

C868 – Software Capstone Project Summary

Task 2 – Section C



Capstone Proposal Project Name: AppTrack- Appointment Tracker

Student Name: Anton Tran

Table of Contents

Table of Contents

Table of Contents.....	2
Application Design and Testing.....	4
Design Document.....	4
Class Design.....	4
UI Design.....	6
Test Plan.....	6
Introduction.....	6
Purpose.....	6
Overview.....	7
Test Plan.....	7
Items.....	7
Functions/Features.....	8
Deliverables/Outcomes.....	8
Tasks.....	8
Needs.....	9
Pass/Fail Criteria.....	9
Specifications.....	10

Procedures.....	10
Results.....	11
Source Code.....	12
User Guide.....	12
Installation and Using the Application.....	12
Login with an Admin Account.....	18
Adding Customers.....	19
Updating Customers.....	21
Searching Customer.....	23
Add Appointments.....	24
Updating Appointments.....	26
Deleting Appointments and Customers.....	27
Radio Button for filtering Appointments.....	29
Reports.....	30

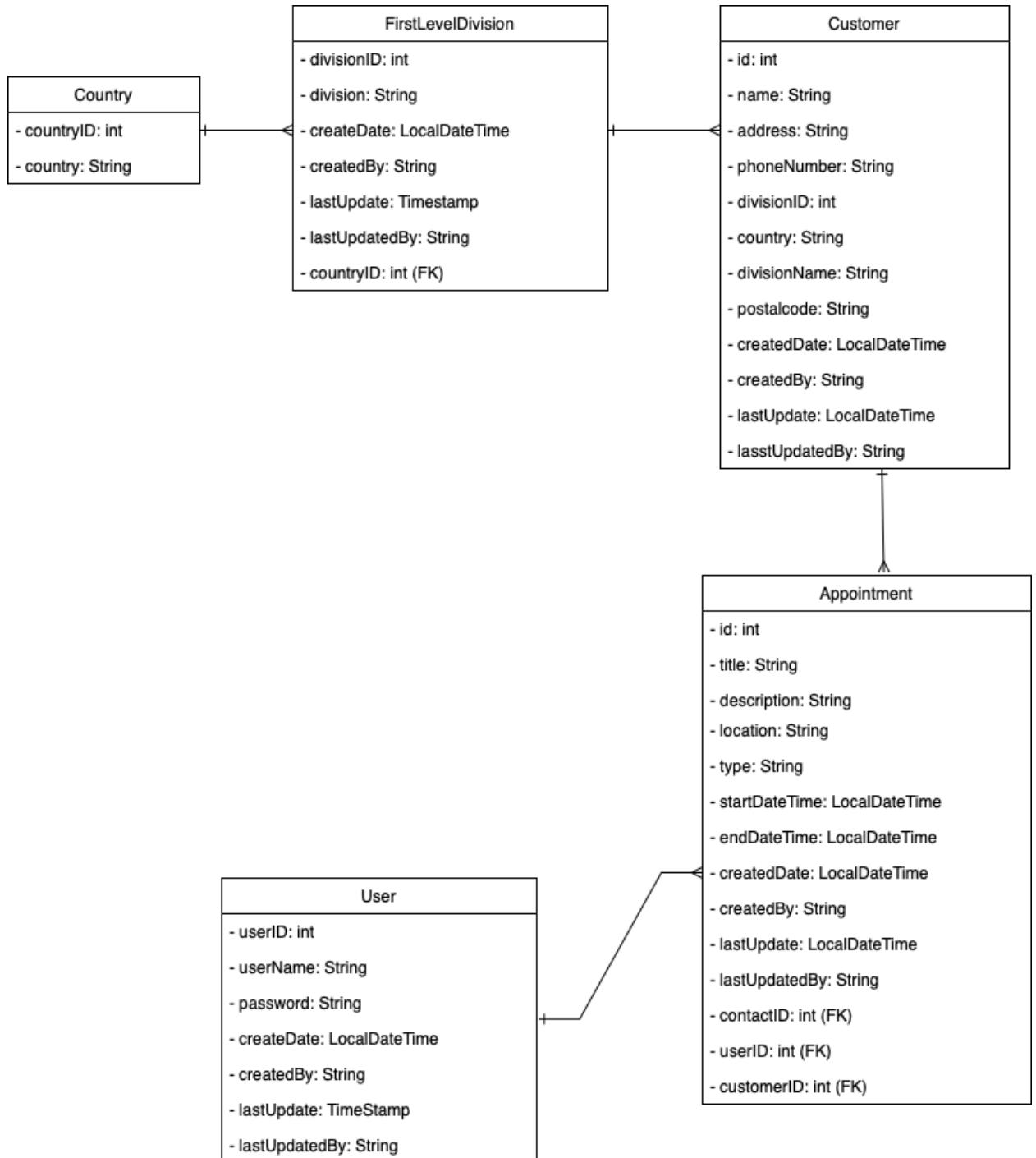
Task 2 Part C – C868 Software Development Capstone

Application Design and Testing

Design Document

Class Design

The class diagram provided below is for the AppTrack software application. The application has 5 model objects: Customer, User, Appointment, FirstLevelDivision, and Country. There is a one-to-many relationship between the Country and the FirstLevelDivision(Each division/state can only belong to one country). The FirstLevelDivision has a one-to-many relationship with the Customer(Many customers can reside within the same state) and the Customer has a one-to-many relationship with the Appointment(Each customer can have one or more appointments). The Customer has a many-to-one relationship with the user(A user can hold many customers). A customer can have appointments scheduled for them by the user.



UI Design

The pictures provided below show the high fidelity design. After the login is successful the main page will be displayed. From the main page you will be able to access all other features like adding, updating, and deleting customers and their appointments. You'll also be able to access the reports form.

figure 1. Login Form

figure 2. Main Page of application for tracking appointments and customer information

figure 3. Add/update appointment form

figure 4. Add/update customer form

figure 5. reports form

Unit Test Plan

Introduction

Purpose

The main purpose of this software application is to track customer appointments. In order to ensure the application function smoothly I had to test the validation of the

AppointmentController class. In order to do this I had to create unit tests using IntelliJ IDEA and write all test cases in java. I needed to make sure that an appointment cannot be duplicated when an employee adds an appointment to the table. Another test that I needed to validate was to make sure that a customer appointment is within business hours. These tests ensure that the customer appointments are valid.

