SWE 4x **Übung zu Softwareentwicklung mit modernen Plattformen 4** **SS 2016, Übung 6**

Abgabetermin: SA in der KW 23

☒ **Gr. 1, Dr. E. Pitzer**
□ **Gr. 2, DI F. Gruber-Leitner**

Name Robert Wurm **Aufwand in h** 20

Punkte ________ **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 verdächtigt 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

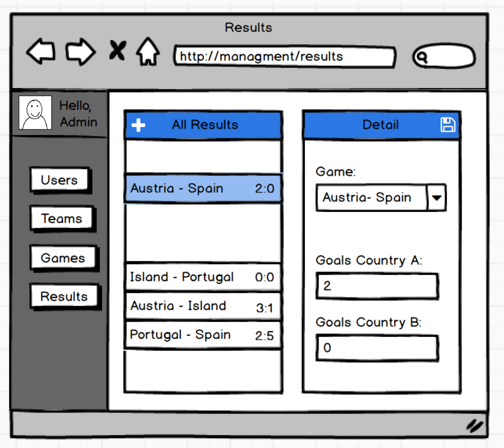
1 Euro-Bet-UI

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 kummunizieren über Event Handler und deren öffentlichen Methoden.

1.1.1 Mockup

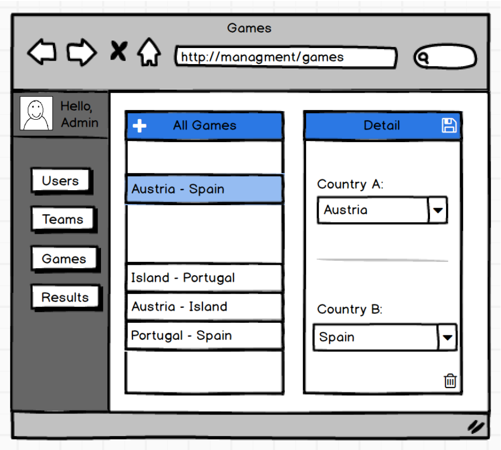
Users



Austria - Spain 2:0
Island - Portugal 0:0
Austria - Island 3:1
Portugal - Spain 2:5

Game: Austria- Spain
Goals Country A: 2
Goals Country B: 0

Teams



Austria - Spain

Island - Portugal

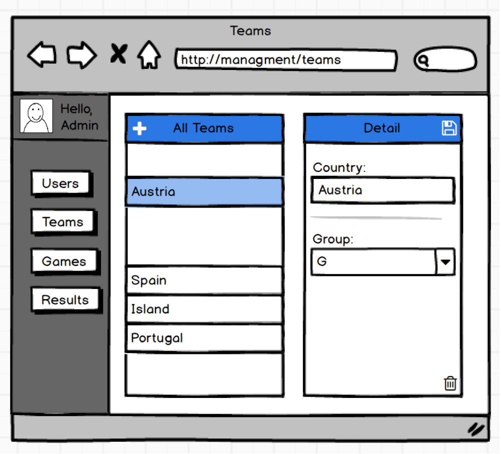
Austria - Island

Portugal - Spain

Country A: Austria

Country B: Spain

Games



Austria

Spain

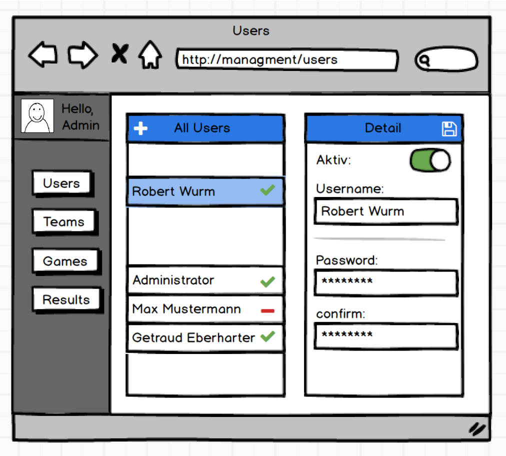
Island

Portugal

Country: Austria

Group: G

Results



Robert Wurm ✓

Administrator ✓

Max Mustermann −

Getraud Eberharter ✓

Aktiv: ●

Username: Robert Wurm

Password: ********

confirm: ********

1.2 Quellcode

1.2.1 swe4.gui

EuroBetUI.java

```java
package swe4.gui;

import java.util.logging.LogManager;
import javafx.application.Application;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.input.KeyEvent;
import javafx.scene.layout.Background;
import javafx.scene.layout.BackgroundFill;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import swe4.gui.components.RouteComponent;
import swe4.gui.components.SidebarComponent;
import swe4.gui.pages.GamesPage;
import swe4.gui.pages.ResultsPage;
import swe4.gui.pages.TeamsPage;
import swe4.gui.pages.UsersPage;

public class EuroBetUI extends Application {

    private static final Background ROOT_BACKGROUND = new Background
            (new BackgroundFill(Color.WHITE, null, null));

    private static final String URL_USERS = "/users";
    private static final String URL_TEAMS = "/teams";
    private static final String URL_GAMES = "/games";
    private static final String URL_RESULTS = "/results";

    public static void main(String[] args) {
        LogManager.getLogManager().reset();
        launch(args);
    }


    @Override
    public void start(Stage primaryStage) throws Exception {


        RouteComponent pages = new RouteComponent();
        pages.setPadding(new Insets(0,0,0,0));

        // add all pages to routing
```

```
                pages.addPage(URL_USERS, new UsersPage());
                pages.addPage(URL_TEAMS, new TeamsPage());
                pages.addPage(URL_GAMES, new GamesPage());
                pages.addPage(URL_RESULTS, new ResultsPage());

                // show default page
                pages.showPage(URL_USERS);

                SidebarComponent sideBar = new SidebarComponent();


                sideBar.addNavigationButton("Users", "/user.png", event -> {
                        pages.showPage(URL_USERS);
                });

                sideBar.addNavigationButton("Teams", "/teams.png", event -> {
                        pages.showPage(URL_TEAMS);
                });


                sideBar.addNavigationButton("Games", "/games.png", event -> {
                        pages.showPage(URL_GAMES);
                });


                sideBar.addNavigationButton("Results","/results.png", event -> {
                        pages.showPage(URL_RESULTS);
                });


                BorderPane rootPane = new BorderPane();
                rootPane.setLeft(sideBar);
                rootPane.setCenter(pages);
                rootPane.setBackground(ROOT_BACKGROUND); // main background

                Scene scene = new Scene(rootPane, 500, 500);
                scene.getStylesheets().add(
                                getClass().getResource("/euro-bet-ui.css").toExternalForm());

                primaryStage.setScene(scene);


                primaryStage.setMinWidth(1070);
                primaryStage.setMaxWidth(1070);
                primaryStage.setMinHeight(600);
                primaryStage.setMaxHeight(600);

                primaryStage.setTitle("EURO-Bet-UI");
                primaryStage.show();
        }
}
```

NumberTextField.java

```java
package swe4.gui;

import javafx.scene.control.TextField;

public class NumberTextField extends TextField {

    @Override
    public void replaceText(int start, int end, String text)
    {
        if (validate(text))
        {
            super.replaceText(start, end, text);
        }
    }

    @Override
    public void replaceSelection(String text)
    {
        if (validate(text))
        {
            super.replaceSelection(text);
        }
    }

    private boolean validate(String text)
    {
        return text.matches("[0-9]*");
    }
}
```

