Alex Reyes

COMP 296

Capstone Project Application Manual

**FISER Application Manual**

**Table Of Contents:**

1. **Logging into FISER Application Demonstration (Login\_S.java GUI).**
   1. Enter User Name
   2. Login Button
   3. Reset Button
   4. Exit Button
2. **Pizza User Interface Application Demonstration (Pizza\_UI.java GUI).**
   1. Ordering Menu Pane
   2. Ordering Menu Drop Down
   3. Delivery Tabs and buttons.
   4. Till Tab
   5. About Tab
   6. Admin Tab
3. **SQLite Query Table Demonstration (Tables.java GUI).**
   1. Admin Tabs

A1. Display Users

A2. Entering New Admin Users

A3. Edit Admin Users

A4. Remove Admin Users

* 1. Employees Tab

B1. Entering New Employee Users

B2. Edit New Employees

B3. Remove Employees

* 1. Customers Tab

C1. Enter Customers

C2. Edit Customers

C3. Delete Customers

* 1. Menu Items Tab

D1. Display Menu Items

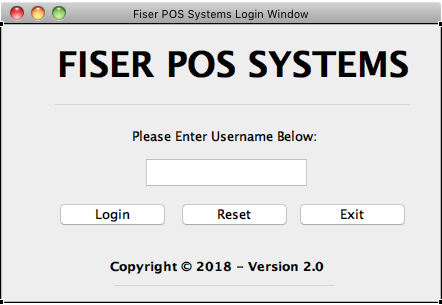
* 1. Tills Tab

E1. Display Tills

E2. Delete Tills

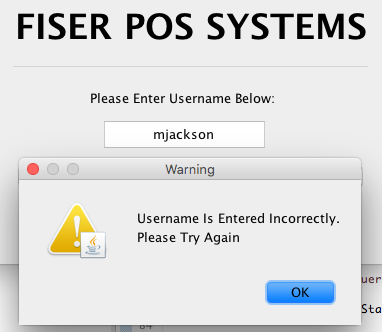
1. **Logging into FISER Application Demonstration (Login\_S.java GUI).**

Description: To login to the application, we must enter the employee username or admin username to access/bypass the login function. No password is necessary for this portion. If the end users input the incorrect value, a popup warning will display, prompting the end user that the incorrect username has been entered.



1A. Enter the username or admin username into the text field.

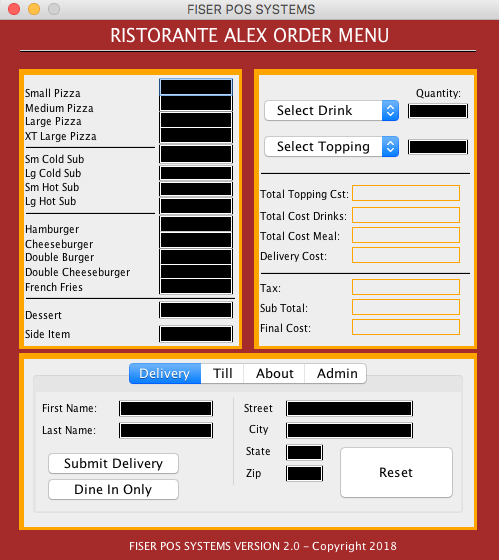
1B. The login button will allow the end user to access the FISER POS Application (Pizza\_UI.gui). If the user enters the incorrect value, it will display that the username is incorrect. Else, it will prompt the user that they have successfully logged in.



1C. The reset button will reset the values of the text field.

1D. The exit button will close the application.

1. **Pizza User Interface Application Demonstration (Pizza\_UI.java GUI).**



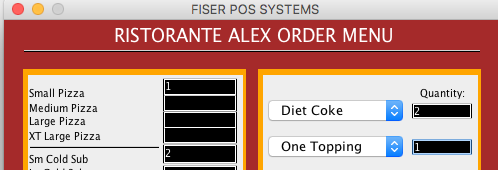
Description: After the end user has entered their username correctly from the Login GUI, the Fiser POS Application will load. The end user will input what the customer will want to order. On the left pane we have the menu items that the restaurant has provided us. The right pane has two drop down menus if the customer wants a drink, or if they want to add toppings (three toppings selections: One Topping, Two Toppings, Three Toppings). Below that, has the calculations that break down each menu item price, delivery price (if it’s a delivery) and finally the tax, subtotal, and final cost.

