RESTAURANT MANAGEMENT

Description

Our project is restaurant management system, where we provide all the details that a manager needs to run the restaurant from ordering the food to generating bill and payment. In our project only a manager have authorization.

We have table customers for customer information, menu for list of food, tables through which a customer is assigned a table, orders and waiter table to take and manage orders. We also manage no. of people which are together to generate a common bill for all food ordered by them.

We allocate table based on vacancy status and an unoccupied waiter does that job for serving food to the customers. After a customer entered their information then we display menu for them to order.

Also, we keep a record of the waiters, like name, gender, salary etc. Menu is displayed with name of the dish and its price. We have unique dish ID to uniquely manage the orders.

And we generate bill automatically based on their order.

USER STORY

Restaurant Manager

- Manager should be able to update information about restaurant.
- Manager should be able to view info about waiters and customers.

Customer

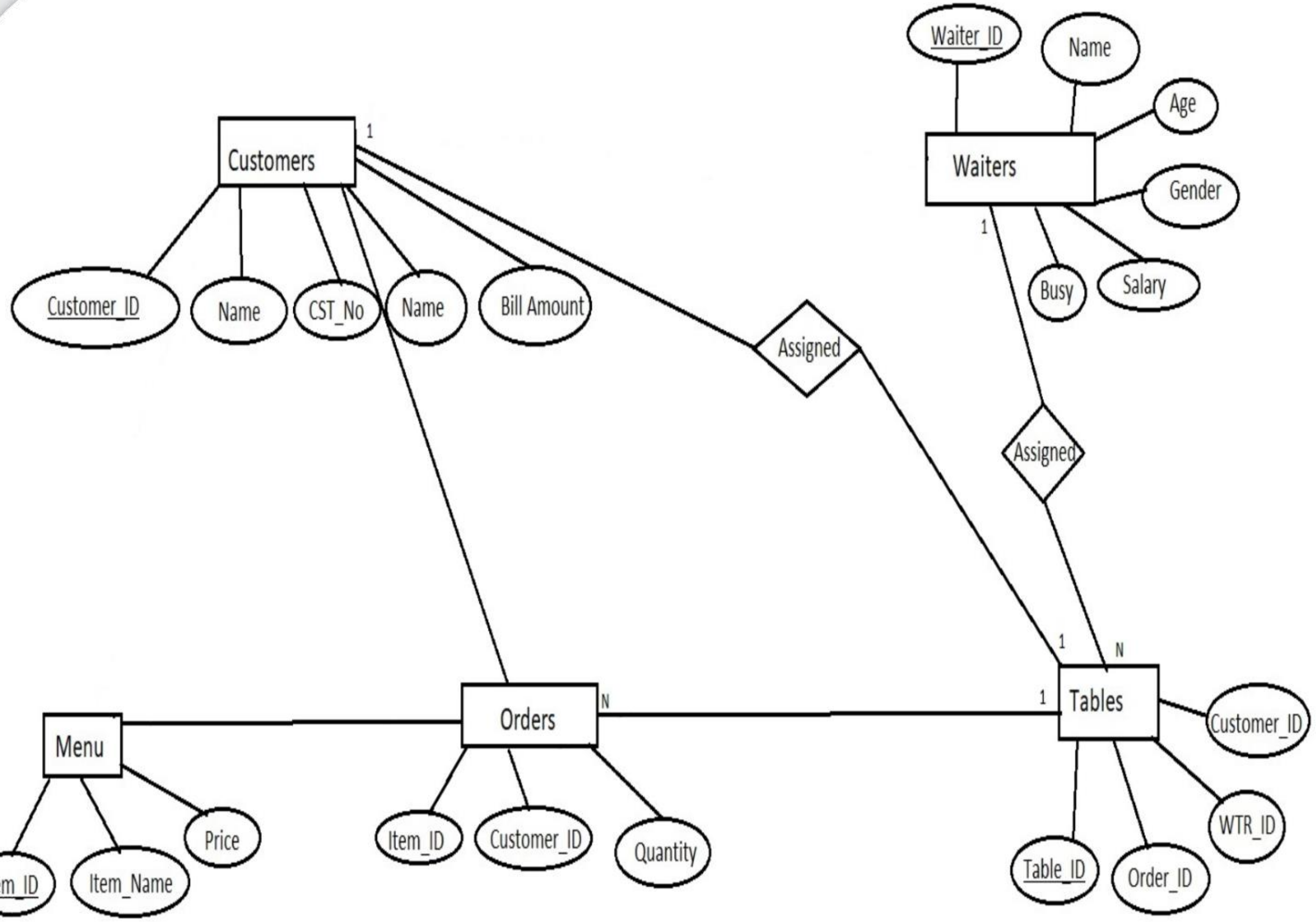
- Customers should be able to browse through the menu and look at the various food options available in the restaurant along with the price for each item.
- Customer should be able to select items from the menu and add them to order.

