**C195 – Advanced Java Final Project**

**Project Name:** mySchedule

**Author:** Bradley Duderstadt (bradd)

**Student ID:** 000326240

**Mentor:** Elaine Moran

The default username and password for the application is:

**Username:** test

**Password:** test

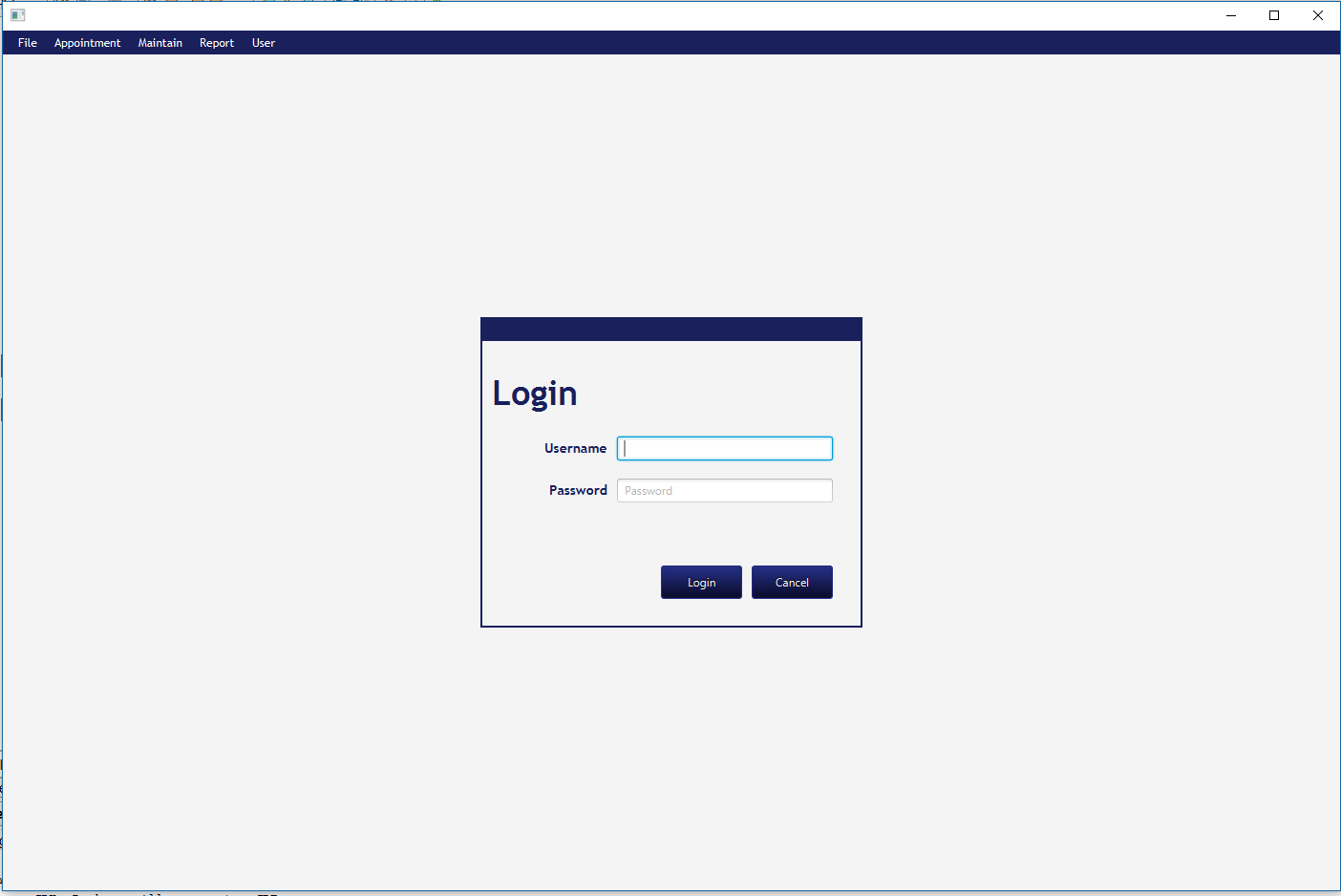
The application uses multiple services to provide various functionality. These can be found in myschedule.service. There are three (3) classes: Common.java, DB.java, and Logging.java. Their names describe the functionality that they contain.

