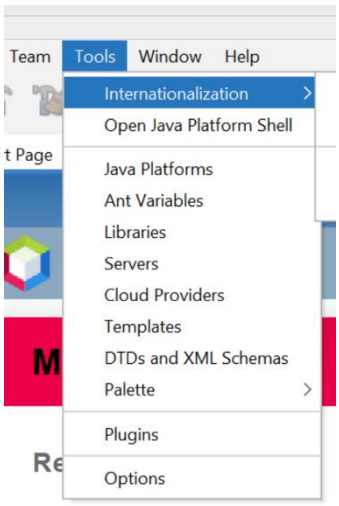
Programmazione avanzata

Lezione 3

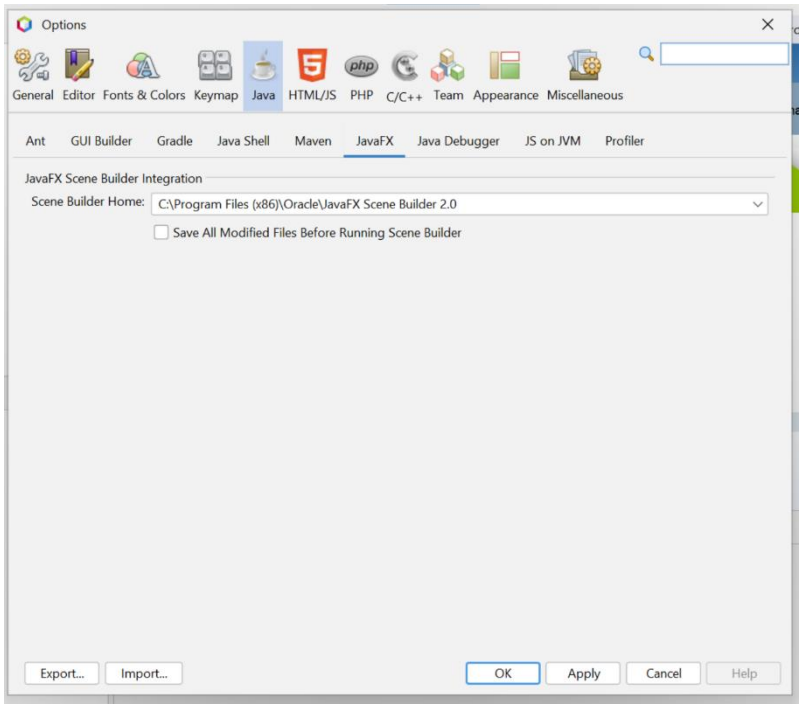
Operazioni di base con JavaFX per applicazioni con grafica

Configurazione

Prima di cominciare a sviluppare un'applicazione con interfaccia grafica utilizzando la framework JavaFX dobbiamo agganciare il tool JavaFX Builder con netbeans andando in Tools->Options.



