CHAPTER 1

INTRODUCTION

1.1 Overview of Food ordering and Delivery System

Computers have become part of the life for accessing almost any kind of information. Life in the 21st century is full of technological advancement and in this technological age it is very difficult for any organization to survive without utilizing technology. The World Wide Web contributes greatly to the creation of an ever-increasing global information database. It could also be used as a mechanism to share information within an enterprise.

In today's age of fast food and take-out, many restaurants have chosen to focus on quick preparation and speedy delivery of orders rather than offering a rich dining experience. Until very recently, all of these delivery orders were placed over the phone, but there are many disadvantages to this system, including the inconvenience of the customer needing to have a physical copy of the menu, lack of a visual confirmation that the order was placed correctly, and the necessity for the restaurant to have an employee answering the phone and taking orders. What I propose is an online ordering system, which is a technique of ordering foods online applicable in any food delivery industry. The main advantage of my system is that it greatly simplifies the ordering process for both the customer and the restaurant. When the customer visits the ordering webpage, they are presented with an interactive and up-to-date menu ,complete with all available options and dynamically adjusting prices based on the selected options. After making a selection, the item is then added to their order, which the customer can review the details of at any time before checking out. This provides instant visual confirmation of what was selected and ensures that items in the order are, in fact, what was intended. This system also greatly lightens the load on the restaurant's end, as the entire process of taking orders is automated. Once an order is placed on the webpage, it is entered into the database and then retrieved, in pretty much real-time, by a desktop application on the restaurant's end. Within this application, all items in the order are displayed, along with their corresponding options and delivery details, in a concise and easy to read manner. This allows restaurant employees to quickly go through the orders as they are placed and produce the necessary items with minimal delay and confusion.

1.2 Problem Statement

The challenges encountered by the existing system serve as a major drawback to the realization of efficiency and customer satisfaction. The experience of ordering in most fast food restaurants is not pleasant for the customers. Customers will have to make long queues before placing their orders especially during peak hours and then the ordering staff will record customer orders. Having placed their order, the customer must then wait near the counter until their order is ready for collection. The other problem in the food service industry is that restaurants are not realizing the efficiencies that would result from better application of technology in their daily operations. Fast food business in a very competitive business and one way to stand out from competitors is through improving the business process where business process automation can assist business improvement. The other problem with the current system is that the customers are not able to see the ingredients of the meals before they place their order and also they only have to pay for an order online.

1.3 Objectives

General objectives

- To increase efficiency and improve services provided to the customers through better application of technology in daily operations.
- To be able to stand out from competitors in the food service industry.

Specific objectives

- To enable customers to order custom meals that aren’t in the menu.
- To enable customers to have a visual confirmation that the order was placed correctly.
- To reduce restaurant’s food wastage.
- To ensure correct placement of orders through visual confirmation.
- prove efficiency of restaurant’s staff.
- Eliminate paper work and increase level of accuracy.
- Increase speed of service, sales volume and customer satisfaction.

CHAPTER2

SYSTEM REQUIREMENTS

2.1 Software Requirements

Front End is created using JAVA GUI.

Swing is a GUI portion which is much improved programming model compared to many other languages and development environments. Java beans is frame work of this library . Swing java consists of ‘Look and feel’, ‘Accessibility’, Java 2D’, ‘Drag and Drop. In swings the event model is much clearer, with object-oriented approach along with addition of Java eclipse.Swing is reasonably straight foward, it supports radical feature “pluggable look and feel” under different platforms .Swing applications will be built inside a basic JFrame, which creates the window in whatever OS we are using. Swing is built on top of AWT and is entirely written in Java, using AWT’s lightweight component support.Inparticular, unlike AWT ,the architecture Swing components makes it easy to customize both their appearance and behavior. In Swing, classes that represent GUI components have names beginning with the letter J. Some examples are JButton, JLabel, JSlider etc.

BACK END DATABASE USED –MYSQL 5.5 Command Line Client.

MYSQL Database was used for populating all the tables. MYSQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools or use MYSQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MYSQL front-end tools, MYSQL Workbench is actively developed by Oracle, and is freely available for use. MYSQL ships with many command line tools, from which the main interface is 'mysql' client Third parties have also developed tools to manage MySQL servers

Hardware Requirements

Main Processor : CORE i5

RAM : 4.00 GB

Hard Disk Space : 16 GB

CHAPTER 3

SYSTEM DESIGN

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

3.1 ER Diagram

ER-diagram is a data modeling technique that graphically illustrates an information system entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity frame work infrastructure.

