

|  |  |  |
| --- | --- | --- |
| Acknowledgement |  |  |

**I would like to express my sincere gratitude to several individuals who have supported me throughout the development of this project.**

**First and foremost, I would like to thank my computer teacher, Sir Subhajit Dutta, for his guidance and encouragement, which have been instrumental in enhancing my technical knowledge and shaping the direction of this project. He had also provided the idea of the interactive seat selection feature in the movie booking system.**

**I am also deeply grateful to my school's principal, Br. Solomon Morris, for providing a conducive learning environment that fostered my growth and allowed me to explore my interests in technology.**

**Lastly, I extend my heartfelt thanks to my parents for their unwavering support throughout this journey. Their belief in my abilities has been a constant source of inspiration.**

|  |  |  |
| --- | --- | --- |
| Introduction |  |  |

**Introducing SpectraFlix, a movie streaming and booking application crafted using Java and XML within the Android development environment. SpectraFlix offers a seamless and flexible platform for movie enthusiasts, providing multiple ways to enjoy films. Users can stream movies directly from WebView, input URLs, or access movies stored on their devices. Additionally, trailers can be viewed effortlessly through a built-in YouTube WebView.**

**The booking feature in SpectraFlix sets it apart, allowing users to browse available films and select seats using a visually interactive interface, making the reservation process intuitive and engaging. Built with performance and user experience in mind, the app's robust back-end and fluid design ensure a smooth, responsive experience.**

**Furthermore, SpectraFlix enables users to expand their movie library by adding new films for both streaming and booking, giving them full control over their entertainment.**

|  |  |  |
| --- | --- | --- |
| Software Requirements |  |  |

**Operating System :**

* **Windows(Android Studio) / Android(AIDE) : For Development**
* **Android 8.1.0 or above [Tested by the developer] : For testing and deployment**

**Development Tools :**

* **Java Development Kit (JDK) : Version 6 or higher (Tested)**
* **AIDE : Latest Stable Version**
* **Android Studio : Latest Stable Version**
* **Gradle**

**Languages :**

* **Java : For the app's backend**
* **XML : For designing the user interface (UI)**

**Third Party Tools :**

* **YouTube IFrame API : For embedding trailers via WebView**

**Version Control :**

* **Git : For version control and code management**
* **Github : To store the project repository**

**Device Requirements (For running the application) :**

* **Android phone or tablet with sufficient storage and processing capability (minimum 2 GB RAM, 16 GB storage recommended)**

****

|  |  |  |
| --- | --- | --- |
| MainActivity.java |  |  |

package com.app.spectraflix;

import android.app.Activity;

import android.os.Bundle;

import android.content.Intent;

import android.view.View;

import android.widget.ImageButton;

import android.graphics.drawable.ColorDrawable;

import android.graphics.Color;

import android.app.ActionBar;

import android.widget.ImageButton;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.view.Gravity;

import android.graphics.Typeface;

import android.widget.TextView;

import android.util.DisplayMetrics;

import android.net.Uri;

import java.io.File;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.Manifest;

import android.content.pm.PackageManager;

import java.util.List;

public class MainActivity extends Activity

{

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

ActionBar actionBar = getActionBar();

if(actionBar != null)

actionBar.setBackgroundDrawable(new ColorDrawable(Color.parseColor("#ACD5C2")));

if (checkSelfPermission(Manifest.permission.READ\_EXTERNAL\_STORAGE) != PackageManager.PERMISSION\_GRANTED)

requestPermissions(new String[]{Manifest.permission.READ\_EXTERNAL\_STORAGE}, 1);

Typeface customFont = Typeface.createFromAsset(getAssets(), "fonts/nasalization.ttf");

TextView textView1 = findViewById(R.id.category\_label\_1);

textView1.setTypeface(customFont);

TextView textView2 = findViewById(R.id.category\_label\_2);

textView2.setTypeface(customFont);

TextView textView3 = findViewById(R.id.category\_label\_3);

textView3.setTypeface(customFont);

TextView textView4 = findViewById(R.id.category\_label\_4);

textView4.setTypeface(customFont);

ImageButton button2 = findViewById(R.id.icon\_button2);

button2.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), activity2.class);

startActivity(intent);

overridePendingTransition(0, 0);

}

});

ImageButton button3 = findViewById(R.id.icon\_button3);

button3.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), activity3.class);

startActivity(intent);

overridePendingTransition(0, 0);

}

});

DisplayMetrics displayMetrics = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

int width = displayMetrics.widthPixels;

DataManager dataManager = (DataManager) getApplication();

List<Movie> movies = dataManager.getMovies();

streamingMovies categories = new streamingMovies();

for(Movie mov : movies)

{

switch(mov.getCategory())

{

case 1:

categories.addAnimatedMovie(mov);

break;

case 2:

categories.addPopularMovie(mov);

break;

case 3:

categories.addHorrorMovie(mov);

break;

case 4:

categories.addDevelopersChoiceMovie(mov);

break;

}

}

List<Movie> animatedMovies = categories.getAnimatedMovies();

List<Movie> popularMovies = categories.getPopularMovies();

List<Movie> horrorMovies = categories.getHorrorMovies();

List<Movie> developersChoiceMovies = categories.getDevelopersChoiceMovies();

List<Movie>[] moviesList = {animatedMovies, popularMovies, horrorMovies, developersChoiceMovies};

int[] scrollViews = {R.id.animated\_movies\_scroll\_view, R.id.popular\_movies\_scroll\_view, R.id.horror\_movies\_scroll\_view, R.id.developers\_choice\_scroll\_view};

for(int i = 0 ; i < 4 ; i++)

for(Movie mov : moviesList[i])

addStreamingMovie(scrollViews[i], mov, width);

}

public void addStreamingMovie(int scrollView, Movie mov, int width)

{

LinearLayout parentLayout = findViewById(scrollView);

LinearLayout newLayout = new LinearLayout(this);

newLayout.setLayoutParams(new LinearLayout.LayoutParams(

LinearLayout.LayoutParams.WRAP\_CONTENT,

LinearLayout.LayoutParams.MATCH\_PARENT));

newLayout.setOrientation(LinearLayout.VERTICAL);

newLayout.setGravity(Gravity.CENTER);

ImageView movieTitleView = new ImageView(this);

LinearLayout.LayoutParams movieTitleParams = new LinearLayout.LayoutParams(200, 150);

movieTitleParams.gravity = Gravity.CENTER;

movieTitleView.setLayoutParams(movieTitleParams);

if(mov.getResTitleImagePath() == -1)

{

try

{

File file = new File(mov.getTitleImagePath());

Uri imageUri = Uri.fromFile(file);

Bitmap bitmap = BitmapFactory.decodeStream(getContentResolver().openInputStream(imageUri));

movieTitleView.setImageBitmap(bitmap);

}

catch(Exception e)

{}

}

else

movieTitleView.setImageResource(mov.getResTitleImagePath());

movieTitleView.setScaleType(ImageView.ScaleType.FIT\_CENTER);

ImageView moviePosterView = new ImageView(this);

LinearLayout.LayoutParams moviePosterParams = new LinearLayout.LayoutParams(

(int) Math.round(width \* 0.75), 400);

moviePosterParams.gravity = Gravity.CENTER;

moviePosterView.setLayoutParams(moviePosterParams);

if(mov.getResImagePath() == -1)

{

try

{

File file = new File(mov.getImagePath());

Uri imageUri = Uri.fromFile(file);

Bitmap bitmap = BitmapFactory.decodeStream(getContentResolver().openInputStream(imageUri));

moviePosterView.setImageBitmap(bitmap);

}

catch(Exception e)

{}

}

else

moviePosterView.setImageResource(mov.getResImagePath());

moviePosterView.setScaleType(ImageView.ScaleType.FIT\_CENTER);

moviePosterView.setPadding((int) Math.round(width \* 0.05), 0, (int) Math.round(width \* 0.05), 0);

ImageButton imageButton = new ImageButton(this);

LinearLayout.LayoutParams buttonParams = new LinearLayout.LayoutParams(100, 75);

buttonParams.gravity = Gravity.CENTER;

imageButton.setLayoutParams(buttonParams);

imageButton.setImageResource(R.drawable.watching);

imageButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

imageButton.setBackground(null);

final Movie mov\_final = mov;

imageButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.putExtra("Movie", mov\_final);

intent.setClass(getApplicationContext(), PreviewActivity.class);

startActivity(intent);

}

});

newLayout.addView(movieTitleView);

newLayout.addView(moviePosterView);

newLayout.addView(imageButton);

parentLayout.addView(newLayout);

}

}

|  |  |  |
| --- | --- | --- |
| BookableMovie.java |  |  |

package com.app.spectraflix;

import java.io.Serializable;

import java.util.List;

import java.util.LinkedList;

public class BookableMovie implements Serializable

{

private int resMovieTitlePath;

private String movieName, movieTitlePath;

public List<Cinema> cinemas = new LinkedList<Cinema>();

public BookableMovie(String movieName, int resMovieTitlePath, String movieTitlePath)

{

this.movieName = movieName;

this.resMovieTitlePath = resMovieTitlePath;

this.movieTitlePath = movieTitlePath;

}

public List<Cinema> getCinemas()

{

return this.cinemas;

}

public void addCinema(Cinema cinema)

{

this.cinemas.add(cinema);

}

public void removeCinema(Cinema cinema)

{

this.cinemas.remove(cinema);

}

public String getMovieName()

{

return this.movieName;

}

public String getMovieTitlePath()

{

return this.movieTitlePath;

}

public int getResMovieTitlePath()

{

return this.resMovieTitlePath;

}

}

|  |  |  |
| --- | --- | --- |
| Cinema.java |  |  |

package com.app.spectraflix;

import java.util.List;

import java.util.LinkedList;

import java.time.LocalDateTime;

import java.io.Serializable;

import java.time.format.DateTimeFormatter;

public class Cinema implements Serializable

{

private String name;

private int[] seatRowCol;

public List<Runtime> runtimes = new LinkedList<Runtime>();

public Cinema(String name, int numSeats)

{

this.name = name;

this.seatRowCol = this.getMinRowCol(numSeats);

}

public String getName()

{

return this.name;

}

public List<Runtime> getRuntimes()

{

return this.runtimes;

}

public boolean addRuntime(String startDateTime, String endDateTime)

{

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm");

LocalDateTime start = LocalDateTime.parse(startDateTime, formatter);

LocalDateTime end = LocalDateTime.parse(endDateTime, formatter);

for(Runtime rt : this.runtimes)

{

LocalDateTime startRt = rt.getStartDateTime(), endRt = rt.getEndDateTime();

if((startRt.isAfter(start) && startRt.isBefore(end)) || (endRt.isAfter(start) && endRt.isBefore(end)) || (start.isAfter(startRt) && start.isBefore(endRt)) || (end.isAfter(startRt) && end.isBefore(endRt)) || (start.isEqual(startRt) && end.isEqual(endRt)))

return false;

}

Runtime runtime = new Runtime(startDateTime, endDateTime, this.seatRowCol);

runtimes.add(runtime);

return true;

}

public Runtime getRuntime(int index)

{

return this.runtimes.get(index);

}

public int getRuntimesSize()

{

return this.runtimes.size();

}

public boolean removeRuntime(LocalDateTime start)

{

for(int i = 0 ; i < this.runtimes.size() ; i++)

{

Runtime runtime = runtimes.get(i);

if(runtime.getStartDateTime().isEqual(start))

{

runtimes.remove(i);

return true;

}

}

return false;

}

public int[] getMinRowCol(int num)

{

List<int[]> listOfFactors = new LinkedList<int[]>();

for(int i = 1 ; i <= num ; i++)

{

if(num % i != 0)

continue;

else if(i == num / i)

return new int[]{i, i};

int[] factors = {i, num / i};

listOfFactors.add(factors);

}

int[] outArr = {num, 1};

int diff = num - 1;

for(int[] arr : listOfFactors)

{

int numDiff = arr[0] - arr[1];

if(numDiff > 0 && numDiff < diff)

{

outArr = arr;

diff = numDiff;

}

}

return outArr;

}

}

|  |  |  |
| --- | --- | --- |
| DataManager.java |  |  |

package com.app.spectraflix;

import android.app.Application;

import java.util.LinkedList;

import java.util.List;

public class DataManager extends Application

{

private List<Movie> movies;

public List<BookableMovie> bookableMovies;

@Override

public void onCreate()

{

super.onCreate();

this.movies = new LinkedList<Movie>();

this.movies.add(new Movie("Moana", "", "", R.drawable.moana, R.drawable.moana\_title, "LKFuXETZUsI", "", "https://commondatastorage.googleapis.com/gtv-videos-bucket/sample/BigBuckBunny.mp4", "", "Moana, daughter of chief Tui, embarks on a journey to return the heart of goddess Te Fitti from Maui, a demigod, after the plants and the fish on her island start dying due to a blight.", 1));

this.movies.add(new Movie("Zootopia", "", "", R.drawable.zootopia, R.drawable.zootopia\_title, "jWM0ct-OLsM", "<iframe width = \"100%\" height=\"100%\" src = \"https://drive.google.com/file/d/1LUvPO6INLgLZg2EegeKuTDXKe5uKp2ct/preview\" allow = \"autoplay\" sandbox = \"allow-same-origin allow-scripts\" allowfullscreen = \"true\" scrolling = \"no\"></iframe>", "", "", "When Judy Hopps, a rookie officer in the Zootopia Police Department, sniffs out a sinister plot, she enlists the help of a con artist to solve the case in order to prove her abilities to Chief Bogo.", 1));

this.movies.add(new Movie("Toy Story 3", "", "", R.drawable.toy\_story\_3, R.drawable.toy\_story\_3\_title, "2BlMNH1QTeE", "", "https://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ForBiggerBlazes.mp4", "", "Before Andy leaves for college, his toys are mistakenly delivered to a day care centre. Woody convinces the other toys that they were not dumped and leads them on an expedition back home.", 1));

this.movies.add(new Movie("Frozen II", "", "", R.drawable.frozen\_2, R.drawable.frozen\_2\_title, "Zi4LMpSDccc", "<iframe width = \"100%\" height=\"100%\" src = \"https://drive.google.com/file/d/1FilPpbqmjmbtYBu\_1k7QJ-1G1tQZGzLN/preview\" allow = \"autoplay\" sandbox = \"allow-same-origin allow-scripts\" allowfullscreen = \"true\" scrolling = \"no\"></iframe>", "", "", "Three years after her coronation when Elsa celebrates Autumn, she hears a mysterious voice. She follows that voice along with Anna and their friends as they are forced to uncover the truth.", 1));

this.movies.add(new Movie("Oppenheimer", "", "", R.drawable.oppenheimer, R.drawable.oppenheimer\_title, "uYPbbksJxIg", "", "https://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ForBiggerFun.mp4", "", "During World War II, Lt. Gen. Leslie Groves Jr. appoints physicist J. Robert Oppenheimer to work on the top-secret Manhattan Project. Oppenheimer and a team of scientists spend years developing and designing the atomic bomb. Their work comes to fruition on July 16, 1945, as they witness the world's first nuclear explosion, forever changing the course of history.", 2));

this.movies.add(new Movie("Free Guy", "", "", R.drawable.free\_guy, R.drawable.free\_guy\_title, "X2m-08cOAbc", "", "https://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ForBiggerJoyrides.mp4", "", "Guy lives a seemingly peaceful life as a bank teller. However, an encounter with a pretty but mysterious woman makes him realise that he is a non-playable character in a massive online video game.", 2));

this.movies.add(new Movie("Kill", "", "", R.drawable.kill, R.drawable.kill\_title, "da7lKeeS67c", "", "https://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ForBiggerMeltdowns.mp4", "", "A train to New Delhi becomes a combat battleground as a pair of commandos faces off against an army of invading bandits.", 2));

this.movies.add(new Movie("Srikanth", "", "", R.drawable.srikanth, R.drawable.srikanth\_title, "OMeuJP5iBWY", "", "", "", "In a world where the odds are against him, Srikanth defies the norms and makes his way from rural India to becoming the first visually impaired student at MIT.", 2));

this.movies.add(new Movie("The Conjuring", "", "", R.drawable.the\_conjuring, R.drawable.the\_conjuring\_title, "k10ETZ41q5o", "", "https://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ElephantsDream.mp4", "", "Rod and Carolyn find their pet dog dead under mysterious circumstances and experience a spirit that harms their daughter Andrea. They finally call investigators who can help them get out of the mess.", 3));

