



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

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.

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Degree Program in Information Technology

ABSTRACT

Author	Sujit Maharjan
Title	Restaurant Billing System
Year	2017
Language	English
Pages	61
Name of Supervisor	Ghodrat Moghadampour

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

CONTENTS

ABSTRACT

LIST OF ABBREVIATION	1
1 INTRODUCTION	2
1.1 BACKGROUND	2
1.2 Objective of Project	3
1.3 Problem Definition.....	3
1.4 Project Overview	4
1.5 Scopes and Limitations	4
1.6 Tools Used	5
2 FUNCTIONAL REQUIREMENTS.....	6
2.1 Non-functional requirements	6
2.2 Feasibility Study	7
2.3 Technical feasibility.....	8
2.4 Economic feasibility	9
2.5 Operational feasibility.....	9
3 GENERAL ARCHITECTURE	10
3.1 Unified Modeling language (UML).....	11
3.2 Use case diagram	11
3.3 Class Diagram.....	13
3.4 Sequence Diagram	15
3.5 Admin Login.....	15
3.6 Add User	16
3.7 Edit User	17
3.8 Delete User.....	18
3.9 User Login	19
3.10 User Password Change	20
3.11 Reservation	21
3.12 Billing	22
4 DATA MODELING.....	23

4.1	Database	23
4.2	Design of the Database	23
4.3	ER Diagram	23
5	CODING TOOLS	26
5.1	User Interface	26
5.2	Login Page	27
5.3	Dashboard	28
5.4	Manage menu Page	29
5.5	Billing Interface Page	30
5.6	Staff Information Page	31
6	RESULT	32
6.1	Adding a new staff Page	32
6.2	Removing a staff Page	33
6.3	Updating staff information Page	34
6.4	Reservation Page	35
6.5	Billing Output Page	36
6.6	Bill Print Output Page	37
7	IMPLEMENTATION	38
7.1	Admin Login Function	38
7.2	Admin Dashboard Function	39
7.3	Add User Function	40
7.4	Add Category Function	41
7.5	Add Menu Function	42
7.6	Update Menu Function	43
7.7	Delete Employee Function	44
7.8	Change Password Function	45
7.9	Reservation Function	46
7.10	Billing Function	47
8	TESTING	49
9	SUMMARY	52

10 CONCLUSION	53
10.1 Future Enhancement	53
REFERENCES:	54

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

LIST OF ABBREVIATION

RBS	Restaurant Billing System
SQL	Structured Query Language
IT	Information Technology
OS	Operating System
PHP	Personal Home Page
IDE	Integrated Development Environment
SDK	Software Development Kit
DB	Database
DBMS	Database Management System
RDBMS	Relational Database Management system

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/

1.1 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: /2/

- 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:/3/

- 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 standardize 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:/4/

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.

