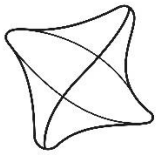


ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Fakulta riadenia a informatiky

Planet Cinema

Dokumentácia

Borys Plotnikov

Žilina 2024

Obsah

1. Popis a analýza riešeného problému

1.1. Špecifikácia zadania a definovanie problému

V dnešnej dobe sa každý stretáva s problémom, že má chuť niečo pozrieť, ale rozhodnúť sa je veľmi ťažké. A keď príde na výber filmu na sledovanie v spoločnosti, to môže trvať nie minúty, ale celé hodiny hľadania "toho správneho" filmu. V mojej práci som sa rozhodol vytvoriť aplikáciu, ktorá sa pokúsi vyriešiť túto náročnú úlohu.

Plánovaná aplikácia je určitým zoznamom, ktorý pomôže používateľovi uložiť filmy, ktoré chce pozrieť, alebo si uložiť svoj názor na konkrétny film. Aplikácia má tiež filtre podľa žánrov, hodnotení a ďalších parametrov, čo umožní používateľom ľahko nájsť film podľa ich požiadaviek.

Jednou z osobitostí aplikácie je koleso. Toto koleso slúži na náhodný výber filmu. Napríklad, ak používateľ má viacero možností filmov na výber, môže použiť toto koleso na náhodný výber filmu. Táto mechanika umožní používateľom ľahšie robiť rozhodnutia.

Ďalšou osobitosťou je vytvorenie špeciálneho menu na výber filmov pomocou swapovania. To znamená, že v tomto menu užívateľ uvidí stručné informácie o filme a bude mať možnosť pretočiť doľava alebo doprava. Podľa smeru pohybu pridá film do svojho zoznamu želaní. Toto umožní užívateľom preskúmať rôzne možnosti a zvýšiť počet filmov na výber.

Táto aplikácia by mala pomôcť používateľom pri výbere filmov a tiež umožní uložiť filmy do ich zoznamu želaní pomocou bežného pridávania, čo umožní zapamätať si filmy, ktoré používateľ chce pozrieť v budúcnosti.

1.2. Podobné aplikácie

Po prevedení niekoľkých prieskumov na internete som objavil podobné aplikácie. V tejto kapitole podrobne opíšem niektoré z nich.

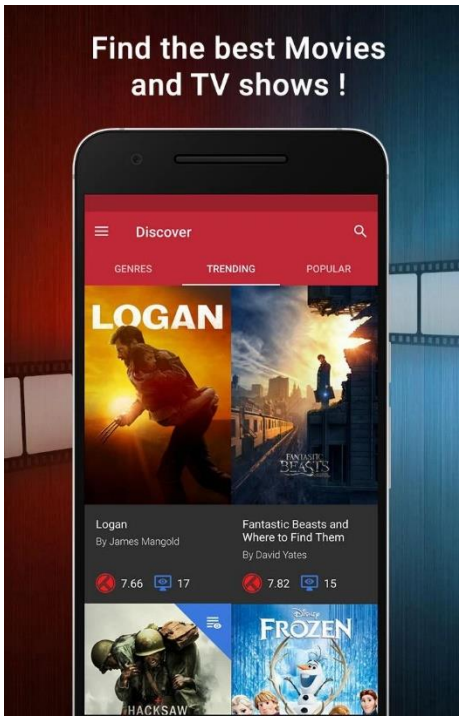
1.2.1. CineTrak

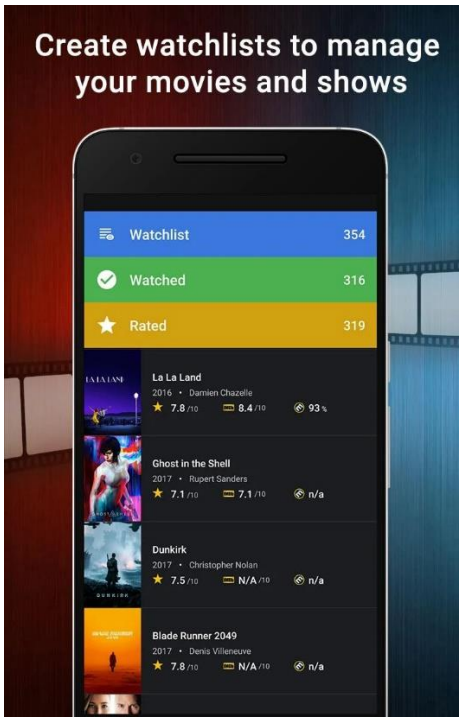
oto je aplikácia pre Android, ktorá ponúka funkcie vyhľadávania a filtrovania filmov, ukladania zoznamov želaní na sledovanie, ako aj možnosť zanechávať recenzie a hodnotenia.

CineTrak umožňuje objavovať najlepšie filmy a televízne programy, vytvárať svoje zoznamy na sledovanie, prezerať si recenzie a hodnotenia na rôznych zdrojoch, ako sú Trakt, IMDB, Metacritic a Rotten Tomatoes. CineTrak je navrhnutý na báze Google Material Design a ponúka hrateľné prostredie, vďaka čomu je sledovanie filmov a televíznych programov ešte zábavnejšie a pútavejšie.