Overview

The first test checks to ensure that duplicate appointments don't overlap. This ensures that the database is free of redundant data and that customers cannot have overlapping appointments.

The second test checks to ensure that appointments being saved into the database are within business hours. This prevents appointments to be saved to the database if it's outside business hours. Both unit tests are written in java and are executed in IntelliJ IDEA. After the tests are executed the run tool at the bottom of the IDE will list the status of the tests that were run whether they passed or failed. IntelliJ provides detailed information about the failure, including the expected and actual results.

Test Plan

Items

To test that an overlapping appointment cannot be created, we need to make an instance of the appointment controller. We will also need to create an appointment list in order to hold the appointment object. We will then add two appointments with one start time being now and the end time in an hour to the list. Now we're going to create a new start and end date time with the start being two days from now and the end being two days plus one hour. We'll then call the isAppointmentOverlapping method with the new start and end date time as the parameters along

with the existing appointments. The expected result is that the new appointment should not overlap with any existing appointments. The assertFalse assertion checks that the result is false.

This second unit test is to ensure that the appointment is created within business hours.

The method creates an instance object of the appointment controller. We then create a new local date time and pass it into the parameter of the method isWithinBusinessHours. The assertTrue assertion will return true if the appointment is within business hours.

Features

- Instance of AppointmentController
- Observable Array list that holds a list of appointments
- Appointment class instance
- LocalDateTime object

Deliverables

When the test is run, IntelliJ IDEA will produce the result into the console output. The console will display a “Test passed. Appointment does not Overlap” or “Test Passed. Appointment is within business hours” if both test cases pass. If the test fails the console will display the test failed as well as the actual and expected results.

Tasks

To execute this unit test the following steps are required:

1. Create a test folder.
2. Create the test for which method you'll be testing inside the test folder.
3. Write the code to be tested.
4. run the test by right clicking on the test and click the run prompt.
5. Examine the output to determine if the test passed or failed.

Needs

In order to unit test our project requires two dependencies in our pom.xml file

- “junit-jupiter-api”: This API provides the annotations and assertions used for writing JUnit Jupiter tests.
- “junit-jupiter-engine”: This is the engine that executes the tests written using JUnit Jupiter.

Pass/Fail Criteria

The criteria for a successful unit test is the console outputs a test passed message as well as the left run pane displays green checks. A failed test will appear red and the console will display an error message as well as the actual and expected results.

Specifications

This is a screenshot of the test code described above.

```
class AppointmentControllerTest {  
  
    @Test  
    void isAppointmentOverlapping() {  
  
        AppointmentController appointmentController = new AppointmentController();  
  
        ObservableList<Appointment> appointments = FXCollections.observableArrayList();  
        appointments.add(new Appointment(1, "Computer Repair", LocalDateTime.now(), LocalDateTime.now().plusHours(1)));  
        appointments.add(new Appointment(2, "Battery Component Repair", LocalDateTime.now().plusDays(1), LocalDateTime.now().plusDays(1).plusHours(1)));  
  
        LocalDateTime newStartTime1 = LocalDateTime.now().plusDays(2);  
        LocalDateTime newEndTime1 = LocalDateTime.now().plusDays(2).plusHours(1);  
        boolean result1 = appointmentController.isAppointmentOverlapping(newStartTime1, newEndTime1, appointments);  
        assertFalse(result1);  
  
        System.out.println("Test passed. Appointment does not overlap");  
    }  
}
```

```
@Test  
void isWithinBusinessHours() {  
    AppointmentController appointmentController = new AppointmentController();  
  
    LocalDateTime dateTime = LocalDateTime.of( year: 2023, month: 9, dayOfMonth: 25, hour: 10, minute: 0); // Monday, 10 AM  
  
    boolean result = appointmentController.isWithinBusinessHours(dateTime);  
  
    assertTrue(result, message: "Expected within business hours");  
}  
}
```

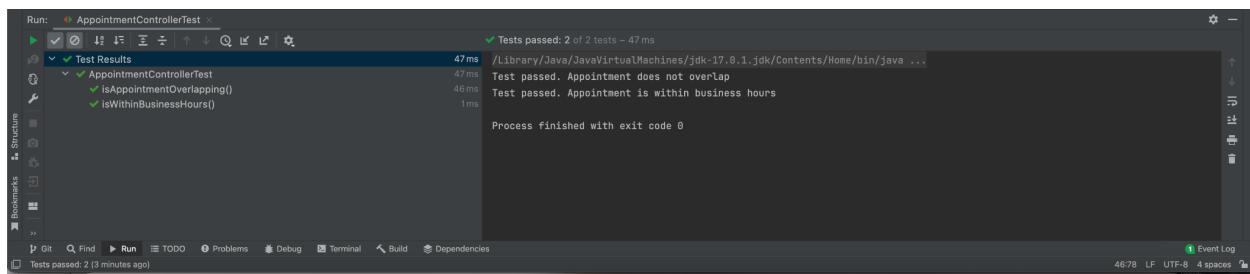
Procedures

The first test that we are testing is the “isAppointmentOverlapping” method. This test ensures that appointments do not overlap with any other appointments. This test adds a new appointment with a start date that is two days ahead and an end date that is two days and an hour

ahead of the other appointment saved in the database. The test should return a message in the console displaying, “Test passed. Appointment does not overlap”. If the test failed the console will display a “test failed” as well as the actual and expected result. For the second test we are testing the “isWithinBusinessHours” method. This test ensures that no appointment that gets added to the database is outside business hours. This test adds a date time that is within business hours. I used September 25th, 2023 10:00 AM which is during business hours and passed this value into the isWithinBusinessHours method. The test passed and the console displays, “Test passed. Appointment is within business hours”. If the test fails the console displays a test failed with the following error message, “Expected within business hours”. The console also displays expected and actual results.

