**C868 – Software Capstone Project Summary**

**Task 2 – Section C**

|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | http://www.idevnews.com/views/images/uploads/general/wgu_logo.png  Customer Database and Appointment Solution for Sam Sam Tech. |
| **Student Name:** | Isabella Cerda Grigolla |

Table of Contents

[Task 2 Part C – C868 Software Development Capstone 4](#_Toc151233881)

[Application Design and Testing 4](#_Toc151233882)

[Design Documents 4](#_Toc151233883)

[Database Design 4](#_Toc151233884)

[UI Design 5](#_Toc151233885)

[Unit Test Plan 7](#_Toc151233886)

[Introduction 7](#_Toc151233887)

[Purpose 7](#_Toc151233888)

[Overview 7](#_Toc151233889)

[Test Plan 7](#_Toc151233890)

[Items 7](#_Toc151233891)

[Features 7](#_Toc151233892)

[Deliverables 7](#_Toc151233893)

[Tasks 7](#_Toc151233894)

[Needs 8](#_Toc151233895)

[Pass/Fail Criteria 8](#_Toc151233896)

[Specifications 9](#_Toc151233897)

[Procedures 10](#_Toc151233898)

[Results 11](#_Toc151233899)

[12](#_Toc151233900)

[User Guide 13](#_Toc151233901)

[Introduction 13](#_Toc151233902)

[Installation the Database and Using the Application 13](#_Toc151233903)

[*Installing the MySQL Database* 13](#_Toc151233904)

[*Setting up the Database References* 16](#_Toc151233905)

[*Setting up the Tables in the Database* 17](#_Toc151233906)

[*Login into the Program* 23](#_Toc151233907)

[*Directory* 23](#_Toc151233908)

*[Customer Records](#_Toc151233909)* [23](#_Toc151233909)

[*Adding, Modifying, Deleting, and Searching the Customer Records Page* 23](#_Toc151233910)

[*Appointments* 24](#_Toc151233911)

[*Adding, Modifying, Deleting, and Searching the Customer Appointments Page* 24](#_Toc151233912)

[*Calendar* 25](#_Toc151233913)

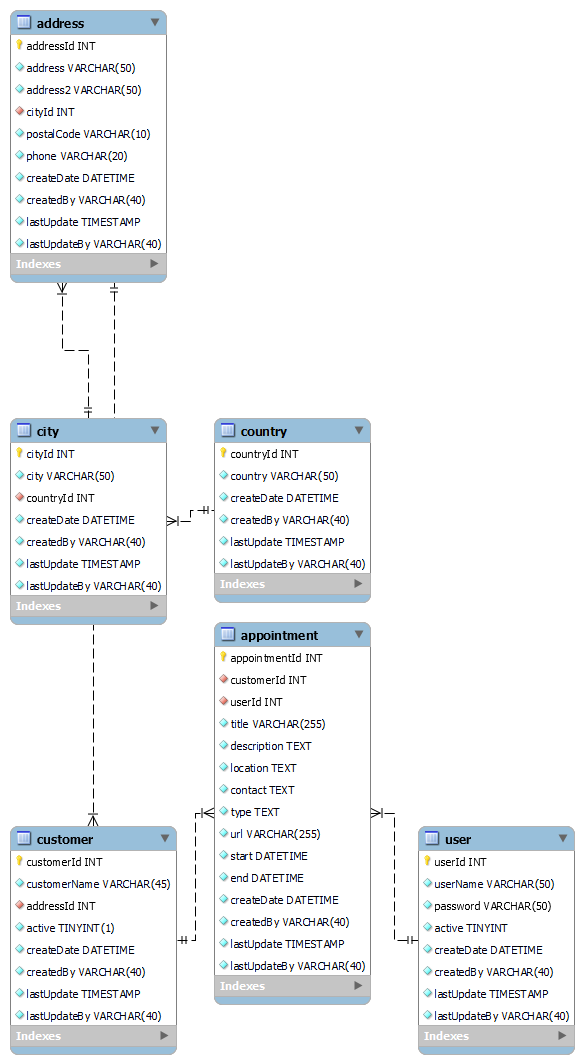
[*Reports* 26](#_Toc151233914)