Taktiež poskytuje rôzne zoznamy, ktoré ľahko poskytujú akékoľvek informácie o televíznych programoch a filmoch. Výber redakcie obsahuje filmy, ktoré získali Oscara a iné ocenenia, ako aj

zoznam celebrít. Zoznamy odporúčaných filmov na sledovanie obsahujú najlepšie filmy z mnohých kategórií, ako sú Najlepší film, Najocenenejšie filmy a Najväčšie trháky, ako aj filmy z mnohých ďalších pravidelne aktualizovaných kategórií. Umožňuje vytvárať značky na filmoch a seriáloch, aby ste mohli svojim priateľom povedať, čo práve sledujete. Offline prezeranie: údaje sú dostupné aj vtedy, keď užívateľ nie je pripojený k sieti.





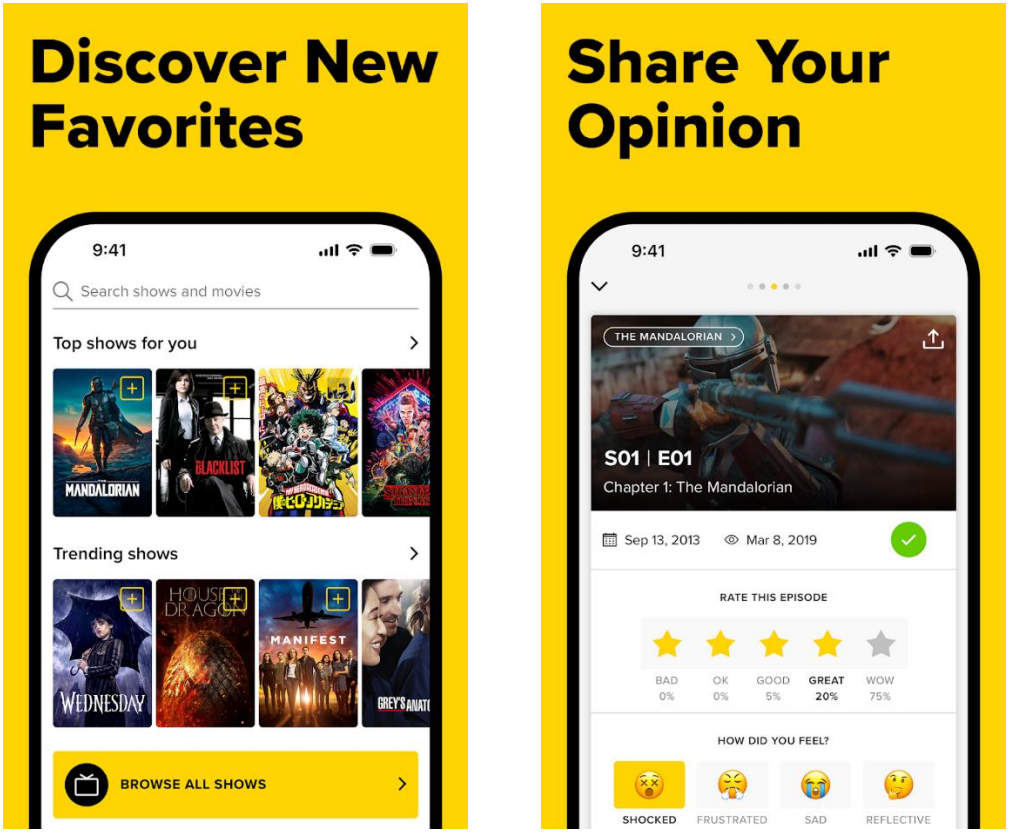
Obrázky aplikácie CineTrak

Link: https://www.cinetrakapp.com/

1.2.2. TV Time

TV Time je aplikácia na sledovanie televíznych relácií a filmov, ktorá je dostupná na mobilných zariadeniach (iOS a Android) aj na webovej platforme. Poskytuje používateľom pohodlný spôsob, ako sledovať svoje obľúbené seriály a filmy, označovať sledované epizódy, dostávať upozornenia o nových epizódach a ďalšie funkcie.

Používatelia môžu tiež nastaviť oznámenia o vydaniach nových epizód svojich obľúbených relácií a filmov, aby boli informovaní o najnovších aktualizáciách. Používatelia môžu pridávať seriály a filmy do zoznamu „Watchlist“, aby na ne neskôr nezabudli. Aplikácia poskytuje kalendár, v ktorom sú zobrazené termíny vydania nových epizód sledovaných seriálov. Aplikácia poskytuje štatistiky o čase strávenom sledovaním, počte zobrazených epizód a ďalšie údaje. Aplikácia ponúka odporúčania na základe preferencií používateľa, predchádzajúcich sledovaniach relácií a filmov, ako aj ich popularite a hodnotení.



