**AlertBox.java**

package sample;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.layout.\*;

import javafx.scene.text.Font;

import javafx.stage.Modality;

import javafx.stage.Stage;

public class AlertBox {

static boolean answer;

public static boolean display(){

Stage window = new Stage();

window.initModality(Modality.APPLICATION\_MODAL);

Pane layout = new Pane();

Label label = new Label();

label.setText("Are you sure you want to quit?");

label.setTranslateX(30);

label.setTranslateY(5);

label.setFont(Font.font(15));

layout.getChildren().add(label);

//Creating buttons for exit

Button yesButton, noButton;

yesButton = new Button("YES");

yesButton.setOnAction(e -> {

answer = true;

window.close();

});

yesButton.setTranslateX(50);

yesButton.setTranslateY(30);

noButton = new Button("NO");

noButton.setOnAction(e -> {

answer = false;

window.close();

});

noButton.setTranslateX(110);

noButton.setTranslateY(30);

layout.getChildren().addAll(yesButton,noButton);

Scene scene = new Scene(layout,250,100);

window.setScene(scene);

window.showAndWait();

return answer;

}

}

**KnapSack.java**

package sample;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.HBox;

import javafx.scene.layout.VBox;

import javafx.scene.text.Text;

import javafx.stage.Stage;

import java.io.File;

import java.io.IOException;

import java.util.Arrays;

import java.util.Scanner;

public class KnapSack {

static String name1;

static int maxWeight ;

static int[] profit;

static int[] weight;

static String[] name;

public static void display(String user){

Stage window = new Stage();

Label maxWeightLabel = new Label("Weight: ");

Button submitButton = new Button("SUBMIT");

TextField passMaxWeight = new TextField();

HBox hbox = new HBox();

hbox.setAlignment(Pos.CENTER);

hbox.setSpacing(10);

hbox.setPadding(new Insets(10,10,10,10));

hbox.getChildren().addAll(maxWeightLabel,passMaxWeight,submitButton);

VBox vbox = new VBox();

//Related information to be put on the screen

Text info = new Text("Welcome " + user + "\nPlease enter the desired weight to get the optimal profit");

vbox.getChildren().addAll(info,hbox);

name1 = user;

//Maximum weight

submitButton.setOnAction(e -> {

maxWeight = Integer.parseInt(passMaxWeight.getText());

calculateMaximum(name1);

});

//System.out.println(profitTable[count-1][m]);

Scene scene = new Scene(vbox,450,300);

window.setScene(scene);

window.show();

}

static public void calculateMaximum(String user){

name = new String[100];

name[0] = "NAN";

profit = new int[100];

profit[0] = 0;

weight = new int[100];

weight[0] = 0;

int count = 0;

try{

File file = new File("D:\\Java\\Knap Sack\\Database\\" + user + ".txt");

Scanner scanner = new Scanner(file);

scanner.nextLine();

scanner.nextLine();

int i = 0;

while(scanner.hasNextLine()){

count++;

String[] data = scanner.nextLine().split(",");

if(data[0].equals("") || data[0].equals(" "))

continue;

name[i] = data[0];

weight[i] = Integer.parseInt(data[1]);

profit[i] = Integer.parseInt(data[2]);

i++;

}

}catch (IOException e){

e.printStackTrace();

}

// count = count + 1;

// maxWeight = maxWeight + 1;

//Starting the algorithm

int[][] profitTable = new int[count+1][maxWeight+1]; //Profit Table

for(int i = 0; i <= maxWeight; i++)

profitTable[0][i] = 0;

for(int j = 0; j <= count; j++)

profitTable[j][0] = 0;

for(int i = 1; i <= count; i++){

for(int j = 1; j <= maxWeight; j++){

if(weight[i] > j){

profitTable[i][j] = profitTable[i-1][j];

}

else{

profitTable[i][j] = Math.max(profit[i] + profitTable[i-1][(j - weight[i])],profitTable[i-1][j]);

}

}

}

System.out.println(profitTable[count][maxWeight]);

//Printing the objects to be included

System.out.println("The objects to be included are:");

int res = profitTable[count][maxWeight];

int w = maxWeight;

for(int i = count; i > 0 && res > 0; i--){

if(res == profitTable[i-1][w])

continue;

else{

System.out.println(name[i-1]);

res = res - profit[i-1];

w = w - weight[i-1];

