



MATURITNÍ PRÁCE

Android aplikace - MovieList

Šalomon Adam

Obor: EZI
Třída: ZI4B

Školní rok: 2016/2017



ZADÁNÍ MATURITNÍ PRÁCE

školní rok	jméno žáka a vlastnoruční podpis	třída	obor
2016/2017	Adam Šalomon <i>Šalomon</i>	ZI4B	EZI

název práce
Android aplikace filmové databáze

zaměření práce
informatika - software

konkretizace zadání:

Na platformě Android vytvořte aplikaci pro získávání informací o filmech z internetu. Vyberte vhodné vývojové prostředí, navrhnete grafiku a uživatelské rozhraní. Informace o filmech bude možno zobrazit detailní informace. Aplikace nabídne uživateli vytvoření vlastního seznamu filmů vhodných pro zhlédnutí – podle vlastní volby a podle preferencí (klíčových slov), které uživatel zadá jako oblíbená (vyskytující se v popisu filmů). Aplikace kromě toho projde TV programy a k filmům v seznamu (watchlistu) vybere televizní stanice, které film v brzké době vysílají, navrhne uživateli výběr TV programu a zařadí do kalendáře.

jednotlivé cíle práce:

- aplikace pro android
- aplikace bere nové filmy a informace o nich z IMDb - hodnocení, žánr, shrnutá zápletka, herci atd...
- uživatel si může dát filmy do "watchlist" - filmy na které se chce podívat
- aplikace vloží filmy do vlastní databáze
- může evidovat filmy které uživatel již viděl nebo které se mu líbily
- aplikace doporučí filmy podobné filmům které si uživatel oblíbil (klíčová slova)
- aplikace vyhledá film v TV programu a navrhne sledování podle času a stanice
- uživatelské rozhraní
- uživatelský manuál

Další formální náležitosti zadání práce podle platného znění vyhlášky MŠMT č. 177/2009 Sb. jsou k dispozici na www adrese: <https://www.roznovskastredni.cz/maturitni-prace>

V Rožnově pod Radhoštěm dne: 17.10.2016

Lukáš Hapl
vedoucí práce

ABSTRAKT

Práce je zaměřena na problematiku tvorby Android aplikací, přesněji tvorbu filmové databáze. Práce obsahuje vysvětlení některých pro práci důležitých pojmů. Je zde popsán použitý software a postup při tvorbě aplikace od jejího grafického návrhu, přes ukázky důležitých částí kódu, až po uživatelskou příručku, v níž je vysvětleno ovládání aplikace.

Klíčová slova: aplikace, Android, databáze

Rád bych poděkoval panu ing. Haplovi, který byl mým odborným konzultantem při této práci. A také všem spolužákům, kteří byly mými testery ve všech verzích aplikace a hledali všemožné chyby.

Prohlašuji, že odevzdaná verze dokumentace maturitní práce a verze elektronická, nahraná do systému MATPRAC, jsou totožné. Při zpracování jsem vycházel z informačních zdrojů uvedených v seznamu na konci dokumentace a také prohlašuji, že je tato práce původní.

V Rožnově pod Radhoštěm

Salomon

podpis žáka

OBSAH

ÚVOD.....	8
1 CÍLE PRÁCE	10
1.1 Tvorba Android aplikace	10
1.2 Vytvoření systému získávání potřebných informací	10
1.3 Vytvoření intuitivního uživatelského prostředí	10
1.4 Tvorba „Seenlist“ a „Watchlist“ záložek.....	10
1.5 Potenciál aplikace	10
2 VÝBĚR TECHNOLOGIÍ PRO ŘEŠENÍ	11
2.1 Použité jazyky	11
2.1.1 Java	11
2.1.2 XML	11
2.2 Použitý software	11
2.2.1 Android Studio	11
2.2.2 Adobe Photoshop	12
2.2.3 Genymotion	13
3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY	15
3.1 Základní návrh aplikace	15
3.1.1 Návrh grafického rozhraní	15
3.1.2 Rozložení obsahu do více aktivit.....	16
3.1.3 Adresářová struktura	16
3.2 Tvorba struktury kódu.....	18
3.2.1 Základní návrh kódu	18
3.2.2 Stahování dat.....	18
3.2.3 Práce se soubory.....	19
3.2.4 Způsob vykreslování jednotlivých buněk.....	20
4 PŘÍRUČKA UŽIVATELE.....	22
4.1 Požadavky pro správný chod a práci aplikace.....	22
4.1.1 Základní požadavky pro spuštění	22
4.1.2 Nutná oprávnění	22
4.2 Používání aplikace	22
4.2.1 První spuštění aplikace.....	22
4.2.2 Orientace v aplikaci	23

4.3	Detail jednotlivých prvků aplikace	23
4.3.1	Tlačítka pro výběr seznamu.....	23
4.3.2	Vyhledávací okénko	24
4.3.3	Levé vysunovací menu	24
4.3.4	Ovládací prvky jednotlivých buněk	25
4.3.5	Vytvoření notifikace na datum vydání filmu	25
	ZÁVĚR	26
	SEZNAM POUŽITÉ LITERATURY	27
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	28
	SEZNAM OBRÁZKŮ	29
	SEZNAM PŘÍLOH.....	30

ÚVOD

Vývoj aplikací pro mobilní telefony (Android, iOS) je v dnešní době a pravděpodobně i v budoucnu velice atraktivní a lukrativní oblast programování, která přináší spoustu nových, na PC těžko uskutečnitelných, nápadů a myšlenek. Přívětivost této oblasti programování jen dále prohlubuje široká komunita, která neváhá poradit při jakémkoliv problému.

K dokončení tohoto projektu bylo zapotřebí nejen hlubšího pochopení jazyku Java a funkcí Android systémů, ale i obeznámení se s možnostmi získávání potřebných informací (databází) z internetu.

Android aplikace se skládá z XML šablon, které tvoří uživatelské grafické rozhraní (GUI), dále ze samotných Java souborů, jež obsahují logickou část celého projektu, a z Android manifestu, ve kterém jsou definována práva a požadavky na systém telefonu. Pomocí kompilátoru Gradle se všechny soubory zpracují a „zabalí“ do *.apk souboru, se kterými již Android smartphone nebo tablet dokáže pracovat.

1. TEORETICKÁ ČÁST

1 CÍLE PRÁCE

1.1 Tvorba Android aplikace

Cílem práce je vytvořit Android aplikaci, která by uživateli poskytovala možnost vytvořit si vlastní knihovnu filmů, na něž by se chtěl podívat nebo které již viděl, společně s informacemi s nimi spjatými.

1.2 Vytvoření systému získávání potřebných informací

Aplikace bude veškeré informace získávat z webové stránky www.themoviedb.org v podobě JSON databáze. Tyto informace si zařízení uloží do své mezi-paměti k pozdějšímu využití.