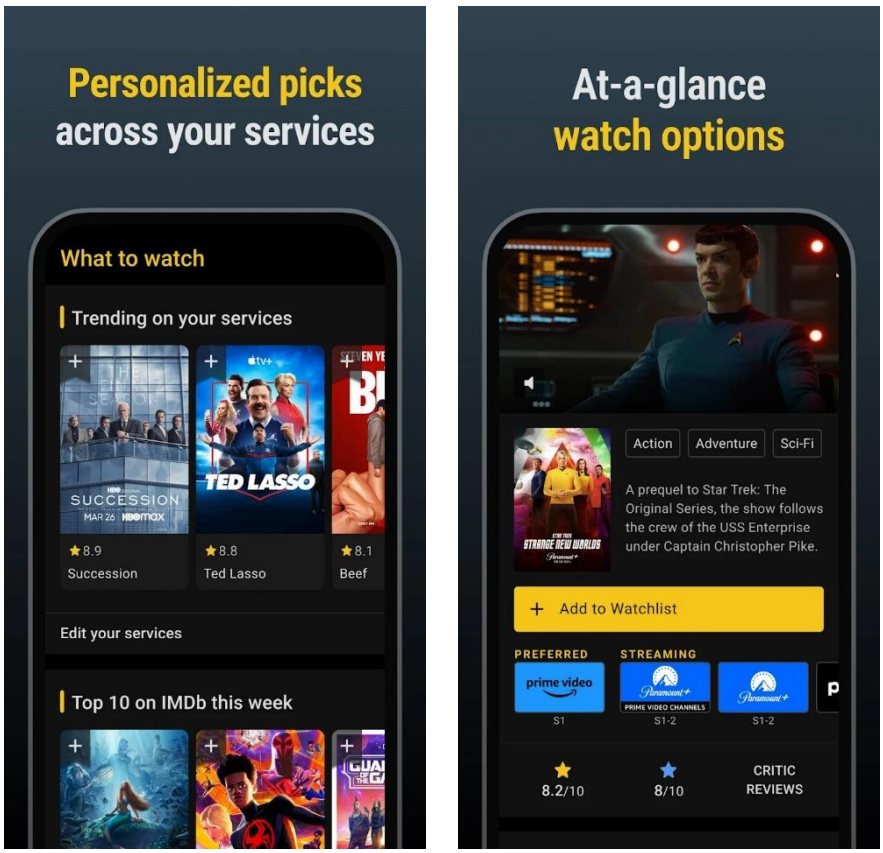
Obrázky aplikácie TV Time

Link: https://www.tvtime.com/de

1.2.3. IMDb: Movies & TV Shows

IMDb: Movies & TV Shows - je oficiálna mobilná aplikácia IMDb (Internet Movie Database), jedného z najpopulárnejších online zdrojov o filme a televízii. Aplikácia je k dispozícii pre zariadenia so systémom Android aj iOS a ponúka užívateľovi rozsiahlu databázu filmov, seriálov, hercov, režisérov a ďalších osobností filmového priemyslu.

IMDb poskytuje najnovšie správy a aktualizácie z oblasti filmového priemyslu, vrátane oznámení o nových filmoch, premiérach a udalostiach. Okrem toho IMDb poskytuje informácie o oceneních a nomináciách, ktoré získali filmy a herci. Používatelia môžu vyhľadávať informácie o filmoch, seriáloch, hercoch, režiséroch a iných osobnostiach spojených s filmovým priemyslom. IMDb poskytuje hodnotenia a recenzie filmov a televíznych relácií, založené na hodnoteniach užívateľov a kritikov. Používatelia môžu prispôsobiť svoj profil a dostávať personalizované odporúčania a ukladať svoje preferencie. Používatelia môžu vytvárať vlastné zoznamy filmov a televíznych relácií, aby sledovali zaujímavý obsah.