this.movies.add(new Movie("The Conjuring 2", "", "", R.drawable.the\_conjuring\_2, R.drawable.the\_conjuring\_2\_title, "VFsmuRPClr4", "<iframe width = \"100%\" height=\"100%\" src = \"https://drive.google.com/file/d/1B6OqDkR45Z2MmPkCoRvKEkg6b5Ol7DWv/preview\" allow = \"autoplay\" sandbox = \"allow-same-origin allow-scripts\" allowfullscreen = \"true\" scrolling = \"no\"></iframe>", "", "", "Peggy, a single mother of four children, seeks the help of occult investigators Ed and Lorraine Warren when she and her children witness strange, paranormal events in their house.", 3));

this.movies.add(new Movie("Annabelle Creation", "", "", R.drawable.annabelle\_creation, R.drawable.annabelle\_creation\_title, "KisPhy7T\_\_Q", "", "https://commondatastorage.googleapis.com/gtv-videos-bucket/sample/Sintel.mp4", "", "A doll-maker and his wife embed the spirit of their deceased daughter inside a doll. Years later, a nun and several girls from a shuttered orphanage become victims of the possessed doll, Anabelle.", 3));

this.movies.add(new Movie("The Nun II", "", "", R.drawable.the\_nun\_2, R.drawable.the\_nun\_2\_title, "QF-oyCwaArU", "", "", "", "In 1956 France, a priest is murdered, and it seems an evil is spreading. Sister Irene once again comes face to face with a demonic force.", 3));

this.movies.add(new Movie("Interstellar", "", "", R.drawable.interstellar, R.drawable.interstellar\_title, "zSWdZVtXT7E", "", "https://commondatastorage.googleapis.com/gtv-videos-bucket/sample/TearsOfSteel.mp4", "", "When Earth becomes uninhabitable in the future, a farmer and ex-NASA pilot, Joseph Cooper, is tasked to pilot a spacecraft, along with a team of researchers, to find a new planet for humans.", 4));

this.movies.add(new Movie("The Boy In The Striped Pyjamas", "", "", R.drawable.the\_boy\_in\_the\_striped\_pyjamas, R.drawable.the\_boy\_in\_the\_striped\_pyjamas\_title, "uwysOfnX2Qg", "<iframe width = \"100%\" height=\"100%\" src = \"https://drive.google.com/file/d/17VcFkKk7mvDxVvqRE5E41KcA5Ziby9Ob/preview\" allow = \"autoplay\" sandbox = \"allow-same-origin allow-scripts\" allowfullscreen = \"true\" scrolling = \"no\"></iframe>", "", "", "Bruno, the son of a Nazi commander, meets Shmuel, a Jewish boy living in a concentration camp. Later, both the children become friends while being oblivious to the reality of the situation.", 4));

this.movies.add(new Movie("RRR", "", "", R.drawable.rrr, R.drawable.rrr\_title, "GY4BgdUSpbE", "", "http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ForBiggerEscapes.mp4", "", "A fearless revolutionary and an officer in the British force, who once shared a deep bond, decide to join forces and chart out an inspirational path of freedom against the despotic rulers.", 4));

this.movies.add(new Movie("Ralph Breaks The Internet", "", "", R.drawable.ralph\_breaks\_the\_internet, R.drawable.ralph\_breaks\_the\_internet\_title, "\_BcYBFC6zfY", "", "", "", "On a quest to save the video game 'Sugar Rush' and to find a replacement, Ralph and his best friend Vanellope travel to the World Wide Web through a Wi-Fi router they find at the arcade.", 4));

this.bookableMovies = new LinkedList<BookableMovie>();

Cinema cinema1 = new Cinema("Theater A", 24);

cinema1.addRuntime("2024-10-14 11:15", "2024-10-14 14:00");

cinema1.addRuntime("2024-10-15 11:15", "2024-10-15 14:00");

Cinema cinema2 = new Cinema("Theater B", 36);

cinema2.addRuntime("2024-10-17 07:00", "2024-10-17 10:00");

cinema2.addRuntime("2024-10-15 17:35", "2024-10-15 20:20");

BookableMovie movie1 = new BookableMovie("Free Guy", R.drawable.free\_guy\_title,null);

movie1.addCinema(cinema1);

movie1.addCinema(cinema2);

BookableMovie movie2 = new BookableMovie("Moana", R.drawable.moana\_title, null);

movie2.addCinema(cinema2);

BookableMovie movie3 = new BookableMovie("Interstellar", R.drawable.interstellar\_title, null);

movie3.addCinema(cinema1);

BookableMovie movie4 = new BookableMovie("RRR", R.drawable.rrr\_title, null);

movie4.addCinema(cinema2);

this.bookableMovies.add(movie1);

this.bookableMovies.add(movie2);

this.bookableMovies.add(movie3);

this.bookableMovies.add(movie4);

}

public List<Movie> getMovies()

{

return this.movies;

}

public void addMovie(Movie mov)

{

this.movies.add(mov);

}

public void removeMovie(String title)

{

for(int i = 0 ; i < this.movies.size() ; i++)

if(this.movies.get(i).getTitle().toLowerCase().equals(title.toLowerCase()))

this.movies.remove(i);

}

public List<BookableMovie> getBookableMovies()

{

return this.bookableMovies;

}

public void removeBookableMovie(String movieName)

{

for(int i = 0 ; i < this.bookableMovies.size() ; i++)

if(this.bookableMovies.get(i).getMovieName().toLowerCase().equals(movieName.toLowerCase()))

this.bookableMovies.remove(i);

}

public void addBookableMovie(BookableMovie mov)

{

this.bookableMovies.add(mov);

}

}

|  |  |  |
| --- | --- | --- |
| EnterDetails.java |  |  |

package com.app.spectraflix;

import android.os.Bundle;

import android.app.Activity;

import android.widget.Spinner;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.ImageView;

import android.widget.ImageButton;

import android.graphics.Typeface;

import android.widget.TextView;

import android.view.View;

import android.view.Gravity;

import android.util.DisplayMetrics;

import android.widget.EditText;

import android.widget.LinearLayout;

import android.widget.RelativeLayout;

import android.content.Intent;

import java.util.List;

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

public class EnterDetails extends Activity

{

private String selectedCinema;

private String selectedRuntime;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.enter\_details);

if(getActionBar() != null)

getActionBar().hide();

DisplayMetrics displayMetrics = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

int width = displayMetrics.widthPixels;

int height = displayMetrics.heightPixels;

Typeface customFont = Typeface.createFromAsset(getAssets(), "fonts/nasalization.ttf");

TextView detailsTextView = findViewById(R.id.details\_label);

detailsTextView.setTypeface(customFont);

RelativeLayout relativeLayout = findViewById(R.id.RelativeLayout);

ImageButton imageButton = new ImageButton(this);

LinearLayout.LayoutParams buttonParams = new LinearLayout.LayoutParams(width / 8, height / 12);

buttonParams.gravity = Gravity.CENTER;

imageButton.setLayoutParams(buttonParams);

imageButton.setImageResource(R.drawable.back);

imageButton.setScaleType(ImageView.ScaleType.FIT\_START);

imageButton.setBackground(null);

imageButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), activity2.class);

startActivity(intent);

}

});

relativeLayout.addView(imageButton);

Intent intent = getIntent();

BookableMovie movie = (BookableMovie) intent.getSerializableExtra("BookableMovie");

Spinner cinemaSpinner = findViewById(R.id.cinemaSpinner);

List<Cinema> cinemas = movie.getCinemas();

String[] cinemaNames = new String[cinemas.size()];

for(int i = 0 ; i < cinemas.size() ; i++)

cinemaNames[i] = cinemas.get(i).getName();

ArrayAdapter<String> cinemaAdapter = new ArrayAdapter<>(this, android.R.layout.simple\_spinner\_item, cinemaNames);

cinemaAdapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

cinemaSpinner.setAdapter(cinemaAdapter);

final List<Cinema> cinemas\_final = cinemas;

cinemaSpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()

{

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id)

{

selectedCinema = parent.getItemAtPosition(position).toString();

Cinema cinema = null;

for(Cinema cinema\_item : cinemas\_final)

if(cinema\_item.getName().equals(selectedCinema))

cinema = cinema\_item;

List<Runtime> runtimes = cinema.getRuntimes();

String[] runtimes\_string = new String[runtimes.size()];

for(int i = 0 ; i < runtimes.size() ; i++)

runtimes\_string[i] = runtimes.get(i).getStartDateTime().format(DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm"));

Spinner runtimeSpinner = findViewById(R.id.runtimeSpinner);

ArrayAdapter<String> runtimeAdapter = new ArrayAdapter<>(EnterDetails.this, android.R.layout.simple\_spinner\_item, runtimes\_string);

runtimeAdapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

runtimeSpinner.setAdapter(runtimeAdapter);

runtimeSpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()

{

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id)

{

selectedRuntime = parent.getItemAtPosition(position).toString();

}

@Override

public void onNothingSelected(AdapterView<?> parent)

{}

});

}

@Override

public void onNothingSelected(AdapterView<?> parent)

{}

});

LinearLayout linearLayout = findViewById(R.id.LinearLayout);

ImageButton nextButton = new ImageButton(this);

LinearLayout.LayoutParams nextButtonParams = new LinearLayout.LayoutParams(width / 6, height / 6);

nextButtonParams.gravity = Gravity.CENTER;

nextButton.setLayoutParams(nextButtonParams);

nextButton.setImageResource(R.drawable.next\_screen);

nextButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

nextButton.setBackground(null);

final BookableMovie mov\_final = movie;

nextButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

EditText editName = findViewById(R.id.editName);

String enteredName = editName.getText().toString();

EditText editPhoneNum = findViewById(R.id.editPhoneNum);

String enteredPhoneNum = editPhoneNum.getText().toString();

EditText editEmail = findViewById(R.id.editEmail);

String enteredEmail = editEmail.getText().toString();

final String[] data = {enteredName, enteredPhoneNum, enteredEmail, selectedCinema, selectedRuntime};

Intent intent = new Intent();

intent.putExtra("Data", data);

intent.putExtra("BookableMovie", mov\_final);

intent.setClass(getApplicationContext(), SelectSeat.class);

startActivity(intent);

}

});

linearLayout.addView(nextButton);

}

}

|  |  |  |
| --- | --- | --- |
| Movie.java |  |  |

package com.app.spectraflix;

import java.io.Serializable;

public class Movie implements Serializable

{

private String title, imagePath, titleImagePath, trailerId, movieEmbedCode, movieURL, localPath, description;

private int resImagePath, resTitleImagePath, category;

public Movie(String title, String imagePath, String titleImagePath, int resImagePath, int resTitleImagePath, String trailerId, String movieEmbedCode, String movieURL, String localPath, String description, int category)

{

this.title = title;

this.imagePath = imagePath;

this.titleImagePath = titleImagePath;

this.trailerId = trailerId;

this.resImagePath = resImagePath;

this.resTitleImagePath = resTitleImagePath;

this.description = description;

this.movieEmbedCode = movieEmbedCode;

this.movieURL = movieURL;

this.localPath = localPath;

this.category = category;

}

public String getTitle()

{

return this.title;

}

public int getCategory()

{

return this.category;

}

public String getImagePath()

{

return this.imagePath;

}

public String getTitleImagePath()

{

return this.titleImagePath;

}

public int getResImagePath()

{

return this.resImagePath;

}

public int getResTitleImagePath()

{

return this.resTitleImagePath;

}

public String getDescription()

{

return this.description;

}

public String getTrailerId()

{

return this.trailerId;

}

public String getMovieEmbedCode()

{

return this.movieEmbedCode;

}

public String getLocalPath()

{

return this.localPath;

}

public String getMovieURL()

{

return this.movieURL;

}

public void setTitle(String title)

{

this.title = title;

}

public void setImagePath(String imagePath)

{

this.imagePath = imagePath;

}

public void setResImagePath(int resImagePath)

{

this.resImagePath = resImagePath;

}

public void setResTitleImagePath(int resTitleImagePath)

{

this.resTitleImagePath = resTitleImagePath;

}

public void setDescription(String description)

{

this.description = description;

}

public void setTitleImagePath(String titleImagePath)

{

this.titleImagePath = titleImagePath;

}

public void setTrailerId(String trailerId)

{

this.trailerId = trailerId;

}

public void setMovieEmbedCode(String movieEmbedCode)

{

this.movieEmbedCode = movieEmbedCode;

}

public void setCategory(int category)

{

this.category = category;

}

public void setMovieURL(String movieURL)

{

this.movieURL = movieURL;

}

public void setLocalPath(String localPath)

{

this.localPath = localPath;

}

}

|  |  |  |
| --- | --- | --- |
| MovieLocalView.java |  |  |

package com.app.spectraflix;

import android.app.Activity;

import android.os.Bundle;

import android.widget.VideoView;

import android.widget.MediaController;

import android.content.Intent;

public class MovieLocalView extends Activity

{

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.movie\_url\_view);

if (getActionBar() != null)

getActionBar().hide();

Intent intent = getIntent();

String moviePath = intent.getStringExtra("MoviePath");

VideoView videoView = findViewById(R.id.videoView);

MediaController mediaController = new MediaController(this);

videoView.setMediaController(mediaController);

mediaController.setAnchorView(videoView);

videoView.setVideoPath(moviePath);

videoView.start();

}

}

|  |  |  |
| --- | --- | --- |
| MovieTrailerView.java |  |  |

package com.app.spectraflix;

import android.os.Bundle;

import android.webkit.WebSettings;

import android.webkit.WebView;

import android.webkit.WebViewClient;

import android.app.Activity;

import android.widget.RelativeLayout;

import android.widget.LinearLayout;

import android.content.Intent;

import android.util.DisplayMetrics;

import android.widget.ImageButton;

import android.view.Gravity;

import android.view.View;

import android.widget.ImageView;

public class MovieTrailerView extends Activity

{

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.movie\_webview);

if (getActionBar() != null)

getActionBar().hide();

DisplayMetrics dM = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(dM);

int width = dM.widthPixels;

int height = dM.heightPixels;

RelativeLayout relativeLayout = findViewById(R.id.RelativeLayout);

ImageButton imageButton = new ImageButton(this);

LinearLayout.LayoutParams buttonParams = new LinearLayout.LayoutParams(width / 8, height / 12);

buttonParams.gravity = Gravity.CENTER;

imageButton.setLayoutParams(buttonParams);

imageButton.setImageResource(R.drawable.back);

imageButton.setScaleType(ImageView.ScaleType.FIT\_START);

imageButton.setBackground(null);

imageButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), MainActivity.class);

startActivity(intent);

}

});

relativeLayout.addView(imageButton);

Intent intent = getIntent();

String movieTrailerId = intent.getStringExtra("MovieTrailerId");

String trailerEmbedCode = "<iframe id=\"player\" type=\"text/html\" width=\"100%\" height=\"100%\" src=\"https://www.youtube.com/embed/" + movieTrailerId + "?enablejsapi=1\" frameborder=\"0\"></iframe>";

WebView webView = findViewById(R.id.webView);

WebSettings webSettings = webView.getSettings();

webSettings.setJavaScriptEnabled(true);

webView.setWebViewClient(new WebViewClient());

webView.loadData(trailerEmbedCode, "text/html", "UTF-8");

}

}

|  |  |  |
| --- | --- | --- |
| MovieURLView.java |  |  |

package com.app.spectraflix;

import android.app.Activity;

import android.os.Bundle;

import android.net.Uri;

import android.widget.VideoView;

import android.content.Intent;

import android.widget.MediaController;

public class MovieURLView extends Activity

{

private VideoView videoView;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.movie\_url\_view);

if (getActionBar() != null)

getActionBar().hide();

Intent intent = getIntent();

String movieURL = intent.getStringExtra("MovieURL");

videoView = findViewById(R.id.videoView);

Uri uri = Uri.parse(movieURL);

videoView.setVideoURI(uri);

MediaController mediaController = new MediaController(this);

mediaController.setAnchorView(videoView);

videoView.setMediaController(mediaController);

videoView.start();

