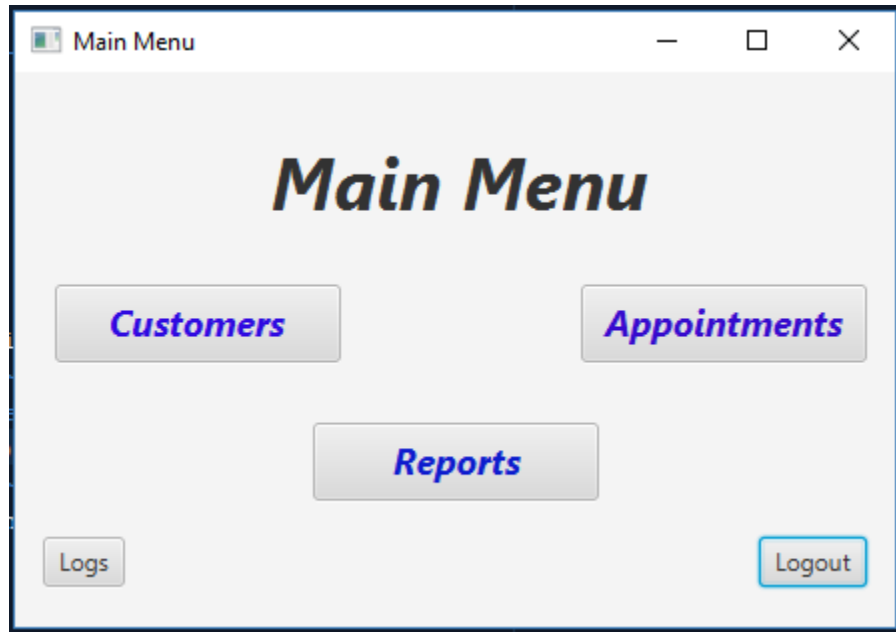


# Software II- Advanced Java Concepts – C195

Eli Armstrong

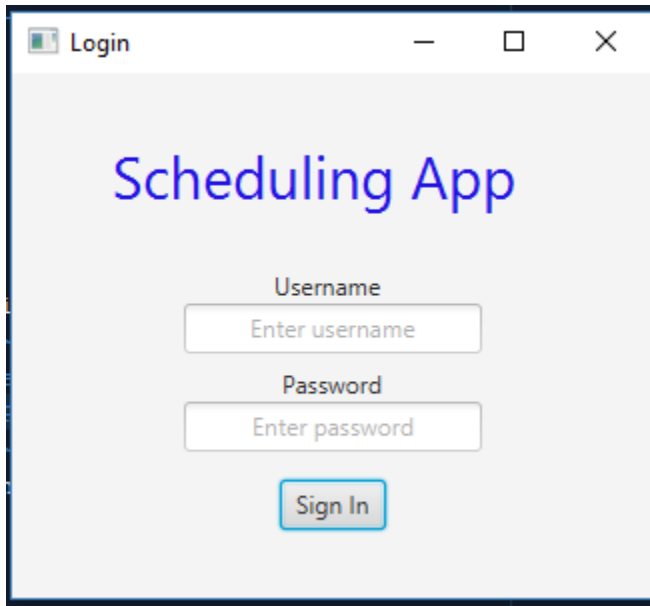
Student ID: 00930363



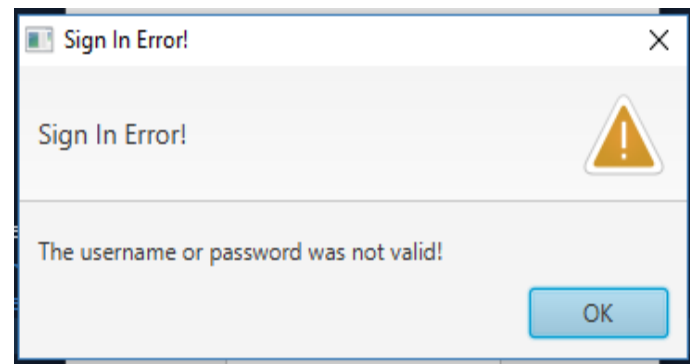
- A. Create a log-in form that can determine the user's location and translate log-in and error control messages (e.g., "The username and password did not match.") into **two** languages.

Valid usernames/Passwords: test/test, luke/notmyfather, ed/password, and candace/123456

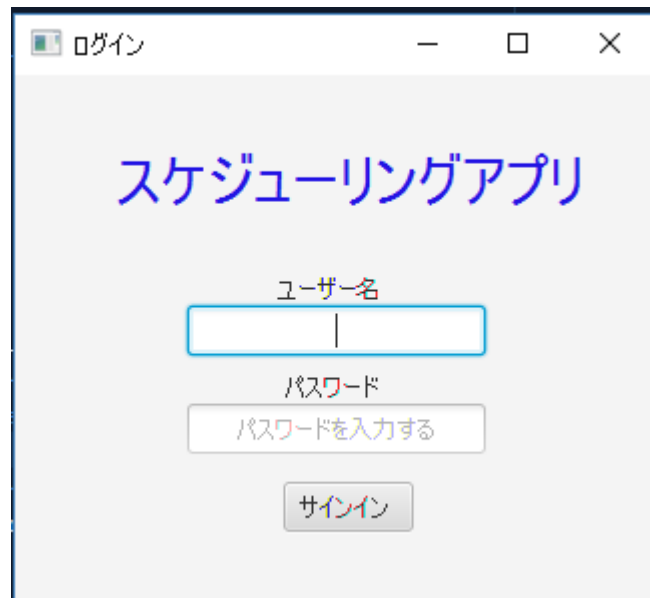
If you **uncomment line 44** in WGUSW2.java file the Locale default will be set to Japan.



