



Gr. 1, Dr. E. Pitzer



Gr. 2, DI F. Gruber-Leitner

Name Robert Wurm Aufwand in h 20Punkte \_\_\_\_\_ Kurzzeichen Tutor / Übungsleiter \_\_\_\_\_ / \_\_\_\_\_

---

Wie angekündigt wollen wir in den restlichen Übungen – also von der vorliegenden 6. bis zur 8. Übung – anhand *einer* Aufgabenstellung verschiedene Möglichkeiten der Java-Klassenbibliothek ausloten. Damit Sie die weiteren Übungen meistern können, sollten Sie von Anfang an mitmachen.

## EURO-Bet – Die SWE4-Wettanwendung zur Fußball-EM

### Kurzbeschreibung

Der FH Hagenberg möchte die bevorstehende Fußball-EM für seine Studierenden und Mitarbeiter besonders spannend machen und eine interne Wett-Plattform ins Leben rufen. Teilnehmer sollen vor – aber auch während – der Spiele Tipps über den Gewinner abgeben können und bekommen (bei richtigen Tipps) nach einem ausgeklügelten System Punkte gutgeschrieben.

Dafür wird die Erstellung einer Software in Auftrag gegeben, die im Wesentlichen folgende Anforderungen erfüllen soll:

### Verwaltungstool

- Eintragung von Mannschaften
- Eintragung von Spielpaarungen
- Eintragung von Spielergebnissen
- Benutzerverwaltung (eintragen, ändern, sperren)

### Wett-Plattform

- Login mit Benutzername und Passwort
- Anzeige der vergangenen und zukünftigen Spiele
- Abgabe von Tipps

### Verwaltungstool

Das Verwaltungstool steht dem Betreiber der Wettplattform zur Verfügung, um die Mannschaften, Spiele und Ergebnisse zu warten bzw. die abgegebenen Tipps zu analysieren.

Ein Spiel besteht aus den beiden Mannschaften, dem Spielort, dem Datum und der Uhrzeit des Spielbeginns. Es soll möglich sein, Spiele bereits vorab einzutragen (noch bevor das Ergebnis bekannt ist). Nach Ablauf des Spiels wird auch das Spielergebnis festgehalten.

Alle Wett-Teilnehmer müssen sich beim Plattformbetreiber persönlich melden. Dieser kann dann im Verwaltungstool neue Benutzer anlegen (Vorname, Nachname, Benutzername, Passwort). Wenn ein Benutzer des Wettbetrugs verdächtig wird, so möchte der Betreiber den Benutzer auch wieder sperren können.

## Online-Wettplattform

Nach der Anmeldung mit den erhaltenen Zugangsdaten erscheint eine Übersicht aller Spiele, bei bereits abgelaufenen Spielen ist auch der Endstand, der ev. abgegebene Tipp des Benutzers und die dadurch erreichte Punktezahl sichtbar. Bei noch nicht abgelaufenen Spielen kann der Benutzer einen Tipp abgeben:

- Mannschaft 1 gewinnt
- Mannschaft 2 gewinnt
- Unentschieden

Der Tipp kann beliebig oft verändert werden, allerdings nur bis zur 80. Spielminute (Sie können davon ausgehen, dass Spiele aufgrund der Fernsehübertragungen pünktlich angepfiffen werden – und es keine Nachspielzeiten gibt).

*Aus dem Fußball-Lexikon:* Ein Fußballspiel dauert 2 x 45 Minuten, unterbrochen durch eine 15-minütige Spielpause.

Wichtig ist, dass die abgegebenen Tipps unterschiedlich gewichtet werden müssen. Ein Tipp, der vor dem Spiel abgegeben wurde, führt natürlich zu mehr Punkten als ein Tipp kurz vor Ende des Spiels. Überlegen Sie sich ein faires System, das auch für Spannung sorgt (belasse ich meinen Tipp, weil er mir im Falle eines korrekten Ergebnisses sehr viele Punkte bringen würde – oder ändere ich aufgrund des Spielverlaufs ständig meinen Tipp und hamstere „sichere“ Punkte ein).

### Ausbaustufe 1: EURO-Bet-UI

**(24 Punkte)**

Der Plattformbetreiber möchte vor der endgültigen Auftragserteilung der Online-Plattform zunächst einen Prototyp des Verwaltungstools sehen.

Entwickeln Sie daher mit Hilfe von JavaFX für das Verwaltungstool einen funktionstüchtigen, ausbaufähigen Benutzeroberflächen-Prototyp. Versuchen Sie eine möglichst intuitiv zu verwendenden Benutzeroberfläche zu entwerfen. Setzen Sie dafür Ihr in UEN erworbenes Wissen ein. Trennen Sie den Entwurfs- vom Implementierungsprozess, indem Sie Ihre Benutzeroberfläche zunächst mit Mockups (Grobentwurf der Benutzeroberfläche) modellieren. Fügen Sie die Mockups zu Ihrer Systemdokumentation hinzu.

Trennen Sie den Code zur Realisierung der grafischen Benutzeroberfläche vom Code zur Repräsentation der Daten Ihrer Anwendung (Benutzer, Mannschaften, Spiele etc.). Durch diese Maßnahme wird Ihre Anwendung einfach erweiterbar, was Ihnen in der nächsten Ausbaustufe zugutekommen sollte. Die Verwendung von FXML und Werkzeugen zum Design der grafischen Benutzeroberfläche (JavaFX Scene Builder) ist nicht erlaubt.

Ihr Prototyp soll es ermöglichen, alle Fenster und Dialoge der Benutzeroberfläche zu öffnen (und diese auch wieder zu schließen). Die Anwendung muss die Eingaben aber noch nicht über die Programmlaufzeit hinaus speichern können. Als Ersatz dafür können Sie mit hart codierten Daten arbeiten. Der Benutzeroberflächen-Prototyp muss folgende Funktionen abdecken:

- Benutzer hinzufügen/sperren
- Mannschaften hinzufügen/bearbeiten/löschen
- Spielpaarungen hinzufügen/bearbeiten/löschen
- Spielergebnisse eintragen

Beachten Sie, dass die Implementierung der anderen Systemkomponenten erst in weiteren Ausbaustufen gefordert ist. Die genauen technischen Anforderungen an diese Komponenten werden im weiteren Verlauf der Übung bekannt gegeben.

# Inhaltsverzeichnis

<b>1</b>	<b>Euro-Bet-UI</b>	<b>4</b>
1.1	Lösungsidee . . . . .	4
1.1.1	Mockup . . . . .	4
1.2	Quellcode . . . . .	6
1.2.1	swe4.gui . . . . .	6
1.2.2	swe4.gui.components . . . . .	10
1.2.3	swe4.gui.helper . . . . .	15
1.2.4	swe4.gui.pages . . . . .	16
1.2.5	swe4.gui.panels . . . . .	21
1.2.6	swe4.models . . . . .	41
1.2.7	Style . . . . .	44
1.3	Testfälle . . . . .	45

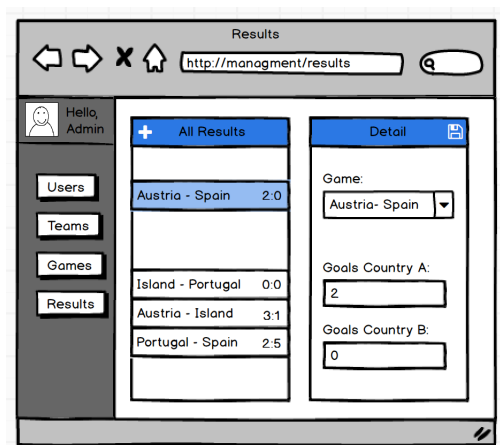
# 1 Euro-Bet-UI

## 1.1 Lösungsidee

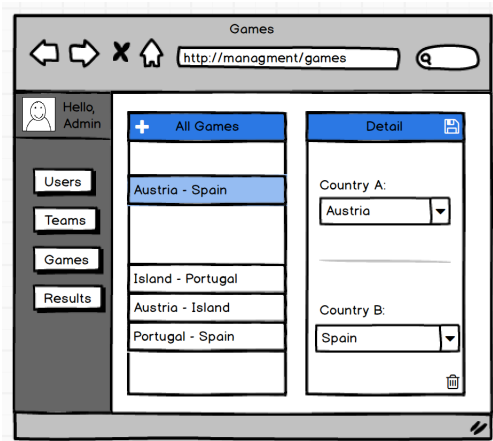
Um eine möglichst modulare Architektur zu erzeugen, habe ich Models (Users, Team, User) erzeugt. Die UI wird mittels Panels aufgebaut. Die Liste der User ist beispielsweise in einem Panel. Panels wie zum Beispiel die Liste der User, die Detailansicht der User erben somit von der abstrakten Klasse 'Panel'. Die abstrakte Panelklasse kümmert sich um den grafischen Aufbau. Das Haupt und das Detailpanel werden in 'Pages' zusammengefasst. Für die Verwaltung, welche Page gerade angezeigt werden sollte, kümmert sich das 'RouteComponent'. Die Navigationsleiste ist ebenso eine Komponente. Alle Komponenten kommunizieren über Event Handler und deren öffentlichen Methoden.