Waiter

- Waiters should be able to view the current queue to see the ready orders for pickup
- Waiter should be able to serve orders to customers assigned to them

E-R DIAGRAM

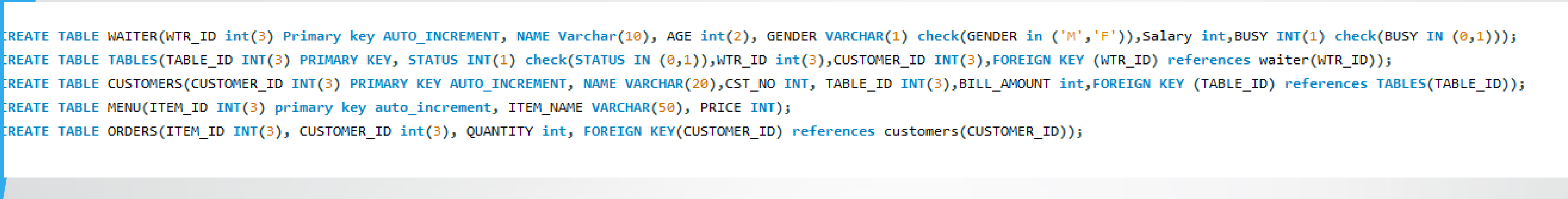
- An E-R diagram is used to represent the E-R model. It contains all known entities for the given scenario, their attributes, identifiers, and the relationships that exist among the entities.
- As seen in figure the ER diagram has five entities which are made up of various attributes including the entity's primary key and other attributes like the name, salary, age etc.