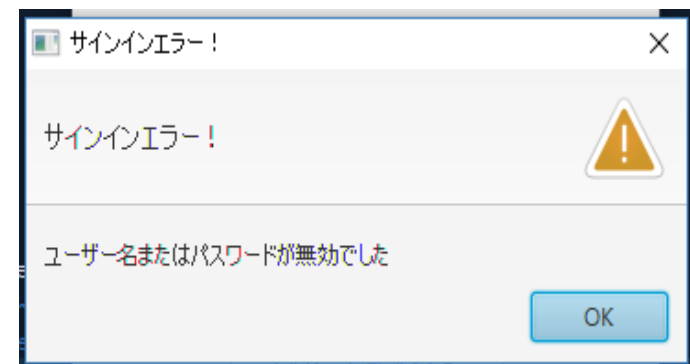
The image shows a Java Swing window titled "Login". The main content area has a light gray background. At the top, the text "Scheduling App" is displayed in a large, blue, sans-serif font. Below this, there are two input fields. The first is labeled "Username" in a small, dark gray font, and the second is labeled "Password" in a small, dark gray font. Both labels are centered above their respective input fields. The input fields have a light gray border and a placeholder text "Enter username" and "Enter password" respectively. Below the input fields, there is a blue button with the text "Sign In" in white.



The image shows a Java Swing dialog box titled "Sign In Error!". The dialog has a light gray background. At the top, the text "Sign In Error!" is displayed in a dark gray font. To the right of the text is a yellow warning icon with a black exclamation mark. Below the text, there is a message "The username or password was not valid!" in a dark gray font. At the bottom right, there is a blue button with the text "OK" in white.



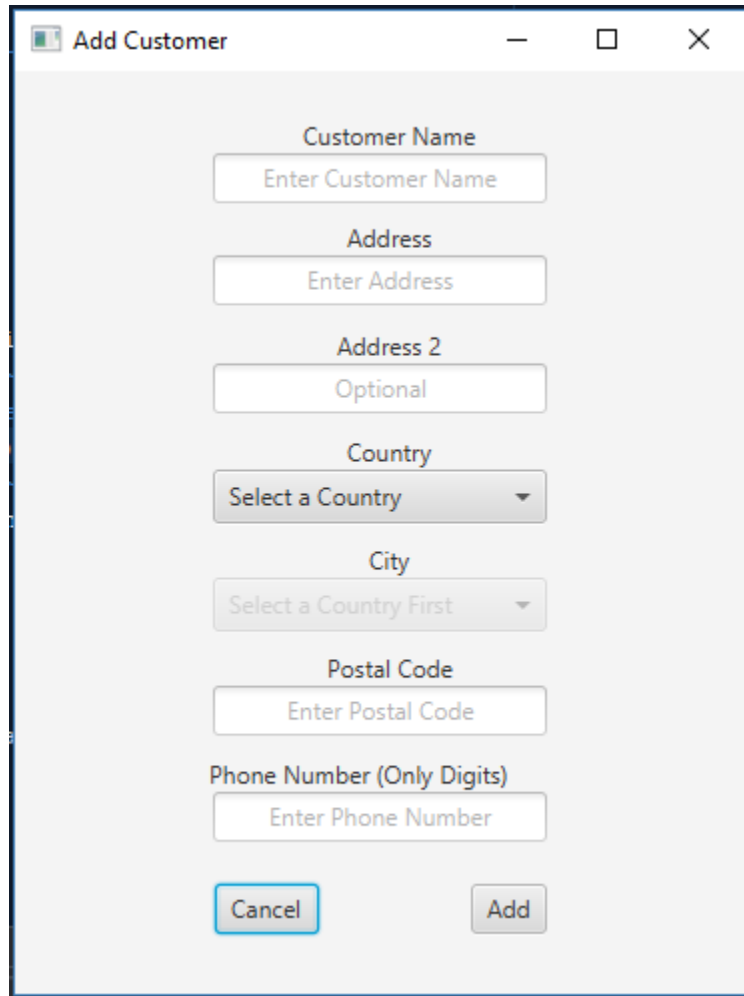
The image shows a Java Swing window titled "ログイン". The main content area has a light gray background. At the top, the text "スケジューリングアプリ" is displayed in a large, blue, sans-serif font. Below this, there are two input fields. The first is labeled "ユーザー名" in a small, dark gray font, and the second is labeled "パスワード" in a small, dark gray font. Both labels are centered above their respective input fields. The input fields have a light gray border and a placeholder text "パスワードを入力する" (Enter password) respectively. Below the input fields, there is a gray button with the text "サインイン" in dark gray.