### 1.1.1 Mockup

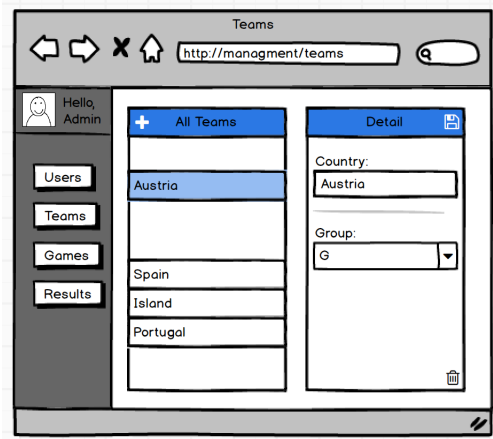
#### Users



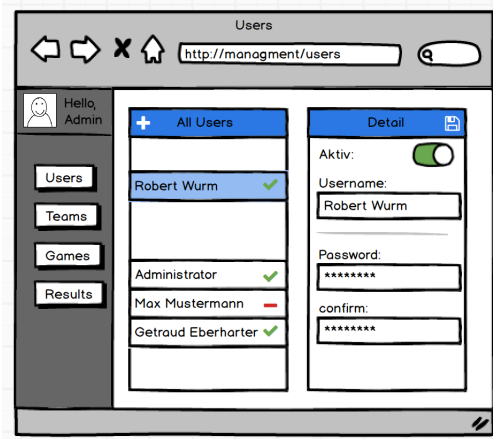
#### Teams



Games



Results



## 1.2 Quellcode

### 1.2.1 swe4.gui

#### EuroBetUI.java

```
1 package swe4.gui;
2
3 import java.util.logging.LogManager;
4 import javafx.application.Application;
5 import javafx.event.EventHandler;
6 import javafx.geometry.Insets;
7 import javafx.scene.Scene;
8 import javafx.scene.input.KeyEvent;
9 import javafx.scene.layout.Background;
10 import javafx.scene.layout.BackgroundFill;
11 import javafx.scene.layout.BorderPane;
12 import javafx.scene.layout.HBox;
13 import javafx.scene.paint.Color;
14 import javafx.stage.Stage;
15 import swe4.gui.components.RouteComponent;
16 import swe4.gui.components.SidebarComponent;
17 import swe4.gui.pages.GamesPage;
18 import swe4.gui.pages.ResultsPage;
19 import swe4.gui.pages.TeamsPage;
20 import swe4.gui.pages.UsersPage;
21
22 public class EuroBetUI extends Application {
23
24     private static final Background ROOT_BACKGROUND = new Background
25         (new BackgroundFill(Color.WHITE, null, null));
26
27     private static final String URL_USERS = "/users";
28     private static final String URL_TEAMS = "/teams";
29     private static final String URL_GAMES = "/games";
30     private static final String URL_RESULTS = "/results";
31
32     public static void main(String[] args) {
33         LogManager.getLogManager().reset();
34         launch(args);
35     }
36
37     @Override
38     public void start(Stage primaryStage) throws Exception {
39
40
41
42         RouteComponent pages = new RouteComponent();
43         pages.setPadding(new Insets(0,0,0,0));
44
45         // add all pages to routing
```

```
46         pages.addPage(URL_USERS, new UsersPage());
47         pages.addPage(URL_TEAMS, new TeamsPage());
48         pages.addPage(URL_GAMES, new GamesPage());
49         pages.addPage(URL_RESULTS, new ResultsPage());
50
51         // show default page
52         pages.showPage(URL_USERS);
53
54         SidebarComponent sideBar = new SidebarComponent();
55
56
57         sideBar.addNavigationButton("Users", "/user.png", event -> {
58             pages.showPage(URL_USERS);
59         });
60
61         sideBar.addNavigationButton("Teams", "/teams.png", event -> {
62             pages.showPage(URL_TEAMS);
63         });
64
65
66         sideBar.addNavigationButton("Games", "/games.png", event -> {
67             pages.showPage(URL_GAMES);
68         });
69
70
71         sideBar.addNavigationButton("Results", "/results.png", event -> {
72             pages.showPage(URL_RESULTS);
73         });
74
75
76         BorderPane rootPane = new BorderPane();
77         rootPane.setLeft(sideBar);
78         rootPane.setCenter(pages);
79         rootPane.setBackground(ROOT_BACKGROUND); // main background
80
81         Scene scene = new Scene(rootPane, 500, 500);
82         scene.getStylesheets().add(
83             getClass().getResource("/euro-bet-ui.css").toExternalForm());
84
85         primaryStage.setScene(scene);
86
87
88         primaryStage.setMinWidth(1070);
89         primaryStage.setMaxWidth(1070);
90         primaryStage.setMinHeight(600);
91         primaryStage.setMaxHeight(600);
92
93         primaryStage.setTitle("EURO-Bet-UI");
94         primaryStage.show();
95     }
96 }
```

## NumberTextField.java

```
1 package swe4.gui;
2
3 import javafx.scene.control.TextField;
4
5 public class NumberTextField extends TextField {
6
7     @Override
8     public void replaceText(int start, int end, String text)
9     {
10         if (validate(text))
11         {
12             super.replaceText(start, end, text);
13         }
14     }
15
16     @Override
17     public void replaceSelection(String text)
18     {
19         if (validate(text))
20         {
21             super.replaceSelection(text);
22         }
23     }
24
25     private boolean validate(String text)
26     {
27         return text.matches("[0-9]*");
28     }
29 }
```

## ToggleSwitch.java

```
1 package swe4.gui;
2
3 import javafx.beans.property.SimpleBooleanProperty;
4 import javafx.geometry.Pos;
5 import javafx.scene.control.Button;
6 import javafx.scene.control.Label;
7 import javafx.scene.layout.HBox;
8
9
10 public class ToggleSwitch extends HBox {
11
12     private final Label label = new Label();
13     private final Button button = new Button();
14 }
```



```
15     private SimpleBooleanProperty switchedOn = new SimpleBooleanProperty(false);
16     public SimpleBooleanProperty switchOnProperty() { return switchedOn; }
17
18     private void init() {
19
20         label.setText("OFF");
21
22         getChildren().addAll(label, button);
23         button.setOnAction((e) -> {
24             switchedOn.set(!switchedOn.get());
25         });
26         label.setOnMouseClicked((e) -> {
27             switchedOn.set(!switchedOn.get());
28         });
29         setStyle();
30         bindProperties();
31     }
32
33     private void setStyle() {
34         // default width
35         setWidth(80);
36         label.setAlignment(Pos.CENTER);
37         setStyle("-fx-background-color: lightgray; "
38             + "-fx-text-fill: black; "
39             + "-fx-background-radius: 0;");
40         setAlignment(Pos.CENTER_LEFT);
41     }
42
43     private void bindProperties() {
44         label.prefWidthProperty().bind(widthProperty().divide(2));
45         label.prefHeightProperty().bind(heightProperty());
46         button.prefWidthProperty().bind(widthProperty().divide(2));
47         button.prefHeightProperty().bind(heightProperty());
48     }
49
50     public ToggleSwitch() {
51         init();
52         switchedOn.addListener((a,b,c) -> {
53             if (c) {
54                 switchOn();
55             }
56             else {
57                 switchOff();
58             }
59         });
60     }
61
62     public void switchOn() {
63         label.setText("ON");
64         setStyle("-fx-background-color: green;");
65         label.toFront();
66     }
67
```

```
68     public void switchOff() {
69         label.setText("OFF");
70         setStyle("-fx-background-color: lightgray;");
71         button.toFront();
72     }
73
74     public void switchToStatus(Boolean status) {
75         if(status) {
76             switchOn();
77         } else {
78             switchOff();
79         }
80     }
81
82 }
```