The bottom pane has four tabs. Delivery, Till, About, and Admin tabs. The delivery tab must be entered first (before we can till the amount). If the customer wants to submit a delivery, they must enter the customers information, otherwise select the dine in option. Once, the end user inputs either “Dine In” or “Delivery information”, we can now begin to work on our menu ordering process.

The Till button will calculate all the information and transfer it to our SQLite Database. The About tab is just a demo of our application information and the restaurant location and contact information. The Admin tab will allow Admin users or the owner to access certain features that regular employees cannot access.

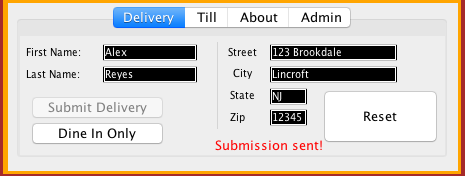
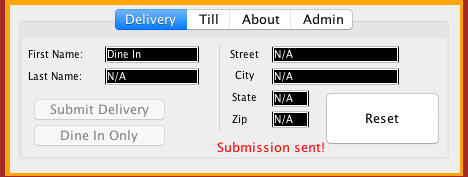
2A. The end user can input what the customer would like by entering integer values into the text fields.

2B. The end user can input if the customer would like to Select a Drink or Select a Topping.



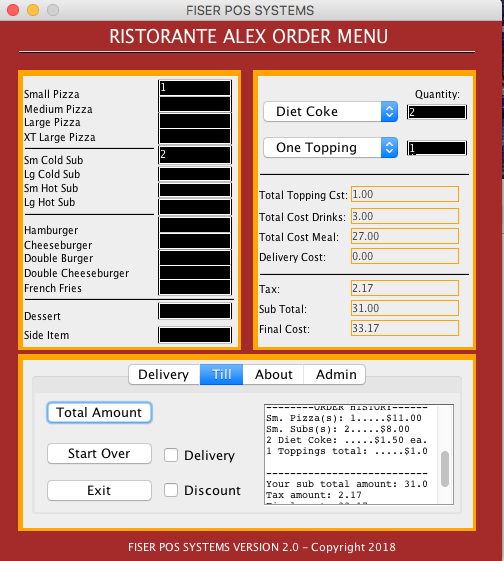
2C. The Delivery tabs has three buttons. Dine In, Delivery, and Reset. The reset button resets all the fields (In case the end user makes a mistake).

If the user wants to dine in, it will display “Dine In” inside the text fields automatically. A “Submission Sent!” text will display on the bottom right corner, to notify the end user the information has been sent to the SQLite Database. The buttons will be disabled so the end user doesn’t submit the same information twice on accident. The same method/function occurs for delivery as well.

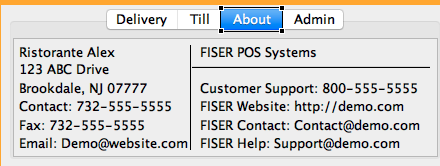


2D. The Till Tab has three buttons, two check boxes, and one text area with a scrollPane. The Total button is disabled, until the end user puts in “Dine In” or “Delivery” option. This will avoid any premature errors that can occur and would create errors within the database. The Start Over button is available, in case the end user has inputted the customer ordering requests incorrectly. The exit button will simply close the application.