The image shows a Java Swing dialog box titled "サインインエラー！". The dialog has a light gray background. At the top, the text "サインインエラー！" is displayed in a dark gray font. To the right of the text is a yellow warning icon with a black exclamation mark. Below the text, there is a message "ユーザー名またはパスワードが無効でした" (Username or password is invalid) in a dark gray font. At the bottom right, there is a blue button with the text "OK" in white.

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

**Add a customer:** From the main menu click customer will bring you to the Customer View. Then click add customer button to bring the up the add customer view.

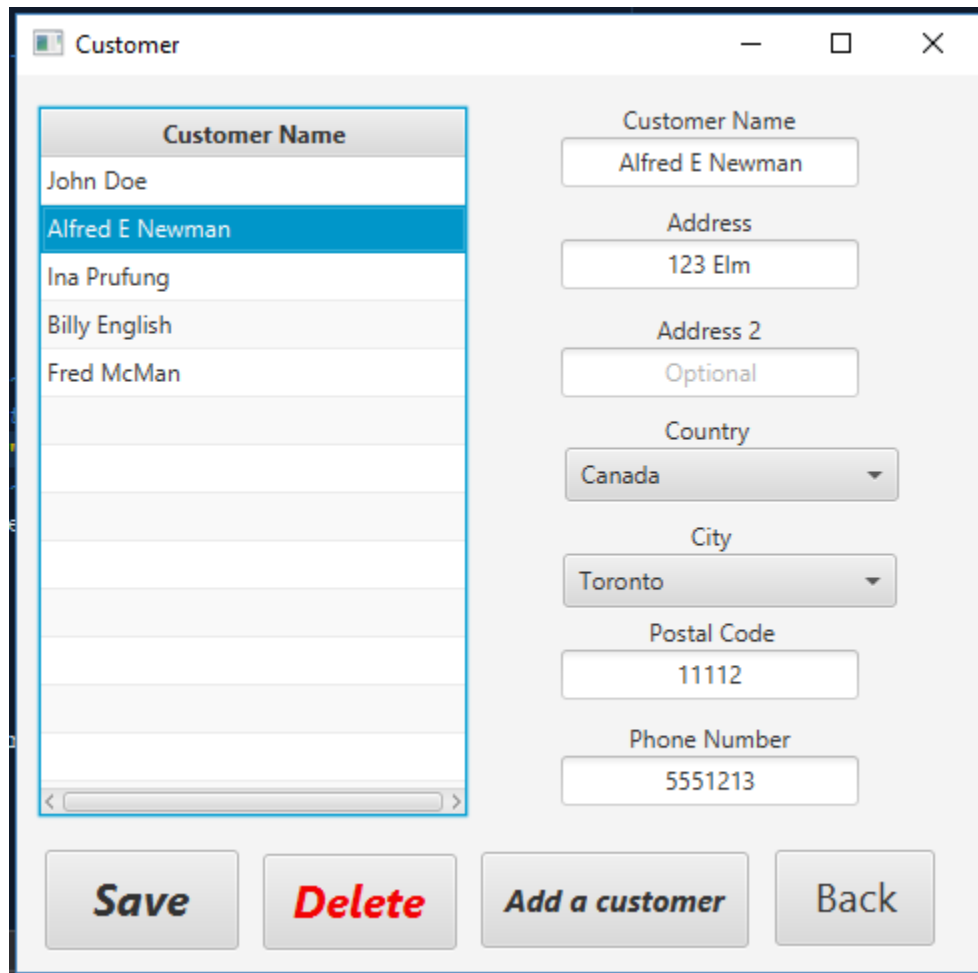


The image shows a screenshot of a web application window titled "Add Customer". The window contains a form with the following fields and controls:

- Customer Name:** A text input field with the placeholder text "Enter Customer Name".
- Address:** A text input field with the placeholder text "Enter Address".
- Address 2:** A text input field with the placeholder text "Optional".
- Country:** A dropdown menu with the text "Select a Country" and a downward arrow.
- City:** A dropdown menu with the text "Select a Country First" and a downward arrow.
- Postal Code:** A text input field with the placeholder text "Enter Postal Code".
- Phone Number (Only Digits):** A text input field with the placeholder text "Enter Phone Number".
- Buttons:** At the bottom of the form, there are two buttons: "Cancel" (highlighted with a blue border) and "Add".

**Update and Delete Customer:** In the Customer View Select a customer update the data in the fields and click save to Update

the customer. Or select a customer and click delete to remove the customer.



A screenshot of a 'Customer' management window. On the left is a list of customer names: John Doe, Alfred E Newman (highlighted), Ina Prufung, Billy English, and Fred McMan. On the right are input fields for the selected customer's details: Customer Name (Alfred E Newman), Address (123 Elm), Address 2 (Optional), Country (Canada), City (Toronto), Postal Code (11112), and Phone Number (5551213). At the bottom are four buttons: Save, Delete, Add a customer, and Back.

