

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

ODPORÚČANIE PRE INTELIGENTNÚ TV
DOKUMENTÁCIA K INŽINIERSKEMU DIELU

Vedúci tímu: Ing. Dušan Zeleník

Členovia tímu: Bc. Demovič Ľuboš, Bc. Fritscher Eduard, Bc. Kříž Jakub, Bc. Kuzmík Ondrej, Bc. Prokop Martin, Bc. Proksa Ondrej, Bc. Vandlíková Diana

Akademický rok: 2012/2013

Obsah

1	Úvod	5
1.1	Celkový pohľad na projekt	5
1.2	Ciele produktu	5
2	1. Šprint	6
2.1	Analýza existujúcich riešení	6
2.2	Vytvorenie prezentačnej stránky	19
3	2. Šprint	24
3.1	Analýza štruktúry entít. Ontológia	24
4	3. Šprint	26
4.1	Služba Neo4j	26
4.2	Parser kín	28
4.3	Model používateľa	30
5	4. Šprint	31
5.1	Prvotný návrh rozhrania webovej aplikácie a prípady použitia	31
6	5. Šprint	38
6.1	Získanie testovacích dát TV programu	38
6.2	Linked Movie Database Parser	40
6.3	Neo4j prídavný modul - Algoritmus spájania farieb	41
6.4	Neo4j prídavný modul - Algoritmus miešania farieb	44
6.5	Neo4j prídavný modul - Algoritmus šírenia energie	45
6.6	Prototyp odporúčania	46
7	6. Šprint	48
7.1	Metóda na napĺňanie databáz z linked movie parsera	48
7.2	Spojenie Neo4j prídavných modulov	49
8	7. Šprint	51
8.1	Neo4j prídavný modul - Dijkstra plugin	51
8.2	Optimalizácia Neo4j prídavého modulu - Energy plugin	52
8.3	Testovanie Neo4j prídavných modulov	52
8.4	Grafické rozhranie č. 1 - Modul odporúčania TeleVido	54
9	8. Šprint	58
9.1	Synchronizátor	58
9.2	Grafické rozhranie č. 2	62
9.3	Nasadenie grafického rozhrania	66

10 Zimný semester	67
10.1 Architektúra aplikácie	67
10.2 Dáta	68
10.3 Neo4j pluginy	68
10.4 Služba	68
10.5 Webová aplikácia	68
11 9. Šprint	70
11.1 ČSFD parser	70
11.2 Pridanie používateľa do systému	71
11.3 Prihlásenie, registrácia, profil filmu / osoby čo som hodnotil, čo ma zaujíma	72
11.4 Komunikácia medzi serverom a android aplikáciou	76
11.5 Návrh používateľského rozhrania Android aplikácie	78
12 10. Šprint	78
12.1 Nasadenie Cronov	78
12.2 API	80
12.3 Môj vlastný TV - program	81
12.4 Navigácia a nastavenia v Android aplikácii	83
12.5 Neo4j prídavný modul - Algoritmus šírenia energie, efektívnejšia verzia	86
12.6 Ďalšie návrhy pre používateľské rozhranie Android aplikácie a ich implementácia do XML	87
12.7 Zobrazenie kín pri odporúčaní	91
13 11. Šprint	92
13.1 Entity pre používateľa	92
13.2 Obrázky pre Android aplikáciu	93
13.3 Poster na konferenciu IIT SRC	93
13.4 Odporúčanie filmov Android aplikácii	94
13.5 Bookmarky, blacklisty	96
13.6 Parser kín - nová verzia	97
14 12. Šprint	98
14.1 Informácie o projekte	98
14.2 Odporúčanie kín v android aplikácii	99
14.3 Prerobenie odporúčania	101
14.4 Výber TV staníc	102
14.5 Zmena v nastavenia android aplikácie	102
14.6 Pridanie galérie hercov a filmov	103
14.7 Výber TV staníc - prezenčná vrstva	104

15	13. Šprint	105
15.1	Grafické rozhranie kontaktnej stránky projektu	105
15.2	Funkcia “Štrng” v android aplikácii	105
15.3	Môj TV-program	108
16	14. Šprint	109
16.1	Grafický návrh nástenky pre prihláseného používateľa	109
16.2	Nástenka prihláseného používateľa	111
16.3	Doladenie android aplikácie	111
16.4	Verejná dokumentácia k API	112
17	Používateľská príručka	113
17.1	Neprihlásený používateľ	113
17.2	Prihlásený používateľ	127

1 Úvod

1.1 Celkový pohľad na projekt

Naším hlavným cieľom je vytvorenie webovej služby, ktorá je schopná odporúčať nielen filmy, ale aj personalizovaný televízny program, či odporúčať filmy práve bežiacie v kinách. Táto služba môže byť súčasťou multimediálnych zariadení alebo byť volaná prostredníctvom nášho webu či mobilnej aplikácie. Pričom sa dá rozšíriť aj na ďalšie zariadenia ako sú napr. tablety. Okrem tohto cieľu sa sústredíme aj na ďalšie akými sú: skupinové odporúčanie, rôzne spôsoby hodnotenia a prehľadávania filmov.

1.2 Ciele produktu

V prvom semestri sa naše úsilie sústreďí na vytvorenie webovej služby, ktorá bude odporúčať filmy na základe modelu používateľa. Táto služba na základe filmov alebo seriálov či relácií, ktoré sa používateľovi páčili odporučí ďalšie, ktoré by sa mu mohli páčiť. Pre tento účel vytvoríme viacero grafových algoritmov, ktoré vyskúšame na reálnych dátach. Tieto algoritmy budú schopné filtrovať filmy či iné entity zo svojho vyhľadávania a tiež vrátiť najzaujímavejšie filmy z určitej množiny. To neskôr môžeme využiť v časovo ohraničenom odporúčaní. Reálne dáta získame parsovaním viacerých zdrojov, pričom nezostaneme len pri filmových databázach, ale budeme získavať dáta aj z programov kín či televíznych programov. Zároveň vytvoríme prototyp webového rozhrania. Tým položíme základy webovej aplikácie. Čo sa týka mobilnej aplikácie, v zimnom semestri bude vytvorený taktiež prototyp rozhrania, pričom základom tejto aplikácie bude rovnaká služba.

V druhom semestri zdokonalíme odporúčanie a vytvoríme webovú aplikáciu, ktorá umožní aj rôzne možnosti vyhľadávania filmov. Používatelia budú môcť prostredníctvom fazetového vyhľadávania prehľadávať filmy a hodnotiť ich. Tak získame ďalšie možnosti pre odporúčanie filmov. Okrem filmov bude aplikácia schopná odporučiť aj televízny program, ktorý bude časovo ohraničený a filmy práve bežiacie v kinách. Tieto všetky možnosti zakomponujeme do webového a niektoré aj do mobilného rozhrania. Jedným z našich cieľov je aj vytvorenie skupinového odporúčania, teda odporúčania filmov pre skupiny priateľov či známych, ktorí si chcú spolu pozrieť film či ísť do kina. Pre hodnotenie filmov zostávajú otvorené možnosti aj v prostredí sociálnych sietí.

2 1. Šprint

2.1 Analýza existujúcich riešení

2.1.1 Jinni

Jinni je vyhľadávač a odporúčač filmov a seriálov, ktorý vyhľadáva filmy na základe ich vlastností a nie na základe napríklad kľúčových slov alebo žánrov, ako je to zvyčajné. Jinni je postavený na projekte The Movie Genome, ktorý hodnotí filmy na základe kritérií, ktoré by mali byť čo najbližšie ľudskému hodnoteniu.

The Movie Genome The Movie Genome je databáza filmov, ktoré sú ohodnotené podľa ich charakteristík. Každý film má priradené tzv. gény rozdelené na dve kategórie: zážitok a príbeh. Gény zážitku sú napríklad atmosféra filmu (ľahká, strašidelná, drsná...). Gény príbehu sa týkajú napríklad zápletky (zamilovanie sa, rodinný život, všetko sa kazí...) alebo štruktúry príbehu (príbeh v príbehu, retrospektívne...). Každému filmu je priradených okolo 50 génov z tisícov možných.

Priradovanie génov sa deje dvoma spôsobmi. Jedným je ručné priradovanie génov odborníkmi. Druhým je použitie strojového učenia a NLP na analýzu textových hodnotení filmov bežnými divákmi a extrakcia metadát.

Vyhľadávanie filmov Filmový vyhľadávač je prístupný aj bez registrácie a funguje na základe génov priradených filmom. Používateľ zadá do vyhľadávača gény, ktoré by chcel vyhľadať a systém mu nájde film, ktorý má k požiadavkám najbližšie.

Gény sú pomenované ľudskou rečou, takže používateľ nemusí vedieť, ako presne vyhľadávanie funguje. Jednoducho zadá: „nenáročná komédia o priateľstve“ a systém rozozná patričné gény. Výsledky vyhľadávania si používateľ môže ďalej prispôbovať pomocou prvkov fazetového vyhľadávača: zobrazujú sa mu jednotlivé gény zaradené do skupín a používateľ si vyberie, ktoré ho zaujímajú, čím spresňuje výsledky vyhľadávania.

Vyhľadávať je možné aj filmy podobné zadanému, napríklad „like:The Shining“. Nájsené výsledky je možné ďalej upravovať špecifikovaním génov.

Odporúčanie filmov Odporúčač je sprístupnený po registrácii a ohodnotení istého počtu filmov. Podľa hodnotení sa systém Jinni snaží vytvoriť model používateľa tzv. Movie Personality. Tento model by mal vyjadrovať, aký typ filmov má používateľ v obľube, t.j. aké gény sa vyskytovali v dobre hodnotených filmoch a teda sú pre neho dôležité a ktoré naopak nemá rád. Model používateľa je viditeľný aj pre samotného používateľa vo forme tag cloudu.

Na základe modelu používateľa systém odporúča používateľovi filmy na prezretie a zlepšuje výsledky vyhľadávania. Používateľ má takisto možnosť pozrieť si svojich susedov, t.j. používateľov, ktorí k nemu majú najbližšie a zobraziť si, ako ďaleko od neho má ktorýkoľvek používateľ, čo je zaujímavý sociálny aspekt.

Nedostatky Zadeľovanie filmov podľa ich charakteristík je výborný nápad, ale má nedostatok v tom, že neberie do úvahy celkové hodnotenie filmu. Dva filmy s presne rovnakými génmi sa môžu páčiť rôzne. Je to jednoducho pre to, že jeden z nich je objektívne lepší film. Zjavným príkladom sú pokračovania filmov, kedy prvý film zo série mohol byť výborný a druhý horší, hoci tón filmu a príbeh mali podobný charakter.

2.1.2 MovieLens

MovieLens je odporúčač filmov vytvorený University of Minnesota. Používa kolaboratívne filtrovanie bežným spôsobom - podľa hodnotenia filmov nájde podobných používateľov a na základe ich hodnotenia odhaduje hodnotenia pre cieľového používateľa.

Hodnotenie a odporúčanie Po registrácii musí používateľ ohodnotiť minimálne 15 filmov, aby mu bolo sprístupnené odporúčanie. Systém mu náhodne generuje viacero filmov, ktoré má ohodnotiť až kým neohodnotí 15. Z nejakého dôvodu sa v týchto zoznamoch na ohodnotenie zobrazujú iba filmy vydané pred rokom 2000.

Z hodnotenia filmov sa generujú štatistiky, ktoré sú prístupne aj používateľovi. Môže si pozrieť, z ktorého žánru má najviac hodnotených filmov ako aj jeho priemerné hodnotenie na žáner.

Následne si používateľ môže prezrieť jeho personalizované odporúčania zoradené podľa odhadovaného hodnotenia. Používateľ má takisto možnosť vybrať si svojich „Movie Buddies“ (ľudí, s ktorými má v pláne pozerat' filmy) a následne si nechať vygenerovať skupinové odporúčanie.

Odporúčanie sa prejaví aj pri vyhľadávaní filmov. Vyhľadávať je možné podľa názvu, žánru, dátumu a tagov pričom výsledky je možné zoradiť podľa odhadovaného hodnotenia. Systém poskytuje aj možnosť pokročilého vyhľadávania, pri ktorom je možné vyhľadávať podľa názvu, režiséra, hercov, žánru jazyka, dátumu vydania, formátu.

Tagovanie Používatelia majú možnosť pridávať filmom tagy - slová alebo frázy, ktoré patria k filmu. Tagy môžu byť označené ako pozitívne, neutrálne alebo negatívne.

Pre každý film systém určí populárne tagy, pomocou ktorých sa dajú filmy vyhľadávať. Používatelia majú možnosť hodnotiť populárne tagy (thumbs up/down) a pozrieť si tag cloud ku každému filmu.

Movie Tuner MovieLens poskytuje zaujímavú funkcionálnosť Movie Tuner, ktorá umožňuje nájsť filmy podobné zadanému s mierne odlišnými charakteristikami.

Toto vyhľadávanie funguje na základe tagov. Systém zobrazí dôležité tagy pre zadaný film a používateľ si zvolí, ktorých tagov by chcel viac a ktorých menej. Napríklad by bolo možné nájsť film podobný The Shining ale s menej násilím a viac nadprirodzenosťou.

Nedostatky MovieLens vyzerá ako typický školský projekt, teda má veľa dobrých myšlienok ale v projekte sú usporiadané chaoticky a nevyvážené. Vidno, že na projekte pracovalo viacero

študentov, z ktorých každý pridal svoju funkcionalitu. Orientácia a prehľad v používateľskom rozhraní potom môže byť problematický. Dizajn stránky je z estetického hľadiska zlý a pôsobí zastaralo.

MovieLens Data Sets MovieLens poskytuje hodnotenia používateľov voľne dostupné na stiahnutie. Je možné stiahnuť 1 000 209 hodnotení približne 3 900 filmov približne 6 040 používateľmi a nie je ich možné použiť na komerčné účely bez súhlasu majiteľov. Informácie o filmoch, používateľoch a hodnotenia sú uložené v súboroch vo formáte podobnom CSV.

2.1.3 RottenTomatoes

Rotten Tomatoes je stránka zameraná na recenzie a informácie o filmoch. Funguje ako agregátor profesionálnych recenzií.

Recenzie sú zhromažďované od rôznych filmových kritikov. Aby sa človek stal kritikom pre stránku musia jeho kritiky dostať určitý počet bodov. Pre každú recenziu sa určí, či je pozitívna (fresh) alebo negatívna (rotten). Systém následne vypočíta percento pozitívnych recenzií voči všetkým recenziám, ktoré funguje ako hodnotenie filmu; ak je nad 60%, film je klasifikovaný ako dobrý. Najpopulárnejší kritici zvyčajne pracujú pre známe noviny alebo časopisy a je možné si zobraziť iba hodnotenie filmu na základe ich kritik. Takýto spôsob určovania hodnotenia filmov je unikátny, väčšina stránok zhromažďuje hodnotenia od bežných používateľov a počíta priemer.

Rotten Tomatoes poskytuje aj hodnotenia pre používateľov, používateľ ohodnotí film od 1-10. Hodnotenie filmu sa vyráta ako v prípade kritikov, čo je opäť neobvyklé - systém vyráta percento ľudí, ktorí ohodnotili film 6 a viac bodmi.

Sociálne funkcionality Stránka obsahuje časť pre používateľov, kde majú možnosť filmy hodnotiť a písať recenzie a komentáre. Prihlasovanie je riešené cez Facebook, vďaka čomu stránka zobrazuje aktivitu kamarátov a pri filmoch ich hodnotenia.

Stránka tiež poskytuje odporúčač filmov, ktorý funguje na základe predchádzajúceho hodnotenia filmov kolaboratívne pomocou hodnotenia ostatných používateľov.

2.1.4 XMLTV to XHTML+RDFa

Tento nástroj umožňuje konvertovať súbor z formátu XMLTV do XHTML súborov. Tieto XHTML súbory tvoria stránku, ktorá umožňuje používateľovi prezrieť si TV program na zoradený a utriedený na základe rôznych kritérií: podľa času a dátumu, žánru a kanála Výstupné XHTML je navyše označené pomocou RDFa.

XMLTV XMLTV je formát založený na XML používaný na uchovávanie TV programu. Tento formát je súčasťou XMLTV Project a je možné ho získavať automaticky pomocou grabberov pre viacero krajín (ale nie Slovensko).

Program je v ňom obsiahnutý z pohľadu používateľa a nie z pohľadu vysielacích staníc. Nedelí sa na stanice a kanály, všetky sú pomiešané dohromady.

XMLTV súbor má záznamy dvoch typov: „channel“, ktoré obsahujú informácie o kanáloch a „program“, ktoré obsahujú informácie a jednotlivých epizódach - názov, popis, kategórie a pod.

2.1.5 TV Browser

Open Source projekt napísaný v jazyku Java. Je navrhnutý tak, aby vyzeral ako klasický televízny program v časopise. Ponúka široký výber televíznych a rádiových programov.

Okno programu je rozdelené na tri záložky:

- **program** - zobrazuje samotný televízny/rádiový program vybraných staníc
- **obľúbené** - do tejto kategórie sa dajú pridávať konkrétne programy (napr. filmy, relácie a pod.) vysielané na rôznych staniach, pre ktoré sa tu zobrazujú dátumy a časy ich vysielania
- **pripomienky** - tu sa dajú nastaviť pripomienky pre vybrané programy, na ktoré nás program upozorní v nejaký stanovený čas pred vysielaním

V zobrazení televízneho programu sa po kliknutí na ľubovoľný film alebo reláciu zobrazí okno, v ktorom sú dostupné všetky informácie o tomto programe (názov, popis, čas vysielania a pod.). Tiež je tu možné tento program ľahko pridať medzi obľúbené položky. Ak sa tento program práve vysielal, je možné sledovať ubehnutý čas vysielania.

TV Browser je jednoduchý program určený len na zobrazovanie televízneho a rádiového programu a túto funkciu splňa veľmi dobre. Výhodou je, že je napísaný v Jave, čo umožňuje použitie programu na viacerých platformách. Jeho funkcionality je možné rozšíriť pomocou veľkého množstva pluginov.

2.1.6 IMDb - Internet Movie Database

IMDb je online databáza informácií o filmoch, TV reláciách, trajleroch, hercoch a takmer o všetkom, čo súvisí s filmovou scénou. Je možné sa prihlásiť pomocou existujúcich účtov na Google, Facebook, Amazon alebo s účtom vytvoreným priamo na stránke IMDb.

Vyhľadávanie Databáza umožňuje vyhľadávanie záznamov podľa rôznych kategórií napr. názov filmu, mená hercov a pod. Výsledkom vyhľadávania môže byť napr. film, relácia, herec, režisér a pod. Po vyhladaní záznamu filmu/relácie máme prístupné rozsiahle informácie: základné informácie, hodnotenia, obrázky, odporúčanie, komentáre, citácie a pod.

Odporúčanie Po prihlásení má užívateľ k dispozícii dva spôsoby, ako môže ovplyvňovať kvalitu odporúčaní, ktoré sú mu poskytované:

- **"watchlist"** - toto je miesto, kam si môže užívateľ pridávať ľubovoľné množstvo svojich obľúbených filmov alebo relácií, ktoré už buď videl alebo sa ich chystá pozerať.

- **hodnotenia** - tu sa nachádza zoznam všetkých filmov a relácií, ktoré užívateľ počas svojej aktivity ohodnotil. Tento zoznam je možné zoradiť a zobraziť rôznymi spôsobmi. Užívateľ tu môže modifikovať svoje hodnotenia, prípadne odstraňovať záznamy, ktoré už nechce mať vo svojich hodnoteniach.

Na hlavnej stránke sa používateľovi zobrazuje odporúčanie. Na vytvorenie odporúčania sa najprv zoberú všetky filmy alebo relácie, ktoré užívateľ buď ohodnotil, alebo si ich pridal do svojho "watchlistu" a porovnajú sa z hodnoteniami, ktoré urobili iní užívatelia. Z tejto analýzy sa vytvorí odporúčanie pre konkrétneho užívateľa a pri každom je viditeľný zoznam filmov alebo relácií, na ktorých bolo toto odporúčanie založené. Veľký vplyv má pozitívne alebo negatívne hodnotenie filmov, kedy sa uprednostňujú filmy s vyšším hodnotením.

Ďalší typ odporúčania sa zobrazuje pri konkrétnych vyhľadaných filmoch. V tomto prípade sa vyhodnocujú užívatelia, ktorí pozitívne ohodnotili tento konkrétny film a zobrazuje sa štatistika ďalších pozitívne ohodnotených filmoch od týchto používateľov.

Databáza IMDb poskytuje časť svojej databázy na stiahnutie. Tieto dáta je možné použiť len pre osobné a nekomerčné účely. IMDb presne nešpecifikuje, aká časť databázy je prístupná. Dáta sú rozdelené do niekoľkých kategórií a každá kategória je skomprimovaná v samostatnom archíve. Dáta sú uložené v textových súboroch vo formáte ".list".

2.1.7 EIDR - The Entertainment Identifier Registry

EIDR je globálny register na identifikáciu filmových a televíznych zdrojov. Je to organizácia, ktorá je zodpovedná za registráciu audiovizuálnych objektov a za poskytovanie potrebných technológií, ktoré umožnia vytváranie a spravovanie metadát k týmto objektom.

EIDR tiež označuje unikátny identifikátor, ktorý tento register používa. Slúži na jedinečné identifikovanie audiovizuálneho objektu a môže byť použité pre fyzický alebo digitálny video objekt.

Štruktúra EIDR registra Register sa skladá z troch hlavných častí:

- **centrálny modul (EIDR Registry)** - slúži na generovanie unikátnych identifikátorov (EIDR) pre jednotlivé audiovizuálne objekty
- **databáza (EIDR Storage)** - tu sú uložené všetky identifikátory spolu s metadátami objektov
- **de-duplikačný modul (Deduplication)** - tento modul zaisťuje aby bolo EIDR pridelené iba raz pre jeden daný objekt

Register má dva typy používateľov:

- **registrátori (Registrants)** - títo užívatelia registrujú nové mediálne objekty rôznych typov a ich metadáta, ktoré ich popisujú

- **vyhľadávači (Lookup Users)** - vyhľadávajú v registroch buď pomocou známeho kľúča alebo podľa rôznych vyhľadávacích kritérií

EIDR poskytuje webové používateľské rozhranie, ktoré využívajú vyhľadávači a web API, ktoré zasa využívajú väčšinou registrátori alebo programátori, pri automatizovanej práci s registrom.

Ako EIDR funguje

1. EIDR registre prijímajú a spracúvajú žiadosti na registrácie od "registrátorov", ktorý môžu byť v tomto prípade napr. vlastníci médií. Registrátor musí zadať objekty spolu s ich hlavnými metadátami, typ objektu a vzťahy k iným objektom.
2. Po podaní registrácie sa automaticky zavolá de-duplikačný modul, ktorý používa množinu prednastavených pravidiel na určenie jedinečnosti objektu. Modul pošle výsledok registru, ktorý môže byť jeden z troch typov: objekt je jedinečný, je duplikát alebo objekt potrebuje manuálnu kontrolu.
3. Ak sa nenájde žiadny duplikát objektu, tak register vygeneruje EIDR číslo pre nový objekt a uloží ho spolu s objektovými metadátami do databázy. Vygenerovanie EIDR by malo zabráť len pár sekúnd.

Použitie vyhľadávania Register poskytuje bezplatný prístup k vyhľadávaniu záznamov vo svojej databáze pomocou webového rozhrania a je možné použiť veľa rôznych kritérií.

Programový prístup do databázy je ale poplatný.

2.1.8 Hunch

Spôsob odporúčania Hunch využíva na odporúčanie takzvaný Taste Graph. Pomocou spojení (hrán) sa pokúša odporúčať, čo máme radi. Skladá sa zo šiestich častí:

1. „**THAY**“ (**Teach Hunch About You**) **otázky priamo na hunch.com**: ide o zaujímavé otázky, ktorých zodpovedaním sa zistia prvé informácie o používateľovi, napríklad akú zmrzlinu má najradšej a podobne
2. **hodnotenie vecí (hviezdičkovanie)**: Taste Graph pomocou informácií, ktoré o používateľovi má, odhaduje ako by sa konkrétna vec mohla používateľovi páčiť, pokiaľ jej používateľ priradí vlastné hodnotenie, tak poskytne systému údaje pre ešte lepšie odporúčanie
3. **facebook likes**: ak si používateľ prepojí svoj Hunch účet s facebookom, tak sa na odporúčanie použijú aj facebook like-y
4. **check-in**: na odporúčanie sa používajú informácie o tom, ktoré miesta používateľ navštívil

5. **sociálne prepojenia:** buď si používateľ prepojí svoj Hunch účet so sociálnou sieťou, alebo priamo na Hunchi je možnosť sledovať osoby, ktoré nás zaujímajú, alebo skôr nás zaujíma obsah, ktorý odporúčajú
6. **otvorená API:** hocikto môže použiť API od Hunchu a uploadnúť dáta, tie sa následne použijú pri odporúčaní

Bez prihlásenia Na stránke sa okrem filmov nachádza aj hudba, knihy, počítačové hry, móda a iné. Bez prihlásenia je na stránke možné prezerať len jednotlivé položky. Odporúčanie ani hodnotenie nie je možné. Zaujímavá je tu vyhľadávanie položiek, kde sa okrem klasického vyhľadávania nachádza aj vyhľadávanie podľa tagov. Najprv si používateľ vyberie niektorý z tagov „prvej úrovne“, napríklad film, kniha, umenie. Podľa toho sa nám zobrazia aktuálne výsledky vyhľadávania. Potom je možné vybrať si tag druhej úrovne, napríklad komédia, dráma či sci-fi. Nájdené položky sa pozmenia tak, aby vyhovovali obidvom tagom. Po druhej úrovni, sa tu už striktné rozdelenie na úrovne nenachádza. Ako nasledujúci tag si je možné vybrať niektorý z druhej úrovne, napríklad ak hľadáme sci-fi komédiu, alebo niektorý z nasledujúcej úrovne, napríklad či sa má jednať o krátky, alebo dlhý film.

V niektorých prípadoch je možné vybrať aj desať tagov, inokedy len jeden. Možnosť pridať tagu závisela od toho, či máme na výber viac ako jednu položku. Pokiaľ áno, tak nám boli poskytnuté tagy, ktoré tieto položky odlišovali, aby sme si dokázali vybrať len jednu. Toto, ale nefungovalo pri knihách, kde bolo možné zadať len tag „knihy“ a žiadne iné nenasledovali. Išlo asi o chybu.

Negatívom systému bolo, že pokiaľ si používateľ chcel vybrať tag šiestej úrovne, alebo presnejšie tag, ktorý sa objaví až neskôr, tak bolo potrebné aby si najprv vybral, niekedy úplne presných, päť tagov, až potom sa objavil ten, ktorý chcel od začiatku. Vzhľadom na túto skutočnosť som bolo prekvapivé, že tagy bolo možné mazať v ľubovoľnom poradí. To znamená, že pokiaľ si používateľ vybral šesť tagov, tak po premazaní bolo možné, aby mu ostal len ten zo „šiestej úrovne“.

Po prihlásení Okrem vytvorenia nového účtu, je na stránke aj možnosť prihlásenia sa cez sociálne siete (facebook, twitter). Následne sme požiadani o zodpovedanie „THAY“ otázok. Sú dve možnosti, ako sa môžeme dostať k odporúčaným veciam. Ako prvé môžeme využiť odporúčanie podľa hodnotenia. Podmienkou na to, aby nám systém začal niečo odporúčať, je ohodnotiť aspoň päť vecí. Druhou možnosťou je prechádzať veci, ktoré naši „priatelia“ odporúčajú.

Pri odporúčaní podľa hodnotenia máme možnosť prechádzať veci podľa tagov. Spôsob je rovnaký ako bolo opísané v časti bez prihlásenia. Okrem toho, ale máme pri každej veci aj odhadované hodnotenie, na ktoré môžeme reagovať. To znamená danú vec ohodnotiť. Tuto bolo veľmi prekvapivé, že odporúčané veci nebolo možné zoradiť podľa odhadovaného hodnotenia, ale len filtrovať podľa tagov. Bolo by lepšie, ak by sa tu možnosť zoradenia nachádzala.

V prípade druhej možnosti sa nám zobrazia, v časovej postupnosti, veci ktoré naši „priatelia“ odporúčajú. Tu je možnosť filtrovať to opäť podľa tagov prvej kategórie (film, hudba,

kniha, atď.), a podľa minimálneho hodnotenia, ktoré daná vec pre musí mať. Minimálne hodnotenie znamená, že sme buď danú vec ohodnotili na vyšší počet hviezdíčiek, alebo je jej odhadované hodnotenie vyššie.

Poslednou funkcionalitou bola možnosť vytvárať vlastné odporúčania. Pokiaľ však nemáme nikoho, kto by nás sledoval, tak to veľmi užitočné nie je.

2.1.9 Zeebox

Zeebox sa, na rozdiel od Hunch-u, zameriava iba na televízne programy. Najväčší rozdiel je však v tom, čo Zeebox poskytuje. Neposkytuje žiadne špeciálne odporúčanie. Jedine odporúčanie je podľa počtu tweetov, počte ľudí ktorí pri sledovaní daného programu používajú zeebox a podľa toho čo pozerajú vaši priatelia. Zeebox je o tom, aby ste si televízny program vychutnali viac ako obvykle.

Funkcionalita Pre lepší zážitok zo sledovania programu, ponúka Zeebox niekoľko funkcií.

Prvou je odporúčanie. Ako už bolo spomenuté, tak nejde o klasické odporúčanie, kde sa zohľadňujú záujmy a podobné veci, ide len o jednoduché odporúčanie na základe záujmu o daný program, alebo podľa toho, aký program pozerajú vaši priatelia.

Zeebox v sebe obsahuje aj chat napojený na sociálne siete. Pokiaľ s vami priatelia pozerajú rovnaký program, tak môžete s nimi chatovať. Ide o spôsob, akým chce Zeebox vytvoriť sociálnu televíziu. Ak s vami priatelia daný program nepozerajú, tak ich môžete pozvať.

Aj ďalšia funkcionalita sa týka sociálnych sietí. Pomocou Zeebox-u si môžete pozeráť ako ľudia o danom televíznom programe tweetujú, prípadne sa do tweetovania zapojiť.

Najdôležitejšou funkcionalitou je získanie informácií o prebiehajúcim televíznom programe. Počas vysielania sa v reálnom čase objavujú tagy o danom programe, napríklad herci, ktorí v ňom hrajú, soundtrack, alebo rôzne iné. Dokonca sa môže objaviť aj tag s názvom chladnička, ak sa v programe chladnička spomenie. Po kliknutí na tag sa objavia rozširujúce informácie o tom, na čo sme klikli. Takto môžeme okamžite zistiť viac o hercovi, ktorý nás zaujal, názov soundtracku, ktorý sme počuli, prípadne informácie o chladničke. Zdroj informácií býva väčšinou wikipedia. Nachádzajú sa tu aj články z rôznych iných stránok, tweety, alebo odkazy na externé zdroje.

Ďalšou zaujímavou funkcionalitou je možnosť stiahnutia produktu. Pokiaľ k danému programu existujú produkty na stiahnutia, prípadne nákup, ako napríklad iná epizóda, hudba, alebo mobilná aplikácia, tak máme možnosť daný produkt priamo stiahnuť a nemusíme ho hľadať. V prípade, že by nás zajal produkt z reklamy, tak môžeme kliknúť na jeho tag a následne pokračovať v jeho zakúpení.

Poslednou, najnovšou funkcionalitou, je možnosť prepojenia s celebrityami. Niektoré osoby majú na Zeeboxe status celebrity. Pokiaľ daná osoba tweetne o nejakom programe, ktorý pozerá, tak sa to ľuďom zobrazí. Vďaka tomuto môžu ľudia zistiť čo ich obľúbené celebrity pozerajú, a pozeráť to tiež. Celebrity taktiež môže pre program vytvoriť skupinový chat a takýmto spôsobom sledovať program so svojimi fanúšikmi a zdieľať svoje názory.

Prechádzanie programov Zeebox má ako základný dizajn vytvorený televízny program pre podporované stanice. V programe sa dajú nájsť programy jeden deň dozadu, aj so všetkými tagmi, a taktiež programy až na týždeň dopredu. Keďže sa tagy pridávajú v reálnom čase, tak skončené programy obsahujú všetky tagy a programy, ktoré ešte nezačali neobsahujú žiadne.

Zhodnotenie Najlepšou funkcionalitou Zeeboxu sú tagy, na základe ktorých si vieme rýchlo zistiť informácie o veci, ktorá nás v programe zaujala. Ostatná funkcionalita je tiež dobrý nápad, ktorý sa môže hodiť. Jediná funkcionalita, ktorá nemusí mať vysoký ohlas, je možnosť chatovania. Pokiaľ pozeráme program, ktorý nás zaujíma, tak nemáme moc času a chuť zaoberať sa chatovaním, ale radšej sa sústredíme na sledovanie programu. Tak či onak ide len o integráciu facebook chatu, ktorý si môžeme bez problémov otvoriť priamo na facebook stránke.

2.1.10 Sémantické vyhľadávanie na webe použitím ontologických axiém

Ide o vedecký článok, v ktorom autori rozoberajú možnosť použitia sémantického vyhľadávania, s použitím ontologických axiém. Ako autori tvrdia, tak v súčasnosti ide o horúcu tému, ktorou sa zaoberajú mnohé inštitúcie.

Sú dve možnosti ako daný problém vyriešiť. Prvou možnosťou, ktorou sa zaoberá väčšie množstvo výskumníkov, je vytvoriť nový vyhľadávací systém pre sémantický web. Druhá možnosť, ktorá podľa autorov nie je skoro vôbec preskúmaná, je vytvoriť do už existujúcich vyhľadávacích systémov časť, ktorá im pridá istú sémantiku. Práve pre túto druhú možnosť navrhli autori riešenie.

Ontologické znalosti môžeme, vzhľadom na to odkiaľ pochádzajú rozdeliť na explicitne definované expertmi, alebo automaticky extrahované z webu. Po istom čase autori očakávajú, že sa budú všetky znalosti získavať automaticky.

Navrhovaný systém Prvou časťou systému sú ontológie. Ide o axiémy, rovnaké ako vo výrokovkej logike. Napríklad axiém hovoriaca: „konferenčné príspevky sú články“.

Ďalšou časťou systému sú anotácie. Tie začínajú byť, s nástupom Web 2.0, priamo vytvorené používateľmi, alebo sa dajú získavať automatizovanými procesmi. Napríklad na stránke profesora, ktorý napísal konferenčný príspevok s názvom „Ontológia“ je možné vytvoriť anotáciu pre príspevok i:

$$Ai = \{KonferenčnýPríspevok(i), titul(i, „Ontológia“)\}$$

Následne s použitím „Inference engine“ dokáže systém spojením anotácii a ontologických axiém zistiť, že konferenčný príspevok i je článok. Tieto spojenia sa potom prevedú do HTML kódu a z neho do RDF formátu.

Poslednou pridanou časťou systému je „Query Evaluator“. Ten rozloží sémantický dopyt na menšie časti, ktoré následne odošle obyčajnému vyhľadávaciemu systému. Po obdržaní výsledku pre každú časť, tieto výsledky vyhodnotí a konečný výsledok zobrazí používateľovi.

Príklad použitia Systém bol otestovaný v IMDb na vzorke viac ako 60 000 filmov a hercov. Ako prvé sa s použitím systému SCRAP vytvorili anotácie pre každý film a herca. SCRAP extrahuje informácie z HTML stránok a uchová ich v XML súbore. Na extrakciu sa používajú extrakčné pravidlá a extrakčná schéma. V extrakčnom pravidle sa uvádza zdroj, odkiaľ má SCRAP čerpať, teda „ČO?“. V extrakčnej schéme sa uvádza forma výstupného XML súboru, to znamená „AKO?“. Keďže bola extrakčná schéma vytvorená na základ IMDb ontológie a štruktúra každej stránky filmu (autora) bola na IMDb rovnaká, tak vynaložené ľudské úsilie na získanie anotácii bolo len vytvorenie extrakčných pravidiel a jednotnej schémy (pre jednotný typ, teda film mal inú schému ako autor).

Potom bolo každé XML automaticky prerobené do presnej anotácie, ktorá bola použitá pri sémantickom vyhľadávaní. Keď prišiel sémantický dopyt, tak sa preformuloval do jazyka výrokovej logiky. Podľa obsahu anotácie sa dokázalo určiť, či je pre danú anotáciu výrok pravdivý, alebo nie. Výsledkom testu bolo, že väčšina dopytov mala čas behu pod jednu sekundu, ale našli sa aj také, ktoré mali čas behu šesť sekúnd. Úspešnosť výstupu bola vždy 100%.

2.1.11 LinkedMDB

Jedná sa o otvorenú databázu, ktorá uchováva sémantické dáta v linkovanom formáte. V databáze sa nachádza viac než 85 000 entít predstavujúcich filmy. Dokopy sa v databáze nachádza cez 500 000 entít (filmy, herci, producenti, atď.). Okrem toho je LinkedMDB napojená aj na dáta v Linking Open Data projekte. Prepojenia sú vygenerované nástrojom ODDLInker, pričom ich je viac než 160 000. Príkladmi z Linking Open Data projektu, na ktoré je LinkedMDB napojená sú DBpedia, YAGO, alebo GeoNames. Nakoniec sa v databáze nachádzajú linky na webové stránky zobrazujúce jednotlivé filmy, ako IMDb, Rotten Tomatoes, alebo Freebase.

Entity sú uložené vo RDF formáte, pričom napríklad film má nalinkovaných hercov, hrdinov, režiséra, hudbu, alebo odkazy na Linking Open Data, či webovú stránku, ktorá sa zaoberá filmami, ako IMDb.

K údajom v Linking Open Data sa pristupuje pomocou jazyka OWL a k údajom v externých stránkach pomocou štandardu FOAF. K dátam sa dá pristupovať pomocou obyčajného webového prehliadača, sémantického webového prehliadača, alebo SPARQL klienta.

2.1.12 Freebase

Freebase predstavuje verejné úložisko dát usporiadané do grafu. Tieto dáta majú rozličný charakter, jedná sa o zväčša o entity reálneho sveta. Pričom tieto dáta pochádzajú z viacerých zdrojov a sú vzájomne prepojené. Hlavným zdrojom dát pre Freebase je Wikipédia a medzi ďalšie patrí napríklad Standfordská univerzitná knižnica, EDGAR, WordNet, TVRage a ISFDB. Projekt je kolaboratívny a dáta sú editované používateľmi veľmi podobne, ako je to vo Wikipédii.

Celkový počet entít sa predstavuje okolo 23 miliónov. Počet filmov v databáze sa pohybuje okolo 191 tisíc a počet hercov je 207 tisíc. Nachádzajú sa tu tiež entity ako žáner, hodnotenie, lokalita či pieseň z filmu.

Čo sa týka informácií o filmoch, väčšina informácií je prebratá z Wikipédie. Výhodou Freebase je, že dáta sú identifikované prostredníctvom ID a dajú sa získavať pomocou MQL (Metaweb Query Language). Freebase zároveň poskytuje API pre získavanie dát, počet volaní Freebase API je limitovaný na 100k za 24 hodín.

2.1.13 Projekt Online TV buzz

Jedná sa o diplomovú prácu z univerzity v Amsterdame. Cieľom je analýza možností sociálnych sietí, ktoré ovplyvnia vytvorenie dobrého predpovedajúceho modelu pre popularitu jednotlivých TV shows. Pritom sa skúma najmä súvislosť medzi popularitou show a hosťami v nej.

Dataset pre túto prácu tvorí 26.690 epizód so živými hercami, ktoré boli extrahované z EPG (Electronic Program Guide). Tieto epizódy obsahujú informácie o čase vysielania, dĺžke, titulku, obsahu, typu, žánru, vystupujúcich ľuďoch a pod. Pre meranie popularity jednotlivých shows si zvolili počet tweetov, ktoré obsahujú meno danej show. Výsledky môžu byť dôležité pre producentov - ktorí hostia vyvolajú najväčšiu popularitu, kedy je najlepší čas pre reklamy a pre potreby odporúčania.