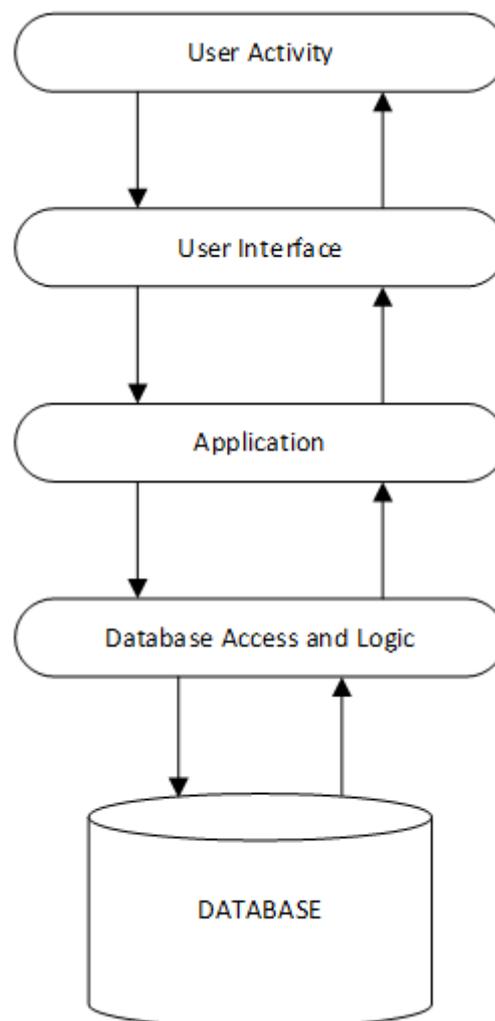


Figure 1: General Architecture of the system

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. /5/

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. /6/

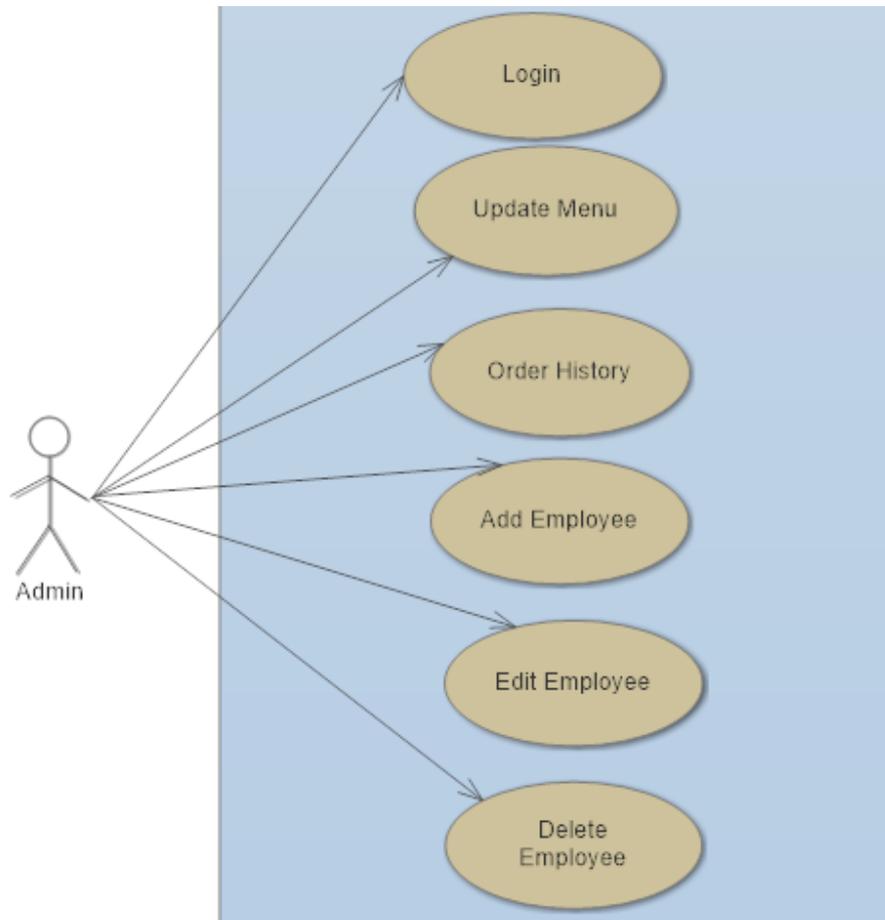


Figure 2: Admin Function

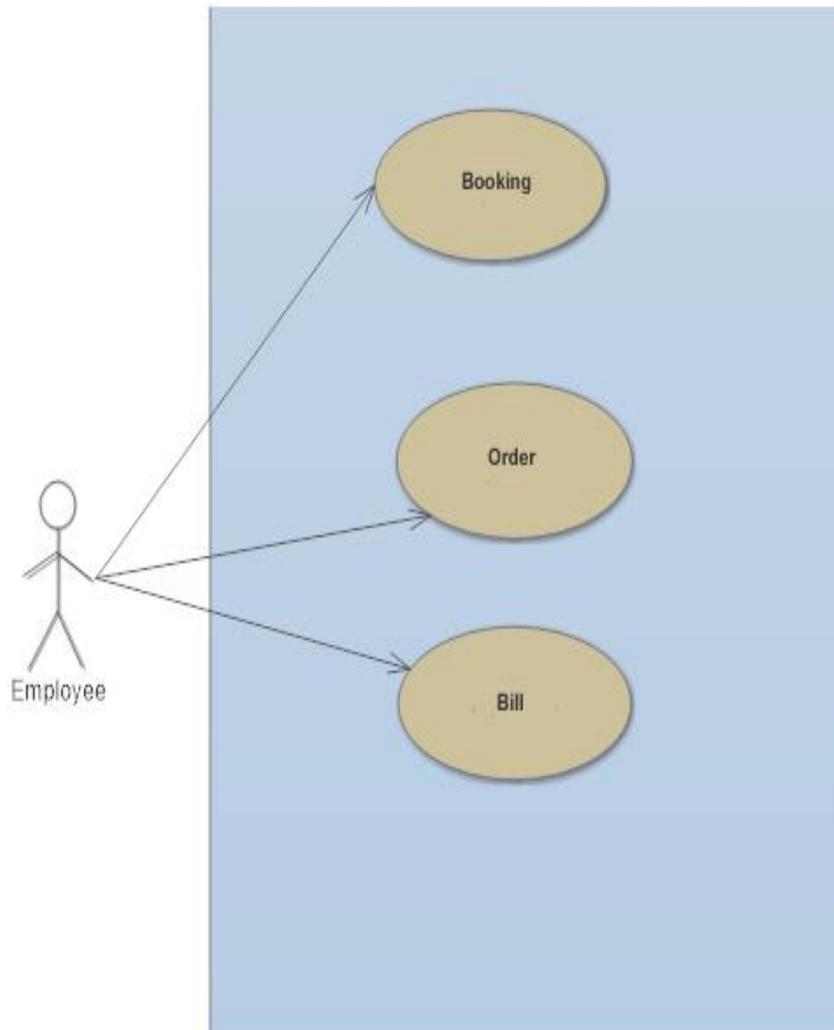


Figure 3: Employee Function

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/

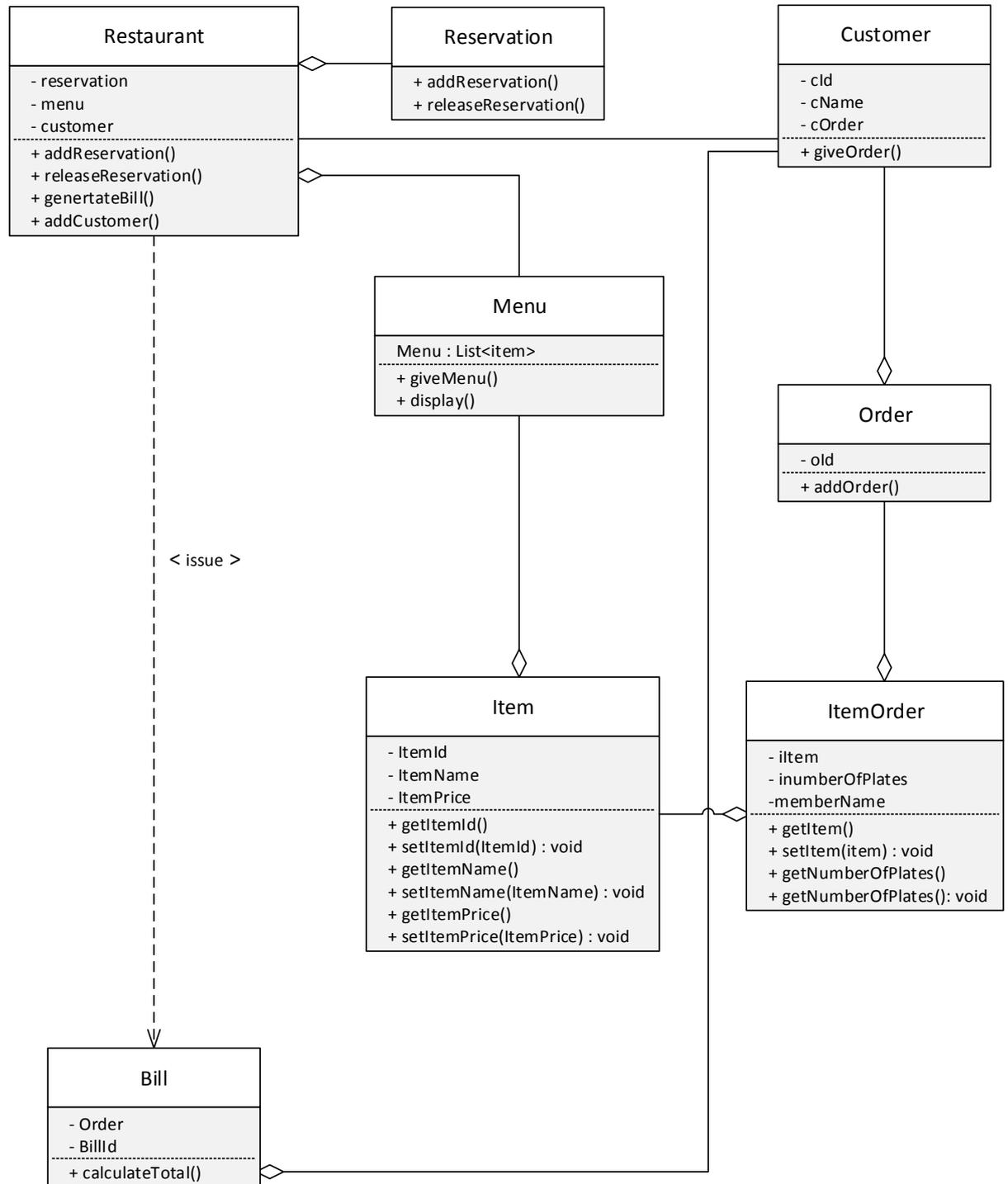


Figure 4: Class Diagram

3.4 Sequence Diagram

Sequence diagram is a 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: /8/

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.

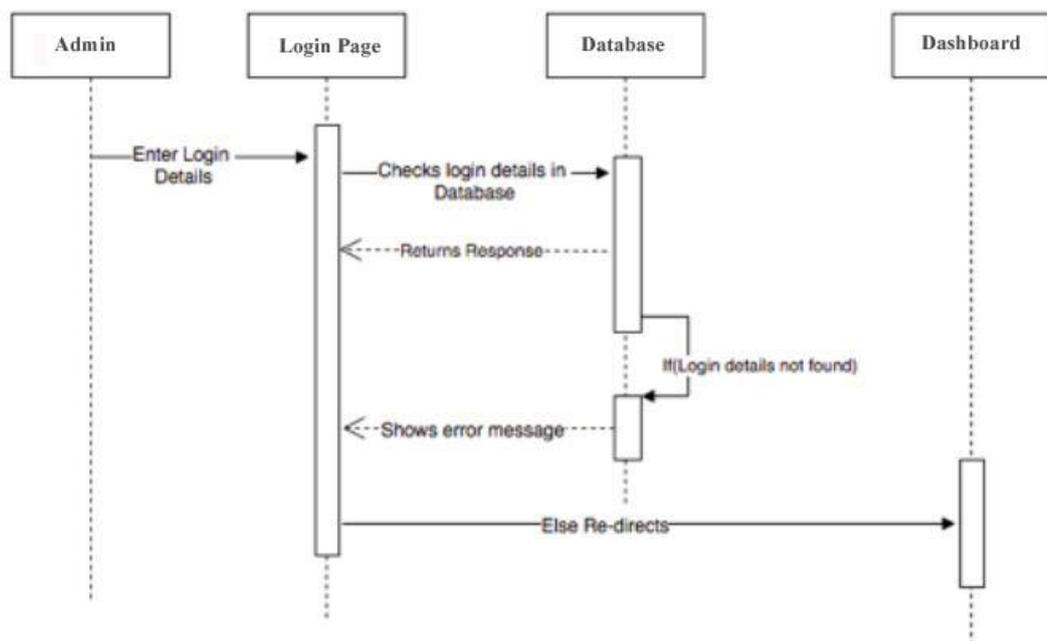


Figure 5: Admin Login

3.6 Add User

This sequence diagram shows the different steps how admin can add 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.