}

}

}

}

**Main.java**

package sample;

import javafx.animation.Animation;

import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.control.cell.TextFieldListCell;

import javafx.scene.control.cell.TextFieldTreeCell;

import javafx.scene.input.MouseButton;

import javafx.scene.input.MouseDragEvent;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.BorderPane;

import javafx.scene.layout.GridPane;

import javafx.scene.layout.HBox;

import javafx.scene.layout.Pane;

import javafx.scene.paint.Color;

import javafx.scene.shape.Circle;

import javafx.scene.shape.Rectangle;

import javafx.scene.shape.Shape;

import javafx.scene.text.Text;

import javafx.stage.Stage;

import javafx.util.Duration;

import tray.animations.AnimationType;

import tray.notification.NotificationType;

import tray.notification.TrayNotification;

import javax.management.Notification;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.Scanner;

public class Main extends Application {

public static void main(String[] args) {

launch(args);

}

Stage window;

boolean answer;

String response2;

@Override

public void start(Stage stage) throws Exception {

stage.setTitle("Intro Page");

GridPane layout = new GridPane();

layout.setPadding(new Insets(10));

layout.setVgap(8);

layout.setHgap(10);

Label userName = new Label("Username: ");

GridPane.setConstraints(userName,0,0);

TextField passUserName = new TextField();

GridPane.setConstraints(passUserName,1,0);

Label password = new Label("Password: ");

GridPane.setConstraints(password,0,1);

PasswordField passPassword = new PasswordField();

passPassword.setPromptText("password");

GridPane.setConstraints(passPassword,1,1);

Text incorrectPassword = new Text();

GridPane.setConstraints(incorrectPassword,2,1);

Text incorrectUsername = new Text();

GridPane.setConstraints(incorrectUsername,2,0);

Button loginButton;

loginButton = new Button("Login");

loginButton.setOnAction(e -> {

String response = login(passUserName,passPassword);

//System.out.println(response);

if(response.equals("Login Successfull")){

TrayNotification tray = new TrayNotification();

AnimationType type = AnimationType.POPUP;

tray.setAnimationType(type);

tray.setTitle("Sign In");

tray.setMessage(response);

tray.setNotificationType(NotificationType.SUCCESS);

tray.showAndDismiss(Duration.millis(50));

//System.out.println(response);

User\_Profile.display(passUserName);

stage.close();

}

if(response.equals("Username Does Not Exists")){

incorrectUsername.setText(response);

incorrectUsername.setFill(Color.RED);

}else {

if (response.equals("Incorrect Password")) {

incorrectPassword.setText(response);

incorrectPassword.setFill(Color.RED);

}

}

});

GridPane.setConstraints(loginButton,1,2);

Label noAccount = new Label("Don't have an account?");

noAccount.setTextFill(Color.BLUE);

GridPane.setConstraints(noAccount,1,3);

Button signupButton = new Button("Sign Up");

GridPane.setConstraints(signupButton,1,4);

//Sign-Up Functions:

signupButton.setOnAction(e -> {

Sign\_Up.display();

});

layout.getChildren().addAll(userName,passUserName,incorrectUsername,loginButton,passPassword,password,incorrectPassword,noAccount,signupButton);

//Creating an exit option

stage.setOnCloseRequest(e -> {

e.consume();

answer = AlertBox.display();

if(answer)

stage.close();

});

stage.setScene(new Scene(layout,450,300));

stage.show();

}

public String login(TextField username,TextField password){

try{

File file = new File("D:\\Java\\Knap Sack\\Database\\" + username.getText() + ".txt");

if(file.exists()) {

Scanner scanner = new Scanner(file);

String pass = scanner.nextLine(); //Correct password present in Database

if (pass.equals(password.getText()))

response2 = "Login Successfull";

else

response2 = "Incorrect Password";

}else{

response2 = "Username Does Not Exists";

}

}catch (FileNotFoundException e){

e.printStackTrace();

}

return response2;

}

}

**Product.java**

package sample;

public class Product {

private String name;

private String weight;

private String profit;

Product(){

this.name = "";

this.weight = "";

this.profit = "";

}

Product(String name,String weight,String profit){

this.name = name;

this.weight = weight;

this.profit = profit;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getWeight() {

return weight;

}

public void setWeight(String weight) {

this.weight = weight;

}

public String getProfit() {

return profit;

}

public void setProfit(String profit) {

this.profit = profit;

