Tim Nesbitt

Tnesbi3@wgu.edu

C195 Project Requirements notes:

Users: Tim, password Lisa, password

A. Create a log-in form that can determine the user’s location and translate log-in and error control messages (e.g., “The username and password did not match.”) into two languages.

**TN: This function can be tested by uncommenting “Locale.setDefault(new Locale("es", "MX"));” in the “setLanguage()” method in the LoginScreenController class. This will force the application to reference Spanish resource file info.**

B. Provide the ability to enter and maintain customer records in the database, including name, address, and phone number.

**TN: Via the “Add New Customer” button on the Customer screen.**

C. Write lambda expression(s) to schedule and maintain appointments, capturing the type of appointment and a link to the specific customer record in the database.

**TN: Lambda expressions are used to get and set FMXL screen values throughout the program.**

D. Provide the ability to view the calendar by month and by week.

**TN: On the Appointment Management screen the “WITHIN NEXT WEEK” and “WITHIN NEXT MONTH” will allow the user to view only the appointments within that timeframe – clicking the adjacent arrows will advance to the next or previous weeks/months worth or appointments.**

E. Provide the ability to automatically adjust appointment times based on user time zones and daylight saving time.

**TN: This can be tested by uncommenting default time zone setting statements on the “initialize” method of the LoginScreenController class. The times on the Appointment screen will adjust (base time in DB is EST).**

F. Write exception controls to prevent each of the following. You may use the same mechanism of exception control more than once, but you must incorporate at least two different mechanisms of exception control.

• scheduling an appointment outside business hours

**TN: Using drop-downs it is not possible to schedule outside of business hours.**

• scheduling overlapping appointments

**TN: If you attempt to schedule a contact for a time they are already scheduled for you will produce an error.**

• entering nonexistent or invalid customer data

**TN: Entering a mis-formatter phone number will produce an error. Failing to enter Name, Address or Phone Number will produce an error.**

• entering an incorrect username and password

**TN: This will produce an error on the login screen.**

G. Use lambda expressions to create standard pop-up and alert messages.

**TN: Used a lambda in the “cancelAppointmentScreen” alert in the AppointmentScreenController class.**

H. Write code to provide reminders and alerts 15 minutes in advance of an appointment, based on the user’s log-in.

**TN: This will appear based on the user login checked against the Contact schedule. If the Contact and the user login match, and there is an appointment scheduled for that Contact within the next 15 minutes.**

I. Provide the ability to generate each of the following reports:

• number of appointment types by month

• the schedule for each consultant

• one additional report of your choice

**TN: All reports can be generated on the Customer screen using the buttons in the upper right hand corner.**

J. Provide the ability to track user activity by recording timestamps for user log-ins in a .txt file. Each new record should be appended to the log file, if the file already exists.

**TN: Log file written to project root.**