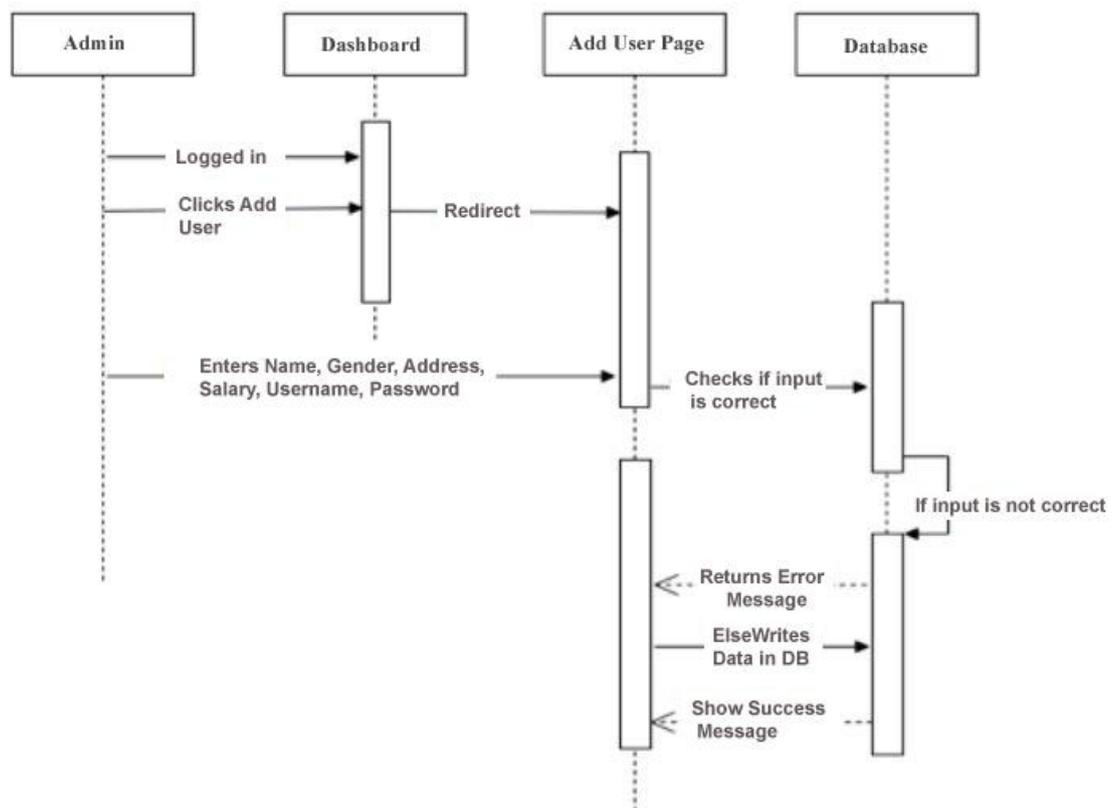


Figure 6: Add User

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.

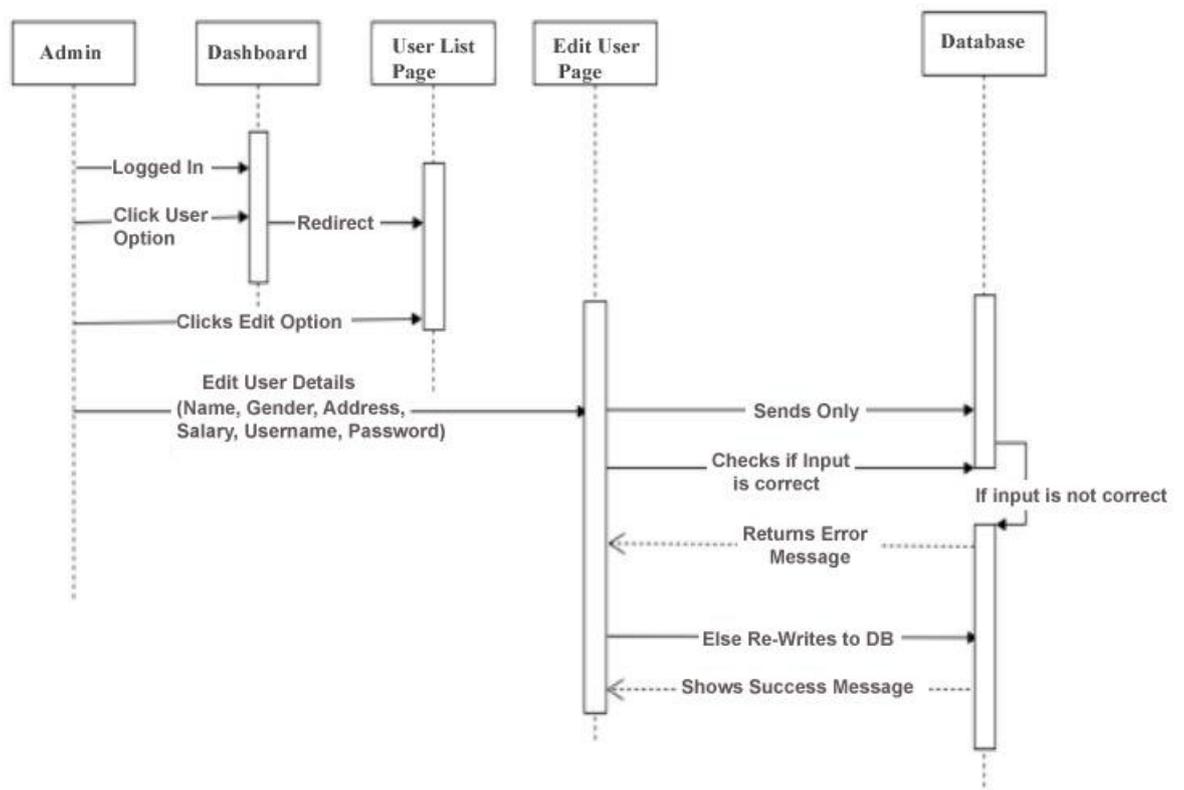


Figure 7: Edit user

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.

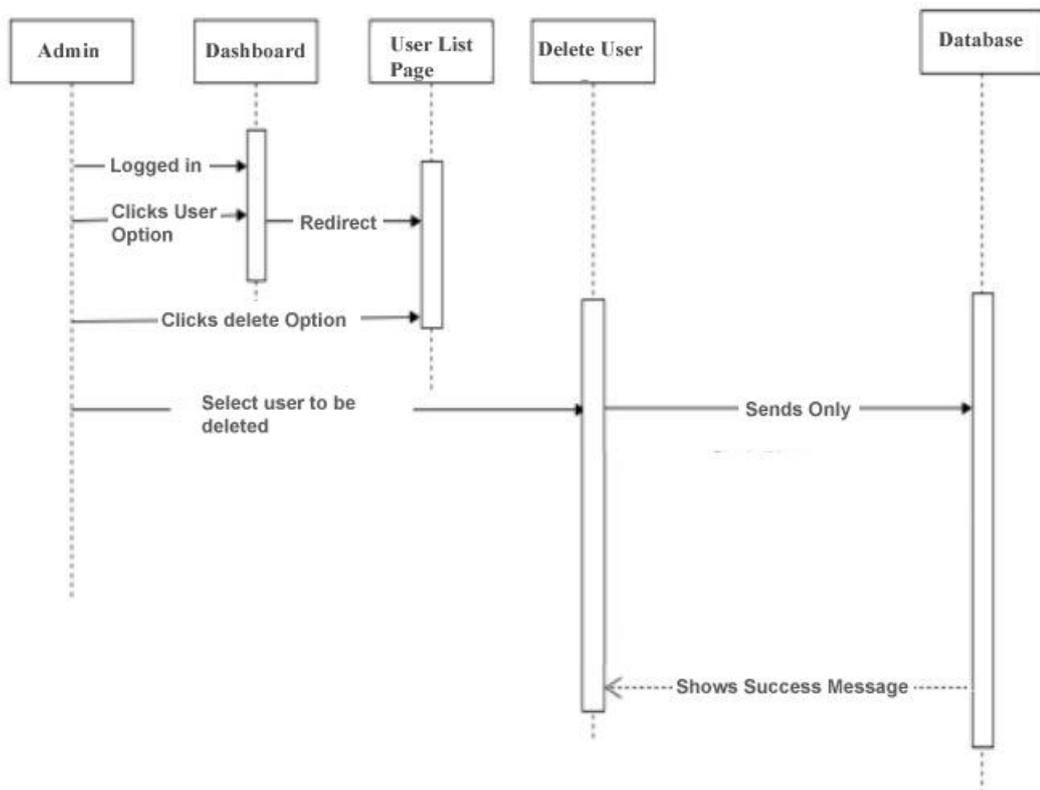


Figure 8: Delete user

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.

