



MARMARA UNIVERSITY FACULTY OF ENGINEERING

CSE2062

Object Oriented Programming

Project2

Instructure: Şakir Bingöl


	Student ID Number	Name & Surname
1	150718031	Alperen EROĞLU
2	150718052	Ravzanur TOK
3	150718056	Berat Asrın CAFEROĞLU
4	150718059	Ezgi SÜNE
5	150718060	Zeynep DOĞAN

FILM & SERIES WATCHING GUIDE APPLICATION

In this project, it is aimed to make a watching guide for the benefit of film-series watchers. Certain elements that should be in a viewing guide are processed for different users.

In the project, different users were added (and their information was kept) by **sign up**, and then different film-series options were presented in different categories. In addition, random content (movie-series) suggestions were also made. After **signing in**, the user can see the contents suitable for his/her taste in the watching guide and access various information about these contents. In addition to the content information, film-series trailers can be accessed on the web. If the user wants to search for a film-series, he/she can go to the content information by using the **search** button. If the user wants to get suggestions, he can see random suggestions on the **homepage** depending on the category.

Main elements of the project:

- Sign up.
 - Sign in.
 - Viewing user information (account details).
 - Categories.
 - Content information.
 - Content search.
 - Content suggestion.
 - Sign out.
- 
- ```
graph LR; A[Sign up.] --- B[]; B --- C[Sign in.]; C --- D[Viewing user information (account details).]; D --- E[Categories.]; E --- F[Content information.]; F --- G[Content search.]; G --- H[Content suggestion.]; H --- I[Sign out.]; B --- J[Watching Guide];
```

This kind of watching guide has been designed with the thought that it may meet some of the needs of TV film-series lovers.

The working principle of the project will first be shown on the application and then the codes will be examined.

(Since the project has a dynamic structure, a more exemplary version of the application will be realized with video presentation.)

Video link:

[https://drive.google.com/file/d/1xaf8m\\_lZ0qlzXySKhbSIwao6AGKefDF7/view?usp=sharing](https://drive.google.com/file/d/1xaf8m_lZ0qlzXySKhbSIwao6AGKefDF7/view?usp=sharing)

Opening:



*Splash Screen*

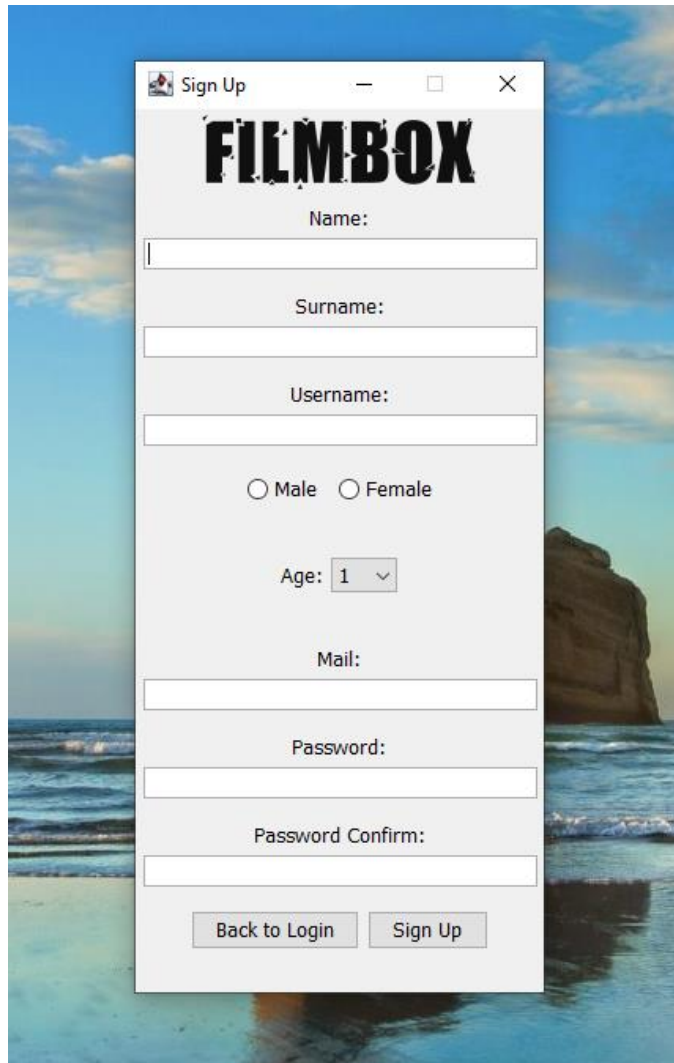
The name of the watching guide is FILMBOX, and the splash screen is shown like this for a few seconds.



*Entrance Screen*

On this screen, the user has **sign in** and **sign up** options. Of course, in order to sign in, sign up must be done first.

### **SIGN UP and SIGN IN**

A screenshot of a web application's 'Sign Up' window. The window has a title bar with a small icon, the text 'Sign Up', and standard window controls (minimize, maximize, close). The main content area has a light gray background with the 'FILMBOX' logo at the top. Below the logo are several form fields: 'Name:' with a text input, 'Surname:' with a text input, 'Username:' with a text input, gender selection with 'Male' and 'Female' radio buttons, 'Age:' with a dropdown menu showing '1', 'Mail:' with a text input, 'Password:' with a text input, and 'Password Confirm:' with a text input. At the bottom are two buttons: 'Back to Login' and 'Sign Up'. The window is overlaid on a background image of a beach with waves and a rock formation.

As seen on the screen, name, surname, gender, age, mail, and password are requested for user sign up.

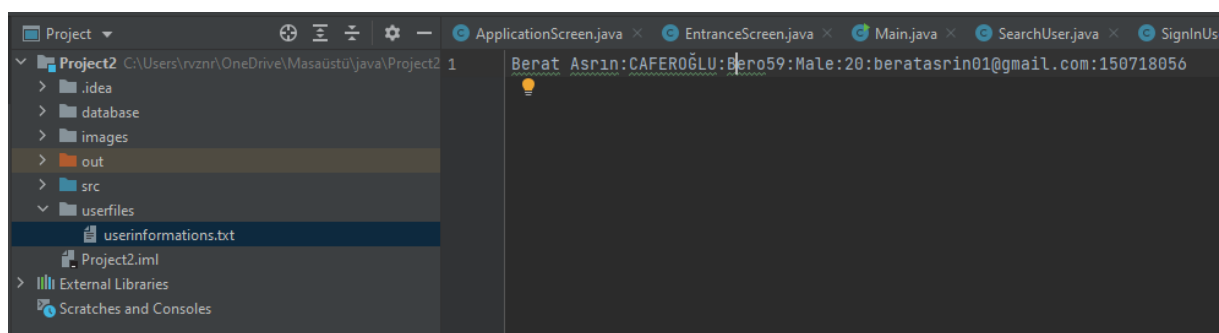
*Sign Up Screen*

*User1 Sign Up*

After the information is entered (here is a small detail, the combobox is used to enter the age) by pressing the **Sign-Up** button, the user is registered. If not, the entrance screen is returned with the **Back to Login** button.

After signing up is completed, **Sign In** will be tried. For sign in, username and password will be used.

*User1 Sign in Using Entrance Screen*

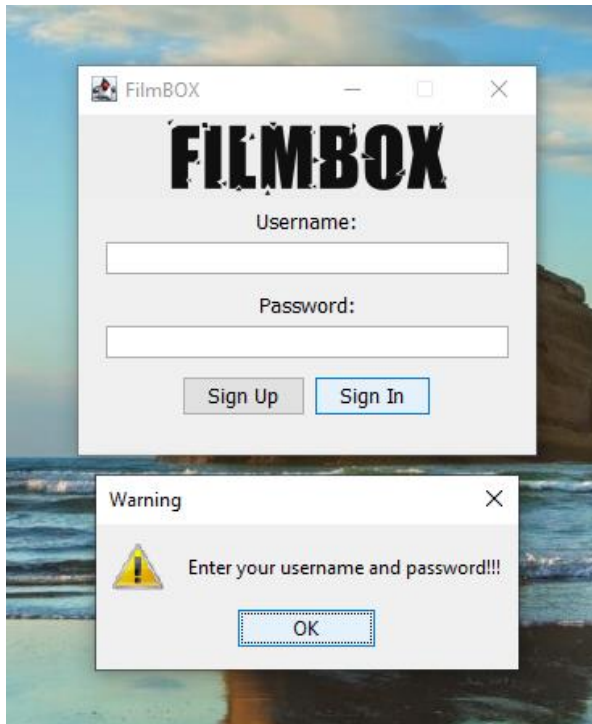


*User Information Text File*

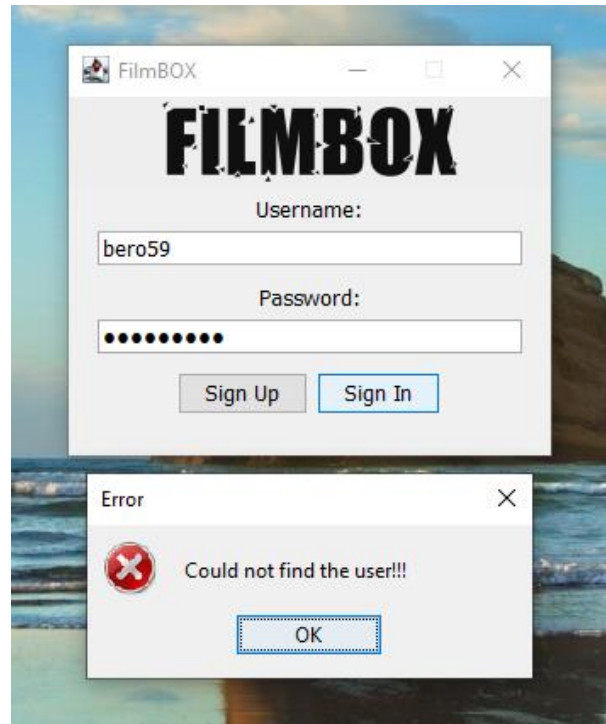
Users are kept in a file; it is seen that the file is updated as the user is added.

Now let's try a few controls for sign in and sign up.

If missing or no information is entered



If wrong information is entered  
(username is Bero59 not bero59)



*Entrance Screen*

It is sufficient not to enter one of the information for the warning message. Likewise, for the Error message, it is sufficient to enter one of the information incorrectly.



If missing or no information is entered

The screenshot shows the FILMBOX Sign Up window. The form fields for Name, Surname, Username, Age, Mail, Password, and Password Confirm are all empty. A 'Warning' dialog box is displayed at the bottom, featuring a yellow warning icon and the text 'Fill the blank areas.' with an 'OK' button.

If mismatched password is entered

The screenshot shows the FILMBOX Sign Up window with the following filled fields: Name (Ravzanur), Surname (TOK), Username (Ravza38), Age (22), and Mail (rvznrtk@gmail.com). The Password and Password Confirm fields contain masked characters (dots). An 'Error' dialog box is displayed at the bottom, featuring a red 'X' icon and the text 'Passwords are not matched.' with an 'OK' button.

*Warnings Using Option-Pane in Sign-Up Screen*

if the previously saved username  
(Bero59) is re-entered

The screenshot shows the 'Sign Up' window for 'FILMBOX'. The 'Username' field contains 'Bero59'. A 'Warning' dialog box is displayed at the bottom, stating 'This username is already taken.' with an 'OK' button. The background of the window is a scenic image of a beach and ocean.

if sign up window is wanted to be  
closed

The screenshot shows the 'Sign Up' window for 'FILMBOX'. A 'Select an Option' dialog box is displayed in the foreground, asking 'Are you sure?' with 'Yes', 'No', and 'Cancel' buttons. The 'Sign Up' window is partially visible behind the dialog box.

Warnings Using OptionPane in Sign-Up Screen

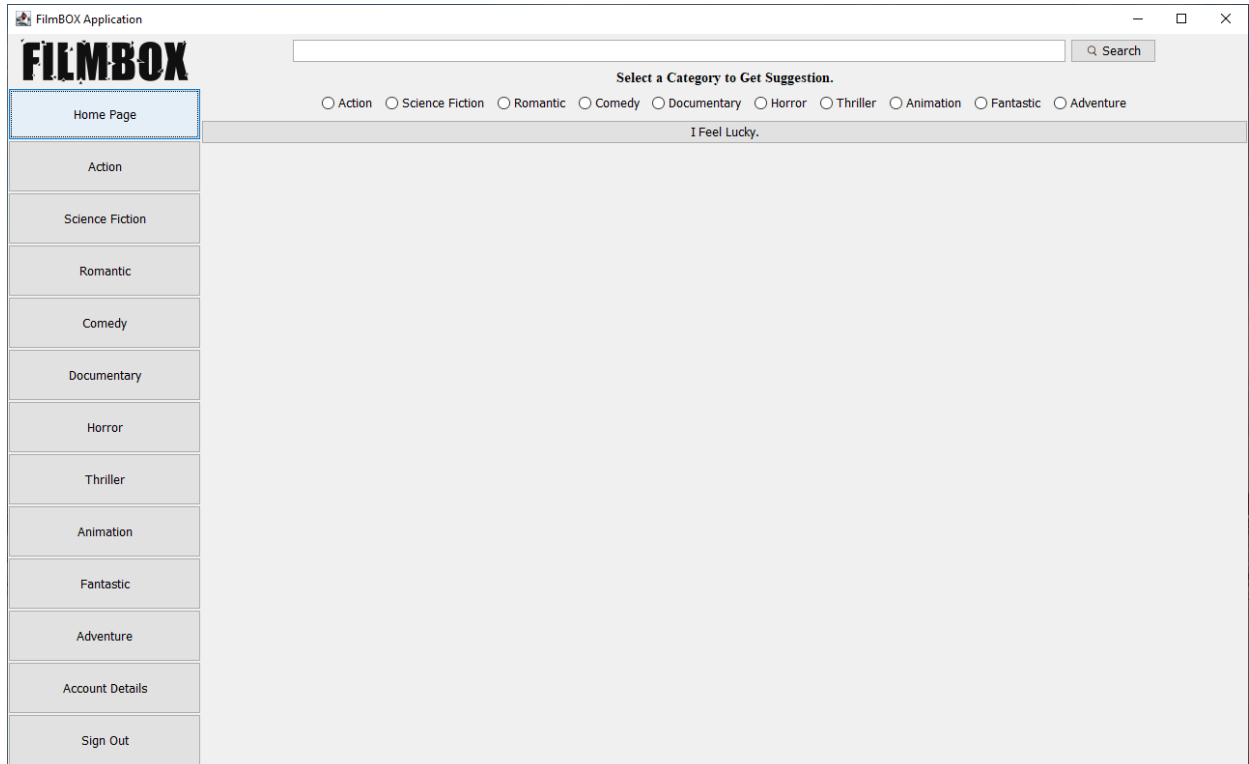
In order not to overlap the screens, the *entrance screen* was closed when the *sign-up* screen was opened. If “Yes” is clicked on the message optionpane, the program is closed, if *No* is clicked, the entrance screen is returned. When *Cancel* is clicked, only the messagebox is closed and the program continues on *sign-up* screen.