Obrázky aplikácie IMDb: Movies & TV Shows

Link: https://play.google.com/store/apps/details?id=com.imdb.mobile&hl=en_US

1.2.4. Rozdiely

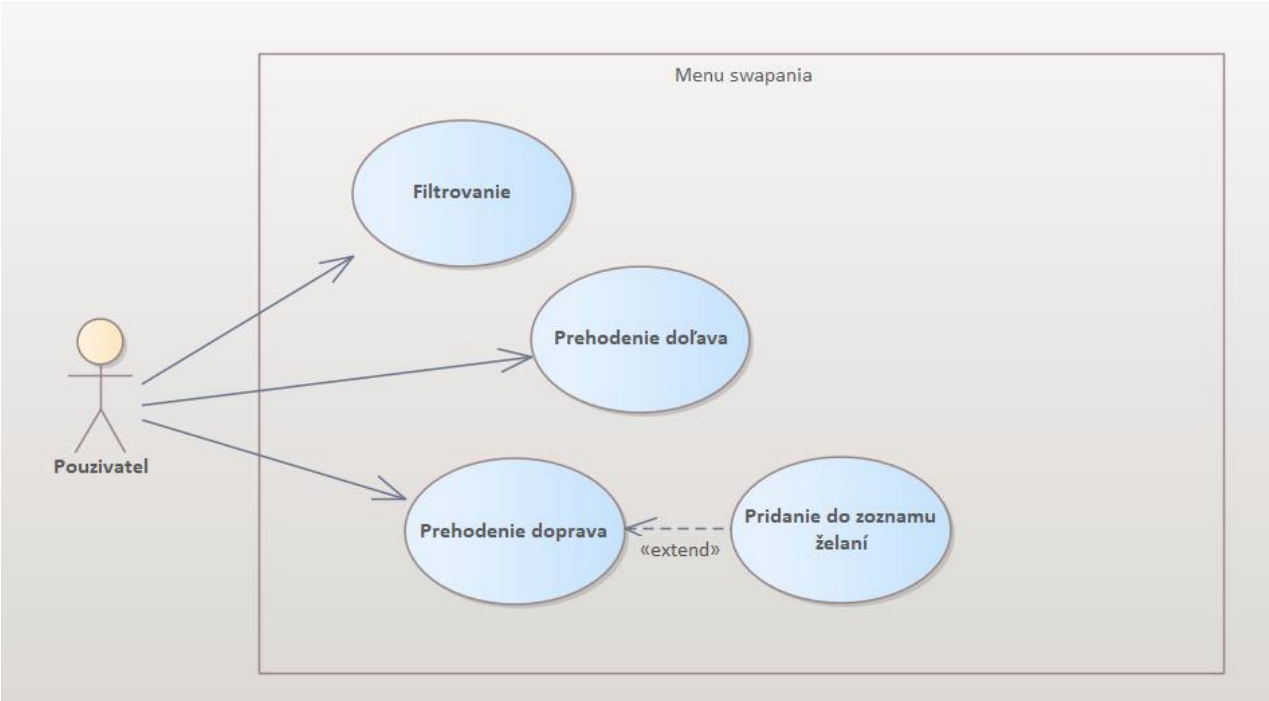
Vytvorená aplikácia obsahuje niektoré podobné aspekty, ale existujú aj rozdiely. V žiadnej z nájdených podobných aplikácií nie je implementovaný systém swapania s možnosťou swapania filmov a systém náhodného výberu s použitím kolesa. Tiež je v aplikácii poskytnutá možnosť porovnania vlastného hodnotenia s hodnotením poskytnutým z externých zdrojov. Taktiež v aplikácii je možné dať opätovné hodnotenie, pričom predchádzajúce zostane zachované.

2. Návrh riešenia problému

2.1. Krátka analýza

2.2.1. Menu swapania

Diagram zobrazuje možné udalosti, ktoré sa môžu vyskytnúť počas interakcie používateľa s menu swapania.



Filtrovanie - používateľ bude mať možnosť filtrovať filmy podľa svojich požiadaviek.

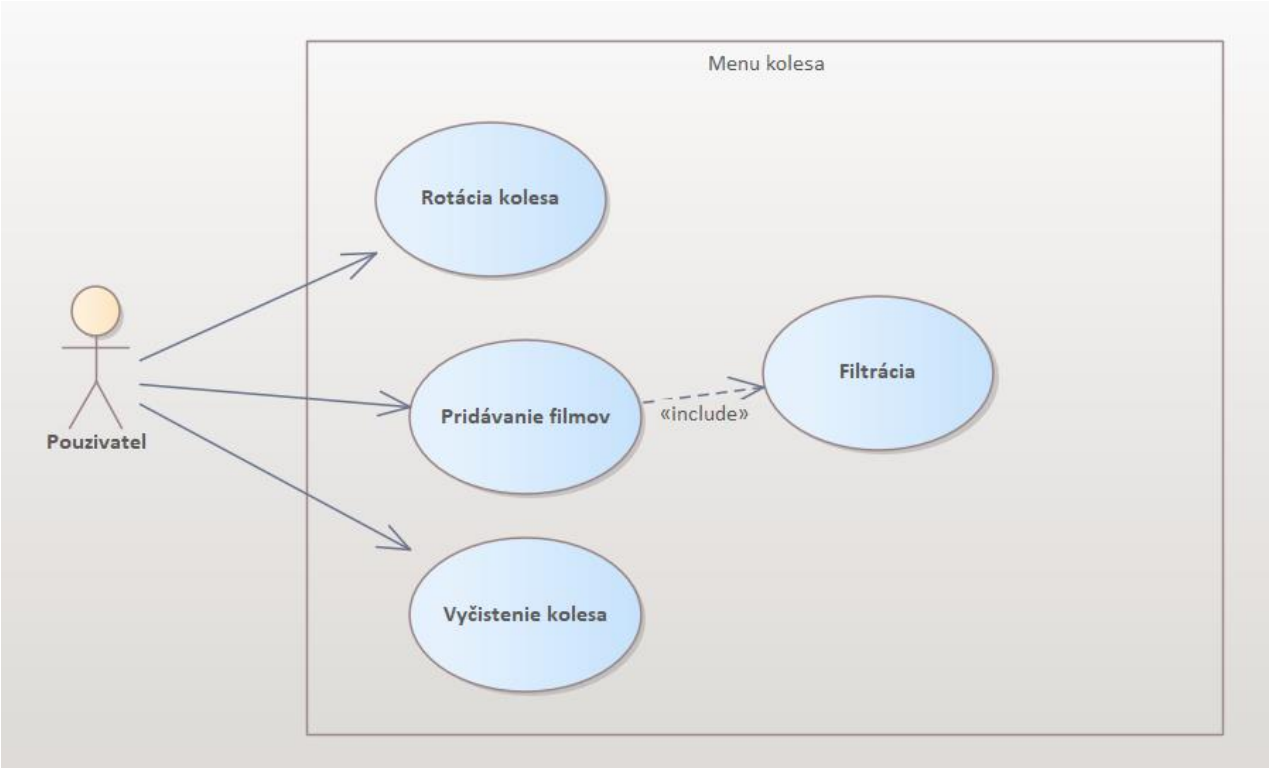
Prehodenie doľava - pri posúvaní filmu doľava sa film odstráni z ponuky a následne sa zobrazí nový film.