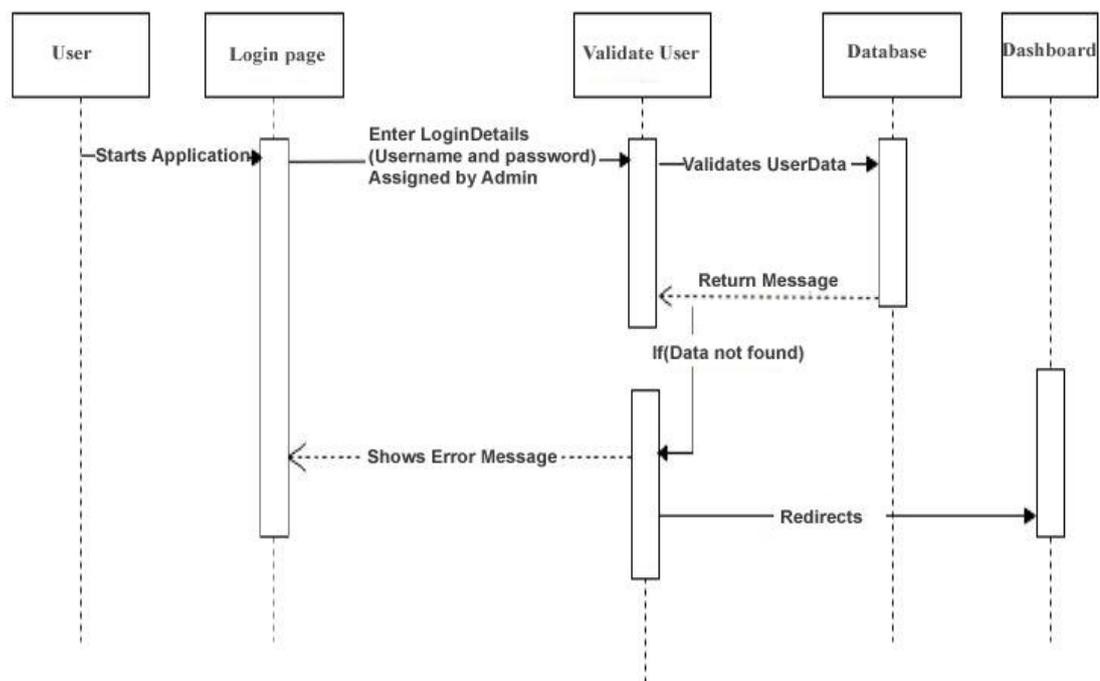


Figure 9: User Login

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.

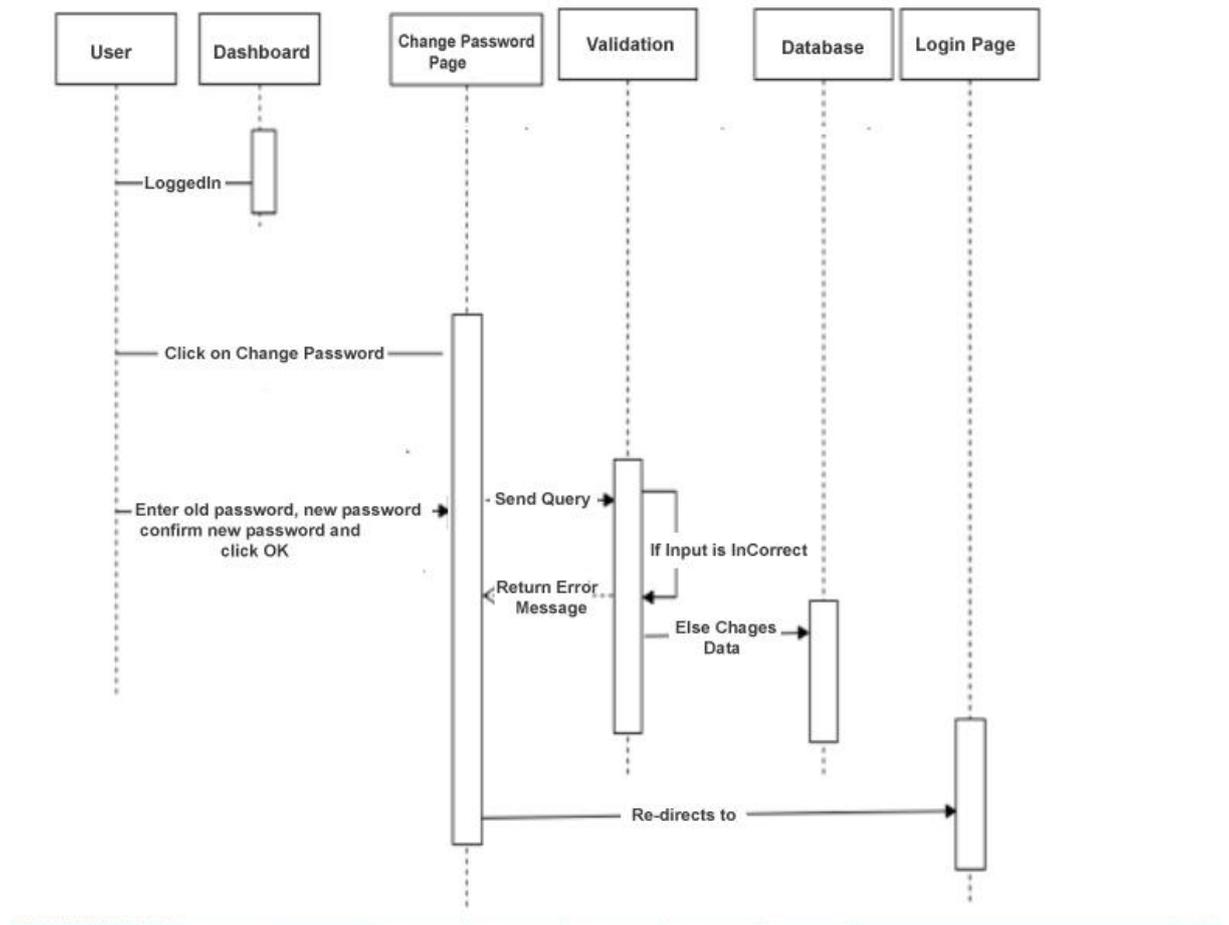


Figure 10: User Password Change

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.

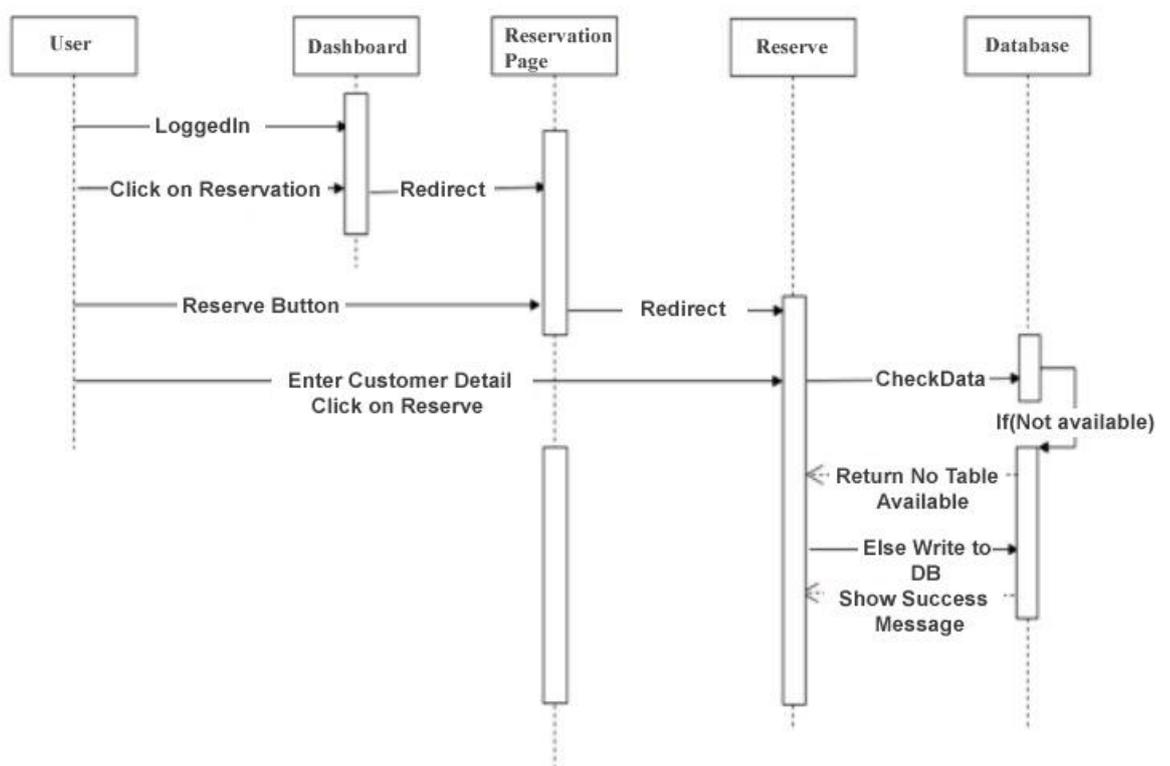


Figure 11: Reservation

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.

