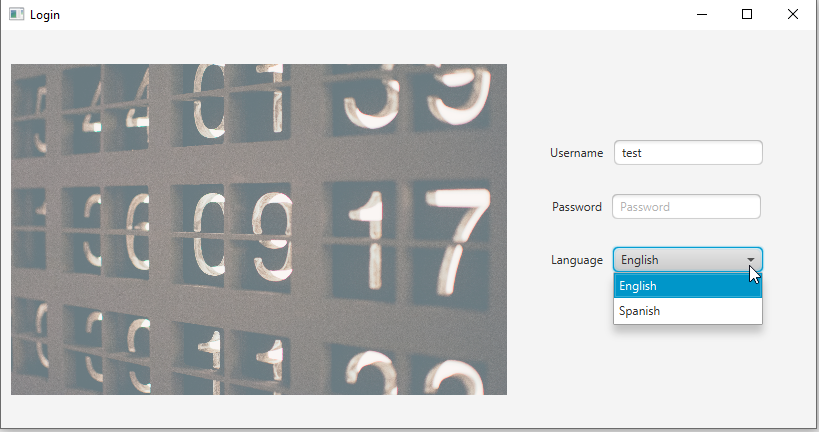
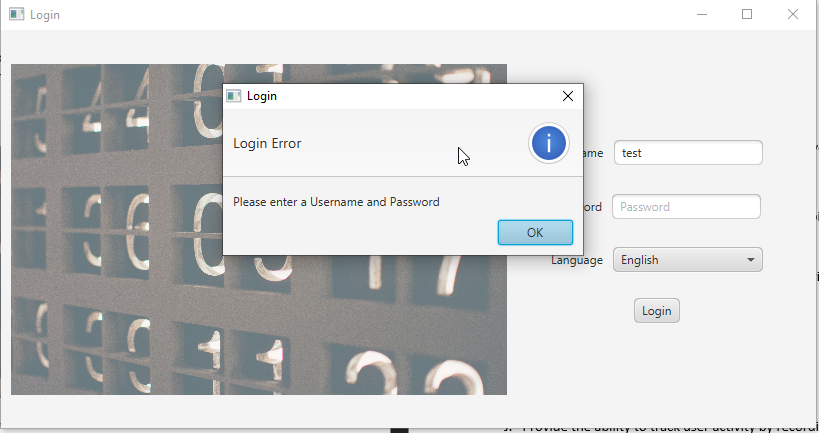
A.   Create a log-in form that can determine the user’s location and translate log-in and error control messages (e.g., “The username and password did not match.”) into **two** languages.



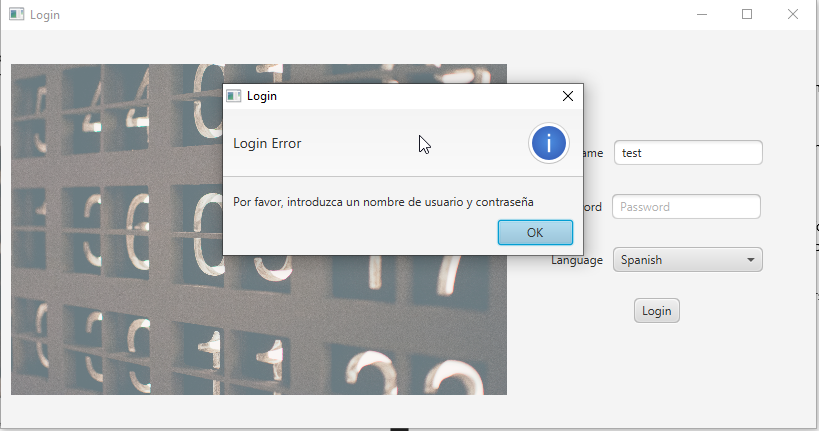


v1 updates:

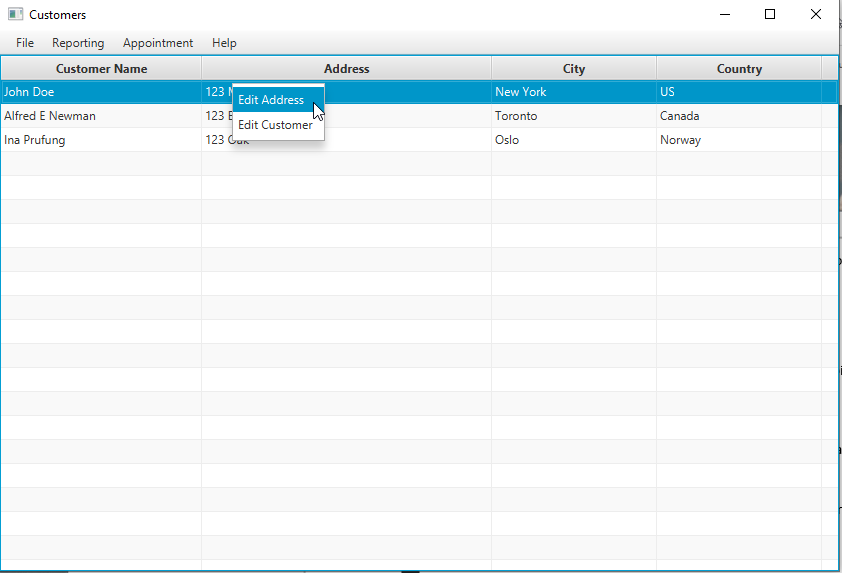
* Screen checks current locale language and sets the language drop down box to it if it is Spanish. Default language is English if not Spanish.

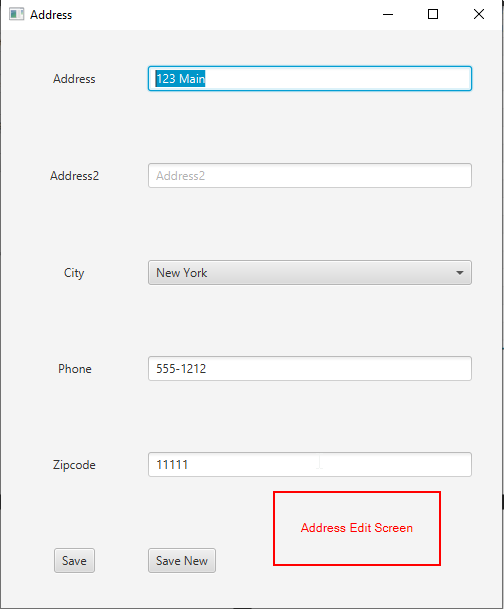
v2 updates:

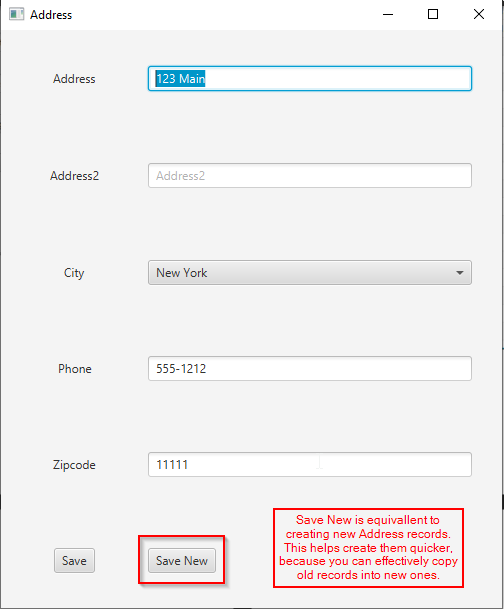
* Fixed issue with code, basically locale language was returned as a non-english word, which caused default state of English being accepted as the only case.