}

}

|  |  |  |
| --- | --- | --- |
| MovieWebView.java |  |  |

package com.app.spectraflix;

import android.os.Bundle;

import android.webkit.WebSettings;

import android.webkit.WebView;

import android.webkit.WebViewClient;

import android.app.Activity;

import android.widget.RelativeLayout;

import android.widget.LinearLayout;

import android.content.Intent;

import android.util.DisplayMetrics;

import android.widget.ImageButton;

import android.view.Gravity;

import android.view.View;

import android.widget.ImageView;

public class MovieWebView extends Activity

{

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.movie\_webview);

if (getActionBar() != null)

getActionBar().hide();

DisplayMetrics dM = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(dM);

int width = dM.widthPixels;

int height = dM.heightPixels;

RelativeLayout relativeLayout = findViewById(R.id.RelativeLayout);

ImageButton imageButton = new ImageButton(this);

LinearLayout.LayoutParams buttonParams = new LinearLayout.LayoutParams(width / 8, height / 12);

buttonParams.gravity = Gravity.CENTER;

imageButton.setLayoutParams(buttonParams);

imageButton.setImageResource(R.drawable.back);

imageButton.setScaleType(ImageView.ScaleType.FIT\_START);

imageButton.setBackground(null);

imageButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), MainActivity.class);

startActivity(intent);

}

});

relativeLayout.addView(imageButton);

Intent intent = getIntent();

String movieEmbedCode = intent.getStringExtra("MovieEmbedCode");

WebView webView = findViewById(R.id.webView);

WebSettings webSettings = webView.getSettings();

webSettings.setJavaScriptEnabled(true);

webView.setWebViewClient(new WebViewClient());

webView.loadData(movieEmbedCode, "text/html", "UTF-8");

}

}

|  |  |  |
| --- | --- | --- |
| PreviewActivity.java |  |  |

package com.app.spectraflix;

import android.os.Bundle;

import android.util.DisplayMetrics;

import android.widget.LinearLayout;

import android.widget.RelativeLayout;

import android.widget.ScrollView;

import android.view.View;

import android.widget.TextView;

import android.widget.ImageView;

import android.widget.ImageButton;

import android.net.Uri;

import java.io.File;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.view.Gravity;

import android.content.Intent;

import android.app.Activity;

import android.graphics.Typeface;

import android.graphics.drawable.Drawable;

import android.widget.Button;

import android.widget.Toast;

public class PreviewActivity extends Activity

{

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.preview);

if (getActionBar() != null)

getActionBar().hide();

Intent intent = getIntent();

Movie mov = (Movie) intent.getSerializableExtra("Movie");

Typeface customFont = Typeface.createFromAsset(getAssets(), "fonts/nasalization.ttf");

TextView movieTextView = findViewById(R.id.watch\_movie);

movieTextView.setTypeface(customFont);

RelativeLayout relativeLayout = findViewById(R.id.RelativeLayout);

LinearLayout linearLayout = findViewById(R.id.LinearLayout);

DisplayMetrics displayMetrics = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

int width = displayMetrics.widthPixels;

int height = displayMetrics.heightPixels;

ImageButton imageButton = new ImageButton(this);

LinearLayout.LayoutParams buttonParams = new LinearLayout.LayoutParams(width / 8, height / 12);

buttonParams.gravity = Gravity.CENTER;

imageButton.setLayoutParams(buttonParams);

imageButton.setImageResource(R.drawable.back);

imageButton.setScaleType(ImageView.ScaleType.FIT\_START);

imageButton.setBackground(null);

imageButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), MainActivity.class);

startActivity(intent);

}

});

relativeLayout.addView(imageButton);

RelativeLayout.LayoutParams layoutParams = new RelativeLayout.LayoutParams(

width / 2,

height / 2

);

layoutParams.addRule(RelativeLayout.ALIGN\_PARENT\_BOTTOM);

layoutParams.addRule(RelativeLayout.ALIGN\_PARENT\_START);

linearLayout.setLayoutParams(layoutParams);

ImageView movieTitleView = new ImageView(this);

LinearLayout.LayoutParams movieTitleParams = new LinearLayout.LayoutParams(width / 3, height / 4);

movieTitleParams.gravity = Gravity.CENTER;

movieTitleView.setLayoutParams(movieTitleParams);

if(mov.getResTitleImagePath() == -1)

{

try

{

File file = new File(mov.getTitleImagePath());

Uri imageUri = Uri.fromFile(file);

Bitmap bitmap = BitmapFactory.decodeStream(getContentResolver().openInputStream(imageUri));

movieTitleView.setImageBitmap(bitmap);

}

catch(Exception e)

{}

}

else

movieTitleView.setImageResource(mov.getResTitleImagePath());

movieTitleView.setScaleType(ImageView.ScaleType.FIT\_CENTER);

ImageView bgImage = findViewById(R.id.background\_image);

if(mov.getResImagePath() == -1)

{

try

{

File file = new File(mov.getImagePath());

Uri imageUri = Uri.fromFile(file);

Bitmap bitmap = BitmapFactory.decodeStream(getContentResolver().openInputStream(imageUri));

bgImage.setImageBitmap(bitmap);

}

catch(Exception e)

{}

}

else

bgImage.setImageResource(mov.getResImagePath());

ScrollView scrollView = new ScrollView(this);

TextView textView = new TextView(this);

textView.setText(mov.getDescription());

textView.setTypeface(customFont);

textView.setTextSize(18);

textView.setBackgroundResource(R.drawable.rounded\_textview);

scrollView.addView(textView);

linearLayout.addView(movieTitleView);

linearLayout.addView(scrollView);

LinearLayout buttonsLayout = findViewById(R.id.WatchMoviesLayout);

RelativeLayout.LayoutParams buttonsParams = new RelativeLayout.LayoutParams(

width / 4,

height / 2

);

buttonsParams.addRule(RelativeLayout.ALIGN\_PARENT\_BOTTOM);

buttonsParams.addRule(RelativeLayout.ALIGN\_PARENT\_END);

buttonsLayout.setLayoutParams(buttonsParams);

ImageButton trailerButton = new ImageButton(this);

RelativeLayout.LayoutParams trailerParams = new RelativeLayout.LayoutParams(

width / 6,

height / 6

);

trailerParams.addRule(RelativeLayout.ALIGN\_PARENT\_TOP);

trailerParams.addRule(RelativeLayout.ALIGN\_PARENT\_END);

trailerButton.setLayoutParams(trailerParams);

trailerButton.setImageResource(R.drawable.trailer);

trailerButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

trailerButton.setBackground(null);

final String trailerId = mov.getTrailerId();

trailerButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

if(trailerId.equals("") || trailerId == null)

Toast.makeText(getApplicationContext(), "TRAILER NOT AVAILABLE!", Toast.LENGTH\_SHORT).show();

else

{

Intent intent = new Intent();

intent.putExtra("MovieTrailerId", trailerId);

intent.setClass(getApplicationContext(), MovieTrailerView.class);

startActivity(intent);

}

}

});

relativeLayout.addView(trailerButton);

int[] buttonIds = {R.id.icon\_button1, R.id.icon\_button2, R.id.icon\_button3};

final String[] strIds = {"MovieEmbedCode", "MovieURL", "MoviePath"};

final String[] strData = {mov.getMovieEmbedCode(), mov.getMovieURL(), mov.getLocalPath()};

final Class<?>[] classes = new Class<?>[] {

MovieWebView.class,

MovieURLView.class,

MovieLocalView.class

};

for(int i = 0 ; i < 3 ; i++)

{

ImageButton button = findViewById(buttonIds[i]);

final String movieStr = strData[i];

final int k = i;

button.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

if(movieStr.equals("") || movieStr == null)

Toast.makeText(getApplicationContext(), "MOVIE NOT AVAILABLE!", Toast.LENGTH\_SHORT).show();

else

{

Intent intent = new Intent();

intent.putExtra(strIds[k], strData[k]);

intent.setClass(getApplicationContext(), classes[k]);

startActivity(intent);

}

}

});

}

}

}

|  |  |  |
| --- | --- | --- |
| Runtime.java |  |  |

package com.app.spectraflix;

import java.util.List;

import java.util.LinkedList;

import java.time.LocalDateTime;

import java.io.Serializable;

import java.time.format.DateTimeFormatter;

public class Runtime implements Serializable

{

private LocalDateTime startDateTime, endDateTime;

public int[][] seats;

public Runtime(String startDateTime, String endDateTime, int[] rowCol)

{

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm");

this.startDateTime = LocalDateTime.parse(startDateTime, formatter);

this.endDateTime = LocalDateTime.parse(endDateTime, formatter);

this.seats = new int[rowCol[0]][rowCol[1]];

for(int i = 0 ; i < rowCol[0] ; i++)

for(int j = 0 ; j < rowCol[1] ; j++)

this.seats[i][j] = 0;

}

public int[][] getSeats()

{

return this.seats;

}

public LocalDateTime getStartDateTime()

{

return this.startDateTime;

}

public LocalDateTime getEndDateTime()

{

return this.endDateTime;

}

public boolean bookSeat(int i, int j)

{

if(this.seats[i][j] == 1)

return false;

this.seats[i][j] = 1;

return true;

}

public boolean cancelSeat(int i, int j)

{

if(this.seats[i][j] == 0)

return false;

this.seats[i][j] = 0;

return true;

}

}

|  |  |  |
| --- | --- | --- |
| SelectSeat.java |  |  |

package com.app.spectraflix;

import android.app.Activity;

import android.os.Bundle;

import android.content.Intent;

import android.widget.GridLayout;

import android.view.Gravity;

import android.widget.ImageButton;

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

import android.widget.LinearLayout;

import android.widget.RelativeLayout;

import android.widget.Toast;

import java.util.List;

import android.view.View;

import android.util.DisplayMetrics;

import android.graphics.Typeface;

import android.widget.TextView;

import android.content.Context;

import android.widget.ImageView;

public class SelectSeat extends Activity

{

private ImageButton[][] buttonArray;

private BookableMovie movie;

private int movie\_index, cinema\_index, runtime\_index, selected\_row = -1, selected\_col = -1;

private List<BookableMovie> bookableMovies;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.select\_seat);

if(getActionBar() != null)

getActionBar().hide();

Typeface customFont = Typeface.createFromAsset(getAssets(), "fonts/nasalization.ttf");

TextView seatTextView = findViewById(R.id.seat\_label);

seatTextView.setTypeface(customFont);

DisplayMetrics displayMetrics = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

int width = displayMetrics.widthPixels;

int height = displayMetrics.heightPixels;

RelativeLayout relativeLayout = findViewById(R.id.RelativeLayout);

ImageButton backButton = new ImageButton(this);

LinearLayout.LayoutParams backButtonParams = new LinearLayout.LayoutParams(width / 8, height / 12);

backButtonParams.gravity = Gravity.CENTER;

backButton.setLayoutParams(backButtonParams);

backButton.setImageResource(R.drawable.back);

backButton.setScaleType(ImageView.ScaleType.FIT\_START);

backButton.setBackground(null);

backButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), activity2.class);

startActivity(intent);

}

});

relativeLayout.addView(backButton);

Intent intent = getIntent();

String[] data = intent.getStringArrayExtra("Data");

movie = (BookableMovie) intent.getSerializableExtra("BookableMovie");

final DataManager dataManager = (DataManager) getApplication();

bookableMovies = dataManager.getBookableMovies();

for(int i = 0 ; i < bookableMovies.size() ; i++)

if(bookableMovies.get(i).getMovieName().equals(movie.getMovieName()))

movie\_index = i;

List<Cinema> cinemas = movie.getCinemas();

for(int i = 0 ; i < cinemas.size() ; i++)

if(cinemas.get(i).getName().equals(data[3]))

cinema\_index = i;

List<Runtime> runtimes = movie.cinemas.get(cinema\_index).getRuntimes();

for(int i = 0 ; i < runtimes.size() ; i++)

if(runtimes.get(i).getStartDateTime().format(DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm")).equals(data[4]))

runtime\_index = i;

LinearLayout linearLayout = findViewById(R.id.linearlayout);

int rows = movie.cinemas.get(cinema\_index).runtimes.get(runtime\_index).seats.length;

int cols = movie.cinemas.get(cinema\_index).runtimes.get(runtime\_index).seats[0].length;

buttonArray = new ImageButton[rows][cols];

GridLayout gridLayout = new GridLayout(this);

gridLayout.setRowCount(rows);

gridLayout.setColumnCount(cols);

LinearLayout.LayoutParams gridParams = new LinearLayout.LayoutParams(

LinearLayout.LayoutParams.WRAP\_CONTENT,

LinearLayout.LayoutParams.WRAP\_CONTENT

);

gridLayout.setLayoutParams(gridParams);

gridLayout.setPadding(20, 20, 20, 20);

for(int row = 0 ; row < rows ; row++)

{

for(int col = 0 ; col < cols ; col++)

{

ImageButton imageButton = new ImageButton(this);

if(bookableMovies.get(movie\_index).cinemas.get(cinema\_index).runtimes.get(runtime\_index).seats[row][col] == 0)

imageButton.setImageResource(R.drawable.chair\_available);

else

imageButton.setImageResource(R.drawable.chair\_booked);

LinearLayout.LayoutParams buttonParams = new LinearLayout.LayoutParams(

width / cols / 2,

height / rows / 2

);

imageButton.setLayoutParams(buttonParams);

imageButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

imageButton.setBackground(null);

imageButton.setPadding(5, 5, 5, 5);

buttonArray[row][col] = imageButton;

final int currentRow = row;

final int currentCol = col;

imageButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

if(bookableMovies.get(movie\_index).cinemas.get(cinema\_index).runtimes.get(runtime\_index).seats[currentRow][currentCol] == 1)

Toast.makeText(SelectSeat.this, "Seat already booked!", Toast.LENGTH\_SHORT).show();

else

{

if(selected\_row != -1)

{

buttonArray[selected\_row][selected\_col].setImageResource(R.drawable.chair\_available);

bookableMovies.get(movie\_index).cinemas.get(cinema\_index).runtimes.get(runtime\_index).seats[selected\_row][selected\_col] = 0;

}

buttonArray[currentRow][currentCol].setImageResource(R.drawable.chair\_booked);

selected\_row = currentRow;

selected\_col = currentCol;

bookableMovies.get(movie\_index).cinemas.get(cinema\_index).runtimes.get(runtime\_index).seats[currentRow][currentCol] = 1;

}

}

});

gridLayout.addView(imageButton);

}

}

linearLayout.addView(gridLayout);

ImageButton bookButton = new ImageButton(this);

LinearLayout.LayoutParams bookButtonParams = new LinearLayout.LayoutParams(width / 4, height / 6);

bookButtonParams.gravity = Gravity.CENTER;

bookButton.setLayoutParams(bookButtonParams);

bookButton.setImageResource(R.drawable.book\_ticket);

bookButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

bookButton.setBackground(null);

final String[] data\_new = data;

bookButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

String[] new\_data = {data\_new[0], data\_new[1], data\_new[2], data\_new[3], data\_new[4], Character.toString((char)(65 + selected\_row)) + selected\_col};

Intent intent = new Intent();

intent.putExtra("Data", new\_data);

intent.setClass(getApplicationContext(), Ticket.class);

startActivity(intent);

}

});

linearLayout.addView(bookButton);

}

}

|  |  |  |
| --- | --- | --- |
| Ticket.java |  |  |

package com.app.spectraflix;

import android.app.Activity;

import android.os.Bundle;

import android.content.Intent;

import android.widget.TextView;

public class Ticket extends Activity

{

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.ticket);

if (getActionBar() != null)

getActionBar().hide();

Intent intent = getIntent();

String[] data = intent.getStringArrayExtra("Data");

String text = "Name : " + data[0] + "\nPhone No. : " + data[1] + "\nEmail Address : " + data[2] + "\nCinema : " + data[3] + "\nRuntime : " + data[4] + "\nSeat : " + data[5] + "\n\nPLEASE TAKE A SCREENSHOT OF THIS PAGE";

TextView textView = findViewById(R.id.ticket\_details);

textView.setText(text);

}

}

|  |  |  |
| --- | --- | --- |
| activity2.java |  |  |

package com.app.spectraflix;

import android.app.Activity;

import android.os.Bundle;

import android.content.Intent;

import android.view.View;

import android.widget.ImageButton;

import android.widget.ImageView;

import android.graphics.drawable.ColorDrawable;

import android.widget.LinearLayout;

import android.view.Gravity;

import android.util.DisplayMetrics;

import android.graphics.Color;

import android.app.ActionBar;

import java.util.List;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.net.Uri;

import java.io.File;

public class activity2 extends Activity

{

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity2);

