НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №4

з дисципліни

“Програмування мобільних систем”

Виконав:

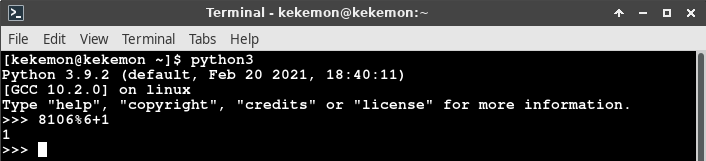
студент групи IВ-81

ЗК 8106

Бухтій О.В.

Київ 2021

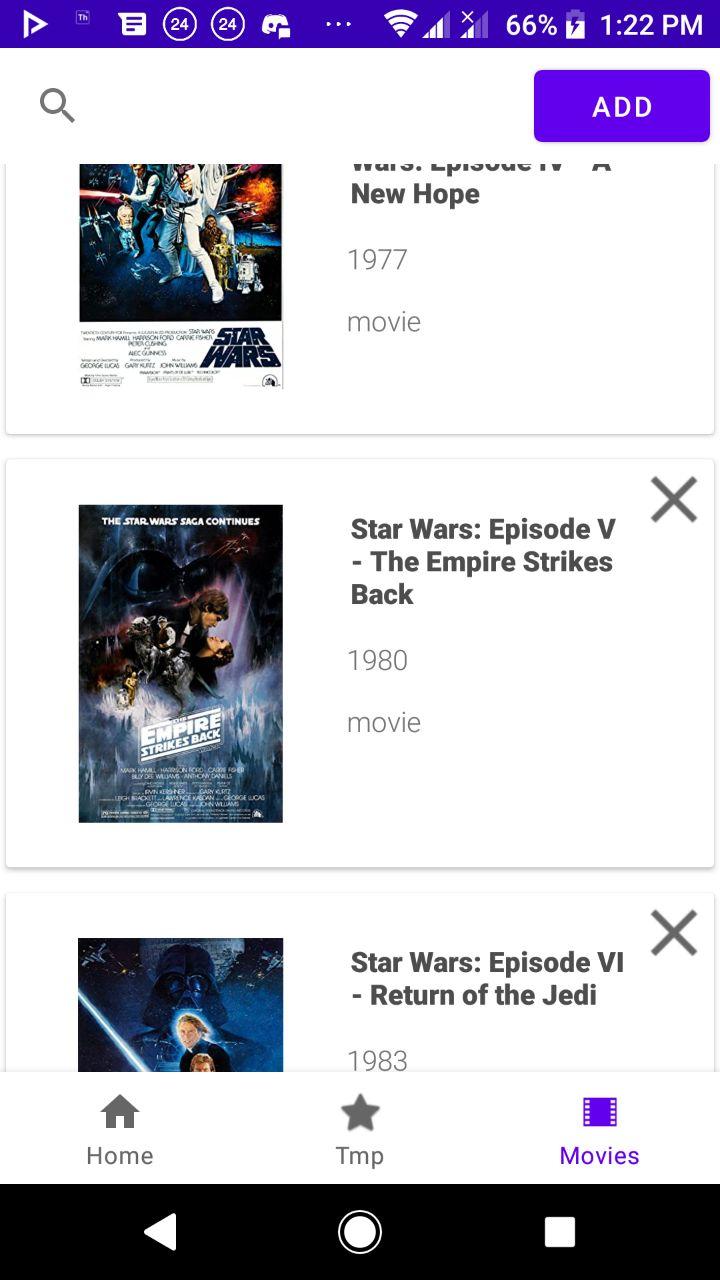