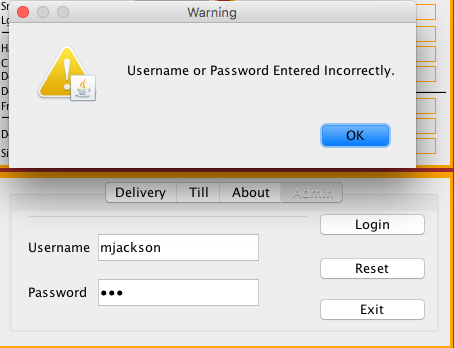
The Delivery check box is to charge the customer a delivery fee and the discount checkbox is for any guests who receive a discount from their order. Lastly, the text area box is a receipt that will print out to the customer at the restaurant. It will show the end user a copy as well.



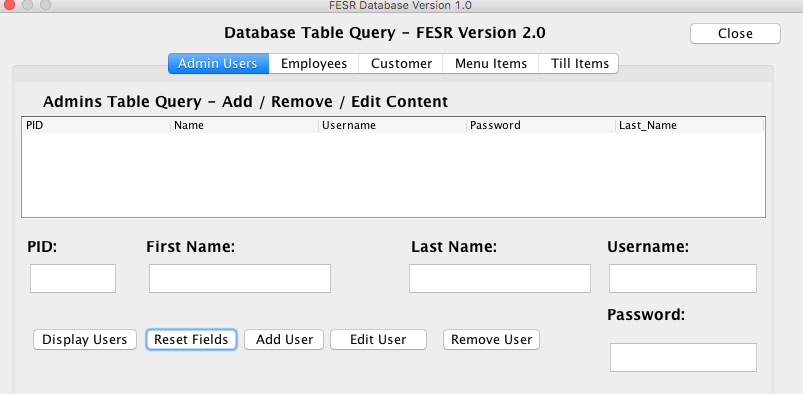
2E. The About Tab is just a mock demonstration of where the restaurant it located and the application support page.



2F. The Admin Tab will give the end user access to the SQLite Database where we can view/edit/delete/add values to our SQLite tables. The login portion is the same process as the Login GUI, but this one has a password field as well. If the end user inputs the information incorrectly, a warning will message will popup.



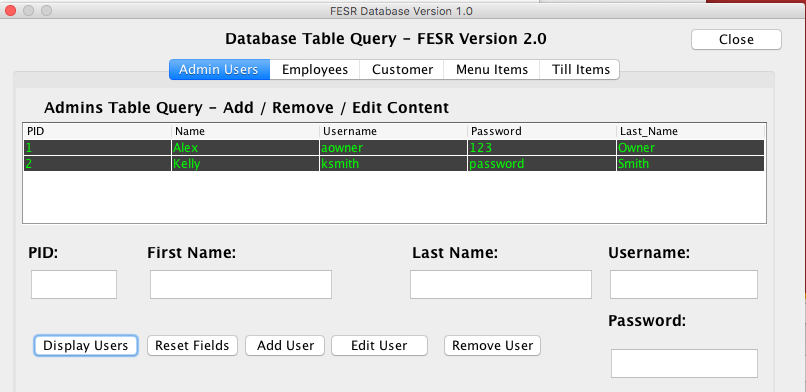
1. **Database Table Query Demonstration (Tables.java GUI).**



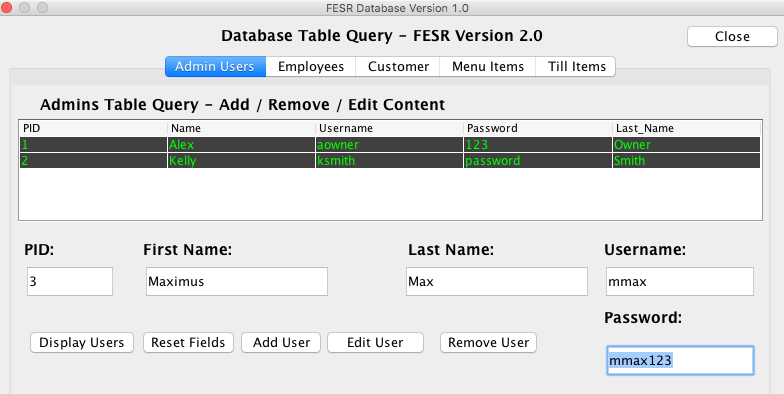
Description: The Database Tables Query is only available to admin users or managers. There are four tabs: Admin Users, Employees, Customer, Menu Items, Till Items. Also, The Database Tables Query will give us access for new hires, customer editing, and voiding tills (in case customer wants a refund). Each one will be demonstrated and described below more in detail.