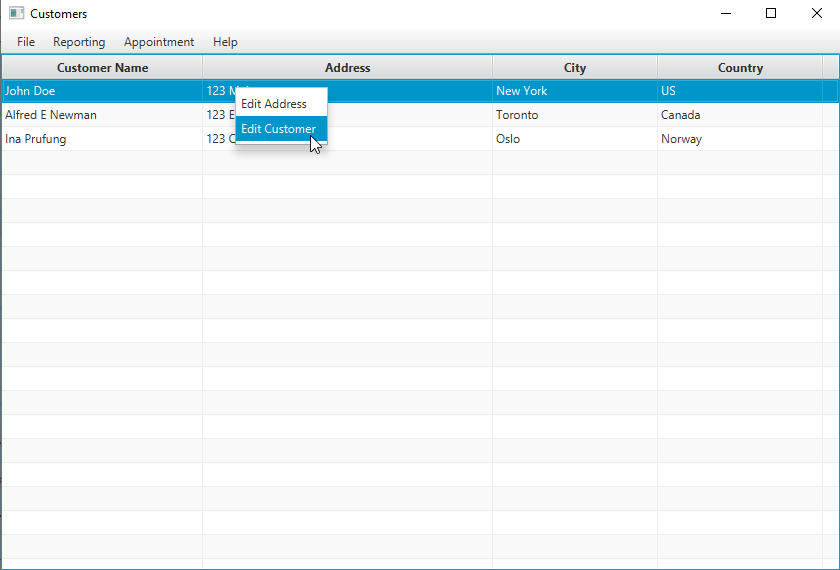


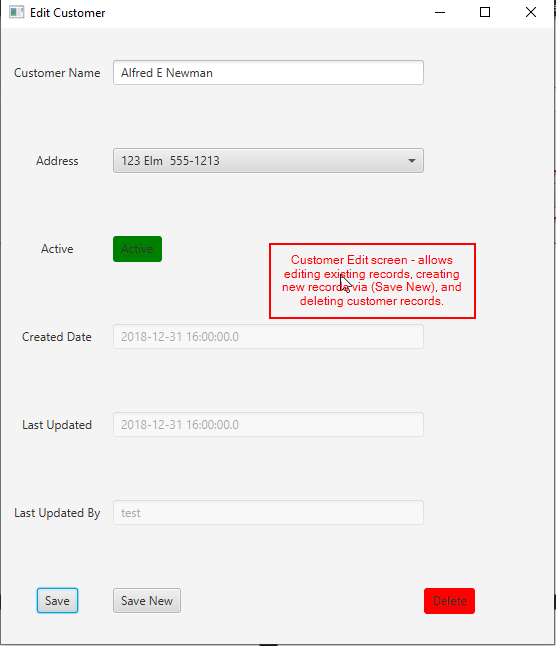
B.   Provide the ability to add, update, and delete customer records in the database, including name, address, and phone number.