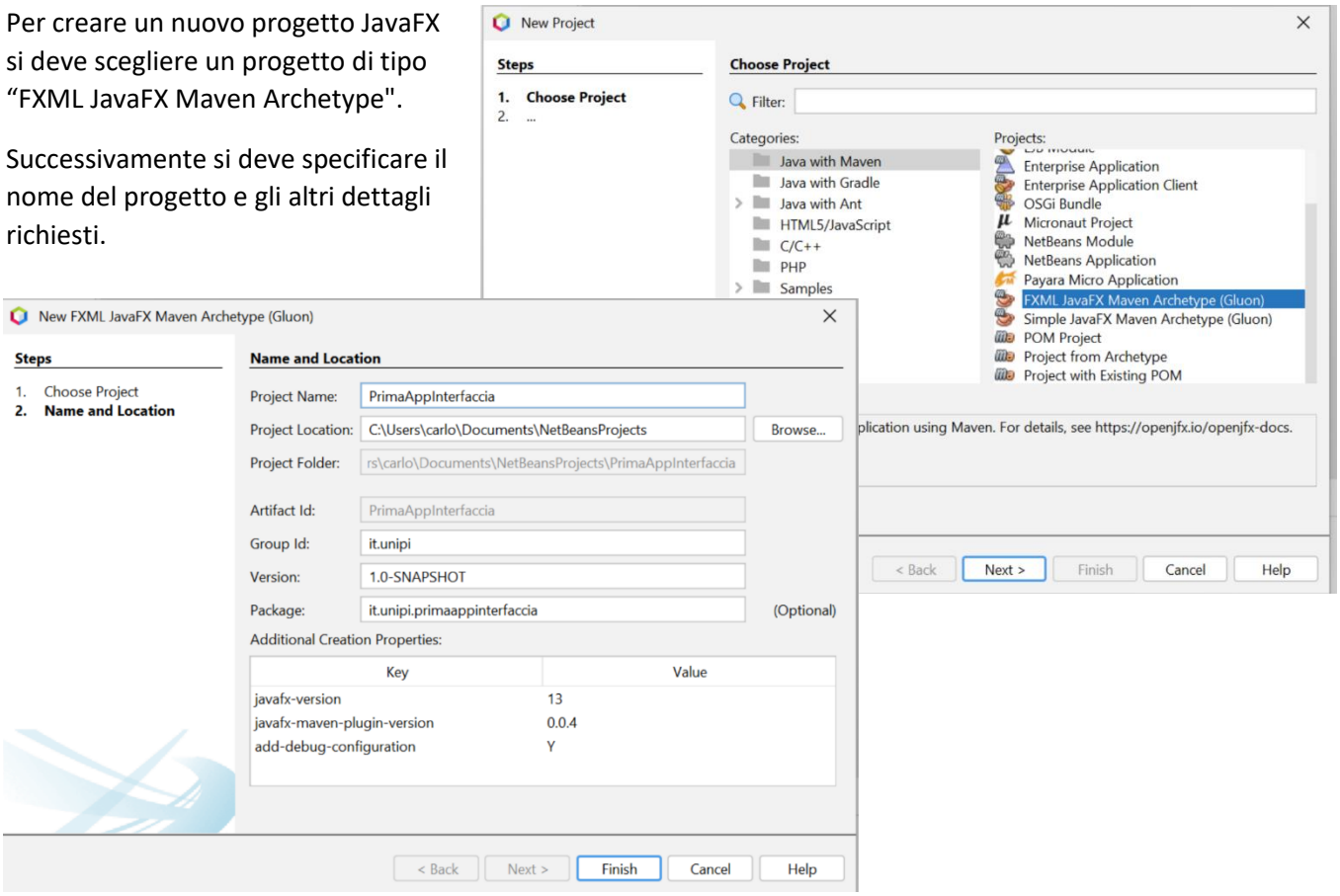
Su questo pannello selezioniamo il tab Java e poi inseriamo il path del tool.



Creazione progetto con JavaFX

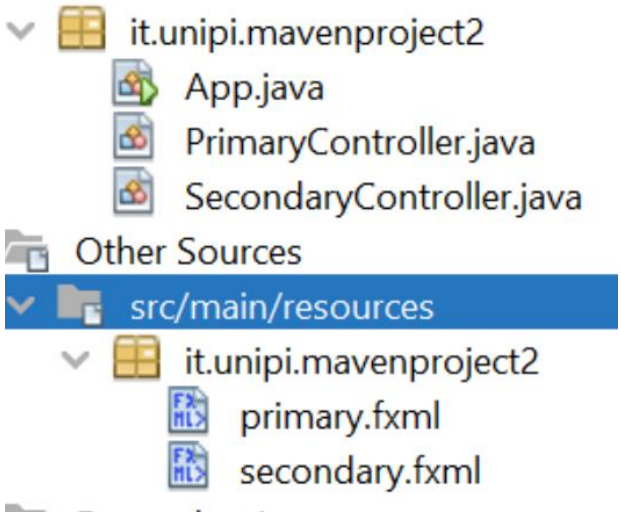
Per creare un nuovo progetto JavaFX si deve scegliere un progetto di tipo “FXML JavaFX Maven Archetype".

Successivamente si deve specificare il nome del progetto e gli altri dettagli richiesti.