ActionBar actionBar = getActionBar();

if(actionBar != null)

actionBar.setBackgroundDrawable(new ColorDrawable(Color.parseColor("#ACD5C2")));

ImageButton button1 = findViewById(R.id.icon\_button1);

button1.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), MainActivity.class);

startActivity(intent);

overridePendingTransition(0, 0);

}

});

ImageButton button3 = findViewById(R.id.icon\_button3);

button3.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), activity3.class);

startActivity(intent);

overridePendingTransition(0, 0);

}

});

DataManager dataManager = (DataManager) getApplication();

List<BookableMovie> bookableMovies = dataManager.getBookableMovies();

DisplayMetrics displayMetrics = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

int height = displayMetrics.heightPixels;

for(BookableMovie movie : bookableMovies)

addBookableMovie(movie, height);

}

public void addBookableMovie(BookableMovie movie, int height)

{

LinearLayout parentLayout = findViewById(R.id.ImageLinearLayout);

ImageButton imageButton = new ImageButton(this);

LinearLayout.LayoutParams buttonParams = new LinearLayout.LayoutParams(

LinearLayout.LayoutParams.MATCH\_PARENT,

height / 3

);

buttonParams.gravity = Gravity.CENTER;

imageButton.setLayoutParams(buttonParams);

if(movie.getResMovieTitlePath() == -1)

{

try

{

File file = new File(movie.getMovieTitlePath());

Uri imageUri = Uri.fromFile(file);

Bitmap bitmap = BitmapFactory.decodeStream(getContentResolver().openInputStream(imageUri));

imageButton.setImageBitmap(bitmap);

}

catch(Exception e)

{}

}

else

imageButton.setImageResource(movie.getResMovieTitlePath());

imageButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

imageButton.setBackground(null);

final BookableMovie mov\_final = movie;

imageButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.putExtra("BookableMovie", mov\_final);

intent.setClass(getApplicationContext(), EnterDetails.class);

startActivity(intent);

}

});

parentLayout.addView(imageButton);

}

}

|  |  |  |
| --- | --- | --- |
| activity3.java |  |  |

package com.app.spectraflix;

import android.app.Activity;

import android.os.Bundle;

import android.content.Intent;

import android.view.View;

import android.widget.ImageButton;

import android.graphics.drawable.ColorDrawable;

import android.widget.LinearLayout;

import android.widget.ImageView;

import android.util.DisplayMetrics;

import android.graphics.Color;

import android.view.Gravity;

import android.app.ActionBar;

import android.graphics.Typeface;

import android.widget.TextView;

public class activity3 extends Activity

{

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity3);

ActionBar actionBar = getActionBar();

if(actionBar != null)

actionBar.setBackgroundDrawable(new ColorDrawable(Color.parseColor("#ACD5C2")));

Typeface customFont = Typeface.createFromAsset(getAssets(), "fonts/nasalization.ttf");

TextView movieView = findViewById(R.id.add\_remove\_movie);

movieView.setTypeface(customFont);

TextView bookableMovieView = findViewById(R.id.add\_remove\_bookable\_movie);

bookableMovieView.setTypeface(customFont);

ImageButton button1 = findViewById(R.id.icon\_button1);

button1.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), MainActivity.class);

startActivity(intent);

overridePendingTransition(0, 0);

}

});

ImageButton button2 = findViewById(R.id.icon\_button2);

button2.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), activity2.class);

startActivity(intent);

overridePendingTransition(0, 0);

}

});

LinearLayout linearLayout = findViewById(R.id.add\_remove\_movie\_layout);

DisplayMetrics displayMetrics = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

int width = displayMetrics.widthPixels;

int height = displayMetrics.heightPixels;

ImageButton addMovieButton = new ImageButton(this);

LinearLayout.LayoutParams addMovieParams = new LinearLayout.LayoutParams(width / 3, height / 3);

addMovieParams.gravity = Gravity.CENTER;

addMovieButton.setLayoutParams(addMovieParams);

addMovieButton.setImageResource(R.drawable.add);

addMovieButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

addMovieButton.setBackground(null);

addMovieButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), addMovie.class);

startActivity(intent);

}

});

ImageButton removeMovieButton = new ImageButton(this);

LinearLayout.LayoutParams removeMovieParams = new LinearLayout.LayoutParams(width / 3, height / 3);

removeMovieParams.gravity = Gravity.CENTER;

removeMovieButton.setLayoutParams(removeMovieParams);

removeMovieButton.setImageResource(R.drawable.remove);

removeMovieButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

removeMovieButton.setBackground(null);

removeMovieButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), removeMovie.class);

startActivity(intent);

}

});

linearLayout.addView(addMovieButton);

linearLayout.addView(removeMovieButton);

LinearLayout linearLayout2 = findViewById(R.id.add\_remove\_bookable\_movie\_layout);

ImageButton addBookableMovieButton = new ImageButton(this);

LinearLayout.LayoutParams addBookableMovieParams = new LinearLayout.LayoutParams(width / 3, height / 3);

addBookableMovieParams.gravity = Gravity.CENTER;

addBookableMovieButton.setLayoutParams(addBookableMovieParams);

addBookableMovieButton.setImageResource(R.drawable.add);

addBookableMovieButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

addBookableMovieButton.setBackground(null);

addBookableMovieButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), addBookableMovie.class);

startActivity(intent);

}

});

ImageButton removeBookableMovieButton = new ImageButton(this);

LinearLayout.LayoutParams removeBookableMovieParams = new LinearLayout.LayoutParams(width / 3, height / 3);

removeBookableMovieParams.gravity = Gravity.CENTER;

removeBookableMovieButton.setLayoutParams(removeBookableMovieParams);

removeBookableMovieButton.setImageResource(R.drawable.remove);

removeBookableMovieButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

removeBookableMovieButton.setBackground(null);

removeBookableMovieButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), removeBookableMovie.class);

startActivity(intent);

}

});

linearLayout2.addView(addBookableMovieButton);

linearLayout2.addView(removeBookableMovieButton);

}

}

|  |  |  |
| --- | --- | --- |
| addBookableMovie.java |  |  |

package com.app.spectraflix;

import android.os.Bundle;

import android.app.Activity;

import android.widget.ImageView;

import android.widget.ImageButton;

import android.graphics.Typeface;

import android.widget.TextView;

import android.view.View;

import android.view.Gravity;

import android.util.DisplayMetrics;

import android.widget.EditText;

import android.widget.Toast;

import android.widget.LinearLayout;

import android.widget.RelativeLayout;

import android.content.Intent;

import java.util.List;

public class addBookableMovie extends Activity

{

private BookableMovie mov = null;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.add\_bookable\_movie);

if(getActionBar() != null)

getActionBar().hide();

DisplayMetrics displayMetrics = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

int width = displayMetrics.widthPixels;

int height = displayMetrics.heightPixels;

Typeface customFont = Typeface.createFromAsset(getAssets(), "fonts/nasalization.ttf");

TextView addTextView1 = findViewById(R.id.add\_bookable\_movie);

addTextView1.setTypeface(customFont);

TextView addTextView2 = findViewById(R.id.add\_cinema);

addTextView2.setTypeface(customFont);

TextView addTextView3 = findViewById(R.id.add\_runtime);

addTextView3.setTypeface(customFont);

RelativeLayout relativeLayout = findViewById(R.id.RelativeLayout);

ImageButton imageButton = new ImageButton(this);

LinearLayout.LayoutParams buttonParams = new LinearLayout.LayoutParams(width / 8, height / 12);

buttonParams.gravity = Gravity.CENTER;

imageButton.setLayoutParams(buttonParams);

imageButton.setImageResource(R.drawable.back);

imageButton.setScaleType(ImageView.ScaleType.FIT\_START);

imageButton.setBackground(null);

imageButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), activity3.class);

startActivity(intent);

}

});

relativeLayout.addView(imageButton);

LinearLayout linearLayout1 = findViewById(R.id.InnerLinearLayout1);

ImageButton addButton1 = new ImageButton(this);

LinearLayout.LayoutParams addButtonParams1 = new LinearLayout.LayoutParams(width / 6, height / 6);

addButtonParams1.gravity = Gravity.CENTER;

addButton1.setLayoutParams(addButtonParams1);

addButton1.setImageResource(R.drawable.add);

addButton1.setScaleType(ImageView.ScaleType.FIT\_CENTER);

addButton1.setBackground(null);

addButton1.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

EditText editName = findViewById(R.id.editMovieName);

String name = editName.getText().toString();

EditText editTitlePath = findViewById(R.id.editMovieTitlePath);

String path = editTitlePath.getText().toString();

if(name.equals("") || path.equals(""))

Toast.makeText(getApplicationContext(), "PLEASE ENTER THE DETAILS", Toast.LENGTH\_SHORT).show();

else

{

mov = new BookableMovie(name, -1, path);

Toast.makeText(getApplicationContext(), "SUCCESSFULLY CREATED!", Toast.LENGTH\_SHORT).show();

}

}

});

linearLayout1.addView(addButton1);

LinearLayout linearLayout2 = findViewById(R.id.InnerLinearLayout2);

ImageButton addButton2 = new ImageButton(this);

LinearLayout.LayoutParams addButtonParams2 = new LinearLayout.LayoutParams(width / 6, height / 6);

addButtonParams2.gravity = Gravity.CENTER;

addButton2.setLayoutParams(addButtonParams2);

addButton2.setImageResource(R.drawable.add);

addButton2.setScaleType(ImageView.ScaleType.FIT\_CENTER);

addButton2.setBackground(null);

addButton2.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

EditText editName = findViewById(R.id.editName);

String name = editName.getText().toString();

EditText editNumSeats = findViewById(R.id.editNumSeats);

String numSeats = editNumSeats.getText().toString();

if(name.equals("") || numSeats.equals(""))

Toast.makeText(getApplicationContext(), "PLEASE ENTER THE DETAILS", Toast.LENGTH\_SHORT).show();

else if(mov == null)

Toast.makeText(getApplicationContext(), "PLEASE CREATE A BOOKABLE MOVIE!", Toast.LENGTH\_SHORT).show();

else

{

Cinema cinema = new Cinema(name, Integer.parseInt(numSeats));

mov.addCinema(cinema);

Toast.makeText(getApplicationContext(), "SUCCESSFULLY CREATED!", Toast.LENGTH\_SHORT).show();

}

}

});

linearLayout2.addView(addButton2);

LinearLayout linearLayout3 = findViewById(R.id.InnerLinearLayout3);

ImageButton addButton3 = new ImageButton(this);

LinearLayout.LayoutParams addButtonParams3 = new LinearLayout.LayoutParams(width / 6, height / 6);

addButtonParams3.gravity = Gravity.CENTER;

addButton3.setLayoutParams(addButtonParams3);

addButton3.setImageResource(R.drawable.add);

addButton3.setScaleType(ImageView.ScaleType.FIT\_CENTER);

addButton3.setBackground(null);

addButton3.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

EditText editStart = findViewById(R.id.editStartDateTime);

String start = editStart.getText().toString();

EditText editEnd = findViewById(R.id.editEndDateTime);

String end = editEnd.getText().toString();

if(start.equals("") || end.equals(""))

Toast.makeText(getApplicationContext(), "PLEASE ENTER THE DETAILS", Toast.LENGTH\_SHORT).show();

else if(mov == null)

Toast.makeText(getApplicationContext(), "PLEASE CREATE A BOOKABLE MOVIE!", Toast.LENGTH\_SHORT).show();

else if(mov.getCinemas().size() == 0)

Toast.makeText(getApplicationContext(), "PLEASE ADD A CINEMA!", Toast.LENGTH\_SHORT).show();

else

{

mov.cinemas.get(mov.getCinemas().size() - 1).addRuntime(start, end);

Toast.makeText(getApplicationContext(), "SUCCESSFULLY CREATED!", Toast.LENGTH\_SHORT).show();

}

}

});

linearLayout3.addView(addButton3);

LinearLayout linearLayout = findViewById(R.id.LinearLayout);

ImageButton addButton = new ImageButton(this);

LinearLayout.LayoutParams addButtonParams = new LinearLayout.LayoutParams(width / 4, height / 4);

addButtonParams.gravity = Gravity.CENTER;

addButton.setLayoutParams(addButtonParams);

addButton.setImageResource(R.drawable.next\_screen);

addButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

addButton.setBackground(null);

addButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

if(mov == null)

Toast.makeText(getApplicationContext(), "PLEASE CREATE A BOOKABLE MOVIE!", Toast.LENGTH\_SHORT).show();

else

{

DataManager dataManager = (DataManager) getApplication();

dataManager.addBookableMovie(mov);

Toast.makeText(getApplicationContext(), "SUCCESSFULLY ADDED!", Toast.LENGTH\_SHORT).show();

}

}

});

linearLayout.addView(addButton);

}

}

|  |  |  |
| --- | --- | --- |
| addMovie.java |  |  |

package com.app.spectraflix;

import android.os.Bundle;

import android.app.Activity;

import android.widget.Spinner;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.ImageView;

import android.widget.ImageButton;

import android.graphics.Typeface;

import android.widget.TextView;

import android.view.View;

import android.view.Gravity;

import android.util.DisplayMetrics;

import android.widget.EditText;

import android.widget.Toast;

import android.widget.LinearLayout;

import android.widget.RelativeLayout;

import android.content.Intent;

import java.util.List;

public class addMovie extends Activity

{

private String selectedCategory;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.add\_movie);

if(getActionBar() != null)

getActionBar().hide();

DisplayMetrics displayMetrics = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

int width = displayMetrics.widthPixels;

int height = displayMetrics.heightPixels;

Typeface customFont = Typeface.createFromAsset(getAssets(), "fonts/nasalization.ttf");

TextView addTextView = findViewById(R.id.add\_movie);

addTextView.setTypeface(customFont);

RelativeLayout relativeLayout = findViewById(R.id.RelativeLayout);

ImageButton imageButton = new ImageButton(this);

LinearLayout.LayoutParams buttonParams = new LinearLayout.LayoutParams(width / 8, height / 12);

buttonParams.gravity = Gravity.CENTER;

imageButton.setLayoutParams(buttonParams);

imageButton.setImageResource(R.drawable.back);

imageButton.setScaleType(ImageView.ScaleType.FIT\_START);

imageButton.setBackground(null);

imageButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), activity3.class);

startActivity(intent);

}

});

relativeLayout.addView(imageButton);

Spinner categorySpinner = findViewById(R.id.movieCategory);

String[] categories = {"Animated Movie", "Popular Movie", "Horror Movie", "Developer's Choice"};

ArrayAdapter<String> categoriesAdapter = new ArrayAdapter<>(this, android.R.layout.simple\_spinner\_item, categories);

categoriesAdapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

categorySpinner.setAdapter(categoriesAdapter);

categorySpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()

{

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id)

{

selectedCategory = parent.getItemAtPosition(position).toString();

}

@Override

public void onNothingSelected(AdapterView<?> parent)

{}

});

LinearLayout linearLayout = findViewById(R.id.LinearLayout);

ImageButton addButton = new ImageButton(this);

LinearLayout.LayoutParams addButtonParams = new LinearLayout.LayoutParams(width / 6, height / 6);

addButtonParams.gravity = Gravity.CENTER;

addButton.setLayoutParams(addButtonParams);

addButton.setImageResource(R.drawable.add);

addButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

addButton.setBackground(null);

addButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

EditText editTitle = findViewById(R.id.editTitle);

String title = editTitle.getText().toString();

EditText editImagePath = findViewById(R.id.editImagePath);

String imagePath = editImagePath.getText().toString();

EditText editTitleImagePath = findViewById(R.id.editTitleImagePath);

String titleImagePath = editTitleImagePath.getText().toString();

EditText editTrailerId = findViewById(R.id.editTrailerId);

String trailerId = editTrailerId.getText().toString();

EditText editCode = findViewById(R.id.editMovieEmbedCode);

String code = editCode.getText().toString();

EditText editURL = findViewById(R.id.editMovieURL);

String url = editURL.getText().toString();

EditText editLocalPath = findViewById(R.id.editMovieLocalPath);

String localPath = editLocalPath.getText().toString();

EditText editDescription = findViewById(R.id.editMovieDescription);

