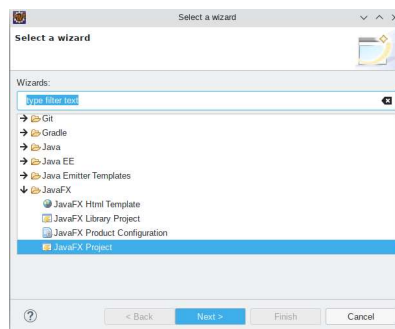
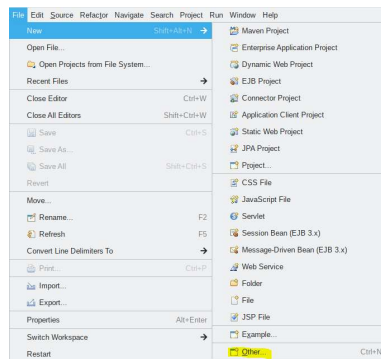


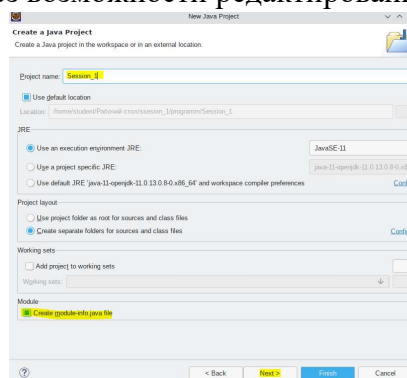
Создание desktop-приложения в IDE Eclipse 2022

Start Styling

1. Создать проект **File>New>Other> JavaFX Project**



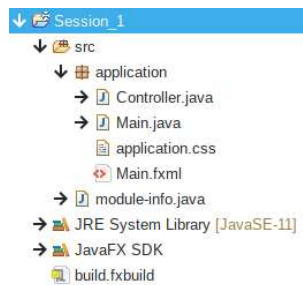
Настроить проект более детально, использовать для этого кнопку **Next**. В противном случае будет создан проект по умолчанию без возможности редактирования в **Scene Builder**.



Обязательно выбрать язык **FXML**, файл с данным расширением будет отвечать за графический интерфейс будущего приложения.



Созданный проект будет иметь следующую структуру:

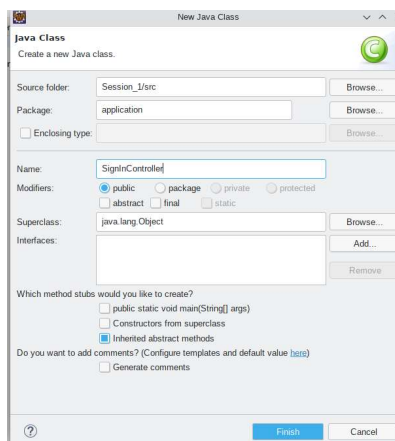
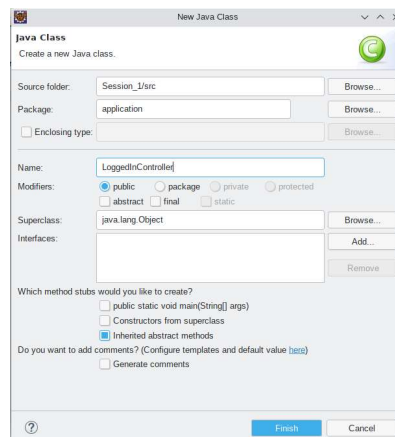


Java-файл **Controller.java** отвечает за виджеты и события, которые хотим назначить, например, на кнопки в desktop-приложении.

