

Lab 4

Objectives:

- JavaFX

Task: Create an app for Starfleet

News of your app development skills has crossed the quadrant! Starfleet command has asked for your assistance organizing their personnel files and critical data. Starfleet has provided additional personnel data, including photos and biographical information. You'll build an app for Starfleet captains to log in and view the crew members aboard their assigned starship.

Getting Started

To begin this lab, create a new JavaFX project named **abc123-lab4**, and create the following:

- **Main.java** - in the application package
- **LoginController.java** - in the application.controller package
- **PersonnelController.java** - in the application.controller package
- **User.java** - in the application.model package
- **CrewMember.java** - in the application.model package
- **Starship.java** - in the application.model package
- **Login.fxml**
- **Personnel.fxml**

(fxml & data files will be at the top of the Eclipse project - creating a new FXML document will put them in this default location)

App Design

Your program will show a view similar to the one shown below when the app is run:



The image shows a login form for Starfleet Personnel. It features the Starfleet Command logo on the left, which is a circular emblem with a blue background, a white star, and a yellow swoosh. The text "STARFLEET COMMAND" is at the top and "UNITED FEDERATION OF PLANETS" is at the bottom. To the right of the logo, the title "Starfleet Personnel" is displayed in a large, bold, blue font. Below the title, there are two input fields. The first is labeled "Enter your username:" and contains the text "kirk". The second is labeled "Passphrase:" and contains a series of asterisks. Below these fields is a black button with the text "Log In" in white.

Starfleet Personnel

Enter your username:

Passphrase:

This view will be the **Login.fxml**.

If the user clicks on the "Login" button on the above view, the app will display a GUI similar to the following:



Log Out

Welcome, Captain Kirk

USS Enterprise NCC1701-A



Commanding Officer
Captain James T. Kirk



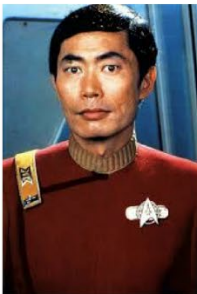
First Officer
Commander Spock



Communications Officer
Lieutenant Nyota Uhura



Chief Engineering Officer
Lieutenant Commander Montgomery Scott



Helmsman
Lieutenant Hikaru Sulu



Navigator
Ensign Pavel Chekov



Chief Medical Officer
Lieutenant Commander Leonard McCoy



Nurse
Crewman Christine Chapel

This view will be the **Personnel.fxml**.

You may customize your app however you choose - this includes images, sizes, fonts, colors, size of the app, configuration.

Remember to ensure your app works on all display sizes.

For this lab, you can do this by making your app **no larger than 800x800**.

The app must have the following GUI components on the first view:

- A label for the app (shown here as "Starfleet Personnel")
- 2 labels for the user name and password
- 1 text field (an editable field)
- 1 password field (an editable field that is masked)
- 1 "Log in" button

On the second view, the app must have the following GUI components:

- A label for the app (shown here as "Welcome, Captain Kirk")
- A photo and one or more labels containing the name, rank, and position of each crew member.
- 1 "Log out" button, which should return the user to the login view and log them out.

For each captain that logs in, the positions should remain the same. That is, the top left crew member should always be the Commanding Officer, next should be the First Officer, and so on.

Model

The **User.java** class will represent users of this application. Each user has a username and a password. Their username is their last name, in all lowercase.

The file **users.csv** contains information for **Users** of the app. Each line in the file contains a user's information, in the following order: name,password. An example of the file follows:

```
kirk,khaaannnn  
picard,teaearlgreyhot
```

There will be a class method called **validate** which should take in a user name and password. This method should be called from the controller.

- If the user exists in the app (if their user name is found in the users.csv file), then the password should be verified. If the given password is correct, then they should be permitted to move to the next view. This method will return a User object, based on a given username.
- If the user does not exist in the app (if their user name is not found in the users.csv file), or if the password provided is incorrect, they should not be permitted to move to the next view.

The **Starship.java** and **CrewMember.java** classes are defined as in the previous lab. **Note that the Fleet may be included in this application - the fleet.csv data file may provide useful information.**

All crew member data is given in the file **personnel.csv**, which has been updated for this new application.

The Starship class should be called upon by the controller of this application to load personnel from the given data file.

The data files outlined above have been provided as data.zip and images.zip.

All classes in the model must always have getters and setters for all class variables. Constructors are required for all required variables in a class.

Making it Work

Main.java will launch the application and show Login.fxml. **LoginController** will be the event handler for this view.

The user should enter a user name and a password, then click the "Log In" button.

The **User** class will validate the entered information.

If the user provided correct credentials, the **Personnel.fxml** view should be shown.

If the user provides incorrect credentials, show an error message on the Login view. *Hint: try adding an extra label on the view, just for error messages.*

SceneBuilder allows you to change font colors!

PersonnelController will be the event handler for Personnel.fxml. The current user's registry information should be used to obtain all personnel assigned to their starship. These crew members should be displayed on the view when a user logs in. When the user logs out, they must log back in again (with valid username and password) to see the personnel view.

*Note that the controllers in your application should **never** read files or update data. Instead, to follow MVC, these classes should call upon the model classes to complete these tasks.*

Testing and Exceptions

Your app should hand exceptions and errors as previously described - invalid input in the log in view. Test out the app to ensure this is working as expected.

In addition, try logging in as different users to ensure the second view is populated correctly. In the event that a user logs in and there is no associated personnel information, they should receive a message on the second view.

Submission: You must export the Eclipse project, including all files & dependencies for the project (this includes images, text files, fxml, etc). As always, follow the instructions on the lab guidelines.

Rubric:

- (25pts) Correctness - app functions as described.
- (15pts) MVC - app is implemented as described, adhering to MVC design pattern.
- (10pts) Login
- (20pts) Personnel
- (20pts) Model - CrewMember and Starship
-

Submissions which do not compile will receive a maximum of 20 points total.