CS410J Project 5: A Rich Internet Application for an Appointment Book (15 points[1])

In this project you will use Google Web Toolkit to implement a Rich Internet Application for your Appointment Book application.

Goal: Gain experience writing a complex web-based user interface with Google Web Toolkit

For this project you will write a user interface using GWT that has all of the functionality of the previous assignments: creating an appointment book, entering appointments, pretty printing an appointment book and its appointments, searching for appointments, etc..

The entry point of your user interface should be the `edu.pdx.cs410J.<login-id>.client.AppointmentBookGWT` class and should be accessible via the following URL when run on the local machine in GWT development mode:

$$\texttt{http://localhost:8888/AppointmentBookGwt.html}$$

The user interface[2] should provide a "Help" menu with a "README" menu item that displays the information for the README. Note that the application should use GWT's mechanisms for communicating with the server, you do not need to use the REST-based client you developed in the previous project.

Error handling: If something goes wrong (for example, the user provides malformed input[3], your user interface should pop up a dialog box that describes the error.

This project gives you the opportunity to explore the tools that GWT provides. I encourage you to use your imagination[4] and creativity when designing your user interface.

To aid you in setting up your Maven project, I have put together an archetype for this project. To create a GWT project execute this Maven command (remember that the backslash in the line continuation character):

```
$ mvn archetype:generate \
  -DarchetypeCatalog=https://dl.bintray.com/davidwhitlock/maven/ \
  -DarchetypeGroupId=edu.pdx.cs410J \
  -DarchetypeArtifactId=apptbook-gwt-archetype \
  -DgroupId=edu.pdx.cs410J.<login-id> \
  -DartifactId=apptbook-gwt \
  -Dversion=1.0-SNAPSHOT
```

---

[1]14 for code, 1 for POA

[2]I highly recommend using a tool like Balsamiq (`http://www.balsamiq.com` to mock-up your UI before implementing it. If you do so, you should include attach it to your POA email.)

[3]Dates and times should be specified in the same format as the previous project.

[4]Since that the `Submit` program only allows you to submit `.java` files, I'd like you to just zip up the contents of your Maven project after performing a `mvn clean` and email it to `sjavata@cs.pdx.edu`. Please do not submit compiled artifacts such as class files and jar files.

You can build your project with `mvn package` and can launch GWT's development mode with `mvn gwt:run`. You can also run your application in production mode with `mvn jetty:run-war` and view your application at:

```
http://localhost:8080/apptbook
```