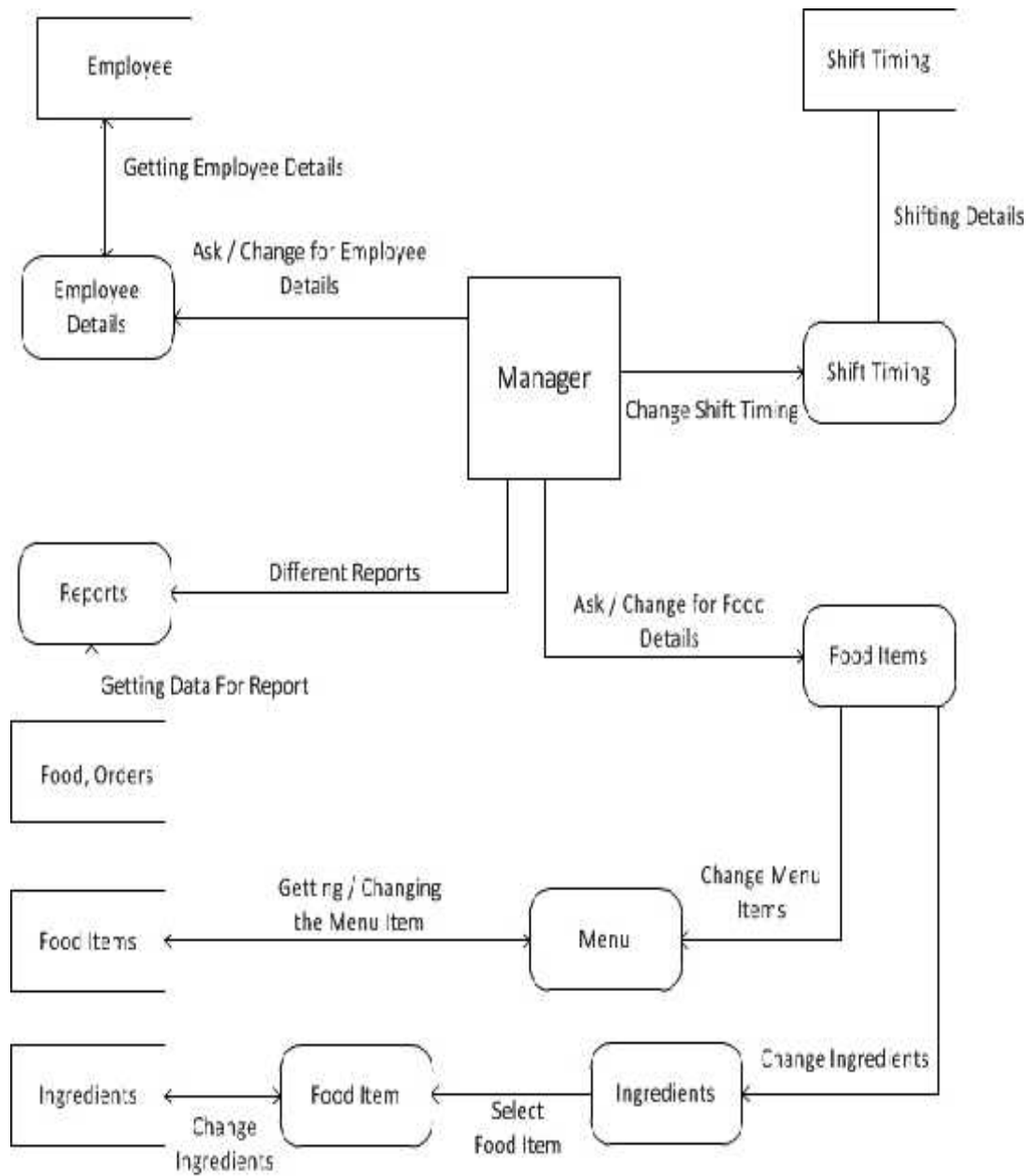


Fig 3.3

- Kitchen Side:

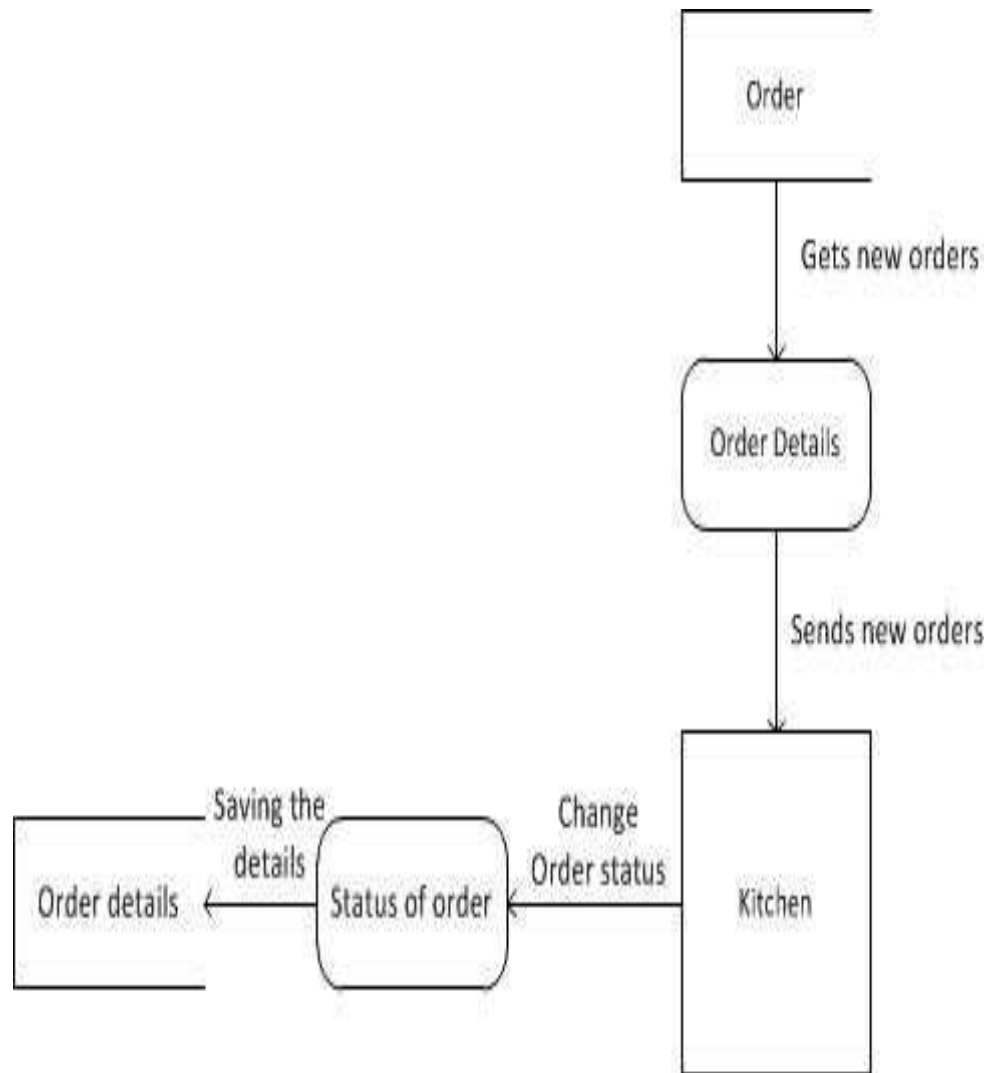


Fig 3.4

- Floor Side

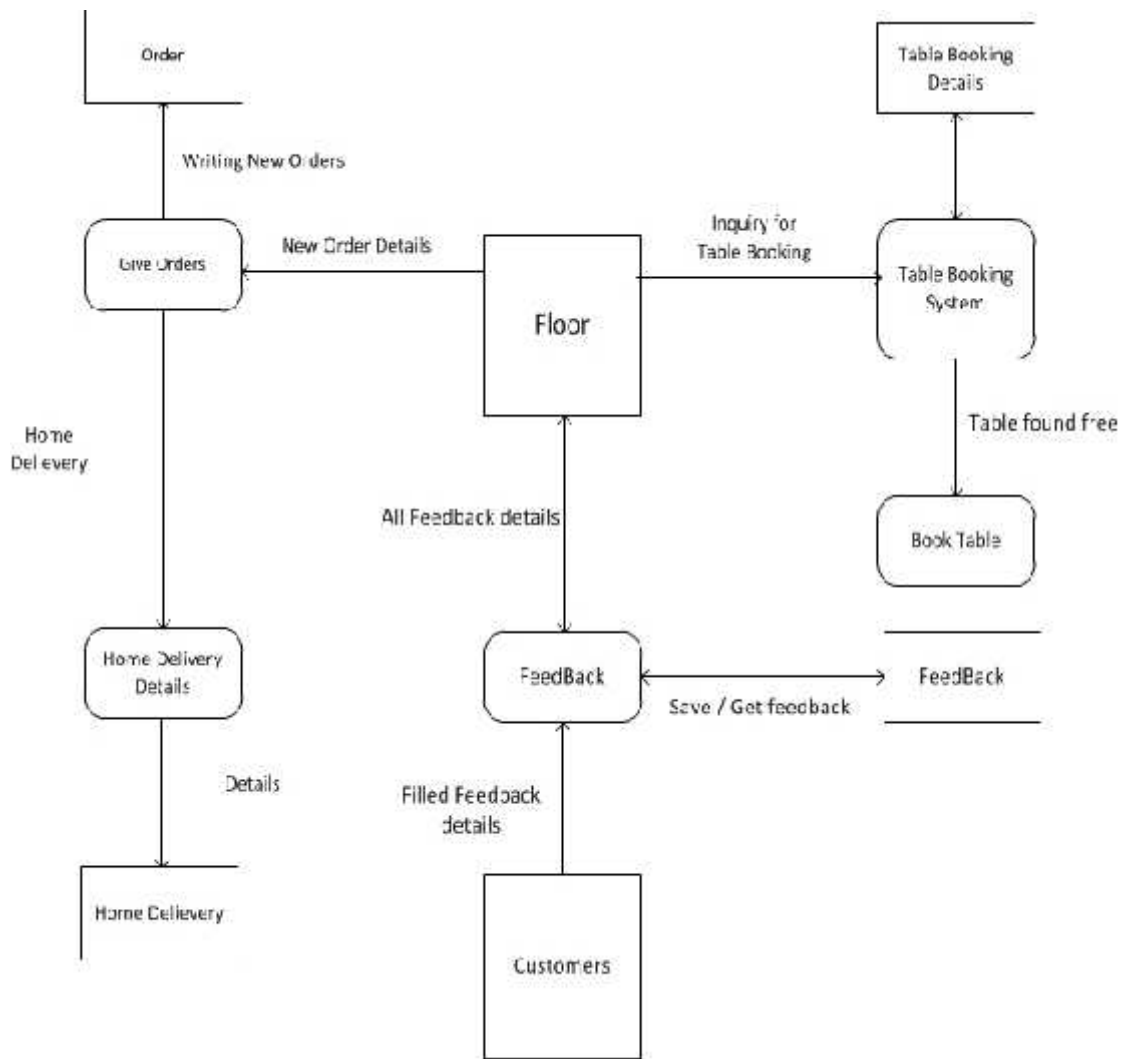


Fig 3.5

3.2.2 ER DIAGRAM

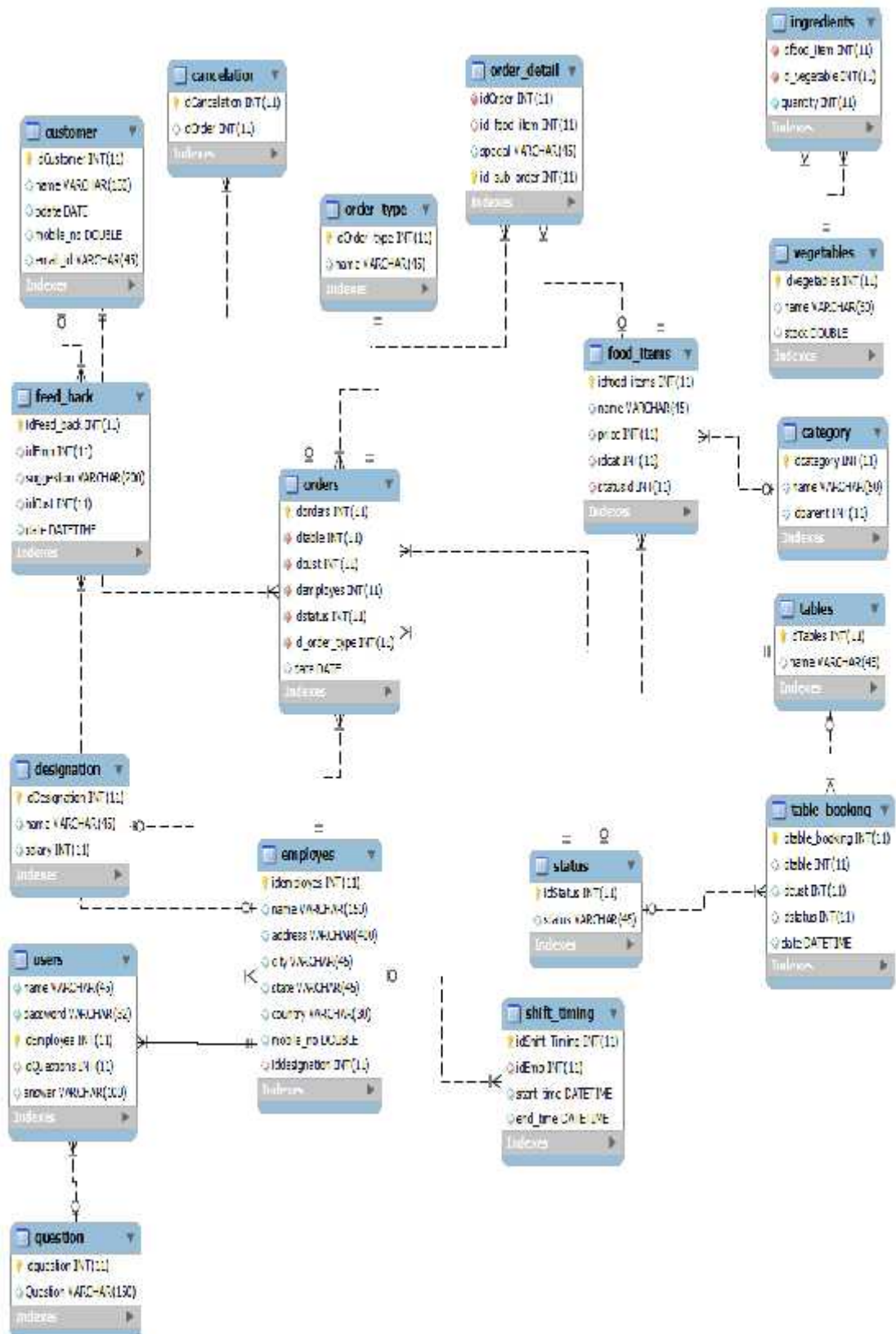


Fig3.6

3.2.3 DATA DICTIONARY

1. Users:

Field Name	Data Type	Size
Name	Varchar	20
Password	Varchar	32
idEmployee	INT	
idQuestion	INT	
Answer	VARCHAR	100

2. Cancellation:

Field Name	Data Type	Size
idCancellation	INT	
idOrder	INT	

3. Category:

Field Name	Data Type	Size
idCategory	INT	
Name	Varchar	32
idParent	INT	

4. Customer:

Field Name	Data Type	Size
idCustomer	INT	
Name	Varchar	150
Bdate	Date	
Mobile_no	Double	
Email_id	Varchar	45

5. Designation:

Field Name	Data Type	Size
idDesignation	INT	
Name	Varchar	45
Salary	INT	

6. Employee:

Field Name	Data Type	Size
idEmployee	INT	
Name	Varchar	150
Address	Varchar	400
City	Varchar	30
State	Varchar	30
Mobile_no	Double	
idDesignation	INT	

7. Feed_Back:

Field Name	Data Type	Size
idFeed_Back	INT	
Suggestion	Varchar	200
idEmployee	INT	
idCustomer	INT	

8. Food Items:

Field Name	Data Type	Size
idFood_item	INT	
Name	Varchar	45
Price	INT	
idCategory	INT	

9. Ingredients:

Field Name	Data Type	Size
idFood_Item	INT	
idVegetable	INT	
Quantity	INT	

10. Order Type:

Field Name	Data Type	Size
idOrder_type	INT	
Name	Varchar	45

11. Order:

Field Name	Data Type	Size
idOrder	INT	
idTable	INT	
idCustomer	INT	
idEmployee	INT	
idStatus	INT	
idOrder_type	INT	
Date	Date	