Task 2 Part C – C868 Software Development Capstone

# Application Design and Testing

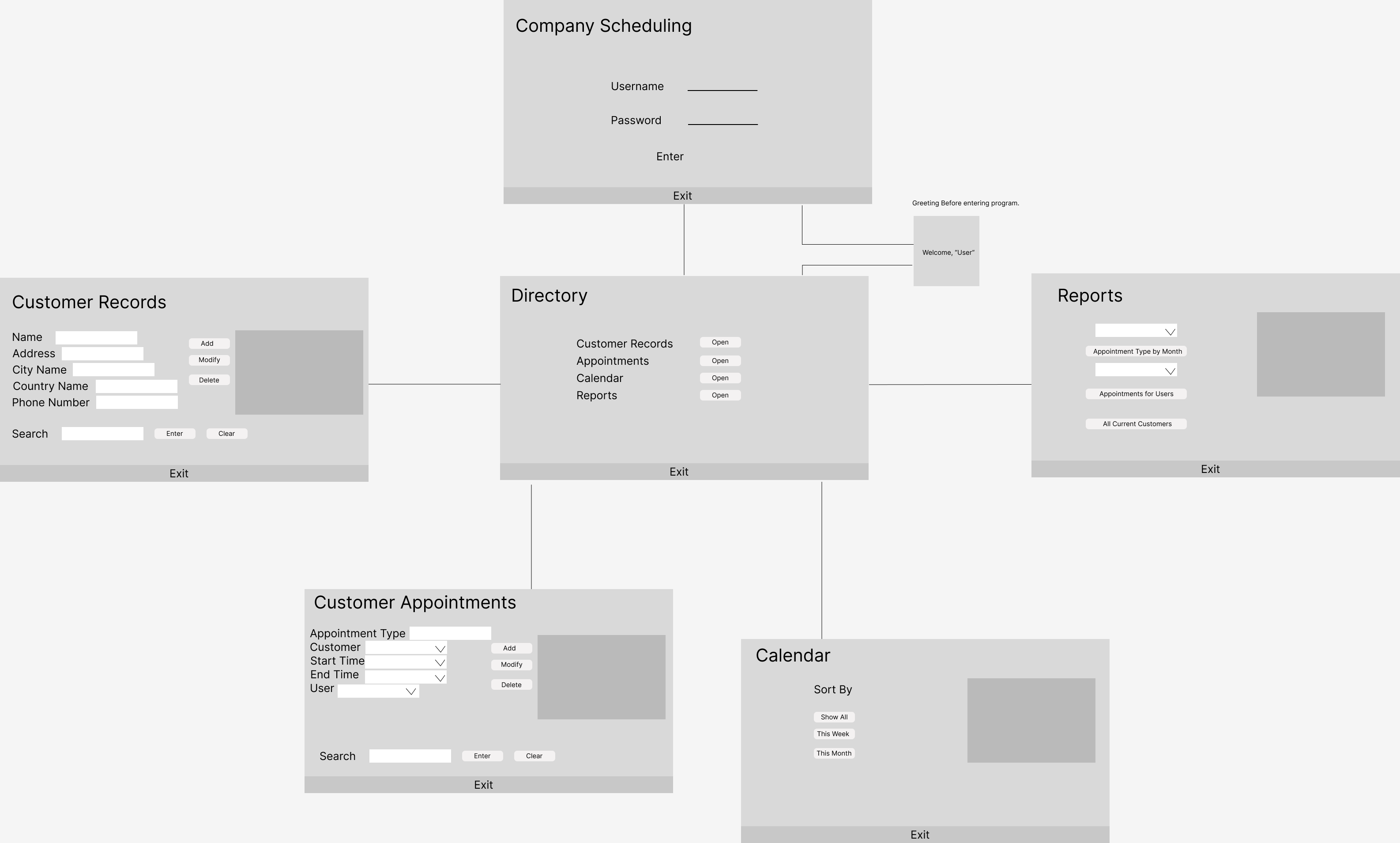
# Design Documents

## Database Design

Below is an image of the overall design of the database for the application. There will be 3 tables that rely off of the customer Table: Address, City, and Country. Each table can function on its own but there is information that is needed when creating a customer. These functions help ensure that once Sam Sam Tech is ready to expand to different locations there will be less of a need to upgrade in the future. The user table is for all the employees that will be using the program. Then there is the appointment table which will house all of the necessary data that will be the core function of the program.

## UI Design

There are two examples of a wireframe for this project. A low-fidelity wireframe and a high-fidelity wireframe. The low-fidelity shows overall how the application function while the high-fidelity version shows what the program will actually look like.



A screenshot of a computer

Description automatically generated

# 

# Unit Test Plan

## Introduction

### Purpose

One of the requirements from our client was a search function in the Customers Records and Appointments. I’ll be reviewing how it functions and why.

### Overview

For the overall project, this was just a small part of the core functionality. There are some functions that are quite similar to this segment but are quite different. Such as the picker elements in the calendar page. This test is to ensure that the functionality of the search for these two pages works. For these tests, we would use test data and test inputs to see if the functionality for this requirement would work. There will be more specific details below.

## Test Plan

### Items

Ensure that the search requirements for both Customer Records and Customer

Appointments works.

### Features

The ability to search the given data table, as well as restoring the data table back to its

original state before the search was used.

### Deliverables

Once the testing procedure is over there will be a short test result shared with the client.

### Tasks

* Required
  + Search all Given columns in the table
  + AJAX-like searching functionality once the submit button is pressed.
  + Have the table be reset to its original state once the clear button is pressed.
* Outcomes
  + At first, the submit button just listed an empty table.
    - It was then identified that it was not indexing the databases correctly.
  + The search would only index 1 column and ignored the rest.
    - This was resolved by fixing the SQL function.