ToggleSwitch.java

```java
package swe4.gui;

import javafx.beans.property.SimpleBooleanProperty;
import javafx.geometry.Pos;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.layout.HBox;


public class ToggleSwitch extends HBox {

        private final Label label = new Label();
        private final Button button = new Button();

        private SimpleBooleanProperty switchedOn = new SimpleBooleanProperty(false);
        public SimpleBooleanProperty switchOnProperty() { return switchedOn; }

        private void init() {

                label.setText("OFF");

                getChildren().addAll(label, button);
                button.setOnAction((e) -> {
                        switchedOn.set(!switchedOn.get());
                });
                label.setOnMouseClicked((e) -> {
                        switchedOn.set(!switchedOn.get());
                });
                setStyle();
                bindProperties();
        }

        private void setStyle() {
                // default width
                setWidth(80);
                label.setAlignment(Pos.CENTER);
                setStyle("-fx-background-color: lightgray; "
                                + "-fx-text-fill:black; "
                                + "-fx-background-radius: 0;");
                setAlignment(Pos.CENTER_LEFT);
        }

        private void bindProperties() {
                label.prefWidthProperty().bind(widthProperty().divide(2));
                label.prefHeightProperty().bind(heightProperty());
                button.prefWidthProperty().bind(widthProperty().divide(2));
                button.prefHeightProperty().bind(heightProperty());
        }

        public ToggleSwitch() {
                init();
                switchedOn.addListener((a,b,c) -> {
                        if (c) {
                                switchOn();
                        }
                        else {
                                switchOff();
                        }
                });
        }

        public void switchOn() {
                label.setText("ON");
                setStyle("-fx-background-color: green;");
                label.toFront();
        }

        public void switchOff() {
                label.setText("OFF");
                setStyle("-fx-background-color: lightgray;");
        button.toFront();
        }

        public void switchToStatus(Boolean status) {
                if(status) {
                        switchOn();
                } else {
                        switchOff();
                }
        }

}
```

1.2.2 swe4.gui.components

PanelComponent.java

```
package swe4.gui.components;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Node;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.layout.Border;
import javafx.scene.layout.BorderStroke;
import javafx.scene.layout.BorderStrokeStyle;
import javafx.scene.layout.GridPane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Rectangle;
import javafx.scene.text.Font;
import swe4.gui.helper.GuiHelper;

public abstract class PanelComponent extends GridPane{

        private static final Border BORDER = new Border(
                        new BorderStroke(Color.LIGHTGRAY, BorderStrokeStyle.SOLID, null, null)
                );

        private int numberOfLeftButtons = 0;
        private int numberOfRightButtons = 0;

        public PanelComponent(String title) {

                this.setBorder(BORDER);
                Rectangle header = new Rectangle();
                header.setX(0);
                header.setY(0);
                header.setWidth(400);
                header.setHeight(40);
                header.setFill(Color.CADETBLUE);


        Label panelTitle = new Label(title);
        panelTitle.setFont(Font.font("Helvetia", 20));
        panelTitle.setStyle("-fx-padding: 0px 0px 0px 150px; -fx-text-fill: #ffffff");



        Rectangle content = new Rectangle();
        content.setX(0);
        content.setY(0);
        content.setWidth(400);
        content.setHeight(450);
        content.setFill(Color.WHITE);


        // add all elements to panel

        this.add(header, 0, 0);
        this.add(panelTitle, 0, 0);
        this.add(content, 0, 1);
        }

        protected void addContent(Node component) {
                this.add(component, 0, 1);
        }


        protected void addHeaderButtonLeft(String id,
                        String iconFile, EventHandler<ActionEvent> handler) {
                GuiHelper helper = new GuiHelper();
                Button button = helper.createIconButton(id, iconFile);

                // calculate padding
                String paddingLeft;
                if(numberOfLeftButtons == 0) {
                        paddingLeft = "10";
                } else {
                        paddingLeft = Integer.toString(10 + numberOfLeftButtons * 40);
                }
                numberOfLeftButtons ++;

                // set properties
                button.setStyle("-fx-background-color: transparent; "
                                + "-fx-padding: 0px 0px 0px"+
                                paddingLeft +"px");
                button.addEventHandler(ActionEvent.ACTION, handler);

```

```java
                // add button to header
                this.add(button, 0, 0);

        }



        protected void addHeaderButtonRight(String id, String iconFile,
                        EventHandler<ActionEvent> handler) {
                GuiHelper helper = new GuiHelper();
                Button button = helper.createIconButton(id, iconFile);

                // calculate padding
                String paddingLeft;
                if(numberOfRightButtons == 0) {
                        paddingLeft = "370";
                } else {
                        paddingLeft = Integer.toString(370 - numberOfRightButtons * 40);
                }
                numberOfRightButtons++;

                // set properties
                button.setStyle("-fx-background-color: transparent;"
                                + " -fx-padding: 0px 0px 0px"
                                + paddingLeft +"px");
                button.addEventHandler(ActionEvent.ACTION, handler);

                // add button to header
                this.add(button, 0, 0);

        }



}
```

RouteComponent.java

```java
package swe4.gui.components;

import java.util.HashMap;
import java.util.Map;
import java.util.NoSuchElementException;

import javafx.geometry.Insets;
import javafx.scene.layout.Background;
import javafx.scene.layout.BackgroundFill;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;

```

```java
public class RouteComponent extends Pane {

        private static final Background BACKGROUND = new Background
                        (new BackgroundFill(Color.WHITE, null, null));

        private Map<String, Pane> pages = new HashMap<String, Pane>();

        public RouteComponent() {

                //this.setPadding(new Insets(100));
                this.setBackground(BACKGROUND);
        }


        public void addPage(String url, Pane page) {
                pages.put(url, page);
        }

        public void showPage(String url) {
                Pane page = pages.get(url);

                if(page == null) {
                        System.out.println("Unable to show Page with url: " + url);
                        return;
                }
                page.setPadding(new Insets(50, 0, 0, 50));


                this.getChildren().clear();
                this.getChildren().add(page);
        }



}
```

SidebarComponent.java

```java
package swe4.gui.components;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.layout.Background;
import javafx.scene.layout.BackgroundFill;
import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.scene.shape.HLineTo;
import javafx.scene.shape.MoveTo;
import javafx.scene.shape.Path;
import swe4.gui.helper.GuiHelper;

public class SidebarComponent extends Pane {


        private static final Background BACKGROUND = new Background
                        (new BackgroundFill(Color.web("0x21252b",1.0), null, null));

        private VBox navigationButtonsPane;


        public SidebarComponent() {

                // set background
                this.setBackground(BACKGROUND);

                // create panes
                Pane userPane = createUserPane();
                navigationButtonsPane = new VBox();
                navigationButtonsPane.setSpacing(20);
                navigationButtonsPane.setPadding(new Insets(100, 20, 0, 25));

                // create horizontal line
                Path path = new Path();
        path.getElements().add(new MoveTo(0.0f, 80.0f));
        path.getElements().add(new HLineTo(145.0f));
        path.setStroke(Color.WHITE);

        // add all elements
                this.getChildren().addAll(userPane, path, navigationButtonsPane);
        }


        private Pane createUserPane() {

                Label currentUserText = new Label("Welcome,\n Admin");
                currentUserText.setId("nav-welcome");

                Image currentUserImage = new Image("/avatar.png");

                ImageView currentUserAvatar = new ImageView();
                currentUserAvatar.setImage(currentUserImage);


                HBox userPane = new HBox(currentUserAvatar, currentUserText);
                userPane.setSpacing(5);
                userPane.setAlignment(Pos.CENTER);
                userPane.setPadding(new Insets(15,15,0,15));

                return userPane;
        }



        public void addNavigationButton(String title, String iconFile,
                        EventHandler<ActionEvent> handler) {

                GuiHelper helper = new GuiHelper();
                Button button = helper.createTextIconButton("nav-button", title, iconFile);
                button.addEventHandler(ActionEvent.ACTION, handler);

                navigationButtonsPane.getChildren().add(button);
        }



}
```