App.java is the class that extends Application. This will be the class that needs to be modified to change the Locale. Locale is implemented in the **public void start(Stage stage)** method which can be found beginning on line 148 (App.java). I set up resource files to handle both English (EN) and German (DE) for the multi-language requirement.

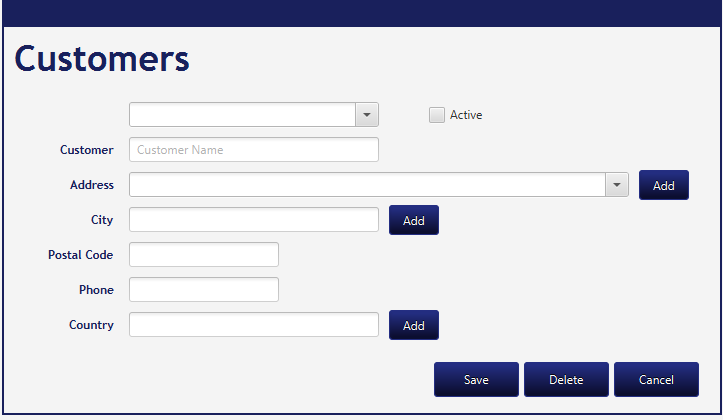
I did utilize the initial database load script to pre-populate some data in the database tables. I also utilized the auto-increment script that changed the ID columns of the tables so that when records are added they will automatically receive unique IDs. \*\*\* NOTE \*\*\* that I have to “old” schema for the database. Because of the old schema being what was setup for me, I utilized the description column of the appointment table as the “type” of the appointment.

**Requirements were met in the following ways:**

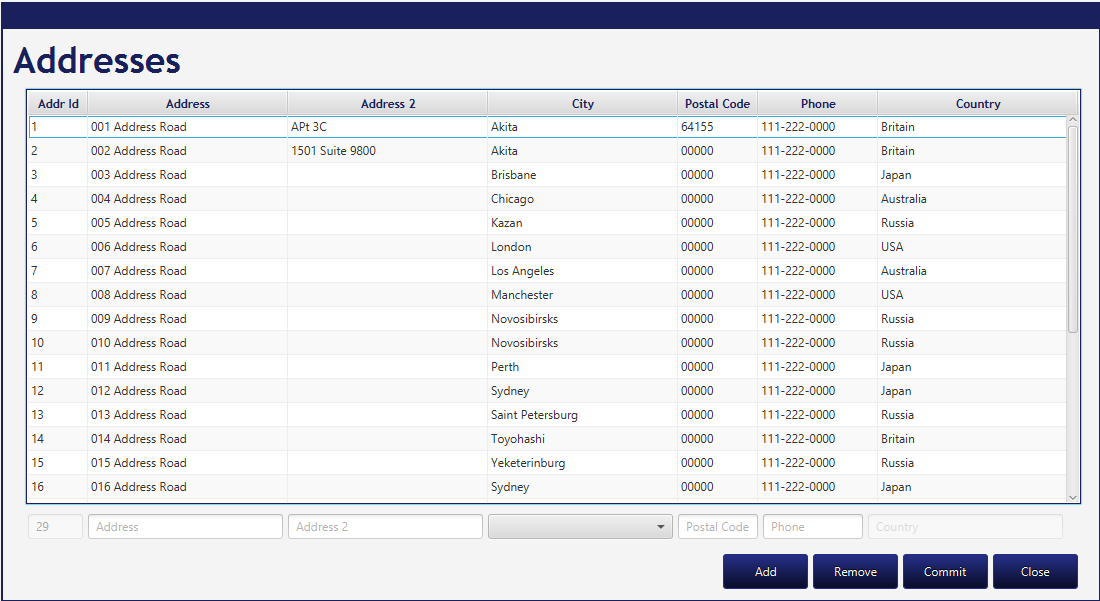
1. Log-In Form: The login form will be the first screen that is displayed. Menu options are disabled until a successful log-in has been completed. Please note that login attempts will be logged to the myschedule.log file. This satisfies item J as well.