### Needs

In order to ensure that this functionality was acceptable and in space for testing we had to have test data. In order to do a full search functionality we would have to be able to search data, without any data that would be impossible.

### Pass/Fail Criteria

For our Pass and Fail Criteria, it was pretty simple. Our Pass Scenarios would ensure that it would successfully search every column and table to check for that specific given term from the user. Then once that functionality was confirmed successful we looked for our clear button. Our pass scenario for the clear button was just putting the table back to its original state. Our Fail case scenarios would be an empty table, a table that remained the same, wrong data shown, clear functionality not initializing, and buttons not initializing the scripts. These pass-and-fail scenarios were realized fairly quickly once attempting to implement this function.

## Specifications

Here is an example of the search and clear functionality code used. I will be providing both text and a screenshot.

private void button5\_Click(object sender, EventArgs e)

{

//Search Button

try

{

//search.SelectCommand = new MySqlCommand("SELECT \* FROM customer WHERE CONCAT('customerId', 'customerName', 'createdBy', 'lastUpdateBy') LIKE '" + textBox5.Text + "'", con);

search.SelectCommand = new MySqlCommand("SELECT \* FROM customer WHERE customerId LIKE '%" + textBox5.Text + "%' OR customerName LIKE '%" + textBox5.Text + "%' OR createdBy LIKE '%" + textBox5.Text + "%' OR lastUpdateBy LIKE '%" + textBox5.Text + "%'", con);

DataTable custTableView = new DataTable();

search.Fill(custTableView);

BindingSource customerSearch = new BindingSource();

customerSearch.DataSource = custTableView;

search.Update(custTableView);

dataGridView1.DataSource = custTableView;

}

catch (Exception x)

{

MessageBox.Show(x.ToString(), "Message", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button6\_Click(object sender, EventArgs e)

{

//Clear Button

try

{

{

cust.SelectCommand = new MySqlCommand("SELECT \* FROM customer", con);

DataTable custTableView = new DataTable();

cust.Fill(custTableView);

dataGridView1.DataSource = custTableView;

}

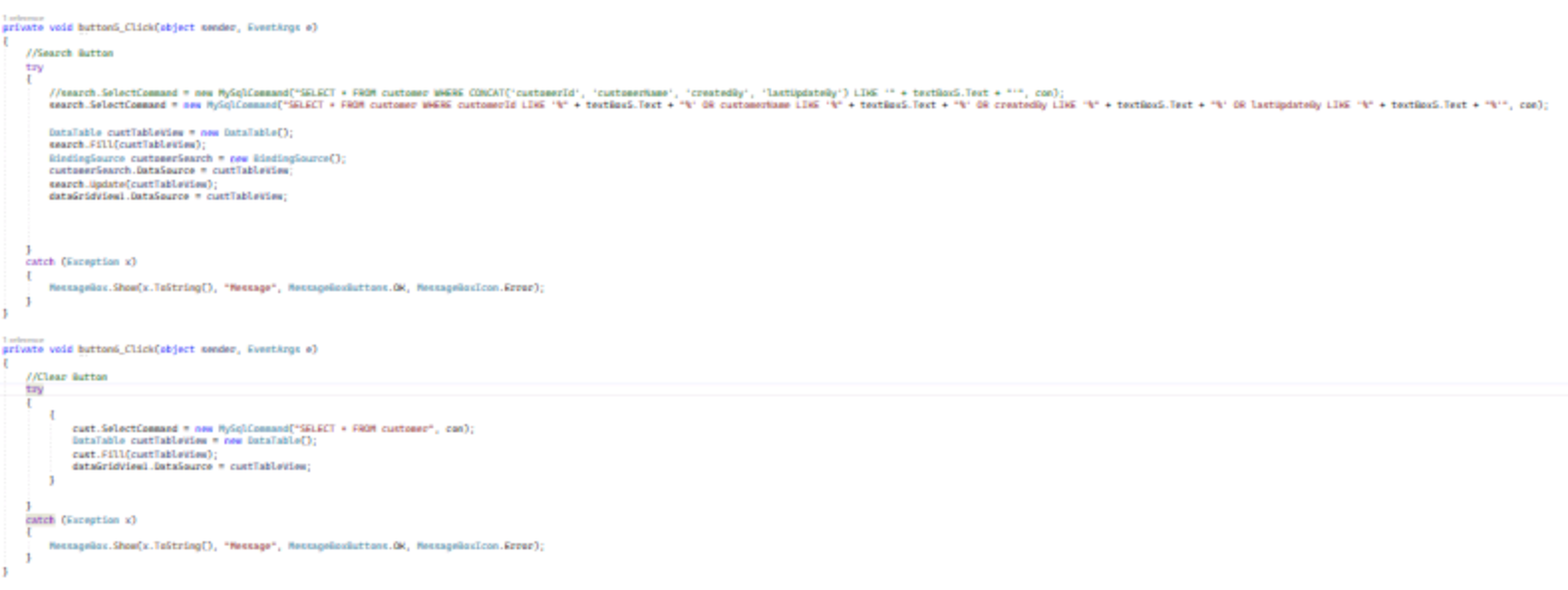
}

catch (Exception x)