String description = editDescription.getText().toString();

int category = -1;

if(selectedCategory.equals("Animated Movie"))

category = 1;

else if(selectedCategory.equals("Popular Movie"))

category = 2;

else if(selectedCategory.equals("Horror Movie"))

category = 3;

else

category = 4;

if(title.equals("") || imagePath.equals("") || titleImagePath.equals(""))

Toast.makeText(getApplicationContext(), "PLEASE ENTER THE DETAILS", Toast.LENGTH\_SHORT).show();

else

{

Movie newMovie = new Movie(title, imagePath, titleImagePath, -1, -1, trailerId, code, url, localPath, description, category);

DataManager dataManager = (DataManager) getApplication();

dataManager.addMovie(newMovie);

Toast.makeText(getApplicationContext(), "SUCCESSFULLY ADDED!", Toast.LENGTH\_SHORT).show();

}

}

});

linearLayout.addView(addButton);

}

}

|  |  |  |
| --- | --- | --- |
| removeBookableMovie.java |  |  |

package com.app.spectraflix;

import android.os.Bundle;

import android.app.Activity;

import android.widget.ImageView;

import android.widget.ImageButton;

import android.graphics.Typeface;

import android.widget.TextView;

import android.view.View;

import android.view.Gravity;

import android.util.DisplayMetrics;

import android.widget.EditText;

import android.widget.Toast;

import android.widget.LinearLayout;

import android.widget.RelativeLayout;

import android.content.Intent;

import java.util.List;

public class removeBookableMovie extends Activity

{

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.remove\_bookable\_movie);

if(getActionBar() != null)

getActionBar().hide();

DisplayMetrics displayMetrics = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

int width = displayMetrics.widthPixels;

int height = displayMetrics.heightPixels;

Typeface customFont = Typeface.createFromAsset(getAssets(), "fonts/nasalization.ttf");

TextView removeTextView = findViewById(R.id.remove\_bookable\_movie);

removeTextView.setTypeface(customFont);

RelativeLayout relativeLayout = findViewById(R.id.RelativeLayout);

ImageButton imageButton = new ImageButton(this);

LinearLayout.LayoutParams buttonParams = new LinearLayout.LayoutParams(width / 8, height / 12);

buttonParams.gravity = Gravity.CENTER;

imageButton.setLayoutParams(buttonParams);

imageButton.setImageResource(R.drawable.back);

imageButton.setScaleType(ImageView.ScaleType.FIT\_START);

imageButton.setBackground(null);

imageButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), activity3.class);

startActivity(intent);

}

});

relativeLayout.addView(imageButton);

LinearLayout linearLayout = findViewById(R.id.LinearLayout);

ImageButton removeButton = new ImageButton(this);

LinearLayout.LayoutParams removeButtonParams = new LinearLayout.LayoutParams(width / 6, height / 6);

removeButtonParams.gravity = Gravity.CENTER;

removeButton.setLayoutParams(removeButtonParams);

removeButton.setImageResource(R.drawable.remove);

removeButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

removeButton.setBackground(null);

removeButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

EditText editTitle = findViewById(R.id.editTitle);

String title = editTitle.getText().toString();

if(title.equals(""))

Toast.makeText(getApplicationContext(), "PLEASE ENTER THE TITLE!", Toast.LENGTH\_SHORT).show();

else

{

DataManager dataManager = (DataManager) getApplication();

dataManager.removeBookableMovie(title);

Toast.makeText(getApplicationContext(), "SUCCESSFULLY REMOVED!", Toast.LENGTH\_SHORT).show();

}

}

});

linearLayout.addView(removeButton);

}

}

|  |  |  |
| --- | --- | --- |
| removeMovie.java |  |  |

package com.app.spectraflix;

import android.os.Bundle;

import android.app.Activity;

import android.widget.ImageView;

import android.widget.ImageButton;

import android.graphics.Typeface;

import android.widget.TextView;

import android.view.View;

import android.view.Gravity;

import android.util.DisplayMetrics;

import android.widget.EditText;

import android.widget.Toast;

import android.widget.LinearLayout;

import android.widget.RelativeLayout;

import android.content.Intent;

import java.util.List;

public class removeMovie extends Activity

{

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.remove\_movie);

if(getActionBar() != null)

getActionBar().hide();

DisplayMetrics displayMetrics = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

int width = displayMetrics.widthPixels;

int height = displayMetrics.heightPixels;

Typeface customFont = Typeface.createFromAsset(getAssets(), "fonts/nasalization.ttf");

TextView removeTextView = findViewById(R.id.remove\_movie);

removeTextView.setTypeface(customFont);

RelativeLayout relativeLayout = findViewById(R.id.RelativeLayout);

ImageButton imageButton = new ImageButton(this);

LinearLayout.LayoutParams buttonParams = new LinearLayout.LayoutParams(width / 8, height / 12);

buttonParams.gravity = Gravity.CENTER;

imageButton.setLayoutParams(buttonParams);

imageButton.setImageResource(R.drawable.back);

imageButton.setScaleType(ImageView.ScaleType.FIT\_START);

imageButton.setBackground(null);

imageButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent();

intent.setClass(getApplicationContext(), activity3.class);

startActivity(intent);

}

});

relativeLayout.addView(imageButton);

LinearLayout linearLayout = findViewById(R.id.LinearLayout);

ImageButton removeButton = new ImageButton(this);

LinearLayout.LayoutParams removeButtonParams = new LinearLayout.LayoutParams(width / 6, height / 6);

removeButtonParams.gravity = Gravity.CENTER;

removeButton.setLayoutParams(removeButtonParams);

removeButton.setImageResource(R.drawable.remove);

removeButton.setScaleType(ImageView.ScaleType.FIT\_CENTER);

removeButton.setBackground(null);

removeButton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

EditText editTitle = findViewById(R.id.editTitle);

String title = editTitle.getText().toString();

if(title.equals(""))

Toast.makeText(getApplicationContext(), "PLEASE ENTER THE TITLE!", Toast.LENGTH\_SHORT).show();

else

{

DataManager dataManager = (DataManager) getApplication();

dataManager.removeMovie(title);

Toast.makeText(getApplicationContext(), "SUCCESSFULLY REMOVED!", Toast.LENGTH\_SHORT).show();

}

}

});

linearLayout.addView(removeButton);

}

}

|  |  |  |
| --- | --- | --- |
| streamingMovies.java |  |  |

package com.app.spectraflix;

import java.util.List;

import java.util.LinkedList;

public class streamingMovies

{

private List<Movie> animatedMovies = new LinkedList<Movie>();

private List<Movie> horrorMovies = new LinkedList<Movie>();

private List<Movie> popularMovies = new LinkedList<Movie>();

private List<Movie> developersChoiceMovies = new LinkedList<Movie>();

private List<Movie> documentaries = new LinkedList<Movie>();

public List<Movie> getHorrorMovies()

{

return this.horrorMovies;

}

public List<Movie> getAnimatedMovies()

{

return this.animatedMovies;

}

public List<Movie> getPopularMovies()

{

return this.popularMovies;

}

public List<Movie> getDevelopersChoiceMovies()

{

return this.developersChoiceMovies;

}

public List<Movie> getDocumentaries()

{

return this.documentaries;

}

public void addHorrorMovie(Movie movie)

{

this.horrorMovies.add(movie);

}

public void addAnimatedMovie(Movie movie)

{

this.animatedMovies.add(movie);

}

public void addPopularMovie(Movie movie)

{

this.popularMovies.add(movie);

}

public void addDevelopersChoiceMovie(Movie movie)

{

this.developersChoiceMovies.add(movie);

}

public void addDocumentary(Movie movie)

{

this.documentaries.add(movie);

}

}



|  |  |  |
| --- | --- | --- |
| AndroidManifest.xml |  |  |

<?xml version = "1.0" encoding = "utf-8"?>

<manifest

xmlns:android = "http://schemas.android.com/apk/res/android"

package = "com.app.spectraflix"

>

<uses-permission

android:name = "android.permission.INTERNET"

/>

<uses-permission

android:name = "android.permission.READ\_EXTERNAL\_STORAGE"

/>

<application

android:name = ".DataManager"

android:requestLegacyExternalStorage = "true"

android:allowBackup = "true"

android:icon = "@drawable/app\_icon"

android:label = "@string/app\_name"

android:theme = "@style/AppTheme"

android:resizeableActivity = "true"

>

<activity

android:name = ".MainActivity"

android:label = "@string/app\_name"

>

<intent-filter>

<action android:name = "android.intent.action.MAIN" />

<category android:name = "android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity android:name = ".PreviewActivity" />

<activity android:name = ".MovieWebView" />

<activity android:name = ".MovieURLView" />

<activity android:name = ".MovieLocalView" />

<activity android:name = ".SelectSeat" />

<activity android:name = ".Ticket" />

<activity android:name = ".addMovie" />

<activity android:name = ".removeMovie" />

<activity android:name = ".addBookableMovie" />

<activity android:name = ".removeBookableMovie" />

<activity android:name = ".MovieTrailerView" />

<activity android:name = ".activity2" />

<activity android:name = ".EnterDetails" />

<activity android:name = ".activity3" />

</application>

</manifest>

|  |  |  |
| --- | --- | --- |
| main.xml |  |  |

<?xml version="1.0" encoding="utf-8"?>

<GridLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:layout\_width = "fill\_parent"

android:layout\_height = "fill\_parent"

android:rowCount = "2"

android:columnCount = "1"

android:orientation = "horizontal"

android:layout\_gravity = "fill"

>

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:layout\_width = "match\_parent"

android:layout\_height = "0dp"

android:background = "@drawable/bg"

android:orientation = "vertical"

android:gravity = "center"

android:layout\_rowWeight = "9"

>

<ScrollView

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

>

<LinearLayout

android:layout\_width = "wrap\_content"

android:layout\_height = "wrap\_content"

android:gravity = "center"

android:orientation = "vertical"

>

<TextView

android:id = "@+id/category\_label\_1"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Animated Movies"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "start"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<HorizontalScrollView

android:id = "@+id/horizontalScrollView"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:layout\_marginStart = "1dp"

android:layout\_marginTop = "1dp"

android:layout\_marginEnd = "1dp"

android:layout\_marginBottom = "1dp"

android:foregroundGravity = "center\_vertical"

>

<LinearLayout

android:id = "@+id/animated\_movies\_scroll\_view"

android:layout\_width = "wrap\_content"

android:layout\_height = "match\_parent"

android:foregroundGravity = "center"

android:gravity = "center"

android:orientation = "horizontal"

/>

</HorizontalScrollView>

<TextView

android:id = "@+id/category\_label\_2"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Popular Movies"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "start"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<HorizontalScrollView

android:id = "@+id/horizontalScrollView"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:layout\_marginStart = "1dp"

android:layout\_marginTop = "1dp"

android:layout\_marginEnd = "1dp"

android:layout\_marginBottom = "1dp"

android:foregroundGravity = "center\_vertical"

>

<LinearLayout

android:id = "@+id/popular\_movies\_scroll\_view"

android:layout\_width = "wrap\_content"

android:layout\_height = "match\_parent"

android:foregroundGravity = "center"

android:gravity = "center"

android:orientation = "horizontal"

/>

</HorizontalScrollView>

<TextView

android:id = "@+id/category\_label\_3"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Horror Movies"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "start"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<HorizontalScrollView

android:id = "@+id/horizontalScrollView"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:layout\_marginStart = "1dp"

android:layout\_marginTop = "1dp"

android:layout\_marginEnd = "1dp"

android:layout\_marginBottom = "1dp"

android:foregroundGravity = "center\_vertical"

>

<LinearLayout

android:id = "@+id/horror\_movies\_scroll\_view"

android:layout\_width = "wrap\_content"

android:layout\_height = "match\_parent"

android:foregroundGravity = "center"

android:gravity = "center"

android:orientation = "horizontal"

/>

</HorizontalScrollView>

<TextView

android:id = "@+id/category\_label\_4"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Developer's Choice"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "start"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<HorizontalScrollView

android:id = "@+id/horizontalScrollView"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:layout\_marginStart = "1dp"

android:layout\_marginTop = "1dp"

android:layout\_marginEnd = "1dp"

android:layout\_marginBottom = "1dp"

android:foregroundGravity = "center\_vertical"

>

<LinearLayout

android:id = "@+id/developers\_choice\_scroll\_view"

android:layout\_width = "wrap\_content"

android:layout\_height = "match\_parent"

android:foregroundGravity = "center"

android:gravity = "center"

android:orientation = "horizontal"

/>

</HorizontalScrollView>

</LinearLayout>

</ScrollView>

</LinearLayout>

<GridLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:layout\_height = "0dp"

android:layout\_width = "match\_parent"

android:layout\_rowWeight = "1"

android:background = "#ACD5C2"

android:rowCount = "1"

android:columnCount = "3"

android:orientation = "horizontal"

android:layout\_gravity = "fill"

>

<ImageButton

android:id = "@+id/icon\_button1"

android:layout\_height = "match\_parent"

android:layout\_width = "0dp"

android:layout\_columnWeight = "1"

android:src = "@drawable/icon1"

android:background = "?android:attr/selectableItemBackgroundBorderless"

android:contentDescription = "Icon button"

android:scaleType = "centerInside"

android:padding = "8dp"

/>

<ImageButton

android:id = "@+id/icon\_button2"

android:layout\_height = "match\_parent"

android:layout\_width = "0dp"

android:layout\_columnWeight = "1"

android:src = "@drawable/icon2"

android:background = "?android:attr/selectableItemBackgroundBorderless"

android:contentDescription = "Icon button"

android:scaleType = "centerInside"

android:padding = "8dp"

/>

<ImageButton

android:id = "@+id/icon\_button3"

android:layout\_height = "match\_parent"

android:layout\_width = "0dp"

android:layout\_columnWeight = "1"

android:src = "@drawable/icon3"

android:background = "?android:attr/selectableItemBackgroundBorderless"

android:contentDescription = "Icon button"

android:scaleType = "centerInside"

android:padding = "8dp"

/>

</GridLayout>

</GridLayout>

|  |  |  |
| --- | --- | --- |
| activity2.xml |  |  |

<?xml version="1.0" encoding="utf-8"?>

<GridLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:layout\_width = "fill\_parent"

android:layout\_height = "fill\_parent"

android:rowCount = "2"

android:columnCount = "1"

android:orientation = "horizontal"

android:layout\_gravity = "fill"

>

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:layout\_width = "match\_parent"

android:layout\_height = "0dp"

android:background = "@drawable/bg"

android:orientation = "vertical"

android:gravity = "center"

android:layout\_rowWeight = "9"

>

<ScrollView

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

>

<LinearLayout

android:id = "@+id/ImageLinearLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:gravity = "center"

android:orientation = "vertical"

android:padding = "16dp"

/>

</ScrollView>

</LinearLayout>

<GridLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:layout\_height = "0dp"

android:layout\_width = "match\_parent"

android:layout\_rowWeight = "1"

android:background = "#ACD5C2"

android:rowCount = "1"

android:columnCount = "3"

android:orientation = "horizontal"

android:layout\_gravity = "fill"

>

<ImageButton

android:id = "@+id/icon\_button1"

android:layout\_height = "match\_parent"

android:layout\_width = "0dp"

android:layout\_columnWeight = "1"

android:src = "@drawable/icon1"

android:background = "?android:attr/selectableItemBackgroundBorderless"

android:contentDescription = "Icon button"

android:scaleType = "centerInside"

android:padding = "8dp"

/>

<ImageButton

android:id = "@+id/icon\_button2"

android:layout\_height = "match\_parent"

android:layout\_width = "0dp"

android:layout\_columnWeight = "1"

android:src = "@drawable/icon2"

android:background = "?android:attr/selectableItemBackgroundBorderless"

android:contentDescription = "Icon button"

android:scaleType = "centerInside"

android:padding = "8dp"

/>

<ImageButton

android:id = "@+id/icon\_button3"

android:layout\_height = "match\_parent"

android:layout\_width = "0dp"

android:layout\_columnWeight = "1"

android:src = "@drawable/icon3"

android:background = "?android:attr/selectableItemBackgroundBorderless"

android:contentDescription = "Icon button"

android:scaleType = "centerInside"

android:padding = "8dp"

