C868 – Software Development Capstone

Task 2 Section C

# By Gerald Seth Scheller

**Application Design**

Class design

All the classes used to create the Appointment Scheduler application are shown below in figure 1. I have chosen to show both the model classes and the controller classes. Figure 1 shows the flow of the application and the dependencies each class has on another.

The *LoginScreen* class is used to initialize the program and validate credentials. It relies on MySQLWrapper model to validate the user’s log in credentials and then sends them to the Appointment screen. Each attempt, whether successful or not is logged into a file. That class method can be on the diagram and is called *printLoginAttempt*. From there you will be sent to the *AppointmentController*. The *AppointmentController* allows you to view, edit and delete appointments. the *AppointmentController* also provides access to the different scenes, where you can add customers, edit customers or pull reports. Those scenes are controlled by *EditCustomerController*, *AddCustomerController*, *ContactAppointmentCountController*, *AppointmentTypeMonthController* and *ContactScheduleController*.

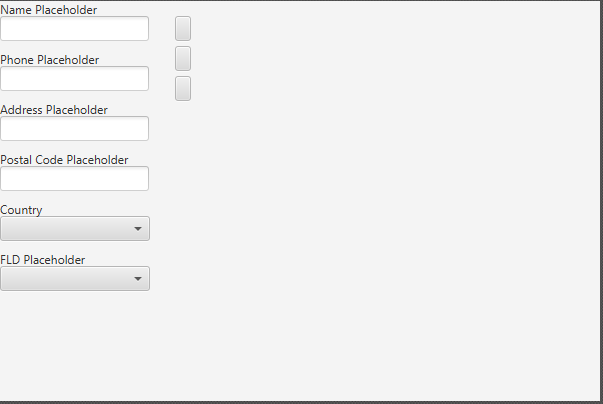
The Model classes for the application can be seen in the diagram and are labeled appointment, appointment report, contact report, customer and MySQLWrapper. An appointment can only be created by one user and can only have one contact. A customer can have many appointments, but an appointment can only have one customer.



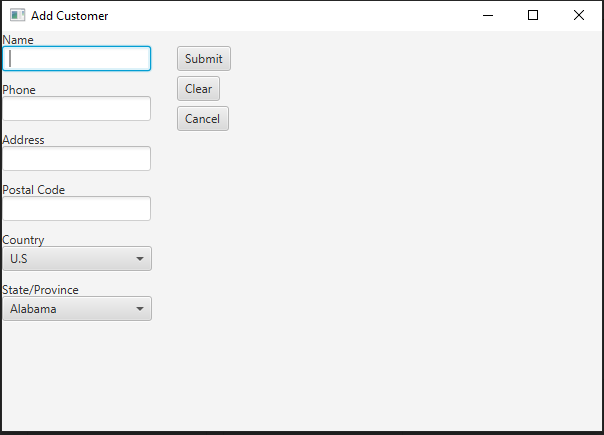
*Figure 1: UML diagram for class/design*

UI Design

Below are the low and high-fidelity versions of the add customer screen. On the left side of the screen, you have the customer’s name, phone number, address, postal code, country and state/providence. To the right of that you have an interactive button to submit the form, clear the form or cancel the for and return to the previous page.



*Low fidelity wireframe of add customer screen.*



*High-fidelity version of the add customer screen.*

**Test Plan**

Introduction

Purpose

Booking appointments should be easy and should prevent double booking. This was a main issue with the current method of scheduling appointments. The new scheduling application allows for a quick method to book appointments but will provide alerts if you try to double book appointments.

Overview

The core of the scheduling application is to ensure that appointments get created and saved in a universal database that all consultants can use. Each appointment that is schedule needs to ensure that they are being properly vetted. There needs to be validations that the appointments aren't being double booked.

This unit test plan will run each of those scenarios through the scheduling application. There will be predetermined data for entry into the application. If any of these items are allowed through or aren't allowed through will determine whether the test is a success or failure.

Test plan

Items

To test that each appointment made for a consultant doesn’t already have a conflicting appointment made for that same time, we need two model instances: one consultant and two appointments. Both appointments will use the same consultant. The first appointment will be created using a specific start time and date. The second will then be scheduled for the same time and date. The second appointment should receive an error.

A second test can be created to ensure that a consultant can have more than one appointment for the same day, just not at the same times.

Function/Features

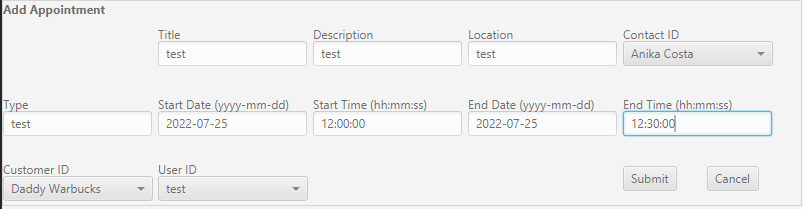
To validate that the appointment being created does not overlap, I created the *appointmentTimePassesAllRules* class method. This method compares if the current appointment (String) is a match to any of the pre-existing appointments that are currently in the *cachedAppointments* by using a for loop. It will then use if statements to create objects, and if there is a match the method will return false and an error is created. If there are no matches the method will return true and add the appointment to the database. This method is called twice from the appointments class. The first time is in the *submitNewAppointment,* when a new appointment is submitted to the database. The second time this is called is from the *submitUpdateToAppointment*, when there is an attempt to update an existing appointment.