{

MessageBox.Show(x.ToString(), "Message", MessageBoxButtons.OK, MessageBoxIcon.Error);

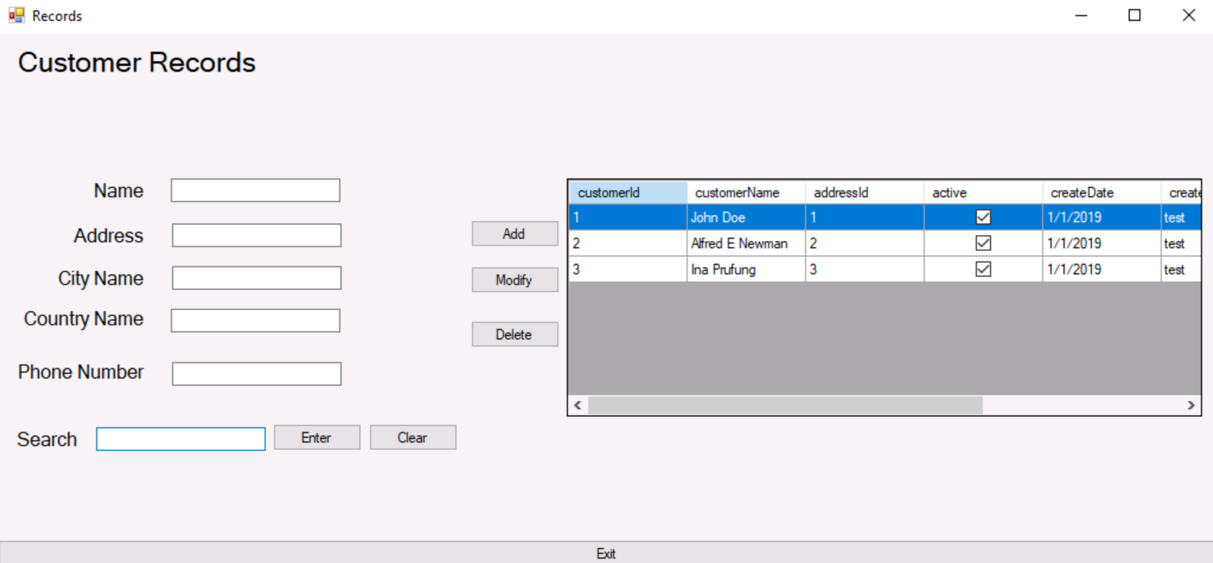
}

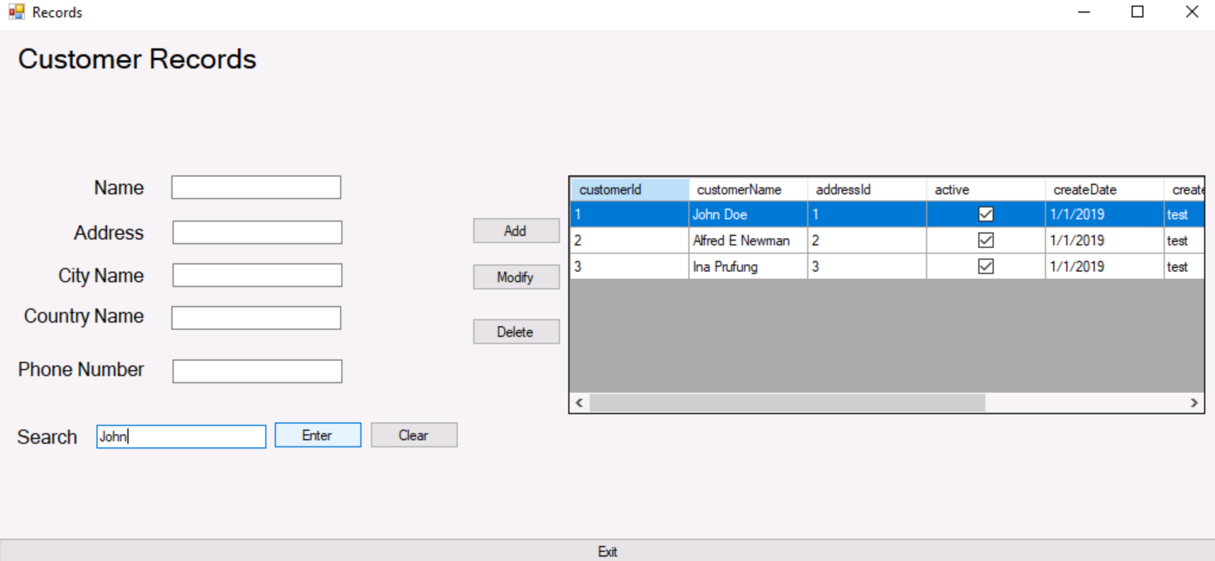
}

## Procedures

In my code, there is a commented-out original attempt at the functionality. I first attempted the function by using a CONCAT action but it failed and did not look for all of the data. In my second attempt, there was a discovery that the LIKE functionality for the MySQL search was exactly what was needed for this functionality to work. Using the CONCAT method ended up with results that were making the table empty. Once I changed the code again to filter through using a LIKE functionality, it did not work. I was missing the “%” lettering in the code in order for it to actually look through the other tables. It was only looking through the first column on the LIKE function which was “customerId” that was being searched through. Once