CSCI370: Database Application Project

Sami Al-Qusus

April 16, 2018

VIU

Please Note:

- You can view the webpage on VIU's network not via wifi with this link:
  `wwwstu.csci.viu.ca/~alqususs/project/index.php`
- If you wish to have access to mysql database, I'll be happy to help.
- Please feel free to email me for any clarification, at sami.qusus@gmail.com
- Thank you for a great semester, Sami!

Introduction

For this project, I built a database application that will help small restaurant owners manage different aspects of their restaurants by installing this database application at each of their restaurants. The application will help managers and servers with managing orders and processing transactions. It will help owners keep track of different branch operations and necessary data on a larger scale.

The intended audience is the owner of one or more restaurants, the branch manager and servers at different braches. The customers are not direct users of the system, but are indirectly involved. Although customers will not interact directly with the system, the functionality of generating a receipt that will most probably be printed out and provided to them shows that although they might not be directly involved its important to think about what data they will want to see on the receipt. For example, having the branch id on the receipt without the branch address might be sufficient for the owner, managers, and servers, but not for the customers. Therefore, for this reason they were also considered when designing certain aspects of this system.

Application Requirements and Functionalities

For this project I am assuming that the owner has a database administrator that maintains all the overall systems. Therefore, certain aspects and functionalities, like adding a new server or a new branch and updating branch or server information has been left out for the database administrator to do manually do. With this, we focus on the following functions that our application provides to the different users of the system. The application helps create orders and store the data to manage the orders for different tables throughout the day. It keeps track of the tables each server is serving and it generates a receipt at the end with details of that order. All order details are stored in the database once the order is created and a receipt is generated. The manager or owner can edit or update orders when necessary by providing a password, which help maintain the integrity of the system.

1. Create an order.
2. View details of a specific order.
3. Edit an order after it has been submitted after providing a password.
4. Calculate subtotal, calculate tax on subtotal, calculate total by adding subtotal and tax.
5. Search for a specific order by its id.

Database Design

For each order, to create the order we collect:

- Server's ID
- Note (for allergies or certain requests)
- Dish name and price (for each dish ordered)
- Drink name and price (for each drink ordered)
- Table number

For making changes or updates to an order we collect:

- Password (to access the edit page)

Possibly collect any of the following to be updated:

- Server's ID
- Note (for allergies or certain requests)
- Dish name and price (for each dish ordered)
- Drink name and price (for each drink ordered)
- Table number

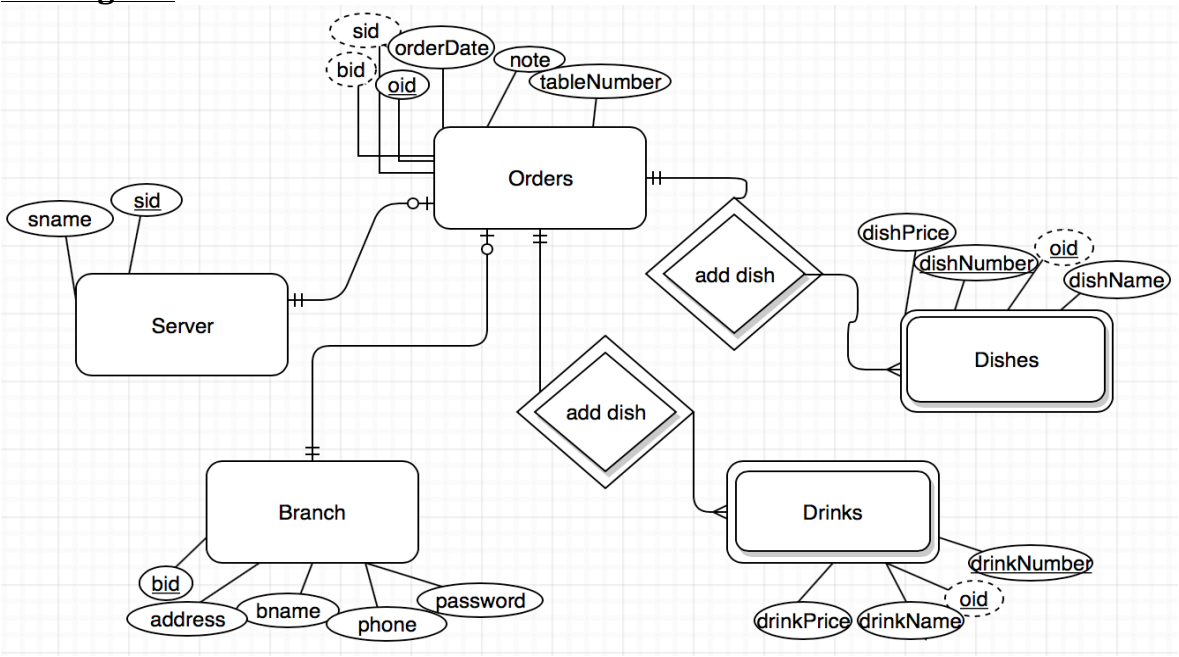
For searching for a specific order we collect:

- Order ID

*Everything else is inserted/updated/removed by database administrator.

- Branch Info
- Servers Info
- Managers Password

ER Diagram



Database schema *(mysql statements)*

```
CREATE TABLE Server (
sid INT(4) AUTO_INCREMENT NOT NULL,
sname VARCHAR(20) NOT NULL,
PRIMARY KEY (sid)
);

CREATE TABLE Branch(
bid INT(4) AUTO_INCREMENT NOT NULL,
bname VARCHAR(20) NOT NULL,
phone VARCHAR(10),
address VARCHAR(200) NOT NULL,
password VARCHAR(30) NOT NULL,
PRIMARY KEY (bid)
);

CREATE TABLE Orders (
oid INT(4) AUTO_INCREMENT NOT NULL,
bid INT(4) DEFAULT '0001',
sid INT(4) NOT NULL,
tableNumber INT(2),
orderDate DATETIME DEFAULT CURRENT_TIMESTAMP,
note VARCHAR(200),
PRIMARY KEY (oid),
FOREIGN KEY (bid) REFERENCES Branch(bid),
FOREIGN KEY (sid) REFERENCES Server(sid)
);

CREATE TABLE Dishes (
dishNumber INT(2) NOT NULL,
oid INT(4) NOT NULL,
dishName VARCHAR(20) NOT NULL,
dishPrice DECIMAL(4, 2) NOT NULL,
PRIMARY KEY (dishNumber, oid),
FOREIGN KEY (oid) REFERENCES Orders(oid)
);

CREATE TABLE Drinks (
drinkNumber INT(2) NOT NULL,
oid INT(4) NOT NULL,
drinkName VARCHAR(20) NOT NULL,
drinkPrice DECIMAL(4, 2)  NOT NULL,
PRIMARY KEY (drinkNumber, oid),
FOREIGN KEY (oid) REFERENCES Orders(oid)
);
```

Sample Data

Orders

oid	bid	sid	tableNumber	orderDate	note
1	1	1	1	2018-04-16 19:38:31	No ice in coke
2	1	2	2	2018-04-16 19:41:11	
3	1	1	3	2018-04-16 19:43:52	no garlic sauce in shawarma
4	1	3	4	2018-04-16 19:47:19	
5	1	2	1	2018-04-16 19:49:55	make sure steak is well done
6	1	1	5	2018-04-16 19:51:59	Extra Black Pepper

Dishes

dishNumber	oid	dishName	dishPrice
1	1	Cheese Burger	13.00
1	2	spaghetti and meatba	23.00
1	3	Small Fries	4.75
1	4	Chocolate Ice Cream	10.10
1	5	Steak well done	33.00
1	6	Special of the day	17.90
2	2	Cheese Cake	10.00
2	3	Falafel Sandwich	8.00
2	4	Vanilla Ice Cream Co	7.25
2	5	Baked Salmon with ve	29.80
3	3	Shawarma wrap	10.50

Drinks

drinkNumber	oid	drinkName	drinkPrice
1	1	Coke	2.00
1	2	Sprite	2.00
1	3	coke	2.00
1	4	Water	0.00
1	5	Coke	2.00
1	6	Mango Shake	6.00
2	2	Coke	2.00
2	3	coke	2.00
2	4	Water	0.00
2	5	water	2.00
3	2	water	0.00
4	2	water	0.00