Prehodenie doprava - pri posunutí filmu doprava sa film tiež odstráni zo zoznamu ponúkaných a zobrazí sa nový film, ale tento film sa pridá do zoznamu želaného.

Pridanie do zoznamu zelani - pridáva všetky informácie o filme a samotný film do zoznamu želaného používateľa, s ktorým následne môže interagovať.

2.2.2. Menu kolesa

Diagram zobrazuje možné udalosti, ktoré sa môžu vyskytnúť počas interakcie používateľa s menu kolesa.



Rotácia kolesa - umožňuje používateľovi otáčať kolieskom a získať náhodný film z tohto kolieska. Ak používateľ koliesko nevyplní, nemôže ho otáčať.

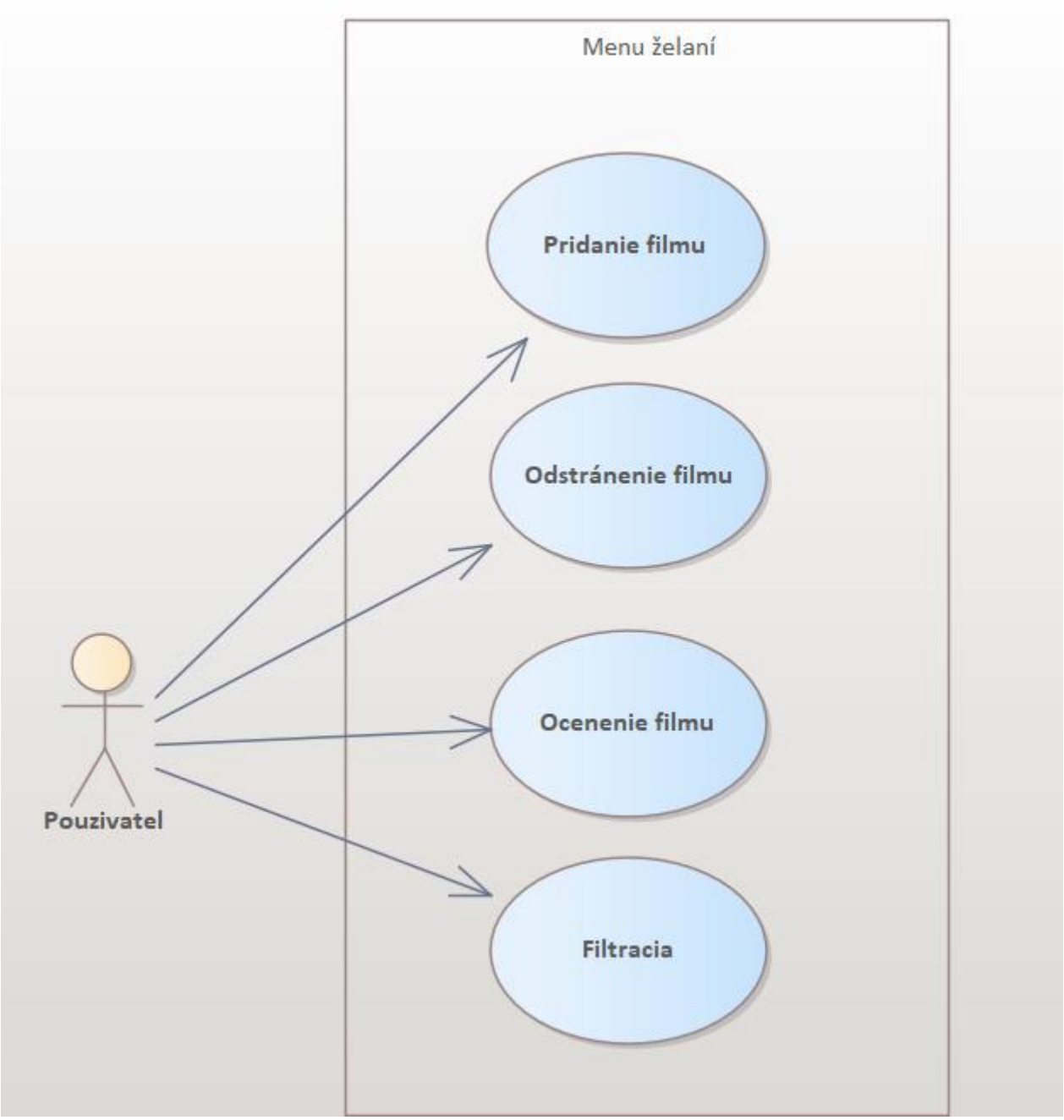
Pridávanie filmov - umožňuje používateľovi pridať film zo svojho zoznamu želaní do kolieska.