Deliverables/Outcomes

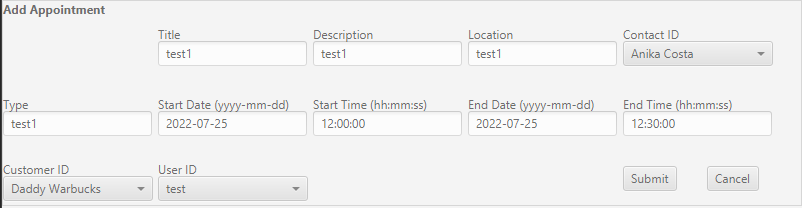
When the test is run and fails to pass the validation the appointment view will create a specific error. The exact error message should be displayed as “Appointment overlaps with another appointment. Please choose a different date or time.”

Tasks

1. Start the application, enter “test” as both the username and password. Then click the login button.
2. Locate the “Add Appointment” section of the appointments screen.
3. Using the preloaded ContactID for “Anika Costa”, create the following appointment.



4. Next, create a second appointment with the following information.



5. When you “submit” the second appointment the bottom of the appointments screen should display the following error message.



6. Determine if the test passed or failed base on the output.

Needs

I used the following software for testing:

* IntelliJ IDEA 2022.1.2 (Community Edition)
* JDK version 11.0.13
* JavaFX 18.0.1
* MySQL workbench, mysql-0connector-java-8.0.23

Pass/fail Criteria

The criteria for a successful validation of an overlapping appointment instance are that the method call to *appointmentTimePassesAllRules* returns false. When the method returns false the appropriate error message displays at the bottom of the screen. A failed test would pass true and therefore not display an error.

Specifications

The following screenshots are the test code used to validate the duplication of appointments. It is a except from the *appointmentcontroller* class. This method is checked twice. Once when creating new appointments and once when updating existing appointments.



Procedures

The procedures I followed for performing the test were:

1. Opened IntelliJ IDE from my local machine.
2. Opened the project named Appointment scheduler and ran the project.
3. Followed steps 1-6 listed above in the “tasks” section of this document.
4. Captured screenshots for illustration purposes.
5. Verified the test results by comparing them to the “Pass/Fail Criteria”.

Results

The following screenshot is the output I received when performing the tests. The message received at the bottom of the screen is the expected message and so the test can be confirmed as pass.



**Source Code**

The attached Zip file labeled “c868 Capstone” contains the application source code.

The executable file is also attached and shown below.



**User Guide**

Introduction

This user guide will explain how to navigate and use the appointment scheduling application. Explanations and screenshots will be included for all facets of the application.

Login

The first screen you will see when you run the application is the login screen, shown below.

Graphical user interface, application

Description automatically generated

1. Enter your credentials into the “Username” and “Password” fields.
   1. Testing Username = “test”, Password = “test”
2. Click Submit

Main Menu

Once you have logged in you will be presented with the main appointment menu. The main menu serves as your base location for viewing, editing, adding, and deleting all appointments. Below is a screenshot illustrating each section of the main menu.

View reports:

Count by type/month

By Contact

Count by contact

Add/edit Customers

A screenshot of a computer

Description automatically generated with medium confidence

View/select appointment

Edit/delete appointment

Add appointment

View/Select appointment

Table

Description automatically generated

5

4

3

2

1

The following numbers correspond to the numbers on the image and describe each function on the screen.

1. Filters the appointment view by month.
2. Filters the appointment view by week.
3. Filters the appointment view by all appointments.
4. Filters the appointment view by search criteria entered. The search field is dynamic and will filter all appointments based on user input.
5. Allows you to select a specific appointment. By selecting an appointment, you can add, update, or delete that appointment.

Edit/Delete Appointment

Graphical user interface, application, Word

Description automatically generated

3

2

1

Once you have selected an appointment, the information will be populated into the edit/delete appointment section of the form. You can edit any field except the appointment id.

The following numbers correspond to the numbers on the image and describe each function on the screen.

1. Delete button is used to delete the selected appointment.
2. Update button is used to update the edited field.
3. Cancel button is used to cancel the selection and clear the fields, putting that appointment back into the database without any changes made.

Add Appointment

Graphical user interface, application, Word

Description automatically generated

12

11

10

9

8

7

6

5

4

3

2

1

The add appointment section is where you will add new appointments. Fill out all the information text fields for the appointment. The following numbers correspond to the numbers on the image and describe each function on the screen.

1. The appointment title.
2. The appointment description.
3. The appointment location.
4. The consultant id.
5. The type of appointment.
6. Start date of the appointment.
7. Start time of the appointment.
8. End date of the appointment.
9. End time of the appointment.
10. Customer id.
11. Submit the appointment and save to the database.
12. Cancel the appointment and clear the fields.

Add Customer

Graphical user interface

Description automatically generated

3

2

1

This is the screen you will use to add a customer to the database. Please fill out all the fields for the customer. The following numbers correspond to the numbers on the image and describe each function on the screen.

1. Submit the customer and information to the database.
2. Clear the form and start over with new information.
3. Cancel the customer creation and return to the previous screen.

Edit Customer

3

Graphical user interface, application

Description automatically generated

2

1

This is the screen you will use to edit or delete customers already existing in the database. First you will select the customer you want to edit/delete. To **edit**, you only need to change the information you want and select update. To **delete** a customer, just select the customer and delete. The following numbers correspond to the numbers on the image and describe each function on the screen.

1. Update is used to save the changed information to the database.
2. Delete is used to delete the selected customer.
3. Is used to exit the page and return to the appointment screen.

Reports

**Count of meeting types by month –** This report is used to show the number of meetings by month and the type of meetings.

Table

Description automatically generated

**Contact Schedule –** This report is used to show a specific customer’s appointments. To view a specific customer, select them from the drop-down list.

Graphical user interface, text, application, email

Description automatically generated

**Contact appointment count –** This report is used to view the number of appointments each consultant has per month.

Table

Description automatically generated