

<b>SWE4</b>	<b>Übung zu Softwareentwicklung mit modernen Plattformen 4</b>	<b>SS 2022, Übung 5</b>
<input type="checkbox"/> Gruppe 1 (J. Heinzelreiter)		
<input checked="" type="checkbox"/> Gruppe 2 (P. Kulczycki)      Name: <u>Angelos Angelis</u> Aufwand [h]: <u>15</u>		
<input type="checkbox"/> Gruppe 3 (M. Hava)      Peer-Review von: _____		

## Ausbaustufe 1: Pomagaju-UI

Beispiel	Lösungsidee (max. 100%)	Implement. (max. 100%)	Testen (max. 100%)
1a (50 P)	100	100	100
1b (50 P)	100	100	100

Der Plattformbetreiber möchte vor der endgültigen Auftragserteilung der Spendenplattform zunächst einen Prototyp des Verwaltungs- und der Spendenanwendung sehen.

Entwickeln Sie daher mit Hilfe von JavaFX für beide Programme einen funktionstüchtigen, ausbaufähigen Benutzeroberflächen-Prototyp. Versuchen Sie eine möglichst intuitiv zu verwendende Benutzeroberfläche zu entwerfen. Setzen Sie dafür Ihr in der Usability-LVA 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, Menüs, Bestellungen etc.). Durch diese Maßnahme wird Ihre Anwendung einfach erweiterbar, was Ihnen in der nächsten Ausbaustufe zugutekommen sollte.

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 hartcodierten Daten arbeiten. Der Benutzeroberflächen-Prototyp muss folgende Funktionen abdecken:

### a) Verwaltungsanwendung

- Zuordnung eines Benutzers zu einer Annahmestelle beim Starten der Anwendung
- Annahmestellen verwalten (Hinzufügen, Ändern, Löschen, inaktiv setzen)
- Erfassen/Ändern des Bedarfs an Hilfsgütern
- Zuordnung von Hilfsgütern zu einer Kategorie (und einer Annahmestelle), diese muss auch verändert werden können
- Spendenankündigungen mit Filtermöglichkeit anzeigen

Berücksichtigen Sie bei der Verwaltungsanwendung das Gestaltungsprinzip des responsiven Designs. Das Layout der Anwendung sollte also beim Verändern der Fenstergröße sinnvoll angepasst werden.

## b) Spendenanwendung

- Registrierung und Login eines Benutzers
- Anzeige des Bedarfs an Hilfsgütern, Möglichkeit nach Kategorie und/oder Region zu filtern.
- Erstellen einer Spendenankündigung mit Tokenausstellung

Gestalten Sie das Layout der Spendenanwendung so, dass diese später (nicht in diesem Übungsprojekt) einfach in eine mobile Applikation transformiert werden kann. Gehen Sie von einer fixen Bildschirmgröße eines handelsüblichen Mobilgeräts aus.

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.

## Lösungsidee:

Ich habe das Programm mit Scenebuilder gelöst. Die Mockups habe ich auch direkt dort gemacht da es so einfach geht und man es direkt in Code umwandeln kann.

### A)

So stelle ich mir vor wie das Programm funktionieren soll: Ein User kann sich am Anfang anmelden und eine Annahmestelle aussuchen. Nach dem einloggen sieht er auf einer Tabelle alle Annahmestellen die er gleich bearbeiten kann oder auch löschen kann. Der User kann auch mehr Annahmestellen hinzufügen. Dann hat man einen weiteren Tab wo der Bedarf in der am Anfang ausgesuchten Annahmestelle angezeigt wird. Ein weiterer Tab zeigt die Ankündigungen für die ausgesuchte Annahmestelle.

### Annahmen:

Kategorien habe ich wie eine Preisklasse interpretiert. In dem Fall ist 1 = günstig; 2 = Mittelteuer; 3 = Teuer

Bei der Filtermöglichkeit für die Ankündigungen wird nach allen Spalten gefiltert

### B)

So stelle ich mir vor wie das Programm funktionieren soll: Ein Benutzer kann sich anmelden oder Registrieren. Beim Registrieren wird ein neues Dialogfenster geöffnet wo sich der User registrieren kann. Die Daten werden anschließend auf eine Map gespeichert. Beim Login werden die Daten mit Hilfe der Map überprüft. Falls das Überprüfen passt kommt man aufs Dashboard. Dort sieht man den Bedarf für die Annahmestellen in der Nähe. Man kann nach Kategorie und Region filtern um es den User um einiges leichter zu machen sich zu entscheiden. In einem anderen Tab kann der User eine Spende ankündigen wodurch er einen Token (In meinem Fall ein Random String) bekommt.

### Annahmen:

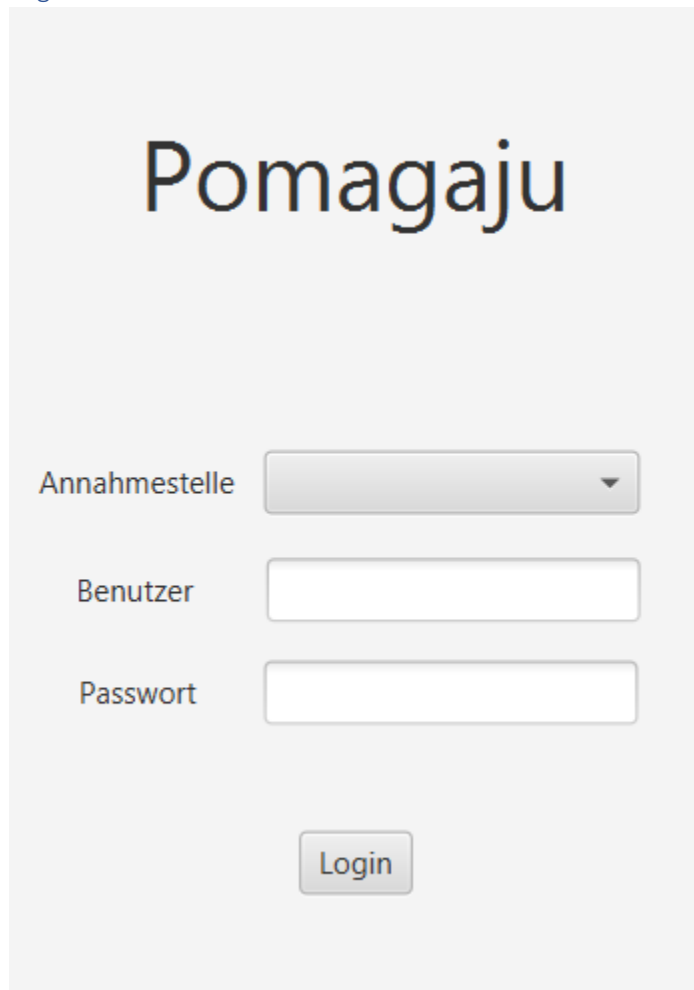
User geben gültige Username und Passwort ein

User geben gültiges Datum ein

Token ist ein beliebiger String

Mockups Verwaltungssoftware:

Login:



The mockup shows a login interface for a system named 'Pomagaju'. The title 'Pomagaju' is displayed in a large, dark blue font at the top. Below it, there are three input fields: a dropdown menu labeled 'Annahmestelle', a text box labeled 'Benutzer', and another text box labeled 'Passwort'. A 'Login' button is positioned below the password field. The entire form is set against a light gray background.

