

Assignment App Project Document

Login Page

First when a person downloads the app they will be greeted with a login page and be asked if they want to change register if they are new.

When they log in the app will look through the database to see what type of person they are and send them to their corresponding activity.

Register Page

This page is for newcomers to the app, they will be able to enter their information (username, email, type - staff or customer and password) into the different text areas and be registered onto the app , from there they can then sign in and continue. The registration takes care of no duplicate email addresses . It also adds special security to prevent weak passwords such as having a minimum of eight characters, having at least a capital and lowercase letter, special symbols and numbers

Home page [For customers]

- As a user logs in they will see all their current orders. After an order is complete they will be able to include a rating . If they try to rate an order that is not complete it will deny the request
- In the top right hand corner there will be a button to logout which instantly takes them out of the application
- In the top left they will have a menu slider that will show a list of options such as orders, restaurants and their own individual profile along with total orders made
- Orders are shown in a row style , showing the order status,creator , owner, restaurant its from(in the form of an image or text if the image cant be pulled up) and date and time of its creation

Home Page [For Staff]

- As staff log in they can see their history of past orders , this will include information such as the Ratings, Customer it was delivered to,
- They should be able to see orders currently being delivered
- Staff will be able to change an orders status from pending, ready and collected
- Staff will have their ratings in the nav bar in the form of a progress bar and see the amount of good and bad ratings they have received for their orders

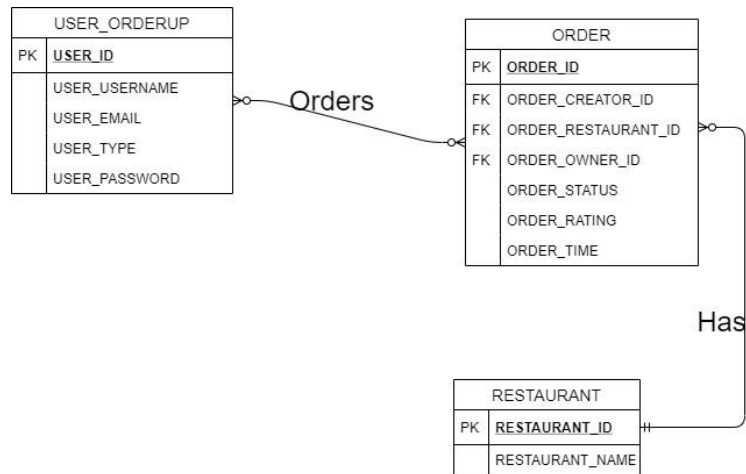
Order page [For Staff]

- In this page staff members can view all orders made and are able to update any orders for any customer by clicking on the order card and selecting what they wish to change the status to be. Staff will see all orders even those they didn't create personally

Business Rules

1. Staff Members can make orders for any registered customer, orders contain restaurants, time the order was created, status of the order (pending as default), the restaurant the customer ordered from and the rating given by the customer
2. A restaurant can have many orders attached to it but an order can only have one restaurant
3. User(Customer/Order Owner) can have 0 to many orders but each order must belong to one and only one user(Customer)
4. Each Order can belong to one and only one order in the order complete history table
5. A user (Staff) creates one and only one order at a time but many orders can belong to the same staff member
6. All orders made are recorded in a separate table, Staff User may make many orders resulting in many history records made. Each order ID can only occur once in the table as each order is unique

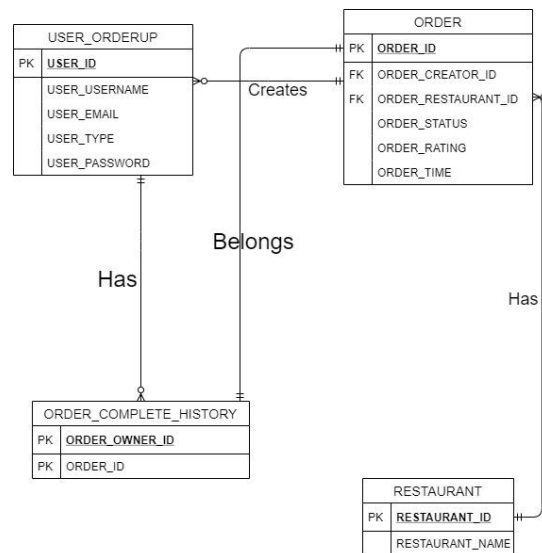
Initial ERD



This is the initial ERD with the Many to Many issues from the users to the various orders that were created in our orders table, this made it difficult to keep track of the various orders made for various users, thus we created a bridge entity to record the history of all the orders for the Final ERD.