1.3 Vytvoření intuitivního uživatelského prostředí

Dalším cílem této práce je vytvoření grafického návrhu v Adobe Photoshop, který by byl uživatelsky přívětivý a intuitivní, dále sepsání XML kódu, jež by onen návrh následoval.

1.4 Tvorba „Seenlist“ a „Watchlist“ záložek

Pokud si uživatel uloží některý z filmů do seznamu snímků, které chce vidět nebo již viděl, aplikace uloží informace o titulu do své interní paměti a uživatel má k němu přístup i bez internetového připojení.

1.5 Potenciál aplikace

V budoucnosti by mohla aplikace uživateli nabídnout možnosti úpravy grafického rozhraní či zálohování vytvořených seznamů na internetové servery.

Práce si klade za dlouhodobý cíl vyvinout aplikaci tak, aby se rozvíjela na základě uživatelské zpětné vazby a aby se její databáze rozšířila o další kinematografické žánry.

2 VÝBĚR TECHNOLOGIÍ PRO ŘEŠENÍ

2.1 Použité jazyky

2.1.1 Java

- Je multiplatformní programovací jazyk, v němž je napsána většina kódu.
- Použitá verze: Java ME JDK 7

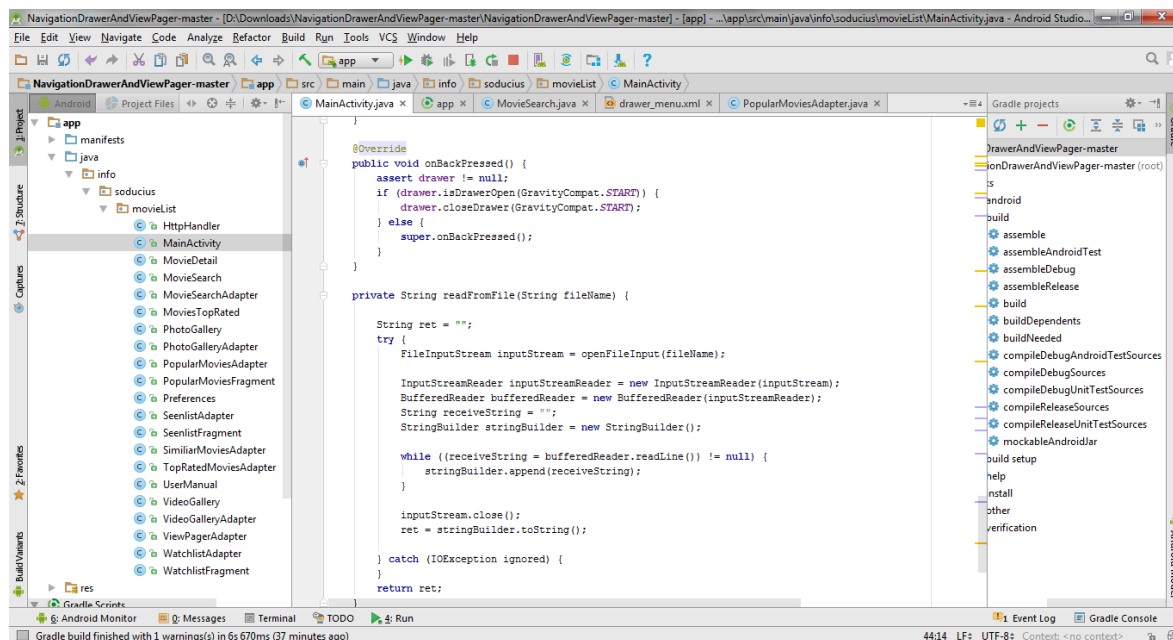
2.1.2 XML

- Je obecný značkovací jazyk, ve kterém je vytvořeno uživatelské rozhraní aplikace.
- Použitá verze: XML 1.1

2.2 Použitý software

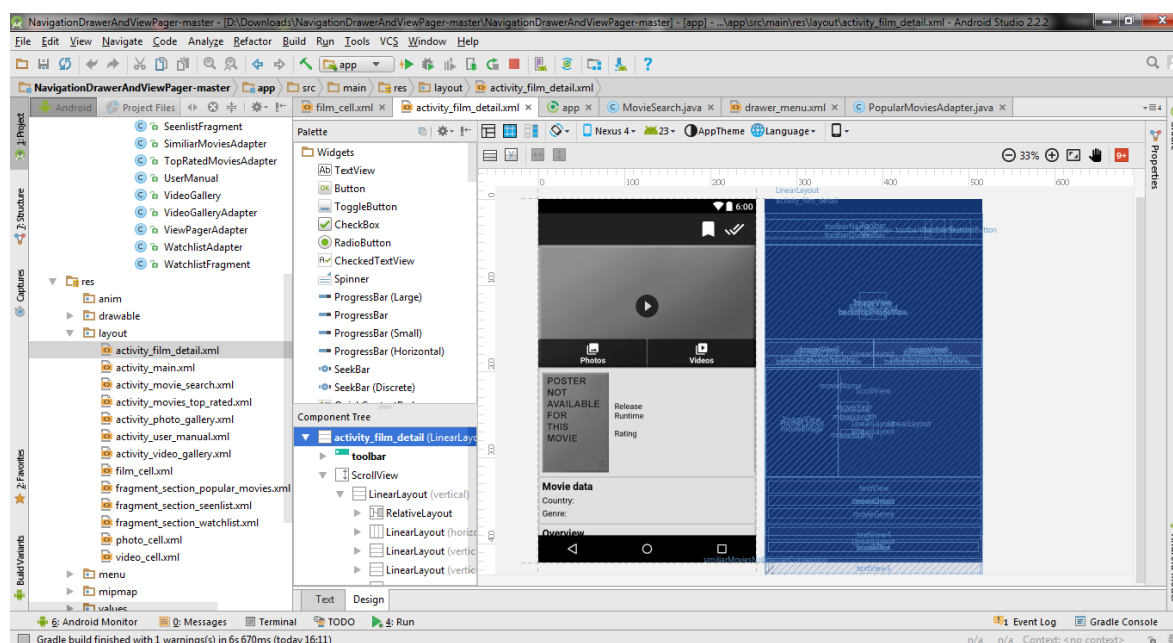
2.2.1 Android Studio

- Android studio je oficiální program od Google pro vývoj Android aplikací.
- Použitá verze: 2.2