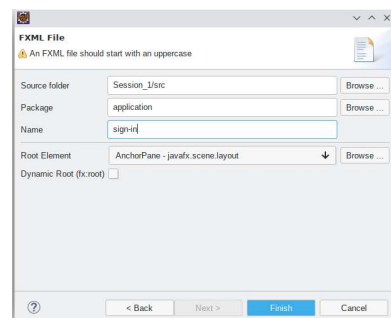
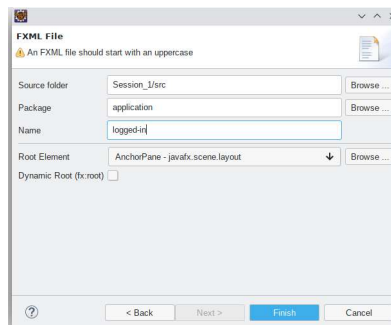
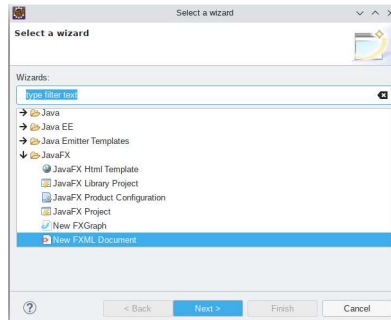
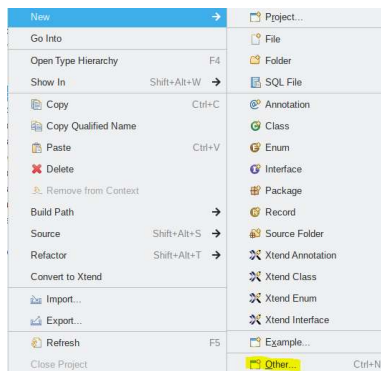
Java-файл **Main.java** точка входа приложения.

Файл **Main.fxml** – это файл инструкций для формирования объектов графики, его открываем через Scene Builder.

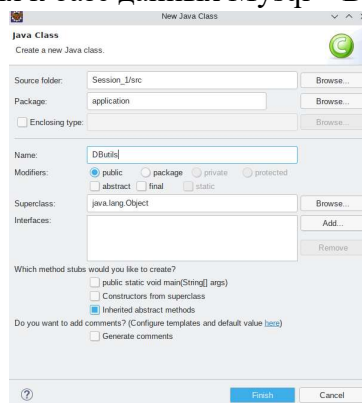
Добавим два контроллера: **LoggedInController.java**(авторизация) и **SignInController.java** (регистрация). Выбираем пакет, в котором будем создавать классы: **New> Class**



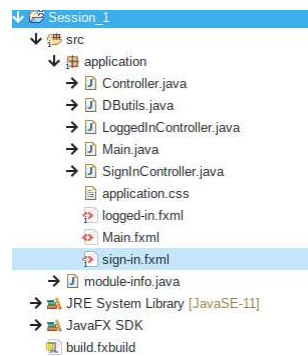
Добавим два **.fxml** файла соответственно: **logged-in.fxml** и **sign-in.fxml**



Создадим класс для подключения к базе данных Mysql – **DButils.java**.



Рабочий каталог примет вид:



Свяжем **fxml-файлы** с контроллерами соответственно.

logged-in.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.layout.BorderPane?>

