श्रवण,मनन,निदिध्यासन

JavaFx Framework

Frontend building platform for Desktop & Mobile

DNYANYOG EDUCATION

Vaibhav Zodge

☎ 7020616260

✉ info.dnyanyog@gmail.com

https://www.dnyanyog.org

https://github.com/zodgevaibhav

JavaFx

“ Open source client application platform for desktop, mobile and embedded systems built on Java ”

JavaFX developed by Sun Microsystems in December 2008 as a modern alternative to Swing

Became popular because of it uses Java programming language

JavaFX provides set of APIs/controls for creating good visual screens including animations, effects, and multimedia

Cross platform support Windows, Linux, MAC & Mobile devices

Supports CSS stylesheet for generic and reusable UI attributes

Work seamlessly Java echo system like Spring, Spring Boot, hibernate etc…

Java Fx Setup

Getting started

https://openjfx.io/openjfx-docs/

Download JavaFx library

https://openjfx.io/

Download JavaFx Scene builder

https://gluonhq.com/products/scene-builder/

Scene Builder & JavaFx SDK

Scene builder helps to create screens using drag and drop kind of tool

Screen built then stored as xml file with special format called “jfxml”

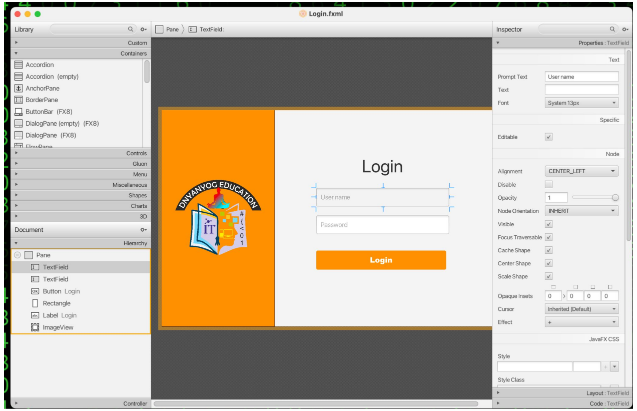
We can load the screen (jfxml) file using JavaFx library (class and methods)

Define buttons and the functions to call on click of those buttons or controllers

JavaFx library have all the required features to run the application

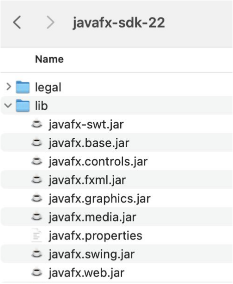
JavaFx library have all the required features to run the application

Scene Builder



Prompt Text: User name
Text:
Font: System 13px
Editable: ☑
Alignment: CENTER_LEFT
Disable: □
Opacity: 1
Node Orientation: INHERIT
Visible: ☑
Focus Traversable: ☑
Cache Shape: ☑
Center Shape: ☑
Scale Shape: ☑
Opaque Insets: 0 0 0 0
Cursor: Inherited (Default)
Effect: +
Style
Style Class