1.2.3 swe4.gui.helper

GuiHelper.java

```
package swe4.gui.helper;

import javafx.geometry.Insets;
import javafx.scene.control.Button;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import swe4.models.Game;

public class GuiHelper {

        public Button createIconButton(String id, String iconFile) {
                Button button = new Button();
                button.setId(id);
                button.setPadding(new Insets(10));
                Image icon = new Image(getClass().getResourceAsStream(iconFile));
                button.setGraphic(new ImageView(icon));

                return button;
        }

        public Button createTextButton(String id, String caption) {
                Button button = new Button(caption);
                button.setId(id);
```

```java
                button.setPadding(new Insets(10));
                button.setPrefSize(50, 40);
                return button;
        }

        public Button createTextIconButton(String id, String caption, String iconFile) {
                Button button = new Button(caption);
                button.setId(id);
                button.setPadding(new Insets(10));
                Image icon = new Image(getClass().getResourceAsStream(iconFile));
                button.setGraphic(new ImageView(icon));

                return button;
        }


}
```

1.2.4 swe4.gui.pages

GamesPage.java

```java
package swe4.gui.pages;

import javafx.geometry.Insets;
import javafx.scene.layout.Background;
import javafx.scene.layout.BackgroundFill;
import javafx.scene.layout.FlowPane;
import javafx.scene.paint.Color;
import swe4.gui.panels.GameDetailPanel;
import swe4.gui.panels.GamesPanel;
import swe4.gui.panels.GamesPanelListener;
import swe4.gui.panels.TeamDetailPanel;
import swe4.gui.panels.TeamsPanelListener;
import swe4.models.Game;
import swe4.models.Team;


public class GamesPage extends FlowPane {

        private static final Background BACKGROUND = new Background
                        (new BackgroundFill(Color.WHITE, null, null));

        public GamesPage() {

                this.setBackground(BACKGROUND);


                this.setPadding(new Insets(5, 0, 5, 0));
                this.setVgap(20);
                this.setHgap(20);
                this.setPrefWidth(1300);


                // all users panel
                GamesPanel allGames = new GamesPanel();

                // add detail panel
        GameDetailPanel detailGame = new GameDetailPanel();

                // handle listeners
                allGames.addListener(new GamesPanelListener() {

                        @Override
                        public void onGameSelected(Game game) {
                                detailGame.setDetails(game);
                        }

                        @Override
                        public void onAddNewGame() {
                                detailGame.createNewDetail();
                        }
                });


        this.getChildren().addAll(allGames, detailGame);
        }




}
```

ResultsPage.java

```
package swe4.gui.pages;

import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.geometry.Insets;
import javafx.scene.control.Label;
import javafx.scene.control.ListView;
import javafx.scene.layout.Background;
import javafx.scene.layout.BackgroundFill;
import javafx.scene.layout.Border;
import javafx.scene.layout.BorderStroke;
import javafx.scene.layout.BorderStrokeStyle;
```

```java
import javafx.scene.layout.FlowPane;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.scene.shape.HLineTo;
import javafx.scene.shape.MoveTo;
import javafx.scene.shape.Path;
import javafx.scene.shape.Rectangle;
import javafx.scene.shape.VLineTo;
import swe4.gui.panels.GamesPanelListener;
import swe4.gui.panels.ResultDetailPanel;
import swe4.gui.panels.ResultsPanel;
import swe4.gui.panels.ResultsPanelListener;
import swe4.gui.panels.UserDetailPanel;
import swe4.gui.panels.UsersPanel;
import swe4.models.Game;


public class ResultsPage extends FlowPane {


        private static final Background BACKGROUND = new Background
                        (new BackgroundFill(Color.WHITE, null, null));



        public ResultsPage() {

                this.setBackground(BACKGROUND);


                this.setPadding(new Insets(5, 0, 5, 0));
                this.setVgap(20);
                this.setHgap(20);
                //this.setPrefWrapLength(1300); // preferred width allows for two columns
                this.setPrefWidth(1300);


                // all users panel
                ResultsPanel allResults = new ResultsPanel();

                // detail panel
                ResultDetailPanel detailResult = new ResultDetailPanel();

                // handle listeners
                allResults.addListener(new ResultsPanelListener() {

                        @Override
                        public void onResultSelected(Game game) {
                                detailResult.setDetails(game);
                        }
```

```

                                @Override
                                public void onAddNewResult() {
                                        detailResult.createNewDetail();

                                }
                        });


                this.getChildren().addAll(allResults, detailResult);
                }




}
```

TeamsPage.java

```
package swe4.gui.pages;

import javafx.geometry.Insets;
import javafx.scene.layout.Background;
import javafx.scene.layout.BackgroundFill;
import javafx.scene.layout.FlowPane;
import javafx.scene.paint.Color;
import swe4.gui.panels.TeamDetailPanel;
import swe4.gui.panels.TeamsPanel;
import swe4.gui.panels.TeamsPanelListener;
import swe4.models.Team;


public class TeamsPage extends FlowPane {


        private static final Background BACKGROUND = new Background
                        (new BackgroundFill(Color.WHITE, null, null));


        public TeamsPage() {

                this.setBackground(BACKGROUND);


                this.setPadding(new Insets(5, 0, 5, 0));
                this.setVgap(20);
                this.setHgap(20);
                //this.setPrefWrapLength(1300); // preferred width allows for two columns
                this.setPrefWidth(1300);

```

```java

                // all users panel
                TeamsPanel allTeams = new TeamsPanel();

                // add detail panel
        TeamDetailPanel detailTeam = new TeamDetailPanel();

        // handle listeners
        allTeams.addListener(new TeamsPanelListener() {

                        @Override
                        public void onTeamSelected(Team team) {
                                detailTeam.setDetails(team);
                        }

                        @Override
                        public void onAddNewTeam() {
                                detailTeam.createNewDetail();
                        }
                });


        this.getChildren().addAll(allTeams, detailTeam);
        }




}
```

UsersPage.java

```java
package swe4.gui.pages;

import javafx.geometry.Insets;
import javafx.scene.layout.Background;
import javafx.scene.layout.BackgroundFill;
import javafx.scene.layout.FlowPane;
import javafx.scene.paint.Color;
import swe4.gui.panels.UserDetailPanel;
import swe4.gui.panels.UsersPanel;
import swe4.gui.panels.UsersPanelListener;
import swe4.models.User;


public class UsersPage extends FlowPane {


        private static final Background BACKGROUND = new Background
                        (new BackgroundFill(Color.WHITE, null, null));


        public UsersPage() {

                this.setBackground(BACKGROUND);


                this.setPadding(new Insets(5, 0, 5, 0));
                this.setVgap(20);
                this.setHgap(20);
                //this.setPrefWrapLength(1300); // preferred width allows for two columns
                this.setPrefWidth(1300);


                // all users panel
                UsersPanel allUsers = new UsersPanel();


        // add detail panel
        UserDetailPanel detailUser = new UserDetailPanel();


        // handle listeners
        allUsers.addListener(new UsersPanelListener() {

                        @Override
                        public void onUserSelected(User user) {
                                detailUser.setDetails(user);
                        }

                        @Override
                        public void onAddNewUser() {
                                detailUser.createNewDetail();
                        }
                });



        this.getChildren().addAll(allUsers, detailUser);
        }
}
```