Results

Both tests passed and on the left pane shows a green check mark next to the unit test method that passed. In the middle is the console. The top displays the total number of test that passed and below that has a console message stating that the test passed and the appointment does not overlap and is within business hours.



C4. Source Code

All source code for the application is submitted in the .zip file with the other materials.

User Guide

Note: This may be included as a separate document if you desire.

Introduction

The following user guide's purpose is to document steps to install and run the application.

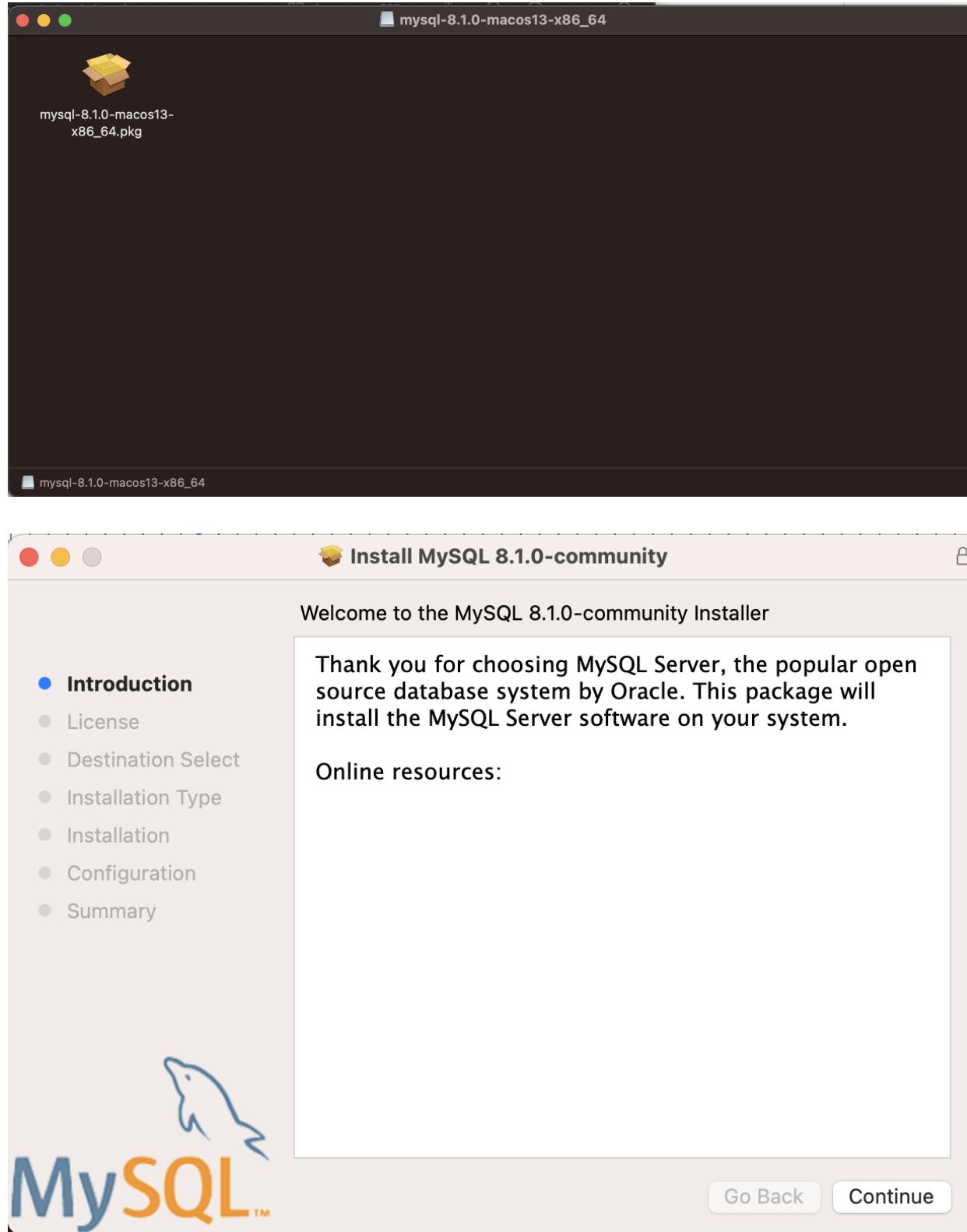
Installation and Using the Application

When creating the mysql database use “root” as username and “Passw0rd!” for the password.