JavaFx SDK

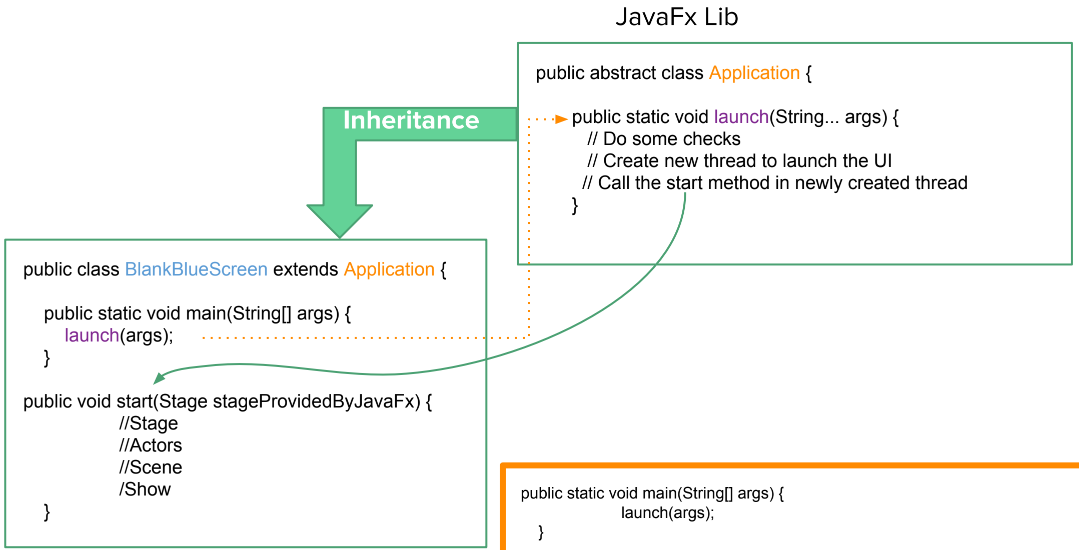


javafx-sdk-22

Name

- legal
- lib
  - javafx-swt.jar
  - javafx.base.jar
  - javafx.controls.jar
  - javafx.fxml.jar
  - javafx.graphics.jar
  - javafx.media.jar
  - javafx.properties
  - javafx.swing.jar
  - javafx.web.jar

Java Fx

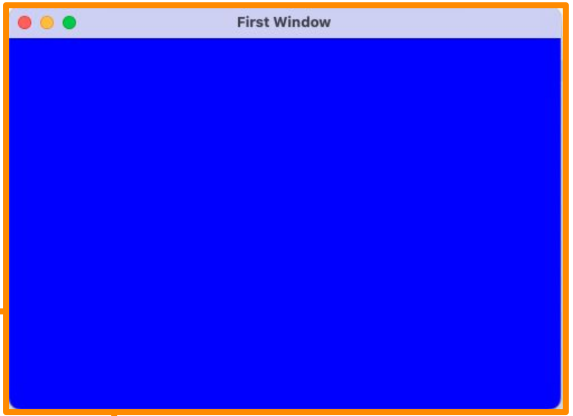


JavaFx Lib

```
public abstract class Application {

    public static void launch(String... args) {
       // Do some checks
       // Create new thread to launch the UI
      // Call the start method in newly created thread
    }
```

```
public class BlankBlueScreen extends Application {

    public static void main(String[] args) {
       launch(args);
    }

public void start(Stage stageProvidedByJavaFx) {
              //Stage
              //Actors
              //Scene
              /Show
    }
```

```
public static void main(String[] args) {
                launch(args);
  }

  @Override // call back function
  public void start(Stage stageProvidedByJavaFx) {
               //Stage stage = new Stage();
               Group actorGroup = new Group(); // Create actorGroup which will perform on the stage

               stageProvidedByJavaFx.setTitle("First Window");  // Arrange the stage
               stageProvidedByJavaFx.setFullScreen(true);
               stageProvidedByJavaFx.setFullScreenExitKeyCombination(KeyCombination.valueOf("q"));

               Scene scene = new Scene(actorGroup, Color.BLUE); // Create Scene using actor and backgrounds

               stageProvidedByJavaFx.setScene(scene);  // Put scene on stage before we show stage to audience
               stageProvidedByJavaFx.show(); // Show the stage to audience

  }
```