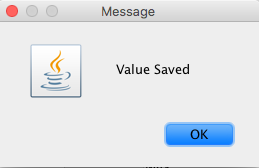
Admin users are users who can have access to the Tables.GUI. The Admin tab has five buttons: (Display Users, Reset Fields, Add User, Edit User, Remove User), five text fields: (PID short for Power User Identification, First Name, Last Name, Username, and Password), and one table that will display our queries.

3A1. Display Users will show two admin users that have been set already.

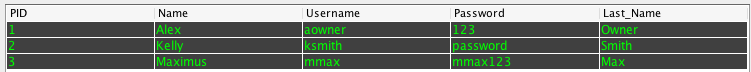


3A2. If the end user wants to add another Admin User, they must fill out the next PID (in this case it will be 3), along with their first name, last name, custom username, and custom password. We will use Maximus as an example. One we fill out the data, it will say “Value Saved.” To notify the end user, it was saved successfully.

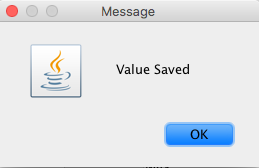


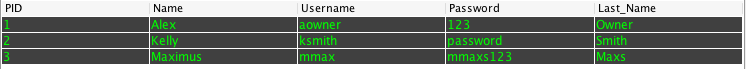


We can now see that Maxmius is successfully stored in the database:

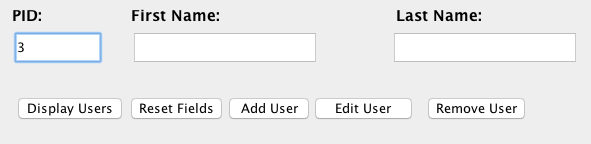


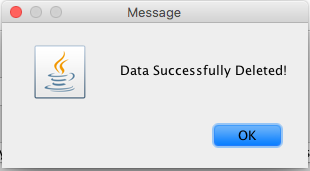
3A3. If the end user has made a mistake by misspelling Maximus’ last name, we can simply edit the correct fields and retype it. *(Note: End user must enter all user’s information when they edit the fields, or the table will update with missing fields)*. A value saved pop up will display to notify the end user that everything has been saved successfully.

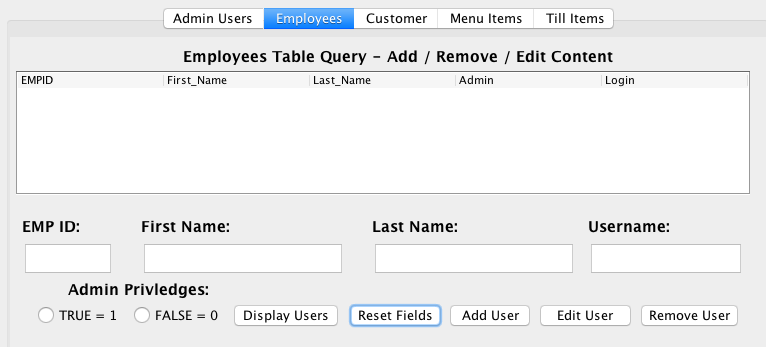




3A4. If Maxmius is no longer with the company, we can simply enter the users PID into the table and click on the “Remove User” button. The query will then delete all values associated with that field. Once we have entered the correct PID. We hit the “Remove User” button and it will successfully remove Maximus. A popup message, “Data Successfully Deleted!” will be displayed to notify the end user that the data has been deleted.