Customer Name
John Doe
Alfred E Newman
Ina Prufung
Billy English
Fred McMan

Customer Name: Alfred E Newman

Address: 123 Elm

Address 2: Optional

Country: Canada

City: Toronto

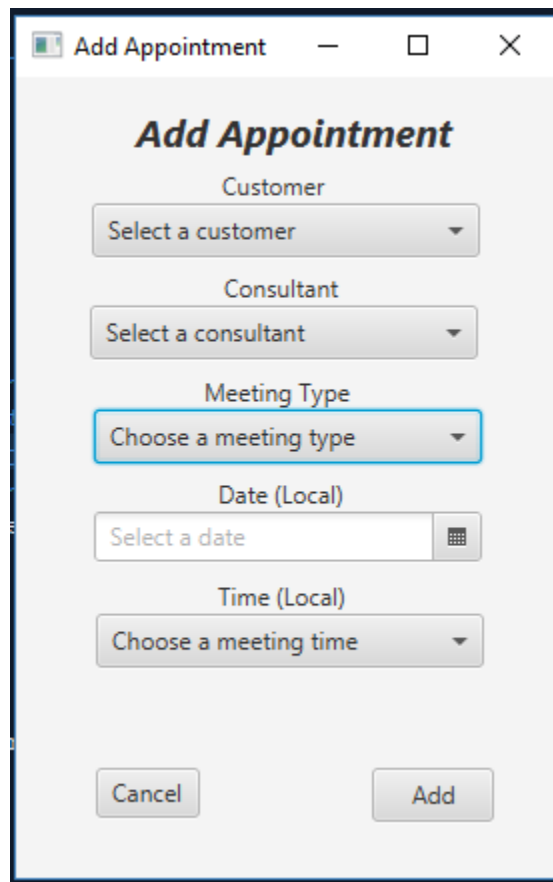
Postal Code: 11112

Phone Number: 5551213

**Save** **Delete** *Add a customer* Back

- 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.

**Add an appointment:** From the Main Menu click the appointments button. This will bring you to the appointment view. Then click the Add Appointment button. Date and meeting time are both entered based on users local time.



The image shows a software window titled "Add Appointment". Inside the window, the title "Add Appointment" is displayed in a bold, italicized font. Below the title, there are five input fields, each with a label and a selection control:

- Customer:** A dropdown menu with the text "Select a customer".
- Consultant:** A dropdown menu with the text "Select a consultant".
- Meeting Type:** A dropdown menu with the text "Choose a meeting type". This field is highlighted with a blue border.
- Date (Local):** A text input field with the text "Select a date" and a calendar icon to its right.
- Time (Local):** A dropdown menu with the text "Choose a meeting time".

At the bottom of the window, there are two buttons: "Cancel" on the left and "Add" on the right.

**Update and Delete an appointment:** In the appointment view select an appointment and edit the date and click save to update or select an appointment and click delete.

The screenshot shows a web application window titled "Appointments". At the top, there are three tabs: "All Appointments", "Next Month", and "Next Week". Below the tabs is a table with four columns: "Customer", "Consultant", "Type", and "Date and Time (UTC)". The table contains six rows of appointment data. The fourth row, for "Billy English" with consultant "test" and type "Initial Meeting" on "2019-07-23 20:15:00.0", is highlighted. Below the table is a form for editing an appointment. It includes dropdown menus for "Customer" (selected: Billy English) and "Consultant" (selected: test). There is a "Date (Local)" field with a calendar icon (selected: 2019/07/23) and a "Time (Local)" dropdown (selected: 1:15 PM). A "Meeting Type" dropdown is also present (selected: Initial Meeting). At the bottom of the form are four buttons: "Save", "Delete" (in red), "Add Appointment", and "Back".

Customer	Consultant	Type	Date and Time (UTC)
John Doe	candace	Follow-Up	2019-07-17 20:15:00.0
Ina Prufung	ed	Initial Meeting	2019-07-19 18:00:00.0
John Doe	test	Initial Meeting	2019-07-20 09:00:00.0
Billy English	test	Initial Meeting	2019-07-23 20:15:00.0
Fred McMan	candace	Interview	2019-07-24 17:00:00.0
John Doe	ed	Interview	2019-07-26 16:00:00.0

Customer	Date (Local)	Meeting Type
Billy English	2019/07/23	Initial Meeting
Consultant	Time (Local)	
test	1:15 PM	

Save Delete Add Appointment Back

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

In the top left corner there are three tabs to show all appointment, Appointments in the next month, and next week.