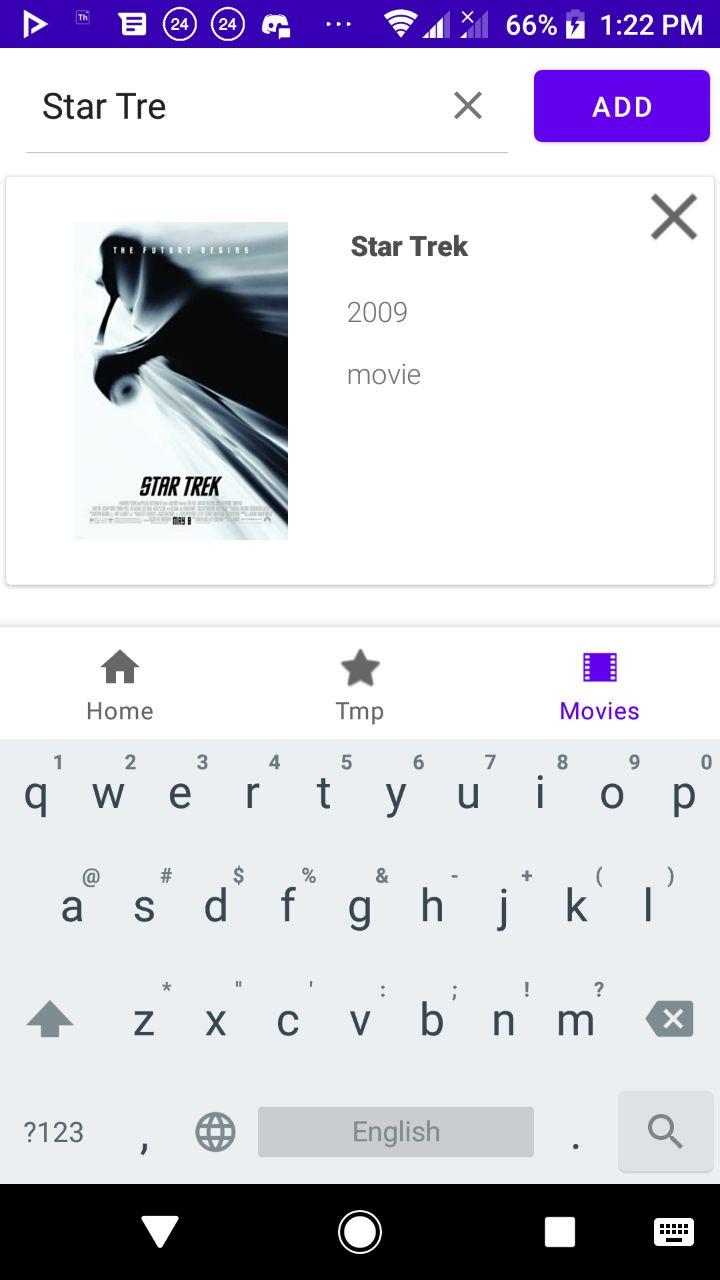
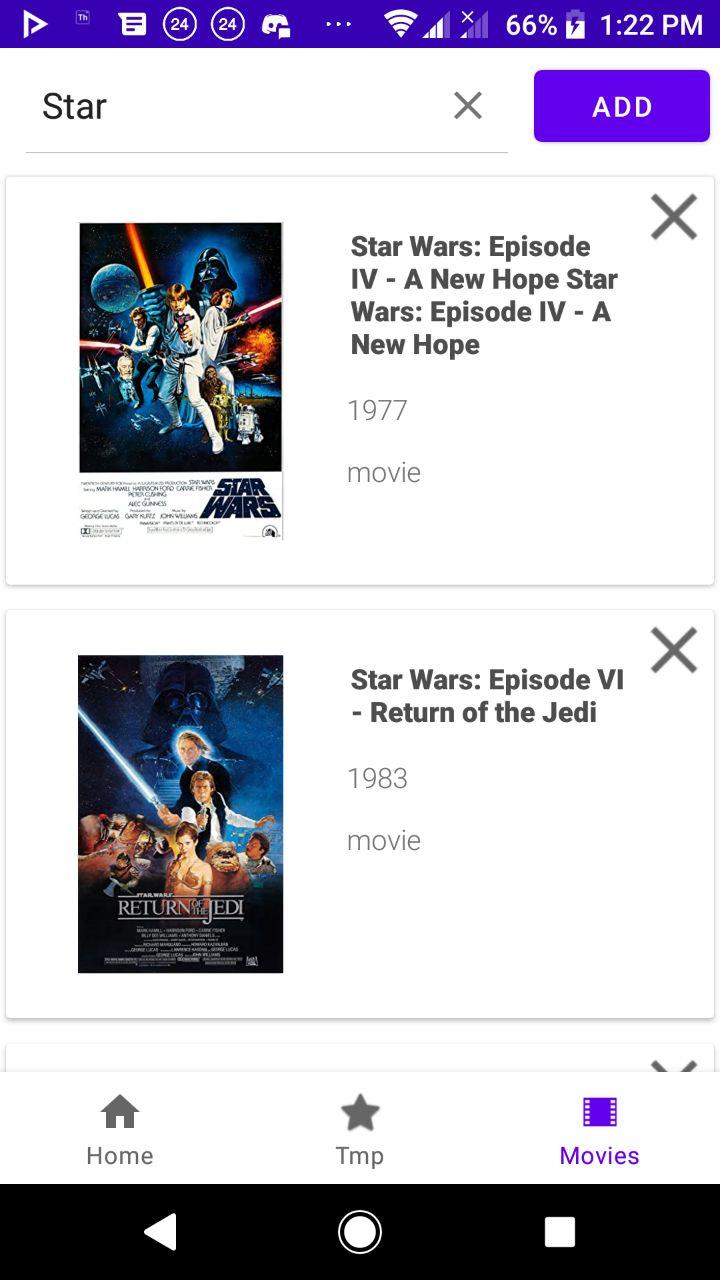
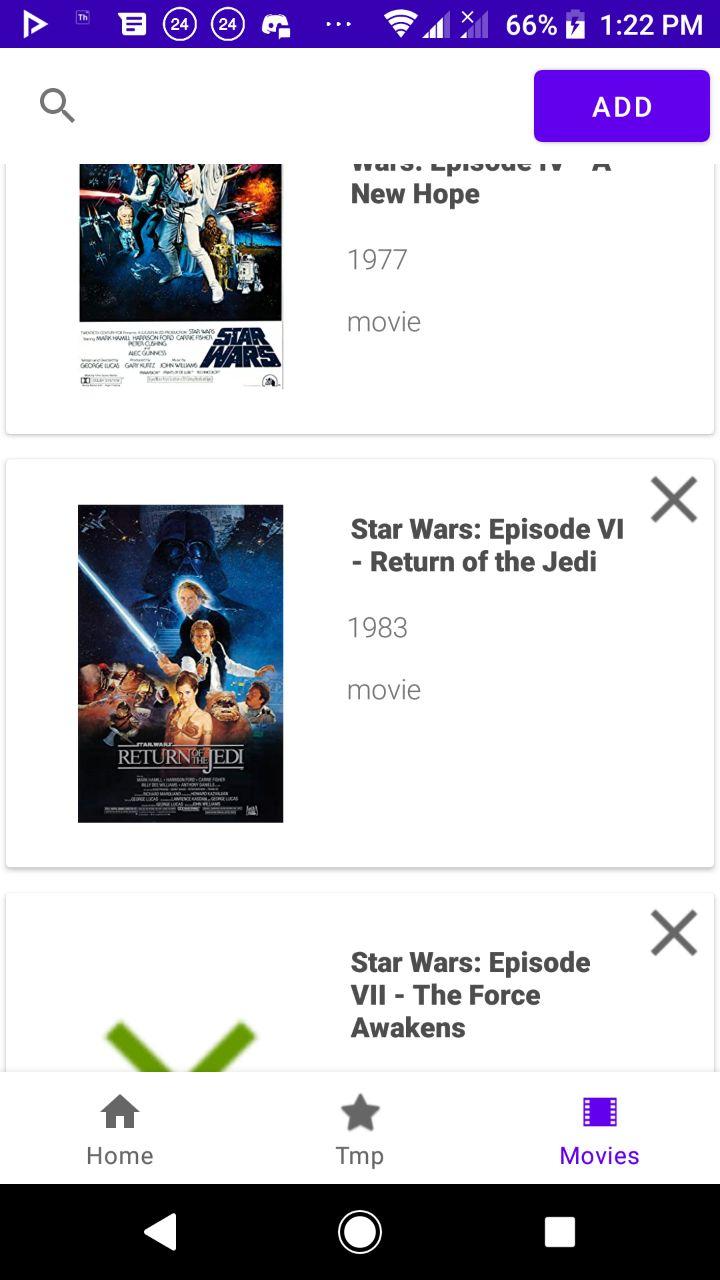
