

READ ME

Login access:

User: Ana90, **Password:** test

User: Joe11, **Password:** bluecake

User: Sarahh, **Password:** waterfall

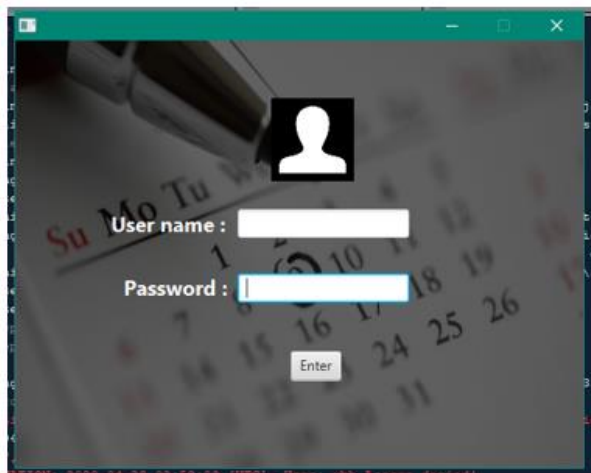
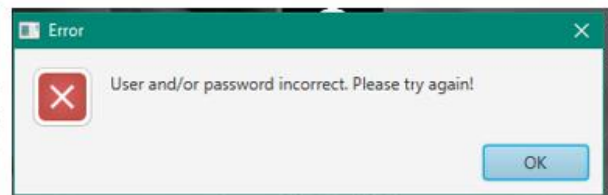
Note: The login is case sensitive

Requirements

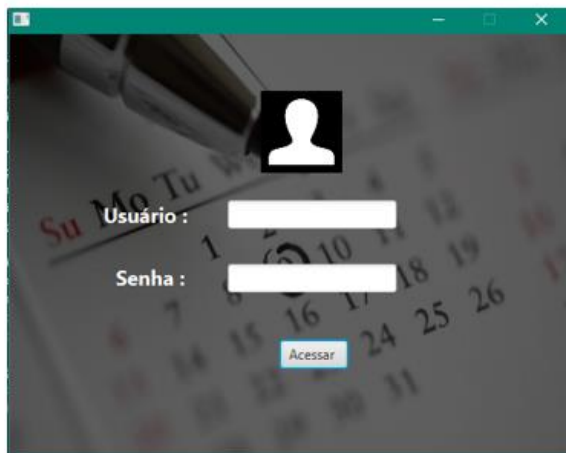
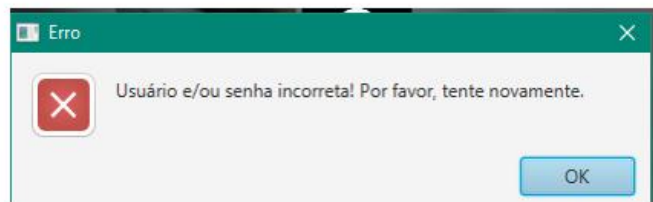
A. Create a log-in form that can determine the user's location and translate log-in and error control messages (e.g., "The username and password did not match.") into **two** languages.

- The second language chosen is **Portuguese from Brazil** (pt_br)

English

A screenshot of a login form window. The background is a dark image of a calendar and a pen. At the top center is a white silhouette of a person's head and shoulders. Below it, the text "User name :" is followed by a white text input field. Underneath that, the text "Password :" is followed by a white text input field. At the bottom center is a small button labeled "Enter".

Portuguese

A screenshot of a login form window in Portuguese. The background is a dark image of a calendar and a pen. At the top center is a white silhouette of a person's head and shoulders. Below it, the text "Usuário :" is followed by a white text input field. Underneath that, the text "Senha :" is followed by a white text input field. At the bottom center is a small button labeled "Acessar".

It takes few seconds to load the fields after I choose the customer from tableView. I believe that It is because my computer is old and slow. To avoid any trouble, I add a note in the screen to let the user know that it may happen.

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.

- You can update or delete just after select one appointment from the table view. One Alert dialog will show case any appointment is not selected.
- All the appointments have duration of 55 minutes.

Appointment Scheduler

Customer

New

Update

Reports

By Month

By Consultant

By Customer

Show

Appointments

Show : All

Customer	Service	Type	Consultant	Start time	End time	Location
George Lee	Health	New appointment	Dr. Meredith Grey	2020-04-29 10:00:00	2020-04-29 10:55:00	Pheonix, Arizona
Alex Green	Health	Follow up	Dr. Alex Karev	2020-05-04 10:00:00	2020-05-04 10:55:00	London, England
Clare Walker	Beauty	New appointment	Katy White	2020-05-13 15:00:00	2020-05-13 15:55:00	London, England
Kathy Mullier	Spa	New appointment	Simon Park	2020-05-13 15:00:00	2020-05-13 15:55:00	New York, New Y...
Clare Walker	Beauty	Follow up	Valeria Sales	2020-04-30 08:00:00	2020-04-30 08:55:00	Pheonix, Arizona

New

Edit

Delete

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

It is located in the main screen, under appointments label. You can choose from the combo box the options to show all appointments, to filter by month and by this week. Once you select the by month, a second combo box will appear at right with the month's options.

The screenshot shows a web application interface titled "Appointments". At the top, there are two dropdown menus: "Show : Month" and "Month : Select". Below these is a table with the following columns: Customer, Service, Type, Consultant, Start time, End time, and Location. The table contains five rows of appointment data. At the bottom of the interface, there are three buttons: "New", "Edit", and "Delete".

