**Opgave 8: UI and JavaFX**

**Opgave 8.1: Kleine desktopapplicatie bouwen in SceneBuilder**

Bestudeer onderstaande screenshot van een login scherm.

Probeer dit scherm na te maken in SceneBuilder.

**Login scherm**



1. Maak binnen IntelliJ een JavaFX-project aan en bereid het project verder voor op JavaFX.
   1. Maak een nieuw project aan: File > New > Project > JavaFX.  
      IntelliJ maakt meteen de volgende voorbeeldklassen aan: Main.java, Controller.java, sample.fxml.
   2. Bereid het werken met JavaFX en Scene Builder verder voor. Volg hiervoor de “Installatiehandleiding JavaFX” van V1OOP op Canvas:
      1. Scene Builder koppelen aan IntelliJ
      2. JavaFX libraries toevoegen aan je project
      3. De JavaFX modules koppelen aan je project
2. Open het sample.fxml bestand met Scene Builder: rechter muisklik > Open in SceneBuilder.
3. Vervang het GridPane door een AnchorPane.
   1. Selecteer binnen het Hierarchy panel de GridPane > rechter muisklik > Delete.
   2. Selecteer AnchorPane (binnen Library > Containers) en sleep het naar het tekengebied.
   3. Wijzig het formaat van het AnchorPane (Inspector > Layout): Pref Width = 500 en Pref Height = 300.
4. Voeg de schermonderdelen toe aan het AnchorPane.
   1. In dit scherm komen de volgende typen GUI-controls voor: Button, Label, TextField.
   2. Selecteer telkens een control (Library > Controls) en sleep het naar de juiste plek op het scherm.
   3. Wijzig eigenschappen van een control binnen het Inspector panel.
      1. Properties: Tekst, Font, Tekst Fill.
5. Save het scherm binnen Scene Builder.
6. Run het scherm.
   1. Selecteer en open Main.java. Bestudeer de code.
      1. Wijzig de titel die gezet wordt van “Hello World” in “Welkom”. En wijzig “300, 275” in “500, 300”.
      2. Save.
   2. Selecteer Main.java en Run.
      1. Als het scherm niet goed opstart, heb je waarschijnlijk de “Installatiehandleiding JavaFX” niet goed gevolgd.

**Opgave 8.2 Complexe desktopapplicatie bouwen van de use case “Registreer Luchthaven”**

Maak een scherm in SceneBuilder van de use case “Registreer Luchthaven”.

* Deze opdracht moet je ook maken voor inleveropdracht 2 van OOAD1.

Het schermontwerp is hieronder afgebeeld.

**Schermontwerp van de use case “Registreer Luchthaven”**



1. Installeer het LuchtvaartMaatschappijSysteem
   1. Download het LuchtvaartMaatschappijSysteem en de bijbehorende installatiehandleiding van de V1OOAD site op Canvas, Module LuchtvaartMaatschappijSysteem.
   2. Volg de installatiehandleiding totdat je het systeem kan runnen.
   3. Selecteer menuoptie 4 en bekijk welke luchthavens er zijn en wat hun kenmerken zijn.
   4. Selecteer menuoptie 3. Er verschijnt een leeg scherm voor Registreer Luchthaven.  
      Dat moet jij nu verder invullen en werkend krijgen.
2. Open main.userInterfaceLaag.RegistreerLuchthaven.fxml met Scene Builder.
3. Ontwerp de structuur van het scherm in JavaFX-componenten.  
   Schets eerst welke containers je nodig hebt in welke volgorde. Werk hierbij van buiten naar binnen.  
   Bepaal daarna welke controls je nodig hebt binnen de containers.   
   Enkele tips:
   1. Bij alle andere schermen van dit desktop-systeem is een AnchorPane als basis-pane gebruikt.
   2. Het TitledPane is geschikt om te gebruiken voor het panel met titel “Nieuwe Luchthaven”.
   3. Met een HBox of VBox kan je controls mooi uitlijnen. Het type uitlijning is instelbaar (Inspector > Properties > Alignment). Als je meerdere controls binnen een box hebt, kan je de afstand daartussen instellen als marge van een geselecteerde control (Inspector > Layout > Margin).
   4. Laat het ontwerp controleren door de docent. Dat kan veel werk schelen.
4. Realiseer het ontwerp met SceneBuilder.  
   Run het scherm regelmatig om te controleren of jouw scherm wel consistent blijft met het meegegeven ontwerp. Zodra het getoonde scherm overeenkomt met het schermontwerp, ben je klaar met de layout.