1.2.5 swe4.gui.panels

GameDetailPanel.java

```
package swe4.gui.panels;

import java.util.ArrayList;
```

```java
import java.util.List;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.geometry.Insets;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import swe4.gui.ToggleSwitch;
import swe4.gui.components.PanelComponent;
import swe4.models.Game;
import swe4.models.Team;
import swe4.models.User;

public class GameDetailPanel extends PanelComponent {


        private ComboBox<String> teamABox;
        private ComboBox<String> teamBBox;

        ArrayList<Team> teams = new ArrayList<Team>();

        public GameDetailPanel() {
                super("Details");

                addHeaderButtonRight("game-save-button", "/save.png", event -> {
                        System.out.println("game-save-button: "+ event.getSource());
                });

                createDetailFields();
        }

        private void createDetailFields() {

                VBox details = new VBox();
                details.setSpacing(50);
                details.setId("game-detail-container");



                // create demo teams
                List<Game> games = new ArrayList<>();

        Team t1 = new Team("Austria", 'G');
        Team t2 = new Team("Spain", 'G');
        Team t3 = new Team("Island", 'G');
        Team t4 = new Team("Portugal", 'A');


                // team a
                VBox groupA = new VBox();
                groupA.setSpacing(3);
                Label groupATitle = new Label("Team A:");
                ObservableList<String> optionsA =
                            FXCollections.observableArrayList(
                                        t1.getName(),
                                        t2.getName(),
                                        t3.getName(),
                                        t4.getName()
                            );
                teamABox = new ComboBox<String>(optionsA);
                teamABox.setMinWidth(340);
                groupA.getChildren().addAll(groupATitle, teamABox);
                VBox.setMargin(groupA, new Insets(60, 30, 0, 30));



                // team b
                VBox groupB = new VBox();
                groupB.setSpacing(3);
                Label groupBTitle = new Label("Team B:");
                ObservableList<String> optionsB =
                            FXCollections.observableArrayList(
                                t1.getName(),
                                t2.getName(),
                                t3.getName(),
                                t4.getName()
                            );
                teamBBox = new ComboBox<String>(optionsB);
                teamBBox.setMinWidth(340);
                groupB.getChildren().addAll(groupBTitle, teamBBox);
                VBox.setMargin(groupB, new Insets(0, 30, 0, 30));


                details.getChildren().addAll(
                                 groupA, groupB
                );


                addContent(details);

        }

        public void setDetails(Game game) {
                if(game == null) { return; }
                teamABox.setValue(game.getTeamA().getName());
                teamBBox.setValue(game.getTeamB().getName());
        }

        public void createNewDetail() {
                teamABox.setValue("");
                teamBBox.setValue("");
        }

```

```

}
```

GamesPanel.java

```
package swe4.gui.panels;

import java.util.ArrayList;
import java.util.List;

import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.scene.control.ListCell;
import javafx.scene.control.ListView;
import javafx.scene.layout.Border;
import javafx.scene.layout.BorderStroke;
import javafx.scene.layout.BorderStrokeStyle;
import javafx.scene.paint.Color;
import javafx.util.Callback;
import swe4.gui.components.PanelComponent;
import swe4.models.Game;
import swe4.models.Team;


public class GamesPanel extends PanelComponent{

        private static final Border LIST_BORDER = new Border(
                        new BorderStroke(Color.TRANSPARENT, BorderStrokeStyle.NONE, null, null)
                );


        private List<GamesPanelListener> listeners = new ArrayList<GamesPanelListener>();

        public GamesPanel() {
                super("All Games");


                addHeaderButtonLeft("games-add-button", "/plus.png", event -> {
                        // System.out.println("games-add-button: "+ event.getSource());

                        // call all added listeners
            for (GamesPanelListener l : listeners) {
                l.onAddNewGame();
            }
                });


                createGamesList();
        }


        private void createGamesList() {
                ListView<Game> gamesListView = new ListView<Game>();
                gamesListView.setBorder(LIST_BORDER);



        // create demo teams
                List<Game> games = new ArrayList<>();

        Team t1 = new Team("Austria", 'G');
        Team t2 = new Team("Spain", 'G');
        Team t3 = new Team("Island", 'G');
        Team t4 = new Team("Portugal", 'A');

        // create demo games
        Game g1 = new Game(t1, t2);
        Game g2 = new Game(t2, t3);
        Game g3 = new Game(t3, t4);
        Game g4 = new Game(t1, t4);
        games.add(g1);
        games.add(g2);
        games.add(g3);
        games.add(g4);


        ObservableList<Game> items = FXCollections.observableList(games);
        gamesListView.setItems(items);


        gamesListView.setCellFactory(new Callback<ListView<Game>, ListCell<Game>>(){

            @Override
            public ListCell<Game> call(ListView<Game> p) {

                ListCell<Game> cell = new ListCell<Game>(){

                    @Override
                    protected void updateItem(Game t, boolean bln) {
                        super.updateItem(t, bln);
                        if (t != null) {
                                setText(t.getTeamA().getName() + " - " + t.getTeamB().getName());
                        }
                    }
                };
                return cell;
            }
        });

        gamesListView.getSelectionModel().selectedItemProperty()
                .addListener(new ChangeListener<Game>() {
```

```

                @Override
                public void changed(ObservableValue<? extends Game> observable,
                                Game oldValue, Game newValue) {

                // call all added listeners
                for (GamesPanelListener l : listeners) {
                    l.onGameSelected(newValue);
                }
                }
        });

        addContent(gamesListView);
        }


        public void addListener(GamesPanelListener listener) {
                listeners.add(listener);
    }


}
```

GamesPanelListener.java

```
package swe4.gui.panels;

import swe4.models.Game;

public interface GamesPanelListener {
    void onGameSelected(Game game);
    void onAddNewGame();
}
```