### 1.2.2 swe4.gui.components

#### PanelComponent.java

```
1 package swe4.gui.components;
2
3 import javafx.event.ActionEvent;
4 import javafx.event.EventHandler;
5 import javafx.scene.Node;
6 import javafx.scene.control.Button;
7 import javafx.scene.control.Label;
8 import javafx.scene.layout.Border;
9 import javafx.scene.layout.BorderStroke;
10 import javafx.scene.layout.BorderStrokeStyle;
11 import javafx.scene.layout.GridPane;
12 import javafx.scene.paint.Color;
13 import javafx.scene.shape.Rectangle;
14 import javafx.scene.text.Font;
15 import swe4.gui.helper.GuiHelper;
16
17 public abstract class PanelComponent extends GridPane{
18
19     private static final Border BORDER = new Border(
20         new BorderStroke(Color.LIGHTGRAY, BorderStrokeStyle.SOLID, null, null)
21     );
22
23     private int numberOfLeftButtons = 0;
24     private int numberOfRightButtons = 0;
25
26     public PanelComponent(String title) {
27
28         this.setBorder(BORDER);
29         Rectangle header = new Rectangle();
```

```
30         header.setX(0);
31         header.setY(0);
32         header.setWidth(400);
33         header.setHeight(40);
34         header.setFill(Color.CADETBBLUE);
35
36
37         Label panelTitle = new Label(title);
38         panelTitle.setFont(Font.font("Helvetica", 20));
39         panelTitle.setStyle("-fx-padding: 0px 0px 0px 150px; -fx-text-fill: #ffffff");
40
41
42
43         Rectangle content = new Rectangle();
44         content.setX(0);
45         content.setY(0);
46         content.setWidth(400);
47         content.setHeight(450);
48         content.setFill(Color.WHITE);
49
50
51         // add all elements to panel
52
53         this.add(header, 0, 0);
54         this.add(panelTitle, 0, 0);
55         this.add(content, 0, 1);
56     }
57
58     protected void addContent(Node component) {
59         this.add(component, 0, 1);
60     }
61
62
63     protected void addHeaderButtonLeft(String id,
64         String iconFile, EventHandler<ActionEvent> handler) {
65         GuiHelper helper = new GuiHelper();
66         Button button = helper.createIconButton(id, iconFile);
67
68         // calculate padding
69         String paddingLeft;
70         if(numberOfLeftButtons == 0) {
71             paddingLeft = "10";
72         } else {
73             paddingLeft = Integer.toString(10 + numberOfLeftButtons * 40);
74         }
75         numberOfLeftButtons ++;
76
77         // set properties
78         button.setStyle("-fx-background-color: transparent; "
79             + "-fx-padding: 0px 0px 0px "+
80             paddingLeft + "px");
81         button.addEventHandler(ActionEvent.ACTION, handler);
82     }
```

```
83         // add button to header
84         this.add(button, 0, 0);
85
86     }
87
88
89
90     protected void addHeaderButtonRight(String id, String iconFile,
91         EventHandler<ActionEvent> handler) {
92         GuiHelper helper = new GuiHelper();
93         Button button = helper.createIconButton(id, iconFile);
94
95         // calculate padding
96         String paddingLeft;
97         if(numberOfRightButtons == 0) {
98             paddingLeft = "370";
99         } else {
100             paddingLeft = Integer.toString(370 - numberOfRightButtons * 40);
101         }
102         numberOfRightButtons ++;
103
104         // set properties
105         button.setStyle("-fx-background-color: transparent;"
106             + " -fx-padding: 0px 0px 0px"
107             + paddingLeft + "px");
108         button.addEventHandler(ActionEvent.ACTION, handler);
109
110         // add button to header
111         this.add(button, 0, 0);
112     }
113
114
115
116
117 }
```

## RouteComponent.java

```
1 package swe4.gui.components;
2
3 import java.util.HashMap;
4 import java.util.Map;
5 import java.util.NoSuchElementException;
6
7 import javafx.geometry.Insets;
8 import javafx.scene.layout.Background;
9 import javafx.scene.layout.BackgroundFill;
10 import javafx.scene.layout.Pane;
11 import javafx.scene.paint.Color;
12
```

```
13 public class RouteComponent extends Pane {
14
15     private static final Background BACKGROUND = new Background
16         (new BackgroundFill(Color.WHITE, null, null));
17
18     private Map<String, Pane> pages = new HashMap<String, Pane>();
19
20     public RouteComponent() {
21
22         //this.setPadding(new Insets(100));
23         this.setBackground(BACKGROUND);
24     }
25
26
27     public void addPage(String url, Pane page) {
28         pages.put(url, page);
29     }
30
31     public void showPage(String url) {
32         Pane page = pages.get(url);
33
34         if(page == null) {
35             System.out.println("Unable to show Page with url: " + url);
36             return;
37         }
38         page.setPadding(new Insets(50, 0, 0, 50));
39
40
41         this.getChildren().clear();
42         this.getChildren().add(page);
43     }
44
45
46
47 }
```

### SidebarComponent.java

```
1 package swe4.gui.components;
2
3 import javafx.event.ActionEvent;
4 import javafx.event.EventHandler;
5 import javafx.geometry.Insets;
6 import javafx.geometry.Pos;
7 import javafx.scene.control.Button;
8 import javafx.scene.control.Label;
9 import javafx.scene.image.Image;
10 import javafx.scene.image.ImageView;
11 import javafx.scene.layout.Background;
12 import javafx.scene.layout.BackgroundFill;
```

```
13 import javafx.scene.layout.HBox;
14 import javafx.scene.layout.Pane;
15 import javafx.scene.layout.VBox;
16 import javafx.scene.paint.Color;
17 import javafx.scene.shape.HLineTo;
18 import javafx.scene.shape.MoveTo;
19 import javafx.scene.shape.Path;
20 import swe4.gui.helper.GuiHelper;
21
22 public class SidebarComponent extends Pane {
23
24
25     private static final Background BACKGROUND = new Background
26         (new BackgroundFill(Color.web("0x21252b"),1.0), null, null));
27
28     private VBox navigationButtonsPane;
29
30
31     public SidebarComponent() {
32
33         // set background
34         this.setBackground(BACKGROUND);
35
36         // create panes
37         Pane userPane = createUserPane();
38         navigationButtonsPane = new VBox();
39         navigationButtonsPane.setSpacing(20);
40         navigationButtonsPane.setPadding(new Insets(100, 20, 0, 25));
41
42         // create horizontal line
43         Path path = new Path();
44         path.getElements().add(new MoveTo(0.0f, 80.0f));
45         path.getElements().add(new HLineTo(145.0f));
46         path.setStroke(Color.WHITE);
47
48         // add all elements
49         this.getChildren().addAll(userPane, path, navigationButtonsPane);
50     }
51
52
53     private Pane createUserPane() {
54
55         Label currentUserText = new Label("Welcome, \n Admin");
56         currentUserText.setId("nav-welcome");
57
58         Image currentUserImage = new Image("/avatar.png");
59
60         ImageView currentUserAvatar = new ImageView();
61         currentUserAvatar.setImage(currentUserImage);
62
63
64         HBox userPane = new HBox(currentUserAvatar, currentUserText);
65         userPane.setSpacing(5);
```

```
66         userPane.setAlignment(Pos.CENTER);
67         userPane.setPadding(new Insets(15,15,0,15));
68
69         return userPane;
70     }
71
72
73
74     public void addNavigationButton(String title, String iconFile,
75                                     EventHandler<ActionEvent> handler) {
76
77         GuiHelper helper = new GuiHelper();
78         Button button = helper.createTextIconButton("nav-button", title, iconFile);
79         button.addEventHandler(ActionEvent.ACTION, handler);
80
81         navigationButtonsPane.getChildren().add(button);
82     }
83
84
85
86 }
```

### 1.2.3 swe4.gui.helper

#### GuiHelper.java

```
1 package swe4.gui.helper;
2
3 import javafx.geometry.Insets;
4 import javafx.scene.control.Button;
5 import javafx.scene.image.Image;
6 import javafx.scene.image.ImageView;
7 import swe4.models.Game;
8
9 public class GuiHelper {
10
11     public Button createIconButton(String id, String iconFile) {
12         Button button = new Button();
13         button.setId(id);
14         button.setPadding(new Insets(10));
15         Image icon = new Image(getClass().getResourceAsStream(iconFile));
16         button.setGraphic(new ImageView(icon));
17
18         return button;
19     }
20
21     public Button createTextButton(String id, String caption) {
22         Button button = new Button(caption);
23         button.setId(id);
```

```
24         button.setPadding(new Insets(10));
25         button.setPrefSize(50, 40);
26         return button;
27     }
28
29     public Button createTextIconButton(String id, String caption, String iconFile) {
30         Button button = new Button(caption);
31         button.setId(id);
32         button.setPadding(new Insets(10));
33         Image icon = new Image(getClass().getResourceAsStream(iconFile));
34         button.setGraphic(new ImageView(icon));
35
36         return button;
37     }
38
39
40 }
```

### 1.2.4 swe4.gui.pages

#### GamesPage.java

```
1 package swe4.gui.pages;
2
3 import javafx.geometry.Insets;
4 import javafx.scene.layout.Background;
5 import javafx.scene.layout.BackgroundFill;
6 import javafx.scene.layout.FlowPane;
7 import javafx.scene.paint.Color;
8 import swe4.gui.panels.GameDetailPanel;
9 import swe4.gui.panels.GamesPanel;
10 import swe4.gui.panels.GamesPanelListener;
11 import swe4.gui.panels.TeamDetailPanel;
12 import swe4.gui.panels.TeamsPanelListener;
13 import swe4.models.Game;
14 import swe4.models.Team;
15
16
17 public class GamesPage extends FlowPane {
18
19     private static final Background BACKGROUND = new Background
20         (new BackgroundFill(Color.WHITE, null, null));
21
22     public GamesPage() {
23
24         this.setBackground(BACKGROUND);
25
26
27         this.setPadding(new Insets(5, 0, 5, 0));
```



```
28         this.setVgap(20);
29         this.setHgap(20);
30         this.setPrefWidth(1300);
31
32
33         // all users panel
34         GamesPanel allGames = new GamesPanel();
35
36         // add detail panel
37         GameDetailPanel detailGame = new GameDetailPanel();
38
39         // handle listeners
40         allGames.addListener(new GamesPanelListener() {
41
42             @Override
43             public void onGameSelected(Game game) {
44                 detailGame.setDetails(game);
45             }
46
47             @Override
48             public void onAddNewGame() {
49                 detailGame.createNewDetail();
50             }
51         });
52
53
54         this.getChildren().addAll(allGames, detailGame);
55     }
56
57
58
59
60 }
```