Pomagaju

Annahmestelle

Benutzer

Passwort

Login

Annahmestellen:

Annahmestellen	Bedarf	Spendenankündigungen			
Name	Bundesland	Adresse	Region	Status	

No content in table

Add

Delete

Toggle Status

Bedarf:

Annahmestellen	Bedarf	Spendenankündigungen				
Bezeichnung	Beschreibung	Zustand	Menge	Kategorie	Bedarf	
No content in table						

Spendenankündigung:

Annahmestellen	Bedarf	Spendenankündigungen
----------------	--------	----------------------

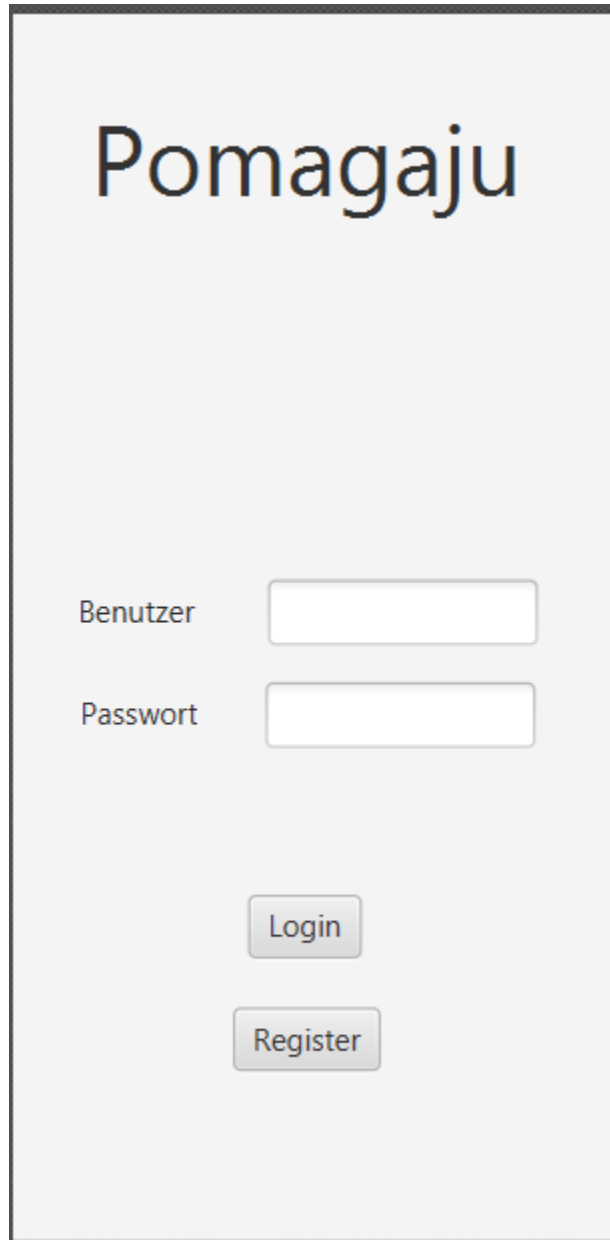
Kommt bald an....

Bezeichnung	Beschreibung	Zustand	Menge	Zeitpunkt
No content in table				

Filter:

Mockups Spendenanwendung:

Login:



The mockup shows a login interface for 'Pomagaju'. It features a light gray background with a dark gray border. The title 'Pomagaju' is at the top in a large, dark blue font. Below it, there are two input fields: 'Benutzer' (User) and 'Passwort' (Password). Each field has a light gray border and a white background. Below the password field, there are two buttons: 'Login' and 'Register'. Both buttons have a light gray background and a dark gray border. The 'Login' button is positioned above the 'Register' button.

# Pomagaju

Benutzer

Passwort

Login

Register

Register:

---

# Register

Benutzer

Passwort

Register



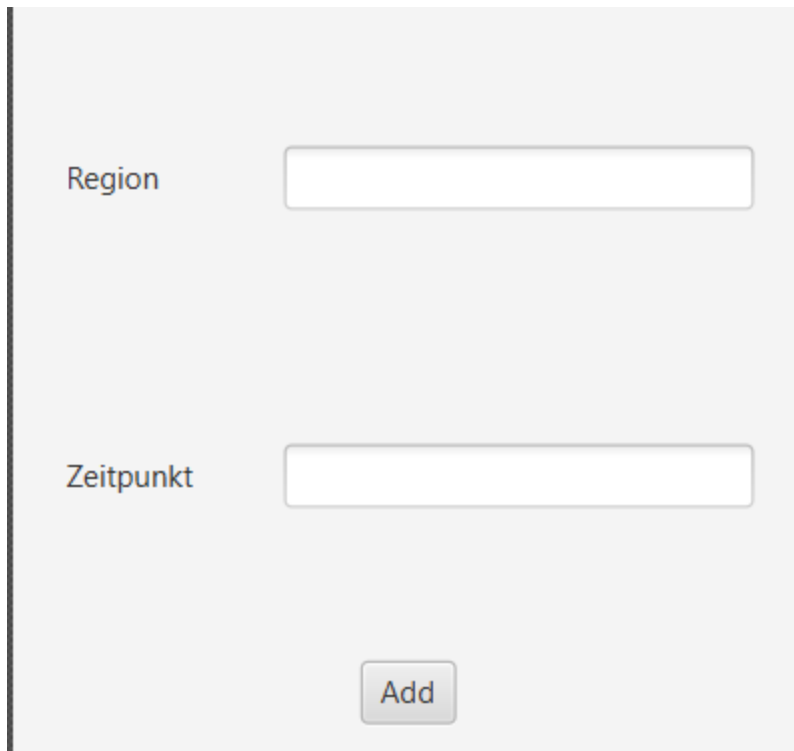
Bedarf:

Bedarf Spendenankündigungen		
Bezeichnung	Beschreibung	Zustand
No content in table		

Filter:

Spendenankündigungen:

Bedarf	Spendenankündigungen	
<h2>Ihre Ankündigungen</h2>		
Token	Ort	Zeitpunkt
No content in table		
<button>Ankündigen</button>		



The image shows a JavaFX user interface with a light gray background. It contains two text input fields. The first field is labeled 'Region' and is empty. The second field is labeled 'Zeitpunkt' and is also empty. Below these two fields is a button labeled 'Add'.

