SUJIT MAHARJAN

RESTAURANT BILLING SYSTEM

Technology and Communication
2017
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis supervisor Dr. Ghodrat Modhadampour who has helped and supported me during my thesis and also I would like to thank VAMK for giving me an opportunity to study here.
ABSTRACT

The main goal of this thesis was to develop a desktop based billing system for a restaurant. This desktop based application is designed to administer its users and customers. RBS is a billing system, made for the effective utilization of modern technology in the organization. It is an automated software that can handle a lot of information about the restaurant’s employees, order history, reservation data. It has the capability to process bills and gather information about its employees and billing history. It is designed for the sole purpose of efficiency, speed and accuracy.

This application allows the admin to view the detail history of day to day operation in terms of sales, add or remove the employee, add new menu, view the detail of the employee like name, address, salary and so on. This application was developed using Netbeans IDE 8.0.2 for the front end and MySQL database was used to store data.

Keywords Desktop Application, MySQL, Netbeans IDE, JAVA
LIST OF FIGURES AND TABLES

Figure 1: General Architecture of the system ............................................ 10
Figure 2: Use-case Admin ........................................................................... 12
Figure 3: Use-case Employee ..................................................................... 13
Figure 4: Class Diagram ............................................................................. 14
Figure 5: Admin Login ............................................................................... 15
Figure 6: Add User ....................................................................................... 16
Figure 7: Edit user ....................................................................................... 17
Figure 8: Delete user ................................................................................... 18
Figure 9: User Login .................................................................................... 19
Figure 10: User Password Change ............................................................... 20
Figure 11: Reservation ............................................................................... 21
Figure 12: Billing ......................................................................................... 22
Figure 13: ER Diagram ............................................................................... 24
Figure 14: Database tables of the application .............................................. 25
Figure 15: Login Interface ......................................................................... 27
Figure 16: Home Interface ......................................................................... 28
Figure 17: Manage menu ............................................................................ 29
Figure 18: Billing Interface ......................................................................... 30
Figure 19: Staff Information Panel ............................................................... 31
Figure 20: Adding staff .............................................................................. 32
Figure 21: Removing staff ......................................................................... 33
Figure 22: Staff Information Updated .......................................................... 34
Figure 23: Reservation Completed ............................................................... 35
Figure 24: Billing Output ............................................................................ 36
Figure 25: Bill Print Output ....................................................................... 37

Table 1: Testing for Desktop Application .................................................... 49
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBS</td>
<td>Restaurant Billing System</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>PHP</td>
<td>Personal Home Page</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>DBMS</td>
<td>Database Management System</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

The “Restaurant Billing System” or “RBS” is an application to automate the process of information recording and billing of a restaurant.

This desktop based application is designed to administer its users and customers. RBS is a billing system, made for the effective utilization of modern technology in the organization. It is an automated software that can handle a lot of information about the restaurant’s employees, order history, reservation data. It has the capability to process bills and gather information about its employees and billing history. It is designed for the sole purpose of efficiency, speed and accuracy.

Waterfall Model is used to develop this software where different requirements are identified at first, analyze the requirements then design the software using various tools and techniques and using the development language the software is developed. Different testing like unit testing and integration testing are done during the development process. 

1.1 BACKGROUND

A restaurant is a business which prepares and serves food and drinks to customers in exchange for money, either paid before the meal, after the meal, or with an open account. Meals are generally served and eaten on premises. Restaurants vary greatly in appearance and offerings, including a wide variety of cuisines and service models ranging from inexpensive fast food restaurants to high-priced luxury establishments.

Restaurants are one of the growing business in recent times. In Nepal, many restaurants are doing very well by providing quality service to its customers. It has to deal with a lot of information about its client and employees, all of which are kept
and recorded manually and physically in a drawer or a room. Due to this situation, finding any reliable data when required and in time is not possible.

1.2 Objective of Project

The main objective of this software is a computerized working environment. This system is made on the assumption that the organization is fully requires manual work for any task. This project will serve the following objectives:

- To add and maintain records of available menu.
- To add and maintain employee details.
- To add and maintain description of new menu.
- To provide view of transaction to the owner.
- To provide a convenient solution of billing pattern.
- To make an easy to use environment for users.
- Easy to use system.

1.3 Problem Definition

The billing process is done manually by manpower. It results in delayed time for the consumer and to the organization while the bill is being processed. So, there is a room for improvement here. A certain computer based billing system could aid the organization to utilize its resources better. Computerized Billing System provide capabilities for entering client, employee and payment information, building a record and managing other related data needs in the organization.