As can be seen, different optionpanes such as **warning**, **message** and **error** are used.

The entrance to the application is over, now the watching guide will be examined.



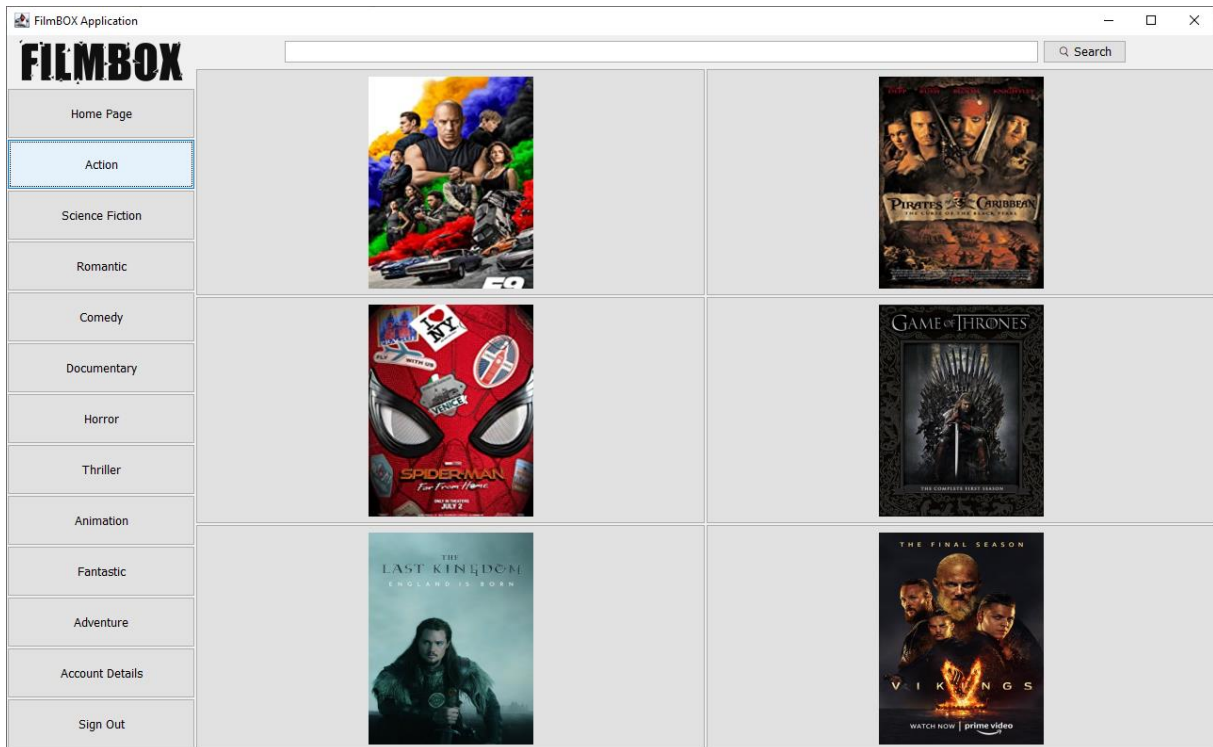
## WATCHING GUIDE



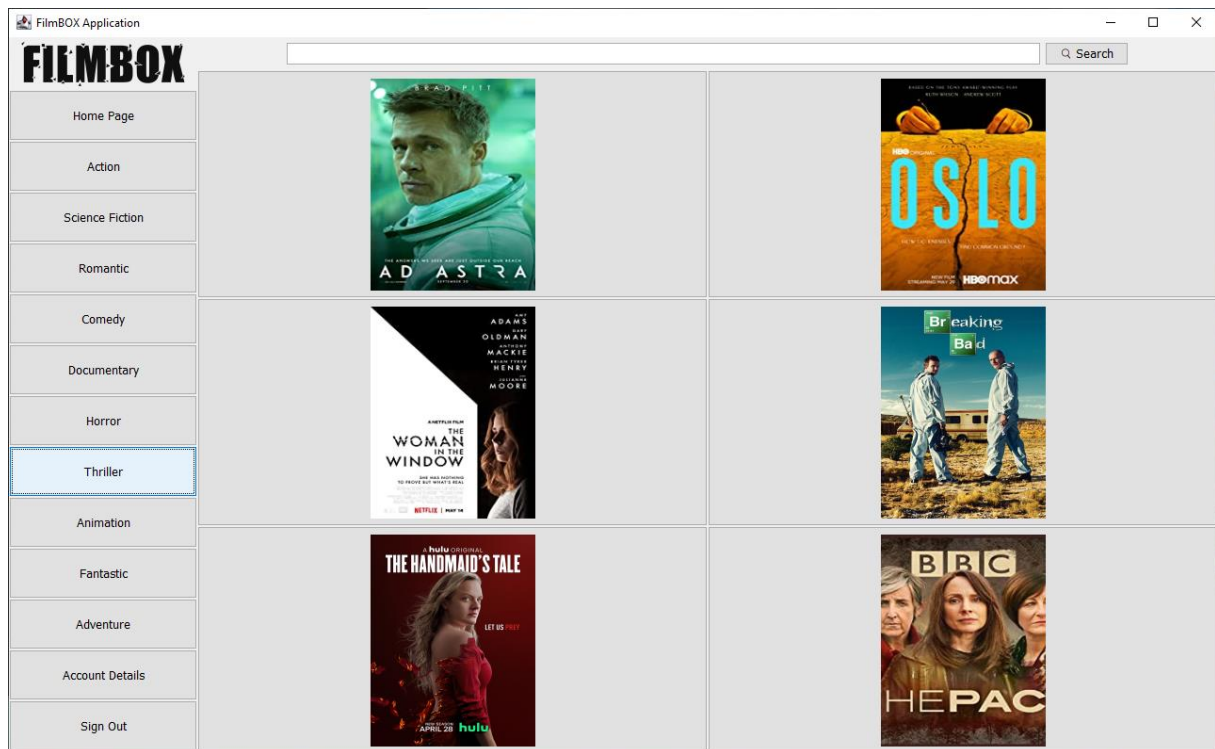
*Application Homepage Screen*

As seen on the main screen, there are categories on the right. At the top, you can see the **Search**, where we can search for content, and **Select a Category to Get Suggestion** sections where we can get suggestions on the **homepage**.

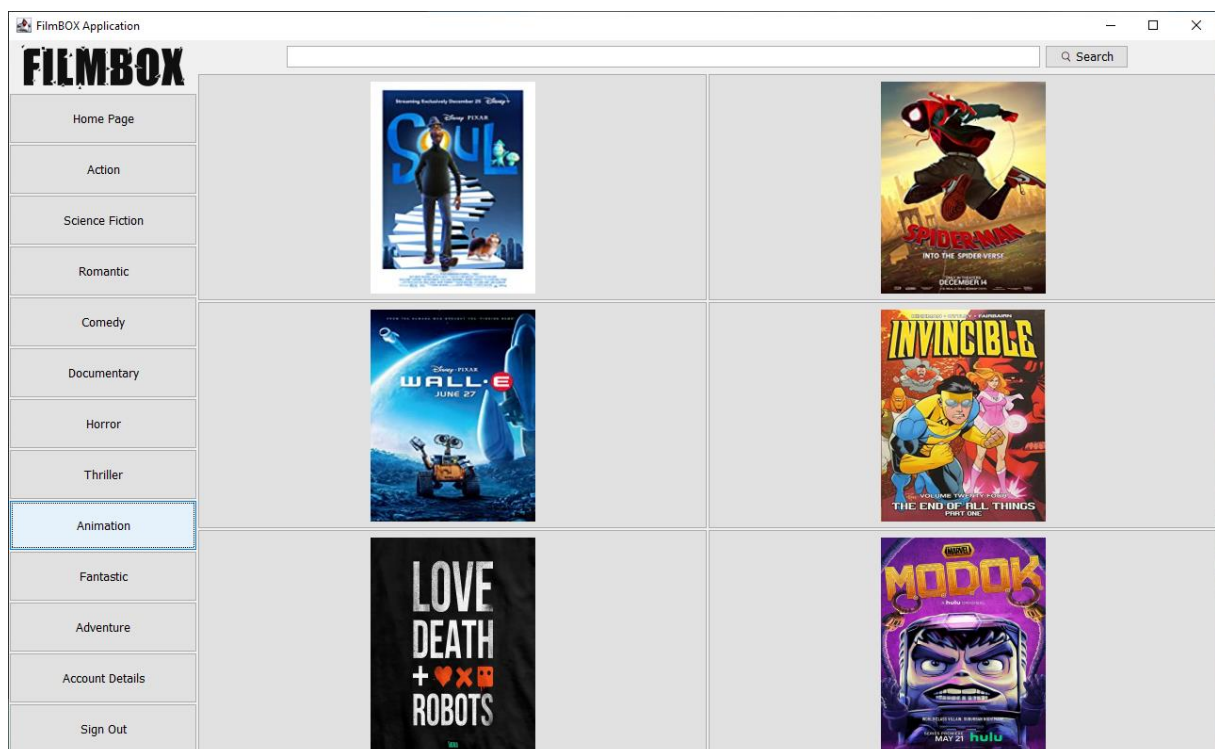
### Categories:



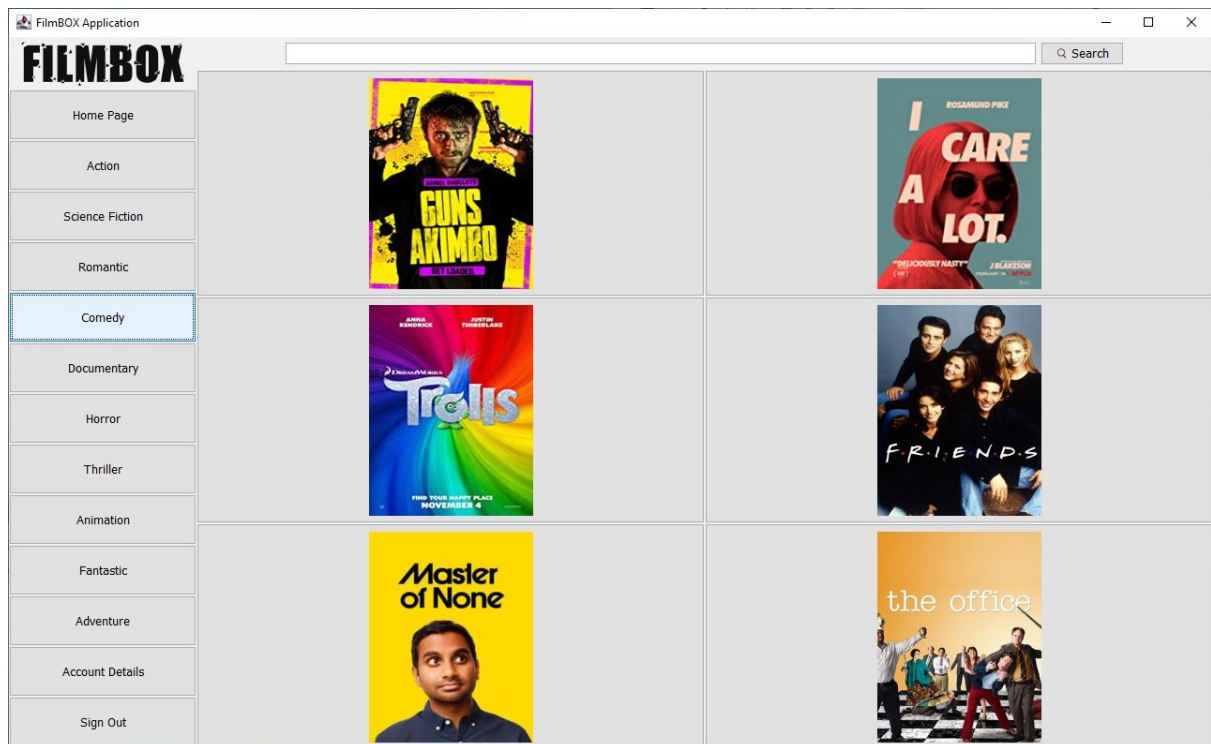
*Action Contents*



Thriller Contents

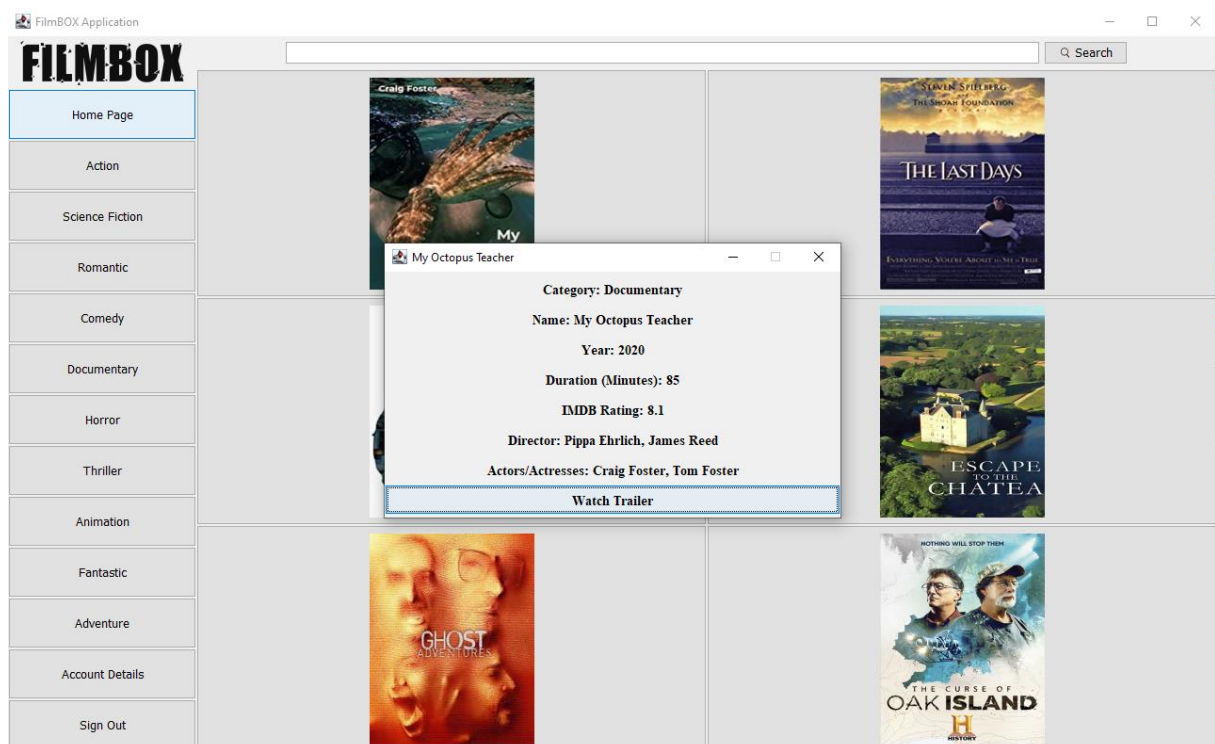


Animation Contents



Comedy Contents


6 contents in 10 different categories can be seen in this way. So, can we get information about the content?