Netbeans crea un progetto JavaFX con anche delle funzionalità di scheletro base con una semplice app con due interfacce grafiche. Provate l’applicazione per vederne il funzionamento. L’applicazione di scheletro nel dettaglio è composta dai seguenti files:

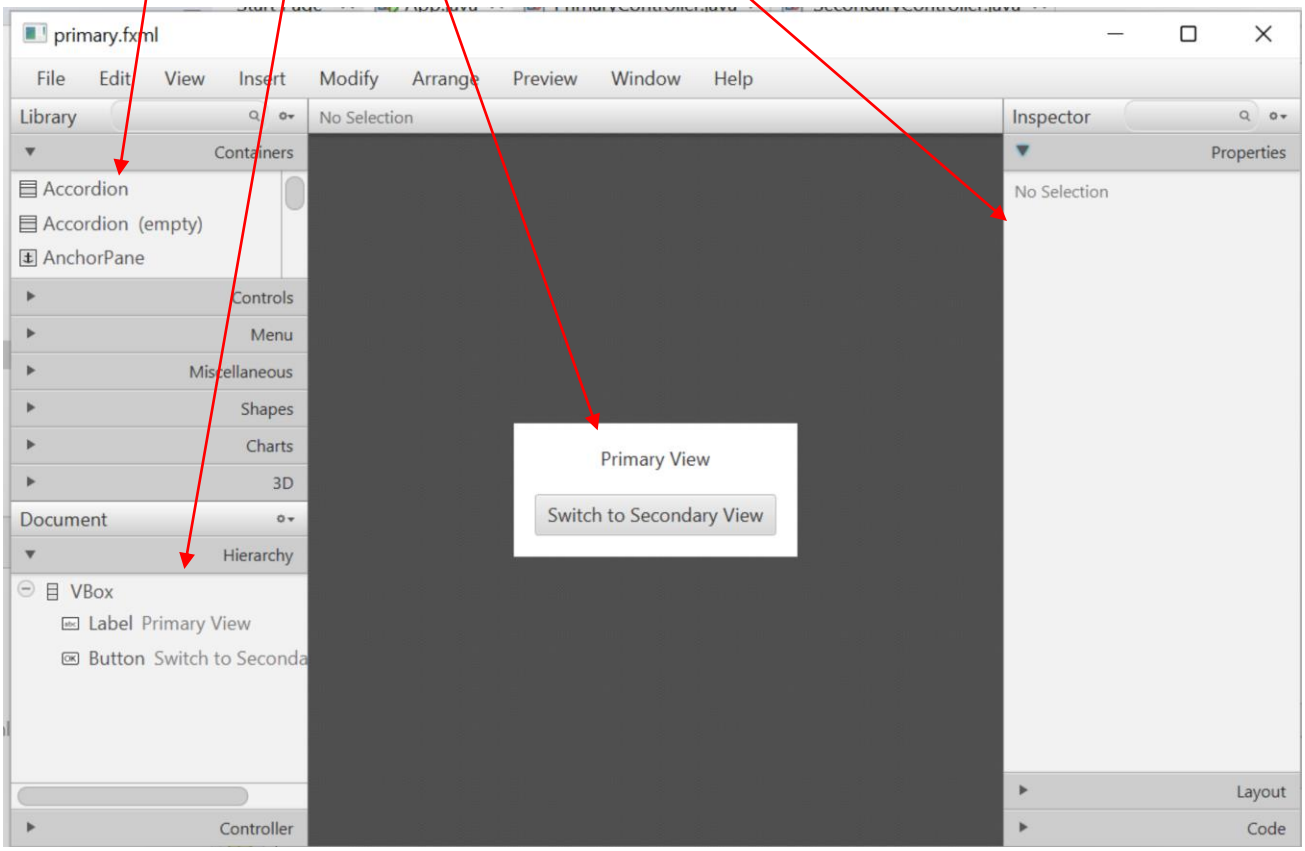
- App.java che contiene la classe principale App che contiene il codice principale dell’applicazione
- PrimaryController.java che contiene la classe PrimaryController che implementa le funzionalità relative alla prima interfaccia
- SecondaryController.java che contiene la classe SecondaryController che implementa le funzionalità relative alla seconda interfaccia