Obr. 2.1 Android Studio

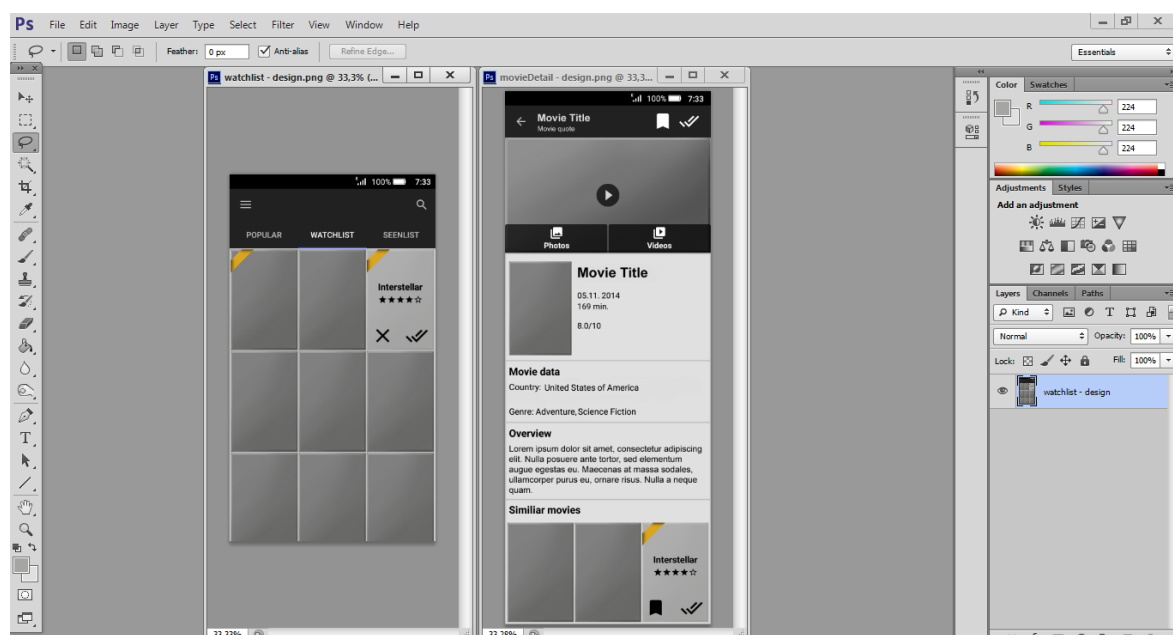
Díky zaměření Android Studio na vývoj Android aplikací (oproti např. Eclipse) má program přehledné rozhraní, ve kterém uživatel vždy najde, co hledá. Mezi další výhody patří povedený XML design editor, v němž je manipulace s jednotlivými prvky značně zjednodušena.



Obr. 2.2 XML Grafický editor Android studia

2.2.2 Adobe Photoshop

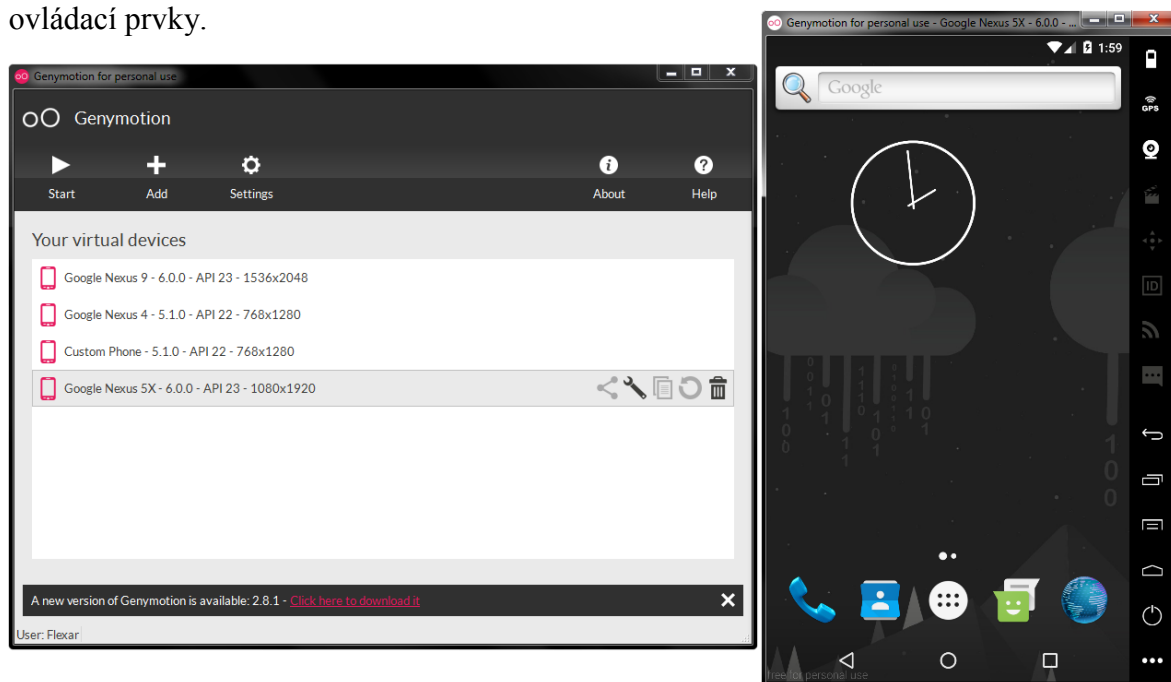
Adobe Photoshop je bitmapový grafický editor, který se používá téměř všude tam, kde se manipuluje s rastrovou grafikou. Photoshop byl použit při návrhu GUI.



Obr. 2.3 Adobe Photoshop CS6

2.2.3 Genymotion

Počítačový program emulující nepřehledné množství Android zařízení. Genymotion byl využit při testování aplikace na různých zařízeních. Jeho výhodou oproti obyčejnému emulátoru, který poskytuje Android Studio, je vyšší rychlost, lepší optimalizace a jednodušší ovládací prvky.



Obr. 2.4 Emulátor Android zařízení Genymotion

2. PRAKTICKÁ ČÁST

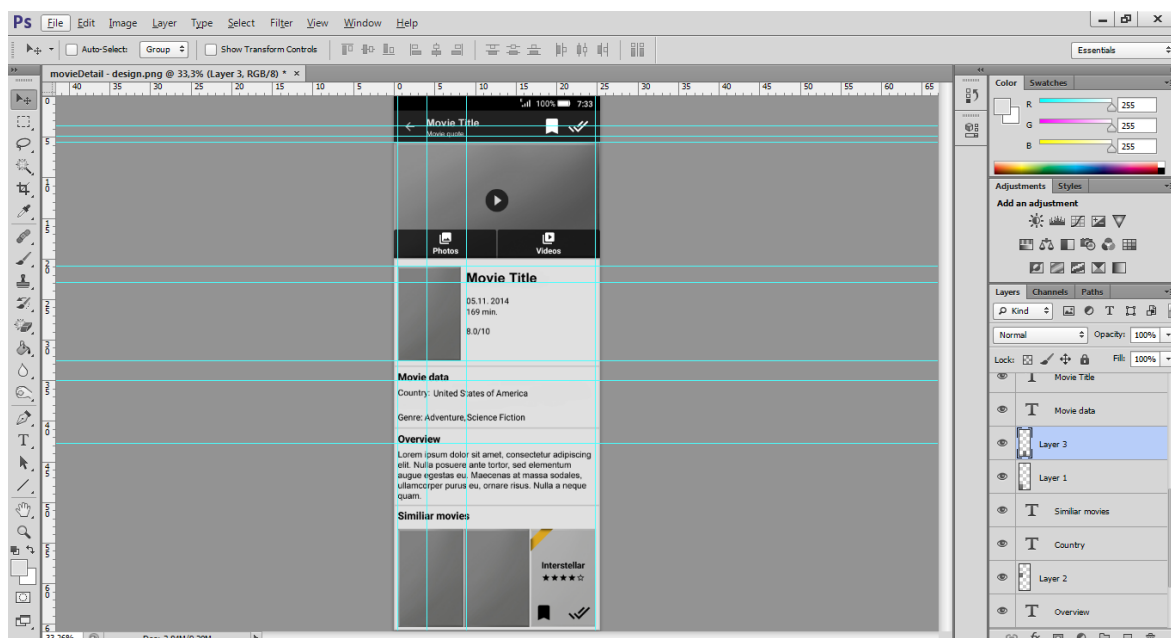
3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