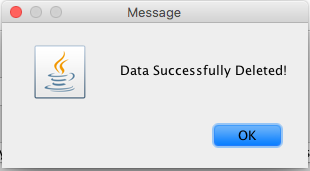


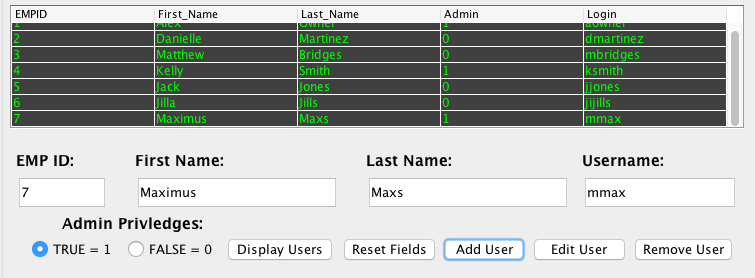




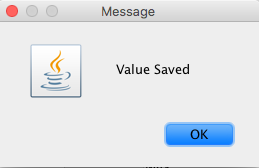
3B. Employees Tab will display all the employees that work for the company. It will provide the end user with: Employees ID (EMPID), First Name, Last Name, Admin Privileges, and Login ID). This tab includes the following java items: Five buttons (Display Users, Reset Fields, Add User, Edit User, and Remove User), Two radio buttons (True or False (for Admin Privileges)), Four Text Fields: (EMPID, First Name, Last Name, Admin User, and Login ID), and finally, one table to display our table query.

3B1. If the end user needs to add a new employee, we can simply input the data into the required fields. For this case, we will choose: Maximus Maxs who will be an admin user. Since they are 6 employees, Maximus will be employee number 7. Our EMP ID will then be 7. After we fill out the correct data, the database will give us a popup to notify that our data has been saved:





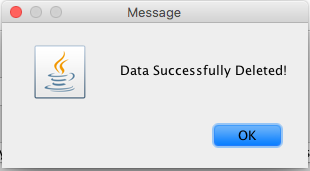
3B2. If the end user has made a mistake by misspelling Maximus’ last name, we can simply edit the correct fields and retype it. *(Note: End user must enter all users information when they edit the fields or the table will update with missing fields)*. A value saved pop up will display to notify the end user that everything has been saved successfully.

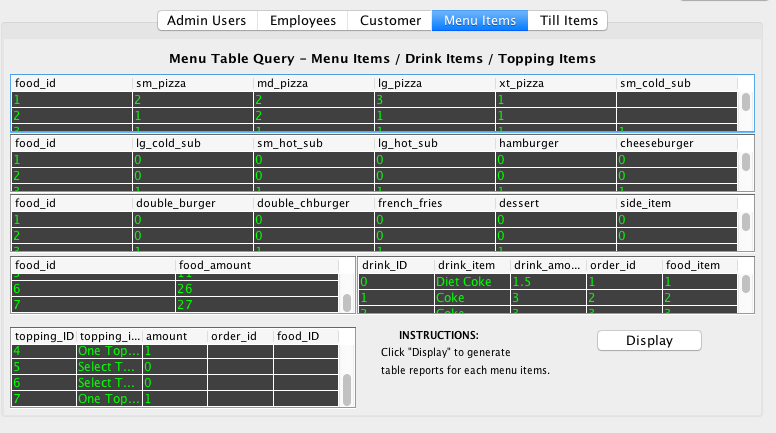




3B3. If Maxmius is no longer with the company, we can simply enter the users PID into the table and click on the “Remove User” button. The query will then delete all values associated with that field. Once we have entered the correct PID. We hit the “Remove User” button and it will successfully remove Maximus. A popup message, “Data Successfully Deleted!” will be displayed to notify the end user that the data has been deleted.