12. Order Details:

Field Name	Data Type	Size
idSub_Order	INT	
idOrder	INT	
idFood_item	INT	
Special	Varchar	45

13. Shift Timing:

Field Name	Data Type	Size
idShift_Time	INT	
idEmployee	INT	
Start_Time	DateTime	
End_Time	DateTime	

14. Status:

Field Name	Data Type	Size
idStatus	INT	
Status	Varchar	45

15. Tables:

Field Name	Data Type	Size
idTable	INT	
Name	Varchar	45

16. Table Booking

Field Name	Data Type	Size
idTable_booking	INT	
idTable	INT	
idCustomer	INT	
idStatus	INT	
Date	Date	

17. Questions:

Field Name	Data Type	Size
idQuestion	INT	
Questions	Varchar	100

18. Vegetables:

Field Name	Data Type	Size
idVegetable	INT	
Name	Varchar	45
Stock	Double	

3.2.4 DATA VALIDATIONS AND CONSTRAINTS:

1. **Required Data Fields:** There are many fields in many forms in which user have to put some data. If you just not give any entry or any text in some kind of fiels then it won't allow because of that, our data is stored very neatly and all the required fields of database won't be blank.
2. **No Digits allowed:** There are many fields in which user won't be allowed to store the digits in some fields. Like many forms having the field like names, in which user won't be allowed to store any digits in that given fields.
3. **No Alphabets allowed:** There are many fields in the application which user won't be able to store any kind of alphabets. Like in the mobile number field, user won't be allowed to store any kind of alphabets, their they just have to enter only numbers.
4. **User Validation:** There is validation for username, over here, we are just validating the username. If similar kind of user found out in the database, then application won't allow the same name of username.
5. **Password Validation:** Whenever user wants change the password then he/she must enter the password in between 5 to 15.
6. **Email Validation:** Whenver application will ask for any email, then he/she must have to enter proper email.
7. **Mobile number Validation:** Whenever user is entering the mobile number then he/she must have to enter proper mobile number like he/she must have to enter 10 digits and like it should be start with 0.

4. IMPLEMENTATION DETAILS

4.1 SAMPLE CODE:

- **Login Screen:**

This is sample code of login screen form, which will first start when user will start the application.

```
/*
 * Class      : Login
 * Description : For Login into database
 *
 */
package UI;

import DB.EmployeeDB;
import DB.UserDB;
import UTIL.Validate;
import java.awt.Color;
import java.awt.Image;
import java.awt.Toolkit;
import java.awt.event.KeyEvent;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;

/**
 *
 * @author Nitz
 */
public class Login extends javax.swing.JFrame {

    /** Creates new form Login */
}
```

```
public JFrame myparent;
public UserDB udb;
public String empid;
public EmployeeDB empdb;
int LoginMnemonic = KeyEvent.VK_L;
int ResetMnemonic = KeyEvent.VK_E;
int CancelMnemonic = KeyEvent.VK_R;
Color defaultColor;

public Login() {
    initComponents();
    java.net.URL url = ClassLoader.getResource("icons/main.jpg");
    Toolkit kit = Toolkit.getDefaultToolkit();
    Image img = kit.createImage(url);
    this.setIconImage(img);

    try {
        udb = new UserDB();
        empdb = new EmployeeDB();
    } catch (ClassNotFoundException ex) {
        JOptionPane.showMessageDialog(null, " Sorry problems with database \n " +
ex.getMessage(), "RMS: Exception in Login",
JOptionPane.INFORMATION_MESSAGE);
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, " Sorry problems with database \n " +
ex.getMessage(), "RMS: Exception in Login", JOptionPane.ERROR_MESSAGE);
    }
    setLocationRelativeTo(null);
    loginBT.setMnemonic(LoginMnemonic);
    resetBT.setMnemonic(ResetMnemonic);
    regBT.setMnemonic(CancelMnemonic);
    }

    public void setParent(JFrame myparent) {
```

```
        this.myparent = myparent;
    }

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

        jButton1 = new javax.swing.JButton();
        userLB = new javax.swing.JLabel();
        unameTF = new javax.swing.JTextField();
        passLB = new javax.swing.JLabel();
        loginBT = new javax.swing.JButton();
        resetBT = new javax.swing.JButton();
        forgetLB = new javax.swing.JLabel();
        errorLB = new javax.swing.JLabel();
        regBT = new javax.swing.JButton();
        passTF = new javax.swing.JPasswordField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("RMS");
        setIconImages(null);
        setName("Login"); // NOI18N
        setResizable(false);
        addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                formKeyPressed(evt);
            }
            public void keyTyped(java.awt.event.KeyEvent evt) {
                formKeyTyped(evt);
            }
        });

        jButton1.setFont(new java.awt.Font("Verdana", 0, 24));
```

```
jButton1.setText("Restaurant Management System");
jButton1.setToolTipText("Always at help !!!!!");

userLB.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
userLB.setText("Username:");

unameTF.setToolTipText("Username for Login");

passLB.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
passLB.setText("Password:");

loginBT.setFont(new java.awt.Font("Tahoma", 1, 11));
loginBT.setText("Login");
loginBT.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        loginBTActionPerformed(evt);
    }
});

resetBT.setFont(new java.awt.Font("Tahoma", 1, 11));
resetBT.setText("Reset");
resetBT.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        resetBTActionPerformed(evt);
    }
});

forgetLB.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
forgetLB.setText("Forget Password?");
forgetLB.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        forgetLBMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
```

```
        forgetLBMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        forgetLBMouseExited(evt);
    }
});
forgetLB.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusGained(java.awt.event.FocusEvent evt) {
        forgetLBFocusGained(evt);
    }
    public void focusLost(java.awt.event.FocusEvent evt) {
        forgetLBFocusLost(evt);
    }
});

errorLB.setFont(new java.awt.Font("Tahoma", 2, 14)); // NOI18N
errorLB.setForeground(new java.awt.Color(255, 0, 0));
errorLB.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        errorLBMouseClicked(evt);
    }
});

regBT.setFont(new java.awt.Font("Tahoma", 1, 11));
regBT.setText("Register");
regBT.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        regBTActionPerformed(evt);
    }
});

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