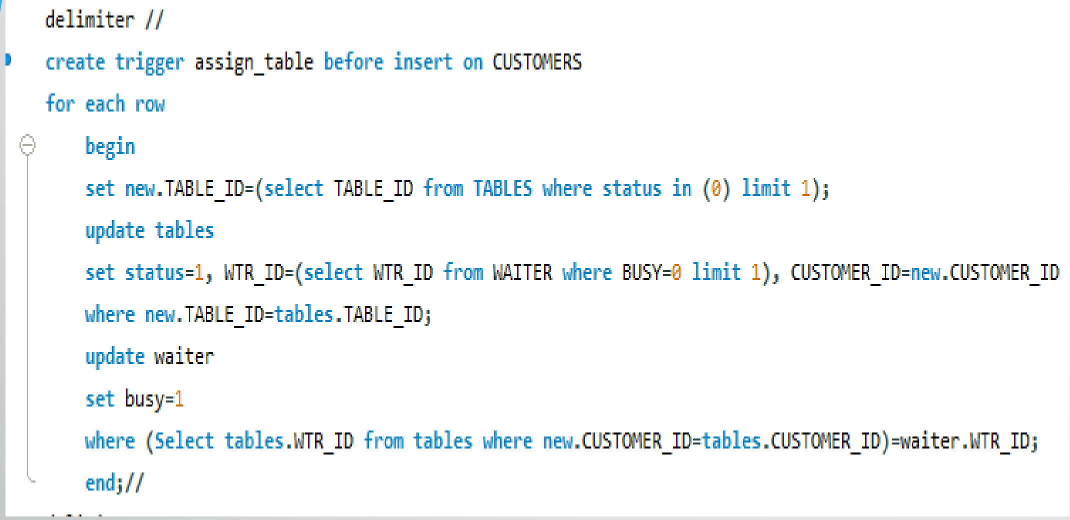
Database Schema and Tables

- WAITER Table: -Waiter_ID attribute is a primary key to uniquely determine a waiter with Name, Age, Gender, Salary, Busy (with check constraint) as other attributes.
- TABLES Table: - TABLE_ID attribute as primary key with Order_ID , WTR_ID for waiter ID, Customer_ID to determine customer on a particular table and , (WTR_ID) as FOREIGN KEY references (WTR_ID) of table waiter.
- ORDERS Table: - ITEM_ID, Customer_ID , Quantity for quantity of items and (CUSTOMER_ID) FOREIGN KEY references (CUSTOMER_ID) of table customers.
- CUSTOMERS Table: - CUSTOMER_ID as PRIMARY KEY to uniquely determine a customer, NAME CST_NO to determine the number of people along, TABLE_ID, BILL_AMOUNT containing total bill amount for customers and (TABLE_ID) as FOREIGN KEY references (TABLE_ID) of table TABLES.
- MENU Table: - ITEM_ID as primary key to uniquely determine Item from menu , ITEM_NAME, PRICE as other attributes.

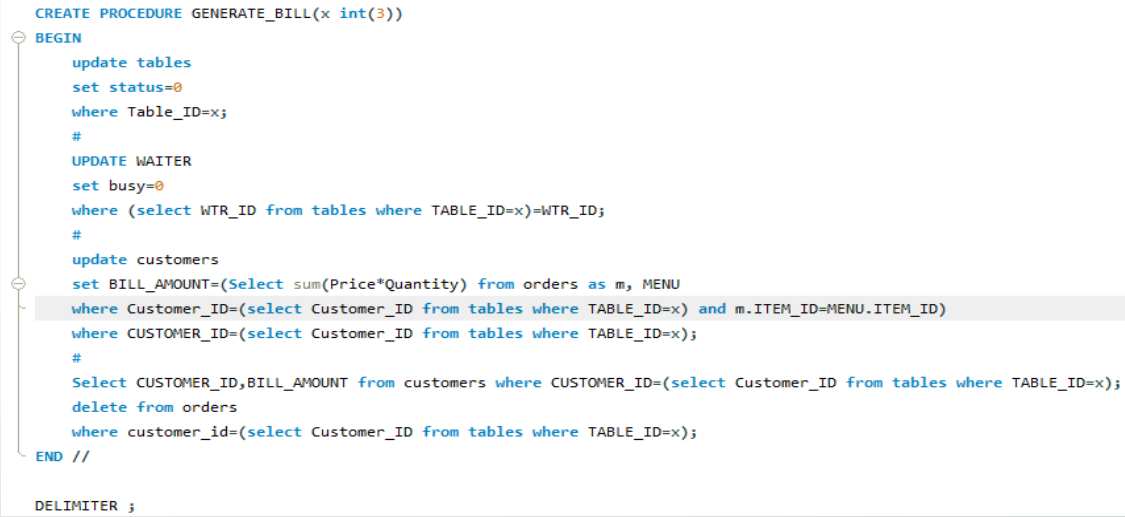
The Creation of database is shown below, which shows the tables created with the necessary constraints applied.



```sql
CREATE TABLE WAITER(WTR_ID int(3) Primary key AUTO_INCREMENT, NAME Varchar(10), AGE int(2), GENDER VARCHAR(1) check(GENDER in ('M','F')),Salary int,BUSY INT(1) check(BUSY IN (0,1)));
CREATE TABLE TABLES(TABLE_ID INT(3) PRIMARY KEY, STATUS INT(1) check(STATUS IN (0,1)),WTR_ID int(3),CUSTOMER_ID INT(3),FOREIGN KEY (WTR_ID) references waiter(WTR_ID));
CREATE TABLE CUSTOMERS(CUSTOMER_ID INT(3) PRIMARY KEY AUTO_INCREMENT, NAME VARCHAR(20),CST_NO INT, TABLE_ID INT(3),BILL_AMOUNT int,FOREIGN KEY (TABLE_ID) references TABLES(TABLE_ID));
CREATE TABLE MENU(ITEM_ID INT(3) primary key auto_increment, ITEM_NAME VARCHAR(50), PRICE INT);
CREATE TABLE ORDERS(ITEM_ID INT(3), CUSTOMER_ID int(3), QUANTITY int, FOREIGN KEY(CUSTOMER_ID) references customers(CUSTOMER_ID));
```



```sql
delimiter //
create trigger assign_table before insert on CUSTOMERS
for each row
    begin
    set new.TABLE_ID=(select TABLE_ID from TABLES where status in (0) limit 1);
    update tables
    set status=1, WTR_ID=(select WTR_ID from WAITER where BUSY=0 limit 1), CUSTOMER_ID=new.CUSTOMER_ID
    where new.TABLE_ID=tables.TABLE_ID;
    update waiter
    set busy=1
    where (Select tables.WTR_ID from tables where new.CUSTOMER_ID=tables.CUSTOMER_ID)=waiter.WTR_ID;
    end;//
```