Customer	Consultant	Type	Date and Time (UTC)
John Doe	candace	Follow-Up	2019-07-17 20:15:00.0
Ina Prufung	ed	Initial Meeting	2019-07-19 18:00:00.0
John Doe	test	Initial Meeting	2019-07-20 09:00:00.0
Billy English	test	Initial Meeting	2019-07-23 20:15:00.0
Fred McMan	candace	Interview	2019-07-24 17:00:00.0
John Doe	ed	Interview	2019-07-26 16:00:00.0

Customer: Billy English Date (Local): 2019/07/23 Meeting Type: Initial Meeting Time (Local): 1:15 PM

Buttons: Save, Delete, Add Appointment, Back

## By Month:

The screenshot shows the 'Appointments' window with the 'Next Month' tab selected. The table below lists the appointments for the next month.

Customer	Consultent	Type	Date and Time (UTC)
John Doe	test	Initial Meeting	2019-07-20 09:00:00.0
Billy English	test	Initial Meeting	2019-07-23 20:15:00.0
Fred McMan	candace	Interview	2019-07-24 17:00:00.0
John Doe	ed	Interview	2019-07-26 16:00:00.0
Fred McMan	candace	Interview	2019-07-26 16:00:00.0
Ina Prufung	luke	Follow-Up	2019-07-31 20:15:00.0

Below the table, there are form fields for adding a new appointment:

- Customer: Billy English
- Date (Local): 2019/07/23
- Meeting Type: Initial Meeting
- Consultent: test
- Time (Local): 1:15 PM

Buttons at the bottom: Save, Delete, Add Appointment, Back.

## By Week:

The screenshot shows the 'Appointments' window with the 'Next Week' tab selected. The table below lists the appointments for the next week.

Customer	Consultent	Type	Date and Time (UTC)
John Doe	test	Initial Meeting	2019-07-20 09:00:00.0
Billy English	test	Initial Meeting	2019-07-23 20:15:00.0
Fred McMan	candace	Interview	2019-07-24 17:00:00.0
John Doe	ed	Interview	2019-07-26 16:00:00.0
Fred McMan	candace	Interview	2019-07-26 16:00:00.0

Below the table, there are form fields for adding a new appointment:

- Customer: Billy English
- Date (Local): 2019/07/23
- Meeting Type: Initial Meeting
- Consultent: test
- Time (Local): 1:15 PM

Buttons at the bottom: Save, Delete, Add Appointment, Back.



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

The user enters appointment dates and time based on there own date and time. The data is stored in the server in UTC and shown in the fields in the users current date/time.

**Add Appointment**

Customer: Alfred E Newman

Consultant: test

Meeting Type: Initial Meeting

Date (Local): 2019/07/19

Time (Local): 10:30 AM

Buttons: Cancel, Add

**Appointments**

Buttons: All Appointments, Next Month, Next Week

Customer	Consultant	Type	Date and Time (UTC)
John Doe	candace	Follow Up	2019-07-17 20:15:00.0
Alfred E Newman	test	Initial Meeting	2019-07-19 17:30:00.0
Ina Prufung	ed	Initial Meeting	2019-07-19 18:00:00.0
John Doe	test	Initial Meeting	2019-07-20 09:00:00.0
Billy English	test	Initial Meeting	2019-07-23 20:15:00.0
Fred McMan	candace	Interview	2019-07-24 17:00:00.0

Customer: Alfred E Newman

Consultant: test

Meeting Type: Initial Meeting

Date (Local): 2019/07/19

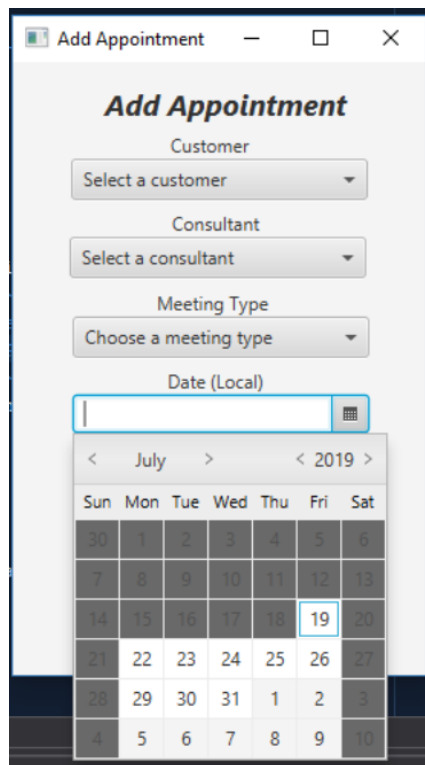
Time (Local): 10:30 AM

Buttons: Save, Delete, Add Appointment, Back

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**

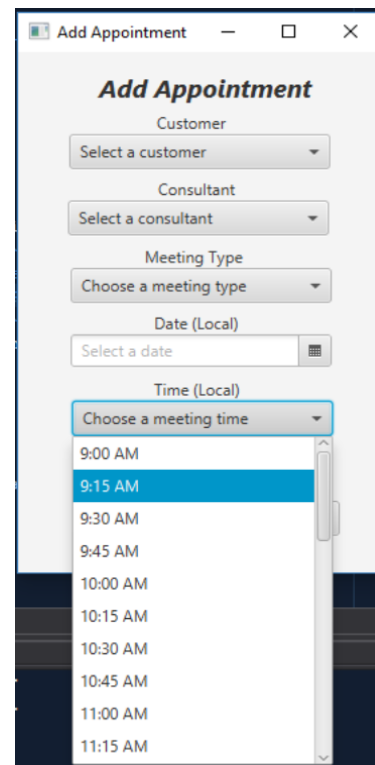
The date picker will only allow the user to select date of today and in the future. The time combo box only allows the user to select time with in business hours.