```
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jButton1)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())

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

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(passLB)
    .addComponent(userLB))
    .addGap(28, 28, 28)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(passTF)
    .addComponent(unameTF,
javax.swing.GroupLayout.DEFAULT_SIZE, 91, Short.MAX_VALUE)))
    .addGroup(layout.createSequentialGroup())
        .addContainerGap()
        .addComponent(forgetLB,
javax.swing.GroupLayout.PREFERRED_SIZE, 123,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 89,
Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
```

```
.addComponent(regBT, javax.swing.GroupLayout.Alignment.LEADING,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
.addComponent(resetBT,  
javax.swing.GroupLayout.Alignment.LEADING,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
.addComponent(loginBT,  
javax.swing.GroupLayout.Alignment.LEADING,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))  
.addGap(66, 66, 66))  
.addGroup(layout.createSequentialGroup()  
.addContainerGap()  
.addComponent(errorLB, javax.swing.GroupLayout.DEFAULT_SIZE, 200,  
Short.MAX_VALUE)  
.addGap(211, 211, 211))  
);  
layout.setVerticalGroup(  
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
.addGroup(layout.createSequentialGroup()  
.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE,  
73, javax.swing.GroupLayout.PREFERRED_SIZE)  
.addGap(22, 22, 22)  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASILI  
NE)  
.addComponent(unameTF,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
.addComponent(loginBT)  
.addComponent(userLB, javax.swing.GroupLayout.PREFERRED_SIZE,  
22, javax.swing.GroupLayout.PREFERRED_SIZE))
```

```

        .addGap(18, 18, 18)

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

            .addComponent(resetBT)
            .addComponent(passTF, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(passLB))

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        NG)

            .addGroup(layout.createSequentialGroup())
                .addGap(7, 7, 7)
                .addComponent(errorLB,
javax.swing.GroupLayout.PREFERRED_SIZE, 17,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(forgetLB))
            .addGroup(layout.createSequentialGroup())
                .addGap(18, 18, 18)
                .addComponent(regBT)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

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

private void forgetLBMouseClicked(java.awt.event.MouseEvent evt) { //GEN-
FIRST:event_forgetLBMouseClicked
    //Errors er = new F(this, rootPaneCheckingEnabled);
    Forget fg = new Forget(this, rootPaneCheckingEnabled);

```

```
this.show(false);
fg.show();
} //GEN-LAST:event_forgetLBMouseClicked

private void errorLBMouseClicked(java.awt.event.MouseEvent evt) { //GEN-
FIRST:event_errorLBMouseClicked
// TODO add your handling code here:
} //GEN-LAST:event_errorLBMouseClicked

private void loginBTActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_loginBTActionPerformed
    try {
        String error = validateDetails();
        if (!Validate.isEmpty(error)) {
            errorLB.setText(error);
        } else {
            empid = udb.getEmpid();
            int designation = empdb.getDesignation(empid);
            System.out.println("My" + designation);

            if (designation == 3) {
                this.setVisible(false);
                new Manager().setVisible(true);
            } else if (designation == 4) {
                this.setVisible(false);
                new Floor().setVisible(true);
            } else {
                }
            }
        } catch (Exception e) {
            }
    } //GEN-LAST:event_loginBTActionPerformed
```

```
private void resetBTActionPerformed(java.awt.event.ActionEvent evt) {//GEN-  
FIRST:event_resetBTActionPerformed  
    unameTF.setText("");  
    passTF.setText("");  
//GEN-LAST:event_resetBTActionPerformed
```

```
private void regBTActionPerformed(java.awt.event.ActionEvent evt) {//GEN-  
FIRST:event_regBTActionPerformed  
    New_User nw = new New_User();  
    this.setVisible(false);  
    nw.setVisible(true);  
//GEN-LAST:event_regBTActionPerformed
```

```
private void formKeyPressed(java.awt.event.KeyEvent evt) {//GEN-  
FIRST:event_formKeyPressed  
//GEN-LAST:event_formKeyPressed
```

```
private void formKeyTyped(java.awt.event.KeyEvent evt) {//GEN-  
FIRST:event_formKeyTyped  
//GEN-LAST:event_formKeyTyped
```

```
private void forgetLBFocusGained(java.awt.event.FocusEvent evt) {//GEN-  
FIRST:event_forgetLBFocusGained  
//GEN-LAST:event_forgetLBFocusGained
```

```
private void forgetLBFocusLost(java.awt.event.FocusEvent evt) {//GEN-  
FIRST:event_forgetLBFocusLost  
//GEN-LAST:event_forgetLBFocusLost
```

```
private void forgetLBMouseEntered(java.awt.event.MouseEvent evt) {//GEN-  
FIRST:event_forgetLBMouseEntered  
    forgetLB.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
    defaultColor = forgetLB.getForeground();  
    forgetLB.setForeground(new java.awt.Color(255, 0, 0));
```