<BorderPane xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.LoggedInController">
    <!-- TODO Add Nodes -->
</BorderPane>
```

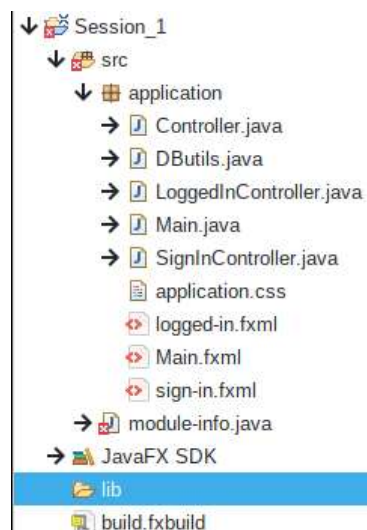
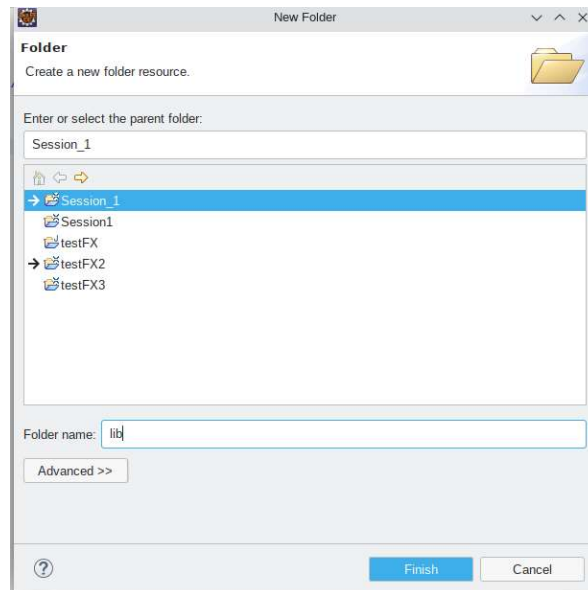
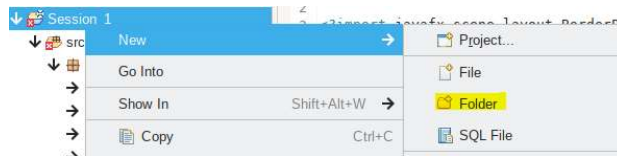


sign-in.fxml

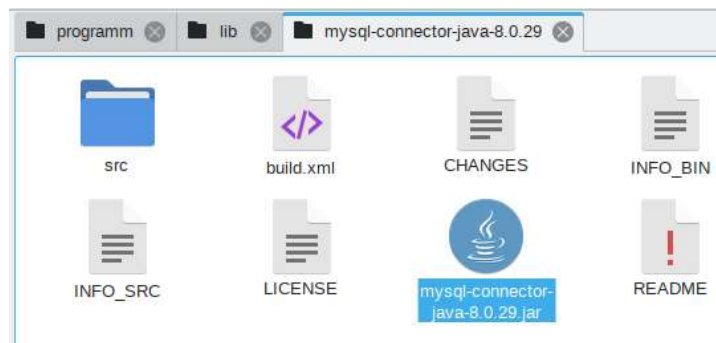
```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.layout.BorderPane?>
<BorderPane xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.SignInController">
    <!-- TODO Add Nodes -->
</BorderPane>
```

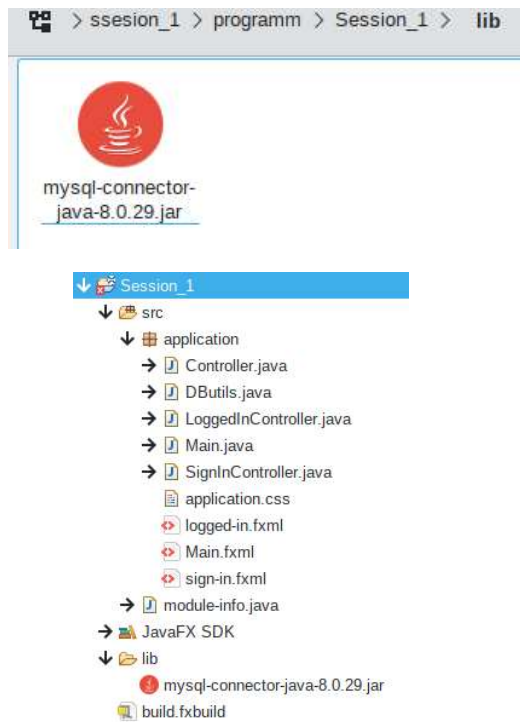