1. Download MySQL Community Edition for Mac

<https://dev.mysql.com/downloads/mysql/>

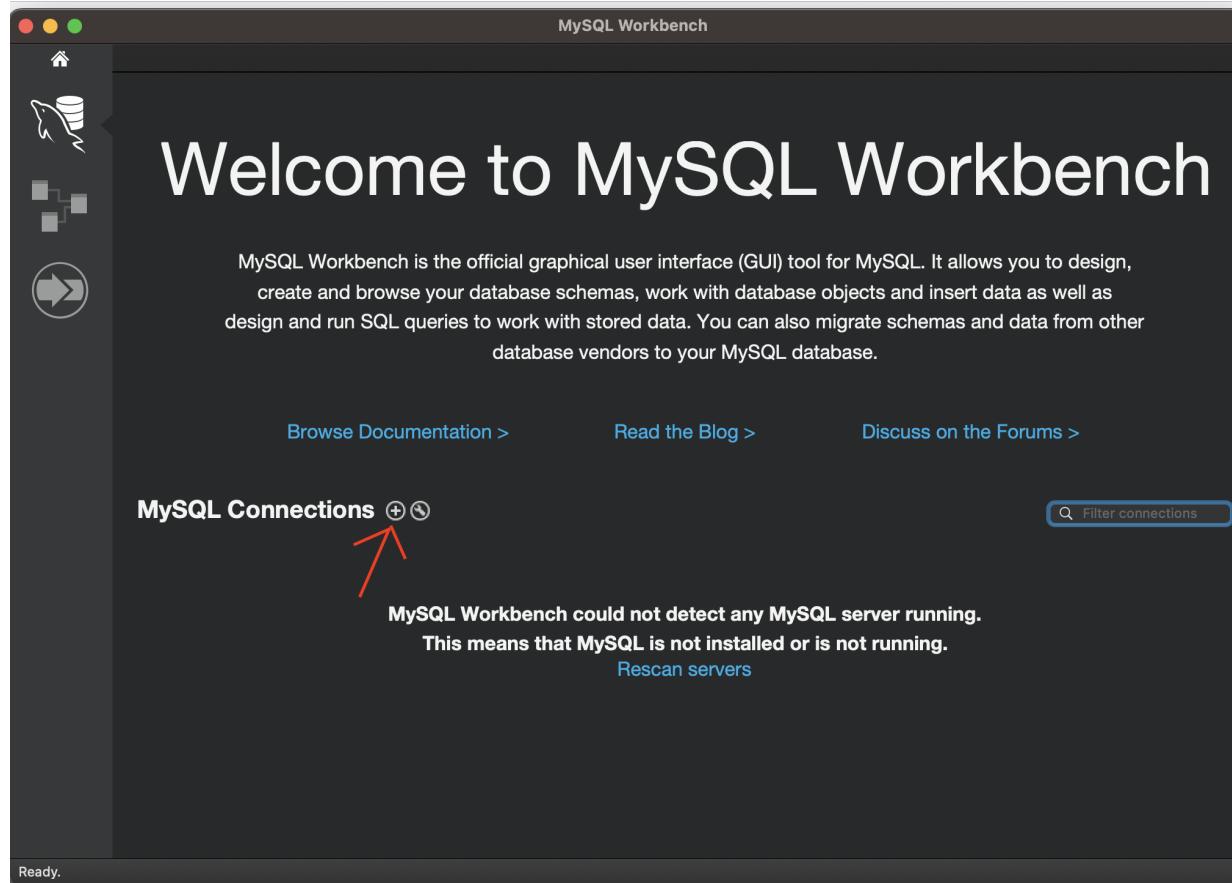
2. Open up the downloaded folder and complete the setup wizard



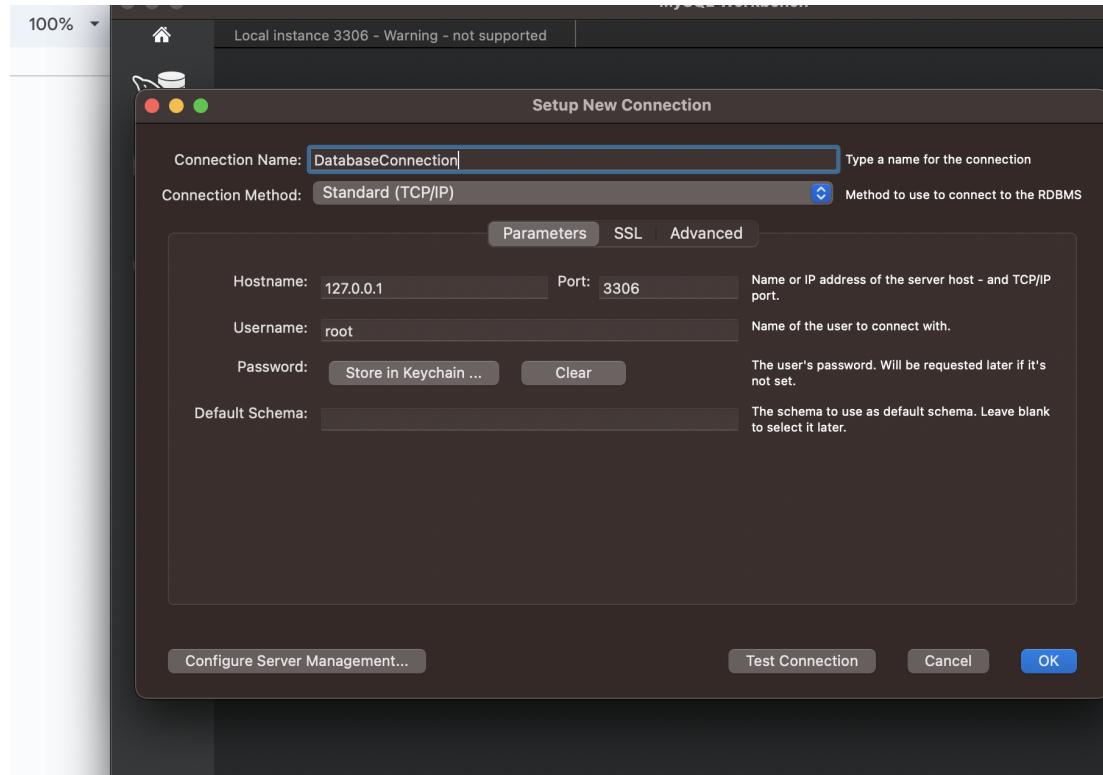
3. Install MySQL Workbench <https://dev.mysql.com/downloads/workbench/>
4. Drag the MySQL Workbench over to the application folder and open the application