Content Information

Category, name, release year (also season information for series), duration, IMDB score, director/s, actors/actresses information can be accessed by clicking the picture of the content. You can also watch the trailer from the web link with the **Watch Trailer** button.

Some examples for **content information**:

 Doctor Who

**Category:** Adventure

**Name:** Doctor Who

**Year:** 13 Seasons (2005–)(Ongoing)

**Duration (Minutes):** 45

**IMDB Rating:** 8.6

**Director:** Sydney Newman

**Actors/Actresses:** Jodie Whittaker, Peter Capaldi, Pearl Mackie, Matt Smith


**Watch Trailer**

Doctor Who - Official Trailer - Yo x

youtube.com/watch?v=S5BgZf9Pmys

Ara

OTURUM AÇ




0:04 / 1:00

BBCEAMERICA

Doctor Who - Official Trailer

168.173 görüntüleme • 20 Eyl 2018


2 B 1,2 B PAYLAŞ KAYDET



The Adventure Begins | Series 1-8 Trailer | Doctor Who

Doctor Who


260 B görüntüleme • 6 yıl önce



Vincent Van Gogh Galeriyi Ziyaret Etti | Vincent ve Dokto...

Doctor Who


16 Mn görüntüleme • 6 yıl önce



The Daleks | Doctor Who

Doctor Who


288 B görüntüleme • 5 ay önce



DUNE Official Trailer (2020) | Timothée Chalamet, Zendaya...

ONE Media

4 Mn görüntüleme • 9 ay önce




Ubisoft Forward E3 2021 Livestream

IGN

4,7 B kullanıcı izliyor

CANLI YAYIN




Ubisoft Forward E3 2021 Livestream

IGN

33 B kullanıcı izliyor

CANLI YAYIN



On birinci Doktor Onuncu Doktoru Karşılar | Doktorun...

Doctor Who

9,7 Mn görüntüleme • 6 yıl önce

**Eternal Beauty**

Category: Romantic

Name: Eternal Beauty

Year: 2019

Duration (Minutes): 95

IMDB Rating: 6.3

Director: Craig Roberts

Actors/Actresses: Sally Hawkins, Morfydd Clark, Robert Pugh

[Watch Trailer](#)

ETERNAL BEAUTY Trailer (2020) Sally Hawkins, Drama, Romance Movie

223.484 görüntüleme • 20 Ağu 2020

2,8 B 67 PAYLAŞ KAYDET

Movie Coverage

**Eternal Beauty UK Trailer**  
Film Cymru Wales  
3,8 B görüntüleme • 9 ay önce

**BEST UPCOMING MOVIE TRAILERS 2021 (MAY)**  
FilmSpot Trailer  
100 B görüntüleme • 5 gün önce

**Dubai. Luxury life tour. Big Episode.**  
Anton somewhere  
5,3 Mn görüntüleme • 2 yıl önce

**Mix - Movie Coverage**  
YouTube

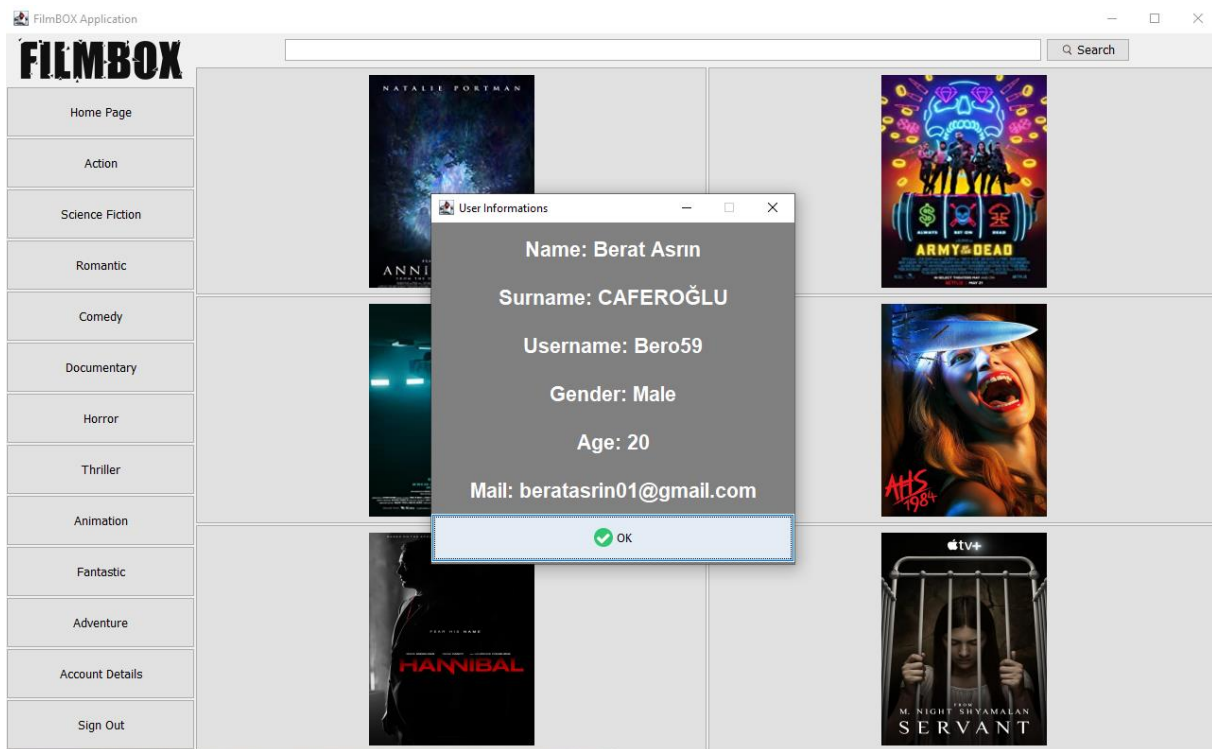
**Eternal Beauty reviewed by Mark Kermode**  
kermodeandmayo  
10 B görüntüleme • 8 ay önce

**Tony Stark Meets Natasha Romanoff - 'I Want One' - Iron...**  
TopMovieClips  
30 Mn görüntüleme • 3 yıl önce

**FAST AND FURIOUS 9 Trailer 2 (NEW 2021) Vin Diesel,...**  
Movie Coverage  
3,7 Mn görüntüleme • 1 ay önce



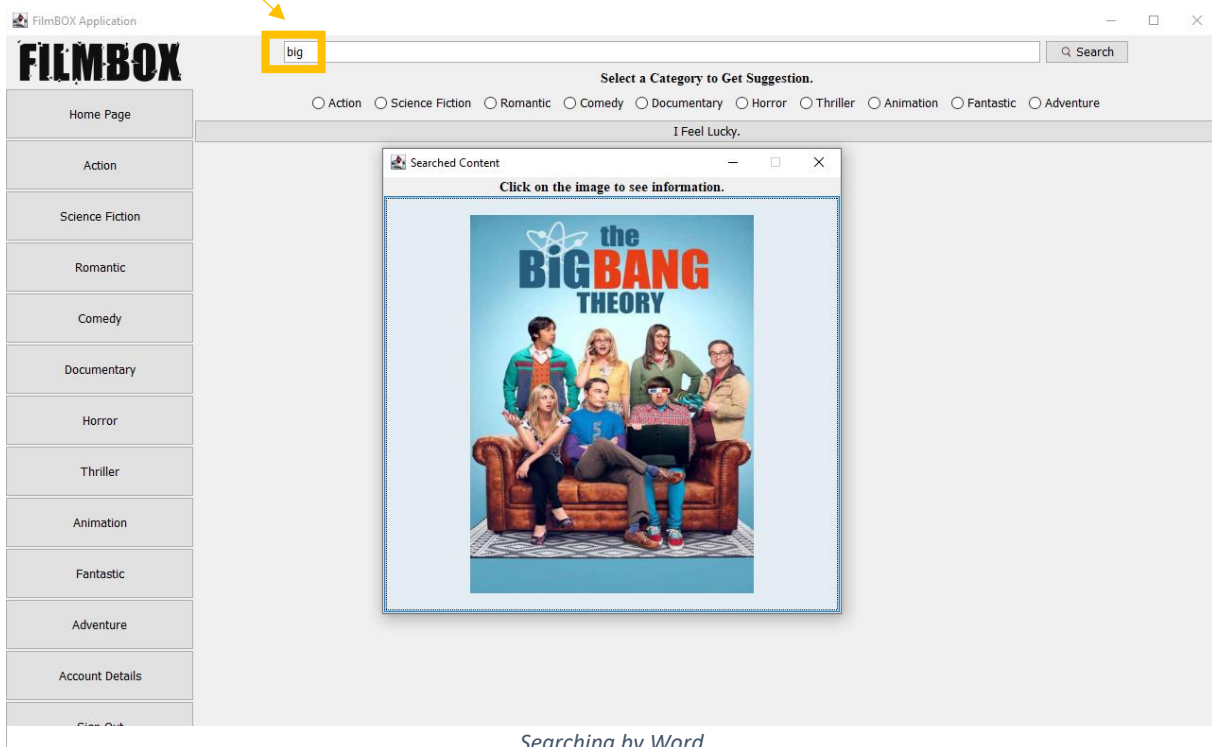
If **Account Details** is clicked, the user can access the information.



*Account Details*

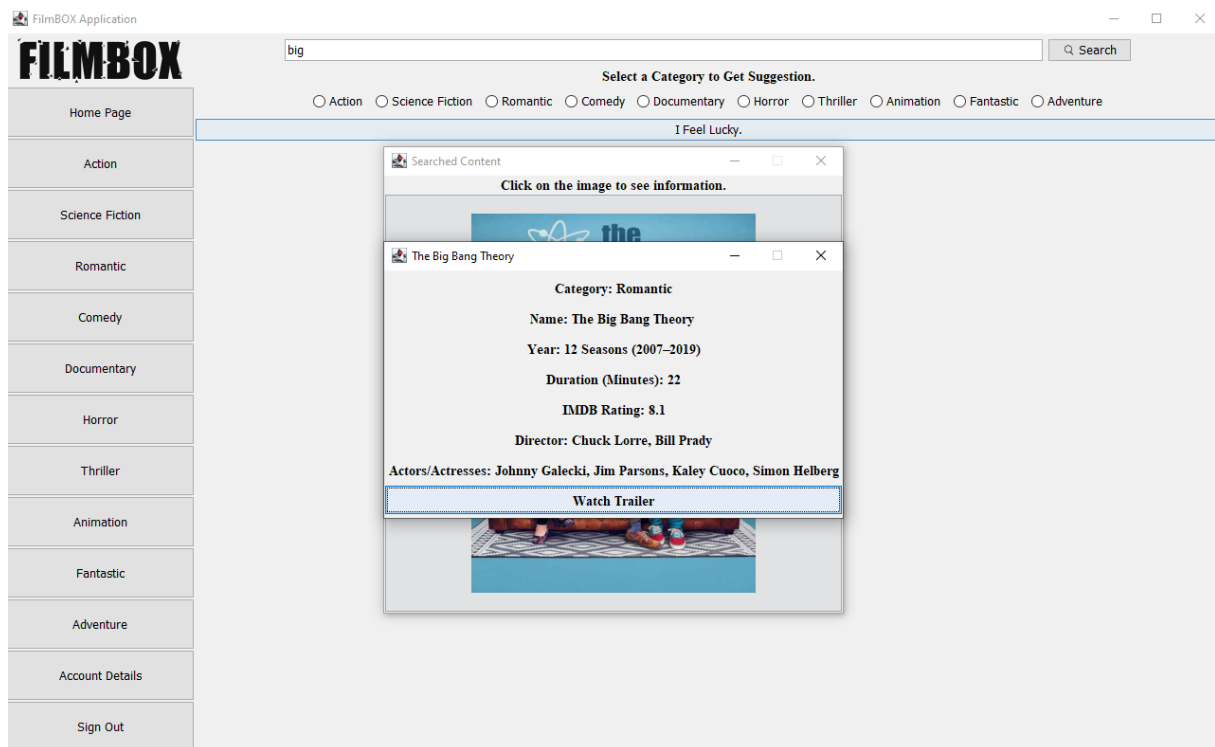
Now let's try to **search** for contents:

The search button is active not only on the homepage but also when the categories button is clicked, so you can search anywhere you want also can search by the full name of the content or the words it contains.