Quellcode: FXML Dateien im pdf nicht enthalten

A)

addAnnahmestelleController:

```
package swe4.ui;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.stage.Modality;
import javafx.stage.Stage;

import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;

public class AddAnnahmestelleController implements Initializable {

    public TextField nameTextField;
```

```
public TextField bundeslandTextField;
public TextField adresseTextField;
public TextField regionTextField;
public TextField statusTextField;

@Override
public void initialize(URL url, ResourceBundle resourceBundle) {

}

public void addToList(ActionEvent event) {
    AnnahmestelleController.annahmestellen.add(new
Annahmestelle(nameTextField.getText(),bundeslandTextField.getText(),adresseTe
xtField.getText(),regionTextField.getText(),statusTextField.getText()));
    //AnnahmestelleController.handle
}
}

Annahmestelle:
package swe4.ui;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;

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

public class Annahmestelle {
    private String name;
    private String bundesland;
    private String Adresse;
    private String Region;
    private String Status;

    public Annahmestelle(String name, String bundesland, String adresse,
String region, String status) {
        this.name = name;
        this.bundesland = bundesland;
        Adresse = adresse;
        Region = region;
        Status = status;
    }

    public String getName() {
        return name;
    }

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

    public String getBundesland() {
        return bundesland;
    }
}
```

```
public void setBundesland(String bundesland) {
    this.bundesland = bundesland;
}

public String getAdresse() {
    return Adresse;
}

public void setAdresse(String adresse) {
    Adresse = adresse;
}

public String getRegion() {
    return Region;
}

public void setRegion(String region) {
    Region = region;
}

public String getStatus() {
    return Status;
}

public void setStatus(String status) {
    Status = status;
}
}
```

AnnahmestelleController:

```
package swe4.ui;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.control.cell.TextFieldTableCell;
import javafx.stage.Modality;
import javafx.stage.Stage;

import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;

public class AnnahmestelleController implements Initializable {
    public Button deleteBtn;
    @FXML
    private TableView<Annahmestelle> tableView;

    @FXML
    public TableColumn<Annahmestelle,String> nameColumn;
```

```
@FXML
public TableColumn<Annahmestelle,String> bundeslandColumn;

@FXML
public TableColumn<Annahmestelle,String> adresseColumn;

@FXML
public TableColumn<Annahmestelle,String> regionColumn;

@FXML
public TableColumn<Annahmestelle,String> statusColumn;

public static ObservableList<Annahmestelle> annahmestellen =
FXCollections.observableArrayList();
public Button reloadBtn;

public void handleAddBtn(ActionEvent actionEvent) throws IOException {
    FXMLLoader fxmlLoader = new
FXMLLoader(getClass().getResource("addAnnahmestelleDialog.fxml"));
    Parent root1 = (Parent) fxmlLoader.load();
    Stage stage = new Stage();
    stage.initModality(Modality.APPLICATION_MODAL);
    stage.setTitle("ABC");
    stage.setScene(new Scene(root1));
    stage.show();
    tableView.setItems(annahmestellen);
}

@Override
public void initialize(URL url, ResourceBundle resourceBundle) {
    nameColumn.setCellValueFactory(new PropertyValueFactory<>("Name"));
    nameColumn.setCellFactory(TextFieldTableCell.forTableColumn());
    bundeslandColumn.setCellValueFactory(new
PropertyValueFactory<>("Bundesland"));
    bundeslandColumn.setCellFactory(TextFieldTableCell.forTableColumn());
    adresseColumn.setCellValueFactory(new
PropertyValueFactory<>("Adresse"));
    adresseColumn.setCellFactory(TextFieldTableCell.forTableColumn());
    regionColumn.setCellValueFactory(new
PropertyValueFactory<>("Region"));
    regionColumn.setCellFactory(TextFieldTableCell.forTableColumn());
    statusColumn.setCellValueFactory(new
PropertyValueFactory<>("Status"));
    statusColumn.setCellFactory(TextFieldTableCell.forTableColumn());
    tableView.setItems(annahmestellen);
    tableView.setEditable(true);
}

public static void initAnnahmestellen() {
    ObservableList<Annahmestelle> a =
FXCollections.observableArrayList();
    annahmestellen.add(new Annahmestelle("A1","OÖ","Softwarepark
69","Kiev","Aktiv"));
    annahmestellen.add(new Annahmestelle("A2","OÖ","Softwarepark
69","Kiev","Aktiv"));
```

```
    }

    public void handleDeleteBtn(ActionEvent event) {

tableView.getItems().removeAll(tableView.getSelectionModel().getSelectedItems
());
    }
}

BedarfController:
package swe4.ui;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;

import java.net.URL;
import java.util.ResourceBundle;

public class BedarfController implements Initializable {
    public TableColumn regColumn;
    @FXML
    private TableView<Hilfsgüter> tableView;

    @FXML
    public TableColumn<Hilfsgüter, String> bezColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> beschColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> zuColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> menColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> katColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> bedColumn;

    public ObservableList<Hilfsgüter> hilfsgüter =
FXCollections.observableArrayList();

    public void initHilfsgüter(){
        ObservableList<String> kategorien =
FXCollections.observableArrayList();
        kategorien.add("1");
        kategorien.add("2");
        kategorien.add("3");
        ObservableList<String> bedarf = FXCollections.observableArrayList();
```

```

        bedarf.add("niedrig");
        bedarf.add("mittel");
        bedarf.add("hoch");
        hilfsgüter.add(new Hilfsgüter("100% Wolle", "T SHirt", "Sehr
gut", "10", kategorien, bedarf));
        hilfsgüter.add(new Hilfsgüter("100% Wolle", "T
SHirt", "gut", "5", kategorien, bedarf));
        hilfsgüter.add(new Hilfsgüter("100% Wolle", "T
SHirt", "Schlecht", "10", kategorien, bedarf));
        hilfsgüter.add(new Hilfsgüter("100% Wolle", "T SHirt", "Sehr
gut", "10", kategorien, bedarf));
    }

    @Override
    public void initialize(URL url, ResourceBundle resourceBundle) {
        bezColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter, String>("Bezeichnung"));
        beschColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter, String>("Beschreibung"));
        zuColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter, String>("Zustand"));
        menColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter, String>("Menge"));
        katColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter, String>("Kategorie"));
        bedColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter, String>("Bedarf"));
        initHilfsgüter();
        tableView.setItems(hilfsgüter);
    }
}

Hilfsgüter:
package swe4.ui;

import javafx.collections.ObservableList;
import javafx.scene.control.ComboBox;

import java.time.LocalDate;

public class Hilfsgüter {
    private String beschreibung;
    private String bezeichnung;
    private String zustand;
    private String menge;
    private ComboBox kategorie;
    private ComboBox bedarf;
    private LocalDate zeitpunkt;

    public Hilfsgüter(String beschreibung, String bezeichnung, String
zustand, String menge, ObservableList kategorie, ObservableList bedarf) {
        this.beschreibung = beschreibung;
        this.bezeichnung = bezeichnung;
        this.zustand = zustand;
        this.menge = menge;
        this.kategorie = new ComboBox(kategorie);
        this.bedarf = new ComboBox(bedarf);
    }
}