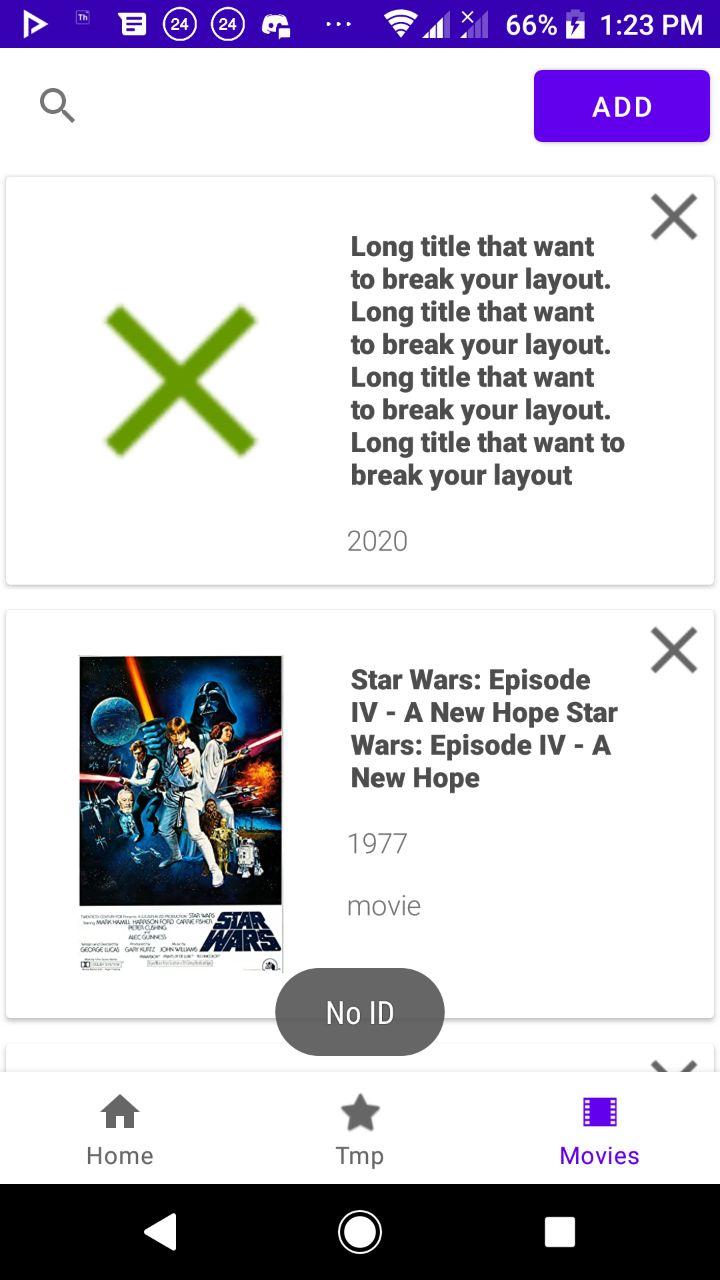
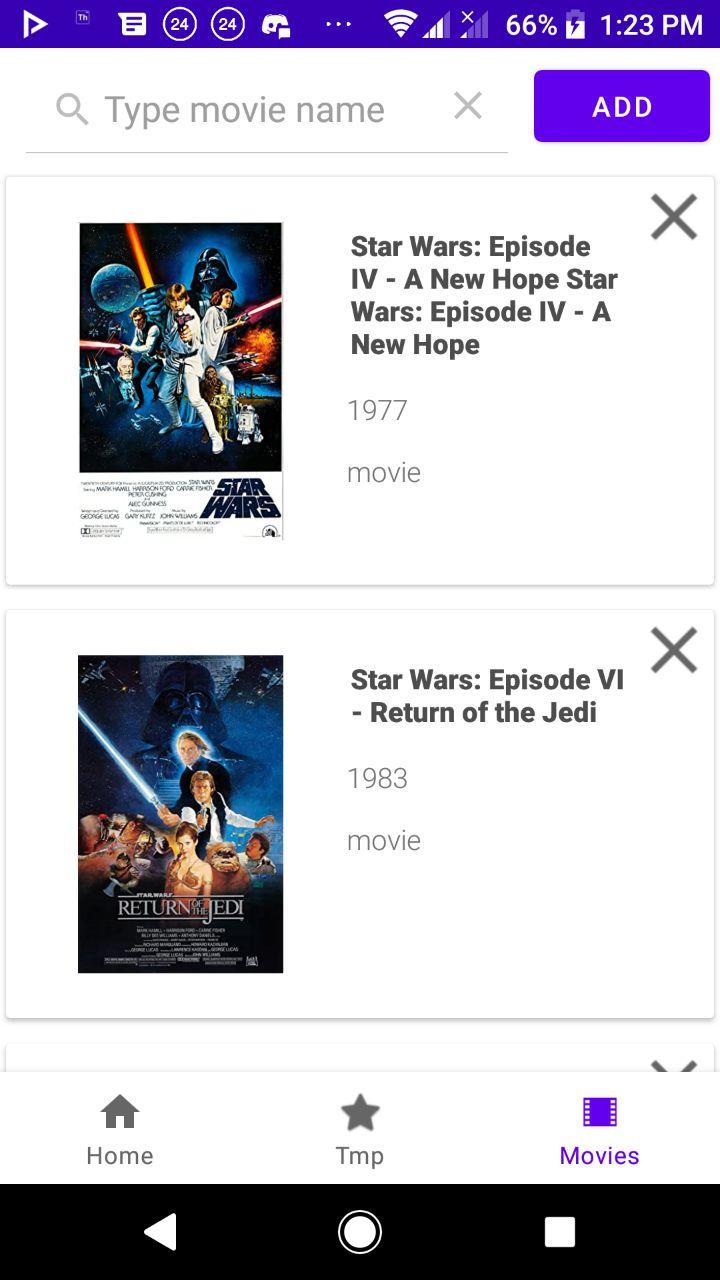
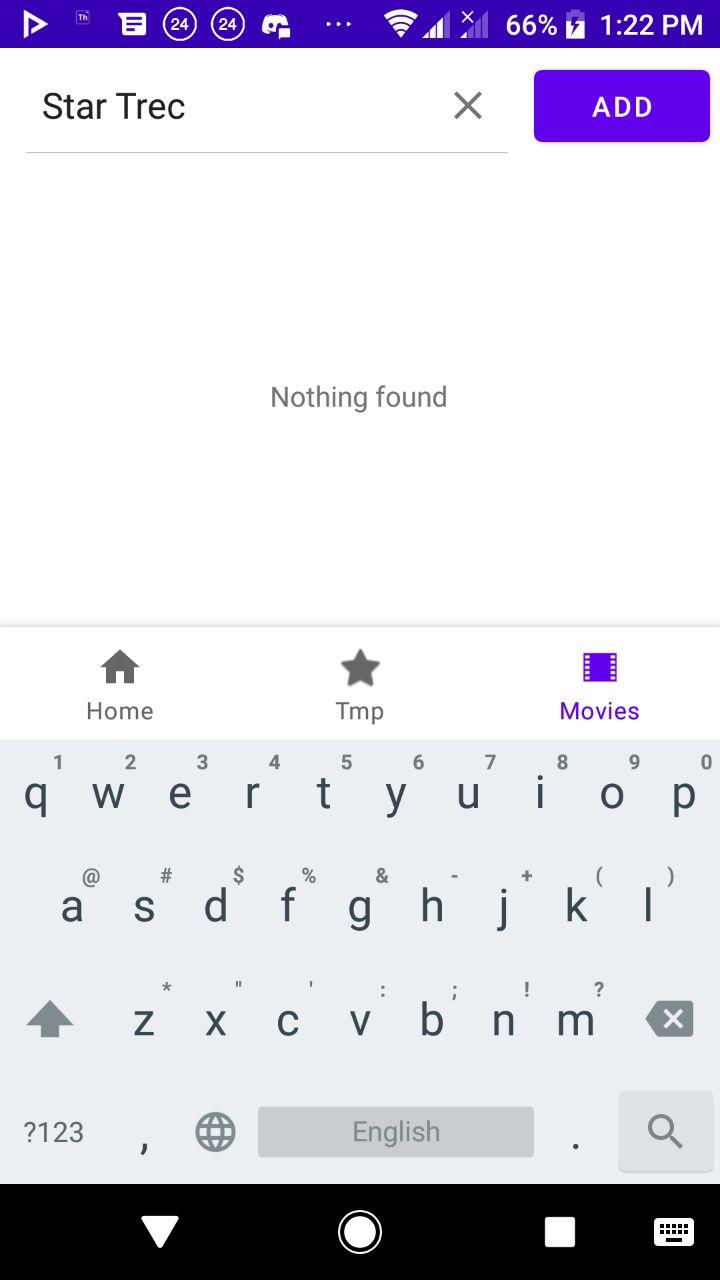