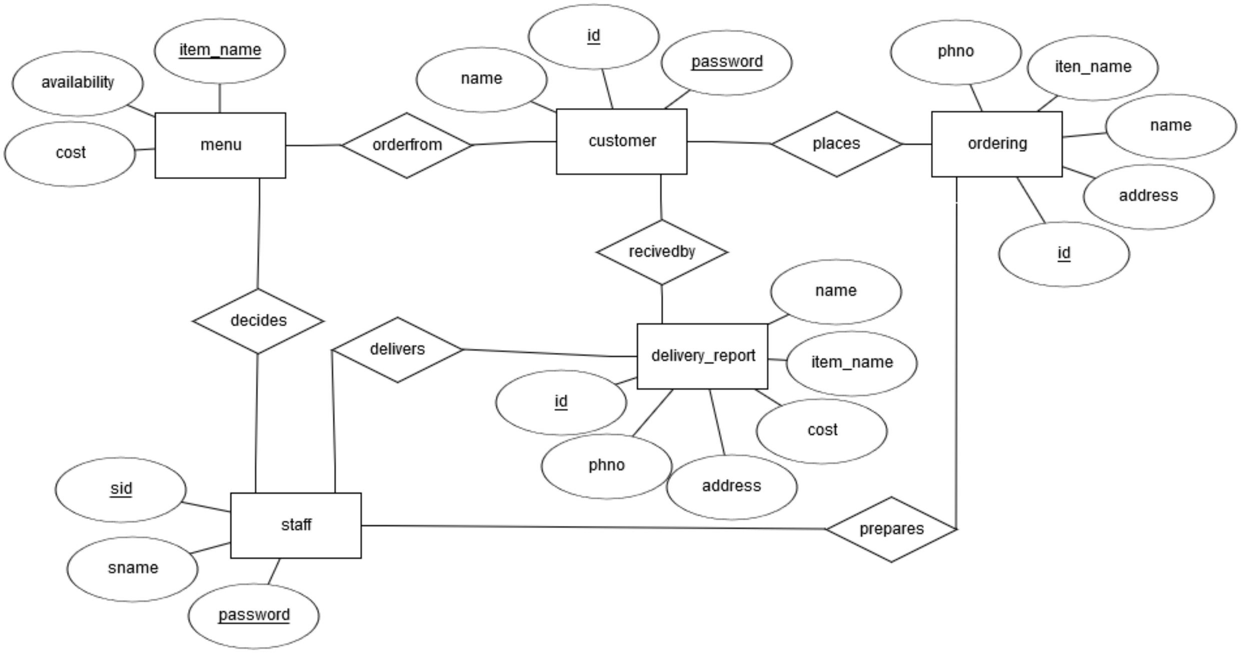


Figure 3.1: An ER diagram for an food ordering and delivery system.

3.2 Schema Diagram

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on data. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It’s the database designers who design the schema to help programmers understand the database and make it useful.

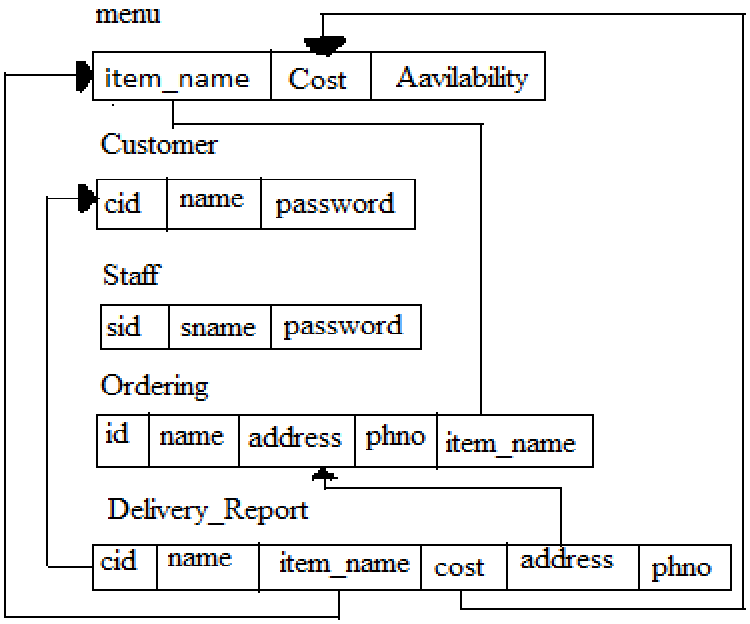


Figure 3.2: An schema diagram for food ordering and delivery system.

3.3 Overview of GUI

Swing GUIs allows to drag and position GUI components from a palette onto a canvas. The GUI Builder automatically takes care of the correct spacing and alignment. We can click onto Jlabels, JButtons, Jtextfields, etc and edit their properties. To design a application using GUI we require software resources like eclipse and JDK .The first step is to create an IDE application for the project in eclipse. To proceed with building an interface, need to create a Java container within which we place other required GUI components. The second step is to create a Jframe container use the palette to populate the applications front end with a Jpane1. add Jtextfields, Jlabels, Jbuttons.

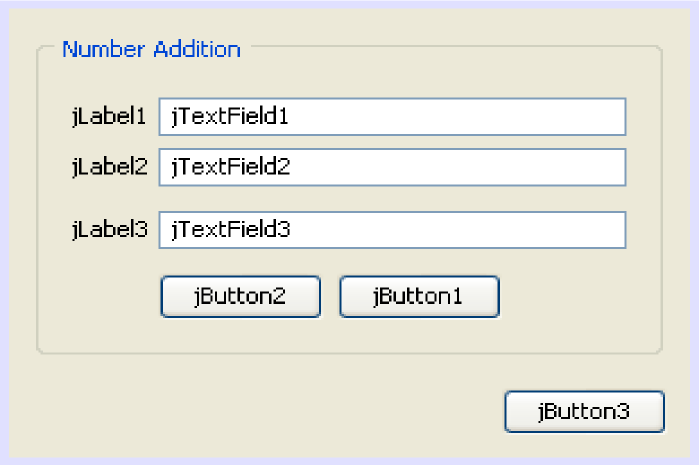


Figure 3.3: Jlable table.

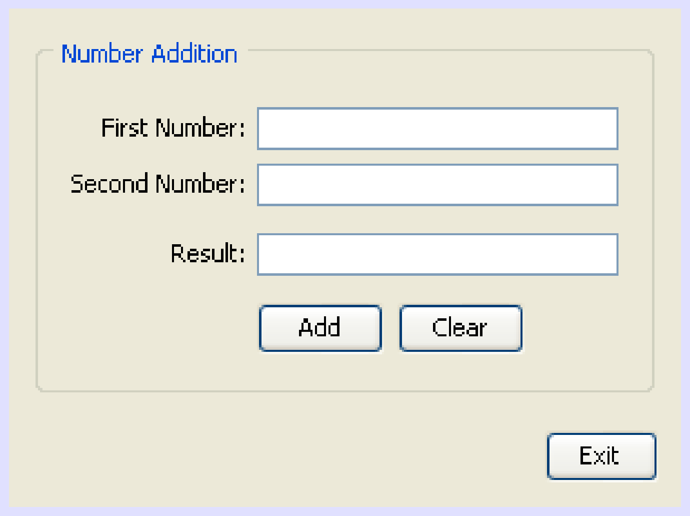


Figure 3.4: Adding component to the Jframe table.

Figure 3.3 also to rename the components that were added to the Jframe.

Next step is to right click on the specific buttons->Events>Action>Action performed:

```
private void btnphysicalActionPerformed(java.awt.event.ActionEvent evt) {
      setVisible(false);  // TODO asetVisible(false);
      physical p=new physical();
      p.setVisible(true);
   }
```

The final step is to save and run the project. But before doing this need to right click on the project>properties>Library>add library>add jar/folder. Once it has added the jar folder, then run the project to display the required output/the webpage which has been designed.