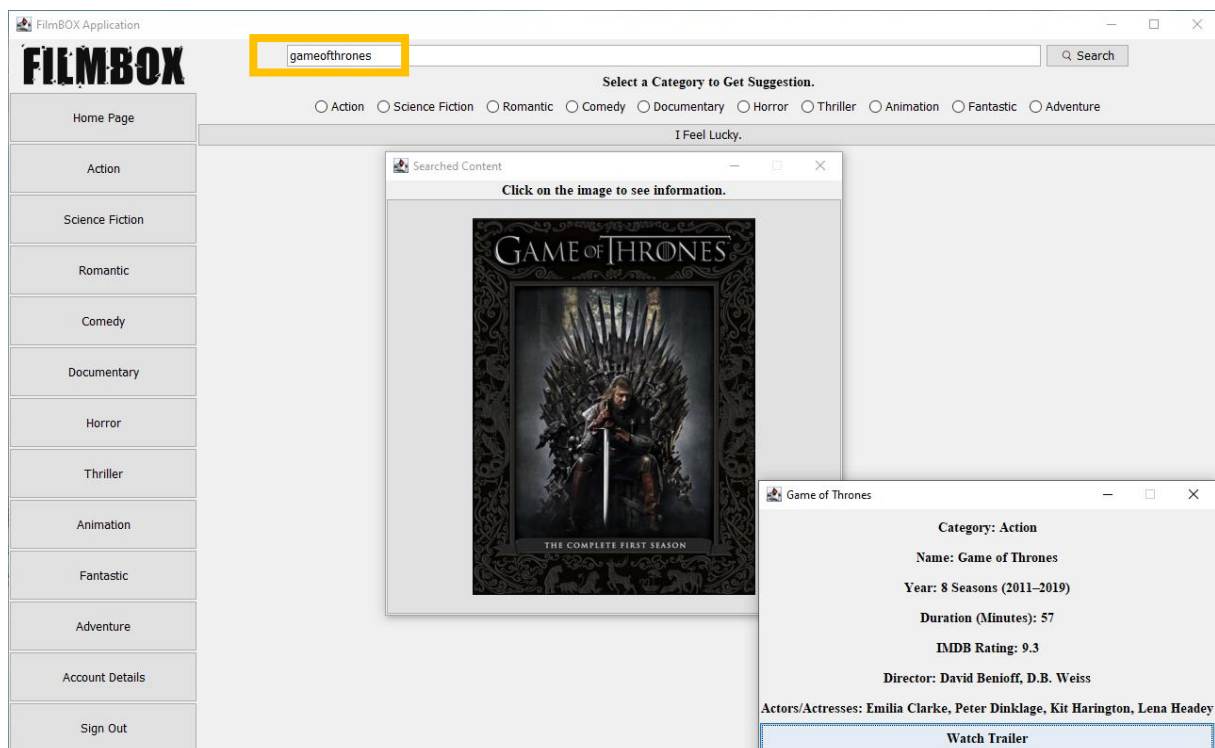


*Searching by Word*





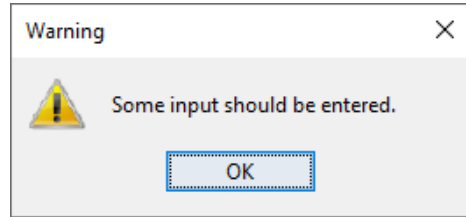
*Searched Content Info*



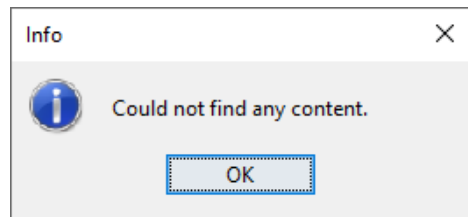
*Searching by Full-Name and Its Info*

Content searching part controls:

If nothing is entered in the search field.



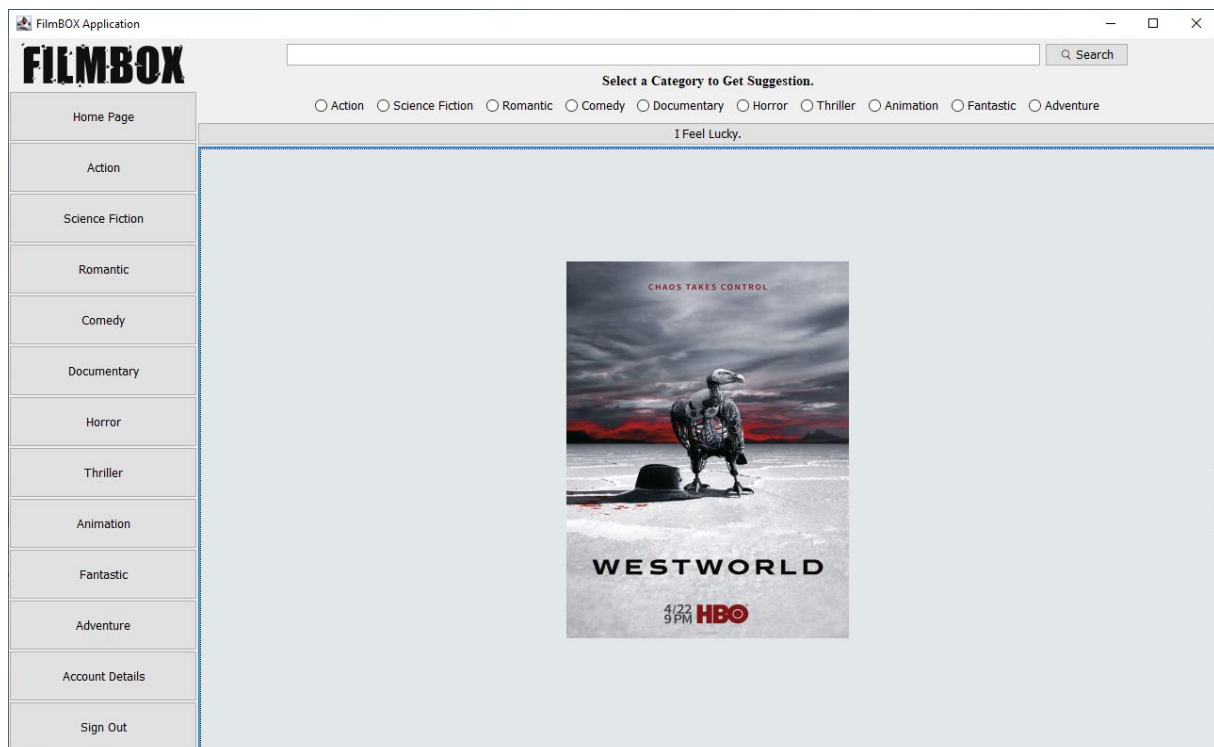
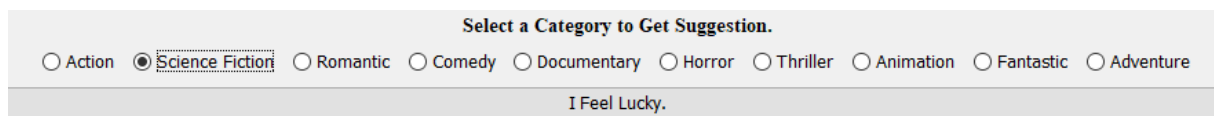
If a content name that is not in the application or a word that does not appear in the content name is entered



As can be seen, different optionpanes are used.

Now, if the **content suggestion** part is to be examined:

After selecting a category then clicking, *I Feel Lucky*, random content is suggested from 6 different content under the relevant category.



*Suggested Content-1 Based on Category*

Westworld

Category: Science Fiction

Name: Westworld

Year: 3 Seasons (2016–)(Ongoing)

Duration (Minutes): 62

IMDB Rating: 8.6

Director: Lisa Joy, Jonathan Nolan

Actors/Actresses: Evan Rachel Wood, Jeffrey Wright, Ed Harris, Thandiwe N...

Watch Trailer

Suggested Content-1 Info

Select a Category to Get Suggestion.

☐ Action
☐ Science Fiction
☐ Romantic
☐ Comedy
☐ Documentary
☐ Horror
☐ Thriller
☐ Animation
☒ Fantastic
☐ Adventure

I Feel Lucky.

FilmBOX Application

FILMBOX

Home Page

Action

Science Fiction

Romantic

Comedy

Documentary

Horror

Thriller

Animation

Fantastic

Adventure


Account Details

Sign Out

Select a Category to Get Suggestion.

☐ Action
☐ Science Fiction
☐ Romantic
☐ Comedy
☐ Documentary
☐ Horror
☐ Thriller
☐ Animation
☐ Fantastic
☐ Adventure

I Feel Lucky.



Suggested Content-2 Based on Category

Stranger Things

Category: Fantastic

Name: Stranger Things

Year: 3 Seasons (2016–)(Ongoing)

Duration (Minutes): 51

IMDB Rating: 8.7

Director: Matt Duffer:Ross Duffer

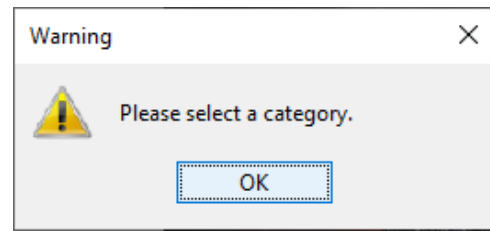
Actors/Actresses: Millie Bobby Brown, Finn Wolfhard, Winona Ryder, David...

Watch Trailer

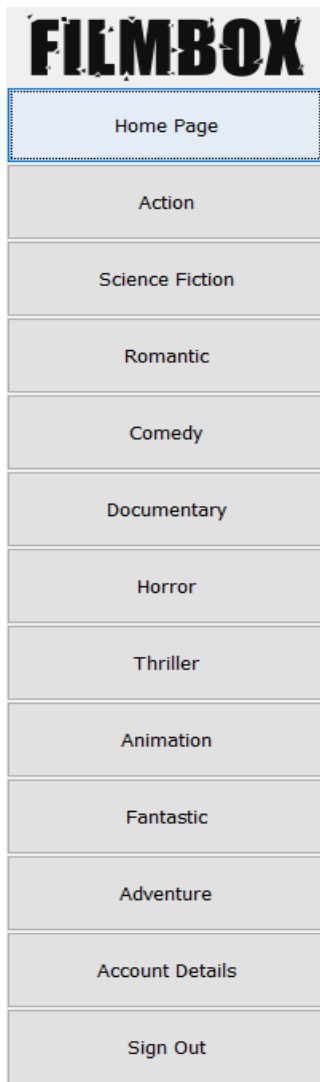
Suggested Content-2 Info

Content suggesting part controls:

if category is not selected

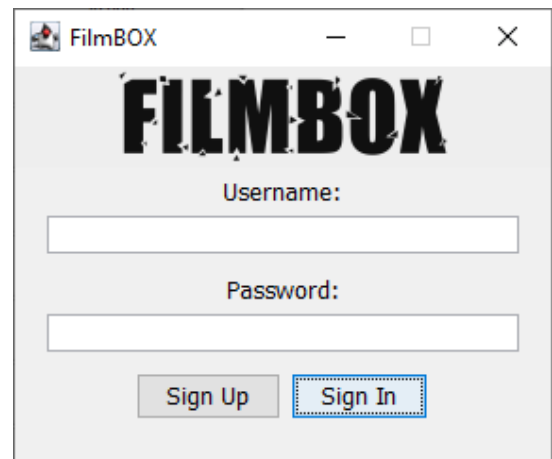


### SIGN OUT



Application Screen Part

When the sign out button is clicked,  
it returns to the entrance screen.



Entrance Screen

## Project Requirements

### GUI Application Based on File Operations:

Reading data from the file example

```
package Codes;

import javax.swing.*;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class SignInUserFile {
 static public boolean searchUser(String usernameInput, String passwordInput){
 try{
 File searchFile = new File("userfiles/userinformations.txt");
 Scanner readFile = new Scanner(searchFile);
 while(readFile.hasNextLine() && !usernameInput.equals("") && !passwordInput.equals("")){
 String data = readFile.nextLine();
 String[] splittedString = data.split(",");
 if(splittedString[2].equals(usernameInput) && splittedString[6].equals(passwordInput)){
 return true;
 }
 }
 if(usernameInput.equals("") || passwordInput.equals("")){
 JOptionPane.showMessageDialog(EntranceScreen.entranceFrame, "Enter your username and password!!!", "Warning", JOptionPane.WARNING_MESSAGE);
 return false;
 }
 } catch (FileNotFoundException e){
 System.out.println("File not found.");
 }
 JOptionPane.showMessageDialog(EntranceScreen.entranceFrame, "Could not find the user!!!", "Error", JOptionPane.ERROR_MESSAGE);
 return false;
 }
}
```

Writing data to file example

```
private void writeToUserInformations(){
 try {
 String dataToWrite = "\n" + nameTextField.getText() + ":" + surnameTextField.getText() + ":" + usernameTextField.getText() + ":" +
 (maleRadioButton.isSelected() ? "Male" : "Female") + ":" + ageComboBox.getSelectedItem() + ":" + mailTextField.getText() + ":" +
 new String(passwordPasswordField.getPassword());
 FileWriter fileToWrite = new FileWriter("userfiles/userinformations.txt", append: true);
 fileToWrite.append(dataToWrite);
 fileToWrite.close();
 System.out.println(usernameTextField.getText()+" is added to userinformations.");
 } catch (Exception error) {
 error.printStackTrace();
 }
}
```

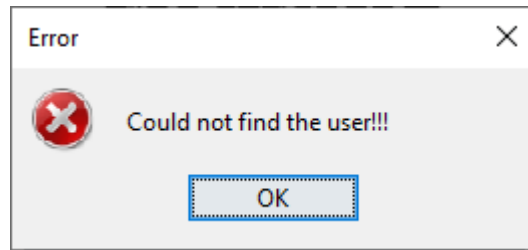
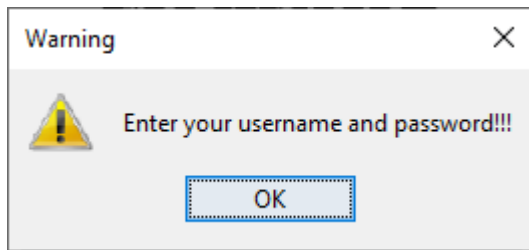
### Screens/ Windows:

As seen in the above outputs, there are many different screens/windows;

- Splash Screen
- Sign In
- Sign Up
- Homepage
- 10 Different Categories
- Content Info(+ trailer link)
- Searched Content and Its Info
- Suggested Content and Its Info
- Account Details