Run Configuration

To show window java needs to communicate with Windowing system

Dynamic Libraries are required to communicate with Windowing system

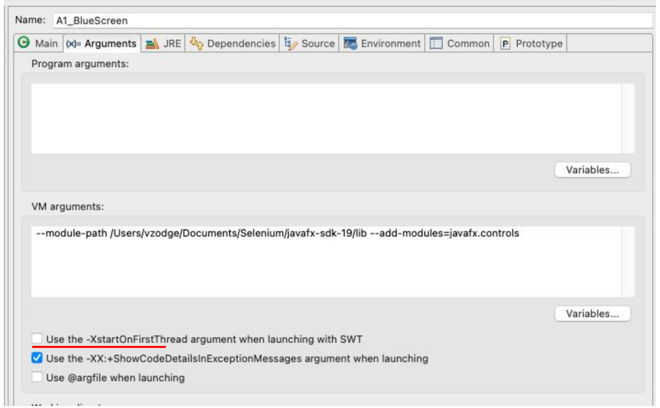
JavaFx provides Dynamic Link Libraries (dyLib-> Mac, DLL-> Windows, SO-> Linux

These DLL path needs to set while running the program along with jar's

VM Argument, give your lib path should be c:\java_development\javafx-sdk-19\lib

--module-path /Users/vzodge/Documents/Selenium/javafx-sdk-19/lib --add-modules=javafx.controls

Provide VM Arguments, uncheck first checkbox and save



Name: A1_BlueScreen

Program arguments:

VM arguments:

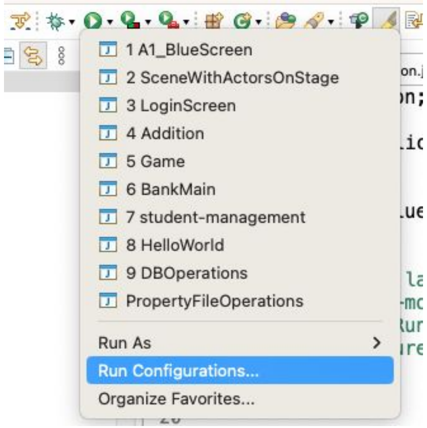
--module-path /Users/vzodge/Documents/Selenium/javafx-sdk-19/lib --add-modules=javafx.controls

□ Use the -XstartOnFirstThread argument when launching with SWT

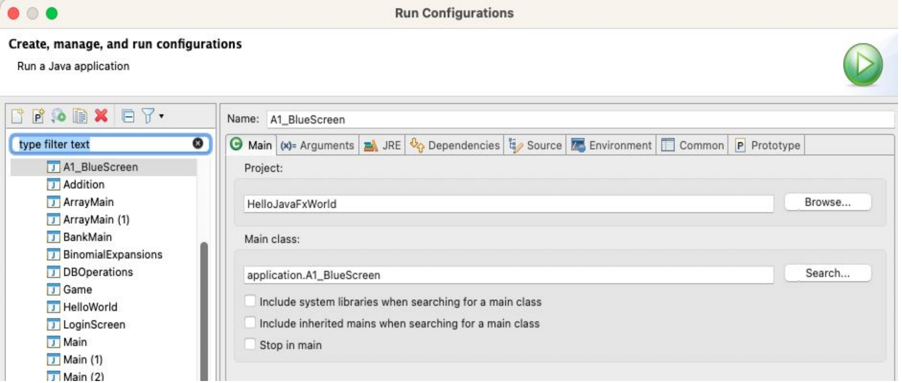
☑ Use the -XX:+ShowCodeDetailsInExceptionMessages argument when launching

□ Use @argfile when launching

Run > Run Configuration



Give Name, Select Project and Main Class



Run Configurations

Create, manage, and run configurations

Run a Java application

Name: A1_BlueScreen

Project:

HelloJavaFxWorld

Main class:

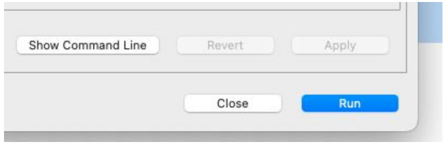
application.A1_BlueScreen

□ Include system libraries when searching for a main class

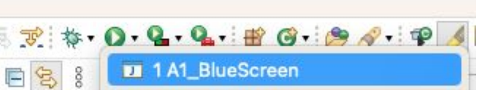
□ Include inherited mains when searching for a main class