```



```
        this.kategorie.getSelectionModel().selectFirst();
        this.bedarf.getSelectionModel().selectFirst();
    }

    public Hilfsgüter(String beschreibung, String bezeichnung, String
zustand, String menge, LocalDate zeitpunkt) {
        this.beschreibung = beschreibung;
        this.bezeichnung = bezeichnung;
        this.zustand = zustand;
        this.menge = menge;
        this.zeitpunkt = zeitpunkt;
    }

    public String getBeschreibung() {
        return beschreibung;
    }

    public void setBeschreibung(String beschreibung) {
        this.beschreibung = beschreibung;
    }

    public String getBezeichnung() {
        return bezeichnung;
    }

    public void setBezeichnung(String bezeichnung) {
        this.bezeichnung = bezeichnung;
    }

    public String getZustand() {
        return zustand;
    }

    public void setZustand(String zustand) {
        this.zustand = zustand;
    }

    public String getMenge() {
        return menge;
    }

    public void setMenge(String menge) {
        this.menge = menge;
    }

    public ComboBox getKategorie() {
        return kategorie;
    }

    public void setKategorie(ComboBox kategorie) {
        this.kategorie = kategorie;
    }

    public ComboBox getBedarf() {
        return bedarf;
    }
}
```

```
public void setBedarf(ComboBox bedarf) {
    this.bedarf = bedarf;
}

public LocalDate getZeitpunkt() {
    return zeitpunkt;
}

}

LoginController:
package swe4.ui;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.stage.Modality;
import javafx.stage.Stage;

import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;

public class LoginController implements Initializable {
    public TextField benutzernameTxtFld;
    public PasswordField passwortTxtFld;
    public ComboBox annahmestelleComboBox;
    public Button loginBtn;
    public Label wrongLogIn;

    @Override
    public void initialize(URL url, ResourceBundle resourceBundle) {
        AnnahmestelleController.initAnnahmestellen();
        ObservableList<String> ann = FXCollections.observableArrayList();
        for (int i = 0; i < AnnahmestelleController.annahmestellen.size();
i++) {
            ann.add(AnnahmestelleController.annahmestellen.get(i).getName());
        }
        annahmestelleComboBox.setItems(ann);
    }

    public void handleLoginBtn(ActionEvent event) throws IOException {
        if(benutzernameTxtFld.getText().toString().equals("Rickroll") &&
passwortTxtFld.getText().toString().equals("123")) {
            wrongLogIn.setText("Success!");

            FXMLLoader fxmlLoader = new
FXMLLoader(getClass().getResource("main.fxml"));
            Parent root1 = (Parent) fxmlLoader.load();
            Stage stage = new Stage();
            stage.initModality(Modality.APPLICATION_MODAL);
            stage.setTitle("ABC");
```

```
        stage.setMinHeight(400);
        stage.setMinWidth(600);
        stage.setScene(new Scene(root1));
        stage.show();
    }

    else if(benutzernameTxtFld.getText().isEmpty() &&
passwordTxtFld.getText().isEmpty()) {
        wrongLogin.setText("Please enter your data.");
    }

    else {
        wrongLogin.setText("Wrong username or password!");
    }
}

}

PomagajuMain:
package swe4.ui;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

import java.io.IOException;

public class PomagajuMain extends Application {
    @Override
    public void start(Stage stage) throws IOException {
        Parent root = FXMLLoader.load(getClass().getResource("Login.fxml"));
        stage.setTitle("Hello!");
        stage.setScene(new Scene(root, 300, 275));
        stage.setMinHeight(450);
        stage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}

SpendenankündigungsController:
package swe4.ui;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.collections.transformation.FilteredList;
import javafx.collections.transformation.SortedList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.*;
```

```
import javafx.scene.control.cell.PropertyValueFactory;

import java.net.URL;
import java.time.LocalDate;
import java.util.ResourceBundle;

public class SpendenankündigungController implements Initializable {
    public TextField filterTxtBox;
    @FXML
    private TableView<Hilfsgüter> tableView;
    @FXML
    public TableColumn<Hilfsgüter, String> bezColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> beschColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> zuColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> menColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> zeitColumn;

    public ObservableList<Hilfsgüter> hilfsgüter =
FXCollections.observableArrayList();

    private void initHilfsgüter() {
        hilfsgüter.add(new Hilfsgüter("100% Wolle", "Hose", "Sehr gut", "10",
LocalDate.now()));
        hilfsgüter.add(new Hilfsgüter("100% Wolle", "T SHirt", "gut", "5",
LocalDate.of(2022, 5, 2)));
        hilfsgüter.add(new Hilfsgüter("100% Wolle", "T
SHirt", "Schlecht", "10", LocalDate.of(2022, 5, 2)));
        hilfsgüter.add(new Hilfsgüter("100% Wolle", "T SHirt", "Sehr
gut", "10", LocalDate.of(2022, 5, 2)));
    }

    @Override
    public void initialize(URL url, ResourceBundle resourceBundle) {
        bezColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter, String>("Bezeichnung"));
        beschColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter, String>("Beschreibung"));
        zuColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter, String>("Zustand"));
        menColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter, String>("Menge"));
        zeitColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter, String>("Zeitpunkt"));
        initHilfsgüter();
        FilteredList<Hilfsgüter> filteredList = new
FilteredList<>(hilfsgüter, b -> true);
        filterTxtBox.textProperty().addListener((observable, oldValue,
newValue) -> {
            filteredList.setPredicate(hilfsgüter -> {
```

```

        // If filter text is empty, display all hilfsgüter.
        if (newValue == null || newValue.isEmpty()) {
            return true;
        }
        String lowerCaseFilter = newValue.toLowerCase();

        if
(hilfsgüter.getBeschreibung().toLowerCase().indexOf(lowerCaseFilter) != -1 )
            return true;
        else if
(hilfsgüter.getBezeichnung().toLowerCase().indexOf(lowerCaseFilter) != -1)
            return true;
        else if
(hilfsgüter.getMenge().toLowerCase().indexOf(lowerCaseFilter) != -1)
            return true;
        else if
(hilfsgüter.getZustand().toLowerCase().indexOf(lowerCaseFilter) != -1)
            return true;
        else if
(hilfsgüter.getZeitpunkt().toString().toLowerCase().indexOf(lowerCaseFilter)
!= -1)
            return true;
        else
            return false;
    });
});
SortedList<Hilfsgüter> sortedData = new SortedList<>(filteredList);
sortedData.comparatorProperty().bind(tableView.comparatorProperty());
tableView.setItems(sortedData);
}

}