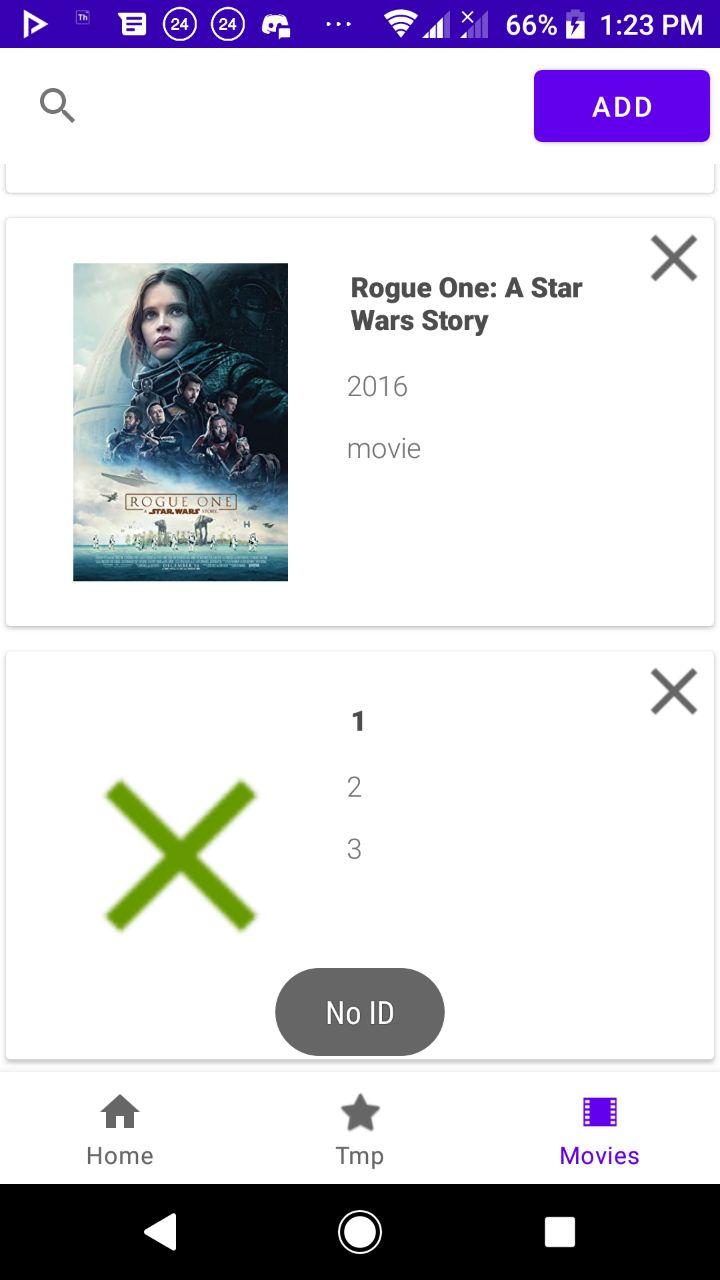
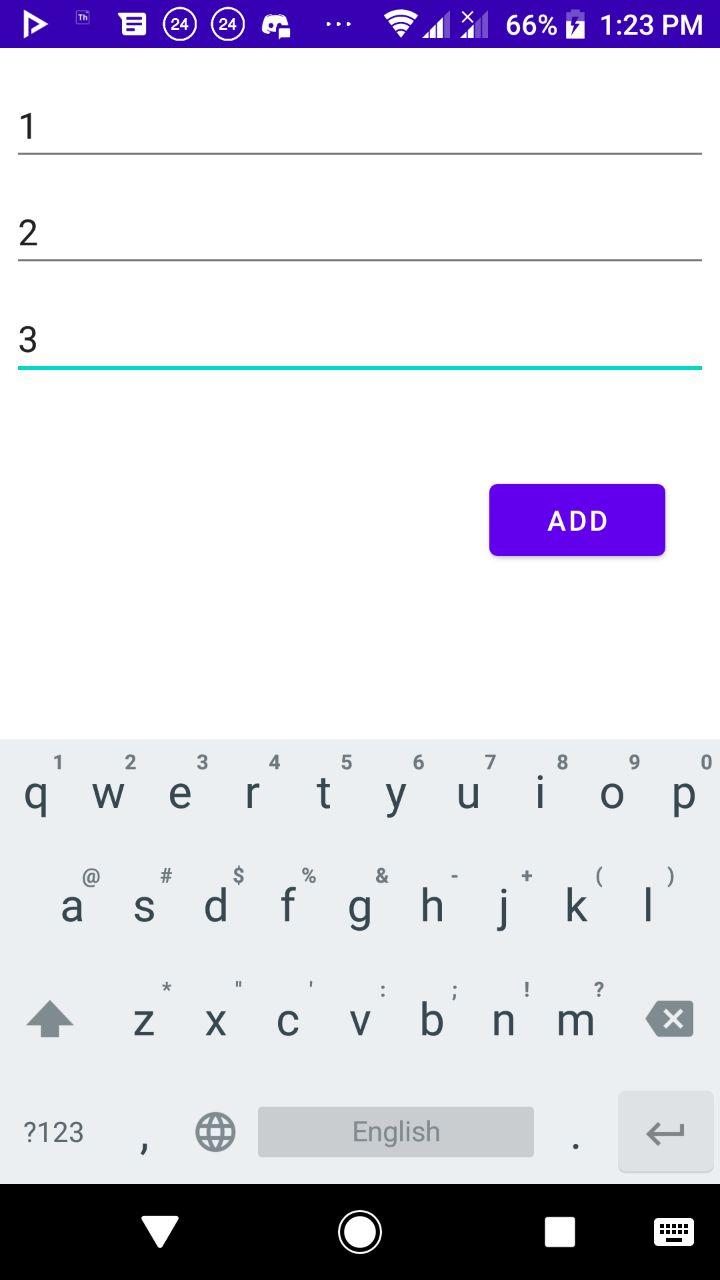
1. **Визначення варіанту:**