3.4 NORMALIZATION

Normalization, is the process of organizing the columns (attributes) and tables (relations) of a relational database to reduce data redundancy and improve data integrity. Normalization is also the process of simplifying the design of a database so that it achieves the optimal structure composed of atomic elements. Normalization involves arranging attributes in relations based on dependencies between attributes, ensuring that the dependencies are properly enforced by database integrity constraints. Normalization is accomplished by applying some formal rules either by a process of synthesis or decomposition. Synthesis creates a normalized database design based on a known set of dependencies. Decomposition takes an existing (insufficiently normalized) database design and improves it based on the known set of dependencies. Edgar F. Codd, the inventor of the relational model (RM), introduced the concept of normalization and what is now known as the first normal form (1NF) in 1970. Codd went on to define the second normal form (2NF) and third normal form (3NF) in 1971 and Codd and Raymond F. Boyce defined the Boyce-Codd normal form (BCNF) in 1974.Informally, a relational database relation is often described as "normalized" if it meets third normal form. Most 3NF relations are free of insertion, update, and deletion anomalies. **First normal form** (**1NF**) is a property of relation in a relational database. A relation is in first normal form if and only if the domain of each attribute contains only atomic (indivisible) values, and the value of each attribute contains only a single value from that domain. First normal form is an essential property of a relation in a relational database. Database normalization is the process of representing a database in terms of relations in standard normal forms, where first normal is a minimal requirement.

Example.

cid	Name	Item_ name	Cost	Address	Phno
1	Praveen	Palav, pizza	RS280	BSK	9880077553

Table 3.1 Delivery _report

Table 3.1 is not in 1NF because 1NF says that every cell should have atomic values. In Item _name there are two values therefore to make it in 1NF both the values should be placed in different rows.

cid	Name	Item _name	Cost	Address	Phno
1	Praveen	Palav	Rs40	BSK	9880077554
1	Praveen	Pizza	Rs140	BSK	9880077554

Table 3.2 Delivery _report.

After normalization table 3.2 is in 1NF**.**

Second normal form (2NF) a relation is in second normal form if it is 1NF and every non key attribute is fully functionally dependent on the primary key.

Example

cid	Name	Item_ name	Cost	Address	Ph no
1	Praveen	Palav	Rs40	BSK	9880077554
1	Praveen	Pizza	Rs140	BSK	9880077554

Table 3.3 Delivery _report.

Table 3.3 has a composite primary key [cid, item_ name].The non-key attribute is[cost, address, ph no, name]. In this case[cost] only depends on [item_ name] ,which is only part of the primary key. Therefore, this table does not satisfy second normal form. To bring this table to second normal form, break the table into two tables.

cid	Name	Item_ name	Address	Ph no
1	Praveen	Palav	BSK	9880077554
1	Praveen	Pizza	BSK	9880077554

Table 3.4Ordering.

cid	Name	Item_ name	Cost	Address	Ph no
1	Praveen	Palav	Rs40	BSK	9880077554
1	Praveen	Pizza	Rs140	BSK	9880077554

Table 3.5 Delivery_ report.

Now in the table 3.5, the column [cost] is fully dependent on primary key of that table, which is [Item_ name].

Third normal form (3NF) is a normal form that is used in normalizing a database design to reduce the duplication of data and ensure referential integrity by ensuring that the entity is in second normal form, and all the attributes in a table are determined only by the candidate keys of that relation and not by any non-prime attributes. 3NF was designed to improve database processing while minimizing storage costs.

cid	Name	Item_ name	Cost	Address	Ph no
1	Praveen	Palav	Rs40	BSK	9880077554
1	Praveen	Pizza	Rs140	BSK	9880077554

Table 3.6 delivery_ report.

In the table 3.6[cid] determines [name],and determines [item _name]. Therefore,[cid] determines [name] via [item _name] and have transitive functional dependency, and this does not satisfy third normal form. To bring this table to third normal form, split the table into two as follows.

Cid	Name	Item _name	cost
1	Praveen	Palav	Rs40
1	Praveen	Pizza	Rs140

Table 3.7 Customer

Cid	Name	Item _name	Address	Ph no
1	Praveen	Palav	BSK	9880077554
1	Praveen	Pizza	BSK	9880077554

Table3.8 delivery _report.

Now all non-key attributes are functionally dependent only on the primary key.