```

B)

addAnkündigungDialog:

```

package swe4.ui;

import javafx.event.ActionEvent;
import javafx.fxml.Initializable;
import javafx.scene.control.TextField;

import java.net.URL;
import java.time.LocalDate;
import java.util.ResourceBundle;

public class addAnkündigungDialog implements Initializable {
    public TextField regionTextField;
    public TextField zTextField;

    @Override
    public void initialize(URL url, ResourceBundle resourceBundle) {

```

```
    }

    public void addToList(ActionEvent event) {
        SpendenankündigungController.hilfsgüters.add(new
Hilfsgüter("Random",regionTextField.getText(),LocalDate.parse(zTextField.getT
ext())));
        //AnnahmestelleController.handle
    }
}
```

Annahmestelle:

```
package swe4.ui;
```

```
public class Annahmestelle {
    private String name;
    private String bundesland;
    private String Adresse;
    private String Region;
    private String Status;

    public Annahmestelle(String name, String bundesland, String adresse,
String region, String status) {
        this.name = name;
        this.bundesland = bundesland;
        Adresse = adresse;
        Region = region;
        Status = status;
    }

    public String getName() {
        return name;
    }

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

    public String getBundesland() {
        return bundesland;
    }

    public void setBundesland(String bundesland) {
        this.bundesland = bundesland;
    }

    public String getAdresse() {
        return Adresse;
    }

    public void setAdresse(String adresse) {
        Adresse = adresse;
    }

    public String getRegion() {
        return Region;
    }
}
```

```
    }

    public void setRegion(String region) {
        Region = region;
    }

    public String getStatus() {
        return Status;
    }

    public void setStatus(String status) {
        Status = status;
    }
}

BedarfController:
package swe4.ui;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.collections.transformation.FilteredList;
import javafx.collections.transformation.SortedList;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;

import java.net.URL;
import java.util.ResourceBundle;

public class BedarfController implements Initializable {

    public TextField filterTxtBox;
    @FXML
    private TableView<Hilfsgüter> tableView;

    @FXML
    public TableColumn<Hilfsgüter, String> bezColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> beschColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> zuColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> menColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> katColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> bedColumn;

    @FXML
```

```

    public TableColumn<Hilfsgüter,String> regColumn;

    public ObservableList<Hilfsgüter> hilfsgüter =
FXCollections.observableArrayList();

    public void initHilfsgüter(){
        ObservableList<String> kategorien =
FXCollections.observableArrayList();
        kategorien.add("1");
        kategorien.add("2");
        kategorien.add("3");
        ObservableList<String> bedarf = FXCollections.observableArrayList();
        bedarf.add("niedrig");
        bedarf.add("mittel");
        bedarf.add("hoch");
        hilfsgüter.add(new Hilfsgüter("100% Wolle", "Hose","Sehr
gut","10",kategorien,bedarf,new Annahmestelle("A1","OÖ","Softwarepark
69","Kiev","Aktiv").getBundesland()));
        hilfsgüter.add(new Hilfsgüter("100% Wolle", "T
SHirt","gut","5",kategorien,bedarf,new Annahmestelle("A1","NÖ","Softwarepark
69","Volyn","Aktiv").getBundesland()));
        hilfsgüter.add(new Hilfsgüter("100% Wolle", "T
SHirt","Schlecht","10",kategorien,bedarf,new
Annahmestelle("A1","OÖ","Softwarepark 69","Kiev","Aktiv").getBundesland()));
        hilfsgüter.add(new Hilfsgüter("100% Wolle", "T SHirt","Sehr
gut","10",kategorien,bedarf,new Annahmestelle("A1","NÖ","Softwarepark
69","Volyn","Aktiv").getBundesland()));
    }

    @Override
    public void initialize(URL url, ResourceBundle resourceBundle) {
        bezColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter,String>("Bezeichnung"));
        beschColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter,String>("Beschreibung"));
        zuColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter,String>("Zustand"));
        menColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter,String>("Menge"));
        katColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter,String>("Kategorie"));
        bedColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter,String>("Bedarf"));
        regColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter,String>("Region"));
        initHilfsgüter();
        FilteredList<Hilfsgüter> filteredList = new
FilteredList<>(hilfsgüter, b -> true);
        filterTxtBox.textProperty().addListener((observable, oldValue,
newValue) -> {
            filteredList.setPredicate(hilfsgüter -> {
                // If filter text is empty, display all hilfsgüter.
                if (newValue == null || newValue.isEmpty()) {
                    return true;
                }
                String lowerCaseFilter = newValue.toLowerCase();
                if

```



```

(hilfsgüter.getKategorie().getSelectionModel().getSelectedItem().toString().toLowerCase().indexOf(lowerCaseFilter) != -1 )
    return true;
else if
(hilfsgüter.getRegion().toLowerCase().indexOf(lowerCaseFilter) != -1)
    return true;
else
    return false;
});
});
SortedList<Hilfsgüter> sortedData = new SortedList<>(filteredList);
sortedData.comparatorProperty().bind(tableView.comparatorProperty());
tableView.setItems(sortedData);
}
}