The currently used system contains the following problems which are listed below:

- Inability of modification of data
- Manual operator control
- Lots of paperwork
• Difficult to record information systematically
• Difficult to retrieve information in time
• Wastage of paper

Based on the given problems, management of the information and billing process can be more efficient with the help of RBS.

1.4 Project Overview

Restaurant Billing System is a computer based billing system with user friendly interface which automatically manages the billing process of the customer very easily taking only a short period of time. The system can large amount of data and also generates bill for the customer. Billing history, reservation information and staff information can also be obtained with the use of RBS.

It is an automated desktop based software which has a simple design and very easy to use also. This project’s main focus is on proper management of information regarding the staffs, billing and reservation records. It is also specialized in automatically processing the customer bills and discounts.

The proposed system either does not require paper work or very few paper works are required. All the data is fetched into the computer immediately and various bills can be generated through computers. Since all the data is kept in a database, no data of the organization can be destroyed. Moreover works become very easy because there is no need to keep data on papers.

1.5 Scopes and Limitations

The software has a lot of features and advantages over the paper based billing system. It has the following scopes:

• This project will help the employee in fast billing.
• The project will enable to see free reservation available.
• Quality and faster service can be given to the customers.
• Easy to maintain in future prospect.

• This project enable employees to maintain a great database of information regarding the billing and reservation.

The limitations of this system are as follows:

• It is a desktop based software. A wider reach of customers cannot be obtained.

• Insufficient time for development.

• Need further more improvements for high class restaurants.

• Employees/staffs should be trained at first to use this system.

1.6 Tools Used

Various tools have been used to develop this software. They are listed below:

Front-end:

• Programming language used: Java
• IDE used: Netbeans IDE 8.0.2
• System used: Microsoft Windows 8 64-bit or above.

Back-end:

• SQL Server used: MySQL Server
• PHP used: PHP version 5.6.11
• Exe4j : for creating java launcher without an installer

Other tools:

• Together: for making various diagrams.
• Microsoft Word 2013 for documentation.
2 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a system and its components. This subsection presents the identified functional requirements for RBS which are listed below.

- **Staff management** – The admin has the full power to manage (add, delete, update, view) the staffs’ records. RBS provides an easy view of the staff records from which the management of the staffs can be done.

- **Make order** – The staffs can record orders by simply selecting the category and type of the item or meal. The order will be added to database.

- **Billing** – The orders made can be billed as well. The staffs as well as the admin can make bills of the orders made by the customers easily.

- **Menu management** – The admin can manage the menu of the restaurant. He/she can add, delete, update or view the menu of the restaurant.

- **Reservation** – Table can also be reserved by the customers by phone. The customer will get a unique reservation number.

- **Order history** – History of the billing or transaction can be viewed easily.

- **To-do list** – Users can add tasks to be done as well as clear the task from the list after its completion.

2.1 Non-functional requirements

The following are the non-functional requirements of RBS:
- **Consistency** - This system provides consistency user interface design to the user. The designs of the screen are standardized and consistent that make the user feel comfortable to use it.

- **Convenience** - The system gives convenience to the user to perform different activities like making order, menu managing, staff management, etc.

- **Usability** – The RBS software is ready to use system. The users will feel easy to use the system without much technical expertise.

- **Security** – This system provides the password security access control to avoid unauthorized user to login to the system. The system also authenticates the staff level to access to some of the admin part.

- **Reliability** – The restaurant billing system provides effective method to maintain the back-end of the system. All the orders are managed by this system effectively.

- **Scalability** - It has the capacity to handle a growing amount of work and its potential to be enlarged in order to accommodate that growth.

### 2.2 Feasibility Study

The feasibility study is carried out to test whether the proposed system is worth being implemented. Feasibility study is a test of system proposed regarding its work ability, its impact on the organization to meet user needs and effective use of resources. It is usually carried out by a small number of people who are familiar with the information system techniques, understand the part of the business or organization that will be involved or affected by the project and are skilled in the system analysis and design process.
All projects are feasible if they have unlimited resources and infinite time. But the development of software is plagued by the scarcity of resources and difficult delivery rates. It is necessary and prudent to evaluate the feasibility of a project at the earliest possible time.