```
//GEN-LAST:event_forgetLBMouseEntered

private void forgetLBMouseExited(java.awt.event.MouseEvent evt) //GEN-
FIRST:event_forgetLBMouseExited
    forgetLB.setFont(new java.awt.Font("Tahoma", 0, 14));
    forgetLB.setForeground(defaultColor);
//GEN-LAST:event_forgetLBMouseExited

public String validateDetails() throws ClassNotFoundException, SQLException {
    String error = "";
    System.out.println("Answer : " + udb.checkUser(unameTF.getText()));
    if (unameTF.getText().length() == 0) {
        error = "Username must be entered";
    } else if (passTF.getText().length() == 0) {
        error = "Password must be entered";
    } else if (udb.checkUser(unameTF.getText()) == null) {
        error = "Username not found";
    } else if (udb.checkDetails(unameTF.getText(), passTF.getText()) == null) {
        error = "Login details not matching";
    }
    return error;
}

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

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {
            try {
```

```
        UIManager.setLookAndFeel(new
org.pushingpixels.substance.api.skin.SubstanceRavenLookAndFeel());

        } catch (UnsupportedLookAndFeelException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
        }

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

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JLabel errorLB;
private javax.swing.JLabel forgetLB;
private javax.swing.JButton jButton1;
private javax.swing.JButton loginBT;
private javax.swing.JLabel passLB;
private javax.swing.JPasswordField passTF;
private javax.swing.JButton regBT;
private javax.swing.JButton resetBT;
private javax.swing.JTextField unameTF;
private javax.swing.JLabel userLB;
// End of variables declaration//GEN-END:variables
}
```

4.2 SCREEN SHOTS

- **Login Form:**