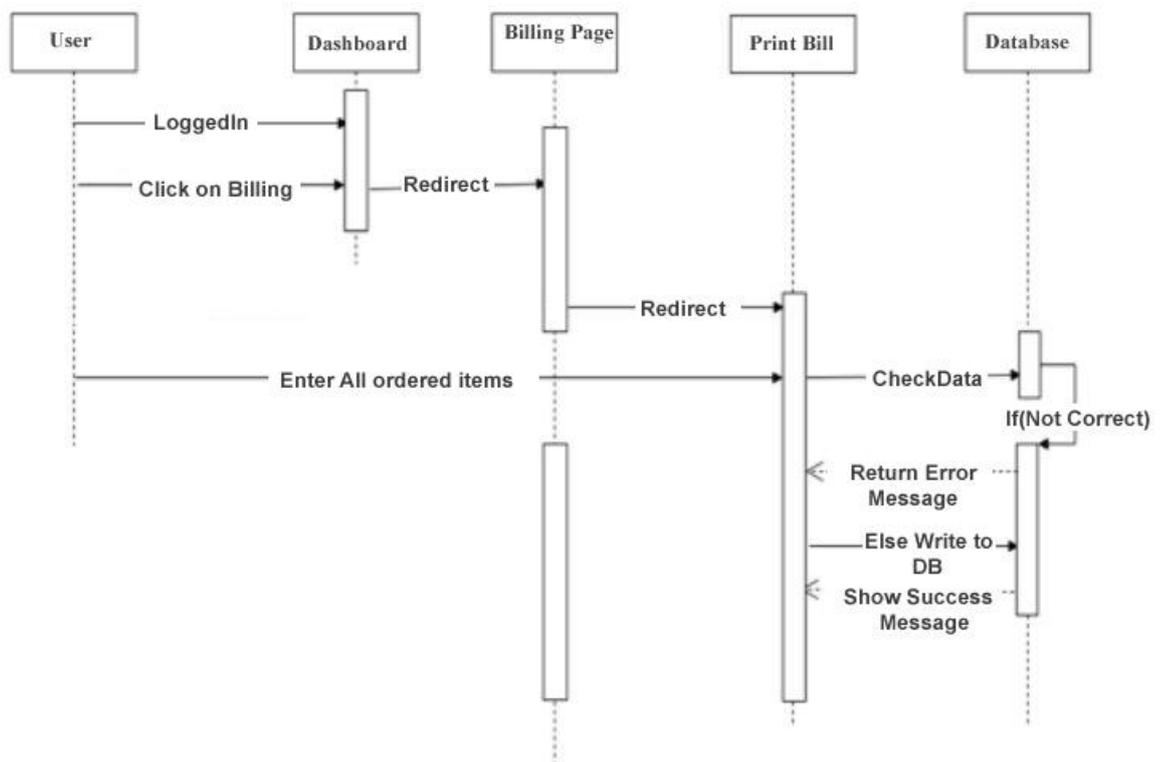


Figure 12: Billing

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/

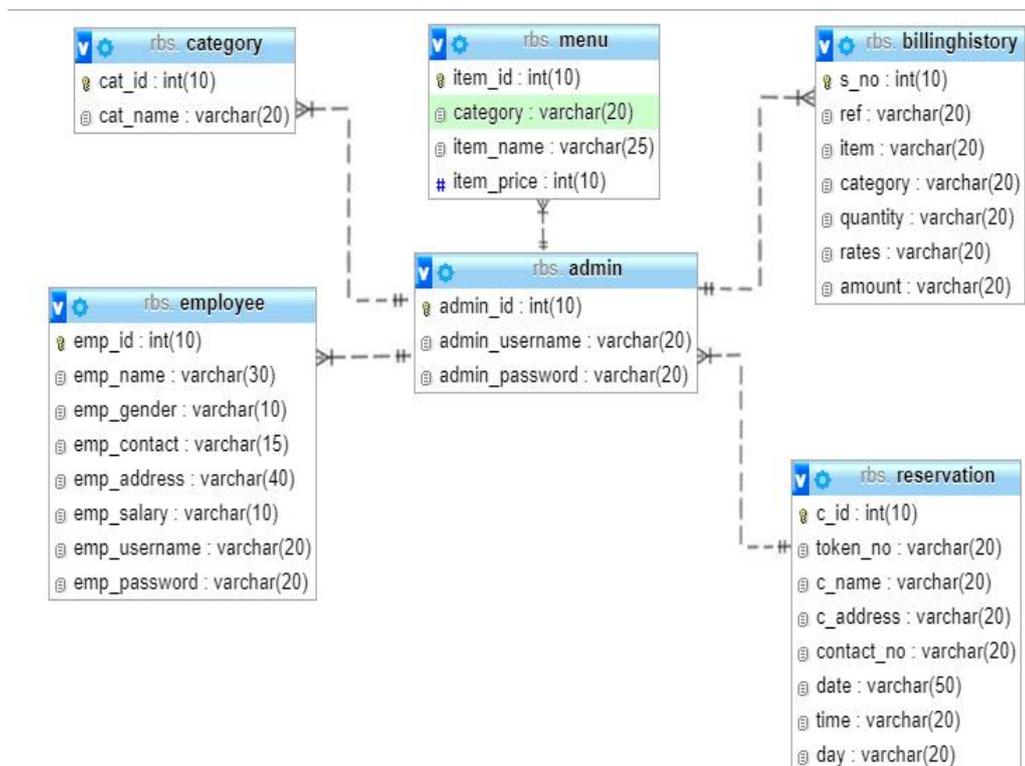


Figure 13: ER Diagram

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.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> admin	★ Browse Structure Search Insert Empty Drop	1	MyISAM	latin1_swedish_ci	2 KiB	-
<input type="checkbox"/> category	★ Browse Structure Search Insert Empty Drop	11	MyISAM	latin1_swedish_ci	2.2 KiB	-
<input type="checkbox"/> billinghistory	★ Browse Structure Search Insert Empty Drop	10	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> reservation	★ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> employee	★ Browse Structure Search Insert Empty Drop	1	MyISAM	latin1_swedish_ci	2.1 KiB	56B
<input type="checkbox"/> menu	★ Browse Structure Search Insert Empty Drop	11	MyISAM	latin1_swedish_ci	2.2 KiB	-

Figure 14: Database tables of the application

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. /10/

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.

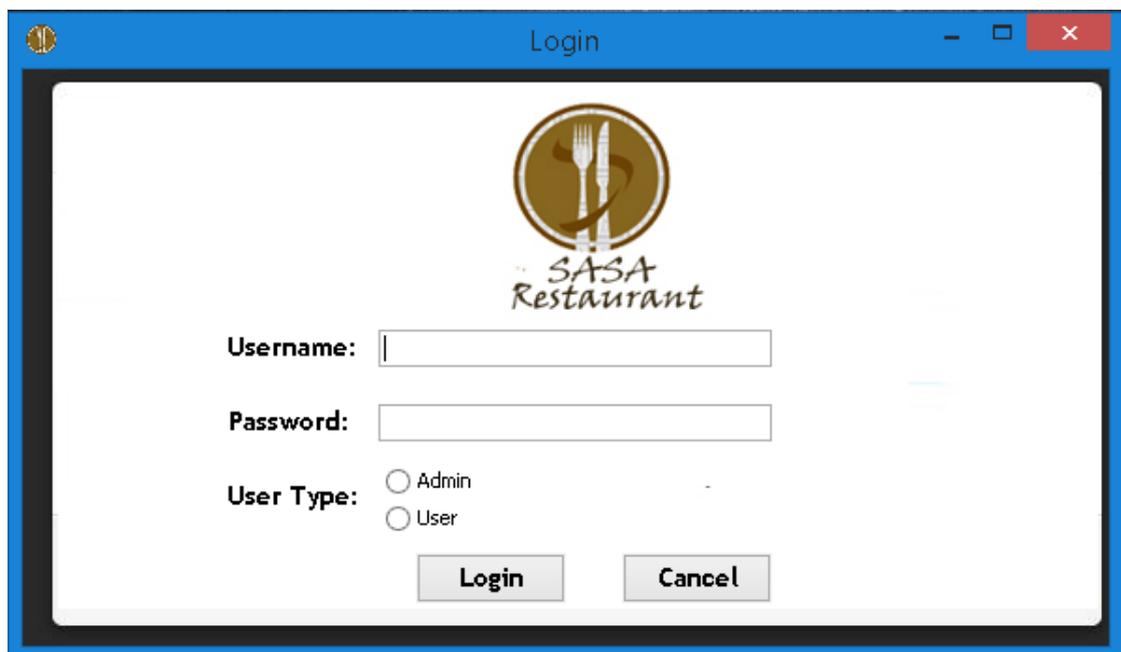


Figure 15: Login Interface

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.