The key consideration involved in the feasibility study are:

1. Technical feasibility
2. Economic feasibility
3. Operational feasibility
4. Schedule feasibility

2.3 **Technical feasibility**

Technical feasibility canters on the existing computer system (hardware, software) and to what extent it can support the proposed system. The technologies used in this project are:

**Software:**

Front End

- Language used : Java

Back End

- SQL

Operating System:

- Windows XP and higher. Our system requires window operating system which is easily available.

**Hardware:**
• Intel based processor run computer system, which have keyboard and mouse as input devices.
• This has been decided for the case of availability and up-gradation.

2.4 Economic feasibility

This feasibility study is to determine the benefits and savings that are expected from a system and compare it with its costs. If a benefit outweighs costs, then the decision is made to design and implement the system. Otherwise further alterations are made in the proposed system. Some of the costs to be considered are:

• Manpower cost
• Hardware and software cost

2.5 Operational feasibility

People are inherently resistant to change and software or applications have been known to facilitate change. In this project, a technical team is required to configure the software. Technical background on Java, database is important.

For users/customers, a short seminar or demo can be given so that they will be familiar with the software. They do not require much technical expertise to use RBS software.
3 GENERAL ARCHITECTURE

The general architecture of the application is shown in the figure below.

Figure 1: General Architecture of the system
3.1 Unified Modeling language (UML)

Unified Modelling language (UML) is a standardized modelling language enabling developers to specify, visualize, construct and document artefacts of a software system. Thus, UML makes these artefacts scalable, secure and robust in execution. UML is an important aspect involved in object-oriented software development. It uses graphic notation to create visual models of software systems.

3.2 Use case diagram

A use case diagram is a graphical representation of the interaction among the elements of a system. A use case is a methodology used in a system analysis to identify, clarify and organize system requirements.
Figure 2: Admin Function
3.3 Class Diagram

A class diagram is an illustration of the relationship and source code dependencies among the classes in the Unified Modelling Language (UML). In this context, a class defines the methods and variables in an object, which is specific entity in a program or the unit code representing the entity. /7/
Reservation
+ addReservation()
+ releaseReservation()

Menu
Menu : List<item>
+ giveMenu()
+ display()

Order
- oid
+ addOrder()

Customer
- cId
- cName
- cOrder
+ giveOrder()

Restaurant
- reservation
- menu
- customer
+ addReservation()
+ releaseReservation()
+ generateBill()
+ addCustomer()

Item
- ItemId
- ItemName
- ItemPrice
+ getItemId()
+ setItem(itemId) : void
+ getItemName()
+ setItemName(ItemName) : void
+ getItemPrice()
+ setItemPrice(ItemPrice) : void

ItemCount
- Item
- inumberOfPlates
- itemName
+ getItem()
+ setItem(item) : void
+ getNumberOfPlates()
+ getNumberOfPlates(): void
+ getMemberName()
+ getNumberOfPlates(): void

Bill
- Order
- BillId
+ calculateTotal()

Figure 4: Class Diagram
3.4 **Sequence Diagram**

Sequence diagram is an user friendly representation of an use case diagram which provides the sequential steps used in making the application. It helps to understand the different process that happens inside the system. There are several sequence diagram in this project which are given below:

3.5 **Admin Login**

This sequence diagram shows the different steps how admin can log in, first admin input his / her username and password then it checks whether the entered values are right or wrong if the data are in database and are for admin login then it will open admin dashboard otherwise it will give an error message. Figure 5 shows the process.

![Admin Login Sequence Diagram](image)

**Figure 5:** Admin Login
3.6 Add User

This sequence diagram shows the different steps how admin can add a new user, after admin login to the application he/she clicks on the add user option which opens add new user page. Then admin can add new user by filling up the necessary details like (Name, Gender, Contact, Address, Salary, username, password) and click add employee button. Then the request is sent to the database, here the entered parameters are checked to make sure that the entered data are correct. If all the entered data are correct then new user is created in the database otherwise error message is displayed.

![Add User Diagram]

Figure 6: Add User
3.7 Edit User

This sequence diagram shows the different steps how admin can edit or update the user information, after admin login to the application he/she clicks on the staff option in the dashboard which opens edit user page. Then admin can make all the necessary changes to the staff information and click edit employee button. Then the request is sent to the database, here the entered parameters are checked to make sure that the entered data are correct. If all the entered data are correct then new user is created in the database otherwise error message is displayed.

Figure 7: Edit user
3.8 Delete User

This sequence diagram shows the different steps how admin can delete the user account, after admin login to the application he/she clicks on the staff option in the dashboard which opens delete user page. Then admin can simply click on the employee to be deleted and click on delete employee button. Then the request is sent to the database and returns the success message.