adding the “%” in the command it populated through all of the additional columns to search and showed only the results with the matching item for that customer.

## Results

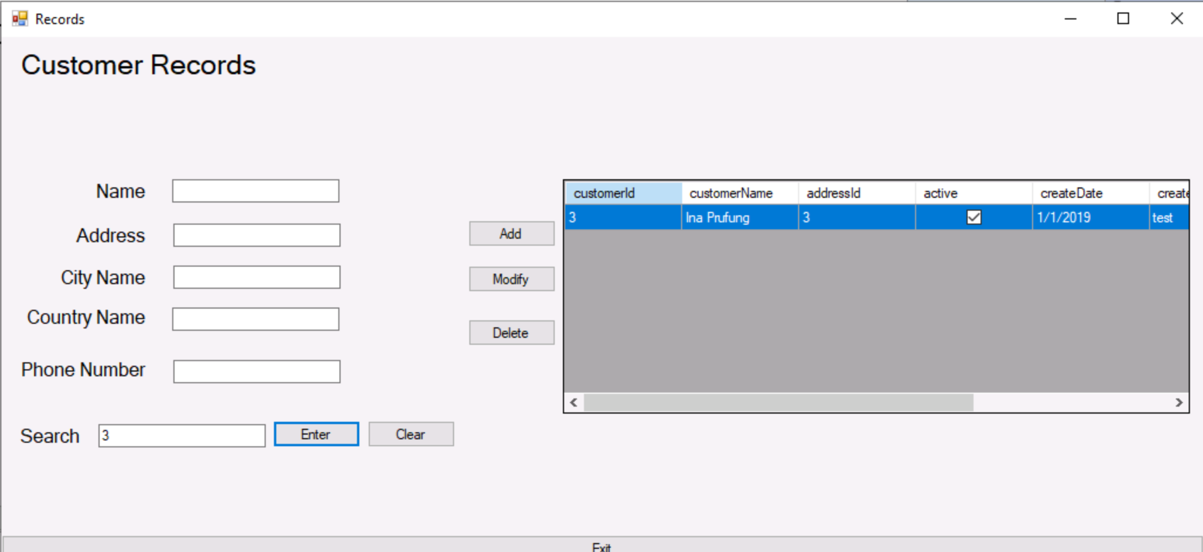
I will include a screenshot of the program searching through the given test data. I will be showing multiple screenshots to show the overall process of the search functionality. I will be inputting 3 different input variables to show the usability of the search function. The 3 inputs will be John, Alfred, and 3. They will show that it works for not only 2 forms of data but also different columns.



# 

A screenshot of a computer

Description automatically generated



# User Guide

## Introduction

This will be a guide for employee use. We will go over the implementation of the MySQL Database and how to set it up, installation, and a step-by-step process on how to use the program for its intended uses.