## ResultsPage.java

```
1 package swe4.gui.pages;
2
3 import javafx.beans.value.ChangeListener;
4 import javafx.beans.value.ObservableValue;
5 import javafx.collections.FXCollections;
6 import javafx.collections.ObservableList;
7 import javafx.geometry.Insets;
8 import javafx.scene.control.Label;
9 import javafx.scene.control.ListView;
10 import javafx.scene.layout.Background;
11 import javafx.scene.layout.BackgroundFill;
12 import javafx.scene.layout.Border;
13 import javafx.scene.layout.BorderStroke;
14 import javafx.scene.layout.BorderStrokeStyle;
```

```
15 import javafx.scene.layout.FlowPane;
16 import javafx.scene.layout.GridPane;
17 import javafx.scene.layout.HBox;
18 import javafx.scene.layout.Pane;
19 import javafx.scene.layout.VBox;
20 import javafx.scene.paint.Color;
21 import javafx.scene.shape.HLineTo;
22 import javafx.scene.shape.MoveTo;
23 import javafx.scene.shape.Path;
24 import javafx.scene.shape.Rectangle;
25 import javafx.scene.shape.VLineTo;
26 import swe4.gui.panels.GamesPanelListener;
27 import swe4.gui.panels.ResultDetailPanel;
28 import swe4.gui.panels.ResultsPanel;
29 import swe4.gui.panels.ResultsPanelListener;
30 import swe4.gui.panels.UserDetailPanel;
31 import swe4.gui.panels.UsersPanel;
32 import swe4.models.Game;
33
34
35 public class ResultsPage extends FlowPane {
36
37
38     private static final Background BACKGROUND = new Background
39         (new BackgroundFill(Color.WHITE, null, null));
40
41
42
43     public ResultsPage() {
44
45         this.setBackground(BACKGROUND);
46
47
48         this.setPadding(new Insets(5, 0, 5, 0));
49         this.setVgap(20);
50         this.setHgap(20);
51         //this.setPrefWrapLength(1300); // preferred width allows for two columns
52         this.setPrefWidth(1300);
53
54
55         // all users panel
56         ResultsPanel allResults = new ResultsPanel();
57
58         // detail panel
59         ResultDetailPanel detailResult = new ResultDetailPanel();
60
61         // handle listeners
62         allResults.addListener(new ResultsPanelListener() {
63
64             @Override
65             public void onResultSelected(Game game) {
66                 detailResult.setDetails(game);
67             }
68         });
69     }
70 }
```

```
68
69         @Override
70         public void onAddNewResult() {
71             detailResult.createNewDetail();
72
73         }
74     });
75
76
77     this.getChildren().addAll(allResults, detailResult);
78 }
79
80
81
82
83 }
```

## TeamsPage.java

```
1 package swe4.gui.pages;
2
3 import javafx.geometry.Insets;
4 import javafx.scene.layout.Background;
5 import javafx.scene.layout.BackgroundFill;
6 import javafx.scene.layout.FlowPane;
7 import javafx.scene.paint.Color;
8 import swe4.gui.panels.TeamDetailPanel;
9 import swe4.gui.panels.TeamsPanel;
10 import swe4.gui.panels.TeamsPanelListener;
11 import swe4.models.Team;
12
13
14 public class TeamsPage extends FlowPane {
15
16
17     private static final Background BACKGROUND = new Background
18         (new BackgroundFill(Color.WHITE, null, null));
19
20
21     public TeamsPage() {
22
23         this.setBackground(BACKGROUND);
24
25
26         this.setPadding(new Insets(5, 0, 5, 0));
27         this.setVgap(20);
28         this.setHgap(20);
29         //this.setPrefWrapLength(1300); // preferred width allows for two columns
30         this.setPrefWidth(1300);
31
32     }
```

```
32
33         // all users panel
34         TeamsPanel allTeams = new TeamsPanel();
35
36         // add detail panel
37         TeamDetailPanel detailTeam = new TeamDetailPanel();
38
39         // handle listeners
40         allTeams.addListener(new TeamsPanelListener() {
41
42             @Override
43             public void onTeamSelected(Team team) {
44                 detailTeam.setDetails(team);
45             }
46
47             @Override
48             public void onAddNewTeam() {
49                 detailTeam.createNewDetail();
50             }
51         });
52
53
54         this.getChildren().addAll(allTeams, detailTeam);
55     }
56
57
58
59
60 }
```

## UsersPage.java

```
1 package swe4.gui.pages;
2
3 import javafx.geometry.Insets;
4 import javafx.scene.layout.Background;
5 import javafx.scene.layout.BackgroundFill;
6 import javafx.scene.layout.FlowPane;
7 import javafx.scene.paint.Color;
8 import swe4.gui.panels.UserDetailPanel;
9 import swe4.gui.panels.UsersPanel;
10 import swe4.gui.panels.UsersPanelListener;
11 import swe4.models.User;
12
13
14 public class UsersPage extends FlowPane {
15
16
17     private static final Background BACKGROUND = new Background
18         (new BackgroundFill(Color.WHITE, null, null));
```

```
19
20
21     public UsersPage() {
22
23         this.setBackground(BACKGROUND);
24
25
26         this.setPadding(new Insets(5, 0, 5, 0));
27         this.setVgap(20);
28         this.setHgap(20);
29         //this.setPrefWrapLength(1300); // preferred width allows for two columns
30         this.setPrefWidth(1300);
31
32
33         // all users panel
34         UsersPanel allUsers = new UsersPanel();
35
36
37         // add detail panel
38         UserDetailPanel detailUser = new UserDetailPanel();
39
40
41         // handle listeners
42         allUsers.addListener(new UsersPanelListener() {
43
44             @Override
45             public void onUserSelected(User user) {
46                 detailUser.setDetails(user);
47             }
48
49             @Override
50             public void onAddNewUser() {
51                 detailUser.createNewDetail();
52             }
53         });
54
55
56
57         this.getChildren().addAll(allUsers, detailUser);
58     }
59 }
```

### 1.2.5 swe4.gui.panels

#### GameDetailPanel.java

```
1 package swe4.gui.panels;
2
3 import java.util.ArrayList;
```

```
4 import java.util.List;
5
6 import javafx.collections.FXCollections;
7 import javafx.collections.ObservableList;
8 import javafx.geometry.Insets;
9 import javafx.scene.control.ComboBox;
10 import javafx.scene.control.Label;
11 import javafx.scene.control.PasswordField;
12 import javafx.scene.control.TextField;
13 import javafx.scene.layout.VBox;
14 import swe4.gui.ToggleSwitch;
15 import swe4.gui.components.PanelComponent;
16 import swe4.models.Game;
17 import swe4.models.Team;
18 import swe4.models.User;
19
20 public class GameDetailPanel extends PanelComponent {
21
22
23     private ComboBox<String> teamABox;
24     private ComboBox<String> teamBBox;
25
26     ArrayList<Team> teams = new ArrayList<Team>();
27
28     public GameDetailPanel() {
29         super("Details");
30
31         addHeaderButtonRight("game-save-button", "/save.png", event -> {
32             System.out.println("game-save-button: " + event.getSource());
33         });
34
35         createDetailFields();
36     }
37
38     private void createDetailFields() {
39
40         VBox details = new VBox();
41         details.setSpacing(50);
42         details.setId("game-detail-container");
43
44
45         // create demo teams
46         List<Game> games = new ArrayList<>();
47
48         Team t1 = new Team("Austria", 'G');
49         Team t2 = new Team("Spain", 'G');
50         Team t3 = new Team("Island", 'G');
51         Team t4 = new Team("Portugal", 'A');
52
53
54         // team a
55         VBox groupA = new VBox();
```

```
57         groupA.setSpacing(3);
58         Label groupATitle = new Label("Team A:");
59         ObservableList<String> optionsA =
60             FXCollections.observableArrayList(
61                 t1.getName(),
62                 t2.getName(),
63                 t3.getName(),
64                 t4.getName()
65             );
66         teamABox = new ComboBox<String>(optionsA);
67         teamABox.setMinWidth(340);
68         groupA.getChildren().addAll(groupATitle, teamABox);
69         VBox.setMargin(groupA, new Insets(60, 30, 0, 30));
70
71
72
73         // team b
74         VBox groupB = new VBox();
75         groupB.setSpacing(3);
76         Label groupBTitle = new Label("Team B:");
77         ObservableList<String> optionsB =
78             FXCollections.observableArrayList(
79                 t1.getName(),
80                 t2.getName(),
81                 t3.getName(),
82                 t4.getName()
83             );
84         teamBBox = new ComboBox<String>(optionsB);
85         teamBBox.setMinWidth(340);
86         groupB.getChildren().addAll(groupBTitle, teamBBox);
87         VBox.setMargin(groupB, new Insets(0, 30, 0, 30));
88
89
90         details.getChildren().addAll(
91             groupA, groupB
92         );
93
94
95         addContent(details);
96
97     }
98
99     public void setDetails(Game game) {
100         if(game == null) { return; }
101         teamABox.setValue(game.getTeamA().getName());
102         teamBBox.setValue(game.getTeamB().getName());
103     }
104
105     public void createNewDetail() {
106         teamABox.setValue("");
107         teamBBox.setValue("");
108     }
109
```

```
110  
111 }
```