Customer	Service	Type	Consultant	Start time	End time	Location
George Lee	Health	New appointment	Dr. Meredith Grey	2020-04-29 10:00:00	2020-04-29 10:55:00	Pheonix, Arizona
Alex Green	Health	Follow up	Dr. Alex Karev	2020-05-04 10:00:00	2020-05-04 10:55:00	London, England
Clare Walker	Beauty	New appointment	Katy White	2020-05-13 15:00:00	2020-05-13 15:55:00	London, England
Kathy Muller	Spa	New appointment	Simon Park	2020-05-13 15:00:00	2020-05-13 15:55:00	New York, New Y...
Clare Walker	Beauty	Follow up	Valeria Sales	2020-04-30 08:00:00	2020-04-30 08:55:00	Pheonix, Arizona

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**

if else statements were used, located at View_Controller package > NewAppointmentController > lines 290 to 296.

- **scheduling overlapping appointments**

if else statements were used, located at View_Controller package > NewAppointmentController > line 286.

An appointment is overlapped if already exists an appointment with the same consultant, at the same time.

- **entering nonexistent or invalid customer data**

It was used a mix of if else and try catch statements. They are located in different methods at View_Controller package > AddCustomerController / UpdateCustomerController.

- **entering an incorrect username and password**

if else statements were used, located at View_Controller package > LoginController lines 102 to 117.

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.

I create three lambda expressions. Located at:

- View_Controller package > MainScreenController > line 486

- View_Controller package > NewAppointmentController > line 256
 - DAO package > DBConnection > line 35
- All the interfaces are located at the Utils package

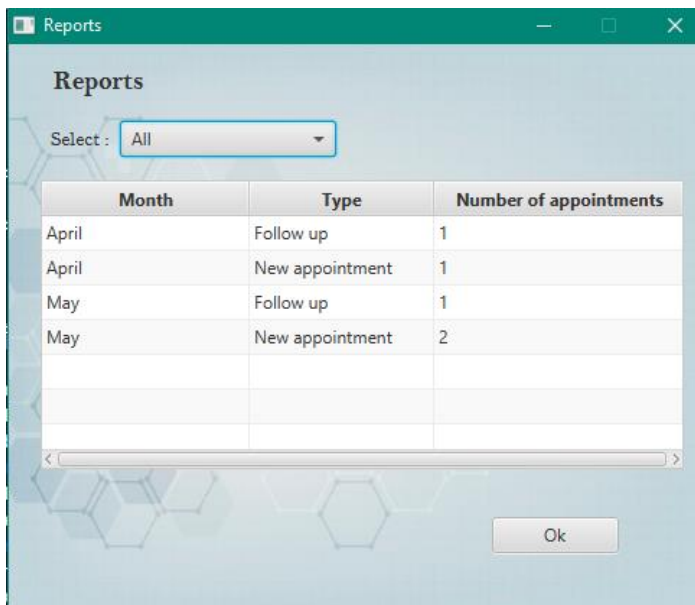
H. Write code to provide an alert if there is an appointment within 15 minutes of the user's login.



Note: If you are going to insert the appointment through MySQL workbench (since you only can schedule appointments every one hour). Insert the start time as ex: 2020-04-28 04:24:39.0 with the last 0 at the end (in red).

- 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 (I choose show the reports by customer)

In the report screen, initially will show all the results for the selected report type. In the combo box you can choose one specific month, consultant or customer



- 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.

log - Bloco de Notas				
Arquivo	Editar	Formatar	Exibir	Ajuda
2020-04-20	18:27:14	[UTC]	- User: test	Access granted!
2020-04-20	18:27:55	[UTC]	- User: test	Access granted!
2020-04-20	18:58:29	[UTC]	- User: test	Access granted!
2020-04-20	19:13:39	[UTC]	- User: test	Access granted!
2020-04-20	19:14:40	[UTC]	- User: sarahh	Access granted!
2020-04-20	20:52:34	[UTC]	- User: test	Access granted
2020-04-20	21:55:04	[UTC]	- User: sarahh	Access granted
2020-04-20	22:03:35	[UTC]	- User: test	Access granted
2020-04-20	22:08:43	[UTC]	- User: t	Access denied!
2020-04-20	22:09:07	[UTC]	- User: test	Access granted
2020-04-22	19:43:13	[UTC]	- User: test	Access granted
2020-04-22	19:44:33	[UTC]	- User: test	Access granted
2020-04-22	20:12:34	[UTC]	- User: test	Access granted
2020-04-22	20:13:15	[UTC]	- User: test	Access granted
2020-04-22	20:35:42	[UTC]	- User: test	Access granted
2020-04-22	20:41:08	[UTC]	- User: test	Access granted
2020-04-22	21:04:20	[UTC]	- User: test	Access granted
2020-04-22	21:05:09	[UTC]	- User: test	Access granted
2020-04-22	21:06:31	[UTC]	- User: test	Access granted
2020-04-22	21:09:16	[UTC]	- User: test	Access granted
2020-04-22	21:20:36	[UTC]	- User: test	Access granted
2020-04-22	21:24:57	[UTC]	- User: test	Access granted
2020-04-22	21:29:39	[UTC]	- User: test	Access granted