Посилання на репозиторій: [https://github.com/KekemonBS/PMS/tree/main/LAB\_](https://github.com/KekemonBS/PMS/tree/main/LAB_5)5

1. **Виконання:**

Скріншоти:

****

1. **Лістинг коду (що були створені/ змінені):**

**---MoviesAdapter.java---**

package ua.kpi.comsys.iv8106.adapters;

import android.content.Intent;

import android.view.LayoutInflater;

import android.view.MotionEvent;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.fragment.app.Fragment;

import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

import ua.kpi.comsys.iv8106.R;

import ua.kpi.comsys.iv8106.model.MovieItem;

import ua.kpi.comsys.iv8106.secondary\_activities.MovieDetailsActivity;

public class MoviesAdapter extends RecyclerView.Adapter<MoviesAdapter.MovieViewHolder> {

private final Fragment context;

private final ArrayList<MovieItem> movies;

private final ArrayList<String> maintitle;

public MoviesAdapter(Fragment context, ArrayList<MovieItem> movies, ArrayList<String> maintitle) {

this.context=context;

this.movies=movies;

this.maintitle = maintitle;

}

public class MovieViewHolder extends RecyclerView.ViewHolder {

private ImageView image;

private TextView titleText;

private TextView yearText;

private TextView typeText;

private ImageView deleteButton;

public MovieViewHolder(View view) {

super(view);

// Define click listener for the ViewHolder's View

this.image = (ImageView) view.findViewById(R.id.poster);

this.titleText = (TextView) view.findViewById(R.id.title);

this.yearText = (TextView) view.findViewById(R.id.year);

this.typeText = (TextView) view.findViewById(R.id.type);

this.deleteButton = (ImageView) view.findViewById(R.id.deleteButton);

}

}

// Create new views (invoked by the layout manager)

@Override

public MovieViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

// Create a new view, which defines the UI of the list item

LayoutInflater inflater = LayoutInflater.from(parent.getContext());

View view = inflater.inflate(R.layout.movie, parent, false);

return new MovieViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull MoviesAdapter.MovieViewHolder holder, int position) {

int drawableResourceId = context.getResources().getIdentifier(

movies.get(position).getPoster().toLowerCase().replace(".jpg", ""),

"drawable", context.getContext().getPackageName());

if (!movies.get(position).isVisible()) {

holder.itemView.setVisibility(View.INVISIBLE);

holder.itemView.getLayoutParams().height = 0;

} else {

holder.itemView.setVisibility(View.VISIBLE);

holder.itemView.getLayoutParams().height = 650;

}

holder.titleText.setText(maintitle.get(position));

holder.yearText.setText(movies.get(position).getYear());

holder.typeText.setText(movies.get(position).getType());

if (drawableResourceId != 0) {

holder.image.setImageResource(drawableResourceId);

} else {

holder.image.setImageResource(R.drawable.ic\_action\_cancel);

}

holder.deleteButton.setOnTouchListener(new View.OnTouchListener() {

@Override

public boolean onTouch(View v, MotionEvent event) {

removeItem(holder.getAdapterPosition());

return true;

}

});

holder.itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(context.getContext(), MovieDetailsActivity.class);

if ((!movies.get(position).getImdbId().isEmpty()) &&

(!movies.get(position).getImdbId().equals("noid"))) {

intent.putExtra("id", movies.get(position).getImdbId());

context.startActivity(intent);