ResultDetailPanel.java

```
package swe4.gui.panels;

import java.util.ArrayList;
import java.util.List;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.geometry.Insets;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
```

```java
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import swe4.gui.NumberTextField;
import swe4.gui.ToggleSwitch;
import swe4.gui.components.PanelComponent;
import swe4.gui.helper.GuiHelper;
import swe4.models.Game;
import swe4.models.Team;
import swe4.models.User;

public class ResultDetailPanel extends PanelComponent {


        private NumberTextField goalsTeamABox;
        private NumberTextField goalsTeamBBox;

        ArrayList<Team> teams = new ArrayList<Team>();
        private ComboBox<String> gameBox;

        public ResultDetailPanel() {
                super("Details");

                addHeaderButtonRight("result-save-button", "/save.png", event -> {
                        System.out.println("result-save-button: "+ event.getSource());
                });

                createDetailFields();
        }

        private void createDetailFields() {

                VBox details = new VBox();
                details.setSpacing(50);
                details.setId("result-detail-container");





        // create demo teams
                List<Game> games = new ArrayList<>();
        Team t1 = new Team("Austria", 'G');
        Team t2 = new Team("Spain", 'G');
        Team t3 = new Team("Island", 'G');
        Team t4 = new Team("Portugal", 'A');

        // create demo games which are finished
        Game g1 = new Game(t1, t2);
        g1.setGameFinished(true);
        g1.setGoalsTeamA(2);
        g1.setGoalsTeamB(0);

        Game g2 = new Game(t2, t3);
        g2.setGameFinished(true);
        g2.setGoalsTeamA(4);
        g2.setGoalsTeamB(1);

        Game g3 = new Game(t3, t4);
        g3.setGameFinished(true);
        g3.setGoalsTeamA(0);
        g3.setGoalsTeamB(3);

        Game g4 = new Game(t1, t4);
        g4.setGameFinished(true);
        g4.setGoalsTeamA(2);
        g4.setGoalsTeamB(2);

        games.add(g1);
        games.add(g2);
        games.add(g3);
        games.add(g4);



        // game
        VBox game = new VBox();
                game.setSpacing(3);
                Label gameTitle = new Label("Game:");
                ObservableList<String> optionsB =
                            FXCollections.observableArrayList(
                                g1.toString(),
                                g2.toString(),
                                g3.toString(),
                                g4.toString()
                            );
                gameBox = new ComboBox<String>(optionsB);
                gameBox.setMinWidth(340);
                game.getChildren().addAll(gameTitle, gameBox);
                VBox.setMargin(game, new Insets(30, 30, 0, 30));


                // goals team a
        VBox goalsA = new VBox();
        goalsA.setSpacing(3);
                Label goalsATitle = new Label("Goals Team A:");
                goalsTeamABox = new NumberTextField ();
                goalsA.getChildren().addAll(goalsATitle, goalsTeamABox);
                VBox.setMargin(goalsA, new Insets(0, 30, 0, 30));


                // goals team b
        VBox goalsB = new VBox();
        goalsB.setSpacing(3);
                Label goalsBTitle = new Label("Goals Team B:");
                goalsTeamBBox = new NumberTextField ();
                goalsA.getChildren().addAll(goalsBTitle, goalsTeamBBox);
                VBox.setMargin(goalsB, new Insets(0, 30, 0, 30));

                details.getChildren().addAll(
                                game,goalsA, goalsB
                );

                addContent(details);

        }

        public void setDetails(Game game) {
                if(game == null) { return; }

                gameBox.setValue(game.toString());
                goalsTeamABox.setText(game.getGoalsTeamA().toString());
                goalsTeamBBox.setText(game.getGoalsTeamB().toString());
        }

        public void createNewDetail() {
                gameBox.setValue("");
                goalsTeamABox.setText("");
                goalsTeamBBox.setText("");
        }


}
```

ResultsPanel.java

```
package swe4.gui.panels;

import java.util.ArrayList;
import java.util.List;

import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.scene.control.ListCell;
import javafx.scene.control.ListView;
import javafx.scene.layout.Border;
import javafx.scene.layout.BorderStroke;
import javafx.scene.layout.BorderStrokeStyle;
import javafx.scene.paint.Color;
import javafx.util.Callback;
import swe4.gui.components.PanelComponent;
import swe4.gui.helper.GuiHelper;
import swe4.models.Game;
import swe4.models.Team;
```

```java


public class ResultsPanel extends PanelComponent{

        private static final Border LIST_BORDER = new Border(
                        new BorderStroke(Color.TRANSPARENT, BorderStrokeStyle.NONE, null, null)
                );


        private List<ResultsPanelListener> listeners = new ArrayList<ResultsPanelListener>();

        public ResultsPanel() {
                super("All Results");

                addHeaderButtonLeft("results-add-button", "/plus.png", event -> {
                        // System.out.println("results-add-button: "+ event.getSource());
                        // call all added listeners
            for (ResultsPanelListener l : listeners) {
                l.onAddNewResult();
            }
                });


                createGamesList();
        }


        private void createGamesList() {
                ListView<Game> gamesListView = new ListView<Game>();
                gamesListView.setBorder(LIST_BORDER);

        List<Game> games = new ArrayList<>();

        // create demo teams
        Team t1 = new Team("Austria", 'G');
        Team t2 = new Team("Spain", 'G');
        Team t3 = new Team("Island", 'G');
        Team t4 = new Team("Portugal", 'A');

        // create demo games which are finished
        Game g1 = new Game(t1, t2);
        g1.setGameFinished(true);
        g1.setGoalsTeamA(2);
        g1.setGoalsTeamB(0);

        Game g2 = new Game(t2, t3);
        g2.setGameFinished(true);
        g2.setGoalsTeamA(4);
        g2.setGoalsTeamB(1);

        Game g3 = new Game(t3, t4);
        g3.setGameFinished(true);
        g3.setGoalsTeamA(0);
        g3.setGoalsTeamB(3);

        Game g4 = new Game(t1, t4);
        g4.setGameFinished(true);
        g4.setGoalsTeamA(2);
        g4.setGoalsTeamB(2);

        games.add(g1);
        games.add(g2);
        games.add(g3);
        games.add(g4);


        ObservableList<Game> items = FXCollections.observableList(games);
        gamesListView.setItems(items);


        gamesListView.setCellFactory(new Callback<ListView<Game>, ListCell<Game>>(){

            @Override
            public ListCell<Game> call(ListView<Game> p) {

                ListCell<Game> cell = new ListCell<Game>(){

                    @Override
                    protected void updateItem(Game g, boolean bln) {
                        super.updateItem(g, bln);
                        if (g != null) {
                                setText(g.getResult() + "    " + g.toString());
                        }
                    }
                };
                return cell;
            }
        });

        gamesListView.getSelectionModel().selectedItemProperty()
                .addListener(new ChangeListener<Game>() {

                @Override
                public void changed(ObservableValue<? extends Game> observable,
                                Game oldValue, Game newValue) {

                // call all added listeners
                for (ResultsPanelListener l : listeners) {
                    l.onResultSelected(newValue);
                }
                }
        });

        addContent(gamesListView);
        }

```

```java

        public void addListener(ResultsPanelListener listener) {
                listeners.add(listener);
    }


}
```

ResultsPanelListener.java

```java
package swe4.gui.panels;

import swe4.models.Game;

public interface ResultsPanelListener {
    void onResultSelected(Game game);
    void onAddNewResult();
}
```

TeamDetailPanel.java

```java
package swe4.gui.panels;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.geometry.Insets;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import swe4.gui.ToggleSwitch;
import swe4.gui.components.PanelComponent;
import swe4.models.Team;
import swe4.models.User;

public class TeamDetailPanel extends PanelComponent {


        private TextField nameField;
        private ComboBox<String> groupBox;
        private final String
                GROUP_A = "GROUPE A", GROUP_B = "GROUPE B", GROUP_C = "GROUPE C",
                GROUP_D = "GROUPE D", GROUP_E = "GROUPE E", GROUP_F = "GROUPE F",
                GROUP_G = "GROUPE G";

        public TeamDetailPanel() {
                super("Details");

                addHeaderButtonRight("team-save-button", "/save.png", event -> {
                        System.out.println("team-save-button: "+ event.getSource());
                });

                createDetailFields();
        }

        private void createDetailFields() {

                VBox details = new VBox();
                details.setSpacing(20);
                details.setId("team-detail-container");

                //details.setPadding(new Insets(100));





                // team name
                VBox username = new VBox();
                username.setSpacing(3);
                Label nameTitle = new Label("Country:");
                nameField = new TextField ();
                username.getChildren().addAll(nameTitle, nameField);
                VBox.setMargin(username, new Insets(30, 30, 0, 30));


                // team group
                VBox group = new VBox();
                group.setSpacing(3);
                Label groupTitle = new Label("Group:");
                ObservableList<String> options =
                            FXCollections.observableArrayList(
                                GROUP_A,
                                GROUP_B,
                                GROUP_C,
                                GROUP_D,
                                GROUP_E,
                                GROUP_F,
                                GROUP_G
                            );
                groupBox = new ComboBox<String>(options);
                group.getChildren().addAll(groupTitle, groupBox);
                VBox.setMargin(group, new Insets(0, 30, 0, 30));


                details.getChildren().addAll(
                                 username, group
                );


                addContent(details);

        }

        public void setDetails(Team team) {
                if(team == null) { return; }

                nameField.setText(team.getName());

                if(team.getGroup() == 'A') {
                        groupBox.setValue(GROUP_A);

                } else if(team.getGroup() == 'B') {
                        groupBox.setValue(GROUP_B);

                } else if(team.getGroup() == 'C') {
                        groupBox.setValue(GROUP_C);

                } else if(team.getGroup() == 'D') {
                        groupBox.setValue(GROUP_D);

                } else if(team.getGroup() == 'E') {
                        groupBox.setValue(GROUP_E);

                } else if(team.getGroup() == 'F') {
                        groupBox.setValue(GROUP_F);

                } else if(team.getGroup() == 'G') {
                        groupBox.setValue(GROUP_G);
                }

        }

        public void createNewDetail() {
                nameField.setText("");
                groupBox.setValue("");

        }


}
```