/>

</GridLayout>

</GridLayout>

|  |  |  |
| --- | --- | --- |
| activity3.xml |  |  |

<?xml version = "1.0" encoding = "utf-8"?>

<GridLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:layout\_width = "fill\_parent"

android:layout\_height = "fill\_parent"

android:rowCount = "2"

android:columnCount = "1"

android:orientation = "horizontal"

android:layout\_gravity = "fill"

>

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:layout\_width = "match\_parent"

android:layout\_height = "0dp"

android:orientation = "vertical"

android:gravity = "center"

android:layout\_rowWeight = "9"

>

<ScrollView

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

>

<LinearLayout

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:gravity = "center"

android:background = "@drawable/bg"

android:orientation = "vertical"

android:padding = "16dp"

>

<TextView

android:id = "@+id/add\_remove\_movie"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Add / Remove Movie"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "center"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<LinearLayout

android:id = "@+id/add\_remove\_movie\_layout"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:gravity = "center"

android:orientation = "vertical"

android:padding = "16dp"

/>

<TextView

android:id = "@+id/add\_remove\_bookable\_movie"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Add / Remove Bookable Movie"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "center"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<LinearLayout

android:id = "@+id/add\_remove\_bookable\_movie\_layout"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:gravity = "center"

android:orientation = "vertical"

android:padding = "16dp"

/>

</LinearLayout>

</ScrollView>

</LinearLayout>

<GridLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:layout\_height = "0dp"

android:layout\_width = "match\_parent"

android:layout\_rowWeight = "1"

android:background = "#ACD5C2"

android:rowCount = "1"

android:columnCount = "3"

android:orientation = "horizontal"

android:layout\_gravity = "fill"

>

<ImageButton

android:id = "@+id/icon\_button1"

android:layout\_height = "match\_parent"

android:layout\_width = "0dp"

android:layout\_columnWeight = "1"

android:src = "@drawable/icon1"

android:background = "?android:attr/selectableItemBackgroundBorderless"

android:contentDescription = "Icon button"

android:scaleType = "centerInside"

android:padding = "8dp"

/>

<ImageButton

android:id = "@+id/icon\_button2"

android:layout\_height = "match\_parent"

android:layout\_width = "0dp"

android:layout\_columnWeight = "1"

android:src = "@drawable/icon2"

android:background = "?android:attr/selectableItemBackgroundBorderless"

android:contentDescription = "Icon button"

android:scaleType = "centerInside"

android:padding = "8dp"

/>

<ImageButton

android:id = "@+id/icon\_button3"

android:layout\_height = "match\_parent"

android:layout\_width = "0dp"

android:layout\_columnWeight = "1"

android:src = "@drawable/icon3"

android:background = "?android:attr/selectableItemBackgroundBorderless"

android:contentDescription = "Icon button"

android:scaleType = "centerInside"

android:padding = "8dp"

/>

</GridLayout>

</GridLayout>

|  |  |  |
| --- | --- | --- |
| add\_bookable\_movie.xml |  |  |

<RelativeLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/RelativeLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<ScrollView

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

>

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/LinearLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:orientation = "vertical"

android:gravity = "center"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/InnerLinearLayout1"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:orientation = "vertical"

android:gravity = "center"

android:padding = "16dp"

>

<TextView

android:id = "@+id/add\_bookable\_movie"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Add Bookable Movie"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "center"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<EditText

android:id = "@+id/editMovieName"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the title"

android:textColor = "#000000"

android:singleLine = "true"

android:inputType = "text"

android:textSize = "18sp"

android:layout\_marginBottom = "16dp"

android:padding = "10dp"

/>

<EditText

android:id = "@+id/editMovieTitlePath"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the movie's title image path. Eg : /storage/emulated/0/img.jpg"

android:textColor = "#000000"

android:layout\_marginBottom = "16dp"

android:inputType = "text"

android:textSize = "18sp"

android:singleLine = "true"

android:padding = "10dp"

/>

</LinearLayout>

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/InnerLinearLayout2"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:orientation = "vertical"

android:gravity = "center"

android:padding = "16dp"

>

<TextView

android:id = "@+id/add\_cinema"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Add Cinema"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "center"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<EditText

android:id = "@+id/editName"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the Cinema's name"

android:textColor = "#000000"

android:singleLine = "true"

android:inputType = "text"

android:textSize = "18sp"

android:layout\_marginBottom = "16dp"

android:padding = "10dp"

/>

<EditText

android:id = "@+id/editNumSeats"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the number of seats in the cinema"

android:textColor = "#000000"

android:layout\_marginBottom = "16dp"

android:inputType = "phone"

android:textSize = "18sp"

android:singleLine = "true"

android:padding = "10dp"

/>

</LinearLayout>

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/InnerLinearLayout3"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:orientation = "vertical"

android:gravity = "center"

android:padding = "16dp"

>

<TextView

android:id = "@+id/add\_runtime"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Add Runtime"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "center"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<EditText

android:id = "@+id/editStartDateTime"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "When does it start? [Format : yyyy-MM-dd HH:mm]"

android:textColor = "#000000"

android:singleLine = "true"

android:inputType = "text"

android:textSize = "18sp"

android:layout\_marginBottom = "16dp"

android:padding = "10dp"

/>

<EditText

android:id = "@+id/editEndDateTime"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "When does it end? [Format : yyyy-MM-dd HH:mm]"

android:textColor = "#000000"

android:layout\_marginBottom = "16dp"

android:inputType = "text"

android:textSize = "18sp"

android:singleLine = "true"

android:padding = "10dp"

/>

</LinearLayout>

</LinearLayout>

</ScrollView>

</RelativeLayout>

|  |  |  |
| --- | --- | --- |
| add\_movie.xml |  |  |

<RelativeLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/RelativeLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<ScrollView

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

>

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/LinearLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:orientation = "vertical"

android:gravity = "center"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<TextView

android:id = "@+id/add\_movie"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Add Movie"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "center"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<EditText

android:id = "@+id/editTitle"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the title"

android:textColor = "#000000"

android:singleLine = "true"

android:inputType = "text"

android:textSize = "18sp"

android:layout\_marginBottom = "16dp"

android:padding = "10dp"

/>

<EditText

android:id = "@+id/editImagePath"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the movie's poster path. Eg : /storage/emulated/0/img.jpg"

android:textColor = "#000000"

android:layout\_marginBottom = "16dp"

android:inputType = "text"

android:textSize = "18sp"

android:singleLine = "true"

android:padding = "10dp"

/>

<EditText

android:id = "@+id/editTitleImagePath"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the movie's title image path. Eg : /storage/emulated/0/img.jpg"

android:singleLine = "true"

android:layout\_marginBottom = "16dp"

android:textColor = "#000000"

android:inputType = "text"

android:textSize = "18sp"

android:padding = "10dp"

/>

<EditText

android:id = "@+id/editTrailerId"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the movie's trailer Id (youtube). Eg : LKFuXETZUsI [OPTIONAL]"

android:singleLine = "true"

android:layout\_marginBottom = "16dp"

android:textColor = "#000000"

android:inputType = "text"

android:textSize = "18sp"

android:padding = "10dp"

/>

<EditText

android:id = "@+id/editMovieEmbedCode"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the movie's embed code (HTML). [OPTIONAL]"

android:singleLine = "true"

android:layout\_marginBottom = "16dp"

android:textColor = "#000000"

android:inputType = "text"

android:textSize = "18sp"

android:padding = "10dp"

/>

<EditText

android:id = "@+id/editMovieURL"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the movie's URL. [OPTIONAL]"

android:singleLine = "true"

android:layout\_marginBottom = "16dp"

android:textColor = "#000000"

android:inputType = "text"

android:textSize = "18sp"

android:padding = "10dp"

/>

<EditText

android:id = "@+id/editMovieLocalPath"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the movie's local path. Eg : /storage/emulated/0/vid.mp4 [OPTIONAL]"

android:singleLine = "true"

android:layout\_marginBottom = "16dp"

android:textColor = "#000000"

android:inputType = "text"

android:textSize = "18sp"

android:padding = "10dp"

/>

<EditText

android:id = "@+id/editMovieDescription"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the movie's description [OPTIONAL]"

android:singleLine = "true"

android:layout\_marginBottom = "16dp"

android:textColor = "#000000"

android:inputType = "text"

android:textSize = "18sp"

android:padding = "10dp"

/>

<Spinner

android:id = "@+id/movieCategory"

android:layout\_width = "wrap\_content"

android:layout\_marginBottom = "16dp"

android:layout\_height = "wrap\_content"

android:gravity = "center"

/>

</LinearLayout>

</ScrollView>

</RelativeLayout>

|  |  |  |
| --- | --- | --- |
| enter\_details.xml |  |  |

<RelativeLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/RelativeLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<ScrollView

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

>

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/LinearLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:orientation = "vertical"

android:gravity = "center"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<TextView

android:id = "@+id/details\_label"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Enter Details"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "center"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<EditText

android:id = "@+id/editName"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter your name"

android:textColor = "#000000"

android:singleLine = "true"

android:inputType = "text"

android:textSize = "18sp"

android:layout\_marginBottom = "16dp"

android:padding = "10dp"

/>

<EditText

android:id = "@+id/editPhoneNum"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter your phone no."

android:textColor = "#000000"

android:layout\_marginBottom = "16dp"

android:inputType = "phone"

android:textSize = "18sp"

android:singleLine = "true"

android:padding = "10dp"

/>

<EditText

android:id = "@+id/editEmail"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter your email address"

android:singleLine = "true"

android:layout\_marginBottom = "16dp"

android:textColor = "#000000"

android:inputType = "textEmailAddress"

android:textSize = "18sp"

android:padding = "10dp"

/>

<Spinner

android:id = "@+id/cinemaSpinner"

android:layout\_width = "wrap\_content"

android:layout\_marginBottom = "16dp"

android:layout\_height = "wrap\_content"

android:gravity = "center"

/>

<Spinner

android:id = "@+id/runtimeSpinner"

android:layout\_width = "wrap\_content"

android:layout\_height = "wrap\_content"

android:gravity = "center"

/>

</LinearLayout>

</ScrollView>

</RelativeLayout>

|  |  |  |
| --- | --- | --- |
| movie\_url\_view.xml |  |  |

<?xml version = "1.0" encoding = "utf-8"?>

<RelativeLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/RelativeLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<VideoView

android:id = "@+id/videoView"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:layout\_centerInParent = "true"

/>

</RelativeLayout>

|  |  |  |
| --- | --- | --- |
| movie\_webview.xml |  |  |

<RelativeLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/RelativeLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

>

<WebView

android:id = "@+id/webView"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:layout\_centerInParent = "true"

/>

</RelativeLayout>

|  |  |  |
| --- | --- | --- |
| preview.xml |  |  |

<RelativeLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/RelativeLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

>

<ImageView

android:id = "@+id/background\_image"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:scaleType = "centerCrop"

android:layout\_alignParentTop = "true"

android:layout\_alignParentStart = "true"

/>

<LinearLayout

android:id = "@+id/LinearLayout"

android:layout\_width = "wrap\_content"

android:layout\_height = "wrap\_content"

android:orientation = "vertical"

android:layout\_alignParentBottom = "true"

android:layout\_alignParentStart = "true"

android:layout\_margin = "16dp"

/>

<LinearLayout

android:id = "@+id/WatchMoviesLayout"

android:layout\_width = "400dp"

android:layout\_height = "500dp"

android:orientation = "vertical"

android:layout\_alignParentBottom = "true"

android:layout\_alignParentEnd = "true"

android:layout\_margin = "16dp"

android:gravity = "center"

android:background = "#30000000"

>

<TextView

android:id = "@+id/watch\_movie"

android:text = "Watch Movie"

android:textColor = "@android:color/white"

android:textSize = "20sp"

android:textStyle = "bold"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:letterSpacing = "0.1"

android:gravity = "center"

/>

<GridLayout

android:layout\_width = "match\_parent"

android:layout\_height = "fill\_parent"

android:orientation = "vertical"

android:rowCount = "3"

android:columnCount = "1"

>

<ImageButton

android:id = "@+id/icon\_button1"

android:layout\_width = "match\_parent"

android:layout\_height = "0dp"

android:layout\_rowWeight = "1"

android:src = "@drawable/embed\_movie"

android:scaleType = "fitCenter"

android:background = "@null"

/>

<ImageButton

android:id = "@+id/icon\_button2"

android:layout\_width = "match\_parent"

android:layout\_height = "0dp"

android:layout\_rowWeight = "1"

android:src = "@drawable/movie\_url"

android:scaleType = "fitCenter"

android:background = "@null"

/>

<ImageButton

android:id = "@+id/icon\_button3"

android:layout\_width = "match\_parent"

android:layout\_height = "0dp"

android:layout\_rowWeight = "1"

android:src = "@drawable/local\_path"

android:scaleType = "fitCenter"

android:background = "@null"

/>

</GridLayout>

</LinearLayout>

</RelativeLayout>

|  |  |  |
| --- | --- | --- |
| remove\_bookable\_movie.xml |  |  |

<RelativeLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/RelativeLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<ScrollView

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

>

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/LinearLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:orientation = "vertical"

android:gravity = "center"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<TextView

android:id = "@+id/remove\_bookable\_movie"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Remove Bookable Movie"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "center"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<EditText

android:id = "@+id/editTitle"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the title"

android:textColor = "#000000"

android:singleLine = "true"

android:inputType = "text"

android:textSize = "18sp"

android:layout\_marginBottom = "16dp"

android:padding = "10dp"

/>

</LinearLayout>

</ScrollView>

</RelativeLayout>

|  |  |  |
| --- | --- | --- |
| remove\_movie.xml |  |  |

<RelativeLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/RelativeLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<ScrollView

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

>

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/LinearLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:orientation = "vertical"

android:gravity = "center"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<TextView

android:id = "@+id/remove\_movie"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Remove Movie"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "center"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<EditText

android:id = "@+id/editTitle"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:hint = "Enter the title"

android:textColor = "#000000"

android:singleLine = "true"

android:inputType = "text"

android:textSize = "18sp"

android:layout\_marginBottom = "16dp"

android:padding = "10dp"

/>

</LinearLayout>

</ScrollView>

</RelativeLayout>

|  |  |  |
| --- | --- | --- |
| select\_seat.xml |  |  |

<RelativeLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/RelativeLayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<ScrollView

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

>

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:id = "@+id/linearlayout"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:orientation = "vertical"

android:gravity = "center"

android:padding = "16dp"

android:background = "@drawable/bg"

>

<TextView

android:id = "@+id/seat\_label"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "Select Seat"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "center"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

</LinearLayout>

</ScrollView>

</RelativeLayout>

|  |  |  |
| --- | --- | --- |
| ticket.xml |  |  |

<LinearLayout

xmlns:android = "http://schemas.android.com/apk/res/android"

android:layout\_width = "match\_parent"

android:layout\_height = "match\_parent"

android:orientation = "vertical"

android:gravity = "center"

android:padding = "16dp"

android:background = "@drawable/ticket\_bg"

>

<TextView

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:text = "TICKET"

android:textColor = "@android:color/white"

android:textSize = "30sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "center"

android:layout\_marginBottom = "12dp"

android:letterSpacing = "0.05"

android:shadowColor = "@android:color/black"

android:shadowDx = "2"

android:shadowDy = "2"

android:shadowRadius = "3"

/>

<TextView

android:id = "@+id/ticket\_details"

android:layout\_width = "match\_parent"

android:layout\_height = "wrap\_content"

android:textColor = "@android:color/white"

android:textSize = "20sp"

android:textStyle = "bold"

android:paddingStart = "16dp"

android:paddingEnd = "16dp"

android:paddingTop = "8dp"

android:paddingBottom = "8dp"

android:gravity = "center"

/>

</LinearLayout>

|  |  |  |
| --- | --- | --- |
| rounded\_textview.xml |  |  |

<?xml version = "1.0" encoding = "utf-8"?>

<shape

xmlns:android = "http://schemas.android.com/apk/res/android"

>

<solid

android:color = "#ACD5C2"

/>

<corners

android:radius = "16dp"

/>

<padding

android:left = "8dp"

android:top = "8dp"

android:right = "8dp"

android:bottom = "8dp"

/>

<stroke

android:width = "7dp"