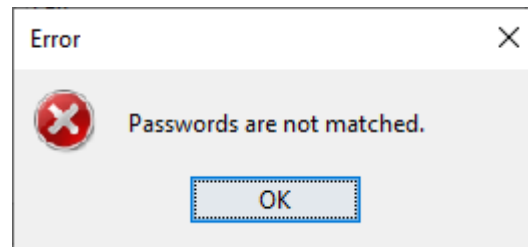
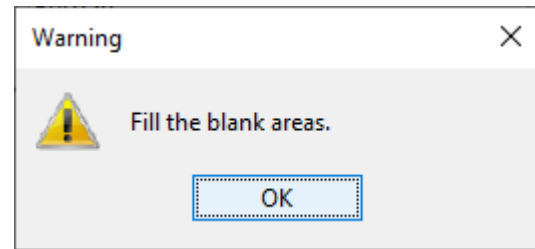
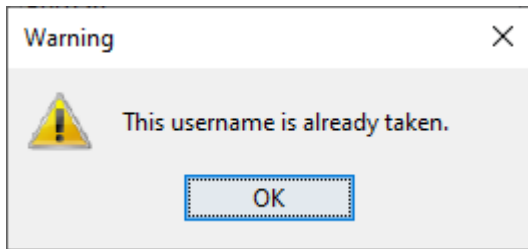
## Dialog boxes:

```
public class SignInUserFile {
 static public boolean searchUser(String usernameInput, String passwordInput){
 try{
 File searchFile = new File(pathname: "userfiles/userinformations.txt");
 Scanner readFile = new Scanner(searchFile);
 while(readFile.hasNextLine() && !usernameInput.equals("") && !passwordInput.equals("")){
 String data = readFile.nextLine();
 String[] splittedString = data.split(regex: "%");
 if(splittedString[2].equals(usernameInput) && splittedString[6].equals(passwordInput)){
 return true;
 }
 }
 if(usernameInput.equals("") || passwordInput.equals("")){
 JOptionPane.showMessageDialog(EntranceScreen.entranceFrame, message: "Enter your username and password!!!", title: "Warning", JOptionPane.WARNING_MESSAGE);
 return false;
 }
 } catch (FileNotFoundException e){
 System.out.println("File not found.");
 }
 JOptionPane.showMessageDialog(EntranceScreen.entranceFrame, message: "Could not find the user!!!", title: "Error", JOptionPane.ERROR_MESSAGE);
 return false;
 }
}
```

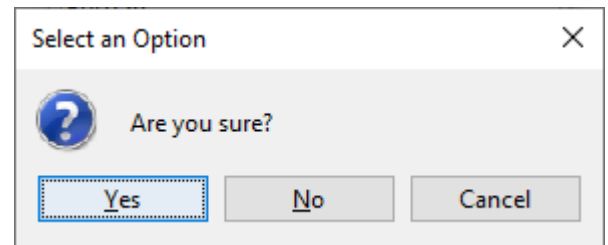


```
else if(e.getSource() == signUpButton){
 if(new String(passwordConfirmPasswordField.getPassword()).equals(new String(passwordPasswordField.getPassword())) && checkString(usernameTextField.getText())
 && checkString(nameTextField.getText()) && checkString(surnameTextField.getText()) && checkString(mailTextField.getText())
 && checkString(new String(passwordPasswordField.getPassword()) && checkString(new String(passwordConfirmPasswordField.getPassword())) && checkRadioButton()){
 if(SearchUser.searchUser(usernameTextField.getText()){
 JOptionPane.showMessageDialog(signUpFrame, message: "This username is already taken.", title: "Warning", JOptionPane.WARNING_MESSAGE);
 return;
 }
 writeToUserInformations();
 signUpFrame.setVisible(false);
 EntranceScreen.userName.setText("");
 EntranceScreen.userPassword.setText("");
 EntranceScreen.entranceFrame.setVisible(true);
 }
 else{
 if(!checkString(new String(passwordPasswordField.getPassword())) || !checkString(new String(passwordConfirmPasswordField.getPassword()))){
 JOptionPane.showMessageDialog(signUpFrame, message: "Fill the blank areas.", title: "Warning", JOptionPane.WARNING_MESSAGE);
 }
 else if(!(new String(passwordConfirmPasswordField.getPassword()).equals(new String(passwordPasswordField.getPassword()))){
 JOptionPane.showMessageDialog(signUpFrame, message: "Passwords are not matched.", title: "Error", JOptionPane.ERROR_MESSAGE);
 }
 else{
 JOptionPane.showMessageDialog(signUpFrame, message: "Fill the blank areas.", title: "Warning", JOptionPane.WARNING_MESSAGE);
 }
 }
}
```

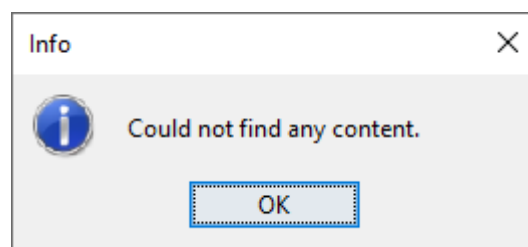
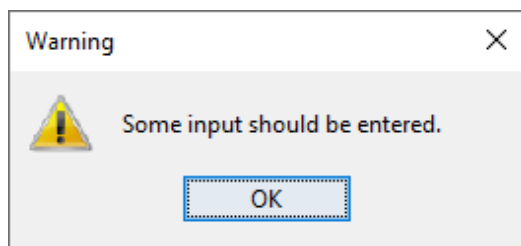




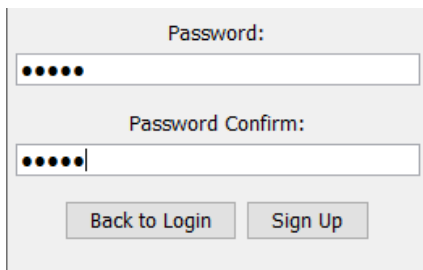
```
public void windowClosing(WindowEvent e) {
 int selection = JOptionPane.showConfirmDialog(signUpFrame, message: "Are you sure?");
 if(selection == JOptionPane.YES_OPTION){
 signUpFrame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
 }
 else if(selection == JOptionPane.NO_OPTION){
 signUpFrame.setVisible(false);
 EntranceScreen.entranceFrame.setVisible(true);
 }
}
```



## Other dialog boxes

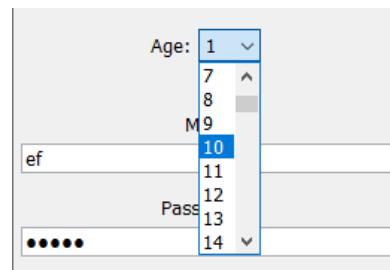


## Controls:



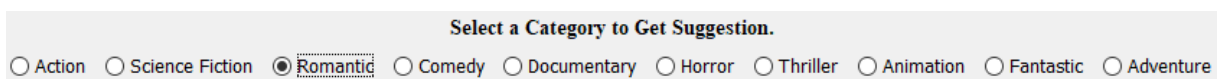
A form with two password fields. The first field is labeled "Password:" and the second is labeled "Password Confirm:". Both fields contain five black dots. Below the fields are two buttons: "Back to Login" and "Sign Up".

*Password Field Control*



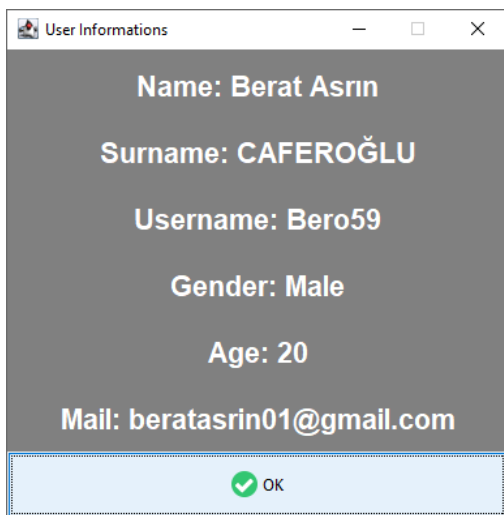
A form with a combobox control. The label "Age:" is next to the combobox, which shows a dropdown menu with values 1, 7, 8, M9, 10, 11, 12, 13, and 14. The value 10 is selected. Below the combobox is a text field containing "ef" and another text field containing "Pass".

*Combobox Control*



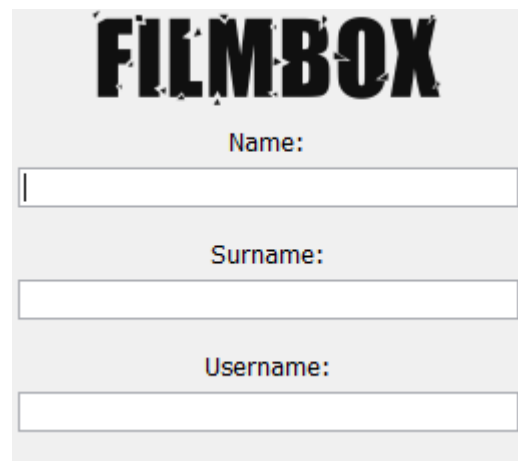
A form with a title "Select a Category to Get Suggestion." and a row of radio buttons. The categories are: Action, Science Fiction, Romantic (selected), Comedy, Documentary, Horror, Thriller, Animation, Fantastic, and Adventure.

*Radio Button Control*



A window titled "User Informations" with a dark gray background. It displays user information: Name: Berat Asrın, Surname: CAFEROĞLU, Username: Bero59, Gender: Male, Age: 20, and Mail: beratasrin01@gmail.com. At the bottom is a blue button with a green checkmark and the text "OK".

*Button Control*



A form with the title "FILMBOX" in a stylized font. Below the title are three text fields labeled "Name:", "Surname:", and "Username:".

*Text Field Control*

### Exception Handling:

```
public void showUserInformationInApplication(){
 try {
 File informationFile = new File(pathname: "userfiles/userinformations.txt");
 Scanner scanInformationFile = new Scanner(informationFile);
 String data;
 String[] userInformation = new String[7];
 while (scanInformationFile.hasNextLine()){
 data = scanInformationFile.nextLine();
 userInformation = data.split(regex: ":");
 if(userInformation[2].equals(loginedUser)){
 break;
 }
 }
 userInformationFrame(userInformation);
 } catch (FileNotFoundException error) {
 error.printStackTrace();
 }
}
```

### Event-Driven and Listener with Exception Handling:

```
case 7 -> {
 JButton trailerButton = new JButton(text: "Watch Trailer");
 trailerButton.addActionListener(e -> {
 try {
 Desktop.getDesktop().browse(new URI(str: "https://www.youtube.com/watch?v="+ contentDetails[7]));
 } catch (Exception error){
 error.printStackTrace();
 }
 });
 labels.add(trailerButton);
}
}
labels.get(i).setFont(new Font(name: "Serif", Font.BOLD, size: 14));
}
```