TeamsPanel.java

```
package swe4.gui.panels;

import java.util.ArrayList;
import java.util.List;

import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.scene.control.ListCell;
import javafx.scene.control.ListView;
import javafx.scene.layout.Border;
import javafx.scene.layout.BorderStroke;
import javafx.scene.layout.BorderStrokeStyle;
import javafx.scene.paint.Color;
import javafx.util.Callback;
import swe4.gui.components.PanelComponent;
import swe4.models.Team;


public class TeamsPanel extends PanelComponent{

        private static final Border LIST_BORDER = new Border(
                        new BorderStroke(Color.TRANSPARENT, BorderStrokeStyle.NONE, null, null)
                );


        private List<TeamsPanelListener> listeners = new ArrayList<TeamsPanelListener>();

        public TeamsPanel() {
                super("All Teams");


                addHeaderButtonLeft("teams-add-button", "/plus.png", event -> {
                        // call all added listeners
            for (TeamsPanelListener l : listeners) {
                l.onAddNewTeam();
            }

                });


                createTeamsList();
        }


        private void createTeamsList() {
                ListView<Team> teamListView = new ListView<Team>();
                teamListView.setBorder(LIST_BORDER);

        List<Team> teams = new ArrayList<>();

        // create demo teams
        Team t1 = new Team("Austria", ’G’);
        teams.add(t1);

        Team t2 = new Team("Spain", ’G’);
```

```java
        teams.add(t2);

        Team t3 = new Team("Island", 'G');
        teams.add(t3);

        Team t4 = new Team("Portugal", 'A');
        teams.add(t4);



        ObservableList<Team> items = FXCollections.observableList(teams);


        teamListView.setItems(items);


        teamListView.setCellFactory(new Callback<ListView<Team>, ListCell<Team>>(){

            @Override
            public ListCell<Team> call(ListView<Team> p) {

                ListCell<Team> cell = new ListCell<Team>(){

                    @Override
                    protected void updateItem(Team t, boolean bln) {
                        super.updateItem(t, bln);
                        if (t != null) {
                                setText(t.getName());
                        }
                    }
                };
                return cell;
            }
        });

        teamListView.getSelectionModel().selectedItemProperty()
                                .addListener(new ChangeListener<Team>() {

                @Override
                public void changed(ObservableValue<? extends Team> observable,
                                Team oldValue, Team newValue) {

                // call all added listeners
                for (TeamsPanelListener l : listeners) {
                    l.onTeamSelected(newValue);
                }
                }
        });

        addContent(teamListView);
        }


        public void addListener(TeamsPanelListener listener) {
                listeners.add(listener);
    }


}
```

TeamsPanelListener.java

```
package swe4.gui.panels;

import swe4.models.Team;

public interface TeamsPanelListener {
    void onTeamSelected(Team team);
    void onAddNewTeam();
}
```

UserDetailPanel.java

```
package swe4.gui.panels;

import javafx.geometry.Insets;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import swe4.gui.ToggleSwitch;
import swe4.gui.components.PanelComponent;
import swe4.models.User;

public class UserDetailPanel extends PanelComponent {

        private ToggleSwitch switchActiv;
        private TextField nameField;
        private PasswordField passwordField;
        private PasswordField confirmField;

        public UserDetailPanel() {
                super("Details");

                addHeaderButtonRight("users-save-button", "/save.png", event -> {
                        System.out.println("users-save-button: "+ event.getSource());
                });

                createDetailFields();
```

```java
    }

    private void createDetailFields() {

        VBox details = new VBox();
        details.setSpacing(20);
        details.setId("user-detail-container");

        //details.setPadding(new Insets(100));



        // active
        VBox active = new VBox();
        Label activTitle = new Label("Active:");
        switchActiv = new ToggleSwitch();
        VBox switchPane = new VBox();
        switchPane.getChildren().add(switchActiv);
        VBox.setMargin(switchPane, new Insets(-20, 0, 0, 250));
        active.getChildren().addAll(activTitle, switchPane);
        VBox.setMargin(active, new Insets(20, 30, 0, 30));


        // username
        VBox username = new VBox();
        username.setSpacing(3);
        Label nameTitle = new Label("Username:");
        nameField = new TextField ();
        username.getChildren().addAll(nameTitle, nameField);
        VBox.setMargin(username, new Insets(0, 30, 0, 30));


        // password
        VBox password = new VBox();
        password.setSpacing(3);
        Label passwordTitle = new Label("Password:");
        passwordField = new PasswordField ();
        Label confirmTitle = new Label("Confirm:");
        confirmField = new PasswordField ();
        password.getChildren().addAll(passwordTitle, passwordField,
                confirmTitle, confirmField);
        VBox.setMargin(password, new Insets(0, 30, 0, 30));

        details.getChildren().addAll(
                active, username, password
        );


        addContent(details);

    }

    public void setDetails(User user) {
```

```
                if(user == null) { return; }
                switchActiv.switchToStatus(user.getActive());
                nameField.setText(user.getName());
                passwordField.setText(user.getPassword());
                confirmField.setText(user.getPassword());
        }

        public void createNewDetail() {
                switchActiv.switchToStatus(false);
                nameField.setText("");
                passwordField.setText("");
                confirmField.setText("");
        }


}
```

UsersPanel.java

```
package swe4.gui.panels;

import java.util.ArrayList;
import java.util.List;

import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.EventHandler;
import javafx.scene.control.ListCell;
import javafx.scene.control.ListView;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.Border;
import javafx.scene.layout.BorderStroke;
import javafx.scene.layout.BorderStrokeStyle;
import javafx.scene.paint.Color;
import javafx.util.Callback;
import swe4.gui.components.PanelComponent;
import swe4.models.User;


public class UsersPanel extends PanelComponent{

        private static final Border LIST_BORDER = new Border(
                        new BorderStroke(Color.TRANSPARENT, BorderStrokeStyle.NONE, null, null)
                );


        private List<UsersPanelListener> listeners = new ArrayList<UsersPanelListener>();

        public UsersPanel() {
                super("All Users");

                addHeaderButtonLeft("users-add-button", "/plus.png", event -> {
                        // call all added listeners
            for (UsersPanelListener l : listeners) {
                l.onAddNewUser();
            }
                });
                createUsersList();
        }

        private void createUsersList() {
                ListView<User> personListView = new ListView<User>();
        personListView.setBorder(LIST_BORDER);

        List<User> users = new ArrayList<>();

        // create demo users
        User u1 = new User(true, "Robert Wurm", "1234");
        users.add(u1);

        User u2 = new User(true, "Administrator", "1234");
        users.add(u2);

        User u3 = new User(false, "Max Mustermann", "1234");
        users.add(u3);

        User u4 = new User(true, "Gertraud Eberharter", "1234");
        users.add(u4);



        ObservableList<User> items = FXCollections.observableList(users);


        personListView.setItems(items);


        personListView.setCellFactory(new Callback<ListView<User>, ListCell<User>>(){

            @Override
            public ListCell<User> call(ListView<User> p) {

                ListCell<User> cell = new ListCell<User>(){

                    @Override
                    protected void updateItem(User t, boolean bln) {
                        super.updateItem(t, bln);
                        if (t != null) {
                                setText(t.getName());
                        }
                    }
```

```java
                };
                return cell;
            }
        });

        personListView.getSelectionModel().selectedItemProperty()
                                .addListener(new ChangeListener<User>() {

                @Override
                public void changed(ObservableValue<? extends User> observable,
                                User oldValue, User newValue) {

                // call all added listeners
                for (UsersPanelListener l : listeners) {
                    l.onUserSelected(newValue);
                }
                }
        });

        addContent(personListView);
        }


        public void addListener(UsersPanelListener listener) {
                listeners.add(listener);
    }


}
```