CHAPTER 4

IMPLEMENTATION

4.1 TABLE CREATION

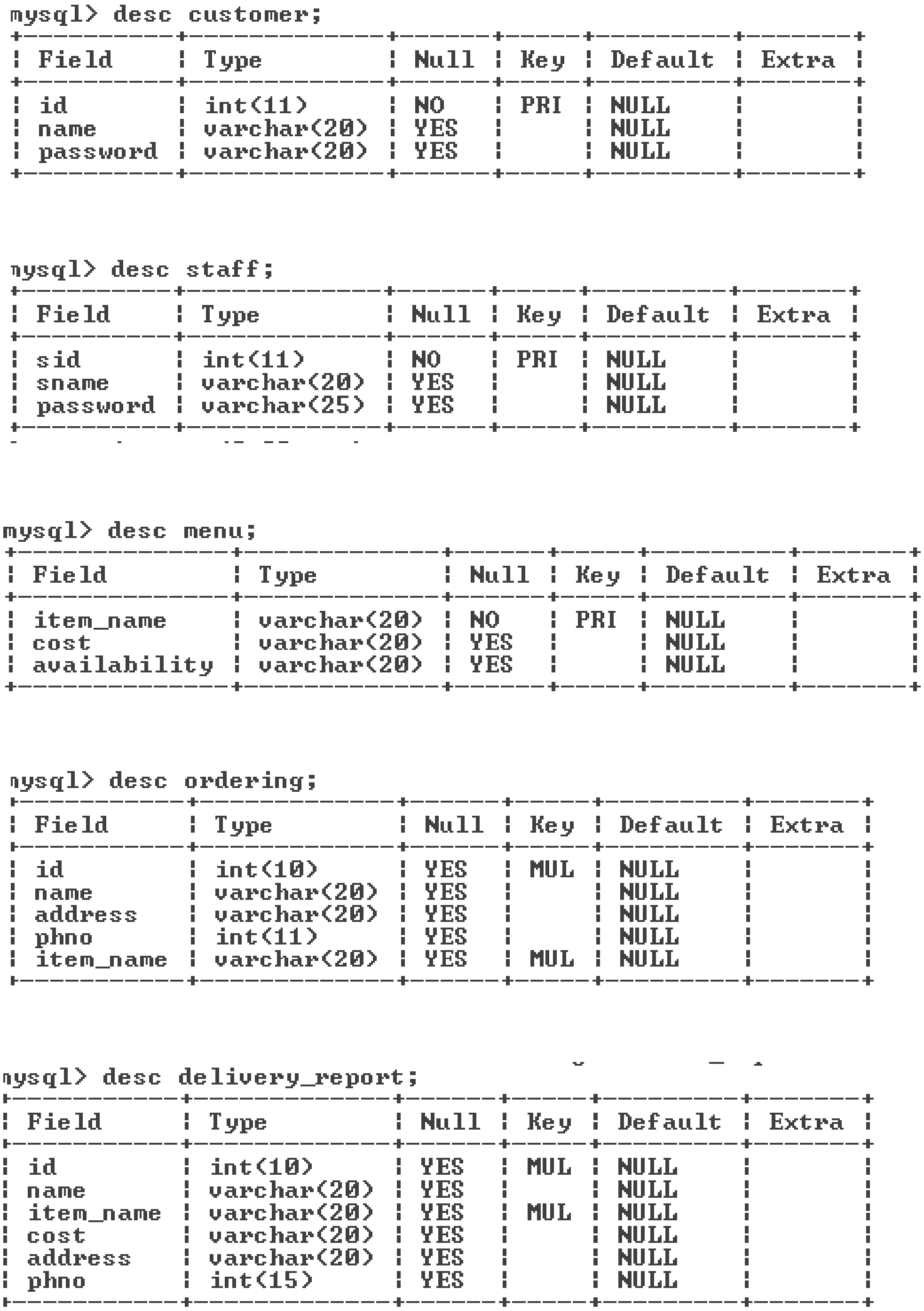
```
create table menu(
item _name varchar (20) primary key,
cost varchar (20),
availability varchar (20));

create  table customer(
id integer primary key,
name varchar (20) ,
password varchar (20));
create table staff(
sid integer primary key,
sname varchar(20),
password varchar(25));

create table ordering(
id int(10) primary key,
foreign key(id) references customer(id)on delete cascade,
name varchar(20),
address varchar(20),
phno integer,
item_name varchar(20),
foriegn key(item_name) references menu(item_name) on delete cascade);

create table delivery_report(
id int(10),
foreign key(id) references customer(id) on delete cascade,
name varchar(20),
item_name varchar(20),
```

foreign key(item_name) references menu(item_name) on delete cascade,

cost varchar(20),

address varchar(20),phno int(15));

4.2 DESCRIPTION OF TABLES



```
mysql> desc customer;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| id       | int(11)     | NO   | PRI | NULL    |       |
| name     | varchar(20) | YES  |     | NULL    |       |
| password | varchar(20) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

```
mysql> desc staff;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| sid      | int(11)     | NO   | PRI | NULL    |       |
| sname    | varchar(20) | YES  |     | NULL    |       |
| password | varchar(25) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

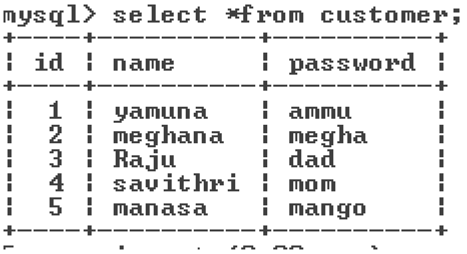
```
mysql> desc menu;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| item_name    | varchar(20) | NO   | PRI | NULL    |       |
| cost         | varchar(20) | YES  |     | NULL    |       |
| availability | varchar(20) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

```
mysql> desc ordering;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| id        | int(10)     | YES  | MUL | NULL    |       |
| name      | varchar(20) | YES  |     | NULL    |       |
| address   | varchar(20) | YES  |     | NULL    |       |
| phno      | int(11)     | YES  |     | NULL    |       |
| item_name | varchar(20) | YES  | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