Final ERD

This is our final ERD which fixes the many to many relationship we had initially by creating the bridging entity



Langiwe's Contribution(2039033)

I contributed by doing the webservice and database implementation of the project , login and registration as well as the sending and receiving of the users information

Sipho's Contribution (2103063)

I contributed to the design and implementation of various Java methods that helped to produce the output of the information for the users and their orders

Recognition of Issues

Our database has no issues , it also has no null attributes because most of our fields have a default values if they are not entered and we always update that information is explained in the video

Implementation

The main parts of the project involves the implementation of php files to send and receive data for the database.

User Table

1. This table contains the password, type, email , ID and username
2. When a user registers it adds their hashed password to the database with their email , type, and username
3. When login button is pressed a request is sent to our PHP files which runs a query to the database which checks whether the user is present in the database
4. This table is used to access users ID in various processes of the app , especially when displaying the users orders and retrieving their orders

The screenshot displays the MySQL Workbench interface. The top pane shows the 'SCHEMAS' list on the left and the 'USER_ORDERUP' table selected. The middle pane shows the 'Result Grid' with the following data:

| USER_ID | USER_USERNAME | USER_PASSWORD | USER_EMAIL | USER_TYPE |
|---------|---------------|---|--------------------------|-----------|
| 36 | Admin | \$2y10\$uq3GZ2dCt6d0QKwoXX1TORczylmWG... | Admin@gmail.com | Staff |
| 37 | staff123 | \$2y10\$B3403ydsKaruBxoQDAOV.8xY6qdQ2L... | spoman@gmail.com | Staff |
| 39 | test | \$2y10\$gADH4H4CooTt0B3D9H0RwLPPY9... | test@gmail.com | Staff |
| 53 | John123 | \$2y10\$B5EhuJk4C3756K80Umpqwg483... | John@gmail.com | Staff |
| 54 | Testrun | \$2y10\$g2LaorXt6v6t15tJc4vVPFY3xwVQ2... | Testrun@gmail.com | Staff |
| 60 | new_staff | \$2y10\$8qtp0b19ERYfNj.mO.oYUQhUYR... | newstaff@gmail.com | Staff |
| 61 | cust123 | \$2y10\$K55y7F93T63a47NP.hUL45Spq9hg... | cust@gmail.com | Customer |
| 62 | staff1234 | \$2y10\$9LdXK5E557KwZ7aD0GL8fP5dQ2Y... | staff@gmail.com | Staff |
| 63 | Almacano | \$2y10\$gQ4APTb6yLMewCZD77Vieg7L7R7M... | allmacano@gmail.com | Customer |
| 64 | Almacano2 | \$2y10\$8c6vHqD66Y9/CZ9T8.YD65v9c3S... | Almacano | Staff |
| 65 | Anna | \$2y10\$9W0YUR4LL7yDk0PHjnuLU0e0VQK... | acomanyane@gmail... | Customer |
| 66 | Brian | \$2y10\$BhAnc23b67J8ymHf811eH4QVmlv... | Dddd@gmail.com | Customer |
| 67 | Dr Avalanche | \$2y10\$B8JL9158F8wvwyF30yde.BahJvT7... | paikante2045@gmail... | Customer |
| 68 | LANGIME | \$2y10\$G2gZ9pChabIT9T98Aa2DabT4Q2u6... | Langimeale@idou... | Customer |
| 69 | rbee091 | \$2y10\$g1EzqCH6HqpfuxrO.PPhvNlmi7q... | mujtaba.rbee091@gmail... | Customer |
| 70 | faisal27 | \$2y10\$RCHJLGuca6dWdnygJLXOsaZHYH2... | saleemf27@gmail.com | Customer |

The bottom pane shows the 'Field Types' table:

| # | Field | Schema | Table | Type | Character Set | Display Size | Precision | Scale |
|---|---------------|----------|--------------|---------|---------------|--------------|-----------|-------|
| 1 | USER_ID | d2039033 | USER_ORDERUP | INT | binary | 5 | 2 | 0 |
| 2 | USER_USERNAME | d2039033 | USER_ORDERUP | VARCHAR | utf8mb4 | 40 | 12 | 0 |
| 3 | USER_PASSWORD | d2039033 | USER_ORDERUP | VARCHAR | utf8mb4 | 255 | 60 | 0 |
| 4 | USER_EMAIL | d2039033 | USER_ORDERUP | TEXT | utf8mb4 | 65535 | 28 | 0 |
| 5 | USER_TYPE | d2039033 | USER_ORDERUP | VARCHAR | utf8mb4 | 10 | 8 | 0 |