## Installation the Database and Using the Application

## *Installing the MySQL Database*

1. *Head over to mysql.com/products/workbench/*
2. *Once on this page, you will see a “Download Now” button underneath the Title “MySQL Workbench”.*
3. *Click the download button.*
4. *Your web browser will now load the download page for MySQL Workbench.*
5. *Click on the Select Version dropdown, and select the most currently up-to-date version.*
6. *The Select Operating System should be set to Windows for Windows Devices.*
7. *The two versions available will be very similar in appearance but are very different.*
   1. *Make sure to select the version with “web” in the file name.*
   2. *i.e.* (mysql-installer-web-community-8.0.35.0.msi)
8. *Once you click the download button you will then need to press the “No thanks, just start my download.” Link to begin the download.*
9. *Once the download has been completed, open the installer file.*
10. *In the installer file agree to the License Agreement then click the next button.*
11. *On the next page click on the Custom option.*
    1. *The other options are either too limited or unnecessary.*
12. *Once on the next page, click on the MySQL Servers dropdown.*
    1. *This will lead to two additional dropdowns, MySQL Server then MySQL then select the most recent version.*
       1. *i.e. 8.0.35.*
    2. *Then click the green arrow on the top pointing to the right to move it to the to-be-installed list.*
13. *Once the server is selected it will appear on the right list clarifying that you have selected this option.*
14. *Next, click on the application dropdown, and then select MySQL Workbench.*
    1. *Similar to the MySQL Server dropdown, select the most recent version*
       1. *i.e. 8.0.34*
    2. *Then click the green arrow on the top pointing to the right to move it to the to-be-installed list.*
15. *(Optional) It is recommended to download and review the MySQL Documentation.*
16. *Then click on next, and on the next page you will be greeted with a confirmation of the software that is selected.*
    1. *The two software that should be on this list is MySQL Server and MySQL Workbench.*
17. *Click on execute which will begin the installation process.*
    1. *Follow the onscreen prompts to ensure the setup process is going smoothly*
    2. *You will have to click execute a couple of times in this process.*
18. *Once installation has finished, hitting the next button will start the server configuration process.*
19. *Click on next to then see the next page, on this page select the Standalone MySQL Server / Classic MySQL Replication.*
20. *On the next page, there will be the Server Configuration Type.*
    1. *In the dropdown for Config Type, select the “Development Computer” option.*
    2. *For the Connectivity section, the options below need to be on and configured for these options.*
       1. *TCP/IP: set to on*
       2. *Port: 3306*
       3. *X Protocol Port: 33060*
       4. *Open Windows Firewall ports for network access: on*
21. *For the next page, Authentication Method, ensure that you are selecting the “Use strong password encryption for authentication” option. Click next to head to the next page.*
22. *For this page, enter “Passw0rd!” for both MySQL Root Password and Repeat Password options*
23. *Under the MySQL User Accounts, click “Add User” and follow the list below to input for each option*
    1. *Username: sqlUser*
    2. *Host: localhost*
    3. *Role: DB Admin*
    4. *Authentication: MySQL*
    5. *Password: Passw0rd!*
24. *For this next page, Windows Service, Ensure that all of these options listed are set to below.*
    1. *Configure MySQL Server as a Windows Service: Set to On*
    2. *Windows Service Name: “Default Option” (Optional, this can be changed)*
    3. *Start the MySQL Server at System Startup: Set to On*
    4. *Run Windows Service as..: Standard System Account*
25. *For this next page ensure that the first option for full access is enabled.*
26. *The apply configuration will be the last page before the server is up and running.*
27. *Once on the next page you are greeted with the Start MySQL, go ahead and leave that option turned on then click Finish.*

## *Setting up the Database References*

1. *If your MySQL Workbench application is not open, open it now.*
2. *Next to the MySQL Connections title, click the “+” icon to configure a new database connection.*
3. *For this page, you can follow the below for easier input.*
   1. *Connection Name: (Optional) Local Server (note, this can be set to anything)*
   2. *Connection Method: Standard (TCP/IP)*
   3. *Hostname: localhost*
   4. *Port: 3306*
   5. *Username: sqlUser*
   6. *Password: Passw0rd!*
      1. *Click on the “Store in Vault” button to reach this page.*
4. *Then click on Test Connection to ensure that the connection and credentials are valid.*
5. *Proceed and click ok, and we are now ready to start editing the database.*