The screenshot shows the 'Add Appointment' form with the following fields:

- Customer: Select a customer (dropdown)
- Consultant: Select a consultant (dropdown)
- Meeting Type: Choose a meeting type (dropdown)
- Date (Local): Select a date (calendar icon)

The date picker is open, showing a calendar for July 2019. The date 19 is selected.



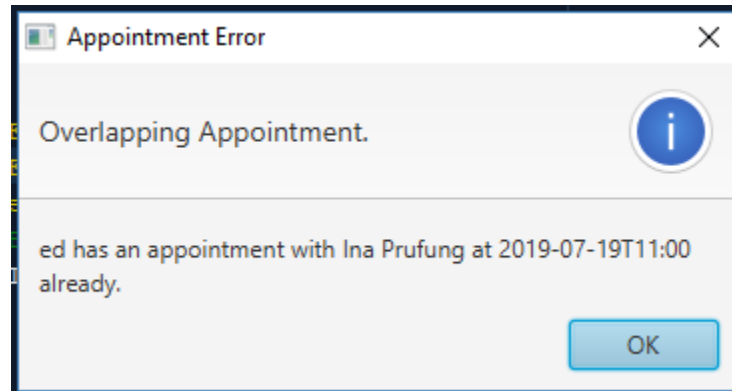
The screenshot shows the 'Add Appointment' form with the following fields:

- Customer: Select a customer (dropdown)
- Consultant: Select a consultant (dropdown)
- Meeting Type: Choose a meeting type (dropdown)
- Date (Local): Select a date (calendar icon)
- Time (Local): Choose a meeting time (dropdown)

The time picker is open, showing a list of times from 9:00 AM to 11:15 AM. The time 9:15 AM is selected.

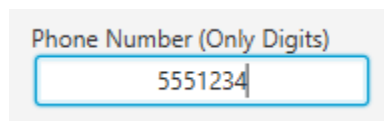
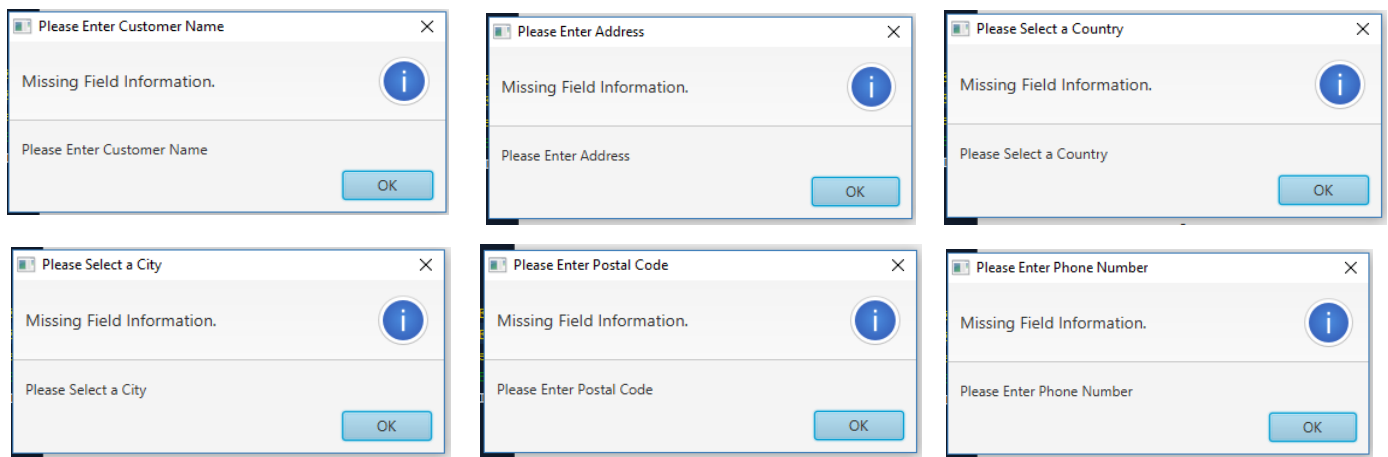
- **scheduling overlapping appointments**

If there is an overlapping appointment with a consultant an alert when an appointment is updated or created. Nothing is changed when an appointment being updated or added. This uses an try/catch.