Figure 8: Delete user
3.9 User Login

This sequence diagram explains the various steps that are needed for the user to log in the application. First user enters his / her username and password, for the very first login they must enter user name and password assigned by the admin. Once the login details are entered and clicked on login button, the request goes on to validate user which checks in the database. If the entered details are found in the database, the user can view dashboard otherwise the application will show an error message.

Figure 9: User Login
3.10 User Password Change

This sequence diagram shows the step by step process how user can change their password. First user log in with their old password once they are logged in they can click change password option from menu and enter their old password and new password they want to update and click on Ok button.

![User Password Change Diagram]

**Figure 10:** User Password Change
3.11 Reservation

This sequence diagram show how the reservations are made. First the booking request is made by the customer then employee checks if there is any free table if there is one then booking is confirmed if not then the customer waits in line. Figure 11 below shows this process.

Figure 11: Reservation
3.12 Billing

This sequence diagram show step by step process of how billing is done once the customer is done with the meal, he/she ask for the bill and staff first click on the billing page and enters all the item's the customer had then the application shows the total amount include service tax and VAT and the user click on print bill option which prints the bill and all the items ordered by the customer are stored in the database. Figure 11 shows the process in detail.

![Billing Diagram]

Figure 12: Billing
4 DATA MODELING

Data modelling is often the first step in database design and object-oriented programming as the designers first create a conceptual model of how data items relate to each other. Data modelling involves a progression from conceptual model to logical model to physical schema.

4.1 Database

In this we design and configure the database for the applications. Since the application is desktop application, we setup database in the local host server.

4.2 Design of the Database

The MySQL database is used to design the database. The database name is rbs it is in the local computer.

4.3 ER Diagram

An entity relationship diagram (E-R diagram) is a graphical representation of an information system that shows the relationship between people, objects, concepts or events within that system. An E-R diagram is a data modeling technique that helps define business process and can be used as the foundation of the relational database. 

/9/
Figure 13: ER Diagram

The ER diagram describes the relationship between the different tables in the database. Figure 13 shows that the application database contains six main database tables which are:

**Admin:** This table is only for admin, admin username and password are in this table.

**Employee:** This table consists of details of all the employees like name, address, contact info, salary.

**Category:** This table consists of all the different type of food category available.

**Menu:** This table consists of all the menu available.
Reservation: This table shows all the tables that are reserved for now and also for the future.

Billing history: This table consists of bill achieve, admin can view this table to know how the business have been progressing so far.

![Figure 14: Database tables of the application](image)
5 CODING TOOLS

There are various tools available for the development of a project. Our Restaurant Billing System software has been developed using the front end java and back end MySQL. Following tools are used for the RBS project.

- JAVA Programming Language

Java is general purpose programming language that is class-based, object-oriented specifically designed to have few complexity as possible.

- Netbeans IDE

Netbeans is the tool we have used to implement the code and design the user interface for the Restaurant Billing System.

- Together

Together is a diagramming and vector graphics application that has been used to design system models like, Use-Case Diagram, Sequence Diagrams and so on.

- Exe4j

exe4j is a Windows launcher generator, that is included for free when you purchase install4j. exe4j is useful if you want to create a Java launcher without an installer.

5.1 User Interface

This part provides a detail description of the user interface for the application. There are two user interfaces.

Admin's Interface and

User's Interface
5.2 Login Page

The login page allows the manager to log in to a dashboard/home page using username and password. The manager can log in using the user name and password created while developing this application and once he/she successfully log in then he/she can create other user's as well as admin users.

The log in page contains two input text fields username and password, two radio field to specify either admin or local user and login button and cancel button. Figure 15 shows the login interface.

![Login Interface](image)

**Figure 15**: Login Interface
5.3 Dashboard

After the successful login user can see the welcome message and his/her user name on the top right side of the screen and on the left side there are various clickable options from where user can perform various tasks and also you can see "To do" form there you can add any information or activities you perform or want someone else to perform. Figure 16 shows the dashboard interface.

![Dashboard Interface]

**Figure 16**: Dashboard Interface
5.4 Manage menu Page

In this page admin can add, remove and edit the items in the menu. Here admin can add food category, food name and its price. For instance

category: Soft Drink

Item type: Coca cola

Item Price: Rs 100

Figure 17 shows the manage menu.

Figure 17: Manage menu
5.5 Billing Interface Page

Here user can add all the items ordered by the customers to prepare the bill for that particular customer. Here user can see the sub total once added all the ordered items and application automatically calculates the grand total including 2% service tax and 13% VAT. And also application calculates the amount to be returned if there is any once the customer pays the bill. The billing interface in given in the figure 18.