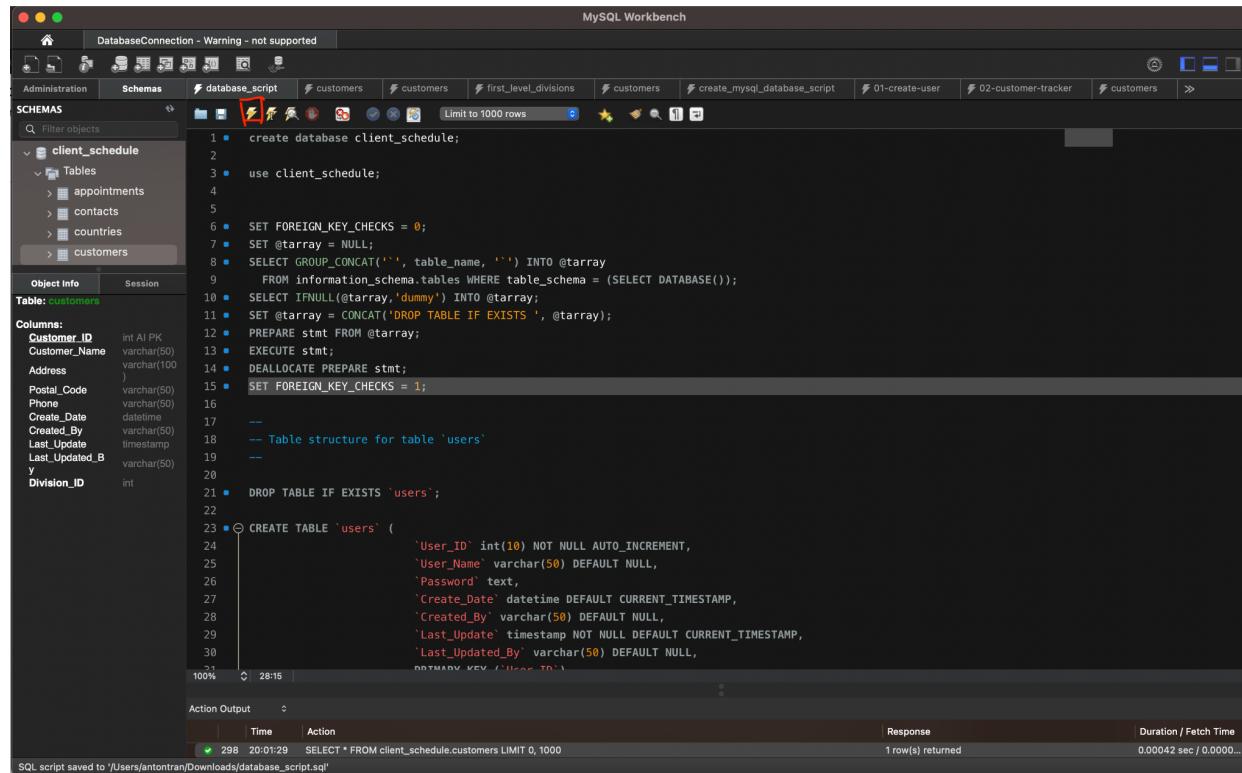
5. Click on the plus(+) icon



6. For Connection Name write “DatabaseConnection” then click “OK”



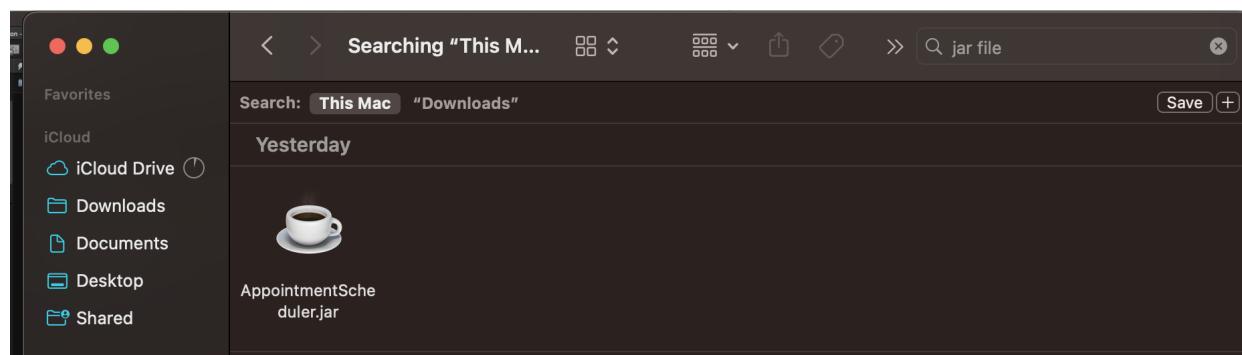
7. Load the sql script into the query view and click the execute(thunderbolt) icon.
 (sql script located with all other file attachments)



```

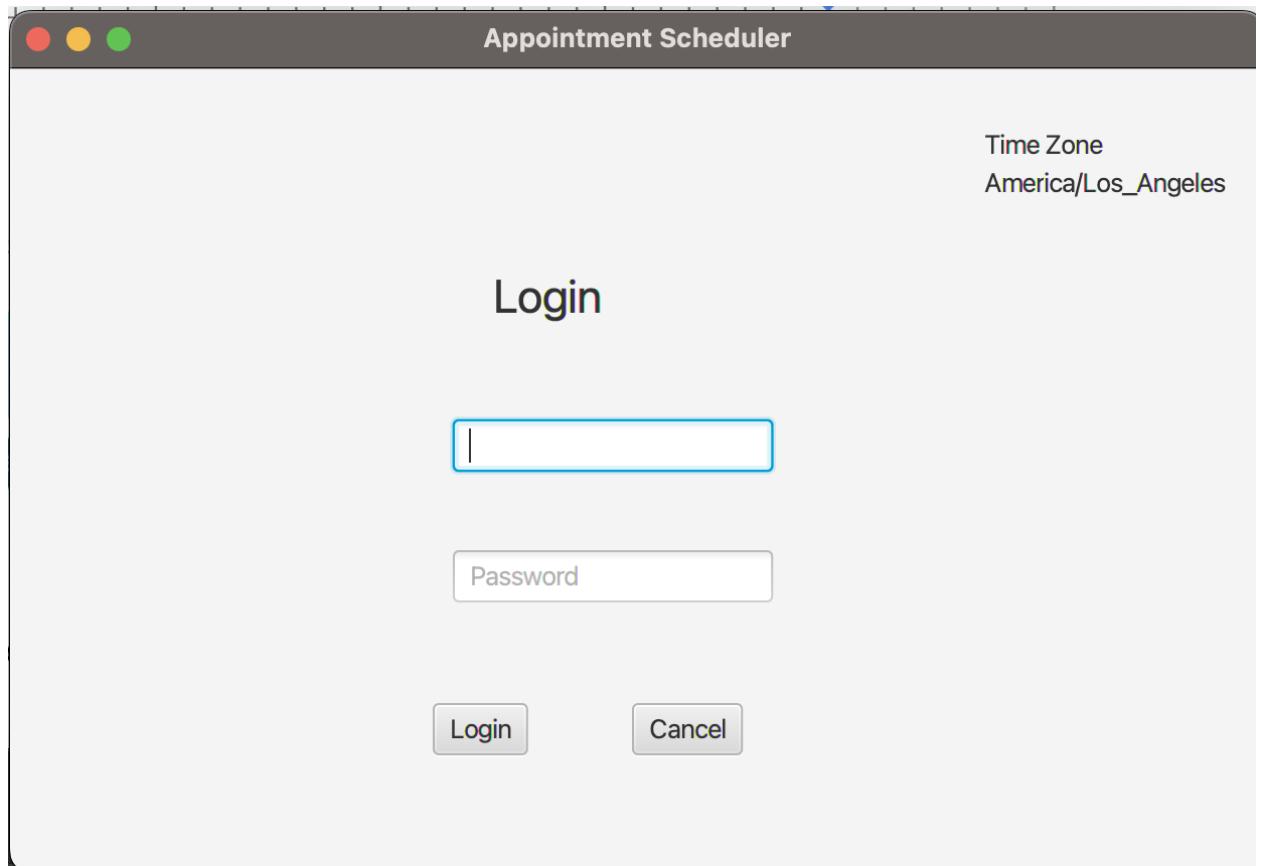
1 create database client_schedule;
2
3 use client_schedule;
4
5
6 SET FOREIGN_KEY_CHECKS = 0;
7 SET @array = NULL;
8 SELECT GROUP_CONCAT('`', table_name, '`') INTO @array
9   FROM information_schema.tables WHERE table_schema = (SELECT DATABASE());
10 SELECT IFNULL(@array, 'dummy') INTO @array;
11 SET @array = CONCAT('DROP TABLE IF EXISTS ', @array);
12 PREPARE stmt FROM @array;
13 EXECUTE stmt;
14 DEALLOCATE PREPARE stmt;
15 SET FOREIGN_KEY_CHECKS = 1;
16
17 --
18 -- Table structure for table `users`
19 --
20
21 DROP TABLE IF EXISTS `users`;
22
23 CREATE TABLE `users` (
24   `User_ID` int(10) NOT NULL AUTO_INCREMENT,
25   `User_Name` varchar(50) DEFAULT NULL,
26   `Password` text,
27   `Create_Date` datetime DEFAULT CURRENT_TIMESTAMP,
28   `Created_By` varchar(50) DEFAULT NULL,
29   `Last_Update` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
30   `Last_Updated_By` varchar(50) DEFAULT NULL,
31   PRIMARY KEY (`User_ID`)
32 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
  