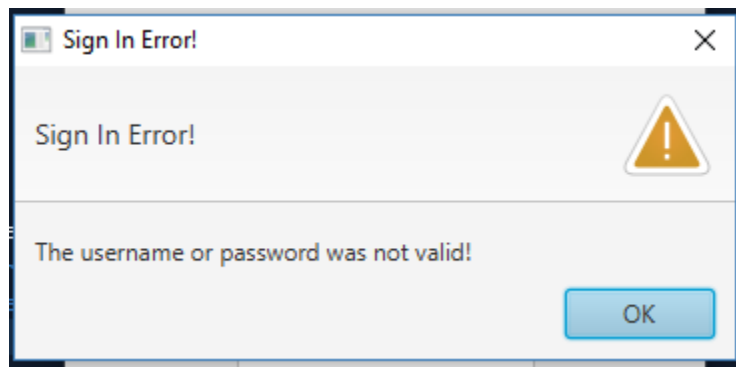
- **entering nonexistent or invalid customer data**

If any fields are left blank besides the address 2 (optional) an alert shows, and nothing is changed. The phone number field only allows numeric key strokes.



- **entering an incorrect username and password**

When an incorrect username and password is entered. This has an throws type of exception.

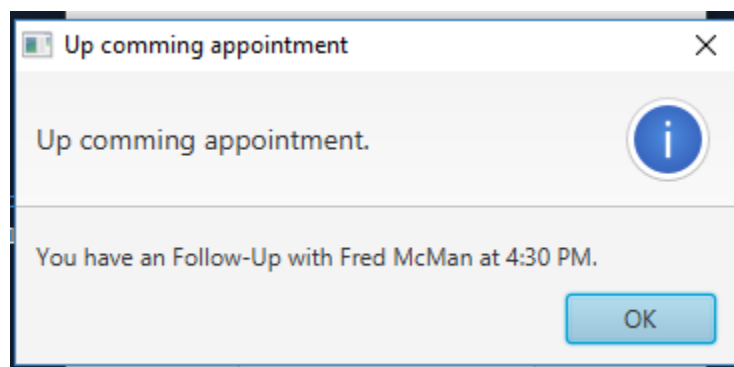


F. Write two or more lambda expressions to make your program more efficient, justifying the use of each lambda expression with an in-line comment.

1. An add listener lambda expression on line 359 in the file CustomerController.java.
2. A Set Cell Value Factory lambda expression on line 366 in the file CustomerController.java.

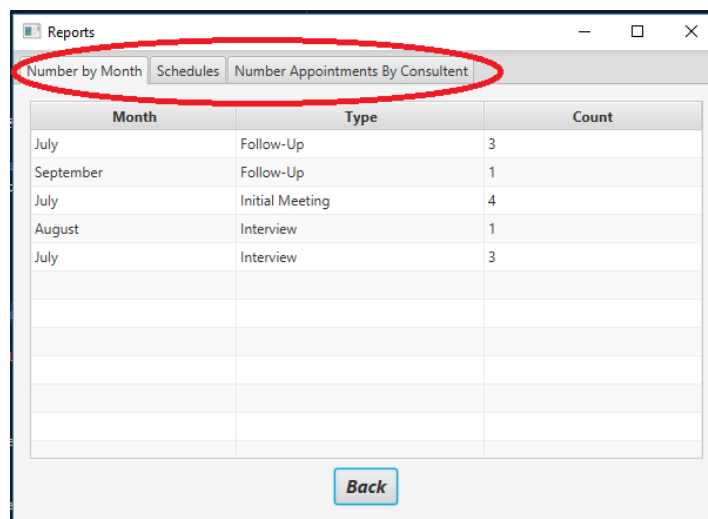
H. Write code to provide an alert if there is an appointment within 15 minutes of the user's log-in.

When a user logs in and there is an appointment related to the user that start within 15 minutes an alert is presented letting the user know what time, the type of meeting, and with who the meeting with.



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

Once again, I use a tab system to show the types of reports.



- number of appointment types by month

The screenshot shows a software window titled 'Reports'. It has three tabs: 'Number by Month' (which is selected and circled in red), 'Schedules', and 'Number Appointments By Consultant'. Below the tabs is a table with three columns: 'Month', 'Type', and 'Count'. The table contains five rows of data, followed by five empty rows. At the bottom of the window is a 'Back' button.

Month	Type	Count
July	Follow-Up	3
September	Follow-Up	1
July	Initial Meeting	4
August	Interview	1
July	Interview	3

- the schedule for each consultant

Select an Consultant from the combo box.

The screenshot shows a software interface for viewing reports. The 'Reports' window has three tabs: 'Number by Month', 'Schedules' (which is active), and 'Number Appointments By Consultant'. Below the tabs is a dropdown menu labeled 'Select an Consultant', which is circled in red. Underneath the dropdown is a table with three columns: 'Customer', 'Type of Appointment', and 'Date of Appointment (Local)'. The table is currently empty, with the text 'No content in table' centered in the body. At the bottom of the window is a 'Back' button.

**Reports**

Number by Month | Schedules | Number Appointments By Consultant

ed ▼

Customer	Type of Appointment	Date of Appointment (Local)
Ina Prufung	Initial Meeting	7/19/19 11:00 AM
John Doe	Interview	7/26/19 9:00 AM
Ina Prufung	Follow-Up	9/20/19 2:00 PM

**Back**

- one additional report of your choice

The report for my choice shows the consultants and the total number of meeting they have in the server.