} else {

Toast toast = Toast.makeText(context.getContext(), "No ID", Toast.LENGTH\_SHORT);

toast.show();

}

}

});

}

@Override

public int getItemCount() {

return movies.size();

}

public void removeItem(int position) {

if (position == -1)

return;

this.movies.remove(position);

this.maintitle.remove(position);

notifyDataSetChanged();

}

}

**---MovieDetailsItem.java---**

package ua.kpi.comsys.iv8106.secondary\_activities;

import android.os.Bundle;

import android.widget.ImageView;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.google.gson.Gson;

import com.google.gson.reflect.TypeToken;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.lang.reflect.Type;

import ua.kpi.comsys.iv8106.R;

import ua.kpi.comsys.iv8106.model.MovieDetailsItem;

public class MovieDetailsActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_movie\_details);

ImageView descImage = (ImageView)findViewById(R.id.imageView);

TextView descText = (TextView) findViewById(R.id.textView2);

String file = getIntent().getStringExtra("id");

System.out.println(file);

Gson gson = new Gson();

Type movieDetailsElement = new TypeToken<MovieDetailsItem>() {}.getType();

MovieDetailsItem item = null;

try {

item = gson.fromJson(ReadTextFile(file), movieDetailsElement);

} catch (IOException e) {

e.printStackTrace();

}

int drawableResourceId = this.getResources().getIdentifier(

item.getPoster().toLowerCase().replace(".jpg", ""),

"drawable", this.getPackageName());

if (drawableResourceId != 0) {

descImage.setImageResource(drawableResourceId);

}

descText.setText(item.toString());

}

public String ReadTextFile(String name) throws IOException {

StringBuilder string = new StringBuilder();

String line = "";

InputStream is = this.getAssets().open(name + ".json");

BufferedReader reader = new BufferedReader(new InputStreamReader(is));

while (true) {

try {

if ((line = reader.readLine()) == null) break;

}

catch (IOException e) {

e.printStackTrace();

}

string.append(line);

}

is.close();

return string.toString();

}

}

**---AddMovieActivity.java---**

package ua.kpi.comsys.iv8106.secondary\_activities;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import ua.kpi.comsys.iv8106.R;

public class AddMovieActivity extends AppCompatActivity {

private String Title;

private String Year;

private String Type;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_add\_movie);

TextView title = (TextView) findViewById(R.id.textInputEditText2);

TextView year = (TextView) findViewById(R.id.textInputEditText4);

TextView type = (TextView) findViewById(R.id.textInputEditText5);

Button button = findViewById(R.id.buttonAddMovie);

button.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

String newMovie = "";

newMovie += "[" + "{\"Title\":\"" + title.getText().toString() + "\"," +

"\"Year\":\"" + year.getText().toString() + "\"," +

"\"imdbID\":\"\"," +

"\"Type\":\"" + type.getText().toString() + "\"," +

"\"Poster\":\"\"}" + "]";

Intent resultIntent = new Intent();

resultIntent.putExtra("movie", newMovie);

setResult(Activity.RESULT\_OK, resultIntent);

finish();

}

});

}

}

**---MovieDetailsItem.java---**

package ua.kpi.comsys.iv8106.model;

public class MovieDetailsItem {

private String Title;

private String Rated;

private String Released;

private String Runtime;

private String Genre;

private String Director;

private String Writer;

private String Actors;

private String Plot;

private String Language;

private String Country;

private String Awards;

private String Year;

private String imdbRating;

private String imdbVotes;

private String imdbID;

private String Type;

private String Poster;

private String Production;

public String getTitle() {

return Title;

}

public String getRated() {

return Rated;

}

public String getReleased() {

return Released;

}

public String getRuntime() {

return Runtime;

}

public String getGenre() {

return Genre;

}

public String getDirector() {

return Director;

}

public String getWriter() {

return Writer;

}

public String getActors() {

return Actors;

}

public String getPlot() {

return Plot;

}

public String getLanguage() {

return Language;

}

public String getCountry() {

return Country;

}

public String getAwards() {

return Awards;

}

public String getYear() {

return Year;

}

public String getImdbRating() {

return imdbRating;

}

public String getImdbVotes() {

return imdbVotes;

}

public String getImdbID() {

return imdbID;

}

public String getType() {

return Type;

}

public String getPoster() {

return Poster;

}

public String getProduction() {

return Production;

}

public String toString() {

return "Title: " + Title +

"\nYear: " + Year +

"\nGenre: " + Genre +

"\n\nDirector: " + Director +

"\nActors: " + Actors +

"\n\nCountry: " + Country +

"\nLanguage: " + Language +

"\nProduction: " + Production +

"\nReleased: " + Released +

"\nRuntime: " + Runtime +

"\n\nAwards: " + Awards +

"\nRating: " + Rated +

"\nPlot: " + Plot;

}

}

**---MovieItem.java---**

package ua.kpi.comsys.iv8106.model;

public class MovieItem {

private boolean isVisible = true;

private String Title;

private String Year;

private String Type;

private String Poster;

private String imdbID;

public String getTitle() {

return Title;