Server

sid	sname
1	Tara
2	Sami
3	Dana
4	Adam

Branch

bid	bname	phone	address	password
1	zamzam	6047177172	6022 Island St	manage1993
2	zamzam	6047177172	2789 Kingsway	manage1993

Different Scenarios and Test Run

Scenario 1: Create a new order

Create New Order

Server ID

2

tableNumber

3

Dish1

Dish Name: Kobe Burger Dish Price: 32.00 remove dish

Dish2

Dish Name: Large Fries Dish Price: 7.00 remove dish

add a Dish

Drink1

Drink Name: Coke Drink Price: 2.00 remove drink

Drink2

Drink Name: Water Drink Price: 0.00 remove drink

add a Drink

note

No Mayo

Submit

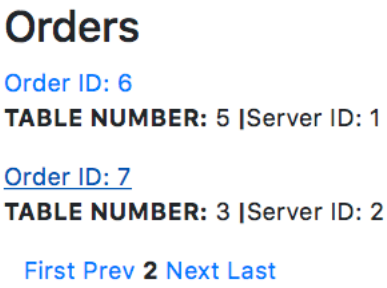
Data Update

```
MariaDB [alqususs]> select * from Branch B, Server S, Orders O, Dishes Dis, Drinks Dr where O.oid=Dis.oid and O.oid=Dr.oid and S.sid=O.sid and B.bid=O.bid and O.oid=7;
+-----+---------+------------+--------------+----------+-----+-------+-----+-----+-----+-------------+---------------------+---------+----------+-----+------------+-----------+-------------+-----+-----------+-----------+
| bid | bname   | phone      | address      | password | sid | sname | oid | bid | sid | tableNumber | orderDate           | note    | dishNumber | oid | dishName   | dishPrice | drinkNumber | oid | drinkName | drinkPrice |
+-----+---------+------------+--------------+----------+-----+-------+-----+-----+-----+-------------+---------------------+---------+----------+-----+------------+-----------+-------------+-----+-----------+-----------+
|   1 | zamzam  | 6047177172 | 6022 Island St | manage1993 |   2 | Sami  |   7 |   1 |   2 |           3 | 2018-04-16 20:04:23 | No Mayo |          1 |   7 | Kobe Burger |     32.00 |           1 |   7 | Coke      |      2.00 |
|   1 | zamzam  | 6047177172 | 6022 Island St | manage1993 |   2 | Sami  |   7 |   1 |   2 |           3 | 2018-04-16 20:04:23 | No Mayo |          1 |   7 | Kobe Burger |     32.00 |           2 |   7 | Water     |      0.00 |
|   1 | zamzam  | 6047177172 | 6022 Island St | manage1993 |   2 | Sami  |   7 |   1 |   2 |           3 | 2018-04-16 20:04:23 | No Mayo |          2 |   7 | Large Fries |      7.00 |           1 |   7 | Coke      |      2.00 |
|   1 | zamzam  | 6047177172 | 6022 Island St | manage1993 |   2 | Sami  |   7 |   1 |   2 |           3 | 2018-04-16 20:04:23 | No Mayo |          2 |   7 | Large Fries |      7.00 |           2 |   7 | Water     |      0.00 |
+-----+---------+------------+--------------+----------+-----+-------+-----+-----+-----+-------------+---------------------+---------+----------+-----+------------+-----------+-------------+-----+-----------+-----------+
```

Scenario 2: Generate Receipt

- *Step 1: From the Orders list find the Order*
- *Step 2: Click on the Order to View it and get system generate receipt*
- *Step 3: Print Hard Copy*