Projekt bol vyvinutý na základe technológií pre sémantický web (RDF, SPARQL, OWL a pod.). Jednotlivé dáta sú uložené v ontológii SEM (Simple Event Model). Model tvoria dve siete:

1. EPG sieť: epizódy, žánre, hostia

- Jednotlivé TV programy sú pospájané nielen na základe rovnakého žánru, ale aj na základe spoločných hostí
- Ako meradlo popularity sa berie do úvahy stupeň jednotlivých vrcholov (počet hrán s inými vrcholmi), teda koľkokrát sa daní hostia a žánre vyskytovali v epizódach v minulosti

2. Sociálna sieť: hostia, pospájaní podľa toho ako sa vyskytovali v rovnakých epizódach

- Ľudia sú spojení podľa toho, či účinkovali v rovnakej televíznej epizóde, každá hrana je pritom ohodnotená číslom - koľkokrát sa daná spolupráca vyskytovala
- V tomto prípade sa neberie do úvahy kvalita prepojení a všetci hostia sú považovaní za rovnocenných, teda čím viac spojení tým lepšie bez ohľadu na kvalitu
- Neberie sa do úvahy ani rola jednotlivých ľudí v epizóde (moderátor vs. hostia)
- Na hostí bola použitá idea PageRanku a cieľom bolo zistiť či prítomnosť „viac prepojených“ ľudí má vplyv na popularitu epizódy

Dataset bol rozdelený na dve časti: populárnu a nepopulárnu. Pre toto rozdelenie bol použitý Bayesov algoritmus. Pričom sa použilo rozdelenie dát na trénovaciu a testovaciu sadu. Aby sa minimalizovali chyby použila sa metóda 10 Fold Cross-validation.

Tweety boli získané pomocou TOPSY API. Počet tweetov za minútu bol normalizovaný cez dĺžku jednotlivých epizód, aby vyjadroval ich popularitu.

Výsledky Populárne aj nepopulárne epizódy sú približne rovnako rozdelené v jednotlivých typoch programov (šport, noviny a pod).

- Populárne epizódy majú nižší stupeň vrcholu ako nepopulárne, kvôli tomu, že neberieme do úvahy jednotlivé roly osôb môžu byť moderátori ohodnotení lepšie, hoci majú nižšiu popularitu
- Populárne epizódy majú hostí s nižšou betweenness - preto to nie je dobré meradlo pre popularitu
- Populárne epizódy majú ale hostí s vyšším PageRankom - ten sa ukazuje ako lepší ukazovateľ popularity

Najlepšie výsledky - teda výsledky blížiac sa k predpokladom sú pre typ programu zvaný „udalosť“. Inak sa ukazuje nízky vplyv hostí na popularitu epizód. Udalosť vyvoláva šum (vysoký počet tweetov), hostia sú pre epizódy naozaj dôležití a titul tohto programu sa dobre rozoznáva v jednotlivých tweetoch. Výsledky pre tento typ programu sú zhrnuté v závere.

Zhodnotenie Na základe výsledkov sa nezistil vzájomný vzťah medzi popularitou epizódy a popularitou hostí. Tento záver pravdepodobne vznikol v dôsledku toho, že sa nebrali do úvahy jednotlivé roly v epizódach a niektoré názvy mohli byť zle identifikované v tweetoch.

Pre jeden typ programu tzv. udalosť sa výsledky blížili k predpokladom. Jedná sa o udalosti, ktoré sa vyskytujú niekoľkokrát do roka. V týchto typoch programu sa ukázal vzťah medzi stupňom vrcholu hostí a popularitou epizódy. Najlepšími ukazovateľmi sú priemer zo stupňov všetkých vrcholov (hostí), variácia stupňov vrcholov a hostí s druhým najvyšším stupňom vrcholu.

Vysoký priemer indikuje prítomnosť moderátora a takmer žiadnych hostí. Nízky priemer zase indikuje prítomnosť viacerých hostí s nízkym stupňom vrcholu. To poukazuje na opačnú situáciu ohľadne popularity v reálnom svete a v tejto sieti.

Nízka variácia indikuje vysokú jednotnosť. Populárne epizódy majú nižšiu variáciu ako nepopulárne. Opäť v dôsledku rolí moderátor - hostia.

Druhý najvyšší stupeň vrcholu je dobrou charakteristikou opäť v dôsledku role moderátora.

Schema Schema.org je súbor pravidiel, akými sa môže označiť obsah stránok. Tento spôsob je navyše kompatibilný s prácou najväčších vyhľadávačov - Google, Microsoft, Yandex a Yahoo. Schema.org má pomôcť vyhľadávateľovi pochopiť obsah stránok.

Na adrese <http://schema.rdfs.org/> sa nachádzajú tutoriály a príklady ohľadne používania a spracovania dát označených pomocou Schema.org. Tieto dáta sa dajú produkovať a konzumovať viacerými spôsobmi prostredníctvom knižníc a nástrojov. Schema.org podporuje prácu s programovacími jazykmi JavaScript, Php, Ruby, Python, Perl a Java.

Microdáta predstavujú HTML5 tagy, ktoré pomáhajú vyhľadávateľovi pochopiť obsah stránky, pričom dizajn stránky zostáva nezmenený.

Schema.org opisuje množstvo rôznych typov položiek. Tieto položky majú vlastnosti, pričom položky môžu byť viac alebo menej špecifické. Najširším pojmom je vec (Thing), ktorá má

štyri vlastnosti: meno, url, obrázok a popis. Viac špecifické položky majú rovnaké vlastnosti ako tie menej špecifické, prípadne ich rozširujú. Popis každej položky a jej vlastností nájdeme na adrese <http://schema.org/Položka>, napr. <http://schema.org/Thing>. Každá vlastnosť má nejaký očakávaný typ (textový reťazec, URL, iná položka..).

Google navyše poskytuje nástroj na testovanie vytvorených microdát.

2.1.14 Yummly

Yummly je stránka určená pre vyhľadávanie receptov. Sľubuje najväčšiu množinu receptov na svete a to z celého sveta. Ako jediná používa pre vyhľadávanie receptov chuť jedla. Berie do úvahy aj diéty, alergie a preferencie používateľov k určitým surovinám.

Jedná sa o fazetové vyhľadávanie, pričom filtre sú rôzneho charakteru. Je umožnené vyhľadávanie nielen podľa ingrediencií, ale tiež chute jedla, sviatkov, národných kuchyní, nutričných hodnôt, alergií, druhu jedál či značky ingrediencií. Po vybratí konkrétneho receptu sa síce zobrazia podrobné informácie, ale pre prípravu jedla je používateľ presmerovaný na stránku, z ktorej recept pochádza. To je spôsobené tým, že recepty pochádzajú z viacerých samostatných zdrojov. Yummly čerpá z ôsmich zdrojov ako: AllRecipes, Epicurious, Food Network, Chow, Martha Stewart. Recepty pridávajú aj známy blogeri o jedle napr. Serious Eats a The Pioneer Woman.

Zozbieraním dát z viacerých veľkých projektov uľahčilo Yummly prácu používateľom, ktorí nemusia navštíviť osem rôznych zdrojov. Jednotlivé recepty ponúkajú možnosť prepočítania ingrediencií podľa počtu porcií. Zdroje sú navyše overené a kvalitné. Zároveň Yummly má sociálne prvky, používatelia môžu zdieľať, čo práve varia alebo niektoré recepty. Oblúbené recepty si môžu používatelia uložiť a organizovať. V rámci alergií existujú skupiny, nie je potrebné pridávať každú surovinu zvlášť.

Každý recept je de-komponovaný na jednotlivé suroviny a tak Yummly vyhodnotí, či je vhodný pre jednotlivé druhy alergií a do akej kuchyne ho môže zaradiť, zároveň sa na základe surovín hodnotí aj chuť receptu. Pri určovaní chute berie do úvahy výskum a rôzne stupnice, napríklad páliivosť jedál je postavená na základe Scovilleho stupnice.

Yummly poskytuje aj odporúčanie, to je pravdepodobne postavené na receptoch, ktoré si používateľ ukladá.

2.2 Vytvorenie prezentačnej stránky

2.2.1 Úloha

Vytvorte prezentačný web pre projekt odporúčania inteligentnej televízie. Prezentačná stránka musí spĺňať všetky štandardy modernej webovej stránky a musí zahŕňať minimálny obsah podľa požiadaviek.

Minimálny obsah stránky:

- názov témy
- riešitelia a základné informácie o nich (napr. z ponuky)
- plán projektu (na semester)
- aktuálny stav plnenia plánu (t.j., úlohy, ktoré vyplynuli zo stretnutí, ich plnenie a vzťah k plánu)
- roly jednotlivých členov tímu (aj dočasné)
- odkazy na doteraz vypracovanú dokumentáciu vrátane záznamov zo stretnutí (záznamy zo stretnutí budú priamo čitateľné na webe, t.j. najlepšie vo formáte pdf, prípadne HTML)
- všetko zaujímavé v súvislosti s projektom a postupom prác na projekte, napr. odkazy na iné zdroje súvisiace s témou projektu

2.2.2 Analýza

Pred konštruovaním návrhu prezentačnej stránky sme si dôkladne zanalyzovali minuloročné webové prezentácie ostatných tímov. Poukázali sme na slabé stránky a takisto aj na silné stránky analyzovaných riešení.

2.2.3 Návrh

Prezentačnú stránku sme si rozdelili na štyri hlavné sekcie:

- **Titulná stránka** - V tejto časti sa nachádzajú všetky dôležité informácie o projekte TeleVido, rozdelené do časti: Čo je TeleVido, Prečo robíme TeleVido, Ako funguje TeleVido a Kto robí TeleVido. Každá časť je podrobne opísaná so zrozumiteľným a triviálnym textom.
- **Náš tím** - Sekcia o našom tíme začína s efektným zobrazovaním výrokov jednotlivých členov tímu. Následne je zobrazený zoznam všetkých členov tímu TeleVido s uvedením mena, profilovej fotografie, manažérskej pozície a stručného opisu osoby.
- **Dokumenty** - V sekcii dokumenty sú zobrazené vypracované dokumentácie vrátane záznamov zo stretnutí. Zároveň sú v nej zobrazené aj obrázky a fotografie súvisiace s projektom TeleVido.

- **Kontakt** - V sekcii Kontakt sú uvedené všetky dôležité kontaktné údaje pre projekt TeleVido. Zároveň sa v nej nachádza prehľadná mapa so zobrazením primárneho sídla členov projektu.

2.2.4 Implementácia

Prezentačná stránka bola implementovaná v HTML5 a CSS3 s využitím programovacieho jazyka PHP. Pre vykreslenie rôznych grafických animácií sme využili JavaScript a jQuery.

Web prezentačnej stránky je k dispozícii na webovej adrese: <http://team01-12.ucebne.fkit.stuba.sk/>



ČO JE TELEVIDO?

TeleVido je inteligentná televízia, ktorá Vám pomôže vybrať si ten najlepší program na sledovanie. Aplikácia Vám odporučí aktuálny program v TV, ako aj vaše žiadané seriály a filmy. TeleVido je skvelý spoločník pre každého diváka.

Téma	Odporúčanie pre inteligentnú TV	Tím	TeleVido, tím #1
Škola	FIIT STUBA	E-mail	televido@googlegroups.com
Ak. rok	2012/2013	Web	televido.zones.sk

PREČO ROBIEME TELEVIDO?

V súčasnosti je televízia jedným z najhlavnejších masmédií na svete. Zasahuje do života mnohých ľudí a najmä do životov detí a mládeže. Ľudia dokážu pri televízii stráviť nespočetné množstvo času a často ich môže pozitívne alebo negatívne ovplyvňovať. Používatelia si pod vplyvom veľkého počtu programov – filmov, seriálov a relácií, veľakrát nevedia vybrať adekvátnu možnosť. Ďalším problémom v dnešnom svete je nedostatok informácií o vybraných programoch. Cieľom projektu je priniesť do multimediálneho sveta platformu, ktorá by používateľom odporúčala vyhovujúci obsah. Projekt pomôže ľuďom, aby sa snažili tráviť svoj čas zmysluplne a pritom sledovali všetky programy, ktoré sú pre nich podstatné a dôležité.

Vytvorenie platformy pokrývajúcej potreby odporúčania multimediálneho obsahu s využitím znalostí, kreativity a nadšenia v spojení s najmodernejšími informačnými technológiami je pre nás veľkou výzvou. Sme presvedčení, že navrhnutý projekt má obrovský potenciál pre dnešnú spoločnosť a jeho realizácia by mnohým ľuďom zlepšila a zefektívnila spôsob života.



AKO FUNGUJE TELEVIDO?

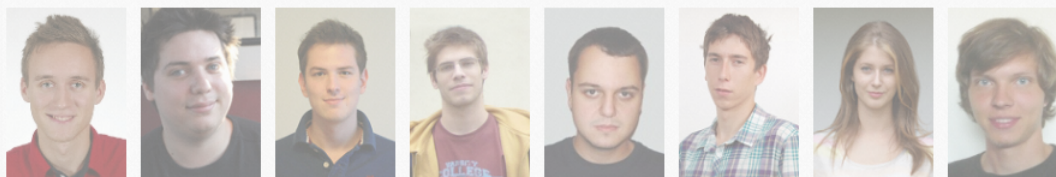


Naša inteligentná televízia vždy vie odporučiť správny obsah a vytvára pre diváka ten najlepší program. Každý si môže v TeleVido nájsť to, čo potrebuje. Najdôležitejšou súčasťou TeleVido je webová služba, ktorá na základe vytvoreného modelu používateľa odporúča vhodný obsah – filmy, seriály, relácie. Táto webová služba môže fungovať ako samostatný produkt, ktorý ponúkame ako podporný nástroj do multimediálnych zariadení.

Pre používateľov je tu webová aplikácia, ktorá sleduje aktivity a záujmy používateľa na internete. Na základe toho bude pre každého diváka vyberať to, čo by si mal v TV pozrieť. TeleVido ponúka aj mobilnú aplikáciu, ktorá bude sledovať záujmy používateľa a jeho aktivity, ktoré vykonáva s mobilným zariadením. Takto bude mať divák ten najlepší TV program vždy poruke.

KTO ROBÍ TELEVIDO?

Projekt TeleVido robia siedmi študenti inžinierskeho štúdia pod vedením svojho mentora – študenta doktorandského štúdia Fakulty informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave.



Obr. 1: Titulná stránka prezentačnej stránky

NÁŠ TÍM

Projekt TeleVido robia siedmi študenti inžinierskeho štúdia pod vedením svojho mentora – študenta doktorandského štúdia Fakulty informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave.

“Mýliť sa je ľudské, ukladať chybné verzie programátorské.

Bc. Jakub Kříž – manažér podpory vývoja

ČLENOVIA



Bc. Ľuboš Demovič – manažér rizika

Navštevoval matematické Gymnázium Grosseingová 18 v Bratislave, ktoré ho výborne pripravilo na vysokú školu, obzvlášť po stránke z matematiky a informatiky. Na FIIT STU v odbore Informatika úspešne obhájil svoju bakalársku prácu, na ktorej v rámci medzinárodnej súťaže Imagine Cup spoločne s tímom viacerých študentov spolupracoval na vytvorení inovatívnej aplikácie. Projekt vyhral slovenské finále a postúpil na svetové finále do Austrálie. Má rád všetko okolo Webu a internetový marketing. Vyzná sa predovšetkým v moderných webových technológiách, pomocou ktorých vytvoril viacero rozsiahlých projektov. Zaujíma sa aj o optimalizáciu pre vyhľadávače, sémantiku webu, copywriting, dizajn a tvorbu efektívnych algoritmov. Vo voľnom čase pravidelne hrá futbal, behá a bicykluje.



Bc. Eduard Fritscher – manažér rozvrhu a plánovania

Absolvent bakalárskeho štúdia FIIT STU v odbore Informatika. Počas štúdia pracoval na vylepšovaní svojich programátorských schopností. Úspešným ukončením štúdia bola obhajoba bakalárskeho projektu s názvom Odporúčanie študijného obsahu na základe kolaboratívneho filtrovania. Projekt bol zameraný na sémantický web a personalizované odporúčanie. Počas štúdia si osvojil základné techniky tvorby softvéru a rôzne programátorské zručnosti a počas riešenia bakalárky si osvojil vedomosti o odporúčaniach. Aktuálne pokračuje v štúdiu na FIIT STU v odbore Softvérové inžinierstvo. Má praktické skúsenosti s jazykmi UML, C, Java, Ruby, ABAP, ďalej so sémantickým webom, analýzou a návrhom softvéru, manažmentom projektov, programovaním veľkých informačných systémov ako SAP a databázovým systémom MySQL.



Bc. Jakub Kříž – manažér podpory vývoja

Bakalárske štúdium absolvoval na FIIT STU. Štúdium ukončil bakalárskou prácou s názvom Získavanie implicitnej spätnej väzby pre potreby webového vyhľadávača, v ktorej sa zaoberal extrahovaním kľúčových slov na základe aktivity používateľa vo webovom prehliadači. Túto prácu prezentoval aj na konferencii IIT.SRC, na ktorej získal ocenenie ACM Slovakia Chapter. Zapájal sa aj do programátorskej súťaže ACM ICPC, v ktorej sa zúčastnil regionálneho kola. Má skúsenosti s tvorbou webových aplikácií v jazykoch PHP a Ruby s frameworkom Ruby on Rails. V priebehu štúdia nadobudol skúsenosti s jazykmi C, C++, Java, JavaScript, UML, Lisp, Prolog a databázovým systémom MySQL.



Bc. Ondrej Kuzmík – manažér kvality

Absolvent bakalárskeho štúdia na FIIT STU. Bakalárska práca s názvom Systém pre organizáciu programátorských maratónov, mu pomohla s bližším oboznámením sa s problematikou bezpečnosti a vytvárania systému pre organizáciu súťaží. Okrem školských projektov sa zapájal aj do programátorských súťaží ACM, TopCoder a spoluorganizoval stredoškolskú súťaž ProFIIT. Aktuálne pokračuje v štúdiu na FIIT STU v odbore softvérové inžinierstvo. Ovláda jazyky UML, C, C++, Ruby, PHP a databázový systém MySQL.



Bc. Ondrej Proksa – vedúci tímu

Absolvent bakalárskeho štúdia FIIT STU v odbore Informatika. Počas štúdia sa venoval programátorským súťažiam ACM a tiež organizovaniu súťaže ProFIIT. V akademickom roku 2011/2012 bol súčasťou výskumnej orientácie, ktorej výsledkom bol projekt NašeObce.sk. Úspešným ukončením štúdia bola obhajoba bakalárskeho projektu s názvom Automatizované čistenie verejných dát, kde si osvojil problematiku strojového učenia a verejných dát. Na bakalárskej práci spolupracoval s Alianciou Fair Play a za prácu získal ocenenie dekana. Má praktické skúsenosti s jazykmi UML, C, C++, Java, Ruby, C# a databázami MySQL, PostgreSQL, SQLite. Predovšetkým sa zaujíma o Web a webové technológie. Aktuálne pokračuje v štúdiu na FIIT STU v odbore Softvérové inžinierstvo.



Bc. Martin Prokop – manažér dokumentovania

Absolvoval bakalárske štúdium na Univerzite sv. Cyrila a Metoda v Trnave odbor Informatika. Témou jeho bakalárskej práce bola Vizualizácia grafových algoritmov, vďaka čomu získal praktické znalosti Microsoft .NET technológie a objektovo orientovaného návrhu a programovania. Počas štúdia pracoval s programovacími jazykmi C, C++, Java a databázovými systémami MySQL a Oracle. Ďalej má skúsenosti s programovaním paralelných programov s využitím knižnice OpenMP a OpenMPI.



Bc. Diana Vandlíková – manažér monitorovania projektu

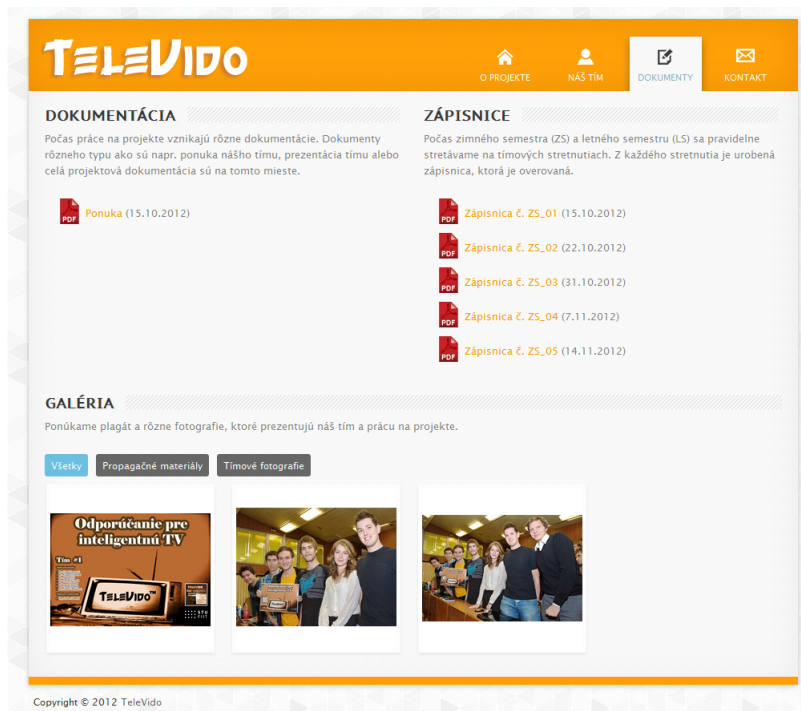
Vyštuovala bakalárske štúdium na FIIT STU v odbore Informatika. Štúdium ukončila bakalárskou prácou Podpora akademického vzdelávania, kde skúmala možnosti automatického hodnotenia akademických prác. Počas štúdia získala skúsenosti s prácou v programovacích jazykoch C, C#, Java, UML, frameworkom .NET a databázovým systémom MySQL. Vo vzdelávaní pokračuje na odbore Softvérové inžinierstvo na FIIT STU, kde zbiera nové vedomosti a skúsenosti.



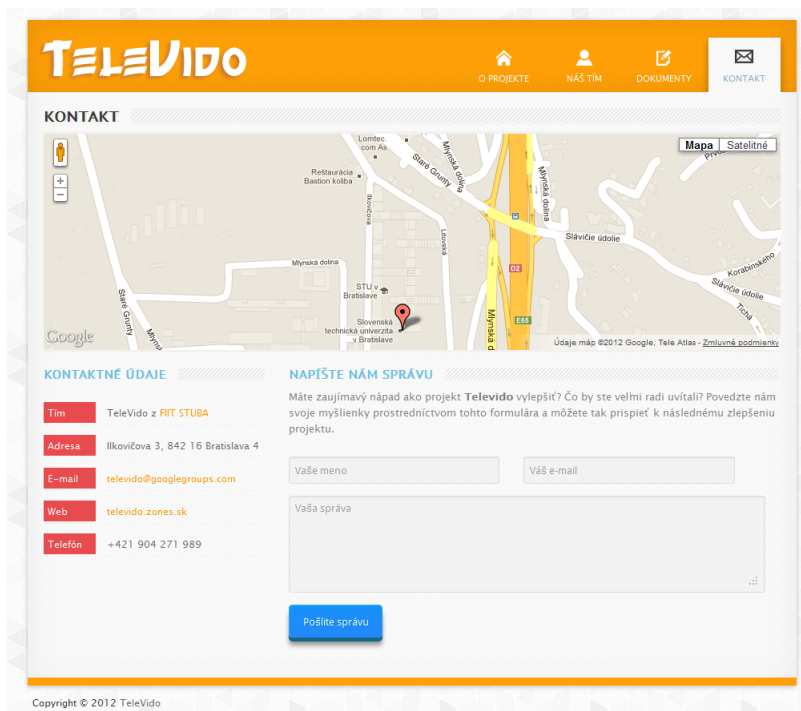
Ing. Dušan Zeleník – mentor tímu

Navštevoval Gymnázium Juraja Hronca v Bratislave. Absolvent inžinierskeho štúdia na FIIT STU v odbore Softvérové inžinierstvo. Spolu so svojím tímom sa zúčastnil medzinárodnej súťaže Imagine Cup, kde vo svetovom finále v Paríži vybojovali striebro. Inovatívna aplikácia sa venuje manažovaniu spotreby energie. Jeho najväčšia vášeň je Ruby on Rails. Medzi jeho najúspešnejšie projekty patrí dotankoch.sk, ktorý spracúva a analyzuje dáta z NRSR. Tohto času dokončuje doktorandské štúdium na FIIT STU. V rámci svojho štúdia sa zaoberá vzormi správania používateľa, dolovaniu dát na internete a odporúčaniami.

Obr. 2: Sekcia predstavenia tímu na prezentačnej stránke



Obr. 3: Sekcia dokumenty na prezentačnej stránke



Obr. 4: Sekcia kontakt na prezentačnej stránke

3 2. Šprint

3.1 Analýza štruktúry entít. Ontológia

3.1.1 Úloha

Vytvorte návrh grafovej databázy pre potreby odporúčania filmov, seriálov a relácií. Grafová databáza bude reprezentovať vzťahy medzi základnými entitami, podrobnosti o jednotlivých entitách sa budú nachádzať v SQL databáze.

3.1.2 Analýza

Ako grafovú databázu sme si zvolili Neo4j. Databáza Neo4j reprezentuje informácie ako trip-lety. Teda ako uzly a vzťahy medzi nimi, každý vzťah má začiatkový a koncový uzol. Pričom uzly aj vzťahy môžu mať vlastnosti. Túto štruktúru reprezentuje Obr. 5. Vlastnosti majú kľúč a hodnotu. Hodnoty majú určitý typ. Tieto typy sú zhodné s primitívnymi typmi, tak ako ich poznáme z programovacích jazykov (boolean, byte, short, long, int, float, double, char, String).

Obr. 5: Základná štruktúra neo4j

3.1.3 Návrh

Kľúčovými entitami sú entity film a používateľ. Zo všetkých podrobností o filme bude grafová databáza obsahovať tie, ktoré určujú dôvod, prečo daný film používateľa zaujíma.

Film vytvárajú rôzne osoby, ktoré sú často kľúčové pre záujem používateľa. Takými osobami sú herci, režiséri, producenti a scenáristi. Používatelia sa pri výbere filmov často riadia žánrom, dátumom vzniku alebo rozpočtom filmu. Vzhľadom na to, že databáza obsahuje nielen filmy, ale aj seriály a relácie, je potrebné pridať entitu typ. Či používateľ preferuje určitý typ filmov nám pomôže zistiť entita tag, ktorá obsahuje niečo ako kľúčové slová pre film napr. „upír“, „Vianoce“, ale môžu sa tam vyskytovať aj typ filmov ako filmy vhodné na „rande“, „rodinné stretnutie“ a pod.

3.1.4 Implementácia

Po zohľadnení možností, ktoré nám ponúka Neo4j sme vytvorili nasledovné typy entít/uzlov:

- **User** - používateľ - reprezentovaný cez (long)ID, môže mať vzťah k ľubovoľnej inej entite prostredníctvom vzťahu likes, napr. user - likes - the_notebook
- **Movie** - film - reprezentovaný cez (long) ID, má vzťahy k nasledovným entitám:
 - *Movie - type - Movie_type*, napr. skyfall - type - film
 - *Movie - budget - Amount*, napr. skyfall - budget - high
 - *Movie - movie_genre - genre*, napr. pride_and_prejudice - genre - romance
 - *Movie - decade - decade*, napr. pride_and_prejudice - decade - 2000s

- *Movie - tag - word*, napr. *love_actually - tag - christmas*
- *Person: acts_in / writes / produces / directs* - *Movie: johnny_depp - acts_in - ed_wood*

- **Type** - typ filmu - reprezentovaný cez (long) ID, má vzťah k entite film, vid' entita film. Reprezentuje to, či sa jedná o film, seriál alebo reláciu
- **Genre** - žáner - reprezentovaný cez (long) ID, má vzťah k entite film, vid' entita film
- **Decade** - dekáda - reprezentovaná cez (long) ID, má vzťah entite film, vid' entita film
- **Word** - slovo/slovné spojenie - vyjadruje slovo alebo slová kľúčové pre film, reprezentované cez (long) ID, má vzťah k entite film
- **Person** - osoba - osoba, ktorá hrá, režíruje, je producentom alebo scenáristom filmu. Nereprezentuje používateľa. Má vzťah k entite film podľa svojej funkcie vo filme. Je reprezentovaná cez (long) ID
- **Amount** - výška/množstvo - reprezentuje výšku rozpočtu, reprezentovaná cez (long) ID a má vzťah k entite film

Všetky entity sú reprezentované cez Neo4j ID. Aby bolo teda jasné, o aký typ entity sa jedná, majú všetky uzly definované vlastnosti a v nich je určený typ entity nasledovne:

- Kľúč tvorí typ entity + koncovka „id“, teda napríklad: *movie_id*, *amount_id*, *type_id*.
- Hodnotu tvorí ID, ktoré je namapované na databázu SQL a je typu int.

4 3. Šprint

4.1 Služba Neo4j

4.1.1 Úloha

Vytvorte službu na prácu s databázou Neo4j mimo prostredia virtuálneho servera. Sprístupnite API, ktoré Neo4j ponúka a vytvorte API na pridávanie tripletov.

4.1.2 Analýza

Neo4j ponúka RESTful API, ktoré je možné volať z lokálneho servera. API ma veľký počet volaní (rádovo 100), takže prepisovať všetky volania je nepraktické. API je možné posilať požiadavky GET, POST, PUT a DELETE cez HTTP protokol, potrebné dáta sú pripojené v JSON objekte.

Vkladanie tripletov zahŕňa spoločné vkladanie dvoch vrcholov a vzťahu medzi nimi.

4.1.3 Návrh

Je potrebné vytvoriť API, ktoré bude spracovávať rovnaké požiadavky a preposielať ich na Neo4j API a následne preposielať odpovede aj s kódom odpovede naspäť. Požiadavky ani odpovede nie je potrebné upravovať.

Vkladanie tripletov pozostáva z troch častí - vloženie oboch vrcholov a vloženie vzťahu. Vloženie každého vrchola prebieha nasledovne:

- ak áno, na tvorenie vzťahu použijeme existujúci vrchol
- ak nie, vytvoríme nový vrchol a na tvorenie vzťahu použijeme nový vrchol

4.1.4 Implementácia

Službu sme vytvorili v pracovnom rámci Ruby on Rails a komunikuje s Neo4 cez HTTP protokol. Služba preposiela všetky volania a vracia získané odpovede.

Volanie na vloženie tripletu prijíma napr. nasledovný JSON objekt (s a o sú subjekt a objekt - dva vrcholy, p je predikát - vzťah medzi nimi):

```
{
  "s" : {"id" : 19, "data" : {"movie_id":3} },
  "p" : {"type" : "is_similar", "data" : {"rating":0.75} },
  "o" : {"id" : 42, "data" : {"movie_id":8} }
}
```

pričom pri vrcholoch stačí zadať jeden s parametrov id a data, pri vzťahu je parameter data nepovinný. Volanie vráti informácie o vytvorenom vzťahu, napr.:

```
{
  "extensions": {},
  "start": "http://localhost:7474/db/data/node/27",
```

```
"property": "http://localhost:7474/db/data/relationship/6/properties/{key}",  
"self": "http://localhost:7474/db/data/relationship/6",  
"properties": "http://localhost:7474/db/data/relationship/6/properties",  
"type": "is_similar",  
"end": "http://localhost:7474/db/data/node/28",  
"data": {  
  "rating": 0.75  
}  
}
```

4.2 Parser kín

4.2.1 Úloha

Vytvorte metódu pre vyparsovanie programu najväčších slovenských kín. Parser musí zahŕňať dve najväčšie siete slovenských kín a to konkrétne Cinema City a CineMax.

4.2.2 Analýza

Cinema City

Príklad URL:

http://www.cinematicity.sk/index.php?action=10101&id_menu=10&program=eurovea&date=2012-10-24

- **action:** nič
- **id_menu:** označenie kina
 - 9: aupark
 - 10: eurovea
 - 11: polus
- **program:** opäť označenie kina, možnosť je aupark/eurovea/polus
- **date:** dátum premietania vo formáte rok-mesiac-deň

CSS rozdelenie:

- **získanie programu:** tr
 - *čas premietania filmu:* a.rezervace
 - *názov filmu (slovenský):* .nazev_filmu
 - *názov filmu (originálny):* otvorenie stránky s detailami filmu a následne, #left h3
 - *prístupnosť:* .podbarvena:nth-child(2)
 - *dĺžka filmu:* .podbarvena:nth-child(4)
 - *znenie:* .podbarvena:nth-child(3)

CineMax

Príklad URL:

<http://www.cine-max.sk/program-kosice/den1>

- **program-mesto:** idem o mesto, o ktoré máme pri prezeraní záujem, namiesto medzery sa používa pomlčka, napríklad <http://www.cine-max.sk/program-banska-bystrica/den1>

- `denx`: aktuálny deň sa zobrazuje ako `den1`, následne sa ďalšie dni zobrazujú ako `den2`, `den3`, ..., `den7`

CSS rozdelenie:

- **získanie programu:** `tr`
 - *čas premietania filmu*: `a span`
 - *názov filmu (slovenský)*: `.moviename`
 - *názov filmu (originálny)*: otvorenie stránky s detailami filmu a následne, `ul:nth-child(1) li:nth-child(1)`
 - *prístupnosť*: `.tblboxblue`
 - *dĺžka filmu*: `b:nth-child(1)`
 - *znenie*: `.tblboxred`

4.2.3 Návrh

Okrem vyparsovania údajov z premietacieho programu, je potrebné aby boli filmy rozpoznateľné v komplexnej databáze filmov pre projekt TeleVido. Preto sa po vyparsovaní originálneho, nie slovenského, názvu skontroluje, či sa daný film v databáze nachádza. Ak nie tak je z neho potrebné vytvoriť nový záznam, ak áno tak pracujeme s už existujúcim záznamom. Pre lepšiu presnosť je z originálneho názvu potrebné odstrániť dodatočné údaje, teda či sa jedná o 2D alebo 3D film. Následne sa informácia o 3D filme zaznamená samostatne.

Parsovanie bude prebiehať každú noc, pričom sa vyparsuje program na týždeň dopredu. Kvôli prípadným nezhodám sa každú noc prepíše starý program novým.

4.2.4 Implementácia

Parser bol naimplementovaný v jazyku Ruby v pracovnom rámci Ruby on Rails. Pre parsovanie boli využité gemy Nokogiri a Mechanize.

4.3 Model používateľa

4.3.1 Úloha

Navrhnuť model používateľa v MySQL databáze.

4.3.2 Návrh

Pre používateľský model boli navrhnuté nasledovné tabuľky:

- **USERS** - Tabuľka obsahuje informácie o používateľoch
- **PERSONS** - Tabuľka obsahuje informácie o ľuďoch, ktorý spĺňajú nejakú rolu v filmoch alebo seriáloch. Napríklad herci, direktori alebo producenti. Tabuľka obsahuje pole "neo4j", ktoré je číslo uzlu reprezentujúcej entity v grafovej databáze.
- **GENRES** - Tabuľka obsahuje všetky žánre. Tabuľka obsahuje pole "neo4j", ktoré je číslo uzlu reprezentujúcej entity v grafovej databáze. V Poli keywords sa nachádzajú kľúčové slová, ktoré identifikujú žánre.
- **MOVIES** - Tabuľka obsahuje všetky filmy a ich špecifické atribúty. Tabuľka obsahuje pole "neo4j", ktoré je číslo uzlu reprezentujúcej entity v grafovej databáze.
- **SERIES** - Tabuľka obsahuje všetky seriály a ich špecifické atribúty. Tabuľka obsahuje pole "neo4j", ktoré je číslo uzlu reprezentujúcej entity v grafovej databáze.
- **RECOMMENDATIONS** - Tabuľka obsahuje už vygenerované odporúčanie pre jednotlivých používateľov.
- **THEATERS** - Tabuľka obsahuje údaje o kinách. Napr. mesto , lokalitu a konkrétnu geografickú pozíciu.
- **THEATERPROGRAMS** - Tabuľka obsahuje program v kinách, vždy pre aktuálny týždeň. Dáta sa získavajú pomocou parserov.
- **TVPROGRAMS** - Tabuľka obsahuje televízny program pre aktuálny týždeň.
- **MYPROGRAM** - Je tabuľka, ktorá slúži na špecifické odporúčanie, kde každý používateľ má personalizovaný plán na televízne programy.
- **USERINTERESTS** - Je polymorfická tabuľka, ktorá obsahuje implicitné hodnotenie používateľov o jednotlivých entitách.
- **USERFEEDBACKS** - Je polymorfická tabuľka, ktorá obsahuje explicitné hodnotenie používateľov o jednotlivých entitách.

Obr. 6: Používateľský model v MySQL

5 4. Šprint

5.1 Prvotný návrh rozhrania webovej aplikácie a prípady použitia

5.1.1 Úloha

Navrhňte prvotný návrh webového rozhrania webovej aplikácie TeleVido. Zároveň uveďte prvý návrh prípadov použitia webovej aplikácie projektu TeleVido. Určte prvotné funkcie inteligentnej televízie.

5.1.2 Analýza

Pred návrhom prvotného rozhrania webovej aplikácie a prípadov použitia sme si dôkladne zanalyzovali podobné existujúce riešenia. Poukázali sme na slabé stránky a takisto aj na silné stránky analyzovaných riešení.

5.1.3 Návrh

V prvej fáze vývoja webovej aplikácie TeleVido sme navrhli prvotné návrhy prípadov použitia. Navrhnuté prípady použitia sú:

- **Hodnotenie**
 - Twitter
 - Komentáre
 - Hviezdičkovanie
 - Implicitné
 - Explicitné
 - Externé (imdb, čsfd)
- **Automatické priradenie génov**
- **Odporúčanie filmov**
 - Kolaboratívne
 - Obsah
 - Twitter a Facebook
 - Gény
 - Čas
 - Zdroj (TV, kino)
- **Real time tags**
- **Vyhľadávanie filmov**

- Fulltext
 - * kľúčové slová
- Fazety
 - * podľa žánru
 - * podľa obdobia
 - * podľa obľúbenosti
 - * podľa pozeranosti
- Rozšírené
 - * rôzne pokročilé nastavenia
- Ostatné
 - * odporúčané
 - * wishlist
 - * moji priatelia

- **TV program**

- Môj vlastný TV program
- kľúčové slová
- Aktuálny TV program
- Detail filmu
- Hlavné informácie o filme
 - * nadpis
 - * fotografia
 - * popis
 - * kategórie a informácie o filme
- Video upútavka filmu
- Komentáre
 - * napísanie komentáru
 - * zobrazenie výpisu komentárov
- Podobné filmy
- Odporúčané filmy
- Čo sa o filme píše

- **Prihlásený používateľ**

- Registrácia používateľa
- Prihlásenie používateľa

- Profil o svojej osobe
- Zoznam priateľov
- Pridanie priateľov

- **Nezaradané funkcie**

- Posielanie noviniek e-mailom
- Prepojenia so sociálnymi sieťami

Požiadavky na rozhranie:

- Intuitívny a moderný dizajn stránky
- Prehľadné kategórie v navigácii Jednoduchosť
- Prehľadnosť
- Vyhľadávací box na stránke
- Možnosti prepojenia na sociálne stránky

Vizuálne prvky:

- Text
- Obrázky
- Button (tlačidlo)
- Check box (zaškrťavacie pole)
- Text area (textové pole)
- Text box (vstupné pole)
- Combo box (rozbaľovacie menu)

Na Obr. 7 - Obr. 10 sú uvedené obrazovky návrhu webovej aplikácie prostredníctvom ručne nakreslených wireframov.

TELEVIDO

Home

Filmy

TV program

Prihlásenie

Registrácia

Nájdite si najlepší program v TV

Kľúčové slová

Vyhľadať

ROZŠIŘENÉ VÝHL

- komedie
 - drámy
 - dokumentárne
- VŠETKY ŽÁRE (25)

- najnovšie
 - najostrejšie
 - najpозeranejšie
- VŠETKY ROZDEL (12)

- filmy
 - seriály
 - TV program
- VŠETKY TYPY (10)
- odporúčané
 - klišist
 - moji priatelia
- VŠETKO (5)



Top funkcia 1



Top funkcia 2



Top funkcia 3

Odporúčané / TOP filmy

5 máj 2013

SAM DOMA 2.