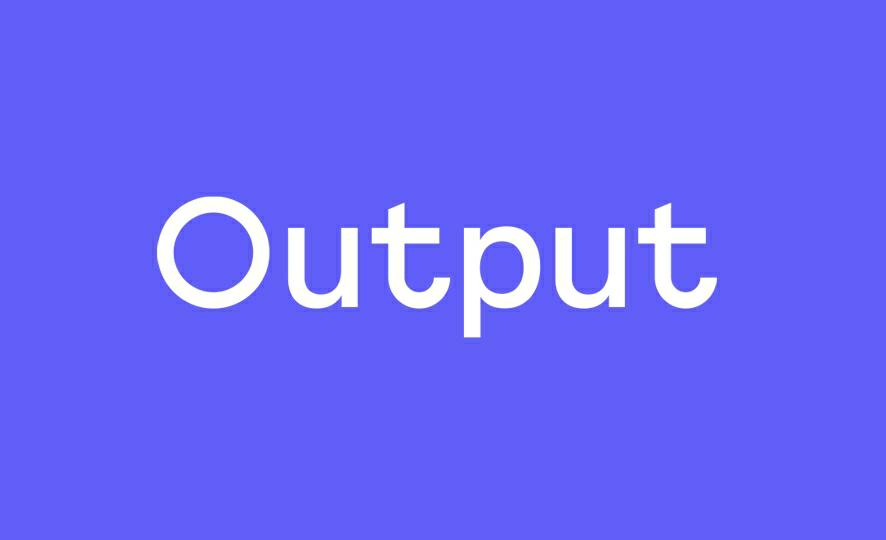
android:color = "#83B09B"

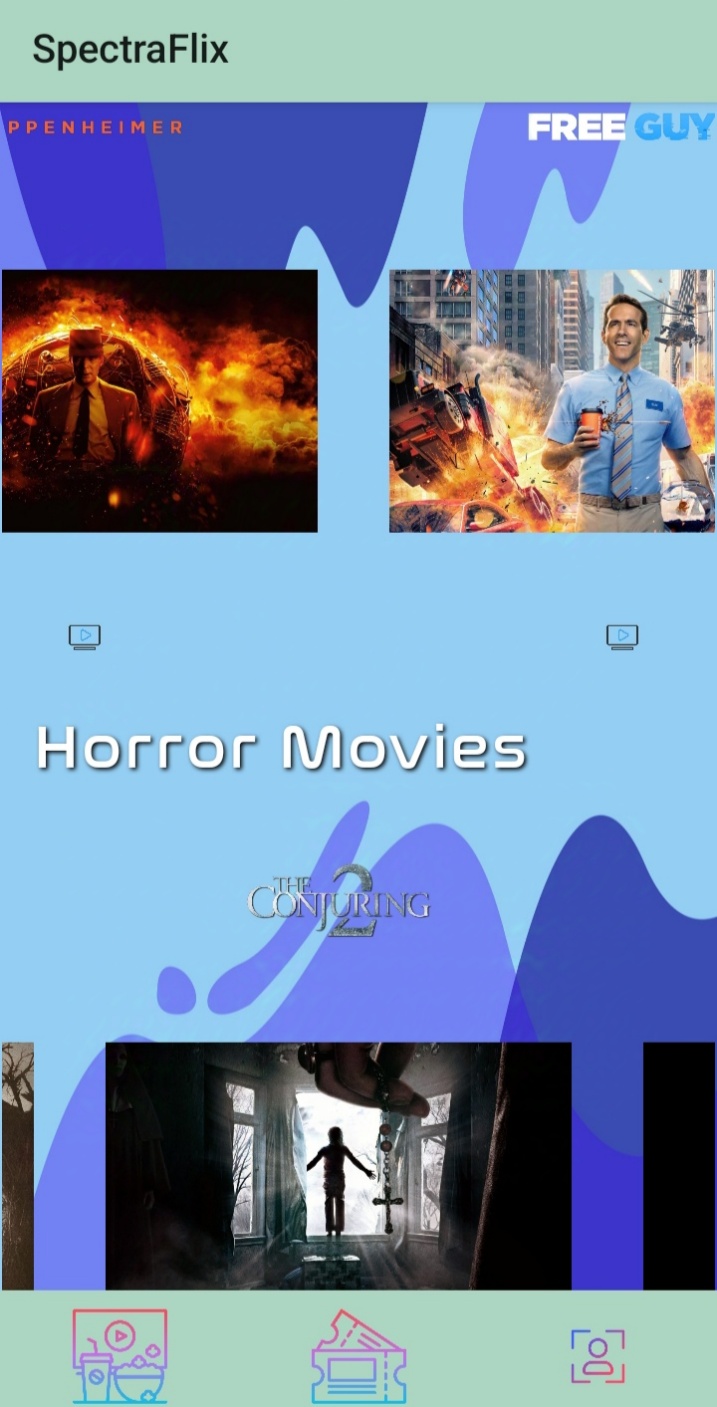
/>

</shape>

|  |  |  |
| --- | --- | --- |
| VARIABLE DESCRIPTION |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **VARIABLE(S)** | **DATA TYPE** | | **PURPOSE** | | |
| [Some variables perform the same function and have the same data type and hence, they are grouped together] |  | |  | | |
| resMovieTitlePath, resTitleImagePath | int | | Stores the numerical path of the movie title image located in the res directory | | |
| movieName | String | | Stores the name of the movie | | |
| movieTitlePath | String | | Stores the external path of the movie title image | | |
| cinemas | List<Cinema> | | Stores the cinemas which run the movie | | |
| animatedMovies | List<Movie> | | Stores the animated movies | | |
| horrorMovies | List<Movie> | | Stores the horror movies | | |
| popularMovies | List<Movie> | | Stores the popular movies | | |
| developersChoiceMovies | List<Movie> | | Stores the movies chosen by the developer(i.e. Omanshu) | | |
| displayMetrics, dM | DisplayMetrics | | An object of the DisplayMetrics class used to collect the width and height of the screen | | |
| **VARIABLE(S)** | **DATA TYPE** | | **PURPOSE** | | |
| [Some variables perform the same function and have the same data type and hence, they are grouped together] |  | |  | | |
| width | int | | Stores the width of the screen in pixels | | |
| height | int | | Stores the height of the screen in pixels | | |
| customFont | Typeface | | An object of the Typeface class which creates a custom font (i.e. nasalization.ttf) | | |
| removeTextView, addTextView, movieView, bookableMovieView,  movieTextView,  textView,  seatTextView | TextView | | An object of the TextView class which stores text view used to display text in the app | | |
| intent | Intent | | An object of the Intent class used to switch between activities and transfer data between them | | |
| relativeLayout | RelativeLayout | | An object of the RelativeLayout class used to store widgets | | |
| linearLayout, linearLayout2 | LinearLayout | | An object of the LinearLayout class used to store widgets | | |
| editTitle, editImagePath, editTitleImagePath, editTrailerId, editCode, editURL, editLocalPath, editDescription, editTitlePath, editNumSeats, editStart, editEnd, editName, editPhoneNum, editEmail | EditText | | An object of the EditText class used to take text inputs from the user (title, image path, title image path, trailer Id, embed code, movie URL, movie local path, description, title path, number of seats, start time, end time, name of he user, phone number, email address of the user) | | |
| **VARIABLE(S)** | | **DATA TYPE** | | **PURPOSE** | |
| [Some variables perform the same function and have the same data type and hence, they are grouped together] | |  | |  | |
| title | | String | | Stores the title of the movie | |
| dataManager | | DataManager | | An object of the custom class “DataManager” used to share data and objects between activities | |
| webView | | WebView | | An object of the WebView class used to display WebViews | |
| webSettings | | WebSettings | | An object of the WebSettings class used to manipulate the settings of WebView | |
| movieEmbedCode | | String | | Stores the HTML code of the movie (through the streaming service, Eg : GDrive) used in the WebView | |
| movieURL | | String | | Stores the URL of the movie(through the streaming service) used in the VideoView | |
| videoView | | VideoView | | An object of the VideoView class used to play videos and movies | |
| uri | | Uri | | An object of the Uri class used to parse the video and image paths of String data type | |
| mediaController | | MediaController | | An object of the MediaController class used to give functionalities to the ViewView | |
| **VARIABLE(S)** | **DATA TYPE** | | **PURPOSE** | | |
| [Some variables perform the same function and have the same data type and hence, they are grouped together] |  | |  | | |
| movieTrailerId | String | | Stores the YouTube video Id of the trailer (of the movie) | | |
| trailerEmbedCode | String | | Stores the HTML code of the YouTube trailer used in the WebView | | |
| moviePath | String | | Stores the external movie path located in the device storage | | |
| resImagePath | int | | Stores the external movie poster path located in the res directory | | |
| description | String | | Stores the description of the movie | | |
| localPath | String | | Stores the external path of the movie located in the device storage | | |
| buttonParams, addButtonParams | LayoutParams | | An object of the LayoutParams class used to modify Buttons and ImageButtons | | |
| imageButton | ImageButton | | An object used to create or access ImageButton widget | | |
| categorySpinner | Spinner | | An object of the Spinner class used to create a Spinner of various movie categories | | |
| selectedCategory | String | | Stores the movie category selected by the user | | |
| categoriesAdapter | ArrayAdapter<String> | | An adapter used to create the categories using a String array | | |
| imagePath | String | | Stores the image path of the movie | | |
| titleImagePath, path | String | | Stores the title image path of the movie | | |
| **VARIABLE(S)** | **DATA TYPE** | | **PURPOSE** | | |
| [Some variables perform the same function and have the same data type and hence, they are grouped together] |  | |  | | |
| trailerId | String | | Stores the movie trailer Id(YouTube Id) | | |
| code | String | | Stores the movie embed code | | |
| url | String | | Stores the movie URL | | |
| description | String | | Stores the decription of the movie | | |
| category | int | | Stores the selected movie category in integer format | | |
| newMovie | Movie | | An object of the custom class “Movie” used to stores the newly added movie | | |
| name | String | | Stores the name of the Cinema | | |
| seatRowCol, rowCol | int[] | | Stores data in this format -> [No. of rows, No. of Columns] | | |
| runtimes | List<Runtime> | | A list of the custom class “Runtime” | | |
| formatter | DateTimeFormatter | | Used to format DateTime data | | |
| start, startDateTime | LocalDateTime | | Stores the date and time when the runtime begins | | |
| end, endDateTime | LocalDateTime | | Stores the date and time when the runtime ends | | |
| rt | Runtime | | Stores the Runtime objects contained in the runtimes variable | | |
| startRt | LocalDateTime | | Stores the date and time when the runtime “rt” begins | | |
| endRt | LocalDateTime | | Stores the date and time when the runtime “rt” ends | | |
| runtime | Runtime | | Stores the newly created Runtime object | | |
| **VARIABLE(S)** | | **DATA TYPE** | | **PURPOSE** | |
| [Some variables perform the same function and have the same data type and hence, they are grouped together] | |  | |  | |
| i | | int | | Used to control the outer loop | |
| j | | int | | Used to control the inner nested loop | |
| listOfFactors | | List<int[]> | | Stores the factors pair in a list | |
| num | | int | | Stores the number given to the “getMinRowCol()” function | |
| factors | | int[] | | Stores the factors pair | |
| outArr | | int[] | | Stores the final factors pair | |
| diff | | int | | Stores the difference between the factors pair | |
| movies | | List<Movie> | | Stores the movies in a list | |
| bookableMovies | | List<BookableMovie> | | Stores the bookable movies in a list | |
| cinema1, cinema2 | | Cinema | | Cinema objects contained in the bookable movies | |
| movie1, movie2, movie3, movie4 | | BookableMovie | | BookableMovie objects contained in the “bookableMovies” list | |
| text | | String | | Stores the text displayed in the ticket | |
| button1, button2, button3 | | ImageButton | | Stores the Image Button widgets of the bottom navigation bar | |
| mov\_final | | BookableMovie | | Stores the newly created BookableMovie object | |
| addMovieButton | | ImageButton | | Stores the ImageButton widget used to add a movie | |
| **VARIABLE(S)** | | **DATA TYPE** | | **PURPOSE** | |
| [Some variables perform the same function and have the same data type and hence, they are grouped together] | |  | |  | |
| addMovieParams | | LayoutParams | | Stores the parameters of “addMovieButton” | |
| addBookableMovieButton | | ImageButton | | Stores the ImageButton widget used to add a bookable movie | |
| addBookableMovieParams | | LayoutParams | | Stores the parameters of “addBookableMovieButton” | |
| removeMovieButton | | ImageButton | | Stores the ImageButton widget used to remove a movie |
| removeMovieParams | | LayoutParams | | Stores the parameters of “removeMovieButton” |
| removeBookableMovieButton | | ImageButton | | Stores the ImageButton widget used to remove a bookable movie | |
| removeBookableMovieParams | | LayoutParams | | Stores the parameters of “removeBookableMovieButton” | |
| addTextView1, addTextView2, addTextView3 | | TextView | | Stores the text views contained in the “addBookableMovie” activity | |
| addButton, addButton1, addButton2, addButton3 | | ImageButton | | Stores the image buttons contained in the “addBookableMovie” activity | |
| addButtonParams, addButtonParams1, addButtonParams2, addButtonParams2 | | LayoutParams | | Stores the parameters of the image buttons contained in the “addBookableMovie” activity | |
| **VARIABLE(S)** | **DATA TYPE** | | **PURPOSE** | | |
| [Some variables perform the same function and have the same data type and hence, they are grouped together] |  | |  | | |
| seats | int[][] | | Stores a double dimensional array containing the seats in the cinema | | |
| bgImage | ImageView | | Stores the ImageView object containing the background image | | |
| buttonsLayout | LinearLayout | | An object of LinearLayout class containing the ImageButtons for streaming the movie | | |
| buttonsParams | LayoutParams | | Stores the parameters for the buttons containined in the “buttonsLayout” | | |
| trailerButton | ImageButton | | An object of the ImageButton class used to display the trailer | | |
| trailerParams | LayoutParams | | Stores the parameters for “trailerButton” | | |
| buttonIds | int[] | | Stores the Ids of the buttons in the ”buttonsLayout” | | |
| strIds | String[] | | Stores the Ids of buttons contained in the “buttonsLayout” in String format | | |
| strData | String[] | | Stores the data to be passed into the intent through the buttons contained in the “buttonsLayout” | | |
| classes | Class<?>[] | | Stores the activity classes to be accessed through the buttons in the “buttonsLayout” | | |
| button | ImageButton | | Accesses each button contained in the “buttonsLayout” inside a for loop | | |
| **VARIABLE(S)** | **DATA TYPE** | | **PURPOSE** | | |
| [Some variables perform the same function and have the same data type and hence, they are grouped together] |  | |  | | |
| movieStr | String | | Accesses each String data contained in the “strData” array | | |
| buttonArray | ImageButton[][] | | Stores an ImageButton for each seat in the cinema | | |
| movie\_index, cinema\_index, runtime\_index | int | | Stores the index of the exact runtime to be accessed inside the cinema withih the movie class | | |
| selected\_row | int | | Stores the row in which the selected seat is located | | |
| selected\_col | int | | Stores the column in which the selected seat is located | | |
| backButton | ImageButton | | Stores the ImageButton widget used to go to the previous activity | | |
| backButtonParams | LayoutParams | | Stores the parameters for the “backButton” | | |
| data, data\_new | String[] | | Stores the data for the ticket | | |
| rows | int | | Stores the number of rows in the cinema | | |
| cols | int | | Stores the number of columns in the Cinema | | |
| gridLayout | GridLayout | | An object of the “GridLayout” class which stores the ImageButtons representing the seats | | |
| currentRow | int | | Stores the row number in the outer nested for loop | | |
| currentCol | int | | Stores the column number in the inner nested for loop | | |
| bookButton | ImageButton | | Used to book a seat in the cinema | | |
| **VARIABLE(S)** | **DATA TYPE** | | **PURPOSE** | | |
| [Some variables perform the same function and have the same data type and hence, they are grouped together] |  | |  | | |
| bookButtonParams | LayoutParams | | Stores the parameters for the “bookButton” | | |
| new\_data | String[] | | Stores the modified data for the ticket | | |
| actionBar | ActionBar | | Stores the top action bar widget | | |
| textView1, textView2, textView3, textView4 | TextView | | Stores the TextView widget for each category | | |
| categories | streamingMovies | | An object of the custom “streamingMovies” class used to get the movies in each category | | |
| mov | Movie | | Stores each movie contained in each category of “categories” | | |
| animatedMovies, popularMovies,  horrorMovies,  developersChoiceMovies | List<Movie> | | Stores a list containing movies in each category | | |
| moviesList | List<Movie>[] | | Stores each category in a list | | |
| scrollView | int | | Stores the HorizontalScrollView Id for a specific category | | |
| scrollViews | int[] | | Stores the HorizontalScrollView Id for all categories | | |
| parentLayout | LinearLayout | | Stores the LinearLayout widget containing a specific HorizontalScrollView | | |
| newLayout | LinearLayout | | Stores a new LinearLayout for the newly added movie | | |
| **VARIABLE(S)** | **DATA TYPE** | | **PURPOSE** | | |
| [Some variables perform the same function and have the same data type and hence, they are grouped together] |  | |  | | |
| movieTitleView | ImageView | | Stores the ImageView widget for displaying the title image | | |
| movieTitleParams | LayoutParams | | Stores the parameters for the “movieTitleView” | | |
| moviePosterView | ImageView | | Stores the ImageView widget for displaying the poster of the movie | | |
| moviePosterParams | LayoutParams | | Stores the parameters for the “moviePosterView” | | |
| file | File | | An object of the File class which accepts the external image path | | |
| imageUri | Uri | | Converts the image path to uri format | | |
| bitmap | Bitmap | | An object of the Bitmap class used to create a bitmap of the image | | |
| selectedCinema | String | | Stores the name of the cinema selected by the user | | |
| selectedRuntime | String | | Stores the runtime selected by the user | | |
| detailsTextView | TextView | | Stores the TextView widget for the “EnterDetails” activity | | |
| movie | BookableMovie | | Stores the movie for which the booking is to be done | | |
| cinemaSpinner | Spinner | | Stores the Spinner widget containing the cinemas to be selected | | |
| cinemaNames | String[] | | Stores the names of the cinemas | | |
| cinemaAdapter | ArrayAdapter<String> | | Stores the Adapter for the “cinemaSpinner” | | |
| **VARIABLE(S)** | | **DATA TYPE** | | **PURPOSE** | |
| [Some variables perform the same function and have the same data type and hence, they are grouped together] | |  | |  | |
| cinemas\_final | | List<Cinema> | | Stores the list of the Cinemas | |
| cinema\_item | | Cinema | | Stores a Cinema object in a for loop | |
| runtimes | | List<Runtime> | | Stores all the runtimes of a specific cinema | |
| runtimes\_string | | String[] | | Stores the startDateTime of each runtime in “runtimes” | |
| runtimeSpinner | | Spinner | | Stores the Spinner containing the runtimes to be selected | |
| runtimeAdapter | | ArrayAdapter<String> | | Stores the Adapter for the “runtimeSpinner” | |
| nextButton | | ImageButton | | Store the ImageButton which leads to the “SelectSeat” activity | |
| nextButtonParams | | LayoutParams | | Stores the parameters for the “nextButton” | |
| enteredName | | String | | Stores the name of the user | |
| enteredPhoneNum | | String | | Stores the phone number of the user | |
| enteredEmail | | String | | Stores the email address of the user | |
| data | | String[] | | Stores the data for the ticket | |
| numSeats | | String | | Stores the number of seats in the cinema in String format | |