Подключим внешнюю библиотеку JDBC для соединения с БД .
Создадим каталог **lib** и перенесем все **.jar- файлы** **Mysql connector J**.

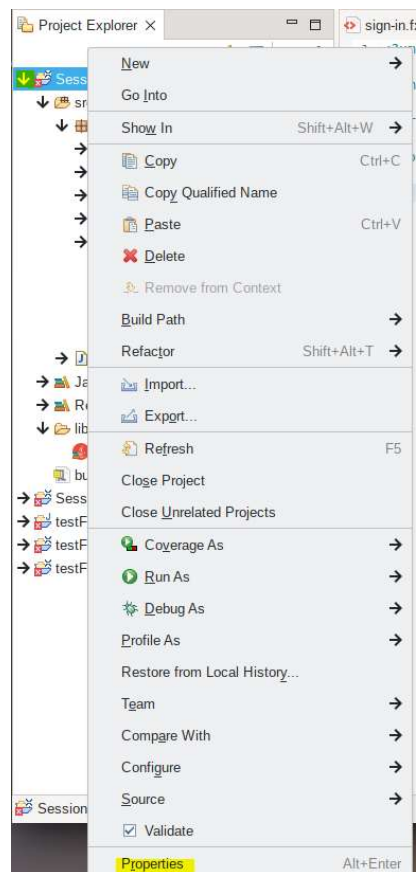


Переносим драйвер JDBC в lib:

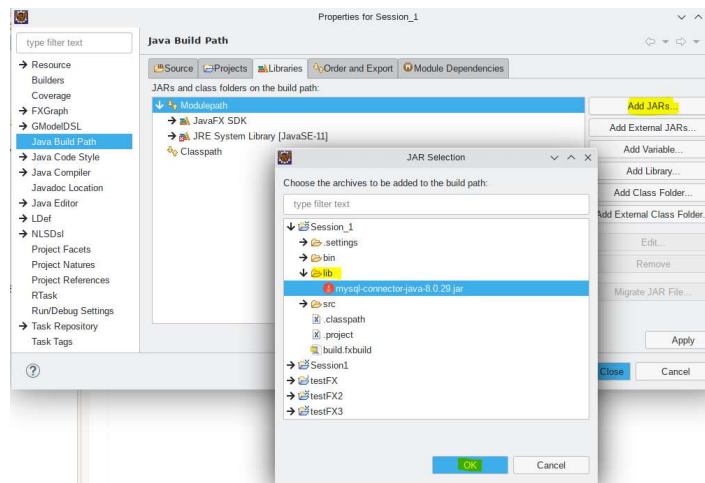




Подключим jar-файл к проекту. Переходим в свойства проекта.

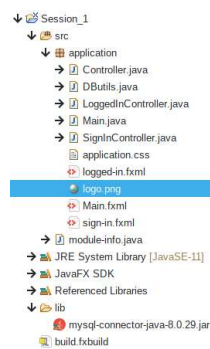


Переходим в **Libraries**, выбираем пункт **Modulepath**, добавляем файл из **Add JARs**. Указываем ранее загруженный драйвер в каталоге **lib**.

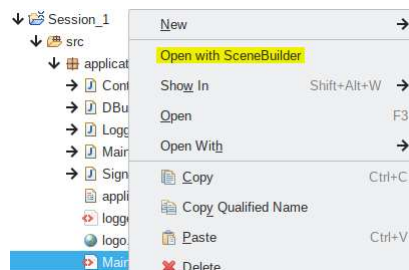


Иногда требуется перезагрузить **Eclipse** при добавлении **JAR-файлов** в проект.