□ Stop in main

Click on Run to run the program



Always use this configuration to run program



FXML

Design the UI using scene builder

Export UI to fxml file

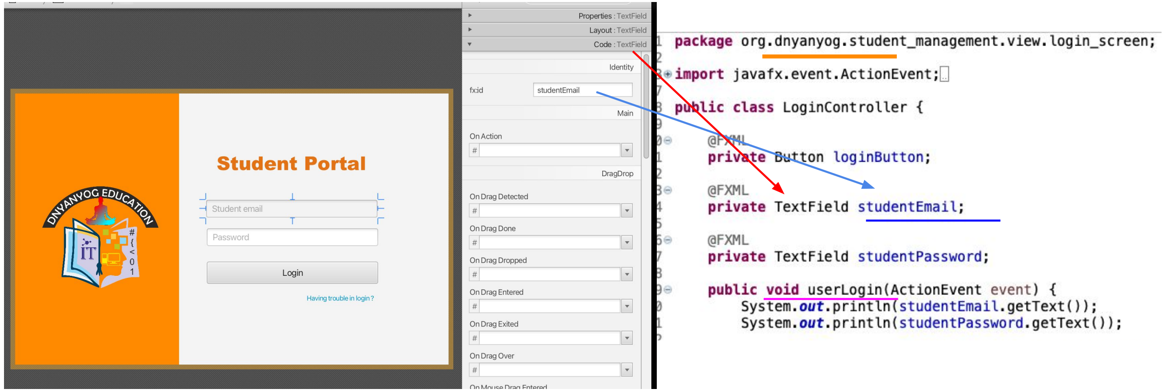
Load the fxml using JavaFx lib in to project

Declare FXML element objects in controller class

Define functions to perform on the event of those elements in controller class

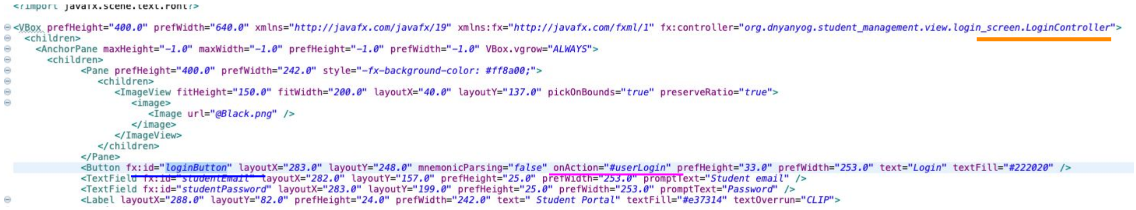
Bind functions from controller class with UI element in fxml file