Table Definition

'USER_ORDERUP', 'CREATE TABLE `USER_ORDERUP` (`USER_ID` int(5) NOT NULL AUTO_INCREMENT, `USER_USERNAME` varchar(40) DEFAULT NULL, `USER_PASSWORD` varchar(255) DEFAULT NULL, `USER_EMAIL` text, `USER_TYPE` varchar(10) DEFAULT NULL, PRIMARY KEY (`USER_ID`))

Restaurant Table

1. This table contains the ID and Name
2. We use this table when the ordering page is loaded for the staff to fetch the list of restaurants that customers can get orders from
3. Restaurants Names get used when processing orders to identify the image graphic needed for each order

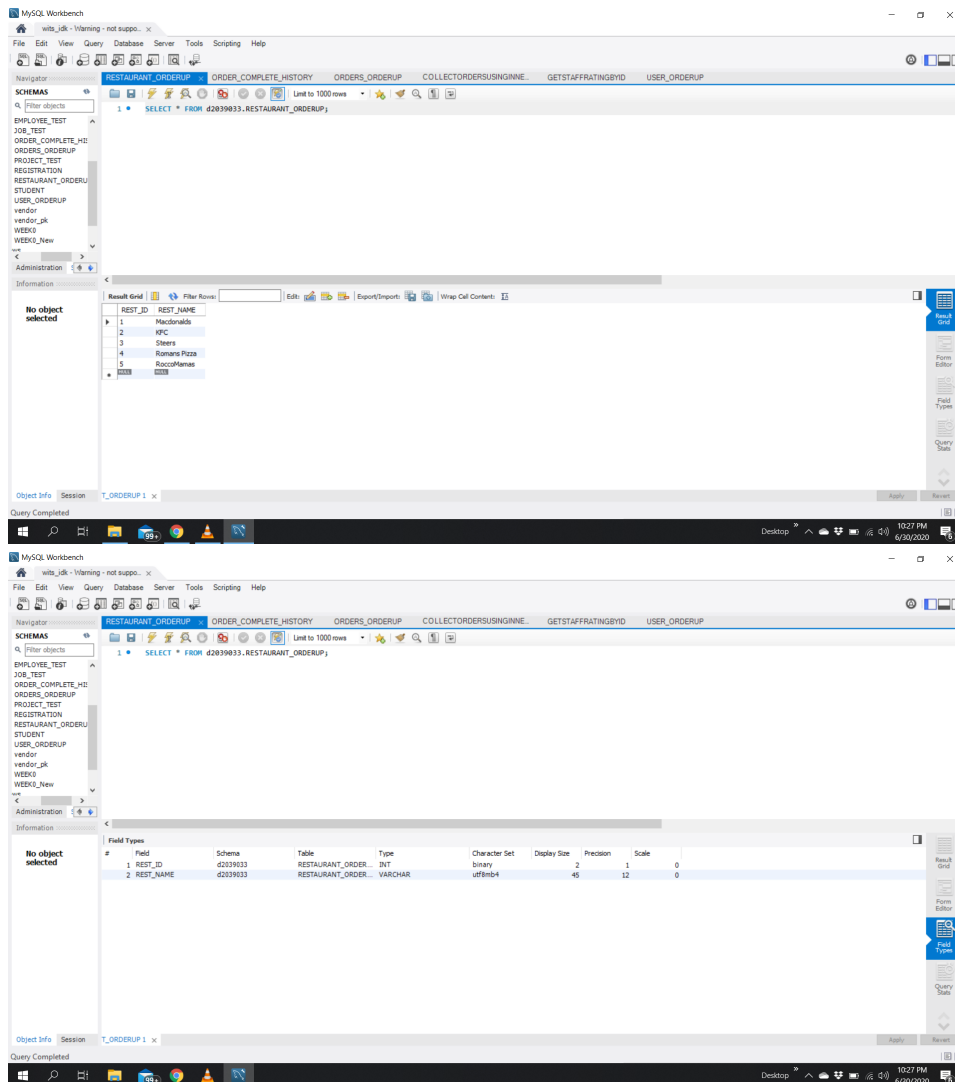


Table Definition

'RESTAURANT_ORDERUP', 'CREATE TABLE 'RESTAURANT_ORDERUP' ('REST_ID' int(2) NOT NULL AUTO_INCREMENT, 'REST_NAME' varchar(45) NOT NULL, PRIMARY KEY ('REST_ID'))

Orders Table

1. This table stores the order creator , ID ,restaurant id (restaurant its from), its current status , time of order and rating(for the creator)
2. This table's information is used for displaying order to customers. For customers it is customer orders and for the staff all the orders are displayed to be able to update the status of the order. This is done with inner joins to the order complete history table to access the customer