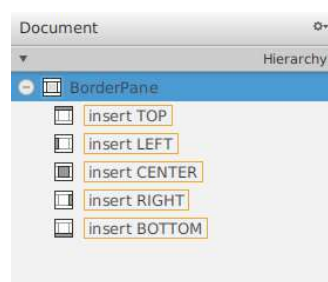
Добавляем в проект **Logo**. Копируем файл в корень проекта.



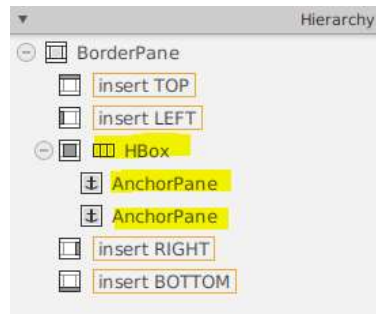
Открываем для редактирования файл **Main.java** через программу **SceneBuilder**.



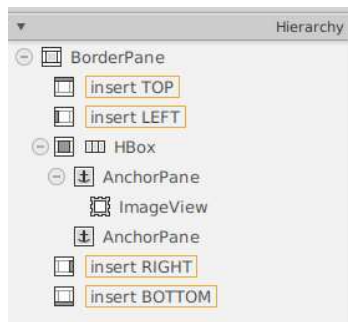
Проверяем стандартную иерархию.



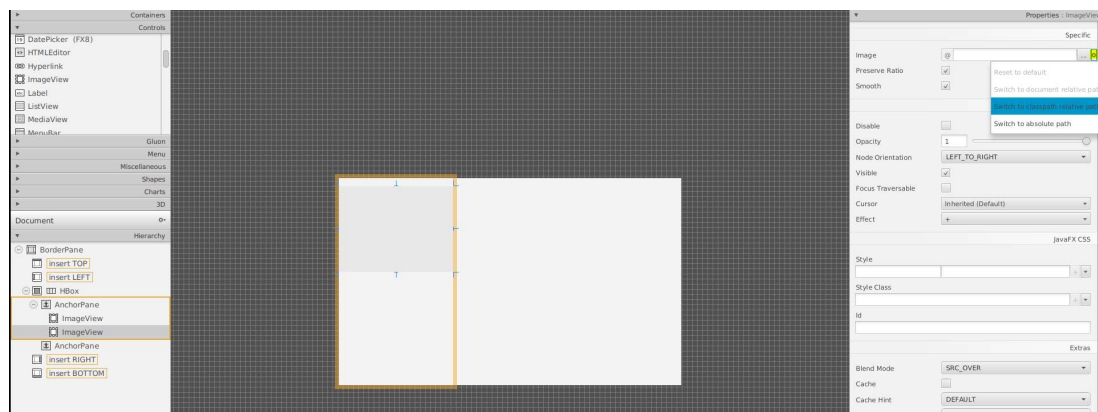
Добавляем из **containers** в иерархию, как показано ниже, следующие объекты:



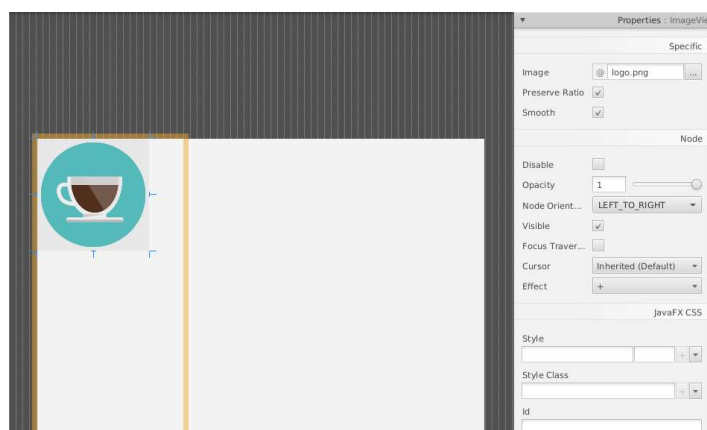
В **controls** выбираем компонент **ImageView**:



Устанавливаем лого проекта, предварительно определив локальный каталог лого.