```

8. Open up the jar file to load the application. (file located with all other file attachments)



Login (An example)

1. For username and password type “admin” then click “Login”



Customers

2. Once logged in, This will enable you to create new appointments and customers. First you need to add a valid customer. Click on Add below the Customers table.

The screenshot shows a desktop application window titled "Appointment Scheduler". At the top, there are three colored window control buttons (red, yellow, green) on the left, and "Appointment Scheduler" on the right. Below the title bar are three radio buttons: "Current Week", "Current Month", and "All Appointments", with "All Appointments" selected. To the right of these buttons are "Reports" and "Logout" buttons. A horizontal table header row is displayed with columns labeled: ID, Title, Description, Type, Location, Start Date/Time, End Date/Time, Customer ID, and User ID. Below this header, a message "No content in table" is centered. At the bottom right of this section are "Add", "Update", and "Delete" buttons. The next section is titled "Customers" and contains a search bar labeled "Search Customer". A horizontal table header row is shown with columns labeled: ID, Name, Address, Phone Number, State, and Postal Code. Below this header, another "No content in table" message is centered. At the bottom right of this section are "Add", "Update", and "Delete" buttons. The entire application has a clean, modern design with a light gray background and white text.

3. Provide the appropriate credentials for the customer and click “Save” to save it to the database or click “Cancel” to close out of the form.

The screenshot shows a window titled "Add Form" with a dark header bar and three circular window control buttons (red, yellow, green) in the top-left corner. The main content area has a title "Add Customer" centered at the top. Below the title are seven input fields arranged in two columns. The first column contains "ID" (Auto Generated ID), "Name" (John Appleseed), "Address" (123 Cloverlane), and "Phone Number" (1234567890). The second column contains "Country" (U.S.), "State" (Alabama), and "Postal Code" (12345). At the bottom right of the form are two buttons: "Save" and "Cancel".

ID	Auto Generated ID
Name	John Appleseed
Address	123 Cloverlane
Phone Number	1234567890
Country	U.S.
State	Alabama
Postal Code	12345

Save Cancel

4. Select on a customer row then click “Update” under the Customers table to update an existing customer.

The screenshot shows the 'Appointment Scheduler' application interface. At the top, there are three radio buttons for filtering appointments: 'Current Week', 'Current Month', and 'All Appointments'. The 'All Appointments' button is selected. To the right are 'Reports' and 'Logout' buttons. Below this is a table header for 'Appointments' with columns: ID, Title, Description, Type, Location, Start Date/Time, End Date/Time, Customer ID, and User ID. A message 'No content in table' is displayed below the table. At the bottom of the screen, there are 'Add', 'Update', and 'Delete' buttons. The 'Customers' section is located at the bottom left, featuring a search bar labeled 'Search Customer'. A table displays customer information with columns: ID, Name, Address, Phone Number, State, and Postal Code. One row is highlighted with a blue border, corresponding to the data in the 'Appointments' table. The highlighted row contains the following data:

ID	Name	Address	Phone Number	State	Postal Code
1	John Appleseed	123 Cloverlane	1234567890	Alabama	12345

5. Change any field on the form and then click “Save”. (In the example I changed the state from Alabama to Washington).

The screenshot shows a window titled "Add Customer". The window has a dark grey header bar with three colored circles (red, yellow, green) on the left. The main area contains fields for customer information:

ID	1
Name	John Appleseed
Address	123 Cloverlane
Phone Number	1234567890
Country	U.S
State	Washington
Postal Code	12345

At the bottom right are two buttons: "Save" and "Cancel". The "State" field is highlighted with a blue border.