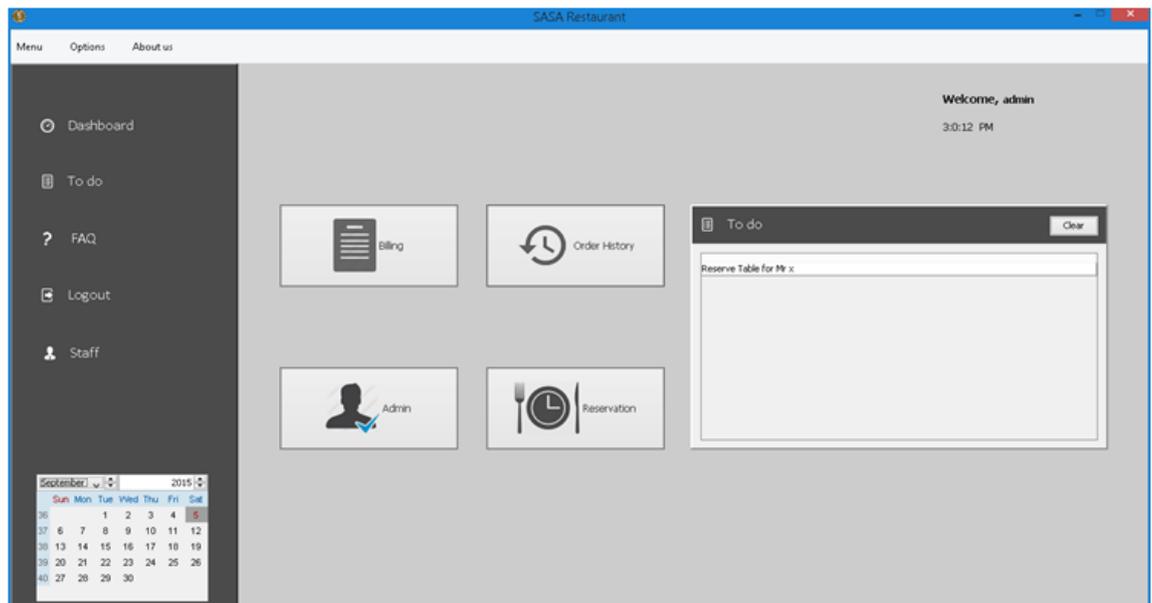


Figure 16: Dashboard Interface

5.4 Manage menu Page

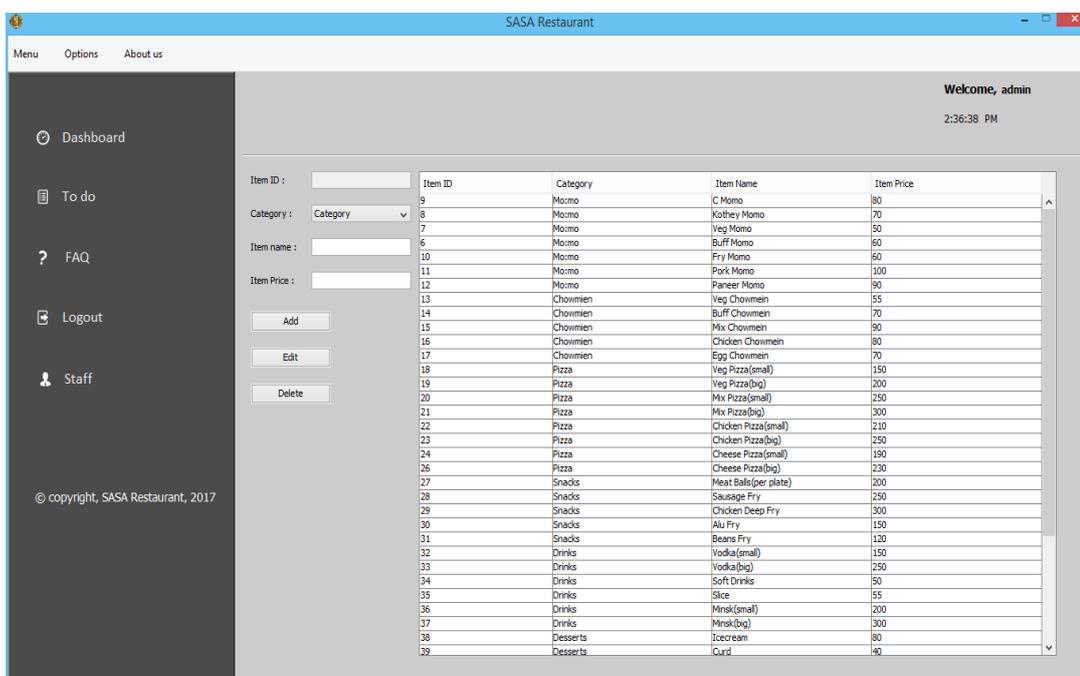
In this page admin can add, remove and edit the items in the menu. Here admin can add food category, food name and its price. For instance

category: Soft Drink

Item type: Coca cola

Item Price: Rs 100

Figure 17 shows the manage menu.



The screenshot displays the 'Manage menu' interface for 'SASA Restaurant'. The page includes a sidebar with navigation links: Dashboard, To do, FAQ, Logout, and Staff. The main content area features a 'Welcome, admin' message and a timestamp of 2:36:38 PM. Below this, there is a form for adding new items with fields for 'Item ID', 'Category', 'Item name', and 'Item Price', along with 'Add', 'Edit', and 'Delete' buttons. A table lists the current menu items with columns for 'Item ID', 'Category', 'Item Name', and 'Item Price'.

Item ID	Category	Item Name	Item Price
9	Momo	C Momo	80
8	Momo	Kolhey Momo	70
7	Momo	Veg Momo	50
6	Momo	Buff Momo	60
10	Momo	Fry Momo	60
11	Momo	Pork Momo	100
12	Momo	Paneer Momo	90
13	Chowmien	Veg Chowmien	55
14	Chowmien	Buff Chowmien	70
15	Chowmien	Mix Chowmien	90
16	Chowmien	Chicken Chowmien	80
17	Chowmien	Egg Chowmien	70
18	Pizza	Veg Pizza(small)	150
19	Pizza	Veg Pizza(big)	200
20	Pizza	Mix Pizza(small)	250
21	Pizza	Mix Pizza(big)	300
22	Pizza	Chicken Pizza(small)	210
23	Pizza	Chicken Pizza(big)	250
24	Pizza	Cheese Pizza(small)	150
26	Pizza	Cheese Pizza(big)	230
27	Snacks	Meat Balls(per plate)	200
28	Snacks	Sausage Fry	250
29	Snacks	Chicken Deep Fry	300
30	Snacks	Alu Fry	150
31	Snacks	Beans Fry	120
32	Drinks	Vodua(small)	150
33	Drinks	Vodua(big)	250
34	Drinks	Soft Drinks	50
35	Drinks	Slice	55
36	Drinks	Minsk(small)	200
37	Drinks	Minsk(big)	300
38	Desserts	Iccream	80
39	Desserts	Curd	40

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 is given in the figure 18.