Launch the main/initial scene from main class



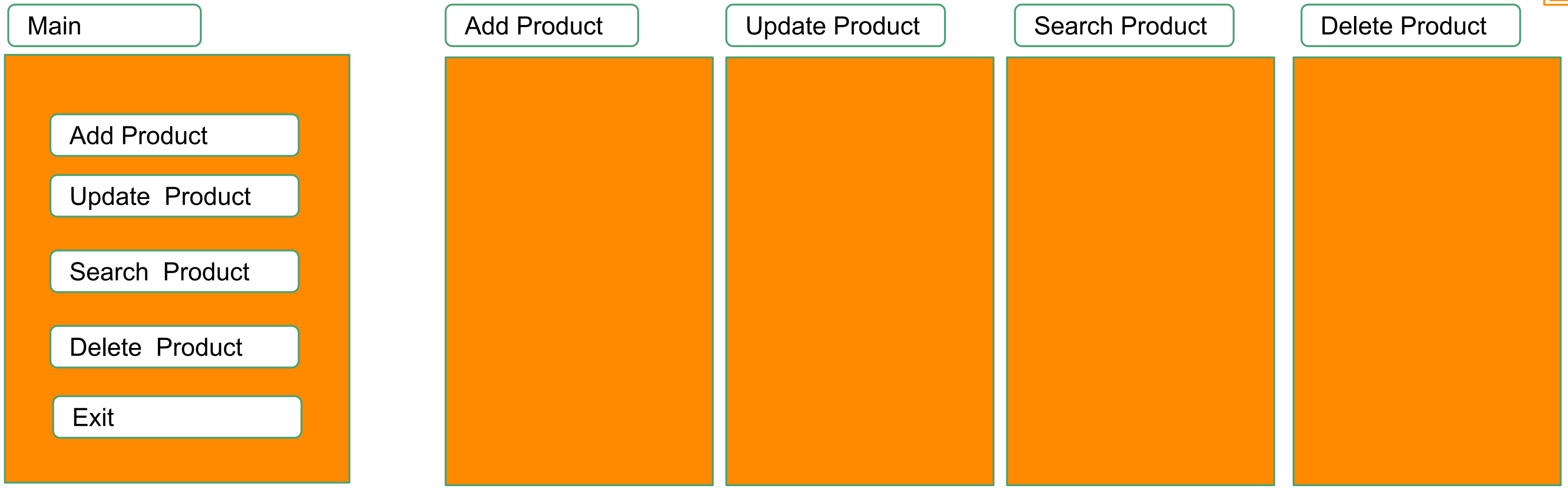
fx:id studentEmail

```java
package org.dnyanyog.student_management.view.login_screen;

import javafx.event.ActionEvent;

public class LoginController {

    @FXML
    private Button loginButton;

    @FXML
    private TextField studentEmail;

    @FXML
    private TextField studentPassword;

    public void userLogin(ActionEvent event) {
        System.out.println(studentEmail.getText());
        System.out.println(studentPassword.getText());
```



```xml
<?import javafx.scene.text.Font?>

<VBox prefHeight="400.0" prefWidth="640.0" xmlns="http://javafx.com/javafx/19" xmlns:fx="http://javafx.com/fxml/1" fx:controller="org.dnyanyog.student_management.view.login_screen.LoginController">
  <children>
    <AnchorPane maxHeight="-1.0" maxWidth="-1.0" prefHeight="-1.0" prefWidth="-1.0" VBox.vgrow="ALWAYS">
      <children>
            <Pane prefHeight="400.0" prefWidth="242.0" style="-fx-background-color: #ff8a00;">
               <children>
                  <ImageView fitHeight="150.0" fitWidth="200.0" layoutX="40.0" layoutY="137.0" pickOnBounds="true" preserveRatio="true">
                     <image>
                        <Image url="@Black.png" />
                     </image>
                  </ImageView>
               </children>
            </Pane>
            <Button fx:id="loginButton" layoutX="283.0" layoutY="248.0" mnemonicParsing="false" onAction="#userLogin" prefHeight="33.0" prefWidth="253.0" text="Login" textFill="#222020" />
            <TextField fx:id="studentEmail" layoutX="282.0" layoutY="157.0" prefHeight="25.0" prefWidth="253.0" promptText="Student email" />
            <TextField fx:id="studentPassword" layoutX="283.0" layoutY="199.0" prefHeight="25.0" prefWidth="253.0" promptText="Password" />
            <Label layoutX="288.0" layoutY="82.0" prefHeight="24.0" prefWidth="242.0" text=" Student Portal" textFill="#e37314" textOverrun="CLIP">
```