The screenshot shows a window titled "RMS" with a dark background. At the top, the text "Restaurant Management System" is displayed. Below this, there are three input fields: "Username:", "Password:", and "Forget Password?". To the right of the "Username:" field is a "Login" button. To the right of the "Password:" field is a "Reset" button. To the right of the "Forget Password?" field is a "Register" button.

Fig 4.1

- **Forgot Password:**



The screenshot shows a window titled "RMS : Foget Password" with a dark background. At the top, the text "Restaurant Management System" is displayed. Below this, the text "Forgot Password" is shown. There are four input fields: "What You Remember ?" with a dropdown menu showing "Username", "Employee ID" (disabled), "Username" (text input), and "Questions" with a dropdown menu showing "What town were you born in?". Below these fields are two buttons: "Change Password" and "Back".

Fig 4.2

- **Registration Form:**



The screenshot shows a registration window titled "Restaurant Management System". It contains the following fields and controls:

- Employee ID:
- Username:
- Password:
- Retype Password:
- Questions:
- Answer:
- Buttons: Register, Reset, Back

Fig 4.3

- **Floor Area [Main Screen]:**

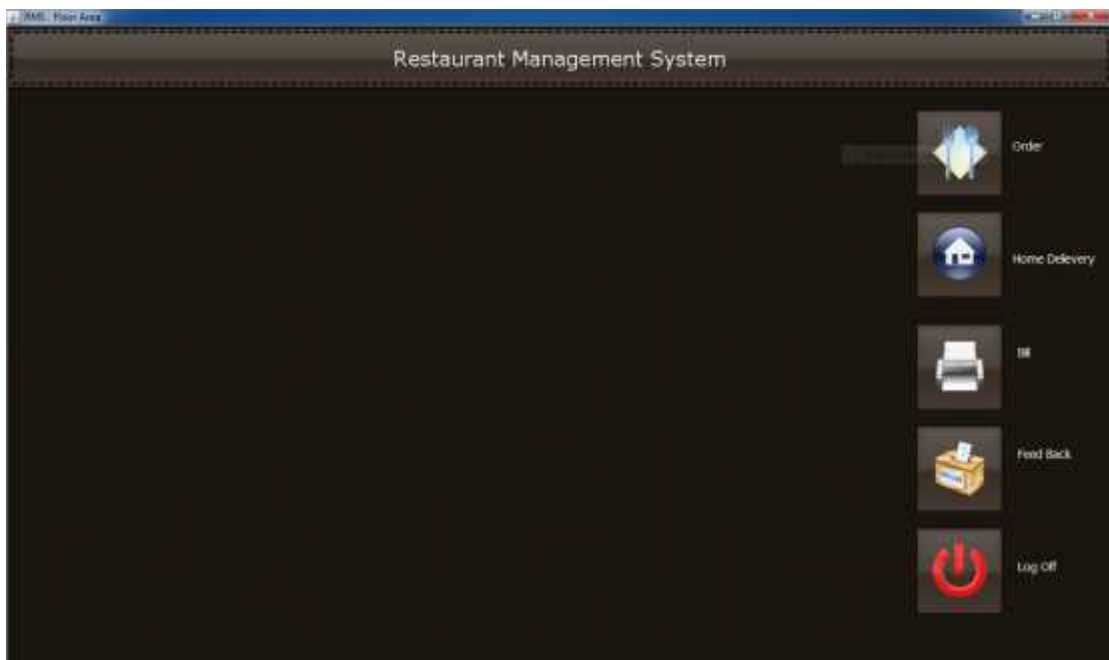


Fig 4.4

- **Manager Area [Main Screen]:**

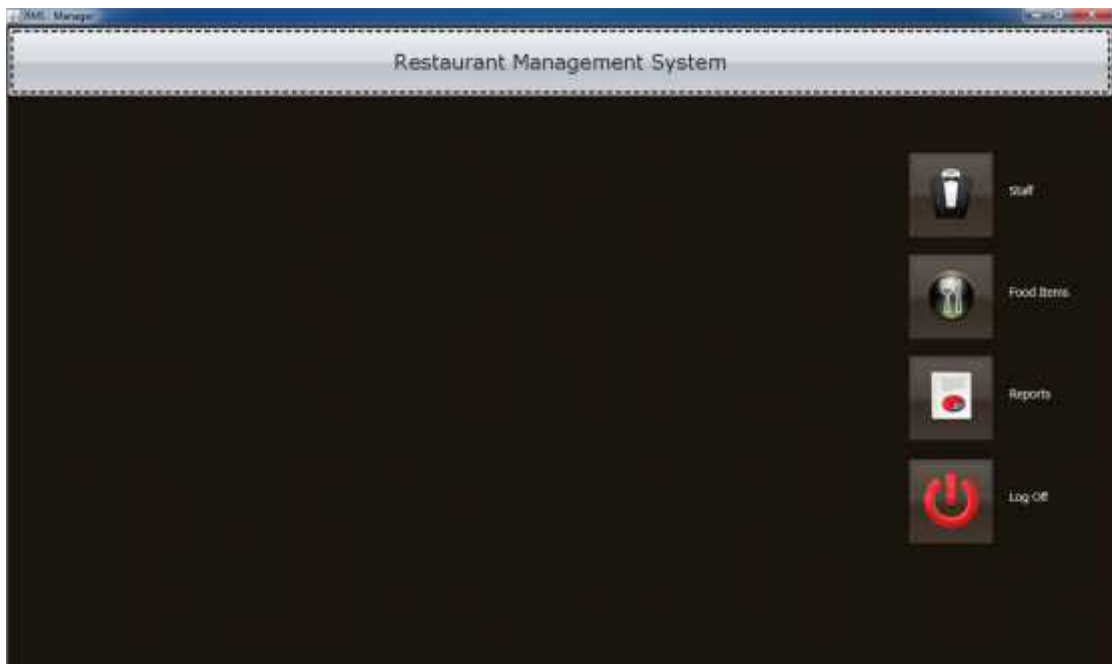


Fig 4.5

- **Employees Information [Manager Area]:**

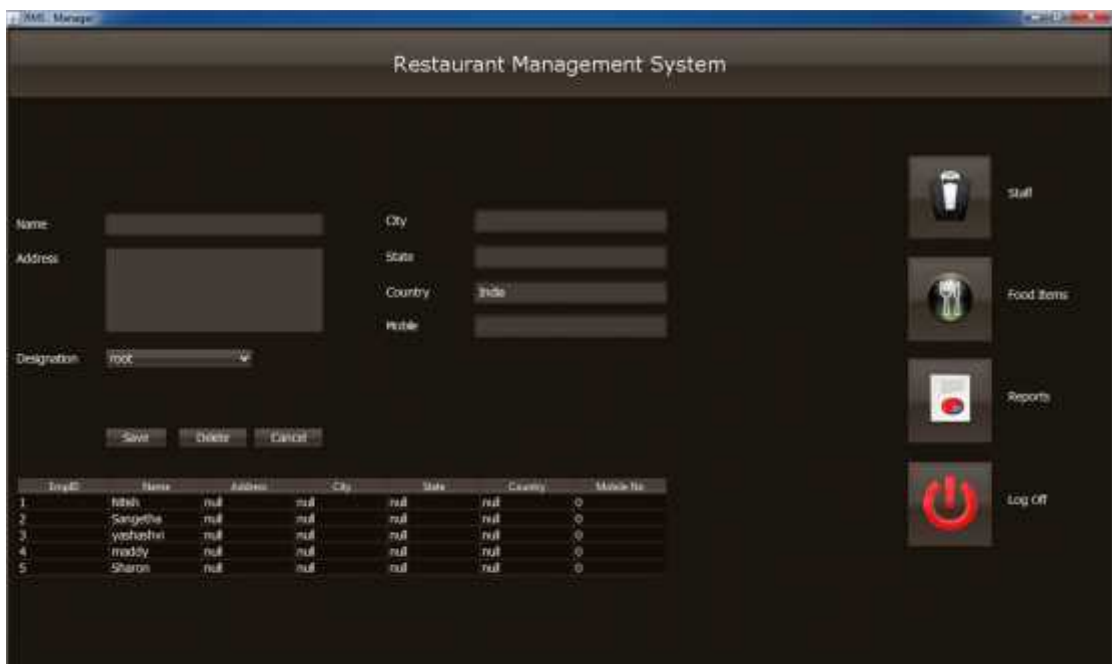


Fig 4.6

5. TESTING

5.1 TEST CASES:

1. JUnit:

JUnit is a simple, open source framework to write and run repeatable tests. It is an instance of the xUnit architecture for unit testing frameworks. JUnit features include:

- Assertions for testing expected results
- Test fixtures for sharing common test data
- Test runners for running tests

JUnit was originally written by Erich Gamma and Kent Beck.