## *Setting up the Tables in the Database*

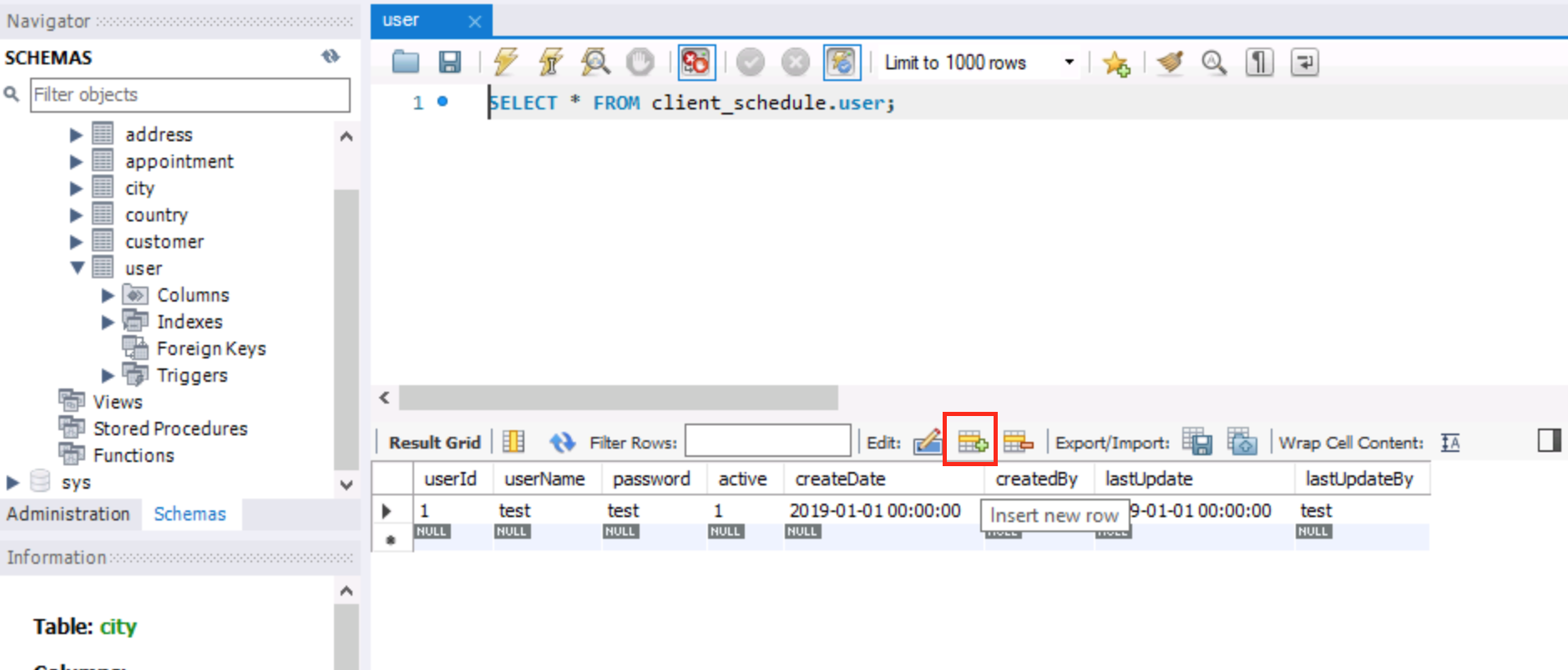
1. *If your MySQL Workbench application is not open, open it now.*
2. *If you do not have a database connection, please refer to the “Setting up the Database” section above.*
3. *Once the connection page has opened look for the icon that states “ Create a new schema in the connected server”*
   1. *An example screenshot will be below*A screenshot of a computer

      Description automatically generated*ow*
4. *A new schema page will pop up, for the Name: client\_schedule*
   1. *Then click Apply.*
5. *Then navigate to the Schemas tab to see the DataTables*
   1. *A screenshot will be below for easier navigation.*

*A screenshot of a computer

Description automatically generated*

1. *Click on the arrows next to client\_schedule to reveal Tables.*
2. *Right-click on tables and click Create Table*
3. *Once on the new table page, we will go over all the options needed for each table.*
   1. *This process is crucial for the functionality of the application.*
   2. *These new Tables will be split into a list below.*
      1. *address*
         1. *addressId – Int – PK – NN – AI*
         2. *address – varchar(50) – NN*
         3. *address2 – varchar(50) – NN*
         4. *cityId – Int – NN*
         5. *postalCode – varchar(10) – NN*
         6. *phone – varchar(20) – NN*
         7. *createDate – Datetime – NN*
         8. *createdBy – varchar(40) – NN*
         9. *lastUpdate – TimeStamp – NN – CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP*
         10. *lastUpdateBy – varchar(40) – NN*
      2. *appointment*
         1. *appointmentId – Int – PK – NN – AI*
         2. *customerId – Int – NN*
         3. *userId – Int – NN*
         4. *title – varchar(255) – NN*
         5. *description – Text – NN*
         6. *location – Text – NN*
         7. *contact – Text – NN*
         8. *type – Text – NN*
         9. *url – varchar(255) – NN*
         10. *start – DateTime – NN*
         11. *end – DateTime – NN*
         12. *createDate – DateTime – NN*
         13. *createdBy – varchar(40) – NN*
         14. *lastUpdate – TimeStamp – NN - CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP*
         15. *lastUpdateBy – varchar(40) – NN*
      3. *city*
         1. *cityId – Int - PK – NN – AI*
         2. *city – varchar(50) – NN*
         3. *countryId – int - NN*
         4. *createDate – DateTime – NN*
         5. *createdBy – varchar(40) – NN*
         6. *lastUpdate - TimeStamp – NN - CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP*
         7. *lastUpdateBy – varchar(40) – NN*
      4. *country*
         1. *countryId – Int – PK – NN -AI*
         2. *country – varchar(50) – NN*
         3. *createDate – DateTime – NN*
         4. *createBy – varchar(40) – NN*
         5. *lastUpdate – TimeStamp - NN - CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP*
         6. *lastUpdateBy – varchar(40) – NN*
      5. *customer*
         1. *customerId – Int – PK – NN – AI*
         2. *customerName – varchar(45) - NN*
         3. *addressId – Int – NN*
         4. *active – TinyInt(1) – NN*
         5. *createDate – DateTime – NN*
         6. *createdBy – varchar(40) – NN*
         7. *lastUpdate - TimeStamp - NN - CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP*
         8. *lastUpdateBy – varchar(40) – NN*
      6. *user*
         1. *userId – Int – PK – NN – AI*
         2. *userName – varchar(50) – NN*
         3. *password – varchar(50) – NN*
         4. *active – TinyInt – NN*
         5. *createDate – DateTime – NN*
         6. *createdBy – varchar(40) – NN*
         7. *lastUpdate - TimeStamp - NN - CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP*
         8. *lastUpdateBy – varchar(40) – NN*
4. *Once done with creating all tables and their respective columns go to the user table and add a new row.*
   1. *A screenshot will be included.*