Укажем название файла с расширением, в нашем случае **logo.png**.



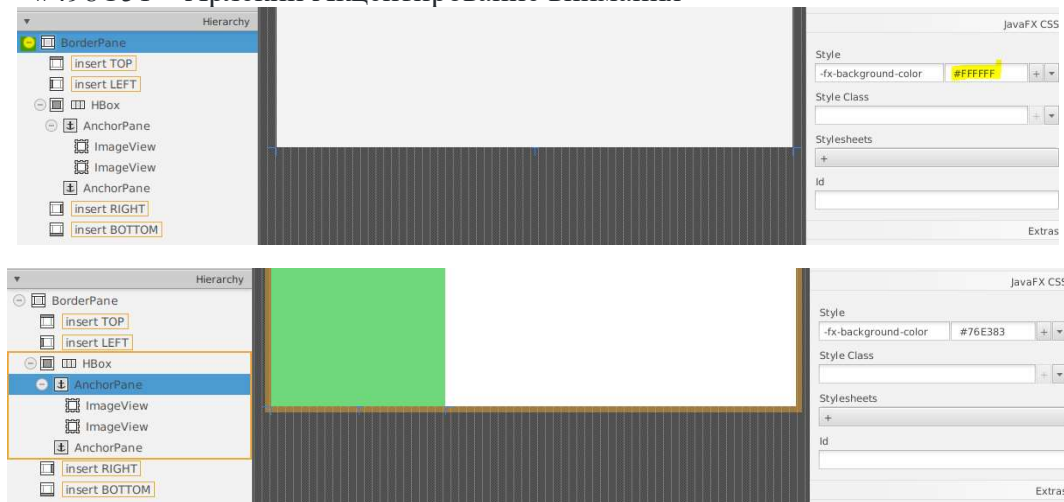
Определяем цветовую палитру проекта согласно **Руководство по стилю.pdf**.

<https://get-color.ru/#!498C51>

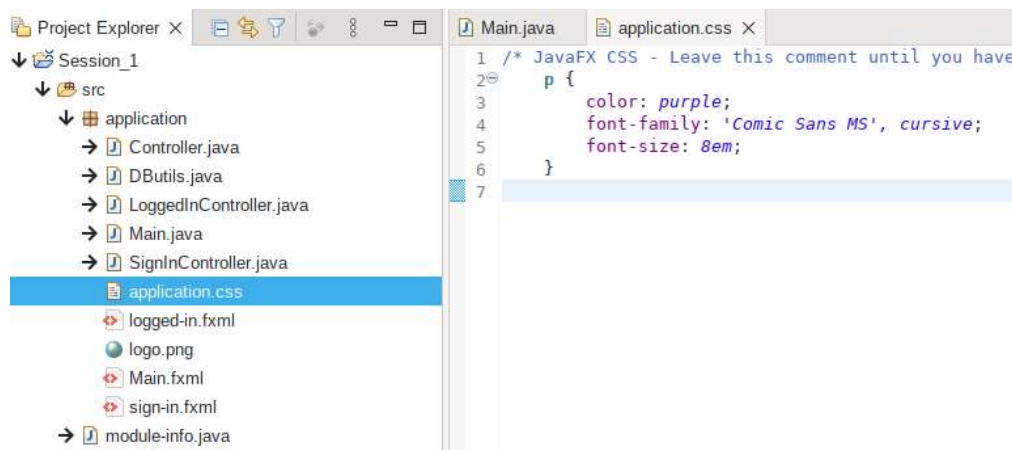
(255, 255, 255) #FFFFFF – белый Основной фон

(118, 227, 131) #76E383 - Пастельно-зеленый Дополнительный фон

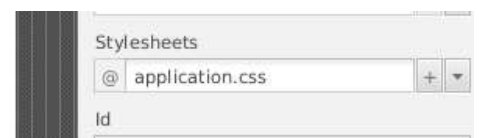
(73, 140,81) #498C51 – Арлекин Акцентирование внимания



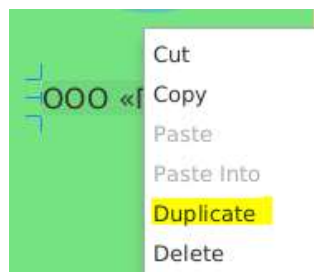