**Preview Movie**

**View Movies (For streaming and watching trailers)**



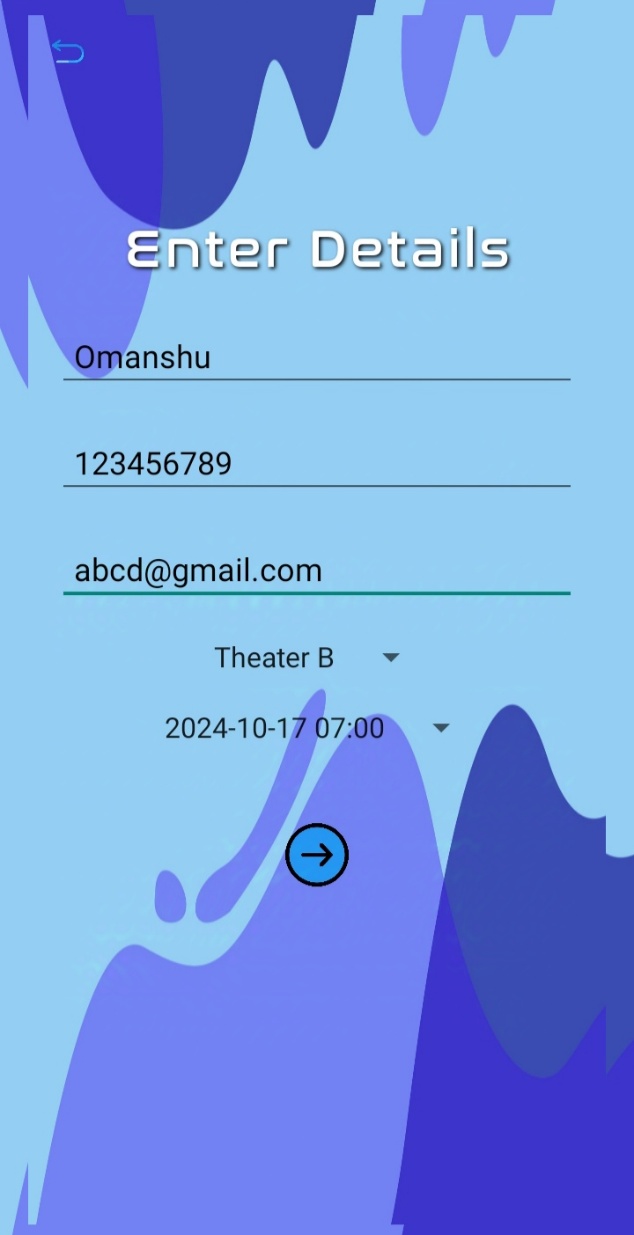
**Watch Movie(URL)**

**Watch Movie(WebView)**



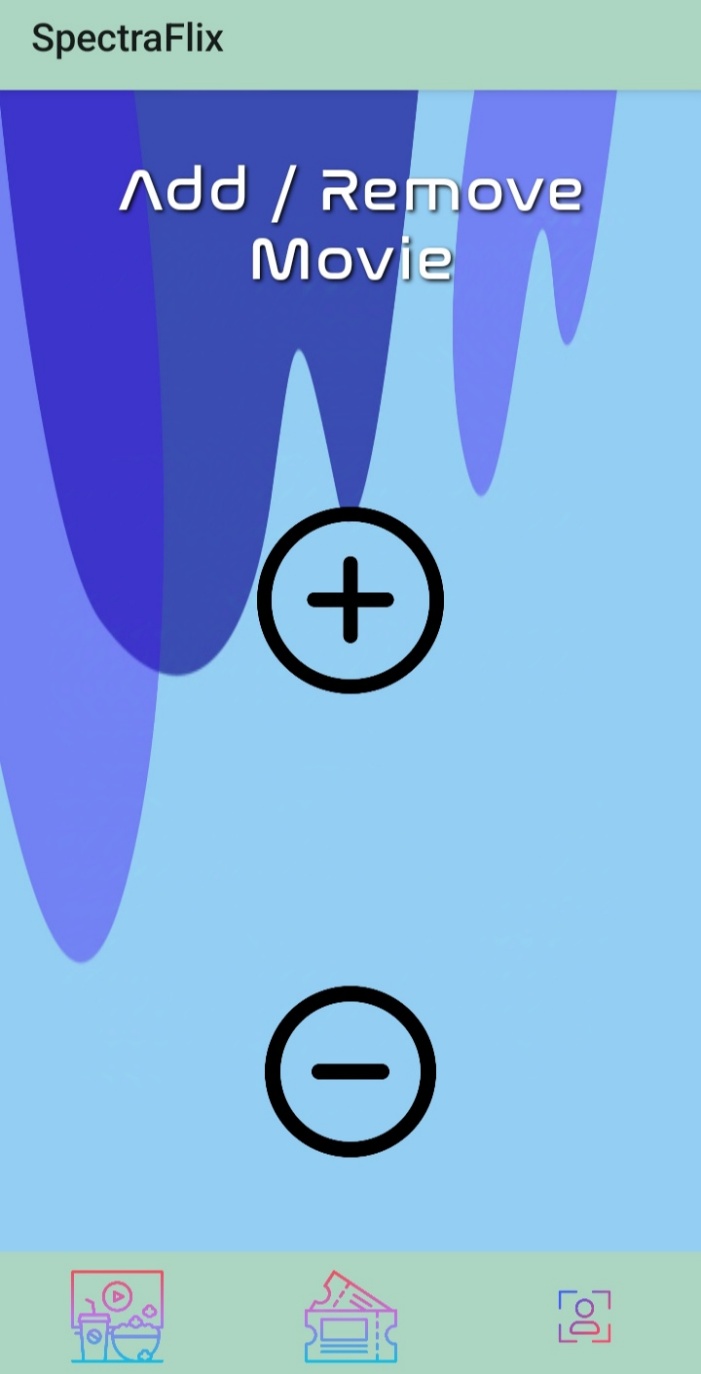
**View Movies (For booking tickets)**

**Watch Trailer (YouTube / WebView)**



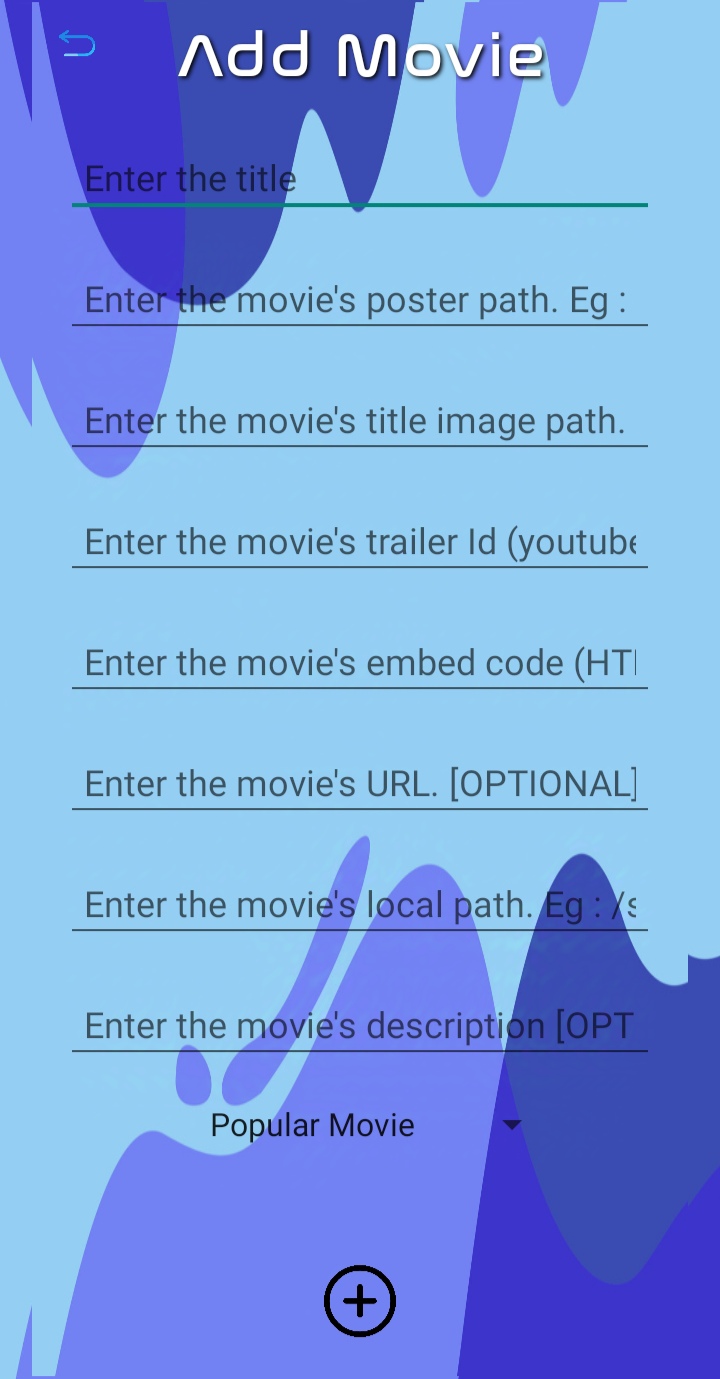
**Enter Your Details And Select Cinema And Runtime**

**Select Your Desired Seat**



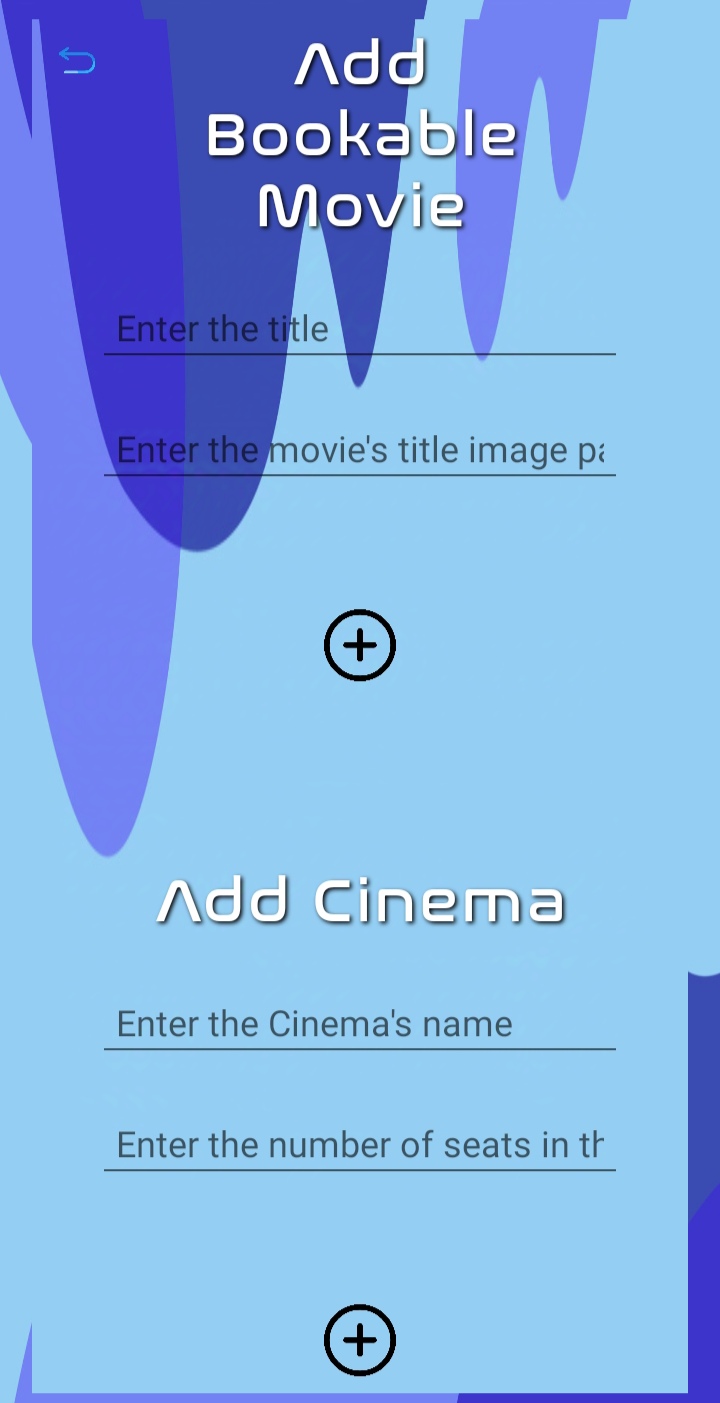
**Add / Remove Movie and Bookable Movie**

**Print Ticket**



**Remove Movie**

**Add Movie**



**Remove Bookable Movie**

**Add Bookable Movie**

|  |  |  |
| --- | --- | --- |
| Conclusion |  |  |

In conclusion, **SpectraFlix** is a comprehensive movie streaming and booking platform that delivers a seamless user experience. With its ability to stream movies from various sources, provide interactive trailer viewing, and offer an intuitive seat booking system, the app meets the diverse needs of modern movie enthusiasts.

Through the development of **SpectraFlix**, I have gained invaluable insights into Java programming, XML layout design, and the Android development module. I have learned how to integrate multiple features into a single cohesive app, ensuring both performance and user satisfaction. This project has not only sharpened my technical skills but also enhanced my ability to create user-friendly applications