```

Hilfsgüter:

```

package swe4.ui;

import javafx.collections.ObservableList;
import javafx.scene.control.ComboBox;

import javax.swing.plaf.synth.Region;
import java.time.LocalDate;
import java.time.ZonedDateTime;

public class Hilfsgüter {
    public String getToken() {
        return token;
    }

    private String token;
    private String beschreibung;
    private String bezeichnung;
    private String zustand;
    private String menge;
    private ComboBox kategorie;
    private ComboBox bedarf;
    private LocalDate zeitpunkt;
    private String region;

    public Hilfsgüter(String beschreibung, String bezeichnung, String
zustand, String menge, ObservableList kategorie, ObservableList bedarf) {
        this.beschreibung = beschreibung;
        this.bezeichnung = bezeichnung;
        this.zustand = zustand;
        this.menge = menge;
        this.kategorie = new ComboBox(kategorie);
        this.bedarf = new ComboBox(bedarf);
        this.kategorie.getSelectionModel().selectFirst();
        this.bedarf.getSelectionModel().selectFirst();
        token = "RANDOM TOKEN";
    }
}

```

```
public Hilfsgüter(String beschreibung, String bezeichnung, String
zustand, String menge, LocalDate zeitpunkt) {
    this.beschreibung = beschreibung;
    this.bezeichnung = bezeichnung;
    this.zustand = zustand;
    this.menge = menge;
    this.zeitpunkt = zeitpunkt;
    token = "RANDOM TOKEN";
}

public Hilfsgüter(String beschreibung, String bezeichnung, String
zustand, String menge, ObservableList<String> kategorie, ObservableList<String> bedarf,
String region) {
    this.beschreibung = beschreibung;
    this.bezeichnung = bezeichnung;
    this.zustand = zustand;
    this.menge = menge;
    this.kategorie = new ArrayList<String>(kategorie);
    this.bedarf = new ArrayList<String>(bedarf);
    this.kategorie.getSelectionModel().selectFirst();
    this.bedarf.getSelectionModel().selectFirst();
    this.region = region;
    token = "RANDOM TOKEN";
}

public Hilfsgüter(String token, String ort, LocalDate zeitpunkt){
    this.token = token;
    this.region = ort;
    this.zeitpunkt = zeitpunkt;
}

public String getBeschreibung() {
    return beschreibung;
}

public void setBeschreibung(String beschreibung) {
    this.beschreibung = beschreibung;
}

public String getBezeichnung() {
    return bezeichnung;
}

public void setBezeichnung(String bezeichnung) {
    this.bezeichnung = bezeichnung;
}

public String getZustand() {
    return zustand;
}

public void setZustand(String zustand) {
    this.zustand = zustand;
}

public String getMenge() {
    return menge;
}
```

```
    }

    public void setMenge(String menge) {
        this.menge = menge;
    }

    public ComboBox getKategorie() {
        return kategorie;
    }

    public void setKategorie(ComboBox kategorie) {
        this.kategorie = kategorie;
    }

    public ComboBox getBedarf() {
        return bedarf;
    }

    public void setBedarf(ComboBox bedarf) {
        this.bedarf = bedarf;
    }

    public LocalDate getZeitpunkt() {
        return zeitpunkt;
    }

    public String getRegion() {
        return region;
    }
}

LoginController:
package swe4.ui;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.stage.Modality;
import javafx.stage.Stage;

import java.io.IOException;
import java.net.URL;
import java.util.List;
import java.util.Map;
import java.util.ResourceBundle;
import java.util.TreeMap;

public class LoginController implements Initializable {
    public TextField benutzernameTxtFld;
    public PasswordField passwortTxtFld;
    public Button loginBtn;
```

```

public Label wrongLogIn;
public Button registerBtn;
public static Map<String,String> users = new TreeMap<>();

@Override
public void initialize(URL url, ResourceBundle resourceBundle) {
    users.put("Rickroll","123");
}

public void handleLoginBtn(ActionEvent event) throws IOException {
    if(users.containsKey(benutzernameTxtFld.getText().toString()) &&
users.containsValue(passwordTxtFld.getText().toString())) {
        wrongLogIn.setText("Success!");
        FXMLLoader fxmllLoader = new
FXMLLoader(getClass().getResource("main.fxml"));
        Parent root1 = (Parent) fxmllLoader.load();
        Stage stage = new Stage();
        stage.initModality(Modality.APPLICATION_MODAL);
        stage.setTitle("ABC");
        stage.setMinHeight(490);
        stage.setMinWidth(240);
        stage.setMaxHeight(490);
        stage.setMaxWidth(240);
        stage.setScene(new Scene(root1));
        stage.show();
    }
    else if(benutzernameTxtFld.getText().isEmpty() &&
passwordTxtFld.getText().isEmpty()) {
        wrongLogIn.setText("Please enter your data.");
    }
    else {
        wrongLogIn.setText("Wrong username or password!");
    }
}

public void handleRegisterBtn(ActionEvent event) throws IOException {
    FXMLLoader fxmllLoader = new
FXMLLoader(getClass().getResource("Register.fxml"));
    Parent root1 = (Parent) fxmllLoader.load();
    Stage stage = new Stage();
    stage.initModality(Modality.APPLICATION_MODAL);
    stage.setTitle("ABC");
    stage.setMinHeight(490);
    stage.setMinWidth(240);
    stage.setMaxHeight(490);
    stage.setMaxWidth(240);
    stage.setScene(new Scene(root1));
    stage.show();
}
}

```

PomagajuController:

```

package swe4.ui;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;

```

```
import javafx.stage.Stage;

import java.io.IOException;

public class PomagajuMain extends Application {
    @Override
    public void start(Stage stage) throws IOException {
        Parent root = FXMLLoader.load(getClass().getResource("Login.fxml"));
        stage.setTitle("Hello!");
        stage.setScene(new Scene(root, 300, 275));
        stage.setMinHeight(490);
        stage.setMinWidth(240);
        stage.setMaxHeight(490);
        stage.setMaxWidth(240);
        stage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}

RegisterController:
package swe4.ui;

import javafx.event.ActionEvent;
import javafx.fxml.Initializable;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import javafx.scene.text.Text;
import javafx.stage.Modality;
import javafx.stage.Stage;

import java.net.URL;
import java.util.ResourceBundle;

public class RegisterController implements Initializable {
    public Button registerBtn;
    public TextField benutzernameTxtFld;
    public PasswordField passwortTxtFld;

    @Override
    public void initialize(URL url, ResourceBundle resourceBundle) {

    }

    public void handleRegisterBtn(ActionEvent event) {

LoginController.users.put(benutzernameTxtFld.getText(), passwortTxtFld.getText());

        Stage dialogStage = new Stage();
        dialogStage.initModality(Modality.WINDOW_MODAL);
```

```

        VBox vbox = new VBox(new Text("Register Succesfull"));
        vbox.setAlignment(Pos.CENTER);
        vbox.setPadding(new Insets(15));
        dialogStage.setScene(new Scene(vbox));
        dialogStage.show();

        Stage stage = (Stage) registerBtn.getScene().getWindow();
        stage.close();
    }
}