The screenshot displays the 'SAsA Restaurant' billing interface. A modal window titled 'New Bill' is open, showing the following details:

- Ref No.: REF (with a note: *Increment by 1, the used Ref No is REF12)
- Bill No.: [Empty]
- Category: Select Category (dropdown)
- Item: Item (dropdown)
- Table No.: Select Table (dropdown)
- Qty: [Empty]
- Rate: [Empty]

Below the input fields is a table with the following structure:

S.No	Item	Category	Qty	Rate	Amount

At the bottom of the modal, there are several calculated fields and a 'Print Bill' button:

- Sub Total: [Empty]
- VAT: 13%
- Service Charge: 2%
- Grand Total: [Empty]
- Paid Amount: [Empty]
- Return Amount: [Empty]
- Discount: 0.00
- Issued By: admin
- Print Bill button

The background interface shows a sidebar with navigation options (Dashboard, To do, FAQ, Logout, Staff) and a calendar for September 2015. The main content area displays 'Welcome, admin' and the time '3:28:17 PM'.

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.

The screenshot displays the 'SAsA Restaurant' staff management interface. On the left, a sidebar contains navigation links: Dashboard, To do, FAQ, Logout, and Staff. The main content area is titled 'Welcome, admin' and shows the time '3:1:18 PM'. Below this, there is a form for adding or editing staff with fields for Id, Name, Gender (radio buttons for Male and Female), Contact, Address, Salary, Username, and Password. At the bottom of the form are buttons for 'Add Employee', 'Edit Employee', and 'Delete Employee'. To the right of the form is a table listing the current staff members.

ID	Name	Gender	Contact	Address	Salary	Username	Password
9	Anil	Male	9843243277	Kathmandu	329432	anil	anil
10	Shristi	Female	9849101466	Kathmandu	3432435	shristi	shrestha
12	Ram	Male	4364	Bratnagar	34343	ram	ram
13	Anjali	Female	87435345	Ranibhawan	353423	anjali	anjali
14	Dipendra	Male	436546	Swyambhu	4654	dpendra	dpendra
15	Pooja	Female	43346	Kathmandu	325345	pooja	pooja
16	Kiran	Male	4364	Pokhara	46563	polhara	polhara
17	Nabin	Male	46454	Kathmandu	43546	nabin	nabin
18	Raman	Male	436546	Chhetrapati	43653	raman	raman
19	Raju	Female	36542	Nardevi	45434	raju	raju
20	Pujita	Female	25345	Kathmandu	3465346	pujita	pujita
21	Tina	Female	3466	Chhetrapati	32423	tina	tina
22	Shrijana	Female	3453634	Pokhara	34535	shrijana	shrijana
23	Manjil	Male	5465474	Shorakhutte	436534	manjil	nepal
26	User	Male	834797349	Pokhara	44334	user	user

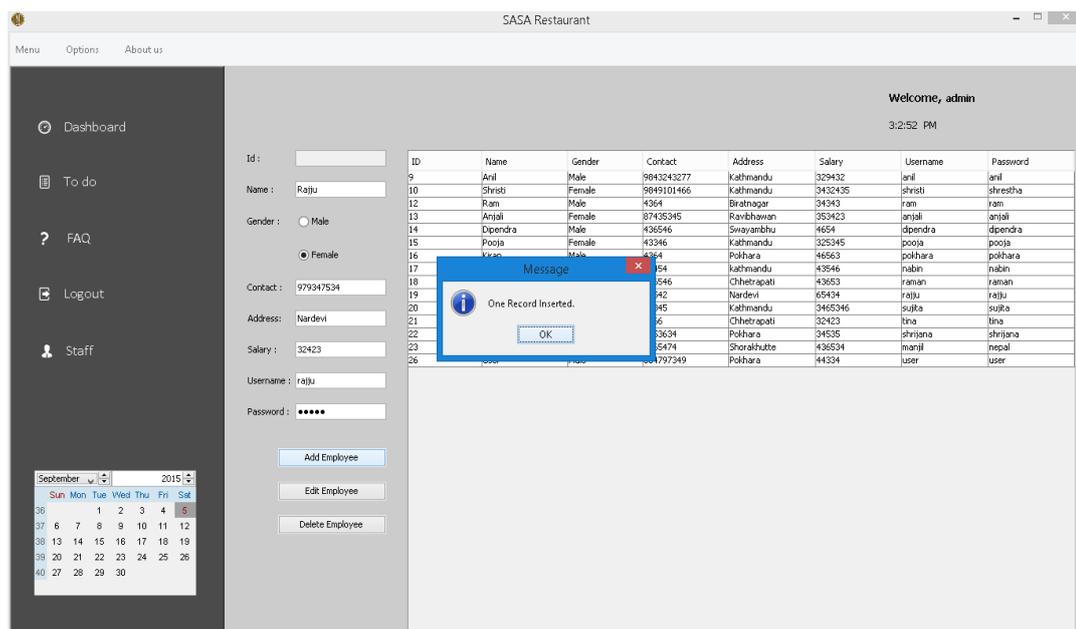
Figure 19: Staff Information Page

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.



The screenshot displays the 'SASA Restaurant' application interface. On the left is a navigation menu with options: Dashboard, To do, FAQ, Logout, and Staff. The main area shows a 'Welcome, admin' message and the time '3:2:52 PM'. Below this is a form for adding a new employee with fields for ID, Name (Rajju), Gender (Female selected), Contact (979347534), Address (Nardevi), Salary (32423), Username (rajju), and Password (masked). A table of existing staff is visible, and a message box states 'One Record Inserted.' The table data is as follows:

ID	Name	Gender	Contact	Address	Salary	Username	Password
9	Anil	Male	9843243277	kathmandu	329432	anil	anil
10	Shristi	Female	9849101466	kathmandu	3432435	shristi	shrestha
12	Ram	Male	4364	Bratnagar	34343	ram	ram
13	Anjali	Female	87455345	Ranbhawan	359423	anjali	anjali
14	Dipendra	Male	4365946	Swayambhu	4654	dipendra	dipendra
15	Pooja	Female	43346	kathmandu	325345	pooja	pooja
16	Kiran	Male	4364	Pokhara	46563	polhara	polhara
17			54	kathmandu	43546	nabin	nabin
18			546	Chhetrapati	43653	raman	raman
19			42	Nardevi	65434	rajju	rajju
20			45	kathmandu	3465346	suja	suja
21			6	Chhetrapati	32423	tina	tina
22			3634	Pokhara	34535	shrijana	shrijana
23			5474	Sherathutte	436534	manjil	mepal
26			984797249	Pokhara	44334	user	user

Figure 20: Adding staff

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.

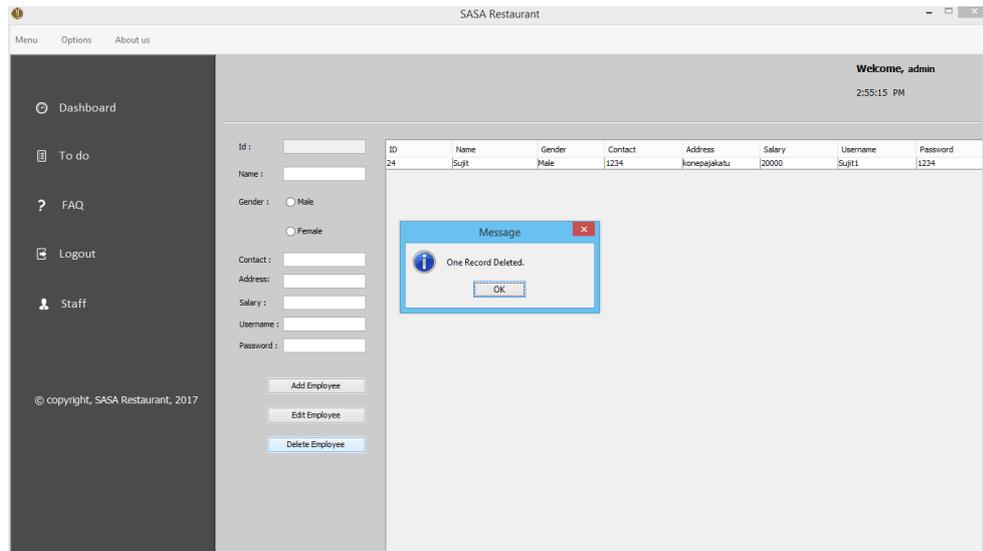


Figure 21: Removing staff

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.

The screenshot displays the 'SASA Restaurant' web application interface. On the left is a navigation sidebar with options: Dashboard, To do, FAQ, Logout, and Staff. The main content area shows a 'Welcome, admin' message and the time '3:33 PM'. Below this is a form for updating an employee, with fields for ID (27), Name (Raju), Gender (Female), Contact (979347534), Address (Nardevi), and Salary (32423). A 'Update Employee' button is visible. A modal message box in the center reads 'Message: One Record Updated.' with an 'OK' button. To the right is a table of staff information.

