Documentation

Description

The goal of my project is to develop a system for managing customers' purchases. In order for me to do this I will have to have a backend database that will hold information about my customers, invoice’s and products. I wanted to make a gui (graphical user interface) for my users because it is a nice and easy way using my system and it is very user friendly. In my system you are able to create, update, delete and retrieve, you can do these operations on all of my database tables, which are called customer, invoice and product. If you are making a system you should make it as robust as you can, in order to do this we can implement error handling. For example, if we have a textbox for a phone number, we should **ONLY** accept integers not strings.

Requirements

The requirement’s for this project is to be able to make a system to be able to allow new users to sign up to my system and after the user has done that, they will be able to login to their account. After the user has signed up the info that they supplied into the sign up form that will be added into a mysql database. And then whenever the user wants to login to their account in the future, whatever info is in the database will be checked depending on what the user types into the login form. If the user supplies the correct info they will be logged into there account, otherwise the user will be prompted with an error message and asked to try again.

Database Layout

Structure for customer table:

A picture containing calendar

Description automatically generated

Data for customer table:

Text

Description automatically generated

Structure for invoice table:

A screenshot of a computer

Description automatically generated with medium confidence

Data for invoice table:

Text

Description automatically generated

Structure for product table:

A screenshot of a computer

Description automatically generated with medium confidence

Data for product table:

Text

Description automatically generated

Er Diagram (Entity-relationship)

Interesting source code snippets

signupPage.java:

Text

Description automatically generated

In the code snippet above I am connecting to my database, then I have a try block in this block of code I insert some details that the user has entered this is done before this code it looked at. Then I have a button with the name “Sign up !” when the user clicks this button my system checks to see if the correct information has been entered, if some of the information is incorrect it will inform the user what is wrong, so the user knows to update that bit of the sign up form. For this part of the project, I found it quite challenging to implement error handling. It took me a lot of time to understand what I needed to do and how I was going to let the user know if they have typed anything in incorrectly. In the code snippet above it looks like I am only error handling the customer id and the customer’s name but underneath that code I have more code for the other bits of information for the user and that code is being error handled also.

login.java:

Text

Description automatically generated

Above in the code snippet I have a login system. I have a button called loginButton “Login”, you cannot see the button because it is above this bit of code from above. But when the user clicks this button, my system gets there email and password, from my database. Then I have a try block of code I have a connection to my database, then I do a query to see if the email and password from the database match what the use entered. Then I make a variable called result, this variable lets me execute a query, if the result is successful it will check every record in the database for the supplied information and if the information supplied is correct the user will be logged in and they will be told a message something like “You have logged in !”, otherwise if the user enters some information that is incorrect my system will still check the database for the information supplied but if it is incorrect the database won’t find any record and my system will return an error message something like “Login failed”.