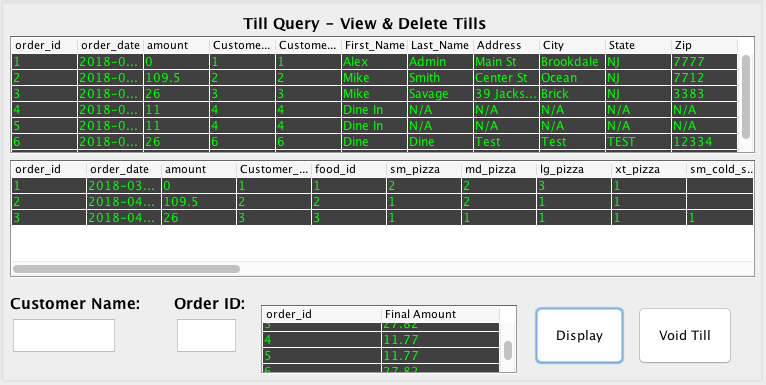




3D. The Menu Items tab will display all the items that correspond to what the customer has ordered (Order\_id table). This is a history log all food items that has been ordered by the customer. The order goes from oldest to newest. Each item auto increments from the food’s Primary Key inside the database.

Menu Items includes 6 tables: Top three tables are all the food items, Two Tables on the bottom display the food id plus the food dollar amount, on the right side is the history of drinks order, and bottom left are the history of toppings ordered, and one display button (to call the query to display on each table).

3D1. Once the end user hits the display button, each food\_id matches the food\_id tables with each order that has been placed. Same applies for the drinks table and toppings table.



3E. Our last tab is our Tills Items Tab. The Till Items Tab is created in case the customer requests a refund on their purchase. This will help the end user do so by looking up the ordering date, customers contact information, the till amount, and the order\_id.

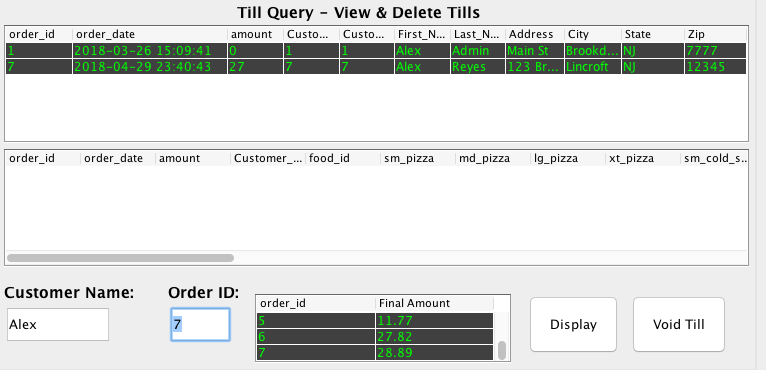
There are three tables: The top table shows the order\_id (each order id matches the customers order), the date, and customer contact information. The second table shows the history of what the customer has ordered. The third table below shows the final dollar amount, on what the customer has paid for.

There are two search bars: Customer Name (case sensitive and must be their first name) and Order ID. The Customer Name is took look up what the customer has ordered and their Order ID.So, this way we can aviod confusion if there are more than one user with the same first name.

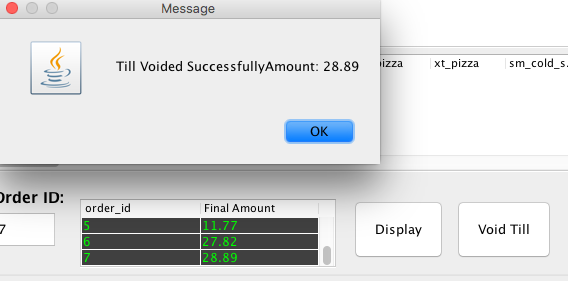
3E1. Lastly, there are two buttons. Display, is to display the transactions for our SQLite Database, and the Void Till button that will void the tills.

3E2. To void a transaction, simply search the customers first name (case senstive) and find their order that it corresponds with. In this case we will type “Alex” and use our last transaction we have placed. *(We’re using the same transaction as we first did on the beginning of the manual).*

The end user will type Alex into the Customer Name and check the correct Order ID.



The end user then checks the item that was entered by the transaction which occurred today (4/29/18). Once we hit the Void Till button, it will void and delete all transactions associated with Customer: Alex Reyes. There will be a popup box to confirm that it has been deleted.



When we display the table again, we can now see that the customer has been successfully deleted.