

Jacob Clark

08/02/2020

C195 Project Requirements Checkoff

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.

The Log In Controller handles this requirement by getting the systems locale via calling the `Locale.getDefault()` method and then calling the `getDisplayLanguage` on the locale, and adjusts the control messages based on the display language.

B. Provide the ability to add, update, and delete customer records in the database, including name, address, and phone number.

The Customer Management screen/controller handle the aforementioned functions by handling the `onClick` events of the buttons on the customer management screen.

The add customer button pulls up the `AddCustomer` screen, which validates the user input entered in to the add customer form, and then inserts the customer into the database by calling the `insertCustomer` method in the `SQLQuery` class.

The modify customer button gets the selected item and pre-fills the modify customer form, and then calls the `modifyCustomer` method in the `SQLQuery` class which modifies the customer entity with the `customerId` retrieved from the selected item.

The delete customer button gets the selected item and calls the `deleteCustomer` method in the `SQLQuery` class and deletes the entity with the given `customerId`.

C. Provide the ability to add, update, and delete appointments, capturing the type of appointment and a link to the specific customer record in the database.

The Appointment Management screen/controller handle the aforementioned functions by handling the `onClick` events of the buttons on the Main screen.

The add appointment button pulls up the `AddAppointment` screen, which validates the user input entered in to the add appointment form, and then inserts the appointment into the database by calling the `insertAppointment` method in the `SQLQuery` class.

The modify appointment button gets the selected item and pre-fills the modify appointment form, and then calls the modifyAppointment method in the SQLQuery class which modifies the appointment entity with the appointmentId retrieved from the selected item. The delete appointment button gets the selected item and calls the deleteAppointment method in the SQLQuery class and deletes the entity with the given appointmentId.

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

Accomplished by a radio button that allows the user to view appointments by the month or week which determines the end localDateTime that is entered as a parameter in the retrieveAppointments Query (LocalDateTime.now() through LocalDateTime.now().plusWeeks(1) for weekly, LocalDateTime.now() through LocalDateTime.now().plusMonths(1) for monthly).

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

Time zone adjustments are made on insertion and retrievals from the database by using the convertLocalDateTimeToUTCDateTime() SQLQuery method on insertion and convertUTCtoLocalDateTime() on retrieval to ensure times are adjusted to UTC when inserting and local when viewing in program.

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**
- **scheduling overlapping appointments**
- **entering nonexistent or invalid customer data**
- **entering an incorrect username and password**

Exception control to prevent the above is handled first by the add appointment/customer controllers to ensure the fields are not empty and match the given Regex patterns, and the SQLQuery class contains a method that can be called to ensure the consultant does not have any existing appointments that overlap with the new appointment to be made.

G. Write two or more lambda expressions to make your program more efficient, justifying the use of each lambda expression with an in-line comment.

Lambda expressions located in main controller and login controller denoted by a comment.

H. Write code to provide an alert if there is an appointment within 15 minutes of the user's log-in.

The retrieveAppointmentsInFifteen() method in the SQLQuery class can be called to query appointments with start times in the next fifteen minutes, and the logIn controller calls that method and creates an alert if there are appointments in 15.

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**

All 3 handled in the ReportsController.

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.

Upon successful login, a logs.txt file is created if one does not exist, and the username of the user that was logged in and a timestamp of when they log in is created via the logInController.

Thank you for reviewing my project!