}

}

**Register.java**

package sample;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ChoiceBox;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.StackPane;

import javafx.scene.layout.VBox;

import javafx.stage.Modality;

import javafx.stage.Stage;

import java.io.\*;

public class Register {

static boolean answer;

public static boolean display(TextField username,TextField name ,TextField password){

try{

File file = new File("D:\\Java\\Knap Sack\\Database\\" + username.getText() + ".txt");

if(file.exists()){

Stage window = new Stage();

window.initModality(Modality.APPLICATION\_MODAL); //To make it the current window and make the background windows useless

window.setTitle("Error");

VBox layout = new VBox();

Label error\_message = new Label("Username already exists!");

Button okButton = new Button("OK");

okButton.setOnAction(e -> window.close());

layout.getChildren().addAll(error\_message,okButton);

layout.setAlignment(Pos.CENTER);

Scene scene = new Scene(layout,200,150);

window.setScene(scene);

window.showAndWait();

answer = false;

//return answer;

}else{

file.createNewFile();

FileWriter fr = new FileWriter(file,true);

BufferedWriter br = new BufferedWriter(fr);

br.append(password.getText());

br.newLine();

br.append(name.getText());

br.close();

fr.close();

answer = true;

//return answer;

}

}catch (IOException e){

e.printStackTrace();

}

return answer;

}

}

**Sign\_Up.java**

package sample;

import javafx.geometry.Insets;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ChoiceBox;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.GridPane;

import javafx.scene.layout.Pane;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import java.io.BufferedWriter;

public class Sign\_Up {

static Stage window;

static boolean answer;

public static void display(){

window = new Stage();

GridPane layout = new GridPane();

layout.setPadding(new Insets(10));

layout.setHgap(10);

layout.setVgap(8);

//UserName must be Unique

Label username = new Label("Username: ");

GridPane.setConstraints(username,0,0);

TextField passUsername = new TextField();

GridPane.setConstraints(passUsername,1,0);

//Name

Label name = new Label("Name");

GridPane.setConstraints(name,0,1);

TextField passName = new TextField();

GridPane.setConstraints(passName,1,1);

//Gender

Label gender = new Label("GENDER: ");

GridPane.setConstraints(gender,0,4);

ChoiceBox<String> passGender = new ChoiceBox<String>();

//passGender.setScaleX(0.8);

//passGender.setScaleY(0.8);

passGender.getItems().addAll("Male","Female","Other");

passGender.getSelectionModel().select(0); //Predefined display -> Display 'Male'

GridPane.setConstraints(passGender,1,4);

Label password = new Label("Password: ");

GridPane.setConstraints(password,0,3);

TextField passPassword = new TextField();

GridPane.setConstraints(passPassword,1,3);

//Age

Label age = new Label("Age: ");

GridPane.setConstraints(age,0,5);

ChoiceBox<Integer> passAge = new ChoiceBox<Integer>();

for(int i = 18; i < 120; i++){

passAge.getItems().addAll(i);

}

passAge.getSelectionModel().select(0); //Will display Age : 18

GridPane.setConstraints(passAge,1,5);

Button submitButton = new Button("SUBMIT");

GridPane.setConstraints(submitButton,1,6);

submitButton.setOnAction(e -> {

answer = Register.display(passUsername,passName,passPassword);

if(answer) {

window.close();

System.out.println("Successfully Registered");

}

});

layout.getChildren().addAll(name,passName,gender,passGender,submitButton,passUsername,username,password,passPassword,age,passAge);

Scene scene = new Scene(layout,300,300);

window.setScene(scene);

window.show();

}

//Testing function for printing the choices in ChoiceBox

public static void print(TextField text, ChoiceBox<String> choice){

System.out.println(text.getText());

System.out.println(choice.getValue());

}

}

**User\_Profile.java**

package sample;

import javafx.application.Application;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.HBox;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import javax.swing.\*;

import java.io.\*;

import java.util.Scanner;

public class User\_Profile {

static TableView<Product> table;

static boolean answer;

static TextField name;

static TextField weight;

static TextField profit;

static String user;

public static void display(TextField username){

user = username.getText();

Stage window = new Stage();

window.setTitle("Welcome " + username.getText());

TableColumn<Product,String> nameColumn = new TableColumn<>("Name");

nameColumn.setMinWidth(200);

nameColumn.cellValueFactoryProperty().setValue(new PropertyValueFactory<>("name"));

TableColumn<Product,Integer> weightColumn = new TableColumn<>("Weight");

weightColumn.setMinWidth(100);

weightColumn.cellValueFactoryProperty().setValue(new PropertyValueFactory<>("weight"));