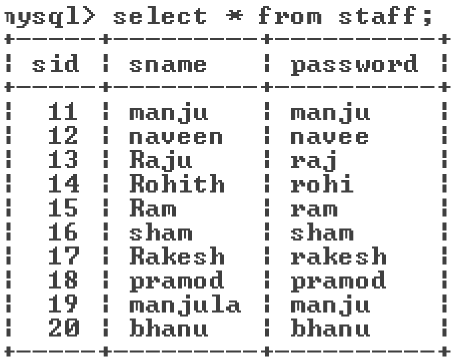
```
mysql> desc delivery_report;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| id        | int(10)     | YES  | MUL | NULL    |       |
| name      | varchar(20) | YES  |     | NULL    |       |
| item_name | varchar(20) | YES  | MUL | NULL    |       |
| cost      | varchar(20) | YES  |     | NULL    |       |
| address   | varchar(20) | YES  |     | NULL    |       |
| phno      | int(15)     | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

Figure 4.1 Description of tables

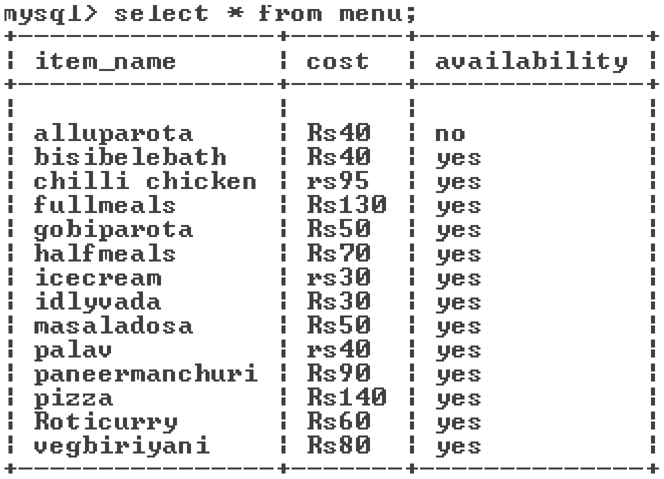
4.3 POPULATED TABLES



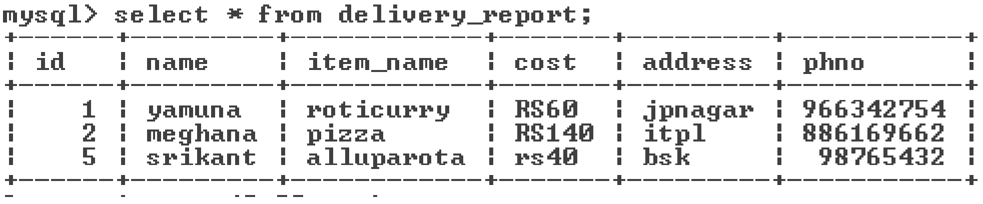
```
mysql> select *from customer;
+----+----------+----------+
¦ id ¦ name     ¦ password ¦
+----+----------+----------+
¦  1 ¦ yamuna   ¦ ammu     ¦
¦  2 ¦ meghana  ¦ megha    ¦
¦  3 ¦ Raju     ¦ dad      ¦
¦  4 ¦ savithri ¦ mom      ¦
¦  5 ¦ manasa   ¦ mango    ¦
+----+----------+----------+
```



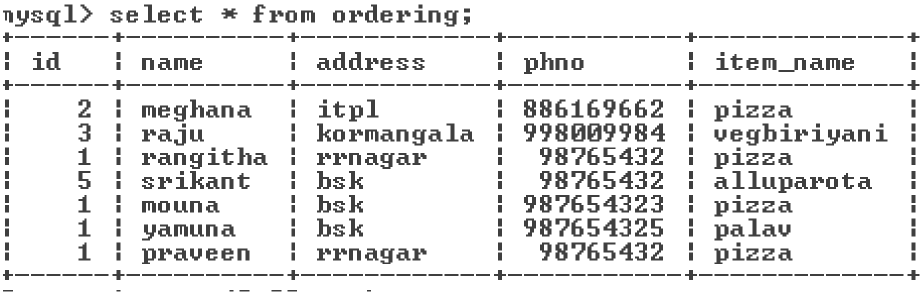
```
mysql> select * from staff;
+-----+---------+----------+
¦ sid ¦ sname   ¦ password ¦
+-----+---------+----------+
¦  11 ¦ manju   ¦ manju    ¦
¦  12 ¦ naveen  ¦ navee    ¦
¦  13 ¦ Raju    ¦ raj      ¦
¦  14 ¦ Rohith  ¦ rohi     ¦
¦  15 ¦ Ram     ¦ ram      ¦
¦  16 ¦ sham    ¦ sham     ¦
¦  17 ¦ Rakesh  ¦ rakesh   ¦
¦  18 ¦ pramod  ¦ pramod   ¦
¦  19 ¦ manjula ¦ manju    ¦
¦  20 ¦ bhanu   ¦ bhanu    ¦
+-----+---------+----------+
```



```
mysql> select * from menu;
+----------------+-------+--------------+
¦ item_name      ¦ cost  ¦ availability ¦
+----------------+-------+--------------+
¦                ¦       ¦              ¦
¦ alluparota     ¦ Rs40  ¦ no           ¦
¦ bisibelebath   ¦ Rs40  ¦ yes          ¦
¦ chilli chicken ¦ rs95  ¦ yes          ¦
¦ fullmeals      ¦ Rs130 ¦ yes          ¦
¦ gobiparota     ¦ Rs50  ¦ yes          ¦
¦ halfmeals      ¦ Rs70  ¦ yes          ¦
¦ icecream       ¦ rs30  ¦ yes          ¦
¦ idlyvada       ¦ Rs30  ¦ yes          ¦
¦ masaladosa     ¦ Rs50  ¦ yes          ¦
¦ palav          ¦ rs40  ¦ yes          ¦
¦ paneermanchuri ¦ Rs90  ¦ yes          ¦
¦ pizza          ¦ Rs140 ¦ yes          ¦
¦ Roticurry      ¦ Rs60  ¦ yes          ¦
¦ vegbiriyani    ¦ Rs80  ¦ yes          ¦
+----------------+-------+--------------+
```



```
mysql> select * from delivery_report;
+------+---------+------------+-------+---------+-----------+
¦ id   ¦ name    ¦ item_name  ¦ cost  ¦ address ¦ phno      ¦
+------+---------+------------+-------+---------+-----------+
¦    1 ¦ yamuna  ¦ roticurry  ¦ RS60  ¦ jpnagar ¦ 966342754 ¦
¦    2 ¦ meghana ¦ pizza      ¦ RS140 ¦ itpl    ¦ 886169662 ¦
¦    5 ¦ srikant ¦ alluparota ¦ rs40  ¦ bsk     ¦  98765432 ¦
+------+---------+------------+-------+---------+-----------+
```



```
nysql> select * from ordering;
+------+----------+------------+-----------+-------------+
| id   | name     | address    | phno      | item_name   |
+------+----------+------------+-----------+-------------+
|    2 | meghana  | itpl       | 886169662 | pizza       |
|    3 | raju     | kormangala | 998009984 | vegbiriyani |
|    1 | rangitha | rrnagar    |  98765432 | pizza       |
|    5 | srikant  | bsk        |  98765432 | alluparota  |
|    1 | mouna    | bsk        | 987654323 | pizza       |
|    1 | yamuna   | bsk        | 987654325 | palav       |
|    1 | praveen  | rrnagar    |  98765432 | pizza       |
+------+----------+------------+-----------+-------------+
```

Figure 4.2 Populated tables

4.3 SQL Triggers & Stored Procedures

SQL Triggers

To create and drop a trigger ,we use the CREATE TRIGGER and DROP TRIGGER statements. The trigger acts as an accumulator, summing the values inserted into one of the columns of the table.

Mysql>Create TRIGGER T1 BEFORE INSERT ON occupation

FOR EACH ROW SET @cost = @cost+ NEW.total;

Query ok, 0 rows affected (0.01 sec)

The CREATE TRIGGER statement creates a trigger named T1 that is associated with the account table. It also includes clauses that specify the trigger action time, the triggering event, and what to do when the trigger activates:

The keyword BEFORE indicates the trigger action time. In this case, the trigger activates before each row inserted into the table. The other permitted keyword here is AFTER.

The keyword INSERT indicates the trigger event; that is, the type of operation that activates the trigger. In the example, INSERT operations cause trigger activation. You can also create triggers for DELETE and UPDATE operations.

The statement following FOR EACH ROW defines the trigger body; that is, the statement to execute each time the trigger activates, which occurs once for each row affected by the triggering event. In the example, the trigger body is a simple SET that accumulates into a user variable the values inserted into the amount column. The statement refers to the column as NEW. total which means “the value of the amount column to be inserted into the new row.” To use the trigger, set the accumulator variable to zero, execute an INSERT statement, and then see what value the variable has afterward:

mysql> SET @cost = 0;