3.1 Základní návrh aplikace

3.1.1 Návrh grafického rozhraní

Návrh na vzhled aplikace byl inspirován standardy Material designu, jenž preferuje jednoduchost a praktičnost před zbytečnými kosmetickými prvky.

Návrh byl uskutečněn v programu Adobe Photoshop CS6. Při vytváření XML kódu se však našly problémy, v podobě těžko realizovatelných grafických prvků, které musely být pozměněny.



Obr. 3.1 Grafický návrh v Adobe Photoshop

3.1.2 Rozložení obsahu do více aktivit

Z důvodu složitosti práce bylo nutné kód rozdělit do několika souborů, přesněji do 21 Java a 23 XML souborů, které obstarávají běh celé aplikace (započítány byly pouze ručně psané kódy). Program je dále rozdělen na tzv. aktivity, což jsou v podstatě malé „pod-programy“. [1] Seznam aktivit:

MainActivity – Je domovská obrazovka, jenž zobrazuje populární filmy, Watchlist, Seenlist a vyhledávací okénko.

MovieDetail – Zobrazuje detailní informace o vybraném filmu.

MovieSearch – Ukazuje výsledky vyhledávání filmů.

TopRatedMovies – Obsahuje filmy seřazené podle jejich hodnocení (sestupně).

MoviePhotoGallery – Obsahuje fotky a plakáty spojené s filmem.

MovieVideoGallery – Obsahuje videa spojená s filmem.

UserManual – Obsahuje návod, jak aplikaci používat.

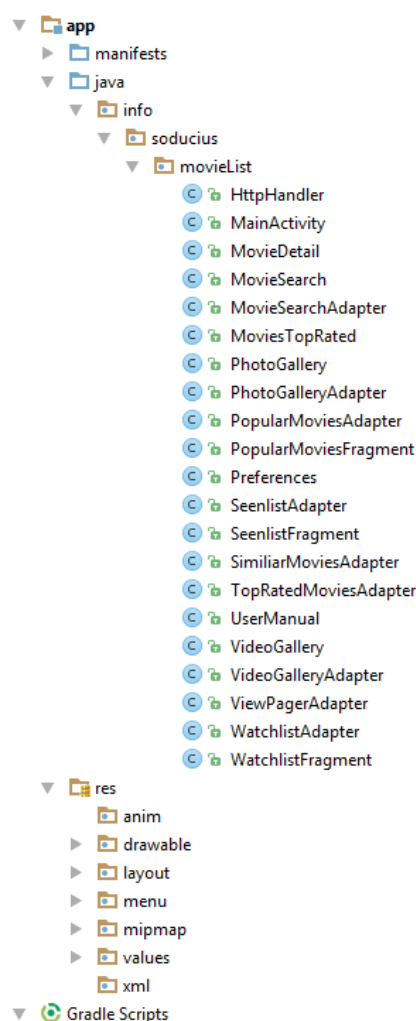
```
<activity
    android:name="info.soducius.mvieList.MainActivity"
    android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name="info.soducius.mvieList.MovieDetail"
    android:screenOrientation="portrait" />
<activity
    android:name="info.soducius.mvieList.MovieSearch"
    android:screenOrientation="portrait"
    android:theme="@style/AppThemeNoAnimation" />
<activity
    android:name="info.soducius.mvieList.UserManual"
    android:screenOrientation="portrait" />
<activity
    android:name="info.soducius.mvieList.PhotoGallery"
    android:screenOrientation="portrait" />
<activity
    android:name="info.soducius.mvieList.VideoGallery"
    android:screenOrientation="portrait" />
<activity
    android:name="info.soducius.mvieList.MoviesTopRated"
    android:screenOrientation="portrait" />
```

Obr. 3.2 Rozložení aktivit v AndroidManifest.xml

3.1.3 Adresářová struktura

Android studio při vytvoření projektu samo vytvoří jednoduchou adresářovou strukturu, kterou programátor nemá důvod jakkoliv měnit. Při potřebě se nové soubory vkládají do předem vytvořených složek nebo do nových složek s předem zadaným jménem.

- manifest – obsahuje AndroidManifest.xml. Tento soubor musí být v základní složce každé aplikace. Obsah tohoto souboru stanovuje požadavky na systém ke správnému chodu aplikace.
- java - je složka s *.java soubory, které v sobě mají veškerou logickou část aplikace.
- res – obsahuje XML soubory (soubory tvořící GUI), aplikací využívané obrázky, soubory obsahující informace o stylech jednotlivých grafických prvků a další podpůrné soubory.
- build – obsahuje automaticky generované soubory potřebné ke správné instalaci a chodu aplikace.



Obr. 3.3 Adresářová struktura

3.2 Tvorba struktury kódu

3.2.1 Základní návrh kódu

Všechn *java kód by bylo možno rozdělit do tří skupin. První by byly aktivity, druhé adaptéry, třetí podpůrné funkce. Aktivity načítají GUI prvky, přidělují adaptéry elementům, které je potřebují, a stahují na pozadí data potřebná k zobrazení filmů. Tato data se předají adaptérům starajícím se o vykreslení informací o filmech uvnitř tzv. GridView (dvojdimenzionální mřížka skládající se z řádků a sloupců, v nichž jsou jednotlivé „buňky“ filmů). Do třetí skupiny patří dvě funkce. První se využívá při stahování dat z internetu a druhá má za úkol pamatovat si určité informace i po vypnutí aplikace či telefonu. Zapisování do vlastní databáze se realizuje pomocí jiných funkcí (viz. Kapitola 3.2.3).