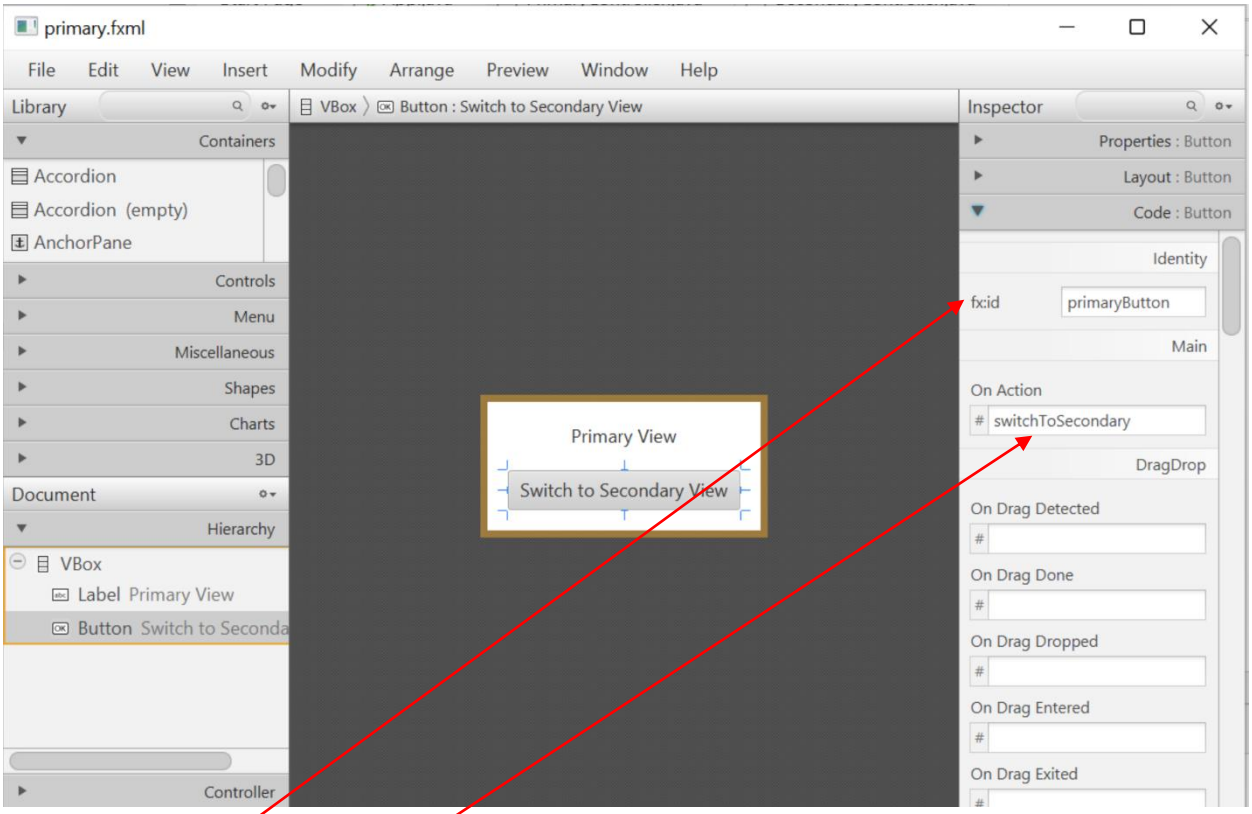
- Due files primary.fxml e secondary.fxml che contengono un codice XML attraverso il quale vengono descritto il layout e le proprietà delle due interfacce.

Se clicchiamo sui due file fxml si apre il JavaFX build editor che permette di modificare l'interfaccia con le sue componenti:

1. Una vista dell'interfaccia
2. Una barra con elementi che possono essere inseriti
3. Un elenco degli elementi inseriti al momento
4. Una colonna con le proprietà dell'elemento selezionato



Una volta selezionato un elemento dall'elenco degli elementi inseriti si possono vedere le proprietà. Il tab "code" in particolare permette di configurare le proprietà dell'elemento e di collegarlo al codice della classe di controllo associata.