UsersPanelListener.java

```java
package swe4.gui.panels;

import swe4.models.User;

public interface UsersPanelListener {
    void onUserSelected(User user);
    void onAddNewUser();
}
```

1.2.6 swe4.models

Game.java

```java
package swe4.models;

public class Game {

        private Team teamA;
        private Team teamB;

        private Boolean gameFinished;
        private Integer goalsTeamA;
        private Integer goalsTeamB;


        public Game() {}
        public Game(Team teamA, Team teamB) {
                this.setTeamA(teamA);
                this.setTeamB(teamB);
        }
        public Team getTeamA() {
                return teamA;
        }
        public void setTeamA(Team teamA) {
                this.teamA = teamA;
        }
        public Team getTeamB() {
                return teamB;
        }
        public void setTeamB(Team teamB) {
                this.teamB = teamB;
        }

        public String getResult() {
                if(getGameFinished() && goalsTeamA != null && goalsTeamB != null) {
                        return getGoalsTeamA().toString() + " : " + getGoalsTeamB().toString();
                }
                return "not played yet";
        }

        public Integer getGoalsTeamA() {
                return goalsTeamA;
        }

        public void setGoalsTeamA(Integer goalsTeamA) {
                this.goalsTeamA = goalsTeamA;
        }

        public Integer getGoalsTeamB() {
                return goalsTeamB;
        }

        public void setGoalsTeamB(Integer goalsTeamB) {
                this.goalsTeamB = goalsTeamB;
        }
        public Boolean getGameFinished() {
                return gameFinished;
        }
        public void setGameFinished(Boolean gameFinished) {
                this.gameFinished = gameFinished;
        }

        @Override
        public String toString() {
                return this.teamA.getName() + " - " + this.teamB.getName().toString();
        }

}
```

Team.java

```java
package swe4.models;

public class Team {

        private String name;
        private Character group;


        public Team() {

        }

        public Team(String name, Character group) {
                this.setName(name);
                this.setGroup(group);
        }

        public Character getGroup() {
                return group;
        }

        public void setGroup(Character group) {
                this.group = group;
        }

        public String getName() {
                return name;
        }

        public void setName(String name) {
                this.name = name;
        }
}
```

User.java

```java
package swe4.models;

public class User {

        private Boolean active;
        private String name;
        private String password;


        public User() {

        }

        public User(Boolean active, String name, String password) {
                this.active = active;
                this.name = name;
                this.password = password;
        }


        public Boolean getActive() {
                return active;
        }
        public void setActive(Boolean active) {
                this.active = active;
        }
        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        public String getPassword() {
                return password;
        }
        public void setPassword(String password) {
                this.password = password;
        }





}
```

1.2.7 Style

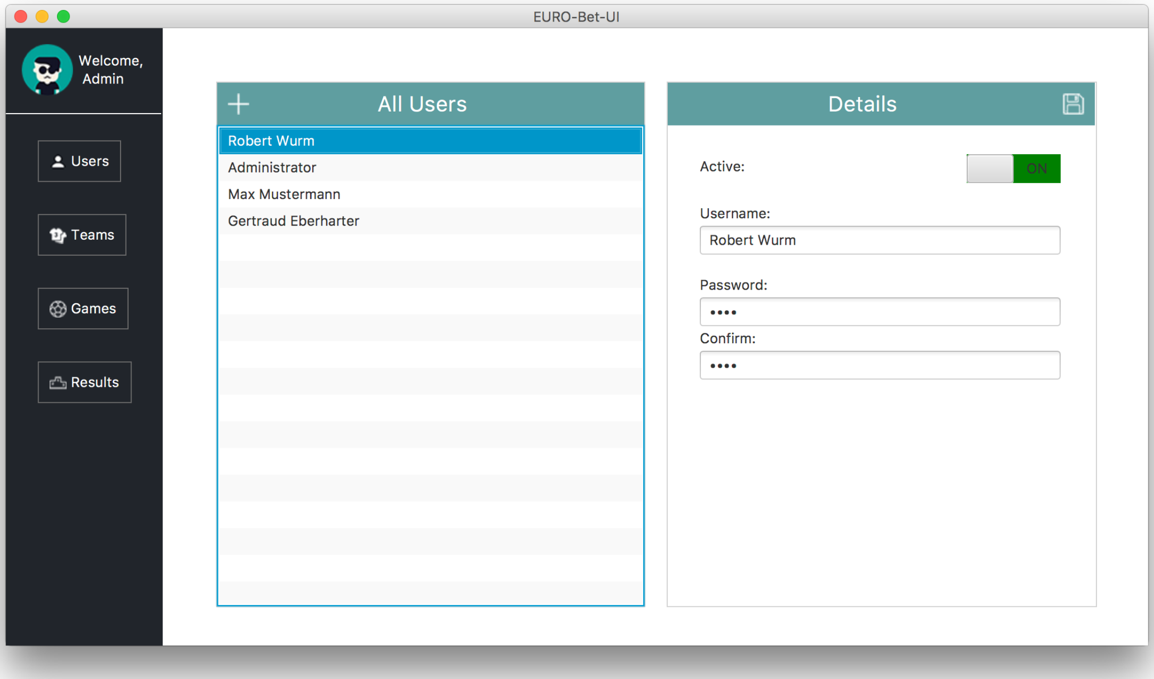
euro-bet-ui.css

```
#nav-welcome {
        -fx-text-fill: #ffffff;
}

#nav-button {
        -fx-border-color: dimgrey;
        -fx-background-color: transparent;
        -fx-text-fill: #ffffff;
        -fx-border-insets: 5px;
}

#nav-button:hover{
    -fx-background-color: #282c34;
}


#user-detail-container {
        -fx-padding: 10.0 3.0 5.0 0.0;
}

#control-pane {
        -fx-spacing: 5;
        -fx-padding: 10,20;
        -fx-border-color: dimgrey;
}
```