~ ~ ~ ~ ~

Kategória: detka, vlastnor. klišist komentár → viac

16 august 2012

TITANIC

~ ~ ~ ~ ~

Kategória: detka, vlastnor. klišist komentár → viac

22 apr 2012

HARRY POTTER

~ ~ ~ ~ ~

Kategória: detka, vlastnor. klišist komentár → viac

Rýchle odpazy

- ~ ~ ~ ~ ~
- ~ ~ ~ ~ ~
- ~ ~ ~ ~ ~
- ~ ~ ~ ~ ~
- ~ ~ ~ ~ ~

Posledné novinky

- @Lubos Demovic ~ ~ ~ ~ ~
- pred 18 min
- @Jatubkri2 ~ ~ ~ ~ ~
- pred 21 min

Novinky e-mailom

- ~ ~ ~ ~ ~
- e-mail
- RSS FB TW YOUTUBE

Copyright © 2013 teleVido

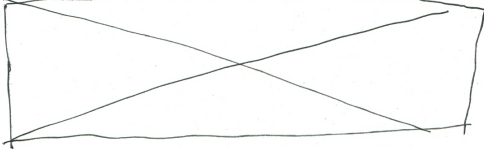
kontakt • Tlačové správy • Spolupráca • Pomoc

Obr. 7: Titulná stránka

TELEVI DO

Film → Horror → Detail filmu
~~TITANIC~~

TITANIC → zäver
→ režisör
→ rok




~~~~~

• ~~~~~ : ~~~~~  
• ~~~~~ : ~~~~~  
• ~~~~~ : ~~~~~

FB TV

Uputāvka




Komentāri

| mens  | datums |
|-------|--------|
| ~~~~~ | ~~~~~  |

Napītis komentāri

Meno


E-mail




OK

## Podobne filmy


Nadpis




Nadpis



Nadpis




Nadpis




...

## Oppozīcijas filmy

Nadpis



Nadpis



...

## Čo sa o filmu piē

@LuborAemovic ~~~~~  
pred 5 min

@bluskaz ~~~~~  
pred 10 min

## Klūcēvē slova

lāsta jack rose  
lōd titanie lādovee

Obr. 8: Detail filmu

Výsledky vyhledávání

NADPIS

KOPÍROVAT

SDÍLEŤ

A hand-drawn search result card. On the left is a square placeholder with an 'X' inside. To its right is the word 'NADPIS'. Below the title are three horizontal wavy lines representing text. At the bottom left are two rectangular buttons labeled 'KOPÍROVAT' and 'SDÍLEŤ'. At the bottom right is a small circular icon containing a document symbol.

NADPIS

A hand-drawn search result card showing a square placeholder with an 'X' and the word 'NADPIS' to its right.

NADPIS

A hand-drawn search result card showing a square placeholder with an 'X' and the word 'NADPIS' to its right.

NADPIS

A hand-drawn search result card showing a square placeholder with an 'X' and the word 'NADPIS' to its right.



Obr. 9: Výsledky vyhledávání

# TELEVIDO

Pondelok  Utorok  Streda  Štvrtok  Piatok  Sobota  Nedeľa

- |                                                                                                                             |                                                                                                                     |                                                                                                              |                                                                                                                                                            |                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <input type="checkbox"/> Filmy<br><input checked="" type="checkbox"/> seriál<br><input checked="" type="checkbox"/> Relácie | <input type="checkbox"/> Dokument<br><input type="checkbox"/> Pre detskú<br><input type="checkbox"/> ostatné (viac) | <input type="checkbox"/> vyhlásenie<br><input type="checkbox"/> odvetaz<br><input type="checkbox"/> celý deň | <input type="checkbox"/> Rozprávka<br><input type="checkbox"/> odporúčanie<br><input type="checkbox"/> unikátny<br><input type="checkbox"/> moji priatelia | <input type="checkbox"/> športové TV<br><input type="checkbox"/> spravodajské TV<br><input type="checkbox"/> titulové TV (ušetť) |
|-----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                  |                            |                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|----------------------------|----------------|
| <p>1   STV 1</p> <p>10:00 Jahodové víno<br/>                 (Film) ~~~~~<br/>                 ~~~~~<br/>                 ~~~~~<br/>                 ~~~~~<br/>                 ~~~~~</p> <p>12:00 Krok za krokom<br/>                 (Seriál) ~~~~~<br/>                 ~~~~~<br/>                 ~~~~~<br/>                 ~~~~~</p> <p>14:00 Hodie z Neke<br/>                 (Relácia) ~~~~~<br/>                 ~~~~~<br/>                 ~~~~~</p> <p>17:00 Správy STV<br/>                 17:00 Počasie</p> | <p>2   STV 2</p> | <p>3   MARKÍZA</p>         | <p>4   JOJ</p> |
| <p>Prima   PRIMA</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | <p>HBO   HBO</p> | <p>Hallmark   HALLMARK</p> | <p>7   ČT1</p> |

Obr. 10: TV program

## 6 5. Šprint

### 6.1 Získanie testovacích dát TV programu

#### 6.1.1 Úloha

Získajte testovacie dáta TV programu na 1 týždeň niekoľkých televíznych staníc.

#### 6.1.2 Analýza

Pre rýchle získanie testovacích dát TV programu sme čerpali z voľne prístupných dát z EPG každej TV.

#### 6.1.3 Návrh

**Vo vzorke dát sa nachádza TV program na celý jeden týždeň pre 91 televíznych staníc:**

AXN, AXN Crime, AXN Sci-Fi, Animal Planet, Barrandov, CS Mini - CS Film, CT 1, CT 2, CT 4 Sport, CT 24, CT HD, Cinemax, Cinemax 2, Digi Sport CZ-SK, Discovery Channel, Discovery Channel HD, Discovery Channel HU, Discovery Science, Discovery World, Disney Channel, Doku CS, Duck TV, Eurosport, Eurosport 2, Eurosport HD, Extreme Sports Channel, Fashion TV, FilmBox Extra 1, FilmBox Family, FilmBox Plus, Film Europe, Filmbox, Filmbox HD, Film plus, Fine Living Network, Fishing and Hunting, Fun 1, HBO, HBO 2, HBO Comedy, HBO HD, History Channel, History Channel HD, JIM JAM, JOJ HD, Joj, Joj Plus, Joj Plus HD, Kino CS, MGM, MTV Czech, Markiza, Markiza HD, Minimax - Animax, Minimax, MusicBox, Musiq1, Muzika CS, National Geographic, National Geographic Wild, Noe TV, Nova, Nova Cinema, Nova Sport, Ocko, PV TV, Paprika, Patriot, Prima Cool, Prima Love, Prima family, Retro Music Television, Romantica, STIL TV, Spektrum, Spektrum Home, Sport 1, Sport 2, Sport 5, Stv 1, Stv 1 HD, Stv 2, TV8, TV Doma, TV Lux, Ta 3, Travel channel, Universal Channel, Viasat Explorer, Viasat History, Viasat Nature, Zone Reality

**Informácie o jednotlivých položiek daného programu:**

- **Eventid** - ID konkrétneho programu v TV
- **Eventstart** - Čas začiatku programu v TV vo formáte timestamp
- **Eventcurrenttime** - Dĺžka trvania programu v TV vo formáte timestamp
- **Eventcurrenttime** - Čas konca programu v TV vo formáte timestamp
- **Eventdescription** - Krátky opis programu v TV.
- **Eventdescriptionextended** - Rozšírený opis programu v TV.
- **Eventservicename** - Názov kanálu programu v TV.

### Ukážka štruktúry jedného programu v TV:

```
<event>
<eventid>1612</eventid>
<eventstart>1343435400</eventstart>
<eventduration>10200</eventduration>
<eventcurrenttime>1343426623</eventcurrenttime>
<eventtitle>Pomsta Jackieho Chana</eventtitle>
<eventdescription>Akčný film</eventdescription/>
<eventdescriptionextended>
“Podrobný popis filmu.”
</eventdescriptionextended>
<eventservicename>Markiza</eventservicename></event>
```

#### 6.1.4 Implementácia

Na získanie testovacích dát TV programu sme využili populárny open source skriptovací jazyk PHP. Na uloženie informácií a dát sme využili databázu MySQL. Získali sme 20 071 programov z 91 rôznych televíznych staníc na obdobie 1 týždňa.

## 6.2 Linked Movie Database Parser

### 6.2.1 Úloha

Vyrobiť parser pre Linked Movie Database.

### 6.2.2 Návrh

Nato aby sme vedeli odporúčať nejaký obsah, potrebujeme vybudovať doménový model a nato potrebujeme dáta. Pre zdroj dát sme vybrali linked movie database ktorá obsahuje cca. 100 000 filmov, 200 000 hercov, direktorov a producentov a rôzne ďalšie informácie prepojené pomocou tripletov v RDF štruktúre. Avšak náš model sa líši od modelu, ktorý používajú v linked movie database. Tak sme museli navrhnúť a implementovať parser, ktorý spracuje dáta, extrahuje entity a znova ich prepojí v správnej štruktúre.

**Fungovanie parsera:** náš parser ma tri hlavné kroky. V prvom kroku sa pomocou regulárnych výrazov snaží získať potrebné entity s RDF štruktúry. Po získaní entít prejde ešte raz celou RDF štruktúrou a pre vybrané vzťahy sa snaží vytvoriť nové, ktoré už budú v správnej štruktúre. Ako tretí krok parser vytvorí dva súbory. Jeden je pre MySQL databázu, ktorá obsahuje všetky entity v forme MySQL dumpu a v druhom súbore sú triplety v forme JSON objektov ktoré naše webové služby dokážu spracovať.



## 6.3 Neo4j prídavný modul - Algoritmus spájania farieb

### 6.3.1 Úloha

Vytvorte prídavný modul do grafovej databázy Neo4j, ktorý bude hľadať najkratšie uzly k vybraným uzlom, ktoré budú zadané na vstupe ako parametre. Algoritmus očakáva na vstupe ďalšie parametre, ktoré budú vytvárať univerzálny nástroj na hľadanie najbližších vrcholov. Typy parametrov môžu byť napr. eliminácia typov vrcholov, eliminácia vzťahov, zoznam vrcholov, cez ktoré treba prejsť a pod.

### 6.3.2 Analýza

Existuje niekoľko prídavných modulov, ktoré sú implementované do grafovej databázy Neo4j a tiež niekoľko algoritmov na hľadanie ciest.

**ShortestPath** - nájde najkratšiu cestu z vybraného vrcholu do všetkých ostatných vrcholov alebo do jedného vybraného vrcholu. Tento vstavaný algoritmus pracuje nad vrcholom a funkciu je možné volať nad ľubovoľným vrcholom v grafe. Algoritmus ponúka možnosť nastaviť si maximálnu hĺbku, do ktorej sa budú vrcholy prehľadávať. Tiež je možné zadať iba typy relácií, cez ktoré bude algoritmus prechádzať.

**GetAll** - algoritmus nájde všetky vzťahy v grafe.

**Prehľadávanie do šírky** - Pri prehľadávaní si zvolíme jeden vrchol, ktorý máme ako začiatočný. Z vrcholu sa postupne začneme širiť do všetkých susedných vrcholov, ktoré sme ešte nenavštívili. Do týchto vrcholov sa vieme dostať s hĺbkou 1. Následne opäť každý vrchol prehľadáme a vyberieme tie vrcholy, ktoré sme ešte nenavštívili. Tie vieme navštíviť na vzdialenosť 2. Takto pokračujeme, až kým nenájdeme hľadaný vrchol alebo množinu. Algoritmus tiež môže končiť vo vybranej hĺbke. Na prehľadávanie do šírky môžeme použiť frontu, kde si budeme ukladať vrcholy, ktoré sa pokúsime navštíviť.

### 6.3.3 Návrh

Vytvoríme algoritmus, ktorý nájde zadaný počet vrcholov, ktoré sú najkratšie k zadanej množine vrcholov. Algoritmus bude používať rovnocenné vzťahy medzi vrcholmi a to s váhou 1.0.

Vytvoríme prídavný modul, ktorý bude mať na vstupe niekoľko voliteľných parametrov a jeden povinný parameter.

- "nodes" : []
  - vstupné vrcholy
  - povinný parameter
- "max\_depth" : <> - maximálna hĺbka - voliteľný parameter, predvolené nekonečno
- "count": <>
  - počet vrcholov na výstupe

- voliteľný parameter, predvolené nekonečno
- "result\_node\_types\_include" : []
  - typy vrcholov, ktoré ma obsahovať výstup
- "result\_node\_types\_exclude" : []
  - typy vrcholov, ktoré nemá obsahovať výstup
- "result\_nodes\_include" : []
  - vrcholy, ktoré ma obsahovať výstup
- "result\_nodes\_exclude" : []
  - vrcholy, ktoré nemá obsahovať výstup
- "pass\_node\_types\_include" : []
  - typy vrcholov, cez ktoré ma prechádzať
- "pass\_node\_types\_exclude" : []
  - typy vrcholov, cez ktoré nemá prechádzať
- "pass\_nodes\_include" : []
  - vrcholy, cez ktoré ma prechádzať
- "pass\_nodes\_exclude" : []
  - vrcholy, cez ktoré nemá prechádzať
- "pass\_relationships\_include" : []
  - typy relácií, cez ktoré ma prechádzať
- "pass\_relationships\_exclude" : []
  - typy relácií, cez ktoré nemá prechádzať
- "weight\_relationships" : []
  - váhy relácií

**Výstupom pluginu bude množina JSON objektov:**

- "neo4j\_id" : <>
  - id v Neo4j databáze
- "entity" : ""
  - názov entity: genre (žáner), movie (film), serie (seriál), person (osoba)
- "entity\_id" : <>
  - id v MySQL databáze
- "depth" : <>
  - hĺbka
- "rating" : <0,1>
  - úspešnosť odporúčania

#### 6.3.4 Implementácia

Algoritmus využíva klasické prehľadávanie do šírky, ktoré sme modifikovali. Na začiatku si okrem začiatočných vrcholov označíme každý vrchol inou farbou a tak začneme prehľadávať graf s farbením ciest každou z týchto farieb. Pri postupnom šírení si budeme označovať vrcholy jednotlivými farbami. Ak narazíme na vrchol, ktorý ma inú farbou ako vrchol, z ktorého sme sa do vrcholu dostali, musíme tieto dve farby spojiť a povedať, že tieto dve farby budú odteraz reprezentovať jednu. Takto budeme postupovať kým nespojíme posledné dve farby. Vtedy sú všetky cesty spojené do jednej a môžeme vrátiť zadaný počet vrcholov. Na spájanie viacerých farieb do jednej sme použili Dátovú štruktúru pre disjunktné množiny. Prídavný modul sme implementovali vo vývojovom prostredí Java a grafovej databázy Neo4j.

## 6.4 Neo4j prídavný modul - Algoritmus miešania farieb

### 6.4.1 Úloha

Vytvorte prídavný modul do Neo4j databázy pre získanie najbližších uzlov ku zadaným uzlom. Algoritmus bude na vstupe očakávať začiatkové uzly, ale taktiež môže dostať aj iné parametre, aby bol kompatibilný s ostatnými vytvorenými prídavnými modulmi.

### 6.4.2 Analýza

Odporúčaným spôsobom vytvorenia prídavného modulu do Neo4j databázy je vytvorenie JAR súboru. Vytvorená trieda musí dediť od triedy `ServerPlugin`. Vytvorený JAR súbor musí v priečinku `META-INF/services` mať vytvorený súbor `org.neo4j.server.plugins.ServerPlugin`, v ktorom sú zaznamenané cesty k vytvoreným triedam. Teda pre triedu `HelloWorld` v balíku `org.neo4j.examples.server.plugins` je potrebné aby bolo v súbore zapísané `org.neo4j.examples.server.plugins.He`

### 6.4.3 Návrh

Algoritmus je založený na klasickom BFS (prehľadávanie do šírky). Každý vstupný vrchol má vlastnú farbu. Šírenie sa začne zo všetkých vstupných vrcholov, a skončí sa, keď už nemáme aký nový vrchol prehľadať, alebo ak sme už našli požadovaný počet výsledkov. Na ukladanie uzlov, z ktorých chceme prehľadávať použijeme frontu.

Ak zafarbíme uzol novou farbou (alebo kombináciou farieb) tak ho zaradíme na koniec fronty. Uzol vysielá susedným uzlom vždy všetky farby, ktoré obsahuje. Teda ak uzol v prvom kroku získa tri farby (bol blízko vstupných vrcholov) tak v ďalšom kroku už susedným uzlom vyšle všetky tri farby naraz. Ak sa stane, že je uzol zafarbený všetkými farbami, tak ho môžeme zaradiť do výsledku algoritmu.

Algoritmus vždy nájde ako prvé tie uzly, kde je vzdialenosť od najvzdialenejšieho uzla minimálna. To znamená, že ak by sme začínali v troch uzloch a mali na výber uzol, ktorý je vzdialený od každého na dva kroky (2, 2, 2) alebo taký, ktorý je vzdialený od prvého na tri a zvyšných na jeden krok (3, 1, 1), tak algoritmus nájde prvé riešenie (2, 2, 2).

### 6.4.4 Implementácia

Ako bolo spomenuté v analýze, tak algoritmus bude vytvorený v programovacom jazyku Java. Vstupné a výstupné parametre budú rovnaké ako v prípade `UnionColorsPlugin`.

## 6.5 Neo4j prídavný modul - Algoritmus šírenia energie

### 6.5.1 Úloha

Vytvorte prídavný modul pre spoluprácu s grafovou databázou Neo4j. Tento modul bude reprezentovať algoritmus šírenia energie na uzloch v databáze. Vstupom bude niekoľko parametrov, ktoré budú upresňovať spôsob šírenia energie na uzloch a budú formátovať výstup. Všetky parametre sú voliteľné okrem parametru „nodes“, ktorý obsahuje množinu začiatočných uzlov.

### 6.5.2 Analýza

Pre spoluprácu s databázou Neo4j je potrebné vytvoriť súbor typu .jar, ktorý obsahuje triedu, ktorá dedí od triedy ServerPlugin. Táto trieda sa nachádza v balíčku org.neo4j.server.plugins. Tento .jar súbor musí obsahovať súbor org.neo4j.server.plugins v priečinku META-INF/services. V tomto súbore sa nachádza cesta k hlavnej triede, ktorá dedí od triedy ServerPlugin teda napr. package org.neo4j.server.plugins.EnergyPlugin.

API poskytuje možnosti pre získanie aktuálneho uzla, jeho vzťahov k iným uzlom, jeho vlastností. Umožňuje tiež vytváranie vlastností, uzlov a vzťahov. Poskytuje spôsob pre získanie parametrov a vytvorenie výstupu.

### 6.5.3 Návrh

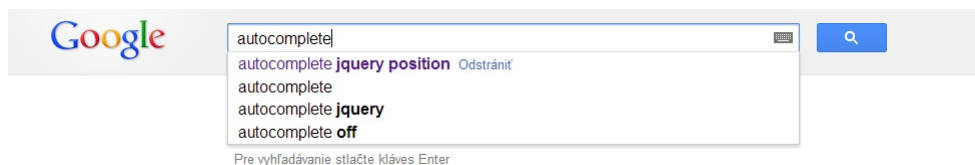
Základom algoritmu je prehľadávanie do šírky. Energia sa šíri zo začiatočných uzlov do ich susedných uzlov dovtedy, kým nie sú prehľadané všetky uzly v určitej hĺbke. Táto hĺbka je buď zadaná ako voliteľný parameter alebo sa určí ako hĺbka, v ktorej boli nájdené prvé výsledky. Výsledkami sú uzly, ktoré boli navštívené zo všetkých začiatočných vrcholov.

Energia sa šíri tak, že začiatočným uzlom je pridelená určitá energia napr. 10, ak má začiatočný uzol 5 nasledovníkov, každému z nich je poslaná energia  $10/5=2$ . Takto sa energia šíri ďalej, pokiaľ nájde uzol, ktorý už raz bol navštívený z daného začiatočného uzla tak mu zvýši energiu, ale nešíri ju z neho ďalej. Po nájdení výsledkov sa výsledky zoradia podľa množstva energie na výstup.

### 6.5.4 Implementácia

Tento algoritmus bol implementovaný v jazyku Java. Vstupné parametre a výstup sú bližšie popísané v časti Neo4j prídavný modul - algoritmus spájania farieb. Z dôvodu vzájomnej kompatibility boli všetky tri prídavné moduly implementované s rovnakými vstupnými parametrami a výstupom.

Čo sa týka implementácie samotného algoritmu, zo všetkých začiatočných uzlov sa energia šíri naraz, striedajú sa postupne po jednom uzle. Každý zo začiatočných uzlov má preto vlastný front kam si ukladá uzly, ktoré má navštíviť. Informácie o uzloch sa ukladajú do objektu Visited. Ten obsahuje informácie pre každý navštívený uzol napr. uzol N: hĺbku uzla N a jeho energiu z každého začiatočného uzla, informácie o tom, z ktorých začiatočných uzlov už bol navštívený.



Obr. 11: Automatické dopĺňanie od Google

## 6.6 Prototyp odporúčania

### 6.6.1 Úloha

Vytvorte jednoduchý prototyp odporúčacej stránky a funkčné inteligentné dopĺňanie vyhľadávaného výrazu, ktorý môže byť názov filmu, seriálu, žánru alebo osoby, ktorá vo filme / seriály vystupuje alebo nejako inak účinkuje. Po vybratí jedného z ponúkaných entít (názov filmu, seriál, žánr, osoba) je entita pridaná do zoznamu tagov, podľa ktorého bude aplikácia odporúčať film.

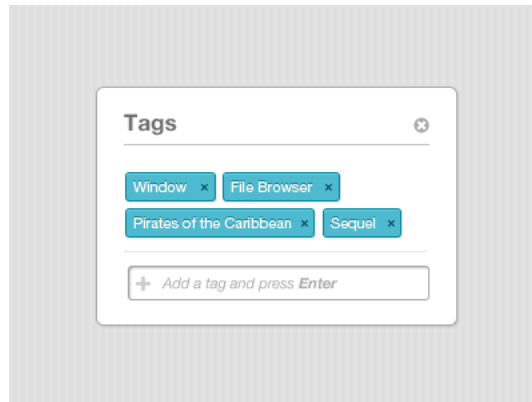
### 6.6.2 Analýza

V súčasnosti je na webe veľa dynamického obsahu, ktorý ponúka rôzne inteligentné funkcie, ktoré nám uľahčujú používanie aplikácií. Jednou z takýchto inteligentných funkcií je aj automatické dopĺňanie textu pri písaní do textového poľa. Na Obr. 11 môžeme vidieť ukážku inteligentného dopĺňania na najpoužívanejšom vyhľadávači od Google.

Z veľkého množstva už implementovaných knižníc sme vybrali autocomplete, ktoré používa jQuery UI (dostupné na stránke <http://jqueryui.com/autocomplete/>). Toto automatické vyhľadávanie ponúka možnosť nastaviť si počet znakov, ktoré je potrebné zadať, aby bolo možné zobraziť ponúkané výrazy. Okrem iného je to otvorená knižnica a je ľahko rozšíriteľná. Pre zobrazenie tagov resp. vybraných entít na odporúčanie, ktoré môžu byť filmy, žánre, seriály, relácie, osoby sme sa inšpirovali funkcionalitu, ktorá je na Obr. 12 Každý nájdený tag (entita) sa pridá do zoznamu vybraných a kedykoľvek je možné ju zmazať kliknutím na tlačidlo "x".

### 6.6.3 Návrh

Vytvoríme jednoduchú stránku, ktorá bude ponúkať textové okno. Do tohto okna je možné napísať akýkoľvek výraz, ktorý sa týka filmovej a televíznej oblasti a tiež nejakú postavu, režiséra a pod. Potom ako používateľ napíše výraz, aplikácia sama začne odporúčať vybranú skupinu entít, ktoré majú zhodný prefix, ako zadaný výraz. Potom ako používateľ klikne na vybranú entitu, entita sa pridá do množiny vybraných tagov a na základe vybraných tagov, aplikácia odporučí vhodný film. Používateľ môže akýkoľvek tag zmazať a služba aktualizuje zoznam odporúčaných filmov.



Obr. 12: Funkcionalita tagov

## Recommendations



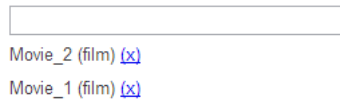
Obr. 13: Automatické dopĺňanie v našej aplikácii

### 6.6.4 Implementácia

Aplikácia bola naprogramovaná v pracovnom rámci Ruby On Rails a jazyku Ruby. Použili sme relačnú databázu MySQL a grafovú databázu Neo4j. Inteligentné dopĺňanie bolo vytvorené pomocou JavaScript, jQuery a AJAX technológiám.

Na Obr. 13 je ukážka automatického dopĺňania v našej aplikácii. Po kliknutí na vybratú entitu sa entita pridá do zoznamu tagov, podľa ktorých je potrebné odporúčať (ako môžeme vidieť na Obr. 14).

## Recommendations



Obr. 14: Zoznam vybratých entít na odporúčanie v našej aplikácii

## 7 6. Šprint

### 7.1 Metóda na napĺňanie databáz z linked movie parsera

V kapitole 6.2 sme parsovali Linked Movie Database, z ktorého nám vyšiel výstup:

- dump MySQL, kde boli konkrétne inserty filmov, žánrov a ľudí
- množina tripletov

#### 7.1.1 Naplnenie MySQL

Dump do MySQL obsahoval riadky insertov do databázy.

```
INSERT INTO Genres
(id,genre,keywords,neo4j,created_at,updated_at)
VALUES (1,"Conspiracy theory","Conspiracy theory","", "2012-11-19 17:37:47","2012-11-19 17:37:47");
```

Celý dump bolo potrebné vložiť ako SQL import do databázy.

MySQL databáza obsahovala približne 85000 filmov, 85000 ľudí a niekoľko žánrov.

#### 7.1.2 Naplnenie Neo4j

Pre naplnenie Neo4j databázy sme dostali množinu tripletov v takomto tvare:

```
{
  "s" : {"data" : {"movie_id":38079} },
  "p" : {"type" : "produced"},
  "o" : {"data" : {"person":35130} }
}
```

Triplety boli zadefinované normou *spo* a vo formáte JSON. Nevýhodou Neo4j id je, že nedokáže vkladať triplety, pokiaľ tam niektorý z nodov neexistuje. Preto bolo potrebné, aby sme si pamätali, ktoré vrcholy sme už do Neo4j vložili a pamätali si ich neo4j\_id, ktoré budeme potrebovať pri odporúčaní. Metóda bola napísaná v Ruby.

Celá množina tripletov obsahovala viac ako 290000 riadkov a bolo potrebné ju rozdeliť, pretože pri vkladaní celej množiny, Neo4j po čase spadlo. Preto sme celú množinu rozdelili na 6 častí a každú vyriešili zvlášť. Keďže pri insertovaní vrcholov sme získali neo4j\_id pridaného vrchola, bolo potrebné, aby sme si zapamätali, že daný node s parametrami *entity\_name* a *entity\_id* už má priradené neo4j\_id. nato sme využili hashovaciu tabuľku, ktorá obsahovala



názvy `entity_name` a každý prvok bola opäť hashovacia mapa, ktorá si pamätala ako index `entity_id` a ako hodnotu `neo4j_id`.

---

**Algoritmus 1** Ukážka metódy - nájdenie `neo4j_id`

---

```
def self.find(item)
  data = item["data"]
  entity_key = data.keys.first.downcase
  entity_name = (entity_key.ends_with? "_id") ? entity_key[0..-4] : entity_key
  entity_id = data.values.first
  entity_id_s = entity_id.to_s

  if ENTITIES[entity_name].has_key? entity_id_s
    return ENTITIES[entity_name][entity_id_s].to_i
  end

  neo4j_id = Wrapper.insert_node(item["data"])

  #zaktualizujeme v MySQL
  case entity_name
  when "movie"
    Movie.update_all("neo4j_id = #{neo4j_id}", "id = #{entity_id}")
  when "person"
    Person.update_all("neo4j_id = #{neo4j_id}", "id = #{entity_id}")
  when "serie"
    Serie.update_all("neo4j_id = #{neo4j_id}", "id = #{entity_id}")
  when "genre"
    Genre.update_all("neo4j_id = #{neo4j_id}", "id = #{entity_id}")
  else
  end

  ENTITIES[entity_name][entity_id_s] = neo4j_id
  neo4j_id
end
```

V algoritme je ukážka hľadania `neo4j_id`, ktorá dostane na vstup celú entitu ako JSON, čiže `entity_name` a `entity_id` a vráti `neo4j_id`. Ak sa ešte taká dvojica nenachádza, tak pridá do Neo4j nový vrchol.

## 7.2 Spojenie Neo4j prídavných modulov

### 7.2.1 Úloha

Vytvorte jeden prídavný modul pomocou refactoringu nad už vytvorenými prídavnými modulmi. Úlohou je odstrániť duplicitný kód, zrušenie potreby vytvárať štyri samostatné prídavné moduly (súbory), zlepšenie prehľadnosti a znovu použiteľnosti kódu.

### 7.2.2 Analýza

Pred vykonaním refactoringu boli hotové dva moduly zo štyroch, pričom obsahovali veľké množstvo duplicitného kódu. Zvyšné dva boli v procese vytvárania a preto bolo potrebné myslieť pri refactoringu aj na ne. Jednotlivé java projekty (prídavné moduly) obsahovali len jednu triedu so všetkými potrebnými metódami, prípadne pomocné triedy. Preto boli už aj v úvodnom štádiu mierne neprehľadné.

### 7.2.3 Návrh

Výsledný java projekt (po refactoringu) bude obsahovať triedu s metódou „execute\_plugin“, ktorá okrem parametrov, ktoré využívajú už samotné prídavné moduly, bude požadovať aj parameter „algorithm“, podľa ktorého sa vyberie, ktorý prídavný modul sa spustí. Ďalej bude projekt obsahovať abstraktnú triedu „Helpers“, v ktorej sa budú nachádzať všetky pomocné metódy, ako napríklad získanie susedov pre konkrétny uzol alebo zistenie jeho typu. Jednotlivé prídavné moduly (pred refactoringom) budú vytvorené ako samostatné triedy v spoločnom balíku „algorithms“. Trieda zásuvného modulu bude obsahovať metódu s názvom „find\_all“, ktorá bude obsahovať logiku samotného algoritmu pre odporúčanie. Triedy budú dediť od abstraktnej triedy „Helpers“. To z dôvodu, aby získali všetky potrebné pomocné metódy, ktoré potrebujú na vykonanie metódy „find\_all“. Pomocné triedy, ako napríklad trieda pre vlastný dátový typ, budú zaradené do spoločného balíku „utils“.

### 7.2.4 Implementácia

Projekt bol implementovaný štýlom, ktorý bol opísaný v návrhu. V čase refactoringu boli dokončené len dva prídavné moduly. Z tohto dôvodu bolo potrebné myslieť pri refactoringu aj na zvyšné dva prídavné moduly, aj keď sa s nimi priamo nepracovalo. Preto boli vytvorené triedy aj pre tieto nedokončené moduly, pričom obsahovali prázdnu metódu „find\_all“, do ktorej sa následne má doplniť logika samotného algoritmu. Prípadné pomocné metódy, ktoré ešte neboli vytvorené, sa prekopírujú do abstraktnej triedy „Helpers“ a pomocné triedy do balíku „utils“.

## 8 7. Šprint

### 8.1 Neo4j prídavný modul - Dijkstra plugin

#### 8.1.1 Úloha

Vytvorte prídavný modul do Neo4j databázy pre získanie najbližších uzlov ku zadaným uzlom. Algoritmus bude na vstupe očakávať začiatkové uzly, ale taktiež môže dostať aj iné parametre, aby bol kompatibilný s ostatnými vytvorenými prídavnými modulmi.

#### 8.1.2 Analýza

Dijkstrov algoritmu pre nájdenie najkratšej cesty medzi dvomi uzlami je založený na postupnom prechádzaní grafu, pričom na začiatku sa všetky uzly ohodnotia, že majú vzdialenosť od začiatkového uzla „nekonečno“. Potom sa v každej iterácii vyberie uzol, ktorý ešte nebol prehľadovaný, pričom má najnižšiu vzdialenosť od začiatkového uzla zo všetkých neprehľadovaných uzlov, inú ako „nekonečno“. Pokiaľ sa počas prehľadávania uzla zistí, že niektorý z jeho susedov môže mať menšiu vzdialenosť, ako je jeho aktuálna vzdialenosť (suseda), tak sa mu aktualizuje. Toto opakujeme až kým neprehľadáme cieľový uzol. Pre zefektívnenie vyberania uzla na prehľadanie sa používa halda.

#### 8.1.3 Návrh

Algoritmus bude založený na Dijkstrovom algoritme. Dijkstrov algoritmus pre nájdenie najkratšej cesty medzi dvoma uzlami, bude v tomto prípade prerobený na algoritmus, ktorý nájde všetky uzly do určitej vzdialenosti. Maximálna vzdialenosť bude buď ako jeden zo vstupných parametrov, alebo sa použije prednastavená. Prehľadávanie uzlov bude, ako som už naznačil, pomocou Dijkstrovho algoritmu. Na vstup však môže dostať viacero začiatkových uzlov. Preto sa algoritmus spustí pre každý zo začiatkových uzlov.

Aby bol vrchol použitý pre výstup algoritmu je potrebné, aby spĺňal všetky očakávané kritéria, ktoré sa zadajú ako parametre, a okrem toho, aby bol objavený pri spustení algoritmu z každého začiatkového uzla. To znamená, aby bol od každého začiatkového uzla vzdialený maximálne o hodnotu, ktorá je zapamätaná v parametre maximálna vzdialenosť. Všetky uzly, ktoré tieto podmienky spĺňajú, sa zoradia podľa súčtu vzdialeností od začiatkových uzlov. Následne z nich dokážeme vybrať požadovaný počet, pričom ako prvé budú zoradené najlepšie uzly (tie ktoré majú súčet vzdialeností minimálny).

To znamená, že ak by sme začínali v troch uzloch a mali na výber uzol, ktorý je vzdialený od každého na dva kroky (2, 2, 2) alebo taký, ktorý je vzdialený od prvého na tri a zvyšných na jeden krok (3, 1, 1), tak algoritmus vyberie ako najlepší uzol prvé riešenie (3, 1, 1).

Výhodou tohto algoritmu je možnosť ohodnotiť si relácie, teda menej výrazným nastaviť väčšiu vzdialenosť a výraznejším (dôležitejším) nastaviť menšiu. Toto v prípade predchádzajúcich dvoch algoritmov, založených na BFS, nebolo možné.

### 8.1.4 Implementácia

Ako bolo spomenuté v analýze, tak algoritmus bol vytvorený v programovacom jazyku Java. Vstupné a výstupné parametre sú rovnaké ako v prípade ostatných prídavných modulov.

## 8.2 Optimalizácia Neo4j prídavého modulu - Energy plugin

### 8.2.1 Úloha

Pri spustení modulu nad dátami na serveri sa ukázalo, že zásuvný modul Energy je výrazne pomalý. Preto vznikla požiadavka na jeho prepracovanie tak, aby dochádzalo k menšej spotrebe prostriedkov.

### 8.2.2 Analýza

Najprv bolo potrebné predovšetkým dôkladne preštudovať zdrojový kód tohto modulu a nájsť možnosti pre jeho zrýchlenie. Jednak sa jednalo o použitie rôznych štruktúr a aj o samotný algoritmus.

### 8.2.3 Návrh

V analýze sa ukázali dva možné zlepšenia. V prvom rade sa v zdrojovom kóde odhalilo prehľadávanie spájaného zoznamu (LinkedList) s cieľom zistiť, či obsahuje daný prvok. Táto operácia tzv. „contains“ má zložitosť  $O(n)$ . Ďalej sa ukázalo, že je priestor pre „zahadzovanie“ uzlov s energiou nižšou ako určitá hodnota. Oba tieto nedostatky sa odstránili v implementácii.

### 8.2.4 Implementácia

Prehľadávanie spájaného zoznamu bolo potrebné nahradiť podmienkou, ktorá by zaručovala, že spájaný zoznam daný prvok neobsahuje. Analýzou kódu sa ukázalo, že je to možné prostredníctvom atribútu hĺbka (depth) objektu Visited. Tento atribút sa nastaví až pri vložení prvku do spájaného zoznamu. Druhé možné zlepšenie si vyžadovalo niekoľko pokusov nad dátami na serveri. Konštantu, pri ktorej sa už nebude vkladať prvok do haldy (implementácia cez spájaný zoznam), som určila na základe výsledkov pri cca 15tich pokusoch. Konštantu som určila tak, aby sa nezmenilo 100 najlepších výsledkov a bola rádovo nižšia ako posledný výsledok.

Táto konštanta bola určená ako:

```
double minEnergy = 0.000001
```

## 8.3 Testovanie Neo4j prídavných modulov

### 8.3.1 Úloha

Vytvorte testy pre každú z metód pre odporúčanie (algoritmus) vytvoreného prídavného modulu. Testy budú slúžiť na zistenie presnosti (správnosti) odporúčania, rýchlosti behu metódy pre odporúčanie a samozrejme na odhalenie prípadných chýb behu metódy. Testy vytvorte pre

rspec. Testy budú spúšťané priamo vo webovej aplikácii. Získajte týmto jednoduché otestovanie integrácie prídavného modulu s webovou aplikáciou.

### 8.3.2 Analýza

Metóda pre odporúčanie môže brať na vstup veľké množstvo parametrov, preto bude potrebné vytvoriť testy pre prácu s každým parametrom. Taktiež niektoré metódy bežia rýchlejšie ako iné, preto je potrebné testovať aj tento problém a samozrejme je potrebné otestovať správnosť odporúčania. Keďže ešte nie sú k dispozícii kompletné dáta, tak niektoré testy na nich nemusia prejsť, aj keď na kompletných dátach by prejsť mohli. Keďže prídavný modul obsahuje štyri rôzne metódy pre odporúčanie, tak pre každú metódu bude vytvorený súbor, ktorý bude obsahovať testy pre danú metódu. Testy v súboroch budú rovnaké, až na názov metódy ktorú volajú.

### 8.3.3 Návrh

Testy budú vytvorené pre rspec. Testovať budú presnosť odporúčania, rýchlosť metódy pre odporúčanie a odhalenie prípadných chýb. Testy pre presnosť odporúčania budú vytvorené podľa ďalej opísaného postupu.

Zadá sa niekoľko vstupných filmov, prípadne hercov a časový limit sa nastaví na 20 sekúnd, aby testovanie nebežalo príliš dlho (testovanie by konzumovalo príliš veľa času), ale nie je ani podmienkou, aby bežalo dostatočne rýchlo (1 sekunda). Potom sa na prvých desiatich filmoch, ktoré metóda vráti, skontroluje, či v sebe obsahujú požadovaný film, ktorý sme taktiež zadali do vstupu testu.

V prípade testovania rýchlosti je spôsob rovnaký, ale časový limit sa nastaví na 1 sekundu. Ako je možné vidieť, tak je potrebné aby metóda odporučila požadovaný film aj pri tomto testovaní. Ak by metóda skončila rýchlo, ale film by neodporučila, tak je rovnako nesprávna, ako keby skončila s tým, že vôbec nezbehla, teda zvládla časový limit, a nič neodporučila. V prípade tretieho testovania sa budú testovať už aj detailnejšie parametre, ktoré môže metóda dostať na vstup ako napríklad, aby odporučila iba niektorý z konkrétnych filmov. Testovať sa bude zadaním detailnejších parametrov (vo forme hešu) do rails metódy pre odporúčanie. Správnosť sa overí rovnako ako v predchádzajúcich testoch.

### 8.3.4 Implementácia

Testy boli implementované tak ako boli opísané v návrhu. Nachádzajú sa priamo vo webovej aplikácii a spúšťajú sa príkazom „rspec cesta\_k\_súboru\_s\_testami“. Po prvom spustení odhalili chybu, ktorú sa ručným testovaním nepodarilo odhaliť. Taktiež niektoré prešli a niektoré nie, preto bude ešte potrebné algoritmy vylepšovať.

## 8.4 Grafické rozhranie č. 1 - Modul odporúčanie TeleVido

### 8.4.1 Úloha

Vytvorte grafické používateľské rozhranie pre projekt odporúčania inteligentnej televízie - **modul odporúčanie TeleVido**. Rozhranie musí spĺňať všetky štandardy modernej webovej stránky a musí zahŕňať všetky základné položky pre korektnú prevádzku uvedeného modulu aplikácie.

### 8.4.2 Analýza

Pred konštruovaním návrhu grafického používateľského rozhrania sme si dôkladne zanalyzovali grafické rozhrania podobne zameraných aplikácií. Poukázali sme na slabé stránky a takisto aj na silné stránky analyzovaných riešení.

### 8.4.3 Návrh

Grafické používateľské rozhranie je zobrazené na obrázku č. 2 a obsahuje nasledovné časti:

**Vrchné menu stránky** V hornej časti stránky sa nachádza hlavné menu aplikácie TeleVido. Menu stránky obsahuje nasledovné položky:

- Odporúčanie TeleVido
- Filmy a seriály
- Môj TV program

V tejto časti dokumentácie sa budeme venovať prvej uvedenej položke - odporúčanie TeleVido.

**Vyhľadávací box** Dominantným prvkom rozhrania je rozsiahly vyhľadávací box určený na vyhľadanie filmov, seriálov, alebo aj mená hercov.

Používateľ prostredníctvom textového poľa zadá začiatočné písmena žiadaného výrazu, a aplikácia TeleVido mu automaticky ponúkne všetky vyhovujúce možnosti prostredníctvom našepkávača. Implementovaný našepkávač je zobrazený na Obr. 15 Používateľ má možnosť určiť si typ vyhľadávacieho algoritmu prostredníctvom zvolenia voľby hneď pod textovým poľom vyhľadávača. Pri zvolení želaného filmu, seriálu, či herca sa následne zadaný výraz zobrazí pod vyhľadávacím poľom so zobrazením krížika pre možnosť odstránenia tejto voľby.

Pre možnosť viacerých druhov vyhľadávání je vo vrchnej časti niekoľko tabov na prepínanie.



Obr. 15: Našepkávač pri vyhľadávaní želaných výrazov

**Vyhľadávací box** Ďalšou významnou časťou opisovaného rozhrania sú výsledky odporúčania zvoleným algoritmom. Každý odporúčaný film obsahuje:

- Názov filmu
- Popis filmu
- Dátum výroby
- Dĺžku trvania
- Krajinu pôvodu
- Réžiu
- Počet zobrazení filmu
- Počet komentárov k filmu
- Obrázok k filmu
- Odkaz na bližšie podrobnosti k filmu

Výsledky odporúčaných filmov je možné zobraziť aj v tabuľkovom rozložení v ktorom sú vynechané niektoré položky. Tabuľkové rozloženie je zobrazené na Obr. 17 Pri nájdení príliš veľa výsledkov je zobrazený číselný posuvník.

**Pätička stránky** V spodnej časti stránky sa nachádza pätička so základnými informáciami o aplikácii.

#### 8.4.4 Implementácia

Grafické používateľské rozhranie bolo implementované v HTML5 a CSS3. Pre vykreslenie rôznych grafických animácií sme využili JavaScript a jQuery. Web aplikácie TeleVido je v súčasnosti k dispozícii na webovej adrese: <http://team01-12.ucebne.fiit.stuba.sk/web/>

**TV TELEVIDEO**    Odporúčanie TeleVido    Filmy a seriály    Mój TV program    Registrácia | Prihlásenie

Odporúčanie TeleVido    BonusTab 1    BonusTab 2    BonusTab 3    BonusTab 4

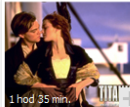
Zadajte film alebo osobu   

Metóda:  Union Colors     Mixing Colors

[Teen Titans: The Judas Contract \(0\) \(film\) ✖](#)  
[Teen Titans: The Judas Contract \(0\) \(film\) ✖](#)

**TeleVido odporúčania**


**Titanic** USA, 1997, 194 min



Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýchla lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžia: [James Cameron](#) | 2,114,568 zobrazení | 10 komentárov    ★★★★★  
[Zobrazit' podrobnosti](#)


**Titanic** USA, 1997, 194 min



Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýchla lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžia: [James Cameron](#) | 2,114,568 zobrazení | 10 komentárov    ★★★★★  
[Zobrazit' podrobnosti](#)

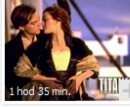
**Titanic** USA, 1997, 194 min



Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýchla lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžia: [James Cameron](#) | 2,114,568 zobrazení | 10 komentárov    ★★★★★  
[Zobrazit' podrobnosti](#)

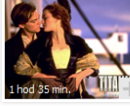
**Titanic** USA, 1997, 194 min



Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýchla lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžia: [James Cameron](#) | 2,114,568 zobrazení | 10 komentárov    ★★★★★  
[Zobrazit' podrobnosti](#)

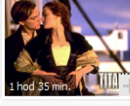
**Titanic** USA, 1997, 194 min



Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýchla lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžia: [James Cameron](#) | 2,114,568 zobrazení | 10 komentárov    ★★★★★  
[Zobrazit' podrobnosti](#)

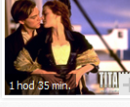
**Titanic** USA, 1997, 194 min



Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýchla lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžia: [James Cameron](#) | 2,114,568 zobrazení | 10 komentárov    ★★★★★  
[Zobrazit' podrobnosti](#)

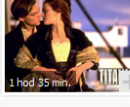
**Titanic** USA, 1997, 194 min



Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýchla lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžia: [James Cameron](#) | 2,114,568 zobrazení | 10 komentárov    ★★★★★  
[Zobrazit' podrobnosti](#)

**Titanic** USA, 1997, 194 min



Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýchla lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžia: [James Cameron](#) | 2,114,568 zobrazení | 10 komentárov    ★★★★★  
[Zobrazit' podrobnosti](#)

◀ 1 2 3 4 5 6 7 ... 44 ▶ Next

Copyright © 2012 [TeleVido](#) - Odporúčanie pre inteligentnú TV   

Obr. 16: Grafické používateľské rozhranie pre modul odporúčanie



**TELEVIDEO** Odporúčanie TeleVido Filmy a seriály Mój TV program Registrácia | Prihlásenie

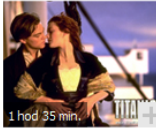
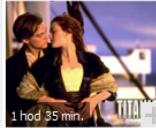
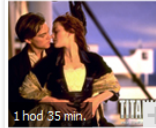


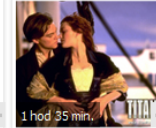
Odporúčanie TeleVido BonusTab 1 BonusTab 2 BonusTab 3 BonusTab 4

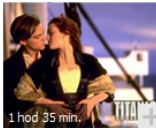
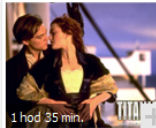
Zadajte film alebo osobu

Metóda:  Union Colors  Mixing Colors

Teen Titans: The Judas Contract (0) (film) ✖  
 Teen Titans: The Judas Contract (0) (film) ✖

**TeleVido odporúčania**

|                                                                                                     |                                                                                                     |                                                                                                     |                                                                                                     |                                                                                                     |                                                                                                      |
|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| <br>1 hod 35 min.  | <br>1 hod 35 min.  | <br>1 hod 35 min.  | <br>1 hod 35 min.  | <br>1 hod 35 min. | <br>1 hod 35 min. |
| <b>Titanic</b><br>USA, 1997, 194 min<br>2,114,568 zobrazení<br>Réžia: <a href="#">James Cameron</a> | <b>Titanic</b><br>USA, 1997, 194 min<br>2,114,568 zobrazení<br>Réžia: <a href="#">James Cameron</a> | <b>Titanic</b><br>USA, 1997, 194 min<br>2,114,568 zobrazení<br>Réžia: <a href="#">James Cameron</a> | <b>Titanic</b><br>USA, 1997, 194 min<br>2,114,568 zobrazení<br>Réžia: <a href="#">James Cameron</a> | <b>Titanic</b><br>USA, 1997, 194 min<br>2,114,568 zobrazení<br>Réžia: <a href="#">James Cameron</a> | <b>Titanic</b><br>USA, 1997, 194 min<br>2,114,568 zobrazení<br>Réžia: <a href="#">James Cameron</a>  |

|                                                                                                     |                                                                                                     |
|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <br>1 hod 35 min. | <br>1 hod 35 min. |
| <b>Titanic</b><br>USA, 1997, 194 min<br>2,114,568 zobrazení<br>Réžia: <a href="#">James Cameron</a> | <b>Titanic</b><br>USA, 1997, 194 min<br>2,114,568 zobrazení<br>Réžia: <a href="#">James Cameron</a> |

◀ 1 2 3 4 5 6 7 ... 44 ▶ Next

Copyright © 2012 TeleVido - Odporúčanie pre inteligentnú TV [Zobraziť vrch stránky](#)

Obr. 17: Zobrazenie odporúčaných výsledkov v tabuľkovom rozložení

## 9 8. Šprint

### 9.1 Synchronizátor

#### 9.1.1 Úloha

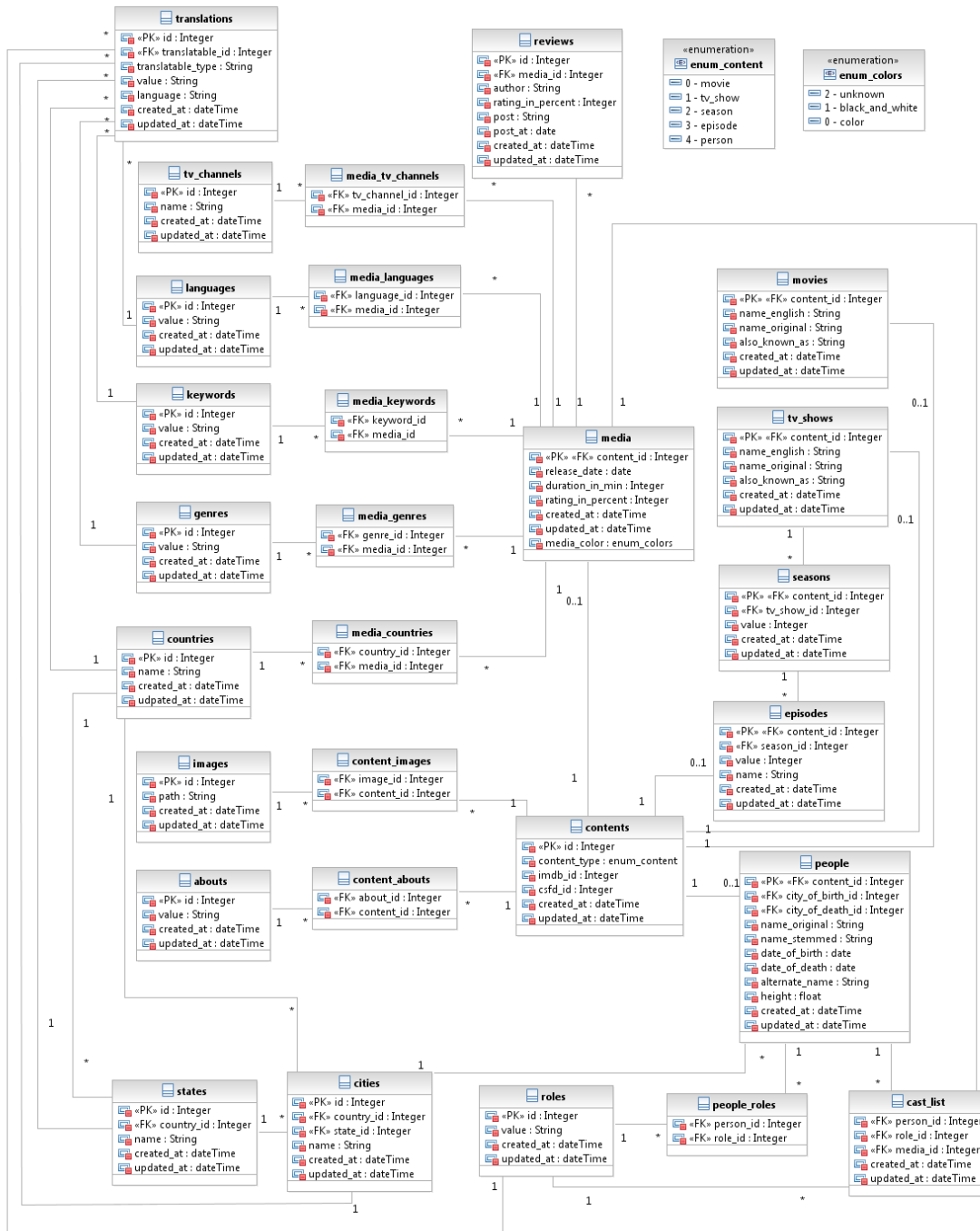
Navrhňte a implementujte synchronizátor pre MySQL a Neo4j databázu. Zdrojom dát bude databáza, ktorú nám poskytol tím 4 (predmet Tímový projekt) a reprezentuje obsah na stránke [www.imdb.sk](http://www.imdb.sk), jedná sa o databázu PostgreSQL. Cieľovou databázou je naša MySQL a Neo4j databáza s odlišným modelom dát.

#### 9.1.2 Analýza

Pred návrhom synchronizátora bolo potrebné dôkladne analyzovať dátový model zdrojového systému - Obr. 19. Tento krok je dôležitý najmä pre správne mapovanie odlišných databáz. V našej databáze sa synchronizátor týka 3 tabuliek: Persons, Movies, Genres - Obr. 18, vzťahy medzi týmito tabuľkami sú reprezentované v Neo4j databáze.

| Genres                                                                                                                             | Movies                                                                                                                                                                                                                                                                                                                                                                                     | Persons                                                                                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>«column» *PK id: INTEGER * genre: VARCHAR(50) neo4j: INTEGER * keywords: TEXT created_at: DATETIME updated_at: DATETIME</pre> | <pre>«column» *PK id: INTEGER neo4j: INTEGER * name: VARCHAR(100) name_original: VARCHAR(100) * created_at: DATETIME * updated_at: DATETIME also_known_as: VARCHAR(100) CSFD_ID: INTEGER IMBD_ID: INTEGER duration: INTEGER rating: REAL release_date: DATE about: TEXT image: TEXT keywords: DATETIME name_svk: VARCHAR(100) accesability: VARCHAR(100) pronanciation: VARCHAR(100)</pre> | <pre>«column» *PK id: INTEGER neo4j: INTEGER name: VARCHAR(50) date_of_birth: DATE city_of_birth: VARCHAR(50) date_of_death: DATE city_of_death: VARCHAR(50) alternate_name: VARCHAR(50) CSFD_ID: INTEGER IMDB_ID: INTEGER about: TEXT * created_at: DATETIME * updated_at: DATETIME keywords: TEXT</pre> |
| <pre>«PK» + PK_Genres(INTEGER)</pre>                                                                                               | <pre>«PK» + PK_Movies(INTEGER)</pre>                                                                                                                                                                                                                                                                                                                                                       | <pre>«PK» + PK_Persons(INTEGER)</pre>                                                                                                                                                                                                                                                                     |

Obr. 18: Tabuľky z cieľovej databázy



Obr. 19: Model dát v zdrojovej databáze

### 9.1.3 Návrh

Synchronizátor sme navrhli ako multivláknovú aplikáciu, ktorá pracuje pomocou Ruby delayed jobs. Ako prvý krok sa vyberie jedna entita. Ak sa jej záznamy v našej databáze nenachádzajú, tak ich najskôr vložíme do Neo4j databázy, aby sme získali ich Neo4j ID. Následne ich pridáme do MySQL databázy. Ak sa už nachádzajú v našej MySQL databáze, tak ich aktualizujeme.

Ďalej je potrebné riešiť vzťahy medzi týmito entitami. Pre každý vložený záznam hľadáme vzťahy v zdrojovej databáze. Tento vzťah potom vytvoríme aj medzi uzlami v Neo4j. Ak sa nenachádza v Neo4j databáze prislúchajúci uzol, tak ho vytvoríme a vytvoríme k nemu záznam v MySQL.

#### 9.1.4 Implementácia

Synchronizátor sme implementovali ako súčasť webovej služby, ktorá je určená pre odporúčanie. Triedy sme implementovali do vlastného modulu Synch. Ako hlavnú databázu v aplikácii používame cieľovú databázu a pomocou deskriptorov prístupujeme k zdrojovej databáze.

---

#### Algoritmus 2 Kód na prístup k zdrojovej databáze

---

```
module Synch
  class Database < ActiveRecord::Base
    establish_connection :synch_database

    def self.image_path(path)
      return "#{path.split("_", 3).third}" unless path.nil?
      nil
    end

    def self.query(sql, params = {})
      where = offset = limit = ""
      where = " WHERE #{params[:where]}" unless params[:where].nil?
      offset = " OFFSET #{params[:offset]}" unless params[:offset].nil?
      limit = " LIMIT #{params[:limit]}" unless params[:limit].nil?
      sql = "#{sql}#{where}#{offset}#{limit}"
      self.connection.execute sql
    end
  end
end
```

---

---

**Algoritmus 3** Postup volaní pri iníciaľnom volaní

---

```
module Synch
  class Load
    def initialize
    end

    def synchronize_entities
      Synch::LoadJobs::SynchMovie.synchronize_all
      Synch::LoadJobs::SynchGenre.synchronize_all
      Synch::LoadJobs::SynchPerson.synchronize_all
    end

    def synchronize_relationships
      Synch::LoadJobs::SynchRelationship.synchronize_all
    end
  end
end
```

---

## Algoritmus 4 Ukázková trieda pre entitu movies

```
module Synch
  module LoadJobs
    class SynchMovie

      LIMIT_PER_PAGE = 100

      def self.synchronize_all
        last_id = ::Movie.last_id
        count = Synch::Movie.count_by_params(:where => "movies.content_id > #{last_id}")
        pages = count / LIMIT_PER_PAGE
        pages += 1 if count % LIMIT_PER_PAGE > 0
        (0..pages-1).each do |page|
          offset = page*LIMIT_PER_PAGE
          delay.synchronize_page(last_id, offset)
        end
      end

      def self.synchronize_page(last_id, offset)
        Synch::Movie.find_all_by_params(:where => "movies.content_id > #{last_id}",
                                       :offset => offset, :limit => LIMIT_PER_PAGE).each do |synch_movie|
          delay.synchronize_movie(synch_movie)
        end
      end

      def self.synchronize_movie(synch_movie)
        movie_id = synch_movie['id'].to_i
        movie = ::Movie.find_or_initialize_by_id(movie_id)
        if movie.new_record?
          movie.neo4j_id = Wrapper.insert_node({:movie_id => movie_id})
        end
        movie.name_original = synch_movie['name_original']
        movie.name_english = synch_movie['name_english']
        movie.also_known_as = synch_movie['also_known_as']
        movie.CSFD_ID = synch_movie['csfd_id']
        movie.IMDB_ID = synch_movie['imdb_id']
        movie.duration = synch_movie['duration']
        movie.rating = synch_movie['rating']
        movie.release_date = synch_movie['release_date']
        movie.about = synch_movie['about']
        movie.image = Synch::Movie.image_path synch_movie['image']
        movie.keywords = synch_movie['keywords']
        movie.save
      end
    end
  end
end
```

## 9.2 Grafické rozhranie č. 2

### 9.2.1 Úloha

Vytvorte grafické používateľské rozhranie pre projekt odporúčania inteligentnej televízie - modul zoznamy a detaily filmov TeleVido. Rozhranie musí spĺňať všetky štandardy modernej webovej stránky a musí zahŕňať všetky základné položky pre korektnú prevádzku uvedeného modulu aplikácie.

### 9.2.2 Analýza

Pred konštruovaním návrhu grafického používateľského rozhrania sme si dôkladne zanalyzovali grafické rozhrania podobne zameraných aplikácií. Poukázali sme na slabé stránky a takisto aj na silné stránky analyzovaných riešení.

### 9.2.3 Návrh

Grafické používateľské rozhranie pre zoznam filmov a seriálov je zobrazený na Obr. 20, a grafické používateľské rozhranie pre detail filmu je zobrazený na Obr. 21.

V tejto časti si opíšeme jednotlivé dva typy rozhraní:

- **Zoznamy filmov a seriálov**


V hornej časti stránky sa nachádza veľký box, v ktorom sú umiestnené všetky kategórie filmov a seriálov rozdelené do dvoch zložiek. Používateľ má možnosť kliknúť na ľubovoľnú kategóriu s cieľom zobrazíť všetky relevantné výsledky spadajúce pod túto voľbu. Následne sa používateľovi zobrazia všetky výsledky spĺňajúce podmienky kategórie. Zobrazenie zoznamu filmov je totožné s výsledkami odporúčaných filmov, ktoré sme si opísali v prvej časti tvorby rozhrania. Aj tu je rovnaká možnosť zobrazíť filmy a seriály v tabuľkovom rozložení. V prípade, že používateľ má záujem vyhľadať konkrétny film alebo seriál, tak má možnosť napísať žiadané kľúčové slová do vyhľadávacieho poľa vo vrchnej časti boxu, v ktorom sa nachádza zoznam jednotlivých kategórií.

- **Detail filmu**

Pri detaile filmu ponechávame pre rýchly prístup zobrazený box s jednotlivými kategóriami filmov či seriálov. Pri detaile filmu zobrazujeme nasledujúce položky:

- Názov filmu
- Opis filmu
- Dátum výroby
- Dĺžku trvania
- Krajinu pôvodu
- Hodnotenie filmu
- Réžiu
- Počet zobrazení filmu
- Obrázok k filmu
- A ďalšie

Po týchto informáciách k detailu filmu zobrazíme aj vložené komentáre, ktoré obsahujú položky ako meno autora príspevku, čas odoslania správy, hodnotenie komentára a samostatný textový príspevok. Komentáre bude možné zobrazovať podľa rôznych kritérií, ako napríklad podľa času, hodnotenia a podobne.


Odporúčanie TeleVido
Filmy a seriály
Môj TV program
Registácia | Prihlásenie

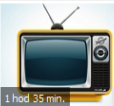
---

Zoznamy filmov
Zoznamy seriálov

**Katégorie** Vyhľadavanie filmov Nájdí

|                   |                   |                   |                   |                      |                   |
|-------------------|-------------------|-------------------|-------------------|----------------------|-------------------|
| ▶ Action          | ▶ Animals         | ▶ Art & Animation | ▶ Commercials     | ▶ Entertainment & TV | ▶ Family          |
| ▶ Food            | ▶ Funny Videos    | ▶ Games           | ▶ Health & Beauty | ▶ How-To             | ▶ Movies & Shorts |
| ▶ Music           | ▶ News & Politics | ▶ People & Vlogs  | ▶ Products & Tech | ▶ Environment        | ▶ Sports          |
| ▶ Transportation  | ▶ Travel          | ▶ Movies & Shorts | ▶ Music           | ▶ News & Politics    | ▶ People & Vlogs  |
| ▶ Products & Tech | ▶ Environment     | ▶ Sports          | ▶ Transportation  | ▶ Travel             |                   |

**TeleVido filmy pre kategóriu: Humor** ☰ ☲



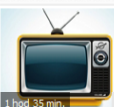
1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)




1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)




1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)




1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)




1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)



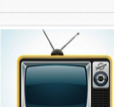
1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)



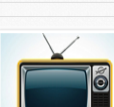
1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)



1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★


[Zobrazit podrobnosti](#)

◀ 1 2 3 4 5 6 7 ... 44 ▶ [Ďalej](#)

Copyright © 2012 [TeleVido - Odporúčanie pre inteligentní TV](#) [Zobrazit vrch stránky](#)

Obr. 20: Grafické používateľské rozhranie pre modul zoznam filmov




Odporúčanie TeleVido
Filmy a seriály
Môj TV program
Registácia | Prihlásenie

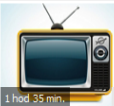
---

Zoznamy filmov
Zoznamy seriálov

**Kategórie** Vyhľadávanie filmov Nájdí

|                   |                   |                   |                   |                      |                   |
|-------------------|-------------------|-------------------|-------------------|----------------------|-------------------|
| ▶ Action          | ▶ Animals         | ▶ Art & Animation | ▶ Commercials     | ▶ Entertainment & TV | ▶ Family          |
| ▶ Food            | ▶ Funny Videos    | ▶ Games           | ▶ Health & Beauty | ▶ How-To             | ▶ Movies & Shorts |
| ▶ Music           | ▶ News & Politics | ▶ People & Vlogs  | ▶ Products & Tech | ▶ Environment        | ▶ Sports          |
| ▶ Transportation  | ▶ Travel          | ▶ Movies & Shorts | ▶ Music           | ▶ News & Politics    | ▶ People & Vlogs  |