```
mysql>INSERT INTOdelivery_reportsVALUES(1‘yamuna’,pizza’,’rs140’,’rrnagar’,9663427554,);
mysql> SELECT @cost AS 'Total amount inserted';
Mysql>show triggers;
1 row in set (0.06 sec)
```

To destroy the trigger, we use a DROP TRIGGER statement. We must specify the schema name if the trigger is not in the default schema:

```
mysql> DROP TRIGGER delivery_report .varchar_cost;
```

If you drop a table, any triggers for the table are also dropped.

4.3 Stored Procedures

After creating database I have created stored procedure for select in the front end. With the help of stored procedure access should be fast and we don’t have need to compile it , can run directly. I have created stored procedure inserting all data of the table with value type and then I have created query for select in their procedure.

Source Code

```
public class storedprocedure {

public static void main(String[] args){
try{
Class.forName("com.mysql.jdbc.Driver");
Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/foodordering","root","root123");
CallableStatement stmt = con.prepareCall("call disp_ordering()");
ResultSet resul = stmt.executeQuery();
while(resul.next()){
System.out.println(resul.getInt(1)+"\t" + resul.getString(2));
JOptionPane.showMessageDialog(null, "procedure created");
}catch(Exception e)
{
JOptionPane.showMessageDialog(null,e);
                              }
}
```

OUTPUT

```
3       raju
1       rangitha
5       srikant
```

can call the procedure any number of times.

4.4 Database Connectivity

Connection made using jdbc connector

JDBC API is a Java API that can access any kind of tabular data, especially data stored in a Relational Database. JDBC works with Java on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. The JDBC API is a Java API that can access any kind of tabular data, especially data stored in a Relational Database. JDBC helps to write Java applications that manage these three programming activities:

1. Connect to a data source, like a database
2. Send queries and update statements to the database
3. Retrieve and process the results received from the database in answer to the query.

The interface for accessing relational databases from Java is Java Database Connectivity (JDBC). Via JDBC we can create a connection to the database, issue database queries and update as well as receive the results. JDBC provides an interface which allows you to perform SQL operations independently of the instance of the used database. To use JDBC, you require the database specific implementation of the JDBC driver

Source Code