3.2.2 Stahování dat

```
private class PopularMoviesJSONParser extends AsyncTask<String, Void, String> {
    //Asynchronní funkce, která je zpracována na nezávislém vlákně
    @Override
    protected String doInBackground(String... params) {
        //Volání vytvořené funkce HttpHandler
        HttpHandler sh = new HttpHandler();
        //základní podoba dotazu (tento API patří pouze mně)
        String basicLink = "https://api.themoviedb.org/3/movie/popular?api_key=95148fc363ff244cab388010dfbfa949&language=en-US&page=";
        //params[0] obsahuje číslo stránky, jenž je potřeba načíst
        String jsonStr = sh.makeServiceCall(basicLink+params[0]);
        try {
            //vytvoření JSON objektu, který obsahuje seznam získaných populárních filmů
            JSONObject wholeJSON = new JSONObject(jsonStr);
            //získání pole "results" - výsledky dotazu
            JSONArray oneArray = wholeJSON.getJSONArray("results");
            //čtení a následně přidání filmů do vlastního seznamu populárních filmů
            for (int i=0; i<oneArray.length(); i++){
                JSONObject oneMovie = oneArray.getJSONObject(i);
                popularMoviesArrayList.add(String.valueOf(oneMovie));
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return jsonStr;
    }
    @Override
    protected void onPostExecute (String i) {
        //Upozornění adaptéru, jenž se stará o vykreslování populárních filmů, že jeho data se změnila
        PopularMoviesGridViewAdapter.notifyDataSetChanged();
    }
}
```

Obr 3.4 Zpracování dotazu na populární filmy

Stahování všech dat z databáze stránky www.themoviedb.org bylo možné díky osobnímu API klíči, jenž byl poskytnut po odeslání žádosti na podporu stránky. Tento klíč však má svá omezení. Jelikož jde o bezplatnou službu, má klíč maximální počet „dotazů“ na server každých 10 vteřin. Po získání žádaných filmů se tituly uloží do příslušných seznamů. [2]

3.2.3 Práce se soubory

V aplikaci bylo nutné vymyslet systém ukládání Watchlist a Seenlist seznamu do paměti telefonu, aby k nim měl uživatel přístup i bez připojení k internetu. Toho bylo dosaženo zapisováním databáze do *.txt souborů (watchlist.txt a seenlist.txt). Tyto soubory se při startu aplikace přečtou a jejich obsah se zobrazí.

```
private void writeToWatchlistOrSeenlistFile(String fileName, String data) {
    try {
        OutputStreamWriter outputStreamWriter = new OutputStreamWriter(context.openFileOutput(fileName, Context.MODE_PRIVATE));
        outputStreamWriter.write(data);
        outputStreamWriter.close();
    }
    catch (IOException e) {
        Log.e("Exception", "File write failed: " + e.toString());
    }
}

private String readFromFile(String fileName) {
    String ret = "";
    try {
        FileInputStream inputStream = context.openFileInput(fileName);

        InputStreamReader inputStreamReader = new InputStreamReader(inputStream);
        BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
        String receiveString = "";
        StringBuilder stringBuilder = new StringBuilder();

        while ((receiveString = bufferedReader.readLine()) != null) {
            stringBuilder.append(receiveString);
        }
        inputStream.close();
        ret = stringBuilder.toString();
    } catch (IOException ignored) {
    }
    return ret;
}
```

Obr. 3.5 Funkce pro zapisování a čtení souborů

Mezi další operace se soubory patří přidávání filmů do Watchlistu a Seenlistu. Tato operace musela být řešena asynchronně pro zachování plynulosti aplikace. Při přidávání je potřeba zkontrolovat zda již film seznam neobsahuje. Pokud ano snímek se do seznamu nepřidá. Databáze má obyčejnou JSON strukturu – celá databáze je JSONObject (objekt ohraničený složenými závorkami), v níž jsou JSONArray (pole ohraničené hranatými závorkami), kdy každý JSONArray obsahuje právě jeden film. Každý JSONArray je pojmenován podle své pozice v databázi (např. “0“, “1“ atd.) [3]

```

private class AddToSeenOrWatchListAsyncTask extends AsyncTask<String, Void, String> {
    String fileName;
    String linkToData;
    AddToSeenOrWatchListAsyncTask(String fileName, String linkToData) {
        this.fileName = fileName;
        this.linkToData = linkToData;
    }
    @Override
    protected String doInBackground(String... params) {
        //přečtení původního souboru
        String fileString = readFromFile(fileName);
        HttpHandler sh = new HttpHandler();
        //provedení dotazu - adresa dotazu je parametr při volání této funkce
        String jsonStr = sh.makeServiceCall(linkToData);
        try {
            JSONObject wholeJSON = new JSONObject();
            //Ověření zda seznam již obsahuje přidávaný film
            Boolean doesContain = false;
            if (!fileString.equals("")) {
                wholeJSON = new JSONObject(fileString);
                for (int i=0; i<wholeJSON.length(); i++){
                    JSONObject JSONStr = new JSONObject(jsonStr);
                    if (String.valueOf(JSONStr.getString("title")).equals(wholeJSON.getJSONArray(String.valueOf(i)).getJSONObject(0).getString("title"))){
                        doesContain = true;
                    }
                }
            }
            if (!doesContain) {
                //pokud jej neobsahuje
                JSONObject oneMovieObject = new JSONObject(jsonStr);

                JSONArray oneMovieArray = new JSONArray();
                oneMovieArray.put(oneMovieObject);
                //film se přidá jako další položka do databáze
                wholeJSON.put(String.valueOf(wholeJSON.length()), oneMovieArray);
                fileString = wholeJSON.toString();
                //přidání do seznamů pro adaptéry
                if (fileName.equals("watchlist.txt")) {
                    MainActivity.watchlistMovies.add(String.valueOf(oneMovieObject));
                } else if (fileName.equals("seenlist.txt")) {
                    MainActivity.seenlistMovies.add(String.valueOf(oneMovieObject));
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        //přepsání nového seznamu starým
        writeToWatchlistOrSeenlistFile(fileName, fileString);
        return fileName;
    }
    @Override
    protected void onPostExecute(String fileName) {
        super.onPostExecute(fileName);
        //po dokončení celé funkce se upozorní adaptéry o změně v jejich datech
        if (fileName.equals("watchlist.txt")) {WatchlistFragment.WatchlistGridViewAdapter.notifyDataSetChanged();}
        if (fileName.equals("seenlist.txt")) {WatchlistFragment.SeenlistGridViewAdapter.notifyDataSetChanged();}
    }
}