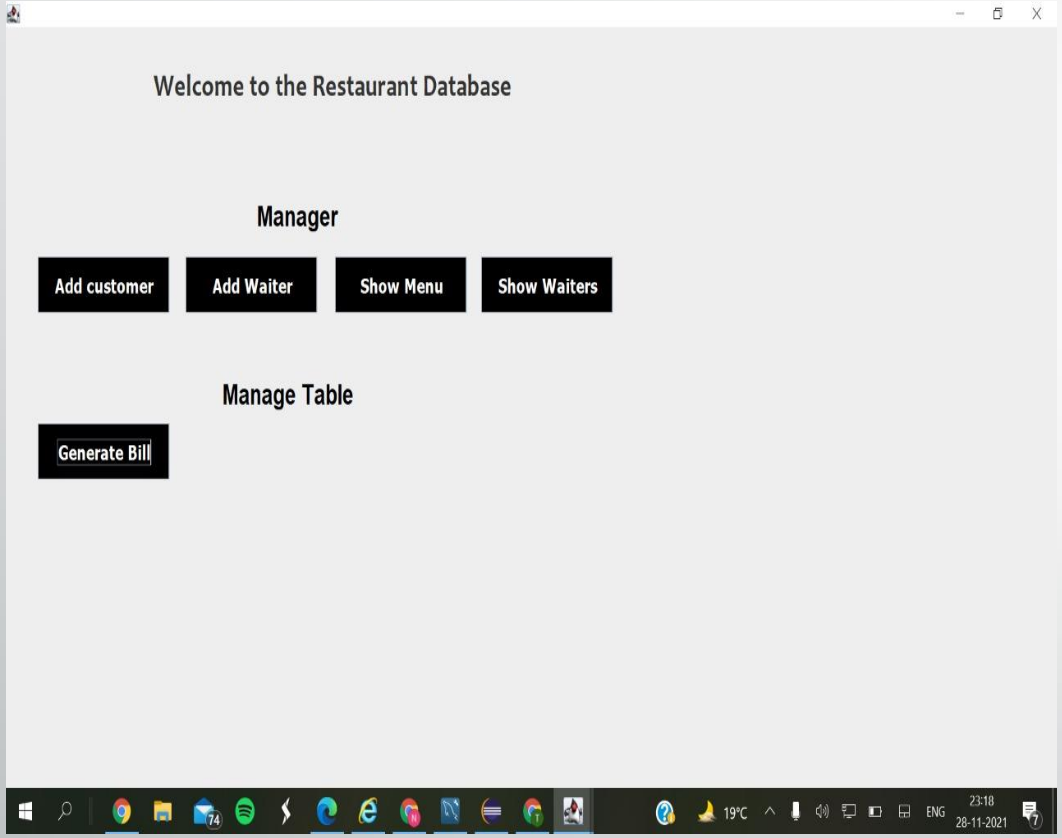
```sql
CREATE PROCEDURE GENERATE_BILL(x int(3))
BEGIN
    update tables
    set status=0
    where Table_ID=x;
    #
    UPDATE WAITER
    set busy=0
    where (select WTR_ID from tables where TABLE_ID=x)=WTR_ID;
    #
    update customers
    set BILL_AMOUNT=(Select sum(Price*Quantity) from orders as m, MENU
    where Customer_ID=(select Customer_ID from tables where TABLE_ID=x) and m.ITEM_ID=MENU.ITEM_ID)
    where CUSTOMER_ID=(select Customer_ID from tables where TABLE_ID=x);
    #
    Select CUSTOMER_ID,BILL_AMOUNT from customers where CUSTOMER_ID=(select Customer_ID from tables where TABLE_ID=x);
    delete from orders
    where customer_id=(select Customer_ID from tables where TABLE_ID=x);
END //

DELIMITER ;
```