| ▶ Products & Tech | ▶ Environment     | ▶ Sports          | ▶ Transportation  | ▶ Travel             |                   |

**TeleVido filmy pre kategóriu: Humor** ☰ ☱



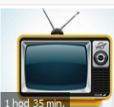
1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)




1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)




1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)




1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)




1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)



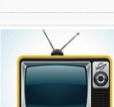
1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)



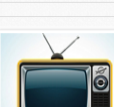
1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)



1 hod 35 min.

**Titanic** USA, 1997, 194 min

Byl obrovský a luxusní. Lidé o něm ve své pýše říkali, že je nepotopitelný. Když vyplouval Titanic na svou první plavbu, byli na jeho palubě také chudý Jack a bohatá Rose. On vyhrál lístek v pokeru, ona měla pronajato jedno z nejluxusnějších apartmá. Prožili spolu nejkrásnější chvíle života a slibli si, že už se nikdy nerozejdou - až do oné osudné noci, kdy pýcha lidstva narazila v Severním moři do ledovce, který Titanic neúprosně poslal ke dnu.

Réžie: [James Cameron](#) | 2,114,568 zobrazení | 10 komentářů ★ ★ ★ ★ ★

[Zobrazit podrobnosti](#)

◀ 1 2 3 4 5 6 7 ... 44 ▶ [Ďalej](#)

Copyright © 2012 [TeleVido - Odporúčanie pre inteligentní TV](#) [Zobrazit vrch stránky](#)

Obr. 21: Grafické používateľské rozhranie pre modul detail filmu

#### 9.2.4 Implementácia

Grafické používateľské rozhranie bolo implementované v HTML5 a CSS3. Pre vykreslenie rôznych grafických animácií sme využili JavaScript a jQuery. Web aplikácie TeleVido je v súčasnosti k dispozícii na webovej adrese: <http://team01-12.ucebne.fiit.stuba.sk/web/>

### 9.3 Nasadenie grafického rozhrania

#### 9.3.1 Úloha

Implementujte a nasadte navrhnuté grafické rozhranie aplikácie. Aplikácia má poskytovať požadovanú funkčnosť prostredníctvom tohto rozhrania.

#### 9.3.2 Analýza

Návrh rozhrania pozostáva z niekoľkých HTML súborov a k nim prislúchajúcim CSS a JavaScriptov. HTML súbory bude nutné rozčleniť a zakomponovať do nich zobrazovanie požadovaného obsahu.

Aplikácia v súčasnosti neposkytuje požadovanú funkčnosť, je potrebné ju vyvinúť popri nasadzovaní rozhrania.

#### 9.3.3 Návrh

Pre aplikáciu používame pracovný rámec Ruby on Rails, takže grafické rozhranie implementujeme do Rails aplikácie pomocou pridávania a upravovania views a pridávania a upravovania assets.

Funkčnosť implementujeme v controllers a models. Pre správnu funkčnosť je potrebné nastaviť konfiguráciu ciest (routes).

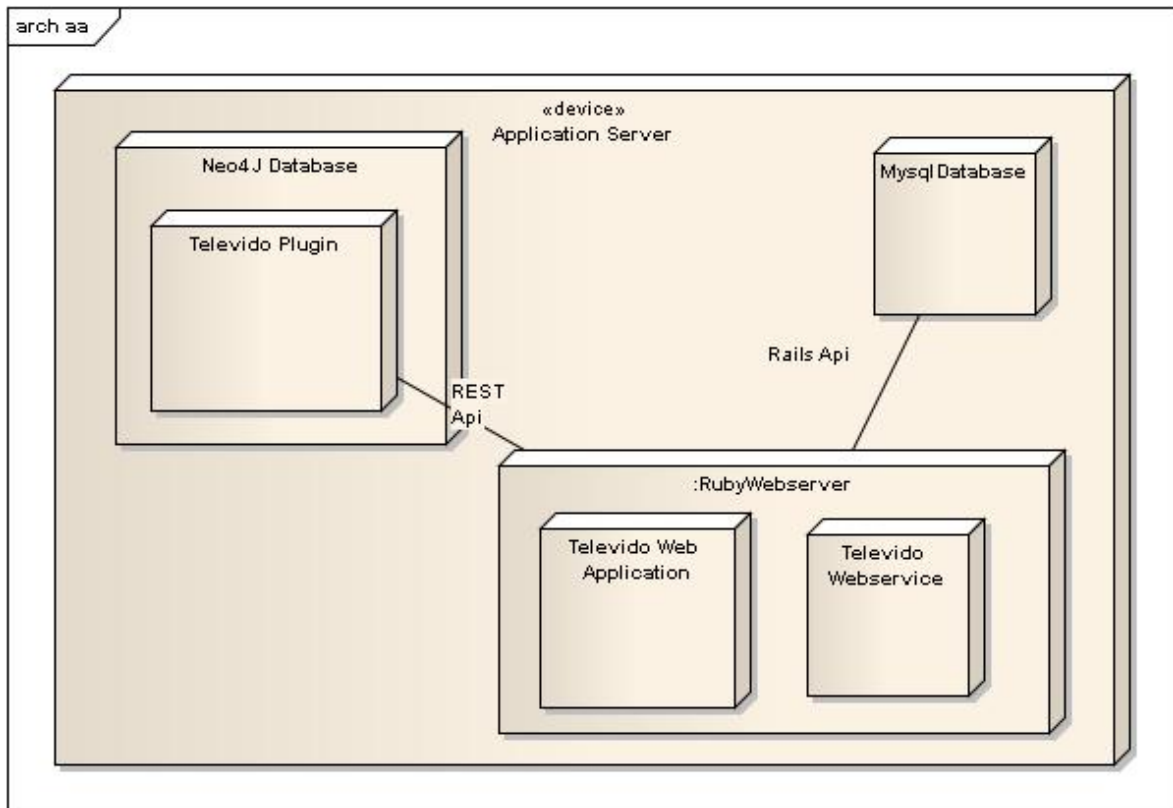
#### 9.3.4 Implementácia

Rozhranie a požadovanú funkčnosť sme implementovali v jazyku Ruby v pracovnom rámci Ruby on Rails. Aplikácia komunikuje s databázami MySQL pomocou ActiveRecord a SQL a s Neo4j pomocou API volaní.

## 10 Zimný semester

### 10.1 Architektúra aplikácie

Naša aplikácia je vytvorená prostredníctvom viacerých technológií a to odráža aj jej architektúra. Architektúru delíme na tri základné komponenty - Obr. 22.



Obr. 22: Architektúra aplikácie

#### 10.1.1 Neo4j databáza

Neo4j databáza je grafová databáza založená na jazyku Java. V našej aplikácii slúži na reprezentáciu vzťahov medzi hlavnými entitami. Nad jej dátami je vytvorený zásuvný modul s názvom Televideo. Ten pozostáva z viacerých zásuvných modulov, ktoré sú implementované v Jave a komunikujú s Neo4j databázou prostredníctvom natívnych knižníc.

#### 10.1.2 MySQL databáza

MySQL databáza je relačná databáza, v ktorej sú uložené atribúty k entitám z Neo4j databázy a k nim prislúchajúce Neo4j ID. Jej existencia znamená hlavne menšiu záťaž grafovej databázy.

### 10.1.3 Ruby Webservice

Televido Webservice je wrapper, ktorý volá základnú funkcionálnosť a funkcionálnosť zásuvného modulu Televido pre Neo4j. Služi na komunikáciu Rails Webservice s databázou Neo4j.

Televido Web Application je webová aplikácia, ktorá je sprístupnená používateľovi. Obsahuje používateľské rozhranie pre zobrazovanie odporúčaní a ostatných dát.

### 10.1.4 Komunikácia medzi komponentmi

Ruby Web Server  $\longleftrightarrow$  Televido Plugin: komunikácia prebieha prostredníctvom REST volaní, ktoré ponúka štandardne Neo4j databáza pre komunikáciu so zásuvnými modulmi. Táto komunikácia je postavená na báze JSON objektov, ktoré sú posielané ako požiadavka aj ako odpoveď.

Mysql Databáza  $\longleftrightarrow$  Ruby Web Server: komunikácia prebieha so štandardným Ruby on Rails API.

## 10.2 Dáta

Využívali sme dve databázy - grafovú Neo4j a relačnú MySQL, ktoré boli prepojené pomocou jedinečných ID. Dáta v databázach sme v prvej časti semestra mali z vlastných zdrojov, ktoré sme získali vyparovaním menšej vzorky dát. V druhej časti semestra sme získali prístup k dátam od druhého tímu (tím #4). Tieto dáta bolo potrebné pomocou ETL procesov synchronizovať s našimi dvomi databázami.

## 10.3 Neo4j pluginy

V priebehu semestra sme vytvorili niekoľko pluginov, ktoré efektívne vyhľadávali vzťahy v grafovej databáze. Využili sme nato modifikované grafové algoritmy - prehľadávanie do šírky, union find a Dijkstrov algoritmus. Pluginy následne využívame pri službe.

## 10.4 Služba

Počas prvého semestra sme vyvíjali webovú službu, ktorá odporúča filmy podľa zadaných parametrov. Služba je dostupná pomocou webového rozhrania a komunikuje pomocou formátu JSON. Webová služba slúži ako základ k odporúčaniam, ktoré bude integrované ako webová aplikácia alebo mobilná aplikácia (v letnom semestri). Vytvorená služba dostane na vstup niekoľko inicializačných vrcholov v grafe a snaží sa nájsť množinu vrcholov, ktoré sú zo zadaných vrcholov najbližšie. Viac o službe nájdeme v dokumentáciách o jednotlivých šprintoch.

## 10.5 Webová aplikácia

Webová aplikácia je realizovaná v jazyku **Ruby** a s použitím rámca **Ruby on Rails**. Grafické používateľské rozhranie je implementované v technológiách **HTML5** a **CSS3**. Rozhranie spĺňa všetky štandardy modernej webovej stránky a zahŕňa všetky základné položky pre korektnú prevádzku opisovaných funkcionálností. Pre vykreslenie rôznych grafických animácií a zvýšenie

interaktivity sme využili **JavaScript** a rámec **jQuery**. Aplikácia používa databázový systém **MySQL** a, pomocou API volaní, **Neo4j**.

Aplikácia TeleVido je v súčasnosti k dispozícii na webovej adrese: <http://team01-12.ucebne.fiit.stuba.sk/web/> a je výsledkom práce v zimnom semestri s nasledujúcou funkcionalitou:

### 10.5.1 Odporúčanie filmov

V tejto sekcii sa nachádza rozsiahly vyhľadávací box určený na vyhľadanie filmov, seriálov, alebo aj mená hercov. Používateľ prostredníctvom textového poľa zadá začiatkové písmena žiadaného výrazu, a aplikácia TeleVido mu automaticky ponúkne všetky vyhovujúce možnosti prostredníctvom našepkávača.

Používateľ má možnosť určiť si typ vyhľadávacieho algoritmu prostredníctvom zvolenia voľby hneď pod textovým poľom vyhľadávača. Pri zvolení želaného filmu, seriálu, či herca sa následne zadaný výraz zobrazí pod vyhľadávacím poľom so zobrazením krížika pre možnosť odstránenia tejto voľby.

Výsledky odporúčaných filmov či seriálov je možné zobrazit' aj v tabuľkovom rozložení v ktorom sú vynechané niektoré položky. Pri nájdení príliš veľa výsledkov je zobrazený číselný posuvník.

### 10.5.2 Zoznamy filmov a seriálov

V tejto sekcii sa nachádza veľký box, v ktorom sú umiestnené všetky kategórie filmov a seriálov rozdelené do dvoch zložiek. Používateľ má možnosť kliknúť na ľubovoľnú kategóriu s cieľom zobrazit' všetky relevantné výsledky spadajúce pod túto voľbu.

Následne sa používateľovi zobrazia všetky výsledky spĺňajúce podmienky kategórie. Zobrazenie zoznamu filmov je totožné s výsledkami odporúčaných filmov. Aj tu je rovnaká možnosť zobrazit' filmy a seriály v tabuľkovom rozložení.

V prípade, že používateľ má záujem vyhľadať konkrétny film alebo seriál, tak má možnosť napísať žiadané kľúčové slová do vyhľadávacieho poľa vo vrchnej časti boxu, v ktorom sa nachádza zoznam jednotlivých kategórii.

### 10.5.3 Detaily k jednotlivým filmom a seriálom

Pri detaile filmu je ponechaný pre rýchly prístup box s jednotlivými kategóriami filmov či seriálov. Pri detaile filmu zobrazujeme nasledujúce položky:

- Názov filmu
- Opis filmu
- Dátum výroby
- Dĺžku trvania
- Krajinu pôvodu

- Hodnotenie filmu
- Réžiu
- Počet zobrazení filmu
- Obrázok k filmu
- A ďalšie

Po týchto informáciách k detailu filmu zobrazíme aj vložené komentáre, ktoré obsahujú položky ako meno autora príspevku, čas odoslania správy, hodnotenie komentára a samostatný textový príspevok.

## 11 9. Šprint

### 11.1 ČSFD parser

#### 11.1.1 Úloha

Navrhните a implementujte automatický sťahovač údajov o filmoch a osobách zo stránky csfd.cz. Údaje predstavujú základné informácie o filmoch a osobách, videá k filmom a prepojenia medzi filmami a osobami a filmami a žánrami.

#### 11.1.2 Analýza

Zoznam žánrov je jednoducho dostupný na samostatnej stránke, odkiaľ je ho možné vyparsovať. Na stránke osoby sa nachádzajú základné informácie o osobe – meno, dátum narodenia, biografia a podobné. Tieto údaje je možné jednoducho vyparsovať. Na stránke filmu sa nachádzajú základné informácie o filme – názvy vo viacerých jazykoch, hodnotenie, obsah a pod. Nachádzajú sa tu aj údaje o prepojeniach s osobami, ktoré vieme zisťovať podľa jednoznačného identifikátora ID a o prepojeniach zo žánrami, ktoré identifikujeme podľa názvu. Problém pri parsovaní filmu je s názvami, ktoré sú uvádzané rôzne podľa pôvodu filmu a dostupnosti. Pri uvedení jediného názvu nie je možné identifikovať, o aký typ ide. Všeobecný problém je možnosť, že niektorý parameter nebude pri konkrétnom filme alebo osobe uvedený. Parser musí byť schopný takéto parametre ignorovať a zvyšok extrahovať správne.

#### 11.1.3 Návrh

Zo stránky žánrov extrahujeme názvy žánrov. Zo stránky osoby extrahujeme nasledovné parametre: meno, biografia, dátum narodenia, miesto narodenia, dátum smrti, miesto smrti, imdb ID, csfd ID, csfd obrázok. Zo stránky filmu extrahujeme nasledovné parametre: kategória, originálny názov, slovenský názov, český názov, obsah, rok vydania, dĺžka v minútach, imdb ID, csfd ID, hodnotenie csfd, počet hodnotení csfd, obrázok na csfd. Vytvoríme prepojenia s osobami a žánrami. Zo samostatnej stránky k filmu vyparsujeme adresy trailerov k filmom, ich dĺžku a názov. Názvy filmov extrahujeme s použitím heuristiky, ktorá nebude vždy presná ale mala by postačovať pre naše potreby.

#### 11.1.4 Implementácia

Parsery boli implementované v jazyku Ruby ako modul do existujúcej aplikácie v Ruby on Rails. Na sťahovanie a parsovanie obsahu boli použité gemy require-uri a nokogiri. Na paralelné vykonávanie extrakcie bol použitý gem delayed\_jobs.

### 11.2 Pridanie používateľa do systému

#### 11.2.1 Úloha

Implementujte a nasadte funkcionality pre používateľa. Aplikácia má poskytovať požadovanú funkcionality prostredníctvom tohto rozhrania.

#### 11.2.2 Analýza

Najdôležitejšie je rozmyslieť, akú funkcionality budeme pre používateľa poskytovať. Pre naše potreby a analýze podobných projektov by som funkcionality rozdelil nasledovne:

- Registrácia - Používateľ má možnosť sa registrovať po zadaní potrebných údajov
- Prihlásenie - Používateľ sa prihlási potom, ako sa registroval
- Odhlásenie - Používateľ má možnosť sa odhlásiť
- Profil - Používateľ môže zobraziť alebo upraviť svoj profil
  - Úprava údajov
  - Zmena hesla
  - Zobrazenie profilu
- Zabudnuté heslo - Ak používateľ nevie heslo systém mu ho vie resetnúť
- Hodnotenie obsahu
  - Hodnotenie ľudí - Používateľ hodnotí ľudí
  - Hodnotenie filmov - Používateľ hodnotí filmy
- Logovanie používateľa - Logovanie na pozadí kde používateľ klikol a čo urobil
- Extrahovanie záujmu používateľa
  - Extrakcia klikania na filmy
  - Extrakcia klikania na ľudí

### 11.2.3 Návrh

Pri registrácii sme sa rozhodli, že potrebujeme tieto údaje: meno, priezvisko, e-mail a heslo. Prihlasovacie údaje sú e-mail a heslo. Pri hodnotení obsahu sme sa rozhodli pre 4-bodovú škálu. Pri logovaní používateľa sme sa kvôli kapacite zatiaľ rozhodli ukladať len navštívené filmy a ľudí.

### 11.2.4 Implementácia

Pre aplikáciu používame pracovný rámec Ruby on Rails, takže grafické rozhranie implementujeme do Rails aplikácie pomocou pridávania a upravovania views a pridávania a upravovania assets. Funkcionalitu implementujeme v controllers a models. Pre správnu funkčnosť je potrebné nastaviť konfiguráciu ciest (routes). Pri registrácii sme využili recaptcha od Google a na extrakciu používateľského záujmu sme využili multivláknové spracovanie pomocou gemu `delayed_jobs`.

## 11.3 Prihlásenie, registrácia, profil filmu / osoby čo som hodnotil, čo ma zaujíma

### 11.3.1 Úloha

Vytvorte grafické webové rozhranie, ktoré umožní registráciu a prihlásenie používateľa a následné zobrazenie jeho základných údajov v systéme. Tiež vytvorte zobrazenie profilu filmu a osoby – to zahŕňa hercov, režisérov a filmových skladateľov.

### 11.3.2 Analýza

Grafické používateľské rozhranie by malo byť intuitívne pre používateľa a zároveň by malo zapadnúť do grafického rozhrania stránky. Prvým krokom bola analýza rôznych používateľských rozhraní vhodných pre prihlásenie a rôznych systémov zameraných na odporúčanie filmov. Pritom sme sa snažili identifikovať spoločné prvky týchto rozhraní a prvky vhodné do nášho používateľského rozhrania.

### 11.3.3 Návrh

#### Prihlásenie

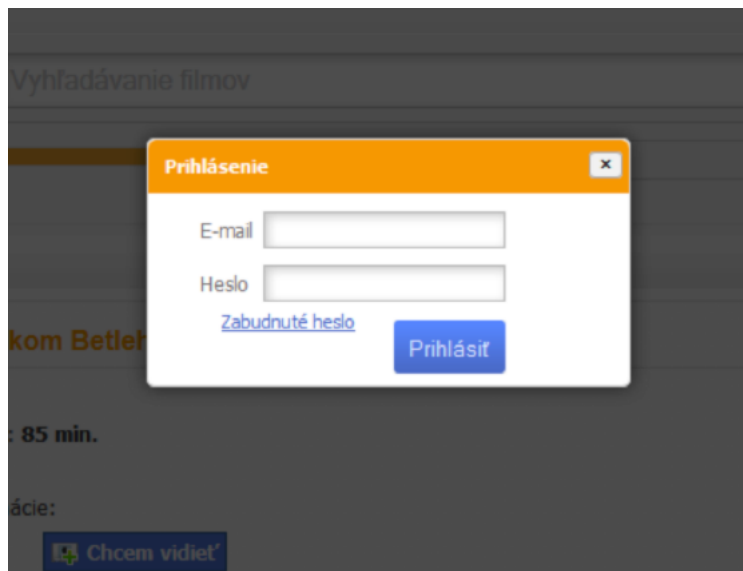
Pre prihlásenie sme sa rozhodli využiť formu tzv. vyskakovacieho okna (pop-up), v ktorom sa bude nachádzať prihlasovací formulár. Pre prípad, že budú zadané zlé údaje alebo používateľ zabudne svoje heslo bolo potrebné vytvoriť aj prihlasovaciu stránku. Pre prihlasovaciu stránku bolo z záhlaví použité existujúce navigačné menu a telo stránky tvorí identický formulár ako vo vyskakovacom okne. Použitím existujúceho menu sme chceli zaistiť vyššiu intuitívnosť a konzistenciu. Navrhnuté rozhrania môžete vidieť na obrázkoch 23 a 24. Registrácia V záhlaví registrácie sa opäť nachádza navigačné menu, identické s prihlasovaním. Vstupné polia majú rovnaký dizajn ako pri prihlasovaní. Formulár bol rozšírený o niekoľko vstupných polí a tiež



bolo nutné zakomponovať prvok CAPTCHA (Obr. 25). Používateľský profil Používateľský profil sme rozdelili do niekoľkých častí, a to:

- Profil
- Úprava profilu
- Čo ma zaujíma
- Čo som hodnotil

Pre navigáciu medzi týmito sekciami sme využili už spomínaný existujúci štýl pre navigáciu. V profile bolo nutné zobraziť základné informácie o používateľskom účte s ohľadom na to, že táto sekcia sa bude časom rozširovať (Obr. 26). Úprava profilu zahŕňa zmenu základných informácií o účte používateľa a to úpravu profilu a úpravu hesla. Pre vyššiu prehľadnosť sme sa rozhodli rozdeliť stránku do dvoch častí a tie zobraziť vedľa seba, tak ako to môžete vidieť na Obr. 27. Sekcia Čo ma zaujíma vznikla za účelom zobrazenia filmov a osôb, ktoré používateľ v poslednom čase navštívil. Preto, že sa počítalo s tým, že táto stránka bude zahŕňať veľké množstvo rôznych titulov a osôb, bola rozdelená na dve časti: osoby a filmy. Taktiež z tohto dôvodu bolo potrebné vytvoriť spôsob zobrazovania ďalších podstránok. Rozhodli sme sa, že pre tento účel využijeme prvok nekonečné rolovanie. Pre lepšiu orientáciu na stránke sme ku každému titulu / osobe pridali tiež obrázok a ďalšie informácie (počet navštívení a dátum posledného navštívenia) tak, ako to môžeme vidieť na obrázku Obr. 28. Sekcia Čo som hodnotil je významovo podobná predchádzajúcej sekcii – zobrazuje však filmy a osoby, ktoré používateľ hodnotil. Kvôli tejto podobnosti sme sa rozhodli využiť rovnaké grafické prvky aj pri tejto sekcii – Obr. 29.



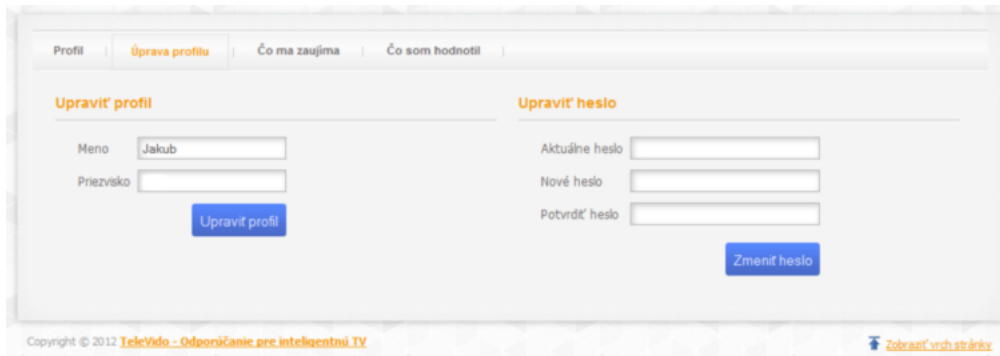
Obr. 23: Prihlasovanie – vyskakovacie okno

Profil filmu a takisto osoby bolo potrebné vytvoriť s ohľadom na existujúce systémy. Teda tak, aby v ňom mohol používateľ nájsť všetky potrebné informácie rýchlo. Preto sme ho

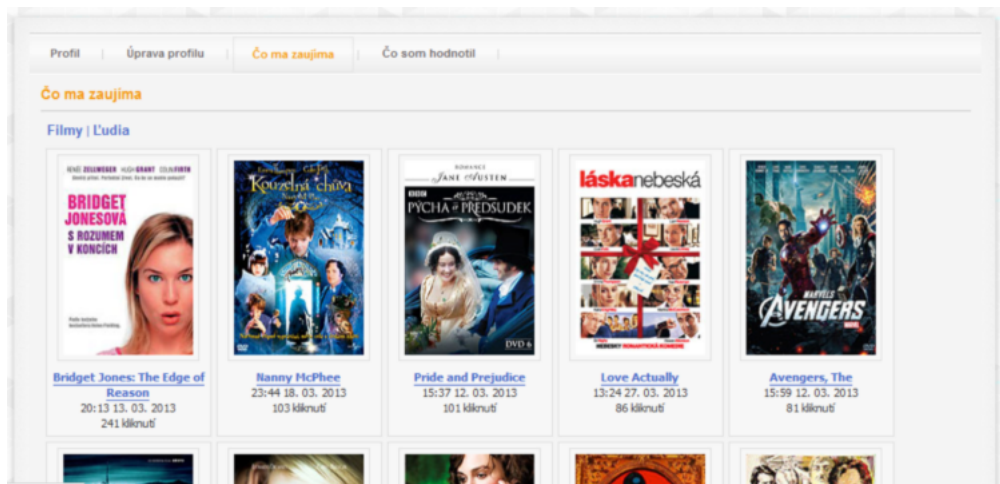
Obr. 24: Prihlasovanie – stránka

Obr. 25: Registrácia

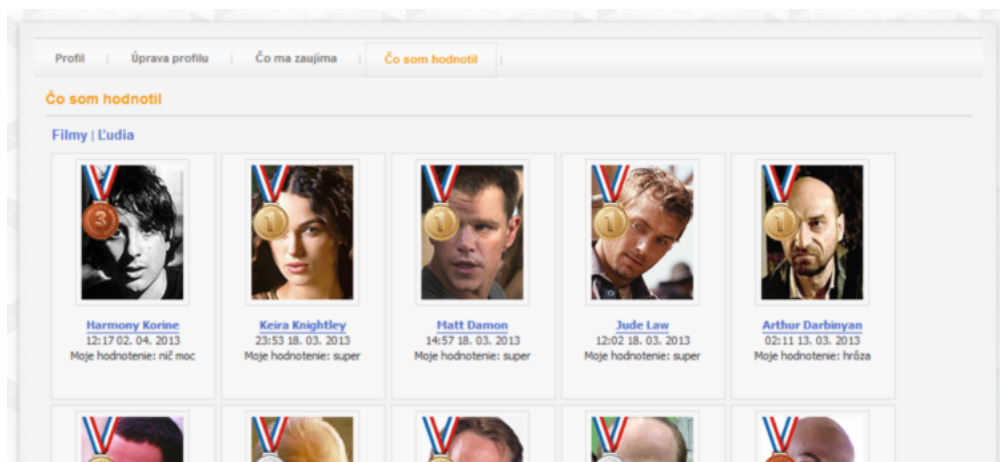
Obr. 26: Profil používateľa



Obr. 27: Úprava informácií z profilu



Obr. 28: Sekcia Čo ma zaujíma - Filmy



Obr. 29: Sekcia Čo som hodnotil – Ľudia



Obr. 30: Profil filmu

vytvorili tak, že v hornej časti sa nachádzajú základné informácie o filme spolu s titulným obrázkom, vpravo má používateľ možnosť hodnotiť film a vidieť jeho hodnotenie v rôznych systémoch a v dolnej časti sa nachádzajú rozšírené informácie (popis, herci a pod). V tejto časti bolo dôležitejšie ako v iných používanie rôznych ikoniek, aby umožnilo zmenšiť množstvo informácií na stránke, ale nie výpovednú hodnotu pre používateľa. Spôsob hodnotenia musel byť názorný a nenáročný, preto sme si boli istý, že sa bude jednať o priradovanie obrázkov ako hodnotenia. Nechceli sme klasický hviezdíčkový systém, skôr niečo nové a charakteristické, čo však ľudia poznajú – zvolili sme medaile. V dolnej časti sa nachádza menu, ktorého časti sa zobrazia bez opätovného načítania stránky (Obr. 30).

Profil osoby je veľmi podobný profilu filmu. Najväčšie rozdiely sú tie, že osoby nemajú ČSFD hodnotenie a množstvo ich podsekcí závisí od samotnej osoby (či bol režisér aj herec aj skladateľ).

### 11.3.4 Implementácia

Pre implementáciu boli použité technológie HTML a CSS, pre vykreslenie animácií a ďalších prvkov sme použili JavaScript a jQuery. Pre dynamické prvky bol použitý jazyk Ruby.

## 11.4 Komunikácia medzi serverom a android aplikáciou

### 11.4.1 Úloha

Implementuje a nasadfe funkcionalitu v android aplikácii na základa čoho sa bude diať komunikácia medzi aplikáciou a servervom pomocou API.

### 11.4.2 Analýza

Komunikačnú vrstvu je potrebné navrhnuť tak aby vyhovovala týmto základným bodom: • Musi byť do istej miery abstraktná aby sa to ľahko rozširovalo keď pribudnú nové funkcionality



Obr. 31: Profil osoby

- Musí fungovať pomocou adapterov aby bolo možné manipulovať s prichádzajúcimi a odchádzajúcimi správami nezávisiac od servisnej vrstvy.
- Pridávanie nových funkcionalít musí byť rutinná práca len s použitím už existujúcich komponentov.

### 11.4.3 Návrh

Pri návrhu komunikačnej vrstvy sme sa rozhodli rozdeliť funkcionalitu na ďalšie 2 vrstvy. Prvá vrstva je servisná kde sa zavolá služba, zaregistrujú sa adaptory pre parsovanie správ a vykonajú sa dodatočné manipulácie s entitami. Druhá vrstva je nízko úrovňová vrstva kde samotná komunikácia sa vykoná pomocou štandardných java 1.6 tried.

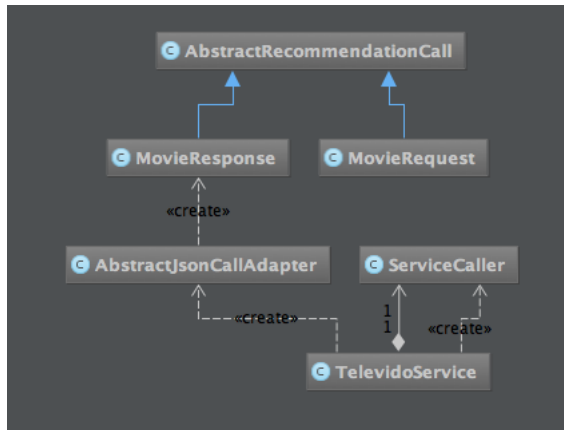
### 11.4.4 Implementácia

Pre parsovanie správ sme sa rozhodli používať externú knižnicu GSON ktorú poskytuje spoločnosť google. Pointa použitia tejto knižnice že dokáže JSON správy rovno mapovať na Java objekty a umožňuje registrovať adaptory k parserom pre flexibilnejšie používanie.

Priebeh implementácie volanie jednej webovej služby:

- Vytvorila sa response a request triedy ktoré budú reprezentovať inštancie JSON správ
- Do servisnej triedy sa pridá volanie služby kde sa použije nízko úrovňová komunikačná trieda ktorú sme navrhli.
- Vytvorí sa vlastný adapter pre parsovanie správ alebo sa rozšíri existujúci aby rozpoznávalo inštancie response a request tried.

No obrázku 32. je diagram tried pre volanie služby.



Obr. 32: Diagram tried pre volanie služby

## 11.5 Návrh používateľského rozhrania Android aplikácie

### 11.5.1 Úloha

Navrhnete používateľské rozhranie mobilnej aplikácie pre android.

### 11.5.2 Analýza

Pred návrhom používateľského rozhrania sme zanalyzovali existujúce riešenia obdobných aplikácií na android ale aj na iOS. Zhodnotili sme aké funkcie by mala naša aplikácia obsahovať.

### 11.5.3 Návrh

Navrhli sme dve obrazovky používateľského rozhrania:

- Obrazovka zoznamu filmov
- Obrazovka detailu filmu

Obrazovka zoznamu filmov (obrázok č. 33.) je navrhnutý ako posuvný nekonečný list, kde sa používateľovi zobrazujú odporúčené filmy.

Detail filmu (obrázok č 34) zobrazuje základné informácie o filme ako aj galériu.

## 12 10. Šprint

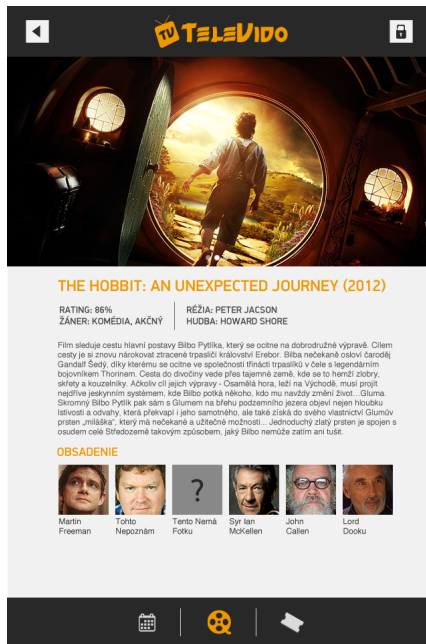
### 12.1 Nasadenie Cronov

#### 12.1.1 Úloha

Nasadte automatické spúšťanie procesov zo služby - extrakcia logov, sťahovanie ČSFD, sťahovanie TV Programu, sťahovanie kín a aktualizácia Neo4j.



Obr. 33: Obrazovka zoznamu filmov



Obr. 34: Obrazovka detailu filmu

### 12.1.2 Analýza

Identifikovali sme tieto procesy, ktoré by sa mali automatizovane spúšťať:

- Extrakcia logov používateľa (každých 10 minút)
- Aktualizovanie entít na odporúčanie pre používateľa (každých 10 minút)

Pre ČSFD dáta bolo potrebné vytvoriť systematické plánovač, pretože niektoré sťahovania sú závislé od iných:

- Sťahovanie ľudí a žánrov z ČSFD (každý týždeň o 3:00)
- Sťahovanie filmov z ČSFD (potom ako sa skončí sťahovanie ľudí a žánrov)
- Aktualizácia Neo4j (potom ako sa dokončí sťahovanie filmov)
- Sťahovanie TV programu z ČSFD (každý deň 4:00)
- Sťahovanie Programu kín (každý deň o 4:00)

### 12.1.3 Návrh a implementácia

V prvej časti sme vytvorili scheduler v službe, ktorý zabezpečil gem whenever. Následne sme exportovali naplánované akcie pre crontab. Nasadenie do crontabov bolo komplikovanejšie, pretože Fedora server sa správa inak ako ostatné linuxové serveri a bolo potrebné manuálne upraviť crontaby.

## 12.2 API

### 12.2.1 Úloha

Implementujte a nasadte funkcionality API do aplikácie. API by malo byť prístupné pre POST a GET requesty vo formáte JSON.

### 12.2.2 Analýza

Pre potreby android aplikácie a tiež poskytovania API pre verejnosť sme vybrali tieto časti:

- Autocomplete
  - Autocomplete ľudí
  - Autocomplete všetkých entít
- Filmy
  - Všetky filmy
  - Aktuálne filmy
  - Zobrazenie filmu



- Zobrazenie trailerov k filmom
- Žánre
  - Zobrazenie žánrov
  - Zobrazenie žánru
- Ľudia
  - Zobrazenie ľudí
  - Zobrazenie režisérov
  - Zobrazenie hercov / herečiek
  - Zobrazenie producentov
  - Zobrazenie osoby
- Požívatelia
  - Prihlásenie
- Odporúčanie pre používateľa
  - Odporúčanie filmov - úplné
- Odporúčanie
  - Odporúčanie filmov - úplné
  - Odporúčanie Kín o Odporúčanie TV Programu

### 12.2.3 Návrh

Kompletná dokumentácia k API je v prílohe API PRÍLOHA.

### 12.2.4 Implementácia

Funkcionalitu implementujeme v controllers, tak že rozšírime konkrétne actions o format.json kde budeme vracat konkrétne entity v JSON formáte.

## 12.3 Môj vlastný TV - program

### 12.3.1 Úloha

Vytvorte grafické používateľské rozhranie pre projekt TeleVido – modul môj vlastný TV program TeleVido. Rozhranie musí spĺňať všetky štandardy modernej webovej stránky a musí zahŕňať všetky základné položky pre korektnú prevádzku uvedeného modulu aplikácie.

The screenshot displays a REST client interface with the following components:

- Environment:** Normal, Basic Auth, OAuth 1.0, No environment.
- URL:** localhost:3000/recommendations/movies\_based\_on\_any
- Headers:** Accept: application/json, Content-Type: application/json. A Manage presets button is visible.
- Body:** form-data, x-www-form-urlencoded, raw, JSON. The raw JSON body is:

```
1 {
2   "method": "union_colors",
3   "nodes": [
4     { "type": "movie", "id": 1 },
5     { "type": "person", "id": 1 },
6     { "type": "genre", "id": 1 }
7   ],
8   "count": 10,
9   "fields": ["id", "title_orig"]
10 }
11 }
```
- Buttons:** Send, Add to collection.
- Response:** Body, Cookies (24), Headers (10), STATUS 200 OK, TIME 155 ms. The response is shown in Pretty format:

```
1 [
2   {
3     "id": 1,
4     "title_orig": "...First Do No Harm"
5   },
6   {
7     "id": 2,
8     "title_orig": "Airplane!"
9   },
10  {
11    "id": 3,
12    "title_orig": "Big Business"
13  },
14  {
15    "id": 4,
16    "title_orig": "Hot Shots!"
17  },
18  {
19    "id": 5,
20    "title_orig": "Hot Shots! Part Deux"
21  },
22  {
23    "id": 6,
24    "title_orig": "Jane Austen's Mafia!"
25  },
26  {
27    "id": 7,
28    "title_orig": "Ruthless People"
29  }
30 ]
```

Obr. 35: Ukážka requestu z API

### 12.3.2 Analýza

Pred konštruovaním návrhu grafického používateľského rozhrania sme si dôkladne zanalyzovali grafické rozhrania podobne zameraných aplikácií. Poukázali sme na slabé stránky a takisto aj na silné stránky analyzovaných riešení.

### 12.3.3 Návrh

Grafické používateľské rozhranie pre môj vlastný TV program je zobrazené na obrázku č. 36.

V tejto časti si opíšeme jednotlivé požiadavky na navrhovaný modul:

- zobrazenie TV programov prostredníctvom horizontálnej časovej osi
- horizontálny posúvnik sa bude nachádzať v hornej aj spodnej časti stránky
- horizontálne posúvanie bude fungovať aj prostredníctvom ťaženia myši
- TV program obsahuje informácie o názve programu s intervalom vysielania
- zobrazenie oficiálnych log televízií
- oranžové programy znázorňujú odporúčané programy
- v hornej časti sa nachádzajú jednotlivé dni
- zvýrazniť aktuálne vysielaný TV program prostredníctvom vertikálnej časovej osi

### 12.3.4 Implementácia

Grafické používateľské rozhranie bolo implementované v HTML5 a CSS3. Pre vykreslenie rôznych grafických animácií sme využili JavaScript a jQuery. Opisovaný modul aplikácie TeleVido je v súčasnosti k dispozícii na webovej adrese: <http://www.televido.tv/tv-program>.

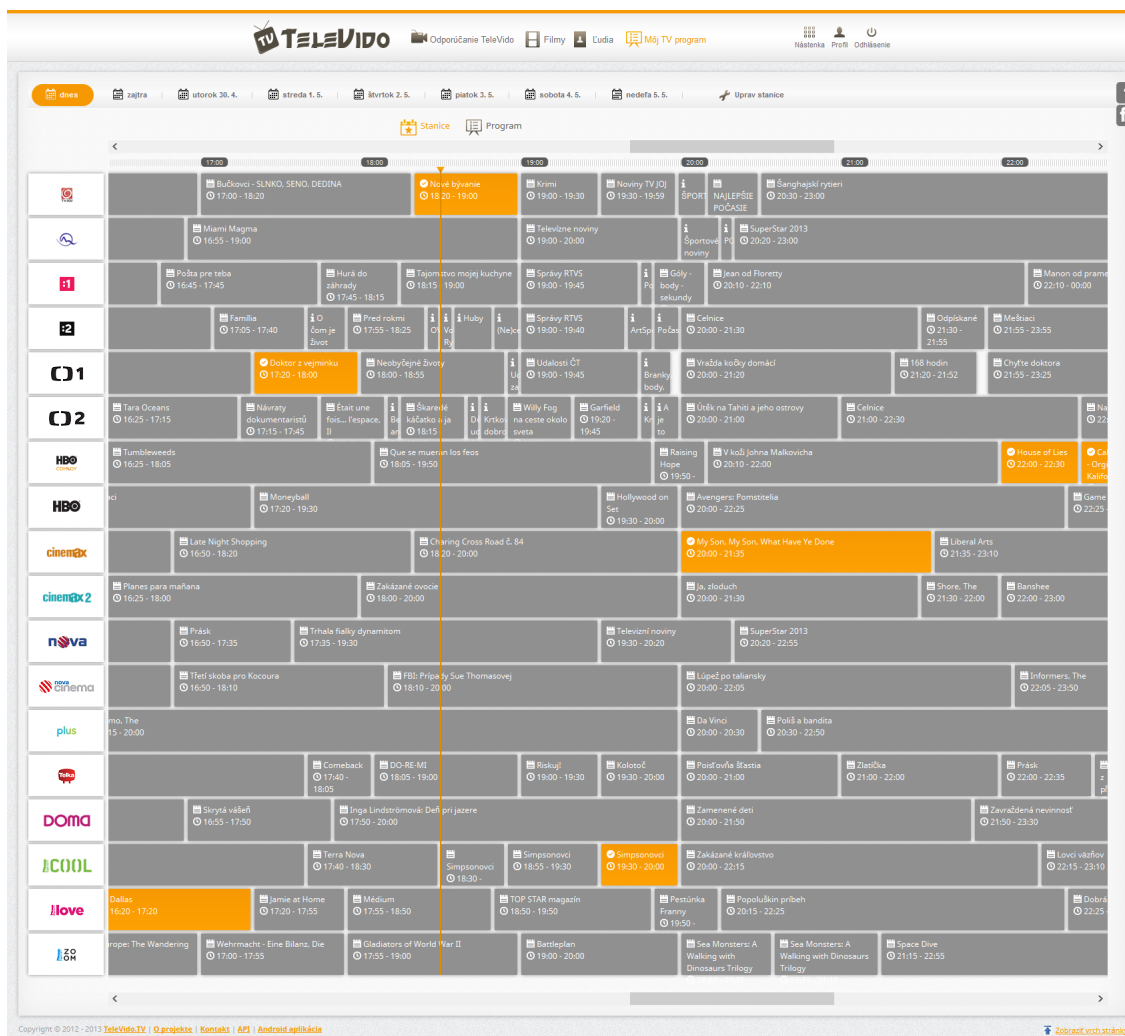
## 12.4 Navigácia a nastavenia v Android aplikácii

### 12.4.1 Úloha

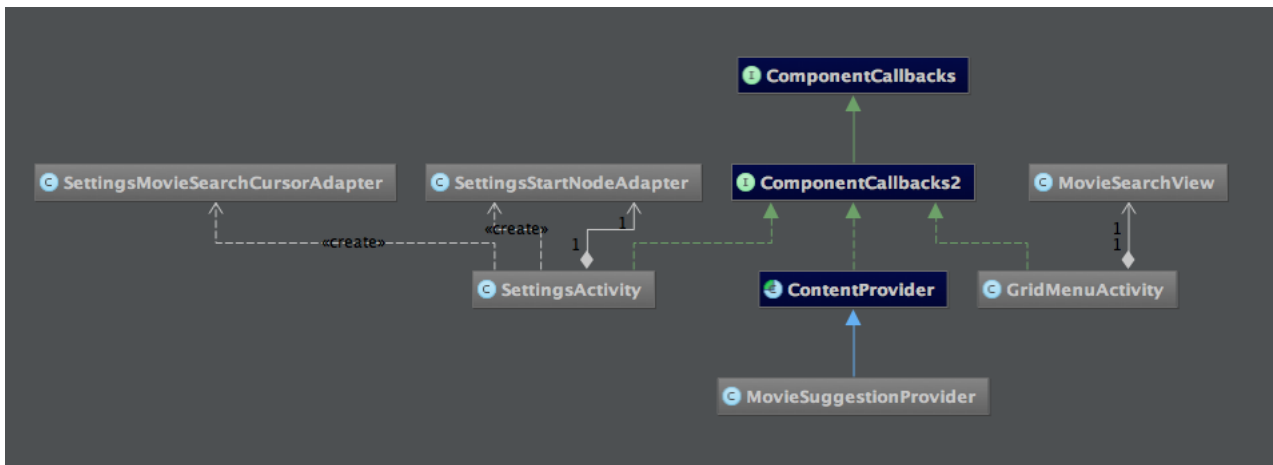
Implementuje a nasadíte funkcionality v android aplikácii na základa čoho sa konfigurovať aplikáciu a proces odporúčania a umožniť navigáciu v aplikácii.

### 12.4.2 Analýza

Navigácia v aplikácii by mala byť intuitívna pre používateľa a taktiež nastavenia. Mala by umožniť dať možnosť používateľovy výber typ odporúčania či na zakláde explicitných uzlov alebo podľa používateľského profilu.



Obr. 36: Grafické používateľské rozhranie pre modul môj vlastný TV program



Obr. 37: Diagram tried pre vyhľadávanie entít

### 12.4.3 Návrh

Pre navigáciu sme sa rozhodli použiť dashboard v aplikácii v ktorej budú možné volať jednotlivé aktivity ktoré budú pristupovať rôzne featury aplikácií. Taktiež použijeme kontextové menu na sprístupnenie niektorých funkcionalít. Tiež implementujeme fulltextový vyhľadávač entít ktorá bude na dosah v každej aktivite.

### 12.4.4 Implementácia

Fulltextový vyhľadávač sme implementovali pomocou komponentu SearchView. Sme vytvorili vlastný komponent na vyhľadávanie entít ktorá dedí od SearchView a používa komponent ContentProvider na volanie služby. Výhoda použitia komponentu ContentProvider že akcie vykonané v tejto inštancii sa spúšťajú v vlastnom vlákne mimo vlákna ktorá sa stará grafické rozhranie. Ďalšou výhodou komponentu že je možné ho napojiť na CursorAdapter týmpadom je znova použiteľný aj pri obyčajných textových fieldoch. V nastavenia sme implementovali tak aby používateľ pomocou radiobuttonov mohol prepínať medzi explicitnou a implicitnou formou odporúčania. Pre implicitné odporúčanie je treba byť prihlásený , tento krok používateľ môže vykonať v nastaveniach. Pre explicitné odporúčanie používateľ musí zdefinovať šartovacie uzly , tento krok môže vykonať v nastaveniach kde je a implementované fulltextové vyhľadávanie entít pomocou toho istého komponentu ako pri navigácii. Všetky volané služby v aktivite sa vykonávajú vo vlastnom vlákne týmpadom neohrozujú hlavné vlákno ktorá sa stará o grafické rozhranie. Na obrázku 37 môžete vidieť diagram tried pre vyhľadávanie entít použitý v hlavnom menu a v nastaveniach.

## 12.5 Neo4j prídavný modul - Algoritmus šírenia energie, efektívnejšia verzia

### 12.5.1 Úloha

Vytvorte prídavný modul do Neo4j databázy pre získanie najbližších uzlov ku zadaným uzlom. Algoritmus bude na vstupe očakávať začiatkové uzly, ale taktiež môže dostať aj iné parametre, aby bol kompatibilný s ostatnými vytvorenými prídavnými modulmi. Ako základ použite algoritmus šírenia energie.

### 12.5.2 Analýza

Odporúčaným spôsobom vytvorenia prídavného modulu do Neo4j databázy je vytvorenie JAR súboru. Vytvorená trieda musí dediť od triedy `ServerPlugin`. Vytvorený JAR súbor musí v priečinku `META-INF/services` mať vytvorený súbor `org.neo4j.server.plugins.ServerPlugin`, v ktorom sú zaznamenané cesty ku vytvoreným triedam. Teda pre triedu `HelloWorld` v balíku `org.neo4j.examples.server.plugins` je potrebné aby bolo v súbore zapísané `org.neo4j.examples`. Už vytvorený algoritmus šírenia energie dáva najlepšie odporúčania spomedzi všetkých algoritmov (prídavných modulov). Problémom však je, že nie je časovo efektívny. Táto časová náročnosť sa prejavuje najmä pri zadaní viacerých vstupných vrcholov. Pri počte 20 vrcholov nedokáže algoritmus vôbec zbehnúť (počas priebehu je ukončený systémom). Problém je najmä v tom, že každý vstupný vrchol má samostatnú frontu - je spúšťaných niekoľko samostatných BFS a vrchol môže byť tým pádom navštívený viac krát (pri jednom BFS len raz).

### 12.5.3 Návrh

Algoritmus bude založený na rovnakom princípe ako predchádzajúci algoritmus šírenia energie. Zmena bude v tom, že algoritmus bude pracovať len s jednou frontou, a teda každý vrchol bude môcť byť navštívený len raz. Spôsob práce s frontou bude rovnaký ako v prípade algoritmu miešania farieb. To znamená, že ak sa vrchol uloží do fronty s hodnotou energie „X“ a počas jeho čakania vo fronte sa jeho energia zmení na „Y“, tak ďalej pošle energiu na základe hodnoty „Y“. Toto riešenie algoritmus výrazne zlepší po časovej stránke, ale o niečo zhorší po stránke odporúčania. To z dôvodu, že ak vrchol rozpošle energiu (už bol vo fronte) a neskôr k nemu dorazí nová energia (zo vzdialenejšieho vrcholu), tak ju už nerozpošle, pretože už bol raz vo fronte (môže tam byť len raz). Pre zlepšenie odporúčania bude taktiež pridaný parameter „nodes\_energy“, s ktorého pomocou bude možné zadať energie začiatkových vrcholov a tým určiť ich silu pre odporúčanie.

### 12.5.4 Implementácia

Ako bolo spomenuté v analýze, tak algoritmus bude vytvorený v programovacom jazyku Java.

## 12.6 Ďalšie návrhy pre používateľské rozhranie Android aplikácie a ich implementácia do XML

### 12.6.1 Úloha

Navrhňte ďalšie obrazovky pre používateľské rozhranie mobilnej aplikácie pre android a implementujte ich v XML.

### 12.6.2 Analýza

AnalYZovali sme podobné riešenia v aplikáciách také, aké sme potrebovali použiť v našej aplikácii. Pre každý element v návrhu sme zvolili prislúchajúci XML element v aplikácii. Použili sme niekoľko základných XML elementov, ktoré sa používajú pri vývoji android aplikácií:

- `LinearLayout`
- `GridView`
- `TextView`
- `ImageView`

### 12.6.3 Návrh

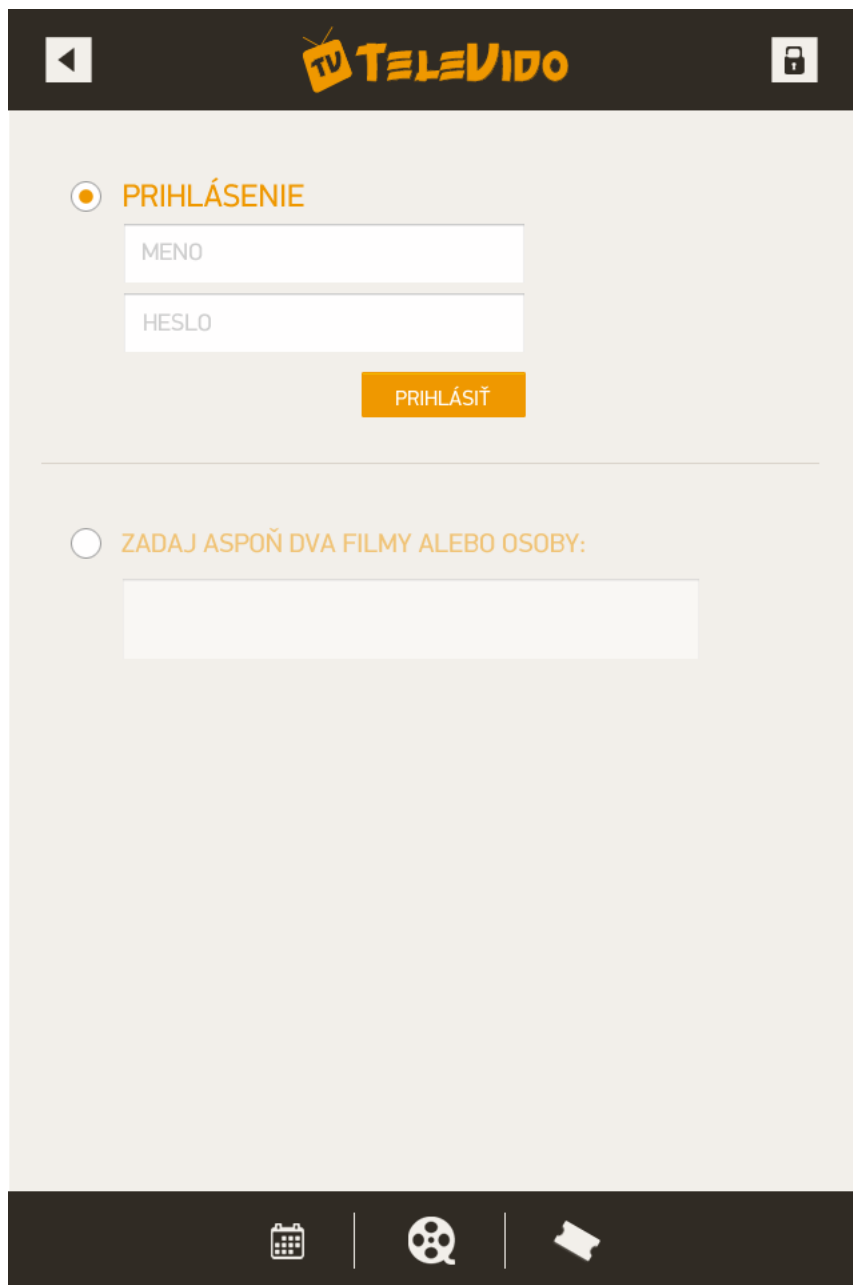
Navrhli sme ďalšie obrazovky používateľského rozhrania:

- Obrazovka nastavení
- Obrazovka dashboardu (úvodnej obrazovky)
- Obrazovka odporúčania filmov v kinách

Obrazovka (obrázok 38.) slúži na nastavenie používateľského konta, alebo ak nie je používateľ prihlásený na to, aby zadal filmy, ktoré má rád a naša aplikácia mu vedela odporučiť filmy. Obrazovka dashboardu (obrázok 39) slúži na orientáciu v aplikácii po jej zapnutí. Počas analýzy sme identifikovali a navrhli 4 základné položky menu:

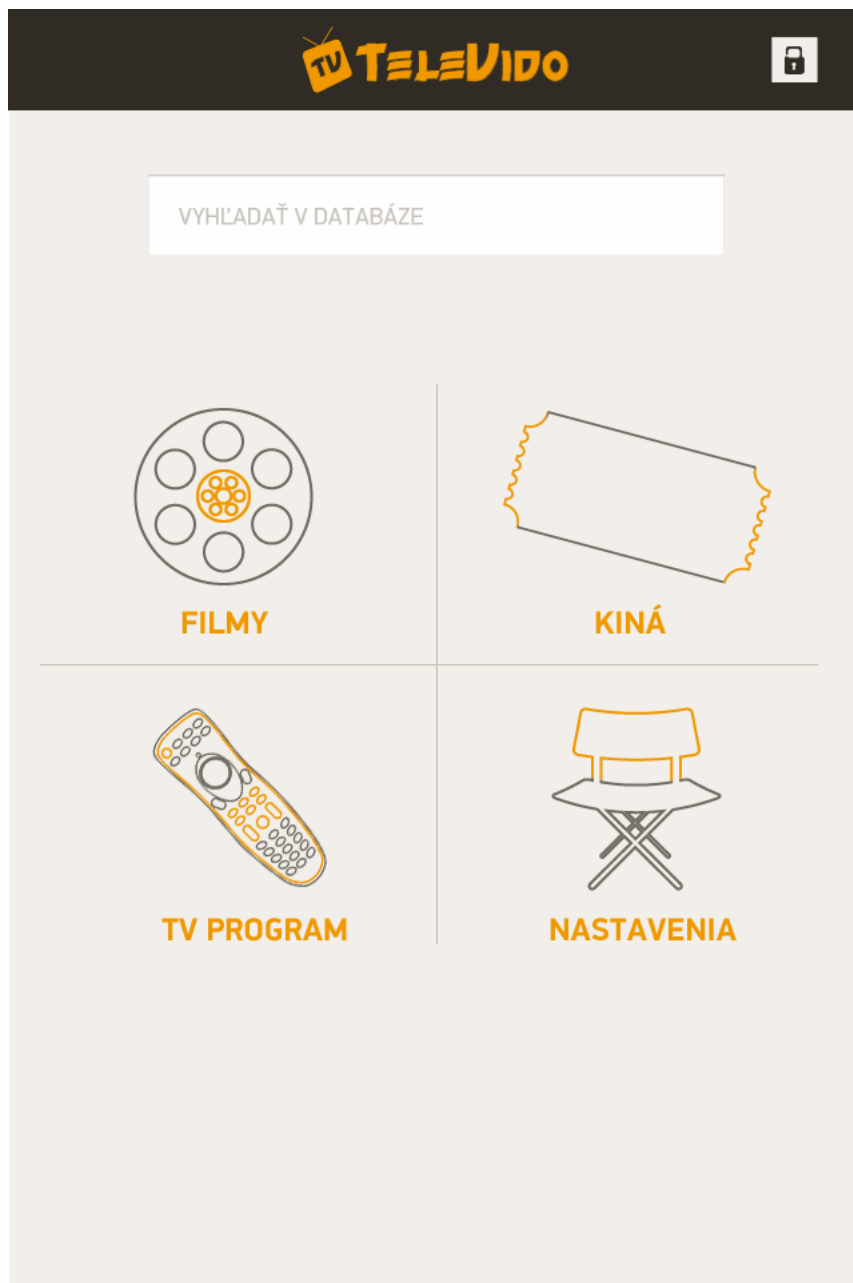
- **Filmy** – odporúčanie filmov používateľovi
- **TV-program** – odporúčanie TV-programu používateľovi
- **Kiná** – odporúčanie filmov v kinách na základe používateľovej polohy
- **Nastavenia** – používateľské nastavenia

V obrazovke odporúčania filmov (obrázok 40) v kinách sme museli zohľadniť geolokáciu používateľa, preto sme použili posuvník, ktorý identifikuje do akej vzdialenosti má naša aplikácia hľadať kiná a filmy v nich.



Obr. 38: Obrazovka nastavení





Obr. 39: Obrazovka nastavení

◀
TELEVIDEO
🔒

VYBER AKO ĎALEKO MÁ KINO BYŤ

5 

 100

KINO DO VZDIALENOSTI: **35 KM**

---

**ODPORÚČANÉ FILMY V KINÁCH**

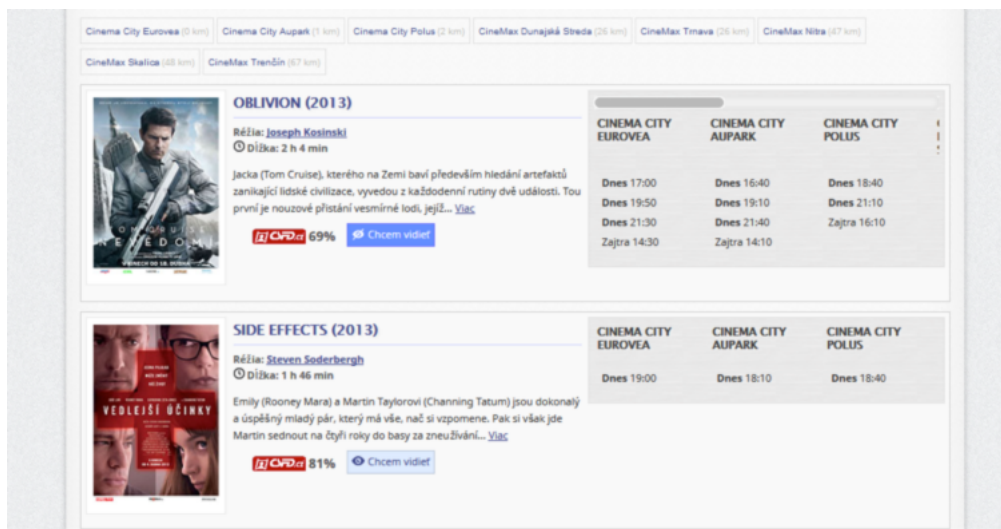
**THE HOBBIT: AN UNEXPECTED JOURNEY (2012)**
KINO: CINEMAX TRNAVA (10 KM)  
ŽÁNER: KOMÉDIA, AKČNÝ

|                      |              |
|----------------------|--------------|
| CINEMAX AUPARK (4KM) | DNES 18:30   |
| KINO HVIEZDA (3KM)   | DNES 18:45   |
| CINEMAX AUPARK (4KM) | DNES 21:00   |
| CINEMAX POLUS (15KM) | ZAJTRA 21:00 |

**FAST FIVE (2012)**
KINO: MERKÚR TRNAVA (17 KM)  
ŽÁNER: KOMÉDIA, AKČNÝ

📅
🎬
👉

Obr. 40: Obrazovka odporúčania filmov v kinách



Obr. 41: Zobrazenie filmov v kinách

## 12.6.4 Implementácia

Implementácia prebiehala v jazyku XML a v elementoch, ktoré používa android SDK. Toto XML sa následne zobrazuje v GridMenuActivity.java a jednotlivé prvky sa ďalej naplňajú v adaptéri MenuImageAdapter.java.

## 12.7 Zobrazenie kín pri odporúčaní

### 12.7.1 Úloha

Vytvorte grafické používateľské rozhranie pre odporúčané filmy tak, aby sa pri každom z nich nachádzali informácie o tom, kedy sa film premieta v kine

### 12.7.2 Analýza

K pôvodným informáciám o filme (titulný obrázok, názov filmu, režia, obsah, hodnotenie a pod) je potrebné pridať názov kín, v ktorých film premietajú a jednotlivé časy. Vzhľad stránky by pritom mal zostať jednotný a informácie musia byť zobrazené v prehľadnej forme. Nesmieme zabudnúť na tlačidlo Chcem vidieť.

### 12.7.3 Návrh

Keďže má zostať jednotný vzhľad, za základ sme si zvolili zoznam filmov z odporúčania filmov. K týmto informáciám bolo potrebné pridať časy premietania v jednotlivých kinách. Počet kín v zobrazovaní sa mení, ako základné zobrazenie sme zvolili tri kiná vedľa seba. Pri väčšom počte kín bolo potrebné zobrazenie sa medzi nimi posúvať pomocou lišty. Tieto kiná sa zobrazujú naľavo od informácií o filme (Obr. 41). Pričom sa zobrazujú iba časy premietania dnes a zajtra.

#### 12.7.4 Implementácia

Pre implementáciu boli použité technológie HTML a CSS, pre vytvorenie posúvacej lišty sme použili JavaScript a jQuery. Pre dynamické prvky bol použitý jazyk Ruby.

## 13 11. Šprint

### 13.1 Entity pre používateľa

#### 13.1.1 Úloha

Implementujte a nasadte vytváranie entít pre každého používateľa na odporúčanie.

#### 13.1.2 Analýza

Identifikovali sme tieto entity, ktoré pre každého používateľa budeme extrahovať:

- maximálne 30 filmov, ktoré ohodnotil najväčšou známku podľa toho kedy hodnotil (najnovšie prvé)
- v prípade, že ich je menej ako 10 pridáme ešte 3 najnovšie, ktoré ohodnotil druhou najlepšou známku
- maximálne 30 ľudí, ktoré ohodnotil najväčšou známku podľa toho kedy hodnotil (najnovší prví)
- 2 žánre, ktoré sú najčastejšie medzi filmami, ktoré má rád (na ktoré kliká)

#### 13.1.3 Návrh

Takáto selekcia by bola pomalá, ak by sme ju robili pri každom volaní. Rozhodli sme sa preto opäť si ukladať získané entity. Pre každého používateľa si pamätáme zoznam entít, ktoré ma aktuálne rád. Tento zoznam sa aktualizuje každých 10 minút (tak ako sa aktualizuje záujem používateľa) a tiež po každom ohodnotenom filme či osobe.

#### 13.1.4 Implementácia

Implementovali sme to štandardne v Ruby On Rails vytvorením nového modelu, ktorý reprezentuje tabuľku v databáze. Pri zavolaní funkcie `entities_for_recommendation` zoberie entity z databázy a nemusí ich vždy vyhľadávať. Pomocou gemu `delayed jobs` a `cron` sa pravidelne aktualizuje tento zoznam. V prípade, že sa niečo pridá do hodnotení (`user_feedback`) tak sa po uložení (`after_save`) tento zoznam opäť aktualizuje.

## 13.2 Obrázky pre Android aplikáciu

### 13.2.1 Úloha

Implementujte a nasadte funkcionalitu generátora obrázkov pre zadaný film. Obrázky, ktoré sú k filmom dostupné z ČSFD sú v malom formáte a preto je potrebné urobiť plugin, ktorý by vytváral obrázky pre film vo väčšom formáte, spojením aj obrázkov ľudí, ktorý v tom filme hrali alebo inak zasiahli.

### 13.2.2 Analýza

Keďže pre zadaný film máme k dispozícii malý obrázok je potrebné urobiť väčšie, ktoré by sme mohli využiť pri zobrazovaní detailu filmu na mobilnej aplikácii. Pre film existuje teda obrázok a máme aj zoznam ľudí, ktorý sa na filme nejakou mierou podieľali (režiséri, herci, scenáristi). Je potrebné aby sme našli spôsob ako z toľkých ľudí vybrať tých zaujímavých, ktorý ten obrázok obohatia. ČSFD si ukladá ľudí podľa obľúbenosti a teda vo väčšine platí, že čím má osoba nižšie ČSFD\_ID tak je zaujímavšia. Pre potreby vytvárania nášho obrázku sme zvolili rozmer, pre 6 malých obrázkov - film a 5 osôb. Preto vyberieme 5 ľudí s najnižším ČSFD\_ID a tie obrázky spojíme. Táto funkcionalita bude dostupná aj pomocou API `<url>/movies/{movie_id}/image`

### 13.2.3 Návrh

Túto funkcionalitu by som rozdelil na dve časti:

**PHP image plugin**, ktorý bude dostupný z vonku. Pluginu sa pošle zoznam obrázkov v url formáte, ktoré sa majú spojiť. PHP plugin bude dostupný cez GET request. Napr. `<url>/televido_image_plugin/?image_urls[0]=http://www.movieposterdb.com/posters/11_08/2002/`

Druhá časť bude už priamo implementovaná v našej aplikácii. Bude to wrapper nad týmto pluginom ako samostatná knižnica, ktorá bude volať PHP wrapper.

### 13.2.4 Implementácia

PHP plugin sme implementovali pomocou základných funkcií na prácu s obrázkom.

Pri implementácii do našej služby sme dorobili cachovanie obrázkov, ktoré keď sa raz pomocou PHP pluginu vytvorili, tak sme si ich uložili a pri ďalšom zavolaní sa to zobrazilo z lokálneho súboru.

## 13.3 Poster na konferenciu IIT SRC

### 13.3.1 Úloha

Vytvorte grafický návrh posteru na študentskú vedeckú konferenciu IIT.STC pre projekt TeleVido. Poster musí spĺňať všetky štandardy moderného grafického návrhu a musí zahŕňať všetky základné položky pre korektnú reprezentáciu projektu na konferencii.

### 13.3.2 Analýza

Pred konštruovaním grafického návrhu posteru sme si dôkladne zanalyzovali grafické návrhy iných posterov z minulých ročníkov konferencie. Poukázali sme na slabé stránky a takisto aj na silné stránky analyzovaných riešení.

### 13.3.3 Návrh

Grafický návrh posteru na konferenciu je zobrazený na obrázku č. 42.

V tejto časti si opíšeme jednotlivé požiadavky na navrhovaný grafický návrh:

- výstižný nadpis projektu TeleVido s uvedením výrazného loga
- uvedenie autorov projektu s uvedením mentora projektu
- zobrazenie tematického obrázku projektu v podobne staromódnej televízie
- uvedenie cieľov projektu
- uvedenie informácií o grafovej databáze
- uvedenie prehľadu architektúry projektu
- uvedenie informácií o grafových algoritmoch
- uvedenie prínosov projektu TeleVido
- na pozadí posteru budú zobrazené prepojenia známych filmov s populárnymi hercami a režisérmi
- prostredníctvom noteboku a smartphonu znázorníme ukážku webovej a mobilnej aplikácie
- v spodnej časti sa budú nachádzať informácie o škole, odkaz na demonštráciu projektu s QR kódom

### 13.3.4 Implementácia

Grafický návrh posteru bol implementovaný prostredníctvom softvéru Microsoft Visio a výstup bol regenerovaný do formátu PDF, ktorý bol následne doručený príslušnej tlačiarni.

## 13.4 Odporúčanie filmov Android aplikácií

### 13.4.1 Úloha

Implementuje a nasadíte funkcionality v android aplikácií na základa čoho bude možné odporúčať filmy a relácie pre explicitnou a implicitnou metódou.

# TV TELEVIDEO

## Graph-based Movie Recommender



Supervisor: Dušan Zelenik

Kubo Križ  
Luboš Demovič  
Ondrej Proksa  
Ondrej Kuzmik  
Di Vandlíková  
Edo Fritscher  
Jakub Šalmík

### Goals

- movie recommender
- TV schedule recommender
- cinema recommender
- information about movies / people

### Graph Database

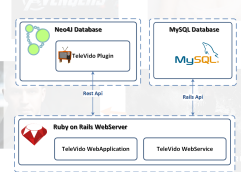
- Neo4j storage with more than 1 million nodes
- stored as triplets (object x predicate x subject)
- for example: Uma Turman stars Pulp Fiction



### Recommending Movies

- adapts to user model or selected movies or persons
- implicitly uses metadata of movies and similarities

### Architecture Overview



### Graph Algorithms

- simulating similarity among entities in graph
- experiment: comparing qualities for 4 algorithms (Union Colors, Mixing Colors, Energy, Dijkstra)

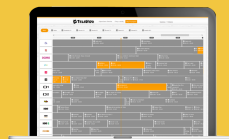


### Contributions

- efficient on-the-fly recommendation
- recommending using user model
- collaborative recommendation
- open for experiments with other algorithms
- API for everyone



TeleVideo Recommender



TeleVideo My TV schedule



TeleVideo Android App

STU  
FIIT  
SLOVENSKÁ TECHNICKÁ  
UNIVERZITA V BRATISLAVE  
FAKULTA INFORMATIKY  
INFORMATIKA A INFORMAČNÉ TECHNOLOGIE

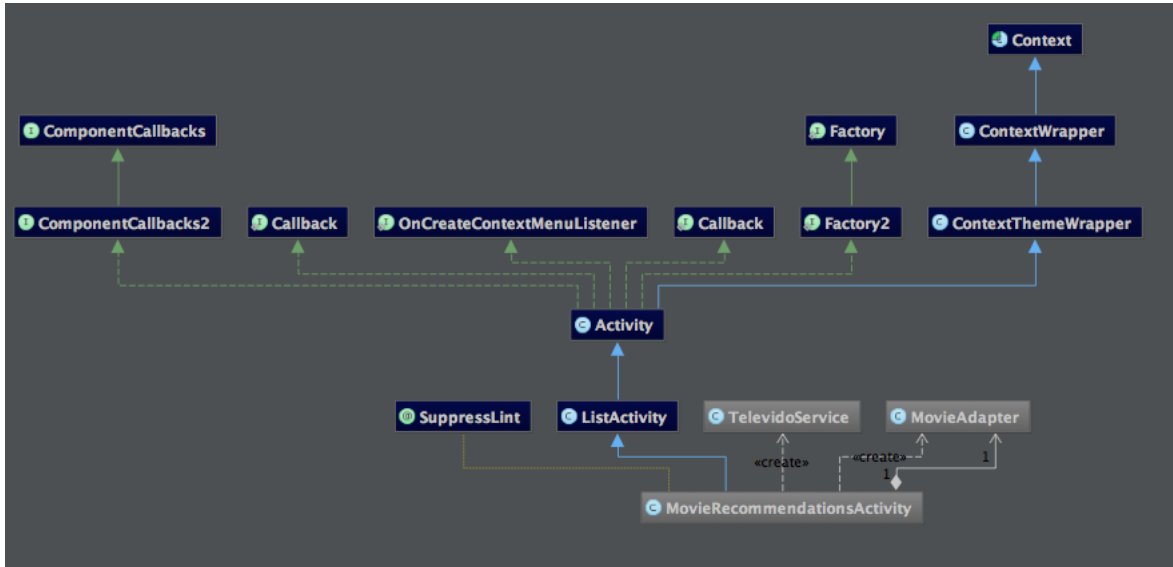


www.televido.tv

TeleVideo is smart TV to help you choose the best program for watching

ITIC SRC  
Informatics and Information Technologies  
Student Research Conference

Obr. 42: Poster



Obr. 43: Diagram tried ktorý reprezentuje funkcionality pre odporúčanie filmov android aplikácií

### 13.4.2 Analýza

Zobrazenie pre používateľa by mala byť intuitívna. Maly by sa zobrazíť niektoré atribúty filmov a potom aby používatelia mali možnosť na zobrazenie filmu alebo relácie.

### 13.4.3 Návrh

Pre zobrazenie filmov a relácií sme rozhodli použiť záznam s vlastným grafickým rozhraním kde bude poster filmu a informácie oňom.

### 13.4.4 Implementácia

Aktivita ktorá bude zobrazovať odporúčanie filmov a relácií bude volať služby servisnej vrstvy vo vlastnom vlákne. Takto sa zabezpečí beh hlavného vlákna ktorá sa stará a grafické rozhranie aplikácií. Zobrazenie jednotlivých záznamov bude uskutočnená pomocou komponentu ListActivity. Pomocou ArrayListAdapterov sa spraví zobrazenie jednotlivých záznamov v liste.

Na obrázku 43 je diagram tried ktorý reprezentuje funkcionality pre odporúčanie filmov android aplikácií.

## 13.5 Bookmarky, blacklisty

### 13.5.1 Úloha

Navrhnete a implementujte funkcionality používania zoznamu záložiek (bookmarks) a zoznamu vyradených filmov (blacklist) pre používateľa. Záložka znamená, že používateľ má záujem si daný film pozrieť, vyradený film znamená, že ho používateľovi nezobrazíme v odporúčaníach.



### 13.5.2 Analýza

Používateľ musí mať jednoducho možnosť pridať filmy do zoznamu záložiek alebo vyradených filmov. Funkcia pridania a odoberania filmov do záložiek bude dostupná pri zobrazení každého filmu vo forme gombíka, ktorý má dvojakú funkciu pridania a odoberania. Používateľ bude mať takisto možnosť zobrazíť si všetky filmy v záložkách a možnosť vyradiť ich zo záložiek priamo odtiaľ. Funkcia pridania filmov do zoznamu vyradených je dostupná pri zobrazení filmu v odporúčaní. Po kliknutí na gombík vyradenia okamžite zmizne zo zoznamu a v budúcnosti sa nebude zobrazovať.

### 13.5.3 Návrh

Pre vytvorenie oboch zoznamov vytvoríme dve tabuľky v databáze a im prislúchajúce modely v Rails aplikácii. Tieto tabuľky predstavujú m:n relácie filmov a používateľov. Zvyšok funkcionality implementujeme do Rails aplikácie štandardným spôsobom. Pre vyradenie filmov zo zoznamu vyradených z odporúčania použijeme volanie odporúčacieho pluginu, ktorému môžeme zadať rôzne parametre podľa návrhu v predchádzajúcich šprintoch. V tomto prípade využijeme parameter `result_nodes_exclude`.

### 13.5.4 Implementácia

Zadanie bolo implementované v existujúcej aplikácii Ruby on Rails. Pre okamžité vyradenie filmu z odporúčaní alebo pre zmeny tlačidiel pri výbere filmu bola použitá JavaScriptová knižnica JQuery. Odporúčanie je implementované v Jave nad databázou Neo4j, pre potreby zadania sme tento odporúčací plugin volali s nastavením potrebných parametrov.

## 13.6 Parser kín - nová verzia

### 13.6.1 Úloha

Vytvorte metódu pre vyparovanie programu najväčších slovenských kín. Parser musí zahŕňať dve najväčšie siete slovenských kín a to konkrétne Cinema City a CineMax.

### 13.6.2 Analýza

Metóda už bola vytvorená, ale vzhľadom na to, že sa zmenila webová stránka CineMax, ju bolo treba pozmeniť. Pri vytváraní novej metódy sa objavili rôzne problémy s parsovaním a po ich zvážení sa pristúpilo k novému riešeniu. Nové riešenie spočívalo v nájdení externého zdroja (iného ako stránka CineMax) a jeho následnom vyparovaní. Ako externý zdroj sa vybrala stránka <http://programy.pravda.sk/kino.aspx>, pretože obsahovala všetky potrebné kiná (v prípade rozšírenia záujmu aj mnoho ďalších) a program kín bol zobrazený jednoducho a priamočiaro pre následne parsovanie. Pre konzistenciu parseru bolo rozhodnuté, že aj program Cinema City bude parsovaný zo stránky <http://programy.pravda.sk/kino.aspx>.

### 13.6.3 Návrh

Príklad URL:

<http://programy.pravda.sk/kino.aspx?kina=50003110&datum=30.4.2013>

- kina: označenie kina
- datum: dátum premietania vo formáte deň.mesiac.rok

CSS rozdelenie:

- získanie programu: `.timeProg`
  - získanie konkrétneho filmu (v konkrétny čas): `div`
    - \* čas premietania filmu: `.time`
    - \* názov filmu (originálny): otvorenie stránky s detailami filmu a následne, `#ctl00_ContentPlaceH`  
`div:nth-child(3)`, alebo `.title` na originálnej stránke (použité pokiaľ stránka s detailami filmu nie je dostupná)

### 13.6.4 Implementácia

Parser bol naimplementovaný v jazyku Ruby v pracovnom rámci Ruby on Rails. Pre parsovanie boli využité gemy Nokogiri a Mechanize.

## 14 12. Šprint

### 14.1 Informácie o projekte

#### 14.1.1 Úloha

Vytvorte grafické používateľské rozhranie pre projekt TeleVido – modul informácie o projekte TeleVido. Rozhranie musí spĺňať všetky štandardy modernej webovej stránky a musí zahŕňať všetky základné položky pre korektnú prevádzku uvedeného modulu aplikácie.

#### 14.1.2 Analýza

Pred konštruovaním návrhu grafického používateľského rozhrania sme si dôkladne zanalyzovali grafické rozhrania podobne zameraných aplikácií. Poukázali sme na slabé stránky a takisto aj na silné stránky analyzovaných riešení.

#### 14.1.3 Návrh

Grafické používateľské rozhranie pre informácie o projekte je zobrazené na obrázku č. 44.

V tejto časti si opíšeme jednotlivé požiadavky na navrhovaný modul:

- stručné a výrazné uvedenie základnej myšlienky Televido v dvoch vetách
- rýchly kontakt na autorov projektu s uvedením školy, na ktorej sa projekt vypracováva

- uvedenie motivácie prečo robíme TeleVido
- vysvetlenie ako funguje TeleVido zrozumiteľnou podobou
- obohatiť obsah peknými obrázkami pre zvýšenie atraktivity projektu

#### 14.1.4 Implementácia

Grafické používateľské rozhranie bolo implementované v HTML5 a CSS3. Opisovaný modul aplikácie TeleVido je v súčasnosti k dispozícii na webovej adrese: <http://www.televido.tv/o-projekte>.

## 14.2 Odporúčanie kín v android aplikácii

### 14.2.1 Úloha

Implementuje a nasadíte funkcionality v android aplikácii na základa čoho bude možné odporúčať filmy v kinách explicitnou a implicitnou metódou na základe pozícií používateľa.

### 14.2.2 Analýza

Zobrazenie pre používateľa by mala byť intuitívna. Mali by sa zobrazovať niektoré atribúty filmov a potom aby používatelia mali možnosť na zobrazenie filmov. Ďalej odporúčanie by malo aj zobrazovať aj čas premitania. Fitúra by mala umožniť používateľovi si zvoliť aj vzdialenosť kín.

### 14.2.3 Návrh

Pre zobrazenie filmov a relácií sme rozhodli použiť záznam s vlastným grafickým rozhraním kde bude poster filmu a informácie oňom. Na zobrazenie vzdialenosť kín sme rozhodli používať slider.

### 14.2.4 Implementácia

Aktivita ktorá bude zobrazovať odporúčanie filmov bude volať služby servisnej vrstvy vo vlastnom vlákne. Takto sa zabezpečí beh hlavného vlákna ktorá sa stará a grafické rozhranie aplikácií. Zobrazenie jednotlivých záznamov bude uskutočnená pomocou komponentu ListActivity. Pomocou ArrayAdapterov sa spraví zobrazenie jednotlivých záznamov v liste. Informácie o premietania bude tiež zrealizovaná pomocou ArrayAdapter a to v nútri prvéh adaptera. Adapter bude ovládaný s sliderovom. Lokácia používateľa bude získaná pomocou POJO triedy ktorá používa android geolokáciu. POJO trieda do konšuktora dostane kontext aktivity na základe čoho bude určovať geolokačnú polohu.

Na obrázku 45 je diagram tried ktorý reprezentuje funkcionality pre odporúčanie filmov v kinách android aplikácií.