1.3 Testfälle

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

Users



Robert Wurm
Administrator
Max Mustermann
Gertraud Eberharter

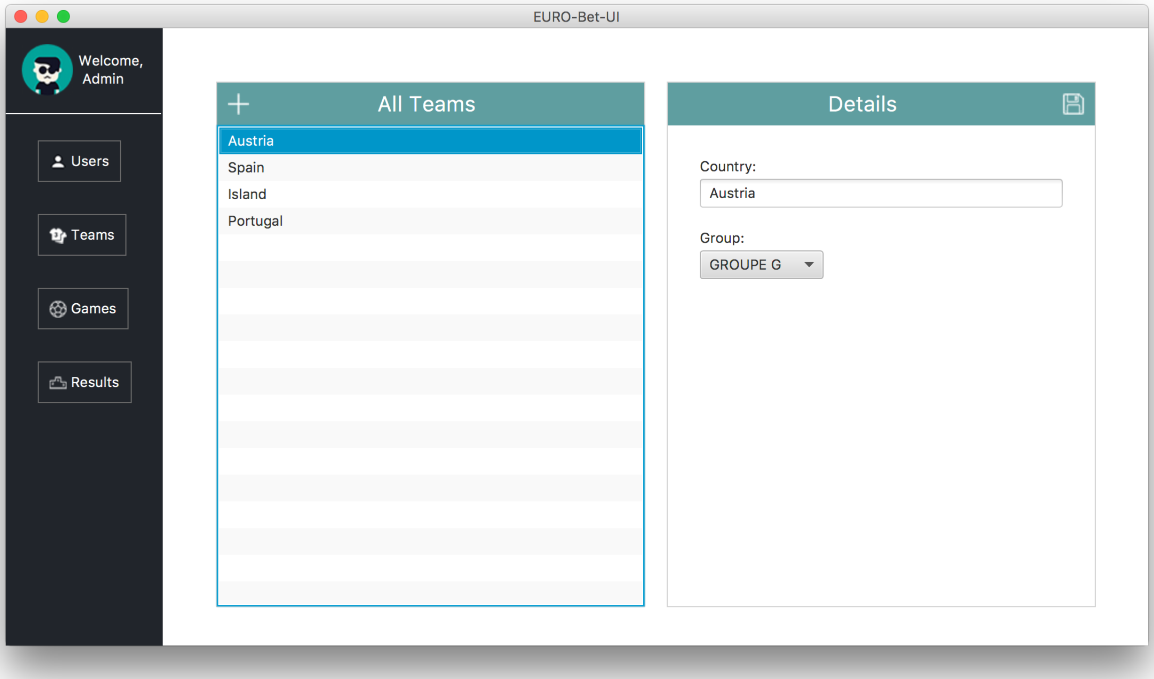
Active: ON

Username: Robert Wurm

Password: ••••

Confirm: ••••

Teams

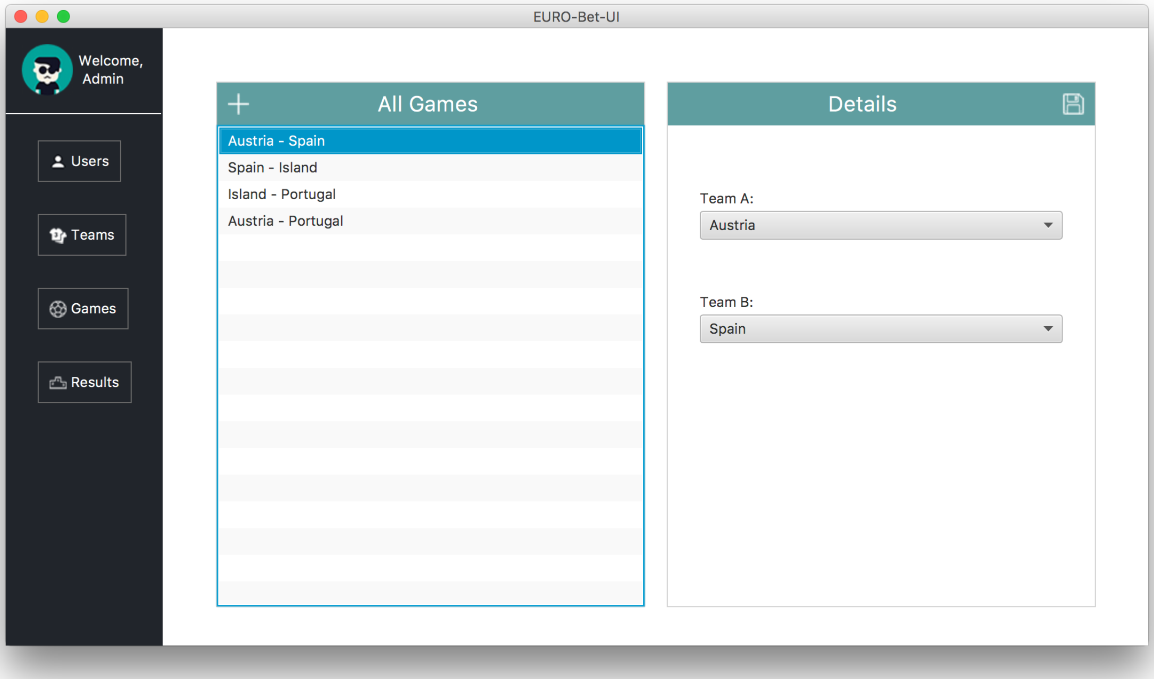


Austria
Spain
Island
Portugal

Country: Austria

Group: GROUPE G

Games

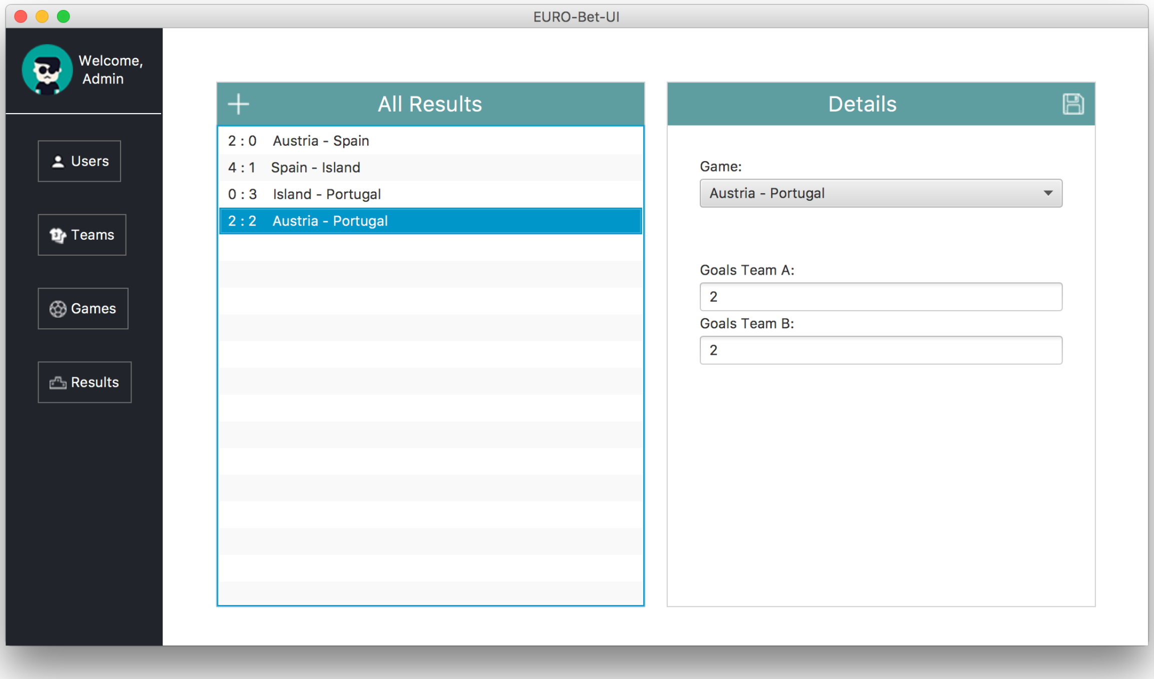


Austria - Spain
Spain - Island
Island - Portugal
Austria - Portugal

Team A: Austria

Team B: Spain

Results



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

Game: Austria - Portugal

Goals Team A: 2

Goals Team B: 2