The screenshot displays the MySQL Workbench interface. The top panel shows the 'ORDERS_ORDERUP' table selected in the 'SCHEMAS' pane. The middle panel shows the table's structure with the following fields:

| # | Field | Schema | Table | Type | Character Set | Display Size | Precision | Scale |
|---|---------------------|----------|----------------|-----------|---------------|--------------|-----------|-------|
| 1 | ORDER_ID | d2039033 | ORDERS_ORDERUP | INT | binary | 11 | 2 | 0 |
| 2 | ORDER_RESTAURANT_ID | d2039033 | ORDERS_ORDERUP | INT | binary | 2 | 1 | 0 |
| 3 | ORDER_CREATOR_ID | d2039033 | ORDERS_ORDERUP | INT | binary | 5 | 2 | 0 |
| 4 | ORDER_TIME | d2039033 | ORDERS_ORDERUP | TIMESTAMP | binary | 19 | 19 | 0 |
| 5 | ORDER_STATUS | d2039033 | ORDERS_ORDERUP | VARCHAR | utf8mb4 | 20 | 9 | 0 |
| 6 | ORDER_RATING | d2039033 | ORDERS_ORDERUP | VARCHAR | utf8mb4 | 5 | 4 | 0 |

The bottom panel shows the 'Result Grid' with the following data:

| ORDER_ID | ORDER_RESTAURANT_ID | ORDER_CREATOR_ID | ORDER_TIME | ORDER_STATUS | ORDER_RATING |
|----------|---------------------|------------------|---------------------|--------------|--------------|
| 80 | 2 | 36 | 2020-06-27 15:16:45 | PENDING | NONE |
| 81 | 5 | 36 | 2020-06-27 15:19:49 | COLLECTED | BAD |
| 82 | 3 | 36 | 2020-06-27 17:57:56 | COLLECTED | GOOD |
| 83 | 1 | 36 | 2020-06-27 20:39:22 | PENDING | NONE |
| 84 | 4 | 36 | 2020-06-28 20:53:16 | COLLECTED | BAD |
| 86 | 3 | 71 | 2020-06-29 22:36:37 | PENDING | NONE |
| 87 | 3 | 71 | 2020-06-29 22:36:37 | COLLECTED | NONE |
| 88 | 3 | 71 | 2020-06-29 22:36:38 | COLLECTED | NONE |
| 89 | 3 | 36 | 2020-06-30 00:29:26 | READY | NONE |
| 91 | 5 | 36 | 2020-06-30 00:30:33 | COLLECTED | GOOD |
| 92 | 5 | 36 | 2020-06-30 01:00:46 | PENDING | NONE |
| 93 | 1 | 73 | 2020-06-30 01:01:24 | PENDING | NONE |
| 94 | 5 | 73 | 2020-06-30 01:02:54 | PENDING | NONE |
| 95 | 5 | 36 | 2020-06-30 15:27:24 | PENDING | NONE |
| 96 | 2 | 36 | 2020-06-30 18:00:31 | READY | NONE |

Table Definition

```
'ORDERS_ORDERUP', 'CREATE TABLE `ORDERS_ORDERUP` ( `ORDER_ID` int(11) NOT NULL  
AUTO_INCREMENT, `ORDER_RESTAURANT_ID` int(2) DEFAULT NULL, `ORDER_CREATOR_ID`  
int(5) DEFAULT NULL, `ORDER_TIME` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
`ORDER_STATUS` varchar(20) DEFAULT 'PENDING', `ORDER_RATING` varchar(5) DEFAULT  
'NONE', PRIMARY KEY (`ORDER_ID`), KEY `ORDER_RESTAURANT_ID_idx`  
(`ORDER_RESTAURANT_ID`), KEY `ORDER_CREATOR_ID_idx` (`ORDER_CREATOR_ID`),  
CONSTRAINT `ORDER_CREATOR_ID` FOREIGN KEY (`ORDER_CREATOR_ID`) REFERENCES  
`USER_ORDERUP` (`USER_ID`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT  
`ORDER_RESTAURANT_ID` FOREIGN KEY (`ORDER_RESTAURANT_ID`) REFERENCES  
`RESTAURANT_ORDERUP` (`REST_ID`) ON DELETE NO ACTION ON UPDATE NO ACTION)
```


Complete Order History

1. This is our bridging entity to avoid many to many relationships. It stores the order Owner and order ID. For other tables to access this order owner we use Inner joins on the Order ID