## ČO JE TELEVIDO?



**TeleVido** je inteligentná televízia, ktorá Vám pomôže vybrať si ten najlepší program na sledovanie. Aplikácia Vám odporučí aktuálny program v TV, ako aj vaše žiadané seriály a filmy. TeleVido je skvelý spoločník pre každého diváka.

|        |                                                                          |
|--------|--------------------------------------------------------------------------|
| Škola  | <a href="#">FIIT STUBA</a>                                               |
| E-mail | <a href="mailto:televido@googlegroups.com">televido@googlegroups.com</a> |
| Web    | <a href="http://www.televido.tv">www.televido.tv</a>                     |

## PREČO ROBIEME TELEVIDO?

V súčasnosti je televízia jedným z najhlavnejších masmédií na svete. Zasahuje do života mnohých ľudí a najmä do životov detí a mládeže. Ľudia dokážu pri televízii stráviť nespočetné množstvo času a často ich môže pozitívne alebo negatívne ovplyvňovať. Používatelia si pod vplyvom veľkého počtu programov - filmov, seriálov a relácií, veľakrát nevedia vybrať adekvátnu možnosť. Ďalším problémom v dnešnom svete je nedostatok informácií o vybraných programoch. Cieľom projektu je priniesť do multimediálneho sveta platformu, ktorá by používateľom odporúčala vyhovujúci obsah. Projekt pomôže ľuďom, aby sa snažili tráviť svoj čas zmysluplne a pritom sledovali všetky programy, ktoré sú pre nich podstatné a dôležité.

Vytvorenie platformy pokrývajúcej potreby odporúčania multimediálneho obsahu s využitím znalostí, kreativity a nadšenia v spojení s najmodernejšími informačnými technológiami je pre nás veľkou výzvou. Sme presvedčení, že navrhnutý projekt má obrovský potenciál pre dnešnú spoločnosť a jeho realizácia by mnohým ľuďom zlepšila a zefektívnila spôsob života.



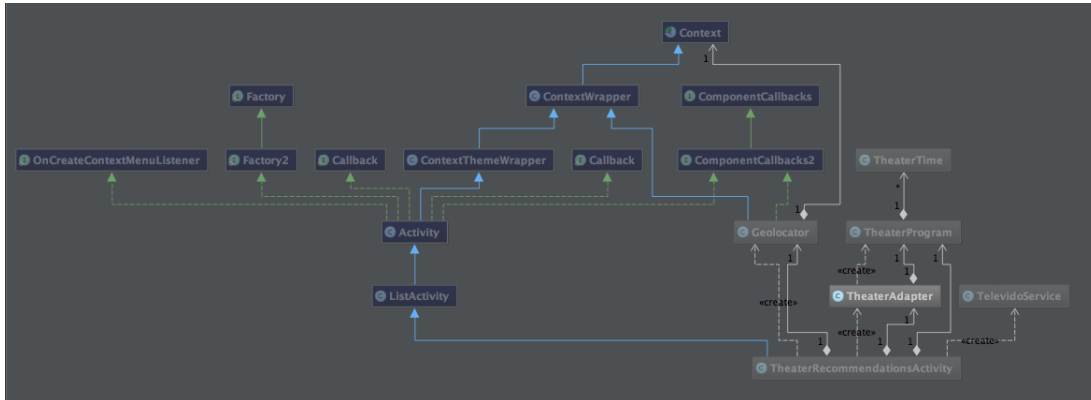
## AKO FUNGUJE TELEVIDO?



Naša inteligentná televízia vždy vie odporučiť správny obsah a vytvára pre diváka ten najlepší program. Každý si môže v TeleVido nájsť to, čo potrebuje. Najdôležitejšou súčasťou TeleVido je webová služba, ktorá na základe vytvoreného modelu používateľa odporúča vhodný obsah - filmy, seriály, relácie. Táto webová služba môže fungovať ako samostatný produkt, ktorý ponúkame ako podporný nástroj do multimediálnych zariadení.

Pre používateľov je tu webová aplikácia, ktorá sleduje aktivity a záujmy používateľa na internete. Na základe toho bude pre každého diváka vyberať to, čo by si mal v TV pozrieť. TeleVido ponúka aj mobilnú aplikáciu, ktorá bude sledovať záujmy používateľa a jeho aktivity, ktoré vykonáva s mobilným zariadením. Takto bude mať divák ten najlepší TV program vždy poruke.

Obr. 44: Grafické používateľské rozhranie pre modul informácie o projekte



Obr. 45: Diagram tried ktorý reprezentuje funkcionality pre odporúčanie filmov v kinách

## 14.3 Prerobenie odporúčania

### 14.3.1 Úloha

Zadávanie vstupu pre priame odporúčanie a zobrazenie výsledkov odporúčania je v existujúcej aplikácii nedostatočné. Súčasný stav pochádza ešte z prvého prototypu, je potrebné ho prerobiť, aby bol konzistentný so zvyškom webu. Takisto je potrebné refaktorovať kód backendu, ktorý je často neefektívny a duplicitný.

### 14.3.2 Analýza

Pridávanie prvkov do odporúčania je robené cez čistý JavaScript. Toto nie je optimálne kvôli starším prehliadačom, pri vypnutí JavaScriptu sa znefunkční celý návrh. Kvôli tomuto sa takisto pri úprave stránky nemení URL v prehliadači, čo je používateľsky neprístupné, pretože následne nie je možné používať tlačidlo späť ani kopírovať URL. Kód aplikácie je duplicitný a chaotický, pretože doňho boli časom pridávané rôzne nečakané funkcionality.

### 14.3.3 Návrh

Odporúčanie najprv prerobíme na čisto statickú verziu, ktorá používa klasické HTML formuláre. Následne túto verziu zlepšime použitím asynchrónneho načítavania, vďaka ktorému nebude potrebné znovu načítavať celú stránku. Pre odstránenie duplicity kódu navrhujeme helper metódy, ktoré obsahujú rovnakú funkcionality, aká sa nachádza na viacerých miestach a viewy, ktoré dokážeme mierne upravovať pomocou lokálnych premenných.

### 14.3.4 Implementácia

Návrh bol implementovaný v existujúcej aplikácii Ruby on Rails. Pre interaktivitu odporúčania sme použili AJAX implementovaný knižnicou JQuery. Pre interaktívnu zmenu URL sme použili HTML5 History API, ktoré nie je podporované vo všetkých prehliadačoch, ale zvyšok funkcionality sa zachová.

## 14.4 Výber TV staníc

### 14.4.1 Úloha

Implementujte a nasadíte výber TV staníc. Každý používateľ si môže vybrať stanice, ktoré chce mať pri TV Programe zobrazené a tak má možnosť filtrovať len ten program, ktorý chce.

### 14.4.2 Analýza

Pri analýze sme sa inšpirovali riešeniami iným konkurenčných portálov. Rozhodli sme sa pre špeciálne tlačidlo, ktoré bude zobrazené v hornej časti. Po kliknutí na tlačidlo bude možné vyberať stanice, ktoré chceme mať zobrazené a naopak aj stanice, ktoré nechceme mať zobrazené.

### 14.4.3 Návrh

Vyberač TV staníc sa zobrazí v novom okne, ktoré vyskočí na obrazovke. V okne budú dva stĺpce, ktoré reprezentujú stanice, ktoré chceme mať zobrazené a stanice, ktoré nechceme mať zobrazené. Medzi týmito dvoma zoznamami je možné presúvať názvy staníc.

### 14.4.4 Implementácia

Vyberač TV staníc sme implementovali pomocou RubyOnRails a jQuery pluginu qTip. Okrem toho sme vytvorili nový model UserPreferences, kde si pamätáme jednotlivé nastavenia pre používateľa. Potom čo si používateľ vyberie stanice a usporiada si ich podľa potreby, je po stlačení aktualizovaná stránka, kde sa zmeny prejavujú. Na pozadí sa po každej zmene zoznamov pošle požiadavka na aktualizovanie používateľských nastavení pre používateľa. V tabuľke user\_preferences si zoznam pamätáme ako JSON pole.

## 14.5 Zmena v nastavenia android aplikácie

### 14.5.1 Úloha

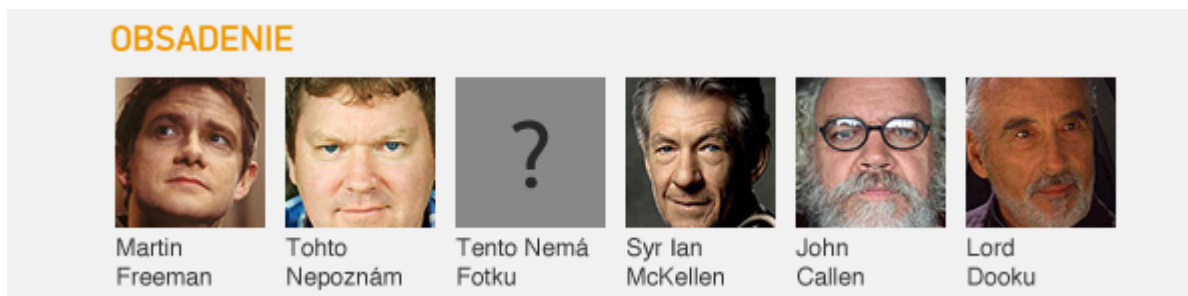
Upravte nastavenia tak, aby sa po kliknutí na výber prihlásenia alebo zadania filmov deaktivovali ostatné textové polia.

### 14.5.2 Analýza

Pri zadávaní prihlasovacích údajov pri nastaveniach alebo pri zadávaní filmov je potrebné aby používateľ nemohol zadávať údaje do oboch možností súčasne. Taktiež sa po kliknutí na textové pole musí tzv. pomocný text zmazať. Na vyriešenie oboch problémov použijeme dostupné android funkcie.

### 14.5.3 Návrh

Kliknutie na výber prihlásenia alebo zadávanie filmov budeme sledovať na pozadí aplikácie a pomocou funkcie `setEnabled()` nastavíme aktiváciu/deaktiváciu príslušných textových polí.



Obr. 46: Galéria hercov pri detaile filmu

Vymazávanie pomocného textu v textových poliach vykonávame pomocou funkcie `setHint()`, do ktorej vložíme prázdny reťazec.

#### 14.5.4 Implementácia

Implementácia prebehla v jazyku Java a v Android SDK. Zmenu sme vykonali v súbore aktivít nastavení `SettingsActivity.java` kde sme použili funkcie uvedené v návrhu.

### 14.6 Pridanie galérie hercov a filmov

#### 14.6.1 Úloha

Pridajte do detailu filmu galériu hercov, ktorý v ňom hrali a do detailu herca galériu filmov v ktorých hral.

#### 14.6.2 Analýza

Pri zobrazení filmu je potrebné aby používateľ vedel aký herci v ňom hrali a takisto pri zobrazení detailu herca je potrebné aby boli zobrazené filmy, v ktorých hral. Pre túto potrebu sme zvolili obrázkovú galériu s fotkami hercov a obrázkami filmov.

#### 14.6.3 Návrh

Galéria (obrázok 46.) sa bude zobrazovať pod popisov filmu a používateľ sa v nej bude orientovať pomocou gesta posuvania prstom po obrazovke. Zvolii sme android komponent `Gallery`, ktorý nám poskytuje príslušné funkcie a je plne manažovateľný.

#### 14.6.4 Implementácia

Implementácia prebehla v jazyku Java a v Android SDK. Zmenu sme vykonali v súbore aktivít filmov a hercov `MovieActivity.java` a `PersonActivity.java`, kde sme použili vybraný modul. Samotné napĺňanie obrázkov prebehlo v adaptéri `ActorsImageAdapter.java` a `MoviesImageAdapter.java`.



Obr. 47: Obrazovka výberu TV staníc

## 14.7 Výber TV staníc - prezenčná vrstva

### 14.7.1 Úloha

Vytvorte rozhranie, ktoré umožní používateľovi výber televíznych staníc, z ktorých sa bude zobrazovať televízny program.

### 14.7.2 Analýza

Pre výber z možností existuje viacero prístupov. Najlepšou možnosťou je vytvorenie tzv. vyskakovacieho okna, v ktorom sa zobrazia televízne stanice. Z týchto staníc si potom používateľ vyberie tie, ktoré chce zobrazovať v televíznom programe. Existujú dve základné možnosti a to buď označenie tých, ktoré chceme vidieť alebo presúvanie medzi možnosťami zobrazovať/nezobrazovať. Potom je ešte potrebné výber potvrdiť. Uprednostnili sme riešenie, ktoré bude čo najviac pohodlné pre používateľa.

### 14.7.3 Návrh

Úprava staníc sa zobrazí vo forme vyskakovacieho okna v strede obrazovky. Rozhodli sme sa pre možnosť zobrazenia dvoch stĺpcov, medzi ktorými bude používateľ presúvať stanice myšou. Tieto dva stĺpce sme farebne odlíšili, aby použitie tohto rozhrania bolo jednoznačné. Aby používateľ vedel, ako zmeniť zastúpenie staníc v stĺpci, zmení sa kurzor nad názvom stanice. Výber staníc sa potvrdzuje pomocou tlačidla Aktualizovať. Jednotlivé stanice sú prehľadne oddelené.



#### 14.7.4 Implementácia

Pre implementáciu boli použité technológie HTML a CSS, pre zobrazenie okna a posúvanie medzi stĺpcami sme použili JavaScript a jQuery. Pre dynamické prvky bol použitý jazyk Ruby.

## 15 13. Šprint

### 15.1 Grafické rozhranie kontaktnej stránky projektu

#### 15.1.1 Úloha

Vytvorte grafické používateľské rozhranie pre projekt TeleVido – modul kontakt TeleVido. Rozhranie musí spĺňať všetky štandardy modernej webovej stránky a musí zahŕňať všetky základné položky pre korektnú prevádzku uvedeného modulu aplikácie.

#### 15.1.2 Analýza

Pred konštruovaním návrhu grafického používateľského rozhrania sme si dôkladne zanalyzovali grafické rozhrania podobne zameraných aplikácií. Poukázali sme na slabé stránky a takisto aj na silné stránky analyzovaných riešení.

#### 15.1.3 Návrh

Grafické používateľské rozhranie pre kontakt je zobrazené na obrázku č. 48.

V tejto časti si opíšeme jednotlivé požiadavky na navrhovaný modul:

- uvedenie základných informácií o tíme TeleVido (názov tímu, adresa, e-mail, web, telefón)
- uvedenie motivácie na kontaktovanie tímu TeleVido
- predstavenie jednotlivých členov s uvedením mena a osobného kontaktu
- znázornenie mapy sídla tímu TeleVido

#### 15.1.4 Implementácia

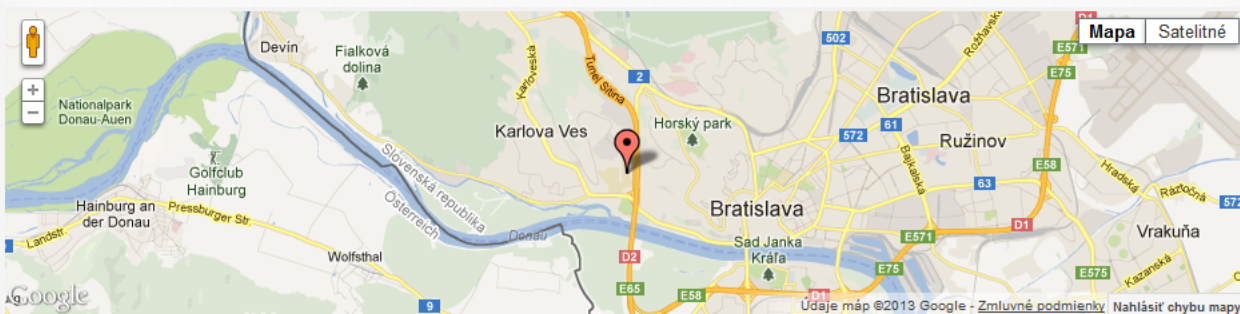
Grafické používateľské rozhranie bolo implementované v HTML5 a CSS3. Pre vykreslenie rôznych grafických animácií sme využili JavaScript a jQuery. Opisovaný modul aplikácie TeleVido je v súčasnosti k dispozícii na webovej adrese: <http://www.televideo.tv/kontakt>

### 15.2 Funkcia “Štrng” v android aplikácii

#### 15.2.1 Úloha

Implementuje a nasadíte funkcionalitu v android aplikácii na základa čoho bude možné skupinovo odporúčať skupinovo filmy na základe profilu používateľov.

KONTAKTNÉ ÚDAJE

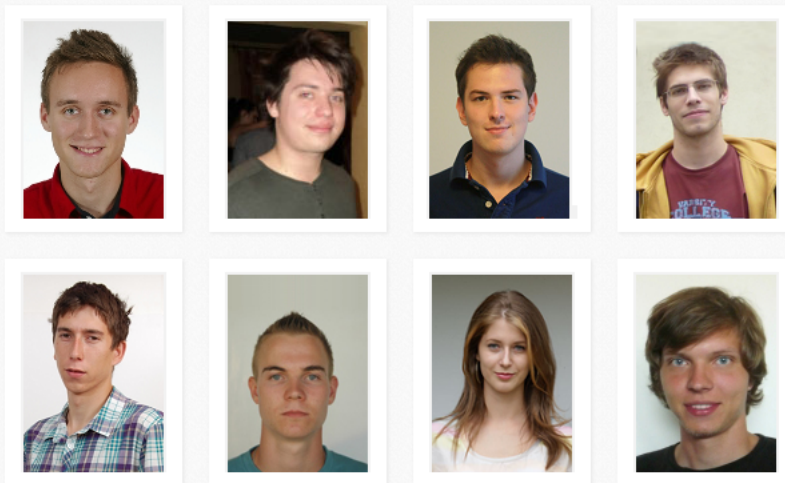


KONTAKTNÉ ÚDAJE

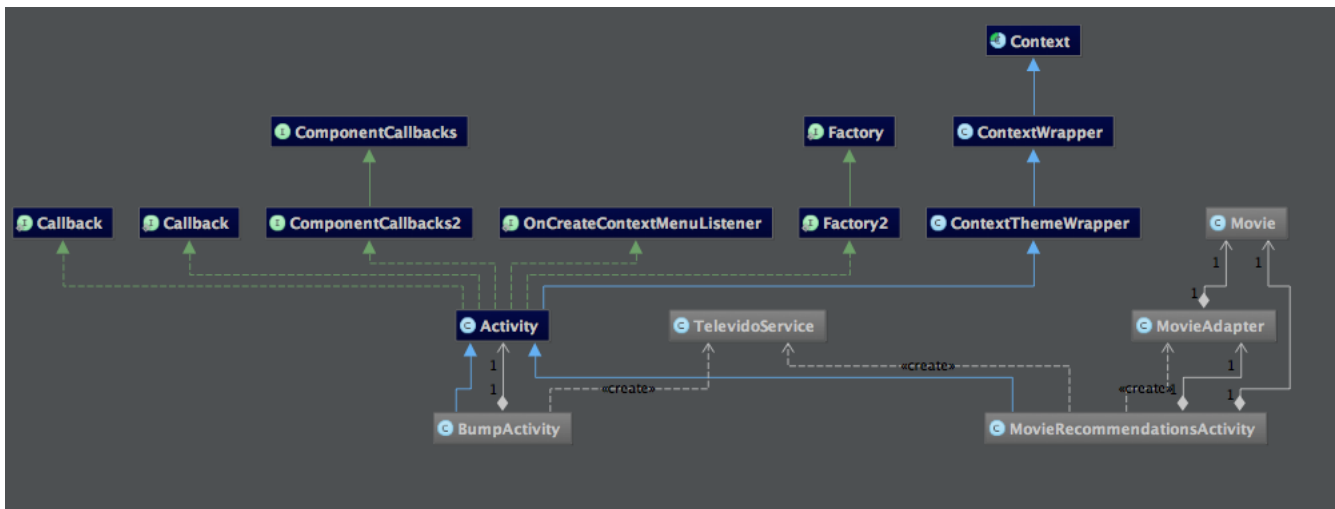
Máte zaujímavý nápad ako projekt **Televido** vylepšiť? Čo by ste veľmi radi uvítali? Napíšte nám svoje myšlienky môžete tak prispieť k následnému zlepšeniu projektu.

- Tím** TeleVido z [FIIT STUBA](#)
- Adresa** Ilkovičova 3, 842 16 Bratislava 4
- E-mail** [info@televido.tv](mailto:info@televido.tv)
- Web** [www.televido.tv](http://www.televido.tv)
- Telefón** +421 904 271 989

ČLENOVIA TÍMU



Obr. 48: Grafické používateľské rozhranie pre modul kontakt



Obr. 49: Diagram tried reprezentujúci funkcionality pre skupinové odporúčanie

### 15.2.2 Analýza

Zobrazenie pre používateľa by mala byť intuitívna. Mali by sa zobraziť niektoré atribúty filmov a aby používatelia mali možnosť na zobrazenie filmov. Ďalej odporúčanie by malo aj zobrazovať čas premietania.

### 15.2.3 Návrh

Pre zobrazenie filmov sme rozhodli použiť záznam s vlastným grafickým rozhraním kde bude poster filmu a informácie o ňom. Na zobrazenie vzdialenosť kín sme rozhodli používať slider.

### 15.2.4 Implementácia

Aktivita ktorá bude zobrazovať odporúčanie filmov bude volať služby servisnej vrstvy vo vlastnom vlákne. Takto sa zabezpečí beh hlavného vlákna ktorá sa stará a grafické rozhranie aplikácií. Zobrazenie jednotlivých záznamov bude uskutočnená pomocou komponentu ListActivity. Pomocou ArrayListAdapterov sa spraví zobrazenie jednolitých záznamov v liste. Vytváranie skupín bude zrealizovaná pomocou časovej pačiatky dopytu a lokácií používateľa. Lokácia používateľa bude získana pomocou POJO triedy ktorá používa android geolokáciu. POJO trieda do konštruktora dostane kontext aktivity na základe čoho bude určovať geolokačnú polohu. Po realizácii vytváranie skupín sa zavolá aktivita ktorá aj slúžila na odporúčanie filmov.

Na obrázku 49 je diagram tried ktorý reprezentuje funkcionality pre skupinové odporúčanie v android aplikácií.

## 15.3 Môj TV-program

### 15.3.1 Úloha

V časti Môj TV program, je potrebné implementovať metódu, ktorá bude vyberať pre používateľa TV program na celý deň, tak aby si používateľ mohol pozrieť čo najviacej relácií s najväčšou možnou energiou, ktorú tie relácie majú z odporúčania.

### 15.3.2 Analýza

Pre implementovanie tejto metódy potrebujeme analyzovať algoritmy dynamického programovania. Celú úlohu si vieme transformovať do problému intervalov. Predstavme si, že máme veľa intervalov, ktoré sa môžu prekrývať a našim cieľom je vybrať intervaly tak, aby sme ich vybrali, čo najviacej alebo aby sme maximalizovali efektívnosť.

V prvom prípade, ak by sme mali maximalizovať počet intervalov, stačí ak intervaly usporiadame podľa času konca a potom postupne prechádzame od začiatku a greedy metódou postupne pridávame intervaly. Ak posledný pridaný interval má čas konca menší ako začiatok ďalšieho tak ho pridáme. Takto vždy zaručene získame maximálny počet intervalov - maximálne počet relácií za deň, ale nie je to vždy ten najlepší výber, pretože nezohľadňujeme energiu, ktorú dostaneme z odporúčania.

V druhom prípade keď chceme vybrať relácie tak, aby sme maximalizovali celkovú energiu je potrebné použiť dynamické programovanie. V princípe si rozdelíme priestor na časy začiatkov a koncov podľa zadaných intervalov. Nás bude zaujímať akú najväčšiu energiu dokážeme získať v jednom čase. Dynamicky pôjdeme od najmenšieho času a postupne až po najväčší s tým, že výsledná energia bude v najväčšom čase. Počas iterácie medzi jednotlivými intervalmi, budeme zisťovať maximálnu energiu v konci daného intervalu. Tá energia môže vzniknúť tak, že pridáme daný interval do výsledného zoznamu a pripočítame tam zoznam intervalov v mieste kde začína daný interval, alebo ho tam nepridáme. Vždy pri každom pridaní je potrebné si zapamätať aj ktorý interval sme pridali, aby sme na konci dostali aj celkový zoznam výsledných intervalov.

### 15.3.3 Návrh

Zoznam odporúčaných relácií implementujeme do samostatného okna, ktoré je možné prepínať zo zoznamu všetkých staníc.

### 15.3.4 Implementácia

Implementácia metódy bola v RubyOnRails za pomoci analyzovaných algoritmov dynamického programovania. Vstupom metódy bol zoznam odporúčaných relácií a ich energie.

## 16 14. Šprint

### 16.1 Grafický návrh nástenky pre prihláseného používateľa

#### 16.1.1 Úloha

Vytvorte grafické používateľské rozhranie pre projekt TeleVido – modul nástenka TeleVido. Rozhranie musí spĺňať všetky štandardy modernej webovej stránky a musí zahŕňať všetky základné položky pre korektnú prevádzku uvedeného modulu aplikácie.

#### 16.1.2 Analýza

Pred konštruovaním návrhu grafického používateľského rozhrania sme si dôkladne zanalyzovali grafické rozhrania podobne zameraných aplikácií. Poukázali sme na slabé stránky a takisto aj na silné stránky analyzovaných riešení.

#### 16.1.3 Návrh

Grafické používateľské rozhranie pre nástenka je zobrazené na obrázku č. 50.

V tejto časti si opíšeme jednotlivé požiadavky na navrhovaný modul: rozdelenie nástenky na dve vertikálne časti, pričom ľavá časť bude dominantná

- v ľavej časti sa budú nachádzať odporúčane s filmy s uvedením názvu filmu, roku vydania, hodnotenia a s rôznymi funkciami podľa uváženia
- v ľavej časti sa budú nachádzať notifikácie pre nadchádzajúce udalosti v TV, v kine, ako aj vo filmovom svete
- v ľavej časti budú zobrazené filmy, ktoré si používateľ označil ako filmy, ktoré chce vidieť
- v pravej časti sa bude nachádzať profilová fotka používateľa
- v pravej časti budú zobrazené odporúčania na programy v TV
- v pravej časti budú zobrazené odporúčania na aktuálne filmy v kine

#### 16.1.4 Implementácia

Grafické používateľské rozhranie bolo implementované v HTML5 a CSS3. Pre vykreslenie rôznych grafických animácií sme využili JavaScript a jQuery. Opisovaný modul aplikácie TeleVido je v súčasnosti k dispozícii na webovej adrese po prihlásení registrovaného používateľa: <http://www.televido.tv/profil/nastenka>

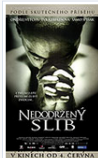
### FILMY PRE TEBA



**FIREPROOF (2008)** **75%**

Caleb Holt zachraňuje ako hasič ostatní ľudí, ale vlastný manželstvi sa mu rozpadá. Kým už stojí na prahu rozvodu, otec ho požiada, aby vyzkoušel čtyřlícetenný experiment na záchranu manželství. Jenže aby uspěl, musí nejprve vyřešit svůj vztah k Bohu. Hrdinou romantického dramatu s náboženským podtextem je kapitán Caleb Holt (Kirk Cameron), odvážný požárník, který neváhá riskovat svůj život pro záchranu ostatních. Své manželství s...

[Chcem vidieť](#)



**NEDODRŽENÝ SĽUB (2009)** **73%**

Drama skutočného osudu slovenského Žida Martina Friedmanna za druhej svetovej vojny. Talentovaný futbalista z rozvätenej rodiny si zpočiatku neuvvedomuje blížiť sa nacistické nebezpečí a dobrovoľne sa vydáva do pracovného tábora v Sereďi, kde mu kamarád Fred zašiel miesto vo futbalovom tímu. Ďalší cesta zavede Martina také do pľúcneho sanatória, do kláštiera i k partyzánom, své židovství na mnoha z týchto miest musí zapírat. Snímek predkládá příběh...

[Chcem vidieť](#)



**VEĽKÝ REŠPEKT (2008)** **46%**

Komédia o mladých ľuďoch, žijúcich na najväčšom bratislavskom sídlisku, ktorí hľadajú kúsok svojho šťastia. Nerozluční kamaráti Peter a Mišo sa spolu potkajú, snažia sa zistiť, ako zapôsobia na baby a odovzdávajú si "životné múdrosti". Petrovi sa páči Michalova mladšia sestra Veronika. Tá má však oči pre iného... Zhodou okolností sa im do života pripletie okradnutý maďarský turista Zoltán. Snažia sa mu pomôcť, ako vedia. Situácia sa však...

[Chcem vidieť](#)



**TANGO S KOMÁRMÍ (2009)** **41%**

Tango s komármí je jedinečným výrazom ľudských citov, je v ňom vášň a napätie, konflikt i súlad pohľadá, dráždenie i bolesť, ako v bodnutí komára, ktorý si neodbytné žiada kvapku krvi, aby mohol prežiť. Tango s komármí je letným obrazom konfliktov mužov a žien. Rozpletá tajomstvá ľudských osudov, ktoré by mohli dospieť aj k ozajstným drámam. Že tieto sú viac trápny než nedorozumeniami ako rafinovanými zločinmi je celkom zámerné. Dvaja emigranti...

[Chcem vidieť](#)



**ONLY THE LONELY (1991)** **57%**

Danny Muldon dobrácky policajt a starý mládenec. I když má dost přátel a všichni ho široko-daleko znají jako výborného člověka a mají ho upřímně rádi, vysedává doma a nechá se týrat a šikanovat svou dobrou maminkou Rose. Despotická a nesnesitelně přísná matka je hrdá Irka, která ze všeho nejvíce nesnáší cizince. Jejím jediným životním zájmem je Danny, kterého mužič neustále starostlivostí. Syn by se rád oženil, ale s matkou po boku se mu to...

[Chcem vidieť](#)



### TV ODPORÚČANIA [Zobrazit Môj TV program](#)

- NOVÉ BÝVANIE**  
práve ide v televízii
- SIMPSONOVCI**  
Dnes, 19:30 -20:00
- MY SON, MY SON, WHAT HAVE YOU DONE**  
Dnes, 20:00 -21:35
- HOUSE OF LIES**  
Dnes, 22:00 -22:30
- CALIFORNICATION - ORGIE V KALIFORNIÍ**  
Dnes, 22:30 -23:00

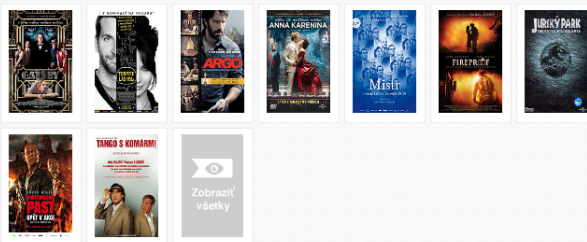
### AKÉ FILMY V KINE [Zobrazit všetky odporúčenia](#)

- JURASSIC PARK**  
Po fantastickém nálezu miliony let starých D.N.A. se pokusí skupina ambiciózních vědců vytvořit přírodní park s žijícími prehistorickými zvířaty na vzdáleném liduprázdném ostrově. Celý projekt podporuje milionář John Hammond (Richard... [Viac](#)
- BROKEN CITY**  
Billy byl kdysi chloubou policejního sboru, jenže když jeden gauner zničí a zavraždí sestru jeho přítelkyně, rozhodl se, že trochu obejde spravedlnost a potrestá ho sám trestem nejvyšším. Z obvinění z vraždy ho vyseká přimluva kamaráda u... [Viac](#)
- SNITCH**  
Aby zachránil svého pubertálního syna před nespravedlivým uvězněním, infiltruje se zoufalý otec do nebezpečného drogového kartelu.(oficiální text distributora) [Viac](#)

### NOTIFIKÁCIE

**Film Jurassic Park** ktorý si chcel vidieť dávajú dnes o 19:20 v Cinema City Aupark, zajtra o 12:10 v Cinema City Aupark, zajtra o 14:00 v Cinema City Polus, zajtra o 14:15 v Cinema City Eurovea, zajtra o 14:40 v Cinema City Aupark.

### FILMY, KTORÉ CHCEM VIDIEŤ



[Zobrazit všetky](#)

Obr. 50: Grafické používateľské rozhranie pre modul nástenka

## 16.2 Nástenka prihláseného používateľa

### 16.2.1 Úloha

Implementujte a nasadte domovskú stránku pre používateľa. Každý používateľ keď sa prihlási vidí presonalizované údaje pre seba. Cieľom úlohy je vymyslieť backend a tiež poskytované údaje na domovskej stránke.

### 16.2.2 Analýza

Pri porovnávaní s existujúcimi riešeniami a po zohľadnení toho, čo na titulke chceme, aby používateľ mal sme sa rozhodli, že budeme poskytovať:

- odporúčané filmy odporúčané filmy v kinách
- TV program na niekoľko hodín dopredu
- notifikácie, ktoré oznamujú používateľovi, že film, ktorý chcel vidieť ide v kine alebo TV
- posledné filmy, ktoré označil, že chce vidieť

Okrem toho sme do domovskej stránky pridali možnosť upraviť profilovú fotku.

### 16.2.3 Návrh

Nakoľko všetky operácie, ktoré zobrazovali a vytvárali danú nástenku boli pomalé (2-3 sekundy) museli sme vymyslieť, aby sa vždy všetko predpočítavalo a memorizovalo. Následne by sme to len vyberali z tabuliek a tak celý proces zobrazovania titulky zrýchlili.

### 16.2.4 Implementácia

Nástenku sme implementovali v RubyOnRails. Využili sme multivláknové procesy, ktoré sa spúšťajú pravidelne každých 10 minút a tak sa domovská stránka aktualizuje vždy podľa potreby a pravidelne. Na nástenke sme využili novú tabuľku, ktorá obsahuje polymorfické triedy.

## 16.3 Doladenie android aplikácie

### 16.3.1 Úloha

Android aplikácia obsahuje základ všetkej plánovanej funkcionality, ale je potrebné ju doladiť a dokončiť detail filmu, detail osoby a výpis odporúčania z kín. Je potrebné odstrániť chyby, doplniť chýbajúce texty a podobne.

### 16.3.2 Analýza

Aplikácia je v dobrom stave hoci sa to z používateľského hľadiska nezdá, pretože často padá. Toto je zvyčajne spôsobené parsovaním JSON odpovedí API, ktoré poskytuje naša webová služba. Tieto JSON objekty sú nie vždy kompletne a treba tomu prispôsobiť parser aplikácie.

Niektoré časti kódu aplikácie vznikli kopírovaním iných častí bez dôkladného testovania, čo tiež spôsobuje problémy. Je potrebné kód odladiť, nájsť chyby a opraviť ich.

Aplikácia nie vždy zobrazuje všetky potrebné informácie tak, ako sú v grafickom návrhu rozhrania a ako sú prijaté z API. Chýbajúce údaje je potrebné doplniť.

### 16.3.3 Návrh

Architektúra aplikácie je navrhnutá, je potrebné iba doplniť chýbajúce časti, opraviť chyby prípadne prerobiť nevhodne napísané časti.

### 16.3.4 Implementácia

Zadanie bolo implementované do existujúcej Android aplikácie, ktorá používa Android SDK a je postavená na Jave. Aplikácia pracuje s našim API, ktoré vracia odpovede v JSON a spracováva ich pomocou knižnice google-gson.

## 16.4 Verejná dokumentácia k API

### 16.4.1 Úloha

Vytvorte verejne prístupnú dokumentáciu k vytvorenému API tak, aby bolo jasné použitie a význam tohto rozhrania.

### 16.4.2 Analýza

Dostupné služby je potrebné zobraziť prehľadne tak, aby sa v nich dalo rýchlo orientovať. Ku každej službe je potrebné napísať opis, dostupnú metódu, url, telo volania a odpoveď. Táto dokumentácia sa bude nachádzať na samostatnej podstránke. Pri návrhu sme sa inšpirovali inými dostupnými dokumentáciami nachádzajúcimi sa na internete.

### 16.4.3 Návrh

V záhlaví sa nachádza ponuka z odkazmi na jednotlivé sekcie. Služby rozdelíme do sekcií ako sú filmy, ľudia, odporúčanie a pod. Pre každú službu sa budú na stránke nachádzať rovnaké časti a to: názov, opis, metódy a príklad, prípadne vysvetlenie jednotlivých parametrov (Obr. 51).



## Pre filmy

Automatické dopĺňanie názvu filmu.

| Metóda | Uri                                             | Telo (JSON)                   |
|--------|-------------------------------------------------|-------------------------------|
| Post   | api/autocomplete/movies                         | {"term": "nazov_filmu_ciasť"} |
| Get    | api/autocomplete/movies?term= nazov_filmu_ciasť |                               |

### Príklad