## GamesPanel.java

```
1 package swe4.gui.panels;  
2  
3 import java.util.ArrayList;  
4 import java.util.List;  
5  
6 import javafx.beans.value.ChangeListener;  
7 import javafx.beans.value.ObservableValue;  
8 import javafx.collections.FXCollections;  
9 import javafx.collections.ObservableList;  
10 import javafx.scene.control.ListCell;  
11 import javafx.scene.control.ListView;  
12 import javafx.scene.layout.Border;  
13 import javafx.scene.layout.BorderStroke;  
14 import javafx.scene.layout.BorderStrokeStyle;  
15 import javafx.scene.paint.Color;  
16 import javafx.util.Callback;  
17 import swe4.gui.components.PanelComponent;  
18 import swe4.models.Game;  
19 import swe4.models.Team;  
20  
21  
22 public class GamesPanel extends PanelComponent{  
23  
24     private static final Border LIST_BORDER = new Border(  
25         new BorderStroke(Color.TRANSPARENT, BorderStrokeStyle.NONE, null, null)  
26     );  
27  
28  
29     private List<GamesPanelListener> listeners = new ArrayList<GamesPanelListener>();  
30  
31     public GamesPanel() {  
32         super("All Games");  
33  
34  
35         addHeaderButtonLeft("games-add-button", "/plus.png", event -> {  
36             // System.out.println("games-add-button: "+ event.getSource());  
37  
38             // call all added listeners  
39             for (GamesPanelListener l : listeners) {  
40                 l.onAddNewGame();  
41             }  
42         });  
43  
44  
45         createGamesList();
```



```
46     }
47
48
49     private void createGamesList() {
50         ListView<Game> gamesListView = new ListView<Game>();
51         gamesListView.setBorder(LIST_BORDER);
52
53
54
55         // create demo teams
56         List<Game> games = new ArrayList<>();
57
58         Team t1 = new Team("Austria", 'G');
59         Team t2 = new Team("Spain", 'G');
60         Team t3 = new Team("Island", 'G');
61         Team t4 = new Team("Portugal", 'A');
62
63         // create demo games
64         Game g1 = new Game(t1, t2);
65         Game g2 = new Game(t2, t3);
66         Game g3 = new Game(t3, t4);
67         Game g4 = new Game(t1, t4);
68         games.add(g1);
69         games.add(g2);
70         games.add(g3);
71         games.add(g4);
72
73
74         ObservableList<Game> items = FXCollections.observableList(games);
75         gamesListView.setItems(items);
76
77
78         gamesListView.setCellFactory(new Callback<ListView<Game>, ListCell<Game>>(){
79
80             @Override
81             public ListCell<Game> call(ListView<Game> p) {
82
83                 ListCell<Game> cell = new ListCell<Game>(){
84
85                     @Override
86                     protected void updateItem(Game t, boolean bln) {
87                         super.updateItem(t, bln);
88                         if (t != null) {
89                             setText(t.getTeamA().getName() + " - " + t.getTeamB().getName());
90                         }
91                     }
92                 };
93                 return cell;
94             }
95         });
96
97         gamesListView.getSelectionModel().selectedItemProperty()
98             .addListener(new ChangeListener<Game>() {
```

```
99
100         @Override
101         public void changed(ObservableValue<? extends Game> observable,
102                             Game oldValue, Game newValue) {
103
104             // call all added listeners
105             for (GamesPanellListener l : listeners) {
106                 l.onGameSelected(newValue);
107             }
108         }
109     });
110
111     addContent(gamesListView);
112 }
113
114
115     public void addListener(GamesPanellListener listener) {
116         listeners.add(listener);
117     }
118
119
120 }
```

### GamesPanellListener.java

```
1 package swe4.gui.panels;
2
3 import swe4.models.Game;
4
5 public interface GamesPanellListener {
6     void onGameSelected(Game game);
7     void onAddNewGame();
8 }
```

### ResultDetailPanel.java

```
1 package swe4.gui.panels;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import javafx.collections.FXCollections;
7 import javafx.collections.ObservableList;
8 import javafx.geometry.Insets;
9 import javafx.scene.control.ComboBox;
10 import javafx.scene.control.Label;
```

```
11 import javafx.scene.control.PasswordField;
12 import javafx.scene.control.TextField;
13 import javafx.scene.layout.VBox;
14 import swe4.gui.NumberTextField;
15 import swe4.gui.ToggleSwitch;
16 import swe4.gui.components.PanelComponent;
17 import swe4.gui.helper.GuiHelper;
18 import swe4.models.Game;
19 import swe4.models.Team;
20 import swe4.models.User;
21
22 public class ResultDetailPanel extends PanelComponent {
23
24
25     private NumberTextField goalsTeamABox;
26     private NumberTextField goalsTeamBBox;
27
28     ArrayList<Team> teams = new ArrayList<Team>();
29     private ComboBox<String> gameBox;
30
31     public ResultDetailPanel() {
32         super("Details");
33
34         addHeaderButtonRight("result-save-button", "/save.png", event -> {
35             System.out.println("result-save-button: " + event.getSource());
36         });
37
38         createDetailFields();
39     }
40
41     private void createDetailFields() {
42
43         VBox details = new VBox();
44         details.setSpacing(50);
45         details.setId("result-detail-container");
46
47
48
49
50
51         // create demo teams
52         List<Game> games = new ArrayList<>();
53         Team t1 = new Team("Austria", 'G');
54         Team t2 = new Team("Spain", 'G');
55         Team t3 = new Team("Island", 'G');
56         Team t4 = new Team("Portugal", 'A');
57
58         // create demo games which are finished
59         Game g1 = new Game(t1, t2);
60         g1.setGameFinished(true);
61         g1.setGoalsTeamA(2);
62         g1.setGoalsTeamB(0);
63
```

```
64     Game g2 = new Game(t2, t3);
65     g2.setGameFinished(true);
66     g2.setGoalsTeamA(4);
67     g2.setGoalsTeamB(1);
68
69     Game g3 = new Game(t3, t4);
70     g3.setGameFinished(true);
71     g3.setGoalsTeamA(0);
72     g3.setGoalsTeamB(3);
73
74     Game g4 = new Game(t1, t4);
75     g4.setGameFinished(true);
76     g4.setGoalsTeamA(2);
77     g4.setGoalsTeamB(2);
78
79     games.add(g1);
80     games.add(g2);
81     games.add(g3);
82     games.add(g4);
83
84
85
86     // game
87     VBox game = new VBox();
88     game.setSpacing(3);
89     Label gameTitle = new Label("Game:");
90     ObservableList<String> optionsB =
91         FXCollections.observableArrayList(
92             g1.toString(),
93             g2.toString(),
94             g3.toString(),
95             g4.toString()
96         );
97     gameBox = new ComboBox<String>(optionsB);
98     gameBox.setMinWidth(340);
99     game.getChildren().addAll(gameTitle, gameBox);
100    VBox.setMargin(game, new Insets(30, 30, 0, 30));
101
102
103    // goals team a
104    VBox goalsA = new VBox();
105    goalsA.setSpacing(3);
106    Label goalsATitle = new Label("Goals Team A:");
107    goalsTeamABox = new NumberTextField();
108    goalsA.getChildren().addAll(goalsATitle, goalsTeamABox);
109    VBox.setMargin(goalsA, new Insets(0, 30, 0, 30));
110
111
112    // goals team b
113    VBox goalsB = new VBox();
114    goalsB.setSpacing(3);
115    Label goalsBTitle = new Label("Goals Team B:");
116    goalsTeamBBox = new NumberTextField();
```

```
117         goalsA.getChildren().addAll(goalsBTitle, goalsTeamBBox);
118         VBox.setMargin(goalsB, new Insets(0, 30, 0, 30));
119
120         details.getChildren().addAll(
121             game, goalsA, goalsB
122         );
123
124         addContent(details);
125
126     }
127
128     public void setDetails(Game game) {
129         if(game == null) { return; }
130
131         gameBox.setValue(game.toString());
132         goalsTeamABox.setText(game.getGoalsTeamA().toString());
133         goalsTeamBBox.setText(game.getGoalsTeamB().toString());
134     }
135
136     public void createNewDetail() {
137         gameBox.setValue("");
138         goalsTeamABox.setText("");
139         goalsTeamBBox.setText("");
140     }
141
142
143 }
```