## Encapsulation:

```
package Codes;

import javax.swing.*;
import java.awt.*;

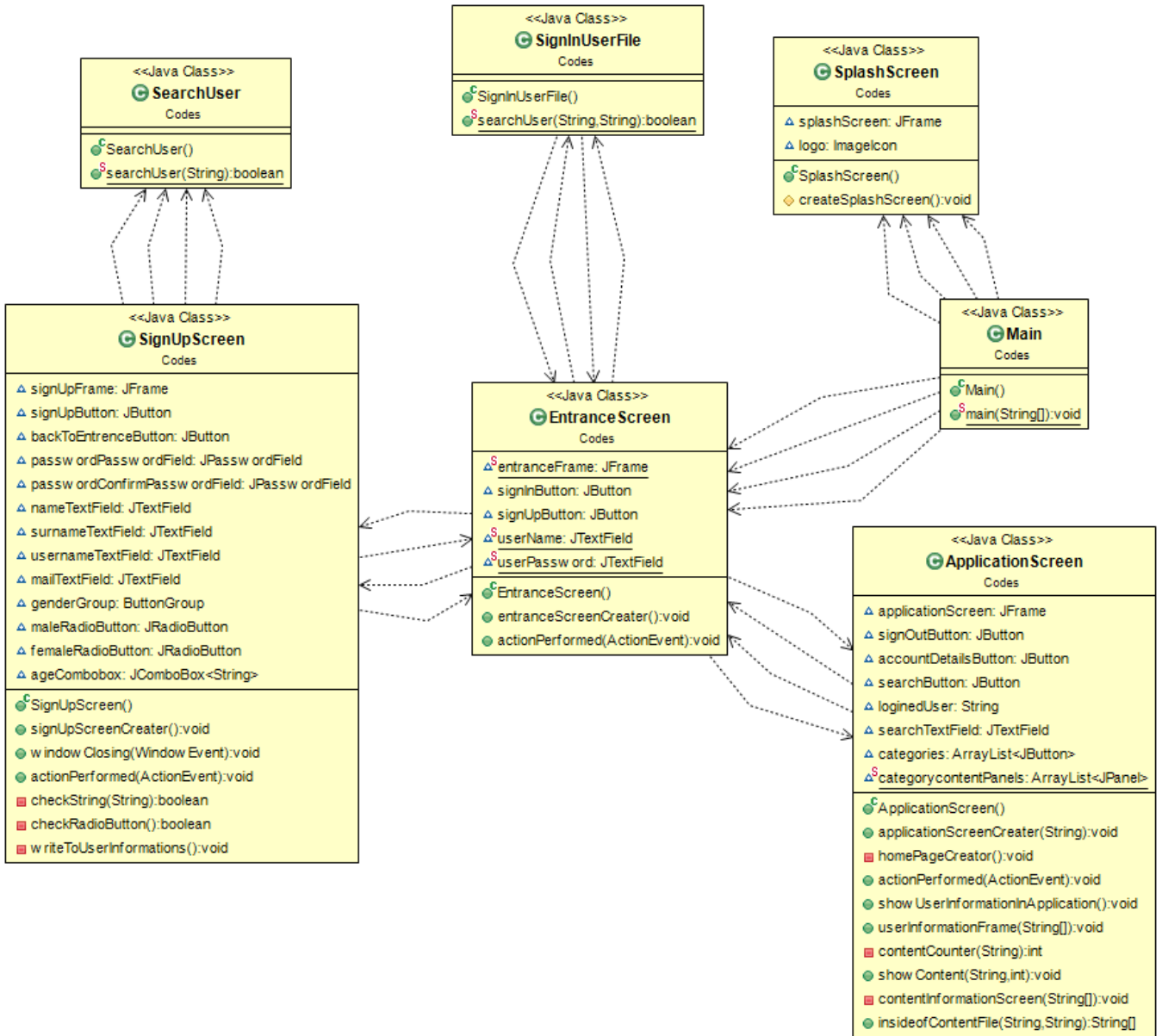
public class SplashScreen {
 JFrame splashScreen;
 ImageIcon logo;

 protected void createSplashScreen(){
 //Image
 logo = new ImageIcon("images/logo1.png");
 JLabel welcomeLogo = new JLabel(logo);

 //Frame
 splashScreen = new JFrame();
 splashScreen.add(welcomeLogo);
 splashScreen.setUndecorated(true);
 splashScreen.setBackground(new Color(1.0f,1.0f,1.0f,0.5f));
 splashScreen.setSize(1300,500);
 splashScreen.setResizable(false);
 splashScreen.setLocationRelativeTo(null);
 splashScreen.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
 splashScreen.setLayout(new CardLayout());
 splashScreen.setVisible(true);

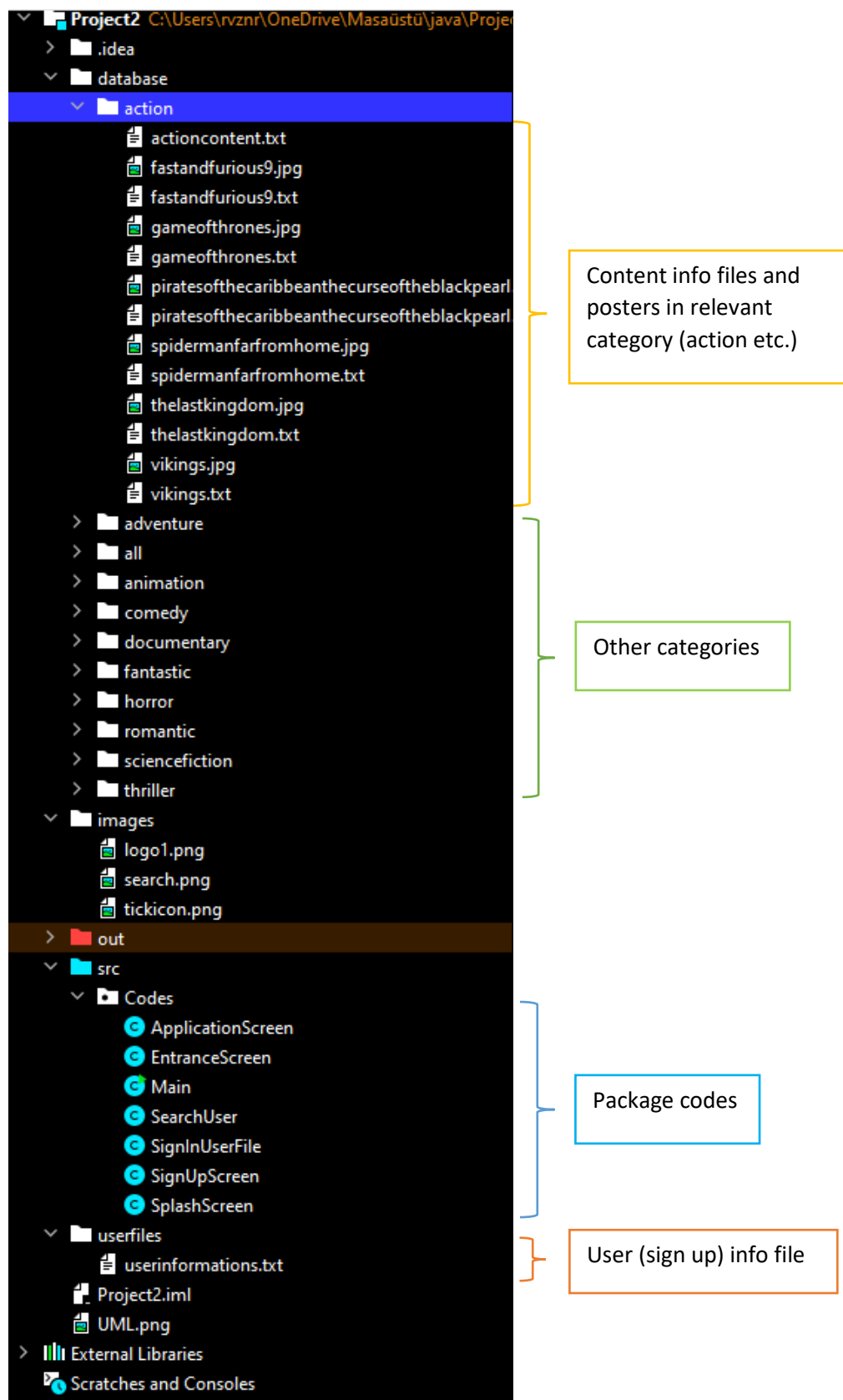
 //Delay using Thread
 try {
 Thread.sleep(5000);
 } catch (InterruptedException e) {
 e.printStackTrace();
 }
 splashScreen.setVisible(false);
 }
}
```

UML Diagram:



### UML Diagram

## Final Hierarchy and The Codes





## Main

```
package Codes;

public class Main {
 public static void main(String[] args) {
 SplashScreen splashScreen = new SplashScreen();
 splashScreen.createSplashScreen();
 EntranceScreen entranceScreen = new EntranceScreen();
 entranceScreen.entranceScreenCreator();
 }
}
```

## SplashScreen

```
package Codes;

import javax.swing.*;
import java.awt.*;

public class SplashScreen {
 JFrame splashScreen;
 ImageIcon logo;
 protected void createSplashScreen() {
 //Image
 logo = new ImageIcon("images/logo1.png");
 JLabel welcomeLogo = new JLabel(logo);

 //Frame
 splashScreen = new JFrame();
 splashScreen.add(welcomeLogo);
 splashScreen.setUndecorated(true);
 splashScreen.setBackground(new Color(1.0f, 1.0f, 1.0f, 0.5f));
 splashScreen.setSize(1300, 500);
 splashScreen.setResizable(false);
 splashScreen.setLocationRelativeTo(null);
 splashScreen.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
 splashScreen.setLayout(new CardLayout());
 splashScreen.setVisible(true);

 //Delay using Thread
 try {
 Thread.sleep(5000);
 } catch (InterruptedException e) {
 e.printStackTrace();
 }
 splashScreen.setVisible(false);
 }
}
```

## EntranceScreen

```
package Codes;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class EntranceScreen implements ActionListener {
 static JFrame entranceFrame = new JFrame("FilmBOX");
 JButton signInButton, signUpButton;
 static JTextField userName, userPassword;
 public void entranceScreenCreator() {
```

```

//UIManager
try {
 UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
} catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
UnsupportedLookAndFeelException e) {
 e.printStackTrace();
}

//Login Logo
ImageIcon logo = new ImageIcon("images/logo1.png");
Image logoReSized = logo.getImage().getScaledInstance(200,200, Image.SCALE_SMOOTH);
logo = new ImageIcon(logoReSized);
JLabel logoLabel = new JLabel(logo);
entranceFrame.add(logoLabel);

//Username Panel
JPanel userNamePanel = new JPanel();
userNamePanel.setLayout(new FlowLayout());
entranceFrame.add(userNamePanel);

//Username Label
JLabel userNameLabel = new JLabel("Username:");
userNamePanel.add(userNameLabel);

//Username Textfield
userName = new JTextField();
userName.setPreferredSize(new Dimension(250,20));
userNamePanel.add(userName);

//Password Panel
JPanel userPasswordPanel = new JPanel();
userPasswordPanel.setLayout(new FlowLayout());
entranceFrame.add(userPasswordPanel);

//Password Label
JLabel userPasswordLabel = new JLabel("Password:");
userPasswordPanel.add(userPasswordLabel);

//Password Textfield
userPassword = new JPasswordField();
userPassword.setPreferredSize(new Dimension(250,20));
userPasswordPanel.add(userPassword);

//Button Panel
JPanel entrancePanel = new JPanel();
entrancePanel.setLayout(new FlowLayout());
entranceFrame.add(entrancePanel);

//Sign up Button
signUpButton = new JButton("Sign Up");
signUpButton.addActionListener(this);
entrancePanel.add(signUpButton);

//Sign in Button
signInButton = new JButton("Sign In");
signInButton.addActionListener(this);
entrancePanel.add(signInButton);

//Frame
entranceFrame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
entranceFrame.setLayout(new GridLayout(4,1));
entranceFrame.setSize(300,250);
entranceFrame.setLocationRelativeTo(null);
entranceFrame.setResizable(false);
entranceFrame.setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {
 if(e.getSource() == signInButton){
 boolean checkFile;
 checkFile = SignInUserFile.searchUser(userName.getText(),userPassword.getText());
 if(checkFile){
 entranceFrame.setVisible(false);
 ApplicationScreen applicationScreen = new ApplicationScreen();
 applicationScreen.applicationScreenCreator(userName.getText());
 }
 }
}

```

```

 }
 else if(e.getSource() == signUpButton){
 entranceFrame.setVisible(false);
 SignUpScreen signUpScreen = new SignUpScreen();
 signUpScreen.signUpScreenCreator();
 }
}
}

```

## SearchUser

```

package Codes;

import java.io.File;
import java.util.Scanner;

public class SearchUser {
 static public boolean searchUser(String username){
 try{
 File file = new File("userfiles/userinformations.txt");
 Scanner scanner = new Scanner(file);
 String[] information;
 String line;
 while (scanner.hasNextLine()){
 line = scanner.nextLine();
 information = line.split(":");
 if(username.equals(information[2])){
 return true;
 }
 }
 } catch (Exception error) {
 error.printStackTrace();
 }
 return false;
 }
}

```

## SignUpScreen

```

package Codes;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.*;

public class SignUpScreen extends WindowAdapter implements ActionListener {
 JFrame signUpFrame = new JFrame("Sign Up");
 JButton signUpButton, backToEntranceButton;
 JPasswordField passwordPasswordField, passwordConfirmPasswordField;
 JTextField nameTextField, surnameTextField, usernameTextField, mailTextField;
 ButtonGroup genderGroup;
 JRadioButton maleRadioButton, femaleRadioButton;
 JComboBox <String> ageCombobox;

 public void signUpScreenCreator() {
 //SignUp Logo Label
 ImageIcon logo = new ImageIcon("images/logo1.png");
 Image logoReSized = logo.getImage().getScaledInstance(200, 200, Image.SCALE_SMOOTH);
 logo = new ImageIcon(logoReSized);
 JLabel logoLabel = new JLabel(logo);
 signUpFrame.add(logoLabel);

 //Name Label
 JLabel nameLabel = new JLabel("Name:");
 nameLabel.setVerticalAlignment(JLabel.CENTER);
 nameLabel.setHorizontalAlignment(JLabel.CENTER);

 //Name TextField
 }
}

```

```

nameTextField = new JTextField();
nameTextField.setPreferredSize(new Dimension(250,20));

//Name Panel
JPanel namePanel = new JPanel();
namePanel.setLayout(new FlowLayout());
namePanel.add(nameLabel);
namePanel.add(nameTextField);
signUpFrame.add(namePanel);

//Surname Label
JLabel surnameLabel = new JLabel("Surname:");
surnameLabel.setVerticalAlignment(JLabel.CENTER);
surnameLabel.setHorizontalAlignment(JLabel.CENTER);

//Surname TextField
surnameTextField = new JTextField();
surnameTextField.setPreferredSize(new Dimension(250,20));

//Surname Panel
JPanel surnamePanel = new JPanel();
surnamePanel.setLayout(new FlowLayout());
surnamePanel.add(surnameLabel);
surnamePanel.add(surnameTextField);
signUpFrame.add(surnamePanel);

//Username Label
JLabel usernameLabel = new JLabel("Username:");
usernameLabel.setHorizontalAlignment(JLabel.CENTER);
usernameLabel.setVerticalAlignment(JLabel.CENTER);

//Username textfield
usernameTextField = new JTextField();
usernameTextField.setPreferredSize(new Dimension(250,20));

//Username input panel
JPanel usernameInput = new JPanel();
usernameInput.add(usernameLabel);
usernameInput.add(usernameTextField);
usernameInput.setLayout(new FlowLayout());
signUpFrame.add(usernameInput);

//Gender RadioButton
maleRadioButton = new JRadioButton("Male");
femaleRadioButton = new JRadioButton("Female");
genderGroup = new ButtonGroup();
genderGroup.add(maleRadioButton);
genderGroup.add(femaleRadioButton);

//GenderPanel
JPanel genderPanel = new JPanel();
genderPanel.setLayout(new FlowLayout());
genderPanel.add(maleRadioButton);
genderPanel.add(femaleRadioButton);
signUpFrame.add(genderPanel);

//Age Panel
String[] ages = new String[99];
for (int i=1;i<100;i++){
 ages[i-1] = Integer.toString(i);
}
ageCombobox = new JComboBox<>(ages);
JLabel ageLabel = new JLabel("Age:");
ageLabel.setVerticalAlignment(JLabel.CENTER);
ageLabel.setHorizontalAlignment(JLabel.CENTER);
JPanel agePanel = new JPanel();
agePanel.setLayout(new FlowLayout());
agePanel.add(ageLabel);
agePanel.add(ageCombobox);
signUpFrame.add(agePanel);

//Mail Label
JLabel mailLabel = new JLabel("Mail:");
mailLabel.setVerticalAlignment(JLabel.CENTER);
mailLabel.setHorizontalAlignment(JLabel.CENTER);

//Mail TextField
mailTextField = new JTextField();
mailTextField.setPreferredSize(new Dimension(250,20));

//Mail Panel
JPanel mailPanel = new JPanel();
mailPanel.setLayout(new FlowLayout());
mailPanel.add(mailLabel);
mailPanel.add(mailTextField);
signUpFrame.add(mailPanel);

//Password Label