1. Customer (records) can be maintained via the Maintain | Customer menu item. The following screen is displayed. A new customer can be modified by selecting an existing one from the drop-down. There is also an option to Add New Customer which is how a new customer is added. Drop-downs are also provided to select an address, city and country. Each of these also have buttons available that allow the user to “jump” to that specific maintenance form. The way the system is designed is to maintain Customers, Addresses, Cities and Countries independently of each other. This ensures data integrity. Below are the screenshots of each. Please note that an editable table view is used for each to allow each cell (field) of a record to be modified. After modifying a field, press <ENTER> and that will save the modification to the table view. There is a Commit button that will save / commit the modification(s) to the database. Also note, that if an attempt is made to leave the form with unsaved changes, the user will be notified and given the opportunity to save their modifications without losing them.



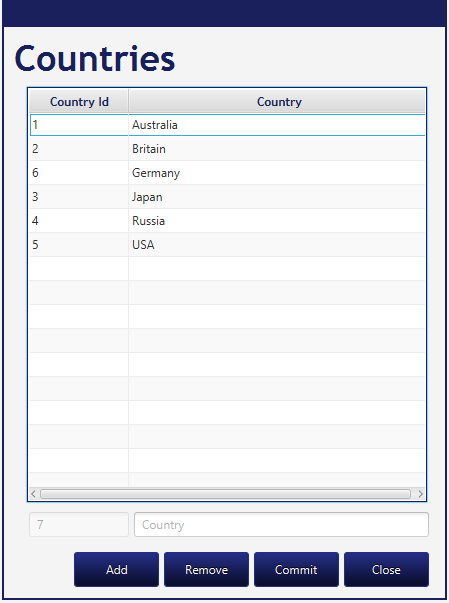
Customer Maintenance Interface



Address Maintenance Interface



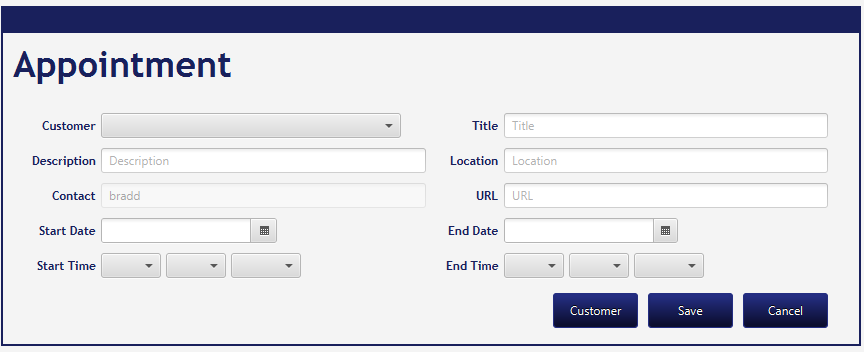
City Maintenance Interface



Country Maintenance Interface

Please note that each form has fields at the bottom that are utilized to add a new record of whichever type. Once the information is entered, you will need to click the Add button. Once again, after additions, deletions or modifications have been made, the application will know about them and you will be prompted to Commit your changes if you haven’t already done so.