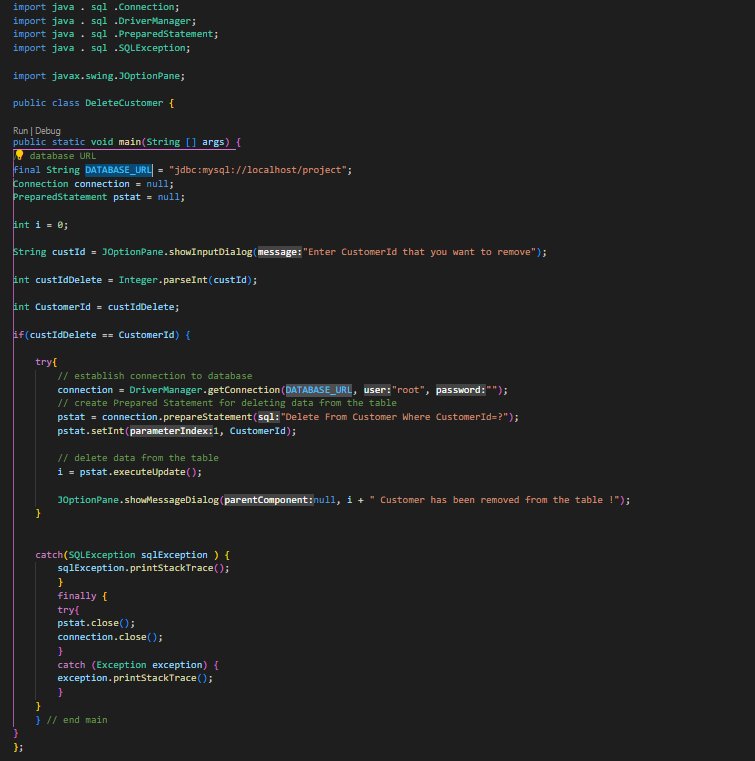
UpdatingCustomer.java

Text

Description automatically generated

Above in the code snippet I have a file to update a customer’s details. I have a try block of code in this block of code, I have a connection to my database, I also have a button called updateButton “Update Customer”, you cannot see the button because it is above this bit of code from above, I also have the update form in this code snippet, but it is in the file. So, depending on what the user wants to update, when the user click’s the “Update Customer” button they will be able to update a certain customer. As you can see I have error handling again for all of the different types of input from the user.

DeleteCustomer.java



In the above code snippet I have a script that lets you delete a customer from the customer table. ­I make a variable called custIdDelete and I convert custId from a string to an integer, then I have an if statement to check to see if the custIdDelete is equal to the CustomerId and if it is equal I connect to my database then I do a sql query then I send that CustomerId value to my database then the variable i is executed. So basically, what is happening is the user types in a customer id that they want to remove completely from the customer table and if the number the user types in is equal to the same number in the customer table, then that customer is removed from the customer table. Note, I don’t flag a customer to be deleted I just completely remove the record from the customer table.

DB.java

Text

Description automatically generated

In the above code snippet, I have a gui (graphical user interface) for doing sql queries. When you run this file, you are able to do normal sql queries but in a nicer way for the user, but as a programmer we tend to use the terminal or cmd because it is much easier but for a user this is a lot more user-friendly instead of using something like a terminal, see image below:

Shape

Description automatically generated with medium confidence

Tests

Create Test:

While I was building this project, I had to do lots of tests. Doing tests for your system is very important because let’s say you write all your code and never test it until you are done the project, when you go to run it, there is a very high chance you will get lots of errors. This is a very bad thing to do, and you should never do that! One of the best things to do is while you are building the project and you get one thing done, test it. For example, when I finished the first part of the crud create, I decided to give it a test to see if it works and works how I want it to work. Sadly, I actual had lots of problems with my create. At the start I was always getting lots of red errors, this was very annoying, but I pursued, and I eventually got something created into my database, I looked at my terminal in vs code and there was no error, but when I looked at my database, something was created but all the bits of information were null (empty). It took me hours to figure it out how to fix it but when I fixed it, I was really happy because now I have a sign-up form (create) a customer.

Retrieve Test:

When I was doing my testing for retrieving data from my database. Surprisingly this was the only program of the crud that worked straight away. You can query any database table that is in my database called “project” but to do this you must have established a connection to my database.

Update Test:

When I was done programming my update customer, I went off and started to test my update customer to see if it worked properly. When I first tested it, it didn’t work at all. For example, you have to enter the customer id you want to update, I did this and I wanted to update on of the customers phone number but when I ran the java file it ran without any errors popping up but it didn’t update the customer id that I wanted it to update. Because it was not working I decided to start debugging my code, and after hours of getting nowhere, I decided to use the internet. I was trying to not use the internet because I was trying to learn this on my own, but it got to a stage where I couldn’t do it anymore so I decided to use Google and YouTube for some help. Even do I was using Google and YouTube for help it still took my hours to figure it out, but I finally got the update customer to work.

Delete Test:

When I was doing my test for my last bit of the crud delete, I started off by connecting to by database then I ask the user what CustomerId do they want to remove, depending on what number the user types in my code will check my database to see if the number the user types in is in the database, if it is that customer will be completely removed from the customer table and a message will be prompted to the user telling them “A user has been removed from the database”. Otherwise, if the user types in a number that is not in the database, my code will still check to see if the number exists in my customer table, but if the number is not there, my code will show a message to the user saying something like “no customer has been removed from the customer table”.

Source code, Database dump, Project API ,Executable jar file:

Here is a link to my GitHub repository -> <https://github.com/KyleKinsella/Object-Oriented-Software-Development---CA3-Customer-Invoice-Product-Management-System>. Here I have my source code for my project, I also have my database dump file this is called “project.sql”, and I have an executable jar file and finally I also have a mysql connector, this allows me to connect my java code to do my database.