```

Obr. 3.6 Funkce na přidávání filmů do seznamů

3.2.4 Způsob vykreslování jednotlivých buněk

Aktivita MainActivity se dělí na 3 „mini-aktivity“ tzv. fragmenty. Nad fragmenty se dá přemýšlet jako nad modulárními částmi aktivity. Fragment má vlastní životní cykly a dotykové vstupy. Životní cyklus fragmentu je však přímo ovlivňován životním cyklem aktivity, v níž se nachází. Fragmenty se dají do aktivity přidávat a odebírat za chodu aktivity. Oddíly „Popular“, „Watchlist“ a „Seenlist“ jsou každé v samostatném fragmentu. Z důvodu plynulosti se vždy načtou sousední fragmenty tak, že uživatel nepozná, že se jedná více souborů. Každý fragment má svůj vlastní adaptér, který dostává informace o tom, co se má uživateli zobrazit. Uvnitř adaptéru se jednotlivé buňky vykreslují postupně jedna po druhé. Každá má své ID (pozici v seznamu), tudíž při vytváření dostane každá buňka informace odpovídající jejímu umístění v mřížce. [4]

```

public WatchListAdapter(Activity mainActivity, ArrayList<String> watchlistMovies) {
    context = mainActivity; //nastaví kontext
    watchlistMoviesArrayList = watchlistMovies; //předá list z argumentu do listu adaptéru
    inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
}

@Override
public int getCount() {return watchlistMoviesArrayList.size();} //funkce getCount vrací celkový počet prvků v mřížce
@Override
public Object getItem(int position) {return position;} //funkce getItem vrací pozici tázaného prvku
@Override
public long getItemId(int position) {return position;}
@Override
public View getView(final int position, View convertView, final ViewGroup parent) {
    final View cellView;
    cellView = inflater.inflate(R.layout.film_cell, null);
    //inicializování všech potřebných prvků v buňce mřížce
    cellView.findViewById(R.id.bookMarkButton).setVisibility(View.GONE);
    cellView.findViewById(R.id.clearBookmarkButton).setVisibility(View.VISIBLE);
    cellView.findViewById(R.id.popularSeenButton).setVisibility(View.GONE);
    cellView.findViewById(R.id.watchListSeenButton).setVisibility(View.VISIBLE);
    ratingFirstStar = (ImageView) cellView.findViewById(R.id.ratingFirstStar);
    ratingSecondStar = (ImageView) cellView.findViewById(R.id.ratingSecondStar);
    ratingThirdStar = (ImageView) cellView.findViewById(R.id.ratingThirdStar);
    ratingFourthStar = (ImageView) cellView.findViewById(R.id.ratingFourthStar);
    ratingFifthStar = (ImageView) cellView.findViewById(R.id.ratingFifthStar);
    //pokud seznam uložených filmů do watchlist není prázdný
    if (!watchlistMoviesArrayList.isEmpty()) {
        String jsonString = readFromFile("watchlist.txt"); //přečtení offline databáze
        try {
            wholeJSON = new JSONObject(jsonString);
            oneMovieJSONArray = wholeJSON.getJSONArray(String.valueOf(position)); //získání filmu podle pozice
            oneMovieJSONObject = oneMovieJSONArray.getJSONObject(0);
        } catch (JSONException e) {
            e.printStackTrace();
        }
        Double rating = 0d;
        try {
            rating = oneMovieJSONObject.getDouble("vote_average"); //získání hodnocení z databáze
        } catch (JSONException e) {
            e.printStackTrace();
        }
        setRatingStars(rating); //nastavení hodnocení (malé hvězdičky u filmu)
        goldStripe = (ImageView) cellView.findViewById(R.id.goldStripe);
        if (rating > 7.5) {
            goldStripe.setVisibility(View.VISIBLE); //pokud má film lepší hodnocení než 7.5 dostane zlatou stuhu
        }

        TextView tv;
        ImageView img;
        tv = (TextView) cellView.findViewById(R.id.textView1); //inicializování TextView - pole pro název filmu
        img = (ImageView) cellView.findViewById(R.id.filmImage); //inicializace ImageView - pole pro obrázek (plakát)

        String url = null;
        try {
            url = "https://image.tmdb.org/t/p/w185" + oneMovieJSONObject.getString("poster_path"); //získání URL obrázku
        } catch (JSONException e) {
            e.printStackTrace();
        }
        Picasso.with(context). //knihovna Picasso, která se stará o načítání obrázků
            load(url).
            fit().
            placeholder(R.drawable.placeholder_film_image). //případná náhrada za obrázek při nepovedeném načtení
            into(img); //cíl do kterého se má obrázek načíst

        try {
            tv.setText(oneMovieJSONObject.getString("title")); //získání a nastavení textu do pole tv (název filmu)
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

```

Obr. 3.7 Vykreslování jednotlivých buněk

Části kódu, jenž se nachází uvnitř tzv. funkce try – catch. Musely být uvnitř této funkce z důvodu, že by databáze, na kterou příkazy volají, byla prázdná nebo vůbec neexistovala. V takovém případě by se celá aplikace vyplala a vypsalala chybovou hlášku.

4 PŘÍRUČKA UŽIVATELE

4.1 Požadavky pro správný chod a práci aplikace

4.1.1 Základní požadavky pro spuštění

Aplikace obsahuje některé funkce a příkazy, které nebyly dostupné v dřívějších verzích Androidu. Z tohoto důvodu je minimální verze Android 4.4 Kitkat a doporučená 5.0 Lollipop nebo vyšší.

Aplikace by se měla sama přizpůsobit obrazovce uživatele, avšak doporučené rozlišení je 1024x768 a vyšší. Toto rozlišení je již v dnešní době minimálním standardem téměř každého nového telefonu. Co se týče hardwarových nároků, je aplikace mírně náročná na paměť a procesor, kvůli velkému počtu dílčích prvků, které se musí najednou načítat. Procesor se dvěma jádry o frekvencích alespoň 1GHz by neměl mít s aplikací větší problémy. Množství operační paměti potřebné pro nepřerušovaný pohyb v aplikaci je 1GB a více.

4.1.2 Nutná oprávnění

Při instalaci se aplikace ptá uživatele, zdali jí poskytne určitá oprávnění potřebná, ke správnému fungování kódu. Aplikace potřebuje oprávnění k zjištění stavu sítě, přístupu na internet, zapisování na externí paměť telefonu a k čtení a zápisu do kalendáře. Pokud uživatel tato oprávnění nepotvrdí, aplikace se nenainstaluje.