6. To search for a specific customer you can use the customer search bar located above the customers table.

The screenshot shows the 'Appointment Scheduler' application interface. At the top, there are three colored window control buttons (red, yellow, green) and a title bar labeled 'Appointment Scheduler'. Below the title bar are three radio buttons: 'Current Week' (unchecked), 'Current Month' (unchecked), and 'All Appointments' (checked). To the right of these buttons are 'Reports' and 'Logout' links. A horizontal menu bar below the title bar contains the word 'Appointments' and a series of columns headers: ID, Title, Description, Type, Location, Start Date/Time, End Date/Time, Customer ID, and User ID. A large table below this menu is currently empty, displaying the message 'No content in table'. In the bottom right corner of this table area are three buttons: 'Add', 'Update', and 'Delete'. Below the table, there is another section titled 'Customers' with a 'Search Customer' input field containing the text 'John Appleseed'. This search term is highlighted with a blue border. A table below this search bar has columns: ID, Name, Address, Phone Number, State, and Postal Code. It contains one row of data: ID 1, Name John Appleseed, Address 123 Cloverlane, Phone Number 1234567890, State Washington, and Postal Code 12345. At the bottom of this table are three buttons: 'Add', 'Update', and 'Delete'. Navigation arrows (< and >) are located at the bottom of the customer table.

Appointments

7. Click on the “Add” button located directly under the Appointments table to add an appointment.

The screenshot shows a desktop application window titled "Appointment Scheduler". At the top, there are three colored window control buttons (red, yellow, green). Below the title bar are three radio buttons: "Current Week", "Current Month", and "All Appointments", with "All Appointments" selected. To the right are "Reports" and "Logout" buttons. The main area has a header "Appointments" with a table header row containing columns for ID, Title, Description, Type, Location, Start Date/Time, End Date/Time, Customer ID, and User ID. A message "No content in table" is displayed below the table. At the bottom of the appointments section are "Add", "Update", and "Delete" buttons. Below this is a "Customers" section with a "Search Customer" input field and a table header row for ID, Name, Address, Phone Number, State, and Postal Code. One customer record is listed: ID 1, Name John Appleseed, Address 123 Cloverlane, Phone Number 1234567890, State Washington, Postal Code 12345. At the bottom of the customers section are "Add", "Update", and "Delete" buttons. Navigation arrows are visible at the bottom of both tables.

ID	Name	Address	Phone Number	State	Postal Code
1	John Appleseed	123 Cloverlane	1234567890	Washington	12345

8. Fill out the form and make sure that the customer id matches a customer. Click “Save” when done or click on “Cancel” to exit the form.

Add Form

Add Appointment

ID	Auto Generated ID
Title	Computer Repair
Type	Desktop
Description	Malfunction component
Location	New York
Start Date/Time	8/30/2023 <input type="button" value="Calendar"/> 09:00 <input type="button" value="Down"/>
End Date/Time	8/31/2023 <input type="button" value="Calendar"/> 09:30 <input type="button" value="Down"/>
User ID	admin <input type="button" value="Down"/>
Customer ID	1

9. Select an appointment row and click the “Update” button under the appointments table in order to update the selected appointment.

The screenshot shows a desktop application window titled "Appointment Scheduler". At the top, there are three radio buttons: "Current Week", "Current Month", and "All Appointments", with "All Appointments" being selected. To the right are "Reports" and "Logout" buttons. Below the buttons is a section titled "Appointments" containing a table with columns: ID, Title, Description, Type, Location, Start Date/Time, End Date/Time, Customer ID, User ID, and a delete icon. A single row is selected, highlighted with a blue border. At the bottom of this section are "Add", "Update", and "Delete" buttons. Below this is a "Customers" section with a search bar labeled "Search Customer". It contains a table with columns: ID, Name, Address, Phone Number, State, and Postal Code. One customer record is listed: ID 1, Name John Appleseed, Address 123 Cloverlane, Phone Number 1234567890, State Washington, Postal Code 12345. At the bottom of this section are also "Add", "Update", and "Delete" buttons. Navigation arrows are visible between the two sections.

ID	Title	Description	Type	Location	Start Date/Time	End Date/Time	Customer ID	User ID	
1	Comput...	Desktop	New York	Malfuncti...	2023-08-30 09:00 AM	2023-08-31 09:00 AM	1	1	

ID	Name	Address	Phone Number	State	Postal Code
1	John Appleseed	123 Cloverlane	1234567890	Washington	12345

10. Fix a field and click “Save” to proceed with the changes. (In the example we changed the appointment start and end date).