## ResultsPanel.java

```
1 package swe4.gui.panels;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import javafx.beans.value.ChangeListener;
7 import javafx.beans.value.ObservableValue;
8 import javafx.collections.FXCollections;
9 import javafx.collections.ObservableList;
10 import javafx.scene.control.ListCell;
11 import javafx.scene.control.ListView;
12 import javafx.scene.layout.Border;
13 import javafx.scene.layout.BorderStroke;
14 import javafx.scene.layout.BorderStrokeStyle;
15 import javafx.scene.paint.Color;
16 import javafx.util.Callback;
17 import swe4.gui.components.PanelComponent;
18 import swe4.gui.helper.GuiHelper;
19 import swe4.models.Game;
20 import swe4.models.Team;
```

```
21
22
23 public class ResultsPanel extends PanelComponent{
24
25     private static final Border LIST_BORDER = new Border(
26         new BorderStroke(Color.TRANSPARENT, BorderStrokeStyle.NONE, null, null)
27     );
28
29
30     private List<ResultsPanelListener> listeners = new ArrayList<ResultsPanelListener>();
31
32     public ResultsPanel() {
33         super("All Results");
34
35         addHeaderButtonLeft("results-add-button", "/plus.png", event -> {
36             // System.out.println("results-add-button: "+ event.getSource());
37             // call all added listeners
38             for (ResultsPanelListener l : listeners) {
39                 l.onAddNewResult();
40             }
41         });
42
43
44         createGamesList();
45     }
46
47
48     private void createGamesList() {
49         ListView<Game> gamesListView = new ListView<Game>();
50         gamesListView.setBorder(LIST_BORDER);
51
52         List<Game> games = new ArrayList<>();
53
54         // create demo teams
55         Team t1 = new Team("Austria", 'G');
56         Team t2 = new Team("Spain", 'G');
57         Team t3 = new Team("Island", 'G');
58         Team t4 = new Team("Portugal", 'A');
59
60         // create demo games which are finished
61         Game g1 = new Game(t1, t2);
62         g1.setGameFinished(true);
63         g1.setGoalsTeamA(2);
64         g1.setGoalsTeamB(0);
65
66         Game g2 = new Game(t2, t3);
67         g2.setGameFinished(true);
68         g2.setGoalsTeamA(4);
69         g2.setGoalsTeamB(1);
70
71         Game g3 = new Game(t3, t4);
72         g3.setGameFinished(true);
73         g3.setGoalsTeamA(0);
```

```
74         g3.setGoalsTeamB(3);
75
76         Game g4 = new Game(t1, t4);
77         g4.setGameFinished(true);
78         g4.setGoalsTeamA(2);
79         g4.setGoalsTeamB(2);
80
81         games.add(g1);
82         games.add(g2);
83         games.add(g3);
84         games.add(g4);
85
86
87         ObservableList<Game> items = FXCollections.observableList(games);
88         gamesListView.setItems(items);
89
90
91         gamesListView.setCellFactory(new Callback<ListView<Game>, ListCell<Game>>(){
92
93             @Override
94             public ListCell<Game> call(ListView<Game> p) {
95
96                 ListCell<Game> cell = new ListCell<Game>(){
97
98                     @Override
99                     protected void updateItem(Game g, boolean bln) {
100                         super.updateItem(g, bln);
101                         if (g != null) {
102                             setText(g.getResult() + "      " + g.toString());
103                         }
104                     }
105                 };
106                 return cell;
107             }
108         });
109
110         gamesListView.getSelectionModel().selectedItemProperty()
111             .addListener(new ChangeListener<Game>() {
112
113             @Override
114             public void changed(ObservableValue<? extends Game> observable,
115                                 Game oldValue, Game newValue) {
116
117                 // call all added listeners
118                 for (ResultsPanelListener l : listeners) {
119                     l.onResultSelected(newValue);
120                 }
121             }
122         });
123
124         addContent(gamesListView);
125     }
126
```

```
127
128     public void addListener(ResultsPanelListener listener) {
129         listeners.add(listener);
130     }
131
132
133 }
```

### ResultsPanelListener.java

```
1 package swe4.gui.panels;
2
3 import swe4.models.Game;
4
5 public interface ResultsPanelListener {
6     void onResultSelected(Game game);
7     void onAddNewResult();
8 }
```

### TeamDetailPanel.java

```
1 package swe4.gui.panels;
2
3 import javafx.collections.FXCollections;
4 import javafx.collections.ObservableList;
5 import javafx.geometry.Insets;
6 import javafx.scene.control.ComboBox;
7 import javafx.scene.control.Label;
8 import javafx.scene.control.PasswordField;
9 import javafx.scene.control.TextField;
10 import javafx.scene.layout.VBox;
11 import swe4.gui.ToggleSwitch;
12 import swe4.gui.components.PanelComponent;
13 import swe4.models.Team;
14 import swe4.models.User;
15
16 public class TeamDetailPanel extends PanelComponent {
17
18
19     private TextField nameField;
20     private ComboBox<String> groupBox;
21     private final String
22         GROUP_A = "GROUPE A", GROUP_B = "GROUPE B", GROUP_C = "GROUPE C",
23         GROUP_D = "GROUPE D", GROUP_E = "GROUPE E", GROUP_F = "GROUPE F",
24         GROUP_G = "GROUPE G";
25 }
```



```
26     public TeamDetailPanel() {
27         super("Details");
28
29         addHeaderButtonRight("team-save-button", "/save.png", event -> {
30             System.out.println("team-save-button: " + event.getSource());
31         });
32
33         createDetailFields();
34     }
35
36     private void createDetailFields() {
37
38         VBox details = new VBox();
39         details.setSpacing(20);
40         details.setId("team-detail-container");
41
42         //details.setPadding(new Insets(100));
43
44
45
46
47
48         // team name
49         VBox username = new VBox();
50         username.setSpacing(3);
51         Label nameTitle = new Label("Country:");
52         nameField = new TextField();
53         username.getChildren().addAll(nameTitle, nameField);
54         VBox.setMargin(username, new Insets(30, 30, 0, 30));
55
56
57         // team group
58         VBox group = new VBox();
59         group.setSpacing(3);
60         Label groupTitle = new Label("Group:");
61         ObservableList<String> options =
62             FXCollections.observableArrayList(
63                 GROUP_A,
64                 GROUP_B,
65                 GROUP_C,
66                 GROUP_D,
67                 GROUP_E,
68                 GROUP_F,
69                 GROUP_G
70             );
71         groupBox = new ComboBox<String>(options);
72         group.getChildren().addAll(groupTitle, groupBox);
73         VBox.setMargin(group, new Insets(0, 30, 0, 30));
74
75
76         details.getChildren().addAll(
77             username, group
78         );
```

```
79
80
81         addContent(details);
82
83     }
84
85     public void setDetails(Team team) {
86         if(team == null) { return; }
87
88         nameField.setText(team.getName());
89
90         if(team.getGroup() == 'A') {
91             groupBox.setValue(GROUP_A);
92
93         } else if(team.getGroup() == 'B') {
94             groupBox.setValue(GROUP_B);
95
96         } else if(team.getGroup() == 'C') {
97             groupBox.setValue(GROUP_C);
98
99         } else if(team.getGroup() == 'D') {
100             groupBox.setValue(GROUP_D);
101
102         } else if(team.getGroup() == 'E') {
103             groupBox.setValue(GROUP_E);
104
105         } else if(team.getGroup() == 'F') {
106             groupBox.setValue(GROUP_F);
107
108         } else if(team.getGroup() == 'G') {
109             groupBox.setValue(GROUP_G);
110         }
111     }
112
113
114     public void createNewDetail() {
115         nameField.setText("");
116         groupBox.setValue("");
117
118     }
119
120
121 }
```