I seguenti campi in particolare sono significativi:

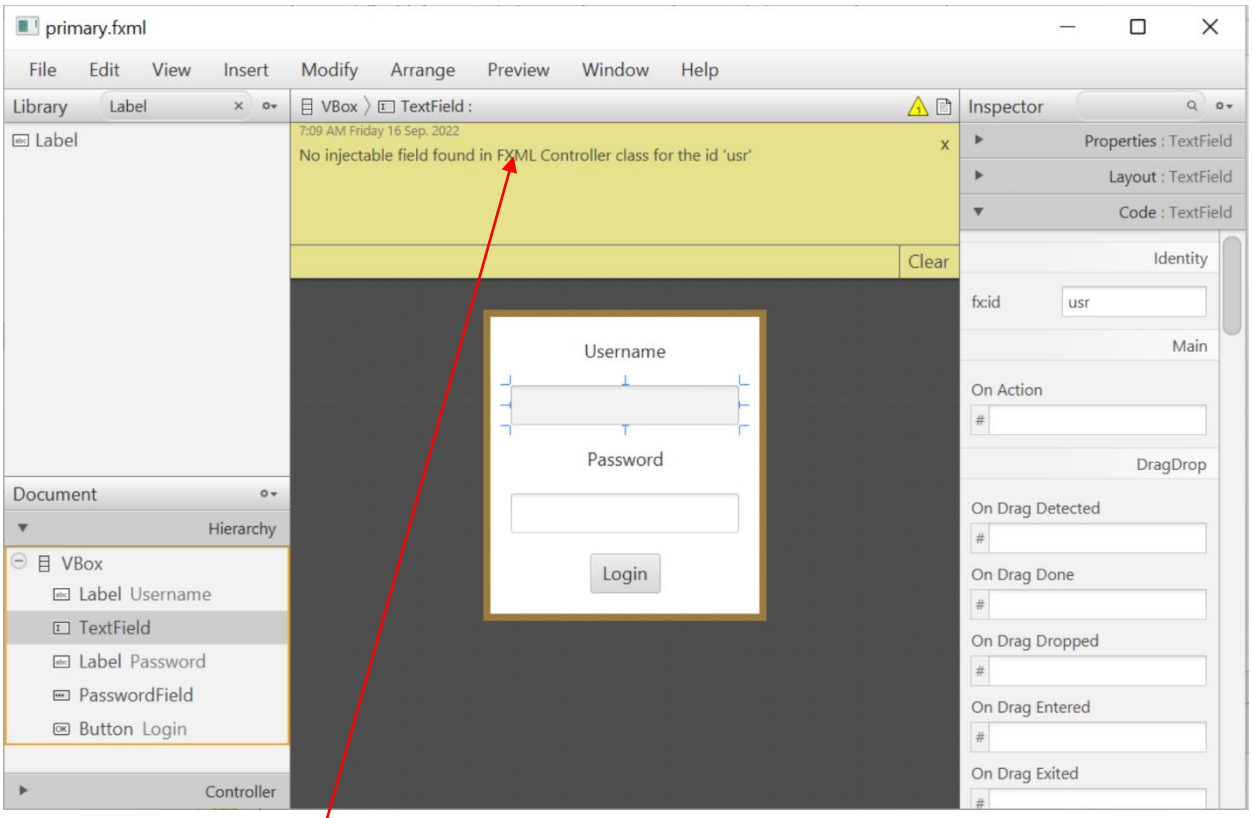
- **fx:id** – rappresenta il nome della variabile che verrà utilizzata nel codice Java per accedere dal codice alle funzionalità dell'elemento e per modificarne le caratteristiche.
- **On Action** – rappresenta il nome della funzione che verrà invocata automaticamente quando l'azione principale dell'elemento viene eseguita dall'utente, e.g. quando viene premuto il bottone in questo caso.

Esercizio

Creare una prima applicazione con JavaFX e lanciarla.

JavaFX Scene Builder

Proviamo a modificare l'interfaccia in modo da ottenere una finestra di login:



Ogni bottone e campo testo dovra' avere un fx:id assegnato, nel caso in cui questo non sia gia' presente nel codice viene mostrato un messaggio d'errore.