Figure 18: Billing Interface
5.6 Staff Information Page

This page is only accessible by the admin. In this page admin can see all the details about the staffs such as name, gender, contact info, salary and also their username and password used for the log in. Admin can also add and remove the user from this page. Figure 19 shows the staff information interface.

![Staff Information Page](image)

**Figure 19: Staff Information Page**
6 RESULT

Commenting on the result obtained by Restaurant Billing System, the system is able to add the information of the staff, make reservation for the customer, billing, etc. Interpreting the result, the staff information can be added, edited, updated and deleted. Reservation can be made and cancelled and the bill can be printed. The outputs obtained by the system are shown in the figure below:

6.1 Adding a new staff Page

In the staff information page, admin can add a new staff once he/she entered all the necessary data then press add employee button to add a new employee. Figure 20 shows the result of adding employee command.

![Figure 20: Adding staff](image-url)
6.2 Removing a staff Page

In the same staff information page, admin can simply click on the name of the staff and press delete employee button to remove the staff information. Figure 21 shows the result of removing the employee command.

![Removing staff](image)

**Figure 21**: Removing staff
6.3 Updating staff information Page

In the same staff information page, admin can update the detail of the employee by simply choosing staff name and clicking update employee button. Figure 22 shows the result of updating the employee command.

Figure 22: Staff Information Updated
6.4 Reservation Page

In this page user can add the reservation time as per the customer's wish. Here customers details like name, contact number and address are entered and also the number of guests they are coming with and the date and time then the reservation is made. Figure 23 shows the output of reservation page once all the details are added.

![Reservation Page](image)

**Figure 23**: Reservation Completed
6.5 Billing Output Page

In this page once you provide all the necessary data then you will get following output as shown in the figure 24.

![Billing Output](image)

**Figure 24**: Billing Output
6.6 Bill Print Output Page

Once all the necessary data in the billing page are entered and click on the print bill option then it will print the bill on the paper. Figure 25 shows the format of the bill printed on the paper.

![Figure 25: Bill Print Output](image-url)

Figure 25: Bill Print Output
7 IMPLEMENTATION

This part consists of description of the implementation of the graphical user interfaces which is built for the application. This portion shows how the codes are used to acquire the desired goal of the project. Java is used for the backend of the application and MySQL is used to store database of the application. The following code snippets shows the implementation of the application.

7.1 Admin Login Function

The admin enter the data to the variables that are assigned and click on login and the data entered are checked in the database if they exist then the dashboard or home page is displayed otherwise an error message is shown.

```java
if(cf.checkLogin(username, password, type)){
    USERNAME=username;
    dispose();
    Dashboard db = new Dashboard();
    db.adminActionPerformed();
    Toolkit theKit = getToolkit();
    Dimension dim = theKit.getScreenSize();
    db.setSize(dim);
    db.setResizable(false);
    db.setVisible(true);
} else {
    JOptionPane.showMessageDialog(null, "Wrong username or password or user type. Please try again.");
}
```

Code Snippet 1: Login Page
7.2 Admin Dashboard Function

After the successful login of admin, the dashboard is displayed. There are several pages in the dashboard and different classes are used to switch between the pages.

When button is clicked a new form is created and set visible (true) will display a add category form, username given by user matched in keyword "admin" then only dashboard is visible or displayed.

```java
private void addcategoryItemMenuActionPerformed(java.awt.event.ActionEvent evt) {
    AddCategory ac=new AddCategory();
    ac.setVisible(true);
    ac.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    ac.setLocationRelativeTo(null);
}
```

```java
private void overPanelMouseClicked(java.awt.event.MouseEvent evt) {
    String empty=nameLabel.getText();
    if(empty.equalsIgnoreCase("admin")){
        frame.setContentPane(new DashboardItem());
    }else{
        frame.setContentPane(new DashboardItem());
    }
}
```