```

METÓDA: POST
URL: api/autocomplete/movies
TELO: {"term": "closet"}
ODPOVEĎ: [{"title_orig": "Skeletons in the Closet"}, {"title_orig": "Rescued From the Closet"}, {"title_orig": "Closet Land"}, {"title_orig": "Oh Dad, Poor Dad, Mama's Hung You in the Closet and I'm Feeling So Sad"}, {"title_orig": "Monster in the Closet"}, {"title_orig": "Celluloid Closet, The"}, {"title_orig": "Oingo Boingo: Skeletons In the Closet"}, {"title_orig": "Veronica's Closet"}, {"title_orig": "Cameron's Closet"}, {"title_orig": "Closet Cases of the Herd Kind"}]
    
```

Obr. 51: Dokumentácia k API

### 16.4.4 Implementácia

Ako základ sme použili framework Bootstrap<sup>1</sup>, kde sme upravili potrebné CSS a text zobrazili prostredníctvom HTM

## 17 Používateľská príručka

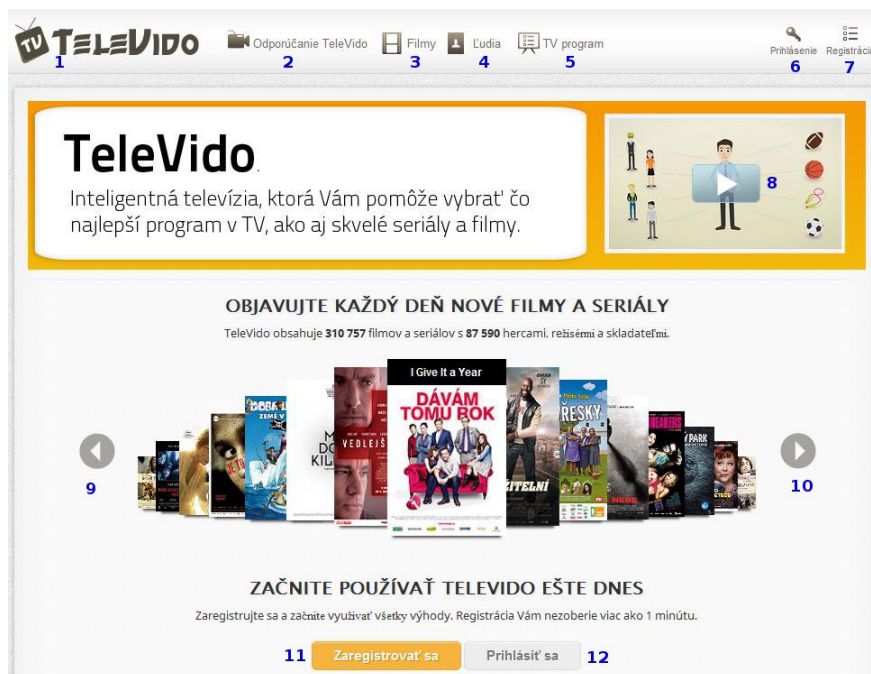
### 17.1 Neprihlásený používateľ

#### 17.1.1 Úvodná stránka

#### Obrázok 52

1. Tlačítko pre zobrazenie úvodnej stránky
2. Tlačítko pre zobrazenie stránky s odporúčaním
3. Tlačítko pre zobrazenie stránky pre prehliadanie filmov
4. Tlačítko pre zobrazenie ľudí – hercov, režisérov a skladateľov
5. Tlačítko pre zobrazenie televízneho programu
6. Tlačítko pre zobrazenie okna pre prihlásenie
7. Tlačítko pre zobrazenie stránky pre registráciu
8. Spustenie videa opisujúceho Televido

<sup>1</sup>[twitter.github.io/bootstrap/](https://twitter.github.io/bootstrap/)



Obr. 52: Úvodná stránka – vrch stránky

9. Tlačítka pre posúvanie ukázkových filmov doľava
10. Tlačítka pre posúvanie ukázkových filmov doprava
11. Tlačítka pre zobrazenie stránky pre registráciu
12. Tlačítka pre zobrazenie stránky pre prihlásenie

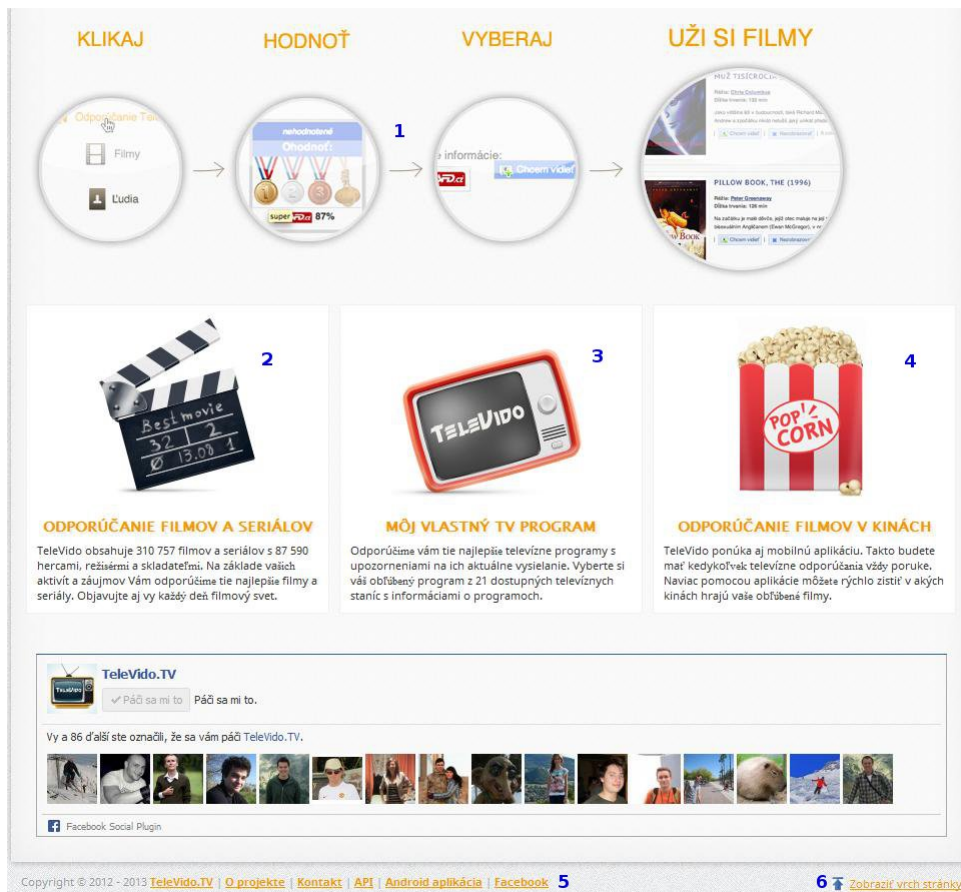
### Obrázok 53

1. Tlačítka pre zobrazenie stránky pre prehliadanie filmov
2. Tlačítka pre zobrazenie stránky s odporúčaním
3. Tlačítka pre zobrazenie stránky s odporúčaním televízneho programu
4. Tlačítka pre zobrazenie stránky s odporúčaním filmov v kinách
5. Tlačítka pre zobrazenie statických stránok s doplňujúcimi informáciami
6. Tlačítka pre zobrazenie vrchu stránky

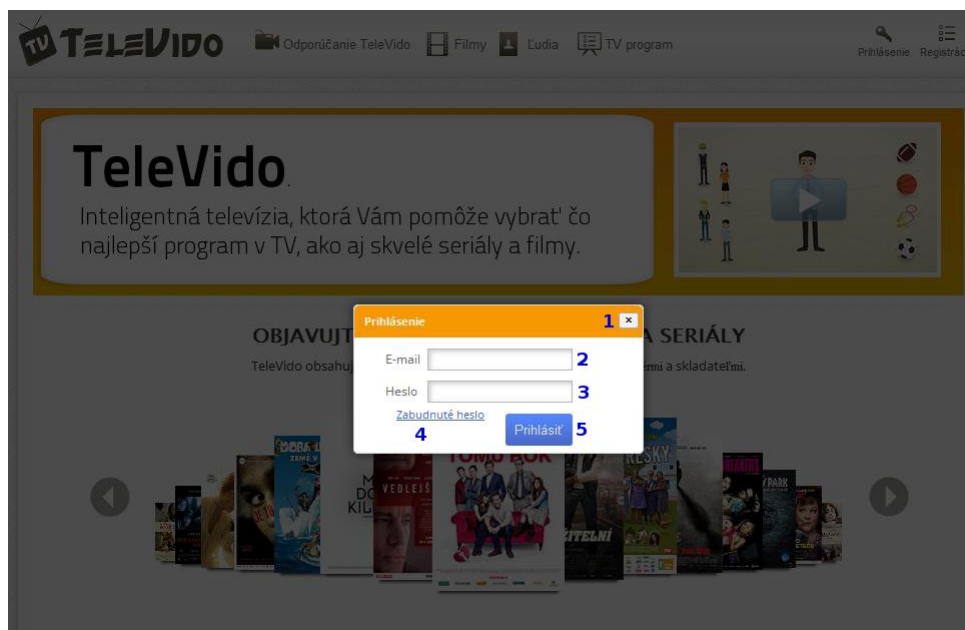
### 17.1.2 Prihlásenie používateľa

#### Obrázok 54

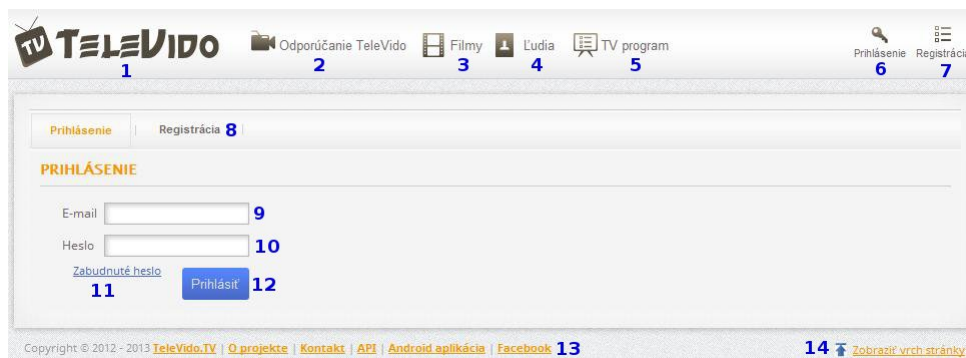
1. Tlačítka pre zrušenie okna pre prihlásenie



Obr. 53: Úvodná stránka – spodok stránky



Obr. 54: Obrazovka pre prihlásenie používateľa

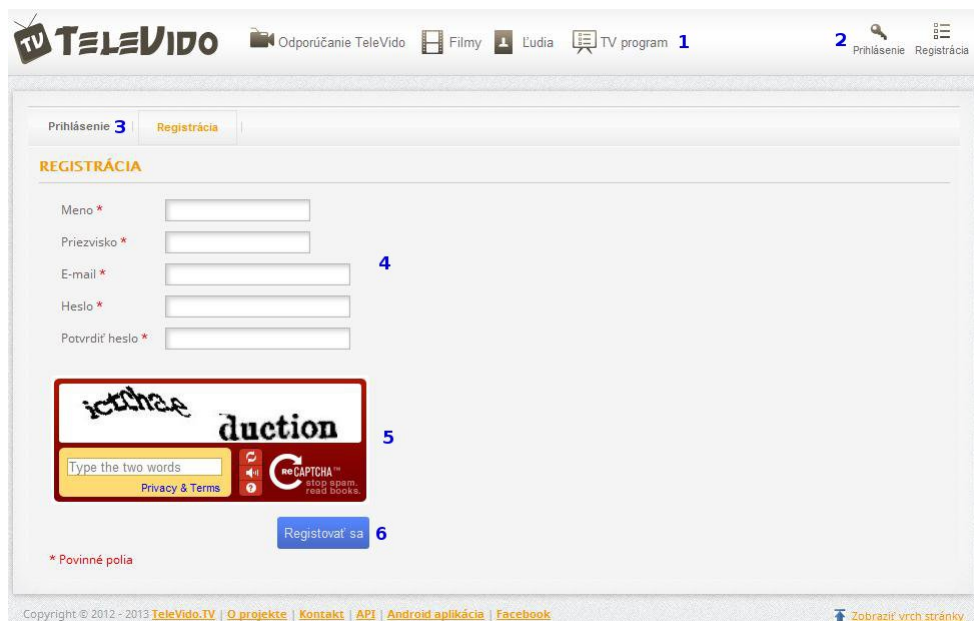


Obr. 55: Stránka pre prihlásenie používateľa

2. Textové pole pre vyplnenie e-mailu používateľa
3. Textové pole pre vyplnenie hesla používateľa
4. Tlačítko pre zobrazenie stránky pre poslanie nového hesla
5. Tlačítko pre odoslanie údajov a následne prihlásenie

#### Obrázok 55

1. Tlačítko pre zobrazenie úvodnej stránky – ďalej už len menu 1
2. Tlačítko pre zobrazenie stránky s odporúčaním – ďalej už len menu 1
3. Tlačítko pre zobrazenie stránky pre prehliadanie filmov – ďalej už len menu 1
4. Tlačítko pre zobrazenie ľudí – hercov, režisérov a skladateľov – ďalej už len menu 1
5. Tlačítko pre zobrazenie televízneho programu – ďalej už len menu 1
6. Tlačítko pre zobrazenie okna pre prihlásenie – ďalej už len menu 2
7. Tlačítko pre zobrazenie stránky pre registráciu – ďalej už len menu 2
8. Tlačítko pre zobrazenie stránky pre registráciu
9. Textové pole pre vyplnenie e-mailu používateľa
10. Textové pole pre vyplnenie hesla používateľa
11. Tlačítko pre zobrazenie stránky pre poslanie nového hesla
12. Tlačítko pre odoslanie údajov a následne prihlásenie
13. Tlačítko pre zobrazenie statických stránok s doplňujúcimi informáciami – ďalej neopisované
14. Tlačítko pre zobrazenie vrchu stránky – ďalej neopisované



Obr. 56: Stránka pre registráciu

### 17.1.3 Registrácia používateľa

#### Obrázok 56

1. Tlačítka pre navigáciu v menu 1
2. Tlačítka pre navigáciu v menu 2
3. Tlačítka pre zobrazenie stránky pre prihlásenie
4. Textové polia pre vyplnenie údajov pre registráciu
5. Captcha
6. Tlačítka pre registráciu používateľa

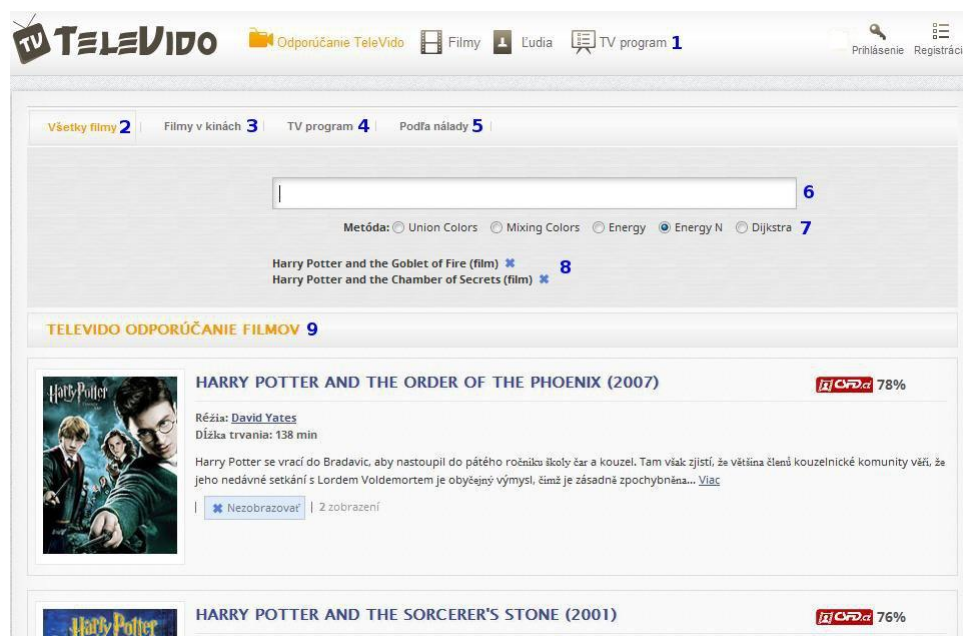
### 17.1.4 Zabudnuté heslo

#### Obrázok 57

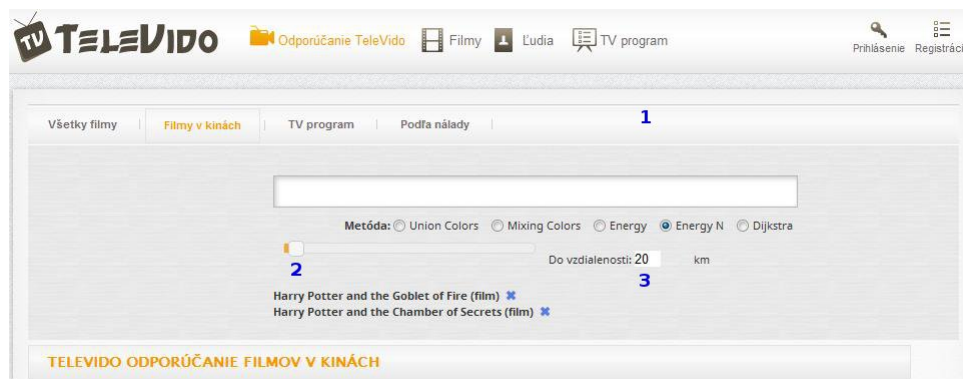
1. Tlačítka pre navigáciu v menu 1 – ďalej už len menu
2. Tlačítka pre navigáciu v menu 2 – ďalej už len menu
3. Textové pole pre e-mail používateľa
4. Captcha
5. Tlačítka pre poslanie hesla používateľovi



Obr. 57: Stránka pre poslanie nového hesla používateľovi



Obr. 58: Odporúčanie zo všetkých filmov



Obr. 59: Odporúčanie filmov v kinách

### 17.1.5 Odporúčanie televido

#### Obrázok 58

1. Tlačítka pre navigáciu v menu
2. Tlačítko pre zobrazenie odporúčania všetkých filmov
3. Tlačítko pre zobrazenie odporúčania filmov v kinách
4. Tlačítko pre zobrazenie odporúčania televízneho programu
5. Tlačítko pre zobrazenie odporúčania podľa nálady
6. Textové pole pre zadanie filmov, podľa ktorých sa má odporúčať
7. Algoritmus odporúčania (pravdepodobne v budúcnosti odstránená možnosť)
8. Filmy, ktoré sme už zadali, zrušiť ich môžeme kliknutím na krížik
9. Odporúčané filmy

### 17.1.6 Odporúčanie v kinách

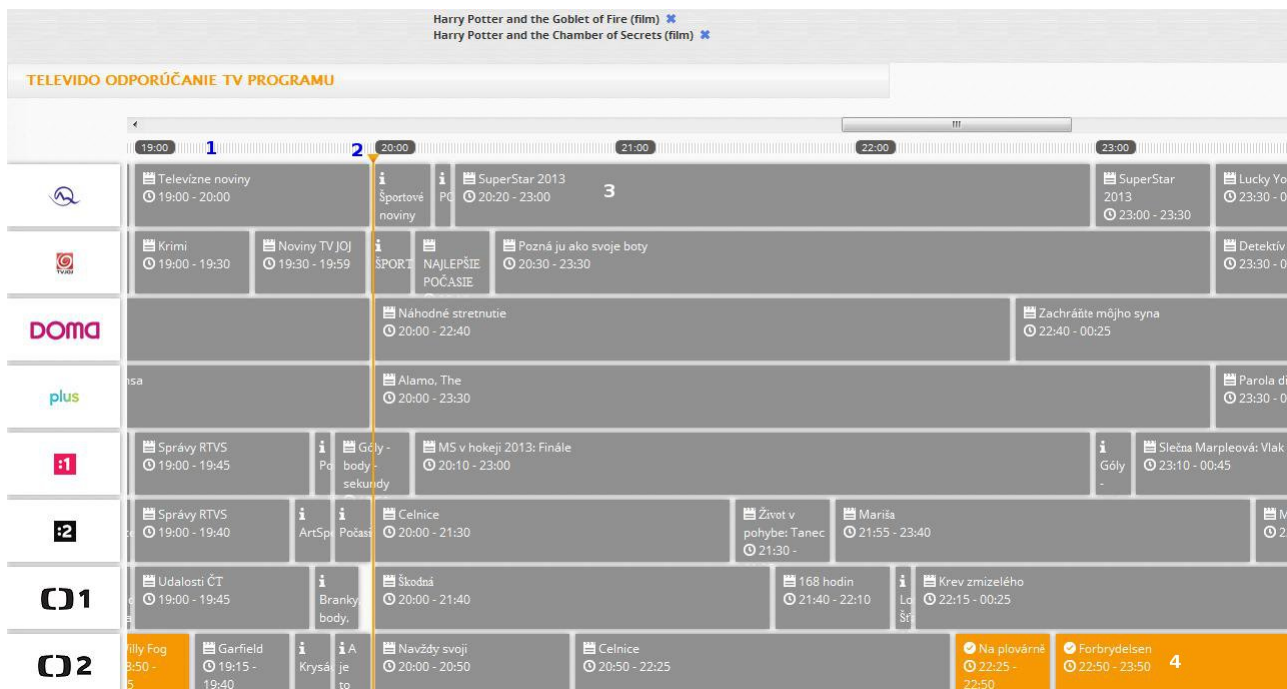
#### Obrázok 59

1. Rovnaké menu ako v opise obr. č. 58
2. Slajder pre nastavenie maximálnej vzdialenosti kina od našej aktuálnej pozície
3. Textové pole pre nastavenie maximálnej vzdialenosti kina od našej aktuálnej pozície

### 17.1.7 Odporúčanie TV programu

#### Obrázok 60

1. Ukazovateľ času pre lepšiu orientáciu



Obr. 60: Odporúčanie televízneho programu

2. Čiara ukazujúca aktuálny čas
3. Film, ktorý nie je odporúčany (šedá farba)
4. Film, ktorý je odporúčany (oranžová farba)

### 17.1.8 Odporúčanie podľa nálady používateľa

Obrázok 61

1. Rovnaké menu ako v opise obr. č. 58
2. Slajder pre nastavenie používateľovej nálady
3. Tvár zobrazujúca nastavenú náladu
4. Možnosť nechať si odporučiť iba tie filmy, ktoré by mali používateľovi náladu zlepšiť
5. Odporúčané filmy

### 17.1.9 Prehliadanie filmov

Obrázok 62

1. Tlačítka pre navigáciu v menu
2. Tlačítka pre preferované zobrazenie aktuálnych filmov



TV **TELEVIDEO** Odporúčanie TeleVido Filmy Ludia TV program Prihlásenie Registrácia

Všetky filmy | Filmy v kinách | TV program | **Podľa nálady** 1

Zadajte film alebo osobu

Metóda:  Union Colors  Mixing Colors  Energy  Energy N  Dijkstra


Harry Potter and the Goblet of Fire (film) ✕  
 Harry Potter and the Chamber of Secrets (film) ✕

Ako sa práve cítite? 3

negatívne   neutrálné  pozitívne

Zlepšíte mi náladu  4

**TELEVIDEO ODPORÚČANIE 5**



**HARRY POTTER AND THE ORDER OF THE PHOENIX (2007)** iZ **OFD.cz** 78%

Réžia: [David Yates](#)  
 Dĺžka trvania: 138 min

Harry Potter se vracia do Bradavíc, aby nastúpil do pátého ročníka školy čar a kouzel. Tam však zistí, že väčšina členů kouzelnické komunity věří, že jeho nedávné setkání s Lordem Voldemortem je obyčejný výmysl, čímž je zásadně zpochybněna... [Viac](#)

[Nezobrazovať](#) | 2 zobrazení

Obr. 61: Odporúčanie podľa nálady

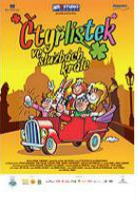
TV **TELEVIDEO** Odporúčanie TeleVido Filmy Ludia TV program 1 Prihlásenie Registrácia

Aktuálne filmy 2 | Všetky filmy 3

4 Vyhľadavanie filmov Nájdí 5

→ Rozšírené vyhľadavanie

Od roku: 1900   7  Do roku: 2012  Žánre: 1 vybrané  9




**ČTYŘLÍSTEK VE SLUŽBÁCH KRÁLE (2012)** iZ **OFD.cz** 53%

Réžia: [Michal Žabka](#)  
 Dĺžka trvania: 90 min

Populární komiksový seriál ČTYŘLÍSTEK oslavil letos už 40 let od svého „narození“, prvního vydání kresleného příběhu autora a výtvarníka Jaroslava Němečka. Teprve nyní se však Fifinka, Myšpálín, Píďka a Bobík dostávají na filmové plátno – a to v... [Viac](#)

| 0 zobrazení



**PARIS - MANHATTAN (2012)** iZ **OFD.cz** 64%

Réžia: [Sophie Lellouche](#)  
 Dĺžka trvania: 74 min

Obr. 62: Prehliadanie filmov

The screenshot shows the website interface for the movie 'GOOD DAY TO DIE HARD, A (2013)'. At the top, there is a navigation bar with 'TV TELEVIDEO' and various menu items like 'Odporúčanie TeleVideo', 'Filmy', 'Ľudia', and 'TV program'. The main content area features a movie poster on the left and a detailed description on the right. A sidebar on the left contains navigation options: 'Obsah', 'Videá', and 'Herci'. Below the main description, there is a section for 'PODOBŇÉ FILMY' with several movie thumbnails. Numbered callouts (1-6) are placed over the page to highlight specific UI elements: 1 points to the top navigation bar, 2 to the movie title, 3 to the rating box, 4 to the genre 'Akční, Krimi, Thriller', 5 to the sidebar navigation, and 6 to a social media button.

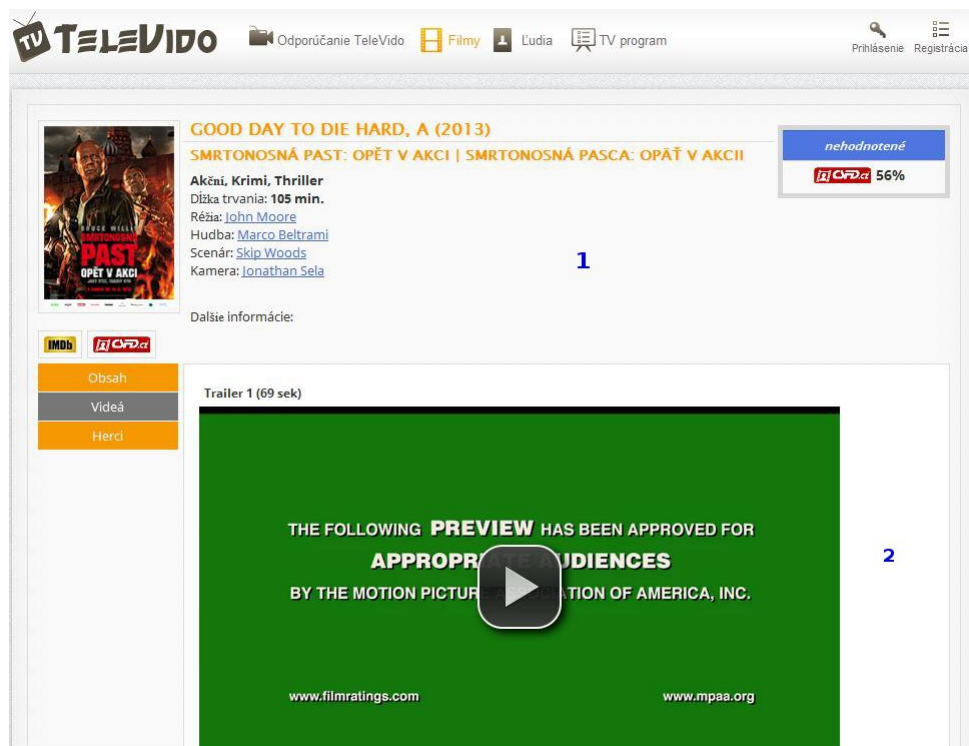
Obr. 63: Detail filmu - obsah

3. Tlačítka pre zobrazenie filmov, bez preferovania aktuálnych (rovnaký štýl obrazovky)
4. Textové pole pre zadanie konkrétneho filmu
5. Tlačítka pre vyhľadanie zadaného filmu
6. Možnosť určiť si rozmedzie rokov vydania filmu – od roku (textové pole)
7. Možnosť určiť si rozmedzie rokov vydania filmu (slajder)
8. Možnosť určiť si rozmedzie rokov vydania filmu – do roku (textové pole)
9. Možnosť vybrať si filmy konkrétnych žánrov
10. Zobrazené filmy

### 17.1.10 Detail filmu

#### Obrázok 63

1. Tlačítka pre navigáciu v menu
2. Originálny, český a slovenský názov filmu
3. Hodnotenie filmu Základné informácie o filme
4. Tlačítka pre navigáciu v detailoch filmu



Obr. 64: Detail filmu - videá

5. Obsah filmu

#### Obrázok 64

1. Rovnaké menu ako v opise obr. č. 63
2. Videá k danému filmu

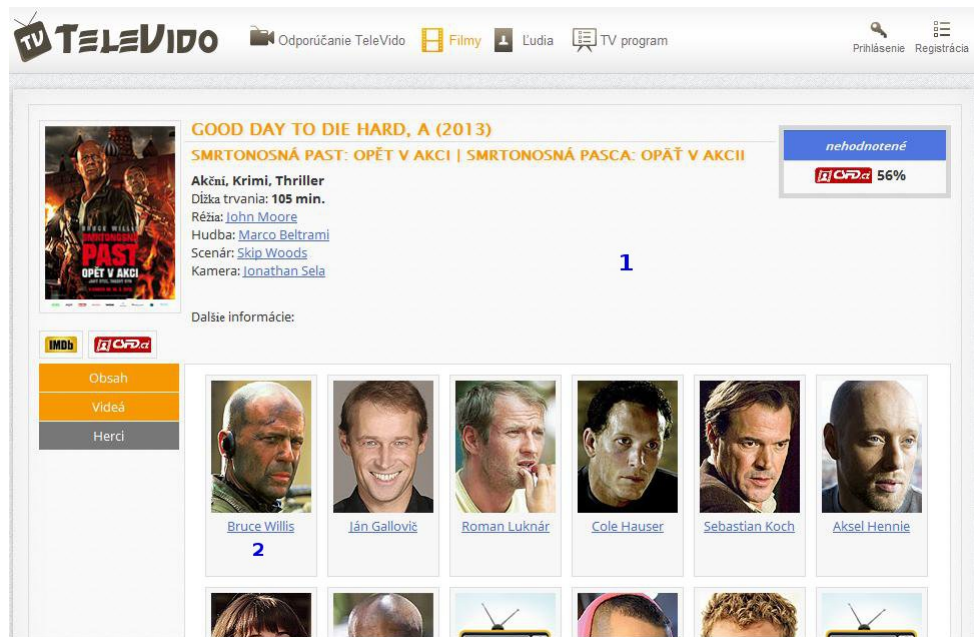
#### Obrázok 65

1. Rovnaké menu ako v opise obr. č. 63
2. Fotografia a meno osoby hrajúcej v danom filme, s možnosť kliknutia na ňu (a zobrazenia ďalších informácií)

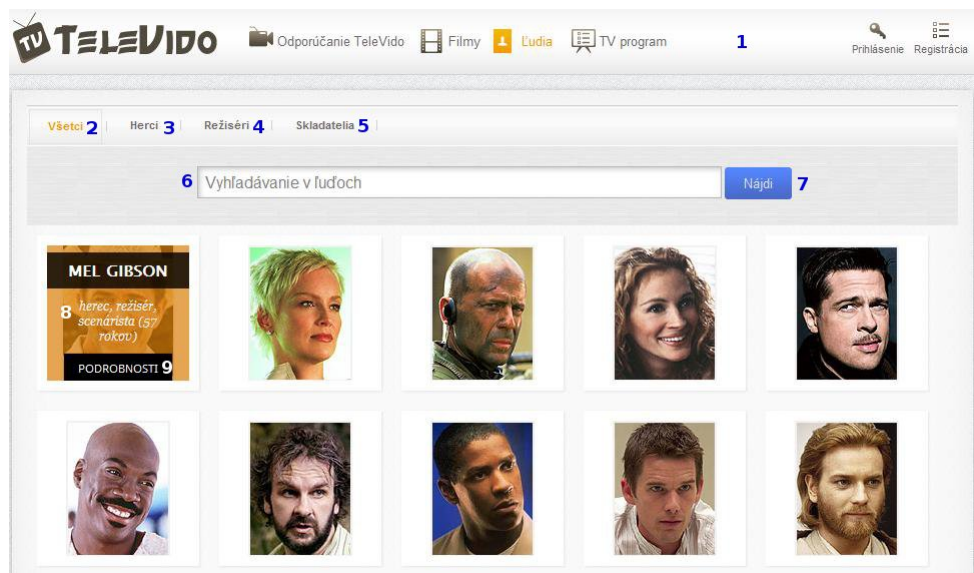
#### 17.1.11 Prehliadanie ľudí (hercov, režisérov, skladateľov)

#### Obrázok 66

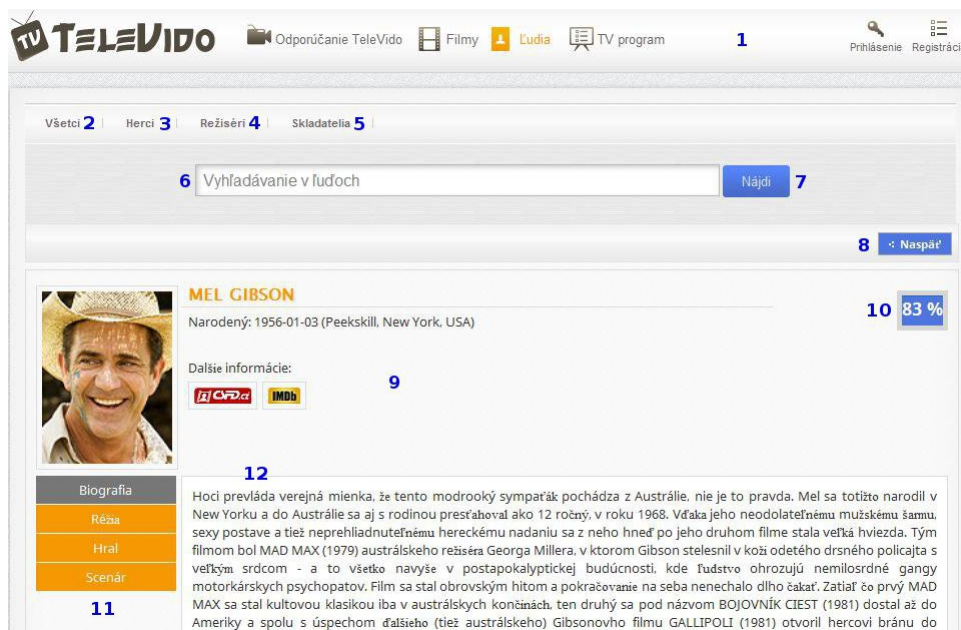
1. Tlačítka pre navigáciu v menu
2. Tlačítka pre prehliadanie všetkých ľudí
3. Tlačítka pre prehliadanie hercov (rovnaký štýl obrazovky ako obr. č. 66)



Obr. 65: Detail filmu - Herci



Obr. 66: Prehliadanie ľudí



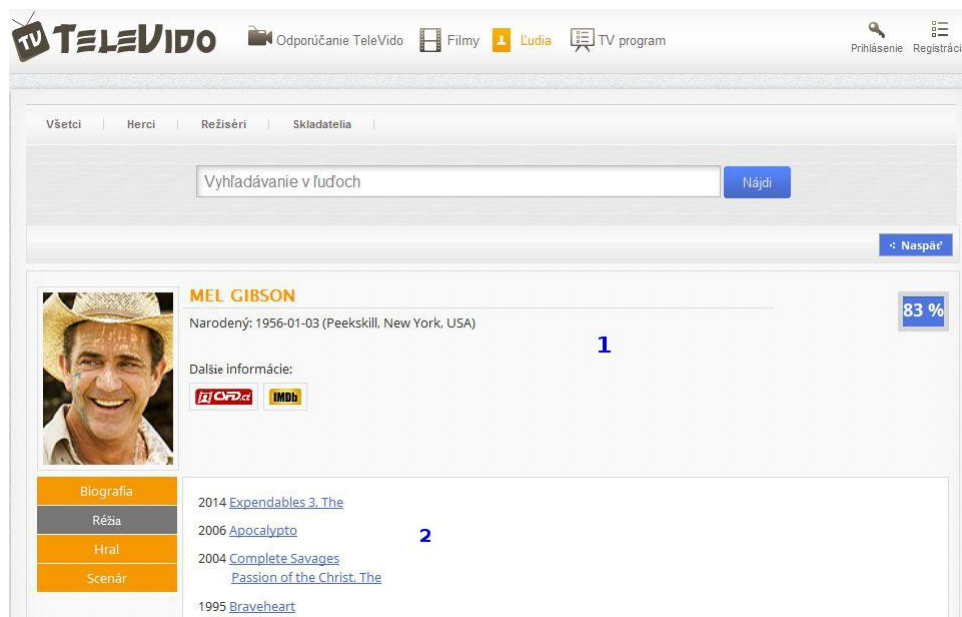
Obr. 67: Prehliadanie ľudí - biografia

4. Tlačítka pre prehliadanie režisérov (rovnaký štýl obrazovky ako obr. č. 66)
5. Tlačítka pre prehliadanie skladateľov (rovnaký štýl obrazovky ako obr. č. 66)
6. Textové pole pre zadanie mena osoby
7. Tlačítka pre vyhľadanie zadanej osoby

#### 17.1.12 Detail osoby

##### Obrázok 67

1. Tlačítka pre navigáciu v menu
2. Tlačítka pre prehliadanie všetkých ľudí (obr. č. 66)
3. Tlačítka pre prehliadanie hercov (rovnaký štýl obrazovky ako obr. č. 66)
4. Tlačítka pre prehliadanie režisérov (rovnaký štýl obrazovky ako obr. č. 66)
5. Tlačítka pre prehliadanie skladateľov (rovnaký štýl obrazovky ako obr. č. 66)
6. Textové pole pre zadanie mena osoby
7. Tlačítka pre vyhľadanie zadanej osoby
8. Tlačítka pre spätnú navigáciu v detailoch osoby
9. Základné informácie o osobe



Obr. 68: Prehliadanie ľudí - Réžia, hral, scenár

10. Hodnotenie osoby
11. Tlačítka pre navigáciu v detailoch osoby
12. Biografia osoby

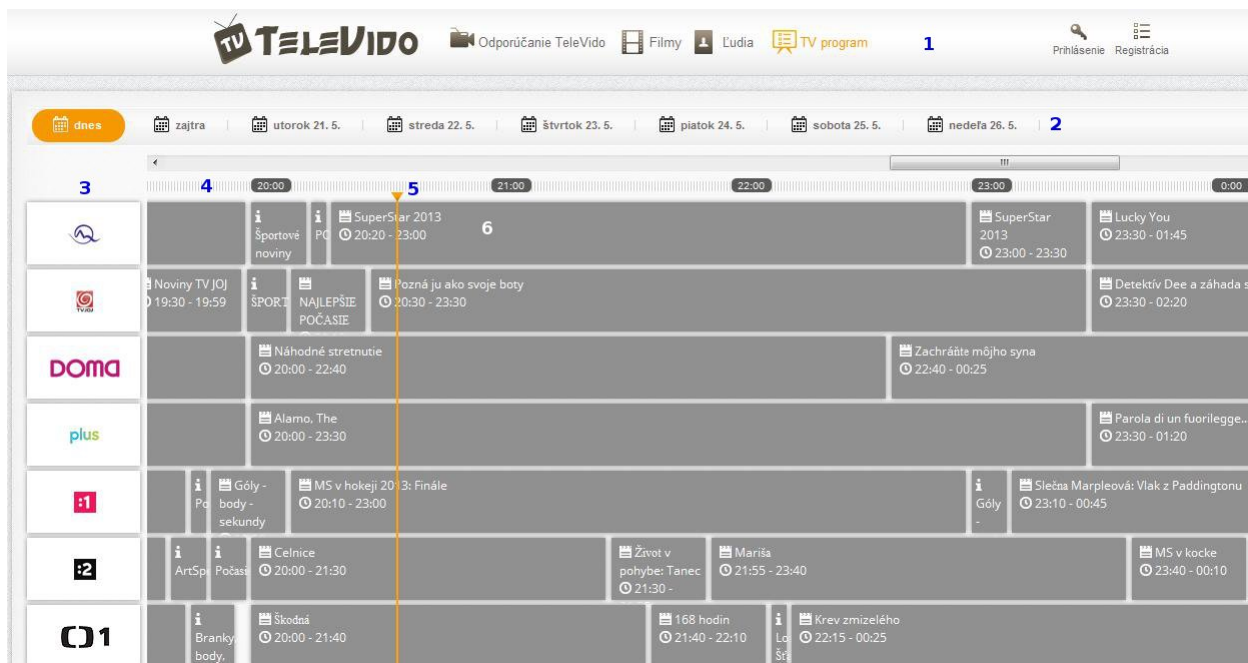
### Obrázok 68

1. Rovnaké menu ako v opise obr. č. 67
2. Rok a názov filmu, ktorý daná osoba režírovala s možnosťou kliknutia na neho (zobrazí sa detail filmu)

### 17.1.13 Zobrazenie televízneho programu

#### Obrázok 69

1. Tlačítka pre navigáciu v menu
2. Tlačítka pre určenie dátumu, pre ktorý sa má televízny program zobrazit'
3. Zoznam televíznych staníc
4. Ukazovateľ času pre lepšiu orientáciu
5. Čiara ukazujúca aktuálny čas
6. Konkrétny televízny program (film, relácia, seriál, atď.)



Obr. 69: Zobrazenie televízneho programu

## 17.2 Prihlásený používateľ

Prihlásený používateľ má tie isté možnosti práce s aplikáciou ako boli opísané vyššie. Rozdiely sú opísané v nasledujúcej kapitole.

### 17.2.1 Nástenka prihláseného používateľa

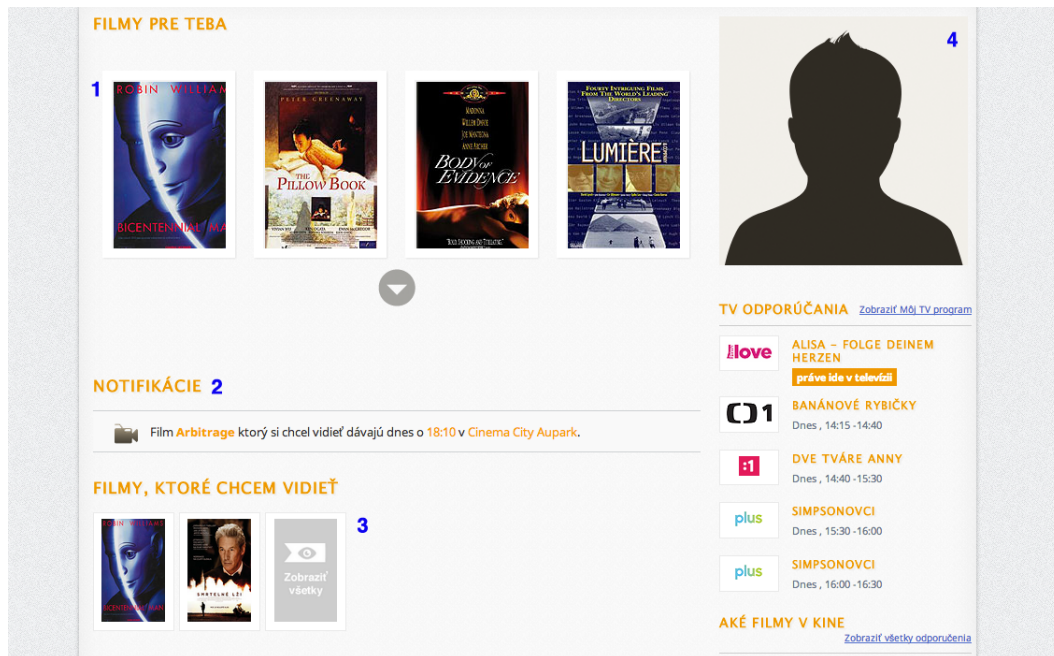
#### Obrázok 70

1. Filmy, ktoré používateľovi odporúča aplikácia
2. Notifikácie pre používateľa
3. Filmy, ktoré používateľ označil, že chce vidieť
4. Profilová fotka používateľa

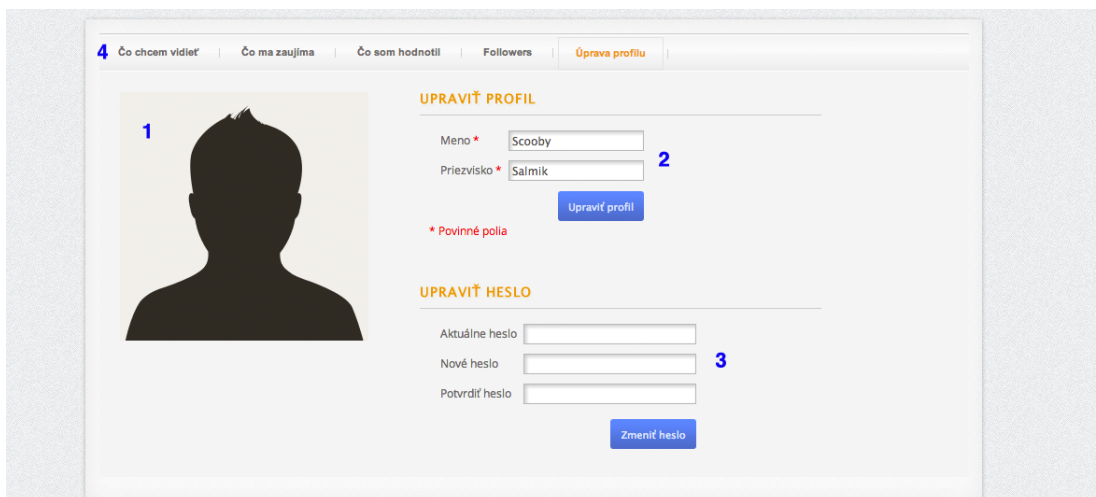
### 17.2.2 Profil používateľa

#### Obrázok 71

1. Úprava fotky používateľa (po prejdení myšou sa zobrazí tlačidlo na úpravu fotky pomocou služby Gravatar)
2. Úprava mena a priezviska používateľa
3. Úprava hesla používateľa
4. Menu s akciami

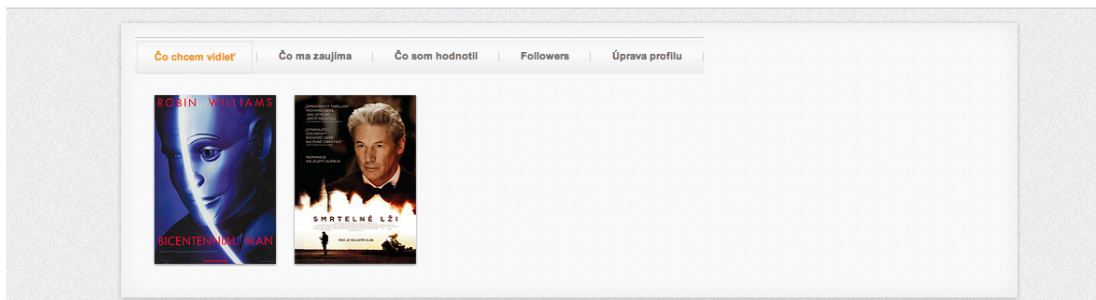


Obr. 70: Nástenka prihláseného používateľa

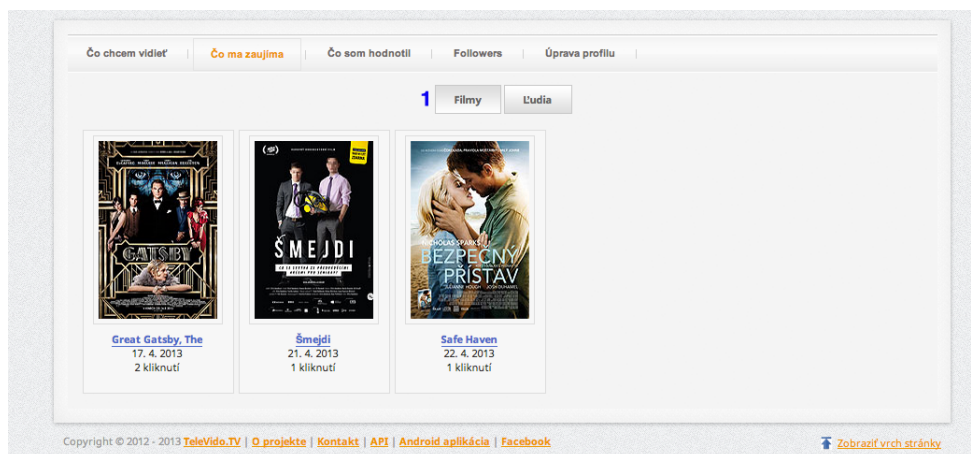


Obr. 71: Úprava profilu používateľa





Obr. 72: Filmy, ktoré chce používateľ vidieť



Obr. 73: Filmy, ktoré používateľa zaujímajú

**Obrázok 72** Pod touto položkou menu používateľ nájde filmy, ktoré označil, že ich chce vidieť.

**Obrázok 73**

1. Prepínač pre filmy a ľudí, ktoré/í používateľa zaujímajú

**Obrázok 74**

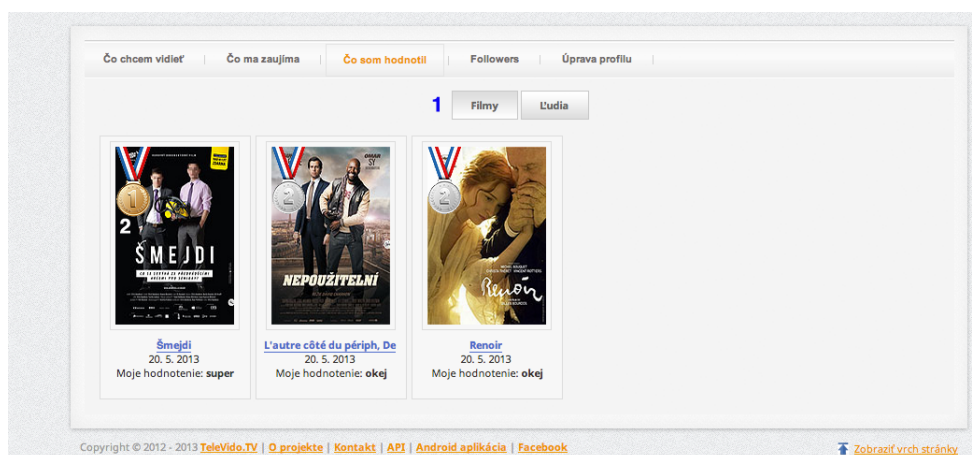
1. Prepínač pre filmy a ľudí
2. Zobrazenie, ako používateľ hodnotil konkrétny film

**Obrázok 75** Na tejto obrazovke používateľ vidí, ktorých používateľ followuje, a ktorí používatelia followujú jeho.

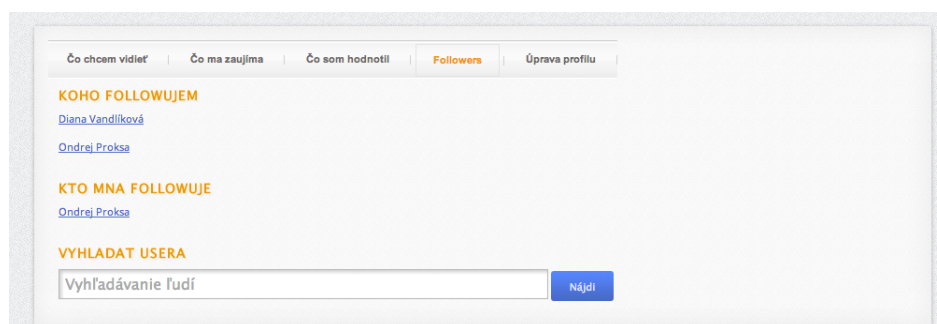
### 17.2.3 Hodnotenie filmu prihláseným používateľom

**Obrázok 76**

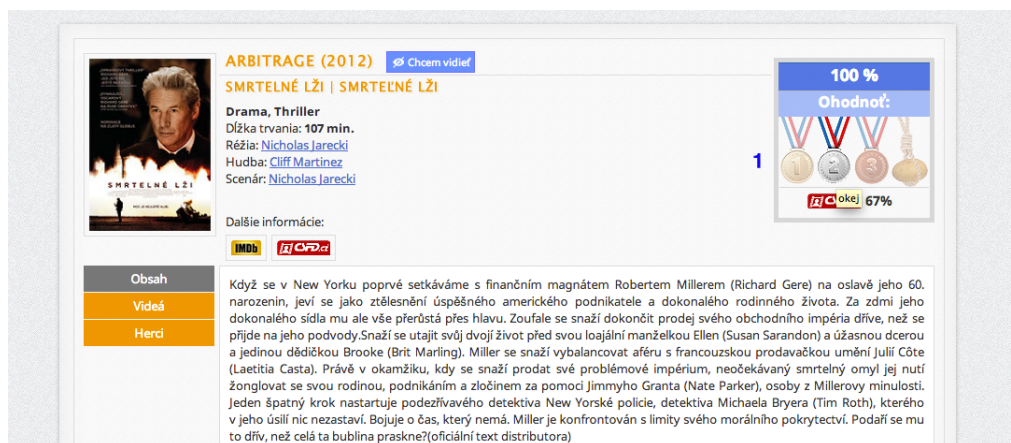
1. Hodnotenie môže používateľ zvoliť zo 4 možností, kde zlatá medaila je najlepšie hodnotenie.



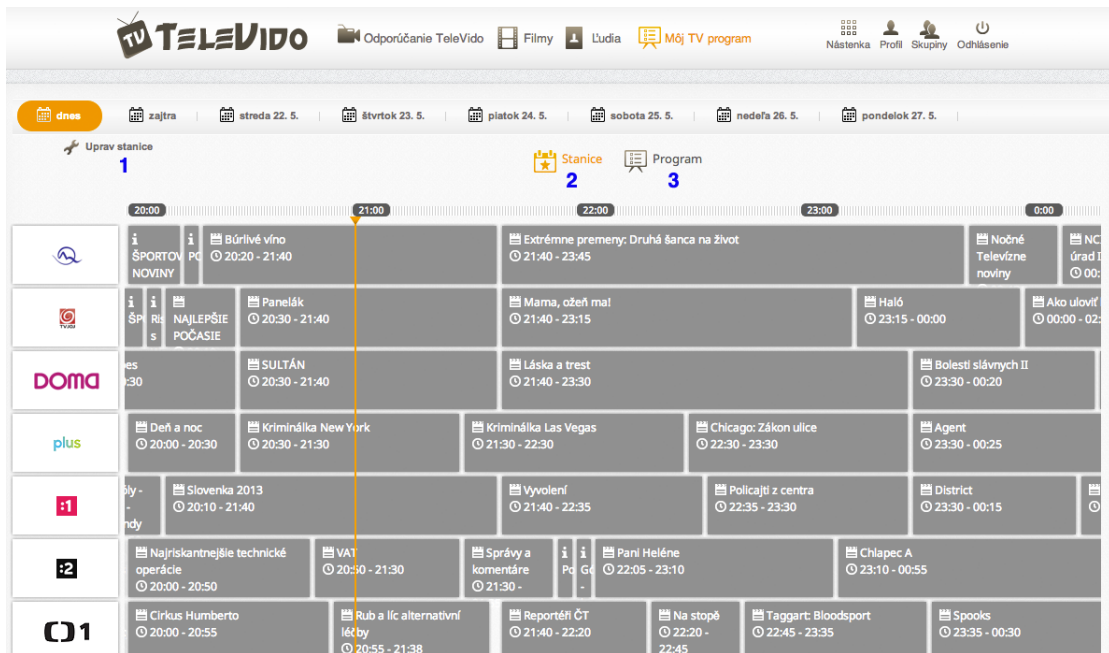
Obr. 74: Filmy, ktoré používateľ ohodnotil



Obr. 75: Followeri používateľa



Obr. 76: Hodnotenie filmu prihláseným používateľom



Obr. 77: Môj TV program - zobrazenie staníc a TV programu

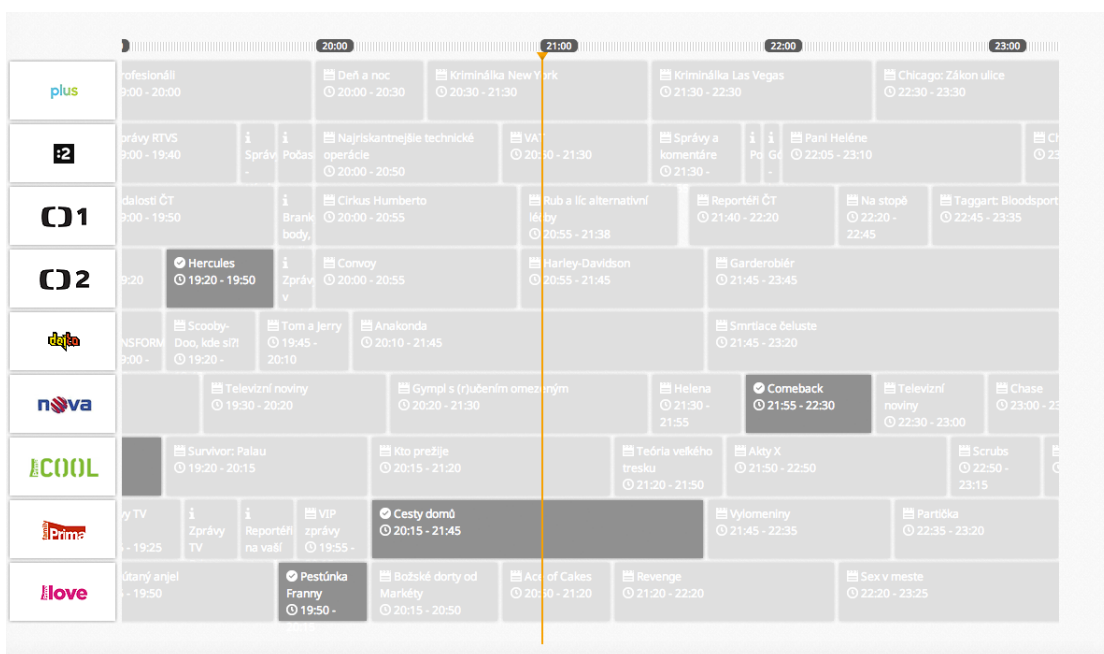
## 17.2.4 Môj TV Program

### Obrázok 77

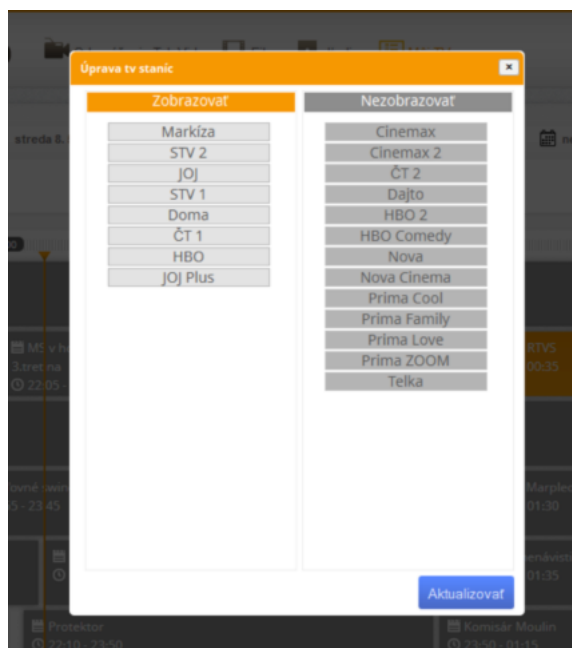
1. Možnosť upraviť stanice (ich poradie a počet obr. č 79).
2. Tlačidlo pre zobrazenie staníc a programu
3. Tlačidlo pre znázornenie personalizovaného TV programu

**Obrázok 78** Personalizovaný TV program - zvýraznené šedé sú odporúčané programy, oranžovou sú filmy, programy alebo seriály, ktoré používateľ označil, že chce vidieť.

**Obrázok 79** Výber TV staníc - presúvanie ich poradia a aj ich počtu. Používateľ presúva stanice pomocou drag and drop.



Obr. 78: Môj TV program - znázornenie personalizovaného TV programu



Obr. 79: Môj TV program - Výber TV staníc