### TeamsPanel.java

```
1 package swe4.gui.panels;
2
3 import java.util.ArrayList;
4 import java.util.List;
```

```
5
6 import javafx.beans.value.ChangeListener;
7 import javafx.beans.value.ObservableValue;
8 import javafx.collections.FXCollections;
9 import javafx.collections.ObservableList;
10 import javafx.scene.control.ListCell;
11 import javafx.scene.control.ListView;
12 import javafx.scene.layout.Border;
13 import javafx.scene.layout.BorderStroke;
14 import javafx.scene.layout.BorderStrokeStyle;
15 import javafx.scene.paint.Color;
16 import javafx.util.Callback;
17 import swe4.gui.components.PanelComponent;
18 import swe4.models.Team;
19
20
21 public class TeamsPanel extends PanelComponent{
22
23     private static final Border LIST_BORDER = new Border(
24         new BorderStroke(Color.TRANSPARENT, BorderStrokeStyle.NONE, null, null)
25     );
26
27
28     private List<TeamsPanelListener> listeners = new ArrayList<TeamsPanelListener>();
29
30     public TeamsPanel() {
31         super("All Teams");
32
33
34         addHeaderButtonLeft("teams-add-button", "/plus.png", event -> {
35             // call all added listeners
36             for (TeamsPanelListener l : listeners) {
37                 l.onAddNewTeam();
38             }
39
40         });
41
42
43         createTeamsList();
44     }
45
46
47     private void createTeamsList() {
48         ListView<Team> teamListView = new ListView<Team>();
49         teamListView.setBorder(LIST_BORDER);
50
51         List<Team> teams = new ArrayList<>();
52
53         // create demo teams
54         Team t1 = new Team("Austria", 'G');
55         teams.add(t1);
56
57         Team t2 = new Team("Spain", 'G');
```

```
58     teams.add(t2);
59
60     Team t3 = new Team("Island", 'G');
61     teams.add(t3);
62
63     Team t4 = new Team("Portugal", 'A');
64     teams.add(t4);
65
66
67
68     ObservableList<Team> items = FXCollections.observableList(teams);
69
70
71     teamListView.setItems(items);
72
73
74     teamListView.setCellFactory(new Callback<ListView<Team>, ListCell<Team>>(){
75
76         @Override
77         public ListCell<Team> call(ListView<Team> p) {
78
79             ListCell<Team> cell = new ListCell<Team>(){
80
81                 @Override
82                 protected void updateItem(Team t, boolean bln) {
83                     super.updateItem(t, bln);
84                     if (t != null) {
85                         setText(t.getName());
86                     }
87                 }
88             };
89             return cell;
90         }
91     });
92
93     teamListView.getSelectionModel().selectedItemProperty()
94         .addListener(new ChangeListener<Team>() {
95
96         @Override
97         public void changed(ObservableValue<? extends Team> observable,
98                             Team oldValue, Team newValue) {
99
100             // call all added listeners
101             for (TeamsPanelListener l : listeners) {
102                 l.onTeamSelected(newValue);
103             }
104         }
105     });
106
107     addContent(teamListView);
108 }
109
110
```

```
111         public void addListener(TeamsPanelListener listener) {
112             listeners.add(listener);
113         }
114
115
116 }
```

### TeamsPanelListener.java

```
1 package swe4.gui.panels;
2
3 import swe4.models.Team;
4
5 public interface TeamsPanelListener {
6     void onTeamSelected(Team team);
7     void onAddNewTeam();
8 }
```

### UserDetailPanel.java

```
1 package swe4.gui.panels;
2
3 import javafx.geometry.Insets;
4 import javafx.scene.control.Label;
5 import javafx.scene.control.PasswordField;
6 import javafx.scene.control.TextField;
7 import javafx.scene.layout.VBox;
8 import swe4.gui.ToggleSwitch;
9 import swe4.gui.components.PanelComponent;
10 import swe4.models.User;
11
12 public class UserDetailPanel extends PanelComponent {
13
14     private ToggleSwitch switchActiv;
15     private TextField nameField;
16     private PasswordField passwordField;
17     private PasswordField confirmPassword;
18
19     public UserDetailPanel() {
20         super("Details");
21
22         addHeaderButtonRight("users-save-button", "/save.png", event -> {
23             System.out.println("users-save-button: " + event.getSource());
24         });
25
26         createDetailFields();
27     }
28 }
```

```
27     }
28
29     private void createDetailFields() {
30
31         VBox details = new VBox();
32         details.setSpacing(20);
33         details.setId("user-detail-container");
34
35         //details.setPadding(new Insets(100));
36
37
38
39         // active
40         VBox active = new VBox();
41         Label activTitle = new Label("Active:");
42         switchActiv = new ToggleSwitch();
43         VBox switchPane = new VBox();
44         switchPane.getChildren().add(switchActiv);
45         VBox.setMargin(switchPane, new Insets(-20, 0, 0, 250));
46         active.getChildren().addAll(activTitle, switchPane);
47         VBox.setMargin(active, new Insets(20, 30, 0, 30));
48
49
50         // username
51         VBox username = new VBox();
52         username.setSpacing(3);
53         Label nameTitle = new Label("Username:");
54         nameField = new TextField();
55         username.getChildren().addAll(nameTitle, nameField);
56         VBox.setMargin(username, new Insets(0, 30, 0, 30));
57
58
59         // password
60         VBox password = new VBox();
61         password.setSpacing(3);
62         Label passwordTitle = new Label("Password:");
63         passwordField = new PasswordField();
64         Label confirmTitle = new Label("Confirm:");
65         confirmField = new PasswordField();
66         password.getChildren().addAll(passwordTitle, passwordField,
67                                     confirmTitle, confirmField);
68         VBox.setMargin(password, new Insets(0, 30, 0, 30));
69
70         details.getChildren().addAll(
71             active, username, password
72         );
73
74
75         addContent(details);
76
77     }
78
79     public void setDetails(User user) {
```

```
80         if(user == null) { return; }
81         switchActiv.switchToStatus(user.getActive());
82         nameField.setText(user.getName());
83         passwordField.setText(user.getPassword());
84         confirmField.setText(user.getPassword());
85     }
86
87     public void createNewDetail() {
88         switchActiv.switchToStatus(false);
89         nameField.setText("");
90         passwordField.setText("");
91         confirmField.setText("");
92     }
93
94
95 }
```

## UsersPanel.java

```
1 package swe4.gui.panels;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import javafx.beans.value.ChangeListener;
7 import javafx.beans.value.ObservableValue;
8 import javafx.collections.FXCollections;
9 import javafx.collections.ObservableList;
10 import javafx.event.EventHandler;
11 import javafx.scene.control.ListCell;
12 import javafx.scene.control.ListView;
13 import javafx.scene.input.MouseEvent;
14 import javafx.scene.layout.Border;
15 import javafx.scene.layout.BorderStroke;
16 import javafx.scene.layout.BorderStrokeStyle;
17 import javafx.scene.paint.Color;
18 import javafx.util.Callback;
19 import swe4.gui.components.PanelComponent;
20 import swe4.models.User;
21
22
23 public class UsersPanel extends PanelComponent{
24
25     private static final Border LIST_BORDER = new Border(
26         new BorderStroke(Color.TRANSPARENT, BorderStrokeStyle.NONE, null, null)
27     );
28
29
30     private List<UsersPanellistener> listeners = new ArrayList<UsersPanellistener>();
31 }
```

```
32     public UsersPanel() {
33         super("All Users");
34
35         addHeaderButtonLeft("users-add-button", "/plus.png", event -> {
36             // call all added listeners
37             for (UsersPanelListener l : listeners) {
38                 l.onAddNewUser();
39             }
40         });
41         createUsersList();
42     }
43
44     private void createUsersList() {
45         ListView<User> personListView = new ListView<User>();
46         personListView.setBorder(LIST_BORDER);
47
48         List<User> users = new ArrayList<>();
49
50         // create demo users
51         User u1 = new User(true, "Robert Wurm", "1234");
52         users.add(u1);
53
54         User u2 = new User(true, "Administrator", "1234");
55         users.add(u2);
56
57         User u3 = new User(false, "Max Mustermann", "1234");
58         users.add(u3);
59
60         User u4 = new User(true, "Gertraud Eberharder", "1234");
61         users.add(u4);
62
63
64
65         ObservableList<User> items = FXCollections.observableList(users);
66
67
68         personListView.setItems(items);
69
70
71         personListView.setCellFactory(new Callback<ListView<User>, ListCell<User>>(){
72
73             @Override
74             public ListCell<User> call(ListView<User> p) {
75
76                 ListCell<User> cell = new ListCell<User>(){
77
78                     @Override
79                     protected void updateItem(User t, boolean bln) {
80                         super.updateItem(t, bln);
81                         if (t != null) {
82                             setText(t.getName());
83                         }
84                     }
85                 }
86             }
87         });
88     }
89 }
```



```
85         };
86         return cell;
87     }
88 });
89
90 personListView.getSelectionModel().selectedItemProperty()
91     .addListener(new ChangeListener<User>() {
92
93         @Override
94         public void changed(ObservableValue<? extends User> observable,
95             User oldValue, User newValue) {
96
97             // call all added listeners
98             for (UsersPanelListener l : listeners) {
99                 l.onUserSelected(newValue);
100             }
101         }
102     });
103
104 addContent(personListView);
105 }
106
107
108 public void addListener(UsersPanelListener listener) {
109     listeners.add(listener);
110 }
111
112
113 }
```

### UsersPanelListener.java

```
1 package swe4.gui.panels;
2
3 import swe4.models.User;
4
5 public interface UsersPanelListener {
6     void onUserSelected(User user);
7     void onAddNewUser();
8 }
```

#### 1.2.6 swe4.models

### Game.java

```
1 package swe4.models;
2
3 public class Game {
4
5     private Team teamA;
6     private Team teamB;
7
8     private Boolean gameFinished;
9     private Integer goalsTeamA;
10    private Integer goalsTeamB;
11
12
13    public Game() {}
14    public Game(Team teamA, Team teamB) {
15        this.setTeamA(teamA);
16        this.setTeamB(teamB);
17    }
18    public Team getTeamA() {
19        return teamA;
20    }
21    public void setTeamA(Team teamA) {
22        this.teamA = teamA;
23    }
24    public Team getTeamB() {
25        return teamB;
26    }
27    public void setTeamB(Team teamB) {
28        this.teamB = teamB;
29    }
30
31    public String getResult() {
32        if(getGameFinished() && goalsTeamA != null && goalsTeamB != null) {
33            return getGoalsTeamA().toString() + " : " + getGoalsTeamB().toString();
34        }
35        return "not played yet";
36    }
37
38    public Integer getGoalsTeamA() {
39        return goalsTeamA;
40    }
41
42    public void setGoalsTeamA(Integer goalsTeamA) {
43        this.goalsTeamA = goalsTeamA;
44    }
45
46    public Integer getGoalsTeamB() {
47        return goalsTeamB;
48    }
49
50    public void setGoalsTeamB(Integer goalsTeamB) {
51        this.goalsTeamB = goalsTeamB;
52    }
53 }
```

```
53     public Boolean getGameFinished() {
54         return gameFinished;
55     }
56     public void setGameFinished(Boolean gameFinished) {
57         this.gameFinished = gameFinished;
58     }
59
60     @Override
61     public String toString() {
62         return this.teamA.getName() + " - " + this.teamB.getName().toString();
63     }
64
65 }
```