Используйте шрифт Comic Sans MS указано в руководстве, однако в Linux нет такого шрифта. Создать стиль в файле **application.css**. Дать пояснения в **Readme.md**.



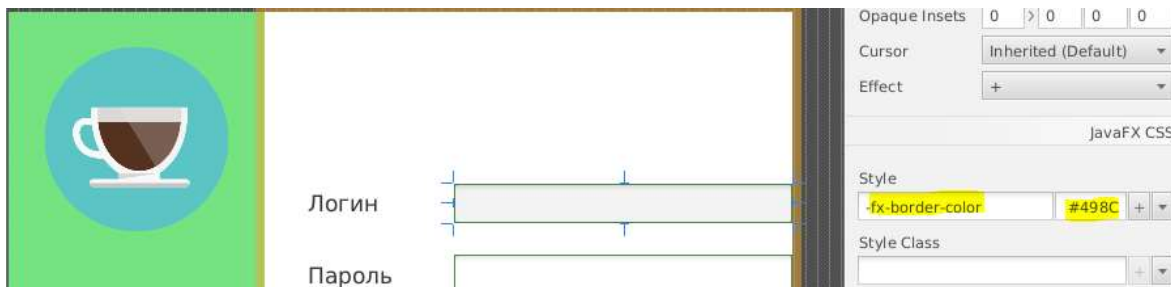
В **SceneBuilder** подключить стилевой файл. Добавляем **Label**.



Используем копирование созданной метки для реплицирования одинакового стиля по всей форме



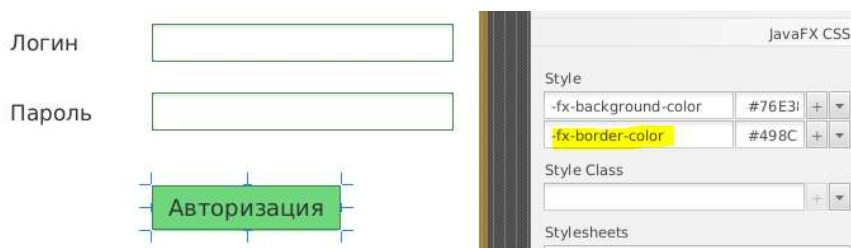
Добавим метки Логин и Пароль. А также из раздела **controls** два **TextFields**.



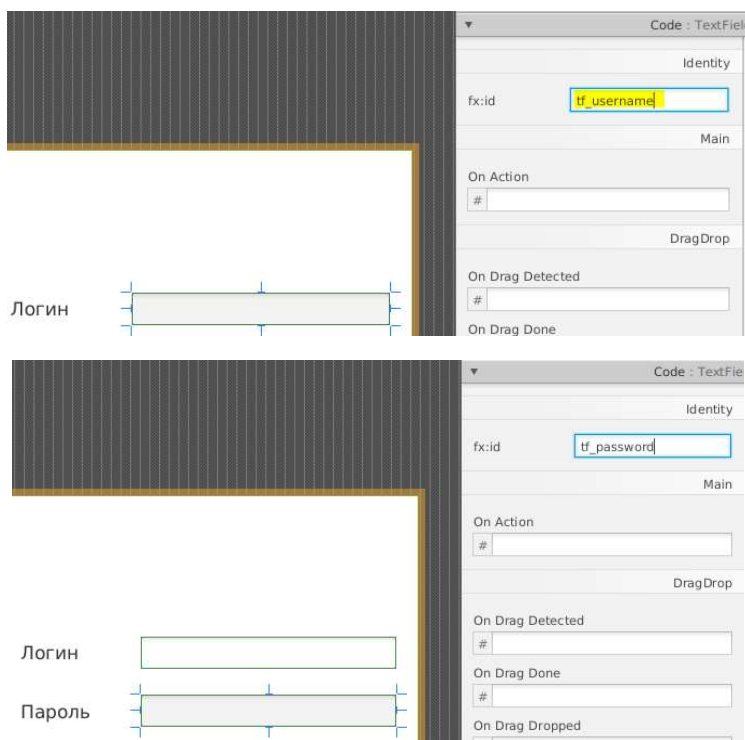
Добавляем из раздела **controls** **Button**.



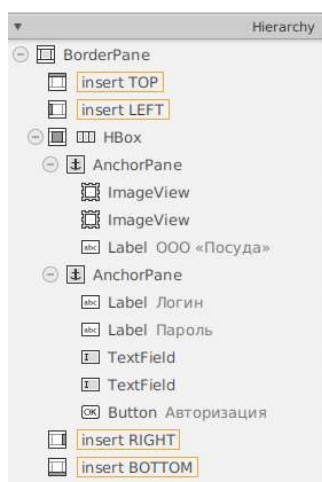