```

SpendenankündigungController:

```

package swe4.ui;

import javafx.beans.property.SimpleStringProperty;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.stage.Modality;
import javafx.stage.Stage;

import java.io.IOException;
import java.net.URL;
import java.time.LocalDate;
import java.util.ResourceBundle;

public class SpendenankündigungController implements Initializable {
    public Button ankBtn;
    @FXML
    private TableView<Hilfsgüter> tableView;

    @FXML
    public TableColumn<Hilfsgüter, String> tokColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> ortColumn;

    @FXML
    public TableColumn<Hilfsgüter, String> zeitColumn;

    public static ObservableList<Hilfsgüter> hilfsgüters =
FXCollections.observableArrayList();

    private void initHilfsgüter() {
        hilfsgüters.add(new Hilfsgüter("Random", "OÖ", LocalDate.now()));
        hilfsgüters.add(new Hilfsgüter("Random", "OÖ", LocalDate.now()));
        hilfsgüters.add(new Hilfsgüter("Random", "OÖ", LocalDate.now()));
    }
}

```

```

    }

    @Override
    public void initialize(URL url, ResourceBundle resourceBundle) {
        tokColumn.setCellValueFactory(data -> new
SimpleStringProperty(data.getValue().getToken()));
        ortColumn.setCellValueFactory(data -> new
SimpleStringProperty(data.getValue().getRegion()));
        zeitColumn.setCellValueFactory(new
PropertyValueFactory<Hilfsgüter,String>("Zeitpunkt"));
        initHilfsgüter();
        tableView.setItems(hilfsgüters);
    }

    public void handleAnkBtn(ActionEvent event) throws IOException {
        FXMLLoader fxmlLoader = new
FXMLLoader(getClass().getResource("addAnkündigungDialog.fxml"));
        Parent root1 = (Parent) fxmlLoader.load();
        Stage stage = new Stage();
        stage.initModality(Modality.APPLICATION_MODAL);
        stage.setTitle("ABC");
        stage.setScene(new Scene(root1));
        stage.show();
        tableView.setItems(hilfsgüters);
    }
}

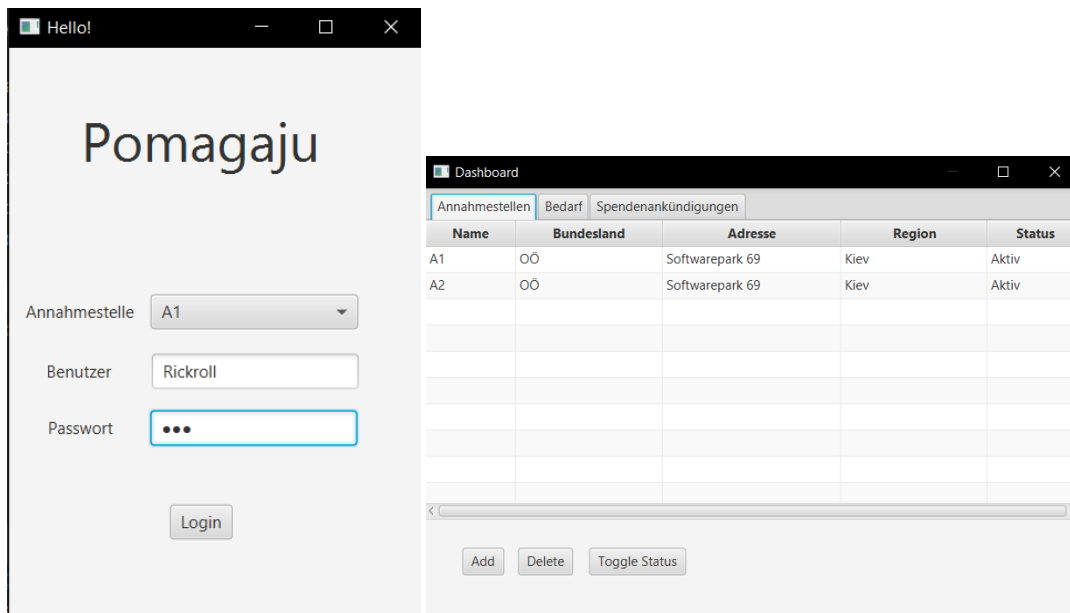
```

## Tests:

A)

Username:Rickroll

Passwort:123



## AddAnnahmestelle:

Dashboard

Name

A3

Bundesland

NÖ

Adresse

Softwarepark 222

Region

Kiev

Status

Aktiv

Add

Name	Bundesland	Adresse	Region	Status
A1	OÖ	Softwarepark 69	Kiev	Aktiv
A2	OÖ	Softwarepark 69	Kiev	Aktiv
A3	NÖ	Softwarepark 222	Kiev	Aktiv

## DeleteAnnahmestelle:

Dashboard

Annahmestellen

Bedarf

Spendenankündigungen

Name	Bundesland	Adresse	Region	Status
A1	OÖ	Softwarepark 69	Kiev	Aktiv
A2	OÖ	Softwarepark 69	Kiev	Aktiv

Add

Delete

Toggle Status

Dashboard

Annahmestellen

Bedarf

Spendenankündigungen

Name	Bundesland	Adresse	Region	Status
A1	OÖ	Softwarepark 69	Kiev	Aktiv

Add

Delete

Toggle Status



EditAnnahmestelle:

Name	Bundesland	Adresse	Region	Status
A1	<input type="text" value="ö"/>	Softwarepark 69	Kiev	Aktiv

Name	Bundesland	Adresse	Region	Status
A1	NÖ	Softwarepark 69	Kiev	Aktiv

EditBedarf

Annahmestellen Bedarf Spendenankündigungen						
Bezeichnung	Beschreibung	Zustand	Menge	Kategorie	Bedarf	
T SHirt	100% Wolle	Sehr gut	10	1	niedrig	
T SHirt	100% Wolle	gut	5	1	niedrig	
T SHirt	100% Wolle	Schlecht	10	1	mittel	
T SHirt	100% Wolle	Sehr gut	10	1	hoch	
					niedrig	

Bezeichnung	Beschreibung	Zustand	Menge	Kategorie	Bedarf	
T SHirt	100% Wolle	Sehr gut	10	1	hoch	
T SHirt	100% Wolle	gut	5	1	niedrig	
T SHirt	100% Wolle	Schlecht	10	1	niedrig	
T SHirt	100% Wolle	Sehr gut	10	1	niedrig	

FilterAnkündigung:

Kommt bald an....