Trigger *assign_table* is created to assign table to the new customer and assign an unoccupied waiter to serve.

Procedure *GENERATE_BILL* is created to generate the bill for the customers order after ordering the food from the menu.

GUI SCREEN DISPLAY

Database Home Page



Welcome to the Restaurant Database

Manager

Manage Table

Table Information Screen

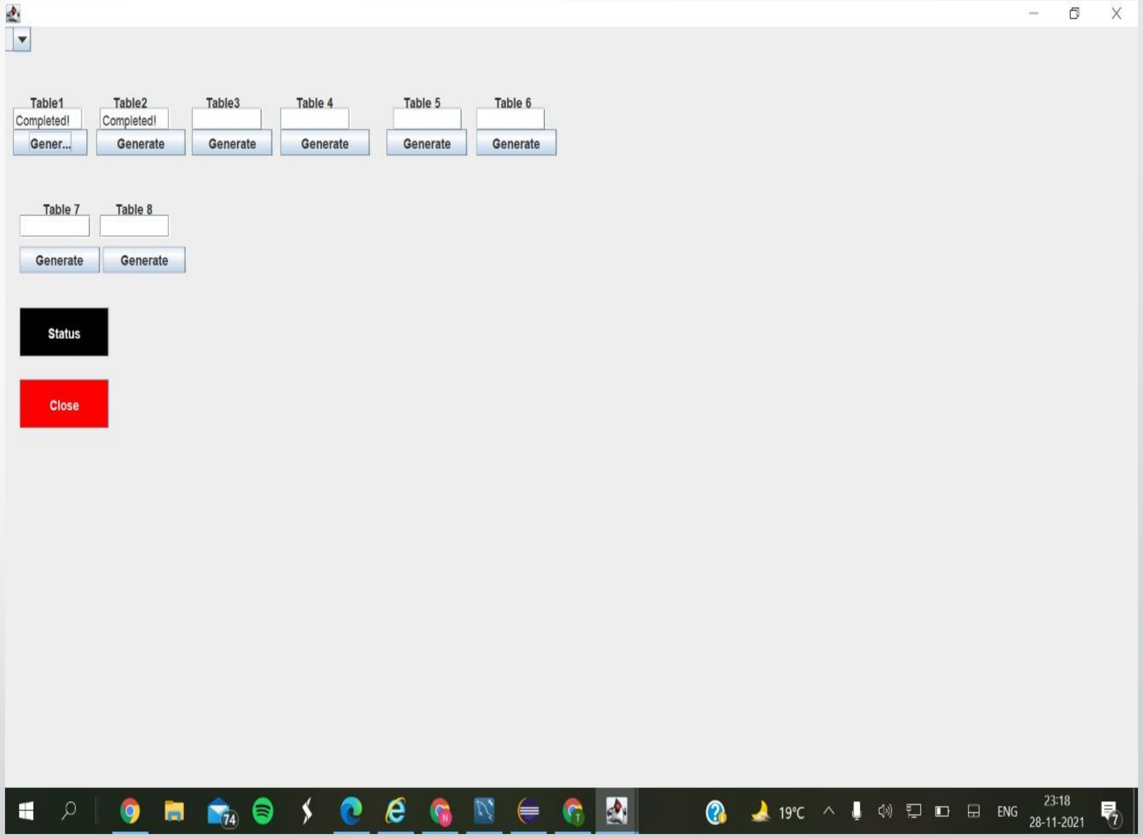
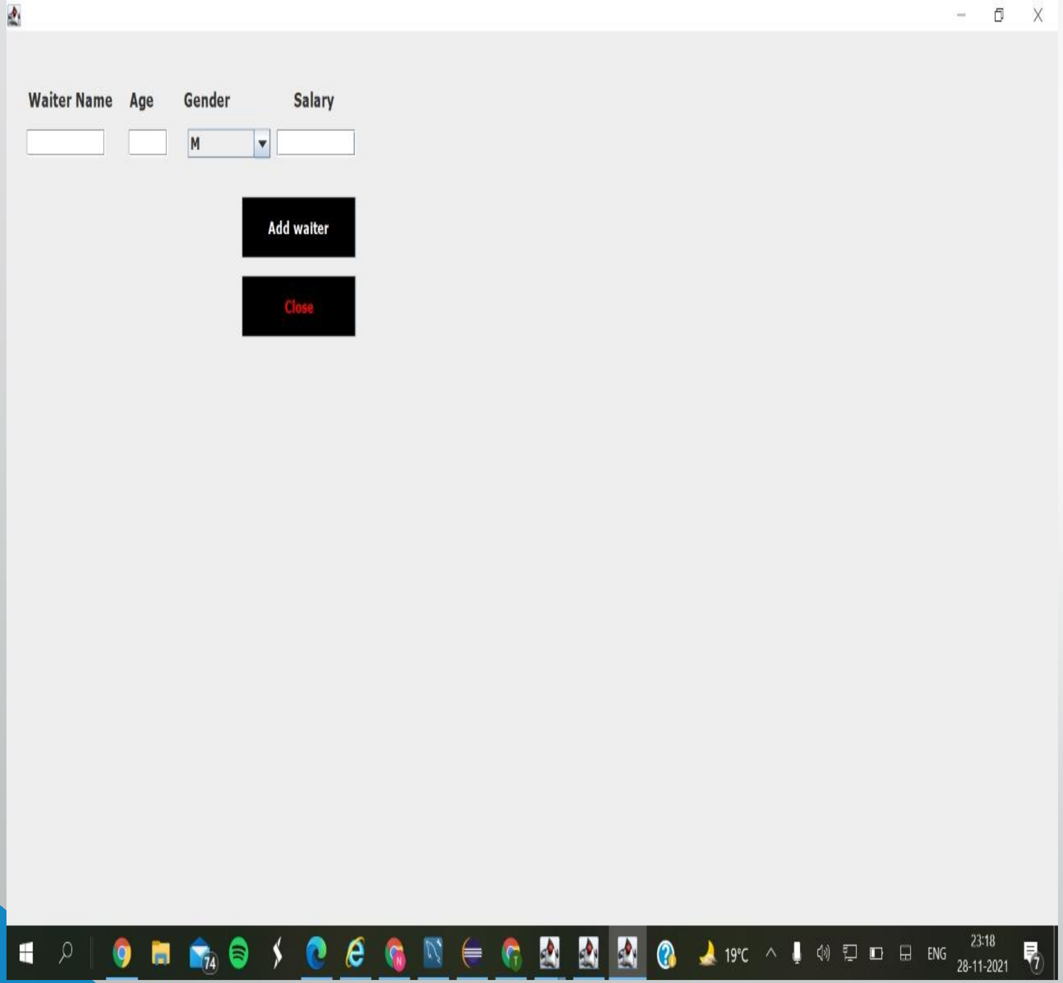