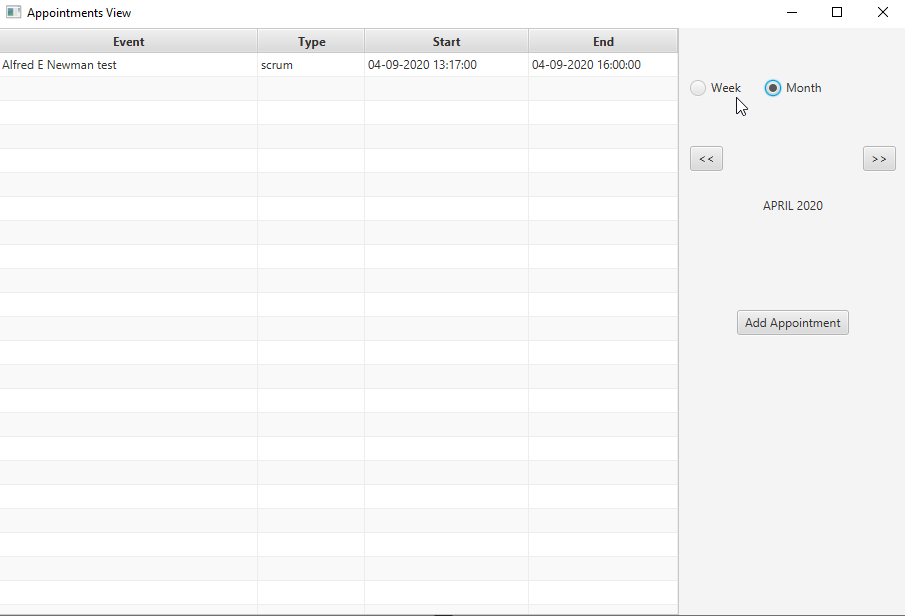


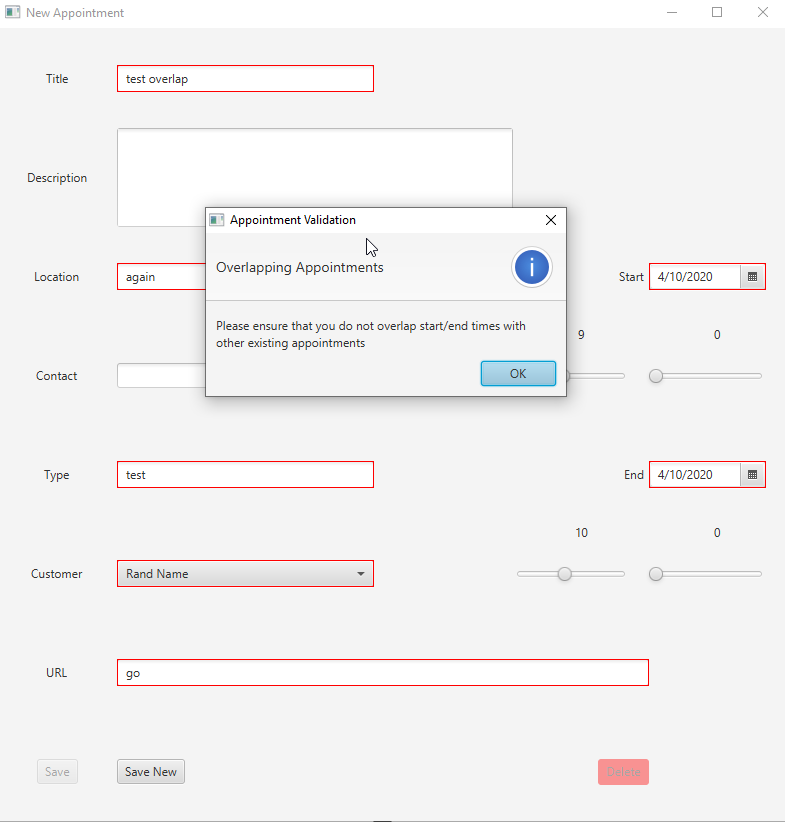




C.   Provide the ability to add, update, and delete appointments, capturing the type of appointment and a link to the specific customer record in the database. 

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





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

Done by using the LocalDateTime and ZoneDateTime API.

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 **🡪set at 9 am to 530 pm locally on appointment entry**

•   scheduling overlapping appointments

**V2 update 🡪 added code to verify if new appointments overlap as well as old appointments that are modified.**

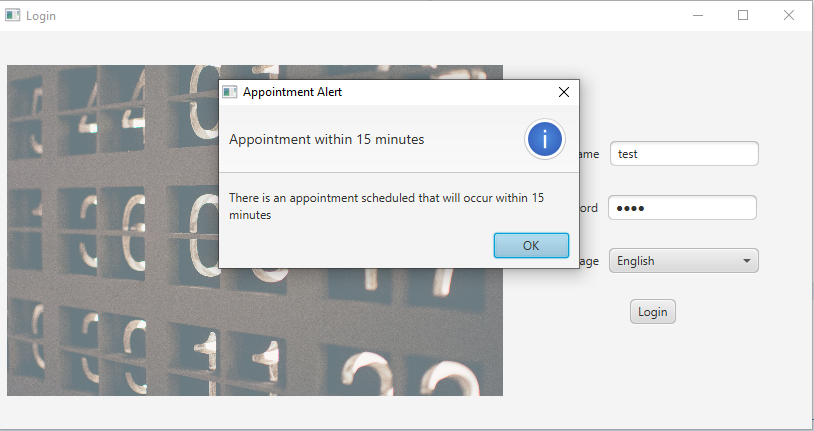
•   entering nonexistent or invalid customer data🡪 **taken care of by using combo boxes in the appointment form, address form and other related forms.**