**Code Snippet 2: Dashboard**
7.3 Add User Function

Class method is used to create a new user

```java
private void addBttnActionPerformed(java.awt.event.ActionEvent evt) {
    EmployeeDTO employee = new EmployeeDTO();
    employee.setName(nameTxt.getText());
    employee.setGender(gender);
    employee.setContact(contactTxt.getText());
    employee.setAddress(addressTxt.getText());
    employee.setSalary(salaryTxt.getText());
    employee.setUsername(usernameTxt.getText());
    String password = new String(passwordTxt.getPassword());
    new EmployeeDAO().addEmployee(employee);
    loadDatas();
    clearFields();
}
```

```java
private void clearFields() {
    idTxt.setText(
    nameTxt.setText(
    buttonGroup1.clearSelection();
    contactTxt.setText(
    addressTxt.setText(
    salaryTxt.setText(
    usernameTxt.setText(
    passwordTxt.setText(

    Code Snippet 3: Text Field
```
Once the user inputs all the necessary data and click on add employee button, the query is send to database.

```java
try {
    Connection con=new ConnectionFactory().getConnection();

    String query="UPDATE login SET password=?
                 WHERE id=?";
    PreparedStatement pstmt;
    pstmt=con.prepareStatement(query);
    String password=new String(passwordTxt.getPassword());
    int id=Integer.parseInt(idTxt.getText());
    int dec_id=id+1;

    pstmt.setString(1,password);
    pstmt.setInt(2,dec_id);
    pstmt.executeUpdate();
    System.out.println("a");
} catch (Exception e) {
}
```

**Code Snippet 4: Add new user**

### 7.4 Add Category Function

Admin can add category simply by add category button. When he/she click that button with the help of following functions, new category is added in the database.
private void jButton1ActionPerformed (java.awt.event.ActionEvent evt) {
    try {
        Connection con=new ConnectionFactory().getConnection();
        String query="INSERT INTO category (cat_name) VALUES (?)";
        PreparedStatement pstmt = (PreparedStatement) con.prepareStatement(query);
        pstmt.setString(1,addcategoryField.getText());
        if(pstmt.executeUpdate()==1){
            JOptionPane.showMessageDialog(null, "Category Added");
            addcategoryField.setText("");
        }
    } catch (Exception e) { }
}

**Code Snippet 5: Add Category**

### 7.5 Add Menu Function

Admin can add menu under the category by simply clicking on manage menu option in the tool bar. When add menu button is clicked under manage menu, following action takes place
public void addItem(MenuDTO menu) {
    try {
        String sql = "INSERT INTO menu(category, item_name, item_price) VALUES(?,?,?)";

        pstmt = (PreparedStatement) con.prepareStatement(sql);
        pstmt.setString(1, menu.getCategory_name());
        pstmt.setString(2, menu.getItem_name());
        pstmt.setInt(3, menu.getItem_price());

        if (pstmt.executeUpdate() == 1) {
            JOptionPane.showMessageDialog(null, "One Record Inserted");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

**Code Snippet 6: Add Menu**

### 7.6 Update Menu Function

Menu can be updated simply by clicking on edit button of the manage menu page, first choose the item to be updated and click on edit button. And after that you can make necessary changes and save it. It will update the data in the database.
```java
public void updateMenu(MenuDTO menuUpdate) {
    try {
        String sql = "UPDATE menu SET 
        category=?,item_name=?,item_price=? WHERE item_id=?";
        pstmt = (PreparedStatement) con.prepareCall(sql);
        pstmt.setString(1, menuUpdate.getCategory_name());
        pstmt.setString(2, menuUpdate.getItem_name());
        pstmt.setInt(3, menuUpdate.getItem_price());
        pstmt.setInt(4, menuUpdate.getItem_id());

        if (pstmt.executeUpdate() == 1) {
            JOptionPane.showMessageDialog(null, "Menu Update Successful");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

**Code Snippet 7: Edit Menu**

### 7.7 Delete Employee Function

When the staff button is clicked in the dashboard, list of all the staffs are displayed. In order to delete the staff who is no longer working, simply click on the name of the staff and click delete employee button. It will send the query to database and will be deleted from the record.

```java
int resulte = new EmployeeDAO().deleteFromEmployee(String.valueOf(table.getValueAt(table.getSelectedRow(), 0)));
if (resulte == 1) {
    loadDatas();
    JOptionPane.showMessageDialog(null, "One Record Deleted.");
}
```

**Code Snippet 8: Delete Employee**
7.8 Change Password Function

Change password command is used to change the password, Public class Change password command is used to change the password.

private void 
jbButton1ActionPerformed(java.awt.event.ActionEvent 
evt) {
    int result = 0;
    String types = null;

    String currentPassword = new
String(oldPasswordField.getPassword());
    String newPassword = new
String(newpasswordField.getPassword());
    ConnectionFactory cdb = new
ConnectionFactory();
    try {
        if (cdb.checkLoginToChangePassword(username, currentPassword, types)) {
            result = cdb.chang
                    ePassword(newPassword, types);
            if (result == 1) {
                db.changePasswordEmp(newPassword);
                JOptionPane.showMessageDialog(null, "Password
Changed...Again Login!!!");
                dispose();
                flag = true;
                new Dashboard();
            } else {
                JOptionPane.showMessageDialog(null, "Password Not
Changed...Error!!!");
            }
        } else {
            JOptionPane.showMessageDialog(null, "current password mismatch");
            oldPasswordField.requestFocus();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    Code Snippet 9: Change password
7.9 Reservation Function

Reservation can be made by telephone. When someone wants to make a reservation, staff ask for the date and time, customers name and address and also table for how many people.