}

public String getYear() {

return Year;

}

public String getType() {

return Type;

}

public String getPoster() {

return Poster;

}

public String getImdbId() { return imdbID; }

public boolean isVisible() {

return isVisible;

}

public void setVisible(boolean visible) {

isVisible = visible;

}

}

**---MoviesFragment.java---**

package ua.kpi.comsys.iv8106.ui.movies;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.SearchView;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.fragment.app.Fragment;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import com.google.gson.Gson;

import com.google.gson.reflect.TypeToken;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.lang.reflect.Type;

import java.util.ArrayList;

import ua.kpi.comsys.iv8106.R;

import ua.kpi.comsys.iv8106.adapters.MoviesAdapter;

import ua.kpi.comsys.iv8106.model.MovieItem;

import ua.kpi.comsys.iv8106.secondary\_activities.AddMovieActivity;

public class MoviesFragment extends Fragment {

private static final int REQUEST\_READ\_EXTERNAL\_STORAGE = 1;

View root;

Type listOfMoviesItemsType = new TypeToken<ArrayList<MovieItem>>() {}.getType();

ArrayList<MovieItem> movie\_list = new ArrayList<>();

ArrayList<String> main\_title = new ArrayList<>();

public View onCreateView(@NonNull LayoutInflater inflater,

ViewGroup container, Bundle savedInstanceState) {

root = inflater.inflate(R.layout.fragment\_movies, container, false);

// int rCheck = ContextCompat.checkSelfPermission(getActivity(), Manifest.permission.READ\_EXTERNAL\_STORAGE);

// if (rCheck != PackageManager.PERMISSION\_GRANTED) {

// ActivityCompat.requestPermissions(getActivity(),

// new String[]{Manifest.permission.READ\_EXTERNAL\_STORAGE}, REQUEST\_READ\_EXTERNAL\_STORAGE);

//

// }

String path = "MoviesList.json";

//File down = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY\_DOWNLOADS);

TextView nothingFound = root.findViewById(R.id.nothingFound);

nothingFound.setVisibility(View.INVISIBLE);

Gson gson = new Gson();

// ContextWrapper cont = new ContextWrapper(getActivity());

// System.out.println(cont.getFilesDir());

try {

this.movie\_list = gson.fromJson(ReadTextFile(path), listOfMoviesItemsType);

} catch (IOException e) {

e.printStackTrace();

}

for (MovieItem movie: movie\_list) {

main\_title.add(movie.getTitle());

}

RecyclerView list = root.findViewById(R.id.noMoviesMessage);

MoviesAdapter adapter\_movie = new MoviesAdapter(this, movie\_list, main\_title);

list.setAdapter(adapter\_movie);

list.setLayoutManager(new LinearLayoutManager(getActivity()));

SearchView searchBar = (SearchView) root.findViewById(R.id.searchBar);

searchBar.setOnQueryTextListener(new SearchView.OnQueryTextListener() {

@Override

public boolean onQueryTextSubmit(String query) {

onQueryTextChange(query);

return false;

}

@Override

public boolean onQueryTextChange(String newText) {

int hiddenItemsCount = 0;

for (MovieItem m : movie\_list){

if (newText.equals("")) {

m.setVisible(true);

nothingFound.setVisibility(View.INVISIBLE);

continue;

}

if (!m.getTitle().startsWith(newText)) {

m.setVisible(false);

hiddenItemsCount++;

} else {

m.setVisible(true);

}

}

if (hiddenItemsCount == movie\_list.size()) {

nothingFound.setVisibility(View.VISIBLE);

}

adapter\_movie.notifyDataSetChanged();

return true;

}

});

Button addMovieButton = (Button) root.findViewById(R.id.addItem);

addMovieButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(getActivity(), AddMovieActivity.class);

startActivityForResult(intent, 1);

adapter\_movie.notifyDataSetChanged();

}

});

return root;

}

private void updateJSON(String newData) {

Gson gson = new Gson();

Type listOfMoviesItemsType = new TypeToken<ArrayList<MovieItem>>() {}.getType();

ArrayList<MovieItem> new\_movie = gson.fromJson(newData, listOfMoviesItemsType);

this.movie\_list.addAll(gson.fromJson(newData, listOfMoviesItemsType));

this.main\_title.add(new\_movie.get(0).getTitle());

}

@Override

public void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == 1) {

if (resultCode == Activity.RESULT\_OK) {

String returnValue = data.getStringExtra("movie");

updateJSON(returnValue);

}

}

}

public String ReadTextFile(String name) throws IOException {

StringBuilder string = new StringBuilder();

String line = "";

InputStream is = getContext().getAssets().open(name);

BufferedReader reader = new BufferedReader(new InputStreamReader(is));

while (true) {

try {

if ((line = reader.readLine()) == null) break;

}

catch (IOException e) {

e.printStackTrace();

}

string.append(line);

}

is.close();

return string.toString();

}

}

1. **Висновок:**

Розширено функціонал попередньої лабораторної роботи, також було зчитано новий JSONфайл по ідентифікатору з основного. Було додано можливість додавати елемент в список.