Reports

Number by Month

Schedules

Number Appointments By Consultant

ed

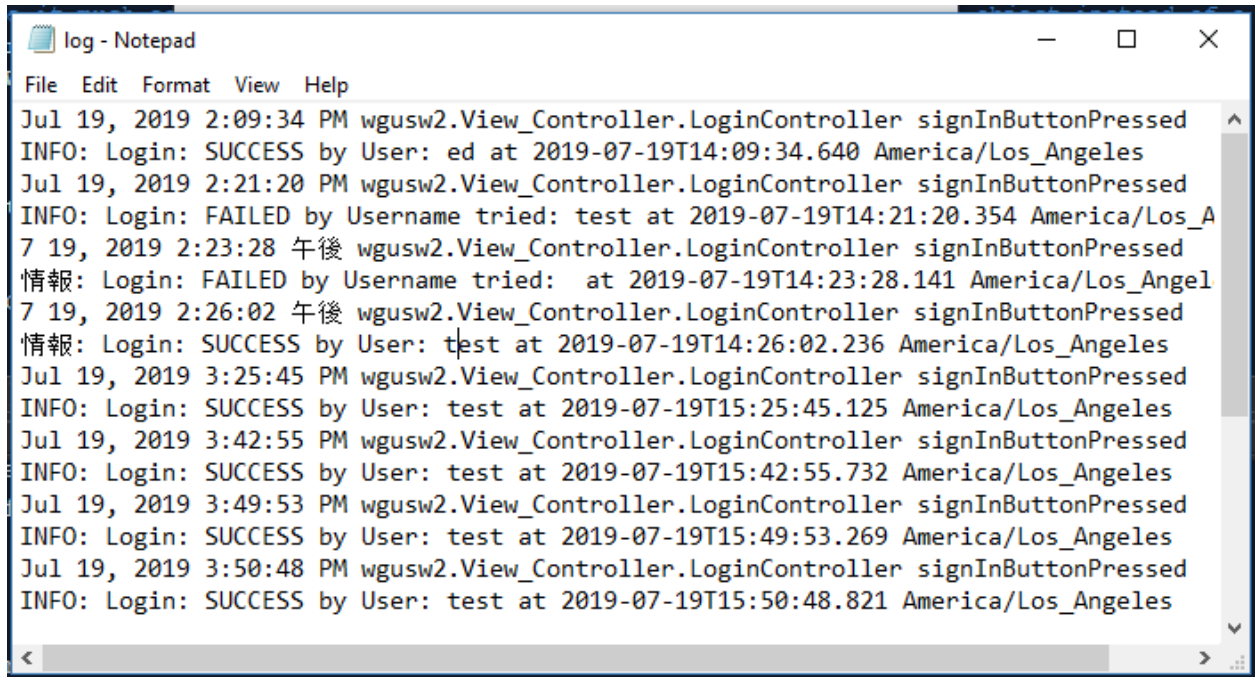
Customer	Type of Appointment	Date of Appointment (Local)
Ina Prufung	Initial Meeting	7/19/19 11:00 AM
John Doe	Interview	7/26/19 9:00 AM
Ina Prufung	Follow-Up	9/20/19 2:00 PM

Back



- 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.

You can see the file by clicking the Logs button on the bottom left of the Main Menu View.



```
log - Notepad
File Edit Format View Help
Jul 19, 2019 2:09:34 PM wgu2.View_Controller.LoginController signInButtonPressed
INFO: Login: SUCCESS by User: ed at 2019-07-19T14:09:34.640 America/Los_Angeles
Jul 19, 2019 2:21:20 PM wgu2.View_Controller.LoginController signInButtonPressed
INFO: Login: FAILED by Username tried: test at 2019-07-19T14:21:20.354 America/Los_Angeles
Jul 19, 2019 2:23:28 午後 wgu2.View_Controller.LoginController signInButtonPressed
情報: Login: FAILED by Username tried: at 2019-07-19T14:23:28.141 America/Los_Angeles
Jul 19, 2019 2:26:02 午後 wgu2.View_Controller.LoginController signInButtonPressed
情報: Login: SUCCESS by User: test at 2019-07-19T14:26:02.236 America/Los_Angeles
Jul 19, 2019 3:25:45 PM wgu2.View_Controller.LoginController signInButtonPressed
INFO: Login: SUCCESS by User: test at 2019-07-19T15:25:45.125 America/Los_Angeles
Jul 19, 2019 3:42:55 PM wgu2.View_Controller.LoginController signInButtonPressed
INFO: Login: SUCCESS by User: test at 2019-07-19T15:42:55.732 America/Los_Angeles
Jul 19, 2019 3:49:53 PM wgu2.View_Controller.LoginController signInButtonPressed
INFO: Login: SUCCESS by User: test at 2019-07-19T15:49:53.269 America/Los_Angeles
Jul 19, 2019 3:50:48 PM wgu2.View_Controller.LoginController signInButtonPressed
INFO: Login: SUCCESS by User: test at 2019-07-19T15:50:48.821 America/Los_Angeles
```