Bezeichnung	Beschreibung	Zustand	Menge	Zeitpunkt	
Hose	100% Wolle	Sehr gut	10	2022-05-18	
T Shirt	100% Wolle	gut	5	2022-05-02	
T Shirt	100% Wolle	Schlecht	10	2022-05-02	
T Shirt	100% Wolle	Sehr gut	10	2022-05-02	

Filter:

Kommt bald an....

Bezeichnung	Beschreibung	Zustand	Menge	Zeitpunkt	
Hose	100% Wolle	Sehr gut	10	2022-05-18	

Filter:

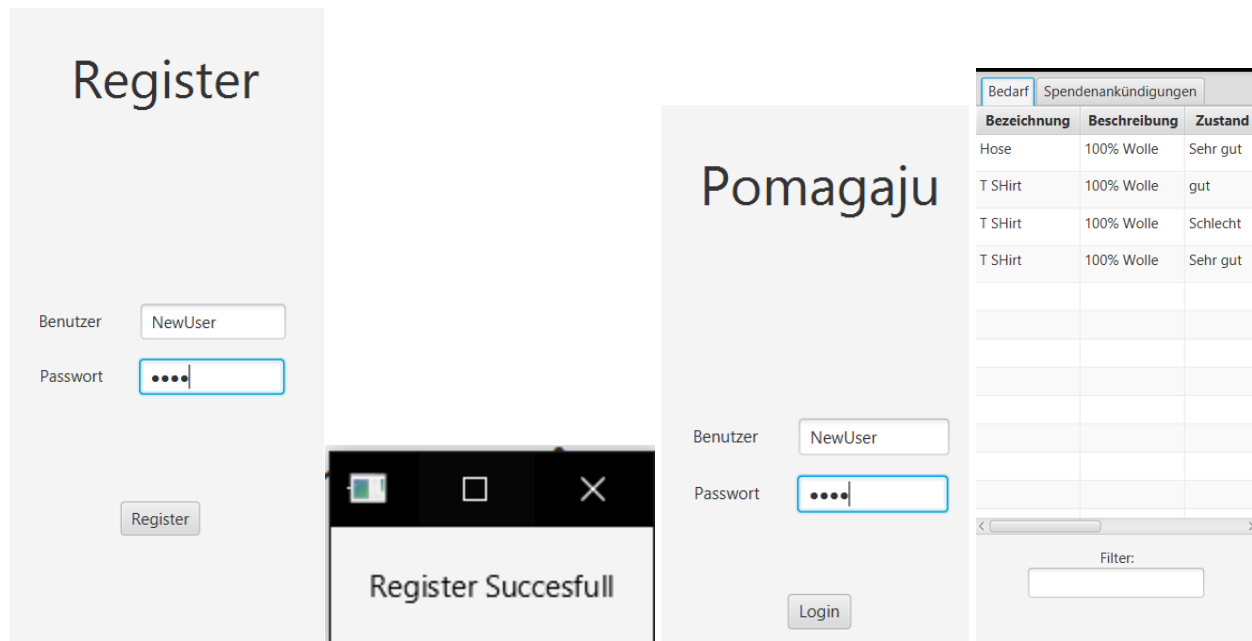
Hose

B)

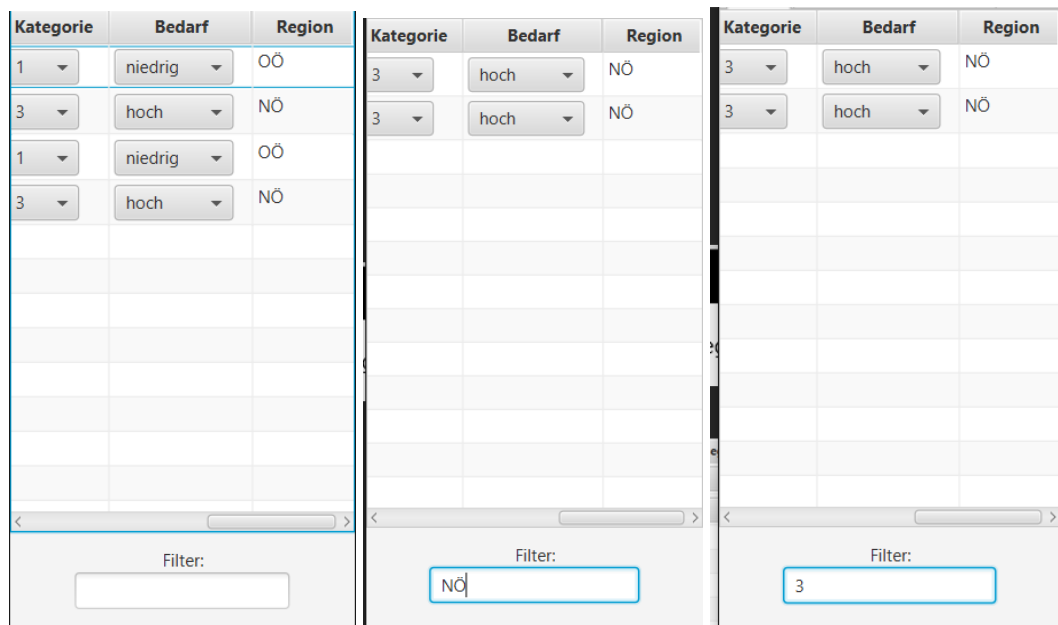
RegisterUser:

Username: NewUser

Password: 2002



FilterBedarf:



SpendeAnkündigen:

## Ihre Ankündigungen

Token	Ort	Zeitpunkt
Random	OÖ	2022-05-18
Random	OÖ	2022-05-18
Random	OÖ	2022-05-18

<

>

Ankündigen

Region

Zeitpunkt

Add

Bedarf

Spendenankündigungen

## Ihre Ankündigungen

Token	Ort	Zeitpunkt
Random	OÖ	2022-05-18
Random	OÖ	2022-05-18
Random	OÖ	2022-05-18
Random	OÖ	2022-07-07

<

>

Ankündigen