**v1 update 🡪 added customer form validation, name + address is required. Added address form validation, address + zipcode + phone number required.**

•   entering an incorrect username and password 🡪 **done with language support from resource bundle api for incorrect login/pass or missing login/pass.**

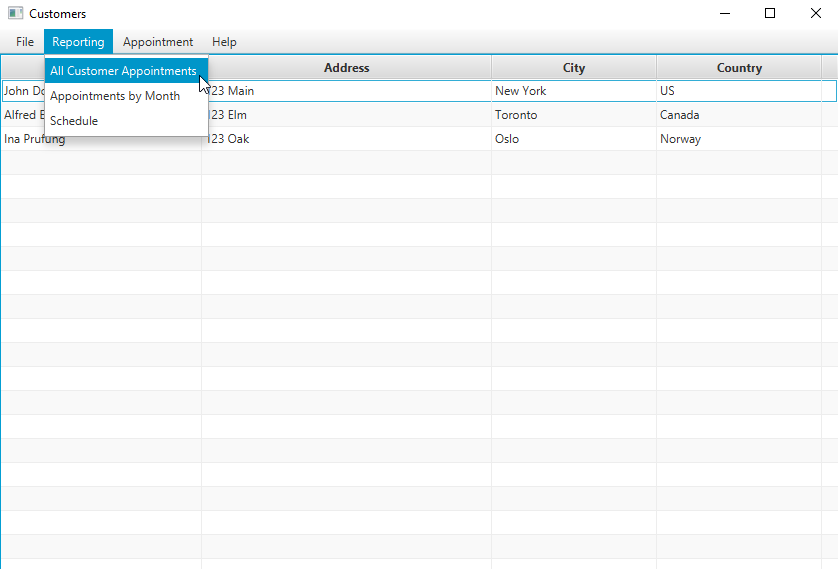
G.  Write two or more lambda expressions to make your program more efficient, justifying the use of each lambda expression with an in-line comment.

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



**NOTE: This alert happens because the current test data has appointments scheduled to start last year. In the submission these appointments will be removed, please test within BUSINESS HOURS(9-530).**

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

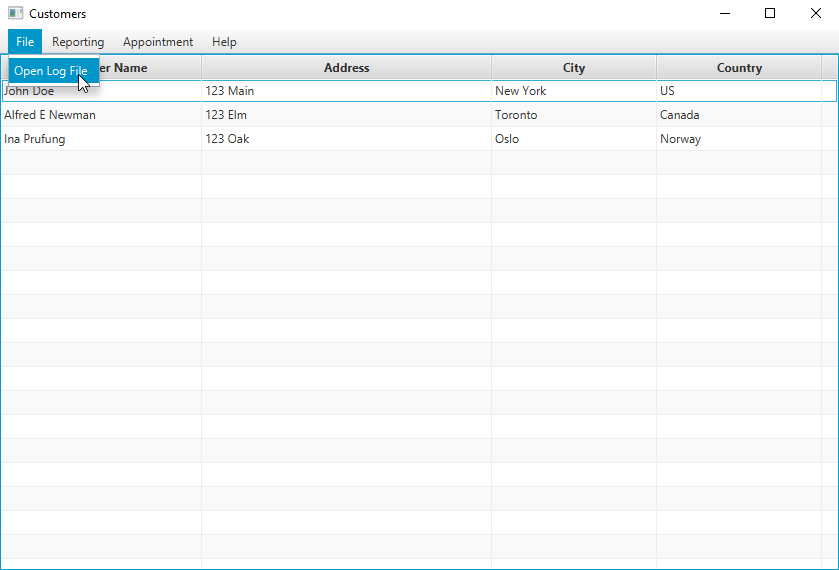


•   number of appointment types by month 🡪 **Appointments by Month**

•   the schedule for each consultant🡪 **Schedule**

•   one additional report of your choice 🡪 **All Customer Appointments**

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.



**Currently only login data including successful/unsuccessful login is showing.**

K. Demonstrate professional communication in the content and presentation of your submission.