In questo caso la classe del controllore dovra' essere aggiornata includendo le seguenti variabili che verranno automaticamente collegate all'elemento grafico.

```
@FXML private TextField usr;
@FXML private PasswordField pwd;
```

Supponiamo di voler cambiare il comportamento del pulsante, in modo da passare alla seconda finestra solo se l'utente ha inserito un username e password corretti. In questo caso dobbiamo modificare il

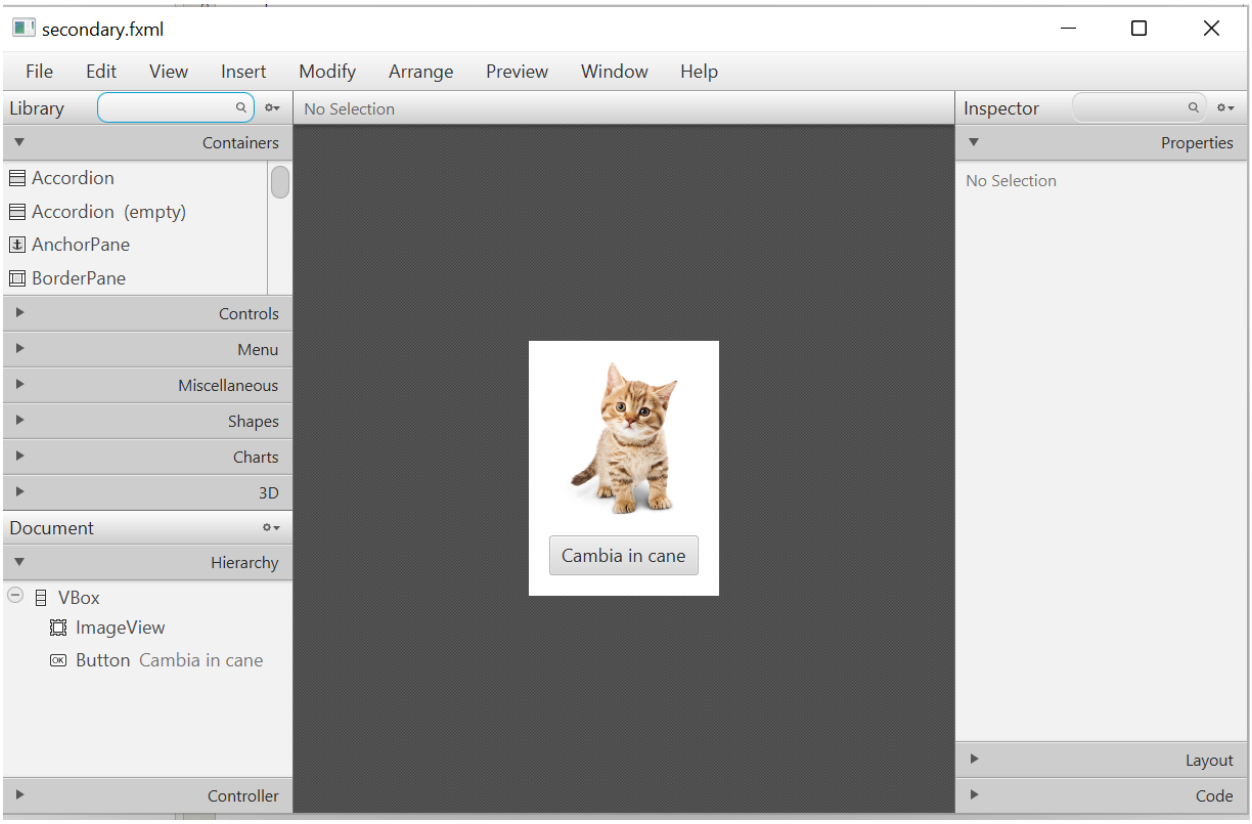
codice della funzione associata all'evento "On Action" del pulsante.

```
@FXML
private void switchToSecondary() throws IOException {
    System.out.print("premuto");

    if(usr.getText().equals("carlo") && pwd.getText().equals("segreto")){
        System.out.print("password corretta");
        App.setRoot("secondary");
    }
}
```

In questo codice si recupera il valore corrente dei due campi di testo e poi si passa alla nuova finestra tramite il comando App.setRoot che richiama una funzione definita nella classe principale che serve per caricare una finestra diversa da quella corrente.

Proviamo adesso a modificare anche la seconda finestra introducendo un'immagine. L'immagine deve essere caricata nel progetto e il path deve essere specificato tra le proprietà.