Filtrácie - umožňuje filtrovať filmy, ktoré majú byť pridané do kolieska podľa rôznych kritérií.

Vyčistenie kolesa - v prípade, že je koleso naplnené, umožňuje vyčistiť.

2.2.3. Menu želaní

Diagram zobrazuje možné udalosti, ktoré sa môžu vyskytnúť počas interakcie používateľa s menu želaní.



Pridanie filmu - používateľ pridáva svoj film s uvedením všetkých parametrov. Film bude pridaný do databázy.

Odstránenie filmu - používateľ odstráni film zo svojho zoznamu želaní.

Ocenenie filmu - používateľ poskytuje svoje hodnotenie. Film je označený ako pozretý.

Filtrácie - Filtrovať zoznam želaní podľa parametrov, ako sú žáner, hodnotenie filmu, trvanie atď.

2.2. Návrh aplikácie

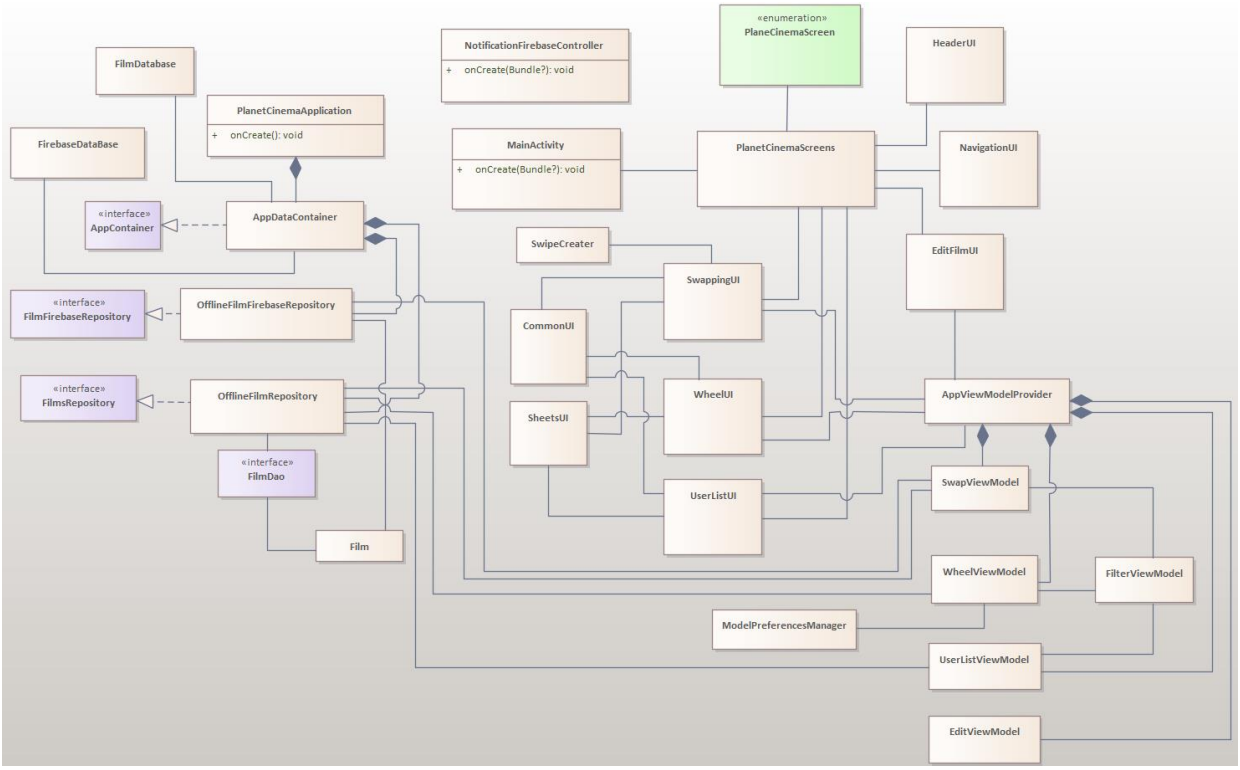


Diagram tried

MainActivity – je trieda, ktorá spúšťa vizuálnu časť projektu a spúšťa všetky Composable funkcie.

PlanetCinemaApp – je súbor, ktorý uchováva všetky Composable funkcie, ktoré zobrazujú jednotlivé obrazovky projektu, a zároveň je zodpovedný za navigáciu.

PlaneCinemaScreen – je enum, ktorý obsahuje názvy pro každý displej.

HeaderUI – obsahuje Composable funkcie na zobrazenie horného panela aplikácie.

NavigationUI – reprezentuje Composable funkcie pre navigačný prvok.