**

* 1. *Following the screenshot above input all the data into that new row*
     1. *userId: 1*
     2. *userName: test*
     3. *password: test*
     4. *active: 1*
     5. *createDate: (Can be any time)*
     6. *createdBy: test*
     7. *lastUpdate: (Can be any time)*
     8. *lastUpdateBy: test*

## *Login into the Program*

1. *Once the database has been set up, we can now open and log into the program.*
2. *Once the program has been started you will then be greeted with the login screen.*
3. *If you followed the MySQL Database setup as shown above, you can use “test” as both the username and password to open the app.*
   1. *If you have not set up the database please do so now and refer back to those instructions.*
   2. *If you cannot log in with test, or your chosen user, please refer back to instructions 9.b of “Setting up the Tables in the Database”*

## *Directory*

*Once you have been logged in, you will be greeted with the Directory Page. There are options consisting of Customer Records, Appointments, Calendar, and Reports. We will now go over the functionality of each page and how to use them.*

### *Customer Records*

1. *To get to the Customer Records Page, click on the Open button next to Customer Records.*
2. *From here the page will load up all of the customers from the customer table.*

### *Adding, Modifying, Deleting, and Searching the Customer Records Page*

1. *First, we will begin by adding a customer*
   * 1. *To Add a customer all boxes on this screen will need to be filled minus the Search box.*
     2. *Once all boxes are full click add to then create a new customer.*
2. *Now we can modify a customer.*
   * 1. *To populate the boxes with a pre-existing customer, click on any row and it will then be inputted into the boxes.*
     2. *You can change any data in these fields.*
     3. *Once ready, click on the modify button and it will then submit those changes,*
3. *Then we have the option to delete a customer.*
   * 1. *Similar to the modify section we will click on the customer row.*
     2. *Then click the delete button to delete the selected customer.*
4. *Finally, we have the search functionality.*
   * 1. *In the text box, enter any data that you would like to search for from the table on this page.*
     2. *Once ready click the Enter button.*
     3. *Once done with the search you can then click Clear or the exit button to leave.*

## *Appointments*

1. *To get to the Appointments page click the open button next to it on the Directory Page.*
2. *Once the appointment page has been opened, the data from the appointments data will populate into the grid.*

### *Adding, Modifying, Deleting, and Searching the Customer Appointments Page*

1. *First, we will begin by creating an Appointment*
   * 1. *Similar to creating a customer, we will have to ensure all data fields are filled.*
     2. *Once all boxes are full click add to then create a new appointment.*
2. *We can now go over how to Modify an appointment.*
   * 1. *To populate the boxes with a pre-existing appointment information, click on any row and it will then be inputted into the boxes.*
     2. *You can change any data in these fields.*
     3. *Once ready, click on the modify button and it will then submit those changes,*
3. *Onto the Delete an appointment.*
   * 1. *Similar to the modify section we will click on the appointment row.*
     2. *Then click the delete button to delete the selected appointment.*
4. *Lastly, we have the search functionality.*
   * 1. *In the text box, enter any data that you would like to search for from the table on this page.*
     2. *Once ready click the Enter button.*
     3. *Once done with the search you can then click Clear or the exit button to leave.*

## *Calendar*

1. *To get to the Calendar page click the open button next to it on the Directory Page.*
2. *The Calendar page has very few functions but they are essential.*
3. *Once the page is open it will load a quick view of some of the data from the Appointments table.*
4. *There are three options to choose from on this page.*
   1. *Show All*
      1. *This option will show all the appointment information that is shown when the page is loaded.*
   2. *This Week*
      1. *This option will show the appointment information that is currently set for that given week.*
   3. *This Month*
      1. *This option will show the appointment information that is currently set for that given month.*

## *Reports*

1. *To get to the Reports page click the open button next to it on the Directory Page.*
2. *Once this page has been opened there will be an empty grid. No information is displayed as the system is waiting for your input.*
3. *There are three buttons on this page with varying functions.*
   1. *Appointment Type by Month*
      1. *The picker above this button will change the data shown as the user can change which months will be shown in the reports.*
      2. *This option will change the grid to show all appointment types for each given month selected in the picker.*
   2. *Appointments for Users*
      1. *The picker for this button will show a list of all current users and allow you to pick which one you’d like to see a report on.*
      2. *Once a user has been selected and the button has been pressed, all the appointment data from that user will be shown.*
   3. *All Current Customers*
      1. *This button will show all of the current customers in the customer table.*