E proviamo a cambiare il comportamento del pulsante in modo tale da cambiare l'immagine quando questo viene premuto attraverso il seguente codice[1]:

[1] Nota: il nome della funzione associata all'azione è stato cambiato, il nome può essere cambiato cambiando il nome della funzione associata a "On Action"

```
@FXML
private void bottonePremuto() throws IOException {
    logger.debug("Cambio in cane premuto!");
    imgCat.setImage(new Image(getClass().getResource("img/dog.png").toExternalForm()));
```

Esercizio

Modificare l'applicazione in modo da una prima finestra di login che mostra la seconda finestra solo quando l'utente introduce le credenziali corrette. Includere nella seconda finestra un'immagine e un pulsante. Il pulsante una volta premuto deve mostrare un cane sostituendo l'immagine inserita inizialmente.

Creazione dinamica degli elementi

Gli elementi dell'interfaccia possono essere aggiunti anche dinamicamente all'interno del codice. Questo può essere fatto per creare degli elementi che non sono definiti nel file fxml ma che invece vengono definiti a runtime.

Proviamo ad aggiungere un nuovo pulsante alla seconda interfaccia quando viene premuto il pulsante esistente. Il nuovo pulsante dovrà rendere invisibile l'immagine se premuto.

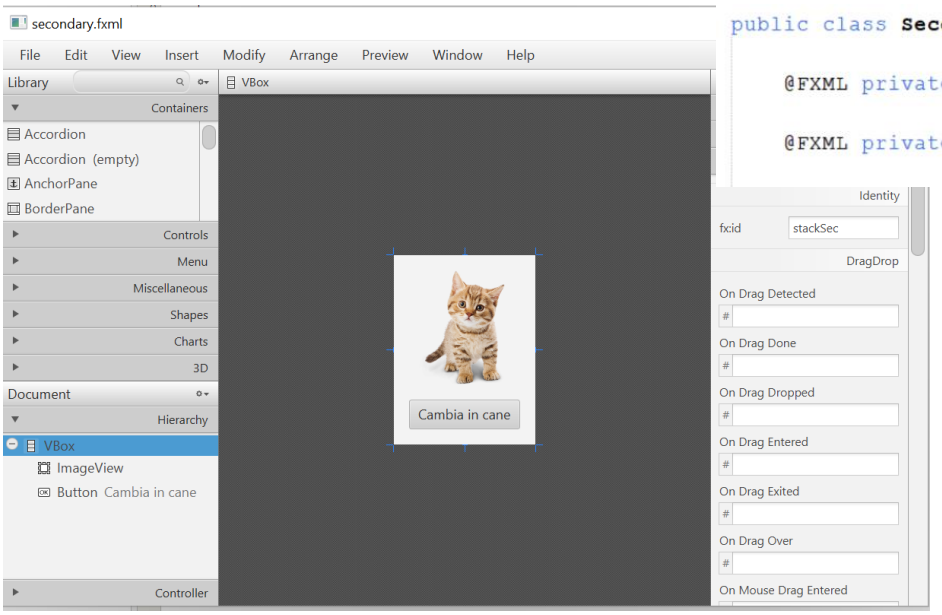
```
Button b = new Button();
b.setText("Nuovo bottone");

// action event
EventHandler<ActionEvent> event = new EventHandler<ActionEvent>() {
    public void handle(ActionEvent e)
    {
        b.setText("Bottone premuto");
        imgCat.setVisible(false);
    }
};

b.setOnAction(event);
stackSec.getChildren().add(b);
```

Il codice contiene:

1. La creazione di un nuovo elemento Button e la configurazione delle sue proprietà
2. La definizione di un nuovo EventHandler con la relativa funzione invocata al momento della pressione dell'utente sul pulsante
3. Il collegamento tra l'handler e il pulsante
4. L'aggiunta del pulsante alla finestra




```
public class SecondaryController {

    @FXML private ImageView imgCat;

    @FXML private VBox stackSec;
```

Per fare quest'ultima operazione abbiamo bisogno di un riferimento alla finestra (non creato di default). Per farlo apriamo l'editor dell'interfaccia e diamo un ID all'elemento VBox che contiene tutti gli elementi della finestra "stackSec" in questo caso e creiamo la variabile corrispondente nella classe di controllo.

Esercizio

Creare dinamicamente un secondo pulsante quando viene premuto il primo pulsante.