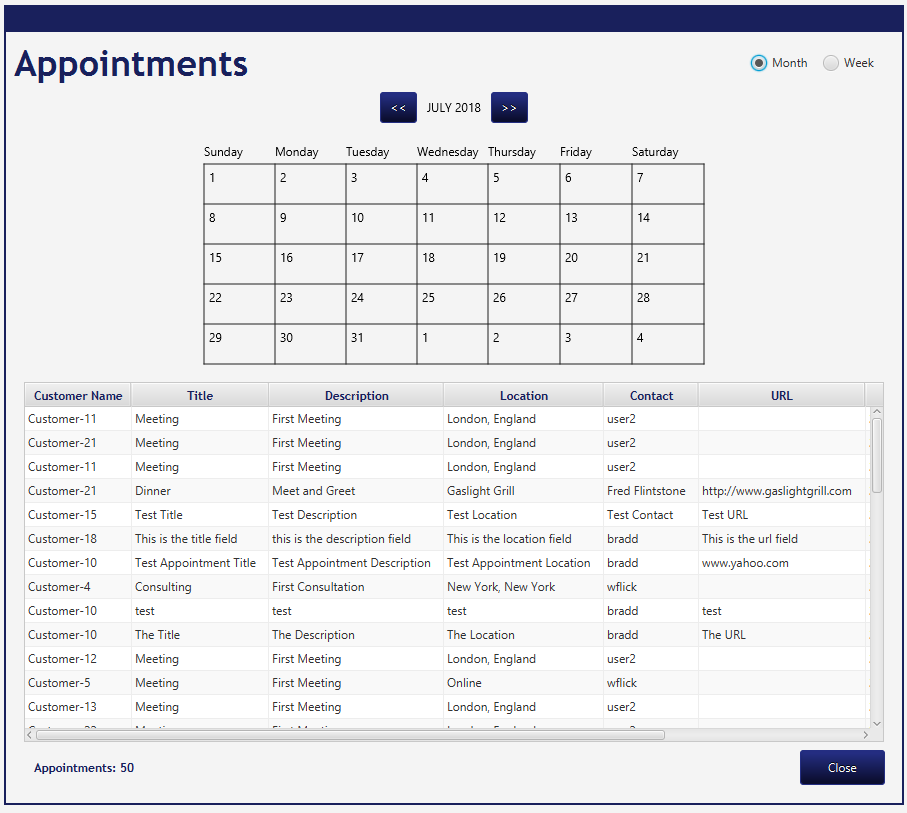
1. Appointments – Appointments work similarly to the Customer’s maintenance interface. You can find it located under the Appointment menu item and then Add. The View is also located under the Appointment menu item.