Add Product Assignment



Main

Add Product

Update Product

Search Product

Delete Product

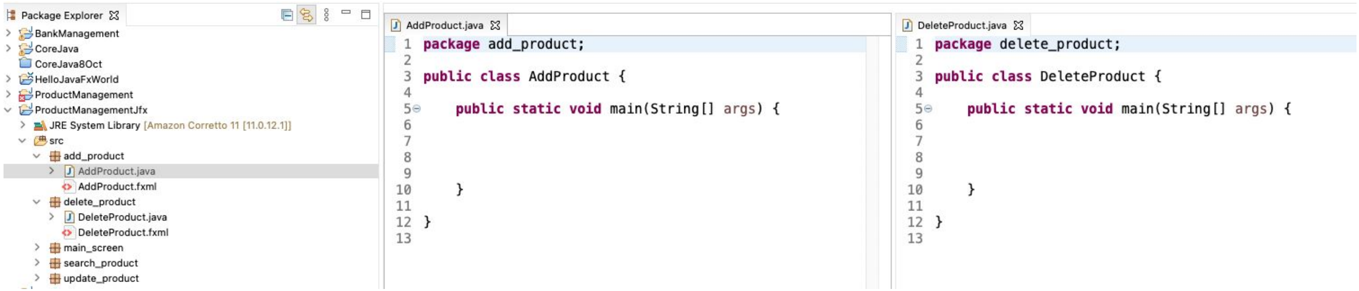
Exit

Add Product

Update Product

Search Product

Delete Product

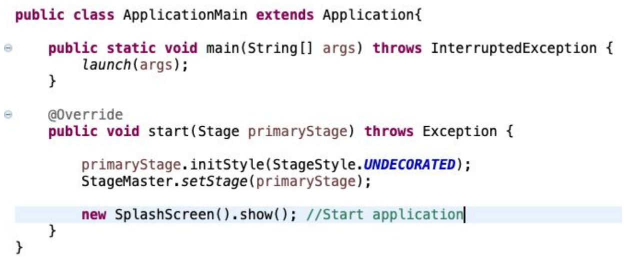


```
> BankManagement
> CoreJava
  CoreJava8Oct
> HelloJavaFxWorld
> ProductManagement
v ProductManagementJfx
  > JRE System Library [Amazon Corretto 11 [11.0.12.1]]
  v src
    v add_product
      > AddProduct.java
        AddProduct.fxml
    v delete_product
      > DeleteProduct.java
        DeleteProduct.fxml
    > main_screen
    > search_product
    > update_product
```

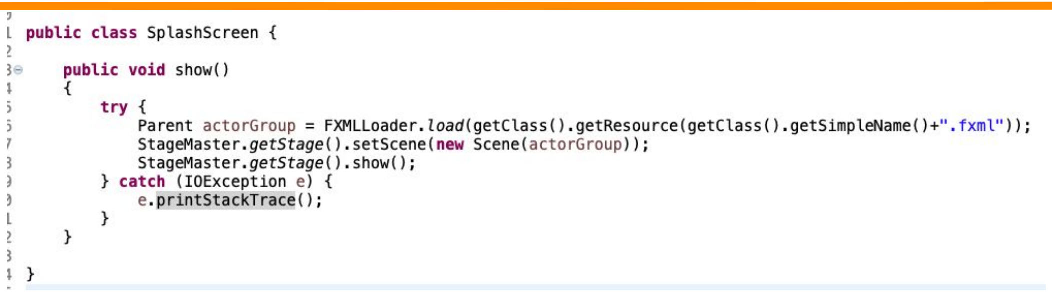
```java
package add_product;

public class AddProduct {

    public static void main(String[] args) {



    }

}

```

```java
package delete_product;

public class DeleteProduct {

    public static void main(String[] args) {



    }

}