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.READ_CALENDAR" />  
<uses-permission android:name="android.permission.WRITE_CALENDAR" />
```

Obr. 4.1 Vyžadovaná oprávnění

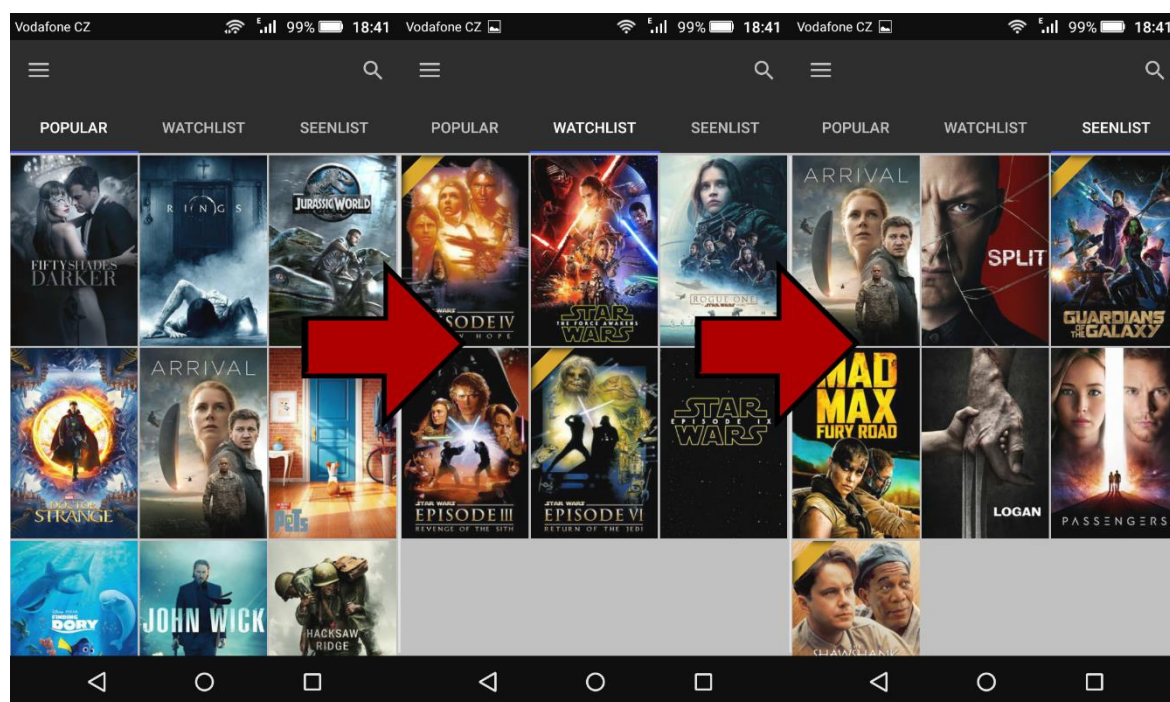
4.2 Používání aplikace

4.2.1 První spuštění aplikace

Při prvním spuštění je uživatel uvítán na obrazovce Aktivitý UserManual, na které mu je vysvětleno co je cílem a důvodem vzniku této aplikace. Tam má možnost projít si malou ukázkou, jak aplikaci používat, nebo vše přeskočit a dostat přímo na domovskou obrazovku.

4.2.2 Orientace v aplikaci

Uživatel přepíná mezi jednotlivými obrazovkami posouváním prstu nebo dotykem na název listu. Pro přepínání se dá také využít levého bočního menu, které tyto obrazovky společně s aktivitami UserManual a TopRatedMovies (seznam nejlépe hodnocených filmů) obsahuje. Domovskou obrazovku představuje obrazovka s populárními filmy.

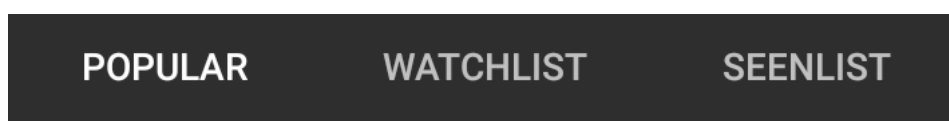


Obr. 4.2 Orientace v aplikaci

4.3 Detail jednotlivých prvků aplikace

4.3.1 Tlačítka pro výběr seznamu

Tyto tlačítka jsou hlavním navigačním elementem v aplikaci. Jejich pomocí se uživatel přesunuje mezi seznamy. Vybraný seznam je znázorněn světlejší barvou textu a modrou čarou ve spodní části.



Obr. 4.3 Navigační menu

4.3.2 Vyhledávací okénko

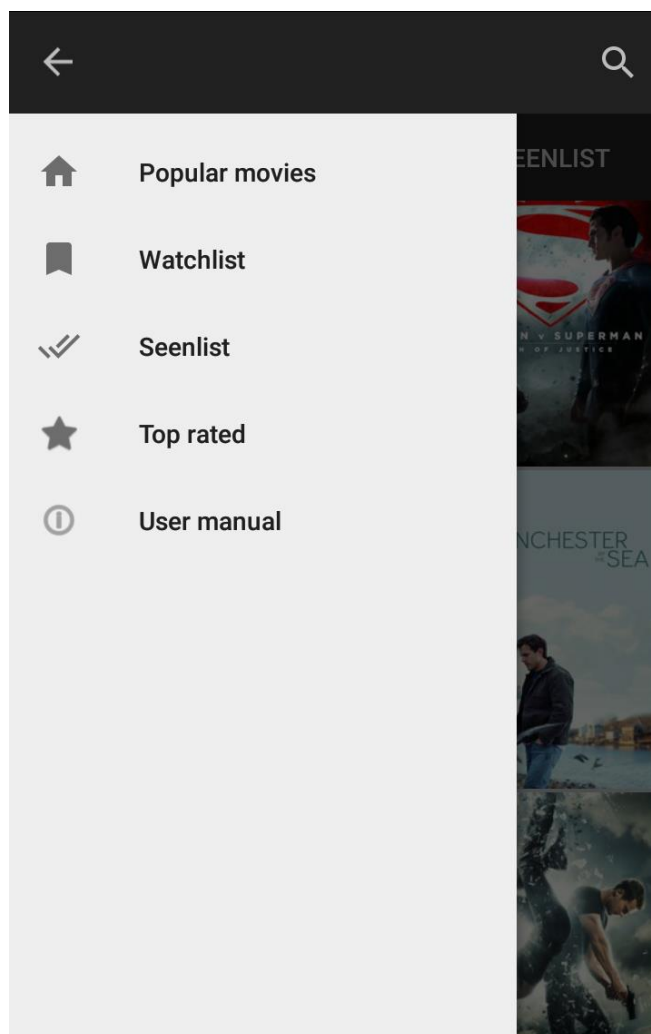
Pro případ, že by uživatel chtěl vyhledat nějaký titul, je v pravém horním rohu ikonka lupy, která, pokud se jí uživatel dotkne, se rozšíří v textové pole. Po napsání a potvrzení jména filmu, je uživatel přesměrován, na další obrazovku s výsledky vyhledávání.



Obr. 4.4 Horní menu s lupou a ikonkou vysunovacího menu

4.3.3 Levé vysunovací menu

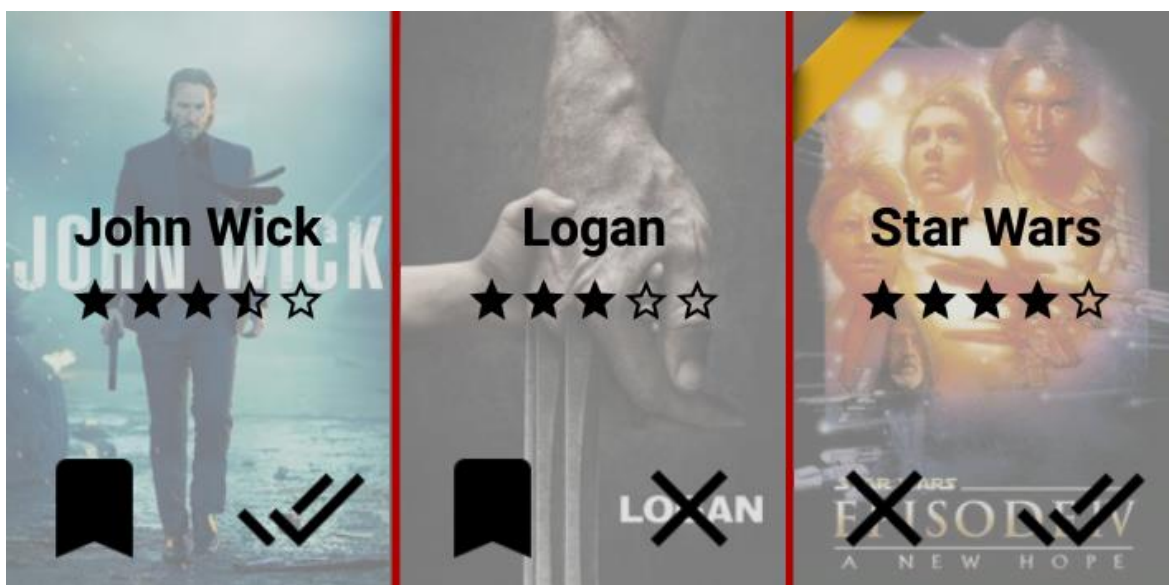
Levé vysouvací menu slouží k zjednodušené navigaci v aplikaci. Po vysunutí se zobrazí 3 hlavní seznamy, nejlépe hodnocené filmy a uživatelský manuál. Po dotyku je uživatel přesměrován na odpovídající obrazovku.



Obr. 4.5 Vysunovací menu

4.3.4 Ovládací prvky jednotlivých buněk

Každá buňka v mřížce obsahuje obrázková tlačítka, jenž po dotyku přidají film do odpovídajících seznamů. Design těchto tlačítek byl zvolen tak, aby uživateli bylo hned jasné, co tlačítka dělají. Pro přehlednost se po stisku tlačítka vypíše na obrazovku text, co aplikace provedla za operace. Pokud je uživatel ve Watchlistu nebo Seenlistu, jsou tlačítka nahrazena za taková tlačítka, jenž po dotyku odstraní film z příslušného seznamu.



Obr. 4.5 Ovládací prvky jednotlivých buněk

4.3.5 Vytvoření notifikace na datum vydání filmu

Uživatel si může nechat automaticky vytvořit upozornění na premiéru filmu pomocí tlačítka umístěného v informačním panelu filmu uvnitř aktivity MovieDetail. Toto tlačítko se zobrazí, pouze pokud film ještě neměl premiéru. Událost bude mít automaticky vyplněné parametry, pokud však uživatel chce, může si ji upravit dle svých představ.



Obr. 4.6 tlačítko na vytvoření události

ZÁVĚR

