

LOTTO

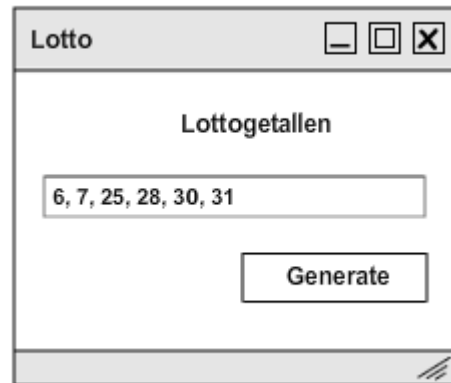
We bouwen een eenvoudige lottogetallen-app. Hiernaast staat de wireframe van de applicatie. Deze is gemaakt met de online tool mockflow.com.

De model-klasse (GetallenGenerator.java – LEERSTOF 2^{de} jaar) is al volledig klaar, de **Main** klasse gedeeltelijk. We moeten enkel de *view* en de *presenter* nog uitwerken.

De voornaamste JavaFX klassen die we voor deze oefening nodig hebben zijn:

- [`javafx.scene.layout.BorderPane`](#)
- [`BorderPane.setTop`](#) en andere
- [`BorderPane.setAlignment`](#)

Raadpleeg in eerste instantie de [JavaFX documentatie](#) als je ergens vast zit!



Figuur 1: wireframe (mockflow.com)

1 WIREFRAME

Bouw de gegeven wireframe na in een tool naar keuze.

2 HOOFDSCHERM AANMAKEN – MVP

De **Main** klasse is al aangemaakt. Bestudeer de code. In de **start**-methode worden het model en de view via een presenter gekoppeld.

- Voeg de view toe in een **Scene** op de **primaryStage**.
- Zorg ervoor dat de app als titel "Lotto" krijgt
- Toon de **primaryStage**

3 UI OPBOUWEN

Hier werken we de klasse **LottoView** uit.

We hebben drie onderdelen nodig: Een **Label**, een **TextField** en een **Button**. Maak ze aan als attributen.

3.1 De methode `initialiseNodes`

Schrijf een nieuwe methode genaamd `initialiseNodes`. Roep deze methode op in de constructor.

In deze methode:

- Voorzie de drie attributen van een waarde.
- Het tekstveld moet een geprefereerde aantal kolommen 10 hebben en het mag niet editeerbaar zijn.

3.2 De methode `layoutNodes`

We plaatsen de controls op de `BorderPane` (dit is de klasse `LottoView` zelf).

Schrijf een nieuwe methode genaamd `layoutNodes`. Roep deze methode op in de constructor.

In de methode `layoutNodes`:

- Maak gebruik van de juiste `set...` methode van de klasse `BorderPane` om de componenten op de juiste plaats te krijgen.
- Zorg voor een horizontale en verticale "marge" van 10 pixels tussen de cellen van de `BorderPane`. Zorg ook voor een "padding" van 10 pixels.

Zoek in de documentatie van [BorderPane](#) naar de juiste methodes!

4 AFHANDELEN EVENTS

We zijn geïnteresseerd in het action event op de knop. Implementatie van de eventhandlers gebeurt in de `Presenter` klasse.

4.1 De methode `handleEvents`

De private methode `handleEvents` is aangemaakt in de klasse `Presenter` en wordt aangeroepen via de constructor. Je hoeft deze methode enkel nog in te vullen.

- Hang een eventhandler aan de knop. Om dit te kunnen doen moet je ervoor zorgen dat de button bereikbaar is via een getter die je aan de klasse `LottoView` toevoegt (package-private).
- Zorg er voor dat de eventhandler de methode `updateView` (zie punt 4.2) oproept.

4.2 De methode `updateView`

Deze methode moet de juiste informatie uit het model ophalen en vervolgens de view up-to-date brengen met deze informatie.

- Bij het klikken op de "Generate" Button moet je de view aanpassen door uit het model nieuwe lottogetallen op te halen. Gebruik hiervoor de methode `getLottoGetallen` uit de klasse `GetallenGenerator` (het model).
- Zorg ervoor dat het `TextField` via een getter in de klasse `LottoView` bereikbaar is om de nieuwe getallen in te vullen.