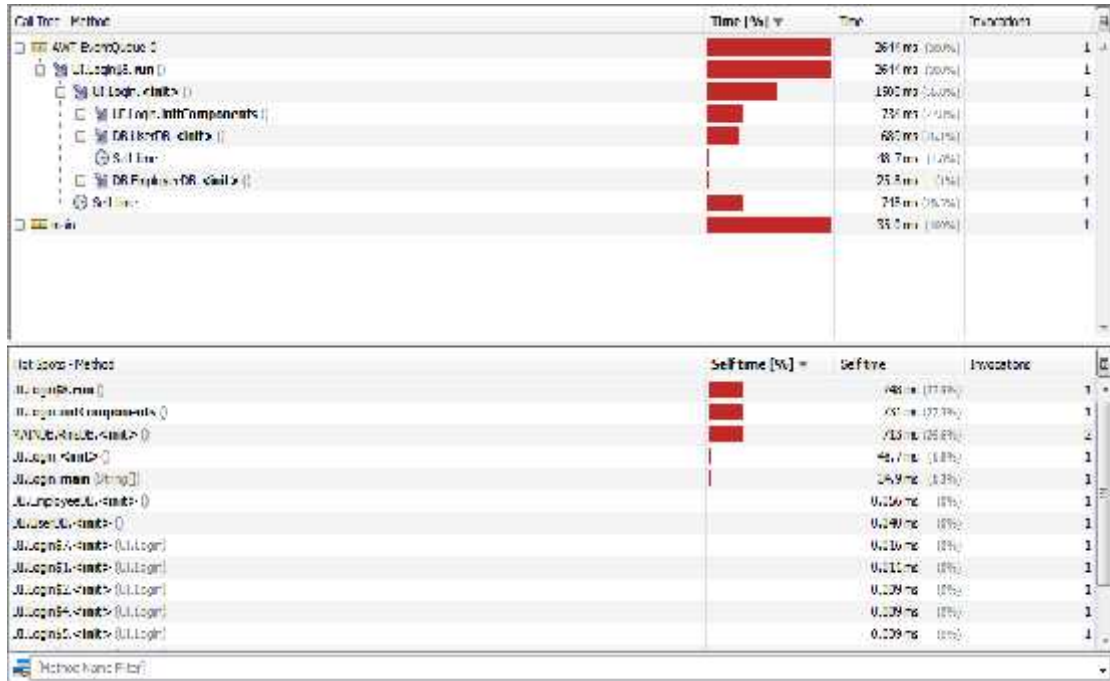
2. NetBeans profiler:

The NetBeans profiler provides expert assistance for optimizing your application's speed and memory usage, and makes it easier to build reliable and scalable Java SE, JavaFX and Java EE applications.

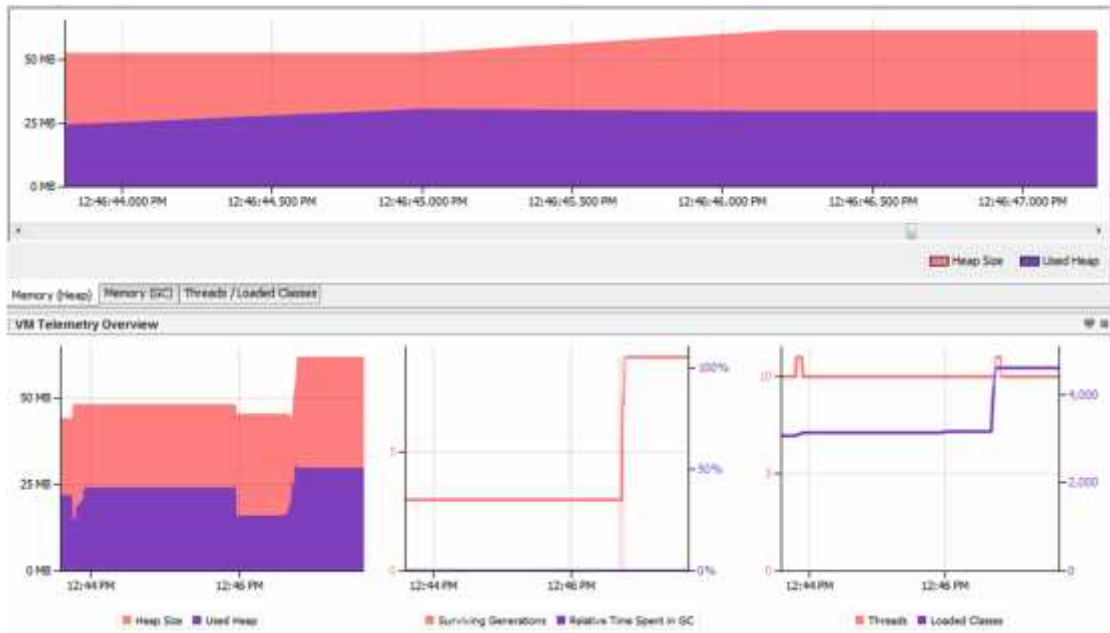
- Task-Based Profiling
- Heap Walker
- Profiling Points
- Identifying CPU Bottlenecks
- Tracking Memory Usage
- Monitoring Threads Status
- Remote Profiling
- Attach Profiler

5.2 TEST REPORTS

- **NetBeans profiler:**



- **Memory Heap:**



6. CONCLUSIONS

6.1 ADVANTAGES

- RMS makes works easy and faster than current applications.
- All the data will be saved in the database. So the administer can view all the data on time.
- This system reduces manual works.
- RMS will stores all the records permanently and also gives the raises and deviations in the status of the restaurant in the markets.
- All the records of the current and ex-staffs will be stored in the database.
- All the shift timings of staffs, salary, extra working time and charge etc. are also updated daily.
- Any cancelations of parties are also updated.
- All the expenses per day will updated in the database daily.
- Also keeps the record of food items prepared and the sales of food and also the record of balance food.
- All the food order details like the order types (normal, home delivery, party order etc.) are stored daily.

6.2 LIMITATIONS:

- Application won't be able to send some notification about any report or any delievery system to owner or customer.
- Application will required the java, mysql and three different system which should be connected to one network for using application to its full functionality.
- User won't able to change the skin or any functionality of application.
- User can only get the reports in some formats only.