Vývoj této aplikace trval autorovi asi 2 měsíce. Většinu času strávil nad studiem a seznamováním se s novými koncepty. Díky tomuto projektu se autor naučil pracovat s databázemi a pochopil základy získávání dat z internetových serverů. Aplikace samotná prošla mnohými úpravami. Od menších grafických změn z důvodu přehodnocení návrhu aplikace, až po rozsáhlé přepisování celého kódu. Pokud by práce nebyla časově omezena, autor by zajisté pokračoval ve vylepšování a optimalizaci aplikace, ať už ze stránky grafické, či logické.

I přesto, že zadání práce se nezdá příliš obtížné, autor při vývoji narazil na mnohé překážky. První z nich bylo vymyšlení způsobu získávání potřebných informací o filmech. Po prvním neúspěšném pokusu stahování dat přímo z HTML stránek zaměřených na kinematografii, našel autor skvělou databázi filmů, jenž mu poskytla osobní (nekomerční) klíč k jejím datům. Dalším z problémů bylo vytvoření grafického prostředí, jenž by správně fungovalo v různých rozlišeních, a na mnohých velikostech dnešních telefonů. Převážná část problémů pramenila z nezkušenosti a nevědomosti autora v oblasti problematiky vývoje Android aplikací.

Zásluhou neúnavného testování se však většina závažnějších problémů našla a včas opravila. Do této chvíle je však chod aplikace na některých zařízeních doprovázen doposud nevysvětlenými jevy, které autor doufá opravit při dalším vývoji. Po dalším rozšíření funkcí aplikace autor plánuje umístit projekt na Google Play, díky čemuž může oslovit širší okruh lidí a získat rozsáhlejší zpětnou vazbu, ovlivňující budoucí podobu aplikace.

SEZNAM POUŽITÉ LITERATURY

- [1] Android - aktivity <https://developer.android.com/reference/android/app/Activity.html>
- [2] API – dokumentace <https://developers.themoviedb.org/3/movies>
- [3] Databáze – JSON <http://www.json.org/json-cz.html>
- [4] Android – fragmenty <https://developer.android.com/guide/components/fragments.html>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

GUI – grafické uživatelské rozhraní (graphic user interface)

API – programové aplikační rozhraní (application program interface)

XML – rozšiřitelný značkovací jazyk (extensible markup language)

HTML – HyperTextový značkovací jazyk (HyperText Markup Language)

SEZNAM OBRÁZKŮ

Obr. 2.1 Android Studio	11
Obr. 2.2 XML Grafický editor Android studia	12
Obr. 2.3 Adobe Photoshop CS6.....	12
Obr. 2.4 Emulátor Android zařízení Genymotion	13
Obr. 3.1 Grafický návrh v Adobe Photoshop	15
Obr. 3.3 Adresářová struktura.....	17
Obr. 3.4 Zpracování dotazu na populární filmy.....	18
Obr. 3.5 Funkce pro zapisování a čtení souborů.....	19
Obr. 3.6 Funkce na přidávání filmů do seznamů	20
Obr. 3.7 Vykreslování jednotlivých buněk	21
Obr. 4.1 Vyžadovaná oprávnění	22
Obr. 4.2 Orientace v aplikaci	23
Obr. 4.3 Navigační menu	23
Obr. 4.4 Horní menu s lupou a ikonkou vysunovacího menu	24
Obr. 4.5 Vysunovací menu	24
Obr. 4.5 Ovládací prvky jednotlivých buněk.....	25
Obr. 4.6 tlačítko na vytvoření událost.....	25

SEZNAM PŘÍLOH

P1: konzultace

P2: CD obsahující práci

DOKLAD O KONZULTACI MATURITNÍ PRÁCE

	Datum	Podpis vedoucího práce
Konzultace č. 1		
Konzultace č. 2		