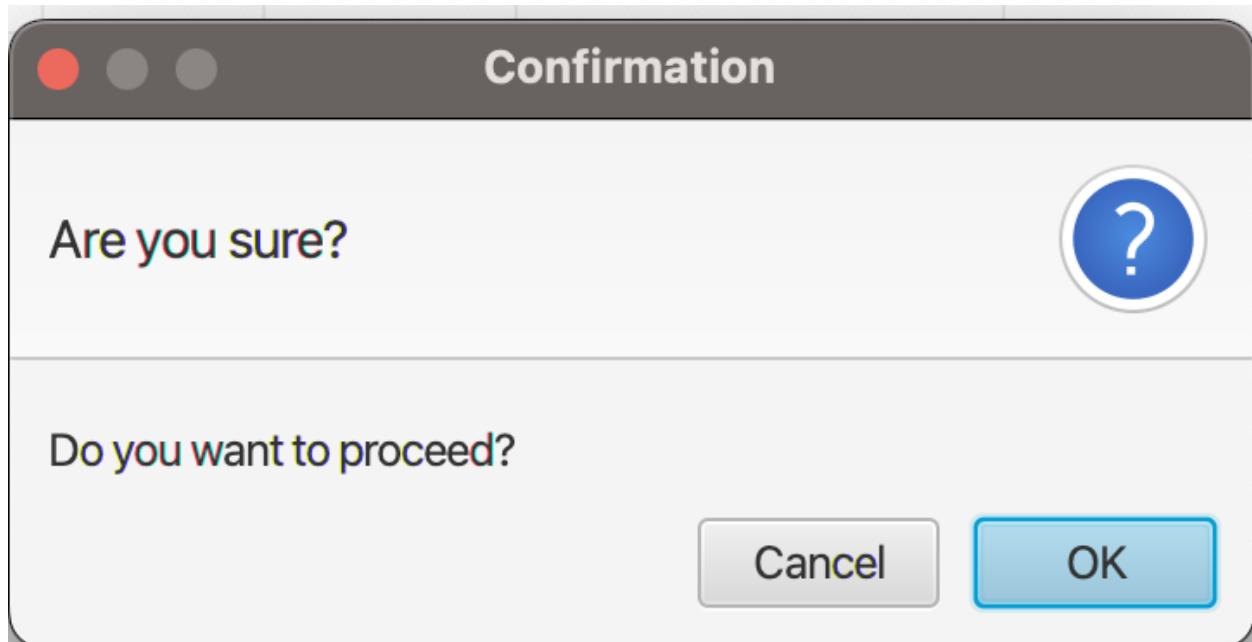
The screenshot shows a window titled "Add Appointment". The form contains the following fields:

- ID: 1
- Title: Computer Repair
- Type: New York
- Description: Desktop
- Location: Malfunction component
- Start Date/Time: 9/1/2023 (highlighted with a blue border)
- End Date/Time: 9/4/2023
- User ID: admin
- Customer ID: 1

At the bottom right are "Save" and "Cancel" buttons.

11. To delete an appointment or customer simply select on the row and click the “Delete” button that's associated with its specific table. Then click “OK” to confirm the

removal.



12. The 3 radio selections on the top of the application allows you to filter out the appointments by current week, month, or all appointments.

The screenshot shows the 'Appointment Scheduler' application interface. At the top, there are three radio buttons: 'Current Week' (unchecked), 'Current Month' (unchecked), and 'All Appointments' (checked). To the right are 'Reports' and 'Logout' buttons. Below this is a table header with columns: ID, Title, Description, Type, Location, Start Date/Time, End Date/Time, Customer ID, and User ID. A message 'No content in table' is displayed below the table. At the bottom, there are 'Add', 'Update', and 'Delete' buttons. The 'Customers' section includes a search bar labeled 'Search Customer'. A table lists one customer: ID 1, Name John Appleseed, Address 123 Cloverlane, Phone Number 1234567890, State Alabama, and Postal Code 12345. Navigation arrows are at the bottom of the customer table, and another set of 'Add', 'Update', and 'Delete' buttons is located at the very bottom.

ID	Name	Address	Phone Number	State	Postal Code
1	John Appleseed	123 Cloverlane	1234567890	Alabama	12345

Reports

13. Click on the “Reports” button to view the reports page.

The screenshot shows the 'Appointment Scheduler' application window. At the top, there are three buttons: 'Current Week', 'Current Month', and 'All Appointments'. The 'All Appointments' button is selected, indicated by a blue border. To the right of these buttons are 'Reports' and 'Logout' links. Below this, the title 'Appointments' is displayed above a table. The table has columns: ID, Title, Description, Type, Location, Start Date/Time, End Date/Time, Customer ID, User ID, and a blank column. One row is visible, showing ID 1, Title 'Comput...', Description 'New York', Type 'Malfunc...', Location 'Desktop', Start Date/Time '2023-08-30 09...', End Date/Time '2023-08-31 09...', Customer ID 1, User ID 1, and a blank cell. At the bottom of this section are 'Add', 'Update', and 'Delete' buttons. Below this is a 'Customers' section with a 'Search Customer' input field. A table follows, with columns: ID, Name, Address, Phone Number, State, and Postal Code. One row is visible, showing ID 1, Name 'John Appleseed', Address '123 Cloverlane', Phone Number '1234567890', State 'Alabama', and Postal Code '12345'. At the bottom of this section are 'Add', 'Update', and 'Delete' buttons. A scroll bar is visible between the two main sections.

14. Here is the reports page with 3 different report types. The top report allows you to select a specific customer to filter and view all their appointments. Use the drop down

combo box and select a customer name.

The screenshot displays the AppTrack Appointment Tracker application interface. At the top, there is a header bar with three colored circles (red, yellow, green) on the left and a search bar on the right containing the text "Customer ID John Appleseed". Below the header is a table titled "Reports" with columns: Title, Type, Description, Start Date/Time, End Date/Time, and Customer ID. A single row is visible: "Computer Repair" (Type: New York, Description: Desktop), "2023-08-30 09:00:00", "2023-08-31 09:00:00", and "John Appleseed". To the left of the main window, there is a sidebar containing two tables. The first table, titled "Division Name", has columns "Division Name" and "Total Customers". It lists states with their respective customer counts: Alabama (1), Arizona (0), Arkansas (0), California (0), Colorado (0), Connecticut (0), and Delaware (0). The second table, titled "Total Appointments", has columns "Total Appointments", "Appointment Type", and "Appointment Month". It shows one appointment entry: "1" (Appointment Type: New York, Appointment Month: 2023-08-30 09:00:00).

Division Name	Total Customers
Alabama	1
Arizona	0
Arkansas	0
California	0
Colorado	0
Connecticut	0
Delaware	0

Total Appointments	Appointment Type	Appointment Month
1	New York	2023-08-30 09:00:00