1. Here it is from the list of orders table 3, order id 7

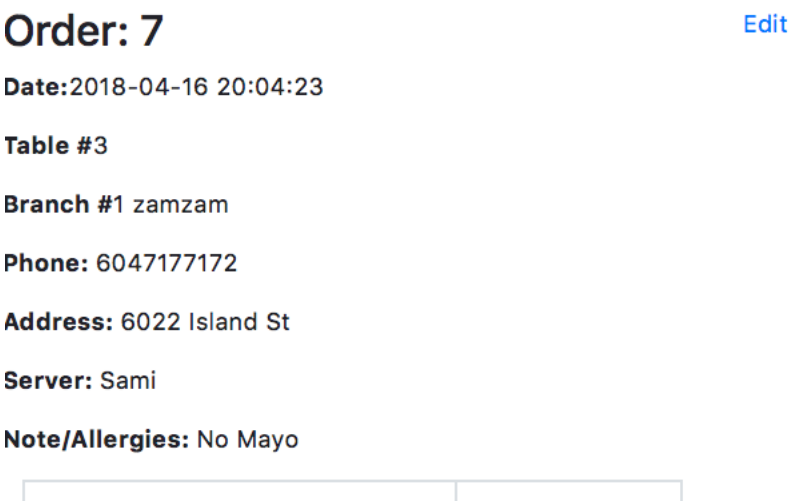


2. Then we click on it and we get redirected to this page

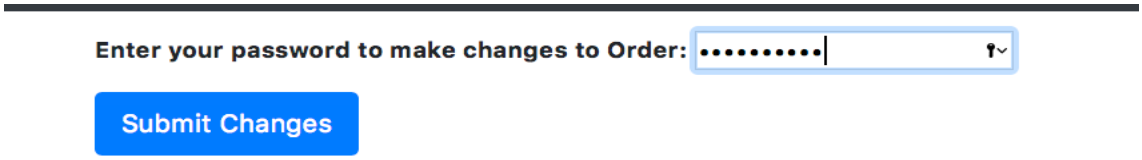
Scenario 3: Order was late so we'll not charge customer for fries, edit/fix order:

- *Step1: Manager has to do it by clicking on edit from above generate page*
- *Step2: Enter password "manage1993"*
- *Step 3: make change*
- *Step 4: submit*

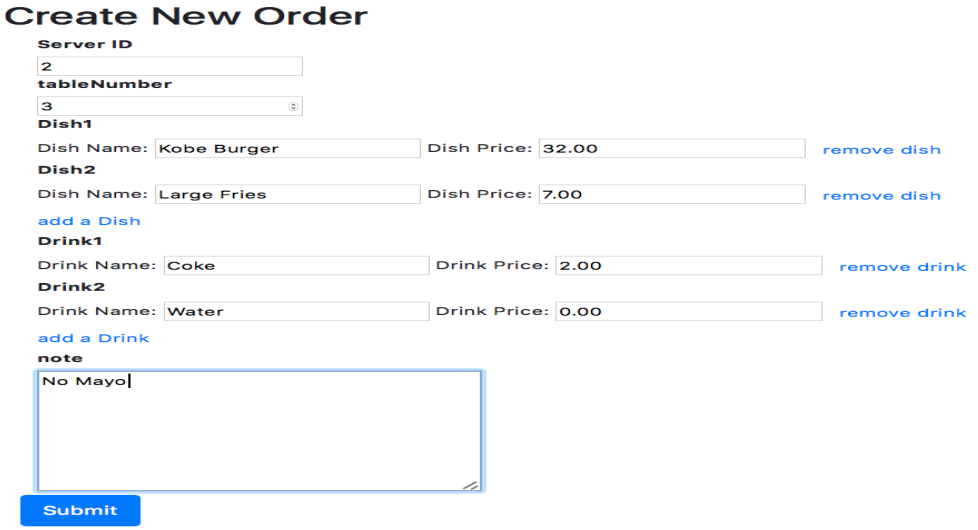
1. *Here is the edit, click on it on the on the right hand side*



2. *Here is the page we get directed to for the password, enter password*



3. *Here is the edit page we get direct to after password success, make changes*



4. *Order 7 after changes have been updated, with fries price $0.00 and total fixed*

Edit

Order: 7

Date:2018-04-16 20:04:23

Table #3

Branch #1 zamzam

Phone: 6047177172

Address: 6022 Island St

Server: Sami

Note/Allergies: No Mayo

Dish List	Price
1.Kobe Burger	32.00
2.Large Fries	0.00

Drinks	Price
1.Coke	2.00
2.Water	0.00

SUBTOTAL 34.00

GST (5%) 1.70

TOTAL 35.70

Scenario 4: Customer from earlier in the day wants the same order they to take out, so we ask for the order/receipt number to access it quicker, they say 1

- *Step 1: Go to search bar and type order id*
- *Step 2: click on it and see details*
- *Step 3: create a new order*