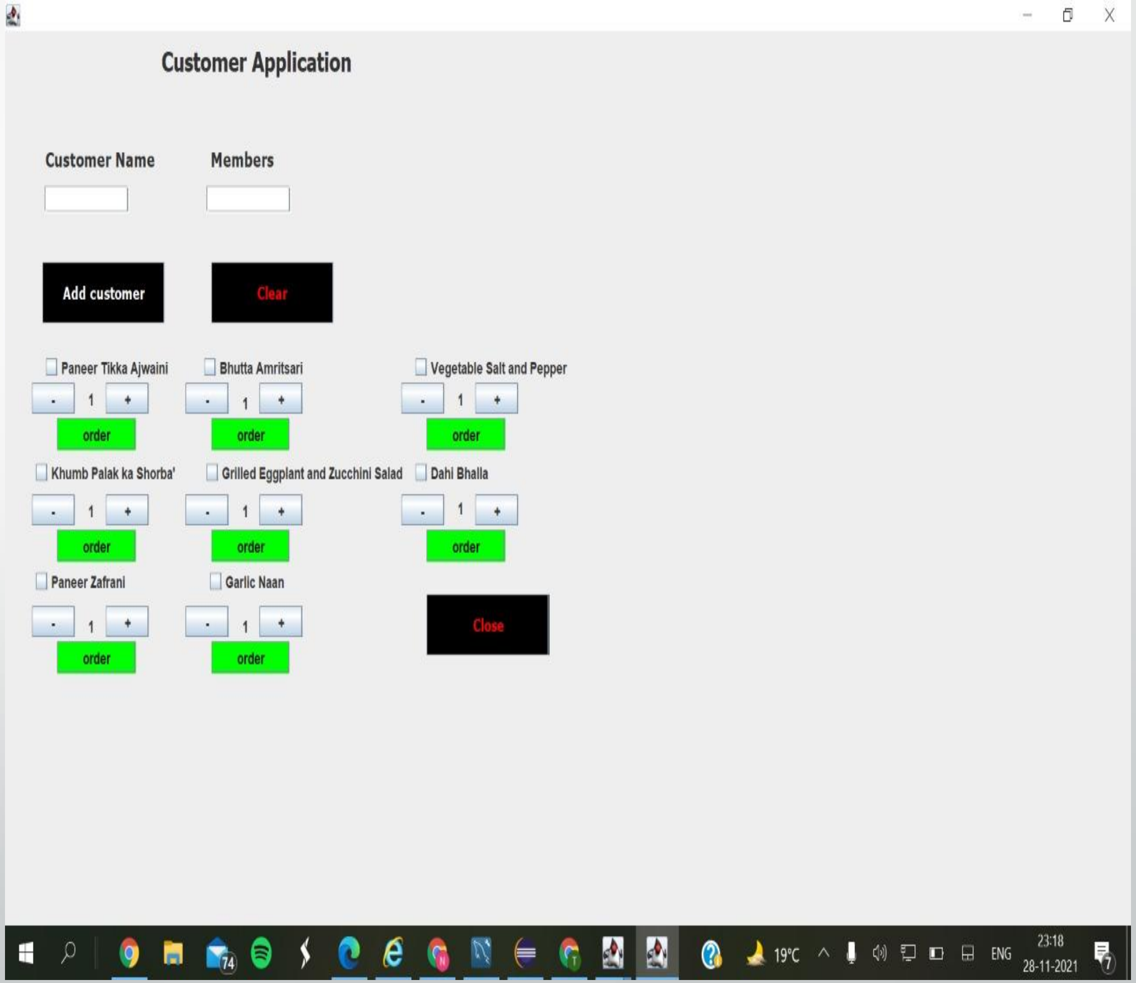
Table1	Table2	Table3	Table 4	Table 5	Table 6
Completed!	Completed!				

GUI SCREEN DISPLAY

For Adding WAITER



For Customer Application



Java Classes And Applications Used are:

- **Java database connectivity** (JDBC) is the JavaSoft specification of a standard application programming interface (API) that allows Java programs to access database management systems.
- GUI (**Graphical User Interface**) in Java is an easy-to-use visual experience builder for Java applications.
- DBtablePrinter CLASS
- MySQL

GROUP NO. 12
TEAM MEMBERS

20UCS111	Manav Jangid
20UCS122	Jain Naman Ramesh
20UCS125	Naman Jain
20UCS149	Priyadarshini Sankar