ID	Name	Gender	Contact	Address	Salary	Username	Password
9	Anil	Male	9843243277	Kathmandu	329432	anil	anil
10	Shristi	Female	98493101466	Kathmandu	3432435	shristi	shrestha
12	Ram	Male	4384	Birabangor	34343	ram	ram
13	Anjali	Female	87435345	Ravibhawan	353423	anjali	anjali
14	Dipendra	Male	436546	Swayambhu	4654	dipendra	dipendra
15	Pooja	Female	43346	Kathmandu	325345	pooja	pooja
16	Kiran	Male	4364	Pokhara	46563	pojhara	pojhara
17			454	kathmandu	43546	nabin	nabin
18			4546	Chhetrapati	43653	raman	raman
19			42	Nardevi	85434	raju	raju
20			45	Kathmandu	3445346	suja	suja
21			45	Chhetrapati	32423	tna	tna
22			33634	Pokhara	34535	shriana	shriana
23			45474	Shorahutte	436534	manil	nepal
26			4797349	Pokhara	44334	user	user
27	Raju	Female	979347534	Nardevi	32423	raju	raju

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.

The screenshot shows the SASA Restaurant reservation page. A message box titled 'Message' with an information icon and 'Reservation Completed' text is displayed over the reservation table. The table lists reservation details including Token No, Customer Name, Customer Address, Contact No, Date, and Table For. The table has 7 columns and 6 rows of data. The last row is highlighted in blue.

Cutom...	Token No	Customer Name	Customer Address	Contact No	Date	Table For
15	RSV234	Anjali	Ravibhavan	435346	2012	4
16	RSV1111	Rajesh Shakya	Kathmandu	4364654	MMW d, yyyy	2
18	RSV	Sita Gurung	Kathmandu	9439868	{DateChooser	2
20	RSV1222	Anil	Kathmandu	325435	Sat Sep 19 18:51:38 NP...	2
21	RSV1223	Sund Maharjan	Kathmandu	9899110464	Sun Sep 13 00:00:00 N...	2

Below the table, there are input fields for reservation details: Id, Token No, Customer Name, Customer Address, Contact No, Date (Sep 13, 2015), Table For (2), and Reserved Token No (RSV). Buttons for Reserve, Check Reservation, Assigned, and Cancel are visible at the bottom.

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.

The screenshot displays the 'SASA Restaurant' interface. A 'New Bill' modal window is open, showing the following details:

- Ref No.: REF13 (Note: *Increment by 1, the used Ref No is REF12)
- Category: Chowmein
- Item: Mix Chowmein
- Table No.: 2
- Qty: 2
- Rate: 90

The item list table is as follows:

Item	Category	Qty	Rate	Amount
C. Menu	Mix.cho	2	90	180
Soft Drinks	Drinks	2	50	100
Mix Chowmein	Chowmein	2	90	180

Summary fields at the bottom of the form:

- Sub Total: 460
- VAT: 59.8
- Service Charge: 9.2
- Grand Total: 529.0
- Paid Amount: 600
- Return Amount: -71.0
- Discount: 0.00
- Issued By: user

The background shows a dashboard with a sidebar (Dashboard, To do, FAQ, Logout) and a calendar for September 2015. A 'Welcome, user' message and the time '3:56:10 PM' are visible in the top right.

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.

The screenshot shows a web application interface for Sasa Restaurant. A central window displays the bill print output. The bill includes the restaurant's logo, name, address, and contact information. It lists the items ordered with their respective quantities, rates, and amounts. The total bill amount is 789.75. The page also shows a sidebar with navigation options like Dashboard, To do, FAQ, Logout, and Staff, and a calendar for April 2017.

Item	Category	Qty	Rate	Amount
Veg Chowmein	Chowmien	3	55	165
Veg Momo	Momo	3	50	150
Kothey Momo	Momo	3	70	210
C Momo	Momo	3	80	240
Veg Pizza(mall)	Pizza	3	150	450

Sub Total: 1215.0
 Valt: 157.95
 Service Charge: 24.3
 Discount: 607.5
 Grand Total: 789.75
 Issued By: admin

Thank you for visiting. 😊

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.

```

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 u
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.

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

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

```
private void addBttnActionPerformed(java.awt.event.Actio
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.getPass
employee.setPassword(password);

    new EmployeeDAO().addEmployee(employee);
    loadDatas();
    clearFields();
}

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.

```

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=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.

```

public void updateMenu(MenuDTO menuUpdate) {
    try {
        String sql = "UPDATE menu SET
category=?,item_name=?,item_price=? WHERE item_id=?";
        pstmt = (PreparedStatement)
con.prepareStatement(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.

```

int resulte = new
EmployeeDAO().deleteFromEmployee(String.valueOf(table.g
etValueAt(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
jButton1ActionPerformed(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, currentPasswo
rd, types)) {
result = cdb.changePassword(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.

```
private void
reserveBtnActionPerformed(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.

```
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

```
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 =
parseInt(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) {

    }

```

Code Snippet 13: Amount After VAT and TAX

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

No	Test Performed	Description	Result	Corrections and Result
1	Application file	Application file was launched	Login page opens	OK
2	Login page	Text field and radio button	This page contains text field for username and password and radio field for selecting either admin or user and login and cancel button.	OK
3	Login using wrong username and password	Entered wrong username and password.	Couldn't login gives error message.	OK
4	Login using correct username	Entered correct username and pass-	Login successful.	OK

	and password	word		
5	Manage menu	In the home page manage menu was click under menu icon.	Entered different categories and items under it.	OK
6	Adding staffs	On the staff page tried to add staff with entering any data.	An error message was displayed.	OK
7	Adding staffs	On the staff page entered all the data and click add employee.	New employee added.	OK
8	Removing staff	On the staff page a staff was selected and clicked remove staff button.	"Are you sure?" message was displayed along with yes, No and cancel button.	OK
9	Removing staff	On the staff page a staff was selected and clicked remove staff button.	Click Yes. Staff deleted.	OK
10	To do list	Click on the to do menu	Input text field pops up	OK

11	To do list	Insert activities wanted to be done or do.	Inserted activities was displayed in the to do list in the dashboard.	OK
12	Reservation	Click on the reservation menu in the dashboard.	Reservation form opens.	OK
13	Reservation	Insert necessary data like Name, address, phone number, date and time for the reservation.	Confirm the reservation.	OK
14	Cancel reservation	Selected the desire reservation to be cancel.	Click on cancel reservation.	OK
15	Billing	Click on the bill under the option menu.	Billing windows open.	OK
16	Billing	Entered all the details in the billing window.	Total bill including service tax and VAT in displayed.	OK
17	Billing	Customer pays certain amount.	The returnable amount after deducting total bill is displayed along with print bill option.	OK

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 data base 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:

- /1/ Restaurant Billing System: Accessed 10.04.2017
[https://www.scribd.com/doc/283903672/Online-Ordering-System- Project](https://www.scribd.com/doc/283903672/Online-Ordering-System-Project)
- /2/ Objective :Accessed 10.04.2017
[https://www.scribd.com/document/36253350/04-Project-Billing- System](https://www.scribd.com/document/36253350/04-Project-Billing-System)
- /3/ Scopes and Limitation: Accessed 10.05.2017
<https://kungfumas.files.wordpress.com/2017/09/099.pdf>
- /4/ Feasibility study: Accessed 1.04.2017
<http://www.slideshare.net/alok104/synopsis-on-billing-system-27487568>
- /5/ UML Diagram : Accessed 12.04.2017
<https://www.techopedia.com/definition/3243/unified-modeling-language-uml/>
- /6/ Use Case Diagram: Accessed 12.04.2017
<http://whatis.techtarget.com/definition/use-case-diagram>
- /7/ Class Diagram: Accessed 12.04.2017
<http://searchmicroservices.techtarget.com/definition/class-diagram>
- /8/ Sequence Diagram : Accessed 14.04.2017
<https://creately.com/blog/diagrams/sequence-diagram-tutorial/>
- /9/ ER Diagram : Accessed 14.04.2017
<http://searchcrm.techtarget.com/definition/entity-relationship-diagram>
- /10/ Interfaces : Accessed 25.04.2017
<https://www.youtube.com/watch?v=9K5sS7j5wWI>