```

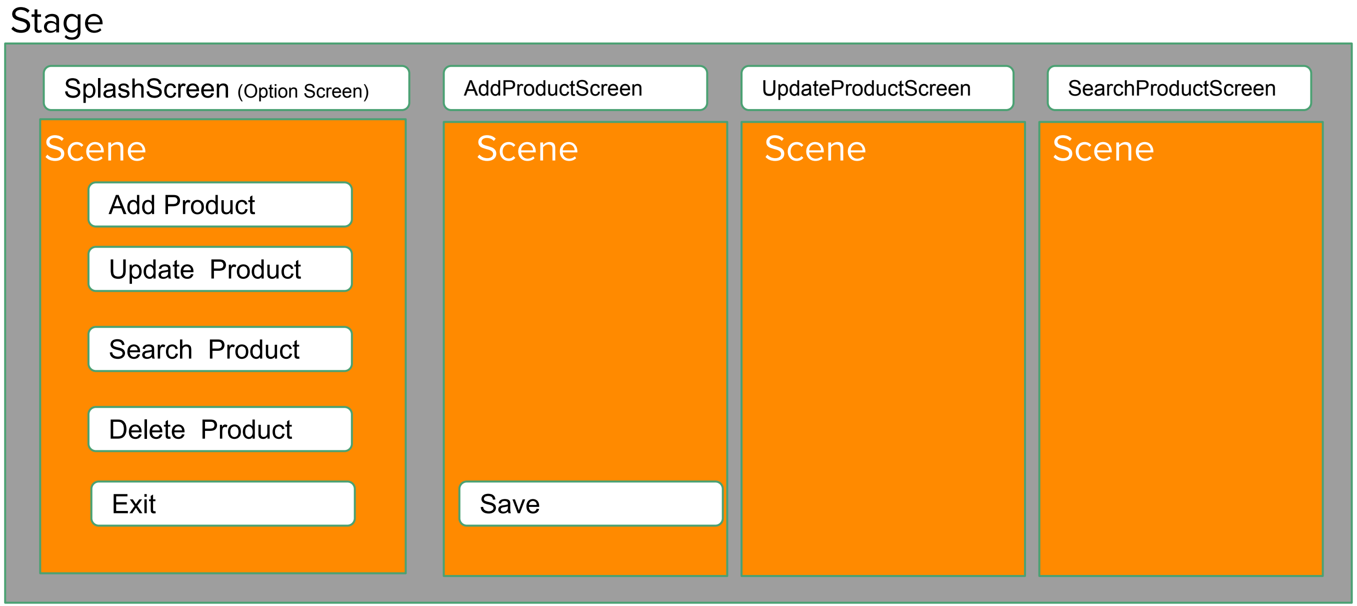
Add Product Assignment

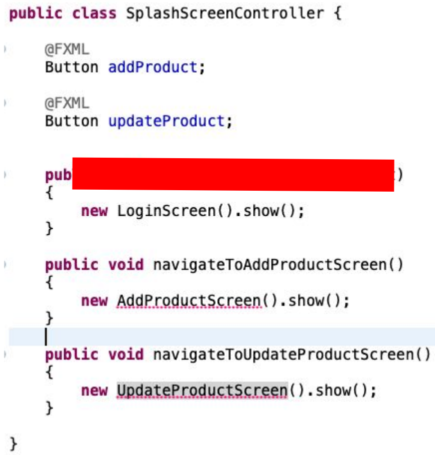


```
public class ApplicationMain extends Application{

    public static void main(String[] args) throws InterruptedException {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) throws Exception {

        primaryStage.initStyle(StageStyle.UNDECORATED);
        StageMaster.setStage(primaryStage);

        new SplashScreen().show(); //Start application
    }
}
```



```
public class SplashScreen {

    public void show()
    {
        try {
            Parent actorGroup = FXMLLoader.load(getClass().getResource(getClass().getSimpleName()+".fxml"));
            StageMaster.getStage().setScene(new Scene(actorGroup));
            StageMaster.getStage().show();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

}
```





```
public class SplashScreenController {

    @FXML
    Button addProduct;

    @FXML
    Button updateProduct;

    pub[redacted])
    {
        new LoginScreen().show();
    }

    public void navigateToAddProductScreen()
    {
        new AddProductScreen().show();
    }

    public void navigateToUpdateProductScreen()
    {
        new UpdateProductScreen().show();
    }

}
```

END