TableColumn<Product,Integer> profitColumn = new TableColumn<>("Profit");

profitColumn.setMinWidth(100);

profitColumn.cellValueFactoryProperty().setValue(new PropertyValueFactory<>("profit"));

table = new TableView<>();

table.setItems(getProduct(user));

table.getColumns().addAll(nameColumn,weightColumn,profitColumn);

//Adding the Editable Window

HBox hbox = new HBox();

hbox.setPadding(new Insets(10));

hbox.setSpacing(10);

name = new TextField();

name.setPromptText("Name");

weight = new TextField();

weight.setPromptText("Weight");

profit = new TextField();

profit.setPromptText("Profit");

Button addButton = new Button("ADD");

addButton.setOnAction(e -> {

addButtonClicked(user);

});

Button deleteButton = new Button("DELETE");

deleteButton.setOnAction(e -> {

deleteButtonClicked(user);

});

Button nextButton = new Button("NEXT");

nextButton.setOnAction(e -> {

KnapSack.display(user);

window.close();

});

hbox.getChildren().addAll(name,weight,profit,addButton,deleteButton,nextButton);

VBox vbox = new VBox();

vbox.getChildren().addAll(table,hbox);

window.setOnCloseRequest(e -> {

e.consume();

answer = AlertBox.display();

if(answer)

window.close();

});

Scene scene = new Scene(vbox);

window.setScene(scene);

window.show();

}

public static void addButtonClicked(String username){

//Adding the data to the window

Product product = new Product();

product.setName(name.getText());

product.setWeight(weight.getText());

product.setProfit(profit.getText());

table.getItems().add(product);

//Adding the data to the user profile

try {

File file = new File("D:\\Java\\Knap Sack\\Database\\" + username + ".txt");

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.newLine();

bw.append(name.getText());

bw.append(",");

bw.append(weight.getText());

bw.append(",");

bw.append(profit.getText());

bw.close();

fw.close();

}catch (IOException e){

e.printStackTrace();

}

name.clear();

weight.clear();

profit.clear();

}

public static void deleteButtonClicked(String username){

ObservableList<Product> productSelected, allProducts;

allProducts = table.getItems();

String removeTerm1 = table.getSelectionModel().getSelectedItem().getName() + ",";

String removeTerm2 = String.valueOf(table.getSelectionModel().getSelectedItem().getWeight()) + ",";

String removeTerm3 = String.valueOf(table.getSelectionModel().getSelectedItem().getProfit());

replaceSelected(username,removeTerm1, removeTerm2,removeTerm3);

productSelected = table.getSelectionModel().getSelectedItems();

productSelected.forEach(allProducts::remove);

}

public static void replaceSelected(String username,String string1, String string2,String string3) {

try {

// input the file content to the StringBuffer "input"

BufferedReader file = new BufferedReader(new FileReader("D:\\Java\\Knap Sack\\Database\\" + username + ".txt"));

StringBuffer inputBuffer = new StringBuffer();

String line;

while ((line = file.readLine()) != null) {

inputBuffer.append(line);

inputBuffer.append('\n');

}

file.close();

String inputStr = inputBuffer.toString();

//System.out.println(inputStr); // display the original file for debugging

// logic to replace lines in the string (could use regex here to be generic)

inputStr = inputStr.replace(string1 + string2 + string3, " ," + " ," + " ");

// display the new file for debugging

//System.out.println("----------------------------------\n" + inputStr);

// write the new string with the replaced line OVER the same file

FileOutputStream fileOut = new FileOutputStream("D:\\Java\\Knap Sack\\Database\\" + username + ".txt");

fileOut.write(inputStr.getBytes());

fileOut.close();

} catch (Exception e) {

System.out.println("Problem reading file.");

}

}

private static ObservableList<Product> getProduct(String user){

ObservableList<Product> products = FXCollections.observableArrayList();

//products.add(new Product("Suitcase",20,10));

//Loading Saved Data

try{

File file = new File("D:\\Java\\Knap Sack\\Database\\" + user + ".txt");

Scanner scanner = new Scanner(file);

scanner.nextLine();

scanner.nextLine(); //To skip the first 2 lines which hold password and name

while(scanner.hasNextLine()){

String[] data = scanner.nextLine().split(",");

if(data[0].equals("") || data[0].equals(" "))

continue;

products.add(new Product(data[0],data[1],data[2]));

}

scanner.close();

}catch (FileNotFoundException e){

e.printStackTrace();

}

return products;

}

}