```
public class OracleConnection {
public static  void Connection(){
try {Class.forName("com.mysql.jdbc.Driver");
} catch(ClassNotFoundException e){
e.printStackTrace();}}
public static void ConnectionToMySql(){
Connection();
String host ="jdbc:mysql://localhost/foodordering";
String username = "root" ;
String password = "";
try {Connection connect =DriverManager.getConnection(host,"root","root123");
JOptionPane.showMessageDialog(null, "connection successfull");
}catch (SQLException e){e.printStackTrace();}
catch(Exception e) {
JOptionPane.showMessageDialog(null, e);}}public static void main(String
args[]){ConnectionToMySql();}}
```

CHAPTER 5

RESULTS

GUI SCREENSHOTS:

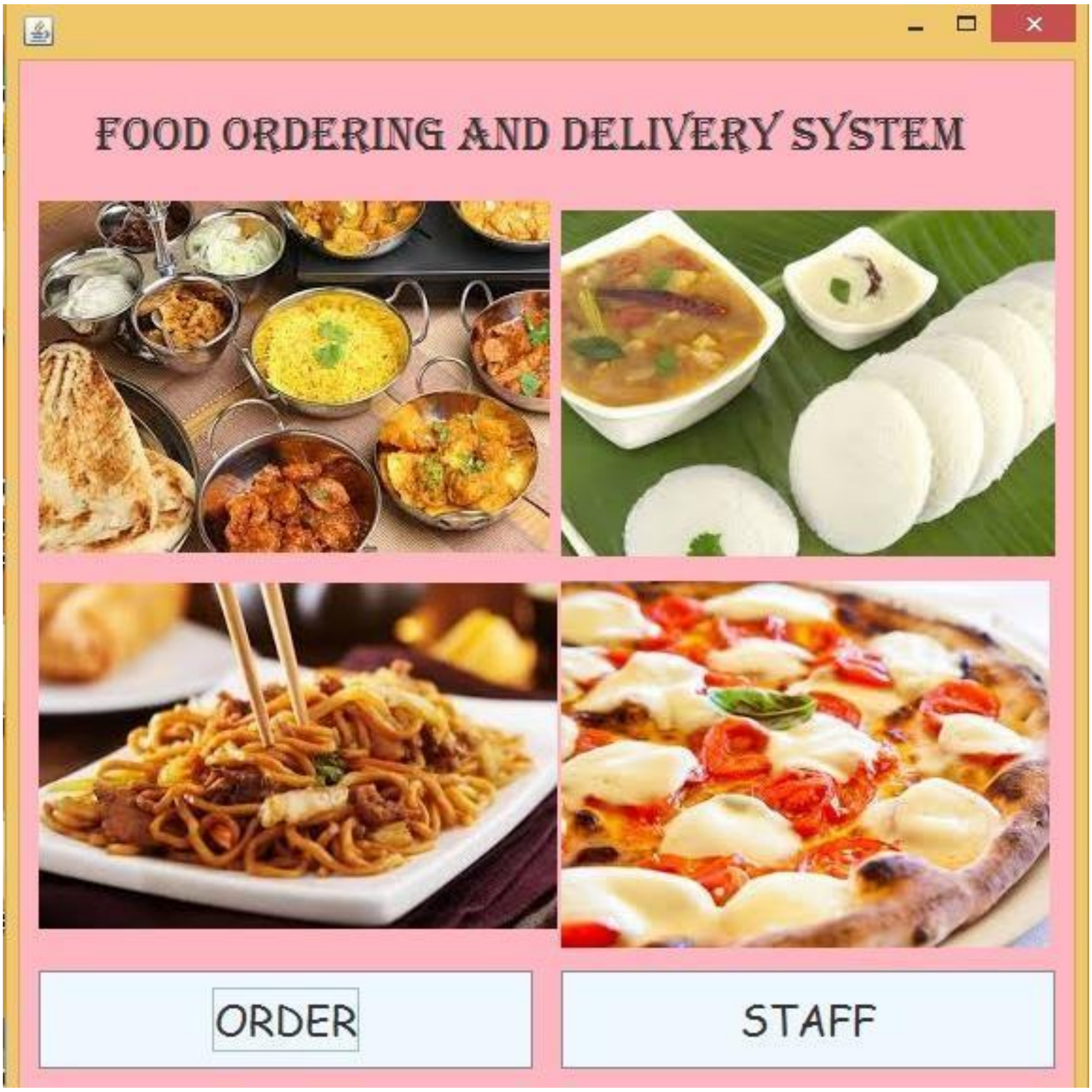
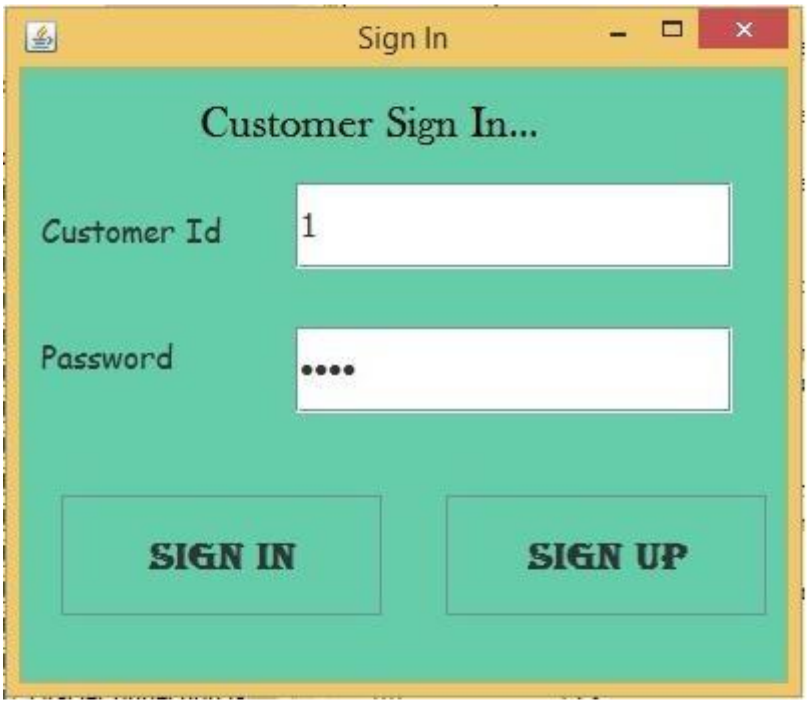


Figure 5.1: Login page for placing order

Figure 5.1 helps the user to place an order by signing into there account and also allows the staff to sign in into there account to check orders and to view the mean and also to update menu for the customer.

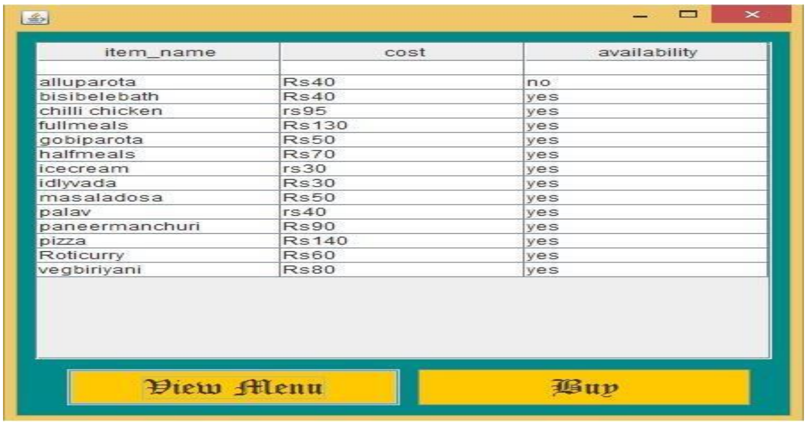


Customer Id 1

Password ••••

Figure 5.2: Customer sign in page

Figure 5.2 allows the customer to sign into their account to place their order.



item_name	cost	availability
alluparota	Rs40	no
bisibelebath	Rs40	yes
chilli chicken	rs95	yes
fullmeals	Rs130	yes
gobiparota	Rs50	yes
halfmeals	Rs70	yes
icecream	rs30	yes
idlyvada	Rs30	yes
masaladosa	Rs50	yes
palav	rs40	yes
paneermanchuri	Rs90	yes
pizza	Rs140	yes
Roticurry	Rs60	yes
vegbiriyani	Rs80	yes

Figure 5.3: Menu view page

Figure 5.3 allows the customer to view the menu and by looking at the menu they can place there order.

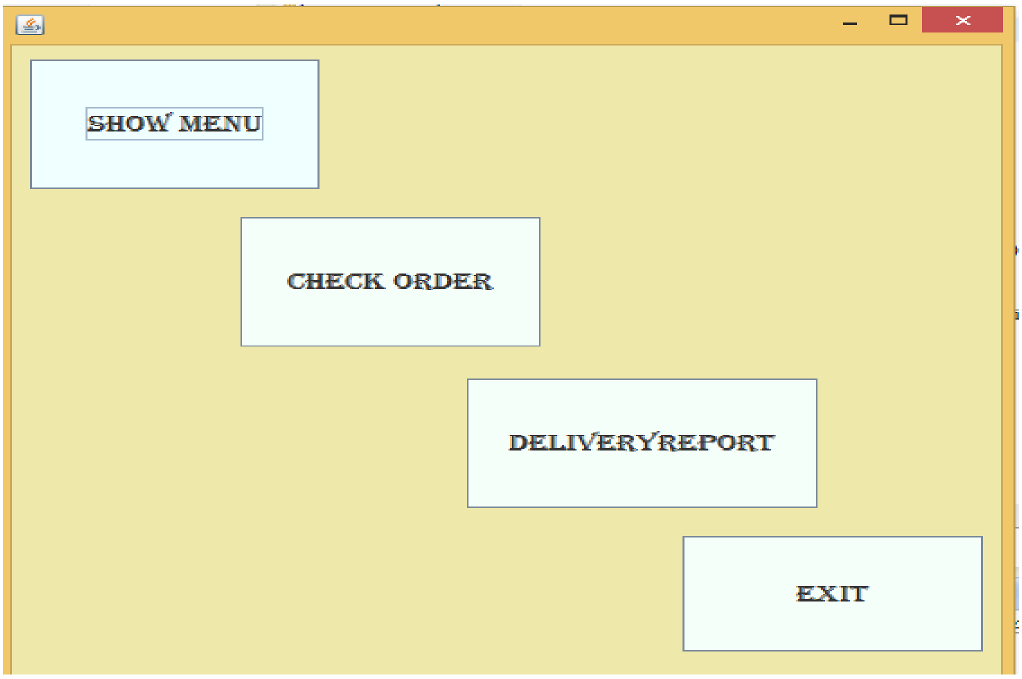
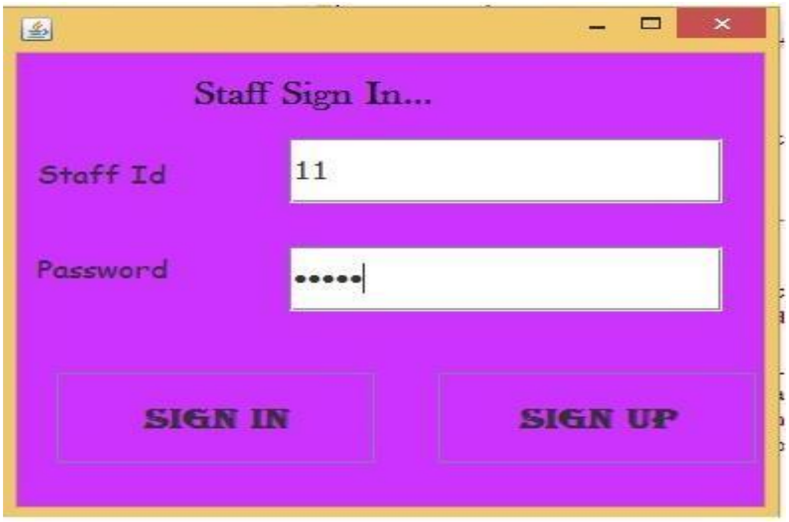


Figure 5.4: Staff view page.