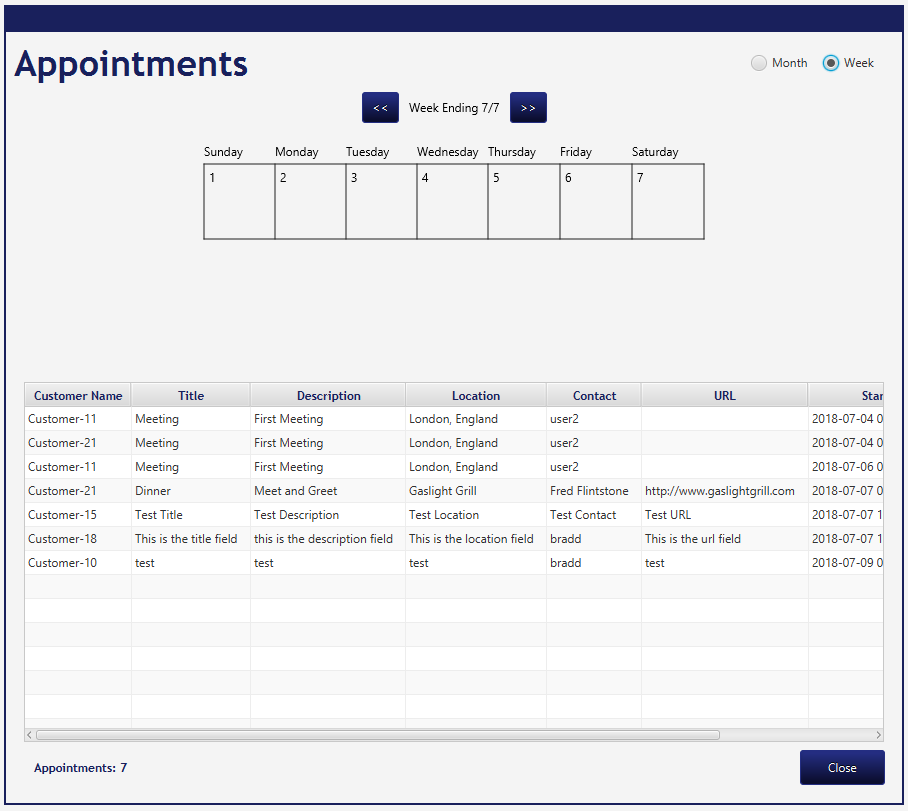
Appointment Maintenance Interface

The appointment interface has a button that allows you to link directly to the Customer maintenance page (once you’ve selected a customer). Note that as you are creating a new appointment, your log-in username will be pre-populated as the Contact and cannot be changed. This is how I was told to do it from the course mentors. Validation exists to ensure that proper dates are utilized and that business times are followed. I used 8:00am – 5:00pm as valid date / times.

1. Calendar Views – Calendar views are implemented in a similar fashion and can be found under the Appointment menu Item | View. A table view is again utilized to display current appointments. There are radio buttons at the top that allow the view to be switched from “month” view to “week” view. To edit an appointment, simply click on one of the rows in the table view and that specific record will be opened up in the appointment maintenance interface.



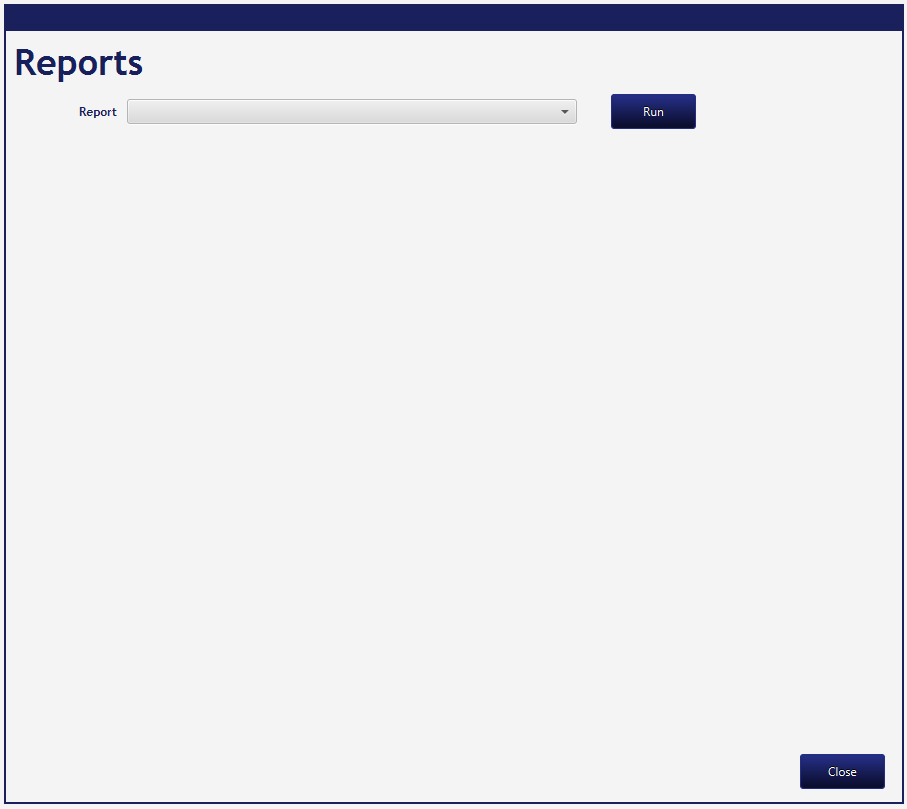
Appointment View Interface (Monthly View)



Appointment View Interface (Weekly View)

Table columns are resizable as needed. Also note the arrow buttons on either side of the table view title. This will allow you to scroll forward or backwards through the available appointments.