### Team.java

```
1 package swe4.models;
2
3 public class Team {
4
5     private String name;
6     private Character group;
7
8
9     public Team() {
10
11     }
12
13     public Team(String name, Character group) {
14         this.setName(name);
15         this.setGroup(group);
16     }
17
18     public Character getGroup() {
19         return group;
20     }
21
22     public void setGroup(Character group) {
23         this.group = group;
24     }
25
26     public String getName() {
27         return name;
28     }
29
30     public void setName(String name) {
31         this.name = name;
32     }
33 }
```

## User.java

```
1 package swe4.models;
2
3 public class User {
4
5     private Boolean active;
6     private String name;
7     private String password;
8
9
10    public User() {
11
12    }
13
14    public User(Boolean active, String name, String password) {
15        this.active = active;
16        this.name = name;
17        this.password = password;
18    }
19
20
21    public Boolean getActive() {
22        return active;
23    }
24    public void setActive(Boolean active) {
25        this.active = active;
26    }
27    public String getName() {
28        return name;
29    }
30    public void setName(String name) {
31        this.name = name;
32    }
33    public String getPassword() {
34        return password;
35    }
36    public void setPassword(String password) {
37        this.password = password;
38    }
39
40
41
42
43
44 }
```

### 1.2.7 Style

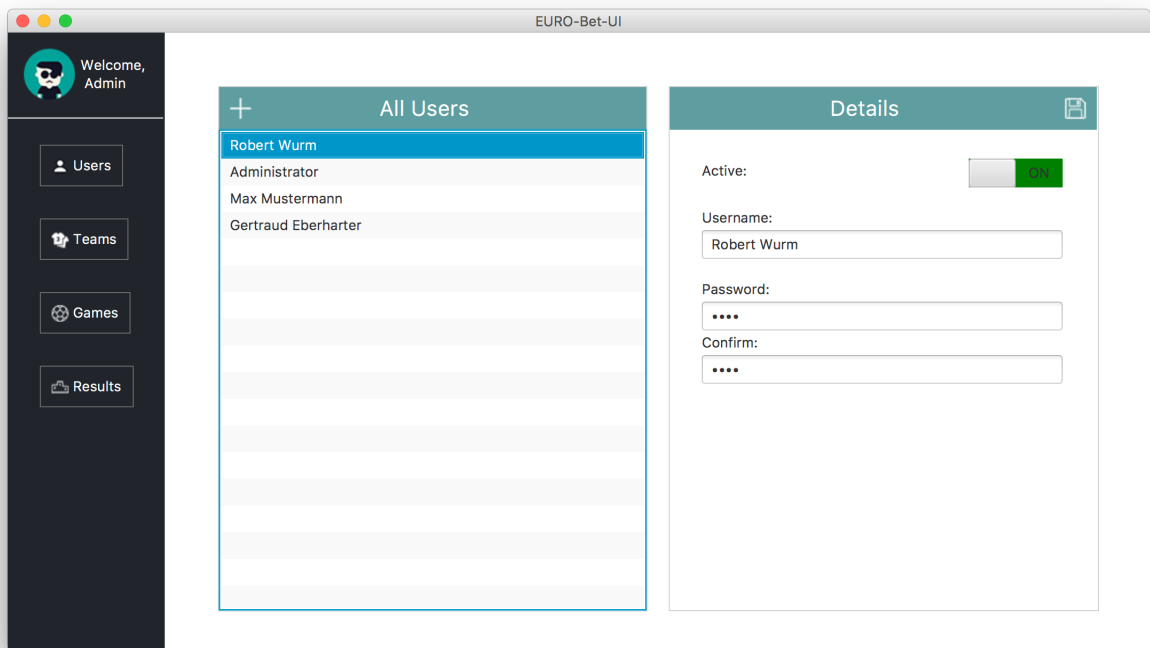
#### euro-bet-ui.css

```
1 #nav-welcome {
2     -fx-text-fill: #ffffff;
3 }
4
5 #nav-button {
6     -fx-border-color: dimgrey;
7     -fx-background-color: transparent;
8     -fx-text-fill: #ffffff;
9     -fx-border-insets: 5px;
10 }
11
12 #nav-button:hover{
13     -fx-background-color: #282c34;
14 }
15
16
17 #user-detail-container {
18     -fx-padding: 10.0 3.0 5.0 0.0;
19 }
20
21 #control-pane {
22     -fx-spacing: 5;
23     -fx-padding: 10,20;
24     -fx-border-color: dimgrey;
25 }
```

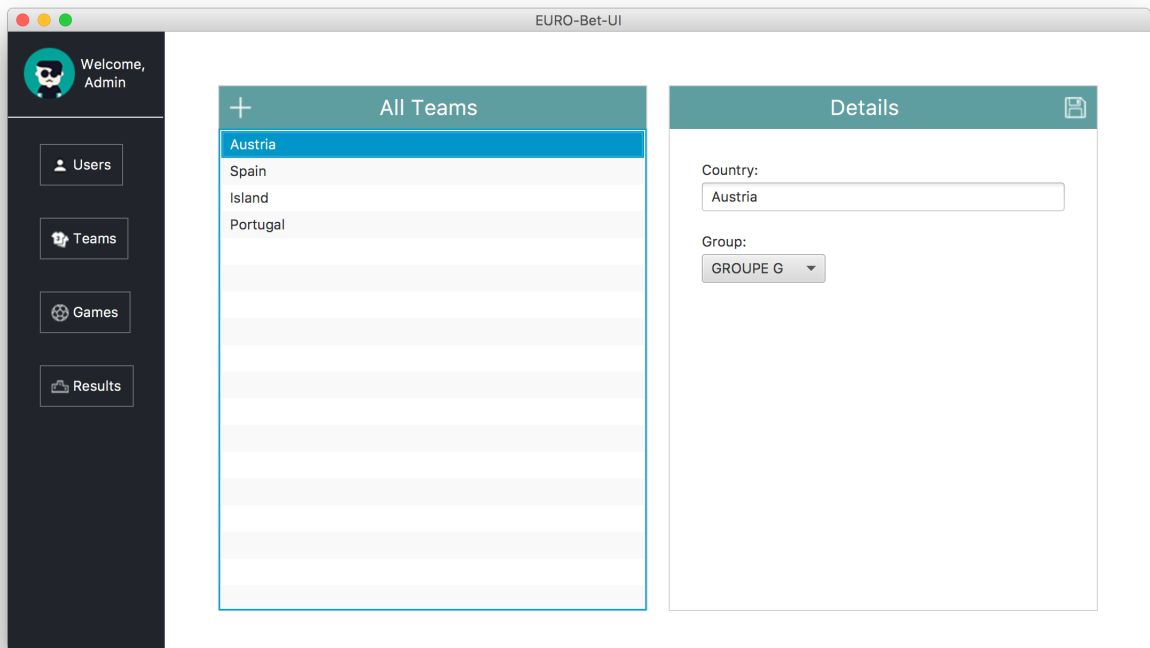
### 1.3 Testfälle

Da es bei dieser Übung rein um die Implementierung des UI ging, führe ich hier nur rein die Screenshots an.

**Users**



Teams



Games

The screenshot shows the EURO-Bet-UI application interface. On the left is a dark sidebar with a user profile (Welcome, Admin) and navigation buttons for Users, Teams, Games, and Results. The main content area is split into two panels. The 'All Games' panel on the left has a teal header with a plus icon and contains a table of games. The 'Details' panel on the right has a teal header with a save icon and contains two dropdown menus for Team A and Team B.

All Games	
Austria - Spain	
Spain - Island	
Island - Portugal	
Austria - Portugal	

Details

Team A:  
Austria

Team B:  
Spain

## Results

The screenshot shows the EURO-Bet-UI application interface with the 'Results' tab selected. The sidebar is identical to the previous view. The 'All Results' panel on the left has a teal header with a plus icon and contains a table of results. The 'Details' panel on the right has a teal header with a save icon and contains a dropdown menu for Game, and two input fields for Goals Team A and Goals Team B.

All Results	
2 : 0	Austria - Spain
4 : 1	Spain - Island
0 : 3	Island - Portugal
2 : 2	Austria - Portugal

Details

Game:  
Austria - Portugal

Goals Team A:  
2

Goals Team B:  
2