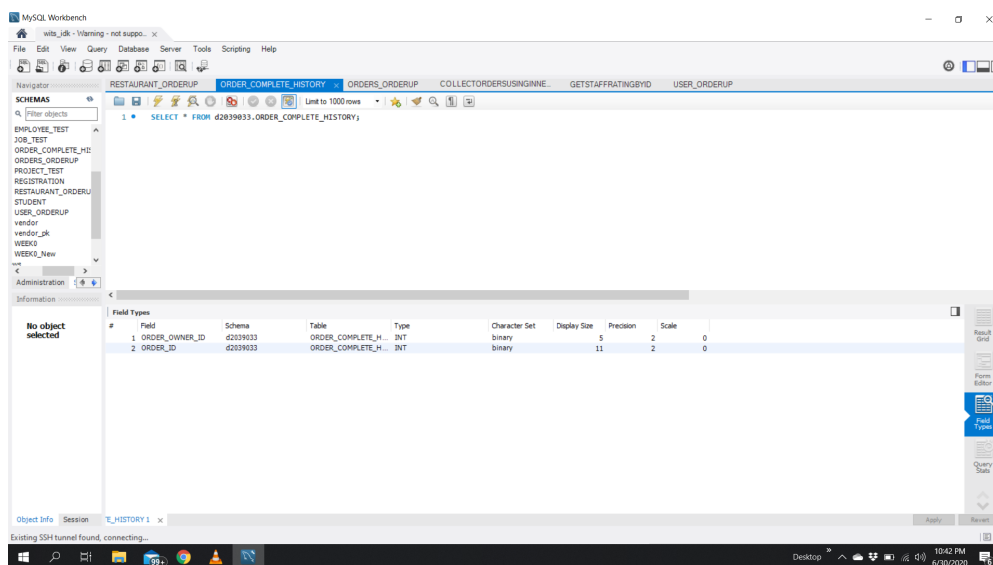
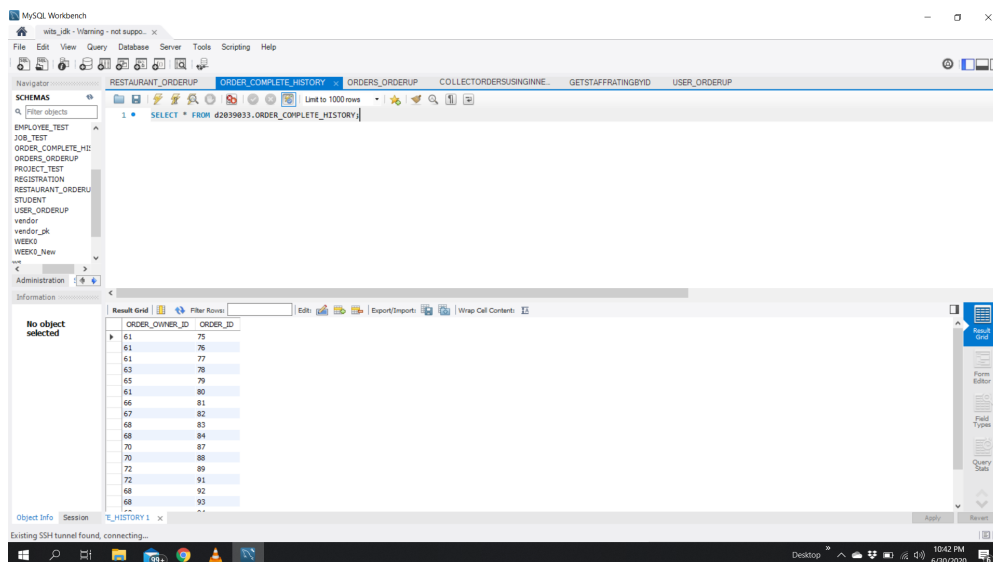


Table Definition

```
'ORDER_COMPLETE_HISTORY', 'CREATE TABLE `ORDER_COMPLETE_HISTORY` (\n`ORDER_OWNER_ID` int(5) NOT NULL,\n `ORDER_ID` int(11) NOT NULL,\n PRIMARY KEY\n(`ORDER_OWNER_ID`,`ORDER_ID`),\n KEY `ORDER_ID_idx` (`ORDER_ID`),\n CONSTRAINT\n`ORDER_OWNER_ID` FOREIGN KEY (`ORDER_OWNER_ID`) REFERENCES `USER_ORDERUP`\n(`USER_ID`) ON DELETE NO ACTION ON UPDATE NO ACTION,\n CONSTRAINT `ORDER_ID`\nFOREIGN KEY (`ORDER_ID`) REFERENCES `ORDERS_ORDERUP` (`ORDER_ID`) ON DELETE NO\nACTION ON UPDATE NO ACTION\n)
```