Background

We bouwen een eenvoudige app om enkele achtergrondkleuren te tonen. Hiernaast staat de wireframe van de applicatie. Deze is gemaakt met de online tool mockflow.com.

De model-klasse **Kleuren**.java 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**](https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/GridPane.html)

Figuur : wireframe (mockflow.com)

* **javafx.geometry.Insets;  
  javafx.geometry.Pos;**

Raadpleeg in eerste instantie de [JavaFX documentatie](https://docs.oracle.com/javase/8/javafx/api/toc.htm) als je ergens vast zit!

# Wireframe

Bouw de gegeven wireframe na in een tool naar keuze.

# 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 “*Colour*” krijgt
* Toon de **primaryStage**, zet de afmetingenmet **setWidth** en **setHeight** oprespectievelijk 400 en 245 pixels

# UI Opbouwen

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

We hebben slechts 1 onderdeel nodig: Een **Button**. Maak hem aan als attribuut.

## De methode initialiseNodes

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

In deze methode:

* Voorzie het attribuut van een waarde.

## De methode layoutNodes

We plaatsen de controls op de **BorderPane** (dit is de klasse **View** 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 component op de juiste plaats te krijgen.
* Zorg voor een “padding” van 10 pixels.

Zoek in de documentatie van [**BorderPane**](https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/GridPane.html) naar de juiste methodes!

# Afhandelen events

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

## 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 **View** toevoegt (package-private).
* Zorg er voor dat de eventhandler de methode **updateView** (zie punt 4.2) oproept.

## 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 "Change Colour" **Button** moet je de achtergrondkleur van de view aanpassen door uit het model een bepaalde kleur op te halen. Via de **setStyle** methode die je op de view toepast kan je de kleur aanpassen.
* In feite passen we CSS toe (en zijn er betere manieren om dit te doen), zie later.