EditFilmUI, SwappingUI, UserListUI, WheelUI – obsahujú Composable funkcie na zobrazenie príslušných panelov aplikácie.

CommonUI – obsahuje Composable funkcie pre spoločné prvky projektu.

SheetsUI – obsahuje Composable funkcie pre vysúvacie panely na zobrazenie filtra a na výber filmov pre koleso.

SwipeCreater – obsahuje Composable funkcie pre jednotlivé prvky pre panel na swapovanie.

AppViewModelProvider – je singleton, ktorý vytvára view modely pre aplikáciu.

SwapViewModel, UserListViewModel, WheelViewModel, EditViewModel - sú view modely pre príslušné obrazovky, ktoré vytvárajú spojenie a komunikáciu medzi UI a stavmi.

FilterViewModel – je pomocným view modelom pre ostatné obrazovky, zodpovedným za filtrovanie a vytváranie spojenia medzi UI objektami a stavmi filtra.

ModelPreferencesManager – je singleton, ktorý ukladá údaje do SharedPreferences.

OfflineFilmRepository – je trieda, ktorá pracuje s dao filmov z lokálnej databázy a implementuje základné funkcie pre prácu s filmami v databáze. Implementuje rozhranie FilmsRepository.

OfflineFilmFirebaseRepository – je trieda, ktorá pracuje s Firebase databázou a implementuje základné funkcie pre prácu s filmami v databáze. Implementuje rozhranie FilmFirebaseRepository.

FilmsRepository, FilmFirebaseRepository – sú rozhrania, ktoré definujú funkcie pre prácu s príslušnými databázami.

FilmDao – je rozhranie, ktoré definuje funkcie pre prácu s filmami z lokálnej databázy.

Film – je trieda, ktorá reprezentuje film ako objekt.

AppDataContainer – je trieda, ktorá vytvára repozitáre a pripája do nich databázy. Implementuje rozhranie AppContainer.

AppContainer – je rozhranie, ktoré definuje, ktoré repozitáre budú vytvorené v projekte.

PlanetCinemaApplication – je trieda na vytvorenie objektu ModelPreferencesManager a AppDataContainer.

FilmDatabase – je reprezentáciou lokálnej databázy, obsahuje singleton pre ďalšie pohodlné použitie.

FirebaseDataBase – je trieda pre vytvorenie pripojenia k vzdialenému serveru Firebase. Implementuje základné funkcie pre prácu s databázou.

NotificationFirebaseController – vytvára požiadavku na používanie oznámení pre používateľa.

3. Popis implementácie

Navigation - v projekte je implementovaná navigácia medzi obrazovkami. Na jej implementáciu som použil knižnicu androidx.navigation. Použil som enum na vytvorenie názvov jednotlivých obrazoviek a s použitím NavHostController vykonávam samotnú navigáciu v projekte.

Funkcia navigácii

Room – v projekte sa používa lokálna databáza, ktorá bola implementovaná pomocou knižnice androidx.room. Táto databáza slúži na uchovávanie uložených filmov v zozname želaní používateľa. Pri implementácii databázy boli použité rozhranie Dao a taktiež repozitár na vytvorenie správnej architektúry. Pri zmene filmu sa všetky údaje menia v lokálnej databáze.

ViewModel – určený na uchovávanie a spravovanie údajov potrebných na zobrazenie používateľského rozhrania a vykonávanie obchodnej logiky v aplikácii. V projekte sú vytvorené pomocou knižnice androidx.lifecycle. Hlavným účelom ViewModelu je zabezpečiť uchovávanie údajov počas zmien konfigurácie a poskytnúť jednoduchý spôsob prístupu k týmto údajom z používateľského rozhrania.

Firebase Database - používa sa na uchovávanie údajov v externom databázovom systéme. Používa sa na získavanie filmov v projekte pri ich swapaní. Z týchto filmov sa vyberajú tie, ktoré nie sú uložené v lokálnej databáze. Na implementáciu tejto databázy bola použitá knižnica com.google.firebase.database. Na prácu s databázou sa používa repozitár na vytvorenie správnej architektúry projektuV databáze Firebase bolo rozhodnuté nepridávať prvky vytvorené používateľom s cieľom kontrolovať filmy, ktoré môžu byť ponúknuté používateľom.

```
suspend fun getFilmById(id : String) : Film? {
    var film : Film? = null
    try {
        val filmSnapshot = database.child( pathString: "films").child(id).get().await()
        if (filmSnapshot.exists()) {
            film = makeFromDataSnapshotToFilm(filmSnapshot)
        } else {
            Log.i(TAG,  msg: "No films found")
        }
    } catch (exception: Exception) {
        Log.e(TAG,  msg: "Error getting data", exception)
    }
    return film
}

 Tobito45
private fun makeFromDataSnapshotToFilm(data : DataSnapshot) : Film {
    val name = data.child( path: "name").getValue(String::class.java)
    val author = data.child( path: "autor").getValue(String::class.java)
    val mark = data.child( path: "mark").getValue(Float::class.java)
    val url = data.child( path: "url").getValue(String::class.java)
    return Film(name = name !!, autor = author !!, mark = mark !!, url = url !!)
}
```

Funkcia na získanie filmu z určitým id

Firebase Messaging - používa sa na možnosť odosielania a prijímania notifikácií používateľovi. Môže byť použité na každodenné oznámenie o sledovaní filmu,, ako aj na oznámenie o novej verzii projektu a požiadavke na aktualizáciu aplikácie. Pred začatím práce si aplikácia vyžiada povolenie na prijímanie notifikácií. Na vytvorenie bola použitá knižnica com.google.firebase.messaging.