```

```

JLabel passwordLabel = new JLabel("Password:");
passwordLabel.setVerticalAlignment(JLabel.CENTER);
passwordLabel.setHorizontalAlignment(JLabel.CENTER);

//Password TextField
passwordPasswordField = new JPasswordField();
passwordPasswordField.setPreferredSize(new Dimension(250,20));

//Password Confirm Label
JLabel passwordConfirmLabel = new JLabel("Password Confirm:");
passwordConfirmLabel.setHorizontalAlignment(JLabel.CENTER);
passwordConfirmLabel.setVerticalAlignment(JLabel.CENTER);

//Password Confirm TextField
passwordConfirmPasswordField = new JPasswordField();
passwordConfirmPasswordField.setPreferredSize(new Dimension(250,20));

//Password Panel
JPanel passwordPanel = new JPanel();
passwordPanel.setLayout(new FlowLayout());
passwordPanel.add(passwordLabel);
passwordPanel.add(passwordPasswordField);
signUpFrame.add(passwordPanel);

//Password Confirm Panel
JPanel passwordConfirmPanel = new JPanel();
passwordConfirmPanel.add(passwordConfirmLabel);
passwordConfirmPanel.add(passwordConfirmPasswordField);
signUpFrame.add(passwordConfirmPanel);

//Buttons
signUpButton = new JButton("Sign Up");
backToEntranceButton = new JButton("Back to Login");
signUpButton.addActionListener(this);
backToEntranceButton.addActionListener(this);

//Button Panel
JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new FlowLayout());
buttonPanel.add(backToEntranceButton);
buttonPanel.add(signUpButton);
signUpFrame.add(buttonPanel);

//Frame
signUpFrame.setDefaultCloseOperation(WindowConstants.DO_NOTHING_ON_CLOSE);
signUpFrame.addWindowListener(this);
signUpFrame.setLayout(new GridLayout(10,1));
signUpFrame.setSize(275,600);
signUpFrame.setLocationRelativeTo(null);
signUpFrame.setResizable(false);
signUpFrame.setVisible(true);
}

@Override
public void windowClosing(WindowEvent e) {
 int selection = JOptionPane.showConfirmDialog(signUpFrame,"Are you sure?");
 if(selection == JOptionPane.YES_OPTION){
 signUpFrame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
 }
 else if(selection == JOptionPane.NO_OPTION){
 signUpFrame.setVisible(false);
 EntranceScreen.entranceFrame.setVisible(true);
 }
}

@Override
public void actionPerformed(ActionEvent e) {
 if(e.getSource() == backToEntranceButton){
 signUpFrame.setVisible(false);
 EntranceScreen.userName.setText("");
 EntranceScreen.userPassword.setText("");
 EntranceScreen.entranceFrame.setVisible(true);
 }
 else if(e.getSource() == signUpButton){
 if(new String(passwordConfirmPasswordField.getPassword()).equals(new
String(passwordPasswordField.getPassword())) && checkString(usernameTextField.getText())
&& checkString(nameTextField.getText()) && checkString(surnameTextField.getText()) &&
checkString(mailTextField.getText())
&& checkString(new String(passwordPasswordField.getPassword()) && checkString(new
String(passwordConfirmPasswordField.getPassword())) && checkRadioButton()){
 if(SearchUser.searchUser(usernameTextField.getText()){
 JOptionPane.showMessageDialog(signUpFrame,"This username is already
taken.", "Warning",JOptionPane.WARNING_MESSAGE);
 return;
 }
 writeToUserInformations();
 signUpFrame.setVisible(false);
 EntranceScreen.userName.setText("");

```

```

 EntranceScreen.userPassword.setText("");
 EntranceScreen.entranceFrame.setVisible(true);
 }
 else{
 if(!checkString(new String(passwordPasswordField.getPassword()) || !checkString(new
String(passwordConfirmPasswordField.getPassword()))){
 JOptionPane.showMessageDialog(signUpFrame,"Fill the blank
areas.", "Warning", JOptionPane.WARNING_MESSAGE);
 }
 else if(!(new String(passwordConfirmPasswordField.getPassword()).equals(new
String(passwordPasswordField.getPassword()))){
 JOptionPane.showMessageDialog(signUpFrame,"Passwords are not
matched.", "Error", JOptionPane.ERROR_MESSAGE);
 }
 else{
 JOptionPane.showMessageDialog(signUpFrame,"Fill the blank
areas.", "Warning", JOptionPane.WARNING_MESSAGE);
 }
 }
}

private boolean checkString(String toCheck){
 return !toCheck.equals("");
}

private boolean checkRadioButton(){
 return maleRadioButton.isSelected() || femaleRadioButton.isSelected();
}

private void writeToUserInformations(){
 try {
 String dataToWrite = "\n" + nameTextField.getText() + ":" + surnameTextField.getText() + ":" +
usernameTextField.getText() + ":"
 + (maleRadioButton.isSelected() ? "Male" : "Female") + ":" +
ageComboBox.getSelectedItem() + ":" + mailTextField.getText() + ":" +
 new String(passwordPasswordField.getPassword());
 FileWriter fileToWrite = new FileWriter("userfiles/userinformations.txt",true);
 fileToWrite.append(dataToWrite);
 fileToWrite.close();
 System.out.println(usernameTextField.getText()+" is added to userinformations.");
 }catch (Exception error) {
 error.printStackTrace();
 }
}
}
}

```

## SignInScreen

```

package Codes;

import javax.swing.*;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class SignInUserFile {
 static public boolean searchUser(String usernameInput, String passwordInput){
 try{
 File searchFile = new File("userfiles/userinformations.txt");
 Scanner readFile = new Scanner(searchFile);
 while(readFile.hasNextLine() && !usernameInput.equals("") && !passwordInput.equals("")){
 String data = readFile.nextLine();
 String[] splittedString = data.split(":");
 if(splittedString[2].equals(usernameInput) && splittedString[6].equals(passwordInput)){
 return true;
 }
 }
 if(usernameInput.equals("") || passwordInput.equals("")){
 JOptionPane.showMessageDialog(EntranceScreen.entranceFrame,"Enter your username and
password!!!", "Warning", JOptionPane.WARNING_MESSAGE);
 return false;
 }
 } catch (FileNotFoundException e){
 System.out.println("File not found.");
 }
 JOptionPane.showMessageDialog(EntranceScreen.entranceFrame,"Could not find the
user!!!", "Error", JOptionPane.ERROR_MESSAGE);
 return false;
 }
}

```



## ApplicationScreen

```
package Codes;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileNotFoundException;
import java.net.URI;
import java.util.ArrayList;
import java.util.Random;
import java.util.Scanner;

// Inside of application after logging in
public class ApplicationScreen extends Container implements ActionListener {
 JFrame applicationScreen = new JFrame("FilmBOX Application");
 JButton signOutButton = new JButton("Sign Out");
 JButton accountDetailsButton = new JButton("Account Details");
 JButton searchButton;
 String loggedInUser = "";
 JTextField searchTextField;
 ArrayList<JButton> categories = new ArrayList<>();
 static ArrayList<JPanel> categoryContentPanels = new ArrayList<>();

 // Frame creator method of application
 public void applicationScreenCreator(String loggedInUser){
 this.loggedInUser = loggedInUser;
 //Search Panel
 searchTextField = new JTextField();
 searchTextField.setPreferredSize(new Dimension(800,23));
 ImageIcon searchImageIcon = new ImageIcon("images/search.png");
 Image searchImage = searchImageIcon.getImage().getScaledInstance(13,13, Image.SCALE_SMOOTH);
 searchImageIcon = new ImageIcon(searchImage);

 searchButton = new JButton("Search",searchImageIcon);
 searchButton.addActionListener(this);

 JPanel searchPanel = new JPanel();
 searchPanel.add(searchTextField);
 searchPanel.add(searchButton);
 searchPanel.setLayout(new FlowLayout());

 //Center Panel
 JPanel centerPanel = new JPanel();
 centerPanel.setLayout(new BorderLayout());
 centerPanel.add(searchPanel,BorderLayout.NORTH);
 applicationScreen.add(centerPanel,BorderLayout.CENTER);

 //Categories
 categories.add(new JButton("Home Page"));
 categories.add(new JButton("Action"));
 categories.add(new JButton("Science Fiction"));
 categories.add(new JButton("Romantic"));
 categories.add(new JButton("Comedy"));
 categories.add(new JButton("Documentary"));
 categories.add(new JButton("Horror"));
 categories.add(new JButton("Thriller"));
 categories.add(new JButton("Animation"));
 categories.add(new JButton("Fantastic"));
 categories.add(new JButton("Adventure"));
 categories.add(accountDetailsButton);
 categories.add(signOutButton);

 //Logo Label
 ImageIcon logo = new ImageIcon("images/logo1.png");
 Image logoReSized = logo.getImage().getScaledInstance(200,200, Image.SCALE_SMOOTH);
 logo = new ImageIcon(logoReSized);
 JLabel logoLabel = new JLabel(logo);

 //Category Panel
 JPanel categoryPanel = new JPanel();
 categoryPanel.setLayout(new GridLayout(14,1));
 categoryPanel.add(logoLabel);
 for (JButton temp: categories) {
 categoryPanel.add(temp);
 temp.addActionListener(this);
 }

 applicationScreen.add(categoryPanel,BorderLayout.WEST);

 //Category Panels
 int i;
 for (i=0;i<11;i++){
 categoryContentPanels.add(new JPanel());
 }
 }
}
```

```

//Category Layout Setups
for (JPanel temp: categorycontentPanels){
 temp.removeAll();
}

homePageCreator();
showContent("action",1);
showContent("sciencefiction",2);
showContent("romantic",3);
showContent("comedy",4);
showContent("documentary",5);
showContent("horror",6);
showContent("thriller",7);
showContent("animation",8);
showContent("fantastic",9);
showContent("adventure",10);
JPanel contentShowPanel = new JPanel();
contentShowPanel.setLayout(new CardLayout());

contentShowPanel.add(categorycontentPanels.get(0));
contentShowPanel.add(categorycontentPanels.get(1));
contentShowPanel.add(categorycontentPanels.get(2));
contentShowPanel.add(categorycontentPanels.get(3));
contentShowPanel.add(categorycontentPanels.get(4));
contentShowPanel.add(categorycontentPanels.get(5));
contentShowPanel.add(categorycontentPanels.get(6));
contentShowPanel.add(categorycontentPanels.get(7));
contentShowPanel.add(categorycontentPanels.get(8));
contentShowPanel.add(categorycontentPanels.get(9));
contentShowPanel.add(categorycontentPanels.get(10));
centerPanel.add(contentShowPanel, BorderLayout.CENTER);

//Frame
applicationScreen.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
applicationScreen.setSize(1300,800);
applicationScreen.setMinimumSize(new Dimension(1300,800));
applicationScreen.setLocationRelativeTo(null);
applicationScreen.setVisible(true);
}

// The panel that is used to get random content
private void homePageCreator() { // Home Page designation
 categorycontentPanels.get(0).setLayout(new BorderLayout());

 //North Panel
 JPanel northPanel = new JPanel();
 northPanel.setLayout(new BorderLayout());

 //North Panel of North Panel
 JPanel northOfNorthPanel = new JPanel();
 northOfNorthPanel.setLayout(new FlowLayout());

 //RadioButtons
 ArrayList<JRadioButton> categoryRadios = new ArrayList<>();
 ButtonGroup radioButtonGroup = new ButtonGroup();
 for(int i=0;i<10;i++){
 switch (i){
 case 0 -> categoryRadios.add(new JRadioButton("Action"));
 case 1 -> categoryRadios.add(new JRadioButton("Science Fiction"));
 case 2 -> categoryRadios.add(new JRadioButton("Romantic"));
 case 3 -> categoryRadios.add(new JRadioButton("Comedy"));
 case 4 -> categoryRadios.add(new JRadioButton("Documentary"));
 case 5 -> categoryRadios.add(new JRadioButton("Horror"));
 case 6 -> categoryRadios.add(new JRadioButton("Thriller"));
 case 7 -> categoryRadios.add(new JRadioButton("Animation"));
 case 8 -> categoryRadios.add(new JRadioButton("Fantastic"));
 case 9 -> categoryRadios.add(new JRadioButton("Adventure"));
 }
 radioButtonGroup.add(categoryRadios.get(i));
 northOfNorthPanel.add(categoryRadios.get(i));
 }
 JLabel categoryLabel = new JLabel("Select a Category to Get Suggestion.");
 categoryLabel.setFont(new Font("Serif", Font.BOLD, 14));
 categoryLabel.setVerticalAlignment(JLabel.CENTER);
 categoryLabel.setHorizontalAlignment(JLabel.CENTER);
 northPanel.add(categoryLabel, BorderLayout.NORTH);
 northPanel.add(northOfNorthPanel, BorderLayout.CENTER);
 JButton luckyButton = new JButton("I Feel Lucky.");
 luckyButton.addActionListener(e -> {
 int flag = 0;
 String categoryInfo="";
 for(JRadioButton temp: categoryRadios){
 if(temp.isSelected()){
 categoryInfo = temp.getText().replaceAll(" ", "").toLowerCase();
 flag = 1;
 break;
 }
 }
 })
}