```java
private void reserveBtttnActionPerformed(java.awt.event.ActionEvent evt) {
    new ReservationPanel();
    ReservationDTO reserve = new ReservationDTO();
    reserve.setTokenNo(tokenField.getText());
    reserve.setCustomerName(customernameField.getText());
    reserve.setCustomerAddress(customeraddressField.getText());
    reserve.setContact(contactField.getText());
    String date = date1Field.getDate().toString();
    reserve.setDate(date);
    reserve.setTime((String) combo.getSelectedItem());
    String dayCombo = (String) dayCombobox.getSelectedItem();
    reserve.setDay(dayCombo);
    new ReservationDAO().addReservation(reserve);
    loadDatas();
    clearFields();
    RefName();
}
```

**Code Snippet 10: Make Reservation**
7.10 Billing Function

Once the customers are done with their meal, they ask for the bill. All the items they have ordered are entered in the application according to the category of the food and it will automatically generate the total amount including tax and VAT.

```java
try {
    Connection con = new ConnectionFactory().getConnection();
    String sql = "SELECT * FROM category";
    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
        String a = rs.getString("cat_name");
        categoryComboBox.addItem(a);
    }
} catch (Exception e) {
}
```

**Code Snippet 11: Selecting Item From Category**

```java
try {
    Connection con = new ConnectionFactory().getConnection();
    Statement stmt = con.createStatement();
    String query = "SELECT amount FROM demo";
    ResultSet rs = stmt.executeQuery(query);
    int b = 0;
    while (rs.next()) {
        int total = rs.getInt("amount");
        b = total + b;
    }
}
```

**Code Snippet 12: Amount Before VAT and TAX**
subtotalField.setText(Integer.toString(b));
    int subtotal =
parsingInt(subtotalField.getText());

    float vat = (float) (subtotal * 0.13);
    vatField.setText(Float.toString(vat));

    float charge = (float) (subtotal * 0.02);

    servicechargeField.setText(Float.toString(charge));

    float discount = 0;
    if (subtotal < 1000) {
        discountField.setText("0.00");
    } else if (subtotal >= 1000 && subtotal <=
1500) {
        discount = (float) (subtotal * 0.5);

    discountField.setText(Float.toString(discount));
    } else {

        discount = (float) (subtotal * 0.10);

    discountField.setText(Float.toString(discount));

    }

    float total;
    total = subtotal + vat + charge - discount;
    totalField.setText(Float.toString(total));

    float paid;
paid=Float.parseFloat(paidField.getText());

    float returns=total-paid;
    returnField.setText(Float.toString(returns))

} catch (Exception e) {

}
8 TESTING

In order to find any errors or the problems in the application, testing is carried out. Software Testing is the validation and certification process where the software fulfils the requirements of the users and project specifications.

Since this is a desktop application testing is done in only one environment i.e. on the Windows 10 environment. Table below consists of various tests performed during the development of this project.

**Table 1:** Testing for Desktop Application

<table>
<thead>
<tr>
<th>No</th>
<th>Test Performed</th>
<th>Description</th>
<th>Result</th>
<th>Corrections and Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Application file</td>
<td>Application file was launched</td>
<td>Login page opens</td>
<td>OK</td>
</tr>
<tr>
<td>2</td>
<td>Login page</td>
<td>Text field and radio button</td>
<td>This page contains text filed for user-name and password and radio field for selecting either admin or user and login and cancel button.</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>and password</td>
<td>word</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>5</strong></td>
<td><strong>Manage menu</strong></td>
<td>In the home page manage menu was click under menu icon.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Entered different categories and items under it.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6</strong></td>
<td><strong>Adding staffs</strong></td>
<td>On the staff page tried to add staff with entering any data.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>An error message was displayed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>7</strong></td>
<td><strong>Adding staffs</strong></td>
<td>On the staff page entered all the data and click add employee.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>New employee added.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>8</strong></td>
<td><strong>Removing staff</strong></td>
<td>On the staff page a staff was selected and clicked remove staff button.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Are you sure?&quot; message was displayed along with yes, No and cancel button.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>9</strong></td>