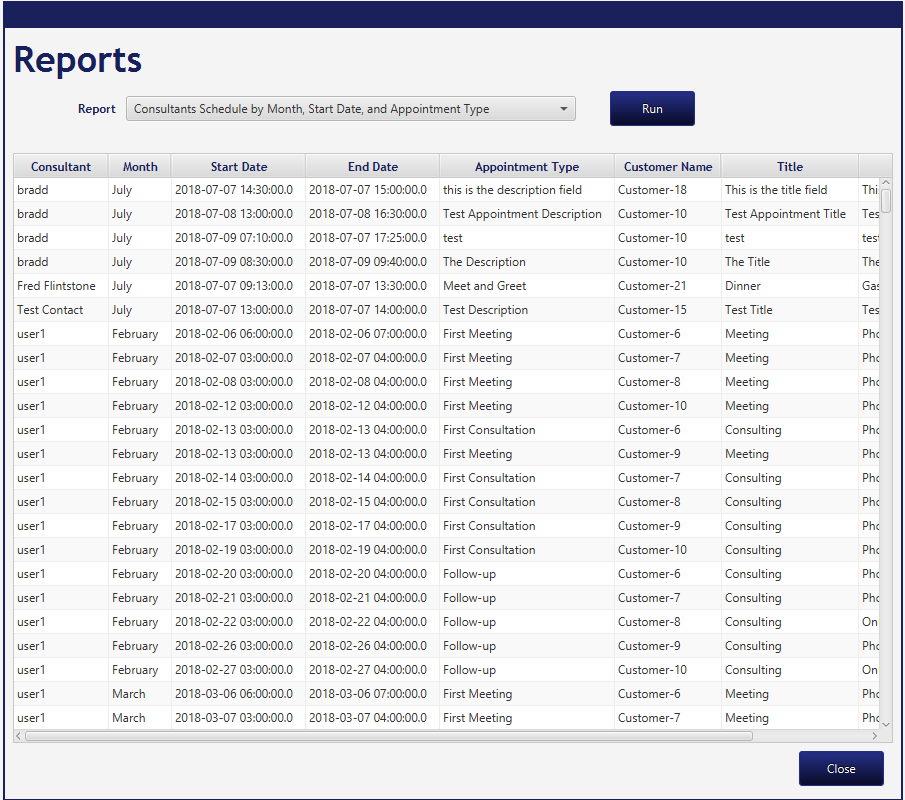
1. Time Zones are implement by calculating the TimeZone offset when the application is first loaded. The database stores all dates as UTC so I utilized MySQL’s CONVERT\_TZ to get the dates stored in the proper time zone. When data is retrieved, the opposite conversion takes place to show the dates / times in the current local time zone. Please be aware that if you change your time zone on your system to test out the Time Zones requirement, you MUST close the application and reload it to obtain the correct time zone for the “session” of the application.
2. Exception control – Try/Catch blocks and Throws/Throwable were utilized throughout the entire application. You should have no issue finding many, many examples of such.
3. Pop-ups. Again, pop-ups are utilized throughout the program to display successful database update messages as well as to indicate issues with updating the database.
4. Reminders and alerts. See above. The display of upcoming appointments is handle through the same mechanism. If there is an alert within 15 minutes of a successful login, that appointment (reminder) will be displayed.
5. The reports requirement is taken care of under the report generator that can be found under the Report | Run menu item.