Акцентирование внимания (темный зеленый) делаем на границе объектов формы:



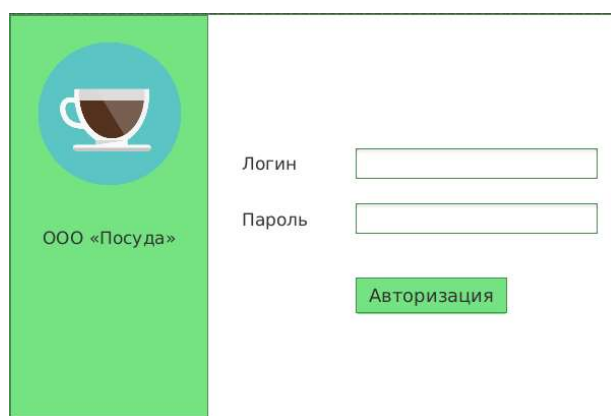
Формируем **id** полей, которые потребуются для авторизации



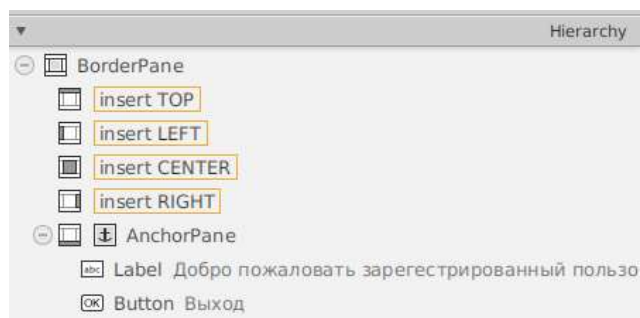
Конечная иерархия формы примет вид:



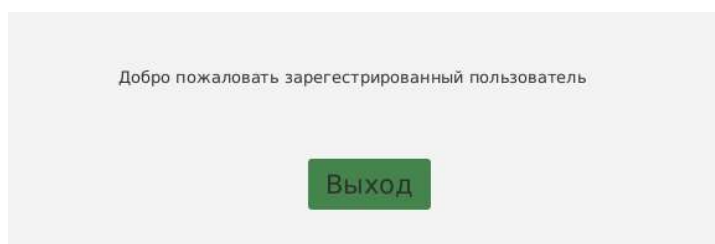
Макет:



Реализуем в Scene Builder для формы **logged-in.fxml** следующую иерархию



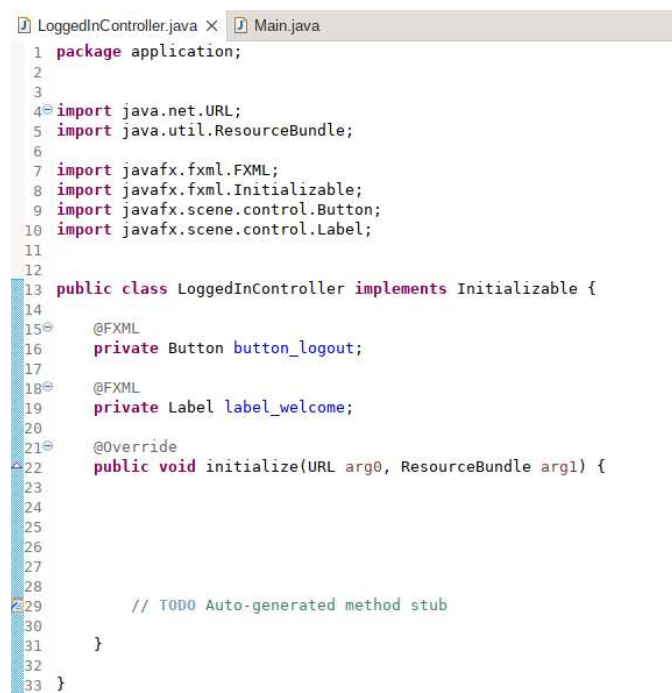
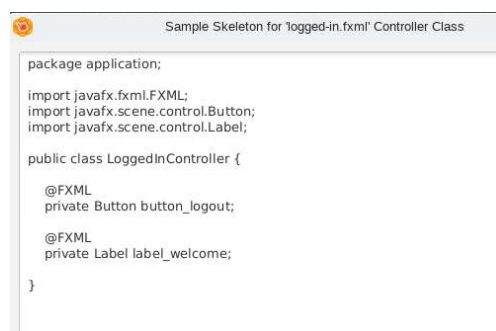
Макет:



Реализуем контроллеры действия на события. Получим представление FXML:



Копируем его в **LoggedInController.java**



Start Coding