Optioneel: Het compleet werkend krijgen van een use case valt eigenlijk buiten de scope van OOAD en binnen de scope van OOP.

1. Maak het scherm werkend op basis van de al uitgewerkte controller-klasse en het domein.  
   Hiervoor moet je de nodige koppelingen maken tussen het scherm en de bijbehorende controller.   
   De eerste koppeling is al gelegd: In RegistreerLuchthaven.fxml staat al gedeclareerd dat RegistreerLuchthavenController.java de bijbehorende controller is.   
     
   Ga verder als volgt te werk (pas de leerstof uit de sheets toe):
   1. Maak koppelingen voor de GUI-componenten.
      1. Bekijk in de controller code welke GUI-componenten daar als attributen gedeclareerd zijn (met @FXML erboven), zoals: ComboBox landComboBox;
      2. Neem de naam van ieder GUI-component-attribuut in het controller-bestand op als “fx:id” bij de overeenkomstige component in het fxml-bestand.
         1. Selecteer de component in Scene Builder 🡪 Inspector > Code > fx:id.
      3. Voor ToggleGroup “werkplaats” werkt het anders. Die wordt gebruikt om de twee RadioButtons te groeperen. Selecteer een RadioButton in het fxml-bestand en voer “werkplaats” in bij ToggleGroup (Inspector > Properties). Doe hetzelfde bij de tweede RadioButton.
      4. Save het fxml-bestand.   
         Als het goed is, staat er daarna in het controller-bestand voor ieder @FXML attribuut een symbool in de kantlijn dat aangeeft dat de GUI-component in het het fxml-bestand is gevonden. Zo niet, los dat dan eerst op.
   2. Activeer de code van initialize() binnen de controller-klasse en test de werking.
      1. Delete binnen initialize() de volgende regels: “/\*\* Activeer na het maken van het scherm”   
         en “\*/ “. Save de controller-klasse daarna.
      2. Test de use case en controleer of de landComboBox nu landen toont na de initialisatie.
   3. Maak koppelingen voor event-afhandeling.
      1. Bestudeer de publieke methoden in de controller-klasse. Ze staan in dezelfde volgorde waarin de GUI-componenten staan die events kunnen afvuren.
      2. **Als er een event wordt afgevuurd door de actor op een GUI-component moet er iets gebeuren.** Hiervoor zijn koppelingen nodig tussen de GUI-component-definities en de controller-klasse. Maak de volgende koppelingen door in Scene Builder de GUI-component te selecteren en bij een event (Inspector > Code) de naam van een methode van de controller-klasse te selecteren.

|  |  |  |
| --- | --- | --- |
| **GUI-component** | **Event** | **Methode** |
| landComboBox | On Action | landNaam |
| jaRadioButton | On Action | werkplaats |
| neeRadioButton | On Action | werkplaats |
| buttonOK | On Action | ok |
| buttonCancel | On Action | cancel |

* 1. Save het fxml-bestand.   
     En test de use case. Nu moet het scherm compleet werken.  
     Controleer met menu-optie 4 of een nieuwe luchthaven ook werkelijk is toegevoegd.

NB Bij de tekstinvoervelden heb je geen event hoeven te activeren. Dit komt omdat een TextField beter werkt als een Focus-event worden gebruikt. Deze ondersteunt Scene Builder niet, dus de code hiervoor moet handmatig in de contraoller-klasse worden opgenomen. Zie hiervoor de code binnen initialize() en ook de code binnen intialize() van RegistreerVliegtuigController. Deze code hoef je niet echt te snappen, maar je kan die kopiëren en aanpassen voor use cases binnen het groepsproject.