```

```

 }
 if(flag == 1){
 try {
 String line;
 String[] getName;
 ArrayList <String> relatedContent = new ArrayList<>();
 File allFile = new File("database/all/allcontent.txt");
 Scanner scanAllFile = new Scanner(allFile);
 while (scanAllFile.hasNextLine()){
 line = scanAllFile.nextLine();
 if(line.contains(categoryInfo)){
 getName = line.split("■");
 relatedContent.add(getName[0]);
 }
 }
 }
 try {
 Random randNumberGenerator = new Random();
 int randomNumber = randNumberGenerator.nextInt(relatedContent.size());
 ImageIcon randomContent = new
ImageIcon("database/"+categoryInfo+"/"+relatedContent.get(randomNumber)+".jpg");
 Image reSized =
randomContent.getImage().getScaledInstance(300,400,Image.SCALE_SMOOTH);
 randomContent = new ImageIcon(reSized);
 categorycontentPanels.get(0).removeAll();
 homePageCreator();
 JButton contentButton = new JButton(randomContent);
 String finalCategoryInfo = categoryInfo;
 contentButton.addActionListener(e1 -> {
 try {
 String[] data;
 data =
insideofContentFile(finalCategoryInfo,relatedContent.get(randomNumber));
 contentInformationScreen(data);
 }catch (Exception error){
 error.printStackTrace();
 }

 });
 categorycontentPanels.get(0).add(contentButton,BorderLayout.CENTER);
 categorycontentPanels.get(0).setVisible(false);
 categorycontentPanels.get(0).setVisible(true);
 }catch (Exception insideError){
 insideError.printStackTrace();
 }
 }catch (Exception error){
 error.printStackTrace();
 }
}
else{
 JOptionPane.showMessageDialog(applicationScreen,"Please select a
category.", "Warning",JOptionPane.WARNING_MESSAGE);
}
});
northPanel.add(luckyButton,BorderLayout.SOUTH);
categorycontentPanels.get(0).add(northPanel, BorderLayout.NORTH);
}

// Action listeners of components
@Override
public void actionPerformed(ActionEvent e) {
 if (e.getSource() == signOutButton){
 EntranceScreen.userName.setText("");
 EntranceScreen.userPassword.setText("");
 applicationScreen.setVisible(false);
 EntranceScreen.entranceFrame.setVisible(true);
 }
 else if (e.getSource() == accountDetailsButton){
 showUserInformationInApplication();
 }
 else if (e.getSource() == searchButton){
 if(searchTextField.getText().equals("")){
 JOptionPane.showMessageDialog(applicationScreen,"Some input should be
entered.", "Warning",JOptionPane.WARNING_MESSAGE);
 }
 else{
 JFrame searchFrame = new JFrame("Search");
 searchFrame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
 try {
 int flag=0;
 File searchDatabase = new File("database/all/allcontent.txt");
 Scanner scanDatabase = new Scanner(searchDatabase);
 String[] words;
 String[] infoOfContent = new String[2];
 words = searchTextField.getText().split("■");
 loop:
 while (scanDatabase.hasNextLine()){
 infoOfContent = scanDatabase.nextLine().split("■");
 for(String temp: words){

```

```

 if(infoOfContent[0].contains(temp)){
 flag = 1;
 break loop;
 }
 }
}
scanDatabase.close();

if(flag == 1){
 try {
 ImageIcon searchedContent = new ImageIcon("database/"+ infoOfContent[1] + "/" +
infoOfContent[0] + ".jpg");
 Image reSized =
searchedContent.getImage().getScaledInstance(300,400,Image.SCALE_SMOOTH);
 searchedContent = new ImageIcon(reSized);
 JButton searchedContentButton = new JButton(searchedContent);
 String[] finalInfoOfContent = infoOfContent;
 searchedContentButton.addActionListener(e1 ->
contentInformationScreen(insideOfContentFile(finalInfoOfContent[1], finalInfoOfContent[0])));
 //Frame
 JFrame searchedContentFrame = new JFrame("Searched Content");
 searchedContentFrame.setLayout(new BorderLayout());

searchedContentFrame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
 searchedContentFrame.add(searchedContentButton,BorderLayout.CENTER);
 JLabel infoLabel = new JLabel("Click on the image to see information.");
 infoLabel.setFont(new Font("Serif", Font.BOLD, 14));
 infoLabel.setHorizontalAlignment(JLabel.CENTER);
 infoLabel.setVerticalAlignment(JLabel.CENTER);
 searchedContentFrame.add(infoLabel,BorderLayout.NORTH);
 searchedContentFrame.setSize(500,500);
 searchedContentFrame.setLocationRelativeTo(null);
 searchedContentFrame.setResizable(false);
 searchedContentFrame.setVisible(true);

 }catch (Exception error){
 error.printStackTrace();
 }
 }
 else{
 JOptionPane.showMessageDialog(applicationScreen,"Could not find any
content.", "Info",JOptionPane.INFORMATION_MESSAGE);
 }

 }catch (Exception error){
 error.printStackTrace();
 }
}

else if (e.getSource() == categories.get(0)){
 int i=0;
 while(i!=11){
 categorycontentPanels.get(i).setVisible(i == 0);
 i++;
 }
}
else if (e.getSource() == categories.get(1)){
 int i=0;
 while(i!=11){
 categorycontentPanels.get(i).setVisible(i == 1);
 i++;
 }
}
else if (e.getSource() == categories.get(2)){
 int i=0;
 while(i!=11){
 categorycontentPanels.get(i).setVisible(i == 2);
 i++;
 }
}
else if (e.getSource() == categories.get(3)){
 int i=0;
 while(i!=11){
 categorycontentPanels.get(i).setVisible(i == 3);
 i++;
 }
}
else if (e.getSource() == categories.get(4)){
 int i=0;
 while(i!=11){
 categorycontentPanels.get(i).setVisible(i == 4);
 i++;
 }
}
else if (e.getSource() == categories.get(5)){
 int i=0;
 while(i!=11){

```

```

 categorycontentPanels.get(i).setVisible(i == 5);
 i++;
 }
}
else if (e.getSource() == categories.get(6)){
 int i=0;
 while(i!=11){
 categorycontentPanels.get(i).setVisible(i == 6);
 i++;
 }
}
else if (e.getSource() == categories.get(7)){
 int i=0;
 while(i!=11){
 categorycontentPanels.get(i).setVisible(i == 7);
 i++;
 }
}
else if (e.getSource() == categories.get(8)){
 int i=0;
 while(i!=11){
 categorycontentPanels.get(i).setVisible(i == 8);
 i++;
 }
}
else if (e.getSource() == categories.get(9)){
 int i=0;
 while(i!=11){
 categorycontentPanels.get(i).setVisible(i == 9);
 i++;
 }
}
else if (e.getSource() == categories.get(10)){
 int i=0;
 while(i!=11){
 categorycontentPanels.get(i).setVisible(i == 10);
 i++;
 }
}
}

// Method that is used to find entered user
public void showUserInformationInApplication(){
 try {
 File informationFile = new File("userfiles/userinformations.txt");
 Scanner scanInformationFile = new Scanner(informationFile);
 String data;
 String[] userInformation = new String[7];
 while (scanInformationFile.hasNextLine()){
 data = scanInformationFile.nextLine();
 userInformation = data.split(":");
 if(userInformation[2].equals(loginedUser)){
 break;
 }
 }
 userInformationFrame(userInformation);
 }catch (FileNotFoundException error) {
 error.printStackTrace();
 }
}

// Method that is used to create frame to show entered user's informations
public void userInformationFrame(String[] userInformation){
 JFrame informationFrame = new JFrame("User Informations");

 //Information Labels
 ArrayList<JLabel> labels = new ArrayList<>();
 JLabel name = new JLabel("Name: " + userInformation[0]);
 JLabel surname = new JLabel("Surname: " + userInformation[1]);
 JLabel username = new JLabel("Username: " + userInformation[2]);
 JLabel gender = new JLabel("Gender: " + userInformation[3]);
 JLabel age = new JLabel("Age: " + userInformation[4]);
 JLabel mail = new JLabel("Mail: " + userInformation[5]);

 labels.add(name);
 labels.add(surname);
 labels.add(username);
 labels.add(gender);
 labels.add(age);
 labels.add(mail);

 for (JLabel temp: labels) {
 temp.setVerticalAlignment(JLabel.CENTER);
 temp.setHorizontalAlignment(JLabel.CENTER);
 temp.setFont(new Font(Font.SANS_SERIF, Font.BOLD, 22));
 temp.setForeground(Color.white);
 }
}

```

```

 informationFrame.add(name);
 informationFrame.add(surname);
 informationFrame.add(username);
 informationFrame.add(gender);
 informationFrame.add(age);
 informationFrame.add(mail);

 //Ok Button
 ImageIcon tickImageIcon = new ImageIcon("images/tickicon.png");
 Image tickImage = tickImageIcon.getImage().getScaledInstance(20,20, Image.SCALE_SMOOTH);
 tickImageIcon = new ImageIcon(tickImage);
 JButton okButton = new JButton("OK", tickImageIcon);

 okButton.addActionListener(e -> informationFrame.setVisible(false));
 informationFrame.add(okButton);

 //Frame
 informationFrame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
 informationFrame.setSize(400,400);
 informationFrame.setLocationRelativeTo(null);
 informationFrame.setLayout(new GridLayout(7,1));
 informationFrame.getContentPane().setBackground(Color.GRAY);
 informationFrame.setResizable(false);
 informationFrame.setVisible(true);
 }

 // Method that counts the contents in related category
 private int contentCounter(String contentcategory){
 int numberOfLines=0;
 try {
 File contentIncludedInCategoryFile = new
File("database/"+contentcategory+"/"+contentcategory+"content.txt");
 Scanner contentIncludedCategory = new Scanner(contentIncludedInCategoryFile);
 while (contentIncludedCategory.hasNextLine()){
 contentIncludedCategory.nextLine();
 numberOfLines++;
 }
 contentIncludedCategory.close();
 } catch (Exception e) {
 e.printStackTrace();
 }
 return numberOfLines;
 }

 // Method that shows the image of content on buttons
 public void showContent(String contentCategory, int index){ // all in lower
 int numberOfContent = contentCounter(contentCategory);
 categorycontentPanels.get(index).setLayout(new GridLayout((int) Math.ceil(numberOfContent/2),2));
 try{
 File contentIncludedInCategoryFile = new File("database/"+ contentCategory + "/" +
contentCategory + "content.txt");
 Scanner contentIncludedCategory = new Scanner(contentIncludedInCategoryFile);
 while(contentIncludedCategory.hasNextLine()){
 String nameOfContent = contentIncludedCategory.nextLine();
 ImageIcon contentImage = new ImageIcon("database/"+ contentCategory
+ "/" + nameOfContent + ".jpg");
 Image resizedImage = contentImage.getImage().getScaledInstance(175,225,Image.SCALE_SMOOTH);
 contentImage = new ImageIcon(resizedImage);
 JButton newContent = new JButton(contentImage);
 newContent.addActionListener(e -> {
 String[] contentDetails = insideofContentFile(contentCategory, nameOfContent);
 contentInformationScreen(contentDetails);
 });
 categorycontentPanels.get(index).add(newContent);
 }
 } catch (Exception error) {
 error.printStackTrace();
 }
 }

 // Method that shows information of content when clicked on content's button
 private void contentInformationScreen(String[] contentDetails){
 ArrayList <JComponent> labels = new ArrayList<>();
 for (int i=0;i<8;i++){
 switch (i) {
 case 0 -> {
 labels.add(new JLabel("Category: " + contentDetails[0]));
 ((JLabel) labels.get(i)).setHorizontalAlignment(JLabel.CENTER);
 ((JLabel) labels.get(i)).setVerticalAlignment(JLabel.CENTER);
 }
 case 1 -> {
 labels.add(new JLabel("Name: " + contentDetails[1]));
 ((JLabel) labels.get(i)).setHorizontalAlignment(JLabel.CENTER);
 ((JLabel) labels.get(i)).setVerticalAlignment(JLabel.CENTER);
 }
 case 2 -> {
 labels.add(new JLabel("Year: " + contentDetails[2]));
 ((JLabel) labels.get(i)).setHorizontalAlignment(JLabel.CENTER);
 }
 }
 }

```

```

 ((JLabel) labels.get(i)).setVerticalAlignment(JLabel.CENTER);
 }
 case 3 -> {
 labels.add(new JLabel("Duration (Minutes): " + contentDetails[3]));
 ((JLabel) labels.get(i)).setHorizontalAlignment(JLabel.CENTER);
 ((JLabel) labels.get(i)).setVerticalAlignment(JLabel.CENTER);
 }
 case 4 -> {
 labels.add(new JLabel("IMDB Rating: " + contentDetails[4]));
 ((JLabel) labels.get(i)).setHorizontalAlignment(JLabel.CENTER);
 ((JLabel) labels.get(i)).setVerticalAlignment(JLabel.CENTER);
 }
 case 5 -> {
 labels.add(new JLabel("Director: " + contentDetails[5]));
 ((JLabel) labels.get(i)).setHorizontalAlignment(JLabel.CENTER);
 ((JLabel) labels.get(i)).setVerticalAlignment(JLabel.CENTER);
 }
 case 6 -> {
 labels.add(new JLabel("Actors/Actresses: " + contentDetails[6]));
 ((JLabel) labels.get(i)).setHorizontalAlignment(JLabel.CENTER);
 ((JLabel) labels.get(i)).setVerticalAlignment(JLabel.CENTER);
 }
 case 7 -> {
 JButton trailerButton = new JButton("Watch Trailer");
 trailerButton.addActionListener(e -> {
 try {
 Desktop.getDesktop().browse(new URI("https://www.youtube.com/watch?v="+
contentDetails[7]));
 } catch (Exception error) {
 error.printStackTrace();
 }
 });
 labels.add(trailerButton);
 }
}
labels.get(i).setFont(new Font("Serif", Font.BOLD, 14));
}

JFrame frame = new JFrame(contentDetails[1]);
frame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
frame.setSize(500,300);
frame.setResizable(false);
frame.setLocationRelativeTo(null);
for (JComponent temp: labels){
 frame.add(temp);
}
frame.setLayout(new GridLayout(8,1));
frame.setVisible(true);
}

// Method that returns the information exist in related content's file.
public String[] insideofContentFile(String contentCategory, String nameOfContent){
 String[] returnString = new String[8];
 try {
 String line;
 int i=0;
 File content = new File("database/"+ contentCategory + "/" + nameOfContent + ".txt");
 Scanner scanContent = new Scanner(content);
 while (scanContent.hasNextLine()){
 line = scanContent.nextLine();
 switch (i) {
 case 0 -> returnString[0] = line;
 case 2 -> returnString[1] = line;
 case 3 -> returnString[2] = line;
 case 4 -> returnString[3] = line;
 case 5 -> returnString[4] = line;
 case 6 -> returnString[5] = line;
 case 7 -> returnString[6] = line;
 case 8 -> returnString[7] = line;
 }
 i++;
 }
 } catch (Exception error) {
 error.printStackTrace();
 }
 return returnString;
}
}

```

## Video Presentation

[https://drive.google.com/file/d/1xaf8m\\_lZ0qlzXySKhbSIwao6AGKefDF7/view?usp=sharing](https://drive.google.com/file/d/1xaf8m_lZ0qlzXySKhbSIwao6AGKefDF7/view?usp=sharing)