<td><strong>Removing staff</strong></td>
<td>On the staff page a staff was selected and clicked remove staff button.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Click Yes. Staff deleted.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>10</strong></td>
<td><strong>To do list</strong></td>
<td>Click on the to do menu</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input text field pops up</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>OK</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>11</td>
<td>To do list</td>
<td>Insert activities wanted to be done or do.</td>
<td>Inserted activities was displayed in the to do list in the dashboard.</td>
<td>OK</td>
</tr>
<tr>
<td>12</td>
<td>Reservation</td>
<td>Click on the reservation menu in the dashboard.</td>
<td>Reservation form opens.</td>
<td>OK</td>
</tr>
<tr>
<td>13</td>
<td>Reservation</td>
<td>Insert necessary data like Name, address, phone number, date and time for the reservation.</td>
<td>Confirm the reservation.</td>
<td>OK</td>
</tr>
<tr>
<td>14</td>
<td>Cancel reservation</td>
<td>Selected the desire reservation to be cancel.</td>
<td>Click on cancel reservation.</td>
<td>OK</td>
</tr>
<tr>
<td>15</td>
<td>Billing</td>
<td>Click on the bill under the option menu.</td>
<td>Billing windows open.</td>
<td>OK</td>
</tr>
<tr>
<td>16</td>
<td>Billing</td>
<td>Entered all the details in the billing window.</td>
<td>Total bill including service tax and VAT in displayed.</td>
<td>OK</td>
</tr>
<tr>
<td>17</td>
<td>Billing</td>
<td>Customer pays certain amount.</td>
<td>The returnable amount after deducting total bill is displayed along with print bill option.</td>
<td>OK</td>
</tr>
</tbody>
</table>
9 SUMMARY

The main goal of developing this desktop application was to make the day to day operation of the SASA restaurant easy and smooth. This application helps to know how busy the restaurant is going to be in a certain day or in a certain time of the day. Since the employee know when and how busy they are going to be in advance they can make a pre plan to make their operation smooth and give their customers high level satisfaction.

The features of this application are that it can be used to create or remove users, view billing history print bill and update username and password.

Since we have tested the application in the Windows environment and it runs smoothly we can state that the main objective of this application had been achieved.
10 CONCLUSION

The documentation includes all necessary information on the structure and the coding of the program created for Restaurant Billing system. Creating the program was an overwhelming task that required a lot of analyzing, research work and personal skills.

Creating this report has been a great experience and numerous facts have been learned since the required tasks were very challenging. Tasks such as creating a system to a restaurant, needed research work as well as personal skills. Creating proper design and smooth flow of operation was a very tiring task that consumed a lot of time. The program has been created successfully with proper design and working flow of operation.

The billing part was the most important and difficult part of this project. Retrieving data from database for the billing purpose was quite a tough task. Making changes on multiple database using SQL queries was also a bit time consuming and though provoking task.

In conclusion, many experiences have been gained specially in coding. Time management and teamwork is very important in the development system.

10.1 Future Enhancement

RBS software offers various functionalities needed to effectively manage restaurant operations such as staff management, making orders, billing, menu management, reservation, viewing order history, to-do list and many more.

In the future, many enhancements/upgrades can be made to the existing software. Various enhancements such as inventory management and control, wireless table side ordering and payment, real-time alerts, online ordering, mobile management capabilities can help increase revenue and cut cost.
REFERENCES:


/2/ Objective: Accessed 10.04.2017

/3/ Scopes and Limitation: Accessed 10.05.2017

/4/ Feasibility study: Accessed 1.04.2017
http://www.slideshare.net/alok104/synopsis-on-billing-system-27487568

https://www.techopedia.com/definition/3243/unified-modeling-language-uml/

http://whatis.techtarget.com/definition/use-case-diagram

/7/ Class Diagram: Accessed 12.04.2017
http://searchmicroservices.techtarget.com/definition/class-diagram

https://creately.com/blog/diagrams/sequence-diagram-tutorial/

http://searchcrm.techtarget.com/definition/entity-relationship-diagram

/10/ Interfaces: Accessed 25.04.2017
https://www.youtube.com/watch?v=9K5sS7j5wWI