Toast – pre niektoré akcie v aplikácii (ako napríklad swapanie, požiadavka na notifikácie, výber kolesa) sa používa krátka informácia vo forme výpisu. Tento výpis je vykonaný pomocou knižnice android.widget.

```
Toast.makeText(mLocalContext,  text: "Next film", Toast.LENGTH_SHORT).show()
```

Vytvorenie oznámenia

AsyncImage – pre zobrazovanie obrázkov k filmom som sa rozhodol použiť načítanie cez URL odkazy, aby si používateľ mohol jednoducho nahrať svoj obrázok a získať obrázok z internetu. Tento odkaz je uložený v databázach a načíta sa pri zobrazení informácií o filme. Na implementáciu bola použitá knižnica coil.

```
@Composable
fun BasicAsyncImage(url : String, modifier: Modifier = Modifier) {
    AsyncImage(
        model = ImageRequest.Builder(LocalContext.current)
            .data(url)
            .crossfade( enable: true)
            .build(),
        contentDescription = null,
        contentScale = ContentScale.Crop,
        error = painterResource(R.drawable.error_icon),
        placeholder = painterResource(R.drawable.loading_icon),
        modifier = modifier
    )
}
```

Funkcia na vytvorenie AsyncImage

SharedPreferences – pre pohodlnú prácu s aplikáciou bolo rozhodnuté, že je možné ukladať hodnoty do kolesa a lokálne ukladať filmy na zariadenie. Pri opätovnom spustení obrazovky s kolesom sa automaticky načítajú údaje zo SharedPreferences. Na implementáciu sa použila knižnica android.content.

```
object ModelPreferencesManager {

    lateinit var preferences: SharedPreferences

    private const val PREFERENCES_FILE_NAME = "PREFERENCES_FILE_NAME"
    Tobito45
    fun with(application: Application) {
        preferences = application.getSharedPreferences(
            PREFERENCES_FILE_NAME, Context.MODE_PRIVATE)
    }

    Tobito45
    fun <T> put(`object`: T, key: String) {
        val jsonString = GsonBuilder().create().toJson(`object`)
        preferences.edit().putString(key, jsonString).apply()
    }

    Tobito45
    inline fun <reified T> get(key: String): T? {
        val value = preferences.getString(key, defValue: null)
        return GsonBuilder().create().fromJson(value, T::class.java)
    }
}
```

Objekt ModelPreferencesManager

SwipeAction – používal sa na vytvorenie efektu a animácie ťahania filmu doľava a doprava. Realizované s použitím knižnice SmartSwipe. Pri ťahaní doľava sa zobrazuje nový film, pri ťahaní doprava sa zobrazuje nový film a predchádzajúci sa ukladá do lokálnej databázy.

```
fun CreateSwipeAction(OnSwipe : () -> Unit, background : Color) : SwipeAction {
    val backgroundWithAlpha = background.copy(alpha = 0.25f)

    return  SwipeAction(
        onSwipe = OnSwipe,
        icon = {
            Icon(
                painter = painterResource(id = R.drawable.empty),
                contentDescription = null,
            )
        },
        isUndo = true,
        weight = 2.0,
        background = backgroundWithAlpha
    )
}
```

Funkcia na vytvorenie SwipeAction

SpinWheel – je objekt, ktorý sa používal na vytvorenie kolieska. Na vytvorenie kolieska sa rozhodlo použiť upravenú knižnicu SpinWheelCompose. Koliesko sa používa na výber náhodného filmu. Na koliesku sa nachádza neviditeľné tlačidlo, ktoré po stlačení otočí koliesko počas 8 sekúnd nekonečne a potom ho zastaví na náhodnom prvku. Získaný výsledok sa vypíše pomocou Toast. Pri výbere nových filmov alebo pri odstránení filmov z kolieska sa automaticky prekreslí a zobrazí nový vzhľad kolieska.

AutoResizedText - je typom textu, ktorý mení svoju veľkosť v závislosti od svojho kontajnera. Používa sa v kolese na automatické upravenie veľkosti textu v kolese pri pridávaní prvkov.

4. Zoznam použitých zdrojov

Predefinovanie kolieska - https://medium.com/voodoo-engineering/creating-a-spin-wheel-in-compose-b71d1b0c7b77

Firebase Database - https://firebase.google.com/docs/database/android/start

Firebase Messaging - https://firebase.google.com/docs/cloud-messaging/android/client

Swipovanie - https://qibilly.com/SmartSwipe-tutorial/

AutoResizedText - https://www.youtube.com/watch?v=ntlyrFw0F9U