Figure 5.4 allows the staff to view the menu and to check orders and also to view the supply details.

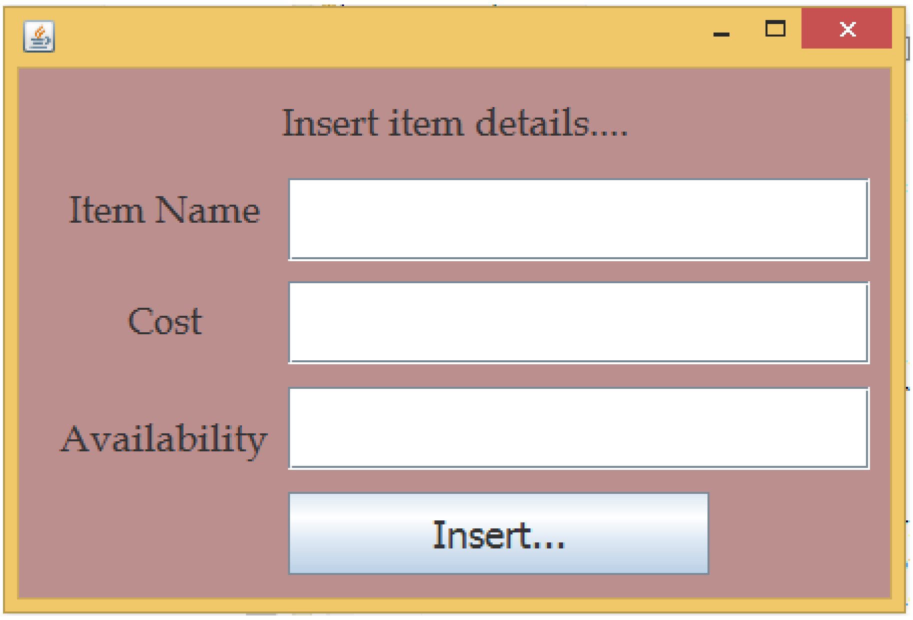

Staff Sign In...

Staff Id 11

Password •••••

Figure 5.5 Staff Sign page.

Figure 5.5 allows the staff to login into their account to modify or to add additional features to it..



Insert item details....

Item Name

Cost

Availability

Figure 5.6 Insert page.

Figure 5.6 allows the staff to add menu to the menu table**.**

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION

This project has been a lot of fun. It has enhanced our thinking skills in new and innovative ways about database management systems. The project implements food ordering and delivery system website to order for a food. The project serves as a purpose for the user to search for a food by entering his or her details by entering into the website all of which were designed using various tools and softwares such as eclipse JDBC connector, MYSql Command Line Client etc. It can be further improved upon to provide better facilities in display and user interface. This application has been developed on windows platform.

6.2 FUTURE ENHANCEMENTS

This project can be further improved upon to provide better facilities in display and user interface.

Some of the future enhancements that can be done on this project are:

1. It is possible to provide the web space to the users for creating his portal.
2. Options to add more number of fields to the website.
3. It is possible to create a chat server so that user can communicate with each other.
4. It is possible to create mail server.
5. Online food ordering specific facilities like payment methods can be implemented to provide payment options such as PAYTM, Netbanking, credit and debit card options, etc.