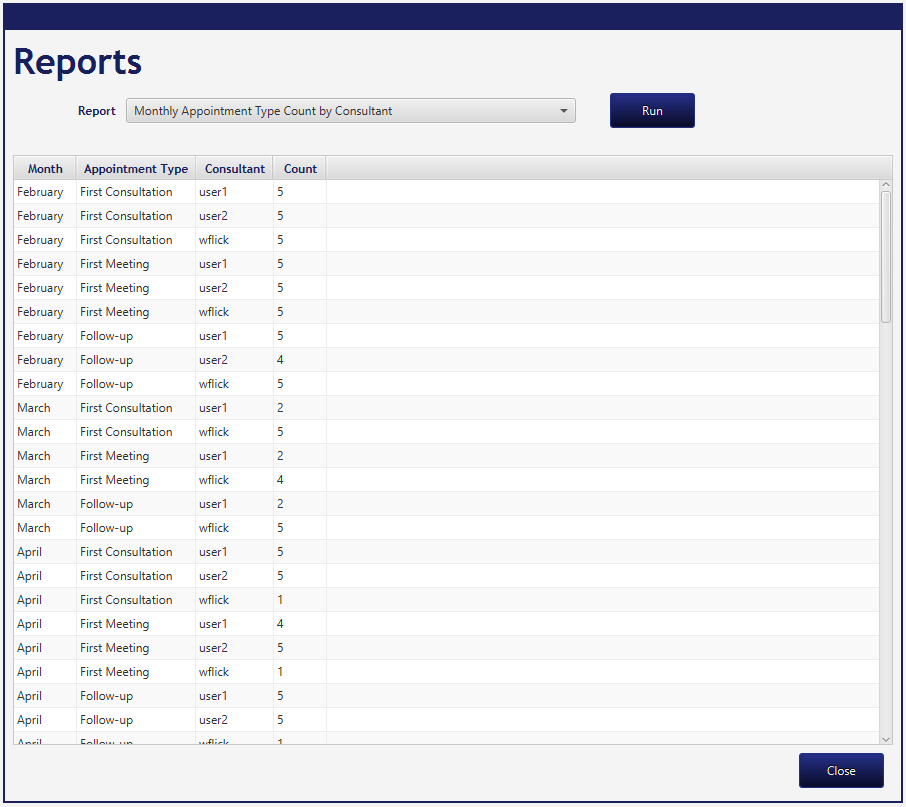
The appropriate report needs to be selected under the Report drop-down and then once the desired report has been selected, click on the Run button and the report will be generated.

1. As discussed earlier, the Activity Log has been satisfied via the myschedule.log file that is generated and located in the mySchedule directory root.

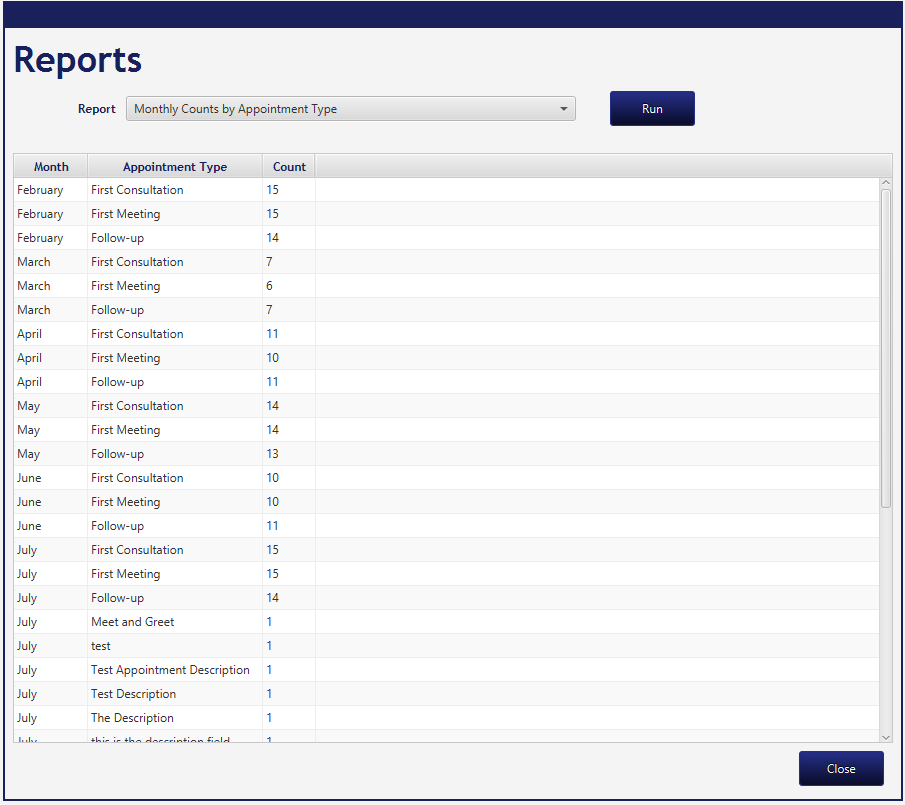
As for the quality code, it should well satisfy any requirements set forth by evaluators.



Consultants Schedule by Month, Start Date and Appointment Type report



Monthly Appointment Type Count by Consultant report



Monthly Counts by Appointment Type report