1. *Here is the search bar we type or id and search*

2. *Here is narrowed down result by a search on oid=1*

Orders

1 result(s) found for **1**

Order ID: 1

TABLE NUMBER: 1 |Server ID: 1

First Prev **1** Next Last

Highlighted Queries Used

Queries used in creating an order:

- `$sql = "INSERT INTO Orders (sid, TableNumber, note) VALUES (:sid,:TableNumber,:note)";`
- `$num = $dbh->query('SELECT MAX(oid) as c from Orders');`
- `$stmt2 = "INSERT INTO Dishes (dishNumber, oid, dishName, dishPrice) VALUES (:num, :currID, :dishName, :dishPrice)";`
- `$stmt3 ="INSERT INTO Drinks (drinkNumber, oid, drinkName, drinkPrice) VALUES (:num, :currID, :drinkName, :drinkPrice)";`

Queries used for listing all the orders on different pages:

- `$result =$dbh->query("SELECT * from Orders LIMIT $curr, $perpage");`
- `$num = $dbh->query('SELECT COUNT(oid) as c from Orders');`

Queries used to check password before redirecting to edit page

- `$result =$dbh->query("SELECT * from Orders O, Branch B where B.bid=O.bid and O.oid= $id");`

Queries used in the edit page to load previous data and update new changes:

- `$result =$dbh->query("SELECT * from Orders O where O.oid= $id");`
- `$stmt =$dbh->query("SELECT * from Dishes D where D.oid= $id");`
- `$stmt =$dbh->query("SELECT * from Drinks D where D.oid= $id");`
- `$updateSQL = "UPDATE ORDERS SET sid=:sid,TableNumber=:TableNumber,note=:note WHERE oid = :id";`
- `$updateSQL="UPDATE Dishes SET dishName= :dishName, dishPrice= :dishPrice WHERE dishNumber=:num and oid=:id";`
- `$updateSQL ="UPDATE Drinks SET drinkName= :drinkName, drinkPrice= :drinkPrice WHERE drinkNumber=:num and oid=:id";`

Queries used to search for a certain order:

- `$result =$dbh->query("SELECT * from Orders where oid=$search LIMIT $curr, $perpage");`
- `$num = $dbh->query("SELECT COUNT(oid) as c from Orders where oid=$search");`

Queries used to populate and generate a receipt or Order details:

- `$result =$dbh->query("SELECT * from Orders O, Server S, Branch B where S.sid=O.sid and B.bid=O.bid and O.oid= $id");`
- `$stmt =$dbh->query("SELECT * from Dishes D where D.oid= $id");`
- `$stmt =$dbh->query("SELECT * from Drinks D where D.oid= $id");`
- `$result =$dbh->query("SELECT SUM(dishPrice) as dishTotal from Dishes D where D.oid= $id");`
- `$result =$dbh->query("SELECT SUM(drinkPrice) as drinkTotal from Drinks D where D.oid= $id");`

Queries used to insert the two restaurant branches:

- `INSERT INTO Branch (bname, phone, address, password) VALUES ('zamzam', '6047177172', "6022 Island St", "manage1993");`
- `INSERT INTO Branch (bname, phone, address, password) VALUES ('zamzam', '6047177172', "2789 Kingsway", "manage1993");`

Queries used to insert the server's as they join the restaurant:

- `INSERT INTO Server (sname) VALUES ('Tara');`
- `INSERT INTO Server (sname) VALUES ('Sami');`
- `INSERT INTO Server (sname) VALUES ('Dana');`
- `INSERT INTO Server (sname) VALUES ('Adam');`

*__Note:__ Almost each function is in its own php file. If you like to see the specific queries executed/used, you can refer to the attached source code files.

Conclusion and Potential Future Updates

As we can see, the system executes different queries to create, update, search, print/generate and manipulate data to help manage and track each restaurant's daily transactions and operations. If I am to take this further, I will add more functions to suit said restaurant owner or manager. For the purpose of this project, I wanted to make it flexible without a menu table to make it easier if the manager wishes to not charge for certain items while still having them on the order to keep track of what was ordered regardless of price. Another thing we can add is the ability for the manager to add and remove servers instead of going back to the database admin. We can also add different functions that can give us the total of money made on a certain day, week, month or year and many more functions depending on the needs of the manager or owner.